DHV11-M

DHV11-M FUNC TST PART 3
CVDHCD0

AH-T657D-MC
1 OF 2   OCT 1985
COPYRIGHT© 1983-85

digital
MADE IN USA

DHV11-M

DHV11-M FUNC TST PART 3
CVDHCD0

AH-T657D-MC
2 OF 2   OCT 1985
COPYRIGHT© 1983-85

digital MADE IN USA

.REM &

IDENTIFICATION

PRODUCT CODE: AC-T656D-MC

PRODUCT NAME: CVDHCDO DHV11-M FUNC TST PART 3

PRODUCT DATE: 26 APRIL 1985

MAINTAINER: Bruce Ribolini - MK Diagnostics Group

AUTHOR: Bert Kleinschmidt
                 Tony Grimshaw

MODIFIED BY: Bert Kleinschmidt
                 Anthony Hart
                  Peter ONeil

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

Original release: 31-OCT-83 (EDITED 11-JUL-83)
                           Bert Kleinschmidt

Version B0   09-OCT-83   Bert Kleinschmidt
        Fixed typographical errors.
        Moved tests from this program to CVDHB (Part 2):
                Old CVDHC (version A) tests 4 through 6 are
                now New CVDHB (version B) tests 13 through 15.
        Added 3 new tests (4 through 6) to increase test coverage.

Version C0   17-JUL-84   Bert Kleinschmidt
        Modified control of processor priority and LTC throughout
        the program to guarantee a less than 2 second
        responce to a Break request while running under APT.

Version C0   28-Sep-84   Peter ONeil
        Modified VDHUC.CUC to turn off clock interrupt if clock
        was enabled.

Version D0   26-Apr-85   Howard L. Marshall
        Changed test 4, DMA Addressing Test to use KPAR5 instead of
        KPAR6, and as to not modify KPAR0, 6 + 7.

        Modified test 7, Single Character Mode Test, to run under the
        new Extended monitor of XXDP+ by changing minimum baud rate
        used from 50 to 300.

        In addition, found test 8, DMA Mode Test, to be a failure under
        the Extended monitor because the DRS and XXDP+ services require
        more time to execute the SETPRI + GETPRI macros. Substituted
        those macros with MTPS + MFPS respectively.

TABLE OF CONTENTS

1.0 GENERAL PROGRAM CONSIDERATIONS

1.1 PROGRAM ABSTRACT

CVDHC is part three of the DHV11-M functional verification test. This part of the test verifies that the major communication functions of the board which use the UARTs are functioning correctly. This program exercises the board by transmitting and receiving large blocks of data in loopback.

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

1.2 SYSTEM REQUIREMENTS

The following hardware is required to run the DHV FVT:

o LSI-11 processor with at least 32 Kbytes of RAM.

o DHV11-M boards installed on the Q-bus.

o Appropriate program load device supporting XXDP+ media or a down-line loading system.

1.3 RELATED DOCUMENTS AND STANDARDS

o DHV11-M Hardware Manual - This manual describes the functions and uses of the DHV11-M device.

o XXDP+ User's Manual - Describes the running of diagnostics under the XXDP+ monitor.

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

The LSI-11 processor, the Q-BUS, the system memory, the
console terminal, and the load media are assumed to have
been tested and found working before this program is run.

2.0 OPERATING INSTRUCTIONS

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

2.1 COMMANDS

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

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

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

2.2 SWITCHES

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

SWITCH	EFFECT
/TESTS:LIST	EXECUTE ONLY THOSE TESTS SPECIFIED IN THE LIST. LIST IS A STRING OF TEST NUMBERS, FOR EXAMPLE - /TESTS:1:5:7-10.

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

EXAMPLE OF SWITCH USAGE:

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

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

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

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

2.3 FLAGS

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

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

*SEE THE ERROR INFORMATION SECTION OF THIS DOCUMENT.

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

/FLAGS:LOE:IER:BOE

2.4 EXTENDED COMMAND SYNTAX

2.4.1 START COMMAND -

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

2.4.1.1 Tests Switch (/TESTS:<TEST-LIST>) -

    <TEST-LIST> Is a sequence of decimal numbers (1:2 etc.)
or ranges of decimal numbers (1-5:8-10 etc.), seperated by
colons, that specify the tests to be executed.Tests will be
executed in numerical order regardless of the order of
specification. The default is to execute all tests. On
this and all switches, the angle brackets <> are punctuation
used in the definition only, and are not to be typed by the
operator. See example at end of "Effect of Start Command"
section.

2.4.1.2 Pass Switch (/PASS:<PASS-CNT>) -

    <PASS-CNT> Is a decimal number indicating the desired
number of passes. A pass is defined as the execution of the
full diagnostic (all selected tests).The default is
non-ending execution. In this case, exit from the program
is accomplished either by typing a control/C or by occurance
of an error with the halt on error flag being set. The exit
is a return to command mode. See example at end of "Effect
of Start Command" section.

2.4.1.3 Flags Switch (/FLAGS:<FLAG-LIST>) -

    <FLAG-LIST> is a sequence of elements of the form
<FLAG>, <FLAG=1>, or <FLAG=0>, separated by colons, where
<FLAG> has one of the following values:

HOE	Halt on error, causing command mode to be entered when an error is encountered.
LOE	Loop on error, causing the diagnostic to loop continuously within the smallest defined block of coding (segment, subtest, or test) containing the error.
IER	Inhibit error reporting.
IBE	Inhibit basic error reports.
IXE	Inhibit extended error reports.
PRI	Direct all messages to a line printer.
PNT	Print number of test being executed.
BOE	Bell on error.
UAM	Run in unattended mode, bypassing manual

```
            intervention.
     ISR    Inhibit statistical reports.
     IDU    Inhibit dropping of units by diagnostic.
     LOT    Loop on test.
```
The flags named or equated to 1 are set. those equated to 0
are cleared. A flag not specified is cleared. If the flags
switch is not given all flags are cleared. See example at
end of "Effect of Start Command" section.

2.4.1.4 End Of Pass Switch (/EOP:<INCR>) -

<INCR> Is a decimal number indicating how often (in
terms of passes) it is desired that the end of pass message
be printed. The default is at the end of every pass.   See
example at end of "Effect of Start Command" section.

2.4.1.5 Effect Of Start Command -

The effect of the start command is to initiate the
hardware   parameter   dialogue,   the   software parameter
dialogue, the initialization questions, and   then   the
diagnostic commences testing.

The hardware parameter dialogue commences with the
question "# UNITS (D) ?" to which the operator should reply
with the number of units to be tested. Following this are
the questions whereby the P-Tables themselves are built.
Each P-Table is a core-resident table containing all the
hardware information for one complete unit.Each question is
followed by the response radix (D for decimal, B for binary,
O for octal, L for Yes/No) in parentheses and the default
value after the parentheses.   For the actual Hardware
P-Table questions see the "Hardware Parameters" section.

Following the hardware questions are the software
questions   to build the software tables, which define
operating parameters of the diagnostic program.   These
Questions   are   described in the "Software Parameters"
section.

EXAMPLE:

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

This command will cause three passes to be made, with
each pass consisting of tests 1,3, and 4. There is no
difference between saying <FLAG> and saying <FLAG=1>.   The
notation <FLAG=0> is meaningful only on a command other than
start to clear a flag that was previously set. Note that on
all commands only the first three letters are scanned.

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

2.4.2.1 Tests, Pass, And Flags Switches -

     <TEST-LIST>, <PASS-CNT>, and <FLAG-LIST> are as in the
start command.

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

2.4.2.3 Effect Of Restart Command -

     The restart command differs from the start command in
that the P-Tables from the previous start command (there
must have been one) are used, instead of new ones being
built.   The software dialogue may optionally be reexecuted
(operator will be asked). The command can be used after
command mode has been reentered in any of the three normal
ways: a) the requested number of passes have been made, b)
an error was encountered with the halt on error flag set, or
c) a control/C was entered by the operator.

2.4.3 Continue Command -
```
****************************************************************
CON(TINUE)/PASS:<PASS-CNT/FLAGS:<FLAG-LIST>
****************************************************************
```
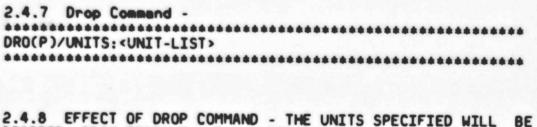
2.4.3.1 Flag Switch (/FLAGS:<FLAG-LIST>) -

<FLAG-LIST> Is same as in the start command, but unspecified flags retain their current value.

2.4.3.2 Effect Of Continue Command -

Continue must follow a start or restart, and command mode must have been entered due to a halt on error or a control/C. The effect of the command is to go to the beginning of the test that was being executed when the halt or control/C took place. Software dialogue may optionally be reexecuted. Hardware parameters may not be changed.

2.4.4 Proceed Command -
```
***************************************************************
PRO(CEED)/FLAGS:<FLAG-LIST>
***************************************************************
```


2.4.4.1 Flags Switch (/FLAGS:<FLAG-LIST>) -

<FLAG-LIST> Is as in the start command, but unspecified flags retain their current value.

2.4.4.2 Effect Of Proceed Command -

Proceed must follow a start, restart, or continue. Command mode must have been entered via a halt on error. The effect of the command is to begin execution at the location following the error call. Neither hardware nor software parameters may be altered.

2.4.5 Add Command -
```
***************************************************************
ADD/UNITS:<UNIT-LIST>
***************************************************************
```


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

2.4.7 Drop Command -
```
***************************************************************
DRO(P)/UNITS:<UNIT-LIST>
***************************************************************
```

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

2.4.9 Print Command -
```
***************************************************************
PRI(NT)
***************************************************************
```

2.4.9.1 Effect Of Print Command - Error summary reporting
is not implemented in this diagnostic, so this command has
no effect.

2.4.10 Display Command -
```
***************************************************************
DIS(PLAY)/UNITS:<UNIT-LIST>
***************************************************************
```

2.4.10.1 Effect Of Display Command -

The hardware P-Tables for all units are printed in the
format in which they were entered.

2.4.11 Flags Command -
```
***************************************************************
FLA(GS)
***************************************************************
```

2.4.11.1 Effect Of Flags Command -

The current settings of all flags are printed.

2.4.12 Zflags Command -
****
ZFL(AGS)
****

2.4.13 Zflags Command -

   All flags are cleared.

2.4.14 Control Characters -

   C A control/C (C) entered during the execution of a
      diagnostic causes a return to command mode.

   Z A control/Z (Z) entered during one of the two
      operator dialogues-- hardware P-Table dialogue or
      software P-Table dialogue causes the defaults to be
      taken for the remainder of that dialogue.

   O A control/O (O) entered during the execution of a
      diagnostic causes all teletype output to be
      surpressed for the remainder of the diagnostic or
      until another control/O is typed, which restores
      normal teletype output.

2.5 HARDWARE QUESTIONS

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

1. CSR ADDRESS - This question requests the CSR address of the
    specified DHV11-M. The default answer for this question is the
    lowest address in the PDP-11 floating address space in which
    a DHV11-M can be placed (160460 Octal).

2. INTERRUPT VECTOR ADDRESS - This question requests   the
    interrupt vector address of the specified DHV11-M.

3. ACTIVE LINES BIT MAP - This question requests an octal bit
    map of the serial communication lines on the DHV11-M which are
    being selected for testing. If the bit in the bit map is set
    which corresponds to a particular line (i.e. bit 3 for line
    3) that line will be tested by the FVT.   With staggered
    loopback a pair of lines with the specified transmit line and
    another receive line will be tested. Therefore, to guarantee
    that both the transmitter and receiver of a specified line
    are tested when using the staggered loopback connector, both
    the intended line AND its mate must be selected (ie. to test
    line 1, select both line 1 and line 3).   In nonstaggered
    testing, a bit in the active lines bit map selects the
    transmitter and receiver for the same line.

4. TYPE OF LOOPBACK (1=INTERNAL, 2=H3277, 3=H325, 4=MODEM,
    5=KEYBOARD ECHO) - This question requests the type of
    loopback to be used in testing the DHV11-M.   The following
    types of loopback are supported:

    o INTERNAL - Only internal UART loopback is to be used in
       testing the DHV.

    o H3277 - Staggered Berg connector(s) are installed at the
       end of the 40 wire cables in place of the DHV11-M
       distribution panels.

    o H325 - Single line, 25 pin loopback connectors (type
       H325) are installed on the lines to be tested. These
       connectors can be installed on the distribution panel or
       on the end of the terminal or modem cable. The H325
       connectors must have the removable jumpers installed.

    o MODEM - The operator is allowed to set up a modem link
       and then perform a transmission and reception test at a
       single baudrate with the modem control signals DTR and
       RTS active. This testing is performed in test 5 which is
       a special test.   All other tests are performed in
       internal loopback.

o KEYBOARD ECHO - The UARTs on the DUT are placed in remote
   loopback.   Terminals (or other communications equipment)
   will have whatever they transmit to the DHV looped back
   to them.

5. BR Level - This questions requests the interrupt BR level of
    the DHV11-M.

2.6 SOFTWARE QUESTIONS

AFTER YOU HAVE ANSWERED THE HARDWARE QUESTIONS OR AFTER A RESTART OR
CONTINUE   COMMAND,   THE RUNTIME SERVICES WILL ASK FOR SOFTWARE
PARAMETERS. THESE PARAMETERS WILL GOVERN SOME DIAGNOSTIC SPECIFIC
OPERATION MODES.   YOU WILL BE PROMPTED BY "CHANGE SW (L) ?" IF YOU
WISH TO CHANGE ANY PARAMETERS, ANSWER BY TYPING "Y".   The following
Software P-Table questions are asked by the program if the operator
indicates that the Software Parameters are to be changed:

1. REPORT UNIT NUMBER AS EACH UNIT IS TESTED - This question
    asks whether the program should report the number of the unit
    which it is testing as it begins to test each unit.

2. NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE - This
    question asks for the number of data errors which should be
    reported individually by this program for each line for each
    transmission   test.   Errors   which   are   not   reported
    individually are reported in summary error reports.

3. REPORT NUMBER OF BITS TESTED IN DMA ADDR TEST - This
    questions asks whether the operator wants a printout
    describing which address bits have been tested when the DMA
    Addressing Test (test 4) executes.-

2.7 EXTENDED P-TABLE DIALOGUE

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

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

```
                # UNITS (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) 1 ?  <CR>
```

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

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

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

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

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

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

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

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

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

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

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

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

2.8 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

1. BOOT XXDP+

2. GIVE THE DATE AND ANSWER THE LSI AND 50HZ (IF THERE
    IS A CLOCK AND THE QUESTION IS ASKED) QUESTIONS

3. TYPE "R NAME", WHERE NAME IS THE NAME OF THE BIN OR
    BIC FILE FOR THIS PROGRAM

4. TYPE "START"

5. ANSWER THE "CHANGE HW" QUESTION WITH "Y"

6. ANSWER ALL THE HARDWARE QUESTIONS

7. ANSWER THE "CHANGE SW" QUESTION WITH "N"

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

THE GENERAL ERROR MESSAGE IS OF THE FORM:

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

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

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

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

                    FLAGS SECTION OF THIS DOCUMENT).
                    THESE MESSAGES ARE PRINTED AFTER THE ASSOCIATED GENERAL ERROR
                    MESSAGE AND ANY ASSOCIATED BASIC ERROR MESSAGES.

    3.2 ERROR MESSAGES

    This program is intended to provide a go/no-go indication of the
    functionality of DHV11-M boards. To execute the program in this mode
    the operator can run with the inhibit basic error reporting switch.
    In this mode the program prints error messages which contain the error
    message header described above, plus the name of the failing test.
    For a list of the test names in this program see the test summaries
    section of this document. An example of such an error message is the
    following:

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

    This error indicates that a fatal error was encountered within the
    test which tests the read/write capability of the DHV11-M registers.

    If the operator requires more extensive error reporting he can run
    with all error reporting enabled by not using the inhibit reporting
    switches. The above error message would then become the following:

    CVDHC DVC FTL ERR 01603 ON UNIT 02 TST 015 SUB 000 PC: 015244
    DEVICE REGISTER WORD READ/WRITE TEST
      BAD BIT(S) IN DEVICE TBUFFAD1 REGISTER FOR LINE 7 (D).
        EXPECTED DATA: 000000 (O).
        ACTUAL DATA: 000023 (O).

4.0 PERFORMANCE AND PROGRESS REPORTS

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

5.0 TEST SUMMARIES

The following tests are included within CVDHC:

1. Device register address test - Verifies that the   UUT
   registers will respond witht the proper Q-BUS handshaking
   when accessed. Verifies that the UUT is at the proper
   address.

2. FRAMING.ERROR test - Verifies that forced framing errors are
   reported correctly.

3. PARITY.ERROR test - Verifies that forced parity errors are
   reported correctly.

4. DMA Addressing test - Verifies that the UUT can access the
   full memory which is on the host machine via DMA accesses.

5. Modem Loopback test - Allows the operator to test modem links
   which are attached to UUT serial ports.

6. Terminal Echo test - Allows the operator to test terminal
   links (or other communication links), which are attached to
   UUT serial ports, from the remote ends of the links.

7. Single character mode TX/RX test - Verifies that the UUT will
   TX and RX data correclty in single character mode.

8. DMA mode TX/RX test - Verifies that the UUT will TX and RX
   data correctly using DMA transmission.

9. Split speed test - Verifies that the UUT will work with
   different TX and RX speeds on each active line.

10. Report BMP codes test - This pseudo test reports the first 32
    BMP codes which were discovered in the FIFO during the
    execution of the other tests. This avoids the interruption
    of other tests by these codes, if they are not critical to
    the tests being performed.

6.0 EXAMPLE ERROR FREE PASS

6.1 Pass With Default Parameters

The following is an example of an error free pass dialogue using a
standard loopback:
.R CVDHCCO
CVDHCCO.BIC

DRS
CVDHC-C-O
DHV11-M FUNC TST PART 3
UNIT IS DHV11-M
RESTART ADDR: 147670
DR>STA

CHANGE HW (L) ? Y

UNITS (D) ? 2

UNIT 0
CSR ADDRESS: (O) 160460 ? ↑Z

UNIT 1
CSR ADDRESS: (O) 160460 ? 160040
INTERRUPT VECTOR ADDRESS: (O) 300 ? 320
ACTIVE LINE BIT MAP: (O) 377 ? <CR>
TYPE OF LOOPBACK (1=INTERNAL, 2=H3277, 3=H325,
                  4=MODEM, 5=KEYBOARD ECHO): (O) 2 ? 1
INTERRUPT BR LEVEL: (O) 4? <CR>

CHANGE SW (L) ? Y

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

 TESTING UNIT : 0(D)

 DMA ADDRESS TEST SUCCESSFUL, BITS 0 TO 18 TESTED (19 BITS).

 TESTING UNIT : 1(D)

 DMA ADDRESS TEST SUCCESSFUL, BITS 0 TO 18 TESTED (19 BITS).

CVDHC EOP   1
    0 CUMULATIVE ERRORS

 TESTING UNIT : 0(D)
↑C
DR> EXIT
.

6.2 Pass With Modem Loopback

The following is an example of an error free pass dialogue with Modem
Loopback selected:
.R CVDHCCO
CVDHCCO.BIC

DRS
CVDHC-C-O
DHV11-M FUNC TST PART 3
UNIT IS DHV11-M
RESTART ADDR: 147670
DR>STA

CHANGE HW (L) ? Y

UNITS (D) ? 1

UNIT 0
CSR ADDRESS: (O) 160460 ? <CR>
INTERRUPT VECTOR ADDRESS: (G) 300 ? <CR>
ACTIVE LINE BIT MAP: (O) 377 ? 5
TYPE OF LOOPBACK (1=INTERNAL, 2=H3277, 3=H325,
                  4=MODEM, 5=KEYBOARD ECHO): (O) 2 ? 4
INTERRUPT BR LEVEL: (O) 4? <CR>

CHANGE SW (L) ? N

EXECUTING THE MODEM LOOPBACK TEST

MODEM BAUDRATE IN BPS: (D) 1200 ? 2400

TYPE <CR) WHEN MODEM LINK ESTABLISHED: (L) Y ?
MODEM STATUS SIGNAL REPORT:
  LINE # 0: DSR=1, RI=1, DCD=1, CTS=1
  LINE # 2: DSR=1, RI=1, DCD=1, CTS=1 .

NUMBER OF 256 BYTE PATTERNS TO SEND ON EACH SELECTED LINE
  (1-255, 0=SEND UNTIL †C): (D) 1 ? 2

PRINT MODEM STATUS SIGNAL REPORT AFTER EACH PATTERN: (L) Y ? N

MODEM LOOPBACK TEST STATUS REPORT: PATTERN #   1 (D) COMPLETED.
MODEM LOOPBACK TEST STATUS REPORT: PATTERN #   2 (D) COMPLETED.

EXIT THE TEST (N = LOOP BACK TO SEND MORE DATA): (L) Y ? <CR>

CVDHC EOP   1
    O CUMULATIVE ERRORS

EXECUTING THE MODEM LOOPBACK TEST

MODEM BAUDRATE IN BPS: (D) 1200 ? †C

DR> EXIT

6.3 Pass With Keyboard Echo

The following is an example of an error free pass dialogue with
Keyboard Echo Loopback selected:

```
.R CVDHCCO
CVDHCCO.BIC

DRS
CVDHC-C-O
DHV11-M FUNC TST PART 3
UNIT IS DHV11-M
RESTART ADDR: 147670
DR>STA

CHANGE HW (L)  ? Y

# UNITS (D)  ? 1

UNIT 0
CSR ADDRESS:  (O)  160460 ? <CR>
INTERRUPT VECTOR ADDRESS:  (O) 300 ? <CR>
ACTIVE LINE BIT MAP:  (O)  377 ? <CR>
TYPE OF LOOPBACK (1=INTERNAL, 2=H3277, 3=H325,
                  4=MODEM, 5=KEYBOARD ECHO): (O)  2 ? 5
INTERRUPT BR LEVEL:  (O)  4? <CR>

CHANGE SW (L)  ? N

EXECUTING THE KEYBOARD ECHO (DHV REMOTE LOOPBACK) TEST

MODEM BAUDRATE IN BPS: (D)  1200 ? 9600

TYPE <CR> TO TERMINATE THE TEST: (L) Y ? <CR>

CVDHC EOP    1
    0 CUMULATIVE ERRORS

EXECUTING THE KEYBOARD ECHO (DHV REMOTE LOOPBACK) TEST

MODEM BAUDRATE IN BPS: (D)  1200 ? ↑C

DR> EXIT
```

```
                                    &
1170                                    .LIST SEQ,LOC,BIN,MEB
1171 000000                             .ENABLE  ABS,AMA,LC
1172                                     .NLIST CND
```

```
1174                                    ;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1175                                    ;
1176                                    ;                VDHC.PHD
1177                                    ;
1178                                    ;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1179
1180                                    .SBTTL  Program Header
1181
1182                                            .MCALL  SVC
1183 000000                                     SVC                                      ; INITIALIZE SUPERVISOR MACROS
1184
1185                                    ;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1186                                    ;    IF STRUCTURED MACROS ARE TO BE USED, ADD ".MCALL STRUCT" AND "STRUCT"
1187                                    ;    TO INITIALIZE THE STRUCTURED MACROS.
1188
1189         000001                     SVCINS= 1       ; LIST INSTRUCTIONS, SHIFTED RIGHT
1190         000001                     SVCTST= 1       ; LIST TEST TAGS, SHIFTED RIGHT
1191         000001                     SVCSUB= 1       ; LIST SUBTEST TAGS, SHIFTED RIGHT
1192         000001                     SVCGBL= 1       ; LIST GLOBAL TAGS, SHIFTED RIGHT
1193         000001                     SVCTAG= 1       ; LIST OTHER TAGS, SHIFTED RIGHT
1194
1195                                    ;    CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
1196                                    ;    TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
1197                                    ;    SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
1198                                    ;    CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
1199                                    ;&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&&
1200
1201 000000                             .ENABL  ABS
1202                                    ;.ENABL AMA
1203         002000                     .       =       2000
1204
1205 002000                                     BGNMOD
1206
1207                                    ;++
1208                                    ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
1209                                    ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
1210                                    ;--
1211
1212 002000                                     POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL
1213
1230
1231 002000                                     HEADER  CVDHC,D,0,200,0,PRIO7
     002000                                                                             L$NAME::
     002000     103                                                                             .ASCII  /C/
     002001     126                                                                             .ASCII  /V/
     002002     104                                                                             .ASCII  /D/
     002003     110                                                                             .ASCII  /H/
     002004     103                                                                             .ASCII  /C/
     002005     000                                                                             .BYTE   0
     002006     000                                                                             .BYTE   0
     002007     000                                                                             .BYTE   0
     002010                                                                             L$REV::
     002010     104                                                                             .ASCII  /D/
     002011                                                                             L$DEPO::
     002011     060                                                                             .ASCII  /0/
     002012                                                                             L$UNIT::
     002012     000000                                                                          .WORD   0
```

```
        002014                          L$TIML::
        002014  000200                          .WORD   200
        002016                          L$HPCP::
        002016  037354                          .WORD   L$HARD
        002020                          L$SPCP::
        002020  037720                          .WORD   L$SOFT
        002022                          L$HPTP::
        002022  002152                          .WORD   L$HW
        002024                          L$SPTP::
        002024  002164                          .WORD   L$SW
        002026                          L$LADP::
        002026  040246                          .WORD   L$LAST
        002030                          L$STA::
        002030  000000                          .WORD   0
        002032                          L$CO::
        002032  000000                          .WORD   0
        002034                          L$DTYP::
        002034  000000                          .WORD   0
        002036                          L$APT::
        002036  000000                          .WORD   0
        002040                          L$DTP::
        002040  002124                          .WORD   L$DISPATCH
        002042                          L$PRIO::
        0C2042  000340                          .WORD   PRIO7
        002044                          L$ENVI::
        002044  000000                          .WORD   0
        002046                          L$EXP1::
        002046  000000                          .WORD   0
        002050                          L$MREV::
        002050     004                          .BYTE   C$REVISION
        002051     000                          .BYTE   C$EDIT
        002052                          L$EF::
        002052  000000                          .WORD   0
        002054  000000                          .WORD   0
        002056                          L$SPC::
        002056  000000                          .WORD   0
        002060                          L$DEVP::
        002060  005366                          .WORD   L$DVTYP
        002062                          L$REPP::
        002062  030114                          .WORD   L$RPT
        002064                          L$EXP4::
        002064  000000                          .WORD   0
        002066                          L$EXP5::
        002066  000000                          .WORD   0
        002070                          L$AUT::
        002070  000000                          .WORD   0
        002072                          L$DUT::
        002072  031024                          .WORD   L$DU
        002074                          L$LUN::
        002074  000000                          .WORD   0
        002076                          L$DESP::
        002076  005376                          .WORD   L$DESC
        002100                          L$LOAD::
        002100  104035                          EMT     E$LOAD
        002102                          L$ETP::
        002102  005316                          .WORD   L$ERRTBL
        002104                          L$ICP::
```

```
           002104   030130                              .WORD   L$INIT
           002106                          L$CCP::
           002106   030774                              .WORD   L$CLEAN
           002110                          L$ACP::
           002110   030772                              .WORD   L$AUTO
           002112                          L$PRT::
           002112   030122                              .WORD   L$PROT
           002114                          L$TEST::
           002114   000000                              .WORD   0
           002116                          L$DLY::
           002116   000000                              .WORD   0
           002120                          L$HIME::
           002120   000000                              .WORD   0
     1232
```

```
1244                                    .SBTTL   DISPATCH TABLE
1245
1246                                    ;++
1247                                    ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1248                                    ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1249                                    ;--
1250
1251 002122                                     DISPATCH 10
     002122  000012                                             .WORD   10
     002124                                    L$DISPATCH::
     002124  031142                                             .WORD   T1
     002126  031432                                             .WORD   T2
     002130  032004                                             .WORD   T3
     002132  032370                                             .WORD   T4
     002134  034166                                             .WORD   T5
     002136  035060                                             .WORD   T6
     002140  035306                                             .WORD   T7
     002142  036036                                             .WORD   T8
     002144  036646                                             .WORD   T9
     002146  037272                                             .WORD   T10
1252
```

```
1260
1261                    ;**********************************************************************************
1262                    ;
1263                    ;                    VDHC.DHT
1264                    ;
1265                    ;**********************************************************************************
********
1266
1267
1268                    .SBTTL   DEFAULT HARDWARE P-TABLE
1269
1270                    ;++
1271                    ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
1272                    ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
1273                    ; IS IDENTICAL TO THE STRUCTURE OF THE HARDWARE P-TABLES,
1274                    ; AND IS USED AS A "TEMPLATE" FOR BUILDING THE P-TABLES.
1275                    ;--
1276
1277 002150                     BGNHW    DFPTBL
     002150  000004                                                                    .WORD   L10000-L$HW/2
     002152                                                          L$HW::
     002152                                                          DFPTBL::
1278
1279 0C2152  160460             .WORD    160460    ;Default CSR Address
1280 002154  000300             .WORD    300       ;Default Vector Address
1281 002156  177777             .WORD    177777    ;Default Active lines bit map
1282 002160     002             .BYTE    2         ;Default Loopback mode
1283 002161     004             .BYTE    4         ;Default BR Level
1284
1285 002162                     ENDHW
     002162                                                          L10000:
```

```
1287
1288                  ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1289
1290                  ;                      VDHC.SWT
1291
1292                  ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1293
1294
1295                  .SBTTL   SOFTWARE P-TABLE
1296
1297                  ;++
1298                  ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
1299                  ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
1300                  ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
1301                  ; AT RUN TIME.
1302                  ;--
1303
1304 002162                  BGNSW   SFPTBL
     002162  000002                                                     .WORD   L10001-L$SW/2
     002164                                                    L$SW::
     002164                                                    SFPTBL::
1305
1306 002164  000020          OPTION::    .WORD   20      ;bit map of program control flags
1307 002166  000000          NDERPT::    .WORD   0       ;Default number of individual data errors to rpt.
1308
1309 002170                  ENDSW
     002170                                                    L10001:
```

```
1311
1312                          ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
*
1313                          ;
1314                          ;              VDHC.EQU
1315                          ;
1316                          ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1317
1318
1319                          .SBTTL   GLOBAL EQUATES SECTION
1320
1330
1331
1332                          ;++
1333                          ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
1334                          ; ARE USED IN MORE THAN ONE TEST.
1335                          ;--
1336
1337       000010                 NUMLNS==10        ;NUMBER OF LINES ON DHV11-M IS 8.
1338       000377                 MAPLNS==377       ;BIT MAP OF LINES ON DHV11-M.
1339
1340                          ;****** DEVICE REGISTER OFFSETS FROM THE CSR'S ADDRESS ******
1341       000000                 CSRO==0           ;CSR REGISTER OFFSET FROM THE CSR ADDRESS
1342       000002                 RBUFO==2          ;RECEIVE REGISTER OFFSET FROM THE CSR ADDRESS
1343       000002                 TXCHRO==2         ;TRANSMIT REGISTER OFFSET FROM THE CSR ADDRESS
1344       000004                 LPRO==4           ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
1345       000006                 STATO==6          ;STATUS REGISTER OFFSET FROM THE CSR ADDRESS
1346       000010                 LNCTRO==10        ;LINE CONTROL REGISTER OFFSET FROM THE CSR ADDRESS
1347       000012                 TXAD1O==12        ;TRANSMIT ADDRESS 1 REGISTER OFFSET FROM THE CSR ADDRESS
1348       000014                 TXAD2O==14        ;TRANSMIT ADDRESS 2 REGISTER OFFSET FROM THE CSR ADDRESS
1349       000016                 TXBFCO==16        ;TRANSMIT COUNT REGISTER OFFSET FROM THE CSR ADDRESS
1350
1351                          ;****** EQUATES USED WITH RESPECT TO THE RX BUFFER ******
1352       000020                 RXBETX==16.       ;LEVEL OF RX BUFFER AT WHICH TO RE-ENABLE TRANSMISSION.
1353       000030                 RXBDTX==24.       ;LEVEL OF RX BUFFER AT WHICH TO DISABLE TRANSMISSION.
1354       000100                 RXBFUL==64.       ;TOTAL CHARACTER CAPACITY OF THE RX BUFFER.
1355
1356
1371 002170                     EQUALS
                               ;
                               ; BIT DIFINITIONS
                               ;
          100000              BIT15== 100000
          040000              BIT14== 40000
          020000              BIT13== 20000
          010000              BIT12== 10000
          004000              BIT11== 4000
          002000              BIT10== 2000
          001000              BIT09== 1000
          000400              BIT08== 400
          000200              BIT07== 200
          000100              BIT06== 100
          000040              BIT05== 40
          000020              BIT04== 20
          000010              BIT03== 10
          000004              BIT02== 4
          000002              BIT01== 2
          000001              BIT00== 1
```

```
001000                   ;
000400                   BIT9==  BIT09
000200                   BIT8==  BIT08
000100                   BIT7==  BIT07
000040                   BIT6==  BIT06
000020                   BIT5==  BIT05
000010                   BIT4==  BIT04
000004                   BIT3==  BIT03
000002                   BIT2==  BIT02
000001                   BIT1==  BIT01
                         BIT0==  BIT00
                         ;
                         ; EVENT FLAG DEFINITIONS
                         ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                         ;
                         ;                            ;  BIT POSITION IN SECOND STATUS WORD
000040                   EF.START==      32.          ; (100000) START COMMAND WAS ISSUED
000037                   EF.RESTART==    31.          ; (040000) RESTART COMMAND WAS ISSUED
000036                   EF.CONTINUE==   30.          ; (020000) CONTINUE COMMAND WAS ISSUED
000035                   EF.NEW==        29.          ; (010000) A NEW PASS HAS BEEN STARTED
000034                   EF.PWR==        28.          ; (004000) A POWER-FAIL/POWER-UP OCCURRED
                         ;
                         ;
                         ; PRIORITY LEVEL DEFINITIONS
                         ;
000340                   PRIO7== 340
000300                   PRIO6== 300
000240                   PRIO5== 240
000200                   PRIO4== 200
000140                   PRIO3== 140
000100                   PRIO2== 100
000040                   PRIO1== 40
000000                   PRIO0== 0
                         ;
                         ;OPERATOR FLAG BITS
                         ;
000004                   EVL==       4
000010                   LOT==      10
000020                   ADR==      20
000040                   IDU==      40
000100                   ISR==     100
000200                   UAM==     200
000400                   BOE==     400
001000                   PNT==    1000
002000                   PRI==    2000
004000                   IXE==    4000
010000                   IBE==   10000
020000                   IER==   20000
040000                   LOE==   40000
100000                   HOE==  100000
```

1372

```
1374
1375                              ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1376                              ;
1377                              ;                VDHC.GDT
1378                              ;
1379                              ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
1380
1381
1382
1383                              .SBTTL  GLOBAL DATA SECTION
1384
1385                              ;++
1386                              ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1387                              ; IN MORE THAN ONE TEST.
1388                              ;--
1389
1390                              ;****************************************************************
1391                              ;       Unit Varible Area
1392                              ;****************************************************************
1393
1394 002170    000300                    RXVECA:: .WORD   300         ;RX VECTOR ADDRESS.
1395 002172    000304                    TXVECA:: .WORD   304         ;TX VECTOR ADDRESS.
1396 002174    000377                    ACTLNS:: .WORD   377         ;ACTIVE LINE BIT MAP.
1397 0C2176    000                       LOPBCK:: .BYTE   0           ;LOOPBACK MODE
1398 002177    004                       BRLEVL:: .BYTE   4           ;INTERRUPT BUS REQUEST LEVEL
1399 002200    000000                    UNITN::  .WORD   0           ;UNIT NUMBER.
1400
1401                              ;****************************************************************
1402                              ;       Device Register Address Table
1403                              ;****************************************************************
1404 002202                             DRADRT::
1405 002202    160000                    CSRA::   .WORD   160000      ;DHV11-M CSR ADDRESS
1406 002204    160002            TXCHA:: RBUFA::  .WORD   160002      ;DHV11-M RECEIVE/TRANSMIT BUFFER ADDRESS
1407 002206    160004                    LPRA::   .WORD   160004      ;DHV11-M LINE PARAMETER REGISTER ADDRESS
1408 002210    160006                    STATA::  .WORD   160006      ;DHV11-M          STATUS REGISTER ADDRESS
1409 002212    160010                    LNCTRA:: .WORD   160010      ;DHV11-M LINE CONTROL REGISTER ADDRESS
1410 002214    160012                    TXAD1A:: .WORD   160012      ;DHV11-M TRANSMIT BUFFER 1 REGISTER ADDRESS
1411 002216    160014                    TXAD2A:: .WORD   160014      ;DHV11-M TRANSMIT BUFFER 2 REGISTER ADDRESS
1412 002220    160016                    TXBFCA:: .WORD   160016      ;DHV11-M TRANSMIT BUFFER COUNT REGISTER ADDRESS
1413
1414                              ;****************************************************************
1415                              ;       Assorted global variables:
1416                              ;****************************************************************
1417 002222    000000                    CTRLCF:: .WORD   0           ;STORAGE FOR THE CONTROL-C FLAG.
1418 002224    000001                    TSTNUM:: .WORD   1           ;STORAGE FOR THE TEST NUMBER.
1419 002226    000000                    IBM::    .WORD   0           ;INACTIVE TX/RX BITS MASK.
1420 002230    031463                    LGRP1M:: .WORD   31463       ;BIT MAP OF LINES IN LINE GROUP I.
1421 002232    146314                    LGRP2M:: .WORD   146314      ;BIT MAP OF LINES IN LINE GROUP II.
1422 002234    000000                    IESTAT:: .WORD   0           ;STORAGE FOR STATES OF THE DUT INT ENABLE BITS.
1423 002236    000000                    PASCNT:: .WORD   0           ;STO'G FOR PASS COUNT USED IN ROM VERSION# TST.
1424 002240    000000                    WORD1::  .WORD   0           ;LOCATION FOR PASSING INDIRECT PARAMETERS.
1425 002242    000000                    RXTOUT:: .WORD   0           ;TIME-OUT VALUE FOR WAITING FOR LAST RX CHAR.
1426 002244    000000                    SAVTEN:: .WORD   0           ;STORAGE FOR TX.ENABLE STATES. (TXROFF, TXRON).
1427 002246    000000                    SAVPRI:: .WORD   0           ;STO'G FOR PROCESSOR PRIORITY. (TXROFF, TXRON).
1428 002250    000000                    TXENBM:: .WORD   0           ;STORAGE FOR TX.ENABLE STATES. (BUFFER MGM'NT).
1429 002252    000000                    TXINTF:: .WORD   0           ;STORAGE FOR TRANSMIT INTERRUPT FLAGS.
1430 002254    000000                    TP4VEC:: .WORD   0           ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
```

```
1431 002256 000000              TP4FLG:: .WORD  0        ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
1432 002260 177777              BITLNG:: .WORD  -1       ;NUMBER OF BITS HOST USES TO DEFINE A UNIQUE
1433                                                     ;ADDR. -1= 16 BITS, 0= 18 BITS, 1= 22 BITS.
1434 002262 000000              FFREM::  .WORD  0        ;STO'G FOR ADR OF FIRST FREE WORD AFTER THE DIAG'TIC
1435 002264 000000              DMTSTA:: .WORD  0        ;STO'G FOR DMA TEST ADDRESS (IN PAR FORM).
1436 002266 000000              GMANWD:: .WORD  0        ;WORD FOR GMANxx CALL RETURN PARAMETERS.
1437 002270 000000              PMSFLG:: .WORD  0        ;FLAG INDICATING WHETHER TO PRINT MODEM STATUS.
1438
1439                     ;******************************************************************************
1440                     ;       Line Time Clock variables and storage.
1441                     ;******************************************************************************
1442 002272 177546              CLKCSR:: .WORD  177546   ;CSR ADDRESS OF THE LTC.
1443 002274 000300              CLKBRL:: .WORD  PRIO6    ;INTERRUPT PRIORITY LEVEL OF THE LTC.
1444 002276 000100              CLKVEC:: .WORD  100      ;INTERRUPT VECTOR ADDRESS OF THE LTC.
1445 002300 000074              CLKHRZ:: .WORD  60.      ;INTERRUPT FREQUENCY OF THE LTC.
1446 002302 000000              TIMER1:: .WORD  0        ;HARDWARE CLOCK COUNTER #1.
1447 002304 000000              TIMER2:: .WORD  0        ;HARDWARE CLOCK COUNTER #2.
1448 002306 000170              TIMER3:: .WORD  120.     ;HARDWARE BREAK COUNTER LOCATION.
1449 002310 000170              BCOUNT:: .WORD  120.     ;BREAK COUNT VALUE IN CLOCK TICKS.
1450 002312 000021              MSTICK:: .WORD  17.      ;NUMBER OF MILLI-SECONDS PER LTC TICK.
1451 002314 000062              MSLCNT:: .WORD  62       ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.
1452
1453                     ;******************************************************************************
1454                     ;       Memmory Management Variables and Flags.
1455                     ;******************************************************************************
1456 002316 177572              MMSRO::  .WORD  177572   ;ADDRESS OF MEM MGT STATUS REGISTER #0.
1457 002320 172516              MMSR3::  .WORD  172516   ;ADDRESS OF MEM MGT STATUS REGISTER #3.
1458 002322 000000              MMPRES:: .WORD  0        ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
1459 002324 000000              MMENAB:: .WORD  0        ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).
1460
1461 002326              PARATB::                        ;BASE OF MEM MGT PAR ADDRESS TABLE.
1462 002326 172340              PAROA:: .WORD   172340   ;ADDRESS OF MEM MGT PAR #0.
1463 002330 172342              PAR1A:: .WORD   172342   ;ADDRESS OF MEM MGT PAR #1.
1464 002332 172344              PAR2A:: .WORD   172344   ;ADDRESS OF MEM MGT PAR #2.
1465 002334 172346              PAR3A:: .WORD   172346   ;ADDRESS OF MEM MGT PAR #3.
1466 002336 172350              PAR4A:: .WORD   172350   ;ADDRESS OF MEM MGT PAR #4.
1467 002340 172352              PAR5A:: .WORD   172352   ;ADDRESS OF MEM MGT PAR #5.
1468 002342 172354              PAR6A:: .WORD   172354   ;ADDRESS OF MEM MGT PAR #6.
1469 002344 172356              PAR7A:: .WORD   172356   ;ADDRESS OF MEM MGT PAR #7.
1470 002346              PARATE::                        ;END OF PAR ADDRESS TABLE.
1471
1472 002346              PDRATB::                        ;BASE OF MEM MGT PDR ADDRESS TABLE.
1473 002346 172300              PDROA:: .WORD   172300   ;ADDRESS OF MEM MGT PDR #0.
1474 002350 172302              PDR1A:: .WORD   172302   ;ADDRESS OF MEM MGT PDR #1.
1475 002352 172304              PDR2A:: .WORD   172304   ;ADDRESS OF MEM MGT PDR #2.
1476 002354 172306              PDR3A:: .WORD   172306   ;ADDRESS OF MEM MGT PDR #3.
1477 002356 172310              PDR4A:: .WORD   172310   ;ADDRESS OF MEM MGT PDR #4.
1478 002360 172312              PDR5A:: .WORD   172312   ;ADDRESS OF MEM MGT PDR #5.
1479 002362 172314              PDR6A:: .WORD   172314   ;ADDRESS OF MEM MGT PDR #6.
1480 002364 172316              PDR7A:: .WCRD   172316   ;ADDRESS OF MEM MGT PDR #7.
1481 002366              PDRATE::                        ;END OF MEM MGT PDR ADDRESS TABLE.
1482
1483                     ;******************************************************************************
1484                     ;       Table of words with corresponding bit set for generation of bit maps.
1485                     ;******************************************************************************
1486 002366 000001              BITTBL:: .WORD  1        ;BIT 0 SET.
1487 002370 000002                       .WORD  2        ;BIT 1 SET.
```

```
1488 002372  000004                      .WORD    4               ;BIT 2 SET.
1489 002374  000010                      .WORD    10              ;BIT 3 SET.
1490 002376  000020                      .WORD    20              ;BIT 4 SET.
1491 002400  000040                      .WORD    40              ;BIT 5 SET.
1492 002402  000100                      .WORD    100             ;BIT 6 SET.
1493 002404  000200                      .WORD    200             ;BIT 7 SET.
1494 002406  000400                      .WORD    400             ;BIT 8 SET.
1495 002410  001000                      .WORD    1000            ;BIT 9 SET.
1496 002412  002000                      .WORD    2000            ;BIT 10 SET.
1497 002414  004000                      .WORD    4000            ;BIT 11 SET.
1498 002416  010000                      .WORD    10000           ;BIT 12 SET.
1499 002420  020000                      .WORD    20000           ;BIT 13 SET.
1500 002422  040000                      .WORD    40000           ;BIT 14 SET.
1501 002424  100000                      .WORD    100000          ;BIT 15 SET.
1502
1503                      ;****************************************************************************
1504                      ;*       Table of DUT Baudrates
1505                      ;****************************************************************************
1506 002426              BRTBLB::                                 ;BASE OF DUT BAUD RATE TABLE.
1507 002426  000062                      .WORD    50.             ;BAUD RATE ENTRY FOR CODE 0.
1508 002430  000113                      .WORD    75.             ;BAUD RATE ENTRY FOR CODE 1.
1509 002432  000156                      .WORD    110.            ;BAUD RATE ENTRY FOR CODE 2.
1510 002434  000206                      .WORD    134.            ;BAUD RATE ENTRY FOR CODE 3.
1511 0C2436  000226                      .WORD    150.            ;BAUD RATE ENTRY FOR CODE 4.
1512 002440  000454                      .WORD    300.            ;BAUD RATE ENTRY FOR CODE 5.
1513 002442  001130                      .WORD    600.            ;BAUD RATE ENTRY FOR CODE 6.
1514 002444  002260                      .WORD    1200.           ;BAUD RATE ENTRY FOR CODE 7.
1515 002446  003410                      .WORD    1800.           ;BAUD RATE ENTRY FOR CODE 8.
1516 002450  003720                      .WORD    2000.           ;BAUD RATE ENTRY FOR CODE 9.
1517 002452  004540                      .WORD    2400.           ;BAUD RATE ENTRY FOR CODE 10.
1518 002454  011300                      .WORD    4800.           ;BAUD RATE ENTRY FOR CODE 11.
1519 002456  016040                      .WORD    7200.           ;BAUD RATE ENTRY FOR CODE 12.
1520 002460  022600                      .WORD    9600.           ;BAUD RATE ENTRY FOR CODE 13.
1521 002462  045400                      .WORD    19200.          ;BAUD RATE ENTRY FOR CODE 14.
1522 002464  113000                      .WORD    38400.          ;BAUD RATE ENTRY FOR CODE 15.
1523 002466              BRTBLE::                                 ;LABEL AFTER END OF DUT BAUDRATE TABLE.
1524                      ;****************************************************************************
1525                      ;*       GPR Save Areas Zero and One.
1526                      ;****************************************************************************
1527 002466              GPRS0B::                                 ;BASE OF GPR SAVE AREA NUMBER ZERO.
1528 002466  000000                      .WORD    0               ;WORD 1, STORAGE FOR R1.
1529 002470  000000                      .WORD    0               ;WORD 2, STORAGE FOR R2.
1530 002472  000000                      .WORD    0               ;WORD 3, STORAGE FOR R3.
1531 002474  000000                      .WORD    0               ;WORD 4, STORAGE FOR R4.
1532 002476  000000                      .WORD    0               ;WORD 5, STORAGE FOR R5.
1533                      ;****************************************************************************
1534                      ;*       Transmission and Reception Variables, Pointers, and Flags.
1535                      ;****************************************************************************
1536 002500  000000      CHRTOT:: .WORD  0                        ;TOTAL RECEIVED CHARACTER COUNTER.
1537 002502  000000      ERSMRF:: .WORD  0                        ;"PRINT ERROR SUMMARY" FLAGS.
1538 002504  000000      TXDONF:: .WORD  0                        ;TRANSMISSION DONE FLAGS.
1539 002506  000000      RXDONF:: .WORD  0                        ;RECEPTION DONE FLAGS.
1540 002510  000000      TXDBLF:: .WORD  0                        ;"TX HAS BEEN DISABLED" FLAG.
1541                      ;****************************************************************************
1542                      ;       Storage area for the BMP code queue.
1543                      ;****************************************************************************
1544 002512  000000      BMPCQP::        .WORD    0               ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
```

```
1545 002514          BMPCQB::        .BLKW   64.     ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
1546 002714          BMPCQE::                        ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1547                 ;*********************************************************************************
1548                 ;*      Receive Buffer and Associated Variables.
1549                 ;*********************************************************************************
1550 002714 000000   RXBOPT:: .WORD  0               ;RX BUFFER OUTPUT POINTER.
1551 002716 000000   RXBIPT:: .WORD  0               ;RX BUFFER INPUT POINTER.
1552 002720 000000   RXBCNT:: .WORD  0               ;COUNT OF NUMBER OF CHARS IN RX BUFFER.
1553 002722          RXBSTA::                        ;LABEL AT BEGINNING OF THE RX BUFFER.
1554 002722                  .BLKW   RXBFUL          ;LEAVE ENOUGH ROOM FOR A FULL BUFFER.
1555 003122 000000   RXBEND:: .WORD  0               ;LABEL AFTER END OF RX BUFFER.
1556                 ;*********************************************************************************
1557                 ;*      TX/RX Control Block.
1558                 ;*********************************************************************************
1559 003124          CBB::                           ;BASE OF TX/RX CONTROL BLOCK.
1560 003124 000000   CBLPRA:: .WORD  0               ;LINE PARAMETER REGISTER CONTENTS.
1561 003126 000000   CBLNCA:: .WORD  0               ;LINE CONTROL REGISTER CONTENTS.
1562 003130 000000   CBDPAA:: .WORD  0               ;START ADDRESS OF DATA PATTERN.
1563 003132 000000   CBDPLA:: .WORD  0               ;LENGTH OF DATA PATTERN.
1564 003134 000000   CBDPNA:: .WORD  0               ;NUMBER OF REPEAT TRANSMISSIONS OF THE DATA PATTERN.
1565 003136 000000   CBMAPA:: .WORD  0               ;BIT MAP OF LINES TO INITIALISE.
1566 003140 000000   CBLPBA:: .WORD  0               ;LOOPBACK MODE (AS IN LOPBCK).
1567 003142 000000   CBOFSA:: .WORD  0               ;AMOUNT OF OFFSET BETWEEN EACH TX START.
1568                 ;*********************************************************************************
1569                 ;*      Transmission and Reception Tables of Pointers and Counters.
1570                 ;*********************************************************************************
1571 003144          DPENDB:: .BLKW  16.             ;TABLE OF END ADDRESSES OF DATA PATTERNS.
1572 003204          DPLENB:: .BLKW  16.             ;TABLE OF LENGTH OF DATA PATTERNS FOR LINES.
1573 003244          EXCNTB:: .BLKW  16.             ;EXTRA RECEIVED CHARACTER COUNTERS TABLE.
1574 003304          ERCNTB:: .BLKW  16.             ;CHARACTER RECEIVE ERROR COUNTERS TABLE.
1575 003344          TXPTRB:: .BLKW  16.             ;TRANSMISSION DATA POINTERS TABLE.
1576 003404          RXPTRB:: .BLKW  16.             ;RECEPTION DATA POINTERS TABLE.
1577 003444          CHCNTB:: .BLKW  16.             ;NUMBER OF CHARACTERS TO BE TXED AND RXED.
1578 003504          TXCNTB:: .BLKW  16.             ;TRANSMISSION CHARACTER COUNTERS TABLE.
1579 003544          RXCNTB:: .BLKW  16.             ;RECEPTION CHARACTER COUNTERS TABLE.
1580                 ;*********************************************************************************
1581                 ;       General table and buffer area--513 words.
1582                 ;*********************************************************************************
1583 003604          BUFBAS::                        ;BASE OF MEMORY BUFFER.
1584 003604          ERLTBL:: .BLKW  128.            ;FIRST HALF OF GENERAL TABLE OR BUFFER.
1585 004204          BUFMID:: .BLKW  64.             ;SECOND HALF OF GENERAL TABLE OR BUFFER.
1586 004404          BUF3QT:: .BLKW  64.             ;LAST QUARTER OF THE BUFFER AREA.
1587 004604          BUFEND::                        ;END OF GENERAL PURPOSE MEMORY BUFFER.
1588 004604          ENDETB:: .BLKW  16.             ;BUFFER OVERFLOW SPACE.
1589                 ;*********************************************************************************
1590                 ;       Table of Data Pattern Resync Queues.
1591                 ;*********************************************************************************
1592 004644          DPRSQB::                        ;DATA PATTERN RESYNC QUEUES TABLE BASE.
1593 004644                  .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 0.
1594 004654                  .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 1.
1595 004664                  .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 2.
1596 004674                  .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 3.
1597 004704                  .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 4.
1598 004714                  .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 5.
1599 004724                  .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 6.
1600 004734                  .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 7.
1601 004744                  .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 8.
```

```
1602 004754                          .BLKW   4          ;DATA PATTERN RESYNC QUEUE FOR LINE 9.
1603 004764                          .BLKW   4          ;DATA PATTERN RESYNC QUEUE FOR LINE 10.
1604 004774                          .BLKW   4          ;DATA PATTERN RESYNC QUEUE FOR LINE 11.
1605 005004                          .BLKW   4          ;DATA PATTERN RESYNC QUEUE FOR LINE 12.
1606 005014                          .BLKW   4          ;DATA PATTERN RESYNC QUEUE FOR LINE 13.
1607 005024                          .BLKW   4          ;DATA PATTERN RESYNC QUEUE FOR LINE 14.
1608 005034                          .BLKW   4          ;DATA PATTERN RESYNC QUEUE FOR LINE 15.
1609 005044                 DPRSQE::                    ;END OF DATA PATTERN RESYNC QUEUES TABLE.
1610                        ;**********************************************************************
1611                        ;        Single Character Mode LPR Field Tables.
1612                        ;**********************************************************************
1613 005044                 SCBCTB::                    ;BASE OF NUMBER OF BITS PER CHAR FIELDS TABLE.
1614 005044 000000                   .WORD   0          ;5 BITS/CHAR LPR FIELD.
1615 005046 000010                   .WORD   10         ;6 BITS/CHAR LPR FIELD.
1616 005050 000020                   .WORD   20         ;7 BITS/CHAR LPR FIELD.
1617 005052 000030                   .WORD   30         ;8 BITS/CHAR LPR FIELD.
1618 005054                 SCBCTE::                    ;END OF NUMBER OF BITS/CHAR FIELDS TABLE.
1619 005054                 SCBRTB::                    ;BASE OF BAUDRATE FIELDS TABLE.
1620 005054 052400                   .WORD   52400      ;300 BAUD LPR FIELDS.(changed from 50 baud)###
1621 005056 073400                   .WORD   73400      ;1.2K BAUD LPR FIELDS.
1622 005060 177400                   .WORD   177400     ;38.4K BAUD LPR FIELDS.
1623 005062                 SCBRTE::                    ;END OF BAUDRATE FIELDS TABLE.
1624 005062                 SCNSTB::                    ;BASE OF NUMBER OF STOP BITS FIELDS TABLE.
1625 005062 000000                   .WORD   0          ;1 STOP BIT LPR FIELD.
1626 005064 000200                   .WORD   200        ;2 STOP BITS LPR FIELD.
1627 005066                 SCNSTE::                    ;END OF BAUDRATE FIELDS TABLE.
1628 005066                 SCTPTB::                    ;BASE OF TYPE OF PARITY FIELDS TABLE.
1629 005066 000000                   .WORD   0          ;NO PARITY LPR FIELD.
1630 005070 000040                   .WORD   40         ;ODD PARITY LPR FIELD.
1631 005072 000140                   .WORD   140        ;EVEN PARITY LPR FIELD.
1632 005074                 SCTPTE::                    ;END OF TYPE OF PARITY FIELDS TABLE.
1633
1634                        ;**********************************************************************
1635                        ;        DMA Mode LPR Field Tables.
1636                        ;    Set up with specified baudrates, 1 stop bit, odd parity, 8 bits/char.
1637                        ;**********************************************************************
1638 005074                 DLPRTB::                    ;BASE OF DMA TEST LPR FIELDS TABLE.
1639 005074 156470                   .WORD   156470     ;9.6K BAUD.
1640 005076 167070                   .WORD   167070     ;19.2K BAUD.
1641 005100 177470                   .WORD   177470     ;38.4K BAUD.
1642 005102                 DLPRTE::                    ;END OF DMA TEST LPR FIELDS TABLE.
1643                        ;**********************************************************************
1644                        ;        SPLIT SPEED LPR PARAMETER TABLE.
1645                        ;**********************************************************************
1646 005102                 SPLPRB::                    ;BASE OF SPLIT SPEED LPR TABLE.
1647 005102 170070                   .WORD   170070     ;TX: 38.4K, RX: 50 BAUD, 1 STOP ODD PAR 8 BITS.
1648 005104 007470                   .WORD   7470       ;TX: 50, RX: 38.4K BAUD, 1 STOP ODD PAR 8 BITS.
1649 005106 000001                   .WORD   1          ;NUMBER OF REPEAT TRANSMISSIONS AT 50 BAUD.
1650 005110 000120                   .WORD   80.        ;NUMBER OF REPEAT TRANSMISSIONS AT 38.4K BAUD.
1651 005112 070470                   .WORD   70470      ;TX: 1200, RX: 75 BAUD, 1 STOP ODD PAR 8 BITS.
1652 005114 013470                   .WORD   13470      ;TX: 75, RX: 1200 BAUD, 1 STOP ODD PAR 8 BITS.
1653 005116 000001                   .WORD   1          ;NUMBER OF REPEAT TRANSMISSIONS AT 75 BAUD.
1654 005120 000016                   .WORD   16         ;NUMBER OF REPEAT TRANSMISSIONS AT 1200 BAUD.
1655 005122 115070                   .WORD   115070     ;TX: 2000, RX:2400 BAUD, 1 STOP ODD PAR 8 BITS.
1656 005124 124470                   .WORD   124470     ;TX: 2400, RX:2000 BAUD, 1 STOP ODD PAR 8 BITS.
1657 005126 000001                   .WORD   1          ;NUMBER OF REPEAT TRANSMISSIONS AT 2400 BAUD.
1658 005130 000002                   .WORD   2          ;NUMBER OF REPEAT TRANSMISSIONS AT 2000 BAUD.
```

```
1659 005132              SPLPRE::                                    ;END OF SPLIT SPEED LPR TABLE.
1660                     ;************************************************************************
1661                     ;       SINGLE CHARACTER DATA PATTERN TABLE.
1662                     ;************************************************************************
1663 005132    000       SDPBAS::.BYTE    0          ;START OF SINGLE CHARACTER DATA PATTERN TABLE.
1664 005133    001               .BYTE    1
1665 005134    010               .BYTE    10
1666 005135    017               .BYTE    17
1667 005136    063               .BYTE    63
1668 005137    074               .BYTE    74
1669 005140    125               .BYTE    125
1670 005141    177               .BYTE    177
1671 005142    200               .BYTE    200
1672 005143    252               .BYTE    252
1673 005144    303               .BYTE    303
1674 005145    314               .BYTE    314
1675 005146    360               .BYTE    360
1676 005147    367               .BYTE    367
1677 005150    376               .BYTE    376
1678 005151    377               .BYTE    377
1679 005152              SDPEND::                    ;END OF SINGLE CHARACTER DATA PATTERN TABLE.
1680 005152    000               .BYTE    0          ;START OF FIRST SHORT DATA PATTERN OVERFLOW AREA.
1681 005153    001               .BYTE    1
1682 0C5154    010               .BYTE    10
1683 005155    017               .BYTE    17
1684
1685                     ;************************************************************************
1686                     ;       SINGLE CHARACTER DATA PATTERN TABLE NUMBER TWO.
1687                     ;************************************************************************
1688 005156    125       SDP2B:: .BYTE    125        ;START OF SECOND SHORT DATA PATTERN.
1689 005157    252               .BYTE    252
1690 005160    124               .BYTE    124
1691 005161    253               .BYTE    253
1692 005162    122               .BYTE    122
1693 005163    255               .BYTE    255
1694 005164    112               .BYTE    112
1695 005165    265               .BYTE    265
1696 005166    052               .BYTE    52
1697 005167    325               .BYTE    325
1698 005170    152               .BYTE    152
1699 005171    225               .BYTE    225
1700 005172    132               .BYTE    132
1701 005173    245               .BYTE    245
1702 005174    126               .BYTE    126
1703 005175    251               .BYTE    251
1704 005176              SDP2E::                      ;END OF SECOND SHORT DATA PATTERN.
1705 005176    125               .BYTE    125        ;START OF SECOND SHORT DATA PATTERN OVERFLOW AREA.
1706 005177    252               .BYTE    252
1707 005200    124               .BYTE    124
1708 005201    253               .BYTE    253
1709 005202    122               .BYTE    122
1710 005203    255               .BYTE    255
1711 005204    112               .BYTE    112
1712 005205    265               .BYTE    265
1713 005206    052               .BYTE    52
1714 005207    325               .BYTE    325
1715 005210    152               .BYTE    152
```

```
1716 005211    225              .BYTE    225
1717 005212    132              .BYTE    132
1718 005213    245              .BYTE    245
1719 005214    126              .BYTE    126
1720 005215    251              .BYTE    251
1721                   ;********************************************************************
1722                   ;         Single character safe proportional delay table.
1723                   ;********************************************************************
1724 005216    372     PROTBL::.BYTE    250.        ;DELAY IN MILLI SECONDS AT 50 BAUD
1725 005217    252              .BYTE    170.        ;DELAY IN MILLI SECONDS AT 75 BAUD
1726 005220    167              .BYTE    119.        ;DELAY IN MILLI SECONDS AT 110 BAUD
1727 005221    143              .BYTE    99.         ;DELAY IN MILLI SECONDS AT 134.5 BAUD
1728 005222    132              .BYTE    90.         ;DELAY IN MILLI SECONDS AT 150 BAUD
1729 005223    062              .BYTE    50.         ;DELAY IN MILLI SECONDS AT 300 BAUD
1730 005224    036              .BYTE    30.         ;DELAY IN MILLI SECONDS AT 600 BAUD
1731 005225    024              .BYTE    20.         ;DELAY IN MILLI SECONDS AT 1200 BAUD
1732 005226    021              .BYTE    17.         ;DELAY IN MILLI SECONDS AT 1800 BAUD
1733 005227    020              .BYTE    16.         ;DELAY IN MILLI SECONDS AT 2000 BAUD
1734 005230    017              .BYTE    15.         ;DELAY IN MILLI SECONDS AT 2400 BAUD
1735 005231    015              .BYTE    13.         ;DELAY IN MILLI SECONDS AT 4800 BAUD
1736 005232    014              .BYTE    12.         ;DELAY IN MILLI SECONDS AT 7200 BAUD
1737 005233    014              .BYTE    12.         ;DELAY IN MILLI SECONDS AT 9600 BAUD
1738 005234    013              .BYTE    11.         ;DELAY IN MILLI SECONDS AT 19200 BAUD
1739 0C5235    012              .BYTE    10.         ;DELAY IN MILLI SECONDS AT 38400 BAUD
1740                            .EVEN
1741                   ;********************************************************************
1742                   ;* Table for storage of RX/TX line number associations.
1743                   ;* The associations are stored as line number times 2 for use as offsets
1744                   ;*    when accessing a table of words.
1745                   ;*    NOTE: Do not write a non-zero value into the upper byte of any entry.
1746                   ;********************************************************************
1747 005236          TXRXLB::                        ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1748 005236 000000              .WORD    0           ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
1749 005240 000002              .WORD    2.          ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
1750 005242 000004              .WORD    4.          ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
1751 005244 000006              .WORD    6.          ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
1752 005246 000010              .WORD    8.          ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
1753 005250 000012              .WORD    10.         ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
1754 005252 000014              .WORD    12.         ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
1755 005254 000016              .WORD    14.         ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
1756 005256 000020              .WORD    16.         ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
1757 005260 000022              .WORD    18.         ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
1758 005262 000024              .WORD    20.         ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
1759 005264 000026              .WORD    22.         ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
1760 005266 000030              .WORD    24.         ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
1761 005270 000032              .WORD    26.         ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
1762 005272 000034              .WORD    28.         ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
1763 005274 000036              .WORD    30.         ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
1764 005276          TXRXLE::                        ;END OF TX/RX LINE NUMBER ASSOCIATION TABLE.
1765                            .EVEN                 ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1766                   ;********************************************************************
1767                   ;* Table of TX/RX line number associations in staggered loopback.
1768                   ;* The associations are stored as line number times 2 for use as offsets
1769                   ;*    when accessing a table of words.
1770                   ;* This is a table of data for reading only.  Use to load the above table.
1771                   ;*    NOTE: Must convert from BYTEs to WORDs when loading above table.
1772                   ;********************************************************************
```

```
1773 005276              STGTRB::                      ;BASE OF STAGGERED TX/RX LINE NUMBER TABLE.
1774 005276      004              .BYTE  4.            ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
1775 005277      006              .BYTE  6.            ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
1776 005300      000              .BYTE  0             ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
1777 005301      002              .BYTE  2.            ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
1778 005302      014              .BYTE  12.           ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
1779 005303      016              .BYTE  14.           ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
1780 005304      010              .BYTE  8.            ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
1781 005305      012              .BYTE  10.           ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
1782 005306      024              .BYTE  20.           ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
1783 005307      026              .BYTE  22.           ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
1784 005310      020              .BYTE  16.           ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
1785 005311      022              .BYTE  18.           ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
1786 005312      034              .BYTE  28.           ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
1787 005313      036              .BYTE  30.           ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
1788 005314      030              .BYTE  24.           ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
1789 005315      032              .BYTE  26.           ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
1790                              .EVEN                ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1803 005316              ERRTBL
     005316                                                       L$ERRTBL::
     005316 000000      ERRTYP::         .WORD  0
     005320 000000      ERRNBR::         .WORD  0
     005322 000000      ERRMSG::         .WORD  0
     0C5324 000000      ERRBLK::         .WORD  0
1804
1805                                     .EVEN
```

```
1807                          .SBTTL  GPR HANDLING ROUTINES FOR SUBROUTINE CALLS.
1808            ;****************************************************************************
1809            ;*          There are 4 routines and macro definitions used for the handling of
1810            ;*          GPR values during subroutine calls within this program.  The four
1811            ;*          routines/macro calls have the following names:
1812            ;*
1813            ;*          SAVE - Macro definition used at the beginning of a subroutine to
1814            ;*                  save the GPR contents for later restoration.
1815            ;*          PASS - Macro definition used at the end of a subroutine to restore
1816            ;*                  the previously saved GPR contents and to leave the contents
1817            ;*                  of the specified GPR(s) intact (NOT restored).
1818            ;*          PREG05 - Subroutine which is called from the SAVE and PASS macro
1819            ;*                  expansions which actually performs the actions on the GPRs.
1820            ;*
1821            ;*          During a subroutine which uses these GPR save routines the values
1822            ;*          of the GPRs are stored on the stack in the following stack frame:
1823            ;*
1824            ;*                  SP       -> RET PC INTO PREG05 ROUTINE.
1825            ;*                  SP+2   -> GPR R0 CONTENTS.
1826            ;*                  SP+4   -> GPR R1 CONTENTS.
1827            ;*                  SP+6   -> GPR R2 CONTENTS.
1828            ;*                  SP+8   -> GPR R3 CONTENTS.
1829            ;*                  SP+10 -> GPR R4 CONTENTS.
1830            ;*                  SP+12 -> GPR R5 CONTENTS.
1831            ;*                  SP+14 -> RET PC INTO CALLER OF SUB'TNE WHICH CALLED PREG05.
1832            ;*
1833            ;*          Each level of sub'tne calling uses 8 words of stack overhead.
1834            ;*          The SAVE and PASS macros can also be used in "straight line code"
1835            ;*          to save and restore the GPR values.  In any case, after the
1836            ;*          issuing of a PASS call the GPRs will be restored to the values
1837            ;*          they had prior to the last SAVE call (except for the excepted,
1838            ;*          or passed intact, GPRs specified as parameters to the PASS call)
1839            ;*          and the SP will also be restored to its condition before the last
1840            ;*          SAVE call.  The programmer must be sure that the SP has the same
1841            ;*          value when the PASS macro is called as it had immediately after
1842            ;*          the SAVE macro was called.
1843            ;****************************************************************************
```

```
1845                                  .SBTTL GPR FRAME ACCESS EQUATES
1846                                  ;+++
1847                                  ;Equates that allow access to the stack frame.  These are the
1848                                  ;offsets into the stack for registers saved during the PREG05
1849                                  ;routine.
1850                                  ;---
1851
1852          000036                  LPCSLT==      36        ;Offset for last return PC.
1853          000016                  PCSLOT==      16        ;Offset for return PC.
1854          000014                  R5SLOT==      14        ;Offset for R5.
1855          000012                  R4SLOT==      12        ;Offset for R4.
1856          000010                  R3SLOT==      10        ;Offset for R3.
1857          000006                  R2SLOT==       6        ;Offset for R2.
1858          000004                  R1SLOT==       4        ;Offset for R1.
1859          000002                  R0SLOT==       2        ;Offset for R0.
```

```
1861                            .SBTTL  GLOBAL MACRO DEFINITION          - SAVE -
1862                            ;**********************************************************************
1863                            ;*      This macro is used at the beginning of a subroutine to save the
1864                            ;*      contents of the GPRs R0 thru R5.
1865                            ;*
1866                            ;* INPUTS:      SP - Unchanged since subroutine was entered
1867                            ;*              R5SLOT - Offset to stack slot for R5 (Equated to 14 Octal)
1868                            ;*
1869                            ;* OUTPUTS:     GPR save area on the stack is loaded with the contents of GPRs
1870                            ;*              TOP OF STACK - Loaded with the return address into PREG05
1871                            ;*
1872                            ;* CALLING SEQUENCE:    SAVE
1873                            ;*
1874                            ;* COMMENTS:    No arguments are allowed.
1875                            ;*              The PASS macro should be called to restore the GPR values.
1876                            ;*
1877                            ;* SUBORDINATE ROUTINES CALLED: PREG05.
1878                            ;**********************************************************************
1879
1880                            .MACRO  SAVE
1881                            .LIST
1882                                            JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
1883                            .NLIST
1884                            .ENDM   SAVE
```

```
1886                        .SBTTL  GLOBAL MACRO DEFINITION              - PASS -
1887                        ;******************************************************************************
1888                        ;*      This macro is used in conjunction with the SAVE macro.  It is
1889                        ;*      called at end of a subroutine to pass parameters in GPRs back to the
1890                        ;*      calling routine by altering the GPR save area on the stack and then
1891                        ;*      returning to PREG05 to restore the GPRs to their saved values.
1892                        ;*
1893                        ;* INPUTS:        Only allowed ARGUMENTS are "R0" thru "R5".
1894                        ;*                R0SLOT thru R5SLOT must be equated to their respective GPR save
1895                        ;*                    slot offsets before calling this macro.
1896                        ;*
1897                        ;* OUTPUTS:       The GPR values are put in their respective slots on the stack.
1898                        ;*
1899                        ;* CALLING SEQUENCE:    PASS    R0,R1,...
1900                        ;*
1901                        ;* COMMENTS:      Any combination of GPR arguments may be listed in any order.
1902                        ;*                For example, the following are legal:
1903                        ;*                     PASS    R1
1904                        ;*                     PASS    R4,R0,R2
1905                        ;*                The GPRs listed as arguments will be passed intact to the
1906                        ;*                calling routine, all other GPRs will be restored.
1907                        ;*                The SP must be at its original value when PASS is called.
1908                        ;*
1909                        ;*                The macro call
1910                        ;*                     PASS    R0,R3
1911                        ;*                expands into the following assembly code:
1912                        ;*                     MOV     R0,R0SLOT(SP)          ;PUT R0 IN STACK SLOT.
1913                        ;*                     MOV     R3,R3SLOT(SP)          ;PUT R3 IN STACK SLOT.
1914                        ;*                     JSR     PC,@(SP)+              ;RETURN TO PREG05 SUBRT.
1915                        ;*                In this example GPRs R1, R2, R4, and R5 will be restored to
1916                        ;*                their values contained in the stack frame and R0 and R3
1917                        ;*                will be left at their values prior to this PASS call.
1918                        ;*
1919                        ;* SUBORDINATE ROUTINES CALLED: (PREGRT - Label within PREG05, value on stack.)
1920                        ;******************************************************************************
1921
1922                        .MACRO  PASS    A,B,C,D,E,F
1923                        .IRP    X,<A,B,C,D,E,F>
1924                        .IF     NB,X
1925                        .LIST
1926                                MOV     X,X'SLOT(SP)           ;PUT X IN STACK SLOT.
1927                        .NLIST
1928                        .ENDC
1929                        .ENDM
1930                        .LIST
1931                                JSR     PC,@(SP)+              ;RETURN TO PREG05 SUBRT.
1932                        .NLIST
1933                        .ENDM   PASS
```

```
 1935                                          .SBTTL GLOBAL SUBROUTINE                  - PREG05 -
 1936                                          ;************************************************************************
 1937                                          ;*      Preserve Registers R0 through R5 for subroutine calls.
 1938                                          ;*
 1939                                          ;* INPUTS:       The return address back into the calling routine must be in
 1940                                          ;*               GPR R5.  (i.e.- Macros use "JSR  R5,PREG05".)
 1941                                          ;*
 1942                                          ;* OUTPUTS:      Registers R0 through R5 are saved on the stack.
 1943                                          ;*
 1944                                          ;*CALLING SEQUENCE:      SAVE              ;Macro expansion calls PREG05.
 1945                                          ;*                       [Subroutine code]...
 1946                                          ;*                       PASS              ;Macro expansion recalls PREG05.
 1947                                          ;*
 1948                                          ;*COMMENTS:      This routine is re-entrant.
 1949                                          ;*
 1950                                          ;*               Parameters may be passed out of a subroutine by modifying the
 1951                                          ;*               register save area on the stack.  Use the PASS GPRn macro
 1952                                          ;*               to return GPR values intact.
 1953                                          ;*               Use the RnSLOT offsets from the SP to pass other parameters.
 1954                                          ;*               [Example:      MOV    VALUE,R0SLOT(SP)            ]
 1955                                          ;*               Make sure the SP is at its original value when you do this.
 1956                                          ;*
 1957                                          ;*SUBORDINATE ROUTINES CALLED:  None.
 1958                                          ;************************************************************************
 1959
 1960 005326                                   PREG05:             ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
 1961 005326 010446                                    MOV    R4,-(SP)      ;SAVE R4
 1962 005330 010346                                    MOV    R3,-(SP)      ;SAVE R3
 1963 005332 010246                                    MOV    R2,-(SP)      ;SAVE R2
 1964 005334 010146                                    MOV    R1,-(SP)      ;SAVE R1
 1965 005336 010046                                    MOV    R0,-(SP)      ;SAVE R0
 1966 005340 010546                                    MOV    R5,-(SP)      ;PUSH RETURN PC ON TOP OF STACK
 1967 005342 016605 000014                             MOV    R5SLOT(SP),R5 ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS
 1968
 1969 005346 004736                                    JSR    PC,@(SP)+     ;Call the subroutine at the return address
 1970                                                                       ;from the PREG05 call, putting the present
 1971                                                                       ;PC on the stack as a return address into
 1972                                                                       ;this (PREG05) routine.
 1973
 1974
 1975                                          ;+++
 1976                                          ;The following code is executed when the calling routine does a
 1977                                          ;"return" [JSR  PC,@(SP)+] using the PC deposited on the stack above.
 1978                                          ;---
 1979 005350 012605                            PREGRT:: MOV    (SP)+,R5      ;Put return PC in R5.
 1980 005352 012600                                    MOV    (SP)+,R0      ;Restore R0.
 1981 005354 012601                                    MOV    (SP)+,R1      ;Restore R1.
 1982 005356 012602                                    MOV    (SP)+,R2      ;Restore R2.
 1983 005360 012603                                    MOV    (SP)+,R3      ;Restore R3.
 1984 005362 012604                                    MOV    (SP)+,R4      ;Restore R4.
 1985
 1986 005364 000205                                    RTS    R5            ;Return to the subroutine which called PREG05.
 1987                                                                         ;restoring R5 in the process.
```

```
 1989                                    .SBTTL  GLOBAL TEXT SECTION
 1991                             ;¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤
 1992                             ;
 1993                             ;                 FVTSKL1.P11
 1994                             ;
 1995                             ;¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤
 1997
 1998
 1999                             ;++
 2000                             ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
 2001                             ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
 2002                             ; MORE THAN ONE TEST.
 2003                             ;--
 2004
 2005                             ;
 2006                             ; NAMES OF DEVICES SUPPORTED BY PROGRAM
 2007                             ;
 2008 005366                             DEVTYP  <DHV11-M>
      005366                                                                  L$DVTYP::
      005366      104   110   126                                                    .ASCIZ  ¤DHV11-M¤
      005371      061   061   055
      005374      115   000
                                                                             .EVEN
 2009
 2015
 2016                             ; TEST DESCRIPTION
 2017                             ;
 2018 005376                             DESCRIPT           <DHV11-M FUNC TEST PART 3>
      005376                                                                  L$DESC::
      005376      104   110   126                                                    .ASCIZ  /DHV11-M FUNC TEST P
ART 3/
      005401      061   061   055
      005404      115   040   106
      005407      125   116   103
      005412      040   124   105
      005415      123   124   040
      005420      120   101   122
      005423      124   040   063
      005426      000
                                                                             .EVEN
 2019                                    .EVEN
 2020
 2027
```

```
2029
2030         ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2031         ;
2032         ;                      VDHC.FMT
2033         ;
2034         ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2035
2036
2037
2038         ;
2039         ; FORMAT STATEMENTS USED IN PRINT CALLS
2040         ;
2041
2052
2053
```

```
2062
2063                          ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2064                          ;
2065                          ;            VDHC.MSG
2066                          ;
2067                          ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2068
2069
2070           .NLIST BIN
2071           .SBTTL  GLOBAL MESSAGE AREA
2072                          ; ********** FORMAT STATEMENTS **********
2073 005430  MFUNIT:: .ASCIZ /%N%A TESTING UNIT :%D4%N/
2074 005461  EF0503:: .ASCIZ /%T%N/
2075 005466  EF1601:: .ASCIZ /%A  %T%A ABORTED %N/
2076 005512  EF1603:: .ASCIZ /%A    ACTUAL DATA:   %06%A (O).%N/
2077 005554  EF4401:: .ASCII /%N%A DMA ADDRESS TEST SUCCESSFUL, BITS 0 TO %D2%A (D) TESTED/
2078 005650           .ASCIZ / (%D2%A BITS).%N/
2079 005671  EF6201:: .ASCIZ \%A FRAMING/PARITY ERROR DETECTION AND REPORTING BAD ON LINES:%D2%A : %D2%N\
2080 006004  EF6202:: .ASCIZ /%A   CHAR RECEIVED WITH FRAMING ERROR BIT %T%A, SHOULD BE %T%N/
2081 006102  EF6203:: .ASCIZ /%A   CHAR RECEIVED WITH PARITY ERROR BIT %T%A, SHOULD BE %T%N/
2082 006177  EF7801:: .ASCIZ /%T%A ON LINE %D2%A DECIMAL.%N/
2083 006235  EF8901:: .ASCIZ /%N%N%AEXECUTING THE %T%N/
2084 0C6266  EF9001:: .ASCIZ /%A   UNEXPECTED %T%A FOUND IN RECEIVE CHAR FIFO:%N/
2085 006350  EF9002:: .ASCIZ /%A     CODE IS ASSOCIATED WITH LINE: %D2%N/
2086 006422  EF9003:: .ASCIZ /%A       CODE IS: %03%N/
2087 006451  EF9004:: .ASCIZ /%A     %T%A VALUE: %03%N/
2088 006501  EF9005:: .ASCIZ /%A     %T%A VALUE: NONE%N/
2089 006532  EF9006:: .ASCIZ /%A   %T%A %D2%N/
2090 006551  EF9007:: .ASCIZ /%A   CHARACTER RECEIVED WITH ERROR FLAG(S) SET ON LINE %D2%N/
2091 006645  EF9008:: .ASCIZ /%A       CHARACTER READ AS: %03%N/
2092 006704  EF9009:: .ASCIZ /%A      %T%A ERROR FLAG SET.%N/
2093 006743  EF9010:: .ASCIZ /%A      NUMBER OF ERRORS DETECTED ON LINE %D2%A IS %D5%N/
2094 007032  EF9012:: .ASCII /%A LINE%D2%A ONLY %T%D5%A BYTES OF%D5%A BYTE/
2095 007106           .ASCIZ / DATA PAT'N TX'D FROM LINE%D2%N/
2096 007146  EF9013:: .ASCIZ /%A   DATA PATTERN NOT COMPLETELY %T%N/
2097 007213  EF9019:: .ASCIZ /%A   %T%A %06%N/
2098 007232  EF9020:: .ASCIZ /%A      TOO FEW TX.ACTIONS GENERATED ON LINE %D2%N/
2099 007313  EF9101:: .ASCIZ /%N/
2100 007316  EF9103:: .ASCIZ /%A        ERROR CONDITION ON LINE %D2%N/
2101 007364  EF9301:: .ASCIZ /%A %T%D2%A, BMP CODE REPORTED :%03%N/
2102 007432  EF9302:: .ASCIZ /%A  OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)%N/
2103 007532  UBRFMT:: .ASCIZ /%D5%A IS NOT A SUPPORTED BAUDRATE, ENTER ANOTHER OR CTRL C.%N/
2104 007630  MSFMT1:: .ASCIZ /%AMODEM STATUS SIGNAL REPORT:%N/
2105 007670  MSFMT2:: .ASCIZ /%A  LINE #%D2%A:  DSR=%B1%A, RI=%B1%A, DCD=%B1%A, CTS=%B1%N/
2106 007764  EDPFMT:: .ASCII /%AMODEM LOOPBACK TEST STATUS REPORT:  /
2107 010032           .ASCIZ /PATTERN #%D5%A (D) COMPLETED.%N/
2108
2109                          ;************ MESSAGE AREA ************
2110 010072  EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
2111 010130  EM0509:: .ASCIZ /SET/
2112 010134  EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
2113 010217  EM4401:: .ASCIZ /DMA ADDRESS TEST /
2114 010241  EM4402:: .ASCIZ /NO SUITABLE ADDR FOUND,TEST ABANDONED/
2115 010307  EM4403:: .ASCIZ /**HOST FAILURE**;WRITE FAILED TO AN ADDR WHICH HAD BEEN READ/
2116 010404  EM4404:: .ASCIZ /NO ACTIVE LINES,TEST ABANDONED/
2117 010443  EM4405:: .ASCIZ /DMA_START BIT FOUND SET BEFORE DMA INITIATED,TEST ABANDONED/
2118 010537  EM4406:: .ASCIZ /TIME-OUT OCCCURED WAITING FOR DMA TO FINISH/
```

```
2119 010613   EM4407:: .ASCIZ /TOO FEW CHARACTERS FOUND IN THE RXFIFO,DMA FAILED/
2120 010675   EM4408:: .ASCIZ /TOO MANY BMP CODES FOUND IN RXFIFO/
2121 010740   EM4409:: .ASCIZ /BAD BITS BETWEEN BITS 0 AND /
2122 010775   EM4410:: .ASCIZ /RXFIFO FAILED TO PURGE/
2123 011024   EM4411:: .ASCIZ /**HOST FAILURE**WRITE ATTEMPT FAILED/
2124 011071   EM5303:: .ASCIZ /BMP CODE FOUND IN FIFO, TEST INVAILDATED/
2125 011142   EM6201:: .ASCIZ /FRAMING ERROR TEST /
2126 011166   EM6202:: .ASCIZ /CLEAR /
2127 011175   EM6301:: .ASCIZ /PARITY ERROR TEST /
2128 011220   EM8901:: .ASCIZ /MODEM LOOPBACK TEST /
2129 011245   EM9001:: .ASCIZ /SINGLE CHARACTER MODE TEST /
2130 011301   EM9003:: .ASCIZ /MODEM STATUS CODE/
2131 011323   EM9004:: .ASCIZ /SELFTEST CODE/
2132 011341   EM9006:: .ASCIZ /CHARACTER RECEIVED ON INACTIVE LINE, LINE:/
2133 011414   EM9007:: .ASCIZ /UNEXPECTED CHAR RECEIVED AFTER RX COMPLETE ON LINE/
2134 011477   EM9008:: .ASCIZ /RECEIVED CHAR MISCOMPARE AGAINST TX DATA ON LINE/
2135 011560   EM9009:: .ASCIZ /EXPECTED OR CORRECT/
2136 011604   EM9010:: .ASCIZ /ACTUAL OR MEASURED /
2137 011630   EM9011:: .ASCIZ /OVERRUN/
2138 011640   EM9012:: .ASCIZ /FRAMING/
2139 011650   EM9013:: .ASCIZ /PARITY/
2140 011657   EM9014:: .ASCIZ /SUMMARY REPORTS FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:/
2141 011753   EM9015:: .ASCIZ /TRANSMITTED/
2142 011767   EM9016:: .ASCIZ /RECV'D/
2143 011776   EM9017:: .ASCII /  FIFO WILL NOT PURGE (DATA.VALID STUCK SET),/
2144 012053            .ASCIZ / REMAINDER OF TEST SKIPPED./
2145 012107   EM9025:: .ASCIZ /MORE THAN TWICE THE EXPECTED NUMBER OF CHARACTERS RECEIVED./
2146 012203   EM9026:: .ASCIZ /     LPR CONTENTS: /
2147 012227   EM9027:: .ASCIZ /EXTRA CHAR RECEIVED WITHIN DATA PATTERN ON LINE/
2148 012307   EM9028:: .ASCIZ /SINGLE CHAR MISSING FROM RECEIVED DATA ON LINE/
2149 012366   EM9030:: .ASCIZ /%A  (NO TX COMPLETION INTERRUPTS RECEIVED)%N/
2150 012443   EM9101:: .ASCIZ /DMA TRANSMISSION MODE TEST /
2151 012477   EM9102:: .ASCIZ /DMA_START BIT SET AFTER RESET OR TX.ACTION ON LINE(S):/
2152 012566   EM9104:: .ASCIZ /  UNEXPECTED DATA FOUND IN FIFO FROM LINE: /
2153 012642   EM9201:: .ASCIZ /SPLIT SPEED TEST /
2154 012664   EM9301:: .ASCIZ /BMP CODE REPORT/
2155 012704   EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
2156 012734   EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
2157 013001   EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
2158 013055   EM9401:: .ASCIZ /KEYBOARD ECHO (DHV REMOTE LOOPBACK) TEST /
2159
2160 013127   BDRMSG:: .ASCIZ /MODEM BAUDRATE IN BPS:/
2161 013156   EMLMSG:: .ASCIZ /TYPE <CR> WHEN MODEM LINK ESTABLISHED:/
2162 013225   EXTMSG:: .ASCIZ /EXIT THE TEST (N = LOOP BACK TO SEND MORE DATA):/
2163 013306   NDPMSG:: .ASCII /NUMBER OF 256 BYTE PATTERNS TO SEND ON EACH SELECTED LINE/
2164 013377            .ASCIZ <15><12>/ (1-255, 0=SEND UNTIL tC):/
2165 013434   PMSMSG:: .ASCIZ /PRINT MODEM STATUS SIGNAL REPORT AFTER EACH PATTERN:/
2166 013521   TERMSG:: .ASCIZ /TYPE <CR> TO TERMINATE THE TEST:/
2167
2168            .EVEN
2169                         .LIST BIN
```

```
2171
2172            ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2173            ;
2174            ;                    FVTSKL2.P11
2175            ;
2176            ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2177
2178
2179
2180            .SBTTL   GLOBAL ERROR REPORT SECTION
2181
2182            ;++
2183            ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
2184            ; USED BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
2185            ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
2186            ;--
```

```
 2188                                     .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0101 -
 2189                                     ;****************************************************************************
 2190                                     ;*      This is an error reporting subroutine which prints additional error
 2191                                     ;*      information if an error is detected in TEST 1 (Register Address
 2192                                     ;*      Access Test).  This subroutine reports the type of access (Read or
 2193                                     ;*      Write or both) which caused a bus time-out trap (004 trap).
 2194                                     ;*      A message indicating that the DHV may be at the wrong Q-bus address
 2195                                     ;*      is also printed.
 2196                                     ;*
 2197                                     ;* INPUTS:       R5 - Error flag word.
 2198                                     ;*                    If bit 0 is set, a read error occured.
 2199                                     ;*                    If bit 1 is set, a write error occured.
 2200                                     ;*
 2201                                     ;* OUTPUTS:      Messages are printed at the operator console.
 2202                                     ;*
 2203                                     ;* CALLING SEQUENCE:     Include the label "ER0101" as the message pointer
 2204                                     ;*                       parameter in the DRS error report macro call.
 2205                                     ;*
 2206                                     ;* COMMENTS:
 2207                                     ;*
 2208                                     ;* SUBORDINATE ROUTINES USED: None.
 2209                                     ;****************************************************************************
 2210
 2211 013562                              BGNMSG  ER0101
      013562                                                                                  ER0101::
 2212 013562                              SAVE                        ;SAVE THE GPR CONTENTS.
      013562   004537  005326                     JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
 2213
 2214 013566   032705  000001             BIT     @BIT0,R5            ;TEST FOR READ ERROR.
 2215 013572   001410                     BEQ     2$                  ;SKIP READ ERROR MSG IF NO READ ERROR.
 2216 013574                              PRINTB  @MSG1               ;PRINT READ ERROR MESSAGE.
      013574   012746  013666                                                             MOV     @MSG1,-(SP)
      013600   012746  000001                                                             MOV     @1,-(SP)
      013604   010600                                                                     MOV     SP,R0
      013606   104414                                                                     TRAP    C$PNTB
      013610   062706  000004                                                             ADD     @4,SP
 2217 013614   032705  000002    2$:      BIT     @BIT1,R5            ;TEST FOR WRITE ERROR.
 2218 013620   001410                     BEQ     4$                  ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
 2219 013622                              PRINTB  @MSG2               ;PRINT WRITE ERROR MESSAGE.
      013622   012746  013744                                                             MOV     @MSG2,-(SP)
      013626   012746  000001                                                             MOV     @1,-(SP)
      013632   010600                                                                     MOV     SP,R0
      013634   104414                                                                     TRAP    C$PNTB
      013636   062706  000004                                                             ADD     @4,SP
 2220 013642                    4$:      PRINTX  @MSG3               ;SUGGEST THAT DHV MAY BE AT WRONG ADDRESS.
      013642   012746  014023                                                             MOV     @MSG3,-(SP)
      013646   012746  000001                                                             MOV     @1,-(SP)
      013652   010600                                                                     MOV     SP,R0
      013654   104415                                                                     TRAP    C$PNTX
      013656   062706  000004                                                             ADD     @4,SP
 2221 013662                              PASS                        ;RESTORE THE GPR CONTENTS.
      013662   004736                             JSR     PC,@(SP)+              ;RETURN TO PREG05 SUBRT.
 2222 013664                              ENDMSG
      013664                                                                      L10002:
      013664   104423                                                                     TRAP    C$MSG
 2223
 2224                                     .NLIST  BEX                 ;$$$
```

```
2225 013666        045     101     102  MSG1::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY READ ATTEMPT.%N/
2226 013744        045     101     102  MSG2::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.%N/
2227 014023        045     101     104  MSG3::  .ASCIZ  /%ADHV MAY BE AT THE WRONG Q-BUS ADDRESS.%N%N/
2228
2229                                            .EVEN
```

```
2231                             .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0503 -
2232                             ;***********************************************************************************
2233                             ;*        This is an error reporting subroutine which prints an additional error
2234                             ;*        message whose address is passed as an input parameter.
2235                             ;*
2236                             ;* INPUTS:      R1 - Address of the message to print.
2237                             ;*
2238                             ;* OUTPUTS:     A messages is printed at the operator console.
2239                             ;*
2240                             ;* CALLING SEQUENCE:     Load the address of the message in R1.
2241                             ;*                      Include the label "ER0503" as the message pointer
2242                             ;*                      parameter in the Diag Super error report macro call.
2243                             ;*
2244                             ;* COMMENTS:    The message is printed as Basic error information.
2245                             ;*
2246                             ;* SUBORDINATE ROUTINES USED: None.
2247                             ;***********************************************************************************
2248
2249 014100                      BGNMSG  ER0503
     014100                                                                          ER0503::
2250
2251 014100                      PRINTB  #EF0503,R1      ;PRINT THE MESSAGE.
     014100  010146                                                          MOV     R1,-(SP)
     014102  012746  005461                                                  MOV     #EF0503,-(SP)
     014106  012746  000002                                                  MOV     #2,-(SP)
     014112  010600                                                          MOV     SP,R0
     014114  104414                                                          TRAP    C$PNTB
     014116  062706  000006                                                  ADD     #6,SP
2252
2253 014122                      ENDMSG
     014122                                                                  L10003:
     014122  104423                                                          TRAP    C$MSG
```

```
2255                         .SBTTL  GLOBAL ERROR REPORTING ROUTINE           - ER1603 -
2256                         ;**************************************************************************
2257                         ;*      This error reporting routine is used to print out a basic error
2258                         ;*      message, along with a message informing the operator which test is
2259                         ;*      about to be aborted.
2260                         ;*
2261                         ;* INPUTS:        R1 - Contains the address of the message to be printed.
2262                         ;*                ERRMSG - Contains the address of the message that indicates
2263                         ;*                the test that is being performed, eg DMA, BREAK etc.
2264                         ;*
2265                         ;* OUTPUTS:       Messages are printed at the operators console.
2266                         ;*                "testname TEST ABORTED"
2267                         ;*
2268                         ;* CALLING SEQUENCE:   Include the lable "ER1603" as the message pointer
2269                         ;*                     parameter in the DRS error report macro call.
2270                         ;*
2271                         ;* COMMENTS:
2272                         ;*
2273                         ;*
2274                         ;* SUBORDINATE ROUTINES CALLED: None.
2275                         ;**************************************************************************
2276 014124                         BGNMSG  ER1603
     014124                                                                    ER1603::
2277 014124                         SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
     014124  004537  005326                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2278
2279 014130                         PRINTB #EF0503,R1       ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
     014130  010146                                                 MOV     R1,-(SP)
     014132  012746  005461                                         MOV     #EF0503,-(SP)
     014136  012746  000002                                         MOV     #2,-(SP)
     014142  010600                                                 MOV     SP,R0
     014144  104414                                                 TRAP    C$PNTB
     014146  062706  000006                                         ADD     #6,SP
2280
2281 014152  013702  005322         MOV     ERRMSG,R2       ;GET THE "TEST MESSAGE".
2282 014156                         PRINTB #EF1601,R2       ;PRINT "TEST ABORTED" MESSAGE.
     014156  010246                                                 MOV     R2,-(SP)
     014160  012746  005466                                         MOV     #EF1601,-(SP)
     014164  012746  000002                                         MOV     #2,-(SP)
     014170  010600                                                 MOV     SP,R0
     014172  104414                                                 TRAP    C$PNTB
     014174  062706  000006                                         ADD     #6,SP
2283
2284 014200                         PASS                    ;RESTORE THE CONTENTS OF THE GPRS.
     014200  004736                 JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2285 014202                         ENDMSG
     014202                                                         L10004:
     014202  104423                                                 TRAP    C$MSG
```

```
2287                                      .SBTTL  GLOBAL ERROR REPORTING ROUTINE        - ER6201 -
2288                                      ;******************************************************************************
2289                                      ;*       This is an error reporting subroutine which is intended for use in the
2290                                      ;*       framing error and parity error tests.  It reports error information
2291                                      ;*       when a character has been read from the DUT with the incorrect
2292                                      ;*       combination of Framing and Parity error bits.
2293                                      ;*
2294                                      ;* INPUTS:        R2 - Data byte read from the DUT, including error flags.
2295                                      ;*                R3 - Line number multiplied by 2.
2296                                      ;*                R5 - Message flags, which messages to report.
2297                                      ;*                   Bit1 and bit3 - indicate which messages are to be
2298                                      ;*                   reported, Framing or Parity respectively.
2299                                      ;*                   Bit0 and bit 2 - "Set"/"Clear" message for
2300                                      ;*                   framing and parity errors bits.
2301                                      ;*
2302                                      ;* OUTPUTS:       Messages are printed at the operator console.
2303                                      ;*
2304                                      ;* CALLING SEQUENCE:    Include the label "ER6201" as the message pointer
2305                                      ;*                     parameter in the Diag Super error report macro call.
2306                                      ;*
2307                                      ;* COMMENTS:      The message is printed as Basic and Extended error information.
2308                                      ;*               The contents of the Indirect address register field of the DUT
2309                                      ;*               CSR may be altered.
2310                                      ;*
2311                                      ;* SUBORDINATE ROUTINES USED: PRTLPR.
2312                                      ;******************************************************************************
2313
2314 014204                              BGNMSG  ER6201
     014204                                                                              ER6201::
2315 014204                              SAVE                        ;SAVE THE CONTENTS OF THE GPR'S.
     014204   004537  005326                      JSR     R5,PREGO5     ;CALL REGISTER SAVE SUBRT.
2316
2317 014210   016304  005236            MOV     TXRXLB(R3),R4    ;GET THE ASSOCIATED TX LINE NUMBER.
2318 014214   006203                    ASR     R3               ;CALCULATE THE RX LINE NUMBER.
2319 014216   006204                    ASR     R4               ;CALCULATE THE ASSOCIATED LINE NUMBER.
2320 014220                             PRINTB  #EF6201,R3,R4    ;REPORT THE ERROR TYPE AND LINE NUMBERS.
     014220   010446                                                              MOV     R4,-(SP)
     014222   010346                                                              MOV     R3,-(SP)
     014224   012746  005671                                                      MOV     #EF6201,-(SP)
     014230   012746  000003                                                      MOV     #3,-(SP)
     014234   010600                                                              MOV     SP,R0
     014236   104414                                                              TRAP    C$PNTB
     014240   062706  000010                                                      ADD     #10,SP
2321                                     ;+
2322                                     ; Report Framing Error Problem.
2323                                     ;-
2324 014244   012704  011166            MOV     #EM6202,R4       ;SELECT THE "ERROR BIT CLEAR" MESSAGE.
2325 014250   012701  010130            MOV     #EM0509,R1       ;SELECT EXPECTED "ERROR BIT SET" MESSAGE.
2326 014254   032705  000002            BIT     #BIT1,R5         ;TEST IF FRAMING ERROR MESSAGE TO BE REPORTED.
2327 014260   001427                    BEQ     6$               ;BRANCH TO REPORT PARITY ERROR.
2328 014262   032705  000001            BIT     #BIT0,R5         ;TEST "ERROR BIT SET/CLEAR" MESSAGE FLAG.
2329 014266   001403                    BEQ     2$               ;BRANCH TO REPORT ERROR BIT "CLEAR".
2330 014270   010401                    MOV     R4,R1            ;SELECT EXPECTED "CLEAR" STATE MESSAGE.
2331 014272   012704  010130            MOV     #EM0509,R4       ;SELECT THE "ERROR BIT SET" MESSAGE.
2332 014276                       2$:    PRINTX  #EF6202,R4,R1    ;REPORT THE SOURCE OF THE PROBLEM.
     014276   010146                                                              MOV     R1,-(SP)
     014300   010446                                                              MOV     R4,-(SP)
```

```
            014302  012746  006004                                                                      MOV      #EF6202,-(SP)
            014306  012746  000003                                                                      MOV      #3,-(SP)
            014312  010600                                                                              MOV      SP,RO
            014314  104415                                                                              TRAP     C$PNTX
            014316  062706  000010                                                                      ADD      #10,SP
      2333  014322  032705  000010              BIT    #BIT3,R5        ;TEST IF PARITY ERROR MESSAGE TO BE REPORTED.
      2334  014326  001424                       BEQ    10$             ;EXIT IF PARITY ERROR REPORT TO BE SKIPPED.
      2335  014330  012704  011166              MOV    #EM6202,R4       ;SELECT THE "CLEAR" MESSAGE.
      2336  014334  012701  010130              MOV    #EM0509,R1       ;SELECT THE EXPECTED "SET" STATE MESSAGE.
      2337                                ;+
      2338                                ; Report Parity Error Problem.
      2339                                ;-
      2340
      2341  014340  032705  000004      6$:     BIT    #BIT2,R5        ;TEST "SET"/"CLEAR" MESSAGE FLAG.
      2342  014344  001403                       BEQ    8$              ;BRANCH TO REPORT ERROR BIT CLEAR.
      2343  014346  010401                       MOV    R4,R1           ;SELECT THE EXPECTED "CLEAR" STATE MESSAGE.
      2344  014350  012704  010130              MOV    #EM0509,R4       ;SELECT THE "ERROR BIT SET" MESSAGE.
      2345  014354                      8$:     PRINTX #EF6203,R4,R1    ;REPORT THE SOURCE OF THE PROBLEM.
            014354  010146                                                                              MOV      R1,-(SP)
            014356  010446                                                                              MOV      R4,-(SP)
            014360  012746  006102                                                                      MOV      #EF6203,-(SP)
            014364  012746  000003                                                                      MOV      #3,-(SP)
            014370  010600                                                                              MOV      SP,RO
            014372  104415                                                                              TRAP     C$PNTX
            014374  062706  000010                                                                      ADD      #10,SP
      2346
      2347  014400                      10$:    PRINTX #EF1603,R2       ;REPORT ACTUAL DATA RECEIVED.
            014400  010246                                                                              MOV      R2,-(SP)
            014402  012746  005512                                                                      MOV      #EF1603,-(SP)
            014406  012746  000002                                                                      MOV      #2,-(SP)
            014412  010600                                                                              MOV      SP,RO
            014414  104415                                                                              TRAP     C$PNTX
            014416  062706  000006                                                                      ADD      #6,SP
      2348
      2349  014422  004737  023070      60$:    JSR    PC,PRTLPR        ;REPORT THE CONTENTS OF THE LPR FOR THIS LINE.
      2350  014426                              PASS                    ;RESTORE THE CONTENTS OF THE GPR'S.
            014426  004736                                     JSR     PC,@(SP)+              ;RETURN TO PREG05 SUBRT.
      2351  014430                              ENDMSG
            014430                                                              L10005:
            014430  104423                                                      TRAP     C$MSG
```

```
2353                                              .SBTTL  GLOBAL ERROR REPORTING ROUTINE           - ER9001 -
2354                                              ;****************************************************************************
2355                                              ;*       This is an error reporting subroutine which reports an unexpected
2356                                              ;*       code which has been found in the DUT CSR.  This code can be a BMP
2357                                              ;*       code, a Self-test code, or a Modem Status code.
2358                                              ;*
2359                                              ;* INPUTS:        R1 - Address of message to print first.
2360                                              ;*                R2 - Single byte code which has been read from the DUT.
2361                                              ;*                R4 - Line number associated with the code.
2362                                              ;*
2363                                              ;* OUTPUTS:       A messages is printed at the operator console.
2364                                              ;*
2365                                              ;* CALLING SEQUENCE:    Include the label "ER9001" as the message pointer
2366                                              ;*                      parameter in the Diag Super error report macro call.
2367                                              ;*
2368                                              ;* COMMENTS:      The message is printed as Basic and Extended error information.
2369                                              ;*
2370                                              ;* SUBORDINATE ROUTINES USED: None.
2371                                              ;****************************************************************************
2372
2373 014432                                       BGNMSG  ER9001
     014432                                                                              ER9001::
2374
2375 014432                                       PRINTB  #EF9001,R1      ;REPORT TYPE OF CODE FOUND.
     014432  010146                                                                      MOV     R1,-(SP)
     014434  012746  006266                                                              MOV     #EF9001,-(SP)
     014440  012746  000002                                                              MOV     #2,-(SP)
     014444  010600                                                                      MOV     SP,R0
     014446  104414                                                                      TRAP    C$PNTB
     014450  062706  000006                                                              ADD     #6,SP
2376 014454                                       PRINTX  #EF9002,R4      ;REPORT THE LINE NUMBER OF THE CODE.
     014454  010446                                                                      MOV     R4,-(SP)
     014456  012746  006350                                                              MOV     #EF9002,-(SP)
     014462  012746  000002                                                              MOV     #2,-(SP)
     014466  010600                                                                      MOV     SP,R0
     014470  104415                                                                      TRAP    C$PNTX
     014472  062706  000006                                                              ADD     #6,SP
2377 014476                                       PRINTX  #EF9003,R2      ;REPORT THE CODE WHICH WAS FOUND.
     014476  010246                                                                      MOV     R2,-(SP)
     014500  012746  006422                                                              MOV     #EF9003,-(SP)
     014504  012746  000002                                                              MOV     #2,-(SP)
     014510  010600                                                                      MOV     SP,R0
     014512  104415                                                                      TRAP    C$PNTX
     014514  062706  000006                                                              ADD     #6,SP
2378
2379 014520                                       ENDMSG
     014520                                                                      L10006:
     014520  104423                                                                      TRAP    C$MSG
```

```
2381                                .SBTTL  GLOBAL ERROR REPORTING ROUTINE        - ER9002 -
2382                                ;***********************************************************************
2383                                ;*      This is an error reporting subroutine which is intended for use in the
2384                                ;*      tranmsission and reception tests.  It reports the type of error which
2385                                ;*      has occurred when incorrect data is received from the DUT.  This
2386                                ;*      routine also reports the read and expected data values.
2387                                ;*
2388                                ;* INPUTS:       R1 - Address of message to print first.
2389                                ;*               R2 - Data byte read from the DUT.
2390                                ;*               R3 - Line number multiplied by 2.
2391                                ;*               R4 - Expected data byte, bit 15 set if "NONE".
2392                                ;*
2393                                ;* OUTPUTS:      A messages is printed at the operator console.
2394                                ;*
2395                                ;* CALLING SEQUENCE:     Include the label "ER9002" as the message pointer
2396                                ;*                       parameter in the Diag Super error report macro call.
2397                                ;*
2398                                ;* COMMENTS:     The message is printed as Basic and Extended error information.
2399                                ;*
2400                                ;* SUBORDINATE ROUTINES USED: PRTLPR.
2401                                ;***********************************************************************
2402
2403 014522                                BGNMSG  ER9002
     014522                                                                        ER9002::
2404
2405 014522  006203                        ASR     R3                      ;CALCULATE THE LINE NUMBER.
2406 014524  042702  177400                BIC     #177400,R2              ;MASK OUT ALL BUT DATA IN READ CHAR.
2407 014530                                PRINTB  #EF9006,R1,R3           ;PRINT THE FIRST LINE OF THE MESSAGE.
     014530  010346                                                                MOV     R3,-(SP)
     014532  010146                                                                MOV     R1,-(SP)
     014534  012746  006532                                                        MOV     #EF9006,-(SP)
     014540  012746  000003                                                        MOV     #3,-(SP)
     014544  010600                                                                MOV     SP,R0
     014546  104414                                                                TRAP    C$PNTB
     014550  062706  000010                                                        ADD     #10,SP
2408 014554                                PRINTX  #EF9004,#EM9010,R2          ;PRINT ACTUAL DATA.
     014554  010246                                                                MOV     R2,-(SP)
     014556  012746  011604                                                        MOV     #EM9010,-(SP)
     014562  012746  006451                                                        MOV     #EF9004,-(SP)
     014566  012746  000003                                                        MOV     #3,-(SP)
     014572  010600                                                                MOV     SP,R0
     014574  104415                                                                TRAP    C$PNTX
     014576  062706  000010                                                        ADD     #10,SP
2409 014602  005704                        TST     R4                      ;CHECK FOR "NONE" CODE SET IN EXPECTED DATA.
2410 014604  100414                        BMI     2$                      ;BRANCH TO PRINT "NONE" MESSAGE IF FLAG SET.
2411 014606                                PRINTX  #EF9004,#EM9009,R4         ;PRINT EXPECTED DATA.
     014606  010446                                                                MOV     R4,-(SP)
     014610  012746  011560                                                        MOV     #EM9009,-(SP)
     014614  012746  006451                                                        MOV     #EF9004,-(SP)
     014620  012746  000003                                                        MOV     #3,-(SP)
     014624  010600                                                                MOV     SP,R0
     014626  104415                                                                TRAP    C$PNTX
     014630  062706  000010                                                        ADD     #10,SP
2412 014634  000412                        BR      60$                     ;EXIT THIS ROUTINE.
2413 014636                        2$:     PRINTX  #EF9005,#EM9009 ;PRINT MESSAGE INDICATING NO EXPECTED DATA.
     014636  012746  011560                                                        MOV     #EM9009,-(SP)
     014642  012746  006501                                                        MOV     #EF9005,-(SP)
```

```
        014646  012746  000002                                              MOV     #2,-(SP)
        014652  010600                                                      MOV     SP,RO
        014654  104415                                                      TRAP    C$PNTX
        014656  062706  000006                                              ADD     #6,SP
2414    014662  004737  023070          60$:    JSR     PC,PRTLPR   ;PRINT CONTENTS OF THE LPR REGISTER.
2415    014666                                  ENDMSG
        014666                                                      L10007:
        014666  104423                                                      TRAP    C$MSG
```

```
2417                                    .SBTTL  GLOBAL ERROR REPORTING ROUTINE            - ER9003 -
2418                                    ;****************************************************************************
2419                                    ;*        This is an error reporting subroutine which is intended for use in the
2420                                    ;*        tranmsission and reception tests.  It reports error information when
2421                                    ;*        a character has been read from the DUT with an error flag or flags
2422                                    ;*        set (ie. over-run, framing, or parity flag).
2423                                    ;*
2424                                    ;* INPUTS:        R2 - Data byte read from the DUT, including error flags.
2425                                    ;*                R3 - Line number multiplied by 2.
2426                                    ;*
2427                                    ;* OUTPUTS:       A messages is printed at the operator console.
2428                                    ;*
2429                                    ;* CALLING SEQUENCE:     Include the label "ER9003" as the message pointer
2430                                    ;*                       parameter in the Diag Super error report macro call.
2431                                    ;*
2432                                    ;* COMMENTS:      The message is printed as Basic and Extended error information.
2433                                    ;*               The contents of the Indirect address register field of the DUT
2434                                    ;*               CSR may be altered.
2435                                    ;*
2436                                    ;* SUBORDINATE ROUTINES USED: None.
2437                                    ;****************************************************************************
2438
2439 014670                                    BGNMSG  ER9003
     014670                                                                                    ER9003::
2440
2441 014670  006203                            ASR     R3                      ;CALCULATE THE LINE NUMBER.
2442 014672                                    PRINTB  #EF9007,R3              ;REPORT THE ERROR TYPE AND LINE NUMBER.
     014672  010346                                                                    MOV     R3,-(SP)
     014674  012746  006551                                                            MOV     #EF9007,-(SP)
     014700  012746  000002                                                            MOV     #2,-(SP)
     014704  010600                                                                    MOV     SP,R0
     014706  104414                                                                    TRAP    C$PNTB
     014710  062706  000006                                                            ADD     #6,SP
2443 014714  010201                            MOV     R2,R1                   ;EXTRACT THE RECEIVED CHARACTER FROM THE
2444 014716  042701  177400                    BIC     #177400,R1              ;  PASSED IN CHAR VALUE WITH FLAGS.
2445 014722                                    PRINTX  #EF9008,R1              ;REPORT THE VALUE OF THE RECEIVED CHAR.
     014722  010146                                                                    MOV     R1,-(SP)
     014724  012746  006645                                                            MOV     #EF9008,-(SP)
     014730  012746  000002                                                            MOV     #2,-(SP)
     014734  010600                                                                    MOV     SP,R0
     014736  104415                                                                    TRAP    C$PNTX
     014740  062706  000006                                                            ADD     #6,SP
2446                                    ;+
2447                                    ; Report OVERRUN flag set if necessary.
2448                                    ;-
2449 014744  012701  011630                    MOV     #EM9011,R1              ;SELECT THE OVERRUN ERROR MESSAGE.
2450 014750  032702  040000                    BIT     #BIT14,R2               ;CHECK OVERRUN ERROR FLAG IN PASSED IN CHAR.
2451 014754  001402                            BEQ     2$                      ;SKIP ERROR IF OVERRUN ERROR FLAG WAS CLEAR.
2452 014756  004737  015020                    JSR     PC,50$                  ;REPORT THE OVERRUN ERROR FLAG WAS SET.
2453                                    ;+
2454                                    ; Report FRAMING flag set if necessary.
2455                                    ;-
2456 014762  012701  011640            2$:     MOV     #EM9012,R1              ;SELECT THE FRAMING ERROR MESSAGE.
2457 014766  032702  020000                    BIT     #BIT13,R2               ;CHECK FRAMING ERROR FLAG IN PASSED IN CHAR.
2458 014772  001402                            BEQ     4$                      ;SKIP ERROR IF FRAMING ERROR FLAG WAS CLEAR.
2459 014774  004737  015020                    JSR     PC,50$                  ;REPORT THE FRAMING ERROR MESSAGE.
2460                                    ;+
```

```
2461                                    ; Report PARITY flag set if necessary.
2462                                    ;-
2463 015000  012701  011650    4$:      MOV    #EM9013,R1        ;SELECT THE PARITY ERROR MESSAGE.
2464 015004  032702  010000             BIT    #BIT12,R2         ;CHECK PARITY ERROR FLAG IN PASSED IN CHAR.
2465 015010  001415                      BEQ    60$               ;EXIT ROUTINE IF PARITY ERRO FLAG WAS CLEAR.
2466 015012  004737  015020             JSR    PC,50$            ;REPORT THE PARITY ERROR MESSAGE.
2467 015016  000412                      BR     60$               ;EXIT THIS ROUTINE.
2468
2469                                    ;+
2470                                    ; Local subroutine to report an error flag status.
2471                                    ;-
2472 015020                     50$:     PRINTX  #EF9009,R1
     015020  010146                                                                        MOV    R1,-(SP)
     015022  012746  006704                                                                MOV    #EF9009,-(SP)
     015026  012746  000002                                                                MOV    #2,-(SP)
     015032  010600                                                                        MOV    SP,RO
     015034  104415                                                                        TRAP   C$PNTX
     015036  062706  000006                                                                ADD    #6,SP
2473 015042  000207                      RTS    PC
2474
2475 015044  004737  023070    60$:     JSR    PC,PRTLPR         ;REPORT THE LPR CONTENTS FOR THIS LINE.
2476 015050                              ENDMSG
     015050                                                                   L10010:
     015050  104423                                                                        TRAP   C$MSG
```

```
2478
2479                                    .SBTTL  GLOBAL ERROR REPORTING ROUTINE        - ER9004 -
2480                                    ;*********************************************************************
2481                                    ;*      This is an error reporting subroutine which reports error summaries
2482                                    ;*      for lines which have exceeded the specified maximum number of
2483                                    ;*      individual reception errors.
2484                                    ;*
2485                                    ;* INPUTS:        R1 - Address of message to print first.
2486                                    ;*                ERCNTB - Label at base of line error counters table.
2487                                    ;*                ERSMRF - "Report error summary for line" flags.
2488                                    ;*
2489                                    ;* OUTPUTS:       A message is printed at the operator console.
2490                                    ;*
2491                                    ;* CALLING SEQUENCE:    Include the label "ER9004" as the message pointer
2492                                    ;*                      parameter in the Diag Super error report macro call.
2493                                    ;*
2494                                    ;* COMMENTS:      The message is printed as Basic and Extended error information.
2495                                    ;*                The contents of GPR's R2, R3, R4, and R5 are destroyed.
2496                                    ;*
2497                                    ;* SUBORDINATE ROUTINES USED: None.
2498                                    ;*********************************************************************
2499 015052                            BGNMSG  ER9004
     015052
2500                                                                                   ER9004::
2501 015052                            PRINTB  #EF0503,#EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
     015052  012746  011657                                                    MOV    #EM9014,-(SP)
     015056  012746  005461                                                    MOV    #EF0503,-(SP)
     015062  012746  000002                                                    MOV    #2,-(SP)
     015066  010600                                                            MOV    SP,R0
     015070  104414                                                            TRAP   C$PNTB
     015072  062706  000006                                                    ADD    #6,SP
2502 015076  005002              CLR    R2              ;CLEAR THE LINE COUNTER.
2503 015100  013703  002502      MOV    ERSMRF,R3       ;GET THE ERROR SUMMARY FLAGS.
2504 015104  005004              CLR    R4              ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
2505 015106  000241        2$:   CLC                    ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
2506 015110  006003              ROR    R3              ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
2507 015112  103013              BCC    4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
2508 015114                      PRINTX  #EF9010,R2,ERCNTB(R4)
     015114  016446  003304                                                    MOV    ERCNTB(R4),-(SP)
     015120  010246                                                            MOV    R2,-(SP)
     015122  012746  006743                                                    MOV    #EF9010,-(SP)
     015126  012746  000003                                                    MOV    #3,-(SP)
     015132  010600                                                            MOV    SP,R0
     015134  104415                                                            TRAP   C$PNTX
     015136  062706  000010                                                    ADD    #10,SP
2509 015142  012405        4$:   MOV    (R4)+,R5        ;INCREMENT THE LINE OFFSET BY 2.
2510 015144  005202              INC    R2              ;INCREMT THE LINE COUNTER.
2511 015146  005703              TST    R3              ;CHECK THE ERROR SUMMARY FLAGS.
2512 015150  001356              BNE    2$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.
2513
2514 015152                      ENDMSG
     015152                                                                    L10011:
     015152  104423                                                            TRAP   C$MSG
```

```
2516                              .SBTTL  GLOBAL ERROR REPORTING ROUTINE        - ER9005 -
2517                              ;*****************************************************************************
2518                              ;*        This is an error reporting subroutine which reports incomplete data
2519                              ;*        transmissions or receptions.
2520                              ;*
2521                              ;* INPUTS:        R1 - Either "TRANSMITTED" or "RECEIVED" to indicate TX or RX.
2522                              ;*                R2 - Bit map of lines which did not complete TX or RX.
2523                              ;*                R4 - Address of base of the correct character counters table.
2524                              ;*                DPLENB - Label at base of data pattern length table.
2525                              ;*                EM9015 - Symbolic address of the "TRANSMITTED" message.
2526                              ;*
2527                              ;* OUTPUTS:       A message is printed at the operator console.
2528                              ;*
2529                              ;* CALLING SEQUENCE:    Include the label "ER9005" as the message pointer
2530                              ;*                      parameter in the Diag Super error report macro call.
2531                              ;*
2532                              ;* COMMENTS:      The message is printed as Basic and Extended error information.
2533                              ;*               The contents of the indirect address field in the DUT CSR may
2534                              ;*               be altered.
2535                              ;*
2536                              ;* SUBORDINATE ROUTINES USED: PRTLPR.
2537                              ;*****************************************************************************
2538
2539 015154                      BGNMSG  ER9005
     015154                                                                    ER9005::
2540 015154                      SAVE                            ;SAVE THE CONTENTS OF THE GPR'S.
     015154  004537  005326               JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2541
2542 015160                      PRINTB  #EF9013,R1      ;REPORT THE SECONDARY ERROR MESSAGE.
     015160  010146                                       MOV     R1,-(SP)
     015162  012746  007146                               MOV     #EF9013,-(SP)
     015166  012746  000002                               MOV     #2,-(SP)
     015172  010600                                       MOV     SP,R0
     015174  104414                                       TRAP    C$PNTB
     015176  062706  000006                               ADD     #6,SP
2543 015202  005003              CLR     R3              ;CLEAR THE LINE COUNTER.
2544 015204  022701  011753      CMP     #EM9015,R1      ;CHECK IF ADDRESS CORRESPONDS TO TX MESSAGE.
2545 015210  001032              BNE     6$              ;BRANCH IF RECEPTION MESSAGE TO BE PRINTED.
2546
2547                             ;*
2548                             ; Perform TX incomplete error message reporting.
2549                             ;-
2550 015212                      PRINTX  #EM9030         ;PRINT "NO TX COMPLETION INTERRUPTS RECEIVED"
     015212  012746  012366                               MOV     #EM9030,-(SP)
     015216  012746  000001                               MOV     #1,-(SP)
     015222  010600                                       MOV     SP,R0
     015224  104415                                       TRAP    C$PNTX
     015226  062706  000004                               ADD     #4,SP
2551 015232  000241      2$:     CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
2552 015234  006002              ROR     R2              ;SHIFT "TX NOT DONE" FLAG INTO CARRY.
2553 015236  103013              BCC     4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
2554 015240                      PRINTX  #EF9020,R3      ;PRINT "TOO FEW TX.ACTIONS GENERATED" MSG.
     015240  010346                                       MOV     R3,-(SP)
     015242  012746  007232                               MOV     #EF9020,-(SP)
     015246  012746  000002                               MOV     #2,-(SP)
     015252  010600                                       MOV     SP,R0
     015254  104415                                       TRAP    C$PNTX
```

```
          015256   062706   000006                                                                              ADD     #6,SP
     2555 015262   004737   023070                    JSR     PC,PRTLPR       ;REPORT CONTENTS OF LPR REGISTER FOR THIS LINE.
     2556 015266   005203                      4$:    INC     R3              ;INCREMENT LINE COUNTER.
     2557 015270   005702                             TST     R2              ;CHECK THE "TX NOT DONE FLAGS".
     2558 015272   001357                             BNE     2$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.
     2559 015274   000440                             BR      10$             ;EXIT THIS ROUTINE.
     2560                                      ;*
     2561                                      ; Perform RX incomplete error message reporting.
     2562                                      ;-
     2563 015276   000241                      6$:    CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
     2564 015300   006002                             ROR     R2              ;SHIFT "RX NOT DONE" FLAG INTO CARRY.
     2565 015302   103031                             BCC     8$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
     2566 015304   006303                             ASL     R3              ;SHIFT LINE # TO GIVE CORRECT TABLE OFFSET.
     2567 015306   016305   005236                     MOV     TXRXLB(R3),R5   ;GET THE "ASSOCIATED" RECEIVE LINE OFFSET.
     2568 015312   010246                             MOV     R2,-(SP)        ;SAVE THE "RX NOT DONE" FLAGS ON THE STACK.
     2569 015314   010502                             MOV     R5,R2           ;COPY THE ASSOCIATED TX LINE OFFSET.
     2570 015316   016505   003444                     MOV     CHCNTB(R5),R5   ;GET THE TOTAL NUMBER OF EXPECTED CHARS.
     2571 015322   006202                             ASR     R2              ;SHIFT THE TABLE OFFSET TO GIVE A LINE NUMBER.
     2572 015324   006203                             ASR     R3              ;SHIFT TABLE OFFSET TO GIVE LINE NUMBER.
     2573 015326                                      PRINTX  #EF9012,R3,R1,(R4),R5,R2  ;REPORT NUMBER OF CHARS ON LINE.
          015326   010246                                                                              MOV     R2,-(SP)
          015330   010546                                                                              MOV     R5,-(SP)
          015332   011446                                                                              MOV     (R4),-(SP)
          015334   010146                                                                              MOV     R1,-(SP)
          015336   010346                                                                              MOV     R3,-(SP)
          015340   012746   007032                                                                     MOV     #EF9012,-(SP)
          015344   012746   000006                                                                     MOV     #6,-(SP)
          015350   010600                                                                              MOV     SP,R0
          015352   104415                                                                              TRAP    C$PNTX
          015354   062706   000016                                                                     ADD     #16,SP
     2574 015360   012602                             MOV     (SP)+,R2        ;RESTORE THE "RX NOT DONE" FLAGS.
     2575 015362   004737   023070                     JSR     PC,PRTLPR       ;REPORT CONTENTS OF LPR REGISTER FOR THIS LINE.
     2576 015366   005724                      8$:    TST     (R4)+           ;INCREMENT THE CHARACTER COUNTER TABLE.
     2577 015370   005203                             INC     R3              ;INCREMENT THE LINE COUNTER.
     2578 015372   005702                             TST     R2              ;CHECK THE "RX NOT DONE FLAGS".
     2579 015374   001340                             BNE     6$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.
     2580 015376                              10$:    PASS                    ;RESTORE THE CONTENTS OF THE GPRS.
          015376   004736                                                     JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
     2581 015400                                      ENDMSG
          015400                                                              L10012:
          015400   104423                                                              TRAP    C$MSG
```

```
2583                              .SBTTL  GLOBAL ERROR REPORTING ROUTINE        - ER9101 -
2584                              ;*******************************************************************
2585                              ;*      This is a general error reporting subroutine which reports a message
2586                              ;*      which takes a single, 2 digit decimal argument after the end of an
2587                              ;*      ASCII message.
2588                              ;*
2589                              ;* INPUTS:        R1 - Value to be printed after msg as 2 decimal digits.
2590                              ;*                R2 - Address of message to print first.
2591                              ;*
2592                              ;* OUTPUTS:       A messages is printed at the operator console.
2593                              ;*
2594                              ;* CALLING SEQUENCE:   Include the label "ER9101" as the message pointer
2595                              ;*                     parameter in the Diag Super error report macro call.
2596                              ;*
2597                              ;* COMMENTS:   The message is printed as Basic error information.
2598                              ;*
2599                              ;* SUBORDINATE ROUTINES USED: None.
2600                              ;*******************************************************************
2601
2602 015402                         BGNMSG  ER9101
     015402                                                              ER9101::
2603
2604 015402                         PRINTB  #EF9006,R2,R1   ;REPORT THE STRING FOLLOWED BY THE NUMBER.
     015402  010146                                                MOV   R1,-(SP)
     015404  010246                                                MOV   R2,-(SP)
     015406  012746  006532                                        MOV   #EF9006,-(SP)
     015412  012746  000003                                        MOV   #3,-(SP)
     015416  010600                                                MOV   SP,R0
     015420  104414                                                TRAP  C$PNTB
     015422  062706  000010                                        ADD   #10,SP
2605
2606 015426                         ENDMSG
     015426                                                              L10013:
     015426  104423                                                TRAP  C$MSG
```

```
2608                                     .SBTTL  GLOBAL ERROR REPORTING ROUTINE           - ER9102 -
2609                                     ;**********************************************************************
2610                                     ;*      This is an error reporting subroutine which prints additional error
2611                                     ;*      information after the error message header.
2612                                     ;*      This routine is passed a bit map which specifies the lines for which
2613                                     ;*      the error condition should be reported.
2614                                     ;*
2615                                     ;* INPUTS:         R1 - Address of the message to be printed by this routine.
2616                                     ;*                 R2 - Bit map of lines for which to report errors.
2617                                     ;*
2618                                     ;* OUTPUTS:        Messages are printed at the operator console.
2619                                     ;*
2620                                     ;* CALLING SEQUENCE:    Load the address of the message in R1.
2621                                     ;*                      Load the bit map of lines with errors in R2.
2622                                     ;*                      Include the label "ER9102" as the message pointer
2623                                     ;*                         (ERRBLK) in the Diag Super error report macro call.
2624                                     ;*
2625                                     ;* COMMENTS:       The output format of this message is:
2626                                     ;*                 "TEXT MESSAGE POINTED TO BY R1"
2627                                     ;*                     "ERROR CONDITION ON LINE nn"
2628                                     ;*                     "ERROR CONDITION ON LINE ..."
2629                                     ;*                 The top message, and the message for each line are printed
2630                                     ;*                 as basic error information.
2631                                     ;*
2632                                     ;* SUBORDINATE ROUTINES USED: None.
2633                                     ;**********************************************************************
2634
2635 015430                             BGNMSG  ER9102
     015430                                                                            ER9102::
2636 015430                             SAVE                       ;SAVE THE CONTENTS OF THE GPRS.
     015430  004537  005326                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2637
2638 015434                             PRINTB  #EF0503,R1         ;PRINT THE FIRST LINE OF THE MESSAGE.
     015434  010146                                                 MOV     R1,-(SP)
     015436  012746  005461                                        MOV     #EF0503,-(SP)
     015442  012746  000002                                        MOV     #2,-(SP)
     015446  010600                                                 MOV     SP,R0
     015450  104414                                                 TRAP    C$PNTB
     015452  062706  000006                                        ADD     #6,SP
2639 015456  005003                     CLR     R3                ;CLEAR THE LINE NUMBER.
2640 015460  000241             2$:     CLC                       ;PREPARE TO ROTATE NEXT BIT OUT OF MAP.
2641 015462  006002                     ROR     R2                ;GET THE NEXT BIT OF THE BIT MAP.
2642 015464  103011                     BCC     4$                ;SKIP PRINTING MESSAGE IF THE BIT IS CLEAR.
2643 015466                             PRINTB  #EF9103,R3         ;REPORT THIS LINE HAD THE ERROR.
     015466  010346                                                 MOV     R3,-(SP)
     015470  012746  007316                                        MOV     #EF9103,-(SP)
     015474  012746  000002                                        MOV     #2,-(SP)
     015500  010600                                                 MOV     SP,R0
     015502  104414                                                 TRAP    C$PNTB
     015504  062706  000006                                        ADD     #6,SP
2644 015510  005203             4$:     INC     R3                ;INCREMENT THE LINE COUNTER.
2645 015512  005702                     TST     R2                ;CHECK THE BIT MAP.
2646 015514  001361                     BNE     2$                ;LOOP IF NOT ALL SET BITS REMOVED FROM BIT MAP.
2647 015516                             PRINTB  #EF9101            ;PRINT A BLANK LINE.
     015516  012746  007313                                        MOV     #EF9101,-(SP)
     015522  012746  000001                                        MOV     #1,-(SP)
     015526  010600                                                 MOV     SP,R0
```

```
        015530  104414                                                              TRAP    C$PNTB
        015532  062706  000004                                                      ADD     #4,SP
2648 015536                           60$:    PASS            ;RESTORE THE SAVED CONTENTS OF THE GPRS.
        015536  004736                                 JSR    PC,@(SP)+        ;RETURN TO PREGO5 SUBRT.
2649 015540                           ENDMSG
        015540                                                                L10014:
        015540  104423                                                              TRAP    C$MSG
```

```
2651                                    .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9301 -
2652                                    ;********************************************************************
2653                                    ;*      This is an error reporting subroutine which prints any BMP codes
2654                                    ;*      that are found in the BMP code queue, together with the the number of
2655                                    ;*      the test that was executing at the time the BMP code was logged.
2656                                    ;*
2657                                    ;* INPUTS:       R1 - The address of the first message to be reported.
2658                                    ;*               R2 - The address of the next empty cell in the queue.
2659                                    ;*
2660                                    ;* OUTPUTS:      The test number followed by the BMP code are printed at the
2661                                    ;*               operator console.
2662                                    ;*
2663                                    ;* CALLING SEQUENCE:    Include the label "ER9301" as the message pointer
2664                                    ;*                      parameter in the Diag Super error report macro call.
2665                                    ;*
2666                                    ;* COMMENTS:     The message is printed as Basic error information.
2667                                    ;*
2668                                    ;* SUBORDINATE ROUTINES USED: None.
2669                                    ;********************************************************************
2670
2671 015542                            BGNMSG  ER9301
     015542
2672 015542                                                                    ER9301::
     015542                            SAVE                    ;SAVE THE GPRS ON THE STACK.
     015542  004537  005326                    JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
2673
2674 015546                            PRINTB  #EF0503,R1      ;REPORT UNEXPECTED BMP CODES FOUND.
     015546  010146                                                    MOV     R1,-(SP)
     015550  012746  005461                                           MOV     #EF0503,-(SP)
     015554  012746  000002                                           MOV     #2,-(SP)
     015560  010600                                                   MOV     SP,R0
     015562  104414                                                   TRAP    C$PNTB
     015564  062706  000006                                           ADD     #6,SP
2675 015570  012703  002514                    MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
2676 015574  012705  012704                    MOV     #EM9302,R5      ;GET THE MESSAGE TO BE REPORTED.
2677 015600  012301              2$:           MOV     (R3)+,R1        ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
2678 015602  012304                            MOV     (R3)+,R4        ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
2679 015604  004737  015666                    JSR     PC,50$          ;GO REPORT THE BMP CODE.
2680 015610  020302                            CMP     R3,R2           ;CHECK IF ALL CODES HAVE BEEN REPORTED.
2681 015612  103772                            BLO     2$              ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
2682
2683                                    ;+
2684                                    ; Check if overflow has occurred.
2685                                    ; The conditions for overflow are: the pointer contains the address of the
2686                                    ; last cell in the queue, and a bmp code has already been written into that
2687                                    ; cell.
                                        ;-
2688 015614  020227  002710                    CMP     R2,#BMPCQE-4    ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
2689 015620  001036                            BNE     60$             ;EXIT IF NOT AT THE LAST LOCATION.
2690 015622  005762  000002                    TST     2(R2)           ;CHECK FOR A BMP CODE IN THE LAST CELL
2691 015626  001433                            BEQ     60$             ;EXIT IF NO OVERFLOW HAS OCCURED. CELL EMPTY.
2692 015630  012301                            MOV     (R3)+,R1        ;GET THE TEST NUMBER OFF THE QUEUE.
2693 015632  011304                            MOV     (R3),R4         ;GET THE BMP CODE OFF THE QUEUE.
2694 015634  012705  012734                    MOV     #EM9303,R5      ;SELECT THE MESSAGE TO BE REPORTED.
2695 015640                            PRINTX  #EF9302         ;REPORT OVERFLOW CONDITION.
     015640  012746  007432                                           MOV     #EF9302,-(SP)
     015644  012746  000001                                           MOV     #1,-(SP)
     015650  010600                                                   MOV     SP,R0
     015652  104415                                                   TRAP    C$PNTX
```

```
       015654  062706  000004                                                    ADD      #4,SP
  2696 015660  004737  015666          JSR     PC,50$              ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
  2697 015664  000414                  BR      60$                 ;EXIT.
  2698
  2699 015666                  50$:    PRINTX  #EF9301,R5,R1,R4  ;PRINT THE MESSAGE.
       015666  010446                                                             MOV      R4,-(SP)
       015670  010146                                                             MOV      R1,-(SP)
       015672  010546                                                             MOV      R5,-(SP)
       015674  012746  007364                                                     MOV      #EF9301,-(SP)
       015700  012746  000004                                                     MOV      #4,-(SP)
       015704  010600                                                             MOV      SP,R0
       015706  104415                                                             TRAP     C$PNTX
       015710  062706  000012                                                     ADD      #12,SP
  2700 015714  000207                  RTS     PC                  ;RETURN.
  2701 015716                  60$:    PASS                        ;RESTORE THE GPR CONTENTS.
       015716  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
  2702
  2703 015720                  ENDMSG
       015720                                                      L10015:
       015720  104423                                                     TRAP     C$MSG
```

```
2705                              .SBTTL  GLOBAL SUBROUTINES SECTION
2707                              ;######################################################################
2708                              ;
2709                              ;                    FVTSKL3.P11
2710                              ;
2711                              ;######################################################################
2713
2714
2715                              ;++
2716                              ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2717                              ; THAT ARE USED IN MORE THAN ONE TEST.
2718                              ;--
```

```
2720                                    .SBTTL  GLOBAL SUBROUTINE                    - ALTFLD -
2721                                    ;** ********************************************************************************
2722                                    ;*                - Alter Device Register Fields Routine -
2723                                    ;*      This subroutine alters the specified field of the specified device
2724                                    ;*      register for the specified lines.  This routine can be used to set
2725                                    ;*      or clear bits within selected fields of selected registers.
2726                                    ;*      Use examples:   Set RX.BAUD.RATE fields on lines 3 and 6.
2727                                    ;*                      Clear TX.DMA bits on all lines.
2728                                    ;*
2729                                    ;* INPUTS:       R1 - Address of the registers to alter.
2730                                    ;*               R2 - Bit fields set to desired states.
2731                                    ;*               R3 - Bit map of lines for which to alter register.
2732                                    ;*               R4 - Mask of bits to alter (1 indicates change bit).
2733                                    ;*               CSRA - Contains the address of the device CSR.
2734                                    ;*               IESTAT - Saved states of the interrupt enable bits.
2735                                    ;*
2736                                    ;* OUTPUTS:      DEVICE REGISTERS - Specified register fields altered.
2737                                    ;*               CSR IND.ADR.REG field - Destroyed.
2738                                    ;*
2739                                    ;* CALLING SEQUENCE:    JSR     PC,ALTFLD
2740                                    ;*
2741                                    ;* COMMENTS:     This routine reads the specified registers for all lines
2742                                    ;*               with numbers lower than the highes specified line.
2743                                    ;*               This routine does not read the CSR.
2744                                    ;*
2745                                    ;* SUBROUTINES CALLED: None.
2746                                    ;-- ********************************************************************************
2747
2748 015722                             ALTFLD:: SAVE                      ;SAVE CONTENTS OF GPRS R0 THRU R5.
     015722  004537  005326                      JSR     R5,PREG05        ;CALL REGISTER SAVE SUBRT.
2749
2750                                    ;+
2751                                    ; Set up to loop for each line:
2752                                    ;    Prepare the word to be ORed into the register contents.
2753                                    ;    Set up the word to write into the IND.ADR.REG field of the CSR.
2754                                    ;-
2755 015726  010400                              MOV     R4,R0            ;CALCULATE THE NEW CONTENTS OF THE
2756 015730  005100                              COM     R0               ;  REGISTER FIELDS WHICH ARE TO BE
2757 015732  040002                              BIC     R0,R2            ;  ALTERRED BY THIS ROUTINE.
2758 015734  013705  002234                      MOV     IESTAT,R5        ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.
2759                                    ;+
2760                                    ; Loop once for each line, altering the specified field in the specified
2761                                    ;    register if the line has been selected for altering.
2762                                    ; Exit the loop if no more lines to alter, or if we have altered the max
2763                                    ;    allowable number of lines (as specified by NUMLNS).
2764                                    ;-
2765 015740  000241                              CLC                      ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
2766 015742  006003                      2$:     ROR     R3               ;GET THE LINE SELECT BIT FOR THIS LINE.
2767 015744  103006                              BCC     4$               ;SKIP SETUP IF LINE IS NOT SELECTED.
2768 015746  010577  164230                      MOV     R5,@CSRA         ;SET DUT CSR IND.ADR.REG FIELD TO THIS LINE.
2769 015752  011100                              MOV     (R1),R0          ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
2770 015754  040400                              BIC     R4,R0            ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
2771 015756  050200                              BIS     R2,R0            ;OR IN THE NEW STATES OF THE FIELDS.
2772 015760  010011                              MOV     R0,(R1)          ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
2773 015762  005205                      4$:     INC     R5               ;SET LINE NUMBER TO THE NEXT LINE.
2774 015764  005703                              TST     R3               ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
2775 015766  001365                              BNE     2$               ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.
```

GLOBAL SUBROUTINE   - ALTFLD -

```
2776
2777 015770                         60$:    PASS                    ;RESTORE GPRS.
     015770  004736                                  JSR     PC,@(SP)+              ;RETURN TO PREG05 SUBRT.
2778 015772  000207                         RTS     PC              ;RETURN TO CALLING ROUTNE.
```

```
2780                                      .SBTTL  GLOBAL SUBROUTINE                    - CALMSL -
2781                                      ;** ******************************************************************************
2782                                      ;*                - Calibrate Milli Second Loop count subroutine -
2783                                      ;*       This subroutine calibrates the timing loop which is used in the MSLOOP
2784                                      ;*       routine.  This subroutine calculates a value for the MSLCNT variable
2785                                      ;*       which is the number of software loops which takes 1 ms to execute in
2786                                      ;*       the MSLOOP routine.  This routine calibrates the count by using the
2787                                      ;*       Line Time Clock (LTC), so if no LTC is available the default value for
2788                                      ;*       the delay count must be used.
2789                                      ;*
2790                                      ;*
2791                                      ;* INPUTS:        MSLCNT - Default 1 ms delay loop count value, or
2792                                      ;*                         value from previous calibration.
2793                                      ;*                MSTICK - Number of MS per LTC clock tick.
2794                                      ;*                TIMER1 - Timer counter changed by LTC interrupt service rtn.
2795                                      ;*                CLKHRZ - Number of LTC clicks per second (50 or 60).
2796                                      ;*
2797                                      ;* OUTPUTS:       CARRY - Set if LTC is available, and new calibration performed.
2798                                      ;*                MSLCNT - New 1 ms delay loop count value if LTC available, or
2799                                      ;*                         unchanged if no LTC is available.
2800                                      ;*
2801                                      ;* CALLING SEQUENCE:    JSR    PC,CALMSL
2802                                      ;*
2803                                      ;* COMMENTS:
2804                                      ;*
2805                                      ;* SUBORDINATE ROUTINES CALLED: UNSDIV,OOPS.
2806                                      ;-- ******************************************************************************
2807
2808 015774                              CALMSL:: SAVE                       ;SAVE CONTENTS OF GPRS R0 THRU R5.
     015774  004537  005326                        JSR    R5,PREG05         ;CALL REGISTER SAVE SUBRT.
2809 016000  005037  016214                        CLR    62$               ;CLEAR THE 2ND TIME FLAG.
2810                                      ;*
2811                                      ; Synchronize with the LTC.
2812                                      ;-
2813 016004  012705  000001              2$:       MOV    #1,R5             ;SET OUTER LOOP COUNTER TO 1 LOOP.
2814                                                                        ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE <**
2815                                                                        ;FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <**
2816 016010  005000                                CLR    R0                ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
2817 016012  012737  000001  002302               MOV    #1,TIMER1         ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
2818 016020  005737  002302              4$:       TST    TIMER1            ;CHECK FOR COUNTER HAVING GONE TO ZERO.
2819 016024  001410                                BEQ    6$                ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
2820 016026  005200                                INC    R0                ;COUNT THIS ITERATION OF THE INNER LOOP.
2821 016030  001373                                BNE    4$                ;LOOP IF COUNTER HAS NOT TURNED OVER.
2822 016032  005305                                DEC    R5                ;DECREMENT THE INNER LOOP COUNTER.
2823 016034  003371                                BGT    4$                ;LOOP IF OUTER LOOP COUNT NOT UP.
2824                                      ;*
2825                                      ; If we got no LTC interrupt, indicate that there is no LTC available.
2826                                      ; LTC must be flakey, or not really an LTC at all.
2827                                      ;-
2828 016036  005037  002300                        CLR    CLKHRZ            ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
2829 016042  000241                                CLC                      ;INDICATE FAILURE FOR RETURN.
2830 016044  000461                                BR     60$               ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
2831                                      ;*
2832                                      ; We are now synchronized with the LTC.
2833                                      ; Set up for the calibration loop.
2834                                      ;-
2835 016046  012704  002302              6$:       MOV    #TIMER1,R4        ;WILL TEST TIMER1 IN THE LOOP BELOW.
```

GLOBAL SUBROUTINE   - CALMSL -

```
2836 016052  005001                 CLR     R1              ;CLEAR THE OUTER LOOP COUNTER.
2837 016054  005002                 CLR     R2              ;INDICATE TO CHECK ALL BITS OF TIMER1.
2838 016056  005003                 CLR     R3              ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2839 016060  012714  000001         MOV     #1,(R4)         ;LOAD TIMER1 WITH COUNT OF 1.
2840
2841 016064  013705  002314  8$:    MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2842 016070  011400          10$:   MOV     (R4),R0         ;GET THE TIMER1 VALUE.
2843 016072  010037  016216         MOV     R0,64$          ;SAVE WORD (LIKE IN THE REAL LOOP).
2844 016076  040200                 BIC     R2,R0           ;LEAVE ALL THE BITS.
2845 016100  020003                 CMP     R0,R3           ;COMPARE AGAINST ZERO.
2846 016102  000261                 SEC                     ;SET CARRY IN CASE OF SUCCESS.
2847 016104  001406                 BEQ     12$             ;EXIT LOOP IF TIMER1 HAS CLEARED.
2848 016106  005305                 DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2849 016110  001367                 BNE     10$             ;LOOP IF MS NOT UP.
2850 016112  005301                 DEC     R1              ;DECREMENT THE MS TIME COUNT.
2851 016114  001363                 BNE     8$              ;KEEP LOOPING.
2852 016116  004737  022364         JSR     PC,OOPS         ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
2853
2854                            ;+
2855                            ; We have now have loop count information for one clock tick.
2856                            ; We have negative of number of outer loops in R1, each is MSLCNT inner loops.
2857                            ; We have the portion of the last outer loop not executed, in R5.
2858                            ; Now we calculate the total number of inner loops executed.
2859 016122  005401          12$:   NEG     R1              ;GET NUMBER OF OUTER LOOPS.
2860 016124  013702  002314         MOV     MSLCNT,R2       ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2861 016130  010203                 MOV     R2,R3           ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2862 016132  160502                 SUB     R5,R2           ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2863 016134  010204                 MOV     R2,R4           ;   AND ADD TO ACCUMULATOR LSWORD.
2864 016136  005005                 CLR     R5              ;CLEAR ACCUMULATOR MSWORD.
2865 016140  005301          14$:   DEC     R1              ;CHECK R1 FOR 0 CONDITION
2866 016142  100403                 BMI     16$             ;  SKIP MULTIPLICATION IF ZERO
2867 016144  060304                 ADD     R3,R4           ;MULTIPLY NUMBER OF INNER
2868 016146  005505                 ADC     R5              ;  LOOPS PER OUTER LOOP BY
2869 016150  000773                 BR      14$             ;NUMBER OF OUTER LOOPS PERFORMED.
2870
2871                            ;+
2872                            ; Divide the total number of inner loops by the number of MS per LTC tick.
2873 016152  013701  002312  16$:   MOV     MSTICK,R1       ;# OF MS PER LTC TICK IS DIVISOR.
2874 016156  010403                 MOV     R4,R3           ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2875 016160  010502                 MOV     R5,R2           ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2876 016162  004737  026530         JSR     PC,UNSDIV       ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
2877 016166  103402                 BCS     18$             ;BYPASS OOPS IF WE'RE OK.
2878 016170  004737  022364         JSR     PC,OOPS         ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2879 016174  010137  002314  18$:   MOV     R1,MSLCNT       ;SET NEW VALUE FOR MS LOOP COUNT.
2880 016200  005137  016214         COM     62$             ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2881 016204  001277                 BNE     2$              ;BRANCH IF ONLY ONE ITERATION DONE.
2882 016206  000261                 SEC                     ;SET THE SUCCESS FLAG FOR EXIT.
2883
2884 016210                  60$:   PASS                    ;RESTORE GPRS,
     016210  004736                         JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
2885 016212  000207                 RTS     PC              ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
2886
2887 016214  000000          62$:   .WORD   0               ;2ND CALIBRATION ITERATION FLAGS.
2888 016216  000000          64$:   .WORD   0               ;DUMMY WORD FOR STORAGE OF THE READ WORD.
```

```
2890                                   .SBTTL  GLOBAL SUBROUTINE                - CHKEXT -
2891                                   ;++ *********************************************************************
2892                                   ;*                   - Check For Extra Character Routine -
2893                                   ;*        This subroutine checks for the condition which indicates that an extra
2894                                   ;*        character has been received during the reception of a data pattern.
2895                                   ;*        If this routine determines that it is likely that an extra character
2896                                   ;*        has been received it indicates this in the status information returned
2897                                   ;*        to the calling routine.
2898                                   ;*
2899                                   ;* INPUTS:        R3 - RX line number multiplied by 2 (offset into word tables).
2900                                   ;*                R4 - Base address of resync que containing RX chars.
2901                                   ;*                R5 - Mask of "inactive" (non-data) bits of RX and TX chars.
2902                                   ;*                CHCNTB - Base of number of chars to TX on each line table.
2903                                   ;*                RXCNTB - Base of the RX character counters table.
2904                                   ;*                RXPTRB - Base of the RX character pointers table.
2905                                   ;*                TXRXLB - Base of TX/RX line number association table.
2906                                   ;*
2907                                   ;* OUTPUTS:       CARRY - Set if extra character condition is verified.
2908                                   ;*
2909                                   ;* CALLING SEQUENCE:    JSR     PC,CHKEXT
2910                                   ;*
2911                                   ;* COMMENTS:  The following symbols are used in line comments:
2912                                   ;*                CHR0 - Character at bottom of resync que (first received).
2913                                   ;*                CHR1, CHR2 - 2 characters received after CHR0.
2914                                   ;*                EXP0 - Character expected to be received next.
2915                                   ;*                EXP1, EXP2 - Character expected to be received after EXP0, etc.
2916                                   ;*
2917                                   ;* SUBORDINATE ROUTINES CALLED: None.
2918                                   ;-- *********************************************************************
2919 016220                           CHKEXT:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
     016220  004537  005326                    JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2920 016224  016302  003404                    MOV     RXPTRB(R3),R2   ;GET THE RX DATA POINTER.
2921 016230  005724                            TST     (R4)+           ;INCREMENT R4 BY 2 TO POINT TO CHR1.
2922 016232  012400                            MOV     (R4)+,R0        ;GET CHR1 FROM THE QUE, DATA.VALID INTO N FLAG.
2923 016234  100026                            BPL     52$             ;EXIT WITH "FAILURE" IF CHR1 NOT VALID.
2924 016236  040500                            BIC     R5,R0           ;REMOVE INACTIVE BITS FROM CHR1 VALUE.
2925 016240  112201                            MOVB    (R2)+,R1        ;GET EXP0 FROM THE DATA PATTERN.
2926 016242  040501                            BIC     R5,R1           ;REMOVE INACTIVE BITS FROM EXP0 VALUE.
2927 016244  120100                            CMPB    R1,R0           ;COMPARE CHR1 AND EXP0.
2928 016246  001021                            BNE     52$             ;EXIT WITH "FAILURE" IF CHR1 <> EXP0.
2929 016250  016300  003544                    MOV     RXCNTB(R3),R0   ;COMPARE THE PRESENT RX CHARACTER COUNT PLUS 1
2930 016254  005200                            INC     R0              ;   WITH THE EXPECTED NUMBER OF CHARS TO RX ON
2931 016256  016301  005236                    MOV     TXRXLB(R3),R1   ;   LINE (NUMBER TRANSMITTED AND LOOPED BACK) TO
2932 016262  020061  003444                    CMP     R0,CHCNTB(R1)   ;   DETERMINE IF CHR1 IS LAST EXPECTED CHAR.
2933 016266  001407                            BEQ     50$             ;EXIT WITH "SUCCESS" IF CHR1 IS LAST CHAR.
2934 016270  011400                            MOV     (R4),R0         ;GET CHR2 FROM THE QUE, DATA.VALID INTO N FLAG.
2935 016272  100005                            BPL     50$             ;EXIT WITH "SUCCESS" IF CHR1 WAS LAST IN QUE.
2936 016274  040500                            BIC     R5,R0           ;REMOVE INACTIVE BITS FROM CHR2 VALUE.
2937 016276  111201                            MOVB    (R2),R1         ;GET THE EXP1 VALUE.
2938 016300  040501                            BIC     R5,R1           ;REMOVE INACTIVE BITS FROM EXP1 VALUE.
2939 016302  020001                            CMP     R0,R1           ;COMPARE CHR2 AND EXP1.
2940 016304  001002                            BNE     52$             ;EXIT WITH "FAILURE" IF CHR2 <> EXP1.
2941
2942
2943                                   ;+
2944                                   ; It is likely that we received an extra character within the data pattern.
2945                                   ; Indicate "success" and exit.
                                       ;-
```

```
2946 016306  000261          50$:    SEC                        ;SET THE SUCCESS FLAG.
2947 016310  000401                  BR      60$                ;EXIT THE ROUTINE.
2948
2949                         ;+
2950                         ; We didn't receive a single extra character at this point in the data pattern.
2951                         ; Indicate "failure" and exit.
2952                         ;-
2953 016312  000241          52$:    CLC                        ;CLEAR THE SUCCESS FLAG.
2954
2955 016314                  60$:    PASS                       ;RESTORE GPRS.
     016314  004736                  JSR     PC,@(SP)+          ;RETURN TO PREGO5 SUBRT.
2956 016316  000207                  RTS     PC                 ;CARRY - SET IF SUCCESS (EXTRA CHAR RXED).
```

```
2958                                    .SBTTL   GLOBAL SUBROUTINE                      - CHKLOS -
2959                                    ;++ *******************************************************************************
2960                                    ;*                       - Check For Lost Character Routine -
2961                                    ;*        This subroutine checks for the condition which indicates that a char
2962                                    ;*        has been "lost" from the looped back data pattern during a transmission
2963                                    ;*        and reception test.  If this routine determines that it is likely that
2964                                    ;*        a character has been lost, it indicates this in the status information
2965                                    ;*        returned to the calling routine.
2966                                    ;*
2967                                    ;* INPUTS:         R3 - RX line number multiplied by 2 (offset into word tables).
2968                                    ;*                 R4 - Base address of resync que containing RX chars.
2969                                    ;*                 R5 - Mask of "inactive" (non-data) bits of RX and TX chars with
2970                                    ;*                       all set bits in a single, left justified group.
2971                                    ;*                 CHCNTB - Base of number of chars to TX on each line table.
2972                                    ;*                 RXCNTB - Base of the RX character counters table.
2973                                    ;*                 RXPTRB - Base of the RX character pointers table.
2974                                    ;*                 TXRXLB - Base of TX/RX line number association table.
2975                                    ;*
2976                                    ;* OUTPUTS:        CARRY - Set if lost character condition is verified.
2977                                    ;*
2978                                    ;* CALLING SEQUENCE:    JSR     PC,CHKLOS
2979                                    ;*
2980                                    ;* COMMENTS:   The following symbols are used in line comments:
2981                                    ;*                 CHR0 - Character at bottom of resync que (first received).
2982                                    ;*                 CHR1, CHR2 - 2 characters received after CHR0.
2983                                    ;*                 EXP0 - Character expected to be received next.
2984                                    ;*                 EXP1, EXP2 - Character expected to be received after EXP0, etc.
2985                                    ;*
2986                                    ;* SUBORDINATE ROUTINES CALLED: None.
2987                                    ;-- *******************************************************************************
2988 016320                            CHKLOS:: SAVE                                 ;SAVE CONTENTS OF GPRS R0 THRU R5.
     016320   004537   005326                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
2989 016324   016301   003544                MOV     RXCNTB(R3),R1   ;COMPARE THE PRESENT RX CHARACTER COUNT PLUS 1
2990 016330   005201                          INC     R1              ;   WITH THE EXPECTED NUMBER OF CHARS TO RX ON
2991 016332   016300   005236                MOV     TXRXLB(R3),R0   ;   LINE (NUMBER TXED AND LOOPED BACK) TO
2992 016336   016002   003444                MOV     CHCNTB(R0),R2   ;   DETERMINE IF THE POSSIBLE LOST CHAR
2993 016342   020102                          CMP     R1,R2           ;   WOULD BE THE LAST EXPECTED RX CHAR.
2994 016344   001423                          BEQ     52$             ;EXIT WITH "FAILURE" IF LOST CHR WOULD BE LAST.
2995 016346   005201                          INC     R1              ;DETERMINE (AS ABOVE) IF CHR0 WOULD BE THE LAST
2996 016350   160201                          SUB     R2,R1           ;   RX CHAR AND SAVE RESULT FOR LATER.
2997 016352   016302   003404                MOV     RXPTRB(R3),R2   ;GET THE RX DATA POINTER.
2998 016356   005202                          INC     R2              ;CALCULATE POINTER TO EXP1 LOCATION.
2999 016360   112200                          MOVB    (R2)+,R0        ;GET EXP1 VALUE FROM DATA PATTERN.
3000 016362   162400                          SUB     (R4)+,R0        ;COMPARE CHR0 AND EXP1 VALUES.
3001 016364   040500                          BIC     R5,R0           ;REMOVE INACTIVE BITS FROM RESULT.  (NO ACTIVE
3002                                                                  ;   BITS ALLOWED TO LEFT OF ANY INACTIVE BITS.)
3003 016366   001012                          BNE     52$             ;EXIT WITH "FAILURE" IF CHR0 <> EXP1.
3004 016370   005701                          TST     R1              ;CHECK CHR0 TEST RESULT SAVED ABOVE.
3005 016372   001406                          BEQ     50$             ;EXIT WITH "SUCCESS" IF CHR0 IS LAST CHAR.
3006 016374   011401                          MOV     (R4),R1         ;GET CHR1 FROM THE QUE, DATA.VALID INTO N FLAG.
3007 016376   100004                          BPL     50$             ;EXIT WITH "SUCCESS" IF CHR0 WAS LAST QUE CHAR.
3008 016400   111200                          MOVB    (R2),R0         ;GET THE EXP2 VALUE FROM THE DATA PATTERN.
3009 016402   160001                          SUB     R0,R1           ;COMPARE THE EXP2 AND THE CHR1 VALUES.
3010 016404   040501                          BIC     R5,R1           ;REMOVE INACTIVE BITS FROM RESULT OF COMPARE.
3011                                                                  ;   (NO ACTIVE BITS LEFT OF INACTIVE BITS.)
3012 016406   001002                          BNE     52$             ;EXIT WITH "FAILURE" IF CHR1 <> EXP2.
3013
```

```
3014                                    ;*
3015                                    ; It is likely that we lost a character from the data pattern.
3016                                    ; Indicate "success" and exit.
3017                                    ;-
3018 016410  000261         50$:    SEC                 ;SET THE SUCCESS FLAG.
3019 016412  000401                 BR      60$         ;EXIT THE ROUTINE.
3020
3021                                    ;*
3022                                    ; We didn't lose a single extra character at this point in the data pattern.
3023                                    ; Indicate "failure" and exit.
3024                                    ;-
3025 016414  000241         52$:    CLC                 ;CLEAR THE SUCCESS FLAG.
3026
3027 016416                 60$:    PASS                ;RESTORE GPRS.
     016416  004736                         JSR     PC,@(SP)+         ;RETURN TO PREG05 SUBRT.
3028 016420  000207                 RTS     PC          ;CARRY - SET IF SUCCESS (LOST CHAR LIKELY).
```

```
3030                              .SBTTL  GLOBAL SUBROUTINE                - CHRMSK -
3031                              ;++ *********************************************************************
3032                              ;*              - Form a Bit Mask of Unused TX/RX Bits Routine -
3033                              ;*       This subroutine constructs a bit mask of character bits which are not
3034                              ;*       used during transmission and reception.  This mask can be used
3035                              ;*       to remove the flags, line number, DATA.VALID bits, and unused data bits
3036                              ;*       from a character word which has been read from the DUT FIFO.
3037                              ;*
3038                              ;* INPUTS:       R1 - DUT LPR contents used to determine character length.
3039                              ;*
3040                              ;* OUTPUTS:      IBM - Bit mask of unused TX/RX bits (including upper byte):
3041                              ;*                    Examples: 177400 returned for 8 bits/char.
3042                              ;*                              177700 returned for 6 bits/char.
3043                              ;*
3044                              ;* CALLING SEQUENCE:    JSR      PC,CHRMSK
3045                              ;*
3046                              ;* COMMENTS:     If this mask is to be used to just remove the inactive bits
3047                              ;*               within the data byte of a word read from the DUT FIFO, the
3048                              ;*               upper byte of the mask must be cleared.
3049                              ;*
3050                              ;* SUBORDINATE ROUTINES CALLED: None.
3051                              ;-- *********************************************************************
3052
3053 016422                      CHRMSK:: SAVE                              ;SAVE CONTENTS OF GPRS RO THRU R5.
     016422  004537  005326                       JSR     R5,PREGO5        ;CALL REGISTER SAVE SUBRT.
3054 016426  052701  177740               BIS     #177740,R1              ;PREPARE TO COUNT BITS SHIFTED INTO MASK BY
3055                                                                       ;  USING THE LPR BITS/CHAR FIELD CONTENTS.
3056 016432  012703  177400               MOV     #177400,R3              ;CLEAR THE UNUSED BIT MAP LOWER BYTE.
3057 016436  062701  000010      2$:      ADD     #10,R1                  ;DETERMINE IF ANOTHER BIT WOULD BE TOO MANY.
3058 016442  103402                       BCS     4$                      ;EXIT THE SHIFT LOOP IF IT WOULD BE TOO MANY.
3059 016444  006203                       ASR     R3                      ;SHIFT A BIT INTO THE UNUSED BIT MASK LOW BYTE.
3060 016446  000773                       BR      2$                      ;LOOP TO CHECK FOR DONE.
3061
3062 016450  010337  002226      4$:      MOV     R3,IBM                  ;LOAD THE INACTIVE BITS MASK STORAGE IN MEMORY.
3063
3064 016454                      60$:     PASS                            ;RESTORE GPRS.
     016454  004736                       JSR     PC,@(SP)+                         ;RETURN TO PREGO5 SUBRT.
3065 016456  000207                       RTS     PC
```

```
3067                                    .SBTTL  GLOBAL SUBROUTINE              - CKCHR -
3068                                    ;++ *****************************************************************************
3069                                    ;*                  - Check Character For Errors Routine -
3070                                    ;*        This subroutine checks the character at the bottom of the resync queue
3071                                    ;*        to determine if it is correct.  Pointers and counters which are related
3072                                    ;*        to the reception of the character are updated.  If the character is
3073                                    ;*        incorrect, an analysis of the error is done and parameters are set up
3074                                    ;*        for the reporting of the correct error.
3075                                    ;*
3076                                    ;* INPUTS:       R3 - Line offset for access of word tables of line variables.
3077                                    ;*               R4 - Base address of the resync queue for this line.
3078                                    ;*               R5 - Mask of the inactives bits in a TX or RX char byte.
3079                                    ;*               BITTBL - Table of words with bits set for use in forming maps.
3080                                    ;*               DPRSQ - Data Pattern Resync Que with valid char at bottom.
3081                                    ;*               EXCNTB - Base of the extra character couters table.
3082                                    ;*               RXDONF - Receive done flags.
3083                                    ;*               RXPTRB - Base of the RX character pointers table.
3084                                    ;*               Error Message Labels - EM9007,EM9008,EM9027,EM9028
3085                                    ;*
3086                                    ;* OUTPUTS:      R1 - Contains the address of the error message to be reported.
3087                                    ;*               R2 - Contains the actual received data.
3088                                    ;*               R4 - Contains the expected data.
3089                                    ;*               CARRY - "Success" flag (set if no error is found).
3090                                    ;*            Following variables updated for line on which char was received:
3091                                    ;*               EXCNT - Count of the number of extra chars received on line.
3092                                    ;*               RXCNT - Count of the number of characters received on line.
3093                                    ;*               RXPTR - Updated to point to the next expected char on line.
3094                                    ;*               ERRBLK - Contents destroyed.
3095                                    ;*
3096                                    ;* CALLING SEQUENCE:    JSR     PC,CKCHR
3097                                    ;*
3098                                    ;* COMMENTS:
3099                                    ;*
3100                                    ;* SUBORDINATE ROUTINES CALLED: CHKEXT,CHKLOS,UPDCHR.
3101                                    ;-- *****************************************************************************
3102 016460                            CKCHR:: SAVE                           ;SAVE CONTENTS OF GPRS RO THRU R5.
     016460  004537  005326                    JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
3103                                    ;+
3104                                    ; Check for the RX of a char after RX should be complete on this line.
3105                                    ;-
3106 016464  036337  002366  002506            BIT     BITTBL(R3),RXDONF       ;TEST THE RX DONE FLAG FOR THIS LINE.
3107 016472  001407                            BEQ     2$                      ;SKIP ERROR REPORT IF RX NOT COMPLETE ON LINE.
3108                                    ;+
3109                                    ; We have received an extra character on this line.
3110                                    ; Set up for error report and exit to report the error.
3111                                    ; Count the extra character.
3112                                    ; Exit to report "UNEXPECTED CHAR RECEIVED AFTER RX COMPLETE ON LINE nn"
3113                                    ;-
3114 016474  012701  011414                    MOV     #EM9007,R1      ;SELECT "EXTRA CHAR ON LINE" ERROR MESSAGE.
3115 016500  011402                            MOV     (R4),R2         ;GET THE ACTUAL DATA FOR ERROR REPORT.
3116 016502  040502                            BIC     R5,R2           ;REMOVE THE INACTIVE BITS.
3117 016504  052704  100000                    BIS     #BIT15,R4       ;INDICATE "NONE" EXPECTED DATA FOR ERROR RPT.
3118 016510  000452                            BR      12$             ;GO COUNT EXTRA CHAR AND EXIT WITH "FAILURE".
3119                                    ;+
3120                                    ; Get the pointer to the next expected receive data character.
3121                                    ;-
3122 016512  016302  003404            2$:     MOV     RXPTRB(R3),R2
```

```
3123                                     ;+
3124                                     ; Compare the actual data with the expected data.
3125                                     ;-
3126 016516  011400                              MOV    (R4),R0          ;GET THE ACTUAL DATA.
3127 016520  040500                              BIC    R5,R0            ;REMOVE THE INACTIVE BITS.
3128 016522  111201                              MOVB   (R2),R1          ;GET THE EXPECTED DATA.
3129 016524  040501                              BIC    R5,R1            ;REMOVE THE INACTIVE BITS.
3130 016526  120001                              CMPB   R0,R1            ;COMPARE ACTUAL AND EXPECTED.
3131 016530  001003                              BNE    4$               ;CHECK FURTHER IF DATA MISCOMPARE.
3132 016532  004737  026664                      JSR    PC,UPDCHR        ;UPDATE PTRS AND COUNTERS FOR THE CHAR.
3133 016536  000446                              BR     50$              ;EXIT WITH "SUCCESS", NO ERROR FOUND.
3134                                     ;+
3135                                     ; Actual and expected data miscompare.
3136                                     ; Determine if it's likely we received an extra char within the data pattern.
3137                                     ;-
3138 016540  004737  016220      4$:             JSR    PC,CHKEXT        ;CHECK FOR EXTRA CHAR RX'ED IN PATTERN.
3139 016544  103010                              BCC    6$               ;GO CHECK FOR LOST CHAR IF NO EXTRA CHAR.
3140                                     ;+
3141                                     ; It is likely that we received an extra character within the data pattern.
3142                                     ; Count the char as an extra char, don't count as a standard char.
3143                                     ; Report "EXTRA CHAR RECEIVED WITHIN DATA PATTERN ON LINE nn"
3144                                     ;-
3145 016546  012701  012227                      MOV    #EM9027,R1       ;SELECT "EXTRA CHAR ON LINE" ERROR MSG.
3146 016552  111200                              MOVB   (R2),R0          ;GET THE EXPECTED RECEIVE DATA.
3147 016554  040500                              BIC    R5,R0            ;REMOVE THE INACTIVE BITS FROM EXPECTED DATA.
3148 016556  011402                              MOV    (R4),R2          ;GET THE ACTUAL RECEIVE DATA.
3149 016560  040502                              BIC    R5,R2            ;REMOVE THE INACTIVE BITS FROM ACTUAL DATA.
3150 016562  010004                              MOV    R0,R4            ;PASS EXPECTED DATA TO ERROR REPORT ROUTINE.
3151 016564  000424                              BR     12$              ;GO COUNT EXTRA CHAR AND EXIT WITH "FAILURE".
3152                                     ;+
3153                                     ; Actual and expected data miscompare.
3154                                     ; Not likely that we received an extra character within the data pattern.
3155                                     ; Determine if it's likely we lost a character from the data pattern.
3156                                     ;-
3157 016566  004737  016320      6$:             JSR    PC,CHKLOS        ;CHECK FOR A LOST CHAR CONDITION.
3158 016572  103012                              BCC    8$               ;GO REPORT BAD RX DATA IF NOT LOST CHAR.
3159                                     ;+
3160                                     ; It is likely that we lost a character from the data pattern.
3161                                     ; Count the char in the RX char count as if it had been received.
3162                                     ; Also, count CHR0 as a valid char, because we have verified it above.
3163                                     ; Report "SINGLE CHAR MISSING FROM RECEIVED DATA ON LINE nn"
3164                                     ;-
3165 016574  012701  012307                      MOV    #EM9028,R1       ;SELECT "LOST CHAR ON LINE" ERROR MSG. +++++
3166 016600  111200                              MOVB   (R2),R0          ;GET THE EXPECTED RECEIVE DATA.
3167 016602  040500                              BIC    R5,R0            ;REMOVE THE INACTIVE BITS FROM EXPECTED DATA.
3168 016604  011402                              MOV    (R4),R2          ;GET THE ACTUAL RECEIVE DATA.
3169 016606  040502                              BIC    R5,R2            ;REMOVE THE INACTIVE BITS FROM ACTUAL DATA.
3170 016610  010004                              MOV    R0,R4            ;PASS EXPECTED DATA TO ERROR REPORT ROUTINE.
3171 016612  004737  026664                      JSR    PC,UPDCHR        ;UPDATE PTRS AND COUNTERS FOR THE CHAR.
3172 016616  000404                              BR     10$              ;GO EXIT WITH "FAILURE".
3173                                     ;+
3174                                     ; Did not lose or gain a single character from/to the data pattern.
3175                                     ; Report "RECEIVED CHAR MISCOMPARE AGAINST TX DATA ON LINE nn"
3176                                     ;-
3177 016620  010002              8$:             MOV    R0,R2            ;PASS ACTUAL DATUM TO ERROR REPORT ROUTINE.
3178 016622  010104                              MOV    R1,R4            ;PASS EXPECTED DATUM TO ERROR REPORT ROUTINE.
3179 016624  012701  011477                      MOV    #EM9008,R1       ;SELECT THE "DATA MISCOMPARE" MESSAGE.
```

```
3180                                 ;+
3181                                 ; Update the character counter and RX data pattern pointer for this line.
3182                                 ;-
3183 016630  004737  026664   10$:    JSR     PC,UPDCHR          ;UPDATE RX PTR AND COUNTER FOR THIS LINE.
3184 016634  000405                    BR      14$               ;GO EXIT WITH "FAILURE".
3185                                 ;+
3186                                 ; Count the character as an extra character.
3187                                 ;-
3188 016636  005263  003244   12$:    INC     EXCNTB(R3)        ;INCREMENT THE EXTRA CHAR COUNT FOR THIS LINE.
3189 016642  001002                    BNE     14$               ;EXIT WITH FAILURE IF NO OVERFLOW.
3190 016644  005363  003244            DEC     EXCNTB(R3)        ;DECREMENT BACK TO -1 (MAX VALUE) IF OVERFLOW.
3191                                 ;+
3192                                 ; Indicate "failure" and exit.
3193                                 ;-
3194 016650  000241           14$:    CLC                       ;CLEAR THE "SUCCESS" FLAG.
3195 016652  000401                    BR      60$               ;EXIT THE ROUTINE.
3196
3197                                 ;+
3198                                 ; No error was found.
3199                                 ; Set "success" flag and exit.
3200                                 ;-
3201 016654  000261           50$:    SEC                       ;SET THE "SUCCESS" FLAG.
3202
3203 016656                   60$:    PASS    R1,R2,R4          ;RESTORE GPRS, EXCEPT
     016656  010166  000004            MOV     R1,R1SLOT(SP)        ;PUT R1 IN STACK SLOT.
     016662  010266  000006            MOV     R2,R2SLOT(SP)        ;PUT R2 IN STACK SLOT.
     016666  010466  000012            MOV     R4,R4SLOT(SP)        ;PUT R4 IN STACK SLOT.
     016672  004736                    JSR     PC,@(SP)+            ;RETURN TO PREG05 SUBRT.
3204                                                             ;R1 - CONTAINS THE ADDRESS OF THE ERROR REPORT.
3205                                                             ;R2 - CONTAINS THE ACTUAL DATA RECEIVED.
3206                                                             ;R4 - CONTAINS THE EXPECTED DATA.
3207 016674  000207                    RTS     PC
```

```
3209                              .SBTTL  GLOBAL SUBROUTINE                - CKFRPR -
3210                              ;++ *********************************************************************
3211                              ;*          - Check Framing and Parity Error reporting -
3212                              ;*      This subroutine is used in the Framing error and Parity error tests.
3213                              ;*      It reads the characters from the DUT Receiver Character FIFO,
3214                              ;*      and checks for the correct combination of Parity and Framing
3215                              ;*      error bits in the MSB.  If characters stop appearing in the FIFO with
3216                              ;*      DATA.VALID set or if more than the allowable number of characters
3217                              ;*      has been read from the DUT this routine exits with an RX complete
3218                              ;*      indication.  Each read char is analysed and any necessary errors are
3219                              ;*      reported.
3220                              ;*
3221                              ;* INPUTS:      R5 - Test flag, bit15 set = Framing err, clear = Parity err.
3222                              ;*              ERRNBR - Set to error number of first error in this routine.
3223                              ;*              OSTEND - Address of the end of the output storage fifo buffer.
3224                              ;*              OSTPTR - Pointer to the next byte to read from OSTORE.
3225                              ;*
3226                              ;* OUTPUTS:     RXCNTB - Receive charater count updated for each line.
3227                              ;*              RXPNTB - Receive character pionter is updated for each line.
3228                              ;*
3229                              ;* CALLING SEQUENCE:    JSR     PC,CKFRPR
3230                              ;*
3231                              ;* COMMENTS:    This routine reports errors with numbers Initial ERRNBR
3232                              ;*              thru Initial ERRNBR + 4.
3233                              ;*              ERRNBR is restored before this routine returns.
3234                              ;*
3235                              ;* SUBORDINATE ROUTINES CALLED: PRFRME,PRPARE,WAIBIS.
3236                              ;-- *********************************************************************
3237
3238 016676                      CKFRPR:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
     016676  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3239 016702  013704  005320              MOV     ERRNBR,R4       ;PRESERVE THE INITIAL ERROR NUMBER.
3240 016706  004737  026070              JSR     PC,TXIE1        ;ENABLE TX INTERRUPTS.
3241                              ;+
3242                              ; Wait for a character to appear in the FIFO.
3243                              ; If no character appears within time-out period: exit routine, we're done.
3244                              ;-
3245 016712  013701  002242              MOV     RXTOUT,R1       ;GET MINIMUM TIME OUT VALUE.
3246 016716  023737  002504  002174  2$:  CMP     TXDONF,ACTLNS   ;CHECK FOR TRANSMISSION DONE ON ACTIVE LINES.
3247 016724  001402                      BEQ     4$              ;SKIP ADDING 50 MS DELAY IF TX DONE ALL LINES.
3248 016726  062701  000062              ADD     #50.,R1         ;ADD 50 MILLI SEC TO DELAY IF NOT LAST CHAR.
3249 016732  052701  170000  4$:         BIS     #170000,R1      ;INDICATE TO TEST DATA.VALID BIT.
3250 016736  013702  002204              MOV     RBUFA,R2        ;INDICATE TO CHECK DUT RECEIVE BUFFER (FIFO).
3251 016742  004737  027160              JSR     PC,WAIBIS       ;WAIT FOR RECEIVED CHAR OR TIME-OUT.
3252 016746  103033                      BCC     60$             ;EXIT ROUTINE IF TIME-OUT, WE'RE DONE.
3253
3254 016750  005337  002500              DEC     CHRTOT          ;DECREMENT THE TOTAL CHAR COUNTER.
3255 016754  001011                      BNE     6$              ;SKIP ERROR IF NOT TOO MANY CHARS RECEIVED.
3256 016756  010437  005320              MOV     R4,ERRNBR       ;SET ERROR NUMBER TO INITIAL ERRNBR.
3257 016762  012701  012107              MOV     #EM9025,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
3258 016766  012737  014100  005324      MOV     #ERO503,ERRBLK  ;SELECT THE ERROR REPORT ROUTINE.
3259                              ;+
3260                              ;     Report error at initial ERRNBR.
3261                              ;     "MORE THAN TWICE THE EXPECTED NUMBER OF CHARACTERS RECEIVED"
3262                              ;-
3263 016774                      ERROR           ;                >>>>> ERROR <<<<<.
     016774  104460                              TRAP    C$ERROR
```

```
3264 016776  000417                          BR     60$              ;EXIT THIS ROUTINE WE HAVE GIVEN UP.
3265
3266                              ;+
3267                              ; Extract the line number of the new character.
3268                              ; Calculate offset for accessing tables of line variables.
3269                              ;-
3270 017000  010203              6$:         MOV    R2,R3            ;COPY THE READ CHARACTER.
3271 017002  000303                          SWAB   R3               ;GET THE LINE NUMBER IN THE LSB.
3272 017004  042703   177760                 BIC    #177760,R3       ;CLEAR THE UNWANTED BITS.
3273 017010  006303                          ASL    R3               ;SHIFT LEFT TO FORM OFFSET INTO TABLES.
3274                              ;+
3275                              ; Process the read characters as dictated by the test flag.
3276                              ;-
3277 017012  010505                          MOV    R5,R5            ;DETERMIN WHICH TEST CALLED THIS ROUTINE.
3278 017014  100003                          BPL    8$               ;BRANCH TO PROCESS CHARACTER IN PARITY TEST.
3279
3280 017016  004737   022612                 JSR    PC,PRFRME        ;PROCESS FRAMING ERRORS RECEIVED.
3281 017022  000402                          BR     10$              ;SKIP PROCESSING CHARACTERS FOR PARITY TEST.
3282 017024  004737   022710     8$:         JSR    PC,PRPARE        ;PROCESS PARITY ERRORS RECEIVED.
3283
3284 017030  004737   026664     10$:        JSR    PC,UPDCHR        ;UPDATE POINTERS AND COUNTERS FOR THIS LINE.
3285 017034  000730                          BR     2$               ;LOOP TO READ NEXT CHAR FROM FIFO.
3286
3287 017036  010437   005320     60$:        MOV    R4,ERRNBR        ;RESTORE THE ERROR NUMBER TO ITS INITIAL VALUE.
3288 017042                                  PASS                    ;RESTORE GPRS.
     017042  004736                                  JSR    PC,@(SP)+                ;RETURN TO PREGO5 SUBRT.
3289 017044  000207                          RTS    PC
```

```
3291                                    .SBTTL  GLOBAL SUBROUTINE             - CKINAC -
3292                                 ;+' ***********************************************************************
3293                                 ;*            - Check for New Character on Inactive Line Routine -
3294                                 ;*      This subroutine checks a character to determine if the character
3295                                 ;*      was received on an active line.  If the character was received on
3296                                 ;*      an inactive line this routine records the fact that the character
3297                                 ;*      was received on an inactive line, prepares an error message for
3298                                 ;*      the calling routine, and returns a "failure" status.
3299                                 ;*
3300                                 ;* INPUTS:      R2 - The RX character including error flags and line number.
3301                                 ;*              ACTLNS - Bit map of active DUT lines.
3302                                 ;*              BITTBL - Table of words with bits set for forming bit maps.
3303                                 ;*              EM9006 - Label at "RX ON INACTIVE LINE" error messge.
3304                                 ;*              EXCNTB - Base of the extra character couters table.
3305                                 ;*              TXRXLB - Base of TX/RX line number association table.
3306                                 ;*
3307                                 ;* OUTPUTS:     CARRY - "Success" flag (set if no error found).
3308                                 ;*              R1 - If error found, address of error message.
3309                                 ;*              R3 - Line number offset of passed in character.
3310                                 ;*              R4 - If error found, expected data indication for error rpt.
3311                                 ;*              EXCNT - Extra character count for line (Updated if error).
3312                                 ;*
3313                                 ;* CALLING SEQUENCE:    JSR     PC,CKINAC
3314                                 ;*
3315                                 ;* COMMENTS:
3316                                 ;*
3317                                 ;* SUBORDINATE ROUTINES CALLED: NONE.
3318                                 ;-- ***********************************************************************
3319
3320 017046                         CKINAC:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
     017046  004537  005326                          JSR     R5,PREG05      ;CALL REGISTER SAVE SUBRT.
3321                                 ;+
3322                                 ; Extract the line number from the passed in character and use the line
3323                                 ;   number to form an offset for accessing tables of line variables.
3324                                 ;-
3325 017052  010203                          MOV     R2,R3          ;EXTRACT THE LINE NUMBER
3326 017054  000303                          SWAB    R3             ;  FROM THE CHARACTER WE
3327 017056  042703  177760                  BIC     #177760,R3     ;  ARE COMPARING.
3328 017062  006303                          ASL     R3             ;FORM OFFSET INTO WORD TABLE FROM LINE NUMBER.
3329                                 ;+
3330                                 ; If the character in question is not a valid character, exit with "success".
3331                                 ;-
3332 017064  005702                          TST     R2             ;CHECK DATA.VALID BIT.
3333 017066  100021                          BPL     50$            ;EXIT WITH SUCCESS IF CHAR IS NOT VALID.
3334                                 ;+
3335                                 ; If the TX line which is associated with this RX line is an active line,
3336                                 ;   exit the routine with "success".
3337                                 ;-
3338 017070  016301  005236                  MOV     TXRXLB(R3),R1  ;GET THE TX LINE # OFFSET FOR THIS RX LINE.
3339 017074  036137  002366  002174          BIT     BITTBL(R1),ACTLNS  ;DETERMINE IF TX LINE IS AN ACTIVE LINE.
3340 017102  001013                          BNE     50$            ;EXIT ROUTINE WITH SUCCESS IF LINE IS ACTIVE.
3341                                 ;+
3342                                 ; The character in question was received on an inactive line.
3343                                 ; Count this character as an extra char.
3344                                 ; Set up error information.
3345                                 ; Exit routine with "failure" indication.
3346                                 ;-
```

```
3347 017104  005263  003244              INC    EXCNTB(R3)      ;INCREMENT THE EXTRA CHAR COUNT FOR THIS LINE.
3348 017110  001002                      BNE    2$              ;SKIP SETTING TO MAX VALUE IF NO OVERFLOW.
3349 017112  005363  003244              DEC    EXCNTB(R3)      ;DECREMENT BACK TO -1 (MAX VALUE) IF OVERFLOW.
3350 017116  012701  011341        2$:   MOV    @EM9006,R1      ;SET UP RX ON INACTIVE LINE MESSAGE.
3351 017122  012704  100000              MOV    @BIT15,R4       ;SET UP "NONE" EXPECTED DATA INDICATION.
3352 017126  000241                      CLC                    ;CLEAR THE "SUCCESS" FLAG.
3353 017130  000401                      BR     60$             ;GO REPORT RX CHAR ON INACTIVE LINE.
3354
3355                                ;+
3356                                ; We have not found a "char on inactive line" error situation.
3357                                ; Set the "success" flag and exit the routine.
3358                                ;-
3359 017132  000261                50$:  SEC                    ;SET THE "SUCCESS" FLAG.
3360
3361 017134                         60$:  PASS   R1,R3,R4        ;RESTORE GPRS, EXCEPT OUTPUT GPRS.
     017134  010166  000004                     MOV    R1,R1SLOT(SP)        ;PUT R1 IN STACK SLOT.
     017140  010366  000010                     MOV    R3,R3SLOT(SP)        ;PUT R3 IN STACK SLOT.
     017144  010466  000012                     MOV    R4,R4SLOT(SP)        ;PUT R4 IN STACK SLOT.
     017150  004736                              JSR    PC,@(SP)+            ;RETURN TO PREGO5 SUBRT.
3362 017152  000207                      RTS    PC              ;CARRY - SUCCESS FLAG (SET IF NO ERROR).
```

```
3364                                .SBTTL  GLOBAL SUBROUTINE                 - CKTRAP -
3365                                ;*****************************************************************************
3366                                ;*        Check Trap Routine -
3367                                ;*        This subroutine is used to check for a bus time-out trap (004 trap)
3368                                ;*        which is caused by an access to a non-existent memory or I/O location.
3369                                ;*        If the trap does not occur, this routine returns a success indication.
3370                                ;*
3371                                ;* INPUTS:        R0 - Source address for move.
3372                                ;*               R1 - Destination address for move.
3373                                ;*               (R0) - Source for the move.
3374                                ;*
3375                                ;* OUTPUTS:       (R1) - Written to the contents of (R0).
3376                                ;*               Carry flag - Set on return if no 004 trap detected.
3377                                ;*               TP4FLG - Nonzero if trap occurred, cleared otherwise.
3378                                ;*
3379                                ;* CALLING SEQUENCE:    JSR     PC,CKTRAP
3380                                ;*
3381                                ;* COMMENTS:      If this subroutine causes a trap, either the address which
3382                                ;*               is labeled ADRPTR will be the trap PC address on the stack.
3383                                ;*
3384                                ;* SUBORDINATE ROUTINES CALLED: None.
3385                                ;*****************************************************************************
3386
3387 017154              CKTRAP:: SAVE                        ;SAVE CONTENTS OF GPRS R0 THRU R5.
     017154  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3388 017160  005037  002256              CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAGS.
3389 017164  011011                      MOV     (R0),(R1)       ;PERFORM THE MOVE IN QUESTION.
3390 017166  005737  002256  ADRPTR:: TST     TP4FLG          ;CHECK FOR OCCURENCE OF TRAP.
3391 017172  000261                      SEC                     ;INDICATE SUCCESS.
3392 017174  001401                      BEQ     60$             ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
3393 017176  000241                      CLC                     ;INDICATE FAILURE.
3394 017200              60$:    PASS                        ;RESTORE GPRS.
     017200  004736                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3395 017202  000207                  RTS     PC
```

```
3397                                    .SBTTL  GOBAL SUBROUTINE              - CKTRPB -
3398                                    ;*****************************************************************************
3399                                    ;*                          - CHECK FOR TRAP -
3400                                    ;*      THIS SUBROUTINE IS USED TO CHECK FOR A BUS TIME-OUT TRAP (004 TRAP)
3401                                    ;*      WHICH IS CAUSED BY AN ACCESS TO A NON-EXISTENT MEMORY OR I/O LOCATION
3402                                    ;*      IF A TRAP DOES NOT OCCUR. THIS ROUTINE RETURNS A SUCCESS INDICATION.
3403                                    ;*
3404                                    ;*      INPUTS:             R0 - SOURCE ADDRESS FOR MOVE
3405                                    ;*                          R1 - DESTINATION ADDRESS FOR MOVE
3406                                    ;*                          (R0) - SOURCE FOR THE MOVE
3407                                    ;*
3408                                    ;*      OUTPUTS:            (R1) - WRITTEN TO THE CONTENTS OF (R0)
3409                                    ;*                          CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED
3410                                    ;*                          TP4FLG - NONZERO IF TRAP OCCURED, CLEARED OTHERWISE.
3411                                    ;*
3412                                    ;*      CALLING SEQUENCE:   JSR    PC,CKTRPB
3413                                    ;*
3414                                    ;*      COMMENTS:           IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS
3415                                    ;*                          WHICH IS LABELED TRPAD2 WILL BE THE TRAP PC ADDRESS ON
3416                                    ;*                          THE STACK OR SOME OTHER ADDRESS WHICH WAS PLACED ON
3417                                    ;*                          THE STACK BY AN UNEXPECTED TRAP.
3418                                    ;*                                THIS ROUTINE PERFORMS A BYTE MOV .
3419                                    ;*
3420                                    ;*      SUBORDINATE ROUTINES CALLED:    NONE.
3421                                    ;*****************************************************************************
3422
3423 017204                            CKTRPB::           SAVE
     017204   004537   005326                            JSR     R5,PREG05           ;CALL REGISTER SAVE SUBRT.
3424
3425 017210   005037   002256                   CLR     TP4FLG              ;CLEAR THE 004 TRAP FLAGS
3426 017214   111011                            MOVB    (R0),(R1)           ;PERFORM THE BYTE MOVE
3427 017216   005737   002256            TRPAD2:: TST    TP4FLG              ;CHECK FOR OCCURENCE OF TRAP
3428 017222   000261                            SEC                         ;INDICATE SUCCESS
3429 017224   001401                            BEQ     60$                 ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR
3430 017226   000241                            CLC                         ;INDICATE FAILURE
3431 017230                            60$:       PASS                       ;
     017230   004736                            JSR     PC,@(SP)+           ;RETURN TO PREG05 SUBRT.
3432 017232   000207                            RTS     PC                  ;RETURN
```

```
3434                                    .SBTTL  GLOBAL SUBROUTINE              - CLNRST -
3435                                    ;************************************************************************
3436                                    ;*              - Clean Reset of the Device Under Test -
3437                                    ;*      This subroutine is used to reset the DUT to a known state.
3438                                    ;*      The DUT's self-test is skipped,and the fifo is purged of any error
3439                                    ;*      codes, etc.
3440                                    ;*      If the reset does not succesfully complete, then the carry bit is
3441                                    ;*      passed back to the calling routine (clear).
3442                                    ;*
3443                                    ;* INPUTS:        CSRA - Contains the address of the CSR
3444                                    ;*                TXBFCA - Contains address of DUT DMA Buffer Count register.
3445                                    ;*                ERRNBR - Error number for possible error report.
3446                                    ;*                ERRTBL- ERRTYP,ERNBR,and ERRMSG set up correctly.
3447                                    ;*
3448                                    ;* OUTPUTS:       The DUT performs its reset function into a known state.
3449                                    ;*                CARRY - Clear indicates the test is to be aborted.
3450                                    ;*                ERRBLK - value may be destroyed.
3451                                    ;*                IESTAT - TX and RX interrupt flags are cleared.
3452                                    ;*                TX and RX interrupt enable bits in the DUT's CSR are cleared.
3453                                    ;*
3454                                    ;* CALLING SEQUENCE:    JSR    PC,CLNRST
3455                                    ;*
3456                                    ;* COMMENTS:      This subroutine can report errors with numbers ERRNBR.
3457                                    ;*                This routine does not destroy the value of ERRNBR.
3458                                    ;*
3459                                    ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET,PUFIFO,RESETT.
3460                                    ;************************************************************************
3461
3462 017234                             CLNRST:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     017234  004537  005326                      JSR    R5,PREG05        ;CALL REGISTER SAVE SUBRT.
3463                                    ;+
3464                                    ; Reset the DUT.
3465                                    ;  This routine reports errors with numbers from ERRNBR thru ERRNBR+2.
3466                                    ;-
3467 017240  004737  024246                      JSR    PC,RESETT        ;RESET THE DUT TO A KNOWN STATE.
3468 017244  103002                              BCC    60$              ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
3469                                    ;+
3470                                    ; Purge the FIFO of error codes, save any BMP codes found.
3471                                    ;-
3472 017246  004737  023152                      JSR    PC,PUFIFO        ;PURGE THE FIFO.
3473
3474 017252                             60$:                              ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
3475 017252                                      PASS                     ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
     017252  004736                              JSR    PC,@(SP)+              ;RETURN TO PREG05 SUBRT.
3476                                                                       ;CARRY BIT:IF CLEAR, THEN ABORT THE TEST.
3477 017254  000207                              RTS    PC
```

```
3479                                    .SBTTL   GLOBAL SUBROUTINE                  - CLR16W -
3480                                    ;** ***********************************************************************
3481                                    ;*                         - Clear Sixteen Words Routine -
3482                                    ;*       This subroutine clears 16 words starting with the specified word.
3483                                    ;*
3484                                    ;* INPUTS:       R0 - Address of the first word to clear.
3485                                    ;*
3486                                    ;* OUTPUTS:      (R0) to (R0+15) - 16 words of memory are cleared to 0.
3487                                    ;*
3488                                    ;* CALLING SEQUENCE:    JSR      PC,CLR16W
3489                                    ;*
3490                                    ;* COMMENTS:
3491                                    ;*
3492                                    ;* SUBORDINATE ROUTINES CALLED: None.
3493                                    ;-- ***********************************************************************
3494
3495 017256                            CLR16W:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     017256  004537  005326                              JSR      R5,PREG05        ;CALL REGISTER SAVE SUBRT.
3496 017262  012701  000020                     MOV      #16.,R1         ;SET THE LOOP COUNTER TO 16.
3497 017266  005020                     2$:      CLR      (R0)+           ;CLEAR A WORD OF MEMORY.
3498 017270  005301                              DEC      R1              ;COUNT THIS LOOP.
3499 017272  001375                              BNE      2$              ;LOOP IF NOT 16 WORD CLEARED.
3500 017274                            60$:      PASS                          ;RESTORE GPRS.
     017274  004736                              JSR      PC,@(SP)+                ;RETURN TO PREG05 SUBRT.
3501 017276  000207                              RTS      PC
```

```
3503                                    .SBTTL  GLOBAL SUBROUTINE              - CONMAP -
3504                                    ;++ ***********************************************************************
3505                                    ;*                        - Convert Line bit map.
3506                                    ;*      This subrouitne is used to convert a bit map passed to it , into
3507                                    ;*      another line bit map that is based upon the associated TX/RX line
3508                                    ;*      number/offset table.
3509                                    ;*
3510                                    ;* INPUTS:      R5 - Contains the line bit map to be transformed.
3511                                    ;*              TXRXLB - Base address of associated TX/RX line number table.
3512                                    ;*
3513                                    ;* OUTPUTS:     R5 - Contains an assosciated line bit map.
3514                                    ;*
3515                                    ;* CALLING SEQUENCE:   JSR    PC,CONMAP
3516                                    ;*
3517                                    ;* COMMENTS:    The TX/RX association table must be initialised before this
3518                                    ;*              routine is called.
3519                                    ;*
3520                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
3521                                    ;-- ***********************************************************************
3522
3523 017300                  CONMAP::SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     017300  004537  005326                  JSR     R5,PREG05     ;CALL REGISTER SAVE SUBRT.
3524 017304  012702  005236          MOV     #TXRXLB,R2    ;GET THE BASE ADDRESS OF THE LINE ASSOC TABLE.
3525 017310  010503                  MOV     R5,R3         ;COPY THE BIT MAP TO BE TRANSFORMED.
3526 017312  012704  000010          MOV     #NUMLNS,R4    ;SET MAX LINE COUNTER.
3527 017316  005005                  CLR     R5            ;CLEAR ASSOCIATED LINE BIT MAP.
3528 017320  006203          2$:     ASR     R3            ;SHIFT ACTLNS BIT MAP INT BOOLEAN REGISTER.
3529 017322  103005                  BCC     4$            ;SKIP SETTING ASSOCIATED LINE NUMBER BIT MAP.
3530 017324  011201                  MOV     (R2),R1       ;GET ASSOCIATED LINE NUMBER OFFSET FROM TABLE.
3531 017326  006201                  ASR     R1            ;SHIFT RIGHT TO GET LINE NUMB FROM OFFSET.
3532 017330  004737  021276          JSR     PC,LINBIT     ;GENERATE AN SINGLE BIT MAP FOR THIS LINE.
3533 017334  050005                  BIS     R0,R5         ;SET BIT FOR THIS LINE IN ASSOCIATED BIT MAP.
3534 017336  005722          4$:     TST     (R2)+         ;INCREMENT ADDRESS FOR THE NEXT LINE NUMBER.
3535 017340  005304                  DEC     R4            ;DECREMENT LINE COUNT.
3536 017342  001366                  BNE     2$            ;LOOP IF NOT DONE.
3537 017344                  60$:    PASS    R5            ;RESTORE GPRS, EXCEPT
     017344  010566  000014                  MOV     R5,R5SLOT(SP)         ;PUT R5 IN STACK SLOT.
     017350  004736                          JSR     PC,@(SP)+             ;RETURN TO PREG05 SUBRT.
3538                                    ;R5 - CONTAINS THE ASSOCIATED LINE BIT MAP.
3539 017352  000207                  RTS     PC
```

```
3541                              .SBTTL  GLOBAL SUBROUTINE                - DELAY -
3542                              ;************************************************************************
3543                              ;*                          - DELAY SUBROUTINE -
3544                              ;*         This subroutine is used to delay a variable number of milli-seconds.
3545                              ;*
3546                              ;* INPUTS:        R4 - Contains the number of ms to delay.
3547                              ;*                 MSLCNT.
3548                              ;*
3549                              ;* OUTPUTS:       None.
3550                              ;*
3551                              ;* CALLING SEQUENCE:    JSR     PC,DELAY
3552                              ;*
3553                              ;* COMMENTS:      If no hardware clock interrupts are occuring, control-Cs will
3554                              ;*                not be honored for the duration of the delay.
3555                              ;*
3556                              ;* SUBORDINATE ROUTINES CALLED: None.
3557                              ;************************************************************************
3558
3559 017354              DELAY:: SAVE                        ;SAVE CONTENTS OF GPRS R0 THRU R5.
     017354 004537 005326                 JSR      R5,PREG05   ;CALL REGISTER SAVE SUBRT.
3560 017360 010401               MOV      R4,R1               ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
3561 017362 012702 177777        MOV      #-1,R2              ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
3562 017366 005003               CLR      R3                  ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
3563 017370 012704 017412        MOV      #62$,R4             ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
3564 017374 004737 021600        JSR      PC,MSLOOP           ;DELAY THE REQUESTED # OF MS.
3565 017400 103002               BCC      60$                 ;EXIT ROUTINE IF WE TIMED-OUT.]
3566 017402 004737 022364        JSR      PC,OOPS             ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
3567 017406              60$:     PASS                         ;RESTORE GPRS.
     017406 004736                 JSR      PC,@(SP)+          ;RETURN TO PREG05 SUBRT.
3568 017410 000207               RTS      PC
3569
3570 017412 177777      62$:     .WORD    -1                  ;DUMMY, NON-ZERO WORD.
```

```
3572                                      .SBTTL   GLOBAL SUBROUTINE        - DM16B -
3573                                 ;++  ****************************************************************
3574                                 ;*              - CONVERT TO A 16-BIT PHYSICAL ADDRESS -
3575                                 ;*      THIS ROUTINE CONVERTS FROM PAR FORM TO A 16-BIT PHYSICAL ADDRESS,
3576                                 ;*      OF ALTERNATE 1'S AND 0'S.
3577                                 ;*
3578                                 ;*      INPUTS:          DMTSTA: - CONTAINS THE ADDRESS IN PAR FORM
3579                                 ;*
3580                                 ;*      OUTPUTS:         R0 - CONTAINS THE 16 BIT PHYSICAL ADDRESS
3581                                 ;*
3582                                 ;*      CALLING SEQUENCE:     JSR     PC,DM16B
3583                                 ;*
3584                                 ;*      COMMENTS:    USED IN THE DMA ADDRESS TEST
3585                                 ;*
3586                                 ;*      SUBROUTINES CALLED:   NONE.
3587                                 ;--  ****************************************************************
3588
3589 017414                         DM16B:: SAVE
     017414  004537  005326                 JSR      R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3590 017420  013700  002264                 MOV      DMTSTA,R0       ;SHIFT THE DMA TEST ADDRESS
3591 017424  012702  000006                 MOV      #6,R2           ;SIX PLACES LEFT , TO
3592 017430  006300             2$:         ASL      R0              ;CONVERT IT INTO A .
3593 017432  005302                         DEC      R2              ;16-BIT PHYSICAL ADDRESS
3594 017434  001375                         BNE      2$              ;
3595
3596 017436  012701  000052                 MOV      #52,R1          ;SET UP THE 6 LSB'S
3597 017442  032700  000100                 BIT      #100,R0         ;IF BIT #6 OF THE PHYSICAL
3598 017446  001402                         BEQ      4$              ;ADDRESS IS CLEAR THEN BRANCH
3599 017450  012701  000025                 MOV      #25,R1          ;OTHERWISE CORRECT THE LSB'S
3600
3601 017454  060100             4$:         ADD      R1,R0           ;MREGE THE LSB'S WITH THE PHY ADDR
3602
3603 017456                                 PASS     R0              ;RETURN WITH THE PHY ADDR.
     017456  010066  000002                 MOV      R0,R0SLOT(SP)           ;PUT R0 IN STACK SLOT.
     017462  004736                         JSR      PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3604 017464  000207                         RTS      PC
3605
```

```
3607
3608                                    .SBTTL  GLOBAL SUBROUTINE              - DMRW -
3609                              ;++ *********************************************************************
3610                              ;*                   - READ/WRITE DATA FROM/TO (DMTSTA) -
3611                              ;*        THIS ROUTINE READS DATA BYTES FROM OR WRITES DATA BYTES TO AN ADDR OF
3612                              ;*        ALTERNATE 1'S AND 0'S . BITS 21 TO 6 OF THE ADDR ARE CONTAINED AT
3613                              ;*        DMTSTA. THE ROUTINE APPENDS THE 6 LSB'S TO PRODUCE AN ADDR OF ALTERNATE
3614                              ;*        1'S AND 0'S. THIS ROUTINE IS CALLED FROM THE DMA ADDRESS TEST.
3615                              ;*
3616                              ;* INPUTS:                      R0 - ADDRESS OF THE  DATA TO BE WRITTEN TO (DMTSTA),
3617                              ;*                                   IF A WRITE IS SPECIFIED.
3618                              ;*                              R1 - ADDRESS OF THE AREA IN WHICH DATA FROM (DMTSTA).
3619                              ;*                                   IS TO BE SAVED,IF A READ IS SPECIFIED.
3620                              ;*                              R3 - NUMBER OF DATA BYTES TO BE READ/WRITTEN
3621                              ;*                              R5 - CLEAR , SPECIFIES A READ FROM (DMTSTA)
3622                              ;*                                   SET , SPECIFIES A WRITE TO (DMTSTA).
3623                              ;*                              DMTSTA - CONTAINS BITS 21 TO 6 OF THE ADDR.
3624                              ;*                              MMSR0 - ADDRESS OF MEM MGT STATUS REG #0
3625                              ;*                              MMPRES - BIT #0 SET, INDICATES MEM MGT PRESENT
3626                              ;*                              PARA6 - ADDRESS OF MEM MGT PAR #6
3627                              ;*                              [† is replaced with PAR5] $$$
3628                              ;*                              TP4FLG - 004 TRAP FLAGS
3629                              ;*
3630                              ;* OUTPUTS:                     DATA AT (DMTSTA) SAVED OR WRITTEN
3631                              ;*                              PAR #6 - CONTENTS SET TO CONTENTS OF DMTSTA
3632                              ;*                              [† is replaced with PAR5] $$$
3633                              ;*                              TP4FLG - CLEAR IF READ/WRITE SUCCESSFUL
3634                              ;*                                       SET IF FAIL.
3635                              ;*
3636                              ;* CALLING SEQUENCE:                JSR     PC,DMRW
3637                              ;*
3638                              ;* COMENTS:                     IF MEM MGT IS PRESENT THE SUBROUTINE USES (DMTSTA)
3639                              ;*                              AS THE PAGE ADDRESS , PLACING IT IN PAR #5, AND CREATES
3640                              ;*                              A VIRTUAL ADDR IN THE RANGE OF PAR #5 WHICH CONTAINS
3641                              ;*                              THE SIX LSB'S.
3642                              ;*                              IF IT IS NOT PRESENT THE (DMTSTA) IS CONVERTED INTO
3643                              ;*                              THE EQUIVALENT 16 BIT  PHYSICAL ADDRESS.
3644                              ;*
3645                              ;* SUBORDINATE ROUTINES CALLED: CKTRAP,DM16B.
3646                              ;-- *********************************************************************
3647
3648 017466                      DMRW::  SAVE
     017466  004537  005326                              JSR     R5,PREG05         ;CALL REGISTER SAVE SUBRT.
3649
3650 017472  010004                              MOV     R0,R4             ;SAVE THE SOURCE ADDR
3651 017474  005737  002322                      TST     MMPRES            ;IF MEM MGT IS PRESENT THEN
3652 017500  001003                              BNE     6$                ;JUMP AND SET UP THE PAR #5 $$$
3653 017502  004737  017414                      JSR     PC,DM16B          ;OTHERWISE CONVERT DMTSTA INTO A 16-BIT
3654                                                                       ;PHYSICAL ADDRESS, IN R0.
3655 017506  000416                              BR      10$               ;JUMP TO PERFORM THE MOVE
3656 017510  013777  002264  162622  6$:         MOV     DMTSTA,@PAR5A     ;SET PAR #5 $$$
3657 017516  012700  120052                      MOV     #120052,R0        ;SET THE SIX LSB'S AND CONVERT TO
3658                                                                       ;A VIRTUAL ADDRESS WITHIN THE INFLUENCE
3659                                                                       ;OF PAR #5. $$$
3660 017522  032737  000001  002264              BIT     #1,DMTSTA         ;IF BIT #0 OF DMTSTA IS CLEAR THEN
3661 017530  001402                              BEQ     8$                ;AVOID CHANGING THE LSB'S
3662 017532  012700  120025                      MOV     #120025,R0        ;CHANGE THE LSB'S
```

GLOBAL SUBROUTINE   - DMRW -

```
3663 017536  012777  000001  162552  8$:   MOV   #BITO,@MMSRO      ;ENABLE MEM MGT.
3664 017544  005705                  10$:   TST   R5                ;IF A READ IS SPECIFIED THEN
3665 017546  001402                         BEQ   12$               ;AVOID SWAPING THE SOURCE AND DESTINATION.
3666 017550  010001                         MOV   RO,R1             ;SWAP
3667 017552  010400                         MOV   R4,RO             ;RESTORE THE ORIGINAL SOURCE FOR THE MOVE.
3668 017554  004737  017204          12$:   JSR   PC,CKTRPB         ;PERFORM THE BYTE MOVE.
3669 017560  103004                         BCC   14$               ;EXIT IF A TRAP OCCURED.
3670 017562  005201                         INC   R1                ;INCREMENT THE DESTINATION ADDRESS
3671 017564  005200                         INC   RO                ;INCREMENT THE SOURCE ADDR.
3672 017566  005303                         DEC   R3                ;DECREMENT THE DATA
3673 017570  001371                         BNF   12$               ;REPEAT UNTIL ALL DATA READ/WRITTEN
3674 017572  005737  002322          14$:   TST   MMPRES            ;IF MEM MGT IS PRESENT THEN
3675 017576  001402                         BEQ   16$               ;
3676 017600  005077  162512                 CLR   @MMSRO            ;DISABLE IT.
3677 017604                          16$:   PASS
     017604  004736                                JSR   PC,@(SP)+          ;RETURN TO PREGO5 SUBRT.
3678 017606  000207                         RTS   PC
3679
```

```
3681                                    .SBTTL  GLOBAL SUBROUTINE              - DODMA -
3682                                    ;++ ******************************************************************************
3683                                    ;*                      - Initiate DMA Transmission Routine -
3684                                    ;*          This routine writes the DMA parameter to the specified device and
3685                                    ;*          initiates the DMA transmission.
3686                                    ;*
3687                                    ;*  INPUTS:           R1 - Line number on which to initiate the DMA.
3688                                    ;*                    R2 - Start address of the DMA buffer (16 bit virtual).
3689                                    ;*                    R3 - Character count of the DMA buffer.
3690                                    ;*                    CSRA - Contains address of the DUT CSR.
3691                                    ;*                    IESTAT - Storage for states of the interrupt enable bits.
3692                                    ;*                    MMENAB - Memmory management flag (0 if MEM MGT not enabled).
3693                                    ;*                    HOST MEM MGT PAR REGISTERS - If MEM MGT is in use.
3694                                    ;*                    TXAD1A - Contains address of DMA TX buffer address reg #1.
3695                                    ;*                    TXAD2A - Contains address of DMA TX buffer address reg #2.
3696                                    ;*                    TXBFCA - Contains address of DMA character count register.
3697                                    ;*
3698                                    ;*  OUTPUTS:          CARRY - Success flag (set if DMA_START found clear).
3699                                    ;*                    DUT TBUFFAD1 - LS 16 bits of DMA buffer address (initialized).
3700                                    ;*                    DUT TBUFFAD2 - MS 6 bits of DMA buffer address (initialized),
3701                                    ;*                                   DMA_START bit set.
3702                                    ;*                    DUT TBUFFCT - DMA buffer character count (initialized).
3703                                    ;*
3704                                    ;*  CALLING SEQUENCE:     JSR     PC,DODMA
3705                                    ;*
3706                                    ;*  COMMENTS:         This routine determines if Memory Management is being used
3707                                    ;*                    and sets up the full 22 bit physical address if necessary.
3708                                    ;*
3709                                    ;*  SUBORDINATE ROUTINES CALLED: None.
3710                                    ;-- ******************************************************************************
3711
3712 017610                            DODMA:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
     017610  004537  005326                    JSR     R5,PREG05              ;CALL REGISTER SAVE SUBRT.
3713 017614  012704  000200                    MOV     #200,R4                ;PREPARE TO CLEAR UPPER 6 BITS OF DMA BUFF ADR.
3714 017620  005737  002324                    TST     MMENAB                 ;CHECK FOR MEMORY MANAGEMENT IN USE.
3715 017624  001427                            BEQ     6$                     ;GOTO SET UP DEVICE IF MEM MGT NOT IN USE.
3716                                    ;+
3717                                    ; Memmory management is in use.
3718                                    ; Construct 22 bit physical address from the 16 bit virtual address.
3719                                    ;-
3720 017626  010205                            MOV     R2,R5                  ;STRIP THE MOST SIGNIFICANT 3 BITS OF THE
3721 017630  012700  000005                    MOV     #5,R0                  ;  DMA BUFFER VIRTUAL ADDRESS AND MULTIPLY
3722 017634  006105                    2$:     ROL     R5                     ;  THEIR VALUE BY TWO TO GET AN OFFSET INTO
3723 017636  005300                            DEC     R0                     ;  THE TABLE OF MEMMORY MANAGEMENT PAGE
3724 017640  001375                            BNE     2$                     ;  ADDRESS REGISTERS (PAR).
3725 017642  042705  177761                    BIC     #177761,R5             ;
3726 017646  063705  002326                    ADD     PAR0A,R5               ;ADD IN THE BASE VALUE OF THE MM PAR REGISTERS.
3727 017652  011505                            MOV     (R5),R5                ;GET THE 16 BIT PHYSICAL ADDRESS BLOCK COUNT.
3728 017654  012700  000006                    MOV     #6,R0                  ;SHIFT UPPER 6 BITS OF THE PHYSICAL ADDRESS
3729 017660  006305                    4$:     ASL     R5                     ;  BLOCK COUNT (GOTTEN FROM THE PROPER PAR)
3730 017662  006104                            ROL     R4                     ;  INTO THE LS 6 BITS OF THE WORD TO WRITE
3731 017664  005300                            DEC     R0                     ;  INTO THE DUT TBUFFAD2 REGISTER.
3732 017666  001374                            BNE     4$                     ;
3733 017670  042702  160000                    BIC     #160000,R2             ;ADD THE 13 BIT DISPLACEMENT FIELD FROM VIRTUAL
3734 017674  060502                            ADD     R5,R2                  ;  ADR TO THE SHIFTED BLOCK NUMBER FROM THE
3735 017676  005504                            ADC     R4                     ;  MEMMORY MANAGEMENT PAR.
3736 017700  052704  000200                    BIS     #200,R4                ;SET THE DMA_START BIT IN WORD FOR TBUFFAD2.
```

GLOBAL SUBROUTINE   - DODMA -   SEQ 0099

```
3737                                       ;+
3738                                       ; Write the DMA parameters out to the DUT DMA registers.
3739                                       ;     Disable interrupts.
3740                                       ;     Set up DUT CSR IND.ADR.REG field.
3741                                       ;     Write the DMA transmit character count.
3742                                       ;     Write the least significant 16 bits of the DMA buffer start address.
3743                                       ;     Write the most significant 6 bits of the address,
3744                                       ;       setting the DMA_START bit, and initiating the DMA transmission.
3745                                       ;-
3746 017704  106705                6$:     MFPS    R5              ;GET THE PRESENT PROCESSOR PRIORITY.
3747 017706  106427  000340                MTPS    #PRIO7          ;DISABLE ALL HARDWARE INTERRUPTS.
3748 017712  053701  002234                BIS     IESTAT,R1       ;PREPARE FOR SETUP OF LINE NUMBER IN DUT CSR.
3749 017716  010177  162260                MOV     R1,@CSRA        ;SET UP THE DUT CSR IND.ADR.REG FIELD.
3750 017722  105777  162270                TSTB    @TXAD2A         ;TEST THE DUT DMA_START BIT.
3751 017726  000241                        CLC                     ;INDICATE FAILURE IN CASE DMA.HO BIT IS SET.
3752 017730  100410                        BMI     60$             ;EXIT WITH FAILURE IF DMA.HO BIT IS SET.
3753 017732  010377  162262                MOV     R3,@TXBFCA      ;WRITE THE DMA CHARACTER COUNT.
3754 017736  010277  162252                MOV     R2,@TXAD1A      ;WRITE THE LS 16 BITS OF BUFFER ADDRESS.
3755 017742  110477  162250                MOVB    R4,@TXAD2A      ;WRITE MS 6 BITS OF ADR AND START DMA TX.
3756 017746  106405                        MTPS    R5              ;RESTORE THE PROCESSOR PRIORITY.
3757 017750  000261                        SEC                     ;INDICATE SUCCESS.
3758
3759 017752                        60$:    PASS                    ;RESTORE GPRS,
     017752  004736                                JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3760 017754  000207                        RTS     PC              ; CARRY - SUCCESS FLAG (SET IF SUCCESS).
```

```
3762                              .SBTTL  GLOBAL SUBROUTINE                    - FINACT -
3763                              ;++ **************************************************************************
3764                              ;*                        - FIND FIRST ACTIVE LINE -
3765                              ;*      This subroutine calculates the number of the first active line that
3766                              ;*      is found in the active line bit map ACTLNS.
3767                              ;*
3768                              ;* INPUTS:        ACTLNS - Contains the active line bit map.
3769                              ;*
3770                              ;* OUTPUTS:       R1 - Contains the number of the first active line.
3771                              ;*               R5 - Contains the bit map representation of the active line.
3772                              ;*               Carry set indicates success.
3773                              ;*
3774                              ;* CALLING SEQUENCE:    JSR     PC,FINACT
3775                              ;*
3776                              ;* COMMENTS:
3777                              ;*
3778                              ;* SUBORDINATE ROUTINES CALLED: NONE.
3779                              ;-- **************************************************************************
3780
3781 017756                      FINACT:: SAVE                      ;SAVE CONTENTS OF GPRS R0 THRU R5.
     017756  004537  005326               JSR     R5,PREG05         ;CALL REGISTER SAVE SUBRT.
3782                                  ;+
3783                                  ; Find an active line on which to perform the test.
3784                                  ;-
3785 017762  005001                       CLR     R1                ;CLEAR THE LINE NUMBER COUNTER.
3786 017764  012703  000010               MOV     #NUMLNS,R3        ;GET MAX LINE NUMBER.
3787 017770  013700  002174               MOV     ACTLNS,R0         ;GET THE ACTIVE LINE BIT MAP.
3788 017774  012705  000001               MOV     #1,R5             ;SET UP A LINE BIT MASK.
3789 020000  030500              2$:      BIT     R5,R0             ;LOOK FOR AN ACTIVE LINE.
3790 020002  001006                       BNE     4$                ;BRANCH TO BEGIN TEST IF A LINE HAS BEEN FOUND.
3791 020004  006305                       ASL     R5                ;SHIFT THE BIT MASK FOR THE NEXT LINE.
3792 020006  005201                       INC     R1                ;INCREMENT THE LINE NUMBER COUNTER.
3793 020010  020103                       CMP     R1,R3             ;CHECK IF ALL LINES HAVE BEEN TRIED.
3794 020012  002772                       BLT     2$                ;LOOP TO TRY THE NEXT LINE.
3795 020014  000241                       CLC                       ;CLEAR CARRY BIT. NO ACTIVE LINE FOUND.
3796 020016  000401                       BR      60$               ;EXIT WITH FAILURE.
3797 020020  000261              4$:      SEC                       ;SET CARRY, SUCCESS.
3798
3799 020022                      60$:     PASS    R1,R5             ;RESTORE GPRS, EXCEPT
     020022  010166  000004               MOV     R1,R1SLOT(SP)     ;PUT R1 IN STACK SLOT.
     020026  010566  000014               MOV     R5,R5SLOT(SP)     ;PUT R5 IN STACK SLOT.
     020032  004736                        JSR     PC,@(SP)+         ;RETURN TO PREG05 SUBRT.
3800                                                                 ;R1 - CONTAINS THE NUMBER OF FIRST ACTIVE LINE.
3801                                                                 ;R5 - CONTAINS THE BIT MAP OF THE ACTIVE LINE.
3802                                                                 ;CARRY - SET INDICATES SUCCESS.
3803 020034  000207                       RTS     PC
```

```
3805                                    .SBTTL  GLOBAL SUBROUTINE                  - FRPSUP -
3806                                    ;++ *********************************************************************
3807                                    ;*                  - FRAMING AND PARITY ERROR TRANSMISSION/RECEPTION SET-UP -
3808                                    ;*
3809                                    ;*          This routine is used to initialise both the DUT and the
3810                                    ;*          transmission/reception control parameters to the correct
3811                                    ;*          state, prior to a framing or parity error detection and
3812                                    ;*          reporting test.
3813                                    ;*
3814                                    ;* INPUTS:        R0 - LPR contents for lines in the bit map in GPR4.
3815                                    ;*                R1 - LPR contents for lines not in the bit map in GPR4.
3816                                    ;*                R2 - Start address of data pattern to transmit.
3817                                    ;*                R3 - Length of the data pattern to TX.
3818                                    ;*                R4 - Local line group bit map.
3819                                    ;*                ACTLNS - Contains a bit map of all currently active lines.
3820                                    ;*                LOPBCK - Contains the type of loopback mode selected.
3821                                    ;*                CBB - Label at base of TX/RX control block.
3822                                    ;*
3823                                    ;* OUTPUTS:       The contents of the TXRCB are destroyed.
3824                                    ;*                The indirect address field of the DUT CSR may be destroyed.
3825                                    ;*                The DUT's LPR's and LNC's may be modified.
3826                                    ;*                The following pointers and counters are initialised;
3827                                    ;*                CHCNT,CHRTOT,DPEND,DPLEN,EXCNT,RXCNT,RXDONF,RXPTR,TXCNT,
3828                                    ;*                TXDONF,TXPTR,TXRXL.
3829                                    ;*
3830                                    ;* CALLING SEQUENCE:     JSR     PC,FRPSUP
3831                                    ;*
3832                                    ;* COMMENTS:      This routine should be called twice during the testing of
3833                                    ;*                the framing and parity error detection and reporting test.
3834                                    ;*                So that both line groups are tested on transmission and
3835                                    ;*                reception.
3836                                    ;*                JSR    PC,FRPSUP          ; do set-up.
3837                                    ;*                Execute test for the above set-up.
3838                                    ;*                Complement the line group bit map.
3839                                    ;*                JSR    PC,FRPSUP          ;do set up again.
3840                                    ;*                Execute test again.
3841
3842                                    ;* SUBORDINATE ROUTINES CALLED: TXRINI.
3843
3844                                    ;-- *********************************************************************
3845 020036                            FRPSUP:: SAVE                          ;SAVE THE CONTENTS OF THE GPR'S.
     020036        004537  005326                  JSR      R5,PREG05        ;CALL REGISTER SAVE SUBRT.
3846 020042        010037  020276              MOV      R0,70$               ;SAVE LPR PARAMETER FOR LINE TX.
3847 020046        010137  020300              MOV      R1,72$               ;SAVE LPR PARAMETER FOR LINE RX.
3848                                    ;+
3849                                    ; Set up the Transmission/Reception Control block to initialise the
3850                                    ; active lines in the bit map passed into this routine.
3851                                    ;-
3852 020052        010037  003124              MOV      R0,CBB               ;SET CONTENTS OF LPR PARAMS IN TX/RX C.BLK.
3853 020056        012700  003126              MOV      @CBB+2,R0            ;GET ADDRESS OF THE NEXT WORD IN THE CNTRL BLK.
3854 020062        012720  000004              MOV      #4,(R0)+             ;LNCTRL PARAMETER, ENABLE RECEIVERS.
3855 020066        010220                       MOV      R2,(R0)+             ;START ADDRESS OF DATA PATTERN.
3856 020070        010320                       MOV      R3,(R0)+             ;SET DATA PATTERN LENGTH.
3857 020072        012720  000001              MOV      #1,(R0)+             ;NUMBER OF DATA PATTERNS TO TRANSMIT.
3858 020076        013710  002174              MOV      ACTLNS,(R0)          ;BIT MAP OF LINES TO INITIALISE.
3859 020102        005104                       COM      R4                   ;GENERATE A BIT MAP OF ACTIVE LINES IN GRP1.
3860 020104        040420                       BIC      R4,(R0)+             ;CLEAR THE UNWANTED LINES.
```

```
3861 020106  113720  002176              MOVB    LOPBCK,(R0)+      ;SET LOOPBACK MODE,STAGGARED.
3862 020112  005200                       INC     R0               ;INCREMENT ADDRESS TO GET NEXT WORD IN TABLE.
3863 020114  012710  000001              MOV     #1,(R0)          ;SET AMMOUNT OF OFFSET FOR EACH TX START.
3864
3865                                  ;+
3866                                  ; Initialise the DUT and the associated pointers and counters, to the state
3867                                  ; dictated by the contents of the TX/RX control block.
3868                                  ;-
3869 020120  004737  026114              JSR     PC,TXRINI        ;INITIALISE DUT.
3870                                  ;+
3871                                  ; Set up Control block for lines in group 2.
3872                                  ;-
3873 020124  012700  003124              MOV     #CBB,R0          ;GET START ADDRESS OF CONTROL BLOCK.
3874 020130  010120                       MOV     R1,(R0)+         ;SET LPR PARAMETER FOR RX LINES.
3875 020132  062700  000010              ADD     #10,R0           ;SELECT THE ADDRESS OF THE LINE BIT MAP IN C.B.
3876 020136  013710  002174              MOV     ACTLNS,(R0)      ;BIT MAP OF LINES TO INITIALISE.
3877 020142  005104                       COM     R4               ;GENERATE A BIT MAP OF LINES IN GRP 2.
3878 020144  040410                       BIC     R4,(R0)          ;CLEAR THE UNWANTED LINES.
3879                                  ;+
3880                                  ; Initialise the DUT and the associated pointers and counters, to the state
3881                                  ; dictated by the contents of the TX/RX control block.
3882                                  ;-
3883 020146  004737  026114              JSR     PC,TXRINI        ;INITIALISE DUT.
3884
3885                                  ;+
3886                                  ; Set-up the required LPR parameters needed for the correct reception of data
3887                                  ; on associated in-active lines.
3888                                  ;-
3889                                  ;+
3890                                  ; Initialise LPR parameters for inactive lines in GROUP 2.
3891                                  ;-
3892 020152  012701  000377              MOV     #MAPLNS,R1       ;SET UP BIT MAP CORRESPONDING TO ALL LINES.
3893 020156  013702  002174              MOV     ACTLNS,R2        ;GET THE ACTIVE (TX) LINE BIT MAP.
3894 020162  005101                       COM     R1               ;GENERATE A BIT MAP OF NONE EXISTANT LINES.
3895 020164  005102                       COM     R2               ;GENERATE A BIT MAP OF INACTIVE LINES.
3896 020166  040102                       BIC     R1,R2            ;CLEAR ANY "NONE EXISTANT" INACTIVE LINES.
3897 020170  040402                       BIC     R4,R2            ;
3898 020172  010237  003136              MOV     R2,CBMAPA        ;SET UP BIT MAP IN CONTROL BLOCK.
3899 020176  005037  003134              CLR     CBDPNA           ;CLEAR REPEAT TX COUNT IN CONTROL BLOCK.
3900 020202  013737  020300  003124      MOV     72$,CBLPRA       ;SET-UP COMPLEMENTARY LPR PARM.
3901 020210  004737  026114              JSR     PC,TXRINI        ;INITIALISE INACTIVE LINES.
3902                                  ;+
3903                                  ; Initialise LPR parameters for inaactive lines in GROUP 1.
3904                                  ;-
3905 020214  013702  002174              MOV     ACTLNS,R2        ;GET THE ACTIVE (TX) LINE BIT MAP.
3906 020220  005102                       COM     R2               ;GENERATE A BIT MAP OF INACTIVE LINES.
3907 020222  040102                       BIC     R1,R2            ;CLEAR ANY NONE EXISTANT INACTIVE LINES.
3908 020224  005104                       COM     R4               ;
3909 020226  040402                       BIC     R4,R2            ;ONLY PASS LGRP2 ASSOCIATED LINE BIT MAP.
3910 020230  010237  003136              MOV     R2,CBMAPA        ;SET-UP BIT MAP IN CONTROL BLOCK.
3911 020234  013737  020276  003124      MOV     70$,CBLPRA       ;SET-UP COMPLAMENTARY LPR PARAM FOR LGRP1.
3912 020242  004737  026114              JSR     PC,TXRINI        ;INITIALISE INACTIVE LINES IN LGRP1.
3913                                  ;+
3914                                  ; Disable Receivers on all lines to ensure that only the receivers of the
3915                                  ; associated active (TX) lines are enabled.(staggared loopback)
3916                                  ;  Re-enable reception on the correct associated lines.
3917                                  ;-
```

GLOBAL SUBROUTINE   - FRPSUP -

```
3918 020246 012705 000377                      MOV     @MAPLNS,R5      ;SET-UP BIT MAP FOR ALL LINES.
3919 020252 004737 024460                      JSR     PC,RXDSBL       ;DISABLE RX ON ALL LINES.
3920                             ;+
3921                             ; Enable receivers on associated (RX) lines.
3922                             ;-
3923 020256 013705 002174                      MOV     ACTLNS,R5       ;GET ACTIVE (TX) LINE BIT MAP.
3924 020262 004737 017300                      JSR     PC,CONMAP       ;GENERATE AN ASSOCIATED (RX) LINE BIT MAP.
3925 020266 004737 024554                      JSR     PC,RXENBL       ;ENABLE RECEIVERS ON ASSOCIATED LINES.
3926
3927 020272                 60$:  PASS                                 ;RESTORE GRP'S.
     020272 004736                             JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3928 020274 000207                             RTS     PC
3929 020276 000000           70$:  .WORD   0                           ;LOCAL STORAGE OF LPR PARAMETER TX.
3930 020300 000000           72$:  .WORD   0                           ;LOCAL STORAGE OF LPR PARAMETER RX.
3931
```

```
3933                                   .SBTTL  GLOBAL SUBROUTINE              - GETBDR -
3934                                   ;++ ************************************************************************
3935                                   ;*                      - Get Baudrate Subroutine -
3936                                   ;*      This routine requests a baudrate input from the operator.  This
3937                                   ;*      baudrate is looked up in a table to give the LPR baudrate field
3938                                   ;*      value which is associated with that baudrate.
3939                                   ;*
3940                                   ;* INPUTS:        BDRMSG - Label at the baudrate prompt message.
3941                                   ;*                BRTBLE - Label after end of the baudrate table.
3942                                   ;*                UBRFMT - Label at the unsupported baudrate message.
3943                                   ;*
3944                                   ;* OUTPUTS:       R1 - Baudrate code in LS 4 bits.
3945                                   ;*
3946                                   ;* CALLING SEQUENCE:    JSR     PC,GETBDR
3947                                   ;*
3948                                   ;* COMMENTS:
3949                                   ;*
3950                                   ;* SUBORDINATE ROUTINES CALLED: None.
3951                                   ;-- ************************************************************************
3952
3953 020302              GETBDR:: SAVE                        ;SAVE CONTENTS OF GPRS R0 THRU R5.
     020302  004537  005326                    JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3954 020306  013705  002266            MOV     GMANWD,R5       ;SAVE THE GMAINIx VALUE.
3955                                   ;+
3956                                   ; Prompt the operator: "MODEM BAUDRATE IN BPS:  (D) 1200 ?"
3957                                   ;-
3958 020312  012737  002260  002266 2$:    MOV     #1200.,GMANWD           ;SET UP DEFAULT VALUE TO 1200 BAUD.
3959 020320                            GMANID  BDRMSG,GMANWD,D,177777,0,38400.,YES
     020320  104443                                                               TRAP    C$GMAN
     020322  000406                                                               BR      10000$
     020324  002266                                                               .WORD   GMANWD
     020326  000052                                                               .WORD   T$CODE
     020330  013127                                                               .WORD   BDRMSG
     020332  177777                                                               .WORD   177777
     020334  000000                                                               .WORD   T$LOLIM
     020336  113000                                                               .WORD   T$HILIM
     020340                                                                10000$:
3960 020340  013702  002266            MOV     GMANWD,R2
3961                                   ;+
3962                                   ; Attempt to look the value up in the baudrate table.
3963                                   ;-
3964 020344  012701  000017            MOV     #15.,R1         ;INITIALIZE BAUDRATE CODE TO HIGHEST BAUDRATE.
3965 020350  012703  002466            MOV     #BRTBLE,R3      ;INITIALIZE BAUDRATE POINTER.
3966
3967 020354  020243              4$:    CMP     R2,-(R3)        ;COMPARE BAUDRATE WITH A TABLE ENTRY.
3968 020356  001416                     BEQ     60$             ;BAUDRATES COMPARE?  YES, EXIT WITH CODE.
3969 020360  005301                     DEC     R1              ;NO, SET BAUDRATE CODE TO NEXT LOWER BAUDRATE.
3970 020362  001374                     BNE     4$              ;DONE?  NO, LOOP.
3971
3972 020364  020243                     CMP     R2,-(R3)        ;CHECK IF LAST BAUDRATE MATCHES.
3973 020366  001412                     BEQ     60$             ;BAUDRATES MATCH?  YES, EXIT WITH CODE.
3974
3975                                   ;+
3976                                   ; Report "nnnnn IS NOT A SUPPORTED BAUDRATE, ENTER ANOTHER OR CTRL C."
3977                                   ;-
3978 020370                            PRINTF  #UBRFMT,R2
     020370  010246                                                               MOV     R2,-(SP)
```

GLOBAL SUBROUTINE   - GETBDR -

```
        020372  012746  007532                                                              MOV     #UBRFMT,-(SP)
        020376  012746  000002                                                              MOV     #2,-(SP)
        020402  010600                                                                      MOV     SP,R0
        020404  104417                                                                      TRAP    C$PNTF
        020406  062706  000006                                                              ADD     #6,SP
3979    020412  000737                          BR      2$              ;LOOP TO GET ANOTHER BAUDRATE.
3980
3981    020414  010537  002266          60$:    MOV     R5,GMANWD       ;RESTORE THE GMANIx PARAMETER VALUE.
3982    020420                                  PASS    R1              ;RESTORE GPRS, EXCEPT THE FOLLOWING:
        020420  010166  000004                          MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
        020424  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3983    020426  000207                          RTS     PC              ;  R1 - BAUDRATE CODE.
```

```
3985                              .SBTTL  GLOBAL SUBROUTINE                  - GETCHR -
3986                              ;++ ********************************************************************************
3987                              ;*                  - Get a Character From the RX Buffer Routine -
3988                              ;*          This subroutine gets a character from the RX buffer which is in the
3989                              ;*          host system memory.  If the buffer is empty upon entry of this routine
3990                              ;*          this routine returns a null character with DATA.VALID clear and a
3991                              ;*          buffer empty indication.
3992                              ;*
3993                              ;*  INPUTS:        RXBCNT - RX buffer character count.
3994                              ;*                 RXBEND - Label after end of the RX buffer area in memory.
3995                              ;*                 RXBETX - Equated to RX buffer level at which to enable TX.
3996                              ;*                 RXBOPT - Pointer to next available input slot of RX buffer.
3997                              ;*                 RXBSTA - Label at start of RX buffer area in memory.
3998                              ;*
3999                              ;*  OUTPUTS:       R2 - Character which is read from the buffer.
4000                              ;*                 RXBOPT - Updated to point to next input slot of RX buffer.
4001                              ;*                 RXBCNT - RX buffer character count (Updated).
4002                              ;*                 CARRY - "Success" flag (Set if buffer is not empty on entry).
4003                              ;*
4004                              ;*  CALLING SEQUENCE:    JSR    PC,GETCHR
4005                              ;*
4006                              ;*  COMMENTS:
4007                              ;*
4008                              ;*  SUBORDINATE ROUTINES CALLED: None.
4009                              ;-- ********************************************************************************
4010
4011 020430               GETCHR:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     020430 004537 005326                  JSR    R5,PREG05      ;CALL REGISTER SAVE SUBRT.
4012 020434 005000                 CLR    R0                     ;CLEAR THE "RE-ENABLE" TX FLAG (SUBRTN OUTPUT).
4013 020436 005002                 CLR    R2                     ;GET NULL CHAR IN CASE BUFFER IS EMPTY.
4014 020440 005737 002720          TST    RXBCNT                 ;CHECK FOR RX BUFFER EMPTY, CLEAR CARRY.
4015 020444 001416                 BEQ    60$                    ;EXIT THE ROUTINE IF BUFFER IS EMPTY.
4016 020446 013704 002714          MOV    RXBOPT,R4              ;GET THE BUFFER OUTPUT POINTER.
4017 020452 011402                 MOV    (R4),R2                ;GET A CHARACTER FROM THE BUFFER.
4018 020454 005024                 CLR    (R4)+                  ;DELETE THE READ CHARACTER FROM THE BUFFER.
4019 020456 020427 003122          CMP    R4,@#RXBEND            ;CHECK IF POINTER SHOULD WRAP AROUND.
4020 020462 103402                 BLO    2$                     ;SKIP WRAPAROUND IF POINTER IS NOT AT END.
4021 020464 012704 002722          MOV    @#RXBSTA,R4            ;WRAP INPUT POINTER AROUND.
4022 020470 010437 002714   2$:    MOV    R4,RXBOPT              ;UPDATE THE OUTPUT POINTER STORAGE.
4023
4024 020474 005337 002720          DEC    RXBCNT                 ;REMOVE THIS CHAR FROM THE BUFFER COUNT.
4025 020500 000261                 SEC                           ;SET SUCCESS FLAG, BUFFER WAS NOT EMPTY.
4026
4027 020502               60$:     PASS   R2                     ;RESTORE GPRS, EXCEPT
     020502 010266 000006                 MOV    R2,R2SLOT(SP)              ;PUT R2 IN STACK SLOT.
     020506 004736                         JSR    PC,@(SP)+                 ;RETURN TO PREG05 SUBRT.
4028                                                             ;R2 - CONTAINS THE CHARACTER READ FROM BUFFER.
4029                                                             ;CARRY-"SUCCESS" FLAG, SET IF BUFFER NOT EMPTY.
4030 020510 000207                 RTS    PC
```

```
4032                             .SBTTL  GLOBAL SUBROUTINE                - GETLP1 -
4033                             ;++ **************************************************************************
4034                             ;*              - Get Line Parameters Routine Number One -
4035                             ;*       This routine is used to repeatedly get combinations of line parameter
4036                             ;*       contents for the Single Character Mode TX/RX Test (short data pattern).
4037                             ;*       Each time this routine is called it gets another combination of the
4038                             ;*       paramters in the paramter tables until all combinations have been
4039                             ;*       returned at which point it returns a "failure" indication.
4040                             ;*
4041                             ;* INPUTS:        Single character mode, short data pattern TX/RX tables:
4042                             ;*                      SCBCT - Number of bits per char table (4 entries).
4043                             ;*                      SCBRT - Baudrates table (3 entries).
4044                             ;*                      SCNST - Number of stop bits bits table (2 entries).
4045                             ;*                      SCTPT - Type of parity table (3 entries).
4046                             ;*               Each table has a base and end label consisting of the name of
4047                             ;*               the table with a "B" and "E" appended respectively.
4048                             ;*               R1 thru R4 - Pointers into SCBCT thru SCTPT tables respectivly.
4049                             ;*                     R1 is clear if this is the first call of GETLP1.
4050                             ;*
4051                             ;* OUTPUTS:       R0 - Composed LPR contents, clear if failure (Done).
4052                             ;*               R1 thru R4 - Table pointers (Updated).
4053                             ;*
4054                             ;* CALLING SEQUENCE:   JSR    PC,GETLP1
4055                             ;*
4056                             ;* COMMENTS:      This routine should be used in congunction with a SWAPx
4057                             ;*               routine to avoid destroying the GPR contents.
4058                             ;*
4059                             ;* SUBORDINATE ROUTINES CALLED: None.
4060                             ;-- **************************************************************************
4061
4062 020512           GETLP1:: SAVE                              ;SAVE CONTENTS OF GPRS R0 THRU R5.
     020512  004537  005326               JSR   R5,PREG05        ;CALL REGISTER SAVE SUBRT.
4063 020516  005701           TST   R1                ;TEST FOR THIS BEING FIRST CALL OF GETLP1.
4064 020520  001010           BNE   2$                ;SKIP ORIGINAL SET UP IF NOT FIRST CALL.
4065 020522  012701  005054   MOV   #SCBCTB,R1        ;INITIALIZE BITS PER CHAR TABLE POINTER.
4066 020526  012702  005054   MOV   #SCBRTB,R2        ;INITIALIZE BAUDRATE TABLE POINTER.
4067 020532  012703  005062   MOV   #SCNSTB,R3        ;INITIALIZE # OF STOP BITS TABLE POINTER.
4068 020536  012704  005066   MOV   #SCTPTB,R4        ;INITIALIZE TYPE OF PARITY TABLE POINTER.
4069
4070 020542  020427  005074   2$:     CMP   R4,#SCTPTE        ;CHECK FOR POINTER AT END OF TABLE.
4071 020546  103425           BLO   4$                ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
4072 020550  012704  005066   MOV   #SCTPTB,R4        ;RESET POINTER TO BEGINNING OF TABLE.
4073 020554  005723           TST   (R3)+             ;INC THE # OF STOP BITS TABLE POINTER BY 2.
4074 020556  020327  005066   CMP   R3,#SCNSTE        ;CHECK FOR POINTER AT END OF TABLE.
4075 020562  103417           BLO   4$                ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
4076 020564  012703  005062   MOV   #SCNSTB,R3        ;RESET POINTER TO BEGINNING OF TABLE.
4077 020570  005722           TST   (R2)+             ;INC BAUD RATES TABLE POINTER BY 2.
4078 020572  020227  005062   CMP   R2,#SCBRTE        ;CHECK FOR POINTER AT END OF TABLE.
4079 020576  103411           BLO   4$                ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
4080 020600  012702  005054   MOV   #SCBRTB,R2        ;RESET POINTER TO BEGINNING OF TABLE.
4081 020604  005721           TST   (R1)+             ;INC THE BITS PER CHAR TABLE POINTER BY 2.
4082 020606  020127  005054   CMP   R1,#SCBCTE        ;CHECK FOR POINTER AT END OF TABLE.
4083 020612  103403           BLO   4$                ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
4084 020614  005000           CLR   R0                ;PREPARE TO PASS OUT CLEAR LPR FIELDS.
4085 020616  000241           CLC                     ;INDICATE "FAILURE" FOR EXIT.
4086 020620  000405           BR    60$               ;EXIT WITH "FAILURE", WE'RE DONE.
4087
```

```
4088 020622  011100         4$:    MOV   (R1),R0             ;GET THE BITS/CHAR FIELD OF NEW LPR CONTENTS.
4089 020624  051200                BIS   (R2),R0             ;INCLUDE THE BAUD RATE FIELDS.
4090 020626  051300                BIS   (R3),R0             ;INCLUDE THE NUMBER OF STOP BITS FIELD.
4091 020630  052400                BIS   (R4)+,R0            ;INCLUDE THE TYPE OF PARITY FIELD.
4092
4093 020632  000261                SEC                       ;INDICATE "SUCCESS" FOR EXIT.
4094
4095 020634                 60$:   PASS  R0,R1,R2,R3,R4      ;RESTORE GPR R5, LEAVE THE FOLLOWING INTACT:
     020634  010066  000002               MOV  R0,R0SLOT(SP)          ;PUT R0 IN STACK SLOT.
     020640  010166  000004               MOV  R1,R1SLOT(SP)          ;PUT R1 IN STACK SLOT.
     020644  010266  000006               MOV  R2,R2SLOT(SP)          ;PUT R2 IN STACK SLOT.
     020650  010366  000010               MOV  R3,R3SLOT(SP)          ;PUT R3 IN STACK SLOT.
     020654  010466  000012               MOV  R4,R4SLOT(SP)          ;PUT R4 IN STACK SLOT.
     020660  004736                        JSR  PC,@(SP)+              ;RETURN TO PREG05 SUBRT.
4096 020662  000207                RTS   PC                  ;  R1 THRU R4 - POINTERS, R0 - NEW LPR FIELDS.
```

```
4098                              .SBTTL  GLOBAL SUBROUTINE              - GETLP2 -
4099                              ;++ ***************************************************************************
4100                              ;*              - Get Line Parameters Routine Number Two -
4101                              ;*      This routine is used to repeatedly get combinations of line parameter
4102                              ;*      contents for the Single Character Mode TX/RX Test (long data pattern).
4103                              ;*      Each time this routine is called it gets another combination of the
4104                              ;*      paramters in the paramter tables until all combinations have been
4105                              ;*      returned at which point it returns a "failure" indication.
4106                              ;*
4107                              ;* INPUTS:        Single character mode, short data pattern TX/RX tables:
4108                              ;*                    SCBCT - Number of bits per char table (4 entries).
4109                              ;*                    SCNST - Number of stop bits bits table (2 entries).
4110                              ;*                    SCTPT - Type of parity table (3 entries).
4111                              ;*             Each table has a base and end label consisting of the name of
4112                              ;*             the table with a "B" and "E" appended respectively.
4113                              ;*             R1 thru R3 - Pointers into SCBCT, SCNST, SCTPT tables
4114                              ;*                 R1 is clear if this is the first call of GETLP2.
4115                              ;*
4116                              ;* OUTPUTS:       R0 - Composed LPR contents, clear if failure (Done),
4117                              ;*                    38.4K baudrate is selected.
4118                              ;*                R1 thru R3 - Table pointers (Updated).
4119                              ;*
4120                              ;* CALLING SEQUENCE:    JSR    PC,GETLP2
4121                              ;*
4122                              ;* COMMENTS:      This routine should be used in congunction with a SWAPx
4123                              ;*                routine to avoid destroying the GPR contents.
4124                              ;*
4125                              ;* SUBORDINATE ROUTINES CALLED: None.
4126                              ;-- ***************************************************************************
4127
4128 020664                      GETLP2:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     020664   004537   005326                       JSR    R5,PREG05     ;CALL REGISTER SAVE SUBRT.
4129 020670   005701                       TST    R1                    ;TEST FOR THIS BEING FIRST CALL OF GETLP2.
4130 020672   001006                       BNE    2$                    ;SKIP ORIGINAL SET UP IF NOT FIRST CALL.
4131 020674   012701   005044               MOV    #SCBCTB,R1           ;INITIALIZE BITS PER CHAR TABLE POINTER.
4132 020700   012702   005062               MOV    #SCNSTB,R2           ;INITIALIZE # OF STOP BITS TABLE POINTER.
4133 020704   012703   005066               MOV    #SCTPTB,R3           ;INITIALIZE TYPE OF PARITY TABLE POINTER.
4134
4135 020710   020327   005074      2$:      CMP    R3,#SCTPTE           ;CHECK FOR POINTER AT END OF TABLE.
4136 020714   103417                        BLO    4$                   ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
4137 020716   012703   005066               MOV    #SCTPTB,R3           ;RESET POINTER TO BEGINNING OF TABLE.
4138 020722   005722                        TST    (R2)+                ;INC THE # OF STOP BITS TABLE POINTER BY 2.
4139 020724   020227   005066               CMP    R2,#SCNSTE           ;CHECK FOR POINTER AT END OF TABLE.
4140 020730   103411                        BLO    4$                   ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
4141 020732   012702   005062               MOV    #SCNSTB,R2           ;RESET POINTER TO BEGINNING OF TABLE.
4142 020736   005721                        TST    (R1)+                ;INC BAUD RATES TABLE POINTER BY 2.
4143 020740   020127   005054               CMP    R1,#SCBCTE           ;CHECK FOR POINTER AT END OF TABLE.
4144 020744   103403                        BLO    4$                   ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
4145 020746   005000                        CLR    R0                   ;PREPARE TO PASS OUT CLEAR LPR FIELDS.
4146 020750   000241                        CLC                         ;INDICATE "FAILURE" FOR EXIT.
4147 020752   000406                        BR     60$                  ;EXIT WITH "FAILURE", WE'RE DONE.
4148
4149 020754   012700   177400      4$:      MOV    #177400,R0           ;SET BAUD RATE FIELDS FOR 38.4 K BAUD.
4150 020760   051100                        BIS    (R1),R0              ;GET THE BITS/CHAR FIELD OF NEW LPR CONTENTS.
4151 020762   051200                        BIS    (R2),R0              ;INCLUDE THE NUMBER OF STOP BITS FIELD.
4152 020764   052300                        BIS    (R3)+,R0             ;INCLUDE THE TYPE OF PARITY FIELD.
4153
```

```
4154 020766  000261                         SEC                  ;INDICATE "SUCCESS" FOR EXIT.
4155
4156 020770                       60$:      PASS     R0,R1,R2,R3 ;RESTORE GPRS R4 & R5, LEAVE FOLLOWING INTACT:
     020770  010066  000002                 MOV      R0,R0SLOT(SP)          ;PUT R0 IN STACK SLOT.
     020774  010166  000004                 MOV      R1,R1SLOT(SP)          ;PUT R1 IN STACK SLOT.
     021000  010266  000006                 MOV      R2,R2SLOT(SP)          ;PUT R2 IN STACK SLOT.
     021004  010366  000010                 MOV      R3,R3SLOT(SP)          ;PUT R3 IN STACK SLOT.
     021010  004736                          JSR      PC,@(SP)+             ;RETURN TO PREG05 SUBRT.
4157 021012  000207                          RTS      PC         ;  R1 THRU R3 - POINTERS, R0 - NEW LPR FIELDS.
```

```
4159                                      .SBTTL  GLOBAL SUBROUTINE              - GETTIM -
4160                                      ;++ **********************************************************************
4161                                      ;*               - Get Time-out Value Based on Minimum Baudrate Routine -
4162                                      ;*        This subroutine gets the necessary time-out value to verify that all
4163                                      ;*        chars have been received at the completion of the TX/RX of a data
4164                                      ;*        pattern.  This uses the slowest baudrate which is specified in the
4165                                      ;*        passed in DUT LPR contents to calculate this time-out value.
4166                                      ;*
4167                                      ;* INPUTS:      R1 - DUT LPR contents.
4168                                      ;*
4169                                      ;* OUTPUTS:     RXTOUT - Time-out value for waiting for last RX char.
4170                                      ;*
4171                                      ;* CALLING SEQUENCE:    JSR    PC,GETTIM
4172                                      ;*
4173                                      ;* COMMENTS:
4174                                      ;*
4175                                      ;* SUBORDINATE ROUTINES CALLED: None.
4176                                      ;-- **********************************************************************
4177
4178 021014                              GETTIM:: SAVE                       ;SAVE CONTENTS OF GPRS RO THRU R5.
     021014  004537  005326                              JSR   R5,PREG05    ;CALL REGISTER SAVE SUBRT.
4179 021020  000301                          SWAB  R1               ;PUT THE BAUD RATE FIELDS IN THE LOW BYTE.
4180 021022  042701  177400                  BIC   #177400,R1       ;CLEAR STOP,PARITY,AND CHAR FIELDS.
4181 021026  010102                          MOV   R1,R2            ;COPY BAUD RATE FIELDS.
4182 021030  042701  000360                  BIC   #360,R1          ;SELECT RX BAUD RATE FIELD ONLY.
4183 021034  006202                          ASR   R2               ;SHIFT TX BAUD RATE FIELD
4184 021036  006202                          ASR   R2               ; TO OCCUPY THE LOW FOUR BYTES.
4185 021040  006202                          ASR   R2               ;
4186 021042  006202                          ASR   R2               ;
4187 021044  020102                          CMP   R1,R2            ;CHECK IF SAME BAUD RATE IN EACH FIELD.
4188 021046  101401                          BLOS  2$               ;BRANCH IF RX BAUD RATE IS LOWER OR SAME.
4189 021050  010201                          MOV   R2,R1            ;TX BAUD RATE IS THE SLOWER OF THE TWO.
4190 021052  116102  005216            2$:   MOVB  PROTBL(R1),R2    ;GET PROPORTIONAL DELAY FROM TABLE.
4191 021056  042702  177400                  BIC   #177400,R2       ;CLEAR UPPER BYTE BECAUSE OF SIGN EXTENSION.
4192 021062  010237  002242                  MOV   R2,RXTOUT        ;LOAD THE RX TIME-OUT VARIABLE.
4193
4194 021066                              60$:  PASS                     ;RESTORE GPRS.
     021066  004736                                  JSR   PC,@(SP)+        ;RETURN TO PREG05 SUBRT.
4195 021070  000207                          RTS   PC
```

```
4197                                 .SBTTL  GLOBAL SUBROUTINE                - INICHR -
4198                                 ;++ ********************************************************************************
4199                                 ;*                         - Send Initial Characters Routine -
4200                                 ;*         This routine is used to initiate single character transmission.
4201                                 ;*         This routine sends the initial characters to each active lines to
4202                                 ;*         cause future TX interrupts which will continue the transmission if
4203                                 ;*         more than one character is to be sent to each active line.
4204                                 ;*
4205                                 ;* INPUTS:        ACTLNS - Bit map of active DUT lines.
4206                                 ;*                BITTBL - Label of table of words each with a bit set.
4207                                 ;*                CSRA - Contains the address of the DUT CSR.
4208                                 ;*                DPENDB - Base of the data pattern end table (entry per line).
4209                                 ;*                DPLENB - Base of the data pattern length table.
4210                                 ;*                IBM - Bit mask of inactive TX/RX bits.
4211                                 ;*                IESTAT - States of DUT int enable bits (Other bits clear).
4212                                 ;*                NUMLNS - Equated to the number of lines on the DUT.
4213                                 ;*                TXCHRA - Contains the address of the DUT TXCHAR register.
4214                                 ;*                TXCNTB - Lable at base of the TX character counter table.
4215                                 ;*                TXPTRB - Label at base of the TX data pattern pointers table.
4216                                 ;*
4217                                 ;* OUTPUTS:       CSR - DUT CSR IND.ADR.REG field is destroyed.
4218                                 ;*                TXCHAR - DUT TXCHAR has word written to it.
4219                                 ;*                TXCNTx - Counters incremented for lines on which chars sent.
4220                                 ;*                TXPTRB - Each pointer in table points to next TX char for line.
4221                                 ;*
4222                                 ;* CALLING SEQUENCE:    JSR     PC,INICHR
4223                                 ;*
4224                                 ;* COMMENTS:      This routine assumes that at least one character should be
4225                                 ;*                transmitted on each active line.
4226                                 ;*                Interrupts must be disabled when calling this routine.
4227                                 ;*
4228                                 ;* SUBORDINATE ROUTINES CALLED: None.
4229                                 ;-- ********************************************************************************
4230 021072                         INICHR:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     021072  004537  005326                             JSR     R5,PREGO5  ;CALL REGISTER SAVE SUBRT.
4231 021076  013701  002234                 MOV     IESTAT,R1             ;GET STATE OF TX.IE, RX.IE FOR USE IN SETTING
4232                                                                       ;  UP THE IND.ADR.REG FIELD OF THE DUT CSR.
4233 021102  005002                         CLR     R2                    ;SET LINE NUMBER OFFSET TO LINE 0.
4234 021104  036237  002366  002174  2$:    BIT     BITTBL(R2),ACTLNS     ;TEST THE ACTIVE LINES BIT FOR THIS LINE.
4235 021112  001424                         BEQ     6$                    ;DON'T TX ON THIS LINE IF IT IS NOT ACTIVE.
4236 021114  010177  161062                 MOV     R1,@CSRA              ;SET UP THE IND.ADR.REG FIELD OF THE CSR.
4237 021120  016205  003344                 MOV     TXPTRB(R2),R5         ;GET THE TX DATA PATTERN POINTER FOR THIS LINE.
4238 021124  112504                          MOVB    (R5)+,R4             ;GET THE CHAR TO TX ON THIS LINE, INC POINTER.
4239 021126  020562  003144                 CMP     R5,DPENDB(R2)         ;COMPARE POINTER WITH DATA PATTERN END ADR.
4240 021132  103402                          BLO     4$                   ;SKIP POINTER WRAPAROUND IF NOT AT PATTERN END.
4241 021134  166205  003204                 SUB     DPLENB(R2),R5         ;WRAP TX POINTER AROUND TO BEGINNING OF PAT'N.
4242 021140  010562  003344          4$:    MOV     R5,TXPTRB(R2)         ;UPDATE THE TX POINTER STORAGE TABLE FOR LINE.
4243 021144  043704  002226                 BIC     IBM,R4                ;CLEAR INACTIVE BITS OF TX CHARACTER WORD.
4244 021150  052704  100000                 BIS     #BIT15,R4             ;SET THE TX.DATA.VALID BIT IN THE WORD.
4245 021154  010477  161024                 MOV     R4,@TXCHA             ;TX THE FIRST CHARACTER FOR THIS LINE.
4246 021160  005262  003504                 INC     TXCNTB(R2)            ;INCREMENT TX CHARACTER COUNTER FOR THIS LINE.
4247 021164  005201                  6$:    INC     R1                    ;INCREMENT WORD FOR IND.ADR.REG FIELD SET UP.
4248 021166  062702  000002                 ADD     #2,R2                 ;SET LINE NUMBER OFFSET TO NEXT LINE.
4249 021172  020227  000020                 CMP     R2,#NUMLNS*2          ;COMPARE LINE OFFSET WITH TWICE THE # OF LINES.
4250 021176  002742                          BLT     2$                   ;LOOP TO SEND CHAR TO ANOTHER LINE IF NOT DONE.
4251
4252 021200                          60$:   PASS                          ;RESTORE GPRS.
```

```
        021200  004736                              JSR     PC,@(SP)+              ;RETURN TO PREGO5 SUBRT.
4253    021202  000207              RTS     PC
```

```
4255                                 .SBTTL  GLOBAL SUBROUTINE              - INIDMA -
4256                                 ;++ *************************************************************************
4257                                 ;*                      - Initiate DMA Transmissions Routine -
4258                                 ;*        This routine is used to initiate DMA mode transmission.  It sends
4259                                 ;*        the initial DMA buffer on each active line to cause future TX
4260                                 ;*        interrupts which will continue the transmission if more than one
4261                                 ;*        buffer is to be sent.
4262                                 ;*
4263                                 ;*  INPUTS:         ACTLNS - Active lines bit map.
4264                                 ;*                  BITTBL - Label of table of words each with a bit set.
4265                                 ;*                  CSRA - Contains the address of the DUT CSR.
4266                                 ;*                  DPENDB - Base of the data pattern end table (entry per line).
4267                                 ;*                  DPLENB - Base of the data pattern length table.
4268                                 ;*                  IESTAT - Preserved states of the DUT interrupt enable bits.
4269                                 ;*                  NUMLNS - Equated to number of lines on a DUT.
4270                                 ;*                  TXCNTB - Lable at base of the TX character counter table.
4271                                 ;*                  TXPTRB - Label at base of the TX data pattern pointers table.
4272                                 ;*
4273                                 ;*  OUTPUTS:        CSR - DUT CSR IND.ADR.REG field is destroyed.
4274                                 ;*                  <<<<<<<<<<<<<<< ADD COMMENTS <<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<<
4275                                 ;*                  TXCNTx - Counters incremented for lines on which chars sent.
4276                                 ;*                  TXINTF - TX int flags (bit set if DMA.HO found set on line).
4277                                 ;*
4278                                 ;*  CALLING SEQUENCE:     JSR     PC,INIDMA
4279                                 ;*
4280                                 ;*  COMMENTS:       This routine assumes that at least one data pattern should be
4281                                 ;*                  transmitted on each active line.
4282                                 ;*                  Interrupts must be disabled when calling this routine.
4283                                 ;*
4284                                 ;*  SUBORDINATE ROUTINES CALLED: DODMA.
4285                                 ;-- *************************************************************************
4286 021204                         INIDMA:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     021204  004537  005326                  JSR       R5,PREG05           ;CALL REGISTER SAVE SUBRT.
4287 021210  013705  002174                  MOV       ACTLNS,R5           ;GET THE ACTIVE LINES BIT MAP.
4288                                 ;+
4289                                 ; Set up loop which handles one line per iteration.
4290                                 ;-
4291 021214  005001                          CLR       R1                  ;CLEAR THE LINE NUMBER COUNTER.
4292                                    ;+
4293                                    ; Get a bit map of the selected line.
4294                                    ; If the line is inactive skip to select the next line.
4295                                    ;-
4296 021216  010104                 2$:      MOV       R1,R4               ;CALCULATE AN OFFSET TO THE PROPER LINE
4297 021220  006304                          ASL       R4                  ;  ENTRY IN A WORD TABLE (LINE # TIMES 2).
4298 021222  016402  002366                  MOV       BITTBL(R4),R2       ;GET A BIT MAP FOR THIS LINE.
4299 021226  030205                          BIT       R2,R5               ;TEST THE ACTIVE LINES BIT FOR THIS LINE.
4300 021230  001414                          BEQ       10$                 ;DON'T TX ON THIS LINE IF IT IS NOT ACTIVE.
4301                                 ;+
4302                                 ; Line is active.
4303                                 ; Initiate DMA on this line.
4304                                 ;    Get the data pattern length for this line.
4305                                 ;-
4306 021232  016403  003204                  MOV       DPLENB(R4),R3       ;GET DATA PATTERN LENGTH FOR THIS LINE.
4307 021236  016402  003344                  MOV       TXPTRB(R4),R2       ;PREPARE TO PASS DATA PATTERN ADR TO DODMA RTN.
4308                                    ;+
4309                                    ; Write DMA parameters to the DUT.
4310                                    ;-
```

```
4311 021242 004737 017610                      JSR     PC,DODMA
4312 021246 103403                             BCS     6$                  ;SKIP ERROR IF DODMA WAS SUCCESSFUL.
4313                                    ;+
4314                                    ; Set the proper bit of the TX interrupt flags to indicate the line error.
4315                                    ;-
4316 021250 050537 002252                      BIS     R5,TXINTF           ;INDICATE THE ERROR.
4317 021254 000402                             BR      10$                 ;SKIP UPDATING POINTERS AND COUNTERS.
4318                                    ;+
4319                                    ; Update the TX character count for this line.
4320                                    ;-
4321 021256 060364 003504           6$:        ADD     R3,TXCNTB(R4)       ;ADD THE DATA PATTERN LENGTH TO TX CHAR COUNT.
4322                                    ;+
4323                                    ; Increment line counter,goto next line if not done.
4324                                    ;-
4325 021262 005201               10$:          INC     R1                  ;INCREMENT THE LINE COUNTER.
4326 021264 020127 000010                      CMP     R1,#NUMLNS          ;COMPARE THE LINE COUNTER WITH NUMBER OF LINES.
4327 021270 002752                             BLT     2$                  ;LOOP TO SEND CHAR TO ANOTHER LINE IF NOT DONE.
4328
4329 021272                       60$:         PASS                        ;RESTORE GPRS.
     021272 004736                                             JSR     PC,@(SP)+                    ;RETURN TO PREG05 SUBRT.
4330 021274 000207                             RTS     PC
```

```
4332                              .SBTTL  GLOBAL SUBROUTINE                - LINBIT -
4333                              ;++  **************************************************************************
4334                              ;*              - Line Number to Bit Map conversion subroutine -
4335                              ;*        This subroutine is used to generate a bit map (one bit of 16 set)
4336                              ;*        based on a line number (range: 1 to 16).  Only the LS 4 bits of the
4337                              ;*        line number word are used, the others are masked out (so unmasked
4338                              ;*        MSBytes of DUT CSRs can be passed to this routine without error).
4339                              ;*
4340                              ;*  INPUTS:        R1 - Line number (only LS 4 bits used, others disregarded).
4341                              ;*                 BITTBL - Base label of a 16 word bit table.
4342                              ;*
4343                              ;*  OUTPUTS:       R0 - Bit map, bit corresponding to line number is set:
4344                              ;*                      If line number is 3, then bit3 is set, etc.
4345                              ;*
4346                              ;*  CALLING SEQUENCE:    JSR    PC,LINBIT
4347                              ;*
4348                              ;*  COMMENTS:      No checking is performed to verify that the line number is
4349                              ;*                 a legal line number for the DUT (ie - less than NUMLNS).
4350                              ;*                 NOTE: The line number is not destroyed of altered, so this
4351                              ;*                       routine can be used easily in loops.
4352                              ;*
4353                              ;*  SUBORDINATE ROUTINES CALLED: None.
4354                              ;--  **************************************************************************
4355
4356 021276               LINBIT:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
     021276 004537 005326                  JSR    R5,PREG05        ;CALL REGISTER SAVE SUBRT.
4357 021302 042701 177760          BIC    #177760,R1      ;MASK OUT ALL BUT 4 LSBITS OF THE LINE #.
4358 021306 006301                 ASL    R1              ;MULTIPLY LINE # BY 2 TO GET WORD TABLE OFFSET.
4359 021310 016100 002366          MOV    BITTBL(R1),R0   ;GET THE SINGLE BIT BIT MAP.
4360 021314               60$:     PASS   R0              ;RESTORE GPRS, EXCEPT THE FOLLOWING,
     021314 010066 000002                 MOV    R0,ROSLOT(SP)    ;PUT R0 IN STACK SLOT.
     021320 004736                         JSR    PC,@(SP)+        ;RETURN TO PREG05 SUBRT.
4361 021322 000207          RTS    PC              ;R0 - BIT MAP WITH LINE # BIT SET.
```

```
4363                                .SBTTL  GLOBAL SUBROUTINE                - MODSUP -
4364                                ;++ ************************************************************************
4365                                ;*                      - MODEM LOOPBACK TX/RX SET-UP ROUTINE -
4366                                ;*
4367                                ;*              This routine is used to initialise both the DUT and the
4368                                ;*              transmission/reception control parameters to the correct
4369                                ;*              state, prior to a Modem Loopback test data pattern TX/RX.
4370                                ;*
4371                                ;* INPUTS:      R1 - TX, RX LPR contents.
4372                                ;*              R2 - Start address of data pattern to transmit.
4373                                ;*              R3 - Length of data pattern.
4374                                ;*              ACTLNS - Contains a bit map of all currently active lines.
4375                                ;*              CBB - Label at base of TX/RX control block.
4376                                ;*
4377                                ;* OUTPUTS:     The contents of the TX/RX control block (CCB) are destroyed.
4378                                ;*              The indirect address field of the DUT CSR may be destroyed.
4379                                ;*              The DUT's LPR's and LNC's may be modified.
4380                                ;*              The following pointers and counters are initialised;
4381                                ;*              CHCNT,CHRTOT,DPEND,DPLEN,EXCNT,RXCNT,RXPTR,TXCNT,
4382                                ;*              TXPTR,TXRXL.
4383                                ;*              CHRTOT, RXDONF, TXDONF and TXINTF are cleared.
4384                                ;*
4385                                ;* CALLING SEQUENCE:    JSR     PC,MODSUP
4386                                ;*
4387                                ;* COMMENTS:    DUT is set up with DSR and DTR set.  One data pattern is
4388                                ;*              sent and received from each line.
4389                                ;*
4390                                ;* SUBORDINATE ROUTINES CALLED: CONMAP,RXENBL,TXRINI.
4391                                ;-- ************************************************************************
4392
4393 021324                MODSUP:: SAVE                            ;SAVE CONTENTS OF THE GPR'S R0 THRU R5.
     021324  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
4394 021330  005037  002500         CLR     CHRTOT          ;CLEAR TOTAL RECEIVED CHAR COUNTER.
4395 021334  005037  002252         CLR     TXINTF          ;CLEAR FLAGS USED TO LOG DMA H.OVER ERRORS.
4396 021340  005037  002504         CLR     TXDONF          ;CLEAR THE TX DONE FLAGS.
4397 021344  005037  002506         CLR     RXDONF          ;CLEAR THE RX DONE FLAGS.
4398                                ;+
4399                                ; Set up the Transmission/Reception Control block to the desired state.
4400                                ;-
4401 021350  010137  003124         MOV     R1,CBB          ;SET CONTENTS OF LPR PARAMS IN TX/RX C.BLK.
4402 021354  012701  003124         MOV     @CBB,R1         ;GET BASE ADDRESS OF CONTROL BLOCK.
4403 021360  005201                 INC     R1              ;INCREMENT ADDRESS FOR NEXT WORD
4404 021362  005201                 INC     R1              ;INITIALISE THE FOLLOWING IN THE CNTRL.BLK:
4405 021364  012721  011004         MOV     #11004,(R1)+    ; LNCTRL: RTS, DTR, ENABLE RECEIVERS.
4406 021370  010221                 MOV     R2,(R1)+        ; START ADDRESS OF DATA PATTERN.
4407 021372  010321                 MOV     R3,(R1)+        ; DATA PATTERN LENGTH.
4408 021374  012721  000001         MOV     #1,(R1)+        ; NUMBER OF DATA PATTERNS TO TRANSMIT.
4409 021400  013721  002174         MOV     ACTLNS,(R1)+    ; BIT MAP OF LINES TO INITIALISE.
4410 021404  112721  000003         MOVB    #3,(R1)+        ;SET LOOPBACK MODE TO M325.
4411 021410  005201                 INC     R1              ;INCREMENT ADDRESS FOR THE NEXT WORD.
4412 021412  012711  000002         MOV     #2,(R1)         ;SET AMOUNT OF OFFSET EACH TX STARTS AT TO 2.
4413                                ;+
4414                                ; Initialise the DUT and the associated pointers and counters, to the state
4415                                ; dictated by the contents of the TX/RX control block.
4416                                ;-
4417 021416  004737  026114         JSR     PC,TXRINI       ;INITIALISE DUT.
4418                                ;+
```

```
4419                                      ; Initialise pointers and counters for inactive lines to zero.
4420                                      ;-
4421 021422  012701  000377              MOV      @MAPLNS,R1        ;GET THE LINE BIT MAP FOR ALL LINES.
4422 021426  013702  002174              MOV      ACTLNS,R2         ;GET THE ACTIVE LINE BIT MAP.
4423 021432  005101                      COM      R1                ;
4424 021434  005102                      COM      R2                ;
4425 021436  040102                      BIC      R1,R2             ;GENERATE AN IN-ACTIVE LINE BIT MAP.
4426 021440  010237  003136              MOV      R2,CBMAPA         ;MOVE BIT MAP TO THE CONTROL BLOCK.
4427 021444  005037  003126              CLR      CBLNCA            ;CLEAR THE LNCTRL SET UP PARAMETERS.
4428 021450  005037  003134              CLR      CBDPNA            ;CLEAR THE REPEAT TX COUNT IN CNTRL BLCK.
4429 021454  004737  026114              JSR      PC,TXRINI         ;SET UP PARAMETERS FOR INACTIVE LINES.
4430
4431 021460                      60$:    PASS                       ;RESTORE GPR'S.
     021460  004736                               JSR      PC,@(SP)+          ;RETURN TO PREGO5 SUBRT.
4432 021462  000207              RTS      PC
```

```
4434                                    .SBTTL  GLOBAL SUBROUTINE                    - MSLGET -
4435                                    ;********************************************************************************
4436                                    ;*         - Milli Seconds Loop which returns read word and remaining time -
4437                                    ;*         This subroutine is a general purpose test loop subroutine.  It is used
4438                                    ;*         to verify that a certain action occurs before a time-out period.  The
4439                                    ;*         calling routine passes in which bits should be set and cleared for the
4440                                    ;*         desired condition and the time-out value in milli-seconds.
4441                                    ;*         This routine checks for the desired condition upon entrance into the
4442                                    ;*         routine and then once each milli-second there after.
4443                                    ;*         Upon return, the last word which was read to check for the condition
4444                                    ;*         is returned by this subroutine.
4445
4446                                    ;*  INPUTS:        R1 - Time-out value in milli-seconds (up to 64K ms).
4447                                    ;*                 R2 - Bit map of bits to test (1 indicates to test the bit).
4448                                    ;*                 R3 - Desired states of the indicated fields in R2.
4449                                    ;*                 R4 - Address of the word to test.
4450                                    ;*                 MSLCNT - Milli second software loop count.
4451                                    ;*
4452                                    ;*  OUTPUTS:       R0 - The last word which was read to check for the condition.
4453                                    ;*                 R1 - Remaining number of ms in time-out time.
4454                                    ;*                 CARRY - Success flag (set if condition is met before time-out).
4455
4456                                    ;*  CALLING SEQUENCE:    JSR    PC,MSLGET
4457                                    ;*
4458                                    ;*  COMMENTS:      This routine works with or without a hardware clock, but the
4459                                    ;*                 calibration is only guarenteed when a line clock is available
4460                                    ;*                 on the system.
4461                                    ;*                 This routine can be used as a delay routine, by specifying the
4462                                    ;*                 desired delay as the time-out and specifying a condition to
4463                                    ;*                 look for which will not be met during the delay.
4464                                    ;*                 If a time-out value of 0 is specified, this routine checks for
4465                                    ;*                 the desired condition before returning.  It indicates success
4466                                    ;*                 if the condition is met, failure otherwise.
4467                                    ;*
4468                                    ;*
4469                                    ;*  SUBORDINATE ROUTINES CALLED: None.
4470                                    ;********************************************************************************
4471
4472 021464                            MSLGET:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     021464  004537  005326                     JSR     R5,PREG05          ;CALL REGISTER SAVE SUBRT.
4473                                    ;+
4474                                    ; Set up mask for removing unused bits in the test word, and clear unused
4475                                    ; bits in the desired state word to allow direct comparison.
4476                                    ;-
4477 021470  005102                             COM     R2                 ;GET MASK OF UNUSED BITS.
4478 021472  040203                             BIC     R2,R3              ;MASK OUT UNUSED BITS IN DESIRED STATE WORD.
4479                                    ;+
4480                                    ; Handle the test and exit if we have a 0 time-out value.
4481                                    ;-
4482 021474  005701                             TST     R1                 ;TEST THE TIME-OUT VALUE FOR ZERO.
4483 021476  001011                             BNE     2$                 ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
4484 021500  011400                             MOV     (R4),R0            ;GET THE WORD TO TEST BEFORE EXITING.
4485 021502  010037  021576                     MOV     R0,62$             ;SAVE VALUE SO WE CAN RETURN IT.
4486 021506  040200                             BIC     R2,R0              ;MASK OUT UNTESTED BITS OF WORD.
4487 021510  020003                             CMP     R0,R3              ;COMPARE AGAINST DESIRED STATE WORD.
4488 021512  000261                             SEC                        ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
4489 021514  001420                             BEQ     6$                 ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
```

```
4490 021516  000241                        CLC                      ;INDICATE FAILURE (TIME-OUT).
4491 021520  000416                        BR      6$               ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
4492                              ;+
4493                              ; Non-zero time-out value.  Loop, waiting for condition or time-out.
4494                              ;-
4495 021522  013705  002314      2$:       MOV     MSLCNT,R5        ;LOAD MS LOOP COUNT.
4496 021526  011400              4$:       MOV     (R4),R0          ;GET THE WORD TO TEST.
4497 021530  010037  021576                MOV     R0,62$           ;SAVE WORD IN CASE THIS IS THE LAST.
4498 021534  040200                        BIC     R2,R0            ;MASK OUT UNTESTED BITS OF WORD.
4499 021536  020003                        CMP     R0,R3            ;COMPARE AGAINST DESIRED STATE WORD.
4500 021540  000261                        SEC                      ;SET CARRY IN CASE OF SUCCESS.
4501 021542  001405                        BEQ     6$               ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
4502 021544  005305                        DEC     R5               ;COUNT DOWN THE INSIDE MS LOOP COUNT.
4503 021546  001367                        BNE     4$               ;LOOP IF MS NOT UP.
4504 021550  005301                        DEC     R1               ;DECREMENT THE MS TIME COUNT.
4505 021552  001363                        BNE     2$               ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
4506 021554  000241                        CLC                      ;CLEAR CARRY, WE TIMED-OUT.
4507                              ;+
4508                              ; Have either found condition, or timed-out (possibly from 0 time-out value).
4509                              ; Restore the last contents read from the test word.  Exit routine.
4510                              ;-
4511 021556  013700  021576      6$:       MOV     62$,R0           ;PASS OUT THE LAST READ WORD.
4512 021562                      60$:      PASS    R0,R1            ;RESTORE GPRS, EXCEPT THE FOLLOWING:
     021562  010066  000002                MOV     R0,R0SLOT(SP)            ;PUT R0 IN STACK SLOT.
     021566  010166  000004                MOV     R1,R1SLOT(SP)            ;PUT R1 IN STACK SLOT.
     021572  004736                        JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
4513                                                                 ;R0 - LAST READ WORD CHECKED FOR CONDITION.
4514                                                                 ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
4515 021574  000207                        RTS     PC               ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
4516                              ;+
4517                              ; Local storage.
4518                              ;-
4519 021576  000000             62$:       .WORD   0                ;STORAGE FOR THE LAST READ WORD.
```

```
4521                                    .SBTTL  GLOBAL SUBROUTINE              - MSLOOP -
4522                                    ;********************************************************************
4523                                    ;*         - Test Loop subroutine -
4524                                    ;*         This subroutine is a general purpose test loop subroutine.  It is used
4525                                    ;*         to verify that a certain action occurs before a time-out period.  The
4526                                    ;*         calling routine passes in which bits should be set and cleared for the
4527                                    ;*         desired condition and the time-out value in milli-seconds.
4528                                    ;*         This routine checks for the desired condition upon entrance into the
4529                                    ;*         routine and then once each milli-second thereafter.
4530                                    ;*
4531                                    ;* INPUTS:       R1 - Time-out value in milli-seconds (up to 64K ms).
4532                                    ;*               R2 - Bit map of bits to test (1 indicates to test the bit).
4533                                    ;*               R3 - Desired states of the indicated fields in R2.
4534                                    ;*               R4 - Address of the word to test.
4535                                    ;*               MSLCNT - Milli second software loop count.
4536                                    ;*
4537                                    ;* OUTPUTS:      CARRY - Success flag (set if condition is met before time-out).
4538                                    ;*
4539                                    ;* CALLING SEQUENCE:    JSR    PC,MSLOOP
4540                                    ;*
4541                                    ;* COMMENTS:     This routine works with or without a hardware clock, but the
4542                                    ;*               calibration is only guarenteed when a line clock is available
4543                                    ;*               on the system.
4544                                    ;*               This routine can be used as a delay routine, by specifying the
4545                                    ;*               desired delay as the time-out and specifying a condition to
4546                                    ;*               look for which will not be met during the delay.
4547                                    ;*               If a time-out value of 0 is specified, this routine checks for
4548                                    ;*               the desired condition before returning.  It indicates success
4549                                    ;*               if the condition is met, failure otherwise.
4550                                    ;*
4551                                    ;* SUBORDINATE ROUTINES CALLED: MSLGET.
4552                                    ;********************************************************************
4553
4554 021600                    MSLOOP:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     021600  004537  005326                            JSR    R5,PREG05        ;CALL REGISTER SAVE SUBRT.
4555
4556                                    ;+
4557                                    ; Calling the MSLGET routine from the MSLOOP routine isolates the caller of
4558                                    ; MSLOOP from the returned test word and remaining time-out values.
4559                                    ;-
4560 021604  004737  021464                    JSR    PC,MSLGET        ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.
4561
4562 021610                    60$:    PASS                          ;RESTORE GPRS,
     021610  004736                            JSR    PC,@(SP)+             ;RETURN TO PREG05 SUBRT.
4563 021612  000207                    RTS    PC                     ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
```

```
4565                                .SBTTL  GLOBAL SUBROUTINE                    - MSSRPT -
4566                                ;++ *******************************************************************************
4567                                ;*                       - Modem Status Signal Report Routine -
4568                                ;*      This subroutine is used to report the states of the modem status
4569                                ;*      signals for all active lines.
4570                                ;*
4571                                ;* INPUTS:       ACTLNS - Bit map of active lines.
4572                                ;*               CSRA - Contains address of the DUT CSR.
4573                                ;*               EF9101 - Label at format statement for blank line.
4574                                ;*               IESTAT - Contains states of the DUT Interrupt Enable bits.
4575                                ;*               STATA - Contains address of the DUT STAT register.
4576                                ;*               NUMLNS - Equated to the number of lines on the device.
4577                                ;*
4578                                ;* OUTPUTS:      DUT CSR IND.ADR.REG Field - Contents destroyed.
4579                                ;*               Report messages are printed on the operator's console.
4580                                ;*
4581                                ;* CALLING SEQUENCE:    JSR     PC,MSSRPT
4582                                ;*
4583                                ;* COMMENTS:
4584                                ;*
4585                                ;* SUBORDINATE ROUTINES CALLED: None.
4586                                ;-- *******************************************************************************
4587
4588 021614                MSSRPT:: SAVE                              ;SAVE CONTENTS OF GPRS R0 THRU R5.
     021614  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
4589                                ;+
4590                                ; Print the basic modem status message.
4591                                ;    "MODEM STATUS SIGNAL REPORT:"
4592                                ;-
4593 021620                        PRINTF  #MSFMT1
     021620  012746  007630                                                          MOV     #MSFMT1,-(SP)
     021624  012746  000001                                                          MOV     #1,-(SP)
     021630  010600                                                                  MOV     SP,R0
     021632  104417                                                                  TRAP    C$PNTF
     021634  062706  000004                                                          ADD     #4,SP
4594
4595 021640  005001                  CLR     R1              ;START WITH LINE 0.
4596 021642  012702  000001          MOV     #1,R2
4597 021646  013703  002202          MOV     CSRA,R3         ;GET THE CSR ADDRESS.
4598 021652  013704  002234          MOV     IESTAT,R4       ;GET THE STATES OF THE INTERRUPT ENABLE BITS.
4599 021656  013705  002174          MOV     ACTLNS,R5       ;GET THE ACTIVE LINES BIT MAP.
4600
4601 021662  030205          2$:     BIT     R2,R5           ;TEST LINE BIT IN ACTIVE LINES BIT MAP.
4602 021664  001442                  BEQ     4$              ;LINE ACTIVE?  NO, SKIP REPORT FOR LINE.
4603
4604 021666  010400                  MOV     R4,R0           ;SET UP DUT CSR IND.ADR.REG FIELD
4605 021670  050100                  BIS     R1,R0           ;  LEAVING THE INTERRUPT ENABLE
4606 021672  010013                  MOV     R0,(R3)         ;  BITS IN THE SPECIFIED STATE.
4607 021674  017700  160310          MOV     @STATA,R0       ;READ THE DUT STATUS REG FOR THIS LINE.
4608 021700                  SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
     021700  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
4609 021704  005002                  CLR     R2              ;CLEAR THE SIGNAL STATUS INDICATORS.
4610 021706  005003                  CLR     R3
4611 021710  005004                  CLR     R4
4612 021712  005005                  CLR     R5
4613 021714  006300                  ASL     R0              ;SHIFT DSR INTO CARRY,
4614 021716  006102                  ROL     R2              ;  THEN ROTATE INTO INDICATOR.
```

GLOBAL SUBROUTINE   - MSSRPT -   SEQ 0123

```
4615 021720  006300                              ASL    RO           ;SHIFT BLANK SLOT INTO CARRY,
4616 021722  006300                              ASL    RO           ;  SHIFT RI INTO CARRY,
4617 021724  006103                              ROL    R3           ;   THEN ROTATE INTO INDICATOR.
4618 021726  006300                              ASL    RO           ;SHIFT DCD INTO CARRY,
4619 021730  006104                              ROL    R4           ;   THEN ROTATE INTO INDICATOR.
4620 021732  006300                              ASL    RO           ;SHIFT CTS INTO CARRY,
4621 021734  006105                              ROL    R5           ;   THEN ROTATE INTO INDICATOR.
4622                               ;+
4623                               ; Print the status for this line.
4624                               ;    "LINE #n:  DSR=n, RI=n, DCD=n, CTS=n"
4625                               ;-
4626 021736                              PRINTF  #MSFMT2,R1,R2,R3,R4,R5
     021736  010546                                                                            MOV    R5,-(SP)
     021740  010446                                                                            MOV    R4,-(SP)
     021742  010346                                                                            MOV    R3,-(SP)
     021744  010246                                                                            MOV    R2,-(SP)
     021746  010146                                                                            MOV    R1,-(SP)
     021750  012746  007670                                                                    MOV    #MSFMT2,-(SP)
     021754  012746  000006                                                                    MOV    #6,-(SP)
     021760  010600                                                                            MOV    SP,RO
     021762  104417                                                                            TRAP   C$PNTF
     021764  062706  000016                                                                    ADD    #16,SP
4627 021770                              PASS                        ;RESTORE ALL THE GPRS.
     021770  004736                                      JSR    PC,@(SP)+         ;RETURN TO PREGO5 SUBRT.
4628
4629 021772  006302               4$:    ASL    R2                   ;SHIFT LINE BIT MAP TO NEXT LINE.
4630 021774  005201                      INC    R1                   ;INCREMENT THE LINE COUNTER.
4631 021776  020127  000010              CMP    R1,#NUMLNS           ;CMP LINE COUNTER WITH # OF LINES ON DEVICE.
4632 022002  002727                      BLT    2$                   ;ALL LINES DONE?  NO, LOOP TO DO NEXT LINE.
4633
4634 022004                              PRINTF #EF9101              ;PRINT A BLANK LINE.
     022004  012746  007313                                                                    MOV    #EF9101,-(SP)
     022010  012746  000001                                                                    MOV    #1,-(SF)
     022014  010600                                                                            MOV    SP,RO
     022016  104417                                                                            TRAP   C$PNTF
     022020  062706  000004                                                                    ADD    #4,SP
4635
4636 022024               60$:    PASS                        ;RESTORE GPRS.
     022024  004736                                      JSR    PC,@(SP)+                 ;RETURN TO PREGO5 SUBRT.
4637 022026  000207                      RTS    PC
```

```
4639                                    .SBTTL  GLOBAL SUBROUTINE                   - MUL16U -
4640                                    ;++ ********************************************************************
4641                                    ;*                     - 16 Bit Unsigned Multiply Routine -
4642                                    ;*      This routine multiplies 2 16 bit unsigned numbers and returns a 16 bit
4643                                    ;*      unsigned result.  The multiplication is performed by iterative
4644                                    ;*      addition of one number to a sum while decrementing the other number
4645                                    ;*      to zero.  If overflow occurrs (177777 to 0) the product is invalid.
4646                                    ;*
4647                                    ;* INPUTS:      R1 - Multiplicand (16 bit unsigned).
4648                                    ;*              R2 - Multiplier (16 bit unsigned).
4649                                    ;*
4650                                    ;* OUTPUTS:     R1 - Product (16 bit unsigned), -1 if overflow.
4651                                    ;*              CARRY - Set if success (no overflow), clear otherwise.
4652                                    ;*
4653                                    ;* CALLING SEQUENCE:    JSR     PC,MUL16U
4654                                    ;*
4655                                    ;* COMMENTS:    Note: For minimum execution time R2 should contain the
4656                                    ;*                    smaller of the 2 arguments.
4657                                    ;*
4658                                    ;* SUBORDINATE ROUTINES CALLED: None.
4659                                    ;-- ********************************************************************
4660
4661 022030                            MUL16U:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     022030  004537  005326                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
4662 022034  005003                            CLR     R3              ;CLEAR THE PRODUCT.
4663 022036  005702                            TST     R2              ;CHECK THE MULTIPLIER.
4664 022040  001003                            BNE     2$              ;GO TO DO MULTIPLICATION IF NOT ZERO.
4665 022042  005001                            CLR     R1              ;RETURN A PRODUCT OF ZERO.
4666 022044  000261                            SEC                     ;INDICATE SUCCESS.
4667 022046  000412                            BR      60$             ;EXIT THE ROUTINE.
4668
4669 022050  060103                  2$:       ADD     R1,R3           ;ADD THE MULTIPLICAND TO THE PRODUCT.
4670 022052  103405                            BCS     50$             ;EXIT WITH OVERFLOW IF ONE OCCURRED.
4671 022054  005302                            DEC     R2              ;DECREMENT THE MULTIPLIER.
4672 022056  001374                            BNE     2$              ;LOOP IF MULTIPLIER NOT ZERO.
4673 022060  010301                            MOV     R3,R1           ;PREPARE TO PASS OUT THE PRODUCT.
4674 022062  000261                            SEC                     ;INDICATE SUCCESS.
4675 022064  000403                            BR      60$             ;EXIT WITH SUCCESS.
4676
4677 022066  012701  177777          50$:      MOV     #-1,R1          ;FORCE PRODUCT TO MAX VALUE, WE OVERLFOWED.
4678 022072  000241                            CLC                     ;INDICATE FAILURE.
4679
4680 022074                          60$:      PASS    R1              ;RESTORE GPRS, EXCEPT THE FOLLOWING:
     022074  010166  000004                          MOV     R1,R1SLOT(SP)   ;PUT R1 IN STACK SLOT.
     022100  004736                          JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
4681                                                                  ;  R1 - PRODUCT (16 BIT UNSIGNED),
4682 022102  000207                            RTS     PC              ;  CARRY - SET IF SUCCESS (NO OVERFLOW).
```

```
4684                                           .SBTTL  GLOBAL SUBROUTINE                    - NEWCHR -
4685                                           ;++ ********************************************************************
4686                                           ;*                      - New Character Handling Routine -
4687                                           ;*        This subroutine handles a new character which has been read from
4688                                           ;*        the DUT.  The counters and pointers which are involved with the
4689                                           ;*        character are updated.  The character is checked for errors and
4690                                           ;*        any errors which are found are reported.
4691                                           ;*
4692                                           ;* INPUTS:      R2 - The read character including error flags and line number.
4693                                           ;*              R3 - Mask of the inactives bits in a TX or RX char byte.
4694                                           ;*              ACTLNS  - Bit map of active DUT lines.
4695                                           ;*              DPRSQB - Label at data pattern resync queues table base.
4696                                           ;*              TXRXLB - Base of TX/RX line number association table.
4697                                           ;*              BITTBL - Table of words with bits set for use in forming maps.
4698                                           ;*              ERSMRF - "Print error summary for line" flags.
4699                                           ;*              ERRTBL - Error information (ERRNBR, ERRMSG, ERRTYP).
4700                                           ;*              ERCNTB - Base of the RX character error counters table.
4701                                           ;*              NDERPT - Contains number of char errors to report on a line.
4702                                           ;* INPUTS TO SUBROUTINES:         CHCNTB, DPENDB, DPLEN, DPRSQE, EXCNTB, RXCNTB,
4703                                           ;*                               RXPTRB, ERRNBR, ERMSG, ERRTYP.
4704                                           ;*
4705                                           ;* OUTPUTS:     ERRBLK - Contents destroyed.
4706                                           ;*           Following variables updated for line on which char was received:
4707                                           ;*              DPRSQ - Data pattern resync que of received characters.
4708                                           ;*              ERCNT - Count of the number of character errors on line.
4709                                           ;*              ERSMRF - Updated "print error summary for line" flags.
4710                                           ;*              EXCNT - Count of the number of extra chars received on line.
4711                                           ;*              RXCNT - Count of the number of characters received on line.
4712                                           ;*              RXPTR - Updated to point to the next expected char on line.
4713                                           ;*
4714                                           ;* CALLING SEQUENCE:    JSR     PC,NEWCHR
4715                                           ;*
4716                                           ;* COMMENTS:    This routine can report errors with numbers Initial ERRNBR
4717                                           ;*              and Initial ERRNBR + 1.  ERRNBR is restored to its initial
4718                                           ;*              value before this routine returns.
4719                                           ;*
4720                                           ;* SUBROUTINES CALLED: CKCHR,CKINAC,TXROFF,TXRON.
4721                                           ;* INDIRECT SUBROUTINES: CHKEXT,CHKLOS,ER9002,ER9003,UPDCHR.
4722                                           ;-- ********************************************************************
4723
4724 022104                            NEWCHR:: SAVE                               ;SAVE CONTENTS OF GPRS R0 THRU R5.
     022104   004537   005326                                JSR     R5,PREG05    ;CALL REGISTER SAVE SUBRT.
4725 022110   010305                                MOV     R3,R5                 ;GET THE BIT MAP OF INACTIVE DATA BYTE BITS.
4726 022112   052705   177400                       BIS     #177400,R5            ;ALL UPPER BITS OF EXPECTED DATA ARE INACTIVE.
4727 022116   005037   022362                        CLR     70$                  ;CLEAR THE "ERROR FOUND" FLAG.
4728                                           ;+
4729                                           ; If the new character is valid on an inactive line, go report error.
4730                                           ; Routine used also extracts line number from the new character.
4731                                           ;-
4732 022122   004737   017046                        JSR     PC,CKINAC            ;CHECK FOR CHAR ON INACTIVE LINE.
4733 022126   103043                                 BCC     4$                   ;GO REPORT ERROR IF ON INACTIVE LINE.
4734                                           ;+
4735                                           ; Push the new character on the resync que for this line.
4736                                           ;-
4737 022130   010304                                 MOV     R3,R4                ;CALCULATE BASE ADDRESS OF THE
4738 022132   006304                                 ASL     R4                   ;  DATA PATTERN RESYNCH QUEUE
4739 022134   006304                                 ASL     R4                   ;  (QUEUE IS 4 WORDS LONG) FOR
```

```
4740 022136  062704  004644                    ADD    #DPRSQB,R4       ;   THIS LINE.
4741 022142  010401                            MOV    R4,R1            ;GET THE BASE OF THE QUEUE.
4742 022144  016121  000002                    MOV    2(R1),(R1)+      ;MOVE FROM CHR1 SLOT TO CHR0 SLOT.
4743 022150  016121  000002                    MOV    2(R1),(R1)+      ;MOVE FROM CHR2 SLOT TO CHR1 SLOT.
4744 022154  010211                            MOV    R2,(R1)          ;PUT NEW CHAR INTO CHR2 SLOT.
4745                                    ;+
4746                                    ; Check the DATA.VALID for the character at the bottom of the queue.
4747                                    ; If DATA.VALID is clear, exit the routine--nothing to analyze.
4748                                    ;-
4749 022156  011402                            MOV    (R4),R2          ;GET CHR0 VALUE, SET FLAGS.
4750 022160  100076                            BPL    60$              ;EXIT ROUTINE IF DATA.VALID IS CLEAR.
4751                                    ;+
4752                                    ; Test for any of the error bits set in CHR0.
4753                                    ;-
4754 022162  032702  070000                    BIT    #70000,R2        ;TEST FOR ANY CHR0 ERROR BITS SET.
4755 022166  001420                            BEQ    2$               ;SKIP THIS ERROR IF NO ERROR BITS SET.
4756                                    ;+
4757                                    ; We have at least one error flag set on the received char.
4758                                    ; Report data error flag error if not in summary mode.
4759                                    ;-
4760 022170  005337  022362                    DEC    70$              ;SET THE "ERROR FOUND" FLAG.
4761 022174  016300  005236                    MOV    TXRXLB(R3),R0    ;GET THE TX LINE OFFSET FOR THIS RX LINE.
4762 022200  036037  002366  002502            BIT    BITTBL(R0),ERSMRF  ;CHECK THE ERROR SUMMARY FLAG FOR TX LINE.
4763 022206  001010                            BNE    2$               ;IF ERROR SUMMARY FLAG SET, SKIP NEXT REPORT.
4764 022210  012737  014670  005324            MOV    #ER9003,ERRBLK   ;SELECT THE ER9003 ERROR REPORT ROUTINE.
4765 022216  004737  026370                    JSR    PC,TXROFF        ;TURN OFF TX AND RX DURING ERROR REPORTING.
4766 022222                                    ERROR                   ;            >>>>> ERROR <<<<<.
     022222  104460                                                                    TRAP    C$ERROR
4767 022224  004737  026424                    JSR    PC,TXRON         ;TURN TX AND RX BACK ON.
4768                                    ;+
4769                                    ; Check the character at the bottom of the resync que for data errors.
4770                                    ;-
4771 022230  004737  016460            2$:     JSR    PC,CKCHR         ;CHECK THE CHR0 CHAR FOR ERRORS.
4772 022234  103424                            BCS    6$               ;SKIP ERROR REPORT IF CHR0 IS CORRECT.
4773                                    ;+
4774                                    ; We have some sort of data error so report it (unless in summary report mode).
4775                                    ;-
4776 022236  005337  022362            4$:     DEC    70$              ;SET THE "ERROR FOUND" FLAG.
4777 022242  016300  005236                    MOV    TXRXLB(R3),R0    ;GET THE TX LINE OFFSET FOR THIS RX LINE.
4778 022246  036037  002366  002502            BIT    BITTBL(R0),ERSMRF  ;CHECK THE ERROR SUMMARY FLAG FOR THIS LINE.
4779 022254  001014                            BNE    6$               ;SKIP ERROR REPORT IF ERROR SUMMARY FLAG SET.
4780 022256  012737  014522  005324            MOV    #ER9002,ERRBLK   ;SELECT THE ER9002 ERROR REPORT ROUTINE.
4781 022264  005237  005320                    INC    ERRNBR           ;SELECT INITIAL ERRNBR + 1.
4782 022270  004737  026370                    JSR    PC,TXROFF        ;TURN OFF TX AND RX DURING ERROR REPORTING.
4783 022274                                    ERROR                   ;            >>>>> ERROR <<<<<.
     022274  104460                                                                    TRAP    C$ERROR
4784 022276  004737  026424                    JSR    PC,TXRON         ;TURN TX AND RX BACK ON.
4785 022302  005337  005320                    DEC    ERRNBR           ;RESTORE INITIAL ERRNBR.
4786                                    ;+
4787                                    ; Count a character error if one occurred.
4788                                    ; Update the "report error summary" flag for line based on error count.
4789                                    ;-
4790 022306  005737  022362            6$:     TST    70$              ;CHECK THE "ERROR FOUND" FLAG.
4791 022312  001421                            BEQ    60$              ;SKIP COUNTING AN ERROR IF FLAG IS CLEAR.
4792 022314  005263  003304                    INC    ERCNTB(R3)       ;INCREMENT THE ERROR COUNTER FOR THIS LINE.
4793 022320  001002                            BNE    8$               ;SKIP SETTING COUNTER TO MAX IF NO OVERFLOW.
4794 022322  005363  003304                    DEC    ERCNTB(R3)       ;RESET THE ERROR COUNTER TO -1 (MAX VALUE).
```

GLOBAL SUBROUTINE   - NEWCHR -

```
4795 022326  005737  002166           8$:     TST     NDERPT              ;DISABLE ERROR SUMMARY FUNCTION IF
4796 022332  001411                            BEQ     60$                 ; NUMBER OF DATA ERRORS TO REPORT IS 0.
4797 022334  026337  003304  002166            CMP     ERCNTB(R3),NDERPT   ;COMPARE ERROR COUNT WITH # OF ERR'S TO RPT.
4798 022342  103405                            BLO     60$                 ;SKIP SETTING OF SUMMARY FLAG IF NOT TOO MANY.
4799 022344  016300  005236                    MOV     TXRXLB(R3),R0       ;GET THE TX LINE OFFSET FOR THIS RX LINE.
4800 022350  056037  002366  002502            BIS     BITTBL(R0),ERSMRF   ;SET "PRINT ERROR SUMMARY" FLAG FOR LINE.
4801
4802 022356                          60$:     PASS                        ;RESTORE GPRS.
     022356  004736                                    JSR     PC,@(SP)+           ;RETURN TO PREGO5 SUBRT.
4803 022360  000207                            RTS     PC
4804
4805 022362  000000                  70$:     .WORD   0                   ;LOCAL STORAGE FOR ERROR OCCURRED FLAG.
```

```
4807                                    .SBTTL  GLOBAL SUBROUTINE                - OOPS -
4808                                    ;++ ***************************************************************************
4809                                    ;*                      - Program abort subroutine -
4810                                    ;*      This subroutine is used to abort the program when a fatal error is
4811                                    ;*      detected in the program or the host system hardware.  An error message
4812                                    ;*      is printed giving some information about the nature of the abort.
4813                                    ;*
4814                                    ;* INPUTS:      R1 - Error code giving reason for abort.
4815                                    ;*
4816                                    ;* OUTPUTS:     An error message is printed.
4817                                    ;*              A list of return PC values for all subroutine calls is printed.
4818                                    ;*
4819                                    ;* CALLING SEQUENCE:    JSR    PC,OOPS
4820                                    ;*
4821                                    ;* COMMENTS:
4822                                    ;*
4823                                    ;* SUBORDINATE ROUTINES CALLED: None.
4824                                    ;-- ***************************************************************************
4825
4826  022364                  OOPS::   SAVE                         ;SAVE CONTENTS OF GPRS R0 THRU R5.
       022364  004537  005326                   JSR    R5,PREG05    ;CALL REGISTER SAVE SUBRT.
4827                             ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
4828  022370                             ERRSF  101,EM0101
       022370  104454                                                      TRAP    C$ERSF
       022372  000145                                                      .WORD   101
       022374  022430                                                      .WORD   EM0101
       022376  000000                                                      .WORD   0
4829                                   ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
4830  022400                             PRINTF #EM0102
       022400  012746  022514                                              MOV     #EM0102,-(SP)
       022404  012746  000001                                              MOV     #1,-(SP)
       022410  010600                                                      MOV     SP,R0
       022412  104417                                                      TRAP    C$PNTF
       022414  062706  000004                                              ADD     #4,SP
4831  022420                  2$:      BREAK                        ;LOOK FOR OPERATOR CONTROL-C INPUT.
       022420  104422                                                      TRAP    C$BRK
4832  022422  000776                   BR     2$                    ;INFINITE LOOP.
4833  022424                  60$:     PASS                         ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
       022424  004736                           JSR    PC,@(SP)+     ;RETURN TO PREG05 SUBRT.
4834  022426  000207                   RTS    PC                   ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.
4835
4836                                    .NLIST BEX
4837  022430   110   117   123 EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
4838  022514   045   116   045 EM0102:: .ASCIZ /$N$APROGRAM HUNG, WAITING FOR A CONTROL-C. <***************$N$N/
4839                                    .EVEN
```

```
4841                                 .SBTTL  GLOBAL SUBROUTINE                    - PRFRME -
4842                                 ;++ **********************************************************************
4843                                 ;*                          - PROCESS FRAMING ERRORS -
4844                                 ;*      This subroutine is used in the Framing error bit test, to verify that
4845                                 ;*      all received characters have their framing error bit set and parity
4846                                 ;*      error bit clear.
4847                                 ;*
4848                                 ;* INPUTS:         R2 - Contains the character read from the FIFO.
4849                                 ;*                 ERRNBR - Error number of errors in this routine.
4850                                 ;*                 ERSMRF - "Report Error Summary for line" flags
4851                                 ;*
4852                                 ;* OUTPUTS:        ERRBLK - The contents of this word are destroyed.
4853                                 ;*                 ERCNTB - The error count for this line is updated.
4854                                 ;*                 Messages may be printed at the operators console.
4855                                 ;*
4856                                 ;*
4857                                 ;* CALLING SEQUENCE:    JSR    PC,PRFRME
4858                                 ;*
4859                                 ;* COMMENTS:       This routine reports errors with INITIAL number.
4860                                 ;*                 ERRNBR is restored to its initial value before this subroutine
4861                                 ;*                 returns.
4862                                 ;*
4863                                 ;* SUBORDINATE ROUTINES CALLED: ER6201.
4864                                 ;-- **********************************************************************
4865
4866 022612                         PRFRME::SAVE                              ;SAVE CONTENTS OF GPRS R0 THRU R5.
     022612   004537   005326                       JSR    R5,PREG05         ;CALL REGISTER SAVE SUBRT.
4867 022616   013704   005320               MOV    ERRNBR,R4           ;SAVE THE CONTENTS OF THE INITIAL ERROR NUMBER.
4868 022622   005005                         CLR    R5                 ;CLEAR ERROR/MESSAGE FLAGS.
4869
4870                                 ;+
4871                                 ; Test Framing and parity error bits in turn.  Report any errors found, ie.
4872                                 ;  Framing error bit clear, or Parity error bit set.
4873                                 ;-
4874 022624   012737   014204   005324     MOV    #ER6201,ERRBLK     ;SET UP THE ADDRESS OF THE ERROR ROUTINE.
4875 022632   032702   020000               BIT    #BIT13,R2          ;CHECK ON STATE OF THE FRAMING ERROR BIT.
4876 022636   001002                         BNE    6$                 ;BRANCH IF FRAMING ERROR BIT SET.
4877 022640   052705   000002                 BIS    #BIT1,R5           ;SET REPORT FRAMING ERROR FLAG.
4878
4879 022644   032702   010000         6$:     BIT    #BIT12,R2          ;CHECK ON THE STATE OF THE PARITY ERROR BIT.
4880 022650   001402                         BEQ    8$                 ;BRANCH IF PARITY ERROR BIT CLEAR.
4881 022652   052705   000014                 BIS    #14,R5             ;SET REPORT "PARITY ERROR SET" FLAGS.
4882 022656   005705                 8$:     TST    R5                 ;CHECK IF ANY ERROR FLAGS SET.
4883 022660   001407                         BEQ    60$                ;EXIT IF ALL FLAGS CLEAR.
4884 022662   036337   002366   002502       BIT    BITTBL(R3),ERSMRF  ;CHECK THE ERROR SUMMARY FLAG FOR THIS LINE.
4885 022670   001001                         BNE    10$                ;SKIP ERROR REPORT IF ERROR SUMMARY FLAG SET.
4886
4887                                 ;REPORT ERROR "CHARACTER RECEIVED WITH PARITY/FRAMING ERROR BIT SET".
4888 022672                         ERROR                        ;                >>>>> ERROR <<<<<.
     022672   104460                                                   TRAP   C$ERROR
4889 022674   005263   003304         10$:    INC    ERCNTB(R3)         ;INCREMENT ERROR COUNT FOR THIS LINE.
4890 022700   010437   005320         60$:    MOV    R4,ERRNBR          ;RESTORE ERROR NUMBER.
4891 022704                         PASS                           ;RESTORE GPRS.
     022704   004736                                     JSR    PC,@(SP)+            ;RETURN TO PREG05 SUBRT.
4892 022706   000207                         RTS    PC
```

```
4894                                    .SBTTL  GLOBAL SUBROUTINE                - PRPARE -
4895                                    ;++ **********************************************************************
4896                                    ;*                    - PROCESS PARITY ERRORS -
4897                                    ;*       This subroutine is used in the Parity error test, to verify that
4898                                    ;*       all received characters have their parity error bit set and framming
4899                                    ;*       error bit clear.
4900                                    ;*
4901                                    ;* INPUTS:          R2 - Contains the character read from the FIFO.
4902                                    ;*                  R3 - Contains 2 * line number of the read char.
4903                                    ;*                  ERRNBR - Error number of errors in this routine.
4904                                    ;*                  ERSMRF - "Report Error Summary for line" flags
4905                                    ;*
4906                                    ;* OUTPUTS:         ERRBLK - The contents of this word are destroyed.
4907                                    ;*                  ERCNTB - The error count for this line is updated.
4908                                    ;*                  Messages may be printed at the operators console.
4909                                    ;*
4910                                    ;*
4911                                    ;* CALLING SEQUENCE:    JSR    PC,PRPARE
4912                                    ;*
4913                                    ;* COMMENTS:        This routine reports errors with INITIAL ERRNBR thru ERRNBR+1.
4914                                    ;*                  ERRNBR is restored to its initial value before this subroutine
4915                                    ;*                  returns.
4916                                    ;*                  The contents of the ERRBLK are destroyed.
4917                                    ;*
4918                                    ;* SUBORDINATE ROUTINES CALLED: ER9002,ER6201.
4919                                    ;-- **********************************************************************
4920
4921 022710                            PRPARE::SAVE                         ;SAVE CONTENTS OF GPRS R0 THRU R5.
     022710  004537  005326                        JSR    R5,PREG05        ;CALL REGISTER SAVE SUBRT.
4922 022714  013746  005320                MOV    ERRNBR,-(SP)            ;SAVE THE CONTENTS OF THE INITIAL ERROR NUMBER.
4923 022720  005005                        CLR    R5                      ;CLEAR ERROR/MESSAGE FLAGS.
4924
4925                                    ;+
4926                                    ; Test Framming and parity error bits in turn.  Report any errors found, ie.
4927                                    ;   Parity error bit clear, or Framming error bit set.
4928                                    ;-
4929 022722  012737  014204  005324        MOV    #ER6201,ERRBLK          ;SET UP THE ADDRESS OF THE ERROR ROUTINE.
4930 022730  032702  010000                BIT    #BIT12,R2               ;CHECK ON STATE OF THE PARITY ERROR BIT.
4931 022734  001002                        BNE    6$                      ;BRANCH IF PARITY ERROR BIT SET.
4932 022736  052705  000010                BIS    #BIT3,R5                ;SET REPORT PARITY ERROR FLAG.
4933 022742  032702  020000        6$:     BIT    #BIT13,R2               ;CHECK ON THE STATE OF THE FRAMMING ERROR BIT.
4934 022746  001402                        BEQ    8$                      ;BRANCH IF FRAMMING ERROR BIT CLEAR.
4935 022750  052705  000003                BIS    #3,R5                   ;SET REPORT "FRAMMING ERROR SET" FLAGS.
4936 022754  005705                8$:     TST    R5                      ;CHECK IF ANY ERROR FLAGS SET.
4937 022756  001405                        BEQ    12$                     ;BRANCH TO MAKE DATA CHECK IF ALL FLAGS CLEAR.
4938 022760  036337  002366  002502        BIT    BITTBL(R3),ERSMRF       ;CHECK THE ERROR SUMMARY FLAG FOR THIS LINE.
4939 022766  001024                        BNE    14$                     ;SKIP ALL ERROR REP IF IN ERROR SUMMARY MODE.
4940                                    ;REPORT ERROR "CHAR RECEIVED WITH PARITY/FRAMMING ERROR BIT SET/CLEAR".
4941 022770                                ERROR                          ;            >>>>> ERROR <<<<<.
     022770  104460                                TRAP   C$ERROR
4942
4943
4944                                    ;+
4945                                    ; Compare actual data with expected data to check for multiple errors.
4946 022772  005237  005320            ;-
                                        12$:    INC    ERRNBR              ;INCREMENT ERROR NUMBER.
4947 022776  016304  003404                MOV    RXPTRB(R3),R4           ;GET THE POINTER TO THE EXPECTED DATA.
4948 023002  111404                        MOVB   (R4),R4                 ;GET THE EXPECTED DATA.
```

```
4949 023004  120204                        CMPB    R2,R4           ;COMPARE ACTUAL AND EXPECTED DATA.
4950 023006  001424                        BEQ     18$             ;SKIP ERROR REPORT IF DATA CORRECT.
4951 023010  042704  100000                BIC     #BIT15,R4       ;CLEAR "NONE" EXPECTED MESSAGE FLAG.
4952 023014  036337  002366  002502        BIT     BITTBL(R3),ERSMRF  ;CHECK THE ERROR SUMMARY FLAG FOR THIS LINE.
4953 023022  001014                        BNE     16$             ;SKIP ERROR REPORT IF ERROR SUMMARY FLAG SET.
4954 023024  036337  002366  002506        BIT     BITTBL(R3),RXDONF  ; CHECK FOR RECEPTION COMPLETE ON THIS LINE.
4955 023032  001402                        BEQ     14$             ;SKIP SETTING NONE EXPECTED FLAG.
4956 023034  052704  100000                BIS     #BIT15,R4       ;SET "NONE" EXPECTED MESSAGE FLAG.
4957 023040  012701  011477        14$:    MOV     #EM9008,R1      ;SELECT ERROR MESSAGE TO BE REPORTED.
4958 023044  012737  014522  005324        MOV     #ER9002,ERRBLK  ;SELECT ERROR REPORTING ROUTINE.
4959                                        ;REPORT ERROR"RECEIVE CHARACTER MISCOMPARE"
4960 023052                                 ERROR
     023052  104460                                                                           TRAP    C$ERROR
4961
4962 023054  005263  003304        16$:    INC     ERCNTB(R3)      ;INCREMENT ERROR COUNT FOR THIS LINE.
4963 023060  012637  005320        18$:    MOV     (SP)+,ERRNBR    ;RESTORE ERROR NUMBER.
4964
4965 023064                        60$:    PASS                    ;RESTORE GPRS.
     023064  004736                                         JSR    PC,@(SP)+             ;RETURN TO PREG05 SUBRT.
4966 023066  000207                        RTS     PC
```

```
4968                                .SBTTL  GLOBAL SUBROUTINE                   - PRTLPR -
4969                                ;++ ******************************************************************************
4970                                ;*                          -Print the contents of the LPR.
4971                                ;*          This routine is used to print out extended information on the
4972                                ;*          contents of the Line Parameter Register (LPR).
4973                                ;*
4974                                ;* INPUTS:        R3 - Contains the number of the line you wish to examine.
4975                                ;*                CSRA - Contains the address of the DUT's CSR.
4976                                ;*                IESTAT - Contains the current status of the TX and RX interrupt
4977                                ;*                    enable bits in the DUT's CSR.
4978                                ;*                LPRA - Contains the address of the DUT's LPR register.
4979                                ;*
4980                                ;* OUTPUTS:       An extended information message is printed on the operators
4981                                ;*                console.
4982                                ;*
4983                                ;* CALLING SEQUENCE:    JSR    PC,PRTLPR
4984                                ;*
4985                                ;* COMMENTS:      This routine changes the indirect address field of the device
4986                                ;*                under test's CSR.
4987                                ;*
4988                                ;* SUBORDINATE ROUTINES CALLED: NONE.
4989                                ;-- ******************************************************************************
4990
4991 023070                        PRTLPR::SAVE                        ;SAVE CONTENTS OF GPRS R0 THRU R5.
     023070  004537  005326                      JSR    R5,PREG05        ;CALL REGISTER SAVE SUBRT.
4992 023074  013701  002202                MOV    CSRA,R1             ;GET THE CSR ADDRESS.
4993 023100  013702  002206                MOV    LPRA,R2             ;GET THE LPR ADDRESS.
4994 023104  042703  177760                BIC    #177760,R3          ;CLEAR ANY UNWANTED BITS.
4995 023110  053703  002234                BIS    IESTAT,R3           ;SET STATE OF TX AND RX INTERRUPT ENABLE BITS.
4996 023114  010311                        MOV    R3,(R1)             ;SELECT LINE.
4997 023116  011204                        MOV    (R2),R4             ;GET CONTENTS OF THE LPR.
4998                                ;PRINT MESSAGE"CONTENTS OF THE LPR:nnnnnn"
4999 023120                        PRINTX #EF9019,#EM9026,R4;PRINT OUT MESSAGE ON OPERATORS CONSOLE.
     023120  010446                                                              MOV    R4,-(SP)
     023122  012746  012203                                                      MOV    #EM9026,-(SP)
     023126  012746  007213                                                      MOV    #EF9019,-(SP)
     023132  012746  000003                                                      MOV    #3,-(SP)
     023136  010600                                                              MOV    SP,R0
     023140  104415                                                              TRAP   C$PNTX
     023142  062706  000010                                                      ADD    #10,SP
5000 023146                        60$:    PASS                        ;RESTORE GPRS.
     023146  004736                                      JSR    PC,@(SP)+                ;RETURN TO PREG05 SUBRT.
5001 023150  000207                        RTS    PC
```

```
5003                                 .SBTTL  GLOBAL SUBROUTINE                    - PUFIFO -
5004                                 ;********************************************************************************
5005                                 ;*      - PURGE THE FIFO
5006                                 ;*         This routine tries to remove all the characters from the FIFO.
5007                                 ;*         Any BMP codes that are found are saved on the BMP code queue.
5008                                 ;*
5009                                 ;* INPUTS:        RBUFA- Contains the  address of the Receiver.
5010                                 ;*
5011                                 ;*
5012                                 ;* OUTPUTS:       Carry bit - Indicates the state of the fifo, set:= purged.
5013                                 ;*                BMPCQ - The contents of the BMP code queue may be updated.
5014                                 ;*
5015                                 ;* CALLING SEQUENCE:   JSR    PC,PUFIFO
5016                                 ;*
5017                                 ;* COMMENTS:
5018                                 ;*
5019                                 ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
5020                                 ;********************************************************************************
5021
5022 023152                         PUFIFO::SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
     023152  004537  005326                  JSR     R5,PREG05         ;CALL REGISTER SAVE SUBRT.
5023 023156  012701  001000                  MOV     #512.,R1          ;SET MAXIMUM TRY COUNT OF 512.
5024 023162  013704  002204                  MOV     RBUFA,R4          ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.
5025
5026 023166  011402             2$:          MOV     (R4),R2           ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
5027 023170  100016                          BPL     6$                ;EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.
5028                                 ;+
5029                                 ; Check if the read character is actually a BMP code.
5030                                 ;  If it is, then save it on the BMP code queue to be reported later.
5031                                 ;-
5032 023172  012700  070000                  MOV     #70000,R0         ;GENERATE A BIT MAP OF CHAR ERROR BITS
5033 023176  040200                          BIC     R2,R0             ; WHICH ARE NOT SET FOR CHAR.
5034 023200  001006                          BNE     4$                ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
5035                                 ;+
5036                                 ; Check if the read data is modem status , BMP or Selftest?.
5037                                 ;-
5038 023202  012700  000300                  MOV     #300,R0           ; CHECK IF BMP OR SELFTEST?.
5039 023206  040200                          BIC     R2,R0             ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
5040 023210  001002                          BNE     4$                ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
5041 023212  004737  024726                  JSR     PC,SAVBMP         ;SAVE BMP CODE ON THE QUEUE.
5042
5043 023216  005301             4$:          DEC     R1                ;DECREMENT THE TRY COUNT.
5044 023220  001362                          BNE     2$                ;LOOP TO TRY AGAIN.
5045 023222  000241                          CLC                       ;CLEAR CARRY,TO INDICATE FIFO NOT PURGED.
5046 023224  000401                          BR      60$               ;EXIT WITH CARRY CLEAR.
5047 023226  000261             6$:          SEC                       ;SET CARRY, TO INDICATE FIFO PURGED.
5048
5049 023230                      60$:         PASS                      ;RESTORE GPRS,
     023230  004736                           JSR     PC,@(SP)+                    ;RETURN TO PREG05 SUBRT.
5050                                                                    ;CARRY BIT, SET INDICATES FIFO PURGED.
5051 023232  000207                          RTS     PC
```

```
5053                                      .SBTTL  GLOBAL SUBROUTINE              - PUFIFR -
5054                                      ;*******************************************************************************
5055                                      ;*      - Purge FIFO report any errors found.
5056                                      ;*      This routine removes all data from the FIFO.  Any BMP codes that are
5057                                      ;*      found are save on the queue to be reported later in the BMP report test.
5058                                      ;*      Any unxpected data (ie any non-status inforamtion) that are found,
5059                                      ;*      are reported as an error.
5060                                      ;*      If the FIFO will not purge after 512 attempts, then the current test
5061                                      ;*      that called this routine receives a failure flag that should be used
5062                                      ;*      to abort the test.
5063                                      ;*
5064                                      ;* INPUTS:       ERRTBL - ERRTYPE, ERRMSG, ERRNBR are set up correctly.
5065                                      ;*               RBUFA- Contains the  address of the Receiver.
5066                                      ;*
5067                                      ;* OUTPUTS:      Carry bit - Abort test flag, Clr = ABORT TEST, Set = OK.
5068                                      ;*               ERRBLK - Value will be dastroyed.
5069                                      ;*               BMPCQP - The BMP code queue pointer may be updated.
5070                                      ;*               The contents of the BMP code queue may be udated.
5071                                      ;*
5072                                      ;* CALLING SEQUENCE:    JSR     PC,PUFIFR
5073                                      ;*
5074                                      ;* COMMENTS:     This routine reports errors with numbers initial ERRNBR
5075                                      ;*               thru to ERRNBR+2.
5076                                      ;*               The ERRNBR is restored to its INITIAL value before returning.
5077                                      ;*
5078                                      ;* SUBORDINATE ROUTINES CALLED: ER1603,ER9001,ER9002,SAVBMP.
5079                                      ;*******************************************************************************
5080
5081 023234                              PUFIFR::SAVE                         ;SAVE CONTENTS OF GPRS R0 THRU R5.
     023234  004537  005326                              JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
5082 023240  013746  005320                      MOV     ERRNBR,-(SP)    ;SAVE THE CONTENTS OF THE ERROR NUMBER.
5083 023244  012705  001000                      MOV     #512.,R5        ;SET MAXIMUM READ COUNTER TO 2*FIFO SIZE.
5084
5085                                      ; Read data from the FIFO until DATA VALID is clear of read counter is zero.
5086                                      ;  Report any BMP or Unexpected data as errors.
5087                                      ;-
5088 023250  017702  156730              2$:     MOV     @RBUFA,R2       ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
5089 023254  100057                              BPL     8$              ;EXIT IF DATA VALID CLEAR, ie. FIFO PURGED.
5090
5091                                      ; Check if read data is status or unexpected character.
5092                                      ;-
5093 023256  012700  070000                      MOV     #70000,R0       ;GENERATE A BIT MAP OF CHAR ERROR BITS
5094 023262  040200                              BIC     R2,R0           ; WHICH ARE NOT SET FOR CHAR.
5095 023264  001012                              BNE     4$              ;SKIP BMP CHECK IF IT IS UNEXPECTED DATA.
5096
5097                                      ; Check if the read data is modem status , BMP or Selftest?.
5098                                      ;  If it is a BMP code then save it on the queue.
5099                                      ;-
5100 023266  012737  014432  005324              MOV     #ER9001,ERRBLK  ;SET UP THE CORRECT ERROR REPORTING ROUTINE.
5101 023274  012700  000300                      MOV     #300,R0         ; CHECK IF BMP OR SELFTEST?.
5102 023300  040200                              BIC     R2,R0           ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
5103 023302  001003                              BNE     4$              ;SKIP BMP ERROR REPORT IF MODEM OR SELFTEST?.
5104 023304  004737  024726                      JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
5105 023310  000424                              BR      6$              ;BRANCH TO CHECK READ COUNT.
5106
5107                                      ; Check if the read data is Modem, Selftest or Unexpected data.
5108                                      ;-
```

```
5109 023312  032702  000001        4$:    BIT    #BIT0,R2         ;TEST THE MODEM STATUS INDICATION BIT.
5110 023316  001421                       BEQ    6$               ;DO NOT REPORT ANY ERROR IF MODEM STATUS.
5111 023320  012701  012566               MOV    #EM9104,R1       ;PASS THE CORRECT ERROR MESSAGE TO REPORT.
5112 023324  010203                        MOV    R2,R3            ;EXTRACT THE LINE NUMBER FROM
5113 023326  000303                        SWAB   R3               ; THE READ DATA.
5114 023330  042703  177760                BIC    #177760,R3       ;
5115 023334  006303                        ASL    R3               ;FORM LINE NUMBER TIMES 2 FOR ER9002 ROUTINE.
5116 023336  052704  100000                BIS    #BIT15,R4        ;SET THE "NONE" EXPECTED MAESSAGE FLAG.
5117 023342  005237  005320                INC    ERRNBR           ;SET ERROR NUMBER TO INTIAL ERRBR+1.
5118 023346  012737  014522  005324        MOV    #ER9002,ERRBLK   ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
5119                                 ;REPORT ERROR "UNEXPECTED DATA FOUND IN FIFO".
5120 023354                               ERROR            ;                    >>>>> ERROR <<<<<.
     023354  104460                                                        TRAP    C$ERROR
5121 023356  005337  005320                DEC    ERRNBR           ;RESTORE ERROR NUMBER TO INTIAL ERRNBR.
5122
5123 023362  005305                 6$:    DEC    R5               ;DECREMENT READ COUNTER.
5124 023364  001331                        BNE    2$               ;LOOP TO READ NEXT CHAR FROM FIFO IF COUNT > 0.
5125                                 ;+
5126                                 ; The FIFO will not clear, report the error and indicate that the test is to
5127                                 ; be ABORTED.
5128                                 ;-
5129 023366  062737  000002  005320        ADD    #2,ERRNBR        ;SET ERROR NUMBER TO INTIAL ERRNBR+2.
5130 023374  012737  014124  005324        MOV    #ER1603,ERRBLK   ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
5131 023402  012701  011776               MOV    #EM9017,R1       ;PASS THE MESSAGE TO BE REPORTED.
5132                                 ;REPORT THE ERROR "FIFO WILL NOT PURGE, (DATA VALID STUCK SET)"
5133                                 ;                  "?????? TEST ABORTED".
5134 023406                               ERROR            ;                    >>>>> ERROR <<<<<.
     023406  104460                                                        TRAP    C$ERROR
5135 023410  000241                        CLC                     ;INDICATE THE TEST IS TO BE ABORTED.
5136 023412  000401                        BR     10$              ;EXIT THIS ROUTINE AND ABORT THE CURRENT TEST.
5137
5138 023414  000261                 8$:    SEC                     ;SET THE CARRY, DO NOT ABORT THE TEST.
5139
5140 023416  012637  005320         10$:   MOV    (SP)+,ERRNBR     ;RESTORE INITIAL ERROR NUMBER.
5141 023422                         60$:   PASS                    ;RESTORE GPRS,
     023422  004736                                JSR    PC,@(SP)+                ;RETURN TO PREG05 SUBRT.
5142                                                                ;CARRY BIT, SET INDICATES FIFO PURGED, DO NOT
5143                                                                ; ABORT THE TEST.
5144 023424  000207                        RTS    PC
```

```
5146                                    .SBTTL  GLOBAL SUBROUTINE                     - PURRXB -
5147                                    ;++  ***********************************************************************
5148                                    ;*                     - Purge the RX Buffer in Memory Routine -
5149                                    ;*        This subroutine is used before the beginning of a TX/RX of data
5150                                    ;*        patterns to clear out the RX buffer and to initialize the various
5151                                    ;*        counters and pointers related to that buffer.
5152                                    ;*
5153                                    ;* INPUTS:        RXBSTA - Label at the beginning of the RX buffer.
5154                                    ;*
5155                                    ;* OUTPUTS:       RXBCNT - Count of # of chars in RX buffer (Cleared).
5156                                    ;*                RXBIPT - Input pointer to RX buffer (Initialized).
5157                                    ;*                RXBOPT - Output pointer to RX buffer (Initialized).
5158                                    ;*                The contents of the RX BUFFER are cleared.
5159                                    ;*
5160                                    ;* CALLING SEQUENCE:     JSR     PC,PURRXB
5161                                    ;*
5162                                    ;* COMMENTS:
5163                                    ;*
5164                                    ;* SUBORDINATE ROUTINES CALLED: None.
5165                                    ;-- ***********************************************************************
5166
5167 023426                            PURRXB:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
     023426  004537  005326                     JSR      R5,PREG05      ;CALL REGISTER SAVE SUBRT.
5168
5169 023432  012701  002714                     MOV      #RXBOPT,R1      ;GET THE ADDRESS OF THE RX OUTPUT POINTER.
5170 023436  012721  002722                     MOV      #RXBSTA,(R1)+   ;INITIALIZE THE RX BUFFER OUTPUT POINTER.
5171 023442  012721  002722                     MOV      #RXBSTA,(R1)+   ;INITIALIZE THE RX BUFFER INPUT POINTER.
5172 023446  005021                2$:   CLR      (R1)+           ;CLEAR CHAR COUNT AND THE BUFFER AREA.
5173 023450  020127  003122              CMP      R1,#RXBEND      ;CHECK IF LAST LOCATION HAS BEEN CLEARED.
5174 023454  101774                      BLOS     2$              ;LOOP IF NOT DONE.
5175
5176 023456                        60$:  PASS                            ;RESTORE GPRS.
     023456  004736                       JSR      PC,@(SP)+                   ;RETURN TO PREG05 SUBRT.
5177 023460  000207                      RTS      PC
```

```
5179                                       .SBTTL  GLOBAL SUBROUTINE                - RDCHRS -
5180                                       ;++ ***************************************************************************
5181                                       ;*               - Read and Compare Input Characters Routine -
5182                                       ;*      This subroutine reads the characters from the RX buffer in memory.
5183                                       ;*      If characters stop appearing in the buffer with DATA.VALID set,
5184                                       ;*      or if more than the allowable number of characters has been read from
5185                                       ;*      the buffer this routine exits with an RX complete indication.
5186                                       ;*      Each read char is analyzed and any necessary errors are reported.
5187                                       ;*
5188                                       ;* INPUTS:        ACTLNS - Bit map of the active DUT lines.
5189                                       ;*                ERRNBR - Set to error number of first error in this routine.
5190                                       ;*                IBM - Mask of the inactive bits in a TX or RX char byte.
5191                                       ;*                OSTEND - Address of the end of the output storage fifo buffer.
5192                                       ;*                OSTPTR - Pointer to the next byte to read from OSTORE.
5193                                       ;*                RXBOPT - Pointer into the RX char buffer in memory.
5194                                       ;*                RXTOUT - Time-out value for RX of last char.
5195                                       ;*
5196                                       ;* OUTPUTS:       Error messages may be printed at the operator's console.
5197                                       ;*                TXDBLF - TX/RX disabled flag (Cleared).
5198                                       ;*                TXENBM - TX.ENABLE state mask (Destroyed).
5199                                       ;*                SAVPRI - Storage for processor priority (Destroyed).
5200                                       ;*                SAVTEN - Storage for TX.ENABLE states (Destroyed).
5201                                       ;*
5202                                       ;* CALLING SEQUENCE:    JSR     PC,RDCHRS
5203                                       ;*
5204                                       ;* COMMENTS:      This routine reports errors with numbers Initial ERRNBR
5205                                       ;*                thru Initial ERRNBR + 4.
5206                                       ;*                ERRNBR is restored before this routine returns.
5207                                       ;*
5208                                       ;* SUBROUTINES CALLED:  CKCHR,NEWCHR,REPCOD,RXIE0,RXIE1,TXENBL,TXIE0,TXIE1,
5209                                       ;*                      WAIBIS.
5210                                       ;-- ***************************************************************************
5211
5212 023462                                RDCHRS:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     023462  004537  005326                            JSR     R5,PREG05        ;CALL REGISTER SAVE SUBRT.
5213 023466  013704  005320                   MOV     ERRNBR,R4               ;PRESERVE THE INITIAL ERROR NUMBER.
5214 023472  013703  002226                   MOV     IBM,R3                  ;GET THE INACTIVE BIT MASK.
5215 023476  005037  002510                   CLR     TXDBLF                  ;CLEAR THE TX DISABLED FLAG.
5216 023502  004737  024702                   JSR     PC,RXIE1                ;TURN ON DUT RECEPTION INTERRUPTS.
5217 023506  004737  026070                   JSR     PC,TXIE1                ;TURN ON DUT TRANSMISSION INTERRUPTS.
5218
5219                                       ;+
5220                                       ; Clear all resync queues for all lines.
5221                                       ;-
5222 023512  012701  004644                   MOV     #DPRSQB,R1              ;GET BASE ADDRESS OF RESYNC QUEUES TABLE.
5223 023516  012702  005044                   MOV     #DPRSQE,R2              ;GET END ADDRESS OF RESYNC QUEUES TABLE.
5224 023522  005021               2$:         CLR     (R1)+                   ;CLEAR A WORD OF THE TABLE.
5225 023524  020102                            CMP     R1,R2                   ;CHECK IF POINTER AT END OF TABLE.
5226 023526  103775                            BLO     2$                      ;LOOP UNTIL TABLE IS CLEAR.
5227                                       ;+
5228                                       ; Wait for a character to appear in the FIFO.
5229                                       ; If no character appears within time-out period: exit routine, we're done.
5230                                       ;-
5231 023530  013701  002242       4$:         MOV     RXTOUT,R1               ;GET TIME-OUT FOR SLOWEST BAUD RATE IN USE.
5232 023534  023737  002504  002174           CMP     TXDONF,ACTLNS          ;CHECK FOR TRANSMISSION DONE ON ACTIVE LINES.
5233 023542  001402                            BEQ     6$                      ;SKIP ADDING 50 MS DELAY IF TX DONE ALL LINES.
5234 023544  062701  000062                    ADD     #50.,R1                 ;ADD 50 MILLI SEC TO DELAY IF NOT LAST CHAR.
```

```
5235 023550 052701 170000      6$:     BIS     #170000,R1      ;INDICATE TO TEST DATA.VALID BIT.
5236 023554 013702 002714              MOV     RXBOPT,R2       ;INDICATE TO CHECK MEMORY RECEIVE BUFFER.
5237 023560 004737 027160              JSR     PC,WAIBIS       ;WAIT FOR RECEIVED CHAR OR TIME-OUT.
5238 023564 103073                     BCC     18$             ;EXIT ROUTINE IF TIME-OUT, WE'RE DONE.
5239
5240 023566 004737 020430              JSR     PC,GETCHR       ;READ A CHARACTER FROM THE MEMORY BUFFER.
5241                            ;+
5242                            ; Check if the TX ISR is disabled.
5243                            ; Re-enable RX ISR if the space for new chars is low enough.
5244                            ;  If the buffer can accomodate more chars then re-enable transmission.
5245                            ;-
5246 023572 005737 002510      8$:     TST     TXDBLF          ;CHECK IF TX IS DISABLED.
5247 023576 100024                     BPL     10$             ;SKIP RX/TX CHECK IF TX NOT DISABLED.
5248 023600 023727 002720 000020       CMP     RXBCNT,@RXBETX  ;COMPARE BUFFER COUNT WITH LEVEL TO ENABLE RX.
5249 023606 101020                     BHI     10$             ;SKIP ENABLE RX IF BUFFER TOO FULL.
5250 023610 004737 024702              JSR     PC,RXIE1        ;ENABLE RECEPTION INTERRUPTS.
5251 023614 013705 002250              MOV     TXENBM,R5       ;GET THE PRESERVED TX.ENABLE STATES.
5252 023620 023727 002720 000020       CMP     RXBCNT,@RXBETX  ;COMPARE BUFFER COUNT WITH LEVEL TO ENABLE TX.
5253 023626 101010                     BHI     10$             ;SKIP ENABLING TX IF BUFFER TOO FULL.
5254 023630 106701                     MFPS    R1              ;SAVE THE CURRENT PROCESSOR PRIORITY.
5255 023632 106427 000340              MTPS    #PRI07          ;DISABLE INTERRUPTS.
5256 023636 004737 025646              JSR     PC,TXENBL       ;ENABLE TRANSMISSION.
5257 023642 005037 002510              CLR     TXDBLF          ;CLEAR THE TX DISABLE FLAG.
5258 023646 106401                     MTPS    R1              ;RE-ENABLE INTERRUPTS.
5259 023650                    10$:
5260
5261 023650 005337 002500              DEC     CHRTOT          ;DECREMENT THE TOTAL CHAR COUNTER.
5262 023654 001011                     BNE     12$             ;SKIP ERROR IF NOT TOO MANY RECEIVED.
5263 023656 010437 005320              MOV     R4,ERRNBR       ;SET ERROR NUMBER TO INITIAL ERRNBR.
5264 023662 012701 012107              MOV     #EM9025,R1      ;SELECT THE PROPER ERROR MESSAGE.
5265 023666 012737 014100 005324       MOV     #ER0503,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
5266                            ;+
5267                            ; Report error at Initial ERRNBR.
5268                            ; "MORE THAN TWICE THE EXPECTED NUMBER OF CHARACTERS RECEIVED."
5269                            ;-
5270 023674                            ERROR                   ;              >>>>> ERROR <<<<<.
     023674 104460                                                    TRAP    C$ERROR
5271 023676 000452                     BR      60$             ;EXIT THE ROUTINE, WE'RE GIVING UP.
5272                            ;+
5273                            ; Determine if the character is data or a status code.
5274                            ;-
5275 023700 012700 070000      12$:    MOV     #70000,R0       ;GENERATE A BIT MAP OF CHARACTER ERROR BITS
5276 023704 040200                     BIC     R2,R0           ;  WHICH ARE NOT SET FOR THE CHARACTER.
5277 023706 001007                     BNE     14$             ;SKIP REPORTING OF ERROR CODE IF WE HAVE CHAR.
5278                            ;+
5279                            ; The data is either a BMP code or a Modem Status code.
5280                            ; Report that the code was found.
5281                            ; Errors reported with error numbers >>>>> ERRNBR+1 and ERRNBR+2 <<<<<.
5282                            ;-
5283 023710 010437 005320              MOV     R4,ERRNBR       ;GET THE ERROR NUMBER PASSED INTO THIS ROUTINE.
5284 023714 005237 005320              INC     ERRNBR          ;SET ERROR NUMBER TO INITIAL ERRNBR+1.
5285 023720 004737 024066              JSR     PC,REPCOD       ;REPORT THE BMP OR MODEM STATUS CHANGE CODE.
5286 023724 000407                     BR      16$             ;BRANCH TO GET THE NEXT CHARACTER.
5287                            ;+
5288                            ; The data is a valid character:
5289                            ;    Compare the read data with the expected data.
5290                            ;    Update expected data pointer.
```

```
5291                                             ;     Errors reported with error numbers >>>>> ERRNBR+3 and ERRNBR+4 <<<<<.
5292                                             ;-
5293 023726  010437  005320          14$:    MOV    R4,ERRNBR       ;CALCULATE THE STARTING ERROR NUMBER FOR THE
5294 023732  062737  000003  005320          ADD    #3,ERRNBR       ;  NEXT ROUTINE CALL (INITIAL ERRNBR+3).
5295 023740  004737  022104                  JSR    PC,NEWCHR       ;HANDLE THE NEW DATA CHARACTER.
5296                                             ;+
5297                                             ; Done processing this character.
5298                                             ; Read another char from the DUT FIFO.
5299                                             ; If DATA.VALID is set, loop to check the received character.
5300                                             ; If DATA.VALID is clear loop to wait for it set or time-out.
5301                                             ;-
5302 023744  004737  020430          16$:    JSR    PC,GETCHR       ;READ A CHARACTER FROM THE RX BUFFER.
5303 023750  103710                          BCS    8$              ;IF DATA.VALID SET, GO TO CHECK THE RX CHAR.
5304 023752  000666                          BR     4$              ;LOOP TO WAIT CHAR OR TIME-OUT IF BUFFER EMPTY.
5305                                             ;+
5306                                             ; Use dummy characters to force analysis of characters in resync queues.
5307                                             ;-
5308 023754  004737  024650          18$:    JSR    PC,RXIEO        ;TURN OFF DUT RX INTERRUPTS.
5309 023760  004737  025442                  JSR    PC,TXDONE       ;CHECK IF TX DONE, TURN OFF DUT TX INTERRUPTS.
5310 023764  005002                          CLR    R2              ;CLEAR THE DUMMY CHARACTER.
5311 023766  005001                          CLR    R1              ;CLEAR THE LOOP COUNTER.
5312 023770  004737  022104          20$:    JSR    PC,NEWCHR       ;FORCE ONE RESYNC QUE CHAR TO BE ANALYZED.
5313 023774  062702  000400                  ADD    #400,R2         ;INCREMENT THE LINE NUMBER IN THE DUMMY CHAR.
5314 024000  005201                          INC    R1              ;INCREMENT THE LOOP COUNTER.
5315 024002  120127  000010                  CMPB   R1,#NUMLNS      ;TEST FOR LOOP COUNTER EQUAL TO # OF DUT LINES.
5316 024006  002770                          BLT    20$             ;LOOP IF LOOP COUNT IS NOT ALL LINES DONE.
5317 024010  005701                          TST    R1              ;CHECK FOR SECOND TIME AROUND OUTER LOOP.
5318 024012  100404                          BMI    60$             ;EXIT IF OUTER LOOP DONE TWICE.
5319 024014  005002                          CLR    R2              ;CLEAR THE DUMMY CHAR FOR 2ND TIME AROUND LOOP.
5320 024016  012701  100000                  MOV    #100000,R1      ;CLEAR LOOP COUNT, SET OUTER LOOP FLAG.
5321 024022  000762                          BR     20$             ;LOOP THE SECOND TIME AROUND OUTER LOOP.
5322
5323 024024  010437  005320          60$:    MOV    R4,ERRNBR       ;RESTORE THE ERROR NUMBER TO ITS INITIAL VALUE.
5324 024030                                  PASS                   ;RESTORE GPRS.
     024030  004736                                  JSR    PC,@(SP)+              ;RETURN TO PREGO5 SUBRT.
5325 024032  000207                          RTS    PC
```

```
5327                                      .SBTTL   GLOBAL SUBROUTINE                 - RDMAST -
5328                                      ;++ ***********************************************************************
5329                                      ;*                   - Report DMA_START Bit Errors Routine -
5330                                      ;*      This subroutine checks for lines which have DMA_START bit errors
5331                                      ;*      during the just completed DMA transmission.  If any are found,
5332                                      ;*      they are reported.
5333                                      ;*
5334                                      ;* INPUTS:        ERRMSG - Address of primary error message for this routine.
5335                                      ;*                ERRNBR - Error number of error reported in this routine.
5336                                      ;*                TXINTF - Contains bit map of lines with DMA_START bit errors.
5337                                      ;*
5338                                      ;* OUTPUTS:       ERRBLK - Address of the error reporting routine (Destroyed).
5339                                      ;*                Messages may be printed at the operator console.
5340                                      ;*
5341                                      ;* CALLING SEQUENCE:    JSR    PC,RDMAST
5342                                      ;*
5343                                      ;* COMMENTS:   If no lines have DMA_START bit errors, no messages are printed.
5344                                      ;*
5345                                      ;* SUBORDINATE ROUTINES CALLED: ER9102.
5346                                      ;-- ***********************************************************************
5347
5348 024034                   RDMAST:: SAVE                      ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024034  004537  005326                      JSR    R5,PREG05        ;CALL REGISTER SAVE SUBRT.
5349 024040  013702  002252           MOV    TXINTF,R2      ;GET COPY OF THE DMA_START ERRORS BIT MAP.
5350 024044  001406                    BEQ    60$            ;EXIT IF NO DMA_START ERROR BITS ARE SET.
5351                                      ;+
5352                                      ; We have some DMA_START bit errors to report.
5353                                      ;-
5354 024046  012737  015430  005324    MOV    #ER9102,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
5355 024054  012701  012477            MOV    #EM9102,R1      ;INDICATE THAT WE HAVE DMA_START BIT ERROR.
5356                                      ;+
5357                                      ; Report "DMA_START BIT SET AFTER RESET OR TX.ACTION ... ON LINES(S):"
5358                                      ;-
5359 024060                            ERROR              ;            >>>>> ERROR <<<<<.
     024060  104460                               TRAP    C$ERROR
5360
5361 024062                   60$:     PASS               ;RESTORE GPRS.
     024062  004736                               JSR    PC,@(SP)+        ;RETURN TO PREG05 SUBRT.
5362 024064  000207            RTS    PC
```

```
5364                                    .SBTTL  GLOBAL SUBROUTINE              - REPCOD -
5365                                    ;++ **********************************************************************
5366                                    ;*               - Routine to Report Error Code From DUT -
5367                                    ;*        This routine reports an error code which has been read from the DUT
5368                                    ;*        FIFO.  The code is checked to determine whether it is a Selftest code
5369                                    ;*        an Modem Status Change code or a BMP code.  This routine assumes that
5370                                    ;*        the code indicates an error.  If a BMP code is found it is not reported
5371                                    ;*        immediately, but is saved on the BMP code queue to be reported later.
5372                                    ;*
5373                                    ;* INPUTS:        R2 - Contains the error code complete with flags and line #.
5374                                    ;*                ERRTBL - ERRTYP,ERRNBR,and ERRMSG set up correctly.
5375                                    ;*
5376                                    ;* OUTPUTS:       ERRBLK - Value may be destroyed.
5377                                    ;*                BMPCQP - Maybe updated if a BMP code is added to the queue.
5378                                    ;*
5379                                    ;* CALLING SEQUENCE:    JSR     PC,REPCOD
5380                                    ;*
5381                                    ;* COMMENTS:      ERRNBR is restored to its entering value by this routine.
5382                                    ;*                This routine reports errors with numbers ERRNBR thru ERRNBR+1.
5383                                    ;*
5384                                    ;* SUBORDINATE ROUTINES CALLED: ER9001,SAVBMP.
5385                                    ;-- **********************************************************************
5386 024066                            REPCOD:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024066  004537  005326                              JSR     R5,PREG05      ;CALL REGISTER SAVE SUBRT.
5387 024072  012737  014432  005324            MOV     #ER9001,ERRBLK ;SELECT THE ERROR REPORTING ROUTINE.
5388 024100  013703  005320                    MOV     ERRNBR,R3      ;PRESERVE THE ERROR NUMBER.
5389 024104  010204                            MOV     R2,R4          ;EXTRACT THE LINE NUMBER FIELD
5390 024106  000304                            SWAB    R4             ;   FROM THE ERROR CODE WHICH WAS
5391 024110  042704  177760                    BIC     #177760,R4     ;   PASSED INTO THIS ROUTINE.
5392                                    ;+
5393                                    ; Determine the type of code which is to be reported.
5394                                    ;-
5395 024114  012701  011301                    MOV     #EM9003,R1     ;SELECT MODEM STATUS CODE MESSAGE.
5396 024120  032702  000001                    BIT     #BIT0,R2       ;TEST THE MODEM STATUS INDICATION BIT.
5397 024124  001422                            BEQ     4$             ;GOTO REPORT ERROR IF MODEM STATUS CODE.
5398 024126  005237  005320                    INC     ERRNBR         ;SELECT THE SELFTEST CODE ERROR NUMBER.
5399 024132  012701  011323                    MOV     #EM9004,R1     ;SELECT SELFTEST CODE MESSAGE.
5400 024136  012700  000300                    MOV     #300,R0        ;CHECK IF SELF-TEST OR BMP CODE.
5401 024142  040200                            BIC     R2,R0          ;TRY TO CLEAR BMP BITS.
5402 024144  001003                            BNE     2$             ;GO CHECK FOR SELFTEST CODE IF NOT BMP.
5403 024146  004737  024726                    JSR     PC,SAVBMP      ;SAVE THE BMP CODE ON THE QUEUE.
5404 024152  000420                            BR      60$            ;EXIT THIS ROUTINE.
5405 024154  122702  000201            2$:     CMPB    #201,R2        ;CHECK FOR SELF TEST NULL CODE.
5406 024160  001413                            BEQ     6$             ;EXIT ROUTINE IF NULL CODE FOUND.
5407 024162  122702  000203                    CMPB    #203,R2        ;CHECK FOR SKIP SELF TEST CODE.
5408 024166  001410                            BEQ     6$             ;EXIT ROUTINE IF SKIP SELF TEST CODE FOUND.
5409 024170  000400                            BR      4$             ;GO REPORT SELF TEST ERROR.
5410                                    ;+
5411                                    ; Report "UNEXPECTED xxxxx CODE FOUND IN RECEIVE CHAR FIFO."
5412                                    ;-
5413 024172  042702  177400            4$:     BIC     #177400,R2     ;REMOVE UPPER BYTE OF CODE TO BE REPORTED.
5414 024176  004737  026370                    JSR     PC,TXROFF      ;TURN OFF TX AND RX DURING ERROR REPORTING.
5415 024202                                    ERROR                  ;              >>>>> ERROR <<<<<.
     024202  104460                                                            TRAP    C$ERROR
5416 024204  004737  026424                    JSR     PC,TXRON       ;TURN TX AND RX BACK ON.
5417                                    ;+
5418                                    ; Restore the initial error number.
```

```
5419                          ;-
5420 024210  010337  005320   6$:     MOV     R3,ERRNBR
5421
5422 024214                   60$:    PASS                    ;RESTORE GPRS.
     024214  004736                           JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
5423 024216  000207                   RTS     PC
```

```
5425                                       .SBTTL   GLOBAL SUBROUTINE                    - REPSMR -
5426                                       ;++  ********************************************************************************
5427                                       ;*                        - Report Error Summary Routine -
5428                                       ;*       This subroutine reports an error summary for those lines which have
5429                                       ;*       exceeded the number of individual errors to report for a single line
5430                                       ;*       in a single test.  This parameter can be specified by the operator if
5431                                       ;*       he/she answers the Software Parameter Questions.
5432                                       ;*
5433                                       ;* INPUTS:        ERCNTB - Label at base of line error counters table.
5434                                       ;*                ERRMSG - Address of primary error message.
5435                                       ;*                ERRNBR - Error number of errors in this routine.
5436                                       ;*                ERSMRF - "Report error summary for line" flags.
5437                                       ;*
5438                                       ;* OUTPUTS:       ERRBLK - Address of error reporting routine (Destroyed).
5439                                       ;*                Summary messages may be printed at the operator console.
5440                                       ;*
5441                                       ;* CALLING SEQUENCE:   JSR     PC,REPSMR
5442                                       ;*
5443                                       ;* COMMENTS:      If no lines have exceeded the maximum number of individual
5444                                       ;*                errors to report, no messages are printed by this routine.
5445                                       ;*                Error summaries in this routine are reported as errors.
5446                                       ;*                The contents of ERRBLK are destroyed.
5447                                       ;*
5448                                       ;* SUBORDINATE ROUTINES CALLED:
5449                                       ;--  ********************************************************************************
5450
5451 024220                                REPSMR:: SAVE                              ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024220  004537  005326                                  JSR     R5,PREG05        ;CALL REGISTER SAVE SUBRT.
5452 024224  005737  002502                        TST     ERSMRF                    ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
5453 024230  001404                                BEQ     60$                       ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
5454                                       ;+
5455                                       ; We have some error summaries to report.
5456                                       ;-
5457 024232  012737  015052  005324        MOV     #ER9004,ERRBLK   ;SELECT ERROR REPORTING ROUTINE.
5458                                            ;+
5459                                            ; Report
5460                                            ;   "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
5461                                            ;-
5462 024240                                     ERROR
     024240  104460                                                                  TRAP    C$ERROR
5463
5464 024242                                60$:    PASS                              ;RESTORE GPRS.
     024242  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
5465 024244  000207                                RTS     PC
```

```
5467
5468                                      .SBTTL  GLOBAL SUBROUTINE                 - RESETT -
5469                                      ;*********************************************************************
5470                                      ;*                      - Reset Device Under Test -
5471                                      ;*      This subroutine is used to reset the DUT to a known state.
5472                                      ;*      If reset does not successfully complete, ie. time-out occurs, then
5473                                      ;*      an abort test error message is reported.
5474                                      ;*
5475                                      ;* INPUTS:        CSRA - Contains the address of the CSR
5476                                      ;*                TXBFCA - Contains address of DUT DMA Buffer Count register.
5477                                      ;*                ERRTBL- ERRTYP,ERNBR,and ERRMSG set up correctly.
5478                                      ;*
5479                                      ;* OUTPUTS:       The DUT performs its reset function into a known state.
5480                                      ;*                CARRY - Clear indicates the test is to be aborted.
5481                                      ;*                ERRBLK - value may be destroyed.
5482                                      ;*                IESTAT - TX and RX interrupt flags are cleared.
5483                                      ;*                TX and RX interrupt enable bits in the DUT's CSR are cleared.
5484                                      ;*
5485                                      ;* CALLING SEQUENCE:    JSR    PC,RESETT
5486                                      ;*
5487                                      ;* COMMENTS:      This subroutine can report errors with numbers initial ERRNBR
5488                                      ;*                This routine does not destroy the value of ERRNBR.
5489                                      ;*
5490                                      ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET.
5491                                      ;*********************************************************************
5492 024246                               RESETT:: SAVE                         ;SAVE CONTENTS OF GPRS R0 THRU R5.
        024246  004537  005326                     JSR    R5,PREG05      ;CALL REGISTER SAVE SUBRT.
5493 024252  012702  000040                MOV    #BIT05,R2       ;SET BIT MASK OF MASTER RESET BIT.
5494                                      ;+
5495                                      ; Test the state of the master reset bit in the CSR.
5496                                      ;  If MR is set then wait for self-test to complete.
5497                                      ;   If time-out occurs, report the error and pass-out abort test indicator.
5498                                      ;-
5499 024256  013704  002202                MOV    CSRA,R4         ;GET THE ADDRESS OF THE DUT'S CSR.
5500 024262  030214                         BIT    R2,(R4)         ;CHECK STATE OF MASTER RESET BIT.
5501 024264  001406                         BEQ    2$              ;DON'T DELAY IF MR IS ALREADY CLEAR.
5502 024266  005003                         CLR    R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
5503 024270  012701  004704                MOV    #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
5504 024274  004737  021464                JSR    PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
5505 024300  103012                         BCC    4$              ;GO REPORT ERROR IF TIMEOUT OCCURRED.
5506
5507                                      ;+
5508                                      ; Set Master Reset bit in CSR.  Clear TX and RX enable bits, etc.
5509                                      ;  Skip the selftest.
5510                                      ;  Time-out of 2.5 secs. just in case the self-test executes.
5511                                      ;-
5512 024302  010277  155674                2$:   MOV    R2,@CSRA        ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
5513 024306  004737  024774                JSR    PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
5514                                      ;+
5515                                      ;   Set Self-test time-out of 2.5 seconds, and wait for M.R to clear.
5516                                      ;   If Time-out occurs, then report the fatal error and pass-out the abort
5517                                      ;   test indicator.
5518                                      ;-
5519 024312  005003                         CLR    R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
5520 024314  012701  004704                MOV    #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
5521 024320  004737  021464                JSR    PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
5522 024324  103410                         BCS    6$              ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
```

```
 5523                                        ;+
 5524                                        ; Set up error message to report "fatal error found during reset,test aborted".
 5525                                        ; Indicate test is to be aborted by clearing the carry bit.
 5526                                        ;-
 5527 024326  012701  010134        4$:      MOV     #EM1601,R1        ;PASS ERROR MESSAGE TO REPORT.
 5528 024332  012737  014124  005324         MOV     #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
 5529                                         ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
 5530                                         ; "TEST ABORTED"
 5531 024340                                  ERROR                    ;                   >>>>> ERROR <<<<<
      024340  104460                                                                        TRAP     C$ERROR
 5532 024342  000241                          CLC                      ;INDICATE TEST IS TO BE ABORTED.
 5533 024344  000403                          BR      60$              ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
 5534                                        ;+
 5535                                        ; Clear TX and RX Interrupt enable status flags in IESTAT.
 5536                                        ; Exit with continue test indicator set (ie,carry set).
 5537                                        ;-
 5538 024346  005037  002234        6$:      CLR     IESTAT           ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
 5539 024352  000261                          SEC                      ;INDICATE SUCCESS, CONTINUE TEST.
 5540
 5541 024354                        60$:      PASS                     ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
      024354  004736                                    JSR     PC,@(SP)+            ;RETURN TO PREG05 SUBRT.
 5542                                                                   ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
 5543 024356  000207                          RTS     PC
 5544
```

```
5546                             .SBTTL  GLOBAL SUBROUTINE              - RRXNDN -
5547                             ;++ ****************************************************************************
5548                             ;*                       - Report Reception Not Completed Routine -
5549                             ;*          This subroutine checks for lines which did not receive the complete
5550                             ;*          data pattern.  If any are found, they are reported.
5551                             ;*
5552                             ;* INPUTS:          R5 - Local active lines bit map.
5553                             ;*                  DPLENB - Base of table of data pattern lengths.
5554                             ;*                  ERRMSG - Address of primary error message for this routine.
5555                             ;*                  ERRNBR - Error number of error reported in this routine.
5556                             ;*                  RXCNTB - Label at base of the RX character counters table.
5557                             ;*                  RXDONF - Reception done flags.
5558                             ;*
5559                             ;* OUTPUTS:         ERRBLK - Address of the error reporting routine (Destroyed).
5560                             ;*                  Messages may be printed at the operator console.
5561                             ;*
5562                             ;* CALLING SEQUENCE:    JSR    PC,RRXNDN
5563                             ;*
5564                             ;* COMMENTS:        If no lines failed to complete their reception, no messages
5565                             ;*                  are printed.
5566                             ;*
5567                             ;* SUBORDINATE ROUTINES CALLED: ER9005.
5568                             ;-- ****************************************************************************
5569
5570 024360            RRXNDN:: SAVE                         ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024360  004537  005326              JSR       R5,PREG05          ;CALL REGISTER SAVE SUBRT.
5571 024364  010502                      MOV       R5,R2              ;GET COPY OF THE LOCAL ACTIVE LINES BIT MAP.
5572 024366  043702  002506              BIC       RXDONF,R2          ;GET MAP OF ACTIVE LINES WITH RX DONE FLAG CLR.
5573 024372  001410                      BEQ       60$                ;EXIT IF NO ACTIVE LINES HAVE RX DONE FLAG CLR.
5574                             ;+
5575                             ; We have some "RX not completed" errors to report.
5576                             ;-
5577 024374  012737  015154  005324      MOV       #ER9005,ERRBLK     ;SELECT THE ERROR REPORTING ROUTINE.
5578 024402  012701  011767              MOV       #EM9016,R1         ;INDICATE THAT WE ARE DEALING WITH RECEPTION.
5579 024406  012704  003544              MOV       #RXCNTB,R4         ;PASS BASE OF RX CHAR COUNTERS TABLE TO ER9005.
5580                             ;+
5581                             ; Report "SINGLE CHARACTER MODE TEST ERROR:"
5582                             ;     "DATA PATTERN NOT COMPLETELY RECEIVED ON ALL LINES:"
5583                             ;     ...
5584                             ;-
5585 024412                      ERROR
     024412  104460                                                   TRAP     C$ERROR
5586
5587
5588 024414            60$:     PASS                         ;RESTORE GPRS.
     024414  004736                      JSR       PC,@(SP)+                   ;RETURN TO PREG05 SUBRT.
5589 024416  000207             RTS       PC
```

```
5591                                  .SBTTL  GLOBAL SUBROUTINE              - RTXNDN -
5592                                  ;++ ***************************************************************************
5593                                  ;*            - Report Transmission Not Completed Routine -
5594                                  ;*        This subroutine checks for lines which did not transmit the complete
5595                                  ;*        data pattern.  If any are found, they are reported.
5596                                  ;*
5597                                  ;* INPUTS:       R5 - Local active lines bit map.
5598                                  ;*               DPLENB - Label at base of data pattern length table.
5599                                  ;*               ERRMSG - Address of primary error message for this routine.
5600                                  ;*               ERRNBR - Error number of error reported in this routine.
5601                                  ;*               TXCNTB - Label at base of the TX character counters table.
5602                                  ;*               TXDONF - Transmission done flags.
5603                                  ;*
5604                                  ;* OUTPUTS:      ERRBLK - Address of the error reporting routine (Destroyed).
5605                                  ;*               Messages may be printed at the operator console.
5606                                  ;*
5607                                  ;* CALLING SEQUENCE:   JSR     PC,RTXNDN
5608                                  ;*
5609                                  ;* COMMENTS:     If no lines failed to complete their transmission, no messages
5610                                  ;*               are printed.
5611                                  ;*
5612                                  ;* SUBORDINATE ROUTINES CALLED: ER9005.
5613                                  ;-- ***************************************************************************
5614
5615 024420                RTXNDN:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024420  004537  005326                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
5616 024424  010502                        MOV     R5,R2           ;GET COPY OF THE LOCAL ACTIVE LINES BIT MAP.
5617 024426  043702  002504                BIC     TXDONF,R2       ;GET MAP OF ACTIVE LINES WITH TX DONE FLAG CLR.
5618 024432  001410                        BEQ     60$             ;EXIT IF NO ACTIVE LINES HAVE TX DONE FLAG CLR.
5619                                  ;+
5620                                  ; We have some "TX not completed" errors to report.
5621                                  ;-
5622 024434  012737  015154  005324    MOV     #ER9005,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
5623 024442  012701  011753            MOV     #EM9015,R1      ;INDICATE WE ARE DEALING WITH TRANSMISSION.
5624 024446  012704  003504            MOV     #TXCNTB,R4      ;PASS BASE OF TX CHAR COUNTERS TO TABLE ER0805.
5625
5626                                      ;+
5627                                      ; Report "SINGLE CHARACTER MODE TEST ERROR:"
5628                                      ;    "DATA PATTERN NOT COMPLETELY TRANSMITTED ON ALL LINES:"
5629                                      ;       ...
5630                                      ;-
5631 024452                ERROR                   ;                      >>>>> ERROR <<<<<.
     024452  104460                                              TRAP    C$ERROR
5632
5633
5634 024454                60$:    PASS                          ;RESTORE GPRS.
     024454  004736                                JSR     PC,@(SP)+                ;RETURN TO PREG05 SUBRT.
5635 024456  000207                RTS     PC
```

```
5637                                    .SBTTL  GLOBAL SUBROUTINE                - RXDSBL -
5638                                    ;++ ***********************************************************************
5639                                    ;*                         - Disable Receivers -
5640                                    ;*       This subroutine is used to disable reception on selected lines by,
5641                                    ;*       clearing the associated RX_ENABLE bit on the DUT.
5642                                    ;*
5643                                    ;* INPUTS:       R5 - Bit's set correspond to lines on which to clear RX_ENABLE.
5644                                    ;*               CSRA - Contains the address of the DUT CSR.
5645                                    ;*               IESTAT - Contains the state of TXIE and RXIE bits in the CSR.
5646                                    ;*               NUMLNS - Equated to be the maximum number of lines available.
5647                                    ;*               LNCTRA - Contains the address of the LNCTRL register.
5648                                    ;*
5649                                    ;* OUTPUTS:      R5 - Bit's set indicate initial states of all RX_ENBLE bits.
5650                                    ;*               LNCTRA - The state of the RX_ENBLE bit may be altered.
5651                                    ;*               The contents of the IND_ADD_REG field in the CSR are destroyed.
5652                                    ;*
5653                                    ;* CALLING SEQUENCE:     JSR     PC,RXDSBL
5654                                    ;*
5655                                    ;* COMMENTS:
5656                                    ;*
5657                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
5658                                    ;-- ***********************************************************************
5659
5660 024460                            RXDSBL:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024460  004537  005326                             JSR     R5,PREGO5     ;CALL REGISTER SAVE SUBRT.
5661 024464  010500                             MOV     R5,R0              ;COPY BIT MAP OF LINES TO DISABLE RECEPTION.
5662 024466  012701  000001                     MOV     #BIT0,R1           ;INITIALIZE THE SELECTED LINE BIT MASK.
5663 024472  013702  002212                     MOV     LNCTRA,R2          ;GET THE ADDRESS OF THE LNCTRL REGISTER.
5664 024476  012703  000010                     MOV     #NUMLNS,R3         ;GET MAXIMUM LINE NUMBER PLUS ONE.
5665 024502  013704  002234                     MOV     IESTAT,R4          ;GET THE STATES OF THE INT ENABLE BITS.
5666 024506  005005                             CLR     R5                 ;LOG POSSIBLE RX DISABLED ON ALL LINES.
5667                                    ;+
5668                                    ; Select every line in turn, and log the state of each RX_ENABLE bit.
5669                                    ;-
5670 024510  010477  155466            2$:      MOV     R4,@CSRA           ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
5671 024514  032712  000004                     BIT     #BIT2,(R2)         ;CHECK STATE OF RX_ENABLE BIT ON SELECTED LINE.
5672 024520  001401                             BEQ     4$                 ;SKIP NEXT INSTRUCTION IF RX_ENABLE CLEAR.
5673 024522  050105                             BIS     R1,R5              ;LOG RX ENABLE BIT SET FOR SELECTED LINE.
5674                                    ;+
5675                                    ; Clear RX_ENBLE on lines that have a corresponding bit set in the rx disable
5676                                    ; line bit map.
5677                                    ;-
5678 024524  030100                     4$:      BIT     R1,R0              ;CHECK STATE OF DISABLE LINE BIT MAP.
5679 024526  001402                             BEQ     6$                 ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
5680 024530  042712  000004                     BIC     #BIT2,(R2)         ;CLEAR RX_ENABLE BIT ON SELECTED LINE.
5681 024534  005204                     6$:      INC     R4                 ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
5682 024536  006301                             ASL     R1                 ;SHIFT BIT MAP FOR NEXT LINE.
5683 024540  005303                             DEC     R3                 ;DECREMENT LINE NUMBER.
5684 024542  001362                             BNE     2$                 ;LOOP TO CHECK NEXT LINE.
5685
5686 024544                            60$:     PASS    R5                 ;RESTORE GPRS,EXCEPT
     024544  010566  000014                             MOV     R5,R5SLOT(SP)     ;PUT R5 IN STACK SLOT.
     024550  004736                             JSR     PC,@(SP)+         ;RETURN TO PREGO5 SUBRT.
5687                                                                       ;R5 - PREVIOUS STATES OF ALL RX_ENABLE BITS.
5688 024552  000207                             RTS     PC
```

```
5690                                    .SBTTL  GLOBAL SUBROUTINE                    - RXENBL -
5691                                    ;++ ********************************************************************
5692                                    ;*                         - Enable Receiver -
5693                                    ;*      This subroutine is used to enable reception on selected lines by
5694                                    ;*      setting the associated RX.ENABLE bit on the DUT.
5695                                    ;*
5696                                    ;* INPUTS:       R5 - Bit's set correspond to lines on which to set RX.ENABLE.
5697                                    ;*               CSRA - Contains the address of the DUT CSR.
5698                                    ;*               IESTAT - Contains the state of TXIE and RXIE bits in the CSR.
5699                                    ;*               NUMLNS - Equated to be the maximum number of lines available.
5700                                    ;*               LNCTRA - Contains the address of the LNCTRL register.
5701                                    ;*
5702                                    ;* OUTPUTS:      R5 - Bit's set indicate previously disabled lines.
5703                                    ;*               LNCTRA - The state of the RX.ENABLE bit may be altered.
5704                                    ;*               The contents of the IND.ADD.REG field in the CSR are destroyed.
5705                                    ;*
5706                                    ;* CALLING SEQUENCE:    JSR     PC,RXENBL
5707                                    ;*
5708                                    ;* COMMENTS:
5709                                    ;*
5710                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
5711                                    ;-- ********************************************************************
5712
5713 024554                            RXENBL:: SAVE                     ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024554  004537  005326                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
5714 024560  010500                      MOV     R5,R0           ;COPY BIT MAP OF LINES TO ENABLE.
5715 024562  012701  000001              MOV     #BIT0,R1        ;INITIALIZE THE SELECTED LINE BIT MASK.
5716 024566  013702  002212              MOV     LNCTRA,R2       ;GET THE ADDRESS OF THE LNCTRL REGISTER.
5717 024572  012703  000010              MOV     #NUMLNS,R3      ;GET MAXIMUM LINE NUMBER.
5718 024576  013704  002234              MOV     IESTAT,R4       ;GET THE STATES OF THE INT ENABLE BITS.
5719 024602  005005              CLR     R5              ;CLEAR RX.ENABLE BIT LOG OF DISABLED LINES.
5720                                    ;+
5721                                    ; Select every line in turn,and log any RX.ENBLE bit that is clear.
5722                                    ;-
5723 024604  010477  155372     2$:     MOV     R4,@CSRA        ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
5724 024610  032712  000004              BIT     #BIT2,(R2)      ;CHECK STATE OF RX.ENABLE BIT ON SELECTED LINE.
5725 024614  001001                      BNE     4$              ;SKIP NEXT INSTRUCTION IF RX.ENABLE SET.
5726 024616  050105                      BIS     R1,R5           ;LOG RX ENABLE BIT CLEAR FOR SELECTED LINE.
5727                                    ;+
5728                                    ;  Set RX.ENBLE on lines that have a corresponding bit set in the rx enable
5729                                    ;  line bit map.
5730                                    ;-
5731 024620  030100              4$:     BIT     R1,R0           ;CHECK STATE OF RX.ENABLE LINE BIT MAP.
5732 024622  001402                      BEQ     6$              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
5733 024624  052712  000004              BIS     #BIT2,(R2)      ;ENABLE RECEPTIONON SELECTED LINE.
5734 024630  005204              6$:     INC     R4              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
5735 024632  006301                      ASL     R1              ;SHIFT BIT MAP FOR NEXT LINE.
5736 024634  005303                      DEC     R3              ;DECREMENT LINE NUMBER.
5737 024636  001362                      BNE     2$              ;LOOP TO CHECK NEXT LINE.
5738
5739 024640              60$:    PASS    R5              ;RESTORE GPRS,EXCEPT
     024640  010566  000014              MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
     024644  004736              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
5740                                    ;R5 - LINE BIT MAP CORRESPONDING TO THE
5741                                    ; PREVIOUS LINES THAT WERE DISABLED.
5742 024646  000207              RTS     PC
```

```
5744                                    .SBTTL  GLOBAL SUBROUTINE              - RXIEO -
5745                                    ;++ ******************************************************************
5746                                    ;*                    - RECEIVER INTERRUPT DISABLE -
5747                                    ;*      This routine is used to disable receiver interrupts in the DHV11-M.
5748                                    ;*
5749                                    ;* INPUTS:      NONE.
5750                                    ;*
5751                                    ;* OUTPUTS:     The RX.INT.ENBL bit is cleared in the DUT CSR.
5752                                    ;*              IESTST -contains the updated status of the TX and RX interrupt
5753                                    ;*              enable bits.
5754                                    ;*
5755                                    ;* CALLING SEQUENCE:    JSR     PC,RXIEO
5756                                    ;*
5757                                    ;* COMMENTS:    The contents of the indirect address register field in
5758                                    ;*              the DUT CSR are destroyed.
5759                                    ;*
5760                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
5761                                    ;-- ******************************************************************
5762 024650  010046                     RXIEO::  MOV     RO,-(SP)        ;SAVE CONTENTS OF RO ON THE STACK.
5763 024652  106746                              MFPS    -(SP)           ;SAVE PROCESSOR PRIORITY ON STACK.
5764 024654  106427   000340                     MTPS    @PRIO7          ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
5765 024660  042737   137777   002234            BIC     #137777,IESTAT  ;CLEAR RX.INT.ENBL BIT IN IESTAT.
5766 024666  013777   002234   155306            MOV     IESTAT,@CSRA    ;DISABLE RX INTERRUPTS.
5767 024674  106426                              MTPS    (SP)+           ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
5768 024676  012600                              MOV     (SP)+,RO        ;RESTORE RO.
5769 024700  000207                              RTS     PC
```

```
5771                                    .SBTTL  GLOBAL SUBROUTINE                    - RXIE1 -
5772                                    ;++ ***********************************************************************
5773                                    ;*                     - RECEIVER INTERRUPT ENABLE -
5774                                    ;*     This routine is used to enable receiver interrupts in the DHV11-M.
5775                                    ;*
5776                                    ;* INPUTS:      NONE.
5777                                    ;*
5778                                    ;* OUTPUTS:     The RX.INT.ENBL bit is set in the DUT CSR.
5779                                    ;*              IESTST -contains the updated status of the TX and RX interrupt
5780                                    ;*              enable bits.
5781                                    ;*
5782                                    ;* CALLING SEQUENCE:    JSR     PC,RXIE1
5783                                    ;*
5784                                    ;* COMMENTS:    The contents of the indirect address register field in
5785                                    ;*              the DUT CSR are destroyed.
5786                                    ;*
5787                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
5788                                    ;-- ***********************************************************************
5789
5790 024702  052737  000100  002234    RXIE1::  BIS     #BIT06,IESTAT    ;SET RX.INT.ENBL BIT IN IESTAT.
5791 024710  042737  137677  002234             BIC     #137677,IESTAT   ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
5792 024716  013777  002234  155256             MOV     IESTAT,@CSRA     ;ENABLE RX INTERRUPTS.
5793 024724  000207                             RTS     PC
```

```
5795                             .SBTTL  GLOBAL SUBROUTINE               - SAVBMP -
5796                             ;++ ****************************************************************************
5797                             ;*                      - Save BMP codes Routine -
5798                             ;*      This routine saves the parameter passed in, onto the BMP code queue
5799                             ;*      together with the number of the currently executing test.
5800                             ;*
5801                             ;* INPUTS:        R2 - Contains the BMP code that is to be placed on the queue.
5802                             ;*                BMPCQP - Contains address of next location in the bmp queue.
5803                             ;*                BMPCQB - Label at base of the BMP code queue.
5804                             ;*                BMPCQE - Label of next location after the end of the BMP queue.
5805                             ;*                TSTNUM - Contains the number of the current test.
5806                             ;*
5807                             ;* OUTPUTS:       BMPCQP -  Incremented by 4.
5808                             ;*                The contents of the BMP code queue are updated.
5809                             ;*
5810                             ;* CALLING SEQUENCE:    JSR     PC,SAVBMP
5811                             ;*
5812                             ;* COMMENTS:      If the overflow occurs then the last location will be
5813                             ;*                overwritten by any subsequent attempts to update the queue.
5814                             ;*
5815                             ;* SUBORDINATE ROUTINES CALLED: None.
5816                             ;-- ****************************************************************************
5817
5818 024726                     SAVBMP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024726  004537  005326                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
5819 024732  013704  002512              MOV     BMPCQP,R4       ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
5820 024736  113724  002224              MOVB    TSTNUM,(R4)+    ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
5821 024742  005204                      INC     R4              ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
5822 024744  042702  177400              BIC     #177400,R2      ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
5823 024750  010224                      MOV     R2,(R4)+        ;SAVE THE BMP CODE ON THE QUEUE.
5824 024752  020427  002714              CMP     R4,#BMPCQE      ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
5825 024756  103402                      BLO     2$              ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
5826 024760  162704  000004              SUB     #4,R4           ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
5827 024764  010437  002512      2$:     MOV     R4,BMPCQP       ;SAVE THE POINTER.
5828
5829 024770                      60$:    PASS                    ;RESTORE GPRS.
     024770  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
5830 024772  000207              RTS     PC
```

```
5832                              .SBTTL  GLOBAL SUBROUTINE              - SKPSTS -
5833                              ;++ *************************************************************************
5834                              ;*                         - Skip Selftest Routine -
5835                              ;*        This subroutine is used to skip the selftest after a DUT reset has been
5836                              ;*        initiated.  It must be entered immediately after setting the DUT Master
5837                              ;*        Reset routine or after the execution of a bus reset (because of timing
5838                              ;*        considerations).
5839                              ;*
5840                              ;* INPUTS:      CSRA - Contains address of the DUT CSR.
5841                              ;*              TXBFCA - Contains address of DUT DMA Buffer Count register.
5842                              ;*
5843                              ;* OUTPUTS:     Skip selftest codes are written to the DUT registers.
5844                              ;*
5845                              ;* CALLING SEQUENCE:    JSR     PC,SKPSTS
5846                              ;*
5847                              ;* COMMENTS:
5848                              ;*
5849                              ;* SUBORDINATE ROUTINES CALLED: DELAY.
5850                              ;-- *************************************************************************
5851
5852 024774                      SKPSTS:: SAVE                       ;SAVE CONTENTS OF GPRS R0 THRU R5.
     024774  004537  005326               JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
5853 025000  012704  000012               MOV     #10.,R4         ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
5854 025004  004737  017354               JSR     PC,DELAY        ;DELAY FOR 10 MILLI-SECONDS.
5855                              ;+
5856                              ; Write skip self-test code (52525) to all the indexed DUT Registers.
5857                              ;-
5858 025010  012701  000050               MOV     #NUMLNS!BITO5,R1  ;FORM IND.ADR.REG FIELD (PLUS M.R. BIT) WORD.
5859                              ;THE ABOVE INCLUSION OF THE M.R. BIT IS NECESSARY BECAUSE OF THE
5860                              ;  LACK OF A M.R. BIT WRITE LOCK-OUT ON THE DHV11-M.
5861 025014  012703  052525               MOV     #52525,R3       ;INITIALISE THE SKIP SELF-TEST CODE.
5862 025020  005301            4$:        DEC     R1              ;SELECT THE NEXT SET OF DEVICE REGISTERS.
5863 025022  013704  002202               MOV     CSRA,R4         ;GET THE ADDRESS OF THE CSR OF THE DUT.
5864 025026  010124                       MOV     R1,(R4)+        ;SELECT A BANK OF DUT REGISTERS.
5865 025030  010324            6$:        MOV     R3,(R4)+        ;WRITE THE CODE TO A DUT REGISTER.
5866 025032  020437  002220               CMP     R4,TXBFCA       ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
5867 025036  103774                       BLO     6$              ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
5868 025040  032701  000017               BIT     #17,R1          ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
5869 025044  001365                       BNE     4$              ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.
5870
5871 025046                      60$:     PASS                    ;RESTORE GPRS.
     025046  004736                       JSR     PC,@(SP)+                       ;RETURN TO PREGO5 SUBRT.
5872 025050  000207                       RTS     PC
```

```
5874                                    .SBTTL  GLOBAL SUBROUTINE              - SPLSUP -
5875                                    ;++ **********************************************************************
5876                                    ;*                  - SPLIT SPEED TRANSMISSION/RECEPTION SET-UP -
5877                                    ;*
5878                                    ;*              This routine is used to initialise both the DUT and the
5879                                    ;*              transmission/reception control parameters to the correct
5880                                    ;*              state, prior to split speed transmission/reception.
5881                                    ;*
5882                                    ;* INPUTS:      RO - TX,RX LPR contents for lines in group II.
5883                                    ;*              R1 - TX,RX LPR contents for lines in group I.
5884                                    ;*              R2 - Start address of data pattern to transmit.
5885                                    ;*              R3 - Number of time data pattern to be TX on lines in LINGRP1.
5886                                    ;*              R4 - Number of time data pattern to be TX on lines in LINGRP2.
5887                                    ;*              ACTLNS - Contains a bit map of all currently active lines.
5888                                    ;*              LGRP1M - Contains the bit map of line group I lines.
5889                                    ;*              LOPBCK - Contains the type of loopback mode selected.
5890                                    ;*              CBB - Label at base of TX/RX control block.
5891                                    ;*
5892                                    ;* OUTPUTS:     The contents of the Control Block are destroyed.
5893                                    ;*              The indirect address field of the DUT CSR may be destroyed.
5894                                    ;*              The DUT's LPR's and LNC's may be modified.
5895                                    ;*              The following pointers and counters are initialised;
5896                                    ;*              CHCNT,CHRTOT,DPEND,DPLEN,EXCNT,RXCNT,RXDONF,RXPTR,TXCNT,
5897                                    ;*              TXDONF,TXPTR,TXRXL.
5898                                    ;*
5899                                    ;* CALLING SEQUENCE:    JSR     PC,SPLSUP
5900                                    ;*
5901                                    ;* COMMENTS:    This routine should be called twice during the testing of
5902                                    ;*              the split speed capabilities of the DUT.
5903                                    ;*              So that both line groups are tested on transmission and
5904                                    ;*              reception.
5905                                    ;*              eg.     R1 - LPR contents for lines in LGRP2M,TX=Y,RX=Z baud.
5906                                    ;*                      R2 - LPR contents for lines in LGRP1M,TX=Z,RX=Y baud.
5907                                    ;*                      R3 - Repeat TX on lines in line group 1 = X times.
5908                                    ;*                      R4 - Repeat TX on lines in line group 2 = W times.
5909                                    ;*              JSR   PC,SPLSUP           ; do set-up.
5910                                    ;*              Execute test for the above set-up.
5911                                    ;*              Swap the contents of R1 and R2.
5912                                    ;*              Swap the contents of R3 and R4.
5913                                    ;*                      R1 - LPR contents for lines in LGRP2M,TX=Z,RX=Y baud.
5914                                    ;*                      R2 _ LPR contents for lines in LGRP1M,TX=Y,RX=Z baud.
5915                                    ;*                      R3 - Repeat TX on lines in line group 1 = W times.
5916                                    ;*                      R4 - Repeat TX on lines in line group 2 = X times.
5917                                    ;*              JSR   PC,SPLSUP           ;do set up again.
5918                                    ;*              Execute test again.
5919                                    ;*
5920                                    ;* SUBORDINATE ROUTINES CALLED: CONMAP,RXDSBL,RXENBL,TXRINI.
5921                                    ;-- **********************************************************************
5922
5923 025052                            SPLSUP:: SAVE                           ;SAVE CONTENTS OF THE GPR'S RO THRU R5.
     025052  004537  005326                             JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
5924 025056  010037  025346                    MOV     RO,70$                  ;SAVE LPR PARAMETER FOR LINE GRP2.
5925 025062  010137  025350                    MOV     R1,72$                  ;SAVE LPR PARAMETER FOR LINE GRP1.
5926 025066  005037  002504                    CLR     TXDONF                  ;CLEAR THE TX DONE FLAGS FOR ALL LINES.
5927 025072  005037  002506                    CLR     RXDONF                  ;CLEAR THE RX DONE FLAGS FOR ALL LINES.
5928                                    ;+
5929                                    ; Set up the Transmission/Reception Control block to initialise the lines
```

```
5930                                              ; in group II.
5931                                              ;-
5932  025076  010037  003124              MOV    R0,CBB           ;SET CONTENTS OF LPR PARAMS IN TX/RX C.BLK.
5933  025102  012700  003126              MOV    #CBB+2,R0        ;GET BASE ADDRESS OF CONTROL BLOCK.
5934  025106  012720  000004              MOV    #4,(R0)+         ;LNCTRL PARAMETER, ENABLE RECEIVERS.
5935  025112  010220                       MOV    R2,(R0)+         ;START ADDRESS OF DATA PATTERN.
5936  025114  012720  000020              MOV    #16.,(R0)+       ;DATA PATTERN LENGTH SET TO 16.
5937  025120  010420                       MOV    R4,(R0)+         ;NUMBER OF DATA PATTNS TO TRANSMIT ON LINGRP2.
5938  025122  013710  002174              MOV    ACTLNS,(R0)      ;BIT MAP OF LINES TO INITIALISE.
5939  025126  043720  002230              BIC    LGRP1M,(R0)+     ;CLEAR THE UNWANTED LINES FROM BIT MAP.
5940  025132  113720  002176              MOVB   LOPBCK,(R0)+     ;SET LOOPBACK MODE.
5941  025136  005200                       INC    R0               ;INCREMENT ADDRESS TO ACCESS NEXT WORD.
5942  025140  012710  000002              MOV    #2,(R0)          ;SET OFFSET FOR EACH TRANSMISSION START TO 2.
5943                                              ;+
5944                                              ; Initialise the DUT and the associated pointers and counters, to the state
5945                                              ; dictated by the contents of the TX/RX control block.
5946                                              ;-
5947  025144  004737  026114              JSR    PC,TXRINI        ;INITIALISE DUT.
5948                                              ;+
5949                                              ; Set up Control block for lines in group I.
5950                                              ;-
5951  025150  012700  003124              MOV    #CBB,R0          ;GET START ADDRESS OF CONTROL BLOCK.
5952  025154  010120                       MOV    R1,(R0)+         ;SET LPR PARAMETER FOR LINES TO RECEIVE DATA.
5953  025156  012720  000004              MOV    #4,(R0)+         ;LNCTRL PARAMETER, ENABLE RECEIVERS.
5954  025162  010220                       MOV    R2,(R0)+         ;START ADDRESS OF DATA PATTERN.
5955  025164  012720  000020              MOV    #16.,(R0)+       ;DATA PATTERN LENGTH SET TO 16.
5956  025170  010320                       MOV    R3,(R0)+         ;NUMBER OF DATA PATTNS TO TRANSMIT ON LINGRP1.
5957  025172  013710  002174              MOV    ACTLNS,(R0)      ;BIT MAP OF LINES TO INITIALISE.
5958  025176  043720  002232              BIC    LGRP2M,(R0)+     ;CLEAR THE UNWANTED LINES FROM BIT MAP.
5959  025202  113720  002176              MOVB   LOPBCK,(R0)+     ;SET LOOPBACK MODE.
5960  025206  005200                       INC    R0               ;INCREMENT ADDRESS TO ACCESS NEXT WORD.
5961  025210  012710  000002              MOV    #2,(R0)          ;SET OFFSET FOR EACH TRANSMISSION START TO 2.
5962                                              ;+
5963                                              ; Initialise the DUT and the associated pointers and counters, to the state
5964                                              ; dictated by the contents of the TX/RX control block.
5965                                              ;-
5966  025214  004737  026114              JSR    PC,TXRINI        ;INITIALISE DUT.
5967
5968                                              ;+
5969                                              ; Set-up the required LPR parameters needed for the correct reception of data
5970                                              ; on associated in-active lines.
5971                                              ;-
5972                                              ;+
5973                                              ; Initialise LPR parameters for line group 1.
5974                                              ;-
5975  025220  012701  000377              MOV    #MAPLNS,R1       ;SET UP BIT MAP CORRESPONDING TO ALL LINES.
5976  025224  013702  002174              MOV    ACTLNS,R2        ;GET THE ACTIVE (TX) LINE BIT MAP.
5977  025230  005101                       COM    R1               ;GENERATE A BIT MAP OF NONE EXISTANT LINES.
5978  025232  005102                       COM    R2               ;GENERATE A BIT MAP OF INACTIVE LINES.
5979  025234  040102                       BIC    R1,R2            ;CLEAR ANY "NONE EXISTANT" INACTIVE LINES.
5980  025236  043702  002232              BIC    LGRP2M,R2        ;ONLY PASS LGRP1 ASSOCIATED LINE BIT MAP.
5981  025242  010237  003136              MOV    R2,CBMAPA        ;SET UP BIT MAP IN CONTROL BLOCK.
5982  025246  005037  003134              CLR    CBDPNA           ;CLEAR REPEAT TX COUNT IN CONTROL BLOCK.
5983  025252  013737  025350  003124      MOV    72$,CBLPRA       ;SET-UP COMPLEMENTARY LPR PARM FOR LGRP2.
5984  025260  004737  026114              JSR    PC,TXRINI        ;INITIALISE INACTIVE LINES IN LGRP2.
5985                                              ;+
5986                                              ; Initialise LPR parameters for line group 2.
```

```
 5987                                    ;-
 5988 025264  013702  002174                    MOV     ACTLNS,R2        ;GET THE ACTIVE (TX) LINE BIT MAP.
 5989 025270  005102                             COM     R2               ;GENERATE A BIT MAP OF INACTIVE LINES.
 5990 025272  040102                             BIC     R1,R2            ;CLEAR ANY NONE EXISTANT INACTIVE LINES.
 5991 025274  043702  002230                     BIC     LGRP1M,R2        ;ONLY PASS LGRP2 ASSOCIATED LINE BIT MAP.
 5992 025300  010237  003136                     MOV     R2,CBMAPA        ;SET-UP BIT MAP IN CONTROL BLOCK.
 5993 025304  013737  025346  003124             MOV     70$,CBLPRA       ;SET-UP COMPLAMENTARY LPR PARAM FOR LGRP1.
 5994 025312  004737  026114                     JSR     PC,TXRINI        ;INITIALISE INACTIVE LINES IN LGRP1.
 5995                                    ;+
 5996                                    ; Disable Receivers on all lines to ensure that only the receivers of the
 5997                                    ; associated active (TX) lines are enabled.(staggered loopback)
 5998                                    ;  Re-enable reception on the correct associated lines.
 5999                                    ;-
 6000 025316  012705  000377                     MOV     #MAPLNS,R5       ;SET-UP BIT MAP FOR ALL LINES.
 6001 025322  004737  024460                     JSR     PC,RXDSBL        ;DISABLE RX ON ALL LINES.
 6002                                    ;+
 6003                                    ; Enable receivers on associated (RX) lines.
 6004                                    ;-
 6005 025326  013705  002174                     MOV     ACTLNS,R5        ;GET ACTIVE (TX) LINE BIT MAP.
 6006 025332  004737  017300                     JSR     PC,CONMAF        ;GENERATE AN ASSOCIATED (RX) LINE BIT MAP.
 6007 025336  004737  024554                     JSR     PC,RXENBL        ;ENABLE RECEIVERS ON ASSOCIATED LINES.
 6008
 6009 025342                          60$:       PASS                     ;RESTORE GRP'S.
      025342  004736                             JSR     PC,@(SP)+                 ;RETURN TO PREGO5 SUBRT.
 6010 025344  000207                             RTS     PC
 6011 025346  000000                  70$:       .WORD   0                ;LOCAL STORAGE OF LPR PARAMETER LGRP2.
 6012 025350  000000                  72$:       .WORD   0                ;LOCAL STORAGE OF LPR PARAMETER LGRP1.
```

```
6014                                    .SBTTL  GLOBAL SUBROUTINE          - STPSW -
6015                                    ;++ ********************************************************************
6016                                    ;*                         - SET PROCESSOR STATUS WORD -
6017                                    ;*              THIS ROUTINE SETS THE PSW TO THE CONTENTS OF R1.
6018                                    ;*
6019                                    ;*      INPUTS:           R1 - CONTAINS THE NEW PSW SETTINGS
6020                                    ;*
6021                                    ;*      OUTPUTS:          PSW - SET TO THE CONTENTS OF R1
6022                                    ;*
6023                                    ;*      CALLING SEQUENCE:     JSR     PC,STPSW
6024                                    ;*
6025                                    ;*      COMMENTS:         USED IN THE DMA ADDRESS TEST TO SET THE PROCESSOR
6026                                    ;*                        PRIORITY WITHOUT MAKING A CALL TO THE DRS.
6027                                    ;*
6028                                    ;*      SUBROUTINES CALLED:    NONE.
6029                                    ;********************************************************************
6030
6031 025352                            STPSW:: SAVE
     025352  004537  005326                     JSR     R5,PREG05        ;CALL REGISTER SAVE SUBRT.
6032 025356  010146                             MOV     R1,-(SP)         ;PUSH THE NEW PSW CONTENTS ONTO THE STACK
6033 025360  012746  025366                     MOV     @ADDR,-(SP)      ;PUSH THE NEW PC VALUE ONTO THE STACK
6034 025364  000002                             RTI                      ;LOAD THE NEW PC AND PSW
6035 025366                            ADDR:   PASS                     ;
     025366  004736                             JSR     PC,@(SP)+                ;RETURN TO PREG05 SUBRT.
6036 025370  000207                             RTS     PC               ;RETURN
6037
6038
6039
6040
```

```
6042
6043                                    .SBTTL   GLOBAL SUBROUTINE                 - SWAPO -
6044                             ;++ ********************************************************************
6045                             ;*                    - Swap GPRs With GPR Set 0 Routine -
6046                             ;*       This subroutine swaps the present contents of GPRs R1 thru R5 with
6047                             ;*       the contents of the number zero GPR save area.  The contents of R0
6048                             ;*       are not altered by this subroutine.
6049                             ;*
6050                             ;* INPUTS:        GPR contents R1 thru R5.
6051                             ;*                GPRSOB - Label at base of GPR save area number zero.
6052                             ;*
6053                             ;* OUTPUTS:       R1 thru R5 contain the previous contents of GPR save area
6054                             ;*                       zero words 1 thru 5 respectively.
6055                             ;*                GPRSO - GPR save area 0 words 1 thru 5, contain previous
6056                             ;*                       contents of GPRs R1 thru R5 respectively.
6057
6058                             ;* CALLING SEQUENCE:   JSR     PC,SWAPO
6059                             ;*
6060                             ;* COMMENTS:      The state of the CARRY flag is not alterred by this routine.
6061                             ;*
6062                             ;* SUBORDINATE ROUTINES CALLED: None.
6063                             ;-- ********************************************************************
6064
6065 025372  010046             SWAPO::  MOV     R0,-(SP)          ;SAVE THE CONTENTS OF R0.
6066                             ;+
6067                             ; Load the stack from the GPRs.
6068                             ;-
6069 025374  010146                      MOV     R1,-(SP)          ;SAVE THE CONTENTS OF R1.
6070 025376  010246                      MOV     R2,-(SP)          ;SAVE THE CONTENTS OF R2.
6071 025400  010346                      MOV     R3,-(SP)          ;SAVE THE CONTENTS OF R3.
6072 025402  010446                      MOV     R4,-(SP)          ;SAVE THE CONTENTS OF R4.
6073 025404  010546                      MOV     R5,-(SP)          ;SAVE THE CONTENTS OF R5.
6074
6075                             ; Load the GPRs from the GPR save area 0.
6076                             ;-
6077 025406  012700  002466             MOV     @GPRSOB,R0        ;GET THE BASE ADDRESS OF GPR SAVE AREA 0.
6078 025412  012001                      MOV     (R0)+,R1          ;LOAD R1 WITH GPR SAVE AREA 0 WORD 1.
6079 025414  012002                      MOV     (R0)+,R2          ;LOAD R1 WITH GPR SAVE AREA 0 WORD 2.
6080 025416  012003                      MOV     (R0)+,R3          ;LOAD R1 WITH GPR SAVE AREA 0 WORD 3.
6081 025420  012004                      MOV     (R0)+,R4          ;LOAD R1 WITH GPR SAVE AREA 0 WORD 4.
6082 025422  012005                      MOV     (R0)+,R5          ;LOAD R1 WITH GPR SAVE AREA 0 WORD 5.
6083                             ;+
6084                             ; Load the GPR save area 0 from the stack.
6085                             ;-
6086 025424  012640                      MOV     (SP)+,-(R0)       ;LOAD GPR SAVE AREA 0 WORD 5 WITH SAVED R5.
6087 025426  012640                      MOV     (SP)+,-(R0)       ;LOAD GPR SAVE AREA 0 WORD 4 WITH SAVED R4.
6088 025430  012640                      MOV     (SP)+,-(R0)       ;LOAD GPR SAVE AREA 0 WORD 3 WITH SAVED R3.
6089 025432  012640                      MOV     (SP)+,-(R0)       ;LOAD GPR SAVE AREA 0 WORD 2 WITH SAVED R2.
6090 025434  012640                      MOV     (SP)+,-(R0)       ;LOAD GPR SAVE AREA 0 WORD 1 WITH SAVED R1.
6091
6092 025436  012600                      MOV     (SP)+,R0          ;RESTORE THE INITIAL VALUE OF R0.
6093
6094 025440  000207                      RTS     PC
```

```
6096                                    .SBTTL  GLOBAL SUBROUTINE              - TXDONE -
6097                                    ;+* *******************************************************************
6098                                    ;*                      - TRANSMISSION DONE -
6099                                    ;*      This subroutine is used in the transmission/reception tests to allow
6100                                    ;*      time for transmission to complete on outstanding lines.
6101                                    ;*
6102                                    ;* INPUTS:       ACTLNS - Contains bit map of all active lines.
6103                                    ;*               TXDONF - TX done flags, set for lines that have sent all chars.
6104                                    ;*               CHCNT - Table contaning the number of chars to be TX'd.
6105                                    ;*
6106                                    ;*
6107                                    ;* OUTPUTS:      Transmission interrupts are disabled.
6108                                    ;*
6109                                    ;* CALLING SEQUENCE:    JSR    PC,TXDONE
6110                                    ;*
6111                                    ;* COMMENTS:
6112                                    ;*
6113                                    ;* SUBORDINATE ROUTINES CALLED: MSLOOP,MUL16U.
6114                                    ;-- *******************************************************************
6115
6116 025442             TXDONE:: SAVE                          ;SAVE CONTENTS OF GPRS R0 THRU R5.
     025442  004537  005326                    JSR     R5,PREG05        ;CALL REGISTER SAVE SUBRT.
6117                                    ;+
6118                                    ; Check if all active lines have completed transmission.
6119                                    ;  If any have not yet completed, determine the tx char count for a
6120                                    ;  line that has outstanding characters to transmit.  Using this value,
6121                                    ;  calculate the time-out value needed at the currently selected baud rate.
6122                                    ;-
6123 025446  013703  002174                    MOV     ACTLNS,R3        ;GET THE ACTIVE LINE BIT MAP.
6124 025452  013702  002504                    MOV     TXDONF,R2        ;GET THE BIT MAP OF LINES THAT HAVE COMPLETED.
6125 025456  040203                            BIC     R2,R3            ;GENERATE A BIT MAP OF LINES THAT ARE STILL TX.
6126 025460  005703                            TST     R3               ;CHECK IF ALL LINES HAVE COMPLETED TX.
6127 025462  001427                            BEQ     6$               ;GO DISABLE TX INTERRUPTS IF ALL DONE.
6128                                    ;+
6129                                    ; Find a line that has not completed transmission.
6130                                    ; Obtain the expected character count for that line (which is the same for
6131                                    ; all other lines with outstanding tx's).
6132                                    ; Calculate time-out value.
6133                                    ;-
6134 025464  005004                            CLR     R4               ;CLEAR LINE NUMBER COUNTER.
6135 025466  012702  000001                    MOV     #1,R2            ;SELECT BIT MAP FOR THE FIRST LINE.
6136 025472  030203             2$:            BIT     R2,R3            ;SEE IF THIS LINE HAS COMPLETED.
6137 025474  001003                            BNE     4$               ;BRANCH IF THIS LINE HAS NOT COMPLETED TX.
6138 025476  006102                            ROL     R2               ;SHIFT THE LINE BIT MAP FOR THE NEXT LINE.
6139 025500  005204                            INC     R4               ;INCREMENT THE LINE NUMBER COUNTER.
6140 025502  000773                            BR      2$               ;LOOP TO CHECK THE NEXT LINE.
6141 025504  006304             4$:            ASL     R4               ;LINE NUMBER X 2 TO OBTAIN OFFSET INTO TABLE.
6142 025506  016401  003444                    MOV     CHCNTB(R4),R1    ;GET THE EXPECTED NUMBER OF CHARS TO BE TX'D.
6143 025512  013702  002242                    MOV     RXTOUT,R2        ;GET THE CURRENT TIME-OUT VALUE FOR ONE CHAR.
6144 025516  004737  022030                    JSR     PC,MUL16U        ;(NUMBER OF CHARS TO TX) X (TIME-OUT OF 1 CHAR)
6145 025522  006301                            ASL     R1               ;MULTIPLY DELAY TIME BY 2 TO GIVE A SAFE VALUE.
6146                                    ;+
6147                                    ; Wait for all oustanding transmissions to complete or time-out.
6148                                    ; Disable all transmission interrupts.
6149                                    ;-
6150 025524  013702  002174                    MOV     ACTLNS,R2        ;PASS A BIT MAP OF THE BITS TO TEST.
6151 025530  010203                            MOV     R2,R3            ;PASS THE EXPECTED STATE OF THE TXDONF.
```

```
6152 025532 012704 002504                    MOV     @TXDONF,R4          ;PASS THE ADDRESS OF THE WORD TO TEST.
6153 025536 004737 021600                    JSR     PC,MSLOOP           ;WAIT FOR TIME-OUT OF TX COMPLETION.
6154 025542 004737 026036           6$:      JSR     PC,TXIEO            ;DISABLE ALL TX INTERRUPTS.
6155
6156 025546                         60$:     PASS                        ;RESTORE GPRS.
     025546 004736                                             JSR     PC,@(SP)+                    ;RETURN TO PREGO5 SUBRT.
6157 025550 000207                           RTS     PC
```

```
6159
6160                                  .SBTTL   GLOBAL SUBROUTINE                    - TXDSBL -
6161                                  ;++ ***********************************************************************
6162                                  ;*                           - Transmitter Disable -
6163                                  ;*        This subroutine is used to disable transmission on selected lines by,
6164                                  ;*        clearing the associated TX.ENABLE bit on the DUT.
6165                                  ;*
6166                                  ;* INPUTS:        R5 - Bit's set correspond to lines on which to clear TX.ENABLE.
6167                                  ;*                CSRA - Contains the address of the DUT CSR.
6168                                  ;*                IESTAT - Contains the state of TXIE and RXIE bits in the CSR.
6169                                  ;*                NUMLNS - Equated to be the maximum number of lines available.
6170                                  ;*                TXAD2A - Contains the address of the TBUFFAD2 register.
6171                                  ;*
6172                                  ;* OUTPUTS:       R5 - Bit's set indicate the initial states of all TX.ENBLE bits.
6173                                  ;*                TBUFFAD2 - The state of the TX.ENBLE bit may be altered.
6174                                  ;*                The contents of the IND.ADD.REG field in the CSR are destroyed.
6175                                  ;*
6176                                  ;* CALLING SEQUENCE:    JSR    PC,TXDSBL
6177                                  ;*
6178                                  ;* COMMENTS:
6179                                  ;*
6180                                  ;* SUBORDINATE ROUTINES CALLED: NONE.
6181                                  ;-- ***********************************************************************
6182
6183 025552                          TXDSBL:: SAVE                      ;SAVE CONTENTS OF GPRS R0 THRU R5.
     025552  004537  005326                   JSR    R5,PREGO5          ;CALL REGISTER SAVE SUBRT.
6184 025556  010500                           MOV    R5,R0              ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
6185 025560  012701  000001                   MOV    #BIT0,R1           ;INITIALIZE THE SELECTED LINE BIT MASK.
6186 025564  013702  002216                   MOV    TXAD2A,R2          ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
6187 025570  005202                           INC    R2                 ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
6188 025572  012703  000010                   MOV    #NUMLNS,R3         ;GET MAXIMUM LINE NUMBER PLUS ONE.
6189 025576  013704  002234                   MOV    IESTAT,R4          ;GET THE STATES OF THE INT ENABLE BITS.
6190 025602  005005                           CLR    R5                 ;LOG POSSIBLE TX DISABLED ON ALL LINES.
6191
6192                                  ;+
6193                                  ; Select every line in turn, and log the state of each TX.ENABLE bit.
                                      ;-
6194 025604  010477  154372          2$:      MOV    R4,@CSRA           ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
6195 025610  105712                           TSTB   (R2)               ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
6196 025612  100001                           BPL    4$                 ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
6197 025614  050105                           BIS    R1,R5              ;LOG TX ENABLE BIT SET FOR SELECTED LINE.
6198                                  ;+
6199                                  ;  Clear TX.ENBLE on lines that have a corresponding bit set in the tx disable
6200                                  ;  line bit map.
6201                                  ;-
6202 025616  030100                  4$:      BIT    R1,R0              ;CHECK STATE OF DISABLE LINE BIT MAP.
6203 025620  001402                           BEQ    6$                 ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
6204 025622  142712  000200                   BICB   #BIT7,(R2)         ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
6205 025626  005204                  6$:      INC    R4                 ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
6206 025630  006301                           ASL    R1                 ;SHIFT BIT MAP FOR NEXT LINE.
6207 025632  005303                           DEC    R3                 ;DECREMENT LINE NUMBER.
6208 025634  001363                           BNE    2$                 ;LOOP TO CHECK NEXT LINE.
6209
6210 025636                          60$:     PASS   R5                 ;RESTORE GPRS,EXCEPT
     025636  010566  000014                   MOV    R5,R5SLOT(SP)      ;PUT R5 IN STACK SLOT.
     025642  004736                           JSR    PC,@(SP)+          ;RETURN TO PREGO5 SUBRT.
6211                                                                    ;R5 - PREVIOUS STATES OF ALL TX.ENABLE BITS.
6212 025644  000207                           RTS    PC
```

```
6214                                    .SBTTL   GLOBAL SUBROUTINE              - TXENBL -
6215                                    ;++ *************************************************************************
6216                                    ;*                         - Transmitter Enable -
6217                                    ;*        This subroutine is used to enable transmission on selected lines by
6218                                    ;*        setting the associated TX.ENABLE bit on the DUT.
6219                                    ;*
6220                                    ;* INPUTS:        R5 - Bit's set correspond to lines on which to set TX.ENABLE.
6221                                    ;*                CSRA - Contains the address of the DUT CSR.
6222                                    ;*                IESTAT - Contains the state of TXIE and RXIE bits in the CSR.
6223                                    ;*                NUMLNS - Equated to be the maximum number of lines available.
6224                                    ;*                TXAD2A - Contains the address of the TBUFFAD2 register.
6225                                    ;*
6226                                    ;* OUTPUTS:       R5 - Bit's set indicate previously disabled lines.
6227                                    ;*                TBUFFAD2 - The state of the TX.ENBLE bit may be altered.
6228                                    ;*                The contents of the IND.ADD.REG field in the CSR are destroyed.
6229                                    ;*
6230                                    ;* CALLING SEQUENCE:    JSR      PC,TXENBL
6231                                    ;*
6232                                    ;* COMMENTS:
6233                                    ;*
6234                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
6235                                    ;-- *************************************************************************
6236
6237 025646                            TXENBL:: SAVE                        ;SAVE CONTENTS OF GPRS RO THRU R5.
     025646   004537   005326                   JSR       R5,PREGO5        ;CALL REGISTER SAVE SUBRT.
6238 025652   010500                   MOV       R5,R0                     ;COPY BIT MAP OF LINES TO ENABLE.
6239 025654   012701   000001          MOV       #BITO,R1                  ;INITIALIZE THE SELECTED LINE BIT MASK.
6240 025660   013702   002216          MOV       TXAD2A,R2                 ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
6241 025664   005202                   INC       R2                        ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
6242 025666   012703   000010          MOV       #NUMLNS,R3                ;GET MAXIMUM LINE NUMBER.
6243 025672   013704   002234          MOV       IESTAT,R4                 ;GET THE STATES OF THE INT ENABLE BITS.
6244 025676   005005                   CLR       R5                        ;CLEAR TX.ENABLE BIT LOG OF DISABLED LINES.
6245
6246                                    ;+
6247                                    ; Select every line in turn,and log any TX.ENBLE bit that is clear.
                                        ;-
6248 025700   010477   154276          2$:      MOV       R4,@CSRA         ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
6249 025704   105712                   TSTB      (R2)                      ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
6250 025706   100401                   BMI       4$                        ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
6251 025710   050105                   BIS       R1,R5                     ;LOG TX ENABLE BIT CLEAR FOR SELECTED LINE.
6252                                    ;+
6253                                    ;  Set TX.ENBLE on lines that have a corresponding bit set in the tx enable
6254                                    ;  line bit map.
6255                                    ;-
6256 025712   030100                   4$:      BIT       R1,R0            ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
6257 025714   001402                   BEQ       6$                        ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
6258 025716   152712   000200          BISB      #BIT7,(R2)                ;ENABLE TRANSMISSION ON SELECTED LINE.
6259 025722   005204                   6$:      INC       R4               ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
6260 025724   006301                   ASL       R1                        ;SHIFT BIT MAP FOR NEXT LINE.
6261 025726   005303                   DEC       R3                        ;DECREMENT LINE NUMBER.
6262 025730   001363                   BNE       2$                        ;LOOP TO CHECK NEXT LINE.
6263
6264 025732                            60$:     PASS      R5               ;RESTORE GPRS,EXCEPT
     025732   010566   000014                   MOV       R5,R5SLOT(SP)    ;PUT R5 IN STACK SLOT.
     025736   004736                            JSR       PC,@(SP)+        ;RETURN TO PREGO5 SUBRT.
6265                                    ;R5 - LINE BIT MAP CORRESPONDING TO THE
6266                                    ; PREVIOUS LINES THAT WERE DISABLED.
6267 025740   000207                   RTS       PC
```

```
6269                              .SBTTL  GLOBAL SUBROUTINE               - TXFRPR -
6270                              ;++ ***********************************************************************
6271                              ;*                      - Transmit framming error data Routine -
6272                              ;*      This routine is used to initiate DMA mode transmission
6273                              ;*      in the Framming error test.  It sends a single character DMA buffer on
6274                              ;*      each active line in the bit map, to cause future TX interrupts which
6275                              ;*      will continue the transmission if more than one buffer is to be sent.
6276                              ;*
6277                              ;* INPUTS:      R4 - Contains the Lines on which TX is to take place.
6278                              ;*              ACTLNS - Active lines bit map.
6279                              ;*              BITTBL - Label of table of words each with a bit set.
6280                              ;*              CSRA - Contains the address of the DUT CSR.
6281                              ;*              DPENDB - Base of the data pattern end table (entry per line).
6282                              ;*              DPLENB - Base of the data pattern length table.
6283                              ;*              IESTAT - Preserved states of the DUT interrupt enable bits.
6284                              ;*              NUMLNS - Equated to number of lines on a DUT.
6285                              ;*              TXCNTB - Lable at base of the TX character counter table.
6286                              ;*              TXPTRB - Label at base of the TX data pattern pointers table.
6287                              ;*
6288                              ;* OUTPUTS:     CSR - DUT CSR IND.ADR.REG field is destroyed.
6289                              ;*              TXCNTx - Counters incremented for lines on which chars sent.
6290                              ;*              TXINTF - TX int flags (bit set if DMA.HO found set on line).
6291                              ;*
6292                              ;* CALLING SEQUENCE:    JSR     PC,TXFRPR
6293                              ;*
6294                              ;* COMMENTS:    This routine assumes that at least one data pattern should be
6295                              ;*              transmitted on each active line.
6296                              ;*              Interrupts must be disabled when calling this routine.
6297                              ;*
6298                              ;* SUBORDINATE ROUTINES CALLED: DODMA.
6299                              ;-- ***********************************************************************
6300 025742                      TXFRPR:: SAVE                         ;SAVE CONTENTS OF GPRS R0 THRU R5.
     025742  004537  005326                       JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
6301 025746  013705  002174               MOV     ACTLNS,R5            ;GET THE ACTIVE LINE BIT MAP.
6302 025752  005104                        COM     R4                   ;GET BIT MAP OF LINES THAT WILL RECEIVE DATA.
6303 025754  040405                        BIC     R4,R5                ;CLEAR LINES THAT WILL RX FROM TX LINE BIT MAP.
6304                              ;+
6305                              ; Set up loop which handles one line per iteration.
6306                              ;-
6307 025756  005001               CLR     R1                   ;CLEAR THE LINE NUMBER COUNTER.
6308                                ;+
6309                                ; If the line is inactive skip to select the next line.
6310                                ;-
6311 025760  000241              2$:      CLC                          ;CLEAR BOOLEAN REGISTER.
6312 025762  006005                        ROR     R5                   ;SHIFT BIT MAP OF LINES TO TX ON INTO BOOL.REG.
6313 025764  103017                        BCC     6$                   ;DON'T TX ON THIS LINE IF IT IS NOT ACTIVE.
6314                              ;+
6315                              ; Line is active.
6316                              ; Initiate DMA on this line.
6317                              ;   Get the data pattern length for this line.
6318                              ;-
6319 025766  010104               MOV     R1,R4                ;COPY LINE NUMBER.
6320 025770  006304                        ASL     R4                   ;CALCULATE WORD OFFSET FOR THIS LINE.
6321 025772  016403  003204               MOV     DPLENB(R4),R3        ;GET DATA PATTERN LENGTH FOR THIS LINE.
6322 025776  016402  003344               MOV     TXPTRB(R4),R2        ;PREPARE TO PASS DATA PATTERN ADR TO DODMA RTN.
6323                                ;+
6324                                ; Write DMA parameters to the DUT.
```

```
      6325                            ;-
      6326 026002  004737  017610            JSR     PC,DODMA
      6327 026006  103404                    BCS     4$                      ;SKIP ERROR IF DODMA WAS SUCCESSFUL.
      6328                            ;+
      6329                            ; Set the proper bit of the TX interrupt flags to indicate the line error.
      6330                            ;-
      6331 026010  056437  002366  002252    BIS     BITTBL(R4),TXINTF        ;INDICATE THE ERROR.
      6332 026016  000402                    BR      6$                      ;SKIP UPDATING POINTERS AND COUNTERS.
      6333                            ;+
      6334                            ; Update the TX character count for this line.
      6335                            ;-
      6336 026020  060364  003504    4$:     ADD     R3,TXCNTB(R4)   ;ADD THE DATA PATTERN LENGTH TO TX CHAR COUNT.
      6337                            ;+
      6338                            ; Increment line counter,goto next line if not done.
      6339                            ;-
      6340 026024  005201    6$:     INC     R1                      ;INCREMENT THE LINE COUNTER.
      6341 026026  005705            TST     R5                      ;TEST THE TX LINE BIT MAP.
      6342 026030  001353            BNE     2$                      ;LOOP TO SEND CHAR TO ANOTHER LINE IF NOT DONE.
      6343
      6344 026032            60$:    PASS                            ;RESTORE GPRS.
           026032  004736                            JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
      6345 026034  000207            RTS     PC
```

```
6347                                   .SBTTL  GLOBAL SUBROUTINE                  - TXIEO -
6348                                   ;++ ***********************************************************************
6349                                   ;*                         - TRANSMITTER INTERRUPT DISABLE -
6350                                   ;*      This routine is used to disable transmitter interrupts in the DHV11-M.
6351                                   ;*
6352                                   ;* INPUTS:      NONE.
6353                                   ;*
6354                                   ;* OUTPUTS:     The TX.INT.ENBL bit is cleared in the DUT CSR.
6355                                   ;*              IESTST -contains the updated status of the TX and RX interrupt
6356                                   ;*              enable bits.
6357                                   ;*
6358                                   ;* CALLING SEQUENCE:    JSR    PC,TXIEO
6359                                   ;*
6360                                   ;* COMMENTS:    The contents of the indirect address register field in
6361                                   ;*              the DUT CSR are destroyed.
6362                                   ;*
6363                                   ;* SUBORDINATE ROUTINES CALLED: NONE.
6364                                   ;-- ***********************************************************************
6365 026036   010046                  TXIEO::  MOV    R0,-(SP)          ;SAVE CONTENTS OF R0 ON THE STACK.
6366 026040   106746                           MFPS   -(SP)             ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
6367 026042   106427   000340                  MTPS   @PRIO7            ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
6368 026046   042737   177677   002234          BIC    #177677,IESTAT    ;CLEAR TX.INT.ENBL BIT IN IESTAT.
6369 026054   013777   002234   154120          MOV    IESTAT,@CSRA      ;DISABLE TX INTERRUPTS.
6370 026062   106426                           MTPS   (SP)+             ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
6371 026064   012600                           MOV    (SP)+,R0          ;RESTORE R0.
6372 026066   000207                           RTS    PC
```

```
6374                                      .SBTTL  GLOBAL SUBROUTINE                    - TXIE1 -
6375                                      ;++ ********************************************************************
6376                                      ;*                      - TRANSMITTER INTERRUPT ENABLE -
6377                                      ;*      This routine is used to enable transmitter interrupts in the DHV11-M.
6378                                      ;*
6379                                      ;* INPUTS:      NONE.
6380                                      ;*
6381                                      ;* OUTPUTS:     The TX.INT.ENBL bit is set in the DUT CSR.
6382                                      ;*              IESTST -contains the updated status of the TX and RX interrupt
6383                                      ;*              enable bits.
6384                                      ;*
6385                                      ;* CALLING SEQUENCE:    JSR     PC,TXIE1
6386                                      ;*
6387                                      ;* COMMENTS:    The contents of the indirect address register field in
6388                                      ;*              the DUT CSR are destroyed.
6389                                      ;*
6390                                      ;* SUBORDINATE ROUTINES CALLED: NONE.
6391                                      ;-- ********************************************************************
6392
6393 026070  052737  040000  002234  TXIE1::  BIS    #BIT14,IESTAT    ;SET TX.INT.ENBL BIT IN IESTAT.
6394 026076  042737  137677  002234       BIC    #137677,IESTAT   ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
6395 026104  013777  002234  154070       MOV    IESTAT,@CSRA     ;DISABLE TX INTERRUPTS.
6396 026112  000207                        RTS    PC
```

```
6398                            .SBTTL  GLOBAL SUBROUTINE              - TXRINI -
6399                            ;++ ***********************************************************************
6400                            ;*                   - Transmit and Receive Initialization Routine -
6401                            ;*         This subroutine performs the initialization of the various pointers,
6402                            ;*         counters, and flags which are used during the transmission and
6403                            ;*         reception portion of a test.  This initialization is performed on
6404                            ;*         the specified lines only, other line variables remain unchanged.
6405                            ;*
6406                            ;* INPUTS:           CHCNTB - Label at base of line character count table.
6407                            ;*                   CHRTOT - Max # of chars to RX on lines already initialized.
6408                            ;*                   DPENDB - Label at base of line data pattern end table.
6409                            ;*                   DPLENB - Label at base of line data pattern length table.
6410                            ;*                   EXCNTB - Label at base address of extra char counters table.
6411                            ;*                   IESTAT - Present state of the RX.IE and TX.IE bits.
6412                            ;*                   NUMLNS - Equated to number of lines on the DUT.
6413                            ;*                   RXCNTB - Label at base address of RX character counters table.
6414                            ;*                   RXPTRB - Lable at base adr of "next RX char" pointers table.
6415                            ;*                   TXCNTB - Label at base address of TX character counters table.
6416                            ;*                   TXPTRB - Label at base adr of "next TX char" pointers table.
6417                            ;*                   CBB - Label at base of TX/RX control block.
6418                            ;*                   CB Contents - TX/RX control block contains the following:
6419                            ;*                       CBLPRA - DUT LPR contents.
6420                            ;*                       CBLNCA - DUT LNCTRL contents.
6421                            ;*                       CBDPAA - Address of beginning of data pattern.
6422                            ;*                       CBDPLA - Length in bytes of data pattern.
6423                            ;*                       CBDPNA - Number of data patterns to transmit.
6424                            ;*                       CBMAPA - Bit map of lines to be initialized.
6425                            ;*                       CBLPBA - Type of loopback to be used for test.
6426                            ;*                       CBOFSA - Amount to offset each TX start in the data pat.
6427                            ;*                   TXRXLB - Label at base of TX/RX line association table.
6428                            ;*
6429                            ;* OUTPUTS:          CHCNT - Table of number of line TX characters (Initialized).
6430                            ;*                   CHRTOT - Maximum number of chars to receive (2 * pat length).
6431                            ;*                   DPEND - Table of data pattern ends (Initialized).
6432                            ;*                   DPLEN - Table of data pattern lengths (Initialized).
6433                            ;*                   DUT LNCTRL - Line control registers (Initialized).
6434                            ;*                   DUT LPR - Line parameter registers (Initialized).
6435                            ;*                   EXCNT - Table of extra RX char counts (Clred, selected lines).
6436                            ;*                   RXCNT - Table of RX character counts (Clred, selected lines).
6437                            ;*                   RXDONF - "Reception Done" flags (Cleared for selected lines).
6438                            ;*                   RXPTR - Table of receive pointers (Initialized).
6439                            ;*                   TXCNT - Table of TX character counters (Clred, selected lines).
6440                            ;*                   TXDONF - "Transmission Done" flags (Clred for selected lines).
6441                            ;*                   TXPTR - Table of transmit pointers (Initialized).
6442                            ;*                   TXRXL - TX/RX line association table (Initialized).
6443                            ;*
6444                            ;* CALLING SEQUENCE:    JSR     PC,TXRINI
6445                            ;*
6446                            ;* COMMENTS:         If the calculation of the CHRTOT value (2 times the data
6447                            ;*                   pattern length) results in a number greater than 64K then
6448                            ;*                   CHRTOT is initialized to 64K - 1.
6449                            ;*                   This routine will not force internal loopback based on the
6450                            ;*                   loopback type in CBLPBA.  The user must set up CBLNCA correctly
6451                            ;*                   to get internal loopback.
6452                            ;*
6453                            ;* SUBORDINATE ROUTINES CALLED: WTWLNC,WTWLPR,
6454                            ;-- ***********************************************************************
```

```
 6455 026114              TXRINI:: SAVE                    ;SAVE CONTENTS OF GPRS RO THRU R5.
      026114 004537 005326                        JSR    R5,PREGO5        ;CALL REGISTER SAVE SUBRT.
 6456                              ;+
 6457                              ; Set up the LPR and LNCTRL registers as specified in the TX/RX Control Block.
 6458                              ;-
 6459 026120 013705 003136                MOV    CBMAPA,R5        ;GET THE BIT MAP OF SELECTED LINES.
 6460 026124 013700 003126                MOV    CBLNCA,RO        ;GET THE NEW LNCTRL CONTENTS.
 6461 026130 023727 003140 000001         CMP    CBLPBA,#1        ;CHECK IF INTERNAL LOOPBACK HAS BEEN SELECTED.
 6462 026136 001002                        BNE    2$               ;SKIP SETTING INT. LOPBCK IN MAINTENANCE FIELD.
 6463 026140 052700 000200                BIS    #200,RO          ;SET INTERNAL LOOPBACK IN MAINTENANCE FIELD.
 6464 026144 004737 027234         2$:    JSR    PC,WTWLNC        ;SET UP THE LNCTRL REGS FOR SELECTED LINES.
 6465 026150 013700 003124                MOV    CBLPRA,RO        ;GET THE NEW LPR CONTENTS.
 6466 026154 004737 027264                JSR    PC,WTWLPR        ;SET UP THE LPR REGISTERS FOR SELECTED LINES.
 6467 026160 004737 025646                JSR    PC,TXENBL        ;ENABLE TX FOR ALL SELECTED LINES.
 6468                              ;+
 6469                              ; Set up and begin loop which handles one line per iteration.
 6470                              ;-
 6471 026164 005004                        CLR    R4               ;CLEAR THE LINE OFFSET.
 6472 026166 013705 003130                MOV    CBDPAA,R5        ;INITIALIZE THE TX START ADDRESS VALUE.
 6473 026172 013703 003132                MOV    CBDPLA,R3        ;GET THE LENGTH OF THE DATA PATTERN.
 6474 026176 060503                        ADD    R5,R3            ;CALCULATE END ADDRESS OF THE DATA PATTERN.
 6475 026200 036437 002366 003136  4$:    BIT    BITTBL(R4),CBMAPA ;CHECK IF THIS LINE IS SELECTED FOR INIT.
 6476 026206 001452                        BEQ    12$              ;SKIP SET UP IF LINE IS NOT SELECTED.
 6477                              ;+
 6478                              ; This line is selected for initialization.
 6479                              ;   Set up proper entry in number of chars to TX and RX table.
 6480                              ;   Include char count on this line in max allowable char total for all lines.
 6481                              ;-
 6482 026210 013701 003132                MOV    CBDPLA,R1        ;GET THE LENGTH OF THIS LINE'S DATA PATTERN.
 6483 026214 013702 003134                MOV    CBDPNA,R2        ;GET THE NUMBER OF PATTERNS TO TX AND RX.
 6484 026220 004737 022030                JSR    PC,MUL16U        ;CALCULATE THE TOTAL NUMBER OF CHARS TO TX/RX.
 6485 026224 010164 003444                MOV    R1,CHCNTB(R4)    ;SET UP THE NUMBER OF TX/RX CHARS FOR LINE.
 6486 026230 060137 002500                ADD    R1,CHRTOT        ;ADD TWICE THE NUMBER OF CHARACTERS TO TX/RX
 6487 026234 103403                        BCS    6$               ;   ON THIS LINE TO THE TOTAL NUMBER OF CHARS
 6488 026236 060137 002500                ADD    R1,CHRTOT        ;   WHICH WE WILL ALLOW TO BE RECEIVED ON
 6489 026242 103003                        BCC    8$               ;   ALL LINES.
 6490 026244 012737 177777 002500  6$:    MOV    #-1,CHRTOT       ;   SET MAX CHAR TOTAL TO -1 IF OVERFLOW.
 6491 026252                        8$:
 6492                              ;+
 6493                              ; Set up the data pattern end and length for this line.
 6494                              ;-
 6495 026252 013764 003132 003204         MOV    CBDPLA,DPLENB(R4) ;SET UP TX DATA PATTERN LENGTH FOR THIS LINE.
 6496 026260 010364 003144                MOV    R3,DPENDB(R4)    ;SET UP TX DATA PAT END ADDRESS FOR THIS LINE.
 6497                              ;+
 6498                              ; Set up the TX counter and character pointer for this line.
 6499                              ;-
 6500 026264 005064 003504                CLR    TXCNTB(R4)       ;CLEAR THE TX COUNTER FOR THIS LINE.
 6501 026270 010564 003344                MOV    R5,TXPTRB(R4)    ;SET UP THE TX CHAR POINTER FOR THIS LINE.
 6502                              ;+
 6503                              ; Set up the TX/RX line association offset table entry for this line.
 6504                              ;-
 6505 026274 010402                        MOV    R4,R2            ;SELECT LINE OFFSET FOR NON-STAGGERED LPBK.
 6506 026276 023727 003140 000002         CMP    CBLPBA,#2        ;TEST FOR STAGGERED LOOPBACK.
 6507 026304 001003                        BNE    10$              ;SKIP SETTING STAGGERED LPBK IF NOT.
 6508 026306 006202                        ASR    R2               ;FORM BYTE OFFSET INTO TABLE FROM TX LINE #.
 6509 026310 116202 005276                MOVB   STGTRB(R2),R2    ;GET THE RX LINE CORRESPONDING WITH TX LINE.
 6510 026314 010264 005236         10$:   MOV    R2,TXRXLB(R4)    ;LOAD TX TABLE ENTRY WITH RX LINE OFFSET.
```

```
6511
6512                                     ;+
6513                                     ; Set up the RX counters and character pointer for the RX line which
6514                                     ;   is associated with this TX line.
6515 026320  005062  003544             ;-
6516 026324  005062  003244                 CLR    RXCNTB(R2)       ;CLEAR THE RX COUNTER FOR THIS RX LINE.
6517 026330  010562  003404                 CLR    EXCNTB(R2)       ;CLEAR THE EXTRA CHAR COUNTER FOR THIS RX LINE.
6518                                         MOV    R5,RXPTRB(R2)    ;SET UP THE RX CHAR POINTER FOR THIS RX LINE.
6519                                     ;+
6520                                     ; Update the TX start pointer in preparation for the next line.
6521 026334  063705  003142             ;-
6522 026340  020503                  12$:   ADD    CBOFSA,R5        ;ADD THE TX OFFSET TO THE TX START POINTER.
6523 026342  103403                  14$:   CMP    R5,R3            ;COMPARE TX START WITH END OF DATA PATTERN.
6524 026344  163705  003132                 BLO    16$              ;SKIP WRAPAROUND IF START IS BEFORE PAT END.
6525 026350  000773                         SUB    CBDPLA,R5        ;SUBTRACT DATA PATTERN LENGTH FROM START.
6526                                         BR     14$              ;LOOP UNTIL START IS WITHIN DATA PATTERN.
6527                                     ;+
6528                                     ; Update the TX line number offset to the next line.
6529 026352  005204                     ;-
6530 026354  005204                  16$:   INC    R4
6531                                         INC    R4
6532                                     ;+
6533                                     ; Test for done handling all possible lines on the device.
6534 026356  020427  000020             ;-
6535 026362  002706                         CMP    R4,#NUMLNS*2     ;COMPARE OFFSET WITH 2 TIMES MAX # OF LINES.
6536                                         BLT    4$               ;LOOP IF NOT ALL LINES DONE.
6537 026364                           60$:   PASS                    ;RESTORE GPRS.
     026364  004736                                 JSR    PC,@(SP)+         ;RETURN TO PREG05 SUBRT.
6538 026366  000207                         RTS    PC
```

```
6540                              .SBTTL  GLOBAL SUBROUTINE                  - TXROFF -
6541                              ;** ****************************************************************************
6542                              ;*                          - Turn TX and RX off Routine -
6543                              ;*      This subroutine is used to turn off DUT transmission and reception.
6544                              ;*      This routine achieves this by boosting processor priority to 5 to
6545                              ;*      avoid RX interrupts and by clearing all the DUT TX.ENABLE bits to
6546                              ;*      halt TX (either DMA or single character TX).  The states of the
6547                              ;*      TX.ENABLE bits and the processor priority are saved for restoration
6548                              ;*      when TX and RX are re-enabled.
6549                              ;*
6550                              ;* INPUTS:        MAPLNS - Bit map of all possible lines on the DUT.
6551                              ;*
6552                              ;* OUTPUTS:       SAVPRI - Saved processor priority.
6553                              ;*                SAVTEN - Bit map of TX.ENBL bits (Bit set if TX.ENBL was set).
6554                              ;*
6555                              ;* CALLING SEQUENCE:   JSR      PC,TXROFF
6556                              ;*
6557                              ;* COMMENTS:
6558                              ;*
6559                              ;* SUBORDINATE ROUTINES CALLED: TXDSBL.
6560                              ;-- ****************************************************************************
6561
6562 026370                      TXROFF:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
     026370 004537 005326                        JSR      R5,PREG05        ;CALL REGISTER SAVE SUBRT.
6563 026374 106737 002246                MFPS     SAVPRI                  ;GET THE PRESENT PROCESSOR PRIORITY.
6564 026400 106427 000240                MTPS     #PRIO5                  ;DISABLE DUT INTERRUPTS.
6565 026404 012705 000377                MOV      #MAPLNS,R5              ;PREPARE TO DISABLE TX ON ALL DUT LINES.
6566 026410 004737 025552                JSR      PC,TXDSBL               ;CLEAR ALL DUT TX.ENABLE BITS.
6567 026414 010537 002244                MOV      R5,SAVTEN               ;PRESERVE THE PREVIOUS TX.ENABLE BIT STATES.
6568 026420                      60$:     PASS                            ;RESTORE GPRS.
     026420 004736                        JSR      PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
6569 026422 000207                RTS      PC
```

```
6571                                    .SBTTL  GLOBAL SUBROUTINE                  - TXRON -
6572                                    ;++ ****************************************************************************
6573                                    ;*                        - Turn TX and RX on Routine -
6574                                    ;*      This subroutine is used to turn on DUT transmission and reception.
6575                                    ;*      This routine restores the DUT TX.ENABLE bits and the processor priority
6576                                    ;*      to the states saved by the TXROFF routine.
6577                                    ;*
6578                                    ;* INPUTS:        SAVPRI - Saved processor priority.
6579                                    ;*                SAVTEN - Bit map of TX.ENBL bits (Bit setif TX.ENBL was set).
6580                                    ;*
6581                                    ;* OUTPUTS:       DUT TX.ENABLE bits - Set to specified states.
6582                                    ;*                PROCESSOR PRIORITY - Set to specified priority.
6583                                    ;*
6584                                    ;* CALLING SEQUENCE:   JSR    PC,TXRON
6585                                    ;*
6586                                    ;* COMMENTS:
6587                                    ;*
6588                                    ;* SUBORDINATE ROUTINES CALLED: TXENBL.
6589                                    ;-- ****************************************************************************
6590
6591 026424         .                  TXRON:: SAVE                      ;SAVE CONTENTS OF GPRS R0 THRU R5.
     026424  004537  005326                             JSR    R5,PREGO5    ;CALL REGISTER SAVE SUBRT.
6592 026430  013705  002244                     MOV    SAVTEN,R5          ;GET THE SAVED STATES OF THE TX.ENABLE BITS.
6593 026434  004737  025646                     JSR    PC,TXENBL          ;SET THE SPECIFIED TX.ENABLE BITS.
6594 026440  106437  002246                     MTPS   SAVPRI             ;RESTORE THE PROCESSOR PRIORITY.
6595 026444                            60$:     PASS                      ;RESTORE GPRS.
     026444  004736                             JSR    PC,@(SP)+                  ;RETURN TO PREGO5 SUBRT.
6596 026446  000207                             RTS    PC
```

```
6598                                    .SBTTL  GLOBAL SUBROUTINE                    - TXRREP -
6599                                    ;++  ***********************************************************************
6600                                    ;*                     - Report Final TX/RX Errors Routine -
6601                                    ;*       This subroutine reports errors which are found after the completion
6602                                    ;*       of the TX, RX, and verification of data patterns.  It reports errors
6603                                    ;*       dealing with incomplete TX or RX and with DMA_START bits.
6604                                    ;*
6605                                    ;*  INPUTS:         ACTLNS - Bit map of active DUT lines.
6606                                    ;*                  DPLENB - Label at base of the data pattern lengths table.
6607                                    ;*                  ERRMSG - Address of primary error message for this routine.
6608                                    ;*                  ERRNBR - Error number of error reported in this routine.
6609                                    ;*                  RXCNTB - Label at base of the RX character counters table.
6610                                    ;*                  RXDONF - Reception done flags.
6611                                    ;*                  TXCNTB - Label at base of the TX character counters table.
6612                                    ;*                  TXDONF - Transmission done flags.
6613                                    ;*                  TXINTF - Contains bit map of lines with DMA_START bit errors.
6614                                    ;*
6615                                    ;*  OUTPUTS:        CARRY FLAG - Restored to its entering value.
6616                                    ;*                  ERRBLK - Address of the error reporting routine (Destroyed).
6617                                    ;*                  Messages may be printed at the operator console.
6618                                    ;*
6619                                    ;*  CALLING SEQUENCE:    JSR      PC,TXRREP
6620                                    ;*
6621                                    ;*  COMMENTS:       This routine reports errors at Initial ERRNBR thru
6622                                    ;*                  Initial ERRMBR+2.
6623                                    ;*                  If no lines failed to complete their reception or failed to
6624                                    ;*                  complete their transmission or had DMA_START bit errors
6625                                    ;*                  then no messages are printed.
6626                                    ;*
6627                                    ;*  SUBORDINATE ROUTINES CALLED: CONMAP,ER9005,ER9102,RDMAST,RRXNDN,RTXNDN.
6628                                    ;--  ***********************************************************************
6629
6630 026450                            TXRREP:: SAVE                              ;SAVE CONTENTS OF GPRS R0 THRU R5.
     026450  004537  005326                     JSR      R5,PREG05       ;CALL REGISTER SAVE SUBRT.
6631 026454  006003                             ROR      R3              ;ROTATE CARRY INTO GPR TO SAVE CARRY STATE.
6632 026456  013704  005320                     MOV      ERRNBR,R4       ;SAVE THE INITIAL ERROR NUMBER VALUE.
6633 026462  013705  002174                     MOV      ACTLNS,R5       ;GET THE ACTIVE LINES BIT MAP.
6634 026466  004737  024034                     JSR      PC,RDMAST       ;REPORT ANY DMA_START BIT ERRORS.
6635 026472  005237  005320                     INC      ERRNBR          ;SELECT INTIAL ERROR NUMBER + 1.
6636 026476  004737  024420                     JSR      PC,RTXNDN       ;REPORT TX NOT COMPLETE IF NECESSARY.
6637 026502  005237  005320                     INC      ERRNBR          ;SELECT INITIAL ERROR NUMBER + 2.
6638 026506  004737  017300                     JSR      PC,CONMAP       ;GENERATE AN ASSOCIATED LINE BIT MAP.
6639 026512  004737  024360                     JSR      PC,RRXNDN       ;REPORT RX NOT COMPLETE IF NECESSARY.
6640 026516  010437  005320                     MOV      R4,ERRNBR       ;RESTORE THE INITIAL ERROR NUMBER VALUE.
6641
6642 026522  006103                             ROL      R3              ;ROTATE SAVED CARRY STATE BACK INTO CARRY.
6643 026524                            60$:      PASS                              ;RESTORE GPRS, THIS ROUTINE PRESERVES THE
     026524  004736                             JSR      PC,@(SP)+                 ;RETURN TO PREG05 SUBRT.
6644 026526  000207                             RTS      PC              ;  INITIAL CARRY STATE.
```

```
6646                                    .SBTTL  GLOBAL SUBROUTINE                 - UNSDIV -
6647                                    ;++ **********************************************************************
6648                                    ;*                        - Unsigned Divide Routine -
6649                                    ;*      This subroutine is used to divide a 32 bit unsigned dividend by a
6650                                    ;*      16 bit unsigned divisor giving a 16 bit quotient.  All numbers are
6651                                    ;*      considered to be unsigned.  A success flag is not set on return if
6652                                    ;*      the quotient was too big to be contained in 16 bits.
6653                                    ;*
6654                                    ;* INPUTS:        R1 - The divisor, unsigned, 16 bits.
6655                                    ;*                R2 - Most significant word of the dividend, unsigned, 16 bits.
6656                                    ;*                R3 - Least significant word of the dividend, unsigned, 16 bits.
6657                                    ;*
6658                                    ;* OUTPUTS:       R1 - Quotient, unsigned, 16 bits (177777 if overflow).
6659                                    ;*                CARRY - Success flag, set if complete quotient fits in 16 bits.
6660                                    ;*
6661                                    ;* CALLING SEQUENCE:   JSR     PC,UNSDIV
6662                                    ;*
6663                                    ;* COMMENTS:      If the divisor is 0 the quotient is returned as all ones
6664                                    ;*                (177777) and the carry is clear regardless of the dividend.
6665                                    ;*
6666                                    ;* SUBORDINATE ROUTINES CALLED: None.
6667                                    ;-- **********************************************************************
6668
6669 026530                            UNSDIV:: SAVE                        ;SAVE CONTENTS OF GPRS R0 THRU R5.
     026530  004537  005326                     JSR     R5,PREG05           ;CALL REGISTER SAVE SUBRT.
6670                                    ;+
6671                                    ; Check for quotient greater than 16 bits condition.
6672                                    ;-
6673 026534  010204                             MOV     R2,R4               ;GET MSW OF DIVIDEND FOR SUBTRACT.
6674 026536  160104                             SUB     R1,R4               ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
6675 026540  103403                             BCS     2$                  ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
6676 026542  012701  177777                     MOV     #-1,R1              ;SET QUOTIENT TO ALL ONES (177777).
6677 026546  000442                             BR      60$                 ;EXIT WITH CARRY CLEAR.
6678                                    ;+
6679                                    ; Set up counters and various working GPRs.
6680                                    ;-
6681 026550  005004                     2$:     CLR     R4                  ;CLEAR THE LSW OF THE DIVISOR.
6682 026552  000241                             CLC                         ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
6683 026554  006001                             ROR     R1                  ; DIVISOR BY
6684 026556  006004                             ROR     R4                  ; 2(UNSIGNED)
6685 026560  012700  000020                     MOV     #16.,R0             ;SET UP INITIAL SHIFT COUNT TO 16.
6686                                    ;+
6687                                    ; The subtract and shift loop.
6688                                    ;-
6689 026564  010246                     4$:     MOV     R2,-(SP)            ;SAVE MSWORD OF DIVIDEND.
6690 026566  010346                             MOV     R3,-(SP)            ;SAVE LSWORD OF DIVIDEND.
6691 026570  160403                             SUB     R4,R3               ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
6692 026572  005602                             SBC     R2                  ;MSWORD DIVIDEND - BORROW .
6693 026574  103402                             BCS     6$                  ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
6694 026576  160102                             SUB     R1,R2               ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
6695 026600  103003                             BCC     8$                  ;IF NO BORROW, IT WENT, CARRY IS CLEAR.
6696                                    ;+
6697                                    ; It didn't go, so we shift a 1 into the quotient (complemented later).
6698                                    ; Carry is set.
6699                                    ;-
6700 026602  012603                     6$:     MOV     (SP)+,R3            ;RESTORE LSWORD OF DIVIDEND.
6701 026604  012602                             MOV     (SP)+,R2            ;RESTORE MSWORD OF DIVIDEND.
```

```
6702 026606  000401                  BR      10$             ;GOTO SHIFT 1 INTO THE QUOTIENT.
6703
6704                         ;+
6705                         ; It went, so we restore the stack and shift a 0 into quotient (will be
6706                         ; complemented later).   Carry is clear.
6707 026610  012626         ;-
                            8$:      MOV     (SP)+,(SP)+     ;POP THE SAVED DIVIDEND OFF OF THE STACK.
6708
6709                         ;+
6710                         ; Shift the result of the subtract attempt into the quotient shift reg.
6711 026612  006105         ;-
                            10$:     ROL     R5              ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
6712 026614  000241                  CLC                     ;DIVIDE THE
6713 026616  006001                  ROR     R1              ;  DEVISOR BY
6714 026620  006004                  ROR     R4              ;  2 (UNSIGNED).
6715 026622  005300                  DEC     R0              ;COUNT THIS SHIFT AND SUBTRACT.
6716 026624  001357                  BNE     4$              ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
6717 026626  005105                  COM     R5              ;GET QUOTIENT FROM INVERTED QUOTIENT.
6718                         ;+
6719                         ; Now we either round up or leave quotient alone.
6720                         ;-
6721 026630  000241                  CLC                     ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
6722 026632  006103                  ROL     R3              ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
6723 026634  103402                  BCS     12$             ;IF CARRY FROM SHIFT, ROUND UP.
6724 026636  160403                  SUB     R4,R3           ;SUBTRACT DIVISOR FROM DIVIDEND.
6725 026640  103403                  BCS     14$             ;IF BORROW, DON'T ROUND UP.
6726                         ;+
6727                         ; Round up, extra subtract went.
6728                         ;-
6729 026642  005205         12$:     INC     R5              ;INCREMENT THE QUOTIENT BY ONE.
6730 026644  001001                  BNE     14$             ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
6731 026646  005305                  DEC     R5              ;DON'T LET ROUNDING CAUSE OVERFLOW.
6732                         ;+
6733                         ; All done, pass quotient and exit.
6734                         ;-
6735 026650  010501         14$:     MOV     R5,R1           ;PASS QUOTIENT BACK IN R1.
6736 026652  000261                  SEC                     ;INDICATE NO OVERFLOW.
6737
6738 026654                 60$:     PASS    R1              ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
     026654  010166  000004                  MOV     R1,R1SLOT(SP)          ;PUT R1 IN STACK SLOT.
     026660  004736                           JSR     PC,@(SP)+              ;RETURN TO PREGO5 SUBRT.
6739                                                          ;R1 - 16 BIT, UNSIGNED QUOTIENT,
6740 026662  000207                  RTS     PC              ;CARRY - SET INDICATES NO OVERFLOW (SUCCESS).
```

```
 6742                                          .SBTTL  GLOBAL SUBROUTINE                  - UPDCHR -
 6743                                          ;++ ********************************************************************************
 6744                                          ;*                    - Update Character Pointers and Counters Routine -
 6745                                          ;*          This subroutine updates the pointers and counters associated with
 6746                                          ;*          the reception of a character on a specified line.  The receive char
 6747                                          ;*          pointer is set to the next expected character, the receive char count
 6748                                          ;*          is incremented, and the count is checked to determine if the reception
 6749                                          ;*          is complete.  If the reception is complete the Reception Done Flag
 6750                                          ;*          is set for the specified line.
 6751                                          ;*
 6752                                          ;* INPUTS:          R3 - Line number times 2 of line on which char was received.
 6753                                          ;*                  BITTBL - Label of table of words used to form single bit maps.
 6754                                          ;*                  CHCNTB - Base of number of chars to TX on each line table.
 6755                                          ;*                  DPENDB - Base of data pattern end addresses table.
 6756                                          ;*                  DPLENB - Base of data pattern lengths table.
 6757                                          ;*                  RXCNTB - Base of the RX character counters table.
 6758                                          ;*                  RXPTRB - Base of the RX character pointers table.
 6759                                          ;*                  TXRXLB - Base of TX/RX line number association table.
 6760                                          ;*
 6761                                          ;* OUTPUTS:   Following variables updated for line on which char was received:
 6762                                          ;*                  RXCNT - Count of the number of characters received on line.
 6763                                          ;*                  RXDONF - RX done flags with BIT0 for line 0 ... (Updated).
 6764                                          ;*                  RXPTR - Updated to point to the next expected char on line.
 6765                                          ;*
 6766                                          ;* CALLING SEQUENCE:     JSR     PC,UPDCHR
 6767                                          ;*
 6768                                          ;* COMMENTS:
 6769                                          ;*
 6770                                          ;* SUBORDINATE ROUTINES CALLED: None.
 6771                                          ;-- ********************************************************************************
 6772 026664                          UPDCHR:: SAVE                             ;SAVE CONTENTS OF GPRS R0 THRU R5.
      026664   004537   005326 -                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
 6773 026670   016302   005236                 MOV     TXRXLB(R3),R2    ;GET TX LINE NUMBER OFFSET FOR THIS RX LINE.
 6774                                          ;+
 6775                                          ; Update the RX data pointer with wraparound at the end of the data pattern.
 6776                                          ;-
 6777 026674   016301   003404                 MOV     RXPTRB(R3),R1    ;GET THE RX DATA POINTER FROM THE RX PTR TABLE.
 6778 026700   005201                          INC     R1               ;INCREMENT THE RX POINTER VALUE BY 1.
 6779 026702   020162   003144                 CMP     R1,DPENDB(R2)    ;CMP RX PTR VALUE WITH ADR OF END OF DATA PAT.
 6780 026706   103402                          BLO     2$               ;SKIP WRAPPING RX PTR AROUND IF NOT AT END.
 6781 026710   166201   003204                 SUB     DPLENB(R2),R1    ;WRAP RX PTR AROUND TO START OF DATA PATTERN.
 6782 026714   010163   003404          2$:    MOV     R1,RXPTRB(R3)    ;UPDATE THE RX POINTER WITH THE NEW VALUE.
 6783                                          ;+
 6784                                          ; Update the RX character count with overflow detection.
 6785                                          ;-
 6786 026720   016301   003544                 MOV     RXCNTB(R3),R1    ;GET THE RX CHARACTER COUNT.
 6787 026724   005201                          INC     R1               ;INCREMENT THE RX CHAR COUNT VALUE BY 1.
 6788 026726   001002                          BNE     4$               ;SKIP SETTING COUNT TO MAX IF NO OVERFLOW.
 6789 026730   012701   177777                 MOV     #-1,R1           ;SET RX CHAR COUNT VALUE TO MAX VALUE.
 6790 026734   010163   003544          4$:    MOV     R1,RXCNTB(R3)    ;UPDATE THE RX CHAR COUNT WITH NEW VALUE.
 6791                                          ;+
 6792                                          ; Check for RX completion on this line.
 6793                                          ; If RX is complete on this line, set the correct RX done flag.
 6794                                          ;-
 6795 026740   016204   003444                 MOV     CHCNTB(R2),R4    ;GET THE NUMBER OF TX CHARS IN COMPLETE TX.
 6796 026744   020104                          CMP     R1,R4            ;COMPARE RX CHAR COUNT WITH NUMBER OF TX CHARS.
 6797 026746   103403                          BLO     60$              ;EXIT ROUTINE IF NOT ALL CHARS RECEIVED.
```

GLOBAL SUBROUTINE   - UPDCHR -

```
    6798 026750  056337  002366  002506        BIS     BITTBL(R3),RXDONF        ;SET THE RX DONE FLAG FOR THIS LINE.
    6799
    6800 026756                          60$:    PASS                            ;RESTORE GPRS.
         026756  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
    6801 026760  000207                          RTS     PC
```

```
6803                                    .SBTTL  GLOBAL SUBROUTINE               - VANSUP -
6804                                    ;++ ******************************************************************************
6805                                    ;*                      - TRANSMISSION / RECEPTION SET-UP ROUTINE -
6806                                    ;*
6807                                    ;*           This routine is used to initialise both the DUT and the
6808                                    ;*           transmission/reception control parameters to the correct
6809                                    ;*           state, prior to a Single character or DMA tranmission,
6810                                    ;*           reception test.
6811                                    ;*
6812                                    ;* INPUTS:    R1 - TX, RX LPR contents.
6813                                    ;*            R2 - Start address of data pattern to transmit.
6814                                    ;*            R3 - Length of data pattern.
6815                                    ;*            R4 - Number of patterns to transmit.
6816                                    ;*            ACTLNS - Contains a bit map of all currently active lines.
6817                                    ;*            LOPBCK - Contains the type of loopback mode selected.
6818                                    ;*            CBB - Label at base of TX/RX control block.
6819                                    ;*
6820                                    ;* OUTPUTS:   The contents of the TX/RX control block (CCB) are destroyed.
6821                                    ;*            The indirect address field of the DUT CSR may be destroyed.
6822                                    ;*            The DUT's LPR's and LNC's may be modified.
6823                                    ;*            The following pointers and counters are initialised:
6824                                    ;*            CHCNT,CHRTOT,DPEND,DPLEN,EXCNT,RXCNT,RXPTR,TXCNT,
6825                                    ;*            TXPTR,TXRXL.
6826                                    ;*            CHRTOT, RXDONF, TXDONF and TXINTF are cleared.
6827                                    ;*
6828                                    ;* CALLING SEQUENCE:    JSR     PC,VANSUP
6829                                    ;*
6830                                    ;* COMMENTS:  Modem loopback mode is inhibited if it has been selected
6831                                    ;*            via Hardware P-table questions, and internal loopback mode
6832                                    ;*            is forced to take place.
6833                                    ;*
6834                                    ;*
6835                                    ;* SUBORDINATE ROUTINES CALLED: CONMAP,RXENBL,TXRINI.
6836                                    ;-- ******************************************************************************
6837
6838 026762                    VANSUP:: SAVE                       ;SAVE CONTENTS OF THE GPR'S RO THRU R5.
     026762  004537  005326                    JSR     R5,PREGO5      ;CALL REGISTER SAVE SUBRT.
6839 026766  005037  002500            CLR     CHRTOT             ;CLEAR TOTAL RECEIVED CHAR COUNTER.
6840 026772  005037  002252            CLR     TXINTF             ;CLEAR FLAGS USED TO LOG DMA H.OVER ERRORS.
6841 026776  005037  002504            CLR     TXDONF             ;CLEAR THE TX DONE FLAGS.
6842 027002  005037  002506            CLR     RXDONF             ;CLEAR THE RX DONE FLAGS.
6843                                    ;+
6844                                    ;  Set up the Transmission/Reception Control block to the desired state.
6845                                    ;-
6846 027006  010137  003124            MOV     R1,CBB             ;SET CONTENTS OF LPR PARAMS IN TX/RX C.BLK.
6847 027012  012701  003124            MOV     #CBB,R1            ;GET BASE ADDRESS OF CONTROL BLOCK.
6848 027016  005201                     INC     R1                 ;INCREMENT ADDRESS FOR NEXT WORD
6849 027020  005201                     INC     R1                 ;INITIALISE THE FOLLOWING IN THE CNTRL.BLK:
6850 027022  012721  000004            MOV     #4,(R1)+           ; LNCTRL PARAMETER, ENABLE RECEIVERS.
6851 027026  010221                     MOV     R2,(R1)+           ; START ADDRESS OF DATA PATTERN.
6852 027030  010321                     MOV     R3,(R1)+           ; DATA PATTERN LENGTH.
6853 027032  010421                     MOV     R4,(R1)+           ; NUMBER OF DATA PATTERNS TO TRANSMIT.
6854 027034  013721  002174            MOV     ACTLNS,(R1)+       ; BIT MAP OF LINES TO INITIALISE.
6855 027040  032737  000004  002176    BIT     #BIT2,LOPBCK       ;TEST IF MODEM LOOPBACK MODE HAS BEEN SELECTED.
6856 027046  001404                     BEQ     2$                 ;DONT SELECT INTERNL LOPBCK IF STAGRD OR LOCAL.
6857 027050  012702  000001            MOV     #1,R2              ;FORCE INTERNAL LOOPBACK MODE TO BE SELECTED.
6858 027054  110221                     MOVB    R2,(R1)+           ;INITIALISE LOOPBACK MODE IN CONTROL BLOCK.
```

GLOBAL SUBROUTINE   - VANSUP -

```
6859 027056  000402                         BR      4$              ;SKIP NEXT INSTRUCTION IF IN MODEM LOOPBACK.
6860 027060  113721  002176       2$:       MOVB    LOPBCK,(R1)+    ;SET LOOPBACK MODE.
6861 027064  005201              4$:        INC     R1              ;INCREMENT ADDRESS FOR THE NEXT WORD.
6862 027066  012711  000002                 MOV     #2,(R1)         ;SET AMOUNT OF OFFSET EACH TX STARTS AT TO 2.
6863                              ;+
6864                              ; Initialise the DUT and the associated pointers and counters, to the state
6865                              ; dictated by the contents of the TX/RX control block.
6866                              ;-
6867 027072  004737  026114                 JSR     PC,TXRINI       ;INITIALISE DUT.
6868                              ;+
6869                              ; Initialise pointers and counters for inactive lines to zero.
6870                              ;-
6871 027076  012701  000377                 MOV     #MAPLNS,R1      ;GET THE LINE BIT MAP FOR ALL LINES.
6872 027102  013702  002174                 MOV     ACTLNS,R2       ;GET THE ACTIVE LINE BIT MAP.
6873 027106  005101                          COM     R1              ;
6874 027110  005102                          COM     R2              ;
6875 027112  040102                          BIC     R1,R2           ;GENERATE AN IN-ACTIVE LINE BIT MAP.
6876 027114  010237  003136                 MOV     R2,CBMAPA       ;MOVE BIT MAP TO THE CONTROL BLOCK.
6877 027120  005037  003134                 CLR     CBDPNA          ;CLEAR THE REPEAT TX COUNT IN CNTRL BLCK.
6878 027124  004737  026114                 JSR     PC,TXRINI       ;SET UP PARAMETERS FOR INACTIVE LINES.
6879                              ;+
6880                              ; Disable Receivers on all lines to ensure correct initialisation of only the
6881                              ; lines that are selected.
6882                              ;-
6883 027130  012705  000377                 MOV     #MAPLNS,R5      ;SET-UP BIT MAP FOR ALL LINES.
6884 027134  004737  024460                 JSR     PC,RXDSBL       ;DISABLE RX ON ALL LINES.
6885                              ;+
6886                              ; Enable receivers on associated (RX) lines.
6887                              ;-
6888 027140  013705  002174                 MOV     ACTLNS,R5       ;GET THE ACTIVE LINE BIT MAP.
6889 027144  004737  017300                 JSR     PC,CONMAP       ;GENERATE AN ASSOCIATED LINE BIT MAP.
6890 027150  004737  024554                 JSR     PC,RXENBL       ;ENABLE RECEIVERS ON ASSOCIATED LINES.
6891 027154              60$:       PASS                             ;RESTORE GPR'S.
     027154  004736                          JSR     PC,@(SP)+              ;RETURN TO PREG05 SUBRT.
6892 027156  000207                          RTS     PC
```

```
6894                            .SBTTL  GLOBAL SUBROUTINE                - WAIBIS -
6895                            ;++ ***************************************************************
6896                            ;*                  - Wait For Bit Set Routine -
6897                            ;*      This subroutine waits for the specified bit to become set.  If the
6898                            ;*      specified bit goes to a set state within the specified time-out
6899                            ;*      period a success indication is returned by this routine.
6900                            ;*      The last value which is read looking for the condition is returned to
6901                            ;*      allow the use of this routine to look for destructive read conditions.
6902                            ;*
6903                            ;* INPUTS:        R1 - Time-out value and bit number indication:
6904                            ;*                     Bits 15 thru 12 - Number of bit to test (range 0 thru 15).
6905                            ;*                     Bits 11 thru 0  - Time-out value in milli-seconds (4095 max).
6906                            ;*                R2 - Address of word containing the bit to test.
6907                            ;*                     MSLCNT.
6908                            ;*
6909                            ;* OUTPUTS:       R2 - The last word which was read to check for the condition.
6910                            ;*                CARRY - Success flag (CARRY set if bit set before time-out).
6911                            ;*
6912                            ;* CALLING SEQUENCE:    MOV     #130040,R1      ;PASS BIT 11 (13 OCTAL) AND
6913                            ;*                                             ;  32 (40 OCTAL) MS DELAY.
6914                            ;*                     MOV     @LABEL,R2       ;TEST BIT IN WORD AT "LABEL".
6915                            ;*                     JSR     PC,WAIBIS       ;WAIT 32 MS FOR BIT 11 TO SET.
6916                            ;*
6917                            ;* COMMENTS:
6918                            ;*
6919                            ;* SUBORDINATE ROUTINES CALLED: MSLGET.
6920                            ;-- ***************************************************************
6921
6922 027160                    WAIBIS:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
     027160 004537 005326               JSR     R5,PREG05      ;CALL REGISTER SAVE SUBRT.
6923 027164 010204                      MOV     R2,R4          ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
6924 027166 010102                      MOV     R1,R2
6925 027170 042701 170000               BIC     #170000,R1     ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
6926 027174 042702 007777               BIC     #7777,R2       ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
6927 027200 000302                      SWAB    R2             ;PUT LINE NUMBER FIELD IN LSBYTE.
6928 027202 006202                      ASR     R2             ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
6929 027204 006202                      ASR     R2             ;  POSITION TO USE IT AS A WORD TABLE OFFSET
6930 027206 006202                      ASR     R2             ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
6931 027210 016202 002366               MOV     BITTBL(R2),R2  ;GET BIT MAP OF LINE TO TEST FROM TABLE.
6932 027214 010203                      MOV     R2,R3          ;INDICATE THAT THE BIT SHOULD BE SET.
6933 027216 004737 021464       JSR     PC,MSLGET      ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
6934                                                            ;  CARRY IS CORRECT UPON MSLGET RETURN.
6935 027222 010002                      MOV     R0,R2          ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
6936 027224               60$:  PASS    R2             ;RESTORE GPRS, EXCEPT THE FOLLOWING:
     027224 010266 000006               MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
     027230 004736                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
6937                                                            ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
6938 027232 000207                      RTS     PC             ;  CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).
```

```
6940                                      .SBTTL  GLOBAL SUBROUTINE                    - WTWLNC -
6941                                      ;++ ********************************************************************
6942                                      ;*                    - Line Control Register Setup Routine -
6943                                      ;*       This subroutine is used to set the Device Under Test (DUT) Line
6944                                      ;*       Control Registers (LNCTRL) to the specified state.  Only the LNCTRLS
6945                                      ;*       for the specified lines are altered.
6946                                      ;*
6947                                      ;* INPUTS:        RO - New line parameters.
6948                                      ;*                R5 - Bit map of lines to be altered.
6949                                      ;*                CSRA - Contains address of the DUT CSR.
6950                                      ;*                IESTAT - Contains the current state of the TX and RX interrupt
6951                                      ;*                  enable bits in the CSR.
6952                                      ;*                LNCTRA - Contains address of the DUT LNCTRL registers.
6953                                      ;*
6954                                      ;* OUTPUTS:       LNCTRL - Specified DUT Line Control Registers are altered.
6955                                      ;*
6956                                      ;* CALLING SEQUENCE:     JSR      PC,WTWLNC
6957                                      ;*
6958                                      ;* COMMENTS:
6959                                      ;*
6960                                      ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
6961                                      ;-- ********************************************************************
6962
6963 027234                              WTWLNC:: SAVE                              ;SAVE CONTENTS OF GPRS RO THRU R5.
     027234  004537  005326                               JSR      R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
6964                                      ;+
6965                                      ; Set up the parameters for the call to ALTFLD.
6966                                      ;-
6967 027240  013701  002212                       MOV      LNCTRA,R1       ;SET UP THE REGISTER ADDRESS PARAMETER.
6968 027244  010002                               MOV      RO,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
6969 027246  010503                               MOV      R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
6970 027250  012704  177777                       MOV      #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
6971                                      ;+
6972                                      ; Call the subroutine which alters the register contents.
6973                                      ;-
6974 027254  004737  015722                       JSR      PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.
6975
6976 027260                              60$:     PASS                              ;RESTORE GPRS.
     027260  004736                               JSR      PC,@(SP)+                   ;RETURN TO PREGO5 SUBRT.
6977 027262  000207                               RTS      PC
```

```
6979                                    .SBTTL  GLOBAL SUBROUTINE                    - WTWLPR -
6980                                    ;++ *********************************************************************
6981                                    ;*                      - Line Parameter Register Setup Routine -
6982                                    ;*          This subroutine is used to set the Device Under Test (DUT) Line
6983                                    ;*          Parameter Registers (LPR) to the specified state.  Only the LPRs for
6984                                    ;*          the specified lines are altered.
6985                                    ;*
6986                                    ;* INPUTS:        R0 - New line parameters.
6987                                    ;*                R5 - Bit map of lines to be altered.
6988                                    ;*                CSRA - Contains address of the DUT CSR.
6989                                    ;*                IESTAT - Contains the current state of the TX and RX interrupt
6990                                    ;*                 enable bits in the CSR.
6991                                    ;*                LPRA - Contains address of the DUT LPR.
6992                                    ;*
6993                                    ;* OUTPUTS:       LPR - Specified DUT Line Paramter Registers are altered.
6994                                    ;*
6995                                    ;* CALLING SEQUENCE:    JSR     PC,WTWLPR
6996                                    ;*
6997                                    ;* COMMENTS:
6998                                    ;*
6999                                    ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
7000                                    ;-- *********************************************************************
7001
7002 027264                            WTWLPR:: SAVE                       ;SAVE CONTENTS OF GPRS R0 THRU R5.
     027264  004537  005326                     JSR     R5,PREG05          ;CALL REGISTER SAVE SUBRT.
7003                                    ;+
7004                                    ; Set up the parameters for the call to ALTFLD.
7005                                    ;-
7006 027270  013701  002206                     MOV     LPRA,R1            ;SET UP THE REGISTER ADDRESS PARAMETER.
7007 027274  010002                             MOV     R0,R2              ;SET UP THE DESIRED REGISTER CONTENTS.
7008 027276  010503                             MOV     R5,R3              ;SET UP THE BIT MAP OF LINES TO ALTER.
7009 027300  012704  177777                     MOV     #-1,R4             ;SELECT ALL REGISTER BITS TO BE ALTERED.
7010                                    ;+
7011                                    ; Call the subroutine which alters the register contents.
7012                                    ;-
7013 027304  004737  015722                     JSR     PC,ALTFLD          ;ALTER THE REGISTER CONTENTS.
7014
7015 027310                            60$:     PASS                       ;RESTORE GPRS.
     027310  004736                             JSR     PC,@(SP)+                    ;RETURN TO PREG05 SUBRT.
7016 027312  000207                             RTS     PC
```

```
7018                                    .SBTTL   INTERRUPT SERVICE ROUTINE          - CLKINT -
7019                                    ;++ *******************************************************************************
7020                                    ;*         This routine is executed CLKHRZ times per second.  It decrements the
7021                                    ;*         two timer counters down to zero.
7022                                    ;*
7023                                    ;* INPUTS:         TIMER1 - Timer counter #1.
7024                                    ;*                 TIMER2 - Timer counter #2.
7025                                    ;*                 TIMER3 - Timer counter for call of BREAK macro.
7026                                    ;*
7027                                    ;* OUTPUTS:        The 2 timer counters are decremented if they are not zero.
7028                                    ;*
7029                                    ;* CALLING SEQUENCE:    Put #CLKINT in the clock interrupt vector slot.
7030                                    ;*                      Put the desired time period (seconds times CLKHRZ) in
7031                                    ;*                      either TIMER1 or TIMER2 and poll the respective timer
7032                                    ;*                      counter to detect its going to 0 on time-out.
7033                                    ;*
7034                                    ;* COMMENTS:       The 2 counters will not wraparound but will stop at 0.  This
7035                                    ;*                 allows the detection of a time-out any time after the time-out
7036                                    ;*                 has occurred until the timer counter is set to another value.
7037                                    ;*
7038                                    ;* SUBORDINATE ROUTINES CALLED: None.
7039                                    ;-- *******************************************************************************
7040
7041 027314  005737  002302       CLKINT:: TST    TIMER1          ;CHECK FOR TIMER1 AT ZERO.
7042 027320  001402                         BEQ    2$              ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
7043 027322  005337  002302                DEC    TIMER1          ;DECREMENT TIME COUNT.
7044 027326  005737  002304       2$:      TST    TIMER2          ;CHECK FOR TIMER2 AT ZERO.
7045 027332  001402                         BEQ    4$              ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
7046 027334  005337  002304                DEC    TIMER2          ;DECREMENT TIME COUNT.
7047 027340  005337  002306       4$:      DEC    TIMER3          ;DECREMENT THE BREAK COUNT.
7048 027344  001006                         BNE    60$             ;EXIT IF NOT TIME TO CALL BREAK.
7049 027346  013737  002310  002306       MOV    BCOUNT,TIMER3   ;SET UP TIME TILL NEXT BREAK.
7050 027354  010046                         MOV    R0,-(SP)        ;SAVE CONTENTS OF R0 FROM BREAK MACRO.
7051 027356                                 BREAK                  ;CHECK FOR OPERATOR CONTROL/C.
     027356  104422                                                         TRAP    C$BRK
7052 027360  012600                         MOV    (SP)+,R0        ;RESTORE CONTENTS OF R0.
7053 027362  000002       60$:     RTI
```

```
7055                                    .SBTTL   INTERRUPT SERVICE ROUTINE       - RXCHRS -
7056                                    ;++ ***************************************************************************
7057                                    ;*            - DMA RECEIVE Interrupt Service Routine -
7058                                    ;*        This routine executes in response to an interrupt caused by the DUT
7059                                    ;*        RX.DATA.AVAIL bit becoming active.  This routine reads characters from
7060                                    ;*        the DUT receive character FIFO and deposits them into the receive
7061                                    ;*        buffer in memory.  If the number of characters in the receive buffer
7062                                    ;*        exceeds a specified threshold, transmission is halted (by clearing all
7063                                    ;*        DUT TX.ENABLE bits) and if the receive buffer is full reception is
7064                                    ;*        halted (by disabling RX interrupts).  The routine exits if the receive
7065                                    ;*        buffer becomes full or if a character is read from the FIFO with the
7066                                    ;*        DATA.VALID bit clear.
7067                                    ;*
7068                                    ;* INPUTS:        RBUFA - Contains address of the DUT RX character FIFO.
7069                                    ;*                RXBCNT - RX buffer character count.
7070                                    ;*                RXBDTX - Equated to RX buffer level at which to disable TX.
7071                                    ;*                RXBEND - Label after end of the RX buffer area in memory.
7072                                    ;*                RXBFUL - Equated to the capacity of the RX buffer.
7073                                    ;*                RXBIPT - Pointer to next available input slot of RX buffer.
7074                                    ;*                RXBSTA - Label at start of RX buffer area in memory.
7075                                    ;*
7076                                    ;* OUTPUTS:       RXBIPT - Updated to point to next input slot of RX buffer.
7077                                    ;*                RXBCNT - RX buffer character count (Incremented).
7078                                    ;*                TXENBM - Map of previous DUT TX.ENABLE states.
7079                                    ;*                CARRY - "Success" flag (Set if buffer is not full).
7080                                    ;*
7081                                    ;* CALLING SEQUENCE:    Put the address of the label RXCHRS in the vector
7082                                    ;*                     location.
7083                                    ;*
7084                                    ;* COMMENTS:      If the RX buffer is full upon entry, this routine aborts the
7085                                    ;*                program.
7086                                    ;*
7087                                    ;* SUBORDINATE ROUTINES CALLED: RXIEO,TXDSBL.
7088                                    ;-- ***************************************************************************
7089
7090 027364 010246                     RXCHRS::      MOV      R2,-(SP)        ;SAVE CONTENTS OF GPR R2.
7091 027366 017702 152612              2$:    MOV      @RBUFA,R2       ;READ A CHARACTER FROM THE DUT RX FIFO.
7092 027372 100054                            BPL      60$             ;EXIT THE ROUTINE IF THE DATA.VALID BIT IS CLR.
7093
7094 027374 023727 002720 000100              CMP      RXBCNT,@RXBFUL  ;COMPARE BUFFER COUNT WITH BUFFER CAPACITY.
7095 027402 103402                            BLO      4$              ;SKIP ABORT IF BUFFER IS NOT FULL.
7096 027404 004737 022364                     JSR      PC,OOPS         ;ABORT, MUST BE A PROGRAM BUG.
7097 027410 010277 153302              4$:    MOV      R2,@RXBIPT      ;PUT THE CHAR IN THE BUFFER.
7098 027414 062737 000002 002716              ADD      #2,RXBIPT       ;UPDATE POINTER TO THE NEXT BUFFER SLOT.
7099 027422 023727 002716 003122              CMP      RXBIPT,@RXBEND  ;CHECK IF POINTER SHOULD WRAP AROUND.
7100 027430 103403                            BLO      6$              ;SKIP WRAPAROUND IF POINTER IS NOT AT END.
7101 027432 012737 002722 002716              MOV      @RXBSTA,RXBIPT  ;WRAP INPUT POINTER AROUND.
7102
7103 027440 005237 002720              6$:    INC      RXBCNT          ;COUNT THIS CHARACTER AS BEING IN THE BUFFER.
7104 027444 023727 002720 000030              CMP      RXBCNT,@RXBDTX  ;CHECK FOR BUFFER AT DISABLE TX LEVEL.
7105 027452 001016                            BNE      8$              ;SKIP DISABLING TX IF BUFFER LEVEL NOT CORRECT.
7106 027454 005737 002510                     TST      TXDBLF          ;CHECK STATE OF TX DISABLE FLAG.
7107 027460 100413                            BMI      8$              ;BRANCH IF TRANSMISSION ALREADY DISABLED.
7108 027462 010546                            MOV      R5,-(SP)        ;SAVE THE VALUE OF GPR R5.
7109 027464 012705 000377                     MOV      #MAPLNS,R5      ;SPECIFY THAT ALL LINES SHOULD BE AFFECTED.
7110 027470 004737 025552                     JSR      PC,TXDSBL       ;CLEAR THE TX ENABLES FOR ALL LINES.
7111 027474 010537 002250                     MOV      R5,TXENBM       ;SAVE PREVIOUS TX ENABLE STATES IN STORAGE.
```

```
7112 027500  012605                      MOV    (SP)+,R5          ;RESTORE GPR R5.
7113 027502  012737  100000  002510      MOV    #BIT15,TXDBLF     ;PREVENT TX FROM BEING DISABLED AGAIN.
7114
7115 027510  023727  002720  000100 8$:  CMP    RXBCNT,#RXBFUL    ;CHECK FOR BUFFER FULL CONDITION.
7116 027516  103723                       BLO    2$               ;LOOP TO READ ANOTHER CHAR IF BUFFER NOT FULL.
7117
7118 027520  004737  024650               JSR    PC,RXIE0         ;BUFFER IS FULL, DISABLE RX INTERRUPTS.
7119
7120 027524  012602                  60$: MOV    (SP)+,R2         ;RESTORE R2 TO ITS SAVED VALUE.
7121 027526  000002                       RTI
```

```
7123                                    .SBTTL  TRAP SERVICE ROUTINE              - TP4BRT -
7124                                    ;*********************************************************************
7125                                    ;*      Bus Time-out Trap (004 trap) Service Routine -
7126                                    ;*      This routine is used during the DMA ADDRESS TEST.
7127                                    ;*      It determines if the 004 trap was caused by an "expected" error or
7128                                    ;*      not by examining the return PC value on the stack.  If the trap is
7129                                    ;*      unexpected, this routine jumps to the normal Diagnostic Supervisor
7130                                    ;*      004 trap handling routine.
7131                                    ;*
7132                                    ;* INPUTS:        SP - Points to the PC where the trap occured.
7133                                    ;*                TRPAD2 - Label at the address where "expected" traps occur.
7134                                    ;*                TP4FLG - 004 trap flags.
7135                                    ;*
7136                                    ;* OUTPUTS:       TP4FLG - Bit 15 is set if "expected" trap occured.
7137                                    ;*
7138                                    ;* CALLING SEQUENCE:    Put address pointed to by TP4BRT in 004 vector.
7139                                    ;*                      Occurence of 004 trap vectors to this routine.
7140                                    ;*
7141                                    ;* COMMENTS:      Any 004 trap which occurs at an address other than that labeled
7142                                    ;*                TRPAD2 will be handled by the normal 004 trap service routine.
7143                                    ;*                This routine is used in conjunction with CKTRPB subroutine.
7144                                    ;*
7145                                    ;* SUBORDINATE ROUTINES CALLED: None.
7146                                    ;*********************************************************************
7147
7148 027530  021627  017216   TP4BRT:: CMP     (SP),@TRPAD2      ;COMPARE EXPECTED ADDR WITH TRAP RET PC.
7149 027534  001402                     BEQ     2$                ;IF THEY MATCH, CONTINUE THIS ROUTINE.
7150 027536  000177  152512             JMP     @TP4VEC           ;IF NOT, JUMP TO NORMAL 004 TRAP SERVICE RTN.
7151 027542  052737  100000   002256 2$: BIS    @BIT15,TP4FLG     ;SET THE 004 TRAP OCCURED FLAG.
7152 027550  000002                     RTI                       ;ALL DONE, GO BACK TO THE TEST.
```

```
7154                                    .SBTTL  GLOBAL TRAP SERVICE ROUTINE     - TP4RTN -
7155                                    ;***********************************************************************
7156                                    ;*      Bus Time-out Trap (004 trap) Service Routine -
7157                                    ;*      This routine is used during the Device Register Address Access Test.
7158                                    ;*      It determines if the 004 trap was caused by an "expected" error or
7159                                    ;*      not by examining the return PC value on the stack.  If the trap is
7160                                    ;*      unexpected, this routine jumps to the normal Diagnostic Supervisor
7161                                    ;*      004 trap handling routine.
7162                                    ;*
7163                                    ;* INPUTS:        SP - Points to the PC where the trap occured.
7164                                    ;*                ADRPTR - Label at the address where "expected" traps occur.
7165                                    ;*                TP4FLG - 004 trap flags.
7166                                    ;*
7167                                    ;* OUTPUTS:       TP4FLG - Bit 15 is set if "expected" trap occured.
7168                                    ;*
7169                                    ;* CALLING SEQUENCE:   Put address pointed to by TP4RTN in 004 vector.
7170                                    ;*                     Occurence of 004 trap vectors to this routine.
7171                                    ;*
7172                                    ;* COMMENTS:      Any 004 trap which occurs at an address other than that labeled
7173                                    ;*                ADRPTR will be handled by the normal 004 trap service routine.
7174                                    ;*
7175                                    ;* SUBORDINATE ROUTINES CALLED: None.
7176                                    ;***********************************************************************
7177
7178 027552  021627  017166    TP4RTN:: CMP    (SP),#ADRPTR     ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
7179 027556  001402               BEQ    2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
7180 027560  000177  152470       JMP    @TP4VEC         ;IF NOT,JUMP TO NORMAL 004 TRAP SERVICE RTN.
7181 027564  052737  100000  002256  2$:  BIS    #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
7182 027572  000002               RTI                    ;ALL DONE, GO BACK TO THE TEST.
```

```
7184                                    .SBTTL   INTERRUPT SERVICE ROUTINE        - TXDMA -
7185                                    ;++ ******************************************************************************
7186                                    ;*                  - DMA Transmit Interrupt Service Routine -
7187                                    ;*        This routine executes in response to an interrupt caused by the DUT
7188                                    ;*        TX.ACTION bit becoming active.  This routine initiates the TX of a
7189                                    ;*        new DMA buffer of characters or sets the TX Done Flag for the correct
7190                                    ;*        line if TX is complete on that line.
7191                                    ;*
7192                                    ;* INPUTS:        BITTBL - Label of table of words each with a bit set.
7193                                    ;*                CNCNTB - Base of # of chars to TX/RX table.
7194                                    ;*                CSRA - Contains the address of the DUT CSR.
7195                                    ;*                DPENDB - Base of the data pattern end table (entry per line).
7196                                    ;*                DPLENB - Base of the data pattern length table.
7197                                    ;*                IESTAT - Preserved states of the DUT interrupt enable bits.
7198                                    ;*                TXCNTB - Lable at base of the TX character counter table.
7199                                    ;*                TXPTRB - Label at base of the TX data pattern pointers table.
7200                                    ;*
7201                                    ;* OUTPUTS:       TXCNTx - Counters incremented for lines on which chars sent.
7202                                    ;*                TXDONF - TX done flags set for lines which have sent all chars.
7203                                    ;*                TXINTF - TX int flags (bit set if DMA.HO found set on line).
7204                                    ;*
7205                                    ;* CALLING SEQUENCE:    Put the address of the label TXDMA in the vector
7206                                    ;*                      location.
7207                                    ;*
7208                                    ;* COMMENTS:
7209                                    ;*
7210                                    ;* SUBORDINATE ROUTINES CALLED: DODMA.
7211                                    ;-- ******************************************************************************
7212
7213 027574                            TXDMA:: SAVE                             ;SAVE CONTENTS OF GPRS R0 THRU R5.
     027574  004537  005326                            JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
7214 027600  017701  152376                    MOV     @CSRA,R1                ;READ THE CONTENTS OF THE DUT CSR.
7215 027604  010100                            MOV     R1,R0                   ;SAVE INITIAL CONTENTS OF IND.ADR.REG FIELD.
7216 027606  000402                            BR      4$                      ;BRANCH TO SKIP DOUBL READING OF DUT CSR.
7217                                    ;+
7218                                    ; Read the contents of the DUT CSR.  This will clear the TX.ACTION CSR bit.
7219                                    ; If TX.ACTION is not set, exit this routine.
7220                                    ; Determine the line for which the TX.ACTION was set.
7221                                    ; Calculate an offset for use in accessing tables (2 times the line number).
7222                                    ; Get the bit map of this line.
7223                                    ;-
7224 027610  017701  152366            2$:     MOV     @CSRA,R1                ;READ THE CONTENTS OF THE DUT CSR.
7225 027614  100033                    4$:     BPL     60$                     ;EXIT ROUTINE IF TX.ACTION IS CLEAR.
7226 027616  000301                            SWAB    R1                      ;CALCULATE THE LINE NUMBER OF THE LINE WHICH IS
7227 027620  042701  177760                    BIC     #177760,R1              ;  ASSOCIATED WITH THE TX.ACTION.
7228 027624  010104                            MOV     R1,R4                   ;CALCULATE AN OFFSET FOR USE IN ACCESSING
7229 027626  006304                            ASL     R4                      ;  LINE COUNTER AND POINTER IN TABLES.
7230 027630  016405  002366                    MOV     BITTBL(R4),R5           ;GET THE BIT MAP OF THIS LINE.
7231                                    ;+
7232                                    ; Get the TX character counter for this line.
7233                                    ; If all the characters have been sent for this line:
7234                                    ;    Set the TX done flag for this line.
7235                                    ;    Don't send a char to the line (no more TX.ACTIONS on this line).
7236                                    ;    Loop to check the TX.ACTION for another line.
7237                                    ;-
7238 027634  026464  003504  003444            CMP     TXCNTB(R4),CHCNTB(R4)   ;COMPARE # CHARS SENT AND TX COUNT.
7239 027642  103403                            BLO     6$                      ;GO TO SEND A CHAR IF NOT ALL CHARS SENT.
```

```
7240 027644  050537  002504               BIS    R5,TXDONF      ;SET THIS LINE'S TX DONE FLAG.
7241 027650  000757                        BR     2$             ;LOOP TO CHECK TX.ACTION AGAIN.
7242                                 ;+
7243                                 ; Start the DMA of the next buffer (data pattern) on this line.
7244                                 ;    Get the data pattern length for this line.
7245                                 ;    Get the start address of the data pattern.
7246                                 ;-
7247 027652  016403  003204         6$:    MOV    DPLENB(R4),R3  ;PASS DATA PATTERN LENGTH FOR LINE TO DODMA.
7248 027656  016402  003344                MOV    TXPTRB(R4),R2  ;PASS THE TX START ADR TO DODMA.
7249                                   ;+
7250                                   ; Write DMA parameters to the DUT.
7251                                   ;-
7252 027662  004737  017610                JSR    PC,DODMA
7253 027666  103403                        BCS    8$             ;SKIP ERROR IF DODMA WAS SUCCESSFUL.
7254                                   ;+
7255                                   ; Set the proper bit of the TX interrupt flags to indicate the line error.
7256                                   ;-
7257 027670  050537  002252                BIS    R5,TXINTF      ;INDICATE THE ERROR.
7258 027674  000402                        BR     10$            ;SKIP UPDATING POINTERS AND COUNTERS.
7259                                   ;+
7260                                   ; Update the TX character for this line.
7261                                   ; Update the TX buffer pointer for this line.
7262                                   ;-
7263 027676  060364  003504         8$:    ADD    R3,TXCNTB(R4)  ;ADD THE DATA PAT LENGTH TO THE TX COUNT.
7264                                 ;+
7265                                 ; Loop to check the TX.ACTION bit for another line.
7266                                 ;-
7267 027702  000742                 10$:   BR     2$             ;LOOP BACK TO CHECK TX.ACTION BIT AGAIN.
7268
7269 027704  013701  002234         60$:   MOV    IESTAT,R1      ;GET THE PRESENT STATES OF TX.IE & RX.IE BITS.
7270 027710  042700  177760                BIC    #177760,R0     ;GET SAVED IND.ADR.REG FIELD BITS.
7271 027714  050001                         BIS    R0,R1          ;COMBINE IND.ADR.REG FIELD BITS WITH IE BITS.
7272 027716  010177  152260                MOV    R1,@CSRA       ;RESTORE THE DUT CSR IND.ADR.REG FIELD.
7273 027722                                PASS                  ;RESTORE GPRS.
     027722  004736                                JSR    PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
7274 027724  000002                        RTI
```

```
7276                                    .SBTTL   INTERRUPT SERVICE ROUTINE        - TXSCHR -
7277                                    ;++ ************************************************************************
7278                                    ;*        - Single Character Mode Transmit Interrupt Service Routine -
7279                                    ;*        This routine executes in response to an iterrupt caused by the DUT
7280                                    ;*        TX.ACTION bit becoming active.  This routine sends the next TX char
7281                                    ;*        or sets the TX Done Flag for the correct line if TX is complete on
7282                                    ;*        that line.
7283                                    ;*
7284                                    ;* INPUTS:        ACTLNS - Bit map of active DUT lines.
7285                                    ;*                BITTBL - Label of table of words each with a bit set.
7286                                    ;*                CHCNTB - Base of # of chars to TX/RX table.
7287                                    ;*                CSRA - Contains the address of the CSR.
7288                                    ;*                DPENDB - Base of the data pattern end table (entry per line).
7289                                    ;*                DPLENB - Base of the data pattern length table.
7290                                    ;*                IACBIT - Bit mask of inactive TX/RX bits.
7291                                    ;*                IBM - Inactive bits mask (reflecting bits per char).
7292                                    ;*                TXCHRA - Contains the address of the DUT TXCHAR register.
7293                                    ;*                TXCNTB - Label at base of TX character counters table.
7294                                    ;*                TXPTRB - Label at the base address of the TX pointers table.
7295                                    ;*
7296                                    ;* OUTPUTS:       CSR - DUT CSR IND.ADR.REG field is destroyed.
7297                                    ;*                TXCHAR - DUT TXCHARs have words written to them.
7298                                    ;*                TXCNTx - Counters incremented for lines on which chars sent.
7299                                    ;*                TXDONF - TX done flags set for lines which have sent all chars.
7300                                    ;*                TXPTRB - Each pointer in table points to next TX char for line.
7301                                    ;*
7302                                    ;* CALLING SEQUENCE:    Put the address of the label TXSCHR in the vector
7303                                    ;*                     location.
7304                                    ;*
7305                                    ;* COMMENTS:
7306                                    ;*
7307                                    ;* SUBORDINATE ROUTINES CALLED: None.
7308                                    ;-- ************************************************************************
7309
7310 027726                            TXSCHR:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
     027726  004537  005326                      JSR       R5,PREG05   ;CALL REGISTER SAVE SUBRT.
7311 027732  017701  152244                      MOV       @CSRA,R1    ;READ THE CONTENTS OF THE DUT CSR.
7312 027736  010105                              MOV       R1,R5       ;SAVE THE CONTENTS OF THE DUT IND.ADR.REG
7313 027740  042705  137660                      BIC       #137660,R5  ;  FIELD FOR RESTORATION BEFORE RETURN.
7314 027744  010546                              MOV       R5,-(SP)    ;SAVE THE DUT IN.ADD FIELD ON THE STACK.
7315 027746  005005                              CLR       R5          ;CLEAR CHARACTER TRANSMISSION COUNTER.
7316 027750  005701                              TST       R1          ;SET FLAG FOR TEST AFTER THE FOLLOWING BRANCH.
7317 027752  000402                              BR        4$          ;GO HANDLE THE LINE THAT GOT THE TX.ACTION.
7318                                    ;+
7319                                    ; Read the contents of the DUT CSR.  This will clear the TX.ACTION CSR bit.
7320                                    ; If TX.ACTION is not set, exit this routine.
7321                                    ; Determine the line for which the TX.ACTION was set.
7322                                    ; Calculate an offset for use in accessing tables (2 times the line number).
7323                                    ; Determine the states of the DUT CSR interrupt enable bits.
7324                                    ;-
7325 027754  017701  152222            2$:       MOV       @CSRA,R1    ;READ THE CONTENTS OF THE DUT CSR.
7326 027760  100051                     4$:       BPL       60$         ;EXIT ROUTINE IF TX.ACTION IS CLEAR.
7327 027762  010102                              MOV       R1,R2       ;CALCULATE THE LINE NUMBER
7328 027764  000302                              SWAB      R2          ;  OF THE LINE WHICH IS
7329 027766  042702  177760                      BIC       #177760,R2  ;  ASSOCIATED WITH THE TX.ACTION.
7330 027772  010203                              MOV       R2,R3       ;CALCULATE AN OFFSET FOR USE IN ACCESSING
7331 027774  006303                              ASL       R3          ;  LINE COUNTER AND POINTER IN TABLES.
```

```
7332 027776 042701 137677                 BIC     #137677,R1        ;GET BIT MASK OF INTERRUPT ENABLE STATES.
7333                                  ;+
7334                                  ; Get the TX character counter for this line.
7335                                  ; If all the characters have been sent for this line:
7336                                  ;     Set the TX done flag for this line.
7337                                  ;     Don't send a char to the line (no more TX.ACTIONS on this line).
7338                                  ;     Loop to check the TX.ACTION for another line.
7339                                  ;-
7340 030002 026363 003504 003444          CMP     TXCNTB(R3),CHCNTB(R3)  ;COMPARE TX CHAR COUNT AND # TO TX.
7341 030010 103404                         BLO     6$                ;GO TO SEND A CHAR IF NOT ALL CHARS SENT.
7342 030012 056337 002366 002504          BIS     BITTBL(R3),TXDONF     ;SET THIS LINE'S TX DONE FLAG.
7343 030020 000757                         BR      4$                ;LOOP TO CHECK TX.ACTION AGAIN.
7344                                  ;+
7345                                  ; Send the next char to the specified line.
7346                                  ;     Set up the IND.ADR.REG field of the DUT CSR using the previously read
7347                                  ;       states of the interrupt enable bits.
7348                                  ;     Fetch the correct character from the data pattern.
7349                                  ;     Update the data pattern pointer for this line using wraparound.
7350                                  ;     Mask out inactive data bits and send the character.
7351                                  ;     Count the character on the TX char counter for the line.
7352                                  ;     Decrement the FIFO slack count to reserve room for this char in the FIFO.
7353                                  ;     Exit if a maximum of 8 characters have been transmitted, ie.give reception
7354                                  ;     a chance to remove characters from the FIFO.
7355                                  ;-
7356 030022 050201                  6$:    BIS     R2,R1             ;SET UP THE IND.ADR.REG FIELD OF THE DUT CSR
7357 030024 010177 152152                  MOV     R1,@CSRA          ;   WITHOUT AFFECTING THE INTERRUPT ENABLES.
7358 030030 016304 003344                  MOV     TXPTRB(R3),R4     ;FETCH THE TX POINTER FOR THIS LINE.
7359 030034 112400                         MOVB    (R4)+,R0          ;GET THE NEXT CHAR FOR THIS LINE.
7360 030036 020463 003144                  CMP     R4,DPENDB(R3)     ;COMPARE POINTER WITH END OF DATA PATTERN.
7361 030042 103402                         BLO     8$                ;SKIP RESETTING OF POINTER IF NOT PAST END.
7362 030044 166304 003204                  SUB     DPLENB(R3),R4     ;WRAP POINTER AROUND TO BEGINNING OF PATTERN.
7363 030050 010463 003344          8$:     MOV     R4,TXPTRB(R3)     ;UPDATE THE TX POINTER FOR THIS LINE.
7364 030054 043700 002226                  BIC     IBM,R0            ;CLEAR UNUSED BITS OF THE TX CHAR WORD.
7365 030060 052700 100000                  BIS     #BIT15,R0         ;SET THE TX.DATA.VALID BIT OF TX CHAR WORD.
7366 030064 010077 152114                  MOV     R0,@TXCHA         ;SEND THE CHAR TO THE DUT.
7367 030070 005205                         INC     R5                ;INCREMENT TX CHAR COUNT.
7368 030072 005263 003504                  INC     TXCNTB(R3)        ;INCREMENT THE TX CHAR COUNT FOR THIS LINE.
7369                                  ;+
7370                                  ; Loop to check the TX.ACTION bit for another line.
7371                                  ;-
7372 030076 020527 000010                  CMP     R5,#NUMLNS        ;CHECK IF MAX NUMBER OF CHAR HAVE BEEN TX'D.
7373 030102 103724                         BLO     2$                ;LOOP BACK TO CHECK TX.ACTION BIT AGAIN.
7374 030104 012677 152072          60$:    MOV     (SP)+,@CSRA       ;RESTORE THE IND.ADR.REG FIELD OF THE DUT CSR.
7375 030110                                 PASS                      ;RESTORE GPRS.
     030110 004736                         JSR     PC,@(SP)+                    ;RETURN TO PREGO5 SUBRT.
7376 030112 000002                         RTI
```

```
7378
7379                            ;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
$$
7380                            ;
7381                            ;            VDHC.RPT
7382                            ;
7383                            ;$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
7384
7385
7386
7387                            .SBTTL   REPORT CODING SECTION
7388
7389                            ;++
7390                            ; THE REPORT CODING SECTION CONTAINS THE
7391                            ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
7392                            ;--
7393
7394 030114                             BGNRPT
     030114                                                                 L$RPT::
7395
7396 030114                             EXIT    RPT
     030114  000167                                                                 .WORD    J$JMP
     030116  000000                                                                 .WORD    L10016-2-.
7397
7398                                     .EVEN
7399
7400 030120                             ENDRPT
     030120                                                                 L10016:
     030120  104425                                                                 TRAP     C$RPT
```

PROTECTION TABLE

```
7402                                    .SBTTL  PROTECTION TABLE
7403
7404                                    ;####################################################################
7405                                    ;
7406                                    ;                 FVTSKL4.P11
7407                                    ;
7408                                    ;####################################################################
7409
7410
7411
7412                                    ;++
7413                                    ; THIS TABLE IS USED BY THE RUNTIME SERVICES
7414                                    ; TO PROTECT THE LOAD MEDIA.
7415                                    ;--
7416
7417 030122                            BGNPROT
     030122                                                            L$PROT::
7418
7419 030122  177777                        -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
7420 030124  177777                        -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
7421 030126  177777                        -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER
7422
7423 030130                            ENDPROT
7424
```

```
7439
7440                              ;**************************************************************************
7441                              ;
7442                              ;               VDHC.INI
7443                              ;
7444                              ;**************************************************************************
7445
7446
7447
7448                              .SBTTL  INITIALIZE SECTION
7449                              ;++
7450                              ;********************************************************************************
7451                              ;*      This section contains the code which is performed at the beginning of
7452                              ;*      each pass or after a continue command.
7453                              ;*      This code performs the following actions:
7454                              ;*
7455                              ;*      Moves the information held in the hardware P-table into the global
7456                              ;*      data area.
7457                              ;*
7458                              ;********************************************************************************
7459                              ;--
7460 030130                              BGNINIT
     030130                                                                              L$INIT::
7461                              ;SEE IF PROGRAM JUST STARTED, BR IF YES
7462 030130                              READEF  #EF.START
     030130  012700  000040                                                      MOV     #EF.START,R0
     030134  104447                                                              TRAP    C$REFG
7463 030136                              BCOMPLETE       NEWSTA
     030136  103416                                                              BCS     NEWSTA
7464                              ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
7465 030140                              READEF  #EF.RESTART
     030140  012700  000037                                                      MOV     #EF.RESTART,R0
     030144  104447                                                              TRAP    C$REFG
7466 030146                              BCOMPLETE       NEWRES
     030146  103555                                                              BCS     NEWRES
7467                              ;SEE IF THIS IS A NEW PASS, BR IF YES
7468 030150                              READEF  #EF.NEW
     030150  012700  000035                                                      MOV     #EF.NEW,R0
     030154  104447                                                              TRAP    C$REFG
7469 030156                              BCOMPLETE       NEWPAS
     030156  103554                                                              BCS     NEWPAS
7470                              ;SEE IF PROGRAM WAS JUST CONTINUED
7471 030160                              READEF  #EF.CONTINUE
     030160  012700  000036                                                      MOV     #EF.CONTINUE,R0
     030164  104447                                                              TRAP    C$REFG
7472 030166                              BNCOMPLETE      GETPRM
     030166  103160                                                              BCC     GETPRM
7473 030170  000137  030750      NEWSTA:  JMP     ENDIT
7474 030174                      NEWSTA:
7475 030174                              BRESET                          ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
     030174  104433                                                              TRAP    C$RESET
7476                              ;+
7477                              ; Set up for Line Time Clock interrupts.
7478                              ;-
7479 030176                              CLOCK   L,R1                    ;GET THE CLOCK PARAMETERS.
     030176  012700  000114                                                      MOV     #'L,R0
     030202  104462                                                              TRAP    C$CLCK
```

INITIALIZE SECTION

```
        030204  010001                                              MOV     R0,R1
7480 030206  012137  002272              MOV     (R1)+,CLKCSR       ;STORE CLOCK CSR ADDRESS.
7481 030212  012137  002274              MOV     (R1)+,CLKBRL       ;STORE CLOCK BUS REQ INT LEVEL.
7482 030216  012137  002276              MOV     (R1)+,CLKVEC       ;STORE CLOCK INTERRUPT VECTOR.
7483 030222  012137  002300              MOV     (R1)+,CLKHRZ       ;STORE CLOCK FREQUENCY.
7484 030226  023727  002300 000062       CMP     CLKHRZ,#50.        ;TEST FOR 50HZ LINE FREQUENCY.
7485 030234  001004                       BNE     2$                ;BRANCH IF CLOCK IS NOT 50HZ.
7486 030236  012737  000024 002312       MOV     #20.,MSTICK        ;INDICATE 20MS PER CLOCK TICK.
7487 030244  000403                       BR      4$
7488 030246  012737  000021 002312  2$:  MOV     #17.,MSTICK        ;INDICATE 17 MS PER CLOCK TICK.
7489 030254                          4$:  SETVEC  CLKVEC,#CLKINT,#PRI06  ;INITIALIZE CLOCK INTERRUPT VECTOR.
        030254  012746  000300                                      MOV     #PRI06,-(SP)
        030260  012746  027314                                      MOV     #CLKINT,-(SP)
        030264  013746  002276                                      MOV     CLKVEC,-(SP)
        030270  012746  000003                                      MOV     #3,-(SP)
        030274  104437                                              TRAP    C$SVEC
        030276  062706  000010                                      ADD     #10,SP
7490 030302  013700  002300              MOV     CLKHRZ,R0          ;INITIALIZE THE BREAK COUNT
7491 030306  006200                       ASR     R0                ; TO CAUSE A BREAK
7492 030310  010037  002310              MOV     R0,BCOUNT          ; EVERY 1/2 SECOND.
7493 030314  106427  000240              MTPS    #PRI05             ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
7494                              ;+
7495                              ; Enable the Line Time Clock (LTC) checking to make sure that the CSR
7496                              ; is accessable.
7497                              ;    First set up to catch any 004 traps which occur:
7498                              ;-
7499 030320  013737  000004 002254       MOV     @#4,TP4VEC         ;SAVE THE EXISTING 004 TRAP VECTOR.
7500 030326  012737  027552 000004       MOV     #TP4RTN,@#4        ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
7501                              ;+
7502                              ;    Enable LTC checking for 004 trap in case CSR is not there.
7503                              ;-
7504 030334  005037  002256              CLR     TP4FLG             ;CLEAR THE 004 TRAP FLAG.
7505 030340  012737  000100 002240       MOV     #BIT6,WORD1        ;SET UP TO SET BIT6 OF THE LTC CSR.
7506 030346  012700  002240              MOV     #WORD1,R0          ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
7507 030352  013701  002272              MOV     CLKCSR,R1          ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
7508 030356  004737  017154              JSR     PC,CKTRAP          ;MOVE AND CHECK FOR TRAP.
7509 030362  013737  002254 000004       MOV     TP4VEC,@#4         ;RESTORE THE NORMAL 004 TRAP VECTOR.
7510 030370  103403                       BCS     6$                ;IF NO TRAP, LTC IS THERE SO CONTINUE.
7511 030372  005037  002300              CLR     CLKHRZ             ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
7512 030376  000402                       BR      8$                ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
7513                              ;+
7514                              ; Calibrate the DELAY routine milli-second delay count value.
7515                              ;-
7516 030400  004737  015774  6$:  JSR     PC,CALMSL
7517                              ;+
7518                              ; Check for Memmory Management present on this machine.
7519                              ; If MEM MGT is present, disable it.
7520                              ;-
7521 030404  013737  000004 002254  8$:  MOV     @#4,TP4VEC         ;SAVE THE EXISTING 004 TRAP VECTOR.
7522 030412  012737  027552 000004       MOV     #TP4RTN,@#4        ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
7523 030420  005037  002256              CLR     TP4FLG             ;CLEAR THE 004 TRAP FLAG.
7524 030424  005037  002240              CLR     WORD1              ;PREPARE TO CLEAR THE MEM MGT SRO REGISTER.
7525 030430  012700  002240              MOV     #WORD1,R0          ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
7526 030434  013701  002316              MOV     MMSRO,R1           ;SELECT MEM MGT SRO REGISTER AS DESTINATION.
7527 030440  005037  002322              CLR     MMPRES             ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
7528 030444  005037  002324              CLR     MMENAB             ;INDICATE MEM MGT IS NOT ENABLED.
7529 030450  004737  017154              JSR     PC,CKTRAP          ;CLEAR THE MEM MCT SRO REG AND CHECK FOR TRAP.
```

```
7530 030454 013737 002254 000004        MOV     TP4VEC,@#4      ;RESTORE THE NORMAL 004 TRAP VECTOR.
7531 030462 103003                       BCC     10$             ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
7532 030464 012737 000001 002322         MOV     #1,MMPRES       ;INDICATE THAT MEM MGT IS PRESENT.
7533 030472 005037 002236        10$:    CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
7534 030476 000137 030510                JMP     NEWPAS          ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
7535
7536 030502                      NEWRES: BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
     030502 104433                                                        TRAP    C$RESET
7537 030504 005037 002236                CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
7538
7539 030510                      NEWPAS:
7540 030510 012737 177777 002200         MOV     #-1,UNITN       ;RESET LOGICAL DEVICE TO -1
7541                              ;+
7542                              ; Increment the pass counter, correct for any overflow.
7543                              ;  This counter is used in the Rom version test.
7544                              ;-
7545 030516 005237 002236                INC     PASCNT          ;INCREMENT THE PASS COUNTER.
7546 030522 001002                       BNE     GETPRM          ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
7547 030524 005337 002236                DEC     PASCNT          ;SET PASS COUNT TO 177777 OCTAL.
7548
7549                              ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
7550 030530                      GETPRM:
7551 030530 005237 002200                INC     UNITN           ;INCREMENT LOGICAL DEVICE NUMBER
7552 030534 023737 002200 002012         CMP     UNITN,L$UNIT    ;SEE IF MAXIMUM UNIT NO. EXCEEDED
7553 030542 002362                       BGE     NEWPAS          ;BR IF YES
7554
7555 030544                              GPHARD  UNITN,R1        ;GET P-TABLE POINTER INTO R1
     030544 013700 002200                                                MOV     UNITN,R0
     030550 104442                                                       TRAP    C$GPHRD
     030552 010001                                                       MOV     R0,R1
7556 030554                              BCOMPLETE    30$        ;BR IF DEVICE AVAILABLE
     030554 103401                                                       BCS     30$
7557 030556 000764                       BR      GETPRM          ;SKIP THIS DEVICE
7558
7559
7560                              ;*********** HARDWARE PARAMETER MOVING CODE **************
7561 030560 012137 002202        30$:    MOV     (R1)+,CSRA      ;STORE DHV11-M CSR ADDRESS IN DEV.REG.ADDRESS TABLE
7562 030564 012102                       MOV     (R1)+,R2        ;GET THE RX INTERRUPT VECTOR ADDRESS.
7563 030566 010237 002170                MOV     R2,RXVECA       ;STORE RX INT VECTOR ADDRESS.
7564 030572 062702 000004                ADD     #4,R2           ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
7565 030576 010237 002172                MOV     R2,TXVECA       ;STORE TX INT VECTOR ADDRESS.
7566 030602 012137 002174                MOV     (R1)+,ACTLNS    ;STORE DHV11-M ACTIVE LINE BIT MAP
7567 030606 012702 000377                MOV     #MAPLNS,R2      ;GET THE BIT MAP FOR ALL LINES.
7568 030612 005102                       COM     R2              ;GET A BIT MAP OF NON-EXISTANT LINES.
7569 030614 040237 002174                BIC     R2,ACTLNS       ;CLEAR NON-EXISTANT LINES FROM ACTLNS.
7570 030620 112137 002176                MOVB    (R1)+,LOPBCK    ;STORE DHV11-M LOOPBACK MODE
7571 030624 112137 002177                MOVB    (R1)+,BRLEVL    ;STORE DHV11-M INTERRUPT BUS REQUEST LEVEL
7572                              ;+
7573                              ;     CALCULTATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
7574                              ;     DEVICE REGISTER ADDRESS TABLE.
7575                              ;-
7576 030630 013701 002202                MOV     CSRA,R1         ;COPY CSR ADDRESS
7577 030634 005201                       INC     R1              ;INCREMENT CSR ADDRESS
7578 030636 005201                       INC     R1              ; COPY BY 2.
7579 030640 012703 000007                MOV     #7,R3           ;SET UP REGISTER COUNT
7580 030644 012702 002204                MOV     #RBUFA,R2       ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
7581 030650 010122                12$:    MOV     R1,(R2)+        ;STORE REGISTER ADDRESS IN TABLE
```

```
7582 030652 005201                    INC     R1              ;INCREMENT REGISTER ADDRESS
7583 030654 005201                    INC     R1              ; BY 2,FOR THE NEXT DEVICE REGISTER.
7584 030656 005303                    DEC     R3              ;DECREMENT REGISTER COUNT
7585 030660 001373                    BNE     12$             ;LOOP IF NOT DONE
7586
7587                            ;+
7588                            ; Initialise the BMP code queue.
7589                            ;-
7590 030662 012700 002514            MOV     #BMPCQB,R0       ;GET THE START ADDRESS OF THE QUEUE.
7591 030666 012701 002714            MOV     #BMPCQE,R1       ;GET THE END ADDRESS OF THE QUEUE.
7592 030672 010037 002512            MOV     R0,BMPCQP        ;SET THE POINTER TO THE START OF THE QUEUE.
7593 030676 005020            14$:    CLR     (R0)+            ;CLEAR OUT THE CONTENTS OF THE QUEUE.
7594 030700 020001                    CMP     R0,R1            ;CHECK IF END OF QUEUE HAS BEEN REACHED.
7595 030702 103775                    BLO     14$              ;LOOP IF NOT ALL DONE.
7596                            ;+
7597                            ; Report the Unit number if the software P-table question was answered YES,
7598                            ;  and the maximum unit number is greater than 1.
7599                            ;-
7600 030704 032737 000020 002164      BIT     #BIT4,OPTION     ;CHECK IF THE QUESTION WAS ANSWERED YES.
7601 030712 001416                    BEQ     16$              ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
7602 030714 023727 002012 000001      CMP     L$UNIT,#1        ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
7603 030722 003412                    BLE     16$              ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
7604 030724                           PRINTF  #MFUNIT,UNITN    ;REPORT UNIT NUMBER.
     030724 013746 002200                                                          MOV     UNITN,-(SP)
     030730 012746 005430                                                          MOV     #MFUNIT,-(SP)
     030734 012746 000002                                                          MOV     #2,-(SP)
     030740 010600                                                                 MOV     SP,R0
     030742 104417                                                                 TRAP    C$PNTF
     030744 062706 000006                                                          ADD     #6,SP
7605 030750                    16$:
7606
7607 030750                    ENDIT:
7608                            ;+
7609                            ; Set the processor priority to disable all but LTC interrupts.
7610                            ;-
7611 030750 106427 000240            MTPS    #PRIO5                   ;SET PROCESSOR PRIORITY TO 5.
7612                            ;+
7613                            ;   Enable Line Time Clock if one is available.
7614                            ;-
7615 030754 005737 002300            TST     CLKHRZ           ;CHECK FOR A LTC BEING PRESENT.
7616 030760 001403                    BEQ     18$              ;LTC PRESENT? NO, SKIP LTC ENABLE.
7617 030762 012777 000100 151302      MOV     #BIT6,@CLKCSR    ;YES, ENABLE THE LTC.
7618 030770                    18$:
7619
7620 030770                    ENDINIT
     030770                                                             L10020:
     030770 104411                                                         TRAP    C$INIT
7621
7622        000000            TNUM == 0                ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.
```

```
7625                    ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7626                    ;
7627                    ;                VDHC.ATD
7628                    ;
7629                    ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7631
7632
7633                    .SBTTL   AUTODROP SECTION
7634
7635
7636                    ;++
7637                    ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
7638                    ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
7639                    ; SEE IF THEY WILL RESPOND.  THOSE THAT DON'T ARE IMMEDIATELY
7640                    ; DROPPED FROM TESTING.
7641                    ;--
7642
7643 030772                     BGNAUTO
     030772                                                              L$AUTO::
7644
7651
7652 030772                     ENDAUTO
     030772                                                              L10021:
     030772  104461                                                         TRAP    C$AUTO
```

```
7654
7655                          ;############################################################
7656                          ;
7657                          ;              VDHC.CUC
7658                          ;
7659                          ;############################################################
7660
7661
7662
7663                          .SBTTL   CLEANUP CODING SECTION
7664
7665                          ;++
7666                          ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
7667                          ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
7668                          ;--
7669
7670 030774                          BGNCLN
     030774                                                              L$CLEAN::
7671
7680
7681 030774  005737  002222          TST     CTRLCF          ;DID WE GET HERE BY CTRL-C FROM TEST?
7682 031000  001401                  BEQ     2$              ;CTRL-C FROM TEST?  NO, SKIP BUS RESET.
7683 031002                          BRESET                  ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
     031002  104433                                                  TRAP    C$RESET
7684 031004                   2$:
7685 031004  005737  002300          TST CLKHRZ              ;IS CLOCK ENABLED
7686 031010  001402                  BEQ 3$                  ;IF NO BRANCH
7687 031012  005077  151254          CLR @CLKCSR            ;TURN OFF CLOCK
7688 031016                   3$:
7689 031016                          EXIT    CLN
     031016  104432                                                  TRAP    C$EXIT
     031020  000002                                                  .WORD   L10022-.
7690
7702
7703                          .EVEN
7704
7705 031022                          ENDCLN
     031022                                                  L10022:
     031022  104412                                                  TRAP    C$CLEAN
```

```
7707
7708                          ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7709                          ;
7710                          ;            VDHC.DRP
7711                          ;
7712                          ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7713
7714
7715
7716                          .SBTTL   DROP UNIT SECTION
7717
7718                          ;**
7719                          ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
7720                          ; TO NO LONGER BE TESTED.
7721                          ;--
7722
7723 031024                          BGNDU
     031024                                                                   L$DU::
7724
7725                          ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7726                          ;       INSERT DROP CODE HERE.   THIS CODE WILL BE EXECUTED AFTER
7727                          ;       A "DROP" COMMAND OR A "DODU" MACRO EXECUTION.   THE PURPOSE
7728                          ;       OF THIS CODE IS TO DO ANY NECESSARY HOUSEKEEPING AFTER A
7729                          ;       UNIT HAS BEEN DROPPED.   THIS SECTION IS OPTIONAL.
7730                          ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7731 031024                          PRINTF  #DROP,R0          ;REPORT UNIT THAT HAS BEEN DROPPED.
     031024   010046                                                          MOV    R0,-(SP)
     031026   012746   031050                                                 MOV    #DROP,-(SP)
     031032   012746   000002                                                 MOV    #2,-(SP)
     031036   010600                                                          MOV    SP,R0
     031040   104417                                                          TRAP   C$PNTF
     031042   062706   000006                                                 ADD    #6,SP
7732 031046   000427                  BR      EDROP            ;BRANCH AROUND THE MESSAGE.
7733
7734                          .NLIST  BEX
7735 031050   045  101  040  DROP:   .ASCIZ/%A UNIT%D6%A DROPPED FROM FURTHER TESTING.%N/
7736                          .EVEN
7737 031126                  EDROP:
7738
7739 031126                          EXIT    DU
     031126   000167                                                          .WORD  J$JMP
     031130   000000                                                          .WORD  L10023-2-.
7740
7741
7742 031132                          ENDDU
     031132                                                                   L10023:
     031132   104453                                                          TRAP   C$DU
```

```
7744
7745                    ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7746                    ;
7747                    ;                    VDHC.ADD
7748                    ;
7749                    ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7750
7751
7752
7753                    .SBTTL   ADD UNIT SECTION
7754
7755                    ;++
7756                    ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
7757                    ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
7758                    ;  TO THE TEST CYCLE.
7759                    ;--
7760
7761 031134                    BGNAU
     031134                                                            L$AU::
7762
7763                    ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7764                    ;      INSERT ADD CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
7765                    ;      AN "ADD" COMMAND.  THE PURPOSE OF THIS CODE IS TO DO ANY
7766                    ;      HOUSEKEEPING THAT MAY BE NECESSARY AFTER A UNIT HAS BEEN ADDED.
7767                    ;      THIS SECTION IS OPTIONAL.
7768                    ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
7769
7770 031134                    EXIT    AU
     031134  000167                                                  .WORD   J$JMP
     031136  000000                                                  .WORD   L10024-2-.
7771
7772
7773                    .EVEN
7774
7775 031140                    ENDAU
     031140                                                          L10024:
     031140  104452                                                  TRAP    C$AU
7776
```

```
 7778                                   .SBTTL  HARDWARE TEST            - ADRA -
 7779                                   ;++
 7780                                   ;*******************************************************************
 7781                                   ;*                     - REGISTER ADDRESS TEST -
 7782                                   ;*
 7783                                   ;*      This test verifies that the Q-bus can read and write to the DHV11-M
 7784                                   ;*      device registers.  If the DHV11-M does not respond to the access
 7785                                   ;*      attempts (If the DHV11-M is at the wrong address, for example) the
 7786                                   ;*      004 bus time-out trap is detected by this routine and an error
 7787                                   ;*      is reported.
 7788                                   ;*
 7789                                   ;*******************************  *********************************************
 7790                                   ;--
 7791
 7792 031142                           BGNTST
      031142                                                                      T1::
 7793          000001                   TNUM == TNUM + 1         ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 7794 031142   012737  000001  002224   MOV     #TNUM,TSTNUM     ;SET UP THE TEST NUMBER.          (1)
 7795 031150   012737  177777  002222   MOV     #-1,CTRLCF       ;INDICATE THAT WE ARE IN A TEST.
 7796                                   ;+
 7797                                   ; Set up to catch any 004 traps which occur:
 7798                                   ;-
 7799 031156   013737  000004  002254   MOV     @#4,TP4VEC.      ;SAVE THE EXISTING 004 TRAP VECTOR.
 7800 031164   012737  027552  000004   MOV     #TP4RTN,@#4      ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
 7801 031172   005005                   CLR     R5               ;CLEAR THE ERROR FLAGS.
 7802
 7803                                   ;+
 7804                                   ; Set up for the initial iteration of the test loop:
 7805                                   ;-
 7806 031174   005004                   CLR     R4               ;CLEAR THE LINE COUNTER.
 7807
 7808                                   ;+
 7809                                   ; Here begins the loop to test the registers for a line.
 7810                                   ;     First test the CSR and set the IND.ADR.REG (I.A.R) field.
 7811                                   ;-
 7812 031176   005037  002256   2$:     CLR     TP4FLG           ;CLEAR THE 004 TRAP FLAG.
 7813 031202   013700  002202           MOV     CSRA,R0          ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
 7814 031206   012701  031422           MOV     #52$,R1          ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
 7815 031212   004737  017154           JSR     PC,CKTRAP        ;MOVE AND CHECK FOR TRAP.
 7816 031216   103402                   BCS     4$               ;IF NO TRAP, BYPASS ERROR.
 7817 031220   052705  100001           BIS     #100001,R5       ;SET FATAL READ ERROR FLAGS.
 7818 031224   042737  000017  031422 4$: BIC   #17,52$          ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
 7819 031232   050437  031422           BIS     R4,52$           ;OR IN THE LINE COUNTER TO THE I.A.R FIELD.
 7820 031236   010100                   MOV     R1,R0            ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
 7821 031244   013701  002202           MOV     CSRA,R1          ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
 7822 031244   004737  017154           JSR     PC,CKTRAP        ;MOVE AND CHECK FOR TRAP.
 7823 031250   103403                   BCS     6$               ;IF NO TRAP, BYPASS ERROR.
 7824 031252   052705  100002           BIS     #100002,R5       ;SET FATAL WRITE ERROR FLAGS.
 7825 031256   000440                   BR      40$              ;EXIT AND REPORT FATAL ERROR.
 7826                                   ;+
 7827                                   ;     Now, we test each register for this line.
 7828                                   ;-
 7829 031260   012702  000010   6$:     MOV     #10,R2           ;INIT REGISTER COUNTER TO 8.
 7830 031264   013737  002202  031420   MOV     CSRA,50$              ;INITIALIZE THE REGISTER POINTER.
 7831 031272   012700  031420   8$:     MOV     #50$,R0          ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
 7832 031276   012701  031422           MOV     #52$,R1          ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
 7833 031302   004737  017154           JSR     PC,CKTRAP        ;PERFORM THE MOVE, CHECK FOR TRAP.
```

```
7834 031306  103402                      BCS     10$          ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
7835 031310  052705  100001              BIS     #100001,R5   ;SET FATAL READ ERROR FLAGS.
7836 G31314  010100             10$:     MOV     R1,RO        ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
7837 031316  012701  031420              MOV     #50$,R1      ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE.
7838 031322  004737  017154              JSR     PC,CKTRAP    ;PERFORM THE MOVE, CHECK FOR TRAP.
7839 031326  103402                      BCS     12$          ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
7840 031330  052705  100002              BIS     #100002,R5   ;SET FATAL WRITE ERROR FLAGS.
7841 031334  005237  031420     12$:     INC     50$          ;INCREMENT THE REGISTER
7842 031340  005237  031420              INC     50$          ; POINTER BY 2.
7843 031344  005302                      DEC     R2           ;COUNT THE REGISTER.
7844 031346  001351                      BNE     8$           ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
7845
7846
7847                             ;+
7848                             ; Now we set up to test the next line, or to exit if we are done.
                                 ;-
7849 031350  005204                      INC     R4           ;INCREMENT THE LINE COUNTER.
7850 031352  020427  000010              CMP     R4,#NUMLNS   ;COMPARE LINE COUNTER AGAINST NUMBER OF LINES.
7851 031356  002707                      BLT     2$           ;LOOP TO TEST THE NEXT LINE IF WE'RE NOT DONE.
7852
7853                             ;+
7854                             ; Done checking device register addresses.
7855                             ;    Report any errors and exit.
7856                             ;-
7857 031360  013737  002254  000004  40$: MOV    TP4VEC,@#4   ;RESTORE THE NORMAL 004 TRAP VECTOR.
7858 031366  005705                      TST     R5           ;CHECK THE ERROR FLAGS.
7859 031370  100015                      BPL     60$          ;EXIT ROUTINE IF NO ERRORS.
7860
7861                             ; REPORT "DEVICE REGISTER ACCESS ERRORS"
7862 031372                              ERRDF   101,EM0103,ER0101;            >>>>> ERROR #101 <<<<<.
     031372  104455                                                                       TRAP    C$ERDF
     031374  000145                                                                       .WORD   101
     031376  010072                                                                       .WORD   EM0103
     031400  013562                                                                       .WORD   ER0101
7863
7864 031402                              DODU    UNITN        ;DROP THIS UNIT FROM FUTHER TESTING.
     031402  013700  002200                                                               MOV     UNITN,RO
     031406  104451                                                                       TRAP    C$DODU
7865 031410  005037  002222              CLR     CTRLCF       ;INDICATE NO CTRL-C ABORT FROM TEST.
7866 031414                              DOCLN                ;ABORT THIS SUB PASS.
     031414  104444                                                                       TRAP    C$DCLN
7867 031416  000402                      BR      60$          ;
7868                             ;+
7869                             ; Local storage.
7870                             ;-
7871 031420  000000             50$:     .WORD   0            ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
7872 031422  000000             52$:     .WORD   0            ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
7873 031424  005037  002222     60$:     CLR     CTRLCF       ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7874 031430                              ENDTST
     031430                                                           L10025:
     031430  104401                                                                       TRAP    C$ETST
```

```
 7876                                        .SBTTL  HARDWARE TEST              - FRMERR -
 7877                                        ;+·********************************************************************
 7878                                        ;*                    - FRAMING ERROR GENERATION TEST -
 7879                                        ;*
 7880                                        ;*      This test is used to verify the framing error detection capabilities
 7881                                        ;*      of the DHV11-M.
 7882                                        ;*      When in staggared loopback mode, characters are transmitted from
 7883                                        ;*      one group of lines at 8 bits/char,and received by the other group
 7884                                        ;*      at 5 bits/char.This will generate a framing error for each character.
 7885                                        ;*      This test will only execute if the staggared loopback mode is selected.
 7886                                        ;*      The special staggared loopback BERG connector must be fitted.
 7887                                        ;*      The active lines bit mask is used to indicate which lines have been
 7888                                        ;*      removed from further testing.
 7889                                        ;*
 7890                                        ;--********************************************************************
 7891 031432                                        BGNTST
      031432                                                                        T2::
 7892                                        ;+
 7893                                        ; Execute this test in  staggared loopback mode only.
 7894                                        ;-
 7895 031432 123727 002176 000002                   CMPB    LOPBCK,#2         ;CHECK MODE SELECTED.
 7896 031440 001154                                 BNE     60$               ;EXIT IF STAGGERD LOOPBACK MODE NOT SELECTED.
 7897 031442 012737 177777 002222                   MOV     #-1,CTRLCF        ;INDICATE THAT WE ARE IN A TEST.
 7898        000002                                 TNUM == TNUM + 1          ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 7899 031450 012737 000002 002224                   MOV     #TNUM,TSTNUM      ;SET UP THE TEST NUMBER.        (62)
 7900 031456 012737 000001 005316                   MOV     #1,ERRTYP         ;SET ERROR TYPE IN ERROR TABLE.
 7901 031464 012737 014071 005320                   MOV     #6201.,ERRNBR     ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
 7902 031472 012737 011142 005322                   MOV     #EM6201,ERRMSG    ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
 7903 031500 005037 002502                          CLR     ERSMRF            ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
 7904                                        ;+
 7905                                        ; Reset the DUT to a known state, remove status codes from the fifo.
 7906                                        ;  Clear TX and  RX interrupt enable bits.
 7907                                        ; This subroutine reports error >>>>> 6201 <<<<<.
 7908                                        ;-
 7909 031504 004737 017234                          JSR     PC,CLNRST         ;RESET THE DUT.
 7910 031510 103130                                 BCC     60$               ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
 7911                                        ;+
 7912                                        ; Disable all interrupts.
 7913                                        ; Set up  DMA TX and RX interrupt service routines.
 7914                                        ;-
 7915 031512 106427 000240                          MTPS    #PRIO5            ;DISABLE DEVICE INTERRUPTS.
 7916 031516                                        SETVEC  TXVECA,#TXDMA,#PRIO5   ;SELECT DMA TX INT SERVICE RTN.
      031516 012746 000240                                                                   MOV     #PRIO5,-(SP)
      031522 012746 027574                                                                   MOV     #TXDMA,-(SP)
      031526 013746 002172                                                                   MOV     TXVECA,-(SP)
      031532 012746 000003                                                                   MOV     #3,-(SP)
      031536 104437                                                                          TRAP    C$SVEC
      031540 062706 000010                                                                   ADD     #10,SP
 7917 031544 106427 000000                          MTPS    #PRIO0            ;ALLOW INTERRUPTS.
 7918                                        ;+
 7919                                        ; Clear TX, RX, and DMA_Start error flags.
 7920                                        ;-
 7921 031550 005037 002504                          CLR     TXDONF            ;CLEAR TX DONE FLAGS FOR ALL LINES.
 7922 031554 005037 002506                          CLR     RXDONF            ;CLEAR RX DONE FLAGS FOR ALL LINES.
 7923 031560 005037 002252                          CLR     TXINTF            ;CLEAR TX ERROR FLAGS FOR ALL LINES.
 7924                                        ;+
 7925                                        ; Set up Error table and data pattern table.
```

```
7926                                          ; The numerical value of the character indicates the number of the line
7927                                          ; that transmitted it.
7928                                          ;-
7929 031564  012700  003304                       MOV     #ERCNTB,R0        ;PASS THE ADDRESS OF THE TABLE TO BE CLEARED.
7930 031570  004737  017256                       JSR     PC,CLR16W         ;CLEAR THE RX ERROR COUNTERS TABLE.
7931 031574  005037  003604                       CLR     BUFBAS            ;SET SINGLE CHAR DATA TO BE A NULL.
7932                                          ;+
7933                                          ; Initialise DMA parameters in the control block.
7934                                          ;   Transmission on line group 1 at 8 bits/char,1 stop bits,odd parity.
7935                                          ;   Reception on line group 2 at 5 bits/char,1 stop,odd parity.
7936                                          ;-
7937 031600  012700  156470                       MOV     #156470,R0        ;PASS LPR PARAMETER FOR 8 BITS/CHAR.
7938 031604  012701  156440                       MOV     #156440,R1        ;PASS LPR PARAMETER FOR 5 BITS/CHAR.
7939 031610  004737  021014                       JSR     PC,GETTIM         ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
7940 031614  012702  003604                       MOV     #BUFBAS,R2        ;PASS START ADDRESS OF DATA PATTERN.
7941 031620  012703  000001                       MOV     #1,R3             ;PASS LENGTH OF DATA PATTERN.
7942 031624  013704  002230                       MOV     LGRP1M,R4         ;PASS LINE GROUP OF LINES THAT ARE TO TX.
7943 031630  004737  020036                       JSR     PC,FRPSUP         ;SET UP DUT FOR TRANSMISSION AND RECEPTION.
7944
7945                                          ;+
7946                                          ; Purge the FIFO of any un-wanted characters. This routine reports errors
7947                                          ;  with with error numbers from >>>>> 6202 thru 6204 <<<<<.
7948                                          ; Perform transmission and reception at 9600 baud.
7949                                          ;  Report any errors found, ie. Framing error bit clear or Parity error set.
7950                                          ;-
7951 031634  005237  005320                       INC     ERRNBR            ;SET THE ERROR REPORT NUMBER TO 6202.
7952 031640  004737  023234                       JSR     PC,PUFIFR         ;CLEAN OUT THE FIFO.
7953 031644  103052                               BCC     60$               ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
7954 031646  012737  014075  005320              MOV     #6205.,ERRNBR     ;SET THE ERROR NUMBER TO 6205.
7955 031654  004737  025742                       JSR     PC,TXFRPR         ;TX DATA PATTERN ON SELECTED ACTIVE LINES.
7956 031660  012705  100000                       MOV     #100000,R5        ;PASS FRAMING ERROR TEST FLAG.
7957                                          ;+
7958                                          ; This subroutine reports error number >>>>> 6205 <<<<<.
7959                                          ;-
7960 031664  004737  016676                       JSR     PC,CKFRPR         ;READ CHARACTERS, REPORT ANY ERRORS FOUND.
7961                                          ;+
7962                                          ; Reverse transmission/reception roles on all active lines, and repeat test.
7963                                          ;-
7964 031670  005104                               COM     R4                ;REVERSE ROLES FOR TRANSMISSION AND RECEPTION.
7965 031672  004737  020036                       JSR     PC,FRPSUP         ;SET UP DUT FOR TRANSMISSION AND RECEPTION.
7966 031676  005237  005320                       INC     ERRNBR            ;SET ERROR NUMBER TO 6206.
7967                                          ;+
7968                                          ; This routne reports errors with numbers >>>>> 6206 thru 6208 <<<<<.
7969                                          ;-
7970 031702  004737  023234                       JSR     PC,PUFIFR         ;CLEAN OUT THE FIFO.
7971 031706  103031                               BCC     60$               ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
7972 031710  012737  014101  005320              MOV     #6209.,ERRNBR     ;SET ERROR NUMBER TO 6209.
7973 031716  004737  025742                       JSR     PC,TXFRPR         ;TX DATA PATTERN ON SELECTED ACTIVE LINES.
7974 031722  012705  100000                       MOV     #100000,R5        ;PASS FRAMING ERROR TEST FLAG.
7975                                          ;+
7976                                          ; This subroutine reports errors >>>>> 6209 <<<<<.
7977                                          ;-
7978 031726  004737  016676                       JSR     PC,CKFRPR         ;READ CHARACTERS, REPORT ANY ERRORS FOUND.
7979 031732  005237  005320                       INC     ERRNBR            ;SET ERROR NUMBER TO 6210.
7980                                          ;+
7981                                          ;   Disable interrupts.
7982                                          ;   Clear the interrupt vectors.
```

```
 7983                                    ;    Update the active lines bit map to reflect lines removed from testing.
 7984                                    ;     This subroutine reports errors >>>>> 6210 thru 6212 <<<<<.
 7985                                    ;-
 7986 031736  004737  026036                  JSR     PC,TXIEO        ;DISABLE ALL TX INTERRUPTS.
 7987 031742  004737  026450                  JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM TX/RX.
 7988
 7989 031746  106427  000240                  MTPS    #PRIO5          ;DISABLE DEVICE INTERRUPTS.
 7990 031752                                   CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
      031752  013700  002172                                                                      MOV     TXVECA,RO
      031756  104436                                                                              TRAP    C$CVEC
 7991 031760  012737  014105  005320           MOV    #6213.,ERRNBR   ;SET ERROR NUMBER TO 6213.
 7992                               ;+
 7993                               ; This subroutine reports errors >>>>> 6213 <<<<<.
 7994                               ;-
 7995 031766  004737  024220                   JSR    PC,REPSMR       ;REPORT ERROR SUMMARIES IF CALLED FOR.
 7996 031772  106427  000240           60$:    MTPS   #PRIO5          ;DISABLE DEVICE INTERRUPTS.
 7997 031776  005037  002222                   CLR    CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7998
 7999 032002                                   ENDTST
      032002                                                                           L10026:
      032002  104401                                                                   TRAP    C$ETST
```

```
8001                                    .SBTTL  HARDWARE TEST              - PARERR -
8002                                    ;+*****************************************************************************
8003                                    ;*                        - PARITY ERROR GENERATION TEST -
8004                                    ;*
8005                                    ;*      This test is used to verify the parity error detection and report
8006                                    ;*      capabilities of the DUT.
8007                                    ;*      When staggared loopback mode is selected, data is transmitted
8008                                    ;*      on all active lines in line group 1 with odd parity selected,
8009                                    ;*      and received on lines in group 2 with even parity selected.
8010                                    ;*      This will generate a parity error for each character received.
8011                                    ;*      The parity selection is then reversed on the lines in each group
8012                                    ;*      and the test is repeated.
8013                                    ;*      This test will only execute if the staggared loopback mode is selected.
8014                                    ;*      The special staggared loopback BERG connector must be fitted.
8015                                    ;*
8016                                    ;--*****************************************************************************
8017 032004                                     BGNTST
     032004                                                                                 T3::
8018                                    ;+
8019                                    ; Execute this test in  staggared loopback mode only.
8020                                    ;-
8021 032004  123727  002176  000002             CMPB    LOPBCK,#2            ;CHECK MODE SELECTED.
8022 032012  001161                              BNE     60$                 ;EXIT IF STAGGERD LOOPBACK MODE NOT SELECTED.
8023 032014  012737  177777  002222             MOV     #-1,CTRLCF          ;INDICATE THAT WE ARE IN A TEST.
8024         000003                              TNUM == TNUM + 1            ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
8025 032022  012737  000003  002224             MOV     #TNUM,TSTNUM        ;SET UP THE TEST NUMBER.        (63)
8026 032030  012737  000001  005316             MOV     #1,ERRTYP           ;SET ERROR TYPE IN ERROR TABLE.
8027 032036  012737  014235  005320             MOV     #6301.,ERRNBR       ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
8028 032044  012737  011175  005322             MOV     #EM6301,ERRMSG      ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
8029 032052  005037  002502                      CLR     ERSMRF              ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
8030                                    ;+
8031                                    ; Reset the DUT to a known state, remove status codes from the fifo.
8032                                    ;  Clear TX and  RX interrupt enable bits.
8033                                    ; This subroutine reports error >>>>> 6301 <<<<<.
8034                                    ;-
8035 032056  004737  017234                      JSR     PC,CLNRST           ;RESET THE DUT.
8036 032062  103135                              BCC     60$                 ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
8037                                    ;+
8038                                    ; Disable all interrupts.
8039                                    ; Set up  DMA TX and RX interrupt service routines.
8040                                    ;-
8041 032064  106427  000240                      MTPS    #PRIO5              ;DISABLE DEVICE INTERRUPTS.
8042 032070                                       SETVEC  TXVECA,#TXDMA,#PRIO5     ;SELECT DMA TX INT SERVICE RTN.
     032070  012746  000240                                                                 MOV     #PRIO5,-(SP)
     032074  012746  027574                                                                 MOV     #TXDMA,-(SP)
     032100  013746  002172                                                                 MOV     TXVECA,-(SP)
     032104  012746  000003                                                                 MOV     #3,-(SP)
     032110  104437                                                                         TRAP    C$SVEC
     032112  062706  000010                                                                 ADD     #10,SP
8043 032116  106427  000000                      MTPS    #PRIO0              ;ALLOW INTERRUPTS.
8044                                    ;+
8045                                    ; Clear TX/RX  flags.
8046                                    ;-
8047 032122  005037  002504                      CLR     TXDONF              ;CLEAR TX DONE FLAGS FOR ALL LINES.
8048 032126  005037  002506                      CLR     RXDONF              ;CLEAR RX DONE FLAGS FOR ALL LINES.
8049 032132  005037  002252                      CLR     TXINTF              ;CLEAR TX ERROR FLAGS FOR ALL LINES.
8050                                    ;+
```

```
 8051                                    ; Set up Error counter table.
 8052                                    ;-
 8053 032136  012700  003304                     MOV     #ERCNTB,R0      ;PASS THE ADDRESS OF THE TABLE TO BE CLEARED.
 8054 032142  004737  017256                     JSR     PC,CLR16W       ;CLEAR THE RX ERROR COUNTERS TABLE.
 8055
 8056                                    ;+
 8057                                    ; Initialise DMA parameters in the control block.
 8058                                    ;-
 8059 032146  012700  156470                     MOV     #156470,R0      ;PASS LPR PARAMETER WITH ODD PARITY.
 8060 032152  012701  156570                     MOV     #156570,R1      ;PASS LPR PARAMETER WITH EVEN PARITY.
 8061 032156  004737  021014                     JSR     PC,GETTIM       ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
 8062 032162  012702  005156                     MOV     #SDP2B,R2       ;PASS START ADDRESS OF DATA PATTERN.
 8063 032166  012703  000020                     MOV     #16.,R3         ;PASS LENGTH OF DATA PATTERN.
 8064 032172  013704  002230                     MOV     LGRP1M,R4       ;PASS BIT MAP OF LINES TO BE SET WITH ODD PAR.
 8065 032176  004737  020036                     JSR     PC,FRPSUP       ;SET UP DUT FOR TRANSMISSION AND RECEPTION.
 8066
 8067                                    ;+
 8068                                    ; Purge the FIFO of any un-wanted characters.
 8069                                    ; Perform transmission and reception of the 16 byte data pattern at 9600 baud.
 8070                                    ;  Transmission on line in  group 1, 8 bits/char, 1 stop bits, odd parity.
 8071                                    ;  Reception on lines in  group 2 at 8 bits/char, 1 stop, even parity.
 8072                                    ;  Remove characters from the FIFO and look for the parity error bit being set.
 8073                                    ;  Report any errors found, ie. Framming error bit set or Parity error clear.
 8074                                    ;-
 8075                                      ;+
 8076                                      ; This routine reports errors with numbers >>>>> 6302 thru 6304 <<<<<.
 8077                                      ;-
 8078 032202  004737  023234                     JSR     PC,PUFIFR       ;CLEAN OUT THE FIFO.
 8079 032206  103063                             BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
 8080 032210  012737  014241  005320             MOV     #6305.,ERRNBR   ;SET ERROR NUMBER TO 6305
 8081 032216  004737  021204                     JSR     PC,INIDMA       ;TX DATA PATTERN ON ALL ACTIVE LINES.
 8082 032222  005005                             CLR     R5              ;PASS PARITY ERROR TEST FLAG.
 8083                                    ;+
 8084                                    ; This subroutine reports error number  >>>>> 6305 <<<<<.
 8085                                    ;-
 8086 032224  004737  016676                     JSR     PC,CKFRPR       ;READ CHARACTERS, REPORT ANY ERRORS FOUND.
 8087 032230  005237  005320                     INC     ERRNBR          ;SET ERROR NUMBER TO 6306.
 8088                                    ;+
 8089                                    ; This subroutine reports errors with numbers >>>>> 6306 thru 6309 <<<<<
 8090                                    ;-
 8091 032234  004737  026450                     JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM TX/RX.
 8092 032240  012737  014246  005320             MOV     #6310.,ERRNBR   ;SET ERROR NUMBER TO 6310.
 8093 032246  005037  002504                     CLR     TXDONF          ;CLEAR TX DONE FLAGS FOR ALL LINES.
 8094 032252  005037  002506                     CLR     RXDONF          ;CLEAR RX DONE FLAGS FOR ALL LINES.
 8095 032256  005037  002252                     CLR     TXINTF          ;CLEAR TX DMA HANDOVER ERROR FLAGS.
 8096                                    ;+
 8097                                    ; Reverse transmission/reception roles on all active lines, and repeat test.
 8098                                    ;-
 8099 032262  005104                             COM     R4              ;REVERSE ROLES FOR TRANSMISSION AND RECEPTION.
 8100 032264  004737  020036                     JSR     PC,FRPSUP       ;SET UP DUT FOR TRANSMISSION AND RECEPTION.
 8101                                    ;+
 8102                                    ; This routine reports errors with numbers >>>>> 6310 thru 6311 <<<<<.
 8103                                    ;-
 8104 032270  004737  023234                     JSR     PC,PUFIFR       ;CLEAN OUT THE FIFO.
 8105 032274  103030                             BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
 8106 032276  012737  014250  005320             MOV     #6312.,ERRNBR   ;SET ERROR NUMBER TO 6312.
 8107 032304  004737  021204                     JSR     PC,INIDMA       ;TX DATA PATTERN ON SELECTED ACTIVE LINES.
```

```
8108                                        ;+
8109                                        ; This subroutine reports errors with numbers >>>>> 6312 thru 6316 <<<<<.
8110                                        ;-
8111 032310  004737  016676                      JSR      PC,CKFRPR          ;READ CHARACTERS, REPORT ANY ERRORS FOUND.
8112 032314  012737  014255  005320               MOV      #6317.,ERRNBR      ;SET ERROR NUMBER TO 6317.
8113                                        ;+
8114                                        ;     Disable interrupts.
8115                                        ;     Clear the interrupt vectors.
8116                                        ;     Update the active lines bit map to reflect lines removed from testing.
8117                                        ;-
8118 032322  004737  026036                      JSR      PC,TXIE0           ;DISABLE ALL TX INTERRUPTS.
8119                                        ;+
8120                                        ;   This subroutine reports errors >>>>> 6317 thru 6320 <<<<<.
8121                                        ;-
8122 032326  004737  026450                      JSR      PC,TXRREP          ;REPORT FINAL ERRORS FROM TX/RX.
8123
8124 032332  106427  000240                      MTPS     @PRIO5             ;DISABLE DEVICE INTERRUPTS.
8125 032336                                       CLRVEC   TXVECA             ;RETURN TX INT VECTOR TO UNUSED POOL.
     032336  013700  002172                                                                              MOV    TXVECA,R0
     032342  104436                                                                                      TRAP   C$CVEC
8126
8127                                        ;+
8128                                        ; This subroutine reports errors with numbers >>>>> 6321 <<<<<.
8129                                        ;-
8130 032344  012737  014261  005320               MOV      #6321.,ERRNBR      ;SET ERROR NUMBER TO 6321.
8131 032352  004737  024220                      JSR      PC,REPSMR          ;REPORT ERROR SUMMARIES IF CALLED FOR.
8132
8133 032356  106427  000240          60$:       MTPS     @PRIO5             ;DISABLE DEVICE INTERRUPTS.
8134 032362  005037  002222                      CLR      CTRLCF             ;INDICATE THAT WE ARE NOT WITHIN A TEST.
8135 032366                                       ENDTST
     032366                                                                                 L10027:
     032366  104401                                                                                      TRAP   C$ETST
```

```
8137                                      .SBTTL  HARDWARE TEST                        - DMAADR -
8138                                      ;++ ******************************************************************************
8139                                      ;*                               - DMA ADDRESSING TEST -
8140                                      ;*      THIS TEST VERIFIES , AS FAR AS POSSIBLE , THAT THE DUT CAN PERFORM A
8141                                      ;*      DMA FROM A FULL 22 BIT OR 18 BIT ADDRESS. THE TEST RELIES ON FINDING A
8142                                      ;*      COMPLEMENTARY PAIR OF ADDRESSES BETWEEN THE TOP OF PHYSICAL MEMORY AND
8143                                      ;*      THE START OF THE TOP OF THE DIAGNOSTIC PROGRAM .
8144                                      ;*      THIS MAY INVOLVE REMOVING PART OF THE DIAGNOSTIC RUNTIME SERVICES AND
8145                                      ;*      THEN RESTORING. THE NUMBER OF BITS THAT HAVE BEEN SUCCESSFULLY TESTED
8146                                      ;*      WILL BE PRINTED AT THE CONSOLE AT THE END OF THE TEST.
8147                                      ;*
8148                                      ;*
8149                                      ;-- ******************************************************************************
8150 032370                                   BGNTST
     032370                                                                          T4::
8151
8152
8153          000004                          TNUM ==         TNUM + 1            ;INCREMENT THE ASSEMBLY TIME TEST COUNTER
8154 032370   012737  000004  002224          MOV     @TNUM,TSTNUM         ;SET UP THE TEST NUMBER
8155 032376   012737  177777  002222          MOV     @-1,CTRLCF           ;INDICATE THAT WE ARE IN A TEST
8156 032404   012737  000001  005316          MOV     #1,ERRTYP            ;SET ERROR TYPE AS FATAL IN ERROR TABLE
8157 032412   012737  010461  005320          MOV     #4401.,ERRNBR        ;SET ERROR NUMBER TO 4401
8158 032420   012737  010217  005322          MOV     #EM4401,ERRMSG       ;SET ERROR MESSAGE ADDRESS IN TABLE
8159 032426   012737  014100  005324          MOV     #ER0503,ERRBLK       ;SELECT THE CORRECT ERROR REPORTING ROUTINE
8160
8161                                      ;+
8162                                      ; CLEAR THE SUCCESS FLAG TO INDICATE TEST FAILURE IN CASE IT DOES
8163                                      ;-
8164
8165 032434   005037  034162               CLR     SUCSS                ;INDICATE FAILURE , IN CASE THE DUT FAILS
8166
8167                                      ;+
8168                                      ; SET UP THE 004 TRAP VECTOR TO CATCH ANY EXPECTED TRAPS THAT OCCUR
8169                                      ;-
8170
8171 032440   013737  000004  002254          MOV     @#4,TP4VEC           ;SAVE EXISTING 004 TRAP VECTOR
8172 032446   012737  027552  000004          MOV     #TP4RTN,@#4          ;SET 004 TRAP VECTOR TO OUR SERVICE ROUTINE
8173
8174                                      ;+
8175                                      ; RESET THE DUT TO A KNOWN STATE.REMOVE THE STATUS CODES FROM THE FIFO.
8176                                      ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR
8177                                      ;-
8178
8179 032454   004737  017234               JSR     PC,CLNRST            ;RESET THE DHV , REPORT ANY ERRORS
8180 032460   103402                        BCS     .+6                  ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
8181 032462   000137  034124               JMP     60$                  ;EXIT THE TEST, FATAL ERROR WAS FOUND.
8182
8183                                      ;+
8184                                      ; DETERMINE WHETHER MEMORY MANAGEMENT IS PRESENT
8185                                      ;-
8186
8187 032466   005737  002322               TST     MMPRES               ;IF MEM MGT IS PRESENT THEN
8188 032472   001007                        BNE     1$                   ;AVOID SETTING THE DMA TEST ADDR FOR
8189                                                                     ;A 16 BIT MACHINE.
8190 032474   012737  001252  002264          MOV     #1252,DMTSTA         ;SET UP THE FIRST DMA TEST ADDR FOR
8191                                                                     ;A 16 BIT MACHINE
8192 032502   012737  000021  034152          MOV     #17.,BITSTD          ;SET THE BITS TESTED TO 16 + 1
```

```
8193 032510  000513                          BR      10$           ;SINCE MEM MGT ISN'T PRESENT
8194                                                                ;THERE'S NO NEED TO DETERMINE WHETHER ITS
8195                                                                ;A 22 OR AN 18 BIT MACHINE, SO BRANCH.
8196
8197                              ;+
8198                              ; DETERMINE WHETHER THE HOST IS AN 18 OR A 22 BIT MACHINE. THIS IS ACCOMPLISHED
8199                              ; BY TRYING TO READ A 22-BIT ADDRESS AND COMPARING THE READ DATA WITH THAT
8200                              ; FROM THE EQUIVALENT 18 BIT ADDRESS.
8201                              ;-
8202
8203                              ;+
8204                              ; SET UP THE PARS 1 THROUGH 5 TO RELOCATE TO THE SAME ADDRESS
8205                              ;-
8206
8207 032512  012700  000200      1$:     MOV     #200,R0        ;SET THE PAGE BASE ADDRESS TO 200 $$$
8208 032516  012701  002330              MOV     #PARATB+2,R1   ;POINT AT THE START OF THE PAR ADDRESS TABLE $$$
8209
8210 032522  010031              2$:     MOV     R0,@(R1)+      ;LOAD THE PAR
8211 032524  062700  000200              ADD     #200,R0        ;CALCULATE THE NEXT PAGE ADDRESS
8212 032530  022701  002342              CMP     #PAR6A,R1      ;LOOP UNTIL PARS 0 THROUGH 5 $$$
8213 032534  001372                      BNE     2$             ;ARE LOADED.
8214
8215                              ;+
8216                              ; SET UP THE PDRS FOR , NO ABORT/TRAP,UPWARD EXPANSION,128 BLOCKS PER PAGE
8217                              ;-
8218
8219 032536  012700  077406              MOV     #77406,R0      ;BIT PATTERN FOR THE PDRS
8220 032542  012701  002346              MOV     #PDRATB,R1     ;POINT AT START OF PDR ADDR TABLE
8221 032546  010031              4$:     MOV     R0,@(R1)+      ;
8222 032550  022701  002366              CMP     #PDRATE,R1     ;LOOP UNTIL ALL PDRS HAVE
8223 032554  001374                      BNE     4$             ;BEEN SET UP.
8224                              ;
8225                              ;+
8226                              ; SET THE MEM MGT STATUS REG #3 FOR, 22 BIT ADDRESSSING,
8227                              ;  NO UNIBUS MAPPING, NO D SPACE
8228                              ;-
8229                              ;
8230 032556  012777  000020  147534      MOV     #20,@MMSR3     ;SET UP STATUS REG #3
8231                              ;
8232                              ;+
8233                              ; USE PAR #5 TO DETERMINE WHETHER THIS IS AN 18 OR A 22 BIT MACHINE BY SETTING
8234                              ; IT TO 100000. THIS WILL SELECT A 22 BIT ADDRESS ON A 22 BIT MACHINE.
8235                              ; ON AN 18 BIT MACHINE HOWEVER, THE MSBS WILL BE LOST AND THE MEMORY LOCATION
8236                              ; READ WILL BE THE EQUIVALENT 18 BIT ADDRESS.
8237                              ;   PAR #5 IS USED BECAUSE WE CAN BE SURE THAT THE DIAGNOSTIC WILL NOT COME
8238                              ; UNDER ITS INFLUENCE.
8239                              ;-
8240
8241 032564  012777  100000  147546      MOV     #100000,@PAR5A ;LOAD THE PAGE ADDR INTO THE PAGE ADDR REGISTER
8242
8243                              ;+
8244                              ; SET UP THE LOOP TO ATTEMPT TO READ A 22 BIT ADDR
8245                              ;-
8246
8247 032572  012703  000005              MOV     #5,R3          ;INITIALISE LOOP COUNT
8248 032576  012702  005132              MOV     #SDPBAS,R2     ;SELECT THE VIRTUAL ADDRESS @SDPBAS, THIS WILL BE
8249                                                             ;BITS 0 TO 12 OF THE PHYSICAL ADDRESS
```

```
8250 032602  010204                          MOV    R2,R4                ;SAVE THE DATA ADDRESS
8251 032604  062704  120000                  ADD    #120000,R4           ;CONVERT THE VIRTUAL ADDR INTO AN ADDR WITHIN
8252                                                                      ;THE INFLUENCE OF PAR #5.
8253 032610  012737  000001  002260          MOV    #BITO,BITLNG         ;INDICATE A 22 BIT MACHINE IN CASE IT IS
8254
8255 032616  106427  000340          6$:     MTPS   #PRIO7               ;DISABLE CLOCK INTERRUPTS
8256
8257 032622  010400                          MOV    R4,RO                ;SET UP THE MOVE SOURCE
8258 032624  012701  034156                  MOV    #DEST,R1             ;SET UP THE DESTINATION
8259
8260
8261                                  ;+
8262                                  ; ENABLE THE MEM MGT AND ATTEMPT TO READ THE 22 BIT ADDRESS. IF A TRAP OCCURS
8263                                  ;  THEN THE HOST MUST BE A 22 BIT MACHINE SINCE WE HAVE ENSURED THAT
8264                                  ;  THE 18 BIT ADDRESS EXISTS.
8265                                  ;-
8266 032630  012777  000001  147460          MOV    #BITO,@MMSRO         ;ENABLE MEM MGT
8267 032636  004737  017154                  JSR    PC,CKTRAP            ;PERFORM THE MOVE AND CHECK FOR A TRAP
8268 032642  005077  147450                  CLR    @MMSRO               ;DISABLE MEM MGT
8269
8270 032646  106427  000240                  MTPS   #PRIO5               ;ENABLE CLOCK INTERRUPTS
8271
8272 032652  005737  002256                  TST    TP4FLG               ;DID A TRAP OCCUR ?
8273 032656  001022                           BNE    8$                   ;YES , THEN JUMP AND INDICATE A 22 BIT MACHINE
8274
8275
8276                                  ;+
8277                                  ; SINCE A TRAP HASN'T OCCURED THEN EITHER THE MACHINE IS A 22 BIT MACHINE WITH
8278                                  ;  MEMORY AT THE ADDRESS JUST READ , OR , ITS AN 18 BIT MACHINE IN WHICH CASE
8279                                  ;  THE ADDRESS JUST READ WOULD BE ONE OF THE DATA WORDS AT ADDRESS #SDPBAS.
8280                                  ;-
8281 032660  023712  034156                  CMP    DEST,(R2)            ;COMPARE READ WORD WITH DATA WORD
8282 032664  001017                           BNE    8$                   ;IF NOT THE SAME THEN JUMP AND INDICATE A
8283                                                                      ; 22 BIT MACHINE.
8284
8285                                  ;+
8286                                  ; IN ORDER TO MAKE SURE THAT THIS REALLY ISN'T A 22 BIT MACHINE  I.E. THAT
8287                                  ;  THE MATCH OF THE READ WORD AND THE DATA WORD WAS NOT MERELY COINCIDENCE,
8288                                  ;  ANOTHER FOUR ADDRESSES ARE READ AND COMPARED WITH THE DATA.
8289                                  ;-
8290
8291 032666  062702  000002                  ADD    #2,R2                ;SELECT NEXT TEST DATA ADDRESS
8292 032672  062704  000002                  ADD    #2,R4                ;SELECT NEXT READ ADDRESS
8293 032676  005303                          DEC    R3                   ;DECREMENT THE DATA COUNT
8294 032700  001346                          BNE    6$                   ;REPEAT THE READ ATTEMPTS UNTILL ALL FIVE DATA
8295                                                                      ;WORDS HAVE BEEN SUCCESSFULLY MATCHED, AND THEN
8296 032702  005037  002260                  CLR    BITLNG               ;INDICATE AN 18 BIT MACHINE.
8297
8298                                  ;+
8299                                  ; SET UP THE HIGHEST POSSIBLE TEST ADDRESS IN DMTSTA
8300                                  ;-
8301
8302 032706  012737  005252  002264          MOV    #5252,DMTSTA         ;SET UP THE FIRST DMA TEST ADDRESS FOR THE
8303                                                                      ; 18 BIT MACHINE.
8304 032714  012737  000023  034152          MOV    #19.,BITSTD          ;SET THE BITS TESTED TO 18 BITS + 1.
8305 032722  000406                          BR     10$                  ;AVOID SETTING DMTSTA FOR THE 22 BIT MACHINE
8306 032724  012737  125252  002264  8$:     MOV    #125252,DMTSTA       ;SET UP THE FIRST PAGE ADDR FOR
```

```
 8307
 8308 032732  012737  000027  034152          MOV     #23.,BITSTD      ;A 22 BIT MACHINE.
 8309                                                                   ;SET THE BITS TESTED TO 22 BITS +1.
 8310                                  ;+
 8311                                  ; TRY AND FIND A COMPLEMENTARY PAIR OF ADDRESSES WITHIN THE MEMORY AND SAVE
 8312                                  ;  THE CONTENTS OF THE TWO AREAS. THE TEST IS ABANDONED IF A COMPLEMENTARY
 8313                                  ;  PAIR HAS NOT BEEN FOUND BEFORE THE AREA OF MEMORY CONTAINING THE
 8314                                  ;  DIAGNOSTIC IS ENCOUNTERED.
 8315                                  ;-
 8316
 8317 032740  012737  027530  000004  10$:    MOV     #TP4BRT,@#4      ;CHANGE THE 004 TRAP VECTOR TO POINT TO
 8318                                                                   ;TP4BRT SINCE THIS IS THE ROUTINE ASSOCIATED
 8319                                                                   ;WITH THE BYTE SUBROUTINE CKTRPB.
 8320
 8321 032746                                  MEMORY  FFREM           ;GET THE ADDRESS OF THE FIRST FREE WORD
      032746  104431                                                             TRAP    C$MEM
      032750  010037  002262                                                     MOV     R0,FFREM
 8322                                                                   ;OF MEMORY ABOVE THE DIAGNOSTIC.
 8323
 8324 032754  012701  003604                  MOV     #BUFBAS,R1      ;POINT AT THE BUFFER WHERE THE CONTENTS OF
 8325                                                                   ;THE MEMORY BEING READ ARE TO BE SAVED.
 8326 032760  005004                          CLR     R4              ;CLEAR THE COMPLEMENTARY PAIR INDICATOR (CPI)
 8327 032762  106427  000340                  MTPS    #PRIO7          ;DISABLE LTC INTERRUPTS
 8328
 8329 032766  005204                  12$:    INC     R4              ;INCREMENT THE CPI
 8330 032770  005005                          CLR     R5              ;INDICATE THAT A SAVE OF THE DATA AT
 8331                                                                   ;(DMTSTA) IS REQUIRED.
 8332 032772  012703  000020                  MOV     #16.,R3         ;SET THE NUMBER OF BYTES TO BE READ
 8333 032776  004737  017466                  JSR     PC,DMRW         ;SAVE THE DATA CONTAINED AT ADDRESS DMTSTA.
 8334 033002  012701  004204                  MOV     #BUFMID,R1      ;POINT AT SECOND STORAGE AREA
 8335 033006  005737  002256                  TST     TP4FLG          ;IF WE HAVE VALID MEMORY THEN AVOID CLEARING
 8336 033012  001403                          BEQ     14$             ;THE CPI AND RESETING THE SAVE AREA ADDR
 8337
 8338 033014  005004                          CLR     R4              ;CLEAR THE CPI.
 8339 033016  012701  003604                  MOV     #BUFBAS,R1      ;RESET THE ADDR FOR THE SAVED DATA STORE
 8340 033022  022704  000002          14$:    CMP     #2,R4           ;IF A PAIR OF COMPLEMENTARY ADDRESSSES HAVE
 8341                                                                   ;BEEN FOUND THEN
 8342 033026  001447                          BEQ     17$             ;GO AND WRITE THE TEST DATA TO THESE ADDRS.
 8343 033030  013737  002264  034160          MOV     DMTSTA,ODTSTA   ;SAVE THE OLD DMTSTA
 8344 033036  000241                          CLC                     ;CLEAR CARRY READY FOR THE ROTATION
 8345 033040  006037  002264                  ROR     DMTSTA          ;COMPLEMENT THE DMTSTA TO PRODUCE THE NEXT
 8346                                                                   ; DMA TEST ADDR.
 8347 033044  005337  034152                  DEC     BITSTD          ;DECREMENT THE NUMBER OF BITS TESTED COUNT
 8348
 8349                                  ;+
 8350                                  ; CHECK THAT THE NEW DMTSTA IS NOT INSIDE THE DIAGNOSTIC PROGRAM
 8351                                  ;-
 8352
 8353 033050  032737  176000  002264          BIT     #176000,DMTSTA  ;IS THE DMTSTA > 1252 . IF IT IS THEN WE'RE
 8354                                                                   ; SAFE SO.
 8355 033056  001343                          BNE     12$             ;BRANCH AND CONTINUE WITH THE SEARCH
 8356 033060  004737  017414                  JSR     PC,DM16B        ;CONVERT THE DMTSTA TO A PHYSICAL ADDR.
 8357 033064  020037  002262                  CMP     R0,FFREM        ;ARE WE INSIDE THE DIAGNOSTIC REGION ?
 8358 033070  103336                          BHIS    12$             ;NO . THEN BRANCH AND CONTINUE WITH THE SEARCH
 8359
 8360                                  ;+
 8361                                  ;SINCE WE ARE NOW INSIDE THE DIAGNOSTIC, WE INCREMENT BIT #14 OF THE DMTSTA
```

```
8362                                                   ;PHYSICAL ADDRESSS AND IF WE'RE STILL INSIDE THE DIAGNOSTIC WE ABANDON THE
8363                                                   ;TEST. ONCE WE ARE IN THIS REGION WE ARE ONLY ABLE TO TEST THE LOWEST 14 BITS.
8364                                                   ;-
8365
8366
8367 033072  022737  000252  002264        CMP    #252,DMTSTA     ;IF THE BIT HAS ALREADY BEEN SET THEN
8368 033100  001014                         BNE    15$             ;ABANDON THE TEST,AFTER REPORTING THE ERROR .
8369                                                                ; BECAUSE NO SUITABLE MEMORY HAS BEEN FOUND.
8370 033102  012737  000652  002264         MOV    #652,DMTSTA     ;SET THE BIT
8371 033110  062700  040000                 ADD    #40000,R0       ;ADD THE BIT INTO THE PHYSICAL ADDR
8372 033114  020037  002262                 CMP    R0,FFREM        ;IF WE'RE NOW STILL INSIDE THE DIAGNOSTIC THEN
8373 033120  103404                          BLO    15$             ;REPORT ERROR AND ABANDON THE TEST.
8374 033122  012737  000016  034152         MOV    #14.,BITSTD     ;OTHERWISE SET THE BITS TESTED TO 14 BITS.
8375 033130  000716                          BR     12$             ;CONTINUE WITH THE SEARCH.
8376
8377
8378 033132  005237  005320        15$:      INC    ERRNBR          ;SET THE ERROR NUMBER TO 4402
8379 033136  012701  010241                  MOV    #EM4402,R1      ;SELECT MESSAGE TO BE REPORTED.
8380                                                                ; " NO SUITABLE ADDR FOUND. DMA TEST ABORTED "
8381
8382
8383 033142  000137  034122        16$:      JMP    34$             ;JUMP TO THE ERROR.
8384
8385
8386                                          ;+
8387                                          ; WRITE THE TEST DATA INTO THE TWO AREAS JUST FOUND. IF A TRAP OCCURS WHILE
8388                                          ; WE ARE WRITING DATA INTO THESE AREAS THEN THE HOST MACHINE IS AT FAULT.
8389                                          ;-
8390
8391 033146  012700  005132        17$:      MOV    #SDPBAS,R0      ;SET UP THE SOURCE ADDR FOR THE MOVE AS OUR
8392                                                                ;TEST DATA PATTERN.
8393 033152  013737  002264  034154          MOV    DMTSTA,DUMY     ;SAVE THE LOWER DMTSTA
8394 033160  013737  034160  002264          MOV    ODTSTA,DMTSTA   ;START WITH THE HIGHER OF THE TWO
8395                                                                ; COMPLEMENTARY ADDRESSES.
8396 033166  012703  000020                  MOV    #16.,R3         ;SET THE NUMBER OF DATA BYTES TO BE WRITTEN
8397 033172  012705  000001                  MOV    #1,R5           ;INDICATE TO WRITE TO DMTSTA
8398
8399
8400 033176  012701  000340                  MOV    #340,R1         ;SET PRIORITY 7 TO DISABLE THE CLOCK
8401 033202  004737  025352                  JSR    PC,STPSW        ;
8402
8403 033206  005237  005320                  INC    ERRNBR          ;SET THE ERROR NUMBER TO 4403
8404 033212  012701  010307                  MOV    #EM4403,R1      ;SELECT THE MESSAGE,
8405                                                                ; "HOST FAILURE. WRITE FAILED TO AN ADDR WHICH
8406                                                                ;HAD BEEN SUCCESSFULLY READ, TEST ABANDONED "
8407
8408 033216  004737  017466                  JSR    PC,DMRW         ;PERFORM THE TRANSFER
8409 033222  005737  002256                  TST    TP4FLG          ;EXIT IF HOST FAILURE
8410 033226  001345                          BNE    16$             ;AND REPORT ERROR.
8411 033230  013737  034154  002264          MOV    DUMY,DMTSTA     ;SELECT THE LOWER DMA TEST ADDR.
8412 033236  012700  005156                  MOV    #SDP2B,R0       ;SELECT THE NEXT DATA PATTERN
8413 033242  004737  017466                  JSR    PC,DMRW         ;PERFORM THE TRANSFER
8414 033246  005737  002256                  TST    TP4FLG          ;EXIT IF HOST FAILURE
8415 033252  001333                          BNE    16$             ;
8416
8417                                          ;+
8418                                          ; SET UP THE DHV11-M TO PERFORM THE DMA.
```

```
 8419                                        ;-
 8420
 8421                                        ;+
 8422                                        ; SET INTERNAL LOOPBACK, ENABLE THE RECIEVER FUNCTION ON THE LINE.
 8423                                        ; SET THE LPR ON THE LINE TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
 8424                                        ; 2 STOP BITS. ENABLE THE TRANSMITTER ON THE LINE.
 8425                                        ;-
 8426
 8427 033254  005237  005320                       INC     ERRNBR          ;SET THE ERRNBR TO 4404
 8428
 8429
 8430 033260  004737  017756                       JSR     PC,FINACT       ;FIRST FIND AN ACTIVE LINE ON WHICH TO PERFORM
 8431                                                                       ; THE DMA.
 8432 033264  010102                               MOV     R1,R2           ;SAVE THE LINE NUMBER ON WHICH THE DMA WILL OCCUR
 8433 033266  012701  010404                       MOV     #EM4404,R1      ;SELECT THE MESSAGE,
 8434                                                                       ; "NO ACTIVE LINES , TEST ABANDONED"
 8435 033272  103402                               BCS     .+6             ;EXIT IF A LINE COULD NOT BE FOUND ,AFTER FIRST
 8436 033274  000137  033724                       JMP     30$             ;RESTORING THE CONTENTS OF MEMORY.
 8437 033300  010201                               MOV     R2,R1           ;RESTORE THE ACTIVE LINE NUMBER.
 8438                                        ;+
 8439                                        ; AN ACTIVE LINE HAS BEEN FOUND
 8440                                        ;-
 8441
 8442 033302  012700  000204                       MOV     #204,R0         ;PASS THE LNCTRL CONTENTS
 8443 033306  004737  027234                       JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTER
 8444 033312  012700  177670                       MOV     #177670,R0      ;PASS THE LPR CONTENTS
 8445 033316  004737  027264                       JSR     PC,WTWLPR       ;INITIALISE THE LPR REGISTER
 8446 033322  004737  025646                       JSR     PC,TXENBL       ;ENABLE TRANSMITTER ON THE LINE
 8447
 8448                                        ;+
 8449                                        ; INITIATE THE DMA
 8450                                        ;-
 8451
 8452 033326  013705  034160                       MOV     ODTSTA,R5       ;START FROM THE HIGHER OF THE PAIR OF ADDR.
 8453 033332  012704  005132                       MOV     #SDPBAS,R4      ;SET UP THE ADDR OF THE DATA PATTERN
 8454 033336  010137  034150                       MOV     R1,80$          ;SAVE THE LINE NUMBER FOR THE DMA
 8455 033342  012737  000002  034146               MOV     #2,70$          ;INITIALISE THE LOOP COUNT
 8456
 8457 033350  012700  000052        18$:           MOV     #52,R0          ;SET UP THE LSB'S
 8458 033354  005003                               CLR     R3              ;CLEAR THE REG THAT WILL HOLD THE 6 MSB'S
 8459 033356  012702  000006                       MOV     #6,R2           ;CONVERT THE DMTSTA INTO
 8460 033362  006305                20$:           ASL     R5              ;A PHYSICAL ADDRESS WITH
 8461 033364  006103                               ROL     R3              ;THE MSB'S IN REG #3.
 8462 033366  005302                               DEC     R2              ;
 8463 033370  001374                               BNE     20$             ;
 8464 033372  032705  000100                       BIT     #100,R5         ;TEST BIT #6 OF THE DMTSTA
 8465 033376  001402                               BEQ     22$             ;
 8466 033400  012700  000025                       MOV     #25,R0          ;ALTER THE LSB'S IF BIT #6 WAS SET.
 8467 033404  060005                22$:           ADD     R0,R5           ;ADD IN THE LSB'S
 8468 033406  052703  000200                       BIS     #200,R3         ;SET BIT #7.
 8469
 8470 033412  013777  034150  146562               MOV     80$,@CSRA       ;SELECT THE LINE ON WHICH TO PERFORM THE DMA.
 8471
 8472 033420  012737  010465  005320               MOV     #4405.,ERRNBR   ;SET ERROR NUMBER 4405
 8473 033426  012701  010443                       MOV     #EM4405,R1      ;SELECT THE MESSAGE,
 8474                                                                       ; " DMA_START BIT FOUND SET BEFORE DMA INIT.
 8475                                                                       ;TEST ABANDONED"
```

```
8476 033432  105777  146560              TSTB   @TXAD2A        ;TEST THE DUT DMA-START BIT
8477 033436  100532                      BMI    30$            ;EXIT WITH ERROR IF SET ,AFTER FIRST RESTORING
8478                                                           ;THE CONTENTS OF MEMORY.
8479 033440  012777  000020  146552      MOV    #16.,@TXBFCA   ;SET UP CHARACTER COUNT
8480 033446  010577  146542              MOV    R5,@TXAD1A     ;SET UP BITS 0 TO 15 OF THE PHYISCAL ADDR.
8481 033452  110377  146540              MOVB   R3,@TXAD2A     ;SET UP BITS 16 TP 21 , AND INITIATE THE DMA.
8482
8483                              ;+
8484                              ; WAIT FOR THE DMA TO COMPLETE AND THE LAST CHARACTER TO BE RECIEVED
8485                              ;-
8486
8487 033456  012701  170144              MOV    #170144,R1     ;TEST BIT 15, TIME-OUT OF 100 MS.
8488 033462  013702  002202              MOV    CSRA,R2        ;PASS THE ADDR OF THE REG TO TEST.
8489
8490 033466  005237  005320              INC    ERRNBR         ;SET ERROR NUMBER TO 4406'
8491
8492 033472  004737  027160              JSR    PC,WAIBIS      ;WAIT FOR BIT TO SET
8493 033476  012701  010537              MOV    #EM4406,R1     ;SELECT THE MESSAGE ,
8494                                                           ; " TIME-OUT OCCURED WAITING FOR DMA TO
8495                                                           ;COMPLETE. TEST ABANDONED"
8496 033502  103110                      BCC    30$            ;EXIT IF TIME-OUT OCCCURED, AFTER FIRST .
8497                                                           ;RESTORING THE CONTENTS OF MEMORY.
8498 033504  010402                      MOV    R4,R2          ;SAVE R4
8499 033506  012704  000005              MOV    #5,R4          ;SET 5 MS DELAY
8500 033512  004737  017354              JSR    PC,DELAY       ;DELAY TO ALLOW LAST CHARACTER TO BE RECIEVED
8501 033516  010204                      MOV    R2,R4          ;RESTORE R4
8502
8503                              ;+
8504                              ; READ THE CONTENTS OF THE RXFIFO AND COMPARE THEM WITH THE CORRECT DATA
8505                              ;-
8506
8507 033520  005003                      CLR    R3             ;CLEAR THE READ DATA COUNTER
8508 033522  012705  000200              MOV    #128.,R5       ;SET THE MAX BMP CODE READ COUNT
8509
8510 033526  012737  010467  005320 24$: MOV    #4407.,ERRNBR  ;SET THE ERRNBR TO 4407
8511 033534  012701  010613              MOV    #EM4407,R1     ;SELECT THE MESSAGE ,
8512                                                           ; " RXFIFO EMPTY TOO SOON, DMA FAILED
8513                                                           ;TEST ABANDONED"
8514
8515 033540  017702  146440              MOV    @RBUFA,R2      ;READ THE CHARACTER FROM THE FIFO
8516 033544  100067                      BPL    30$            ;BRANCH TO REPORT ERROR IF FIFO EMPTY TOO SOON,
8517                                                           ;AFTER FIRST RESTORING THE CONTENTS OF MEMORY.
8518 033546  012700  170301              MOV    #170301,R0     ;SET UP BIT MASK OF A BMP CODE
8519 033552  040200                      BIC    R2,R0          ;TRY TO CLEAR THE BMP CODE MASK
8520 033554  001011                      BNE    28$            ;BRANCH IF NOT A BMP CODE
8521 033556  004737  024726              JSR    PC,SAVBMP      ;SAVE THE BMP CODE ON THE QUEUE
8522
8523 033562  005237  005320              INC    ERRNBR         ;SET THE ERRNBR TO 4408
8524 033566  012701  010675              MOV    #EM4408,R1     ;SELECT THE MESSAGE,
8525                                                           ; " TOO MANY BMP CODES FOUND IN THE RXFIFO,
8526                                                           ;TEST ABANDONED"
8527
8528 033572  005305                      DEC    R5             ;DEC THE MAX BMP CODE READ COUNT
8529 033574  001453                      BEQ    30$            ;GO REPORT ERROR IF TOO MANY BMP CODES FOUND .
8530                                                           ;AFTER FIRST RESTORING THE CONTENTS OF MEMORY.
8531 033576  000753                      BR     24$            ;DON'T COUNT THE BMP CODE AS A VALID CHARACTER
8532
```

```
 8533
 8534 033600   012737   010471   005320  28$:    MOV     #4409.,ERRNBR       ;SET THE ERRNBR TO 4409
 8535 033606   010201                            MOV     R2,R1               ;SAVE THE CHARACTER FROM THE FIFO
 8536 033610   012702   010740                   MOV     #EM4409,R2          ;SELECT THE MESSAGE ,
 8537                                                                        ; " BAD BIT BETWEEN BITS 0 AND "
 8538 033614   012737   015402   005324          MOV     #ER9101,ERRBLK      ;SELECT THE ERROR ROUTINE.
 8539 033622   012737   177777   034162          MOV     #-1,SUCSS           ;INDICATE 'BAD BITS' FAILURE
 8540
 8541 033630   122401                            CMPB    (R4)+,R1            ;COMPARE CHAR FROM FIFO WITH THE CORRECT DATA.
 8542 033632   001034                            BNE     30$                 ;BRANCH IF INCORRECT AND RESTORE MEM CONT'S.
 8543 033634   005037   034162                   CLR     SUCSS               ;INDICATE NON TEST SPECIFIC FAILURE  E.G. TIME-OUTS
 8544 033640   005203                            INC     R3                  ;COUNT THIS CHARACTER.
 8545 033642   022703   000020                   CMP     #16.,R3             ;HAVE WE RECIEVED ALL THE CHARACTERS ?
 8546 033646   001327                            BNE     24$                 ;LOOP UNTIL ALL CHARACTERS (NON-BMP) ARE READ.
 8547 033650   005337   034146                   DEC     70$                 ;DECREMENT THE LOOP COUNT
 8548 033654   001420                            BEQ     29$                 ;BRANCH IF BOTH DMA'S ARE COMPLETED
 8549 033656   012704   005156                   MOV     #SDP2B,R4           ;SET UP THE SECOND DATA PATTERN
 8550 033662   013705   002264                   MOV     DMTSTA,R5           ;SET UP THE OTHER DMA TEST ADDRESS
 8551
 8552 033666   012737   010472   005320          MOV     #4410.,ERRNBR       ;SET ERRNBR TO 4410
 8553 033674   012701   010775                   MOV     #EM4410,R1          ;SELECT THE MESSAGE ,
 8554                                                                        ; " RXFIFO FAILED TO PURGE, TEST ABANDONED "
 8555 033700   012737   014100   005324          MOV     #ER0503,ERRBLK      ;SELECT THE ERROR ROUTINE
 8556
 8557 033706   004737   023152                   JSR     PC,PUFIFO           ;PURGE THE RXFIFO
 8558 033712   103004                            BCC     30$                 ;EXIT WITH ERROR IF FIFO WOULD NOT PURGE ,
 8559                                                                        ;AFTER FIRST RESTORING THE CONTENTS OF MEMORY.
 8560 033714   000615                            BR      18$                 ;OTHERWISE REPEAT.
 8561
 8562 033716   012737   000001   034162  29$:    MOV     #1,SUCSS            ;INDICATE THAT WE HAVE BEEN ABLE TO TEST,
 8563                                                                        ;SOME OF THE BITS.
 8564
 8565
 8566                                     ;+
 8567                                     ; RESTORE THE ORIGINAL DATA IN THE MEMORY
 8568                                     ;-
 8569
 8570 033724   013737   002264   034154  30$:    MOV     DMTSTA,DUMY         ;START WITH THE HIGHER OF THE PAIR OF DMTSTA
 8571 033732   013737   034160   002264          MOV     ODTSTA,DMTSTA       ;
 8572 033740   012700   003604                   MOV     #BUFBAS,R0          ;POINT AT THE START OF THE SAVED DATA AREA
 8573 033744   012705   000001                   MOV     #1,R5               ;SELECT WRITE TO (DMTSTA)
 8574 033750   012703   000020                   MOV     #16.,R3             ;PASS NUMBER OF BYTES TO BE WRITTEN
 8575 033754   004737   017466                   JSR     PC,DMRW             ;RESTORE THE DATA
 8576 033760   005737   002256                   TST     TP4FLG              ;GO REPORT ERROR IF A TRAP OCCURED
 8577 033764   001012                            BNE     31$                 ;
 8578 033766   013737   034154   002264          MOV     DUMY,DMTSTA         ;NOW RESTORE THE DATA FROM THE LOWER
 8579                                                                        ;OF THE PAIR OF TEST ADDRESSES.
 8580 033774   012700   004204                   MOV     #BUFMID,R0          ;POINT AT THE START OF THE SAVED DATA AREA
 8581 034000   004737   017466                   JSR     PC,DMRW             ;RESTORE THE DATA
 8582
 8583 034004   005737   002256                   TST     TP4FLG              ;GO REPORT ANY ERRORS IF A NO TRAP
 8584 034010   001411                            BEQ     32$                 ; OCCURED DURING THE RESTORE.
 8585
 8586 034012   012737   010473   005320  31$:    MOV     #4411.,ERRNBR       ;SET THE ERROR NUMBER TO 4411
 8587 034020   012701   011024                   MOV     #EM4411,R1          ;SELECT THE MESSAGE ,
 8588                                                                        ; " HOST FAILURE. WRITE FAILURE TO AN ADDR
 8589                                                                        ;WHICH HAD PREVIOUSLY BEEN SUCCESSFULLY
```

```
8590
8591 034024  012737  014100  005324              MOV     #ER0503,ERRBLK   ;WRITTEN TO. "
                                                                           ;SELECT THE ERROR ROUTINE
8592 034032  000433                               BR      34$              ;REPORT THE ERROR
8593
8594                                       ;+
8595                                       ; HAS THE TEST BEEN SUCCESSFUL, PRINT THE BITS TESTED IF IT HAS,
8596                                       ; REPORT THE ERRORS OTHERWISE.
8597                                       ;-
8598
8599
8600 034034  005737  034162       32$:     TST     SUCSS            ;IF THE ERROR IS NON TEST SPECIFIC THEN
8601 034040  001430                         BEQ     34$              ;BRANCH TO REPORT ERRORS
8602 034042  013701  034152                 MOV     BITSTD,R1        ;LOAD THE NUMBER OF BITS TESTED
8603 034046  005301                          DEC     R1               ;DEC TO GIVE THE BIT POSITION OF THE MSB TESTED.
8604 034050  022737  000001  034162          CMP     #1,SUCSS         ;IF THE BITS TESTED ARE BAD THEN
8605 034056  001021                          BNE     34$              ;BRANCH AND REPORT ERRORS.
8606
8607                                       ;+
8608                                       ; OTHERWISE DETERMINE IF PRINTING OF THE SUCCESSFULLY TESTED BITS WAS REQUESTED.
8609                                       ;-
8610
8611 034060  032737  000040  002164          BIT     #BIT05,OPTION    ; PRINT THE BITS TESTED IF THE SOFTWARE
8612 034066  001416                          BEQ     60$              ;OPTION HAS REQUESTED IT
8613 034070  010102                          MOV     R1,R2            ;CALCULATE THE NUMBER OF BITS WHICH HAVE
8614 034072  005202                          INC     R2               ;  BEEN TESTED SUCCESSFULLY.
8615 034074                                  PRINTB  #EF4401,R1,R2    ;PRINT THE NUMBER OF BITS TESTED MESSSAGE.
     034074  010246                                                                       MOV    R2,-(SP)
     034076  010146                                                                       MOV    R1,-(SP)
     034100  012746  005554                                                               MOV    #EF4401,-(SP)
     034104  012746  000003                                                               MOV    #3,-(SP)
     034110  010600                                                                       MOV    SP,R0
     034112  104414                                                                       TRAP   C$PNTB
     034114  062706  000010                                                               ADD    #10,SP
8616 034120  000401                          BR      60$              ;EXIT THE TEST
8617
8618
8619 034122                         34$:     ERROR                    ;  REPORT ERRORS
     034122  104460                                                                       TRAP   C$ERROR
8620
8621
8622 034124  106427  000240       60$:     MTPS    #PRIO5           ;ENABLE THE CLOCK
8623 034130  013737  002254  000004          MOV     TP4VEC,@#4       ;RESTORE THE NORMAL 004 TRAP VECTOR
8624 034136  005037  002222                  CLR     CTRLCF           ;INDICATE THAT WE ARE NOT WITHIN A TEST
8625
8626 034142                                  EXIT    TST
     034142  104432                                                                       TRAP   C$EXIT
     034144  000020                                                                       .WORD  L10030-.
8627
8628
8629                                       ;+
8630                                       ; ********************** LOCAL VARIBLE AREA *******************************
8631                                       ;-
8632
8633 034146  000000              70$:     .WORD   0                ;COUNTER FOR THE NUMBER OF DMA'S COMPLETED
8634 034150  000000              80$:     .WORD   0                ;SAVE AREA FOR THE ACTIVE LINE NUMBER
8635 034152  000000              BITSTD:  .WORD   0                ;NUMBER OF BITS TESTED
8636 034154  000000              DUMY:    .WORD   0                ;DUMMY VARIABLE
```

```
8637 034156  000000           DEST:    .WORD   0        ;SAVE AREA FOR READ WORD , WHEN DETERMINING
8638                                                     ;MACHINE ADDRESS LENGTH.
8639 034160  000000           ODTSTA:  .WORD   0        ;HIGHER OF THE PAIR OF COMPLEMENTARY ADDR.
8640 034162  000000           SUCSS:   .WORD   0        ;SUCCESS INDICATOR. -1 - ERROR DUE TO BAD BITS
8641                                                     ;              1 - SUCCESSFUL TEST
8642                                                     ;              0 - OTHER ERRORS
8643
8644                           ;+
8645                           ;**************************** END *****************************************
8646                           ;-
8647
8648
8649
8650 034164                        ENDTST
     034164                                                            L10030:
     034164  104401                                                         TRAP    C$ETST
8651
```

```
8653                                     .SBTTL  HARDWARE TEST              - MODLPB -
8654                                     ;++  ***********************************************************************
8655                                     ;*                    - Modem Loopback Test -
8656                                     ;*        This test is used to move data through a modem which is connected to
8657                                     ;*        one of the device serial ports.  This test is run only if Modem
8658                                     ;*        Loopback is specified.  This test utilizes the following operator
8659                                     ;*        dialogue:
8660                                     ;*                MODEM BAUDRATE IN BPS:  (D) 1200 ?
8661                                     ;*                TYPE <CR> WHEN MODEM LINK ESTABLISHED:  (L)  Y ?
8662                                     ;*                MODEM STATUS SIGNAL REPORT:
8663                                     ;*                    LINE #n:  DSR=n, RI=n, DCD=n, CTS=n
8664                                     ;*                    ...    repeated for each active line
8665                                     ;*                NUMBER OF 256 BYTE DATA PATTERNS TO SEND ON EACH SELECTED LINE
8666                                     ;*                    (1-255, 0=SEND UNTIL +C): (D)  1 ?
8667                                     ;*                PRINT MODEM STATUS SIGNAL REPORT AFTER EACH PATTERN: (L)  Y ?
8668                                     ;*
8669                                     ;*        At the completion of sending the specified number of data patterns the
8670                                     ;*        test issues the following prompt:
8671                                     ;*                EXIT THE TEST (N = LOOP BACK TO SEND MORE DATA): (L) Y ?
8672                                     ;*
8673                                     ;*        If extended error reporting is allowed, a report is printed at the end
8674                                     ;*        of each data pattern with the following format:
8675                                     ;*          MODEM LOOPBACK TEST STATUS REPORT:   PATTERN #nnn (D) COMPLETED.
8676                                     ;*
8677                                     ;*        This test is performed using 8 bits per character, 1 stop bit, and no
8678                                     ;*        parity.  This test does not support split speed.  All selected lines
8679                                     ;*        are tested at the selected baudrate.  An error summary is reported at
8680                                     ;*        the end of the test if any lines have exceeded the number of individual
8681                                     ;*        data errors to report as selected in the Software P-Table dialogue.
8682                                     ;*
8683                                     ;--  ***********************************************************************
8684 034166                             BGNTST
     034166                                                                            T5::
8685             000005                  TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
8686 034166 012737  000005  002224       MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.          (89)
8687                                     ;+
8688                                     ; Verify that the test should be performed.  Must have the following:
8689                                     ;       Modem loopback selected.
8690                                     ;       Manual intervention allowed.
8691                                     ;-
8692 034174 123727  002176  000004       CMPB    LOPBCK,#4      ;TEST THE LOOPBACK TYPE INDICATOR.
8693 034202 001402                        BEQ     2$             ;MODEM LOOPBACK SELECTED? YES, CONTINUE TEST.
8694 034204                               EXIT    TST            ;NO, ABORT THE TEST.
     034204 104432                                                                     TRAP    C$EXIT
     034206 000650                                                                     .WORD   L10031-.
8695 034210                       2$:     MANUAL                 ;CHECK FOR MANUAL INTERVENTION ALLOWED.
     034210 104450                                                                     TRAP    C$MANI
8696 034212                               BCOMPLETE       4$     ;MANUAL INTERVENTION ALLOWED? YES, DO TEST.
     034212 103402                                                                     BCS     4$
8697 034214                               EXIT    TST            ;NO, ABORT THE TEST.
     034214 104432                                                                     TRAP    C$EXIT
     034216 000640                                                                     .WORD   L10031-.
8698
8699 034220 012737  177777  002222 4$:    MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
8700 034226 012737  000001  005316        MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
8701 034234 012737  021305  005320        MOV     #8901.,ERRNBR  ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
8702 034242 012737  011220  005322        MOV     #EM8901,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
```

```
 8703 034250  005037  002502                    CLR     ERSMRF           ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
 8704                                    ;+
 8705                                    ; Reset the DUT to a known state, remove the status codes from the fifo.
 8706                                    ;  Clear TX and RX interrupt enable bits in the CSR.
 8707                                    ; This subroutine reports error >>>>> 8901 <<<<<.
 8708                                    ;-
 8709 034254  004737  017234                    JSR     PC,CLNRST        ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
 8710 034260  103402                             BCS     .+6              ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
 8711 034262  000137  035046                     JMP     60$              ;RESET FAILURE, ABORT THIS TEST.
 8712
 8713                                    ;+
 8714                                    ; Set up for Transmit and Receive interrupts.
 8715                                    ;-
 8716 034266  106427  000240                     MTPS    #PRIO5           ;DISABLE DEVICE INTERRUPTS.
 8717 034272                                     SETVEC  TXVECA,#TXDMA,#PRIO5    ;SELECT DMA TX INT SERVICE RTN.
      034272  012746  000240                                                                      MOV     #PRIO5,-(SP)
      034276  012746  027574                                                                      MOV     #TXDMA,-(SP)
      034302  013746  002172                                                                      MOV     TXVECA,-(SP)
      034306  012746  000003                                                                      MOV     #3,-(SP)
      034312  104437                                                                              TRAP    C$SVEC
      034314  062706  000010                                                                      ADD     #10,SP
 8718 034320                                     SETVEC  RXVECA,#RXCHRS,#PRIO5   ;SELECT RX INT SERVICE RTN.
      034320  012746  000240                                                                      MOV     #PRIO5,-(SP)
      034324  012746  027364                                                                      MOV     #RXCHRS,-(SP)
      034330  013746  002170                                                                      MOV     RXVECA,-(SP)
      034334  012746  000003                                                                      MOV     #3,-(SP)
      034340  104437                                                                              TRAP    C$SVEC
      034342  062706  000010                                                                      ADD     #10,SP
 8719 034346  106427  000000                     MTPS    #PRIO0           ;ALLOW INTERRUPTS.
 8720                                    ;+
 8721                                    ; Clear the cumulative error counters (one for each line).
 8722                                    ;-
 8723 034352  012700  003304                     MOV     #ERCNTB,R0
 8724 034356  004737  017256                     JSR     PC,CLR16W        ;CLEAR THE RX ERROR COUNTERS TABLE.
 8725                                    ;+
 8726                                    ; Print the the test name.
 8727                                    ;-
 8728 034362                                     PRINTF  #EF8901,#EM8901
      034362  012746  011220                                                                      MOV     #EM8901,-(SP)
      034366  012746  006235                                                                      MOV     #EF8901,-(SP)
      034372  012746  000002                                                                      MOV     #2,-(SP)
      034376  010600                                                                              MOV     SP,R0
      034400  104417                                                                              TRAP    C$PNTF
      034402  062706  000006                                                                      ADD     #6,SP
 8729                                    ;+
 8730                                    ; Prepare to call the set up routine.
 8731                                    ;    Get the desired baudrate from the operator.
 8732                                    ;    Calculate proper DUT LPR contents.
 8733                                    ;    Calculate the proper RX time-out value for this speed.
 8734                                    ;    Set up the bit map of unused TX/RX bits.
 8735                                    ;-
 8736 034406  004737  020302                     JSR     PC,GETBDR
 8737 034412  010100                             MOV     R1,R0
 8738 034414  006301                             ASL     R1
 8739 034416  006301                             ASL     R1
 8740 034420  006301                             ASL     R1               ;GET DUPLICATE COPIES OF BAUDRATE CODE
 8741 034422  006301                             ASL     R1               ;  IN THE UPPER BYTE OF THE NEW
```

```
8742 034424  050001                        BIS      R0,R1              ; LPR CONTENTS.
8743 034426  000301                        SWAB     R1
8744 034430  042701  000377                BIC      #377,R1            ;SET UP 1 STOP BIT, NO PARITY, 8 BITS/CHAR
8745 034434  052701  000030                BIS      #30,R1             ;  IN THE LPR CONTENTS.
8746
8747 034440  004737  021014                JSR      PC,GETTIM          ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
8748 034444  012737  177400  002226         MOV     #177400,IBM        ;FORM BIT MAP OF UNUSED TX/RX BITS.
8749                               ;+
8750                               ;    Set up a 256 byte data pattern.
8751                               ;-
8752 034452  005003                        CLR      R3                 ;PREPARE TO START DATA PATTERN AT 255.
8753 034454  012702  003604                MOV      #BUFBAS,R2         ;GET THE BASE OF THE DATA PATTERN BUFFER.
8754 034460  010204                        MOV      R2,R4
8755 034462  105303                6$:     DECB     R3                 ;GET THE NEXT BYTE OF THE DATA PATTERN.
8756 034464  110324                        MOVB     R3,(R4)+           ;WRITE A BYTE OF THE DATA PATTERN.
8757 034466  105703                        TSTB     R3                 ;CHECK FOR DONE WRITING DATA PATTERN.
8758 034470  001374                        BNE      6$                 ;DATA PATTERN DONE?  NO, LOOP TO DO NEXT BYTE.
8759                                                                   ;YES, WRITE 32 BYTE OVERFLOW REGION.
8760 034472  010205                        MOV      R2,R5              ;PREPARE SOURCE POINTER.
8761 034474  012700  000020                MOV      #16.,R0            ;PREPARE LOOP COUNTER.
8762 034500  012524                8$:     MOV      (R5)+,(R4)+        ;WRITE 2 BYTES OF THE OVERFLOW PATTERN.
8763 034502  005300                        DEC      R0                 ;COUNT THESE 2 BYTES.
8764 034504  001375                        BNE      8$                 ;16 WORDS WRITTEN?  NO, LOOP TO WRITE ANOTHER.
8765                                                                   ;YES, COMPLETE DATA PATTERN IS DONE.
8766 034506  012703  000400                MOV      #256.,R3           ;SET DATA PATTERN LENGTH TO 256.
8767                               ;+
8768                               ; Set the DUT RTS and DTR bits for the active lines.
8769                               ;-
8770 034512  012700  011000                MOV      #11000,R0          ;SPECIFY TO SET RTS AND DTR.
8771 034516  013705  002174                MOV      ACTLNS,R5          ;SPECIFY ACTIVE LINES.
8772 034522  004737  027234                JSR      PC,WTWLNC          ;SET DUT RTS AND DTR ON ALL ACTIVE LINES.
8773                               ;+
8774                               ; Wait for the operator to establish the modem connection.
8775                               ;   Prompt "TYPE <CR> WHEN MODEM LINK ESTABLISHED:"
8776                               ;-
8777 034526  012737  000001  002266         MOV     #1,GMANWD          ;SET UP DEFAULT ANSWER TO YES.
8778 034534                                 GMANIL  EMLMSG,GMANWD,1,YES
     034534  104443                                                                    TRAP    C$GMAN
     034536  000404                                                                    BR      10000$
     034540  002266                                                                    .WORD   GMANWD
     034542  000130                                                                    .WORD   T$CODE
     034544  013156                                                                    .WORD   EMLMSG
     034546  000001                                                                    .WORD   1
     034550                                                                   10000$:
8779                               ;+
8780                               ; Report the state of the modem status signals.
8781                               ; Set default of printing modem status after every data pattern.
8782                               ;-
8783 034550  004737  021614                JSR      PC,MSSRPT
8784 034554  012737  000001  002270         MOV     #1,PMSFLG
8785                               ;+
8786                               ; Ask operator for the number of data patterns to send.
8787                               ;   Prompt: "NUMBER OF 256 BYTE DATA PATTERNS TO SEND ON EACH SELECTED LINE
8788                               ;                  (1-255, 0=SEND UNTIL ↑C): (D)  1 ?"
8789                               ;-
8790 034562  012737  000001  002266 10$:    MOV     #1,GMANWD          ;SET DEFAULT NUMBER OF PATTERNS TO 1.
8791 034570                                 GMANID  NDPMSG,GMANWD,D,377,0,255,YES
```

```
          034570  104443                                                                            TRAP    C$GMAN
          034572  000406                                                                            BR      10001$
          034574  002266                                                                            .WORD   GMANWD
          034576  000052                                                                            .WORD   T$CODE
          034600  013306                                                                            .WORD   NDPMSG
          034602  000377                                                                            .WORD   377
          034604  000000                                                                            .WORD   T$LOLIM
          034606  000255                                                                            .WORD   T$HILIM
          034610
 8792 034610  013704  002266                              MOV     GMANWD,R4                 10001$:
 8793 034614  005005                                      CLR     R5                 ;CLEAR THE DATA PATTERN COUNTER.
 8794                                         ;*
 8795                                         ; Ask if modem status signals should be reported after each data pattern.
 8796                                         ;   Prompt: "PRINT MODEM STATUS SIGNAL REPORT AFTER EACH PATTERN: (L)  Y ?"
 8797                                         ;   Use last response as default (default of Yes the first time).
 8798                                         ;-
 8799 034616                                          GMANIL  PMSMSG,PMSFLG,1,YES
          034616  104443                                                                            TRAP    C$GMAN
          034620  000404                                                                            BR      10002$
          034622  002270                                                                            .WORD   PMSFLG
          034624  000130                                                                            .WORD   T$CODE
          034626  013434                                                                            .WORD   PMSMSG
          034630  000001                                                                            .WORD   1
          034632                                                                            10002$:
 8800
 8801                                         ;*
 8802                                         ; Set up the DUT and TX/RX variables.
 8803                                         ;     R1 - TX, RX LPR contents.
 8804                                         ;     R2 - Start address of data pattern to TX/RX.
 8805                                         ;     R3 - Length of data pattern.
 8806                                         ; Send the data.
 8807                                         ;*
 8808 034632  005205                          12$:    INC     R5                 ;COUNT THIS DATA PATTERN.
 8809 034634  004737  021324                          JSR     PC,MODSUP          ;SET UP THE DUT AND TX/RX VARIABLES.
 8810
 8811 034640  004737  023152                          JSR     PC,PUFIFO          ;PURGE THE DUT RECEIVE CHARACTER FIFO.
 8812 034644  103100                                  BCC     60$                ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
 8813
 8814 034646  004737  023426                          JSR     PC,PURRXB          ;PURGE THE RX CHAR BUFFER IN MEMORY.
 8815 034652  004737  021204                          JSR     PC,INIDMA          ;SEND THE FIRST BATCH OF DATA PATTERNS.
 8816 034656  012737  021306  005320                  MOV     #8902.,ERRNBR      ;SET ERROR NUMBER TO 8905.
 8817                                         ;*
 8818                                         ; This routine reports errors with numbers >>>>> 8902 thru 8907 <<<<<.
 8819                                         ;-
 8820 034664  004737  023462                          JSR     PC,RDCHRS          ;READ AND VERIFY THE RX CHARACTERS.
 8821 034670  012737  021314  005320                  MOV     #8908.,ERRNBR      ;SET ERROR NUMBER TO 8908.
 8822                                         ;*
 8823                                         ; This routine reports errors with numbers >>>>> 8908 thru 8911 <<<<<.
 8824                                         ;-
 8825 034676  004737  026450                          JSR     PC,TXRREP          ;REPORT FINAL ERRORS FROM RX/RX.
 8826                                         ;*
 8827                                         ; Report end of data pattern if allowed.
 8828                                         ;   "MODEM LOOPBACK TEST STATUS REPORT:   PATTERN #nnn (D) COMPLETED."
 8829                                         ; Report the modem status signal states if requested.
 8830                                         ;-
 8831 034702                                          PRINTX  #EDPFMT,R5
          034702  010546                                                                            MOV     R5,-(SP)
```

```
         034704  012746  007764                                                          MOV     #EDPFMT,-(SP)
         034710  012746  000002                                                          MOV     #2,-(SP)
         034714  010600                                                                  MOV     SP,R0
         034716  104415                                                                  TRAP    C$PNTX
         034720  062706  000006                                                          ADD     #6,SP
8832 034724  005737  002270               TST     PMSFLG      ;CHECK THE "PRINT MODEM STATUS" FLAG.
8833 034730  001402                        BEQ     14$         ;PRINT MODEM STATUS? NO, SKIP PRINTING.
8834 034732  004737  021614               JSR     PC,MSSRPT   ;REPORT THE MODEM STATUS.
8835                                    ;+
8836                                    ; If there are more data patterns to send, loop back to send again.
8937                                    ;-
8838 034736  005304                14$:   DEC     R4          ;COUNT THIS DATA PATTERN.
8839 034740  001403                        BEQ     16$         ;LAST DATA PAT SENT?  YES, PROMPT FOR EXIT.
8840 034742  100333                        BPL     12$         ;NO, CONTINOUS SENDING?  NO, SEND NEXT PAT.
8841 034744  005204                        INC     R4          ;YES, RESTORE PATTERN COUNTER.
8842 034746  000731                        BR      12$         ;GO TO SEND NEXT DATA PATTERN.
8843                                    ;+
8844                                    ; Prompt for exit of the test or sending of more data patterns.
8845                                    ;   Prompt: "EXIT THE TEST (N = LOOP BACK TO SEND MORE DATA): (L) Y ?"
8846                                    ;-
8847 034750  012737  000001  002266  16$:  MOV    #1,GMANWD    ;SET DEFAULT ANSWER TO YES.
8848 034756                             GMANIL  EXTMSG,GMANWD,1,YES
         034756  104443                                                                  TRAP    C$GMAN
         034760  000404                                                                  BR      10003$
         034762  002266                                                                  .WORD   GMANWD
         034764  000130                                                                  .WORD   T$CODE
         034766  013225                                                                  .WORD   EXTMSG
         034770  000001                                                                  .WORD   1
         034772                                                                  10003$:
8849 034772  023727  002266  000001               CMP     GMANWD,#1   ;CHECK OPERATOR RESPONSE.
8850 035000  001270                               BNE     10$         ;EXIT RESPONSE?  NO, LOOP TO SEND MORE DATA.
8851                                                                   ;NO, EXIT ROUTINE.
8852                                    ;+
8853                                    ; All done.  Have been told to exit.
8854                                    ;   Clear device DTR and RTS signals.
8855                                    ;   Disable interrupts.
8856                                    ;   Clear the interrupt vectors.
8857                                    ;   Report any necessary error summaries.
8858                                    ;-
8859 035002  005000                               CLR     R0          ;INDICATE TO CLEAR ALL LNCTRL BITS.
8860 035004  012705  000377                        MOV     #MAPLNS,R5  ;INDICATE TO CLEAR FOR ALL LINES.
8861 035010  004737  027234                        JSR     PC,WTWLNC   ;CLEAR ALL THE RTS AND DTR SIGNALS.
8862
8863 035014  106427  000240                        MTPS    #PRI05      ;DISABLE DEVICE INTERRUPTS.
8864 035020                                        CLRVEC  TXVECA      ;RETURN TX INT VECTOR TO UNUSED POOL.
         035020  013700  002172                                                          MOV     TXVECA,R0
         035024  104436                                                                  TRAP    C$CVEC
8865 035026                                        CLRVEC  RXVECA      ;RETURN RX INT VECTOR TO UNUSED POOL.
         035026  013700  002170                                                          MOV     RXVECA,R0
         035032  104436                                                                  TRAP    C$CVEC
8866
8867 035034  012737  021320  005320               MOV     #8912.,ERRNBR ;SELECT NUMBER 8912 FOR THE NEXT ERROR REPORT.
8868 035042  004737  024220                        JSR     PC,REPSMR   ;REPORT ERROR SUMMARIES IF CALLED FOR.
8869 035046  106427  000240                60$:    MTPS    #PRI05      ;DISABLE DEVICE INTERRUPTS.
8870 035052  005037  002222                        CLR     CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
8871 035056                                        ENDTST
         035056                                                                  L10031:
```

HARDWARE TEST   - MODLPB -

        035056 104401   TRAP   C$ETST

```
8873                                          .SBTTL  HARDWARE TEST            - KBECHO -
8874                                          ;++ ************************************************************
8875                                          ;*                      - Keyboard Echo Test -
8876                                          ;*      This is a test which puts UARTS for the active lines into remote
8877                                          ;*      loopback mode.  The active line UARTS are set up with a baudrate
8878                                          ;*      which is specified by the operator.  The test executes indefinitely
8879                                          ;*      until terminated by the operator.
8880                                          ;*
8881                                          ;*      This test can be used for looping back terminal keyboard input onto
8882                                          ;*      a terminal CRT or it can be used as a general loopback method for
8883                                          ;*      testing communications links to the DUT from the other end of the
8884                                          ;*      channel.  DTR and RTS are set on the selected lines during this
8885                                          ;*      test to allow the testing of modem links.
8886                                          ;*
8887                                          ;-- ************************************************************
8888 035060                                       BGNTST
     035060                                                                                       T6::
8889         000006                               TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
8890 035060   012737  000006  002224              MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.            (94)
8891                                          ;+
8892                                          ; Verify that the test should be performed.  Must have the following:
8893                                          ;       Keyboard Echo loopback selected.
8894                                          ;       Manual intervention allowed.
8895                                          ;-
8896 035066   123727  002176  000005              CMPB    LOPBCK,#5       ;TEST THE LOOPBACK TYPE INDICATOR.
8897 035074   001402                               BEQ     2$              ;KBD ECHO LPBCK SELECTED? YES, CONTINUE TEST.
8898 035076                                        EXIT    TST             ;NO, ABORT THE TEST.
     035076   104432                                                                               TRAP    C$EXIT
     035100   000204                                                                               .WORD   L10032-.
8899 035102                                   2$:   MANUAL                  ;CHECK FOR MANUAL INTERVENTION ALLOWED.
     035102   104450                                                                               TRAP    C$MANI
8900 035104                                        BCOMPLETE        4$     ;MANUAL INTERVENTION ALLOWED? YES, DO TEST.
     035104   103402                                                                               BCS     4$
8901 035106                                        EXIT    TST             ;NO, ABORT THE TEST.
     035106   104432                                                                               TRAP    C$EXIT
     035110   000174                                                                               .WORD   L10032-.
8902
8903 035112   012737  177777  002222   4$:   MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
8904 035120   012737  000001  005316         MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
8905 035126   012737  022271  005320         MOV     #9401.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
8906 035134   012737  013055  005322         MOV     #EM9401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
8907 035142   005037  002502                 CLR     ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
8908                                          ;+
8909                                          ; Reset the DUT to a known state, remove the status codes from the fifo.
8910                                          ;  Clear TX and RX interrupt enable bits in the CSR.
8911                                          ; This subroutine reports error >>>>> 9401 <<<<<.
8912                                          ;-
8913 035146   004737  017234                 JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
8914 035152   103402                          BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
8915 035154   000137  035300                 JMP     60$             ;RESET FAILURE, ABORT THIS TEST.
8916                                          ;+
8917                                          ; Print the test name.
8918                                          ;-
8919 035160                                       PRINTF  #EF8901,#EM9401
     035160   012746  013055                                                                       MOV     #EM9401,-(SP)
     035164   012746  006235                                                                       MOV     #EF8901,-(SP)
     035170   012746  000002                                                                       MOV     #2,-(SP)
```

```
              035174  010600                                                          MOV    SP,RC
              035176  104417                                                          TRAP   C$PNTF
              035200  062706  000006                                                  ADD    #6,SP
8920
8921
8922                                        ;+
8923                                        ; Set up the DUT UARTS with the proper line parameters.
8924                                        ;    Get the desired baudrate from the operator.
8925                                        ;    Calculate proper DUT LPR contents.
8926                                        ;    Set up the DUT LPR registers.
8927                                        ;    Get the proper DUT LNCTRL register contents.
8928                                        ;    Set up the DUT LNCTRL registers.
8929 035204  004737  020302                 ;-
8930 035210  010100                               JSR    PC,GETBDR
8931 035212  006301                               MOV    R1,R0
8932 035214  006301                               ASL    R1
8933 035216  006301                               ASL    R1
8934 035220  006301                               ASL    R1       ;GET DUPLICATE COPIES OF BAUDRATE CODE
8935 035222  050100                               ASL    R1       ;   IN THE UPPER BYTE OF THE NEW
8936 035224  000300                               BIS    R1,R0    ;   LPR CONTENTS.
8937 035226  042700  000377                       SWAB   R0
8938 035232  052700  000030                       BIC    #377,R0  ;SET UP 1 STOP BIT, NO PARITY, 8 BITS/CHAR
8939                                               BIS    #30,R0   ;   IN THE LPR CONTENTS.
8940 035236  013705  002174                       MOV    ACTLNS,R5  ;GET THE ACTIVE LINES BIT MAP.
8941 035242  004737  027264                       JSR    PC,WTWLPR  ;SET UP THE DUT LPR REGISTERS FOR ACTIVE LINES.
8942
8943 035246  012700  011304                       MOV    #11304,R0  ;SET UP DTR, RTS, REMOTE LPBK, AND RX ENABLE.
8944 035252  004737  027234                       JSR    PC,WTWLNC  ;SET UP THE DUT LNCTRL REGS FOR ACTIVE LINES.
8945
8946                                        ;+
8947                                        ; Wait for the operator to terminate the test.
8948                                        ;   Prompt "TYPE <CR> TO TERMINATE THE TEST:"
8949                                        ;-
8950 035256  012737  000001  002266                MOV    #1,GMANWD   ;SET UP DEFAULT ANSWER TO YES.
8951 035264                                         GMANIL TERMSG,GMANWD,1,YES
     035264  104443                                                                    TRAP   C$GMAN
     035266  000404                                                                    BR     10000$
     035270  002266                                                                    .WORD  GMANWD
     035272  000130                                                                    .WORD  T$CODE
     035274  013521                                                                    .WORD  TERMSG
     035276  000001                                                                    .WORD  1
     035300                                                                     10000$:
8952
8953
8954                                        ;+
8955                                        ; We got a response from the operator, so terminate the test.
                                            ;-
8956 035300  005037  002222                 60$:   CLR    CTRLCF      ;INDICATE THAT WE ARE NOT WITHIN A TEST.
8957 035304                                         ENDTST
     035304                                                                      L10032:
     035304  104401                                                                    TRAP   C$ETST
```

```
8959                                           .SBTTL  HARDWARE TEST              - SINGLC -
8960                                           ;++ ****************************************************************************
8961                                           ;*                    - Single Character Mode Test -
8962                                           ;*      This test verifies that the Device Under Test (DUT) will perform
8963                                           ;*      transmission and reception correctly using the single character
8964                                           ;*      mode interrupts.  The test is performed at 3 baudrates (slowest,
8965                                           ;*      middle, and highest) at all combinations of # of stop bits, # of
8966                                           ;*      bits per character, and types of parity using short data patterns.
8967                                           ;*      A high speed test is also performed at the highest baudrate with all
8968                                           ;*      combinations of line parameters using longer data patterns.
8969                                           ;*      This test is performed in internal loopback regardless of the type
8970                                           ;*      of loopback which is selected for the DUT in the hardware P-table.
8971                                           ;*
8972                                           ;-- ****************************************************************************
8973
8974 035306                                    BGNTST
     035306                                                                                    T7::
8975          000007                            TNUM == TNUM + 1         ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
8976 035306   012737 000007 002224             MOV     #TNUM,TSTNUM      ;SET UP THE TEST NUMBER.          (90)
8977 035314   012737 177777 002222             MOV     #-1,CTRLCF        ;INDICATE THAT WE ARE IN A TEST.
8978 035322   012737 000001 005316             MOV     #1,ERRTYP         ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
8979 035330   012737 021451 005320             MOV     #9001.,ERRNBR     ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
8980 035336   012737 011245 005322             MOV     #EM9001,ERRMSG    ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
8981 035344   005037 002502                    CLR     ERSMRF            ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
8982
8983                                           ;+
8984                                           ; Reset the DUT to a known state, remove the status codes from the fifo.
8985                                           ;  Clear TX and RX interrupt enable bits in the CSR.
8986                                           ; This subroutine reports error >>>>> 9001 <<<<<.
                                               ;-
8987 035350   004737 017234                    JSR     PC,CLNRST         ;RESET THE DHV11-M. REPORT ANY ERRORS FOUND.
8988 035354   103402                            BCS     .+6               ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
8989 035356   000137 036010                    JMP     60$               ;EXIT THE TEST, FATAL ERROR WAS FOUND.
8990 035362   012737 021452 005320             MOV     #9002.,ERRNBR     ;SET THE ERROR NUMBER.
8991
8992                                           ;+
8993                                           ; Set up for Transmit and Receive interrupts.
8994                                           ;-
8995 035370   106427 000240                    MTPS    #PRIO5            ;DISABLE DEVICE INTERRUPTS.
8996 035374                                     SETVEC  TXVECA,#TXSCHR,#PRIO5   ;SELECT SINGLE CHAR TX INT SERVICE RTN.
     035374   012746 000240                                                         MOV     #PRIO5,-(SP)
     035400   012746 027726                                                         MOV     #TXSCHR,-(SP)
     035404   013746 002172                                                         MOV     TXVECA,-(SP)
     035410   012746 000003                                                         MOV     #3,-(SP)
     035414   104437                                                                TRAP    C$SVEC
     035416   062706 000010                                                         ADD     #10,SP
8997 035422                                     SETVEC  RXVECA,#RXCHRS,#PRIO5   ;SELECT RX INT SERVICE RTN.
     035422   012746 000240                                                         MOV     #PRIO5,-(SP)
     035426   012746 027364                                                         MOV     #RXCHRS,-(SP)
     035432   013746 002170                                                         MOV     RXVECA,-(SP)
     035436   012746 000003                                                         MOV     #3,-(SP)
     035442   104437                                                                TRAP    C$SVEC
     035444   062706 000010                                                         ADD     #10,SP
8998 035450   106427 000000                    MTPS    #PRIO0            ;ALLOW INTERRUPTS.
8999                                           ;+
9000                                           ; Clear the error counter table.
9001                                           ; This table will accumulate error count totals for each line during this test.
9002                                           ;-
```

```
 9003 035454  012700  003304              MOV     #ERCNTB,R0
 9004 035460  004737  017256              JSR     PC,CLR16W          ;CLEAR THE RX ERROR COUNTERS TABLE.
 9005                              ;+
 9006                              ; Transmit and receive short data pattern in all combinations of 3 baudrates,
 9007                              ; all #s of stop bits, all #s of bits per character, and all types of parity.
 9008                              ;   Set up line control parameters for single char mode DUT operation.
 9009                              ;-
 9010 035464  005037  002466              CLR     GPRSOB             ;CLEAR THE GPR SAVE AREA R1 STORAGE TO INDICATE
 9011                                                                ;   THAT THIS IS THE FIRST TIME IN GETLP1.
 9012 035470  004737  025372      2$:     JSR     PC,SWAPO           ;SWAP GPRS WITH GPR SAVE AREA ZERO FOR GETLP1.
 9013 035474  004737  020512              JSR     PC,GETLP1          ;GET NEXT SET OF LPR CONTENTS, OR CARRY CLEAR.
 9014 035500  004737  025372              JSR     PC,SWAPO           ;SWAP BACK GPRS AND GPR SAVE AREA ZERO.
 9015 035504  103055                      BCC     4$                 ;EXIT LOOP IF ALL COMBINATIONS OF LPRS DONE.
 9016 035506  010001                      MOV     R0,R1              ;PASS THE LPR CONTENTS TO GETTIM AND VANSUP.
 9017 035510  004737  021014              JSR     PC,GETTIM          ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
 9018 035514  012702  005132              MOV     #SDPBAS,R2         ;SET UP POINTER TO START OF SHORT DATA PATTERN.
 9019 035520  012703  000020              MOV     #SDPEND-SDPBAS,R3  ;SET UP THE DATA PATTRN LENGTH.
 9020 035524  012704  000001              MOV     #1,R4              ;SPECIFY TO SEND 1 DATA PATTERN TO EACH LINE.
 9021 035530  004737  026762              JSR     PC,VANSUP          ;SET UP "VANILLA FLAVORED" TX/RX.
 9022 035534  004737  016422              JSR     PC,CHRMSK          ;GET THE BIT MASK OF UNUSED TX/RX BITS.
 9023 035540  004737  023152              JSR     PC,PUFIFO          ;PURGE THE DUT RECEIVE CHARACTER FIFO.
 9024 035544  004737  023426              JSR     PC,PURRXB          ;PURGE THE RX CHAR BUFFER IN MEMORY.
 9025 035550  004737  021072              JSR     PC,INICHR          ;SEND INITIAL CHARS TO ALL ACTIVE LINES.
 9026 035554  012737  021452  005320      MOV     #9002.,ERRNBR      ;SET THE ERROR NUMBER TO 9002.
 9027                              ;+
 9028                              ; The following routine reports the error with numbers 9002 thru 9008.
 9029                              ;-
 9030 035562                              DELAY   1                  ;Pause, allow transmition $$$
      035562  012727  000001                                                             MOV    #1,(PC)+
      035566  000000                                                                     .WORD  0
      035570  013727  002116                                                             MOV    L$DLY,(PC)+
      035574  000000                                                                     .WORD  0
      035576  005367  177772                                                             DEC    -6(PC)
      035602  001375                                                                     BNE    .-4
      035604  005367  177756                                                             DEC    -22(PC)
      035610  001367                                                                     BNE    .-20
 9031 035612  004737  023462              JSR     PC,RDCHRS          ;READ AND VERIFY THE RX CHARACTERS.
 9032 035616  012737  021461  005320      MOV     #9009.,ERRNBR      ;SET THE ERROR NUMBER TO 9009.
 9033                              ;+
 9034                              ; The following routine reports the error with numbers 9009 thru 9012.
 9035                              ;-
 9036 035624  004737  026450              JSR     PC,TXRREP          ;REPORT FINAL ERRORS FROM RX/RX.
 9037                              ;+
 9038                              ; Loop to select the next baudrate and line parameters.
 9039                              ;-
 9040 035630  000717                      BR      2$
 9041                              ;+
 9042                              ; Transmit and receive long data patterns at maximum baudrate and all
 9043                              ; combinations of all numbers of stop bits, all numbers of bits per character,
 9044                              ; and all types of parity.
 9045                              ;-
 9046                              ;+
 9047                              ; Initialize the long data pattern and parameters for the SCHTST call.
 9048                              ;-
 9049 035632  012737  021465  005320      MOV     #9013.,ERRNBR      ;SET THE ERROR NUMBER TO 9013.
 9050 035640  012702  003604      4$:     MOV     #BUFBAS,R2         ;INITIALIZE THE LONG DATA
 9051 035644  005003                      CLR     R3                 ;  PATTERN IN THE GENERAL
```

```
9052 035646 110322          6$:     MOVB    R3,(R2)+            ; DATA BUFFER TO A 256
9053 035650 005203                  INC     R3                 ; BYTE PATTERN COUNTING
9054 035652 020227 004204           CMP     R2,#BUFMID         ; FROM ZERO TO 255.
9055 035656 103773                  BLO     6$                 ;
9056                         ;+
9057                         ; Initialize for, and get the LPR contents.
9058                         ;-
9059 035660 005037 002466           CLR     GPRS0B             ;CLEAR THE GPR SAVE AREA R1 STORAGE TO INDICATE
9060                                                           ;  THAT THIS IS THE FIRST TIME IN GETLP2.
9061 035664 004737 025372   8$:     JSR     PC,SWAPO           ;SWAP GPRS WITH GPR SAVE AREA ZERO FOR GETLP1.
9062 035670 004737 020664           JSR     PC,GETLP2          ;GET NEXT SET OF LPR CONTENTS, OR CARRY CLEAR.
9063 035674 004737 025372           JSR     PC,SWAPO           ;SWAP BACK GPRS AND GPR SAVE AREA ZERO.
9064 035700 103036                  BCC     10$                ;EXIT LOOP IF ALL COMBINATIONS OF LPRS DONE.
9065 035702 010001                  MOV     R0,R1              ;PASS THE LPR CONTENTS TO GETTIM AND VANSUP.
9066 035704 004737 021014           JSR     PC,GETTIM          ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
9067 035710 012702 003604           MOV     #BUFBAS,R2         ;SET UP POINTER TO START OF SHORT DATA PATTERN.
9068 035714 012703 000400           MOV     #BUFMID-BUFBAS,R3  ;SET UP THE DATA PATTRN LENGTH.
9069 035720 012704 000001           MOV     #1,R4              ;SPECIFY TO SEND 1 DATA PATTERN TO EACH LINE.
9070 035724 004737 026762           JSR     PC,VANSUP          ;SET UP "VANILLA FLAVORED" TX/RX.
9071 035730 004737 016422           JSR     PC,CHRMSK          ;GET THE BIT MASK OF UNUSED TX/RX BITS.
9072 035734 004737 023152           JSR     PC,PUFIFO          ;PURGE THE DUT RECEIVE CHARACTER FIFO.
9073 035740 004737 023426           JSR     PC,PURRXB          ;PURGE THE RX CHAR BUFFER IN MEMORY.
9074 035744 004737 021072           JSR     PC,INICHR          ;SEND INITIAL CHARS TO ALL ACTIVE LINES.
9075 035750 012737 021465 005320     MOV     #9013.,ERRNBR     ;SET THE ERROR NUMBER TO 9013.
9076
9077                         ;+
                             ; The following routine reports the error with numbers 9013 thru 9018.
9078                         ;-
9079 035756 004737 023462           JSR     PC,RDCHRS          ;READ AND VERIFY THE RX CHARACTERS.
9080 035762 012737 021473 005320     MOV     #9019.,ERRNBR     ;SET THE ERROR REPORT NUMBER TO 9019.
9081
9082                         ;+
                             ; The following routine reports the error with numbers 9019 thru 9022.
9083                         ;-
9084 035770 004737 026450           JSR     PC,TXRREP          ;REPORT FINAL ERRORS FROM RX/RX.
9085
9086                         ;+
                             ; Loop to select the next baudrate and line parameters.
9087                         ;-
9088 035774 000733                  BR      8$
9089 035776 012737 021477 005320 10$: MOV   #9023.,ERRNBR      ;SELECT NUMBER 9023 FOR THE NEXT ERROR REPORT.
9090 036004 004737 024220           JSR     PC,REPSMR          ;REPORT ERROR SUMMARIES IF CALLED FOR.
9091
9092                         ;+
9093                         ; All done, have completed the test.
9094                         ;   Disable interrupts.
9095                         ;   Clear the interrupt vectors.
9096                         ;-
9097 036010 106427 000240   60$:    MTPS    #PRIO5             ;DISABLE DEVICE INTERRUPTS.
9098 036014                         CLRVEC  TXVECA             ;RETURN TX INT VECTOR TO UNUSED POOL.
     036014 013700 002172                                                             MOV     TXVECA,R0
     036020 104436                                                                    TRAP    C$CVEC
9099 036022                         CLRVEC  RXVECA             ;RETURN RX INT VECTOR TO UNUSED POOL.
     036022 013700 002170                                                             MOV     RXVECA,R0
     036026 104436                                                                    TRAP    C$CVEC
9100 036030 005037 002222           CLR     CTRLCF             ;INDICATE THAT WE ARE NOT WITHIN A TEST.
9101
9102 036034                         ENDTST
     036034                                                                  L10033:
     036034 104401                                                                    TRAP    C$ETST
```

```
9104                                    .SBTTL  HARDWARE TEST          - DMA -
9105                                    ;++ **********************************************************************
9106                                    ;*                         - DMA Mode Test -
9107                                    ;*       This test verifies that the Device Under Test (DUT) will perform
9108                                    ;*       transmission and reception correctly using the DMA mode transmission.
9109                                    ;*       The test is performed at all baudrates (except 50 baud), 8 bits per
9110                                    ;*       character, 1 stop bit, and with parity checking (both odd and even).
9111                                    ;*       A high speed test is also performed at the highest 3 baudrates at
9112                                    ;*       both 5 and 8 bits per character, 1 stop bit, and no parity checking.
9113                                    ;*       This test is performed with the type of loopback which was specified
9114                                    ;*       in the DUT hardware P-table on all active lines.
9115                                    ;*
9116                                    ;-- **********************************************************************
9117 036036                                 BGNTST
     036036
                                                                                          T8::
9118        000010                           TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
9119 036036 012737 000010 002224             MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (91)
9120 036044 012737 177777 002222             MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
9121 036052 012737 000001 005316             MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
9122 036060 012737 021615 005320             MOV     #9101.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
9123 036066 012737 012443 005322             MOV     #EM9101,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
9124 036074 005037 002502                    CLR     ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
9125                                    ;+
9126                                    ; Reset the DUT to a known state, remove the status codes from the fifo.
9127                                    ;  Clear TX and RX interrupt enable bits in the CSR.
9128                                    ; This subroutine reports error >>>>> 9101 <<<<<.
9129                                    ;-
9130 036100 004737 017234                    JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
9131 036104 103402                            BCS     2$              ;SKIP AROUND TEST EXIT IF NO FATAL ERROR FOUND.
9132 036106 000137 036634                     JMP     60$             ;RESET FAILURE, ABORT THIS TEST.
9133                                    ;+
9134                                    ; Set up for Transmit interrupts.
9135                                    ;-
9136 036112 106427 000240             2$:    MTPS    #PRIO5          ;DISABLE DEVICE INTERRUPTS.
9137 036116                                   SETVEC  TXVECA,#TXDMA,#PRIO5    ;SELECT DMA TX INT SERVICE RTN.
     036116 012746 000240                                                                         MOV    #PRIO5,-(SP)
     036122 012746 027574                                                                         MOV    #TXDMA,-(SP)
     036126 013746 002172                                                                         MOV    TXVECA,-(SP)
     036132 012746 000003                                                                         MOV    #3,-(SP)
     036136 104437                                                                                TRAP   C$SVEC
     036140 062706 000010                                                                         ADD    #10,SP
9138 036144                                   SETVEC  RXVECA,#RXCHRS,#PRIO5   ;SELECT RX INT SERVICE RTN.
     036144 012746 000240                                                                         MOV    #PRIO5,-(SP)
     036150 012746 027364                                                                         MOV    #RXCHRS,-(SP)
     036154 013746 002170                                                                         MOV    RXVECA,-(SP)
     036160 012746 000003                                                                         MOV    #3,-(SP)
     036164 104437                                                                                TRAP   C$SVEC
     036166 062706 000010                                                                         ADD    #10,SP
9139 036172 106427 000000                     MTPS    #PRIO0          ;ALLOW INTERRUPTS.
9140                                    ;+
9141                                    ; Transmit and receive short data pattern at all baudrates,
9142                                    ;   with 8 bits per character, 1 stop bit, and both types of parity.
9143                                    ;   Both line groups (LGPRS) TX and RX with the same parameters.
9144                                    ;-
9145 036176 012700 003304                     MOV     #ERCNTB,R0
9146 036202 004737 017256                     JSR     PC,CLR16W       ;CLEAR THE RX ERROR COUNTERS TABLE.
9147 036206 012701 010470                     MOV     #10470,R1       ;SET UP LPR CONTENTS FOR TX/RX AT 75 BAUD.
```

```
 9148 036212  004737  021014        4$:    JSR     PC,GETTIM       ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
 9149 036216  012702  005156               MOV     #SDP2B,R2       ;SET UP THE START ADR OF THE DATA PATTERN.
 9150 036222  012703  000020               MOV     #SDP2E-SDP2B,R3 ;SET UP THE DATA PATTERN LENGTH.
 9151 036226  012704  000001               MOV     #1,R4           ;SPECIFY TO SEND 1 DATA PATTERN TO EACH LINE.
 9152 036232  004737  026762               JSR     PC,VANSUP       ;SET UP "VANILLA FLAVORED" TX/RX.
 9153 036236  012737  177400  002226       MOV     #177400,IBM     ;FORM BIT MAP OF UNUSED TX/RX BITS.
 9154 036244  012737  021616  005320       MOV     #9102.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 9102.
 9155                                ;+
 9156                                ; This routine reports errors with numbers >>>>> 9102 thru 9104 <<<<<.
 9157                                ;-
 9158 036252  004737  023234               JSR     PC,PUFIFR       ;PURGE THE DUT RECEIVE CHARACTER FIFO.
 9159 036256  103166                        BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
 9160
 9161 036260  004737  023426               JSR     PC,PURRXB       ;PURGE THE RX CHAR BUFFER IN MEMORY.
 9162 036264  004737  021204               JSR     PC,INIDMA       ;SEND THE FIRST BATCH OF DATA PATTERNS.
 9163 036270  012737  021621  005320       MOV     #9105.,ERRNBR   ;SET ERROR NUMBER TO 9105.
 9164                                ;+
 9165                                ; This routine reports errors with numbers >>>>> 9105 thru 9110 <<<<<.
 9166                                ;-
 9167 036276  004737  023462               JSR     PC,RDCHRS       ;READ AND VERIFY THE RX CHARACTERS.
 9168 036302  012737  021627  005320       MOV     #9111.,ERRNBR   ;SET ERROR NUMBER TO 9111.
 9169                                ;+
 9170                                ; This routine reports errors with numbers >>>>> 9111 thru 9114 <<<<<.
 9171                                ;-
 9172 036310  004737  026450               JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM RX/RX.
 9173                                ;+
 9174                                ; Toggle the parity type bit specifier in the TX/RX setup parameters.
 9175                                ; Select the next baudrate and perform the test again if not done.
 9176                                ;-
 9177 036314  010100                        MOV     R1,R0           ;COMPLEMENT THE PARITY TYPE
 9178 036316  042701  000100               BIC     #100,R1         ;   BIT IN THE TX/RX LPR SETUP
 9179 036322  005100                        COM     R0              ;   PARAMETER LEAVING THE
 9180 036324  042700  177677               BIC     #177677,R0      ;   OTHER LPR PARAMETER
 9181 036330  050001                        BIS     R0,R1           ;   BITS UNCHANGED.
 9182 036332  062701  010400               ADD     #10400,R1       ;SELECT THE NEXT BAUDRATE.
 9183 036336  103325                        BCC     4$              ;LOOP TO TX/RX AGAIN IF NOT PAST LAST BAUDRATE.
 9184                                ;+
 9185                                ; Perform wide open DMA test.
 9186                                ; Transmit and receive 512 byte data patterns at all combinations of 9.6K,
 9187                                ; 19.2K, and 38.4K buadrates and 5 and 8 bits per character.  Use 1 stop bit
 9188                                ; and no parity generation or detection.
 9189                                ;-
 9190                                     ;+
 9191                                     ; Initialize the 512 byte pattern and the various data pattern pointers.
 9192                                     ;-
 9193 036340  005001                        CLR     R1              ;CLEAR THE DATA BYTE COUNTER.
 9194 036342  012702  003604               MOV     #BUFBAS,R2      ;GET THE BASE OF THE DATA PATTERN BUFFER.
 9195 036346  110122              6$:       MOVB    R1,(R2)+        ;WRITE A BYTE OF THE DATA PATTERN.
 9196 036350  105201                        INCB    R1              ;GET THE NEXT BYTE FOR THE DATA PATTERN.
 9197 036352  001375                        BNE     6$              ;LOOP UNTIL FIRST 1/2 OF PATTERN IS DONE.
 9198 036354  105301              8$:       DECB    R1              ;GET THE NEXT BYTE FOR THE DATA PATTERN.
 9199 036356  110122                        MOVB    R1,(R2)+        ;WRITE A BYTE OF THE DATA PATTERN.
 9200 036360  105701                        TSTB    R1              ;CHECK FOR DONE WRITING DATA PATTERN.
 9201 036362  001374                        BNE     8$              ;LOOP IF DATA PATTERN IS NOT DONE.
 9202 036364  110122              10$:      MOVB    R1,(R2)+        ;WRITE A BYTE OF THE 32 BYTE OVERFLOW REGION.
 9203 036366  005201                        INC     R1              ;COUNT THIS BYTE.
 9204 036370  020127  000040               CMP     R1,#32.         ;TEST FOR 32 BYTES WRITTEN.
```

```
9205 036374 001373                          BNE     10$                 ;LOOP UNTIL 32 BYTES ARE WRITTEN.
9206                                    ;+
9207                                    ; Prepare to loop on the 3 different baudrates (9.6K, 19.2K, and 38.4K).
9208                                    ;-
9209 036376 012705 005074               MOV     #DLPRTB,R5          ;GET THE BASE ADR OF THE DMA BAUDRATE TABLE.
9210                                    ;+
9211                                    ; Specify the proper baudrate.
9212                                    ; Specify 8 bits per character.
9213                                    ; Perform DMA transmission and reception of 512 byte data pattern.
9214                                    ;-
9215                                        ;+
9216                                        ; The following routine reports the error with numbers 914 thru 921.
9217                                        ;    LPR CHANGE bit error flags may be set by this subroutine.
9218                                        ;-
9219 036402 012501                12$:   MOV     (R5)+,R1            ;SET UP LPR PARAM AT NEXT BAUD, 8 BITS/CHAR.
9220 036404 004737 021014               JSR     PC,GETTIM           ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
9221 036410 012702 003604               MOV     #BUFBAS,R2          ;SET UP THE START ADR OF THE DATA PATTERN.
9222 036414 012703 001000               MOV     #512.,R3            ;SET UP THE DATA PATTERN LENGTH.
9223 036420 012704 000001               MOV     #1,R4               ;SPECIFY TO SEND 1 DATA PATTERN TO EACH LINE.
9224 036424 004737 026762               JSR     PC,VANSUP           ;SET UP "VANILLA FLAVORED" TX/RX.
9225 036430 012737 177400 002226         MOV     #177400,IBM         ;FORM BIT MAP OF UNUSED BITS FOR 8 BITS/CHAR.
9226 036436 012737 021633 005320         MOV     #9115.,ERRNBR       ;SET ERROR NUMBER TO 9115.
9227                                        ;+
9228                                        ; This routine reports erros with numbers >>>>> 9115 thru 9117 <<<<<.
9229                                        ;-
9230 036444 004737 023234               JSR     PC,PUFIFR           ;PURGE THE DUT RECEIVE CHARACTER FIFO.
9231 036450 103071                       BCC     60$                 ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
9232 036452 012737 021636 005320         MOV     #9118.,ERRNBR       ;SET ERROR NUMBER TO 9118.
9233
9234 036460 004737 023426               JSR     PC,PURRXB           ;PURGE THE RX CHAR BUFFER IN MEMORY.
9235 036464 004737 021204               JSR     PC,INIDMA           ;SEND THE FIRST BATCH OF DATA PATTERNS.
9236                                    ;+
9237                                    ; This routine reports the error with numbers >>>>> 9118 thru 9123 <<<<<.
9238                                    ;-
9239 036470 004737 023462               JSR     PC,RDCHRS           ;READ AND VERIFY THE RX CHARACTERS.
9240 036474 012737 021644 005320         MOV     #9124.,ERRNBR       ;SET ERROR NUMBER TO 9124.
9241                                    ;+
9242                                    ; This routine reports errors with numbers >>>>> 9124 thru 9127 <<<<<.
9243                                    ;-
9244 036502 004737 026450               JSR     PC,TXRREP           ;REPORT FINAL ERRORS FROM RX/RX.
9245 036506 012737 021650 005320         MOV     #9128.,ERRNBR       ;SET ERROR NUMBER TO 9128.
9246                                    ;+
9247                                    ; Specify 5 bits per character.
9248                                    ; Perform DMA transmission and reception of 512 byte data pattern.
9249                                    ;-
9250 036514 042701 000030               BIC     #30,R1              ;SET UP CHAR LENGTH PARAM TO 5 BITS/CHAR.
9251 036520 004737 026762               JSR     PC,VANSUP           ;SET UP "VANILLA FLAVORED" TX/RX.
9252 036524 012737 177740 002226         MOV     #177740,IBM         ;FORM BIT MAP OF UNUSED BITS FOR 5 BITS/CHAR.
9253                                        ;+
9254                                        ; This  routine reports the error with numbers >>> 9128 thru 9131 <<<.
9255                                        ;-
9256 036532 004737 023234               JSR     PC,PUFIFR           ;PURGE THE DUT RECEIVE CHARACTER FIFO.
9257 036536 103036                       BCC     60$                 ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
9258 036540 012737 021654 005320         MOV     #9132.,ERRNBR       ;SET THE ERROR REPORT NUMBER TO 9132.
9259
9260 036546 004737 023426               JSR     PC,PURRXB           ;PURGE THE RX CHAR BUFFER IN MEMORY.
9261 036552 004737 021204               JSR     PC,INIDMA           ;SEND THE FIRST BATCH OF DATA PATTERNS.
```

```
9262                                      ;+
9263                                      ; This routine reports the error with numbers >>>>> 9132 thru 9137 <<<<<.
9264                                      ;-
9265 036556  004737  023462                  JSR     PC,RDCHRS           ;READ AND VERIFY THE RX CHARACTERS.
9266 036562  012737  021662  005320          MOV     #9138.,ERRNBR       ;SET ERROR NUMBER TO 9138.
9267                                      ;+
9268                                      ; This routine reports the error with numbers >>>>> 9138 thru 9141 <<<<<.
9269                                      ;-
9270 036570  004737  026450                  JSR     PC,TXRREP           ;REPORT FINAL ERRORS FROM RX/RX.
9271 036574  020527  005102                  CMP     R5,#DLPRTE          ;COMPARE DMA BAUDRATE TABLE PTR WITH TABLE END.
9272 036600  103700                           BLO     12$                 ;LOOP IF NOT ALL BAUDRATES DONE YET.
9273                                  ;+
9274                                  ; All done.  Have either run out of active lines, or completed the test.
9275                                  ;    Disable interrupts.
9276                                  ;    Clear the interrupt vectors.
9277                                  ;-
9278 036602  106427  000240                  MTPS    #PRIO5              ;DISABLE DEVICE INTERRUPTS.
9279 036606                                   CLRVEC  TXVECA              ;RETURN TX INT VECTOR TO UNUSED POOL.
     036606  013700  002172                                                       MOV     TXVECA,R0
     036612  104436                                                                TRAP    C$CVEC
9280 036614                                   CLRVEC  RXVECA              ;RETURN RX INT VECTOR TO UNUSED POOL.
     036614  013700  002170                                                       MOV     RXVECA,R0
     036620  104436                                                                TRAP    C$CVEC
9281
9282 036622  012737  021666  005320          MOV     #9142.,ERRNBR       ;SELECT NUMBER 9142 FOR THE NEXT ERROR REPORT.
9283 036630  004737  024220                  JSR     PC,REPSMR           ;REPORT ERROR SUMMARIES IF CALLED FOR.
9284 036634  106427  000240           60$:   MTPS    #PRIO5              ;DISABLE DEVICE INTERRUPTS.
9285 036640  005037  002222                  CLR     CTRLCF              ;INDICATE THAT WE ARE NOT WITHIN A TEST.
9286 036644                                   ENDTST
     036644                                                              L10034:
     036644  104401                                                                TRAP    C$ETST
```

```
9288                                    .SBTTL   HARDWARE TEST              - SPLSPD -
9289                                    ;+********************************************************************************
9290                                    ;*                       - SPLIT SPEED TEST -
9291                                    ;*       This test is used to verify the split speed capabilities of the DHV11-M,
9292                                    ;*       and the correct operation of the A & B baud rate group selection.
9293                                    ;*       The test uses three sets of baud rates (38.4,50; 1200,75; 2000,2400).
9294                                    ;*       This test will only execute if the staggered loopback mode is selected.
9295                                    ;*       The special staggered loopback BERG connector must be fitted.
9296                                    ;*
9297                                    ;--********************************************************************************
9298 036646                                 BGNTST
     036646                                                                        T9::
9299 036646  123727  002176  000002        CMPB     LOPBCK,#2          ;CHECK MODE SELECTED.
9300 036654  001402                         BEQ      2$                 ;DO NOT EXIT IF STAGGERD LOPBCK MODE SELECTED.
9301 036656  000137  037264                 JMP      60$                ;EXIT THIS TEST.
9302 036662  000011                  2$:    TNUM == TNUM + 1            ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
9303 036662  012737  000011  002224         MOV      #TNUM,TSTNUM       ;SET UP THE TEST NUMBER.         (92)
9304 036670  012737  177777  002222         MOV      #-1,CTRLCF         ;INDICATE THAT WE ARE IN A TEST.
9305 036676  012737  000001  005316         MOV      #1,ERRTYP          ;SET ERROR TYPE IN ERROR TABLE.
9306 036704  012737  021761  005320         MOV      #9201.,ERRNBR      ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
9307 036712  012737  012642  005322         MOV      #EM9201,ERRMSG     ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
9308 036720  005037  002502                 CLR      ERSMRF             ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
9309                                    ;+
9310                                    ; Reset the DUT to a known state, remove status codes from the fifo.
9311                                    ;  Clear TX and  RX interrupt enable bits.
9312                                    ; This subroutine reports error >>>>> 9201 <<<<<.
9313                                    ;-
9314 036724  004737  017234                 JSR      PC,CLNRST          ;RESET THE DUT.
9315 036730  103155                          BCC      60$                ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
9316                                    ;+
9317                                    ; Disable all interrupts.
9318                                    ; Set up  DMA TX and RX interrupt service routines.
9319                                    ;-
9320 036732  106427  000240                 MTPS     #PRI05             ;DISABLE DEVICE INTERRUPTS.
9321 036736                                  SETVEC   TXVECA,#TXDMA,#PRI05    ;SELECT DMA TX INT SERVICE RTN.
     036736  012746  000240                                                               MOV    #PRI05,-(SP)
     036742  012746  027574                                                               MOV    #TXDMA,-(SP)
     036746  013746  002172                                                               MOV    TXVECA,-(SP)
     036752  012746  000003                                                               MOV    #3,-(SP)
     036756  104437                                                                       TRAP   C$SVEC
     036760  062706  000010                                                               ADD    #10,SP
9322 036764                                  SETVEC   RXVECA,#RXCHRS,#PRI05   ;SELECT RX INT SERVICE RTN.
     036764  012746  000240                                                               MOV    #PRI05,-(SP)
     036770  012746  027364                                                               MOV    #RXCHRS,-(SP)
     036774  013746  002170                                                               MOV    RXVECA,-(SP)
     037000  012746  000003                                                               MOV    #3,-(SP)
     037004  104437                                                                       TRAP   C$SVEC
     037006  062706  000010                                                               ADD    #10,SP
9323 037012  106427  000000                 MTPS     #PRI00             ;ALLOW INTERRUPTS.
9324                                    ;+
9325                                    ; Enable transmitters on all lines.
9326                                    ;-
9327 037016  012705  000377                 MOV      #MAPLNS,R5         ;PASS ACTIVE LINE BIT MAP.
9328 037022  004737  025646                 JSR      PC,TXENBL          ;ENABLE TRANSMISSIONS ON ALL LINES.
9329                                    ;+
9330
9331                                    ; Clear error table prior to perfoming TX/RX test.
```

```
 9332
 9333 037026  012700  003304              MOV     @ERCNTB,R0      ;GET THE BASE ADDRESS OF THE ERROR COUNTER TBL.
 9334 037032  004737  017256              JSR     PC,CLR16W       ;CLEAR THE RX ERROR COUNTERS TABLE.
 9335
 9336                               ;+
 9337                               ; Perform Split speed DMA TX and RX on all selected lines at the following
 9338                               ; baud rates.
 9339                               ; 38.4K, 50 ; 1200, 75 ; 2000, 2400.
 9340                               ;-
 9341                               ;+
 9342                               ; Initialise DMA TX/RX parameters in the control block fr each of the baud
 9343                               ; rates mentioned above.
 9344                               ;  8 bits/char,1 stop bits,odd parity.
 9345                               ;-
 9346 037036  012705  005102              MOV     @SPLPRB,R5      ;GET BASE ADDRESS OF LPR PARAMETER TABLE.
 9347 037042  012500          4$:         MOV     (R5)+,R0        ;GET LPR CONTENTS FOR LINGRP II.
 9348 037044  012501                      MOV     (R5)+,R1        ;GET LPR CONTENTS FOR LINGRP I.
 9349 037046  004737  021014              JSR     PC,GETTIM       ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
 9350 037052  012702  005156              MOV     @SDP2B,R2       ;SET UP THE START ADR OF THE DATA PATTERN.
 9351 037056  012503                      MOV     (R5)+,R3        ;GET NUMBER OF REPEAT TRANSMISSION ON LINGRPII.
 9352 037060  012504                      MOV     (R5)+,R4        ;GET NUMBER OF REPEAT TRANSMISSION ON LINGRPI.
 9353 037062  004737  025052              JSR     PC,SPLSUP       ;SET UP CONTROL BLOCK ETC, FOR TX/RX.
 9354 037066  012737  021762  005320      MOV     #9202.,ERRNBR   ;SET THE ERROR NUMBER TO 9202.
 9355
 9356                               ;+
 9357                               ; This routine reports errors with numbers >>>>> 9202 thru 9204 <<<<<.
                                    ;-
 9358 037074  004737  023234              JSR     PC,PUFIFR       ;PURGE THE DUT RECEIVE CHARACTER FIFO.
 9359 037100  103071                      BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
 9360 037102  012737  021765  005320      MOV     #9205.,ERRNBR   ;SET ERROR NUMBER TO 9205.
 9361
 9362 037110  004737  023426              JSR     PC,PURRXB       ;PURGE THE RX CHAR BUFFER IN MEMORY.
 9363 037114  004737  021204              JSR     PC,INIDMA       ;SEND THE FIRST BATCH OF DATA PATTERNS.
 9364                               ;+
 9365                               ; This routine reports errors with numbers >>>>> 9205 thru 9210 <<<<<.
 9366                               ;-
 9367 037120  004737  023462              JSR     PC,RDCHRS       ;READ AND VERIFY THE RX CHARACTERS.
 9368 037124  012737  021773  005320      MOV     #9211.,ERRNBR   ;SET THE ERROR NUMBER TO 9211.
 9369                               ;+
 9370                               ; This routine reports errors with numbers >>>>> 9211 thru 9214 <<<<<.
 9371                               ;-
 9372 037132  004737  026450              JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM RX/RX.
 9373 037136  012737  021777  005320      MOV     #9215.,ERRNBR   ;SET ERROR NUMBER TO 9215.
 9374                               ;+
 9375                               ; Swap parameters to allow for both channels to be exercised.
 9376                               ;-
 9377 037144  010246                      MOV     R2,-(SP)        ;PUSH THE START ADDRESS ONTO THE STACK.
 9378 037146  010002                      MOV     R0,R2           ;
 9379 037150  010100                      MOV     R1,R0           ;
 9380 037152  010201                      MOV     R2,R1           ;SWAP THE TWO SETS OF
 9381 037154  010302                      MOV     R3,R2           ; PARAMETERS OVER.
 9382 037156  010403                      MOV     R4,R3           ;
 9383 037160  010204                      MOV     R2,R4           ;
 9384 037162  012602                      MOV     (SP)+,R2        ;RESTORE THE START ADDRESS.
 9385 037164  004737  025052              JSR     PC,SPLSUP       ;SET UP CONTROL BLOCK ETC, FOR TX/RX.
 9386
 9387                               ;+
 9388                               ; This routine reports errors with numbers >>>>> 9215 thru 9217 <<<<<.
```

```
 9389
 9390 037170  004737  023234                    ;-
                                                        JSR     PC,PUFIFR       ;PURGE THE DUT RECEIVE CHARACTER FIFO.
 9391 037174  103033                                    BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
 9392 037176  012737  022002  005320                    MOV     #9218.,ERRNBR   ;SET ERROR NUMBER TO 9218.
 9393
 9394 037204  004737  023426                            JSR     PC,PURRXB       ;PURGE THE RX CHAR BUFFER IN MEMORY.
 9395 037210  004737  021204                            JSR     PC,INIDMA       ;SEND THE FIRST BATCH OF DATA PATTERNS.
 9396                                            ;+
 9397                                            ; This routine reports errors with numbers >>>>> 9218 thru 9223 <<<<<.
 9398                                            ;-
 9399 037214  004737  023462                            JSR     PC,RDCHRS       ;READ AND VERIFY THE RX CHARACTERS.
 9400 037220  012737  022010  005320                    MOV     #9224.,ERRNBR   ;SET ERROR NUMBER TO 9224.
 9401                                            ;+
 9402                                            ; This routine reports errors with numbers >>>>> 9224 thru 9227 <<<<<.
 9403                                            ;-
 9404 037226  004737  026450                            JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM RX/RX.
 9405 037232  020527  005132                            CMP     R5,#SPLPRE      ;CHECK IF ALL PARAMETERS HAVE BEEN DONE.
 9406 037236  103701                                    BLO     4$              ;IF NOT DONE LOOP TO SELECT THE NEXT PARAMETER.
 9407                                            ;+
 9408                                            ;   Disable interrupts.
 9409                                            ;   Clear the interrupt vectors.
 9410                                            ;-
 9411 037240  106427  000240                            MTPS    #PRIO5          ;DISABLE DEVICE INTERRUPTS.
 9412 037244                                            CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
      037244  013700  002172                                                                            MOV     TXVECA,R0
      037250  104436                                                                                    TRAP    C$CVEC
 9413
 9414 037252  012737  022014  005320                    MOV     #9228.,ERRNBR   ;SELECT NUMBER 9228 FOR THE NEXT ERROR REPORT.
 9415 037260  004737  024220                            JSR     PC,REPSMR       ;REPORT ERROR SUMMARIES IF CALLED FOR.
 9416 037264  005037  002222          60$:              CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 9417 037270                                            ENDTST
      037270                                                                    L10035:
      037270  104401                                                                    TRAP    C$ETST
```

```
9419                                         .SBTTL  HARDWARE TEST            - REPBMP -
9420                                         ;** *********************************************************************
9421                                         ;*                           - Report any BMP codes in the queue -
9422                                         ;*       This is a pseudo-test used to report any BMP codes that were found
9423                                         ;*       in the DUT's FIFO during previous test, and logged in the BMP code
9424                                         ;*       queue.
9425                                         ;*       It is unlikely that running this pseudo-test alone will produce any
9426                                         ;*       error reports.
9427                                         ;*
9428                                         ;-- *********************************************************************
9429 037272                                  BGNTST
     037272                                                                                   T10::
9430          000012                          TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
9431 037272   012737  000012  002224          MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.       (93)
9432 037300   012737  177777  002222          MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
9433 037306   013702  002512                  MOV     BMPCQP,R2      ;GET THE CONTENTS OF THE POINTER.
9434 037312   012703  002514                  MOV     #BMPCQB,R3     ;GET THE START ADDRESS OF THE QUEUE.
9435 037316   020203                          CMP     R2,R3          ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
9436 037320   001411                          BEQ     60$            ;EXIT NO CODES IN THE QUEUE.
9437                                         ;+
9438                                         ; There is at least one BMP code in the queue. Report the error.
9439                                         ;-
9440                                             ;Report error BMP CODE FOUND IN TEST nn, BMP CODE:nnnnnn"
9441
9442 037322   012701  013001                  MOV     #EM9304,R1     ;PASS THE FIRST MESSAGE TO BE REORTED.
9443 037326                                   ERRDF   9301,EM9301,ER9301 ;            >>>>> ERROR #9301 <<<<<.
     037326   104455                                                                         TRAP    C$ERDF
     037330   022125                                                                         .WORD   9301
     037332   012664                                                                         .WORD   EM9301
     037334   015542                                                                         .WORD   ER9301
9444
9445 037336   012737  002514  002512          MOV     #BMPCQB,BMPCQP ;SET POINTER BACK TO THE BEGINING OF THE QUE.
9446
9447 037344   005037  002222          60$:    CLR     CTRLCF         ;INDICATE THAT WE ARE NOT WITHIN A TEST.
9448 037350                                   ENDTST
     037350                                                                             L10036:
     037350   104401                                                                         TRAP    C$ETST
```

```
9451                             ;¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤
9452                             ;
9453                             ;              VDHC.HWQ
9454                             ;
9455                             ;¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤¤
9457
9458
9459                             .SBTTL  HARDWARE PARAMETER CODING SECTION
9460
°961
9462
9463                             ;++
9464                             ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
9465                             ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
9466                             ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
9467                             ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
9468                             ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
9469                             ; WITH THE OPERATOR.
9470                             ;--
9471
9472 037352                          BGNHRD
     037352   000027                                                                        .WORD L10037-L$HARD/2
     037354                                                                         L$HARD::
9473
9483                             ;DEVICE CSR ADDRESS QUESTION:
9484 037354                          GPRMA   HWPTQ1,0,0,160000,177776,YES
     037354   000031                                                                        .WORD   T$CODE
     037356   037432                                                                        .WORD   HWPTQ1
     037360   160000                                                                        .WORD   T$LOLIM
     037362   177776                                                                        .WORD   T$HILIM
9485                             ;DEVICE INTERRUPT VECTOR QUESTION:
9486 037364                          GPRMA   HWPTQ2,2,0,40,776,YES
     037364   001031                                                                        .WORD   T$CODE
     037366   037450                                                                        .WORD   HWPTQ2
     037370   000040                                                                        .WORD   T$LOLIM
     037372   000776                                                                        .WORD   T$HILIM
9487                             ;ACTIVE LINES BIT MAP QUESTION:
9488 037374                          GPRMD   HWPTQ3,4,0,MAPLNS,0,MAPLNS,YES
     037374   002032                                                                        .WORD   T$CODE
     037376   037503                                                                        .WORD   HWPTQ3
     037400   000377                                                                        .WORD   MAPLNS
     037402   000000                                                                        .WORD   T$LOLIM
     037404   000377                                                                        .WORD   T$HILIM
9489                             ;TYPE OF LOOPBACK QUESTION:
9490 037406                          GPRMD   HWPTQ4,6,0,377,1,5,YES
     037406   003032                                                                        .WORD   T$CODE
     037410   037531                                                                        .WORD   HWPTQ4
     037412   000377                                                                        .WORD   377
     037414   000001                                                                        .WORD   T$LOLIM
     037416   000005                                                                        .WORD   T$HILIM
9491                             ;INTERRUPT BR LEVEL QUESTION:
9492 037420                          GPRMD   HWPTQ5,6,0,177400,0,6,YES
     037420   003032                                                                        .WORD   T$CODE
     037422   037670                                                                        .WORD   HWPTQ5
     037424   177400                                                                        .WORD   177400
     037426   000000                                                                        .WORD   T$LOLIM
     037430   000006                                                                        .WORD   T$HILIM
```

```
    9493
    9494
    9495 037432                                  ENDHRD

         037432                                                                    .EVEN
    9496                                                                   L10037:
    9503
    9504                                          .NLIST  BEX
    9505 037432     103     123     122   HWPTQ1: .ASCIZ  /CSR ADDRESS: /
    9506 037450     111     116     124   HWPTQ2: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
    9507 037503     101     103     124   HWPTQ3: .ASCIZ  /ACTIVE LINE BIT MAP: /
    9508 037531     124     131     120   HWPTQ4: .ASCII  /TYPE OF LOOPBACK (1=INTERNAL, 2=H3277, 3=H325/<15><12>
    9509 037610     040     040     040           .ASCIZ  /                   4=MODEM, 5=KEYBOARD ECHO): /
    9510 037670     111     116     124   HWPTQ5: .ASCIZ  /INTERRUPT BR LEVEL: /
    9511
    9512                                          .EVEN
```

```
 9515                           ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
 9516                           ;
 9517                           ;                    VDHC.SWQ
 9518                           ;
 9519                           ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

 9521
 9522
 9523                           .SBTTL   SOFTWARE PARAMETER CODING SECTION
 9524
 9525                           ;++
 9526                           ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
 9527                           ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
 9528                           ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
 9529                           ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
 9530                           ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
 9531                           ; WITH THE OPERATOR.
 9532                           ;--
 9533
 9534 037716                          BGNSFT
      037716  000013                                                       .WORD L10040-L$SOFT/2
      037720                                                       L$SOFT::
 9535
 9544                           ;UNIT NUMBER PRINTOUT QUESTION:
 9545 037720                          GPRML    SWPTQ1,0,20,YES
      037720  000130                                                       .WORD    T$CODE
      037722  037746                                                       .WORD    SWPTQ1
      037724  000020                                                       .WORD    20
 9546                           ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
 9547 037726                          GPRMD    SWPTQ2,2,D,177777,0,177777,YES
      037726  001052                                                       .WORD    T$CODE
      037730  040022                                                       .WORD    SWPTQ2
      037732  177777                                                       .WORD    177777
      037734  000000                                                       .WORD    T$LOLIM
      037736  177777                                                       .WORD    T$HILIM
 9548                           ;REPORT NUMB OF BITS TESTED IN DMA ADDR TEST QUESTION:
 9549 037740                          GPRML    SWPTQ3,0,40,YES
      037740  000130                                                       .WORD    T$CODE
      037742  040111                                                       .WORD    SWPTQ3
      037744  000040                                                       .WORD    40
 9550                                  .EVEN
 9551
 9552 037746                          ENDSFT
                                                                           .EVEN
      037746                                                       L10040:
 9553
 9554
 9561                                  .NLIST   BEX
 9562 037746     122    105    120     SWPTQ1:  .ASCIZ  /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /
 9563 040022     116    125    115     SWPTQ2:  .ASCIZ  /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /
 9564 040111     122    105    120     SWPTQ3:  .ASCIZ  /REPORT NUMBER OF BITS TESTED IN DMA ADDR TEST: /
 9565                                  .EVEN
```

```
9567
9568                              ;※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※
9569                              ;
9570                              ;              FVTSKL6.P11
9571                              ;
9572                              ;※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※※
9573
9574 040172                       $PATCH::
9575 040172                              .BLKW   24
9576
9583
9584 040242                              LASTAD
                                                                              .EVEN
     040242  000000                                                          .WORD   0
     040244  000000                                                          .WORD   0
     040246                       L$LAST::
9585 040246                              ENDMOD
9586          000001                     .END
```

Symbol table

Symbol	Value	G
ACTLNS	002174	G
ADDR	025366	
ADR =	000020	G
ADRPTR	017166	G
ALTFLD	015722	G
ASSEMB=	000010	
BCOUNT	002310	G
BDRMSG	013127	G
BITLNG	002260	G
BITSTD	034152	
BITTBL	002366	G
BIT0 =	000001	G
BIT00 =	000001	G
BIT01 =	000002	G
BIT02 =	000004	G
BIT03 =	000010	G
BIT04 =	000020	G
BIT05 =	000040	G
BIT06 =	000100	G
BIT07 =	000200	G
BIT08 =	000400	G
BIT09 =	001000	G
BIT1 =	000002	G
BIT10 =	0C2000	G
BIT11 =	004000	G
BIT12 =	010000	G
BIT13 =	020000	G
BIT14 =	040000	G
BIT15 =	100000	G
BIT2 =	000004	G
BIT3 =	000010	G
BIT4 =	000020	G
BIT5 =	000040	G
BIT6 =	000100	G
BIT7 =	000200	G
BIT8 =	000400	G
BIT9 =	001000	G
BMPCQB	002514	G
BMPCQE	002714	G
BMPCQP	002512	G
BOE =	000400	G
BRLEVL	002177	G
BRTBLB	002426	G
BRTBLE	002466	G
BUFBAS	003604	G
BUFEND	004604	G
BUFMID	004204	G
BUF3QT	004404	G
CALMSL	015774	G
CBB	003124	G
CBDPAA	003130	G
CBDPLA	003132	G
CBDPNA	003134	G
CBLNCA	003126	G
CBLPBA	003140	G
CBLPRA	003124	G
CBMAPA	003136	G
CBOFSA	003142	G
CHCNTB	003444	G
CHKEXT	016220	G
CHKLOS	016320	G
CHRMSK	016422	G
CHRTOT	002500	G
CKCHR	016460	G
CKFRPR	016676	G
CKINAC	017046	G
CKTRAP	017154	G
CKTRPB	017204	G
CLKBRL	002274	G
CLKCSR	002272	G
CLKHRZ	002300	G
CLKINT	027314	G
CLKVEC	002276	G
CLNRST	017234	G
CLR16W	017256	G
CONMAP	017300	G
CSRA	002202	G
CSRO =	000000	G
CTRLCF	002222	G
C$AU =	000052	
C$AUTO=	000061	
C$BRK =	000022	
C$BSEG=	000004	
C$BSUB=	000002	
C$CLCK=	000062	
C$CLEA=	000012	
C$CLOS=	000035	
C$CLP1=	000006	
C$CPBF=	000074	
C$CPME=	000075	
C$CVEC=	000036	
C$DCLN=	000044	
C$DODU=	000051	
C$DRPT=	000024	
C$DU =	000053	
C$EDIT=	000000	
C$ERDF=	000055	
C$ERHR=	000056	
C$ERRO=	000060	
C$ERSF=	000054	
C$ERSO=	000057	
C$ESCA=	000010	
C$ESEG=	000005	
C$ESUB=	000003	
C$ETST=	000001	
C$EXIT=	000032	
C$FREQ=	000101	
C$FRME=	000100	
C$GETB=	000026	
C$GETW=	000027	
C$GMAN=	000043	
C$GPHR=	000042	
C$GPRI=	000040	
C$INIT=	000011	
C$INLP=	000020	
C$MANI=	000050	
C$MAP =	000102	
C$MEM =	000031	
C$MMU =	000103	
C$MSG =	000023	
C$OPNR=	000034	
C$OPNW=	000104	
C$PNTB=	000014	
C$PNTF=	000017	
C$PNTS=	000016	
C$PNTX=	000015	
C$PUTB=	000072	
C$PUTW=	000073	
C$QIO =	000377	
C$RDBU=	000007	
C$REFG=	000047	
C$REL =	000077	
C$RESE=	000033	
C$REVI=	000004	
C$RFLA=	000021	
C$RPT =	000025	
C$SEFG=	000046	
C$SPRI=	000041	
C$SVEC=	000037	
C$TOME=	000076	
DELAY	017354	G
DEST	034156	
DFPTBL	002152	G
DIAGMC=	000000	
DLPRTB	005074	G
DLPRTE	005102	G
DMRW	017466	G
DMTSTA	002264	G
DM16B	017414	G
DODMA	017610	G
DPENDB	003144	G
DPLENB	003204	G
DPRSQB	004644	G
DPRSQE	005044	G
DRADRT	002202	G
DROP	031050	
DUMY	034154	
EDPFMT	007764	G
EDROP	031126	
EF.CON=	000036	G
EF.NEW=	000035	G
EF.PWR=	000034	G
EF.RES=	000037	G
EF.STA=	000040	G
EF0503	005461	G
EF1601	005466	G
EF1603	005512	G
EF4401	005554	G
EF6201	005671	G
EF6202	006004	G
EF6203	006102	G
EF7801	006177	G
EF8901	006235	G
EF9001	006266	G
EF9002	006350	G
EF9003	006422	G
EF9004	006451	G
EF9005	006501	G
EF9006	006532	G
EF9007	006551	G
EF9008	006645	G
EF9009	006704	G
EF9010	006743	G
EF9012	007032	G
EF9013	007146	G
EF9019	007213	G
EF9020	007232	G
EF9101	007313	G
EF9103	007316	G
EF9301	007364	G
EF9302	007432	G
EMLMSG	013156	G
EM0101	022430	G
EM0102	022514	G
EM0103	010072	G
EM0509	010130	G
EM1601	010134	G
EM4401	010217	G
EM4402	010241	G
EM4403	010307	G
EM4404	010404	G
EM4405	010443	G
EM4406	010537	G
EM4407	010613	G
EM4408	010675	G
EM4409	010740	G
EM4410	010775	G
EM4411	011024	G
EM5303	011071	G
EM6201	011142	G
EM6202	011166	G
EM6301	011175	G
EM8901	011220	G
EM9001	011245	G
EM9003	011301	G
EM9004	011323	G
EM9006	011341	G
EM9007	011414	G
EM9008	011477	G
EM9009	011560	G
EM9010	011604	G
EM9011	011630	G
EM9012	011640	G
EM9013	011650	G
EM9014	011657	G
EM9015	011753	G
EM9016	011767	G
EM9017	011776	G
EM9025	012107	G
EM9026	012203	G
EM9027	012227	G
EM9028	012307	G
EM9030	012366	G
EM9101	012443	G
EM9102	012477	G
EM9104	012566	G
EM9201	012642	G
EM9301	012664	G
EM9302	012704	G
EM9303	012734	G
EM9304	013001	G
EM9401	013055	G
ENDETB	004604	G
ENDIT	030750	
ERCNTB	003304	G
ERLTBL	003604	G
ERRBLK	005324	G
ERRMSG	005322	G
ERRNBR	005320	G
ERRTYP	005316	G
ERSMRF	002502	G
ER0101	013562	G
ER0503	014100	G
ER1603	014124	G
ER6201	014204	G
ER9001	014432	G
ER9002	014522	G
ER9003	014670	G
ER9004	015052	G
ER9005	015154	G
ER9101	015402	G
ER9102	015430	G
ER9301	015542	G
EVL =	000004	G
EXCNTB	003244	G
EXTMSG	013225	G
E$END =	002100	
E$LOAD=	000035	
FFREM	002262	G
FINACT	017756	G
FRPSUP	020036	G
F$AU =	000015	
F$AUTO=	000020	
F$BGN =	000040	
F$CLEA=	000007	
F$DU =	000016	
F$END =	000041	
F$HARD=	000004	
F$HW =	000013	
F$INIT=	000006	
F$JMP =	000050	
F$MOD =	000000	
F$MSG =	000011	
F$PROT=	000021	
F$PWR =	000017	

Symbol table

F$RPT = 000012	I$MSG = 000041	L$LAST 040246 G	MMSR3 002320 G	PREGRT 005350 G
F$SEG = 000003	I$PROT= 000040	L$LOAD 002100 G	MODSUP 021324 G	PREG05 005326
F$SOFT= 000005	I$PTAB= 000041	L$LUN 002074 G	MSFMT1 007630 G	PRFRME 022612 G
F$SRV = 000010	I$PWR = 000041	L$MREV 002050 G	MSFMT2 007670 G	PRI = 002000 G
F$SUB = 000002	I$RPT = 000041	L$NAME 002000 G	MSG1 013666 G	PRI00 = 000000 G
F$SW = 000014	I$SEG = 000041	L$PRIO 002042 G	MSG2 013744 G	PRI01 = 000040 G
F$TEST= 000001	I$SETU= 000041	L$PROT 030122 G	MSG3 014023 G	PRI02 = 000100 G
GETBDR 020302 G	I$SFT = 000041	L$PRT 002112 G	MSLCNT 002314 G	PRI03 = 000140 G
GETCHR 020430 G	I$SRV = 000041	L$REPP 002062 G	MSLGET 021464 G	PRI04 = 000200 G
GETLP1 020512 G	I$SUB = 000041	L$REV 002010 G	MSLOOP 021600 G	PRI05 = 000240 G
GETLP2 020664 G	I$TST = 000041	L$RPT 030114 G	MSSRPT 021614 G	PRI06 = 000300 G
GETPRM 030530	J$JMP = 000167	L$SOFT 037720 G	MSTICK 002312 G	PRI07 = 000340 G
GETTIM 021014 G	LGRP1M 002230 G	L$SPC 002056 G	MUL16U 022030 G	PROTBL 005216 G
GMANWD 002266 G	LGRP2M 002232 G	L$SPCP 002020 G	NDERPT 002166 G	PRPARE 022710 G
GPRSOB 002466 G	LINBIT 021276 G	L$SPTP 002024 G	NDPMSG 013306 G	PRTLPR 023070 G
G$CNTO= 000200	LNCTRA 002212 G	L$STA 002030 G	NEWCHR 022104 G	PUFIFO 023152 G
G$DELM= 000372	LNCTRO= 000010 G	L$SW 002164 G	NEWPAS 030510	PUFIFR 023234 G
G$DISP= 000003	LOE = 040000 G	L$TEST 002114 G	NEWRES 030502	PURRXB 023426 G
G$EXCP= 000400	LOPBCK 002176 G	L$TIML 002014 G	NEWSTA 030174	RBUFA 002204 G
G$HILI= 000002	LOT = 000010 G	L$UNIT 002012 G	NUMLNS= 000010 G	RBUFO = 000002 G
G$LOLI= 000001	LPCSLT= 000036 G	L10000 002162	ODTSTA 034160	RDCHRS 023462 G
G$NO = 000000	LPRA 002206 G	L10001 002170	OOPS 022364 G	RDMAST 024034 G
G$OFFS= 000400	LPRO = 000004 G	L10002 013664	OPTION 002164 G	REPCOD 024066 G
G$OFSI= 0C0376	L$ACP 002110 G	L10003 014122	O$APTS= 000000	REPSMR 024220 G
G$PRMA= 000001	L$APT 002036 G	L10004 014202	O$AU = 000000	RESETT 024246 G
G$PRMD= 000002	L$AU 031134 G	L10005 014430	O$BGNR= 000001	RRXNDN 024360 G
G$PRML= 000000	L$AUT 002070 G	L10006 014520	O$BGNS= 000001	RTXNDN 024420 G
G$RADA= 000140	L$AUTO 030772 G	L10007 014666	O$DU = 000000	RXBCNT 002720 G
G$RADB= 000000	L$CCP 002106 G	L10010 015050	O$ERRT= 000001	RXBDTX= 000030 G
G$RADD= 000040	L$CLEA 030774 G	L10011 015152	O$GNSW= 000001	RXBEND 003122 G
G$RADL= 000120	L$CO 002032 G	L10012 015400	O$POIN= 000001	RXBETX= 000020 G
G$RADO= 000020	L$DEPO 002011 G	L10013 015426	O$SETU= 000000	RXBFUL= 000100 G
G$XFER= 000004	L$DESC 005376 G	L10014 015540	PARATB 002326 G	RXBIPT 002716 G
G$YES = 000010	L$DESP 002076 G	L10015 015720	PARATE 002346 G	RXBOPT 002714 G
HELP = 000000	L$DEVP 002060 G	L10016 030120	PAROA 002326 G	RXBSTA 002722 G
HOE = 100000 G	L$DISP 002124 G	L10020 030770	PAR1A 002330 G	RXCHRS 027364 G
HWPTQ1 037432	L$DLY 002116 G	L10021 030772	PAR2A 002332 G	RXCNTB 003544 G
HWPTQ2 037450	L$DTP 002040 G	L10022 031022	PAR3A 002334 G	RXDONF 002506 G
HWPTQ3 037503	L$DTYP 002034 G	L10023 031132	PAR4A 002336 G	RXDSBL 024460 G
HWPTQ4 037531	L$DU 031024 G	L10024 031140	PAR5A 002340 G	RXENBL 024554 G
HWPTQ5 037670	L$DUT 002072 G	L10025 031430	PAR6A 002342 G	RXIEO 024650 G
IBE = 010000 G	L$DVTY 005366 G	L10026 032002	PAR7A 002344 G	RXIE1 024702 G
IBM 002226 G	L$EF 002052 G	L10027 032366	PASCNT 002236 G	RXPTRB 003404 G
IDU = 000040 G	L$ENVI 002044 G	L10030 034164	PCSLOT= 000016 G	RXTOUT 002242 G
IER = 020000 G	L$ERRT 005316 G	L10031 035056	PDRATB 002346 G	RXVECA 002170 G
IESTAT 002234 G	L$ETP 002102 G	L10032 035304	PDRATE 002366 G	ROSLOT= 000002 G
INICHR 021072 G	L$EXP1 002046 G	L10033 036034	PDROA 002346 G	R1SLOT= 000004 G
INIDMA 021204 G	L$EXP4 002064 G	L10034 036644	PDR1A 002350 G	R2SLOT= 000006 G
ISR = 000100 G	L$EXP5 002066 G	L10035 037270	PDR2A 002352 G	R3SLOT= 000010 G
IXE = 004000 G	L$HARD 037354 G	L10036 037350	PDR3A 002354 G	R4SLOT= 000012 G
I$AU = 000041	L$HIME 002120 G	L10037 037432	PDR4A 002356 G	R5SLOT= 000014 G
I$AUTO= 000041	L$HPCP 002016 G	L10040 037746	PDR5A 002360 G	SAVBMP 024726 G
I$CLN = 000041	L$HPTP 002022 G	MAPLNS= 000377 G	PDR6A 002362 G	SAVPRI 002246 G
I$DU = 000041	L$HW 002152 G	MFUNIT 005430 G	PDR7A 002364 G	SAVTEN 002244 G
I$HRD = 000041	L$ICP 002104 G	MMENAB 002324 G	PMSFLG 002270 G	SCBCTB 005044 G
I$INIT= 000041	L$INIT 030130 G	MMPRES 002322 G	PMSMSG 013434 G	SCBCTE 005054 G
I$MOD = 000041	L$LADP 002026 G	MMSRO 002316 G	PNT = 001000 G	SCBRTB 005054 G

Symbol table

SCBRTE	005062 G	SWPTQ3	040111	TXDSBL	025552 G	T$LTNO=	000012	T$$TES=	010036
SCNSTB	005062 G	S$LSYM=	010000	TXENBL	025646 G	T$NEST=	177777	T1	031142 G
SCNSTE	005066 G	TERMSG	013521 G	TXENBM	002250 G	T$NSO =	000000	T10	037272 G
SCTPTB	005066 G	TIMER1	002302 G	TXFRPR	025742 G	T$NS1 =	000005	T2	031432 G
SCTPTE	005074 G	TIMER2	002304 G	TXIEO	026036 G	T$PTNU=	000000	T3	032004 G
SDPBAS	005132 G	TIMER3	002306 G	TXIE1	026070 G	T$SAVL=	177777	T4	032370 G
SDPEND	005152 G	TNUM =	000012	TXINTF	002252 G	T$SEGL=	177777	T5	034166 G
SDP2B	005156 G	TP4BRT	027530 G	TXPTRB	003344 G	T$SUBN=	000000	T6	035060 G
SDP2E	005176 G	TP4FLG	002256 G	TXRINI	026114 G	T$TAGL=	177777	T7	035306 G
SFPTBL	002164 G	TP4RTN	027552 G	TXROFF	026370 G	T$TAGN=	010041	T8	036036 G
SKPSTS	024774 G	TP4VEC	002254 G	TXRON	026424 G	T$TEMP=	000000	T9	036646 G
SPLPRB	005102 G	TRPAD2	017216 G	TXRREP	026450 G	T$TEST=	000012	UAM =	000200 G
SPLPRE	005132 G	TSTNUM	002224 G	TXRXLB	005236 G	T$TSTM=	177777	UBRFMT	007532 G
SPLSUP	025052 G	TXAD1A	002214 G	TXRXLE	005276 G	T$TSTS=	000001	UNITN	002200 G
STATA	002210 G	TXAD1O=	000012 G	TXSCHR	027726 G	T$$AU =	010024	UNSDIV	026530 G
STATO =	000006 G	TXAD2A	002216 G	TXVECA	002172 G	T$$AUT=	010021	UPDCHR	026664 G
STGTRB	005276 G	TXAD2O=	000014 G	T$ARGC=	000002	T$$CLE=	010022	VANSUP	026762 G
STPSW	025352 G	TXBFCA	002220 G	T$CODE=	000130	T$$DU =	010023	WAIBIS	027160 G
SUCSS	034162	TXBFCO=	000016 G	T$ERRN=	022125	T$$HAR=	010037	WORD1	002240 G
SVCGBL=	000000	TXCHA	002204 G	T$EXCP=	000000	T$$HW =	010000	WTWLNC	027234 G
SVCINS=	000001	TXCHRO=	000002 G	T$FLAG=	000040	T$$INI=	010020	WTWLPR	027264 G
SVCSUB=	000001	TXCNTB	003504 G	T$GMAN=	000000	T$$MSG=	010015	X$ALWA=	000000
SVCTAG=	000001	TXDBLF	002510 G	T$HILI=	177777	T$$PRO=	010017	X$FALS=	000040
SVCTST=	0C0001	TXDMA	027574 G	T$LAST=	000001	T$$RPT=	010016	X$OFFS=	000400
SWAPO	025372 G	TXDONE	025442 G	T$LOLI=	000000	T$$SOF=	010040	X$TRUE=	000020
SWPTQ1	037746	TXDONF	002504 G	T$LSYM=	010000	T$$SW =	010001	$PATCH	040172 G
SWPTQ2	040022								

```
. ABS.  040246    000   (RW,I,GBL,ABS,OVR)
        000000    001   (RW,I,LCL,REL,CON)
Errors detected:  0

*** Assembler statistics


Work  file  reads: 436
Work  file  writes: 393
Size of work file: 35680 Words  ( 140 Pages)
Size of core pool: 17728 Words  ( 68 Pages)
Operating  system: RSX-11M/PLUS

Elapsed time: 00:06:29.23
CVDHCDO.OBJ,CVDHCDO.LST/CRF/-SP=SVC40/ML,CVDHCDO.P11
```

SYMBOL CROSS REFERENCE   CREF   03.00

```
SYMBOL   VALUE         REFERENCES

ACTLNS   002174  G     #33-1396    62-3246    63-3339    73-3787    74-3858    74-3876    74-3893    74-3905    74-3923
                       80-4234     81-4287    83-4409    83-4422    86-4599    96-5232    109-5938   109-5957   109-5976
                       109-5988    109-6005   112-6123   112-6150   115-6301   121-6633   124-6854   124-6872   124-6888
                       *136-7566   *136-7569  145-8771   146-8940
ADDR     025366        110-6033    #110-6035
ADR    = 000020  G     #32-1371
ADRPTR   017166  G     #64-3390    131-7178
ALTFLD   015722  G     #56-2748    126-6974   127-7013
ASSEMB = 000010        28-1183     28-1183
BCOUNT   002310  G     #33-1449    128-7049   *136-7492
BDRMSG   013127  G     #41-2160    75-3959
BITLNG   002260  G     #33-1432    *144-8253  *144-8296
BITSTD   034152        *144-8192   *144-8304  *144-8308  *144-8347  *144-8374  144-8602   #144-8635
BITTBL   002366  G     #33-1486    61-3106    63-3339    80-4234    81-4298    82-4359    88-4762    88-4778    88-4800
                       90-4884     91-4938    91-4952    91-4954    115-6331   118-6475   123-6798   125-6931   132-7230
                       133-7342
BIT0   = 000001  G     #32-1371    43-2214    46-2328    71-3663    94-5109    98-5396    103-5662   104-5715   113-6185
                       114-6239    144-8253   144-8266
BIT00  = 000001  G     #32-1371    32-1371
BIT01  = 000002  G     #32-1371    32-1371
BIT02  = 000004  G     #32-1371    32-1371
BIT03  = 000010  G     #32-1371    32-1371
BIT04  = 000020  G     #32-1371    32-1371
BIT05  = 000040  G     #32-1371    32-1371    100-5493   108-5858   144-8611
BIT06  = 000100  G     #32-1371    32-1371    106-5790
BIT07  = 000200  G     #32-1371    32-1371
BIT08  = 000400  G     #32-1371    32-1371
BIT09  = 001000  G     #32-1371    32-1371
BIT1   = 000002  G     #32-1371    43-2217    46-2326    90-4877
BIT10  = 002000  G     #32-1371
BIT11  = 004000  G     #32-1371
BIT12  = 010000  G     #32-1371    49-2464    90-4879    91-4930
BIT13  = 020000  G     #32-1371    49-2457    90-4875    91-4933
BIT14  = 040000  G     #32-1371    49-2450    117-6393
BIT15  = 100000  G     #32-1371    61-3117    63-3351    80-4244    91-4951    91-4956    94-5116    129-7113   130-7151
                       131-7181    133-7365
BIT2   = 000004  G     #32-1371    46-2341    103-5671   103-5680   104-5724   104-5733   124-6855
BIT3   = 000010  G     #32-1371    46-2333    91-4932
BIT4   = 000020  G     #32-1371    136-7600
BIT5   = 000040  G     #32-1371
BIT6   = 000100  G     #32-1371    136-7505   136-7617
BIT7   = 000200  G     #32-1371    113-6204   114-6258
BIT8   = 000400  G     #32-1371
BIT9   = 001000  G     #32-1371
BMPCQB   002514  G     #33-1545    54-2675    136-7590   150-9434   150-9445
BMPCQE   002714  G     #33-1546    54-2688    107-5824   136-7591
BMPCQP   002512  G     #33-1544    107-5819   *107-5827  *136-7592  150-9433   *150-9445
BOE    = 000400  G     #32-1371
BRLEVL   002177  G     #33-1398    *136-7571
BRTBLB   002426  G     #33-1506
BRTBLE   002466  G     #33-1523    75-3965
BUFBAS   003604  G     #33-1583    *142-7931  142-7940   144-8324   144-8339   144-8572   145-8753   147-9050   147-9067
```

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL   VALUE   REFERENCES

```
                           147-9068    148-9194    148-9221
BUFEND   004604   G    #33-1587
BUFMID   004204   G    #33-1585    144-8334    144-8580    147-9054    147-9068
BUF3QT   004404   G    #33-1586
CALMSL   015774   G    #57-2808    136-7516
CBB      003124   G    #33-1559    *74-3852    74-3853     74-3873     *83-4401    83-4402     *109-5932   109-5933    109-5951
                      *124-6846    124-6847
CBDPAA   003130   G    #33-1562    118-6472
CBDPLA   003132   G    #33-1563    118-6473    118-6482    118-6495    118-6524
CBDPNA   003134   G    #33-1564    *74-3899    *83-4428    *109-5982   118-6483    *124-6877
CBLNCA   003126   G    #33-1561    *83-4427    118-6460
CBLPBA   003140   G    #33-1566    118-6461    118-6506
CBLPRA   003124   G    #33-1560    *74-3900    *74-3911    *109-5983   *109-5993   118-6465
CBMAPA   003136   G    #33-1565    *74-3898    *74-3910    *83-4426    *109-5981   *109-5992   118-6459    118-6475    *124-6876
CBOFSA   003142   G    #33-1567    118-6521
CHCNTB   003444   G    #33-1577    51-2570     58-2932     59-2992     112-6142    *118-6485   123-6795    132-7238    133-7340
CHKEXT   016220   G    #58-2919    61-3138
CHKLOS   016320   G    #59-2988    61-3157
CHRMSK   016422   G    #60-3053    147-9022    147-9071
CHRTOT   002500   G    #33-1536    *62-3254    *83-4394    *96-5261    *118-6486   *118-6488   *118-6490   *124-6839
CKCHR    016460   G    #61-3102    88-4771
CKFRPR   016676   G    #62-3238    142-7960    142-7978    143-8086    143-8111
CKINAC   017046   G    #63-3320    88-4732
CKTRAP   017154   G    #64-3387    136-7508    136-7529    141-7815    141-7822    141-7833    141-7838    144-8267
CKTRPB   017204   G    #65-3423    71-3668
CLKBRL   002274   G    #33-1443    *136-7481
CLKCSR   002272   G    #33-1442    *136-7480   136-7507    136-7617    138-7687
CLKHRZ   002300   G    #33-1445    *57-2828    *136-7483   136-7484    136-7490    *136-7511   136-7615    138-7685
CLKINT   027314   G    #128-7041   136-7489
CLKVEC   002276   G    #33-1444    *136-7482   136-7489
CLNRST   017234   G    #66-3462    142-7909    143-8035    144-8179    145-8709    146-8913    147-8987    148-9130    149-9314
CLR16W   017256   G    #67-3495    142-7930    143-8054    145-8724    147-9004    148-9146    149-9334
CONMAP   017300   G    #68-3523    74-3924     109-6006    121-6638    124-6889
CSRA     002202   G    #33-1405    56-2768     72-3749     80-4236     86-4597     92-4992     100-5499    100-5512    103-5670
                       104-5723    105-5766    106-5792    108-5863    113-6194    114-6248    116-6369    117-6395    132-7214
                       132-7224    132-7272    133-7311    133-7325    133-7357    133-7374    *136-7561   136-7576    141-7813
                       141-7821    141-7830    144-8470    144-8488
CSRO   = 000000   G    #32-1341
CTRLCF   002222   G    #33-1417    138-7681    *141-7795   *141-7865   *141-7873   *142-7897   *142-7997   *143-8023   *143-8134
                      *144-8155   *144-8624   *145-8699   *145-8870   *146-8903   *146-8956   *147-8977   *147-9100   *148-9120
                      *148-9285   *149-9304   *149-9416   *150-9432   *150-9447
C#AU   = 000052        #28-1183    140-7775
C#AUTO = 000061        #28-1183    137-7652
C#BRK  = 000022        #28-1183    89-4831     128-7051
C#BSEG = 000004        #28-1183
C#BSUB = 000002        #28-1183
C#CLCK = 000062        #28-1183    136-7479
C#CLEA = 000012        #28-1183    138-7705
C#CLOS = 000035        #28-1183
C#CLP1 = 000006        #28-1183
C#CPBF = 000074        #28-1183
C#CPME = 000075        #28-1183
```

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL	VALUE	REFERENCES							
C$CVEC	= 000036	@28-1183	142-7990	143-8125	145-8864	145-8865	147-9098	147-9099	148-9279	148-9280
		149-9412							
C$DCLN	= 000044	@28-1183	141-7866						
C$DODU	= 000051	@28-1183	141-7864						
C$DRPT	= 000024	@28-1183							
C$DU	= 000053	@28-1183	139-7742						
C$EDIT	= 000000	@28-1183	28-1231						
C$ERDF	= 000055	@28-1183	141-7862	150-9443					
C$ERHR	= 000056	@28-1183							
C$ERRO	= 000060	@28-1183	62-3263	88-4766	88-4783	90-4888	91-4941	91-4960	94-5120	94-5134
		96-5270	97-5359	98-5415	99-5462	100-5531	101-5585	102-5631	144-8619
C$ERSF	= 000054	@28-1183	89-4828						
C$ERSO	= 000057	@28-1183							
C$ESCA	= 000010	@28-1183							
C$ESEG	= 000005	@28-1183							
C$ESUB	= 000003	@28-1183							
C$ETST	= 000001	@28-1183	141-7874	142-7999	143-8135	144-8650	145-8871	146-8957	147-9102	148-9286
		149-9417	150-9448						
C$EXIT	= 000032	@28-1183	138-7689	144-8626	145-8694	145-8697	146-8898	146-8901	
C$FREQ	= 000101	@28-1183							
C$FRME	= 000100	@28-1183							
C$GETB	= 000026	@28-1183							
C$GETW	= 000027	@28-1183							
C$GHAN	= 000043	@28-1183	75-3959	145-8778	145-8791	145-8799	145-8848	146-8951	
C$GPHR	= 000042	@28-1183	136-7555						
C$GPRI	= 000040	@28-1183							
C$INIT	= 000011	@28-1183	136-7620						
C$INLP	= 000020	@28-1183							
C$MANI	= 000050	@28-1183	145-8695	146-8899					
C$MAP	= 000102	@28-1183							
C$MEM	= 000031	@28-1183	144-8321						
C$MMU	= 000103	@28-1183							
C$MSG	= 000023	@28-1183	43-2222	44-2253	45-2285	46-2351	47-2379	48-2415	49-2476	50-2514
		51-2581	52-2606	53-2649	54-2703				
C$OPNR	= 000034	@28-1183							
C$OPNW	= 000104	@28-1183							
C$PNTB	= 000014	@28-1183	43-2216	43-2219	44-2251	45-2279	45-2282	46-2320	47-2375	48-2407
		49-2442	50-2501	51-2542	52-2604	53-2638	53-2643	53-2647	54-2674	144-8615
C$PNTF	= 000017	@28-1183	75-3978	86-4593	86-4626	86-4634	89-4830	136-7604	139-7731	145-8728
		146-8919							
C$PNTS	= 000016	@28-1183							
C$PNTX	= 000015	@28-1183	43-2220	46-2332	46-2345	46-2347	47-2376	47-2377	48-2408	48-2411
		48-2413	49-2445	49-2472	50-2508	51-2550	51-2554	51-2573	54-2695	54-2699
		92-4999	145-8831						
C$PUTB	= 000072	@28-1183							
C$PUTW	= 000073	@28-1183							
C$QIO	= 000377	@28-1183							
C$RDBU	= 000007	@28-1183							
C$REFG	= 000047	@28-1183	136-7462	136-7465	136-7468	136-7471			
C$REL	= 000077	@28-1183							
C$RESE	= 000033	@28-1183	@28-1183	136-7475	136-7536	138-7683			
C$REVI	= 000004	@28-1183	28-1231						

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL	VALUE		REFERENCES								
C$RFLA	= 000021		#28-1183								
C$RPT	= 000025		#28-1183	134-7400							
C$SEFG	= 000046		#28-1183								
C$SPRI	= 000041		#28-1183								
C$SVEC	= 000037		#28-1183	136-7489	142-7916	143-8042	145-8717	145-8718	147-8996	147-8997	148-9137
			148-9138	149-9321	149-9322						
C$TOME	= 000076		#28-1183								
DELAY	017354	G	#69-3559	108-5854	144-8500						
DEST	034156		144-8258	144-8281	#144-8637						
DFPTBL	002152	G	#30-1277								
DIAGMC	= 000000		28-1183	28-1183							
DLPRTB	005074	G	#33-1638	148-9209							
DLPRTE	005102	G	#33-1642	148-9271							
DMRW	017466	G	#71-3648	144-8333	144-8408	144-8413	144-8575	144-8581			
DMTSTA	002264	G	#33-1435	70-3590	71-3656	71-3660	*144-8190	*144-8302	*144-8306	144-8343	*144-8345
			144-8353	144-8367	*144-8370	144-8393	*144-8394	*144-8411	144-8550	144-8570	*144-8571
			*144-8578								
DM16B	017414	G	#70-3589	71-3653	144-8356						
DODMA	017610	G	#72-3712	81-4311	115-6326	132-7252					
DPENDB	003144	G	#33-1571	80-4239	*118-6496	123-6779	133-7360				
DPLENB	003204	G	#33-1572	80-4241	81-4306	115-6321	*118-6495	123-6781	132-7247	133-7362	
DPRSQB	004644	G	#33-1592	88-4740	96-5222						
DPRSQE	005044	G	#33-1609	96-5223							
DRADRT	002202	G	#33-1404								
DROP	031050		139-7731	#139-7735							
DUMY	034154		*144-8393	144-8411	*144-8570	144-8578	#144-8636				
EDPFMT	007764	G	#41-2106	145-8831							
EDROP	031126		139-7732	#139-7737							
EF.CON	= 000036	G	#32-1371	136-7471							
EF.NEW	= 000035	G	#32-1371	136-7468							
EF.PWR	= 000034	G	#32-1371								
EF.RES	= 000037	G	#32-1371	136-7465							
EF.STA	= 000040	G	#32-1371	136-7462							
EF0503	005461	G	#41-2074	44-2251	45-2279	50-2501	53-2638	54-2674			
EF1601	005466	G	#41-2075	45-2282							
EF1603	005512	G	#41-2076	46-2347							
EF4401	005554	G	#41-2077	144-8615							
EF6201	005671	G	#41-2079	46-2320							
EF6202	006004	G	#41-2080	46-2332							
EF6203	006102	G	#41-2081	46-2345							
EF7801	006177	G	#41-2082								
EF8901	006235	G	#41-2083	145-8728	146-8919						
EF9001	006266	G	#41-2084	47-2375							
EF9002	006350	G	#41-2085	47-2376							
EF9003	006422	G	#41-2086	47-2377							
EF9004	006451	G	#41-2087	48-2408	48-2411						
EF9005	006501	G	#41-2088	48-2413							
EF9006	006532	G	#41-2089	48-2407	52-2604						
EF9007	006551	G	#41-2090	49-2442							
EF9008	006645	G	#41-2091	49-2445							
EF9009	006704	G	#41-2092	49-2472							
EF9010	006743	G	#41-2093	50-2508							

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL	VALUE		REFERENCES				
EF9012	007032	G	41-2094	51-2573			
EF9013	007146	G	41-2096	51-2542			
EF9019	007213	G	41-2097	92-4999			
EF9020	007232	G	41-2098	51-2554			
EF9101	007313	G	41-2099	53-2647	86-4634		
EF9103	007316	G	41-2100	53-2643			
EF9301	007364	G	41-2101	54-2699			
EF9302	007432	G	41-2102	54-2695			
EMLMSG	013156	G	41-2161	145-8778			
EM0101	022430	G	89-4828	89-4837			
EM0102	022514	G	89-4830	89-4838			
EM0103	010072	G	41-2110	141-7862			
EM0509	010130	G	41-2111	46-2325	46-2331	46-2336	46-2344
EM1601	010134	G	41-2112	100-5527			
EM4401	010217	G	41-2113	144-8158			
EM4402	010241	G	41-2114	144-8379			
EM4403	010307	G	41-2115	144-8404			
EM4404	010404	G	41-2116	144-8433			
EM4405	010443	G	41-2117	144-8473			
EM4406	010537	G	41-2118	144-8493			
EM4407	010613	G	41-2119	144-8511			
EM4408	010675	G	41-2120	144-8524			
EM4409	010740	G	41-2121	144-8536			
EM4410	010775	G	41-2122	144-8553			
EM4411	011024	G	41-2123	144-8587			
EM5303	011071	G	41-2124				
EM6201	011142	G	41-2125	142-7902			
EM6202	011166	G	41-2126	46-2324	46-2335		
EM6301	011175	G	41-2127	143-8028			
EM8901	011220	G	41-2128	145-8702	145-8728		
EM9001	011245	G	41-2129	147-8980			
EM9003	011301	G	41-2130	98-5395			
EM9004	011323	G	41-2131	98-5399			
EM9006	011341	G	41-2132	63-3350			
EM9007	011414	G	41-2133	61-3114			
EM9008	011477	G	41-2134	61-3179	91-4957		
EM9009	011560	G	41-2135	48-2411	48-2413		
EM9010	011604	G	41-2136	48-2408			
EM9011	011630	G	41-2137	49-2449			
EM9012	011640	G	41-2138	49-2456			
EM9013	011650	G	41-2139	49-2463			
EM9014	011657	G	41-2140	50-2501			
EM9015	011753	G	41-2141	51-2544	102-5623		
EM9016	011767	G	41-2142	101-5578			
EM9017	011776	G	41-2143	94-5131			
EM9025	012107	G	41-2145	62-3257	96-5264		
EM9026	012203	G	41-2146	92-4999			
EM9027	012227	G	41-2147	61-3145			
EM9028	012307	G	41-2148	61-3165			
EM9030	012366	G	41-2149	51-2550			
EM9101	012443	G	41-2150	148-9123			
EM9102	012477	G	41-2151	97-5355			

SYMBOL CROSS REFERENCE   CREF   03.00

```
SYMBOL    VALUE          REFERENCES

EM9104    012566  G    #41-2152     94-5111
EM9201    012642  G    #41-2153    149-9307
EM9301    012664  G    #41-2154    150-9443
EM9302    012704  G    #41-2155     54-2676
EM9303    012734  G    #41-2156     54-2694
EM9304    013001  G    #41-2157    150-9442
EM9401    013055  G    #41-2158    146-8906    146-8919
ENDETB    004604  G    #33-1588
ENDIT     030750         136-7473  #136-7607
ERCNTB    003304  G    #33-1574     50-2508   *88-4792   *88-4794    88-4797   *90-4889   *91-4962    142-7929    143-8053
                        145-8723    147-9003   148-9145   149-9333
ERLTBL    003604  G    #33-1584
ERRBLK    005324  G    #33-1803    *62-3258   *88-4764   *88-4780   *90-4874   *91-4929   *91-4958   *94-5100    *94-5118
                       *94-5130    *96-5265   *97-5354   *98-5387   *99-5457  *100-5528  *101-5577   *102-5622   *144-8159
                      *144-8538   *144-8555  *144-8591
ERRMSG    005322  G    #33-1803     45-2281   *142-7902  *143-8028  *144-8158  *145-8702  *146-8906   *147-8980   *148-9123
                      *149-9307
ERRNBR    005320  G    #33-1803     62-3239   *62-3256   *62-3287   *88-4781   *88-4785    90-4867   *90-4890     91-4922
                       *91-4946    *91-4963   94-5082   *94-5117   *94-5121   *94-5129   *94-5140     96-5213    *96-5263
                       *96-5283    *96-5284  *96-5293   *96-5294   *96-5323    98-5388   *98-5398    *98-5420     121-6632
                      *121-6635   *121-6637 *121-6640  *142-7901  *142-7951  *142-7954  *142-7966   *142-7972   *142-7979
                      *142-7991   *143-8027 *143-8080  *143-8087  *143-8092  *143-8106  *143-8112   *143-8130   *144-8157
                      *144-8378   *144-8403 *144-8427  *144-8472  *144-8490  *144-8510  *144-8523   *144-8534   *144-8552
                      *144-8586   *145-8701 *145-8816  *145-8821  *145-8867  *146-8905  *147-8979   *147-8990   *147-9026
                      *147-9032   *147-9049 *147-9075  *147-9080  *147-9089  *148-9122  *148-9154   *148-9163   *148-9168
                      *148-9226   *148-9232 *148-9240  *148-9245  *148-9258  *148-9266  *148-9282   *149-9306   *149-9354
                      *149-9368   *149-9373 *149-9392  *149-9400  *149-9414
ERRTYP    005316  G    #33-1803   *142-7900  *143-8026  *144-8156  *145-8700  *146-8904  *147-8978   *148-9121   *149-9305
ERSMRF    002502  G    #33-1537    50-2503    88-4762    88-4778   *88-4800    90-4884    91-4938     91-4952     99-5452
                      *142-7903   *143-8029 *145-8703  *146-8907  *147-8981  *148-9124  *149-9308
ER0101    013562  G    #43-2211    141-7862
ER0503    014100  G    #44-2249    62-3258    96-5265    144-8159   144-8555   144-8591
ER1603    014124  G    #45-2276    94-5130    100-5528
ER6201    014204  G    #46-2314    90-4874    91-4929
ER9001    014432  G    #47-2373    94-5100    98-5387
ER9002    014522  G    #48-2403    88-4780    91-4958    94-5118
ER9003    014670  G    #49-2439    88-4764
ER9004    015052  G    #50-2499    99-5457
ER9005    015154  G    #51-2539    101-5577   102-5622
ER9101    015402  G    #52-2602    144-8538
ER9102    015430  G    #53-2635    97-5354
ER9301    015542  G    #54-2671    150-9443
EVL     = 000004  G    #32-1371
EXCNTB    003244  G    #33-1573   *61-3188   *61-3190   *63-3347   *63-3349   *118-6516
EXTMSG    013225  G    #41-2162    145-8848
E$END   = 002100       #28-1183
E$LOAD  = 000035       #28-1183    28-1231
FFREM     002262  G    #33-1434   *144-8321   144-8357    144-8372
FINACT    017756  G    #73-3781    144-8430
FRPSUP    020036  G    #74-3845    142-7943   142-7965    143-8065   143-8100
F$AU    = 000015       #28-1183    140-7761   140-7775
F$AUTO  = 000020       #28-1183    137-7643   137-7652
```

SYMBOL	VALUE		REFERENCES							
F$BGN	= 000040		#28-1183	28-1205	43-2211	44-2249	45-2276	46-2314	47-2373	48-2403	49-2439
			50-2499	51-2539	52-2502	53-2635	54-2671	134-7394	135-7417	136-7460	49-2439
			138-7670	138-7689	139-7723	140-7761	141-7792	141-7874	142-7891	142-7999	143-8017
			143-8135	144-8150	144-8626	144-8650	145-8684	145-8694	145-8697	145-8871	146-8888
			146-8898	146-8901	146-8957	147-8974	147-9102	148-9117	148-9286	149-9298	149-9417
			150-9429	150-9448	151-9472	152-9534	153-9585				
F$CLEA	= 000007		#28-1183	138-7670	138-7705						
F$DU	= 000016		#28-1183	139-7723	139-7742						
F$END	= 000041		#28-1183	28-1183	28-1183	28-1183	28-1183	28-1183	28-1183	28-1183	28-1183
			28-1183	28-1183	28-1183	28-1183	28-1183	28-1183	28-1183	28-1183	28-1205
			43-2222	44-2253	45-2285	46-2351	47-2379	48-2415	49-2476	50-2514	51-2581
			52-2606	53-2649	54-2703	134-7396	134-7400	136-7620	137-7652	138-7689	138-7705
			139-7739	139-7742	140-7770	140-7775	141-7792	141-7792	141-7792	141-7874	141-7874
			142-7891	142-7891	142-7891	142-7999	142-7999	143-8017	143-8017	143-8017	143-8135
			143-8135	144-8150	144-8150	144-8150	144-8626	144-8650	144-8650	145-8684	145-8684
			145-8684	145-8694	145-8697	145-8871	145-8871	146-8888	146-8888	146-8888	146-8898
			146-8901	146-8957	146-8957	147-8974	147-8974	147-9102	147-9102	148-9117
			148-9117	148-9117	148-9286	148-9286	149-9298	149-9298	149-9298	149-9417	149-9417
			150-9429	150-9429	150-9429	150-9448	150-9448	151-9495	152-9552	153-9585
F$HARD	= 000004		#28-1183	151-9472	151-9495						
F$HW	= 000013		#28-1183	30-1277	30-1285						
F$INIT	= 000006		#28-1183	136-7460	136-7620						
F$JMP	= 000050		#28-1183	134-7396	134-7396	138-7689	139-7739	139-7739	140-7770	140-7770	144-8626
			145-8694	145-8697	146-8898	146-8901					
F$MOD	= 000000		#28-1183	28-1205	153-9585						
F$MSG	= 000011		#28-1183	43-2211	43-2222	44-2249	44-2253	45-2276	45-2285	46-2314	46-2351
			47-2373	47-2379	48-2403	48-2415	49-2439	49-2476	50-2499	50-2514	51-2539
			51-2581	52-2602	52-2606	53-2635	53-2649	54-2671	54-2703		
F$PROT	= 000021		#28-1183	135-7417	135-7423						
F$PWR	= 000017		#28-1183								
F$RPT	= 000012		#28-1183	134-7394	134-7400						
F$SEG	= 000003		#28-1183								
F$SOFT	= 000005		#28-1183	152-9534	152-9552						
F$SRV	= 000010		#28-1183								
F$SUB	= 000002		#28-1183								
F$SW	= 000014		#28-1183	31-1304	31-1309						
F$TEST	= 000001		#28-1183	141-7792	141-7874	142-7891	142-7999	143-8017	143-8135	144-8150	144-8650
			145-8684	145-8871	146-8888	146-8957	147-8974	147-9102	148-9117	148-9286	149-9298
			149-9417	150-9429	150-9448						
GETBDR	020302	G	#75-3953	145-8736	146-8929						
GETCHR	020430	G	#76-4011	96-5240	96-5302						
GETLP1	020512	G	#77-4062	147-9013							
GETLP2	020664	G	#78-4128	147-9062							
GETPRM	030530		136-7472	136-7546	#136-7550	136-7557					
GETTIM	021014	G	#79-4178	142-7939	143-8061	145-8747	147-9017	147-9066	148-9148	148-9220	149-9349
GMANWD	002266	G	#33-1436	75-3954	*75-3958	75-3959	75-3960	*75-3981	*145-8777	145-8778	*145-8790
			145-8791	145-8792	*145-8847	145-8848	145-8849	*146-8950	146-8951		
GPRSOB	002466	G	#33-1527	111-6077	*147-9010	*147-9059					
G$CNTO	= 000200		#28-1183								
G$DELM	= 000372		#28-1183	147-9030							
G$DISP	= 000003		#28-1183								
G$EXCP	= 000400		#28-1183								

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL   VALUE   REFERENCES

```
G$HILI  = 000002        #28-1183
G$LOLI  = 000001        #28-1183
G$NO    = 000000        #28-1183
G$OFFS  = 000400        #28-1183      75-3959     145-8778    145-8791    145-8799    145-8848    146-8951    151-9484    151-9486
                        151-9488    151-9490    151-9492    152-9545    152-9547    152-9549
G$OFSI  = 000376        #28-1183      75-3959     145-8778    145-8791    145-8799    145-8848    146-8951    151-9484    151-9486
                        151-9488    151-9490    151-9492    152-9545    152-9547    152-9549
G$PRMA  = 000001        #28-1183     151-9484    151-9486
G$PRMD  = 000002        #28-1183      75-3959     145-8791    151-9488    151-9490    151-9492    152-9547
G$PRML  = 000000        #28-1183     145-8778    145-8799    145-8848    146-8951    152-9545    152-9549
G$RADA  = 000140        #28-1183
G$RADB  = 000000        #28-1183
G$RADD  = 000040        #28-1183      75-3959     145-8791    152-9547
G$RADL  = 000120        #28-1183     145-8778    145-8799    145-8848    146-8951    152-9545    152-9549
G$RADO  = 000020        #28-1183     151-9484    151-9486    151-9488    151-9490    151-9492
G$XFER  = 000004        #28-1183
G$YES   = 000010        #28-1183      75-3959     145-8778    145-8791    145-8799    145-8848    146-8951    151-9484    151-9486
                        151-9488    151-9490    151-9492    152-9545    152-9547    152-9549
HELP    = 000000        #2-5          28-1214     28-1233     29-1253     32-1321     32-1357     33-1791     39-1990     39-2010
                        39-2021      40-2042     40-2054     55-2706     135-7425    137-7624    137-7645    138-7672    138-7691
                        151-9450    151-9474    151-9497    152-9514    152-9536    152-9555    153-9577
HOE     = 100000  G     #32-1371
HWPTQ1   037432         151-9484    #151-9505
HWPTQ2   037450         151-9486    #151-9506
HWPTQ3   037503         151-9488    #151-9507
HWPTQ4   037531         151-9490    #151-9508
HWPTQ5   037670         151-9492    #151-9510
IBE     = 010000  G     #32-1371
IBM      002226  G     #33-1419     *60-3062     80-4243     96-5214     133-7364    *145-8748   *148-9153   *148-9225   *148-9252
IDU     = 000040  G     #32-1371
IER     = 020000  G     #32-1371
IESTAT   002234  G     #33-1422      56-2758     72-3748     80-4231     86-4598     92-4995    *100-5538    103-5665    104-5718
                        *105-5765    105-5766    *106-5790   *106-5791   106-5792    113-6189    114-6243    *116-6368   116-6369
                        *117-6393    *117-6394   117-6395    132-7269
INICHR   021072  G     #80-4230     147-9025    147-9074
INIDMA   021204  G     #81-4286     143-8081    143-8107    145-8815    148-9162    148-9235    148-9261    149-9363    149-9395
ISR     = 000100  G     #32-1371
IXE     = 004000  G     #32-1371
I$AU    = 000041        #28-1183    #140-7761   #140-7775
I$AUTO  = 000041        #28-1183    #137-7643   #137-7652
I$CLN   = 000041        #28-1183    #138-7670    138-7689   #138-7705
I$DU    = 000041        #28-1183    #139-7723   #139-7742
I$HRD   = 000041        151-9472   #151-9495
I$INIT  = 000041        #28-1183    #136-7460   #136-7620
I$MOD   = 000041        #28-1183     28-1205    #28-1205     153-9585   #153-9585
I$MSG   = 000041        #28-1183    #43-2211    #43-2222    #44-2249    #44-2253    #45-2276    #45-2285    #46-2314    #46-2351
                        #47-2373    #47-2379    #48-2403    #48-2415    #49-2439    #49-2476    #50-2499    #50-2514    #51-2539
                        #51-2581    #52-2602    #52-2606    #53-2635    #53-2649    #54-2671    #54-2703
I$PROT  = 000040        #28-1183    #135-7417
I$PTAB  = 000041        #28-1183
I$PWR   = 000041        #28-1183
I$RPT   = 000041        #28-1183    #134-7394   #134-7400
```

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL VALUE   REFERENCES

SYMBOL	VALUE		REFERENCES							
I$SEG	= 000041		#28-1183	141-7792	142-7891	143-8017	144-8150	145-8684	146-8888	147-8974	148-9117
			149-9298	150-9429						
I$SETU	= 000041		#28-1183							
I$SFT	= 000041		#152-9534	#152-9552						
I$SRV	= 000041		#28-1183							
I$SUB	= 000041		#28-1183	141-7792	142-7891	143-8017	144-8150	145-8684	146-8888	147-8974	148-9117
			149-9298	150-9429						
I$TST	= 000041		#28-1183	141-7792	#141-7792	141-7874	#141-7874	#141-7874	142-7891	#142-7891	142-7999
			#142-7999	#142-7999	143-8017	#143-8017	143-8135	#143-8135	#143-8135	144-8150	#144-8150
			144-8626	144-8650	#144-8650	145-8684	#145-8684	145-8694	145-8697	145-8871
			#145-8871	#145-8871	146-8888	#146-8888	146-8898	146-8901	146-8957	#146-8957
			147-8974	#147-8974	147-9102	#147-9102	#147-9102	148-9117	#148-9117	148-9286	#148-9286
			#148-9286	149-9298	#149-9298	149-9417	#149-9417	#149-9417	150-9429	#150-9429	150-9448
			#150-9448	#150-9448						
J$JMP	= 000167		#28-1183	134-7396	139-7739	140-7770				
LGRP1M	002230	G	#33-1420	109-5939	109-5991	142-7942	143-8064			
LGRP2M	002232	G	#33-1421	109-5958	109-5980					
LINBIT	021276	G	68-3532	#82-4356						
LNCTRA	002212	G	#33-1409	103-5663	104-5716	126-6967				
LNCTRO	= 000010	G	#32-1346							
LOE	= 040000	G	#32-1371							
LOPBCK	002176	G	#33-1397	74-3861	109-5940	109-5959	124-6855	124-6860	*136-7570	142-7895	143-8021
			145-8692	146-8896	149-9299					
LOT	= 000010	G	#32-1371							
LPCSLT	= 000036	G	#35-1852							
LPRA	002206	G	#33-1407	92-4993	127-7006					
LPRO	= 000004	G	#32-1344							
L$ACP	002110	G	#28-1231							
L$APT	002036	G	#28-1231							
L$AU	031134	G	#140-7761							
L$AUT	002070	G	#28-1231							
L$AUTO	030772	G	28-1231	#137-7643						
L$CCP	002106	G	#28-1231							
L$CLEA	030774	G	28-1231	#138-7670						
L$CO	002032	G	#28-1231							
L$DEPO	002011	G	#28-1231							
L$DESC	005376	G	28-1231	#39-2018						
L$DESP	002076	G	#28-1231							
L$DEVP	002060	G	#28-1231							
L$DISP	002124	G	28-1231	#29-1251						
L$DLY	002116	G	#28-1231	147-9030						
L$DTP	002040	G	#28-1231							
L$DTYP	002034	G	#28-1231							
L$DU	031024	G	28-1231	#139-7723						
L$DUT	002072	G	#28-1231							
L$DVTY	005366	G	28-1231	#39-2008						
L$EF	002052	G	#28-1231							
L$ENVI	002044	G	#28-1231							
L$ERRT	005316	G	28-1231	#33-1803						
L$ETP	002102	G	#28-1231							
L$EXP1	002046	G	#28-1231							
L$EXP4	002064	G	#28-1231							

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL	VALUE		REFERENCES		
L$EXP5	002066	G	#28-1231		
L$HARD	037354	G	28-1231	151-9472	#151-9472
L$HIME	002120	G	#28-1231		
L$HPCP	002016	G	#28-1231		
L$HPTP	002022	G	#28-1231		
L$HW	002152	G	28-1231	30-1277	#30-1277
L$ICP	002104	G	#28-1231		
L$INIT	030130	G	28-1231	#136-7460	
L$LADP	002026	G	#28-1231		
L$LAST	040246	G	28-1231	#153-9584	
L$LOAD	002100	G	#28-1231		
L$LUN	002074	G	#28-1231		
L$MREV	002050	G	#28-1231		
L$NAME	002000	G	#28-1231		
L$PRIO	002042	G	#28-1231		
L$PROT	030122	G	28-1231	#135-7417	
L$PRT	002112	G	#28-1231		
L$REPP	002062	G	#28-1231		
L$REV	002010	G	#28-1231		
L$RPT	030114	G	28-1231	#134-7394	
L$SOFT	037720	G	28-1231	152-9534	#152-9534
L$SPC	002056	G	#28-1231		
L$SPCP	002020	G	#28-1231		
L$SPTP	002024	G	#28-1231		
L$STA	002030	G	#28-1231		
L$SW	002164	G	28-1231	31-1304	#31-1304
L$TEST	002114	G	#28-1231		
L$TIML	002014	G	#28-1231		
L$UNIT	002012	G	#28-1231	136-7552	136-7602
L10000	002162		30-1277	#30-1285	
L10001	002170		31-1304	#31-1309	
L10002	013664		#43-2222		
L10003	014122		#44-2253		
L10004	014202		#45-2285		
L10005	014430		#46-2351		
L10006	014520		#47-2379		
L10007	014666		#48-2415		
L10010	015050		#49-2476		
L10011	015152		#50-2514		
L10012	015400		#51-2581		
L10013	015426		#52-2606		
L10014	015540		#53-2649		
L10015	015720		#54-2703		
L10016	030120		134-7396	#134-7400	
L10020	030770		#136-7620		
L10021	030772		#137-7652		
L10022	031022		138-7689	#138-7705	
L10023	031132		139-7739	#139-7742	
L10024	031140		140-7770	#140-7775	
L10025	031430		#141-7874		
L10026	032002		#142-7999		
L10027	032366		#143-8135		

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL VALUE   REFERENCES

SYMBOL	VALUE		REFERENCES							
L10030	034164		144-8626	#144-8650						
L10031	035056		145-8694	145-8697	#145-8871					
L10032	035304		146-8898	146-8901	#146-8957					
L10033	036034		#147-9102							
L10034	036644		#148-9286							
L10035	037270		#149-9417							
L10036	037350		#150-9448							
L10037	037432		151-9472	#151-9495						
L10040	037746		152-9534	#152-9552						
MAPLNS	= 000377	G	#32-1338	74-3892	74-3918	83-4421	109-5975	109-6000	119-6565	124-6871	124-6883
			129-7109	136-7567	145-8860	149-9327	151-9488	151-9488		
MFUNIT	005430	G	#41-2073	136-7604						
MMENAB	002324	G	#33-1459	72-3714	*136-7528					
MMPRES	002322	G	#33-1458	71-3651	71-3674	*136-7527	*136-7532	144-8187		
MMSR0	002316	G	#33-1456	71-3663	71-3676	136-7526	144-8266	144-8268		
MMSR3	002320	G	#33-1457	144-8230						
MODSUP	021324	G	#83-4393	145-8809						
MSFMT1	007630	G	#41-2104	86-4593						
MSFMT2	007670	G	#41-2105	86-4626						
MSG1	013666	G	43-2216	#43-2225						
MSG2	013744	G	43-2219	#43-2226						
MSG3	014023	G	43-2220	#43-2227						
MSLCNT	002314	G	#33-1451	57-2841	57-2860	*57-2879	84-4495			
MSLGET	021464	G	#84-4472	85-4560	100-5504	100-5521	125-6933			
MSLOOP	021600	G	69-3564	#85-4554	112-6153					
MSSRPT	021614	G	#86-4588	145-8783	145-8834					
MSTICK	002312	G	#33-1450	57-2873	*136-7488					
MUL16U	022030	G	#87-4661	112-6144	118-6484					
NDERPT	002166	G	#31-1307	88-4795	88-4797					
NDPMSG	013306	G	#41-2163	145-8791						
NEWCHR	022104	G	#88-4724	96-5295	96-5312					
NEWPAS	030510		136-7469	136-7534	#136-7539	136-7553				
NEWRES	030502		136-7466	#136-7536						
NEWSTA	030174		136-7463	#136-7474						
NUMLNS	= 000010	G	#32-1337	68-3526	73-3786	80-4249	81-4326	86-4631	96-5315	103-5664	104-5717
			108-5858	113-6188	114-6242	118-6534	133-7372	141-7850		
ODTSTA	034160		*144-8343	144-8394	144-8452	144-8571	#144-8639			
OOPS	022364	G	57-2852	57-2878	69-3566	#89-4826	129-7096			
OPTION	002164	G	#31-1306	136-7600	144-8611					
O$APTS	= 000000		#28-1183	28-1231						
O$AU	= 000000		#28-1183	28-1231						
O$BGNR	= 000001		#28-1183	#28-1212	28-1231					
O$BGNS	= 000001		#28-1183	#28-1212	28-1231					
O$DU	= 000001		#28-1183	#28-1212	28-1231					
O$ERRT	= 000001		#28-1183	#28-1212	28-1231					
O$GNSW	= 000001		#28-1183	#28-1212	28-1231					
O$POIN	= 000001		#28-1183	#28-1212	#28-1212	#28-1212	#28-1212	#28-1212	28-1212	28-1231
O$SETU	= 000000		#28-1183	28-1231	153-9584					
PARATB	002326	G	#33-1461	144-8208						
PARATE	002346	G	#33-1470							
PAROA	002326	G	#33-1462	72-3726						
PAR1A	002330	G	#33-1463							

SYMBOL VALUE   REFERENCES

SYMBOL	VALUE		REFERENCES								
PAR2A	002332	G	#33-1464								
PAR3A	002334	G	#33-1465								
PAR4A	002336	G	#33-1466								
PAR5A	002340	G	#33-1467	71-3656	144-8241						
PAR6A	002342	G	#33-1468	144-8212							
PAR7A	002344	G	#33-1469								
PASCNT	002236	G	#33-1423	*136-7533	*136-7537	*136-7545	*136-7547				
PCSLOT	= 000016	G	#35-1853								
PDRATB	002346	G	#33-1472	144-8220							
PDRATE	002366	G	#33-1481	144-8222							
PDR0A	002346	G	#33-1473								
PDR1A	002350	G	#33-1474								
PDR2A	002352	G	#33-1475								
PDR3A	002354	G	#33-1476								
PDR4A	002356	G	#33-1477								
PDR5A	002360	G	#33-1478								
PDR6A	002362	G	#33-1479								
PDR7A	002364	G	#33-1480								
PMSFLG	002270	G	#33-1437	*145-8784	145-8799	145-8832					
PMSMSG	013434	G	#41-2165	145-8799							
PNT	= 001000	G	#32-1371								
PREGRT	005350	G	#38-1979								
PREGO5	005326		#38-1960	43-2212	45-2277	46-2315	51-2540	53-2636	54-2672	56-2748	57-2808
			58-2919	59-2988	60-3053	61-3102	62-3238	63-3320	64-3387	65-3423	66-3462
			67-3495	68-3523	69-3559	70-3589	71-3648	72-3712	73-3781	74-3845	75-3953
			76-4011	77-4062	78-4128	79-4178	80-4230	81-4286	82-4356	83-4393	84-4472
			85-4554	86-4588	86-4608	87-4661	88-4724	89-4826	90-4866	91-4921	92-4991
			93-5022	94-5081	95-5167	96-5212	97-5348	98-5386	99-5451	100-5492	101-5570
			102-5615	103-5660	104-5713	107-5818	108-5852	109-5923	110-6031	112-6116	113-6183
			114-6237	115-6300	118-6455	119-6562	120-6591	121-6630	122-6669	123-6772	124-6838
			125-6922	126-6963	127-7002	132-7213	133-7310				
PRFRME	022612	G	62-3280	#90-4866							
PRI	= 002000	G	#32-1371								
PRI00	= 000000	G	#32-1371	142-7917	143-8043	145-8719	147-8998	148-9139	149-9323		
PRI01	= 000040	G	#32-1371								
PRI02	= 000100	G	#32-1371								
PRI03	= 000140	G	#32-1371								
PRI04	= 000200	G	#32-1371								
PRI05	= 000240	G	#32-1371	119-6564	136-7493	136-7611	142-7915	142-7916	142-7989	142-7996	143-8041
			143-8042	143-8124	143-8133	144-8270	144-8622	145-8716	145-8717	145-8718	145-8863
			145-8869	147-8995	147-8996	147-8997	147-9097	148-9136	148-9137	148-9138	148-9278
			148-9284	149-9320	149-9321	149-9322	149-9411				
PRI06	= 000300	G	#32-1371	33-1443	136-7489						
PRI07	= 000340	G	28-1231	#32-1371	72-3747	96-5255	105-5764	116-6367	144-8255	144-8327	
PROTBL	005216	G	#33-1724	79-4190							
PRPARE	022710	G	62-3282	#91-4921							
PRTLPR	023070	G	46-2349	48-2414	49-2475	51-2555	51-2575	#92-4991			
PUFIFO	023152	G	66-3472	#93-5022	144-8557	145-8811	147-9023	147-9072			
PUFIFR	023234	G	#94-5081	142-7952	142-7970	143-8078	143-8104	148-9158	148-9230	148-9256	149-9358
			149-9390								
PURRXB	023426	G	#95-5167	145-8814	147-9024	147-9073	148-9161	148-9234	148-9260	149-9362	149-9394
RBUFA	002204	G	#33-1406	62-3250	93-5024	94-5088	129-7091	136-7580	144-8515		

SYMBOL	VALUE		REFERENCES							
RBUFO	= 000002	G	#32-1342							
RDCHRS	023462	G	#96-5212	145-8820	147-9031	147-9079	148-9167	148-9239	148-9265	149-9367	149-9399
RDMAST	024034	G	#97-5348	121-6634							
REPCOD	024066	G	96-5285	#98-5386							
REPSMR	024220	G	#99-5451	142-7995	143-8131	145-8868	147-9090	148-9283	149-9415		
RESETT	024246	G	66-3467	#100-5492							
RRXNDN	024360	G	#101-5570	121-6639							
RTXNDN	024420	G	#102-5615	121-6636							
RXBCNT	002720	G	#33-1552	76-4014	*76-4024	96-5248	96-5252	129-7094	*129-7103	129-7104	129-7115
RXBDTX	= 000030	G	#32-1353	129-7104							
RXBEND	003122	G	#33-1555	76-4019	95-5173	129-7099					
RXBETX	= 000020	G	#32-1352	96-5248	96-5252						
RXBFUL	= 000100	G	#32-1354	33-1554	129-7094	129-7115					
RXBIPT	002716	G	#33-1551	129-7097	*129-7098	129-7099	*129-7101				
RXBOPT	002714	G	#33-1550	76-4016	*76-4022	95-5169	96-5236				
RXBSTA	002722	G	#33-1553	76-4021	95-5170	95-5171	129-7101				
RXCHRS	027364	G	#129-7090	145-8718	147-8997	148-9138	149-9322				
RXCNTB	003544	G	#33-1579	58-2929	59-2989	101-5579	*118-6515	123-6786	*123-6790		
RXDONF	002506	G	#33-1539	61-3106	*83-4397	91-4954	101-5572	*109-5927	*123-6798	*124-6842	*142-7922
			*143-8048	*143-8094							
RXDSBL	024460	G	74-3919	#103-5660	109-6001	124-6884					
RXENBL	024554	G	74-3925	#104-5713	109-6007	124-6890					
RXIEO	024650	G	96-5308	#105-5762	129-7118						
RXIE1	024702	G	96-5216	96-5250	#106-5790						
RXPTRB	003404	G	#33-1576	58-2920	59-2997	61-3122	91-4947	*118-6517	123-6777	*123-6782	
RXTOUT	002242	G	#33-1425	62-3245	*79-4192	96-5231	112-6143				
RXVECA	002170	G	#33-1394	*136-7563	145-8718	145-8865	147-8997	147-9099	148-9138	148-9280	149-9322
ROSLOT	= 000002	G	#35-1859	*70-3603	*77-4095	*78-4156	*82-4360	*84-4512			
R1SLOT	= 000004	G	#35-1858	*61-3203	*63-3361	*73-3799	*75-3982	*77-4095	*78-4156	*84-4512	*87-4680
			*122-6738								
R2SLOT	= 000006	G	#35-1857	*61-3203	*76-4027	*77-4095	*78-4156	*125-6936			
R3SLOT	= 000010	G	#35-1856	*63-3361	*77-4095	*78-4156					
R4SLOT	= 000012	G	#35-1855	*61-3203	*63-3361	*77-4095					
R5SLOT	= 000014	G	#35-1854	38-1967	*68-3537	*73-3799	*103-5686	*104-5739	*113-6210	*114-6264	
SAVBMP	024726	G	93-5041	94-5104	98-5403	#107-5818	144-8521				
SAVPRI	002246	G	#33-1427	*119-6563	120-6594						
SAVTEN	002244	G	#33-1426	*119-6567	120-6592						
SCBCTB	005044	G	#33-1613	77-4065	78-4131						
SCBCTE	005054	G	#33-1618	77-4082	78-4143						
SCBRTB	005054	G	#33-1619	77-4066	77-4080						
SCBRTE	005062	G	#33-1623	77-4078							
SCNSTB	005062	G	#33-1624	77-4067	77-4076	78-4132	78-4141				
SCNSTE	005066	G	#33-1627	77-4074	78-4139						
SCTPTB	005066	G	#33-1628	77-4068	77-4072	78-4133	78-4137				
SCTPTE	005074	G	#33-1632	77-4070	78-4135						
SDPBAS	005132	G	#33-1663	144-8248	144-8391	144-8453	147-9018	147-9019			
SDPEND	005152	G	#33-1679	147-9019							
SDP2B	005156	G	#33-1688	143-8062	144-8412	144-8549	148-9149	148-9150	149-9350		
SDP2E	005176	G	#33-1704	148-9150							
SFPTBL	002164	G	#31-1304								
SKPSTS	024774	G	100-5513	#108-5852							
SPLPRB	005102	G	#33-1646	149-9346							

```
SYMBOL   VALUE        REFERENCES

SPLPRE   005132   G   #33-1659     149-9405
SPLSUP   025052   G   #109-5923    149-9353     149-9385
STATA    002210   G   #33-1408     86-4607
STATO  = 000006   G   #32-1345
STGTRB   005276   G   #33-1773     118-6509
STPSW    025352   G   #110-6031    144-8401
SUCSS    034162       *144-8165    *144-8539    *144-8543    *144-8562    144-8600     144-8604     #144-8640
SVCGBL = 000000       #28-1183     #28-1192     28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      29-1251      30-1277      30-1277      31-1304      31-1304
                      33-1803      39-2008      39-2008      43-2211      44-2249      45-2276      46-2314      47-2373      48-2403
                      49-2439      50-2499      51-2539      52-2602      53-2635      54-2671      134-7394     135-7417     136-7460
                      137-7643     138-7670     139-7723     140-7761     151-9472     152-9534     #153-9584    153-9584     153-9584
                      153-9584
SVCINS = 000001       #28-1183     #28-1189     28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231
                      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      28-1231      29-1251
                      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251
                      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251
                      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251      29-1251
                      29-1251      29-1251      29-1251      29-1251      29-1251      30-1277      30-1277      30-1277      31-1304
                      31-1304      31-1304      39-2008      39-2008      39-2008      39-2008      39-2008      39-2008      39-2018
                      39-2018      39-2018      39-2018      39-2018      39-2018      43-2216      43-2216      43-2216      43-2216
                      43-2216      43-2216      43-2216      43-2216      43-2216      43-2216      43-2216      43-2216      43-2216
                      43-2216      43-2216      43-2219      43-2219      43-2219      43-2219      43-2219      43-2219      43-2219
                      43-2220      43-2220      43-2220      43-2220      43-2220      43-2220      43-2220      43-2220      43-2220
                      43-2220      43-2220      43-2220      43-2220      43-2220      43-2222      43-2222      43-2222      44-2251
                      44-2251      44-2251      44-2251      44-2251      44-2251      44-2251      44-2251      44-2251      44-2251
                      44-2251      44-2251      44-2251      44-2251      44-2251      44-2251      44-2251      44-2251      44-2253
                      44-2253      44-2253      45-2279      45-2279      45-2279      45-2279      45-2279      45-2279      45-2279
                      45-2279      45-2279      45-2279      45-2279      45-2279      45-2279      45-2279      45-2279      45-2279
                      45-2279      45-2279      45-2282      45-2282      45-2282      45-2282      45-2282      45-2282      45-2282
                      45-2282      45-2282      45-2282      45-2282      45-2282      45-2282      45-2282      45-2282      45-2282
                      45-2282      45-2282      45-2285      45-2285      45-2285      46-2320      46-2320      46-2320      46-2320
                      46-2320      46-2320      46-2320      46-2320      46-2320      46-2320      46-2320      46-2320      46-2320
                      46-2320      46-2320      46-2320      46-2320      46-2320      46-2320      46-2320      46-2320      46-2332
```

SYMBOL VALUE   REFERENCES

```
46-2332   46-2332   46-2332   46-2332   46-2332   46-2332   46-2332   46-2332   46-2332
46-2332   46-2332   46-2332   46-2332   46-2332   46-2332   46-2332   46-2332   46-2332
46-2332   46-2332   46-2345   46-2345   46-2345   46-2345   46-2345   46-2345   46-2345
46-2345   46-2345   46-2345   46-2345   46-2345   46-2345   46-2345   46-2347   46-2347
46-2347   46-2347   46-2347   46-2347   46-2347   46-2347   46-2347   46-2347   46-2347
46-2347   46-2347   46-2347   46-2347   46-2347   46-2351   46-2351   46-2351   47-2375
47-2375   47-2375   47-2375   47-2375   47-2375   47-2375   47-2375   47-2375   47-2375
47-2375   47-2375   47-2375   47-2375   47-2375   47-2375   47-2375   47-2375   47-2376
47-2376   47-2376   47-2376   47-2376   47-2376   47-2376   47-2376   47-2376   47-2376
47-2376   47-2376   47-2376   47-2376   47-2376   47-2376   47-2376   47-2376   47-2377
47-2377   47-2377   47-2377   47-2377   47-2377   47-2377   47-2377   47-2377   47-2377
47-2377   47-2377   47-2377   47-2377   47-2377   47-2377   47-2377   47-2377   47-2379
47-2379   47-2379   48-2407   48-2407   48-2407   48-2407   48-2407   48-2407   48-2407
48-2407   48-2407   48-2407   48-2407   48-2407   48-2407   48-2407   48-2407   48-2407
48-2407   48-2407   48-2407   48-2407   48-2407   48-2408   48-2408   48-2408   48-2408
48-2408   48-2408   48-2408   48-2408   48-2408   48-2408   48-2408   48-2408   48-2408
48-2408   48-2408   48-2408   48-2408   48-2408   48-2408   48-2408   48-2408   48-2411
48-2411   48-2411   48-2411   48-2411   48-2411   48-2411   48-2411   48-2411   48-2411
48-2411   48-2411   48-2411   48-2411   48-2411   48-2411   48-2411   48-2411   48-2411
48-2411   48-2411   48-2413   48-2413   48-2413   48-2413   48-2413   48-2413   48-2413
48-2413   48-2413   48-2413   48-2413   48-2413   48-2413   48-2413   48-2413   48-2413
48-2413   48-2413   48-2415   48-2415   48-2415   49-2442   49-2442   49-2442   49-2442
49-2442   49-2442   49-2442   49-2442   49-2442   49-2442   49-2442   49-2442   49-2442
49-2442   49-2442   49-2442   49-2442   49-2442   49-2442   49-2445   49-2445   49-2445
49-2445   49-2445   49-2445   49-2445   49-2445   49-2445   49-2445   49-2445   49-2445
49-2472   49-2472   49-2472   49-2472   49-2472   49-2472   49-2472   49-2472   49-2472
49-2472   49-2472   49-2472   49-2472   49-2472   49-2476   49-2476   49-2476   50-2501
50-2501   50-2501   50-2501   50-2501   50-2501   50-2501   50-2501   50-2501   50-2501
50-2501   50-2501   50-2501   50-2501   50-2501   50-2501   50-2501   50-2501   50-2508
50-2508   50-2508   50-2508   50-2508   50-2508   50-2508   50-2508   50-2508   50-2508
50-2508   50-2508   50-2514   50-2514   50-2514   51-2542   51-2542   51-2542   51-2542
51-2542   51-2542   51-2542   51-2542   51-2542   51-2542   51-2542   51-2542   51-2542
51-2542   51-2542   51-2542   51-2542   51-2542   51-2550   51-2550   51-2550   51-2550
51-2550   51-2550   51-2550   51-2550   51-2550   51-2550   51-2550   51-2550   51-2554
51-2554   51-2554   51-2554   51-2554   51-2554   51-2554   51-2554   51-2554   51-2554
51-2554   51-2554   51-2573   51-2573   51-2573   51-2573   51-2573   51-2573   51-2573
51-2573   51-2573   51-2573   51-2573   51-2573   51-2573   51-2573   51-2573   51-2573
51-2573   51-2573   51-2573   51-2573   51-2573   51-2581   51-2581   51-2581   52-2604
52-2604   52-2604   52-2604   52-2604   52-2604   52-2604   52-2604   52-2604   52-2604
52-2604   52-2604   52-2604   52-2604   52-2604   52-2604   52-2604   52-2604   52-2604
52-2604   52-2604   52-2606   52-2606   52-2606   53-2638   53-2638   53-2638   53-2638
53-2638   53-2638   53-2638   53-2638   53-2638   53-2638   53-2638   53-2638   53-2638
53-2643   53-2643   53-2643   53-2643   53-2643   53-2643   53-2643   53-2643   53-2643
53-2643   53-2643   53-2643   53-2643   53-2643   53-2647   53-2647   53-2647   53-2647
53-2647   53-2647   53-2647   53-2647   53-2647   53-2647   53-2647   53-2647   53-2647
53-2647   53-2647   53-2649   53-2649   53-2649   54-2674   54-2674   54-2674   54-2674
```

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL   VALUE   REFERENCES

54-2674	54-2674	54-2674	54-2674	54-2674	54-2674	54-2674	54-2674	54-2674
54-2674	54-2674	54-2674	54-2674	54-2674	54-2695	54-2695	54-2695	54-2695
54-2695	54-2695	54-2695	54-2695	54-2695	54-2695	54-2695	54-2695	54-2695
54-2695	54-2695	54-2699	54-2699	54-2699	54-2699	54-2699	54-2699	54-2699
54-2699	54-2699	54-2699	54-2699	54-2699	54-2699	54-2699	54-2699	54-2699
54-2699	54-2699	54-2699	54-2699	54-2699	54-2699	54-2699	54-2699	54-2703
54-2703	54-2703	62-3263	62-3263	62-3263	75-3959	75-3959	75-3959	75-3959
75-3959	75-3959	75-3959	75-3959	75-3959	75-3959	75-3959	75-3959	75-3959
75-3959	75-3959	75-3959	75-3978	75-3978	75-3978	75-3978	75-3978	75-3978
75-3978	75-3978	75-3978	75-3978	75-3978	75-3978	75-3978	75-3978	75-3978
75-3978	75-3978	86-4593	86-4593	86-4593	86-4593	86-4593	86-4593	86-4593
86-4593	86-4593	86-4593	86-4593	86-4593	86-4593	86-4593	86-4593	86-4626
86-4626	86-4626	86-4626	86-4626	86-4626	86-4626	86-4626	86-4626	86-4626
86-4626	86-4626	86-4626	86-4626	86-4626	86-4626	86-4626	86-4626	86-4626
86-4626	86-4626	86-4634	86-4634	86-4634	86-4634	86-4634	86-4634	86-4634
86-4634	86-4634	86-4634	86-4634	86-4634	86-4634	86-4634	86-4634	88-4766
88-4766	88-4766	88-4783	88-4783	88-4783	89-4828	89-4828	89-4828	89-4828
89-4828	89-4828	89-4828	89-4828	89-4828	89-4828	89-4828	89-4828	89-4830
89-4830	89-4830	89-4830	89-4830	89-4830	89-4830	89-4830	89-4830	89-4830
89-4830	89-4830	89-4830	89-4830	89-4830	89-4831	89-4831	89-4831	90-4888
90-4888	90-4888	91-4941	91-4941	91-4941	91-4960	91-4960	91-4960	92-4999
92-4999	92-4999	92-4999	92-4999	92-4999	92-4999	92-4999	92-4999	92-4999
92-4999	92-4999	94-5120	94-5120	94-5120	94-5134	94-5134	94-5134	96-5270
96-5270	96-5270	97-5359	97-5359	97-5359	98-5415	98-5415	98-5415	99-5462
99-5462	99-5462	100-5531	100-5531	100-5531	101-5585	101-5585	101-5585	102-5631
102-5631	102-5631	128-7051	128-7051	128-7051	134-7396	134-7396	134-7396	134-7396
134-7396	134-7396	134-7400	134-7400	134-7400	136-7462	136-7462	136-7462	136-7462
136-7462	136-7462	136-7463	136-7463	136-7463	136-7465	136-7465	136-7465	136-7465
136-7465	136-7465	136-7466	136-7466	136-7466	136-7468	136-7468	136-7468	136-7468
136-7468	136-7468	136-7469	136-7469	136-7469	136-7471	136-7471	136-7471	136-7471
136-7471	136-7471	136-7472	136-7472	136-7472	136-7475	136-7475	136-7475	136-7479
136-7479	136-7479	136-7479	136-7479	136-7479	136-7479	136-7479	136-7489	136-7489
136-7489	136-7489	136-7489	136-7489	136-7489	136-7489	136-7489	136-7489	136-7489
136-7489	136-7489	136-7489	136-7489	136-7489	136-7489	136-7489	136-7489	136-7536
136-7536	136-7536	136-7555	136-7555	136-7555	136-7555	136-7555	136-7555	136-7555
136-7555	136-7555	136-7555	136-7555	136-7556	136-7604	136-7604	136-7604	136-7604
136-7604	136-7604	136-7604	136-7604	136-7604	136-7620	136-7620	136-7620	136-7604
137-7652	137-7652	138-7683	138-7683	138-7683	138-7689	138-7689	138-7689	137-7652
138-7689	138-7689	138-7705	138-7705	138-7705	139-7731	139-7731	139-7731	138-7689
139-7731	139-7731	139-7731	139-7731	139-7731	139-7731	139-7731	139-7731	139-7731
139-7739	139-7739	139-7742	139-7742	139-7742	140-7770	140-7770	140-7770	139-7739
140-7770	140-7770	140-7775	140-7775	140-7775	141-7862	141-7862	141-7862	140-7770
141-7862	141-7862	141-7862	141-7862	141-7862	141-7862	141-7862	141-7862	141-7862
141-7864	141-7864	141-7864	141-7864	141-7864	141-7866	141-7866	141-7866	141-7864
141-7874	141-7874	142-7916	142-7916	142-7916	142-7916	142-7916	142-7916	141-7874
142-7916	142-7916	142-7990	142-7990	142-7990	142-7990	142-7990	142-7916	142-7916
								142-7999

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL VALUE   REFERENCES

142-7999	142-7999	143-8042	143-8042	143-8042	143-8042	143-8042	143-8042	143-8042
143-8042	143-8042	143-8042	143-8042	143-8042	143-8042	143-8042	143-8042	143-8042
143-8042	143-8042	143-8125	143-8125	143-8125	143-8125	143-8125	143-8125	143-8135
143-8135	143-8135	144-8321	144-8321	144-8321	144-8321	144-8321	144-8321	144-8615
144-8615	144-8615	144-8615	144-8615	144-8615	144-8615	144-8615	144-8615	144-8615
144-8615	144-8615	144-8615	144-8615	144-8615	144-8615	144-8615	144-8615	144-8615
144-8615	144-8615	144-8615	144-8619	144-8619	144-8626	144-8626	144-8626	144-8626
144-8626	144-8626	144-8650	144-8650	144-8650	145-8694	145-8694	145-8694	145-8694
145-8694	145-8694	145-8695	145-8695	145-8695	145-8696	145-8696	145-8696	145-8697
145-8697	145-8697	145-8697	145-8697	145-8697	145-8717	145-8717	145-8717	145-8717
145-8717	145-8717	145-8717	145-8717	145-8717	145-8717	145-8717	145-8717	145-8717
145-8717	145-8717	145-8717	145-8717	145-8717	145-8718	145-8718	145-8718	145-8718
145-8718	145-8718	145-8718	145-8718	145-8718	145-8718	145-8718	145-8718	145-8718
145-8718	145-8718	145-8718	145-8718	145-8718	145-8728	145-8728	145-8728	145-8728
145-8728	145-8728	145-8728	145-8728	145-8728	145-8728	145-8728	145-8728	145-8728
145-8728	145-8728	145-8728	145-8728	145-8728	145-8778	145-8778	145-8778	145-8778
145-8778	145-8778	145-8778	145-8778	145-8778	145-8778	145-8778	145-8778	145-8778
145-8778	145-8778	145-8778	145-8778	145-8791	145-8791	145-8791	145-8791	145-8791
145-8791	145-8791	145-8791	145-8791	145-8791	145-8791	145-8791	145-8791	145-8791
145-8791	145-8791	145-8791	145-8791	145-8791	145-8791	145-8791	145-8791	145-8791
145-8791	145-8791	145-8799	145-8799	145-8799	145-8799	145-8799	145-8799	145-8799
145-8799	145-8799	145-8799	145-8799	145-8799	145-8799	145-8799	145-8799	145-8799
145-8799	145-8799	145-8831	145-8831	145-8831	145-8831	145-8831	145-8831	145-8831
145-8831	145-8831	145-8831	145-8831	145-8831	145-8831	145-8831	145-8831	145-8831
145-8831	145-8831	145-8848	145-8848	145-8848	145-8848	145-8848	145-8848	145-8848
145-8848	145-8848	145-8848	145-8848	145-8848	145-8848	145-8848	145-8848	145-8848
145-8848	145-8848	145-8864	145-8864	145-8864	145-8864	145-8864	145-8864	145-8865
145-8865	145-8865	145-8865	145-8865	145-8865	145-8871	145-8871	145-8871	146-8898
146-8898	146-8898	146-8898	146-8898	146-8898	146-8899	146-8899	146-8899	146-8900
146-8900	146-8900	146-8901	146-8901	146-8901	146-8901	146-8901	146-8901	146-8919
146-8919	146-8919	146-8919	146-8919	146-8919	146-8919	146-8919	146-8919	146-8919
146-8919	146-8919	146-8919	146-8919	146-8919	146-8919	146-8919	146-8919	146-8951
146-8951	146-8951	146-8951	146-8951	146-8951	146-8951	146-8951	146-8951	146-8951
146-8951	146-8951	146-8951	146-8951	146-8951	146-8951	146-8951	146-8951	146-8957
146-8957	146-8957	147-8996	147-8996	147-8996	147-8996	147-8996	147-8996	147-8996
147-8996	147-8996	147-8996	147-8996	147-8996	147-8996	147-8996	147-8996	147-8996
147-8997	147-8997	147-8997	147-8997	147-8997	147-8997	147-8997	147-8997	147-8997
147-8997	147-8997	147-9030	147-9030	147-9030	147-9030	147-9030	147-9030	147-9030
147-9030	147-9030	147-9030	147-9030	147-9030	147-9030	147-9030	147-9030	147-9098
147-9098	147-9098	147-9098	147-9098	147-9098	147-9099	147-9099	147-9099	147-9099
147-9099	147-9099	147-9102	147-9102	147-9102	148-9137	148-9137	148-9137	148-9137
148-9137	148-9137	148-9137	148-9137	148-9137	148-9137	148-9137	148-9137	148-9137
148-9137	148-9137	148-9137	148-9137	148-9137	148-9138	148-9138	148-9138	148-9138
148-9138	148-9138	148-9138	148-9138	148-9138	148-9138	148-9138	148-9138	148-9138
148-9138	148-9138	148-9138	148-9138	148-9138	148-9279	148-9279	148-9279	148-9279
148-9279	148-9279	148-9280	148-9280	148-9280	148-9280	148-9280	148-9280	148-9286
148-9286	148-9286	149-9321	149-9321	149-9321	149-9321	149-9321	149-9321	149-9321
149-9321	149-9321	149-9321	149-9321	149-9321	149-9321	149-9321	149-9321	149-9321
149-9321	149-9321	149-9322	149-9322	149-9322	149-9322	149-9322	149-9322	149-9322
149-9322	149-9322	149-9322	149-9322	149-9322	149-9322	149-9322	149-9322	149-9322

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL VALUE   REFERENCES

```
                    149-9322    149-9322    149-9412    149-9412    149-9412    149-9412    149-9412    149-9412    149-9417
                    149-9417    149-9417    150-9443    150-9443    150-9443    150-9443    150-9443    150-9443    150-9443
                    150-9443    150-9443    150-9443    150-9443    150-9443    150-9448    150-9448    150-9448    151-9472
                    151-9472    151-9472    151-9484    151-9484    151-9484    151-9484    151-9484    151-9484    151-9484
                    151-9484    151-9484    151-9484    151-9484    151-9484    151-9486    151-9486    151-9486    151-9486
                    151-9486    151-9486    151-9486    151-9486    151-9486    151-9486    151-9486    151-9486    151-9488
                    151-9488    151-9488    151-9488    151-9488    151-9488    151-9488    151-9488    151-9488    151-9488
                    151-9488    151-9488    151-9488    151-9488    151-9488    151-9490    151-9490    151-9490    151-9490
                    151-9490    151-9490    151-9490    151-9490    151-9490    151-9490    151-9490    151-9490    151-9490
                    151-9492    151-9492    151-9492    151-9492    151-9492    151-9492    151-9492    151-9492    151-9492
                    151-9492    151-9492    151-9492    151-9492    151-9492    151-9492    151-9492    151-9492    151-9495
                    151-9495    151-9495    152-9534    152-9534    152-9534    152-9545    152-9545    152-9545    152-9545
                    152-9545    152-9545    152-9545    152-9545    152-9545    152-9547    152-9547    152-9547    152-9547
                    152-9547    152-9547    152-9547    152-9547    152-9547    152-9547    152-9547    152-9547    152-9547
                    152-9549    152-9549    152-9549    152-9549    152-9549    152-9549    152-9549    152-9549    152-9549
                    153-9584    153-9584    152-9552    152-9552    152-9552    153-9584    153-9584    153-9584    153-9584
                    153-9584    153-9584    153-9584    153-9584    153-9584
SVCSUB  = 000001    #28-1183    #28-1191
SVCTAG  = 000001    #28-1183    #28-1193    30-1285     31-1309     43-2222     44-2253     45-2285     46-2351     47-2379
                    48-2415     49-2476     50-2514     51-2581     52-2606     53-2649     54-2703     75-3959     134-7400
                    136-7620    137-7652    138-7705    139-7742    140-7775    141-7874    142-7999    143-8135    144-8650
                    145-8778    145-8791    145-8799    145-8848    145-8871    146-8951    146-8957    147-9102    148-9286
                    149-9417    150-9448    151-9495    152-9552
SVCTST  = 000001    #28-1183    #28-1190    141-7792    142-7891    143-8017    144-8150    145-8684    146-8888    147-8974
                    148-9117    149-9298    150-9429
SWAPO     025372  G  #111-6065   147-9012    147-9014    147-9061    147-9063
SWPTQ1    037746     152-9545    #152-9562
SWPTQ2    040022     152-9547    #152-9563
SWPTQ3    040111     152-9549    #152-9564
S$LSYM  = 010000    #28-1183    #30-1285    #31-1309    #43-2222    #44-2253    #45-2285    #46-2351    #47-2379    #48-2415
                    #49-2476    #50-2514    #51-2581    #52-2606    #53-2649    #54-2703    75-3959     75-3959     75-3959
                    #75-3959    #134-7400   #136-7620   #137-7652   #138-7705   #139-7742   #140-7775   #141-7874   #142-7999
                    #143-8135   #144-8650   145-8778    145-8778    145-8778    #145-8778   145-8791    145-8791    145-8791
                    #145-8791   145-8799    145-8799    #145-8799   145-8848    145-8848    145-8848    #145-8848
                    #145-8871   146-8951    146-8951    146-8951    #146-8951   #146-8957   #147-9102   #148-9286   #149-9417
                    #150-9448   #151-9495   #152-9552
TERMSG    013521  G  #41-2166    146-8951
TIMER1    002302  G  #33-1446    *57-2817    57-2818     57-2835     128-7041    *128-7043
TIMER2    002304  G  #33-1447    *128-7046
TIMER3    002306  G  #33-1448    *128-7047   *128-7049
TNUM    = 000012  G  #136-7622   141-7793    #141-7793   141-7794    142-7898    #142-7898   142-7899    143-8024    #143-8024
                    143-8025    144-8153    #144-8153   144-8154    145-8685    #145-8685   145-8686    146-8889    #146-8889
                    146-8890    147-8975    #147-8975   147-8976    148-9118    #148-9118   148-9119    149-9302    #149-9302
                    149-9303    150-9430    #150-9430   150-9431
TP4BRT    027530  G  #130-7148   144-8317
TP4FLG    002256  G  #33-1431    *64-3388    64-3390     *65-3425    65-3427     *130-7151   *131-7181   *136-7504   *136-7523
                    *141-7812   144-8272    144-8335    144-8409    144-8414    144-8576    144-8583
TP4RTN    027552  G  #131-7178   136-7500    136-7522    141-7800    144-8172
TP4VEC    002254  G  #33-1430    130-7150    131-7180    *136-7499   136-7509    *136-7521   136-7530    *141-7799   141-7857
                    *144-8171   144-8623
TRPAD2    017216  G  #65-3427    130-7148
TSTNUM    002224  G  #33-1418    107-5820    *141-7794   *142-7899   *143-8025   *144-8154   *145-8686   *146-8890   *147-8976
```

SYMBOL VALUE   REFERENCES

SYMBOL	VALUE		REFERENCES							
			*148-9119	*149-9303	*150-9431					
TXAD1A	002214	G	#33-1410	72-3754	144-8480					
TXAD10	= 000012	G	#32-1347							
TXAD2A	002216	G	#33-1411	72-3750	72-3755	113-6186	114-6240	144-8476	144-8481	
TXAD20	= 000014	G	#32-1348							
TXBFCA	002220	G	#33-1412	72-3753	108-5866	144-8479				
TXBFCO	= 000016	G	#32-1349							
TXCHA	002204	G	#33-1406	80-4245	133-7366					
TXCHRO	= 000002	G	#32-1343							
TXCNTB	003504	G	#33-1578	*80-4246	*81-4321	102-5624	*115-6336	*118-6500	132-7238	*132-7263	133-7340
			*133-7368							
TXDBLF	002510	G	#33-1540	*96-5215	96-5246	*96-5257	129-7106	*129-7113		
TXDMA	027574	G	#132-7213	142-7916	143-8042	145-8717	148-9137	149-9321		
TXDONE	025442	G	96-5309	#112-6116						
TXDONF	002504	G	#33-1538	62-3246	*83-4396	96-5232	102-5617	*109-5926	112-6124	112-6152	*124-6841
			*132-7240	*133-7342	*142-7921	*143-8047	*143-8093			
TXDSBL	025552	G	#113-6183	119-6566	129-7110					
TXENBL	025646	G	96-5256	#114-6237	118-6467	120-6593	144-8446	149-9328		
TXENBM	002250	G	#33-1428	96-5251	*129-7111					
TXFRPR	025742	G	#115-6300	142-7955	142-7973					
TXIEO	026036	G	112-6154	#116-6365	142-7986	143-8118				
TXIE1	026070	G	62-3240	96-5217	#117-6393					
TXINTF	002252	G	#33-1429	*81-4316	*83-4395	97-5349	*115-6331	*124-6840	*132-7257	*142-7923	*143-8049
			*143-8095							
TXPTRB	003344	G	#33-1575	80-4237	*80-4242	81-4307	115-6322	*118-6501	132-7248	133-7358	*133-7363
TXRINI	026114	G	74-3869	74-3883	74-3901	74-3912	83-4417	83-4429	109-5947	109-5966	109-5984
			109-5994	#118-6455	124-6867	124-6878				
TXROFF	026370	G	88-4765	88-4782	98-5414	#119-6562				
TXRON	026424	G	88-4767	88-4784	98-5416	#120-6591				
TXRREP	026450	G	#121-6630	142-7987	143-8091	143-8122	145-8825	147-9036	147-9084	148-9172	148-9244
			148-9270	149-9372	149-9404					
TXRXLB	005236	G	#33-1747	46-2317	51-2567	58-2931	59-2991	63-3338	68-3524	88-4761	88-4777
			88-4799	*118-6510	123-6773					
TXRXLE	005276	G	#33-1764							
TXSCHR	027726	G	#133-7310	147-8996						
TXVECA	002172	G	#33-1395	*136-7565	142-7916	142-7990	143-8042	143-8125	145-8717	145-8864	147-8996
			147-9098	148-9137	148-9279	149-9321	149-9412			
T$ARGC	= 000002		#28-1231	28-1231	#28-1231	28-1231	28-1231	#28-1231	28-1231	28-1231	#28-1231
			28-1231	28-1231	#28-1231	28-1231	28-1231	#28-1231	28-1231	28-1231	#43-2216
			43-2216	43-2216	#43-2219	43-2219	43-2219	#43-2220	43-2220	43-2220	#44-2251
			44-2251	#44-2251	44-2251	44-2251	#45-2279	45-2279	45-2279	45-2279	45-2279
			#45-2282	45-2282	#45-2282	45-2282	45-2282	#46-2320	46-2320	#46-2320	46-2320
			#46-2320	46-2320	46-2320	#46-2332	46-2332	#46-2332	46-2332	#46-2332	46-2332
			46-2332	#46-2345	46-2345	#46-2345	46-2345	#46-2345	46-2345	46-2345	#46-2347
			46-2347	#46-2347	46-2347	46-2347	#47-2375	47-2375	#47-2375	47-2375	47-2375
			#47-2376	47-2376	#47-2376	47-2376	47-2376	#47-2377	47-2377	#47-2377	47-2377
			47-2377	#48-2407	48-2407	#48-2407	48-2407	#48-2407	48-2407	48-2407	#48-2408
			48-2408	#48-2408	48-2408	#48-2408	48-2408	48-2408	#48-2411	48-2411	#48-2411
			48-2411	#48-2411	48-2411	48-2411	#48-2413	48-2413	#48-2413	48-2413	48-2413
			#49-2442	49-2442	#49-2442	49-2442	49-2442	#49-2445	49-2445	#49-2445	49-2445
			49-2445	#49-2472	49-2472	#49-2472	49-2472	49-2472	#50-2501	50-2501	#50-2501
			50-2501	50-2501	#50-2508	50-2508	#50-2508	50-2508	#50-2508	50-2508	50-2508

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL	VALUE	REFERENCES							
		#51-2542	51-2542	#51-2542	51-2542	51-2542	#51-2550	51-2550	51-2550	#51-2554
		51-2554	#51-2554	51-2554	51-2554	#51-2573	51-2573	#51-2573	51-2573	#51-2573
		51-2573	#51-2573	51-2573	#51-2573	51-2573	52-2604	51-2573	51-2573	#52-2604
		52-2604	#52-2604	52-2604	#52-2604	52-2604	52-2604	#53-2638	53-2638	#53-2638
		53-2638	53-2638	#53-2643	53-2643	#53-2643	53-2643	53-2643	#53-2647	53-2647
		53-2647	#54-2674	54-2674	#54-2674	54-2674	54-2674	#54-2695	54-2695	54-2695
		#54-2699	54-2699	#54-2699	54-2699	#54-2699	54-2699	#54-2699	54-2699	54-2699
		#75-3978	75-3978	#75-3978	75-3978	75-3978	#86-4593	86-4593	86-4593	#86-4626
		86-4626	#86-4626	86-4626	#86-4626	86-4626	#86-4626	86-4626	#86-4626	86-4626
		#86-4626	86-4626	86-4626	#86-4634	86-4634	86-4634	#89-4830	89-4830	89-4830
		#92-4999	92-4999	#92-4999	92-4999	#92-4999	92-4999	92-4999	#136-7604	136-7604
		#136-7604	136-7604	136-7604	#139-7731	139-7731	#139-7731	139-7731	139-7731	#144-8615
		144-8615	#144-8615	144-8615	#144-8615	144-8615	144-8615	#145-8728	145-8728	#145-8728
		145-8728	145-8728	#145-8831	145-8831	#145-8831	145-8831	145-8831	#146-8919	146-8919
		#146-8919	146-8919	146-8919						
T$CODE	= 000130	#75-3959	75-3959	#75-3959	75-3959	#75-3959	75-3959	#145-8778	145-8778	#145-8778
		145-8778	#145-8778	145-8778	#145-8791	145-8791	#145-8791	145-8791	#145-8791	145-8791
		#145-8799	145-8799	#145-8799	145-8799	#145-8799	145-8799	#145-8848	145-8848	#145-8848
		145-8848	#145-8848	145-8848	#146-8951	146-8951	#146-8951	146-8951	#146-8951	146-8951
		#151-9484	151-9484	#151-9484	151-9484	#151-9484	151-9484	#151-9486	151-9486	#151-9486
		151-9486	#151-9486	151-9486	#151-9488	151-9488	#151-9488	151-9488	#151-9488	151-9488
		#151-9490	151-9490	#151-9490	151-9490	#151-9490	151-9490	#151-9492	151-9492	#151-9492
		151-9492	#151-9492	151-9492	#152-9545	152-9545	#152-9545	152-9545	#152-9545	152-9545
		152-9549	#152-9549	152-9549	152-9547	#152-9547	152-9547	#152-9549	152-9549	#152-9549
T$ERRN	= 022125	#28-1183	#89-4828	89-4828	#141-7862	141-7862	#150-9443	150-9443		
T$EXCP	= 000000	#75-3959	75-3959	#145-8791	145-8791	#151-9484	151-9484	#151-9486	151-9486	#151-9488
		151-9488	#151-9490	151-9490	#151-9492	151-9492	#152-9547	152-9547		
T$FLAG	= 000040	#134-7396	#134-7396	134-7396	#138-7689	138-7689	138-7689	138-7689	#139-7739	#139-7739
		139-7739	#140-7770	#140-7770	140-7770	#144-8626	144-8626	144-8626	144-8626	#145-8694
		#145-8694	145-8694	145-8694	#145-8697	#145-8697	145-8697	145-8697	#146-8898	#146-8898
		146-8898	146-8898	#146-8901	#146-8901	146-8901	146-8901			
T$GMAN	= 000000	#28-1183	#75-3959	#75-3959	#145-8791	#145-8791				
T$HILI	= 177777	#75-3959	75-3959	#145-8791	145-8791	#151-9484	151-9484	#151-9486	151-9486	#151-9488
		151-9488	#151-9490	151-9490	#151-9492	151-9492	#152-9547	152-9547		
T$LAST	= 000001	#28-1183	#153-9584							
T$LOLI	= 000000	#75-3959	75-3959	#145-8791	145-8791	#151-9484	151-9484	#151-9486	151-9486	#151-9488
		151-9488	#151-9490	151-9490	#151-9492	151-9492	#152-9547	152-9547		
T$LSYM	= 010000	#28-1183	28-1183	30-1285	31-1309	43-2222	44-2253	45-2285	46-2351	47-2379
		48-2415	49-2476	50-2514	51-2581	52-2606	53-2649	54-2703	134-7400	136-7620
		137-7652	138-7705	139-7742	140-7775	141-7874	142-7999	143-8135	144-8650	145-8871
		146-8957	147-9102	148-9286	149-9417	150-9448	151-9495	152-9552		
T$LTNO	= 000012	#153-9584								
T$NEST	= 177777	#28-1183	28-1205	#28-1205	28-1205	30-1277	#30-1277	30-1277	30-1285	30-1285
		30-1285	#30-1285	31-1304	#31-1304	31-1304	31-1309	31-1309	31-1309	#31-1309
		43-2211	#43-2211	43-2211	43-2222	43-2222	43-2222	#43-2222	44-2249	#44-2249
		44-2249	44-2253	44-2253	44-2253	#44-2253	45-2276	#45-2276	45-2276	45-2285
		45-2285	45-2285	#45-2285	46-2314	#46-2314	46-2314	46-2351	46-2351	46-2351
		#46-2351	47-2373	#47-2373	47-2373	47-2379	47-2379	47-2379	#47-2379	48-2403
		#48-2403	48-2415	48-2415	48-2415	48-2415	#48-2415	49-2439	#49-2439	49-2439
		49-2476	49-2476	49-2476	#49-2476	50-2499	#50-2499	50-2499	50-2514	50-2514
		50-2514	#50-2514	51-2539	#51-2539	51-2539	51-2581	51-2581	51-2581	#51-2581

SYMBOL CROSS REFERENCE   CREF   03.00

SYMBOL VALUE   REFERENCES

```
                       52-2602    #52-2602    52-2602     52-2606     52-2606    52-2606    #52-2606    53-2635    #53-2635
                       53-2635    53-2649     53-2649     53-2649    #53-2649    54-2671   #54-2671     54-2671    54-2703
                       54-2703    54-2703    #54-2703     134-7394   #134-7394   134-7394   134-7400    134-7400   134-7400
                      #134-7400   135-7417   #135-7417    135-7417    135-7423   135-7423   135-7423   #135-7423   136-7460
                      #136-7460   136-7460    136-7620    136-7620    136-7620   #136-7620   137-7643   #137-7643   137-7643
                       137-7652   137-7652    137-7652   #137-7652    138-7670  #138-7670   138-7670    138-7705   138-7705
                       138-7705  #138-7705    139-7723   #139-7723    139-7723   139-7742   139-7742    139-7742  #139-7742
                       140-7761  #140-7761    140-7761    140-7775    140-7775   140-7775  #140-7775    141-7792  #141-7792
                       141-7792   141-7874    141-7874    141-7874   #141-7874   142-7891  #142-7891    142-7891   142-7999
                       142-7999   142-7999   #142-7999    143-8017   #143-8017   143-8017   143-8135    143-8135   143-8135
                      #143-8135   144-8150   #144-8150    144-8150    144-8650   144-8650   144-8650   #144-8650   145-8684
                      #145-8684   145-8684    145-8871    145-8871    145-8871  #145-8871   146-8888   #146-8888   146-8888
                       146-8957   146-8957    146-8957   #146-8957    147-8974  #147-8974   147-8974    147-9102   147-9102
                       147-9102  #147-9102    148-9117   #148-9117    148-9117   148-9286   148-9286    148-9286  #148-9286
                       149-9298  #149-9298    149-9298    149-9417    149-9417   149-9417  #149-9417    150-9429  #150-9429
                       150-9429   150-9448    150-9448    150-9448   #150-9448   151-9472  #151-9472    151-9472   151-9495
                       151-9495   151-9495   #151-9495    152-9534   #152-9534   152-9534   152-9552    152-9552   152-9552
                      #152-9552   153-9585    153-9585    153-9585   #153-9585
T$NS0   = 000000      #28-1205    153-9585
T$NS1   = 000005      #30-1277    30-1285    #31-1304     31-1309    #43-2211    43-2222   #44-2249     44-2253   #45-2276
                       45-2285   #46-2314     46-2351    #47-2373     47-2379   #48-2403    48-2415    #49-2439    49-2476
                      #50-2499    50-2514    #51-2539     51-2581    #52-2602    52-2606   #53-2635     53-2649   #54-2671
                       54-2703   #134-7394    134-7400   #135-7417    135-7423  #136-7460   136-7620   #137-7643   137-7652
                      #138-7670   138-7705   #139-7723    139-7742   #140-7761   140-7775  #141-7792    141-7874  #142-7891
                       142-7999  #143-8017    143-8135   #144-8150    144-8650  #145-8684   145-8871   #146-8888   146-8957
                      #147-8974   147-9102   #148-9117    148-9286   #149-9298   149-9417  #150-9429    150-9448  #151-9472
                       151-9495  #152-9534    152-9552
T$PTNU  = 000000      #28-1183
T$SAVL  = 177777      #28-1183
T$SEGL  = 177777      #28-1183
T$SUBN  = 000000      #28-1183   #141-7792   #142-7891    #143-8017   #144-8150  #145-8684  #146-8888   #147-8974  #148-9117
                      #149-9298  #150-9429
T$TAGL  = 177777      #28-1183
T$TAGN  = 010041      #28-1183    30-1277     30-1277    #30-1277     31-1304    31-1304   #31-1304     43-2211    43-2211
                       43-2211    44-2249     44-2249    #44-2249     45-2276    45-2276   #45-2276     46-2314    46-2314
                       46-2314    47-2373     47-2373    #47-2373     48-2403    48-2403   #48-2403     49-2439    49-2439
                       49-2439    50-2499     50-2499    #50-2499     51-2539    51-2539   #51-2539     52-2602    52-2602
                      #52-2602    53-2635     53-2635    #53-2635     54-2671    54-2671   #54-2671     134-7394   134-7394
                      #134-7394   135-7417    135-7417   #135-7417    136-7460   136-7460  #136-7460    137-7643   137-7643
                      #137-7643   138-7670    138-7670   #138-7670    139-7723   139-7723  #139-7723    140-7761   140-7761
                       140-7761   141-7792    141-7792   #141-7792    142-7891   142-7891  #142-7891    143-8017   143-8017
                      #143-8017   144-8150    144-8150   #144-8150    145-8684   145-8684  #145-8684    146-8888   146-8888
                      #146-8888   147-8974    147-8974   #147-8974    148-9117   148-9117  #148-9117    149-9298   149-9298
                      #149-9298   150-9429    150-9429   #150-9429    151-9472   151-9472  #151-9472    152-9534   152-9534
                      #152-9534
T$TEMP  = 000000      #29-1251    29-1251     29-1251    #29-1251     29-1251    29-1251   #29-1251     29-1251    29-1251
                      #29-1251    29-1251     29-1251    #29-1251     29-1251    29-1251   #29-1251     29-1251    29-1251
                      #29-1251    29-1251     29-1251    #29-1251     29-1251    29-1251   #29-1251     29-1251    29-1251
                      #29-1251    29-1251     29-1251    #29-1251    #30-1285    30-1285   #31-1309     31-1309   #43-2222
                       43-2222   #44-2253     44-2253    #45-2285     45-2285   #46-2351    46-2351    #47-2379    47-2379
                      #48-2415    48-2415    #49-2476     49-2476    #50-2514     50-2514   #51-2581     51-2581   #52-2606
                       52-2606   #53-2649     53-2649    #54-2703     54-2703   #75-3959    75-3959    #75-3959    75-3959
```

SYMBOL	VALUE	REFERENCES

```
                             #75-3959     75-3959    #134-7396    134-7396   #134-7400    134-7400   #135-7423    135-7423   #136-7620
                             136-7620    #137-7652    137-7652   #138-7689    138-7689   #138-7705    138-7705   #139-7739    139-7739
                            #139-7742    139-7742    #140-7770    140-7770   #140-7775    140-7775   #141-7874    141-7874   #142-7999
                             142-7999    #143-8135    143-8135   #144-8626    144-8626   #144-8650    144-8650   #145-8694    145-8694
                            #145-8697    145-8697    #145-8778    145-8778   #145-8778    145-8778   #145-8778    145-8778   #145-8791
                             145-8791    #145-8791    145-8791   #145-8791    145-8791   #145-8799    145-8799   #145-8799    145-8799
                            #145-8799    145-8799    #145-8848    145-8848   #145-8848    145-8848   #145-8871    145-8871   #145-8871
                             145-8871    #146-8898    146-8898   #146-8901    146-8901   #146-8951    146-8951   #146-8951    146-8951
                            #146-8951    146-8951    #146-8957    146-8957   #147-9102    147-9102   #148-9286    148-9286   #149-9417
                             149-9417    #150-9448    150-9448   #151-9484    151-9484   #151-9484    151-9484   #151-9484    151-9484
                            #151-9486    151-9486    #151-9486    151-9486   #151-9486    151-9486   #151-9488    151-9488   #151-9488
                             151-9488    #151-9488    151-9488   #151-9490    151-9490   #151-9490    151-9490   #151-9490    151-9490
                            #151-9492    151-9492    #151-9492    151-9492   #151-9492    151-9492   #151-9495    151-9495   #152-9545
                             152-9545    #152-9545    152-9545   #152-9545    152-9545   #152-9547    152-9547   #152-9547    152-9547
                            #152-9547    152-9547    #152-9549    152-9549   #152-9549    152-9549   #152-9549    152-9549   #152-9552
                             152-9552    #153-9585    153-9585
T$TEST  = 000012            #28-1183     141-7792    #141-7792    141-7792    142-7891   #142-7891    142-7891    143-8017   #143-8017
                             143-8017    144-8150    #144-8150    144-8150    145-8684    145-8684    146-8888   #146-8888
                             146-8888    147-8974    #147-8974    147-8974    148-9117   #148-9117    148-9117    149-9298   #149-9298
                             149-9298    150-9429    #150-9429    150-9429    153-9584
T$TSTM  = 177777            #28-1183      43-2216     43-2219     43-2220     43-2222     44-2251     44-2253     45-2279     45-2282
                             45-2285     46-2320     46-2332     46-2345     46-2347     46-2351     47-2375     47-2376     47-2377
                             47-2379     48-2407     48-2408     48-2411     48-2415     49-2442     49-2445     49-2472
                             49-2476     50-2501     50-2508     50-2514     51-2542     51-2550     51-2554     51-2573     51-2581
                             52-2604     52-2606     53-2638     53-2643     53-2647     53-2649     54-2674     54-2695     54-2699
                             54-2703     62-3263     75-3959     75-3978     86-4593     86-4626     86-4634     88-4766     88-4783
                             89-4828     89-4830     89-4831     90-4888     91-4941     91-4960     92-4999     94-5120     94-5134
                             96-5270     97-5359     98-5415     99-5462    100-5531    101-5585    102-5631    128-7051    134-7400
                             136-7462    136-7465    136-7468    136-7471    136-7475    136-7479    136-7489    136-7536    136-7555
                             136-7604    136-7620    137-7652    138-7683    138-7689    138-7705    139-7731    139-7742    140-7775
                             141-7862    141-7864    141-7866    141-7874    142-7916    142-7990    142-7999    143-8042    143-8125
                             143-8135    144-8321    144-8615    144-8619    144-8626    144-8650    145-8694    145-8695    145-8697
                             145-8717    145-8718    145-8728    145-8778    145-8791    145-8799    145-8831    145-8848    145-8864
                             145-8865    145-8871    146-8898    146-8899    146-8919    146-8951    146-8957    147-8996
                             147-8997    147-9098    147-9099    147-9102    148-9137    148-9138    148-9279    148-9280    148-9286
                             149-9321    149-9322    149-9412    149-9417    150-9443    150-9448
T$TSTS  = 000001            #28-1183     #141-7792   #142-7891   #143-8017   #144-8150   #145-8684   #146-8888   #147-8974   #148-9117
                            #149-9298   #150-9429
T$$AU   = 010024            #140-7761    140-7770    140-7775
T$$AUT  = 010021            #137-7643    137-7652
T$$CLE  = 010022            #138-7670    138-7689    138-7705
T$$DU   = 010023            #139-7723    139-7739    139-7742
T$$HAR  = 010037            #151-9472    151-9472    151-9495
T$$HW   = 010000            #30-1277     30-1277     30-1285
T$$INI  = 010020            #136-7460    136-7620
T$$MSG  = 010015            #43-2211     43-2222    #44-2249     44-2253    #45-2276     45-2285    #46-2314     46-2351    #47-2373
                            47-2379     #48-2403     48-2415    #49-2439     49-2476    #50-2499     50-2514    #51-2539     51-2581
                            #52-2602     52-2606    #53-2635     53-2649    #54-2671     54-2703
T$$PRO  = 010017            #135-7417
T$$RPT  = 010016            #134-7394    134-7396    134-7400
T$$SOF  = 010040            #152-9534    152-9534    152-9552
T$$SW   = 010001            #31-1304     31-1304     31-1309
```

SYMBOL   VALUE   REFERENCES

SYMBOL	VALUE		REFERENCES							
T$$TES	= 010036		#141-7792	141-7874	#142-7891	142-7999	#143-8017	143-8135	#144-8150	144-8626	144-8650
			#145-8684	145-8694	145-8697	145-8871	#146-8888	146-8898	146-8901	146-8957	#147-8974
			147-9102	#148-9117	148-9286	#149-9298	149-9417	#150-9429	150-9448		
T1	031142	G	29-1251	#141-7792							
T10	037272	G	29-1251	#150-9429							
T2	031432	G	29-1251	#142-7891							
T3	032004	G	29-1251	#143-8017							
T4	032370	G	29-1251	#144-8150							
T5	034166	G	29-1251	#145-8684							
T6	035060	G	29-1251	#146-8888							
T7	035306	G	29-1251	#147-8974							
T8	036036	G	29-1251	#148-9117							
T9	036646	G	29-1251	#149-9298							
UAM	= 000200	G	#32-1371								
UBRFMT	007532	G	#41-2103	75-3978							
UNITN	002200	G	#33-1399	*136-7540	*136-7551	136-7552	136-7555	136-7604	141-7864		
UNSDIV	026530	G	57-2876	#122-6669							
UPDCHR	026664	G	61-3132	61-3171	61-3183	62-3284	#123-6772				
VANSUP	026762	G	#124-6838	147-9021	147-9070	148-9152	148-9224	148-9251			
WAIBIS	027160	G	62-3251	96-5237	#125-6922	144-8492					
WORD1	002240	G	#33-1424	*136-7505	136-7506	*136-7524	136-7525				
WTWLNC	027234	G	118-6464	#126-6963	144-8443	145-8772	145-8861	146-8944			
WTWLPR	027264	G	118-6466	#127-7002	144-8445	146-8941					
X$ALWA	= 000000		#28-1183								
X$FALS	= 000040		#28-1183								
X$OFFS	= 000400		#28-1183								
X$TRUE	= 000020		#28-1183								
$PATCH	040172	G	#153-9574								

MACRO NAME	REFERENCES								
BCOMPL	136-7463	136-7466	136-7469	136-7556	145-8696	146-8900			
BGNAU	140-7761								
BGNAUT	137-7643								
BGNCLN	138-7670								
BGNDU	139-7723								
BGNHRD	151-9472								
BGNHW	30-1277								
BGNINI	136-7460								
BGNMOD	28-1205								
BGNMSG	43-2211	44-2249	45-2276	46-2314	47-2373	48-2403	49-2439	50-2499	51-2539	52-2602
	53-2635	54-2671							
BGNPRO	135-7417								
BGNRPT	134-7394								
BGNSFT	152-9534								
BGNSW	31-1304								
BGNTST	141-7792	142-7891	143-8017	144-8150	145-8684	146-8888	147-8974	148-9117	149-9298	150-9429
BNCOMP	136-7472								
BREAK	89-4831	128-7051							
BRESET	136-7475	136-7536	138-7683						
CLOCK	136-7479								
CLRVEC	142-7990	143-8125	145-8864	145-8865	147-9098	147-9099	148-9279	148-9280	149-9412	
DELAY	147-9030								
DESCRI	39-2018								
DEVTYP	39-2008								
DISPAT	29-1251								
DOCLN	141-7866								
DODU	141-7864								
ENDAU	140-7775								
ENDAUT	137-7652								
ENDCLN	138-7705								
ENDDU	139-7742								
ENDHRD	151-9495								
ENDHW	30-1285								
ENDINI	136-7620								
ENDMOD	153-9585								
ENDMSG	43-2222	44-2253	45-2285	46-2351	47-2379	48-2415	49-2476	50-2514	51-2581	52-2606
	53-2649	54-2703							
ENDPRO	135-7423								
ENDRPT	134-7400								
ENDSFT	152-9552								
ENDSW	31-1309								
ENDTST	141-7874	142-7999	143-8135	144-8650	145-8871	146-8957	147-9102	148-9286	149-9417	150-9448
EQUALS	32-1371								
ERRDF	141-7862	150-9443							
ERROR	62-3263	88-4766	88-4783	90-4888	91-4941	91-4960	94-5120	94-5134	96-5270	97-5359
	98-5415	99-5462	100-5531	101-5585	102-5631	144-8619			
ERRSF	89-4828								
ERRTBL	33-1803								
EXIT	134-7396	138-7689	139-7739	140-7770	144-8626	145-8694	145-8697	146-8898	146-8901	
GMANID	75-3959	145-8791							
GMANIL	145-8778	145-8799	145-8848	146-8951					
GPHARD	136-7555								

MACRO NAME	REFERENCES								
GPRMA	151-9484	151-9486							
GPRMD	#75-3959	75-3959	#145-8791	145-8791	151-9488	151-9490	151-9492	152-9547	
GPRML	#145-8778	145-8778	#145-8799	145-8799	#145-8848	145-8848	#146-8951	146-8951	152-9545	152-9549
HEADER	28-1231								
LASTAD	153-9584								
MANUAL	145-8695	146-8899							
MEMORY	144-8321								
M$BYTE	#28-1231	28-1231	28-1231	28-1231					
M$CHEC	#134-7396	134-7396	#138-7689	138-7689	#139-7739	139-7739	#140-7770	140-7770	#144-8626	144-8626
	#145-8694	145-8694	#145-8697	145-8697	#146-8898	146-8898	#146-8901	146-8901		
M$CNTO	#75-3959	75-3959	#145-8778	145-8778	#145-8791	145-8791	#145-8799	145-8799	#145-8848	145-8848
	#146-8951	146-8951	#151-9484	151-9484	#151-9486	151-9486	#151-9488	151-9488	#151-9490	151-9490
	#151-9492	151-9492	#152-9545	152-9545	#152-9547	152-9547	#152-9549	152-9549		
M$COUN	#43-2216	43-2216	#43-2219	43-2219	#43-2220	43-2220	#44-2251	44-2251	#45-2279	45-2279
	#45-2282	45-2282	#46-2320	46-2320	46-2320	#46-2332	46-2332	46-2332	#46-2345	46-2345
	46-2345	#46-2347	46-2347	#47-2375	47-2375	#47-2376	47-2376	#47-2377	47-2377	#48-2407
	48-2407	48-2407	#48-2408	48-2408	48-2408	#48-2411	48-2411	#48-2413	48-2413	
	#49-2442	49-2442	#49-2445	49-2445	#49-2472	49-2472	#50-2501	50-2501	#50-2508	50-2508
	50-2508	#51-2542	51-2542	#51-2550	51-2550	#51-2554	51-2554	#51-2573	51-2573	51-2573
	51-2573	51-2573	51-2573	#52-2604	52-2604	52-2604	#53-2638	53-2638	#53-2643	53-2643
	#53-2647	53-2647	#54-2674	54-2674	54-2695	#54-2699	54-2699	54-2699	54-2699
	#75-3978	75-3978	#86-4593	86-4593	#86-4626	86-4626	86-4626	86-4626	86-4626
	#86-4634	86-4634	#89-4830	89-4830	#92-4999	92-4999	92-4999	#136-7604	136-7604	#139-7731
	139-7731	#144-8615	144-8615	144-8615	#145-8728	145-8728	#145-8831	145-8831	#146-8919	146-8919
M$DATA	#28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231
	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	#28-1231	28-1231	28-1231	28-1231
	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231
	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	#39-2008
	39-2008	#39-2018	39-2018							
M$DECR	#30-1285	30-1285	#31-1309	31-1309	#43-2222	43-2222	#44-2253	44-2253	#45-2285	45-2285
	#46-2351	46-2351	#47-2379	47-2379	#48-2415	48-2415	#49-2476	49-2476	#50-2514	50-2514
	#51-2581	51-2581	#52-2606	52-2606	#53-2649	53-2649	#54-2703	54-2703	#134-7400	134-7400
	#135-7423	135-7423	#136-7620	136-7620	#137-7652	137-7652	#138-7705	138-7705	#139-7742	139-7742
	#140-7775	140-7775	#141-7874	141-7874	#142-7999	142-7999	#143-8135	143-8135	#144-8650	144-8650
	#145-8871	145-8871	#146-8957	146-8957	#147-9102	147-9102	#148-9286	148-9286	#149-9417	149-9417
	#150-9448	150-9448	#151-9495	151-9495	#152-9552	152-9552	#153-9585	153-9585		
M$DEFA	#75-3959	75-3959	#145-8778	145-8778	#145-8791	145-8791	#145-8799	145-8799	#145-8848	145-8848
	#146-8951	146-8951	#151-9484	151-9484	#151-9486	151-9486	#151-9488	151-9488	#151-9490	151-9490
	#151-9492	151-9492	#152-9545	152-9545	#152-9547	152-9547	#152-9549	152-9549		
M$ENDE	#30-1285	#31-1309	#43-2222	#44-2253	#45-2285	#46-2351	#47-2379	#48-2415	#49-2476	#50-2514
	#51-2581	#52-2606	#53-2649	#54-2703	#134-7400	#136-7620	#137-7652	#138-7705	#139-7742	#140-7775
	#141-7874	#142-7999	#143-8135	#144-8650	#145-8871	#146-8957	#147-9102	#148-9286	#149-9417	#150-9448
	#151-9495	#152-9552	#153-9585							
M$ERRI	#89-4828	89-4828	#141-7862	141-7862	#150-9443	150-9443				
M$EXCP	#75-3959	75-3959	75-3959	#145-8791	145-8791	145-8791	#151-9484	151-9484	151-9484	#151-9486
	151-9486	151-9486	#151-9488	151-9488	151-9488	#151-9490	151-9490	151-9490	#151-9492	151-9492
	151-9492	#152-9547	152-9547	152-9547						
M$EXI1	#134-7396	#138-7689	138-7689	#139-7739	#140-7770	#144-8626	144-8626	#145-8694	145-8694	#145-8697
	145-8697	#146-8898	146-8898	#146-8901	146-8901					
M$EXSE	#134-7396	#138-7689	#139-7739	#140-7770	#144-8626	#145-8694	#145-8697	#146-8898	#146-8901	
M$EXTJ	#134-7396	134-7396	#138-7689	#139-7739	139-7739	#140-7770	140-7770	#144-8626	#145-8694	#145-8697
	#146-8898	#146-8901								

MACRO NAME   REFERENCES

```
M$GEN      #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231
           #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231
           #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231
           #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231
           #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231
           #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231
           #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #29-1251   29-1251   #30-1277   30-1277
           #30-1277   30-1277   #30-1285   30-1285   #31-1304   31-1304   #31-1304   31-1304   #31-1309   31-1309
           #33-1803   33-1803   #39-2008   39-2008   #39-2018   39-2018   #43-2211   43-2211   #43-2222   43-2222
           #44-2249   44-2249   #44-2253   44-2253   #45-2276   45-2276   #45-2285   45-2285   #46-2314   46-2314
           #46-2351   46-2351   #47-2373   47-2373   #47-2379   47-2379   #48-2403   48-2403   #48-2415   48-2415
           #49-2439   49-2439   #49-2476   49-2476   #50-2499   50-2499   #50-2514   50-2514   #51-2539   51-2539
           #51-2581   51-2581   #52-2602   52-2602   #52-2606   52-2606   #53-2635   53-2635   #53-2649   53-2649
           #54-2671   54-2671   #54-2703   54-2703   #75-3959   75-3959   #134-7394  134-7394  #134-7400  134-7400
           #135-7417  135-7417  #136-7460  136-7460  #136-7620  136-7620  #137-7643  137-7643  #137-7652  137-7652
           #138-7670  138-7670  #138-7705  138-7705  #139-7723  139-7723  #139-7742  139-7742  #140-7761  140-7761
           #140-7775  140-7775  #141-7792  141-7792  #141-7874  141-7874  #142-7891  142-7891  #142-7999  142-7999
           #143-8017  143-8017  #143-8135  143-8135  #144-8150  144-8150  #144-8650  144-8650  #145-8684  145-8684
           #145-8778  145-8778  #145-8791  145-8791  #145-8799  145-8799  #145-8848  145-8848  #145-8871  145-8871
           #146-8888  146-8888  #146-8951  146-8951  #146-8957  146-8957  #147-8974  147-8974  #147-9102  147-9102
           #148-9117  148-9117  #148-9286  148-9286  #149-9298  149-9298  #149-9417  149-9417  #150-9429  150-9429
           #150-9448  150-9448  #151-9472  151-9472  #151-9495  151-9495  #152-9534  152-9534  #152-9552  152-9552
           #153-9584  153-9584
M$GENB     #75-3959   75-3959   #145-8778  145-8778  #145-8791  145-8791  #145-8799  145-8799  #145-8848  145-8848
           #146-8951  146-8951
M$GETS     #30-1285   30-1285   #31-1309   31-1309   #43-2222   43-2222   #44-2253   44-2253   #45-2285   45-2285
           #46-2351   46-2351   #47-2379   47-2379   #48-2415   48-2415   #49-2476   49-2476   #50-2514   50-2514
           #51-2581   51-2581   #52-2606   52-2606   #53-2649   53-2649   #54-2703   54-2703   #134-7400  134-7400
           #135-7423  135-7423  #136-7620  136-7620  #137-7652  137-7652  #138-7705  138-7705  #139-7742  139-7742
           #140-7775  140-7775  #141-7874  141-7874  #142-7999  142-7999  #143-8135  143-8135  #144-8650  144-8650
           #145-8871  145-8871  #146-8957  146-8957  #147-9102  147-9102  #148-9286  148-9286  #149-9417  149-9417
           #150-9448  150-9448  #151-9495  151-9495  #152-9552  152-9552  #153-9585  153-9585
M$GETT     #134-7396  #138-7689  #139-7739  #140-7770  #144-8626  #145-8694  #145-8697  #146-8898  #146-8901
M$GNGB     #28-1205   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231
           28-1231    #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231
           28-1231    #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231
           28-1231    #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231
           28-1231    #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231
           28-1231    #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231
           28-1231    #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #29-1251   29-1251   #30-1277
           30-1277    30-1277    #31-1304   31-1304   31-1304   #33-1803   33-1803   #39-2008   39-2008   #39-2018
           39-2018    #43-2211   43-2211   #44-2249   44-2249   #45-2276   45-2276   #46-2314   46-2314   #47-2373
           47-2373    #48-2403   48-2403   #49-2439   49-2439   #50-2499   50-2499   #51-2539   51-2539   #52-2602
           52-2602    #53-2635   53-2635   #54-2671   54-2671   #134-7394  134-7394  #135-7417  135-7417  #136-7460
           151-9472   #152-9534  152-9534  #153-9584  153-9584  #139-7723  139-7723  #140-7761  140-7761  #151-9472
M$GNIN     #28-1231   28-1231   28-1231   28-1231   28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231
           #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231
           #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231
           #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231   #28-1231   28-1231
```

MACRO CROSS REFERENCE   CREF   03.00

MACRO NAME   REFERENCES

#28-1231	28-1231	#28-1231	#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231	28-1231
#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231
#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231
#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231
#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231	#28-1231	28-1231
#29-1251	29-1251	#29-1251	29-1251	#29-1251	29-1251	#29-1251	29-1251	#29-1251	29-1251
#29-1251	29-1251	#29-1251	29-1251	#29-1251	29-1251	#29-1251	29-1251	#29-1251	29-1251
#29-1251	29-1251	#30-1277	30-1277	#31-1304	31-1304	#39-2008	#39-2008	39-2008	39-2008
#39-2018	#39-2018	39-2018	39-2018	#43-2216	#43-2216	43-2216	#43-2216	43-2216	43-2216
#43-2216	43-2216	43-2216	#43-2219	#43-2219	43-2219	#43-2219	43-2219	43-2219	#43-2219
43-2219	43-2219	#43-2220	#43-2220	43-2220	#43-2220	43-2220	43-2220	#43-2220	43-2220
43-2220	#43-2222	43-2222	#44-2251	#44-2251	44-2251	#44-2251	44-2251	#44-2251	44-2251
44-2251	#44-2251	44-2251	44-2251	#44-2253	44-2253	#45-2279	#45-2279	45-2279	#45-2279
45-2279	#45-2279	45-2279	45-2279	#45-2279	45-2279	45-2279	#45-2282	#45-2282	45-2282
#45-2282	45-2282	#45-2282	45-2282	45-2282	#45-2282	45-2282	45-2282	#45-2282	45-2285
#46-2320	#46-2320	46-2320	#46-2320	46-2320	#46-2320	46-2320	#46-2320	46-2320	46-2320
#46-2320	46-2320	46-2320	#46-2332	#46-2332	46-2332	#46-2332	46-2332	#46-2332	46-2332
#46-2332	46-2332	46-2332	#46-2332	46-2332	46-2332	#46-2345	#46-2345	46-2345	#46-2345
46-2345	#46-2345	46-2345	#46-2345	46-2345	46-2345	46-2345	#46-2345	46-2345	#46-2347
#46-2347	46-2347	#46-2347	46-2347	#46-2347	46-2347	46-2347	#46-2347	46-2347	46-2347
#46-2351	46-2351	#47-2375	#47-2375	47-2375	#47-2375	47-2375	#47-2375	47-2375	47-2375
#47-2375	47-2375	47-2375	#47-2376	#47-2376	47-2376	#47-2376	47-2376	#47-2376	47-2376
47-2376	#47-2376	47-2376	47-2376	#47-2377	#47-2377	47-2377	#47-2377	47-2377	#47-2377
47-2377	47-2377	#47-2377	47-2377	47-2377	#47-2379	47-2379	#48-2407	#48-2407	48-2407
#48-2407	48-2407	#48-2407	48-2407	#48-2407	48-2407	48-2407	#48-2407	48-2407	48-2407
#48-2408	#48-2408	48-2408	#48-2408	48-2408	#48-2408	48-2408	#48-2408	48-2408	48-2408
#48-2408	48-2408	48-2408	#48-2411	#48-2411	48-2411	#48-2411	48-2411	#48-2411	48-2411
#48-2411	48-2411	48-2411	#48-2411	48-2411	48-2411	48-2411	#48-2413	#48-2413	48-2413
48-2413	#48-2413	48-2413	48-2413	#48-2413	48-2413	48-2413	#48-2413	48-2415	#49-2442
#49-2442	49-2442	#49-2442	49-2442	#49-2442	49-2442	#49-2442	49-2442	49-2442	49-2442
#49-2445	#49-2445	49-2445	#49-2445	49-2445	#49-2445	49-2445	49-2445	#49-2445	49-2445
49-2445	#49-2472	#49-2472	49-2472	#49-2472	49-2472	#49-2472	49-2472	49-2472	#49-2472
49-2472	49-2472	#49-2476	49-2476	#50-2501	#50-2501	50-2501	#50-2501	50-2501	#50-2501
50-2501	50-2501	#50-2501	50-2501	50-2501	#50-2508	#50-2508	50-2508	#50-2508	50-2508
#50-2508	50-2508	#50-2508	50-2508	50-2508	#50-2508	50-2508	50-2508	#50-2514	50-2514
#51-2542	#51-2542	51-2542	#51-2542	51-2542	#51-2542	51-2542	51-2542	#51-2542	51-2542
51-2542	#51-2550	#51-2550	51-2550	#51-2550	51-2550	51-2550	51-2550	#51-2550	51-2550
#51-2554	#51-2554	51-2554	#51-2554	51-2554	#51-2554	51-2554	51-2554	#51-2554	51-2554
51-2554	#51-2573	#51-2573	51-2573	#51-2573	51-2573	#51-2573	51-2573	#51-2573	51-2573
#51-2581	51-2581	#52-2604	#52-2604	52-2604	#52-2604	52-2604	#52-2604	52-2604	#52-2604
52-2604	52-2604	#52-2604	52-2604	52-2604	#52-2606	52-2606	#53-2638	#53-2638	53-2638
#53-2638	53-2638	#53-2638	53-2638	53-2638	#53-2638	53-2638	53-2638	#53-2643	53-2643
53-2643	#53-2643	53-2643	#53-2643	53-2643	53-2643	#53-2643	53-2643	53-2643	#53-2647
#53-2647	53-2647	#53-2647	53-2647	53-2647	#53-2647	53-2647	53-2647	#53-2649	53-2649
#54-2674	#54-2674	54-2674	#54-2674	54-2674	#54-2674	54-2674	54-2674	#54-2674	54-2674
54-2674	#54-2695	#54-2695	54-2695	#54-2695	54-2695	54-2695	#54-2695	54-2695	54-2695
#54-2699	#54-2699	54-2699	#54-2699	54-2699	#54-2699	54-2699	#54-2699	54-2699	#54-2699
54-2699	54-2699	#54-2699	54-2699	54-2699	#54-2703	54-2703	#62-3263	62-3263	#75-3959
75-3959	#75-3959	75-3959	#75-3959	75-3959	#75-3959	75-3959	75-3959	75-3959	75-3959
75-3959	#75-3978	#75-3978	75-3959	#75-3978	75-3978	#75-3978	75-3978	75-3978	#75-3978
75-3978	75-3978	#86-4593	#86-4593	86-4593	#86-4593	86-4593	86-4593	#86-4593	86-4593

MACRO CROSS REFERENCE   CREF   03.00

MACRO NAME   REFERENCES

86-4593	#86-4626	#86-4626	86-4626	#86-4626	86-4626	#86-4626	86-4626	#86-4626	86-4626
#86-4626	86-4626	#86-4626	86-4626	#86-4626	86-4626	86-4626	#86-4626	86-4626	86-4626
#86-4634	#86-4634	86-4634	#86-4634	86-4634	86-4634	#86-4634	86-4634	86-4634	#88-4766
88-4766	#88-4783	88-4783	#89-4828	#89-4828	89-4828	#89-4828	89-4828	#89-4828	89-4828
#89-4828	89-4828	#89-4830	#89-4830	89-4830	#89-4830	89-4830	89-4830	#89-4830	89-4830
89-4830	#89-4831	89-4831	#90-4888	90-4888	#91-4941	91-4941	#91-4960	91-4960	#92-4999
#92-4999	92-4999	#92-4999	92-4999	#92-4999	92-4999	#92-4999	92-4999	92-4999	#92-4999
92-4999	92-4999	#94-5120	94-5120	#94-5134	94-5134	#96-5270	96-5270	#97-5359	97-5359
#98-5415	98-5415	#99-5462	99-5462	#100-5531	100-5531	#101-5585	101-5585	#102-5631	102-5631
#128-7051	128-7051	#134-7396	134-7396	#134-7396	134-7396	#134-7400	134-7400	#136-7462	136-7462
#136-7462	136-7462	#136-7463	136-7463	#136-7465	136-7465	#136-7465	136-7465	#136-7466	136-7466
#136-7468	136-7468	#136-7468	136-7468	#136-7469	136-7469	#136-7471	136-7471	#136-7471	136-7471
#136-7472	136-7472	#136-7475	136-7475	#136-7479	136-7479	#136-7479	136-7479	#136-7479	136-7479
#136-7489	#136-7489	136-7489	#136-7489	136-7489	#136-7489	136-7489	#136-7489	136-7489	#136-7489
136-7489	136-7489	#136-7536	136-7536	#136-7555	136-7555	#136-7555	136-7555	#136-7555	136-7555
#136-7556	136-7556	#136-7604	#136-7604	136-7604	#136-7604	136-7604	#136-7604	136-7604	136-7604
#136-7604	136-7604	136-7604	#136-7620	136-7620	#137-7652	137-7652	#138-7683	138-7683	#138-7689
138-7689	#138-7689	138-7689	#138-7705	138-7705	#139-7731	139-7731	#139-7731	139-7731	139-7731
#139-7731	139-7731	139-7731	#139-7731	139-7731	139-7731	#139-7739	139-7739	#139-7731	139-7731
#139-7742	139-7742	#140-7770	140-7770	#140-7770	140-7770	#140-7775	140-7775	#141-7862	#141-7862
141-7862	#141-7862	141-7862	#141-7862	141-7862	#141-7862	141-7862	#141-7864	141-7864	#141-7864
141-7864	#141-7866	141-7866	#141-7874	141-7874	#142-7916	#142-7916	142-7916	#142-7916	#141-7864
#142-7916	142-7916	#142-7916	142-7916	#142-7916	142-7916	142-7916	#142-7990	142-7990	#142-7990
142-7990	#142-7999	142-7999	#143-8042	#143-8042	143-8042	#143-8042	143-8042	#143-8042	143-8042
#143-8042	143-8042	#143-8042	143-8042	143-8042	#143-8125	143-8125	#143-8125	143-8125	#143-8135
143-8135	#144-8321	144-8321	#144-8321	144-8321	#144-8615	#144-8615	144-8615	#144-8615	144-8615
#144-8615	144-8615	#144-8615	144-8615	144-8615	#144-8615	144-8615	144-8615	#144-8619	144-8619
#144-8626	144-8626	#144-8626	144-8626	#144-8650	144-8650	#145-8694	145-8694	#145-8694	145-8694
#145-8695	145-8695	#145-8696	145-8696	#145-8697	145-8697	#145-8697	145-8697	#145-8717	#145-8717
145-8717	#145-8717	145-8717	#145-8717	145-8717	#145-8717	145-8717	#145-8717	145-8717	145-8717
#145-8718	#145-8718	145-8718	#145-8718	145-8718	#145-8718	145-8718	#145-8718	145-8718	#145-8718
145-8718	145-8718	#145-8728	#145-8728	145-8728	#145-8728	145-8728	#145-8728	145-8728	145-8728
#145-8728	145-8728	145-8728	#145-8778	145-8778	#145-8778	145-8778	#145-8778	145-8778	#145-8778
145-8778	145-8778	145-8778	#145-8791	145-8791	#145-8791	145-8791	#145-8791	145-8791	#145-8791
145-8791	145-8791	145-8791	145-8791	145-8791	#145-8799	145-8799	145-8831	145-8831	#145-8791
145-8799	#145-8799	145-8799	145-8799	145-8799	#145-8831	#145-8831	145-8831	#145-8831	#145-8799
#145-8831	145-8831	145-8831	#145-8831	145-8831	145-8831	#145-8848	145-8848	#145-8848	145-8848
#145-8848	145-8848	#145-8848	145-8848	145-8848	145-8848	#145-8864	145-8864	#145-8864	145-8864
#145-8865	145-8865	#145-8865	145-8865	#145-8871	145-8871	#146-8898	146-8898	#146-8898	146-8898
#146-8899	146-8899	#146-8900	146-8900	#146-8901	146-8901	#146-8901	146-8901	#146-8919	146-8919
146-8919	#146-8919	146-8919	#146-8919	146-8919	146-8919	#146-8919	146-8919	146-8919	#146-8951
146-8951	#146-8951	146-8951	#146-8951	146-8951	#146-8951	146-8951	146-8951	146-8951	#146-8957
146-8957	#147-8996	#147-8996	147-8996	#147-8996	147-8996	#147-8996	147-8996	#147-8996	147-8996
#147-8996	147-8996	147-8996	#147-8997	#147-8997	147-8997	#147-8997	147-8997	#147-8997	147-8997
#147-8997	147-8997	#147-8997	147-8997	147-8997	#147-8997	147-8997	147-8997	147-8997	147-8997
147-9030	147-9030	147-9030	147-9030	#147-9098	147-9098	#147-9098	147-9098	#147-9099	147-9099
#147-9099	147-9099	#147-9102	147-9102	#148-9137	#148-9137	148-9137	#148-9137	148-9137	#148-9137
148-9137	#148-9137	148-9137	#148-9137	148-9137	148-9137	#148-9138	#148-9138	148-9138	#148-9138
148-9138	#148-9138	148-9138	#148-9138	148-9138	#148-9138	148-9138	148-9138	148-9138	#148-9138
#148-9279	148-9279	#148-9280	148-9280	#148-9280	148-9280	#148-9286	148-9286	#148-9279	148-9279
149-9321	#149-9321	149-9321	#149-9321	149-9321	#149-9321	149-9321	#149-9321	149-9321	#149-9321
#149-9322	#149-9322	149-9322	#149-9322	149-9322	#149-9322	149-9322	#149-9322	149-9322	#149-9322

MACRO CROSS REFERENCE   CREF   03.00   SEQ 0273

```
MACRO NAME    REFERENCES

              149-9322    149-9322   #149-9412    149-9412   #149-9412    149-9412   #149-9417    149-9417   #150-9443   #150-9443
              150-9443   #150-9443    150-9443   #150-9443    150-9443   #150-9443    150-9443   #150-9448    150-9448   #151-9472
              151-9472   #151-9484    151-9484    151-9484    151-9484    151-9484   #151-9486    151-9486    151-9486    151-9486
              151-9486   #151-9488    151-9488    151-9488    151-9488    151-9488    151-9488   #151-9490    151-9490    151-9490
              151-9490    151-9490    151-9490   #151-9492    151-9492    151-9492    151-9492    151-9492    151-9492   #151-9495
              152-9547    152-9547    152-9547   #152-9549    152-9545    152-9545    152-9545   #152-9547    152-9547    152-9547
              153-9584   #153-9584    153-9584   #153-9584    152-9549    152-9549    152-9549   #152-9552    152-9552   #153-9584
M$GNLS       #75-3959     75-3959    #145-8778    145-8778   #145-8791    145-8791   #145-8799    145-8799   #145-8848    145-8848
             #146-8951    146-8951
M$GNTA       #30-1285     30-1285    #31-1309     31-1309    #43-2222     43-2222    #44-2253     44-2253    #45-2285     45-2285
             #46-2351     46-2351    #47-2379     47-2379    #48-2415     48-2415    #49-2476     49-2476    #50-2514     50-2514
             #51-2581     51-2581    #52-2606     52-2606    #53-2649     53-2649    #54-2703     54-2703    #134-7400    134-7400
             #136-7620    136-7620   #137-7652    137-7652   #138-7705    138-7705   #139-7742    139-7742   #140-7775    140-7775
             #141-7874    141-7874   #142-7999    142-7999   #143-8135    143-8135   #144-8650    144-8650   #145-8871    145-8871
             #146-8957    146-8957   #147-9102    147-9102   #148-9286    148-9286   #149-9417    149-9417   #150-9448    150-9448
             #151-9495    151-9495   #152-9552    152-9552
M$GNTE       #141-7792    141-7792   #142-7891    142-7891    143-8017    143-8017   #144-8150    144-8150   #145-8684    145-8684
             #146-8888    146-8888   #147-8974    147-8974   #148-9117    148-9117   #149-9298    149-9298   #150-9429    150-9429
M$HAPT       #28-1231     28-1231
M$HNAP       #28-1231     28-1231
M$INCR       #28-1205     28-1205    #30-1277    #30-1277     30-1277     30-1277    #31-1304    #31-1304     31-1304     31-1304
             #43-2211    #43-2211     43-2211     43-2211    #43-2216    #43-2219    #43-2220    #43-2222    #44-2249    #44-2249
              44-2249     44-2249    #44-2251    #44-2253    #45-2276    #45-2276     45-2276     45-2276    #45-2279    #45-2282
             #45-2285    #46-2314    #46-2314     46-2314     46-2314    #46-2320    #46-2332    #46-2345    #46-2347    #46-2351
             #47-2373    #47-2373     47-2373     47-2373    #47-2375    #47-2376    #47-2377    #47-2379    #48-2403    #48-2403
              48-2403     48-2403    #48-2407    #48-2408    #48-2411    #48-2413    #48-2415    #49-2439    #49-2439     49-2439
              49-2439    #49-2442    #49-2445    #49-2472    #49-2476    #50-2499    #50-2499     50-2499     50-2499    #50-2501
             #50-2508    #50-2514    #51-2539    #51-2539     51-2539     51-2539    #51-2542    #51-2550    #51-2554    #51-2573
             #51-2581    #52-2602    #52-2602     52-2602     52-2602    #52-2604    #52-2606    #53-2635    #53-2635     53-2635
              53-2635    #53-2638    #53-2643    #53-2647    #53-2649    #54-2671    #54-2671     54-2671     54-2671    #54-2674
             #54-2695    #54-2699    #54-2703    #62-3263    #75-3959    #75-3959     75-3959    #75-3978    #86-4593    #86-4626
             #86-4634    #88-4766    #88-4783    #89-4828    #89-4830    #89-4831    #90-4888    #91-4941    #91-4960    #92-4999
             #94-5120    #94-5134    #96-5270    #97-5359    #98-5415    #99-5462    #100-5531   #101-5585   #102-5631   #128-7051
             #134-7394   #134-7394    134-7394    134-7394   #134-7400   #135-7417   #135-7417    135-7417    135-7417   #136-7460
             #136-7460    136-7460    136-7460   #136-7462   #136-7465   #136-7468   #136-7471   #136-7475   #136-7479   #136-7489
             #136-7536   #136-7555   #136-7604   #136-7620   #137-7643   #137-7643    137-7643    137-7643   #137-7652   #138-7670
             #138-7670    138-7670    138-7670   #138-7683   #138-7689   #138-7705   #139-7723   #139-7723    139-7723    139-7723
             #139-7731   #139-7742   #140-7761   #140-7761    140-7761    140-7761   #140-7775   #141-7792   #141-7792    141-7792
             #141-7792    141-7792    141-7792   #141-7862   #141-7864   #141-7866   #141-7874   #142-7891   #142-7891    142-7891
             #142-7891    142-7891    142-7891   #142-7916   #142-7990   #142-7999   #143-8017   #143-8017    143-8017   #143-8017
              143-8017    143-8017   #143-8042   #143-8125   #143-8135   #144-8150   #144-8150    144-8150   #144-8150    144-8150
              144-8150   #144-8321   #144-8615   #144-8619   #144-8626   #144-8650   #145-8684   #145-8684    145-8684   #145-8684
              145-8684    145-8684   #145-8694   #145-8695   #145-8697   #145-8717   #145-8718   #145-8728   #145-8778   #145-8778
              145-8778   #145-8791   #145-8791    145-8791   #145-8799   #145-8799    145-8799   #145-8831   #145-8848   #145-8848
              145-8848   #145-8864   #145-8865   #145-8871   #146-8888   #146-8888    146-8888   #146-8888    146-8888    146-8888
             #146-8898   #146-8899   #146-8901   #146-8919   #146-8951   #146-8951    146-8951   #146-8957   #147-8974   #147-8974
              147-8974   #147-8974    147-8974    147-8974   #147-9098   #147-9099   #147-9102   #148-9117   #148-9117   #148-9117
             #148-9117    148-9117    148-9117   #148-9137   #148-9138   #148-9279   #148-9280   #148-9286   #149-9298   #149-9298
              149-9298   #149-9298    149-9298    149-9298   #149-9321   #149-9322   #149-9412   #149-9417   #150-9429   #150-9429
              150-9429   #150-9429    150-9429    150-9429   #150-9443   #150-9448   #151-9472   #151-9472
              151-9472    151-9472   #152-9534   #152-9534    152-9534    152-9534
```

MACRO CROSS REFERENCE   CREF   03.00

MACRO NAME   REFERENCES

M$LDRO	#136-7462	136-7462	#136-7465	136-7465	#136-7468	136-7468	#136-7471	136-7471	#136-7479	136-7479
	#136-7555	136-7555	#141-7864	141-7864	#142-7990	142-7990	#143-8125	143-8125	#145-8864	145-8864
	#145-8865	145-8865	#147-9098	147-9098	#147-9099	147-9099	#148-9279	148-9279	#148-9280	148-9280
	#149-9412	149-9412								
M$MCHI	#28-1183	28-1183								
M$MCLO	#28-1183	28-1183								
M$POP	#30-1285	30-1285	#31-1309	31-1309	#43-2222	43-2222	#44-2253	44-2253	#45-2285	45-2285
	#46-2351	46-2351	#47-2379	47-2379	#48-2415	48-2415	#49-2476	49-2476	#50-2514	50-2514
	#51-2581	51-2581	#52-2606	52-2606	#53-2649	53-2649	#54-2703	54-2703	#134-7400	134-7400
	#135-7423	135-7423	#136-7620	136-7620	#137-7652	137-7652	#138-7705	138-7705	#139-7742	139-7742
	#140-7775	140-7775	#141-7874	141-7874	#142-7999	142-7999	#143-8135	143-8135	#144-8650	144-8650
	#145-8871	145-8871	#146-8957	146-8957	#147-9102	147-9102	#148-9286	148-9286	#149-9417	149-9417
	#150-9448	150-9448	#151-9495	151-9495	#152-9552	152-9552	#153-9585	153-9585		
M$PRIN	#43-2216	43-2216	#43-2219	43-2219	#43-2220	43-2220	#44-2251	44-2251	#45-2279	45-2279
	#45-2282	45-2282	#46-2320	46-2320	#46-2332	46-2332	#46-2345	46-2345	#46-2347	46-2347
	#47-2375	47-2375	#47-2376	47-2376	#47-2377	47-2377	#48-2407	48-2407	#48-2408	48-2408
	#48-2411	48-2411	#48-2413	48-2413	#49-2442	49-2442	#49-2445	49-2445	#49-2472	49-2472
	#50-2501	50-2501	#50-2508	50-2508	#51-2542	51-2542	#51-2550	51-2550	#51-2554	51-2554
	#51-2573	51-2573	#52-2604	52-2604	#53-2638	53-2638	#53-2643	53-2643	#53-2647	53-2647
	#54-2674	54-2674	#54-2695	54-2695	#54-2699	54-2699	#75-3978	75-3978	#86-4593	86-4593
	#86-4626	86-4626	#86-4634	86-4634	#89-4830	89-4830	#92-4999	92-4999	#136-7604	136-7604
	#139-7731	139-7731	#144-8615	144-8615	#145-8728	145-8728	#145-8831	145-8831	#146-8919	146-8919
M$PUSH	#28-1205	28-1205	#30-1277	30-1277	#31-1304	31-1304	#43-2211	43-2211	#44-2249	44-2249
	#45-2276	45-2276	#46-2314	46-2314	#47-2373	47-2373	#48-2403	48-2403	#49-2439	49-2439
	#50-2499	50-2499	#51-2539	51-2539	#52-2602	52-2602	#53-2635	53-2635	#54-2671	54-2671
	#134-7394	134-7394	#135-7417	135-7417	#136-7460	136-7460	#137-7643	137-7643	#138-7670	138-7670
	#139-7723	139-7723	#140-7761	140-7761	#141-7792	141-7792	#142-7891	142-7891	#143-8017	143-8017
	#144-8150	144-8150	#145-8684	145-8684	#146-8888	146-8888	#147-8974	147-8974	#148-9117	148-9117
	#149-9298	149-9298	#150-9429	150-9429	#151-9472	151-9472	#152-9534	152-9534		
M$PUT	#43-2216	43-2216	43-2216	#43-2219	43-2219	43-2219	#43-2220	43-2220	43-2220	#44-2251
	44-2251	44-2251	44-2251	#45-2279	45-2279	45-2279	45-2279	#45-2282	45-2282	45-2282
	45-2282	#46-2320	46-2320	46-2320	46-2320	46-2320	#46-2332	46-2332	46-2332	46-2332
	46-2332	#46-2345	46-2345	46-2345	46-2345	46-2345	#46-2347	46-2347	46-2347	46-2347
	#47-2375	47-2375	47-2375	47-2375	#47-2376	47-2376	47-2376	47-2376	#47-2377	47-2377
	47-2377	47-2377	#48-2407	48-2407	48-2407	48-2407	48-2407	#48-2408	48-2408	48-2408
	48-2408	48-2408	#48-2411	48-2411	48-2411	48-2411	48-2411	#48-2413	48-2413	48-2413
	48-2413	#49-2442	49-2442	49-2442	49-2442	#49-2445	49-2445	49-2445	49-2445	#49-2472
	49-2472	49-2472	49-2472	#50-2501	50-2501	50-2501	50-2501	#50-2508	50-2508	50-2508
	50-2508	50-2508	#51-2542	51-2542	51-2542	51-2542	#51-2550	51-2550	51-2550	#51-2554
	51-2554	51-2554	51-2554	#51-2573	51-2573	51-2573	51-2573	51-2573	51-2573	51-2573
	51-2573	#52-2604	52-2604	52-2604	52-2604	52-2604	#53-2638	53-2638	53-2638	53-2638
	#53-2643	53-2643	53-2643	53-2643	#53-2647	53-2647	53-2647	#54-2674	54-2674	#51-2554
	54-2674	#54-2695	54-2695	54-2695	#54-2699	54-2699	54-2699	54-2699	54-2699	54-2699
	#75-3978	75-3978	75-3978	75-3978	#86-4593	86-4593	86-4593	#86-4626	86-4626	86-4626
	86-4626	86-4626	86-4626	86-4626	86-4626	#86-4634	86-4634	86-4634	#89-4830	89-4830
	89-4830	#92-4999	92-4999	92-4999	92-4999	92-4999	#136-7489	136-7489	136-7489	136-7489
	136-7489	#136-7604	136-7604	136-7604	136-7604	#139-7731	139-7731	139-7731	139-7731	#142-7916
	142-7916	142-7916	142-7916	142-7916	#143-8042	143-8042	143-8042	143-8042	#144-8615	144-8615
	144-8615	144-8615	144-8615	#145-8717	145-8717	145-8717	145-8717	145-8717	#145-8718	145-8718
	145-8718	145-8718	145-8718	#145-8728	145-8728	145-8728	145-8728	#145-8831	145-8831	145-8831
	145-8831	#146-8919	146-8919	146-8919	146-8919	#147-8996	147-8996	147-8996	147-8996	147-8996
	147-8996	#147-8997	147-8997	147-8997	147-8997	147-8997	#148-9137	148-9137	148-9137	148-9137

MACRO CROSS REFERENCE   CREF   03.00   SEQ 0275

MACRO NAME	REFERENCES									
	148-9137	#148-9138	148-9138	148-9138	148-9138	148-9138	#149-9321	149-9321	149-9321	149-9321
	149-9321	#149-9322	149-9322	149-9322	149-9322	149-9322				
M$PUT1	#43-2216	#43-2216	43-2216	43-2216	#43-2219	#43-2219	43-2219	43-2219	#43-2220	#43-2220
	43-2220	43-2220	#44-2251	#44-2251	#44-2251	44-2251	44-2251	44-2251	#45-2279	#45-2279
	#45-2279	45-2279	45-2279	45-2279	#45-2282	#45-2282	#45-2282	45-2282	45-2282	45-2282
	#46-2320	#46-2320	#46-2320	#46-2320	46-2320	46-2320	46-2320	46-2320	#46-2332	#46-2332
	#46-2332	#46-2332	46-2332	46-2332	46-2332	46-2332	#46-2345	#46-2345	#46-2345	#46-2345
	46-2345	46-2345	46-2345	46-2345	#46-2347	#46-2347	#46-2347	46-2347	46-2347	46-2347
	#47-2375	#47-2375	#47-2375	47-2375	47-2375	47-2375	#47-2376	#47-2376	#47-2376	47-2376
	47-2376	47-2376	#47-2377	#47-2377	#47-2377	47-2377	47-2377	47-2377	#48-2407	#48-2407
	#48-2407	#48-2407	48-2407	48-2407	48-2407	48-2407	#48-2408	#48-2408	#48-2408	#48-2408
	48-2408	48-2408	48-2408	48-2408	#48-2411	#48-2411	#48-2411	48-2411	48-2411	48-2411
	48-2411	48-2411	#48-2413	#48-2413	#48-2413	48-2413	48-2413	48-2413	#49-2442	#49-2442
	#49-2442	49-2442	49-2442	49-2442	#49-2445	#49-2445	#49-2445	49-2445	49-2445	49-2445
	#49-2472	#49-2472	#49-2472	49-2472	49-2472	49-2472	#50-2501	#50-2501	#50-2501	50-2501
	50-2501	50-2501	#50-2508	#50-2508	#50-2508	#50-2508	50-2508	50-2508	50-2508	50-2508
	#51-2542	#51-2542	#51-2542	51-2542	51-2542	51-2542	#51-2550	#51-2550	51-2550	51-2550
	#51-2554	#51-2554	#51-2554	51-2554	51-2554	51-2554	#51-2573	#51-2573	51-2573	51-2573
	#51-2573	#51-2573	#51-2573	51-2573	51-2573	51-2573	51-2573	51-2573	51-2573	51-2573
	#52-2604	#52-2604	#52-2604	#52-2604	52-2604	52-2604	52-2604	52-2604	#53-2638	#53-2638
	#53-2638	53-2638	53-2638	53-2638	#53-2643	#53-2643	#53-2643	53-2643	53-2643	53-2643
	#53-2647	#53-2647	53-2647	53-2647	#54-2674	#54-2674	#54-2674	54-2674	54-2674	54-2674
	#54-2695	#54-2695	54-2695	54-2695	#54-2699	#54-2699	#54-2699	54-2699	#54-2699	54-2699
	54-2699	54-2699	54-2699	54-2699	#75-3978	#75-3978	#75-3978	75-3978	75-3978	75-3978
	#86-4593	#86-4593	86-4593	86-4593	#86-4626	#86-4626	#86-4626	#86-4626	#86-4626	#86-4626
	#86-4626	86-4626	86-4626	86-4626	86-4626	86-4626	86-4626	86-4626	#86-4634	#86-4634
	86-4634	86-4634	#89-4830	#89-4830	89-4830	89-4830	#92-4999	#92-4999	#92-4999	#92-4999
	92-4999	92-4999	92-4999	92-4999	#136-7489	#136-7489	#136-7489	#136-7489	136-7489	136-7489
	136-7489	136-7489	#136-7604	#136-7604	#136-7604	136-7604	136-7604	136-7604	#139-7731	#139-7731
	#139-7731	139-7731	139-7731	139-7731	#142-7916	#142-7916	#142-7916	#142-7916	142-7916	142-7916
	142-7916	142-7916	#143-8042	#143-8042	#143-8042	#143-8042	143-8042	143-8042	143-8042	143-8042
	#144-8615	#144-8615	#144-8615	#144-8615	144-8615	144-8615	144-8615	144-8615	#145-8717	#145-8717
	#145-8717	#145-8717	145-8717	145-8717	145-8717	145-8717	#145-8718	#145-8718	#145-8718	#145-8718
	145-8718	145-8718	145-8718	145-8718	#145-8728	#145-8728	#145-8728	145-8728	145-8728	145-8728
	#145-8831	#145-8831	#145-8831	145-8831	145-8831	145-8831	#146-8919	#146-8919	#146-8919	146-8919
	146-8919	146-8919	#147-8996	#147-8996	#147-8996	#147-8996	147-8996	147-8996	147-8996	147-8996
	#147-8997	#147-8997	#147-8997	#147-8997	147-8997	147-8997	147-8997	147-8997	#148-9137	#148-9137
	#148-9137	#148-9137	148-9137	148-9137	148-9137	148-9137	#148-9138	#148-9138	#148-9138	#148-9138
	148-9138	148-9138	148-9138	148-9138	#149-9321	#149-9321	#149-9321	149-9321	149-9321	149-9321
	149-9321	149-9321	#149-9322	#149-9322	#149-9322	#149-9322	149-9322	149-9322	149-9322	149-9322
M$RADI	#75-3959	75-3959	#145-8778	145-8778	#145-8791	145-8791	#145-8799	145-8799	#145-8848	145-8848
	#146-8951	146-8951	#151-9484	151-9484	#151-9486	151-9486	#151-9488	151-9488	#151-9490	151-9490
	#151-9492	151-9492	#152-9545	152-9545	#152-9547	152-9547	#152-9549	152-9549		
M$RNRO	#136-7479	136-7479	#136-7555	136-7555	#144-8321	144-8321				
M$SETS	#28-1205	28-1205	#30-1277	30-1277	#31-1304	31-1304	#43-2211	43-2211	#44-2249	44-2249
	#45-2276	45-2276	#46-2314	46-2314	#47-2373	47-2373	#48-2403	48-2403	#49-2439	49-2439
	#50-2499	50-2499	#51-2539	51-2539	#52-2602	52-2602	#53-2635	53-2635	#54-2671	54-2671
	#134-7394	134-7394	#135-7417	135-7417	#136-7460	136-7460	#137-7643	137-7643	#138-7670	138-7670
	#139-7723	139-7723	#140-7761	140-7761	#141-7792	141-7792	#142-7891	142-7891	#143-8017	143-8017
	#144-8150	144-8150	#145-8684	145-8684	#146-8888	146-8888	#147-8974	147-8974	#148-9117	148-9117
	#149-9298	149-9298	#150-9429	150-9429	#151-9472	151-9472	#152-9534	152-9534		
M$SVC	#43-2216	43-2216	#43-2219	43-2219	#43-2220	43-2220	#43-2222	43-2222	#44-2251	44-2251

MACRO CROSS REFERENCE   CREF   03.00

MACRO NAME   REFERENCES

@44-2253	44-2253	@45-2279	45-2279	@45-2282	45-2282	@45-2285	45-2285	@46-2320	46-2320
@46-2332	46-2332	@46-2345	46-2345	@46-2347	46-2347	@46-2351	46-2351	@47-2375	47-2375
@47-2376	47-2376	@47-2377	47-2377	@47-2379	47-2379	@48-2407	48-2407	@48-2408	48-2408
@48-2411	48-2411	@48-2413	48-2413	@48-2415	48-2415	@49-2442	49-2442	@49-2445	49-2445
@49-2472	49-2472	@49-2476	49-2476	@50-2501	50-2501	@50-2508	50-2508	@50-2514	50-2514
@51-2542	51-2542	@51-2550	51-2550	@51-2554	51-2554	@51-2573	51-2573	@51-2581	51-2581
@52-2604	52-2604	@52-2606	52-2606	@53-2638	53-2638	@53-2643	53-2643	@53-2647	53-2647
@53-2649	53-2649	@54-2674	54-2674	@54-2695	54-2695	@54-2699	54-2699	@54-2703	54-2703
@62-3263	62-3263	@75-3959	75-3959	@75-3978	75-3978	@86-4593	86-4593	@86-4626	86-4626
@86-4634	86-4634	@88-4766	88-4766	@88-4783	88-4783	89-4828	@89-4830	89-4830	@89-4831
89-4831	@90-4888	90-4888	@91-4941	91-4941	@91-4960	91-4960	@92-4999	92-4999	@94-5120
94-5120	@94-5134	94-5134	@96-5270	96-5270	@97-5359	97-5359	@98-5415	98-5415	@99-5462
99-5462	@100-5531	100-5531	@101-5585	101-5585	@102-5631	102-5631	@128-7051	128-7051	@134-7396
@134-7400	134-7400	@136-7462	136-7462	@136-7465	136-7465	@136-7468	136-7468	@136-7471	136-7471
@136-7475	136-7475	@136-7479	136-7479	@136-7489	136-7489	@136-7536	136-7536	@136-7555	136-7555
@136-7604	136-7604	@136-7620	136-7620	@137-7652	137-7652	@138-7683	138-7683	@138-7689	138-7689
@138-7705	138-7705	@139-7731	139-7731	@139-7739	@139-7742	139-7742	@140-7770	@140-7775	140-7775
141-7862	@141-7864	141-7864	@141-7866	141-7866	@141-7874	141-7874	@142-7916	142-7916	@142-7990
142-7990	@142-7999	142-7999	@143-8042	143-8042	@143-8125	143-8125	@143-8135	143-8135	@144-8321
144-8321	@144-8615	144-8615	@144-8619	144-8619	@144-8626	144-8626	@144-8650	144-8650	@145-8694
145-8694	@145-8695	145-8695	@145-8697	145-8697	@145-8717	145-8717	@145-8718	145-8718	@145-8728
145-8728	@145-8778	145-8778	@145-8791	145-8791	@145-8799	145-8799	@145-8831	145-8831	@145-8848
145-8848	@145-8864	145-8864	@145-8865	145-8865	@145-8871	145-8871	@146-8898	146-8898	@146-8899
146-8899	@146-8901	146-8901	@146-8919	146-8919	@146-8951	146-8951	@146-8957	146-8957	@147-8996
147-8996	@147-8997	147-8997	@147-9098	147-9098	@147-9099	147-9099	@147-9102	147-9102	@148-9137
148-9137	@148-9138	148-9138	@148-9279	148-9279	@148-9280	148-9280	@148-9286	148-9286	@149-9321
149-9321	@149-9322	149-9322	@149-9412	149-9412	@149-9417	149-9417	150-9443	@150-9448	150-9448

MACRO NAME									
M$TLAB	@43-2216	@43-2219	@43-2220	@43-2222	@44-2251	@44-2253	@45-2279	@45-2282	@45-2285	@46-2320
	@46-2332	@46-2345	@46-2347	@46-2351	@47-2375	@47-2376	@47-2377	@47-2379	@48-2407	@48-2408
	@48-2411	@48-2413	@48-2415	@49-2442	@49-2445	@49-2472	@49-2476	@50-2501	@50-2508	@50-2514
	@51-2542	@51-2550	@51-2554	@51-2573	@51-2581	@52-2604	@52-2606	@53-2638	@53-2643	@53-2647
	@53-2649	@54-2674	@54-2695	@54-2699	@54-2703	@62-3263	@75-3959	@75-3978	@86-4593	@86-4626
	@86-4634	@88-4766	@88-4783	@89-4828	@89-4830	@89-4831	@90-4888	@91-4941	@91-4960	@92-4999
	@94-5120	@94-5134	@96-5270	@97-5359	@98-5415	@99-5462	@100-5531	@101-5585	@102-5631	@128-7051
	@134-7400	@136-7462	@136-7465	@136-7468	@136-7471	@136-7475	@136-7479	@136-7489	@136-7536	@136-7555
	@136-7604	@136-7620	@137-7652	@138-7683	@138-7689	@138-7705	@139-7731	@139-7742	@140-7775	@141-7862
	@141-7864	@141-7866	@142-7916	@142-7990	@142-7999	@143-8042	@143-8125	@143-8135	@144-8321
	@144-8615	@144-8619	@144-8626	@144-8650	@145-8694	@145-8695	@145-8697	@145-8717	@145-8718	@145-8728
	@145-8778	@145-8791	@145-8799	@145-8831	@145-8848	@145-8864	@145-8865	@145-8871	@146-8898	@146-8899
	@146-8901	@146-8919	@146-8951	@146-8957	@147-8996	@147-8997	@147-9098	@147-9099	@147-9102	@148-9137
	@148-9138	@148-9279	@148-9280	@148-9286	@149-9321	@149-9322	@149-9412	@149-9417	@150-9443	@150-9448
M$TSTL	@43-2216	43-2216	@43-2219	43-2219	@43-2220	43-2220	@43-2222	43-2222	@44-2251	44-2251
	@44-2253	44-2253	@45-2279	45-2279	@45-2282	45-2282	@45-2285	45-2285	@46-2320	46-2320
	@46-2332	46-2332	@46-2345	46-2345	@46-2347	46-2347	@46-2351	46-2351	@47-2375	47-2375
	@47-2376	47-2376	@47-2377	47-2377	@47-2379	47-2379	@48-2407	48-2407	@48-2408	48-2408
	@48-2411	48-2411	@48-2413	48-2413	@48-2415	48-2415	@49-2442	49-2442	@49-2445	49-2445
	@49-2472	49-2472	@49-2476	49-2476	@50-2501	50-2501	@50-2508	50-2508	@50-2514	50-2514
	@51-2542	51-2542	@51-2550	51-2550	@51-2554	51-2554	@51-2573	51-2573	@51-2581	51-2581
	@52-2604	52-2604	@52-2606	52-2606	@53-2638	53-2638	@53-2643	53-2643	@53-2647	53-2647
	@53-2649	53-2649	@54-2674	54-2674	@54-2695	54-2695	@54-2699	54-2699	@54-2703	54-2703
	@62-3263	62-3263	@75-3959	75-3959	@75-3978	75-3978	@86-4593	86-4593	@86-4626	86-4626
	@86-4634	86-4634	@88-4766	88-4766	@88-4783	88-4783	@89-4828	89-4828	@89-4830

MACRO CROSS REFERENCE   CREF   03.00

MACRO NAME	REFERENCES								
	89-4830	#89-4831	89-4831	#90-4888	90-4888	#91-4941	91-4941	#91-4960	91-4960	#92-4999
	92-4999	#94-5120	94-5120	#94-5134	94-5134	#96-5270	96-5270	#97-5359	97-5359	#98-5415
	98-5415	#99-5462	99-5462	#100-5531	100-5531	#101-5585	101-5585	#102-5631	102-5631	#128-7051
	128-7051	#134-7400	134-7400	#136-7462	136-7462	#136-7465	136-7465	#136-7468	136-7468	#136-7471
	136-7471	#136-7475	136-7475	#136-7479	136-7479	#136-7489	136-7489	#136-7536	136-7536	#136-7555
	136-7555	#136-7604	136-7604	#136-7620	136-7620	#137-7652	137-7652	#138-7683	138-7683	#138-7689
	138-7689	#138-7705	138-7705	#139-7731	139-7731	#139-7742	139-7742	#140-7775	140-7775	#141-7862
	#141-7862	141-7862	#141-7864	141-7864	#141-7866	141-7866	#141-7874	141-7874	#142-7916	142-7916
	#142-7990	142-7990	#142-7999	142-7999	#143-8042	143-8042	#143-8125	143-8125	#143-8135	143-8135
	#144-8321	144-8321	#144-8615	144-8615	#144-8619	144-8619	#144-8626	144-8626	#144-8650	144-8650
	#145-8694	145-8694	#145-8695	145-8695	#145-8697	145-8697	#145-8717	145-8717	#145-8718	145-8718
	#145-8728	145-8728	#145-8778	145-8778	#145-8791	145-8791	#145-8799	145-8799	#145-8831	145-8831
	#145-8848	145-8848	#145-8864	145-8864	#145-8865	145-8865	#145-8871	145-8871	#146-8898	146-8898
	#146-8899	146-8899	#146-8901	146-8901	#146-8919	146-8919	#146-8951	146-8951	#146-8957	146-8957
	#147-8996	147-8996	#147-8997	147-8997	#147-9098	147-9098	#147-9099	147-9099	#147-9102	147-9102
	#148-9137	148-9137	#148-9138	148-9138	#148-9279	148-9279	#148-9280	148-9280	#148-9286	148-9286
	#149-9321	149-9321	#149-9322	149-9322	#149-9412	149-9412	#149-9417	149-9417	#150-9443	#150-9443
	150-9443	#150-9448	150-9448							
M$WORD	#28-1231	28-1231	#29-1251	29-1251	29-1251	29-1251	29-1251	29-1251	29-1251	29-1251
	29-1251	29-1251	29-1251	29-1251	#75-3959	75-3959	#75-3959	75-3959	#89-4828	89-4828
	89-4828	89-4828	#134-7396	134-7396	#138-7689	#139-7739	139-7739	#140-7770	140-7770	#141-7862
	141-7862	141-7862	141-7862	#144-8626	#145-8694	#145-8697	#145-8778	145-8778	#145-8778	145-8778
	#145-8791	145-8791	#145-8791	145-8791	#145-8799	145-8799	#145-8799	145-8799	#145-8848	145-8848
	#145-8848	145-8848	#146-8898	#146-8901	#146-8951	146-8951	#146-8951	146-8951	#150-9443	150-9443
	150-9443	150-9443	#151-9484	151-9484	#151-9486	151-9486	#151-9488	151-9488	#151-9490	151-9490
	#151-9492	151-9492	#152-9545	152-9545	#152-9547	152-9547	#152-9549	152-9549	153-9584	153-9584
PASS	#37-1922	43-2221	45-2284	46-2350	51-2580	53-2648	54-2701	56-2777	57-2884	58-2955
	59-3027	60-3064	61-3203	62-3288	63-3361	64-3394	65-3431	66-3475	67-3500	68-3537
	69-3567	70-3603	71-3677	72-3759	73-3799	74-3927	75-3982	76-4027	77-4095	78-4156
	79-4194	80-4252	81-4329	82-4360	83-4431	84-4512	85-4562	86-4627	86-4636	87-4680
	88-4802	89-4833	90-4891	91-4965	92-5000	93-5049	94-5141	95-5176	96-5324	97-5361
	98-5422	99-5464	100-5541	101-5588	102-5634	103-5686	104-5739	107-5829	108-5871	109-6009
	110-6035	112-6156	113-6210	114-6264	115-6344	118-6537	119-6568	120-6595	121-6643	122-6738
	123-6800	124-6891	125-6936	126-6976	127-7015	132-7273	133-7375			
POINTE	28-1212									
PRINTB	43-2216	43-2219	44-2251	45-2279	45-2282	46-2320	47-2375	48-2407	49-2442	50-2501
	51-2542	52-2604	53-2638	53-2643	53-2647	54-2674	144-8615			
PRINTF	75-3978	86-4593	86-4626	86-4634	89-4830	136-7604	139-7731	145-8728	146-8919	
PRINTX	43-2220	46-2332	46-2345	46-2347	47-2376	47-2377	48-2408	48-2411	48-2413	49-2445
	49-2472	50-2508	51-2550	51-2554	51-2573	54-2695	54-2699	92-4999	145-8831	
READEF	136-7462	136-7465	136-7468	136-7471						
SAVE	#36-1880	43-2212	45-2277	46-2315	51-2540	53-2636	54-2672	56-2748	57-2808	58-2919
	59-2988	60-3053	61-3102	62-3238	63-3320	64-3387	65-3423	66-3462	67-3495	68-3523
	69-3559	70-3589	71-3648	72-3712	73-3781	74-3845	75-3953	76-4011	77-4062	78-4128
	79-4178	80-4230	81-4286	82-4356	83-4393	84-4472	85-4554	86-4588	86-4608	87-4661
	88-4724	89-4826	90-4866	91-4921	92-4991	93-5022	94-5081	95-5167	96-5212	97-5348
	98-5386	99-5451	100-5492	101-5570	102-5615	103-5660	104-5713	107-5818	108-5852	109-5923
	110-6031	112-6116	113-6183	114-6237	115-6300	118-6455	119-6562	120-6591	121-6630	122-6669
	123-6772	124-6838	125-6922	126-6963	127-7002	132-7213	133-7310			
SETVEC	136-7489	142-7916	143-8042	145-8717	145-8718	147-8996	147-8997	148-9137	148-9138	149-9321
	149-9322									
SVC	#28-1182	28-1183								

MACRO CROSS REFERENCE   CREF   03.00

MACRO NAME   REFERENCES

XFER   #134-7396 #138-7689 #139-7739 #140-7770 #144-8626 #145-8694 #145-8697 #146-8898 #146-8901