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

ORIGINAL RELEASE: 15-DEC 1983  ANTHONY HART

VERSION 00 3-MAR-1984  ANTHONY HART

THE FOLLOWING MODIFICATIONS HAVE BEEN MADE TO THE OLD CZDHWA:

THE HARDWARE QUESTION "TYPE OF LOOPBACK" HAS BEEN ALTERED TO INCLUDE THE
STAGGERED LOOPBACK CONNECTORS ON THE DMUII DISTRIBUTION PANEL (M3029).

THE HARDWARE QUESTION "interrupt vector" HAS BEEN REMOVED.

THE HARDWARE QUESTION "BR LEVEL" HAS BEEN REMOVED.

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

THE MODEM SIGNAL TESTS FROM PART CZDHVA HAVE BEEN TRANSFERRED INTO THIS PART.
THEY WERE TESTS 16 THRU 23 IN CZDHVA.
# Table of Contents

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

1.1 PROGRAM ABSTRACT

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

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

1.2 SYSTEM REQUIREMENTS

THE FOLLOWING HARDWARE IS REQUIRED TO RUN THE DHU11 FVT:

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

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

2.0 OPERATING INSTRUCTIONS

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

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.

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

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 LABELED EXTENDED COMMAND SYNTAX.
2.2 SWITCHES

There are several switches which are used to modify supervisor operation. These switches are appended to the legal commands. All of the legal switches are tabulated below with a brief description of each. In the descriptions below, a decimal number is designated by "DDDDD".

<table>
<thead>
<tr>
<th>SWITCH</th>
<th>EFFECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>/TESTS:LIST</td>
<td>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.</td>
</tr>
<tr>
<td>/PASS:DDDDD</td>
<td>Execute DDDD passes (DDDDD = 1 to 64000)</td>
</tr>
<tr>
<td>/FLAGS:FLGS</td>
<td>Set specified flags. See the flags section of this document.</td>
</tr>
<tr>
<td>/EOP:DDDDD</td>
<td>Report end of pass message after every DDDD passes only. (DDDDD = 1 to 64000)</td>
</tr>
<tr>
<td>/UNITS:LIST</td>
<td>Test/ADD/DROP only those units specified in the list. List example: /UNITS:0,5,10,11,12 use units 0,5,10,11,12 (UNIT NUMBERS = 0 63)</td>
</tr>
</tbody>
</table>

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 "/TESTS:1-5" instead of "/TESTS:1-5".

Below is a table that specifies which switches can be used by each command.

<table>
<thead>
<tr>
<th>SWITCH</th>
<th>PASS</th>
<th>FLAGS</th>
<th>EOP</th>
<th>UNITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>RESTART</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CONTINUE</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PROCEED</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DROP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ADD</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRINT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DISPLAY</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>FLAGS</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ZFLAGS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EXIT</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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.

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

*SEE THE ERROR INFORMATION SECTION OF THIS DOCUMENT.

SEE THE XXDPG 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

********************************************************************************
STARTRT/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
(ALLE 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 OCCURRENCE OF AN ERROR WITH THE MALT ON ERROR FLAG
BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END
OF "EFFECT OF START COMMAND" SECTION.

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

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

- MHE: MALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED.
- LEO: 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.
- IRE: INHIBIT ERROR REPORTS.
- IXE: INHIBIT EXTENDED ERROR REPORTS.
- PRI: DIRECT ALL MESSAGES TO A LINE PRINTER.
- PNI: PRINT NUMBER OF TEST BEING EXECUTED.
- BOE: BELL ON ERROR (NOT RELATED TO BELL PROMPTING).
- UAM: RUN IN UNATTENDED MODE, BYPASSING MANUAL INTERVENTION (ILLEGAL FOR THIS DIAGNOSTIC).
- ISA: INHIBIT STATISTICAL REPORTS.
IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC.
HAS NO EFFECT IN THIS DIAGNOSTIC.
LOT LOOP ON TEST.
THE FLAGS NAMED OR EQUIVALED TO 1 ARE SET, THOSE EQUIVALED 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 "O
UNITS (D)?" TO WHICH THE OPERATOR SHOULD REPLY WITH THE NUMBER OF
UNITS TO BE TESTED. FOLLOWING THIS ARE THE QUESTIONS WHEREBY THE
P-TABLES THEMSELVES ARE BUILT. EACH P-TABLE IS A CORE-RESIDENT TABLE
CONTAINING ALL THE HARDWARE INFORMATION FOR ONE COMPLETE UNIT. EACH
QUESTION IS FOLLOWED BY THE RESPONSE RADIX (D FOR DECIMAL, B FOR
BINARY, O FOR OCTAL, L FOR YES/NO) IN PARENTHESES AND THE DEFAULT
VALUE AFTER THE PARENTHESES. FOR THE ACTUAL HARDWARE P-TABLE
QUESTIONS SEE THE "HARDWARE PARAMETERS" SECTION.

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

EXAMPLE:
STA/TESTS:1:3-4:/PASS:3/FLAGS:IER:MOE:1

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

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

2.4.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

2.4.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)  <UNIT-LIST> IS A SEQUENCE
OF DECIMAL NUMBERS (0.1 ETC.) OR RANGES OF DECIMAL NUMBERS (0.5, 8 10
ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED
BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF
UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES
THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE
HARDWARE DIALOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN
DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP
COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN
DROPPED BY A DROP COMMAND.

2.4.2.3 EFFECT OF RESTART COMMAND -

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

2.4.3 CONTINUE COMMAND -

***********************************************************************
CONTINUE)/PASS:<PASS-CNT>/FLAGS:<FLAG-LIST>
***********************************************************************

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

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

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

2.4.4 PROCEED COMMAND

***************************************************************************
PROCEED/FLAGS:<FLAG-LIST>
***************************************************************************

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

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

2.4.4.2 EFFECT OF PROCEED COMMAND

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

2.4.5 ADD COMMAND

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

2.4.6 EFFECT OF ADD COMMAND -

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

2.4.7 DROP COMMAND

***************************************************************************
DROP(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 -
***********************************************************************
PR(IN(N))
***********************************************************************

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

2.4.10 DISPLAY COMMAND -
******************************************************************************
DISPLAY/UNITS:<UNIT-LIST>
******************************************************************************

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

2.4.11 FLAGS COMMAND
******************************************************************************
FLAGS
******************************************************************************

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 TELETYPewriter OUTPUT TO BE SURPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER CONTROL/O IS TYPED, WHICH RESTORES NORMAL TELETYPewriter 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 XDOP+ USER'S MANUAL). WHEN YOU ANSWER THIS
QUESTION WITH A "Y", THE RUNTIME SERVICES WILL ASK FOR THE NUMBER
OF UNITS (IN DECIMAL). YOU WILL THEN BE ASKED THE FOLLOWING
QUESTIONS FOR EACH UNIT.

1. CSR ADDRESS - THIS QUESTION REQUESTS THE CSR ADDRESS OF THE
   SPECIFIED DMU-11. THE DEFAULT ANSWER FOR THIS QUESTION IS
   ADDRESS 160460 (OCTAL).

2. ACTIVE LINES BIT MAP - THIS QUESTION REQUESTS AN OCTAL BIT
   MAP** OF THE SERIAL COMMUNICATION LINES ON THE DMU11 WHICH ARE
   BEING SELECTED FOR TESTING. IF THE BIT IN THE BIT MAP IS SET
   WHICH CORRESPONDS TO A PARTICULAR LINE (I.E., BIT 5 FOR LINE
   5) THAT LINE WILL BE TESTED BY THE FVT. THE DEFAULT ANSWER
   FOR THIS QUESTION IS ALL LINES I.E., 177777.

3. TYPE OF LOOPBACK (1=INTERNAL, 2=M3029 OR M3277, 3=M325)
   THIS QUESTION REQUESTS THE TYPE OF LOOPBACK TO BE USED WHEN
   TESTING THE DMU-11.

THE FOLLOWING TYPES ARE SUPPORTED:

  0 INTERNAL - ONLY INTERNAL UART LOOPBACK IS TO BE USED IN
    TESTING THE DMU-11. SINCE ALL THE TESTS IN THIS PART
    REQUIRE EITHER STAGGERED OR SINGLE LINE LOOPBACK, SPECIFYING
    INTERNAL LOOPBACK WILL CAUSE THE TESTS TO BE SKIPPED. THIS
    WILL NOT HOWEVER CAUSE ANY ERRORS TO BE REPORTED.

  0 M3029 OR M3277 - STAGGERED LOOPBACK CONNECTORS ARE PROVIDED
    ON THE DMU11 DISTRIBUTION PANEL (M3029) IF THIS DISTRIBUTION
    PANEL IS NOT PRESENT THEN M3277 STAGGERED BERG CONNECTOR(S)
    MUST BE INSTALLED ON THE BERG CONNECTOR SOCKETS OF THE DMU11.

  0 M325 - SINGLE LINE, 25 PIN LOOPBACK CONNECTORS (TYPE M325)
    ARE INSTALLED ON THE LINES TO BE TESTED.
2.6 SOFTWARE QUESTIONS

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

1. REPORT UNIT NUMBER AS EACH UNIT IS TESTED - THIS QUESTION ASKS WHETHER THE PROGRAM SHOULD REPORT THE NUMBER OF THE UNIT WHICH IT IS TESTING AS IT BEGINS TO TEST THAT UNIT. 

2. EXTENDED ERROR REPORTING - THIS QUESTION ASKS WHETHER EXTENDED ERROR INFORMATION IS REQUIRED OTHER THAN THE "TEST FAILED" MESSAGE, ON EACH ERROR REPORTED. THE DEFAULT IS "NO" I.E. ONLY A MESSAGE REPORTING THE FACT THAT THE TEST FAILED WILL BE PRINTED. 

3. NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE - THIS QUESTION IS ASKED ONLY IF THE PREVIOUS QUESTION WAS ANSWERED "YES". THE QUESTION ASKS FOR THE NUMBER OF DATA ERRORS WHICH SHOULD BE REPORTED INDIVIDUALLY BY THIS PROGRAM FOR EACH LINE FOR EACH TRANSMISSION TEST. ERRORS WHICH ARE NOT REPORTED INDIVIDUALLY ARE REPORTED IN SUMMARY ERROR REPORTS.
2.7 EXTENDED P TABLE DIALOGUE

WHEN YOU ANSWER THE HARDWARE QUESTIONS, YOU ARE BUILDING ENTRIES IN A TABLE THAT DESCRIBES THE DEVICES UNDER TEST. THE SIMPLEST WAY TO BUILD THIS TABLE IS TO ANSWER ALL QUESTIONS FOR EACH UNIT TO BE TESTED. IF YOU HAVE A MULTIPILEXED 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 REPETITIVE.

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 C FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

0 UNITS (D) ? 8<CR>

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

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

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

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

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

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

UNIT 7
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE @ (O) ? 6<CR>
Q-FACTOR (O) 0 ? 1<CR>
UNIT 8
CSR ADDRESS (O) 160000<CR>
SUB-DEVICE # (O) ? 7<CR>
Q-FACTOR (O) ? <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.

0 UNITS (O) ? 8<CR>

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

UNIT 3
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 2.5<CR>
Q-FACTOR (O) 0 ? 0<CR>

UNIT 7
CSR ADDRESS (O) ? 160000<CR>
SUB-DEVICE # (O) ? 6.7<CR>
Q-FACTOR (O) 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.

0 UNITS (O) ? 8<CR>
UNIT 1
CSR ADDRESS (O) ? 16000<CR>
SUB-DEVICE # (O) ? 0 7<CR>
Q-FACTOR (O) 0 ? 0,1,0,...,1,1<CR>

As you can see from this example, null replies (commag enclosing a null field) tell the runtime services to repeat the last reply.

2.8 Quick Start-Up Procedure (XXDP*)

To start up this program:

1. Boot XXDP*

2. Give the date and answer the LSI/UNIBUS and 50Hz (if there is a clock) questions. Note, not all versions of XXDP* ask for the clock frequency.

3. Type "R NAME", where NAME is the name of the BIN or BIC file for this program.

4. Type "START"

5. Answer the "CHANGE HW" question with "Y"

6. Answer all the hardware questions

7. Answer the "CHANGE SW" question with "N"

When you follow this procedure you will be using only the defaults for flags and software parameters. For default information see the sections within this document on flags, and hardware questions.
3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

There are three levels of error messages that may be issued by:
A diagnostic, general, basic and extended. General error messages
are always printed unless the "IER" flag is set (see the flags section
of this document).

The general error message is of the form:

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

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

Basic error messages are messages that contain some additional
information about the error. These are always printed unless
the "IER" or "IBR" flags are set (see the flags section of this
document). These messages are printed after the associated general message.

Extended error messages contain supplementary error information
such as register contents or good/bad data. These are always
printed unless the "IER", "IBR" or "IXR" flags are set (see the
flags section of this document).

These messages are printed after the associated general error
message and any associated basic error messages.
3.2 SPECIFIC ERROR MESSAGES

This program is intended to provide a go/no-go indication of the functionality of the DHU-11 boards. To execute this program in this mode the operator need only answer the "extended error reporting" software question with "no". The program will then only print the name of the failing test, the test and error numbers. For a list of the test names in this program see the test summaries section of this document. An example of such a an error message is the following:

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

This error indicates that a fatal error was encountered during the test which tests the data terminal ready control signal.

If the operator had requested extended error reporting the same error would be reported as follows:

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

4.0 PERFORMANCE AND PROGRESS REPORTS

At the end of each pass, the pass count is given along with the total number of errors reported since the diagnostic was started. The "EOF" switch can be used to control how often the end of pass message is printed. For further information see the switches section of this document.
5.0 TEST SUMMARIES

THE FOLLOWING ARE INCLUDED WITHIN CZDHWB:

1. DEVICE REGISTER ACCESS TEST - VERIFIES THAT THE UUT REGISTERS
   WILL RESPOND WITH THE CORRECT UNIBUS HANDSHAKING SIGNALS.
   VERIFIES THAT THE UUT IS AT THE CORRECT ADDRESS.

2. DTR TEST - VERIFIES THAT CHANGING THE STATE OF THE DTR BIT
   AFFECTS THE STATE OF THE DTR CONTROL LINE.

3. RTS TEST - VERIFIES THAT CHANGING THE STATE OF THE RTS BIT
   AFFECTS THE STATE OF THE RTS CONTROL LINE.

4. DSR TEST - VERIFIES THAT THE DSR STATUS SIGNAL CORRECTLY
   REPORTS THE STATE OF THE LOOPED BACK DTR CONTROL LINE.

5. RI TEST - VERIFIES THAT THE RI STATUS SIGNAL CORRECTLY
   REPORTS THE STATE OF THE LOOPED BACK DTR CONTROL LINE.

6. CTS TEST - VERIFIES THAT THE CTS STATUS SIGNAL CORRECTLY
   REPORTS THE STATE OF THE LOOPED BACK RTS CONTROL LINE.

7. DCD TEST - VERIFIES THAT THE DCD STATUS SIGNAL CORRECTLY
   REPORTS THE STATE OF THE LOOPED BACK RTS CONTROL LINE.

8. DTR INTERACTIONS TEST - VERIFIES THAT CHANGING THE STATE
   OF THE DTR CONTROL SIGNAL ON ANY LINE DOES NOT AFFECT THE
   STATE OF ANY STATUS SIGNALS THAT IT IS NOT LOOPED BACK TO.

9. RTS INTERACTIONS TEST - VERIFIES THAT CHANGING THE STATE
   OF THE RTS CONTROL SIGNAL ON ANY LINE DOES NOT AFFECT THE
   STATE OF ANY STATUS SIGNALS THAT IT IS NOT LOOPED BACK TO.

10. REPORT BMP CODES TEST - THIS PSEUDO TEST REPORTS THE FIRST 32
    CHARACTERS WHICH WERE DISCOVERED IN THE FIFO DURING THE
    EXECUTION OF THE OTHER TESTS. THIS-AVOIDS INTERRUPTION OF
    THE OTHER TESTS BY THESE CODES IF THEY ARE NOT CRITICAL TO
    THE PERFORMANCE OF THE TESTS.
6.0 EXAMPLE ERROR FREE PASS

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

.R CZDMWBO
CZDMWBO.BIN
DRS
CZDMW.B-O
DMU-11 FUNC TST PART3
UNIT IS DMU-11
RESTART ADDR: 147670
DR>STA/PAS:1

CHANGE HW (L) ? Y

@ UNITS (D) ? 2

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

UNIT 1
CSR ADDRESS: (0) 160460 ? 160500
ACTIVE LINE BIT MAP: (0) 17777 ? <CR>
TYPE OF LOOPBACK (1*INTERNAL, 2*H3029 OR H3277, 3*H325): (0) 2 ?

CHANGE SW (L) ? Y

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

TESTING UNIT : 0
TESTING UNIT : 1
CZDMW EOP 1
0 TOTAL ERRS
DR>
.LIST SEQ,LOC,BIN,MEB
.NLIST CND

.SBTTL PROGRAM HEADER

.MCALL SVC

; INITIALIZE SUPERVISOR MACROS

; IF STRUCTURED MACROS ARE TO BE USED, ADD "MCALL STRUCT" AND "STRUCT"
; TO INITIALIZE THE STRUCTURED MACROS.

SVCINS* 1 ; LIST INSTRUCTIONS, SHIFTED RIGHT
SVCST* 1 ; LIST TEST TAGS, SHIFTED RIGHT
SVCSTB* 1 ; LIST SUBTEST TAGS, SHIFTED RIGHT
SVCGBL* 1 ; LIST GLOBAL TAGS, SHIFTED RIGHT
SVCGBL* 1 ; LIST OTHER TAGS, SHIFTED RIGHT

; CHANGE THE VALUES OF THE SVC...SYMBOLS TO BE ZERO IF YOU WISH
; TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS. CHANGE THE
; SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS. YOU MAY
; CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.

.ENABLE ABS
.ENABLE AAM
   * 2000
      BGNMOD

; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
; *
;
POINT BGNRPT,BGNST,BGNSFT,BGNDU,ERRTBL

HEADER C20H.B,0,10.0,PRI07

LNAME:: .ASCII /C/
          .ASCII /Z/
          .ASCII /X/
          .ASCII /W/
          BYTE 0
          BYTE 0
          BYTE 0

LSREV::  .ASCII /B/
LDEPO::  .ASCII /O/
LSNIT::  .ASCII /O/
LSTIML:: .WORD 0
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
0

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

1085

PROGRAM HEADER
.DISPATCH TABLE

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

DISPATCH 10

.L4DISPATCH::

   .WORD  10

   .WORD  T1
   .WORD  T2
   .WORD  T3
   .WORD  T4
   .WORD  T5
   .WORD  T6
   .WORD  T7
   .WORD  T8
   .WORD  T9
   .WORD  T10

002122  000002
002124  012526
002126  013010
002130  013544
002132  014300
002134  014750
002136  015420
002140  016070
002142  016540
002144  017170
002146  017620

1105
.SBIL 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.
.
BGNMW DFPTBL

.LIMW: .WORD L10000 L$IMW/2
DFPTBL::

.LIMW: .WORD 160460 ;DEFAULT CSR ADDRESS
DFPTBL::

.LIMW: .WORD 177777 ;DEFAULT ACTIVE LINES BIT MAP
DFPTBL::

.BYTE 2 ;DEFAULT LOOPBACK TYPE

.ENDMW

.L10000:
**SBTL 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.**

```
BDNSW SFPTBL

002160 000002
002162
002162

002162 000020
002164 000000

002166
002166

OPTION:: .WORD 20  ; BIT MAP OF PROGRAM CONTROL FLAGS
NDERPT:: .WORD 0   ; DEFAULT NUMBER OF INDIVIDUAL DATA ERRORS TO RPT.

ENDSW

.LISW::

SFP1BL::

.WORD L10001::L15SW/2

L10001:
```
SBTIL GLOBAL EQUATES SECTION

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

NUMLN5=20 ; NUMBER OF LINES ON DHU11 IS 8.
MAPLN5=177777 ; BIT MAP OF LINES ON DHU11.

****** DEVICE REGISTER OFFSETS FROM THE CSR'S ADDRESS ******
CSRO=0 ; CSR REGISTER OFFSET FROM THE CSR ADDRESS
RBUP0=2 ; RECEIVE REGISTER OFFSET FROM THE CSR ADDRESS
RXTIMO=2 ; RECEIVE TIMER REGISTER OFFSET FROM THE CSR ADDRESS
LPRO=4 ; LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
FSLSD=6 ; FIFO SIZE/STATUS REGISTER OFFSET FROM THE CSR ADDRESS
FDATO=6 ; FIFO DATA REGISTER OFFSET FROM THE CSR ADDRESS
LNCTRO=10 ; LINE CONTROL REGISTER OFFSET FROM THE CSR ADDRESS
TXAD10=12 ; TRANSMIT ADDRESS 1 REGISTER OFFSET FROM THE CSR ADDRESS
TXAD20=14 ; TRANSMIT ADDRESS 2 REGISTER OFFSET FROM THE CSR ADDRESS
TXBFCO=16 ; TRANSMIT COUNT REGISTER OFFSET FROM THE CSR ADDRESS

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

EQUALS

; BIT DEFINITIONS

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

001000 BIT9=BIT09
000400 BIT8=BIT08
000200 BIT7=BIT07
000100 BIT6=BIT06
00000040   BI5** BIT05
00000020   BI4** BIT04
00000010   BI3** BIT03
00000004   BI2** BIT02
00000002   BI1** BIT01
00000001   BI0** BIT00

; EVENT FLAG DEFINITIONS
; EF32:EF17 RESERVED FOR SUPERVISOP TO PROGRAM COMMUNICATION
0000640   EF.START**  32. ; START COMMAND WAS ISSUED
0000637   EF.RESTART** 31. ; RESTART COMMAND WAS ISSUED
0000636   EF.CONTINUE** 30. ; CONTINUE COMMAND WAS ISSUED
0000635   EF.NEW** 29. ; A NEW PASS HAS BEEN STARTED
0000634   EF.PWR**  28. ; A POWER-FAIL/POWER UP OCCURRED

; PRIORITY LEVEL DEFINITIONS
0000340   PRI07** 340
0000300   PRI06** 300
0000240   PRI05** 240
0000200   PRI04** 200
0000140   PRI03** 140
0000100   PRI02** 100
0000040   PRI01**  40
0000000   PRI00**  0

; OPERATOR FLAG BITS
0000004   EVL**  4
0000010   LOF** 10
0000020   ADR**  20
0000040   IDU**  40
0000100   ISR** 100
0000200   UAM** 200
0000400   BOE** 400
001000   PNT** 1000
002000   PRI** 2000
004000   IEX** 4000
010000   IBE** 10000
020000   IER** 20000
040000   LOE** 40000
100000   MDE** 100000
.SBTL GLOBAL DATA SECTION

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

;******************************************************************************
;******************************************************************************
;******************************************************************************
;******************************************************************************
;******************************************************************************
;******************************************************************************
;******************************************************************************

UNIT VARIABLE AREA

******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************

RXVECA: .WORD 200 ;RX VECTOR ADDRESS.
TXIQA: .WORD 204 ;TX VECTOR ADDRESS.
ACTLNS: .WORD 177777 ;ACTIVE LINE BIT MAP.
LOPBCK: .BYTE 0 ;LOOPBACK MODE.
BALEV: .BYTE 4 ;INTERCEPT BUS REQUEST LEVEL.
UNITN: .WORD 0 ;UNIT NUMBER.

******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************

DEVICE REGISTER ADDRESS TABLE

******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************

DRDAT: ;
CSR#: .WORD 160020 ;DMU-1: CSR ADDRESS.
RXTMA: .WORD 160022 ;DMU-1 RECEIVE BUFFER/TIMER ADDRESS.
LPR#: .WORD 160024 ;DMU-1 LINE PARAMETER REGISTER ADDRESS.
FDATA: .WORD 160026 ;DMU-1 FIFO SIZE/LINE STATUS REGISTER ADDRESS.
;AND FIFO DATA REGISTER ADDRESS.
LCNTRA: .WORD 160030 ;DMU-1 LINE CONTROL REGISTER ADDRESS.
TXADIA: .WORD 160032 ;DMU-1 TRANSMIT BUFFER 1 REGISTER ADDRESS.
TXDA#: .WORD 160034 ;DMU-1 TRANSMIT BUFFER 2 REGISTER ADDRESS.
TXBPCA: .WORD 160036 ;DMU-1 TRANSMIT BUFFER COUNT REGISTER ADDRESS.

******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************

ASSORTED GLOBAL VARIABLES:

******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************

BUFPTR: .WORD 0 ;STORAGE FOR RECEIVE CHARACTER BUFFER POINTER.
CTRL#: .WORD 0 ;STORAGE FOR THE CONTROL-C FLAG.
EXE#: .WORD 0 ;"EXIT ON ERROR" FLAG.
TEST#: .WORD 0 ;STORAGE FOR TEST NUMBER.
IEST#: .WORD 0 ;STORAGE FOR STATES OF THE DUT INT ENABLE BITS.
LGPR#: .WORD 31463 ;BIT MAP OF LINES IN LINE GROUP 1.
LGPR#: .WORD 146314 ;BIT MAP OF LINES IN LINE GROUP 2.
PSCT#: .WORD 0 ;STORAGE FOR PASS COUNT USED IN ROM VERSION TST.
RXINT#: .WORD 0 ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
RXINF#: .WORD 0 ;STORAGE FOR RECEIVER INTERRUPT FLAGS.
TXINF#: .WORD 0 ;STORAGE FOR TRANSMIT INTERRUPT COUNT.
TPA#: .WORD 0 ;STORAGE FOR THE NORM/L Q04 TRAP VECTOR.
TPA#: .WORD 0 ;FLAGS SET WHEN AN EXPECTED Q04 TRAP OCCURS.
LOC#: .WORD 0 ;LOCATION FOR PASSING INDIRECT PARAMETERS.

******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************

LINE CLOCK VARIABLES AND STORAGE.

******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************
******************************************************************************

CLKCSR: .WORD 177546 ;CSR ADDRESS OF THE LTC.
CLKBRL: .WORD PRI06 ;INTERCEPT PRIORITY LEVEL OF THE LTC.
G3

GLOBAL DATA SECTION

1294 002262 000100
1295 002264 000074
1296 002266 000000
1297 002270 000000
1298 002272 000170
1299 002274 000170
1300 002278 000217
1301 002300 000062
1302
1303
1304
1305
1306 002302 177572
1307 002304 000000
1308 002306 000000
1309
1310
1311
1312
1313 002310 000001
1314 002312 000002
1315 002314 000004
1316 002316 000010
1317 002320 000020
1318 002322 000040
1319 002324 000100
1320 002326 000200
1321 002330 000400
1322 002332 001000
1323 002334 002000
1324 002336 004000
1325 002340 010000
1326 002342 020000
1327 002344 040000
1328 002346 100000
1329
1330
1331
1332
1333 002350 000000
1334 002352 000000
1335 002354 000000
1336
1337
1338
1339
1340 002552 000000
1341 002554 000000
1342
1343
1344
1345
1346 002610 000000
1347 002612 000000
1348 002614 000000
1349 002616 000000
1350 002618 000000

BITBL:: .WORD 1 ;BIT 0 SET.
... ;BIT 1 SET.
... ;BIT 2 SET.
... ;BIT 3 SET.
... ;BIT 4 SET.
... ;BIT 5 SET.
... ;BIT 6 SET.
... ;BIT 7 SET.
... ;BIT 8 SET.
... ;BIT 9 SET.
... ;BIT 10 SET.
... ;BIT 11 SET.
... ;BIT 12 SET.
... ;BIT 13 SET.
... ;BIT 14 SET.
... ;BIT 15 SET.

; STORAGE AREA FOR THE BMP CODE QUEUE.

BMPREP:: .WORD 0 ;ERROR SUMMARY TABLE AND FLAGS.
BERSMFR:: .WORD 0 ;BASE OF OUT STAT STORAGE TABL.
GLOBAL DATA SECTION

1351 002620 000000
1352 002620 000000
1353 002620 000000
1354 002620 000000
1355 002620 000000
1356 002620 000000
1357 002620 000000
1358 002620 000000
1359 002620 000000
1360 002620 000000
1361 002620 000000
1362 002620 000000
1363 002620 000000
1364
1365
1366
1367
1368 002620
1369 002620
1370 002620
1371 002620
1372 002620
1373 002620
1374
1375
1376
1377
1378
1379
1380
1381 002710
1382 002710 000000
1383 002712 000000
1384 002714 000000
1385 002716 000000
1386 002720 000010
1387 002722 000012
1388 002724 000014
1389 002726 000016
1390 002730 000020
1391 002732 000022
1392 002734 000024
1393 002736 000026
1394 002740 000030
1395 002742 000032
1396 002744 000034
1397 002746 000036
1398 002750
1399
1400
1401
1402
1403
1404
1405
1406 002750
1407 002750 000

**TABLE FOR STORAGE OF R/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.

**TABLE FOR STORAGE OF R/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.
GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.

**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 SUBRNE WHICH CALLED PREG05.
```

* Each level of subroutine 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 exception, 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.

************************************************************************************
1506
1507 .SBTTL GPR FRAME ACCESS EQUATES
1508 
1509 ;***
1510 ;EQUATES THAT ALLOW ACCESS TO THE STACK FRAME. THESE ARE THE
1511 ;OFFSETS INTO THE STACK FOR REGISTERS SAVED DURING THE PREVIOUS
1512 ;ROUTINE.
1513 
1514 000036 LPCSLT** 36 ;OFFSET FOR LAST RETURN PC.
1515 000016 PCSLT** 16 ;OFFSET FOR RETURN PC.
1516 000014 R5SLT** 14 ;OFFSET FOR R5.
1517 000012 R4SLT** 12 ;OFFSET FOR R4.
1518 000010 R3SLT** 10 ;OFFSET FOR R3.
1519 000006 R2SLT**  6 ;OFFSET FOR R2.
1520 000004 R1SLT**  4 ;OFFSET FOR R1.
1520 000002 ROSLOT**  2 ;OFFSET FOR R0.
.SBTIL GLOBAL MACRO DEFINITION

;******************************************************************************
;** THIS MACRO IS USED AT THE BEGINNING OF A SUBROUTINE TO SAVE THE
;** CONTENTS OF THE GPRS RO THRU R5.
;**
;** INPUTS: SP UNCHANGED SINCE SUBROUTINE WAS ENTERED.
;**         R5SLOT - OFFSET TO STACK SLOT FOR R5 (EQUIVATED 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

.JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.

.ENDM SAVE
GLOBAL MACRO DEFINITION

**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 GP RS BACK TO
  THE CALLING ROUTINE BY ALTERING THE GPR SAVE AREA ON THE STACK AND THEN
  RETURNING TO PREGOS TO RESTORE THE GP RS TO THEIR SAVED VALUES.

* INPUTS:    ONLY ALLOWED ARGUMENTS ARE "R0" THRU "R5".
  ROSLOT THRU RSSLOT MUST BE EQ UATED 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 GP RS LISTED AS ARGUMENTS WILL BE PAS S INTACT TO THE
  CALLING ROUTINE. ALL OTHER GP RS 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 RO IN STACK SLOT.
  MOV    R3,RSSLOT(SP) ;PUT R3 IN STACK SLOT.
  JSR    PC,R(SP) ;RETURN TO PREGOS SUBR.

  IN TH I S EXAMPLE GP RS R1, R2, R4, AND R5 WILL BE RESTORED TO
  THEIR VALUES CONTAINED IN THE STACK FRAME AND RO AND R3
  WILL BE LEFT AT THEIR VALUES PRIOR TO THIS PASS CALL.

* SUBORDINATE ROUTINES CALLED: (PREGRT - LABEL WITHIN PREGOS, VALUE ON STACK.)

**************************************************
.SBTL GLOBAL SUBROUTINE
/*
* PREERVE REGISTERS RO THROUGH R5 FOR SUBROUTINE CALLS.
*/
* INPUTS: THE RETURN ADDRESS BACK INTO THE CALLING ROUTINE MUST BE IN
* GPR R5, (I.E. MACROS USE "JSR R5, PREG05").
* OUTPUTS: REGISTERS RO THROUGH R5 ARE SAVED ON THE STACK.
* CALLING SEQUENCE: [SUBROUTINE CODE]
* [SUBROUTINE CODE] ; MACRO EXPANSION CALLS PREG05.
* PASS ; MACRO EXPANSION RECALLS PREG05.
* COMMENTS: THIS ROUTINE IS RE ENTRANT.
* PARAMETERS MAY BE PASSED OUT OF A SUBROUTINE BY MODIFYING THE
* REGISTER SAVE AREA ON THE STACK. USE THE PASS GPRN MACRO
* TO RETURN GPR VALUES INTACT.
* USE THE RN5LOT OFFSETS FROM THE SP TO PASS OTHER PARAMETERS.
* [EXAMPLE: MOV VALUE,RN5LOT(SP)]
* MAKE SURE THE SP IS AT ITS ORIGINAL VALUE WHEN YOU DO THIS.
* SUBORDINATE ROUTINES CALLED: NONE.
*
*----------------------------------------------------------------------
*
PREG05:

R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL

 MOV R4,-(SP) ;SAVE R4
 MOV R3,-(SP) ;SAVE R3
 MOV R2,-(SP) ;SAVE R2
 MOV R1,-(SP) ;SAVE R1
 MOV R0,-(SP) ;SAVE R0
 MOV R5,-(SP) ;PUSH RETURN PC ON TOP OF STACK
 MOV RN5LOT(SP),R5 ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS

 JSR PC,@(SP)+ ;CALL THE SUBROUTINE AT THE RETURN ADDRESS
 ;FROM THE PREG05 CALL, PUTTING THE PRESENT
 ;PC ON THE STACK AS A RETURN ADDRESS INTO
 ;THIS (PREG05) ROUTINE.

 ;THE FOLLOWING CODE IS EXECUTED WHEN THE CALLING ROUTINE DOES A
 ;"RETURN" [JSR PC,@(SP)+] USING THE PC DEPOSITED ON THE STACK ABOVE.

 PREGRT::

 MOV (SP)+,R5 ;PUT RETURN PC IN R5.
 MOV (SP)+,R0 ;RESTORE R0.
 MOV (SP)+,R1 ;RESTORE R1.
 MOV (SP)+,R2 ;RESTORE R2.
 MOV (SP)+,R3 ;RESTORE R3.
 MOV (SP)+,R4 ;RESTORE R4.

 RTS R5 ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05.
 ;RESTORING R5 IN THE PROCESS.
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

DEVTYPE "DMU 11"

ASCIZ "DMU 11"

TEST DESCRIPTION

DESCRIPT "DMU 11 FUNC TST PARTS"

ASCIZ "DMU-11 FUNC TST PAR"
.LIST BIN

; ********** FORMAT STATEMENTS USED IN PRINT CALLS **********

EF0503:: .ASCIZ /#uA#N/
EF1601:: .ASCIZ /#uA#MA, TEST ABORTED #N/
EF7801:: .ASCIZ /#uA#MA ON LINE #D2#MA DECIMAL #N/
EF8401:: .ASCIZ /#uA#MA FOR LINE #D2#MA(D) AFFECTS OTHER MODEM SIGNALS #N/
EF8402:: .ASCIZ /#uA# MA CHANGING #uA#MA FOR LINE #D2#MA(D) AFFECTED /
EF8403:: .ASCIZ /#uA#MA FOR LINE #D2#MA(D), #N/
EF9301:: .ASCIZ /#uA#MA #D2#MA(D), BMP CODE REPORTED ;#D3#MA#D#MA #N/
EF9302:: .ASCIZ /#uA#MA OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE) #N/
MFUNIT:: .ASCIZ /#uA#MA TESTING UNIT ;#D4#MA #N/
.EVEN

.LIST BIN
.LIST BIN

;********** GLOBAL ERROR MESSAGES **********

EM0103:.ASCIIZ /DEVICE REGISTER ACCESS ERRORS/
EM1601:.ASCIIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
EM7801:.ASCIIZ /MODEM CONTROL DTR BIT TEST FAILED/
EM5015:.ASCIIZ /DTR BIT FAULTY/
EM7901:.ASCIIZ /MODEM CONTROL RTS BIT TEST FAILED/
EM5077:.ASCIIZ /RTS BIT FAULTY/
EM5117:.ASCIIZ /DSR MODEM STATUS SIGNAL TEST FAILED/
EM5163:.ASCIIZ /DSR MODEM STATUS SIGNAL DEFECTIVE/
EM8002:.ASCIIZ /RI MODM STATUS SIGNAL TEST FAILED/
EM5227:.ASCIIZ /RI MODM STATUS SIGNAL DEFECTIVE/
EM8102:.ASCIIZ /RI MODM STATUS SIGNAL DEFECTIVE/
EM5335:.ASCIIZ /CTS MODEM STATUS SIGNAL TEST FAILED/
EM8202:.ASCIIZ /CTS MODEM STATUS SIGNAL DEFECTIVE/
EM5401:.ASCIIZ /DCD MODEM STATUS SIGNAL TEST FAILED/
EM5445:.ASCIIZ /DCD MODEM STATUS SIGNAL DEFECTIVE/
EM8302:.ASCIIZ /DCD MODEM STATUS SIGNAL DEFECTIVE/
EM5551:.ASCIIZ /DTR MODEM CONTROL SIGNAL INTERACTIONS TEST FAILED/
EM5637:.ASCIIZ /DTR/
EM8403:.ASCIIZ /DSR/
EM5647:.ASCIIZ /RI/
EM8404:.ASCIIZ /DCD/
EM5652:.ASCIIZ /DCD/
EM8405:.ASCIIZ /CTS/
EM5662:.ASCIIZ /RTS MODEM CONTROL SIGNAL INTERACTIONS TEST FAILED/
EM5744:.ASCIIZ /RTS/
EM8502:.ASCIIZ /RTS/
EM5750:.ASCIIZ /BMP CODES WERE REPORTED DURING THIS DIAGNOSTIC/
EM9301:.ASCIIZ /BMP CODE FOUND IN TEST /
EM9302:.ASCIIZ /THE LAST BMP CODE WAS FOUND IN TEST /
EM9303:.ASCIIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
EM6124:.ASCIIZ /EVEN

.EVEN

.LIST BIN
.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.

***
**GLOBAL ERROR REPORTING ROUTINE**

- **SBLTL** GLOBAL ERROR REPORTING ROUTINE
- **ER0101**

- **INPUTS:**
  - **R5** - ERROR FLAG WORD.
  - **IF BIT 0 IS SET, A READ ERROR OCCURRED.**
  - **IF BIT 1 IS SET, A WRITE ERROR OCCURRED.**

- **OUTPUTS:**
  - **MESSAGES ARE PRINTED AT THE OPERATOR CONSOLE.**

- **CALLING SEQUENCE:**
  - INCLUDE THE LABEL "ER0101" AS THE MESSAGE POINTER PARAMETER IN THE DRS ERROR REPORT MACRO CALL.

- **COMMENTS:**
  - **SUBORDINATE ROUTINES USED: NONE.**

---

**BGNMSG ER0101**

SAVE

JSR $R5,PREG05 ; CALL REGISTER SAVE SUBRT.

MOV #BIT06,RO ; SET-UP THE BIT MAP FOR REPORT EXT'D ERROR INFO.

BIC #REGISTER,RO ; TRY AND CLEAR THE FLAG.

BNE #6$ ; EXIT IF OPTION NOT SELECTED.

; REPORT EXTENDED ERROR INFORMATION

BIT #BIT0,R5 ; TEST FOR READ ERROR.

BEQ 2$ ; SKIP READ ERROR MSG IF NO READ ERROR.

PRINTB @MSG1 ; PRINT READ ERROR MESSAGE.

MOV @MSG1,-(SP)

MOV @1, (SP)

MOV SP,RO

TRAP @IPNTB

ADD #4,SP

2$: BIT #BIT1,R5 ; TEST FOR WRITE ERROR.

BEQ 4$ ; SKIP WRITE ERROR MSG IF NO WRITE ERROR.

PRINTB @MSG2 ; PRINT WRITE ERROR MESSAGE.

MOV @MSG2,-(SP)

MOV @1, (SP)

MOV SP,RO

TRAP @IPNTB

ADD #4,SP

4$: PRINTX @MSG3 ; SUGGEST THAT DMU MAY BE AT WRONG ADDRESS.

MOV @MSG3,-(SP)

MOV @1, (SP)

MOV SP,RO

TRAP @IPNTB

ADD #4,SP
6F:  PASS                JSR PC,B((SP))
     ENDMG
     RETURN TO PREVIOUS SUBRT.

L10002: TRAP C(MSG

MAINMSG   104423

006316 045 101 102 MSG1:   .ASCIZ /*ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.*/
006321 125 123 040
006324 124 111 115
006327 105 055 117
006332 125 124 040
006335 124 122 101
006340 120 040 103
006343 101 125 123
006346 105 104 040
006351 102 131 040
006354 122 105 101
006357 104 040 101
006362 124 124 105
006365 115 120 124
006370 056 045 116
006373 000

006374 045 101 102 MSG2:   .ASCIZ /*ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.*/
006377 125 123 040
006402 124 111 115
006405 105 055 117
006410 125 124 040
006413 124 122 101
006416 120 040 103
006421 101 125 123
006424 105 104 040
006427 102 131 040
006432 127 122 111
006435 124 105 040
006440 101 124 124
006443 105 115 120
006446 124 056 045
006451 116 000

006453 045 101 104 MSG3:   .ASCIZ /*ADMLU MAY BE AT THE WRONG UNIBUS ADDRESS.*/
006456 110 125 040
006461 115 101 131
006464 040 102 105
006467 040 101 124
006472 040 124 110
006475 105 040 127
006500 122 117 116
006503 107 040 125
006506 116 111 102
006511 125 123 040
006514 101 104 104
006517 122 105 123
006522 123 056 045
006525 116 045 116
0L 530 000

1821 .EVEN
**GLOBAL ERROR REPORTING ROUTINE**

**SBTIL**  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, PROVIDED
* EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
* INPUTS:  R1  ADDRESS OF THE MESSAGE TO PRINT.
* OUTPUTS:  A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
* INCLUDE THE LABEL "ER0503" AS THE MESSAGE POINTER.
* PARAMETER IN THE DIAE SUPER ERROR REPORT MACRO CALL.
* COMMENTS:  THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
* SUBORDINATE ROUTINES USED:  NONE.

**BGNMSG**  ER0503

ER0503::

MOV  #BIT06,R0 ;TRY TO CLEAR THE
BIC  DPTRN,R0 ;EXT'D ERROR REPORTING FLAG
BNE  21 ;EXIT IF FLAG NOT SET.

PRINTB  #EF0507; R1 ;PRINT THE MESSAGE.

21:

ENDMSG

\10003:

TRAP  C\#MSG
GLOBAL ERROR REPORTING ROUTINE - ER1603

**DESCRIPTION:**
This error reporting routine is used to print out a basic error message, along with a message informing the operator which test is about to be aborted, provided extended error information has been requested. Otherwise only a "test failure" message will be printed.

**INPUTS:**
- R1 contains the address of the message to be printed.
- ERMSG - contains the address of the message that indicates the test that is being performed, e.g. 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 DM2 error report macro call.

**COMMENTS:**

**SUBORDINATE Routines Called:** None.

**BEGIN**
BGMSG ER1603

```assembly
SAVE RS, PREGO5 ; Call register save subrt.
JSR R5, PREGO5 ; Call register save subrt.

MOV .BIT06, R0 ; Try to clear the

BIC OPTION, R0 ; Exit if error reporting flag

BNE 2* ; Exit if flag not set.

PRINTB #EF0503, R1 ; Print basic message on operators console.

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

PRINTB #EF1601, R2 ; Print "test aborted" message.

MOV ERRMSG, R2

PRINTB #EF1601, R2

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)

ENDMSG

2*: ; Return to PREGO5 subrt.

L10094: TRAP C#MSG
```
GLOBAL ERROR REPORTING ROUTINE

THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS AN ADDITIONAL ERROR MESSAGE WHOSE ADDRESS IS PASSED AS AN INPUT PARAMETER. A LINE NUMBER IS INCLUDED AT THE END OF THE MESSAGE. THE MESSAGE IS PRINTED ONLY IF EXTENDED ERROR REPORTING IS REQUESTED.

INPUTS:  R1 ADDRESS OF THE MESSAGE TO PRINT.
         R3 NUMBER OF LINE ON WHICH ERROR OCCURRED.

OUTPUTS: A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.

                    LOAD THE LINE NUMBER INTO R3.
                    INCLUDE THE LABEL "ER7801" AS THE MESSAGE POINTER
                    IN THE DIAG SUPER ERROR REPORT MACRO CALL.

COMMENTS:  THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.

SUBORDINATE ROUTINES USED: NONE.

BGMSG  ER7801

ER7801:

1:

; EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

2:

BIT  #BIT06,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
BEQ  2F ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
            ;DURING THE SOFTWARE QUESTIONS.

PRINTB  #EF7801,R1,R3 ;PRINT THE MESSAGE.

MOV  R3,-(SP)
MOV  R1,-(SP)
MOV  #EF7801,-(SP)
MOV  #3,-(SP)
MOV  SP,R0
TRAP  CIPNTB
ADD  @10,SP

ENDMSG

L10005:

TRAP  CMSG
GLOBAL ERROR REPORTING ROUTINE

THIS ERROR REPORTING SUBROUTINE IS INTENDED TO REPORT INTERACTIONS
WHICH HAVE BEEN FOUND BETWEEN A MODEM SIGNAL AND OTHER MODEM SIGNALS.
IT ANALYZES THE MODEN STATUS WHICH IS STORED IN THE STAT STORAGE AREA
AND REPORTS ANY DISCREPANCIES WHICH ARE FOUND BETWEEN THIS STORED DATA
AND THE PRESENT STATE OF THE STAT REGISTERS. SPECIFIED BITS ON THE
LINE ASSOCIATED WITH THE SPECIFIED LINE ARE IGNORED.

INPUTS:
R1 - ADDRESS OF SIGNAL NAME MESSAGE.
R2 - BIT MAP OF BITS TO IGNORE ON SPECIFIED LINE.
R3 - NUMBER OF SPECIFIED LINE.
CSR - CONTAINS THE ADDRESS OF THE OUT CSR.
NUMLS - EQUATED TO THE NUMBER OF LINES ON THE OUT.
FSLSA - CONTAINS THE ADDRESS OF THE OUT STAT REGISTER.
STSTB - LABEL AT BASE OF STAT STORAGE TABLE.
TAR NB - LABEL AT BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.

OUTPUTS:
A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.

CALLING SEQUENCE:
INCLUDE THE LABEL "ER8401" 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 ER8401
SAVE JSR R5,PREG05 ;CALL REGISTER SAVE SUBRT.

EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

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

PRINTB @EF8401,R1,R3 ;PRINT THE BASIC MESSAGE.
MOV R3,-(SP)
MOV R1,-(SP)
MOV @EF8401,-(SP)
MOV R3,-(SP)
MOV SP,RO
TRAP C1PNTB
ADD @10,SP

MOV R1,44$ ;SAVE THE ADDRESS OF THE SIGNAL NAME MESSAGE.
CLR R1 ;CLEAR THE LINE COUNTER.
MOV @STSTB,R4 ;SET UP STAT STORAGE POINTER TO BASE OF TABLE.
MOV R1,BGSA ;SET UP THE CSR INDX.ADR.REG FIELD.
MOV @FSLSA,RO ;GET THE CONTENTS OF THIS LINE'S STAT REGISTER.
MOV (R4),R5 ;GET THE PREVIOUS CONTENTS FROM STORAGE.
GLOBAL ERROR REPORTING ROUTINE

1975 007010 050005
1976 007012 012700 043777
1977 007016 120163 003750
1978 007022 001002
1979 007024 056600 000006
1980 007026 004006 45: BIS R0,R5
1981 007032 023705 100000
1982 007036 001404
1983 007040 012702 005643
1984 007044 004767 000064
1985 007050 032705 020000
1986 007054 001404
1987 007056 012702 005647
1988 007062 004767 000046
1989 007066 032705 010000
1990 007072 001404
1991 007074 012702 005652
1992 007100 004767 000030
1993 007104 032705 004000
1994 007110 001404
1995 007112 012702 005656
1996 007116 004767 000012
1997
1998 007122 005201
1999 007124 020127 000020
2000 007130 002720
2001 007132 000417
2002
2003
2004
2005 007134
2006 007166 000207
2007 007170 000000
2008 007172 004736
2009 007174
2010

; LOCAL ERROR MESSAGE LINE PRINTING ROUTINE.

1975 BIS R0,R5
1976 MOV @43777,R0
1977 CMPP R1, TXRNBB(R5)
1978 BNE 46:
1979 BIS R2SLOT(SP), R0
1980 GET BIT MAP OF UNDESIRED CHANGES.
1981 BIT @BIT15,R5
1982 BEQ 66:
1983 MOV @EMB403,R2
1984 JSR PC,404:
1985 BIT @BIT13,R5
1986 BEQ 88:
1987 MOV @EMB404,R2
1988 JSR PC,401:
1989 BIT @BIT12,R5
1990 BEQ 101:
1991 MOV @EMB405,R2
1992 JSR PC,404:
1993 BIT @BIT11,R5
1994 BEQ 121:
1995 MOV @EMB406,R2
1996 JSR PC,408:

; SELECT NEXT LINE.
1997 INC R1
1998 CMP R1, @NUMLNS
1999 ALL LINES DONE?
2000 RET 60:
2001 ;EXIT THIS ROUTINE.
2002
2003

; LOCAL STORAGE FOR ADDRESS OF SIGNAL NAME.
2004

; PREPARE TO MASK OUT UNUSED BITS.
2005 BIS R0,R5
2006 MOV @43777,R0
2007 CMPP R1, TXRNBB(R5)
2008 BNE 46:
2009 DON'T MASK OUT SPECIFIED BITS IF IT IS NOT.
2010 MASK OUT SPECIFIED BITS.
2011 GET BIT MAP OF UNDESIRED CHANGES.
2012 BIT @BIT15,R5
2013 CHECK FOR DSR SIGNAL INTERACTION.
2014 SKIP PRINTING LINE IF NO DSR INTERACTION.
2015 SELECT DSR ERROR MESSAGE.
2016 PRINT THE LINE OF THE ERROR MESSAGE.
2017 CHECK FOR RI SIGNAL INTERACTION.
2018 SKIP PRINTING LINE IF NO RI INTERACTION.
2019 SELECT RI ERROR MESSAGE.
2020 PRINT THE LINE OF THE ERROR MESSAGE.
2021 CHECK FOR DCD SIGNAL INTERACTION.
2022 SKIP PRINTING LINE IF NO DCD INTERACTION.
2023 SELECT DCD ERROR MESSAGE.
2024 PRINT THE LINE OF THE ERROR MESSAGE.
2025 CHECK FOR RTS SIGNAL INTERACTION.
2026 SKIP PRINTING LINE IF NO RTS INTERACTION.
2027 SELECT RTS ERROR MESSAGE.
2028 PRINT THE LINE OF THE ERROR MESSAGE.
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.
** PROVIDED EXTENDED ERROR REPORTING HAS BEEN ENABLED.
**
** 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.

```
BGMSG ER9301
    SAVE JSR R5, PREG05 ; CALL REGISTER SAVE SUBRT.
    MOV @BIT06,R0 ; TRY TO CLEAR THE
    BTIC OPTION,R0 ; EXT'D ERROR REPORTING FLAG
    BNE 60$ ; EXIT IF FLAG NOT SET.
    PRINTB @EF0503,R1 ; REPORT UNEXPECTED BMP CODES FOUND.
    MOV R1,-(SP)
    MOV @EF0503,-(SP)
    MOV R2,-(SP)
    MOV SP,RO
    TRAP CIPNTB
    ADD 06, SP

    MOV @BMPQ08,R3 ; GET THE START ADDRESS OF THE BMP CODE QUEUE.
    MOV @EM9302,R5 ; GET THE MESSAGE TO BE REPORTED.
    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,50$ ; GO REPORT THE BMP CODE.
    CMP R3, R2 ; CHECK IF ALL CODES HAVE BEEN REPORTED.
    BLD 2$ ; 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 60$ ; EXIT IF NOT AT THE LAST LOCATION.
    TST 2(R2) ; CHECK FOR A BMP CODE IN THE LAST CELL
    BEQ 60$ ; EXIT IF NO OVERFLOW HAS OCCURRED, 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.
```
2060 007306 007306 012746 004501
007312 012746 000001
007316 010600
007320 104415
007322 062706 000004
2061 007326 004767 000002
2062 007332 000414
2063
2064 007334 010446
007336 010146
007340 010546
007342 012746 004423
007346 012746 000004
007352 010600
007354 104415
007356 062706 000012
2065 007362 000207
2066 007364
007364 004736
2067 007366
2068 007366
007366
007366 104423

PRINTX @EF9302  ;REPORT OVERFLOW CONDITION.
       MOV @EF9302, (SP)
       MOV @r1, (SP)
       MOV SP, RO
       TRAP @SPNX
       ADD @4, SP
       JSR PC, 50$
       ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
       BR 60$
       ;EXIT.

50$
      PRINTX @EF9301, R5, R1, R4  ;PRINT THE MESSAGE.
       MOV R4, -(SP)
       MOV R1, -(SP)
       MOV R5, -(SP)
       MOV @EF9301, (SP)
       MOV @4, -(SP)
       MOV SP, RO
       TRAP @SPNX
       ADD @12, SP

60$
      RTS PC  ;RETURN.
       JSR PC, @(SP)$
       ;RESTORE THE GPR CONTENTS.
       ;RETURN TO PREG05 SUBRT.

L10007: TRAP @CMNX

ENDMSG
2070  SBTL GLOBAL SUBROUTINES SECTION
2074
2078
2080
2081
2085
2089

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

```
.SBITL GLOBAL SUBROUTINE . ALTFLD

****  ***********************************************************************
****  ***  ALTER DEVICE REGISTER FIELDS ROUTINE  ***
****  ***  THIS SUBROUTINE ALTERS THE SPECIFIED FIELD OF THE SPECIFIED DEVICE
****  ***  REGISTER FOR ALL SPECIFIED LINES. THIS ROUTINE CAN BE USED TO SET
****  ***  OR CLEAR BITS WITHIN SELECTED FIELDS OF SELECTED REGISTERS.
****  ***  USE EXAMPLES: SET RX.BAUD RATE FIELDS ON LINES 3 AND 6.
****  ***  CLEAR TX.DMA BITS ON ALL LINES.
****  ***********************************************************************

* INPUTS: *
  R1 - ADDRESS OF THE REGISTERS TO ALTER.
  R2 - BIT FIELDS SET TO DESIRED STATES.
  R3 - BIT MAP OF LINES FOR WHICH TO ALTER REGISTER.
  R4 - MASK OF BITS TO ALTER (1 INDICATES CHANGE BIT).
  CSRA - CONTAINS THE ADDRESS OF THE DEVICE CSA.
  ISTAT - SAVED STATES OF THE INTERRUPT ENABLE BITS.

* OUTPUTS: *
  DEVICE REGISTERS - SPECIFIED REGISTER FIELDS ALTERED.
  CSR IND.ADR.REG FIELD - DESTROYED.

* CALLING SEQUENCE: *
  JSA PC,ALTFLD

* COMMENTS: *
  THIS ROUTINE READS THE SPECIFIED REGISTERS FOR ALL LINES
  WITH NUMBERS LOWER THAN THE HIGHEST SPECIFIED LINE.
  THIS ROUTINE DOES NOT READ THE CSR.

* SUBROUTINES CALLED: *
  NONE.
```

```
ALTFLD:: SAVE .SAV CONTENTS OF GPR5 RO THRU R5.
          .JSA RS,PREG5  ;CALL REGISTER SAVE SUBRT.

POUT UP TO LOOP FOR EACH LINE:
  PREPARE THE WORD TO BE ORED INTO THE REGISTER CONTENTS.
  SET UP THE WORD TO WRITE INTO THE IND.ADR.REG FIELD OF THE CSR.

  MOV R4,RO  ;CALCULATE THE NEW CONTENTS OF THE
  COM R0     ;REGISTER FIELDS WHICH ARE TO BE
  BIC R0,R2  ;ALTERED BY THIS ROUTINE.
  MOV ISTAT,R5  ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.

LOOP ONCE FOR EACH LINE, ALTERING THE SPECIFIED FIELD IN THE SPECIFIED
REGISTER IF THE LINE HAS BEEN SELECTED FOR ALTERING.
EXIT THE LOOP IF NO MORE LINES TO ALTER, OR IF WE HAVE ALTERED THE MAX
ALLOWABLE NUMBER OF LINES (AS SPECIFIED BY NUMLN).

  CLC       ;PREPARE FOR ROTATE. "TST RS" DOES THIS BELOW.
  ROR R3    ;GET THE LINE SELECT BIT FOR THIS LINE.
  BCC 41  ;IS SKIP SETUP IF LINE IS NOT SELECTED.
  MOV R5,CSRA  ;SET OUT CSA IND.ADR.REG FIELD TO THIS LINE.
  MOV (R1),RO  ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
  BIC R4,RO  ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
  BIS R2,R0  ;OR IN THE NEW STATES OF THE FIELDS.
  MOV R0,(R1)  ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
  BNE 41  ;SET LINE NUMBER TO THE NEXT LINE.
  TST R5  ;CHECK FOR UNHANDED LINES, CLEAR CARRY FLAG.
  LOOP IF SELECTED LINE(S) IS NOT HANDLED.
```
2141
2142 007436 PASS
007436 004736 JSR PC,0(SP)
2143 007440 RTS PC  ;RETURN TO PREGOS SUBRT.
000207  ;RETURN TO CALLING ROUTINE.
.SBTR subroutine

- Set up associated line number tables routine

This routine sets up the two tables which contain information about the tx/rx line which is associated with a particular rx/tx line. One table is a table of words which contains word offset values and the other table is a table of bytes which contains line number values.

Inputs:
- LOPBCK - Storage for the type of loopback on the dut.
- NUMLS - Equated to the number of lines on the dut.
- STGBTB - Label at base of staggered line association tbl.
- TXRXLB - Label at base of byte tx/rx line number table.
- TXRXLE - Label at end of word tx/rx line number table.

Outputs:
- TXRXL, TXRLN tables initialized for selected loopback.

Calling sequence: JSR PC, ASNLNL

Comments:

Subordinate routines called: None.

ASNLNL: save

JSR R5, PREGS

CALL REGISTER SAVE SUBRT.

CMPB LOPBCK, #2

TEST FOR STAGGERED LOOPBACK.

BEQ 41

GO SET UP STAGGERED TABLE IF STAGGERED LOPBCK.

SET UP THE WORD TABLE FOR NON STAGGERED LOOPBACK.

CLR R5

CLEAR THE LINE COUNTER.

MOV R5, TXRXLB(R5)

SET UP A WORD OF THE TABLE.

INC R5

SET LINE COUNTER TO NEXT LINE OFFSET.

CMP R5, #0

TEST FOR ALL LINES DONE.

BLT 21

LOOP UNTIL ALL LINES DONE.

BR 81

GO SET UP THE BYTE TABLE.

SET UP THE WORD TABLE FOR STAGGERED LOOPBACK.

MOV #STGBTB, R1

SET UP THE SOURCE POINTER.

MOV @TXRXLB, R2

SET UP THE DESTINATION POINTER.

MOVB (R1), (R2)

MOVE A BYTE INTO THE TABLE.

CLRB (R2)

CLEAR THE UPPER BYTE OF WORD TABLE ENTRY.

CMP R2, @TXRXLE

COMPARE POINTER WITH END ADDR OF TABLE.

BLT 61

LOOP IF NOT AT END YET.

SET UP THE BYTE TABLE BASED ON THE WORD ASSOCIATION TABLE.

MOV @TXRXLB, R1

SET UP THE SOURCE POINTER.

MOV @TXRXLB, R2

SET UP THE DESTINATION POINTER.

MOV (R1), R3

GET THE WORD OFFSET VALUE FROM WORD TABLE.

ASR R3

DIVIDE BY 2 TO GET LINE NUMBER VALUE.

MOVB R3, (R2)

LOAD THE BYTE LINE NUMBER INTO TABLE.

CMP R1, @TXRXLE

COMPARE SOURCE POINTER WITH ADDR OF TABLE END.
2201 007544 002172  BLT 101  ; LOOP IF NOT AT END OF TABLE YET.
2202
2203 007546  601:  PASS  ; RESTORE GPR5.
2204 007546  004736  JSR PC,(SP)+  ; RETURN TO PREG05 SUBRT.
2207 007550  000207  RTS PC
GLOBAL SUBROUTINE - CALMSL

; CALMSL: SAVE
CLR 62
JSR RS, PREGS
CALL REGISTER SAVE SUBRT.
CLEAR THE 2ND TIME FLAG.

; SYNCHRONIZE WITH THE LTC.
MOV $1, RS
INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE
; FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <<<
CLR RO
CLEAR THE WAIT FOR CLOCK INT COUNTER.
MOV $1, TMR
SET UP COUNT OF 1 TO SYNCH WITH LTC.
BEQ 64
JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
INC RO
COUNT THIS ITERATION OF THE INNER LOOP.
DEC RS
IF COUNTER HAS NOT TURNED OVER.
BGT 48
IF OUTER LOOP COUNT NOT UP.

; IF WE GOT NO LTC INTERRUPT, INDICATE THAT THERE IS NO LTC AVAILABLE.
; LTC MUST BE FLAKY, OR NOT REALLY AN LTC AT ALL.

; WE ARE NOW SYNCHRONIZED WITH THE LTC.
; SET UP FOR THE CALIBRATION LOOP.
MOV @TMR, RS
WILL TEST TMR IN THE LOOP BELOW.
CLR R1 ; CLEAR THE OUTER LOOP COUNTER.
CLR R2 ; INDICATE TO CHECK ALL BITS OF TIMER1.
CLR R3 ; INDICATE TO CHECK FOR TIMER1 CLEAR.
MOV @1,(A4) ; LOAD MS LOOP COUNT.
 MOV MS,CNT,R5 ; LOAD MS LOOP COUNT.
 MOV (A4),R0 ; GET THE TIMER1 VALUE.
 MOV R0,64 ; SAVE WORD (LIKE IN THE REAL LOOP).
 MOV R1,R2 ; LEAVE ALL THE BITS.
 CMP R0,R3 ; COMPARE AGAINST ZERO.
 SEC ; SET CARRY IN CASE OF SUCCESS.
 BEQ 12$ ; EXIT LOOP IF TIMER1 HAS CLEARED.
 DEC R5 ; COUNT DOWN THE INSIDE MS LOOP COUNT.
 BNE 10$ ; LOOP IF MS NOT UP.
 DEC R1 ; DECREMENT THE MS TIME COUNT.
 BNE 8$ ; KEEP LOOPING.
 JSR PC,005 ; IF OVERFLOWED, SOMETHING IS WRONG, ABORT.

; WE HAVE NOW HAVE LOOP COUNT INFORMATION FOR ONE CLOCK TICK.
; WE HAVE NEGATIVE OF NUMBER OF OUTER LOOPS IN R1. EACH IS MS,CNT INNER LOOPS.
; WE HAVE THE PORTION OF THE LAST OUTER LOOP NOT EXECUTED, IN R5.
; NOW WE CALCULATE THE TOTAL NUMBER OF INNER LOOPS EXECUTED.

NEG R1 ; GET NUMBER OF OUTER LOOPS.
 MOV MS,CNT,R2 ; GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
 MOV R2,R3 ; COPY NUMBER OF LOOPS FOR MULIPLY.
 SUB R5,R2 ; CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP.
 MOV R2,R4 ; AND ADD TO ACCUMULATOR LSWORD.
 CLR R5 ; CLEAR ACCUMULATOR MSWORD.
 DEC R1 ; CHECK R1 FOR 0 CONDITION.
 BMI 16$ ; SKIP MULIPICATION IF ZERO.
 ADD R3,R4 ; MULIPLY NUMBER OF INNER
 ADC R5 ; LOOPS PER OUTER LOOP BY.
 BR 14$ ; NUMBER OF OUTER LOOPS PERFORMED.

; DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER L TC TIC.

MOV MS,TICK,R1 ; # OF MS PER LTC TICK IS DIVISOR.
 MOV R4,R3 ; LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
 MOV R5,R2 ; MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
 JSR PC,UNS DIV ; DIVIDE NUMBER OF LOOPS BY MS PER LTC TIC.
 BCS 18$ ; BYPASS OOPS IF WE'RE OK.
 JSR PC,005 ; CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
 MOV R1,MS,CNT ; SET NEW VALUE FOR MS LOOP COUNT.
 COM 62$ ; SET THE 2ND ITERATION FLGS IF 1ST ITERATION.
 BNE 2$ ; BRANCH IF ONLY ONE ITERATION DONE.
 SEC ; SET THE SUCCESS FLAG FOR EXIT.

PASS ; RESTORE GPR.
 JSR PC,0(3P) ; RETURN TO PREVIOUS SUBRT.
 RTS PC ; CARRY - SUCCESS FLAG. SET IF SUCCESS.

; 2ND CALIBRATION ITERATION FLGS.
.WORD 0
.WORD 0
; DUMMY WORD FOR STORAGE OF THE READ WORD.
GLOBAL SUBROUTINE

**SRATL GLOBAL SUBROUTINE**
**CKTRAP**

**CHECK TRAP ROUTINE**
**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:**
- RO - SOURCE ADDRESS FOR MOVE.
- R1 - DESTINATION ADDRESS FOR MOVE.
- (RO) - SOURCE FOR THE MOVE.

**OUTPUTS:**
- (R1) - WRITTEN TO THE CONTENTS OF (RO).
- CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED.
- TP4FLG - NONZERO IF TRAP OCCURRED, CLEARED OTHERWISE.

**CALLING SEQUENCE:**
- JR PC,CKTRAP

**COMMENTS:**
- IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS WHICH
  IS LABELED ADPRTR WILL BE THE TRAP PC ADDRESS ON THE STACK.
- SUBORDINATE ROUTINES CALLED: NONE.

**CKTRAP:**
- SAVE RO T4RU R5.
- CLR TP4FLG.
- MDV (RO),(R1).
- ADRPRTR: JR PC,CKTRAP.
- SEC.
- BEQ 60+.
- CLC.
- 60+:
- PASS.
- RTS PC.

**ADPRTR:**
- T5 TP4FLG.
- JSR R5,PREGOS.
- JSR PC,R(SP)+.
GLOBAL SUBROUTINE

**DESCRIPTION:**

- **GLOBAL SUBROUTINE**
- **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
- **TXRSCA** - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
- **ERRNBR** - ERROR NUMBER FOR POSSIBLE ERROR REPORT.
- **ERRBL** - ERRTPR, ERRBR, AND ERMSG SET UP CORRECTLY.

**OUTPUTS:**

- **ERRNBR** - VALUE MAY BE DESTROYED.
- **IESTAT** - RX AND RX INTERRUPT FLAGS ARE CLEARED.
- **TX AND RX INTERRUPT ENABLE BITS IN THE DUT’S CSR ARE CLEARED.**

**CALLING SEQUENCE:**

- **JSR PC, CLNRST**

**COMMITS:**

- **THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS ERRNBR.**
- **THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERRNBR.**

**SUBORDINATE ROUTINES CALLED:**

- **DELAY, MSGLE, PUFIFO, RESETT.**

---

**CLNRST:**

- **SAVE CONTENTS OF GPUS R0 THRU R5.**
- **JSR R5, PREGS ;CALL REGISTER SAVE SUBRT.**
- **RESET THE DUT.**
- **THIS ROUTINE REPORTS ERRORS WITH NUMBERS FROM ERRNBR THRU ERRNBR.**
- **JSR PC, RESETT ;RESET THE DUT TO A KNOWN STATE.**
- **BCC 61 ;EXIT ROUTINE WITH ABORT TEST INDICATOR.**
- **PURGE THE FIFO OF ERROR CODES, SAVE ANY BMP CODES FOUND.**
- **JSR PC, PUFIFO ;PURGE THE FIFO.**
- **601 ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.**
- **PASS**
- **JSR PC, BCP ;RETURN TO PREGS SUBRT.**
- **RTS PC ;CARRY BIT: IF CLEAR, THEN ABORT THE TEST.**
GLOBAL SUBROUTINE CMPSM:

**CMPSM**

- **CALLING SEQUENCE:** JSR PC,CMPSM
- **SUBORDINATE ROUTINES CALLED:** NONE.

```
CMPSM:: SAVE
   R5,PREGS
   JSR R5,PREGS
   CALL REGISTER SAVE SUBRT.
CLR R3
   CLEAR THE LINE COUNTER.
MOV #STSTB,R4
   SET STAT STORAGE POINTER TO BASE OF TABLE.
MOV R3,BCSR
   SET UP THE CSR IND.ADR.REG FIELD.
MOV SBLSA,R0
   GET CONTENTS OF THIS LINE'S STAT REGISTER.
MOV (R4),R5
   GET THE PREVIOUS CONTENTS FROM STORAGE.
BIC R0,R5
   XOR PRESENT AND STORED STAT VALUES.
BIC (R4),R0
   XOR PRESENT AND STORED STAT VALUES.
CMPB R3,R1
   TEST FOR THIS BEING SPECIFIED LINE.
BNE 10$
   DON'T MASK OUT SPECIFIED BITS IF IT IS NOT.
BIS R2,R0
   MASK OUT SPECIFIED BITS.
BNE R0,R5
   GET BIT MAP OF UNDESIRABLE CHANGES.
INC R3
   SELECT NEXT LINE.
CMP R5,#NUMLNS
   ALL LINES DONE?
BLT 2$
   LOOP IF NOT ALL LINES DONE.
SEC
   INDICATE SUCCESS.
BR 60$
   EXIT THIS ROUTINE WITH SUCCESS.
50$
   CLC
   INDICATE FAILURE.
60$
   PASS
   RESTORE GPRS.
RTS PC
   JSR PC,B(CSP),
   RETURN TO PREGS SUBRT.
CARRY - SUCCESS Flag (SET IF SUCCESS).```
DELAY: SAVE ;SAVE CONTENTS OF GPRS RO THRU R5.
JSR R5, PREGOS ;CALL REGISTER SAVE SUBRT.

MOV R4, R1 ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
MOV #1, R2 ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
CLR R3 ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
MOV #624, R4 ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
JSR PC, MSLOOP ;DELAY THE REQUESTED # OF MS.
BCC 60$ ;EXIT ROUTINE IF WE TIMED-OUT.
JSR PC, 0OPS ;IF NO TIME OUT, BAD PROGRAM OR HOST MACHINE.
JSR PC, B(SP) ;RESTORE GPRS.
JSA PC, B(SP) ;RETURN TO PREGOS SUBRT.

RTS PC

62$: .WORD -1 ;DUMMY, NON-ZERO WORD.
GLOBAL SUBROUTINE

**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.
- MSLEN - MILLI SECOND SOFTWARE LOOP COUNT.

**OUTPUTS:**
- RO - 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:**
`JSA PC, MSLGET`

**COMMENTS:**
- THIS ROUTINE WORKS WITH OR WITHOUT A HARDWARE CLOCK. BUT THE CALIBRATION IS ONLY GUARANTEED 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:**

```plaintext
010202 004567 173612
010202 005102
010210 040203
```

**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`
- `BIC R2, R3`

**HANDLE THE TEST AND EXIT IF WE HAVE A 0 TIME-OUT VALUE.**
- `TST R1`
- `BNE 2f`
- `MOV (R4), R0`
- `MOV R0, 000070`
- `CMP R0, R3`
- `SEC`
- `BEQ 6f`

**SAVE CONTENTS OF GPRs RO THRU RS.**
**CALL REGISTER SAVE SUBRT.**

**CALL REGISTER SAVE SUBRT.**
- **SAVE CONTENTS OF GPRs RO THRU RS.**
- **CALL REGISTER SAVE SUBRT.**
2531 010234 000241  CLC           :INDICATE FAILURE (TIME OUT).
2532 010236 000416        BR    64           :EXIT WITH FAILURE, WORDS AREN'T EQUAL.
2533                        ;NON-ZERO TIME OUT VALUE. LOOP, WAITING FOR CONDITION OR TIME-OUT.
2534 2535 010240 016705 172034 24:  MOV    MSLCNT,R5       :LOAD MS LOOP COUNT.
2536 010244 011400        44:  MOV    (R4),R0       :GET THE WORD TO TEST.
2537 010246 010067 000042  MOV    R0,624       :SAVE WORD IN CASE THIS IS THE LAST.
2538 010252 040200        MOV    R2,R0       :MASK OUT UTESTED BITS OF WORD.
2539 010254 020003        CMP    R0,R3       :COMPARE AGAINST DESIRED STATE WORD.
2540 010256 000261        SEC    ;SET CARRY IN CASE OF SUCCESS.
2541 010260 001405        BEQ    64;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
2542 010262 005305        DEC    R5       ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2543 010264 001367        DEC    R1       ;LOOP IF MS NOT UP.
2544 010266 005301        BNE    24;DECREMENT THE MS TIME COUNT.
2545 010270 001363        CLC    ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
2546 010272 000241                        ;CLEAR CARRY, WE TIMED-OUT.

2552 010274 016700 000014 64:  MOV    624,R0       ;PASS OUT THE LAST READ WORD.
2553 010500 010066 000002 604:  PASS    R0,R1       ;RESTORE GPRS, EXCEPT THE FOLLOWING:
2554 010304 010166 000004  MOV    R0,ROSLOT(SP) ;PUT R0 IN STACK SLOT.
2555 010310 004736        MOV    R1,RISLOT(SP) ;PUT R1 IN STACK SLOT.
2556 010512 000207        JSR    PRE05_SUBRT. ;RETURN TO PRE05 SUBRT.
2557                        ;RD - LAST READ WORD CHECKED FOR CONDITION.
2558                        ;RL - REMAINING TIME (0 IF TIME-OUT OCCURRED).
2559                        ;RL = CLEAR IF SUCCESS. CARRY IF TIME-OUT.
2560 010314 000000 624:  .WORD    0       ;STORAGE FOR THE LAST READ WORD.
**GLOBAL SUBROUTINE**

**MSLOOP**

**.SBTIL** 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.
  - MSCLNT - 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 guaranteed 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:**

**MSGET**

---

**MSLOOP**

```
.****
MSLOOP:: SAVE JSA R0 THRU R5.
JSR R5,PREC205 I CALL REGISTER SAVE SUBRT.
**
```

---

**CALLING THE MSGET ROUTINE FROM THE MSLOOP ROUTINE ISOLATES THE CALLER OF THE MSLOOP FROM THE RETURNED TEST WORD AND REMAINING TIME-OUT VALUES.

```
JSR PC,MSGET I CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.
```

---

**601:**

```
PASS JSR PC,(BSP) I RESTORE GPRS.
RTS PC I CARRY SET IF SUCCESS, CLEAR IF TIME OUT.
```

---
**GLOBAL SUBROUTINE**

***SBTL* GLOBAL SUBROUTINE***

```
*** 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
JSR RS,PREGOS
; CALL REGISTER SAVE SUBRT.
REPORT "MOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
ERRSF 101,EM0101
```

---

```
TRAP C:ERSF
.WORD 101
.WORD EM0101
```

---

```
REPORT "PROGRAM MUNG, WAITING FOR A CONTROL C."
PRINT #EM0102
```

---

```
MOV @EM0102, (SP)
MOV #1, (SP)
MOV SP,RO
TRAP C:PNIF
ADD $4,SP
```

---

```
BREAK
; LOOK FOR OPERATOR CONTROL-C INPUT.
```

---

```
BR 2:
; INFINITE LOOP.
```

---

```
601: PASS
JSR PC,(SP)
; DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS PC. (SP)
; RETURN TO PREGOS SUBRT.
RTS PC
; ROUTINE IN THE FUTURE, SO BE CONSISTENT.
```

---

EM0101::

```
.ASC 7 /MOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED.```

---

```
010332 004567 173462
010332
010336 104454
010340 000145
010342 010376
010344 000000
010346 012746 010462
010352 012746 000001
010356 010600
010360 010417
010362 062706 000004
010366 010422
010370 000776
010372 004736
010374 000207
010376 110 117 123
010401 124 040 103
010404 117 115 120
010407 125 124 105
010412 122 040 104
010415 101 122 104
010420 127 101 122
010423 105 040 117
010426 122 040 123
010431 117 106 124
010434 127 101 122
010437 105 040 102
010442 125 107 040
010445 105 116 103
010450 117 125 116
010453 124 105 122
```
2636       010456   105  104  056
2636       010461   045  116  045
2636       010465   101  120  122
2636       010470   117  107  122
2636       010473   101  115  040
2636       010476   110  125  116
2636       010501   107  054  040
2636       010504   127  101  111
2636       010507   124  111  116
2636       010512   107  040  106
2636       010515   117  122  040
2636       010520   101  040  103
2636       010523   117  116  124
2636       010526   122  117  114
2636       010531   055  103  056
2636       010534   040  074  052
2636       010537   052  052  052
2636       010542   052  052  052
2636       010545   052  052  052
2636       010550   052  052  052
2636       010553   045  116  045
2636       010556   116  000

2637       .EVEN
GLOBAL SUBROUTINE

PEND - PURGE THE FIFO

THIS ROUTINE TRIES TO REMOVE ALL THE CHARACTERS FROM THE FIFO.

ANY BMP CODES THAT ARE FOUND ARE SAVED ON THE BMP CODE QUEUE.

INPUTS: RBUFA - CONTAINS THE ADDRESS OF THE RECEIVER.

OUTPUTS: CARRY BIT - INDICATES THE STATE OF THE FIFO, SET: PURGED.

BMPQ - THE CONTENTS OF THE BMP CODE QUEUE MAY BE UPDATED.

CALLING SEQUENCE: JSR PC, PUFIFO

COMMENTS:

SUBORDINATE ROUTINES CALLED: SAVBMP.

PFIFO - SAVE

SAVE CONTENTS OF GPRS RO THRU R5.

JSR PC,PREGOS, CALL REGISTER SAVE SUBRT.

MOV 0512,R1
GET MAXIMUM TRY COUNT OF 512.

MOV RBUFA,R4
GET ADDRESS OF THE RECEIVER BUFFER REGISTER.

MOV (R4),R2
GET THE CONTENTS OF THE RECEIVER BUFFER REG.

BPL 61
EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.

CHECK IF THE READ CHARACTER IS ACTUALLY A BMP CODE.

IF IT IS, THEN SAVE IT ON THE BMP CODE QUEUE TO BE REPORTED LATER.

MOV 070000,R0
GENERATE A BIT MAP OF CHAR ERROR BITS

BIC R2,R0
WHICH ARE NOT SET FOR CHAR.

BNE 41
THROW CARRY AWAY IF NOT BMP OR SELFTEST CODE.

CHECK IF THE READ DATA IS MODEM STATUS, BMP OR SELFTEST?

MOV 0301,R0 CHECK IF BMP.

BIC R2,R0
TRY TO CLEAR BMP FLAGS IN THE READ DATA.

BNE 41 IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.

JSR PC, SAVBMP SAVE BMP CODE ON THE QUEUE.

DEC R1 DECREMENT THE TRY COUNT.

BNE 21 LOOP TO TRY AGAIN.

CLC CLEAR CARRY, TO INDICATE FIFO NOT PURGED.

BRC 601 EXIT WITH CARRY CLEAR.

SEC SET CARRY, TO INDICATE FIFO PURGED.

PASS 01040
RESTORE GPRS.

JSR PC,(PS), RETURN TO PREVIOUS SUBRT.

CARRY BIT, SET INDICATES FIFO PURGED.

RTS PC
2689 GLOBAL SUBROUTINE
2690 RESELT
2691
2692 .SBTL GLOBAL SUBROUTINE
2693 ;******************************************************************************
2694 ; \* - RESET DEVICE UNDER TEST
2695 ; \* THIS SUBROUTINE IS USED TO RESET THE DUT TO A KNOWN STATE.
2696 ; \* IF RESET DOES NOT SUCCESSFULLY COMPLETE, IE. TIME-OUT OCCURS, THEN
2697 ; \* AN ABORT TEST ERROR MESSAGE IS REPORTED.
2698 ; \* INPUTS:
2699 ; \* CSRA - CONTAINS THE ADDRESS OF THE CSR
2700 ; \* INBFRCA - CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
2701 ; \* ERRBL, ERRTP, ERNBR, AND ERMSG SET UP CORRECTLY.
2702 ; \* OUTPUTS:
2703 ; \* THE DUT PERFORMS ITS RESET FUNCTION INTO A KNOWN STATE.
2704 ; \* CARRY - CLEAR INDICATES THE TEST IS TO BE ABORTED.
2705 ; \* ERRBLK - VALUE MAY BE DESTROYED.
2706 ; \* IESTAT - TX AND RX INTERRUPT FLAGS ARE CLEARED.
2707 ; \* TX AND RX INTERRUPT ENABLE BITS IN THE DUT'S CSR ARE CLEARED.
2708 ; \* CALLING SEQUENCE: JSR PC,RESELT
2709 ; \* COMMENTS: THIS SUBROUTINE CAN REPORT ERRORS WITH NUMBERS INITIAL ERNBR
2710 ; \* THIS ROUTINE DOES NOT DESTROY THE VALUE OF ERNBR.
2711 ; \* SUBORDINATE ROUTINES CALLED: DELAY, MSLGET.
2712 ;******************************************************************************
2713 RESELT: SAVE JSR ; SAVE CONTENTS OF GPR S R0 THRU R5.
2714 010642 JSR R5, PREGOS \* CALL REGISTER SAVE SUBRT.
2715 000040 JSR R5, PREGOS \* CALL REGISTER SAVE SUBRT.
2716 000040 MOV 0BITOS, R2 ; SET BIT MASK OF MASTER RESET BIT.
2717 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2718 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2719 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2720 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2721 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2722 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2723 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2724 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2725 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2726 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2727 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2728 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2729 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2730 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2731 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2732 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2733 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2734 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2735 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2736 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2737 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2738 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2739 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2740 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2741 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2742 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2743 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2744 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2745 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2746 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2747 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2748 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2749 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2750 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2751 000040 MOV CSRA, R4 ; GET THE ADDRESS OF THE DUT'S CSR.
2745 2746 2747 2748 2749 2750 2751 2752 2753 2754 2755 2756 2757 2758 2759 2760 2761 2762 2763 2764 2765 2766

1: SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET, TEST ABORTED".
   INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
44: MOV @EM1501.R1 ; PASS ERROR MESSAGE TO REPORT.
2750 010726 012767 006570 173062
44: MOV @ER1003.ERRBLK, PASS ADDRESS OF ERROR HANDLING ROUTINE.
   REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR",
   "TEST ABORTED"
    ERROR:
    ERR <<<<< TRAP ERR
    CLR C
    BR 60
    :INDICATE TEST IS TO BE ABORTED.
    :EXIT THIS SUBROUTINE. ABORT TEST INDICATOR.
1: CLEAR TX AND RX INTERRUPT ENABLE STATUS FLAGS IN IESTAT.
1: EXIT WITH CONTINUE TEST INDICATOR SET (IE, CARRY SET).
64: CLR IESTAT ; CLEAR TX AND RX INTERRUPT STATUS FLAGS.
2761 010746 000261
64: SEC ; INDICATE SUCCESS, CONTINUE TEST.
2762
60: PASS JSR PC, @ (SP), ; RETURN TO PREGOS SUBRT.
2763 010750 004736
60: RTS PC ; CARRY BIT: IF CLEAR, INDICATES ABORT TEST.
.SBITL 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.
- BMPCL - CONTAINS ADDRESS OF NEXT LOCATION IN THE BMP QUEUE.
- BMPQB - LABEL AT BASE OF THE BMP CODE QUEUE.
- BMPGR - LABEL OF NEXT LOCATION AFTER THE END OF THE BMP QUEUE.
- TSTNUM - CONTAINS THE NUMBER OF THE CURRENT TEST.

**OUTPUTS:**
- BMPGQ - 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 RO THRU R5.
- JSR R5, PREGS ; CALL REGISTER SAVE SUBRT.
- MOV R5, 173040 ; GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
- MOV R4, 171364 ; GET THE CURRENT TEST NUMBER ON THE QUEUE.
- INC R4 ; INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
- MOV R2, 177400 ; CLEAR THE UNWANTED BITS FROM THE BMP CODE.
- MOV R2, (R4) ; SAVE THE BMP CODE ON THE QUEUE.
- CMP R4, (BMPQG) ; CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
- BLO 2: ; GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
- SUB R4, R4 ; RESET THE POINTER TO THE LAST LOCATION IN QUE.
- MOV R4, BMPQG ; SAVE THE POINTER.

2: ; RESTORE GPRS.
- JSR PC, @SP, ; RETURN TO PREGS SUBRT.
- RTS PC

604: PASS

---
GLOBAL SUBROUTINE

*** - SAVMST -

- SAVE MODEM STATUS ROUTINE

- THIS ROUTINE SAVES THE PRESENT CONTENTS OF THE OUT STAT REGISTERS IN
  THE STAT STORAGE TABLE.

- INPUTS:
  CSRA - CONTAINS THE ADDRESS OF THE OUT CSR.
  ISTAT - STATE OF THE OUT CSR INTERRUPT ENABLE BITS.
  NUMLNS - EQUIVALENT TO THE NUMBER OF LINES ON THE OUT.
  FLSLA - CONTAINS THE ADDRESS OF THE OUT STAT REGISTER.
  SISTB - LABEL AT BASE OF THE STAT STORAGE TABLE.

- OUTPUTS:
  SIST TABLE - OVERWRITTEN WITH PRESENT STAT CONTENTS.
  CSR INO.ADR.REG FIELD - DESTROYED.

- CALLING SEQUENCE:
  JSR PC, SAVMST

- COMMENTS:
  IF THE CONTENTS OF ISTAT CHANGES DURING THIS TEST THE CSR
  INTERRUPT ENABLE BITS WILL NOT TRACK THE CHANGE.

- SUBORDINATE ROUTINES CALLED: NONE.

SAVMST: SAVE

;SAVE CONTENTS OF GPR5 RO THRU R5.

JSR R5, PREGOS ;CALL REGISTER SAVE SUBRT.

MOV IESTAT, R1 ;GET I STAT BUS FOR UPDATING INO.ADR.REG FIELD.

MOV @SISTB, R2 ;SET UP STAT STORAGE POINTER TO BASE OF TABLE.

MOV @NUMLNS, R3

BIS R1, R3 ;FORM COMPLETION COMPARISON WORD.

MOV R1, BC SRA ;SET UP THE CSR INO.ADR.REG FIELD.

MOV BFSLSA, (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

JSR PC, 0(SP) ;RESTORE GPRS.

RTS PC ;RETURN TO PREGOS SUBRT.
; SBTL: GLOBAL SUBROUTINE - SKPSTS
; **********************************************************
;  ^ -----  ^
;  THIS SUBROUTINE IS USED TO SKIP THE SELFTEST AFTER A DUT RESET HAS BEEN
;  INITIATED. IT MUST BE ENTERED IMMEDIATELY AFTER SETTING THE DUT MASTER
;  RESET ROUTINE OR AFTER THE EXECUTION OF A BUS RESET (BECAUSE OF TIMING
;  CONSIDERATIONS).
;  ^
;  INPUTS: CSRA - CONTAINS ADDRESS OF THE DUT CSR.
;  ^
;  TXREFCA CONTAINS ADDRESS OF DUT DMA BUFFER COUNT REGISTER.
;  ^
;  OUTPUTS: SKIP SELFTEST CODES ARE WRITTEN TO THE DUT REGISTERS.
;  ^
;  CALLING SEQUENCE: JSR PC,SKPSTS
;  ^
;  COMMENTS:
;  ^
;  SUBORDINATE ROUTINES CALLED: DELAY.
;  **********************************************************

; SKPSTS: SAVE
; JSR RS,PREGOS ; CALL REGISTER SAVE SUBRT.
; MOV @10.,R4 ; PASS DELAY VALUE OF 10 MILI SECONDS
; JSR PC,DELAY ; DELAY FOR 10 MILI-SECONDS.

; WRITE SKIP SELF-TEST CODE (52525) TO ALL THE Indexed DUT REGISTERS.
; MOV @NUMLS:BITOS,R1 ; FORM IND.ADR.REG FIELD (PLUS M.R. BIT) WORD.
; THE ABOVE INCLUSION OF THE M.R. BIT IS NECESSARY BECAUSE OF THE
; LACK OF A M.R. BIT WRITE LOCK-OUT ON THE DHU-11.
; MOV #52525,R3 ; INITIALISE THE SKIP SELF-TEST CODE.
; DEC R1 ; SELECT THE NEXT SET OF DEVICE REGISTERS.
; MOV CSR,A,R4 ; GET THE ADDRESS OF THE CSR OF THE DUT.
; MOV R1,(R4) ; SELECT A BANK OF DUT REGISTERS.
; MOV R3,(R4) ; WRITE THE CODE TO A DUT REGISTER.
; CMP R4,1XBFCA ; COMPARE POINTER WITH LAST REGISTER ADDRESS.
; BLO 6$ ; LOOP IF NOT ALL REGS DONE IN THIS BANK.
; BIT #17,R1 ; TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
; BNE 4$ ; LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.

; 601: PASS
; JSR PC,(SP) ; RETURN TO PREGOS SUBRT.
; RTS PC
global subroutine unsdiv

;******************************************************************************
;* SUBTL GLOBAL SUBROUTINE     UNSDIV  
;******************************************************************************
;* unsigned divide routine  
;* this subroutine is used to divide a 32 bit unsigned dividend by a 
;* 16 bit unsigned divisor giving a 16 bit quotient. all numbers are 
;* considered to be unsigned. a success flag is not set on return if 
;* the quotient was too big to be contained in 16 bits.  
;* inputs: r1 - the divisor, unsigned, 16 bits.  
;*         r2 - most significant word of the dividend, unsigned, 16 bits.  
;*         r3 - least significant word of the dividend, unsigned, 16 bits.  
;* outputs: r1 - quotient, unsigned, 16 bits (777777 if overflow).  
;*          carry success flag, set if complete quotient fits in 16 bits.  
;* calling sequence: jsr pc,unsdiv  
;* comments: if the divisor is 0 the quotient is returned as all ones 
;*           (777777) and the carry is clear regardless of the dividend.  
;* subordinate routines called: none.  
;******************************************************************************

unsdiv::: save
jsr r5,preg05 ;call register save subrt.

;* check for quotient greater than 16 bits condition.
;
mov r2,r4 ;get msb of dividend for subtract.
sub r1,r4 ;subtract divisor from msb of dividend.
bsc 2,1 ;if it didn't go, we have quotient < 16 bits.
mov 0,1,r1 ;set quotient to all ones (777777).
br 60$ ;exit with carry clear.

;* set up counters and various working gprs.
;
clr r4 ;clear the lsw of the divisor.
clc ;clear carry for the shift of the divisor.
rdr r1 ;divisor by
rdr r4 ;(unsigned)
mov 016..r0 ;set up initial shift count to 16.

;* the subtract and shift loop.
;
41: mov r2, (sp) ;save msword of dividend.
mov r3, (sp) ;save lsword of dividend.
sbc r4,r3 ;lsword dividend - lsword of divisor.
sbc r2, r0 ;msword dividend - borrow.
bsc 64,1 ;if borrow from borrow subtract, it didn't go.
sbc r1,r2 ;msword dividend - msword of divisor.
bcc 84$ ;if no borrow, it went, carry is clear.

;* it didn't go, so we shift a 1 into the quotient (complemented later).
;
;* carry is set.
;
51: mov (sp)+,r3 ;restore lsword of dividend.
mov (sp)+,r2 ;restore msword of dividend.
BR 10$ ;GOTO SHIFT 1 INTO THE QUOTIENT.

; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
; COMPLEMENTED LATER). CARRY IS CLEAR.

8$; MOV (SP)+,(SP) ;POP THE SAVED DIVIDEND OFF OF THE STACK.

; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.

10$; ROL R5 ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.

CLC ;DIVIDE THE

ROR R1 ;DIVISOR BY

ROR R4 ; 2 (UNSIGNED).

DEC R0 ;COUNT THIS SHIFT AND SUBTRACT.

BNE 4$ ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.

COM R5 ;GET QUOTIENT FROM INVERTED QUOTIENT.

; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.

; CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.

12$; ROL R3 ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.

BCS 12$ ;IF CARRY FROM SHIFT, ROUND UP.

SUB R4,R3 ;SUBTRACT DIVISOR FROM DIVIDEND.

14$ ;IF BORROW, DON'T ROUND UP.

; ROUND UP, EXTRA SUBTRACT WENT.

; INCREMENT THE QUOTIENT BY ONE.

14$; INC R5 ;IF NO OVERFLOW, WE LEAVE THE ROUND UP

BNE 14$ ;DON'T LET ROUNING CAUSE OVERFLOW.

; ALL DONE, PASS QUOTIENT AND EXIT.

; PASS QUOTIENT BACK IN R1.

14$; MOV R5,R1 ;INDICATE NO OVERFLOW.

SEC

; RESTORE GPRS, LEAVE THE FOLLOWING INACT:

60$; PASS R1 MOV R1,R1LSLOT(SP) ;PUT R1 IN STACK SLOT.

JSR PC,BS(SP) ;RETURN TO PREG05 SUBRT.

; (R1 - 16 BIT, UNSIGNED QUOTIENT).

RTS PC ;CARRY - SET INDICATES NO OVERFLOW (SUCCESS).
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.

* 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 0130040,R1 ;PASS BIT 11 (13 OCTAL) AND
  ; 32 (40 OCTAL) MS DELAY.
  MOV 0LABEL,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

JSR RS,PREGOS ;CALL REGISTER SAVE SUBRT.

MOV R2,R4 ;SET UP THE ADDRESS PARAMETER FOR MSLGET.

MOV R1,R2 ;SEPERATE DELAY COUNT OUT OF PASSED PARAM.

BIC 0170000,R1 ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.

BIC 0777777,R2 ;SEPERATE LINE NUMBER FIELD INTO THE PROPER

SHR R2 ;POSITION TO USE IT AS A WORD TABLE OFFSET

ASR R2 ;FOR THE TABLE LOOKUP OF THE LINE BIT MAP.

MOV BITBL(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.

MOV R0,R2 ;PASS LAST VALUE READ AS OUTPUT PARAMETER.

604: MOV R2,R2 ;RESTORE GPRS. EXCEPT THE FOLLOWING:

R2,25SLT(SP) ;PUT R2 IN STACK SLOT.

JSR PC,(SP) ;RETURN TO PREGOS SUBRT.

RTS PC ;R2 - LAST VALUE READ LOOKING FOR CONDITION.

CARR: - SUCCESS FLAG (SET IF BIT FOUND CLR).
SUBLT GLOBAL SUBROUTINE

- WAIT FOR BIT SET ROUTINE -
  THIS SUBROUTINE WAITS FOR THE SPECIFIED BIT TO BECOME SET. IF THE
  SPECIFIED BIT GOES TO A SET STATE WITHIN THE SPECIFIED TIME OUT
  PERIOD A SUCCESS INDICATION IS RETURNED BY THIS ROUTINE
  THE LAST VALUE WHICH IS READ LOOKING FOR THE CONDITION IS RETURNED TO
  ALLOW THE USE OF THIS ROUTINE TO LOOK FOR DESTRUCTIVE READ CONDITIONS.

  INPUTS:  A1 - 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 MILI-SECONDS (4095 MAX).
  A2 - ADDRESS OF WORD CONTAINING THE BIT TO TEST.

  MSLCNT.

  OUTPUTS:  A2 - THE LAST WORD WHICH WAS READ TO CHECK FOR THE CONDITION.
  CARRY - SUCCESS FLAG (CARRY SET IF BIT SET BEFORE TIME-OUT).

  CALLING SEQUENCE: MOV A130040.H1 ; PASS BIT 11 (13 OCTAL) AND
                  ; 32 (40 OCTAL) MS DELAY.
                  ; MOV 0LABEL,H2 ; TEST BIT IN WORD AT "LABEL".
                  ; JSR PC,WAIIBS ; WAIT 32 MS FOR BIT 11 TO SET.

  SUBROUTINE Routines Called: MSLGET.

WAIIBS:

  SAVE R5,PREGS ; CALL REGISTER SAVE SUBRT.
  SAVE CONTENTS OF GPRS RO THR US R5.
  SET UP THE ADDRESS PARAMETER FOR MSLGET.
  SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
  SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
  PUT LINE NUMBER FIELD IN LSBYTE.
  SHIFT THE LINE NUMBER FIELD INTO THE PROPER
  POSITION TO USE IT AS A WORD TABLE OFFSET
  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
  GET BIT MAP OF LINE TO TEST FROM TABLE.
  INDICATE THAT THE BIT SHOULD BE SET.
  WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
  CARRY IS CORRECT UPON MSLGET RETURN.

  PASS LAST VALUE READ AS OUTPUT PARAMETER.
  RESTORE GPRS, EXCEPT THE FOLLOWING:
  PUT R2 IN STACK SLOT.
  RETURN TO PREDS SUBRT.
  R2 - LAST VALUE READ LOOKING FOR CONDITION.
  CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).
.SBITL GLOBAL SUBROUTINE

/* ----------------------------- */
/* THIS SUBROUTINE IS USED TO SET THE DEVICE UNDER TEST (OUT) LINE */
/* CONTROL REGISTERS (LNCTRL) TO THE SPECIFIED STATE. ONLY THE LNCTRLS */
/* FOR THE SPECIFIED LINES ARE ALTERED. */
/* */
/* INPUTS: */
/* R0  NEW LINE PARAMETERS. */
/* R5  BIT MAP OF LINES TO BE ALTERED. */
/* CSRA - CONTAINS ADDRESS OF THE OUT CSR. */
/* IESTAT - CONTAINS CURRENT STATE OF THE TX AND RX INTERRUPT */
/* ENABLE BITS IN THE CSR. */
/* LNCTRL - CONTAINS ADDRESS OF THE OUT LNCTRL REGISTERS. */
/* */
/* OUTPUTS: */
/* LNCTRL - SPECIFIED OUT LINE CONTROL REGISTERS ARE ALTERED. */
/* */
/* CALLING SEQUENCE: JSR PC,WTWLNC */
/* */
/* COMMENTS: */
/* */
/* SUBORDINATE ROUTINES CALLED: ALTFLD. */
/* */
/* ----------------------------- */

WTWLNC:: SAVE 

;SAVE CONTENTS OF GPRS R0 THRU R5. 

JSR R5,PREGOS ;CALL REGISTER SAVE SUBRT. 

;SET UP THE PARAMETERS FOR THE CALL TO ALTFLD. 

;MOV LNCTRLA,R1 ;SET UP THE REGISTER ADDRESS PARAMETER. 

MOV R0,R2 ;SET UP THE DESIRED REGISTER CONTENTS. 

MOV R5,R3 ;SET UP THE BIT MAP OF LINES TO ALTER. 

MOV #1,R4 ;SELECT ALL REGISTER BITS TO BE ALTERED. 

;CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS. 

JSR PC,ALTFLD ;ALTER THE REGISTER CONTENTS. 

604: PASS 

JSR PC,(SP) ;RESTORE GPRS. 

RTS PC ;RETURN TO PREGOS SUBRT.
**SBITL** INTERRUPT SERVICE ROUTINE

---

* **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 SBITL IN THE CLOCK INTERRUPT VECTOR SLOT.
  - PUT THE DESIRED TIME PERIOD (SECONDS TIMES CLKHZ) 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:**
  - CLI

---

```
3154 011460 005767 170602 CLKINT:: TST TIMER1 ; CHECK FOR TIMER1 AT ZERO.
3155 011464 001402 BEQ 24 ; BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
3156 011466 005367 170574 DEC TIMER1 ; DECREMENT TIME COUNT.
3157 011472 005767 170572 24: TST TIMER2 ; CHECK FOR TIMER2 AT ZERO.
3158 011476 001402 BEQ 41 ; BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
3159 011500 005367 170564 DEC TIMER2 ; DECREMENT TIME COUNT.
3160 011504 005367 170562 41: DEC TIMER3 ; DECREMENT THE BREAK COUNT.
3161 011510 001006 BNE 60$ ; EXIT IF NOT TIME TO CALL BREAK.
3162 011512 016767 170556 170552 MOV BCOUNT,TIMER3 ; SET UP TIME TILL NEXT BREAK.
3163 011520 010046 MOV RO, -(SP) ; SAVE CONTENTS OF RO FROM BREAK MACRO.
3164 011522 104422 BREAK ; CHECK FOR OPERATOR CONTROL/C. TRAP CIBRK
3165 011524 012600 MOV (SP)+,RO ; RESTORE CONTENTS OF RO.
3166 011526 000002 60$: RTI
```
GLOBAL TRAP SERVICE ROUTINE

********************************************************************************
** BUS TIME-OUT TRAP (004 TRAP) SERVICE ROUTINE
** THIS ROUTINE DETERMINES IF THE 004 TRAP WAS CAUSED BY AN "EXPECTED" ERROR OR
** NOT BY EXAMINING THE RETURN PC VALUE ON THE STACK. IF THE TRAP IS UNEXPECTED,
** THIS ROUTINE JUMPS TO THE NORMAL DIAGNOSTIC SUPERVISOR 004 TRAP HANDLING
** ROUTINE.
**
** INPUTS: SP - POINTS TO THE PC WHERE THE TRAP OCCURRED.
** ADAPTR - LABEL AT THE ADDRESS WHERE "EXPECTED" TRAPS OCCUR.
** TP4FLG - 004 TRAP FLAGS.
**
** OUTPUTS: TP4FLG - BIT 15 IS SET IF "EXPECTED" TRAP OCCURRED.
**
** CALLING SEQUENCE: PUT ADDRESS POINTED TO BY TP4RTN IN 004 VECTOR.
** OCCURRENCE OF 004 TRAP VECTORS TO THIS ROUTINE.
**
** COMMENTS: ANY 004 TRAP WHICH OCCURS AT AN ADDRESS OTHER THAN THAT LABELED
** ADAPTR WILL BE HANDLED BY THE NORMAL 004 TRAP SERVICE ROUTINE.
**
** SUBORDINATE ROUTINES CALLED: NONE.
********************************************************************************

TP4RTN: CMP (SP),ADAPTR ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
BEQ 2: ;IF THEY MATCH, CONTINUE THIS ROUTINE.
JMP BT4VEC ;IF NOT, JUMP TO NORMAL 004 TRAP SERVICE RTN.
BIS @BIT15,TP4FLG ;SET THE 004 TRAP OCCURRED FLAG.
ATI ;DONE, GO BACK TO THE TEST.
.SBTL REPORT CODING SECTION

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

BGNRPT

EXIT RPT

.EVEN

ENDRPT

.L10010: TRAP CRPT

.L1RPT::

.WORD J1JMP

.WORD L10010-2..
.SBTTL PROTECTION TABLE

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

011560       BGNPROT
011560
11       OFFSET INTO P TABLE FOR CSR ADDRESS
011562       -1       OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
011564       -1       OFFSET INTO P-TABLE FOR DRIVE NUMBER
011566       ENDPROT
.SBTL 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 0EF.START
    .IN:
        MOV    #EF.START,RO
        TRAP   CREFG
        BCOMPLETE NEWSTA
        BCS    NEWSTA
    .OUT:
        MOV    #EF.RESTART,RO
        TRAP   CREFG
        BCOMPLETE NEWRES
        BCS    NEWRES
    .IN:
        MOV    #EF.NEW,RO
        TRAP   CREFG
        BCOMPLETE NEWPAS
        BCS    NEWPAS
    .OUT:
        MOV    #EF.CONTINUE,RO
        TRAP   CREFG
        BCOMPLETE GETPRM
        BCS    GETPRM
    .OUT:
        BRESET
        ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
        JMP    ENDIT
    .OUT:
        ;SET UP FOR LINE TIME CLOCK INTERRUPTS.
        ;
        CLOCK L.R1
        ;GET THE CLOCK PARAMETERS.
        MOV    0;L.R0
        TRAP   CICLCK
        MOV    RO.R1
        MOV    (R1)+,CLKCSR
        ;STORE CLOCK CSR ADDRESS.
        MOV    (R1)+,CLKBCR
        ;STORE CLOCK BUS REQ INT LEVEL.
        MOV    (R1)+,CLKVEC
        ;STORE CLOCK INTERRUPT VECTOR.
        MOV    (R1)+,CLKHIZ
        ;STORE CLOCK FREQUENCY.
        CMP    CLKHIZ,050.
        ;TEST FOR 50HZ LINE FREQUENCY.
        BNE    21
        ;BRANCH IF CLOCK IS NOT 50HZ.
3286 011674 012767 000024 170374 MOV @00..MSTICK ;INDICATE 20MS PER CLOCK TICK.
3287 011702 000403 BR 41
3288 011704 012767 000021 170364 24: MOV @17..MSTICK ;INDICATE 17 MS PER CLOCK TICK.
3289 011712 012746 000300 170534 SETVEC CLKVEC,CLKINT,#PRI06 ;INITIALIZE CLOCK INTERRUPT VECTOR.
3290 011716 012746 011460 MOV @PT06..(SP) ;
3291 011722 016746 170334 MOV @CLKINT..(SP) ;
3292 011726 012746 000003 MOV @05..(SP) ;
3293 011732 104437 TRAP C#5VEC ;
3294 011734 062706 000010 ADR #10,SP
3295 011740 016700 170320 MOV CLKH2Z,RO
3296 011744 006500 PSL RO
3297 011746 010307 170322 MOV RO,BCOUNT ;TO CAUSE A BREAK.
3298 011752 170322 MOV #0,R0 ;EVERY 2 SECONDS.
3299 011752 012700 000240
3300 011756 104441

;ENABLE THE LINE TIME CLOCK (LTC) CHECKING TO MAKE SURE THAT THE CSR
;IS ACCESSIBLE.
;FIRST SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:

3301 011760 016767 166020 170262 MOV 4,TP4VEC ;SAVE THE EXISTING 004 TRAP VECTOR.
3302 011766 012767 011530 166010 MOV @PARTN,4 ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.

;ENABLE LTC CHECKING FOR 004 TRAP IN CASE CSR IS NOT THERE.

3304 011774 005067 170252 CLR TP4FLG ;CLEAR THE 004 TRAP FLAG.
3305 011780 012600 001000 170246 MOV #BIT6,WORD1 ;SET UP TO SET BIT6 OF THE LTC CSR.
3306 011785 012600 010000 170246 MOV #W001 AS THE CKTRAP MOVE SOURCE.
3307 012012 016701 170240 MOV CLKCSR,R1
3308 012016 004767 175754 JSR PC,CKTRAP ;MOVE AND CHECK FOR TRAP.
3309 012022 016767 170222 165754 MOV TP4VEC,4 ;RESTORE THE NORMAL 004 TRAP VECTOR.
3310 012030 103405 BCS 61
3311 012032 005067 170226 CLR CLKH2Z ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
3312 012036 000402 BR 81 ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.

;CALIBRATE THE DELAY ROUTINE MILLI-SECOND DELAY COUNT VALUE.

3313 012040 004767 175506 61: JSR PC,CALMSL
3314 012044 016767 165734 170176 64: ;CHECK FOR MEMORY MANAGEMENT PRESENT ON THIS MACHINE.
3315 012048 011530 165724 IF MEM MGT IS PRESENT, DISABLE IT.
3316 012052 012767 011530 165724 MOV @PARTN,A ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
3317 012060 005067 170166 CLR TP4FLG ;CLEAR THE 004 TRAP FLAG.
3318 012064 005067 170164 CLR WORD1 ;PREPARE TO CLEAR THE MEM MGT SRO REGISTER.
3319 012070 012700 002254 MOV #W001,RO ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
3320 012074 016701 170202 MOV #M500,R1 ;SELECT MEM MGT SRO REGISTER AS DESTINATION.
3321 012078 005067 170200 CLR M5R5S3 ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
3322 012104 005067 170176 CLR M5N5AB ;INDICATE MEM MGT IS NOT ENABLED.
3323 012108 004767 175662 JSR PC,CKTRAP ;CLEAR THE MEM MGT SRO REG AND CHECK FOR TRAP.
3324 012112 016767 170130 165662 MOV TP4VEC,4 ;RESTORE THE NORMAL 004 TRAP VECTOR.
3325 012116 103003 BCC 101; ;SKIIuI MNG MGT PRESENT IF IT ISN'T.
3326 012120 012767 000001 170152 MOV #1,MMPRES ;INDICATE THAT MEM MGT IS PRESENT.
3327 012124 005067 170100 101: CLR PASCNT ;CLR COUNTER USED IN REPORTING ROM VERSION A.
3328 012132 000167 000006 JMP NEWPAS ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
; INITIALIZE SECTION

;NEWRES: BRESET  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
;  TRAP CI; RESET

;NEWPAS: CLRPASN  ;CLR COUNTER USED IN REPORTING ROM VERSION.
;  MOV 01,UNITN  ;RESET LOGICAL DEVICE TO 1.

;  INC PASCNT  ;INCREMENT THE PASS COUNTER. CORRECT FOR ANY OVERFLOW.
;  BNE GETPRM  ;BRANCH IF WE HAVE NOT YET OVERFLOWED.
;  DEC PASCNT  ;SET PASS COUNT TO 1777777777.

;  GET THE HARDWARE PARAMETERS FOR THIS UNIT.
;  GETPRM:

;  INC UNITN  ;INCREMENT LOGICAL DEVICE NUMBER
;  CMP UNITN,1;UNIT
;  BNE NEWPAS  ;SEE IF MAXIMUM UNIT NO. EXCEEDED

;  GPHARD UNITN,R1  ;GET P TABLE POINTER INTO R1
;  MOV UNITN,RO  ;TRAP CI;GPHARD
;  MOV RO,R1

;  BCOMPLETE 30$  ;BR IF DEVICE AVAILABLE
 ; BCS 30$  ;SKIP THIS DEVICE

;  *************************************************** HARDWARE PARAMETER MOVING CODE ***************************************************

;  MOV (R1)*,CSRA  ;STORE DHU-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
;  MOV (R1)*,ACTLNS  ;STORE DHU-11 ACTIVE LINE BIT MAP
;  MOV (R1)*,LOOPBACK  ;STORE DHU-11 LOOPBACK MODE

;  CALCULATE DEVICE REGISTER ADDRESSES, AND PUT THEM IN THE DEVICE REGISTER ADDRESS TABLE.

;  MOV CSRA,R1  ;COPIE CSR ADDRESS
;  INC R1  ;INCREMENT CSR ADDRESS
;  INC R1  ;COPY BY 2.
;  MOV @7,R3  ;SET UP REGISTER COUNT
;  MOV @ABUFA,R2  ;GET LOCATION WHERE AUF ADDRESS GOES IN TABLE

;  MOV R1,(R2)  ;STORE REGISTER ADDRESS IN TABLE
;  INC R1  ;INCREMENT REGISTER ADDRESS
;  INC R1  ;BY 2, FOR THE NEXT DEVICE REGISTER.
;  DEC R3  ;DECCREMENT REGISTER COUNT
;  BNE 12$  ;LOOP IF NOT DONE

;  INITIALISE THE BMP CODE QUEUE.

;  MOV @BMPCCB,RO  ;GET THE START ADDRESS OF THE QUEUE.
;  MOV @BMPCCB.R1  ;GET THE END ADDRESS OF THE QUEUE.
;  MOV RO,BMPCCP  ;SET THE POINTER TO THE START OF THE QUEUE.
;  CLR (RO)  ;CLEAR OUT THE CONTENTS OF THE QUEUE.
;  CMP RO,R1  ;CHECK IF END OF QUEUE HAS BEEN REACHED.
;  BLO 14$  ;LOOP IF NOT ALL DONE.
3387 3388 3389 3390 3391 012310 032767 000020 167644
3392 012316 001416
3393 012320 026727 167466 000001
3394 012326 003412
3395 012330 012354 012746 004601
3396 012350 062706 000006
3397
3398 012354 005067 167642
3399
3400 3401 3402 012360 012700 000340
3403 012366 012366 104441
3404 3405 012366 102700 000340
3404 012366 104441
3405 000000


BIT @BIT4, OPTION ; CHECK IF THE QUESTION WAS ANSWERED YES.
BEQ 164 ; SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
CMP @UNIT, @1 ; CHECK MAXIMUM NUMBER OF UNITS SELECTED.
BLE 164 ; DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
PRINTF @#UNIT, UNIT ; REPORT UNIT NUMBER.

MOV @UNIT, -(SP)
MOV @@UNIT, -(SP)
MOV @2, -(SP)
MOV @SP, RO
TRAP CIPNTF
ADD @#6, SP

164:

ENDIT: CLR CTRLCF ; CLR THE CTRL C TEST ABORT FLAG.

'' SET THE PROCESSOR PRIORITY TO DISABLE ALL INTERRUPTS.

SETPRI @PRI07 ; SET PROCESSOR PRIORITY TO 7.
MOV @PRI07, RO
TRAP CISPRI

ENDINIT

L10012: TRAP CINIT

TNUM := 0 ; INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.
SBTTL AUTODROP SECTION

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

BGNAUTO

ENDAUTO

L#AUTO:

L10013:  TRAP C#AUTO
.SBTTL CLEANUP CODING SECTION

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

 BGNCLN

 L$CLEAN:

 IST CTRLCF ; DID WE GET HERE BY CTRL.C FROM TEST?
 BEQ 24 ; CTRL.C FROM TEST? NO, SKIP BUS RESET.
 BRESET ; YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.

 TRAP CIRESET

 24:

 EXIT CLN

 TRAP CIEXIT

 .WORD L10014 .

 .EVEN

 ENDCLN

 L10014: TRAP CI$CLEAN
SBTLT DROP UNIT SECTION

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

BGN DU

LIDU: :

PRINTF @DROP, RO ; REPORT UNIT THAT HAS BEEN DROPPED.

MOV RO, -(SP)

MOV @DROP, -(SP)

MOV 02, (SP)

MOV SP, RO

TRAP CIDU

ADD 06, SP

BR EDROP ; BRANCH AROUND THE MESSAGE.

DROP: .ASCIZ "A UNIT#D6#A DROPPED FROM FURTHER TESTING.\n"

12437 125 116 111

12442 124 045 104

12445 066 045 101

12450 040 104 122

12453 117 120 120

12456 105 104 040

12461 106 122 117

12464 115 040 106

12467 125 122 124

12472 110 105 122

12475 040 124 105

12500 123 124 111

12503 116 107 056

12506 045 116 000

EVEN

EDROP:

EXIT DU

.SHORT J1JMP

.SHORT L10015-2-

FNDDU

L10015:

TRAP CIDU
**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

EXIT AU

.EVEN

ENDAU

L10016:

 TRAP C1AU
`SBTIL, HARDWARE TEST`  
- ADRA

**-- REGISTER ADDRESS TEST --**

**-- THIS TEST VERIFIES THAT THE DEVICE REGISTERS WILL RESPOND TO THE PROPER --**
**-- UNIBUS HANDSHAKING SIGNALS WHEN ACCESSED. IF THE DHU11 DOES NOT RESPOND --**
**-- TO THE ADRESS ATTEMPTS IF THE DHU11 IS AT THE WRONG ADDRESS, FOR EXAMPLE --**
**-- THE 004 BUS TIME-OUT TRAP IS DETECTED BY THIS ROUTINE AND AN ERROR --**
**-- IS REPORTED. THIS TEST IS PERFORMED ON LINE 0 ONLY. --**

---

```
BGNXT
012526 012526 000001 000001 167472
012526 012526 012767 000001 167472
012534 012767 177777 167460
012542 012767 000145 171242
012550 012767 004632 171236
012556 012767 006200 171232

TNUM = TNUM + 1
INCREMNET THE ASSEMBLY TIME TEST COUNTER.
MOV @TNUM,TSTNUM
SET UP THE TEST NUMBER.
MOV @-1,CTRLCF
INDICATE THAT WE ARE IN A TEST.
MOV #101,EERRNER
SET THE TEST ERROR NUMBER IN THE TABLE.
MOV #00103,EERRMSG
SET UP THE TEST FAILURE MESSAGE IN THE TABLE.
MOV #00101,EERRBLK
SET UP THE ERROR ROUTINE IN THE ERROR TABLE.

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

HERE BEGINS THE LOOP TO TEST THE REGISTERS FOR A LINE.
FIRST TEST THE CSR AND SET THE IND.ADR.REG (I.A.R) FIELD.

MOV CSRA,R0
SET UP CSR AS THE CKTRAP MOVE SOURCE.
MOV #521,R1
SET UP DESTINATION LOCATION FOR CKTRAP MOVE.

JMP PC,CKTRAP
MOVE AND CHECK FOR TRAP.

BCS 41
IF NO TRAP, BYPASS ERROR.

BIS #100001,R5
SET FATAL READ ERROR FLAGS.

BIC #17,521
CLEAR THE I.A.R FIELD OF THE CSR DATA.

MOV R1,R0
USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.

MOV CSRA,R1
SET UP CSR AS THE CKTRAP MOVE DESTINATION.

JMP PC,CKTRAP
MOVE AND CHECK FOR TRAP.

BCS 61
IF NO TRAP, BYPASS ERROR.

BIS #100002,R5
SET FATAL WRITE ERROR FLAGS.

BR 401
EXIT AND REPORT FATAL ERROR.

NOW, WE TEST EACH REGISTER FOR THIS LINE.

MOV #8,R2
INIT REGISTER COUNTER TO 8.

MOV CSRA,501
INITIALIZE THE REGISTER POINTER.

MOV #521,R1
SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.

JMP PC,CKTRAP
PERFORM THE MOVE, CHECK FOR TRAP.

BCS 101
IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.

BIS #100001,R5
SET FATAL READ ERROR FLAGS.

MOV R1,R0
USE OLD DEST AS SRC FOR CKTRAP MOVE.

BR 501
EXIT AND REPORT FATAL ERROR.

```

---

"3544 3545 3546 3547 3548 3549 3550 3551 3552 3553 3554 3555 3556 3557 3558 3559 3560 3561 3562 3563 3564 3565 3566 3567 3568 3569 3570 3571 3572 3573 3574 3575 3576 3577 3578 3579 3580 3581 3582 3583 3584 3585 3586 3587 3588 3589 3590 3591 3592 3593 3594 3595 3596 3597 3598 3599 3600"
3600 012716 004767 175054 JSR PC.CKTRAP ;PERFORM THE MOVE, CHECK FOR TRAP.
3601 012722 103402 BCS 12$ ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
3602 012724 052705 100002 BIS #100002,R5 ;SET FATAL WRITE ERROR FLAGS.
3603 012730 005267 000042 INC 50$ ;INCREmENT THE REGISTER
3604 012734 005267 000036 INC 50$ ;PRINTER BY 2.
3605 012740 005302 DEC R2 ;COUNT THE REGISTER.
3606 012742 001351 BNE 8$ ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
3607
3608
3609 ;DONE CHECKING DEVICE REGISTER ADDRESSES.
3610 ;REPORT ANY ERRORS AND EXIT.
3611
3612
3613 012744 016767 167300 165032 MOV 1P4VEC,4 ;RESTORE THE NORMAL 004 TRAP VECTOR.
3614 012752 005705 TST R5 ;CHECK THE ERROR FLAGS.
3615 012754 100012 BPL 60$ ;EXIT ROUTINE IF NO ERRORS.
3616
3617 ;REPORT "DEVICE REGISTER ACCESS TEST FAILED"
3618
3619 012756 012756 104460 ERROR
3620
3621
3622 012760 016700 167212 DODU UNIIN ;DROP THIS UNIT FROM FURTHER TESTING.
3623 012764 104451 MOV UNIIN,RO
3624 012766 009607 167230 TRAP CIOODU
3625 012766 005067 167230 CLR CTRLCF ;INDICATE NO CTRL-C ABORT FROM TEST.
3626 012772 104444 DOCLN ;ABORT THIS SUB PASS.
3627 012774 000402 BR 60$ ;
3628
3629
3630
3631
3632 012776 000000 501: ;WORD 0 ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
3633 013000 000000 52$: ;WORD 0 ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
3634
3635
3636
3637
3638 012776 000000 501: ;END OF LOCAL STORAGE
3639
3640
3641
3642
3643 013002 005067 167214 60$: ;CLR CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.
3644 013000 000000 ENDST
3645 013006 104401 L10017: TRAP CIEST
SBTL HARDWARE TEST - DTPMC

DATA TERMINAL READY MODE CONTROL SIGNAL TEST

THIS TEST VERIFIES THAT THE DTR MODE 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

T2:

; ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.

BIT SBTL,LOPBACK ; CHECK TYPE OF LOOPBACK MODE SELECTED.
BNE 21
JMP 691 ; EXIT THIS TEST IF IN INTERNAL LOOPBACK.

21: SETPRI #PRI05 ; ALLOW LTC INTERRUPTS.

MOV #PRI05,RO
TRAP C1SPRI

TNUM = TNUM + 1 ; INCREMENT THE ASSEMBLY TIME TEST COUNTER.

MOV #TNUM,TSTNUM ; SET UP THE TEST NUMBER.

MOV #1,CTRLCF ; INDICATE THAT WE ARE IN A TEST.

MOV #1,ERRTY ; SET ERROR TYPE IN ERROR TABLE.

MOV #TSB01,ERRMRS ; SET THE FIRST ERROR NUMBER IN ERROR TABLE.

MOV #EH7B01,ERRMSG ; SET ERROR MESSAGE ADDRESS IN ERROR TABLE.

RESET THE DUT TO AN UNKNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.

; CLEAR TX AND RX INTERRUPT ENABLE BITS.

; THIS SUBROUTINE REPORTS ERROR > > > 7801 << << <<.

JSA PC,CLNARST ; RESET THE DUT.

JMP 691 ; ABORT THE TEST IF FATAL ERROR FOUND IN CLEAR.

SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.

JSR PC,ASLNL01 ; SET UP THE ASSOCIATED LINE TABLES.

; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.

; THIS LOOP CLEAR ALL THE DTR'S AND THEN SETS THEM INDIVIDUALLY AND WAITS FOR
; A RESPONSE ON THE ASSOCIATED RI AN' TSB SIGNALS.

; THIS LOOP WILL CLEAR THE TX.IE AND TX.IE BITS IF THEY ARE NOT EXIST.

CLR R3 ; CLEAR THE LINE COUNTER.

MOV R3,RO

ASL RO

BIT BITBL(RO),ACTLNS

BEQ 121 ; DON'T TEST IF NOT ACTIVE LINE.

CLR R0 ; CLEAR ALL THE DUT LNRCTL REGISTERS DTR BITS.

MOV #MAPLNS,R5 ; SPECIFY THAT ALN LNRCTL BITS TO BE CLEARED.

MOV #MAPLNS,R5 ; SPECIFY THAT ALN LNRCTL BITS TO BE CHANGED.
; CLEAR ALL THE DUT DIR BITS.
JSR PC.WTLWC
; DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.
JSR PC.DELAY

; CHECK THAT AT LEAST ONE OF ASSOCIATED DSR OR RI IS CLEAR AND RECORD STATES.
; GET THE ASSOCIATED LINE NUMBER.
MOVB TXRNLN(BR3), R4
; SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
MOV R4, BCSRA
; SET THE STATE OF THE ASSOCIATED DSR, RI BITS.
MOV @BIT5+BIT13.R0
; CHECK FOR BOTH DSR AND RI SET.
BIC R5.R0
BEQ 10:
; GO REPORT DTR IS BAD IF BOTH ARE SET.

; SET THE DTR FOR THE SELECTED LINE AND WAIT FOR EITHER DSR OR RI TO SET.
; SELECT THE SELECTED LINE IND.ADR.REG FIELD.
MOV R3, BCSRA
; SET THE SELECTED LINE DTR.
BIS @BIT0+BITNTA
; SPECIFY TO WAIT UP TO 60 MS FOR RI TO SET.
MOV @BIT0+BITA8.R1
; CHECK PREVIOUS STATE OF DSR BIT.
BIT @BIT5.R5
BNE 8:
; GO USE RI IF DSR BIT WAS NOT CLEAR.
MOV $00, R4
; WAIT UP TO 60 MS FOR SIGNAL TO GO SET.
BNE 8:
; SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.
MOV @FSLCSA.R0
; GET THE STATUS REGISTER CONTENTS.
MOV @BIT7+BIT7.R7
; REMOVE ALL BUT THE DSR AND RI BITS.
BIC R5.R0
; TEST FOR SIGNAL ONCE CLEAR, BUT NOW SET.
BNE 12:
; GO LOOP IF SIGNAL HAS GONE FROM CLR TO SET.
MOV #7802, ERRNR
; SELECT THE ERROR NUMBER.
MOVB ERRBLK, #7801
; SELECT THE ERROR PRINT ROUTINE.
MOV @EM7802.R1
; SELECT THE ERROR MESSAGE.

; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED.
; EXIT WITH TEST FAILURE MESSAGE IF.
BIT @BIT0+BITA6.OPTION
BEQ 60:
; NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
; DURING THE SOFTWARE QUESTIONS.
INC R3
; SELECT THE NEXT LINE NUMBER.
BIR R4, #NUMLN
; TEST FOR ALL LINES DONE.
BLT 61:
; LOOP IF NOT ALL LINES DONE.

; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
; THIS LOOP SETS ALL THE DTR'S AND THEN CLEARS 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.
; CLEAR THE LINE COUNTER.
CLR R3
; DON'T TEST IF NOT ACTIVE LINE.
BEQ 20:
; SET THE DUT LNTCL REGISTERS DTR BITS.

3745  01334  012700  001000  MOV  @BIT9,RO  \(\text{Specify that DTR bits are to be set.}\)
3747  013346  012705  177777  MOV  @APLNS,R5  \(\text{Specify that all LNCRLS are to be changed.}\)
3748  013347  004767  176060  JSR  PC,WTLNC  \(\text{Set all the DUT DTR bits.}\)
3749  013348  012704  000074  MOV  @R8,RO  \(\text{Check 60 ms to allow signals to settle.}\)
3750  013355  004767  174562  JSR  PC,DELAY  \(\text{Check that at least one associated DSR or RI is set and record states.}\)
3751  013360  116304  003750  MOVB  TXRLNB(R3),R4  \(\text{Get the associated line number.}\)
3752  013364  010477  166610  MOV  R4,BCSRA  \(\text{Select associated line ind.ADR. REG field.}\)
3753  013367  017705  166616  MOV  @FSLSA,R5  \(\text{Get the state of the associated DSR, RI bits.}\)
3754  013370  012701  010500  MOV  R5,RO  \(\text{Check for both DSR and RI clear.}\)
3755  013371  042700  057777  BIC  #57777,RO  \(\text{Go report DTR is bad if both are clear.}\)
3756  013372  001431  \(\text{Clear the DTR for the selected line and wait for either DSR or RI to clear.}\)
3757  013340  010377  166570  MOV  R3,BCSRA  \(\text{Select the selected line ind.ADR. REG field.}\)
3758  013341  042777  001000  166572  BIC  #BIT9,9,INCRA  \(\text{Clear the selected line DTR.}\)
3759  013344  012701  150074  MOV  @150074,R1  \(\text{Specify to wait up to 60 ms for RI to clear.}\)
3760  013342  032705  100000  BIT  #BIT5,RO  \(\text{Check previous state of DSR bit.}\)
3761  013346  001402  BEQ  168  \(\text{Go use RI if DSR bit was not set.}\)
3762  013343  012701  170074  MOV  @170074,R1  \(\text{Specify to wait up to 60 ms for DSR clear.}\)
3763  013344  016702  166546  MOV  FSLSA,R2  \(\text{Specify to look in stat reg for bit to CLR.}\)
3764  013340  010477  166534  MOV  R4,BCSRA  \(\text{Select associated line ind.ADR. REG field.}\)
3765  013344  004767  175630  JSR  PC,WAIBIC  \(\text{Wait up to 60 ms for signal to go clear.}\)
3766  013345  103423  BCS  204  \(\text{Select next line and loop if signal is clear.}\)
3767  013342  017700  166530  MOV  @FSLSA,R0  \(\text{Get the status register contents.}\)
3768  013345  042705  057777  BIC  #57777,RO  \(\text{Test for signal once set, but now clear.}\)
3769  013346  040005  BIC  RO,R5  \(\text{Go loop if signal has gone from set to clr.}\)
3770  013346  001015  BNE  204  \(\text{Report DTR mode control signal defective on line nn.}\)
3771  013346  012767  017192  170316  MOV  #7803,ERRNR  \(\text{Select the error number.}\)
3772  013347  012767  166612  170314  MOV  @ERR7801,EMBLK  \(\text{Select the error print routine.}\)
3773  013350  012701  005015  MOV  @EM7802,R1  \(\text{Select the error message.}\)
3774  013350  104460  \(\text{Exit the test if extended error reporting has not been enabled.}\)
3775  013351  052767  000100  166444  BIT  #BIT06,OPTION  \(\text{Exit with test failure message if no extended error reporting has been requested.}\)
3776  013351  001404  BEQ  608  \(\text{During the software questions.}\)
3777  013352  005203  \(\text{Select the next line number.}\)
3778  013352  020327  000020  CMP  R3,0,NUMLN5  \(\text{Test for all lines done.}\)
3779  013352  002674  BLT  148  \(\text{Loop if not all lines done.}\)
3780  013353  005067  166466  CLR  CTLRCF  \(\text{Indicate that we are not within a test.}\)
3781  013353  012700  00340  SETPRI #PRI07  \(\text{Disable all interrupts.}\)
3782  013354  104441  MOV  @PRIORITY,RO  \(\text{Trap CIESPP.}\)
3783  013354  104401  ENDTST  \(\text{L10020:}\)
3784  013354  104401  L10020:  \(\text{Trap CIESPP.}\)
SBTIL HARDWARE TEST - RTSMS

- 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 2S PIN OR STAGGERED LOOPBACK IS SPECIFIED. THIS TEST USES THE LOOPED BACK SIGNALS CTS AND DCD TO TEST THE RTS SIGNAL. THIS TEST IS PERFORMED ON ALL ACTIVE LINES.

BGNTST

T3:

* ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.

BIT #BIT1,LOOPBK ;CHECK TYPE OF LOOPBACK MODE SELECTED.
BNE 18
JMP 60$ ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.

SETPRI #PRIOS ;ALLOW LTC INTERRUPTS.
MOV #PRIOS,RO
TRAP CISPRI

TNUM = TNUM + 1 ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
MOV @TNUM,TNUM ;SET UP THE TEST NUMBER. (79)
MOV #T1,CTRCLF ;INDICATE THAT WE ARE IN A TEST.
MOV 0:1,EEATYP ;SET ERROR TYPE IN ERROR TABLE.
MOV #7901,ERRNBR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
MOV #7901,ERMSG ;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.
BCC 31
JMP 60$ ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.

SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.

JSA PC,ASNLTL ;SET UP THE ASSOCIATED LINE TABLES.

SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
THIS LOOP CLEARS ALL THE RSS 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 A3 ;CLEAR THE LINE COUNTER.
MOV R3,RO
ASL RO
BIT BITBTL(RO),ACIUTS
BEQ 8$ ;DON'T TEST IF NOT ACTIVE LINE.

CLEAR ALL THE DUT LINCTRL REGISTERS RTS BITS.
CLR RO ;SPECIFY THAT ALL LINCTRL BITS TO BE CLEARED.
MOV #APLNS,R5 ;SPECIFY THAT ALL LINCTRLS ARE TO BE CHANGED.
JSR PC,NTMLNC ;CLEAR ALL THE DUT RTS BITS.
MOV #60,R4
JSR PC,DELAY ;DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.

; CHECK THAT AT LEAST ONE OF ASSOCIATED DCD OR CTS IS CLEAR AND RECORD STATES.

MOVB.TXRLNB(R3),R4 ;GET THE ASSOCIATED LINE NUMBER.
MOV R4,ACSRA ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
MOV #FSLSA,R5 ;GET THE STATE OF THE ASSOCIATED DCD, CTS BIT.
MOV #BIT12,BIT11,R0 ;CHECK FOR BOTH DCD AND CTS SET.
BEQ 6$ ;GO REPORT RTS IS BAD IF BOTH ARE SET.

6$: ;SET THE RTS FOR THE SELECTED LINE AND WAIT FOR EITHER DCD OR CTS TO SET.

MOV R3,ACSRA ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
BIT #BIT12,BNCLR1A ;SET THE SELECTED LINE RTS.
MOV #13007,R1 ;SPECIFY TO WAIT UP TO 60 MS FOR CTS TO SET.
BIT #BIT12,R5 ;CHECK PREVIOUS STATE OF DCD BIT.
BEQ 4$ ;GO USE CTS IF DCD BIT WAS NOT CLEAR.
MOV #14007A,R1 ;SPECIFY TO WAIT UP TO 60 MS FOR DCD SET.
MOV FSLSA,R2 ;SPECIFY TO LOOK IN STAT REG FOR BIT TO SET.
MOV R4,ACSRA ;SELECT ASSOCIATED LINE IND.ADR.REG FIELD.
MOV #FSLSA,R0 ;GET THE STATUS REGISTER CONTENTS.
MOV #BIT12,BIT11,R0 ;WAIT UP TO 60 MS FOR SIGNAL TO SET.
BNE 4$ ;SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.
MOV #BLSA,R0 ;SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.
BNE 6$ ;SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.
MOV #BIT12,R5 ;SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.
BNE 4$ ;SELECT NEXT LINE AND LOOP IF SIGNAL IS SET.

4$: ;REPORT AT MODERN CONTROL SIGNAL DEFECTIVE ON LINE NN.
MOV #7902, ERNRDR ;SELECT THE ERROR NUMBER.
MOV #ER7803, ERRBLK ;SELECT THE ERROR PRINT ROUTINE.
MOV #ER7902, R1 ;SELECT THE ERROR MESSAGE.
ERROR 6$ ;ERROR <<<<><<.<< TRAP CERROR

EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED.

BIT #BIT06, OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
BEQ 60$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED.

INC R3 ;SELECT THE NEXT LINE NUMBER.
CMP R3, #NUMNS ;TEST FOR ALL LINES DONE.
BLT 2$ ;LOOP IF NOT ALL LINES DONE.

SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.

THIS LOOP SETS ALL THE RTS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR A RESPONSE ON THE ASSOCIATED CTS AND DCD SIGNALS.

THIS LOOP WILL CLEAR THE TXIE AND RXIE BITS IF THEY ARE SET.

CLR R3 ;CLEAR THE LINE COUNTER.
MOV R5, RO ;DON'T TEST IF NOT ACTIVE LINE.
ASL R0
BIT BITBL(R0), ACTLS
BEQ 16$ ;DON'T TEST IF NOT ACTIVE LINE.

SET ALL THE DUT LNCRL REGISTERS RTS BITS.
MOV @BIT12, R0 ; SPECIFY THAT RTS BITS ARE TO BE SET.
MOV #MAPLNS, R5 ; SPECIFY THAT ALL LMCTLs ARE TO BE CHANGED.
JSR PC, WTHLC  ; SET ALL OUT RTS BITS.
JSR PC, DELAY  ; DELAY FOR 60 MS TO ALLOW SIGNALS TO SETTLE.

; CHECK THAT AT LEAST ONE OF ASSOCIATED DCD OR CTS IS SET AND RECORD STATES.
MOV B, TXRNLNB(R3), R4 ; GET THE ASSOCIATED LINE NUMBER.
MOV R4, ACRSA ; SELECT ASSOCIATED LINE IN ADR, REG FIELD.
MOV #FLSA, R5 ; GET THE STATE OF THE ASSOCIATED DCD, CTS BITS.
MOV R5, R0
BIC @16377, R0 ; CHECK FOR BOTH DCD AND CTS CLEAR.
BEQ 14$ ; GO REPORT RTS IS BAD IF BOTH ARE CLEAR.

; CLEAR THE RTS FOR THE SELECTED LINE AND WAIT FOR EITHER DCD OR CTS TO CLEAR.
MOV R3, ACRSA ; SELECT THE SELECTED LINE IN ADR, REG FIELD.
BIC @BIT12, @NCTR A ; CLEAR THE SELECTED LINE RTS.
MOV @130074, R1 ; SPECIFY TO WAIT UP TO 60 MS FOR CTS TO CLEAR.
BIT @BIT12, R5 ; CHECK PREVIOUS STATE OF DCD BIT.
BEQ 12$ ; GO USE CTS IF DCD BIT WAS NOT SET.

MOV @140074, R1 ; SPECIFY TO WAIT UP TO 60 MS FOR DCD CLEAR.
MOV #FLSA, R2 ; SELECT TO LOOK IN STATUS REG FOR BIT TO CLR.
MOV R4, ACRSA ; SELECT ASSOCIATED LINE IN ADR, REG FIELD.
JSR PC, WABIC ; WAIT UP TO 60 MS FOR SIGNAL TO GON CLEAR.
BCS 16$ ; SELECT NEXT LINE AND LOOP IF SIGNAL IS CLEAR.
MOV #FLSA, R0
BIC @16377, R5
BNE 16$ ; TEST FOR SIGNAL ONCE SET, BUT NOW CLEAR.
BEQ 10$ ; GO LOOP IF SIGNAL HAS GONE FROM SET TO CLR.

REPORT RTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
MOV #17903, ERRNR ; SELECT THE ERROR NUMBER.
MOV #17801, ERRBLK ; SELECT THE ERROR PRINT ROUTINE.
MOV #EM7902, R1 ; SELECT THE ERROR MESSAGE.
ERROR 14$ ; REPORT ERRORS >>> ERROR <<<. TRAP CLERROR

EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

BIT #BIT06, OPTION ; EXIT WITH TEST FAILURE MESSAGE IF
BEQ 60$ ; NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED

DURING THE SOFTWARE QUESTIONS.

INC R3 ; SELECT THE NEXT LINE NUMBER.
CMP R3, #0UMLSN ; TEST FOR ALL LINES DONE.
BL 10$ ; LOOP IF NOT ALL LINES DONE.

CLRA CTRLCF ; INDICATE THAT WE ARE NOT WITHIN A TEST.
SETPRI #PRI07 ; DISABLE ALL INTERRUPTS.
MOV #P R I0, R0
TRAP CLSPRI

ENDST

L10021: TRAP CLINT
**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**

`14:$`

Only perform this test if the DUT is in external or staggered loopback mode.

BIT `@BIT1,LOOPBACK`; CHECK TYPE OF LOOPBACK MODE SELECTED.

BNE `2:`; CHECK TYPE OF LOOPBACK MODE.

JMP `6O:`; EXIT THIS TEST IF IN INTERNAL LOOPBACK.

SET PRI `@PRI05`; ALLOW LTC INTERRUPTS.

MOV `@PRI05,RO`; TRAP `C:SPRI`.

**16:$**

TNUM `= = TNUM + 1`; INCREMENT THE ASSEMBLY TIME TEST COUNTER.

MOV `@TNUM,TSTNUM`; SET UP THE TEST NUMBER.

MOV `#1,LTALCF`; INDICATE THAT WE ARE IN A TEST.

MOV `#1,ERRTPY`; SET ERROR TYPE IN ERROR TABLE.

MOV `#8001,ERANBR`; SET THE FIRST ERROR NUMBER IN ERROR TABLE.

MOV `#8001,ERRMSG`; SET ERROR MESSAGE ADDRESS IN ERROR TABLE.

**18:$**

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 <<<<<<.

**20:$**

JSR `PC,CLWST`; RESET THE DUT.

BCS `4:`; ABORT THE TEST IF FATAL ERROR FOUND IN RESET.

JMP `6O:`; ABORT THE TEST IF FATAL ERROR FOUND IN RESET.

**24:$**

SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.

**28:$**

JSR `PC,ASNLTL`; SET UP THE ASSOCIATED LINE TABLES.

**30:$**

SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.

THIS LOOP CLEARS ALL THE DIRS 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,RO`; CLEAR THE LINE.

ASL `RO`; CLEAR THE LINE.

BIT `BITB(RO),ACLNS`; DON'T TEST IF NOT ACTIVE LINE.

**36:$**

CLR `RO`; SPECIFY THAT ALL LCNTLR BITS TO BE CLEARED.

MOV `@MAPLNS,R5`; SPECIFY THAT ALL LCNTLRs ARE TO BE CHANGED.

**38:$**

CLEAR ALL THE DUT LCNTLR REGISTER DTR BITS.

CLR `R0`; CLEAR THE DUT LCNTLR REGISTER DIR BITS.
K8

4008 014422 004767 175002 JSR PC,W1NLNC ;CLEAR ALL THE DUT DTR BITS.
4009 014426 012704 000050 MOV @00,R4 ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4010 014432 004767 173504 JSR PC,DELAY
4011 ;CHECK THAT THE SPECIFIED DSR IS CLEAR.
4012
4013 014436 010377 165536 MOV R5,BCSRA
4014 014442 032777 100000 165536 SET INO,ADR,REG FIELD TO SELECTED LINE.
4015 014450 001020 BIT #BIT15,BFLSLA
4016 BNE #4 ;GO REPORT DSR IS BAD IF BIT IS NOT CLEAR.
4017 ;SET THE DTR FOR THE ASSOCIATED LINE.
4018
4019 NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED
4020 IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
4021
4022 014452 116304 003750 MVB TXXLNB(R3),R4 ;GET THE ASSOCIATED LINE NUMBER.
4023 014456 010477 165516 MVB R4,BCSRA ;SET INO,ADR,REG FIELD TO ASSOCIATED LINE.
4024 014466 052777 001000 165520 BIS #BIT9,BCLKTR
4025 ;SET THE ASSOCIATED LINE DTR.
4026
4027 ;CHECK THAT THE SELECTED LINE DSR IS ACTIVE.
4028
4029 014470 010377 165504 MOV R3,BCSRA
4030 014474 012701 170050 MOV @170050,R1 ;PASS TIMEOUT OF 40 MILI-SEC, AND BIT TO TEST.
4031 014500 016702 165502 MOV FLSLA,R2 ;PASS THE ADDRESS OF THE REGISTER TO TEST.
4032 014504 004767 174644 JSR PC,W1BIS ;WAIT FOR DSR TO BECOME SET OR TIMEOUT.
4033 014510 103415 BCS 101 ;SKIP ERROR REPORT IF SELECTED DSR IS SET.
4034
4035 014512 88 ;REPORT DSR MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
4036 014512 12767 017502 167272 MOV #8002,EARNR ;SELECT THE ERROR NUMBER.
4037 014520 012767 006662 167270 MOV @#7801,ERUBLK ;SELECT THE ERROR PRINT ROUTINE.
4038 014526 012701 005163 MOV #EM8002,R1 ;SELECT THE ERROR MESSAGE.
4039 014532 ERROR TRAP C1ERROR
4040
4041 ;EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4042
4043
4044 014534 032767 000100 165420 BIT #BIT06,OPTION
4045 014542 001474 EXITH WITH TEST FAILURE MESSAGE IF
4046
4047
4048 014544 005203 BEQ 60 ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4049 014546 020327 000020 ;DURING THE SOFTWARE QUESTIONS.
4050 014552 002712
4051
4052 SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
4053 ;THIS LOOP SETS ALL THE DTR'S AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
4054 A RESPONSE ON THE SELECTED DSR SIGNAL.
4055 ;THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
4056
4057 014554 005003 CLR R3 ;CLEAR THE LINE COUNTER.
4058 014556 010300 MOV R3,RO
4059 014560 006300 ASL R0
4060 014562 036067 002310 165402 BIT BITBLC(R0),ACTLNS
4061 014570 001455 BEQ 164 ;DON'T TEST IF NOT ACTIVE LINE.
4062
4063 SET ALL THE DUT LNMCTRL REGISTERS DTR BITS.
4064  014572  012700  001000
4066  014576  012705  177777
4067  014602  004767  174622
4068  014606  012704  000050
4069  014612  004767  173324
4070
4071
4072
4073  014616  010377  165356
4074  014622  032777  100000  165356
4075  014630  001420
4076
4077
4078
4079
4080
4081  014632  116304  003750
4082  014636  010477  165336
4083  014642  042777  001000  165340
4084
4085
4086
4087  014650  010377  165324
4088  014654  012701  170050
4089  014660  016702  163322
4090  014664  004767  174410
4091  014670  103415
4092
4093  014672
4094  014672  012767  017503  167112
4095  014700  012767  006662  167110
4096  014706  012701  005163
4097  014712  014712  104460
4098
4099
4100
4101
4102  014714  032767  000100  165240
4103  014722  001404
4104
4105
4106  014724  005203
4107  014726  020327  000020
4108  014732  002711
4109
4110  014734  005067  165262
4111  014740  012700  000340
4112
4113  014746  014746  014746  0104401

;MOV @BIT9,R0
;SPECIFY THAT DTR BITS ARE TO BE SET.
;MOV @MAPLS,R5
;SPECIFY THAT ALL LNCRLS ARE TO BE CHANGED.
;JSR PC.MTHLNC
;SET ALL THE OUT DTR BITS.
;JSR PC.DELAY
;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.

;CHECK THAT THE SPECIFIED DSR IS SET.

;MOV R3,BCSRA
;SET IND.ADR.REG FIELD TO SELECTED LINE.
;BEQ 14:
;GO REPORT DSR IS BAD IF BIT IS NOT SET.

;CLEAR THE DTR FOR THE ASSOCIATED LINE.
;NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED
;IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).

;MOV @XRLNB(R3),R4
;GET THE ASSOCIATED LINE NUMBER.
;MOV R4,BCSRA
;SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
;BIC @BIT9,QNCTRA
;CLEAR THE ASSOCIATED LINE DTR.

;CHECK THAT THE SELECTED LINE DSR IS CLEAR.

;MOV R3,BCSRA
;SET IND.ADR.REG FIELD TO SELECTED LINE.
;MOV @170050,R1
;PASS TIMEOUT OF 40 MILLI-SEC, AND BIT TO TEST.
;MOV FSLSA,R2
;PASS THE ADDRESS OF THE REGISTER TO TEST.
;JSR PC.WAIBIC
;WAIT FOR DSR TO BECOME CLEAR OR TIMEOUT.
;BCS 16:
;SKIP ERROR REPORT IF SELECTED DSR IS CLEAR.

;REPORT DSR MODEM CONTROL SIGNAL DEFECTIVE ON LINE N.

;MOV @ERR03,ERRNR
;SELECT THE ERROR NUMBER.
;MOV @ERR7801,ERRBLK
;SELECT THE ERROR PRINT ROUTINE.
;MOV @ERR002,R1
;SELECT THE ERROR MESSAGE.

;EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

;BIT @BIT06,OPTION
;EXIT WITH TEST FAILURE MESSAGE IF
;BEQ 601:
;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
;DURING THE SOFTWARE QUESTIONS.

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

;CLR CTRLDF
;INDICATE THAT WE ARE NOT WITHIN A TEST.
;SETPRI @PRI07
;DISABLE ALL INTERRUPTS.

;MOV @PRI07,RO
;TRAP C1SPRI

;ENDST
.SBITL 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.

:----------------------------------

BGNST

T5:

: ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.

: BIT @B1,LOPBCK ;CHECK TYPE OF LOOPBACK MODE SELECTED.
: BNE 2$ ;EXIT THIS TEST IF IN INTERNAL LOOPBACK.
: JMP 60+ ;ALLOW LTC INTERRUPTS.

2$:

: SETPRI @PRI05,RO ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
: MOV @NUM,TSTNUM ;SET UP THE TEST NUMBER.
: MOV #1,CTRLCF ;INDICATE THAT WE ARE IN A TEST.
: MOV #1,ERTYPE ;SET ERROR TYPE IN ERROR TABLE.
: MOV #8101,ERRNR ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
: MOV @EB8101,EMMSG ;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,CLRTST ;RESET THE DUT.
: BCS 4$ ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.

: SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.

: JSR PC,ASNLTL ;SET UP THE ASSOCIATED LINE TABLES.

: SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
: THIS LOOP CLEARS ALL THE DTR'S 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.

: MOV R3,RO
: ASL RO
: BIT BITBL(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 CLEAR.
: MOV #MAPLNS,R5 ;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
;  ; CLEAR ALL THE DUT DTR BITS.
;  ; DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
;  ; CHECK THAT THE SPECIFIED RI IS CLEAR.
;  ; SET THE DTR FOR THE ASSOCIATED LINE.
;  ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED
;  ; IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
;  ; GET THE ASSOCIATED LINE NUMBER.
;  ; SET IN.DR.REG FIELD TO ASSOCIATED LNF.
;  ; SET THE ASSOCIATED LINE DTR.
;  ; CHECK THAT THE SELECTED LINE RI IS ACTIVE.
;  ; REPORT RI MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
;  ; SELECT THE ERROR NUMBER.
;  ; SELECT THE ERROR PRINT ROUTINE.
;  ; SELECT THE ERROR MESSAGE.
;  ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
;  ; EXIT WITH TEST FAILURE MESSAGE IF
;  ; NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
;  ; DURING THE SOFTWARE QUESTIONS.
;  ; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
;  ; THIS LOOP SETS ALL THE DTS AND THEN CLEARS THEM INDIVIDUALLY AND WAITS FOR
;  ; A RESPONSE ON THE SELECTED RI SIGNAL.
;  ; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.
;  ; CLEAR THE LINE COUNTER.
;  ; DON'T TEST IF NOT ACTIVE LINE.
;  ; SET ALL THE DUT LNTAL REGISTERS DTR BITS.


4225  MOV  @BIT9,R0  ; Specify that DTR bits are to be set.
4226  MOV  @MAPLSN,R5  ; Specify that all nclnts are to be changed.
4227  JSR  PC,MMCN  ; Set all the OUT DTR bits.
4228  MOV  &40,R4  ; Delay for 40 ms to allow signals to settle.
4229  JSR  PC,DELAY

4230  CHECK THAT THE SPECIFIED RI IS SET.

4231  MOV  R3,BCSRA  ; Set ind.adr.reg field to selected line.
4232  BIT  @BIT13,BSLSA  ; Go report RI is bad if bit is not set.
4233  BEQ  144

4234  CLEAR THE DTR FOR THE ASSOCIATED LINE.

4235  NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, DTR WILL NOT HAVE BEEN TESTED

4236  IN THE DTR TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).

4237  MOVB TXRLNRD(R3),R4  ; Get the associated line number.
4238  MOV  R4,BCSRA  ; Set ind.adr.reg field to associated line.
4239  BIC  @BIT9,BNCTR  ; Clear the associated line dtr.

4240  CHECK THAT THE SELECTED LINE RI IS CLEAR.

4241  MOV  R3,BCSRA  ; Set ind.adr.reg field to selected line.
4242  MOV  @150050,R1  ; Pass timeout of 40 milli-sec. and bit to test.
4243  MOV  FSLSA,R2  ; Pass the address of the register to test.
4244  JSR  PC,NAIBIC  ; Wait for RI to become clear or timeout.
4245  BCS  166  ; Skip error report if selected RI is clear.

4246  REPORT RI MODERN CONTROL SIGNAL DEFECTIVE ON LINE NN.

4247  MOV  @B103,ERROR  ; Select the error number.
4248  MOV  @ET7801,ERRORL  ; Select the error print routine.
4249  MOV  @EM102,R1  ; Select the error message.

4250  TRAP CERROR

4251  EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

4252  BIT  @BIT0,OPTION  ; Exit with test failure message if
4253  BEQ  608  ; No extended error reporting has been requested.

4254  DURING THE SOFTWARE QUESTIONS.

4255  INC  R3  ; Select the next line number.

4256  CMP  R5,#NUMLN5  ; Test for all lines done.
4257  BLT  124  ; Loop if not all lines done.

4258  CLR  CTRLCF  ; Indicate that we are not within a test.

4259  SETPRI @PRI07  ; Disable all interrupts.

4260  MOV  @PRI07,R0  ;

4261  TRAP CISPRI

4262  ENDSTT
SBITL HARDWARE TEST - CTSMS

- CLEAR TO SEND MODEM SIGNAL TEST

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

-----------------------------------------------

BGNST

T6:

: ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.

: BIT @BITI,LOOPBK
: CHECK TYPE OF LOOPBACK MODE SELECTED.
: BNE 2:
: JMP 60:
: EXIT THIS TEST IF IN INTERNAL LOOPBACK.

2:

: SETPRI @PRI05,0
: 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. (02)
: MOV #1,CT1LCF
: INDICATE THAT WE ARE IN A TEST.
: MOV #1,ERRTP
: SET ERROR TYPE IN ERROR TABLE.
: MOV @PM201,ERRMGR
: SET FIRST ERROR NUMBER IN ERROR TABLE.
: MOV @PM201,ERRMSG
: SET ERROR MESSAGE ADDRESS IN ERROR TABLE.

: L1:

: CLEAR TX AND RX INTERRUPT ENABLE BITS.
: THIS SUBROUTINE REPORTS ERROR >> >> A201 << <<.

: JSR PC,CLRST
: RESET THE CTS.
: BCS 4:
: JMP 60:
: ABORT THE TEST IF FATAL ERROR FOUND IN CLEAR.

: SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.

: JSR PC,ASLNL
: SET UP THE ASSOCIATED LINE TABLES.

: SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
: THIS LOOP CLEARLS ALL THE RTS'S AND THEN SETS THEM INDIVIDUALLY AND WAITS FOR
: A RESPONSE ON THE ASSOCIATED CTS SIGNAL.
: THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.

: CLR R3
: CLEAR THE LINE COUNTER.
: MOV R3,RO
: ASL RO
: BIT BITBL(RO),ACTLS
: BEQ 10:
: DON'T TEST IF NOT ACTIVE LINE.

: CLR R0
: CLEAR ALL THE DUT LNCRL REGISTERS RTS BITS.
: MOV @MAPLNS,R5
: SPECIFY THAT ALL LNCRL BITS TO BE CLEARED.
: MOV 05000
: SPECIFY THAT ALL LNCRLS ARE TO BE CHANGED.
HARDWARE TEST

4330 015542 004767 173662  
JSR  PC, WILNC
CLEAR ALL THE DUT RTS BITS.
4331 015546 012704 000050
MOV  #40, R4
4332 015552 004767 172564  
JSR  PC, DELAY
DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4333
4334
4335
4336 015556 010377 164416
MOV  R3, BCSRA
CHECK THAT THE SPECIFIED CTS IS CLEAR.
4337 015562 032777 004000 164416
BIT  #BOIT13, $0FSLSA
NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, RTS WILL NOT HAVE BEEN TESTED
4338 015570 001020
BNE  81
IN THE RTS TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
4339
4340 015572 116304 003750
MOV  TXRNBD(R3), R4
SET THE ASSOCIATED LINE NUMBER.
4341 015576 010477 164376
MOV  R4, BCSRA
SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
4342 015602 052777 010000 164400
BIS  #BOIT12, $0LNCRA
SET THE ASSOCIATED LINE RTS.
4343
4344 015610 010377 164364
MOV  R3, BCSRA
CHECK THAT THE SELECTED LINE RTS IS ACTIVE.
4345 015614 012701 130050
MOV  $130050, R1
PASS THE ADDRESS OF THE REGISTER TO TEST.
4346 015620 016702 164362
MOV  $0FSLSA, R2
PASS THE ADDRESS OF THE REGISTER TO TEST.
4347 015624 004767 173524
JSR  PC, WAIBIS
WAIT FOR CTS TO BECOME SET OR TIMEOUT.
4348 015630 103415
BCS  108
SKIP ERROR REPORT IF SELECTED RTS IS SET.
4349
4350 015632
MOV  R3, BCSRA
REPORT CTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE N.
4351 015636 012767 020012 166152
MOV  #8202, ERRNBR
SELECT THE ERROR NUMBER.
4352 015640 012767 006662 166150
MOV  #87801, ERSlK
SELECT THE ERROR PRINT ROUTINE.
4353 015644 012701 005401
MOV  #EMB202, R1
SELECT THE ERROR MESSAGE.
4354 015648 015652
ERROR

trap  CERROR
4355
4356

8$:
REPORT CTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE N.

4357 015652
MOV  #8202, ERRNBR
SELECT THE ERROR NUMBER.
4358 015656 012767 006662 166150
MOV  #87801, ERSlK
SELECT THE ERROR PRINT ROUTINE.
4359 015660 012701 005401
MOV  #EMB202, R1
SELECT THE ERROR MESSAGE.

4360 015664 005203
ERROR

4361 015668 005203
ERROR

4362

EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

4363

4364

4365

4366 015654 032767 000100 164300
BIT  #BIT06, OPTION
EXIT WITH TEST FAILURE MESSAGE IF
4367 015662 001474
BEQ  608
EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4368

DURING THE SOFTWARE QUESTIONS.

4369

4370 015664 005203
ERROR

4371 015666 005203
ERROR

4372 015667 002712
اما 108:
INC  R5
SELECT THE NEXT LINE NUMBER.
4373 015672 002712
CYC  R3, #NUMLNS
TEST FOR ALL LINES DONE.
4374

4375

4376

4377

4378

4379 015674 005003
CLR  R5
CLEAR THE LINE COUNTER.
4380 015676 010300
MOV  R3, R0
4381 015700 006300
ASL  R0
4382 015702 036067 002310 164262
BIT  BITBL(R0), ACTLNS
4383 015710 001455
BEQ  168
DON'T TEST IF NOT ACTIVE LINE.
4384

4385

SET ALL THE DUT LNCTRAL REGISTERS RTS BITS.
4385 MOV @BIT12,R0 ; SPECIFY THAT RTS BITS ARE TO BE SET.
4388 MOV @MAPLNS,R5 ; SPECIFY THAT ALL LNCRTLS ARE TO BE CHANGED.
4391 JSR PC,WITH NC ; SET ALL THE DUT RTS BITS.
4394 JSR PC,DELAY ; DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4397 ; CHECK THAT THE SPECIFIED CTS IS SET.
4399 MOV R5,BCSRA ; SET IND.ADR.REG FIELD TO SELECTED LINE.
4402 BIT @BIT11,DFLSLA ; GO REPORT CTS IS BAD IF BIT IS NOT SET.
4405 ; CLEAR THE RTS FOR THE ASSOCIATED LINE.
4408 ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, RTS WILL NOT HAVE BEEN TESTED
4411 ; IN THE RTS TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
4414 MOV TXRLCD(R3),R4 ; GET THE ASSOCIATED LINE NUMBER.
4417 MOV R4,BCSRA ; SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
4420 BIC @BIT12,DSNCTRA ; CLEAR THE ASSOCIATED LINE RTS.
4423 ; CHECK THAT THE SELECTED LINE CTS IS CLEAR.
4426 MOV R5,BCSRA ; SET IND.ADR.REG FIELD TO SELECTED LINE.
4429 MOV @130050,R1 ; PASS TIMEOUT OF 40 MILLI-SEC. AND BIT TO TEST.
4432 MOV FLSLA,R2 ; PASS THE ADDRESS OF THE REGISTER TO TEST.
4435 JSR PC,WAIBIC ; WAIT FOR CTS TO BECOME CLEAR OR TIMEOUT.
4438 BCS 164 ; SKIP ERROR REPORT IF SELECTED CTS IS CLEAR.
4441 ; REPORT CTS MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.
4444 MOV @0203,ERRARR ; SELECT THE ERROR NUMBER
4447 MOV @ER7601,ERRBLK ; SELECT THE ERROR PRINT ROUTINE.
4450 MOV @EMB202,R1 ; SELECT THE ERROR MESSAGE.
4453 ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4456 ; EXIT WITH TEST FAILURE MESSAGE IF
4459 BIT @BIT06,OPTION ; NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4462 BEQ 604 ; DURING THE SOFTWARE QUESTIONS.
4465 INC R3 ; SELECT THE NEXT LINE NUMBER.
4468 CMP R3,0,PNUMLS ; TEST FOR ALL LINES DONE.
4471 BLT 124 ; LOOP IF NOT ALL LINES DONE.
4474 CLR CTRLCF ; INDICATE THAT WE ARE NOT WITHIN A TEST.
4477 SETPRI @PRI07 ; DISABLE ALL INTERRUPTS.
4480 MOV @PRI07,R0
4483 TRAP C4SPRI
4486 104460 ENDST
4489 L10024; TRAP C4ETS1
SBTIL  HARDWARE TEST         -  DCDMS

- DATA CARRIER DETECTED MODM 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.

BGNTS!

T7:

* ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.

* BIT #8 (LSEP)
* CHECK TYPE OF LOOPBACK MODE SELECTED.

BNE 2:

JMP 601

* EXIT THIS TEST IF IN INTERNAL LOOPBACK.

2:

SETPRI #PRI05

* ALLOW LTC INTERRUPTS.

MOV #PRI05,RO

TRAP CISPRI

4457  000007

4458  016112  012767  000007  164106

4459  016120  012767  177777  164074

4460  016126  012767  000001  165654

4461  016134  012767  020155  165650

4462  016142  012767  005445  165644

4463:

* RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.

4464  CLEAR TX AND RX INTERRUPT ENABLE BITS.

4466  THIS SUBROUTINE REPORTS ERROR >>>>>> 8301 <<<<<.

4467  JSR PC,CLNRS1

4468  BCS 4:

4469  JMP 601

* ABORT THE DUT ERROR IN RESET.

4470  016150  004767  171652

4471  000000

4472  000167  00342

4473  016156

4474  016162  004767  171254

4475  JSR PC,ASLN1

4476  SET UP THE ASSOCIATED LINE NUMBER TABLE.

4477  I:

4478  JSR PC,ASLN1T

4479  SET UP THE ASSOCIATED LINE TABLES.

4480  016166  05003

4481  016170  010300

4482  016172  006300

4483  016174  036067  002310  163770

4484  016202  001454

4485  CLR R3

4486  MOV R3,RO

4487  ASL R0

4488  BIT BTBL(RO),ACI

4489  BEQ 101

4490  DON'T TEST IF NOT ACTIVE LINE.

4491:

4492  CLR RO

4493  MOV #MAPR, R5

4494  SPECIFY THAT ALL LNCNTL BITS ARE TO BE CHANGED.
; CLEAR ALL THE DUT RTS BITS.
JSR PC,WTWLN

; DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
JSR PC,DELAY

; CHECK THAT THE SPECIFIED DCD IS CLEAR.

; SET IND.ADR.REG FIELD TO SELECTED LINE.
MOV R3,BCSRA

; GO REPORT DCD IS BAD IF BIT IS NOT CLEAR.
BNE 8$

; SET THE RTS FOR THE ASSOCIATED LINE.

; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, RTS WILL NOT HAVE BEEN TESTED

; IN THE RTS TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).

; GET THE ASSOCIATED LINE NUMBER.
MOV B,TXRLNB(R3),R4

; SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
MOV R4,BCSRA

; SET THE ASSOCIATED LINE RTS.
BIS @BIT12,BLNCTRA

; CHECK THAT THE SELECTED LINE DCD IS ACTIVE.

; SET IND.ADR.REG FIELD TO SELECTED LINE.
MOV R3,BCSRA

; PASS TIMEOUT OF 40 MILI-SEC. AND BIT TO TEST.
MOV #1400050.R1

; PASS THE ADDRESS OF THE REGISTER TO TEST.
MOV FSLSA,R2

; WAIT FOR DCD TO BECOME SET OR TIMEOUT.
JSR PC,WAIBIS

; SKIP ERROR REPORT IF SELECTED DCD IS SET.

; REPORT DCD MODEM CONTROL SIGNAL DEFECTIVE ON LINE NN.

; SELECT THE ERROR NUMBER.
MOV #20156,ERMB

; SELECT THE ERROR PRINT ROUTINE.
MOV #ER7801,ERBLK

; SELECT THE ERROR MESSAGE.
MOV #EM3032.R1,ERROR

; TRAP CERROR

; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

; EXIT WITH TEST FAILURE MESSAGE IF
BIT @BIT106,OPTION

; NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
BEQ 6$

; DURING THE SOFTWARE QUESTIONS.

; SELECT THE NEXT LINE NUMBER.
INC R3

; TEST FOR ALL LINES DONE.
CMP R3,#NUMLN

; LOOP IF NOT ALL LINES DONE.
BLT 6$

; SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.

; THIS LOOP SETS ALL THE RTS'S AND THEN CLEAR THEM INDIVIDUALLY AND WAITS FOR
; A RESPONSE ON THE SELECTED DCD SIGNAL.

; THIS LOOP WILL CLEAR THE TX.IE AND RX.IE BITS IF THEY ARE SET.

; CLEAR THE LINE COUNTER.
CLR R3

; DON'T TEST IF NOT ACTIVE LINE.
BIT BITBLC(RO),ACTLNS
BEQ 16$

; SET ALL THE DUT LNCTRL REGISTERS RTS BITS.
4547 MOV @BIT12, R0 ; SPECIFY THAT RTS BITS ARE TO BE SET.
4549 MOV @MAPLSR, R5 ; SPECIFY THAT ALL LNCTLS ARE TO BE CHANGED.
4550 JSR PC, WITLNC ; SET ALL THE BUT RTS BITS.
4551 JSR PC, DELAY ; DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.
4553 ; CHECK THAT THE SPECIFIED DCD IS SET.
4554 MOV R3, @CSRA ; SET IND.ADR.REG FIeld TO SELECTED LINE.
4555 BIT #BIT12, @BFSLSA ; GO REPORT DCD IS BAD IF BIT IS NOT SET.
4556 MOV R4, @BCSRA ; CLEAR THE RTS FOR THE ASSOCIATED LINE.
4557 ; NOTE: IF THE ASSOCIATED LINE IS NOT SELECTED, RTS WILL NOT HAVE BEEN TESTED
4558 ; IN THE RTS TEST (ONLY AN ISSUE IN STAGGERED LOOPBACK).
4559 MOV B, IXLANB(R3), R4 ; GET THE ASSOCIATED LINE NUMBER.
4560 MOV R4, @BCSRA ; SET IND.ADR.REG FIELD TO ASSOCIATED LINE.
4561 BIT #BIT12, @BCNTRA ; CLEAR THE ASSOCIATED LINE RTS.
4562 ; CHECK THAT THE SELECTED LINE DCD IS CLEAR.
4563 MOV R3, @CSRA ; SET IND.ADR.REG FIELD TO SELECTED LINE.
4564 MOV R1, #140050, R1 ; PASS TIMEOUT OF 40 MILI-SEC. AND BIT TO TEST.
4565 MOV R1, #160702, R2 ; PASS THE ADDRESS OF THE REGISTER TO TEST.
4566 JSR PC, WABIC ; WAIT FOR DCD TO BECOME CLEAR OR TIMEOUT.
4567 BCS 161 ; SKIP ERROR REPORT IF SELECTED DCD IS CLEAR.
4568 ; REPORT DCD MODEM CONTROL SIGNAL DEFECTIVE ON LINE NO.
4569 MOV @ER7801, ERRBLK ; SELECT THE ERROR NUMBER.
4570 MOV @ER7801, ERRBLK ; SELECT THE ERROR PRINT ROUTINE.
4571 MOV @EM8302, R1 ; SELECT THE ERROR MESSAGE.
4572 MOVE 141 ; TRAP @ERROR
4573 ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4574 ; EXIT WITH TEST FAILURE MESSAGE IF NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4575 BCS 604 ; DURING THE SOFTWARE QUESTIONS.
4576 MOV @BIT06, OPTION ; SELECT THE NEXT LINE NUMBER.
4577 MOV @BIT06, OPTION ; TEST FOR ALL LINES DONE.
4578 CMP R3, @NUMLN ; LOOP IF NOT ALL LINES DONE.
4579 BLT 163 ; INDICATE THAT WE ARE NOT WITHIN A TEST.
4580 MOV @PR107, R0 ; DISABLE ALL INTERRUPTS.
4581 ; TRAP @ERROR
4582 ENDTST
4583
4584 MOV @BIT06, OPTION
4585 BIT 00100, 163450
4586 BEQ 604
4587
4588 INC R3
4589 CMP R3, @NUMLN
4590 BLT 163
4591
4592 CLR CTRLCF
4593 SETPRI @PR107
4594 MOV @PR107, R0
4595 TRAP @PR107
4596 ENDTST
**SBITL** HARDWARE TEST

**DPRINT**

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.*

**BGNST**

**TB:**

*Only perform this test if the DUT is in external or staggered loopback mode.*

- **BIT** `#B1,LOPBACK` // Check type of loopback mode selected.
- **JMP** `601` // Exit this test if in internal loopback.
- **MOV** `#PRIOS,RO` // Set PRIORITIZATION.
- **TRAP** `CISPRI` // Increment the assembly time test counter.
- **MOV** `#NUM,#TSTNUM` // Set up the test number.
- **MOV** `#ERROR_TYPE,#ERROR_TABLE` // Set error type in error table.
- **MOV** `#ERROR_NUMBER,#ERROR_TABLE` // Set first error number in error table.

*Reset the DUT to a known state, remove status codes from the FIFO.*
- **JSR** `PC,CLNRST` // Reset the DUT.
- **BCS** `41` // Abort test if fatal error found in reset.
- **JSR** `PC,ASNLTL` // Set up the associated line tables.

*Set up a loop which handles one line per iteration.*
- **CLR** `R3` // Clear the line counter.
- **MOV** `R3,RO` // Clear all the DUT LNCRTL registers DTR bits.
- **ASL** `RO` // Specify that all LNCRTL bits to be cleared.
- **BIT** `#MAPRNS,R5` // Specify that all LNCRTL are to be changed.
- **JSR** `PC,MTWLNDC` // Clear all the DUT DTR bits.
MOV @A0,R4
JSR PC,DELAY ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.

;RECORD THE STATES OF THE MODEM STATUS SIGNALS.
;JSR PC,SAVMS
;SAVE THE PRESENT MODEM STATUS STATES.

;SET THE DTR FOR THE SELECTED LINE.

MOV R3,DCSA
;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
BIS @BIT9,DLNC
;SET THE SELECTED LINE DTR.
MOV @A0,R4
JSR PC,DELAY ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.

;CHECK THE PRESENT OUT STAT REGISTER CONTENTS AGAINST PREVIOUS.
;IF ANY UNDESIRED CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.

MOVB TXRLNB(R3),R1 ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
MOV @BIT15@BIT13,R2 ;IGNORE DSR AND RI ON ASSOCIATED LINE.
JSR PC,CMPS
;COMPARE OLD AND NEW STAT CONTENTS.
BSC 81 ;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.

;REPORT INTERACTIONS FOUND BETWEEN DTR FOR LINE NN AND THE FOLLOWING SIGNALS:
MOV @A0,R4
MOV @ERB401,ERBLK
;SELECT THE ERROR PRINT ROUTINE.
MOV @ERR402,R2
;SELECT THE DTR ERROR MESSAGES.
ERROR @ERR402,R1
;TRAP CERROR

;EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

BIT @BIT6,OPTION
BEQ 60$ ;EXIT WITH TEST FAILURE MESSAGE IF
;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED DURING THE SOFTWARE QUESTIONS.

;SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.

INC R3
;SELECT THE NEXT LINE NUMBER.
CMP R3,0NUMLNS
;TEST FOR ALL LINES DONE.
BLT 61$ ;LOOP IF NOT ALL LINES DONE.

;SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
;THIS LOOP SETS ALL THE DTRs AND THEN CHECKS 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.

MOV R3,RO
ASL RO
BIT BITBL(RO),ACTLNS
BEQ 12$ ;DON'T TEST IF NOT ACTIVE LINE.

SET ALL THE OUT NCNTL REGISTERS DTR BITS.

MOV @BIT9,RO
;SPECIFY THAT DTR BITS ARE TO BE SET.
MOV @HAPLN,R5
;SPECIFY THAT ALL NCNTL REG ARE TO BE CHANGED.
JSR PC,WTMNC
;SET ALL THE OUT DTR BITS.
4708 017036 012704 000050
4709 017042 004767 171074
4710
4711 ;RECORD THE STATES OF THE MODEM STATUS SIGNALS.
4712
4713 017046 004767 171750
4714 ;CLEAR THE DTA FOR THE SELECTED LINE.
4715
4716 017052 010377 163122
4717 017056 042777 001000 163124
4718 017064 012704 000050
4719 017070 004767 171046
4720
4721 ;CHECK THE PRESENT DUT STAT REGISTER CONTENTS AGAINST PREVIOUS.
4722 ;IF ANY UNDESIRED CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.
4723
4724 017074 116301 003750
4725 017100 012702 120000
4726 017104 004767 170740
4727 017110 103415
4728
4729 ;REPORT INTERACTIONS FOUND BETWEEN DTR FOR LINE NN AND THE FOLLOWING SIGNALS:
4730 017112 012767 020323 164672
4731 017120 012767 006420 164670
4732 017126 012701 005637
4733 017132 ERROR 104460
4734 ;EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4735
4736 017134 032767 000100 163020
4737 017142 001404
4738
4739 BIT @BIT06.OPTION :EXIT WITH TEST FAILURE MESSAGE IF
4740 BEQ 60:
4741 ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
4742 ;DURING THE SOFTWARE QUESTIONS.
4743
4744 ;SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.
4745
4746 017144 005203
4747 017146 020327 000020
4748 017152 002715
4749
4750 017154 005067 163042
4751 017160 SETPRI @PRI07
4752 017166 ENDST L10026:
4753 017166 TRAP CIETST
4754
**SBTTL**  HARDWARE TEST
**RISINT**

---

**REQUEST TO SEND SIGNAL INTERACTIONS TEST**

**THIS TEST VERIFIES THAT THE RTS SIGNAL (AND THE LOOPED BACK DCD AND CTS STATUS SIGNALS) DO NOT INTERACT WITH ANY OTHER MODEM STATUS SIGNALS.**

**IT WILL ONLY BE PERFORMED IF EITHER 25 PIN OR STAGGERED LOOPBACK IS SPECIFIED. THIS TEST IS PERFORMED ON ALL ACTIVE LINES.**

---

**BGNTST**

---

**T9:**

**ONLY PERFORM THIS TEST IF THE DUT IS IN EXTERNAL OR STAGGERED LOOPBACK MODE.**

**1:**

**BIT **@BIT1,LOPBC

**; CHECK TYPE OF LOOPBACK MODE SELECTED.**

**BNE 21**

**JMP 60**

**; EXIT THIS TEST IF IN INTERNAL LOOPBACK.**

**2:**

**SETPRI @PRI05

**; ALLOW LTC INTERRUPTS.**

**MDV @PRI05,RO**

**TRAP CISPR1**

---

**INUM = INUM + 1**

**; INCREMENT THE ASSEMBLY TIME TEST COUNTER.**

**MDV @INUM,TSTNUM**

**; SET UP THE TEST NUMBER.**

**; (BS3)**

**MDV @1,CTRLCF**

**; INDICATE THAT WE ARE IN A TEST.**

**MDV @1,ERTYPE**

**; SET ERROR TYPE IN ERROR TABLE.**

**MDV @88501,ERRNR**

**; SET THE FIRST ERROR NUMBER IN ERROR TABLE.**

**MDV @R8501,ERRMSG**

**; SET ERROR MESSAGE ADDRESS IN ERROR TABLE.**

---

**1:**

**; RESET THE DUT TO A KNOWN STATE. REMOVE STATUS CODES FROM THE FIFO.**

**1:**

**CLEAR TX AND RX INTERRUPT ENABLE BITS.**

**1:**

**THIS SUBROUTINE REPORTS ERROR >> >> >> 8501 << << <<.**

---

**JSR PC,CLNST**

**; RESET THE DUT.**

**BCS 41**

**JMP 60**

**; ABORT THE TEST IF FATAL ERROR FOUND IN RESET.**

---

**1:**

**SET UP THE TX/RX ASSOCIATED LINE NUMBER TABLE.**

---

**4:**

**JSR PC,ASLNLT**

**; SET UP THE ASSOCIATED LINE TABLES.**

**1:**

**SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.**

**; THIS LOOP CLEARS ALL THE RTS AND THEN SETS THEM INDIVIDUALLY AND CHECKS**

**; FOR ANY RESPONSES ON SIGNALS OTHER THAN THE ASSOCIATED DCD AND CTS SIGNALS.**

**1:**

**1:**

**1:**

**1:**

**1:**

---

**CLR R5**

**; CLEAR THE LINE COUNTER.**

**6:**

**MDV R5,RO**

**ASL RO**

**BIT BITBL(RO),ACTLN5**

**BEQ B4**

**; DON'T TEST IF NOT ACTIVE LINE.**

---

**CLEAR ALL THE DUT LNCRTL REGISTERS RTS BITS.**

---

**CLR RO**

**; SPECIFY THAT ALL LNCRTL BITS TO BE CLEARED.**

**MDV @MAPLNS,R5**

**; SPECIFY THAT ALL LNCRTL BITS ARE TO BE CHANGED.**

**JSR PC,WTNLNC**

**; CLEAR ALL THE DUT RTS BITS.**
4808 017316 012704 000000
4809 017322 004767 170614
4810
4811
4812
4813 017326 004767 17147C
4814
4815
4816
4817 017332 010577 162642
4818 017336 052777 010000 162644
4819 017344 012704 0000500
4820 017350 004767 170566
4821
4822
4823
4824
4825 017354 116301 0037500
4826 017360 012702 014000
4827 017364 004767 170460
4828 017370 103415
4829
4830 017372 012767 020466 164412
4831 017400 012767 006720 164410
4832 017406 012701 005744
4833 017412
4834
4835
4836
4837
4838
4839 017414 032767 000100 162540
4840 017422 001470
4841
4842
4843
4844 017424 005203
4845 017426 020327 000020
4846 017432 002716
4847
4848
4849
4850
4851
4852
4853 017434 005003
4854 017436 010300
4855 017440 006300
4856 017442 036067 002310 162522
4857 017450 001451
4858
4859
4860
4861 017452 012700 010000
4862 017456 012705 177777
4863 017462 004767 171742

MOV #40,R4
JSR PC,DELAY
;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.

;RECORD THE STATES OF THE MODEM STATUS SIGNALS.
;JSR PC,SAYMST
;SAVE THE PRESENT MODEM STATUS STATES.

;SET THE RTS FOR THE SELECTED LINE.

;MOV R3,BCSRA
;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
;BIS @BIT12,ALNCTR
;SET THE SELECTED LINE RTS.

;MOV #40,R4
;JSR PC,DELAY
;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.

;CHECK THE PRESENT DUT STAT REGISTER CONTENTS AGAINST PREVIOUS.
;IF ANY UNDESIRE CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.

;MOV YACTLB(R3),R1
;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
;MOV @BIT121BIT11,R2
;IGNORE DCD AND CTS ON ASSOCIATED LINE.
;JSR PC,CMPLST
;COMPARE OLD AND NEW STAT CONTENTS.
;BCS 81
;SKIP ERROR REPORT IF NO DISCREPANCIES FOUND.

;REPORT INTERACTIONS FOUND BETWEEN RTS FOR LINE NN AND THE FOLLOWING SIGNALS:
;MOV #502,ERRNR
;SELECT THE ERROR NUMBER.
;MOV @ERB401,ERRBLK
;SELECT THE ERROR PRINT ROUTINE.
;MOV @EM8502,R1
;SELECT THE RTS ERROR MESSAGES.
;ER1901 USES R1, R2, AND R3 VALUES.
;TRAP CERROR

;EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
;BIT @BIT6,OPTION
;EXIT WITH TEST FAILURE MESSAGE IF
;BEQ 60

;SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.
;INC R3
;SELECT THE NEXT LINE NUMBER.
;CMP R3,#NUMLNS
;TEST FOR ALL LINES DONE.
;BLT 64
;LOOP IF NOT ALL LINES DONE.

;SET UP A LOOP WHICH HANDLES ONE LINE PER ITERATION.
;THIS LOOP SETS ALL THE RTS AND THEN CHECKS THEM INDIVIDUALLY AND CHECKS
;FOR ANY RESPONSES ON SIGNALS OTHER THAN THE ASSOCIATED DCD AND CTS SIGNALS.
;THIS LOOP WILL CLEAR THE TX,IE AND RX,IE BITS IF THEY ARE SET.

;CLR R3
;CLEAR THE LINE COUNTER.
;MOV R3,RO
;ASL RO
;BIT BIT1B(RO),ACTLNS
;BEQ 12
;DON'T TEST IF NOT ACTIVE LINE.

;SET ALL THE DUT LNCTRL REGISTERS RTS BITS.
;MOV @BIT12,RO
;SPECIFY THAT RTS BITS ARE TO BE SET.
;MOV @APLNS,R5
;SPECIFY THAT ALL LNCTRLS ARE TO BE CHANGED.
;JSR PC,WMLNCC
;SET ALL THE DUT RTS BITS.
MOV @40,R4
JSR PC,DELAY ;DELAY FOR 40 MS TO ALLOW SIGNALS TO SETTLE.

;RECORD THE STATES OF THE MODEM STATUS SIGNALS.

;JSR PC,SÅVMST ;SAVE THE PRESENT MODEM STATUS STATES.

;CLEAR THE RTS FOR THE SELECTED LINE.

MOV R3,ACSRA ;SELECT THE SELECTED LINE IND.ADR.REG FIELD.
BIC @BIT12,ALNCTRA ;CLEAR THE SELECTED LINE RTS.
MOV @40,R4
JSR PC,DELAY ;ALLOW 40 MS FOR STATUS SIGNALS TO STABILIZE.

;CHECK THE PRESENT OUT STAT REGISTER CONTENTS AGAINST PREVIOUS.

;IF ANY UNDESIRABLE CHANGES HAVE TAKEN PLACE, REPORT THE ERRORS.

MOVB TXRLNB(R3),R1 ;SELECT SPECIAL TREATMENT FOR ASSOCIATED LINE.
MOV @BIT12,BIT11,R2 ;IGNORE DCD AND CTS ON ASSOCIATED LINE.
JSR PC,CMPS5 ;COMPARE OLD AND NEW STAT CONTENTS.

BSC 12$ ;SKIP ERROR REPORT IF NO DIScrepancies FOUND.

;REPORT INTERACTIONS FOUND BETWEEN RTS FOR LINE NH AND THE FOLLOWING SIGNALS:

MOV @8503,ERRNBR ;SELECT THE ERROR NUMBER.
MOV @ER8411,ERRBLK ;SELECT THE ERROR PRINT ROUTINE.

MOV @EM8502,R1 ;SELECT THE RTS ERROR MESSAGES.

ERROR 1ER1901 USES R1, R2, AND R3 VALUES.

;EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED

;EXIT WITH TEST FAILURE MESSAGE IF

BIT @BITO6,OPTION ;EXIT WITH TEST FAILURE MESSAGE IF
BEQ 60$ ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED

;DURING THE SOFTWARE QUESTIONS.

;SELECT THE NEXT LINE AND LOOP IF NOT ALL POSSIBLE LINES HAVE BEEN HANDLED.

INC R3 ;SELECT THE NEXT LINE NUMBER.
CMP R3,NUMLNS ;TEST FOR ALL LINES DONE.
BLT 10$ ;LOOP IF NOT ALL LINES DONE.

CLR CTRLCF ;INDICATE THAT WE ARE NOT WITHIN A TEST.

SETPRI @PR107 ;DISABLE ALL INTERRUPTS.

MOV @PR107,RO 
TRAP @10027 ;END TEST
**SBTLT HARDWARE TEST - REPBM**

1. **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.**
2. **IT IS UNLIKELY THAT RUNNING THIS PSEUDO-TEST ALONE WILL PRODUCE ANY ERROR REPORTS.**

**BGNST**

```
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 BMPCOR,B2 ; GET THE CONTENTS OF THE POINTER.
MOV @BMPCOR,B3 ; GET THE START ADDRESS OF THE QUEUE.
CMP R2,R3 ; SEE IF THE POINTER HAS MOVED FROM THE BASE.
BEQ 60D ; EXIT NO CODES IN THE QUEUE.
```

```
60D: THERE IS AT LEAST ONE BMP CODE IN THE QUEUE. REPORT THE ERROR.
    ; REPORT ERROR BMP CODE FOUND IN TEST N, BMP CODE:NNNNNN
    MOV @EM9304,R1 ; PASS THE FIRST MESSAGE TO BE REPORTED.
    ERRF 9301,EM9301,ER9301 ; >>> ERROR 9301 <<<
```

```
TRAP CERDF
    .WORD 9301
    .WORD EM9301
    .WORD ER9301
```

```
MOV @BMPCOR,BMPCOR ; SET POINTER BACK TO THE BEGINNING OF THE QUE.
```

```
60F: CLR CTRLCF ; INDICATE THAT WE ARE NOT WITHIN A TEST.
ENDST
```

L10030: TRAP CESTST
SBITL HARDWARE PARAMETER CODING SECTION

**THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS WITH THE OPERATOR.**

BGNHARD

.LWORD L10031-L1HARD/2

.DEVICE CSR ADDRESS QUESTION:
GPRMA HWPT01,0,0,160000,177776,YES

.LWORD TICODE
.LWORD HWPT01
.LWORD TILOLIM
.LWORD TIMILIM

.ACTIVE LINES BIT MAP QUESTION:
GPRMD HWPT02,2,0,MAPLNS,0,177777,YES

.LWORD TICODE
.LWORD HWPT02
.LWORD MAPLNS
.LWORD TILOLIM
.LWORD TIMILIM

.TYPE OF LOOPBACK QUESTION:
GPRMD HWPT03,4,0,377,1,3,YES

.LWORD TICODE
.LWORD HWPT03
.LWORD 377
.LWORD TILOLIM
.LWORD TIMILIM

ENDHARD

.EVEN

.LWORD L10031:

HWPT01: .ASCIZ /CSR ADDRESS:

017741 040 101 104
017744 104 122 105
017747 123 123 072
017752 040 000

HWPT02: .ASCIZ /ACTIVE LINE BIT MAP:

017754 101 103 124
017757 111 126 105
017762 040 114 111
017765 116 105 040
017770 102 111 124
017773 040 115 101
HMPTQ3: .ASCIZ /TYPE OF LOOPBACK (1=INTERNAL, 2=H3029 OR H3277, 3=H325):/

.ASCIZ
THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES. THE MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES. THE MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS WITH THE OPERATOR.

General Purpose Register Map (GPRM):

UNIT NUMBER PRINTOUT QUESTION:
GPRM1 SWPTQ1,0.20,YES

EXTENDED ERROR REPORTING QUESTION:
GPRM1 SWPTQ2,0,100,YES

IF EXTENDED ERROR REPORTING IS NOT REQUIRED THEN SKIP THE NEXT QUESTION.
XFER END

NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
GPRM1 SWPTQ3,2,0,177777.0,177777,YES

END:
<table>
<thead>
<tr>
<th>Address</th>
<th>Value 1</th>
<th>Value 2</th>
<th>Value 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>020161</td>
<td>125</td>
<td>116</td>
<td>111</td>
</tr>
<tr>
<td>020164</td>
<td>123</td>
<td>040</td>
<td>111</td>
</tr>
<tr>
<td>020167</td>
<td>123</td>
<td>040</td>
<td>124</td>
</tr>
<tr>
<td>020172</td>
<td>105</td>
<td>123</td>
<td>124</td>
</tr>
<tr>
<td>020175</td>
<td>105</td>
<td>104</td>
<td>072</td>
</tr>
<tr>
<td>020200</td>
<td>040</td>
<td>000</td>
<td></td>
</tr>
<tr>
<td>020202</td>
<td>105</td>
<td>130</td>
<td>124</td>
</tr>
<tr>
<td>020205</td>
<td>105</td>
<td>116</td>
<td>104</td>
</tr>
<tr>
<td>020210</td>
<td>105</td>
<td>104</td>
<td>040</td>
</tr>
<tr>
<td>020213</td>
<td>105</td>
<td>122</td>
<td>122</td>
</tr>
<tr>
<td>020216</td>
<td>117</td>
<td>122</td>
<td>040</td>
</tr>
<tr>
<td>020219</td>
<td>122</td>
<td>105</td>
<td>120</td>
</tr>
<tr>
<td>020221</td>
<td>117</td>
<td>122</td>
<td>124</td>
</tr>
<tr>
<td>020224</td>
<td>117</td>
<td>122</td>
<td>107</td>
</tr>
<tr>
<td>020227</td>
<td>111</td>
<td>116</td>
<td>107</td>
</tr>
<tr>
<td>020232</td>
<td>072</td>
<td>040</td>
<td>000</td>
</tr>
<tr>
<td>020235</td>
<td>116</td>
<td>125</td>
<td>115</td>
</tr>
<tr>
<td>020240</td>
<td>102</td>
<td>105</td>
<td>122</td>
</tr>
<tr>
<td>020243</td>
<td>040</td>
<td>117</td>
<td>106</td>
</tr>
<tr>
<td>020246</td>
<td>040</td>
<td>111</td>
<td>116</td>
</tr>
<tr>
<td>020251</td>
<td>104</td>
<td>111</td>
<td>126</td>
</tr>
<tr>
<td>020254</td>
<td>111</td>
<td>104</td>
<td>125</td>
</tr>
<tr>
<td>020257</td>
<td>101</td>
<td>114</td>
<td>040</td>
</tr>
<tr>
<td>020262</td>
<td>104</td>
<td>101</td>
<td>124</td>
</tr>
<tr>
<td>020265</td>
<td>101</td>
<td>040</td>
<td>105</td>
</tr>
<tr>
<td>020270</td>
<td>122</td>
<td>122</td>
<td>117</td>
</tr>
<tr>
<td>020273</td>
<td>122</td>
<td>123</td>
<td>040</td>
</tr>
<tr>
<td>020276</td>
<td>124</td>
<td>117</td>
<td>040</td>
</tr>
<tr>
<td>020281</td>
<td>122</td>
<td>105</td>
<td>120</td>
</tr>
<tr>
<td>020284</td>
<td>117</td>
<td>122</td>
<td>124</td>
</tr>
<tr>
<td>020287</td>
<td>040</td>
<td>117</td>
<td>116</td>
</tr>
<tr>
<td>020290</td>
<td>040</td>
<td>101</td>
<td>040</td>
</tr>
<tr>
<td>020295</td>
<td>114</td>
<td>111</td>
<td>116</td>
</tr>
<tr>
<td>020300</td>
<td>105</td>
<td>072</td>
<td>040</td>
</tr>
</tbody>
</table>

SMPDQ2: .ASCIZ /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE:/

SMPDQ3: .ASCIZ /EXTENDED ERROR REPORTING:/

.EVEN
```assembly
PATCH:
.LKW 24

LASTAD
020374 000000
020376 000000
020400

LAST:
. ENDMOD

5077 020400
5078
5079
5080
5081
5082
5083
5084
5085 000001
.END
```
<table>
<thead>
<tr>
<th>ACILNS</th>
<th>002172 G</th>
<th>CBST10000061</th>
<th>CDROP</th>
<th>002152 G</th>
<th>EDFM</th>
<th>0000004 G</th>
<th>EVL</th>
<th>0000004 G</th>
<th>IES</th>
<th>002250 G</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADR1</td>
<td>000200 G</td>
<td>CBRK</td>
<td>002222</td>
<td>CDRK</td>
<td>002222</td>
<td>EDFRM</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>ADAPR</td>
<td>010010 G</td>
<td>CBSEG</td>
<td>000044</td>
<td>CDSEG</td>
<td>000044</td>
<td>EDFMT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>ALTFL3</td>
<td>007370 G</td>
<td>CBSUM</td>
<td>000023</td>
<td>CDSUM</td>
<td>000023</td>
<td>DFM</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>ASLN</td>
<td>014422 G</td>
<td>CEPCS</td>
<td>000066</td>
<td>CDRCS</td>
<td>000066</td>
<td>DFGA</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>ASMB</td>
<td>0000001 G</td>
<td>CCLCK</td>
<td>000062</td>
<td>CDRCK</td>
<td>000062</td>
<td>DFDA</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>ASMBTS</td>
<td>0000001 G</td>
<td>CCLC</td>
<td>000012</td>
<td>CDLC</td>
<td>000012</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>ASMTL</td>
<td>012274 G</td>
<td>CDLCR</td>
<td>000035</td>
<td>CDRCL</td>
<td>000035</td>
<td>DFDS</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT</td>
<td>000127 G</td>
<td>CDLCS</td>
<td>000035</td>
<td>CDRCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT8</td>
<td>010000 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT80</td>
<td>000000 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT100</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT1000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT10000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT100000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT1000000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT10000000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT100000000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT1000000000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT10000000000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT100000000000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT1000000000000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT10000000000000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
<tr>
<td>BCT100000000000000</td>
<td>0000001 G</td>
<td>CDLST</td>
<td>000035</td>
<td>CDSCU</td>
<td>000035</td>
<td>DFDT</td>
<td>0000004</td>
<td>EVL</td>
<td>0000004</td>
<td>IAP</td>
</tr>
</tbody>
</table>