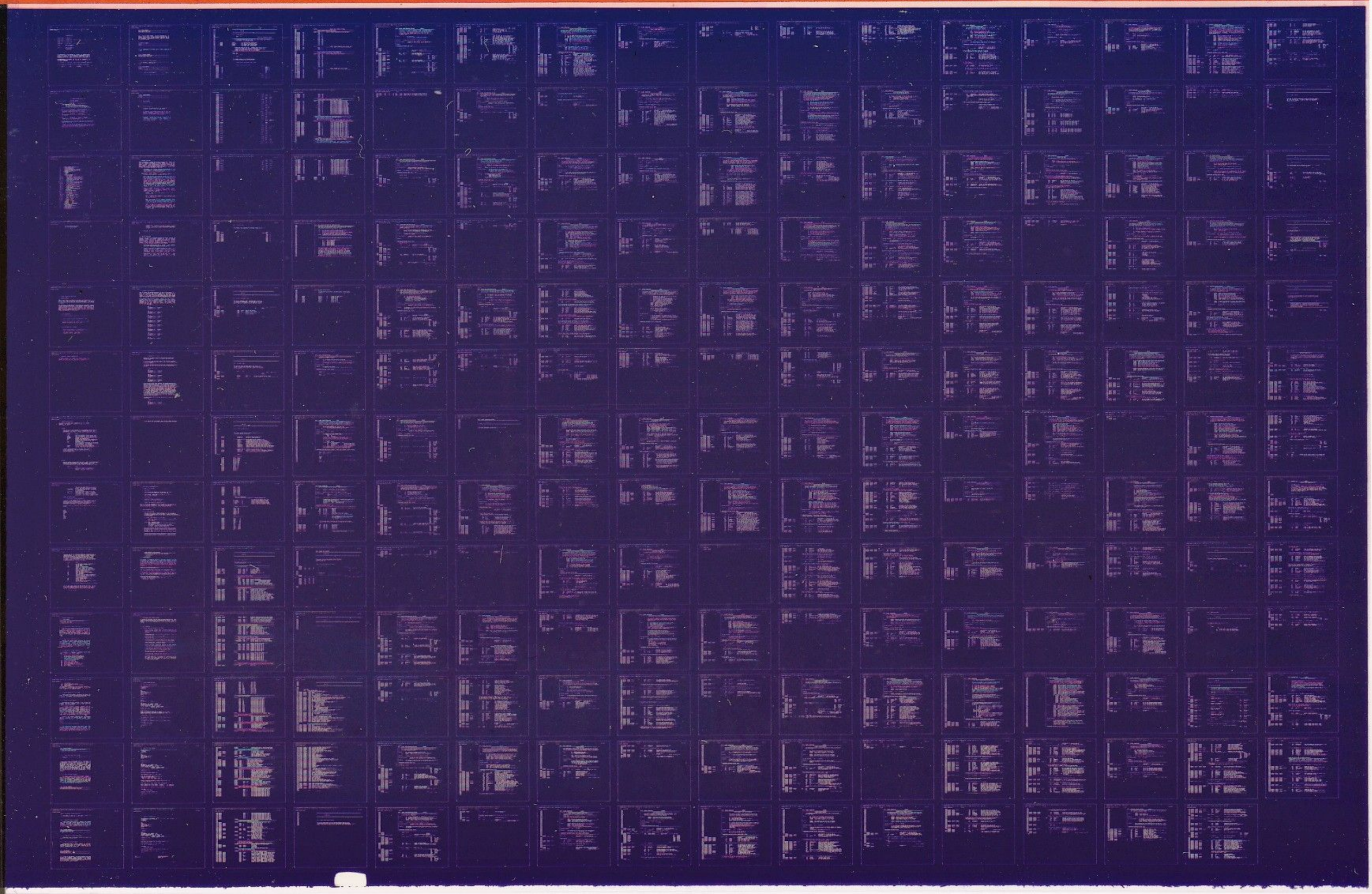
DHV11-M

DHV11-M FUNC TST PART 3
CVDHCD0

AH-T657D-MC
1 OF 2 OCT 1985

digital
MADE IN USA

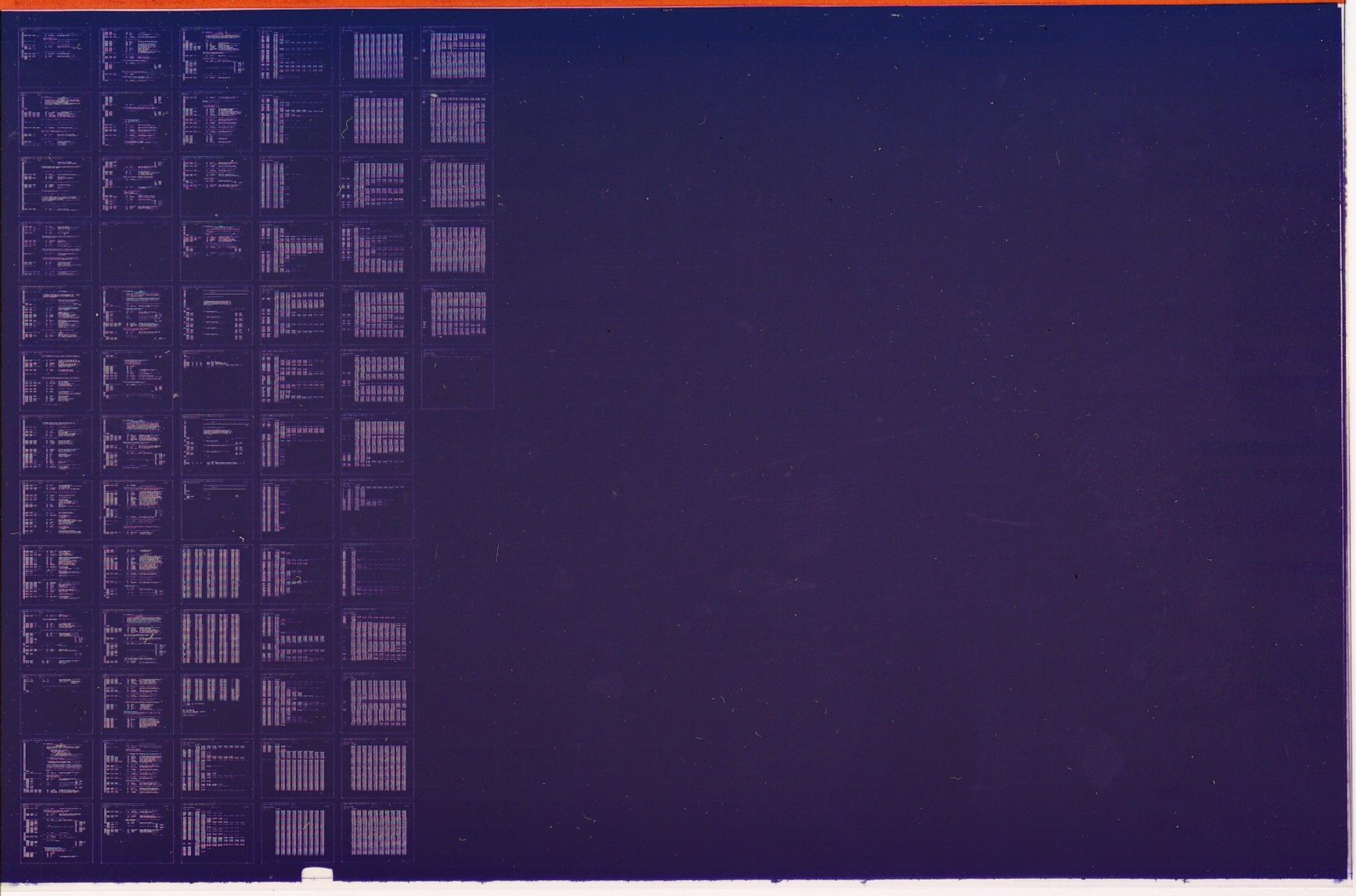


DHV11-M

DHV11-M FUNC TST PART 3
CVDHCD0

AH-T657D-MC
2 OF 2 OCT 1985

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 pr'ority 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:

- LSI-11 processor with at least 32 Kbytes of RAM.
- DHV11-M boards installed on the Q-bus.
- Appropriate program load device supporting XXDP+ media or a down-line loading system.

1.3 RELATED DOCUMENTS AND STANDARDS

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

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

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

   - 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.

   - 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 0 THROUGH 7. THERE IS ONE HARDWARE PARAMETER THAT CAN VARY AMONG UNITS CALLED THE Q-FACTOR. THIS Q-FACTOR MAY BE 0 OR 1. BELOW IS A SIMPLE WAY TO BUILD A TABLE FOR ONE XY11 WITH EIGHT UNITS.

```
# UNITS (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 (0) 1 ?  <CR>
```

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

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

THE RUNTIME SEPVICES 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) 0 ?  1,0<CR>

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

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

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

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

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

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

AS YOU CAN SEE FROM THIS EXAMPLE, NULL REPLIES (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 A'D 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 ' 'ME
TYPE = ERROR TYPE (SYS FATAL, DEV FATAL, HARD OR SOFT)
NUMBER = ERROR NUMBER
UNIT NUMBER = 0 - N (N IS LAST UNIT IN PTABLE)
TST NUMBER = TEST AND SUBTEST WHERE ERROR OCCURRED
PC:XXXXXX = ADDRESS OF ERROR MESSAGE CALL

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

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

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

3.2 ERROR MESSAGES

This program is intended to provide a go/no-go indication of the functionality of 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 mo · 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 (0).
    ACTUAL DATA:   000023 (0).
```

4.0 PERFORMANCE AND PROGRESS REPORTS

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

5.0 TEST SUMMARIES

The following tests are included within 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-0
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 ADDRES. 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-0
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:  (C) 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
    0 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 CVDHCC
CVDHCC0.BIC

DRS
CVDHC-C-0
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 LOOP9ACK) 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
```

```
                        ;******************************************************************************
                        ;
                        ;               VDHC.PHD
                        ;
                        ;******************************************************************************

                        .SBTTL  Program Header

                                .MCALL  SVC
000000                          SVC                                     ; INITIALIZE SUPERVISOR MACROS

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

000001                  SVCINS= 1       ; LIST INSTRUCTIONS, SHIFTED RIGHT
000001                  SVCTST= 1       ; LIST TEST TAGS, SHIFTED RIGHT
000001                  SVCSUB= 1       ; LIST SUBTEST TAGS, SHIFTED RIGHT
000001                  SVCGBL= 1       ; LIST GLOBAL TAGS, SHIFTED RIGHT
000001                  SVCTAG= 1       ; LIST OTHER TAGS, SHIFTED RIGHT

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

000000                  .ENABL  ABS
                        ;.ENABL AMA
002000                  .       =       2000

002000                          BGNMOD

                        ;++
                        ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
                        ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
                        ;

002000                          POINTER BGNRPT,BGNSW,BGNSFT,BGNDU,ERRTBL


002000                          HEADER  CVDHC,D,0,200,0,PRIO7
002000
002000  103                                                     L$NAME::
002001  126                                                             .ASCII /C/
002002  104                                                             .ASCII /V/
002003  110                                                             .ASCII /D/
002004  103                                                             .ASCII /H/
002005  000                                                             .ASCII /C/
002006  000                                                             .BYTE   0
002007  000                                                             .BYTE   0
002010                                                                  .BYTE   0
002010  104                                                     L$REV::
002011                                                                  .ASCII  /D/
002011  060                                                     L$DEPO::
002012                                                                  .ASCII  /0/
002012  000000                                                  L$UNIT::
                                                                        .WORD   0
```

Program Header

SEQ 0028

```
002014
002014  000200
002016
002016  037354
002020
002020  037720
002022
002022  002152
002024
002024  002164
002026
002026  040246
002030
002030  000000
002032
002032  000000
002034
002034  000000
002036
002036  000000
002040
002040  002124
002042
002042  000340
002044
002044  000000
002046
002046  000000
002050
002050     004
002051     000
002052
002052  000000
002054  000000
002056
002056  000000
002060
002060  005366
002062
002062  030114
002064
002064  000000
002066
002066  000000
002070
002070  000000
002072
002072  031024
002074
002074  000000
002076
002076  005376
002100
002100  104035
002102
002102  005316
002104
```

```
L$TIML::
        .WORD   200
L$HPCP::
        .WORD   L$HARD
L$SPCP::
        .WORD   L$SOFT
L$HPTP::
        .WORD   L$HW
L$SPTP::
        .WORD   L$SW
L$LADP::
        .WORD   L$LAST
L$STA::
        .WORD   0
L$CO::
        .WORD   0
L$DTYP::
        .WORD   0
L$APT::
        .WORD   0
L$DTP::
        .WORD   L$DISPATCH
L$PRIO::
        .WORD   PRIO7
L$ENVI::
        .WORD   0
L$EXP1::
        .WORD   0
L$MREV::
        .BYTE   C$REVISION
        .BYTE   C$EDIT
L$EF::
        .WORD   0
        .WORD   0
L$SPC::
        .WORD   0
L$DEVP::
        .WORD   L$DVTYP
L$REPP::
        .WORD   L$RPT
L$EXP4::
        .WORD   0
L$EXP5::
        .WORD   0
L$AUT::
        .WORD   0
L$DUT::
        .WORD   L$DU
L$LUN::
        .WORD   0
L$DESP::
        .WORD   L$DESC
L$LOAD::
        EMT     E$LOAD
L$ETP::
        .WORD   L$ERRTBL
L$ICP::
```

SEQ 0029

Program Header

```
        002104  030130
        002106
        002106  030774
        002110
        002110  030772
        002112
        002112  030122
        002114
        002114  000000
        002116
        002116  000000
        002120
        002120  000000
  1232
```

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

DISPATCH TABLE


```
                        .SBTTL  DISPATCH TABLE

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

                                DISPATCH 10
002122                                                          .WORD   10
002122  000012                                          L$DISPATCH::
002124                                                          .WORD   T1
002124  031142                                                  .WORD   T2
002126  031432                                                  .WORD   T3
002130  032004                                                  .WORD   T4
002132  032370                                                  .WORD   T5
002134  034166                                                  .WORD   T6
002136  035060                                                  .WORD   T7
002140  035306                                                  .WORD   T8
002142  036036                                                  .WORD   T9
002144  036646                                                  .WORD   T10
002146  037272
```

```
                        ;*****************************************************************************
                        ;
                        ;               VDHC.DHT
                        ;
                        ;*****************************************************************************


                        .SBTTL  DEFAULT HARDWARE P-TABLE

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

002150                          BGNHW   DFPTBL
002150  000004                                                          .WORD   L10000-L$HW/2
002152                                                          L$HW::
002152                                                          DFPTBL::

002152  160460                  .WORD   160460  ;Default CSR Address
002154  000300                  .WORD   300     ;Default Vector Address
002156  177777                  .WORD   177777  ;Default Active lines bit map
002160     002                  .BYTE   2       ;Default Loopback mode
002161     004                  .BYTE   4       ;Default BR Level

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:
```

```
                        ;*****************************************************************************
                        ;
                        ;               VDHC.EQU
                        ;
                        ;*****************************************************************************


                        .SBTTL  GLOBAL EQUATES SECTION



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

000010                          NUMLNS==10      ;NUMBER OF LINES ON DHV11-M IS 8.
000377                          MAPLNS==377     ;BIT MAP OF LINES ON DHV11-M.

                        ;****** DEVICE REGISTER OFFSETS FROM THE CSR'S ADDRESS ******
000000                          CSR0==0         ;CSR REGISTER OFFSET FROM THE CSR ADDRESS
000002                          RBUF0==2        ;RECEIVE REGISTER OFFSET FROM THE CSR ADDRESS
000002                          TXCHR0==2       ;TRANSMIT REGISTER OFFSET FROM THE CSR ADDRESS
000004                          LPR0==4         ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
000006                          STAT0==6        ;STATUS REGISTER OFFSET FROM THE CSR ADDRESS
000010                          LNCTR0==10      ;LINE CONTROL REGISTER OFFSET FROM THE CSR ADDRESS
000012                          TXAD10==12      ;TRANSMIT ADDRESS 1 REGISTER OFFSET FROM THE CSR ADDRESS
000014                          TXAD20==14      ;TRANSMIT ADDRESS 2 REGISTER OFFSET FROM THE CSR ADDRESS
000016                          TXBFC0==16      ;TRANSMIT COUNT REGISTER OFFSET FROM THE CSR ADDRESS

                        ;****** EQUATES USED WITH RESPECT TO THE RX BUFFER ******
000020                          RXBETX==16.     ;LEVEL OF RX BUFFER AT WHICH TO RE-ENABLE TRANSMISSION.
000030                          RXBDTX==24.     ;LEVEL OF RX BUFFER AT WHICH TO DISABLE TRANSMISSION.
000100                          RXBFUL==64.     ;TOTAL CHARACTER CAPACITY OF THE RX BUFFER.


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
```

```
                        ;
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                  PRI07=: 340
000300                  PRI06=: 300
000240                  PRI05=: 240
000200                  PRI04=: 200
000140                  PRI03=: 140
000100                  PRI02=: 100
000040                  PRI01=: 40
000000                  PRI00=: 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL=:           4
000010                  LOT=:          10
000020                  ADR=:          20
000040                  IDU=:          40
000100                  ISR=:         100
000200                  UAM=:         200
000400                  BOE=:         400
001000                  PNT=:        1000
002000                  PRI=:        2000
004000                  IXE=:        4000
010000                  IBE=:       10000
020000                  IER=:       20000
040000                  LOE=:       40000
100000                  HOE=:      100000
```

```
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 002176     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 POM 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.
```

GLOBAL DATA SECTION

```
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  PRI06   ;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                MMSR0:: .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                PAR0A:: .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                PDR0A:: .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:: .WORD   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.
```

GLOBAL DATA SECTION

```
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 002436  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.
```

GLOBAL DATA SECTION

```
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.
```

GLOBAL DATA SECTION SEQ 0040

```
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 005154      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
```

GLOBAL DATA SECTION

```
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 005235      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                            ;*****************************************************************************
```

GLOBAL DATA SECTION

```
1773  005276                 
1774  005276      004        STGTRB::        .BYTE   4.              ;BASE OF STAGGERED TX/RX LINE NUMBER TABLE.
1775  005277      006                        .BYTE   6.              ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
1776  005300      000                        .BYTE   0               ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
1777  005301      002                        .BYTE   2.              ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
1778  005302      014                        .BYTE   12.             ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
1779  005303      016                        .BYTE   14.             ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
1780  005304      010                        .BYTE   8.              ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
1781  005305      012                        .BYTE   10.             ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
1782  005306      024                        .BYTE   20.             ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
1783  005307      026                        .BYTE   22.             ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
1784  005310      020                        .BYTE   16.             ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
1785  005311      022                        .BYTE   18.             ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
1786  005312      034                        .BYTE   28.             ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
1787  005313      036                        .BYTE   30.             ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
1788  005314      030                        .BYTE   24.             ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
1789  005315      032                        .BYTE   26.             ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
1790                                         .EVEN                   ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
1803  005316                               ERRTBL                    ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
      005316
      005316  000000     ERRTYP::          .WORD   0                                   L$ERRTBL::
      005320  000000     ERRNBR::          .WORD   0
      005322  000000     ERRMSG::          .WORD   0
      005324  000000     ERRBLK::          .WORD   0
1804
1805                                       .EVEN
```

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

```
                        .SBTTL GPR FRAME ACCESS EQUATES
                               ;+++
                               ;Equates that allow access to the stack frame.  These are the
                               ;offsets into the stack for registers saved during the PREGOS
                               ;routine.
                               ;---

000036                         LPCSLT==        36      ;Offset for last return PC.
000016                         PCSLOT==        16      ;Offset for return PC.
000014                         R5SLOT==        14      ;Offset for R5.
000012                         R4SLOT==        12      ;Offset for R4.
000010                         R3SLOT==        10      ;Offset for R3.
000006                         R2SLOT==        6       ;Offset for R2.
000004                         R1SLOT==        4       ;Offset for R1.
000002                         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
```

```
                        .SBTTL GLOBAL SUBROUTINE                    PREG05 -
                        ;*******************************************************************************
                        ;*      Preserve Registers R0 through R5 for subroutine calls.
                        ;*
                        ;* INPUTS:      The return address back into the calling routine must be in
                        ;*              GPR R5.  (i.e.- Macros use "JSR  R5,PREG05".)
                        ;*
                        ;* OUTPUTS:     Registers R0 through R5 are saved on the stack.
                        ;*
                        ;*CALLING SEQUENCE:     SAVE            ;Macro expansion calls PREG05.
                        ;*                      [Subroutine code]...
                        ;*                      PASS            ;Macro expansion recalls PREG05.
                        ;*
                        ;*COMMENTS:     This routine is re-entrant.
                        ;*
                        ;*              Parameters may be passed out of a subroutine by modifying the
                        ;*              register save area on the stack.  Use the PASS GPRn macro
                        ;*              to return GPR values intact.
                        ;*              Use the RnSLOT offsets from the SP to pass other parameters.
                        ;*              [Example:       MOV    VALUE,R0SLOT(SP)        ]
                        ;*              Make sure the SP is at its original value when you do this.
                        ;*
                        ;*SUBORDINATE ROUTINES CALLED:  None.
                        ;*******************************************************************************

005326                  PREG05:                 ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
005326  010446                  MOV     R4,-(SP)        ;SAVE R4
005330  010346                  MOV     R3,-(SP)        ;SAVE R3
005332  010246                  MOV     R2,-(SP)        ;SAVE R2
005334  010146                  MOV     R1,-(SP)        ;SAVE R1
005336  010046                  MOV     R0,-(SP)        ;SAVE R0
005340  010546                  MOV     R5,-(SP)        ;PUSH RETURN PC ON TOP OF STACK
005342  016605  000014          MOV     R5SLOT(SP),R5   ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS

005346  004736                  JSR     PC,@(SP)+       ;Call the subroutine at the return address
                                                        ;from the PREG05 call, putting the present
                                                        ;PC on the stack as a return address into
                                                        ;this (PREG05) routine.

                                .+++
                                ;The following code is executed when the calling routine does a
                                ;"return" [JSR  PC,@(SP)+] using the PC deposited on the stack above.
                                ;---

005350  012605          PREGRT:: MOV    (SP)+,R5        ;Put return PC in R5.
005352  012600                   MOV    (SP)+,R0        ;Restore R0.
005354  012601                   MOV    (SP)+,R1        ;Restore R1.
005356  012602                   MOV    (SP)+,R2        ;Restore R2.
005360  012603                   MOV    (SP)+,R3        ;Restore R3.
005362  012604                   MOV    (SP)+,R4        ;Restore R4.

005364  000205                   RTS    R5              ;Return to the subroutine which called PREG05.
                                                          ;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
```

```
;*****************************************************************************
;
;                       VDHC.FMT
;
;*****************************************************************************



;
; FORMAT STATEMENTS USED IN PRINT CALLS
;
```

```
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;
                                ;               VDHC.MSG
                                ;
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%


        .NLIST BIN
        .SBTTL  GLOBAL MESSAGE AREA
                ; ********* FORMAT STATEMENTS **********
005430  MFUNIT:: .ASCIZ /%N%A TESTING UNIT :%D4%N/
005461  EF0503:: .ASCIZ /%T%N/
005466  EF1601:: .ASCIZ /%A  %T%A ABORTED %N/
005512  EF1603:: .ASCIZ /%A    ACTUAL DATA:   %O6%A (O).%N/
005554  EF4401:: .ASCII /%N%A DMA ADDRESS TEST SUCCESSFUL, BITS 0 TO %D2%A (D) TESTED/
005650           .ASCIZ / (%D2%A BITS).%N/
005671  EF6201:: .ASCIZ \%A FRAMING/PARITY ERROR DETECTION AND REPORTING BAD ON LINES:%D2%A : %D2%N\
006004  EF6202:: .ASCIZ /%A  CHAR RECEIVED WITH FRAMING ERROR BIT %T%A, SHOULD BE %T%N/
006102  EF6203:: .ASCIZ /%A  CHAR RECEIVED WITH PARITY ERROR BIT %T%A, SHOULD BE %T%N/
006177  EF7801:: .ASCIZ /%T%A ON LINE %D2%A DECIMAL.%N/
006235  EF8901:: .ASCIZ /%N%N%AEXECUTING THE %T%N/
006266  EF9001:: .ASCIZ /%A  UNEXPECTED %T%A FOUND IN RECEIVE CHAR FIFO:%N/
006350  EF9002:: .ASCIZ /%A    CODE IS ASSOCIATED WITH LINE: %D2%N/
006422  EF9003:: .ASCIZ /%A      CODE IS: %O3%N/
006451  EF9004:: .ASCIZ /%A    %T%A VALUE: %O3%N/
006501  EF9005:: .ASCIZ /%A    %T%A VALUE: NONE%N/
006532  EF9006:: .ASCIZ /%A  %T%A %D2%N/
006551  EF9007:: .ASCIZ /%A  CHARACTER RECEIVED WITH ERROR FLAG(S) SET ON LINE %D2%N/
006645  EF9008:: .ASCIZ /%A    CHARACTER READ AS: %O3%N/
006704  EF9009:: .ASCIZ /%A      %T%A ERROR FLAG SET.%N/
006743  EF9010:: .ASCIZ /%A    NUMBER OF ERRORS DETECTED ON LINE %D2%A IS %D5%N/
007032  EF9012:: .ASCII /%A LINE%D2%A ONLY %T%D5%A BYTES OF%D5%A BYTE/
007106           .ASCIZ / DATA PAT'N TX'D FROM LINE%D2%N/
007146  EF9013:: .ASCIZ /%A  DATA PATTERN NOT COMPLETELY %T%N/
007213  EF9019:: .ASCIZ /%A  %T%A %O6%N/
007232  EF9020:: .ASCIZ /%A    TOO FEW TX.ACTIONS GENERATED ON LINE %D2%N/
007313  EF9101:: .ASCIZ /%N/
007316  EF9103:: .ASCIZ /%A      ERROR CONDITION ON LINE %D2%N/
007364  EF9301:: .ASCIZ /%A %T%D2%A,  BMP CODE REPORTED :%O3%N/
007432  EF9302:: .ASCIZ /%A  OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)%N/
007532  UBRFMT:: .ASCIZ /%D5%A IS NOT A SUPPORTED BAUDRATE, ENTER ANOTHER OR CTRL C.%N/
007630  MSFMT1:: .ASCIZ /%AMODEM STATUS SIGNAL REPORT:%N/
007670  MSFMT2:: .ASCIZ /%A  LINE #%D2%A:  DSR=%B1%A, RI=%B1%A, DCD=%B1%A, CTS=%B1%N/
007764  EDPFMT:: .ASCII /%AMODEM LOOPBACK TEST STATUS REPORT:  /
010032           .ASCIZ /PATTERN #%D5%A (D) COMPLETED.%N/

               ;************ MESSAGE AREA ************
010072  EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
010130  EM0509:: .ASCIZ /SET/
010134  EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
010217  EM4401:: .ASCIZ /DMA ADDRESS TEST /
010241  EM4402:: .ASCIZ /NO SUITABLE ADDR FOUND,TEST ABANDONED/
010307  EM4403:: .ASCIZ /**HOST FAILURE**;WRITE FAILED TO AN ADDR WHICH HAD BEEN READ/
010404  EM4404:: .ASCIZ /NO ACTIVE LINES,TEST ABANDONED/
010443  EM4405:: .ASCIZ /DMA_START BIT FOUND SET BEFORE DMA INITIATED,TEST ABANDONED/
010537  EM4406:: .ASCIZ /TIME-OUT OCCCURED WAITING FOR DMA TO FINISH/
```

GLOBAL MESSAGE AREA SEQ 0051

```
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 ^C):/
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                    ;--
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0101 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which prints additional error
                        ;*      information if an error is detected in TEST 1 (Register Address
                        ;*      Access Test).  This subroutine reports the type of access (Read or
                        ;*      Write or both) which caused a bus time-out trap (004 trap).
                        ;*      A message indicating that the DHV may be at the wrong Q-bus address
                        ;*      is also printed.
                        ;*
                        ;* INPUTS:      R5 - Error flag word.
                        ;*                      If bit 0 is set, a read error occured.
                        ;*                      If bit 1 is set, a write error occured.
                        ;*
                        ;* OUTPUTS:     Messages are printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER0101" as the message pointer
                        ;*                      parameter in the DRS error report macro call.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*************************************************************************

                                BGNMSG  ER0101
                                                                        ER0101::
                                SAVE                    ;SAVE THE GPR CONTENTS.
013562  004537  005326                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

013566  032705  000001          BIT     #BIT0,R5        ;TEST FOR READ ERROR.
013572  001410                  BEQ     2$              ;SKIP READ ERROR MSG IF NO READ ERROR.
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$PNTB
013610  062706  000004                                                  ADD     #4,SP
013614  032705  000002          2$:     BIT     #BIT1,R5        ;TEST FOR WRITE ERROR.
013620  001410                  BEQ     4$              ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
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$PNTB
013636  062706  000004                                                  ADD     #4,SP
013642                          4$:     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$PNTX
013656  062706  000004                                                  ADD     #4,SP
013662                          PASS                    ;RESTORE THE GPR CONTENTS.
013662  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
013664                          ENDMSG
013664                                                                  L10002:
013664  104423                                                                  TRAP    C$MSG

                        .NLIST  BEX             ;$$$
```

GLOBAL ERROR REPORTING ROUTINE - ER0101 SEQ 0054

```
 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
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER0503 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which prints an additional error
                        ;*      message whose address is passed as an input parameter.
                        ;*
                        ;* INPUTS:      R1 - Address of the message to print.
                        ;*
                        ;* OUTPUTS:     A messages is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Load the address of the message in R1.
                        ;*                      Include the label "ER0503" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The message is printed as Basic error information.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*************************************************************************

014100                          BGNMSG  ER0503
014100                                                                  ER0503::
                        
014100                          PRINTB  #ER0503,R1      ;PRINT THE MESSAGE.
014100  010146                                                                  MOV     R1, (SP)
014102  012746  005461                                                          MOV     #ER0503, (SP)
014106  012746  000002                                                          MOV     #2,-(SP)
014112  010600                                                                  MOV     SP,R0
014114  104414                                                                  TRAP    C$PNTB
014116  062706  000006                                                          ADD     #6,SP

014122                          ENDMSG
014122                                                                  L10003:
014122  104423                                                                  TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER1603 -
                        ;*******************************************************************************
                        ;*      This error reporting routine is used to print out a basic error
                        ;*      message, along with a message informing the operator which test is
                        ;*      about to be aborted.
                        ;*
                        ;* INPUTS:      R1 - Contains the address of the message to be printed.
                        ;*              ERRMSG - Contains the address of the message that indicates
                        ;*              the test that is being performed, eg DMA, BREAK etc.
                        ;*
                        ;* OUTPUTS:     Messages are printed at the operators console.
                        ;*              "testname TEST ABORTED"
                        ;*
                        ;* CALLING SEQUENCE:    Include the lable "ER1603" as the message pointer
                        ;*                      parameter in the DRS error report macro call.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;*******************************************************************************
014124                          BGNMSG  ER1603
014124
014124                          SAVE                                    ER1603::
014124  004537  005326                  JSR     R5,PREGO5       ;SAVE THE CONTENTS OF THE GPRS.
                                                                ;CALL REGISTER SAVE SUBRT.
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$PNTB
014146  062706  000006                                                  ADD     #6,SP
014152  013702  005322          MOV     ERRMSG,R2       ;GET THE "TEST MESSAGE".
014156                          PRINTB #EF1601,R2       ;PRINT "TEST ABORTED" MESSAGE.
014156  010246                                                          MOV     R2, (SP)
014160  012746  005466                                                  MOV     #EF1601, (SP)
014164  012746  000002                                                  MOV     #2, (SP)
014170  010600                                                          MOV     SP,R0
014172  104414                                                          TRAP    C$PNTB
014174  062706  000006                                                  ADD     #6,SP
014200                  PASS            ;RESTORE THE CONTENTS OF THE GPRS.
014200  004736                  JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
014202                  ENDMSG
014202                                                          L10004:
014202  104423                                                          TRAP    C$MSG
```

GLOBAL ERROR REPORTING ROUTINE - ER6201 - SEQ 0057

```
                                .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER6201 -
                                ;*******************************************************************************
                                ;*      This is an error reporting subroutine which is intended for use in the
                                ;*      fram'ng error and parity error tests.  It reports error information
                                ;*      when a character has been read from the DUT with the incorrect
                                ;*      combination of Framing and Parity error bits.
                                ;*
                                ;* INPUTS:      R2   Data byte read from the DUT, including error flags.
                                ;*              R3 - Line number multiplied by 2.
                                ;*              R5 - Message flags, which messages to report.
                                ;*                 Bit1 and bit3 - indicate which messages are to be
                                ;*                 reported, Framing or Parity respectively.
                                ;*                 B't0 and bit 2 - "Set"/"Clear" message for
                                ;*                 framing and parity errors bits.
                                ;*
                                ;* OUTPUTS:     Messages are printed at the operator console.
                                ;*
                                ;* CALLING SEQUENCE:    Include the label "ER6201" as the message pointer
                                ;*                      parameter in the Diag Super error report macro call.
                                ;*
                                ;* COMMENTS:    The message is printed as Basic and Extended error information.
                                ;*              The contents of the Indirect address register field of the DUT
                                ;*              CSR may be altered.
                                ;*
                                ;* SUBORDINATE ROUTINES USED: PRTLPR.
                                ;*******************************************************************************

                                        BGNMSG  ER6201

014204                                                                  ER6201::
014204                          SAVE                    ;SAVE THE CONTENTS OF THE GPR'S.
014204  004537  005326                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.

014210  016304  005236          MOV     TXRXLB(R3),R4   ;GET THE ASSOCIATED TX LINE NUMBER.
014214  006203                  ASR     R3              ;CALCULATE THE RX LINE NUMBER.
014216  006204                  ASR     R4              ;CALCULATE THE ASSOCIATED LINE NUMBER.
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

                                ;+
                                ; Report Framing Error Problem.
                                ;-
014244  012704  011166          MOV     #EM6202,R4      ;SELECT THE "ERROR BIT CLEAR" MESSAGE.
014250  012701  010130          MOV     #EM0509,R1      ;SELECT EXPECTED "ERROR BIT SET" MESSAGE.
014254  032705  000002          BIT     #BIT1,R5        ;TEST IF FRAMING ERROR MESSAGE TO BE REPORTED.
014260  001427                  BEQ     6$              ;BRANCH TO REPORT PARITY ERROR.
014262  032705  000001          BIT     #BIT0,R5        ;TEST "ERROR BIT SET/CLEAR" MESSAGE FLAG.
014266  001403                  BEQ     2$              ;BRANCH TO REPORT ERROR BIT "CLEAR".
014270  010401                  MOV     R4,R1           ;SELECT EXPECTED "CLEAR" STATE MESSAGE.
014272  012704  010130          MOV     #EM0509,R4      ;SELECT THE "ERROR BIT SET" MESSAGE.
014276                  2$:     PRINTX  #EF6202,R4,R1   ;REPORT THE SOURCE OF THE PROBLEM.
014276  010146                                                  MOV     R1,-(SP)
014300  010446                                                  MOV     R4,-(SP)
```

GLOBAL ERROR REPORTING ROUTINE - ER6201 - SEQ 0058

```
        014302  012746  006004                                                  MOV     #EF6202,-(SP)
        014306  012746  000003                                                  MOV     #3,-(SP)
        014312  010600                                                          MOV     SP,R0
        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$:     PRINTX  #EF6203,R4,R1   ;REPORT THE SOURCE OF THE PROBLEM.
        014354  010146                                                          MOV     R1,-(SP)
        014356  010446                                                          MOV     R4,-(SP)
        014360  012746  006102                                                  MOV     #EF6203,-(SP)
        014364  012746  000003                                                  MOV     #3,-(SP)
        014370  010600                                                          MOV     SP,R0
        014372  104415                                                          TRAP    C$PNTX
        014374  062706  000010                                                  ADD     #10,SP
  2346
  2347  014400                        10$:    PRINTX  #EF1603,R2      ;REPORT ACTUAL DATA RECEIVED.
        014400  010246                                                          MOV     R2,-(SP)
        014402  012746  005512                                                  MOV     #EF1603,-(SP)
        014406  012746  000002                                                  MOV     #2,-(SP)
        014412  010600                                                          MOV     SP,R0
        014414  104415                                                          TRAP    C$PNTX
        014416  062706  000006                                                  ADD     #6,SP
  2348
  2349  014422  004737  023070                JSR     PC,PRTLPR       ;REPORT THE CONTENTS OF THE LPR FOR THIS LINE.
  2350  014426                        60$:    PASS                    ;RESTORE THE CONTENTS OF THE GPR'S.
        014426  004736                                JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
  2351  014430                        ENDMSG
        014430                                                          L10005:
        014430  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9001 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which reports an unexpected
                        ;*      code which has been found in the DUT CSR.  This code can be a BMP
                        ;*      code, a Self-test code, or a Modem Status code.
                        ;*
                        ;* INPUTS:      R1 - Address of message to print first.
                        ;*              R2 - Single byte code which has been read from the DUT.
                        ;*              R4 - Line number associated with the code.
                        ;*
                        ;* OUTPUTS:     A messages is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER9001" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The message is printed as Basic and Extended error information.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*************************************************************************

                                BGNMSG  ER9001
                                                                        ER9001::

                                PRINTB  #EF9001,R1      ;REPORT TYPE OF CODE FOUND.
                                                                        MOV     R1,-(SP)
                                                                        MOV     #EF9001,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTB
                                                                        ADD     #6,SP
                                PRINTX  #EF9002,R4      ;REPORT THE LINE NUMBER OF THE CODE.
                                                                        MOV     R4,-(SP)
                                                                        MOV     #EF9002,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #6,SP
                                PRINTX  #EF9003,R2      ;REPORT THE CODE WHICH WAS FOUND.
                                                                        MOV     R2, (SP)
                                                                        MOV     #EF9003,-(SP)
                                                                        MOV     #2,-(SP)
                                                                        MOV     SP,R0
                                                                        TRAP    C$PNTX
                                                                        ADD     #6,SP

                                ENDMSG
                                                                L10006:
                                                                        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                                ;*      transmission 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)
```

GLOBAL ERROR REPORTING ROUTINE - ER9002 - SEQ 0061

```
        014646  012746  000002                                                  MOV     #2,-(SP)
        014652  010600                                                          MOV     SP,R0
        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
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9003 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which is intended for use in the
                        ;*      transmission and reception tests.  It reports error information when
                        ;*      a character has been read from the DUT with an error flag or flags
                        ;*      set (ie. over-run, framing, or parity flag).
                        ;*
                        ;* INPUTS:      R2 - Data byte read from the DUT, including error flags.
                        ;*              R3 - Line number multiplied by 2.
                        ;*
                        ;* OUTPUTS:     A messages is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER9003" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The message is printed as Basic and Extended error information.
                        ;*              The contents of the Indirect address register field of the DUT
                        ;*              CSR may be altered.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*************************************************************************

014670                          BGNMSG  ER9003
014670                                                                  ER9003::

014670  006203                   ASR     R3              ;CALCULATE THE LINE NUMBER.
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$PRTB
014710  062706  000006                                                 ADD     #6,SP
014714  010201                   MOV     R2,R1           ;EXTRACT THE RECEIVED CHARACTER FROM THE
014716  042701  177400           BIC     #177400,R1      ;  PASSED IN CHAR VALUE WITH FLAGS.
014722                          PRINTX  #EF9008,R1      ;REPORT THE VALUE OF THE RECEIVED CHAR.
014722  010146                                                          MOV     R1,-(SP)
014724  012746  006645                                                 MOV     #EF9008,-(SP)
014730  012746  000002                                                 MOV     #2,-(SP)
014734  010600                                                          MOV     SP,R0
014736  104415                                                          TRAP    C$PNTX
014740  062706  000006                                                 ADD     #6,SP
                        ;+
                        ; Report OVERRUN flag set if necessary.
                        ;-
014744  012701  011630           MOV     #EM9011,R1      ;SELECT THE OVERRUN ERROR MESSAGE.
014750  032702  040000           BIT     #BIT14,R2       ;CHECK OVERRUN ERROR FLAG IN PASSED IN CHAR.
014754  001402                   BEQ     2$              ;SKIP ERROR IF OVERRUN ERROR FLAG WAS CLEAR.
014756  004737  015020           JSR     PC,50$          ;REPORT THE OVERRUN ERROR FLAG WAS SET.
                        ;+
                        ; Report FRAMING flag set if necessary.
                        ;-
014762  012701  011640   2$:     MOV     #EM9012,R1      ;SELECT THE FRAMING ERROR MESSAGE.
014766  032702  020000           BIT     #BIT13,R2       ;CHECK FRAMING ERROR FLAG IN PASSED IN CHAR.
014772  001402                   BEQ     4$              ;SKIP ERROR IF FRAMING ERROR FLAG WAS CLEAR.
014774  004737  015020           JSR     PC,50$          ;REPORT THE FRAMING ERROR MESSAGE.
                        ;+
```

GLOBAL ERROR REPORTING ROUTINE - ER9003 - SEQ 0063

```
 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$:    PRINTX  #EF9009,R1
      015020  010146                                                                  MOV     R1,-(SP)
      015022  012746  006704                                                         MOV     #EF9009,-(SP)
      015026  012746  000002                                                         MOV     #2,-(SP)
      015032  010600                                                                 MOV     SP,R0
      015034  104415                                                                 TRAP    C$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
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9004 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which reports error summaries
                        ;*      for lines which have exceeded the specified maximum number of
                        ;*      individual reception errors.
                        ;*
                        ;* INPUTS:      R1 - Address of message to print first.
                        ;*              ERCNTB - Label at base of line error counters table.
                        ;*              ERSMRF - "Report error summary for line" flags.
                        ;*
                        ;* OUTPUTS:     A message is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER9004" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The message is printed as Basic and Extended error information.
                        ;*              The contents of GPR's R2, R3, R4, and R5 are destroyed.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*************************************************************************

015052                          BGNMSG  ER9004
015052                                                                  ER9004::
                        
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$PNTB
015072  062706  000006                                                          ADD     #6,SP
015076  005002                  CLR     R2              ;CLEAR THE LINE COUNTER.
015100  013703  002502          MOV     ERSMRF,R3       ;GET THE ERROR SUMMARY FLAGS.
015104  005004                  CLR     R4              ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
015106  000241          2$:     CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
015110  006003                  ROR     R3              ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
015112  103013                BCC     4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
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$PNTX
015136  062706  000010                                                          ADD     #10,SP
015142  012405          4$:     MOV     (R4)+,R5        ;INCREMENT THE LINE OFFSET BY 2.
015144  005202                  INC     R2              ;INCREMT THE LINE COUNTER.
015146  005703                  TST     R3              ;CHECK THE ERROR SUMMARY FLAGS.
015150  001356                BNE     2$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.

015152                          ENDMSG
015152                                                                  L10011:
015152  104423                                                                  TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9005 -
                        ;*************************************************************************
                        ;*      This is an error reporting subroutine which reports incomplete data
                        ;*      transmissions or receptions.
                        ;*
                        ;* INPUTS:      R1 - Either "TRANSMITTED" or "RECEIVED" to indicate TX or RX.
                        ;*              R2 - Bit map of lines which did not complete TX or RX.
                        ;*              R4 - Address of base of the correct character counters table.
                        ;*              DPLENB - Label at base of data pattern length table.
                        ;*              EM9015 - Symbolic address of the "TRANSMITTED" message.
                        ;*
                        ;* OUTPUTS:     A message is printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Include the label "ER9005" as the message pointer
                        ;*                      parameter in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The message is printed as Basic and Extended error information.
                        ;*              The contents of the indirect address field in the DUT CSR may
                        ;*              be altered.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: PRTLPR.
                        ;*************************************************************************

                              BGNMSG  ER9005

015154                                                          ER9005::
                              SAVE                      ;SAVE THE CONTENTS OF THE GPR'S.
015154  004537  005326              JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

                              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$PNTB
015176  062706  000006                                          ADD     #6,SP
015202  005003                CLR     R3              ;CLEAR THE LINE COUNTER.
015204  022701  011753        CMP     #EM9015,R1      ;CHECK IF ADDRESS CORRESPONDS TO TX MESSAGE.
015210  001032                BNE     6$              ;BRANCH IF RECEPTION MESSAGE TO BE PRINTED.

                        ;+
                        ; Perform TX incomplete error message reporting.
                        ;-
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$PNTX
015226  062706  000004                                          ADD     #4,SP
015232  000241        2$:     CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
015234  006002                ROR     R2              ;SHIFT "TX NOT DONE" FLAG INTO CARRY.
015236  103013                BCC     4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
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
```

GLOBAL ERROR REPORTING ROUTINE - ER9005 - SEQ 0066

```
                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$:    PASS                            ;RESTORE THE CONTENTS OF THE GPRS.
                015376  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
2581            015400                  ENDMSG
                015400                                                  L10012:
                015400  104423                                                  TRAP    C$MSG
```

GLOBAL ERROR REPORTING ROUTINE - ER9101 - SEQ 0067

```
                                .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9101 -
                                ;*****************************************************************************
                                ;*      This is a general error reporting subroutine which reports a message
                                ;*      which takes a single, 2 digit decimal argument after the end of an
                                ;*      ASCII message.
                                ;*
                                ;* INPUTS:      R1 - Value to be printed after msg as 2 decimal digits.
                                ;*              R2 - Address of message to print first.
                                ;*
                                ;* OUTPUTS:     A messages is printed at the operator console.
                                ;*
                                ;* CALLING SEQUENCE:    Include the label "ER9101" as the message pointer
                                ;*                      peremeter in the Diag Super error report macro call.
                                ;*
                                ;* COMMENTS:    The message is printed as Basic error information.
                                ;*
                                ;* SUBORDINATE ROUTINES USED: None.
                                ;*****************************************************************************

                                        BGNMSG  ER9101
                                                                                ER9101::
                                        PRINTB  #EF9006,R2,R1   ;REPORT THE STRING FOLLOWED BY THE NUMBER.
                                                                                MOV     R1,-(SP)
                                                                                MOV     R2,-(SP)
                                                                                MOV     #EF9006,-(SP)
                                                                                MOV     #3,-(SP)
                                                                                MOV     SP,R0
                                                                                TRAP    C$PNTB
                                                                                ADD     #10,SP
                                        ENDMSG
                                                                        L10013:
                                                                                TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9102 -
                        ;*****************************************************************************
                        ;*      This is an error reporting subroutine which prints additional error
                        ;*      information after the error message header.
                        ;*      This routine is passed a bit map which specifies the lines for which
                        ;*      the error condition should be reported.
                        ;*
                        ;* INPUTS:      R1 - Address of the message to be printed by this routine.
                        ;*              R2 - B t map of lines for which to report errors.
                        ;*
                        ;* OUTPUTS:     Messages are printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    Load the address of the message in R1.
                        ;*                      Load the bit map of lines with errors in R2.
                        ;*                      Include the label "ER9102" as the messag pointer
                        ;*                         (ERRBLK) in the Diag Super error report macro call.
                        ;*
                        ;* COMMENTS:    The output format of this message is:
                        ;*              "TEXT MESSAGE POINTED TO BY R1"
                        ;*                      "ERROR CONDITION ON LINE nn"
                        ;*                      "ERROR CONDITION ON LINE ..."
                        ;*              The top message, and the message for each l ne are pr nted
                        ;*              as bas c error information.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: None.
                        ;*****************************************************************************

                                BGNMSG  ER9102
                                                                                ER9102::
                                SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
015430  004537  005326              JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

                                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$PNTB
015452  062706  000006                                          ADD     #6,SP
015456  005003                  CLR     R3              ;CLEAR THE LINE NUMBER.
015460  000241          2$:     CLC                     ;PREPARE TO ROTATE NEXT BIT OUT OF MAP.
015462  006002                  ROR     R2              ;GET THE NEXT BIT OF THE BIT MAP.
015464  103011                  BCC     4$              ;SKIP PRINTING MESSAGE IF THE BIT IS CLEAR.
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$PNTB
015504  062706  000006                                          ADD     #6,SP
015510  005203          4$:     INC     R3              ;INCREMENT THE LINE COUNTER.
015512  005702                  TST     R2              ;CHECK THE BIT MAP.
015514  001361                  BNE     2$              ;LOOP IF NOT ALL SET BITS REMOVED FROM BIT MAP.
015516                          PRINTB  #EF9101         ;PRINT A BLANK LINE.
015516  012746  007313                                          MOV     #EF9101, (SP)
015522  012746  000001                                          MOV     #1,-(SP)
015526  010600                                                  MOV     SP,R0
```

GLOBAL ERROR REPORTING ROUTINE ER9102 - SEQ 0069

```
                015530  104414                                                          TRAP    C$PNTB
                015532  062706  000004                                                  ADD     #4,SP
   2648         015536                  60$:    PASS
                015536  004736                                  JSR     PC,@(SP)+       ;RESTORE THE SAVED CONTENTS OF THE GPRS.
   2649         015540                          ENDMSG                                  ;RETURN TO PREG05 SUBRT.
                015540
                015540  104423                                                  L10014:
                                                                                        TRAP    C$MSG
```

GLOBAL ERROR REPORTING ROUTINE - ER9301 - SEQ 0070

```
                                .SBTTL  GLOBAL ERROR REPORTING ROUTINE          ER9301 -
                                ;*************************************************************************
                                ;*      This is an error reporting subroutine which prints any BMP codes
                                ;*      that are found in the BMP code queue, together with the the number of
                                ;*      the test that was executing at the time the BMP code was logged.
                                ;*
                                ;* INPUTS:      R1 - The address of the first message to be reported.
                                ;*              R2 - The address of the next empty cell in the queue.
                                ;*
                                ;* OUTPUTS:     The test number followed by the BMP code are printed at the
                                ;*              operator console.
                                ;*
                                ;* CALLING SEQUENCE:    Include the label "ER9301" as the message pointer
                                ;*                      parameter in the Diag Super error report macro call.
                                ;*
                                ;* COMMENTS:    The message is printed as Basic error information.
                                ;*
                                ;* SUBORDINATE ROUTINES USED: None.
                                ;*************************************************************************

015542                                  BGNMSG  ER9301
015542
015542                                  SAVE                                    ER9301::
015542  004537  005326                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
015546
015546  010146                  PRINTB  #EF0503,R1      ;REPORT UNEXPECTED BMP CODES FOUND.
015550  012746  005461                                                          MOV     R1,-(SP)
015554  012746  000002                                                          MOV     #EF0503,-(SP)
015560  010600                                                                  MOV     #2, (SP)
015562  104414                                                                  MOV     SP,R0
015564  062706  000006                                                          TRAP    C$PNTB
015570  012703  002514                                                          ADD     #6,SP
015574  012705  012704                  MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
015600  012301                          MOV     #EM9302,R5      ;GET THE MESSAGE TO BE REPORTED.
015602  012304          2$:             MOV     (R3)+,R1        ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
015604  004737  015666                  MOV     (R3)+,R4        ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
015610  020302                          JSR     PC,50$          ;GO REPORT THE BMP CODE.
015612  103772                          CMP     R3,R2           ;CHECK IF ALL CODES HAVE BEEN REPORTED.
                                        BLO     2$              ;IF IT IS NOT THE LAST BMP CODE THEN LOOP.
                                ;+
                                ; Check if overflow has occurred.
                                ;  The conditions for overflow are: the pointer contains the address of the
                                ;  last cell in the queue, and a bmp code has already been written into that
                                ;  cell.
                                ;-
015614  020227  002710                  CMP     R2,#BMPCQE+4    ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
015620  001036                          BNE     60$             ;EXIT IF NOT AT THE LAST LOCATION.
015622  005762  000002                  TST     2(R2)           ;CHECK FOR A BMP CODE IN THE LAST CELL
015626  001433                          BEQ     60$             ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
015630  012301                          MOV     (R3)+,R1        ;GET THE TEST NUMBER OFF THE QUEUE.
015632  011304                          MOV     (R3),R4         ;GET THE BMP CODE OFF THE QUEUE.
015634  012705  012734                  MOV     #EM9303,R5      ;SELECT THE MESSAGE TO BE REPORTED.
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
```

GLOBAL ERROR REPORTING ROUTINE ER9301 - SEQ 0071

```
        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
```

GLOBAL SUBROUTINES SECTION SEQ 0072

```
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                            ;-
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - ALTFLD -
                        ;** ************************************************************************
                        ;*              - Alter Device Register Fields Routine -
                        ;*      This subroutine alters the specified field of the specified device
                        ;*      register for the specified lines.  This routine can be used to set
                        ;*      or clear bits within selected fields of selected registers.
                        ;*      Use examples:   Set RX.BAUD.RATE fields on lines 3 and 6.
                        ;*                      Clear TX.DMA bits on all lines.
                        ;*
                        ;* INPUTS:      R1 - Address of the registers to alter.
                        ;*              R2 - Bit fields set to desired states.
                        ;*              R3 - Bit map of lines for which to alter register.
                        ;*              R4 - Mask of bits to alter (1 indicates change bit).
                        ;*              CSRA - Contains the address of the device CSR.
                        ;*              IESTAT - Saved states of the interrupt enable bits.
                        ;*
                        ;* OUTPUTS:     DEVICE REGISTERS - Specified register fields altered.
                        ;*                  CSR IND.ADR.REG field - Destroyed.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,ALTFLD
                        ;*
                        ;* COMMENTS:    This routine reads the specified registers for all lines
                        ;*              with numbers lower than the highes specified line.
                        ;*              This routine does not read the CSR.
                        ;*
                        ;* SUBROUTINES CALLED: None.
                        ; - ************************************************************************

015722                  ALTFLD:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
015722  004537  005326                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.

                        ;*
                        ; Set up to loop for each line:
                        ;    Prepare the word to be ORed into the register contents.
                        ;    Set up the word to write into the IND.ADR.REG field of the CSR.
                        ;-
015726  010400                      MOV     R4,R0           ;CALCULATE THE NEW CONTENTS OF THE
015730  005100                      COM     R0              ;  REGISTER FIELDS WHICH ARE TO BE
015732  040002                      BIC     R0,R2           ;  ALTERRED BY THIS ROUTINE.
015734  013705  002234              MOV     IESTAT,R5       ;SET UP TO WRITE IND.ADR.REG FIELD TO 0.
                        ;*
                        ; Loop once for each line, altering the specified field in the specified
                        ;    register if the line has been selected for altering.
                        ; Exit the loop if no more lines to alter, or if we have altered the max
                        ;    allowable number of lines (as specified by NUMLNS).
                        ;-
015740  000241                      CLC                     ;PREPARE FOR ROTATE, "TST R5" DOES THIS BELOW.
015742  006003          2$:         ROR     R3              ;GET THE LINE SELECT BIT FOR THIS LINE.
015744  103006                      BCC     4$              ;SKIP SETUP IF LINE IS NOT SELECTED.
015746  010577  164230              MOV     R5,@CSRA        ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
015752  011100                      MOV     (R1),R0         ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
015754  040400                      BIC     R4,R0           ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
015756  050200                      BIS     R2,R0           ;OR IN THE NEW STATES OF THE FIELDS.
015760  010011                      MOV     R0,(R1)         ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
015762  005205          4$:         INC     R5              ;SET LINE NUMBER TO THE NEXT LINE.
015764  005703                      TST     R3              ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
015766  001365                      BNE     2$              ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.
```

GLOBAL SUBROUTINE - ALTFLD - SEQ 0074

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

```
                        .SBTTL  GLOBAL SUBROUTINE               - CALMSL -
                        ;** ************************************************************************
                        ;*              - Calibrate Milli Second Loop count subroutine -
                        ;*      This subroutine calibrates the timing loop which is used in the MSLOOP
                        ;*      routine.  This subroutine calculates a value for the MSLCNT variable
                        ;*      which is the number of software loops which takes 1 ms to execute in
                        ;*      the MSLOOP routine.  This routine calibrates the count by using the
                        ;*      Line Time Clock (LTC), so if no LTC is available the default value for
                        ;*      the delay count must be used.
                        ;*
                        ;*
                        ;* INPUTS:      MSLCNT - Default 1 ms delay loop count value, or
                        ;*                       value from previous calibration.
                        ;*              MSTICK - Number of MS per LTC clock tick.
                        ;*              TIMER1   Timer counter changed by LTC interrupt service rtn.
                        ;*              CLKHRZ - Number of LTC clicks per second (50 or 60).
                        ;*
                        ;* OUTPUTS:     CARRY   Set if LTC is available, and new calibration performed.
                        ;*              MSLCNT - New 1 ms delay loop count value if LTC available, or
                        ;*                       unchanged if no LTC is available.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CALMSL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: UNSDIV,OOPS.
                        ;-- ************************************************************************

2808 015774             CALMSL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     015774  004537  005326             JSR     R5,PREGOS       ;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 - SEQ 0076

```
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                                ; We have now have loop count information for one clock tick.
2855                                ; We have negative of number of outer loops in R1, each is MSLCNT inner loops.
2856                                ; We have the portion of the last outer loop not executed, in R5.
2857                                ; Now we calculate the total number of inner loops executed.
2858                                ;-
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                                ; Divide the total number of inner loops by the number of MS per LTC tick.
2872                                ;
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.
```

GLOBAL SUBROUTINE - CHKEXT -

```
                        .SBTTL  GLOBAL SUBROUTINE               - CHKEXT -
                        ;++ ************************************************************************
                        ;*                      - Check For Extra Character Routine -
                        ;*      This subroutine checks for the condition which indicates that an extra
                        ;*      character has been received during the reception of a data pattern.
                        ;*      If this routine determines that it is likely that an extra character
                        ;*      has been received it indicates this in the status information returned
                        ;*      to the calling routine.
                        ;*
                        ;* INPUTS:      R3 - RX line number multiplied by 2 (offset into word tables).
                        ;*              R4 - Base address of resync que containing RX chars.
                        ;*              R5 - Mask of "inactive" (non-data) bits of RX and TX chars.
                        ;*              CHCNTB - Base of number of chars to TX on each line table.
                        ;*              RXCNTB - Base of the RX character counters table.
                        ;*              RXPTRB - Base of the RX character pointers table.
                        ;*              TXRXLB - Base of TX/RX line number association table.
                        ;*
                        ;* OUTPUTS:     CARRY - Set if extra character condition is verified.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CHKEXT
                        ;*
                        ;* COMMENTS:  The following symbols are used in line comments:
                        ;*              CHR0 - Character at bottom of resync que (first received).
                        ;*              CHR1, CHR2 - 2 characters received after CHR0.
                        ;*              EXP0 - Character expected to be received next.
                        ;*              EXP1, EXP2 - Character expected to be received after EXP0, etc.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ************************************************************************
016220                  CHKEXT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
016220  004537  005326          JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
016224  016302  003404          MOV     RXPTRB(R3),R2   ;GET THE RX DATA POINTER.
016230  005724                  TST     (R4)+           ;INCREMENT R4 BY 2 TO POINT TO CHR1.
016232  012400                  MOV     (R4)+,R0        ;GET CHR1 FROM THE QUE, DATA.VALID INTO N FLAG.
016234  100026                  BPL     52$             ;EXIT WITH "FAILURE" IF CHR1 NOT VALID.
016236  040500                  BIC     R5,R0           ;REMOVE INACTIVE BITS FROM CHR1 VALUE.
016240  112201                  MOVB    (R2)+,R1        ;GET EXP0 FROM THE DATA PATTERN.
016242  040501                  BIC     R5,R1           ;REMOVE INACTIVE BITS FROM EXP0 VALUE.
016244  120100                  CMPB    R1,R0           ;COMPARE CHR1 AND EXP0.
016246  001021                  BNE     52$             ;EXIT WITH "FAILURE" IF CHR1 <> EXP0.
016250  016300  003544          MOV     RXCNTB(R3),R0   ;COMPARE THE PRESENT RX CHARACTER COUNT PLUS 1
016254  005200                  INC     R0              ;  WITH THE EXPECTED NUMBER OF CHARS TO RX ON
016256  016301  005236          MOV     TXRXLB(R3),R1   ;  LINE (NUMBER TRANSMITTED AND LOOPED BACK) TO
016262  020061  003444          CMP     R0,CHCNTB(R1)   ;  DETERMINE IF CHR1 IS LAST EXPECTED CHAR.
016266  001407                  BEQ     50$             ;EXIT WITH "SUCCESS" IF CHR1 IS LAST CHAR.
016270  011400                  MOV     (R4),R0         ;GET CHR2 FROM THE QUE, DATA.VALID INTO N FLAG.
016272  100005                  BPL     50$             ;EXIT WITH "SUCCESS" IF CHR1 WAS LAST IN QUE.
016274  040500                  BIC     R5,R0           ;REMOVE INACTIVE BITS FROM CHR2 VALUE.
016276  111201                  MOVB    (R2),R1         ;GET THE EXP1 VALUE.
016300  040501                  BIC     R5,R1           ;REMOVE INACTIVE BITS FROM EXP1 VALUE.
016302  020001                  CMP     R0,R1           ;COMPARE CHR2 AND EXP1.
016304  001002                  BNE     52$             ;EXIT WITH "FAILURE" IF CHR2 <> EXP1.

                        ;+
                        ; It is likely that we received an extra character within the data pattern.
                        ; Indicate "success" and exit.
                        ;-
```

GLOBAL SUBROUTINE - CHKEXT - SEQ 0078

```
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 PREGOS SUBRT.
2956 016316  000207                        RTS     PC              ;CARRY - SET IF SUCCESS (EXTRA CHAR RXED).
```

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

GLOBAL SUBROUTINE CHKLOS - SEQ 0080

```
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 PREGO5 SUBRT.
3028 016420  000207                             RTS     PC              ;CARRY - SET IF SUCCESS (LOST CHAR LIKELY).
```

```
                        ;BIIL  GLOBAL SUBROUTINE                CHRMSK -
                        ;++ ************************************************************************
                        ;*              Form a Bit Mask of Unused TX/RX Bits Routine -
                        ;*      This subroutine constructs a bit mask of character bits which are not
                        ;*      used during transmission and reception.  This mask can be used
                        ;*      to remove the flags, line number, DATA.VALID bits, and unused data bits
                        ;*      from a character word which has been read from the DUT FIFO.
                        ;*
                        ;* INPUTS:      R1   DUT LPR contents used to determine character length.
                        ;*
                        ;* OUTPUTS:     IBM - Bit mask of unused TX/RX bits (including upper byte):
                        ;*                    Examples: 177400 returned for 8 bits/char.
                        ;*                              177700 returned for 6 bits/char.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CHRMSK
                        ;*
                        ;* COMMENTS:    If this mask is to be used to just remove the inactive bits
                        ;*              within the data byte of a word read from the DUT FIFO, the
                        ;*              upper byte of the mask must be cleared.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ************************************************************************

016422                  CHRMSK:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016422  004537  005326                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
016426  052701  177740          BIS     #177740,R1              ;PREPARE TO COUNT BITS SHIFTED INTO MASK BY
                                                                ;  USING THE LPR BITS/CHAR FIELD CONTENTS.
016432  012703  177400          MOV     #177400,R3              ;CLEAR THE UNUSED BIT MAP LOWER BYTE.
016436  062701  000010  2$:     ADD     #10,R1                  ;DETERMINE IF ANOTHER BIT WOULD BE TOO MANY.
016442  103402                  BCS     4$                      ;EXIT THE SHIFT LOOP IF IT WOULD BE TOO MANY.
016444  006203                  ASR     R3                      ;SHIFT A BIT INTO THE UNUSED BIT MASK LOW BYTE.
016446  000773                  BR      2$                      ;LOOP TO CHECK FOR DONE.

016450  010337  002226  4$:     MOV     R3,IBM                  ;LOAD THE INACTIVE BITS MASK STORAGE IN MEMORY.

016454                  60$:    PASS                            ;RESTORE GPRS.
016454  004736                          JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
016456  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - CKCHR -
                        ;** ************************************************************************
                        ;*                      - Check Character For Errors Routine -
                        ;*      This subroutine checks the character at the bottom of the resync queue
                        ;*      to determine if it is correct.  Pointers and counters which are related
                        ;*      to the reception of the character are updated.  If the character is
                        ;*      incorrect, an analysis of the error is done and parameters are set up
                        ;*      for the reporting of the correct error.
                        ;*
                        ;* INPUTS:      R3 - Line offset for access of word tables of line variables.
                        ;*              R4 - Base address of the resync queue for this line.
                        ;*              R5 - Mask of the inactives bits in a TX or RX char byte.
                        ;*              BITTBL - Table of words with bits set for use in forming maps.
                        ;*              DPRSQ - Data Pattern Resync Que with valid char at bottom.
                        ;*              EXCNTB - Base of the extra character couters table.
                        ;*              RXDONF - Receive done flags.
                        ;*              RXPTRB - Base of the RX character pointers table.
                        ;*              Error Message Labels - EM9007,EM9008,EM9027,EM9028
                        ;*
                        ;* OUTPUTS:     R1 - Contains the address of the error message to be reported.
                        ;*              R2 - Contains the actual received data.
                        ;*              R4 - Contains the expected data.
                        ;*             CARRY - "Success" flag (set if no error is found).
                        ;*           Following variables updated for line on which char was received:
                        ;*             EXCNT   Count of the number of extra chars received on line.
                        ;*             RXCNT - Count of the number of characters received on line.
                        ;*             RXPTR - Updated to point to the next expected char on line.
                        ;*             ERRBLK - Contents destroyed.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CKCHR
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: CHKEXT,CHKLOS,UPDCHR.
                        ;-- ************************************************************************
016460                  CKCHR:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
016460  004537  005326                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; Check for the RX of a char after RX should be complete on this line.
                        ;-
016464  036337  002366  002506                BIT     BITTBL(R3),RXDONF       ;TEST THE RX DONE FLAG FOR THIS LINE.
016472  001407                              BEQ     2$              ;SKIP ERROR REPORT IF RX NOT COMPLETE ON LINE.
                        ;*
                        ; We have received an extra character on this line.
                        ; Set up for error report and exit to report the error.
                        ; Count the extra character.
                        ; Exit to report "UNEXPECTED CHAR RECEIVED AFTER RX COMPLETE ON LINE nn"
                        ;-
016474  012701  011414                        MOV     #EM9007,R1      ;SELECT "EXTRA CHAR ON LINE" ERROR MESSAGE.
016500  011402                              MOV     (R4),R2         ;GET THE ACTUAL DATA FOR ERROR REPORT.
016502  040502                              BIC     R5,R2           ;REMOVE THE INACTIVE BITS.
016504  052704  100000                      BIS     #BIT15,R4       ;INDICATE "NONE" EXPECTED DATA FOR ERROR RPT.
016510  000452                             BR      12$             ;GO COUNT EXTRA CHAR AND EXIT WITH "FAILURE".
                        ;*
                        ; Get the pointer to the next expected receive data character.
                        ;
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 t . 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.
```

```
                                        ;+
                                        ; Update the character counter and RX data pattern pointer for this line.
                                        ;
016630  004737  026664          10$:    JSR     PC,UPDCHR       ;UPDATE RX PTR AND COUNTER FOR THIS LINE.
016634  000405                          BR      14$             ;GO EXIT WITH "FAILURE".
                                        ;+
                                        ; Count the character as an extra character.
                                        ;-
016636  005263  003244          12$:     INC    EXCNTB(R3)      ;INCREMENT THE EXTRA CHAR COUNT FOR THIS LINE.
016642  001002                          BNE     14$             ;EXIT WITH FAILURE IF NO OVERFLOW.
016644  005363  003244                   DEC    EXCNTB(R3)      ;DECREMENT BACK TO -1 (MAX VALUE) IF OVERFLOW.
                                        ;+
                                        ; Indicate "failure" and exit.
                                        ;-
016650  000241                  14$:     CLC                    ;CLEAR THE "SUCCESS" FLAG.
016652  000401                          BR      60$             ;EXIT THE ROUTINE.

                                        ;+
                                        ; No error was found.
                                        ; Set "success" flag and ex't.
                                        ;-
016654  000261                  50$:     SEC                    ;SET THE "SUCCESS" FLAG.

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 PREGO5 SUBRT.
                                                ;R1 - CONTAINS THE ADDRESS OF THE ERROR REPORT.
                                                ;R2 - CONTAINS THE ACTUAL DATA RECEIVED.
                                                ;R4 - CONTAINS THE EXPECTED DATA.
016674  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - CKFRPR -
                        ;** ************************************************************************
                        ;*              - Check Framing and Parity Error reporting -
                        ;*      This subroutine is used in the Framing error and Parity error tests.
                        ;*      It reads the characters from the DUT Receiver Character FIFO,
                        ;*      and checks for the correct combination of Parity and Framing
                        ;*      error bits in the MSB.  If characters stop appearing in the FIFO with
                        ;*      DATA.VALID set or if more than the allowable number of characters
                        ;*      has been read from the DUT this routine exits wit  an RX complete
                        ;*      indication.  Each read char is analysed and any necessary errors are
                        ;*      reported.
                        ;*
                        ;* INPUTS:      R5 - Test flag, bit15 set = Framing err, clear = Parity err.
                        ;*              ERRNBR - Set to error number of first error in this routine.
                        ;*              OSTEND - Address of the end of the output storage fifo buffer.
                        ;*              OSTPTR - Pointer to the next byte to read from OSTORE.
                        ;*
                        ;* OUTPUTS:     RXCNTB - Receive charater count updated for each line.
                        ;*              RXPNTB - Receive character pionter is updated for each line.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CKFRPR
                        ;*
                        ;* COMMENTS:    This routine reports errors with numbers Initial ERRNBR
                        ;*              thru Initial ERRNBR + 4.
                        ;*              ERRNBR is restored before this routine returns.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: PRFRME,PRPARE,WAIBIS.
                        ;-- ************************************************************************

016676                  CKFRPR:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
016676  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016702  013704  005320          MOV     ERRNBR,R4       ;PRESERVE THE INITIAL ERROR NUMBER.
016706  004737  026070          JSR     PC,TXIE1        ;ENABLE TX INTERRUPTS.
                        ;+
                        ; Wait for a character to appear in the FIFO.
                        ; If no character appears within time-out period: exit routine, we're done.
                        ;-
016712  013701  002242          MOV     RXTOUT,R1       ;GET MINIMUM TIME OUT VALUE.
016716  023737  002504  002174  2$:     CMP     TXDONF,ACTLNS   ;CHECK FOR TRANSMISSION DONE ON ACTIVE LINES.
016724  001402                  BEQ     4$              ;SKIP ADDING 50 MS DELAY IF TX DONE ALL LINES.
016726  062701  000062          ADD     #50.,R1         ;ADD 50 MILLI SEC TO DELAY IF NOT LAST CHAR.
016732  052701  170000          4$:     BIS     #170000,R1      ;INDICATE TO TEST DATA.VALID BIT.
016736  013702  002204          MOV     RBUFA,R2        ;INDICATE TO CHECK DUT RECEIVE BUFFER (FIFO).
016742  004737  027160          JSR     PC,WAIBIS       ;WAIT FOR RECEIVED CHAR OR TIME-OUT.
016746  103033                  BCC     60$             ;EXIT ROUTINE IF TIME OUT, WE'RE DONE.

016750  005337  002500          DEC     CHRTOT          ;DECREMENT THE TOTAL CHAR COUNTER.
016754  001011                  BNE     6$              ;SKIP ERROR IF NOT TOO MANY CHARS RECEIVED.
016756  010437  005320          MOV     R4,ERRNBR       ;SET ERROR NUMBER TO INITIAL ERRNBR.
016762  012701  012107          MOV     #EM9025,R1      ;SELECT THE ERROR MESSAGE TO BE REPORTED.
016766  012737  014100  005324  MOV     #ER0503,ERRBLK  ;SELECT THE ERROR REPORT ROUTINE.
                        ;+
                        ;       Report error at initial ERRNBR.
                        ;       "MORE THAN TWICE THE EXPECTED NUMBER OF CHARACTERS RECEIVED"
                        ;-
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
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - CKINAC -
                        ;+  ************************************************************************
                        ;*              - Check for New Character on Inactive Line Routine -
                        ;*      This subroutine checks a character to determine if the character
                        ;*      was received on an active line.  If the character was received on
                        ;*      an inactive line this routine records the fact that the character
                        ;*      was received on an inactive line, prepares an error message for
                        ;*      the calling routine, and returns a "failure" status.
                        ;*
                        ;* INPUTS:      R2 - The RX character including error flags and line number.
                        ;*              ACTLNS - Bit map of active DUT lines.
                        ;*              BITTBL - Table of words with bits set for forming bit maps.
                        ;*              EM9006 - Label at "RX ON INACTIVE LINE" error messge.
                        ;*              EXCNTB - Base of the extra character couters table.
                        ;*              TXRXLB - Base of TX/RX line number association table.
                        ;*
                        ;* OUTPUTS:     CARRY - "Success" flag (set if no error found).
                        ;*              R1 - If error found, address of error message.
                        ;*              R3 - Line number offset of passed in character.
                        ;*              R4 - If error found, expected data indication for error rpt.
                        ;*              EXCNT - Extra character count for line (Updated if error).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CKINAC
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

017046                  CKINAC:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
017046  004537  005326          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; Extract the line number from the passed in character and use the line
                        ;   number to form an offset for accessing tables of line variables.
                        ;-
017052  010203                  MOV     R2,R3           ;EXTRACT THE LINE NUMBER
017054  000303                  SWAB    R3              ;  FROM THE CHARACTER WE
017056  042703  177760          BIC     #177760,R3      ;  ARE COMPARING.
017062  006303                  ASL     R3              ;FORM OFFSET INTO WORD TABLE FROM LINE NUMBER.
                        ;+
                        ; If the character in question is not a valid character, exit with "success".
                        ;-
017064  005702                  TST     R2              ;CHECK DATA.VALID BIT.
017066  100021                  BPL     50$             ;EXIT WITH SUCCESS IF CHAR IS NOT VALID.
                        ;+
                        ; If the TX line which is associated with this RX line is an active line,
                        ;  exit the routine with "success".
                        ;
017070  016301  005236          MOV     TXRXLB(R3),R1   ;GET THE TX LINE # OFFSET FOR THIS RX LINE.
017074  036137  002366  002174  BIT     BITTBL(R1),ACTLNS  ;DETERMINE IF TX LINE IS AN ACTIVE LINE.
017102  001013                  BNE     50$             ;EXIT ROUTINE WITH SUCCESS IF LINE IS ACTIVE.
                        ;+
                        ; The character in question was received on an inactive line.
                        ; Count this character as an extra char.
                        ; Set up error information.
                        ; Exit routine with "failure" indication.
                        ;-
```

GLOBAL SUBROUTINE - CKINAC - SEQ 0088

```
 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).
```

```
                                .SBTTL  GLOBAL SUBROUTINE               - CKTRAP -
                                ;*******************************************************************************
                                ;*      Check Trap Routine -
                                ;*      This subroutine is used to check for a bus time-out trap (004 trap)
                                ;*      which is caused by an access to a non-existent memory or I/O location.
                                ;*      If the trap does not occur, this routine returns a success indication.
                                ;*
                                ;* INPUTS:      R0 - Source address for move.
                                ;*              R1 - Destination address for move.
                                ;*              (R0)   Source for the move.
                                ;*
                                ;* OUTPUTS:     (R1) - Written to the contents of (R0).
                                ;*              Carry flag - Set on return if no 004 trap detected.
                                ;*              TP4FLG - Nonzero if trap occurred, cleared otherwise.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,CKTRAP
                                ;*
                                ;* COMMENTS:    If this subroutine causes a trap, either the address which
                                ;*              is labeled ADRPTR will be the trap PC address on the stack.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;*******************************************************************************

017154                          CKTRAP:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
017154  004537  005326                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017160  005037  002256                CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAGS.
017164  011011                        MOV     (R0),(R1)       ;PERFORM THE MOVE IN QUESTION.
017166  005737  002256        ADRPTR:: TST    TP4FLG          ;CHECK FOR OCCURENCE OF TRAP.
017172  000261                        SEC                     ;INDICATE SUCCESS.
017174  001401                        BEQ     60$             ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
017176  000241                        CLC                     ;INDICATE FAILURE.
017200                          60$:    PASS                    ;RESTORE GPRS.
017200  004736                                JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
017202  000207                        RTS     PC
```

```
                        .SBTTL  GOBAL SUBROUTINE                - CKTRPB -
                        ;*************************************************************************
                        ;*                              - CHECK FOR TRAP -
                        ;*      THIS SUBROUTINE IS USED TO CHECK FOR A BUS TIME-OUT TRAP (004 TRAP)
                        ;*      WHICH IS CAUSED BY AN ACCESS TO A NON-EXISTENT MEMORY OR I/O LOCATION
                        ;*      IF A TRAP DOES NOT OCCUR, THIS ROUTINE RETURNS A SUCCESS INDICATION.
                        ;*
                        ;*      INPUTS:         R0 - SOURCE ADDRESS FOR MOVE
                        ;*                      R1 - DESTINATION ADDRESS FOR MOVE
                        ;*                      (R0) - SOURCE FOR THE MOVE
                        ;*
                        ;*      OUTPUTS:        (R1) - WRITEN TO THE CONTENTS OF (R0)
                        ;*                      CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED
                        ;*                      TP4FLG - NONZERO IF TRAP OCCURED, CLEARED OTHERWISE.
                        ;*
                        ;*      CALLING SEQUENCE:       JSR     PC,CKTRPB
                        ;*
                        ;*      COMMENTS:       IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS
                        ;*                      WHICH IS LABELED TRPAD2 WILL BE THE TRAP PC ADDRESS ON
                        ;*                      THE STACK OR SOME OTHER ADDRESS WHICH WAS PLACED ON
                        ;*                      THE STACK BY AN UNEXPECTED TRAP.
                        ;*                              THIS ROUTINE PERFORMS A BYTE MOV .
                        ;*
                        ;*      SUBORDINATE ROUTINES CALLED:    NONE.
                        ;*************************************************************************

017204                  CKTRPB::        SAVE
017204  004537  005326                          JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.

017210  005037  002256                  CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAGS
017214  111011                          MOVB    (R0),(R1)       ;PERFORM THE BYTE MOVE
017216  005737  002256          TRPAD2:: TST    TP4FLG          ;CHECK FOR OCCURENCE OF TRAP
017222  000261                          SEC                     ;INDICATE SUCCESS
017224  001401                        BEQ     60$             ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR
017226  000241                          CLC                     ;INDICATE FAILURE
017230                          60$:    PASS                    ;
017230  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
017232  000207                          RTS     PC              ;RETURN
```

```
                                .SBTTL  GLOBAL SUBROUTINE               - CLNRST -
                                ;*******************************************************************************
                                ;*              - Clean Reset of the Device Under Test -
                                ;*      This subroutine is used to reset the DUT to a known state.
                                ;*      The DUT's self-test is skipped,and the fifo is purged of any error
                                ;*      codes, etc.
                                ;*      If the reset does not succesfully complete, then the carry bit is
                                ;*      passed back to the calling routine (clear).
                                ;*
                                ;* INPUTS:      CSRA - Contains the address of the CSR
                                ;*              TXBFCA - Contains address of DUT DMA Buffer Count register.
                                ;*              ERRNBR - Error number for possible error report.
                                ;*              ERRTBL- ERRTYP,ERNBR,and ERRMSG set up correctly.
                                ;*
                                ;* OUTPUTS:     The DUT performs its reset function into a known state.
                                ;*              CARRY - Clear indicates the test is to be aborted.
                                ;*              ERRBLK - value may be destroyed.
                                ;*              IESTAT - TX and RX interrupt flags are cleared.
                                ;*              TX and RX interrupt enable bits in the DUT's CSR are cleared.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,CLNRST
                                ;*
                                ;* COMMENTS:    This subroutine can report errors with numbers ERRNBR.
                                ;*              This routine does not destroy the value of ERRNBR.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET,PUFIFO,RESETT.
                                ;*******************************************************************************

017234                          CLNRST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017234  004537  005326                                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                ;+
                                ; Reset the DUT.
                                ;  This routine reports errors with numbers from ERRNBR thru ERRNBR+2.
                                ;-
017240  004737  024246                  JSR     PC,RESETT       ;RESET THE DUT TO A KNOWN STATE.
017244  103002                  BCC     60$             ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
                                ;+
                                ; Purge the FIFO of error codes, save any BMP codes found.
                                ;-
017246  004737  023152                  JSR     PC,PUFIFO       ;PURGE THE FIFO.

017252                          60$:                            ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
017252                  PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
017252  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                        ;CARRY BIT:IF CLEAR, THEN ABORT THE TEST.
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
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - CONMAP -
                        ;** ****************************************************************************
                        ;*                       - Convert Line bit map.
                        ;*      This subroutne is used to convert a bit map passed to it , into
                        ;*      another line bit map that is based upon the associated TX/RX line
                        ;*      number/offset table.
                        ;*
                        ;* INPUTS:      R5   Contains the line bit map to be transformed.
                        ;*              TXRXLB - Base address of associated TX/RX line number table.
                        ;*
                        ;* OUTPUTS:     R5 - Contains an associated line bit map.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,CONMAP
                        ;*
                        ;* COMMENTS:    The TX/RX association table must be initialised before this
                        ;*              routine is called.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ****************************************************************************

017300                  CONMAP::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
017300  004537  005326          JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
017304  012702  005236          MOV     #TXRXLB,R2      ;GET THE BASE ADDRESS OF THE LINE ASSOC TABLE.
017310  010503                  MOV     R5,R3           ;COPY THE BIT MAP TO BE TRANSFORMED.
017312  012704  000010          MOV     #NUMLNS,R4      ;SET MAX LINE COUNTER.
017316  005005                  CLR     R5              ;CLEAR ASSOCIATED LINE BIT MAP.
017320  006203          2$:     ASR     R3              ;SHIFT ACTLNS BIT MAP INT BOOLEAN REGISTER.
017322  103005                  BCC     4$              ;SKIP SETTING ASSOCIATED LINE NUMBER BIT MAP.
017324  011201                  MOV     (R2),R1         ;GET ASSOCIATED LINE NUMBER OFFSET FROM TABLE.
017326  006201                  ASR     R1              ;SHIFT RIGHT TO GET LINE NUMB FROM OFFSET.
017330  004737  021276          JSR     PC,LINBIT       ;GENERATE AN SINGLE BIT MAP FOR THIS LINE.
017334  050005                  BIS     R0,R5           ;SET BIT FOR THIS LINE IN ASSOCIATED BIT MAP.
017336  005722          4$:     TST     (R2)+           ;INCREMENT ADDRESS FOR THE NEXT LINE NUMBER.
017340  005304                  DEC     R4              ;DECREMENT LINE COUNT.
017342  001366                  BNE     2$              ;LOOP IF NOT DONE.
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 PREGOS SUBRT.
                                ;R5   CONTAINS THE ASSOCIATED LINE BIT MAP.
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.
```

```
                        .SBTTL  GLOBAL SUBROUTINE       - DM16B -
                        ;** ************************************************************************
                        ;*              - CONVERT TO A 16-BIT PHYSICAL ADDRESS -
                        ;*      THIS ROUTINE CONVERTS FROM PAR FORM TO A 16-BIT PHYSICAL ADDRESS,
                        ;*      OF ALTERNATE 1'S AND 0'S.
                        ;*
                        ;*      INPUTS:         DMTSTA: - CONTAINS THE ADDRESS IN PAR FORM
                        ;*
                        ;*      OUTPUTS:        R0 - CONTAINS THE 16 BIT PHYSICAL ADDRESS
                        ;*
                        ;*      CALLING SEQUENCE:       JSR     PC,DM16B
                        ;*
                        ;*      COMMENTS:       USED IN THE DMA ADDRESS TEST
                        ;*
                        ;*      SUBROUTINES CALLED:     NONE.
                        ;-- ************************************************************************

017414                  DM16B:: SAVE
017414  004537  005326                        JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
017420  013700  002264          MOV     DMTSTA,R0       ;SHIFT THE DMA TEST ADDRESS
017424  012702  000006          MOV     #6,R2           ;SIX PLACES LEFT , TO
017430  006300          2$:     ASL     R0              ;CONVERT IT INTO A ,
017432  005302          DEC     R2              ;16 BIT PHYSICAL ADDRESS
017434  001375          BNE     2$              ;

017436  012701  000052          MOV     #52,R1          ;SET UP THE 6 LSB'S
017442  032700  000100          BIT     #100,R0         ;IF BIT #6 OF THE PHYSICAL
017446  001402          BEQ     4$              ;ADDRESS IS CLEAR THEN BRANCH
017450  012701  000025          MOV     #25,R1          ;OTHERWISE CORRECT THE LSB'S

017454  060100          4$:     ADD     R1,R0           ;MREGE THE LSB'S WITH THE PHY ADDR

017456                  PASS    R0                      ;RETURN WITH THE PHY ADDR.
017456  010066  000002                        MOV     R0,ROSLOT(SP)           ;PUT R0 IN STACK SLOT.
017462  004736                              JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
017464  000207          RTS     PC
```



```
                        .SBTTL  GLOBAL SUBROUTINE       - DM16B -
                        ;** ************************************************************************
                        ;*              - CONVERT TO A 16-BIT PHYSICAL ADDRESS -
                        ;*      THIS ROUTINE CONVERTS FROM PAR FORM TO A 16-BIT PHYSICAL ADDRESS,
                        ;*      OF ALTERNATE 1'S AND 0'S.
                        ;*
                        ;*      INPUTS:         DMTSTA: - CONTAINS THE ADDRESS IN PAR FORM
                        ;*
                        ;*      OUTPUTS:        R0 - CONTAINS THE 16 BIT PHYSICAL ADDRESS
                        ;*
                        ;*      CALLING SEQUENCE:       JSR     PC,DM16B
                        ;*
                        ;*      COMMENTS:       USED IN THE DMA ADDRESS TEST
                        ;*
                        ;*      SUBROUTINES CALLED:     NONE.
                        ;-- ************************************************************************

017414                  DM16B:: SAVE
017414  004537  005326                        JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
017420  013700  002264          MOV     DMTSTA,R0       ;SHIFT THE DMA TEST ADDRESS
017424  012702  000006          MOV     #6,R2           ;SIX PLACES LEFT , TO
017430  006300          2$:     ASL     R0              ;CONVERT IT INTO A ,
017432  005302          DEC     R2              ;16 BIT PHYSICAL ADDRESS
017434  001375          BNE     2$              ;

017436  012701  000052          MOV     #52,R1          ;SET UP THE 6 LSB'S
017442  032700  000100          BIT     #100,R0         ;IF BIT #6 OF THE PHYSICAL
017446  001402          BEQ     4$              ;ADDRESS IS CLEAR THEN BRANCH
017450  012701  000025          MOV     #25,R1          ;OTHERWISE CORRECT THE LSB'S

017454  060100          4$:     ADD     R1,R0           ;MREGE THE LSB'S WITH THE PHY ADDR

017456                  PASS    R0                      ;RETURN WITH THE PHY ADDR.
017456  010066  000002                        MOV     R0,ROSLOT(SP)           ;PUT R0 IN STACK SLOT.
017462  004736                              JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
017464  000207          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - DMRW -
                        ;** ************************************************************************
                        ;*                      - READ/WRITE DATA FROM/TO (DMTSTA) -
                        ;*      THIS ROUTINE READS DATA BYTES FROM OR WRITES DATA BYTES TO AN ADDR OF
                        ;*      ALTERNATE 1'S AND 0'S . BITS 21 TO 6 OF THE ADDR ARE CONTAINED AT
                        ;*      DMTSTA. THE ROUTINE APPENDS THE 6 LSB'S TO PRODUCE AN ADDR OF ALTERNATE
                        ;*      1'S AND 0'S. THIS ROUTINE IS CALLED FROM THE DMA ADDRESS TEST.
                        ;*
                        ;* INPUTS:              R0 - ADDRESS OF THE  DATA TO BE WRITTEN TO (DMTSTA),
                        ;*                           IF A WRITE IS SPECIFIED.
                        ;*                      R1   ADDRESS OF THE AREA IN WHICH DATA FROM (DMTSTA),
                        ;*                           IS T' BE SAVED,IF A READ IS SPECIFIED.
                        ;*                      R3   NUMBER OF DATA BYTES TO BE READ/WRITTEN
                        ;*                      R5 - CLEAR , SPECIFIES A READ FROM (DMTSTA)
                        ;*                           SET , SPECIFIES A WRITE TO (DMTSTA).
                        ;*                      DMTSTA - CONTAINS BITS 21 TO 6 OF THE ADDR.
                        ;*                      MMSR0 - ADDRESS OF MEM MGT STATUS REG #0
                        ;*                      MMPRES   BIT #0 SET, INDICATES MEM MGT PRESENT
                        ;*                      PAR6 - ADDRESS OF MEM MGT PAR #6
                        ;*                      [↑ is replaced with PAR5] $$$
                        ;*                      TP4FLG - 004 TRAP FLAGS
                        ;*
                        ;* OUTPUTS:             DATA AT (DMTSTA) SAVED OR WRITTEN
                        ;*                      PAR #6 - CONTENTS SET TO CONTENTS OF DMTSTA
                        ;*                      [↑ is replaced with PAR5] $$$
                        ;*                      TP4FLG - CLEAR IF READ/WRITE SUCCESSFUL
                        ;*                               SET IF FAIL.
                        ;*
                        ;* CALLING SEQUENCE:            JSR     PC,DMRW
                        ;*
                        ;* COMENTS:             IF MEM MGT IS PRESENT THE SUBROUTINE USES (DMTSTA)
                        ;*                      AS THE PAGE ADDRESS , PLACING IT IN PAR #5, AND CREATES
                        ;*                      A VIRTUAL ADDR IN THE RANGE OF PAR #5 WHICH CONTAINS
                        ;*                      THE SIX LSB'S.
                        ;*                      IF IT IS NOT PRESENT THE (DMTSTA) IS CONVERTED INTO
                        ;*                      THE EQUIVALENT 16 BIT  PHYSICAL ADDRESS.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: CKTRAP,DM16B.
                        ;-- ************************************************************************

017466                  DMRW::  SAVE
017466  004537  005326                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

017472  010004                  MOV     R0,R4           ;SAVE THE SOURCE ADDR
017474  005737  002322          TST     MMPRES          ;IF MEM MGT IS PRESENT THEN
017500  001003                  BNE     6$              ;JUMP AND SET UP THE PAR #5 $$$
017502  004737  017414          JSR     PC,DM16B        ;OTHERWISE CONVERT DMTSTA INTO A 16 BIT
                                                        ;PHYSICAL ADDRESS, IN R0.
017506  000416                  BR      10$             ;JUMP TO PERFORM THE MOVE
017510  013777  002264  162622  6$:     MOV     DMTSTA,@PAR5A   ;SET PAR #5 $$$
017516  012700  120052          MOV     #120052,R0      ;SET THE SIX LSB'S AND CONVERT TO
                                                        ;A VIRTUAL ADDRESS WITHIN THE INFLUENCE
                                                        ;OF PAR #5. $$$
017522  032737  000001  002264          BIT     #1,DMTSTA       ;IF BIT #0 OF DMTSTA IS CLEAR THEN
017530  001402                  BEQ     8$              ;AVOID CHANGING THE LSB'S
017532  012700  120025          MOV     #120025,R0      ;CHANGE THE LSB'S
```

GLOBAL SUBROUTINE - DMRW - SEQ 0097

```
3663 017536  012777  000001  162552  8$:     MOV     #BIT0,@MMSR0    ;ENABLE MEM MGT.
3664 017544  005705                  10$:    TST     R5              ;IF A READ IS SPECIFIED THEN
3665 017546  001402                          BEQ     12$             ;AVOID SWAPING THE SOURCE AND DESTINATION.
3666 017550  010001                          MOV     R0,R1           ;SWAP
3667 017552  010400                          MOV     R4,R0           ;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     R0              ;INCREMENT THE SOURCE ADDR.
3672 017566  005303                          DEC     R3              ;DECREMENT THE DATA
3673 017570  001371                          BNE     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     @MMSR0          ;DISABLE IT.
3677 017604                          16$:    PASS
     017604  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3678 017606  000207                          RTS     PC
3679
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - DODMA -
                        ;** ************************************************************************
                        ;*                      - Initiate DMA Transmission Routine -
                        ;*      This routine writes the DMA parameter to the specified device and
                        ;*      initiates the DMA transmission.
                        ;*
                        ;* INPUTS:      R1 - Line number on which to initiate the DMA.
                        ;*              R2 - Start address of the DMA buffer (16 bit virtual).
                        ;*              R3 - Character count of the DMA buffer.
                        ;*              CSRA - Contains address of the DUT CSR.
                        ;*              IESTAT - Storage for states of the interrupt enable bits.
                        ;*              MMENAB - Memmory management flag (0 if MEM MGT not enabled).
                        ;*              HOST MEM MGT PAR REGISTERS - If MEM MGT is in use.
                        ;*              TXAD1A - Contains address of DMA TX buffer address reg #1.
                        ;*              TXAD2A - Contains address of DMA TX buffer address reg #2.
                        ;*              TXBFCA - Contains address of DMA character count register.
                        ;*
                        ;* OUTPUTS:     CARRY - Success flag (set if DMA_START found clear).
                        ;*              DUT TBUFFAD1 - LS 16 bits of DMA buffer address (initialized).
                        ;*              DUT TBUFFAD2 - MS 6 bits of DMA buffer address (initialized),
                        ;*                             DMA_START bit set.
                        ;*              DUT TBUFFCT - DMA buffer character count (initialized).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,DODMA
                        ;*
                        ;* COMMENTS:    This routine determines if Memory Management is being used
                        ;*              and sets up the full 22 bit physical address if necessary.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ************************************************************************

017610                  DODMA:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
017610  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017614  012704  000200          MOV     #200,R4                 ;PREPARE TO CLEAR UPPER 6 BITS OF DMA BUFF ADR.
017620  005737  002324          TST     MMENAB                  ;CHECK FOR MEMORY MANAGEMENT IN USE.
017624  001427                 BEQ     6$                      ;GOTO SET UP DEVICE IF MEM MGT NOT IN USE.
                        ;*
                        ; Memmory management is in use.
                        ; Construct 22 bit physical address from the 16 bit virtual address.
                        ;-
017626  010205          MOV     R2,R5                   ;STRIP THE MOST SIGNIFICANT 3 BITS OF THE
017630  012700  000005          MOV     #5,R0           ;  DMA BUFFER VIRTUAL ADDRESS AND MULTIPLY
017634  006105         2$:     ROL     R5              ;  THEIR VALUE BY TWO TO GET AN OFFSET INTO
017636  005300                 DEC     R0              ;  THE TABLE OF MEMMORY MANAGEMENT PAGE
017640  001375                BNE     2$              ;  ADDRESS REGISTERS (PAR).
017642  042705  177761          BIC     #177761,R5      ;
017646  063705  002326          ADD     PAR0A,R5        ;ADD IN THE BASE VALUE OF THE MM PAR REGISTERS.
017652  011505                  MOV     (R5),R5         ;GET THE 16 BIT PHYSICAL ADDRESS BLOCK COUNT.
017654  012700  000006          MOV     #6,R0           ;SHIFT UPPER 6 BITS OF THE PHYSICAL ADDRESS
017660  006305         4$:     ASL     R5              ;  BLOCK COUNT (GOTTEN FROM THE PROPER PAR)
017662  006104                  ROL     R4              ;  INTO THE LS 6 BITS OF THE WORD TO WRITE
017664  005300                  DEC     R0              ;  INTO THE DUT TBUFFAD2 REGISTER.
017666  001374                 BNE     4$              ;
017670  042702  160000          BIC     #160000,R2      ;ADD THE 13 BIT DISPLACEMENT FIELD FROM VIRTUAL
017674  060502                  ADD     R5,R2           ;  ADR TO THE SHIFTED BLOCK NUMBER FROM THE
017676  005504                  ADC     R4              ;  MEMMORY MANAGEMENT PAR.
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$:     MFPS    R5              ;GET THE PRESENT PROCESSOR PRIORITY.
 3747 017706  106427  000340                   MTPS    #PRI07          ;DISABLE ALL HARDWARE INTERRUPTS.
 3748 017712  053701  002234                    BIS    IESTAT,R1       ;PREPARE FOR SETUP OF LINE NUMBER IN DUT CSR.
 3749 017716  010177  162260                    MOV    R1,@CSRA        ;SET UP THE DUT CSR IND.ADR.REG FIELD.
 3750 017722  105777  162270                    TSTB   @TXAD2A         ;TEST THE DUT DMA_START BIT.
 3751 017726  000241                            CLC                    ;INDICATE FAILURE IN CASE DMA.HO BIT IS SET.
 3752 017730  100410                           BMI     60$             ;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 PREG05 SUBRT.
 3760 017754  000207                           RTS     PC              ;  CARRY - SUCCESS FLAG (SET IF SUCCESS).
```

GLOBAL SUBROUTINE - FINACT - SEQ 0100

```
                        .SBTTL  GLOBAL SUBROUTINE               - FINACT -
                        ;** ****************************************************************************
                        ;*                          - FIND FIRST ACTIVE LINE -
                        ;*      This subroutine calculates the number of the first active line that
                        ;*      is found in the active line bit map ACTLNS.
                        ;*
                        ;* INPUTS:      ACTLNS - Contains the active line bit map.
                        ;*
                        ;* OUTPUTS:     R1 - Contains the number of the first active line.
                        ;*              R5 - Contains the bit map representation of the active line.
                        ;*              Carry set indicates success.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,FINACT
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ****************************************************************************

017756                  FINACT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017756  004537  005326                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                            ;+
                            ; Find an active line on which to perform the test.
                            ;-
017762  005001                  CLR     R1                      ;CLEAR THE LINE NUMBER COUNTER.
017764  012703  000010          MOV     @NUMLNS,R3              ;GET MAX LINE NUMBER.
017770  013700  002174          MOV     ACTLNS,R0               ;GET THE ACTIVE LINE BIT MAP.
017774  012705  000001          MOV     #1,R5                   ;SET UP A LINE BIT MASK.
020000  030500          2$:     BIT     R5,R0                   ;LOOK FOR AN ACTIVE LINE.
020002  001006                  BNE     4$                      ;BRANCH TO BEGIN TEST IF A LINE HAS BEEN FOUND.
020004  006305                  ASL     R5                      ;SHIFT THE BIT MASK FOR THE NEXT LINE.
020006  005201                  INC     R1                      ;INCREMENT THE LINE NUMBER COUNTER.
020010  020103                  CMP     R1,R3                   ;CHECK IF ALL LINES HAVE BEEN TRIED.
020012  002772                  BLT     2$                      ;LOOP TO TRY THE NEXT LINE.
020014  000241                  CLC                             ;CLEAR CARRY BIT, NO ACTIVE LINE FOUND.
020016  000401                  BR      60$                     ;EXIT WITH FAILURE.
020020  000261          4$:     SEC                             ;SET CARRY, SUCCESS.

020022                  60$:    PASS    R1,R5
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.
                                                    ;R1 - CONTAINS THE NUMBER OF FIRST ACTIVE LINE.
                                                    ;R5 - CONTAINS THE BIT MAP OF THE ACTIVE LINE.
                                                    ;CARRY - SET INDICATES SUCCESS.
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.
```

GLOBAL SUBROUTINE - FRPSUP - SEQ 0102

```
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$,CBLPRA      ;SET-UP COMPLEMENTARY LPR PARM.
3901 020210  004737  026114               JSR     PC,TXRINI       ;INITIALISE INACTIVE LINES.
3902                                    ;+
3903                                    ; Initialise LPR parameters for inactive lines in GROUP 1.
3904                                    ;-
3905 020214  013702  002174                MOV     ACTLNS,R2       ;GET THE ACTIVE (TX) LINE BIT MAP.
3906 020220  005102                        COM     R2              ;GENERATE A BIT MAP OF INACTIVE LINES.
3907 020222  040102                        BIC     R1,R2           ;CLEAR ANY NONE EXISTANT INACTIVE LINES.
3908 020224  005104                        COM     R4              ;
3909 020226  040402                        BIC     R4,R2           ;ONLY PASS LGRP2 ASSOCIATED LINE BIT MAP.
3910 020230  010237  003136                MOV     R2,CBMAPA       ;SET-UP BIT MAP IN CONTROL BLOCK.
3911 020234  013737  020276  003124        MOV     70$,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
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - GETBDR -
                                ;** ****************************************************************************
                                ;*                        - Get Baudrate Subroutine -
                                ;*      This routine requests a baudrate input from the operator.  This
                                ;*      baudrate is looked up in a table to give the LPR baudrate field
                                ;*      value which is associated with that baudrate.
                                ;*
                                ;* INPUTS:      BDRMSG - Label at the baudrate prompt message.
                                ;*              BRTBLE - Label after end of the baudrate table.
                                ;*              UBRFMT - Label at the unsupported baudrate message.
                                ;*
                                ;* OUTPUTS:     R1 - Baudrate code in LS 4 bits.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,GETBDR
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;-- ****************************************************************************

020302                          GETBDR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020302  004537  005326                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020306  013705  002266                  MOV     GMANWD,R5               ;SAVE THE GMAINIx VALUE.
                                ;+
                                ; Prompt the operator: "MODEM BAUDRATE IN BPS:  (D) 1200 ?"
                                ;-
020312  012737  002260  002266  2$:     MOV     #1200.,GMANWD           ;SET UP DEFAULT VALUE TO 1200 BAUD.
020320                                 GMANID  BDRMSG,GMANWD,D,177777,0,38400.,YES
020320  104443                                                                          TRAP    C$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$HILIM
020340                                                                          10000$:
020340  013702  002266                  MOV     GMANWD,R2
                                ;+
                                ; Attempt to look the value up in the baudrate table.
                                ;-
020344  012701  000017                  MOV     #15.,R1                 ;INITIALIZE BAUDRATE CODE TO HIGHEST BAUDRATE.
020350  012703  002466                  MOV     #BRTBLE,R3              ;INITIALIZE BAUDRATE POINTER.

020354  020243                  4$:     CMP     R2,-(R3)                ;COMPARE BAUDRATE WITH A TABLE ENTRY.
020356  001416                          BEQ     60$                     ;BAUDRATES COMPARE?  YES, EXIT WITH CODE.
020360  005301                          DEC     R1                      ;NO, SET BAUDRATE CODE TO NEXT LOWER BAUDRATE.
020362  001374                          BNE     4$                      ;DONE?  NO, LOOP.

020364  020243                          CMP     R2,-(R3)                ;CHECK IF LAST BAUDRATE MATCHES.
020366  001412                          BEQ     60$                     ;BAUDRATES MATCH?  YES, EXIT WITH CODE.

                                ;+
                                ; Report "nnnnn IS NOT A SUPPORTED BAUDRATE, ENTER ANOTHER OR CTRL C."
                                ;-
020370                                  PRINTF  #UBRFMT,R2
020370  010246                                                                          MOV     R2,-(SP)
```

GLOBAL SUBROUTINE - GETBDR - SEQ 0105

```
        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 PREG05 SUBRT.
  3983  020426  000207                      RTS     PC                 ;  R1 - BAUDRATE CODE.
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - GETCHR -
                        ;** ************************************************************************
                        ;*              - Get a Character From the RX Buffer Routine -
                        ;*      This subroutine gets a character from the RX buffer which is in the
                        ;*      host system memory.  If the buffer is empty upon entry of this routine
                        ;*      this routine returns a null character with DATA.VALID clear and a
                        ;*      buffer empty indication.
                        ;*
                        ;* INPUTS:      RXBCNT - RX buffer character count.
                        ;*              RXBEND - Label after end of the RX buffer area in memory.
                        ;*              RXBETX - Equated to RX buffer level at which to enable TX.
                        ;*              RXBOPT - Pointer to next available input slot of RX buffer.
                        ;*              RXBSTA - Label at start of RX buffer area in memory.
                        ;*
                        ;* OUTPUTS:     R2 - Character which is read from the buffer.
                        ;*              RXBOPT - Updated to point to next input slot of RX buffer.
                        ;*              RXBCNT - RX buffer character count (Updated).
                        ;*              CARRY - "Success" flag (Set if buffer is not empty on entry).
                        ;*
                        ;* CALLING SEQUENCE:    JSP     PC,GETCHR
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ************************************************************************

020430                  GETCHR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020430  004537  005326                 JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
020434  005000                  CLR     R0                      ;CLEAR THE "RE-ENABLE" TX FLAG (SUBRTN OUTPUT).
020436  005002                  CLR     R2                      ;GET NULL CHAR IN CASE BUFFER IS EMPTY.
020440  005737  002720          TST     RXBCNT                  ;CHECK FOR RX BUFFER EMPTY, CLEAR CARRY.
020444  001416                 BEQ     60$                     ;EXIT THE ROUTINE IF BUFFER IS EMPTY.
020446  013704  002714          MOV     RXBOPT,R4               ;GET THE BUFFER OUTPUT POINTER.
020452  011402                  MOV     (R4),R2                 ;GET A CHARACTER FROM THE BUFFER.
020454  005024                  CLR     (R4)+                   ;DELETE THE READ CHARACTER FROM THE BUFFER.
020456  020427  003122          CMP     R4,@RXBEND              ;CHECK IF POINTER SHOULD WRAP AROUND.
020462  103402                 BLO     2$                      ;SKIP WRAPAROUND IF POINTER IS NOT AT END.
020464  012704  002722          MOV     @RXBSTA,R4              ;WRAP INPUT POINTER AROUND.
020470  010437  002714  2$:     MOV     R4,RXBOPT               ;UPDATE THE OUTPUT POINTER STORAGE.

020474  005337  002720          DEC     RXBCNT                  ;REMOVE THIS CHAR FROM THE BUFFER COUNT.
020500  000261                  SEC                             ;SET SUCCESS FLAG, BUFFER WAS NOT EMPTY.

020502  010266  000006  60$:     PASS    R2                      ;RESTORE GPRS, EXCEPT
                                        MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
020506  004736                  JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
                                ;R2 - CONTAINS THE CHARACTER READ FROM BUFFER.
                                ;CARRY-'SUCCESS" FLAG, SET IF BUFFER NOT EMPTY.
020510  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - GETLP1 -
                        ;++ ******************************************************************************
                        ;*              - Get Line Parameters Routine Number One -
                        ;*      This routine is used to repeatedly get combinations of line parameter
                        ;*      contents for the Single Character Mode TX/RX Test (short data pattern).
                        ;*      Each time this routine is called it gets another combination of the
                        ;*      paramters in the parameter tables until all combinations have been
                        ;*      returned at which point it returns a "failure" indication.
                        ;*
                        ;* INPUTS:      Single character mode, short data pattern TX/RX tables:
                        ;*                      SCBCT - Number of bits per char table (4 entries).
                        ;*                      SCBRT - Baudrates table (3 entries).
                        ;*                      SCNST - Number of stop bits bits table (2 entries).
                        ;*                      SCTPT - Type of parity table (3 entries).
                        ;*              Each table has a base and end label consisting of the name of
                        ;*              the table with a "B" and "E" appended respectively.
                        ;*              R1 thru R4 - Pointers into SCBCT thru SCTPT tables respectivly.
                        ;*                  R1 is clear if this is the first call of GETLP1.
                        ;*
                        ;* OUTPUTS:     R0   Composed LPR contents, clear if failure (Done).
                        ;*              R1 thru R4 - Table pointers (Updated).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,GETLP1
                        ;*
                        ;* COMMENTS:    This routine should be used in congunction with a SWAPx
                        ;*              routine to avoid destroying the GPR contents.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ******************************************************************************

020512                  GETLP1:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020512  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020516  005701          TST     R1      ;TEST FOR THIS BEING FIRST CALL OF GETLP1.
020520  001010          BNE     2$      ;SKIP ORIGINAL SET UP IF NOT FIRST CALL.
020522  012701  005044  MOV     #SCBCTB,R1      ;INITIALIZE BITS PER CHAR TABLE POINTER.
020526  012702  005054  MOV     #SCBRTB,R2      ;INITIALIZE BAUDRATE TABLE POINTER.
020532  012703  005062  MOV     #SCNSTB,R3      ;INITIALIZE # OF STOP BITS TABLE POINTER.
020536  012704  005066  MOV     #SCTPTB,R4      ;INITIALIZE TYPE OF PARITY TABLE POINTER.

020542  020427  005074  2$:     CMP     R4,#SCTPTE      ;CHECK FOR POINTER AT END OF TABLE.
020546  103425          BLO     4$      ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
020550  012704  005066  MOV     #SCTPTB,R4      ;RESET POINTER TO BEGINNING OF TABLE.
020554  005723          TST     (R3)+   ;INC THE # OF STOP BITS TABLE POINTER BY 2.
020556  020327  005066  CMP     R3,#SCNSTE      ;CHECK FOR POINTER AT END OF TABLE.
020562  103417          BLO     4$      ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
020564  012703  005062  MOV     #SCNSTB,R3      ;RESET POINTER TO BEGINNING OF TABLE.
020570  005722          TST     (R2)+   ;INC BAUD RATES TABLE POINTER BY 2.
020572  020227  005062  CMP     R2,#SCBRTE      ;CHECK FOR POINTER AT END OF TABLE.
020576  103411          BLO     4$      ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
020600  012702  005054  MOV     #SCBRTB,R2      ;RESET POINTER TO BEGINNING OF TABLE.
020604  005721          TST     (R1)+   ;INC THE BITS PER CHAR TABLE POINTER BY 2.
020606  020127  005054  CMP     R1,#SCBCTE      ;CHECK FOR POINTER AT END OF TABLE.
020612  103403          BLO     4$      ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
020614  005000          CLR     R0      ;PREPARE TO PASS OUT CLEAR LPR FIELDS.
020616  000241          CLC             ;INDICATE "FAILURE" FOR EXIT.
020620  000405          BR      60$     ;EXIT WITH "FAILURE", WE'RE DONE.
```

GLOBAL SUBROUTINE - GETLP1 - SEQ 0108

```
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.
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - GETLP2 -
                        ;** ************************************************************************
                        ;*              - Get Line Parameters Routine Number Two -
                        ;*      This routine is used to repeatedly get combinations of line parameter
                        ;*      contents for the Single Character Mode TX/RX Test (long data pattern).
                        ;*      Each time this routine is called it gets another combination of the
                        ;*      paramters in the paramter tables until all combinations have been
                        ;*      returned at which point it returns a "failure" indication.
                        ;*
                        ;* INPUTS:      Single character mode, short data pattern TX/RX tables:
                        ;*                      SCBCT - Number of bits per char table (4 entries).
                        ;*                      SCNST - Number of stop bits bits table (2 entries).
                        ;*                      SCTPT - Type of parity table (3 entries).
                        ;*              Each table has a base and end label consisting of the name of
                        ;*              the table with a "B" and "E" appended respectively.
                        ;*              R1 thru R3 - Pointers into SCBCT, SCNST, SCTPT tables
                        ;*                  R1 is clear if this is the first call of GETLP2.
                        ;*
                        ;* OUTPUTS:     R0 - Composed LPR contents, clear if failure (Done),
                        ;*                  38.4K baudrate is selected.
                        ;*              R1 thru R3 - Table pointers (Updated).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,GETLP2
                        ;*
                        ;* COMMENTS:    This routine should be used in congunction with a SWAPx
                        ;*              routine to avoid destroying the GPR contents.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;  ************************************************************************

020664                  GETLP2:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020664  004537  005326                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020670  005701                          TST     R1              ;TEST FOR THIS BEING FIRST CALL OF GETLP2.
020672  001006                          BNE     2$              ;SKIP ORIGINAL SET UP IF NOT FIRST CALL.
020674  012701  005044                  MOV     #SCBCTB,R1      ;INITIALIZE BITS PER CHAR TABLE POINTER.
020700  012702  005062                  MOV     #SCNSTB,R2      ;INITIALIZE # OF STOP BITS TABLE POINTER.
020704  012703  005066                  MOV     #SCTPTB,R3      ;INITIALIZE TYPE OF PARITY TABLE POINTER.

020710  020327  005074          2$:     CMP     R3,#SCTPTE      ;CHECK FOR POINTER AT END OF TABLE.
020714  103417                          BLO     4$              ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
020716  012703  005066                  MOV     #SCTPTB,R3      ;RESET POINTER TO BEGINNING OF TABLE.
020722  005722                          TST     (R2)+           ;INC THE # OF STOP BITS TABLE POINTER BY 2.
020724  020227  005066                  CMP     R2,#SCNSTE      ;CHECK FOR POINTER AT END OF TABLE.
020730  103411                          BLO     4$              ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
020732  012702  005062                  MOV     #SCNSTB,R2      ;RESET POINTER TO BEGINNING OF TABLE.
020736  005721                          TST     (R1)+           ;INC BAUD RATES TABLE POINTER BY 2.
020740  020127  005054                  CMP     R1,#SCBCTE      ;CHECK FOR POINTER AT END OF TABLE.
020744  103403                          BLO     4$              ;GO GET LPR CONTENTS IF NOT AT END OF TABLE.
020746  005000                          CLR     R0              ;PREPARE TO PASS OUT CLEAR LPR FIELDS.
020750  000241                          CLC                     ;INDICATE "FAILURE" FOR EXIT.
020752  000406                          BR      60$             ;EXIT WITH "FAILURE", WE'RE DONE.

020754  012700  177400          4$:     MOV     #177400,R0      ;SET BAUD RATE FIELDS FOR 38.4 K BAUD.
020760  051100                          BIS     (R1),R0         ;GET THE BITS/CHAR FIELD OF NEW LPR CONTENTS.
020762  051200                          BIS     (R2),R0         ;INCLUDE THE NUMBER OF STOP BITS FIELD.
020764  052300                          BIS     (R3)+,R0        ;INCLUDE THE TYPE OF PARITY FIELD.
```

GLOBAL SUBROUTINE - GETLP2 - SEQ 0110

```
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 PREGOS SUBRT.
4157 021012  000207                    RTS     PC              ;  R1 THRU R3 - POINTERS, R0 - NEW LPR FIELDS.
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - GETTIM -
                        ;** ****************************************************************************
                        ;*              - Get Time-out Value Based on Minimum Baudrate Routine -
                        ;*      This subroutine gets the necessary time-out value to verify that all
                        ;*      chars have been received at the completion of the TX/RX of a data
                        ;*      pattern.  This uses the slowest baudrate which is specified in the
                        ;*      passed in DUT LPR contents to calculate this time-out value.
                        ;*
                        ;* INPUTS:      R1 - DUT LPR contents.
                        ;*
                        ;* OUTPUTS:     RXTOUT - Time-out value for waiting for last RX char.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,GETTIM
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ****************************************************************************

021014                  GETTIM:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021014  004537  005326                       JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021020  000301                   SWAB    R1              ;PUT THE BAUD RATE FIELDS IN THE LOW BYTE.
021022  042701  177400           BIC     #177400,R1      ;CLEAR STOP,PARITY,AND CHAR FIELDS.
021026  010102                   MOV     R1,R2           ;COPY BAUD RATE FIELDS.
021030  042701  000360           BIC     #360,R1         ;SELECT RX BAUD RATE FIELD ONLY.
021034  006202                   ASR     R2              ;SHIFT TX BAUD RATE FIELD
021036  006202                   ASR     R2              ; TO OCCUPY THE LOW FOUR BYTES.
021040  006202                   ASR     R2              ;
021042  006202                   ASR     R2              ;
021044  020102                   CMP     R1,R2           ;CHECK IF SAME BAUD RATE IN EACH FIELD.
021046  101401                  BLOS    2$              ;BRANCH IF RX BAUD RATE IS LOWER OR SAME.
021050  010201                   MOV     R2,R1           ;TX BAUD RATE IS THE SLOWER OF THE TWO.
021052  116102  005216   2$:     MOVB    PROTBL(R1),R2   ;GET PROPORTIONAL DELAY FROM TABLE.
021056  042702  177400           BIC     #177400,R2      ;CLEAR UPPER BYTE BECAUSE OF SIGN EXTENSION.
021062  010237  002242           MOV     R2,RXTOUT       ;LOAD THE RX TIME OUT VARIABLE.

021066                  60$:    PASS                            ;RESTORE GPRS.
021066  004736                           JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
021070  000207                   RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - INICHR -
                        ;** ****************************************************************************
                        ;*                      - Send Initial Characters Routine -
                        ;*      This routine is used to initiate single character transmission.
                        ;*      This routine sends the initial characters to each active lines to
                        ;*      cause future TX interrupts which will continue the transmission if
                        ;*      more than one character is to be sent to each active line.
                        ;*
                        ;* INPUTS:      ACTLNS - Bit map of active DUT lines.
                        ;*              BITTBL - Label of table of words each with a bit set.
                        ;*              CSRA - Contains the address of the DUT CSR.
                        ;*              DPENDB - Base of the data pattern end table (entry per line).
                        ;*              DPLENB - Base of the data pattern length table.
                        ;*              IBM - Bit mask of inactive TX/RX bits.
                        ;*              IESTAT - States of DUT int enable bits (Other bits clear).
                        ;*              NUMLNS - Equated to the number of lines on the DUT.
                        ;*              TXCHRA - Contains the address of the DUT TXCHAR register.
                        ;*              TXCNTB - Lable at base of the TX character counter table.
                        ;*              TXPTRB - Label at base of the TX data pattern pointers table.
                        ;*
                        ;* OUTPUTS:     CSR - DUT CSR IND.ADR.REG field is destroyed.
                        ;*              TXCHAR - DUT TXCHAR has word written to it.
                        ;*              TXCNTx - Counters incremented for lines on which chars sent.
                        ;*              TXPTRB - Each pointer in table points to next TX char for line.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,INICHR
                        ;*
                        ;* COMMENTS:    This routine assumes that at least one character should be
                        ;*              transmitted on each active line.
                        ;*              Interrupts must be disabled when calling this routine.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ****************************************************************************
021072                  INICHR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021072  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021076  013701  002234          MOV     IESTAT,R1               ;GET STATE OF TX.IE, RX.IE FOR USE IN SETTING
                                                                ;  UP THE IND.ADR.REG FIELD OF THE DUT CSR.
021102  005002                  CLR     R2                      ;SET LINE NUMBER OFFSET TO LINE 0.
021104  036237  002366  002174  2$:     BIT     BITTBL(R2),ACTLNS  ;TEST THE ACTIVE LINES BIT FOR THIS LINE.
021112  001424                  BEQ     6$                      ;DON'T TX ON THIS LINE IF IT IS NOT ACTIVE.
021114  010177  161062          MOV     R1,@CSRA                ;SET UP THE IND.ADR.REG FIELD OF THE CSR.
021120  016205  003344          MOV     TXPTRB(R2),R5           ;GET THE TX DATA PATTERN POINTER FOR THIS LINE.
021124  112504                  MOVB    (R5)+,R4                ;GET THE CHAR TO TX ON THIS LINE, INC POINTER.
021126  020562  003144          CMP     R5,DPENDB(R2)           ;COMPARE POINTER WITH DATA PATTERN END ADR.
021132  103402                  BLO     4$                      ;SKIP POINTER WRAPAROUND IF NOT AT PATTERN END.
021134  166205  003204          SUB     DPLENB(R2),R5           ;WRAP TX POINTER AROUND TO BEGINNING OF PAT'N.
021140  010562  003344          4$:     MOV     R5,TXPTRB(R2)   ;UPDATE THE TX POINTER STORAGE TABLE FOR LINE.
021144  043704  002226          BIC     IBM,R4                  ;CLEAR INACTIVE BITS OF TX CHARACTER WORD.
021150  052704  100000          BIS     #BIT15,R4               ;SET THE TX.DATA.VALID BIT IN THE WORD.
021154  010477  161024          MOV     R4,@TXCHA               ;TX THE FIRST CHARACTER FOR THIS LINE.
021160  005262  003504          INC     TXCNTB(R2)              ;INCREMENT TX CHARACTER COUNTER FOR THIS LINE.
021164  005201                  6$:     INC     R1              ;INCREMENT WORD FOR IND.ADR.REG FIELD SET UP.
021166  062702  000002          ADD     #2,R2                   ;SET LINE NUMBER OFFSET TO NEXT LINE.
021172  020227  000020          CMP     R2,#NUMLNS*2            ;COMPARE LINE OFFSET WITH TWICE THE # OF LINES.
021176  002742                  BLT     2$                      ;LOOP TO SEND CHAR TO ANOTHER LINE IF NOT DONE.

021200                  60$:    PASS                            ;RESTORE GPRS.
```

GLOBAL SUBROUTINE - INICHR - SEQ 0113

```
        021200  004736                          JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
   4253 021202  000207          RTS     PC
```

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

GLOBAL SUBROUTINE - INIDMA - SEQ 0115

```
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 PREGOS SUBRT.
4330 021274  000207                        RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - LINBIT -
                        ;++ ******************************************************************************
                        ;*              - Line Number to Bit Map conversion subroutine -
                        ;*      This subroutine is used to generate a bit map (one bit of 16 set)
                        ;*      based on a line number (range: 1 to 16).  Only the LS 4 bits of the
                        ;*      line number word are used, the others are masked out (so unmasked
                        ;*      MSBytes of DUT CSRs can be passed to this routine without error).
                        ;*
                        ;* INPUTS:      R1 - Line number (only LS 4 bits used, others disregarded).
                        ;*              BITTBL - Base label of a 16 word bit table.
                        ;*
                        ;* OUTPUTS:     R0 - Bit map, bit corresponding to line number is set:
                        ;*                      If line number is 3, then bit3 is set, etc.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,LINBIT
                        ;*
                        ;* COMMENTS:    No checking is performed to verify that the line number is
                        ;*              a legal line number for the DUT (ie - less than NUMLNS).
                        ;*              NOTE: The line number is not destroyed of altered, so this
                        ;*                      routine can be used easily in loops.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ******************************************************************************

021276                  LINBIT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021276  004537  005326          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021302  042701  177760          BIC     #177760,R1      ;MASK OUT ALL BUT 4 LSBITS OF THE LINE #.
021306  006301                  ASL     R1              ;MULTIPLY LINE # BY 2 TO GET WORD TABLE OFFSET.
021310  016100  002366          MOV     BITTBL(R1),R0   ;GET THE SINGLE BIT BIT MAP.
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.
021322  000207                  RTS     PC                ;R0 - BIT MAP WITH LINE # BIT SET.
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - MODSUP -
                        ;++ ************************************************************************
                        ;*                      - MODEM LOOPBACK TX/RX SET-UP ROUTINE -
                        ;*
                        ;*              This routine is used to initialise both the DUT and the
                        ;*              transmission/reception control parameters to the correct
                        ;*              state, prior to a Modem Loopback test data pattern TX/RX.
                        ;*
                        ;* INPUTS:      R1 - TX, RX LPR contents.
                        ;*              R2 - Start address of data pattern to transmit.
                        ;*              R3 - Length of data pattern.
                        ;*              ACTLNS - Contains a bit map of all currently active lines.
                        ;*              CBB - Label at base of TX/RX control block.
                        ;*
                        ;* OUTPUTS:     The contents of the TX/RX control block (CCB) are destroyed.
                        ;*              The indirect address field of the DUT CSR may be destroyed.
                        ;*              The DUT's LPR's and LNC's may be modified.
                        ;*              The following pointers and counters are initialised;
                        ;*              CHCNT,CHRTOT,DPEND,DPLEN,EXCNT,RXCNT,RXPTR,TXCNT,
                        ;*              TXPTR,TXRXL.
                        ;*              CHRTOT, RXDONF, TXDONF and TXINTF are cleared.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,MODSUP
                        ;*
                        ;* COMMENTS:    DUT is set up with DSR and DTR set.  One data pattern is
                        ;*              sent and received from each line.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: CONMAP,RXENBL,TXRINI.
                        ;-- ************************************************************************

021324                  MODSUP:: SAVE                           ;SAVE CONTENTS OF THE GPR'S R0 THRU R5.
021324  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021330  005037  002500          CLR     CHRTOT                  ;CLEAR TOTAL RECEIVED CHAR COUNTER.
021334  005037  002252          CLR     TXINTF                  ;CLEAR FLAGS USED TO LOG DMA H.OVER ERRORS.
021340  005037  002504          CLR     TXDONF                  ;CLEAR THE TX DONE FLAGS.
021344  005037  002506          CLR     RXDONF                  ;CLEAR THE RX DONE FLAGS.
                        ;+
                        ; Set up the Transmission/Reception Control block to the desired state.
                        ;-
021350  010137  003124          MOV     R1,CBB                  ;SET CONTENTS OF LPR PARAMS IN TX/RX C.BLK.
021354  012701  003124          MOV     #CBB,R1                 ;GET BASE ADDRESS OF CONTROL BLOCK.
021360  005201                  INC     R1                      ;INCREMENT ADDRESS FOR NEXT WORD
021362  005201                  INC     R1                      ;INITIALISE THE FOLLOWING IN THE CNTRL.BLK:
021364  012721  011004          MOV     #11004,(R1)+            ; LNCTRL: RTS, DTR, ENABLE RECEIVERS.
021370  010221                  MOV     R2,(R1)+                ; START ADDRESS OF DATA PATTERN.
021372  010321                  MOV     R3,(R1)+                ; DATA PATTERN LENGTH.
021374  012721  000001          MOV     #1,(R1)+                ; NUMBER OF DATA PATTERNS TO TRANSMIT.
021400  013721  002174          MOV     ACTLNS,(R1)+            ; BIT MAP OF LINES TO INITIALISE.
021404  112721  000003          MOVB    #3,(R1)+                ;SET LOOPBACK MODE TO M325.
021410  005201                  INC     R1                      ;INCREMENT ADDRESS FOR THE NEXT WORD.
021412  012711  000002          MOV     #2,(R1)                 ;SET AMOUNT OF OFFSET EACH TX STARTS AT TO 2.
                        ;+
                        ; Initialise the DUT and the associated pointers and counters, to the state
                        ; dictated by the contents of the TX/RX control block.
                        ;-
021416  004737  026114          JSR     PC,TXRINI               ;INITIALISE DUT.
                        ;+
```

GLOBAL SUBROUTINE - MODSUP - SEQ 0118

```
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 PREGOS SUBRT.
4432 021462  000207                            RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - MSLGET -
                        ;*************************************************************************
                        ;*      - Milli Seconds Loop which returns read word and remaining time -
                        ;*      This subroutine is a general purpose test loop subroutine.  It is used
                        ;*      to verify that a certain action occurs before a time-out period.  The
                        ;*      calling routine passes in which bits should be set and cleared for the
                        ;*      desired condition and the time-out value in milli-seconds.
                        ;*      This routine checks for the desired condition upon entrance into the
                        ;*      routine and then once each milli-second there after.
                        ;*      Upon return, the last word which was read to check for the condition
                        ;*      is returned by this subroutine.
                        ;*
                        ;* INPUTS:      R1 - Time-out value in milli-seconds (up to 64K ms).
                        ;*              R2 - Bit map of bits to test (1 indicates to test the bit).
                        ;*              R3   Desired states of the indicated fields in R2.
                        ;*              R4   Address of the word to test.
                        ;*              MSLCNT - Milli second software loop count.
                        ;*
                        ;* OUTPUTS:     R0 - The last word which was read to check for the condition.
                        ;*              R1 - Remaining number of ms in time-out time.
                        ;*              CARRY - Success flag (set if condition is met before time out).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,MSLGET
                        ;*
                        ;* COMMENTS:    This routine works with or without a hardware clock, but the
                        ;*              calibration is only guarenteed when a line clock is available
                        ;*              on the system.
                        ;*              This routine can be used as a delay routine, by specifying the
                        ;*              desired delay as the time-out and specifying a condition to
                        ;*              look for which will not be met during the delay.
                        ;*              If a time-out value of 0 is specified, this routine checks for
                        ;*              the desired condition before returning.  It indicates success
                        ;*              if the condition is met, failure otherwise.
                        ;*
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;*************************************************************************

021464                  MSLGET:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021464  004537  005326                          JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; Set up mask for removing unused bits in the test word, and clear unused
                        ; bits in the desired state word to allow direct comparison.
                        ;
021470  005102                  COM     R2              ;GET MASK OF UNUSED BITS.
021472  040203                  BIC     R2,R3           ;MASK OUT UNUSED BITS IN DESIRED STATE WORD.
                        ;*
                        ; Handle the test and exit if we have a 0 time-out value.
                        ;-
021474  005701                  TST     R1              ;TEST THE TIME-OUT VALUE FOR ZERO.
021476  001C11                  BNE     2$              ;IF NON-ZERO TIME OUT, GO LOOP AND TEST.
021500  011400                  MOV     (R4),R0         ;GET THE WORD TO TEST BEFORE EXITING.
021502  010037  021576          MOV     R0,6$           ;SAVE VALUE SO WE CAN RETURN IT.
021506  040200                  BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
021510  020003                  CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
021512  000261                  SEC                     ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
021514  001420                  BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
```

GLOBAL SUBROUTINE MSLGET - SEQ 0120

```
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$:    PASS    R0,R1           ;RESTORE GPRS, EXCEPT THE FOLLOWING:
     021562  010066  000002                        MOV     R0,R0SLOT(SP)           ;PUT R0 IN STACK SLOT.
     021566  010166  000004                        MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
     021572  004736                                JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
4513                                                 ;R0 - LAST READ WORD CHECKED FOR CONDITION.
4514                                                 ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
4515 021574  000207                        RTS     PC              ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
4516                                ;+
4517                                ; Local storage.
4518                                ;
4519 021576  000000                62$:    .WORD   0               ;STORAGE FOR THE LAST READ WORD.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                       MSLOOP -
                        ;*******************************************************************************
                        ;*      - Test Loop subroutine -
                        ;*      This subroutine is a general purpose test loop subroutine.  It is used
                        ;*      to verify that a certain action occurs before a time-out period.  The
                        ;*      calling routine passes in which bits should be set and cleared for the
                        ;*      desired condition and the time-out value in milli-seconds.
                        ;*      This routine checks for the desired condition upon entrance into the
                        ;*      routine and then once each milli-second thereafter.
                        ;*
                        ;* INPUTS:      R1 - Time-out value in milli-seconds (up to 64K ms).
                        ;*              R2 - Bit map of bits to test (1 indicates to test the bit).
                        ;*              R3 - Desired states of the indicated fields in R2.
                        ;*              R4 - Address of the word to test.
                        ;*              MSLCNT - Milli second software loop count.
                        ;*
                        ;* OUTPUTS:     CARRY - Success flag (set if condition is met before time-out).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,MSLOOP
                        ;*
                        ;* COMMENTS:    This routine works with or without a hardware clock, but the
                        ;*              calibration is only guarenteed when a line clock is available
                        ;*              on the system.
                        ;*              This routine can be used as a delay routine, by specifying the
                        ;*              desired delay as the time-out and specifying a condition to
                        ;*              look for which will not be met during the delay.
                        ;*              If a time-out value of 0 is specified, this routine checks for
                        ;*              the desired condition before returning.  It indicates success
                        ;*              if the condition is met, failure otherwise.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: MSLGET.
                        ;*******************************************************************************

021600                  MSLOOP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021600  004537  005326                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

                        ;+
                        ; Calling the MSLGET routine from the MSLOOP routine isolates the caller of
                        ; MSLOOP from the returned test word and remaining time-out values.
                        ;-
021604  004737  021464          JSR     PC,MSLGET       ;CALL THE MULTI PURPOSE MS LOOP AND SEARCH RTN.

021610                  60$:    PASS                            ;RESTORE GPRS,
021610  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
021612  000207          RTS     PC                        ;CARRY - SET IF SUCCESS, CLEAR IF TIME OUT.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                MSSRPT -
                        ;** ************************************************************************
                        ;*                      - Modem Status Signal Report Routine -
                        ;*      This subroutine is used to report the states of the modem status
                        ;*      signals for all active lines.
                        ;*
                        ;* INPUTS:      ACTLNS - Bit map of active lines.
                        ;*              CSRA - Contains address of the DUT CSR.
                        ;*              EF9101 - Label at format statement for blank line.
                        ;*              IESTAT - Contains states of the DUT Interrupt Enable bits.
                        ;*              STATA - Contains address of the DUT STAT register.
                        ;*              NUMLNS - Equated to the number of lines on the device.
                        ;*
                        ;* OUTPUTS:     DUT CSR IND.ADR.REG Field - Contents destroyed.
                        ;*              Report messages are printed on the operator's console.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,MSSRPT
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ************************************************************************

021614                  MSSRPT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5
021614  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; Print the basic modem status message.
                        ;    "MODEM STATUS SIGNAL REPORT:"
                        ;-
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

021640  005001                  CLR     R1              ;START WITH LINE 0.
021642  012702  000001          MOV     #1,R2
021646  013703  002202          MOV     CSRA,R3         ;GET THE CSR ADDRESS.
021652  013704  002234          MOV     IESTAT,R4       ;GET THE STATES OF THE INTERRUPT ENABLE BITS.
021656  013705  002174          MOV     ACTLNS,R5       ;GET THE ACTIVE LINES BIT MAP.

021662  030205          2$:     BIT     R2,R5           ;TEST LINE BIT IN ACTIVE LINES BIT MAP.
021664  001442                  BEQ     4$              ;LINE ACTIVE?  NO, SKIP REPORT FOR LINE.

021666  010400                  MOV     R4,R0           ;SET UP DUT CSR IND.ADR.REG FIELD
021670  050100                  BIS     R1,R0           ;  LEAVING THE INTERRUPT ENABLE
021672  010013                  MOV     R0,(R3)         ;  BITS IN THE SPECIFIED STATE.
021674  017700  160310          MOV     @STATA,R0       ;READ THE DUT STATUS REG FOR THIS LINE.
021700                          SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
021700  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021704  005002                  CLR     R2              ;CLEAR THE SIGNAL STATUS INDICATORS.
021706  005003                  CLR     R3
021710  005004                  CLR     R4
021712  005005                  CLR     R5
021714  006300                  ASL     R0              ;SHIFT DSR INTO CARRY,
021716  006102                  ROL     R2              ;  THEN ROTATE INTO INDICATOR.
```

```
4615 021720  006300                    ASL     R0              ;SHIFT BLANK SLOT INTO CARRY,
4616 021722  006300                    ASL     R0              ;  SHIFT RI INTO CARRY,
4617 021724  006103                    ROL     R3              ;  THEN ROTATE INTO INDICATOR.
4618 021726  006300                    ASL     R0              ;SHIFT DCD INTO CARRY,
4619 021730  006104                    ROL     R4              ;  THEN ROTATE INTO INDICATOR.
4620 021732  006300                    ASL     R0              ;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,R0
     021762  104417                                                        TRAP    C$PNTF
     021764  062706  000016                                               ADD     #16,SP
4627 021770                             PASS                            ;RESTORE ALL THE GPRS.
     021770  004736                                     JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
4628
4629 021772  006302                4$:   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,R0
     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 PREGOS SUBRT.
4637 022026  000207                      RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE               - MUL16U -
                                ;** ************************************************************************
                                ;*                      - 16 Bit Unsigned Multiply Routine -
                                ;*      This routine multiplies 2 16 bit unsigned numbers and returns a 16 bit
                                ;*      unsigned result.  The multiplication is performed by iterative
                                ;*      addition of one number to a sum while decrementing the other number
                                ;*      to zero.  If overflow occurrs (177777 to 0) the product is invalid.
                                ;*
                                ;* INPUTS:      R1 - Multiplicand (16 bit unsigned).
                                ;*              R2 - Multiplier (16 bit unsigned).
                                ;*
                                ;* OUTPUTS:     R1 - Product (16 bit unsigned), -1 if overflow.
                                ;*              CARRY - Set if success (no overflow), clear otherwise.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,MUL16U
                                ;*
                                ;* COMMENTS:    Note: For minimum execution time R2 should contain the
                                ;*                    smaller of the 2 arguments.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;-- ************************************************************************

022030                          MUL16U:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
022030  004537  005326                          JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
022034  005003                          CLR     R3                      ;CLEAR THE PRODUCT.
022036  005702                          TST     R2                      ;CHECK THE MULTIPLIER.
022040  001003                         BNE     2$                      ;GO TO DO MULTIPLICATION IF NOT ZERO.
022042  005001                          CLR     R1                      ;RETURN A PRODUCT OF ZERO.
022044  000261                          SEC                             ;INDICATE SUCCESS.
022046  000412                         BR      60$                     ;EXIT THE ROUTINE.

022050  060103                 2$:      ADD     R1,R3                   ;ADD THE MULTIPLICAND TO THE PRODUCT.
022052  103405                        BCS     50$                     ;EXIT WITH OVERFLOW IF ONE OCCURRED.
022054  005302                          DEC     R2                      ;DECREMENT THE MULTIPLIER.
022056  001374                         BNE     2$                      ;LOOP IF MULTIPLIER NOT ZERO.
022060  010301                          MOV     R3,R1                   ;PREPARE TO PASS OUT THE PRODUCT.
022062  000261                          SEC                             ;INDICATE SUCCESS.
022064  000403                         BR      60$                     ;EXIT WITH SUCCESS.

022066  012701  177777         50$:     MOV     #-1,R1                  ;FORCE PRODUCT TO MAX VALUE, WE OVERLFOWED.
022072  000241                          CLC                             ;INDICATE FAILURE.

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 PREGOS SUBRT.
                                                                        ;  R1 - PRODUCT (16 BIT UNSIGNED),
022102  000207                         RTS     PC                      ;  CARRY - SET IF SUCCESS (NO OVERFLOW).
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - NEWCHR -
                        ;** ************************************************************************
                        ;*                       - New Character Handling Routine -
                        ;*      This subroutine handles a new character which has been read from
                        ;*      the DUT.  The counters and pointers which are involved with the
                        ;*      character are updated.  The character is checked for errors and
                        ;*      any errors which are found are reported.
                        ;*
                        ;* INPUTS:      R2 - The read character including error flags and line number.
                        ;*              R3 - Mask of the inactives bits in a TX or RX char byte.
                        ;*              ACTLNS  - Bit map of active DUT lines.
                        ;*              DPRSQB - Label at data pattern resync queues table base.
                        ;*              TXRXLB - Base of TX/RX line number association table.
                        ;*              BITTBL - Table of words with bits set for use in forming maps.
                        ;*              ERSMRF - "Print error summary for line" flags.
                        ;*              ERRTBL - Error information (ERRNBR, ERRMSG, ERRTYP).
                        ;*              ERCNTB - Base of the RX character error counters table.
                        ;*              NDERPT - Contains number of char errors to report on a line.
                        ;* INPUTS TO SUBROUTINES:       CHCNTB, DPENDB, DPLEN, DPRSQE, EXCNTB, RXCNTB,
                        ;*                              RXPTRB, ERRNBR, ERMSG, ERRTYP.
                        ;*
                        ;* OUTPUTS:     ERRBLK - Contents destroyed.
                        ;*            Following variables updated for line on which char was received:
                        ;*              DPRSQ - Data pattern resync que of received characters.
                        ;*              ERCNT - Count of the number of character errors on line.
                        ;*              ERSMRF - Updated "print error summary for line" flags.
                        ;*              EXCNT - Count of the number of extra chars received on line.
                        ;*              RXCNT - Count of the number of characters received on line.
                        ;*              RXPTR - Updated to point to the next expected char on line.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,NEWCHR
                        ;*
                        ;* COMMENTS:    This routine can report errors with numbers Initial ERRNBR
                        ;*              and Initial ERRNBR + 1.  ERRNBR is restored to its initial
                        ;*             value before this routine returns.
                        ;*
                        ;* SUBROUTINES CALLED:  CKCHR,CKINAC,TXROFF,TXRON.
                        ;* INDIRECT SUBROUTINES:  CHKEXT,CHKLOS,ER9002,ER9003,UPDCHR.
                        ;-- ************************************************************************

022104                  NEWCHR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
022104  004537  005326                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
022110  010305          MOV     R3,R5           ;GET THE BIT MAP OF INACTIVE DATA BYTE BITS.
022112  052705  177400  BIS     #177400,R5      ;ALL UPPER BITS OF EXPECTED DATA ARE INACTIVE.
022116  005037  022362  CLR     70$             ;CLEAR THE "ERROR FOUND" FLAG.
                        ;+
                        ; If the new character is valid on an inactive line, go report error.
                        ; Routine used also extracts line number from the new character.
                        ;-
022122  004737  017046          JSR     PC,CKINAC       ;CHECK FOR CHAR ON INACTIVE LINE.
022126  103043          BCC     4$              ;GO REPORT ERROR IF ON INACTIVE LINE.
                        ;+
                        ; Push the new character on the resync que for this line.
                        ;-
022130  010304          MOV     R3,R4           ;CALCULATE BASE ADDRESS OF THE
022132  006304          ASL     R4              ;  DATA PATTERN RESYNCH QUEUE
022134  006304          ASL     R4              ;  (QUEUE IS 4 WORDS LONG) FOR
```

GLOBAL SUBROUTINE - NEWCHR - SEQ 0126

```
 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 botton 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 - SEQ 0127

```
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$:    PASS                    ;RESTORE GPRS.
     022356  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
4803 022360  000207                         RTS     PC
4804
4805 022362  000000                  70$:    .WORD   0               ;LOCAL STORAGE FOR ERROR OCCURRED FLAG.
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - OOPS -
                        ;++ ******************************************************************************
                        ;*                      - Program abort subroutine -
                        ;*      This subroutine is used to abort the program when a fatal error is
                        ;*      detected in the program or the host system hardware.  An error message
                        ;*      is printed giving some information about the nature of the abort.
                        ;*
                        ;* INPUTS:      R1 - Error code giving reason for abort.
                        ;*
                        ;* OUTPUTS:     An error message is printed.
                        ;*              A list of return PC values for all subroutine calls is printed.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,OOPS
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ******************************************************************************

022364                  OOPS::  SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
022364  004537  005326          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
022370                      ERRSF   101,EM0101
022370  104454                                                          TRAP    C$ERSF
022372  000145                                                          .WORD   101
022374  022430                                                          .WORD   EM0101
022376  000000                                                          .WORD   0
                        ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
022400                      PRINTF  #EM0102
022400  012746  022514                                                  MOV     #EM0102,-(SP)
022404  012746  000001                                                  MOV     #1,-(SP)
022410  010600                                                          MOV     SP,R0
022412  104417                                                          TRAP    C$PNTF
022414  062706  000004                                                  ADD     #4,SP
022420                  2$:     BREAK                   ;LOOK FOR OPERATOR CONTROL-C INPUT.
022420  104422                                                          TRAP    C$BRK
022422  000776                  BR      2$              ;INFINITE LOOP.
022424                  60$:    PASS                    ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
022424  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
022426  000207                  RTS     PC              ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.

                                .NLIST  BEX
022430  110  117  123  EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
022514  045  116  045  EM0102:: .ASCIZ /%N%APROGRAM HUNG, WAITING FOR A CONTROL-C. <**************%N%N/
                                .EVEN
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - PRFRME -
                        ;++ ************************************************************************
                        ;*                      - PROCESS FRAMING ERRORS -
                        ;*      This subroutine is used in the Framing error bit test, to verify that
                        ;*      all received characters have their framing error bit set and parity
                        ;*      error bit clear.
                        ;*
                        ;* INPUTS:      R2 - Contains the character read from the FIFO.
                        ;*              ERRNBR - Error number of errors in this routine.
                        ;*              ERSMRF - "Report Error Summary for line" flags
                        ;*
                        ;* OUTPUTS:     ERRBLK - The contents of this word are destroyed.
                        ;*              ERCNTB - The error count for this line is updated.
                        ;*              Messages may be printed at the operators console.
                        ;*
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,PRFRME
                        ;*
                        ;* COMMENTS:    This routine reports errors with INITIAL number.
                        ;*              ERRNBR is restored to its initial value before this subroutine
                        ;*              returns.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ER6201.
                        ;-- ************************************************************************

022612                  PRFRME::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
022612  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022616  013704  005320          MOV     ERRNBR,R4       ;SAVE THE CONTENTS OF THE INITIAL ERROR NUMBER.
022622  005005                  CLR     R5              ;CLEAR ERROR/MESSAGE FLAGS.

                        ;+
                        ; Test Framing and parity error bits in turn.  Report any errors found, ie.
                        ;  Framing error bit clear, or Parity error bit set.
                        ;-
022624  012737  014204  005324          MOV     #ER6201,ERRBLK  ;SET UP THE ADDRESS OF THE ERROR ROUTINE.
022632  032702  020000          BIT     #BIT13,R2       ;CHECK ON STATE OF THE FRAMING ERROR BIT.
022636  001002                  BNE     6$              ;BRANCH IF FRAMING ERROR BIT SET.
022640  052705  000002          BIS     #BIT1,R5        ;SET REPORT FRAMING ERROR FLAG.

022644  032702  010000  6$:     BIT     #BIT12,R2       ;CHECK ON THE STATE OF THE PARITY ERROR BIT.
022650  001402                  BEQ     8$              ;BRANCH IF PARITY ERROR BIT CLEAR.
022652  052705  000014          BIS     #14,R5          ;SET REPORT "PARITY ERROR SET" FLAGS.
022656  005705          8$:     TST     R5              ;CHECK IF ANY ERROR FLAGS SET.
022660  001407                  BEQ     60$             ;EXIT IF ALL FLAGS CLEAR.
022662  036337  002366  002502          BIT     BITTBL(R3),ERSMRF  ;CHECK THE ERROR SUMMARY FLAG FOR THIS LINE.
022670  001001                  BNE     10$             ;SKIP ERROR REPORT IF ERROR SUMMARY FLAG SET.

                                ;REPORT ERROR "CHARACTER RECEIVED WITH PARITY/FRAMING ERROR BIT SET".
022672                          ERROR                   ;                       >>>>> ERROR <<<<<.
022672  104460                                                                          TRAP    C$ERROR
022674  005263  003304  10$:     INC     ERCNTB(R3)      ;INCREMENT ERROR COUNT FOR THIS LINE.
022700  010437  005320  60$:     MOV     R4,ERRNBR       ;RESTORE ERROR NUMBER.
022704                          PASS                    ;RESTORE GPRS.
022704  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
022706  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - PRPARE -
                        ;++ ************************************************************************
                        ;*                      - PROCESS PARITY ERRORS -
                        ;*      This subroutine is used in the Parity error test, to verify that
                        ;*      all received characters have their parity error bit set and framming
                        ;*      error bit clear.
                        ;*
                        ;* INPUTS:      R2 - Contains the character read from the FIFO.
                        ;*              R3 - Contains 2 * line number of the read char.
                        ;*              ERRNBR - Error number of errors in this routine.
                        ;*              ERSMRF - "Report Error Summary for line" flags
                        ;*
                        ;* OUTPUTS:     ERRBLK - The contents of this word are destroyed.
                        ;*              ERCNTB - The error count for this line is updated.
                        ;*              Messages may be printed at the operators console.
                        ;*
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,PRPARE
                        ;*
                        ;* COMMENTS:    This routine reports errors with INITIAL ERRNBR thru ERRNBR+1.
                        ;*              ERRNBR is restored to its initial value before this subroutine
                        ;*              returns.
                        ;*              The contents of the ERRBLK are destroyed.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ER9002,ER6201.
                        ;-- ************************************************************************

022710                  PRPARE::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
022710  004537  005326          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022714  013746  005320          MOV     ERRNBR,-(SP)    ;SAVE THE CONTENTS OF THE INITIAL ERROR NUMBER.
022720  005005                  CLR     R5              ;CLEAR ERROR/MESSAGE FLAGS.

                        ;+
                        ; Test Framming and parity error bits in turn.  Report any errors found, ie.
                        ;   Parity error bit clear, or Framming error bit set.
                        ;-
022722  012737  014204  005324          MOV     #ER6201,ERRBLK  ;SET UP THE ADDRESS OF THE ERROR ROUTINE.
022730  032702  010000                  BIT     #BIT12,R2       ;CHECK ON STATE OF THE PARITY ERROR BIT.
022734  001002                  BNE     6$              ;BRANCH IF PARITY ERROR BIT SET.
022736  052705  000010                  BIS     #BIT3,R5        ;SET REPORT PARITY ERROR FLAG.
022742  032702  020000          6$:     BIT     #BIT13,R2       ;CHECK ON THE STATE OF THE FRAMMING ERROR BIT.
022746  001402                  BEQ     8$              ;BRANCH IF FRAMMING ERROR BIT CLEAR.
022750  052705  000003                  BIS     #3,R5           ;SET REPORT "FRAMMING ERROR SET" FLAGS.
022754  005705                  8$:     TST     R5              ;CHECK IF ANY ERROR FLAGS SET.
022756  001405                  BEQ     12$             ;BRANCH TO MAKE DATA CHECK IF ALL FLAGS CLEAR.
022760  036337  002366  002502          BIT     BITTBL(R3),ERSMRF  ;CHECK THE ERROR SUMMARY FLAG FOR THIS LINE.
022766  001024                  BNE     14$             ;SKIP ALL ERROR REP IF IN ERROR SUMMARY MODE.
                                ;REPORT ERROR "CHAR RECEIVED WITH PARITY/FRAMMING ERROR BIT SET/CLEAR".
022770                          ERROR                   ;               >>>>> ERROR <<<<<.
022770  104460                                                                  TRAP    C$ERROR

                        ;+
                        ; Compare actual data with expected data to check for multiple errors.
                        ;-
022772  005237  005320  12$:     INC     ERRNBR          ;INCREMENT ERROR NUMBER.
022776  016304  003404          MOV     RXPTRB(R3),R4   ;GET THE POINTER TO THE EXPECTED DATA.
023002  111404                  MOVB    (R4),R4         ;GET THE EXPECTED DATA.
```

GLOBAL SUBROUTINE - PRPARE - SEQ 0131

```
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 PREGOS 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
```

```
                                .SBTTL  GLOBAL SUBROUTINE                PUFIFO -
                                ;*************************************************************************
                                ;*      - PURGE THE FIFO
                                ;*      This routine tries to remove all the characters from the FIFO.
                                ;*      Any BMP codes that are found are saved on the BMP code queue.
                                ;*
                                ;* INPUTS:      RBUFA- Contains the  address of the Receiver.
                                ;*
                                ;*
                                ;* OUTPUTS:     Carry bit - Indicates the state of the fifo, set:= purged.
                                ;*              BMPCQ - The contents of the BMP code queue may be updated.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,PUFIFO
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: SAVBMP.
                                ;*************************************************************************

023152                          PUFIFO::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5
023152  004537  005326                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023156  012701  001000                  MOV     #512.,R1        ;SET MAXIMUM TRY COUNT OF 512.
023162  013704  002204                  MOV     RBUFA,R4        ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.

023166  011402                  2$:     MOV     (R4),R2         ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
023170  100016                         BPL     6$              ;EXIT IF THE FIFO IS EMPTY, DATA_VALID C_R.
                                ;+
                                ; Check if the read character is actually a BMP code.
                                ;  If it is, then save it on the BMP code queue to be reported later.
                                ;-
023172  012700  070000                  MOV     #70000,R0       ;GENERATE A BIT MAP OF CHAR ERROR BITS
023176  040200                         BIC     R2,R0           ; WHICH ARE NOT SET FOR CHAR.
023200  001006                         BNE     4$              ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
                                  ;+
                                  ; Check if the read data is modem status , BMP or Selftest?.
                                  ;-
023202  012700  000300                   MOV     #300,R0         ; CHECK IF BMP OR SELFTEST?.
023206  040200                          BIC     R2,R0           ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
023210  001002                         BNE     4$              ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
023212  004737  024726                 JSR     PC,SAVBMP       ;SAVE BMP CODE ON THE QUEUE.

023216  005301                  4$:     DEC     R1              ;DECREMENT THE TRY COUNT.
023220  001362                         BNE     2$              ;LOOP TO TRY AGAIN.
023222  000241                          CLC                     ;CLEAR CARRY,TO INDICATE FIFO NOT PURGED.
023224  000401                         BR      60$             ;EXIT WITH CARRY CLEAR.
023226  000261                  6$:     SEC                     ;SET CARRY, TO INDICATE FIFO PURGED.

023230                          60$:    PASS                    ;RESTORE GPRS,
023230  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;CARRY BIT, SET INDICATES FIFO PURGED.
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 unxepected 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 destroyed.
 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 POUTINES 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                                   ;-
```

GLOBAL SUBROUTINE - PUFIFR - SEQ 0135

```
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 PREGO5 SUBRT.
5142                                         ;CARRY BIT, SET INDICATES FIFO PURGED, DO NOT
5143                                         ; ABORT THE TEST.
5144 023424  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - PURRXB -
                        ;++ ******************************************************************************
                        ;*                          - Purge the RX Buffer in Memory Routine -
                        ;*      This subroutine is used before the beginning of a TX/RX of data
                        ;*      patterns to clear out the RX buffer and to initialize the various
                        ;*      counters and pointers related to that buffer.
                        ;*
                        ;* INPUTS:      RXBSTA - Label at the beginning of the RX buffer.
                        ;*
                        ;* OUTPUTS:     RXBCNT - Count of # of chars in RX buffer (Cleared).
                        ;*              RXBIPT - Input pointer to RX buffer (Initialized).
                        ;*              RXBOPT - Output pointer to RX buffer (Initialized).
                        ;*              The contents of the RX BUFFER are cleared.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,PURRXB
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ******************************************************************************

023426                  PURRXB:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023426  004537  005326                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.

023432  012701  002714          MOV     @RXBOPT,R1      ;GET THE ADDRESS OF THE RX OUTPUT POINTER.
023436  012721  002722          MOV     @RXBSTA,(R1)+   ;INITIALIZE THE RX BUFFER OUTPUT POINTER.
023442  012721  002722          MOV     @RXBSTA,(R1)+   ;INITIALIZE THE RX BUFFER INPUT POINTER.
023446  005021          2$:     CLR     (R1)+           ;CLEAR CHAR COUNT AND THE BUFFER AREA.
023450  020127  003122          CMP     R1,@RXBEND      ;CHECK IF LAST LOCATION HAS BEEN CLEARED.
023454  101774                  BLOS    2$              ;LOOP IF NOT DONE.

023456                  60$:    PASS                            ;RESTORE GPRS.
023456  004736                  JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
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                            ;+
5229                            ; Wait for a character to appear in the FIFO.
5230                            ; If no character appears within time-out period: exit routine, we're done.
5231                            ;-
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$:    JSR     PC,RXIE0        ;TURN OFF DUT RX INTERRUPTS.
5309 023760  004737  025442                    JSR     PC,TXDONE       ;CHECK IF TX DONE, TURN OFF DUT TX INTERRUPTS.
5310 023764  005002                            CLR     R2              ;CLEAR THE DUMMY CHARACTER.
5311 023766  005001                            CLR     R1              ;CLEAR THE LOOP COUNTER.
5312 023770  004737  022104            20$:    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,PREGOS       ;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)+
 5362 024064  000207                            RTS     PC                              ;RETURN TO PREGOS SUBRT.
```

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

SEQ 0142

GLOBAL SUBROUTINE - REPCOD -

```
                                        ;-
024210  010337  005320          6$:     MOV     R3,ERRNBR
                                        
024214                          60$:    PASS                    ;RESTORE GPRS.
024214  004736                          JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
024216  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE               - REPSMR -
                                ;** ************************************************************************
                                ;*                      - Report Error Summary Routine -
                                ;*      This subroutine reports an error summary for those lines which have
                                ;*      exceeded the number of individual errors to report for a single line
                                ;*      in a single test.  This parameter can be specified by the operator if
                                ;*      he/she answers the Software Parameter Questions.
                                ;*
                                ;* INPUTS:      ERCNTB - Label at base of line error counters table.
                                ;*              ERRMSG - Address of primary error message.
                                ;*              ERRNBR - Error number of errors in this routine.
                                ;*              ERSMRF - "Report error summary for line" flags.
                                ;*
                                ;* OUTPUTS:     ERRBLK - Address of error reporting routine (Destroyed).
                                ;*              Summary messages may be printed at the operator console.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,REPSMR
                                ;*
                                ;* COMMENTS:    If no lines have exceeded the maximum number of individual
                                ;*              errors to report, no messages are printed by this routine.
                                ;*              Error summaries in this routine are reported as errors.
                                ;*              The contents of ERRBLK are destroyed.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED:
                                ;-- ************************************************************************

024220                          REPSMR:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
024220  004537  005326                         JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
024224  005737  002502                 TST     ERSMRF          ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
024230  001404                         BEQ     60$             ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
                                ;*
                                ; We have some error summaries to report.
                                ;-
024232  012737  015052  005324          MOV     #ER9004,ERRBLK  ;SELECT ERROR REPORTING ROUTINE.
                                    ;*
                                    ; Report
                                    ;    "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
                                    ;-
024240                                  ERROR
024240  104460                                                                  TRAP    C$ERROR

024242                          60$:    PASS                    ;RESTORE GPRS.
024242  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
024244  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - RESETT -
                        ;*******************************************************************************
                        ;*              - Reset Device Under Test -
                        ;*      This subroutine is used to reset the DUT to a known state.
                        ;*      If reset does not succesfully complete, ie. time-out occurs, then
                        ;*      an abort test error message is reported.
                        ;*
                        ;* INPUTS:      CSRA - Contains the address of the CSR
                        ;*              TXBFCA - Contains address of DUT DMA Buffer Count register.
                        ;*              ERRTBL- ERRTYP,ERNBR,and ERRMSG set up correctly.
                        ;*
                        ;* OUTPUTS:     The DUT performs its reset function into a known state.
                        ;*              CARRY - Clear indicates the test is to be aborted.
                        ;*              ERRBLK - value may be destroyed.
                        ;*              IESTAT - TX and RX interrupt flags are cleared.
                        ;*              TX and RX interrupt enable bits in the DUT's CSR are cleared.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RESETT
                        ;*
                        ;* COMMENTS:    This subroutine can report errors with numbers initial ERRNBR
                        ;*              This routine does not destroy the value of ERRNBR.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: DELAY,MSLGET.
                        ;*******************************************************************************

024246                  RESETT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
024246  004537  005326          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
024252  012702  000040          MOV     #BIT05,R2       ;SET BIT MASK OF MASTER RESET BIT.
                        ;*
                        ; Test the state of the master reset bit in the CSR.
                        ;  If MR is set then wait for self-test to complete.
                        ;   If time-out occurs, report the error and pass-out abort test indicator.
                        ;-
024256  013704  002202          MOV     CSRA,R4         ;GET THE ADDRESS OF THE DUT'S CSR.
024262  030214                  BIT     R2,(R4)         ;CHECK STATE OF MASTER RESET BIT.
024264  001406                  BEQ     2$              ;DON'T DELAY IF MR IS ALREADY CLEAR.
024266  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
024270  012701  004704          MOV     #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
024274  004737  021464          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
024300  103012                  BCC     4$              ;GO REPORT ERROR IF TIMEOUT OCCURRED.

                        ;*
                        ; Set Master Reset bit in CSR.  Clear TX and RX enable bits, etc.
                        ;  Skip the selftest.
                        ;  Time-out of 2.5 secs, just in case the self-test executes.
                        ;-
024302  010277  155674  2$:     MOV     R2,@CSRA        ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
024306  004737  024774          JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
                        ;*
                        ;   Set Self-test time-out of 2.5 seconds, and wait for M.R to clear.
                        ;   If Time out occurs, then report the fatal error and pass-out the abort
                        ;   test indicator.
                        ;-
024312  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
024314  012701  004704          MOV     #2500.,R1       ;PASS TIME-OUT VALUE OF 2.5 SECONDS.
024320  004737  021464          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
024324  103410                  BCS     6$              ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
```

GLOBAL SUBROUTINE - RESETT - SEQ 0145

```
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 PREGO5 SUBRT.
5542                                                                    ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
5543 024356  000207                            RTS     PC
5544
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - RRXNDN -
                        ;** ************************************************************************
                        ;*              - Report Reception Not Completed Routine -
                        ;*      This subroutine checks for lines which did not receive the complete
                        ;*      data pattern.  If any are found, they are reported.
                        ;*
                        ;* INPUTS:      R5 - Local active lines bit map.
                        ;*              DPLENB - Base of table of data pattern lengths.
                        ;*              ERRMSG - Address of primary error message for this routine.
                        ;*              ERRNBR - Error number of error reported in this routine.
                        ;*              RXCNTB - Label at base of the RX character counters table.
                        ;*              RXDONF - Reception done flags.
                        ;*
                        ;* OUTPUTS:     ERRBLK   Address of the error reporting routine (Destroyed).
                        ;*              Messages may be printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RRXNDN
                        ;*
                        ;* COMMENTS:    If no lines failed to complete their reception, no messages
                        ;*              are printed.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ER9005.
                        ;-- ************************************************************************

024360                  RRXNDN:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024360  004537  005326                 JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
024364  010502                  MOV     R5,R2                   ;GET COPY OF THE LOCAL ACTIVE LINES BIT MAP.
024366  043702  002506          BIC     RXDONF,R2               ;GET MAP OF ACTIVE LINES WITH RX DONE FLAG CLR.
024372  001410                  BEQ     60$                     ;EXIT IF NO ACTIVE LINES HAVE RX DONE FLAG CLR.
                        ;+
                        ; We have some "RX not completed" errors to report.
                        ;-
024374  012737  015154  005324          MOV     #ER9005,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
024402  012701  011767                  MOV     #EM9016,R1      ;INDICATE THAT WE ARE DEALING WITH RECEPTION.
024406  012704  003544                  MOV     #RXCNTB,R4      ;PASS BASE OF RX CHAR COUNTERS TABLE TO ER9005.
                        ;+
                        ; Report "SINGLE CHARACTER MODE TEST ERROR:"
                        ;    "DATA PATTERN NOT COMPLETELY RECEIVED ON ALL LINES:"
                        ;      ...
                        ;-
024412                      ERROR
024412  104460                                                   TRAP    C$ERROR

024414                  60$:    PASS                            ;RESTORE GPRS.
024414  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
024416  000207                  RTS     PC
```

GLOBAL SUBROUTINE - RTXNDN - SEQ 0147

```
                        .SBTTL  GLOBAL SUBROUTINE               - RTXNDN -
                        ;** ************************************************************************
                        ;*              - Report Transmission Not Completed Routine -
                        ;*      This subroutine checks for lines which did not transmit the complete
                        ;*      data pattern.  If any are found, they are reported.
                        ;*
                        ;* INPUTS:      R5 - Local active lines bit map.
                        ;*              DPLENB - Label at base of data pattern length table.
                        ;*              ERRMSG - Address of primary error message for this routine.
                        ;*              ERRNBR - Error number of error reported in this routine.
                        ;*              TXCNTB - Label at base of the TX character counters table.
                        ;*              TXDONF - Transmission done flags.
                        ;*
                        ;* OUTPUTS:     ERRBLK - Address of the error reporting routine (Destroyed).
                        ;*              Messages may be printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RTXNDN
                        ;*
                        ;* COMMENTS:    If no lines failed to complete their transmission, no messages
                        ;*              are printed.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ER9005.
                        ;-- ************************************************************************

024420                  RTXNDN:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024420  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
024424  010502                  MOV     R5,R2                   ;GET COPY OF THE LOCAL ACTIVE LINES BIT MAP.
024426  043702  002504          BIC     TXDONF,R2               ;GET MAP OF ACTIVE LINES WITH TX DONE FLAG CLR.
024432  001410                  BEQ     60$                     ;EXIT IF NO ACTIVE LINES HAVE TX DONE FLAG CLR.
                        ;+
                        ; We have some "TX not completed" errors to report.
                        ;-
024434  012737  015154  005324          MOV     @ER9005,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
024442  012701  011753          MOV     @EM9015,R1              ;INDICATE WE ARE DEALING WITH TRANSMISSION.
024446  012704  003504          MOV     @TXCNTB,R4              ;PASS BASE OF TX CHAR COUNTERS TO TABLE ER0805.

                        ;+
                        ; Report "SINGLE CHARACTER MODE TEST ERROR:"
                        ;    "DATA PATTERN NOT COMPLETELY TRANSMITTED ON ALL LINES:"
                        ;      ...
                        ;-
024452                      ERROR                       ;               >>>>> ERROR <<<<<.
024452  104460                                                              TRAP    C$ERROR

024454                  60$:    PASS                    ;RESTORE GPRS.
024454  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
024456  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - RXDSBL -
                                ;** ************************************************************************
                                ;*                          - Disable Receivers -
                                ;*      This subroutine is used to disable reception on selected lines by,
                                ;*      clearing the associated RX_ENABLE bit on the DUT.
                                ;*
                                ;* INPUTS:      R5 - Bit's set correspond to lines on which to clear RX_ENABLE.
                                ;*              CSRA - Contains the address of the DUT CSR.
                                ;*              IESTAT - Contains the state of TXIE and RXIE bits in the CSR.
                                ;*              NUMLNS - Equated to be the maximum number of lines available.
                                ;*              LNCTRA - Contains the address of the LNCTRL register.
                                ;*
                                ;* OUTPUTS:     R5 - Bit's set indicate initial states of all RX_ENBLE bits.
                                ;*              LNCTRA - The state of the RX_ENBLE bit may be altered.
                                ;*              The contents of the IND_ADD_REG field in the CSR are destroyed.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,RXDSBL
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************

024460                          RXDSBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024460  004537  005326                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
024464  010500                  MOV     R5,R0           ;COPY BIT MAP OF LINES TO DISABLE RECEPTION.
024466  012701  000001          MOV     #BIT0,R1        ;INITIALIZE THE SELECTED LINE BIT MASK.
024472  013702  002212          MOV     LNCTRA,R2       ;GET THE ADDRESS OF THE LNCTRL REGISTER.
024476  012703  000010          MOV     #NUMLNS,R3      ;GET MAXIMUM LINE NUMBER PLUS ONE.
024502  013704  002234          MOV     IESTAT,R4       ;GET THE STATES OF THE INT ENABLE BITS.
024506  005005                  CLR     R5              ;LOG POSSIBLE RX DISABLED ON ALL LINES.
                                ;+
                                ; Select every line in turn, and log the state of each RX_ENABLE bit.
                                ;-
024510  010477  155466          2$:     MOV     R4,@CSRA        ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
024514  032712  000004                  BIT     #BIT2,(R2)      ;CHECK STATE OF RX_ENABLE BIT ON SELECTED LINE.
024520  001401                  BEQ     4$              ;SKIP NEXT INSTRUCTION IF RX_ENABLE CLEAR.
024522  050105                  BIS     R1,R5           ;LOG RX ENABLE BIT SET FOR SELECTED LINE.
                                ;+
                                ;  Clear RX ENBLE on lines that have a corresponding bit set in the rx disable
                                ;  line bit map.
                                ;
024524  030100                  4$:     BIT     R1,R0           ;CHECK STATE OF DISABLE LINE BIT MAP.
024526  001402                  BEQ     6$              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
024530  042712  000004                  BIC     #BIT2,(R2)      ;CLEAR RX_ENABLE BIT ON SELECTED LINE.
024534  005204                  6$:     INC     R4              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
024536  006301                  ASL     R1              ;SHIFT BIT MAP FOR NEXT LINE.
024540  005303                  DEC     R3              ;DECREMENT LINE NUMBER.
024542  001362                  BNE     2$              ;LOOP TO CHECK NEXT LINE.

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 PREG05 SUBRT.
                                                                ;R5 - PREVIOUS STATES OF ALL RX_ENABLE BITS.
024552  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - RXENBL -
                        ;++ ************************************************************************
                        ;*                      - Enable Receiver -
                        ;*      This subroutine is used to enable reception on selected lines by
                        ;*      setting the associated RX.ENABLE bit on the DUT.
                        ;*
                        ;* INPUTS:      R5 - Bit's set correspond to lines on which to set RX.ENABLE.
                        ;*              CSRA - Contains the address of the DUT CSR.
                        ;*              IESTAT - Contains the state of TXIE and RXIE bits in the CSR.
                        ;*              NUMLNS - Equated to be the maximum number of lines available.
                        ;*              LNCTRA - Contains the address of the LNCTRL register.
                        ;*
                        ;* OUTPUTS:     R5 - Bit's set indicate previously disabled lines.
                        ;*              LNCTRA - The state of the RX.ENABLE bit may be altered.
                        ;*              The contents of the IND.ADD.REG field in the CSR are destroyed.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RXENBL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

024554                  RXENBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024554  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
024560  010500          MOV     R5,R0           ;COPY BIT MAP OF LINES TO ENABLE.
024562  012701  000001  MOV     #BIT0,R1        ;INITIALIZE THE SELECTED LINE BIT MASK.
024566  013702  002212  MOV     LNCTRA,R2       ;GET THE ADDRESS OF THE LNCTRL REGISTER.
024572  012703  000010  MOV     #NUMLNS,R3      ;GET MAXIMUM LINE NUMBER.
024576  013704  002234  MOV     IESTAT,R4       ;GET THE STATES OF THE INT ENABLE BITS.
024602  005005          CLR     R5              ;CLEAR RX.ENABLE BIT LOG OF DISABLED LINES.
                        ;+
                        ; Select every line in turn,and log any RX.ENBLE bit that is clear.
                        ;-
024604  010477  155372  2$:     MOV     R4,@CSRA        ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
024610  032712  000004  BIT     #BIT2,(R2)      ;CHECK STATE OF RX.ENABLE BIT ON SELECTED LINE.
024614  001001          BNE     4$              ;SKIP NEXT INSTRUCTION IF RX.ENABLE SET.
024616  050105          BIS     R1,R5           ;LOG RX ENABLE BIT CLEAR FOR SELECTED LINE.
                        ;+
                        ;  Set RX.ENBLE on lines that have a corresponding bit set in the rx enable
                        ;  line bit map.
                        ;-
024620  030100          4$:     BIT     R1,R0           ;CHECK STATE OF RX.ENABLE LINE BIT MAP.
024622  001402          BEQ     6$              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
024624  052712  000004  BIS     #BIT2,(R2)      ;ENABLE RECEPTIONON SELECTED LINE.
024630  005204          6$:     INC     R4              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
024632  006301          ASL     R1              ;SHIFT BIT MAP FOR NEXT LINE.
024634  005303          DEC     R3              ;DECREMENT LINE NUMBER.
024636  001362          BNE     2$              ;LOOP TO CHECK NEXT LINE.

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.
                                                                ;R5 - LINE BIT MAP CORRESPONDING TO THE
                                                                ; PREVIOUS LINES THAT WERE DISABLED.
024646  000207          RTS     PC
```

GLOBAL SUBROUTINE - RXIE0 - SEQ 0150

```
                                .SBTTL  GLOBAL SUBROUTINE               - RXIE0 -
                                ;++ ******************************************************************************
                                ;*                          - RECEIVER INTERRUPT DISABLE -
                                ;*      This routine is used to disable receiver interrupts in the DHV11-M.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      The RX.INT.ENBL bit is cleared in the DUT CSR.
                                ;*               IESTST -contains the updated status of the TX and RX interrupt
                                ;*               enable bits.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,RXIE0
                                ;*
                                ;* COMMENTS:     The contents of the indirect address register field in
                                ;*               the DUT CSR are destroyed.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ******************************************************************************
024650  010046                  RXIE0:: MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
024652  106746                          MFPS    -(SP)           ;SAVE PROCESSOR PRIORITY ON STACK.
024654  106427  000340                  MTPS    #PRI07          ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
024660  042737  137777  002234          BIC     #137777,IESTAT  ;CLEAR RX.INT.ENBL BIT IN IESTAT.
024666  013777  002234  155306          MOV     IESTAT,@CSRA    ;DISABLE RX INTERRUPTS.
024674  106426                          MTPS    (SP)+           ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
024676  012600                          MOV     (SP)+,R0        ;RESTORE R0.
024700  000207                         RTS     PC
```

GLOBAL SUBROUTINE - RXIE1 -

```
                                .SBTTL  GLOBAL SUBROUTINE               - RXIE1 -
                                ;++ ******************************************************************************
                                ;*                          - RECEIVER INTERRUPT ENABLE -
                                ;*      This routine is used to enable receiver interrupts in the DHV11-M.
                                ;*
                                ;* INPUTS:      NONE.
                                ;*
                                ;* OUTPUTS:     The RX.INT.ENBL bit is set in the DUT CSR.
                                ;*              IESTST -contains the updated status of the TX and RX interrupt
                                ;*              enable bits.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,RXIE1
                                ;*
                                ;* COMMENTS:    The contents of the indirect address register field in
                                ;*              the DUT CSR are destroyed.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ******************************************************************************

 024702  052737  000100  002234  RXIE1::  BIS     #BIT06,IESTAT   ;SET RX.INT.ENBL BIT IN IESTAT.
 024710  042737  137677  002234           BIC     #137677,IESTAT  ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
 024716  013777  002234  155256           MOV     IESTAT,@CSRA    ;ENABLE RX INTERRUPTS.
 024724  000207                          RTS     PC
```

GLOBAL SUBROUTINE - SAVBMP - SEQ 0152

```
                        .SBTTL  GLOBAL SUBROUTINE               - SAVBMP -
                        ;** ************************************************************************
                        ;*                          - Save BMP codes Routine -
                        ;*      This routine saves the parameter passed in, onto the BMP code queue
                        ;*      together with the number of the currently executing test.
                        ;*
                        ;* INPUTS:      R2 - Contains the BMP code that is to be placed on the queue.
                        ;*              BMPCQP - Contains address of next location in the bmp queue.
                        ;*              BMPCQB - Label at base of the BMP code queue.
                        ;*              BMPCQE - Label of next location after the end of the BMP queue.
                        ;*              TSTNUM - Contains the number of the current test.
                        ;*
                        ;* OUTPUTS:     BMPCQP    Incremented by 4.
                        ;*              The contents of the BMP code queue are updated.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,SAVBMP
                        ;*
                        ;* COMMENTS:    If the overflow occurs then the last location will be
                        ;*              overwritten by any subsequent attempts to update the queue.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ; - ************************************************************************

024726                  SAVBMP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024726  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
024732  013704  002512          MOV     BMPCQP,R4               ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
024736  113724  002224          MOVB    TSTNUM,(R4)+            ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
024742  005204                  INC     R4                      ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
024744  042702  177400          BIC     #177400,R2              ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
024750  010224                  MOV     R2,(R4)+                ;SAVE THE BMP CODE ON THE QUEUE.
024752  020427  002714          CMP     R4,#BMPCQE              ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
024756  103402                  BLO     2$                      ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
024760  162704  000004          SUB     #4,R4                   ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
024764  010437  002512  2$:     MOV     R4,BMPCQP               ;SAVE THE POINTER.

024770                  60$:    PASS                            ;RESTORE GPRS.
024770  004736                          JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
024772  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE               - SKPSTS -
                                ;** ************************************************************************
                                ;*                      - Skip Selftest Routine -
                                ;*      This subroutine is used to skip the selftest after a DUT reset has been
                                ;*      initiated.  It must be entered immediately after setting the DUT Master
                                ;*      Reset routine or after the execution of a bus reset (because of timing
                                ;*      considerations).
                                ;*
                                ;* INPUTS:      CSRA - Contains address of the DUT CSR.
                                ;*              TXBFCA - Contains address of DUT DMA Buffer Count register.
                                ;*
                                ;* OUTPUTS:     Skip selftest codes are written to the DUT registers.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,SKPSTS
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: DELAY.
                                ;-- ************************************************************************

024774                          SKPSTS:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
024774  004537  005326                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
025000  012704  000012                  MOV     #10.,R4         ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
025004  004737  017354                 JSR     PC,DELAY        ;DELAY FOR 10 MILLI-SECONDS.
                                ;+
                                ; Write skip self-test code (52525) to all the indexed DUT Registers.
                                ;-
025010  012701  000050                  MOV     @NUMLNS!BIT05,R1 ;FORM IND.ADR.REG FIELD (PLUS M.R. BIT) WORD.
                                       ;THE ABOVE INCLUSION OF THE M.R. BIT IS NECESSARY BECAUSE OF THE
                                       ;  LACK OF A M.R. BIT WRITE LOCK-OUT ON THE DHV11-M.
025014  012703  052525                  MOV     #52525,R3       ;INITIALISE THE SKIP SELF-TEST CODE.
025020  005301                  4$:     DEC     R1              ;SELECT THE NEXT SET OF DEVICE REGISTERS.
025022  013704  002202                  MOV     CSRA,R4         ;GET THE ADDRESS OF THE CSR OF THE DUT.
025026  010124                          MOV     R1,(R4)+        ;SELECT A BANK OF DUT REGISTERS.
025030  010324                  6$:     MOV     R3,(R4)+        ;WRITE THE CODE TO A DUT REGISTER.
025032  020437  002220                  CMP     R4,TXBFCA       ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
025036  103774                         BLO     6$              ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
025040  032701  000017                  BIT     #17,R1          ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
025044  001365                         BNE     4$              ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.

025046                          60$:    PASS                    ;RESTORE GPRS.
025046  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
025050  000207                         RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - SPLSUP -
                        ;** ************************************************************************
                        ;*              - SPLIT SPEED TRANSMISSION/RECEPTION SET-UP -
                        ;*
                        ;*              This routine is used to initialise both the DUT and the
                        ;*              transmission/reception control parameters to the correct
                        ;*              state, prior to split speed transmission/reception.
                        ;*
                        ;* INPUTS:      R0 - TX,RX LPR contents for lines in group II.
                        ;*              R1 - TX,RX LPR contents for lines in group I.
                        ;*              R2 - Start address of data pattern to transmit.
                        ;*              R3 - Number of time data pattern to be TX on lines in LINGRP1.
                        ;*              R4 - Number of time data pattern to be TX on lines in LINGRP2.
                        ;*              ACTLNS - Contains a bit map of all currently active lines.
                        ;*              LGRP1M - Contains the bit map of line group I lines.
                        ;*              LOPBCK - Contains the type of loopback mode selected.
                        ;*              CBB - Label at base of TX/RX control block.
                        ;*
                        ;* OUTPUTS:     The contents of the Control Block are destroyed.
                        ;*              The indirect address field of the DUT CSR may be destroyed.
                        ;*              The DUT's LPR's and LNC's may be modified.
                        ;*              The following pointers and counters are initialised;
                        ;*              CHCNT,CHRTOT,DPEND,DPLEN,EXCNT,RXCNT,RXDONF,RXPTR,TXCNT,
                        ;*              TXDONF,TXPTR,TXRXL.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,SPLSUP
                        ;*
                        ;* COMMENTS:    This routine should be called twice during the testing of
                        ;*              the split speed capabilities of the DUT.
                        ;*              So that both line groups are tested on transmission and
                        ;*              reception.
                        ;*              eg.     R1 - LPR contents for lines in LGRP2M,TX=Y,RX=Z baud.
                        ;*                      R2 - LPR contents for lines in LGRP1M,TX=Z,RX=Y baud.
                        ;*                      R3 - Repeat TX on lines in line group 1 = X times.
                        ;*                      R4 - Repeat TX on lines in line group 2 = W times.
                        ;*                JSR   PC,SPLSUP       ; do set-up.
                        ;*                Execute test for the above set-up.
                        ;*                Swap the contents of R1 and R2.
                        ;*                Swap the contents of R3 and R4.
                        ;*                      R1 - LPR contents for lines in LGRP2M,TX=Z,RX=Y baud.
                        ;*                      R2 _ LPR contents for lines in LGRP1M,TX=Y,RX=Z baud.
                        ;*                      R3 - Repeat TX on lines in line group 1 = W times.
                        ;*                      R4 - Repeat TX on lines in line group 2 = X times.
                        ;*                JSR   PC,SPLSUP       ;do set up again.
                        ;*                Execute test again.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: CONMAP,RXDSBL,RXENBL,TXRINI.
                        ;-- ************************************************************************

5923 025052             SPLSUP:: SAVE                   ;SAVE CONTENTS OF THE GPR'S R0 THRU R5.
     025052  004537  005326                     JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
5924 025056  010037  025346             MOV     R0,70$          ;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
```

GLOBAL SUBROUTINE - SPLSUP -

```
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,PREGOS       ;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 PREGOS SUBRT.
 6036 025370  000207                             RTS     PC              ;RETURN
 6037
 6038
 6039
 6040
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - SWAPO -
                        ;** ****************************************************************************
                        ;*                        - Swap GPRs With GPR Set 0 Routine -
                        ;*      This subroutine swaps the present contents of GPRs R1 thru R5 with
                        ;*      the contents of the number zero GPR save area.  The contents of R0
                        ;*      are not altered by this subroutine.
                        ;*
                        ;* INPUTS:      GPR contents R1 thru R5.
                        ;*              GPRS0B - Label at base of GPR save area number zero.
                        ;*
                        ;* OUTPUTS:     R1 thru R5 contain the previous contents of GPR save area
                        ;*                      zero words 1 thru 5 respectively.
                        ;*              GPRS0 - GPR save area 0 words 1 thru 5, contain previous
                        ;*                      contents of GPRs R1 thru R5 respectively.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,SWAP0
                        ;*
                        ;* COMMENTS:    The state of the CARRY flag is not alterred by this routine.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ****************************************************************************

025372  010046          SWAP0::  MOV    R0,-(SP)        ;SAVE THE CONTENTS OF R0.
                        ;*
                        ; Load the stack from the GPRs.
                        ;-
025374  010146                   MOV    R1,-(SP)        ;SAVE THE CONTENTS OF R1.
025376  010246                   MOV    R2,-(SP)        ;SAVE THE CONTENTS OF R2.
025400  010346                   MOV    R3,-(SP)        ;SAVE THE CONTENTS OF R3.
025402  010446                   MOV    R4,-(SP)        ;SAVE THE CONTENTS OF R4.
025404  010546                   MOV    R5,-(SP)        ;SAVE THE CONTENTS OF R5.
                        ;*
                        ; Load the GPRs from the GPR save area 0.
                        ;-
025406  012700  002466           MOV    #GPRS0B,R0      ;GET THE BASE ADDRESS OF GPR SAVE AREA 0.
025412  012001                   MOV    (R0)+,R1        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 1.
025414  012002                   MOV    (R0)+,R2        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 2.
025416  012003                   MOV    (R0)+,R3        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 3.
025420  012004                   MOV    (R0)+,R4        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 4.
025422  012005                   MOV    (R0)+,R5        ;LOAD R1 WITH GPR SAVE AREA 0 WORD 5.
                        ;*
                        ; Load the GPR save area 0 from the stack.
                        ;-
025424  012640                   MOV    (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 5 WITH SAVED R5.
025426  012640                   MOV    (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 4 WITH SAVED R4.
025430  012640                   MOV    (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 3 WITH SAVED R3.
025432  012640                   MOV    (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 2 WITH SAVED R2.
025434  012640                   MOV    (SP)+,-(R0)     ;LOAD GPR SAVE AREA 0 WORD 1 WITH SAVED R1.

025436  012600                   MOV    (SP)+,R0        ;RESTORE THE INITIAL VALUE OF R0.

025440  000207                  RTS    PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXDONE -
                        ;* ************************************************************************
                        ;*                          - TRANSMISSION DONE -
                        ;*      This subroutine is used in the transmission/reception tests to allow
                        ;*      time for transmission to complete on outstanding lines.
                        ;*
                        ;* INPUTS:      ACTLNS - Contains bit map of all active lines.
                        ;*              TXDONF - TX done flags, set for lines that have sent all chars.
                        ;*              CHCNT - Table contening the number of chars to be TX'd.
                        ;*
                        ;*
                        ;* OUTPUTS:     Transmission interrupts are disabled.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TXDONE
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: MSLOOP,MUL16U.
                        ;- ************************************************************************

025442                  TXDONE:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
025442  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; Check if all active lines have completed transmission.
                        ;  If any have not yet completed, determine the tx char count for a
                        ;  line that has outstanding characters to transmit.  Using this value,
                        ;  calculate the time-out value needed at the currently selected baud rate.
                        ;-
025446  013703  002174          MOV     ACTLNS,R3       ;GET THE ACTIVE LINE BIT MAP.
025452  013702  002504          MOV     TXDONF,R2       ;GET THE BIT MAP OF LINES THAT HAVE COMPLETED.
025456  040203                  BIC     R2,R3           ;GENERATE A BIT MAP OF LINES THAT ARE STILL TX.
025460  005703                  TST     R3              ;CHECK IF ALL LINES HAVE COMPLETED TX.
025462  001427                 BEQ     6$              ;GO DISABLE TX INTERRUPTS IF ALL DONE.
                          ;+
                          ; Find a line that has not completed transmission.
                          ; Obtain the expected character count for that line (which is the same for
                          ; all other lines with outstanding tx's).
                          ; Calculate time-out value.
                          ;-
025464  005004                  CLR     R4              ;CLEAR LINE NUMBER COUNTER.
025466  012702  000001          MOV     #1,R2           ;SELECT BIT MAP FOR THE FIRST LINE.
025472  030203          2$:     BIT     R2,R3           ;SEE IF THIS LINE HAS COMPLETED.
025474  001003                  BNE     4$              ;BRANCH IF THIS LINE HAS NOT COMPLETED TX.
025476  006102                  ROL     R2              ;SHIFT THE LINE BIT MAP FOR THE NEXT LINE.
025500  005204                  INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
025502  000773                 BR      2$              ;LOOP TO CHECK THE NEXT LINE.
025504  006304          4$:     ASL     R4              ;LINE NUMBER X 2 TO OBTAIN OFFSET INTO TABLE.
025506  016401  003444          MOV     CHCNTB(R4),R1   ;GET THE EXPECTED NUMBER OF CHARS TO BE TX'D.
025512  013702  002242          MOV     RXTOUT,R2       ;GET THE CURRENT TIME-OUT VALUE FOR ONE CHAR.
025516  004737  022030         JSR     PC,MUL16U       ;(NUMBER OF CHARS TO TX) X (TIME-OUT OF 1 CHAR)
025522  006301                  ASL     R1              ;MULTIPLY DELAY TIME BY 2 TO GIVE A SAFE VALUE.
                          ;+
                          ; Wait for all oustanding transmissions to complete or time-out.
                          ; Disable all transmission interrupts.
                          ;
025524  013702  002174          MOV     ACTLNS,R2       ;PASS A BIT MAP OF THE BITS TO TEST.
025530  010203                  MOV     R2,R3           ;PASS THE EXPECTED STATE OF THE TXDONF.
```

GLOBAL SUBROUTINE - TXDONE - SEQ 0160

```
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$:       JSR     PC,TXIE0        ;DISABLE ALL TX INTERRUPTS.
6155
6156 025546                          60$:      PASS                    ;RESTORE GPRS.
     025546  004736                                      JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
6157 025550  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXDSBL -
                        ;** ************************************************************************
                        ;*                        - Transmitter Disable -
                        ;*      This subroutine is used to disable transmission on selected lines by,
                        ;*      clearing the associated TX.ENABLE bit on the DUT.
                        ;*
                        ;* INPUTS:      R5 - Bit's set correspond to lines on which to clear TX.ENABLE.
                        ;*              CSRA - Contains the address of the DUT CSR.
                        ;*              IESTAT - Contains the state of TXIE and RXIE bits in the CSR.
                        ;*              NUMLNS - Equated to be the maximum number of lines available.
                        ;*              TXAD2A - Contains the address of the TBUFFAD2 register.
                        ;*
                        ;* OUTPUTS:     R5 - Bit's set indicate the initial states of all TX.ENBLE bits.
                        ;*              TBUFFAD2   The state of the TX.ENBLE bit may be altered.
                        ;*              The contents of the IND.ADD.REG field in the CSR are destroyed.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TXDSBL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

025552                  TXDSBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
025552  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
025556  010500          MOV     R5,R0           ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
025560  012701  000001  MOV     #BIT0,R1        ;INITIALIZE THE SELECTED LINE BIT MASK.
025564  013702  002216  MOV     TXAD2A,R2       ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
025570  005202          INC     R2              ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
025572  012703  000010  MOV     #NUMLNS,R3      ;GET MAXIMUM LINE NUMBER PLUS ONE.
025576  013704  002234  MOV     IESTAT,R4       ;GET THE STATES OF THE INT ENABLE BITS.
025602  005005          CLR     R5              ;LOG POSSIBLE TX DISABLED ON ALL LINES.
                        ;+
                        ; Select every line in turn, and log the state of each TX.ENABLE bit.
                        ;-
025604  010477  154372  2$:     MOV     R4,@CSRA        ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
025610  105712          TSTB    (R2)            ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
025612  100001          BPL     4$              ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
025614  050105          BIS     R1,R5           ;LOG TX ENABLE BIT SET FOR SELECTED LINE.
                        ;+
                        ;  Clear TX.ENBLE on lines that have a corresponding bit set in the tx disable
                        ;  line bit map.
                        ;-
025616  030100          4$:     BIT     R1,R0           ;CHECK STATE OF DISABLE LINE BIT MAP.
025620  001402          BEQ     6$              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
025622  142712  000200          BICB    #BIT7,(R2)      ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
025626  005204          6$:     INC     R4              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
025630  006301          ASL     R1              ;SHIFT BIT MAP FOR NEXT LINE.
025632  005303          DEC     R3              ;DECREMENT LINE NUMBER.
025634  001363          BNE     2$              ;LOOP TO CHECK NEXT LINE.

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 PREG05 SUBRT.
                                                        ;R5 - PREVIOUS STATES OF ALL TX.ENABLE BITS.
025644  000207          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - TXENBL -
                        ;** ****************************************************************************
                        ;*                       - Transmitter Enable -
                        ;*      This subroutine is used to enable transmission on selected lines by
                        ;*      setting the associated TX.ENABLE bit on the DUT.
                        ;*
                        ;* INPUTS:      R5 - Bit's set correspond to lines on which to set TX ENABLE.
                        ;*              CSRA - Contains the address of the DUT CSR.
                        ;*              IESTAT - Contains the state of TXIE and RXIE bits in the CSR.
                        ;*              NUMLNS - Equated to be the maximum number of lines available.
                        ;*              TXAD2A - Contains the address of the TBUFFAD2 register.
                        ;*
                        ;* OUTPUTS:     R5 - Bit's set indicate previously disabled lines.
                        ;*              TBUFFAD2 - The state of the TX.ENBLE bit may be altered.
                        ;*              The contents of the IND.ADD.REG field in the CSR are destroyed.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TXENBL
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ****************************************************************************

025646                  TXENBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
025646  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
025652  010500          MOV     R5,R0           ;COPY BIT MAP OF LINES TO ENABLE.
025654  012701  000001  MOV     #BIT0,R1        ;INITIALIZE THE SELECTED LINE BIT MASK.
025660  013702  002216  MOV     TXAD2A,R2       ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
025664  005202          INC     R2              ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
025666  012703  000010  MOV     #NUMLNS,R3      ;GET MAXIMUM LINE NUMBER.
025672  013704  002234  MOV     IESTAT,R4       ;GET THE STATES OF THE INT ENABLE BITS.
025676  005005          CLR     R5              ;CLEAR TX.ENABLE BIT LOG OF DISABLED LINES.
                        ;+
                        ; Select every line in turn,and log any TX.ENBLE bit that is clear.
                        ;-
025700  010477  154276  2$:     MOV     R4,@CSRA        ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
025704  105712          TSTB    (R2)            ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
025706  100401          BMI     4$              ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
025710  050105          BIS     R1,R5           ;LOG TX ENABLE BIT CLEAR FOR SELECTED LINE.
                        ;+
                        ;  Set TX.ENBLE on lines that have a corresponding bit set in the tx enable
                        ;  line bit map.
                        ;-
025712  030100          4$:     BIT     R1,R0           ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
025714  001402          BEQ     6$              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
025716  152712  000200  BISB    #BIT7,(R2)      ;ENABLE TRANSMISSION ON SELECTED LINE.
025722  005204          6$:     INC     R4              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
025724  006301          ASL     R1              ;SHIFT BIT MAP FOR NEXT LINE.
025726  005303          DEC     R3              ;DECREMENT LINE NUMBER.
025730  001363          BNE     2$              ;LOOP TO CHECK NEXT LINE.

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 PREG05 SUBRT.
                                                        ;R5 - LINE BIT MAP CORRESPONDING TO THE
                                                        ; PREVIOUS LINES THAT WERE DISABLED.
025740  000207          RTS     PC
```

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

GLOBAL SUBROUTINE - TXFRPR -

```
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
```

GLOBAL SUBROUTINE - TXIE0 - SEQ 0165

```
                                .SBTTL  GLOBAL SUBROUTINE               - TXIE0 -
                                ;** ************************************************************************
                                ;*                      - TRANSMITTER INTERRUPT DISABLE -
                                ;*      This routine is used to disable transmitter interrupts in the DHV11-M.
                                ;*
                                ;* INPUTS:      NONE.
                                ;*
                                ;* OUTPUTS:     The TX.INT.ENBL bit is cleared in the DUT CSR.
                                ;*              IESTST -contains the updated status of the TX and RX interrupt
                                ;*              enable bits.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TXIE0
                                ;*
                                ;* COMMENTS:    The contents of the indirect address register field in
                                ;*              the DUT CSR are destroyed.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ************************************************************************
026036  010046                  TXIE0:: MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
026040  106746                          MFPS    -(SP)           ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
026042  106427  000340                  MTPS    @PRI07          ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
026046  042737  177677  002234          BIC     #177677,IESTAT  ;CLEAR TX.INT.ENBL BIT IN IESTAT.
026054  013777  002234  154120          MOV     IESTAT,@CSRA    ;DISABLE TX INTERRUPTS.
026062  106426                          MTPS    (SP)+           ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
026064  012600                          MOV     (SP)+,R0        ;RESTORE R0.
026066  000207                          RTS     PC
```

GLOBAL SUBROUTINE - TXIE1 -

```
                                .SBTTL  GLOBAL SUBROUTINE               - TXIE1 -
                                ;++ ****************************************************************************
                                ;*                          - TRANSMITTER INTERRUPT ENABLE -
                                ;*      This routine is used to enable transmitter interrupts in the DHV11-M.
                                ;*
                                ;* INPUTS:       NONE.
                                ;*
                                ;* OUTPUTS:      The TX.INT.ENBL bit is set in the DUT CSR.
                                ;*               IESTST -contains the updated status of the TX and RX interrupt
                                ;*               enable bits.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TXIE1
                                ;*
                                ;* COMMENTS:     The contents of the indirect address register field in
                                ;*               the DUT CSR are destroyed.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ****************************************************************************

026070  052737  040000  002234  TXIE1::  BIS     #BIT14,IESTAT   ;SET TX.INT.ENBL BIT IN IESTAT.
026076  042737  137677  002234          BIC     #137677,IESTAT  ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
026104  013777  002234  154070          MOV     IESTAT,@CSRA    ;DISABLE TX INTERRUPTS.
026112  000207                         RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - TXRINI -
                        ;** ************************************************************************
                        ;*              - Transmit and Receive Initialization Routine -
                        ;*      This subroutine performs the initialization of the various pointers,
                        ;*      counters, and flags which are used during the transmission and
                        ;*      reception portion of a test.  This initialization is performed on
                        ;*      the specified lines only, other line variables remain unchanged.
                        ;*
                        ;* INPUTS:      CHCNTB - Label at base of line character count table.
                        ;*              CHRTOT - Max # of chars to RX on lines already initialized.
                        ;*              DPENDB - Label at base of line data pattern end table.
                        ;*              DPLENB - Label at base of line data pattern length table.
                        ;*              EXCNTB - Label at base address of extra char counters table.
                        ;*              IESTAT - Present state of the RX.IE and TX.IE bits.
                        ;*              NUMLNS - Equated to number of lines on the DUT.
                        ;*              RXCNTB - Label at base address of RX character counters table.
                        ;*              RXPTRB - Lable at base adr of "next RX char" pointers table.
                        ;*              TXCNTB - Label at base address of TX character counters table.
                        ;*              TXPTRB - Label at base adr of "next TX char" pointers table.
                        ;*              CBB - Label at base of TX/RX control block.
                        ;*              CB Contents - TX/RX control block contains the following:
                        ;*                  CBLPRA - DUT LPR contents.
                        ;*                  CBLNCA - DUT LNCTRL contents.
                        ;*                  CBDPAA - Address of beginning of data pattern.
                        ;*                  CBDPLA - Length in bytes of data pattern.
                        ;*                  CBDPNA - Number of data patterns to transmit.
                        ;*                  CBMAPA - Bit map of lines to be initialized.
                        ;*                  CBLPBA - Type of loopback to be used for test.
                        ;*                  CBOFSA - Amount to offset each TX start in the data pat.
                        ;*              TXRXLB - Label at base of TX/RX line association table.
                        ;*
                        ;* OUTPUTS:     CHCNT - Table of number of line TX characters (Initialized).
                        ;*              CHRTOT - Maximum number of chars to receive (2 * pat length).
                        ;*              DPEND - Table of data pattern ends (Initialized).
                        ;*              DPLEN - Table of data pattern lengths (Initialized).
                        ;*              DUT LNCTRL - Line control registers (Initialized).
                        ;*              DUT LPR - Line parameter registers (Initialized).
                        ;*              EXCNT - Table of extra RX char counts (Clred, selected lines).
                        ;*              RXCNT - Table of RX character counts (Clred, selected lines).
                        ;*              RXDONF - "Reception Done" flags (Cleared for selected lines).
                        ;*              RXPTR - Table of receive pointers (Initialized).
                        ;*              TXCNT - Table of TX character counters (Clred, selected lines).
                        ;*              TXDONF - "Transmission Done" flags (Clred for selected lines).
                        ;*              TXPTR - Table of transmit pointers (Initialized).
                        ;*              TXRXL - TX/RX line association table (Initialized).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TXRINI
                        ;*
                        ;* COMMENTS:    If the calculation of the CHRTOT value (2 times the data
                        ;*              pattern length) results in a number greater than 64K then
                        ;*              CHRTOT is initialized to 64K - 1.
                        ;*              This routine will not force internal loopback based on the
                        ;*              loopback type in CBLPBA.  The user must set up CBLNCA correctly
                        ;*              to get internal loopback.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: WTWLNC,WTWLPR,
                        ;-- ************************************************************************
```

GLOBAL SUBROUTINE - TXRINI - SEQ 0168

```
 6455 026114                            TXRINI:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
      026114  004537  005326                      JSR     R5,PREG05       ;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,R0       ;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,R0         ;SET INTERNAL LOOPBACK IN MAINTENANCE FIELD.
 6464 026144  004737  027234          2$:       JSR     PC,WTWLNC       ;SET UP THE LNCTRL REGS FOR SELECTED LINES.
 6465 026150  013700  003124                      MOV     CBLPRA,R0       ;GET THE NEW LPR CONTENTS.
 6466 026154  004737  027264                    JSR     PC,WTWLPR       ;SET UP THE LPR REGISTERS FOR SELECTED LINES.
 6467 026160  004737  025646                    JSR     PC,TXENBL       ;ENABLE TX FOR ALL SELECTED LINES.
 6468                                   ;+
 6469                                   ; Set up and begin loop which handles one line per iteration.
 6470                                   ;-
 6471 026164  005004                            CLR     R4              ;CLEAR THE LINE OFFSET.
 6472 026166  013705  003130                      MOV     CBDPAA,R5       ;INITIALIZE THE TX START ADDRESS VALUE.
 6473 026172  013703  003132                      MOV     CBDPLA,R3       ;GET THE LENGTH OF THE DATA PATTERN.
 6474 026176  060503                            ADD     R5,R3           ;CALCULATE END ADDRESS OF THE DATA PATTERN.
 6475 026200  036437  002366  003136  4$        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      LINE.
 6502                                       ;+
 6503                                       ; Set up the TX/RX line association offset table entry for      .ine.
 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.
```

GLOBAL SUBROUTINE - TXRINI - SEQ 0169

```
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          12$:   ;-
6522 026340  020503                  14$:       ADD     CBOFSA,R5       ;ADD THE TX OFFSET TO THE TX START POINTER.
6523 026342  103403                             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                  16$:   ;-
6530 026354  005204                             INC     R4
6531                                            INC     R4
6532                                        ;+
6533                                        ; Test for done handling all possible lines on the device.
6534 026356  020427  000020                ;-
6535 026362  002706                             CMP     R4,#NUMLNS*2    ;COMPARE OFFSET WITH 2 TIMES MAX # OF LINES.
6536                                          BLT     4$              ;LOOP IF NOT ALL LINES DONE.
6537 026364                          60$:
     026364  004736                             PASS                    ;RESTORE GPRS.
6538 026366  000207                                     JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
                                                RTS     PC
```

GLOBAL SUBROUTINE - TXROFF - SEQ 0170

```
 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 setif 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    #PRI05          ;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
```

GLOBAL SUBROUTINE - TXRON - SEQ 0171

```
 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,PREG05       ;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 PREG05 SUBRT.
 6596 026446  000207                        RTS     PC
```

GLOBAL SUBROUTINE - TXRREP - SEQ 0172

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXRREP -
                        ;** ************************************************************************
                        ;*                      - Report Final TX/RX Errors Routine -
                        ;*      This subroutine reports errors which are found after the completion
                        ;*      of the TX, RX, and verification of data patterns.  It reports errors
                        ;*      dealing with incomplete TX or RX and with DMA_START bits.
                        ;*
                        ;* INPUTS:      ACTLNS - Bit map of active DUT lines.
                        ;*              DPLENB - Label at base of the data pattern lengths table.
                        ;*              ERRMSG - Address of primary error message for this routine.
                        ;*              ERRNBR - Error number of error reported in this routine.
                        ;*              RXCNTB - Label at base of the RX character counters table.
                        ;*              RXDONF - Reception done flags.
                        ;*              TXCNTB - Label at base of the TX character counters table.
                        ;*              TXDONF - Transmission done flags.
                        ;*              TXINTF - Contains bit map of lines with DMA_START bit errors.
                        ;*
                        ;* OUTPUTS:     CARRY FLAG - Restored to its entering value.
                        ;*              ERRBLK - Address of the error reporting routine (Destroyed).
                        ;*              Messages may be printed at the operator console.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TXRREP
                        ;*
                        ;* COMMENTS:    This routine reports errors at Initial ERRNBR thru
                        ;*              Initial ERRMBR+2.
                        ;*              If no lines failed to complete their reception or failed to
                        ;*              complete their transmission or had DMA_START bit errors
                        ;*              then no messages are printed.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: CONMAP,ER9005,ER9102,RDMAST,RRXNDN,RTXNDN.
                        ;-- ************************************************************************

026450                  TXRREP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
026450  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
026454  006003            ROR     R3      ;ROTATE CARRY INTO GPR TO SAVE CARRY STATE.
026456  013704  005320    MOV     ERRNBR,R4       ;SAVE THE INITIAL ERROR NUMBER VALUE.
026462  013705  002174    MOV     ACTLNS,R5       ;GET THE ACTIVE LINES BIT MAP.
026466  004737  024034    JSR     PC,RDMAST       ;REPORT ANY DMA_START BIT ERRORS.
026472  005237  005320    INC     ERRNBR          ;SELECT INTIAL ERROR NUMBER + 1.
026476  004737  024420    JSR     PC,RTXNDN       ;REPORT TX NOT COMPLETE IF NECESSARY.
026502  005237  005320    INC     ERRNBR          ;SELECT INITIAL ERROR NUMBER + 2.
026506  004737  017300    JSR     PC,CONMAP       ;GENERATE AN ASSOCIATED LINE BIT MAP.
026512  004737  024360    JSR     PC,RRXNDN       ;REPORT RX NOT COMPLETE IF NECESSARY.
026516  010437  005320    MOV     R4,ERRNBR       ;RESTORE THE INITIAL ERROR NUMBER VALUE.

026522  006103            ROL     R3              ;ROTATE SAVED CARRY STATE BACK INTO CARRY.
026524                  60$:    PASS                    ;RESTORE GPRS, THIS ROUTINE PRESERVES THE
026524  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
026526  000207            RTS     PC              ;  INITIAL CARRY STATE.
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - UNSDIV -
                        ;++ ******************************************************************************
                        ;*                      - Unsigned Divide Routine -
                        ;*      This subroutine is used to divide a 32 bit unsigned dividend by a
                        ;*      16 bit unsigned divisor giving a 16 bit quotient.  All numbers are
                        ;*      considered to be unsigned.  A success flag is not set on return if
                        ;*      the quotient was too big to be contained in 16 bits.
                        ;*
                        ;* INPUTS:      R1 - The divisor, unsigned, 16 bits.
                        ;*              R2 - Most significant word of the dividend, unsigned, 16 bits.
                        ;*              R3 - Least significant word of the dividend, unsigned, 16 bits.
                        ;*
                        ;* OUTPUTS:     R1 - Quotient, unsigned, 16 bits (177777 if overflow).
                        ;*              CARRY - Success flag, set if complete quotient fits in 16 bits.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,UNSDIV
                        ;*
                        ;* COMMENTS:    If the divisor is 0 the quotient is returned as all ones
                        ;*              (177777) and the carry is clear regardless of the dividend.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ******************************************************************************

026530                  UNSDIV:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
026530  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; Check for quotient greater than 16 bits condition.
                        ;-
026534  010204                  MOV     R2,R4           ;GET MSW OF DIVIDEND FOR SUBTRACT.
026536  160104                  SUB     R1,R4           ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
026540  103403                  BCS     2$              ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
026542  012701  177777          MOV     #-1,R1          ;SET QUOTIENT TO ALL ONES (177777),
026546  000442                  BR      60$             ;EXIT WITH CARRY CLEAR.
                        ;+
                        ; Set up counters and various working GPRs.
                        ;-
026550  005004          2$:     CLR     R4              ;CLEAR THE LSW OF THE DIVISOR.
026552  000241                  CLC                     ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
026554  006001                  ROR     R1              ; DIVISOR BY
026556  006004                  ROR     R4              ; 2(UNSIGNED)
026560  012700  000020          MOV     #16.,R0         ;SET UP INITIAL SHIFT COUNT TO 16.
                        ;+
                        ; The subtract and shift loop.
                        ;-
026564  010246          4$:     MOV     R2,-(SP)        ;SAVE MSWORD OF DIVIDEND.
026566  010346                  MOV     R3,-(SP)        ;SAVE LSWORD OF DIVIDEND.
026570  160403                  SUB     R4,R3           ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
026572  005602                  SBC     R2              ;MSWORD DIVIDEND - BORROW .
026574  103402                  BCS     6$              ;IF BORROW FROM BORROW SUBTRACT, IT DIDN T GO.
026576  160102                  SUB     R1,R2           ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
026600  103003                  BCC     8$              ;IF NO BORROW, IT WENT, CARRY IS CLEAR.
                        ;+
                        ; It didn't go, so we shift a 1 into the quotient (complemented later).
                        ; Carry is set.
                        ;-
026602  012603          6$:     MOV     (SP)+,R3        ;RESTORE LSWORD OF DIVIDEND.
026604  012602                  MOV     (SP)+,R2        ;RESTORE MSWORD OF DIVIDEND.
```

```
6702 026606  000401                    BR      10$             ;GOTO SHIFT 1 INTO THE QUOTIENT.
6703                                  ;+
6704                                  ; It went, so we restore the stack and shift a 0 into quot'ent (will be
6705                                  ; complemented later).   Carry is clear.
6706                                  ;-
6707 026610  012626                8$:     MOV     (SP)+,(SP)+     ;POP THE SAVED DIVIDEND OFF OF THE STACK.
6708                                  ;+
6709                                  ; Shift the result of the subtract attempt into the quotient shift reg.
6710                                  ;-
6711 026612  006105                10$:    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
     026654  010166  000004                        MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
     026660  004736                        JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
6739                                                                          ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
6740 026662  000207                        RTS     PC                      ;R1 - 16 BIT, UNSIGNED QUOTIENT,
                                                                            ;CARRY - SET INDICATES NO OVERFLOW (SUCCESS).
```

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

SEQ 0176

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 PREGO5 SUBRT.
 6801 026760  000207                          RTS     PC
```


SEQ 0177

GLOBAL SUBROUTINE - VANSUP -

```
                        .SBTTL  GLOBAL SUBROUTINE               - VANSUP -
                        ;++ ************************************************************************
                        ;*              - TRANSMISSION / RECEPTION SET-UP ROUTINE -
                        ;*
                        ;*              This routine is used to initialise both the DUT and the
                        ;*              transmission/reception control parameters to the correct
                        ;*              state, prior to a Single character or DMA transmission,
                        ;*              reception test.
                        ;*
                        ;* INPUTS:      R1 - TX, RX LPR contents.
                        ;*              R2 - Start address of data pattern to transmit.
                        ;*              R3 - Length of data pattern.
                        ;*              R4 - Number of patterns to transmit.
                        ;*              ACTLNS - Contains a bit map of all currently active lines.
                        ;*              LOPBCK - Contains the type of loopback mode selected.
                        ;*              CBB - Label at base of TX/RX control block.
                        ;*
                        ;* OUTPUTS:     The contents of the TX/RX control block (CCB) are destroyed.
                        ;*              The indirect address field of the DUT CSR may be destroyed.
                        ;*              The DUT's LPR's and LNC's may be modified.
                        ;*              The following pointers and counters are initialised;
                        ;*              CHCNT,CHRTOT,DPEND,DPLEN,EXCNT,RXCNT,RXPTR,TXCNT,
                        ;*              TXPTR,TXRXL.
                        ;*              CHRTOT, RXDONF, TXDONF and TXINTF are cleared.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,VANSUP
                        ;*
                        ;* COMMENTS:    Modem loopback mode is inhibited if it has been selected
                        ;*              via Hardware P-table questions, and internal loopback mode
                        ;*              is forced to take place.
                        ;*
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: CONMAP,RXENBL,TXRINI.
                        ;-- ************************************************************************

026762                  VANSUP:: SAVE                   ;SAVE CONTENTS OF THE GPR'S R0 THRU R5.
026762  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
026766  005037  002500          CLR     CHRTOT          ;CLEAR TOTAL RECEIVED CHAR COUNTER.
026772  005037  002252          CLR     TXINTF          ;CLEAR FLAGS USED TO LOG DMA H.OVER ERRORS.
026776  005037  002504          CLR     TXDONF          ;CLEAR THE TX DONE FLAGS.
027002  005037  002506          CLR     RXDONF          ;CLEAR THE RX DONE FLAGS.
                        ;+
                        ;  Set up the Transmission/Reception Control block to the desired state.
                        ;-
027006  010137  003124          MOV     R1,CBB          ;SET CONTENTS OF LPR PARAMS IN TX/RX C.BLK.
027012  012701  003124          MOV     #CBB,R1         ;GET BASE ADDRESS OF CONTROL BLOCK.
027016  005201                  INC     R1              ;INCREMENT ADDRESS FOR NEXT WORD
027020  005201                  INC     R1              ;INITIALISE THE FOLLOWING IN THE CNTRL.BLK:
027022  012721  000004          MOV     #4,(R1)+        ; LNCTRL PARAMETER, ENABLE RECEIVERS.
027026  010221                  MOV     R2,(R1)+        ; START ADDRESS OF DATA PATTERN.
027030  010321                  MOV     R3,(R1)+        ; DATA PATTERN LENGTH.
027032  010421                  MOV     R4,(R1)+        ; NUMBER OF DATA PATTERNS TO TRANSMIT.
027034  013721  002174          MOV     ACTLNS,(R1)+    ; BIT MAP OF LINES TO INITIALISE.
027040  032737  000004  002176  BIT     #BIT2,LOPBCK    ;TEST IF MODEM LOOPBACK MODE HAS BEEN SELECTED.
027046  001404                  BEQ     2$              ;DONT SELECT INTERNL LOPBCK IF STAGRD OR LOCAL.
027050  012702  000001          MOV     #1,R2           ;FORCE INTERNAL LOOPBACK MODE TO BE SELECTED.
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 PREGOS SUBRT.
6892 027156  000207            RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - WAIBIS -
                        ;** ************************************************************************
                        ;*                      - Wait For Bit Set Routine -
                        ;*      This subroutine waits for the specified bit to become set.  If the
                        ;*      specified bit goes to a set state within the specified time-out
                        ;*      period a success indication is returned by this routine.
                        ;*      The last value which is read looking for the condition is returned to
                        ;*      allow the use of this routine to look for destructive read conditions.
                        ;*
                        ;* INPUTS:      R1 - Time-out value and bit number indication:
                        ;*                Bits 15 thru 12 - Number of bit to test (range 0 thru 15).
                        ;*                Bits 11 thru 0  - Time-out value in milli-seconds (4095 max).
                        ;*              R2 - Address of word containing the bit to test.
                        ;*                   MSLCNT.
                        ;*
                        ;* OUTPUTS:     R2 - The last word which was read to check for the condition.
                        ;*              CARRY - Success flag (CARRY set if bit set before time-out).
                        ;*
                        ;* CALLING SEQUENCE:    MOV     #130040,R1      ;PASS BIT 11 (13 OCTAL) AND
                        ;*                                              ;  32 (40 OCTAL) MS DELAY.
                        ;*                      MOV     @LABEL,R2       ;TEST BIT IN WORD AT "LABEL".
                        ;*                      JSR     PC,WAIBIS       ;WAIT 32 MS FOR BIT 11 TO SET.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: MSLGET.
                        ;-- ************************************************************************

027160                  WAIBIS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
027160  004537  005326                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
027164  010204                MOV     R2,R4           ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
027166  010102                MOV     R1,R2
027170  042701  170000        BIC     #170000,R1      ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
027174  042702  007777        BIC     #7777,R2        ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
027200  000302                SWAB    R2              ;PUT LINE NUMBER FIELD IN LSBYTE.
027202  006202                ASR     R2              ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
027204  006202                ASR     R2              ;  POSITION TO USE IT AS A WORD TABLE OFFSET
027206  006202                ASR     R2              ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
027210  016202  002366        MOV     BITTBL(R2),R2   ;GET BIT MAP OF LINE TO TEST FROM TABLE.
027214  010203                MOV     R2,R3           ;INDICATE THAT THE BIT SHOULD BE SET.
027216  004737  021464        JSR     PC,MSLGET       ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
                                                        ;  CARRY IS CORRECT UPON MSLGET RETURN.
027222  010002                MOV     R0,R2           ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
027224                  60$:    PASS    R2              ;RESTORE GPRS, EXCEPT THE FOLLOWING:
027224  010266  000006                        MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
027230  004 >6                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                        ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
027232  000207                RTS     PC              ;  CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).
```

GLOBAL SUBROUTINE - WTWLNC -

```
                        .SBTTL  GLOBAL SUBROUTINE               - WTWLNC -
                        ;++ ******************************************************************************
                        ;*                      - Line Control Register Setup Routine -
                        ;*      This subroutine is used to set the Device Under Test (DUT) Line
                        ;*      Control Registers (LNCTRL) to the specified state.  Only the LNCTRLS
                        ;*      for the specified lines are altered.
                        ;*
                        ;* INPUTS:      R0 - New line parameters.
                        ;*              R5 - Bit map of lines to be altered.
                        ;*              CSRA - Contains address of the DUT CSR.
                        ;*              IESTAT - Contains the current state of the TX and RX interrupt
                        ;*               enable bits in the CSR.
                        ;*              LNCTRA - Contains address of the DUT LNCTRL registers.
                        ;*
                        ;* OUTPUTS:     LNCTRL - Specified DUT Line Control Registers are altered.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WTWLNC
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
                        ;-- ******************************************************************************

027234                  WTWLNC:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
027234  004537  005326                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; Set up the parameters for the call to ALTFLD.
                        ;-
027240  013701  002212          MOV     LNCTRA,R1       ;SET UP THE REGISTER ADDRESS PARAMETER.
027244  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
027246  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
027250  012704  177777          MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                        ;+
                        ; Call the subroutine which alters the register contents.
                        ;-
027254  004737  015722          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

027260                  60$:    PASS                    ;RESTORE GPRS.
027260  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
027262  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE               - WTWLPR -
                        ;** ************************************************************************
                        ;*                      - Line Parameter Register Setup Routine -
                        ;*      This subroutine is used to set the Device Under Test (DUT) Line
                        ;*      Parameter Registers (LPR) to the specified state.  Only the LPRs for
                        ;*      the specified lines are altered.
                        ;*
                        ;* INPUTS:      R0 - New line parameters.
                        ;*              R5 - Bit map of lines to be altered.
                        ;*              CSRA - Contains address of the DUT CSR.
                        ;*              IESTAT - Contains the current state of the TX and RX interrupt
                        ;*               enable bits in the CSR.
                        ;*              LPRA - Contains address of the DUT LPR.
                        ;*
                        ;* OUTPUTS:     LPR - Specified DUT Line Parameter Registers are altered.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WTWLPR
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
                        ;-- ************************************************************************

027264                  WTWLPR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
027264  004537  005326                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; Set up the parameters for the call to ALTFLD.
                        ;-
027270  013701  002206          MOV     LPRA,R1         ;SET UP THE REGISTER ADDRESS PARAMETER.
027274  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
027276  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
027300  012704  177777          MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                        ;+
                        ; Call the subroutine which alters the register contents.
                        ;-
027304  004737  015722          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

027310                  60$:    PASS                            ;RESTORE GPRS.
027310  005736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
027312  000207                  RTS     PC
```

```
                                .SBTTL  INTERRUPT SERVICE ROUTINE       - CLKINT -
                                ;++ ************************************************************************
                                ;*      This routine is executed CLKHRZ times per second.  It decrements the
                                ;*      two timer counters down to zero.
                                ;*
                                ;* INPUTS:      TIMER1 - Timer counter #1.
                                ;*              TIMER2 - Timer counter #2.
                                ;*              TIMER3 - Timer counter for call of BREAK macro.
                                ;*
                                ;* OUTPUTS:     The 2 timer counters are decremented if they are not zero.
                                ;*
                                ;* CALLING SEQUENCE:    Put @CLKINT in the clock interrupt vector slot.
                                ;*                      Put the desired time period (seconds times CLKHRZ) in
                                ;*                      either TIMER1 or TIMER2 and poll the respective timer
                                ;*                      counter to detect its going to 0 on time-out.
                                ;*
                                ;* COMMENTS:    The 2 counters will not wraparound but will stop at 0.  This
                                ;*              allows the detection of a time-out any time after the time-out
                                ;*              has occurred until the timer counter is set to another value.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;-- ************************************************************************

027314  005737  002302        CLKINT:: TST     TIMER1          ;CHECK FOR TIMER1 AT ZERO.
027320  001402                         BEQ     2$              ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
027322  005337  002302                 DEC     TIMER1          ;DECREMENT TIME COUNT.
027326  005737  002304        2$:      TST     TIMER2          ;CHECK FOR TIMER2 AT ZERO.
027332  001402                         BEQ     4$              ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
027334  005337  002304                 DEC     TIMER2          ;DECREMENT TIME COUNT.
027340  005337  002306        4$:      DEC     TIMER3          ;DECREMENT THE BREAK COUNT.
027344  001006                         BNE     60$             ;EXIT IF NOT TIME TO CALL BREAK.
027346  013737  002310  002306        MOV     BCOUNT,TIMER3   ;SET UP TIME TILL NEXT BREAK.
027354  010046                         MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 FROM BREAK MACRO.
027356                                 BREAK                   ;CHECK FOR OPERATOR CONTROL/C.
027356  104422                                                                 TRAP    C$BRK
027360  012600                         MOV     (SP)+,R0        ;RESTORE CONTENTS OF R0.
027362  000002                 60$:    RTI
```

```
                        .SBTTL  INTERRUPT SERVICE ROUTINE       - RXCHRS -
                        ;** ****************************************************************************
                        ;*              - DMA RECEIVE Interrupt Service Routine -
                        ;*      This routine executes in response to an interrupt caused by the DUT
                        ;*      RX.DATA.AVAIL bit becoming active.  This routine reads characters from
                        ;*      the DUT receive character FIFO and deposits them into the receive
                        ;*      buffer in memory.  If the number of characters in the receive buffer
                        ;*      exceeds a specified threshold, transmission is halted (by clearing all
                        ;*      DUT TX.ENABLE bits) and if the receive buffer is full reception is
                        ;*      halted (by disabling RX interrupts).  The routine exits if the receive
                        ;*      buffer becomes full or if a character is read from the FIFO with the
                        ;*      DATA.VALID bit clear.
                        ;*
                        ;* INPUTS:      RBUFA - Contains address of the DUT RX character FIFO.
                        ;*              RXBCNT - RX buffer character count.
                        ;*              RXBDTX - Equated to RX buffer level at which to disable TX.
                        ;*              RXBEND - Label after end of the RX buffer area in memory.
                        ;*              RXBFUL - Equated to the capacity of the RX buffer.
                        ;*              RXBIPT - Pointer to next available input slot of RX buffer.
                        ;*              RXBSTA - Label at start of RX buffer area in memory.
                        ;*
                        ;* OUTPUTS:     RXBIPT - Updated to point to next input slot of RX buffer.
                        ;*              RXBCNT - RX buffer character count (Incremented).
                        ;*              TXENBM - Map of previous DUT TX.ENABLE states.
                        ;*              CARRY - "Success" flag (Set if buffer is not full).
                        ;*
                        ;* CALLING SEQUENCE:    Put the address of the label RXCHRS in the vector
                        ;*                      location.
                        ;*
                        ;* COMMENTS:    If the RX buffer is full upon entry, this routine aborts the
                        ;*              program.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: RXIEO,TXDSBL.
                        ;-- ****************************************************************************

027364  010246          RXCHRS::        MOV     R2,-(SP)        ;SAVE CONTENTS OF GPR R2.
027366  017702  152612  2$:     MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE DUT RX FIFO.
027372  100054          BPL     60$             ;EXIT THE ROUTINE IF THE DATA.VALID BIT IS CLR.

027374  023727  002720  000100          CMP     RXBCNT,#RXBFUL  ;COMPARE BUFFER COUNT WITH BUFFER CAPACITY.
027402  103402          BLO     4$              ;SKIP ABORT IF BUFFER IS NOT FULL.
027404  004737  022364          JSR     PC,OOPS         ;ABORT, MUST BE A PROGRAM BUG.
027410  010277  153302  4$:     MOV     R2,@RXBIPT      ;PUT THE CHAR IN THE BUFFER.
027414  062737  000002  002716          ADD     #2,RXBIPT       ;UPDATE POINTER TO THE NEXT BUFFER SLOT.
027422  023727  002716  003122          CMP     RXBIPT,#RXBEND  ;CHECK IF POINTER SHOULD WRAP AROUND.
027430  103403          BLO     6$              ;SKIP WRAPAROUND IF POINTER IS NOT AT END.
027432  012737  002722  002716          MOV     #RXBSTA,RXBIPT  ;WRAP INPUT POINTER AROUND.

027440  005237  002720  6$:     INC     RXBCNT          ;COUNT THIS CHARACTER AS BEING IN THE BUFFER.
027444  023727  002720  000030          CMP     RXBCNT,#RXBDTX  ;CHECK FOR BUFFER AT DISABLE TX LEVEL.
027452  001016          BNE     8$              ;SKIP DISABLING TX IF BUFFER LEVEL NOT CORRECT.
027454  005737  002510          TST     TXDBLF          ;CHECK STATE OF TX DISABLE FLAG.
027460  100413          BMI     8$              ;BRANCH IF TRANSMISSION ALREADY DISABLED.
027462  010546          MOV     R5,-(SP)        ;SAVE THE VALUE OF GPR R5.
027464  012705  000377          MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LINES SHOULD BE AFFECTED.
027470  004737  025552          JSR     PC,TXDSBL       ;CLEAR THE TX ENABLES FOR ALL LINES.
027474  010537  002250          MOV     R5,TXENBM       ;SAVE PREVIOUS TX ENABLE STATES IN STORAGE.
```

INTERRUPT SERVICE ROUTINE - RXCHRS - SEQ 0184

```
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
```

CVDHCDO DHV11-M FUNC TST PART 3 MACRO V05.00 Thursday 25-Apr 85 16:37 Page 129-1

```
                                .SBTTL  TRAP SERVICE ROUTINE            - TP4BRT -
                                ;*************************************************************************
                                ;*      Bus Time-out Trap (004 trap) Service Routine -
                                ;*      This routine is used during the DMA ADDRESS TEST.
                                ;*      It determines if the 004 trap was caused by an "expected" error or
                                ;*      not by examining the return PC value on the stack.  If the trap is
                                ;*      unexpected, this routine jumps to the normal Diagnostic Supervisor
                                ;*      004 trap handling routine.
                                ;*
                                ;* INPUTS:      SP - Points to the PC where the trap occured.
                                ;*              TRPAD2 - Label at the address where "expected" traps occur.
                                ;*              TP4FLG - 004 trap flags.
                                ;*
                                ;* OUTPUTS:     TP4FLG - Bit 15 is set if "expected" trap occured.
                                ;*
                                ;* CALLING SEQUENCE:    Put address pointed to by TP4BRT in 004 vector.
                                ;*                      Occurence of 004 trap vectors to this routine.
                                ;*
                                ;* COMMENTS:    Any 004 trap which occurs at an address other than that labeled
                                ;*              TRPAD2 will be handled by the normal 004 trap service routine.
                                ;*               This routine is used in conjunction with CKTRPB subroutine.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;*************************************************************  *************

027530  021627  017216          TP4BRT:: CMP    (SP),@TRPAD2    ;COMPARE EXPECTED ADDR WITH TRAP RET PC.
027534  001402                         BEQ    2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
027536  000177  152512                 JMP    @TP4VEC         ;IF NOT, JUMP TO NORMAL 004 TRAP SERVICE RTN.
027542  052737  100000  002256  2$:     BIS    #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
027550  000002                         RTI                    ;ALL DONE, GO BACK TO THE TEST.
```

```
                                .SBTTL  GLOBAL TRAP SERVICE ROUTINE     - TP4RTN -
                                ;**************************************************************************
                                ;*      Bus Time-out Trap (004 trap) Service Routine -
                                ;*      This routine is used during the Device Register Address Access Test.
                                ;*      It determines if the 004 trap was caused by an "expected" error or
                                ;*      not by examining the return PC value on the stack.  If the trap is
                                ;*      unexpected, this routine jumps to the normal Diagnostic Supervisor
                                ;*      004 trap handling routine.
                                ;*
                                ;* INPUTS:      SP - Points to the PC where the trap occured.
                                ;*              ADRPTR - Label at the address where "expected" traps occur.
                                ;*              TP4FLG - 004 trap flags.
                                ;*
                                ;* OUTPUTS:     TP4FLG - Bit 15 is set if "expected" trap occured.
                                ;*
                                ;* CALLING SEQUENCE:    Put address pointed to by TP4RTN in 004 vector.
                                ;*                      Occurence of 004 trap vectors to this routine.
                                ;*
                                ;* COMMENTS:    Any 004 trap which occurs at an address other than that labeled
                                ;*              ADRPTR will be handled by the normal 004 trap service routine.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: None.
                                ;**************************************************************************

027552  021627  017166          TP4RTN:: CMP     (SP),@ADRPTR    ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
027556  001402                           BEQ     2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
027560  000177  152470                   JMP     @TP4VEC         ;IF NOT,JUMP TO NORMAL 004 TRAP SERVICE RTN.
027564  052737  100000  002256  2$:      BIS     #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
027572  000002                           RTI                     ;ALL DONE, GO BACK TO THE TEST.
```

```
                                .SBTTL  INTERRUPT SERVICE ROUTINE       - TXDMA -
                                ;** ************************************************************************
                                ;*              - DMA Transmit Interrupt Service Routine -
                                ;*      This routine executes in response to an interrupt caused by the DUT
                                ;*      TX.ACTION bit becoming active.  This routine initiates the TX of a
                                ;*      new DMA buffer of characters or sets the TX Done Flag for the correct
                                ;*      line if TX is complete on that line.
                                ;*
                                ;* INPUTS:      BITTBL - Label of table of words each with a bit set.
                                ;*              CNCNTB - Base of # of chars to TX/RX table.
                                ;*              CSRA - Contains the address of the DUT CSR.
                                ;*              DPENDB - Base of the data pattern end table (entry per line).
                                ;*              DPLENB - Base of the data pattern length table.
                                ;*              IESTAT - Preserved states of the DUT interrupt enable bits.
                                ;*              TXCNTB - Lable at base of the TX character counter table.
                                ;*              TXPTRB - Label at base of the TX data pattern pointers table.
                                ;*
                                ;* OUTPUTS:     TXCNTx - Counters incremented for lines on which chars sent.
                                ;*              TXDONF - TX done flags set for lines which have sent all chars.
                                ;*              TXINTF - TX int flags (bit set if DMA.HO found set on line).
                                ;*
                                ;* CALLING SEQUENCE:    Put the address of the label TXDMA in the vector
                                ;*                      location.
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: DODMA.
                                ;-- ************************************************************************

027574                          TXDMA:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
027574  004537  005326                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
027600  017701  152376                  MOV     @CSRA,R1        ;READ THE CONTENTS OF THE DUT CSR.
027604  010100                          MOV     R1,R0           ;SAVE INITIAL CONTENTS OF IND.ADR.REG FIELD.
027606  000402                         BR      4$              ;BRANCH TO SKIP DOUBL READING OF DUT CSR.
                                ;+
                                ; Read the contents of the DUT CSR.  This will clear the TX.ACTION CSR bit.
                                ; If TX.ACTION is not set, exit this routine.
                                ; Determine the line for which the TX.ACTION was set.
                                ; Calculate an offset for use in accessing tables (2 times the line number).
                                ; Get the bit map of this line.
                                ;-
027610  017701  152366          2$:     MOV     @CSRA,R1        ;READ THE CONTENTS OF THE DUT CSR.
027614  100033                  4$:    BPL     60$             ;EXIT ROUTINE IF TX.ACTION IS CLEAR.
027616  000301                          SWAB    R1              ;CALCULATE THE LINE NUMBER OF THE LINE WHICH IS
027620  042701  177760                  BIC     #177760,R1      ;  ASSOCIATED WITH THE TX.ACTION.
027624  010104                          MOV     R1,R4           ;CALCULATE AN OFFSET FOR USE IN ACCESSING
027626  006304                          ASL     R4              ;  LINE COUNTER AND POINTER IN TABLES.
027630  016405  002366                  MOV     BITTBL(R4),R5   ;GET THE BIT MAP OF THIS LINE.
                                ;+
                                ; Get the TX character counter for this line.
                                ; If all the characters have been sent for this line:
                                ;    Set the TX done flag for this line.
                                ;    Don't send a char to the line (no more TX.ACTIONS on this line).
                                ;    Loop to check the TX.ACTION for another line.
                                ;
027634  026464  003504  003444          CMP     TXCNTB(R4),CHCNTB(R4) ;COMPARE # CHARS SENT AND TX COUNT.
027642  103403                         BLO     6$              ;GO TO SEND A CHAR IF NOT ALL CHARS SENT.
```

INTERRUPT SERVICE ROUTINE - TXDMA -

```
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 PREG05 SUBRT.
7274 027724  000002                       RTI
```

```
                        .SBTTL  INTERRUPT SERVICE ROUTINE       - TXSCHR -
                        ;** ************************************************************************
                        ;*      - Single Character Mode Transmit Interrupt Service Routine -
                        ;*      This routine executes in response to an iterrupt caused by the DUT
                        ;*      TX.ACTION bit becoming active.  This routine sends the next TX char
                        ;*      or sets the TX Done Flag for the correct line if TX is complete on
                        ;*      that line.
                        ;*
                        ;* INPUTS:      ACTLNS - Bit map of active DUT lines.
                        ;*              BITTBL - Label of table of words each with a bit set.
                        ;*              CHCNTB - Base of # of chars to TX/RX table.
                        ;*              CSRA - Contains the address of the CSR.
                        ;*              DPENDB - Base of the data pattern end table (entry per line).
                        ;*              DPLENB - Base of the data pattern length table.
                        ;*              IACBIT - Bit mask of inactive TX/RX bits.
                        ;*              IBM - Inactive bits mask (reflecting bits per char).
                        ;*              TXCHRA - Contains the address of the DUT TXCHAR register.
                        ;*              TXCNTB - Label at base of TX character counters table.
                        ;*              TXPTRB - Label at the base address of the TX pointers table.
                        ;*
                        ;* OUTPUTS:     CSR - DUT CSR IND.ADR.REG field is destroyed.
                        ;*              TXCHAR - DUT TXCHARs have words written to them.
                        ;*              TXCNTx - Counters incremented for lines on which chars sent.
                        ;*              TXDONF - TX done flags set for lines which have sent all chars.
                        ;*              TXPTRB - Each pointer in table points to next TX char for line.
                        ;*
                        ;* CALLING SEQUENCE:    Put the address of the label TXSCHR in the vector
                        ;*                      location.
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: None.
                        ;-- ************************************************************************

 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.
```

INTERRUPT SERVICE ROUTINE - TXSCHR -

```
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$PPT::
   7395
   7396 030114                                 EXIT    RPT
        030114  000167                                                                  .WORD   J$JMP
        030116  000000                                                                  .WORD   L10016-2-.
   7397
   7398                                 .EVEN
   7399
   7400 030120                                 ENDRPT
        030120                                                                  L10016:
        030120  104425                                                                  TRAP    C$RPT
```

PROTECTION TABLE

```
                                .SBTTL  PROTECTION TABLE
                                ;*************************************************************************
*
                                ;
                                ;               FVTSKL4.P11
                                ;
                                ;*************************************************************************



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

030122                                  BGNPROT
030122                                                                  L$PROT::

030122  177777                          -1              ;OFFSET INTO P-TABLE FOR CSR ADDRESS
030124  177777                          -1              ;OFFSET INTO P TABLE FOR MASSBUS ADDRESS
030126  177777                          -1              ;OFFSET INTO P TABLE FOR DRIVE NUMBER

030130                                  ENDPROT
```

PROTECTION TABLE

```
                                ;*****************************************************************************************
                                ;
                                ;               VDHC.INI
                                ;
                                ;*****************************************************************************************



                                .SBTTL  INITIALIZE SECTION
                                ;++
                                ;*****************************************************************************
                                ;*      This section contains the code which is performed at the beginning of
                                ;*      each pass or after a continue command.
                                ;*      This code performs the following actions:
                                ;*
                                ;*      Moves the information held in the hardware P-table into the global
                                ;*      data area.
                                ;*
                                ;*****************************************************************************
                                ;--
030130                                  BGNINIT
030130                                                                  L$INIT::
                                ;SEE IF PROGRAM JUST STARTED, BR IF YES
030130                                  READEF  #EF.START
030130  012700  000040                                                  MOV     #EF.START,R0
030134  104447                                                          TRAP    C$REFG
030136                                  BCOMPLETE       NEWSTA
030136  103416                                                          BCS     NEWSTA
                                ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
030140                                  READEF  #EF.RESTART
030140  012700  000037                                                  MOV     #EF.RESTART,R0
030144  104447                                                          TRAP    C$REFG
030146                                  BCOMPLETE       NEWRES
030146  103555                                                          BCS     NEWRES
                                ;SEE IF THIS IS A NEW PASS, BR IF YES
030150                                  READEF  #EF.NEW
030150  012700  000035                                                  MOV     #EF.NEW,R0
030154  104447                                                          TRAP    C$REFG
030156                                  BCOMPLETE       NEWPAS
030156  103554                                                          BCS     NEWPAS
                                ;SEE IF PROGRAM WAS JUST CONTINUED
030160                                  READEF  #EF.CONTINUE
030160  012700  000036                                                  MOV     #EF.CONTINUE,R0
030164  104447                                                          TRAP    C$REFG
030166                                  BNCOMPLETE      GETPRM
030166  103160                                                          BCC     GETPRM
030170  000137  030750                  JMP     ENDIT
030174                          NEWSTA:
030174                                  BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
030174  104433                                                          TRAP    C$RESET
                                ;+
                                ; Set up for Line Time Clock interrupts.
                                ;-
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$:     JSR     PC,CALMSL
 7517                                ;+
 7518                                ; Check for Memory Management present on this machine.
 7519                                ; If MEM MGT is present, disable it.
 7520                                ;-
 7521 030404  013737  000004  002254  8$:      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 SR0 REGISTER.
 7525 030430  012700  002240                MOV     #WORD1,R0       ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
 7526 030434  013701  002316                MOV     MMSR0,R1        ;SELECT MEM MGT SR0 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 SR0 REG AND CHECK FOR TRAP.
```

INITIALIZE SECTION

```
7530 030454  013737  002254  000004          MOV     TP4VEC,#04      ;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$:    CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
7534 030476  000137  030510          NEWRES: JMP     NEWPAS          ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
7535
7536 030502                          NEWRES: BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
     030502  104433                                                                  TRAP    C$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$:    MOV     (R1)+,CSRA      ;STORE DHV11-M CSR ADDRESS IN DEV.REG.ADDRESS TABLE
7562 030564  012102                  MOV     (R1)+,R2        ;GET THE RX INTERRUPT VECTOR ADDRESS.
7563 030566  010237  002170          MOV     R2,RXVECA       ;STORE RX INT VECTOR ADDRESS.
7564 030572  062702  000004          ADD     #4,R2           ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
7565 030576  010237  002172          MOV     R2,TXVECA       ;STORE TX INT VECTOR ADDRESS.
7566 030602  012137  002174          MOV     (R1)+,ACTLNS    ;STORE DHV11-M ACTIVE LINE BIT MAP
7567 030606  012702  000377          MOV     #MAPLNS,R2      ;GET THE BIT MAP FOR ALL LINES.
7568 030612  005102                  COM     R2              ;GET A BIT MAP OF NON-EXISTANT LINES.
7569 030614  040237  002174          BIC     R2,ACTLNS       ;CLEAR NON-EXISTANT LINES FROM ACTLNS.
7570 030620  112137  002176          MOVB    (R1)+,LOPBCK    ;STORE DHV11-M LOOPBACK MODE
7571 030624  112137  002177          MOVB    (R1)+,BRLEVL    ;STORE DHV11-M INTERRUPT BUS REQUEST LEVEL
7572                                 ;+
7573                                 ;       CALCULATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
7574                                 ;       DEVICE REGISTER ADDRESS TABLE.
7575                                 ;-
7576 030630  013701  002202          MOV     CSRA,R1         ;COPY CSR ADDRESS
7577 030634  005201                  INC     R1              ;INCREMENT CSR ADDRESS
7578 030636  005201                  INC     R1              ; COPY BY 2.
7579 030640  012703  000007          MOV     #7,R3           ;SET UP REGISTER COUNT
7580 030644  012702  002204          MOV     #RBUFA,R2       ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
7581 030650  010122          12$:    MOV     R1,(R2)+        ;STORE REGISTER ADDRESS IN TABLE
```

INITIALIZE SECTION

```
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    #PRI05                  ;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.
```

INITIALIZE SECTION SEQ 0197

```
                        ;*****************************************************************************
                        ;
                        ;               VDHC.ATD
                        ;
                        ;*****************************************************************************


                        .SBTTL  AUTODROP SECTION


                        ;++
                        ; THIS CODE IS EXECUTED IMMEDIATELY AFTER THE INITIALIZE CODE IF
                        ; THE "ADR" FLAG WAS SET.  THE UNIT(S) UNDER TEST ARE CHECKED TO
                        ; SEE IF THEY WILL RESPOND.  THOSE THAT DON T ARE IMMEDIATELY
                        ; DROPPED FROM TESTING.
                        ;--

030772                          BGNAUTO
030772                                                          L$AUTO::
                        
                        
030772                          ENDAUTO
030772                                                          L10021:
030772  104461                                                          TRAP    C$AUTO
```

```
                                ;*****************************************************************************
                                ;
                                ;               VDHC.CUC
                                ;
                                ;*****************************************************************************


                                .SBTTL  CLEANUP CODING SECTION

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

030774                                  BGNCLN
030774                                                                          L$CLEAN::

030774  005737  002222                    TST     CTRLCF          ;DID WE GET HERE BY CTRL ^ FROM TEST?
031000  001401                          BEQ     2$              ;CTRL-C FROM TEST?  NO, SKIP BUS RESET.
031002                                  BRESET                  ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
031002  104433                                                                  TRAP    C$RESET
031004                          2$:
031004  005737  002300                  TST CLKHRZ              ;IS CLOCK ENABLED
031010  001402                          BEQ 3$                  ;IF NO BRANCH
031012  005077  151254                  CLR @CLKCSR             ;TURN OFF CLOCK
031016                          3$:
031016                                  EXIT    CLN
031016  104432                                                                  TRAP    C$EXIT
031020  000002                                                                  .WORD   L10022-.


                                .EVEN

                                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
```

DROP UNIT SECTION

```
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
```

```
                                .SBTTL  HARDWARE TEST           - ADRA -
                                ;++
                                ;*****************************************************************
                                ;*                      - REGISTER ADDRESS TEST -
                                ;*
                                ;*      This test verifies that the Q-bus can read and write to the DHV11-M
                                ;*      device registers.  If the DHV11-M does not respond to the access
                                ;*      attempts (If the DHV11-M is at the wrong address, for example) the
                                ;*      004 bus time-out trap is detected by this routine and an error
                                ;*      is reported.
                                ;*
                                ;**********************  *************************************************
                                ;--

                                        BGNTST
                                                                                        T1::
                                        TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031142  012737  000001  002224          MOV     @TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (1)
031150  012737  177777  002222          MOV     @-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.

                                ;+
                                ; Set up to catch any 004 traps which occur:
                                ;-
031156  013737  000004  002254          MOV     @04,TP4VEC      ;SAVE THE EXISTING 004 TRAP VECTOR.
031164  012737  027552  000004          MOV     @TP4RTN,@04     ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
031172  005005                          CLR     R5              ;CLEAR THE ERROR FLAGS.

                                ;+
                                ; Set up for the initial iteration of the test loop:
                                ;-
031174  005004                          CLR     R4              ;CLEAR THE LINE COUNTER.

                                ;+
                                ; Here begins the loop to test the registers for a line.
                                ;    First test the CSR and set the IND.ADR.REG (I.A.R) field.
                                ;-
031176  005037  002256          2$:     CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
031202  013700  002202                  MOV     CSRA,R0         ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
031206  012701  031422                  MOV     @52$,R1         ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
031212  004737  017154                  JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
031216  103402                          BCS     4$              ;IF NO TRAP, BYPASS ERROR.
031220  052705  100001                  BIS     @100001,R5      ;SET FATAL READ ERROR FLAGS.
031224  042737  000017  031422  4$:     BIC     @17,52$         ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
031232  050437  031422                  BIS     R4,52$          ;OR IN THE LINE COUNTER TO THE I.A.R FIELD.
031236  010100                          MOV     R1,R0           ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
031240  013701  002202                  MOV     CSRA,R1         ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
031244  004737  017154                  JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
031250  103403                          BCS     6$              ;IF NO TRAP, BYPASS ERROR.
031252  052705  100002                  BIS     @100002,R5      ;SET FATAL WRITE ERROR FLAGS.
031256  000440                          BR      40$             ;EXIT AND REPORT FATAL ERROR.

                                ;+
                                ;    Now, we test each register for this line.
                                ;-
031260  012702  000010          6$:     MOV     @10,R2          ;INIT REGISTER COUNTER TO 8.
031264  013737  002202  031420          MOV     CSRA,50$                ;INITIALIZE THE REGISTER POINTER.
031272  012700  031420          8$:     MOV     @50$,R0         ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
031276  012701  031422                  MOV     @52$,R1         ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
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 031314  010100                 10$:    MOV     R1,R0           ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
 7837 031316  012701  031420                 MOV     #50$,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                                ; Now we set up to test the next line, or to exit if we are done.
 7848                                ;-
 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$ERDF
      031374  000145                                                                  .WORD   101
      031376  010072                                                                  .WORD   EM0103
      031400  013562                                                                  .WORD   ER0101
 7863
 7864 031402                                DODU    UNITN           ;DROP THIS UNIT FROM FUTHER TESTING.
      031402  ^13700  002200                                                          MOV     UNITN,R0
      031406  104451                                                                  TRAP    C$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
```

```
                                .SBTTL  HARDWARE TEST           - FRMERR -
                                ;+*****************************************************************************
                                ;*                      - FRAMING ERROR GENERATION TEST -
                                ;*
                                ;*      This test is used to verify the framing error detection capabilities
                                ;*      of the DHV11-M.
                                ;*      When in staggared loopback mode, characters are transmitted from
                                ;*      one group of lines at 8 bits/char,and received by the other group
                                ;*      at 5 bits/char.This will generate a framing error for each character.
                                ;*      This test will only execute if the staggared loopback mode is selected.
                                ;*      The special staggared loopback BERG connector must be fitted.
                                ;*      The active lines bit mask is used to indicate which lines have been
                                ;*      removed from further testing.
                                ;*
                                ;-*****************************************************************************
031432                                  BGNTST
031432                                                                          T2::
                                ;+
                                ; Execute this test in  staggared loopback mode only.
                                ;-
031432  123727  002176  000002          CMPB    LOPBCK,#2       ;CHECK MODE SELECTED.
031440  001154                          BNE     60$             ;EXIT IF STAGGERD LOOPBACK MODE NOT SELECTED.
031442  012737  177777  002222          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
        000002                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031450  012737  000002  002224          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (62)
031456  012737  000001  005316          MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
031464  012737  014071  005320          MOV     #6201.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
031472  012737  011142  005322          MOV     #EM6201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
031500  005037  002502                  CLR     ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
                                ;+
                                ; Reset the DUT to a known state, remove status codes from the fifo.
                                ;  Clear TX and  RX interrupt enable bits.
                                ; This subroutine reports error >>>>> 6201 <<<<<.
                                ;-
031504  004737  017234                  JSR     PC,CLNRST       ;RESET THE DUT.
031510  103130                          BCC     60$             ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; Disable all interrupts.
                                ; Set up  DMA TX and RX interrupt service routines.
                                ;-
031512  106427  000240                  MTPS    #PRI05          ;DISABLE DEVICE INTERRUPTS.
031516                                  SETVEC  TXVECA,#TXDMA,#PRI05    ;SELECT DMA TX INT SERVICE RTN.
031516  012746  000240                                                          MOV     #PRI05,-(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
031544  106427  000000                  MTPS    #PRI00          ;ALLOW INTERRUPTS.
                                ;+
                                ; Clear TX, RX, and DMA_Start error flags.
                                ;-
031550  005037  002504                  CLR     TXDONF          ;CLEAR TX DONE FLAGS FOR ALL LINES.
031554  005037  002506                  CLR     RXDONF          ;CLEAR RX DONE FLAGS FOR ALL LINES.
031560  005037  002252                  CLR     TXINTF          ;CLEAR TX ERROR FLAGS FOR ALL LINES.
                                ;+
                                ; Set up Error table and data pattern table.
```

```
                                        ;  The numerical value of the character indicates the number of the line
                                        ;  that transmitted it.
                                        ;-
031564  012700  003304                          MOV     #ERCNTB,R0      ;PASS THE ADDRESS OF THE TABLE TO BE CLEARED.
031570  004737  017256                          JSR     PC,CLR16W       ;CLEAR THE RX ERROR COUNTERS TABLE.
031574  005037  003604                          CLR     BUFBAS          ;SET SINGLE CHAR DATA TO BE A NULL.
                                        ;+
                                        ; Initialise DMA parameters in the control block.
                                        ;  Transmission on line group 1 at 8 bits/char,1 stop bits,odd parity.
                                        ;  Reception on line group 2 at 5 bits/char,1 stop,odd parity.
                                        ;-
031600  012700  156470                          MOV     #156470,R0      ;PASS LPR PARAMETER FOR 8 BITS/CHAR.
031604  012701  156440                          MOV     #156440,R1      ;PASS LPR PARAMETER FOR 5 BITS/CHAR.
031610  004737  021014                          JSR     PC,GETTIM       ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
031614  012702  003604                          MOV     #BUFBAS,R2      ;PASS START ADDRESS OF DATA PATTERN.
031620  012703  000001                          MOV     #1,R3           ;PASS LENGTH OF DATA PATTERN.
031624  013704  002230                          MOV     LGRP1M,R4       ;PASS LINE GROUP OF LINES THAT ARE TO TX.
031630  004737  020036                          JSR     PC,FRPSUP       ;SET UP DUT FOR TRANSMISSION AND RECEPTION.

                                        ;+
                                        ; Purge the FIFO of any un-wanted characters. This routine reports errors
                                        ;  with with error numbers from >>>>> 6202 thru 6204 <<<<<.
                                        ; Perform transmission and reception at 9600 baud.
                                        ;  Report any errors found, ie. Framing error bit clear or Parity error set.
                                        ;-
031634  005237  005320                          INC     ERRNBR          ;SET THE ERROR REPORT NUMBER TO 6202.
031640  004737  023234                          JSR     PC,PUFIFR       ;CLEAN OUT THE FIFO.
031644  103052                                  BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
031646  012737  014075  005320                  MOV     #6205.,ERRNBR   ;SET THE ERROR NUMBER TO 6205.
031654  004737  025742                          JSR     PC,TXFRPR       ;TX DATA PATTERN ON SELECTED ACTIVE LINES.
031660  012705  100000                          MOV     #100000,R5      ;PASS FRAMING ERROR TEST FLAG.
                                        ;+
                                        ; This subroutine reports error number >>>>> 6205 <<<<<.
                                        ;-
031664  004737  016676                          JSR     PC,CKFRPR       ;READ CHARACTERS, REPORT ANY ERRORS FOUND.
                                        ;+
                                        ; Reverse transmission/reception roles on all active lines, and repeat test.
                                        ;-
031670  005104                                  COM     R4              ;REVERSE ROLES FOR TRANSMISSION AND RECEPTION.
031672  004737  020036                          JSR     PC,FRPSUP       ;SET UP DUT FOR TRANSMISSION AND RECEPTION.
031676  005237  005320                          INC     ERRNBR          ;SET ERROR NUMBER TO 6206.
                                        ;+
                                        ; This routne reports errors with numbers >>>>> 6206 thru 6208 <<<<<.
                                        ;-
031702  004737  023234                          JSR     PC,PUFIFR       ;CLEAN OUT THE FIFO.
031706  103031                                  BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
031710  012737  014101  005320                  MOV     #6209.,ERRNBR   ;SET ERROR NUMBER TO 6209.
031716  004737  025742                          JSR     PC,TXFRPR       ;TX DATA PATTERN ON SELECTED ACTIVE LINES.
031722  012705  100000                          MOV     #100000,R5      ;PASS FRAMING ERROR TEST FLAG.
                                        ;+
                                        ; This subroutine reports errors >>>>> 6209 <<<<<.
                                        ;-
031726  004737  016676                          JSR     PC,CKFRPR       ;READ CHARACTERS, REPORT ANY ERRORS FOUND.
031732  005237  005320                          INC     ERRNBR          ;SET ERROR NUMBER TO 6210.
                                        ;+
                                        ;    Disable interrupts.
                                        ;    Clear the interrupt vectors.
```

HARDWARE TEST - FRMERR -

```
 7983                                   ;    Update the active lines bit map to reflect  ines 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,R0
      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 BENG 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 snd  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    #PRI05          ;DISABLE DEVICE INTERRUPTS.
 8042 032070                                    SETVEC  TXVECA,#TXDMA,#PRI05    ;SELECT DMA TX INT SERVICE RTN.
      032070  012746  000240                                                                    MOV     #PRI05,-(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    #PRI00          ;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.
```

```
                                        ;+
                                        ; This subroutine reports errors with numbers >>>>> 6312 thru 6316 <<<<<.
                                        ;-
032310  004737  016676                          JSR     PC,CKFRPR       ;READ CHARACTERS, REPORT ANY ERRORS FOUND.
032314  012737  014255  005320                   MOV     #6317.,ERRNBR   ;SET ERROR NUMBER TO 6317.
                                        ;+
                                        ;    Disable interrupts.
                                        ;    Clear the interrupt vectors.
                                        ;    Update the active lines bit map to reflect lines removed from testing.
                                        ;-
032322  004737  026036                          JSR     PC,TXIE0        ;DISABLE ALL TX INTERRUPTS.
                                        ;+
                                        ;     This subroutine reports errors >>>>> 6317 thru 6320 <<<<<.
                                        ;-
032326  004737  026450                          JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM TX/RX.

032332  106427  000240                          MTPS    #PRI05          ;DISABLE DEVICE INTERRUPTS.
032336                                          CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
032336  013700  002172                                                          MOV     TXVECA,R0
032342  104436                                                                  TRAP    C$CVEC

                                        ;+
                                        ; This subroutine reports errors with numbers >>>>> 6321 <<<<<.
                                        ;-
032344  012737  014261  005320                   MOV     #6321.,ERRNBR   ;SET ERROR NUMBER TO 6321.
032352  004737  024220                          JSR     PC,REPSMR       ;REPORT ERROR SUMMARIES IF CALLED FOR.

032356  106427  000240          60$:    MTPS    #PRI05          ;DISABLE DEVICE INTERRUPTS.
032362  005037  002222                   CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
032366                                   ENDTST
032366                                                                  L10027:
032366  104401                                                                  TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST                   - DMAADR -
                                ;** ************************************************************************
                                ;*                              - DMA ADDRESSING TEST -
                                ;*      THIS TEST VERIFIES , AS FAR AS POSSIBLE , THAT THE DUT CAN PERFORM A
                                ;*      DMA FROM A FULL 22 BIT OR 18 BIT ADDRESS. THE TEST RELIES ON FINDING A
                                ;*      COMPLEMENTARY PAIR OF ADDRESSES BETWEEN THE TOP OF PHYSICAL MEMORY AND
                                ;*      THE START OF THE TOP OF THE DIAGNOSTIC PROGRAM .
                                ;*      THIS MAY INVOLVE REMOVING PART OF THE DIAGNOSTIC RUNTIME SERVICES AND
                                ;*      THEN RESTORING. THE NUMBER OF BITS THAT HAVE BEEN SUCCESSFULLY TESTED
                                ;*      WILL BE PRINTED AT THE CONSOLE AT THE END OF THE TEST.
                                ;*
                                ;*
                                ;-- ************************************************************************
032370                                  BGNTST
032370                                                                  T4::

                                 TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER
000004
032370  012737  000004  002224    MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER
032376  012737  177777  002222    MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST
032404  012737  000001  005316    MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE
032412  012737  010461  005320    MOV     #4401.,ERRNBR   ;SET ERROR NUMBER TO 4401
032420  012737  010217  005322    MOV     #EM4401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN TABLE
032426  012737  014100  005324    MOV     #ER0503,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE

                                ;+
                                ; CLEAR THE SUCCESS FLAG TO INDICATE TEST FAILURE IN CASE IT DOES
                                ;-
032434  005037  034162                  CLR     SUCSS           ;INDICATE FAILURE , IN CASE THE DUT FAILS

                                ;+
                                ; SET UP THE 004 TRAP VECTOR TO CATCH ANY EXPECTED TRAPS THAT OCCUR
                                ;-
032440  013737  000004  002254          MOV     @#4,TP4VEC      ;SAVE EXISTING 004 TRAP VECTOR
032446  012737  027552  000004          MOV     #TP4RTN,@#4     ;SET 004 TRAP VECTOR TO OUR SERVICE ROUTINE

                                ;+
                                ; RESET THE DUT TO A KNOWN STATE,REMOVE THE STATUS CODES FROM THE FIFO.
                                ; CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR
                                ;-
032454  004737  017234                  JSR     PC,CLNRST       ;RESET THE DHV , REPORT ANY ERRORS
032460  103402                          BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
032462  000137  034124                  JMP     60$             ;EXIT THE TEST, FATAL ERROR WAS FOUND.

                                ;+
                                ; DETERMINE WHETHER MEMORY MANAGEMENT IS PRESENT
                                ;-
032466  005737  002322                  TST     MMPRES          ;IF MEM MGT IS PRESENT THEN
032472  001007                          BNE     1$              ;AVOID SETTING THE DMA TEST ADDR FOR
                                                                ;A 16 BIT MACHINE.
032474  012737  001252  002264          MOV     #1252,DMTSTA    ;SET UP THE FIRST DMA TEST ADDR FOR
                                                                ;A 16 BIT MACHINE
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 REC #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     #BIT0,BITLNG    ;INDICATE A 22 BIT MACHINE IN CASE IT IS
 8254
 8255 032616  106427  000340        6$:     MTPS    #PRI07          ;DISABLE CLOCK INTERRUPTS
 8256
 8257 032622  010400                        MOV     R4,R0           ;SET UP THE MOVE SOURCE
 8258 032624  012701  034156                MOV     #DEST,R1        ;SET UP THE DESTINATION
 8259
 8260                                ;+
 8261                                ; ENABLE THE MEM MGT AND ATTEMPT TO READ THE 22 BIT ADDRESS. IF A TRAP OCCURS
 8262                                ;  THEN THE HOST MUST BE A 22 BIT MACHINE SINCE WE HAVE ENSURED THAT
 8263                                ;  THE 18 BIT ADDRESS EXISTS.
 8264                                ;-
 8265
 8266 032630  012777  000001  147460        MOV     #BIT0,@MMSR0    ;ENABLE MEM MGT
 8267 032636  004737  017154                JSR     PC,CKTRAP       ;PERFORM THE MOVE AND CHECK FOR A TRAP
 8268 032642  005077  147450                CLR     @MMSR0          ;DISABLE MEM MGT
 8269
 8270 032646  106427  000240                MTPS    #PRI05          ;ENABLE CLOCK INTERRUPTS
 8271
 8272 032652  005737  002256                TST     TP4FLG          ;DID A TRAP OCCUR ?
 8273 032656  001022                        BNE     8$              ;YES , THEN JUMP AND INDICATE A 22 BIT MACHINE
 8274
 8275                                ;+
 8276                                ; SINCE A TRAP HASN'T OCCURED THEN EITHER THE MACHINE IS A 22 BIT MACHINE WITH
 8277                                ;  MEMORY AT THE ADDRESS JUST READ , OR , ITS AN 18 BIT MACHINE IN WHICH CASE
 8278                                ;  THE ADDRESS JUST READ WOULD BE ONE OF THE DATA WORDS AT ADDRESS #SDPBAS.
 8279                                ;
 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.,BITSTC     ;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    #PRI07          ;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
```

```
                                        ;PHYSICAL ADDRESSS AND IF WE'RE STILL INSIDE THE DIAGNOSTIC WE ABANDON THE
                                        ;TEST. ONCE WE ARE IN THIS REGION WE ARE ONLY ABLE TO TEST THE LOWEST 14 BITS.
                                        ;-

033072  022737  000252  002264          CMP     #252,DMTSTA     ;IF THE BIT HAS ALREADY BEEN SET THEN
033100  001014                          BNE     15$             ;ABANDON THE TEST,AFTER REPORTING THE ERROR ,
                                                                ; BECAUSE NO SUITABLE MEMORY HAS BEEN FOUND.
033102  012737  000652  002264          MOV     #652,DMTSTA     ;SET THE BIT
033110  062700  040000                  ADD     #40000,R0       ;ADD THE BIT INTO THE PHYSICAL ADDR
033114  020037  002262                  CMP     R0,FFREM        ;IF WE'RE NOW STILL INSIDE THE DIAGNOSTIC THEN
033120  103404                          BLO     15$             ;REPORT ERROR AND ABANDON THE TEST.
033122  012737  000016  034152          MOV     #14.,BITSTD     ;OTHERWISE SET THE BITS TESTED TO 14 BITS.
033130  000716                          BR      12$             ;CONTINUE WITH THE SEARCH.

033132  005237  005320          15$:    INC     ERRNBR          ;SET THE ERROR NUMBER TO 4402
033136  012701  010241                  MOV     #EM4402,R1      ;SELECT MESSAGE TO BE REPORTED.
                                                                ; " NO SUITABLE ADDR FOUND. DMA TEST ABORTED "

033142  000137  034122          16$:    JMP     34$             ;JUMP TO THE ERROR.

                                        ;+
                                        ; WRITE THE TEST DATA INTO THE TWO AREAS JUST FOUND. IF A TRAP OCCURS WHILE
                                        ; WE ARE WRITING DATA INTO THESE AREAS THEN THE HOST MACHINE IS AT FAULT.
                                        ;-

033146  012700  005132          17$:    MOV     #SDPBAS,R0      ;SET UP THE SOURCE ADDR FOR THE MOVE AS OUR
                                                                ;TEST DATA PATTERN.
033152  013737  002264  034154          MOV     DMTSTA,DUMY     ;SAVE THE LOWER DMTSTA
033160  013737  034160  002264          MOV     ODTSTA,DMTSTA   ;START WITH THE HIGHER OF THE TWO
                                                                ; COMPLEMENTARY ADDRESSES.
033166  012703  000020                  MOV     #16.,R3         ;SET THE NUMBER OF DATA BYTES TO BE WRITTEN
033172  012705  000001                  MOV     #1,R5           ;INDICATE TO WRITE TO DMTSTA

033176  012701  000340                  MOV     #340,R1         ;SET PRIORITY 7 TO DISABLE THE CLOCK
033202  004737  025352                  JSR     PC,STPSW        ;

033206  005237  005320                  INC     ERRNBR          ;SET THE ERROR NUMBER TO 4403
033212  012701  010307                  MOV     #EM4403,R1      ;SELECT THE MESSAGE,
                                                                ; "HOST FAILURE. WRITE FAILED TO AN ADDR WHICH
                                                                ;HAD BEEN SUCCESSFULLY READ, TEST ABANDONED "

033216  004737  017466                  JSR     PC,DMRW         ;PERFORM THE TRANSFER
033222  005737  002256                  TST     TP4FLG          ;EXIT IF HOST FAILURE
033226  001345                          BNE     16$             ;AND REPORT ERROR.
033230  013737  034154  002264          MOV     DUMY,DMTSTA     ;SELECT THE LOWER DMA TEST ADDR.
033236  012700  005156                  MOV     #SDP2B,R0       ;SELECT THE NEXT DATA PATTERN
033242  004737  017466                  JSR     PC,DMRW         ;PERFORM THE TRANSFER
033246  005737  002256                  TST     TP4FLG          ;EXIT IF HOST FAILURE
033252  001333                          BNE     16$             ;

                                        ;+
                                        ; 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  DDRESS 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,WAIBIT       ;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
```

```
                                                                        ;WRITTEN TO. "
034024  012737  014100  005324          MOV     @ER0503,ERRBLK        ;SELECT THE ERROR ROUTINE
034032  000433                        BR      34$                     ;REPORT THE ERROR

                                ;+
                                ; HAS THE TEST BEEN SUCCESSFUL, PRINT THE BITS TESTED IF IT HAS,
                                ; REPORT THE ERRORS OTHERWISE.
                                ;-

034034  005737  034162          32$:    TST     SUCSS                   ;IF THE ERROR IS NON TEST SPECIFIC THEN
034040  001430                        BEQ     34$                     ;BRANCH TO REPORT ERRORS
034042  013701  034152                MOV     BITSTD,R1               ;LOAD THE NUMBER OF BITS TESTED
034046  005301                        DEC     R1                      ;DEC TO GIVE THE BIT POSITION OF THE MSB TESTED.
034050  022737  000001  034162        CMP     @1,SUCSS                ;IF THE BITS TESTED ARE BAD THEN
034056  001021                        BNE     34$                     ;BRANCH AND REPORT ERRORS.

                                ;+
                                ; OTHERWISE DETERMINE IF PRINTING OF THE SUCCESSFULLY TESTED BITS WAS REQUESTED.
                                ;-

034060  032737  000040  002164          BIT     @BIT05,OPTION           ; PRINT THE BITS TESTED IF THE SOFTWARE
034066  001416                        BEQ     60$                     ;OPTION HAS REQUESTED IT
034070  010102                        MOV     R1,R2                   ;CALCULATE THE NUMBER OF BITS WHICH HAVE
034072  005202                        INC     R2                      ;  BEEN TESTED SUCCESSFULLY.
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$PNTB
034114  062706  000010                                                          ADD     @10,SP
034120  000401                        BR      60$                     ;EXIT THE TEST

034122                          34$:    ERROR                           ;  REPORT ERRORS
034122  104460                                                                  TRAP    C$ERROR

034124  106427  000240          60$:    MTPS    @PRI05                  ;ENABLE THE CLOCK
034130  013737  002254  000004          MOV     TP4VEC,@#4              ;RESTORE THE NORMAL 004 TRAP VECTOR
034136  005037  002222                CLR     CTRLCF                  ;INDICATE THAT WE ARE NOT WITHIN A TEST

034142                                EXIT    TST
034142  104432                                                          TRAP    C$EXIT
034144  000020                                                          .WORD   L10030-.

                                ;+
                                ; ********************** LOCAL VARIBLE AREA *********************************
                                ;-

034146  000000                 70$:    .WORD   0                       ;COUNTER FOR THE NUMBER OF DMA'S COMPLETED
034150  000000                 80$:    .WORD   0                       ;SAVE AREA FOR THE ACTIVE LINE NUMBER
034152  000000                 BITSTD: .WORD   0                       ;NUMBER OF BITS TESTED
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
```

```
                                .SBTTL  HARDWARE TEST           - MODLPB -
                                ;++ ******************************************************************************
                                ;*                      - Modem Loopback Test -
                                ;*      This test is used to move data through a modem which is connected to
                                ;*      one of the device serial ports.  This test is run only if Modem
                                ;*      Loopback is specified.  This test utilizes the following operator
                                ;*      dialogue:
                                ;*              MODEM BAUDRATE IN BPS:  (D) 1200 ?
                                ;*              TYPE <CR> WHEN MODEM LINK ESTABLISHED:  (L)  Y ?
                                ;*              MODEM STATUS SIGNAL REPORT:
                                ;*                  LINE #n:  DSR=n, RI=n, DCD=n, CTS=n
                                ;*                  ...    repeated for each active line
                                ;*              NUMBER OF 256 BYTE DATA PATTERNS TO SEND ON EACH SELECTED LINE
                                ;*                      (1-255, 0=SEND UNTIL ^C): (D)  1 ?
                                ;*              PRINT MODEM STATUS SIGNAL REPORT AFTER EACH PATTERN: (L)  Y ?
                                ;*
                                ;*      At the completion of sending the specified number of data patterns the
                                ;*      test issues the following prompt:
                                ;*              EXIT THE TEST (N = LOOP BACK TO SEND MORE DATA): (L) Y ?
                                ;*
                                ;*      If extended error reporting is allowed, a report is printed at the end
                                ;*      of each data pattern with the following format:
                                ;*          MODEM LOOPBACK TEST STATUS REPORT:  PATTERN #nnn (D) COMPLETED.
                                ;*
                                ;*      This test is performed using 8 bits per character, 1 stop bit, and no
                                ;*      parity.  This test does not support split speed.  All selected lines
                                ;*      are tested at the selected baudrate.  An error summary is reported at
                                ;*      the end of the test if any lines have exceeded the number of individual
                                ;*      data errors to report as selected in the Software P-Table dialogue.
                                ;*
                                ;-- ******************************************************************************
                                        BGNTST
 034166                                                                                 T5::
 034166                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 034166 012737  000005  002224   MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (89)
                                ;+
                                ; Verify that the test should be performed.  Must have the following:
                                ;       Modem loopback selected.
                                ;       Manual intervention allowed.
                                ;-
 034174 123727  002176  000004   CMPB   LOPBCK,#4       ;TEST THE LOOPBACK TYPE INDICATOR.
 034202 001402                  BEQ    2$              ;MODEM LOOPBACK SELECTED? YES, CONTINUE TEST.
 034204                         EXIT   TST             ;NO, ABORT THE TEST.
 034204 104432                                                                          TRAP    C$EXIT
 034206 000650                                                                          .WORD   L10031-.
 034210                 2$:     MANUAL                 ;CHECK FOR MANUAL INTERVENTION ALLOWED.
 034210 104450                                                                          TRAP    C$MANI
 034212                         BCOMPLETE      4$      ;MANUAL INTERVENTION ALLOWED? YES, DO TEST.
 034212 103402                                                                          BCS     4$
 034214                         EXIT   TST             ;NO, ABORT THE TEST.
 034214 104432                                                                          TRAP    C$EXIT
 034216 000640                                                                          .WORD   L10031-.

 034220 012737  177777  002222  4$:     MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 034226 012737  000001  005316          MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 034234 012737  021305  005320          MOV    #8901.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
 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  NOPMSG,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$HILIM
        034610                                                          10001$:
 8792   034610  013704  002266                  MOV     GMANWD,R4
 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.
 8837                                  ;-
 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  100373                          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    lave been told to exit.
 8854                                  ;    Clear    vice 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    @PRIO5          ;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    @PRIO5          ;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 - SEQ 0224

035056 104401 TRAP C$ETST

```
                                .SBTTL  HARDWARE TEST           - KBECHO -
                                ;++ ******************************************************************************
                                ;*                          - Keyboard Echo Test -
                                ;*      This is a test which puts UARTS for the active lines into remote
                                ;*      loopback mode.  The active line UARTS are set up with a baudrate
                                ;*      which is specified by the operator.  The test executes indefinitely
                                ;*      until terminated by the operator.
                                ;*
                                ;*      This test can be used for looping back terminal keyboard input onto
                                ;*      a terminal CRT or it can be used as a general loopback method for
                                ;*      testing communications links to the DUT from the other end of the
                                ;*      channel.  DTR and RTS are set on the selected lines during this
                                ;*      test to allow the testing of modem links.
                                ;*
                                ;-  ******************************************************************************
035060                                  BGNTST
035060                                                                          T6::
                        000006           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
035060  012737  000006  002224           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (94)
                                ;+
                                ; Verify that the test should be performed.  Must have the following:
                                ;       Keyboard Echo loopback selected.
                                ;       Manual intervention allowed.
                                ;-
035066  123727  002176  000005           CMPB   LOPBCK,#5       ;TEST THE LOOPBACK TYPE INDICATOR.
035074  001402                          BEQ    2$              ;KBD ECHO LPBCK SELECTED? YES, CONTINUE TEST.
035076                                  EXIT   TST             ;NO, ABORT THE TEST.
035076  104432                                                                  TRAP    C$EXIT
035100  000204                                                                  .WORD   L10032-.
035102                          2$:     MANUAL                 ;CHECK FOR MANUAL INTERVENTION ALLOWED.
035102  104450                                                                  TRAP    C$MANI
035104                                  BCOMPLETE      4$      ;MANUAL INTERVENTION ALLOWED? YES, DO TEST.
035104  103402                                                                  BCS     4$
035106                                  EXIT   TST             ;NO, ABORT THE TEST.
035106  104432                                                                  TRAP    C$EXIT
035110  000174                                                                  .WORD   L10032-.

035112  012737  177777  002222  4$:      MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
035120  012737  000001  005316           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
035126  012737  022271  005320           MOV    #9401.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
035134  012737  013055  005322           MOV    #EM9401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
035142  005037  002502                   CLR    ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
                                ;+
                                ; Reset the DUT to a known state, remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports error >>>>> 9401 <<<<<.
                                ;-
035146  004737  017234                  JSR    PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
035152  103402                          BCS    .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
035154  000137  035300                  JMP    60$             ;RESET FAILURE, ABORT THIS TEST.
                                ;+
                                ; Print the test name.
                                ;-
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,R0
        035176  104417                                                  TRAP    C$PNTF
        035200  062706  000006                                          ADD     #6,SP
   8920
   8921                                 ;+
   8922                                 ; Set up the DUT UARTS with the proper line parameters.
   8923                                 ;    Get the desired baudrate from the operator.
   8924                                 ;    Calculate proper DUT LPR contents.
   8925                                 ;    Set up the DUT LPR registers.
   8926                                 ;    Get the proper DUT LNCTRL register contents.
   8927                                 ;    Set up the DUT LNCTRL registers.
   8928                                 ;-
   8929 035204  004737  020302                  JSR     PC,GETBDR
   8930 035210  010100                           MOV     R1,R0
   8931 035212  006301                           ASL     R1
   8932 035214  006301                           ASL     R1
   8933 035216  006301                           ASL     R1              ;GET DUPLICATE COPIES OF BAUDRATE CODE
   8934 035220  006301                           ASL     R1              ;  IN THE UPPER BYTE OF THE NEW
   8935 035222  050100                           BIS     R1,R0           ;  LPR CONTENTS.
   8936 035224  000300                           SWAB    R0
   8937 035226  042700  000377                   BIC     #377,R0         ;SET UP 1 STOP BIT, NO PARITY, 8 BITS/CHAR
   8938 035232  052700  000030                   BIS     #30,R0          ;  IN THE LPR CONTENTS.
   8939
   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                                 ; We got a response from the operator, so terminate the test.
   8955                                 ;-
   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
```

```
                                .SBTTL  HARDWARE TEST           - SINGLC -
                                ;++ ******************************************************************************
                                ;*              - Single Character Mode Test -
                                ;*      This test verifies that the Device Under Test (DUT) will perform
                                ;*      transmission and reception correctly using the single character
                                ;*      mode interrupts.  The test is performed at 3 baudrates (slowest,
                                ;*      middle, and highest) at all combinations of # of stop bits, # of
                                ;*      bits per character, and types of parity using short data patterns.
                                ;*      A high speed test is also performed at the highest baudrate with all
                                ;*      combinations of line parameters using longer data patterns.
                                ;*      This test is performed in internal loopback regardless of the type
                                ;*      of loopback which is selected for the DUT in the hardware P-table.
                                ;*
                                ;-- ******************************************************************************

035306                                  BGNTST
035306                                                                          T7::
        000007                   TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
035306  012737  000007  002224          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (90)
035314  012737  177777  002222          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
035322  012737  000001  005316          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
035330  012737  021451  005320          MOV     #9001.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
035336  012737  011245  005322          MOV     #EM9001,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
035344  005037  002502                  CLR     ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
                                ;+
                                ; Reset the DUT to a known state, remove the status codes from the fifo.
                                ;  Clear TX and RX interrupt enable bits in the CSR.
                                ; This subroutine reports error >>>>> 9001 <<<<<.
                                ;-
035350  004737  017234                  JSR     PC,CLNRST       ;RESET THE DHV11-M, REPORT ANY ERRORS FOUND.
035354  103402                          BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
035356  000137  036010                  JMP     60$             ;EXIT THE TEST, FATAL ERROR WAS FOUND.
035362  012737  021452  005320          MOV     #9002.,ERRNBR   ;SET THE ERROR NUMBER.

                                ;+
                                ; Set up for Transmit and Receive interrupts.
                                ;-
035370  106427  000240                  MTPS    #PRI05          ;DISABLE DEVICE INTERRUPTS.
035374                                  SETVEC  TXVECA,#TXSCHR,#PRI05   ;SELECT SINGLE CHAR TX INT SERVICE RTN.
035374  012746  000240                                                          MOV     #PRI05,-(SP)
035400  012746  027726                                                          MOV     #TXSCHR,-(SP)
035404  013746  002172                                                          MOV     TXVECA,-(SP)
035410  012746  000003                                                          MOV     #3,-(SP)
035414  104437                                                                  TRAP    C$SVEC
035416  062706  000010                                                          ADD     #10,SP
035422                                  SETVEC  RXVECA,#RXCHRS,#PRI05   ;SELECT RX INT SERVICE RTN.
035422  012746  000240                                                          MOV     #PRI05,-(SP)
035426  012746  027364                                                          MOV     #RXCHRS,-(SP)
035432  013746  002170                                                          MOV     RXVECA,-(SP)
035436  012746  000003                                                          MOV     #3,-(SP)
035442  104437                                                                  TRAP    C$SVEC
035444  062706  000010                                                          ADD     #10,SP
035450  106427  000000                  MTPS    #PRI00          ;ALLOW INTERRUPTS.
                                ;+
                                ; Clear the error counter table.
                                ; This table will accumulate error count totals for each line during this test.
                                ;-
```

HARDWARE TEST - SINGLC -

```
 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     GPRS0B          ;CLEAR THE GPR SAVE AREA R1 STORAGE TO INDICATE
 9011                                                                    ;  THAT THIS IS THE FIRST TIME IN GETLP1.
 9012 035470  004737  025372            2$:     JSR     PC,SWAP0        ;SWAP GPRS WITH GPR SAVE AREA ZERO FOR GETLP1.
 9013 035474  004737  020512                    JSR     PC,GETLP1       ;GET NEXT SET OF LPR CONTENTS, OR CARRY CLEAR.
 9014 035500  004737  025372                    JSR     PC,SWAP0        ;SWAP BACK GPRS AND GPR SAVE AREA ZERO.
 9015 035504  103055                            BCC     4$              ;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, allo  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$:     JSR     PC,SWAP0        ;SWAP GPRS WITH GPR SAVE AREA ZERO FOR GETLP1.
 9062 035670  004737  020664                    JSR     PC,GETLP2       ;GET NEXT SET OF LPR CONTENTS, OR CARRY CLEAR.
 9063 035674  004737  025372                    JSR     PC,SWAP0        ;SWAP BACK GPRS AND GPR SAVE AREA ZERO.
 9064 035700  103036                           BCC     10$             ;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    #PRI05          ;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$:     MTPS    #PRI05          ;DISABLE DEVICE INTERRUPTS.
 9137 036116                                   SETVEC  TXVECA,#TXDMA,#PRI05    ;SELECT DMA TX INT SERVICE RTN.
      036116   012746   000240                                                          MOV     #PRI05,-(SP)
      036122   012746   027574                                                          MOV     #TXDMA,-(SP)
      036126   013746   002172                                                          MOV     TXVECA,-(SP)
      036132   012746   000003                                                          MOV     #3,-(SP)
      036136   104437                                                                  TRAP    C$SVEC
      036140   062706   000010                                                          ADD     #10,SP
 9138 036144                                   SETVEC  RXVECA,#RXCHRS,#PRI05   ;SELECT RX INT SERVICE RTN.
      036144   012746   000240                                                          MOV     #PRI05,-(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    #PRI00          ;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.
```

HARDWARE TEST - DMA -

```
 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,TXRRE%       ;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    #PRI05          ;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$:  MTPS    #PRI05          ;DISABLE DEVICE INTERRUPTS.
 9285 036640  005037  002222                 CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 9286 036644                                ENDTST
      036644                                                        L10034:
      036644  104401                                                           TRAP    C$ETST
```

```
 9288                                   .SBTTL  HARDWARE TEST           - SPLSPD -
 9289                                   ;+*****************************************************************************
 9290                                   ;*                        - SPLIT SPEED TEST -
 9291                                   ;*      This test 's 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$SVEC
      036760  062706  000010                                                              ADD     #10,SP
 9322 036764                                    SETVEC  RXVECA,#RXCHRS,#PRI05   ;SELECT RX INT SERVICE RTN.
      036764  012746  0002?0                                                              MOV     #PRI05,-(SP)
      036770  012746  027364                                                              MOV     #RXCHRS,-(SP)
      036774  013746  002170                                                              MOV     RXVECA,-(SP)
      037000  012746  000003                                                              MOV     #3,-(SP)
      037004  104437                                                                      TRAP    C$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 perfom'ng 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$:     MOV     (R5)+,R0        ;GET LPR CONTENTS FOR LINGRP II.
 9348 037044  012501                            MOV     (R5)+,R1        ;GET LPR CONTENTS FOR LINGRP I.
 9349 037046  004737  021014                   JSR     PC,GETTIM       ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
 9350 037052  012702  005156                    MOV     #SDP28,R2       ;SET UP THE START ADR OF THE DATA PATTERN.
 9351 037056  012503                            MOV     (R5)+,R3        ;GET NUMBER OF REPEAT TRANSMISSION ON LINGRPII.
 9352 037060  012504                            MOV     (R5)+,R4        ;GET NUMBER OF REPEAT TRANSMISSION ON LINGRPI.
 9353 037062  004737  025052                   JSR     PC,SPLSUP       ;SET UP CONTROL BLOCK ETC, FOR TX/RX.
 9354 037066  012737  021762  005320            MOV     #9202.,ERRNBR   ;SET THE ERROR NUMBER TO 9202.
 9355                                    ;+
 9356                                    ; This routine reports errors with numbers >>>>> 9202 thru 9204 <<<<<.
 9357                                    ;-
 9358 037074  004737  023234                    JSR     PC,PUFIFR       ;PURGE THE DUT RECEIVE CHARACTER FIFO.
 9359 037100  103071                           BCC     60$             ;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    #PRI05          ;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
```

```
                                .SBTTL  HARDWARE TEST           - REPBMP -
                                ;** ************************************************************************
                                ;*                          - Report any BMP codes in the queue -
                                ;*      This is a pseudo-test used to report any BMP codes that were found
                                ;*      in the DUT's FIFO during previous test, and logged in the BMP code
                                ;*      queue.
                                ;*      It is unlikely that running this pseudo-test alone will produce any
                                ;*      error reports.
                                ;*
                                ;-- ************************************************************************
                                        BGNTST
                                                                                        T10::
                                         TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
037272  012737  000012  002224          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (93)
037300  012737  177777  002222          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
037306  013702  002512                  MOV     BMPCQP,R2       ;GET THE CONTENTS OF THE POINTER.
037312  012703  002514                  MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
037316  020203                          CMP     R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
037320  001411                        BEQ     60$             ;EXIT NO CODES IN THE QUEUE.
                                ;+
                                ; There is at least one BMP code in the queue. Report the error.
                                ;-
                                        ;Report error BMP CODE FOUND IN TEST nn, BMP CODE:nnnnnn"

037322  012701  013001                MOV     #EM9304,R1      ;PASS THE FIRST MESSAGE TO BE REORTED.
037326                                 ERRDF   9301,EM9301,ER9301 ;           >>>>> ERROR #9301 <<<<<.
037326  104455                                                                          TRAP    C$ERDF
037330  022125                                                                        .WORD   9301
037332  012664                                                                        .WORD   EM9301
037334  015542                                                                        .WORD   ER9301

037336  012737  002514  002512          MOV     #BMPCQB,BMPCQP  ;SET POINTER BACK TO THE BEGINING OF THE QUE.

037344  005037  002222          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
037350                                 ENDTST
037350                                                                        L10036:
037350  104401                                                                      TRAP    C$ETST
```

```
 9451                           ;*****************************************************************************
 9452                           ;
 9453                           ;               VDHC.HWQ
 9454                           ;
 9455                           ;*****************************************************************************
 9457
 9458
 9459                           .SBTTL  HARDWARE PARAMETER CODING SECTION
 9460
 9461
 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 L1003/ !.$HARD/2
      037354                                                             L$HARD::
 9473
 9483
 9484 037354                   ;DEVICE CSR ADDRESS QUESTION:
                                       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$HILIM
 9485
 9486 037364                   ;DEVICE INTERRUPT VECTOR QUESTION:
                                       GPRMA   HWPTQ2,2,0,40,776,YES
      037364  001031                                                      .WORD   T$CODE
      037366  037450                                                      .WORD   HWPTQ2
      037370  000040                                                      .WORD   T$LOLIM
      037372  000776                                                      .WORD   T$HILIM
 9487
 9488 037374                   ;ACTIVE LINES BIT MAP QUESTION:
                                       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$HILIM
 9489
 9490 037406                   ;TYPE OF LOOPBACK QUESTION:
                                       GPRMD   HWPTQ4,6,0,377,1,5,YES
      037406  003032                                                      .WORD   T$CODE
      037410  037531                                                      .WORD   HWPTQ4
      037412  000377                                                      .WORD   377
      037414  000001                                                      .WORD   T$LOLIM
      037416  000005                                                      .WORD   T$HILIM
 9491
 9492 037420                   ;INTERRUPT BR LEVEL QUESTION.
                                       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
```

HARDWARE PARAMETER CODING SECTION SEQ 0239

```
 9493
 9494
 9495 037432                                ENDHRD
                                                                          .EVEN
      037432                                                          L10037:
 9496
 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
   9545 037720                  ;UNIT NUMBER PRINTOUT QUESTION:
                                        GPRML   SWPTQ1,0,20,YES
        037720  000130                                                          .WORD   T$CODE
        037722  037746                                                          .WORD   SWPTQ1
        037724  000020                                                          .WORD   20
   9546
   9547 037726                  ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
                                        GPRMD   SWPTQ2,2,D,177777,0,177777,YES
        037726  001052                                                          .WORD   T$CODE
        037730  040022                                                          .WORD   SWPTQ2
        037732  177777                                                          .WORD   177777
        037734  000000                                                          .WORD   T$LOLIM
        037736  177777                                                          .WORD   T$HILIM
   9548
   9549 037740                  ;REPORT NUMB OF BITS TESTED IN DMA ADDR TEST QUESTION:
                                        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
```

SOFTWARE PARAMETER CODING SECTION SEQ 0241

```
                        ;*****************************************************************************
                        ;
                        ;               FVTSKL6.P11
                        ;
                        ;*****************************************************************************

                        $PATCH::
                                .BLKW   24

                                LASTAD

                                                                                .EVEN
 040242  000000                                                                .WORD   0
 040244  000000                                                                .WORD   0
 040246                 L$LAST::
                                ENDMOD
                                .END
```

SEQ 0242

Symbol table

```
ACTLNS  002174 G        CBOFSA  003142 G        C$INLP= 000020          EF7801  006177 G        EM9025  012107 G
ADDR    025366          CHCNTB  003444 G        C$MANI= 000050          EF8901  006235 G        EM9026  012203 G
ADR   = 000020 G        CHKEXT  016220 G        C$MAP = 000102          EF9001  006266 G        EM9027  012227 G
ADRPTR  017166 G        CHKLOS  016320 G        C$MEM = 000031          EF9002  006350 G        EM9028  012307 G
ALTFLD  015722 G        CHRMSK  016422 G        C$MMU = 000103          EF9003  006422 G        EM9030  012366 G
ASSEMB= 000010          CHRTOT  002500 G        C$MSG = 000023          EF9004  006451 G        EM9101  012443 G
BCOUNT  002310 G        CKCHR   016460 G        C$OPNR= 000034          EF9005  006501 G        EM9102  012477 G
BDRMSG  013127 G        CKFRPR  016676 G        C$OPNW= 000104          EF9006  006532 G        EM9104  012566 G
 ITLNG  002260 G        CKINAC  017046 G        C$PNTB= 000014          EF9007  006551 G        EM9201  012642 G
BITSTD  034152          CKTRAP  017154 G        C$PNTF= 000017          EF9008  006645 G        EM9301  012664 G
BITTBL  002366 G        CKTRPB  017204 G        C$PNTS= 000016          EF9009  006704 G        EM9302  012704 G
BIT0  = 000001 G        CLKBRL  002274 G        C$PNTX= 000015          EF9010  006743 G        EM9303  012734 G
BIT00 = 000001 G        CLKCSR  002272 G        C$PUTB= 000072          EF9012  007032 G        EM9304  013001 G
BIT01 = 000002 G        CLKHRZ  002300 G        C$PUTW= 000073          EF9013  007146 G        EM9401  013055 G
BIT02 = 000004 G        CLKINT  027314 G        C$QIO = 000377          EF9019  007213 G        ENDETB  004604 G
BIT03 = 000010 G        CLKVEC  002276 G        C$RDBU= 000007          EF9020  007232 G        ENDIT   030750
BIT04 = 000020 G        CLNRST  017234 G        C$REFG= 000047          EF9101  007313 G        ERCNTB  003304 G
BIT05 = 000040 G        CLR16W  017256 G        C$REL = 000077          EF9103  007316 G        ERLTBL  003604 G
BIT06 = 000100 G        CONMAP  017300 G        C$RESE= 000033          EF9301  007364 G        ERRBLK  005324 G
BIT07 = 000200 G        CSRA    002202 G        C$REVI= 000004          EF9302  007432 G        ERRMSG  005322 G
BIT08 = 000400 G        CSRO  = 000000 G        C$RFLA= 000021          EMLMSG  013156 G        ERRNBR  005320 G
BIT09 = 001000 G        CTRLCF  002222 G        C$RPT = 000025          EM0101  022430 G        ERRTYP  005316 G
BIT1  = 000002 G        C$AU  = 000052          C$SEFG= 000046          EM0102  022514 G        ERSMRF  002502 G
BIT10 = 002000 G        C$AUTO= 000061          C$SPRI= 000041          EM0103  010072 G        ER0101  013562 G
BIT11 = 004000 G        C$BRK = 000022          C$SVEC= 000037          EM0509  010130 G        ER0503  014100 G
BIT12 = 010000 G        C$BSEG= 000004          C$TOME= 000076          EM1601  010134 G        ER1603  014124 G
BIT13 = 020000 G        C$BSUB= 000002          DELAY   017354 G        EM4401  010217 G        ER6201  014204 G
BIT14 = 040000 G        C$CLCK= 000062          DEST    034156          EM4402  010241 G        ER9001  014432 G
BIT15 = 100000 G        C$CLEA= 000012          DFPTBL  002152 G        EM4403  010307 G        ER9002  014522 G
BIT2  = 000004 G        C$CLOS= 000035          DIAGMC= 000000          EM4404  010404 G        ER9003  014670 G
BIT3  = 000010 G        C$CLP1= 000006          DLPRTB  005074 G        EM4405  010443 G        ER9004  015052 G
BIT4  = 000020 G        C$CPBF= 000074          DLPRTE  005102 G        EM4406  010537 G        ER9005  015154 G
BIT5  = 000040 G        C$CPME= 000075          DMRW    017466 G        EM4407  010613 G        ER9101  015402 G
BIT6  = 000100 G        C$CVEC= 000036          DMTSTA  002264 G        EM4408  010675 G        ER9102  015430 G
BIT7  = 000200 G        C$DCLN= 000044          DM16B   017414 G        EM4409  010740 G        ER9301  015542 G
BIT8  = 000400 G        C$DODU= 000051          DODMA   017610 G        EM4410  010775 G        EVL   = 000004 G
BIT9  = 001000 G        C$DRPT= 000024          DPENDB  003144 G        EM4411  011024 G        EXCNTB  003244 G
BMPCQB  002514 G        C$DU  = 000053          DPLENB  003204 G        EM5303  011071 G        EXTMSG  013225 G
BMPCQE  002714 G        C$EDIT= 000000          DPRSQB  004644 G        EM6201  011142 G        E$END = 002100
BMPCQP  002512 G        C$ERDF= 000055          DPRSQE  005044 G        EM6202  011166 G        E$LOAD= 000035
BOE   = 000400 G        C$ERHR= 000056          DRADRT  002202 G        EM6301  011175 G        FFREM   002262 G
BRLEVL  002177 G        C$ERRO= 000060          DROP    031050          EM8901  011220 G        FINACT  017756 G
BRTBLB  002426 G        C$ERSF= 000054          DUMY    034154          EM9001  011245 G        FRPSUP  020036 G
BRTBLE  002466 G        C$ERSO= 000057          EDPFMT  007764 G        EM9003  011301 G        F$AU  = 000015
BUFBAS  003604 G        C$ESCA= 000010          EDROP   031126          EM9004  011323 G        F$AUTO= 000020
BUFEND  004604 G        C$ESEG= 000005          EF.CON= 000036 G        EM9006  011341 G        F$BGN = 00004C
BUFMID  004204 G        C$ESUB= 000003          EF.NEW= 000035 G        EM9007  011414 G        F$CLEA= 000007
BUF3QT  004404 G        C$ETST= 000001          EF.PWR= 000034 G        EM9008  011477 G        F$DU  = 000016
CALMSL  015774 G        C$EXIT= 000032          EF.RES= 000037 G        EM9009  011560 G        F$END = 000041
CBB     003124 G        C$FREQ= 000101          EF.STA= 000040 G        EM9010  011604 G        F$HARD= 000004
CBDPAA  003130 G        C$FRME= 000100          EF0503  005461 G        EM9011  011630 G        F$HW  = 000013
CBDPLA  003132 G        C$GETB= 000026          EF1601  005466 G        EM9012  011640 G        F$INIT= 000006
CBDPNA  003134 G        C$GETW= 000027          EF1603  005512 G        EM9013  011650 G        F$JMP = 000050
CBLNCA  003126 G        C$GMAN= 000043          EF4401  005554 G        EM9014  011657 G        F$MOD = 000000
CBLPBA  003140 G        C$GPHR= 000042          EF6201  005671 G        EM9015  011753 G        F$MSG = 000011
CBLPRA  003124 G        C$GPRI= 000040          EF6202  006004 G        EM9016  011767 G        F$PROT= 000021
CBMAPA  003136 G        C$INIT= 000011          EF6203  006102 G        EM9017  011776 G        F$PWR = 000017
```

SEQ 0243

Symbol table

```
F$RPT = 000012
F$SEG = 000003
F$SOFT= 000005
F$SRV = 000010
F$SUB = 000002
F$SW  = 000014
F$TEST= 000001
GETBDR  020302 G
GETCHR  020430 G
GETLP1  020512 G
GETLP2  020664 G
GETPRM  030530
GETTIM  021014 G
GMANWD  002266 G
GPRSOB  002466 G
G$CNTO= 000200
G$DELM= 000372
G$DISP= 000003
G$EXCP= 000400
G$HILI= 000002
G$LOLI= 000001
G$NO  = 000000
G$OFFS= 000400
G$OFSI= 000376
G$PRMA= 000001
G$PRMD= 000002
G$PRML= 000000
G$RADA= 000140
G$RADB= 000000
G$RADD= 000040
G$RADL= 000120
G$RADO= 000020
G$XFER= 000004
G$YES = 000010
HELP  = 000000
HOE   = 100000 G
HWPTQ1  037432
HWPTQ2  037450
HWPTQ3  037503
HWPTQ4  037531
HWPTQ5  037670
IBE   = 010000 G
IBM     002226 G
IDU   = 000040 G
IER   = 020000 G
IESTAT  002234 G
INICHR  021072 G
INIDMA  021204 G
ISR   = 000100 G
IXE   = 004000 G
I$AU  = 000041
I$AUTO= 000041
I$CLN = 000041
I$DU  = 000041
I$HRD = 000041
I$INIT= 000041
I$MOD = 000041
I$MSG = 000041
I$PROT= 000040
I$PTAB= 000041
I$PWR = 000041
I$RPT = 000041
I$SEG = 000041
I$SETU= 000041
I$SFT = 000041
I$SRV = 000041
I$SUB = 000041
I$TST = 000041
J$JMP = 000167
LGRP1M  002230 G
LGRP2M  002232 G
LINBIT  021276 G
LNCTRA  002212 G
LNCTRO= 000010 G
LOE   = 040000 G
LOPBCK  002176 G
LOT   = 000010 G
LPCSLT= 000036 G
LPRA    002206 G
LPRO  = 000004 G
L$ACP   002110 G
L$APT   002036 G
L$AU    031134 G
L$AUT   002070 G
L$AUTO  030772 G
L$CCP   002106 G
L$CLEA  030774 G
L$CO    002032 G
L$DEPO  002011 G
L$DESC  005376 G
L$DESP  002076 G
L$DEVP  002060 G
L$DISP  002124 G
L$DLY   002116 G
L$DTP   002040 G
L$DTYP  002034 G
L$DU    031024 G
L$DUT   002072 G
L$DVTY  005366 G
L$EF    002052 G
L$ENVI  002044 G
L$ERRT  005316 G
L$ETP   002102 G
L$EXP1  002046 G
L$EXP4  002064 G
L$EXP5  002066 G
L$HARD  037354 G
L$HIME  002120 G
L$HPCP  002016 G
L$HPTP  002022 G
L$HW    002152 G
L$ICP   002104 G
L$INIT  030130 G
L$LADP  002026 G
L$LAST  040246 G
L$LOAD  002100 G
L$LUN   002074 G
L$MREV  002050 G
L$NAME  002000 G
L$PRIO  002042 G
L$PROT  030122 G
L$PRT   002112 G
L$REPP  002062 G
L$REV   002010 G
L$RPT   030114 G
L$SOFT  037720 G
L$SPC   002056 G
L$SPCP  002020 G
L$SPTP  002024 G
L$STA   002030 G
L$SW    002164 G
L$TEST  002114 G
L$TIML  002014 G
L$UNIT  002012 G
L10000  002162
L10001  002170
L10002  013664
L10003  014122
L10004  014202
L10005  014430
L10006  014520
L10007  014666
L10010  015050
L10011  015152
L10012  015400
L10013  015426
L10014  015540
L10015  015720
L10016  030120
L10020  030770
L10021  030772
L10022  031022
L10023  031132
L10024  031140
L10025  031430
L10026  032002
L10027  032366
L10030  034164
L10031  035056
L10032  035304
L10033  036034
L10034  036644
L10035  037270
L10036  037350
L10037  037432
L10040  037746
MAPLNS= 000377 G
MFUNIT  005430 G
MMENAB  002324 G
MMPRES  002322 G
MMSR0   002316 G
MMSR3   002320 G
MODSUP  021324 G
MSFMT1  007630 G
MSFMT2  007670 G
MSG1    013666 G
MSG2    013744 G
MSG3    014023 G
MSLCNT  002314 G
MSLGET  021464 G
MSLOOP  021600 G
MSSRPT  021614 G
MSTICK  002312 G
MUL16U  022030 G
NDERPT  002166 G
NDPMSG  013306 G
NEWCHR  022104 G
NEWPAS  030510
NEWRES  030502
NEWSTA  030174
NUMLNS= 000010 G
ODTSTA  034160
OOPS    022364 G
OPTION  002164 G
O$APTS= 000000
O$AU  = 000000
O$BGNR= 000001
O$BGNS= 000001
O$DU  = 000001
O$ERRT= 000001
O$GNSW= 000001
O$POIN= 000001
O$SETU= 000000
PARATB  002326 G
PARATE  002346 G
PAR0A   002326 G
PAR1A   002330 G
PAR2A   002332 G
PAR3A   002334 G
PAR4A   002336 G
PAR5A   002340 G
PAR6A   002342 G
PAR7A   002344 G
PASCNT  002236 G
PCSLOT= 000016 G
PDRATB  002346 G
PDRATE  002366 G
PDR0A   002346 G
PDR1A   002350 G
PDR2A   002352 G
PDR3A   002354 G
PDR4A   002356 G
PDR5A   002360 G
PDR6A   002362 G
PDR7A   002364 G
PMSFLG  002270 G
PMSMSG  013434 G
PNT   = 001000 G
PREGRT  005350 G
PREG05  005326
PRFRME  022612 G
PRI   = 002000 G
PRI00 = 000000 G
PRI01 = 000040 G
PRI02 = 000100 G
PRI03 = 000140 G
PRI04 = 000200 G
PRI05 = 000240 G
PRI06 = 000300 G
PRI07 = 000340 G
PROTBL  005216 G
PRPARE  022710 G
PRTLPR  023070 G
PUFIFO  023152 G
PUFIFR  023234 G
PURRXB  023426 G
RBUFA   002204 G
RBUFO = 000002 G
RDCHRS  023462 G
RDMAST  024034 G
REPCOD  024066 G
REPSMR  024220 G
RESETT  024246 G
RRXNDN  024360 G
RTXNDN  024420 G
RXBCNT  002720 G
RXBDTX= 000030 G
RXBEND  003122 G
RXBETX= 000020 G
RXBFUL= 000100 G
RXBIPT  002716 G
RXBOPT  002714 G
RXBSTA  002722 G
RXCHRS  027364 G
RXCNTB  003544 G
RXDONF  002506 G
RXDSBL  024460 G
RXENBL  024554 G
RXIE0   024650 G
RXIE1   024702 G
RXPTRB  003404 G
RXTOUT  002242 G
RXVECA  002170 G
R0SLOT= 000002 G
R1SLOT= 000004 G
R2SLOT= 000006 G
R3SLOT= 000010 G
R4SLOT= 000012 G
R5SLOT= 000014 G
SAVBMP  024726 G
SAVPRI  002246 G
SAVTEN  002244 G
SCBCTB  005044 G
SCBCTE  005054 G
SCBRTB  005054 G
```

SEQ 0244

Symbol table

```
SCBRTE  005062 G        SWPTQ3  040111          TXDSBL  025552 G        T$LTNO= 000012          T$$TES= 010036
SCNSTB  005062 G        S$LSYM= 010000          TXENBL  025646 G        T$NEST= 177777          T1      031142 G
SCNSTE  005066 G        TERMSG  013521 G        TXENBM  002250 G        T$NS0 = 000000          T10     037272 G
SCTPTB  005066 G        TIMER1  002302 G        TXFRPR  025742 G        T$NS1 = 000005          T2      031432 G
SCTPTE  005074 G        TIMER2  002304 G        TXIE0   026036 G        T$PTNU= 000000          T3      032004 G
SDPBAS  005132 G        TIMER3  002306 G        TXIE1   026070 G        T$SAVL= 177777          T4      032370 G
SDPEND  005152 G        TNUM   = 000012 G        TXINTF  002252 G        T$SEGL= 177777          T5      034166 G
SDP2B   005156 G        TP4BRT  027530 G        TXPTRB  003344 G        T$SUBN= 000000          T6      035060 G
SDP2E   005176 G        TP4FLG  002256 G        TXRINI  026114 G        T$TAGL= 177777          T7      035306 G
SFPTBL  002164 G        TP4RTN  027552 G        TXROFF  026370 G        T$TAGN= 010041          T8      036036 G
SKPSTS  024774 G        TP4VEC  002254 G        TXRON   026424 G        T$TEMP= 000000          T9      036646 G
SPLPRB  005102 G        TRPAD2  017216 G        TXRREP  026450 G        T$TEST= 000012          UAM    = 000200 G
SPLPRE  005132 G        TSTNUM  002224 G        TXRXLB  005236 G        T$TSTM= 177777          UBRFMT  007532 G
SPLSUP  025052 G        TXAD1A  002214 G        TXRXLE  005276 G        T$TSTS= 000001          UNITN   002200 G
STATA   002210 G        TXAD10= 000012 G        TXSCHR  027726 G        T$$AU = 010024          UNSDIV  026530 G
STATO = 000006 G        TXAD2A  002216 G        TXVECA  002172 G        T$$AUT= 010021          UPDCHR  026664 G
STGTRB  005276 G        TXAD20= 000014 G        T$ARGC= 000002          T$$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$FALS= 000040
SVCTST= 000001          TXDMA   027574 G        T$LAST= 000001          T$$RPT= 010016          X$OFFS= 000400
SWAP0   025372 G        TXDONE  025442 G        T$LOLI= 000000          T$$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 SEQ 0245

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 SEQ 0246

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					
CSR0	• 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								

SEQ 0247

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$GETB	= 000026	#28-1183
C$GFTW	= 000027	#28-1183
C$GPAN	= 000043	#28-1183 75-3959 145-8778 145-8791 145-8799 145-8848 146-8951
C$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 SEQ 0248

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	034150		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	*15-6321	*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	48-2417	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 SEQ 0249

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	#89-48 8			
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			
EM4[illegible]	010537	G	#41-2118	144-8493			
EM440[illegible]	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 SEQ 0250

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 *149-9360 *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	670	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 CROSS REFERENCE CREF 03.00 SEQ 0251

SYMBOL	VALUE		REFERENCES
F$BGN	= 000040		#28-1183 28-1205 43-2211 44-2249 45-2276 46-2314 47-2373 48-2403 49-2439 50-2499 51-2539 52-[illegible] 53-2635 54-2671 134-7394 135-7417 136-7460 137-7643 138-7670 138-7689 139-7723 140-7761 141-7792 141-7874 142-7891 142-7999 143-8017 143-8135 144-8150 144-8626 144-8650 145-8684 145-8694 145-8697 145-8871 146-8888 146-8898 146-8901 146-8957 147-8974 147-9102 148-9117 148-9286 149-9298 149-9417 150-9429 150-9448 151-9472 152-9534 153-9585
F$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 28-1205 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 8626 144-8650 144-8650 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 147-8974 147-9102 147-9102 148-9117 148-9117 148-9286 148-9286 149-9298 149-9298 149-9298 149-9417 149-9417 149-9429 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$[illegible]	= 000012		#28-1183 134-7394 134-7400
F$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$TEST	= 000001		#28-1183 141-7792 141-7874 142-7891 142-7999 143-8017 145 8135 144-8150 144-8650 145-8684 145-8871 146-8888 146-8957 147-8974 147-9102 148-9117 148-9286 149-9298 149-9417 150-9429 150-9448
GETBDR	020302	G	#75-3953 145-8736 146-8929
GETCHR	020430	G	#76-4011 96-5240 96-5302
GETLP1	020512	G	#77-4062 147-9013
GETLP2	020664	G	#78-4128 147-9062
GETPRM	030530		136-7472 136-7546 #136-7550 136-7557
GETTIM	021014	G	#79-4178 142-7939 143-8061 145-8747 147-9017 147-9066 148-9148 148-9220 149-9349
GMANWD	002266	G	#33-1436 75-3954 *75-3958 75-3959 75-3960 *75-3981 *145-8777 145-8778 *145-8790 145-8791 145-8792 *145-8847 145-8848 145-8849 *146-8950 146-8951
GPRSOB	002466	G	#33-1527 111-6077 *147-9010 *147-9059
G$CNTO	= 000200		#28-1183
G$DELM	= 000372		#28-1183 147-9030
G$DISP	= 000003		#28-1183
G$EXCP	= 000400		#28-1183

SEQ 0252

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$MSG	= 000041		#28-1183	#43-2211	#43-2222	#44-2249	#44-2253	#45-2276	#45-2285	#46-2314	#46-2351
			#47-2373	#47-2379	#48-2403	#48-2415	#49-2439	#49-2476	#50-2489	#50-2514	#51-2539
			#51-2581	#52-2602	#52-2606	#53-2635	#53-2649	#54-2671	#54-2703		
I$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 SEQ 0253

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 #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 #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$JMP	= 000167		#28-1185 134-7396 139-7739 140-7770
LGRP1M	002230	G	#33-1420 109-5939 109-5991 142-7942 143-8064
LGRP2M	002232	G	#33-1421 109-5958 109-5980
LINBIT	021276	G	68-3532 #82-4356
LNCTRA	002212	G	#33-1409 103-5663 104-5716 126-6967
LNCTRO	= 000010	G	#32-1346
LOE	= 040000	G	#32-1371
LOPBCK	002176	G	#33-1397 74-3861 109-5940 109-5959 124-6855 124-6860 *136-7570 142-7895 143-8021 145-8692 146-8896 149-9299
LOT	= 000010	G	#32-1371
LPCSLT	= 000036	G	#35-1852
LPRA	002206	G	#33-1407 92-4993 127-7006
LPRO	= 000004	G	#32-1344
L$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 SEQ 0254

SYMBOL	VALUE		REFERENCES		
L$EXPS	002066	G	#28-1231		
L$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 SEQ 0255

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	*136-7486	*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								

SEQ 0256

SYMBOL CROSS REFERENCE CREF 03.00

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
PREGOS	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

SEQ 0257

SYMBOL CROSS REFERENCE CREF 03.00

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					
RXIE0	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
R0SLOT	= 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 CROSS REFERENCE   CREF   03.00

```
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    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    39-2018    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    28-1231
                         28-1231    28-1231    28-1231    28-1231    28-1231    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
                         43-2219    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    43-2220
                         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
```

SEQ 0259

SYMBOL CROSS REFERENCE CREF 03.00

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	46-2345	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	46-2347	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	49-2445	49-2445	49-2445	49-2445
49-2445	49-2445	49-2445	49-2445	49-2445	49-2445	49-2445	49-2445	49-2445
49-2445	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	49-2472	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	50-2508	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	51-2550
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	51-2554	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	51-2573	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
53-2638	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 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 SEQ 0260

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 75-3959 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 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 86-4626 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 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 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 136-7479 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 136-7536
136-7536 136-7536 136-7555 136-7555 136-7555 136-7555 136-7555 136-7555 136-7555
136-7555 136 7555 136-7556 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 136-7604 136-7604
136-7604 136-7604 136-7604 136-7604 136-7604 136-7620 136-7620 136-7620 137-7652
137-7652 137-7652 138-7683 138-7683 138-7683 138-7689 138-7689 138-7689 138-7689
138-7689 138-7689 138-7705 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 139-7731 139-7731
139-7731 139-7731 139-7731 139-7731 139-7731 139-7739 139-7739 139-7739 139-7739
139-7739 139-7739 139-7742 139-7742 139-7742 140-7770 140-7770 140-7770 140-7770
140-7770 140-7770 140-7775 140-7775 140-7775 141-7862 141-7862 141-7862 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 141-7864 141-7866 141-7866 141-7866 141-7874
141-7874 141-7874 142-7916 142-7916 142-7916 142-7916 142-7916 142-7916 142-7916
142-7916 142-7916 142-7916 142-7916 142-7916 142-7916 142-7916 142-7916 142-7916
142-7916 142-7916 142-7990 142-7990 142-7990 142-7990 142-7990 142-7990 142-7999

SEQ 0261

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 144-8619 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 145-8 18 145-8718 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 145-8778 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 145-8799 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
147-8996 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 147-8997 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 147-9030
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 SEQ 0262

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
			151-9490	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	151-9495
			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
			152-9547	152-9547	152-9549	152-9549	152-9549	152-9549	152-9549	152-9549	152-9549
			152-9549	152-9549	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	#145-8799	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	128-7044	*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	131-7160	*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 CROSS REFERENCE CREF 03.00 SEQ 0263

SYMBOL	VALUE		REFERENCES								
			*148-9119	*149-9303	*150-9431						
TXAD1A	002214	G	#33-1410	72-3754	144-8480						
TXAD1O	= 000012	G	#32-1347								
TXAD2A	002216	G	#33-1411	72-3750	72-3755	113-6186	114-6240	144-8476	144-8481		
TXAD2O	= 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	*142-792?	*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 SEQ 0264

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	#51-2573	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	[illegible]	#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$CODE	= 000130	#75-3959	75-3959	#75-3959	75-3959	#75-3959	75-3959	#145-8778	145-8778	#145-8778
		145-8778	#145-8778	145-8778	#145-8791	145-8791	#145-8791	145-8791	#145-8791	145-8791
		#145-8799	145-8799	#145-8799	145-8799	#145-8799	145-8799	#145-8848	145-8848	#145-8848
		145-8848	#145-8848	145-8848	#146-8951	146-8951	#146-8951	146-8951	#146-8951	146-8951
		#151-9484	151-9484	#151-9484	151-9484	#151-9484	151-9484	#151-9486	151-9486	#151-9486
		151-9486	#151-9486	151-9486	#151-9488	151-9488	#151-9488	151-9488	#151-9488	151-9488
		#151-9490	151-9490	#151-9490	151-9490	#151-9490	151-9490	#151-9492	151-9492	#151-9492
		151-9492	#151-9492	151-9492	#152-9545	152-9545	#152-9545	152-9545	#152-9545	152-9545
		#152-9547	152-9547	#152-9547	152-9547	#152-9547	152-9547	#152-9549	152-9549	#152-9549
		152-9549	#152-9549	152-9549						
T$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$GMAN	= 000000	#28-1183	#75-3959	[illegible]-3959	#145-8791	#145-8791				
T$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	#145-879[illegible]	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$LSYM	= 010000	#28 3	28-1183	30-1285	31-1309	43-2222	44-2253	45-2285	46-2351	47-2379
		48-2415	49-2476	50-2514	51-2581	52-2606	53-2649	54-2703	134-7400	136-7620
		137-7652	138-7705	139-7742	140-7775	141-7874	142-7999	143-8135	144-8650	145-8871
		146-8957	147-9102	148-9286	149-9417	150-9448	151-9495	152-9552		
T$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	48-2403	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	#51-253[illegible]	51-2539	51-2581	51-2581	51-2581	#51-2581

SEQ 0265

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

SEQ 0266

SYMBOL CROSS REFERENCE CREF 03.00

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 #145-8848 145-8848 #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$TEST	= 000012	#28-1183 141-7792 #141-7792 141-7792 142-7891 #142-7891 142-7891 143-8017 #143-8017
		143-8017 144-8150 #144-8150 144-8150 145-8684 #145-8684 145-8684 146-8888 #146-8888
		146-8888 147-8974 #147-8974 147-8974 148-9117 #148-9117 148-9117 149-9298 #149-9298
		149-9298 150-9429 #150-9429 150-9429 153-9584
T$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 48-2413 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 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 146-8901 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
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	#[illegible] 43-2222 #44-2249 44-2253 #45-2276 45-2285 #46-2314 46-2351 #47-2373
		[illegible] #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 CROSS REFERENCE CREF 03.00 SEQ 0267

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	29-12 .	#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 CROSS REFERENCE CREF 03.00 SEQ 0268

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 CROSS REFERENCE CREF 03.00 SEQ 0269

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	48-2411	#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	54-2695	#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	86-4626
	#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$DATA	#28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231
	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	#28-1231	28-1231	28-1231	28-1231
	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231
	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231	28-1231
	39-2008	#39-2018	39-2018							
M$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$EXIT	#134-7396	#138-7689	138-7689	#139-7739	#140-7770	#144-8626	144-8626	#145-8694	145-8694	#145-8697
	145-8697	#146-8898	146-8898	#146-8901	146-8901					
M$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								

```
CVDHCDO    CREATED BY  MACRO  ON 25-APR-85 AT 16:39      PAGE 26

MACRO CROSS REFERENCE                               CREF    03.00                                   SEQ 0270

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  #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$GNGB          #28-1205  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231
                 28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231
                 28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231
                 28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231
                 28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231
                 28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231
                 28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231
                 28-1231  #28-1231   28-1231  #28-1231   28-1231  #28-1231   28-1231  #29-1251   29-1251  #30-1277
                 30-1277   30-1277  #31-1304   31-1304   31-1304  #33-1803   33-1803  #39-2008   39-2008  #39-2018
                 39-2018  #43-2211   43-2211  #44-2249   44-2249  #45-2276   45-2276  #46-2314   46-2314  #47-2373
                 47-2373  #48-2403   48-2403  #49-2439   49-2439  #50-2499   50-2499  #51-2539   51-2539  #52-2602
                 52-2602  #53-2635   53-2635  #54-2671   54-2671  #134-7394  134-7394 #135-7417  135-7417 #136-7460
                 136-7460 #137-7643  137-7643 #138-7670  138-7670 #139-7723  139-7723 #140-7761  140-7761 #151-9472
                 151-9472 #152-9534  152-9534 #153-9584  153-9584
M$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 SEQ 0271

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  #45-2285  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  #46-2332  #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
#48-24..  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  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
#51-2573  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  75-3978  #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
```

MACRO CROSS REFERENCE CREF 03.00 SEQ 0272

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 #139-7739 139-7739
#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 142-7916
#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 #145-8799 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 #145-8848 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 #147-9030 147-9030 147 9030 147-9030 147-9030
147-9030 147-9030 147-9030 147-9030 #147-9098 147-9098 #147-9098 147-9098 #147-9099 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 #148-9137 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 #148-9279 148 9279
#148-9279 148-9279 #148-9280 148-9280 #148-9280 148-9280 #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
#149-9322 #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
	151-9495	#152-9534	152-9534	#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	#152-9552	152-9552	#153-9584
	153-9584	#153-9584	153-9584	#153-9584	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	#147-8996	#147-8997	#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 SEQ 0274

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	54-2674
	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	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	#145-8831	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	#147-8997	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 SEQ 0276

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
M$TLAB	#43-2216	#43-2219	#43-2220	#43-2222	#44-2251	#44-2253	#45-2279	#45-2282	#45-2285	#46-2320
	#46-2332	#46-2345	#46-2347	#46-2351	#47-2375	#47-2376	#47-2377	#47-2379	#48-2407	#48-2408
	#48-2411	#48-2413	#48-2415	#49-2442	#49-2445	#49-2472	#49-2476	#50-2501	#50-2508	#50-2514
	#51-2542	#51-2550	#51-2554	#51-2573	#51-2581	#52-2604	#52-2606	#53-2638	#53-2643	#53-2647
	#53-2649	#54-2674	#54-2695	#54-2699	#54-2703	#62-3263	#75-3959	#75-3978	#86-4593	#86-4626
	#86-4634	#88-4766	#88-4783	#89-4828	#89-4830	#89-4831	#90-4888	#91-4941	#91-4960	#92-4999
	#94-5120	#94-5134	#96-5270	#97-5359	#98-5415	#99-5462	#100-5531	#101-5585	#102-5631	#128-7051
	#134-7400	#136-7462	#136-7465	#136-7468	#136-7471	#136-7475	#136-7479	#136-7489	#136-7536	#136-7555
	#136-7604	#136-7620	#137-7652	#138-7683	#138-7689	#138-7705	#139-7731	#139-7742	#140-7775	#141-7862
	#141-7864	#141-7866	#141-7874	#142-7916	#142-7990	#142-7999	#143-8042	#143-8125	#143-8135	#144-8321
	#144-8615	#144-8619	#144-8626	#144-8650	#145-8694	#145-8695	#145-8697	#145-8717	#145-8718	#145-8728
	#145-8778	#145-8791	#145-8799	#145-8831	#145-8848	#145-8864	#145-8865	#145-8871	#146-8898	#146-8899
	#146-8901	#146-8919	#146-8951	#146-8957	#147-8996	#147-8997	#147-9098	#147-9099	#147-9102	#148-9137
	#148-9138	#148-9279	#148-9280	#148-9286	#149-9321	#149-9322	#149-9412	#149-9417	#150-9443	#150-9448
M$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	89-4828	#89-4830

MACRO CROSS REFERENCE CREF 03.00 SEQ 0277

MACRO NAME REFERENCES

	89-4830	#89-4831	89-4831	#90-4888	90-4888	#91-4941	91-4941	#91-4960	91-4960	#92 [illegible]9
	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 [illegible]99	146-8899	#146-8901	146-8901	#146-8919	146-8919	#146-8951	146-8951	#146-89[illegible]7	146-8957
	#147-8996	147-8996	#147-8997	147-8997	#147-9098	147-9098	#147-9099	147-9099	#147-9[illegible]02	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	63-336[illegible]	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-[illegible]51	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-[illegible]	145-8728	146-8919	
PRINTX	43-2220	46-2332	46-2345	46-2347	47-2376	47-2377	48-[illegible]	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 SEQ 0278

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