

DHW 11

DHW-11 FUNC TST PART 2  
CVDHBCO

COPYRIGHT (c) 1983-84  
AH-T8550-MC  
FICHE 1 OF 2

APR 1984  
digital  
Made In USA

This microfiche card contains 144 frames of data, arranged in a 12x12 grid. Each frame displays a small, high-contrast image of a document page, likely a technical manual or test procedure. The frames are arranged in a regular grid pattern across the card. The text in the frames is too small to be legible, but they appear to contain various diagrams, tables, and text blocks. The overall appearance is that of a dense collection of micro-documented information.

DHV 11

DHV-11 FUNC TST PART 2  
CVDHBCO

COPYRIGHT (c) 1983-84  
AH-T655C-MC  
FICHE 2 OF 2

APR 1984

digital

Made In USA

.REM 8

IDENTIFICATION  
-----

PRODUCT CODE: AC-T654C-MC  
PRODUCT NAME: CVDHBCO DHV-11 FUNC TST PART2  
PRODUCT DATE: 16 DECEMBER 1983  
MAINTAINER: ENE DIAGNOSTICS GROUP  
AUTHOR: BERT KLEINSCHMIDT  
TONY GRIMSHAW  
MODIFIED BY: BERT KLEINSCHMIDT  
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) 1983, 1984 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL  
DEC

PDP  
DECUS

UNIBUS  
DECTAPE

MASSBUS

\*\*\*\*\* MODIFICATION HISTORY \*\*\*\*\*

ORIGINAL RELEASE: 31-OCT-83 (EDITED 11-JUL-83)  
BERT KLEINSCHMIDT

VERSION B0 09-OCT-83 BERT KLEINSCHMIDT  
FIXED TYPOGRAPHICAL ERRORS.  
MOVED TESTS FROM THIS PROGRAM TO CVDHA (PART 1):  
OLD CVDHB (VERSION A) TESTS 2 THROUGH 8 ARE  
NOW NEW CVDHA (VERSION B) TESTS 20 THROUGH 26.  
MOVED TEST FROM CVDHC (PART 3) INTO THIS PROGRAM:  
OLD CVDHC (VERSION A) TESTS 4 THROUGH 6 ARE  
NOW NEW CVDHB (VERSION B) TESTS 13 THROUGH 15.

VERSION C0 16-DEC-83 BERT KLEINSCHMIDT  
FIXED TYPOGRAPHICAL ERRORS DISCOVERED TO DATE.  
ADDED 2 TESTS TO ALLOW THIS PROGRAM TO BE USED FOR A  
COMPLETE CHECKOUT OF CABLES AND DISTRIBUTION PANELS.  
NEW TEST 24 - TESTS CONTINUITY OF TX/RX LINES.  
NEW TEST 25 - DETECTS TX LINE INTERACTIONS.

TABLE OF CONTENTS

1.0	GENERAL PROGRAM CONSIDERATIONS . . . . .	4
1.1	PROGRAM ABSTRACT . . . . .	4
1.2	SYSTEM REQUIREMENTS . . . . .	4
1.3	RELATED DOCUMENTS AND STANDARDS . . . . .	4
1.4	DIAGNOSTIC HIERARCY PREREQUISITES . . . . .	5
2.0	OPERATING INSTRUCTIONS . . . . .	5
2.1	COMMANDS . . . . .	5
2.2	SWITCHES . . . . .	6
2.3	FLAGS . . . . .	7
2.4	EXTENDED COMMAND SYNTAX . . . . .	8
2.4.1	START COMMAND . . . . .	8
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 . . . . .	10
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 . . . . .	10
2.4.3.1	FLAG SWITCH (/FLAGS:<FLAG-LIST>) . . . . .	
2.4.3.2	EFFECT OF CONTINUE COMMAND . . . . .	
2.4.4	PROCEED COMMAND . . . . .	11
2.4.4.1	FLAGS SWITCH (/FLAGS:<FLAG-LIST>) . . . . .	
2.4.4.2	EFFECT OF PROCEED COMMAND . . . . .	
2.4.5	ADD COMMAND . . . . .	11
2.4.6	EFFECT OF ADD COMMAND . . . . .	11
2.4.7	DROP COMMAND . . . . .	12
2.4.8	EFFECT OF DROP COMMAND . . . . .	12
2.4.9	PRINT COMMAND . . . . .	12
2.4.9.1	EFFECT OF PRINT COMMAND . . . . .	
2.4.10	DISPLAY COMMAND . . . . .	12
2.4.10.1	EFFECT OF DISPLAY COMMAND . . . . .	
2.4.11	FLAGS COMMAND . . . . .	12
2.4.11.1	EFFECT OF FLAGS COMMAND . . . . .	
2.4.12	ZFLAGS COMMAND . . . . .	13
2.4.13	ZFLAGS COMMAND . . . . .	13
2.4.14	CONTROL CHARACTERS . . . . .	13
2.5	HARDWARE QUESTIONS . . . . .	13
2.6	SOFTWARE QUESTIONS . . . . .	14
2.7	EXTENDED P-TABLE DIALOGUE . . . . .	15
2.8	QUICK START-UP PROCEDURE (XXDP*) . . . . .	17
3.0	ERROR INFORMATION . . . . .	17
3.1	TYPES OF ERROR MESSAGES . . . . .	17
3.2	ERROR MESSAGES . . . . .	18
4.0	PERFORMANCE AND PROGRESS REPORTS . . . . .	18
5.0	TEST SUMMARIES . . . . .	19
6.0	EXAMPLE ERROR FREE PASS . . . . .	21

## 1.0 GENERAL PROGRAM CONSIDERATIONS

### 1.1 PROGRAM ABSTRACT

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

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

### 1.2 SYSTEM REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DHV FVT:

- 0 LSI-11 PROCESSOR WITH AT LEAST 32 KBYTES OF RAM.
- 0 DHV11 BOARDS INSTALLED ON THE Q-BUS.
- 0 APPROPRIATE PROGRAM LOAD DEVICE SUPPORTING XXDP+ MEDIA OR A DOWN-LINE LOADING SYSTEM.

### 1.3 RELATED DOCUMENTS AND STANDARDS

- 0 DHV-11 HARDWARE MANUAL - THIS MANUAL DESCRIBES THE FUNCTIONS AND USES OF THE DHV-11 DEVICE.
- 0 XXDP+ USER'S MANUAL - DESCRIBES THE RUNNING OF DIAGNOSTICS UNDER THE XXDP+ MONITOR.

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

THE LSI-11 PROCESSOR, THE Q-BUS, 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 "DDDD".

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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



### 2.3 FLAGS

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

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

\*SEE THE ERROR INFORMATION SECTION OF THIS DOCUMENT.

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

/FLAGS:LOE:IER:BOE

## 2.4 EXTENDED COMMAND SYNTAX

### 2.4.1 START COMMAND -

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

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

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

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

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

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

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

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED.
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAINING THE ERROR.
IER	INHIBIT ERROR REPORTING.
IBE	INHIBIT BASIC ERROR REPORTS.
IXE	INHIBIT EXTENDED ERROR REPORTS.
PRI	DIRECT ALL MESSAGES TO A LINE PRINTER.
PNT	PRINT NUMBER OF TEST BEING EXECUTED.
BOE	BELL ON ERROR.
UAM	RUN IN UNATTENDED MODE, BYPASSING MANUAL

INTERVENTION.  
ISR INHIBIT STATISTICAL REPORTS.  
IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC.  
LOT LOOP ON TEST.

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

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

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

2.4.1.5 EFFECT OF START COMMAND -

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

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

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

EXAMPLE:

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

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

2.4.2 RESTART COMMAND -

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

2.4.2.1 TESTS, PASS, AND FLAGS SWITCHES -

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

2.4.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>) - <UNIT-LIST> IS  
A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF  
DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO  
BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE  
NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS  
SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER  
INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS  
ENTERED DURING THE HARDWARE 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 SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED  
(OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER  
COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL  
WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE, B)  
AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET, OR  
C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

2.4.3 CONTINUE COMMAND -

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

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

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

2.4.3.2 EFFECT OF CONTINUE COMMAND -

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

2.4.4 PROCEED COMMAND -

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

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

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

2.4.4.2 EFFECT OF PROCEED COMMAND -

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

2.4.5 ADD COMMAND -

\*\*\*\*\*  
ADD/UNITS:<UNIT-LIST>  
\*\*\*\*\*

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

2.4.7 DROP COMMAND -

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

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

2.4.9 PRINT COMMAND -

\*\*\*\*\*  
PRI(NT)  
\*\*\*\*\*

2.4.9.1 EFFECT OF PRINT COMMAND - ERROR SUMMARY REPORTING IS NOT IMPLEMENTED IN THIS DIAGNOSTIC, SO THIS COMMAND HAS NO EFFECT.

2.4.10 DISPLAY COMMAND -

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

2.4.10.1 EFFECT OF DISPLAY COMMAND -

THE HARDWARE P-TABLES FOR ALL UNITS ARE PRINTED IN THE FORMAT IN WHICH THEY WERE ENTERED.

2.4.11 FLAGS COMMAND -

\*\*\*\*\*  
FLA(GS)  
\*\*\*\*\*

2.4.11.1 EFFECT OF FLAGS COMMAND -

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

2.4.12 ZFLAGS COMMAND -

\*\*\*\*\*  
ZFL(AGS)  
\*\*\*\*\*

2.4.13 ZFLAGS COMMAND -

ALL FLAGS ARE CLEARED.

2.4.14 CONTROL CHARACTERS -

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

2.5 HARDWARE QUESTIONS

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

1. CSR ADDRESS - THIS QUESTION REQUESTS THE CSR ADDRESS OF THE SPECIFIED DHV11. THE DEFAULT ANSWER FOR THIS QUESTION IS THE LOWEST ADDRESS IN THE PDP-11 FLOATING ADDRESS SPACE IN WHICH A DHV-11 CAN BE PLACED (160460 OCTAL).
2. INTERRUPT VECTOR ADDRESS - THIS QUESTION REQUESTS THE INTERRUPT VECTOR ADDRESS OF THE SPECIFIED DHV11.
3. ACTIVE LINES BIT MAP - THIS QUESTION REQUESTS AN OCTAL BIT MAP OF THE SERIAL COMMUNICATION LINES ON THE DHV11 WHICH ARE BEING SELECTED FOR TESTING. IF THE BIT IN THE BIT MAP IS SET WHICH CORRESPONDS TO A PARTICULAR LINE (I.E. BIT 3 FOR LINE

- 3) THAT LINE WILL BE TESTED BY THE FVT. WITH STAGGERED LOOPBACK A PAIR OF LINES WITH THE SPECIFIED TRANSMIT LINE AND ANOTHER RECEIVE LINE WILL BE TESTED. THEREFORE, TO GUARANTEE THAT BOTH THE TRANSMITTER AND RECEIVER OF A SPECIFIED LINE ARE TESTED WHEN USING THE STAGGERED LOOPBACK CONNECTOR, BOTH THE INTENDED LINE AND ITS MATE MUST BE SELECTED (IE. TO TEST LINE 1, SELECT BOTH LINE 1 AND LINE 3). IN NONSTAGGERED TESTING, A BIT IN THE ACTIVE LINES BIT MAP SELECTS THE TRANSMITTER AND RECEIVER FOR THE SAME LINE.
4. TYPE OF LOOPBACK (1=INTERNAL, 2=H3277, 3=H325) - THIS QUESTION REQUESTS THE TYPE OF LOOPBACK TO BE USED IN TESTING THE DHV11. THE FOLLOWING TYPES OF LOOPBACK ARE SUPPORTED:
- 0 INTERNAL - ONLY INTERNAL UART LOOPBACK IS TO BE USED IN TESTING THE DHV.
  - 0 H3277 - STAGGERED BERG CONNECTOR(S) ARE INSTALLED AT THE END OF THE 40 WIRE CABLES IN PLACE OF THE DHV11 DISTRIBUTION PANELS.
  - 0 H325 - SINGLE LINE, 25 PIN LOOPBACK CONNECTORS (TYPE H325) ARE INSTALLED ON THE LINES TO BE TESTED. THESE CONNECTORS CAN BE INSTALLED ON THE DISTRIBUTION PANEL OR ON THE END OF THE TERMINAL OR MODEM CABLE. THE H325 CONNECTORS MUST HAVE THE REMOVABLE JUMPERS INSTALLED.
5. BR LEVEL - THIS QUESTIONS REQUESTS THE INTERRUPT BR LEVEL OF THE DHV11.

## 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 FOLLOWING SOFTWARE P-TABLE QUESTIONS ARE ASKED BY THE PROGRAM IF THE OPERATOR INDICATES THAT THE SOFTWARE PARAMETERS ARE TO BE CHANGED:

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 EACH UNIT. THE UNIT NUMBER WILL ONLY BE REPORTED IF MORE THAN ONE UNIT IS BEING TESTED.
2. NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE - THIS 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) ? 160000<CR>  
SUB-DEVICE # (0) ? 0-7<CR>  
Q-FACTOR (0) 0 ? 0,1,0,....,1,1<CR>

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

## 2.8 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

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

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

## 3.0 ERROR INFORMATION

### 3.1 TYPES OF ERROR MESSAGES

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

THE GENERAL ERROR MESSAGE IS OF THE FORM:

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

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

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

DOCUMENT).  
THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL MESSAGE.

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

3.2 ERROR MESSAGES

THIS PROGRAM IS INTENDED TO PROVIDE A GO/NO-GO INDICATION OF THE FUNCTIONALITY OF DHV-11 BOARDS. TO EXECUTE THE PROGRAM IN THIS MODE THE OPERATOR CAN RUN WITH THE INHIBIT BASIC ERROR REPORTING SWITCH. IN THIS MODE THE PROGRAM PRINTS ERROR MESSAGES WHICH CONTAIN THE ERROR MESSAGE HEADER DESCRIBED ABOVE, PLUS THE NAME OF THE FAILING TEST. FOR A LIST OF THE TEST NAMES IN THIS PROGRAM SEE THE TEST SUMMARIES SECTION OF THIS DOCUMENT. AN EXAMPLE OF SUCH AN ERROR MESSAGE IS THE FOLLOWING:

CVDHB DVC FTL ERR 01603 ON UNIT 02 TST 015 SUB 000 PC: 015244  
DEVICE REGISTER WORD READ/WRITE TEST

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

IF THE OPERATOR REQUIRES MORE EXTENSIVE ERROR REPORTING HE CAN RUN WITH ALL ERROR REPORTING ENABLED BY NOT USING THE INHIBIT REPORTING SWITCHES. THE ABOVE ERROR MESSAGE WOULD THEN BECOME THE FOLLOWING:

CVDHB DVC FTL ERR 01603 ON UNIT 02 TST 015 SUB 000 PC: 015244  
DEVICE REGISTER WORD READ/WRITE TEST  
BAD BIT(S) IN DEVICE TBUFAD1 REGISTER FOR LINE 7 (D).  
EXPECTED DATA: 000000 (0).  
ACTUAL DATA: 000023 (0).

4.0 PERFORMANCE AND PROGRESS REPORTS

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

## 5.0 TEST SUMMARIES

THE FOLLOWING TESTS ARE INCLUDED WITHIN CVDHB:

1. DEVICE REGISTER ADDRESS TEST - VERIFIES THAT THE UUT REGISTERS WILL RESPOND WITH THE PROPER Q-BUS HANDSHAKING WHEN ACCESSED. VERIFIES THAT THE UUT IS AT THE PROPER ADDRESS.
2. DMA.START TEST - VERIFIES THAT EACH DMA.START BIT WILL INITIATE A DMA TX ON A LINE.
3. DMA.ABORT TEST - VERIFIES THAT THE DMA.ABORT BIT ON EACH LINE WILL STOP A DMA TRANSMISSION AND RETURN A TX.ACTION AND THAT THE DMA CAN THEN BE RESTARTED.
4. O.AUTO INACTIVE TEST - VERIFIES THAT THE UUT WILL NOT RESPOND TO INCOMING XON AND XOFF CHARACTERS WHEN O.AUTO IS INACTIVE.
5. O.AUTO ACTIVE TEST - VERIFIES THAT THE UUT RESPOND CORRECTLY TO INCOMING XON AND XOFF CHARACTERS WHEN O.AUTO IS ACTIVE.
6. I.AUTO INACTIVE TEST - VERIFIES THAT THE UUT WILL NOT GENERATE AND TX XON OR XOFF CHARACTERS IN RESPONSE TO THE FIFO CONDITIONS IF THE I.AUTO BIT IS INACTIVE.
7. I.AUTO ACTIVE TEST - VERIFIES THAT THE UUT WILL GENERATE AND TX XON AND XOFF CHARACTERS IN RESPONSE TO THE FIFO CONDITIONS IF THE I.AUTO BIT IS ACTIVE.
8. FIFO DATA TEST - VERIFIES THAT THE FIFO WILL HOLD 256 CHARACTERS WITHOUT CORRUPTING DATA.
9. FIFO 3/4 LEVEL INACTIVE TEST - VERIFIES THAT THE FIFO 3/4 ALARM DOES NOT BECOME ACTIVE UNTIL THE FIFO BECOMES 3/4 FULL.
10. FIFO 3/4 LEVEL ACTIVE TEST - VERIFIES THAT THE FIFO 3/4 ALARM BECOMES ACTIVE, AND REMAINS ACTIVE, WHEN THE FIFO IS MORE THAN 3/4 FULL.
11. FIFO 3/4 LEVEL ACTIVE/INACTIVE TEST - VERIFIES THAT THE FIFO 3/4 ALARM, ONCE ACTIVATED, REMAINS ACTIVE UNTIL THE FIFO LEVEL IS REDUCED BELOW 1/2.
12. FIFO 1/2 LEVEL TEST - VERIFIES THAT FIFO 1/2 LEVEL INDICATOR BECOMES ACTIVE AND REMAINS ACTIVE AS THE FIFO LEVEL IS REDUCED BELOW THE 1/2 FULL POINT.
13. BREAK TEST - VERIFIES THAT SETTING THE BREAK BIT ON ANY LINE CAUSES THAT LINE TO GO TO A SPACING STATE AND THAT CLEARING THE BREAK BIT REMOVES THE LINE FROM THE SPACING STATE.
14. NO OVERRUN.ERROR TEST - VERIFIES THAT THE UUT WILL RECEIVE THE MAXIMUM NUMBER OF CHARACTERS WITHOUT CAUSING AN OVERRUN ERROR.

15. OVERRUN.ERROR TEST - VERIFIES THAT IF MORE THAN THE MAXIMUM NUMBER OF CHARACTERS ARE SENT TO THE UUT OVERRUN ERRORS OCCUR.
16. DTR TEST - VERIFIES THAT CHANGING THE UUT LNCTRL DTR BIT AFFECTS THE STATE OF THE DTR CONTROL LINE.
17. RTS TEST - VERIFIES THAT CHANGING THE UUT LNCTRL RTS BIT AFFECTS THE STATE OF THE RTS CONTROL LINE.
18. DSR TEST - VERIFIES THAT DSR STATUS SIGNAL CORRECTLY REPORTS THE STATE OF THE LOOPED BACK DTR CONTROL LINE.
19. RI TEST - VERIFIES THAT RI STATUS SIGNAL CORRECTLY REPORTS THE STATE OF THE LOOPED BACK DTR CONTROL LINE.
20. CTS TEST - VERIFIES THAT CTS STATUS SIGNAL CORRECTLY REPORTS THE STATE OF THE LOOPED BACK RTS CONTROL LINE.
21. DCD TEST - VERIFIES THAT DCD STATUS SIGNAL CORRECTLY REPORTS THE STATE OF THE LOOPED BACK RTS CONTROL LINE.
22. 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.
23. 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.
24. TX LINES TEST - VERIFIES THAT EACH ACTIVE LINE HAS CONTINUITY TO THE CORRECT RECEIVE LINE. THIS TEST IS ONLY RUN IF H325 OR H3277 LOOPBACK IS SPECIFIED IN THE HARDWARE QUESTIONS DIALOGUE. THIS TEST IS INTENDED TO BE USED TO VERIFY CABLE AND DISTRIBUTION PANEL DATA LINE CONNECTIONS.
25. TX LINES INTERACTIONS TEST - VERIFIES THAT EACH ACTIVE TX LINE DOES NOT INFLUENCE ANY RX LINES (EXCEPT THE ONE THAT IT IS LOOPED BACK TO) OR ANY MODEM STATUS SIGNALS. THIS TEST IS ONLY RUN IF H325 OR H3277 LOOPBACK IS SPECIFIED IN THE HARDWARE QUESTIONS DIALOGUE.
26. REPORT BMP CODES TEST - THIS PSEUDO TEST REPORTS THE FIRST 32 BMP CODES WHICH WERE DISCOVERED IN THE FIFO DURING THE EXECUTION OF THE OTHER TESTS. THIS AVOIDS THE INTERRUPTION OF OTHER TESTS BY THESE CODES, IF THEY ARE NOT CRITICAL TO THE TESTS BEING PERFORMED.

6.0 EXAMPLE ERROR FREE PASS

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

.R CVDHBCO  
CVDHBCO.BIC

DRS  
CVDHB-C-0  
DHV-11 FUNC TST PART2  
UNIT IS DHV-11  
RESTART ADDR: 147670  
DR>STA

CHANGE HW (L) ? Y

# UNITS (D) ? 2

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

UNIT 1  
CSR ADDRESS: (0) 160460 ? 160040  
INTERRUPT VECTOR ADDRESS: (0) 300 ? 320  
ACTIVE LINE BIT MAP: (0) 377 ? <CR>  
TYPE OF LOOPBACK (1=INTERNAL, 2=H3277, 3=H325): (0) 2 ? 1  
INTERRUPT BR LEVEL: (0) 4? <CR>

CHANGE SW (L) ? Y

REPORT UNIT NUMBER AS EACH UNIT IS TESTED: (L) Y ? <CR>  
NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: (D) 0 ? 4

TESTING UNIT : 0(D)

TESTING UNIT : 1(D)

CVDHB EOP 1  
0 CUMULATIVE ERRORS

TESTING UNIT : 0(D)

+C  
DR> EXIT

```

1082          .LIST SEQ,LOC,BIN,MEB
1083          .NLIST CND
1084
1085
1086          ;*****
1087          ;
1088          ;           FVTA.PHD
1089          ;
1090          ;*****
1091
1092
1093
1094          .SBTTL PROGRAM HEADER
1095
1096
1097          .MCALL SVC
1098 000000          SVC           ; INITIALIZE SUPERVISOR MACROS
1099
1100          ;*****
1101          ; IF STRUCTURED MACROS ARE TO BE USED, ADD ".MCALL STRUCT" AND "STRUCT"
1102          ; TO INITIALIZE THE STRUCTURED MACROS.
1103
1104          000001          SVCINS= 1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
1105          000001          SVCTST= 1     ; LIST TEST TAGS, SHIFTED RIGHT
1106          000001          SVCSUB= 1    ; LIST SUBTEST TAGS, SHIFTED RIGHT
1107          000001          SVCGBL= 1   ; LIST GLOBAL TAGS, SHIFTED RIGHT
1108          000001          SVCTAG= 1   ; LIST OTHER TAGS, SHIFTED RIGHT
1109
1110          ; CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1111          ; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
1112          ; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
1113          ; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1114          ;*****
1115
1116 000000          .ENABL ABS
1117          ;.ENABL AMA
1118          "           "           2000
1119
1120 002000          BGNMOD
1121
1122          ;**
1123          ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1124          ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1125          ;--
1126
1127 002000          POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL
1128
1145
1146          002000          HEADER CVDHB,B.0,22.0,PRI07
1147
1148          002000          L$NAME::
1149          002001          .ASCII /C/
1150          002002          .ASCII /V/
1151          002003          .ASCII /D/
1152          002004          .ASCII /H/
1153          002005          .ASCII /B/
1154          002006          .BYTE 0
1155          002007          .BYTE 0

```



```

002007      000
002010
002010      102
002011
002011      060
002012
002012     000000
002014
002014     000022
002016
002016     036164
002020
002020     036446
002022
002022     002212
002024
002024     002224
002026
002026     036706
002030
002030     000000
002032
002032     000000
002034
002034     000000
002036
002036     000000
002040
002040     002124
002042
002042     000340
002044
002044     000000
002046
002046     000000
002050
002050       003
002051       003
002052
002052     000000
002054
002054     000000
002056
002056     000000
002060
002060     004154
002062
002062     017062
002064
002064     000000
002066
002066     000000
002070
002070     000000
002072
002072     017754
002074
002074     000000

```

```

L$REV:: .BYTE 0
L$DEPO:: .ASCII /B/
L$UNIT:: .ASCII /O/
L$TIML:: .WORD 0
L$HPCP:: .WORD 22
L$SPCP:: .WORD L$HARD
L$HPTP:: .WORD L$SOFT
L$SPTP:: .WORD L$HW
L$LADP:: .WORD L$SW
L$STA:: .WORD L$LAST
L$CO:: .WORD 0
L$DTYP:: .WORD 0
L$APT:: .WORD 0
L$DTP:: .WORD 0
L$PRIO:: .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
        .WORD 0

```

PROGRAM HEADER

002076  
002076 004164  
002100  
002100 104035  
002102  
002102 004104  
002104  
002104 017076  
002106  
002106 017736  
002110  
002110 017734  
002112  
002112 017070  
002114  
002114 000000  
002116  
002116 000000  
002120  
002120 000000

1147

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

1159  
 1160  
 1161  
 1162  
 1163  
 1164  
 1165  
 1166 002122  
 002122 000032  
 002124  
 002124 020072  
 002126 020362  
 002130 020756  
 002132 021404  
 002134 022152  
 002136 022720  
 002140 023322  
 002142 023744  
 002144 024242  
 002146 024552  
 002150 025252  
 002152 025750  
 002154 026422  
 002156 027046  
 002160 027554  
 002162 030510  
 002164 031224  
 002166 031740  
 002170 032370  
 002172 033020  
 002174 033450  
 002176 034100  
 002200 034510  
 002202 035120  
 002204 035452  
 002206 036102  
 1167

.SBTTL DISPATCH TABLE

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

DISPATCH 26

.WORD 26  
 L\$DISPATCH::  
 .WORD T1  
 .WORD T2  
 .WORD T3  
 .WORD T4  
 .WORD T5  
 .WORD T6  
 .WORD T7  
 .WORD T8  
 .WORD T9  
 .WORD T10  
 .WORD T11  
 .WORD T12  
 .WORD T13  
 .WORD T14  
 .WORD T15  
 .WORD T16  
 .WORD T17  
 .WORD T18  
 .WORD T19  
 .WORD T20  
 .WORD T21  
 .WORD T22  
 .WORD T23  
 .WORD T24  
 .WORD T25  
 .WORD T26

```

1175
1176
1177
1178
1179
1180
*****
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193 002210
      002210 000004
      002212
      002212
1194
1195 002212 160460
1196 002214 000300
1197 002216 177777
1198 002220 002
1199 002221 004
1200
1201 002222
      002222

```

```

:*****
:
:           FVTA.DHT
:
:*****

```

```

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

```

```

      BGNHW  DFPTBL

```

```

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

```

```

      .WORD  160460 ;DEFAULT CSR ADDRESS
      .WORD  300    ;DEFAULT VECTOR ADDRESS
      .WORD  177777 ;DEFAULT ACTIVE LINES BIT MAP
      .BYTE  2      ;DEFAULT LOOPBACK MODE
      .BYTE  4      ;DEFAULT BR LEVEL

```

```

      ENDPHW

```

```

      L10000:

```

1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226

002222  
002222 000002  
002224  
002224  
002224 000020  
002226 000000  
002230  
002230

```

;*****
;
;           FVTA.SWT
;
;*****

.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:

```

```
1228
1229 ;*****
1230 ;
1231 ;           FVTA.EQU
1232 ;
1233 ;*****
1234
1235
1236 .SBTTL GLOBAL EQUATES SECTION
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251 ;**
1252 ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
1253 ; ARE USED IN MORE THAN ONE TEST.
1254 ;--
1255           000010           NUMLNS==10           ;NUMBER OF LINES ON DHV11 IS 8.
1256           000377           MAPLNS==377          ;BIT MAP OF LINES ON DHV11.
1257
1258 ;***** DEVICE REGISTER OFFSETS FROM THE CSR'S ADDRESS *****
1259           000000           CSRO==0             ;CSR REGISTER OFFSET FROM THE CSR ADDRESS
1260           000002           RBUFO==2            ;RECEIVE REGISTER OFFSET FROM THE CSR ADDRESS
1261           000002           TXCHRO==2           ;TRANSMIT REGISTER OFFSET FROM THE CSR ADDRESS
1262           000004           LPRO==4             ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
1263           000006           STATO==6           ;STATUS REGISTER OFFSET FROM THE CSR ADDRESS
1264           000010           LNCTRO==10          ;LINE CONTROL REGISTER OFFSET FROM THE CSR ADDRESS
1265           000012           TXAD10==12          ;TRANSMIT ADDRESS 1 REGISTER OFFSET FROM THE CSR ADDRESS
1266           000014           TXAD20==14          ;TRANSMIT ADDRESS 2 REGISTER OFFSET FROM THE CSR ADDRESS
1267           000016           TXBFCO==16          ;TRANSMIT COUNT REGISTER OFFSET FROM THE CSR ADDRESS
1268
1269 ;***** EQUATES USED WITH RESPECT TO THE RX BUFFER *****
1270           000020           RXBETX==16.          ;LEVEL OF RX BUFFER AT WHICH TO RE-ENABLE TRANSMISSION.
1271           000030           RXBDTX==24.          ;LEVEL OF RX BUFFER AT WHICH TO DISABLE TRANSMISSION.
1272           000100           RXBFUL==64.          ;TOTAL CHARACTER CAPACITY OF THE RX BUFFER.
1273
1274
1275
1276
1277
1278
1279           002230           EQUALS
1280
1281 ;
1282 ; BIT DIFINITIONS
1283 ;
1284           100000           BIT15== 100000
1285           040000           BIT14== 40000
1286           020000           BIT13== 20000
1287           010000           BIT12== 10000
1288           004000           BIT11== 4000
1289           002000           BIT10== 2000
1290           001000           BIT09== 1000
1291           000400           BIT08== 400
1292           000200           BIT07== 200
1293           000100           BIT06== 100
1294           000040           BIT05== 40
1295           000020           BIT04== 20
1296           000010           BIT03== 10
1297           000004           BIT02== 4
1298           000002           BIT01== 2
```

```

000001      BIT00-- 1
001000      BIT9--  BIT09
000400      BIT8--  BIT08
000200      BIT7--  BIT07
000100      BIT6--  BIT06
000040      BIT5--  BIT05
000020      BIT4--  BIT04
000010      BIT3--  BIT03
000004      BIT2--  BIT02
000002      BIT1--  BIT01
000001      BIT0--  BIT00
;
; EVENT FLAG DEFINITIONS
; EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
;
000040      EF.START--      32.      ; START COMMAND WAS ISSUED
000037      EF.RESTART--    31.      ; RESTART COMMAND WAS ISSUED
000036      EF.CONTINUE--   30.      ; CONTINUE COMMAND WAS ISSUED
000035      EF.NEW--        29.      ; A NEW PASS HAS BEEN STARTED
000034      EF.PWR--        28.      ; A POWER-FAIL/POWER-UP OCCURRED
;
; PRIORITY LEVEL DEFINITIONS
;
000340      PRI07-- 340
000300      PRI06-- 300
000240      PRI05-- 240
000200      PRI04-- 200
000140      PRI03-- 140
000100      PRI02-- 100
000040      PRI01-- 40
000000      PRI00-- 0
;
; OPERATOR FLAG BITS
;
000004      EVL--      4
000010      LOT--     10
000020      ADR--     20
000040      IDU--     40
000100      ISR--    100
000200      UAM--    200
000400      BOE--    400
001000      PNT--   1000
002000      PRI--   2000
004000      IXE--   4000
010000      IBE--  10000
020000      IER--  20000
040000      LOE--  40000
100000      HOE-- 100000

```

1292  
 1293  
 1294  
 1295  
 1296  
 1297  
 1298  
 1299  
 1300  
 1301  
 1302  
 1303  
 1304  
 1305  
 1306  
 1307  
 1308  
 1309  
 1310  
 1311  
 1312 002230 000300  
 1313 002232 000304  
 1314 002234 000377  
 1315 002236 000  
 1316 002237 004  
 1317 002240 000000  
 1318  
 1319  
 1320  
 1321  
 1322 002242  
 1323 002242 160000  
 1324 002244 160002  
 1325 002246 160004  
 1326 002250 160006  
 1327 002252 160010  
 1328 002254 160012  
 1329 002256 160014  
 1330 002260 160016  
 1331  
 1332  
 1333  
 1334  
 1335 002262 000000  
 1336 002264 000000  
 1337 002266 000001  
 1338 002270 000000  
 1339 002272 031463  
 1340 002274 146314  
 1341 002276 000000  
 1342 002300 000000  
 1343 002302 000000  
 1344 002304 000000  
 1345 002306 000000  
 1346 002310 000000  
 1347 002312 000000  
 1348 002314 000000

```

;*****
;
;           FVTB.GDT
;
;*****

.SBTTL  GLOBAL DATA SECTION

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

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

          RXVECA:: .WORD 300      ;RX VECTOR ADDRESS.
          TXVECA:: .WORD 304      ;TX VECTOR ADDRESS.
          ACTLNS:: .WORD 377     ;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
;*****
TXCHA::  DRADRT::
          CSRA:: .WORD 160000    ;DHV-11 CSR ADDRESS
          RBUFA:: .WORD 160002   ;DHV-11 RECEIVE/TRANSMIT BUFFER ADDRESS
          LPRA:: .WORD 160004    ;DHV-11 LINE PARAMETER REGISTER ADDRESS
          STATA:: .WORD 160006   ;DHV-11 STATUS REGISTER ADDRESS
          LNCTRA:: .WORD 160010  ;DHV-11 LINE CONTROL REGISTER ADDRESS
          TXAD1A:: .WORD 160012  ;DHV-11 TRANSMIT BUFFER 1 REGISTER ADDRESS
          TXAD2A:: .WORD 160014  ;DHV-11 TRANSMIT BUFFER 2 REGISTER ADDRESS
          TXBFCA:: .WORD 160016  ;DHV-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.
          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 NORMAL 004 TRAP VECTOR.
          TP4FLG:: .WORD 0       ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
          WORD1::  .WORD 0       ;LOCATION FOR PASSING INDIRECT PARAMETERS.

```



1349  
 1350  
 1351  
 1352 002316 177546  
 1353 002320 000300  
 1354 002322 000100  
 1355 002324 000074  
 1356 002326 000000  
 1357 002330 000000  
 1358 002332 000170  
 1359 002334 000170  
 1360 002336 000021  
 1361 002340 000062  
 1362  
 1363  
 1364  
 1365  
 1366 002342 177572  
 1367 002344 000000  
 1368 002346 000000  
 1369  
 1370 002350  
 1371 002350 172340  
 1372 002352 172342  
 1373 002354 172344  
 1374 002356 172346  
 1375 002360 172350  
 1376 002362 172352  
 1377 002364 172354  
 1378 002366 172356  
 1379 002370  
 1380  
 1381  
 1382  
 1383 002370 000001  
 1384 002372 000002  
 1385 002374 000004  
 1386 002376 000010  
 1387 002400 000020  
 1388 002402 000040  
 1389 002404 000100  
 1390 002406 000200  
 1391 002410 000400  
 1392 002412 001000  
 1393 002414 002000  
 1394 002416 004000  
 1395 002420 010000  
 1396 002422 020000  
 1397 002424 040000  
 1398 002426 100000  
 1399  
 1400  
 1401  
 1402  
 1403 002430  
 1404 002430 000000  
 1405 002432 000000

```

;*****
; LINE TIME CLOCK VARIABLES AND STORAGE.
;*****
CLKCSR:: .WORD 177546 ;CSR ADDRESS OF THE LTC.
CLKBRL:: .WORD PRI06 ;INTERRUPT PRIORITY LEVEL OF THE LTC.
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).

PARATB:: ;BASE OF MEM MGT PAR ADDRESS TABLE.
PAR0A:: .WORD 172340 ;ADDRESS OF MEM MGT PAR #0.
PAR1A:: .WORD 172342 ;ADDRESS OF MEM MGT PAR #1.
PAR2A:: .WORD 172344 ;ADDRESS OF MEM MGT PAR #2.
PAR3A:: .WORD 172346 ;ADDRESS OF MEM MGT PAR #3.
PAR4A:: .WORD 172350 ;ADDRESS OF MEM MGT PAR #4.
PAR5A:: .WORD 172352 ;ADDRESS OF MEM MGT PAR #5.
PAR6A:: .WORD 172354 ;ADDRESS OF MEM MGT PAR #6.
PAR7A:: .WORD 172356 ;ADDRESS OF MEM MGT PAR #7.

PARATE:: ;END OF PAR ADDRESS TABLE.

;*****
; 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.

;*****
; * GPR SAVE AREAS ZERO AND ONE.
;*****
GPRS0B:: ;BASE OF GPR SAVE AREA NUMBER ZERO.
; .WORD 0 ;WORD 1, STORAGE FOR R1.
; .WORD 0 ;WORD 2, STORAGE FOR R2.

```

```

1406 002434 000000          .WORD 0          ;WORD 3, STORAGE FOR R3.
1407 002436 000000          .WORD 0          ;WORD 4, STORAGE FOR R4.
1408 002440 000000          .WORD 0          ;WORD 5, STORAGE FOR R5.
1409
1410          ;*****
          ; STORAGE AREA FOR THE BMP CODE QUEUE.
1411          ;*****
1412 002442 000000  BMPCQP::          .WORD 0          ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
1413 002444          BMPCQB::          .BLKW 64.        ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
1414 002644          BMPCQE::          ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1415          ;*****
1416          ; STORAGE AREA FOR THE CONTENTS OF THE DUT STAT REGISTER STATES.
1417          ;*****
1418 002644          STSTB::          ;BASE OF DUT STAT STORAGE TABLE.
1419 002644 000000          .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 0.
1420 002646 000000          .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 1.
1421 002650 000000          .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 2.
1422 002652 000000          .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 3.
1423 002654 000000          .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 4.
1424 002656 000000          .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 5.
1425 002660 000000          .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 6.
1426 002662 000000          .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 7.
1427 002664 000000          .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 8.
1428 002666 000000          .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 9.
1429 002670 000000          .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 10.
1430 002672 000000         .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 11.
1431 002674 000000         .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 12.
1432 002676 000000         .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 13.
1433 002700 000000         .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 14.
1434 002702 000000         .WORD 0          ;STORAGE FOR STAT REGISTER FOR LINE 15.
1435 002704          STSTE::          ;END OF DUT STAT STORAGE TABLE.
1436          ;*****
1437          ; GENERAL TABLE AND BUFFER AREA--513 WORDS.
1438          ;*****
1439 002704          BUFBAS::          ;BASE OF MEMORY BUFFER.
1440 002704          ERLTBL::          .BLKW 128.       ;FIRST HALF OF GENERAL TABLE OR BUFFER.
1441 003304          BUFMID::          .BLKW 64.        ;SECOND HALF OF GENERAL TABLE OR BUFFER.
1442 003504          BUF3QT::          .BLKW 64.        ;LAST QUARTER OF THE BUFFER AREA.
1443 003704          BUFEND::          ;END OF GENERAL PURPOSE MEMORY BUFFER.
1444 003704          ENDETB::          .BLKW 16.        ;BUFFER OVERFLOW SPACE.
1445          ;*****
1446          ; RECEPTION TABLE OF COUNTERS
1447          ;*****
1448 003744          RXCNTB::          .BLKW 16.        ;RECEPTION CHARACTER COUNTERS TABLE.
1449          ;*****
1450          ;* TABLE FOR STORAGE OF RX/TX LINE NUMBER ASSOCIATIONS.
1451          ;* THE ASSOCIATIONS ARE STORED AS LINE NUMBER TIMES 2 FOR USE AS OFFSETS
1452          ;* WHEN ACCESSING A TABLE OF WORDS.
1453          ;* NOTE: DO NOT WRITE A NON-ZERO VALUE INTO THE UPPER BYTE OF ANY ENTRY.
1454          ;*****
1455 004004          TXRXLB::          ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1456 004004 000000          .WORD 0          ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
1457 004006 000002          .WORD 2.          ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
1458 004010 000004          .WORD 4.          ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
1459 004012 000006          .WORD 6.          ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
1460 004014 000010          .WORD 8.          ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
1461 004016 000012          .WORD 10.         ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
1462 004020 000014          .WORD 12.         ;TX/RX LINE OFFSET FOR RX/TX LINE 6.

```

1463 004022 000016  
 1464 004024 000020  
 1465 004026 000022  
 1466 004030 000024  
 1467 004032 000026  
 1468 004034 000030  
 1469 004036 000032  
 1470 004040 000034  
 1471 004042 000036  
 1472 004044  
 1473  
 1474  
 1475  
 1476  
 1477  
 1478  
 1479 004044  
 1480 004044 000  
 1481 004045 001  
 1482 004046 002  
 1483 004047 003  
 1484 004050 004  
 1485 004051 005  
 1486 004052 006  
 1487 004053 007  
 1488 004054 010  
 1489 004055 011  
 1490 004056 012  
 1491 004057 013  
 1492 004060 014  
 1493 004061 015  
 1494 004062 016  
 1495 004063 017  
 1496 004064  
 1497  
 1498  
 1499  
 1500  
 1501  
 1502  
 1503  
 1504  
 1505 004064  
 1506 004064 004  
 1507 004065 006  
 1508 004066 000  
 1509 004067 002  
 1510 004070 014  
 1511 004071 016  
 1512 004072 010  
 1513 004073 012  
 1514 004074 024  
 1515 004075 026  
 1516 004076 020  
 1517 004077 022  
 1518 004100 034  
 1519 004101 036

```

      .WORD 14.      ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
      .WORD 16.      ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
      .WORD 18.      ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
      .WORD 20.      ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
      .WORD 22.      ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
      .WORD 24.      ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
      .WORD 26.      ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
      .WORD 28.      ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
      .WORD 30.      ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
TXRXLE::
      .EVEN          ;END OF TX/RX LINE NUMBER ASSOCIATION TABLE.
                    ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
;*****
;* TABLE FOR STORAGE OF RX/TX LINE NUMBER ASSOCIATIONS.
;* THE ASSOCIATIONS ARE STORED AS LINE NUMBERS WHICH CAN BE USED AS SUCH OR
;* AS OFFSETS WHEN ACCESSING A TABLE OF BYTES.
;*****
TXRLNB::
      .BYTE 0        ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
      .BYTE 1.      ;TX/RX LINE FOR RX/TX LINE 0.
      .BYTE 2.      ;TX/RX LINE FOR RX/TX LINE 1.
      .BYTE 3.      ;TX/RX LINE FOR RX/TX LINE 2.
      .BYTE 4.      ;TX/RX LINE FOR RX/TX LINE 3.
      .BYTE 5.      ;TX/RX LINE FOR RX/TX LINE 4.
      .BYTE 6.      ;TX/RX LINE FOR RX/TX LINE 5.
      .BYTE 7.      ;TX/RX LINE FOR RX/TX LINE 6.
      .BYTE 8.      ;TX/RX LINE FOR RX/TX LINE 7.
      .BYTE 9.      ;TX/RX LINE FOR RX/TX LINE 8.
      .BYTE 10.     ;TX/RX LINE FOR RX/TX LINE 9.
      .BYTE 11.     ;TX/RX LINE FOR RX/TX LINE 10.
      .BYTE 12.     ;TX/RX LINE FOR RX/TX LINE 11.
      .BYTE 13.     ;TX/RX LINE FOR RX/TX LINE 12.
      .BYTE 14.     ;TX/RX LINE FOR RX/TX LINE 13.
      .BYTE 15.     ;TX/RX LINE FOR RX/TX LINE 14.
      .BYTE 15.     ;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.      ;BASE OF STAGGERED TX/RX LINE NUMBER TABLE.
      .BYTE 6.      ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
      .BYTE 0.      ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
      .BYTE 2.      ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
      .BYTE 12.     ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
      .BYTE 14.     ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
      .BYTE 8.      ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
      .BYTE 10.     ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
      .BYTE 20.     ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
      .BYTE 22.     ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
      .BYTE 16.     ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
      .BYTE 18.     ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
      .BYTE 28.     ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
      .BYTE 30.     ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
      .BYTE 30.     ;TX/RX LINE OFFSET FOR RX/TX LINE 13.

```

1520 004102 030  
1521 004103 032  
1522  
1535 004104  
004104  
004104 000000  
004106 000000  
004110 000000  
004112 000000  
1536  
1537

.BYTE 24.  
.BYTE 26.  
.EVEN  
ERRTBL

;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.

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

L\$ERRTBL::

.EVEN

1539  
1540  
1541  
1542  
1543  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556  
1557  
1558  
1559  
1560  
1561  
1562  
1563  
1564  
1565  
1566  
1567  
1568  
1569  
1570  
1571  
1572  
1573  
1574  
1575

```
.SBTTL GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.
*****
;*
;*   THERE ARE 4 ROUTINES AND MACRO DEFINITIONS USED FOR THE HANDLING OF
;*   GPR VALUES DURING SUBROUTINE CALLS WITHIN THIS PROGRAM.  THE FOUR
;*   ROUTINES/MACRO CALLS HAVE THE FOLLOWING NAMES:
;*
;*   SAVE - MACRO DEFINITION USED AT THE BEGINNING OF A SUBROUTINE TO
;*           SAVE THE GPR CONTENTS FOR LATER RESTORATION.
;*   PASS - MACRO DEFINITION USED AT THE END OF A SUBROUTINE TO RESTORE
;*           THE PREVIOUSLY SAVED GPR CONTENTS AND TO LEAVE THE CONTENTS
;*           OF THE SPECIFIED GPR(S) INTACT (NOT RESTORED).
;*   PREG05 - SUBROUTINE WHICH IS CALLED FROM THE SAVE AND PASS MACRO
;*            EXPANSIONS WHICH ACTUALLY PERFORMS THE ACTIONS ON THE GPRS.
;*
;*   DURING A SUBROUTINE WHICH USES THESE GPR SAVE ROUTINES THE VALUES
;*   OF THE GPRS ARE STORED ON THE STACK IN THE FOLLOWING STACK FRAME:
;*
;*           SP    -> RET PC INTO PREG05 ROUTINE.
;*           SP+2  -> GPR R0 CONTENTS.
;*           SP+4  -> GPR R1 CONTENTS.
;*           SP+6  -> GPR R2 CONTENTS.
;*           SP+8  -> GPR R3 CONTENTS.
;*           SP+10 -> GPR R4 CONTENTS.
;*           SP+12 -> GPR R5 CONTENTS.
;*           SP+14 -> RET PC INTO CALLER OF SUB'TNE WHICH CALLED PREG05.
;*
;*   EACH LEVEL OF SUB'TNE CALLING USES 8 WORDS OF STACK OVERHEAD.
;*   THE SAVE AND PASS MACROS CAN ALSO BE USED IN "STRAIGHT LINE CODE"
;*   TO SAVE AND RESTORE THE GPR VALUES.  IN ANY CASE, AFTER THE
;*   ISSUING OF A PASS CALL THE GPRS WILL BE RESTORED TO THE VALUES
;*   THEY HAD PRIOR TO THE LAST SAVE CALL (EXCEPT FOR THE EXCEPTED,
;*   OR PASSED INTACT, GPRS SPECIFIED AS PARAMETERS TO THE PASS CALL)
;*   AND THE SP WILL ALSO BE RESTORED TO ITS CONDITION BEFORE THE LAST
;*   SAVE CALL.  THE PROGRAMMER MUST BE SURE THAT THE SP HAS THE SAME
;*   VALUE WHEN THE PASS MACRO IS CALLED AS IT HAD IMMEDIATELY AFTER
;*   THE SAVE MACRO WAS CALLED.
*****
```

1577  
1578  
1579  
1580  
1581  
1582  
1583  
1584  
1585  
1586  
1587  
1588  
1589  
1590  
1591

.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.  
;---

000036	LPCSLT==	36	;OFFSET FOR LAST RETURN PC.
000016	PCSLOT==	16	;OFFSET FOR RETURN PC.
000014	R5SLOT==	14	;OFFSET FOR R5.
000012	R4SLOT==	12	;OFFSET FOR R4.
000010	R3SLOT==	10	;OFFSET FOR R3.
000006	R2SLOT==	6	;OFFSET FOR R2.
000004	R1SLOT==	4	;OFFSET FOR R1.
000002	ROSLOT==	2	;OFFSET FOR R0.

1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616

```

.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

```

1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660  
1661  
1662  
1663  
1664  
1665

```

.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
.NLIST
.JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
.ENDM PASS

```



```

1667 .SBTTL GLOBAL SUBROUTINE - PREG05 -
1668 ;*****
1669 ;* PRESERVE REGISTERS R0 THROUGH R5 FOR SUBROUTINE CALLS.
1670 ;*
1671 ;* INPUTS: THE RETURN ADDRESS BACK INTO THE CALLING ROUTINE MUST BE IN
1672 ;* GPR R5. (I.E.- MACROS USE "JSR R5,PREG05".)
1673 ;*
1674 ;* OUTPUTS: REGISTERS R0 THROUGH R5 ARE SAVED ON THE STACK.
1675 ;*
1676 ;*CALLING SEQUENCE: SAVE ;MACRO EXPANSION CALLS PREG05.
1677 ;* [SUBROUTINE CODE]...
1678 ;* PASS ;MACRO EXPANSION RECALLS PREG05.
1679 ;*
1680 ;*COMMENTS: THIS ROUTINE IS RE-ENTRANT.
1681 ;*
1682 ;* PARAMETERS MAY BE PASSED OUT OF A SUBROUTINE BY MODIFYING THE
1683 ;* REGISTER SAVE AREA ON THE STACK. USE THE PASS GPRN MACRO
1684 ;* TO RETURN GPR VALUES INTACT.
1685 ;* USE THE RNSLOT OFFSETS FROM THE SP TO PASS OTHER PARAMETERS.
1686 ;* [EXAMPLE: MOV VALUE,R0SLOT(SP) ]
1687 ;* MAKE SURE THE SP IS AT ITS ORIGINAL VALUE WHEN YOU DO THIS.
1688 ;*
1689 ;*SUBORDINATE ROUTINES CALLED: NONE.
1690 ;*****
1691
1692 004114 PREG05: ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
1693 004114 010446 MOV R4,-(SP) ;SAVE R4
1694 004116 010346 MOV R3,-(SP) ;SAVE R3
1695 004120 010246 MOV R2,-(SP) ;SAVE R2
1696 004122 010146 MOV R1,-(SP) ;SAVE R1
1697 004124 010046 MOV R0,-(SP) ;SAVE R0
1698 004126 010546 MOV R5,-(SP) ;PUSH RETURN PC ON TOP OF STACK
1699 004130 016605 000014 MOV R5SLOT(SP),R5 ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS
1700
1701 004134 004736 JSR PC,8(SP)+ ;CALL THE SUBROUTINE AT THE RETURN ADDRESS
1702 ;FROM THE PREG05 CALL, PUTTING THE PRESENT
1703 ;PC ON THE STACK AS A RETURN ADDRESS INTO
1704 ;THIS (PREG05) ROUTINE.
1705
1706 ;+++
1707 ;THE FOLLOWING CODE IS EXECUTED WHEN THE CALLING ROUTINE DOES A
1708 ;"RETURN" [JSR PC,8(SP)+] USING THE PC DEPOSITED ON THE STACK ABOVE.
1709 ;---
1710
1711 004136 012605 PREGRT: MOV (SP)+,R5 ;PUT RETURN PC IN R5.
1712 004140 012600 MOV (SP)+,R0 ;RESTORE R0.
1713 004142 012601 MOV (SP)+,R1 ;RESTORE R1.
1714 004144 012602 MOV (SP)+,R2 ;RESTORE R2.
1715 004146 012603 MOV (SP)+,R3 ;RESTORE R3.
1716 004150 012604 MOV (SP)+,R4 ;RESTORE R4.
1717
1718 004152 000205 RTS R5 ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05.
1719 ;RESTORING R5 IN THE PROCESS.

```

1721  
1723  
1724  
1725  
1726  
1727  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
  
1741  
1747  
1748  
1749  
1750  
  
RT2/  
  
1751  
1752  
1759

004154  
004154  
004154 104 110 126  
004157 055 061 061  
004162 000  
  
004164  
004164  
004164 104 110 126  
  
004167 055 061 061  
004172 040 106 125  
004175 116 103 040  
004200 124 105 123  
004203 124 040 120  
004206 101 122 124  
004211 062 000

```
.SBTTL GLOBAL TEXT SECTION
;*****
;
;           FVTSKL1.P11
;*****
```

```
;;
; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
; MORE THAN ONE TEST.
;--
```

```
;
; NAMES OF DEVICES SUPPORTED BY PROGRAM
;
;           DEVTYP <DMV-11>
```

```
L$DVTYP::
.ASCIZ /DMV-11/

.EVEN
```

```
; TEST DESCRIPTION
;
;           DESCRIPT <DMV-11 FUNC TEST PART2>
```

```
L$DESC::
.ASCIZ /DMV-11 FUNC TEST PA
```

```
.EVEN
```

```
.EVEN
```

1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1784  
1785

```
*****  
:  
:          FVTA.FMT  
:  
*****  
:  
: FORMAT STATEMENTS USED IN PRINT CALLS  
:
```

1794  
1795  
1796  
1797  
1798  
1799

```
*****  
: FVTB.MSG  
:*****
```

1800

1801  
1802  
1803  
1804

```
.NLIST BIN  
.SBTTL GLOBAL MESSAGE AREA  
***** FORMAT STATEMENTS *****
```

```
1805 004214 MFUNIT:: .ASCIZ /%N%A TESTING UNIT :%D4%N/  
1806 004245 EF0503:: .ASCIZ /%T%N/  
1807 004252 EF0505:: .ASCIZ /%A %D5%A ILLEGAL INTERRUPTS RECEIVED.%N/  
1808 004325 EF1601:: .ASCIZ /%A %T%A ABORTED %N/  
1809 004351 EF3001:: .ASCIZ /%A EXPECTED OR CORRECT VALUE: %03%N/  
1810 004420 EF3002:: .ASCIZ /%A ACTUAL OR MEASURED VALUE: %03%N/  
1811 004467 EF6401:: .ASCIZ /%A %D2%A(D).%N/  
1812 004544 EF7801:: .ASCIZ /%T%A ON LINE %D2%A DECIMAL.%N/  
1813 004602 EF8401:: .ASCIZ /%A %T%A FOR LINE %D2%A(D) AFFECTS OTHER MODEM SIGNALS.%N/  
1814 004674 EF8402:: .ASCII /%A CHANGING %T%A FOR LINE %D2%A(D) AFFECTED /  
1815 004757 .ASCIZ /%T%A FOR LINE %D2%A(D).%N/  
1816 005011 EF9001:: .ASCIZ /%A UNEXPECTED %T%A FOUND IN RECEIVE CHAR FIFO:%N/  
1817 005073 EF9002:: .ASCIZ /%A CODE IS ASSOCIATED WITH LINE: %D2%N/  
1818 005145 EF9003:: .ASCIZ /%A CODE IS: %03%N/  
1819 005174 EF9004:: .ASCIZ /%A %T%A VALUE: %03%N/  
1820 005224 EF9005:: .ASCIZ /%A %T%A VALUE: NONE%N/  
1821 005255 EF9006:: .ASCIZ /%A %T%A %D2%N/  
1822 005274 EF9019:: .ASCIZ /%A %T%A %06%N/  
1823 005313 EF9301:: .ASCIZ /%A %T%D2%A(D), BMP CODE REPORTED :%03%A(0)%N/  
1824 005371 EF9302:: .ASCIZ /%A OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)%N/  
1825 ***** MESSAGE AREA *****
```

```
1826 005471 EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/  
1827 005527 EM0525:: .ASCIZ / RX INTERRUPT(S) RECEIVED WITH RX INTERRUPTS DISABLED./  
1828 005617 EM0526:: .ASCIZ / TX INTERRUPT(S) RECEIVED WITH TX INTERRUPTS DISABLED./  
1829 005707 EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/  
1830 005772 EM4001:: .ASCIZ /DMA_START BIT TEST/  
1831 006015 EM4002:: .ASCIZ /DMA_START BIT BAD ON LINE: /  
1832 006051 EM4101:: .ASCIZ /DMA_ABORT BIT TEST/  
1833 006074 EM4102:: .ASCIZ /DMA_ABORT BIT BAD ON LINE: /  
1834 006130 EM4103:: .ASCIZ /DMA_START BIT FOUND SET AFTER DMA ABORTED ON LINE: /  
1835 006214 EM4901:: .ASCIZ /OAUTO (INACTIVE) BIT TEST/  
1836 006246 EM4902:: .ASCIZ / OAUTO BIT BAD ON LINE: /  
1837 006300 EM5001:: .ASCIZ /OAUTO (ACTIVE) BIT TEST/  
1838 006330 EM5101:: .ASCIZ /IAUTO (INACTIVE) TEST/  
1839 006356 EM5102:: .ASCIZ /IAUTO BIT FOUND SET ON LINE: /  
1840 006414 EM5103:: .ASCIZ /IAUTO BIT BAD ON LINE: /  
1841 006444 EM5201:: .ASCIZ /IAUTO (ACTIVE) TEST/  
1842 006470 EM5202:: .ASCIZ /IAUTO BIT FOUND CLR ON LINE: /  
1843 006526 EM5301:: .ASCIZ /FIFO VALID DATA TEST/  
1844 006553 EM5302:: .ASCIZ /FIFO BAD DATA FIELD CORRUPTED, TEST USED LINE: /  
1845 006632 EM5303:: .ASCIZ /BMP CODE FOUND IN FIFO, TEST INVAILEDATED/  
1846 006703 EM5401:: .ASCIZ \FIFO 3/4 ALARM (INACTIVE) TEST\  
1847 006742 EM5402:: .ASCIZ /FIFO BAD, ALARM SIGNAL DEFECTIVE/  
1848 007003 EM5501:: .ASCIZ \FIFO 3/4 ALARM (ACTIVE) TEST\  
1849 007040 EM5601:: .ASCIZ \FIFO 3/4 ALARM (ACTIVE/INACTIVE) TEST\  
1850 007106 EM5701:: .ASCIZ \FIFO 1/2 LEVEL (ACTIVE/INACTIVE) TEST\  

```

```

1851 007154 EM6401:: .ASCIZ /BREAK GENERATION TEST /
1852 007203 EM6402:: .ASCIZ / BREAK NOT RECEIVED ON LINE(S):/
1853 007244 EM6601:: .ASCIZ /NO OVERRUN ERROR TEST/
1854 007272 EM6602:: .ASCIZ / OVERRUN ERROR REPORTED WHEN NONE FORCED/
1855 007344 EM6701:: .ASCIZ /OVERRUN ERROR TEST/
1856 007367 EM6702:: .ASCIZ / NO OVERRUN ERROR REPORTED, OVERRUN FORCED/
1857 007444 EM7801:: .ASCIZ /MODEM CONTROL DTR BIT TEST/
1858 007477 EM7802:: .ASCIZ / DTR BIT FAULTY ON LINE:/
1859 007530 EM7901:: .ASCIZ /MODEM CONTROL RTS BIT TEST/
1860 007563 EM7902:: .ASCIZ / RTS BIT FAULTY ON LINE:/
1861 007614 EM8001:: .ASCIZ /DSR MODEM STATUS SIGNAL TEST /
1862 007652 EM8002:: .ASCIZ / DSR MODEM STATUS SIGNAL DEFECTIVE/
1863 007716 EM8101:: .ASCIZ /RI MODEM STATUS SIGNAL TEST /
1864 007753 EM8102:: .ASCIZ / RI MODEM STATUS SIGNAL DEFECTIVE/
1865 010016 EM8201:: .ASCIZ /CTS MODEM STATUS SIGNAL TEST /
1866 010054 EM8202:: .ASCIZ / CTS MODEM STATUS SIGNAL DEFECTIVE/
1867 010120 EM8301:: .ASCIZ /DCD MODEM STATUS SIGNAL TEST /
1868 010156 EM8302:: .ASCIZ / DCD MODEM STATUS SIGNAL DEFECTIVE/
1869 010222 EM8401:: .ASCIZ /DTR MODEM CONTROL SIGNAL INTERACTIONS TEST /
1870 010276 EM8402:: .ASCIZ /DTR/
1871 010302 EM8403:: .ASCIZ /DSR/
1872 010306 EM8404:: .ASCIZ /RI/
1873 010311 EM8405:: .ASCIZ /DCD/
1874 010315 EM8406:: .ASCIZ /CTS/
1875 010321 EM8501:: .ASCIZ /RTS MODEM CONTROL SIGNAL INTERACTIONS TEST /
1876 010375 EM8502:: .ASCIZ /RTS/
1877 010401 EM8601:: .ASCIZ /CABLE, PANEL, LOOPBACK CONNECTOR DATA LINE TEST /
1878 010462 EM8602:: .ASCIZ \TX/RX LINE ERROR ON THE FOLLOWING LOOPED BACK TX LINES:\
1879 010552 EM8701:: .ASCIZ /TX DATA LINE INTERACTIONS TEST /
1880 010612 EM8702:: .ASCIZ /DATA LINE INTERACTIONS/
1881 010641 EM8703:: .ASCIZ /TX/
1882 010644 EM9009:: .ASCIZ /EXPECTED OR CORRECT/
1883 010670 EM9010:: .ASCIZ /ACTUAL OR MEASURED /
1884 010714 EM9017:: .ASCII / FIFO WILL NOT PURGE (DATA.VALID STUCK SET)./
1885 010771 .ASCIZ / REMAINDER OF TEST SKIPPED./
1886 011025 EM9026:: .ASCIZ / LPR CONTENTS: /
1887 011051 EM9104:: .ASCIZ / UNEXPECTED DATA FOUND IN FIFO FROM LINE: /
1888 011125 EM9301:: .ASCIZ /BMP CODE REPORT/
1889 011145 EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
1890 011175 EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
1891 011242 EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
1892
1893 .EVEN
1894 .LIST BIN

```

1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911

```
*****  
:  
: FVTSKL2.P11  
:  
*****
```

.SBTTL GLOBAL ERROR REPORT SECTION

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

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

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

```
011316 BGNMSG ER0101
011316 ER0101::
011316 SAVE JSR ;SAVE THE GPR CONTENTS.
004567 172572 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
011322 032705 000001 BIT #BIT0,R5 ;TEST FOR READ ERROR.
011326 001410 BEQ 2# ;SKIP READ ERROR MSG IF NO READ ERROR.
011330 012746 011422 PRINTB #MSG1 ;PRINT READ ERROR MESSAGE.
011334 012746 000001 MOV #MSG1,-(SP)
011340 010600 MOV #1,-(SP)
011342 104414 MOV SP,R0
011344 062706 000004 TRAP C#PNTB
011350 032705 000002 2#: BIT #BIT1,R5 ;TEST FOR WRITE ERROR.
011354 001410 BEQ 4# ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
011356 012746 011500 PRINTB #MSG2 ;PRINT WRITE ERROR MESSAGE.
011362 012746 000001 MOV #MSG2,-(SP)
011366 010600 MOV #1,-(SP)
011370 104414 MOV SP,R0
011372 062706 000004 4#: PRINTX #MSG3 ;SUGGEST THAT DHV MAY BE AT WRONG ADDRESS.
011376 012746 011557 MOV #MSG3,-(SP)
011402 012746 000001 MOV #1,-(SP)
011406 010600 MOV SP,R0
011410 104415 TRAP C#PNTX
011412 062706 000004 PASS ADD #4,SP
011416 004736 JSR ;RESTORE THE GPR CONTENTS.
011420 ENDMSG PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
011420 104423 L10002: TRAP C#MSG
011422 045 101 102 MSG1:: .ASCIZ /#ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.#N/
```

	011425	125	123	040	
	011430	124	111	115	
	011433	105	055	117	
	011436	125	124	040	
	011441	124	122	101	
	011444	120	040	103	
	011447	101	125	123	
	011452	105	104	040	
	011455	102	131	040	
	011460	122	105	101	
	011463	104	040	101	
	011466	124	124	105	
	011471	115	120	124	
	011474	056	045	116	
	011477	000			
1950	011500	045	101	102	MSG2:: .ASCIZ /#ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.#N/
	011503	125	123	040	
	011506	124	111	115	
	011511	105	055	117	
	011514	125	124	040	
	011517	124	122	101	
	011522	120	040	103	
	011525	101	125	123	
	011530	105	104	040	
	011533	102	131	040	
	011536	127	122	111	
	011541	124	105	040	
	011544	101	124	124	
	011547	105	115	120	
	011552	124	056	045	
	011555	116	000		
1951	011557	045	101	104	MSG3:: .ASCIZ /#ADHV MAY BE AT THE WRONG Q-BUS ADDRESS.#N#N/
	011562	110	126	040	
	011565	115	101	131	
	011570	040	102	105	
	011573	040	101	124	
	011576	040	124	110	
	011601	105	040	127	
	011604	122	117	116	
	011607	107	040	121	
	011612	055	102	125	
	011615	123	040	101	
	011620	104	104	122	
	011623	105	123	123	
	011626	056	045	116	
	011631	045	116	000	
1952					
1953					.EVEN



1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966  
1967  
1968  
1969  
1970  
1971  
1972

```

.SBTTL GLOBAL ERROR REPORTING ROUTINE - ER0503 -
;*****
;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
;* MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER.
;*
;* INPUTS: R1 - ADDRESS OF THE MESSAGE TO PRINT.
;*
;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
;*
;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE MESSAGE IN R1.
;* INCLUDE THE LABEL "ER0503" AS THE MESSAGE POINTER
;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
;*
;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
;*
;* SUBORDINATE ROUTINES USED: NONE.
;*****

```

1973 011634  
011634

1974

1975 011634  
011634 010146  
011636 012746 004245  
011642 012746 000002  
011646 010600  
011650 104414  
011652 062706 000006

1976

1977 011656  
011656  
011656 104423

```

BGNMSG ER0503

PRINTB #EF0503,R1 ;PRINT THE MESSAGE.

ENDMSG

```

```

ER0503:
MOV R1,-(SP)
MOV #EF0503,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTB
ADD #6,SP

L10003:
TRAP C#MSG

```

1979  
1980  
1981  
1982  
1983  
1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999

```

.SBTTL GLOBAL ERROR REPORTING ROUTINE - ER1603 -
*****
;* THIS ERROR REPORTING ROUTINE IS USED TO PRINT OUT A BASIC ERROR
;* MESSAGE, ALONG WITH A MESSAGE INFORMING THE OPERATOR WHICH TEST IS
;* ABOUT TO BE ABORTED.
;*
;* INPUTS: R1 - CONTAINS THE ADDRESS OF THE MESSAGE TO BE PRINTED.
;* ERRMSG - CONTAINS THE ADDRESS OF THE MESSAGE THAT INDICATES
;* THE TEST THAT IS BEING PERFORMED, EG DMA, BREAK ETC.
;*
;* OUTPUTS: MESSAGES ARE PRINTED AT THE OPERATORS CONSOLE.
;* "TESTNAME TEST ABORTED"
;*
;* CALLING SEQUENCE: INCLUDE THE LABEL "ER1603" AS THE MESSAGE POINTER
;* PARAMETER IN THE DRS ERROR REPORT MACRO CALL.
;*
;* COMMENTS:
;*
;* SUBORDINATE ROUTINES CALLED: NONE.
*****

```

2000 011660  
011660  
2001 011660  
011660 004567 172230  
2002  
2003 011664  
011664 010146  
011666 012746 004245  
011672 012746 000002  
011676 010600  
011700 104414  
011702 062706 000006  
2004  
2005 011706 016702 172176  
2006 011712  
011712 010246  
011714 012746 004325  
011720 012746 000002  
011724 010600  
011726 104414  
011730 062706 000006  
2007  
2008 011734  
011734 004736  
2009 011736  
011736  
011736 104423

```

BGNMSG ER1603
ER1603::
SAVE ;SAVE THE CONTENTS OF THE GPRS.
JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
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
PASS ;RESTORE THE CONTENTS OF THE GPRS.
JSR PC,#(SP)+ ;RETURN TO PREG05 SUBRT.
ENDMSG
L10004:
TRAP C#MSG

```

2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036

```

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

```

2037  
2038  
2039  
2040  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051

```

011740 BGNMSG ER6401
011740 ER6401::
011740 SAVE R5,PREG05 ;SAVE THE CONTENTS OF THE GPRS.
004567 172150 JSR ;CALL REGISTER SAVE SUBRT.
011744 005002 CLR R2 ;CLEAR LINE NUMBER TO ZERO.
011746 012703 000010 MOV #NUMLNS,R3 ;SET UP MAX LINE COUNT.
011752 010146 PRINTB #EF0503,R1 ;PRINT MESSAGE.
011754 012746 004245 MOV R1,-(SP)
011760 012746 000002 MOV #EF0503,-(SP)
011764 010600 MOV #2,-(SP)
011766 104414 MOV SP,R0
011770 062706 000006 TRAP C#PNTB
011774 000241 ADD #6,SP
011776 006205 2#: CLC ;CLEAR CARRY.
012000 103011 ASR R5 ;SHIFT FLAG OUT INTO CARRY BIT.
012002 010246 BCC 4# ;SKIP ERROR REPORT IF CLEAR.
012004 012746 004467 PRINTB #EF6401,R2 ;PRINT MESSAGE.
012010 012746 000002 MOV R2,-(SP)
012014 010600 MOV #EF6401,-(SP)
012016 104414 MOV #2,-(SP)
012020 062706 000006 MOV SP,R0
012024 005202 4#: TRAP C#PNTB
012026 020302 INC R2 ;INCREMENT LINE COUNT.
012030 001362 CMP R3,R2 ;CHECK IF MAX LINE COUNT EXCEEDED.
012032 004736 BNE 2# ;LOOP IF NOT DONE.
012034 012034 PASS 60#: ;RESTORE THE SAVED CONTENTS OF THE GPRS.
ENDMSG JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
L10005:

```

CVDHBCO DHV-11 FUNC TST PART2  
GLOBAL ERROR REPORTING ROUTINE

MACRO M1200 02-DEC-83 15:07 PAGE 41-1  
- ER6401 -

L4

SEQ 50

012034 104423

TRAP C\$MSG

```

2053 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER7801 -
2054 ;*****
2055 ;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR
2056 ;* MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER. A LINE NUMBER
2057 ;* IS INCLUDED AT THE END OF THE MESSAGE.
2058 ;*
2059 ;* INPUTS: R1 - ADDRESS OF THE MESSAGE TO PRINT.
2060 ;* R3 - NUMBER OF LINE ON WHICH ERROR OCCURRED.
2061 ;*
2062 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
2063 ;*
2064 ;* CALLING SEQUENCE: LOAD THE ADDRESS OF THE MESSAGE IN R1.
2065 ;* LOAD THE LINE NUMBER INTO R3.
2066 ;* INCLUDE THE LABEL "ER7801" AS THE MESSAGE POINTER
2067 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2068 ;*
2069 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
2070 ;*
2071 ;* SUBORDINATE ROUTINES USED: NONE.
2072 ;*****
2073
2074 012036 BGNMSG ER7801
2075 012036 ER7801::
2076 012036 PRINTB #EF7801,R1,R3 ;PRINT THE MESSAGE.
2077 012036 010346 MOV R3,-(SP)
2078 012040 010146 MOV R1,-(SP)
012042 012746 004544 MOV #EF7801,-(SP)
012046 012746 000003 MOV #3,-(SP)
012052 010600 MOV SP,R0
012054 104414 TRAP C#PNTB
012056 062706 000010 ADD #10,SP
2077 012062 ENDMSG
2078 012062 L10006: TRAP C#MSG
012062 104423

```

```

2080 .SBTTL GLOBAL ERROR REPORTING ROUTINE - ER8401 -
2081 ;*****
2082 ;* THIS ERROR REPORTING SUBROUTINE IS INTENDED TO REPORT INTERACTIONS
2083 ;* WHICH HAVE BEEN FOUND BETWEEN A MODEM SIGNAL AND OTHER MODEM SIGNALS.
2084 ;* IT ANALYZES THE MODEM STATUS WHICH IS STORED IN THE STAT STORAGE AREA
2085 ;* AND REPORTS ANY DISCREPANCIES WHICH ARE FOUND BETWEEN THIS STORED DATA
2086 ;* AND THE PRESENT STATE OF THE STAT REGISTERS. SPECIFIED BITS ON THE
2087 ;* LINE ASSOCIATED WITH THE SPECIFIED LINE ARE IGNORED.
2088 ;*
2089 ;* INPUTS: R1 - ADDRESS OF SIGNAL NAME MESSAGE.
2090 ;* R2 - BIT MAP OF BITS TO IGNORE ON SPECIFIED LINE.
2091 ;* R3 - NUMBER OF SPECIFIED LINE.
2092 ;* CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
2093 ;* NUMLNS - EQUATED TO THE NUMBER OF LINES ON THE DUT.
2094 ;* STATA - CONTAINS THE ADDRESS OF THE DUT STAT REGISTER.
2095 ;* STSTB - LABEL AT BASE OF STAT STORAGE TABLE.
2096 ;* TXRLNB - LABEL AT BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
2097 ;*
2098 ;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
2099 ;*
2100 ;* CALLING SEQUENCE: INCLUDE THE LABEL "ER8401" AS THE MESSAGE POINTER
2101 ;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
2102 ;*
2103 ;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
2104 ;*
2105 ;* SUBORDINATE ROUTINES USED: NONE.
2106 ;*****
2107
2108 012064 BGNMSG ER8401
2109 012064 ER8401::
012064 004567 172024 SAVE ;PRESERVE THE CONTENTS OF THE GPRS.
;CALL REGISTER SAVE SUBRT.
2110
2111 012070 PRINTB #EF8401,R1,R3 ;PRINT THE BASIC MESSAGE.
012070 010346 MOV R3,-(SP)
012072 010146 MOV R1,-(SP)
012074 012746 004602 MOV #EF8401,-(SP)
012100 012746 000003 MOV #3,-(SP)
012104 010600 MOV SP,R0
012106 104414 TRAP C#PNTB
012110 062706 000010 ADD #10,SP
2112
2113 012114 010167 000204 MOV R1,4# ;SAVE THE ADDRESS OF THE SIGNAL NAME MESSAGE.
2114 012120 005001 CLR R1 ;CLEAR THE LINE COUNTER.
2115 012122 012704 002644 MOV #STSTB,R4 ;SET UP STAT STORAGE POINTER TO BASE OF TABLE.
2116 012126 010177 170110 2# : MOV R1,#CSRA ;SET UP THE CSR IND.ADR.REG FIELD.
2117 012132 017700 170112 MOV #STATA,R0 ;GET THE CONTENTS OF THIS LINE'S STAT REGISTER.
2118 012136 011405 MOV (R4),R5 ;GET THE PREVIOUS CONTENTS FROM STORAGE.
2119 012140 040005 BIC R0,R5
2120 012142 042400 BIC (R4)+,R0
2121 012144 050005 BIS R0,R5 ;XOR PRESENT AND STORED STAT VALUES.
2122 012146 012700 043777 MOV #43777,R0 ;PREPARE TO MASK OUT UNUSED BITS.
2123 012152 120163 004044 CMPB R1, TXRLNB(R3) ;IS THIS LINE ASSOCIATED WITH SPECIFIED LINE?
2124 012156 001002 BNE 4# ;DON'T MASK OUT SPECIFIED BITS IF IT IS NOT.
2125 012160 056600 000006 BIS R2SLOT(SP),R0 ;MASK OUT SPECIFIED BITS.
2126 012164 040005 4# : BIC R0,R5 ;GET BIT MAP OF UNDESIRED CHANGES.
2127 012166 032705 100000 BIT #BIT15,R5 ;CHECK FOR DSR SIGNAL INTERACTION.

```

```

2128 012172 001404          BEQ      6#          ;SKIP PRINTING LINE IF NO DSR INTERACTION.
2129 012174 012702 010302    MOV      #EM8403,R2    ;SELECT DSR ERROR MESSAGE.
2130 012200 004767 000064    JSR      PC,40#       ;PRINT THE LINE OF THE ERROR MESSAGE.
2131 012204 032705 020000    6#:     BIT      #BIT13,R5 ;CHECK FOR RI SIGNAL INTERACTION.
2132 012210 001404          BEQ      8#          ;SKIP PRINTING LINE IF NO RI INTERACTION.
2133 012212 012702 010306    MOV      #EM8404,R2    ;SELECT RI ERROR MESSAGE.
2134 012216 004767 000046    JSR      PC,40#       ;PRINT THE LINE OF THE ERROR MESSAGE.
2135 012222 032705 010000    8#:     BIT      #BIT12,R5 ;CHECK FOR DCD SIGNAL INTERACTION.
2136 012226 001404          BEQ     10#         ;SKIP PRINTING LINE IF NO DCD INTERACTION.
2137 012230 012702 010311    MOV      #EM8405,R2    ;SELECT DCD ERROR MESSAGE.
2138 012234 004767 000030    JSR      PC,40#       ;PRINT THE LINE OF THE ERROR MESSAGE.
2139 012240 032705 004000    10#:    BIT      #BIT11,R5 ;CHECK FOR CTS SIGNAL INTERACTION.
2140 012244 001404          BEQ     12#         ;SKIP PRINTING LINE IF NO CTS INTERACTION.
2141 012246 012702 010315    MOV      #EM8406,R2    ;SELECT CTS ERROR MESSAGE.
2142 012252 004767 000012    JSR      PC,40#       ;PRINT THE LINE OF THE ERROR MESSAGE.
2143
2144 012256 005201          12#:    INC      R1        ;SELECT NEXT LINE.
2145 012260 020127 000010    CMP      R1,#NUMLNS   ;ALL LINES DONE?
2146 012264 002720          BLT      2#          ;LOOP IF NOT ALL LINES DONE.
2147 012266 000417          BR       60#        ;EXIT THIS ROUTINE.
2148
2149          ; LOCAL ERROR MESSAGE LINE PRINTING ROUTINE.
2150          ;
2151 012270          40#:    PRINTX #EF8402,44#,R3,R2,R1
2151 012270 010146          MOV      R1,-(SP)
2151 012272 010246          MOV      R2,-(SP)
2151 012274 010346          MOV      R3,-(SP)
2151 012276 016746 000022    MOV      44#,-(SP)
2151 012302 012746 004674    MOV      #EF8402,-(SP)
2151 012306 012746 000005    MOV      #5,-(SP)
2151 012312 010600          MOV      SP,R0
2151 012314 104415          TRAP    C#PNTX
2151 012316 062706 000014    ADD     #14,SP
2152 012322 000207          ;
2153 012324 000000          44#:    .WORD   PC
2154 012326 004736          60#:    PASS
2155 012330          JSR      PC,@(SP)
2155 012330          ;LOCAL STORAGE FOR ADDRESS OF SIGNAL NAME.
2155 012330          ;RESTORE ALL THE GPRS TO THE PRESERVED VALUES.
2155 012330          ;RETURN TO PREG05 SUBRT.
2155 012330 104423          L10007: TRAP    C#MSG

```

2157  
2158  
2159  
2160  
2161  
2162  
2163  
2164  
2165  
2166  
2167  
2168  
2169  
2170  
2171  
2172  
2173  
2174  
2175  
2176  
2177  
2178  
2179  
2180  
2181  
2182  
2183

012332  
012332  
012332 010146  
012334 012746 005011  
012340 012746 000002  
012344 010600  
012346 104414  
012350 062706 000006  
012354 010446  
012356 012746 005073  
012362 012746 000002  
012366 010600  
012370 104415  
012372 062706 000006  
012376 010246  
012400 012746 005145  
012404 012746 000002  
012410 010600  
012412 104415  
012414 062706 000006  
012420  
012420  
012420 104423

```
.SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9001 -
;*****
;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH REPORTS AN UNEXPECTED
;* CODE WHICH HAS BEEN FOUND IN THE DUT CSR. THIS CODE CAN BE A BMP
;* CODE, A SELF-TEST CODE, OR A MODEM STATUS CODE.
;*
;* INPUTS: R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
;* R2 - SINGLE BYTE CODE WHICH HAS BEEN READ FROM THE DUT.
;* R4 - LINE NUMBER ASSOCIATED WITH THE CODE.
;*
;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
;*
;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9001" AS THE MESSAGE POINTER
;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
;*
;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
;*
;* SUBORDINATE ROUTINES USED: NONE.
;*****
```

BGNMSG ER9001

ER9001::

PRINTB #EF9001,R1 ;REPORT TYPE OF CODE FOUND.

```
MOV R1,-(SP)
MOV #EF9001,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTB
ADD #6,SP
```

PRINTX #EF9002,R4 ;REPORT THE LINE NUMBER OF THE CODE.

```
MOV R4,-(SP)
MOV #EF9002,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #6,SP
```

PRINTX #EF9003,R2 ;REPORT THE CODE WHICH WAS FOUND.

```
MOV R2,-(SP)
MOV #EF9003,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTX
ADD #6,SP
```

ENDMSG

L10010: TRAP C#MSG



2185  
2186  
2187  
2188  
2189  
2190  
2191  
2192  
2193  
2194  
2195  
2196  
2197  
2198  
2199  
2200  
2201  
2202  
2203  
2204  
2205  
2206  
2207 012422  
012422  
2208  
2209 012422 006203  
2210 012424 042702 177400  
2211 012430  
012430 010346  
012432 010146  
012434 012746 005255  
012440 012746 000003  
012444 010600  
012446 104414  
012450 062706 000010  
2212 012454  
012454 010246  
012456 012746 010670  
012462 012746 005174  
012466 012746 000003  
012472 010600  
012474 104415  
012476 062706 000010  
2213 012502 005704  
2214 012504 100414  
2215 012506  
012506 010446  
012510 012746 010644  
012514 012746 005174  
012520 012746 000003  
012524 010600  
012526 104415  
012530 062706 000010  
2216 012534 000412  
2217 012536  
012536 012746 010644  
012542 012746 005224

```
.SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9002 -
;*****
;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS INTENDED FOR USE IN THE
;* TRANSMISSION AND RECEPTION TESTS. IT REPORTS THE TYPE OF ERROR WHICH
;* HAS OCCURRED WHEN INCORRECT DATA IS RECEIVED FROM THE DUT. THIS
;* ROUTINE ALSO REPORTS THE READ AND EXPECTED DATA VALUES.
;*
;* INPUTS: R1 - ADDRESS OF MESSAGE TO PRINT FIRST.
;* R2 - DATA BYTE READ FROM THE DUT.
;* R3 - LINE NUMBER MULTIPLIED BY 2.
;* R4 - EXPECTED DATA BYTE, BIT 15 SET IF "NONE".
;*
;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
;*
;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9002" AS THE MESSAGE POINTER
;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
;*
;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC AND EXTENDED ERROR INFORMATION.
;*
;* SUBORDINATE ROUTINES USED: PRTLPR.
;*****
```

BGNMSG ER9002

ER9002::

```
ASR R3 ;CALCULATE THE LINE NUMBER.
BIC #177400,R2 ;MASK OUT ALL BUT DATA IN READ CHAR.
PRINTB #EF9006,R1,R3 ;PRINT THE FIRST LINE OF THE MESSAGE.
MOV R3,-(SP)
MOV R1,-(SP)
MOV #EF9006,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C:PNTB
ADD #10,SP
PRINTX #EF9004,#EM9010,R2 ;PRINT ACTUAL DATA.
MOV R2,-(SP)
MOV #EM9010,-(SP)
MOV #EF9004,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C:PNTX
ADD #10,SP
TST R4 ;CHECK FOR "NONE" CODE SET IN EXPECTED DATA.
BMI 21 ;BRANCH TO PRINT "NONE" MESSAGE IF FLAG SET.
PRINTX #EF9004,#EM9009,R4 ;PRINT EXPECTED DATA.
MOV R4,-(SP)
MOV #EM9009,-(SP)
MOV #EF9004,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C:PNTX
ADD #10,SP
BR 601 ;EXIT THIS ROUTINE.
21: PRINTX #EF9005,#EM9009 ;PRINT MESSAGE INDICATING NO EXPECTED DATA.
MOV #EM9009,-(SP)
MOV #EF9005,-(SP)
```

	012546	012746	000002						MOV	#2,-(SP)
	012552	010600							MOV	SP,R0
	012554	104415							TRAP	C#PNTX
	012556	062706	000006						ADD	#6,SP
2218	012562	004767	002162	604:	JSR	PC,PRTLPR				
2219	012566				ENDMSG					
	012566	104423							L10011:	
									TRAP	C#MSG

;PRINT CONTENTS OF THE LPR REGISTER.

2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240 012570  
012570  
2241  
2242 012570  
012570 010146  
012572 010246  
012574 012746 005255  
012600 012746 000003  
012604 010600  
012606 104414  
012610 062706 000010  
2243  
2244 012614  
012614  
012614 104423

```
.SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9101 -
;*****
;* THIS IS A GENERAL ERROR REPORTING SUBROUTINE WHICH REPORTS A MESSAGE
;* WHICH TAKES A SINGLE, 2 DIGIT DECIMAL ARGUMENT AFTER THE END OF AN
;* ASCII MESSAGE.
;*
;* INPUTS: R1 - VALUE TO BE PRINTED AFTER MSG AS 2 DECIMAL DIGITS.
;* R2 - ADDRESS OF MESSAGE TO PRINT FIRST.
;*
;* OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
;*
;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9101" AS THE MESSAGE POINTER
;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
;*
;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
;*
;* SUBORDINATE ROUTINES USED: NONE.
;*****
BGNMSG ER9101
ER9101::
PRINTB #EF9006,R2,R1 ;REPORT THE STRING FOLLOWED BY THE NUMBER.
MOV R1,-(SP)
MOV R2,-(SP)
MOV #EF9006,-(SP)
MOV #3,-(SP)
MOV SP,R0
TRAP C#PNTB
ADD #10,SP
ENDMSG
L10012: TRAP C#MSG
```

2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254  
2255  
2256  
2257  
2258  
2259  
2260  
2261  
2262  
2263  
2264  
2265  
2266 012616  
012616  
2267 012616  
012616 004567 171272  
2268  
2269 012622  
012622 010146  
012624 012746 004245  
012630 012746 000002  
012634 010600  
012636 104414  
012640 062706 000006  
2270 012644 012703 002444  
2271 012650 012705 011145  
2272 012654 012301  
2273 012656 012304  
2274 012660 004767 000056  
2275 012664 020302  
2276 012666 103772  
2277  
2278  
2279  
2280  
2281  
2282  
2283 012670 020227 002640  
2284 012674 001036  
2285 012676 005762 000002  
2286 012702 001433  
2287 012704 012301  
2288 012706 011304  
2289 012710 012705 011175  
2290 012714  
012714 012746 005371  
012720 012746 000001  
012724 010600  
012726 104415

```

.SBTTL GLOBAL ERROR REPORTING ROUTINE - ER9301 -
;*****
;* THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ANY BMP CODES
;* THAT ARE FOUND IN THE BMP CODE QUEUE, TOGETHER WITH THE THE NUMBER OF
;* THE TEST THAT WAS EXECUTING AT THE TIME THE BMP CODE WAS LOGGED.
;*
;* INPUTS: R1 - THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
;* R2 - THE ADDRESS OF THE NEXT EMPTY CELL IN THE QUEUE.
;*
;* OUTPUTS: THE TEST NUMBER FOLLOWED BY THE BMP CODE ARE PRINTED AT THE
;* OPERATOR CONSOLE.
;*
;* CALLING SEQUENCE: INCLUDE THE LABEL "ER9301" AS THE MESSAGE POINTER
;* PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
;*
;* COMMENTS: THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
;*
;* SUBORDINATE ROUTINES USED: NONE.
;*****
BGNMSG ER9301
ER9301::
SAVE ;SAVE THE GPRS ON THE STACK.
;CALL REGISTER SAVE SUBRT.
JSR R5,PREG05
PRINTB #EF0503,R1 ;REPORT UNEXPECTED BMP CODES FOUND.
MOV R1,-(SP)
MOV #EF0503,-(SP)
MOV #2,-(SP)
MOV SP,R0
TRAP C#PNTB
ADD #6,SP
MOV #BMPQ08,R3 ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
MOV #EM9302,R5 ;GET THE MESSAGE TO BE REPORTED.
24: MOV (R3)+,R1 ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
MOV (R3)+,R4 ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
JSR PC,504 ;GO REPORT THE BMP CODE.
CMP R3,R2 ;CHECK IF ALL CODES HAVE BEEN REPORTED.
BLO 24 ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
;
; CHECK IF OVERFLOW HAS OCCURRED.
; THE CONDITIONS FOR OVERFLOW ARE: THE POINTER CONTAINS THE ADDRESS OF THE
; LAST CELL IN THE QUEUE, AND A BMP CODE HAS ALREADY BEEN WRITTEN INTO THAT
; CELL.
;-
CMP R2,#BMPQ0E-4 ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
BNE 604 ;EXIT IF NOT AT THE LAST LOCATION.
TST 2(R2) ;CHECK FOR A BMP CODE IN THE LAST CELL
BEQ 604 ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
MOV (R3)+,R1 ;GET THE TEST NUMBER OFF THE QUEUE.
MOV (R3),R4 ;GET THE BMP CODE OFF THE QUEUE.
MOV #EM9303,R5 ;SELECT THE MESSAGE TO BE REPORTED.
PRINTX #EF9302 ;REPORT OVERFLOW CONDITION.
MOV #EF9302,-(SP)
MOV #1,-(SP)
MOV SP,R0
TRAP C#PNTX

```

```

012730 062706 000004
2291 012734 004767 000002      JSR    PC,50$      ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
2292 012740 000414              BR     60$        ;EXIT.
2293
2294 012742      50$: PRINTX  #EF9301,R5,R1,R4 ;PRINT THE MESSAGE.
      012742      010446
      012744      010146
      012746      010546
      012750      012746 005313
      012754      012746 000004
      012760      010600
      012762      104415
      012764      062706 000012
2295 012770      000207
2296 012772      004736      60$: RTS    PC      ;RETURN.
      012772
      ENDMSG              JSR    PC,#(SP)+ ;RESTORE THE GPR CONTENTS.
      ;RETURN TO PREG05 SUBRT.
      L10013: TRAP  C#MSG

```

```

      ADD    #4,SP
      MOV    R4,-(SP)
      MOV    R1,-(SP)
      MOV    R5,-(SP)
      MOV    #EF9301,-(SP)
      MOV    #4,-(SP)
      MOV    SP,R0
      TRAP  C#PNTX
      ADD    #12,SP

```

2300  
2302  
2303  
2304  
2305  
2306  
2308  
2309  
2310  
2311  
2312  
2313

```
.SBTTL GLOBAL SUBROUTINES SECTION
:*****
:
:           FVTSKL3.P11
:*****
:
:++
: THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
: THAT ARE USED IN MORE THAN ONE TEST.
:--
```

```

2315 .SBTTL GLOBAL SUBROUTINE - ALTFLD -
2316 ;* *****
2317 ;* - ALTER DEVICE REGISTER FIELDS ROUTINE -
2318 ;* THIS SUBROUTINE ALTERS THE SPECIFIED FIELD OF THE SPECIFIED DEVICE
2319 ;* REGISTER FOR THE SPECIFIED LINES. THIS ROUTINE CAN BE USED TO SET
2320 ;* OR CLEAR BITS WITHIN SELECTED FIELDS OF SELECTED REGISTERS.
2321 ;* USE EXAMPLES: SET RX.BAUD.RATE FIELDS ON LINES 3 AND 6.
2322 ;* CLEAR TX.DMA BITS ON ALL LINES.
2323 ;*
2324 ;* INPUTS: R1 - ADDRESS OF THE REGISTERS TO ALTER.
2325 ;* R2 - BIT FIELDS SET TO DESIRED STATES.
2326 ;* R3 - BIT MAP OF LINES FOR WHICH TO ALTER REGISTER.
2327 ;* R4 - MASK OF BITS TO ALTER (1 INDICATES CHANGE BIT).
2328 ;* CSRA - CONTAINS THE ADDRESS OF THE DEVICE CSR.
2329 ;* IESTAT - SAVED STATES OF THE INTERRUPT ENABLE BITS.
2330 ;*
2331 ;* OUTPUTS: DEVICE REGISTERS - SPECIFIED REGISTER FIELDS ALTERED.
2332 ;* CSR IND.ADR.REG FIELD - DESTROYED.
2333 ;*
2334 ;* CALLING SEQUENCE: JSR PC,ALTFLD
2335 ;*
2336 ;* COMMENTS: THIS ROUTINE READS THE SPECIFIED REGISTERS FOR ALL LINES
2337 ;* WITH NUMBERS LOWER THAN THE HIGHEST SPECIFIED LINE.
2338 ;* THIS ROUTINE DOES NOT READ THE CSR.
2339 ;*
2340 ;* SUBROUTINES CALLED: NONE.
2341 ;*
2342 ;*
2343 012776 004567 171112 ALTFLD:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
; R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2344
2345 ;*
2346 ;* SET UP TO LOOP FOR EACH LINE:
2347 ;* PREPARE THE WORD TO BE ORED INTO THE REGISTER CONTENTS.
2348 ;* SET UP THE WORD TO WRITE INTO THE IND.ADR.REG FIELD OF THE CSR.
2349 ;*
2350 013002 010400 ;MOV R4,R0 ;CALCULATE THE NEW CONTENTS OF THE
2351 013004 005100 ;COM R0 ; REGISTER FIELDS WHICH ARE TO BE
2352 013006 040002 ;BIC R0,R2 ; ALTERED BY THIS ROUTINE.
2353 013010 016705 167254 ;MOV IESTAT,R5 ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.
2354 ;*
2355 ;* LOOP ONCE FOR EACH LINE, ALTERING THE SPECIFIED FIELD IN THE SPECIFIED
2356 ;* REGISTER IF THE LINE HAS BEEN SELECTED FOR ALTERING.
2357 ;* EXIT THE LOOP IF NO MORE LINES TO ALTER, OR IF WE HAVE ALTERED THE MAX
2358 ;* ALLOWABLE NUMBER OF LINES (AS SPECIFIED BY NUMLNS).
2359 ;*
2360 013014 000241 ;CLC ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
2361 013016 006003 20: ROR R3 ;GET THE LINE SELECT BIT FOR THIS LINE.
2362 013020 103006 BCC 40 ;SKIP SETUP IF LINE IS NOT SELECTED.
2363 013022 010577 167214 MOV R5,@CSRA ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
2364 013026 011100 MOV (R1),R0 ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
2365 013030 040400 BIC R4,R0 ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
2366 013032 050200 BIS R2,R0 ;OR IN THE NEW STATES OF THE FIELDS.
2367 013034 010011 MOV R0,(R1) ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
2368 013036 005205 40: INC R5 ;SET LINE NUMBER TO THE NEXT LINE.
2369 013040 005703 TST R3 ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
2370 013042 001365 BNE 20 ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.

```

K5

CVDHBCO DHV-11 FUNC TST PART2  
GLOBAL SUBROUTINE

MACRO M1200 02-DEC-83 15:07 PAGE 49-1  
- ALTFLD -

SEQ 62

2371  
2372 013044 604: PASS ;RESTORE GPRS.  
013044 004736 ;PC,0(SP); ;RETURN TO PREG05 SUBRT.  
2373 013046 000207 RTS PC JSR ;RETURN TO CALLING ROUTNE.



```

2375 .SBTTL GLOBAL SUBROUTINE - ASLNTL -
2376 ;** *****
2377 ;* - SETUP ASSOCIATED LINE NUMBER TABLES ROUTINE -
2378 ;* THIS ROUTINE SETS UP THE TWO TABLES WHICH ARE CONTAIN INFORMATION
2379 ;* ABOUT THE TX/RX LINE WHICH IS ASSOCIATED WITH A PARTICULAR RX/TX
2380 ;* LINE. ONE TABLE IS A TABLE OF WORDS WHICH CONTAINS WORD OFFSET
2381 ;* VALUES AND THE OTHER TABLE IS A TABLE OF BYTES WHICH CONTAINS
2382 ;* LINE NUMBER VALUES.
2383 ;*
2384 ;* INPUTS: LOPBCK - STORAGE FOR THE TYPE OF LOOPBACK ON THE DUT.
2385 ;* NUMLNS - EQUATED TO THE NUMBER OF LINES ON THE DUT.
2386 ;* STGTRB - LABEL AT BASE OF STAGGERED LINE ASSOCIATION TBL.
2387 ;* TXRLNB - LABEL AT BASE OF BYTE TX/RX LINE NUMBER TABLE.
2388 ;* TXRXLB - LABEL AT BASE OF WORD TX/RX LINE NUMBER TABLE.
2389 ;* TXRXLE - LABEL AT END OF WORD TX/RX LINE NUMBER TABLE.
2390 ;*
2391 ;* OUTPUTS: TXRXL, TXRLN - TABLES INITIALIZED FOR SELECTED LOOPBACK.
2392 ;*
2393 ;* CALLING SEQUENCE: JSR PC,ASLNTL
2394 ;*
2395 ;* COMMENTS:
2396 ;*
2397 ;* SUBORDINATE ROUTINES CALLED: NONE.
2398 ;-- *****
2399
2400 013050 ASLNTL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
2401 013050 004567 171040 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2402 013054 126727 167156 000002 CMPB LOPBCK,#2 ;TEST FOR STAGGERED LOOPBACK.
2403 013062 001411 BEQ 4# ;GO SET UP STAGGERED TABLE IF STAGGERED LPBCK.
2404 ;+
2405 ; SET UP THE WORD TABLE FOR NON-STAGGERED LOOPBACK.
2406 ;-
2406 013064 005005 CLR R5 ;CLEAR THE LINE COUNTER
2407 013066 010565 004004 MOV R5, TXRXLB(R5) ;SET UP A WORD OF THE TABLE.
2408 013072 005205 INC R5
2409 013074 005205 INC R5 ;SET LINE COUNTER TO NEXT LINE OFFSET.
2410 013076 020527 000020 CMP R5,#2*NUMLNS ;TEST FOR ALL LINES DONE.
2411 013102 002771 BLT 2# ;LOOP UNTIL ALL LINES DONE.
2412 013104 000411 BR 8# ;GO SET UP THE BYTE TABLE.
2413 ;+
2414 ; SET UP THE WORD TABLE FOR STAGGERED LOOPBACK.
2415 ;-
2416 013106 012701 004064 MOV #STGTRB,R1 ;SET UP THE SOURCE POINTER.
2417 013112 012702 004004 MOV #TXRXLB,R2 ;SET UP THE DESTINATION POINTER.
2418 013116 112122 6# MOV (R1)+,(R2)+ ;MOVE A BYTE INTO THE TABLE.
2419 013120 105022 CLR (R2)+ ;CLEAR THE UPPER BYTE OF WORD TABLE ENTRY.
2420 013122 020227 004044 CMP R2,#TXRXLE ;COMPARE POINTER WITH END ADR OF TABLE.
2421 013126 002773 BLT 6# ;LOOP IF NOT AT END YET.
2422 ;+
2423 ; SET UP THE BYTE TABLE BASED ON THE WORD ASSOCIATION TABLE.
2424 ;-
2425 013130 012701 004004 MOV #TXRXLB,R1 ;SET UP THE SOURCE POINTER.
2426 013134 012702 004044 MOV #TXRLNB,R2 ;SET UP THE DESTINATION POINTER.
2427 013140 012103 10# MOV (R1)+,R3 ;GET THE WORD OFFSET VALUE FROM WORD TABLE.
2428 013142 006203 ASR R3 ;DIVIDE BY 2 TO GET LINE NUMBER VALUE.
2429 013144 110322 MOV (R3),R2 ;LOAD THE BYTE LINE NUMBER INTO TABLE.
2430 013146 020127 004044 CMP R1,#TXRXLE ;COMPARE SOURCE POINTER WITH ADR OF TABLE END.

```

2431	013152	002772		BLT	10\$				;LOOP IF NOT AT END OF TABLE YET.
2432									
2433	013154		60\$:	PASS					;RESTORE GPRS.
	013154	004736				: JSR			PC,@(SP)+
2434	013156	000207		RTS	PC				;RETURN TO PREG05 SUBRT.

```

2436 .SBTTL GLOBAL SUBROUTINE - CALMSL -
2437 ;* *****
2438 ;* - CALIBRATE MILLI SECOND LOOP COUNT SUBROUTINE -
2439 ;* THIS SUBROUTINE CALIBRATES THE TIMING LOOP WHICH IS USED IN THE MSLOOP
2440 ;* ROUTINE. THIS SUBROUTINE CALCULATES A VALUE FOR THE MSLCNT VARIABLE
2441 ;* WHICH IS THE NUMBER OF SOFTWARE LOOPS WHICH TAKES 1 MS TO EXECUTE IN
2442 ;* THE MSLOOP ROUTINE. THIS ROUTINE CALIBRATES THE COUNT BY USING THE
2443 ;* LINE TIME CLOCK (LTC), SO IF NO LTC IS AVAILABLE THE DEFAULT VALUE FOR
2444 ;* THE DELAY COUNT MUST BE USED.
2445 ;*
2446 ;*
2447 ;* INPUTS: MSLCNT - DEFAULT 1 MS DELAY LOOP COUNT VALUE, OR
2448 ;* VALUE FROM PREVIOUS CALIBRATION.
2449 ;* MSTICK - NUMBER OF MS PER LTC CLOCK TICK.
2450 ;* TIMER1 - TIMER COUNTER CHANGED BY LTC INTERRUPT SERVICE RTN.
2451 ;* CLKHRZ - NUMBER OF LTC CLICKS PER SECOND (50 OR 60).
2452 ;*
2453 ;* OUTPUTS: CARRY - SET IF LTC IS AVAILABLE, AND NEW CALIBRATION PERFORMED.
2454 ;* MSLCNT - NEW 1 MS DELAY LOOP COUNT VALUE IF LTC AVAILABLE, OR
2455 ;* UNCHANGED IF NO LTC IS AVAILABLE.
2456 ;*
2457 ;* CALLING SEQUENCE: JSR PC,CALMSL
2458 ;*
2459 ;* COMMENTS:
2460 ;*
2461 ;* SUBORDINATE ROUTINES CALLED: UNSDIV,OOPS.
2462 ;*
2463 ;*
2464 013160 CALMSL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
013160 004567 170730 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
013164 005067 000210 CLR 62# ;CLEAR THE 2ND TIME FLAG.
2466 ;*
2467 ;* SYNCHRONIZE WITH THE LTC.
2468 ;*
2469 013170 012705 000001 2#: MOV #1,R5 ;SET OUTER LOOP COUNTER TO 1 LOOP.
2470 ;* INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE ***
2471 ;* FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. ***
2472 013174 005000 CLR R0 ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
2473 013176 012767 000001 167122 MOV #1,TIMER1 ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
2474 013204 005767 167116 4#: TST TIMER1 ;CHECK FOR COUNTER HAVING GONE TO ZERO.
2475 013210 001410 BEQ 6# ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
2476 013212 005200 INC R0 ;COUNT THIS ITERATION OF THE INNER LOOP.
2477 013214 001373 BNE 4# ;LOOP IF COUNTER HAS NOT TURNED OVER.
2478 013216 005305 DEC R5 ;DECREMENT THE INNER LOOP COUNTER.
2479 013220 003371 BGT 4# ;LOOP IF OUTER LOOP COUNT NOT UP.
2480 ;*
2481 ;* IF WE GOT NO LTC INTERRUPT, INDICATE THAT THERE IS NO LTC AVAILABLE.
2482 ;* LTC MUST BE FLAKEY, OR NOT REALLY AN LTC AT ALL.
2483 ;*
2484 013222 005067 167076 CLR CLKHRZ ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
2485 013226 000241 CLC ;INDICATE FAILURE FOR RETURN.
2486 013230 000461 BR 60# ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
2487 ;*
2488 ;* WE ARE NOW SYNCHRONIZED WITH THE LTC.
2489 ;* SET UP FOR THE CALIBRATION LOOP.
2490 ;*
2491 013232 012704 002326 6#: MOV #TIMER1,R4 ;WILL TEST TIMER1 IN THE LOOP BELOW.

```

```

2492 013236 005001          CLR    R1          ;CLEAR THE OUTER LOOP COUNTER.
2493 013240 005002          CLR    R2          ;INDICATE TO CHECK ALL BITS OF TIMER1.
2494 013242 005003          CLR    R3          ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2495 013244 012714 000001    MOV    #1,(R4)     ;LOAD TIMER1 WITH COUNT OF 1.
2496
2497 013250 016705 167064    8#:   MOV    MSLCNT,R5 ;LOAD MS LOOP COUNT.
2498 013254 011400          10#:  MOV    (R4),R0     ;GET THE TIMER1 VALUE.
2499 013256 010067 000120    MOV    R0,64#     ;SAVE WORD (LIKE IN THE REAL LOOP).
2500 013262 040200          BIC    R2,R0      ;LEAVE ALL THE BITS.
2501 013264 020003          CMP    R0,R3      ;COMPARE AGAINST ZERO.
2502 013266 000261          SEC          ;SET CARRY IN CASE OF SUCCESS.
2503 013270 001406          BEQ    12#        ;EXIT LOOP IF TIMER1 HAS CLEARED.
2504 013272 005305          DEC    R5         ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2505 013274 001367          BNE    10#        ;LOOP IF MS NOT UP.
2506 013276 005301          DEC    R1         ;DECREMENT THE MS TIME COUNT.
2507 013300 001363          BNE    8#         ;KEEP LOOPING.
2508 013302 004767 001214    JSR    PC,OOPS    ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
2509
2510          ;*
2511          ; WE HAVE NOW HAVE LOOP COUNT INFORMATION FOR ONE CLOCK TICK.
2512          ; WE HAVE NEGATIVE OF NUMBER OF OUTER LOOPS IN R1, EACH IS MSLCNT INNER LOOPS.
2513          ; WE HAVE THE PORTION OF THE LAST OUTER LOOP NOT EXECUTED, IN R5.
2514          ; NOW WE CALCULATE THE TOTAL NUMBER OF INNER LOOPS EXECUTED.
2515
2515 013306 005401          12#:  NEG    R1         ;GET NUMBER OF OUTER LOOPS.
2516 013310 016702 167024    MOV    MSLCNT,R2  ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2517 013314 010203          MOV    R2,R3      ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2518 013316 160502          SUB    R5,R2      ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2519 013320 010204          MOV    R2,R4      ; AND ADD TO ACCUMULATOR LSWORD.
2520 013322 005005          CLR    R5         ;CLEAR ACCUMULATOR MSWORD.
2521 013324 005301          14#:  DEC    R1         ;CHECK R1 FOR 0 CONDITION
2522 013326 100403          BMI    16#        ; SKIP MULTIPLICATION IF ZERO
2523 013330 060304          ADD    R3,R4      ;MULTIPLY NUMBER OF INNER
2524 013332 005505          ADC    R5         ; LOOPS PER OUTER LOOP BY
2525 013334 000773          BR    14#        ;NUMBER OF OUTER LOOPS PERFORMED.
2526
2527          ;*
2528          ; DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER LTC TICK.
2529
2529 013336 016701 166774    16#:  MOV    MSTICK,R1  ;# OF MS PER LTC TICK IS DIVISOR.
2530 013342 010403          MOV    R4,R3      ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2531 013344 010502          MOV    R5,R2      ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2532 013346 004767 003012    JSR    PC,UNSDIV  ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
2533 013352 103402          BCS    18#        ;BYPASS OOPS IF WE'RE OK.
2534 013354 004767 001142    JSR    PC,OOPS    ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2535 013360 010167 166754    18#:  MOV    R1,MSLCNT ;SET NEW VALUE FOR MS LOOP COUNT.
2536 013364 005167 000010    COM    62#        ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2537 013370 001277          BNE    2#         ;BRANCH IF ONLY ONE ITERATION DONE.
2538 013372 000261          SEC          ;SET THE SUCCESS FLAG FOR EXIT.
2539
2540 013374          60#:  PASS          ;RESTORE GPRS.
2540 013374 004736          RTS    PC        JSR    PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
2541 013376 000207          ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
2542
2543 013400 000000          62#:  .WORD 0      ;2ND CALIBRATION ITERATION FLAGS.
2544 013402 000000          64#:  .WORD 0      ;DUMMY WORD FOR STORAGE OF THE READ WORD.

```

2546  
2547  
2548  
2549  
2550  
2551  
2552  
2553  
2554  
2555  
2556  
2557  
2558  
2559  
2560  
2561  
2562  
2563  
2564  
2565  
2566  
2567  
2568  
2569  
2570  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579

013404  
013404 004567 170504  
013410 012700 170301  
013414 040200  
013416 001011  
013420 004767 002116  
013424 012701 006632  
013430 012767 011660 170454  
013436 000261  
013440 000401  
013442 000241  
013444 010166 000004  
013444 004736  
013450 004736  
013452 000207

```
.SBTTL GLOBAL SUBROUTINE - CHKBMP -
;*****
;* - CHECK IF CHARACTER IS A BMP CODE -
;* THIS SUBROUTINE IS USED TO CHECK FOR BMP CODES.
;* IF A BMP CODE IS DETECTED, IT WILL BE SAVED ON THE QUEUE TO BE REPORTED
;* LATER. THE CARRY IS USED AS A FLAG TO INDICATE A CODE HAS BEEN FOUND.
;*
;* INPUTS: R2 - CONTAINS THE DATA TO BE CHECKED.
;*
;* OUTPUTS: R1 - CONTAINS THE MESSAGE TO BE REPORTED.
;* ERRBLK - CONTAINS THE ERROR REPORTING ROUTINE.
;* CARRY BIT IS USED TO INDICATE A BMP CODE FOUND, CARRY SET.
;*
;* CALLING SEQUENCE: JSR PC,CHKBMP
;*
;* COMMENTS:
;*
;* SUBORDINATE ROUTINES CALLED: SAVBMP.
;*****
CHKBMP:: SAVE
;SAVE CONTENTS OF GPRS R0 THRU R5.
R5,PREG05 ;CALL REGISTER SAVE SUBRT.
;SET UP THE FLAGS OF A BMP CODE.
;TRY TO CLEAR THE BMP CODE FLAGS.
BNE 2; ;IF NOT A BMP CODE, EXIT WITH FAILURE.
;SAVE THE BMP CODE ON THE QUEUE.
JSR PC,SAVBMP ;SAVE THE BMP CODE ON THE QUEUE.
;PASS THE MESSAGE TO BE REPORTED.
MOV #EM5303,R1 ;PASS THE MESSAGE TO BE REPORTED.
;SELECT THE CORRECT ERROR REPORTING ROUTINE.
MOV #ER1603,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
;PASS FLAG TO INDICATE SUCCESS, BMP CODE FOUND.
SEC
;EXIT.
BR 60;
;PASS FLAG TO INDICATE FAILURE.
2;: CLC
60;: PASS R1
;RESTORE GPRS, EXCEPT
MOV R1,R1SLOT(SP) ;PUT R1 IN STACK SLOT.
JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
;R1 - CONTAINS THE ADDRESS OF ERROR MESSAGE.
;CARRY BIT - SET INDICATES SUCCESS.
RTS PC
```

2581  
2582  
2583  
2584  
2585  
2586  
2587  
2588  
2589  
2590  
2591  
2592  
2593  
2594  
2595  
2596  
2597  
2598  
2599  
2600  
2601  
2602  
2603  
2604  
2605  
2606  
2607  
2608  
2609  
2610  
2611  
2612

013454  
013454 004567 170434  
013460 005067 166626  
013464 011011  
013466 005767 166620  
013472 000261  
013474 001401  
013476 000241  
013500  
013500 004736  
013502 000207

```

.SBTTL GLOBAL SUBROUTINE - CKTRAP -
;*****
;* CHECK TRAP ROUTINE -
;* THIS SUBROUTINE IS USED TO CHECK FOR A BUS TIME-OUT TRAP (004 TRAP)
;* WHICH IS CAUSED BY AN ACCESS TO A NON-EXISTENT MEMORY OR I/O LOCATION.
;* IF THE TRAP DOES NOT OCCUR, THIS ROUTINE RETURNS A SUCCESS INDICATION.
;*
;* INPUTS: R0 - SOURCE ADDRESS FOR MOVE.
;* R1 - DESTINATION ADDRESS FOR MOVE.
;* (R0) - SOURCE FOR THE MOVE.
;*
;* OUTPUTS: (R1) - WRITTEN TO THE CONTENTS OF (R0).
;* CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED.
;* TP4FLG - NONZERO IF TRAP OCCURRED, CLEARED OTHERWISE.
;*
;* CALLING SEQUENCE: JSR PC,CKTRAP
;*
;* COMMENTS: IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS WHICH
;* IS LABELED ADRPTR WILL BE THE TRAP PC ADDRESS ON THE STACK.
;*
;* SUBORDINATE ROUTINES CALLED: NONE.
;*****
CKTRAP:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
CLR TP4FLG ;CLEAR THE 004 TRAP FLAGS.
MOV (R0),(R1) ;PERFORM THE MOVE IN QUESTION.
ADRPTR:: TST TP4FLG ;CHECK FOR OCCURENCE OF TRAP.
SEC ;INDICATE SUCCESS.
BEQ 606 ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
CLC ;INDICATE FAILURE.
606: PASS ;RESTORE GPRS.
;RTS PC,0(SP) ;RETURN TO PREG05 SUBRT.
RTS PC JSR

```

2614  
2615  
2616  
2617  
2618  
2619  
2620  
2621  
2622  
2623  
2624  
2625  
2626  
2627  
2628  
2629  
2630  
2631  
2632  
2633  
2634  
2635  
2636  
2637  
2638  
2639  
2640  
2641  
2642 013504  
013504 004567 170404  
2643  
2644  
2645  
2646  
2647 013510 004767 001654  
2648 013514 103002  
2649  
2650  
2651  
2652 013516 004767 001310  
2653  
2654 013522  
2655 013522 004736  
2656  
2657 013524 000207

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

2659  
2660  
2661  
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669  
2670  
2671  
2672  
2673  
2674  
2675  
2676  
2677  
2678  
2679  
2680  
2681

013526  
013526 004567 170362  
013532 012701 000020  
013536 005020  
013540 005301  
013542 001375  
013544  
013544 004736  
013546 000207

```

.SBTTL GLOBAL SUBROUTINE - CLR16W -
; * *****
; * - CLEAR SIXTEEN WORDS ROUTINE -
; * THIS SUBROUTINE CLEARS 16 WORDS STARTING WITH THE SPECIFIED WORD.
; *
; * INPUTS: R0 - ADDRESS OF THE FIRST WORD TO CLEAR.
; *
; * OUTPUTS: (R0) TO (R0+15) - 16 WORDS OF MEMORY ARE CLEARED TO 0.
; *
; * CALLING SEQUENCE: JSR PC,CLR16W
; *
; * COMMENTS:
; *
; * SUBORDINATE ROUTINES CALLED: NONE.
; - *****

CLR16W:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
R5,PREG05 ;CALL REGISTER SAVE SUBRT.
;SET THE LOOP COUNTER TO 16.
2#: MOV #16.,R1 ;CLEAR A WORD OF MEMORY.
CLR (R0)+ ;COUNT THIS LOOP.
DEC R1 ;LOOP IF NOT 16 WORD CLEARED.
BNE 2# ;RESTORE GPRS.
60#: PASS JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
RTS PC

```



2683  
2684  
2685  
2686  
2687  
2688  
2689  
2690  
2691  
2692  
2693  
2694  
2695  
2696  
2697  
2698  
2699  
2700  
2701  
2702  
2703  
2704  
2705  
2706  
2707 013550  
013550 004567 170340  
2708 013554 005003  
2709 013556 012704 002644  
2710 013562 010377 166454  
2711 013566 017700 166456  
2712 013572 011405  
2713 013574 040005  
2714 013576 042400  
2715 013600 050005  
2716 013602 012700 043777  
2717 013606 120301  
2718 013610 001001  
2719 013612 050200  
2720 013614 040005  
2721 013616 001006  
2722 013620 005203  
2723 013622 020327 000010  
2724 013626 002755  
2725 013630 000261  
2726 013632 000401  
2727  
2728 013634 000241  
2729  
2730 013636  
013636 004736  
2731 013640 000207

```

.SBTTL GLOBAL SUBROUTINE - CMPMST -
;+ *****
;* - COMPARE MODEM STATUS ROUTINE -
;* THIS ROUTINE IS USED TO COMPARE THE PRESENT MODEM STATUS AGAINST THE
;* MODEM STATUS WHICH IS STORED IN THE MODEM STATUS STORAGE TABLE. IT
;* IGNORES THE STATES OF THE SPECIFIED SIGNALS ON A SPECIFIED LINE.
;*
;* INPUTS: R1 - LINE NUMBER OF SPECIFIED LINE.
;* R2 - BIT MAP OF BITS TO IGNORE ON SPECIFIED LINE.
;* CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
;* NUMLNS - EQUATED TO THE NUMBER OF LINES ON THE DUT.
;* STATA - CONTAINS THE ADDRESS OF THE DUT STAT REGISTER.
;* STSTB - LABEL AT BASE OF STAT STORAGE TABLE.
;* TXRLNB - LABEL AT BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
;*
;* OUTPUTS: CARRY - SUCCESS FLAG (SET IF NO DISCREPANCIES WERE FOUND).
;*
;* CALLING SEQUENCE: JSR PC,CMPMST
;*
;* COMMENTS:
;*
;* SUBORDINATE ROUTINES CALLED: NONE.
;-- *****

CMPMST:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
R5,PREG05 ;CALL REGISTER SAVE SUBRT.
CLR R3 ;CLEAR THE LINE COUNTER.
MOV #STSTB,R4 ;SET UP STAT STORAGE POINTER TO BASE OF TABLE.
24: MOV R3,BCSRA ;SET UP THE CSR IND.ADR.REG FIELD.
MOV #STATA,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
BIS R0,R5 ;XOR PRESENT AND STORED STAT VALUES.
MOV #43777,R0 ;PREPARE TO MASK OUT UNUSED BITS.
CMPB R3,R1 ;TEST FOR THIS BEING SPECIFIED LINE.
BNE 104 ;DON'T MASK OUT SPECIFIED BITS IF IT IS NOT.
BIS R2,R0 ;MASK OUT SPECIFIED BITS.
BIC R0,R5 ;GET BIT MAP OF UNDESIRED CHANGES.
BNE 504 ;EXIT WITH FAILURE IF CHANGES OCCURRED.
INC R3 ;SELECT NEXT LINE.
CMP R3,#NUMLNS ;ALL LINES DONE?
BLT 24 ;LOOP IF NOT ALL LINES DONE.
SEC ;INDICATE SUCCESS.
BR 604 ;EXIT THIS ROUTINE WITH SUCCESS.

504: CLC ;INDICATE FAILURE.

604: PASS ;RESTORE GPRS.
RTS PC JSR PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
; CARRY - SUCCESS FLAG (SET IF SUCCESS).

```

2733  
2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742  
2743  
2744  
2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753 013642  
013642 004567 170246  
2754 013646 012702 004004  
2755 013652 010503  
2756 013654 012704 000010  
2757 013660 005005  
2758 013662 006203  
2759 013664 103005  
2760 013666 011201  
2761 013670 006201  
2762 013672 004767 000414  
2763 013676 050005  
2764 013700 005722  
2765 013702 005304  
2766 013704 001366  
2767 013706  
013706 010566 000014  
013712 004736  
2768  
2769 013714 000207

```

.SBTTL GLOBAL SUBROUTINE - CONMAP -
; ** *****
; * - CONVERT LINE BIT MAP.
; * THIS SUBROUTINE IS USED TO CONVERT A BIT MAP PASSED TO IT , INTO
; * ANOTHER LINE BIT MAP THAT IS BASED UPON THE ASSOCIATED TX/RX LINE
; * NUMBER/OFFSET TABLE.
; *
; * INPUTS: R5 - CONTAINS THE LINE BIT MAP TO BE TRANSFORMED.
; * TXRXLB - BASE ADDRESS OF ASSOCIATED TX/RX LINE NUMBER TABLE.
; *
; * OUTPUTS: R5 - CONTAINS AN ASSOCIATED LINE BIT MAP.
; *
; * CALLING SEQUENCE: JSR PC,CONMAP
; *
; * COMMENTS: THE TX/RX ASSOCIATION TABLE MUST BE INITIALISED BEFORE THIS
; * ROUTINE IS CALLED.
; *
; * SUBORDINATE ROUTINES CALLED: NONE.
; -- *****

CONMAP::SAVE
                JSR          ;SAVE CONTENTS OF GPRS R0 THRU R5.
                MOV          R5,PREG05 ;CALL REGISTER SAVE SUBRT.
                MOV          #TXRXLB,R2 ;GET THE BASE ADDRESS OF THE LINE ASSOC TABLE.
                MOV          R5,R3      ;COPY THE BIT MAP TO BE TRANSFORMED.
                MOV          #NUMLNS,R4 ;SET MAX LINE COUNTER.
                CLR          R5         ;CLEAR ASSOCIATED LINE BIT MAP.
2$:             ASR          R3        ;SHIFT ACTLNS BIT MAP INT BOOLEAN REGISTER.
                BCC          4$        ;SKIP SETTING ASSOCIATED LINE NUMBER BIT MAP.
                MOV          (R2),R1   ;GET ASSOCIATED LINE NUMBER OFFSET FROM TABLE.
                ASR          R1        ;SHIFT RIGHT TO GET LINE NUMB FROM OFFSET.
                JSR          PC,LINBIT ;GENERATE AN SINGLE BIT MAP FOR THIS LINE.
                BIS          R0,R5     ;SET BIT FOR THIS LINE IN ASSOCIATED BIT MAP.
                TST          (R2)+    ;INCREMENT ADDRESS FOR THE NEXT LINE NUMBER.
                DEC          R4        ;DECREMENT LINE COUNT.
                BNE          2$        ;LOOP IF NOT DONE.
60$:           PASS          R5       ;RESTORE GPRS, EXCEPT
                MOV          R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
                JSR          PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
                ;R5 - CONTAINS THE ASSOCIATED LINE BIT MAP.

                RTS          PC

```

2771  
2772  
2773  
2774  
2775  
2776  
2777  
2778  
2779  
2780  
2781  
2782  
2783  
2784  
2785  
2786  
2787  
2788  
2789 013716  
013716 004567 170172  
2790 013722 010401  
2791 013724 012702 177777  
2792 013730 005003  
2793 013732 012704 013754  
2794 013736 004767 000544  
2795 013742 103002  
2796 013744 004767 000552  
2797 013750  
013750 004736  
2798 013752 000207  
2799  
2800 013754 177777

```
.SBTTL GLOBAL SUBROUTINE - DELAY -
;*****
;* - DELAY SUBROUTINE -
;* THIS SUBROUTINE IS USED TO DELAY A VARIABLE NUMBER OF MILLI-SECONDS.
;*
;* INPUTS: R4 - CONTAINS THE NUMBER OF MS TO DELAY.
;* MSLCNT.
;*
;* OUTPUTS: NONE.
;*
;* CALLING SEQUENCE: JSR PC,DELAY
;*
;* COMMENTS: IF NO HARDWARE CLOCK INTERRUPTS ARE OCCURING, CONTROL-CS WILL
;* NOT BE HONORED FOR THE DURATION OF THE DELAY.
;*
;* SUBORDINATE ROUTINES CALLED: NONE.
;*****
DELAY:: SAVE
MOV R4,R1 JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
MOV #-1,R2 ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
CLR R3 ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
MOV #62#,R4 ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
JSR PC,MSLOOP ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
BCC 60# ;DELAY THE REQUESTED # OF MS.
JSR PC,OOPS ;EXIT ROUTINE IF WE TIMED-OUT.]
;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
;RESTORE GPRS.
60#: PASS JSR PC,8(SP) ;RETURN TO PREG05 SUBRT.
RTS PC
62#: .WORD -1 ;DUMMY, NON-ZERO WORD.
```

2802  
2803  
2804  
2805  
2806  
2807  
2808  
2809  
2810  
2811  
2812  
2813  
2814  
2815  
2816  
2817  
2818  
2819  
2820  
2821  
2822  
2823  
2824  
2825  
2826  
2827  
2828  
2829  
2830  
2831  
2832  
2833 013756  
013756 004567 170132  
2834 013762 012704 000200  
2835 013766 005767 166354  
2836 013772 001427  
2837  
2838  
2839  
2840  
2841 013774 010205  
2842 013776 012700 000005  
2843 014002 006105  
2844 014004 005300  
2845 014006 001375  
2846 014010 042705 177761  
2847 014014 066705 166330  
2848 014020 011505  
2849 014022 012700 000006  
2850 014026 006305  
2851 014030 006104  
2852 014032 005300  
2853 014034 001374  
2854 014036 042702 160000  
2855 014042 060502  
2856 014044 005504  
2857 014046 052704 000200

```

.SBTTL GLOBAL SUBROUTINE - DODMA -
;+ *****
;* - INITIATE DMA TRANSMISSION ROUTINE -
;* THIS ROUTINE WRITES THE DMA PARAMETER TO THE SPECIFIED DEVICE AND
;* INITIATES THE DMA TRANSMISSION.
;*
;* INPUTS: R1 - LINE NUMBER ON WHICH TO INITIATE THE DMA.
;* R2 - START ADDRESS OF THE DMA BUFFER (16 BIT VIRTUAL).
;* R3 - CHARACTER COUNT OF THE DMA BUFFER.
;* CSRA - CONTAINS ADDRESS OF THE DUT CSR.
;* IESTAT - STORAGE FOR STATES OF THE INTERRUPT ENABLE BITS.
;* MMENAB - MEMORY MANAGEMENT FLAG (0 IF MEM MGT NOT ENABLED).
;* HOST MEM MGT PAR REGISTERS - IF MEM MGT IS IN USE.
;* TXAD1A - CONTAINS ADDRESS OF DMA TX BUFFER ADDRESS REG #1.
;* TXAD2A - CONTAINS ADDRESS OF DMA TX BUFFER ADDRESS REG #2.
;* TXBFCA - CONTAINS ADDRESS OF DMA CHARACTER COUNT REGISTER.
;*
;* OUTPUTS: CARRY - SUCCESS FLAG (SET IF DMA_START FOUND CLEAR).
;* DUT TBUFFAD1 - LS 16 BITS OF DMA BUFFER ADDRESS (INITIALIZED).
;* DUT TBUFFAD2 - MS 6 BITS OF DMA BUFFER ADDRESS (INITIALIZED).
;* DMA_START BIT SET.
;* DUT TBUFFCT - DMA BUFFER CHARACTER COUNT (INITIALIZED).
;*
;* CALLING SEQUENCE: JSR PC,DODMA
;*
;* COMMENTS: THIS ROUTINE DETERMINES IF MEMORY MANAGEMENT IS BEING USED
;* AND SETS UP THE FULL 22 BIT PHYSICAL ADDRESS IF NECESSARY.
;*
;* SUBORDINATE ROUTINES CALLED: NONE.
;-- *****
DODMA:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
MOV #200,R4 ;PREPARE TO CLEAR UPPER 6 BITS OF DMA BUFF ADR.
TST MMENAB ;CHECK FOR MEMORY MANAGEMENT IN USE.
BEQ 66 ;GOTO SET UP DEVICE IF MEM MGT NOT IN USE.
;+
; MEMORY MANAGEMENT IS IN USE.
; CONSTRUCT 22 BIT PHYSICAL ADDRESS FROM THE 16 BIT VIRTUAL ADDRESS.
;-
MOV R2,R5 ;STRIP THE MOST SIGNIFICANT 3 BITS OF THE
MOV #5,R0 ;DMA BUFFER VIRTUAL ADDRESS AND MULTIPLY
24: ROL R5 ;THEIR VALUE BY TWO TO GET AN OFFSET INTO
DEC R0 ;THE TABLE OF MEMORY MANAGEMENT PAGE
BNE 24 ;ADDRESS REGISTERS (PAR).
BIC #177761,R5
ADD PAR0A,R5 ;ADD IN THE BASE VALUE OF THE MM PAR REGISTERS.
MOV (R5),R5 ;GET THE 16 BIT PHYSICAL ADDRESS BLOCK COUNT.
MOV #6,R0 ;SHIFT UPPER 6 BITS OF THE PHYSICAL ADDRESS
44: ASL R5 ;BLOCK COUNT (GOTTEN FROM THE PROPER PAR)
ROL R4 ;INTO THE LS 6 BITS OF THE WORD TO WRITE
DEC R0 ;INTO THE DUT TBUFFAD2 REGISTER.
BNE 44
BIC #160000,R2 ;ADD THE 13 BIT DISPLACEMENT FIELD FROM VIRTUAL
ADD R5,R2 ;ADR TO THE SHIFTED BLOCK NUMBER FROM THE
ADC R4 ;MEMORY MANAGEMENT PAR.
BIS #200,R4 ;SET THE DMA_START BIT IN WORD FOR TBUFFAD2.

```

```

2858
2859 ; WRITE THE DMA PARAMETERS OUT TO THE DUT DMA REGISTERS.
2860 ; DISABLE INTERRUPTS.
2861 ; SET UP DUT CSR IND.ADR.REG FIELD.
2862 ; WRITE THE DMA TRANSMIT CHARACTER COUNT.
2863 ; WRITE THE LEAST SIGNIFICANT 16 BITS OF THE DMA BUFFER START ADDRESS.
2864 ; WRITE THE MOST SIGNIFICANT 6 BITS OF THE ADDRESS,
2865 ; SETTING THE DMA_START BIT, AND INITIATING THE DMA TRANSMISSION.
2866 ;
2867 014052 60: GETPRI R5 ;GET THE PRESENT PROCESSOR PRIORITY.
      014052 104440 ;
      014054 010005 ; TRAP C#GPRI
      ; MOV RO,R5
2868 014056 SETPRI #PRI07 ;DISABLE ALL HARDWARE INTERRUPTS.
      014056 012700 000340 ;
      014062 104441 ; MOV #PRI07,RO
      ; TRAP C#SPRI
2869 014064 056701 166200 BIS IESTAT,R1 ;PREPARE FOR SETUP OF LINE NUMBER IN DUT CSR.
2870 014070 010177 166146 MOV R1,@CSRA ;SET UP THE DUT CSR IND.ADR.REG FIELD.
2871 014074 105777 166156 TSTB @TXAD2A ;TEST THE DUT DMA_START BIT.
2872 014100 000241 CLC ;INDICATE FAILURE IN CASE DMA.HO BIT IS SET.
2873 014102 100411 BMI 60$ ;EXIT WITH FAILURE IF DMA.HO BIT IS SET.
2874 014104 010377 166150 MOV R3,@TXBFCA ;WRITE THE DMA CHARACTER COUNT.
2875 014110 010277 166140 MOV R2,@TXAD1A ;WRITE THE LS 16 BITS OF BUFFER ADDRESS.
2876 014114 110477 166136 MOVB R4,@TXAD2A ;WRITE MS 6 BITS OF ADR AND START DMA TX.
2877 014120 SETPRI R5 ;RESTORE THE PROCESSOR PRIORITY.
      014120 010500 ;
      014122 104441 ; MOV R5,RO
      ; TRAP C#SPRI
2878 014124 000261 SEC ;INDICATE SUCCESS.
2879
2880 014126 60$: PASS ;RESTORE GPRS.
      014126 004736 ; PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
2881 014130 000207 RTS PC JSR ; CARRY - SUCCESS FLAG (SET IF SUCCESS).

```

```

2883 .SBTTL GLOBAL SUBROUTINE - FINACT -
2884 ;+ *****
2885 ;* - FIND FIRST ACTIVE LINE -
2886 ;* THIS SUBROUTINE CALCULATES THE NUMBER OF THE FIRST ACTIVE LINE THAT
2887 ;* IS FOUND IN THE ACTIVE LINE BIT MAP ACTLNS.
2888 ;*
2889 ;* INPUTS: ACTLNS - CONTAINS THE ACTIVE LINE BIT MAP.
2890 ;*
2891 ;* OUTPUTS: R1 - CONTAINS THE NUMBER OF THE FIRST ACTIVE LINE.
2892 ;* R5 - CONTAINS THE BIT MAP REPRESENTATION OF THE ACTIVE LINE.
2893 ;* CARRY SET INDICATES SUCCESS.
2894 ;*
2895 ;* CALLING SEQUENCE: JSR PC,FINACT
2896 ;*
2897 ;* COMMENTS:
2898 ;*
2899 ;* SUBORDINATE ROUTINES CALLED: NONE.
2900 ;-- *****
2901
2902 014132 FINACT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
014132 004567 167756 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2903
2904 ;+
2905 ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
2906 ;-
2906 014136 005001 CLR R1 ;CLEAR THE LINE NUMBER COUNTER.
2907 014140 012703 000010 MOV #NUMLNS,R3 ;GET MAX LINE NUMBER.
2908 014144 016700 166064 MOV ACTLNS,R0 ;GET THE ACTIVE LINE BIT MAP.
2909 014150 012705 000001 MOV #1,R5 ;SET UP A LINE BIT MASK.
2910 014154 030500 2#: BIT R5,R0 ;LOOK FOR AN ACTIVE LINE.
2911 014156 001006 BNE 4# ;BRANCH TO BEGIN TEST IF A LINE HAS BEEN FOUND.
2912 014160 006305 ASL R5 ;SHIFT THE BIT MASK FOR THE NEXT LINE.
2913 014162 005201 INC R1 ;INCREMENT THE LINE NUMBER COUNTER.
2914 014164 020103 CMP R1,R3 ;CHECK IF ALL LINES HAVE BEEN TRIED.
2915 014166 002772 BLT 2# ;LOOP TO TRY THE NEXT LINE.
2916 014170 000241 CLC ;CLEAR CARRY BIT, NO ACTIVE LINE FOUND.
2917 014172 000401 BR 60# ;EXIT WITH FAILURE.
2918 014174 000261 4#: SEC ;SET CARRY, SUCCESS.
2919
2920 014176 60#: PASS R1,R5 ;RESTORE GPRS, EXCEPT
014176 010166 000004 MOV R1,R1SLOT(SP) ;PUT R1 IN STACK SLOT.
014202 010566 000014 MOV R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
014206 004736 JSR PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
2921 ;R1 - CONTAINS THE NUMBER OF FIRST ACTIVE LINE.
2922 ;R5 - CONTAINS THE BIT MAP OF THE ACTIVE LINE.
2923 ;CARRY - SET INDICATES SUCCESS.
2924 014210 000207 RTS PC

```

```

2926 .SBTTL GLOBAL SUBROUTINE - INDATP -
2927 ;* *****
2928 ;* - INITIALISE DATA PATTERN -
2929 ;* THIS SUBROUTINE IS USED TO INITIALISE AN INCREMENTAL BYTE DATA PATTERN
2930 ;* IN THE GENERAL BUFFER AREA.
2931 ;* THE DATA PATTERN WILL BE SEQUENTIAL FROM 0 TO 255 (DECIMAL).
2932 ;*
2933 ;* INPUTS: BUFBAS - ADDRESS OF THE START OF THE GENERAL BUFFER AREA.
2934 ;* BUFMID - ADDRESS OF THE 255 TH LOCATION.
2935 ;*
2936 ;* OUTPUTS: THE FIRST 255 LOCATIONS OF THE GENERAL BUFFER AREA CONTAIN DATA
2937 ;*
2938 ;* CALLING SEQUENCE: JSR PC,INDATP
2939 ;*
2940 ;* COMMENTS:
2941 ;*
2942 ;* SUBORDINATE ROUTINES CALLED: NONE.
2943 ;*
2944 ;*
2945 014212 INDATP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
014212 004567 167676 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2946
2947 014216 012702 002704 MOV #BUFBAS,R2 ;INITIALIZE THE DATA PATTERN IN THE GENERAL
2948 014222 005003 CLR R3 ; DATA BUFFER TO A 256 BYTE PATTERN.
2949 014224 110322 2$: MOVB R3,(R2)+ ;
2950 014226 005203 INC R3 ;SELECT THE NEXT CHARACTER.
2951 014230 020227 003304 CMP R2,#BUFMID ;CHECK IF WE HAVE 256 DATA PATTERNS.
2952 014234 103773 BLO 2$ ;
2953
2954 014236 60$: PASS ;RESTORE GPRS.
014236 004736 JSR PC,#(SP)+ ;RETURN TO PREG05 SUBRT.
2955 014240 000207 RTS PC

```

```

2957 .SBTTL GLOBAL SUBROUTINE - INDTPX -
2958 ;* *****
2959 ;* - INITIALISE DATA PATTERN WITHOUT XON OR XOFF -
2960 ;* THIS SUBROUTINE IS USED TO INITIALISE AN INCREMENTAL BYTE DATA PATTERN
2961 ;* IN THE GENERAL BUFFER AREA.
2962 ;* THE DATA PATTERN WILL BE FROM 0 TO 255, BUT WILL EXCLUDE THE FOLLOWING
2963 ;* TWO CHARACTERS; (ASCII DC1, DC3) XON AND XOFF. THIS WILL CAUSE THE
2964 ;* LAST TWO DATA CHARACTERS TO BE THE SAME AS THE FIRST TWO.
2965 ;*
2966 ;* INPUTS: BUFBAS - ADDRESS OF THE START OF THE GENERAL BUFFER AREA.
2967 ;*          BUFMID - ADDRESS OF THE 255 TH LOCATION.
2968 ;*
2969 ;* OUTPUTS: THE FIRST 255 LOCATIONS OF THE GENERAL BUFFER AREA CONTAIN DATA
2970 ;*
2971 ;* CALLING SEQUENCE: JSR PC,INDTPX
2972 ;*
2973 ;* COMMENTS:
2974 ;*
2975 ;* SUBORDINATE ROUTINES CALLED: NONE.
2976 ;* - - - - -
2977
2978 014242 004567 167646 INDTPX:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
2979 014242 ; R5,PREG05 ;CALL REGISTER SAVE SUBRT.
2980 ;*
2981 ; INITIALIZE THE 256 BYTE DATA PATTERN.
2982 ; ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
2983 ; NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
2984 ; -
2984 014246 012702 002704 MOV #BUFBAS,R2 ;INITIALIZE THE DATA PATTERN IN THE GENERAL
2985 014252 005003 CLR R3 ; DATA BUFFER TO A 256 BYTE PATTERN.
2986 014254 110322 2#: MOVB R3,(R2)+ ;
2987 014256 105203 INCB R3 ;SELECT THE NEXT CHARACTER.
2988 014260 122703 000021 CMPB #21,R3 ;CHECK FOR AN XON CHARACTER.
2989 014264 001001 BNE 4# ;BRANCH IF CHAR NOT AN XON.
2990 014266 105203 INCB R3 ;FORCE THE NEXT CHARACTER.
2991 014270 122703 000023 4#: CMPB #23,R3 ;CHECK FOR AN XOFF CHARACTER.
2992 014274 001001 BNE 6# ;BRANCH IF NOT AN XOFF CHARACTER.
2993 014276 105203 INCB R3 ;FORCE THE NEXT CHARACTER.
2994 014300 020227 003304 6#: CMP R2,#BUFMID ;CHECK IF WE HAVE 256 DATA PATTERNS.
2995 014304 103763 BLO 2# ;
2996
2997 014306 60#: PASS ;RESTORE GPRS.
2998 014306 004736 JSR PC,#(SP)+ ;RETURN TO PREG05 SUBRT.
2998 014310 000207 RTS PC

```



3000  
3001  
3002  
3003  
3004  
3005  
3006  
3007  
3008  
3009  
3010  
3011  
3012  
3013  
3014  
3015  
3016  
3017  
3018  
3019  
3020  
3021  
3022  
3023  
3024  
3025  
3026  
3027  
3028  
3029

014312  
014312 004567 167576  
014316 042701 177760  
014322 006301  
014324 016100 002370  
014330  
014330 010066 000002  
014334 004736  
014336 000207

```

.SBTTL GLOBAL SUBROUTINE - LINBIT -
;.. *****
; * - LINE NUMBER TO BIT MAP CONVERSION SUBROUTINE -
; * THIS SUBROUTINE IS USED TO GENERATE A BIT MAP (ONE BIT OF 16 SET)
; * BASED ON A LINE NUMBER (RANGE: 1 TO 16). ONLY THE LS 4 BITS OF THE
; * LINE NUMBER WORD ARE USED, THE OTHERS ARE MASKED OUT (SO UNMASKED
; * MSBYTES OF DUT CSRS CAN BE PASSED TO THIS ROUTINE WITHOUT ERROR).
; *
; * INPUTS: R1 - LINE NUMBER (ONLY LS 4 BITS USED, OTHERS DISREGARDED).
; * BITTBL - BASE LABEL OF A 16 WORD BIT TABLE.
; *
; * OUTPUTS: R0 - BIT MAP, BIT CORRESPONDING TO LINE NUMBER IS SET:
; * IF LINE NUMBER IS 3, THEN BIT3 IS SET, ETC.
; *
; * CALLING SEQUENCE: JSR PC,LINBIT
; *
; * COMMENTS: NO CHECKING IS PERFORMED TO VERIFY THAT THE LINE NUMBER IS
; * A LEGAL LINE NUMBER FOR THE DUT (IE - LESS THAN NUMLNS).
; * NOTE: THE LINE NUMBER IS NOT DESTROYED OF ALTERED, SO THIS
; * ROUTINE CAN BE USED EASILY IN LOOPS.
; *
; * SUBORDINATE ROUTINES CALLED: NONE.
;-- *****
LINBIT:: SAVE R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
;CALL REGISTER SAVE SUBRT.
BIC #177760,R1 ;MASK OUT ALL BUT 4 LSBITS OF THE LINE #.
ASL R1 ;MULTIPLY LINE # BY 2 TO GET WORD TABLE OFFSET.
MOV BITTBL(R1),R0 ;GET THE SINGLE BIT BIT MAP.
601: PASS R0 ;RESTORE GPRS, EXCEPT THE FOLLOWING.
MOV R0,ROSLOT(SP) ;PUT R0 IN STACK SLOT.
JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
RTS PC ;R0 - BIT MAP WITH LINE # BIT SET.

```

```

3031 .SBTTL GLOBAL SUBROUTINE - MAPCNT -
3032 ;* *****
3033 ;* - COUNT BITS IN BIT MAP ROUTINE -
3034 ;* THIS SUBROUTINE COUNTS THE NUMBER OF BITS WHICH ARE SET IN A BIT MAP.
3035 ;*
3036 ;* INPUTS: R2 - THE BIT MAP FOR WHICH TO COUNT THE BITS.
3037 ;*
3038 ;* OUTPUTS: R2 - COUNT OF THE NUMBER OF BITS THAT WERE SET.
3039 ;*
3040 ;* CALLING SEQUENCE: JSR PC,MAPCNT
3041 ;*
3042 ;* COMMENTS:
3043 ;*
3044 ;* SUBORDINATE ROUTINES CALLED: NONE.
3045 ;* - - - - -
3046
3047 014340 004567 167550 MAPCNT:: SAVE JSR ;SAVE CONTENTS OF GPRS R0 THRU R5.
014340 010201 R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3048 014344 001405 MOV R2,R1
3049 014346 001405 BEQ 601 ;EXIT WITH ZERO IF NO BITS ARE SET IN MAP.
3050
3051 014350 005002 CLR R2 ;CLEAR THE BIT COUNT.
3052 014352 000261 SEC ;COUNT THE LAST BIT TO BE SHIFTED OUT.
3053
3054 014354 005502 21: ADC R2 ;COUNT THE BIT IF IT WAS SET.
3055 014356 006301 ASL R1 ;SHIFT ANOTHER BIT OUT OF THE MAP.
3056 014360 001375 BNE 21 ;LOOP IF ALL BITS NOT SHIFTED OUT OF MAP.
3057
3058 014362 010266 000006 601: PASS R2 ;RESTORE GPRS, EXCEPT THE FOLLOWING:
014362 004736 MOV R2,R2SLOT(SP) ;PUT R2 IN STACK SLOT.
014366 004736 JSR PC,B(SP) ;RETURN TO PREG05 SUBRT.
3059 014370 000207 RTS PC ; R2 - COUNT OF BITS SET IN BIT MAP.

```

3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078  
3079  
3080  
3081  
3082  
3083  
3084  
3085  
3086  
3087  
3088  
3089  
3090  
3091  
3092  
3093  
3094  
3095  
3096  
3097  
3098  
3099 014372 004567 167516  
3100  
3101  
3102  
3103  
3104 014376 005102  
3105 014400 040203  
3106  
3107  
3108  
3109 014402 005701  
3110 014404 001011  
3111 014406 011400  
3112 014410 010067 000070  
3113 014414 040200  
3114 014416 020003  
3115 014420 000261  
3116 014422 001420

```

.SBTTL GLOBAL SUBROUTINE - MSLGET -
*****
;* - MILLI SECONDS LOOP WHICH RETURNS READ WORD AND REMAINING TIME -
;* THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
;* TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD. THE
;* CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
;* DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
;* THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
;* ROUTINE AND THEN ONCE EACH MILLI-SECOND THERE AFTER.
;* UPON RETURN, THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION
;* IS RETURNED BY THIS SUBROUTINE.
;*
;* INPUTS: R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
;* R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
;* R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
;* R4 - ADDRESS OF THE WORD TO TEST.
;* MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
;*
;* OUTPUTS: R0 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
;* R1 - REMAINING NUMBER OF MS IN TIME-OUT TIME.
;* CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
;*
;* CALLING SEQUENCE: JSR PC,MSLGET
;*
;* COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
;* CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
;* ON THE SYSTEM.
;* THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
;* DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
;* LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
;* IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
;* THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
;* IF THE CONDITION IS MET, FAILURE OTHERWISE.
;*
;* SUBORDINATE ROUTINES CALLED: NONE.
*****
MSLGET:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
; R5,PREG05 ;CALL REGISTER SAVE SUBRT.
;*
;* SET UP MASK FOR REMOVING UNUSED BITS IN THE TEST WORD, AND CLEAR UNUSED
;* BITS IN THE DESIRED STATE WORD TO ALLOW DIRECT COMPARISON.
;*-
COM R2 ;GET MASK OF UNUSED BITS.
BIC R2,R3 ;MASK OUT UNUSED BITS IN DESIRED STATE WORD.
;*
;* HANDLE THE TEST AND EXIT IF WE HAVE A 0 TIME-OUT VALUE.
;*-
TST R1 ;TEST THE TIME-OUT VALUE FOR ZERO.
BNE 2$ ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
MOV (R4),R0 ;GET THE WORD TO TEST BEFORE EXITING.
MOV R0,62$ ;SAVE VALUE SO WE CAN RETURN IT.
BIC R2,R0 ;MASK OUT UNTESTED BITS OF WORD.
CMP R0,R3 ;COMPARE AGAINST DESIRED STATE WORD.
SEC ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
BEQ 6$ ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.

```

```

3117 014424 000241          CLC          ;INDICATE FAILURE (TIME-OUT).
3118 014426 000416          BR          6$          ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
3119                          ;*
3120                          ; NON-ZERO TIME-OUT VALUE. LOOP, WAITING FOR CONDITION OR TIME-OUT.
3121                          ;-
3122 014430 016705 165704 2$: MOV MSLCNT,R5 ;LOAD MS LOOP COUNT.
3123 014434 011400 4$: MOV (R4),R0 ;GET THE WORD TO TEST.
3124 014436 010067 000042 MOV R0,62$ ;SAVE WORD IN CASE THIS IS THE LAST.
3125 014442 040200 BIC R2,R0 ;MASK OUT UNTESTED BITS OF WORD.
3126 014444 020003 CMP R0,R3 ;COMPARE AGAINST DESIRED STATE WORD.
3127 014446 000261 SEC ;SET CARRY IN CASE OF SUCCESS.
3128 014450 001405 BEQ 6$ ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
3129 014452 005305 DEC R5 ;COUNT DOWN THE INSIDE MS LOOP COUNT.
3130 014454 001367 BNE 4$ ;LOOP IF MS NOT UP.
3131 014456 005301 DEC R1 ;DECREMENT THE MS TIME COUNT.
3132 014460 001363 BNE 2$ ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
3133 014462 000241 CLC ;CLEAR CARRY, WE TIMED-OUT.
3134                          ;*
3135                          ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME-OUT VALUE).
3136                          ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD. EXIT ROUTINE.
3137                          ;-
3138 014464 016700 000014 6$: MOV 62$,R0 ;PASS OUT THE LAST READ WORD.
3139 014470 014470 010066 000002 60$: PASS R0,R1 ;RESTORE GPRS, EXCEPT THE FOLLOWING:
                                MOV R0,R0SLOT(SP) ;PUT R0 IN STACK SLOT.
                                MOV R1,R1SLOT(SP) ;PUT R1 IN STACK SLOT.
                                JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
3140                          ;R0 - LAST READ WORD CHECKED FOR CONDITION.
3141                          ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
3142 014502 000207 RTS PC ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
3143                          ;*
3144                          ; LOCAL STORAGE.
3145                          ;-
3146 014504 000000 62$: .WORD 0 ;STORAGE FOR THE LAST READ WORD.

```

3148  
3149  
3150  
3151  
3152  
3153  
3154  
3155  
3156  
3157  
3158  
3159  
3160  
3161  
3162  
3163  
3164  
3165  
3166  
3167  
3168  
3169  
3170  
3171  
3172  
3173  
3174  
3175  
3176  
3177  
3178  
3179  
3180  
3181 014506  
014506 004567 167402  
3182  
3183  
3184  
3185  
3186  
3187 014512 004767 177654  
3188  
3189 014516  
014516 004736  
3190 014520 000207

```

.SBTTL GLOBAL SUBROUTINE - MSLOOP -
;*****
;* - TEST LOOP SUBROUTINE -
;* THIS SUBROUTINE IS A GENERAL PURPOSE TEST LOOP SUBROUTINE. IT IS USED
;* TO VERIFY THAT A CERTAIN ACTION OCCURS BEFORE A TIME-OUT PERIOD. THE
;* CALLING ROUTINE PASSES IN WHICH BITS SHOULD BE SET AND CLEARED FOR THE
;* DESIRED CONDITION AND THE TIME-OUT VALUE IN MILLI-SECONDS.
;* THIS ROUTINE CHECKS FOR THE DESIRED CONDITION UPON ENTRANCE INTO THE
;* ROUTINE AND THEN ONCE EACH MILLI-SECOND THEREAFTER.
;*
;* INPUTS: R1 - TIME-OUT VALUE IN MILLI-SECONDS (UP TO 64K MS).
;* R2 - BIT MAP OF BITS TO TEST (1 INDICATES TO TEST THE BIT).
;* R3 - DESIRED STATES OF THE INDICATED FIELDS IN R2.
;* R4 - ADDRESS OF THE WORD TO TEST.
;* MSLCNT - MILLI SECOND SOFTWARE LOOP COUNT.
;*
;* OUTPUTS: CARRY - SUCCESS FLAG (SET IF CONDITION IS MET BEFORE TIME-OUT).
;*
;* CALLING SEQUENCE: JSR PC,MSLOOP
;*
;* COMMENTS: THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK, BUT THE
;* CALIBRATION IS ONLY GUARENTEED WHEN A LINE CLOCK IS AVAILABLE
;* ON THE SYSTEM.
;* THIS ROUTINE CAN BE USED AS A DELAY ROUTINE, BY SPECIFYING THE
;* DESIRED DELAY AS THE TIME-OUT AND SPECIFYING A CONDITION TO
;* LOOK FOR WHICH WILL NOT BE MET DURING THE DELAY.
;* IF A TIME-OUT VALUE OF 0 IS SPECIFIED, THIS ROUTINE CHECKS FOR
;* THE DESIRED CONDITION BEFORE RETURNING. IT INDICATES SUCCESS
;* IF THE CONDITION IS MET, FAILURE OTHERWISE.
;*
;* SUBORDINATE ROUTINES CALLED: MSLGET.
;*****
MSLOOP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
; JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
;
;*
;* CALLING THE MSLGET ROUTINE FROM THE MSLOOP ROUTINE ISOLATES THE CALLER OF
;* MSLOOP FROM THE RETURNED TEST WORD AND REMAINING TIME-OUT VALUES.
;*
;*
;* JSR PC,MSLGET ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.
604: PASS ;RESTORE GPRS.
; JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
RTS PC ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.

```

```

3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211 014522
      014522 004567 167366
3212
3213 014526
      014526 104454
      014530 000145
      014532 014566
      014534 000000
3214
3215 014536
      014536 012746 014652
      014542 012746 000001
      014546 010600
      014550 104417
      014552 062706 000004
3216 014556
      014556 104422
3217 014560 000776
3218 014562
      014562 004736
3219 014564 000207
3220
3221 014566      110      117      123
      014571      124      040      103
      014574      117      115      120
      014577      125      124      105
      014602      122      040      110
      014605      101      122      104
      014610      127      101      122
      014613      105      040      117
      014616      122      040      123
      014621      117      106      124
      014624      127      101      122
      014627      105      040      102
      014632      125      107      040
      014635      105      116      103
      014640      117      125      116
      014643      124      105      122

```

```

.SBTTL GLOBAL SUBROUTINE - OOPS -
;+ *****
;* - PROGRAM ABORT SUBROUTINE -
;* THIS SUBROUTINE IS USED TO ABORT THE PROGRAM WHEN A FATAL ERROR IS
;* DETECTED IN THE PROGRAM OR THE HOST SYSTEM HARDWARE. AN ERROR MESSAGE
;* IS PRINTED GIVING SOME INFORMATION ABOUT THE NATURE OF THE ABORT.
;*
;* INPUTS: R1 - ERROR CODE GIVING REASON FOR ABORT.
;*
;* OUTPUTS: AN ERROR MESSAGE IS PRINTED.
;*          A LIST OF RETURN PC VALUES FOR ALL SUBROUTINE CALLS IS PRINTED.
;*
;* CALLING SEQUENCE: JSR PC,OOPS
;*
;* COMMENTS:
;*
;* SUBORDINATE ROUTINES CALLED: NONE.
;-- *****

OOPS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
        JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
        ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
        ERRSF 101,EM0101

        TRAP C#ERSF
        .WORD 101
        .WORD EM0101
        .WORD 0

        ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
        PRINTF #EM0102

        MOV #EM0102,-(SP)
        MOV #1,-(SP)
        MOV SP,R0
        TRAP C#PNTF
        ADD #4,SP

2#: BREAK ;LOOK FOR OPERATOR CONTROL-C INPUT.
        TRAP C#BRK

60#: BR 2# ;INFINITE LOOP.
        PASS ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
        JSR PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
        RTS PC ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.

EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./

```

	014646	105	104	056
	014651	000		
3222	014652	045	116	045
	014655	101	120	122
	014660	117	107	122
	014663	101	115	040
	014666	110	125	116
	014671	107	054	040
	014674	127	101	111
	014677	124	111	116
	014702	107	040	106
	014705	117	122	040
	014710	101	040	103
	014713	117	116	124
	014716	122	117	114
	014721	055	103	056
	014724	040	074	052
	014727	052	052	052
	014732	052	052	052
	014735	052	052	052
	014740	052	052	052
	014743	045	116	045
3223	014746	116	000	

EM0102:: .ASCIZ /N\$APROGRAM HUNG, WAITING FOR A CONTROL-C. <\*\*\*\*\*N\$N/

.EVEN

```

3225
3226
3227
3228
3229
3230
3231
3232
3233
3234
3235
3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248 014750
      014750 004567 167140
3249 014754 016701 165262
3250 014760 016702 165262
3251 014764 042703 177760
3252 014770 056703 165274
3253 014774 010311
3254 014776 011204
3255
3256 015000
      015000 010446
      015002 012746 011025
      015006 012746 005274
      015012 012746 000003
      015016 010600
      015020 104415
      015022 062706 000010
3257 015026
      015026 004736
3258 015030 000207

```

```

.SBTTL GLOBAL SUBROUTINE - PRTLPR -
; * *****
; * -PRINT THE CONTENTS OF THE LPR.
; * THIS ROUTINE IS USED TO PRINT OUT EXTENDED INFORMATION ON THE
; * CONTENTS OF THE LINE PARAMETER REGISTER (LPR).
; *
; * INPUTS: R3 - CONTAINS THE NUMBER OF THE LINE YOU WISH TO EXAMINE.
; * CSRA - CONTAINS THE ADDRESS OF THE DUT'S CSR.
; * IESTAT - CONTAINS THE CURRENT STATUS OF THE TX AND RX INTERRUPT
; * ENABLE BITS IN THE DUT'S CSR.
; * LPRA - CONTAINS THE ADDRESS OF THE DUT'S LPR REGISTER.
; *
; * OUTPUTS: AN EXTENDED INFORMATION MESSAGE IS PRINTED ON THE OPERATORS
; * CONSOLE.
; *
; * CALLING SEQUENCE: JSR PC,PRTLPR
; *
; * COMMENTS: THIS ROUTINE CHANGES THE INDIRECT ADDRESS FIELD OF THE DEVICE
; * UNDER TEST'S CSR.
; *
; * SUBORDINATE ROUTINES CALLED: NONE.
; * - - *****
PRTLPR::SAVE
      JSR R5,PREG05 ;SAVE CONTENTS OF GPRS R0 THRU R5.
      MOV CSRA,R1 ;CALL REGISTER SAVE SUBRT.
      MOV LPRA,R2 ;GET THE CSR ADDRESS.
      BIC #177760,R3 ;GET THE LPR ADDRESS.
      BIS IESTAT,R3 ;CLEAR ANY UNWANTED BITS.
      MOV R3,(R1) ;SET STATE OF TX AND RX INTERRUPT ENABLE BITS.
      MOV (R2),R4 ;SELECT LINE.
      ;PRINT MESSAGE "CONTENTS OF THE LPR:NNNNNN"
      PRINTX #EF9019,#EM9026,R4 ;GET CONTENTS OF THE LPR.
      ;PRINT OUT MESSAGE ON OPERATORS CONSOLE.
      MOV R4,-(SP)
      MOV #EM9026,-(SP)
      MOV #EF9019,-(SP)
      MOV #3,-(SP)
      MOV SP,R0
      TRAP C#PNTX
      ADD #10,SP
604: PASS ;RESTORE GPRS.
      RTS PC JSR PC,#(SP)+ ;RETURN TO PREG05 SUBRT.

```



```

3260 .SBTTL GLOBAL SUBROUTINE - PUFIFO -
3261 ;*****
3262 ;* - PURGE THE FIFO
3263 ;* THIS ROUTINE TRIES TO REMOVE ALL THE CHARACTERS FROM THE FIFO.
3264 ;* ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE BMP CODE QUEUE.
3265 ;*
3266 ;* INPUTS: RBUFA- CONTAINS THE ADDRESS OF THE RECEIVER.
3267 ;*
3268 ;*
3269 ;* OUTPUTS: CARRY BIT - INDICATES THE STATE OF THE FIFO, SET:= PURGED.
3270 ;* BMPCQ - THE CONTENTS OF THE BMP CODE QUEUE MAY BE UPDATED.
3271 ;*
3272 ;* CALLING SEQUENCE: JSR PC,PUFIFO
3273 ;*
3274 ;* COMMENTS:
3275 ;*
3276 ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
3277 ;*****
3278
3279 015032 PUFIFO::SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
015032 004567 167056 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3280 015036 012701 001000 MOV #512,R1 ;SET MAXIMUM TRY COUNT OF 512.
3281 015042 016704 165176 MOV RBUFA,R4 ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
3282
3283 015046 011402 2#: MOV (R4),R2 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
3284 015050 100016 BPL 6# ;EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.
3285
3286 ;*
3287 ; CHECK IF THE READ CHARACTER IS ACTUALLY A BMP CODE.
3288 ; IF IT IS, THEN SAVE IT ON THE BMP CODE QUEUE TO BE REPORTED LATER.
3289 015052 012700 070000 ;*
3290 015056 040200 MOV #70000,R0 ;GENERATE A BIT MAP OF CHAR ERROR BITS
3291 015060 001006 BIC R2,R0 ; WHICH ARE NOT SET FOR CHAR.
3292 BNE 4# ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
3293
3294 ;*
3295 ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
3296 015062 012700 000300 ;*
3297 015066 040200 MOV #300,R0 ; CHECK IF BMP OR SELFTEST?.
3298 015072 004767 000444 BIC R2,R0 ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
3299 BNE 4# ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
3300 015076 005301 JSR PC,SAVBMP ;SAVE BMP CODE ON THE QUEUE.
3301 015100 001362 4#: DEC R1 ;DECREMENT THE TRY COUNT.
3302 015102 000241 BNE 2# ;LOOP TO TRY AGAIN.
3303 015104 000401 CLC ;CLEAR CARRY, TO INDICATE FIFO NOT PURGED.
3304 015106 000261 BR 60# ;EXIT WITH CARRY CLEAR.
3305 6#: SEC ;SET CARRY, TO INDICATE FIFO PURGED.
3306 015110 004736 60#: PASS ;RESTORE GPRS.
015110 JSR PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
3307 ;CARRY BIT, SET INDICATES FIFO PURGED.
3308 015112 000207 RTS PC

```

```

3310 .SBTTL GLOBAL SUBROUTINE - PUFIFR -
3311 ;*****
3312 ;* - PURGE FIFO REPORT ANY ERRORS FOUND.
3313 ;* THIS ROUTINE REMOVES ALL DATA FROM THE FIFO. ANY BMP CODES THAT ARE
3314 ;* FOUND ARE SAVE ON THE QUEUE TO BE REPORTED LATER IN THE BMP REPORT TEST.
3315 ;* ANY UNEXPECTED DATA (IE ANY NON-STATUS INFORAMTION) THAT ARE FOUND,
3316 ;* ARE REPORTED AS AN ERROR.
3317 ;* IF THE FIFO WILL NOT PURGE AFTER 512 ATTEMPTS, THEN THE CURRENT TEST
3318 ;* THAT CALLED THIS ROUTINE RECEIVES A FAILURE FLAG THAT SHOULD BE USED
3319 ;* TO ABORT THE TEST.
3320 ;*
3321 ;* INPUTS: ERRTBL - ERRTYPE, ERRMSG, ERRNBR ARE SET UP CORRECTLY.
3322 ;* RBUFA- CONTAINS THE ADDRESS OF THE RECEIVER.
3323 ;*
3324 ;* OUTPUTS: CARRY BIT - ABORT TEST FLAG, CLR = ABORT TEST, SET = OK.
3325 ;* ERRBLK - VALUE WILL BE DASTROYED.
3326 ;* BMPCQP - THE BMP CODE QUEUE POINTER MAY BE UPDATED.
3327 ;* THE CONTENTS OF THE BMP CODE QUEUE MAY BE UDATED.
3328 ;*
3329 ;* CALLING SEQUENCE: JSR PC,PUFIFR
3330 ;*
3331 ;* COMMENTS: THIS ROUTINE REPORTS ERRORS WITH NUMBERS INITIAL ERRNBR
3332 ;* THRU TO ERRNBR+2.
3333 ;* THE ERRNBR IS RESTORED TO ITS INITIAL VALUE BEFORE RETURNING.
3334 ;*
3335 ;* SUBORDINATE ROUTINES CALLED: ER1603,ER9001,ER9002,SAVBMP.
3336 ;*****
3337
3338 015114 PUFIFR::SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3339 015114 004567 166774 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3340 015120 016746 166762 ;MOV ERRNBR,-(SP) ;SAVE THE CONTENTS OF THE ERROR NUMBER.
3341 015124 012705 001000 ;MOV #512.,R5 ;SET MAXIMUM READ COUNTER TO 2*FIFO SIZE.
3342 ;*
3343 ;* READ DATA FROM THE FIFO UNTIL DATA VALID IS CLEAR OF READ COUNTER IS ZERO.
3344 ;* REPORT ANY BMP OR UNEXPECTED DATA AS ERRORS.
3345 015130 017702 165110 ;*
3346 015134 100057 2# : MOV BRBUFA,R2 ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
3347 ;* BPL 8# ;EXIT IF DATA VALID CLEAR, IE. FIFO PURGED.
3348 ;*
3349 ;* CHECK IF READ DATA IS STATUS OR UNEXPECTED CHARACTER.
3350 015136 012700 070000 ;*
3351 015142 040200 ;MOV #70000,R0 ;GENERATE A BIT MAP OF CHAR ERROR BITS
3352 015144 001012 ;BIC R2,R0 ;WHICH ARE NOT SET FOR CHAR.
3353 ;* ;SKIP BMP CHECK IF IT IS UNEXPECTED DATA.
3354 ;*
3355 ;* CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
3356 ;* IF IT IS A BMP CODE THEN SAVE IT ON THE QUEUE.
3357 015146 012767 012332 166736 ;*
3358 015154 012700 000300 ;MOV #ER9001,ERRBLK ;SET UP THE CORRECT ERROR REPORTING ROUTINE.
3359 015160 040200 ;MOV #300,R0 ;CHECK IF BMP OR SELFTEST?.
3360 015162 001003 ;BIC R2,R0 ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
3361 015164 004767 000352 ;BNE 4# ;SKIP BMP ERROR REPORT IF MODEM OR SELFTEST?.
3362 015170 000424 ;JSR PC,SAVBMP ;SAVE THE BMP CODE ON THE QUEUE.
3363 ;* ;BRANCH TO CHECK READ COUNT.
3364 ;*
3365 ;* CHECK IF THE READ DATA IS MODEM, SELFTEST OR UNEXPECTED DATA.

```

```

3366 015172 032702 000001      4$:   BIT    #BIT0,R2      ;TEST THE MODEM STATUS INDICATION BIT.
3367 015176 001421              BEQ    6$              ;DO NOT REPORT ANY ERROR IF MODEM STATUS.
3368 015200 012701 011051      MOV    #EM9104,R1     ;PASS THE CORRECT ERROR MESSAGE TO REPORT.
3369 015204 010203              MOV    R2,R3         ;EXTRACT THE LINE NUMBER FROM
3370 015206 000303              SWAB   R3             ; THE READ DATA.
3371 015210 042703 177760      BIC    #177760,R3    ;
3372 015214 006303              ASL    R3             ;FORM LINE NUMBER TIMES 2 FOR ER9002 ROUTINE.
3373 015216 052704 100000      BIS    #BIT15,R4     ;SET THE "NONE" EXPECTED MESSAGE FLAG.
3374 015222 005267 166660      INC    ERRNBR        ;SET ERROR NUMBER TO INTIAL ERRNR+1.
3375 015226 012767 012422 166656  MOV    #ER9002,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
3376                                ;REPORT ERROR "UNEXPECTED DATA FOUND IN FIFO".
3377 015234                                ;
3378 015234 104460                                >>>>> ERROR <<<<<.
3379 015236 005367 166644      DEC    ERRNBR        ;RESTORE ERROR NUMBER TO INTIAL ERRNR.
3380 015242 005305              DEC    R5            ;DECREMENT READ COUNTER.
3381 015244 001331              BNE    2$            ;LOOP TO READ NEXT CHAR FROM FIFO IF COUNT > 0.
3382                                ;
3383                                ;*
3384                                ; THE FIFO WILL NOT CLEAR, REPORT THE ERROR AND INDICATE THAT THE TEST IS TO
3385                                ; BE ABORTED.
3386 015246 062767 000002 166632  ADD    #2,ERRNBR     ;SET ERROR NUMBER TO INTIAL ERRNR+2.
3387 015254 012767 011660 166630  MOV    #ER1603,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
3388 015262 012701 010714      MOV    #EM9017,R1     ;PASS THE MESSAGE TO BE REPORTED.
3389                                ;REPORT THE ERROR "FIFO WILL NOT PURGE, (DATA VALID STUCK SET)"
3390                                ;"?????? TEST ABORTED".
3391 015266                                ;
3392 015266 104460                                >>>>> ERROR <<<<<.
3393 015270 000241                                TRAP   C#ERROR
3394 015272 000401      CLC                    ;INDICATE THE TEST IS TO BE ABORTED.
3395 015274 000261              BR     10$           ;EXIT THIS ROUTINE AND ABORT THE CURRENT TEST.
3396                                ;
3397 015276 012667 166604      8$:   SEC                    ;SET THE CARRY, DO NOT ABORT THE TEST.
3398 015302 004736              10$:  MOV    (SP)+,ERRNBR ;RESTORE INITIAL ERROR NUMBER.
3399                                60$:  PASS                    ;RESTORE GPRS.
3400                                JSR    PC,#(SP)+      ;RETURN TO PREG05 SUBRT.
3401 015304 000207      RTS    PC            ;CARRY BIT, SET INDICATES FIFO PURGED, DO NOT
                                ; ABORT THE TEST.

```

```

3403 .SBTTL GLOBAL SUBROUTINE - READBX -
3404 ;* *****
3405 ;* - READ CHARACTERS FROM THE FIFO AND CHECKS FOR BMPS AND XONS-
3406 ;* THIS SUBROUTINE IS USED IN THE FIHAVL.TST.
3407 ;* IT READS THE SPECIFIED NUMBER OF CHARACTERS FROM THE FIFO AND CHECKS
3408 ;* FOR BMP CODES AND XON CHARACTERS.
3409 ;*
3410 ;* INPUTS: R0 - CONTAINS THE NUMBER OF CHARS TO READ FROM THE FIFO.
3411 ;*
3412 ;* OUTPUTS: R1 - CONTAINS ADDRESS OF ERROR MESSAGE TO BE REPORTED
3413 ;* CLEAR IF NO ERROR FOUND.
3414 ;* CARRY USED TO INDICATE IF FIFO WAS FOUND EMPTY, CARRY CLEAR.
3415 ;*
3416 ;* CALLING SEQUENCE: JSR PC,READ
3417 ;*
3418 ;* COMMENTS:
3419 ;*
3420 ;* SUBORDINATE ROUTINES CALLED: CHKBMP.
3421 ;*
3422 ;* *****
3423 READBX:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
015306 004567 166602 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
015306 005001 ;CLEAR GPR THAT HOLDS THE ADDRESS OF ERRMSG.
3424 015312 005001 ;GET THE ADDRESS OF THE RECEIVER BUFFER REG.
3425 015314 016703 164724 2#: MOV RBUFA,R3 ;READ A CHARACTER FROM THE FIFO.
3426 015320 011302 MOV (R3),R2 ;BRANCH IF FIFO IS EMPTY.
3427 015322 100015 BPL 8#
3428 ;*
3429 ;* CHECK IF THE READ CHARACTER IS A BMP CODE.
3430 ;* IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
3431 ;* ABORT THE TEST.
3432 ;*
3433 015324 004767 176054 JSR PC,CHKBMP ;CHECK IF CHARACTER IS A BMP CODE.
3434 015330 103410 BCS 6# ;BRANCH IF A BMP CODE WAS FOUND.
3435 015332 120227 000021 CMPB R2,#21 ;CHECK IF IT IS AN XON.
3436 015336 001003 BNE 4# ;BRANCH IF NOT AN XON.
3437 015340 012701 006742 MOV #EM5402,R1 ;PASS THE MESSAGE TO BE REPORTED.
3438 015344 000402 BR 6# ;GO EXIT TEST.
3439 015346 005300 4#: DEC R0 ;DECREMENT THE READ COUNT.
3440 015350 001363 BNE 2#
3441 015352 000261 6#: SEC ;SET CARRY TO INDICATE SUCCESS.
3442 015354 000401 BR 60# ;EXIT
3443 015356 000241 8#: CLC ;CLEAR CARRY BIT TO INDICATE FAILURE.
3444
3445 015360 010166 000004 60#: PASS R1 ;RESTORE GPRS,
015360 004736 MOV R1,R1SLOT(SP) ;PUT R1 IN STACK SLOT.
015364 000207 JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
3446 015366 000207 RTS PC

```

```
3448 .SBTTL GLOBAL SUBROUTINE - RESETT -
3449 ;*****
3450 ;* - RESET DEVICE UNDER TEST -
3451 ;* THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
3452 ;* IF RESET DOES NOT SUCCESSFULLY COMPLETE, IE. TIME-OUT OCCURS, THEN
3453 ;* AN ABORT TEST ERROR MESSAGE IS REPORTED.
3454 ;*
3455 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR
3456 ;* TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
3457 ;* ERRRTBL- ERRRTYP,ERNBR,AND ERRMSG SET UP CORRECTLY.
3458 ;*
3459 ;* OUTPUTS: THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
3460 ;* CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
3461 ;* ERRBLK - VALUE MAY BE DESTROYED.
3462 ;* IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
3463 ;* TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
3464 ;*
3465 ;* CALLING SEQUENCE: JSR PC,RESETT
3466 ;*
3467 ;* COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERRNBR
3468 ;* THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.
3469 ;*
3470 ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET.
3471 ;*****
3472
3473 015370 RESETT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
015370 004567 166520 ;R5,PREGOS ;CALL REGISTER SAVE SUBRT.
3474 015374 012702 000040 MOV #BIT05,R2 ;SET BIT MASK OF MASTER RESET BIT.
3475
3476 ;*
3477 ; TEST THE STATE OF THE MASTER RESET BIT IN THE CSR.
3478 ; IF MR IS SET THEN WAIT FOR SELF-TEST TO COMPLETE.
3479 ; IF TIME-OUT OCCURS, REPORT THE ERROR AND PASS-OUT ABORT TEST INDICATOR.
3480
3480 015400 016704 164636 MOV CSRA,R4 ;GET THE ADDRESS OF THE DUT'S CSR.
3481 015404 030214 BIT R2,(R4) ;CHECK STATE OF MASTER RESET BIT.
3482 015406 001406 BEQ 2# ;DON'T DELAY IF MR IS ALREADY CLEAR.
3483 015410 005003 CLR R3 ;SET UP DESIRED STATE OF MASTER RESET BIT.
3484 015412 012701 004704 MOV #2500.,R1 ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
3485 015416 004767 176750 JSR PC,MSLGET ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
3486 015422 103012 BCC 4# ;GO REPORT ERROR IF TIMEOUT OCCURRED.
3487
3488 ;*
3489 ; SET MASTER RESET BIT IN CSR. CLEAR TX AND RX ENABLE BITS, ETC.
3490 ; SKIP THE SELFTEST.
3491 ; TIME-OUT OF 2.5 SECS. JUST IN CASE THE SELF-TEST EXECUTES.
3492
3493 015424 010277 164612 2#: MOV R2,@CSRA ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
3494 015430 004767 000266 JSR PC,SKPSTS ;TRY TO SKIP THE SELFTEST.
3495
3496 ;*
3497 ; SET SELF-TEST TIME-OUT OF 2.5 SECONDS, AND WAIT FOR M.R TO CLEAR.
3498 ; IF TIME-OUT OCCURS, THEN REPORT THE FATAL ERROR AND PASS-OUT THE ABORT
3499 ; TEST INDICATOR.
3500
3500 015434 005003 CLR R3 ;SET UP DESIRED STATE OF MASTER RESET BIT.
3501 015436 012701 004704 MOV #2500.,R1 ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
3502 015442 004767 176724 JSR PC,MSLGET ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
3503 015446 103410 BCS 6# ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
```

```

3504
3505      ; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET,TEST ABORTED".
3506      ; INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
3507
3508 015450 012701 005707      ;
3509 015454 012767 011660 166430 48:   MOV    #EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
3510                                     MOV    #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
3511                                     ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
3512 015462                                     ; "TEST ABORTED"
3513 015462 104460                                     ERROR                                ;
3514 015466 000403                                     ;          >>>> ERROR <<<<<
3515                                     CLC                                TRAP    C:ERROR
3516                                     BR     608          ;INDICATE TEST IS TO BE ABORTED.
3517                                     ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
3518      ;
3519      ; CLEAR TX AND RX INTERRUPT ENABLE STATUS FLAGS IN IESTAT.
3520      ; EXIT WITH CONTINUE TEST INDICATOR SET (IE,CARRY SET).
3521
3522 015470 005067 164574      68:   CLR    IESTAT      ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
3523 015474 000261                                     SEC                                ;INDICATE SUCCESS, CONTINUE TEST.
3524
3525 015476 004736      608:   PASS                                     ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
3526                                     JSR    PC,@(SP)      ;RETURN TO PREG05 SUBRT.
3527                                     ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
3528
3529                                     RTS     PC

```

```

3527 .SBTTL GLOBAL SUBROUTINE - RXIEO -
3528 ;* *****
3529 ;* - RECEIVER INTERRUPT DISABLE -
3530 ;* THIS ROUTINE IS USED TO DISABLE RECEIVER INTERRUPTS IN THE DHV11.
3531 ;*
3532 ;* INPUTS: NONE.
3533 ;*
3534 ;* OUTPUTS: THE RX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
3535 ;* IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3536 ;* ENABLE BITS.
3537 ;*
3538 ;* CALLING SEQUENCE: JSR PC,RXIEO
3539 ;*
3540 ;* COMMENTS: THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3541 ;* THE DUT CSR ARE DESTROYED.
3542 ;*
3543 ;* SUBORDINATE ROUTINES CALLED: NONE.
3544 ;* *****
3545 015502 010046 RXIEO:: MOV RO,-(SP) ;SAVE CONTENTS OF RO ON THE STACK.
3546 015504 104440 GETPRI -(SP) ;SAVE PROCESSOR PRIORITY ON STACK.
3547 015510 010046 SETPRI #PRI07 ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
3548 015516 012700 000340 BIC #137777,IESTAT ;CLEAR RX.INT.ENBL BIT IN IESTAT.
3549 015524 016777 164540 164544 MOV IESTAT,@CSRA ;DISABLE RX INTERRUPTS.
3550 015532 012600 SETPRI (SP)+ ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
3551 015536 012600 MOV (SP)+,RO ;RESTORE RO.
3552 015540 000207 RTS PC

```

3554  
3555  
3556  
3557  
3558  
3559  
3560  
3561  
3562  
3563  
3564  
3565  
3566  
3567  
3568  
3569  
3570  
3571  
3572  
3573  
3574  
3575  
3576  
3577  
3578  
3579  
3580  
3581  
3582  
3583  
3584  
3585  
3586  
3587  
3588  
3589

015542  
015542 004567 166346  
015546 016704 164670  
015552 116724 164510  
015556 005204  
015560 042702 177400  
015564 010224  
015566 020427 002644  
015572 103402  
015574 162704 000004  
015600 010467 164636  
015604  
015604 004736  
015606 000207

```

.SBTTL GLOBAL SUBROUTINE - SAVBMP -
; * *****
; * - SAVE BMP CODES ROUTINE -
; * THIS ROUTINE SAVES THE PARAMETER PASSED IN, ONTO THE BMP CODE QUEUE
; * TOGETHER WITH THE NUMBER OF THE CURRENTLY EXECUTING TEST.
; *
; * INPUTS: R2 - CONTAINS THE BMP CODE THAT IS TO BE PLACED ON THE QUEUE.
; * BMPCQP - CONTAINS ADDRESS OF NEXT LOCATION IN THE BMP QUEUE.
; * BMPCQB - LABEL AT BASE OF THE BMP CODE QUEUE.
; * BMPCQE - LABEL OF NEXT LOCATION AFTER THE END OF THE BMP QUEUE.
; * TSTNUM - CONTAINS THE NUMBER OF THE CURRENT TEST.
; *
; * OUTPUTS: BMPCQP - INCREMENTED BY 4.
; * THE CONTENTS OF THE BMP CODE QUEUE ARE UPDATED.
; *
; * CALLING SEQUENCE: JSR PC,SAVBMP
; *
; * COMMENTS: IF THE OVERFLOW OCCURS THEN THE LAST LOCATION WILL BE
; * OVERWRITTEN BY ANY SUBSEQUENT ATTEMPTS TO UPDATE THE QUEUE.
; *
; * SUBORDINATE ROUTINES CALLED: NONE.
; -- *****

SAVBMP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
                JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
                MOV BMPCQP,R4 ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
                MOV TSTNUM,(R4)+ ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
                INC R4 ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
                BIC #177400,R2 ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
                MOV R2,(R4)+ ;SAVE THE BMP CODE ON THE QUEUE.
                CMP R4,#BMPCQE ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
                BLO 2$ ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
                SUB #4,R4 ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
                MOV R4,BMPCQP ;SAVE THE POINTER.

2$:
60$: PASS ;RESTORE GPRS.
                JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.

                RTS PC

```



3591  
3592  
3593  
3594  
3595  
3596  
3597  
3598  
3599  
3600  
3601  
3602  
3603  
3604  
3605  
3606  
3607  
3608  
3609  
3610  
3611  
3612  
3613  
3614 015610  
015610 004567 166300  
3615 015614 016701 164450  
3616 015620 012702 002644  
3617 015624 012703 000010  
3618 015630 050103  
3619 015632 010177 164404  
3620 015636 017722 164406  
3621 015642 005201  
3622 015644 020103  
3623 015646 002771  
3624  
3625 015650  
015650 004736  
3626 015652 000207

```
.SBTTL GLOBAL SUBROUTINE - SAVMST -
; * *****
; * - SAVE MODEM STATUS ROUTINE -
; * THIS ROUTINE SAVES THE PRESENT CONTENTS OF THE DUT STAT REGISTERS IN
; * THE STAT STORAGE TABLE.
; *
; * INPUTS: CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
; * IESTAT - STATE OF THE DUT CSR INTERRUPT ENABLE BITS.
; * NUMLNS - EQUATED TO THE NUMBER OF LINES ON THE DUT.
; * STATA - CONTAINS THE ADDRESS OF THE DUT STAT REGISTER.
; * STSTB - LABEL AT BASE OF THE STAT STORAGE TABLE.
; *
; * OUTPUTS: STST TABLE - OVERWRITTEN WITH PRESENT STAT CONTENTS.
; * CSR IND.ADR.REG FIELD - DESTROYED.
; *
; * CALLING SEQUENCE: JSR PC,SAVMST
; *
; * COMMENTS: IF THE CONTENTS OF IESTAT CHANGES DURING THIS TEST THE CSR
; * INTERRUPT ENABLE BITS WILL NOT TRACK THE CHANGE.
; *
; * SUBORDINATE ROUTINES CALLED: NONE.
; -- *****
SAVMST:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
;R5,PREG05 ;CALL REGISTER SA/E SUBRT.
MOV IESTAT,R1 ;GET IE STATES FOR UPDATING IND.ADR.REG FIELD.
MOV @STSTB,R2 ;SET UP STAT STORAGE POINTER TO BASE OF TABLE.
MOV @NUMLNS,R3
BIS R1,R3 ;FORM COMPLETION COMPARISON WORD.
2#: MOV R1,@CSRA ;SET UP THE CSR IND.ADR.REG FIELD.
MOV @STATA,(R2)+ ;SAVE CONTENTS OF THIS LINE'S STAT REGISTER.
INC R1 ;SET LINE COUNTER TO NEXT LINE.
CMP R1,R3 ;CHECK FOR ALL LINES DONE.
BLT 2# ;LOOP IF NOT ALL LINES DONE.
60#: PASS ;RESTORE GPRS.
;PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
RTS PC JSR
```

```

3628 .SBTTL GLOBAL SUBROUTINE - SETPAR -
3629 ;** *****
3630 ;* - SET TX AND CONTROL PARAMETERS -
3631 ;* THIS SUROUTINE IS USED IN THE FIHAVL.TST.
3632 ;* IT INITIALISES THE SELECTED LINE TO THE FOLLOWING STATE:
3633 ;* INTERNAL LOOPBACK, IAUTO ENABLED, LPR:38.4K, 8 BITS/CHAR, 2 STOP,
3634 ;* ODD PARITY.
3635 ;*
3636 ;* INPUTS: R1 - CONTAINS NUMBER OF THE LINE TO BE INITIALISED.
3637 ;*
3638 ;* OUTPUTS: LNCTRL AND LPR REGISTERS FOR THE SELECTED LINE ARE DESTROYED.
3639 ;*
3640 ;* CALLING SEQUENCE: JSR PC,SETPAR
3641 ;*
3642 ;* COMMENTS:
3643 ;*
3644 ;* SUBORDINATE ROUTINES CALLED: DELAY,WTWLNC,WTWLPR.
3645 ;-- *****
3646
3647 015654 SETPAR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
015654 004567 166234 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3648 015660 004767 176426 JSR PC,LINBIT ;GET A BIT MAP FOR THIS LINE.
015664 010005 MOV RO,R5 ;COPY THE LINE BIT MAP.
3650 015666 012700 000206 MOV #206,R0 ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
3651 015672 004767 001012 JSR PC,WTWLNC ;INITILAISE THE LINE CONTROL REGISTER.
3652 015676 012700 177670 MOV #177670,R0 ;PASS THE LPR CONTENTS.
3653 015702 004767 001032 JSR PC,WTWLPR ;SET THE LPR CONTENTS TO 38.4K BAUD.
3654 015706 016704 162100 MOV 10.,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
3655 015712 004767 176000 JSR PC,DELAY ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
3656
3657 015716 608: PASS ;RESTORE GPRS.
015716 004736 JSR PC,8(SP)+ ;RETURN TO PREG05 SUBRT.
3658 015720 000207 RTS PC

```

```

3660 .SBTTL GLOBAL SUBROUTINE - SKPSTS -
3661 ;** *****
3662 ;* - SKIP SELFTEST ROUTINE -
3663 ;* THIS SUBROUTINE IS USED TO SKIP THE SELFTEST AFTER A DUT RESET HAS BEEN
3664 ;* INITIATED. IT MUST BE ENTERED IMMEDIATELY AFTER SETTING THE DUT MASTER
3665 ;* RESET ROUTINE OR AFTER THE EXECUTION OF A BUS RESET (BECAUSE OF TIMING
3666 ;* CONSIDERATIONS).
3667 ;*
3668 ;* INPUTS: CSRA - CONTAINS ADDRESS OF THE DUT CSR.
3669 ;* TXBFCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
3670 ;*
3671 ;* OUTPUTS: SKIP SELFTEST CODES ARE WRITTEN TO THE DUT REGISTERS.
3672 ;*
3673 ;* CALLING SEQUENCE: JSR PC,SKPSTS
3674 ;*
3675 ;* COMMENTS:
3676 ;*
3677 ;* SUBORDINATE ROUTINES CALLED: DELAY.
3678 ;-- *****
3679
3680 015722 SKPSTS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
3681 015722 004567 166166 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3682 015726 012704 000012 MOV #10,R4 ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
3683 015732 004767 175760 JSR PC,DELAY ;DELAY FOR 10 MILLI-SECONDS.
3684 ;*
3685 ; WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
3686 ;--
3687 MOV #NUMLNS!BIT05,R1 ;FORM IND.ADR.REG FIELD (PLUS M.R. BIT) WORD.
3688 ;THE ABOVE INCLUSION OF THE M.R. BIT IS NECESSARY BECAUSE OF THE
3689 ; LACK OF A M.R. BIT WRITE LOCK-OUT ON THE DMV-11.
3690 015742 012703 052525 MOV #52525,R3 ;INITIALISE THE SKIP SELF-TEST CODE.
3691 015746 005301 4#: DEC R1 ;SELECT THE NEXT SET OF DEVICE REGISTERS.
3692 015750 016704 164266 MOV CSRA,R4 ;GET THE ADDRESS OF THE CSR OF THE DUT.
3693 015754 010124 6#: MOV R1,(R4)+ ;SELECT A BANK OF DUT REGISTERS.
3694 015756 010324 MOV R3,(R4)+ ;WRITE THE CODE TO A DUT REGISTER.
3695 015760 020467 164274 CMP R4,TXBFCA ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
3696 015764 103774 BLO 6# ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
3697 015766 032701 000017 BIT #17,R1 ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
3698 015772 001365 BNE 4# ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.
3699 015774 60#: PASS ;RESTORE GPRS.
3700 015774 004736 JSR PC,B(SP)+ ;RETURN TO PREG05 SUBRT.
3700 015776 000207 RTS PC

```

```

3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723 016000
      016000 004567 166110
3724 016004 012701 016022
3725 016010 012767 011660 166074
3726 016016
      016016 104460
      016020 000432
3727 016022 040 116 117
      016025 116 055 122
      016030 105 114 101
      016033 124 105 104
      016036 040 124 105
      016041 123 124 040
      016044 105 122 122
      016047 117 122 040
      016052 106 117 125
      016055 116 104 040
      016060 104 125 122
      016063 111 116 107
      016066 040 124 105
      016071 123 124 040
      016074 105 130 105
      016077 103 125 124
      016102 111 117 116
      016105 000
3729
3730 016106
      016106 004736
3731 016110 000207

```

```

.SBTTL GLOBAL SUBROUTINE - TSABRT -
; * *****
; * - TEST ABORT ROUTINE -
; * THIS SUBROUTINE IS USED WHEN A NON-TEST RELATED ERROR HAS BEEN FOUND
; * DURING THE EXECUTION OF THE CURRENT TEST.
; * IT IS USED TO INFORM THE OPERATOR THAT THE CURRENT TEST HAS BEEN
; * ABORTED.
; *
; * INPUTS: ERRMSG - CONTAINS THE NAME OF THE CURRENT TEST.
; *          ERRNBR - CONTAINS THE CORRECT ERROR NUMBER.
; *          THE REMAINDER OF THE ERRTABL IS CORRECTLY INITIALISED.
; *
; * OUTPUTS: MESSAGES ARE REPORTED TO THE OPERATOR.
; *
; * CALLING SEQUENCE: JSR PC,TSABRT
; *
; * COMMENTS:
; *
; * SUBORDINATE ROUTINES CALLED: ER1603.
; - - *****
TSABRT:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
              JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
              MOV #2,R1 ;PASS ADDRESS OF FIRST MESSAGE TO BE REPORTED.
              MOV #ER1603,ERRBLK ;SET-UP THE ERROR REPORTING ROUTINE.
              ERROR ; >>>> ERROR <<<<<.
                                     TRAP C#ERROR
              BR 60$ ;
2$: .ASCIZ / NON-RELATED TEST ERROR FOUND DURING TEST EXECUTION/

.EVEN
60$: PASS ;RESTORE GPRS.
              JSR PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
              RTS PC

```

3733  
3734  
3735  
3736  
3737  
3738  
3739  
3740  
3741  
3742  
3743  
3744  
3745  
3746  
3747  
3748  
3749  
3750  
3751  
3752  
3753  
3754  
3755  
3756  
3757  
3758

016112  
016112 004567 165776  
016116 010003  
016120 012702 002704  
016124 004767 175626  
016130  
016130 004736  
016132 000207

```

.SBTTL GLOBAL SUBROUTINE - TXDATP -
; * *****
; * - TRANSMIT DATA PATTERN -
; * THIS SUBROUTINE IS USED IN THE FIHAVL.TST.
; * IT TRANSMITS A SPECIFIED NUMBER OF DATA BYTES ON THE SPECIFIED LINE.
; *
; * INPUTS: R0 - CONTAINS THE NUMBER OF DATA BYTES TO TX.
; * R1 - CONTAINS LINE NUMB ON WHICH TRANSMISSION IS TO TAKE PLACE.
; * BUFBAS TO BUF MID CONTAINS A 256 BYTE DATA PATTERN.
; *
; * OUTPUTS: DATA IS SENT OUT ON THE SPECIFIED LINE.
; * CARRY SET = TX SUCCESSFUL.
; *
; * CALLING SEQUENCE: TXDATP
; *
; * COMMENTS:
; *
; * SUBORDINATE ROUTINES CALLED: DODMA.
; * - - - - -
TXDATP:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
; R5,PREG05 ;CALL REGISTER SAVE SUBRT.
MOV R0,R3 JSR ;PASS THE NUMBER OF CHARS TO TX.
MOV #BUFBAS,R2 ;PASS THE START OF THE DATA PATTERN TO TX.
JSR PC,DODMA ;TRANSMIT THE DATA PATTERN.
60: PASS ;RESTORE GPRS.
; PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
RTS PC JSR

```

GLOBAL SUBROUTINE

- TXDSBL -

```

3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783 016134
      016134 004567 165754
3784 016140 010500
3785 016142 012701 000001
3786 016146 016702 164104
3787 016152 005202
3788 016154 012703 000010
3789 016160 016704 164104
3790 016164 005005
3791
3792
3793
3794 016166 010477 164050
3795 016172 105712
3796 016174 100001
3797 016176 050105
3798
3799
3800
3801
3802 016200 030100
3803 016202 001402
3804 016204 142712 000200
3805 016210 005204
3806 016212 006301
3807 016214 005303
3808 016216 001363
3809
3810 016220
      016220 010566 000014
      016224 004736
3811
3812 016226 000207

```

```

.SBTTL GLOBAL SUBROUTINE - TXDSBL -
; * *****
; * - TRANSMITTER DISABLE -
; * THIS SUBROUTINE IS USED TO DISABLE TRANSMISSION ON SELECTED LINES BY,
; * CLEARING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
; *
; * INPUTS: R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO CLEAR TX.ENABLE.
; * CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
; * IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
; * NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
; * TXAD2A - CONTAINS THE ADDRESS OF THE TBUFAD2 REGISTER.
; *
; * OUTPUTS: R5 - BIT'S SET INDICATE THE INITIAL STATE OF ALL TX.ENABLE BITS.
; * TBUFAD2 - THE STATE OF THE TX.ENABLE BIT MAY BE ALTERED.
; * THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
; *
; * CALLING SEQUENCE: JSR PC,TXDSBL
; *
; * COMMENTS:
; *
; * SUBORDINATE ROUTINES CALLED: NONE.
; * -- *****
TXDSBL:: SAVE
; SAVE CONTENTS OF GPRS R0 THRU R5.
; R5,PREG05 CALL REGISTER SAVE SUBRT.
; COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
; INITIALIZE THE SELECTED LINE BIT MASK.
; GET THE ADDRESS OF THE TBUFAD2 REGISTER.
; GET THE ADDRESS OF THE MSBYTE OF TBUFAD2 REG.
; GET MAXIMUM LINE NUMBER PLUS ONE.
; GET THE STATES OF THE INT ENABLE BITS.
; LOG POSSIBLE TX DISABLED ON ALL LINES.
; *
; * SELECT EVERY LINE IN TURN, AND LOG THE STATE OF EACH TX.ENABLE BIT.
; *
; *
; * WRITE TO DUT CSR TO SELECT LINE REGISTERS.
; * CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
; * SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
; * LOG TX ENABLE BIT SET FOR SELECTED LINE.
; *
; * CLEAR TX.ENABLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX DISABLE
; * LINE BIT MAP.
; *
; * CHECK STATE OF DISABLE LINE BIT MAP.
; * BRANCH IF THIS LINE TO REMAIN UNALTERED.
; * CLEAR TX.ENABLE BIT ON SELECTED LINE.
; * PREPARE TO SELECT REGISTERS FOR NEXT LINE.
; * SHIFT BIT MAP FOR NEXT LINE.
; * DECREMENT LINE NUMBER.
; * LOOP TO CHECK NEXT LINE.
; *
; * RESTORE GPRS,EXCEPT
; * R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
; * PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
; * R5 - PREVIOUS STATES OF ALL TX.ENABLE BITS.
; *
RTS PC

```

GLOBAL SUBROUTINE

- TXENBL -

```

3814 .SBTTL GLOBAL SUBROUTINE - TXENBL -
3815 ;* *****
3816 ;* - TRANSMITTER ENABLE -
3817 ;* THIS SUBROUTINE IS USED TO ENABLE TRANSMISSION ON SELECTED LINES BY
3818 ;* SETTING THE ASSOCIATED TX.ENABLE BIT ON THE DUT.
3819 ;*
3820 ;* INPUTS: R5 - BIT'S SET CORRESPOND TO LINES ON WHICH TO SET TX.ENABLE.
3821 ;* CSRA - CONTAINS THE ADDRESS OF THE DUT CSR.
3822 ;* IESTAT - CONTAINS THE STATE OF TXIE AND RXIE BITS IN THE CSR.
3823 ;* NUMLNS - EQUATED TO BE THE MAXIMUM NUMBER OF LINES AVAILABLE.
3824 ;* TXAD2A - CONTAINS THE ADDRESS OF THE TBUFAD2 REGISTER.
3825 ;*
3826 ;* OUTPUTS: R5 - BIT'S SET INDICATE PREVIOUSLY DISABLED LINES.
3827 ;* TBUFAD2 - THE STATE OF THE TX.ENABLE BIT MAY BE ALTERED.
3828 ;* THE CONTENTS OF THE IND.ADD.REG FIELD IN THE CSR ARE DESTROYED.
3829 ;*
3830 ;* CALLING SEQUENCE: JSR PC,TXENBL
3831 ;*
3832 ;* COMMENTS:
3833 ;*
3834 ;* SUBORDINATE ROUTINES CALLED: NONE.
3835 ;*
3836 ;* *****
3837 TXENBL:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
016230 004567 165660 JSR R5,PREGOS ;CALL REGISTER SAVE SUBRT.
016234 010500 MOV R5,R0 ;COPY BIT MAP OF LINES TO ENABLE.
3838 016234 010500 MOV #BIT0,R1 ;INITIALIZE THE SELECTED LINE BIT MASK.
3839 016236 012701 000001 MOV TXAD2A,R2 ;GET THE ADDRESS OF THE TBUFAD2 REGISTER.
3840 016242 016702 164010 INC R2 ;GET THE ADDRESS OF THE MSBYTE OF TBUFAD2 REG.
3841 016246 005202 MOV #NUMLNS,R3 ;GET MAXIMUM LINE NUMBER.
3842 016250 012703 000010 MOV IESTAT,R4 ;GET THE STATES OF THE INT ENABLE BITS.
3843 016254 016704 164010 CLR R5 ;CLEAR TX.ENABLE BIT LOG OF DISABLED LINES.
3844 016260 005005
3845 ;*
3846 ; SELECT EVERY LINE IN TURN,AND LOG ANY TX.ENABLE BIT THAT IS CLEAR.
3847 ;
3848 016262 010477 163754 2: MOV R4,@CSRA ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
3849 016266 105712 TSTB (R2) ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
3850 016270 100401 BMI 4: ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
3851 016272 050105 BIS R1,R5 ;LOG TX ENABLE BIT CLEAR FOR SELECTED LINE.
3852 ;*
3853 ; SET TX.ENABLE ON LINES THAT HAVE A CORRESPONDING BIT SET IN THE TX ENABLE
3854 ; LINE BIT MAP.
3855 ;
3856 016274 030100 4: BIT R1,R0 ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
3857 016276 001402 BEQ 6: ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
3858 016300 152712 000200 BISB #BIT7,(R2) ;ENABLE TRANSMISSION ON SELECTED LINE.
3859 016304 005204 6: INC R4 ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
3860 016306 006301 ASL R1 ;SHIFT BIT MAP FOR NEXT LINE.
3861 016310 005303 DEC R3 ;DECREMENT LINE NUMBER.
3862 016312 001363 BNE 2: ;LOOP TO CHECK NEXT LINE.
3863 ;
3864 016314 010566 000014 60: PASS R5 ;RESTORE GPRS,EXCEPT
016314 010566 000014 MOV R5,R5SLOT(SP) ;PUT R5 IN STACK SLOT.
016320 004736 JSR PC,@(SP)+ ;RETURN TO PREGOS SUBRT.
3865 ;R5 - LINE BIT MAP CORRESPONDING TO THE
3866 ; PREVIOUS LINES THAT WERE DISABLED.
3867 016322 000207 RTS PC

```

GLOBAL SUBROUTINE

- TXIE0 -

```

3869 .SBTTL GLOBAL SUBROUTINE - TXIE0 -
3870 ;** *****
3871 ;* - TRANSMITTER INTERRUPT DISABLE -
3872 ;* THIS ROUTINE IS USED TO DISABLE TRANSMITTER INTERRUPTS IN THE DHV11.
3873 ;*
3874 ;* INPUTS: NONE.
3875 ;*
3876 ;* OUTPUTS: THE TX.INT.ENBL BIT IS CLEARED IN THE DUT CSR.
3877 ;* IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
3878 ;* ENABLE BITS.
3879 ;*
3880 ;* CALLING SEQUENCE: JSR PC,TXIE0
3881 ;*
3882 ;* COMMENTS: THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
3883 ;* THE DUT CSR ARE DESTROYED.
3884 ;*
3885 ;* SUBORDINATE ROUTINES CALLED: NONE.
3886 ;-- *****
3887 016324 010046 TXIE0:: MOV RO,-(SP) ;SAVE CONTENTS OF RO ON THE STACK.
3888 016326 104440 GETPRI -(SP) ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
016330 010046 TRAP C#GPRI
3889 016332 SETPRI #PRI07 ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
016332 012700 000340 MOV RO,-(SP)
016336 104441 TRAP C#SPRI
3890 016340 042767 177677 163722 BIC #177677,IESTAT ;CLEAR TX.INT.ENBL BIT IN IESTAT.
3891 016346 016777 163716 163666 MOV IESTAT,@CSRA ;DISABLE TX INTERRUPTS.
3892 016354 SETPRI (SP)+ ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
016354 012600 MOV (SP)+,RO
016356 104441 TRAP C#SPRI
3893 016360 012600 MOV (SP)+,RO
3894 016362 000207 RTS PC ;RESTORE RO.

```



GLOBAL SUBROUTINE

- UNSDIV -

```

3896 .SBTTL GLOBAL SUBROUTINE - UNSDIV -
3897 ;* *****
3898 ;* - UNSIGNED DIVIDE ROUTINE -
3899 ;* THIS SUBROUTINE IS USED TO DIVIDE A 32 BIT UNSIGNED DIVIDEND BY A
3900 ;* 16 BIT UNSIGNED DIVISOR GIVING A 16 BIT QUOTIENT. ALL NUMBERS ARE
3901 ;* CONSIDERED TO BE UNSIGNED. A SUCCESS FLAG IS NOT SET ON RETURN IF
3902 ;* THE QUOTIENT WAS TOO BIG TO BE CONTAINED IN 16 BITS.
3903 ;*
3904 ;* INPUTS: R1 - THE DIVISOR, UNSIGNED, 16 BITS.
3905 ;* R2 - MOST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
3906 ;* R3 - LEAST SIGNIFICANT WORD OF THE DIVIDEND, UNSIGNED, 16 BITS.
3907 ;*
3908 ;* OUTPUTS: R1 - QUOTIENT, UNSIGNED, 16 BITS (177777 IF OVERFLOW).
3909 ;* CARRY - SUCCESS FLAG, SET IF COMPLETE QUOTIENT FITS IN 16 BITS.
3910 ;*
3911 ;* CALLING SEQUENCE: JSR PC,UNSDIV
3912 ;*
3913 ;* COMMENTS: IF THE DIVISOR IS 0 THE QUOTIENT IS RETURNED AS ALL ONES
3914 ;* (177777) AND THE CARRY IS CLEAR REGARDLESS OF THE DIVIDEND.
3915 ;*
3916 ;* SUBORDINATE ROUTINES CALLED: NONE.
3917 ;*
3918 ;*
3919 016364 UNSDIV:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
016364 004567 165524 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
3920 ;*
3921 ; CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
3922 ; -
3923 016370 010204 MOV R2,R4 ;GET MSW OF DIVIDEND FOR SUBTRACT.
3924 016372 160104 SUB R1,R4 ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
3925 016374 103403 BCS 2# ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
3926 016376 012701 177777 MOV #1,R1 ;SET QUOTIENT TO ALL ONES (177777).
3927 016402 000442 BR 60# ;EXIT WITH CARRY CLEAR.
3928 ;*
3929 ; SET UP COUNTERS AND VARIOUS WORKING GPRS.
3930 ; -
3931 016404 005004 2# CLR R4 ;CLEAR THE LSW OF THE DIVISOR.
3932 016406 000241 CLC ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
3933 016410 006001 ROR R1 ; DIVISOR BY
3934 016412 006004 ROR R4 ; 2(UNSIGNED)
3935 016414 012700 000020 MOV #16.,R0 ;SET UP INITIAL SHIFT COUNT TO 16.
3936 ;*
3937 ; THE SUBTRACT AND SHIFT LOOP.
3938 ; -
3939 016420 010246 4# MOV R2,-(SP) ;SAVE MSWORD OF DIVIDEND.
3940 016422 010346 MOV R3,-(SP) ;SAVE LSWORD OF DIVIDEND.
3941 016424 160403 SUB R4,R3 ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
3942 016426 005602 SBC R2 ;MSWORD DIVIDEND - BORROW
3943 016430 103402 BCS 6# ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
3944 016432 160102 SUB R1,R2 ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
3945 016434 103003 BCC 8# ;IF NO BORROW, IT WENT, CARRY IS CLEAR.
3946 ;*
3947 ; IT DIDN'T GO, SO WE SHIFT A 1 INTO THE QUOTIENT (COMPLEMENTED LATER).
3948 ; CARRY IS SET.
3949 ; -
3950 016436 012603 6# MOV (SP)+,R3 ;RESTORE LSWORD OF DIVIDEND.
3951 016440 012602 MOV (SP)+,R2 ;RESTORE MSWORD OF DIVIDEND.

```

GLOBAL SUBROUTINE

- UNSDIV -

```

3952 016442 000401          BR      10$          ;GOTO SHIFT 1 INTO THE QUOTIENT.
3953                          ;+
3954                          ; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
3955                          ; COMPLEMENTED LATER).  CARRY IS CLEAR.
3956                          ;-
3957 016444 012626      8$:  MOV      (SP)+,(SP)+      ;POP THE SAVED DIVIDEND OFF OF THE STACK.
3958                          ;+
3959                          ; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.
3960                          ;-
3961 016446 006105      10$:  ROL      R5          ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
3962 016450 000241          CLC          ;DIVIDE THE
3963 016452 006001          ROR      R1          ; DEVISOR BY
3964 016454 006004          ROR      R4          ; 2 (UNSIGNED).
3965 016456 005300          DEC      R0          ;COUNT THIS SHIFT AND SUBTRACT.
3966 016460 001357          BNE     4$          ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
3967 016462 005105          COM      R5          ;GET QUOTIENT FROM INVERTED QUOTIENT.
3968                          ;+
3969                          ; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.
3970                          ;-
3971 016464 000241          CLC          ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
3972 016466 006103          ROL      R3          ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
3973 016470 103402          BCS     12$         ;IF CARRY FROM SHIFT, ROUND UP.
3974 016472 160403          SUB     R4,R3       ;SUBTRACT DIVISOR FROM DIVIDEND.
3975 016474 103403          BCS     14$         ;IF BORROW, DON'T ROUND UP.
3976                          ;+
3977                          ; ROUND UP, EXTRA SUBTRACT WENT.
3978                          ;-
3979 016476 005205      12$:  INC      R5          ;INCREMENT THE QUOTIENT BY ONE.
3980 016500 001001          BNE     14$         ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
3981 016502 005305          DEC     R5          ;DON'T LET ROUNDING CAUSE OVERFLOW.
3982                          ;+
3983                          ; ALL DONE, PASS QUOTIENT AND EXIT.
3984                          ;-
3985 016504 010501      14$:  MOV     R5,R1       ;PASS QUOTIENT BACK IN R1.
3986 016506 000261          SEC          ;INDICATE NO OVERFLOW.
3987                          ;+
3988 016510 010166 000004  60$:  PASS     R1          ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
                                MOV     R1,R1SLOT(SP) ;PUT R1 IN STACK SLOT.
                                JSR     PC,@(SP)+    ;RETURN TO PREG05 SUBRT.
3989                          ;R1 - 16 BIT, UNSIGNED QUOTIENT.
3990 016516 000207          RTS     PC          ;CARRY - SET INDICATES NO OVERFLOW (SUCCESS).

```

GLOBAL SUBROUTINE

- WAIBIC -

3992  
3993  
3994  
3995  
3996  
3997  
3998  
3999  
4000  
4001  
4002  
4003  
4004  
4005  
4006  
4007  
4008  
4009  
4010  
4011  
4012  
4013  
4014  
4015  
4016  
4017  
4018  
4019  
4020  
4021  
4022  
4023  
4024  
4025  
4026  
4027  
4028  
4029  
4030  
4031  
4032  
4033  
4034  
4035  
4036

016520  
016520 004567 165370  
016524 010204  
016526 010102  
016530 042701 170000  
016534 042702 007777  
016540 000302  
016542 006202  
016544 006202  
016546 006202  
016550 016202 002370  
016554 005003  
016556 004767 175610  
016562 010002  
016564 010266 000006  
016570 004736  
016572 000207

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

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

GLOBAL SUBROUTINE

- WAIBIS -

```

4038 .SBTTL GLOBAL SUBROUTINE - WAIBIS -
4039 ;* *****
4040 ;* - WAIT FOR BIT SET ROUTINE -
4041 ;* THIS SUBROUTINE WAITS FOR THE SPECIFIED BIT TO BECOME SET. IF THE
4042 ;* SPECIFIED BIT GOES TO A SET STATE WITHIN THE SPECIFIED TIME-OUT
4043 ;* PERIOD A SUCCESS INDICATION IS RETURNED BY THIS ROUTINE.
4044 ;* THE LAST VALUE WHICH IS READ LOOKING FOR THE CONDITION IS RETURNED TO
4045 ;* ALLOW THE USE OF THIS ROUTINE TO LOOK FOR DESTRUCTIVE READ CONDITIONS.
4046 ;*
4047 ;* INPUTS: R1 - TIME-OUT VALUE AND BIT NUMBER INDICATION:
4048 ;* BITS 15 THRU 12 - NUMBER OF BIT TO TEST (RANGE 0 THRU 15).
4049 ;* BITS 11 THRU 0 - TIME-OUT VALUE IN MILLI-SECONDS (4095 MAX).
4050 ;* R2 - ADDRESS OF WORD CONTAINING THE BIT TO TEST.
4051 ;* MSLCNT.
4052 ;*
4053 ;* OUTPUTS: R2 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
4054 ;* CARRY - SUCCESS FLAG (CARRY SET IF BIT SET BEFORE TIME-OUT).
4055 ;*
4056 ;* CALLING SEQUENCE: MOV #130040,R1 ;PASS BIT 11 (13 OCTAL) AND
4057 ;* ; 32 (40 OCTAL) MS DELAY.
4058 ;* MOV #LABEL,R2 ;TEST BIT IN WORD AT "LABEL".
4059 ;* JSR PC,WAIBIS ;WAIT 32 MS FOR BIT 11 TO SET.
4060 ;*
4061 ;* COMMENTS:
4062 ;*
4063 ;* SUBORDINATE ROUTINES CALLED: MSLGET.
4064 ;* - - - - -
4065
4066 016574 WAIBIS:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
016574 004567 165314 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4067 016600 010204 MOV R2,R4 ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
4068 016602 010102 MOV R1,R2
4069 016604 042701 170000 BIC #170000,R1 ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
4070 016610 042702 007777 BIC #7777,R2 ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
4071 016614 000302 SWAB R2 ;PUT LINE NUMBER FIELD IN LSBYTE.
4072 016616 006202 ASR R2 ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
4073 016620 006202 ASR R2 ; POSITION TO USE IT AS A WORD TABLE OFFSET
4074 016622 006202 ASR R2 ; FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
4075 016624 016202 002370 MOV BITTBL(R2),R2 ;GET BIT MAP OF LINE TO TEST FROM TABLE.
4076 016630 010203 MOV R2,R3 ;INDICATE THAT THE BIT SHOULD BE SET.
4077 016632 004767 175534 JSR PC,MSLGET ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
4078 ; CARRY IS CORRECT UPON MSLGET RETURN.
4079 016636 010002 MOV R0,R2 ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
4080 016640 601: PASS R2 ;RESTORE GPRS, EXCEPT THE FOLLOWING:
016640 010266 000006 MOV R2,R2SLOT(SP) ;PUT R2 IN STACK SLOT.
016644 004736 JSR PC,B(SP) ;RETURN TO PREG05 SUBRT.
4081 ; R2 - LAST VALUE READ LOOKING FOR CONDITION.
4082 016646 000207 RTS PC ; CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).

```

GLOBAL SUBROUTINE

- WAITTX -

```

4084 .SBTTL GLOBAL SUBROUTINE - WAITTX -
4085 ;** *****
4086 ;* - WAIT FOR TX TO FINISH -
4087 ;* THIS SUBROUTINE IS USED IN THE FIAVL.TST.
4088 ;* IT WAITS FOR TRANSMISSION TO COMPLETE IE TX_ACTION. THEN DELAYS
4089 ;* FOR 5 MILLISECONDS TO ALLOW TIME FOR THE LAST CHARACTER TO GET INTO
4090 ;* THE FIFO.
4091 ;*
4092 ;* INPUTS: CSRA - CONTAINS THE ADDRESS OF THE CSR.
4093 ;*
4094 ;* OUTPUTS: CARRY - SET INDICATES SUCCESS.
4095 ;*
4096 ;* CALLING SEQUENCE: JSR PC,WAITTX
4097 ;*
4098 ;* COMMENTS:
4099 ;*
4100 ;* SUBORDINATE ROUTINES CALLED: DELAY,WAIBIS.
4101 ;-- *****
4102
4103 016650 WAITTX:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
4104 016650 004567 165240 ;R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4105 016654 012701 170536 ;MOV #170536,R1 ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
4106 016660 016702 163356 ;MOV CSRA,R2 ;PASS THE ADDRESS OF THE CSR.
4107 016664 004767 177704 ;JSR PC,WAIBIS ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
4108 016670 103005 ;BCC 60H ;BRANCH IF FIFO EMPTY, ABORT THE TEST.
4109 016672 012704 000005 ;MOV #5,R4 ;PASS DELAY OF 5 MILLI SECS.
4110 016676 004767 175014 ;JSR PC,DELAY ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
4111 ;SEC ;SET CARRY TO INDICATE SUCCESS.
4112 016704 60H: PASS ;RESTORE GPRS.
4113 016704 004736 ;PC,@(SP)+ ;RETURN TO PREG05 SUBRT.
4114 016706 000207 ;RTS PC ;PASS THE CARRY BIT, SET INDICATES SUCCESS.

```

GLOBAL SUBROUTINE

- WTWLNC -

```

4116 .SBTTL GLOBAL SUBROUTINE - WTWLNC -
4117 ;* *****
4118 ;* - LINE CONTROL REGISTER SETUP ROUTINE -
4119 ;* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
4120 ;* CONTROL REGISTERS (LNCTRL) TO THE SPECIFIED STATE. ONLY THE LNCTRLS
4121 ;* FOR THE SPECIFIED LINES ARE ALTERED.
4122 ;*
4123 ;* INPUTS: R0 - NEW LINE PARAMETERS.
4124 ;* R5 - BIT MAP OF LINES TO BE ALTERED.
4125 ;* CSRA - CONTAINS ADDRESS OF THE DUT CSR.
4126 ;* IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
4127 ;* ENABLE BITS IN THE CSR.
4128 ;* LNCTRA - CONTAINS ADDRESS OF THE DUT LNCTRL REGISTERS.
4129 ;*
4130 ;* OUTPUTS: LNCTRL - SPECIFIED DUT LINE CONTROL REGISTERS ARE ALTERED.
4131 ;*
4132 ;* CALLING SEQUENCE: JSR PC,WTWLNC
4133 ;*
4134 ;* COMMENTS:
4135 ;*
4136 ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
4137 ;*
4138 ;* *****
4139 016710 WTWLNC:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
016710 004567 165200 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4140 ;*
4141 ;* SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
4142 ;*
4143 016714 016701 163332 MOV LNCTRA,R1 ;SET UP THE REGISTER ADDRESS PARAMETER.
4144 016720 010002 MOV R0,R2 ;SET UP THE DESIRED REGISTER CONTENTS.
4145 016722 010503 MOV R5,R3 ;SET UP THE BIT MAP OF LINES TO ALTER.
4146 016724 012704 177777 MOV #-1,R4 ;SELECT ALL REGISTER BITS TO BE ALTERED.
4147 ;*
4148 ;* CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
4149 ;*
4150 016730 004767 174042 JSR PC,ALTFLD ;ALTER THE REGISTER CONTENTS.
4151 ;*
4152 016734 601: PASS ;RESTORE GPRS.
016734 004736 JSR PC,0(SP) ;RETURN TO PREG05 SUBRT.
4153 016736 000207 RTS PC

```

GLOBAL SUBROUTINE

- WTWLPR -

```

4155 .SBTTL GLOBAL SUBROUTINE - WTWLPR -
4156 ;* *****
4157 ;* - LINE PARAMETER REGISTER SETUP ROUTINE -
4158 ;* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (DUT) LINE
4159 ;* PARAMETER REGISTERS (LPR) TO THE SPECIFIED STATE. ONLY THE LPRS FOR
4160 ;* THE SPECIFIED LINES ARE ALTERED.
4161 ;*
4162 ;* INPUTS: R0 - NEW LINE PARAMETERS.
4163 ;* R5 - BIT MAP OF LINES TO BE ALTERED.
4164 ;* CSRA - CONTAINS ADDRESS OF THE DUT CSR.
4165 ;* IESTAT - CONTAINS THE CURRENT STATE OF THE TX AND RX INTERRUPT
4166 ;* ENABLE BITS IN THE CSR.
4167 ;* LPRA - CONTAINS ADDRESS OF THE DUT LPR.
4168 ;*
4169 ;* OUTPUTS: LPR - SPECIFIED DUT LINE PARAMETER REGISTERS ARE ALTERED.
4170 ;*
4171 ;* CALLING SEQUENCE: JSR PC,WTWLPR
4172 ;*
4173 ;* COMMENTS:
4174 ;*
4175 ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
4176 ;* - *****
4177
4178 016740 WTWLPR:: SAVE ;SAVE CONTENTS OF GPRS R0 THRU R5.
016740 004567 165150 JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.
4179 ;*
4180 ; SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
4181 ; -
4182 016744 016701 163276 MOV LPRA,R1 ;SET UP THE REGISTER ADDRESS PARAMETER.
4183 016750 010002 MOV R0,R2 ;SET UP THE DESIRED REGISTER CONTENTS.
4184 016752 010503 MOV R5,R3 ;SET UP THE BIT MAP OF LINES TO ALTER.
4185 016754 012704 177777 MOV #-1,R4 ;SELECT ALL REGISTER BITS TO BE ALTERED.
4186 ;*
4187 ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
4188 ; -
4189 016760 004767 174012 JSR PC,ALTFLD ;ALTER THE REGISTER CONTENTS.
4190 ;*
4191 016764 004736 600: PASS ;RESTORE GPRS.
016764 004736 JSR PC,0(SP) ;RETURN TO PREG05 SUBRT.
4192 016766 000207 RTS PC
    
```

INTERRUPT SERVICE ROUTINE

- CLKINT -

4194  
4195  
4196  
4197  
4198  
4199  
4200  
4201  
4202  
4203  
4204  
4205  
4206  
4207  
4208  
4209  
4210  
4211  
4212  
4213  
4214  
4215  
4216  
4217 016770 005767 163332  
4218 016774 001402  
4219 016776 005367 163324  
4220 017002 005767 163322  
4221 017006 001402  
4222 017010 005367 163314  
4223 017014 005367 163312  
4224 017020 001006  
4225 017022 016767 163306 163302  
4226 017030 010046  
4227 017032  
017032 104422  
4228 017034 012600  
4229 017036 000002

```

.SBTTL INTERRUPT SERVICE ROUTINE - CLKINT -
; ** *****
; * THIS ROUTINE IS EXECUTED CLKHRZ TIMES PER SECOND. IT DECREASES THE
; * TWO TIMER COUNTERS DOWN TO ZERO.
; *
; * INPUTS: TIMER1 - TIMER COUNTER #1.
; * TIMER2 - TIMER COUNTER #2.
; * TIMER3 - TIMER COUNTER FOR CALL OF BREAK MACRO.
; *
; * OUTPUTS: THE 2 TIMER COUNTERS ARE DECREMENTED IF THEY ARE NOT ZERO.
; *
; * CALLING SEQUENCE: PUT #CLKINT IN THE CLOCK INTERRUPT VECTOR SLOT.
; * PUT THE DESIRED TIME PERIOD (SECONDS TIMES CLKHRZ) IN
; * EITHER TIMER1 OR TIMER2 AND POLL THE RESPECTIVE TIMER
; * COUNTER TO DETECT ITS GOING TO 0 ON TIME-OUT.
; *
; * COMMENTS: THE 2 COUNTERS WILL NOT WRAPAROUND BUT WILL STOP AT 0. THIS
; * ALLOWS THE DETECTION OF A TIME-OUT ANY TIME AFTER THE TIME-OUT
; * HAS OCCURRED UNTIL THE TIMER COUNTER IS SET TO ANOTHER VALUE.
; *
; * SUBORDINATE ROUTINES CALLED: NONE.
; -- *****
CLKINT:: TST TIMER1 ;CHECK FOR TIMER1 AT ZERO.
        BEQ 2$ ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
        DEC TIMER1 ;DECREMENT TIME COUNT.
2$: TST TIMER2 ;CHECK FOR TIMER2 AT ZERO.
    BEQ 4$ ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
    DEC TIMER2 ;DECREMENT TIME COUNT.
4$: DEC TIMER3 ;DECREMENT THE BREAK COUNT.
    BNE 60$ ;EXIT IF NOT TIME TO CALL BREAK.
    MOV BCOUNT,TIMER3 ;SET UP TIME TILL NEXT BREAK.
    MOV RO,-(SP) ;SAVE CONTENTS OF RO FROM BREAK MACRO.
    BREAK ;CHECK FOR OPERATOR CONTROL/C. TRAP C#BRK
60$: MOV (SP)+,RO ;RESTORE CONTENTS OF RO.
     RTI

```



GLOBAL TRAP SERVICE ROUTINE - TP4RTN -

```

4231 .SBTTL GLOBAL TRAP SERVICE ROUTINE - TP4RTN -
4232 ;*****
4233 ;* BUS TIME-OUT TRAP (004 TRAP) SERVICE ROUTINE -
4234 ;* THIS ROUTINE IS USED DURING THE DEVICE REGISTER ADDRESS ACCESS TEST.
4235 ;* IT DETERMINES IF THE 004 TRAP WAS CAUSED BY AN "EXPECTED" ERROR OR
4236 ;* NOT BY EXAMINING THE RETURN PC VALUE ON THE STACK. IF THE TRAP IS
4237 ;* UNEXPECTED, THIS ROUTINE JUMPS TO THE NORMAL DIAGNOSTIC SUPERVISOR
4238 ;* 004 TRAP HANDLING ROUTINE.
4239 ;*
4240 ;* INPUTS: SP - POINTS TO THE PC WHERE THE TRAP OCCURED.
4241 ;* ADRPTR - LABEL AT THE ADDRESS WHERE "EXPECTED" TRAPS OCCUR.
4242 ;* TP4FLG - 004 TRAP FLAGS.
4243 ;*
4244 ;* OUTPUTS: TP4FLG - BIT 15 IS SET IF "EXPECTED" TRAP OCCURED.
4245 ;*
4246 ;* CALLING SEQUENCE: PUT ADDRESS POINTED TO BY TP4RTN IN 004 VECTOR.
4247 ;* OCCURENCE OF 004 TRAP VECTORS TO THIS ROUTINE.
4248 ;*
4249 ;* COMMENTS: ANY 004 TRAP WHICH OCCURS AT AN ADDRESS OTHER THAN THAT LABELED
4250 ;* ADRPTR WILL BE HANDLED BY THE NORMAL 004 TRAP SERVICE ROUTINE.
4251 ;*
4252 ;* SUBORDINATE ROUTINES CALLED: NONE.
4253 ;*****
4254
4255 017040 021627 013466 TP4RTN:: CMP (SP),#ADRPTR ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
4256 017044 001402 BEQ 2# ;IF THEY MATCH, CONTINUE THIS ROUTINE.
4257 017046 000177 163236 JMP @TP4VEC ;IF NOT, JUMP TO NORMAL 004 TRAP SERVICE RTN.
4258 017052 052767 100000 163232 2#: BIS @BIT15,TP4FLG ;SET THE 004 TRAP OCCURED FLAG.
4259 017060 000002 RTI ;ALL DONE, GO BACK TO THE TEST.

```

GLOBAL TRAP SERVICE ROUTINE - TP4RTN -

```

4261
4262 ;*****
4263 ;
4264 ;           FVTA.RPT
4265 ;
4266 ;*****
4267
4268
4269
4270 .SBTTL  REPORT CODING SECTION
4271
4272 ;**
4273 ; THE REPORT CODING SECTION CONTAINS THE
4274 ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
4275 ;--
4276
4277 017062      BGNRPT
4278 017062
4279 017062      EXIT  RPT
4279 017062      000167
4279 017064      000000
4280
4281
4282 .EVEN
4283 017066      ENDRPT
4283 017066
4283 017066      104425

```

```

L$RPT::
      .WORD  J$JMP
      .WORD  L10014-2-.

```

```

L10014: TRAP  C$RPT

```

PROTECTION TABLE

```

4285           .SBTTL  PROTECTION TABLE
4286
4287           ;*****
4288           ;
4289           ;           FVTSKL4.P11
4290           ;
4291           ;*****
4292
4293
4294
4295           ;**
4296           ; THIS TABLE IS USED BY THE RUNTIME SERVICES
4297           ; TO PROTECT THE LOAD MEDIA.
4298           ;--
4299
4300 017070      BGNPROT
4301 017070
4302 017070 177777
4303 017072 177777
4304 017074 177777
4305
4306 017076
4307           ENDPROT

                                L$PROT::
                                -1      ;OFFSET INTO P-TABLE FOR CSR ADDRESS
                                -1      ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
                                -1      ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

```

PROTECTION TABLE

```

4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343 017076
      017076
4344
4345 017076
      017076 012700 000040
      017102 104447
4346 017104
      017104 103416
4347
4348 017106
      017106 012700 000037
      017112 104447
4349 017114
      017114 103556
4350
4351 017116
      017116 012700 000035
      017122 104447
4352 017124
      017124 103555
4353
4354 017126
      017126 012700 000036
      017132 104447
4355 017134
      017134 103161
4356 017136 000167 000556
4357 017142
4358 017142
      017142 104433
4359
4360
4361
4362 017144
      017144 012700 000114
      017150 104462

```

```

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

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

```

INITIALIZE SECTION

```

017152 010001
4363 017154 012167 163136      MOV      (R1)+,CLKCSR      ;STORE CLOCK CSR ADDRESS.
4364 017160 012167 163134      MOV      (R1)+,CLKBRL     ;STORE CLOCK BUS REQ INT LEVEL.
4365 017164 012167 163132      MOV      (R1)+,CLKVEC     ;STORE CLOCK INTERRUPT VECTOR.
4366 017170 012167 163130      MOV      (R1)+,CLKHRZ     ;STORE CLOCK FREQUENCY.
4367 017174 026727 163124 000062  CMP      CLKHRZ,#50.      ;TEST FOR 50HZ LINE FREQUENCY.
4368 017202 001004              BNE      2#              ;BRANCH IF CLOCK IS NOT 50HZ.
4369 017204 012767 000024 163124  MOV      #20.,MSTICK      ;INDICATE 20MS PER CLOCK TICK.
4370 017212 000403              BR       4#
4371 017214 012767 000021 163114 2#:  MOV      #17.,MSTICK      ;INDICATE 17 MS PER CLOCK TICK.
4372 017222 012767 000021 163114 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
017222 016746 161052
017226 012746 016770
017232 016746 163064
017236 012746 000003
017242 104437
017244 062706 000010
4373 017250 016700 163050      MOV      CLKHRZ,RO       ;INITIALIZE THE BREAK COUNT
4374 017254 006300              ASL      RO              ; TO CAUSE A BREAK
4375 017256 010067 163052      MOV      RO,BCOUNT       ; EVERY 2 SECONDS.
4376 017262 012700 000240      SETPRI  #PRI05           ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
                                MOV      #PRI05,RO
                                TRAP     C#SPRI
017262 012700 000240
017266 104441
4377
4378
4379
4380
4381
4382 017270 016767 160510 163012  ;*
4383 017276 012767 017040 160500  ; ENABLE THE LINE TIME CLOCK (LTC) CHECKING TO MAKE SURE THAT THE CSR
                                ; IS ACCESSABLE.
                                ; FIRST SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
4384
4385
4386
4387 017304 005067 163002      ;-
4388 017310 012767 000100 162776      MOV      4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
4389 017316 012700 002314      MOV      #TP4RTN,4      ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
4390 017322 016701 162770      ;*
4391 017326 004767 174122      ; ENABLE LTC CHECKING FOR 004 TRAP IN CASE CSR IS NOT THERE.
4392 017332 016767 162752 160444      ;-
4393 017340 103403              CLR      TP4FLG          ;CLEAR THE 004 TRAP FLAG.
4394 017342 005067 162756      MOV      #BIT6,WORD1     ;SET UP TO SET BIT6 OF THE LTC CSR.
4395 017346 000402              MOV      #WORD1,RO       ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
                                MOV      CLKCSR,R1          ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
                                JSR      PC,CKTRAP        ;MOVE AND CHECK FOR TRAP.
                                MOV      TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
                                BCS      6#              ;IF NO TRAP, LTC IS THERE SO CONTINUE.
                                CLR      CLKHRZ           ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
                                BR       8#              ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
4396
4397
4398
4399 017350 004767 173604      ;*
4400
4401
4402
4403
4404 017354 016767 160424 162726 8#:  JSR      PC,CALMSL       ; CALIBRATE THE DELAY ROUTINE MILLI-SECOND DELAY COUNT VALUE.
4405 017362 012767 017040 160414
4406 017370 005067 162716
4407 017374 005067 162714
4408 017400 012700 002314
4409 017404 016701 162732
4410 017410 005067 162730
                                ;*
                                ; CHECK FOR MEMORY MANAGEMENT PRESENT ON THIS MACHINE.
                                ; IF MEM MGT IS PRESENT, DISABLE IT.
                                ;-
                                8#:  MOV      4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
                                MOV      #TP4RTN,4      ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
                                CLR      TP4FLG          ;CLEAR THE 004 TRAP FLAG.
                                CLR      WORD1          ;PREPARE TO CLEAR THE MEM MGT SRO REGISTER.
                                MOV      #WORD1,RO       ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
                                MOV      #MSRO,R1        ;SELECT MEM MGT SRO REGISTER AS DESTINATION.
                                CLR      #MPRES         ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.

```

INITIALIZE SECTION

```

4411 017414 005067 162726          CLR    MMENAB          ;INDICATE MEM MGT IS NOT ENABLED.
4412 017420 004767 174030          JSR    PC,CKTRAP      ;CLEAR THE MEM MCT SRO REG AND CHECK FOR TRAP.
4413 017424 016767 162660 160352  MOV    TP4VEC,4       ;RESTORE THE NORMAL 004 TRAP VECTOR.
4414 017432 103003                   BCC    10$           ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
4415 017434 012767 000001 162702  MOV    #1,MMPRES     ;INDICATE THAT MEM MGT IS PRESENT.
4416 017442 005067 162630 10$:  CLR    PASCNT        ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4417 017446 000167 000006          JMP    NEWPAS        ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
4418
4419 017452          NEWRES: BRESET      ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
      017452 104433          TRAP    C$RESET
4420 017454 005067 162616          CLR    PASCNT        ;CLR COUNTER USED IN REPORTING ROM VERSION #.
4421 017460          NEWPAS:
4422 017460 012767 177777 162552  MOV    #-1,UNITN     ;RESET LOGICAL DEVICE TO -1
4423
4424          ;+
4425          ; INCREMENT THE PASS COUNTER, CORRECT FOR ANY OVERFLOW.
4426          ; THIS COUNTER IS USED IN THE ROM VERSION TEST.
4427 017466 005267 162604          INC    PASCNT        ;INCREMENT THE PASS COUNTER.
4428 017472 001002          BNE    GETPRM       ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
4429 017474 005367 162576          DEC    PASCNT        ;SET PASS COUNT TO 177777 OCTAL.
4430
4431          ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
4432 017500          GETPRM:
4433 017500 005267 162534          INC    UNITN         ;INCREMENT LOGICAL DEVICE NUMBER
4434 017504 026767 162530 162300  CMP    UNITN,L$UNIT  ;SEE IF MAXIMUM UNIT NO. EXCEEDED
4435 017512 002362          BGE    NEWPAS        ;BR IF YES
4436
4437 017514          GPHARD UNITN,R1    ;GET P-TABLE POINTER INTO R1
      017514 016700 162520          MOV    UNITN,RO
      017520 104442          TRAP   C$GPHRD
      017522 010001          MOV    RO,R1
4438 017524          BCOMPLETE          30$    ;BR IF DEVICE AVAILABLE
4439 017524 103401          BR    GETPRM         ;SKIP THIS DEVICE
4440
4441
4442          ;***** HARDWARE PARAMETER MOVING CODE *****
4443 017530 012167 162506 30$:  MOV    (R1)+,CSRA    ;STORE DHV-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
4444 017534 012102          MOV    (R1)+,R2     ;GET THE RX INTERRUPT VECTOR ADDRESS.
4445 017536 010267 162466          MOV    R2,RXVECA    ;STORE RX INT VECTOR ADDRESS.
4446 017542 062702 000004          ADD    #4,R2        ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
4447 017546 010267 162460          MOV    R2,TXVECA    ;STORE TX INT VECTOR ADDRESS.
4448 017552 012167 162456          MOV    (R1)+,ACTLNS ;STORE DHV-11 ACTIVE LINE BIT MAP
4449 017556 012702 000377          MOV    #MAPLNS,R2   ;GET THE BIT MAP FOR ALL LINES.
4450 017562 005102          COM    R2           ;GET A BIT MAP OF NON-EXISTANT LINES.
4451 017564 040267 162444          BIC    R2,ACTLNS    ;CLEAR NON-EXISTANT LINES FROM ACTLNS.
4452 017570 112167 162442          MOVB   (R1)+,LOPBCK ;STORE DHV-11 LOOPBACK MODE
4453 017574 112167 162437          MOVB   (R1)+,BRLEVL ;STORE DHV-11 INTERUPT BUS REQUEST LEVEL
4454
4455          ;+
4456          ; CALCULATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
4457          ; DEVICE REGISTER ADDRESS TABLE.
4458 017600 016701 162436          MOV    CSRA,R1      ;COPY CSR ADDRESS
4459 017604 005201          INC    R1           ;INCREMENT CSR ADDRESS
4460 017606 005201          INC    R1           ; COPY BY 2.
4461 017610 012703 000007          MOV    #7,R3        ;SET UP REGISTER COUNT
4462 017614 012702 002244          MOV    #RBUFA,R2    ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE

```

## INITIALIZE SECTION

```

4463 017620 010122      12$:  MOV    R1,(R2)+    ;STORE REGISTER ADDRESS IN TABLE
4464 017622 005201      INC    R1            ;INCREMENT REGISTER ADDRESS
4465 017624 005201      INC    R1            ; BY 2, FOR THE NEXT DEVICE REGISTER.
4466 017626 005303      DEC    R3            ;DECREMENT REGISTER COUNT
4467 017630 001373      BNE   12$           ;LOOP IF NOT DONE
4468
4469
4470
4471      ;+
4471      ; INITIALISE THE BMP CODE QUEUE.
4471      ;-
4472 017632 012700 002444      MOV    #BMPQCB,R0    ;GET THE START ADDRESS OF THE QUEUE.
4473 017636 012701 002644      MOV    #BMPQCE,R1    ;GET THE END ADDRESS OF THE QUEUE.
4474 017642 010067 162574      MOV    R0,BMPQCP     ;SET THE POINTER TO THE START OF THE QUEUE.
4475 017646 005020      14$:  CLR    (R0)+         ;CLEAR OUT THE CONTENTS OF THE QUEUE.
4476 017650 020001      CMP    R0,R1         ;CHECK IF END OF QUEUE HAS BEEN REACHED.
4477 017652 103775      BLO   14$           ;LOOP IF NOT ALL DONE.
4478
4479      ;+
4480      ; REPORT THE UNIT NUMBER IF THE SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
4481      ; AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
4481      ;-
4482 017654 032767 000020 162342      BIT    #BIT4,OPTION  ;CHECK IF THE QUESTION WAS ANSWERED YES.
4483 017662 001416      BEQ   16$           ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
4484 017664 026727 162122 000001      CMP    L$UNIT,#1    ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
4485 017672 003412      BLE   16$           ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
4486 017674      PRINTF #MFUNIT,UNITN ;REPORT UNIT NUMBER.
4486 017674 016746 162340      MOV    UNITN,-(SP)
4486 017700 012746 004214      MOV    #MFUNIT,-(SP)
4486 017704 012746 000002      MOV    #2,-(SP)
4486 017710 010600      MOV    SP,R0
4486 017712 104417      TRAP  C$PNTF
4486 017714 062706 000006      ADD   #6,SP
4487 017720
4488
4489 017720 005067 162340      16$:  ENDIT: CLR    CTRLCF    ;CLR THE CTRL-C TEST ABORT FLAG.
4490
4491      ;+
4492      ; SET THE PROCESSOR PRIORITY TO ALLOW LTC INTERRUPTS BUT NOT OTHERS.
4492      ;-
4493 017724      SETPRI #PRI07        ;SET PROCESSOR PRIORITY TO 7.
4493 017724 012700 000340      MOV    #PRI07,R0
4493 017730 104441      TRAP  C$SPRI
4494 017732      ENDINIT
4494 017732      L10016: TRAP  C$INIT
4494 017732 104411
4495
4496      TNUM == 0      ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.

```

INITIALIZE SECTION

```

4499 ;
4500 ;
4501 ;          FVTA.ATD
4502 ;
4503 ;
4505 ;

```

.SBTTL AUTODROP SECTION

```

4509 ;
4510 ;
4511 ; **
4512 ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
4513 ; THE "ADR" FLAG WAS SET. THE UNIT(S) UNDER TEST ARE CHECKED TO
4514 ; SEE IF THEY WILL RESPOND. THOSE THAT DON'T ARE IMMEDIATELY
4515 ; DROPPED FROM TESTING.
4516 ; **

```

```

4517 017734          BGNAUTO
4518 017734
4518 ;
4525 ;
4526 017734          ENDAUTO
4526 017734
4526 017734 104461
                                L$AUTO::
                                L10017: TRAP C$AUTO

```



AUTODROP SECTION

4528  
4529  
4530  
4531  
4532  
4533  
4534  
4535  
4536  
4537  
4538  
4539  
4540  
4541  
4542  
4543  
4544 017736  
017736  
4545  
4546 017736 005767 162322  
4547 017742 001401  
4548 017744  
017744 104433  
4549 017746  
4550  
4559  
4560 017746  
017746 104432  
017750 000002  
4561  
4573  
4574  
4575  
4576 017752  
017752  
017752 104412

```
*****  
:  
:          FVT.CUC  
:  
:*****
```

.SBTTL CLEANUP CODING SECTION

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

```
          BGNCLN  
          L#CLEAN: :  
          TST  CTRLCF  ;DID WE GET HERE BY CTRL-C FROM TEST?  
          BEQ  2#      ;CTRL-C FROM TEST? NO, SKIP BUS RESET.  
          BRESET                ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.  
                                TRAP  C#RESET  
2#:  
          EXIT  CLN  
                                TRAP  C#EXIT  
                                .WORD L10020-  
          .EVEN  
          ENDCLN  
                                L10020: TRAP  C#CLEAN
```

CLEANUP CODING SECTION

4578  
4579  
4580  
4581  
4582  
4583  
4584  
4585  
4586  
4587  
4588  
4589  
4590  
4591  
4592  
4593  
4594  
4595  
4604  
4605  
4606  
4607  
4608  
4609  
4610  
4611  
4612  
4613  
4614

017754  
017754  
017754 010046  
017756 012746 020000  
017762 012746 000002  
017766 010600  
017770 104417  
017772 062706 000006  
017776 000427  
  
020000 045 101 040  
020003 125 116 111  
020006 124 045 104  
020011 066 045 101  
020014 040 104 122  
020017 117 120 120  
020022 105 104 040  
020025 106 122 117  
020030 115 040 106  
020033 125 122 124  
020036 110 105 122  
020041 040 124 105  
020044 123 124 111  
020047 116 107 056  
020052 045 116 000  
  
020056  
020056 000167  
020060 000000  
  
020062  
020062  
020062 104453

```

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

.SBTTL DROP UNIT SECTION

```

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

```

        BGNDU
                                L#DU::
        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
                                .WORD   J#JMP
                                .WORD   L10021-2-.

        ENDDU
                                L10021:
                                TRAP    C#DU
    
```

DROP UNIT SECTION

4616  
4617  
4618  
4619  
4620  
4621  
4622  
4623  
4624  
4625  
4626  
4627  
4628  
4629  
4630  
4631  
4632  
4633  
4634  
4643  
4644  
4645  
4646  
4647  
4648  
4649  
4650

020064  
020064  
  
020064 000167  
020066 000000  
  
020070  
020070  
020070 104452

\*\*\*\*\*  
: FVTA.ADD  
:\*\*\*\*\*

.SBTTL ADD UNIT SECTION

\*\*\*  
: THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES  
: TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK  
: TO THE TEST CYCLE.  
:--

BGNAU

L\$AU::

EXIT AU

.WORD J\$JMP  
.WORD L10022-2-.

.EVEN

ENDAU

L10022: TRAP C\$AU

HARDWARE TEST - ADRA -

```

4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666 020072
      020072
4667      000001
4668 020072 012767 000001 162166
4669 020100 012767 177777 162156
4670
4671
4672
4673 020106 016767 157672 162174
4674 020114 012767 017040 157662
4675 020122 005005
4676
4677
4678
4679
4680 020124 005004
4681
4682
4683
4684
4685
4686 020126 005067 162160
4687 020132 016700 162104
4688 020136 012701 020352
4689 020142 004767 173306
4690 020146 103402
4691 020150 052705 100001
4692 020154 042767 000017 000170
4693 020162 050467 000164
4694 020166 010100
4695 020170 016701 162046
4696 020174 004767 173254
4697 020200 103403
4698 020202 052705 100002
4699 020206 000440
4700
4701
4702
4703 020210 012702 000010
4704 020214 016767 162022 000126
4705 020222 012700 020350
4706 020226 012701 020352
4707 020232 004767 173216

```

```

.SBTTL HARDWARE TEST - ADRA -
; **
; *****
; * - REGISTER ADDRESS TEST -
; *
; * THIS TEST VERIFIES THAT THE Q-BUS CAN READ AND WRITE TO THE DHV11
; * DEVICE REGISTERS. IF THE DHV11 DOES NOT RESPOND TO THE ACCESS
; * ATTEMPTS (IF THE DHV11 IS AT THE WRONG ADDRESS, FOR EXAMPLE) THE
; * 004 BUS TIME-OUT TRAP IS DETECTED BY THIS ROUTINE AND AN ERROR
; * IS REPORTED.
; *
; *****
; --
      BGNTST
      TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV #TNUM,TSTNUM    ;SET UP THE TEST NUMBER. (1)
      MOV #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
; *
; * SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
; *
      MOV 4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
      MOV #TP4RTN,4      ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
      CLR R5              ;CLEAR THE ERROR FLAGS.
; *
; * SET UP FOR THE INITIAL ITERATION OF THE TEST LOOP:
; *
      CLR R4              ;CLEAR THE LINE COUNTER.
; *
; * 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.
; *
20:   CLR TP4FLG           ;CLEAR THE 004 TRAP FLAG.
      MOV CSRA,R0        ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
      MOV #52,R1         ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
      JSR PC,CKTRAP      ;MOVE AND CHECK FOR TRAP.
      BCS 4#             ;IF NO TRAP, BYPASS ERROR.
      BIS #100001,R5     ;SET FATAL READ ERROR FLAGS.
44:   BIC #17,52#        ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
      BIS R4,52#        ;OR IN THE LINE COUNTER TO THE I.A.R FIELD.
      MOV R1,R0         ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
      MOV CSRA,R1       ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
      JSR PC,CKTRAP      ;MOVE AND CHECK FOR TRAP.
      BCS 6#            ;IF NO TRAP, BYPASS ERROR.
      BIS #100002,R5     ;SET FATAL WRITE ERROR FLAGS.
      BR 40#            ;EXIT AND REPORT FATAL ERROR.
; *
; * NOW, WE TEST EACH REGISTER FOR THIS LINE.
; *
68:   MOV #10,R2         ;INIT REGISTER COUNTER TO 8.
      MOV CSRA,50#      ;INITIALIZE THE REGISTER POINTER.
88:   MOV #50,R0        ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
      MOV #52,R1       ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
      JSR PC,CKTRAP     ;PERFORM THE MOVE, CHECK FOR TRAP.

```

## HARDWARE TEST

- ADRA -

```

4708 020236 103402          BCS      10#          ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
4709 020240 052705 100001    BIS      #100001,R5    ;SET FATAL READ ERROR FLAGS.
4710 020244 010100          10#:    MOV      R1,R0      ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
4711 020246 012701 020350    MOV      #50#,R1     ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.
4712 020252 004767 173176    JSR      PC,CKTRAP   ;PERFORM THE MOVE, CHECK FOR TRAP.
4713 020256 103402          BCS      12#          ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
4714 020260 052705 100002    BIS      #100002,R5    ;SET FATAL WRITE ERROR FLAGS.
4715 020264 005267 000060    12#:    INC      50#      ;INCREMENT THE REGISTER
4716 020270 005267 000054    INC      50#          ; POINTER BY 2.
4717 020274 005302          DEC      R2           ;COUNT THE REGISTER.
4718 020276 001351          BNE      8#           ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
4719
4720
4721          ;*
4722          ; NOW WE SET UP TO TEST THE NEXT LINE, OR TO EXIT IF WE ARE DONE.
4723 020300 005204          ;-
4724 020302 020427 000010    INC      R4           ;INCREMENT THE LINE COUNTER.
4725 020306 002707          CMP      R4,#NUMLNS   ;COMPARE LINE COUNTER AGAINST NUMBER OF LINES.
4726          BLT      2#           ;LOOP TO TEST THE NEXT LINE IF WE'RE NOT DONE.
4727
4728          ;*
4729          ; DONE CHECKING DEVICE REGISTER ADDRESSES.
4730          ; REPORT ANY ERRORS AND EXIT.
4731 020310 016767 161774 157466 40#:    MOV      TP4VEC,4     ;RESTORE THE NORMAL 004 TRAP VECTOR.
4732 020316 005705          TST      R5           ;CHECK THE ERROR FLAGS.
4733 020320 100015          BPL      60#          ;EXIT ROUTINE IF NO ERRORS.
4734          ; REPORT "DEVICE REGISTER ACCESS ERRORS"
4735 020322          ERRDF 101,EM0103,ER0101; >>>> ERROR #101 <<<<<.
4736          TRAP  C#ERDF
4737          .WORD 101
4738          .WORD EM0103
4739          .WORD ER0101
4740
4741          DODU  UNITN          ;DROP THIS UNIT FROM FUTHER TESTING.
4742          MOV      UNITN,R0    ;
4743          TRAP  C#DODU
4744 020332 016700 161702    CLR      CTRLCF       ;INDICATE NO CTRL-C ABORT FROM TEST.
4745 020336 104451          DOCLN          ;ABORT THIS SUB PASS.
4746 020340 005067 161720    BR       60#          ;
4747 020344 104444          BR       60#          ;
4748 020346 000402          ;
4749
4750          ;*
4751          ; LOCAL STORAGE.
4752          ;-
4753          .WORD 0          ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
4754          .WORD 0          ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
4755 020350 000000 161704    CLR      CTRLCF       ;INDICATE THAT WE ARE NOT WITHIN A TEST.
4756 020352 000000          ENDTST
4757 020354 005067 161704
4758 020360
4759 020360 104401          L10023:
4760          TRAP  C#ETST

```

HARDWARE TEST

- DMASTA -

4749  
4750  
4751  
4752  
4753  
4754  
4755  
4756  
4757 020362  
020362  
4758 020362  
020362 012700 000240  
020366 104441  
4759 000002  
4760 020370 012767 000002 161670  
4761 020376 012767 177777 161660  
4762 020404 012767 000001 163472  
4763 020412 012767 007641 163466  
4764 020420 012767 005772 163462  
4765 020426 012767 012570 163456  
4766  
4767  
4768  
4769  
4770  
4771 020434 004767 173044  
4772 020440 103141  
4773  
4774 020442 004767 173544  
4775  
4776  
4777  
4778  
4779  
4780  
4781 020446 016705 161562  
4782 020452 012700 000204  
4783 020456 004767 176226  
4784 020462 012700 177670  
4785 020466 004767 176246  
4786 020472 004767 175532  
4787  
4788  
4789  
4790 020476 016705 161532  
4791 020502 005001  
4792 020504 012767 007642 163374 20:  
4793 020512 000241  
4794 020514 006005  
4795 020516 103106  
4796 020520 004767 174306  
4797 020524 103107  
4798  
4799  
4800  
4801  
4802

```

.SBTTL HARDWARE TEST          - DMASTA -
;+ *****
;*          - DMA START BIT TEST -
;* THIS TEST VERIFIES THAT THE DMA_START BIT IN THE DUT'S LINE CONTROL
;* REGISTERS WILL INITIATE DMA TRANSMISSION ON THE SELECTED LINE.
;* THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, ON ALL ACTIVE LINES.
;*
;-- *****
      BGNTST
      SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.          T2::
                                MOV      #PRI05,R0
                                TRAP    C$SPRI
      TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV      #TNUM,TSTNUM ;SET UP THE TEST NUMBER.          (40)
      MOV      #-1,CTRLCF   ;INDICATE THAT WE ARE IN A TEST.
      MOV      #1,ERRTYP    ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
      MOV      #4001.,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
      MOV      #EM4001,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTABL.
      MOV      #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
;+
; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; THIS SUBROUTINE REPORTS ERROR >>>> 4001 <<<<.
;--
      JSR      PC,CLNRST     ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
      BCC     50$           ;RESET FAILURE?, ABORT THIS TEST.
      JSR      PC,INDATP     ;INITIALSE THE 256 BYTE DATA PATTERN.
;+
; SET INTERNAL LOOPBACK,ENABLE RECEIVER FUNCTIONS ON ALL ACTIVE LINES.
; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
; 2 STOP BITS.
; ENABLE TRANSMITTERS ON ALL ACTIVE LINES.
;--
      MOV      ACTLNS,R5     ;PASS THE ACTIVE LINE BIT MAP.
      MOV      #204,R0       ;PASS THE LNCTRL CONTENTS.
      JSR      PC,WTWLNC     ;INITIALISE THE LNCTRL REGISTERS.
      MOV      #177670,R0    ;PASS THE LPR CONTENTS.
      JSR      PC,WTWLPR     ;INITIALSE THE LPR REGISTERS ON ALL LINES.
      JSR      PC,TXENBL     ;ENABLE TRANSMITTERS ON ALL LINES.
;+
; SET-UP OUTER LOOP TO TEST THE DMA_START BIT ON ALL ACTIVE LINES.
;--
      MOV      ACTLNS,R5     ;GET THE ACTIVE LINE BIT MAP.
      CLR      R1            ;CLEAR THE LINE NUMBER COUNTER.
      MOV      #4002.,ERRNBR ;SET THE ERROR NUMBER TO 4002.
      CLC                    ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
      ROR      R5            ;SHIFT THE BIT MAP INTO THE CARRY BIT.
      BCC     14$           ;DO NOT TEST THE LINE IF IT IS INACTIVE.
      JSR      PC,PUFIFO     ;PURGE THE FIFO.
      BCC     50$           ;GO REPORT ERROR IF FIFO WILL NOT CLEAR.
;+
; PERFORM DMA_START BIT TESTING ON EACH LINE INDIVIDUALLY.
; TEST EACH DMA_START BIT BEFORE TX'ING DATA PATTERN, REPORT ERROR IF SET.
; SET DMA_START BIT ON LUT, VERIFY IT IS SET, REPORT ERROR IF CLEAR.
; WAIT FOR DMA TO COMPLETE.

```

HARDWARE TEST - DMASTA -

```

4803          ; VERIFY DMA_START BIT IS CLEAR, REPORT ERROR IF SET.
4804          ; VERIFY CORRECT NUMBER OF CHARS WERE RECEIVED, REPORT ERROR IF < EXPECTED.
4805          ;
4806 020526 005267 163354          INC     ERRNBR          ;SET ERROR NUMBER TO 4003.
4807 020532 012702 002704          MOV     #8UFBAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
4808 020536 012703 000144          MOV     #100.,R3       ;PASS THE LENGTH OF THE DATA PATTERN.
4809 020542 004767 173210          JSR     PC,DODMA       ;TRANSMIT THE DATA PATTERN.
4810 020546 103067          BCC     12#           ;GO REPORT ERROR IF DMA_START BIT SET.
4811          ;
4812          ; TEST THE STATE OF THE DMA_START BIT ON THE LINE UNDER TEST.
4813          ; REPORT ERROR IF DMA_START BIT IS CLEAR.
4814          ;
4815 020550 005267 163332          INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4004.
4816 020554 010177 161462          MOV     R1,BCSRA       ;SELECT THE LINE CURRENTLY UNDER TEST.
4817 020560 105777 161472          TSTB   #TXAD2A        ;TEST THE STATE OF THE DMA_START BIT.
4818 020564 100060          BPL     12#           ;GO REPORT ERROR IF BIT IS CLEAR.
4819          ;
4820          ; WAIT FOR DMA TRANSMISSION TO COMPLETE.
4821          ;
4822 020566 005267 163314          4# : INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4005.
4823 020572 010103          MOV     R1,R3          ;SAVE THE LINE NUMBER.
4824 020574 012701 170226          MOV     #170226,R1     ;TEST BIT 15, TIMEOUT OF 150 MILLI SECS.
4825 020600 016702 161436          MOV     CSRA,R2       ;PASS THE ADDRESS OF THE REGISTER TO TEST.
4826 020604 004767 175764          JSR     PC,WAIBIS      ;WAIT FOR DMA TO COMPLETE.
4827 020610 103045          BCC     10#           ;GO REPORT ERROR IF TIMEOUT OCCURRED.
4828 020612 012704 000005          MOV     #5,R4         ;PASS DELAY OF 5 MILLI SECS.
4829 020616 004767 173074          JSR     PC,DELAY       ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
4830 020622 010301          MOV     R3,R1         ;RESTORE THE CURRENT LINE NUMBER.
4831          ;
4832          ; TEST THE STATE OF THE DMA_START BIT ON THE LINE UNDER TEST.
4833          ; REPORT ERROR IF DMA_START BIT IS SET.
4834          ;
4835 020624 005267 163256          INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4006.
4836 020630 010177 161406          MOV     R1,BCSRA       ;SELECT THE LINE CURRENTLY UNDER TEST.
4837 020634 105777 161416          TSTB   #TXAD2A        ;TEST THE STATE OF THE DMA_START BIT.
4838 020640 100432          BMI     12#           ;GO REPORT ERROR IF BIT IS STILL SET.
4839          ;
4840          ; VERIFY THE NUMBER OF CHARS RECEIVED = NUMBER OF CHARS EXPECTED.
4841          ; REPORT ERROR IF COUNT IS INCORRECT.
4842          ; IF MORE THAN 128 BMP CODES ARE FOUND THEN REPORT ERROR AND EXIT TEST.
4843          ;
4844 020642 005003          CLR     R3             ;CLEAR THE READ COUNTER.
4845 020644 012704 000200          MOV     #128.,R4       ;SET UP MAX BMP CODE READ COUNT.
4846 020650 012767 007647          MOV     #4007.,ERRNBR  ;SET ERROR NUMBER TO 4007.
4847 020656 017702 161362          MOV     BRBUFA,R2     ;READ THE CHARACTER FROM THE FIFO.
4848 020662 100021          BPL     12#           ;GO REPORT ERROR IF FIFO EMPTY TOO SOON.
4849 020664 012700 170301          MOV     #170301,R0     ;SET-UP BIT MASK OF A BMP CODE.
4850 020670 040200          BIC     R2,R0         ;TRY TO CLEAR THE BMP CODE MASK.
4851 020672 001007          BNE     8#            ;BRANCH IF NOT A BMP CODE.
4852 020674 005267 163206          INC     ERRNBR          ;INCREMENT ERROR NUMBER TO 4008.
4853 020700 004767 174636          JSR     PC,SAVBMP     ;SAVE THE BMP CODE ON THE QUEUE.
4854 020704 005304          DEC     R4            ;DECREMENT MAX BMP CODE READ COUNT.
4855 020706 001416          BEQ     50#          ;GO REPORT ERROR IF TOO MANY BMP CODES FOUND.
4856 020710 000757          BR      6#            ;DO NOT COUNT THE BMP CODE AS A VALID CHAR.
4857 020712 005203          8# : INC     R3             ;COUNT THIS CHARACTER.
4858 020714 020327 000144          CMP     R3,#100.      ;HAVE WE RECIEVED 100 CHARACTERS?.
4859 020720 002753          BLT     6#            ;LOOP UNTIL 100 (NON-BMP) CHARS ARE READ.

```

HARDWARE TEST - DMASTA -

```

4860 020722 000404      BR      14$      ;SKIP AROUND THE ERROR REPORT.
4861
4862
4863                    ;+
4864                    ; REPORT ERROR, SKIP FURTHER TESTING ON THIS LINE.
4865 020724 010301      10$:    MOV      R3,R1      ;RESTORE THE CURRENT LINE NUMBER.
4866
4867 020726 012702 006015 12$:    MOV      @EM4002,R2 ;PASS THE ERROR MESSAGE TO BE REPORTED.
4868                    ; "DMA_START BIT BAD ON LINE NN".
4869 020732            ERROR          ;
4870 020732 104460            ;          >>>> ERROR <<<<<.
4871                    ;                                TRAP      C#ERROR
4872 020734 005201      14$:    INC      R1          ;INCREMENT THE LINE NUMBER COUNTER.
4873 020736 005705            TST      R5          ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
4874 020740 001261            BNE     2$          ;YES; BRANCH TO TEST THE NEXT LINE.
4875 020742 000402            BR      60$          ;NO; EXIT THIS TEST.
4876 020744 004767 175030 50$:    JSR     PC,TSABRT ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
4877 020750 005067 161310 60$:    CLR     CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
4878
4879 020754            ENDTST
4879 020754            L10024:    TRAP      C#ETST
4879 020754 104401

```



HARDWARE TEST - DMABRT -

```

4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891 020756
      020756
4892 020756 012700 000240
      020756 104441
      020762 000003
4893
4894 020764 012767 000003 161274
4895 020772 012767 177777 161264
4896 021000 012767 000001 163076
4897 021006 012767 010005 163072
4898 021014 012767 006051 163066
4899 021022 012767 012570 163062
4900
4901
4902
4903
4904
4905 021030 004767 172450
4906 021034 103160
4907
4908 021036 004767 173150
4909
4910
4911
4912
4913
4914
4915 021042 016705 161166
4916 021046 012700 000204
4917 021052 004767 175632
4918 021056 012700 177670
4919 021062 004767 175652
4920 021066 004767 175136
4921
4922
4923
4924 021072 016705 161136
4925 021076 005001
4926 021100 012767 010006 163000 24:
4927 021106 000241
4928 021110 006005
4929 021112 103123
4930 021114 004767 173712
4931 021120 103124
4932
4933
4934

```

```

.SBTTL  HARDWARE TEST          - DMABRT -
; * *****
; * - DMA ABORT/RESTART TEST -
; * THIS TEST VERIFIES THAT EACH DMA_ABORT BIT WILL CORRECTLY HALT
; * A DMA TRANSMISSION, AND RETURN A TX_ACTION.
; * IT WILL ALSO VERIFY THAT THE ABORTED DMA TRANSMISSION CAN BE RESUMMED,
; * AND THAT A TX_ACTION IS RETURNED UPON COMPLETION.
; * THIS TEST IS PERFORMED IN INTERNAL LOOPBACK, ON ALL ACTIVE LINES.
; * *****
; - - - - -
      BGNTST
; - - - - -
;          T3::
      SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
;          MOV  #PRI05,R0
;          TRAP C#SPRI
      TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV  #TNUM,TSTNUM     ;SET UP THE TEST NUMBER. (41)
      MOV  #-1,CTRLCF       ;INDICATE THAT WE ARE IN A TEST.
      MOV  #1,ERRTYP        ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
      MOV  #4101.,ERRNBR    ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
      MOV  #EM4101,ERRMSG   ;SET ERROR MESSAGE ADDRESS IN ERRTABL.
      MOV  #ER9101,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
; *
; * RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; * CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; * THIS SUBROUTINE REPORTS ERROR >>>> 4101 <<<<<.
; - - - - -
      JSR  PC,CLNRST        ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
      BCC  60#              ;RESET FAILURE?, ABORT THIS TEST.
; - - - - -
      JSR  PC,INDATP        ;INITIALISE 256 BYTE DATA PATTERN.
; *
; * SET INTERNAL LOOPBACK,ENABLE RECEIVER FUNCTIONS ON ALL ACTIVE LINES.
; * SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
; * 2 STOP BITS.
; * ENABLE TRANSMITTERS ON ALL ACTIVE LINES.
; - - - - -
      MOV  ACTLNS,R5        ;PASS THE ACTIVE LINE BIT MAP.
      MOV  #204,R0          ;PASS THE LNCTRL CONTENTS.
      JSR  PC,WTMLNC        ;INITIALISE THE LNCTRL REGISTERS.
      MOV  #177670,R0       ;PASS THE LPR CONTENTS.
      JSR  PC,WTMLPR        ;INITIALSE THE LPR REGISTERS ON ALL LINES.
      JSR  PC,TXENBL        ;ENABLE TRANSMITTERS ON ALL LINES.
; *
; * PERFORM DMA_ABORT BIT TESTING ON EACH INDIVIDUAL (ACTIVE) LINE.
; - - - - -
      MOV  ACTLNS,R5        ;GET THE ACTIVE LINE BIT MAP.
      CLR  R1               ;CLEAR THE LINE NUMBER COUNTER.
      MOV  #4102.,ERRNBR   ;SET THE ERROR NUMBER TO 4102.
      CLC                    ;CLEAR THE CARRY BIT PRIOR TO SHIFTING BIT MAP.
      ROR  R5               ;SHIFT THE BIT MAP INTO THE CARRY BIT.
      BCC  10#              ;DO NOT TEST THE LINE IF IT IS INACTIVE.
      JSR  PC,PUFIFO        ;PURGE THE FIFO.
      BCC  50#              ;GO REPORT ERROR IF FIFO WILL NOT CLEAR.
; *
; * CHECK THE DMA_ABORT BIT BEFORE ENABLING DMA, REPORT ERROR IF SET.
; - - - - -

```

## HARDWARE TEST

- DMABRT -

```

4935 021122 005267 162760      INC      ERRNBR      ;INCREMENT ERROR NUMBER TO 4103.
4936 021126 010177 161110      MOV      R1,@CSRA    ;SELECT THE LINE CURRENTLY UNDER TEST.
4937 021132 032777 000001 161112  BIT      @BIT0,@LNCTRA ;TEST THE STATE OF THE DMA_ABORT BIT.
4938 021140 001105                BNE      6$          ;GO REPORT ERROR IF BIT IS SET.
4939
4940      ;+
4941      ; ENABLE DMA TX ON SELECTED LINE, WAIT FOR DMA TO TX APPROX 1/4 OF DATA.
4942      ; ABORT THE DMA TRANSMISSION. WAIT FOR TX_ACTION TO BE RETURNED.
4943 021142 005267 162740      INC      ERRNBR      ;SET ERROR NUMBER TO 4104.
4944 021146 012702 002704      MOV      @BUFBA,R2   ;PASS THE START OF THE DATA PATTERN TO TX.
4945 021152 012703 000400      MOV      @256.,R3    ;PASS THE LENGTH OF THE DATA PATTERN.
4946 021156 004767 172574      JSR      PC,DODMA    ;TRANSMIT THE DATA PATTERN.
4947 021162 103103                BCC      50$         ;GO REPORT ERROR IF THERE ARE TX PROBLEMS.
4948
4949      ;+
4950      ; WAIT FOR DMA TO TRANSMIT 1/4 OF THE DATA BEFORE ABORTING.
4951 021164 010177 161052      MOV      R1,@CSRA    ;SELECT THE LINE CURRENTLY UNDER TEST.
4952 021170 012704 000062      MOV      @50.,R4     ;PASS THE DELAY TIME OF 40 MILLI SECONDS.
4953 021174 004767 172516      JSR      PC,DELAY    ;WAIT FOR APPROX 1/4 OF DATA TO BE TX'D.
4954 021200 052777 000001 161044  BIS      @BIT0,@LNCTRA ;ABORT THE DMA TRANSMISSION.
4955
4956      ;+
4957      ; WAIT FOR TX_ACTION TO BE RETURNED, REPORT ERROR IF TIME-OUT OCCURS.
4958 021206 005267 162674      INC      ERRNBR      ;INCREMENT ERROR NUMBER TO 4105.
4959 021212 010103                MOV      R1,R3       ;SAVE THE LINE NUMBER.
4960 021214 012701 170012      MOV      @170012,R1  ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
4961 021220 016702 161016      MOV      CSRA,R2     ;PASS THE ADDRESS OF THE REGISTER TO TEST.
4962 021224 004767 175344      JSR      PC,WAIBIS   ;WAIT FOR DMA TO COMPLETE.
4963 021230 103050                BCC      4$          ;GO REPORT ERROR IF TIMEOUT OCCURRED.
4964 021232 010301                MOV      R3,R1       ;RESTORE THE CURRENT LINE NUMBER.
4965
4966      ;+
4967      ; VERIFY DMA_START BIT CLEAR, REPORT ERROR IF SET.
4968 021234 005267 162646      INC      ERRNBR      ;INCREMENT ERROR NUMBER TO 4106.
4969 021240 012702 006130      MOV      @EM4103,R2  ;SELECT MESSAGE TO BE REPORTED.
4970
4971 021244 010177 160772      MOV      R1,@CSRA    ;"DMA_START BIT FOUND SET AFTER DMA ABORTED".
4972 021250 105777 161002      TSTB    @TXAD2A     ;SELECT THE LINE CURRENTLY UNDER TEST.
4973 021254 100441                BMI      8$          ;TEST THE STATE OF THE DMA_START BIT.
4974
4975      ;+
4976      ; RESUME DMA TRANSMISSION BY CLEARING DMA_ABORT AND SETTING DMA_START.
4977 021256 042777 000001 160766  BIC      @BIT0,@LNCTRA ;CLEAR THE DMA_ABORT BIT.
4978 021264 052777 000200 160764  BIS      @BIT7,@TXAD2A ;SET THE DMA_START BIT.
4979
4980      ;+
4981      ; WAIT FOR DMA TRANSMISSION TO COMPLETE.
4982 021272 005267 162610      INC      ERRNBR      ;INCREMENT ERROR NUMBER TO 4107.
4983 021276 010103                MOV      R1,R3       ;SAVE THE LINE NUMBER.
4984 021300 012701 170536      MOV      @170536,R1  ;TEST BIT 15, TIMEOUT OF 350 MILLI SECS.
4985 021304 016702 160732      MOV      CSRA,R2     ;PASS THE ADDRESS OF THE REGISTER TO TEST.
4986 021310 004767 175260      JSR      PC,WAIBIS   ;WAIT FOR DMA TO COMPLETE.
4987 021314 103016                BCC      4$          ;GO REPORT ERROR IF TIMEOUT OCCURRED.
4988 021316 012704 000002      MOV      @2,R4       ;PASS TIME-OUT OF 2 MILLI SECS.
4989 021322 004767 172370      JSR      PC,DELAY    ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
4990 021326 010301                MOV      R3,R1       ;RESTORE THE CURRENT LINE NUMBER.
4991

```

HARDWARE TEST

- DMABRT -

```

4992                                     ; TEST THE STATE OF THE DMA_ABORT BIT ON THE LINE UNDER TEST.
4993                                     ; REPORT ERROR IF DMA_ABORT BIT IS SET.
4994                                     ;-
4995 021330 005267 162552                 INC   ERRNBR           ; INCREMENT ERROR NUMBER TO 4108.
4996 021334 010177 160702                 MOV   R1,@CSRA        ; SELECT THE LINE CURRENTLY UNDER TEST.
4997 021340 032777 000001 160704         BIT   @BIT0,@LNCTRA   ; TEST THE STATE OF THE DMA_ABORT BIT.
4998 021346 001002                         BNE   6$              ; GO REPORT ERROR IF BIT IS SET.
4999 021350 000404                         BR    10$             ; BRANCH TO CHECK FOR ANY MORE LINES TO TEST.
5000                                     ;+
5001                                     ; REPORT ERROR, SKIP FURTHER TESTING ON THIS LINE.
5002                                     ;-
5003 021352 010301                         4$:   MOV   R3,R1      ; RESTORE THE CURRENT LINE NUMBER.
5004                                     ;+
5005 021354 012702 006074                 6$:   MOV   @EM4102,R2 ; PASS THE ERROR MESSAGE TO BE REPORTED.
5006                                     ; "DMA_ABORT BIT BAD ON LINE NN".
5007 021360                                     8$:   ERROR           ; >>>> ERROR <<<<<.
5008 021360 104460                                     TRAP   C#ERROR
5009                                     ;+
5010                                     ; VERIFY ALL ACTIVE LINES HAVE BEEN TESTED.
5011 021362 005201                         10$:  INC   R1          ; INCREMENT THE LINE NUMBER COUNTER.
5012 021364 005705                         TST   R5              ; ARE THERE ANY MORE ACTIVE LINES TO TEST?.
5013 021366 001244                         BNE   2$              ; YES; BRANCH TO TEST THE NEXT LINE.
5014 021370 000402                         BR    60$             ; NO; EXIT THIS TEST.
5015                                     ;+
5016 021372 004767 174402                 50$:  JSR   PC,TSABRT  ; REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5017 021376 005067 160662                 60$:  CLR   CTRLCF     ; INDICATE THAT WE ARE NOT WITHIN A TEST.
5018                                     ;-
5019 021402                                     ENDTST
5019 021402                                     L10025:
5019 021402 104401                                     TRAP   C#ETST

```

HARDWARE TEST - OAUTOI -

```

5021 .SBTTL HARDWARE TEST - OAUTOI -
5022 ;*****
5023 ;* - OAUTO BIT INACTIVE TEST -
5024 ;*
5025 ;* THIS TEST VERIFIES THAT THE DUT'S OAUTO FUNCTION BEHAVES CORRECTLY
5026 ;* WHEN INACTIVE, IE OAUTO BIT CLEAR.
5027 ;* THIS TEST WILL ONLY EXECUTE IF STAGGERED LOOPBACK MODE IS SELECTED.
5028 ;* THE SPECIAL STAGGERED LOOPBACK CONNECTOR MUST BE FITTED.
5029 ;*
5030 ;-----*****
5031
5032 021404 BGNTST
5033 021404 126727 160626 000002 CMPB LOPBCK,#2 ;CHECK MODE SELECTED. T4::
5034 021412 001402 BEQ .+6 ;DO NOT EXIT IF STAGGERD LOPBCK MODE SELECTED.
5035 021414 000167 000524 JMP 60$ ;EXIT THIS TEST.
5036 021420 012700 000240 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
5037 021424 104441 MOV #PRI05,R0
5038 021426 012767 000004 160632 TRAP C$SPRI
5039 021434 012767 177777 160622 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5040 021442 012767 000001 162434 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (49)
5041 021450 012767 011445 162430 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
5042 021456 012767 006214 162424 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5043 021464 012767 012570 162420 MOV #4901,ERRNBR ;SET ERROR NUMBER TO 4901.
5044 ;
5045 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
5046 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
5047 ; THIS SUBROUTINE REPORTS ERROR >>>> 4901 <<<<<.
5048 ;
5049 021472 004767 172006 JSR PC,CLNRST ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
5050 021476 103402 BCS .+6 ;DO NOT EXIT IF RESET WAS SUCCESSFUL.
5051 021500 000167 000440 JMP 60$ ;EXIT THIS TEST.
5052 ;
5053 ; SET-UP THE ASSOCIATED TX/RX LINE NUMBER TABLES.
5054 ;
5055 021504 004767 171340 JSR PC,ASLNTL ;INITIALISE THE ASSOCIATED TX/RX TABLES.
5056 ;
5057 ; SET EXTERNAL LOOPBACK, DISABLE OAUTO AND ENABLE RECEIVER ON ALL ACTIVE LINES.
5058 ; SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
5059 ; 2 STOP BITS.
5060 ; ENABLE TRANSMITTERS ON ALL LINES.
5061 ;
5062 021510 016705 160520 MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
5063 021514 012700 000004 MOV #4,R0 ;PASS THE LNCTRL CONTENTS.
5064 021520 004767 175164 JSR PC,WTWLNLC ;INITIALISE THE LNCTRL REGISTERS.
5065 021524 012705 000377 MOV #MAPLNS,R5 ;PASS BIT MAP OF ALL LINES.
5066 021530 012700 177670 MOV #177670,R0 ;PASS THE LPR CONTENTS.
5067 021534 004767 175200 JSR PC,WTWLPR ;INITIALISE THE LPR REGISTERS ON ALL LINES.
5068 021540 004767 174464 JSR PC,TXENBL ;ENABLE TRANSMITTERS ON ALL LINES.
5069 ;
5070 ; SET UP OUTER LOOP FOR TESTING ACTIVE LINES IN BOTH LINE GROUPS.
5071 ;
5072 021544 012703 100000 MOV #100000,R3 ;SET-UP LOOP CONTROL FLAG.
5073 021550 016705 160460 MOV ACTLNS,R5 ;GET THE ACTIVE LINE BIT MAP.
5074 021554 046705 160514 BIC LGRP2M,R5 ;REMOVE LINES IN GROUP 2.

```

HARDWARE TEST - OAUTOI -

```

5075 021560 010567 000352      2:      MOV      R5,45:      ;SAVE THE CURRENT LINE GROUP.
5076 021564 005067 000344      CLR      40:      ;CLEAR THE LINE NUMBER COUNTER.
5077 021570 016701 000340      4:      MOV      40:,R1    ;COPY THE LINE NUMBER.
5078 021574 000241      CLC      ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5079 021576 006005      ROR      R5        ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5080 021600 103054      BCC      8:        ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5081
5082      ;*
5083      ; TEST THE STATE OF THE OAUTO BIT ON THE LINE UNDER TEST.
5084      ; REPORT ERROR IF IT IS FOUND SET, AND SKIP FURTHER TESTING OF THAT LINE.
5085 021602 012767 011446 162276      MOV      #4902,ERRNBR ;SET THE ERROR NUMBER TO 4902.
5086 021610 010177 160426      MOV      R1,BCSRA   ;SELECT THE LINE TO BE TESTED.
5087 021614 032777 000020 160430      BIT      #BIT4,BLNCTRA ;TEST THE STATE OF THE OAUTO BIT.
5088 021622 001404      BEQ      6:        ;SKIP ERROR REPORT IF OAUTO BIT IS CLEAR.
5089 021624 012702 006246      MOV      #EM4902,R2 ;PASS THE ERROR MESSAGE.
5090      ; "OAUTO BIT BAD ON LINE NN"
5091 021630      ERROR      ;
5092 021630 104460      ; >>>> ERROR #4902 <<<<<.
5093 021632 000437      BR      #          TRAP      C:ERROR
5094      ;*
5095      ; TRANSMIT THE XOFF (ASCII DC3) ON THE ASSOCIATED LINE.
5096 021634 116177 004044 160400 6:      MOV      TXLNB(R1),BCSRA ;SELECT THE ASSOCIATED TX LINE.
5097 021642 012777 100023 160374      MOV      #100023,CTXCHA ;TRANSMIT THE XOFF CHARACTER TO THE LUT.
5098
5099      ;*
5100      ; WAIT FOR TRANSMISSION TO COMPLETE.
5101 021650 005267 162232      INC      ERRNBR     ;INCREMENT ERROR NUMBER TO 4903.
5102 021654 012701 170012      MOV      #170012,R1 ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
5103 021660 016702 160356      MOV      CSRA,R2   ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5104 021664 004767 174704      JSR      PC,WAIBIS ;WAIT FOR DMA TO COMPLETE.
5105 021670 103123      BCC      50:       ;ABORT TEST IF TIMEOUT OCCURRED.
5106 021672 012704 000005      MOV      #5,R4     ;PASS TIME-OUT OF 5 MILLI SECS.
5107 021676 004767 172014      JSR      PC,DELAY  ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5108
5109      ;*
5110      ; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
5111      ; REPORT ERROR IF TX_ENABLE BIT IS CLEAR.
5112 021702 005267 162200      INC      ERRNBR     ;INCREMENT ERROR NUMBER TO 4904.
5113 021706 016701 000222      MOV      40:,R1    ;GET THE NUMBER OF THE LINE TEST.
5114 021712 010177 160324      MOV      R1,BCSRA  ;SELECT THE LINE CURRENTLY UNDER TEST.
5115 021716 005777 160334      TST      #TXAD2A   ;TEST THE STATE OF THE TX_ENABLE BIT.
5116 021722 100403      BMI      8:        ;SKIP ERROR REPORT IF BIT IS SET.
5117 021724 012702 006246      MOV      #EM4902,R2 ;PASS THE MESSAGE TO BE REPORTED.
5118      ; "OAUTO BIT BAD ON LINE NN".
5119 021730      ERROR      ;
5120 021730 104460      ; >>>> ERROR #4904 <<<<<.
5121 021732 005267 000176      8:      INC      40:       ;INCREMENT THE LINE NUMBER.
5122 021736 005705      TST      R5        ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5123 021740 001313      BNE      4:        ;
5124
5125      ;*
5126      ; DISABLE TRANSMITTERS ON THE SELECTED LINES IN THE CURRENT LINE GROUP.
5127 021742 016705 000170      MOV      45:,R5    ;RESTORE THE CURRENT LINE ACTIVE LINE GROUP.
5128 021746 004767 174162      JSR      PC,TXDSBL ;DISABLE TRANSMITTERS ON THE SELECTED LINES.
5129 021752 016705 000160      MOV      45:,R5    ;GET THE CURRENT ACTIVE LINE GROUP AGAIN.

```

HARDWARE TEST - OAUTOI -

```

5130 021756 005067 000152          CLR    408          ;CLEAR THE LINE COUNTER.
5131 021762 012767 011451 162116 108:  MOV    #4905.,ERRNBR ;SET ERROR NUMBER TO 4905.
5132 021770 016701 000140          MOV    408,R1       ;COPY THE LINE NUMBER.
5133 021774 000241          CLC          ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5134 021776 006005          ROR     R5         ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5135 022000 103035          BCC    128        ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5136
5137          ;*
5138          ; TRANSMIT THE XON (ASCII DC1) ON THE ASSOCIATED LINE.
5139 022002 116177 004044 160232  MOVB   TXRLNB(R1),BCSRA ;SELECT THE ASSOCIATED TX LINE.
5140 022010 012777 100021 160226  MOV    #100021,BTXCHA ;TRANSMIT THE XON CHARACTER TO THE LUT.
5141
5142          ;*
5143          ; WAIT FOR TRANSMISSION TO COMPLETE.
5144 022016 012701 170012          MOV    #170012,R1   ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
5145 022022 016702 160214          MOV    CSRA,R2     ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5146 022026 004767 174542          JSR    PC,WAIBIS   ;WAIT FOR DMA TO COMPLETE.
5147 022032 103042          BCC    508        ;ABORT TEST IF TIMEOUT OCCURRED.
5148 022034 012704 000005          MOV    #5,R4      ;PASS TIME-OUT OF 5 MILLI SECS.
5149 022040 004767 171652          JSR    PC,DELAY    ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5150
5151          ;*
5152          ; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
5153          ; REPORT ERROR IF TX_ENABLE BIT IS SET.
5154 022044 005267 162036          INC    ERRNBR      ;INCREMENT ERROR NUMBER TO 4906.
5155 022050 016701 000060          MOV    408,R1     ;GET THE NUMBER OF THE LINE UNDER TEST.
5156 022054 010177 160162          MOV    R1,BCSRA   ;SELECT THE LINE CURRENTLY UNDER TEST.
5157 022060 005777 160172          TST    BTXAD2A    ;TEST THE STATE OF THE TX_ENABLE BIT.
5158 022064 100003          BPL    128        ;SKIP ERROR REPORT IF BIT IS CLEAR.
5159 022066 012702 006246          MOV    #EM4902,R2 ;PASS THE MESSAGE TO BE REPORTED.
5160
5161          ERROR          ; "OAUTO BIT BAD ON LINE NN".
5162 022072 104460          ;          >>>>> ERROR #4906 <<<<<.
5163          ;          TRAP    C#ERROR
5164 022074 005267 000034 128:  INC    408        ;INCREMENT THE LINE NUMBER.
5165 022100 005705          TST    R5         ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5166 022102 001327          BNE    108        ;
5167
5168          ;*
5169          ; CHECK LOOP CONTROL FLAG TO DETERMINE IF BOTH SETS OF LINES HAVE BEEN TESTED
5170          ; IF THIS IS THE FIST TIME AROUND, RE-ENABLE TX ON ALL LINES, GENERATE ACTIVE
5171          ; BIT MAP FOR SECOND LINE GROUP.
5171 022104 005703          ;*
5172 022106 001416          TST    R3         ;HAVE BOTH LINE GROUPS BEEN TESTED?.
5173 022110 005003          BEQ    608        ;YES; THEN EXIT THIS TEST.
5174 022112 012705 000377          CLR    R3        ;NO; CLEAR THE LOOP CONTROL FLAG.
5175 022116 004767 174106          MOV    #MAPLNS,R5 ;PASS THE BIT MAP OF ALL AVAILABLE LINE.
5176 022122 016705 160106          JSR    PC,TXENBL  ;RE-ENABLE TRANSMISSION ON ALL LINES.
5177 022126 046705 160140          MOV    ACTLNS,R5  ;GET THE ACTIVE LINE BIT MAP.
5178 022132 000612          BIC    LGRP1M,R5 ;REMOVE ALL ACTIVE LINES IN GROUP 1.
5179          BR     28   ;ONCE MORE AROUND AND WE ARE DONE.
5180 022134 000000 408:  .WORD  0          ;STORAGE FOR CURRENT LINE NUMBER.
5181 022136 000000 458:  .WORD  0          ;STORAGE FOR CURRENT ACTIVE LINE BIT MAP.
5182 022140 004767 173634 508:  JSR    PC,TSABRT  ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5183 022144 005067 160114 608:  CLR    CTRLCF     ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5184
5185 022150          ENDTST

```

HARDWARE TEST - OAUTOI -

022150  
022150 104401

L10026: TRAP C#ETST

HARDWARE TEST - OAUTOI -

```

5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199 022152
      022152
5200 022152 126727 160060 000002
5201 022160 001402
5202 022162 000167 000524
5203 022166
      022166 012700 000240
      022172 104441
5204      000005
5205 022174 012767 000005 160064
5206 022202 012767 177777 160054
5207 022210 012767 000001 161666
5208 022216 012767 011611 161662
5209 022224 012767 006300 161656
5210 022232 012767 012570 161652
5211
5212
5213
5214
5215
5216 022240 004767 171240
5217 022244 103402
5218 022246 000167 000440
5219
5220
5221
5222 022252 004767 170572
5223
5224
5225
5226
5227
5228
5229 022256 016705 157752
5230 022262 012700 000024
5231 022266 004767 174416
5232 022272 012705 000377
5233 022276 012700 177670
5234 022302 004767 174432
5235 022306 004767 173716
5236
5237
5238
5239 022312 012703 100000
5240 022316 016705 157712

```

```

.SBTTL HARDWARE TEST - OAUTOA -
;*****
;*
;* - OAUTO BIT ACTIVE TEST -
;*
;* THIS TEST VERIFIES THAT THE DUT'S OAUTO FUNCTION BEHAVES CORRECTLY
;* WHEN ACTIVE, IE OAUTO BIT ASSERTED HIGH.
;* THIS TEST WILL ONLY EXECUTE IF THE STAGGERED LOOPBACK MODE IS SELECTED.
;* THE SPECIAL STAGGERED LOOPBACK CONNECTOR MUST BE FITTED.
;*****
;-----
BGNTST
;
; T5::
;
; CMPB LOPBCK,#2 ;CHECK MODE SELECTED.
; BEQ .+6 ;DO NOT EXIT IF STAGGERD LOPBCK MODE SELECTED.
; JMP 601 ;EXIT THIS TEST.
; SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
;
; MOV #PRI05,R0
; TRAP C:SPRI
;
; TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
; MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (50)
; MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
; MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
; MOV #5001,ERRNBR ;SET ERROR NUMBER TO 5001.
; MOV #EM5001,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
; MOV #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
;
; *
; * RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; * CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; * THIS SUBROUTINE REPORTS ERROR >>>> 5001 <<<<.
; *
; *
; * JSR PC,CLNRST ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
; * BCS .+6 ;DO NOT EXIT IF RESET WAS SUCCESSFUL.
; * JMP 601 ;EXIT THIS TEST.
; *
; *
; * SET-UP THE ASSOCIATED TX/RX LINE NUMBER TABLES.
; *
; * JSR PC,ASLNTL ;INITIALISE THE ASSOCIATED TX/RX TABLES.
; *
; *
; * SET EXTERNAL LOOPBACK,ENABLE OAUTO AND RECEIVER FUNCTIONS ON ALL ACTIVE LINES
; * SET LPR ON ALL LINES TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
; * 2 STOP BITS.
; * ENABLE TRANSMITTERS ON ALL LINES.
; *
; *
; * MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
; * MOV #24,R0 ;PASS THE LNCTRL CONTENTS.
; * JSR PC,WTMLNC ;INITIALISE THE LNCTRL REGISTERS.
; * MOV #MAPLNS,R5 ;PASS BIT MAP OF ALL LINES.
; * MOV #177670,R0 ;PASS THE LPR CONTENTS.
; * JSR PC,WTMLPR ;INITIALISE THE LPR REGISTERS ON ALL LINES.
; * JSR PC,TXENBL ;ENABLE TRANSMITTERS ON ALL LINES.
; *
; *
; * SET UP OUTER LOOP FOR TESTING ACTIVE LINES IN BOTH LINE GROUPS.
; *
; *
; * MOV #100000,R3 ;SET-UP LOOP CONTROL FLAG.
; * MOV ACTLNS,R5 ;GET THE ACTIVE LINE BIT MAP.

```



HARDWARE TEST

- OAUTOA -

```

5241 022322 046705 157746          BIC    LGRP2M,R5      ;REMOVE LINES IN GROUP 2.
5242 022326 010567 000352      2$:   MOV    R5,45$     ;SAVE THE CURRENT LINE GROUP.
5243 022332 005067 000344          CLR    40$          ;CLEAR THE LINE NUMBER COUNTER.
5244 022336 016701 000340      4$:   MOV    40$,R1      ;COPY THE LINE NUMBER.
5245 022342 000241          CLC          ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5246 022344 006005          ROR    R5          ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5247 022346 103054          BCC    8$          ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5248
5249          ;*
5250          ; TEST THE STATE OF THE OAUTO BIT ON THE LINE UNDER TEST.
5251          ; REPORT ERROR IF IT IS FOUND CLEAR, AND SKIP FURTHER TESTING OF THAT LINE.
5252 022350 012767 011612 161530      MOV    #5002.,ERRNBR ;SET THE ERROR NUMBER TO 5002.
5253 022356 010177 157660          MOV    R1,@CSRA     ;SELECT THE LINE TO BE TESTED.
5254 022362 032777 000020 157662      BIT    @BIT4,@LNCTRA ;TEST THE STATE OF THE OAUTO BIT.
5255 022370 001004          BNE    6$          ;SKIP ERROR REPORT IF OAUTO BIT IS SET.
5256 022372 012702 006246          MOV    @EM4902,R2   ;PASS THE ERROR MESSAGE.
5257          ; "OAUTO BIT BAD ON LINE NN"
5258 022376          ERROR          ; >>>>> ERROR #5002 <<<<<.
5259 022400 000437          BR     8$          ;SKIP FURTHER TESTING OF THIS LINE.
5260          ;*
5261          ; TRANSMIT THE XOFF (ASCII DC3) ON THE ASSOCIATED LINE.
5262          ;*
5263 022402 116177 004044 157632 6$:   MOVB   TXRLNB(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
5264 022410 012777 100023 157626      MOV    #100023,@TXCHA ;TRANSMIT THE XOFF CHARACTER TO THE LUT.
5265          ;*
5266          ; WAIT FOR TRANSMISSION TO COMPLETE.
5267          ;*
5268 022416 005267 161464          INC    ERRNBR       ;INCREMENT ERROR NUMBER TO 5003.
5269 022422 012701 170012          MOV    #170012,R1   ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
5270 022426 016702 157610          MOV    CSRA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5271 022432 004767 174136          JSR    PC,WAIBIS    ;WAIT FOR DMA TO COMPLETE.
5272 022436 103123          BCC    50$         ;ABORT TEST IF TIMEOUT OCCURRED.
5273 022440 012704 000005          MOV    #5,R4        ;PASS TIME-OUT OF 5 MILLI SECS.
5274 022444 004767 171246          JSR    PC,DELAY     ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5275          ;*
5276          ; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
5277          ; REPORT ERROR IF TX_ENABLE BIT IS SET.
5278          ;*
5279 022450 005267 161432          INC    ERRNBR       ;INCREMENT ERROR NUMBER TO 5004.
5280 022454 016701 000222          MOV    40$,R1      ;GET THE NUMBER OF THE LINE TEST.
5281 022460 010177 157556          MOV    R1,@CSRA     ;SELECT THE LINE CURRENTLY UNDER TEST.
5282 022464 005777 157566          TST    @TXAD2A     ;TEST THE STATE OF THE TX_ENABLE BIT.
5283 022470 100003          BPL    8$          ;SKIP ERROR REPORT IF BIT IS CLEAR.
5284 022472 012702 006246          MOV    @EM4902,R2   ;PASS THE MESSAGE TO BE REPORTED.
5285          ; "OAUTO BIT BAD ON LINE NN".
5286 022476          ERROR          ; >>>>> ERROR #5004 <<<<<.
5287          ;*
5288 022500 005267 000176      8$:   INC    40$         ;INCREMENT THE LINE NUMBER.
5289 022504 005705          TST    R5          ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5290 022506 001313          BNE    4$          ;
5291          ;*
5292          ; DISABLE TRANSMITTERS ON THE SELECTED LINES IN THE CURRENT LINE GROUP.
5293          ;*
5294 022510 016705 000170          MOV    45$,R5      ;RESTORE THE CURRENT LINE ACTIVE LINE GROUP.
5295 022514 004767 173414          JSR    PC,TXDSBL   ;DISABLE TRANSMITTERS ON THE SELECTED LINES.

```

## HARDWARE TEST

- OAUTOA -

```

5296 022520 016705 000160      MOV    45$,R5      ;GET THE CURRENT LINE ACTIVE LINE GROUP AGAIN.
5297 022524 005067 000152      CLR    40$        ;CLEAR THE LINE COUNTER.
5298 022530 012767 011615 161350 10$:  MOV    #5005.,ERRNBR ;SET ERROR NUMBER TO 5005.
5299 022536 016701 000140      MOV    40$,R1     ;COPY THE LINE NUMBER.
5300 022542 000241      CLC          ;CLEAR CARRY BIT PRIOR TO SHIFTING BIT MAP.
5301 022544 006005      ROR    R5        ;SHIFT ACTIVE LINE BIT MAP INTO CARRY BIT.
5302 022546 103035      BCC    12$       ;SKIP TESTING THIS LINE IF IT IS INACTIVE.
5303
5304      ;*
5304      ; TRANSMIT THE XON (ASCII DC1) ON THE ASSOCIATED LINE.
5305      ;-
5306 022550 116177 004044 157464      MOVB   TXRLNB(R1),@CSRA ;SELECT THE ASSOCIATED TX LINE.
5307 022556 012777 100021 157460      MOV    #100021,@TXCHA ;TRANSMIT THE XON CHARACTER TO THE LUT.
5308
5309      ;*
5309      ; WAIT FOR TRANSMISSION TO COMPLETE.
5310      ;-
5311 022564 012701 170012      MOV    #170012,R1  ;TEST BIT 15, TIMEOUT OF 10 MILLI SECS.
5312 022570 016702 157446      MOV    CSRA,R2    ;PASS THE ADDRESS OF THE REGISTER TO TEST.
5313 022574 004767 173774      JSR    PC,WAIBIS  ;WAIT FOR DMA TO COMPLETE.
5314 022600 103042      BCC    50$       ;ABORT TEST IF TIMEOUT OCCURRED.
5315 022602 012704 000005      MOV    #5,R4     ;PASS TIME-OUT OF 5 MILLI SECS.
5316 022606 004767 171104      JSR    PC,DELAY   ;WAIT FOR CHAR TO BE RECEIVED AND PROCESSED.
5317
5318      ;*
5318      ; TEST THE STATE OF THE TX_ENABLE BIT ON THE LINE UNDER TEST.
5319      ; REPORT ERROR IF TX_ENABLE BIT IS CLEAR.
5320      ;-
5321 022612 005267 161270      INC    ERRNBR     ;INCREMENT ERROR NUMBER TO 5006.
5322 022616 016701 000060      MOV    40$,R1    ;GET THE NUMBER OF THE LINE UNDER TEST.
5323 022622 010177 157414      MOV    R1,@CSRA  ;SELECT THE LINE CURRENTLY UNDER TEST.
5324 022626 005777 157424      TST   @TXAD2A    ;TEST THE STATE OF THE TX_ENABLE BIT.
5325 022632 100403      BMI    12$       ;SKIP ERROR REPORT IF BIT IS CLEAR.
5326 022634 012702 006246      MOV    #EM4902,R2 ;PASS THE MESSAGE TO BE REPORTED.
5327
5328 022640      ERROR          ; "OAUTO BIT BAD ON LINE NN".
5328 022640 104460      ;          >>>>> ERROR #5006 <<<<<.
5329      ;          TRAP    C#ERROR
5330 022642 005267 000034 12$:  INC    40$       ;INCREMENT THE LINE NUMBER.
5331 022646 005705      TST   R5        ;CHECK IF THERE ARE ANY MORE LINES TO TEST.
5332 022650 001327      BNE   10$       ;
5333
5334      ;*
5334      ; CHECK LOOP CONTROL FLAG TO DETERMINE IF BOTH SETS OF LINES HAVE BEEN TESTED
5335      ; IF THIS IS THE FIST TIME AROUND, RE-ENABLE TX ON ALL LINES, GENERATE ACTIVE
5336      ; BIT MAP FOR SECOND LINE GROUP.
5337      ;-
5338 022652 005703      TST   R3        ;HAVE BOTH LINE GROUPS BEEN TESTED?.
5339 022654 001416      BEQ   60$       ;YES; THEN EXIT THIS TEST.
5340 022656 005003      CLR   R3        ;NO; CLEAR THE LOOP CONTROL FLAG.
5341 022660 012705 000377      MOV   #MAPLNS,R5 ;PASS THE BIT MAP OF ALL AVAILABLE LINE.
5342 022664 004767 173340      JSR   PC,TXENBL ;RE-ENABLE TRANSMISSION ON ALL LINES.
5343 022670 016705 157340      MOV   ACTLNS,R5 ;GET THE ACTIVE LINE BIT MAP.
5344 022674 046705 157372      BIC   LGRP1M,R5 ;REMOVE ALL ACTIVE LINES IN GROUP 1.
5345 022700 000612      BR    2$        ;ONCE MORE AROUND AND WE ARE DONE.
5346
5347 022702 000000 40$:  .WORD 0         ;STORAGE FOR CURRENT LINE NUMBER.
5348 022704 000000 45$:  .WORD 0         ;STORAGE FOR CURRENT ACTIVE LINE BIT MAP.
5349 022706 004767 173066 50$:  JSR   PC,TSABRT ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5350 022712 005067 157346 60$:  CLR   CTRLCF    ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5351

```

H11

HARDWARE TEST - OAUTOA -

5352 022716  
022716  
022716 104401

ENDTST

L10027: TRAP C#ETST

HARDWARE TEST - OAUTOA -

5354  
5355  
5356  
5357  
5358  
5359  
5360  
5361  
5362  
5363  
5364  
5365  
5366  
5367  
5368  
5369  
5370  
5371  
5372 022720  
022720  
5373 022720  
022720 012700 000240  
022724 104441  
000006  
5374  
5375 022726 012767 000006 157332  
5376 022734 012767 177777 157322  
5377 022742 012767 000001 161134  
5378 022750 012767 011755 161130  
5379 022756 012767 006330 161124  
5380 022764 012767 012570 161120  
5381  
5382  
5383  
5384  
5385  
5386 022772 004767 170506  
5387 022776 103146  
5388  
5389  
5390  
5391  
5392  
5393  
5394 023000 004767 171236  
5395  
5396  
5397  
5398  
5399 023004 016705 157224  
5400 023010 012700 000204  
5401 023014 004767 173670  
5402 023020 012700 177670  
5403 023024 004767 173710  
5404 023030 016704 154756  
5405 023034 004767 170656  
5406  
5407

```

.SBTTL HARDWARE TEST - IAUTOI -
;*****
; - IAUTO BIT INACTIVE TEST -
;
; THIS TEST VERIFIES THAT THE DUT'S IAUTO FUNCTION BEHAVES CORRECTLY
; WHEN INACTIVE, IE. IAUTO BIT CLEAR.
; ALL ACTIVE LINES ARE TESTED INDIVIDUALLY BY FILLING THE FIFO
; THEN READING THE RECEIVED DATA CHECKING FOR THE PRESENCE OF
; XOFF(ASCII DC3) OR XON (ASCII DC1) CHARACTERS.
; IF ANY ARE FOUND THEN APPROPRIATE ERRORS ARE REPORTED.
; ANY BMP CODES THAT ARE FOUND WILL BE PLACED ON THE BMP CODE QUEUE,
; TO BE REPORTED LATER.
; THE CHARACTERS ARE TRANSMITTED ON ALL ACTIVE LINES, IN INTERNAL
; LOOPBACK MODE.
;*****

BGNTST

SETPRI @PRI05 ;ALLOW LTC INTERRUPTS. T6::
MOV @PRI05,R0
TRAP C@SPRI

TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
MOV @TNUM,TSTNUM ;SET UP THE TEST NUMBER. (51)
MOV @-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
MOV @1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
MOV @5101.,ERRNBR ;SET ERROR NUMBER TO 5101.
MOV @EM5101,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
MOV @ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.

;
; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; THIS SUBROUTINE REPORTS ERROR >>>> 5101 <<<<<.
;
JSR PC,CLNRST ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
BCC 600 ;EXIT TEST IF FATAL ERROR FOUND.

;
; INITIALIZE THE 256 BYTE DATA PATTERN.
; ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
; NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
;
JSR PC,INDTPX ;INITIALISE DATA PATTERN.

;
; SET INTERNAL LOOPBACK, DISABLE IAUTO, ENABLE RECEIVER ON THE SELECTED LINE.
; SET LPR TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
;
MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
MOV @204,R0 ;PASS INT'L LOPBCK, ENABLE RX, DISABLE IAUTO.
JSR PC,WTMLNC ;INITIALISE THE LINE CONTROL REGISTER.
MOV @177670,R0 ;PASS THE LPR CONTENTS.
JSR PC,WTMLPR ;SET THE LPR CONTENTS TO 38.4K BAUD.
MOV 10.,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
JSR PC,DELAY ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
;

```

HARDWARE TEST - IAUTOI -

```

5408 ; SET UP LOOP FOR ALL ACTIVE LINES.
5409 ; TEST THE STATE OF THE IAUTO BIT PRIOR TO TRANSMITTING THE DATA PATTERN.
5410 ; IF THE BIT IS SET, THEN REPORT THE ERROR AND SKIP TRANSMITTING
5411 ; THE DATA PATTERN ON THE SELECTED LINE.
5412 ; TRANSMIT A 256 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
5413 ; EMPTY THE FIFO, AND VERIFY NO XOFF OR XON CHARS WERE FOUND.
5414 ;
5415 023040 005001 ; CLR R1 ; CLEAR THE LINE NUMBER COUNTER.
5416 023042 005067 000244 ; CLR 55 ; CLEAR STORAGE FOR LINE NUMBER.
5417 023046 012767 011756 161032 2: ; MOV #5102.,ERRNBR ; SET THE ERROR NUMBER TO 5102.
5418 023054 004767 171752 ; JSR PC,PUFIFO ; PURGE THE FIFO.
5419 023060 103111 ; BCC 50 ; GO REPORT ERROR IF FIFO DID NOT PURGE.
5420 023062 000241 ; CLC ; CLEAR CARRY PRIOR TO ROTATING BIT MAP.
5421 023064 006005 ; ROR R5 ; ROTATE THE BIT MAP INTO THE CARRY BIT.
5422 023066 103077 ; BCC 12 ; BRANCH IF LINE IS INACTIVE.
5423 ;
5424 ; TEST THE IAUTO BIT ON THE SELECTED ACTIVE LINE.
5425 ; REPORT ERROR IF IT IS SET.
5426 ; DO NOT TRANSMIT THE DATA PATTERN ON THE SELECTED LINE.
5427 ;
5428 023070 005267 161012 ; INC ERRNBR ; SET ERROR NUMBER TO 5103.
5429 023074 010177 157142 ; MOV R1,BCSRA ; SELECT LINE TO TEST.
5430 023100 032777 000002 157144 ; BIT #BIT1,BLNCTRA ; TEST THE STATE OF THE IAUTO BIT ON THIS LINE.
5431 023106 001404 ; BEQ 4 ; SKIP ERROR IF IAUTO BIT CLEAR.
5432 023110 012702 006356 ; MOV #EM5102,R2 ; PASS THE CORRECT ERROR MESSAGE.
5433 023114 ; ERROR ; >>>> ERROR <<<<.
5434 023116 000463 ; BR 12 ; SKIP TRANSMITTING DATA PATTERN. TRAP C:ERROR
5435 ;
5436 ;
5437 ; TRANSMIT DATA PATTERN OF 256 CHARS.
5438 ;
5439 023120 005267 160762 4: ; INC ERRNBR ; SET ERROR NUMBER TO 5104.
5440 023124 012702 002704 ; MOV #BUFBA,R2 ; PASS THE START OF THE DATA PATTERN TO TX.
5441 023130 012703 000400 ; MOV #256.,R3 ; PASS THE LENGTH OF THE DATA PATTERN.
5442 023134 004767 170616 ; JSR PC,DODMA ; TRANSMIT THE DATA PATTERN.
5443 023140 103061 ; BCC 50 ; ABORT THE TEST IF ERROR FOUND DURING DMA TX.
5444 ;
5445 ;
5446 ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER PLUS XOFF
5447 ; TO ARRIVE IN THE FIFO.
5448 ;
5449 023142 005267 160740 ; INC ERRNBR ; SET ERROR NUMBER TO 5105.
5450 023146 012701 170536 ; MOV #170536,R1 ; PASS TIME-OUT VALUE OF 350 MILLI SECS.
5451 023152 016702 157064 ; MOV CSRA,R2 ; PASS THE ADDRESS OF THE CSR.
5452 023156 004767 173412 ; JSR PC,WAIBIS ; WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
5453 023162 103050 ; BCC 50 ; IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
5454 023164 012704 000012 ; MOV #10.,R4 ; PASS DELAY OF 10 MILLI SECS.
5455 023170 004767 170522 ; JSR PC,DELAY ; WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
5456 ;
5457 ;
5458 ; READ 256 CHARS FROM THE FIFO. REPORT ERROR IF ANY XOFF'S OR XON'S
5459 ; ARE FOUND.
5460 ;
5461 023174 005267 160706 ; INC ERRNBR ; INCREMENT ERROR NUMBER TO 5106.
5462 023200 012701 000400 ; MOV #256.,R1 ; INITIALISE THE READ COUNTER.
5463 023204 017702 157034 6: ; MOV SRBUFA,R2 ; READ CHAR FROM THE FIFO.

```

HARDWARE TEST - IAUTOI -

```

5464 023210 100035      BPL      50$      ;GO REPORT ERROR IF FIFO EMPTY.
5465
5466                    ;*
5467                    ; CHECK FOR BMP CODE IN THE FIFO.  SAVE ANY FOUND ON THE QUEUE.
5468 023212 012700 170301  MOV      #170301,R0    ;SET UP BMP BIT MASK.
5469 023216 040200      BIC      R2,R0        ;TRY TO CLEAR ALL THE BMP BITS.
5470 023220 001002      BNE      8$          ;SKIP BMPSAV IF NOT A BMP CODE.
5471 023222 004767 172314 JSR      PC,SAVBMP    ;SAVE THE BMP CODE ON THE QUEUE.
5472
5473                    ;*
5474                    ; CHECK FOR XOFF AND XON CHARACTERS.
5475 023226 120227 000023 8$:      CMPB   R2,#23      ;IS IT AN XOFF CHARACTER?.
5476 023232 001406      BEQ     10$          ;YES; GO REPORT ERROR.
5477 023234 120227 000021  CMPB   R2,#21      ;NO; IS IT AN XON CHARACTER?.
5478 023240 001403      BEQ     10$          ;YES; GO REPORT ERROR.
5479 023242 005301      DEC     R1          ;DECREMENT THE READ COUNT.
5480 023244 001357      BNE     6$          ;LOOP TO READ THE NEXT CHAR.
5481 023246 000407      BR      12$         ;GO CHECK FOR ANY UNTESTED ACTIVE LINES.
5482
5483 023250 005267 160632 10$:     INC     ERRNBR    ;SET ERROR NUMBER TO 5107.
5484 023254 016701 000032  MOV     55$,R1      ;PASS THE LINE NUMBER TO BE REPORTED.
5485 023260 012702 006414  MOV     #EM5103,R2  ;PASS THE ERROR MESSAGE TO BE REPORTED.
5486 023264 023264 104460  ERROR   ;          >>>> ERROR <<<<<.
5487                                     TRAP   C#ERROR
5488
5489                    ;*
5490                    ; CHECK IF ALL ACTIVE LINES HAVE BEEN TESTED.
5491 023266 005267 000020 12$:     INC     55$        ;INCREMENT LINE NUMBER.
5492 023272 016701 000014  MOV     55$,R1      ;GET NUMBER OF THE NEXT LINE TO TEST.
5493 023276 005705      TST    R5          ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
5494 023300 001262      BNE    2$          ;LOOP TO CHECK NEXT LINE.
5495 023302 000404      BR     60$        ;EXIT TEST.
5496 023304 004767 172470 50$:     JSR    PC,TSABRT    ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5497 023310 000401      BR     60$        ;EXIT THIS TEST.
5498 023312 000000      .WORD  0          ;STORAGE FOR LINE NUMBER.
5499 023314 005067 156744 60$:     CLR    CTRLCF     ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5500
5501 023320      ENDTST
5502 023320
5503 023320 104401      L10030: TRAP   C#ETST

```

HARDWARE TEST - IAUTOA -

```

5503 .SBTTL HARDWARE TEST - IAUTOA -
5504 ;*****
5505 ;*
5506 ;*
5507 ;*
5508 ;* THIS TEST VERIFIES THAT THE DUT'S IAUTO FUNCTION BEHAVES CORRECTLY
5509 ;* WHEN ACTIVE, IE IAUTO ASSERTED HIGH.
5510 ;* ALL ACTIVE LINES ARE TESTED INDIVIDUALLY BY FILLING THE FIFO, AND
5511 ;* CHECKING FOR THE PRESENCE OF AT LEAST ONE XOFF(ASCII DC3) CHARACTER
5512 ;* AND ONE XON (ASCII DC1) CHARACTER.
5513 ;* ANY BMP CODES THAT ARE FOUND WILL BE PLACED ON THE BMP CODE QUEUE,
5514 ;* TO BE REPORTED LATER.
5515 ;* THE CHARACTERS ARE TRANSMITTED ON ALL ACTIVE LINES, IN INTERNAL
5516 ;* LOOPBACK MODE.
5517 ;*
5518 ;-----*****
5519 023322 BGNTST
5520 023322 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T7::
5521 023322 012700 000240 ;MOV #PRI05,RO
5522 023322 104441 ;TRAP C#SPRI
5523 023322 000007 ;TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
5524 023330 012767 000007 156730 ;MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (52)
5525 023336 012767 177777 156720 ;MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
5526 023344 012767 000001 160532 ;MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
5527 023352 012767 012121 160526 ;MOV #5201,ERRNBR ;SET ERROR NUMBER TO 5201.
5528 023360 012767 006444 160522 ;MOV #EM5201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
5529 023366 012767 012570 160516 ;MOV #ER9101,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
5530 ;*
5531 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
5532 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
5533 ; THIS SUBROUTINE REPORTS ERROR >>>> 5201 <<<<<.
5534 ;-
5535 JSR PC,CLNRST ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
5536 BCC 60# ;EXIT TEST IF FATAL ERROR FOUND.
5537 ;*
5538 ; INITIALIZE THE 256 BYTE DATA PATTERN.
5539 ; ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
5540 ; NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
5541 ;-
5542 JSR PC,INDTPX ;INITIALISE DATA PATTERN.
5543 ;*
5544 ; SET INTERNAL LOOPBACK, ENABLE IAUTO AND RECEIVER ON THE SELECTED LINE.
5545 ; SET LPR TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
5546 ;-
5547 MOV ACTLNS,R5 ;PASS THE ACTIVE LINE BIT MAP.
5548 MOV #206,R0 ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
5549 JSR PC,WTWLNCR ;INITIALISE THE LINE CONTROL REGISTER.
5550 MOV #177670,R0 ;PASS THE LPR CONTENTS.
5551 JSR PC,WTWLPR ;SET THE LPR CONTENTS TO 38.4K BAUD.
5552 MOV 10.,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
5553 JSR PC,DELAY ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
5554 ;*
5555 ; SET UP LOOP FOR ALL ACTIVE LINES.
5556 ; TEST THE STATE OF THE OAUTO BIT PRIOR TO TRANSMITTING THE DATA PATTERN.

```

HARDWARE TEST

- IAUTOA -

```

5557      ; IF THE BIT IS CLEAR, THEN REPORT THE ERROR AND SKIP TRANSMITTING
5558      ; THE DATA PATTERN ON THE SELECTED LINE.
5559      ; TRANSMIT A 224 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
5560      ; EMPTY THE FIFO, AND COUNT THE XOFF AND AN XON CHARS FOUND.
5561      ;-
5562 023442 005001          CLR      R1          ;CLEAR THE LINE NUMBER COUNTER.
5563 023444 005067 000264  CLR      55#         ;CLEAR STORAGE FOR LINE NUMBER.
5564 023450 012767 012122 160430 2#:  MOV     #5202.,ERRNBR ;SET THE ERROR NUMBER TO 5202.
5565 023456 004767 171350      JSR     PC,PUFIFO    ;PURGE THE FIFO.
5566 023462 103121          BCC     50#         ;GO REPORT ERROR IF FIFO DID NOT PURGE.
5567 023464 000241          CLC          ;CLEAR CARRY PRIOR TO ROTATING BIT MAP.
5568 023466 006005          ROR      R5          ;ROTATE THE BIT MAP INTO THE CARRY BIT.
5569 023470 103107          BCC     16#         ;BRANCH IF LINE IS INACTIVE.
5570
5571      ;+
5572      ; TEST THE IAUTO BIT ON THE SELECTED ACTIVE LINE.
5573      ; REPORT ERROR IF IT IS CLEAR.
5574      ; DO NOT TRANSMIT THE DATA PATTERN ON THE SELECTED LINE.
5575      ;-
5575 023472 005267 160410      INC     ERRNBR      ;SET ERROR NUMBER TO 5203.
5576 023476 010177 156540      MOV     R1,BCSRA    ;SELECT LINE TO TEST.
5577 023502 032777 000002 156542  BIT     #BIT1,BLNCTRA ;TEST THE STATE OF THE IAUTO BIT ON THIS LINE.
5578 023510 001004          BNE     4#          ;SKIP ERROR IF IAUTO BIT SET.
5579 023512 012702 006470      MOV     #EM5202,R2 ;PASS THE CORRECT ERROR MESSAGE.
5580      ; "IAUTO BIT FOUND CLEAR ON LINE NN"
5581 023516          ERROR          ;
5582 023520 104460          TRAP     C#ERROR
5583 000473          BR      16#         ;SKIP TRANSMITTING DATA PATTERN.
5584
5585      ;+
5586      ; TRANSMIT DATA PATTERN TO FILL THE FIFO, 223 CHARS + 32 XOFF'S + XON.
5587      ;-
5587 023522 005267 160360 4#:  INC     ERRNBR      ;SET ERROR NUMBER TO 5204.
5588 023526 012702 002704      MOV     #BUFBA,R2  ;PASS THE START OF THE DATA PATTERN TO TX.
5589 023532 012703 000337      MOV     #223.,R3   ;PASS THE LENGTH OF THE DATA PATTERN.
5590 023536 004767 170214      JSR     PC,DODMA    ;TRANSMIT THE DATA PATTERN.
5591 023542 103071          BCC     50#         ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
5592
5593      ;+
5594      ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER PLUS XOFF
5595      ; TO ARRIVE IN THE FIFO.
5596      ;-
5597 023544 005267 160336      INC     ERRNBR      ;SET ERROR NUMBER TO 5205.
5598 023550 012701 170536      MOV     #170536,R1 ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
5599 023554 016702 156462      MOV     CSRA,R2     ;PASS THE ADDRESS OF THE CSR.
5600 023560 004767 173010      JSR     PC,WAIBIS   ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
5601 023564 103060          BCC     50#         ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
5602 023566 012704 000012      MOV     #10.,R4    ;PASS DELAY OF 10 MILLI SECS.
5603 023572 004767 170120      JSR     PC,DELAY    ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
5604
5605      ;+
5606      ; READ 256 CHARS FROM THE FIFO, COUNT ANY XOFF OR XON CHARS FOUND.
5607      ;-
5608 023576 005003          CLR     R3          ;CLEAR XOFF COUNTER.
5609 023600 005004          CLR     R4          ;CLEAR XON COUNTER.
5610 023602 005267 160300      INC     ERRNBR      ;INCREMENT ERROR NUMBER TO 5206.
5611 023606 012701 000400      MOV     #256.,R1   ;INITIALISE THE READ COUNTER.
5612 023612 017702 156426 6#:  MOV     @RBUFA,R2   ;READ CHAR FROM THE FIFO.

```



## HARDWARE TEST

- IAUTOA -

```

5613 023616 100043      BPL      50$           ;GO REPORT ERROR IF FIFO EMPTY.
5614                    ;+
5615                    ; CHECK FOR BMP CODE IN THE FIFO.  SAVE ANY FOUND ON THE QUEUE.
5616                    ;-
5617 023620 012700 170301  MOV     #170301,R0      ;SET UP BMP BIT MASK.
5618 023624 040200      BIC     R2,R0          ;TRY TO CLEAR ALL THE BMP BITS.
5619 023626 001002      BNE     8$             ;SKIP BMPSAV IF NOT A BMP CODE.
5620 023630 004767 171706  JSR     PC,SAVBMP      ;SAVE THE BMP CODE ON THE QUEUE.
5621                    ;+
5622                    ; CHECK FOR XOFF AND XON CHARACTERS.
5623                    ;-
5624 023634 120227 000023  8$:     CMPB   R2,#23      ;IS IT AN XOFF CHARACTER?.
5625 023640 001001      BNE     10$           ;NO, BRANCH TO SEE IF IT IS AN XON.
5626 023642 005203      INC     R3             ;COUNT THE XOFF CHAR.
5627 023644 120227 000021  10$:    CMPB   R2,#21      ;IS IT AN XON CHARACTER?.
5628 023650 001001      BNE     12$           ;NO, SKIP THE NEXT INSTRUCTION.
5629 023652 005204      INC     R4             ;COUNT THE XON.
5630 023654 005301      DEC     R1             ;DECREMENT THE READ COUNT.
5631 023656 001355      BNE     6$             ;LOOP TO READ THE NEXT CHAR.
5632                    ;+
5633                    ; VERIFY THAN AT LEAST 1 XOFF AND 1 XON WAS FOUND IN THE FIFO.
5634                    ; REPORT ERROR IF NONE WERE FOUND.
5635                    ;-
5636 023660 005703      TST     R3             ;CHECK XOFF COUNT.
5637 023662 001403      BEQ     14$           ;GO REPORT ERROR IF NONE FOUND.
5638 023664 020427 000001  CMP     R4,#1         ;CHECK XON COUNT = 1.
5639 023670 001407      BEQ     14$           ;SKIP THE ERROR REPORT IF ONE XON WAS FOUND.
5640 023672 005267 160210  14$:    INC     ERRNBR      ;SET ERROR NUMBER TO 5207.
5641 023676 016701 000032  MOV     55$,R1        ;PASS THE LINE NUMBER TO BE REPORTED.
5642 023702 012702 006414  MOV     #EM5103,R2    ;PASS THE ERROR MESSAGE TO BE REPORTED.
5643                    ; "IAUTO BIT BAD ON LINE NN".
5644 023706      ERROR      ; >>>>> ERROR <<<<<.
5645                    ; TRAP      C$ERROR
5646                    ;+
5647                    ; CHECK IF ALL ACTIVE LINES HAVE BEEN TESTED.
5648                    ;-
5648 023710 005267 000020  16$:    INC     55$         ;INCREMENT LINE NUMBER.
5649 023714 016701 000014  MOV     55$,R1        ;GET NUMBER OF THE NEXT LINE TO TEST.
5650 023720 005705      TST     R5             ;ARE THERE ANY MORE ACTIVE LINES TO TEST?.
5651 023722 001252      BNE     2$             ;LOOP TO CHECK NEXT LINE.
5652 023724 000404      BR      60$           ;EXIT TEST.
5653                    ;-
5654 023726 004767 172046  50$:    JSR     PC,TSABRT      ;REPORT TEST ABORTED. NON-TEST RELATED ERROR.
5655 023732 000401      BR      60$           ;EXIT THIS TEST.
5656 023734 000000      .WORD  0             ;STORAGE FOR LINE NUMBER.
5657 023736 005067 156322  60$:    CLR     CTRLCF        ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5658                    ;-
5659 023742      ENDTST
023742
023742 104401
L10031: TRAP      C$ETST

```

HARDWARE TEST

- FIFDAT -

```

5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677 023744
      023744
5678 023744
      023744 012700 000240
      023750 104441
5679      000010
5680 023752 012767 000010 156306
5681 023760 012767 177777 156276
5682 023766 012767 000001 160110
5683 023774 012767 012265 160104
5684 024002 012767 006526 160100
5685
5686
5687
5688
5689
5690 024010 004767 167470
5691 024014 103107
5692
5693
5694
5695
5696 024016 004767 170110
5697 024022 103104
5698 024024 004767 170162
5699
5700
5701
5702
5703
5704
5705
5706
5707 024030 012700 000204
5708 024034 004767 172650
5709 024040 012700 177670
5710 024044 004767 172670
5711 024050 016704 153736
5712 024054 004767 167636
5713 024060 012702 002704
5714 024064 012703 000400

```

```

.SBTTL HARDWARE TEST - FIFDAT -
;.....
; - FIFO VALID DATA TEST -
;
; THIS TEST VERIFIES THAT THE DUT IS CAPABLE OF HOLDING 256 VALID
; CHARACTERS IN ITS FIFO.
; THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
; INTERNAL LOOPBACK MODE.
; THE DATA FOUND IN THE FIFO IS COMPARED WITH THE EXPECTED DATA, AND ANY
; DISCREPANCIES ARE REPORTED.
; ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
; HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
; REPORTED LATER.
;.....
BGNTST
      T8::
SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
      MOV #PRI05,R0
      TRAP C$SPRI
      TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (53)
      MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
      MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
      MOV #5301,ERRNBR ;SET ERROR NUMBER TO 5301.
      MOV #EM5301,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
;
; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; THIS SUBROUTINE REPORTS ERROR >>>> 5301 <<<<<.
;-
      JSR PC,CLRST ;RESET THE DMV-11, REPORT ANY ERRORS FOUND.
      BCC 600 ;EXIT TEST IF FATAL ERROR FOUND.
;
; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
; INITIALISE 256 BYTE DATA PATTERN.
;-
      JSR PC,FINACT ;FIND AN ACTIVE LINE.
      BCC 600 ;EXIT IF NO ACTIVE LINES FOUND.
      JSR PC,INDATP ;INITIALISE THE DATA PATTERN.
;
; TRANSMIT A 265 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
;-
;
; SET INTERNAL LOOPBACK ON THE SELECTED LINE.
; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
;-
      MOV #204,R0 ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
      JSR PC,WMLNC ;INITILAISE THE LINE CONTROL REGISTER.
      MOV #177670,R0 ;PASS THE LPR CONTENTS.
      JSR PC,WMLPR ;SET THE LPR CONTENTS TO 38.4K BAUD.
      MOV 10,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
      JSR PC,DELAY ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
      MOV #BUFBAS,R2 ;PASS THE START OF THE DATA PATTERN TO TX.
      MOV #BUFMID-BUFBAS,R3 ;PASS THE LENGTH OF THE DATA PATTERN.

```

HARDWARE TEST

- FIFDAT -

```

5715 024070 005267 160012      INC      ERRNBR      ;SET ERROR NUMBER TO 5302.
5716 024074 004767 167656      JSR      PC,DODMA    ;TRANSMIT THE DATA PATTERN.
5717 024100 103053              BCC      50$         ;ABORT TEST IF ERROR FOUND DURING DMA TX.
5718
5719      ;*
5720      ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
5721      ; THE FIFO.
5722 024102 005267 160000      INC      ERRNBR      ;SET ERROR NUMBER TO 5303.
5723 024106 010103              MOV      R1,R3       ;SAVE THE NUMBER OF THE SELECTED ACTIVE LINE.
5724 024110 012701 170536      MOV      #170536,R1  ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
5725 024114 016702 156122      MOV      CSRA,R2     ;PASS THE ADDRESS OF THE CSR.
5726 024120 004767 172450      JSR      PC,WAIE...  ;WAIT FOR DMA TO COMPLETE, TX ACTION SET.
5727 024124 103041              BCC      50$         ;BRANCH IF FIFO EMPTY, ABORT THE TEST.
5728 024126 012704 000005      MOV      #5,R4       ;PASS DELAY OF 5 MILLI SECS.
5729 024132 004767 167560      JSR      PC,DELAY    ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
5730
5731      ;*
5732      ; READ THE FIFO CHECKING FOR DATA CORRUPTION, REPORT ANY ERRORS FOUND.
5733      ; ABORT THE TEST IF A BMP CODE WAS FOUND IN THE FIFO.
5734 024136 006303              ASL      R3           ;MULTIPLY BY 2.
5735 024140 005004              CLR      R4           ;INITIALISE THE EXPECTED DATA.
5736 024142 016705 156076      MOV      RBUFA,R5    ;GET THE ADDRESS OF THE RECEIVER BUFFER REG.
5737 024146 012767 012270 157732 2$:  MOV      #5304,,ERRNBR ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
5738 024154 011502              MOV      (R5),R2     ;GET THE ACTUAL DATA FROM THE FIFO.
5739 024156 100024              BPL      50$         ;ABORT THE TEST IF THE FIFO IS EMPTY.
5740
5741      ;*
5742      ; CHECK IF THE READ CHARACTER IS A BMP CODE.
5743      ; IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
5744      ; ABORT THE TEST.
5745 024160 005267 157722      INC      ERRNBR      ;SET ERROR NUMBER TO 5305.
5746 024164 004767 167214      JSR      PC,CHKBMP   ;CHECK IF CHARACTER IS A BMP CODE.
5747 024170 103002              BCC      4$         ;BRANCH IF NOT A BMP CODE.
5748 024172              ERROR              ;
5749 024174 000417              BR       60$        ;ABORT THIS TEST.
5750
5751 024176 005267 157704      4$: INC      ERRNBR      ;SET ERROR NUMBER TO 5306.
5752 024202 120402              CMPB    R4,R2       ;COMPARE THE EXPECTED WITH THE ACTUAL DATA.
5753 024204 001406              BEQ     8$         ;SKIP ERROR REPORT IF DATA IS OK.
5754 024206 012767 012422 157676      MOV      #ER9002,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
5755 024214 012701 006553      MOV      #EM5302,R1  ;PASS THE MESSAGE TO BE REPORTED.
5756
5757 024220              ;REPORT THE ERROR "FIFO BAD, DATA FIELD CORRUPTED"
5758 024220 104460              6$: ERROR              ;
5759 024222 105204              8$: INCB    R4         ;INCREMENT THE EXPECTED DATA.
5760 024224 001350              BNE     2$         ;LOOP IF NOT DONE.
5761 024226 000402              BR      60$        ;EXIT
5762 024230 004767 171544      50$: JSR      PC,TSABRT   ;ABORT THE TEST, REASON SHOWN BY ERROR NUMBER.
5763 024234 005067 156024      60$: CLR      CTRLCF   ;INDICATE THAT WE ARE NOT WITHIN A TEST.
5764
5765 024240              ENDTST
024240
024240 104401              L10032: TRAP      C!ETST

```

HARDWARE TEST

- FI3QLI -

```

5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783 024242
      024242
5784 024242
      024242 012700 000240
      024246 104441
5785      000011
5786 024250 012767 000011 156010
5787 024256 012767 177777 156000
5788 024264 012767 000001 157612
5789 024272 012767 012431 157606
5790 024300 012767 006703 157602
5791 024306 012767 011634 157576
5792
5793
5794
5795
5796
5797 024314 004767 167164
5798 024320 103111
5799
5800
5801
5802 024322 004767 167604
5803 024326 103106
5804
5805
5806
5807
5808
5809
5810 024330 004767 167706
5811
5812
5813
5814
5815
5816
5817
5818
5819 024334 012700 000206
5820 024340 004767 172344

```

```

.SBTTL HARDWARE TEST - FI3QLI -
;*****
; - FIFO 3/4 LEVEL INACTIVE TEST -
;
; THIS TEST VERIFIES THAT THE DUT'S FIFO 3/4 LEVEL ALARM SYSTEM
; REMAINS INACTIVE WHILE IT CONTAINS 191 CHARACTERS OR LESS.
; THE TEST LOOKS FOR AN XOFF (ASCII DC3) CHARACTER IN THE FIFO.
; IF ANY XOFF'S ARE FOUND AN ERROR WILL BE REPORTED AND THE TEST ABORTED.
; ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
; HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
; REPORTED LATER.
; THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
; INTERNAL LOOPBACK MODE.
;*****
      BGNTST
      SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T9::
      MOV #PRI05,R0
      TRAP C#SPRI
      TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (54)
      MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
      MOV #1,ERRTYP ;SET FATAL ERROR TYPE IN ERROR TABLE.
      MOV #5401,ERRNBR ;SET ERROR NUMBER TO 5401.
      MOV #EM5401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
      MOV #ER0503,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
;
; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; THIS SUBROUTINE REPORTS ERROR >>>> 5401 <<<<.
;
      JSR PC,CLNRST ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
      BCC 601 ;EXIT TEST IF FATAL ERROR FOUND.
;
; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
;
      JSR PC,FINACT ;FIND THE NUMBER OF THE FIRST ACTIVE LINE.
      BCC 601 ;EXIT IF NO LINES ARE AVAILABLE.
;
; INITIALIZE THE 256 BYTE DATA PATTERN.
; ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
; NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
;
      JSR PC,INDTPX ;INITIALISE THE DATA PATTERN.
;
; TRANSMIT A 191 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
;
;
; SET INTERNAL LOOPBACK, ENABLE IAUTO AND RX ON THE SELECTED LINE.
; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
;
      MOV #206,R0 ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
      JSR PC,WMLNC ;INITILAISE THE LINE CONTROL REGISTER.

```

## HARDWARE TEST

- FI3QLI -

```

5821 024344 012700 177670      MOV    #177670,R0      ;PASS THE LPR CONTENTS.
5822 024350 004767 172364      JSR    PC,WTWLP      ;SET THE LPR CONTENTS TO 38.4K BAUD.
5823 024354 016704 153432      MOV    10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
5824 024360 004767 167332      JSR    PC,DELAY      ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
5825 024364 012702 002704      MOV    #0BFBAS,R2    ;PASS THE START OF THE DATA PATTERN TO TX.
5826 024370 012703 000277      MOV    #191.,R3      ;PASS THE LENGTH OF THE DATA PATTERN.
5827 024374 004767 167356      JSR    PC,DODMA      ;TRANSMIT THE DATA PATTERN.
5828 024400 103057              BCC    50#           ;IF ERROR FOUND DURING DMA THEN ABORT TEST.
5829
5830
5831                          ;*
5832                          ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
5833                          ; THE FIFO.
5834 024402 005267 157500      ;*
5835 024406 012701 170454      ;*
5836 024412 016702 155624      ;*
5837 024416 004767 172152      ;*
5838 024422 103046              ;*
5839 024424 012704 000005      ;*
5840 024430 004767 167262      ;*
5841                          ;*
5842                          ; READ THE CONTENTS OF THE FIFO. IF ANY OF THE FOLLOWING CONDITIONS OCCUR
5843                          ; REPORT THE ERROR AND ABORT THE TEST;
5844                          ; FIFO EMPTY TOO SOON.
5845                          ; BMP CODE FOUND.
5846                          ; XOFF CODE FOUND.
5847                          ; EXTRA (192) CHARACTER FOUND IN FIFO.
5848                          ;*
5849 024434 005004              ;*
5850 024436 016705 155602      ;*
5851 024442 012767 012267 157436 21:  CLR    R4           ;CLEAR THE CHARACTER COUNT.
5852 024450 011502              ;*
5853 024452 100032              ;*
5854 024454 005204              ;*
5855                          ;*
5856                          ; CHECK IF THE READ CHARACTER IS A BMP CODE.
5857                          ; IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
5858                          ; ABORT THE TEST.
5859                          ;*
5860 024456 005267 157424      ;*
5861 024462 004767 166716      ;*
5862 024466 103001              ;*
5863                          ;*
5864 024470 000421              ;*
5865                          ;*
5866                          ; CHECK IF THE CHARACTER IS AN XOFF. REPORT THE ERROR IF ONE IS FOUND.
5867                          ;*
5868                          ;*
5869 024472 005267 157410      41:   INC    ERRNBR      ;SET ERROR NUMBER TO 5405.
5870 024476 122702 000023      ;*
5871 024502 001003              ;*
5872 024504 012701 006742      ;*
5873                          ;*
5874 024510 000411              ;*
5875                          ;*
5876 024512 005267 157370      ;*
5877 024516 020427 000277      ;*

```



HARDWARE TEST - FI3QLA -

5893  
5894  
5895  
5896  
5897  
5898  
5899  
5900  
5901  
5902  
5903  
5904  
5905  
5906  
5907  
5908  
5909  
5910 024552  
024552  
5911 024552  
024552 012700 000240  
024556 104441  
5912 000012  
5913 024560 012767 000012 155500  
5914 024566 012767 177777 155470  
5915 024574 012767 000001 157302  
5916 024602 012767 012575 157276  
5917 024610 012767 007003 157272  
5918  
5919  
5920  
5921  
5922  
5923 024616 004767 166662  
5924 024622 103402  
5925 024624 000167 000414  
5926  
5927  
5928  
5929 024630 004767 167276  
5930 024634 103402  
5931 024636 000167 000402  
5932  
5933  
5934  
5935  
5936  
5937 024642 004767 167374  
5938  
5939  
5940  
5941  
5942  
5943  
5944  
5945  
5946 024646 005267 157234

```

.SBTTL HARDWARE TEST - FI3QLA -
;*****
;* - FIFO 3/4 LEVEL ACTIVE TEST -
;*
;* THIS TEST VERIFIES THAT THE DUT'S FIFO 3/4 LEVEL ALARM SYSTEM
;* BECOMES ACTIVE WHEN THE FIFO CONTAINS > 192 CHARACTERS.
;* THE TEST COMPARES THE ACTUAL NUMBER OF XOFF (ASCII DC3)
;* CHARACTERS THAT ARE FOUND IN THE FIFO WITH THE EXPECTED NUMBER.
;* AN ERROR WILL BE REPORTED, IF THE COUNTS ARE FOUND TO DIFFER.
;* ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
;* HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
;* REPORTED LATER.
;* THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
;* INTERNAL LOOPBACK MODE.
;*****
BGNTST
                                T10::
SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
                                MOV #PRI05,R0
                                TRAP C#SPRI
TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (55)
MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
MOV #5501,ERRNBR ;SET ERROR NUMBER TO 5501.
MOV #EM5501,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
;
; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; THIS SUBROUTINE REPORTS ERROR >>>> 5501 <<<<.
;
JSR PC,CLNRST ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
BCS .+6 ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
JMP 60# ;EXIT TEST FATAL ERROR FOUND.
;
; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
;
JSR PC,FINACT ;FIND AN ACTIVE LINE.
BCS .+6 ;SKIP EXIT OF TEST IF ACTIVE LINE FOUND.
JMP 60# ;EXIT TEST.
;
; INITIALIZE THE 256 BYTE DATA PATTERN.
; ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
; NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
;
JSR PC,INDTPX ;INITIALISE DATA PATTERN.
;
; TRANSMIT A 256 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
;
;
; SET INTERNAL LOOPBACK, ENABLE IAUTO AND RECEIVER ON THE SELECTED LINE.
; TRANSMIT THE FIRST 191 CHARACTERS ON THE FIRST AVAILABLE ACTIVE LINE.
;
2# INC ERRNBR ;SET ERROR NUMBER TO 5502.

```

## HARDWARE TEST

- FI3QLA -

```

5947 024652 012700 000206      MOV      #206,R0      ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
5948 024656 004767 172026      JSR      PC,WTWLNLC   ;INITIALISE THE LINE CONTROL REGISTER.
5949 024662 012700 177670      MOV      #177670,R0   ;PASS THE LPR CONTENTS.
5950 024666 004767 172046      JSR      PC,WTWLPR    ;SET THE LPR CONTENTS TO 38.4K BAUD.
5951 024672 016704 153114      MOV      10.,R4       ;PASS DELAY TIME OF 10 MILLI SECONDS.
5952 024676 004767 167014      JSR      PC,DELAY     ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
5953 024702 010105                MOV      R1,R5        ;COPY THE LINE NUMBER.
5954 024704 012702 002704      MOV      #8UFBAS,R2   ;PASS THE START OF THE DATA PATTERN TO TX.
5955 024710 012703 000277      MOV      #191.,R3    ;PASS THE LENGTH OF THE DATA PATTERN.
5956 024714 004767 167036      JSR      PC,DODMA     ;TRANSMIT THE DATA PATTERN.
5957 024720 103147                BCC      50#          ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
5958
5959
5960                               ;*
5961                               ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
5962                               ; THE FIFO.
5963                               ;-
5963 024722 005267 157160      INC      ERRNBR       ;SET ERROR NUMBER TO 5503.
5964 024726 012701 170454      MOV      #170454,R1   ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
5965 024732 016702 155304      MOV      CSRA,R2      ;PASS THE ADDRESS OF THE CSR.
5966 024736 004767 171632      JSR      PC,WAIBIS    ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
5967 024742 103136                BCC      50#          ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
5968 024744 012704 000005      MOV      #5,R4        ;PASS DELAY OF 5 MILLI SECS.
5969 024750 004767 166742      JSR      PC,DELAY     ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
5970
5971                               ;*
5972                               ; TRANSMIT A NULL CHARACTER WHICH WILL CAUSE AN XOFF TO BE GENERATED.
5973                               ;-
5973 024754 005267 157126      INC      ERRNBR       ;SET ERROR NUMBER TO 5504.
5974 024760 010501                MOV      R5,R1        ;PASS THE LINE NUMBER.
5975 024762 012702 002704      MOV      #8UFBAS,R2   ;PASS THE START OF THE DATA PATTERN TO TX.
5976 024766 012703 000001      MOV      #1,R3        ;PASS THE NUMBER OF
5977 024772 004767 166760      JSR      PC,DODMA     ;TX A NULL CHARACTER TO CAUSE AN XOFF.
5978 024776 103120                BCC      50#          ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
5979
5980                               ;*
5981                               ; WAIT FOR THE XOFF TO BE RECEIVED BEFORE TX THE NEXT 42 CHARACTERS
5982                               ; WHICH WILL CAUSE A FURTHER 21 XOFF'S TO BE GENERATED.
5983                               ;-
5984 025000 005267 157102      INC      ERRNBR       ;SET ERROR NUMBER TO 5505.
5985 025004 012701 170012      MOV      #170012,R1   ;PASS TIME-OUT VALUE OF 10 MILLI SECS.
5986 025010 016702 155226      MOV      CSRA,R2      ;PASS THE ADDRESS OF THE CSR.
5987 025014 004767 171554      JSR      PC,WAIBIS    ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
5988 025020 103107                BCC      50#          ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
5989 025022 012704 000005      MOV      #5,R4        ;PASS DELAY OF 5 MILLI SECS.
5990 025026 004767 166664      JSR      PC,DELAY     ;WAIT FOR XOFF TO GET INTO THE FIFO.
5991
5992                               ;*
5993                               ; INITIALISE THE 256 BYTE DATA PATTERN TO ALL NULLS.
5994                               ;-
5994 025032 012702 002704      MOV      #8UFBAS,R2   ;INITIALIZE THE DATA PATTERN TO BE
5995 025036 105022                CLRB     (R2)+        ; ALL NULLS.
5996 025040 020227 003304      CMP      R2,#8UFBAS  ;
5997 025044 103774                BLO     4#           ;
5998
5999
6000                               ;*
6001                               ; TRANSMIT A FURTHER 31 NULL CHARACTERS WHICH WILL CAUSE 31 XOFF'S TO BE
6002                               ; GENERATED.
6003                               ;-
6003 025046 005267 157034      INC      ERRNBR       ;SET ERROR NUMBER TO 5506.

```



HARDWARE TEST

- FI3QLA -

```

6004 025052 010501          MOV    R5,R1          ;PASS THE LINE NUMBER.
6005 025054 012702 002704  MOV    #BUFBAS,R2     ;PASS THE START OF THE DATA PATTERN TO TX.
6006 025060 012703 000037  MOV    #31.,R3        ;PASS THE LENGTH OF THE DATA PATTERN.
6007 025064 004767 166666  JSR    PC,DODMA       ;TRANSMIT THE DATA PATTERN.
6008 025070 103063          BCC    50#            ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6009
6010          ;+
6011          ; WAIT FOR THE XOFF'S AND THE NULL CHARACTERS TO BE RECEIVED.
6012 025072 005267 157010  ;-
6013 025076 012701 170454  INC    ERRNBR         ;SET ERROR NUMBER TO 5507.
6014 025102 016702 155134  MOV    #170454,R1     ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6015 025106 004767 171462  MOV    CSRA,R2        ;PASS THE ADDRESS OF THE CSR.
6016 025112 103052          JSR    PC,WAIBIS      ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6017 025114 012704 000005  BCC    50#            ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6018 025120 004767 166572  MOV    #5,R4          ;PASS DELAY OF 5 MILLI SECS.
6019          JSR    PC,DELAY      ;WAIT FOR XOFF TO GET INTO THE FIFO.
6020          ;+
6021          ; READ THE FIFO UNTIL EMPTY, COUNTING THE NUMBER OF XOFF CHARACTERS
6022          ; THAT ARE FOUND.
6023 025124 005004          ;-
6024 025126 005003          CLR    R4             ;CLEAR CHARACTER COUNTER.
6025 025130 012701 170001  CLR    R3             ;CLEAR THE XOFF FOUND COUNTER.
6026 025134 012767 012604 156744 6# MOV    #170001,R1     ;INDICATE TO TEST DATA.VALID BIT, TIME-OUT 1MS.
6027 025142 016702 155076  MOV    #5508.,ERRNBR ;SET UP ERROR NUMBER EACH TIME AROUND THE LOOP.
6028 025146 004767 171422  MOV    RBUFA,R2       ;INDICATE TO CHECK RECEIVE BUFFER REGISTER.
6029 025152 103032          JSR    PC,WAIBIS      ;WAIT FOR RECEIVED CHAR OR TIME-OUT.
6030 025154 005204          BCC    50#            ;GO REPORT ERROR IF FIFO EMPTY.
6031          INC    R4             ;COUNT THE CHARACTER.
6032          ;+
6033          ; CHECK IF FOR BMP CODES IN THE FIFO, ABORT THE TEST IF ANY ARE FOUND.
6034          ; SAVE THE BMP CODE ON THE QUEUE TO BE REPORTED LATER.
6035 025156 005267 156724  ;-
6036 025162 004767 166216  INC    ERRNBR         ;SET ERROR NUMBER TO 5509.
6037 025166 103422          JSR    PC,CHKBMP      ;CHECK IF WE HAVE GOT A BMP CODE.
6038          BCS    12#            ;GO REPORT THE ERROR IF WE FOUND A BMP CODE.
6039          ;+
6040          ; CHECK FOR XOFF CHARACTER.
6041 025170 122702 000023 8# CMPB   #23,R2        ;CHECK IF THE RECEIVED CHARACTER WAS AN XOFF.
6042 025174 001001          BNE    10#            ;BRANCH IF CHARACTER WAS NOT AN XOFF.
6043 025176 005203          INC    R3             ;INCREMENT XOFF FOUND COUNT.
6044
6045          ;+
6046          ; CHECK IF ALL THE CHARACTERS INCLUDING THE XON HAVE BEEN REMOVED.
6047 025200 020427 000400 10# CMP     R4,#256.      ;CHECK IF WE HAVE REMOVED ALL THE CHARACTERS.
6048 025204 002753          BLT    6#            ;GO GET THE NEXT CHAR IF WE HAVE NOT FINISHED.
6049
6050          ;+
6051          ; CHECK IF THE CORRECT NUMBER OF XOFF'S WERE FOUND IN THE FIFO,
6052          ; REPORT ERROR IF COUNT IS INCORRECT.
6053          ;-
6054 025206 016767 165374 156672 MOV    5510.,ERRNBR   ;SET UP THE ERROR NUMBER TO 5510.
6055 025214 022703 000040  CMP    #32.,R3        ;COMPARE EXPECTED XOFF COUNT WITH ACTUAL COUNT.
6056 025220 001411          BEQ    60#            ;EXIT TEST IF SUCCESS.
6057 025222 012767 011634 156662 MOV    #ER0503,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6058 025230 012701 006742  MOV    #EM5402,R1     ;PASS THE MESSAGE TO BE REPORTED.
6059          ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
6060 025234          12# ERROR      ; >>>>> ERROR <<<<<.

```



HARDWARE TEST

- FI3QAI -

```

6068 .SBTTL HARDWARE TEST - FI3QAI -
6069 ;*****
6070 ;*
6071 ;* - FIFO 3/4 ALARM LEVEL ACTIVE/INACTIVE TEST -
6072 ;*
6073 ;* THIS TEST VERIFIES THAT THE DUT'S FIFO 3/4 LEVEL ALARM SYSTEM
6074 ;* BECOMES ACTIVE AND INACTIVE AT THE CORRECT LEVELS.
6075 ;* ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
6076 ;* HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
6077 ;* REPORTED LATER.
6078 ;* THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
6079 ;* INTERNAL LOOPBACK MODE.
6080 ;*
6081 ;*****
6082 BGNTST
6083 025252 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T11::
6084 025252 012700 000240 ;
6085 025252 104441 ;
6086 025252 000013 ;
6087 025260 012767 000013 155000 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6088 025266 012767 177777 154770 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (56)
6089 025274 012767 000001 156602 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
6090 025302 012767 012741 156576 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6091 025310 012767 007040 156572 MOV #5601,ERRNBR ;SET ERROR NUMBER TO 5601.
6092 ; MOV #EM5601,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
6093 ;
6094 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
6095 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6096 ; THIS SUBROUTINE REPORTS ERROR >>>> 5601 <<<<<.
6097 ;
6098 025316 004767 166162 JSR PC,CLNRST ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
6099 025322 103402 BCS 2# ;SKIP EXITING TEST A SUCCESSFUL RESET.
6100 025324 000167 000412 JMP 60# ;EXIT THIS TEST.
6101 ;
6102 ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
6103 ;
6104 025330 004767 166576 JSR PC,FINACT ;FIND AN ACTIVE LINE.
6105 025334 103402 BCS .+6 ;SKIP EXIT OF TEST IF ACTIVE LINE FOUND.
6106 025336 000167 000400 JMP 60# ;EXIT TEST.
6107 ;
6108 ; INITIALIZE THE 256 BYTE DATA PATTERN.
6109 ; ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
6110 ; NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
6111 ;
6112 025342 004767 166674 JSR PC,INDTPX ;INITIALISE THE DATA PATTERN.
6113 ;
6114 ; TRANSMIT A 256 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
6115 ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
6116 ;
6117 ; SET INTERNAL LOOPBACK, ENABLE IAUTO AND RECEIVER ON THE SELECTED LINE.
6118 ; TRANSMIT THE FIRST 191 CHARACTERS ON THE FIRST AVAILABLE ACTIVE LINE.
6119 ;
6120 025346 005267 156534 INC ERRNBR ;SET ERROR NUMBER TO 5602.
6121 025352 012700 000206 MOV #206,RO ;PASS INTERNAL LOPBCK, ENABLE RX AND IAUTO.
6122 025356 004767 171326 JSR PC,WTLNC ;INITILAISE THE LINE CONTROL REGISTER.
    
```

## HARDWARE TEST

- FI3QAI -

```

6122 025362 012700 177670      MOV    #177670,R0      ;PASS THE LPR CONTENTS.
6123 025366 004767 171346      JSR    PC,WTWLP      ;SET THE LPR CONTENTS TO 38.4K BAUD.
6124 025372 016704 152414      MOV    10.,R4         ;PASS DELAY TIME OF 10 MILLI SECONDS.
6125 025376 004767 166314      JSR    PC,DELAY      ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6126 025402 010105              MOV    R1,R5         ;COPY THE LINE NUMBER.
6127 025404 012702 002704      MOV    #BUFBA5,R2    ;PASS THE START OF THE DATA PATTERN TO TX.
6128 025410 012703 000277      MOV    #191.,R3     ;PASS THE LENGTH OF THE DATA PATTERN.
6129 025414 004767 166336      JSR    PC,DODMA     ;TRANSMIT THE DATA PATTERN.
6130 025420 103146              BCC    50#          ;EXIT IF ERROR FOUND DURING DMA TX.
6131
6132      ;+
6133      ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
6134      ; THE FIFO.
6135 025422 005267 156460      INC    ERRNBR        ;SET ERROR NUMBER TO 5603.
6136 025426 012701 170454      MOV    #170454,R1   ;PASS TIME-OUT VALUE OF 300 MILLI SECS.
6137 025432 016702 154604      MOV    CSRA,R2      ;PASS THE ADDRESS OF THE CSR.
6138 025436 004767 171132      JSR    PC,WAIBIS    ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6139 025442 103135              BCC    50#          ;BRANCH IF FIFO EMPTY, ABORT THE TEST.
6140 025444 012704 000005      MOV    #5,R4        ;PASS DELAY OF 5 MILLI SECS.
6141 025450 004767 166242      JSR    PC,DELAY     ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6142
6143      ;+
6144      ; TRANSMIT A NULL CHARACTER WHICH WILL CAUSE AN XOFF TO BE GENERATED.
6145      ;-
6146 025454 005267 156426      INC    ERRNBR        ;SET ERROR NUMBER TO 5604.
6147 025460 010501              MOV    R5,R1        ;PASS THE LINE NUMBER.
6148 025462 012702 002704      MOV    #BUFBA5,R2   ;PASS THE START OF THE DATA PATTERN TO TX.
6149 025466 012703 000001      MOV    #1,R3        ;PASS THE NUMBER OF
6150 025472 004767 166260      JSR    PC,DODMA     ;TX A NULL CHARACTER TO CAUSE AN XOFF.
6151 025476 103117              BCC    50#          ;ABORT THE TEST IF ERROR FOUND DURING DMA TX.
6152
6153      ;+
6154      ; WAIT FOR THE XOFF TO BE RECEIVED BEFORE CONTINUING THE TEST.
6155      ;-
6155 025500 005267 156402      INC    ERRNBR        ;SET ERROR NUMBER TO 5605.
6156 025504 012701 170012      MOV    #170012,R1   ;PASS TIME-OUT VALUE OF 10 MILLI SECS.
6157 025510 016702 154526      MOV    CSRA,R2      ;PASS THE ADDRESS OF THE CSR.
6158 025514 004767 171054      JSR    PC,WAIBIS    ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6159 025520 103106              BCC    50#          ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6160 025522 012704 000005      MOV    #5,R4        ;PASS DELAY OF 5 MILLI SECS.
6161 025526 004767 166164      JSR    PC,DELAY     ;WAIT FOR XOFF TO GET INTO THE FIFO.
6162
6163 025532 010577 154504      MOV    R5,@CSRA     ;SELECT THE LINE READY FOR TRANSMISSION.
6164
6165      ;+
6166      ; READ THREE CHARACTERS, TRANSMIT ONE CHARACTER UNTIL THE FIRST 192 CHARACTERS
6167      ; HAVE BEEN READ FROM THE FIFO, IE UNTIL THE HALF LEVEL IS REACHED.
6168      ; THEN READ THE FIFO UNTIL EMPTY.
6169      ; COUNT ALL XOFF'S THAT ARE DETECTED.
6170 025536 005005              CLR    R5           ;CLEAR THE TX FLAG.
6171 025540 005004              CLR    R4           ;CLEAR THE CHARACTER COUNTER.
6172 025542 012703 000300      MOV    #192.,R3     ;SET UP READ COUNTER FOR THE FIRST 192 CHARS.
6173
6174 025546 012700 000003      4# : MOV    #3,R0      ;SET READ COUNTER.
6175 025552 012701 170005      6# : MOV    #170005,R1 ;INDICATE TO TEST DATA.VALID BIT, TIME-OUT SMS.
6176 025556 016702 154462      MOV    RBUFA,R2    ;INDICATE TO CHECK RECEIVE BUFFER REGISTER.
6177 025562 004767 171006      JSR    PC,WAIBIS    ;WAIT FOR RECEIVED CHAR OR TIME-OUT.
6178 025566 103046              BCC    14#         ;EXIT LOOP IF TIME-OUT, FIFO EMPTY.

```

HARDWARE TEST

- FI3QAI -

```

6179 025570 005300          DEC R0          ;DECREMENT READ COUNTER.
6180 025572 005303          DEC R3          ;DECREMENT CHAR COUNTER.
6181 025574 003002          BGT 8#         ;SKIP DISBL'G TX IF FIRST 192 CHARS NOT READ.
6182 025576 052705 100000  BIS #BIT15,R5   ;DISABLE ANY FURTHER TRANSMISSIONS.
6183
6184          ;+
6185          ; CHECK IF THE READ CHARACTER IS A BMP CODE.
6186          ; IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
6187          ; ABORT THE TEST.
6188 025602 012767 012746 156276 8# :   MOV #5606.,ERRNBR ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
6189 025610 004767 165570          JSR PC,CHKBMP  ;CHECK IF CHARACTER IS A BMP CODE.
6190 025614 103446          BCS 16#        ;GO REPORT ERROR AND ABORT TEST IF BMP FOUND.
6191
6192          ;+
6193          ; CHECK FOR XOFF CHARACTER. IF ONE IS FOUND, COUNT IT.
6194          ; TRANSMIT A NULL CHARACTER UNTIL THE FIRST 192 CHARS HAVE BEEN READ.
6195 025616 122702 000023 10# :   CMPB #23,R2    ;CHECK IF THE RECEIVED CHARACTER WAS AN XOFF.
6196 025622 001001          BNE 12#        ;BRANCH IF CHARACTER WAS NOT AN XOFF.
6197 025624 005204          INC R4         ;INCREMENT THE XOFF CHAR FOUND COUNTER.
6198
6199 025626 005700          12# :   TST R0         ;CHECK READ COUNT, TO SEE IF A CHAR CAN BE TX.
6200 025630 001350          BNE 6#         ;BRANCH IF 3 CHARS HAVE NOT YET BEEN READ.
6201 025632 005705          TST R5         ;CHECK THE TRANSMISSION ENABLED FLAG.
6202 025634 100744          BMI 4#         ;SKIP TRANSMITTING A CHARACTER IF TX DISABLED.
6203 025636 012777 100000 154400  MOV #100000,@TXCHA ;TX A NULL CHARACTER.
6204 025644 010446          MOV R4,-(SP)   ;SAVE THE XOFF COUNT ON THE STACK.
6205
6206          ;+
6207          ; WAIT FOR THE CHARACTER TO BE RECEIVED BEFORE CONTINUING THE TEST.
6208 025646 005267 156234          INC ERRNBR    ;SET ERROR NUMBER TO 5607.
6209 025652 012701 170012          MOV #170012,R1 ;PASS TIME-OUT VALUE OF 10 MILLI SECS.
6210 025656 016702 154360          MOV CSRA,R2   ;PASS THE ADDRESS OF THE CSR.
6211 025662 004767 170706          JSR PC,WAIBIS ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6212 025666 103023          BCC 50#       ;IF NO TX_ACTION WAS RECEIVED, ABORT THE TEST.
6213 025670 012704 000005          MOV #5,R4     ;PASS DELAY OF 5 MILLI SECS.
6214 025674 004767 166016          JSR PC,DELAY  ;WAIT FOR XOFF TO GET INTO THE FIFO.
6215 025700 012604          MOV (SP)+,R4  ;RESTORE THE XOFF COUNT.
6216 025702 000721          BR 4#        ;GO RESET THE READ COUNT AND GET NEXT CHAR.
6217
6218          ;+
6219          ; CHECK IF THE CORRECT NUMBER OF XOFF'S WERE FOUND IN THE FIFO
6220          ; REPORT ERROR IF COUNT IS INCORRECT.
6221
6222 025704 012767 012750 156174 14# :   MOV #5608.,ERRNBR ;SET ERROR NUMBER TO 5608.
6223 025712 020427 000077          CMP R4,#63.   ;COMPARE THE EXPECTED AND ACTUAL XOFF COUNTS.
6224 025716 001411          BEQ 60#       ;EXIT TEST IF SUCCESS.
6225 025720 012767 011634 156164  MOV #ER0503,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6226 025726 012701 006742          MOV #EM5402,R1 ;PASS THE MESSAGE TO BE REPORTED.
6227          ;REPORT THE ERROR "FIFO BAD, ALARM SIGNAL DEFECTIVE".
6228 025732          16# :   ERROR          ;
6229 025734 104460          TRAP C#ERROR
6230 025736 000402          BR 60#       ;EXIT THIS TEST.
6231 025742 004767 170036 50# :   JSR PC,TSABRT ;REPORT TEST ABORTED. ERROR # INDICATES FAULT.
6232 025744 005067 154316 60# :   CLR CTRLCF  ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6233
6234 025746          ENDTST
    
```

N12

HARDWARE TEST - FI3QAI -

025746  
025746 104401

L10035: TRAP C\$ETST

HARDWARE TEST - FIMAVL -

6236  
6237  
6238  
6239  
6240  
6241  
6242  
6243  
6244  
6245  
6246  
6247  
6248  
6249  
6250 025750  
025750  
6251 025750  
025750 012700 000240  
025754 104441  
6252 000014  
6253 025756 012767 000014 154302  
6254 025764 012767 177777 154272  
6255 025772 012767 000001 156104  
6256 026000 012767 013105 156100  
6257 026006 012767 007106 156074  
6258 026014 012767 011634 156070  
6259  
6260  
6261  
6262  
6263  
6264 026022 004767 165456  
6265 026026 103402  
6266 026030 000167 000360  
6267 026034  
6268  
6269  
6270  
6271 026034 004767 166072  
6272 026040 103165  
6273  
6274  
6275  
6276  
6277  
6278 026042 004767 166174  
6279  
6280  
6281  
6282  
6283  
6284  
6285  
6286  
6287  
6288 026046 005267 156034  
6289 026052 004767 167576

```

.SBTTL HARDWARE TEST - FIMAVL -
.....
; FIFO HALF LEVEL ACTIVE/INACTIVE TEST
;
; THIS TEST CHECKS THAT THE DUT'S FIFO HALF LEVEL ALARM SYSTEM
; BECOMES ACTIVE AND INACTIVE AT THE CORRECT LEVELS.
; ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
; HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
; REPORTED LATER.
; THE CHARACTERS ARE TRANSMITTED ON THE FIRST AVAILABLE ACTIVE LINE, IN
; INTERNAL LOOPBACK MODE.
.....

BGNTST
;
; SETPRI @PRIOS ;ALLOW LTC INTERRUPTS. T12::
;
; MOV @PRIOS,RO ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
; TRAP C@SPRI ;SET UP THE TEST NUMBER. (57)
;
; TNUM == TNUM + 1 ;INDICATE THAT WE ARE IN A TEST.
; MOV @TNUM,TSTNUM ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
; MOV @-1,CTRLCF ;SET ERROR NUMBER TO 5701.
; MOV @1,ERRTYP ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
; MOV @5701,ERRNBR ;SELECT THE ERROR REPORTING ROUTINE.
; MOV @EM5701,ERRMSG
; MOV @ER0503,ERRBLK
;
; *
; * RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; * CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; * THIS SUBROUTINE REPORTS ERROR >>>> 5701 <<<<<.
; *
; * JSR PC,CLRST ;RESET THE DMV-11, REPORT ANY ERRORS FOUND.
; * BCS 28 ;SKIP EXITING TEST A SUCCESSFUL RESET.
; * JMP 608 ;EXIT THIS TEST.
; *
; *
; * 28:
; *
; * FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
; *
; * JSR PC,FINACT ;FIND AN ACTIVE LINE.
; * BCC 608 ;EXIT IF NO ACTIVE LINES AVAILABLE.
; *
; *
; * INITIALIZE THE 256 BYTE DATA PATTERN.
; * ENSURE THE DATA PATTERN IS FREE FROM XON'S OR XOFF'S TO PREVENT ERRORS.
; * NOTE: THE FIRST TWO CHARACTERS AND THE LAST TWO CHARACTERS WILL BE THE SAME.
; *
; * JSR PC,INDTPX ;INITIALISE THE DATA PATTERN.
; *
; *
; * *
; * * FILL THE FIFO BY TRANSMITTING 225 CHARS (IE 225 + 31 XOFF'S).
; * * TRANSMIT DATA PATTERN USING DMA, ON A SINGLE CHANNEL
; * * AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
; * *
; *
; * *
; * * SET INTERNAL LOOPBACK, ENABLE IAUTO AND RECEIVER ON THE SELECTED LINE.
; * * TRANSMIT THE 225 CHARACTERS ON THE FIRST AVAILABLE ACTIVE LINE.
; * *
; *
; * INC ERRNBR ;SET ERROR NUMBER TO 5702.
; * JSR PC,SETPAR ;SET UP PARAMETERS FOR TRANSMISSION.

```

HARDWARE TEST - FIMAVL -

6290	026056	012700	000341		MOV	#225.,R0	;	PASS LENGTH OF DATA PATTERN.
6291	026062	004767	170024		JSR	PC, TXDATP	;	TRANSMIT DATA PATTERN.
6292	026066	103150			BCC	50#	;	EXIT IF ERROR FOUND DURING TX.
6293	026070	010105			MOV	R1,R5	;	COPY THE LINE NUMBER.
6294							;	
6295							;	
6296							;	
6297							;	
6298	026072	005267	156010				;	
6299	026076	004767	170546		INC	ERRNBR	;	SET ERROR NUMBER TO 5703.
6300	026102	103142			JSR	PC, WAITTX	;	WAIT FOR TRANSMISSION TO COMPLETE.
6301					BCC	50#	;	GO REPORT ERROR IF TX FAILED TO COMPLETE.
6302							;	
6303							;	
6304							;	
6305							;	
6306	026104	005267	155776				;	
6307	026110	012700	000202		INC	ERRNBR	;	SET ERROR NUMBER TO 5704.
6308	026114	004767	167166		MOV	#130.,R0	;	PASS THE NUMBER OF CHARS TO READ.
6309	026120	103133			JSR	PC, READBX	;	READ THE FIRST 130 CHARS FROM THE FIFO.
6310	026122	005267	155760		BCC	50#	;	GO REPORT ERROR IF BMP CODE FOUND.
6311	026126	005701			INC	ERRNBR	;	SET ERROR NUMBER TO 5705.
6312	026130	001125			TST	R1	;	CHECK IF AN XON WAS FOUND.
6313					BNE	40#	;	GO REPORT ERROR IF AN XON WAS FOUND.
6314							;	
6315							;	
6316							;	
6317	026132	010577	154104				;	
6318	026136	012777	100000	154100	MOV	R5, BCSRA	;	SELECT THE LINE READY FOR TRANSMISSION.
6319	026144	005267	155736		MOV	#100000, @TXCHA	;	TRANSMIT A NULL CHARACTER.
6320	026150	004767	170474		INC	ERRNBR	;	SET ERROR NUMBER TO 5706.
6321	026154	103115			JSR	PC, WAITTX	;	WAIT FOR TX TO COMPLETE.
6322					BCC	50#	;	GO REPORT ERROR IF TX DID NOT COMPLETE.
6323							;	
6324							;	
6325	026156	005267	155724				;	
6326	026162	012700	000003		INC	ERRNBR	;	SET ERROR NUMBER TO 5707.
6327	026166	004767	167114		MOV	#3,R0	;	SET THE READ COUNT TO 3.
6328	026172	103106			JSR	PC, READBX	;	READ 3 CHARACTERS FROM THE FIFO.
6329	026174	005267	155706		BCC	50#	;	GO REPORT ERROR IF FIFO EMPTY.
6330	026200	005701			INC	ERRNBR	;	SET ERROR NUMBER TO 5708.
6331	026202	001100			TST	R1	;	CHECK IF AN XON WAS FOUND.
6332					BNE	40#	;	GO REPORT ERROR IF AN XON WAS FOUND.
6333							;	
6334							;	
6335	026204	012700	000076				;	
6336	026210	010501			MOV	#62.,R0	;	PASS LENGTH OF DATA PATTERN.
6337	026212	005267	155670		MOV	R5,R1	;	PASS THE LINE NUMBER.
6338	026216	004767	167670		INC	ERRNBR	;	SET ERROR NUMBER TO 5709.
6339	026222	103072			JSR	PC, TXDATP	;	TRANSMIT DATA PATTERN.
6340					BCC	50#	;	EXIT IF ERROR FOUND DURING TX.
6341							;	
6342							;	
6343							;	
6344							;	
6345	026224	005267	155656				;	
6346	026230	004767	170414		INC	ERRNBR	;	SET ERROR NUMBER TO 5710.
					JSR	PC, WAITTX	;	WAIT FOR TX TO COMPLETE.



HARDWARE TEST - FIMAVL -

```

6347 026234 103065          BCC 500          ;GO REPORT ERROR IF TX FAILED TO COMPLETE.
6348
6349
6350
6351
6352
6353 026236 005267 155644          INC  ERRNBR          ;SET ERROR NUMBER TO 5711.
6354 026242 012700 000176          MOV  #126.,R0        ;SET UP READ COUNTER.
6355 026246 004767 167034          JSR  PC,READBX       ;READ THE FIRST 126 CHARS.
6356 026252 103056          BCC 500          ;GO REPORT THE ERROR IF FIFO EMPTY.
6357 026254 005267 155626          INC  ERRNBR          ;SET ERROR NUMBER TO 5712.
6358 026260 005701          TST  R1             ;CHECK IF AN XON WAS FOUND.
6359 026262 001050          BNE  400          ;GO REPORT ERROR IF AN XON WAS FOUND.
6360 026264 005267 155616          INC  ERRNBR          ;SET ERROR NUMBER TO 5713.
6361 026270 012701 006742          MOV  #EM5402,R1     ;PASS THE MESSAGE TO BE REPORTED.
6362 026274 016703 153744          MOV  RBUFA,R3       ;GET THE RECEIVER BUFFER ADDRESS.
6363 026300 011302          MOV  (R3),R2        ;READ THE NULL CHARACTER FROM THE FIFO.
6364 026302 120227 000000          CMPB R2,#000        ;CHECK IF IT IS A NULL CHARACTER.
6365 026306 001036          BNE  400          ;GO REPORT THE ERROR IF NOT THE SAME.
6366 026310 005267 155572          INC  ERRNBR          ;SET ERROR NUMBER TO 5714.
6367 026314 011302          MOV  (R3),R2        ;READ THE XOFF FROM THE FIFO.
6368 026316 120227 000023          CMPB R2,#23        ;CHECK IF THE READ CHAR IS AN XOFF.
6369 026322 001030          BNE  400          ;GO REPORT THE ERROR IF NOT THE SAME.
6370 026324 011302          MOV  (R3),R2        ;READ THE XON FROM THE FIFO.
6371 026326 005267 155554          INC  ERRNBR          ;SET ERROR NUMBER TO 5715.
6372 026332 120227 000021          CMPB R2,#21        ;CHECK IF THE READ CHARACTER IS AN XON.
6373 026336 001022          BNE  400          ;GO REPORT THE ERROR IF NOT THE SAME.
6374 026340 005267 155542          INC  ERRNBR          ;SET ERROR NUMBER TO 5716.
6375 026344 011302          MOV  (R3),R2        ;READ THE NULL CHARACTER FROM THE FIFO.
6376 026346 120227 000000          CMPB R2,#000        ;CHECK IF IT IS A NULL CHARACTER.
6377 026352 001014          BNE  400          ;GO REPORT THE ERROR IF NOT THE SAME.
6378
6379
6380
6381
6382 026354 012700 000075          ;*
6383 026360 005267 155522          ; READ THE REMAINING CHARACTERS FROM THE FIFO.
6384 026364 004767 166716          ;-
6385 026370 103007          600:  MOV  #61.,R0        ;SET UP READ COUNTER.
6386 026372 005267 155510          INC  ERRNBR          ;SET ERROR NUMBER TO 5717.
6387 026376 005701          JSR  PC,READBX       ;READ THE FIRST 61 CHARS.
6388 026400 001001          BCC 500          ;GO REPORT THE ERROR IF FIFO EMPTY.
6389 026402 000404          INC  ERRNBR          ;SET ERROR NUMBER TO 5718.
6390 026404          TST  R1             ;CHECK IF AN XON WAS FOUND.
6391 026406 000402          BNE  400          ;GO REPORT ERROR IF AN XON WAS FOUND.
6392          BR   600          ;EXIT THE TEST.
6393 026410 004767 167364          400:  ERROR          ;
6394 026414 005067 153644          BR   600          ;EXIT THE TEST.
6395          BR   600          ;EXIT THE TEST.
6396 026420          JSR  PC,TSABRT       ;REPORT TEST ABORTED. ERROR # INDICATES FAULT.
026420          CLR  CTRLCF        ;INDICATE THAT WE ARE NOT WITHIN A TEST.
026420 104401          600:  ENDTST
L10036:  TRAP  C0ETST

```

HARDWARE TEST - BREAKB -

```

6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408
6409 026422
      026422
6410
6411 026422 012767 177777 153634
6412          000015
6413 026430 012767 000015 153630
6414 026436 012767 000001 155440
6415 026444 012767 014401 155434
6416 026452 012767 007154 155430
6417
6418
6419
6420
6421
6422 026460 004767 165020
6423 026464 103165
6424
6425
6426
6427
6428
6429 026466 004767 167632
6430 026472 004767 167004
6431 026476 012705 000377
6432 026502 012700 000200
6433 026506 004767 170176
6434 026512 012704 000012
6435 026516 004767 165174
6436
6437
6438
6439
6440 026522 012700 156430
6441 026526 004767 170206
6442
6443
6444
6445 026532 016705 153476
6446 026536 004767 167466
6447
6448
6449
6450
6451 026542 005267 155340
6452 026546 004767 166342
6453 026552 103132

```

```

.SBTTL HARDWARE TEST - BREAKB -
;*****
;* - BREAK GENERATION TEST -
;* THIS TEST VERIFIES THAT ALL SERIAL TRANSMIT LINES CAN GENERATE A BREAK
;* BY SETTING THE BRK BIT IN THE ASSOCIATED LNCTRL REGISTER.
;* USE OF THE INTERNAL LOOPBACK FEATURE OF THE DUARTS IS MADE TO MINIMISE
;* ANY EXTERNAL EFFECTS CAUSED ON THE SERIAL LINES BY THIS TEST.
;* FRAMING ERROR DETECTION IS USED TO INDICATE THE PRESENCE OF A BREAK,
;* BY SETTING THE APPROPRIATE BIT IN THE RBUF REGISTER.
;*****
      BGNTST
      T13::
      MOV    #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
      TNUM  == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV    #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (64)
      MOV    #1,ERRTYP  ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
      MOV    #6401,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
      MOV    #EM6401,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERTBL.
;
; *
; * RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; * CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; * THIS SUBROUTINE REPORTS ERROR >>>> 6401 <<<<.
; *
      JSR    PC,CLNRST ;RESET THE DHV-11. REPORT ANY ERRORS FOUND.
      BCC    600 ;EXIT TEST IF FATAL ERROR FOUND.
;
; *
; * SET UP DEVICE UNDER TEST (DUT) TO:
; * DISABLE TRANSMISSION AND RECEPTION INTERRUPTS.
; * DELAY FOR 10 MILLI-SECONDS TO ALLOW TIME TO CLEAR ANY BREAKS.
; *
      JSR    PC,TXIEO ;DISABLE TRANSMISSION INTERRUPTS.
      JSR    PC,RXIEO ;DISABLE RECEPTION INTERRUPTS.
      MOV    #MAPLNS,R5 ;PASS ACTIVE LINE BIT MAP.
      MOV    #200,R0 ;PASS INTERNAL LOOPBACK MODE.
      JSR    PC,WTMLNC ;SELECT INTERNAL LOOPBACK,DISABLE DMA.
      MOV    #10,,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
      JSR    PC,DELAY ;DELAY TO ALLOW ANY BREAKS TO BE CLEARED.
;
; *
; * SET UP TRANSMISSION AN RECEPTION PARAMETERS FOR ALL LINES.
; * 9600 BAUD,8 CHAR,1 STOPBIT,NO PARITY.
; *
      MOV    #156430,R0 ;SET UP BAUD RATE,ETC.
      JSR    PC,WTMLPR ;SET COMMUNICATION PARAMETERS ON ALL LINES.
;
; *
; * ENABLE TRANSMITTERS ON ALL ACTIVE LINES.
; *
      MOV    ACTLNS,R5 ;PASS ACTIVE LINE BIT MAP.
      JSR    PC,TXENBL ;ENABLE TRANSMISSIONS ON ALL LINES.
;
; *
; * PURGE THE FIFO OF ANY UNWANTED CHARACTERS.
; * THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>> 6402 THRU 6404 <<<<.
; *
      INC    ERRNBR ;SET ERROR NUMBER TO 6402.
      JSR    PC,PUFIFR ;PURGE FIFO.
      BCC    600 ;ABORT TEST IF FIFO WILL NOT CLEAR.

```

HARDWARE TEST

- BREAKB -

```

6454
6455
6456
6457
6458
6459
6460
6461 026554 005002
6462 026556 012703 000001
6463 026562 030367 153446
6464 026566 001434
6465 026570 012700 000200
6466 026574 004767 170110
6467 026600 012704 000012
6468 026604 004767 165106
6469
6470
6471
6472
6473
6474
6475 026610 010305
6476 026612 012700 000214
6477 026616 004767 170066
6478
6479
6480
6481
6482 026622 012704 000005
6483 026626 004767 165064
6484 026632 017700 153406
6485 026636 032700 020000
6486 026642 001006
6487 026644 012701 007203
6488
6489 026650
        026650 104455
        026652 014405
        026654 007154
        026656 011740
6490 026660 006303
6491 026662 005202
6492 026664 020227 000010
6493 026670 001334
6494
6495
6496
6497
6498
6499
6500
6501
6502 026572 012705 000377
6503 026676 012700 000200
6504 026702 004767 170002
6505 026706 012704 000012
6506 026712 004767 165000
;
; *
; VERIFY BREAK GENERATION ON INDIVIDUAL LINES.
; CLEAR BREAKS ON ALL LINES.
; DELAY FOR 10 MILLI-SECONDS TO ALLOW TIME FOR ANY BREAKS TO BE CLEARED.
; SELECT LINE,SET BREAK BIT IN LNCYRL REGISTER.
; TEST FOR A CHARACTER IN THE FIFO WITH FRAME ERROR.
;
; -
20: CLR R2 ;CLEAR LINE COUNTER.
;
;
40: MOV #1,R3 ;SET UP ACTIVE LINE BIT MASK.
; BIT R3,ACTLNS ;CHECK IF THIS LINE IS ACTIVE.
; BEQ B0 ;GO SELECT NEXT LINE IF THIS ONE IS INACTIVE.
; MOV #200,R0 ;SET UP PARAMETER TO CLEAR BREAK BITS.
; JSR PC,WTWLN ;CLEAR BREAK BIT,RESELECT INTERNAL LOOPBACK.
; MOV #10.,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
; JSR PC,DELAY ;DELAY TO ALLOW BREAKS TO BE CLEARED.
;
; *
; SET BREAK BIT ON SELECTED LINE.
; SET UP PARAMETERS TO TEST FOR THE FRAME ERROR BIT SET IN RBUF.
; TIME-OUT = 5 MILLI SECONDS.
; CALL ROUTINE TO CHECK FOR CONDITION FOUND.
;
; -
60: MOV R3,R5 ;COPY ACTIVE LINE BIT MASK.
; MOV #214,R0 ;SET BREAK,RESELECT LOOPBACK,ENABLE RECEPTION.
; JSR PC,WTWLN ;SET BREAK ON SELECTED LINE.
;
; *
; DELAY FOR 5 MS TO ALLOW TIME FOR BREAK TO BE GENERATED AND RECEIVED.
; VERIFY RECEPTION OF A CHARACTER WITH FRAME ERROR BIT SET.
;
; -
; MOV #5.,R4 ;SET DELAY VALUE TO 5 MILLI SECS.
; JSR PC,DELAY ;ALLOW TIME FOR CHARACTER RECEPTION.
; MOV @RBUFA,R0 ;GET CHARACTER FROM RBUF REGISTER.
; BIT #BIT13,R0 ;CHECK FOR FRAME ERROR BIT.
; BNE B0 ;SKIP ERROR REPORT IF SET.
; MOV #EM6402,R1 ;SELECT MESSAGE TO BE PRINTED.
; REPORT ERROR "BREAK NOT RECEIVED ON LINE #NN"
; ERDF 6405,EM6401,ER6401 ; >>>>> ERROR #6405 <<<<<.
; TRAP C#ERDF
; .WORD 6405
; .WORD EM6401
; .WORD ER6401
;
;
80: ASL R3 ;SHIFT BIT MASK FOR NEXT LINE.
; INC R2 ;NEXT LINE
; CMP R2,#NUMLNS ;CHECK FOR MAX LINE COUNT.
; BNE B0 ;IF <>,LOOP TO CHECK NEXT LINE
;
; *
; VERIFY BREAK GENERATION ON ALL LINES SIMULTANEOUSLY.
; CLEAR BREAKS ON ALL LINES.
; DELAY FOR 10 MILLI-SECONDS TO ALLOW TIME FOR ANY BREAKS TO BE CLEARED.
; PURGE THE FIFO.
; SET BREAK BIT IN LNCYRL REGISTERS ON ALL ACTIVE LINES.
; TEST FOR CHARACTERS IN THE FIFO WITH FRAME ERROR.
;
; -
; MOV #MAPLNS,R5 ;SET UP LINE TO CLEAR BREAKS ON.
; MOV #200,R0 ;SET UP PARAMETER TO CLEAR BREAK BITS.
; JSR PC,WTWLN ;CLEAR BREAK BIT,RESELECT INTERNAL LOOPBACK.
; MOV #10.,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
; JSR PC,DELAY ;DELAY TO ALLOW BREAKS TO BE CLEARED.

```

HARDWARE TEST - BREAKB -

```

6507
6508
6509
6510 026716 004767 166110
6511 026722 103044
6512
6513
6514
6515
6516 026724 016705 153304
6517 026730 012700 000214
6518 026734 004767 167750
6519
6520
6521
6522
6523 026740 012704 000012
6524 026744 004767 164746
6525 026750 010502
6526 026752 004767 165362
6527 026756 017701 153262
6528 026762 100011
6529 026764 032701 020000
6530 026770 001406
6531 026772 000301
6532 026774 042701 177400
6533 027000 004767 165306
6534 027004 040005
6535 027006 005302
6536 027010 001362
6537 027012 005705
6538 027014 001411
6539 027016 012701 007203
6540
6541 027022
        027022 104455
        027024 014406
        027026 007154
        027030 011740
6542 027032 000402
6543
6544 027034 004767 166740
6545 027040 005067 153220
6546 027044
        027044
        027044 104401

;+
; PURGE THE FIFO OF UNWANTED CHARACTERS.
;-
        JSR    PC,PUFIFO    ;PURGE FIFO.
        BCC    50$         ;GO REPORT ERROR IF FAILED TO CLEAN_OUT FIFO.

;+
; SET UP PARAMETERS FOR SETTING THE BREAK BIT ON ALL ACTIVE LINES.
; THEN CALL ROUTINE TO DO IT.
;-
10$:    MOV    ACTLNS,R5    ;SET UP ACTIVE LINE BIT MASK.
        MOV    #214,R0     ;SET BREAK,RESELECT LOOPBACK,ENABLE RECEPTION.
        JSR    PC,WTWLCN   ;SET BREAK ON SELECTED LINES.

;+
; DELAY FOR 10 MILLI SECONDS,TO ALLOW TIME FOR RECEPTION.
; TEST FOR CHARACTERS IN FIFO WITH FRAME ERROR BIT SET.
;-
        MOV    #10.,R4     ;SET DELAY VALUE TO 10 MILLI SECS.
        JSR    PC,DELAY    ;ALLOW TIME FOR CHARACTER RECEPTION.
        MOV    R5,R2      ;COPY ACTIVE LINE BIT MAP.
        JSR    PC,MAPCNT   ;COUNT THE NUMBER OF LINES AVAILABLE.
12$:    MOV    BRBUFA,R1   ;GET CHARACTER FROM RBUF REGISTER.
        BPL    14$        ;BRANCH IF DATA_VALID NOT SET.
        BIT    #BIT13,R1  ;CHECK FOR FRAME ERROR BIT.
        BEQ    14$        ;DO NOT CLR FLG FOR THIS LINE IF FRAME BIT CLR.
        SWAB   R1         ;GET LINE NUMBER IN LOW BYTE.
        BIC    #177400,R1 ;CLEAR EVERYTHING BUT THE LINE NUMBER.
        JSR    PC,LINBIT   ;CALC BIT MASK FROM LINE NUMBER.
        BIC    R0,R5      ;CLEAR LINE FLAG.
14$:    DEC    R2         ;DECREMENT THE LINE NUMBER COUNTER.
        BNE    12$        ;LOOP TO GET THE NEXT CHARACTER.
        TST    R5         ;CHECK IF ANY BREAKS NOT RECEIVED.
        BEQ    60$        ;EXIT TEST IF ALL CLEAR.
        MOV    #EM6402,R1  ;SELECT MESSAGE TO BE PRINTED.
;REPORT ERROR "BREAK NOT RECEIVED ON LINE #NN".
        ERDF  6406,EM6401,ER6401; >>>>> ERROR #6407 <<<<<.
                                     TRAP    C#ERDF
                                     .WORD   6406
                                     .WORD   EM6401
                                     .WORD   ER6401

        BR    60$         ;EXIT THE TEST.

50$:    JSR    PC,TSABRT   ;ABORT THE TEST.
60$:    CLR    CTRLCF     ;INDICATE THAT WE ARE NOT WITHIN A TEST.
        ENDTST

L10037: TRAP    C#ETST

```

HARDWARE TEST - NORERR -

```

6548
6549
6550
6551
6552
6553
6554
6555
6556
6557
6558
6559
6560
6561
6562 027046
      027046
6563 027046
      027046 012700 000240
      027052 104441
6564      000016
6565 027054 012767 000016 153204
6566 027062 012767 177777 153174
6567 027070 012767 000001 155006
6568 027076 012767 014711 155002
6569 027104 012767 007244 154776
6570
6571
6572
6573
6574
6575 027112 004767 164366
6576 027116 103402
6577 027120 000167 000422
6578
6579
6580
6581
6582 027124 004767 165002
6583 027130 103402
6584 027132 000167 000410
6585 027136 004767 165050
6586
6587
6588
6589
6590
6591
6592
6593
6594 027142 005267 154740
6595 027146 012700 000204
6596 027152 004767 167532
6597 027156 012700 177670
6598 027162 004767 167552
6599 027166 016704 150620
6600 027172 004767 164520
6601 027176 012702 002704

```

```

.SBTTL HARDWARE TEST - NORERR -
;*****
;* - NO OVERRUN ERROR TEST -
;*
;* THIS TEST VERIFIES THAT THE DUT WILL NOT REPORT DATA OVERRUN
;* ERRORS WHEN THEY DO NOT OCCUR.
;* THIS TEST PUTS 256 CHARACTERS IN THE DUT FIFO PLUS 4 IN EACH ACTIVE
;* UART AND VERIFIES THAT NO OVERRUN ERRORS ARE REPORTED.
;* ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
;* HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
;* REPORTED LATER.
;*
;*****
      BGNTST
      SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T14::
      MOV #PRI05,R0
      TRAP C#SPRI
      TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (66)
      MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
      MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
      MOV #6601,ERRNBR ;SET ERROR NUMBER TO 6601.
      MOV #EM6601,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
;+
; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
; THIS SUBROUTINE REPORTS ERROR >>>> 6601 <<<<<.
;-
      JSR PC,CLNRST ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
      BCS .+6 ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
      JMP 60# ;EXIT THE TEST, FATAL ERROR WAS FOUND.
;+
; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
; INITIALIZE THE 256 BYTE DATA PATTERN.
;-
      JSR PC,FINACT ;FIND AN ACTIVE LINE.
      BCS .+6 ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
      JMP 60# ;EXIT THE TEST, FATAL ERROR WAS FOUND.
      JSR PC,INDATP ;INITIALISE DATA PATTERN.
;+
; TRANSMIT A 265 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
;-
;+
; SET INTERNAL LOOPBACK ON THE SELECTED LINE.
; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
;-
      INC ERRNBR ;SET THE ERROR REPORT NUMBER TO 6602.
      MOV #204,R0 ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
      JSR PC,WTMLNC ;INITILAISE THE LINE CONTROL REGISTER.
      MOV #177670,R0 ;PASS THE LPR CONTENTS.
      JSR PC,WTMLPR ;SET THE LPR CONTENTS TO 38.4K BAUD.
      MOV 10.,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
      JSR PC,DELAY ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
      MOV #8,BUFBAS,R2 ;PASS THE START OF THE DATA PATTERN TO TX.

```

HARDWARE TEST - NORERR -

```

6602 027202 012703 000400      MOV    #BUF MID-BUF BAS,R3 ;PASS THE LENGTH OF THE DATA PATTERN.
6603 027206 004767 164544      JSR    PC,DODMA             ;TRANSMIT THE DATA PATTERN.
6604 027212 103153              BCC    50$                 ;EXIT IF ERROR FOUND DURING DMA TX.
6605
6606                          ;*
6607                          ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
6608                          ; THE FIFO.
6609 027214 005267 154666      INC    ERRNBR              ;SET ERROR NUMBER TO 6603.
6610 027220 012701 170536      MOV    #170536,R1         ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
6611 027224 016702 153012      MOV    CSRA,R2            ;PASS THE ADDRESS OF THE CSR.
6612 027230 004767 167340      JSR    PC,WAIBIS          ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6613 027234 103142              BCC    50$                 ;ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
6614 027236 012704 000005      MOV    #5,R4              ;PASS DELAY OF 5 MILLI SECS.
6615 027242 004767 164450      JSR    PC,DELAY           ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6616
6617
6618                          ;*
6619                          ; TRANSMIT 4 CHARACTERS ON EACH ACTIVE LINE.
6620 027246 016705 152762      MOV    ACTLNS,R5          ;ALTER PARAMETERS FOR ALL ACTIVE LINES.
6621 027252 012700 000204      MOV    #204,R0            ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
6622 027256 004767 167426      JSR    PC,WTW LNC         ;INITIALISE THE LINE CONTROL REGISTER.
6623 027262 012700 177670      MOV    #177670,R0         ;PASS THE LPR CONTENTS.
6624 027266 004767 167446      JSR    PC,WTW LPR         ;SET THE LPR CONTENTS TO 38.4K BAUD.
6625 027272 016704 150514      MOV    10.,R4             ;PASS DELAY TIME OF 10 MILLI SECONDS.
6626 027276 004767 164414      JSR    PC,DELAY           ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6627
6628 027302 012702 002704      MOV    #BUF BAS,R2        ;PASS THE START OF THE DATA PATTERN TO TX.
6629 027306 012703 000004      MOV    #4,R3              ;PASS THE LENGTH OF THE DATA PATTERN.
6630 027312 005001              CLR    R1                  ;CLEAR THE LINE COUNTER.
6631 027314 005267 154566      INC    ERRNBR             ;SET ERROR NUMBER TO 6604.
6632 027320 010100      2$:  MOV    R1,R0
6633 027322 006300      ASL    R0                  ;CALCULATE THE LINE OFFSET FROM THE LINE #.
6634 027324 036067 002370 152702  BIT    BITTBL(R0),ACTLNS ;TEST FOR THIS LINE BEING ACTIVE.
6635 027332 001403      BEQ    4$                  ;SKIP THE TX ON THIS LINE IF IT IS NOT ACTIVE.
6636 027334 004767 164416      JSR    PC,DODMA           ;TRANSMIT THE 5 CHAR DATA PATTERN.
6637 027340 103100      BCC    50$                 ;ABORT IF ERROR FOUND DURING DMA TX.
6638 027342 005201      4$:  INC    R1               ;INCREMENT THE LINE COUNTER.
6639 027344 020127 000010      CMP    R1,#NUM LNS        ;TEST FOR ALL POSSIBLE LINES HANDLED
6640 027350 002763      BLT    2$                  ;LOOP IF NOT ALL LINES HANDLED.
6641
6642 027352 005267 154530      INC    ERRNBR             ;SET ERROR NUMBER TO 6605.
6643 027356 012701 170040      MOV    #170040,R1         ;PASS TIME-OUT VALUE OF 32 MILLI SECS.
6644 027362 016702 152654      MOV    CSRA,R2            ;PASS THE ADDRESS OF THE CSR.
6645 027366 004767 167202      JSR    PC,WAIBIS          ;WAIT FOR A DMA TO COMPLETE, TX_ACTION SET.
6646 027372 103063      BCC    50$                 ;ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
6647 027374 012704 000005      MOV    #5,R4              ;PASS DELAY OF 5 MILLI SECS.
6648 027400 004767 164312      JSR    PC,DELAY           ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6649
6650                          ;*
6651                          ; READ THE FIFO CHECKING FOR OVERRUN ERRORS. REPORT ERRORS IF FOUND.
6652                          ; ABORT THE TEST IF A BMP CODE WAS FOUND IN THE FIFO.
6653 027404 016702 152624      MOV    ACTLNS,R2
6654 027410 004767 164724      JSR    PC,MAPCNT          ;GET THE NUMBER OF ACTIVE LINES.
6655 027414 006302      ASL    R2
6656 027416 006302      ASL    R2                  ;MULTIPLY NUMBER OF ACTIVE LINES BY 4.
6657 027420 012705 000400      MOV    #256.,R5
6658 027424 060205      ADD    R2,R5              ;CALCULATE NUMBER OF CHARACTERS TO RX.

```

HARDWARE TEST

- NORERR -

```

6659 027426 005004          CLR    R4          ;CLEAR THE CHARACTER COUNTER.
6660 027430 012767 014716 154450 6$:  MOV    #6606.,ERRNBR ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
6661 027436 017702 152602          MOV    @RBUFA,R2    ;READ A CHARACTER FROM THE FIFO.
6662 027442 100032          BPL    10$          ;EXIT THE READ LOOP IF THE FIFO IS EMPTY.
6663
6664          ;*
6665          ; CHECK IF THE READ CHARACTER IS A BMP CODE.
6666          ; IF IT IS A BMP CODE SAVE IT ON THE QUEUE TO BE REPORTED LATER, AND
6667          ; ABORT THE TEST.
6668 027444 004767 163734          JSR    PC,CHKBMP    ;CHECK IF CHARACTER IS A BMP CODE.
6669 027450 103002          BCC    8$          ;BRANCH IF NOT A BMP CODE.
6670 027452 104460          ERROR          ;
6671 027454 000434          BR    60$          ;EXIT THIS TEST.
6672
6673 027456 005267 154424          8$:  INC    ERRNBR      ;SET ERROR NUMBER TO 6607.
6674 027462 005204          INC    R4          ;COUNT THIS CHARACTER.
6675 027464 020405          CMP    R4,R5      ;COMPARE # OF CHARS WITH MAX # OF CHARS.
6676 027466 003025          BGT    50$        ;ABORT TEST IF TOO MANY VALID CHARS READ.
6677 027470 032702 040000          BIT    @BIT14,R2  ;TEST THE OVERRUN BIT OF THE READ CHAR.
6678 027474 001755          BEQ    6$          ;LOOP TO READ THE NEXT CHAR IF NO ERROR.
6679 027476 005267 154404          INC    ERRNBR      ;SET ERROR NUMBER TO 6608.
6680 027502 012767 012036 154402          MOV    @ER7801,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6681 027510 012701 007272          MOV    @EM6602,R1 ;PASS THE MESSAGE TO BE REPORTED.
6682 027514 010203          MOV    R2,R3
6683 027516 000303          SWAB   R3
6684 027520 042703 177760          BIC    @177760,R3 ;GET FAILING LINE NUMBER.
6685          ;REPORT "OVERRUN ERROR REPORTED WHEN NONE FORCED, ON LINE NN ..."
6686 027524 104460          ERROR          ;
6687 027526 000740          BR    6$          ;LOOP TO READ THE NEXT CHAR.
6688
6689 027530 012767 014721 154350 10$: MOV    #6609.,ERRNBR ;SET ERROR NUMBER TO 6609.
6690 027536 020405          CMP    R4,R5      ;COMPARE NUMBER OF CHARS READ WITH EXPECTED.
6691 027540 001402          BEQ    60$        ;EXIT TEST WITHOUT ABORT IF CORRECT # OF CHARS.
6692
6693 027542 004767 166232          50$: JSR    PC,TSABRT   ;ABORT THE TEST, NON-RELATED TEST ERROR FOUND.
6694 027546 005067 152512          60$: CLR    CTRLCF     ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6695 027552          ENDTST
        027552 104401          L10040: TRAP    C#ETST
    
```

HARDWARE TEST - ORERR -

```

6697 .SBTTL HARDWARE TEST - ORERR -
6698 ;*****
6699 ;*
6700 ;* - OVERRUN ERROR TEST -
6701 ;*
6702 ;* THIS TEST VERIFIES THAT THE DUT WILL REPORT DATA OVERRUN ERRORS WHEN
6703 ;* THEY OCCURR.
6704 ;* THIS TEST PUTS 256 CHARACTERS IN THE DUT FIFO PLUS 5 IN EACH ACTIVE
6705 ;* UART AND VERIFIES THAT OVERRUN ERRORS ARE REPORTED ON ALL ACTIVE LINES.
6706 ;* ANY BMP CODE FOUND WILL INVALIDATE THE TEST AND CAUSE IT TO BE ABORTED.
6707 ;* HOWEVER THE BMP CODE WILL BE PLACED ON THE BMP CODE QUEUE, TO BE
6708 ;* REPORTED LATER.
6709 ;*
6710 ;*****
6711 BGNTST
6712 027554 SETPRI #PRI05 ;ALLOW LTC INTERRUPTS. T15::
027554 012700 000240 ;
027560 104441 ;
027560 104441 ;
6713 000017 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
6714 027562 012767 000017 152476 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (67)
6715 027570 012767 177777 152466 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
6716 027576 012767 000001 154300 MOV #1,ERRTYP ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
6717 027604 012767 015055 154274 MOV #6701,ERRNBR ;SET ERROR NUMBER TO 6701.
6718 027612 012767 007344 154270 MOV #EM6701,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
6719 ;
6720 ;*
6721 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
6722 ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
6723 ; THIS SUBROUTINE REPORTS ERROR >>>> 6701 <<<<<.
6724 027620 004767 163660 JSR PC,CLNRST ;RESET THE DHV-11, REPORT ANY ERRORS FOUND.
6725 027624 103402 BCS .+6 ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
6726 027626 000167 000650 JMP 60# ;EXIT THE TEST, FATAL ERROR WAS FOUND.
6727 ;
6728 ;*
6729 ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
6730 ; INITIALIZE THE 256 BYTE DATA PATTERN.
6731 027632 004767 164274 JSR PC,FINACT ;FIND AN ACTIVE LINE.
6732 027636 103402 BCS .+6 ;IF ACTIVE LINE IS FOUND, DON'T ABORT TEST.
6733 027640 000167 000636 JMP 60# ;ABORT THE TEST, NO ACTIVE LINES WERE FOUND.
6734 027644 004767 164342 JSR PC,INDATP ;INITIALISE DATA PATTERN.
6735 ;
6736 ;*
6737 ; TRANSMIT A 265 CHARACTER DATA PATTERN USING DMA, ON A SINGLE CHANNEL
6738 ; AT 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY, 2 STOP BITS.
6739 ;
6740 ;*
6741 ; SET INTERNAL LOOPBACK ON THE SELECTED LINE.
6742 ; TRANSMIT THE DATA PATTERN ON THE FIRST AVAILABLE ACTIVE LINE.
6743 027650 005267 154232 INC ERRNBR ;SET ERROR NUMBER TO 6702.
6744 027654 012700 000204 MOV #204,R0 ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
6745 027660 004767 167024 JSR PC,WTWLNCR ;INITILAISE THE LINE CONTROL REGISTER.
6746 027664 012700 177670 MOV #177670,R0 ;PASS THE LPR CONTENTS.
6747 027670 004767 167044 JSR PC,WTWLPR ;SET THE LPR CONTENTS TO 38.4K BAUD.
6748 027674 016704 150112 MOV 10.,R4 ;PASS DELAY TIME OF 10 MILLI SECONDS.
6749 027700 004767 164012 JSR PC,DELAY ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6750 027704 012702 002704 MOV #BUFBAS,R2 ;PASS THE START OF THE DATA PATTERN TO TX.

```



HARDWARE TEST - ORERR -

```

6751 027710 012703 000400      MOV      #BUF MID-BUF BAS,R3 ;PASS THE LENGTH OF THE DATA PATTERN.
6752 027714 004767 164036      JSR      PC,DODMA           ;TRANSMIT THE DATA PATTERN.
6753 027720 103402             BCS      .+6                ;IF NO ERROR FOUND DURING DMA TX, DON'T ABORT.
6754 027722 000167 000550      JMP      50$               ;ABORT TEST, ERROR FOUND DURING DMA TX.
6755
6756                          ;+
6757                          ; WAIT FOR DMA TO COMPLETE, THEN WAIT FOR THE LAST CHARACTER TO ARRIVE IN
6758                          ; THE FIFO.
6759 027726 005267 154154      INC      ERRNBR            ;SET ERROR NUMBER TO 6703.
6760 027732 012701 170536      MOV      #170536,R1       ;PASS TIME-OUT VALUE OF 350 MILLI SECS.
6761 027736 016702 152300      MOV      CSRA,R2          ;PASS THE ADDRESS OF THE CSR.
6762 027742 004767 166626      JSR      PC,WAIBIS        ;WAIT FOR DMA TO COMPLETE, TX_ACTION SET.
6763 027746 103402             BCS      .+6                ;IF NO TIME-OUT ON DMA COMPLETION, DON'T ABORT.
6764 027750 000167 000522      JMP      50$               ;ABORT TEST, TIME-OUT ON DMA COMPLETION.
6765 027754 012704 000005      MOV      #5,R4            ;PASS DELAY OF 5 MILLI SECS.
6766 027760 004767 163732      JSR      PC,DELAY         ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6767
6768                          ;+
6769                          ; TRANSMIT 5 CHARACTERS ON EACH ACTIVE LINE.
6770 027764 016705 152244      MOV      ACTLNS,R5        ;ALTER PARAMETERS FOR ALL ACTIVE LINES.
6771 027770 012700 000204      MOV      #204,R0          ;PASS PARAMETER FOR INTERNAL LOPBCK,ENABLE RX.
6772 027774 004767 166710      JSR      PC,WTW LNC       ;INITILAISE THE LINE CONTROL REGISTER.
6773 030000 012700 177670      MOV      #177670,R0       ;PASS THE LPR CONTENTS.
6774 030004 004767 166730      JSR      PC,WTW LPR       ;SET THE LPR CONTENTS TO 38.4K BAUD.
6775 030010 016704 147776      MOV      10.,R4           ;PASS DELAY TIME OF 10 MILLI SECONDS.
6776 030014 004767 163676      JSR      PC,DELAY         ;WAIT FOR LNCTRL AND LPR REGS TO BE UPDATED.
6777
6778 030020 012702 002704      MOV      #BUF BAS,R2      ;PASS THE START OF THE DATA PATTERN TO TX.
6779 030024 012703 000005      MOV      #5,R3            ;PASS THE LENGTH OF THE DATA PATTERN.
6780 030030 005001             CLR      R1                ;CLEAR THE LINE COUNTER.
6781 030032 005267 154050      INC      ERRNBR            ;SET ERROR NUMBER TO 6704.
6782 030036 010100             2$:    MOV      R1,R0
6783 030040 006300             ASL      R0                ;CALCULATE LINE OFFSET FROM THE LINE #.
6784 030042 036067 002370 152164 BIT      BITTBL(R0),ACTLNS ;TEST FOR THIS LINE BEING ACTIVE.
6785 030050 001405             BEQ      4$                ;SKIP THE TX ON THIS LINE IF IT IS NOT ACTIVE.
6786 030052 004767 163700      JSR      PC,DODMA         ;TRANSMIT THE 5 CHAR DATA PATTERN.
6787 030056 103402             BCS      .+6                ;IF NO TIME-OUT ON DMA COMPLETION, DON'T ABORT.
6788 030060 000167 000412      JMP      50$               ;ABORT TEST, TIME-OUT ON DMA COMPLETION.
6789 030064 005201             4$:    INC      R1                ;INCREMENT THE LINE NUMBER COUNTER.
6790 030066 020127 000010      CMP      R1,#NUM LNS      ;TEST FOR ALL POSSIBLE LINES HANDLED
6791 030072 002761             BLT      2$                ;LOOP IF NOT ALL LINES HANDLED.
6792
6793 030074 005267 154006      INC      ERRNBR            ;SET ERROR NUMBER TO 6705.
6794 030100 012701 170040      MOV      #170040,R1       ;PASS TIME-OUT VALUE OF 32 MILLI SECS.
6795 030104 016702 152132      MOV      CSRA,R2          ;PASS THE ADDRESS OF THE CSR.
6796 030110 004767 166460      JSR      PC,WAIBIS        ;WAIT FOR A DMA TO COMPLETE, TX_ACTION SET.
6797 030114 103170             BCC      50$               ;ABORT THE TEST IF TIME-OUT ON DMA COMPLETION.
6798 030116 012704 000005      MOV      #5,R4            ;PASS DELAY OF 5 MILLI SECS.
6799 030122 004767 163570      JSR      PC,DELAY         ;WAIT FOR LAST CHAR TO ARRIVE IN THE FIFO.
6800
6801                          ;+
6802                          ; READ 256 CHARS FROM THE FIFO CHECKING FOR BMP CODES.
6803                          ; ABORT THE TEST IF A BMP CODE WAS FOUND IN THE FIFO.
6804 030126 012704 000400      MOV      #256.,R4         ;SET UP THE CHARACTER COUNTER.
6805 030132 012767 015062 153746 6$:  MOV      #6706.,ERRNBR    ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
6806 030140 017702 152100      MOV      @RBUFA,R2        ;READ A CHARACTER FROM THE FIFO.
6807 030144 100154             BPL      50$               ;ABORT THE TEST IF DATA.VALID IS CLEAR.

```

HARDWARE TEST

- ORERR -

```

6808 030146 005267 153734          INC  ERRNBR      ;SET ERROR NUMBER TO 6707.
6809 030152 004767 163226          JSR  PC,CHKBMP  ;CHECK IF CHARACTER IS A BMP CODE.
6810 030156 103545                   BCS  24$        ;REPORT ERROR AND ABORT TEST IF A BMP CODE.
6811 030160 005304                   DEC  R4         ;COUNT THIS CHARACTER.
6812 030162 001363                   BNE  6$        ;LOOP IF NOT 256 CHARS READ FROM FIFO.
6813
6814          ;+
6815          ; READ THE REMAINING AND VERIFY 1 OVERRUN PLUS 1 CHAR FROM EACH LINE.
6816          ;-
6816 030164 005004                   CLR  R4         ;CLEAR THE OVERRUN ERROR FLAGS.
6817 030166 012700 003744          MOV  #RXCNTB,R0
6818 030172 004767 163330          JSR  PC,CLR16W  ;CLEAR RX CHAR COUNT TABLE.
6819 030176 012767 015064 153702 8$:  MOV  #6708.,ERRNBR ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
6820 030204 017702 152034          MOV  #RBUFA,R2 ;READ A CHARACTER FROM THE FIFO.
6821 030210 100047                   BPL  14$        ;GO ANALYZE THE RESULTS IF ALL CHARS READ.
6822 030212 004767 163166          JSR  PC,CHKBMP  ;CHECK IF CHAR IS A BMP CODE.
6823 030216 103525                   BCS  24$        ;REPORT ERROR AND ABORT TEST IF A BMP CODE.
6824 030220 005267 153662          INC  ERRNBR     ;SET ERROR NUMBER TO 6709.
6825 030224 010200                   MOV  R2,R0
6826 030226 000300                   SWAB R0
6827 030230 042700 177760          BIC  #177760,R0 ;CALCULATE THE LINE NUMBER OF THE CHAR.
6828 030234 006300                   ASL  R0         ;FORM WORD TABLE OFFSET FOR TABLE ACCESS.
6829 030236 042702 007400          BIC  #7400,R2  ;REMOVE LINE NUMBER FROM THE READ CHAR.
6830 030242 036067 002370 151764   BIT  BITTBL(R0),ACTLNS ;TEST FOR ACTIVE LINE.
6831 030250 001512                   BEQ  50$        ;ABORT TEST IF FOR INACTIVE LINE.
6832 030252 005267 153630          INC  ERRNBR     ;SET ERROR NUMBER TO 6710.
6833 030256 005760 003744          TST  RXCNTB(R0) ;CHECK THE RX CHAR COUNTER FOR THIS LINE.
6834 030262 001006                   BNE  10$        ;IS THIS FIRST CHAR ON LINE?
6835 030264 020227 140000          CMP  R2,#140000 ;YES, TEST FOR NULL CHAR WITH OVERRUN.
6836 030270 001414                   BEQ  12$        ;IS CHAR A NULL?
6837 030272 056004 002370          BIS  BITTBL(R0),R4 ;NO, SET THE OVERRUN BIT ERROR FLAG FOR LINE.
6838 030276 000411                   BR   12$        ;GO COUNT THE CHAR AND CONTINUE.
6839 030300 026027 003744 000004 10$:  CMP  RXCNTB(R0),#4
6840 030306 002073                   BGE  50$        ;5TH CHAR ON THIS LINE? YES, ABORT.
6841 030310 032702 040000          BIT  #BIT14,R2 ;NO, CHECK OVERRUN BIT.
6842 030314 001402                   BEQ  12$        ;IS OVERRUN BIT CLEAR? YES, GO COUNT CHAR.
6843 030316 056004 002370          BIS  BITTBL(R0),R4 ;NO, SET THE OVERRUN BIT ERROR FLAG FOR LINE.
6844 030322 005260 003744          INC  RXCNTB(R0) ;COUNT THIS CHARACTER.
6845 030326 000723                   BR   8$        ;LOOP UNTIL ALL CHARS ARE READ FROM FIFO.
6846
6847          ;+
6848          ; TEST FOR ABORT CONDITIONS. ONLY NONE ABORT CONDITIONS ARE:
6849          ; 1) 2 CHARS RXED ON A LINE AND NO OVERRUN ERROR BIT FAILURE DETECTED.
6850          ; 2) 2 TO 4 CHARS RXED ON A LINE AND AN OVERRUN BIT FAILURE DETECTED.
6851          ;-
6851 030330 005001 14$:  CLR  R1         ;INITIALIZE LINE LOOP, CLEAR LINE OFFSET.
6852 030332 012767 015067 153546 16$:  MOV  #6711.,ERRNBR ;SET UP ERROR NUMBER EACH TIME AROUND LOOP.
6853 030340 036167 002370 151666   BIT  BITTBL(R1),ACTLNS
6854 030346 001415                   BEQ  18$        ;LINE ACTIVE? NO, NEXT LINE.
6855 030350 026127 003744 000002   CMP  RXCNTB(R1),#2 ;YES.
6856 030356 002447                   BLT  50$        ;FEWER THAN 2 CHARS RXED? YES, ABORT.
6857 030360 036104 002370          BIT  BITTBL(R1),R4 ;NO.
6858 030364 001006                   BNE  18$        ;OVERRUN BIT ERROR FLAG SET? YES, NEXT LINE.
6859 030366 005267 153514          INC  ERRNBR     ;SET LINE NUMBER TO 6712.
6860 030372 026127 003744 000002   CMP  RXCNTB(R1),#2
6861 030400 001036                   BNE  50$        ;NOT 2 CHARS RXED? YES, ABORT. NO, NEXT LINE.
6862 030402 062701 000002 18$:  ADD  #2,R1     ;SET LINE OFFSET TO THE NEXT LINE.
6863 030406 020127 000020          CMP  R1,#NUMLNS*2
6864 030412 002747                   BLT  16$        ;ALL LINES DONE? NO, LOOP. YES, CONTINUE.

```

HARDWARE TEST

- ORERR -

```

6865
6866          ;+
6867          ; CHECK FOR OVERRUN ERROR BIT FAILURES, PRINT ERROR MESSAGE IF FOUND.
6868 030414 012767 015071 153464          ;-
6869 030422 005001          MOV #6713.,ERRNBR ;SET UP ERROR NUMBER.
6870 030424 010102          CLR R1 ;INITIALIZE LOOP. CLEAR LINE OFFSET.
6871 030426 036104 002370 20$: MOV R1,R2 ;COPY THE LINE OFFSET.
6872 030432 001411          BIT BITTBL(R1),R4 ;OVERRUN BIT FAILURE FLAGS ARE IN R4.
6873 030434 010103          BEQ 22$ ;ERROR FLAG CLEAR? YES, NEXT LINE.
6874 030436 006203          MOV R1,R3
6875 030440 012767 012036 153444          ASR R3 ;CALCULATE LINE NUMBER FROM LINE OFFSET.
6876 030446 012701 007367          MOV #ER7801,ERRBLK ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
6877          MOV #EM6702,R1 ;PASS THE MESSAGE TO BE REPORTED.
6878 030452          ;REPORT "OVERRUN ERROR NOT REPORTED CORRECTLY WHEN FORCED, ON LINE NN ..."
        030452 104460          ERROR ; >>>> ERROR #6713 <<<<<.
6879 030454 010201          MOV R2,R1 ;RESTORE THE LINE OFFSET THAT WAS DESTROYED.
6880 030456 046104 002370 22$: BIC 3ITTBL(R1),R4 ;CLEAR THE LINE ERROR FLAG WE JUST HANDLED.
6881 030462 001407          BEQ 60$ ;ALL FAILURE BITS HANDLED? YES, EXIT TEST.
6882 030464 062701 000002          ADD #2,R1 ;NO, INCREMENT THE LINE OFFSET.
6883 030470 000755          BR 20$ ;LOOP TO HANDLE THE NEXT LINE.
6884
6885 030472          24$: ;REPORT "BMP CODE FOUND IN FIFO, TEST INVALIDATED."
6886 030472          ERROR ; >>>> ERROR <<<<<.
        030472 104460          TRAP C$ERROR
6887 030474 000402          BR 60$ ;EXIT THIS TEST.
6888
6889 030476 004767 165276 50$: JSR PC,TSABRT ;ABORT THE TEST. ERROR # INDICATES FAULT TYPE.
6890 030502 005067 151556 60$: CLR CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
6891
6892 030506          ENDTST
        030506
        030506 104401          L10041: TRAP C$ETST
    
```

HARDWARE TEST - DTRMCS -

```

6894
6895
6896
6897
6898
6899
6900
6901
6902
6903
6904
6905
6906 030510
      030510
6907
6908
6909
6910 030510 032767 000002 151520
6911 030516 001002
6912 030520 000167 000464
6913 030524
      030524 012700 000240
      030530 104441
6914      000020
6915 030532 012767 000020 151526
6916 030540 012767 177777 151516
6917 030546 012767 000001 153330
6918 030554 012767 017171 153324
6919 030562 012767 007444 153320
6920
6921
6922
6923
6924
6925 030570 004767 162710
6926 030574 103402
6927 030576 000167 000406
6928
6929
6930
6931 030602 004767 162242
6932
6933
6934
6935
6936
6937
6938 030606 005003
6939 030610 010300
6940 030612 006300
6941 030614 036067 002370 151412
6942 030622 001465
6943
6944
6945
6946 030624 005000
6947 030626 012705 000377

```

```

.SBTTL HARDWARE TEST - DTRMCS -
;.....
; DATA TERMINAL READY MODEM CONTROL SIGNAL TEST -
;
; THIS TEST VERIFIES THAT THE DTR 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 RI
; AND DSR TO TEST THE DTR SIGNAL. THIS TEST IS PERFORMED ON ALL
; ACTIVE LINES.
;.....
      BGNTST
      T16::
; ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
;
      BIT      @BIT1,LOPBC  ;CHECK TYPE OF LOOPBACK MODE SELECTED.
      BNE     28
      JMP     608
28:      SETPRI @PRIOS
      ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
      ;ALLOW LTC INTERRUPTS.
      MOV     @PRIOS,RO
      TRAP   C@SPRI
      TNUM == TNUM + 1
      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV     @TNUM,TSTNUM ;SET UP THE TEST NUMBER. (78)
      MOV     @-1,CTRLCF   ;INDICATE THAT WE ARE IN A TEST.
      MOV     @1,ERRTYP    ;SET ERROR TYPE IN ERROR TABLE.
      MOV     @7801,,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
      MOV     @EM7801,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 >>>> 7801 <<<<<.
;
      JSR     PC,CLRST      ;RESET THE DUT.
      BCS     48
      JMP     608
      ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
;
; SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
;
48:      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 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.
68:      MOV     R3,RO
      ASL     RO
      BIT     BITBL(RO),ACTLNS
      BEQ     128
      ;DON'T TEST IF NOT ACTIVE LINE.
;
; CLEAR ALL THE DUT LNCTRL REGISTERS DTR BITS.
;
      CLR     RO
      MOV     @MAPLNS,R5
      ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.

```

## HARDWARE TEST

- DTRMCS -

```

6948 030632 004767 166052      JSR    PC,WTWLNK      ;CLEAR ALL THE DUT DTR BITS.
6949 030636 012704 000074      MOV    #60.,R4
6950 030642 004767 163050      JSR    PC,DELAY      ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.
6951
6952      ;*
6953      ; CHECK THAT AT LEAST ONE OF ASSOCIATED DSR OR RI IS CLEAR AND RECORD STATES.
6954 030646 116304 004044      MOV    TXRLNB(R3),R4 ;GET THE ASSOCIATED LINE NUMBER.
6955 030652 010477 151364      MOV    R4,BCSRA      ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
6956 030656 017705 151366      MOV    @STATA,R5     ;GET THE STATE OF THE ASSOCIATED DSR, RI BITS.
6957 030662 012700 120000      MOV    #BIT15:BIT13,R0
6958 030666 040500      BIC    R5,R0         ;CHECK FOR BOTH DSR AND RI SET.
6959 030670 001431      BEQ    10#          ;GO REPORT DTR IS BAD IF BOTH ARE SET.
6960
6961      ;*
6962      ; SET THE DTR FOR THE SELECTED LINE AND WAIT FOR EITHER DSR OR RI TO SET.
6963 030672 010377 151344      MOV    R3,BCSRA      ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
6964 030676 052777 001000 151346      BIS    #BIT9,@LNCTRA ;SET THE SELECTED LINE DTR.
6965 030704 012701 150074      MOV    #150074,R1    ;SPECIFY TO WAIT UP TO 60 MS FOR RI TO SET.
6966 030710 032705 100000      BIT    #BIT15,R5     ;CHECK PREVIOUS STATE OF DSR BIT.
6967 030714 001002      BNE    8#           ;GO USE RI IF DSR BIT WAS NOT CLEAR.
6968 030716 012701 170074      MOV    #170074,R1    ;SPECIFY TO WAIT UP TO 60 MS FOR DSR SET.
6969 030722 016702 151322 8#:      MOV    STATA,R2      ;SPECIFY TO LOOK IN STAT REG FOR BIT TO SET.
6970 030726 010477 151310      MOV    R4,BCSRA      ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
6971 030732 004767 165636      JSR    PC,WAIBIS     ;WAIT UP TO 60 MS FOR SIGNAL TO GO SET.
6972 030736 103417      BCS    12#          ;SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.
6973 030740 017700 151304      MOV    @STATA,R0     ;GET THE STATUS REGISTER CONTENTS.
6974 030744 042700 057777      BIC    #57777,R0     ;REMOVE ALL BUT THE DSR AND RI BITS.
6975 030750 040500      BIC    R5,R0         ;TEST FOR SIGNAL ONCE CLEAR, BUT NOW SET.
6976 030752 001011      BNE    12#          ;GO LOOP IF SIGNAL HAS GONE FROM CLR TO SET.
6977 030754
6978 030754 012767 017172 153124 10#: ;REPORT DTR MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
6979 030762 012767 012036 153122      MOV    #7802.,ERRNBR ;SELECT THE ERROR NUMBER.
6980 030770 012701 007477      MOV    #ER7801,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
6981 030774      MOV    #EM7802,R1    ;SELECT THE ERROR MESSAGE.
6982 030774 104460      ERROR
6983 030776 005203      TRAP    C#ERROR
6984 031000 020327 000010 12#:      INC    R3           ;SELECT THE NEXT LINE NUMBER.
6985 031004 002701      CMP    R3,#NUMLNS   ;TEST FOR ALL LINES DONE.
6986      BLT    6#         ;LOOP IF NOT ALL LINES DONE.
6987
6988      ;*
6989      ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
6990      ; THIS LOOP SETS ALL THE DTRS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
6991      ; A RESPONSE ON THE ASSOCIATED RI AND DSR SIGNALS.
6992      ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
6993
6994 031006 005003      ;*
6995 031010 010300 14#:      CLR    R3           ;CLEAR THE LINE COUNTER.
6996 031012 006300      MOV    R3,R0
6997 031014 036067 002370 151212      ASL    R0
6998 031022 001466      BIT    BITTBL(R0),ACTLNS
6999      BEQ    20#          ;DON'T TEST IF NOT ACTIVE LINE.
7000
7001      ;*
7002      ; SET ALL THE DUT LNCTRL REGISTERS DTR BITS.
7003
7004 031024 012700 001000      MOV    #BIT9,R0     ;SPECIFY THAT DTR BITS ARE TO BE SET.
7005 031030 012705 000377      MOV    #MAPLNS,R5   ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
7006 031034 004767 165650      JSR    PC,WTWLNK     ;SET ALL THE DUT DTR BITS.
7007 031040 012704 000074      MOV    #60.,R4
7008 031044 004767 162646      JSR    PC,DELAY      ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.

```

HARDWARE TEST

- DTRMCS -

```

7004
7005 ; CHECK THAT AT LEAST ONE OF ASSOCIATED DSR OR RI IS SET AND RECORD STATES.
7006 ;
7007 031050 116304 004044      MOV      TXRLNB(R3),R4      ;GET THE ASSOCIATED LINE NUMBER.
7008 031054 010477 151162      MOV      R4,@CSRA         ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
7009 031060 017705 151164      MOV      @STATA,R5        ;GET THE STATE OF THE ASSOCIATED DSR, RI BITS.
7010 031064 010500
7011 031066 042700 057777      MOV      R5,R0
7012 031072 001431            BIC      @57777,R0        ;CHECK FOR BOTH DSR AND RI CLEAR.
7013 ;                                     BEQ      18#             ;GO REPORT DTR IS BAD IF BOTH ARE CLEAR.
7014 ; CLEAR THE DTR FOR THE SELECTED LINE AND WAIT FOR EITHER DSR OR RI TO CLEAR.
7015 ;
7016 031074 010377 151142      MOV      R3,@CSRA         ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
7017 031100 042777 001000 151144 BIC      @BIT9,@LNCTRA     ;CLEAR THE SELECTED LINE DTR.
7018 031106 012701 150074      MOV      @150074,R1       ;SPECIFY TO WAIT UP TO 60 MS FOR RI TO CLEAR.
7019 031112 032705 100000      BIT      @BIT15,R5        ;CHECK PREVIOUS STATE OF DSR BIT.
7020 031116 001402            BEQ      16#             ;GO USE RI IF DSR BIT WAS NOT SET.
7021 031120 012701 170074      MOV      @170074,R1       ;SPECIFY TO WAIT UP TO 60 MS FOR DSR CLEAR.
7022 031124 016702 151120 16# : MOV      STATA,R2         ;SPECIFY TO LOOK IN STAT REG FOR BIT TO CLR.
7023 031130 010477 151106      MOV      R4,@CSRA         ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
7024 031134 004767 165360      JSR      PC,WAIBIC        ;WAIT UP TO 60 MS FOR SIGNAL TO GO CLEAR.
7025 031140 103417            BCS      20#             ;SELECT NEXT LINE AND LOOP IF SIGNAL IS CLEAR.
7026 031142 017700 151102      MOV      @STATA,R0        ;GET THE STATUS REGISTER CONTENTS.
7027 031146 042705 057777      BIC      @57777,R5
7028 031152 040005            BIC      R0,R5           ;TEST FOR SIGNAL ONCE SET, BUT NOW CLEAR.
7029 031154 001011            BNE      20#             ;GO LOOP IF SIGNAL HAS GONE FROM SET TO CLR.
7030 031156            ;REPORT DTR MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
7031 031156 012767 017173 152722 18# : MOV      @7803,ERRNBR     ;SELECT THE ERROR NUMBER.
7032 031164 012767 012036 152720 MOV      @ER7801,ERRBLK   ;SELECT THE ERROR PRINT ROUTINE.
7033 031172 012701 007477      MOV      @EM7802,R1       ;SELECT THE ERROR MESSAGE.
7034 031176            ERROR
7035 031200 104460            TRAP    C#ERROR
7036 031202 005203 20# : INC      R3             ;SELECT THE NEXT LINE NUMBER.
7037 031206 020327 000010      CMP      R3,@NUMLNS      ;TEST FOR ALL LINES DONE.
7038 ;                                     BLT      14#             ;LOOP IF NOT ALL LINES DONE.
7039 031210 005067 151050 60# : CLR      CTRLCF         ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7040 031214 012700 000340      SETPRI  @PRI07           ;DISABLE ALL INTERRUPTS.
7041 ;                                     MOV      @PRI07,R0
7042 031222            TRAP    C#SPRI
031222 104401            L10042: TRAP    C#ETST

```

HARDWARE TEST - RTSMCS -

```

7044
7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056 031224
      031224
7057
7058
7059
7060 031224 032767 000002 151004
7061 031232 001002
7062 031234 000167 000464
7063 031240
      031240 012700 000240
      031244 104441
7064      000021
7065 031246 012767 000021 151012
7066 031254 012767 177777 151002
7067 031262 012767 000001 152614
7068 031270 012767 017335 152610
7069 031276 012767 007530 152604
7070
7071
7072
7073
7074
7075 031304 004767 162174
7076 031310 103402
7077 031312 000167 000406
7078
7079
7080
7081 031316 004767 161526
7082
7083
7084
7085
7086
7087
7088 031322 005003
7089 031324 010300
7090 031326 006300
7091 031330 036067 002370 150676
7092 031336 001465
7093
7094
7095
7096 031340 005000
7097 031342 012705 000377

```

```

.SBTTL HARDWARE TEST - 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
T17::
;*
;* ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGARED LOOPBACK MODE.
;*
;* BIT #BIT1,LOPBC ;CHECK TYPE OF LOOPBACK MODE SELECTED.
;* BNE 10
;* JMP 600 ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
10: SETPRI @PRI05 ;ALLOW LTC INTERRUPTS.
;*
;* TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
;* MOV @TNUM,TSTNUM ;SET UP THE TEST NUMBER. (79)
;* MOV @-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
;* MOV @1,ERRTYP ;SET ERROR TYPE IN ERROR TABLE.
;* MOV @7901,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
;* MOV @EM7901,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 30
;* JMP 600 ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
;*
;* SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
;*
30: 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.
;* MOV R3,R0
;* ASL R0
;* BIT BITTBL(R0),ACTLNS
;* BEQ 80 ;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 @MAPLNS,R5 ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.

```

HARDWARE TEST

- RTSMCS -

```

7098 031346 004767 165336      JSR    PC,WTWLNK      ;CLEAR ALL THE DUT RTS BITS.
7099 031352 012704 000074      MOV    #60.,R4
7100 031356 004767 162334      JSR    PC,DELAY       ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.
7101
7102      ; CHECK THAT AT LEAST ONE OF ASSOCIATED DCD OR CTS IS CLEAR AND RECORD STATES.
7103      ; -
7104 031362 116304 004044      MOV    TXRLNB(R3),R4  ;GET THE ASSOCIATED LINE NUMBER.
7105 031366 010477 150650      MOV    R4,BCSRA      ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
7106 031372 017705 150652      MOV    #STATA,R5     ;GET THE STATE OF THE ASSOCIATED DCD, CTS BITS.
7107 031376 012700 014000      MOV    #BIT12:BIT11,R0
7108 031402 040500      BIC    R5,R0         ;CHECK FOR BOTH DCD AND CTS SET.
7109 031404 001431      BEQ    #0            ;GO REPORT RTS IS BAD IF BOTH ARE SET.
7110
7111      ; SET THE RTS FOR THE SELECTED LINE AND WAIT FOR EITHER DCD OR CTS TO SET.
7112      ; -
7113 031406 010377 150630      MOV    R3,BCSRA      ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
7114 031412 052777 010000 150632  BIS    #BIT12,BLNCTRA ;SET THE SELECTED LINE RTS.
7115 031420 012701 130074      MOV    #130074,R1    ;SPECIFY TO WAIT UP TO 60 MS FOR CTS TO SET.
7116 031424 032705 010000      BIT    #BIT12,R5     ;CHECK PREVIOUS STATE OF DCD BIT.
7117 031430 001002      BNE    #0            ;GO USE CTS IF DCD BIT WAS NOT CLEAR.
7118 031432 012701 140074      MOV    #140074,R1    ;SPECIFY TO WAIT UP TO 60 MS FOR DCD SET.
7119 031436 016702 150606 40:   MOV    STATA,R2      ;SPECIFY TO LOOK IN STAT REG FOR BIT TO SET.
7120 031442 010477 150574      MOV    R4,BCSRA      ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
7121 031446 004767 165122      JSR    PC,WAIBIS     ;WAIT UP TO 60 MS FOR SIGNAL TO GO SET.
7122 031452 103417      BCS    #0            ;SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.
7123 031454 017700 150570      MOV    #STATA,R0     ;GET THE STATUS REGISTER CONTENTS.
7124 031460 042700 163777      BIC    #163777,R0    ;REMOVE ALL BUT THE DCD AND CTS BITS.
7125 031464 040500      BIC    R5,R0         ;TEST FOR SIGNAL ONCE CLEAR, BUT NOW SET.
7126 031466 001011      BNE    #0            ;GO LOOP IF SIGNAL HAS GONE FROM CLR TO SET.
7127 031470 60:   ;REPORT RTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
7128 031470 012767 017336 152410  MOV    #7902.,ERRNBR ;SELECT THE ERROR NUMBER.
7129 031476 012767 012036 152406  MOV    #ER7801,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
7130 031504 012701 007563      MOV    #EM7902,R1    ;SELECT THE ERROR MESSAGE.
7131 031510      ERROR              ; >>>>> ERROR <<<<<.
7132 031512 104460      TRAP    C#ERROR
7133 031514 005203 80:   INC    R3            ;SELECT THE NEXT LINE NUMBER.
7134 031514 020327 000010  CMP    R3,#NUMPLNS  ;TEST FOR ALL LINES DONE.
7135 031514 002701      BLT    #0            ;LOOP IF NOT ALL LINES DONE.
7136
7137      ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
7138      ; THIS LOOP SETS ALL THE RTSS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
7139      ; A RESPONSE ON THE ASSOCIATED CTS AND DCD SIGNALS.
7140      ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
7141      ; -
7141 031522 005003 100:  CLR    R3            ;CLEAR THE LINE COUNTER.
7142 031524 010300      MOV    R3,R0
7143 031526 006300      ASL    R0
7144 031530 036067 002370 150476  BIT    BITTBL(R0),ACTLNS
7145 031536 001466      BEQ    #160         ;DON'T TEST IF NOT ACTIVE LINE.
7146
7147      ; SET ALL THE DUT LNCTRL REGISTERS RTS BITS.
7148      ; -
7149 031540 012700 010000      MOV    #BIT12,R0     ;SPECIFY THAT RTS BITS ARE TO BE SET.
7150 031544 012705 000377      MOV    #MAPLNS,R5    ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
7151 031550 004767 165134      JSR    PC,WTWLNK     ;SET ALL THE DUT RTS BITS.
7152 031554 012704 000074      MOV    #60.,R4
7153 031560 004767 162132      JSR    PC,DELAY       ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.

```



HARDWARE TEST

- RTSMCS -

```

7154
7155 ; CHECK THAT AT LEAST ONE OF ASSOCIATED DCD OR CTS IS SET AND RECORD STATES.
7156 ;-
7157 031564 116304 004044      MOVB  TXRLNB(R3),R4      ;GET THE ASSOCIATED LINE NUMBER.
7158 031570 010477 150446      MOV   R4,@CSRA          ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
7159 031574 017705 150450      MOV   @STATA,R5         ;GET THE STATE OF THE ASSOCIATED DCD, CTS BITS.
7160 031600 010500
7161 031602 042700 163777      MOV   R5,R0
7162 031606 001431            BIC   @163777,R0        ;CHECK FOR BOTH DCD AND CTS CLEAR.
7163 ;-                               BEQ   14#                   ;GO REPORT RTS IS BAD IF BOTH ARE CLEAR.
7164 ; CLEAR THE RTS FOR THE SELECTED LINE AND WAIT FOR EITHER DCD OR CTS TO CLEAR.
7165 ;-
7166 031610 010377 150426      MOV   R3,@CSRA          ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
7167 031614 042777 010000 150430 BIC   @BIT12,@LNCTRA    ;CLEAR THE SELECTED LINE RTS.
7168 031622 012701 130074      MOV   @130074,R1        ;SPECIFY TO WAIT UP TO 60 MS FOR CTS TO CLEAR.
7169 031626 032705 010000      BIT   @BIT12,R5         ;CHECK PREVIOUS STATE OF DCD BIT.
7170 031632 001402            BEQ   12#                   ;GO USE CTS IF DCD BIT WAS NOT SET.
7171 031634 012701 140074      MOV   @140074,R1        ;SPECIFY TO WAIT UP TO 60 MS FOR DCD CLEAR.
7172 031640 016702 150404 12# : MOV   STATA,R2          ;SPECIFY TO LOOK IN STAT REG FOR BIT TO CLR.
7173 031644 010477 150372      MOV   R4,@CSRA          ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
7174 031650 004767 164644      JSR   PC,WAIBIC         ;WAIT UP TO 60 MS FOR SIGNAL TO GO CLEAR.
7175 031654 103417            BCS   16#                   ;SELECT NEXT LINE AND LOOP IF SIGNAL IS CLEAR.
7176 031656 017700 150366      MOV   @STATA,R0        ;GET THE STATUS REGISTER CONTENTS.
7177 031662 042705 163777      BIC   @163777,R5
7178 031666 040005            BIC   R0,R5                ;TEST FOR SIGNAL ONCE SET, BUT NOW CLEAR.
7179 031670 001011            BNE   16#                   ;GO LOOP IF SIGNAL HAS GONE FROM SET TO CLR.
7180 031672 14# : ;REPORT RTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
7181 031672 012767 017337 152206 MOV   @7903.,ERRNBR     ;SELECT THE ERROR NUMBER.
7182 031700 012767 012036 152204 MOV   @ER7801,ERRBLK    ;SELECT THE ERROR PRINT ROUTINE.
7183 031706 012701 007563      MOV   @EM7902,R1        ;SELECT THE ERROR MESSAGE.
7184 031712      ERROR                    ; >>>>> ERROR <<<<<<.
7185 031714 104460            TRAP   C#ERROR
7186 031716 005203 000010 16# : INC   R3                   ;SELECT THE NEXT LINE NUMBER.
7187 031722 020327 000010      CMP   R3,@NUMLNS       ;TEST FOR ALL LINES DONE.
7188 031722 002700            BLT   10#                   ;LOOP IF NOT ALL LINES DONE.
7189 031724 005067 150334 60# : CLR   CTRLCF            ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7190 031730      SETPRI @PRI07          ;DISABLE ALL INTERRUPTS.
7191 031730 012700 000340      MOV   @PRI07,R0
7192 031734 104441            TRAP   C#SPRI
7191
7192 031736      ENDTST
031736
031736 104401      L10043: TRAP   C#ETST

```

HARDWARE TEST

- DSRMS -

```

7194
7195
7196
7197
7198
7199
7200
7201
7202
7203
7204
7205
7206 031740
      031740
7207
7208
7209
7210 031740 032767 000002 150270
7211 031746 001002
7212 031750 000167 000400
7213 031754
      031754 012700 000240
      031760 104441
7214      000022
7215 031762 012767 000022 150276
7216 031770 012767 177777 150266
7217 031776 012767 000001 152100
7218 032004 012767 017501 152074
7219 032012 012767 007614 152070
7220
7221
7222
7223
7224
7225 032020 004767 161460
7226 032024 103402
7227 032026 000167 000322
7228
7229
7230
7231 032032 004767 161012
7232
7233
7234
7235
7236
7237
7238 032036 005003
7239 032040 010300
7240 032042 006300
7241 032044 036067 002370 150162
7242 032052 001450
7243
7244
7245
7246 032054 005000
7247 032056 012705 000377

```

```

.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
;
; T18::
; ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
;
; 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.
;
; TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
; MOV      @TNUM,TSTNUM     ;SET UP THE TEST NUMBER. (80)
; MOV      @-1,CTRLCF       ;INDICATE THAT WE ARE IN A TEST.
; MOV      @1,ERRTYP        ;SET ERROR TYPE IN ERROR TABLE.
; MOV      @8001,ERRNBR     ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
; MOV      @EM8001,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               ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
; MOV      @MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.

```

## HARDWARE TEST

- DSRMS -

```

7248 032062 004767 164622      JSR    PC,WTWLNLC      ;CLEAR ALL THE DUT DTR BITS.
7249 032066 012704 000050      MOV    #40.,R4
7250 032072 004767 161620      JSR    PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
7251
7252      ;+
7253      ; CHECK THAT THE SPECIFIED DSR IS CLEAR.
7254 032076 010377 150140      MOV    R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7255 032102 032777 100000 150140  BIT    #BIT15,@STATA
7256 032110 001020      BNE    8#              ;GO REPORT DSR IS BAD IF BIT IS NOT CLEAR.
7257
7258      ;+
7259      ; SET THE DTR FOR THE ASSOCIATED LINE.
7260      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED
7261      ; IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
7262 032112 116304 004044      MOVB   TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
7263 032116 010477 150120      MOV    R4,@CSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
7264 032122 052777 001000 150122  BIS    #BIT9,@LNCTRA   ;SET THE ASSOCIATED LINE DTR.
7265
7266      ;+
7267      ; CHECK THAT THE SELECTED LINE DSR IS ACTIVE.
7268 032130 010377 150106      MOV    R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7269 032134 012701 170050      MOV    #170050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
7270 032140 016702 150104      MOV    STATA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7271 032144 004767 164424      JSR    PC,WAIBIS       ;WAIT FOR DSR TO BECOME SET OR TIMEOUT.
7272 032150 103411      BCS    10#            ;SKIP ERROR REPORT IF SELECTED DSR IS SET.
7273
7274
7275 032152      8#:      ;REPORT DSR MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
7276 032152 012767 017502 151726  MOV    #9002.,ERRNBR   ;SELECT THE ERROR NUMBER.
7277 032160 012767 012036 151724  MOV    #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
7278 032166 012701 007652      MOV    #EM8002,R1      ;SELECT THE ERROR MESSAGE.
7279 032172      ERROR
7280 032172 104460      TRAP   C#ERROR
7281 032174 005203      10#:      INC    R3              ;SELECT THE NEXT LINE NUMBER.
7282 032176 020327 000010      CMP    R3,#NUMLNS     ;TEST FOR ALL LINES DONE.
7283 032202 002716      BLT    6#              ;LOOP IF NOT ALL LINES DONE.
7284
7285      ;+
7286      ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
7287      ; THIS LOOP SETS ALL THE DTRS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
7288      ; A RESPONSE ON THE SELECTED DSR SIGNAL.
7289      ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
7289 032204 005003      ;-
7290 032206 010300      CLR    R3              ;CLEAR THE LINE COUNTER.
7291 032210 006300      12#:      MOV    R3,R0
7292 032212 036067 002370 150014  ASL    R0
7293 032220 001451      BIT    BITTBL(R0),ACTLNS
7294      BEQ    16#          ;DON'T TEST IF NOT ACTIVE LINE.
7295
7296      ;+
7297      ; SET ALL THE DUT LNCTRL REGISTERS DTR BITS.
7297 032222 012700 001000      ;-
7298 032226 012705 000377      MOV    #BIT9,R0        ;SPECIFY THAT DTR BITS ARE TO BE SET.
7299 032232 004767 164452      MOV    #MAPLNS,R5      ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
7300 032236 012704 000050      JSR    PC,WTWLNLC     ;SET ALL THE DUT DTR BITS.
7301 032242 004767 161450      MOV    #40.,R4
7302      JSR    PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
7303
7303      ;+
7303      ; CHECK THAT THE SPECIFIED DSR IS SET.

```

HARDWARE TEST

- DSRMS -

```

7304
7305 032246 010377 147770      ;-      MOV    R3,@CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7306 032252 032777 100000 147770      BIT    #BIT15,@STATA
7307 032260 001420              BEQ    14#              ;GO REPORT DSR IS BAD IF BIT IS NOT SET.
7308
7309      ;+
7310      ; CLEAR THE DTR FOR THE ASSOCIATED LINE.
7311      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED
7312      ; IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
7313 032262 116304 004044      ;-      MOVVB  TXRLNB(R3),R4  ;GET THE ASSOCIATED LINE NUMBER.
7314 032266 010477 147750      MOV    R4,@CSRA      ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
7315 032272 042777 001000 147752      BIC    #BIT9,@LNCTRA ;CLEAR THE ASSOCIATED LINE DTR.
7316
7317      ;+
7318      ; CHECK THAT THE SELECTED LINE DSR IS CLEAR.
7319 032300 010377 147736      ;-      MOV    R3,@CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7320 032304 012701 170050      MOV    #170050,R1    ;PASS TIMEOUT OF 40 MILLI-SEC. AND BIT TO TEST.
7321 032310 016702 147734      MOV    STATA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7322 032314 004767 164200      JSR    PC,WAIBIC     ;WAIT FOR DSR TO BECOME CLEAR OR TIMEOUT.
7323 032320 103411              BCS    16#           ;SKIP ERROR REPORT IF SELECTED DSR IS CLEAR.
7324
7325 032322              14#: ;REPORT DSR MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
7326 032322 012767 017503 151556      MOV    #8003.,ERRNBR ;SELECT THE ERROR NUMBER.
7327 032330 012767 012036 151554      MOV    #ER7801,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
7328 032336 012701 007652      MOV    #EM8002,R1    ;SELECT THE ERROR MESSAGE.
7329 032342
7330 032342 104460              ERROR
7330 032344 005203              16#: INC    R3              ;SELECT THE NEXT LINE NUMBER. TRAP    C#ERROR
7331 032346 020327 000010      CMP    R3,#NUMLNS   ;TEST FOR ALL LINES DONE.
7332 032352 002715              BLT    12#           ;LOOP IF NOT ALL LINES DONE.
7333
7334 032354 005067 147704      60#: CLR    CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7335 032360      SETPRI #PRI07     ;DISABLE ALL INTERRUPTS.
7336 032360 012700 000340              MOV    #PRI07,RO
7337 032366      104441              TRAP   C#SPRI
032366
032366 104401              L10044: TRAP   C#ETST

```

HARDWARE TEST - RINGI -

```

7339
7340
7341
7342
7343
7344
7345
7346
7347
7348
7349
7350
7351 032370
      032370
7352
7353
7354
7355 032370 032767 000002 147640
7356 032376 001002
7357 032400 000167 000400
7358 032404
      032404 012700 000240
      032410 104441
7359
7360 032412 012767 000023 147646
7361 032420 012767 177777 147636
7362 032426 012767 000001 151450
7363 032434 012767 017645 151444
7364 032442 012767 007716 151440
7365
7366
7367
7368
7369
7370 032450 004767 161030
7371 032454 103402
7372 032456 000167 000322
7373
7374
7375
7376 032462 004767 160362
7377
7378
7379
7380
7381
7382
7383 032466 005003
7384 032470 010300
7385 032472 006300
7386 032474 036067 002370 147532
7387 032502 001450
7388
7389
7390
7391 032504 005000
7392 032506 012705 000377

```

```

.SBTTL HARDWARE TEST - RINGI -
;*****
;* - RING INDICATOR MODEM SIGNAL TEST -
;*
;* THIS TEST VERIFIES THAT THE RI 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 RI SIGNAL. THIS TEST IS PERFORMED ON ALL THE ACTIVE
;* LINES.
;*****
      BGNTST
      T19::
;+
; ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGARED LOOPBACK MODE.
;-
      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
      TRAP  C#SPRI
      TNUM  == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV    #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (81)
      MOV    #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
      MOV    #1,ERRTYP ;SET ERROR TYPE IN ERROR TABLE.
      MOV    #8101,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
      MOV    #EM8101,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 >>>> 8101 <<<<.
;-
      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 DTRS AND THEN SETS THEM INDIVIDUALLY AND WAITS FOR
; A RESPONSE ON THE ASSOCIATED RI 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 DTR BITS.
;-
      CLR    R0 ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
      MOV    #MAPLNS,R5 ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.

```

## HARDWARE TEST

- RINGI -

```

7393 032512 004767 164172          JSR    PC,WTWLNLC      ;CLEAR ALL THE DUT DTR BITS.
7394 032516 012704 000050          MOV    #40.,R4
7395 032522 004767 161170          JSR    PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
7396
7397          ;+
7398          ; CHECK THAT THE SPECIFIED RI IS CLEAR.
7399 032526 010377 147510          MOV    R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7400 032532 032777 020000 147510  BIT    #BIT13,@STATA
7401 032540 001020          BNE    8#              ;GO REPORT RI IS BAD IF BIT IS NOT CLEAR.
7402
7403          ;+
7404          ; SET THE DTR FOR THE ASSOCIATED LINE.
7405          ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED
7406          ; IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
7407 032542 116304 004044          MOV    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
7408 032546 010477 147470          MOV    R4,@CSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
7409 032552 052777 001000 147472  BIS    #BIT9,@LNCTRA   ;SET THE ASSOCIATED LINE DTR.
7410
7411          ;+
7412          ; CHECK THAT THE SELECTED LINE RI IS ACTIVE.
7413 032560 010377 147456          MOV    R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7414 032564 012701 150050          MOV    #150050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
7415 032570 016702 147454          MOV    STATA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7416 032574 004767 163774          JSR    PC,WAIBIS       ;WAIT FOR RI TO BECOME SET OR TIMEOUT.
7417 032600 103411          BCS    10#            ;SKIP ERROR REPORT IF SELECTED RI IS SET.
7418
7419
7420 032602          8#: ;REPORT RI MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
7421 032602 012767 017646 151276  MOV    #8102.,ERRNBR   ;SELECT THE ERROR NUMBER.
7422 032610 012767 012036 151274  MOV    #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
7423 032616 012701 007753          MOV    #EM8102,R1      ;SELECT THE ERROR MESSAGE.
7424 032622          ERROR
7425 032624 005203          10#: INC    R3              ;SELECT THE NEXT LINE NUMBER. TRAP    C#ERROR
7426 032626 020327 000010          CMP    R3,#NUMLNS     ;TEST FOR ALL LINES DONE.
7427 032632 002716          BLT    6#              ;LOOP IF NOT ALL LINES DONE.
7428
7429          ;+
7430          ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
7431          ; THIS LOOP SETS ALL THE DTRS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
7432          ; A RESPONSE ON THE SELECTED RI SIGNAL.
7433          ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
7434 032634 005003          ;-
7435 032636 010300          CLR    R3              ;CLEAR THE LINE COUNTER.
7436 032640 006300          12#: MOV    R3,R0
7437 032642 036067 002370 147364  ASL    R0
7438 032650 001451          BIT    BITTBL(R0),ACTLNS
7439          BEQ    16#              ;DON'T TEST IF NOT ACTIVE LINE.
7440          ;+
7441          ; SET ALL THE DUT LNCTRL REGISTERS DTR BITS.
7442 032652 012700 001000          ;-
7443 032656 012705 000377          MOV    #BIT9,R0        ;SPECIFY THAT DTR BITS ARE TO BE SET.
7444 032662 004767 164022          MOV    #MAPLNS,R5     ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
7445 032666 012704 000050          JSR    PC,WTWLNLC     ;SET ALL THE DUT DTR BITS.
7446 032672 004767 161020          MOV    #40.,R4
7447          JSR    PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
7448          ;+
7448          ; CHECK THAT THE SPECIFIED RI IS SET.

```

## HARDWARE TEST

- RINGI -

```

7449
7450 032676 010377 147340      ; -      MOV    R3,@CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7451 032702 032777 020000 147340      BIT    @BIT13,@STATA
7452 032710 001420              BEQ    14$              ;GO REPORT RI IS BAD IF BIT IS NOT SET.
7453
7454      ; +
7455      ; CLEAR THE DTR FOR THE ASSOCIATED LINE.
7456      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED
7457      ; IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
7458 032712 116304 004044      ; -      MOVB   TXRLNB(R3),R4      ;GET THE ASSOCIATED LINE NUMBER.
7459 032716 010477 147320      MOV    R4,@CSRA      ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
7460 032722 042777 001000 147322      BIC    @BIT9,@LNCTRA  ;CLEAR THE ASSOCIATED LINE DTR.
7461
7462      ; +
7463      ; CHECK THAT THE SELECTED LINE RI IS CLEAR.
7464 032730 010377 147306      ; -      MOV    R3,@CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7465 032734 012701 150050      MOV    @150050,R1     ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
7466 032740 016702 147304      MOV    STATA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7467 032744 004767 163550      JSR   PC,WAIBIC      ;WAIT FOR RI TO BECOME CLEAR OR TIMEOUT.
7468 032750 103411              BCS    16$              ;SKIP ERROR REPORT IF SELECTED RI IS CLEAR.
7469
7470 032752              14$: ;REPORT RI MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
7471 032752 012767 017647 151126      MOV    @8103.,ERRNBR ;SELECT THE ERROR NUMBER.
7472 032760 012767 012036 151124      MOV    @ER7801,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
7473 032766 012701 007753      MOV    @EM8102,R1    ;SELECT THE ERROR MESSAGE.
7474 032772
7475 032772 104460              ERROR
7475 032774 005203              TRAP   C$ERROR
7476 032776 020327 000010      16$:  INC    R3              ;SELECT THE NEXT LINE NUMBER.
7477 033002 002715      CMP    R3,@NUMLNS    ;TEST FOR ALL LINES DONE.
7478      BLT    12$              ;LOOP IF NOT ALL LINES DONE.
7479 033004 005067 147254      60$:  CLR    CTRLCF        ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7480 033010      SETPRI @PRI07        ;DISABLE ALL INTERRUPTS.
7481      MOV    @PRI07,R0
7482 033016      TRAP   C$SPRI
033016      ENDTST
033016 104401              L10045: TRAP   C$ETST

```

HARDWARE TEST - CTSMS -

```

7484 .SBTTL HARDWARE TEST - CTSMS -
7485 ;*****
7486 ;* - CLEAR TO SEND MODEM SIGNAL TEST -
7487 ;*
7488 ;* THIS TEST VERIFIES THAT THE CTS MODEM STATUS SIGNAL IS WORKING
7489 ;* CORRECTLY. IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR STAGGERED
7490 ;* LOOPBACK IS SPECIFIED. THIS TEST USES THE LOOPED BACK RTS SIGNALS
7491 ;* TO TEST THE CTS SIGNAL. THIS TEST IS PERFORMED ON ALL THE ACTIVE
7492 ;* LINES.
7493 ;*
7494 ;--*****
7495
7496 033020 BGNTST
033020
7497 ;+
7498 ; ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
7499 ;-
7500 033020 032767 000002 147210 BIT #BIT1,LOPCK ;CHECK TYPE OF LOOPBACK MODE SELECTED.
7501 033026 001002 BNE 2$
7502 033030 000167 000400 JMP 60$ ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
7503 033034 2$: SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
033034 012700 000240
033040 104441
7504 000024 TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
7505 033042 012767 000024 147216 MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (82)
7506 033050 012767 177777 147206 MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
7507 033056 012767 000001 151020 MOV #1,ERRTYP ;SET ERROR TYPE IN ERROR TABLE.
7508 033064 012767 020011 151014 MOV #8201,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
7509 033072 012767 010016 151010 MOV #EM8201,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
7510 ;+
7511 ; RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.
7512 ; CLEAR TX AND RX INTERRUPT ENABLE BITS.
7513 ; THIS SUBROUTINE REPORTS ERROR >>>> 8201 <<<<<.
7514 ;-
7515 033100 004767 160400 JSR PC,CLNRST ;RESET THE DUT.
7516 033104 103402 BCS 4$
7517 033106 000167 000322 JMP 60$ ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
7518 ;+
7519 ; SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.
7520 ;-
7521 033112 004767 157732 4$: JSR PC,ASLNTL ;SET UP THE ASSOCIATED LINE TABLES.
7522 ;+
7523 ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
7524 ; THIS LOOP CLEARS ALL THE RTS'S AND THEN SETS THEM INDIVIDUALLY AND WAITS FOR
7525 ; A RESPONSE ON THE ASSOCIATED CTS SIGNAL.
7526 ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
7527 ;-
7528 033116 005003 CLR R3 ;CLEAR THE LINE COUNTER.
7529 033120 010300 6$: MOV R3,R0
7530 033122 006300 ASL R0
7531 033124 036067 002370 147102 BIT BITBL(R0),ACTLNS
7532 033132 001450 BEQ 10$ ;DON'T TEST IF NOT ACTIVE LINE.
7533 ;+
7534 ; CLEAR ALL THE DUT LNCTRL REGISTERS RTS BITS.
7535 ;-
7536 033134 005000 CLR R0 ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
7537 033136 012705 000377 MOV #MAPLNS,R5 ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.

```



HARDWARE TEST - CTSMS -

```

7538 033142 004767 163542      JSR    PC,WTWLNLC      ;CLEAR ALL THE DUT RTS BITS.
7539 033146 012704 000050      MOV    #40.,R4
7540 033152 004767 160540      JSR    PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
7541
7542      ; CHECK THAT THE SPECIFIED CTS IS CLEAR.
7543
7544 033156 010377 147060      MOV    R3,BCSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7545 033162 032777 004000 147060  BIT    #BIT11,BSATA
7546 033170 001020                BNE    #0                ;GO REPORT CTS IS BAD IF BIT IS NOT CLEAR.
7547
7548      ; SET THE RTS FOR THE ASSOCIATED LINE.
7549      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, RTS WILL NOT HAVE BEEN TESTED
7550      ; IN THE RTS TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
7551
7552 033172 116304 004044      MOV    TXRLNB(R3),R4    ;GET THE ASSOCIATED LINE NUMBER.
7553 033176 010477 147040      MOV    R4,BCSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
7554 033202 052777 010000 147042  BIS    #BIT12,BLNCTRA  ;SET THE ASSOCIATED LINE RTS.
7555
7556      ; CHECK THAT THE SELECTED LINE CTS IS ACTIVE.
7557
7558 033210 010377 147026      MOV    R3,BCSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7559 033214 012701 130050      MOV    #130050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
7560 033220 016702 147024      MOV    STATA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7561 033224 004767 163344      JSR    PC,WAIBIS       ;WAIT FOR CTS TO BECOME SET OR TIMEOUT.
7562 033230 103411                BCS    10#              ;SKIP ERROR REPORT IF SELECTED CTS IS SET.
7563
7564
7565 033232                8#:  ;REPORT CTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
7566 033232 012767 020012 150646  MOV    #8202.,ERRNBR   ;SELECT THE ERROR NUMBER.
7567 033240 012767 012036 150644  MOV    #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
7568 033246 012701 010054                MOV    #EM8202,R1      ;SELECT THE ERROR MESSAGE.
7569 033252                ERROR
7570 033254 005203                TRAP  C!ERROR
7571 033256 020327 000010      10#:  INC    R3          ;SELECT THE NEXT LINE NUMBER.
7572 033262 002716                CMP    R3,#NUMLNS     ;TEST FOR ALL LINES DONE.
7573                BLT    #6#          ;LOOP IF NOT ALL LINES DONE.
7574
7575      ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
7576      ; THIS LOOP SETS ALL THE RTSS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
7577      ; A RESPONSE ON THE SELECTED CTS SIGNAL.
7578      ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
7579
7579 033264 005003                ;-      CLR    R3          ;CLEAR THE LINE COUNTER.
7580 033266 010300      12#:  MOV    R3,R0
7581 033270 006300                ASL    R0
7582 033272 036067 002370 146734  BIT    BITTBL(R0),ACTLNS
7583 033300 001451                BEQ    #16#           ;DON'T TEST IF NOT ACTIVE LINE.
7584
7585      ; SET ALL THE DUT LNCTRL REGISTERS RTS BITS.
7586
7587 033302 012700 010000      ;-      MOV    #BIT12,R0    ;SPECIFY THAT RTS BITS ARE TO BE SET.
7588 033306 012705 000377      MOV    #MAPLNS,R5     ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
7589 033312 004767 163372      JSR    PC,WTWLNLC     ;SET ALL THE DUT RTS BITS.
7590 033316 012704 000050      MOV    #40.,R4
7591 033322 004767 160370      JSR    PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
7592
7593      ; CHECK THAT THE SPECIFIED CTS IS SET.

```

HARDWARE TEST - CTSMS -

```

7594
7595 033326 010377 146710      ;
7596 033332 032777 004000 146710      MOV   R3,BCSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7597 033340 001420                BIT   #BIT11,@STATA ;GO REPORT CTS IS BAD IF BIT IS NOT SET.
7598                                BEQ   14#
7599                                ;*
7600                                ; CLEAR THE RTS FOR THE ASSOCIATED LINE.
7601                                ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, RTS WILL NOT HAVE BEEN TESTED
7602                                ; IN THE RTS TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
7603 033342 116304 004044      ;
7604 033346 010477 146670      MOVB  TXRLNB(R3),R4  ;GET THE ASSOCIATED LINE NUMBER.
7605 033352 042777 010000 146672      MOV   R4,BCSRA      ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
7606                                BIC   #BIT12,@LNCTRA ;CLEAR THE ASSOCIATED LINE RTS.
7607                                ;*
7608                                ; CHECK THAT THE SELECTED LINE CTS IS CLEAR.
7609 033360 010377 146656      ;
7610 033364 012701 130050      MOV   R3,BCSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7611 033370 016702 146654      MOV   #130050,R1    ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
7612 033374 004767 163120      MOV   STATA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7613 033400 103411                JSR   PC,WAIBIC     ;WAIT FOR CTS TO BECOME CLEAR OR TIMEOUT.
7614                                BCS   16#           ;SKIP ERROR REPORT IF SELECTED CTS IS CLEAR.
7615 033402                14#: ;REPORT CTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
7616 033402 012767 020013 150476      MOV   #8203,ERRNBR ;SELECT THE ERROR NUMBER.
7617 033410 012767 012036 150474      MOV   #ER7801,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
7618 033416 012701 010054      MOV   #EM8202,R1   ;SELECT THE ERROR MESSAGE.
7619 033422                ERROR
7620 033424 104460                TRAP   C#ERROR
7621 033426 005203                16#: INC   R3           ;SELECT THE NEXT LINE NUMBER.
7622 033432 020327 000010      CMP   R3,#NUMLNS  ;TEST FOR ALL LINES DONE.
7623                                BLT   12#           ;LOOP IF NOT ALL LINES DONE.
7624 033434 005067 146624      60#: CLR   CTRLCF     ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7625 033440 012700 000340      SETPRI #PRI07     ;DISABLE ALL INTERRUPTS.
7626                                MOV   #PRI07,R0
7627 033446                TRAP   C#SPRI
7628 033446 104401                L10046: TRAP   C#ETST
7629 033446

```

HARDWARE TEST - DCDMS -

```

7629
7630
7631
7632
7633
7634
7635
7636
7637
7638
7639
7640
7641 033450
      033450
7642
7643
7644
7645 033450 032767 000002 146560
7646 033456 001002
7647 033460 000167 000400
7648 033464
      033464 012700 000240
      033470 104441
7649      000025
7650 033472 012767 000025 146566
7651 033500 012767 177777 146556
7652 033506 012767 000001 150370
7653 033514 012767 020155 150364
7654 033522 012767 010120 150360
7655
7656
7657
7658
7659
7660 033530 004767 157750
7661 033534 103402
7662 033536 000167 000322
7663
7664
7665
7666 033542 004767 157302
7667
7668
7669
7670
7671
7672
7673 033546 005003
7674 033550 010300
7675 033552 006300
7676 033554 036067 002370 146452
7677 033562 001450
7678
7679
7680
7681 033564 005000
7682 033566 012705 000377

```

```

.SBTTL HARDWARE TEST - DCDMS -
;*****
;* - DATA CARRIER DETECTED MODEM SIGNAL TEST -
;*
;* THIS TEST VERIFIES THAT THE DCD 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 DCD SIGNAL. THIS TEST IS PERFORMED ON ALL THE ACTIVE
;* LINES.
;*****
      BGNTST
      T21::
;* ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
;*
      BIT      @BIT1,LOPBCK      ;CHECK TYPE OF LOOPBACK MODE SELECTED.
      BNE      2@
      JMP      60@
;*EXIT THIS TEST IF IN INTERNAL LOOPBACK.
;*ALLOW LTC INTERRUPTS.
2@:   SETPRI   @PRI05
      MOV      @PRI05,R0
      TRAP    C@SPRI
      TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV      @TNUM,TSTNUM     ;SET UP THE TEST NUMBER. (83)
      MOV      @-1,CTRLCF       ;INDICATE THAT WE ARE IN A TEST.
      MOV      @1,ERRTYP        ;SET ERROR TYPE IN ERROR TABLE.
      MOV      @8301,ERRNBR     ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
      MOV      @EM8301,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 >>>> 8301 <<<<<.
;*
      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 WAITS FOR
;* A RESPONSE ON THE ASSOCIATED DCD 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      @MAPLNS,R5       ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.

```

## HARDWARE TEST

- DCDMS -

```

7683 033572 004767 163112      JSR    PC,WTWLNLC      ;CLEAR ALL THE DUT RTS BITS.
7684 033576 012704 000050      MOV    #40.,R4
7685 033602 004767 160110      JSR    PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
7686
7687      ;*
7688      ; CHECK THAT THE SPECIFIED DCD IS CLEAR.
7689 033606 010377 146430      MOV    R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7690 033612 032777 010000 146430  BIT    #BIT12,@STATA
7691 033620 001020      BNE    #0              ;GO REPORT DCD IS BAD IF BIT IS NOT CLEAR.
7692
7693      ;*
7694      ; SET THE RTS FOR THE ASSOCIATED LINE.
7695      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, RTS WILL NOT HAVE BEEN TESTED
7696      ; IN THE RTS TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
7697 033622 116304 004044      MOV    TXRLNB(R3),R4   ;GET THE ASSOCIATED LINE NUMBER.
7698 033626 010477 146410      MOV    R4,@CSRA        ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
7699 033632 052777 010000 146412  BIS    #BIT12,@LNCTRA ;SET THE ASSOCIATED LINE RTS.
7700
7701      ;*
7702      ; CHECK THAT THE SELECTED LINE DCD IS ACTIVE.
7703 033640 010377 146376      MOV    R3,@CSRA        ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7704 033644 012701 140050      MOV    #140050,R1      ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
7705 033650 016702 146374      MOV    STATA,R2        ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7706 033654 004767 162714      JSR    PC,WAIBIS       ;WAIT FOR DCD TO BECOME SET OR TIMEOUT.
7707 033660 103411      BCS    10#            ;SKIP ERROR REPORT IF SELECTED DCD IS SET.
7708
7709
7710 033662      8# : ;REPORT DCD MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
7711 033662 012767 020156 150216  MOV    #8302.,ERRNBR   ;SELECT THE ERROR NUMBER.
7712 033670 012767 012036 150214  MOV    #ER7801,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
7713 033676 012701 010156      MOV    #EM8302,R1      ;SELECT THE ERROR MESSAGE.
7714 033702      ERROR
7715 033704 104460      TRAP    C#ERROR
7716 033706 005203      10# : INC    R3          ;SELECT THE NEXT LINE NUMBER.
7717 033712 020327 000010      CMP    R3,#NUMLNS     ;TEST FOR ALL LINES DONE.
7718      BLT    6#          ;LOOP IF NOT ALL LINES DONE.
7719
7720      ;*
7721      ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
7722      ; THIS LOOP SETS ALL THE RTSS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
7723      ; A RESPONSE ON THE SELECTED DCD SIGNAL.
7724      ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
7725 033714 005003      ;-
7726 033716 010300      CLR    R3              ;CLEAR THE LINE COUNTER.
7727 033720 006300      12# : MOV    R3,R0
7728 033722 036067 002370 146304  ASL    R0
7729 033730 001451      BIT    BITTBL(R0),ACTLNS
7730      BEQ    16#          ;DON'T TEST IF NOT ACTIVE LINE.
7731
7732      ;*
7733      ; SET ALL THE DUT LNCTRL REGISTERS RTS BITS.
7734 033732 012700 010000      ;-
7735 033736 012705 000377      MOV    #BIT12,R0      ;SPECIFY THAT RTS BITS ARE TO BE SET.
7736 033742 004767 162742      MOV    #MAPLNS,R5     ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
7737 033746 012704 000050      JSR    PC,WTWLNLC     ;SET ALL THE DUT RTS BITS.
7738 033752 004767 157740      MOV    #40.,R4
7739      JSR    PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
7740
7741      ;*
7742      ; CHECK THAT THE SPECIFIED DCD IS SET.

```

HARDWARE TEST - DCDMS -

```

7739
7740 033756 010377 146260      ;-      MOV    R3,@CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7741 033762 032777 010000 146260      BIT    @BIT12,@STATA
7742 033770 001420                BEQ    14:              ;GO REPORT DCD IS BAD IF BIT IS NOT SET.
7743
7744      ;+
7745      ; CLEAR THE RTS FOR THE ASSOCIATED LINE.
7746      ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, RTS WILL NOT HAVE BEEN TESTED
7747      ; IN THE RTS TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
7748 033772 116304 004044      ;-      MOVB   TXRLNB(R3),R4    ;GET THE ASSOCIATED LINE NUMBER.
7749 033776 010477 146240      MOV    R4,@CSRA      ;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
7750 034002 042777 010000 146242      BIC    @BIT12,@LNCTRA ;CLEAR THE ASSOCIATED LINE RTS.
7751
7752      ;+
7753      ; CHECK THAT THE SELECTED LINE DCD IS CLEAR.
7754 034010 010377 146226      ;-      MOV    R3,@CSRA      ;SET IND.ADR.REG FIELD TO SELECTED LINE.
7755 034014 012701 140050      MOV    @140050,R1    ;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
7756 034020 016702 146224      MOV    STATA,R2      ;PASS THE ADDRESS OF THE REGISTER TO TEST.
7757 034024 004767 162470      JSR    PC,WAIBIC     ;WAIT FOR DCD TO BECOME CLEAR OR TIMEOUT.
7758 034030 103411                BCS    16:              ;SKIP ERROR REPORT IF SELECTED DCD IS CLEAR.
7759
7760 034032                14: ;REPORT DCD MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
7761 034032 012767 020157 150046      MOV    @8303.,ERRNBR ;SELECT THE ERROR NUMBER.
7762 034040 012767 012036 150044      MOV    @ER7801,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
7763 034046 012701 010156      MOV    @EM8302,R1    ;SELECT THE ERROR MESSAGE.
7764 034052
7765 034052 104460                ERROR
7765 034054 005203                16: ;SELECT THE NEXT LINE NUMBER. TRAP    C#ERROR
7766 034056 020327 000010      INC    R3            ;TEST FOR ALL LINES DONE.
7767 034062 002715                CMP    R3,@NUMLNS    ;LOOP IF NOT ALL LINES DONE.
7768
7769 034064 005067 146174      60: ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7770 034070 012700 000340      CLR    CTRLCF        ;DISABLE ALL INTERRUPTS.
7771 034074 104441                SETPRI @PRI07
7771
7772 034076                ENDTST
7772 034076
7772 034076 104401                L10047: TRAP    C#ETST

```

HARDWARE TEST - DTRINT -

```

7774
7775
7776
7777
7778
7779
7780
7781
7782
7783
7784
7785 034100
      034100
7786
7787
7788
7789 034100 032767 000002 146130
7790 034106 001002
7791 034110 000167 000360
7792 034114
      034114 012700 000240
      034120 104441
7793      000026
7794 034122 012767 000026 146136
7795 034130 012767 177777 146126
7796 034136 012767 000001 147740
7797 034144 012767 020321 147734
7798 034152 012767 010222 147730
7799
7800
7801
7802
7803
7804 034160 004767 157320
7805 034164 103402
7806 034166 000167 000302
7807
7808
7809
7810 034172 004767 156652
7811
7812
7813
7814
7815
7816
7817 034176 005003
7818 034200 010300
7819 034202 006300
7820 034204 036067 002370 146022
7821 034212 001444
7822
7823
7824
7825 034214 005000
7826 034216 012705 000377
7827 034222 004767 162462

```

```

.SBTTL HARDWARE TEST - DTRINT -
;*****
;* - DATA TERMINAL READY SIGNAL INTERACTIONS TEST -
;*
;* THIS TEST VERIFIES THAT THE DTR SIGNAL (AND THE LOOPED BACK DSR AND
;* RI 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
      T22::
;*
;* ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGARED LOOPBACK MODE.
;*
;* 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,RO
      TRAP 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.
;*
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 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,RO
      ASL RO
      BIT BITTBL(RO),ACTLNS
      BEQ 8# ;DON'T TEST IF NOT ACTIVE LINE.
;*
;* CLEAR ALL THE DUT LNCTRL REGISTERS DTR BITS.
;*
;* CLR RO ;SPECIFY THAT ALL LNCTRL BITS TO BE CLEARED.
;* MOV #MAPLNS,R5 ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
;* JSR PC,WTWLNLC ;CLEAR ALL THE DUT DTR BITS.

```

## HARDWARE TEST

- DTRINT -

```

7828 034226 012704 000050          MOV    #40.,R4
7829 034232 004767 157460          JSR    PC,DELAY          ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
7830                                ;+
7831                                ; RECORD THE STATES OF THE MODEM STATUS SIGNALS.
7832                                ;-
7833 034236 004767 161346          JSR    PC,SAVMST        ;SAVE THE PRESENT MODEM STATUS STATES.
7834                                ;+
7835                                ; SET THE DTR FOR THE SELECTED LINE.
7836                                ;-
7837 034242 010377 145774          MOV    R3,@CSRA        ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
7838 034246 052777 001000 145776  BIS    #BIT9,@LNCTRA   ;SET THE SELECTED LINE DTR.
7839 034254 012704 000050          MOV    #40.,R4
7840 034260 004767 157432          JSR    PC,DELAY        ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
7841                                ;+
7842                                ; CHECK THE PRESENT DUT STAT REGISTER CONTENTS AGAINST PREVIOUS.
7843                                ; IF ANY UNDESIRED CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.
7844                                ;-
7845 034264 116301 004044          MOV    TXRLNB(R3),R1   ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
7846 034270 012702 120000          MOV    #BIT15!BIT13,R2 ;IGNORE DSR AND RI ON ASSOCIATED LINE.
7847 034274 004767 157250          JSR    PC,CMPMST       ;COMPARE OLD AND NEW STAT CONTENTS.
7848 034300 103411                    BCS    #0               ;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
7849                                ;REPORT INTERACTIONS FOUND BETWEEN DTR FOR LINE NN AND THE FOLLOWING SIGNALS:
7850 034302 012767 020322 147576  MOV    #8402.,ERRNBR   ;SELECT THE ERROR NUMBER.
7851 034310 012767 012064 147574  MOV    #ER8401,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
7852 034316 012701 010276          MOV    #EM8402,R1     ;SELECT THE DTR ERROR MESSAGES.
7853 034322                                ERROR                    ;ER8401 USES R1, R2, AND R3 VALUES.
7854                                ;+
7855                                ; SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.
7856                                ;-
7857 034324 005203                    80:    INC    R3          ;SELECT THE NEXT LINE NUMBER.
7858 034326 020327 000010          CMP    R3,@NUMLNS     ;TEST FOR ALL LINES DONE.
7859 034332 002722                    BLT    #0             ;LOOP IF NOT ALL LINES DONE.
7860                                ;+
7861                                ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
7862                                ; THIS LOOP SETS ALL THE DTRS AND THEN CLEARS THEM INDIVIDUALLY AND CHECKS
7863                                ; FOR ANY RESPONSES ON SIGNALS OTHER THAN THE ASSOCIATED RI AND DSR SIGNALS.
7864                                ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
7865                                ;-
7866 034334 005003                    100:   CLR    R3              ;CLEAR THE LINE COUNTER.
7867 034336 010300                    MOV    R3,R0
7868 034340 006300                    ASL    R0
7869 034342 036067 002370 145664  BIT    BITTBL(R0),ACTLNS
7870 034350 001445                    BEQ    #120           ;DON'T TEST IF NOT ACTIVE LINE.
7871                                ;+
7872                                ; SET ALL THE DUT LNCTRL REGISTERS DTR BITS.
7873                                ;-
7874 034352 012700 001000          MOV    #BIT9,R0       ;SPECIFY THAT DTR BITS ARE TO BE SET.
7875 034356 012705 000377          MOV    #MAPLNS,R5     ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
7876 034362 004767 162322          JSR    PC,WTWLNLC     ;SET ALL THE DUT DTR BITS.
7877 034366 012704 000050          MOV    #40.,R4
7878 034372 004767 157320          JSR    PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
7879                                ;+
7880                                ; RECORD THE STATES OF THE MODEM STATUS SIGNALS.
7881                                ;-
7882 034376 004767 161206          JSR    PC,SAVMST        ;SAVE THE PRESENT MODEM STATUS STATES.
7883                                ;+

```

HARDWARE TEST

- DTRINT -

```

7884 ; CLEAR THE DTR FOR THE SELECTED LINE.
7885 ; -
7886 034402 010377 145634      MOV    R3,@CSRA      ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
7887 034406 042777 001000 145636  BIC    @BIT9,@LNCTRA ;CLEAR THE SELECTED LINE DTR.
7888 034414 012704 000050      MOV    @40.,R4
7889 034420 004767 157272      JSR    PC,DELAY      ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
7890 ; +
7891 ; CHECK THE PRESENT DUT STAT REGISTER CONTENTS AGAINST PREVIOUS.
7892 ; IF ANY UNDESIRED CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.
7893 ; -
7894 034424 116301 004044      MOVB   TXRLNB(R3),R1 ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
7895 034430 012702 120000      MOV    @BIT15!BIT13,R2 ;IGNORE DSR AND RI ON ASSOCIATED LINE.
7896 034434 004767 157110      JSR    PC,CMFMST    ;COMPARE OLD AND NEW STAT CONTENTS.
7897 034440 103411      BCS    12$          ;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
7898 ;REPORT INTERACTIONS FOUND BETWEEN DTR FOR LINE NN AND THE FOLLOWING SIGNALS:
7899 034442 012767 020323 147436  MOV    @8403.,ERRNBR ;SELECT THE ERROR NUMBER.
7900 034450 012767 012064 147434  MOV    @ER8401,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
7901 034456 012701 010276      MOV    @EM8402,R1   ;SELECT THE DTR ERROR MESSAGES.
7902 034462      ERROR      ;ER8401 USES R1, R2, AND R3 VALUES.
                                TRAP    C#ERROR
7903 ; +
7904 ; SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.
7905 ; -
7906 034464 005203      12$: INC    R3          ;SELECT THE NEXT LINE NUMBER.
7907 034466 020327 000010      CMP    R3,@NUMLNS  ;TEST FOR ALL LINES DONE.
7908 034472 002721      BLT    10$          ;LOOP IF NOT ALL LINES DONE.
7909
7910 034474 005067 145564      60$: CLR    CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7911 034500      SETPRI @PRI07    ;DISABLE ALL INTERRUPTS.
                                MOV    @PRI07,R0
                                TRAP    C#SPRI
7912
7913 034506      ENDTST
                                L10050: TRAP    C#ETST
                                034506 104401

```





## HARDWARE TEST

- RTSINT -

```

7969 034636 012704 000050          MOV    #40.,R4
7970 034642 004767 157050          JSR    PC,DELAY          ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
7971                                     ;+
7972                                     ; RECORD THE STATES OF THE MODEM STATUS SIGNALS.
7973                                     ;-
7974 034646 004767 160736          JSR    PC,SAVMST        ;SAVE THE PRESENT MODEM STATUS STATES.
7975                                     ;+
7976                                     ; SET THE RTS FOR THE SELECTED LINE.
7977                                     ;-
7978 034652 010377 145364          MOV    R3,@CSRA        ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
7979 034656 052777 010000 145366  BIS    #BIT12,@LNCTRA  ;SET THE SELECTED LINE RTS.
7980 034664 012704 000050          MOV    #40.,R4
7981 034670 004767 157022          JSR    PC,DELAY        ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
7982                                     ;+
7983                                     ; CHECK THE PRESENT DUT STAT REGISTER CONTENTS AGAINST PREVIOUS.
7984                                     ; IF ANY UNDESIRED CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.
7985                                     ;-
7986 034674 116301 004044          MOV    TXRLNB(R3),R1   ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
7987 034700 012702 014000          MOV    #BIT12!BIT11,R2 ;IGNORE DCD AND CTS ON ASSOCIATED LINE.
7988 034704 004767 156640          JSR    PC,CMPMST       ;COMPARE OLD AND NEW STAT CONTENTS.
7989 034710 103411                    BCS    #0              ;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
7990                                     ;REPORT INTERACTIONS FOUND BETWEEN RTS FOR LINE NN AND THE FOLLOWING SIGNALS:
7991 034712 012767 020466 147166  MOV    #8502.,ERRNBR   ;SELECT THE ERROR NUMBER.
7992 034720 012767 012064 147164  MOV    #ER8401,ERRBLK  ;SELECT THE ERROR PRINT ROUTINE.
7993 034726 012701 010375          MOV    #EM8502,R1     ;SELECT THE RTS ERROR MESSAGES.
7994 034732                    ERROR          ;ER1901 USES R1, R2, AND R3 VALUES.
7995                                     ;+
7996                                     ; SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.
7997                                     ;-
7998 034734 005203                    INC    R3              ;SELECT THE NEXT LINE NUMBER.
7999 034736 020327 000010          CMP    R3,#NUMLNS     ;TEST FOR ALL LINES DONE.
8000 034742 002722                    BLT    #6              ;LOOP IF NOT ALL LINES DONE.
8001                                     ;+
8002                                     ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
8003                                     ; THIS LOOP SETS ALL THE RTSS AND THEN CLEARS THEM INDIVIDUALLY AND CHECKS
8004                                     ; FOR ANY RESPONSES ON SIGNALS OTHER THAN THE ASSOCIATED DCD AND CTS SIGNALS.
8005                                     ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
8006                                     ;-
8007 034744 005003                    CLR    R3              ;CLEAR THE LINE COUNTER.
8008 034746 010300          10#:  MOV    R3,R0
8009 034750 006300                    ASL    R0
8010 034752 036067 002370 145254  BIT    BITTBL(R0),ACTLNS
8011 034760 001445                    BEQ    #12             ;DON'T TEST IF NOT ACTIVE LINE.
8012                                     ;+
8013                                     ; SET ALL THE DUT LNCTRL REGISTERS RTS BITS.
8014                                     ;-
8015 034762 012700 010000          MOV    #BIT12,R0      ;SPECIFY THAT RTS BITS ARE TO BE SET.
8016 034766 012705 000377          MOV    #MAPLNS,R5    ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
8017 034772 004767 161712          JSR    PC,WTWLNLC    ;SET ALL THE DUT RTS BITS.
8018 034776 012704 000050          MOV    #40.,R4
8019 035002 004767 156710          JSR    PC,DELAY        ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
8020                                     ;+
8021                                     ; RECORD THE STATES OF THE MODEM STATUS SIGNALS.
8022                                     ;-
8023 035006 004767 160576          JSR    PC,SAVMST      ;SAVE THE PRESENT MODEM STATUS STATES.
8024                                     ;+

```

## HARDWARE TEST

- RTSINT -

```

8025 ; CLEAR THE RTS FOR THE SELECTED LINE.
8026 ; -
8027 035012 010377 145224 ; MOV R3,@CSRA ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
8028 035016 042777 010000 145226 ; BIC #BIT12,@LNCTRA ;CLEAR THE SELECTED LINE RTS.
8029 035024 012704 000050 ; MOV #40.,R4
8030 035030 004767 156662 ; JSR PC,DELAY ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.
8031 ; +
8032 ; CHECK THE PRESENT DUT STAT REGISTER CONTENTS AGAINST PREVIOUS.
8033 ; IF ANY UNDESIRED CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.
8034 ; -
8035 035034 116301 004044 ; MOVB TXRLNB(R3),R1 ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
8036 035040 012702 014000 ; MOV #BIT12!BIT11,R2 ;IGNORE DCD AND CTS ON ASSOCIATED LINE.
8037 035044 004767 156500 ; JSR PC,CMPMST ;COMPARE OLD AND NEW STAT CONTENTS.
8038 035050 103411 ; BCS 12# ;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.
8039 ;REPORT INTERACTIONS FOUND BETWEEN RTS FOR LINE NN AND THE FOLLOWING SIGNALS:
8040 035052 012767 020467 147026 ; MOV #8503.,ERRNBR ;SELECT THE ERROR NUMBER.
8041 035060 012767 012064 147024 ; MOV #ER8401,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
8042 035066 012701 010375 ; MOV #EM8502,R1 ;SELECT THE RTS ERROR MESSAGES.
8043 035072 104460 ; ERROR ;ER1901 USES R1, R2, AND R3 VALUES.
; TRAP C#ERROR
8044 ; +
8045 ; SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.
8046 ; -
8047 035074 005203 ; 12#: INC R3 ;SELECT THE NEXT LINE NUMBER.
8048 035076 020327 000010 ; CMP R3,#NUMLNS ;TEST FOR ALL LINES DONE.
8049 035102 002721 ; BLT 10# ;LOOP IF NOT ALL LINES DONE.
8050
8051 035104 005067 145154 ; 60#: CLR CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
8052 035110 ; SETPRI #PRI07 ;DISABLE ALL INTERRUPTS.
; MOV #PRI07,R0
; TRAP C#SPRI
8053
8054 035116 ; ENDTST
; L10051: TRAP C#ETST
035116 104401

```

HARDWARE TEST - TXLNS -

8056  
8057  
8058  
8059  
8060  
8061  
8062  
8063  
8064  
8065  
8066  
8067  
8068  
8069 035120  
035120  
8070  
8071  
8072  
8073 035120 032767 000002 145110  
8074 035126 001002  
8075 035130 000167 000302  
8076 035134  
035134 012700 000240  
035140 104441  
8077 000030  
8078 035142 012767 000030 145116  
8079 035150 012767 177777 145106  
8080 035156 012767 000001 146720  
8081 035164 012767 020631 146714  
8082 035172 012767 010401 146710  
8083  
8084  
8085  
8086  
8087  
8088 035200 004767 156300  
8089 035204 103402  
8090 035206 000167 000224  
8091  
8092  
8093  
8094 035212 004767 155632  
8095  
8096  
8097  
8098  
8099 035216 012705 000377  
8100 035222 005000  
8101 035224 004767 161460  
8102 035230 012700 156430  
8103 035234 004767 161500  
8104  
8105  
8106  
8107 035240 016705 144770  
8108 035244 004767 160760  
8109

```

.SBTTL HARDWARE TEST - TXLNS -
;*****
;* - TX LINES TEST -
;*
;* THIS TEST VERIFIES THAT THE TX LINES AND RX LINES ARE WORKING CORRECTLY
;* THROUGH THE DEVICE CABLES, DISTRIBUTION PANEL (IF INSTALLED), AND
;* LOOPBACK CONNECTOR(S). IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR
;* STAGGERED LOOPBACK IS SPECIFIED. THIS TEST SENDS A CHARACTER ON EACH
;* ACTIVE LINE AND VERIFIES THAT THE PROPER CHARACTER IS DETECTED ON EACH
;* RECEIVE LINE.
;*****
      BGNTST
                                          T24::
;+
; ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGARED LOOPBACK MODE.
;-
      BIT    #BIT1,LOPBCK    ;CHECK TYPE OF LOOPBACK MODE SELECTED.
      BNE   2$
      JMP   60$              ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
2$:    SETPRI #PRI05        ;ALLOW LTC INTERRUPTS.
                                          MOV    #PRI05,R0
                                          TRAP  C$SPRI
      TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV   #TNUM,TSTNUM    ;SET UP THE TEST NUMBER. (86)
      MOV   #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
      MOV   #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
      MOV   #8601,ERRNBR    ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
      MOV   #EM8601,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 >>>> 8601 <<<<<.
;-
      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 TRANSMISSION AN RECEPTION PARAMETERS FOR ALL LINES.
; 9600 BAUD,8 CHAR,1 STOPBIT,NO PARITY.
;-
      MOV   #MAPLNS,R5     ;INDICATE ALL LINES TO BE SET UP.
      CLR  R0              ;SPECIFY ALL LNCTRL BITS TO BE CLEARED.
      JSR  PC,WTWLNLC      ;CLEAR ALL LNCTRL BITS.
      MOV  #156430,R0      ;SET UP BAUD RATE,ETC.
      JSR  PC,WTWLPR       ;SET COMMUNICATION PARAMETERS ON ALL LINES.
;+
; ENABLE TRANSMITTERS ON ALL ACTIVE LINES.
;-
      MOV  ACTLNS,R5       ;PASS ACTIVE LINE BIT MAP.
      JSR  PC,TXENBL      ;ENABLE TRANSMISSIONS ON ALL LINES.
;+

```

HARDWARE TEST

- TXLNS -

```

8110 ; ENABLE RECEPTION ON ALL LINES ASSOCIATED WITH ACTIVE LINES.
8111 ;-
8112 035250 016705 144760 MOV ACTLNS,R5 ;GET ACTIVE LINES BIT MAP TO CONVERT.
8113 035254 004767 156362 JSR PC,CONMAP ;CONVERT TO ASSOCIATED LINE MAP.
8114 035260 012700 000004 MOV #BIT2,R0 ;INDICATE RX.ENABLE BIT TO ' ' SET.
8115 035264 004767 161420 JSR PC,WTWLN ;ENABLE RX ON THE ASSOCIATEL LINES.
8116 ;+
8117 ; SET UP A LOOP WHICH SENDS A CHARACTER ON EACH ACTIVE LINE.
8118 ;-
8119 035270 005003 CLR R3 ;CLEAR THE LINE COUNTER.
8120 035272 010300 6#: MOV R3,R0
8121 035274 006300 ASL R0
8122 035276 036067 002370 144730 BIT BITTBL(R0),ACTLNS
8123 035304 001406 BEQ 8# ;DON'T SEND IF NOT ACTIVE LINE.
8124 035306 010377 144730 MOV R3,@CSRA ;SET UP THE IND.ADR.REG FIELD FOR PROPER LINE.
8125 035312 052700 100000 BIS #BIT15,R0 ;OR IN THE TX.DATA.VALID BIT TO THE TX CHAR.
8126 035316 010077 144722 MOV R0,@TXCHA ;SEND THE CHARACTER.
8127 035322 005203 8#: INC R3 ;SELECT THE NEXT LINE NUMBER.
8128 035324 020327 000010 CMP R3,#NUMLNS ;TEST FOR ALL LINES DONE.
8129 035330 002760 BLT 6# ;LOOP IF NOT ALL LINES DONE.
8130 ;+
8131 ; SET UP LOOP WHICH WAITS FOR ALL CHARS TO BE RECEIVED OR TIME-OUT TO OCCUR.
8132 ;-
8133 035332 016705 144676 MOV ACTLNS,R5
8134 035336 012701 000062 MOV #50.,R1 ;SELECT 50 MS TIME-OUT VALUE.
8135 035342 012702 100000 MOV #BIT15,R2 ;SELECT DATA.VALID BIT TO BE TESTED.
8136 035346 016704 144672 MOV RBUFA,R4 ;SPECIFY THAT BIT IS FOUND IN RBUF REG.
8137 035352 010203 10#: MOV R2,R3 ;SPECIFY TO WAIT FOR BIT TO BE SET.
8138 035354 004767 157012 JSR PC,MSLGET ;WAIT FOR A VALID CHAR TO BE RECEIVED.
8139 035360 103016 BCC 12# ;GO ANALYSE PROBLEM IF TIME-OUT.
8140 035362 010003 MOV R0,R3 ;TEST THE RX CHARACTER TO VERIFY THAT
8141 035364 000303 SWAB R3 ; IT IS A VALID COMBINATION OF LINE
8142 035366 042700 177600 BIC #177600,R0 ; NUMBER AND DATA. IGNORE ANY ERROR
8143 035372 042703 177760 BIC #177760,R3 ; BITS WHICH ARE SET.
8144 035376 006303 ASL R3
8145 035400 026300 004004 CMP TXRXLB(R3),R0
8146 035404 001362 BNE 10# ;RX CHAR A VALID COMBINATION? NO, IGNORE IT.
8147 035406 046005 002370 BIC BITTBL(R0),R5 ;YES, CLEAR THE BIT FOR THE ASSOCIATED LINE.
8148 035412 001357 BNE 10# ;ALL ACTIVE LINES DONE? NO, LOOP.
8149 035414 000410 BR 60# ;YES, SUCCESSFUL COMPLETION. EXIT THE TEST.
8150 ;+
8151 ; NOT ALL ACTIVE LINES RECEIVED CORRECT CHARACTERS.
8152 ; REPORT "TX/RX LINE ERROR ON THE FOLLOWING LOOPED BACK TX LINES:"
8153 ; ERROR NUMBER >>>>> 8602 <<<<<.
8154 ;-
8155 12#: INC ERRNBR ;SELECT PROPER ERROR NUMBER (8602).
8156 035416 005267 146464 MOV #ER6401,ERRBLK ;SELECT PROPER ERROR REPORTING ROUTINE.
8157 035422 012767 011740 146462 MOV #EM8602,R1 ;SELECT PROPER ERROR MESSAGE.
8158 035430 012701 010462 ERROR
8159 035434 104460 TRAP C$ERROR
8160 ;+
8161 035436 005067 144622 60#: CLR CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
8162 035442 #PRI07 SETPRI #PRI07 ;DISABLE ALL INTERRUPTS.
8163 035442 012700 000340 MOV #PRI07,R0
035446 104441 TRAP C$SPRI

```

HARDWARE TEST - TXLNS -

8164 035450  
035450  
035450 104401

ENDTST

L10052: TRAP C8ETST

HARDWARE TEST - TXINT -

```

8166
8167
8168
8169
8170
8171
8172
8173
8174
8175
8176
8177
8178
8179 035452
      035452
8180
8181
8182
8183 035452 032767 000002 144556
8184 035460 001002
8185 035462 000167 000400
8186 035466
      035466 012700 000240
      035472 104441
      000031
8187
8188 035474 012767 000031 144564
8189 035502 012767 177777 144554
8190 035510 012767 000001 146366
8191 035516 012767 020775 146362
8192 035524 012767 010552 146356
8193
8194
8195
8196
8197
8198 035532 004767 155746
8199 035536 103402
8200 035540 000167 000322
8201
8202
8203
8204 035544 004767 155300
8205
8206
8207
8208
8209 035550 012705 000377
8210 035554 005000
8211 035556 004767 161126
8212 035562 012700 156430
8213 035566 004767 161146
8214
8215
8216
8217 035572 016705 144436
8218 035576 004767 160426
8219

```

```

.SBTTL HARDWARE TEST - TXINT -
:*****
:
: TX LINES INTERACTIONS TEST -
:
: THIS TEST VERIFIES THAT EACH TX LINE DOES NOT INTERACT WITH ANY OTHER
: TX LINES OR MODEM CONTROL SIGNALS. IT WILL ONLY BE PERFORMED IF
: EITHER 25 PIN OR STAGGERED LOOPBACK IS SPECIFIED. THIS TEST CAUSES
: A BREAK CONDITION ON EACH ACTIVE LINE INDIVIDUALLY AND LOOKS FOR ANY
: CHARACTERS GENERATED ON OTHER LINES OR ANY CHANGES TO MODEM CONTROL
: SIGNALS ON ACTIVE LINES.
:*****
      BGNTST
      T25::
:
: ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.
:
: BIT #BIT1,LOPBC ;CHECK TYPE OF LOOPBACK MODE SELECTED.
: BNE 2;
: JMP 60; ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
20: SETPRI #PRI05 ;ALLOW LTC INTERRUPTS.
      MOV #PRI05,R0
      TRAP C#SPRI
      TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
      MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (87)
      MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
      MOV #1,ERRTYP ;SET ERROR TYPE IN ERROR TABLE.
      MOV #8701,,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
      MOV #EM8701,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 >>>> 8701 <<<<.
:
: 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.
40:
:
: SET UP TRANSMISSION AN RECEPTION PARAMETERS FOR ALL LINES.
: 9600 BAUD,8 CHAR,1 STOPBIT,NO PARITY.
:
: MOV #MAPLNS,R5 ;INDICATE ALL LINES TO BE SET UP.
: CLR R0 ;SPECIFY ALL LNCTRL BITS TO BE CLEARED.
: JSR PC,WTMLNC ;CLEAR ALL LNCTRL BITS.
: MOV #156430,R0 ;SET UP BAUD RATE,ETC.
: JSR PC,WTMLPR ;SET COMMUNICATION PARAMETERS ON ALL LINES.
:
: ENABLE TRANSMITTERS ON ALL ACTIVE LINES.
:
: MOV ACTLNS,R5 ;PASS ACTIVE LINE BIT MAP.
: JSR PC,TXENBL ;ENABLE TRANSMISSIONS ON ALL LINES.
:

```

HARDWARE TEST

- TXINT -

```

8220 ; ENABLE RECEPTION ON ALL LINES ASSOCIATED WITH ACTIVE LINES.
8221 ;
8222 035602 004767 156034 JSR PC,CONMAP ; CONVERT TO ASSOCIATED LINE MAP.
8223 035606 012700 000004 MOV #BIT2,R0 ; INDICATE RX.ENABLE BIT TO BE SET.
8224 035612 004767 161072 JSR PC,WTWLN ; ENABLE RX ON THE ASSOCIATED LINES.
8225 ;
8226 ; SET UP A LOOP WHICH SETS THE BREAK BIT ON EACH ACTIVE LINE INDIVIDUALLY AND
8227 ; CHECKS FOR INTERACTIONS.
8228 ;
8229 035616 005003 CLR R3 ; CLEAR THE LINE COUNTER.
8230 035620 010300 60: MOV R3,R0
8231 035622 006300 ASL R0
8232 035624 036067 002370 144402 BIT BITTBL(R0),ACTLNS
8233 035632 001503 BEQ 160 ; LINE ACTIVE? NO, SKIP THE LINE.
8234 ;
8235 ; CLEAR THE BREAK BITS FOR ALL LINES.
8236 ; DELAY 10 MS TO ALL LINES TO GET OUT OF BREAK CONDITION.
8237 ;
8238 035634 005000 CLR R0 ; CLEAR LINE COUNTER.
8239 035636 010077 144400 80: MOV R0,BCSRA ; SET UP THE IND.ADR.REG FILED FOR PROPER LINE.
8240 035642 042777 000010 144402 BIC #BIT3,LNCTRA ; CLEAR A BREAK BIT.
8241 035650 005200 INC R0 ; SET LINE COUNTER TO NEXT LINE.
8242 035652 020027 000010 CMP R0,#NULNS
8243 035656 001367 BNE 80 ; DONE? NO, LOOP TO DO NEXT LINE.
8244 035660 012704 000012 MOV #10.,R4 ; SELECT 10 MS DELAY.
8245 035664 004767 156026 JSR PC,DELAY ; DELAY TO ALLOW BREAKS TO BE CLEARED.
8246 ;
8247 ; RECORD THE STATES OF THE MODEM CONTROL SIGNALS.
8248 ; PURGE THE FIFO OF ANY UNWANTED CHARACTERS.
8249 ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>> 8702 THRU 8704 <<<<.
8250 ;
8251 035670 004767 157714 JSR PC,SAVMST ; RECORD THE STATES OF MODEM STATUS SIGNALS.
8252 035674 012767 020776 146204 MOV #8702.,ERRNBR
8253 035702 004767 157206 JSR PC,PUFIFR ; PURGE THE FIFO.
8254 035706 103067 BCC 600 ; PURGE SUCCESSFUL? NO, EXIT TEST.
8255 ;
8256 ; SET THE BREAK ON THE SELECTED LINE.
8257 ;
8258 035710 010377 144326 MOV R3,BCSRA ; SET UP THE IND.ADR.REG FIELD FOR PROPER LINE.
8259 035714 052777 000010 144330 BIS #BIT3,LNCTRA ; SET THE BREAK BIT IN LNCTRL REGISTER.
8260 035722 012704 000012 MOV #10.,R4 ; SELECT 10 MS DELAY.
8261 035726 004767 155764 JSR PC,DELAY ; DELAY TO ALLOW BREAK TO BE SET.
8262 ;
8263 ; VERIFY THAT ONLY THE ASSOCIATED LINE RECEIVES CHARACTERS.
8264 ;
8265 035732 012700 000012 MOV #10.,R0 ; ALLOW UP TO 10 "LEGAL" CHARACTERS.
8266 035736 017701 144302 100: MOV BRBUFA,R1 ; READ A CHARACTER FROM THE DUT.
8267 035742 100022 BPL 140 ; DATA VALID? NO, CHECK MODEM SIGNALS.
8268 035744 000301 SWAB R1
8269 035746 042701 177760 BIC #177760,R1 ; GET LINE NUMBER OF RX CHAR.
8270 035752 126103 004044 CMPB TXRLNB(R1),R3 ; COMPARE AGAINST TX ASSOCIATED LINE NUMBER.
8271 035756 001403 BEQ 120 ; IS CHARACTER ON ASSOCIATED LINE? NO, ERROR.
8272 035760 005300 DEC R0 ; COUNT THIS CHARACTER.
8273 035762 001365 BNE 100 ; TOO MANY RXED? NO, LOOP TO LOOK FOR MORE.
8274 035764 000433 BR 500 ; YES, REPORT ERROR AND ABORT.
8275 ;
8276 ; CHARACTER RECEIVED ON WRONG LINE.

```



HARDWARE TEST - TXINT -

```

8277      ; REPORT "DATA LINE INTERACTIONS ON LINE NN DECIMAL."
8278      ; ERROR NUMBER          >>>> 8705 <<<<<.
8279      ;-
8280 035766 012767 021001 146112 12:  MOV   #8705.,ERRNBR
8281 035774 012701 010612          MOV   #EM8702,R1      ;SELECT THE ERROR MESSAGE.
8282 036000 012767 012036 146104      MOV   #ER7801,ERRBLK ;SELECT THE ERROR REPORT ROUTINE.
8283 036006          ERROR
8284          104460          TRAP   C#ERROR
8285      ;+
8286      ; CHECK FOR MODEM SIGNAL INTERACTIONS.
8287 036010 005002          14:  CLR   R2          ;SELECT NO BITS TO BE IGNORED.
8288 036012 004767 155532          JSR   PC,CMPMST      ;COMPARE OLD AND NEW MODEM STATUS.
8289 036016 103411          BCS   16:          ;STATUS CHANGES? NO, SKIP ERROR REPORT.
8290      ;+
8291      ; MODEM STATUS CHANGES HAVE BEEN DISCOVERED.
8292      ; REPORT          >>>> 8706 <<<<<.
8293      ; "INTERACTIONS FOUND BETWEEN TX FOR LINE NN AND THE FOLLOWING SIGNALS:"
8294      ;-
8295 036020 012767 021002 146060      MOV   #8706.,ERRNBR
8296 036026 012767 012064 146056      MOV   #ER8401,ERRBLK
8297 036034 012701 010641          MOV   #EM8703,R1
8298 036040          ERROR          ;ERR8401 USES R1, R2, R3 VALUES.
8299          104460          TRAP   C#ERROR
8300      ;+
8301      ; SELECT NEXT LINE AND TEST IT IF WE ARE NOT DONE.
8302 036042 005203          16:  INC   R3
8303 036044 020327 000010          CMP   R3,#NUMLNS      ;TEST FOR ALL LINES DONE.
8304 036050 002663          BLT   6:          ;ALL DONE? NO, LOOP TO DO NEXT LINE.
8305 036052 000405          BR    60:          ;YES, EXIT THE TEST.
8306      ;+
8307      ; FOUND AN ERROR NOT DIRECTLY RELATED TO THE TEST.
8308      ; REPORT ERROR          >>>> 8707 <<<<<.
8309      ; ABORT THE TEST.
8310      ;-
8311 036054 012767 021003 146024 50:  MOV   #8707.,ERRNBR
8312 036062 004767 157712          JSR   PC,TSABRT
8313      ;+
8314 036066 005067 144172          60:  CLR   CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
8315 036072          SETPRI #PRI07      ;DISABLE ALL INTERRUPTS.
8315          036072 012700 000340          MOV   #PRI07,R0
8315          036076 104441          TRAP   C#SPRI
8316      ;+
8317 036100          ENDTST
8317          036100          L10053:
8317          036100 104401          TRAP   C#ETST

```

HARDWARE TEST - REP8MP -

8319  
8320  
8321  
8322  
8323  
8324  
8325  
8326  
8327  
8328  
8329 036102  
036102  
8330 000032  
8331 036102 012767 000032 144156  
8332 036110 012767 177777 144146  
8333 036116 016702 144320  
8334 036122 012703 002444  
8335 036126 020203  
8336 036130 001411  
8337  
8338  
8339  
8340  
8341  
8342 036132 012701 011242  
8343 036136 104455  
036140 022125  
036142 011125  
036144 012616  
8344  
8345 036146 012767 002444 144266  
8346  
8347 036154 005067 144104  
8348 036160  
036160  
036160 104401

```

.SBTTL HARDWARE TEST - REP8MP -
; ** *****
;* - REPORT ANY BMP CODES IN THE QUEUE -
;* THIS IS A PSEUDO-TEST USED TO REPORT ANY BMP CODES THAT WERE FOUND
;* IN THE DUT'S FIFO DURING PREVIOUS TEST, AND LOGGED IN THE BMP CODE
;* QUEUE.
;* IT IS UNLIKELY THAT RUNNING THIS PSEUDO-TEST ALONE WILL PRODUCE ANY
;* ERROR REPORTS.
;-- *****
BGNTST
                                T26::
TNUM == TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
MOV #TNUM,TSTNUM ;SET UP THE TEST NUMBER. (93)
MOV #-1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
MOV #BMPQBP,R2 ;GET THE CONTENTS OF THE POINTER.
MOV #BMPQBP,R3 ;GET THE START ADDRESS OF THE QUEUE.
CMP R2,R3 ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
BEQ 60$ ;EXIT NO CODES IN THE QUEUE.

; *
; THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
; -
;REPORT ERROR BMP CODE FOUND IN TEST NN, BMP CODE:NNNNNN"
MOV #EM9304,R1 ;PASS THE FIRST MESSAGE TO BE REORTED.
ERRDF 9301,EM9301,ER9301 ; >>>> ERROR #9301 <<<<<.
                                TRAP C#ERDF
                                .WORD 9301
                                .WORD EM9301
                                .WORD ER9301

MOV #BMPQBP,BMPQBP ;SET POINTER BACK TO THE BEGINING OF THE QUE.
60$: CLR CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
ENDTST
                                L10054:
                                TRAP C#ETST

```

HARDWARE TEST - REPBMP -

```

8351 ;*****
8352 ;
8353 ;           FVTB.HWQ
8354 ;
8355 ;*****
8357
8358
8359 .SBTTL  HARDWARE PARAMETER CODING SECTION
8360
8361
8362
8363 ;**
8364 ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
8365 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
8366 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8367 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
8368 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
8369 ; WITH THE OPERATOR.
8370 ;--
8371
8372 036162          BGNHRD
      036162 000027
      036164
                                     .WORD L10055-L$HARD/2
8373                                     L$HARD::
8383
8384 036164          ;DEVICE CSR ADDRESS QUESTION:
      036164 000031          GPRMA  HWPTQ1,0,0,160000,177776,YES
      036166 036242          .WORD  T$CODE
      036170 160000          .WORD  HWPTQ1
      036172 177776          .WORD  T$LLOLIM
                                     .WORD  T$HILIM
8385
8386 036174          ;DEVICE INTERRUPT VECTOR QUESTION:
      036174 001031          GPRMA  HWPTQ2,2,0,40,776,YES
      036176 036260          .WORD  T$CODE
      036200 000040          .WORD  HWPTQ2
      036202 000776          .WORD  T$LLOLIM
                                     .WORD  T$HILIM
8387
8388 036204          ;ACTIVE LINES BIT MAP QUESTION:
      036204 002032          GPRMD  HWPTQ3,4,0,MAPLNS,0,177777,YES
      036206 036313          .WORD  T$CODE
      036210 000377          .WORD  HWPTQ3
      036212 000000          .WORD  MAPLNS
      036214 177777          .WORD  T$LLOLIM
                                     .WORD  T$HILIM
8389
8390 036216          ;TYPE OF LOOPBACK QUESTION:
      036216 003032          GPRMD  HWPTQ4,6,0,377,1,3,YES
      036220 036341          .WORD  T$CODE
      036222 000377          .WORD  HWPTQ4
      036224 000001          .WORD  377
      036226 000003          .WORD  T$LLOLIM
                                     .WORD  T$HILIM
8391
8392 036230          ;INTERRUPT BR LEVEL QUESTION:
      036230 003032          GPRMD  HWPTQ5,6,0,177400,0,6,YES
      036232 036417          .WORD  T$CODE
      036234 177400          .WORD  HWPTQ5
      036236 000000          .WORD  177400
      036240 000006          .WORD  T$LLOLIM
                                     .WORD  T$HILIM

```

HARDWARE PARAMETER CODING SECTION

8393  
8394  
8395 036242  
  
036242  
8396  
8403  
8404 036242 103 123 122  
036245 040 101 104  
036250 104 122 105  
036253 123 123 072  
036256 040 000  
8405 036260 111 116 124  
036263 105 122 122  
036266 125 120 124  
036271 040 126 105  
036274 103 124 117  
036277 122 040 101  
036302 104 104 122  
036305 105 123 123  
036310 072 040 000  
8406 036313 101 103 124  
036316 111 126 105  
036321 040 114 111  
036324 116 105 040  
036327 102 111 124  
036332 040 115 101  
036335 120 072 040  
036340 000  
8407 036341 124 131 120  
036344 105 040 117  
036347 106 040 114  
036352 117 117 120  
036355 102 101 103  
036360 113 040 050  
036363 061 075 111  
036366 116 124 105  
036371 122 116 101  
036374 114 054 062  
036377 075 110 063  
036402 062 067 067  
036405 054 063 075  
036410 110 063 062  
036413 065 051 072  
036416 000  
8408 036417 111 116 124  
036422 105 122 122  
036425 125 120 124  
036430 040 102 122  
036433 040 114 105  
036436 126 105 114  
036441 072 040 000

ENDHRD

L10055: .EVEN

HWPTQ1: .ASCIZ /CSR ADDRESS: /

HWPTQ2: .ASCIZ /INTERRUPT VECTOR ADDRESS: /

HWPTQ3: .ASCIZ /ACTIVE LINE BIT MAP: /

HWPTQ4: .ASCIZ /TYPE OF LOOPBACK (1=INTERNAL,2=H3277,3=H325):/

HWPTQ5: .ASCIZ /INTERRUPT BR LEVEL: /

.EVEN

HARDWARE PARAMETER CODING SECTION

```

8413 ;*****
% 8414 ;
8415 ; FVTA.SWQ
8416 ;
8417 ;*****
%

```

.SBTTL SOFTWARE PARAMETER CODING SECTION

```

8419 ;**
8420 ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
8421 ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE
8422 ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8423 ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE
8424 ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
8425 ; WITH THE OPERATOR.
8426 ;--
8427
8428
8429
8430
8431
8432

```

```

8432 036444 BGNSFT
      036444 000010
      036446
                                           .WORD L10056-L$SOFT/2
                                           L$SOFT::

```

```

8433
8442 ;UNIT NUMBER PRINTOUT QUESTION:
8443 036446 GPRML SWPTQ1,0,20,YES
      036446 000130
      036450 036466
      036452 000020
                                           .WORD T$CODE
                                           .WORD SWPTQ1
                                           .WORD 20

```

```

8444 ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
8445 036454 GPRMD SWPTQ2,2,D,177777,0,177777,YES
      036454 001052
      036456 036542
      036460 177777
      036462 000000
      036464 177777
                                           .WORD T$CODE
                                           .WORD SWPTQ2
                                           .WORD 177777
                                           .WORD T$LOLIM
                                           .WORD T$HILIM

```

```

8446
8447 .EVEN
8448
8449 036466 ENDSFT
                                           .EVEN
                                           L10056:

```

```

8450
8451
8458 036466 122 105 120 SWPTQ1: .ASCIZ /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /
      036471 117 122 124
      036474 040 125 116
      036477 111 124 040
      036502 116 125 115
      036505 102 105 122
      036510 040 101 123
      036513 040 105 101
      036516 103 110 040
      036521 125 116 111
      036524 124 040 111
      036527 123 040 124
      036532 105 123 124
      036535 105 104 072
      036540 040 000

```

SOFTWARE PARAMETER CODING SECTION

8459	036542	116	125	115
	036545	102	105	122
	036550	040	117	106
	036553	040	111	116
	036556	104	111	126
	036561	111	104	125
	036564	101	114	040
	036567	104	101	124
	036572	101	040	105
	036575	122	122	117
	036600	122	123	040
	036603	124	117	040
	036606	122	105	120
	036611	117	122	124
	036614	040	117	116
	036617	040	101	040
	036622	114	111	116
	036625	105	072	040
	036630	000		

SWPTQ2: .ASCIZ /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /

8460

.EVEN

SOFTWARE PARAMETER CODING SECTION

```

8462
8463 ;*****
8464 ;
8465 ;           FVTSKL6.P11
8466 ;
8467 ;*****
8468
8469
8470
8471 036632 $PATCH:
8472 036632 .BLKW 24
8473
8480
8481
8482
8483
8484 036702 LASTAD
                                .EVEN
                                .WORD 0
                                .WORD 0
      036702 000000
      036704 000000
8485 036706 L$LAST:
8486          ENDMOD
8487
8488
8489
8490
8491
8492
8493          000001          .END

```





SYMBOL TABLE

ISR = 000100 G	L\$EXP5 002066 G	L10035 025746	PASCNT 002276 G	TP4FLG 002312 G
IXE = 004000 G	L\$HARD 036164 G	L10036 026420	PCSL0T= 000016 G	TP4RTN 017040 G
I\$AU = 000041	L\$HIME 002120 G	L10037 027044	PNT = 001000 G	TP4VEC 002310 G
I\$AUTO= 000041	L\$HPCP 002016 G	L10040 027552	PREGRT 004136 G	TSABRT 016000 G
I\$CLN = 000041	L\$HPTP 002022 G	L10041 030506	PREG05 004114	TSTNUM 002266 G
I\$DU = 000041	L\$HMW 002212 G	L10042 031222	PRI = 002000 G	TXAD1A 002254 G
I\$HRD = 000041	L\$ICP 002104 G	L10043 031736	PRI00 = 000000 G	TXAD10= 000012 G
I\$INIT= 000041	L\$INIT 017076 G	L10044 032366	PRI01 = 000040 G	TXAD2A 002256 G
I\$MOD = 000041	L\$LADP 002026 G	L10045 033016	PRI02 = 000100 G	TXAD20= 000014 G
I\$MSG = 000041	L\$LAST 036706 G	L10046 033446	PRI03 = 000140 G	TXBFCA 002260 G
I\$PROT= 000040	L\$LOAD 002100 G	L10047 034076	PRI04 = 000200 G	TXBFCA= 000016 G
I\$PTAB= 000041	L\$LUN 002074 G	L10050 034506	PRI05 = 000240 G	TXCHA 002244 G
I\$PWR = 000041	L\$MREV 002050 G	L10051 035116	PRI06 = 000300 G	TXCHRO= 000002 G
I\$RPT = 000041	L\$NAME 002000 G	L10052 035450	PRI07 = 000340 G	TXDATP 016112 G
I\$SEG = 000041	L\$PRIO 002042 G	L10053 036100	PRTLPR 014750 G	TXDSBL 016134 G
I\$SETU= 000041	L\$PROT 017070 G	L10054 036160	PUFIFO 015032 G	TXENBL 016230 G
I\$SFT = 000041	L\$PRT 002112 G	L10055 036242	PUFIFR 015114 G	TXIE0 016324 G
I\$SRV = 000041	L\$REPP 002062 G	L10056 036466	RBUFA 002244 G	TXINTC 002304 G
I\$SUB = 000041	L\$REV 002010 G	MAPCNT 014340 G	RBUFO = 000002 G	TXINTF 002306 G
I\$TST = 000041	L\$RPT 017062 G	MAPLNS= 000377 G	READBX 015306 G	TXRLNB 004044 G
J\$JMP = 000167	L\$SOFT 036446 G	MFUNIT 004214 G	RESETT 015370 G	TXRLNE 004064 G
LGRP1M 002272 G	L\$SPC 002056 G	MENAB 002346 G	RXB0TX= 000030 G	TXRXLB 004004 G
LGRP2M 002274 G	L\$SPCP 002020 G	MMPRES 002344 G	RXBETX= 000020 G	TXRXLE 004044 G
LINBIT 014312 G	L\$SPTP 002024 G	MMSRO 002342 G	RXBFUL= 000100 G	TXVECA 002232 G
LNCTRA 002252 G	L\$STA 002030 G	MSG1 011422 G	RXCNTB 003744 G	T\$ARGC= 000002
LNCTRO= 000010 G	L\$SW 002224 G	MSG2 011500 G	RXIE0 015502 G	T\$CODE= 001052
LOE = 040000 G	L\$TEST 002114 G	MSG3 011557 G	RXINTC 002300 G	T\$ERRN= 022125
LOPBCK 002236 G	L\$TIML 002014 G	MSLCNT 002340 G	RXINTF 002302 G	T\$EXCP= 000000
LOT = 000010 G	L\$UNIT 002012 G	MSLGET 014372 G	RXVECA 002230 G	T\$FLAG= 000050
LPCSLT= 000036 G	L10000 002222	MSLOOP 014506 G	ROSLOT= 000002 G	T\$GMAN= 000000
LPRA 002246 G	L10001 002230	MSTICK 002336 G	R1SLOT= 000004 G	T\$HILI= 177777
LPRO = 000004 G	L10002 011420	NDERPT 002226 G	R2SLOT= 000006 G	T\$LAST= 000001
L\$ACP 002110 G	L10003 011656	NEWPAS 017460	R3SLOT= 000010 G	T\$LOLI= 000000
L\$APT 002036 G	L10004 011736	NEWRES 017452	R4SLOT= 000012 G	T\$LSYM= 010000
L\$AU 020064 G	L10005 012034	NEWSTA 017142	R5SLOT= 000014 G	T\$LTNO= 000032
L\$AUT 002070 G	L10006 012062	NUMLNS= 000010 G	SAVBMP 015542 G	T\$NEST= 177777
L\$AUTO 017734 G	L10007 012330	OOPS 014522 G	SAVMST 015610 G	T\$NSO = 000000
L\$CCP 002106 G	L10010 012420	OPTION 002224 G	SETPAR 015654 G	T\$NS1 = 000005
L\$CLEA 017736 G	L10011 012566	O\$APTS= 000000	SFPTBL 002224 G	T\$PTNU= 000000
L\$CO 002032 G	L10012 012614	O\$AU = 000000	SKPSTS 015722 G	T\$SAVL= 177777
L\$DEPO 002011 G	L10013 012774	O\$BGNR= 000001	STATA 002250 G	T\$SEGL= 177777
L\$DESC 004164 G	L10014 017066	O\$BGNS= 000001	STATO = 000006 G	T\$SUBN= 000000
L\$DESP 002076 G	L10016 017732	O\$DU = 000001	STGTRB 004064 G	T\$TAGL= 177777
L\$DEVP 002060 G	L10017 017734	O\$ERRT= 000001	STSTB 002644 G	T\$TAGN= 010057
L\$DISP 002124 G	L10020 017752	O\$GNSW= 000001	STSTE 002704 G	T\$TEMP= 000000
L\$DLY 002116 G	L10021 020062	O\$POIN= 000001	SVCGBL= 000000	T\$TEST= 000032
L\$DTP 002040 G	L10022 020070	O\$SETU= 000000	SVCINS= 000001	T\$TSTM= 177777
L\$DTYP 002034 G	L10023 020360	PARATB 002350 G	SVCSUB= 000001	T\$TSTS= 000001
L\$DU 017754 G	L10024 020754	PARATE 002370 G	SVCTAG= 000001	T\$\$AU = 010022
L\$DUT 002072 G	L10025 021402	PAR0A 002350 G	SVCTST= 000001	T\$\$AUT= 010017
L\$DVTY 004154 G	L10026 022150	PAR1A 002352 G	SWPTQ1 036466	T\$\$CLE= 010020
L\$EF 002052 G	L10027 022716	PAR2A 002354 G	SWPTQ2 036542	T\$\$DU = 010021
L\$ENVI 002044 G	L10030 023320	PAR3A 002356 G	S\$LSYM= 010000	T\$\$HAR= 010055
L\$ERRT 004104 G	L10031 023742	PAR4A 002360 G	TIMER1 002326 G	T\$\$HW = 010000
L\$ETP 002102 G	L10032 024240	PAR5A 002362 G	TIMER2 002330 G	T\$\$INI= 010016
L\$EXP1 002046 G	L10033 024550	PAR6A 002364 G	TIMER3 002332 G	T\$\$MSG= 010013
L\$EXP4 002064 G	L10034 025250	PAR7A 002366 G	TNUM = 000032 G	T\$\$PRO= 010015

## SYMBOL TABLE

T\$\$RPT= 010014	T14 027046 G	T22 034100 G	T7 023322 G	WORD1 002314 G
T\$\$SOF= 010056	T15 027554 G	T23 034510 G	T8 023744 G	WTWLNC 016710 G
T\$\$SW = 010001	T16 030510 G	T24 035120 G	T9 024242 G	WTWLPR 016740 G
T\$\$TES= 010054	T17 031224 G	T25 035452 G	UAM = 000200 G	X\$ALWA= 000000
T1 020072 G	T18 031740 G	T26 036102 G	UNITN 002240 G	X\$FALS= 000040
T10 024552 G	T19 032370 G	T3 020756 G	UNSDIV 016364 G	X\$OFFS= 000400
T11 025252 G	T2 020362 G	T4 021404 G	WAIBIC 016520 G	X\$TRUE= 000020
T12 025750 G	T20 033020 G	T5 022152 G	WAIBIS 016574 G	\$PATCH 036632 G
T13 026422 G	T21 033450 G	T6 022720 G	WAITTX 016650 G	

. ABS. 036706 000  
000000 001

ERRORS DETECTED: 0

VIRTUAL MEMORY USED: 28661 WORDS ( 112 PAGES)

DYNAMIC MEMORY: 20060 WORDS ( 77 PAGES)

ELAPSED TIME: 00:04:53

**CVDHBC.BIN,CVDHBC.LST/-SP=SVC34R/ML,CVDHBC.P11**