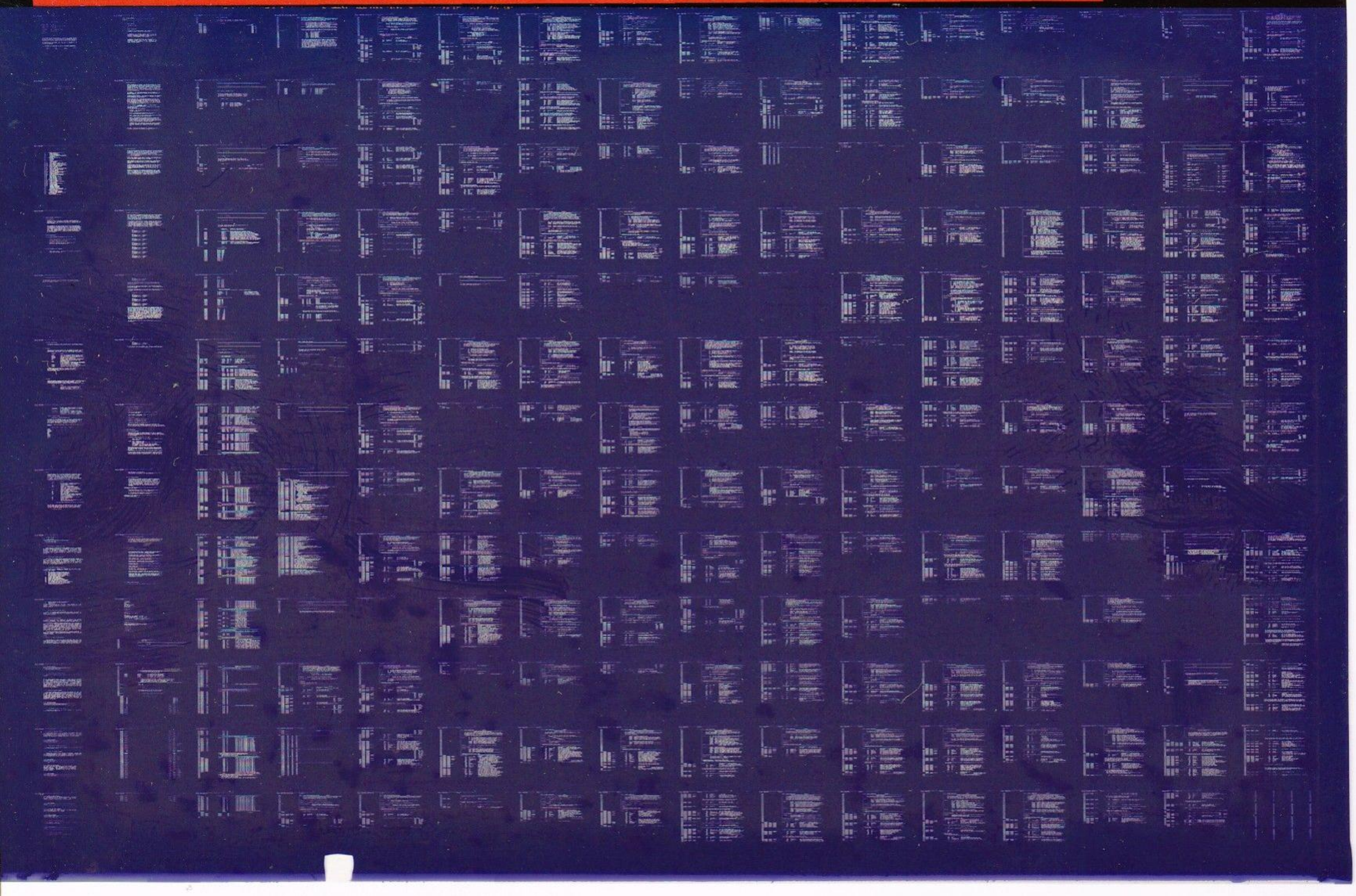
DHU11

DHU-11 FUNC TST PART 3
CZDHWAO

AH-T799A-MC
FICHE 1 OF 2

APR 1984
digital
Made In USA



DHU11

DHU-11 FUNC TST PART 3
CZDHWA0

AH-T799A-MC
FICHE 2 OF 2

APR 1984
digital
Made In USA

```
.REM &


                IDENTIFICATION
                - ------------

PRODUCT CODE:   AC-T798-MC

PRODUCT NAME:   CZDHWAO DHU-11 FUNC TST PART3

PRODUCT DATE:   15 DEC 1983

MAINTAINER:     ENE   DIAGNOSTICS GROUP

AUTHOR:         ANTHONY HART

MODIFIED BY:



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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF
SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS
AFFILIATED COMPANIES.
```


THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

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

ORIGINAL RELEASE: 15 DEC 1983 ANTHONY HART

TABLE OF CONTENTS

1.0 GENERAL PROGRAM CONSIDERATIONS

1.1 PROGRAM ABSTRACT

CZDHWAO IS PART OF THE DHU-11 FUNCTIONAL VERIFICATION TEST. THIS PART OF THE TEST PERFORMS EXTENSIVE DATA TRANSMISSION AND RECEPTION TESTS. THIS PART ALSO INCLUDES A KEYBOARD ECHO AND MODEM LOOPBACK TEST.

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 DHU FVT:

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

1.3 RELATED DOCUMENTS AND STANDARDS

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

1.4 DIAGNOSTIC HIERARCY PREREQUISITES

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

1.5 ASSUMPTIONS

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

```
/FLAGS:LOE:IER:BOE
```

2.4 EXTENDED COMMAND SYNTAX

2.4.1 START COMMAND -

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

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

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

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

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

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

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

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

IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC.
(HAS NO EFFECT IN THIS DIAGNOSTIC.)
LOT LOOP ON TEST.
THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO 0 ARE CLEARED. A FLAG NOT SPECIFIED IS CLEARED. IF THE FLAGS SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED. SEE EXAMPLE AT END OF "EFFECT OF START COMMAND" SECTION.

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

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

2.4.1.5 EFFECT OF START COMMAND -

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

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE QUESTION "# 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 UNITS SWITCH SHOULD NOT BE USED WITH THIS PROGRAM. THE SOFTWARE DIALOGUE MAY OPTIONALLY BE REEXECUTED (OPERATOR WILL BE ASKED). THE COMMAND CAN BE USED AFTER COMMAND MODE HAS BEEN REENTERED IN ANY OF THE THREE NORMAL WAYS: A) THE REQUESTED NUMBER OF PASSES HAVE BEEN MADE, B) AN ERROR WAS ENCOUNTERED WITH THE HALT ON ERROR FLAG SET, OR C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

2.4.3 CONTINUE COMMAND -

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

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

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

2.4.3.2 EFFECT OF CONTINUE COMMAND

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

2.4.4 PROCEED COMMAND -

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

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

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

2.4.4.2 EFFECT OF PROCEED COMMAND -

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

2.4.5 ADD COMMAND -

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

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

2.4.7 DROP COMMAND -

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

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

2.4.9 PRINT COMMAND
****
PRI(NT)
****

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

2.4.10 DISPLAY COMMAND -
****
DIS(PLAY)/UNITS:<UNIT LIST>
****

2.4.10.1 EFFECT OF DISPLAY COMMAND -

THE HARDWARE P-TABLE FOR THE TEST STATION IS PRINTED IN THE FORMAT IN WHICH IT WAS ENTERED.

2.4.11 FLAGS COMMAND -
****
FLA(GS)
****

2.4.11.1 EFFECT OF FLAGS COMMAND -

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

2.4.12 ZFLAGS COMMAND -

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

2.4.13 ZFLAGS COMMAND -

ALL LAGS 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 TYP"NG "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 REQUESTES THE CSR ADDRESS OF THE SPECIFIED DHU-11. THE DEFAULT ANSWER FOR THIS QUESTION IS ADDRESS 160460 (OCTAL).

2. INTERRUPT VECTOR ADDRESS - THIS QUESTION REQUESTS THE INTERRUPT VECTOR ADDRESS OF THE SPECIFIED DHU-11. THE DEFAULT ANSWER FOR THIS QUESTION IS 310 (OCTAL).

3. ACTIVE LINES BIT MAP - THIS QUESTION REQUESTS AN OCTAL BIT MAP OF THE SERIAL COMMUNICATION LINES ON THE DHU11 WHICH ARE BEING SELECTED FOR TESTING. IF THE BIT IN THE BIT MAP IS SET WHICH CORRESPONDS TO A PARTICULAR LINE (I.E. BIT 5 FOR LINE 5) THAT LINE WILL BE TESTED BY THE FVT. THE DEFAULT ANSWER FOR THIS QUESTION IS ALL LINES I.E. 177777.

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 WHEN TESTING THE DHU-11.
THE FOLLOWING TYPES ARE SUPPORTED:

   - INTERNAL - ONLY INTERNAL UART LOOPBACK IS TO BE USED IN TESTING THE DHU-11.
   - H3277 - STAGGERED BERG CONNECTOR(S) ARE INSTALLED ON THE BERG CONNECTOR SOCKETS OF THE DHU-11.
   - H325 - SINGLE LINE, 25 PIN LOOPBACK CONNECTORS (TYPE H325) ARE INSTALLED ON THE LINES TO BE TESTED.
   - 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 IS A SPECIAL TEST AND ALL OTHER TESTS IN THIS PART WILL BE PERFORMED IN INTERNAL LOOPBACK.
   - KEYBOARD ECHO - THE UARTS ON THE DUT ARE PLACE IN REMOTE LOOPBACK. TERMINALS (OR OTHER COMMUNICATIONS EQUIPMENT) WILL HAVE WHATEVER THEY TRANSMIT TO THE DHU LOOPED BACK TO THEM.

5. BR LEVEL - THIS QUESTION REQUESTS THE INTERRUPT BR LEVEL OF THE SPECIFIED DHU-11. THE DEFAULT ANSWER IS BR 5.

2.6 SOFTWARE QUESTIONS

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

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

2. REPORT NUMBER OF BITS TESTED IN DMA ADDR TEST - THIS QUESTION ASKS WHETHER THE OPERATOR WANTS A PRINTOUT DESCRIBING WHICH ADDRESS BITS HAVE BEEN TESTED WHEN THE DMA ADDRESSING TEST EXECUTES.

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

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

2.7 EXTENDED P-TABLE DIALOGUE

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2.8 QUICK START-UP PROCEDURE (XXDP+)

TO START-UP THIS PROGRAM:

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

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

3.0 ERROR INFORMATION

3.1 TYPES OF ERROR MESSAGES

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

THE GENERAL ERROR MESSAGE IS OF THE FORM:

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

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

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

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

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

3.2 SPECIFIC ERROR MESSAGES

THIS PROGRAM IS INTENDED TO PROVIDE A GO/NOGO INDICATION OF THE FUNCTIONALITY OF THE DHU-11 BOARDS. TO EXECUTE THE PROGRAM IN THIS MODE THE OPERATOR NEED ONLY ANSWER THE "EXTENDED ERROR REPORTING" SOFTWARE QUESTION WITH "NO". THE PROGRAM WILL THEN ONLY PRINT THE NAME OF THE FAILING TEST THE TEST AND ERROR NUMBERS. FOR A LIST OF THE TEST NAMES IN THIS PROGRAM SEE THE TEST SUMMARIES SECTION OF THIS DOCUMENT. AN EXAMPLE OF SUCH A AN ERROR MESSAGE IS THE FOLLOWING:

```
CZDHW DVC FTL ERR 4409 ON UNIT 00 TST 04 SUB 000 PC: XXXXXX
DMA ADDRESS TEST FAILED
```

THIS ERROR INDICATES THAT A FATAL ERROR WAS ENCOUNTERED DURING THE TEST WHICH TESTS THE DMA_ABORT BIT.

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

```
CZDHW DVC FTL ERR 4409 ON UNIT 00 TST 04 SUB 000 PC: XXXXXX
DMA ADDRESS TEST FAILED
 BAD BITS BETWEEN BITS 0 AND 15.
```

4.0 PERFORMANCE AND PROGRESS REPORTS

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

5.0 TEST SUMMARIES

THE FOLLOWING ARE INCLUDED WITHIN CZDHWA:

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

2. KEYBOARD ECHO TEST - ALLOWS THE OPERATOR TO TEST TERMINAL LINKS (OR OTHER COMMUNICATIONS LINKS), WHICH ARE ATTACHED TO UUT SERIAL PORTS, FROM REMOTE ENDS OF THE LINKS.

3. MODEM LOOPBACK TEST - ALLOWS THE OPERATOR TO TEST MODEM LINKS WHICH ARE ATTACHED TO THE UUT SERIAL PORTS.

4. DMA ADDR TEST - VERIFIES THAT THE UUT CAN ACCESS THE FULL MEMORY WHICH IS ON THE MACHINE VIA DMA ACCESS.

5. FRAMING ERROR TEST - VERIFIES THAT FORCED FRAMING ERRORS ARE REPORTED CORRECTLY.

6. PARITY ERROR TEST - VERIFIES THAT FORCED PARITY ERRORS ARE REPORTED CORRECTLY.

7. DMA MODE TEST - VERIFIES THAT THE UUT WILL TX AND RX DATA CORRECTLY USING DMA TRANSMISSION.

8. SPLIT SPEED TEST - VERIFIES THAT THE UUT WILL FUNCTION CORRECTLY USING DIFFERENT TX AND RX SPEEDS ON EACH ACTIVE LINE.

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

6.0 EXAMPLE ERROR FREE PASS

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

```
.R CZDHWAO
CZDHWAO.BIN
DRS
CZDHW-A-0
DHU FUNC TST PART3
UNIT IS DHU-11
RESTRT ADDR:  147670
DR>STA/PAS:1

CHANGE HW (L)  ? Y

# UNITS (D)  ? 2

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

UNIT 1
CSR ADDRESS:  (0)  160460  ? 160500
INTERRUPT VECTOR ADDRESS:  (0)  310  ? 320
ACTIVE LINE BIT MAP:  (0)  177777 ? <CR>
TYPE OF LOOPBACK (1=INTERNAL, 2=H3277, 3=H325
                  4=MODEM, 5=KEYBOARD ECHO):  (0)  2  ? 1
INTERRUPT BR LEVEL:  (0)  5  ? <CR>

CHANGE SW  (L)  ? Y

REPORT UNIT NUMBER AS EACH UNIT IS TESTED:  (L) Y  ? <CR>
REPORT NUMBER OF BITS TESTED IN DMA ADDR TEST: (L) N  ? <CR>
EXTENDED ERROR REPORTING:  (L) N  ? <CR>

 TESTING UNIT : 0

 TESTING UNIT : 1

CZDHW EOP       1
    0 TOTAL ERRS

DR>
```

```
                    E

1020                        .LIST SEQ,LOC,BIN,MEB
1021                        .NLIST CND
1022
1023                ;*******************************************************************************
1024                ;
1025                ;               FVTA.PHD
1026                ;
1027                ;*******************************************************************************
1028
1029
1030
1031                .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  CZDHW,A,0,200,0,PRIO7
002000                                                          L$NAME::
002000  103                                                             .ASCII /C/
002001  132                                                             .ASCII /Z/
002002  104                                                             .ASCII /D/
002003  110                                                             .ASCII /H/
002004  127                                                             .ASCII /W/
002005  000                                                             .BYTE   0
002006  000                                                             .BYTE   0
002007  000                                                             .BYTE   0
002010                                                          L$REV::
002010  101                                                             .ASCII  /A/
002011                                                          L$DEPO::
002011  060                                                             .ASCII  /0/
002012                                                          L$UNIT::
002012  000000                                                          .WORD   0
002014                                                          L$TIML::
002014  000200                                                          .WORD   200
002016                                                          L$HPCP::
002016  036702                                                          .WORD   L$HARD
002020                                                          L$SPCP::
002020  037246                                                          .WORD   L$SOFT
```

```
002022
002022  002150
002024
002024  002162
002026
002026  037636
002030
002030  000000
002032
002032  000000
002034
002034  000000
002036
002036  000000
002040
002040  002124
002042
002042  000340
002044
002044  000000
002046
002046  000000
002050
002050     003
002051     003
002052
002052  000000
002054  000000
002056
002056  000000
002060
002060  005364
002062
002062  027652
002064
002064  000000
002066
002066  000000
002070
002070  000000
002072
002072  030526
002074
002074  000000
002076
002076  005374
002100
002100  104035
002102
002102  005314
002104
002104  027666
002106
002106  030510
002110
002110  030506
002112
```

```
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$PRIO::
        .WORD   PRI07
L$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$ICP::
        .WORD   L$INIT
L$CCP::
        .WORD   L$CLEAN
L$ACP::
        .WORD   L$AUTO
L$PRT::
```

```
        002112  027660                          .WORD   L$PROT
        002114                          L$TEST::
        002114  000000                          .WORD   0
        002116                          L$DLY::
        002116  000000                          .WORD   0
        002120                          L$HIME::
        002120  000000                          .WORD   0
   1084
```

DISPATCH TABLE

```
1096                                    .SBTTL  DISPATCH TABLE
1097
1098                                    ;++
1099                                    ; THE DISPATCH TABLE CONTAINS THE STARTING ADDRESS OF EACH TEST.
1100                                    ; IT IS USED BY THE SUPERVISOR TO DISPATCH TO EACH TEST.
1101                                    ; -
1102
1103 002122                                    DISPATCH 9
     002122  000011                                                        .WORD   9
     002124                                                       L$DISPATCH::
     002124  030644                                                        .WORD   T1
     002126  031126                                                        .WORD   T2
     002130  031370                                                        .WORD   T3
     002132  032322                                                        .WORD   T4
     002134  034000                                                        .WORD   T5
     002136  034414                                                        .WORD   T6
     002140  035072                                                        .WORD   T7
     002142  036060                                                        .WORD   T8
     002144  036620                                                        .WORD   T9
1104
```

DISPATCH TABLE

```
                        ;*****************************************************************************
                        ;
                        ;               FVTA.DHT
                        ;
                        ;*****************************************************************************
                        .SBTTL  DEFAULT HARDWARE P-TABLE

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

002146                          BGNHW   DFPTBL
002146  000004                                                          .WORD   L10000-L$HW/2
002150                                                          L$HW::
002150                                                          DFPTBL::

002150  160460                  .WORD   160460  ;DEFAULT CSR ADDRESS
002152  000310                  .WORD   310     ;DEFAULT VECTOR ADDRESS
002154  177777                  .WORD   177777  ;DEFAULT ACTIVE LINES BIT MAP
002156     002                  .BYTE   2       ;DEFAULT LOOPBACK MODE
002157     005                  .BYTE   5       ;DEFAULT BR LEVEL

002160                          ENDHW
002160                                                          L10000:
```

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



                        .SBTTL  SOFTWARE P-TABLE

                        ;++
                        ; THE SOFTWARE TABLE CONTAINS VARIOUS DATA USED BY THE
                        ; PROGRAM AS OPERATIONAL PARAMETERS.  THESE PARAMETERS ARE
                        ; SET UP AT ASSEMBLY TIME AND MAY BE VARIED BY THE OPERATOR
                        ; AT RUN TIME.
                        ;--

002160                          BGNSW   SFPTBL
002160  000002                                                          .WORD   L10001-L$SW/2
002162                                                          L$SW::
002162                                                          SFPTBL::

002162  000020          OPTION::        .WORD   20      ;BIT MAP OF PROGRAM CONTROL FLAGS
002164  000000          NDERPT::        .WORD   0       ;DEFAULT NUMBER OF INDIVIDUAL DATA ERRORS TO RPT.

002166                          ENDSW
002166                                                          L10001:
```

```
                        ;*****************************************************************************
                        ;
                        ;               FVTA.EQU
                        ;
                        ;*****************************************************************************


                        .SBTTL  GLOBAL EQUATES SECTION









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

000020                          NUMLNS==20      ;NUMBER OF LINES ON DHV11 IS 8.
177777                          MAPLNS==177777  ;BIT MAP OF LINES ON DHV11.

                        ;****** 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                          RXTIM0==2       ;RECIEVE TIMER REGISTER OFFSET FROM THE CSR ADDRESS
000004                          LPR0==4         ;LINE PARAMETER REGISTER OFFSET FROM THE CSR ADDRESS
000006                          FSLS0==6        ;FIFOSIZE/STATUS REGISTER OFFSET FROM THE CSR ADDRESS
000006                          FDAT0==6        ;FIFODATA 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.


002166                          EQUALS
                        ;
                        ; BIT DIFINITIONS
                        ;
100000                  BIT15== 100000
040000                  BIT14== 40000
020000                  BIT13== 20000
010000                  BIT12== 10000
004000                  BIT11== 4000
002000                  BIT10== 2000
001000                  BIT09== 1000
000400                  BIT08== 400
000200                  BIT07== 200
000100                  BIT06== 100
000040                  BIT05== 40
000020                  BIT04== 20
000010                  BIT03== 10
000004                  BIT02== 4
```

```
000002                  BIT01== 2
000001                  BIT00== 1
                        ;
001000                  BIT9==  BIT09
000400                  BIT8==  BIT08
000200                  BIT7==  BIT07
000100                  BIT6==  BIT06
000040                  BIT5==  BIT05
000020                  BIT4==  BIT04
000010                  BIT3==  BIT03
000004                  BIT2==  BIT02
000002                  BIT1==  BIT01
000001                  BIT0==  BIT00
                        ;
                        ; EVENT FLAG DEFINITIONS
                        ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                        ;
000040                  EF.START==      32.                     ; START COMMAND WAS ISSUED
000037                  EF.RESTART==    31.                     ; RESTART COMMAND WAS ISSUED
000036                  EF.CONTINUE==   30.                     ; CONTINUE COMMAND WAS ISSUED
000035                  EF.NEW==        29.                     ; A NEW PASS HAS BEEN STARTED
000034                  EF.PWR==        28.                     ; A POWER-FAIL/POWER-UP OCCURRED
                        ;
                        ;
                        ; PRIORITY LEVEL DEFINITIONS
                        ;
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
000140                  PRI03== 140
000100                  PRI02== 100
000040                  PRI01== 40
000000                  PRI00== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==         4
000010                  LOT==        10
000020                  ADR==        20
000040                  IDU==        40
000100                  ISR==       100
000200                  UAM==       200
000400                  BOE==       400
001000                  PNT==      1000
002000                  PRI==      2000
004000                  IXE==      4000
010000                  IBE==     10000
020000                  IER==     20000
040000                  LOE==     40000
100000                  HOE==    100000
```

```
1230
1231                            ;*****************************************************************************
1232                            ;
1233                            ;               FVTC.GDT
1234                            ;
1235                            ;*****************************************************************************
1236
1237
1238
1239                            .SBTTL  GLOBAL DATA SECTION
1240
1241                            ;++
1242                            ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
1243                            ; IN MORE THAN ONE TEST.
1244                            ;--
1245
1246                            ;*****************************************************************************
1247                            ;       UNIT VARIBLE AREA
1248                            ;*****************************************************************************
1249
1250 002166  000300                     RXVECA:: .WORD  300     ;RX VECTOR ADDRESS.
1251 002170  000304                     TXVECA:: .WORD  304     ;TX VECTOR ADDRESS.
1252 002172  000377                     ACTLNS:: .WORD  377     ;ACTIVE LINE BIT MAP.
1253 002174     000                     LOPBCK:: .BYTE  0       ;LOOPBACK MODE
1254 002175     004                     BRLEVL:: .BYTE  4       ;INTERRUPT BUS REQUEST LEVEL
1255 002176  000000                     UNITN:: .WORD   0       ;UNIT NUMBER.
1256
1257
1258                            ;*****************************************************************************
1259                            ;       DEVICE REGISTER ADDRESS TABLE
1260                            ;*****************************************************************************
1261 002200                             DRADRT::
1262 002200  160020                     CSRA::  .WORD   160020  ;DHU-11 CSR ADDRESS.
1263 002202  160022             RXTMA:: RBUFA:: .WORD   160022  ;DHU-11 RECIEVE BUFFER/TIMER ADDRESS.
1264 002204  160024                     LPRA::  .WORD   160024  ;DHU-11 LINE PARAMETER REGISTER ADDRESS.
1265 002206  160026             FDATA:: FSLSA:: .WORD   160026  ;DHU-11 FIFO SIZE/LINE STATUS REGISTER ADDRESS,
1266                                                             ;AND FIFO DATA REGISTER ADDRESS.
1267 002210  160030                     LNCTRA:: .WORD  160030  ;DHU-11 LINE CONTROL REGISTER ADDRESS.
1268 002212  160032                     TXAD1A:: .WORD  160032  ;DHU-11 TRANSMIT BUFFER 1 REGISTER ADDRESS
1269 002214  160034                     TXAD2A:: .WORD  160034  ;DHU-11 TRANSMIT BUFFER 2 REGISTER ADDRESS
1270 002216  160036                     TXBFCA:: .WORD  160036  ;DHU-11 TRANSMIT BUFFER COUNT REGISTER ADDRESS
1271
1272                            ;*****************************************************************************
1273                            ;       ASSORTED GLOBAL VARIABLES:
1274                            ;*****************************************************************************
1275 002220  000000                     CTRLCF:: .WORD  0       ;STORAGE FOR THE CONTROL-C FLAG.
1276 002222  000000                     DMTSTA:: .WORD  0       ;STO'G FOR DMA TEST ADDRESS (IN PAR FORM).
1277 002224  000000                     FERROR:: .WORD  0       ;STORAGE FOR "AT LEAST ONE ERROR" INDICATOR.
1278 002226  000000                     FFREM::  .WORD  0       ;STO'G FOR ADR OF FIRST FREE WORD AFTER THE DIAG'TIC
1279 002230  000000                     GMANWD:: .WORD  0       ;WORD FOR GMANXX CALL RETURN PARAMETERS.
1280 002232  000000                     IBIT::   .WORD  0       ;INACTIVE TX/RX BITS MASK.
1281 002234  000000                     IESTAT:: .WORD  0       ;STORAGE FOR STATES OF THE DUT INT ENABLE BITS.
1282 002236  031463                     LGRP1M:: .WORD  31463   ;BIT MAP OF LINES IN LINE GROUP I.
1283 002240  146314                     LGRP2M:: .WORD  146314  ;BIT MAP OF LINES IN LINE GROUP II.
1284 002242  000000                     PASCNT:: .WORD  0       ;STO'G FOR PASS COUNT USED IN ROM VERSION# TST.
1285 002244  000000                     PMSFLG:: .WORD  0       ;FLAG INDICATING WHETHER TO PRINT MODEM STATUS.
1286 002246  000000                     RXTOUT:: .WORD  0       ;TIME-OUT VALUE FOR WAITING FOR LAST RX CHAR.
```

```
1287 002250  000000                SAVPRI:: .WORD  0       ;STO'G FOR PROCESSOR PRIORITY, (TXROFF, TXRON).
1288 002252  000000                SAVTEN:: .WORD  0       ;STORAGE FOR TX.ENABLE STATES, (TXROFF, TXRON).
1289 002254  000000                TP4FLG:: .WORD  0       ;FLAGS SET WHEN AN EXPECTED 004 TRAP OCCURS.
1290 002256  000000                TP4VEC:: .WORD  0       ;STORAGE FOR THE NORMAL 004 TRAP VECTOR.
1291 002260  000001                TSTNUM:: .WORD  1       ;STORAGE FOR THE TEST NUMBER.
1292 002262  000000                TXENBM:: .WORD  0       ;STORAGE FOR TX.ENABLE STATES, (BUFFER MGM'NT).
1293 002264  000000                TXINTF:: .WORD  0       ;STORAGE FOR TRANSMIT INTERRUPT FLAGS.
1294 002266  000000                WORD1::  .WORD  0       ;LOCATION FOR PASSING INDIRECT PARAMETERS.
1295
1296                        ;*****************************************************************************
1297                        ;       LINE TIME CLOCK VARIABLES AND STORAGE.
1298                        ;*****************************************************************************
1299 002270  177546                CLKCSR:: .WORD  177546  ;CSR ADDRESS OF THE LTC.
1300 002272  000300                CLKBRL:: .WORD  PRI06   ;INTERRUPT PRIORITY LEVEL OF THE LTC.
1301 002274  000100                CLKVEC:: .WORD  100     ;INTERRUPT VECTOR ADDRESS OF THE LTC.
1302 002276  000074                CLKHRZ:: .WORD  60.     ;INTERRUPT FREQUENCY OF THE LTC.
1303 002300  000000                TIMER1:: .WORD  0       ;HARDWARE CLOCK COUNTER #1.
1304 002302  000000                TIMER2:: .WORD  0       ;HARDWARE CLOCK COUNTER #2.
1305 002304  000170                TIMER3:: .WORD  120.    ;HARDWARE BREAK COUNTER LOCATION.
1306 002306  000170                BCOUNT:: .WORD  120.    ;BREAK COUNT VALUE IN CLOCK TICKS.
1307 002310  000021                MSTICK:: .WORD  17.     ;NUMBER OF MILLI-SECONDS PER LTC TICK.
1308 002312  000062                MSLCNT:: .WORD  62      ;LOOP COUNT (USED BY MSLOOP) TO DELAY 1 MS.
1309
1310                        ;*****************************************************************************
1311                        ;       MEMMORY MANAGEMENT VARIABLES AND FLAGS.
1312                        ;*****************************************************************************
1313 002314  177572                MMSR0:: .WORD   177572  ;ADDRESS OF MEM MGT STATUS REGISTER #0.
1314 002316  172516                MMSR3:: .WORD   172516  ;ADDRESS OF MEM MGT STATUS REGISTER #3.
1315 002320  000000                MMPRES:: .WORD  0       ;MEM MGT PRESENT FLAG (0 IF MM NOT PRESENT).
1316 002322  000000                MMENAB:: .WORD  0       ;MEM MGT ENABLED FLAG (0 IF MM NOT ENABLED).
1317
1318 002324                        PARATB::                ;BASE OF MEM MGT PAR ADDRESS TABLE.
1319 002324  172340                        PAR0A:: .WORD   172340  ;ADDRESS OF MEM MGT PAR #0.
1320 002326  172342                        PAR1A:: .WORD   172342  ;ADDRESS OF MEM MGT PAR #1.
1321 002330  172344                        PAR2A:: .WORD   172344  ;ADDRESS OF MEM MGT PAR #2.
1322 002332  172346                        PAR3A:: .WORD   172346  ;ADDRESS OF MEM MGT PAR #3.
1323 002334  172350                        PAR4A:: .WORD   172350  ;ADDRESS OF MEM MGT PAR #4.
1324 002336  172352                        PAR5A:: .WORD   172352  ;ADDRESS OF MEM MGT PAR #5.
1325 002340  172354                        PAR6A:: .WORD   172354  ;ADDRESS OF MEM MGT PAR #6.
1326 002342  172356                        PAR7A:: .WORD   172356  ;ADDRESS OF MEM MGT PAR #7.
1327 002344                        PARATE::                ;END OF PAR ADDRESS TABLE.
1328
1329 002344                        PDRATB::                ;BASE OF MEM MGT PDR ADDRESS TABLE.
1330 002344  172300                        PDR0A:: .WORD   172300  ;ADDRESS OF MEM MGT PDR #0.
1331 002346  172302                        PDR1A:: .WORD   172302  ;ADDRESS OF MEM MGT PDR #1.
1332 002350  172304                        PDR2A:: .WORD   172304  ;ADDRESS OF MEM MGT PDR #2.
1333 002352  172306                        PDR3A:: .WORD   172306  ;ADDRESS OF MEM MGT PDR #3.
1334 002354  172310                        PDR4A:: .WORD   172310  ;ADDRESS OF MEM MGT PDR #4.
1335 002356  172312                        PDR5A:: .WORD   172312  ;ADDRESS OF MEM MGT PDR #5.
1336 002360  172314                        PDR6A:: .WORD   172314  ;ADDRESS OF MEM MGT PDR #6.
1337 002362  172316                        PDR7A:: .WORD   172316  ;ADDRESS OF MEM MGT PDR #7.
1338 002364                        PDRATE::                ;END OF MEM MGT PDR ADDRESS TABLE.
1339
1340                        ;*****************************************************************************
1341                        ;       TABLE OF WORDS WITH CORRESPONDING BIT SET FOR GENERATION OF BIT MAPS.
1342                        ;*****************************************************************************
1343 002364  000001        BITTBL:: .WORD  1               ;BIT 0 SET.
```

```
1344 002366  000002                .WORD   2               ;BIT 1 SET.
1345 002370  000004                .WORD   4               ;BIT 2 SET.
1346 002372  000010                .WORD   10              ;BIT 3 SET.
1347 002374  000020                .WORD   20              ;BIT 4 SET.
1348 002376  000040                .WORD   40              ;BIT 5 SET.
1349 002400  000100                .WORD   100             ;BIT 6 SET.
1350 002402  000200                .WORD   200             ;BIT 7 SET.
1351 002404  000400                .WORD   400             ;BIT 8 SET.
1352 002406  001000                .WORD   1000            ;BIT 9 SET.
1353 002410  002000                .WORD   2000            ;BIT 10 SET.
1354 002412  004000                .WORD   4000            ;BIT 11 SET.
1355 002414  010000                .WORD   10000           ;BIT 12 SET.
1356 002416  020000                .WORD   20000           ;BIT 13 SET.
1357 002420  040000                .WORD   40000           ;BIT 14 SET.
1358 002422  100000                .WORD   100000          ;BIT 15 SET.
1359
1360                        ;*******************************************************************************
1361                        ;*      TABLE OF DUT BAUDRATES
1362                        ;*******************************************************************************
1363 002424                BRTBLB::                        ;BASE OF DUT BAUD RATE TABLE.
1364 002424  000062                .WORD   50.             ;BAUD RATE ENTRY FOR CODE 0.
1365 002426  000113                .WORD   75.             ;BAUD RATE ENTRY FOR CODE 1.
1366 002430  000156                .WORD   110.            ;BAUD RATE ENTRY FOR CODE 2.
1367 002432  000206                .WORD   134.            ;BAUD RATE ENTRY FOR CODE 3.
1368 002434  000226                .WORD   150.            ;BAUD RATE ENTRY FOR CODE 4.
1369 002436  000454                .WORD   300.            ;BAUD RATE ENTRY FOR CODE 5.
1370 002440  001130                .WORD   600.            ;BAUD RATE ENTRY FOR CODE 6.
1371 002442  002260                .WORD   1200.           ;BAUD RATE ENTRY FOR CODE 7.
1372 002444  003410                .WORD   1800.           ;BAUD RATE ENTRY FOR CODE 8.
1373 002446  003720                .WORD   2000.           ;BAUD RATE ENTRY FOR CODE 9.
1374 002450  004540                .WORD   2400.           ;BAUD RATE ENTRY FOR CODE 10.
1375 002452  011300                .WORD   4800.           ;BAUD RATE ENTRY FOR CODE 11.
1376 002454  016040                .WORD   7200.           ;BAUD RATE ENTRY FOR CODE 12.
1377 002456  022600                .WORD   9600.           ;BAUD RATE ENTRY FOR CODE 13.
1378 002460  045400                .WORD   19200.          ;BAUD RATE ENTRY FOR CODE 14.
1379 002462  113000                .WORD   38400.          ;BAUD RATE ENTRY FOR CODE 15.
1380 002464                BRTBLE::                        ;LABEL AFTER END OF DUT BAUDRATE TABLE.
1381                        ;*******************************************************************************
1382                        ;*      GP  SAVE AREAS ZERO AND ONE.
1383                        ;*******************************************************************************
1384 002464                GPRS0B::                        ;BASE OF GPR SAVE AREA NUMBER ZERO.
1385 002464  000000                .WORD   0               ;WORD 1. STORAGE FOR R1.
1386 002466  000000                .WORD   0               ;WORD 2. STORAGE FOR R2.
1387 002470  000000                .WORD   0               ;WORD 3. STORAGE FOR R3.
1388 002472  000000                .WORD   0               ;WORD 4. STORAGE FOR R4.
1389 002474  000000                .WORD   0               ;WORD 5. STORAGE FOR R5.
1390                        ;*******************************************************************************
1391                        ;*      TRANSMISSION AND RECEPTION VARIABLES, POINTERS, AND FLAGS.
1392                        ;*******************************************************************************
1393 002476  000000        CHRTOT:: .WORD  0               ;TOTAL RECEIVED CHARACTER COUNTER.
1394 002500  000000        ERSMRF:: .WORD  0               ;"PRINT ERROR SUMMARY" FLAGS.
1395 002502  000000        TXDONF:: .WORD  0               ;TRANSMISSION DONE FLAGS.
1396 002504  000000        RXDONF:: .WORD  0               ;RECEPTION DONE FLAGS.
1397 002506  000000        TXDBLF:: .WORD  0               ;"TX HAS BEEN DISABLED" FLAG.
1398                        ;*******************************************************************************
1399                        ;       STORAGE AREA FOR THE BMP CODE QUEUE.
1400                        ;*******************************************************************************
```

```
1401 002510  000000                BMPCQP::        .WORD   0       ;POINTER USED TO ACCESS THE NEXT CELL IN QUE.
1402 002512                        BMPCQB::        .BLKW   64.     ;STORAGE FOR 32 CELLS, TEST# PLUS BMP CODE.
1403 002712                        BMPCQE::                        ;LAST ADDRESS PLUS 2 OF THE BMP CODE QUEUE.
1404                               ;*******************************************************************************
1405                               ;*      RECEIVE BUFFER AND ASSOCIATED VARIABLES.
1406                               ;*******************************************************************************
1407 002712  000000                RXBOPT:: .WORD  0               ;RX BUFFER OUTPUT POINTER.
1408 002714  000000                RXBIPT:: .WORD  0               ;RX BUFFER INPUT POINTER.
1409 002716  000000                RXBCNT:: .WORD  0               ;COUNT OF NUMBER OF CHARS IN RX BUFFER.
1410 002720                        RXBSTA::                        ;LABEL AT BEGINNING OF THE RX BUFFER.
1411 002720                                .BLKW   RXBFUL          ;LEAVE ENOUGH ROOM FOR A FULL BUFFER.
1412 003120  000000                RXBEND:: .WORD  0               ;LABEL AFTER END OF RX BUFFER.
1413                               ;*******************************************************************************
1414                               ;*      TX/RX CONTROL BLOCK.
1415                               ;*******************************************************************************
1416 003122                        CBB::                           ;BASE OF TX/RX CONTROL BLOCK.
1417 003122  000000                CBLPRA:: .WORD  0               ;LINE PARAMETER REGISTER CONTENTS.
1418 003124  000000                CBLNCA:: .WORD  0               ;LINE CONTROL REGISTER CONTENTS.
1419 003126  000000                CBDPAA:: .WORD  0               ;START ADDRESS OF DATA PATTERN.
1420 003130  000000                CBDPLA:: .WORD  0               ;LENGTH OF DATA PATTERN.
1421 003132  000000                CBDPNA:: .WORD  0               ;NUMBER OF REPEAT TRANSMISSIONS OF THE DATA PATTERN.
1422 003134  000000                CBMAPA:: .WORD  0               ;BIT MAP OF LINES TO INITIALISE.
1423 003136  000000                CBLPBA:: .WORD  0               ;LOOPBACK MODE (AS IN LOPBCK).
1424 003140  000000                CBOFSA:: .WORD  0               ;AMOUNT OF OFFSET BETWEEN EACH TX START.
1425                               ;*******************************************************************************
1426                               ;*      TRANSMISSION AND RECEPTION TABLES OF POINTERS AND COUNTERS.
1427                               ;*******************************************************************************
1428 003142                        DPENDB:: .BLKW  16.             ;TABLE OF END ADDRESSES OF DATA PATTERNS.
1429 003202                        DPLENB:: .BLKW  16.             ;TABLE OF LENGTH OF DATA PATTERNS FOR LINES.
1430 003242                        EXCNTB:: .BLKW  16.             ;EXTRA RECEIVED CHARACTER COUNTERS TABLE.
1431 003302                        ERCNTB:: .BLKW  16.             ;CHARACTER RECEIVE ERROR COUNTERS TABLE.
1432 003342                        TXPTRB:: .BLKW  16.             ;TRANSMISSION DATA POINTERS TABLE.
1433 003402                        RXPTRB:: .BLKW  16.             ;RECEPTION DATA POINTERS TABLE.
1434 003442                        CHCNTB:: .BLKW  16.             ;NUMBER OF CHARACTERS TO BE TXED AND RXED.
1435 003502                        TXCNTB:: .BLKW  16.             ;TRANSMISSION CHARACTER COUNTERS TABLE.
1436 003542                        RXCNTB:: .BLKW  16.             ;RECEPTION CHARACTER COUNTERS TABLE.
1437                               ;*******************************************************************************
1438                               ;       GENERAL TABLE AND BUFFER AREA--513 WORDS.
1439                               ;*******************************************************************************
1440 003602                        BUFBAS::                        ;BASE OF MEMORY BUFFER.
1441 003602                        ERLTBL::        .BLKW   128.    ;FIRST HALF OF GENERAL TABLE OR BUFFER.
1442 004202                        BUFMID::        .BLKW   64.     ;SECOND HALF OF GENERAL TABLE OR BUFFER.
1443 004402                        BUF3QT::        .BLKW   64.     ;LAST QUARTER OF THE BUFFER AREA.
1444 004602                        BUFEND::                        ;END OF GENERAL PURPOSE MEMORY BUFFER.
1445 004602                        ENDETB::        .BLKW   16.     ;BUFFER OVERFLOW SPACE.
1446                               ;*******************************************************************************
1447                               ;       TABLE OF DATA PATTERN RESYNC QUEUES.
1448                               ;*******************************************************************************
1449 004642                        DPRSQB::                        ;DATA PATTERN RESYNC QUEUES TABLE BASE.
1450 004642                                .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 0.
1451 004652                                .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 1.
1452 004662                                .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 2.
1453 004672                                .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 3.
1454 004702                                .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 4.
1455 004712                                .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 5.
1456 004722                                .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 6.
1457 004732                                .BLKW   4               ;DATA PATTERN RESYNC QUEUE FOR LINE 7.
```

```
1458 004742                                     .BLKW   4       ;DATA PATTERN RESYNC QUEUE FOR LINE 8.
1459 004752                                     .BLKW   4       ;DATA PATTERN RESYNC QUEUE FOR LINE 9.
1460 004762                                     .BLKW   4       ;DATA PATTERN RESYNC QUEUE FOR LINE 10.
1461 004772                                     .BLKW   4       ;DATA PATTERN RESYNC QUEUE FOR LINE 11.
1462 005002                                     .BLKW   4       ;DATA PATTERN RESYNC QUEUE FOR LINE 12.
1463 005012                                     .BLKW   4       ;DATA PATTERN RESYNC QUEUE FOR LINE 13.
1464 005022                                     .BLKW   4       ;DATA PATTERN RESYNC QUEUE FOR LINE 14.
1465 005032                                     .BLKW   4       ;DATA PATTERN RESYNC QUEUE FOR LINE 15.
1466 005042                             DPRSQE::                ;END OF DATA PATTERN RESYNC QUEUES TABLE.
1467                                     ;*******************************************************************************
1468                                     ;       SINGLE CHARACTER MODE LPR FIELD TABLES.
1469                                     ;*******************************************************************************
1470 005042                             SCBCTB::                ;BASE OF NUMBER OF BITS PER CHAR FIELDS TABLE.
1471 005042  000000                             .WORD   0               ;5 BITS/CHAR LPR FIELD.
1472 005044  000010                             .WORD   10              ;6 BITS/CHAR LPR FIELD.
1473 005046  000020                             .WORD   20              ;7 BITS/CHAR LPR FIELD.
1474 005050  000030                             .WORD   30              ;8 BITS/CHAR LPR FIELD.
1475 005052                             SCBCTE::                ;END OF NUMBER OF BITS/CHAR FIELDS TABLE.
1476 005052                             SCBRTB::                ;BASE OF BAUDRATE FIELDS TABLE.
1477 005052  000000                             .WORD   0               ;50 BAUD LPR FIELDS.
1478 005054  073400                             .WORD   73400           ;1.2K BAUD LPR FIELDS.
1479 005056  177400                             .WORD   177400          ;38.4K BAUD LPR FIELDS.
1480 005060                             SCBRTE::                ;END OF BAUDRATE FIELDS TABLE.
1481 005060                             SCNSTB::                ;BASE OF NUMBER OF STOP BITS FIELDS TABLE.
1482 005060  000000                             .WORD   0               ;1 STOP BIT LPR FIELD.
1483 005062  000200                             .WORD   200             ;2 STOP BITS LPR FIELD.
1484 005064                             SCNSTE::                ;END OF BAUDRATE FIELDS TABLE.
1485 005064                             SCTPTB::                ;BASE OF TYPE OF PARITY FIELDS TABLE.
1486 005064  000000                             .WORD   0               ;NO PARITY LPR FIELD.
1487 005066  000040                             .WORD   40              ;ODD PARITY LPR FIELD.
1488 005070  000140                             .WORD   140             ;EVEN PARITY LPR FIELD.
1489 005072                             SCTPTE::                ;END OF TYPE OF PARITY FIELDS TABLE.
1490
1491                                     ;*******************************************************************************
1492                                     ;       DMA MODE LPR FIELD TABLES.
1493                                     ;    SET UP WITH SPECIFIED BAUDRATES, 1 STOP BIT, ODD PARITY, 8 BITS/CHAR.
1494                                     ;*******************************************************************************
1495 005072                             DLPRTB::                ;BASE OF DMA TEST LPR FIELDS TABLE.
1496 005072  156470                             .WORD   156470          ;9.6K BAUD.
1497 005074  167070                             .WORD   167070          ;19.2K BAUD.
1498 005076  177470                             .WORD   177470          ;38.4K BAUD.
1499 005100                             DLPRTE::                ;END OF DMA TEST LPR FIELDS TABLE.
1500                                     ;*******************************************************************************
1501                                     ;       SPLIT SPEED LPR PARAMETER TABLE.
1502                                     ;*******************************************************************************
1503 005100                             SPLPRB::                ;BASE OF SPLIT SPEED LPR TABLE.
1504 005100  170070                             .WORD   170070          ;TX: 38.4K, RX: 50 BAUD, 1 STOP ODD PAR 8 BITS.
1505 005102  007470                             .WORD   7470            ;TX: 50, RX: 38.4K BAUD, 1 STOP ODD PAR 8 BITS.
1506 005104  000001                             .WORD   1               ;NUMBER OF REPEAT TRANSMISSIONS AT 50 BAUD.
1507 005106  000120                             .WORD   80.             ;NUMBER OF REPEAT TRANSMISSIONS AT 38.4K BAUD.
1508 005110  070470                             .WORD   70470           ;TX: 1200, RX: 75 BAUD, 1 STOP ODD PAR 8 BITS.
1509 005112  013470                             .WORD   13470           ;TX: 75, RX: 1200 BAUD, 1 STOP ODD PAR 8 BITS.
1510 005114  000001                             .WORD   1               ;NUMBER OF REPEAT TRANSMISSIONS AT 75 BAUD.
1511 005116  000016                             .WORD   16              ;NUMBER OF REPEAT TRANSMISSIONS AT 1200 BAUD.
1512 005120  115070                             .WORD   115070          ;TX: 2000, RX:2400 BAUD, 1 STOP ODD PAR 8 BITS.
1513 005122  124470                             .WORD   124470          ;TX: 2400, RX:2000 BAUD, 1 STOP ODD PAR 8 BITS.
1514 005124  000001                             .WORD   1               ;NUMBER OF REPEAT TRANSMISSIONS AT 2400 BAUD.
```

```
1515 005126  000002                    .WORD   2               ;NUMBER OF REPEAT TRANSMISSIONS AT 2000 BAUD.
1516 005130                    SPLPRE::                        ;END OF SPLIT SPEED LPR TABLE.
1517                           ;*******************************************************************************
1518                           ;       SINGLE CHARACTER DATA PATTERN TABLE.
1519                           ;*******************************************************************************
1520 005130      000           SDPBAS::.BYTE   0       ;START OF SINGLE CHARACTER DATA PATTERN TABLE.
1521 005131      001                   .BYTE   1
1522 005132      010                   .BYTE   10
1523 005133      017                   .BYTE   17
1524 005134      063                   .BYTE   63
1525 005135      074                   .BYTE   74
1526 005136      125                   .BYTE   125
1527 005137      177                   .BYTE   177
1528 005140      200                   .BYTE   200
1529 005141      252                   .BYTE   252
1530 005142      303                   .BYTE   303
1531 005143      314                   .BYTE   314
1532 005144      360                   .BYTE   360
1533 005145      367                   .BYTE   367
1534 005146      376                   .BYTE   376
1535 005147      377                   .BYTE   377
1536 005150                    SDPEND::                        ;END OF SINGLE CHARACTER DATA PATTERN TABLE.
1537 005150      000                   .BYTE   0       ;START OF FIRST SHORT DATA PATTERN OVERFLOW AREA.
1538 005151      001                   .BYTE   1
1539 005152      010                   .BYTE   10
1540 005153      017                   .BYTE   17
1541
1542                           ;*******************************************************************************
1543                           ;       SINGLE CHARACTER DATA PATTERN TABLE NUMBER TWO.
1544                           ;*******************************************************************************
1545 005154      125           SDP2B:: .BYTE   125     ;START OF SECOND SHORT DATA PATTERN.
1546 005155      252                   .BYTE   252
1547 005156      124                   .BYTE   124
1548 005157      253                   .BYTE   253
1549 005160      122                   .BYTE   122
1550 005161      255                   .BYTE   255
1551 005162      112                   .BYTE   112
1552 005163      265                   .BYTE   265
1553 005164      052                   .BYTE   52
1554 005165      325                   .BYTE   325
1555 005166      152                   .BYTE   152
1556 005167      225                   .BYTE   225
1557 005170      132                   .BYTE   132
1558 005171      245                   .BYTE   245
1559 005172      126                   .BYTE   126
1560 005173      251                   .BYTE   251
1561 005174                    SDP2E::                         ;END OF SECOND SHORT DATA PATTERN.
1562 005174      125                   .BYTE   125     ;START OF SECOND SHORT DATA PATTERN OVERFLOW AREA.
1563 005175      252                   .BYTE   252
1564 005176      124                   .BYTE   124
1565 005177      253                   .BYTE   253
1566 005200      122                   .BYTE   122
1567 005201      255                   .BYTE   255
1568 005202      112                   .BYTE   112
1569 005203      265                   .BYTE   265
1570 005204      052                   .BYTE   52
1571 005205      325                   .BYTE   325
```

```
                        .BYTE   152
                        .BYTE   225
                        .BYTE   132
                        .BYTE   245
                        .BYTE   126
                        .BYTE   251
                ;*****************************************************************************
                ;       SINGLE CHARACTER SAFE PROPORTIONAL DELAY TABLE.
                ;*****************************************************************************
                PROTBL::.BYTE   250.            ;DELAY IN MILLI SECONDS AT 50 BAUD
                        .BYTE   170.            ;DELAY IN MILLI SECONDS AT 75 BAUD
                        .BYTE   119.            ;DELAY IN MILLI SECONDS AT 110 BAUD
                        .BYTE   99.             ;DELAY IN MILLI SECONDS AT 134.5 BAUD
                        .BYTE   90.             ;DELAY IN MILLI SECONDS AT 150 BAUD
                        .BYTE   50.             ;DELAY IN MILLI SECONDS AT 300 BAUD
                        .BYTE   30.             ;DELAY IN MILLI SECONDS AT 600 BAUD
                        .BYTE   20.             ;DELAY IN MILLI SECONDS AT 1200 BAUD
                        .BYTE   17.             ;DELAY IN MILLI SECONDS AT 1800 BAUD
                        .BYTE   16.             ;DELAY IN MILLI SECONDS AT 2000 BAUD
                        .BYTE   15.             ;DELAY IN MILLI SECONDS AT 2400 BAUD
                        .BYTE   13.             ;DELAY IN MILLI SECONDS AT 4800 BAUD
                        .BYTE   12.             ;DELAY IN MILLI SECONDS AT 7200 BAUD
                        .BYTE   12.             ;DELAY IN MILLI SECONDS AT 9600 BAUD
                        .BYTE   11.             ;DELAY IN MILLI SECONDS AT 19200 BAUD
                        .BYTE   10.             ;DELAY IN MILLI SECONDS AT 38400 BAUD
                        .EVEN
                ;*****************************************************************************
                ;* TABLE FOR STORAGE OF RX/TX LINE NUMBER ASSOCIATIONS.
                ;* THE ASSOCIATIONS ARE STORED AS LINE NUMBER TIMES 2 FOR USE AS OFFSETS
                ;*    WHEN ACCESSING A TABLE OF WORDS.
                ;*    NOTE: DO NOT WRITE A NON-ZERO VALUE INTO THE UPPER BYTE OF ANY ENTRY.
                ;*****************************************************************************
                TXRXLB::                        ;BASE OF TX/RX LINE NUMBER ASSOCIATION TABLE.
 005234                 .WORD   0               ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
 005234  000000         .WORD   2.              ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
 005236  000002         .WORD   4.              ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
 005240  000004         .WORD   6.              ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
 005242  000006         .WORD   8.              ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
 005244  000010         .WORD   10.             ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
 005246  000012         .WORD   12.             ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
 005250  000014         .WORD   14.             ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
 005252  000016         .WORD   16.             ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
 005254  000020         .WORD   18.             ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
 005256  000022         .WORD   20.             ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
 005260  000024         .WORD   22.             ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
 005262  000026         .WORD   24.             ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
 005264  000030         .WORD   26.             ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
 005266  000032         .WORD   28.             ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
 005270  000034         .WORD   30.             ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
 005272  000036 TXRXLE::                        ;END OF TX/RX LINE NUMBER ASSOCIATION TABLE.
 005274                 .EVEN                   ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
                ;*****************************************************************************
                ;* TABLE OF TX/RX LINE NUMBER ASSOCIATIONS IN STAGGERED LOOPBACK.
                ;* THE ASSOCIATIONS ARE STORED AS LINE NUMBER TIMES 2 FOR USE AS OFFSETS
                ;*    WHEN ACCESSING A TABLE OF WORDS.
                ;* THIS IS A TABLE OF DATA FOR READING ONLY.  USE TO LOAD THE ABOVE TABLE.
                ;*    NOTE: MUST CONVERT FROM BYTES TO WORDS WHEN LOADING ABOVE TABLE.
```

```
1629                                    ;*******************************************************************************
1630 005274                             STGTRB::                        ;BASE OF STAGGERED TX/RX LINE NUMBER TABLE.
1631 005274       004                           .BYTE   4.              ;TX/RX LINE OFFSET FOR RX/TX LINE 0.
1632 005275       006                           .BYTE   6.              ;TX/RX LINE OFFSET FOR RX/TX LINE 1.
1633 005276       000                           .BYTE   0               ;TX/RX LINE OFFSET FOR RX/TX LINE 2.
1634 005277       002                           .BYTE   2.              ;TX/RX LINE OFFSET FOR RX/TX LINE 3.
1635 005300       014                           .BYTE   12.             ;TX/RX LINE OFFSET FOR RX/TX LINE 4.
1636 005301       016                           .BYTE   14.             ;TX/RX LINE OFFSET FOR RX/TX LINE 5.
1637 005302       010                           .BYTE   8.              ;TX/RX LINE OFFSET FOR RX/TX LINE 6.
1638 005303       012                           .BYTE   10.             ;TX/RX LINE OFFSET FOR RX/TX LINE 7.
1639 005304       024                           .BYTE   20.             ;TX/RX LINE OFFSET FOR RX/TX LINE 8.
1640 005305       026                           .BYTE   22.             ;TX/RX LINE OFFSET FOR RX/TX LINE 9.
1641 005306       020                           .BYTE   16.             ;TX/RX LINE OFFSET FOR RX/TX LINE 10.
1642 005307       022                           .BYTE   18.             ;TX/RX LINE OFFSET FOR RX/TX LINE 11.
1643 005310       034                           .BYTE   28.             ;TX/RX LINE OFFSET FOR RX/TX LINE 12.
1644 005311       036                           .BYTE   30.             ;TX/RX LINE OFFSET FOR RX/TX LINE 13.
1645 005312       030                           .BYTE   24.             ;TX/RX LINE OFFSET FOR RX/TX LINE 14.
1646 005313       032                           .BYTE   26.             ;TX/RX LINE OFFSET FOR RX/TX LINE 15.
1647                                            .EVEN                   ;GUARANTEE THAT NEXT TABLE IS ON WORD BOUNDARY.
1660 005314                                     ERRTBL
     005314                                                                             L$ERRTBL::
     005314  000000                     ERRTYP::        .WORD   0
     005316  000000                     ERRNBR::        .WORD   0
     005320  000000                     ERRMSG::        .WORD   0
     005322  000000                     ERRBLK::        .WORD   0
1661
1662                                            .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).
                        ;*      PREGOS - 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 PREGOS 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 SUBROUTINE WHICH CALLED PREGOS.
                        ;*
                        ;*      EACH LEVEL OF SUBROUTINE CALLING USES 8 WORDS OF STACK OVERHEAD.
                        ;*      THE SAVE AND PASS MACROS CAN ALSO BE USED IN "STRAIGHT LINE CODE"
                        ;*      TO SAVE AND RESTORE THE GPR VALUES.  IN ANY CASE, AFTER THE
                        ;*      ISSUING OF A PASS CALL THE GPRS WILL BE RESTORED TO THE VALUES
                        ;*      THEY HAD PRIOR TO THE LAST SAVE CALL (EXCEPT FOR THE 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                          ROSLOT==        2       ;OFFSET FOR R0.
```

```
1718                    .SBTTL  GLOBAL MACRO DEFINITION         - SAVE -
1719                    ;**************************************************************************
1720                    ;*      THIS MACRO IS USED AT THE BEGINNING OF A SUBROUTINE TO SAVE THE
1721                    ;*      CONTENTS OF THE GPRS R0 THRU R5.
1722                    ;*
1723                    ;* INPUTS:      SP - UNCHANGED SINCE SUBROUTINE WAS ENTERED
1724                    ;*              R5SLOT - OFFSET TO STACK SLOT FOR R5 (EQUATED TO 14 OCTAL)
1725                    ;*
1726                    ;* OUTPUTS:     GPR SAVE AREA ON THE STACK IS LOADED WITH THE CONTENTS OF GPRS
1727                    ;*              TOP OF STACK - LOADED WITH THE RETURN ADDRESS INTO PREG05
1728                    ;*
1729                    ;* CALLING SEQUENCE:    SAVE
1730                    ;*
1731                    ;* COMMENTS:    NO ARGUMENTS ARE ALLOWED.
1732                    ;*              THE PASS MACRO SHOULD BE CALLED TO RESTORE THE GPR VALUES.
1733                    ;*
1734                    ;* SUBORDINATE ROUTINES CALLED: PREG05.
1735                    ;**************************************************************************
1736
1737                            .MACRO  SAVE
1738                            .LIST
1739                                    JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
1740                            .NLIST
1741                            .ENDM   SAVE
```

```
1743                    .SBTTL  GLOBAL MACRO DEFINITION          - PASS -
1744                    ;*************************************************************************
1745                    ;*      THIS MACRO IS USED IN CONJUNCTION WITH THE SAVE MACRO.  IT IS
1746                    ;*      CALLED AT END OF A SUBROUTINE TO PASS PARAMETERS IN GPRS BACK TO THE
1747                    ;*      CALLING ROUTINE BY ALTERING THE GPR SAVE AREA ON THE STACK AND THEN
1748                    ;*      RETURNING TO PREGO5 TO RESTORE THE GPRS TO THEIR SAVED VALUES.
1749                    ;*
1750                    ;* INPUTS:       ONLY ALLOWED ARGUMENTS ARE "R0" THRU "R5".
1751                    ;*               R0SLOT THRU R5SLOT MUST BE EQUATED TO THEIR RESPECTIVE GPR SAVE
1752                    ;*                       SLOT OFFSETS BEFORE CALLING THIS MACRO.
1753                    ;*
1754                    ;* OUTPUTS:      THE GPR VALUES ARE PUT IN THEIR RESPECTIVE SLOTS ON THE STACK.
1755                    ;*
1756                    ;* CALLING SEQUENCE:     PASS    R0,R1,...
1757                    ;*
1758                    ;* COMMENTS:     ANY COMBINATION OF GPR ARGUMENTS MAY BE LISTED IN ANY ORDER.
1759                    ;*               FOR EXAMPLE, THE FOLLOWING ARE LEGAL:
1760                    ;*                       PASS    R1
1761                    ;*                       PASS    R4,R0,R2
1762                    ;*               THE GPRS LISTED AS ARGUMENTS WILL BE PASSED INTACT TO THE
1763                    ;*               CALLING ROUTINE, ALL OTHER GPRS WILL BE RESTORED.
1764                    ;*               THE SP MUST BE AT ITS ORIGINAL VALUE WHEN PASS IS CALLED.
1765                    ;*
1766                    ;*               THE MACRO CALL
1767                    ;*                       PASS    R0,R3
1768                    ;*               EXPANDS INTO THE FOLLOWING ASSEMBLY CODE:
1769                    ;*                       MOV     R0,R0SLOT(SP)           ;PUT R0 IN STACK SLOT.
1770                    ;*                       MOV     R3,R3SLOT(SP)           ;PUT R3 IN STACK SLOT.
1771                    ;*                       JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
1772                    ;*               IN THIS EXAMPLE GPRS R1, R2, R4, AND R5 WILL BE RESTORED TO
1773                    ;*               THEIR VALUES CONTAINED IN THE STACK FRAME AND R0 AND R3
1774                    ;*               WILL BE LEFT AT THEIR VALUES PRIOR TO THIS PASS CALL.
1775                    ;*
1776                    ;* SUBORDINATE ROUTINES CALLED: (PREGRT - LABEL WITHIN PREGO5, VALUE ON STACK.)
1777                    ;*************************************************************************
1778
1779                            .MACRO  PASS    A,B,C,D,E,F
1780                            .IRP    X,<A,B,C,D,E,F>
1781                            .IF     NB,X
1782                            .LIST
1783                                    MOV     X,X'SLOT(SP)            ;PUT X IN STACK SLOT.
1784                            .NLIST
1785                            .ENDC
1786                            .ENDM
1787                            .LIST
1788                                    JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
1789                            .NLIST
1790                            .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,ROSLOT(SP)        ]
                        ;*             MAKE SURE THE SP IS AT ITS ORIGINAL VALUE WHEN YOU DO THIS.
                        ;*
                        ;*SUBORDINATE ROUTINES CALLED:  NONE.
                        ;*****************************************************************************

005324                  PPEG05:                 ;R5 HAS BEEN LOADED ON THE STACK BY THE SUBROUTINE CALL
005324  010446                  MOV     R4,-(SP)        ;SAVE R4
005326  010346                  MOV     R3,-(SP)        ;SAVE R3
005330  010246                  MOV     R2,-(SP)        ;SAVE R2
005332  010146                  MOV     R1,-(SP)        ;SAVE R1
005334  010046                  MOV     R0,-(SP)        ;SAVE R0
005336  010546                  MOV     R5,-(SP)        ;PUSH RETURN PC ON TOP OF STACK
005340  016605  000014          MOV     R5SLOT(SP),R5   ;RESTORE R5 TO VALUE IT HAD BEFORE CALLS
                        
005344  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.
                                ;---

005346  012605          PREGRT:: MOV    (SP)+,R5        ;PUT RETURN PC IN R5.
005350  012600                   MOV    (SP)+,R0        ;RESTORE R0.
005352  012601                   MOV    (SP)+,R1        ;RESTORE R1.
005354  012602                   MOV    (SP)+,R2        ;RESTORE R2.
005356  012603                   MOV    (SP)+,R3        ;RESTORE R3.
005360  012604                   MOV    (SP)+,R4        ;RESTORE R4.

005362  000205                   RTS    R5              ;RETURN TO THE SUBROUTINE WHICH CALLED PREG05.
                                                          ;RESTORING R5 IN THE PROCESS.
```

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


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

                                ;
                                ; NAMES OF DEVICES SUPPORTED BY PROGRAM
                                ;
005364                                  DEVTYP  <DHU-11>
005364                                                                  L$DVTYP::
005364  104     110     125                                                     .ASCIZ  /DHU-11/
005367  055     061     061
005372  000
                                                                                .EVEN

                                ; TEST DESCRIPTION
                                ;
005374                                  DESCRIPT        <DHU-11 FUNC TST PART3>
005374                                                                  L$DESC::
005374  104     110     125                                                     .ASCIZ  /DHU 11 FUNC TST PAR
T3/
005377  055     061     061
005402  040     106     125
005405  116     103     040
005410  124     123     124
005413  040     120     101
005416  122     124     063
005421  000
                                                                                .EVEN
                                .EVEN
                                ;
                                ;
```

```
1886
1887                    ;*****************************************************************************
*
1888                    ;
1889                    ;               FVTA.FMT
1890                    ;
1891                    ;*****************************************************************************
1892
1893
1894
1895                    ;
1896                    ; FORMAT STATEMENTS USED IN PRINT CALLS
1897                    ;
1898
1909
1910
```

```

                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                                ;
                                ;               FVTC.MSG
                                ;
                                ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%


        .NLIST BIN
        .SBTTL  GLOBAL MESSAGE AREA
                ; ********** FORMAT STATEMENTS **********
005422  MFUNIT:: .ASCIZ /%N%A TESTING UNIT :%D4%N/
005453  EF0503:: .ASCIZ /%T%N/
005460  EF1601:: .ASCIZ /%A  %T%A, TEST 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  EF9001:: .ASCIZ /%A  UNEXPECTED %T%A FOUND IN RECEIVE CHAR FIFO:%N/
006317  EF9002:: .ASCIZ /%A    CODE IS ASSOCIATED WITH LINE: %D2%N/
006371  EF9003:: .ASCIZ /%A      CODE IS: %O3%N/
006420  EF9004:: .ASCIZ /%A    %T%A VALUE: %O3%N/
006450  EF9005:: .ASCIZ /%A    %T%A VALUE: NONE%N/
006501  EF9006:: .ASCIZ /%A  %T%A %D2%N/
006520  EF9007:: .ASCIZ /%A  CHARACTER RECEIVED WITH ERROR FLAG(S) SET ON LINE %D2%N/
006614  EF9008:: .ASCIZ /%A    CHARACTER READ AS: %O3%N/
006653  EF9009:: .ASCIZ /%A      %T%A ERROR FLAG SET.%N/
006712  EF9010:: .ASCIZ /%A    NUMBER OF ERRORS DETECTED ON LINE %D2%A IS %D5%N/
007001  EF9012:: .ASCII /%A LINE%D2%A ONLY %T%D5%A BYTES OF%D5%A BYTE/
007055           .ASCIZ / DATA PAT'N TX'D FROM LINE%D2%N/
007115  EF9013:: .ASCIZ /%A  DATA PATTERN NOT COMPLETELY %T%N/
007162  EF9019:: .ASCIZ /%A  %T%A %O6%N/
007201  EF9020:: .ASCIZ /%A    TOO FEW TX.ACTIONS GENERATED ON LINE %D2%N/
007262  EF9101:: .ASCIZ /%N/
007265  EF9103:: .ASCIZ /%A      ERROR CONDITION ON LINE %D2%N/
007333  EF9301:: .ASCIZ /%A %T%D2%A,  BMP CODE REPORTED :%O3%N/
007401  EF9302:: .ASCIZ /%A  OVERFLOW OCCURRED (MORE THAN 31 BMP CODES FOUND IN QUEUE)%N/
007501  UBRFMT:: .ASCIZ /%D5%A IS NOT A SUPPORTED BAUDRATE, ENTER ANOTHER OR CTRL C.%N/
007577  MSFMT1:: .ASCIZ /%AMODEM STATUS SIGNAL REPORT:%N/
007637  MSFMT2:: .ASCIZ /%A  LINE #%D2%A:  DSR=%B1%A, RI=%B1%A, DCD=%B1%A, CTS=%B1%N/
007733  EDPFMT:: .ASCII /%AMODEM LOOPBACK TEST STATUS REPORT:  /
010001           .ASCIZ /PATTERN #%D5%A (D) COMPLETED.%N/

                ;************* MESSAGE AREA ************
010041  EM0103:: .ASCIZ /DEVICE REGISTER ACCESS ERRORS/
010077  EM0509:: .ASCIZ /SET/
010103  EM1601:: .ASCIZ /TIMEOUT OCCURRED WAITING FOR MASTER RESET TO CLEAR/
010166  EM4401:: .ASCIZ /DMA ADDRESS TEST FAILED/
010216  EM4402:: .ASCIZ /NO SUITABLE ADDR FOUND,TEST ABANDONED/
010264  EM4403:: .ASCIZ /**HOST FAILURE**;WRITE FAILED TO AN ADDR WHICH HAD BEEN READ/
010361  EM4404:: .ASCIZ /NO ACTIVE LINES,TEST ABANDONED/
010420  EM4405:: .ASCIZ /DMA_START BIT FOUND SET BEFORE DMA INITIATED,TEST ABANDONED/
010514  EM4406:: .ASCIZ /TIME-OUT OCCCURED WAITING FOR DMA TO FINISH/
010570  EM4407:: .ASCIZ /TOO FEW CHARACTERS FOUND IN THE RXFIFO,DMA FAILED/
```

```
1976 010652  EM4408:: .ASCIZ /TOO MANY BMP CODES FOUND IN RXFIFO/
1977 010715  EM4409:: .ASCIZ /BAD BITS BETWEEN BITS 0 AND /
1978 010752  EM4410:: .ASCIZ /RXFIFO FAILED TO PURGE/
1979 011001  EM4411:: .ASCIZ /**HOST FAILURE**WRITE ATTEMPT FAILED/
1980 011046  EM5303:: .ASCIZ /BMP CODE FOUND IN FIFO, TEST INVAILDATED/
1981 011117  EM6201:: .ASCIZ /FRAMING ERROR TEST FAILED/
1982 011151  EM6202:: .ASCIZ /CLEAR /
1983 011160  EM6301:: .ASCIZ /PARITY ERROR TEST FAILED/
1984 011211  EM8901:: .ASCIZ /MODEM LOOPBACK TEST /
1985 011236  EM9003:: .ASCIZ /MODEM STATUS CODE/
1986 011260  EM9004:: .ASCIZ /SELFTEST CODE/
1987 011276  EM9006:: .ASCIZ /CHARACTER RECEIVED ON INACTIVE LINE, LINE:/
1988 011351  EM9007:: .ASCIZ /UNEXPECTED CHAR RECEIVED AFTER RX COMPLETE ON LINE/
1989 011434  EM9008:: .ASCIZ /RECEIVED CHAR MISCOMPARE AGAINST TX DATA ON LINE/
1990 011515  EM9009:: .ASCIZ /EXPECTED OR CORRECT/
1991 011541  EM9010:: .ASCIZ /ACTUAL OR MEASURED /
1992 011565  EM9011:: .ASCIZ /OVERRUN/
1993 011575  EM9012:: .ASCIZ /FRAMING/
1994 011605  EM9013:: .ASCIZ /PARITY/
1995 011614  EM9014:: .ASCIZ /SUMMARY REPORTS FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:/
1996 011710  EM9015:: .ASCIZ /TRANSMITTED/
1997 011724  EM9016:: .ASCIZ /RECV'D/
1998 011733  EM9017:: .ASCII /  FIFO WILL NOT PURGE (DATA.VALID STUCK SET),/
1999 012010          .ASCIZ / REMAINDER OF TEST SKIPPED./
2000 012044  EM9025 : .ASCIZ /MORE THAN TWICE THE EXPECTED NUMBER OF CHARACTERS RECEIVED./
2001 012140  EM9026:: .ASCIZ /     LPR CONTENTS: /
2002 012164  EM9027:: .ASCIZ /EXTRA CHAR RECEIVED WITHIN DATA PATTERN ON LINE/
2003 012244  EM9028:: .ASCIZ /SINGLE CHAR MISSING FROM RECEIVED DATA ON LINE/
2004 012323  EM9030:: .ASCIZ /%A  (NO TX COMPLETION INTERRUPTS RECEIVED)%N/
2005 012400  EM9101:: .ASCIZ /DMA TRANSMISSION MODE TEST FAILED/
2006 012442  EM9102:: .ASCIZ /DMA_START BIT SET AFTER RESET OR TX.ACTION ON LINE(S):/
2007 012531  EM9104:: .ASCIZ /  UNEXPECTED DATA FOUND IN FIFO FROM LINE: /
2008 012605  EM9201:: .ASCIZ /SPLIT SPEED TEST FAILED/
2009 012635  EM9301:: .ASCIZ /BMP CODES WERE REPORTED DURING THIS DIAGNOSTIC/
2010 012714  EM9302:: .ASCIZ /BMP CODE FOUND IN TEST /
2011 012744  EM9303:: .ASCIZ /THE LAST BMP CODE WAS FOUND IN TEST /
2012 013011  EM9304:: .ASCIZ /UNEXPECTED BMP CODES FOUND DURING THIS PASS/
2013 013065  EM9401:: .ASCIZ /KEYBOARD ECHO (DHU REMOTE LOOPBACK) TEST /
2014
2015 013137  BDRMSG:: .ASCIZ /MODEM BAUDRATE IN BPS:/
2016 013166  EMLMSG:: .ASCIZ /TYPE <CR> WHEN MODEM LINK ESTABLISHED:/
2017 013235  EXTMSG:: .ASCIZ /EXIT THE TEST (N = LOOP BACK TO SEND MORE DATA):/
2018 013316  NDPMSG:: .ASCII /NUMBER OF 256 BYTE PATTERNS TO SEND ON EACH SELECTED LINE/
2019 013407          .ASCIZ <15><12>/ (1-255, 0=SEND UNTIL ^C):/
2020 013444  PMSMSG:: .ASCIZ /PRINT MODEM STATUS SIGNAL REPORT AFTER EACH PATTERN:/
2021 013531  TERMSG:: .ASCIZ /TYPE <CR> TO TERMINATE THE TEST:/
2022
2023                 .EVEN
2024                                 .LIST BIN
```

```
2026
2027                    ;*****************************************************************************
2028                    ;
2029                    ;               FVTSKL2.P11
2030                    ;
2031                    ;*****************************************************************************
2032
2033
2034
2035                    .SBTTL  GLOBAL ERROR REPORT SECTION
2036
2037                    ;++
2038                    ; THE GLOBAL ERROR REPORT SECTION CONTAINS MESSAGE PRINTING AREAS
2039                    ; USED BY MORE THAN ONE TEST TO OUTPUT ADDITIONAL ERROR INFORMATION.  PRINTB
2040                    ; (BASIC) AND PRINTX (EXTENDED) CALLS ARE USED TO CALL PRINT SERVICES.
2041                    ;--
```

```
                        .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). IF THE "EXTENDED ERROR INFO" OPTION HAS BEEN SELECTED
                        ;*      THEN THIS SUBROUTINE WILL REPORT THE TYPE OF ACCESS (READ OR WRITE OR
                        ;*      BOTH) WHICH CAUSED A BUS TIME-OUT TRAP (004 TRAP).A MESSAGE INDICATING
                        ;*      THAT THE DHU MAY BE AT THE WRONG UNIBUS 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.
                        ;**************************************************************************

013572                          BGNMSG  ER0101
013572                                                                  ER0101::
013572                          SAVE                    ;SAVE THE GPR CONTENTS.
013572  004567  171526                    JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

013576  012700  000100           MOV     #BIT06,R0       ;SET-UP THE BIT MAP FOR 'REPORT EXT'D ERROR INFO'
013602  046700  166354           BIC     OPTION,R0       ;TRY AND CLEAR THE FLAG.
013606  001036                  BNE     6$              ;EXIT IF OPTION NOT SELECTED.
                        ;*
                        ; REPORT EXTENDED ERROR INFOMATION
                        ;-

013610  032705  000001           BIT     #BIT0,R5        ;TEST FOR READ ERROR.
013614  001410                  BEQ     2$              ;SKIP READ ERROR MSG IF NO READ ERROR.
013616                          PRINTB  #MSG1           ;PRINT READ ERROR MESSAGE.
013616  012746  013710                                                  MOV     #MSG1,-(SP)
013622  012746  000001                                                  MOV     #1,-(SP)
013626  010600                                                          MOV     SP,R0
013630  104414                                                          TRAP    C$PNTB
013632  062706  000004                                                  ADD     #4,SP
013636  032705  000002   2$:     BIT     #BIT1,R5        ;TEST FOR WRITE ERROR.
013642  001410                  BEQ     4$              ;SKIP WRITE ERROR MSG IF NO WRITE ERROR.
013644                          PRINTB  #MSG2           ;PRINT WRITE ERROR MESSAGE.
013644  012746  013766                                                  MOV     #MSG2,-(SP)
013650  012746  000001                                                  MOV     #1,-(SP)
013654  010600                                                          MOV     SP,R0
013656  104414                                                          TRAP    C$PNTB
013660  062706  000004                                                  ADD     #4,SP
013664          4$:     PRINTX  #MSG3           ;SUGGEST THAT DHU MAY BE AT WRONG ADDRESS.
013664  012746  014045                                                  MOV     #MSG3,-(SP)
013670  012746  000001                                                  MOV     #1,-(SP)
013674  010600                                                          MOV     SP,R0
013676  104415                                                          TRAP    C$PNTX
013700  062706  000004                                                  ADD     #4,SP
```

```
  2083 013704                                6$:     PASS                     ;RESTORE THE GPR CONTENTS.
       013704  004736                                        JSR     PC,@(SP)+              ;RETURN TO PREGOS SUBRT.
  2084 013706                                        ENDMSG
       013706                                                                        L10002:
       013706  104423                                                                        TRAP    C$MSG
  2085
  2086 013710    045     101     102  MSG1::  .ASCIZ  /%ABU"  IME-OUT TRAP CAUSED BY READ ATTEMPT.%N/
       013713    125     123     040
       013716    124     111     115
       013721    105     055     117
       013724    125     124     040
       013727    124     122     101
       013732    120     040     103
       013735    101     125     123
       013740    105     104     040
       013743    102     131     040
       013746    122     105     101
       013751    104     040     101
       013754    124     124     105
       013757    115     120     124
       013762    056     045     116
       013765    000
  2087 013766    045     101     102  MSG2::  .ASCIZ  /%ABUS TIME-OUT TRAP CAUSED BY WRITE ATTEMPT.%N/
       013771    125     123     040
       013774    124     111     115
       013777    105     055     117
       014002    125     124     040
       014005    124     122     101
       014010    120     040     103
       014013    101     125     123
       014016    105     104     040
       014021    102     131     040
       014024    127     122     111
       014027    124     105     040
       014032    101     124     124
       014035    105     115     120
       014040    124     056     045
       014043    116     000
  2088 014045    045     101     104  MSG3::  .ASCIZ  /%ADHU MAY BE AT THE WRONG UNIBUS ADDRESS.%N%N/
       014050    110     125     040
       014053    115     101     131
       014056    040     102     105
       014061    040     101     124
       014064    040     124     110
       014067    105     040     127
       014072    122     117     116
       014075    107     040     125
       014100    116     111     102
       014103    125     123     040
       014106    101     104     104
       014111    122     105     123
       014114    123     056     045
       014117    116     045     116
       014122    000
  2089
  2090                                       .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, PROVIDED
                        ;*      EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
                        ;*
                        ;* 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.
                        ;**************************************************************************

014124                          BGNMSG  ER0503
014124                                                                          ER0503::
014124  012700  000100                   MOV     #BIT06,R0       ;TRY TO CLEAR THE
014130  046700  166026                   BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
014134  001011                          BNE     2$              ;EXIT IF FLAG NOT SET.


014136                          PRINTB  #EF0503,R1      ;PRINT THE MESSAGE.
014136  010146                                                                  MOV     R1,-(SP)
014140  012746  005453                                                          MOV     #EF0503,-(SP)
014144  012746  000002                                                          MOV     #2,-(SP)
014150  010600                                                                  MOV     SP,R0
014152  104414                                                                  TRAP    C$PNTB
014154  062706  000006                                                          ADD     #6,SP

014160                  2$:     ENDMSG
014160                                                                  L10003:
014160  104423                                                                  TRAP    C$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, PROVIDED EXTENDED ERROR INFOMATION HAS BEEN
                        ;*      REQUESTED, OTHERWISE ONLY A "TEST FAILURE" MESSAGE WILL BE PRINTED.
                        ;*
                        ;* 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.
                        ;*******************************************************************************
014162                          BGNMSG  ER1603
014162                                                                  ER1603::
014162                          SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
014162  004567  171136                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.

014166  012700  000100          MOV     #BIT06,R0       ;TRY TO CLEAR THE
014172  046700  165764          BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
014176  001024                  BNE     2$              ;EXIT IF FLAG NOT SET.


014200                          PRINTB  #EF0503,R1      ;PRINT BASIC MESSAGE ON OPERATORS CONSOLE.
014200  010146                                                          MOV     R1,-(SP)
014202  012746  005453                                                  MOV     #EF0503,-(SP)
014206  012746  000002                                                  MOV     #2,-(SP)
014212  010600                                                          MOV     SP,R0
014214  104414                                                          TRAP    C$PNTB
014216  062706  000006                                                  ADD     #6,SP

014222  016702  171072          MOV     ERRMSG,R2       ;GET THE "TEST MESSAGE".
014226                          PRINTB  #EF1601,R2      ;PRINT "TEST ABORTED" MESSAGE.
014226  010246                                                          MOV     R2,-(SP)
014230  012746  005460                                                  MOV     #EF1601,-(SP)
014234  012746  000002                                                  MOV     #2,-(SP)
014240  010600                                                          MOV     SP,R0
014242  104414                                                          TRAP    C$PNTB
014244  062706  000006                                                  ADD     #6,SP

014250  004736          2$:     PASS            ;RESTORE THE CONTENTS OF THE GPRS.
                                JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
014252                          ENDMSG
014252                                                          L10004:
014252  104423                                                          TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE            ER6201
                        ;**************************************************************************
                        ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS INTENDED FOR USE IN THE
                        ;*      FRAMING 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.  HESE ERRORS ARE REPORTED
                        ;*      ONLY IF EXTENDED ERROR REPORTING HAS BEEN REQUESTED.
                        ;*
                        ;* 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.
                        ;*                 BIT0 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 REPOR. 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.
                        ;**************************************************************************

014254                          BGNMSG  ER6201
014254                                                                  ER6201::
014254                          SAVE                    ;SAVE THE CONTENTS OF THE GPR'S.
014254  004567  171044                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

                        ;*
                        ; EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                        ;-
014260  032767  000100  165674          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
014266  001507                  BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                        ;DURING THE SOFTWARE QUESTIONS.

014270  016304  005234                MOV     TXRXLB(R3),R4   ;GET THE ASSOCIATED TX LINE NUMBER.
014274  006203                  ASR     R3              ;CALCULATE THE RX LINE NUMBER.
014276  006204                  ASR     R4              ;CALCULATE THE ASSOCIATED LINE NUMBER.
014300                          PRINTB  #EF6201,R3,R4   ;REPORT THE ERROR TYPE AND LINE NUMBERS.
014300  010446                                                          MOV     R4,-(SP)
014302  010346                                                          MOV     R3,-(SP)
014304  012746  005671                                                 MOV     #EF6201,-(SP)
014310  012746  000003                                                 MOV     #3,-(SP)
014314  010600                                                          MOV     SP,R0
014316  104414                                                          TRAP    C$PNTB
014320  062706  000010                                                 ADD     #10,SP

                        ;*
                        ; REPORT FRAMING ERROR PROBLEM.
                        ;-
014324  012704  011151                  MOV     #EM6202,R4      ;SELECT THE "ERROR BIT CLEAR" MESSAGE.
014330  012701  010077                  MOV     #EM0509,R1      ;SELECT EXPECTED "ERROR BIT SET" MESSAGE.
014334  032705  000002                  BIT     #BIT1,R5        ;TEST IF FRAMING ERROR MESSAGE TO BE REPORTED.
```

```
2208 014340  001427                    BEQ     6$              ;BRANCH TO REPORT PARITY ERROR.
2209 014342  032705  000001             BIT     #BIT0,R5        ;TEST "ERROR BIT SET/CLEAR" MESSAGE FLAG.
2210 014346  001403                    BEQ     2$              ;BRANCH TO REPORT ERROR BIT "CLEAR".
2211 014350  010401                     MOV     R4,R1           ;SELECT EXPECTED "CLEAR" STATE MESSAGE.
2212 014352  012704  010077             MOV     #EM0509,R4      ;SELECT THE "ERROR BIT SET" MESSAGE.
2213 014356                         2$: PRINTX  #EF6202,R4,R1   ;REPORT THE SOURCE OF THE PROBLEM.
     014356  010146                                                     MOV     R1,-(SP)
     014360  010446                                                     MOV     R4,-(SP)
     014362  012746  006004                                             MOV     #EF6202,-(SP)
     014366  012746  000003                                             MOV     #3,-(SP)
     014372  010600                                                     MOV     SP,R0
     014374  104415                                                     TRAP    C$PNTX
     014376  062706  000010                                             ADD     #10,SP
2214 014402  032705  000010             BIT     #BIT3,R5        ;TEST IF PARITY ERROR MESSAGE TO BE REPORTED.
2215 014406  001424                    BEQ     10$             ;EXIT IF PARITY ERROR REPORT TO BE SKIPPED.
2216 014410  012704  011151             MOV     #EM6202,R4      ;SELECT THE "CLEAR" MESSAGE.
2217 014414  012701  010077             MOV     #EM0509,R1      ;SELECT THE EXPECTED "SET" STATE MESSAGE.
2218                                    ;+
2219                                    ; REPORT PARITY ERROR PROBLEM.
2220                                    ;-
2221
2222 014420  032705  000004         6$: BIT     #BIT2,R5        ;TEST "SET"/"CLEAR" MESSAGE FLAG.
2223 014424  001403                    BEQ     8$              ;BRANCH TO REPORT ERROR BIT CLEAR.
2224 014426  010401                     MOV     R4,R1           ;SELECT THE EXPECTED "CLEAR" STATE MESSAGE.
2225 014430  012704  010077             MOV     #EM0509,R4      ;SELECT THE "ERROR BIT SET" MESSAGE.
2226 014434                         8$: PRINTX  #EF6203,R4,R1   ;REPORT THE SOURCE OF THE PROBLEM.
     014434  010146                                                     MOV     R1,-(SP)
     014436  010446                                                     MOV     R4,-(SP)
     014440  012746  006102                                             MOV     #EF6203,-(SP)
     014444  012746  000003                                             MOV     #3,-(SP)
     014450  010600                                                     MOV     SP,R0
     014452  104415                                                     TRAP    C$PNTX
     014454  062706  000010                                             ADD     #10,SP
2227
2228 014460                        10$: PRINTX  #EF1603,R2      ;REPORT ACTUAL DATA RECEIVED.
     014460  010246                                                     MOV     R2,-(SP)
     014462  012746  005512                                             MOV     #EF1603,-(SP)
     014466  012746  000002                                             MOV     #2,-(SP)
     014472  010600                                                     MOV     SP,R0
     014474  104415                                                     TRAP    C$PNTX
     014476  062706  000006                                             ADD     #6,SP
2229
2230 014502  004767  006152             JSR     PC,PRTLPR       ;REPORT THE CONTENTS OF THE LPR FOR THIS LINE.
2231 014506                         60$: PASS                   ;RESTORE THE CONTENTS OF THE GPR'S.
     014506  004736                                                     JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
2232 014510                         ENDMSG
     014510                                                         L10005:
     014510  104423                                                     TRAP    C$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.
                        ;*****************************************************************************

014512                      BGNMSG  ER9001
014512                                                              ER9001::

                            ;*
                            ; EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                            ;-
014512  032767  000100  165442      BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
014520  001433                      BEQ     2$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                            ;DURING THE SOFTWARE QUESTIONS.

014522                      PRINTB  #EF9001,R1      ;REPORT TYPE OF CODE FOUND.
014522  010146                                                  MOV     R1,-(SP)
014524  012746  006235                                         MOV     #EF9001,-(SP)
014530  012746  000002                                         MOV     #2,-(SP)
014534  010600                                                  MOV     SP,R0
014536  104414                                                  TRAP    C$PNTB
014540  062706  000006                                         ADD     #6,SP
014544                      PRINTX  #EF9002,R4      ;REPORT THE LINE NUMBER OF THE CODE.
014544  010446                                                  MOV     R4,-(SP)
014546  012746  006317                                         MOV     #EF9002,-(SP)
014552  012746  000002                                         MOV     #2,-(SP)
014556  010600                                                  MOV     SP,R0
014560  104415                                                  TRAP    C$PNTX
014562  062706  000006                                         ADD     #6,SP
014566                      PRINTX  #EF9003,R2      ;REPORT THE CODE WHICH WAS FOUND.
014566  010246                                                  MOV     R2,-(SP)
014570  012746  006371                                         MOV     #EF9003,-(SP)
014574  012746  000002                                         MOV     #2,-(SP)
014600  010600                                                  MOV     SP,R0
014602  104415                                                  TRAP    C$PNTX
014604  062706  000006                                         ADD     #6,SP

014610              2$:     ENDMSG
014610                                                      L10006:
014610  104423                                                  TRAP    C$MSG
```

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

014612                                  BGNMSG  ER9002
014612                                                                  ER9002::

                                    ;*
                                    ; EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                    ;-
014612  032767  000100  165342          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
014620  001462                          BEQ     62$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                                ;DURING THE SOFTWARE QUESTIONS.

014622  006203                          ASR     R3              ;CALCULATE THE LINE NUMBER.
014624  042702  177400                  BIC     #177400,R2      ;MASK OUT ALL BUT DATA IN READ CHAR.
014630                          PRINTB  #EF9006,R1,R3   ;PRINT THE FIRST LINE OF THE MESSAGE.
014630  010346                                                          MOV     R3,-(SP)
014632  010146                                                          MOV     R1,-(SP)
014634  012746  006501                                                  MOV     #EF9006,-(SP)
014640  012746  000003                                                  MOV     #3,-(SP)
014644  010600                                                          MOV     SP,R0
014646  104414                                                          TRAP    C$PNTB
014650  062706  000010                                                  ADD     #10,SP
014654                          PRINTX  #EF9004,#EM9010,R2      ;PRINT ACTUAL DATA.
014654  010246                                                          MOV     R2,-(SP)
014656  012746  011541                                                  MOV     #EM9010,-(SP)
014662  012746  006420                                                  MOV     #EF9004,-(SP)
014666  012746  000003                                                  MOV     #3,-(SP)
014672  010600                                                          MOV     SP,R0
014674  104415                                                          TRAP    C$PNTX
014676  062706  000010                                                  ADD     #10,SP
014702  005704                          TST     R4              ;CHECK FOR "NONE" CODE SET IN EXPECTED DATA.
014704  100414                  BMI     2$              ;BRANCH TO PRINT "NONE" MESSAGE IF FLAG SET.
014706                          PRINTX  #EF9004,#EM9009,R4      ;PRINT EXPECTED DATA.
014706  010446                                                          MOV     R4,-(SP)
014710  012746  011515                                                  MOV     #EM9009,-(SP)
014714  012746  006420                                                  MOV     #EF9004,-(SP)
014720  012746  000003                                                  MOV     #3,-(SP)
```

```
        014724  010600                                                          MOV     SP,R0
        014726  104415                                                          TRAP    C$PNTX
        014730  062706  000010                                                  ADD     #10,SP
2307    014734  000412                  BR      60$             ;EXIT THIS ROUTINE.
2308    014736                  2$:     PRINTX  #EF9005,#EM9009 ;PRINT MESSAGE INDICATING NO EXPECTED DATA.
        014736  012746  011515                                                  MOV     #EM9009,-(SP)
        014742  012746  006450                                                  MOV     #EF9005,-(SP)
        014746  012746  000002                                                  MOV     #2,-(SP)
        014752  010600                                                          MOV     SP,R0
        014754  104415                                                          TRAP    C$PNTX
        014756  062706  000006                                                  ADD     #6,SP
2309    014762  004767  005672          60$:    JSR     PC,PRTLPR       ;PRINT CONTENTS OF THE LPR REGISTER.
2310    014766                  62$:    ENDMSG
        014766                                                          L10007:
        014766  104423                                                          TRAP    C$MSG
```

```
                                .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9003 -
                                ;*******************************************************************************
                                ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH IS INTENDED FOR USE IN THE
                                ;*      TRANMSISSION 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.
                                ;*******************************************************************************

014770                              BGNMSG  ER9003
014770                                                                  ER9003::

                                    ;*
                                    ; EXIT THE TEST IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                    ;-
014770  032767  000100  165164       BIT     #BIT6,OPTION    ;EXIT WITH TEST FAILURE MESSAGE IF
014776  001470                      BEQ     62$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                            ;DURING THE SOFTWARE QUESTIONS.

015000  006203                       ASR     R3              ;CALCULATE THE LINE NUMBER.
015002                              PRINTB  #EF9007,R3      ;REPORT THE ERROR TYPE AND LINE NUMBER.
015002  010346                                                          MOV     R3,-(SP)
015004  012746  006520                                                  MOV     #EF9007,-(SP)
015010  012746  000002                                                  MOV     #2,-(SP)
015014  010600                                                          MOV     SP,R0
015016  104414                                                          TRAP    C$PNTB
015020  062706  000006                                                  ADD     #6,SP
015024  010201                       MOV     R2,R1           ;EXTRACT THE RECEIVED CHARACTER FROM THE
015026  042701  177400               BIC     #177400,R1      ;  PASSED IN CHAR VALUE WITH FLAGS.
015032                              PRINTX  #EF9008,R1      ;REPORT THE VALUE OF THE RECEIVED CHAR.
015032  010146                                                          MOV     R1,-(SP)
015034  012746  006614                                                  MOV     #EF9008,-(SP)
015040  012746  000002                                                  MOV     #2,-(SP)
015044  010600                                                          MOV     SP,R0
015046  104415                                                          TRAP    C$PNTX
015050  062706  000006                                                  ADD     #6,SP

                                    ;*
                                    ; REPORT OVERRUN FLAG SET IF NECESSARY.
                                    ;-
015054  012701  011565               MOV     #EM9011,R1      ;SELECT THE OVERRUN ERROR MESSAGE.
015060  032702  040000               BIT     #BIT14,R2       ;CHECK OVERRUN ERROR FLAG IN PASSED IN CHAR.
015064  001402                      BEQ     2$              ;SKIP ERROR IF OVERRUN ERROR FLAG WAS CLEAR.
015066  004767  000036               JSR     PC,50$          ;REPORT THE OVERRUN ERROR FLAG WAS SET.
                                    ;*
```

```
                                ; REPORT FRAMING FLAG SET IF NECESSARY.
                                ;-
015072  012701  011575          2$:     MOV     #EM9012,R1      ;SELECT THE FRAMING ERROR MESSAGE.
015076  032702  020000                  BIT     #BIT13,R2       ;CHECK FRAMING ERROR FLAG IN PASSED IN CHAR.
015102  001402                  BEQ     4$              ;SKIP ERROR IF FRAMING ERROR FLAG WAS CLEAR.
015104  004767  000020                  JSR     PC,50$          ;REPORT THE FRAMING ERROR MESSAGE.
                                ;+
                                ; REPORT PARITY FLAG SET IF NECESSARY.
                                ;-
015110  012701  011605          4$:     MOV     #EM9013,R1      ;SELECT THE PARITY ERROR MESSAGE.
015114  032702  010000                  BIT     #BIT12,R2       ;CHECK PARITY ERROR FLAG IN PASSED IN CHAR.
015120  001415                  BEQ     60$             ;EXIT ROUTINE IF PARITY ERRO FLAG WAS CLEAR.
015122  004767  000002                  JSR     PC,50$          ;REPORT THE PARITY ERROR MESSAGE.
015126  000412                  BR      60$             ;EXIT THIS ROUTINE.

                                ;+
                                ; LOCAL SUBROUTINE TO REPORT AN ERROR FLAG STATUS.
                                ;-
015130                          50$:    PRINTX  #EF9009,R1
015130  010146                                                                  MOV     R1,-(SP)
015132  012746  006653                                                          MOV     #EF9009,-(SP)
015136  012746  000002                                                          MOV     #2,-(SP)
015142  010600                                                                  MOV     SP,R0
015144  104415                                                                  TRAP    C$PNTX
015146  062706  000006                                                          ADD     #6,SP
015152  000207                          RTS     PC

015154  004767  005500          60$:    JSR     PC,PRTLPR       ;REPORT THE LPR CONTENTS FOR THIS LINE.
015160                          62$:    ENDMSG
015160                                                                  L10010:
015160  104423                                                                  TRAP    C$MSG
```

J5


```
                        .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, PROVIDED EXTENDED ERROR REPORTING HAS
                        ;*      BEEN REQUESTED BY THE OPERATOR.
                        ;*
                        ;* 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.
                        ;*************************************************************************

015162                          BGNMSG  ER9004
015162                                                                  ER9004::
015162  012700  000100                    MOV     #BIT06,R0       ;TRY TO CLEAR THE
015166  046700  164770                    BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
015172  001040                            BNE     6$              ;EXIT IF FLAG NOT SET.

015174                          PRINTB  #EF0503,#EM9014 ;REPORT THE SECONDARY ERROR MESSAGE.
015174  012746  011614                                                  MOV     #EM9014,-(SP)
015200  012746  005453                                                  MOV     #EF0503,-(SP)
015204  012746  000002                                                  MOV     #2,-(SP)
015210  010600                                                          MOV     SP,R0
015212  104414                                                          TRAP    C$PNTB
015214  062706  000006                                                  ADD     #6,SP
015220  005002                    CLR     R2              ;CLEAR THE LINE COUNTER.
015222  016703  165252            MOV     ERSMRF,R3       ;GET THE ERROR SUMMARY FLAGS.
015226  005004                    CLR     R4              ;CLEAR "LINE COUNTER TIMES 2" OFFSET.
015230  000241          2$:       CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
015232  006003                    ROR     R3              ;SHIFT ANOTHER ERROR SUMMARY FLAG INTO CARRY.
015234  103013                    BCC     4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
015236                          PRINTX  #EF9010,R2,ERCNTB(R4)
015236  016446  003302                                                  MOV     ERCNTB(R4),-(SP)
015242  010246                                                          MOV     R2,-(SP)
015244  012746  006712                                                  MOV     #EF9010,-(SP)
015250  012746  000003                                                  MOV     #3,-(SP)
015254  010600                                                          MOV     SP,R0
015256  104415                                                          TRAP    C$PNTX
015260  062706  000010                                                  ADD     #10,SP
015264  012405          4$:       MOV     (R4)+,R5        ;INCREMENT THE LINE OFFSET BY 2.
015266  005202                    INC     R2              ;INCREMT THE LINE COUNTER.
015270  005703                    TST     R3              ;CHECK THE ERROR SUMMARY FLAGS.
015272  001356                    BNE     2$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.

015274          6$:       ENDMSG
015274                                                          L10011:
```

```
        015274  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.
                        ;*************************************************************************

015276                          BGNMSG  ER9005
015276                                                                  ER9005::
015276                          SAVE                    ;SAVE THE CONTENTS OF THE GPR'S.
015276  004567  170022                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

015302  012700  000100           MOV     #BIT06,R0       ;TRY TO CLEAR THE
015306  046700  164650           BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
015312  001107                  BNE     10$             ;EXIT IF FLAG NOT SET.

015314                          PRINTB  #EF9013,R1      ;REPORT THE SECONDARY ERROR MESSAGE.
015314  010146                                                  MOV     R1,-(SP)
015316  012746  007115                                          MOV     #EF9013,-(SP)
015322  012746  000002                                          MOV     #2,-(SP)
015326  010600                                                  MOV     SP,R0
015330  104414                                                  TRAP    C$PNTB
015332  062706  000006                                          ADD     #6,SP
015336  005003           CLR     R3              ;CLEAR THE LINE COUNTER.
015340  022701  011710           CMP     #EM9015,R1      ;CHECK IF ADDRESS CORRESPONDS TO TX MESSAGE.
015344  001032                  BNE     6$              ;BRANCH IF RECEPTION MESSAGE TO BE PRINTED.

                        ;+
                        ; PERFORM TX INCOMPLETE ERROR MESSAGE REPORTING.
                        ;-
015346                          PRINTX  #EM9030         ;PRINT "NO TX COMPLETION INTERRUPTS RECEIVED"
015346  012746  012323                                          MOV     #EM9030,-(SP)
015352  012746  000001                                          MOV     #1,-(SP)
015356  010600                                                  MOV     SP,R0
015360  104415                                                  TRAP    C$PNTX
015362  062706  000004                                          ADD     #4,SP
015366  000241           2$:     CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
015370  006002           ROR     R2              ;SHIFT "TX NOT DONE" FLAG INTO CARRY.
015372  103013                  BCC     4$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
015374                          PRINTX  #EF9020,R3      ;PRINT "TOO FEW TX.ACTIONS GENERATED" MSG.
015374  010346                                                  MOV     R3,-(SP)
```

```
        015376  012746  007201                                                  MOV     #EF9020, (SP)
        015402  012746  000002                                                  MOV     #2,-(SP)
        015406  010600                                                          MOV     SP,R0
        015410  104415                                                          TRAP    C$PNTX
        015412  062706  000006                                                  ADD     #6,SP
  2466  015416  004767  005236          JSR     PC,PRTLPR       ;REPORT CONTENTS OF LPR REGISTER FOR THIS LINE.
  2467  015422  005203          4$:      INC     R3              ;INCREMENT LINE COUNTER.
  2468  015424  005702                   TST     R2              ;CHECK THE "TX NOT DONE FLAGS".
  2469  015426  001357                  BNE     2$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.
  2470  015430  000440                  BR      10$             ;EXIT THIS ROUTINE.
  2471                                  ;+
  2472                                  ; PERFORM RX INCOMPLETE ERROR MESSAGE REPORTING.
  2473                                  ;-
  2474  015432  000241          6$:      CLC                     ;CLEAR THE CARRY FOR THE FOLLOWING ROTATE.
  2475  015434  006002                   ROR     R2              ;SHIFT "RX NOT DONE" FLAG INTO CARRY.
  2476  015436  103031                  BCC     8$              ;SKIP PRINTING MESSAGE IF FLAG FOR LINE CLEAR.
  2477  015440  006303                   ASL     R3              ;SHIFT LINE # TO GIVE CORRECT TABLE OFFSET.
  2478  015442  016305  005234          MOV     TXRXLB(R3),R5   ;GET THE "ASSOCIATED" RECEIVE LINE OFFSET.
  2479  015446  010246                   MOV     R2,-(SP)        ;SAVE THE "RX NOT DONE" FLAGS ON THE STACK.
  2480  015450  010502                   MOV     R5,R2           ;COPY THE ASSOCIATED TX LINE OFFSET.
  2481  015452  016505  003442          MOV     CHCNTB(R5),R5   ;GET THE TOTAL NUMBER OF EXPECTED CHARS.
  2482  015456  006202                   ASR     R2              ;SHIFT THE TABLE OFFSET TO GIVE A LINE NUMBER.
  2483  015460  006203                   ASR     R3              ;SHIFT TABLE OFFSET TO GIVE LINE NUMBER.
  2484  015462                          PRINTX  #EF9012,R3,R1,(R4),R5,R2  ;REPORT NUMBER OF CHARS ON LINE.
        015462  010246                                                          MOV     R2,-(SP)
        015464  010546                                                          MOV     R5,-(SP)
        015466  011446                                                          MOV     (R4),-(SP)
        015470  010146                                                          MOV     R1,-(SP)
        015472  010346                                                          MOV     R3,-(SP)
        015474  012746  007001                                                  MOV     #EF9012,-(SP)
        015500  012746  000006                                                  MOV     #6,-(SP)
        015504  010600                                                          MOV     SP,R0
        015506  104415                                                          TRAP    C$PNTX
        015510  062706  000016                                                  ADD     #16,SP
  2485  015514  012602                   MOV     (SP)+,R2        ;RESTORE THE "RX NOT DONE" FLAGS.
  2486  015516  004767  005136          JSR     PC,PRTLPR       ;REPORT CONTENTS OF LPR REGISTER FOR THIS LINE.
  2487  015522  005724          8$:      TST     (R4)+           ;INCREMENT THE CHARACTER COUNTER TABLE.
  2488  015524  005203                   INC     R3              ;INCREMENT THE LINE COUNTER.
  2489  015526  005702                   TST     R2              ;CHECK THE "RX NOT DONE FLAGS".
  2490  015530  001340                  BNE     6$              ;IF MORE FLAGS SET, LOOP TO DO OTHER LINES.
  2491  015532          10$:     PASS                            ;RESTORE THE CONTENTS OF THE GPRS.
        015532  004736                           JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
  2492  015534                  ENDMSG
        015534                                                          L10012:
        015534  104423                                                         TRAP    C$MSG
```

```
                        .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9101 -
                        ;*************************************************************************
                        ;*      THIS IS A GENERAL ERROR REPORTING SUBROUTINE WHICH REPORTS A MESSAGE
                        ;*      WHICH TAKES A SINGLE, 2 DIGIT DECIMAL ARGUMENT AFTER THE END OF AN
                        ;*      ASCII MESSAGE.
                        ;*
                        ;* INPUTS:      R1 - VALUE TO BE PRINTED AFTER MSG AS 2 DECIMAL DIGITS.
                        ;*              R2 - ADDRESS OF MESSAGE TO PRINT FIRST.
                        ;*
                        ;* OUTPUTS:     A MESSAGES IS PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9101" AS THE MESSAGE POINTER
                        ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                        ;*
                        ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
                        ;*
                        ;* SUBORDINATE ROUTINES USED: NONE.
                        ;*************************************************************************

015536                          BGNMSG  ER9101
015536                                                                  ER9101::

015536  012700  000100                MOV     #BIT06,R0       ;TRY TO CLEAR THE
015542  046700  164414                BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
015546  001012                        BNE     2$              ;EXIT IF FLAG NOT SET.


015550                          PRINTB  #EF9006,R2,R1   ;REPORT THE STRING FOLLOWED BY THE NUMBER.
015550  010146                                                          MOV     R1,-(SP)
015552  010246                                                          MOV     R2,-(SP)
015554  012746  006501                                                  MOV     #EF9006,-(SP)
015560  012746  000003                                                  MOV     #3,-(SP)
015564  010600                                                          MOV     SP,R0
015566  104414                                                          TRAP    C$PNTB
015570  062706  000010                                                  ADD     #10,SP

015574                  2$:     ENDMSG
015574                                                          L10013:
015574  104423                                                          TRAP    C$MSG
```

```
                                .SBTTL  GLOBAL ERROR REPORTING ROUTINE          - ER9102 -
                                ;*****************************************************************************
                                ;*      THIS IS AN ERROR REPORTING SUBROUTINE WHICH PRINTS ADDITIONAL ERROR
                                ;*      INFORMATION AFTER THE ERROR MESSAGE HEADER, PROVIDED EXTENDED ERROR
                                ;*      REPORTING HAS BEEN REQUESTED.
                                ;*      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 - BIT 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 MESSAGE 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 LINE ARE PRINTED
                                ;*              AS BASIC ERROR INFORMATION.
                                ;*
                                ;* SUBORDINATE ROUTINES USED: NONE.
                                ;*****************************************************************************

015576                                  BGNMSG  ER9102
015576                                                                  ER9102::
015576                                  SAVE                    ;SAVE THE CONTENTS OF THE GPRS.
015576  004567  167522                                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
                                ;*
                                ; EXIT IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
                                ;-
015602  032767  000100  164352          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
015610  001441                        BEQ     60$             ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
                                                                ;DURING THE SOFTWARE QUESTIONS.

015612                                  PRINTB  #EF0503,R1      ;PRINT THE FIRST LINE OF THE MESSAGE.
015612  010146                                                                  MOV     R1,-(SP)
015614  012746  005453                                                         MOV     #EF0503,-(SP)
015620  012746  000002                                                         MOV     #2,-(SP)
015624  010600                                                                  MOV     SP,R0
015626  104414                                                                  TRAP    C$PNTB
015630  062706  000006                                                         ADD     #6,SP
                                        CLR     R3              ;CLEAR THE LINE NUMBER.
                                2$:     CLC                     ;PREPARE TO ROTATE NEXT BIT OUT OF MAP.
                                        ROR     R2              ;GET THE NEXT BIT OF THE BIT MAP.
                                        BCC     4$              ;SKIP PRINTING MESSAGE IF THE BIT IS CLEAR.
                                        PRINTB  #EF9103,R3      ;REPORT THIS LINE HAD THE ERROR.
015644  010346                                                                  MOV     R3,-(SP)
015646  012746  007265                                                         MOV     #EF9103,-(SP)
015652  012746  000002                                                         MOV     #2,-(SP)
015656  010600                                                                  MOV     SP,R0
015660  104414                                                                  TRAP    C$PNTB
015662  062706  000006                                                         ADD     #6,SP
```

```
2567 015666  005203                  4$:     INC     R3              ;INCREMENT THE LINE COUNTER.
2568 015670  005702                          TST     R2              ;CHECK THE BIT MAP.
2569 015672  001361                          BNE     2$              ;LOOP IF NOT ALL SET BITS REMOVED FROM BIT MAP.
2570 015674                                  PRINTB  #EF9101         ;PRINT A BLANK LINE.
     015674  012746  007262                                                        MOV     #EF9101,-(SP)
     015700  012746  000001                                                        MOV     #1,-(SP)
     015704  010600                                                                MOV     SP,R0
     015706  104414                                                                TRAP    C$PNTB
     015710  062706  000004                                                        ADD     #4,SP
2571 015714                                  60$:    PASS                    ;RESTORE THE SAVED CONTENTS OF THE GPRS
     015714  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
2572 015716                                  ENDMSG
     015716                                                                L10014:
     015716  104423                                                                TRAP    C$MSG
```

```
                                .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.
                                ;*      PROVIDED EXTENDED ERROR REPORTING HAS BEEN ENABLED.
                                ;*
                                ;* INPUTS:      R1 - THE ADDRESS OF THE FIRST MESSAGE TO BE REPORTED.
                                ;*              R2 - THE ADDRESS OF THE NEXT EMPTY CELL IN THE QUEUE.
                                ;*
                                ;* OUTPUTS:     THE TEST NUMBER FOLLOWED BY THE BMP CODE ARE PRINTED AT THE
                                ;*              OPERATOR CONSOLE.
                                ;*
                                ;* CALLING SEQUENCE:    INCLUDE THE LABEL "ER9301" AS THE MESSAGE POINTER
                                ;*                      PARAMETER IN THE DIAG SUPER ERROR REPORT MACRO CALL.
                                ;*
                                ;* COMMENTS:    THE MESSAGE IS PRINTED AS BASIC ERROR INFORMATION.
                                ;*
                                ;* SUBORDINATE ROUTINES USED: NONE.
                                ;*******************************************************************************

015720                          BGNMSG  ER9301
015720                                                                  ER9301::
015720                          SAVE                    ;SAVE THE GPRS ON THE STACK.
015720  004567  167400                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

015724  012700  000100          MOV     #BIT06,R0       ;TRY TO CLEAR THE
015730  046700  164226          BIC     OPTION,R0       ;EXT'D ERROR REPORTING FLAG
015734  001064                  BNE     60$             ;EXIT IF FLAG NOT SET.

015736                          PRINTB  #CF0503,R1      ;REPORT UNEXPECTED BMP CODES FOUND.
015736  010146                                                  MOV     R1,-(SP)
015740  012746  005453                                          MOV     #EF0503,-(SP)
015744  012746  000002                                          MOV     #2,-(SP)
015750  010600                                                  MOV     SP,R0
015752  104414                                                  TRAP    C$PNTB
015754  062706  000006                                          ADD     #6,SP
015760  012703  002512          MOV     #BMPCQB,R3      ;GET THE START ADDRESS OF THE BMP CODE QUEUE.
015764  012705  012714          MOV     #EM9302,R5      ;GET THE MESSAGE TO BE REPORTED.
015770  012301          2$:     MOV     (R3)+,R1        ;GET THE NUMBER OF THE TEST THAT WAS EXECUTING.
015772  012304                  MOV     (R3)+,R4        ;GET BMP CODE THAT WAS REPORTED OFF THE QUEUE.
015774  004767  000056          JSR     PC,50$          ;GO REPORT THE BMP CODE.
016000  020302                  CMP     R3,R2           ;CHECK IF ALL CODES HAVE BEEN REPORTED.
016002  103772                  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.
                                ;-
016004  020227  002706          CMP     R2,#BMPCQE-4    ;CHECK IF THE POINTER IS AT THE LAST LOCATION.
016010  001036                  BNE     60$             ;EXIT IF NOT AT THE LAST LOCATION.
016012  005762  000002          TST     2(R2)           ;CHECK FOR A BMP CODE IN THE LAST CELL
016016  001433                  BEQ     60$             ;EXIT IF NO OVERFLOW HAS OCCURED, CELL EMPTY.
016020  012301                  MOV     (R3)+,R1        ;GET THE TEST NUMBER OFF THE QUEUE.
016022  011304                  MOV     (R3),R4         ;GET THE BMP CODE OFF THE QUEUE.
016024  012705  012744          MOV     #EM9303,R5      ;SELECT THE MESSAGE TO BE REPORTED.
```

```
2623 016030                            PRINTX  #EF9302          ;REPORT OVERFLOW CONDITION.
     016030  012746  007401                                                    MOV     #EF9302,-(SP)
     016034  012746  000001                                                    MOV     #1,-(SP)
     016040  010600                                                            MOV     SP,R0
     016042  104415                                                            TRAP    C$PNTX
     016044  062706  000004                                                    ADD     #4,SP
2624 016050  004767  000002            JSR     PC,50$          ;REPORT THE LAST BMP CODE PLACED ON THE QUEUE.
2625 016054  000414                    BR      60$             ;EXIT.
2626
2627 016056                    50$:    PRINTX  #EF9301,R5,R1,R4  ;PRINT THE MESSAGE.
     016056  010446                                                            MOV     R4,-(SP)
     016060  010146                                                            MOV     R1,-(SP)
     016062  010546                                                            MOV     R5,-(SP)
     016064  012746  007333                                                    MOV     #EF9301,-(SP)
     016070  012746  000004                                                    MOV     #4,-(SP)
     016074  010600                                                            MOV     SP,R0
     016076  104415                                                            TRAP    C$PNTX
     016100  062706  000012                                                    ADD     #12,SP
2628 016104  000207                    RTS     PC              ;RETURN.
2629 016106                    60$:    PASS                    ;RESTORE THE GPR CONTENTS.
     016106  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2630
2631 016110                            ENDMSG
     016110                                                                L10015:
     016110  104423                                                            TRAP    C$MSG
```

```
2633                            .SBTTL  GLOBAL SUBROUTINES SECTION
2635                            ;**********************************************************************************************
2636                            ;
2637                            ;               FVTSKL3.P11
2638                            ;
2639                            ;**********************************************************************************************
2641
2642
2643                            ;++
2644                            ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
2645                            ; THAT ARE USED IN MORE THAN ONE TEST.
2646                            ; -
```

```
                                .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.
                                ;-- ************************************************************************

016112                          ALTFLD:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
016112  004567  167206                         JSR     R5,PREG05       ;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.
                                ;-
016116  010400                          MOV     R4,R0           ;CALCULATE THE NEW CONTENTS OF THE
016120  005100                          COM     R0              ;  REGISTER FIELDS WHICH ARE TO BE
016122  040002                          BIC     R0,R2           ;  ALTERRED BY THIS ROUTINE.
016124  016705  164104                  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).
                                ;-
016130  000241                          CLC                     ;PRE ARE FOR ROTATE, "TST R5" DOES THIS BELOW.
016132  006003                  2$:     ROR     R3              ;GET THE LINE SELECT BIT FOR THIS LINE.
016134  103006                          BCC     4$              ;SKIP SETUP IF LINE IS NOT SELECTED.
016136  010577  164036                  MOV     R5,@CSRA        ;SET OUT CSR IND.ADR.REG FIELD TO THIS LINE.
016142  011100                          MOV     (R1),R0         ;GET THE PRESENT CONTENTS OF THE REG TO ALTER.
016144  040400                          BIC     R4,R0           ;CLEAR THE BIT FIELDS WE ARE TO ALTER.
016146  050200                          BIS     R2,R0           ;OR IN THE NEW STATES OF THE FIELDS.
016150  010011                          MOV     R0,(R1)         ;WRITE THE NEW REGISTER CONTENTS TO THE REG.
016152  005205                  4$:     INC     R5              ;SET LINE NUMBER TO THE NEXT LINE.
016154  005703                          TST     R3              ;CHECK FOR UNHANDLED LINES, CLEAR CARRY FLAG.
016156  001365                          BNE     2$              ;LOOP IF SELECTED LINE(S) IS NOT HANDLED.
```

```
2704
2705 016160                        60$:    PASS                    ;RESTORE GPRS.
     016160  004736                                JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
2706 016162  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.
                        ;-- ************************************************************************

016164                  CALMSL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016164  004567  167134                JSR     R5,PREG05         ;CALL REGISTER SAVE SUBRT.
016170  005067  000210          CLR     62$                     ;CLEAR THE 2ND TIME FLAG.
                        ;*
                        ; SYNCHRONIZE WITH THE LTC.
                        ;-
016174  012705  000001  2$:     MOV     #1,R5                   ;SET OUTER LOOP COUNTER TO 1 LOOP.
                                                        ;INCREASE THE VALUE LOADED INTO THIS COUNTER IF THE <**
                                                        ;FOLLOWING LOOP FAILS ON FUTURE, FASTER PROCESSORS. <**
016200  005000                  CLR     R0                      ;CLEAR THE WAIT FOR CLOCK INT COUNTER.
016202  012767  000001  164070  MOV     #1,TIMER1               ;SET UP COUNT OF 1 TO SYNCH WITH LTC.
016210  005767  164064  4$:     TST     TIMER1                  ;CHECK FOR COUNTER HAVING GONE TO ZERO.
016214  001410                  BEQ     6$                      ;JUMP OUT OF LOOP IF LTC HAS INTERRUPTED.
016216  005200                  INC     R0                      ;COUNT THIS ITERATION OF THE INNER LOOP.
016220  001373                  BNE     4$                      ;LOOP IF COUNTER HAS NOT TURNED OVER.
016222  005305                  DEC     R5                      ;DECREMENT THE INNER LOOP COUNTER.
016224  003371                  BGT     4$                      ;LOOP IF OUTER LOOP COUNT NOT UP.
                        ;*
                        ; IF WE GOT NO LTC INTERRUPT, INDICATE THAT THERE IS NO LTC AVAILABLE.
                        ; LTC MUST BE FLAKEY, OR NOT REALLY AN LTC AT ALL.
                        ;-
016226  005067  164044          CLR     CLKHRZ                  ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
016232  000241                  CLC                             ;INDICATE FAILURE FOR RETURN.
016234  000461                  BR      60$                     ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
                        ;*
                        ; WE ARE NOW SYNCHRONIZED WITH THE LTC.
                        ; SET UP FOR THE CALIBRATION LOOP.
                        ;-
016236  012704  002300  6$:     MOV     #TIMER1,R4              ;WILL TEST TIMER1 IN THE LOOP BELOW.
```

```
2764 016242  005001                        CLR     R1              ;CLEAR THE OUTER LOOP COUNTER.
2765 016244  005002                        CLR     R2              ;INDICATE TO CHECK ALL BITS OF TIMER1.
2766 016246  005003                        CLR     R3              ;INDICATE TO CHECK FOR TIMER1 CLEAR.
2767 016250  012714  000001                MOV     #1,(R4)         ;LOAD TIMER1 WITH COUNT OF 1.
2768
2769 016254  016705  164032        8$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
2770 016260  011400                10$:    MOV     (R4),R0         ;GET THE TIMER1 VALUE.
2771 016262  010067  000120                MOV     R0,64$          ;SAVE WORD (LIKE IN THE REAL LOOP).
2772 016266  040200                        BIC     R2,R0           ;LEAVE ALL THE BITS.
2773 016270  020003                        CMP     R0,R3           ;COMPARE AGAINST ZERO.
2774 016272  000261                        SEC                     ;SET CARRY IN CASE OF SUCCESS.
2775 016274  001406                        BEQ     12$             ;EXIT LOOP IF TIMER1 HAS CLEARED.
2776 016276  005305                        DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
2777 016300  001367                        BNE     10$             ;LOOP IF MS NOT UP.
2778 016302  005301                        DEC     R1              ;DECREMENT THE MS TIME COUNT.
2779 016304  001363                        BNE     8$              ;KEEP LOOPING.
2780 016306  004767  003604                JSR     PC,OOPS         ;WE OVERFLOWED, SOMETHING IS WRONG, ABORT.
2781                                ;+
2782                                ; WE HAVE NOW HAVE LOOP COUNT INFORMATION FOR ONE CLOCK TICK.
2783                                ; WE HAVE NEGATIVE OF NUMBER OF OUTER LOOPS IN R1, EACH IS MSLCNT INNER LOOPS.
2784                                ; WE HAVE THE PORTION OF THE LAST OUTER LOOP NOT EXECUTED, IN R5.
2785                                ; NOW WE CALCULATE THE TOTAL NUMBER OF INNER LOOPS EXECUTED.
2786                                ;-
2787 016312  005401                12$:    NEG     R1              ;GET NUMBER OF OUTER LOOPS.
2788 016314  016702  163772                MOV     MSLCNT,R2       ;GET THE NUMBER OF INNER LOOPS PER OUTER LOOP.
2789 016320  010203                        MOV     R2,R3           ;COPY NUMBER OF LOOPS FOR MULTIPLY.
2790 016322  160502                        SUB     R5,R2           ;CALC # OF INNER LOOPS DONE IN LAST OUTER LOOP
2791 016324  010204                        MOV     R2,R4           ;  AND ADD TO ACCUMULATOR LSWORD.
2792 016326  005005                        CLR     R5              ;CLEAR ACCUMULATOR MSWORD.
2793 016330  005301                14$:    DEC     R1              ;CHECK R1 FOR 0 CONDITION
2794 016332  100403                        BMI     16$             ; SKIP MULTIPLICATION IF ZERO
2795 016334  060304                        ADD     R3,R4           ;MULTIPLY NUMBER OF INNER
2796 016336  005505                        ADC     R5              ; LOOPS PER OUTER LOOP BY
2797 016340  000773                        BR      14$             ;NUMBER OF OUTER LOOPS PERFORMED.
2798                                ;+
2799                                ; DIVIDE THE TOTAL NUMBER OF INNER LOOPS BY THE NUMBER OF MS PER LTC TICK.
2800                                ;-
2801 016342  016701  163742        16$:    MOV     MSTICK,R1       ;# OF MS PER LTC TICK IS DIVISOR.
2802 016346  010403                        MOV     R4,R3           ;LSWORD OF LOOP COUNT IS LSWORD OF DIVIDEND.
2803 016350  010502                        MOV     R5,R2           ;MSWORD OF LOOP COUNT IS MSWORD OF DIVIDEND.
2804 016352  004767  010076                JSR     PC,UNSDIV       ;DIVIDE NUMBER OF LOOPS BY MS PER LTC TICK.
2805 016356  103402                        BCS     18$             ;BYPASS OOPS IF WE'RE OK.
2806 016360  004767  003532                JSR     PC,OOPS         ;CLOCK ROUTINES ARE NOT LONG ENOUGH, OR BUG.
2807 016364  010167  163722        18$:    MOV     R1,MSLCNT       ;SET NEW VALUE FOR MS LOOP COUNT.
2808 016370  005167  000010                COM     62$             ;SET THE 2ND ITERATION FLAGS IF 1ST ITERATION.
2809 016374  001277                        BNE     2$              ;BRANCH IF ONLY ONE ITERATION DONE.
2810 016376  000261                        SEC                     ;SET THE SUCCESS FLAG FOR EXIT.
2811
2812 016400                        60$:    PASS                    ;RESTORE GPRS.
     016400  004736                                JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
2813 016402  000207                        RTS     PC              ; CARRY - SUCCESS FLAG. SET IF SUCCESS.
2814
2815 016404  000000                62$:    .WORD   0               ;2ND CALIBRATION ITERATION FLAGS.
2816 016406  000000                64$:    .WORD   0               ;DUMMY WORD FOR STORAGE OF THE READ WORD.
```

```
                        .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.
                        ;-- ************************************************************************
016410                  CHKEXT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
016410  004567  166710                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
016414  016302  003402          MOV     RXPTRB(R3),R2   ;GET THE RX DATA POINTER.
016420  005724                  TST     (R4)+           ;INCREMENT R4 BY 2 TO POINT TO CHR1.
016422  012400                  MOV     (R4)+,R0        ;GET CHR1 FROM THE QUE, DATA.VALID INTO N FLAG.
016424  100026                  BPL     52$             ;EXIT WITH "FAILURE" IF CHR1 NOT VALID.
016426  040500                  BIC     R5,R0           ;REMOVE INACTIVE BITS FROM CHR1 VALUE.
016430  112201                  MOVB    (R2)+,R1        ;GET EXP0 FROM THE DATA PATTERN.
016432  040501                  BIC     R5,R1           ;REMOVE INACTIVE BITS FROM EXP0 VALUE.
016434  120100                  CMPB    R1,R0           ;COMPARE CHR1 AND EXP0.
016436  001021                  BNE     52$             ;EXIT WITH "FAILURE" IF CHR1 <> EXP0.
016440  016300  003542          MOV     RXCNTB(R3),R0   ;COMPARE THE PRESENT RX CHARACTER COUNT PLUS 1
016444  005200                  INC     R0              ;  WITH THE EXPECTED NUMBER OF CHARS TO RX ON
016446  016301  005234          MOV     TXRXLB(R3),R1   ;  LINE (NUMBER TRANSMITTED AND LOOPED BACK) TO
016452  020061  003442          CMP     R0,CHCNTB(R1)   ;  DETERMINE IF CHR1 IS LAST EXPECTED CHAR.
016456  001407                  BEQ     50$             ;EXIT WITH "SUCCESS" IF CHR1 IS LAST CHAR.
016460  011400                  MOV     (R4),R0         ;GET CHR2 FROM THE QUE, DATA.VALID INTO N FLAG.
016462  100005                  BPL     50$             ;EXIT WITH "SUCCESS" IF CHR1 WAS LAST IN QUE.
016464  040500                  BIC     R5,R0           ;REMOVE INACTIVE BITS FROM CHR2 VALUE.
016466  111201                  MOVB    (R2),R1         ;GET THE EXP1 VALUE.
016470  040501                  BIC     R5,R1           ;REMOVE INACTIVE BITS FROM EXP1 VALUE.
016472  020001                  CMP     R0,R1           ;COMPARE CHR2 AND EXP1.
016474  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.
                        ;-
```

```
2874 016476  000261                50$:    SEC                     ;SET THE SUCCESS FLAG.
2875 016500  000401                        BR      60$             ;EXIT THE ROUTINE.
2876
2877                               ;+
2878                               ; WE DIDN'T RECEIVE A SINGLE EXTRA CHARACTER AT THIS POINT IN THE DATA PATTERN.
2879                               ; INDICATE "FAILURE" AND EXIT.
2880                               ;-
2881 016502  000241                52$:    CLC                     ;CLEAR THE SUCCESS FLAG.
2882
2883 016504                        60$:    PASS                    ;RESTORE GPRS.
     016504  004736                                JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
2884 016506  000207                        RTS     PC              ;CARRY - SET IF SUCCESS (EXTRA CHAR RXED).
```

M6


```
                        .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.
                        ;-- ************************************************************************
016510                  CHKLOS:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
016510  004567  166610                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
016514  016301  003542          MOV     RXCNTB(R3),R1   ;COMPARE THE PRESENT RX CHARACTER COUNT PLUS 1
016520  005201                  INC     R1              ;  WITH THE EXPECTED NUMBER OF CHARS TO RX ON
016522  016300  005234          MOV     TXRXLB(R3),R0   ;  LINE (NUMBER TXED AND LOOPED BACK) TO
016526  016002  003442          MOV     CHCNTB(R0),R2   ;  DETERMINE IF THE POSSIBLE LOST CHAR
016532  020102                  CMP     R1,R2           ;  WOULD BE THE LAST EXPECTED RX CHAR.
016534  001423                 BEQ     52$             ;EXIT WITH "FAILURE" IF LOST CHR WOULD BE LAST.
016536  005201                  INC     R1              ;DETERMINE (AS ABOVE) IF CHR0 WOULD BE THE LAST
016540  160201                  SUB     R2,R1           ;  RX CHAR AND SAVE RESULT FOR LATER.
016542  016302  003             MOV     RXPTRB(R3),R2   ;GET THE RX DATA POINTER.
016546  005202                  INC     R2              ;CALCULATE POINTER TO EXP1 LOCATION.
016550  112200                  MOVB    (R2)+,R0        ;GET EXP1 VALUE FROM DATA PATTERN.
016552  162400                  SUB     (R4)+,R0        ;COMPARE CHR0 AND EXP1 VALUES.
016554  040500                  BIC     R5,R0           ;REMOVE INACTIVE BITS FROM RESULT.  (NO ACTIVE
                                                        ;  BITS ALLOWED TO LEFT OF ANY INACTIVE BITS.)
016556  001012                 BNE     52$             ;EXIT WITH "FAILURE" IF CHR0 <> EXP1.
016560  005701                  TST     R1              ;CHECK CHR0 TEST RESULT SAVED ABOVE.
016562  001406                 BEQ     50$             ;EXIT WITH "SUCCESS" IF CHR0 IS LAST CHAR.
016564  011401                  MOV     (R4),R1         ;GET CHR1 FROM THE QUE, DATA.VALID INTO N FLAG.
016566  100004                 BPL     50$             ;EXIT WITH "SUCCESS" IF CHR0 WAS LAST QUE CHAR.
016570  111200                  MOVB    (R2),R0         ;GET THE EXP2 VALUE FROM THE DATA PATTERN.
016572  160001                  SUB     R0,R1           ;COMPARE THE EXP2 AND THE CHR1 VALUES.
016574  040501                  BIC     R5,R1           ;REMOVE INACTIVE BITS FROM RESULT OF COMPARE.
                                                        ;  (NO ACTIVE BITS LEFT OF INACTIVE BITS.)
016576  001002                 BNE     52$             ;EXIT WITH "FAILURE" IF CHR1 <> EXP2.
```

```
2942                                    ;+
2943                                    ; IT IS LIKELY THAT WE LOST A CHARACTER FROM THE DATA PATTERN.
2944                                    ; INDICATE "SUCCESS" AND EXIT.
2945                                    ;-
2946 016600  000261                    50$:    SEC                     ;SET THE SUCCESS FLAG.
2947 016602  000401                            BR      60$             ;EXIT THE ROUTINE.
2948
2949                                    ;+
2950                                    ; WE DIDN'T LOSE A SINGLE EXTRA CHARACTER AT THIS POINT IN THE DATA PATTERN.
2951                                    ; INDICATE "FAILURE" AND EXIT.
2952                                    ;-
2953 016604  000241                    52$:    CLC                     ;CLEAR THE SUCCESS FLAG.
2954
2955 016606                             60$:    PASS                    ;RESTORE GPRS.
     016606  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
2956 016610  000207                            RTS     PC              ;CARRY - SET IF SUCCESS (LOST CHAR LIKELY).
```

```
                        .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.
                        ;-- ************************************************************************
016612                  CKCHR:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
016612  004567  166506                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; CHECK FOR THE RX OF A CHAR AFTER RX SHOULD BE COMPLETE ON THIS LINE.
                        ;-
016616  036367  002364  163660           BIT     BITTBL(R3),RXDONF       ;TEST THE RX DONE FLAG FOR THIS LINE.
016624  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"
                        ;-
016626  012701  011351           MOV     #EM9007,R1      ;SELECT "EXTRA CHAR ON LINE" ERROR MESSAGE.
016632  011402                   MOV     (R4),R2         ;GET THE ACTUAL DATA FOR ERROR REPORT.
016634  040502                   BIC     R5,R2           ;REMOVE THE INACTIVE BITS.
016636  052704  100000           BIS     #BIT15,R4       ;INDICATE "NONE" EXPECTED DATA FOR ERROR RPT.
016642  000452                 BR      12$             ;GO COUNT EXTRA CHAR AND EXIT WITH "FAILURE".
                        ;*
                        ; GET THE POINTER TO THE NEXT EXPECTED RECEIVE DATA CHARACTER.
                        ;-
016644  016302  003402  2$:      MOV     RXPTRB(R3),R2
```

```
3014                                    ;*
3015                                    ; COMPARE THE ACTUAL DATA WITH THE EXPECTED DATA.
3016                                    ;
3017 016650  011400                             MOV     (R4),R0         ;GET THE ACTUAL DATA.
3018 016652  040500                             BIC     R5,R0           ;REMOVE THE INACTIVE BITS.
3019 016654  111201                             MOVB    (R2),R1         ;GET THE EXPECTED DATA.
3020 016656  040501                             BIC     R5,R1           ;REMOVE THE INACTIVE BITS.
3021 016660  120001                             CMPB    R0,R1           ;COMPARE ACTUAL AND EXPECTED.
3022 016662  001003                            BNE     4$              ;CHECK FURTHER IF DATA MISCOMPARE.
3023 016664  004767  007720                     JSR     PC,UPDCHR       ;UPDATE PTRS AND COUNTERS FOR THE CHAR.
3024 016670  000446                            BR      50$             ;EXIT WITH "SUCCESS", NO ERROR FOUND.
3025                                    ;*
3026                                    ; ACTUAL AND EXPECTED DATA MISCOMPARE.
3027                                    ; DETERMINE IF IT'S LIKELY WE RECEIVED AN EXTRA CHAR WITHIN THE DATA PATTERN.
3028                                    ;-
3029 016672  004767  177512            4$:     JSR     PC,CHKEXT       ;CHECK FOR EXTRA CHAR RX'ED IN PATTERN.
3030 016676  103010                            BCC     6$              ;GO CHECK FOR LOST CHAR IF NO EXTRA CHAR.
3031                                    ;*
3032                                    ; IT IS LIKELY THAT WE RECEIVED AN EXTRA CHARACTER WITHIN THE DATA PATTERN.
3033                                    ; COUNT THE CHAR AS AN EXTRA CHAR, DON'T COUNT AS A STANDARD CHAR.
3034                                    ; REPORT "EXTRA CHAR RECEIVED WITHIN DATA PATTERN ON LINE NN"
3035                                    ;-
3036 016700  012701  012164                     MOV     #EM9027,R1      ;SELECT "EXTRA CHAR ON LINE" ERROR MSG.
3037 016704  111200                             MOVB    (R2),R0         ;GET THE EXPECTED RECEIVE DATA.
3038 016706  040500                             BIC     R5,R0           ;REMOVE THE INACTIVE BITS FROM EXPECTED DATA.
3039 016710  011402                             MOV     (R4),R2         ;GET THE ACTUAL RECEIVE DATA.
3040 016712  040502                             BIC     R5,R2           ;REMOVE THE INACTIVE BITS FROM ACTUAL DATA.
3041 016714  010004                             MOV     R0,R4           ;PASS EXPECTED DATA TO ERROR REPORT ROUTINE.
3042 016716  000424                            BR      12$             ;GO COUNT EXTRA CHAR AND EXIT WITH "FAILURE".
3043                                    ;*
3044                                    ; ACTUAL AND EXPECTED DATA MISCOMPARE.
3045                                    ; NOT LIKELY THAT WE RECEIVED AN EXTRA CHARACTER WITHIN THE DATA PATTERN.
3046                                    ; DETERMINE IF IT'S LIKELY WE LOST A CHARACTER FROM THE DATA PATTERN.
3047                                    ;-
3048 016720  004767  177564            6$:     JSR     PC,CHKLOS       ;CHECK FOR A LOST CHAR CONDITION.
3049 016724  103012                            BCC     8$              ;GO REPORT BAD RX DATA IF NOT LOST CHAR.
3050                                    ;*
3051                                    ; IT IS LIKELY THAT WE LOST A CHARACTER FROM THE DATA PATTERN.
3052                                    ; COUNT THE CHAR IN THE RX CHAR COUNT AS IF IT HAD BEEN RECEIVED.
3053                                    ; ALSO, COUNT CHR0 AS A VALID CHAR, BECAUSE WE HAVE VERIFIED IT ABOVE.
3054                                    ; REPORT "SINGLE CHAR MISSING FROM RECEIVED DATA ON LINE NN"
3055                                    ;-
3056 016726  012701  012244                     MOV     #EM9028,R1      ;SELECT "LOST CHAR ON LINE" ERROR MSG. +++++
3057 016732  111200                             MOVB    (R2),R0         ;GET THE EXPECTED RECEIVE DATA.
3058 016734  040500                             BIC     R5,R0           ;REMOVE THE INACTIVE BITS FROM EXPECTED DATA.
3059 016736  011402                             MOV     (R4),R2         ;GET THE ACTUAL RECEIVE DATA.
3060 016740  040502                             BIC     R5,R2           ;REMOVE THE INACTIVE BITS FROM ACTUAL DATA.
3061 016742  010004                             MOV     R0,R4           ;PASS EXPECTED DATA TO ERROR REPORT ROUTINE.
3062 016744  004767  007640                    JSR     PC,UPDCHR       ;UPDATE PTRS AND COUNTERS FOR THE CHAR.
3063 016750  000404                            BR      10$             ;GO EXIT WITH "FAILURE".
3064                                    ;*
3065                                    ; DID NOT LOSE OR GAIN A SINGLE CHARACTER FROM/TO THE DATA PATTERN.
3066                                    ; REPORT "RECEIVED CHAR MISCOMPARE AGAINST TX DATA ON LINE NN"
3067                                    ;-
3068 016752  010002                    8$:     MOV     R0,R2           ;PASS ACTUAL DATUM TO ERROR REPORT ROUTINE.
3069 016754  010104                             MOV     R1,R4           ;PASS EXPECTED DATUM TO ERROR REPORT ROUTINE.
3070 016756  012701  011434                     MOV     #EM9008,R1      ;SELECT THE "DATA MISCOMPARE" MESSAGE.
```

```
3071                                    ;+
3072                                    ; UPDATE THE CHARACTER COUNTER AND RX DATA PATTERN POINTER FOR THIS LINE.
3073                                    ;-
3074 016762  004767  007622             10$:    JSR     PC,UPDCHR       ;UPDATE RX PTR AND COUNTER FOR THIS LINE.
3075 016766  000405                             BR      14$             ;GO EXIT WITH "FAILURE".
3076                                    ;+
3077                                    ; COUNT THE CHARACTER AS AN EXTRA CHARACTER.
3078                                    ;-
3079 016770  005263  003242             12$:     INC    EXCNTB(R3)      ;INCREMENT THE EXTRA CHAR COUNT FOR THIS LINE.
3080 016774  001002                            BNE     14$             ;EXIT WITH FAILURE IF NO OVERFLOW.
3081 016776  005363  003242                     DEC    EXCNTB(R3)      ;DECREMENT BACK TO -1 (MAX VALUE) IF OVERFLOW.
3082                                    ;+
3083                                    ; INDICATE "FAILURE" AND EXIT.
3084                                    ;-
3085 017002  000241                     14$:     CLC                    ;CLEAR THE "SUCCESS" FLAG.
3086 017004  000401                             BR      60$             ;EXIT THE ROUTINE.
3087
3088                                    ;+
3089                                    ; NO ERROR WAS FOUND.
3090                                    ; SET "SUCCESS" FLAG AND EXIT.
3091                                    ;-
3092 017006  000261                     50$:     SEC                    ;SET THE "SUCCESS" FLAG.
3093
3094 017010                             60$:    PASS    R1,R2,R4        ;RESTORE GPRS, EXCEPT
     017010  010166  000004                                     MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
     017014  010266  000006                                     MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
     017020  010466  000012                                     MOV     R4,R4SLOT(SP)           ;PUT R4 IN STACK SLOT.
     017024  004736                                             JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
3095                                                                        ;R1 - CONTAINS THE ADDRESS OF THE ERROR REPORT.
3096                                                                        ;R2 - CONTAINS THE ACTUAL DATA RECEIVED.
3097                                                                        ;R4 - CONTAINS THE EXPECTED DATA.
3098 017026  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 WITH 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.
                        ;-- ************************************************************************

017030                  CKFRPR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017030  004567  166270                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017034  016704  166256          MOV     ERRNBR,R4               ;PRESERVE THE INITIAL ERROR NUMBER.
017040  004767  006706          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.
                        ;-
017044  016701  163176          MOV     RXTOUT,R1               ;GET MINIMUM TIME OUT VALUE.
017050  026767  163426  163114  2$:     CMP     TXDONF,ACTLNS   ;CHECK FOR TRANSMISSION DONE ON ACTIVE LINES.
017056  001402          BEQ     4$                              ;SKIP ADDING 50 MS DELAY IF TX DONE ALL LINES.
017060  062701  000062          ADD     #50.,R1                 ;ADD 50 MILLI SEC TO DELAY IF NOT LAST CHAR.
017064  052701  170000  4$:     BIS     #170000,R1              ;INDICATE TO TEST DATA.VALID BIT.
017070  016702  163106          MOV     RBUFA,R2                ;INDICATE TO CHECK DUT RECEIVE BUFFER (FIFO).
017074  004767  010004          JSR     PC,WAIBIS               ;WAIT FOR RECEIVED CHAR OR TIME-OUT.
017100  103054          BCC     60$                             ;EXIT ROUTINE IF TIME-OUT, WE'RE DONE.
017102  005367  163370          DEC     CHRTOT                  ;DECREMENT THE TOTAL CHAR COUNTER.
017106  001014          BNE     6$                              ;SKIP ERROR IF NOT TOO MANY CHARS RECEIVED.
017110  010467  166202          MOV     R4,ERRNBR               ;SET ERROR NUMBER TO INITIAL ERRNBR.
017114  012701  012044          MOV     #EM9025,R1              ;SELECT THE ERROR MESSAGE TO BE REPORTED.
017120  012767  ^^9124  166174  6$:     MOV     #ER0503,ERRBLK  ;SELECT THE ERROR REPORT ROUTINE.
                        ;*
                        ;       REPORT ERROR AT INITIAL ERRNBR.
                        ;       "MORE THAN TWICE THE EXPECTED NUMBER OF CHARACTERS RECEIVED"
                        ;-
017126                  60$:    ERROR                   ;               >>>>> ERROR <<<<<.
017126  104460                                                                    TRAP    C$ERROR
```

F 1


```
3155 017130  012767  000001  163066          MOV     #1,FERROR       ;INDICATE THAT AN ERROR HAS BEEN FOUND.
3156
3157 017136  000435                          BR      60$             ;EXIT THIS ROUTINE WE HAVE GIVEN UP.
3158
3159                                 ;+
3160                                 ; EXTRACT THE LINE NUMBER OF THE NEW CHARACTER.
3161                                 ; CALCULATE OFFSET FOR ACCESSING TABLES OF LINE VARIABLES.
3162                                 ;-
3163 017140  010203                  6$:     MOV     R2,R3           ;COPY THE READ CHARACTER.
3164 017142  000303                          SWAB    R3              ;GET THE LINE NUMBER IN THE LSB.
3165 017144  042703  177760                  BIC     #177760,R3      ;CLEAR THE UNWANTED BITS.
3166 017150  006303                          ASL     R3              ;SHIFT LEFT TO FORM OFFSET INTO TABLES.
3167                                 ;+
3168                                 ; PROCESS THE READ CHARACTERS AS DICTATED BY THE TEST FLAG.
3169                                 ;-
3170 017152  010505                          MOV     R5,R5           ;DETERMIN WHICH TEST CALLED THIS ROUTINE.
3171 017154  100012                  BPL     8$                      ;BRANCH TO PROCESS CHARACTER IN PARITY TEST.
3172
3173 017156  004767  003162          JSR     PC,PRFRME               ;PROCESS FRAMING ERRORS RECEIVED.
3174 017162  005767  163036          TST     FERROR                  ;HAS AN ERROR BEEN DETECTED ?
3175 017166  001416                  BEQ     10$                     ;NO, THEN SKIP PROCESSING CHARACTERS FOR PARITY
3176                                                                 ;TEST.
3177 017170  032767  000100  162764  BIT     #BIT06,OPTION           ;HAS EXTENDED ERROR REPORTING BEEN ENABLED ?
3178 017176  001012                  BNE     10$                     ;BRANCH IF IT HAS,
3179 017200  000414                  BR      60$                     ;OTHERWISE EXIT.
3180
3181 017202  004767  003242          8$:     JSR     PC,PRPARE       ;PROCESS PARITY ERRORS RECEIVED.
3182 017206  005767  163012          TST     FERROR                  ;HAS AN ERROR BEEN DETECTED ?
3183 017212  001404                  BEQ     10$                     ;NO, THEN BRANCH TO UPDATE POINTERS.
3184 017214  032767  000100  162740  BIT     #BIT06,OPTION           ;HAS EXTENDED ERROR REPORTING BEEN ENABLED ?
3185 017222  001403                  BEQ     60$                     ;EXIT IF IT HASN'T.
3186
3187 017224  004767  007360          10$:    JSR     PC,UPDCHR       ;UPDATE POINTERS AND COUNTERS FOR THIS LINE.
3188 017230  000707                  BR      2$                      ;LOOP TO READ NEXT CHAR FROM FIFO.
3189
3190 017232  010467  166060          60$:    MOV     R4,ERRNBR       ;RESTORE THE ERROR NUMBER TO ITS INITIAL VALUE.
3191 017236                          PASS                            ;RESTORE GPRS.
     017236  004736                                      JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3192 017240  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.
                        ;-- ************************************************************************

017242                  CKINAC:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017242  004567  166056                 JSR     R5,PREGO5       ;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.
                        ;-
017246  010203                  MOV     R2,R3           ;EXTRACT THE LINE NUMBER
017250  000303                  SWAB    R3              ;  FROM THE CHARACTER WE
017252  042703  177760          BIC     #177760,R3      ;  ARE COMPARING.
017256  006303                  ASL     R3              ;FORM OFFSET INTO WORD TABLE FROM LINE NUMBER.
                        ;*
                        ; IF THE CHARACTER IN QUESTION IS NOT A VALID CHARACTER, EXIT WITH "SUCCESS".
                        ;-
017260  005702                  TST     R2              ;CHECK DATA.VALID BIT.
017262  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".
                        ;-
017264  016301  005234          MOV     TXRXLB(R3),R1   ;GET THE TX LINE # OFFSET FOR THIS RX LINE.
017270  036167  002364  162674   BIT     BITTBL(R1),ACTLNS  ;DETERMINE IF TX LINE IS AN ACTIVE LINE.
017276  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.
                        ;-
```

```
3250 017300  005263  003242                    INC     EXCNTB(R3)          ;INCREMENT THE EXTRA CHAR COUNT FOR THIS LINE.
3251 017304  001002                           BNE     2$                  ;SKIP SETTING TO MAX VALUE IF NO OVERFLOW.
3252 017306  005363  003242                    DEC     EXCNTB(R3)          ;DECREMENT BACK TO -1 (MAX VALUE) IF OVERFLOW.
3253 017312  012701  011276          2$:       MOV     #EM9006,R1          ;SET UP RX ON INACTIVE LINE MESSAGE.
3254 017316  012704  100000                    MOV     #BIT15,R4           ;SET UP "NONE" EXPECTED DATA INDICATION.
3255 017322  000241                           CLC                         ;CLEAR THE "SUCCESS" FLAG.
3256 017324  000401                           BR      60$                 ;GO REPORT RX CHAR ON INACTIVE LINE.
3257
3258                                ;+
3259                                ; WE HAVE NOT FOUND A "CHAR ON INACTIVE LINE" ERROR SITUATION.
3260                                ; SET THE "SUCCESS" FLAG AND EXIT THE ROUTINE.
3261                                ;-
3262 017326  000261                50$:      SEC                         ;SET THE "SUCCESS" FLAG.
3263
3264 017330                         60$:      PASS    R1,R3,R4            ;RESTORE GPRS, EXCEPT OUTPUT GPRS.
     017330  010166  000004                                    MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
     017334  010366  000010                                    MOV     R3,R3SLOT(SP)           ;PUT R3 IN STACK SLOT.
     017340  010466  000012                                    MOV     R4,R4SLOT(SP)           ;PUT R4 IN STACK SLOT.
     017344  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3265 017346  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 LOCATIO
                                ;*      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.
                                ;*****************************************************************************

017350                          CKTRAP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
017350  004567  165750                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017354  005067  162674                  CLR     TP4FLG                  ;CLEAR THE 004 TRAP FLAGS.
017360  011011                          MOV     (R0),(R1)               ;PERFORM THE MOVE IN QUESTION.
017362  005767  162666          ADRPTR:: TST    TP4FLG                  ;CHECK FOR OCCURENCE OF TRAP.
017366  000261                          SEC                             ;INDICATE SUCCESS.
017370  001401                          BEQ     60$                     ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR.
017372  000241                          CLC                             ;INDICATE FAILURE.
017374                          60$:    PASS                            ;RESTORE GPRS.
017374  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
017376  000207                          RTS     PC
```

```
3300                                    .SBTTL  GOBAL SUBROUTINE                - CKTRPB -
3301                                    ;**************************************************************************
3302                                    ;*                              - CHECK FOR TRAP -
3303                                    ;*      THIS SUBROUTINE IS USED TO CHECK FOR A BUS TIME-OUT TRAP (004 TRAP)
3304                                    ;*      WHICH IS CAUSED BY AN ACCESS TO A NON-EXISTENT MEMORY OR I/O LOCATION
3305                                    ;*      IF A TRAP DOES NOT OCCUR, THIS ROUTINE RETURNS A SUCCESS INDICATION.
3306                                    ;*
3307                                    ;*      INPUTS:         R0 - SOURCE ADDRESS FOR MOVE
3308                                    ;*                      R1 - DESTINATION ADDRESS FOR MOVE
3309                                    ;*                      (R0) - SOURCE FOR THE MOVE
3310                                    ;*
3311                                    ;*      OUTPUTS:        (R1) - WRITEN TO THE CONTENTS OF (R0)
3312                                    ;*                      CARRY FLAG - SET ON RETURN IF NO 004 TRAP DETECTED
3313                                    ;*                      TP4FLG - NONZERO IF TRAP OCCURED, CLEARED OTHERWISE.
3314                                    ;*
3315                                    ;*      CALLING SEQUENCE:       JSR     PC,CKTRPB
3316                                    ;*
3317                                    ;*      COMMENTS:       IF THIS SUBROUTINE CAUSES A TRAP, EITHER THE ADDRESS
3318                                    ;*                      WHICH IS LABELED TRPAD2 WILL BE THE TRAP PC ADDRESS ON
3319                                    ;*                      THE STACK OR SOME OTHER ADDRESS WHICH WAS PLACED ON
3320                                    ;*                      THE STACK BY AN UNEXPECTED TRAP.
3321                                    ;*                              THIS ROUTINE PERFORMS A BYTE MOV .
3322                                    ;*
3323                                    ;*      SUBORDINATE ROUTINES CALLED:    NONE.
3324                                    ;**************************************************************************
3325
3326 017400                             CKTRPB::        SAVE
     017400  004567  165720                                 JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
3327
3328 017404  005067  162644                     CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAGS
3329 017410  111011                             MOVB    (R0),(R1)       ;PERFORM THE BYTE MOVE
3330 017412  005767  162636             TRPAD2:: TST    TP4FLG          ;CHECK FOR OCCURENCE OF TRAP
3331 017416  000261                             SEC                     ;INDICATE SUCCESS
3332 017420  001401                             BEQ     60$             ;EXIT WITH SUCCESS IF TRAP DID NOT OCCUR
3333 017422  000241                             CLC                     ;INDICATE FAILURE
3334 017424                             60$:    PASS                    ;
     017424  004736                                         JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3335 017426  000207                             RTS     PC              ;RETURN
```

K7


```
                                  .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.
                                  ;*************************************************************************

 017430                           CLNRST:: SAVE                         ;SAVE CONTENTS OF GPRS R0 THRU R5.
 017430  004567  165670                            JSR     R5,PREG05        ;CALL REGISTER SAVE SUBRT.
                                  ;+
                                  ; RESET THE DUT.
                                  ;  THIS ROUTINE REPORTS ERRORS WITH NUMBERS FROM ERRNBR THRU ERRNBR+2.
                                  ;-
 017434  004767  004510                   JSR     PC,RESETT             ;RESET THE DUT TO A KNOWN STATE.
 017440  103002                           BCC     60$                   ;EXIT ROUTINE WITH ABORT TEST INDICATOR.
                                  ;+
                                  ; PURGE THE FIFO OF ERROR CODES, SAVE ANY BMP CODES FOUND.
                                  ;-
 017442  004767  003274                   JSR     PC,PUFIFO             ;PURGE THE FIFO.

 017446                           60$:                                  ;EXIT THE TEST USING RESETT OR PUFIFO STATUS.
 017446                                   PASS                          ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
 017446  004736                                    JSR     PC,@(SP)+            ;RETURN TO PREG05 SUBRT.
                                                                        ;CARRY BIT:IF CLEAR, THEN ABORT THE TEST.
 017450  000207                           RTS     PC
```

```
3382                                    .SBTTL  GLOBAL SUBROUTINE                - CLR16W -
3383                                    ;** ************************************************************************
3384                                    ;*                        - CLEAR SIXTEEN WORDS ROUTINE -
3385                                    ;*      THIS SUBROUTINE CLEARS 16 WORDS STARTING WITH THE SPECIFIED WORD.
3386                                    ;*
3387                                    ;* INPUTS:      R0 - ADDRESS OF THE FIRST WORD TO CLEAR.
3388                                    ;*
3389                                    ;* OUTPUTS:     (R0) TO (R0+15) - 16 WORDS OF MEMORY ARE CLEARED TO 0.
3390                                    ;*
3391                                    ;* CALLING SEQUENCE:    JSR     PC,CLR16W
3392                                    ;*
3393                                    ;* COMMENTS:
3394                                    ;*
3395                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
3396                                    ;-- ************************************************************************
3397
3398 017452                            CLR16W:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     017452  004567  165646                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3399 017456  012701  000020                   MOV     #16.,R1                 ;SET THE LOOP COUNTER TO 16.
3400 017462  005020                    2$:     CLR     (R0)+                   ;CLEAR A WORD OF MEMORY.
3401 017464  005301                            DEC     R1                      ;COUNT THIS LOOP.
3402 017466  001375                            BNE     2$                      ;LOOP IF NOT 16 WORD CLEARED.
3403 017470                            60$:    PASS                            ;RESTORE GPRS.
     017470  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
3404 017472  000207                            RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - CONMAP
                                ;++ ************************************************************************
                                ;*                      - CONVERT LINE BIT MAP.
                                ;*      THIS SUBROUITNE 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 ASSOSCIATED 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.
                                ;-- ************************************************************************

017474                          CONMAP::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
017474  004567  165624                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
017500  012702  005234                 MOV     @TXRXLB,R2      ;GET THE BASE ADDRESS OF THE LINE ASSOC TABLE.
017504  010503                         MOV     R5,R3           ;COPY THE BIT MAP TO BE TRANSFORMED.
017506  012704  000020                 MOV     @NUMLNS,R4      ;SET MAX LINE COUNTER.
017512  005005                         CLR     R5              ;CLEAR ASSOCIATED LINE BIT MAP.
017514  006203                 2$:     ASR     R3              ;SHIFT ACTLNS BIT MAP INT BOOLEAN REGISTER.
017516  103005                        BCC     4$              ;SKIP SETTING ASSOCIATED LINE NUMBER BIT MAP.
017520  011201                         MOV     (R2),R1         ;GET ASSOCIATED LINE NUMBER OFFSET FROM TABLE.
017522  006201                         ASR     R1              ;SHIFT RIGHT TO GET LINE NUMB FROM OFFSET.
017524  004767  001250                JSR     PC,LINBIT       ;GENERATE AN SINGLE BIT MAP FOR THIS LINE.
017530  050005                         BIS     R0,R5           ;SET BIT FOR THIS LINE IN ASSOCIATED BIT MAP.
017532  005722                 4$:     TST     (R2)+           ;INCREMENT ADDRESS FOR THE NEXT LINE NUMBER.
017534  005304                         DEC     R4              ;DECREMENT LINE COUNT.
017536  001366                        BNE     2$              ;LOOP IF NOT DONE.
017540                          60$:    PASS    R5              ;RESTORE GPRS, EXCEPT
017540  010566  000014                                 MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
017544  004736                                         JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                                ;R5 - CONTAINS THE ASSOCIATED LINE BIT MAP.
017546  000207                         RTS     PC
```

```
3444                                    .SBTTL  GLOBAL SUBROUTINE                - DELAY
3445                                    ;*****************************************************************************
3446                                    ;*                          - DELAY SUBROUTINE -
3447                                    ;*      THIS SUBROUTINE IS USED TO DELAY A VARIABLE NUMBER OF MILLI-SECONDS.
3448                                    ;*
3449                                    ;* INPUTS:      R4 - CONTAINS THE NUMBER OF MS TO DELAY.
3450                                    ;*                MSLCNT.
3451                                    ;*
3452                                    ;* OUTPUTS:     NONE.
3453                                    ;*
3454                                    ;* CALLING SEQUENCE:    JSR     PC,DELAY
3455                                    ;*
3456                                    ;* COMMENTS:    IF NO HARDWARE CLOCK INTERRUPTS ARE OCCURING, CONTROL-CS WILL
3457                                    ;*              NOT BE HONORED FOR THE DURATION OF THE DELAY.
3458                                    ;*
3459                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
3460                                    ;*****************************************************************************
3461
3462 017550                            DELAY:: SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
     017550  004567  165550                             JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3463 017554  010401                             MOV     R4,R1                   ;PASS NUMBER OF MS DELAY AS TIME-OUT VALUE.
3464 017556  012702  177777                     MOV     #-1,R2                  ;TELL MSLOOP ROUTINE TO CHECK ALL BITS.
3465 017562  005003                             CLR     R3                      ;TELL MSLOOP RTN TO CHECK FOR ALL BITS CLEAR.
3466 017564  012704  017606                     MOV     #62$,R4                 ;TELL MSLOOP TO CHECK DUMMY NON-ZERO WORD.
3467 017570  004767  001506                    JSR     PC,MSLOOP               ;DELAY THE REQUESTED # OF MS.
3468 017574  103002                             BCC     60$                     ;EXIT ROUTINE IF WE TIMED-OUT.]
3469 017576  004767  002314                    JSR     PC,OOPS                 ;IF NO TIME-OUT, BAD PROGRAM OR HOST MACHINE.
3470 017602                            60$:    PASS                            ;RESTORE GPRS.
     017602  004736                                     JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3471 017604  000207                             RTS     PC
3472
3473 017606  177777                    62$:    .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.
                        ;-- ************************************************************************

017610                  DM16B:: SAVE
017610  004567  165510                        JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
017614  016700  162402          MOV     DMTSTA,R0       ;SHIFT THE DMA TEST ADDRESS
017620  012702  000006          MOV     #6,R2           ;SIX PLACES LEFT , TO
017624  006300          2$:     ASL     R0              ;CONVERT IT INTO A ,
017626  005302          DEC     R2              ;16-BIT PHYSICAL ADDRESS
017630  001375          BNE     2$              ;

017632  012701  000052          MOV     #52,R1          ;SET UP THE 6 LSB'S
017636  032700  000100          BIT     #100,R0         ;IF BIT #6 OF THE PHYSICAL
017642  001402          BEQ     4$              ;ADDRESS IS CLEAR THEN BRANCH
017644  012701  000025          MOV     #25,R1          ;OTHERWISE CORRECT THE LSB'S

017650  060100          4$:     ADD     R1,R0           ;MREGE THE LSB'S WITH THE PHY ADDR

017652                          PASS    R0              ;RETURN WITH THE PHY ADDR.
017652  010066  000002          MOV     R0,ROSLOT(SP)           ;PUT R0 IN STACK SLOT.
017656  004736                  JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
017660  000207          RTS     PC
```

```
3510
3511                            .SBTTL  GLOBAL SUBROUTINE                - DMRW -
3512                            ;** ************************************************************************
3513                            ;*                        - READ/WRITE DATA FROM/TO (DMTSTA)
3514                            ;*      THIS ROUTINE READS DATA BYTES FROM OR WRITES DATA BYTES TO AN ADDR OF
3515                            ;*      ALTERNATE 1'S AND 0'S . BITS 21 TO 6 OF THE ADDR ARE CONTAINED AT
3516                            ;*      DMTSTA. THE ROUTINE APPENDS THE 6 LSB'S TO PRODUCE AN ADDR OF
3517                            ;*      ALTERNATE 1'SAND 0'S. THIS ROUTINE IS CALLED FROM THE DMA ADDRESS TEST.
3518                            ;*
3519                            ;* INPUTS:              R0 - ADDRESS OF THE  DATA TO BE WRITTEN TO (DMTSTA),
3520                            ;*                           IF A WRITE IS SPECIFIED.
3521                            ;*                      R1 - ADDRESS OF THE AREA IN WHICH DATA FROM (DMTSTA),
3522                            ;*                           IS TO BE SAVED,IF A READ IS SPECIFIED.
3523                            ;*                      R3 - NUMBER OF DATA BYTES TO BE READ/WRITTEN
3524                            ;*                      R5 - CLEAR , SPECIFIES A READ FROM (DMTSTA)
3525                            ;*                           SET , SPECIFIES A WRITE TO (DMTSTA).
3526                            ;*                      DMTSTA - CONTAINS BITS 21 TO 6 OF THE ADDR.
3527                            ;*                      MMSR0 - ADDRESS OF MEM MGT STATUS REG #0
3528                            ;*                      MMPRES - BIT #0 SET, INDICATES MEM MGT PRESENT
3529                            ;*                      PAR3 - ADDRESS OF MEM MGT PAR #3
3530                            ;*                      TP4FLG - 004 TRAP FLAGS
3531                            ;*
3532                            ;* OUTPUTS:             DATA AT (DMTSTA) SAVED OR WRITTEN
3533                            ;*                      PAR #3 - CONTENTS SET TO CONTENTS OF DMTSTA
3534                            ;*                      TP4FLG - CLEAR IF READ/WRITE SUCCESSFUL
3535                            ;*                               SET IF FAIL.
3536                            ;*
3537                            ;* CALLING SEQUENCE:            JSR     PC,DMRW
3538                            ;*
3539                            ;* COMENTS:             IF MEM MGT IS PRESENT THE SUBROUTINE USES (DMTSTA)
3540                            ;*                      AS THE PAGE ADDRESS , PLACING IT IN PAR #3, AND CREATES
3541                            ;*                      A VIRTUAL ADDR IN THE RANGE OF PAR #3 WHICH CONTAINS
3542                            ;*                      THE SIX LSB'S.
3543                            ;*                      IF IT IS NOT PRESENT THE (DMTSTA) IS CONVERTED INTO
3544                            ;*                      THE EQUIVALENT 16 BIT  PHYSICAL ADDRESS.
3545                            ;*
3546                            ;* SUBORDINATE ROUTINES CALLED: CKTRAP,DM16B.
3547                            ;-- ************************************************************************
3548
3549 017662                     DMRW::  SAVE
     017662  004567  165436                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
3550
3551 017666  010004                     MOV     R0,R4           ;SAVE THE SOURCE ADDR
3552 017670  005767  162424             TST     MMPRES          ;IF MEM MGT IS PRESENT THEN
3553 017674  001003                     BNE     6$              ;JUMP AND SET UP THE PAR #3
3554 017676  004767  177706             JSR     PC,DM16B        ;OTHERWISE CONVERT DMTSTA INTO A 16-BIT
3555                                                             ;PHYSICAL ADDRESS, IN R0.
3556 017702  000416                     BR      10$             ;JUMP TO PERFORM THE MOVE
3557 017704  016777  162312  162420  6$:     MOV     DMTSTA,@PAR3A   ;SET PAR #3
3558 017712  012700  060052             MOV     #60052,R0       ;SET THE SIX LSB'S AND CONVERT TO
3559                                                             ;A VIRTUAL ADDRESS WITHIN THE INFLUENCE
3560                                                             ;OF PAR #3.
3561 017716  032767  000001  162276     BIT     #1,DMTSTA       ;IF BIT #0 OF DMTSTA IS CLEAR THEN
3562 017724  001402                     BEQ     8$              ;AVOID CHANGING THE LSB'S
3563 017726  012700  060025             MOV     #60025,R0       ;CHANGE THE LSB'S
3564 017732  012777  000001  162354  8$:     MOV     #BIT0,@MMSR0    ;ENABLE MEM MGT.
3565 017740  005705                 10$:     TST     R5              ;IF A READ IS SPECIFIED THEN
```

```
3566 017742  001402                        BEQ     12$             ;AVOID SWAPING THE SOURCE AND DESTINATION.
3567 017744  010001                         MOV     R0,R1           ;SWAP
3568 017746  010400                         MOV     R4,R0           ;RESTORE THE ORIGINAL SOURCE FOR THE MOVE.
3569 017750  004767  177424          12$:   JSR     PC,CKTRPB       ;PERFORM THE BYTE MOVE.
3570 017754  103004                        BCC     14$             ;EXIT IF A TRAP OCCURED.
3571 017756  005201                         INC     R1              ;INCREMENT THE DESTINATION ADDRESS
3572 017760  005200                         INC     R0              ;INCREMENT THE SOURCE ADDR.
3573 017762  005303                         DEC     R3              ;DECREMENT THE DATA
3574 017764  001371                        BNE     12$             ;REPEAT UNTIL ALL DATA READ/WRITTEN;
3575 017766  005767  162326          14$:   TST     MMPRES          ;IF MEM MGT IS PRESENT THEN
3576 017772  001402                        BEQ     16$             ;
3577 017774  005077  162314                CLR     @MMSR0          ;DISABLE IT.
3578 020000                         16$:   PASS
     020000  004736                                JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
3579 020002  000207                        RTS     PC
3580
```

```
                        .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.
                        ;*               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 ASSUMES MEMORY MANAGEMENT IS DISABLED AND
                        ;*               CLEARS THE  TWO MSB OF THE DMA ADDRESS, I.E. BITS 0 AND 1
                        ;*               OF THE TBUFFAD2 REG.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

020004                  DODMA:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
020004  004567  165314                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020010  012704  000200          MOV     #200,R4         ;PREPARE TO CLEAR UPPER 6 BITS OF DMA BUFF ADR.

                        ;*
                        ; WRITE THE DMA PARAMETERS OUT TO THE DUT DMA REGISTERS.
                        ;    DISABLE INTERRUPTS.
                        ;    SET UP DUT CSR IND.ADR.REG FIELD.
                        ;    WRITE THE DMA TRANSMIT CHARACTER COUNT.
                        ;    WRITE THE LEAST SIGNIFICANT 16 BITS OF THE DMA BUFFER START ADDRESS.
                        ;    WRITE THE MOST SIGNIFICANT 6 BITS OF THE ADDRESS,
                        ;      SETTING THE DMA_START BIT, AND INITIATING THE DMA TRANSMISSION.
                        ;-
020014                  6$:     GETPRI  R5              ;GET THE PRESENT PROCESSOR PRIORITY.
020014  104440                                                          TRAP    C$GPRI
020016  010005                                                          MOV     R0,R5
020020                          SETPRI  #PRIO7          ;DISABLE ALL HARDWARE INTERRUPTS.
020020  012700  000340                                                  MOV     #PRIO7,R0
020024  104441                                                          TRAP    C$SPRI
020026  056701  162202          BIS     IESTAT,R1       ;PREPARE FOR SETUP OF LINE NUMBER IN DUT CSR.
020032  010177  162142          MOV     R1,@CSRA        ;SET UP THE DUT CSR IND.ADR.REG FIELD.
020036  105777  162152          TSTB    @TXAD2A         ;TEST THE DUT DMA_START BIT.
020042  000241                  CLC                     ;INDICATE FAILURE IN CASE DMA.HO BIT IS SET.
020044  100411                  BMI     60$             ;EXIT WITH FAILURE IF DMA.HO BIT IS SET.
020046  010377  162144          MOV     R3,@TXBFCA      ;WRITE THE DMA CHARACTER COUNT.
020052  010277  162134          MOV     R2,@TXAD1A      ;WRITE THE LS 16 BITS OF BUFFER ADDRESS.
020056  110477  162132          MOVB    R4,@TXAD2A      ;WRITE MS 6 BITS OF ADR AND START DMA TX.
```

```
3634 020062                          SETPRI  R5              ;RESTORE THE PROCESSOR PRIORITY.
     020062  010500                                                      MOV     R5,R0
     020064  104441                                                      TRAP    C$SPRI
3635 020066  000261                   SEC                    ;INDICATE SUCCESS.
3636
3637 020070                    60$:   PASS                   ;RESTORE GPRS,
     020070  004736                          JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
3638 020072  000207                   RTS     PC             ;  CARRY - SUCCESS FLAG (SET IF SUCCESS).
```

```
                        .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.
                        ;-- ************************************************************************

020074                  FINACT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020074  004567  165224                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                            ;*
                            ; FIND AN ACTIVE LINE ON WHICH TO PERFORM THE TEST.
                            ;-
020100  005001                  CLR     R1              ;CLEAR THE LINE NUMBER COUNTER.
020102  012703  000020          MOV     #NUMLNS,R3      ;GET MAX LINE NUMBER.
020106  016700  162060          MOV     ACTLNS,R0       ;GET THE ACTIVE LINE BIT MAP.
020112  012705  000001          MOV     #1,R5           ;SET UP A LINE BIT MASK.
020116  030500          2$:     BIT     R5,R0           ;LOOK FOR AN ACTIVE LINE.
020120  001006                  BNE     4$              ;BRANCH TO BEGIN TEST IF A LINE HAS BEEN FOUND.
020122  006305                  ASL     R5              ;SHIFT THE BIT MASK FOR THE NEXT LINE.
020124  005201                  INC     R1              ;INCREMENT THE LINE NUMBER COUNTER.
020126  020103                  CMP     R1,R3           ;CHECK IF ALL LINES HAVE BEEN TRIED.
020130  002772                  BLT     2$              ;LOOP TO TRY THE NEXT LINE.
020132  000241                  CLC                     ;CLEAR CARRY BIT, NO ACTIVE LINE FOUND.
020134  000401                  BR      60$             ;EXIT WITH FAILURE.
020136  000261          4$:     SEC                     ;SET CARRY, SUCCESS.

020140                  60$:    PASS    R1,R5                   ;RESTORE GPRS, EXCEPT
020140  010166  000004                          MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
020144  010566  000014                          MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
020150  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.
020152  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - FRPSUP -
                        ;** ************************************************************************
                        ;*              - FRAMING AND PARITY ERROR 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 A FRAMING OR PARITY ERROR DETECTION AND
                        ;*              REPORTING TEST.
                        ;*
                        ;* INPUTS:      R0 - LPR CONTENTS FOR LINES IN THE BIT MAP IN GPR4.
                        ;*              R1 - LPR CONTENTS FOR LINES NOT IN THE BIT MAP IN GPR4.
                        ;*              R2 - START ADDRESS OF DATA PATTERN TO TRANSMIT.
                        ;*              R3 - LENGTH OF THE DATA PATTERN TO TX.
                        ;*              R4 - LOCAL LINE GROUP BIT MAP.
                        ;*              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 TXRCB 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,FRPSUP
                        ;*
                        ;* COMMENTS:    THIS ROUTINE SHOULD BE CALLED TWICE DURING THE TESTING OF
                        ;*              THE FRAMING AND PARITY ERROR DETECTION AND REPORTING TEST.
                        ;*              SO THAT BOTH LINE GROUPS ARE TESTED ON TRANSMISSION AND
                        ;*              RECEPTION.
                        ;*                JSR   PC,FRPSUP       ; DO SET-UP.
                        ;*                EXECUTE TEST FOR THE ABOVE SET-UP.
                        ;*                COMPLEMENT THE LINE GROUP BIT MAP.
                        ;*                JSR   PC,FRPSUP       ;DO SET UP AGAIN.
                        ;*                EXECUTE TEST AGAIN.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: TXRINI.
                        ;-- ************************************************************************

020154                  FRPSUP:: SAVE                   ;SAVE THE CONTENTS OF THE GPR'S.
020154  004567  165144                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020160  010067  000230          MOV     R0,70$          ;SAVE LPR PARAMETER FOR LINE TX.
020164  010167  000226          MOV     R1,72$          ;SAVE LPR PARAMETER FOR LINE RX.
                        ;+
                        ;  SET UP THE TRANSMISSION/RECEPTION CONTROL BLOCK TO INITIALISE THE
                        ;  ACTIVE LINES IN THE BIT MAP PASSED INTO THIS ROUTINE.
                        ;-
020170  010067  162726          MOV     R0,CBB          ;SET CONTENTS OF LPR PARAMS IN TX/RX C.BLK.
020174  012700  003124          MOV     #CBB+2,R0       ;GET ADDRESS OF THE NEXT WORD IN THE CNTRL BLK.
020200  012720  000004          MOV     #4,(R0)+        ;LNCTRL PARAMETER, ENABLE RECEIVERS.
020204  010220                  MOV     R2,(R0)+        ;START ADDRESS OF DATA PATTERN.
020206  010320                  MOV     R3,(R0)+        ;SET DATA PATTERN LENGTH.
020210  012720  000001          MOV     #1,(R0)+        ;NUMBER OF DATA PATTERNS TO TRANSMIT.
020214  016710  161752          MOV     ACTLNS,(R0)     ;BIT MAP OF LINES TO INITIALISE.
020220  005104                  COM     R4              ;GENERATE A BIT MAP OF ACTIVE LINES IN GRP1.
020222  040420                  BIC     R4,(R0)+        ;CLEAR THE UNWANTED LINES.
```

```
3739 020224  116720  161744                  MOVB    LOPBCK,(R0)+    ;SET LOOPBACK MODE,STAGGARED.
3740 020230  005200                          INC     R0              ;INCREMENT ADDRESS TO GET NEXT WORD IN TABLE.
3741 020232  012710  000001                  MOV     #1,(R0)         ;SET AMMOUNT OF OFFSET FOR EACH TX START.
3742
3743                                 ;+
3744                                 ; INITIALISE THE DUT AND THE ASSOCIATED POINTERS AND COUNTERS, TO THE STATE
3745                                 ; DICTATED BY THE CONTENTS OF THE TX/RX CONTROL BLOCK.
3746                                 ;-
3747 020236  004767  005534                      JSR     PC,TXRINI       ;INITIALISE DUT.
3748                                ;+
3749                                ; SET UP CONTROL BLOCK FOR LINES IN GROUP 2.
3750                                ;-
3751 020242  012700  003122                  MOV     #CBB,R0         ;GET START ADDRESS OF CONTROL BLOCK.
3752 020246  010120                          MOV     R1,(R0)+        ;SET LPR PARAMETER FOR RX LINES.
3753 020250  062700  000010                  ADD     #10,R0          ;SELECT THE ADDRESS OF THE LINE BIT MAP IN C.B.
3754 020254  016710  161712                  MOV     ACTLNS,(R0)     ;BIT MAP OF LINES TO INITIALISE.
3755 020260  005104                          COM     R4              ;GENERATE A BIT MAP OF LINES IN GRP 2.
3756 020262  040410                          BIC     R4,(R0)         ;CLEAR THE UNWANTED LINES.
3757                                 ;+
3758                                 ; INITIALISE THE DUT AND THE ASSOCIATED POINTERS AND COUNTERS, TO THE STATE
3759                                 ; DICTATED BY THE CONTENTS OF THE TX/RX CONTROL BLOCK.
3760                                 ;-
3761 020264  004767  005506                      JSR     PC,TXRINI       ;INITIALISE DUT.
3762
3763                                ;+
3764                                ; SET-UP THE REQUIRED LPR PARAMETERS NEEDED FOR THE CORRECT RECEPTION OF DATA
3765                                ; ON ASSOCIATED IN-ACTIVE LINES.
3766                                ;-
3767                                 ;+
3768                                 ; INITIALISE LPR PARAMETERS FOR INACTIVE LINES IN GROUP 2.
3769                                 ;-
3770 020270  012701  177777                  MOV     #MAPLNS,R1      ;SET UP BIT MAP CORRESPONDING TO ALL LINES.
3771 020274  016702  161672                  MOV     ACTLNS,R2       ;GET THE ACTIVE (TX) LINE BIT MAP.
3772 020300  005101                          COM     R1              ;GENERATE A BIT MAP OF NONE EXISTANT LINES.
3773 020302  005102                          COM     R2              ;GENERATE A BIT MAP OF INACTIVE LINES.
3774 020304  040102                          BIC     R1,R2           ;CLEAR ANY "NONE EXISTANT" INACTIVE LINES.
3775 020306  040402                          BIC     R4,R2           ;
3776 020310  010267  162620                  MOV     R2,CBMAPA       ;SET UP BIT MAP IN CONTROL BLOCK.
3777 020314  005067  162612                  CLR     CBDPNA          ;CLEAR REPEAT TX COUNT IN CONTROL BLOCK.
3778 020320  016767  000072  162574          MOV     72$,CBLPRA      ;SET-UP COMPLEMENTARY LPR PARM.
3779 020326  004767  005444                  JSR     PC,TXRINI       ;INITIALISE INACTIVE LINES.
3780                                ;+
3781                                ; INITIALISE LPR PARAMETERS FOR INAACTIVE LINES IN GROUP 1.
3782                                ;-
3783 020332  016702  161634                  MOV     ACTLNS,R2       ;GET THE ACTIVE (TX) LINE BIT MAP.
3784 020336  005102                          COM     R2              ;GENERATE A BIT MAP OF INACTIVE LINES.
3785 020340  040102                          BIC     R1,R2           ;CLEAR ANY NONE EXISTANT INACTIVE LINES.
3786 020342  005104                          COM     R4              ;
3787 020344  040402                          BIC     R4,R2           ;ONLY PASS LGRP2 ASSOCIATED LINE BIT MAP.
3788 020346  010267  162562                  MOV     R2,CBMAPA       ;SET-UP BIT MAP IN CONTROL BLOCK.
3789 020352  016767  000036  162542          MOV     70$,CBLPRA      ;SET-UP COMPLAMENTARY LPR PARAM FOR LGRP1.
3790 020360  004767  005412                  JSR     PC,TXRINI       ;INITIALISE INACTIVE LINES IN LGRP1.
3791                                ;+
3792                                ; DISABLE RECEIVERS ON ALL LINES TO ENSURE THAT ONLY THE RECEIVERS OF THE
3793                                ; ASSOCIATED ACTIVE (TX) LINES ARE ENABLED.(STAGGARED LOOPBACK)
3794                                ;  RE-ENABLE RECEPTION ON THE CORRECT ASSOCIATED LINES.
3795                                ;-
```

```
 3796 020364  012705  177777                    MOV     #MAPLNS,R5      ;SET-UP BIT MAP FOR ALL LINES.
 3797 020370  004767  004002                    JSR     PC,RXDSBL       ;DISABLE RX ON ALL LINES.
 3798                                 ;+
 3799                                 ; ENABLE RECEIVERS ON ASSOCIATED (RX) LINES.
 3800                                 ;-
 3801 020374  016705  161572                    MOV     ACTLNS,R5       ;GET ACTIVE (TX) LINE BIT MAP.
 3802 020400  004767  177070                    JSR     PC,CONMAP       ;GENERATE AN ASSOCIATED (RX) LINE BIT MAP.
 3803 020404  004767  004062                    JSR     PC,RXENBL       ;ENABLE RECEIVERS ON ASSOCIATED LINES.
 3804
 3805 020410                          60$:    PASS                    ;RESTORE GRP'S.
      020410  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
 3806 020412  000207                          RTS     PC
 3807 020414  000000                  70$:    .WORD   0               ;LOCAL STORAGE OF LPR PARAMETER TX.
 3808 020416  000000                  72$:    .WORD   0               ;LOCAL STORAGE OF LPR PARAMETER RX.
 3809
```

```
                                .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.
                                ;-- ************************************************************************

020420                          GETBDR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020420  004567  164700                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
020424  016705  161600                  MOV     GMANWD,R5               ;SAVE THE GMAINIX VALUE.
                                ;+
                                ; PROMPT THE OPERATOR: "MODEM BAUDRATE IN BPS:  (D) 1200 ?"
                                ;-
020430  012767  002260  161572  2$:     MOV     #1200.,GMANWD           ;SET UP DEFAULT VALUE TO 1200 BAUD.
020436                          GMANID  BDRMSG,GMANWD,D,177777,0,38400.,YES
020436  104443                                                                  TRAP    C$GMAN
020440  000406                                                                  BR      10000$
020442  002230                                                                  .WORD   GMANWD
020444  000052                                                                  .WORD   T$CODE
020446  013137                                                                  .WORD   BDRMSG
020450  177777                                                                  .WORD   177777
020452  000000                                                                  .WORD   T$LOLIM
020454  113000                                                                  .WORD   T$HILIM
020456                                                                  10000$:
020456  016702  161546                  MOV     GMANWD,R2
                                ;+
                                ; ATTEMPT TO LOOK THE VALUE UP IN THE BAUDRATE TABLE.
                                ;-
020462  012701  000017                  MOV     #15.,R1                 ;INITIALIZE BAUDRATE CODE TO HIGHEST BAUDRATE.
020466  012703  002464                  MOV     #BRTBLE,R3              ;INITIALIZE BAUDRATE POINTER.

020472  020243                  4$:     CMP     R2,-(R3)                ;COMPARE BAUDRATE WITH A TABLE ENTRY.
020474  001416                          BEQ     60$                     ;BAUDRATES COMPARE?  YES, EXIT WITH CODE.
020476  005301                          DEC     R1                      ;NO, SET BAUDRATE CODE TO NEXT LOWER BAUDRATE.
020500  001374                          BNE     4$                      ;DONE?  NO, LOOP.

020502  020243                          CMP     R2,-(R3)                ;CHECK IF LAST BAUDRATE MATCHES.
020504  001412                          BEQ     60$                     ;BAUDRATES MATCH?  YES, EXIT WITH CODE.

                                ;+
                                ; REPORT "NNNNN IS NOT A SUPPORTED BAUDRATE, ENTER ANOTHER OR CTRL C."
                                ;-
020506                                  PRINTF  #UBRFMT,R2
020506  010246                                                                  MOV     R2,-(SP)
```

```
        020510  012746  007501                                                  MOV     #UBRFMT,-(SP)
        020514  012746  000002                                                  MOV     #2,-(SP)
        020520  010600                                                           MOV     SP,R0
        020522  104417                                                           TRAP    C$PNTF
        020524  062706  000006                                                   ADD     #6,SP
  3857  020530  000737                      BR      2$              ;LOOP TO GET ANOTHER BAUDRATE.
  3858
  3859  020532  010567  161472      60$:     MOV     R5,GMANWD       ;RESTORE THE GMANIX PARAMETER VALUE.
  3860  020536                      PASS    R1                      ;RESTORE GPRS, EXCEPT THE FOLLOWING:
        020536  010166  000004                              MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
        020542  004736                              JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
  3861  020544  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:    JSR     PC,GETCHR
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

020546                  GETCHR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020546  004567  164552                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
020552  005000                  CLR     R0                      ;CLEAR THE "RE-ENABLE" TX FLAG (SUBRTN OUTPUT).
020554  005002                  CLR     R2                      ;GET NULL CHAR IN CASE BUFFER IS EMPTY.
020556  005767  162134          TST     RXBCNT                  ;CHECK FOR RX BUFFER EMPTY, CLEAR CARRY.
020562  001416                  BEQ     60$                     ;EXIT THE ROUTINE IF BUFFER IS EMPTY.
020564  016704  162122          MOV     RXBOPT,R4               ;GET THE BUFFER OUTPUT POINTER.
020570  011402                  MOV     (R4),R2                 ;GET A CHARACTER FROM THE BUFFER.
020572  005024                  CLR     (R4)+                   ;DELETE THE READ CHARACTER FROM THE BUFFER.
020574  020427  003120          CMP     R4,#RXBEND              ;CHECK IF POINTER SHOULD WRAP AROUND.
020600  103402                  BLO     2$                      ;SKIP WRAPAROUND IF POINTER IS NOT AT END.
020602  012704  002720          MOV     #RXBSTA,R4              ;WRAP INPUT POINTER AROUND.
020606  010467  162100  2$:     MOV     R4,RXBOPT               ;UPDATE THE OUTPUT POINTER STORAGE.
                        
020612  005367  162100          DEC     RXBCNT                  ;REMOVE THIS CHAR FROM THE BUFFER COUNT.
020616  000261                  SEC                             ;SET SUCCESS FLAG, BUFFER WAS NOT EMPTY.
                        
020620                  60$:    PASS    R2                      ;RESTORE GPRS, EXCEPT
020620  010266  000006                          MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
020624  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
                                                                ;R2 - CONTAINS THE CHARACTER READ FROM BUFFER.
                                                                ;CARRY-"SUCCESS" FLAG, SET IF BUFFER NOT EMPTY.
020626  000207                  RTS     PC
```

```
                        .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.
                        ;-- ************************************************************************

020630                  GETTIM:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
020630  004567  164470                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
020634  000301          SWAB    R1      ;PUT THE BAUD RATE FIELDS IN THE LOW BYTE.
020636  042701  177400  BIC     #177400,R1      ;CLEAR STOP,PARITY,AND CHAR FIELDS.
020642  010102          MOV     R1,R2   ;COPY BAUD RATE FIELDS.
020644  042701  000360  BIC     #360,R1 ;SELECT RX BAUD RATE FIELD ONLY.
020650  006202          ASR     R2      ;SHIFT TX BAUD RATE FIELD
020652  006202          ASR     R2      ; TO OCCUPY THE LOW FOUR BYTES.
020654  006202          ASR     R2      ;
020656  006202          ASR     R2      ;
020660  020102          CMP     R1,R2   ;CHECK IF SAME BAUD RATE IN EACH FIELD.
020662  101401          BLOS    2$      ;BRANCH IF RX BAUD RATE IS LOWER OR SAME.
020664  010201          MOV     R2,R1   ;TX BAUD RATE IS THE SLOWER OF THE TWO.
020666  116102  005214  2$:     MOVB    PROTBL(R1),R2   ;GET PROPORTIONAL DELAY FROM TABLE.
020672  042702  177400  BIC     #177400,R2      ;CLEAR UPPER BYTE BECAUSE OF SIGN EXTENSION.
020676  010267  161344  MOV     R2,RXTOUT       ;LOAD THE RX TIME-OUT VARIABLE.

020702                  60$:    PASS                            ;RESTORE GPRS.
020702  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
020704  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.
                        ;-- ************************************************************************
020706                  INIDMA:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
020706  004567  164412                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
020712  016705  161254          MOV     ACTLNS,R5       ;GET THE ACTIVE LINES BIT MAP.
                        ;*
                        ; SET UP LOOP WHICH HANDLES ONE LINE PER ITERATION.
                        ;-
020716  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.
                            ;-
020720  010104          2$:     MOV     R1,R4           ;CALCULATE AN OFFSET TO THE PROPER LINE
020722  006304                  ASL     R4              ;  ENTRY IN A WORD TABLE (LINE # TIMES 2).
020724  016402  002364          MOV     BITTBL(R4),R2   ;GET A BIT MAP FOR THIS LINE.
020730  030205                  BIT     R2,R5           ;TEST THE ACTIVE LINES BIT FOR THIS LINE.
020732  001414                 BEQ     10$             ;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.
                        ;-
020734  016403  003202          MOV     DPLENB(R4),R3   ;GET DATA PATTERN LENGTH FOR THIS LINE.
020740  016402  003342          MOV     TXPTRB(R4),R2   ;PREPARE TO PASS DATA PATTERN ADR TO DODMA RTN.
                            ;*
                            ; WRITE DMA PARAMETERS TO THE DUT.
                            ;-
```

```
4004 020744  004767  177034                JSR     PC,DODMA
4005 020750  103403                        BCS     6$              ;SKIP ERROR IF DODMA WAS SUCCESSFUL.
4006                                    ;+
4007                                    ; SET THE PROPER BIT OF THE TX INTERRUPT FLAGS TO INDICATE THE LINE ERROR.
4008                                    ;-
4009 020752  050567  161306                 BIS     R5,TXINTF       ;INDICATE THE ERROR.
4010 020756  000402                        BR      10$             ;SKIP UPDATING POINTERS AND COUNTERS.
4011                                    ;+
4012                                    ; UPDATE THE TX CHARACTER COUNT FOR THIS LINE.
4013                                    ;-
4014 020760  060364  003502         6$:     ADD     R3,TXCNTB(R4)   ;ADD THE DATA PATTERN LENGTH TO TX CHAR COUNT.
4015                                    ;+
4016                                    ; INCREMENT LINE COUNTER,GOTO NEXT LINE IF NOT DONE.
4017                                    ;-
4018 020764  005201                 10$:    INC     R1              ;INCREMENT THE LINE COUNTER.
4019 020766  020127  000020                CMP     R1,@NUMLNS      ;COMPARE THE LINE COUNTER WITH NUMBER OF LINES.
4020 020772  002752                        BLT     2$              ;LOOP TO SEND CHAR TO ANOTHER LINE IF NOT DONE.
4021
4022 020774                         60$:    PASS                    ;RESTORE GPRS.
     020774  004736                                JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
4023 020776  000207                        RTS     PC
```

D()

```
                                .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.
                                ;-- ************************************************************************

021000                          LINBIT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021000  004567  164320                         JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
021004  042701  177760                 BIC     #177760,R1              ;MASK OUT ALL BUT 4 LSBITS OF THE LINE #.
021010  006301                         ASL     R1                      ;MULTIPLY LINE # BY 2 TO GET WORD TABLE OFFSET.
021012  016100  002364                 MOV     BITTBL(R1),R0           ;GET THE SINGLE BIT BIT MAP.
021016                          60$:    PASS    R0                      ;RESTORE GPRS, EXCEPT THE FOLLOWING.
021016  010066  000002                         MOV     R0,ROSLOT(SP)           ;PUT R0 IN STACK SLOT.
021022  004736                         JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
021024  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.
                        ;-- ************************************************************************

021026                  MODSUP:: SAVE                   ;SAVE CONTENTS OF THE GPR'S R0 THRU R5.
021026  004567  164272                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
021032  005067  161440          CLR     CHRTOT          ;CLEAR TOTAL RECEIVED CHAR COUNTER.
021036  005067  161222          CLR     TXINTF          ;CLEAR FLAGS USED TO LOG DMA H.OVER ERRORS.
021042  005067  161434          CLR     TXDONF          ;CLEAR THE TX DONE FLAGS.
021046  005067  161432          CLR     RXDONF          ;CLEAR THE RX DONE FLAGS.
                        ;*
                        ;  SET UP THE TRANSMISSION/RECEPTION CONTROL BLOCK TO THE DESIRED STATE.
                        ;-
021052  010167  162044          MOV     R1,CBB          ;SET CONTENTS OF LPR PARAMS IN TX/RX C.BLK.
021056  012701  003122          MOV     #CBB,R1         ;GET BASE ADDRESS OF CONTROL BLOCK.
021062  005201                  INC     R1              ;INCREMENT ADDRESS FOR NEXT WORD
021064  005201                  INC     R1              ;INITIALISE THE FOLLOWING IN THE CNTRL.BLK:
021066  012721  011004          MOV     #11004,(R1)+    ; LNCTRL: RTS, DTR, ENABLE RECEIVERS.
021072  010221                  MOV     R2,(R1)+        ; START ADDRESS OF DATA PATTERN.
021074  010321                  MOV     R3,(R1)+        ; DATA PATTERN LENGTH.
021076  012721  000001          MOV     #1,(R1)+        ; NUMBER OF DATA PATTERNS TO TRANSMIT.
021102  016721  161064          MOV     ACTLNS,(R1)+    ; BIT MAP OF LINES TO INITIALISE.
021106  112721  000003          MOVB    #3,(R1)+        ;SET LOOPBACK MODE TO H325.
021112  005201                  INC     R1              ;INCREMENT ADDRESS FOR THE NEXT WORD.
021114  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.
                        ;-
021120  004767  004652          JSR     PC,TXRINI       ;INITIALISE DUT.
                        ;*
```

```
4112                                    ; INITIALISE POINTERS AND COUNTERS FOR INACTIVE LINES TO ZERO.
4113                                    ;-
4114 021124  012701  177777                     MOV     #MAPLNS,R1      ;GET THE LINE BIT MAP FOR ALL LINES.
4115 021130  016702  161036                     MOV     ACTLNS,R2       ;GET THE ACTIVE LINE BIT MAP.
4116 021134  005101                             COM     R1              ;
4117 021136  005102                             COM     R2              ;
4118 021140  040102                             BIC     R1,R2           ;GENERATE AN IN-ACTIVE LINE BIT MAP.
4119 021142  010267  161766                     MOV     R2,CBMAPA       ;MOVE BIT MAP TO THE CONTROL BLOCK.
4120 021146  005067  161752                     CLR     CBLNCA          ;CLEAR THE LNCTRL SET UP PARAMETERS.
4121 021152  005067  161754                     CLR     CBDPNA          ;CLEAR THE REPEAT TX COUNT IN CNTRL BLCK.
4122 021156  004767  004614                     JSR     PC,TXRINI       ;SET UP PARAMETERS FOR INACTIVE LINES.
4123
4124 021162                             60$:    PASS                    ;RESTORE GPR'S.
     021162  004736                                     JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
4125 021164  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.
                        ;*************************************************************************

021166                  MSLGET:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021166  004567  164132                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; SET UP MASK FOR REMOVING UNUSED BITS IN THE TEST WORD, AND CLEAR UNUSED
                        ; BITS IN THE DESIRED STATE WORD TO ALLOW DIRECT COMPARISON.
                        ;-
021172  005102                  COM     R2              ;GET MASK OF UNUSED BITS.
021174  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.
                        ;-
021176  005701                  TST     R1              ;TEST THE TIME-OUT VALUE FOR ZERO.
021200  001011                  BNE     2$              ;IF NON-ZERO TIME-OUT, GO LOOP AND TEST.
021202  011400                  MOV     (R4),R0         ;GET THE WORD TO TEST BEFORE EXITING.
021204  010067  000070          MOV     R0,62$          ;SAVE VALUE SO WE CAN RETURN IT.
021210  040200                  BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
021212  020003                  CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
021214  000261                  SEC                     ;INDICATE SUCCESS IN CASE WORDS ARE EQUAL.
021216  001420                  BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
```

```
4183 021220  000241                    CLC                     ;INDICATE FAILURE (TIME-OUT).
4184 021222  000416                    BR      6$              ;EXIT WITH FAILURE, WORDS AREN'T EQUAL.
4185                                ;+
4186                                ; NON-ZERO TIME-OUT VALUE.  LOOP, WAITING FOR CONDITION OR TIME-OUT.
4187                                ;-
4188 021224  016705  161062        2$:     MOV     MSLCNT,R5       ;LOAD MS LOOP COUNT.
4189 021230  011400                4$:     MOV     (R4),R0         ;GET THE WORD TO TEST.
4190 021232  010067  000042                MOV     R0,62$          ;SAVE WORD IN CASE THIS IS THE LAST.
4191 021236  040200                        BIC     R2,R0           ;MASK OUT UNTESTED BITS OF WORD.
4192 021240  020003                        CMP     R0,R3           ;COMPARE AGAINST DESIRED STATE WORD.
4193 021242  000261                        SEC                     ;SET CARRY IN CASE OF SUCCESS.
4194 021244  001405                        BEQ     6$              ;EXIT WITH SUCCESS IF WORDS ARE EQUAL.
4195 021246  005305                        DEC     R5              ;COUNT DOWN THE INSIDE MS LOOP COUNT.
4196 021250  001367                        BNE     4$              ;LOOP IF MS NOT UP.
4197 021252  005301                        DEC     R1              ;DECREMENT THE MS TIME COUNT.
4198 021254  001363                        BNE     2$              ;IF TIME NOT UP, LOOP TO COUNT ANOTHER MS.
4199 021256  000241                        CLC                     ;CLEAR CARRY, WE TIMED-OUT.
4200                                ;+
4201                                ; HAVE EITHER FOUND CONDITION, OR TIMED-OUT (POSSIBLY FROM 0 TIME-OUT VALUE).
4202                                ; RESTORE THE LAST CONTENTS READ FROM THE TEST WORD.  EXIT ROUTINE.
4203                                ;-
4204 021260  016700  000014        6$:     MOV     62$,R0          ;PASS OUT THE LAST READ WORD.
4205 021264                        60$:    PASS    R0,R1                   ;RESTORE GPRS, EXCEPT THE FOLLOWING:
     021264  010066  000002                                MOV     R0,R0SLOT(SP)           ;PUT R0 IN STACK SLOT.
     021270  010166  000004                                MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
     021274  004736                                        JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
4206                                                                ;R0 - LAST READ WORD CHECKED FOR CONDITION.
4207                                                                ;R1 - REMAINING TIME (0 IF TIME-OUT OCCURED).
4208 021276  000207                        RTS     PC              ;CARRY - SET IF SUCCESS, CLEAR IF TIME-OUT.
4209                                ;+
4210                                ; LOCAL STORAGE.
4211                                ;-
4212 021300  000000                62$:    .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.
                        ;**************************************************************************

021302                  MSLOOP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021302  004567  164016                          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.
                        ;-
021306  004767  177654          JSR     PC,MSLGET       ;CALL THE MULTI-PURPOSE MS LOOP AND SEARCH RTN.

021312                  60$:    PASS                            ;RESTORE GPRS,
021312  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
021314  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.
                        ;-- ************************************************************************

021316                  MSSRPT:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
021316  004567  164002                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;*
                        ; PRINT THE BASIC MODEM STATUS MESSAGE.
                        ;    "MODEM STATUS SIGNAL REPORT:"
                        ;-
                                PRINTF  #MSFMT1
021322                  
021322  012746  007577                                                  MOV     #MSFMT1,-(SP)
021326  012746  000001                                                  MOV     #1,-(SP)
021332  010600                                                          MOV     SP,R0
021334  104417                                                          TRAP    C$PNTF
021336  062706  000004                                                  ADD     #4,SP

                                CLR     R1              ;START WITH LINE 0.
                                MOV     #1,R2
                                MOV     CSRA,R3         ;GET THE CSR ADDRESS.
                                MOV     IESTAT,R4       ;GET THE STATES OF THE INTERRUPT ENABLE BITS.
                                MOV     ACTLNS,R5       ;GET THE ACTIVE LINES BIT MAP.

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

                                MOV     R4,R0           ;SET UP DUT CSR IND.ADR.REG FIELD
                                BIS     R1,R0           ;  LEAVING THE INTERRUPT ENABLE
                                MOV     R0,(R3)         ;  BITS IN THE SPECIFIED STATE.
                                MOV     @FSLSA,R0       ;READ THE DUT STATUS REG FOR THIS LINE.
                                SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
                                                JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                                CLR     R2              ;CLEAR THE SIGNAL STATUS INDICATORS.
                                CLR     R3
                                CLR     R4
                                CLR     R5
                                ASL     R0              ;SHIFT DSR INTO CARRY,
                                ROL     R2              ;  THEN ROTATE INTO INDICATOR.
```

```
4308 021422  006300                        ASL     R0              ;SHIFT BLANK SLOT INTO CARRY,
4309 021424  006300                        ASL     R0              ;  SHIFT RI INTO CARRY,
4310 021426  006103                        ROL     R3              ;  THEN ROTATE INTO INDICATOR.
4311 021430  006300                        ASL     R0              ;SHIFT DCD INTO CARRY,
4312 021432  006104                        ROL     R4              ;  THEN ROTATE INTO INDICATOR.
4313 021434  006300                        ASL     R0              ;SHIFT CTS INTO CARRY,
4314 021436  006105                        ROL     R5              ;  THEN ROTATE INTO INDICATOR.
4315                                ;+
4316                                ; PRINT THE STATUS FOR THIS LINE.
4317                                ;    "LINE #N:  DSR=N, RI=N, DCD=N, CTS=N"
4318                                ;-
4319 021440                                PRINTF  #MSFMT2,R1,R2,R3,R4,R5
     021440  010546                                                                MOV     R5,-(SP)
     021442  010446                                                                MOV     R4,-(SP)
     021444  010346                                                                MOV     R3,-(SP)
     021446  010246                                                                MOV     R2,-(SP)
     021450  010146                                                                MOV     R1,-(SP)
     021452  012746  007637                                                        MOV     #MSFMT2,-(SP)
     021456  012746  000006                                                        MOV     #6,-(SP)
     021462  010600                                                                MOV     SP,R0
     021464  104417                                                                TRAP    C$PNTF
     021466  062706  000016                                                        ADD     #16,SP
4320 021472                                PASS                    ;RESTORE ALL THE GPRS.
     021472  004736                                JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
4321
4322 021474  006302                  4$:    ASL     R2              ;SHIFT LINE BIT MAP TO NEXT LINE.
4323 021476  005201                         INC     R1              ;INCREMENT THE LINE COUNTER.
4324 021500  020127  000020                 CMP     R1,#NUMLNS      ;CMP LINE COUNTER WITH # OF LINES ON DEVICE.
4325 021504  002727                        BLT     2$              ;ALL LINES DONE?  NO, LOOP TO DO NEXT LINE.
4326
4327 021506                                PRINTF  #EF9101         ;PRINT A BLANK LINE.
     021506  012746  C07262                                                        MOV     #EF9101,-(SP)
     021512  012746  000001                                                        MOV     #1,-(SP)
     021516  010600                                                                MOV     SP,R0
     021520  104417                                                                TRAP    '$PNTF
     021522  062706  000004                                                        ADD     #4,SP
4328
4329 021526                        60$:    PASS                    ;RESTORE GPRS.
     021526  004736                                JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
4330 021530  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.
                        ;-- ************************************************************************

021532                  MUL16U:: SAVE                           ;SAVE CONTENTS OF GPRS R0  HRU R5.
021532  004567  163566                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
021536  005003                  CLR     R3              ;CLEAR THE PRODUCT.
021540  005702                  TST     R2              ;CHECK THE MULTIPLIER.
021542  001003                 BNE     2$              ;GO TO DO MULTIPLICATION IF NOT ZERO.
021544  005001                  CLR     R1              ;RETURN A PRODUCT OF ZERO.
021546  000261                  SEC                     ;INDICATE SUCCESS.
021550  000412                 BR      60$             ;EXIT THE ROUTINE.

021552  060103          2$:      ADD     R1,R3           ;ADD THE MULTIPLICAND TO THE PRODUCT.
021554  103405                 BCS     50$             ;EXIT WITH OVERFLOW IF ONE OCCURRED.
021556  005302                  DEC     R2              ;DECREMENT THE MULTIPLIER.
021560  001374                 BNE     2$              ;LOOP IF MULTIPLIER NOT ZERO.
021562  010301                  MOV     R3,R1           ;PREPARE TO PASS OUT THE PRODUCT.
021564  000261                  SEC                     ;INDICATE SUCCESS.
021566  000403                 BR      60$             ;EXIT WITH SUCCESS.

021570  012701  177777  50$:     MOV     #-1,R1          ;FORCE PRODUCT TO MAX VALUE, WE OVERLFOWED.
021574  000241                  CLC                     ;INDICATE FAILURE.

021576                  60$:    PASS    R1                      ;RESTORE GPRS, EXCEPT THE FOLLOWING:
021576  010166  000004                         MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
021602  004736                               JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                ;  R1 - PRODUCT (16 BIT UNSIGNED).
021604  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.
                                ;*              NOERPT - 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.
                                ;-- ******************************************************************

021606                          NEWCHR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
 021606  004567  163512                          JSR     R5,PREG05      ;CALL REGISTER SAVE SUBRT.
021612  010305                          MOV     R3,R5                   ;GET THE BIT MAP OF INACTIVE DATA BYTE BITS.
021614  052705  177400                  BIS     #177400,R5              ;ALL UPPER BITS OF EXPECTED DATA ARE INACTIVE.
021620  005067  000270                  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.
                                ;-
021624  004767  175412                  JSR     PC,CKINAC               ;CHECK FOR CHAR ON INACTIVE LINE.
021630  103052                          BCC     4$                      ;GO REPORT ERROR IF ON INACTIVE LINE.
                                ;+
                                ; PUSH THE NEW CHARACTER ON THE RESYNC QUE FOR THIS LINE.
                                ;-
021632  010304                          MOV     R3,R4                   ;CALCULATE BASE ADDRESS OF THE
021634  006304                          ASL     R4                      ; DATA PATTERN RESYNCH QUEUE
021636  006304                          ASL     R4                      ; (QUEUE IS 4 WORDS LONG) FOR
```

```
4433 021640  062704  004642                    ADD     #DPRSQB,R4      ;  THIS LINE.
4434 021644  010401                            MOV     R4,R1           ;GET THE BASE OF THE QUEUE.
4435 021646  016121  000002                    MOV     2(R1),(R1)+     ;MOVE FROM CHR1 SLOT TO CHR0 SLOT.
4436 021652  016121  000002                    MOV     2(R1),(R1)+     ;MOVE FROM CHR2 SLOT TO CHR1 SLOT.
4437 021656  010211                            MOV     R2,(R1)         ;PUT NEW CHAR INTO CHR2 SLOT.
4438                                    ;+
4439                                    ; CHECK THE DATA.VALID FOR THE CHARACTER AT THE BOTTON OF THE QUEUE.
4440                                    ; IF DATA.VALID IS CLEAR, EXIT THE ROUTINE--NOTHING TO ANALYZE.
4441                                    ;-
4442 021660  011402                            MOV     (R4),R2         ;GET CHR0 VALUE, SET FLAGS.
4443 021662  100112                           BPL     60$             ;EXIT ROUTINE IF DATA.VALID IS CLEAR.
4444                                    ;+
4445                                    ; TEST FOR ANY OF THE ERROR BITS SET IN CHR0.
4446                                    ;-
4447 021664  032702  070000                    BIT     #70000,R2       ;TEST FOR ANY CHR0 ERROR BITS SET.
4448 021670  001427                           BEQ     2$              ;SKIP THIS ERROR IF NO ERROR BITS SET.
4449                                    ;+
4450                                    ; WE HAVE AT LEAST ONE ERROR FLAG SET ON THE RECEIVED CHAR.
4451                                    ; REPORT DATA ERROR FLAG ERROR IF NOT IN SUMMARY MODE.
4452                                    ;-
4453 021672  005367  000216                    DEC     70$             ;SET THE "ERROR FOUND" FLAG.
4454 021676  016300  005234                    MOV     TXRXLB(R3),R0   ;GET THE TX LINE OFFSET FOR THIS RX LINE.
4455 021702  036067  002364  160570            BIT     BITTBL(R0),ERSMRF  ;CHECK THE ERROR SUMMARY FLAG FOR TX LINE.
4456 021710  001017                           BNE     2$              ;IF ERROR SUMMARY FLAG SET, SKIP NEXT REPORT.
4457 021712  012767  014770  163402            MOV     #ER9003,ERRBLK  ;SELECT THE ER9003 ERROR REPORT ROUTINE.
4458 021720  004767  004326                    JSR     PC,TXROFF       ;TURN OFF TX AND RX DURING ERROR REPORTING.
4459 021724                                    ERROR                   ;                >>>>> ERROR <<<<<.
     021724  104460                                                                                  TRAP    C$ERROR
4460 021726  012767  000001  160270            MOV     #1,FERROR       ;INDICATE AN ERROR HAS BEEN FOUND.
4461 021734  032767  000100  160220            BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
4462 021742  001462                           BEQ     60$             ;EXIT IF IT HASN'T.
4463
4464 021744  004767  004342                    JSR     PC,TXRON        ;TURN TX AND RX BACK ON.
4465                                    ;+
4466                                    ; CHECK THE CHARACTER AT THE BOTTOM OF THE RESYNC QUE FOR DATA ERRORS.
4467                                    ;-
4468 021750  004767  174636            2$:     JSR     PC,CKCHR        ;CHECK THE CHR0 CHAR FOR ERRORS.
4469 021754  103433                           BCS     6$              ;SKIP ERROR REPORT IF CHR0 IS CORRECT.
4470                                    ;+
4471                                    ; WE HAVE SOME SORT OF DATA ERROR SO REPORT IT (UNLESS IN SUMMARY REPORT MODE).
4472                                    ;-
4473 021756  005367  000132            4$:     DEC     70$             ;SET THE "ERROR FOUND" FLAG.
4474 021762  016300  005234                    MOV     TXRXLB(R3),R0   ;GET THE TX LINE OFFSET FOR THIS RX LINE.
4475 021766  036067  002364  160504            BIT     BITTBL(R0),ERSMRF  ;CHECK THE ERROR SUMMARY FLAG FOR THIS LINE.
4476 021774  001023                           BNE     6$              ;SKIP ERROR REPORT IF ERROR SUMMARY FLAG SET.
4477 021776  012767  014612  163316            MOV     #ER9002,ERRBLK  ;SELECT THE ER9002 ERROR REPORT ROU INE.
4478 022004  005267  163306                    INC     ERRNBR          ;SELECT INITIAL ERRNBR + 1.
4479 022010  004767  004236                    JSR     PC,TXROFF       ;TURN OFF TX AND RX DURING ERROR REPORTING.
4480 022014                                    ERROR                   ;                >>>>> ERROR <<<<<.
     022014  104460                                                                                  TRAP    C$ERROR
4481 022016  012767  000001  160200            MOV     #1,FERROR       ;INDICATE AN ERROR HAS BEEN FOUND.
4482 022024  032767  000100  160130            BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
4483 022032  001426                           BEQ     60$             ;EXIT IF IT HASN'T.
4484
4485 022034  004767  004252                    JSR     PC,TXRON        ;TURN TX AND RX BACK ON.
4486 022040  005367  163252                    DEC     ERRNBR          ;RESTORE INITIAL ERRNBR.
4487                                    ;+
```

```
4488                                    ; COUNT A CHARACTER ERROR IF ONE OCCURRED.
4489                                    ; UPDATE THE "REPORT ERROR SUMMARY" FLAG FOR LINE BASED ON ERROR COUNT.
4490                                    ;
4491 022044  005767  000044            6$:     TST     70$             ;CHECK THE "ERROR FOUND' FLAG.
4492 022050  001417                            BEQ     60$             ;SKIP COUNTING AN ERROR IF FLAG IS CLEAR.
4493 022052  005263  003302                     INC     ERCNTB(P3)      ;INCREMENT THE ERROR COUNTER FOR THIS LINE.
4494 022056  001002                            BNE     8$              ;SKIP SETTING COUNTER TO MAX IF NO OVERFLOW.
4495 022060  005363  003302                     DEC     ERCNTB(R3)      ;RESET THE ERROR COUNTER TO -1 (MAX VALUE).
4496 022064  005767  160074            8$:     TST     NDERPT          ;DISABLE ERROR SUMMARY FUNCTION IF
4497 022070  001407                            BEQ     60$             ; NUMBER OF DATA ERRORS TO REPORT IS 0.
4498 022072  026367  003302  160064            CMP     ERCNTB(R3),NDERPT  ;COMPARE ERROR COUNT WITH # OF ERR'S TO RPT.
4499 022100  103403                            BLO     60$             ;SKIP SETTING OF SUMMARY FLAG IF NOT TOO MANY.
4500 022102  056367  002364  160370            BIS     BITTBL(R3),ERSMRF  ;SET "PRINT ERROR SUMMARY" FLAG FOR LINE.
4501
4502 022110                            60$:    PASS                    ;RESTORE GPRS.
     022110  004736                                    JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
4503 022112  000207                            RTS     PC
4504
4505 022114  000000            70$:    .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.
                                ;-- ************************************************************************

022116                          OOPS::  SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
022116  004567  163202                        JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
                                    ; REPORT "HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED." ERROR.
022122                              ERRSF   101,EM0101
022122  104454                                                                  TRAP    C$ERSF
022124  000145                                                                  .WORD   101
022126  022162                                                                  .WORD   EM0101
022130  000000                                                                  .WORD   0
                                    ; REPORT "PROGRAM HUNG, WAITING FOR A CONTROL-C."
022132                              PRINTF  @EM0102
022132  012746  022246                                                          MOV     @EM0102,-(SP)
022136  012746  000001                                                          MOV     #1,-(SP)
022142  010600                                                                  MOV     SP,R0
022144  104417                                                                  TRAP    C$PNTF
022146  062706  000004                                                          ADD     #4,SP
022152                          2$:     BREAK                   ;LOOK FOR OPERATOR CONTROL C INPUT.
022152  104422                                                                  TRAP    C$BRK
022154  000776                          BR      2$              ;INFINITE LOOP.
022156                          60$:    PASS                    ;DON'T NEED THIS, BUT SOMEBODY MAY CHANGE THIS
022156  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
022160  000207                          RTS     PC              ; ROUTINE IN THE FUTURE, SO BE CONSISTANT.

022162      110     117     123 EM0101:: .ASCIZ /HOST COMPUTER HARDWARE OR SOFTWARE BUG ENCOUNTERED./
022165      124     040     103
022170      117     115     120
022173      125     124     105
022176      122     040     110
022201      101     122     104
022204      127     101     122
022207      105     040     117
022212      122     040     123
022215      117     106     124
022220      127     101     122
022223      105     040     102
022226      125     107     040
022231      105     116     103
022234      117     125     116
022237      124     105     122
```

```
        022242  105     104     056
        022245  000
4537    022246  045     116     045     EM0102:: .ASCIZ /%N%APROGRAM HUNG, WAITING FOR A CONTROL-C. <**************%N%N/
        022251  101     120     122
        022254  117     107     122
        022257  101     115     040
        022262  110     125     116
        022265  107     054     040
        022270  127     101     111
        022273  124     111     116
        022276  107     040     106
        022301  117     122     040
        022304  101     040     103
        022307  117     116     124
        022312  122     117     114
        022315  055     103     056
        022320  040     074     052
        022323  052     052     052
        022326  052     052     052
        022331  052     052     052
        022334  052     052     052
        022337  045     116     045
        022342  116     000
4538                                    .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.
                        ;-- ************************************************************************

022344                  PRFRME::SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
022344  004567  162754                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022350  016704  162742          MOV     ERRNBR,R4               ;SAVE THE CONTENTS OF THE INITIAL ERROR NUMBER.
022354  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.
                        ;-
022356  012767  014254  162736          MOV     #ER6201,ERRBLK  ;SET UP THE ADDRESS OF THE ERROR ROUTINE.
022364  032702  020000          BIT     #BIT13,R2               ;CHECK ON STATE OF THE FRAMING ERROR BIT.
022370  001002                  BNE     6$                      ;BRANCH IF FRAMING ERROR BIT SET.
022372  052705  000002          BIS     #BIT1,R5                ;SET REPORT FRAMING ERROR FLAG.

022376  032702  010000  6$:     BIT     #BIT12,R2               ;CHECK ON THE STATE OF THE PARITY ERROR BIT.
022402  001402                  BEQ     8$                      ;BRANCH IF PARITY ERROR BIT CLEAR.
022404  052705  000014          BIS     #14,R5                  ;SET REPORT "PARITY ERROR SET" FLAGS.
022410  005705          8$:     TST     R5                      ;CHECK IF ANY ERROR FLAGS SET.
022412  001412                  BEQ     60$                     ;EXIT IF ALL FLAGS CLEAR.
022414  036367  002364  160056          BIT     BITTBL(R3),ERSMRF  ;CHECK THE ERROR SUMMARY FLAG FOR THIS LINE.
022422  001004                  BNE     10$                     ;SKIP ERROR REPORT IF ERROR SUMMARY FLAG SET.

                        ;REPORT ERROR "CHARACTER RECEIVED WITH PARITY/FRAMING ERROR BIT SET".
022424                  ERROR                                   ;                  >>>>> ERROR <<<<<.
022424  104460                                                                          TRAP    C$ERROR

022426  012767  000001  157570          MOV     #1,FERROR       ;INDICATE AN ERROR HAS BEEN DETECTED.

022434  005263  003302  10$:     INC     ERCNTB(R3)              ;INCREMENT ERROR COUNT FOR THIS LINE.
022440  010467  162652  60$:     MOV     R4,ERRNBR               ;RESTORE ERROR NUMBER.
022444                  PASS                                    ;RESTORE GPRS.
022444  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
```

```
4594 022446  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.
                        ;*              FERROR - "AT LEAST ONE ERROR FOUND" FLAG.
                        ;*
                        ;* 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.
                        ;-- ************************************************************************

022450                  PRPARE::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
022450  004567  162650          JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
022454  016746  162636          MOV     ERRNBR,-(SP)    ;SAVE THE CONTENTS OF THE INITIAL ERROR NUMBER.
022460  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.
                        ;-
022462  012767  014254  162632          MOV     #ER6201,ERRBLK  ;SET UP THE ADDRESS OF THE ERROR ROUTINE.
022470  032702  010000                  BIT     #BIT12,R2       ;CHECK ON STATE OF THE PARITY ERROR BIT.
022474  001002                          BNE     6$              ;BRANCH IF PARITY ERROR BIT SET.
022476  052705  000010                  BIS     #BIT3,R5        ;SET REPORT PARITY ERROR FLAG.
022502  032702  020000          6$:     BIT     #BIT13,R2       ;CHECK ON THE STATE OF THE FRAMMING ERROR BIT.
022506  001402                          BEQ     8$              ;BRANCH IF FRAMMING ERROR BIT CLEAR.
022510  052705  000003                  BIS     #3,R5           ;SET REPORT "FRAMMING ERROR SET" FLAGS.
022514  005705                  8$:     TST     R5              ;CHECK IF ANY ERROR FLAGS SET.
022516  001414                          BEQ     12$             ;BRANCH TO MAKE DATA CHECK IF ALL FLAGS CLEAR.
022520  036367  002364  157752          BIT     BITTBL(R3),ERSMRF  ;CHECK THE ERROR SUMMARY FLAG FOR THIS LINE.
022526  001035                          BNE     14$             ;SKIP ALL ERROR REP IF IN ERROR SUMMARY MODE.
                                        ;REPORT ERROR "CHAR RECEIVED WITH PARITY/FRAMMING ERROR BIT SET/CLEAR".
022530                                  .ERROR                  ;                >>>>> ERROR <<<<<.
022530  104460                                                                          TRAP    C$ERROR

022532  012767  000001  157464          MOV     #1,FERROR       ;INDICATE AN ERROR HAS BEEN FOUND.
022540  032767  000100  157414          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
022546  001440                          BEQ     18$             ;EXIT IF IT HASN'T.

                        ;*
```

```
4651                                    ; COMPARE ACTUAL DATA WITH EXPECTED DATA TO CHECK FOR MULTIPLE ERRORS.
4652                                    ;
4653 022550  005267  162542        12$:     INC     ERRNBR          ;INCREMENT ERROR NUMBER.
4654 022554  016304  003402             MOV     RXPTRB(R3),R4   ;GET THE POINTER TO THE EXPECTED DATA.
4655 022560  111404                      MOVB    (R4),R4         ;GET THE EXPECTED DATA.
4656 022562  042704  177400             BIC     #177400,R4      ;CLEAR THE UPPER BYTE.
4657 022566  120204                      CMPB    R2,R4           ;COMPARE ACTUAL AND EXPECTED DATA.
4658 022570  001427                    BEQ     18$             ;SKIP ERROR REPORT IF DATA CORRECT.
4659 022572  042704  100000             BIC     #BIT15,R4       ;CLEAR "NONE" EXPECTED MESSAGE FLAG.
4660 022576  036367  002364  157674     BIT     BITTBL(R3),ERSMRF  ;CHECK THE ERROR SUMMARY FLAG FOR THIS LINE.
4661 022604  001017                    BNE     16$             ;SKIP ERROR REPORT IF ERROR SUMMARY FLAG SET.
4662 022606  036367  002364  157670     BIT     BITTBL(R3),RXDONF  ; CHECK FOR RECEPTION COMPLETE ON THIS LINE.
4663 022614  001402                    BEQ     14$             ;SKIP SETTING NONE EXPECTED FLAG.
4664 022616  052704  100000             BIS     #BIT15,R4       ;SET "NONE" EXPECTED MESSAGE FLAG.
4665 022622  012701  011434        14$:     MOV     #EM9008,R1      ;SELECT ERROR MESSAGE TO BE REPORTED.
4666 022626  012767  014612  162466     MOV     #ER9002,ERRBLK  ;SELECT ERROR REPORTING ROUTINE.
4667                                    ;REPORT ERROR"RECEIVE CHARACTER MISCOMPARE"
4668 022634                             ERROR
     022634  104460                                                                    TRAP    C$ERROR
4669 022636  012767  000001  157360     MOV     #1,FERROR       ;INDICATE AN ERROR HAS BEEN FOUND.
4670
4671
4672 022644  005263  003302        16$:     INC     ERCNTB(R3)      ;INCREMENT ERROR COUNT FOR THIS LINE.
4673 022650  012667  162442        18$:     MOV     (SP)+,ERRNBR    ;RESTORE ERROR NUMBER.
4674
4675 022654                        60$:     PASS                    ;RESTORE GPRS.
     022654  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
4676 022656  000207                         RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - PRTLPR -
                        ;** ************************************************************************
                        ;*                      -PRINT THE CONTENTS OF THE LPR.
                        ;*      THIS ROUTINE IS USED TO PRINT OUT EXTENDED INFORMATION ON THE
                        ;*      CONTENTS OF THE LINE PARAMETER REGISTER (LPR).
                        ;*
                        ;* INPUTS:      R3 - CONTAINS THE NUMBER OF THE LINE YOU WISH TO EXAMINE.
                        ;*              CSRA - CONTAINS THE ADDRESS OF THE DUT'S CSR.
                        ;*              IESTAT - CONTAINS THE CURRENT STATUS OF THE TX AND RX INTERRUPT
                        ;*                       ENABLE BITS IN THE DUT'S CSR.
                        ;*              LPRA - CONTAINS THE ADDRESS OF THE DUT'S LPR REGISTER.
                        ;*
                        ;* OUTPUTS:     AN EXTENDED INFORMATION MESSAGE IS PRINTED ON THE OPERATORS
                        ;*              CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,PRTLPR
                        ;*
                        ;* COMMENTS:    THIS ROUTINE CHANGES THE INDIRECT ADDRESS FIELD OF THE DEVICE
                        ;*              UNDER TEST'S CSR.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: NONE.
                        ;-- ************************************************************************

022660                  PRTLPR::SAVE                            ;SAVE CONTENTS OF GPRS R0 THRU R5.
022660  004567  162440                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022664  016701  157310          MOV     CSRA,R1                 ;GET THE CSR ADDRESS.
022670  016702  157310          MOV     LPRA,R2                 ;GET THE LPR ADDRESS.
022674  042703  177760          BIC     #177760,R3              ;CLEAR ANY UNWANTED BITS.
022700  056703  157330          BIS     IESTAT,R3               ;SET STATE OF TX AND RX INTERRUPT ENABLE BITS.
022704  010311                  MOV     R3,(R1)                 ;SELECT LINE.
022706  011204                  MOV     (R2),R4                 ;GET CONTENTS OF THE LPR.
                               ;PRINT MESSAGE"CONTENTS OF THE LPR:NNNNNN"
022710                          PRINTX #EF9019,#EM9026,R4;PRINT OUT MESSAGE ON OPERATORS CONSOLE.
022710  010446                                                          MOV     R4,-(SP)
022712  012746  012140                                                  MOV     #EM9026,-(SP)
022716  012746  007162                                                  MOV     #EF9019,-(SP)
022722  012746  000003                                                  MOV     #3,-(SP)
022726  010600                                                          MOV     SP,R0
022730  104415                                                          TRAP    C$PNTX
022732  062706  000010                                                  ADD     #10,SP
022736                  60$:    PASS                            ;RESTORE GPRS.
022736  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
022740  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.
                        ;*****************************************************************************

022742                  PUFIFO::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
022742  004567  162356                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
022746  012701  001000          MOV     #512.,R1        ;SET MAXIMUM TRY COUNT OF 512.
022752  016704  157224          MOV     RBUFA,R4        ;GET ADDRESS OF THE RECEIVER BUFFER REGISTER.

022756  011402                  2$:     MOV     (R4),R2         ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
022760  100016                  BPL     6$              ;EXIT IF THE FIFO IS EMPTY, DATA_VALID CLR.
                        ;*
                        ; CHECK IF THE READ CHARACTER IS ACTUALLY A BMP CODE.
                        ;  IF IT IS, THEN SAVE IT ON THE BMP CODE QUEUE TO BE REPORTED LATER.
                        ;-
022762  012700  070000          MOV     #70000,R0       ;GENERATE A BIT MAP OF CHAR ERROR BITS
022766  040200                  BIC     R2,R0           ; WHICH ARE NOT SET FOR CHAR.
022770  001006                  BNE     4$              ;THROW CHAR AWAY IF NOT BMP OR SELFTEST CODE.
                          ;*
                          ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
                          ;-
022772  012700  000301          MOV     #301,R0         ; CHECK IF BMP.
022776  040200                  BIC     R2,R0           ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
023000  001002                  BNE     4$              ;IF IT IS MODEM OR SELFTEST CODE THROW IT AWAY.
023002  004767  001644          JSR     PC,SAVBMP       ;SAVE BMP CODE ON THE QUEUE.

023006  005301                  4$:     DEC     R1              ;DECREMENT THE TRY COUNT.
023010  001362                  BNE     2$              ;LOOP TO TRY AGAIN.
023012  000241                  CLC                     ;CLEAR CARRY,TO INDICATE FIFO NOT PURGED.
023014  000401                  BR      60$             ;EXIT WITH CARRY CLEAR.
023016  000261                  6$:     SEC             ;SET CARRY, TO INDICATE FIFO PURGED.

023020                  60$:    PASS                    ;RESTORE GPRS,
023020  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                        ;CARRY BIT, SET INDICATES FIFO PURGED.
023022  000207                  RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - PUFIFR -
                                ;*************************************************************************
                                ;*      - PURGE FIFO REPORT ANY ERRORS FOUND.
                                ;*      THIS ROUTINE REMOVES ALL DATA FROM THE FIFO.  ANY BMP CODES THAT ARE
                                ;*      FOUND ARE SAVE ON THE QUEUE TO BE REPORTED LATER IN THE BMP REPORT TEST.
                                ;*      ANY UNXEPECTED DATA (IE ANY NON-STATUS INFORAMTION) THAT ARE FOUND,
                                ;*      ARE REPORTED AS AN ERROR.
                                ;*      IF THE FIFO WILL NOT PURGE AFTER 512 ATTEMPTS, THEN THE CURRENT TEST
                                ;*      THAT CALLED THIS ROUTINE RECEIVES A FAILURE FLAG THAT SHOULD BE USED
                                ;*      TO ABORT THE TEST.
                                ;*
                                ;* INPUTS:      ERRTBL - ERRTYPE, ERRMSG, ERRNBR ARE SET UP CORRECTLY.
                                ;*              RBUFA- CONTAINS THE  ADDRESS OF THE RECEIVER.
                                ;*
                                ;* OUTPUTS:     CARRY BIT - ABORT TEST FLAG, CLR = ABORT TEST, SET = OK.
                                ;*              ERRBLK - VALUE WILL BE DASTROYED.
                                ;*              BMPCQP - THE BMP CODE QUEUE POINTER MAY BE UPDATED.
                                ;*              THE CONTENTS OF THE BMP CODE QUEUE MAY BE UDATED.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,PUFIFR
                                ;*
                                ;* COMMENTS:    THIS ROUTINE REPORTS ERRORS WITH NUMBERS INITIAL ERRNBR
                                ;*              THRU TO ERRNBR+2.
                                ;*              THE ERRNBR IS RESTORED TO ITS INITIAL VALUE BEFORE RETURNING.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: ER1603,ER9001,ER9002,SAVBMP.
                                ;*************************************************************************

023024                          PUFIFR::SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
023024  004567  162274                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
023030  016746  162262                  MOV     ERRNBR,-(SP)    ;SAVE THE CONTENTS OF THE ERROR NUMBER.
023034  012705  001000                  MOV     #512.,R5        ;SET MAXIMUM READ COUNTER TO 2*FIFO SIZE.
                                ;*
                                ; READ DATA FROM THE FIFO UNTIL DATA VALID IS CLEAR OF READ COUNTER IS ZERO.
                                ;  REPORT ANY BMP OR UNEXPECTED DATA AS ERRORS.
                                ;-
023040  017702  157136          2$:     MOV     @RBUFA,R2       ;GET THE CONTENTS OF THE RECEIVER BUFFER REG.
023044  100063                  BPL     8$                      ;EXIT IF DATA VALID CLEAR, IE. FIFO PURGED.
                                ;*
                                ; CHECK IF READ DATA IS STATUS OR UNEXPECTED CHARACTER.
                                ;-
023046  012700  070000                  MOV     #70000,R0       ;GENERATE A BIT MAP OF CHAR ERROR BITS
023052  040200                  BIC     R2,R0                   ; WHICH ARE NOT SET FOR CHAR.
023054  001012                  BNE     4$                      ;SKIP BMP CHECK IF IT IS UNEXPECTED DATA.
                                ;*
                                ; CHECK IF THE READ DATA IS MODEM STATUS , BMP OR SELFTEST?.
                                ;  IF IT IS A BMP CODE THEN SAVE IT ON THE QUEUE.
                                ;-
023056  012767  014512  162236          MOV     #ER9001,ERRBLK  ;SET UP THE CORRECT ERROR REPORTING ROUTINE.
023064  012700  000300                  MOV     #300,R0         ; CHECK IF BMP OR SELFTEST?.
023070  040200                  BIC     R2,R0                   ;TRY TO CLEAR BMP FLAGS IN THE READ DATA.
023072  001003                  BNE     4$                      ;SKIP BMP ERROR REPORT IF MODEM OR SELFTEST?.
023074  004767  001552                  JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
023100  000430                  BR      6$                      ;BRANCH TO CHECK READ COUNT.
                                ;*
                                ; CHECK IF THE READ DATA IS MODEM, SELFTEST OR UNEXPECTED DATA.
                                ;-
```

```
4819 023102  032702  000001          4$:     BIT     #BIT0,R2        ;TEST THE MODEM STATUS INDICATION BIT.
4820 023106  001425                          BEQ     6$              ;DO NOT REPORT ANY ERROR IF MODEM STATUS.
4821 023110  012701  012531                  MOV     #EM9104,R1      ;PASS THE CORRECT ERROR MESSAGE TO REPORT.
4822 023114  010203                          MOV     R2,R3           ;EXTRACT THE LINE NUMBER FROM
4823 023116  000303                          SWAB    R3              ; THE READ DATA.
4824 023120  042703  177760                  BIC     #177760,R3      ;
4825 023124  006303                          ASL     R3              ;FORM LINE NUMBER TIMES 2 FOR ER9002 RCUTINE.
4826 023126  052704  100000                  BIS     #BIT15,R4       ;SET THE "NONE" EXPECTED MAESSAGE FLAG.
4827 023132  005267  162160                  INC     ERRNBR          ;SET ERROR NUMBER TO INTIAL ERRBR+1.
4828 023136  012767  014612  162156          MOV     #ER9002,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
4829                                         ;REPORT ERROR "UNEXPECTED DATA FOUND IN FIFO".
4830 023144                                  ERROR                   ;               >>>>> ERROR <<<<<.
     023144  104460                                                                      TRAP    C$ERROR
4831                                     ;+
4832                                     ; EXIT WITH FAILURE IF EXTENDED ERROR REPORTING HAS NOT BEEN ENABLED
4833                                     ;-
4834 023146  032767  000100  157006          BIT     #BIT06,OPTION   ;EXIT WITH TEST FAILURE MESSAGE IF
4835 023154  001415                          BEQ     7$              ;NO EXTENDED ERROR REPORTING HAS BEEN REQUESTED
4836                                                                 ;DURING THE SOFTWARE QUESTIONS.
4837
4838 023156  005367  162134                  DEC     ERRNBR          ;RESTORE ERROR NUMBER TO INTIAL ERRNBR.
4839
4840 023162  005305                  6$:     DEC     R5              ;DECREMENT READ COUNTER.
4841 023164  001325                          BNE     2$              ;LOOP TO READ NEXT CHAR FROM FIFO IF COUNT > 0.
4842                                     ;+
4843                                     ; THE FIFO WILL NOT CLEAR, REPORT THE ERROR AND INDICATE THAT THE TEST IS TO
4844                                     ; BE ABORTED.
4845                                     ;-
4846 023166  062767  000002  162122          ADD     #2,ERRNBR       ;SET ERROR NUMBER TO INTIAL ERRNBR+2.
4847 023174  012767  014162  162120          MOV     #ER1603,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE.
4848 023202  012701  011733                  MOV     #EM9017,R1      ;PASS THE MESSAGE TO BE REPORTED.
4849                                         ;REPORT THE ERROR "FIFO WILL NOT PURGE, (DATA VALID STUCK SET)"
4850                                         ;                       "?????? TEST ABORTED".
4851 023206                                  ERROR                   ;               >>>>> ERROR <<<<<.
     023206  104460                                                                      TRAP    C$ERROR
4852 023210  000241                  7$:     CLC                     ;INDICATE THE TEST IS TO BE ABORTED.
4853 023212  000401                          BR      10$             ;EXIT THIS ROUTINE AND ABORT THE CURRENT TEST.
4854
4855 023214  000261                  8$:     SEC                     ;SET THE CARRY, DO NOT ABORT THE TEST.
4856
4857 023216  012667  162074          10$:    MOV     (SP)+,ERRNBR    ;RESTORE INITIAL ERROR NUMBER.
4858 023222                          60$:    PASS                    ;RESTORE GPRS,
     023222  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
4859                                                                 ;CARRY BIT, SET INDICATES FIFO PURGED, DO NOT
4860                                                                 ; ABORT THE TEST.
4861 023224  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.
                                ;-- ************************************************************************

023226                          PURRXB:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023226  004567  162072                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.

023232  012701  002712                  MOV     #RXBOPT,R1      ;GET THE ADDRESS OF THE RX OUTPUT POINTER.
023236  012721  002720                  MOV     #RXBSTA,(R1)+   ;INITIALIZE THE RX BUFFER OUTPUT POINTER.
023242  012721  002720                  MOV     #RXBSTA,(R1)+   ;INITIALIZE THE RX BUFFER INPUT POINTER.
023246  005021                  2$:     CLR     (R1)+           ;CLEAR CHAR COUNT AND THE BUFFER AREA.
023250  020127  003120                  CMP     R1,#RXBEND      ;CHECK IF LAST LOCATION HAS BEEN CLEARED.
023254  101774                          BLOS    2$              ;LOOP IF NOT DONE.

023256                          60$:    PASS                            ;RESTORE GPRS.
023256  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
023260  000207                          RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - RDCHRS -
                        ;++ ************************************************************************
                        ;*              - READ AND COMPARE INPUT CHARACTERS ROUTINE -
                        ;*      THIS SUBROUTINE READS THE CHARACTERS FROM THE RX BUFFER IN MEMORY.
                        ;*      IF CHARACTERS STOP APPEARING IN THE BUFFER WITH DATA.VALID SET
                        ;*      OR IF MORE THAN THE ALLOWABLE NUMBER OF CHARACTERS HAS BEEN READ FROM
                        ;*      THE BUFFER THIS ROUTINE EXITS WITH AN RX COMPLETE INDICATION.
                        ;*      EACH READ CHAR IS ANALYZED AND ANY NECESSARY ERRORS ARE REPORTED.
                        ;*
                        ;* INPUTS:      ACTLNS - BIT MAP OF THE ACTIVE DUT LINES.
                        ;*              ERRNBR - SET TO ERROR NUMBER OF FIRST ERROR IN THIS ROUTINE.
                        ;*              IBM - MASK OF THE INACTIVE BITS IN A TX OR RX CHAR BYTE.
                        ;*              OSTEND - ADDRESS OF THE END OF THE OUTPUT STORAGE FIFO BUFFER.
                        ;*              OSTPTR - POINTER TO THE NEXT BYTE TO READ FROM OSTORE.
                        ;*              RXBOPT - POINTER INTO THE RX CHAR BUFFER IN MEMORY.
                        ;*              RXTOUT - TIME-OUT VALUE FOR RX OF LAST CHAR.
                        ;*
                        ;* OUTPUTS:     ERROR MESSAGES MAY BE PRINTED AT THE OPERATOR'S CONSOLE.
                        ;*              TXDBLF - TX/RX DISABLED FLAG (CLEARED).
                        ;*              TXENBM - TX.ENABLE STATE MASK (DESTROYED).
                        ;*              SAVPRI - STORAGE FOR PROCESSOR PRIORITY (DESTROYED).
                        ;*              SAVTEN - STORAGE FOR TX.ENABLE STATES (DESTROYED).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RDCHRS
                        ;*
                        ;* COMMENTS:    THIS ROUTINE REPORTS ERRORS WITH NUMBERS INITIAL ERRNBR
                        ;*              THRU INITIAL ERRNBR + 4.
                        ;*              ERRNBR IS RESTORED BEFORE THIS ROUTINE RETURNS.
                        ;*
                        ;* SUBROUTINES CALLED:  CKCHR,NEWCHR,REPCOD,RXIE0,RXIE1,TXENBL,TXIE0,TXIE1,
                        ;*                      WAIBIS.
                        ;-- ************************************************************************

023262                  RDCHRS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023262  004567  162036                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023266  016704  162024          MOV     ERRNBR,R4               ;PRESERVE THE INITIAL ERROR NUMBER.
023272  016703  156734          MOV     IBM,R3                  ;GET THE INACTIVE BIT MASK.
023276  005067  157204          CLR     TXDBLF                  ;CLEAR THE TX DISABLED FLAG.
023302  004767  001320          JSR     PC,RXIE1                ;TURN ON DUT RECEPTION INTERRUPTS.
023306  004767  002440          JSR     PC,TXIE1                ;TURN ON DUT TRANSMISSION INTERRUPTS.

                        ;+
                        ; CLEAR ALL RESYNC QUEUES FOR ALL LINES.
                        ;-
023312  012701  004642          MOV     #DPRSQB,R1              ;GET BASE ADDRESS OF RESYNC QUEUES TABLE.
023316  012702  005042          MOV     #DPRSQE,R2              ;GET END ADDRESS OF RESYNC QUEUES TABLE.
023322  005021          2$:     CLR     (R1)+                   ;CLEAR A WORD OF THE TABLE.
023324  020102                  CMP     R1,R2                   ;CHECK IF POINTER AT END OF TABLE.
023326  103775                  BLO     2$                      ;LOOP UNTIL TABLE IS CLEAR.

                        ;+
                        ; WAIT FOR A CHARACTER TO APPEAR IN THE FIFO.
                        ; IF NO CHARACTER APPEARS WITHIN TIME-OUT PERIOD: EXIT ROUTINE, WE'RE DONE.
                        ;-
023330  016701  156712  4$:     MOV     RXTOUT,R1               ;GET TIME-OUT FOR SLOWEST BAUD RATE IN USE.
023334  026767  157142  156630  CMP     TXDONF,ACTLNS           ;CHECK FOR TRANSMISSION DONE ON ACTIVE LINES.
023342  001402                  BEQ     6$                      ;SKIP ADDING 50 MS DELAY IF TX DONE ALL LINES.
023344  062701  000062          ADD     #50.,R1                 ;ADD 50 MILLI SEC TO DELAY IF NOT LAST CHAR.
```

```
4952 023350  052701  170000          6$:     BIS     #170000,R1      ;INDICATE TO TEST DATA.VALID BIT.
4953 023354  016702  157332                  MOV     RXBOPT,R2       ;INDICATE TO CHECK MEMORY RECEIVE BUFFER.
4954 023360  004767  003520                  JSR     PC,WAIBIS       ;WAIT FOR RECEIVED CHAR OR TIME-OUT.
4955 023364  103117                          BCC     18$             ;EXIT ROUTINE IF TIME-OUT, WE'RE DONE.
4956
4957 023366  004767  175154                  JSR     PC,GETCHR       ;READ A CHARACTER FROM THE MEMORY BUFFER.
4958                                  ;+
4959                                  ; CHECK IF THE TX ISR IS DISABLED.
4960                                  ; RE-ENABLE RX ISR IF THE SPACE FOR NEW CHARS IS LOW ENOUGH.
4961                                  ;  IF THE BUFFER CAN ACCOMODATE MORE CHARS THEN RE-ENABLE TRANSMISSION.
4962                                  ;-
4963 023372  005767  157110          8$:     TST     TXDBLF          ;CHECK IF TX IS DISABLED.
4964 023376  100027                          BPL     10$             ;SKIP RX/TX CHECK IF TX NOT DISABLED.
4965 023400  026727  157312  000020          CMP     RXBCNT,#RXBETX  ;COMPARE BUFFER COUNT WITH LEVEL TO ENABLE RX.
4966 023406  101023                          BHI     10$             ;SKIP ENABLE RX IF BUFFER TOO FULL.
4967 023410  004767  001212                  JSR     PC,RXIE1        ;ENABLE RECEPTION INTERRUPTS.
4968 023414  016705  156642                  MOV     TXENBM,R5       ;GET THE PRESERVED TX.ENABLE STATES.
4969 023420  026727  157272  000020          CMP     RXBCNT,#RXBETX  ;COMPARE BUFFER COUNT WITH LEVEL TO ENABLE TX.
4970 023426  101013                          BHI     10$             ;SKIP ENABLING TX IF BUFFER TOO FULL.
4971 023430                                  GETPRI  R1              ;SAVE THE CURRENT PROCESSOR PRIORITY.
     023430  104440                                                                  TRAP    C$GPRI
     023432  010001                                                                  MOV     R0,R1
4972 023434                                  SETPRI  #PRI07          ;DISABLE INTERRUPTS.
     023434  012700  000340                                                          MOV     #PRI07,R0
     023440  104441                                                                  TRAP    C$SPRI
4973 023442  004767  002054                  JSR     PC,TXENBL       ;ENABLE TRANSMISSION.
4974 023446  005067  157034                  CLR     TXDBLF          ;CLEAR THE TX DISABLE FLAG.
4975 023452                                  SETPRI  R1              ;RE-ENABLE INTERUPTS.
     023452  010100                                                                  MOV     R1,R0
     023454  104441                                                                  TRAP    C$SPRI
4976 023456                          10$:
4977
4978 023456  005367  157014                  DEC     CHRTOT          ;DECREMENT THE TOTAL CHAR COUNTER.
4979 023462  001014                          BNE     12$             ;SKIP ERROR IF NOT TOO MANY RECEIVED.
4980 023464  010467  161626                  MOV     R4,ERRNBR       ;SET ERROR NUMBER TO INITIAL ERRNBR.
4981 023470  012701  012044                  MOV     #EM9025,R1      ;SELECT THE PROPER ERROR MESSAGE.
4982 023474  012767  014124  161620          MOV     #ER0503,ERRBLK  ;SELECT THE PROPER ERROR REPORT ROUTINE.
4983                                  ;+
4984                                  ; REPORT ERROR AT INITIAL ERRNBR.
4985                                  ; "MORE THAN TWICE THE EXPECTED NUMBER OF CHARACTERS RECEIVED."
4986                                  ;-
4987 023502                                  ERROR                   ;               >>>>> ERROR <<<<<.
     023502  104460                                                                  TRAP    C$ERROR
4988 023504  012767  000001  156512          MOV     #1,FERROR       ;INDICATE THAT AN ERROR HAS BEEN FOUND.
4989
4990 023512  000477                          BR      60$             ;EXIT THE ROUTINE, WE'RE GIVING UP.
4991                                  ;+
4992                                  ; DETERMINE IF THE CHARACTER IS DATA OR A STATUS CODE.
4993                                  ;-
4994 023514  012700  070000          12$:    MOV     #70000,R0       ;GENERATE A BIT MAP OF CHARACTER ERROR BITS
4995 023520  040200                          BIC     R2,R0           ;  WHICH ARE NOT SET FOR THE CHARACTER.
4996 023522  001016                          BNE     14$             ;SKIP REPORTING OF ERROR CODE IF WE HAVE CHAR.
4997                                  ;+
4998                                  ; THE DATA IS EITHER A BMP CODE OR A MODEM STATUS CODE.
4999                                  ; REPORT THAT THE CODE WAS FOUND.
5000                                  ; ERRORS REPORTED WITH ERROR NUMBERS >>>>> ERRNBR+1 AND ERRNBR+2 <<<<<.
5001                                  ;-
```

```
5002 023524  010467  161566                MOV     R4,ERRNBR       ;GET THE ERROR NUMBER PASSED INTO THIS ROUTINE.
5003 023530  005267  161562                INC     ERRNBR          ;SET ERROR NUMBER TO INITIAL ERRNBR+1.
5004 023534  004767  000222                JSR     PC,REPCOD       ;REPORT THE BMP OR MODEM STATUS CHANGE CODE.
5005
5006 023540  005767  156460                TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
5007 023544  001423                       BEQ     16$             ;NO, THEN BRANCH.
5008 023546  032767  000100  156406        BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED.
5009 023554  001456                       BEQ     60$             ;YES, THEN EXIT WITH TEST FAIL MESSAGE.
5010
5011 023556  000416                       BR      16$             ;BRANCH TO GET THE NEXT CHARACTER.
5012                                ;+
5013                                ; THE DATA IS A VALID CHARACTER:
5014                                ;    COMPARE THE READ DATA WITH THE EXPECTED DATA.
5015                                ;    UPDATE EXPECTED DATA POINTER.
5016                                ;    ERRORS REPORTED WITH ERROR NUMBERS >>>>> ERRNBR+3 AND ERRNBR+4 <<<<<.
5017                                ;-
5018 023560  010467  161532        14$:    MOV     R4,ERRNBR       ;CALCULATE THE STARTING ERROR NUMBER FOR THE
5019 023564  062767  000003  161524        ADD     #3,ERRNBR       ;  NEXT ROUTINE CALL (INITIAL ERRNBR+3).
5020 023572  004767  176010                JSR     PC,NEWCHR       ;HANDLE THE NEW DATA CHARACTER.
5021 023576  005767  156422                TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
5022 023602  001404                       BEQ     16$             ;NO, THEN BRANCH.
5023 023604  032767  000100  156350        BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED.
5024 023612  001437                       BEQ     60$             ;YES, THEN EXIT WITH TEST FAIL MESSAGE.
5025                                ;+
5026                                ; DONE PROCESSING THIS CHARACTER.
5027                                ; READ ANOTHER CHAR FROM THE DUT FIFO.
5028                                ; IF DATA.VALID IS SET, LOOP TO CHECK THE RECEIVED CHARACTER.
5029                                ; IF DATA.VALID IS CLEAR LOOP TO WAIT FOR IT SET OR TIME-OUT.
5030                                ;-
5031 023614  004767  174726        16$:    JSR     PC,GETCHR       ;READ A CHARACTER FROM THE RX BUFFER.
5032 023620  103664                       BCS     8$              ;IF DATA.VALID SET, GO TO CHECK THE RX CHAR.
5033 023622  000642                       BR      4$              ;LOOP TO WAIT CHAR OR TIME-OUT IF BUFFER EMPTY.
5034                                ;+
5035                                ; USE DUMMY CHARACTERS TO FORCE ANALYSIS OF CHARACTERS IN RESYNC QUEUES.
5036                                ;-
5037 023624  004767  000736        18$:    JSR     PC,RXIEO        ;TURN OFF DUT RX INTERRUPTS.
5038 023630  004767  001462                JSR     PC,TXDONE       ;CHECK IF TX DONE, TURN OFF DUT TX INTERRUPTS.
5039 023634  005002                       CLR     R2              ;CLEAR THE DUMMY CHARACTER.
5040 023636  005001                       CLR     R1              ;CLEAR THE LOOP COUNTER.
5041 023640  004767  175742        20$:    JSR     PC,NEWCHR       ;FORCE ONE RESYNC QUE CHAR TO BE ANALYZED.
5042
5043 023644  005767  156354                TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
5044 023650  001404                       BEQ     22$             ;NO, THEN BRANCH.
5045 023652  032767  000100  156302        BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED.
5046 023660  001414                       BEQ     60$             ;YES, THEN EXIT WITH TEST FAIL MESSAGE.
5047
5048 023662  062702  000400        22$:    ADD     #400,R2         ;INCREMENT THE LINE NUMBER IN THE DUMMY CHAR.
5049 023666  005201                       INC     R1              ;INCREMENT THE LOOP COUNTER.
5050 023670  120127  000020                CMPB    R1,#NUMLNS      ;TEST FOR LOOP COUNTER EQUAL TO # OF DUT LINES.
5051 023674  002761                       BLT     20$             ;LOOP IF LOOP COUNT IS NOT ALL LINES DONE.
5052 023676  005701                       TST     R1              ;CHECK FOR SECOND TIME AROUND OUTER LOOP.
5053 023700  100404                       BMI     60$             ;EXIT IF OUTER LOOP DONE TWICE.
5054 023702  005002                       CLR     R2              ;CLEAR THE DUMMY CHAR FOR 2ND TIME AROUND LOOP.
5055 023704  012701  100000                MOV     #100000,R1      ;CLEAR LOOP COUNT, SET OUTER LOOP FLAG.
5056 023710  000753                       BR      20$             ;LOOP THE SECOND TIME AROUND OUTER LOOP.
5057
5058 023712  010467  161400        60$:    MOV     R4,ERRNBR       ;RESTORE THE ERROR NUMBER TO ITS INITIAL VALUE.
```

```
5059 023716                          PASS                    ;RESTORE GPRS.
     023716  004736                                 JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
5060 023720  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - RDMAST -
                        ;** ************************************************************************
                        ;*              - REPORT DMA_START BIT ERRORS ROUTINE -
                        ;*      THIS SUBROUTINE CHECKS FOR LINES WHICH HAVE DMA_START BIT ERRORS
                        ;*      DURING THE JUST COMPLETED DMA TRANSMISSION.  IF ANY ARE FOUND,
                        ;*      THEY ARE REPORTED.
                        ;*
                        ;* INPUTS:      ERRMSG - ADDRESS OF PRIMARY ERROR MESSAGE FOR THIS ROUTINE.
                        ;*              ERRNBR - ERROR NUMBER OF ERROR REPORTED IN THIS ROUTINE.
                        ;*              TXINTF - CONTAINS BIT MAP OF LINES WITH DMA_START BIT ERRORS.
                        ;*
                        ;* OUTPUTS:     ERRBLK - ADDRESS OF THE ERROR REPORTING ROUTINE (DESTROYED).
                        ;*              MESSAGES MAY BE PRINTED AT THE OPERATOR CONSOLE.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,RDMAST
                        ;*
                        ;* COMMENTS:    IF NO LINES HAVE DMA_START BIT ERRORS, NO MESSAGES ARE PRINTED.
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ER9102.
                        ;-- ************************************************************************

023722                  RDMAST:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
023722  004567  161376                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
023726  016702  156332          MOV     TXINTF,R2               ;GET COPY OF THE DMA_START ERRORS BIT MAP.
023732  001411              BEQ     60$                     ;EXIT IF NO DMA_START ERROR BITS ARE SET.
                        ;*
                        ; WE HAVE SOME DMA_START BIT ERRORS TO REPORT.
                        ;-
023734  012767  015576  161360          MOV     #ER9102,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
023742  012701  012442          MOV     #EM9102,R1              ;INDICATE THAT WE HAVE DMA_START BIT ERROR.
                            ;*
                            ; REPORT "DMA_START BIT SET AFTER RESET OR TX.ACTION ... ON LINES(S):"
                            ;-
023746                          ERROR                           ;               >>>>> ERROR <<<<<.
023746  104460                                                                  TRAP    C$ERROR
023750  012767  000001  156246          MOV     #1,FERROR       ;INDICATE AN ERROR HAS BEEN DETECTED.

023756                  60$:    PASS                            ;RESTORE GPRS.
023756  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
023760  000207                  RTS     PC
```

```
                                .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.
                                ;-- ************************************************************************
023762                          REPCOD:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
023762  004567  161336                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
023766  012767  014512  161326          MOV     #ER9001,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
023774  016703  161316                  MOV     ERRNBR,R3       ;PRESERVE THE ERROR NUMBER.
024000  010204                          MOV     R2,R4           ;EXTRACT THE LINE NUMBER FIELD
024002  000304                          SWAB    R4              ;  FROM THE ERROR CODE WHICH WAS
024004  042704  177760                  BIC     #177760,R4      ;  PASSED INTO THIS ROUTINE.
                                ;*
                                ; DETERMINE THE TYPE OF CODE WHICH IS TO BE REPORTED.
                                ;-
024010  012701  011236                  MOV     #EM9003,R1      ;SELECT MODEM STATUS CODE MESSAGE.
024014  032702  000001                  BIT     #BIT0,R2        ;TEST THE MODEM STATUS INDICATION BIT.
024020  001422                          BEQ     4$              ;GOTO REPORT ERROR IF MODEM STATUS CODE.
024022  005267  161270                  INC     ERRNBR          ;SELECT THE SELFTEST CODE ERROR NUMBER.
024026  012701  011260                  MOV     #EM9004,R1      ;SELECT SELFTEST CODE MESSAGE.
024032  012700  000300                  MOV     #300,R0         ;CHECK IF SELF-TEST OR BMP CODE.
024036  040200                          BIC     R2,R0           ;TRY TO CLEAR BMP BITS.
024040  001003                          BNE     2$              ;GO CHECK FOR SELFTEST CODE IF NOT BMP.
024042  004767  000604                  JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE.
024046  000423                          BR      60$             ;EXIT THIS ROUTINE.
024050  122702  000201          2$:     CMPB    #201,R2         ;CHECK FOR SELF TEST NULL CODE.
024054  001416                          BEQ     6$              ;EXIT ROUTINE IF NULL CODE FOUND.
024056  122702  000203                  CMPB    #203,R2         ;CHECK FOR SKIP SELF TEST CODE.
024062  001413                          BEQ     6$              ;EXIT ROUTINE IF SKIP SELF TEST CODE FOUND.
024064  000400                          BR      4$              ;GO REPORT SELF TEST ERROR.
                                ;*
                                ; REPORT "UNEXPECTED XXXXX CODE FOUND IN RECEIVE CHAR FIFO."
                                ;-
024066  042702  177400          4$:     BIC     #177400,R2      ;REMOVE UPPER BYTE OF CODE TO BE REPORTED.
024072  004767  002154                  JSR     PC,TXROFF       ;TURN OFF TX AND RX DURING ERROR REPORTING.
024076                                  ERROR                   ;                >>>>> ERROR <<<<<.
024076  104460                                                                    TRAP    C$ERROR
024100  012767  000001  156116          MOV     #1,FERROR       ;INDICATE THAT AN ERROR HAS BEEN FOUND.
                                
024106  004767  002200                  JSR     PC,TXRON        ;TURN TX AND RX BACK ON.
```

```
                                        ;+
                                        ; RESTORE THE INITIAL ERROR NUMBER.
                                        ;-
5158 024112  010367  161200             6$:     MOV     R3,ERRNBR
5159
5160 024116                             60$:    PASS                    ;RESTORE GPRS.
     024116  004736                             JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
5161 024120  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:
                        ;-- ************************************************************************

024122                  REPSMR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024122  004567  161176                 JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
024126  005767  156346          TST     ERSMRF                  ;CHECK THE "PRINT LINE ERROR SUMMARY" FLAGS.
024132  001404                  BEQ     60$                     ;EXIT WITHOUT ACTION IF NO SUMMARY FLAGS SET.
                        ;*
                        ; WE HAVE SOME ERROR SUMMARIES TO REPORT.
                        ;-
024134  012767  015162  161160          MOV     #ER9004,ERRBLK  ;SELECT ERROR REPORTING ROUTINE.
                        ;*
                        ; REPORT
                        ;    "ERROR SUMMARY REPORT FOR LINES WITH EXCESSIVE NUMBERS OF ERRORS:"
                        ;-
024142                          ERROR
024142  104460                                                                 TRAP    C$ERROR

024144                  60$:    PASS                            ;RESTORE GPRS.
024144  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
024146  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.
                        ;**************************************************************************

024150                  RESETT:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
024150  004567  161150                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
024154  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.
                        ;-
024160  016704  156014          MOV     CSRA,R4         ;GET THE ADDRESS OF THE DUT'S CSR.
024164  030214                  BIT     R2,(R4)         ;CHECK STATE OF MASTER RESET BIT.
024166  001406                  BEQ     2$              ;DON'T DELAY IF MR IS ALREADY CLEAR.
024170  005003                  CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
024172  012701  011610          MOV     #5000.,R1       ;PASS TIME-OUT VALUE OF 5 SECONDS.
024176  004767  174764          JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
024202  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 5 SECS, JUST IN CASE THE SELF-TEST EXECUTES.
                        ;-
024204  010277  155770  2$:     MOV     R2,@CSRA        ;SET MASTER RESET BIT, DISABLE TX AND RX INTS.
024210  004767  000504          JSR     PC,SKPSTS       ;TRY TO SKIP THE SELFTEST.
                          ;*
                          ;   SET SELF-TEST TIME-OUT OF 5 SECONDS, AND WAIT FOR M.R TO CLEAR.
                          ;   IF TIME-OUT OCCURS, THEN REPORT THE FATAL ERROR AND PASS-OUT THE ABORT
                          ;   TEST INDICATOR.
                          ;-
024214  005003                      CLR     R3              ;SET UP DESIRED STATE OF MASTER RESET BIT.
024216  012701  011610              MOV     #5000.,R1       ;PASS TIME-OUT VALUE OF 5 SECONDS.
024222  004767  174740              JSR     PC,MSLGET       ;WAIT FOR SELF-TEST TO COMPLETE, MR CLEAR.
024226  103410                      BCS     6$              ;SKIP ERROR REPORT IF MR CLEARED IN TIME.
```

```
5261                                    ;+
5262                                    ; SET UP ERROR MESSAGE TO REPORT "FATAL ERROR FOUND DURING RESET,TEST ABORTED".
5263                                    ; INDICATE TEST IS TO BE ABORTED BY CLEARING THE CARRY BIT.
5264                                    ;-
5265 024230  012701  010103            4$:     MOV     #EM1601,R1      ;PASS ERROR MESSAGE TO REPORT.
5266 024234  012767  014162  161060            MOV     #ER1603,ERRBLK ;PASS ADDRESS OF ERROR HANDLING ROUTINE.
5267                                            ;REPORT ERROR "TIME-OUT OCCURRED WAITING FOR MASTER RESET TO CLEAR"
5268                                            ; "TEST ABORTED"
5269 024242                                    ERROR                   ;                >>>>> ERROR <<<<<
     024242  104460                                                                                 TRAP    C$ERROR
5270 024244  000241                            CLC                     ;INDICATE TEST IS TO BE ABORTED.
5271 024246  000403                            BR      60$             ;EXIT THIS SUBROUTINE, ABORT TEST INDICATOR.
5272                                    ;+
5273                                    ; CLEAR TX AND RX INTERRUPT ENABLE STATUS FLAGS IN IESTAT.
5274                                    ; EXIT WITH CONTINUE TEST INDICATOR SET (IE,CARRY SET).
5275                                    ;-
5276 024250  005067  155760            6$:      CLR    IESTAT          ;CLEAR TX AND RX INTERRUPT STATUS FLAGS.
5277 024254  000261                             SEC                    ;INDICATE SUCCESS, CONTINUE TEST.
5278
5279 024256                            60$:    PASS                    ;RESTORE GPRS, PASS THE FOLLOWING INTACT:
     024256  004736                                    JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
5280                                            ;CARRY BIT:IF CLEAR,INDICATES ABORT TEST.
5281 024260  000207                            RTS     PC
5282
```

```
                                          .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.
                                          ;-- ********************************************************************

024262                                    RRXNDN:: SAVE                         ;SAVE CONTENTS OF GPRS R0 THRU R5.
024262  004567  161036                                          JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
024266  010502                                     MOV     R5,R2                ;GET COPY OF THE LOCAL ACTIVE LINES BIT MAP.
024270  046702  156210                             BIC     RXDONF,R2            ;GET MAP OF ACTIVE LINES WITH RX DONE FLAG CLR.
024274  001413                                     BEQ     60$                  ;EXIT IF NO ACTIVE LINES HAVE RX DONE FLAG CLR.
                                          ;+
                                          ; WE HAVE SOME "RX NOT COMPLETED" ERRORS TO REPORT.
                                          ;-
024276  012767  015276  161016                     MOV     #ER9005,ERRBLK       ;SELECT THE ERROR REPORTING ROUTINE.
024304  012701  011724                             MOV     #EM9016,R1           ;INDICATE THAT WE ARE DEALING WITH RECEPTION.
024310  012704  003542                             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:"
                                                ;        ...
                                                ;-
024314                                             ERROR
024314  104460                                                                                  TRAP    C$ERROR
024316  012767  000001  155700                     MOV     #1,FERROR            ;INDICATE AN ERROR HAS BEEN FOUND.

024324                                    60$:     PASS                         ;RESTORE GPRS.
024324  004736                                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
024326  000207                                     RTS     PC
```

```
                        .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.
                        ;-- ************************************************************************

024330                  RTXNDN:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024330  004567  160770                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
024334  010502          MOV     R5,R2           ;GET COPY OF THE LOCAL ACTIVE LINES BIT MAP.
024336  046702  156140  BIC     TXDONF,R2       ;GET MAP OF ACTIVE LINES WITH TX DONE FLAG CLR.
024342  001413          BEQ     60$             ;EXIT IF NO ACTIVE LINES HAVE TX DONE FLAG CLR.
                        ;+
                        ; WE HAVE SOME "TX NOT COMPLETED" ERRORS TO REPORT.
                        ;-
024344  012767  015276  160750  MOV     #ER9005,ERRBLK  ;SELECT THE ERROR REPORTING ROUTINE.
024352  012701  011710          MOV     #EM9015,R1      ;INDICATE WE ARE DEALING WITH TRANSMISSION.
024356  012704  003502          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:"
                        ;      ...
                        ;-
024362                      ERROR                       ;               >>>>> ERROR <<<<<.
024362  104460                                                              TRAP    C$ERROR
024364  012767  000001  155632  MOV     #1,FERROR       ;INDICATE THAT AN ERROR HAS BEEN FOUND.

024372                  60$:    PASS                    ;RESTORE GPRS.
024372  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
024374  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.
                        ;-- ****************************************************************************

024376                  RXDSBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024376  004567  160722                 JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
024402  010500                 MOV     R5,R0                  ;COPY BIT MAP OF LINES TO DISABLE RECEPTION.
024404  012701  000001         MOV     #BIT0,R1               ;INITIALIZE THE SELECTED LINE BIT MASK.
024410  016702  155574         MOV     LNCTRA,R2              ;GET THE ADDRESS OF THE LNCTRL REGISTER.
024414  012703  000020         MOV     #NUMLNS,R3             ;GET MAXIMUM LINE NUMBER PLUS ONE.
024420  016704  155610         MOV     IESTAT,R4              ;GET THE STATES OF THE INT ENABLE BITS.
024424  005005                 CLR     R5                     ;LOG POSSIBLE RX DISABLED ON ALL LINES.
                        ;+
                        ; SELECT EVERY LINE IN TURN, AND LOG THE STATE OF EACH RX_ENABLE BIT.
                        ;-
024426  010477  155546  2$:      MOV     R4,@CSRA               ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
024432  032712  000004         BIT     #BIT2,(R2)             ;CHECK STATE OF RX_ENABLE BIT ON SELECTED LINE.
024436  001401                 BEQ     4$                     ;SKIP NEXT INSTRUCTION IF RX_ENABLE CLEAR.
024440  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.
                        ;-
024442  030100          4$:      BIT     R1,R0                  ;CHECK STATE OF DISABLE LINE BIT MAP.
024444  001402                 BEQ     6$                     ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
024446  042712  000004         BIC     #BIT2,(R2)             ;CLEAR RX_ENABLE BIT ON SELECTED LINE.
024452  005204          6$:      INC     R4                     ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
024454  006301                 ASL     R1                     ;SHIFT BIT MAP FOR NEXT LINE.
024456  005303                 DEC     R3                     ;DECREMENT LINE NUMBER.
024460  001362                 BNE     2$                     ;LOOP TO CHECK NEXT LINE.
024462                  60$:    PASS    R5                     ;RESTORE GPRS,EXCEPT
024462  010566  000014                         MOV     R5,R5SLOT(SP)          ;PUT R5 IN STACK SLOT.
024466  004736                         JSR     PC,@(SP)+              ;RETURN TO PREGO5 SUBRT.
                                                               ;R5 - PREVIOUS STATES OF ALL RX_ENABLE BITS.
024470  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.
                                ;-- ************************************************************************

024472                          RXENBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024472  004567  160626                          JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
024476  010500                          MOV     R5,R0                   ;COPY BIT MAP OF LINES TO ENABLE.
024500  012701  000001                  MOV     #BIT0,R1                ;INITIALIZE THE SELECTED LINE BIT MASK.
024504  016702  155500                  MOV     LNCTRA,R2               ;GET THE ADDRESS OF THE LNCTRL REGISTER.
024510  012703  000020                  MOV     #NUMLNS,R3              ;GET MAXIMUM LINE NUMBER.
024514  016704  155514                  MOV     IESTAT,R4               ;GET THE STATES OF THE INT ENABLE BITS.
024520  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.
                                ;-
024522  010477  155452          2$:     MOV     R4,@CSRA                ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
024526  032712  000004                  BIT     #BIT2,(R2)              ;CHECK STATE OF RX.ENABLE BIT ON SELECTED LINE.
024532  001001                          BNE     4$                      ;SKIP NEXT INSTRUCTION IF RX.ENABLE SET.
024534  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.
                                ;-
024536  030100                  4$:     BIT     R1,R0                   ;CHECK STATE OF RX.ENABLE LINE BIT MAP.
024540  001402                          BEQ     6$                      ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
024542  052712  000004                  BIS     #BIT2,(R2)              ;ENABLE RECEPTIONON SELECTED LINE.
024546  005204                  6$:     INC     R4                      ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
024550  006301                          ASL     R1                      ;SHIFT BIT MAP FOR NEXT LINE.
024552  005303                          DEC     R3                      ;DECREMENT LINE NUMBER.
024554  001362                          BNE     2$                      ;LOOP TO CHECK NEXT LINE.

024556                          60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
024556  010566  000014                                  MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
024562  004736                                          JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
                                                        ;R5 - LINE BIT MAP CORRESPONDING TO THE
                                                        ; PREVIOUS LINES THAT WERE DISABLED.
024564  000207                          RTS     PC
```

```
                                         .SBTTL  GLOBAL SUBROUTINE                   RXIEO -
                                         ;** ********************************************************************
                                         ;*                      - RECEIVER INTERRUPT DISABLE -
                                         ;*      THIS ROUTINE IS USED TO DISABLE RECEIVER INTERRUPTS IN THE DHU11.
                                         ;*
                                         ;* 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,RXIEO
                                         ;*
                                         ;* COMMENTS:       THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                         ;*                 THE DUT CSR ARE DESTROYED.
                                         ;*
                                         ;* SUBORDINATE ROUTINES CALLED: NONE.
                                         ;-- ********************************************************************
024566  010046                           RXIEO:: MOV     R0,-(SP)         ;SAVE CONTENTS OF R0 ON THE STACK.
024570                                           GETPRI -(SP)            ;SAVE PROCESSOR PRIORITY ON STACK.
024570  104440                                                                                  TRAP    C$GPRI
024572  010046                                                                                  MOV     R0,-(SP)
024574                                           SETPRI #PRIO7           ;IGNORE ANY INTERRUPT THAT MAY BE GENERATED.
024574  012700  000340                                                                          MOV     #PRIO7,R0
024600  104441                                                                                  TRAP    C$SPRI
024602  042767  137777  155424                   BIC     #137777,IESTAT   ;CLEAR RX.INT.ENBL BIT IN IESTAT.
024610  016777  155420  155362                   MOV     IESTAT,@CSRA     ;DISABLE RX INTERRUPTS.
024616                                           SETPRI (SP)+            ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
024616  012600                                                                                  MOV     (SP)+,R0
024620  104441                                                                                  TRAP    C$SPRI
024622  012600                                   MOV     (SP)+,R0         ;RESTORE R0.
024624  000207                                   RTS     PC
```

```
5509                                    .SBTTL  GLOBAL SUBROUTINE                - RXIE1 -
5510                                    ;** ************************************************************************
5511                                    ;*                          - RECEIVER INTERRUPT ENABLE -
5512                                    ;*      THIS ROUTINE IS USED TO ENABLE RECEIVER INTERRUPTS IN THE DHU11.
5513                                    ;*
5514                                    ;* INPUTS:      NONE.
5515                                    ;*
5516                                    ;* OUTPUTS:     THE RX.INT.ENBL BIT IS SET IN THE DUT CSR.
5517                                    ;*              IESTST -CONTAINS THE UPDATED STATUS OF THE TX AND RX INTERRUPT
5518                                    ;*              ENABLE BITS.
5519                                    ;*
5520                                    ;* CALLING SEQUENCE:    JSR     PC,RXIE1
5521                                    ;*
5522                                    ;* COMMENTS:    THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
5523                                    ;*              THE DUT CSR ARE DESTROYED.
5524                                    ;*
5525                                    ;* SUBORDINATE ROUTINES CALLED: NONE.
5526                                    ;-- ************************************************************************
5527
5528 024626  052767  000100  155400  RXIE1::  BIS     #BIT06,IESTAT   ;SET RX.INT.ENBL BIT IN IESTAT.
5529 024634  042767  137677  155372           BIC     #137677,IESTAT  ;CLEAR ALL OTHER BITS, EXCEPT TX AND RX I.E.
5530 024642  016777  155366  155330           MOV     IES'AT,@CSRA    ;ENABLE RX INTERRUPTS.
5531 024650  000207                           RTS     PC
```

```
                                .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.
                                ;-- ************************************************************************

024652                          SAVBMP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024652  004567  160446                         JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
024656  016704  155626                 MOV     BMPCQP,R4               ;GET THE POINTER TO THE NEXT LOCATION IN QUEUE.
024662  116724  155372                 MOVB    TSTNUM,(R4)+            ;SAVE THE CURRENT TEST NUMBER ON THE QUEUE.
024666  005204                         INC     R4                      ;INCREMENT THE POINTER TO GIVE AN EVEN ADDRESS.
024670  042702  177400                 BIC     #177400,R2              ;CLEAR THE UNWANTED BITS FROM THE BMP CODE.
024674  010224                         MOV     R2,(R4)+                ;SAVE THE BMP CODE ON THE QUEUE.
024676  020427  002712                 CMP     R4,#BMPCQE              ;CHECK IF OVERFLOW WILL OCCUR THE NEXT TIME.
024702  103402                        BLO     2$                      ;GO SAVE THE POINTER IF WE WILL NOT OVERFLOW.
024704  162704  000004                 SUB     #4,R4                   ;RESET THE POINTER TO THE LAST LOCATION IN QUE.
024710  010467  155574          2$:    MOV     R4,BMPCQP               ;SAVE THE POINTER.
                                
024714                          60$:    PASS                            ;RESTORE GPRS.
024714  004736                                  JSR     PC,@(SP)+       ;RETURN TO PREG05 SUBRT.
024716  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.
                        ;-- ************************************************************************

024720                  SKPSTS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
024720  004567  160400                  JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
024724  012704  000012          MOV     #10.,R4                 ;PASS DELAY VALUE OF 10 MILLI-SECONDS.
024730  004767  172614          JSR     PC,DELAY                ;DELAY FOR 10 MILLI-SECONDS.
                        ;*
                        ; WRITE SKIP SELF-TEST CODE (52525) TO ALL THE INDEXED DUT REGISTERS.
                        ;-
024734  012701  000060          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 DHU-11.
024740  012703  052525          MOV     #52525,R3               ;INITIALISE THE SKIP SELF-TEST CODE.
024744  005301          4$:     DEC     R1                      ;SELECT THE NEXT SET OF DEVICE REGISTERS.
024746  016704  155226          MOV     CSRA,R4                 ;GET THE ADDRESS OF THE CSR OF THE DUT.
024752  010124                  MOV     R1,(R4)+                ;SELECT A BANK OF DUT REGISTERS.
024754  010324          6$:     MOV     R3,(R4)+                ;WRITE THE CODE TO A DUT REGISTER.
024756  020467  155234          CMP     R4,TXBFCA               ;COMPARE POINTER WITH LAST REGISTER ADDRESS.
024762  103774                  BLO     6$                      ;LOOP IF NOT ALL REGS DONE IN THIS BANK.
024764  032701  000017          BIT     #17,R1                  ;TEST FOR IND.ADR.REG FIELD DECREMENTED TO 0.
024770  001365                  BNE     4$                      ;LOOP UNTIL ALL REGISTERS CONTAIN THE CODE.

024772          60$:    PASS                                    ;RESTORE GPRS.
024772  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
024774  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.
                        ;-- ************************************************************************

024776                  SPLSUP:: SAVE                   ;SAVE CONTENTS OF THE GPR'S R0 THRU R5.
024776  004567  160322                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
025002  010067  000264          MOV     R0,70$          ;SAVE LPR PARAMETER FOR LINE GRP2.
025006  010167  000262          MOV     R1,72$          ;SAVE LPR PARAMETER FOR LINE GRP1.
025012  005067  155464          CLR     TXDONF          ;CLEAR THE TX DONE FLAGS FOR ALL LINES.
025016  005067  155462          CLR     RXDONF          ;CLEAR THE RX DONE FLAGS FOR ALL LINES.
                        ;*
                        ;  SET UP THE TRANSMISSION/RECEPTION CONTROL BLOCK TO INITIALISE THE LINES
```

```
5668                                    ;  IN GROUP II.
5669                                    ;-
5670 025022  010067  156074                     MOV     R0,CBB          ;SET CONTENTS OF LPR PARAMS IN TX/RX C.BLK.
5671 025026  012700  003124                     MOV     #CBB+2,R0       ;GET BASE ADDRESS OF CONTROL BLOCK.
5672 025032  012720  000004                     MOV     #4,(R0)+        ;LNCTRL PARAMETER, ENABLE RECEIVERS.
5673 025036  010220                             MOV     R2,(R0)+        ;START ADDRESS OF DATA PATTERN.
5674 025040  012720  000020                     MOV     #16.,(R0)+      ;DATA PATTERN LENGTH SET TO 16.
5675 025044  010420                             MOV     R4,(R0)+        ;NUMBER OF DATA PATTNS TO TRANSMIT ON LINGRP2.
5676 025046  016710  155120                     MOV     ACTLNS,(R0)     ;BIT MAP OF LINES TO INITIALISE.
5677 025052  046720  155160                     BIC     LGRP1M,(R0)+    ;CLEAR THE UNWANTED LINES FROM BIT MAP.
5678 025056  116720  155112                     MOVB    LOPBCK,(R0)+    ;SET LOOPBACK MODE.
5679 025062  005200                             INC     R0              ;INCREMENT ADDRESS TO ACCESS NEXT WORD.
5680 025064  012710  000002                     MOV     #2,(R0)         ;SET OFFSET FOR EACH TRANSMISSION START TO 2.
5681                                    ;+
5682                                    ; INITIALISE THE DUT AND THE ASSOCIATED POINTERS AND COUNTERS, TO THE STATE
5683                                    ; DICTATED BY THE CONTENTS OF THE TX/RX CONTROL BLOCK.
5684                                    ;-
5685 025070  004767  000702                   JSR     PC,TXRINI       ;INITIALISE DUT.
5686                                    ;+
5687                                    ; SET UP CONTROL BLOCK FOR LINES IN GROUP I.
5688                                    ;-
5689 025074  012700  003122                     MOV     #CBB,R0         ;GET START ADDRESS OF CONTROL BLOCK.
5690 025100  010120                             MOV     R1,(R0)+        ;SET LPR PARAMETER FOR LINES TO RECEIVE DATA.
5691 025102  012720  000004                     MOV     #4,(R0)+        ;LNCTRL PARAMETER, ENABLE RECEIVERS.
5692 025106  010220                             MOV     R2,(R0)+        ;START ADDRESS OF DATA PATTERN.
5693 025110  012720  000020                     MOV     #16.,(R0)+      ;DATA PATTERN LENGTH SET TO 16.
5694 025114  010320                             MOV     R3,(R0)+        ;NUMBER OF DATA PATTNS TO TRANSMIT ON LINGRP1.
5695 025116  016710  155050                     MOV     ACTLNS,(R0)     ;BIT MAP OF LINES TO INITIALISE.
5696 025122  046720  155112                     BIC     LGRP2M,(R0)+    ;CLEAR THE UNWANTED LINES FROM BIT MAP.
5697 025126  116720  155042                     MOVB    LOPBCK,(R0)+    ;SET LOOPBACK MODE.
5698 025132  005200                             INC     R0              ;INCREMENT ADDRESS TO ACCESS NEXT WORD.
5699 025134  012710  000002                     MOV     #2,(R0)         ;SET OFFSET FOR EACH TRANSMISSION START TO 2.
5700                                    ;+
5701                                    ; INITIALISE THE DUT AND THE ASSOCIATED POINTERS AND COUNTERS, TO THE STATE
5702                                    ; DICTATED BY THE CONTENTS OF THE TX/RX CONTROL BLOCK.
5703                                    ;-
5704 025140  004767  000632                   JSR     PC,TXRINI       ;INITIALISE DUT.
5705
5706                                    ;+
5707                                    ; SET-UP THE REQUIRED LPR PARAMETERS NEEDED FOR THE CORRECT RECEPTION OF DATA
5708                                    ; ON ASSOCIATED IN-ACTIVE LINES.
5709                                    ;-
5710                                    ;+
5711                                    ; INITIALISE LPR PARAMETERS FOR LINE GROUP 1.
5712                                    ;-
5713 025144  012701  177777                     MOV     #MAPLNS,R1      ;SET UP BIT MAP CORRESPONDING TO ALL LINES.
5714 025150  016702  155016                     MOV     ACTLNS,R2       ;GET THE ACTIVE (TX) LINE BIT MAP.
5715 025154  005101                             COM     R1              ;GENERATE A BIT MAP OF NONE EXISTANT LINES.
5716 025156  005102                             COM     R2              ;GENERATE A BIT MAP OF INACTIVE LINES.
5717 025160  040102                             BIC     R1,R2           ;CLEAR ANY "NONE EXISTANT" INACTIVE LINES.
5718 025162  046702  155052                     BIC     LGRP2M,R2       ;ONLY PASS LGRP1 ASSOCIATED LINE BIT MAP.
5719 025166  010267  155742                     MOV     R2,CBMAPA       ;SET UP BIT MAP IN CONTROL BLOCK.
5720 025172  005067  155734                     CLR     CBDPNA          ;CLEAR REPEAT TX COUNT IN CONTROL BLOCK.
5721 025176  016767  000072  155716             MOV     72$,CBLPRA      ;SET-UP COMPLEMENTARY LPR PARM FOR LGRP2.
5722 025204  004767  000566                   JSR     PC,TXRINI       ;INITIALISE INACTIVE LINES IN LGRP2.
5723                                    ;+
5724                                    ; INITIALISE LPR PARAMETERS FOR LINE GROUP 2.
```

```
 5725                                           ;-
 5726 025210  016702  154756                    MOV     ACTLNS,R2       ;GET THE ACTIVE (TX) LINE BIT MAP.
 5727 025214  005102                            COM     R2              ;GENERATE A BIT MAP OF INACTIVE LINES.
 5728 025216  040102                            BIC     R1,R2           ;CLEAR ANY NONE EXISTANT INACTIVE LINES.
 5729 025220  046702  155012                    BIC     LGRP1M,R2       ;ONLY PASS LGRP2 ASSOCIATED LINE BIT MAP.
 5730 025224  010267  155704                    MOV     R2,CBMAPA       ;SET-UP BIT MAP IN CONTROL BLOCK.
 5731 025230  016767  000036  155664            MOV     70$,CBLPRA      ;SET-UP COMPLAMENTARY LPR PARAM FOR LGRP1.
 5732 025236  004767  000534                    JSR     PC,TXRINI       ;INITIALISE INACTIVE LINES IN LGRP1.
 5733                                           ;+
 5734                                           ; DISABLE RECEIVERS ON ALL LINES TO ENSURE THAT ONLY THE RECEIVERS OF THE
 5735                                           ; ASSOCIATED ACTIVE (TX) LINES ARE ENABLED.(STAGGARED LOOPBACK)
 5736                                           ;  RE-ENABLE RECEPTION ON THE CORRECT ASSOCIATED LINES.
 5737                                           ;-
 5738 025242  012705  177777                    MOV     #MAPLNS,R5      ;SET-UP BIT MAP FOR ALL LINES.
 5739 025246  004767  177124                    JSR     PC,RXDSBL       ;DISABLE RX ON ALL LINES.
 5740                                           ;+
 5741                                           ; ENABLE RECEIVERS ON ASSOCIATED (RX) LINES.
 5742                                           ;-
 5743 025252  016705  154714                    MOV     ACTLNS,R5       ;GET ACTIVE (TX) LINE BIT MAP.
 5744 025256  004767  172212                    JSR     PC,CONMAP       ;GENERATE AN ASSOCIATED (RX) LINE BIT MAP.
 5745 025262  004767  177204                    JSR     PC,RXENBL       ;ENABLE RECEIVERS ON ASSOCIATED LINES.
 5746
 5747 025266                            60$:    PASS                    ;RESTORE GRP'S.
      025266  004736                                    JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
 5748 025270  000207                            RTS     PC
 5749 025272  000000                    70$:    .WORD   0               ;LOCAL STORAGE OF LPR PARAMETER LGRP2.
 5750 025274  000000                    72$:    .WORD   0               ;LOCAL STORAGE OF LPR PARAMETER LGRP1.
```

```
                                .SBTTL  GLOBAL SUBROUTINE       - STPSW -
                                ;** ****************************************************************************
                                ;*                      - SET PROCESSOR STATUS WORD -
                                ;*              THIS ROUTINE SETS THE PSW TO THE CONTENTS OF R1.
                                ;*
                                ;*      INPUTS:         R1 - CONTAINS THE NEW PSW SETTINGS
                                ;*
                                ;*      OUTPUTS:        PSW - SET TO THE CONTENTS OF R1
                                ;*
                                ;*      CALLING SEQUENCE:       JSR     PC,STPSW
                                ;*
                                ;*      COMMENTS:       USED IN THE DMA ADDRESS TEST TO SET THE PROCESSOR
                                ;*                      PRIORITY WITHOUT MAKING A CALL TO THE DRS.
                                ;*
                                ;*      SUBROUTINES CALLED:     NONE.
                                ;****************************************************************************

 5769 025276                    STPSW:: SAVE
      025276  004567  160022                            JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
 5770 025302  010146                     MOV    R1,-(SP)        ;PUSH THE NEW PSW CONTENTS ONTO THE STACK
 5771 025304  012746  025312             MOV    #ADDR,-(SP)     ;PUSH THE NEW PC VALUE ONTO THE STACK
 5772 025310  000002                     RTI                    ;LOAD THE NEW PC AND PSW
 5773 025312                    ADDR:    PASS                   ;
      025312  004736                            JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
 5774 025314  000207                     RTS    PC              ;RETURN
 5775
 5776
 5777
 5778
```

```

                                        .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 CONTANING THE NUMBER OF CHARS TO BE TX'D.
                                        ;*
                                        ;*
                                        ;* OUTPUTS:      TRANSMISSION INTERRUPTS ARE DISABLED.
                                        ;*
                                        ;* CALLING SEQUENCE:    JSR     PC,TXDONE
                                        ;*
                                        ;* COMMENTS:
                                        ;*
                                        ;* SUBORDINATE ROUTINES CALLED: MSLOOP,MUL16U.
                                        ;-- *************************************************************

     025316                             TXDONE:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
     025316  004567  160002                              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.
                                        ;-
     025322  016703  154644                     MOV     ACTLNS,R3       ;GET THE ACTIVE LINE BIT MAP.
     025326  016702  155150                     MOV     TXDONF,R2       ;GET THE BIT MAP OF LINES THAT HAVE COMPLETED.
     025332  040203                             BIC     R2,R3           ;GENERATE A BIT MAP OF LINES THAT ARE STILL TX.
     025334  005703                             TST     R3              ;CHECK IF ALL LINES HAVE COMPLETED TX.
     025336  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.
                                          ;-
     025340  005004                             CLR     R4              ;CLEAR LINE NUMBER COUNTER.
     025342  012702  000001                     MOV     #1,R2           ;SELECT BIT MAP FOR THE FIRST LINE.
     025346  030203                     2$:     BIT     R2,R3           ;SEE IF THIS LINE HAS COMPLETED.
     025350  001003                             BNE     4$              ;BRANCH IF THIS LINE HAS NOT COMPLETED TX.
     025352  006102                             ROL     R2              ;SHIFT THE LINE BIT MAP FOR THE NEXT LINE.
     025354  005204                             INC     R4              ;INCREMENT THE LINE NUMBER COUNTER.
     025356  000773                             BR      2$              ;LOOP TO CHECK THE NEXT LINE.
     025360  006304                     4$:     ASL     R4              ;LINE NUMBER X 2 TO OBTAIN OFFSET INTO TABLE.
     025362  016401  003442                     MOV     CHCNTB(R4),R1   ;GET THE EXPECTED NUMBER OF CHARS TO BE TX'D.
     025366  016702  154654                     MOV     RXTOUT,R2       ;GET THE CURRENT TIME-OUT VALUE FOR ONE CHAR.
     025372  004767  174134                     JSR     PC,MUL16U       ;(NUMBER OF CHARS TO TX) X (TIME-OUT OF 1 CHAR)
     025376  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.
                                           ;-
     025400  016702  154566                     MOV     ACTLNS,R2       ;PASS A BIT MAP OF THE BITS TO TEST.
```

```
5836 025404  010203                    MOV     R2,R3           ;PASS THE EXPECTED STATE OF THE TXDONF.
5837 025406  012704  002502              MOV     #TXDONF,R4      ;PASS THE ADDRESS OF THE WORD TO TEST.
5838 025412  004767  173664            JSR     PC,MSLOOP       ;WAIT FOR TIME-OUT OF TX COMPLETION.
5839 025416  004767  000270      6$:   JSR     PC,TXIE0        ;DISABLE ALL TX INTERRUPTS.
5840
5841 025422                      60$:  PASS                    ;RESTORE GPRS.
     025422  004736                          JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
5842 025424  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.
                        ;-- ************************************************************************

025426                  TXDSBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
025426  004567  157672                 JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
025432  010500                 MOV     R5,R0           ;COPY BIT MAP OF LINES TO DISABLE TRANSMISSION.
025434  012701  000001         MOV     #BIT0,R1        ;INITIALIZE THE SELECTED LINE BIT MASK.
025440  016702  154550         MOV     TXAD2A,R2       ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
025444  005202                 INC     R2              ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
025446  012703  000020         MOV     #NUMLNS,R3      ;GET MAXIMUM LINE NUMBER PLUS ONE.
025452  016704  154556         MOV     IESTAT,R4       ;GET THE STATES OF THE INT ENABLE BITS.
025456  005005                 CLR     R5              ;LOG POSSIBLE TX DISABLED ON ALL LINES.
                        ;*
                        ; SELECT EVERY LINE IN TURN, AND LOG THE STATE OF EACH TX.ENABLE BIT.
                        ;-
025460  010477  154514  2$:      MOV     R4,@CSRA        ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
025464  105712                 TSTB    (R2)            ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
025466  100001                 BPL     4$              ;SKIP NEXT INSTRUCTION IF TX.ENABLE CLEAR.
025470  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.
                        ;-
025472  030100         4$:      BIT     R1,R0           ;CHECK STATE OF DISABLE LINE BIT MAP.
025474  001402                 BEQ     6$              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
025476  142712  000200         BICB    #BIT7,(R2)      ;CLEAR TX.ENABLE BIT ON SELECTED LINE.
025502  005204         6$:      INC     R4              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
025504  006301                 ASL     R1              ;SHIFT BIT MAP FOR NEXT LINE.
025506  005303                 DEC     R3              ;DECREMENT LINE NUMBER.
025510  001363                 BNE     2$              ;LOOP TO CHECK NEXT LINE.

025512                  60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
025512  010566  000014                         MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
025516  004736                         JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
                                                                ;R5 - PREVIOUS STATES OF ALL TX.ENABLE BITS.
025520  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.
                        ;-- ************************************************************************

025522                  TXENBL:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
025522  004567  157576                 JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
025526  010500          MOV     R5,R0           ;COPY BIT MAP OF LINES TO ENABLE.
025530  012701  000001  MOV     #BIT0,R1        ;INITIALIZE THE SELECTED LINE BIT MASK.
025534  016702  154454  MOV     TXAD2A,R2       ;GET THE ADDRESS OF THE TBUFFAD2 REGISTER.
025540  005202          INC     R2              ;GET THE ADDRESS OF THE MSBYTE OF TBUFFAD2 REG.
025542  012703  000020  MOV     #NUMLNS,R3      ;GET MAXIMUM LINE NUMBER.
025546  016704  154462  MOV     IESTAT,R4       ;GET THE STATES OF THE INT ENABLE BITS.
025552  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.
                        ;-
025554  010477  154420  2$:     MOV     R4,@CSRA        ;WRITE TO DUT CSR TO SELECT LINE REGISTERS.
025560  105712          TSTB    (R2)            ;CHECK STATE OF TX.ENABLE BIT ON SELECTED LINE.
025562  100401          BMI     4$              ;SKIP NEXT INSTRUCTION IF TX.ENABLE SET.
025564  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.
                        ;-
025566  030100          4$:     BIT     R1,R0           ;CHECK STATE OF TX.ENABLE LINE BIT MAP.
025570  001402          BEQ     6$              ;BRANCH IF THIS LINE TO REMAIN UNALTERED.
025572  152712  000200  BISB    #BIT7,(R2)      ;ENABLE TRANSMISSION ON SELECTED LINE.
025576  005204          6$:     INC     R4              ;PREPARE TO SELECT REGISTERS FOR NEXT LINE.
025600  006301          ASL     R1              ;SHIFT BIT MAP FOR NEXT LINE.
025602  005303          DEC     R3              ;DECREMENT LINE NUMBER.
025604  001363          BNE     2$              ;LOOP TO CHECK NEXT LINE.

025606                  60$:    PASS    R5                      ;RESTORE GPRS,EXCEPT
025606  010566  000014                          MOV     R5,R5SLOT(SP)           ;PUT R5 IN STACK SLOT.
025612  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
                                                ;R5 - LINE BIT MAP CORRESPONDING TO THE
                                                ; PREVIOUS LINES THAT WERE DISABLED.
025614  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.
                        ;-- ************************************************************************
025616                  TXFRPR:: SAVE                    ;SAVE CONTENTS OF GPRS RO THRU R5.
025616  004567  157502                   JSR     R5,PREG05        ;CALL REGISTER SAVE SUBRT.
025622  016705  154344          MOV     ACTLNS,R5        ;GET THE ACTIVE LINE BIT MAP.
025626  005104                  COM     R4               ;GET BIT MAP OF LINES THAT WILL RECEIVE DATA.
025630  040405                  BIC     R4,R5            ;CLEAR LINES THAT WILL RX FROM TX LINE BIT MAP.
                        ;+
                        ; SET UP LOOP WHICH HANDLES ONE LINE PER ITERATION.
                        ;-
025632  005001                  CLR     R1               ;CLEAR THE LINE NUMBER COUNTER.
                            ;+
                            ; IF THE LINE IS INACTIVE SKIP TO SELECT THE NEXT LINE.
                            ;-
025634  000241          2$:     CLC                      ;CLEAR BOOLEAN REGISTER.
025636  006005                  ROR     R5               ;SHIFT BIT MAP OF LINES TO TX ON INTO BOOL.REG.
025640  103017                BCC     6$               ;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.
                        ;-
025642  010104                  MOV     R1,R4            ;COPY LINE NUMBER.
025644  006304                  ASL     R4               ;CALCULATE WORD OFFSET FOR THIS LINE.
025646  016403  003202          MOV     DPLENB(R4),R3    ;GET DATA PATTERN LENGTH FOR THIS LINE.
025652  016402  003342          MOV     TXPTRB(R4),R2    ;PREPARE TO PASS DATA PATTERN ADR TO DODMA RTN.
                            ;+
                            ; WRITE DMA PARAMETERS TO THE DUT.
```

```
6010                                    ;
6011 025656  004767  172122                JSR     PC,DODMA
6012 025662  103404                        BCS     4$                  ;SKIP ERROR IF DODMA WAS SUCCESSFUL.
6013                                    ;+
6014                                    ; SET THE PROPER BIT OF THE TX INTERRUPT FLAGS TO INDICATE THE LINE ERROR.
6015                                    ;-
6016 025664  056467  002364  154372        BIS     BITTBL(R4),TXINTF       ;INDICATE THE ERROR.
6017 025672  000402                        BR      6$                  ;SKIP UPDATING POINTERS AND COUNTERS.
6018                                    ;+
6019                                    ; UPDATE THE TX CHARACTER COUNT FOR THIS LINE.
6020                                    ;-
6021 025674  060364  003502        4$:      ADD     R3,TXCNTB(R4)       ;ADD THE DATA PATTERN LENGTH TO TX CHAR COUNT.
6022                                    ;+
6023                                    ; INCREMENT LINE COUNTER,GOTO NEXT LINE IF NOT DONE.
6024                                    ;-
6025 025700  005201                6$:      INC     R1                  ;INCREMENT THE LINE COUNTER.
6026 025702  005705                        TST     R5                  ;TEST THE TX LINE BIT MAP.
6027 025704  001353                        BNE     2$                  ;LOOP TO SEND CHAR TO ANOTHER LINE IF NOT DONE.
6028
6029 025706                        60$:    PASS                        ;RESTORE GPRS.
     025706  004736                                JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
6030 025710  000207                        RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - TXIEO -
                                ;** ****************************************************************************
                                ;*                          - TRANSMITTER INTERRUPT DISABLE -
                                ;*      THIS ROUTINE IS USED TO DISABLE TRANSMITTER INTERRUPTS IN THE DHU11.
                                ;*
                                ;* 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,TXIEO
                                ;*
                                ;* COMMENTS:     THE CONTENTS OF THE INDIRECT ADDRESS REGISTER FIELD IN
                                ;*               THE DUT CSR ARE DESTROYED.
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: NONE.
                                ;-- ****************************************************************************
025712  010046                  TXIEO::  MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 ON THE STACK.
025714                                   GETPRI  -(SP)           ;SAVE CURRENT PROCESSOR PRIORITY ON THE STACK.
025714  104440                                                                   TRAP    C$GPRI
025716  010046                                                                   MOV     R0,-(SP)
025720                                   SETPRI  #PRI07          ;IGNORE ANY INTERRUPTS THAT MAY BE GENERATED.
025720  012700  000340                                                           MOV     #PRI07,R0
025724  104441                                                                   TRAP    C$SPRI
025726  042767  177677  154300          BIC     #177677,IESTAT  ;CLEAR TX.INT.ENBL BIT IN IESTAT.
025734  016777  154274  154236          MOV     IESTAT,@CSRA    ;DISABLE TX INTERRUPTS.
025742                                   SETPRI  (SP)+           ;ENABLE INTERRUPTS TO THE PROCESSOR AGAIN.
025742  012600                                                                   MOV     (SP)+,R0
025744  104441                                                                   TRAP    C$SPRI
025746  012600                          MOV     (SP)+,R0        ;RESTORE R0.
025750  000207                          RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - TXIE1 -
                                ;** ************************************************************************
                                ;*                          - TRANSMITTER INTERRUPT ENABLE -
                                ;*      THIS ROUTINE IS USED TO ENABLE TRANSMITTER INTERRUPTS IN THE DHU11.
                                ;*
                                ;* 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.
                                ;-- ************************************************************************

 025752  052767  040000  154254  TXIE1::  BIS     #BIT14,IESTAT   ;SET TX.INT.ENBL BIT IN IESTAT.
 025760  042767  137677  154246           BIC     #137677,IESTAT  ;CLEAR ALL BITS EXCEPT TX RX I.E BITS.
 025766  016777  154242  154204           MOV     IESTAT,@CSRA    ;ENABLE TX INTERRUPTS.
 025774  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,
                        ;-- ************************************************************************
```

```
6140 025776                                TXRINI:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
     025776  004567  157322                        JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
6141                                       ;+
6142                                       ; SET UP THE LPR AND LNCTRL REGISTERS AS SPECIFIED IN THE TX/RX CONTROL BLOCK.
6143                                       ;-
6144 026002  016705  155126                        MOV     CBMAPA,R5       ;GET THE BIT MAP OF SELECTED LINES.
6145 026006  016700  155112                        MOV     CBLNCA,R0       ;GET THE NEW LNCTRL CONTENTS.
6146 026012  026727  155120  000001                CMP     CBLPBA,#1       ;CHECK IF INTERNAL LOOPBACK HAS BEEN SELECTED.
6147 026020  001002                               BNE     2$              ;SKIP SETTING INT. LOPBCK IN MAINTENANCE FIELD.
6148 026022  052700  000200                        BIS     #200,R0         ;SET INTERNAL LOOPBACK IN MAINTENANCE FIELD.
6149 026026  004767  001126          2$:           JSR     PC,WTWLNC       ;SET UP THE LNCTRL REGS FOR SELECTED LINES.
6150 026032  016700  155064                        MOV     CBLPRA,R0       ;GET THE NEW LPR CONTENTS.
6151 026036  004767  001146                        JSR     PC,WTWLPR       ;SET UP THE LPR REGISTERS FOR SELECTED LINES.
6152 026042  004767  177454                        JSR     PC,TXENBL       ;ENABLE TX FOR ALL SELECTED LINES.
6153                                       ;+
6154                                       ; SET UP AND BEGIN LOOP WHICH HANDLES ONE LINE PER ITERATION.
6155                                       ;-
6156 026046  005004                               CLR     R4              ;CLEAR THE LINE OFFSET.
6157 026050  016705  155052                        MOV     CBDPAA,R5       ;INITIALIZE THE TX START ADDRESS VALUE.
6158 026054  016703  155050                        MOV     CBDPLA,R3       ;GET THE LENGTH OF THE DATA PATTERN.
6159 026060  060503                               ADD     R5,R3           ;CALCULATE END ADDRESS OF THE DATA PATTERN.
6160 026062  036467  002364  155044  4$:           BIT     BITTBL(R4),CBMAPA  ;CHECK IF THIS LINE IS SELECTED FOR INIT.
6161 026070  001452                               BEQ     12$             ;SKIP SET UP IF LINE IS NOT SELECTED.
6162                                       ;+
6163                                       ; THIS LINE IS SELECTED FOR INITIALIZATION.
6164                                       ;    SET UP PROPER ENTRY IN NUMBER OF CHARS TO TX AND RX TABLE.
6165                                       ;    INCLUDE CHAR COUNT ON THIS LINE IN MAX ALLOWABLE CHAR TOTAL FOR ALL LINES.
6166                                       ;-
6167 026072  010701  155032                        MOV     CBDPLA,R1       ;GET THE LENGTH OF THIS LINE'S DATA PATTERN.
6168 026076  016702  155030                        MOV     CBDPNA,R2       ;GET THE NUMBER OF PATTERNS TO TX AND RX.
6169 026102  004767  173424                        JSR     PC,MUL16U       ;CALCULATE THE TOTAL NUMBER OF CHARS TO TX/RX.
6170 026106  010164  003442                        MOV     R1,CHCNTB(R4)   ;SET UP THE NUMBER OF TX/RX CHARS FOR LINE.
6171 026112  060167  154360                        ADD     R1,CHRTOT       ;ADD TWICE THE NUMBER OF CHARACTERS TO TX/PX
6172 026116  103403                               BCS     6$              ;   ON THIS LINE TO THE TOTAL NUMBER OF CHARS
6173 026120  060167  154352                        ADD     R1,CHRTOT       ;   WHICH WE WILL ALLOW TO BE RECEIVED ON
6174 026124  103003                               BCC     8$              ;   ALL LINES.
6175 026126  012767  177777  154342  6$:           MOV     #-1,CHRTOT      ;   SET MAX CHAR TOTAL TO -1 IF OVERFLOW.
6176 026134                                8$:
6177                                       ;+
6178                                       ; SET UP THE DATA PATTERN END AND LENGTH FOR THIS LINE.
6179                                       ;-
6180 026134  016764  154770  003202                MOV     CBDPLA,DPLENB(R4)  ;SET UP TX DATA PATTERN LENGTH FOR THIS LINE.
6181 026142  010364  003142                        MOV     R3,DPENDB(R4)   ;SET UP TX DATA PAT END ADDRESS FOR THIS LINE.
6182                                       ;+
6183                                       ; SET UP THE TX COUNTER AND CHARACTER POINTER FOR THIS LINE.
6184                                       ;-
6185 026146  005064  003502                        CLR     TXCNTB(R4)      ;CLEAR THE TX COUNTER FOR THIS LINE.
6186 026152  010564  003342                        MOV     R5,TXPTRB(R4)   ;SET UP THE TX CHAR POINTER FOR THIS LINE.
6187                                       ;+
6188                                       ; SET UP THE TX/RX LINE ASSOCIATION OFFSET TABLE ENTRY FOR THIS LINE.
6189                                       ;-
6190 026156  010402                               MOV     R4,R2           ;SELECT LINE OFFSET FOR NON-STAGGERED LPBK.
6191 026160  026727  154752  000002                CMP     CBLPBA,#2       ;TEST FOR STAGGERED LOOPBACK.
6192 026166  001003                               BNE     10$             ;SKIP SETTING STAGGERED LPBK IF NOT.
6193 026170  006202                               ASR     R2              ;FORM BYTE OFFSET INTO TABLE FROM TX LINE #.
6194 026172  116202  005274                        MOVB    STGTRB(R2),R2   ;GET THE RX LINE CORRESPONDING WITH TX LINE.
6195 026176  010264  005234          10$:          MOV     R2,TXRXLB(R4)   ;LOAD TX TABLE ENTRY WITH RX LINE OFFSET.
```

```
6196                                        ;+
6197                                        ; SET UP THE RX COUNTERS AND CHARACTER POINTER FOR THE RX LINE WHICH
6198                                        ;   IS ASSOCIATED WITH THIS TX LINE.
6199                                        ;-
6200 026202  005062  003542                     CLR     RXCNTB(R2)      ;CLEAR THE RX COUNTER FOR THIS RX LINE.
6201 026206  005062  003242                     CLR     EXCNTB(R2)      ;CLEAR THE EXTRA CHAR COUNTER FOR THIS RX LINE.
6202 026212  010562  003402                     MOV     R5,RXPTRB(R2)   ;SET UP THE RX CHAR POINTER FOR THIS RX LINE.
6203                                        ;+
6204                                        ; UPDATE THE TX START POINTER IN PREPARATION FOR THE NEXT LINE.
6205                                        ;-
6206 026216  066705  154716            12$:     ADD     CBOFSA,R5       ;ADD THE TX OFFSET TO THE TX START POINTER.
6207 026222  020503                    14$:     CMP     R5,R3           ;COMPARE TX START WITH END OF DATA PATTERN.
6208 026224  103403                             BLO     16$             ;SKIP WRAPAROUND IF START IS BEFORE PAT END.
6209 026226  166705  154676                     SUB     CBDTLA,R5       ;SUBTRACT DATA PATTERN LENGTH FROM START.
6210 026232  000773                             BR      14$             ;LOOP UNTIL START IS WITHIN DATA PATTERN.
6211                                        ;+
6212                                        ; UPDATE THE TX LINE NUMBER OFFSET TO THE NEXT LINE.
6213                                        ;-
6214 026234  005204                    16$:     INC     R4
6215 026236  005204                             INC     R4
6216                                        ;+
6217                                        ; TEST FOR DONE HANDLING ALL POSSIBLE LINES ON THE DEVICE.
6218                                        ;-
6219 026240  020427  000040                     CMP     R4,#NUMLNS*2    ;COMPARE OFFSET WITH 2 TIMES MAX # OF LINES.
6220 026244  002706                             BLT     4$              ;LOOP IF NOT ALL LINES DONE.
6221
6222 026246                            60$:     PASS                    ;RESTORE GPRS.
     026246  004736                                      JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
6223 026250  000207                             RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXROFF
                        ;** ************************************************************************
                        ;*                        - TURN TX AND RX OFF ROUTINE -
                        ;*      THIS SUBROUTINE IS USED TO TURN OFF DUT TRANSMISSION AND RECEPTION.
                        ;*      THIS ROUTINE ACHIEVES THIS BY BOOSTING PROCESSOR PRIORITY TO 5 TO
                        ;*      AVOID RX INTERRUPTS AND BY CLEARING ALL THE DUT TX.ENABLE BITS TO
                        ;*      HALT TX (EITHER DMA OR SINGLE CHARACTER TX).  THE STATES OF THE
                        ;*      TX.ENABLE BITS AND THE PROCESSOR PRIORITY ARE SAVED FOR RESTORATION
                        ;*      WHEN TX AND RX ARE RE-ENABLED.
                        ;*
                        ;* INPUTS:      MAPLNS - BIT MAP OF ALL POSSIBLE LINES ON THE DUT.
                        ;*
                        ;* OUTPUTS:     SAVPRI - SAVED PROCESSOR PRIORITY.
                        ;*              SAVTEN - BIT MAP OF TX.ENBL BITS (BIT SETIF TX.ENBL WAS SET).
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,TXROFF
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: TXDSBL.
                        ;-- ************************************************************************

026252                  TXROFF:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
026252  004567  157046                 JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
026256                          GETPRI  SAVPRI                  ;GET THE PRESENT PROCESSOR PRIORITY.
026256  104440                                                                  TRAP    C$GPRI
026260  010067  153764                                                          MOV     R0,SAVPRI
026264                          SETPRI  #PRIO6                  ;DISABLE DUT INTERRUPTS.
026264  012700  000300                                                          MOV     #PRIO6,R0
026270  104441                                                                  TRAP    C$SPRI
026272  012705  177777           MOV     #MAPLNS,R5            ;PREPARE TO DISABLE TX ON ALL DUT LINES.
026276  004767  177124          JSR     PC,TXDSBL             ;CLEAR ALL DUT TX.ENABLE BITS.
026302  010567  153744           MOV     R5,SAVTEN             ;PRESERVE THE PREVIOUS TX.ENABLE BIT STATES.
026306                  60$:    PASS                          ;RESTORE GPRS.
026306  004736                                 JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
026310  000207                          RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINE                - TXRON -
                                ;++ ************************************************************************
                                ;*                        - TURN TX AND RX ON ROUTINE -
                                ;*      THIS SUBROUTINE IS USED TO TURN ON DUT TRANSMISSION AND RECEPTION.
                                ;*      THIS ROUTINE RESTORES THE DUT TX.ENABLE BITS AND THE PROCESSOR PRIORITY
                                ;*      TO THE STATES SAVED BY THE TXROFF ROUTINE.
                                ;*
                                ;* INPUTS:      SAVPRI - SAVED PROCESSOR PRIORITY.
                                ;*              SAVTEN - BIT MAP OF TX.ENBL BITS (BIT SETIF TX.ENBL WAS SET).
                                ;*
                                ;* OUTPUTS:     DUT TX.ENABLE BITS - SET TO SPECIFIED STATES.
                                ;*              PROCESSOR PRIORITY - SET TO SPECIFIED PRIORITY.
                                ;*
                                ;* CALLING SEQUENCE:    JSR     PC,TXRON
                                ;*
                                ;* COMMENTS:
                                ;*
                                ;* SUBORDINATE ROUTINES CALLED: TXENBL.
                                ;-- ************************************************************************

026312                          TXRON:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
026312  004567  157006                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
026316  016705  153730                  MOV     SAVTEN,R5       ;GET THE SAVED STATES OF THE TX.ENABLE BITS.
026322  004767  177174                  JSR     PC,TXENBL       ;SET THE SPECIFIED TX.ENABLE BITS.
026326                                  SETPRI  SAVPRI          ;RESTORE THE PROCESSOR PRIORITY.
026326  016700  153716                                                  MOV     SAVPRI,R0
026332  104441                                                          TRAP    C$SPRI
026334                          60$:    PASS                    ;RESTORE GPRS.
026334  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
026336  000207                  RTS     PC
```

```
                        .SBTTL  GLOBAL SUBROUTINE                - TXRREP -
                        ;++ ************************************************************************
                        ;*                      - REPORT FINAL TX/RX ERRORS ROUTINE -
                        ;*      THIS SUBROUTINE REPORTS ERRORS WHICH ARE FOUND AFTER THE COMPLETION
                        ;*      OF THE  X, 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 ERRNBR+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.
                        ;-- ************************************************************************

026340                  TXRREP:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
026340  004567  156760                      JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
026344  006003                  ROR     R3              ;ROTATE CARRY INTO GPR TO SAVE CARRY STATE.
026346  016704  156744          MOV     ERRNBR,R4       ;SAVE THE INITIAL ERROR NUMBER VALUE.
026352  016705  153614          MOV     ACTLNS,R5       ;GET THE ACTIVE LINES BIT MAP.
026356  004767  175340          JSR     PC,RDMAST       ;REPORT ANY DMA_START BIT ERRORS.

026362  032767  000100  153572  BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
026370  001003                  BNE     2$              ;YES, THEN BRANCH.
026372  005767  153626          TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
026376  001024                  BNE     60$             ;BRANCH AND EXIT IF IT HAS.

026400  005267  156712  2$:     INC     ERRNBR          ;SELECT INTIAL ERROR NUMBER + 1.
026404  004767  175720          JSR     PC,RTXNDN       ;REPORT TX NOT COMPLETE IF NECESSARY.

026410  032767  000100  153544  BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
026416  001003                  BNE     4$              ;YES, THEN BRANCH.
026420  005767  153600          TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
026424  001011                  BNE     60$             ;BRANCH AND EXIT IF IT HAS.

026426  005267  156664  4$:     INC     ERRNBR          ;SELECT INITIAL ERROR NUMBER + 2.
026432  004767  171036          JSR     PC,CONMAP       ;GENERATE AN ASSOCIATED LINE BIT MAP.
026436  004767  175620          JSR     PC,RRXNDN       ;REPORT RX NOT COMPLETE IF NECESSARY.
026442  010467  156650          MOV     R4,ERRNBR       ;RESTORE THE INITIAL ERROR NUMBER VALUE.
```

```
6339 026446  006103                        ROL     R3              ;ROTATE SAVED CARRY STATE BACK INTO CARRY.
6340 026450                          60$:   PASS                    ;RESTORE GPRS, THIS ROUTINE PRESERVES THE
     026450  004736                                    JSR     PC,@(SP)+       ;RETURN TO PREGO5 SUBRT.
6341 026452  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.
                        ;-- ************************************************************************

026454                  UNSDIV:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
026454  004567  156644                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; CHECK FOR QUOTIENT GREATER THAN 16 BITS CONDITION.
                        ;-
026460  010204                  MOV     R2,R4           ;GET MSW OF DIVIDEND FOR SUBTRACT.
026462  160104                  SUB     R1,R4           ;SUBTRACT DIVISOR FROM MSW OF DIVIDEND.
026464  103403                  BCS     2$              ;IF IT DIDN'T GO, WE HAVE QUOTIENT < 16 BITS.
026466  012701  177777          MOV     #-1,R1          ;SET QUOTIENT TO ALL ONES (177777),
026472  000442                  BR      60$             ;EXIT WITH CARRY CLEAR.
                        ;+
                        ; SET UP COUNTERS AND VARIOUS WORKING GPRS.
                        ;-
026474  005004          2$:     CLR     R4              ;CLEAR THE LSW OF THE DIVISOR.
026476  000241                  CLC                     ;CLEAR CARRY FOR THE SHIFT OF THE DIVISOR.
026500  006001                  ROR     R1              ; DIVISOR BY
026502  006004                  ROR     R4              ; 2(UNSIGNED)
026504  012700  000020          MOV     #16.,R0         ;SET UP INITIAL SHIFT COUNT TO 16.
                        ;+
                        ; THE SUBTRACT AND SHIFT LOOP.
                        ;-
026510  010246          4$:     MOV     R2,-(SP)        ;SAVE MSWORD OF DIVIDEND.
026512  010346                  MOV     R3,-(SP)        ;SAVE LSWORD OF DIVIDEND.
026514  160403                  SUB     R4,R3           ;LSWORD DIVIDEND - LSWORD OF DIVISOR.
026516  005602                  SBC     R2              ;MSWORD DIVIDEND - BORROW .
026520  103402                  BCS     6$              ;IF BORROW FROM BORROW SUBTRACT, IT DIDN'T GO.
026522  160102                  SUB     R1,R2           ;MSWORD DIVIDEND - MSWORD OF DIVISOR.
026524  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.
                            ;-
026526  012603          6$:     MOV     (SP)+,R3        ;RESTORE LSWORD OF DIVIDEND.
026530  012602                  MOV     (SP)+,R2        ;RESTORE MSWORD OF DIVIDEND.
```

```
6399 026532  000401                        BR      10$             ;GOTO SHIFT 1 INTO THE QUOTIENT.
6400                                    ;+
6401                                    ; IT WENT, SO WE RESTORE THE STACK AND SHIFT A 0 INTO QUOTIENT (WILL BE
6402                                    ; COMPLEMENTED LATER).   CARRY IS CLEAR.
6403                                    ;-
6404 026534  012626                 8$:    MOV     (SP)+,(SP)+     ;POP THE SAVED DIVIDEND OFF OF THE STACK.
6405                                    ;+
6406                                    ; SHIFT THE RESULT OF THE SUBTRACT ATTEMPT INTO THE QUOTIENT SHIFT REG.
6407                                    ;-
6408 026536  006105                 10$:   ROL     R5              ;SHIFT NEXT BIT INTO THE INVERTED QUOTIENT.
6409 026540  000241                        CLC                     ;DIVIDE THE
6410 026542  006001                        ROR     R1              ;  DEVISOR BY
6411 026544  006004                        ROR     R4              ;  2 (UNSIGNED).
6412 026546  005300                        DEC     R0              ;COUNT THIS SHIFT AND SUBTRACT.
6413 026550  001357                        BNE     4$              ;LOOP FOR ANOTHER SHIFT & SUB IF NOT DONE.
6414 026552  005105                        COM     R5              ;GET QUOTIENT FROM INVERTED QUOTIENT.
6415                                    ;+
6416                                    ; NOW WE EITHER ROUND UP OR LEAVE QUOTIENT ALONE.
6417                                    ;-
6418 026554  000241                        CLC                     ;CLEAR THE CARRY FOR THE SHIFT OF THE DIVIDEND.
6419 026556  006103                        ROL     R3              ;MULTIPLY LSWORD OF DIVIDEND BY 2, MSWORD IS 0.
6420 026560  103402                        BCS     12$             ;IF CARRY FROM SHIFT, ROUND UP.
6421 026562  160403                        SUB     R4,R3           ;SUBTRACT DIVISOR FROM DIVIDEND.
6422 026564  103403                        BCS     14$             ;IF BORROW, DON'T ROUND UP.
6423                                    ;+
6424                                    ; ROUND UP, EXTRA SUBTRACT WENT.
6425                                    ;-
6426 026566  005205                 12$:   INC     R5              ;INCREMENT THE QUOTIENT BY ONE.
6427 026570  001001                        BNE     14$             ;IF NO OVERFLOW, WE LEAVE THE ROUND UP.
6428 026572  005305                        DEC     R5              ;DON'T LET ROUNDING CAUSE OVERFLOW.
6429                                    ;+
6430                                    ; ALL DONE, PASS QUOTIENT AND EXIT.
6431                                    ;-
6432 026574  010501                 14$:   MOV     R5,R1           ;PASS QUOTIENT BACK IN R1.
6433 026576  000261                        SEC                     ;INDICATE NO OVERFLOW.
6434
6435 026600                         60$:   PASS    R1                      ;RESTORE GPRS, LEAVE THE FOLLOWING INTACT:
     026600  010166  000004                         MOV     R1,R1SLOT(SP)           ;PUT R1 IN STACK SLOT.
     026604  004736                         JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
6436                                                 ;R1 - 16 BIT, UNSIGNED QUOTIENT.
6437 026606  000207                        RTS     PC                      ;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 BITO 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.
                        ;-- ************************************************************************
026610                  UPDCHR:: SAVE                   ;SAVE CONTENTS OF GPRS R0 THRU R5.
026610  004567  156510                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
026614  016302  005234          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.
                        ;-
026620  016301  003402          MOV     RXPTRB(R3),R1   ;GET THE RX DATA POINTER FROM THE RX PTR TABLE.
026624  005201                  INC     R1              ;INCREMENT THE RX POINTER VALUE BY 1.
026626  020162  003142          CMP     R1,DPENDB(R2)   ;CMP RX PTR VALUE WITH ADR OF END OF DATA PAT.
026632  103402                  BLO     2$              ;SKIP WRAPPING RX PTR AROUND IF NOT AT END.
026634  166201  003202          SUB     DPLENB(R2),R1   ;WRAP RX PTR AROUND TO START OF DATA PATTERN.
026640  010163  003402  2$:     MOV     R1,RXPTRB(R3)   ;UPDATE THE RX POINTER WITH THE NEW VALUE.
                        ;*
                        ; UPDATE THE RX CHARACTER COUNT WITH OVERFLOW DETECTION.
                        ;-
026644  016301  003542          MOV     RXCNTB(R3),R1   ;GET THE RX CHARACTER COUNT.
026650  005201                  INC     R1              ;INCREMENT THE RX CHAR COUNT VALUE BY 1.
026652  001002                  BNE     4$              ;SKIP SETTING COUNT TO MAX IF NO OVERFLOW.
026654  012701  177777          MOV     #-1,R1          ;SET RX CHAR COUNT VALUE TO MAX VALUE.
026660  010163  003542  4$:     MOV     R1,RXCNTB(R3)   ;UPDATE THE RX CHAR COUNT WITH NEW VALUE.
                        ;*
                        ; CHECK FOR RX COMPLETION ON THIS LINE.
                        ; IF RX IS COMPLETE ON THIS LINE, SET THE CORRECT RX DONE FLAG.
                        ;-
026664  016204  003442          MOV     CHCNTB(R2),R4   ;GET THE NUMBER OF TX CHARS IN COMPLETE TX.
026670  020104                  CMP     R1,R4           ;COMPARE RX CHAR COUNT WITH NUMBER OF TX CHARS.
026672  103403                  BLO     60$             ;EXIT ROUTINE IF NOT ALL CHARS RECEIVED.
```

```
6495 026674  056367  002364  153602          BIS     BITTBL(R3),RXDONF       ;SET THE RX DONE FLAG FOR THIS LINE.
6496
6497 026702                          60$:     PASS                            ;RESTORE GPRS.
     026702  004736                                  JSR     PC,@(SP)+               ;RETURN TO PREGOS SUBRT.
6498 026704  000207                          RTS     PC
```

```
                        .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 TRANMISSION,
                        ;*              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.
                        ;-- ************************************************************************

 026706                 VANSUP:: SAVE                          ;SAVE CONTENTS OF THE GPR'S R0 THRU R5.
 026706 004567 156412                   JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
 026712 005067 153560           CLR     CHRTOT                  ;CLEAR TOTAL RECEIVED CHAR COUNTER.
 026716 005067 153342           CLR     TXINTF                  ;CLEAR FLAGS USED TO LOG DMA M.OVER ERRORS.
 026722 005067 153554           CLR     TXDONF                  ;CLEAR THE TX DONE FLAGS.
 026726 005067 153552           CLR     RXDONF                  ;CLEAR THE RX DONE FLAGS.
                        ;*
                        ;  SET UP THE TRANSMISSION/RECEPTION CONTROL BLOCK TO THE DESIRED STATE.
                        ;-
 026732 010167 154164           MOV     R1,CBB                  ;SET CONTENTS OF LPR PARAMS IN TX/RX C.BLK.
 026736 012701 003122           MOV     #CBB,R1                 ;GET BASE ADDRESS OF CONTROL BLOCK.
 026742 005201                  INC     R1                      ;INCREMENT ADDRESS FOR NEXT WORD
 026744 005201                  INC     R1                      ;INITIALISE THE FOLLOWING IN THE CNTRL.BLK:
 026746 012721 000004           MOV     #4,(R1)+                ; LNCTRL PARAMETER, ENABLE RECEIVERS.
 026752 010221                  MOV     R2,(R1)+                ; START ADDRESS OF DATA PATTERN.
 026754 010321                  MOV     R3,(R1)+                ; DATA PATTERN LENGTH.
 026756 010421                  MOV     R4,(R1)+                ; NUMBER OF DATA PATTERNS TO TRANSMIT.
 026760 016721 153206           MOV     ACTLNS,(R1)+            ; BIT MAP OF LINES TO INITIALISE.
 026764 032767 000004 153202    BIT     #BIT2,LOPBCK            ;TEST IF MODEM LOOPBACK MODE HAS BEEN SELECTED.
 026772 001404                  BEQ     2$                      ;DONT SELECT INTERNL LOPBCK IF STAGRD OR LOCAL.
 026774 012702 000001           MOV     #1,R2                   ;FORCE INTERNAL LOOPBACK MODE TO BE SELECTED.
 027000 110221                  MOVB    R2,(R1)+                ;INITIALISE LOOPBACK MODE IN CONTROL BLOCK.
```

```
6556 027002  000402                    BR      4$                  ;SKIP NEXT INSTRUCTION IF IN MODEM LOOPBACK.
6557 027004  116721  153164      2$:    MOVB    LOPBCK,(R1)+        ;SET LOOPBACK MODE.
6558 027010  005201              4$:    INC     R1                  ;INCREMENT ADDRESS FOR THE NEXT WORD.
6559 027012  012711  000002             MOV     #2,(R1)             ;SET AMOUNT OF OFFSET EACH TX STARTS AT TO 2.
6560                                    ;+
6561                                    ; INITIALISE THE DUT AND THE ASSOCIATED POINTERS AND COUNTERS, TO THE STATE
6562                                    ; DICTATED BY THE CONTENTS OF THE TX/RX CONTROL BLOCK.
6563                                    ;-
6564 027016  004767  176754             JSR     PC,TXRINI           ;INITIALISE DUT.
6565                                    ;+
6566                                    ; INITIALISE POINTERS AND COUNTERS FOR INACTIVE LINES TO ZERO.
6567                                    ;-
6568 027022  012701  177777             MOV     #MAPLNS,R1          ;GET THE LINE BIT MAP FOR ALL LINES.
6569 027026  016702  153140             MOV     ACTLNS,R2           ;GET THE ACTIVE LINE BIT MAP.
6570 027032  005101                     COM     R1                  ;
6571 027034  005102                     COM     R2                  ;
6572 027036  040102                     BIC     R1,R2               ;GENERATE AN IN-ACTIVE LINE BIT MAP.
6573 027040  010267  154070             MOV     R2,CBMAPA           ;MOVE BIT MAP TO THE CONTROL BLOCK.
6574 027044  005067  154062             CLR     CBDPNA              ;CLEAR THE REPEAT TX COUNT IN CNTRL BLCK.
6575 027050  004767  176722             JSR     PC,TXRINI           ;SET UP PARAMETERS FOR INACTIVE LINES.
6576                                    ;+
6577                                    ; DISABLE RECEIVERS ON ALL LINES TO ENSURE CORRECT INITIALISATION OF ONLY THE
6578                                    ; LINES THAT ARE SELECTED.
6579                                    ;-
6580 027054  012705  177777             MOV     #MAPLNS,R5          ;SET-UP BIT MAP FOR ALL LINES.
6581 027060  004767  175312             JSR     PC,RXDSBL           ;DISABLE RX ON ALL LINES.
6582                                    ;+
6583                                    ; ENABLE RECEIVERS ON ASSOCIATED (RX) LINES.
6584                                    ;-
6585 027064  016705  153102             MOV     ACTLNS,R5           ;GET THE ACTIVE LINE BIT MAP.
6586 027070  004767  170400             JSR     PC,CONMAP           ;GENERATE AN ASSOCIATED LINE BIT MAP.
6587 027074  004767  175372             JSR     PC,RXENBL           ;ENABLE RECEIVERS ON ASSOCIATED LINES.
6588 027100                     60$:    PASS                        ;RESTORE GPR'S.
     027100  004736                             JSR     PC,@(SP)+           ;RETURN TO PREGOS SUBRT.
6589 027102  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.
                        ;-- ************************************************************************

027104                  WAIBIS:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
027104  004567  156214                  JSR     R5,PREGO5       ;CALL REGISTER SAVE SUBRT.
027110  010204                  MOV     R2,R4                   ;SET UP THE ADDRESS PARAMETER FOR MSLGET.
027112  010102                  MOV     R1,R2
027114  042701  170000          BIC     #170000,R1              ;SEPERATE DELAY COUNT OUT OF PASSED PARAMETER.
027120  042702  007777          BIC     #7777,R2                ;SEPERATE LINE NUMBER FIELD OF PASSED PARAM.
027124  000302                  SWAB    R2                      ;PUT LINE NUMBER FIELD IN LSBYTE.
027126  006202                  ASR     R2                      ;SHIFT THE LINE NUMBER FIELD INTO THE PROPER
027130  006202                  ASR     R2                      ;  POSITION TO USE IT AS A WORD TABLE OFFSET
027132  006202                  ASR     R2                      ;  FOR THE TABLE LOOKUP OF THE LINE BIT MAP.
027134  016202  002364          MOV     BITTBL(R2),R2           ;GET BIT MAP OF LINE TO TEST FROM TABLE.
027140  010203                  MOV     R2,R3                   ;INDICATE THAT THE BIT SHOULD BE SET.
027142  004767  172020          JSR     PC,MSLGET               ;WAIT FOR THE BIT TO BE SET WITHIN TIME-OUT.
                                                                ;  CARRY IS CORRECT UPON MSLGET RETURN.
027146  010002                  MOV     R0,R2                   ;PASS LAST VALUE READ AS OUTPUT PARAMETER.
027150                  60$:    PASS    R2                      ;RESTORE GPRS, EXCEPT THE FOLLOWING:
027150  010266  000006                          MOV     R2,R2SLOT(SP)           ;PUT R2 IN STACK SLOT.
027154  004736                              JSR     PC,@(SP)+               ;RETURN TO PREGO5 SUBRT.
                                                                ;  R2 - LAST VALUE READ LOOKING FOR CONDITION.
027156  000207                  RTS     PC                      ;  CARRY - SUCCESS FLAG (SET IF BIT FOUND SET).
```

```
                        .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.
                        ;-- ************************************************************************

027160                  WTWLNC:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
027160  004567  156140                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
                        ;-
027164  016701  153020          MOV     LNCTRA,R1       ;SET UP THE REGISTER ADDRESS PARAMETER.
027170  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
027172  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
027174  012704  177777          MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                        ;+
                        ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
                        ;-
027200  004767  166706          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

027204                  60$:    PASS                    ;RESTORE GPRS.
027204  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
027206  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 PARAMTER REGISTERS ARE ALTERED.
                        ;*
                        ;* CALLING SEQUENCE:    JSR     PC,WTWLPR
                        ;*
                        ;* COMMENTS:
                        ;*
                        ;* SUBORDINATE ROUTINES CALLED: ALTFLD.
                        ;-- ************************************************************************

027210                  WTWLPR:: SAVE                           ;SAVE CONTENTS OF GPRS R0 THRU R5.
027210  004567  156110                  JSR     R5,PREG05       ;CALL REGISTER SAVE SUBRT.
                        ;+
                        ; SET UP THE PARAMETERS FOR THE CALL TO ALTFLD.
                        ;-
027214  016701  152764          MOV     LPRA,R1         ;SET UP THE REGISTER ADDRESS PARAMETER.
027220  010002                  MOV     R0,R2           ;SET UP THE DESIRED REGISTER CONTENTS.
027222  010503                  MOV     R5,R3           ;SET UP THE BIT MAP OF LINES TO ALTER.
027224  012704  177777          MOV     #-1,R4          ;SELECT ALL REGISTER BITS TO BE ALTERED.
                        ;+
                        ; CALL THE SUBROUTINE WHICH ALTERS THE REGISTER CONTENTS.
                        ;-
027230  004767  166656          JSR     PC,ALTFLD       ;ALTER THE REGISTER CONTENTS.

027234                  60$:    PASS                            ;RESTORE GPRS.
027234  004736                          JSR     PC,@(SP)+               ;RETURN TO PREG05 SUBRT.
027236  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.
                                ;-- ****************************************************************************

027240  005767  153034        CLKINT:: TST     TIMER1          ;CHECK FOR TIMER1 AT ZERO.
027244  001402                          BEQ     2$              ;BRANCH TO LEAVE IT AT ZERO IF IT IS ZERO.
027246  005367  153026                  DEC     TIMER1          ;DECREMENT TIME COUNT.
027252  005767  153024        2$:       TST     TIMER2          ;CHECK FOR TIMER2 AT ZERO.
027256  001402                          BEQ     4$              ;BRANCH TO LEAVE IT ALONE IF IT'S ALREADY ZERO.
027260  005367  153016                  DEC     TIMER2          ;DECREMENT TIME COUNT.
027264  005367  153014        4$:       DEC     TIMER3          ;DECREMENT THE BREAK COUNT.
027270  001006                          BNE     60$             ;EXIT IF NOT TIME TO CALL BREAK.
027272  016767  153010  153004        MOV     BCOUNT,TIMER3   ;SET UP TIME TILL NEXT BREAK.
027300  010046                        MOV     R0,-(SP)        ;SAVE CONTENTS OF R0 FROM BREAK MACRO.
027302                                BREAK                   ;CHECK FOR OPERATOR CONTROL/C.
027302  104422                                                                  TRAP    C$BRK
027304  012600                        MOV     (SP)+,R0        ;RESTORE CONTENTS OF R0.
027306  000002              60$:      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.
                        ;-- ****************************************************************************

027310  010246          RXCHRS::        MOV     R2,-(SP)        ;SAVE CONTENTS OF GPR R2.
027312  017702  152664  2$:     MOV     @RBUFA,R2       ;READ A CHARACTER FROM THE DUT RX FIFO.
027316  100054          BPL     60$             ;EXIT THE ROUTINE IF THE DATA.VALID BIT IS CLR.

027320  026727  153372  000100      CMP     RXBCNT,#RXBFUL  ;COMPARE BUFFER COUNT WITH BUFFER CAPACITY.
027326  103402          BLO     4$              ;SKIP ABORT IF BUFFER IS NOT FULL.
027330  004767  172562          JSR     PC,OOPS         ;ABORT, MUST BE A PROGRAM BUG.
027334  010277  153354  4$:     MOV     R2,@RXBIPT      ;PUT THE CHAR IN THE BUFFER.
027340  062767  000002  153346      ADD     #2,RXBIPT       ;UPDATE POINTER TO THE NEXT BUFFER SLOT.
027346  026727  153342  003120      CMP     RXBIPT,#RXBEND  ;CHECK IF POINTER SHOULD WRAP AROUND.
027354  103403          BLO     6$              ;SKIP WRAPAROUND IF POINTER IS NOT AT END.
027356  012767  002720  153330      MOV     #RXBSTA,RXBIPT  ;WRAP INPUT POINTER AROUND.

027364  005267  153326  6$:     INC     RXBCNT          ;COUNT THIS CHARACTER AS BEING IN THE BUFFER.
027370  026727  153322  000030      CMP     RXBCNT,#RXBDTX  ;CHECK FOR BUFFER AT DISABLE TX LEVEL.
027376  002745          BLT     2$              ;SKIP DISABLING TX IF BUFFER LEVEL NOT CORRECT.
027400  005767  153102          TST     TXDBLF          ;CHECK STATE OF TX DISABLE FLAG.
027404  100413          BMI     8$              ;BRANCH IF TRANSMISSION ALREADY DISABLED.
027406  010546          MOV     R5,-(SP)        ;SAVE THE VALUE OF GPR R5.
027410  012705  177777      MOV     #MAPLNS,R5      ;SPECIFY THAT ALL LINES SHOULD BE AFFECTED.
027414  004767  176006      JSR     PC,TXDSBL       ;CLEAR THE TX ENABLES FOR ALL LINES.
027420  010567  152636      MOV     R5,TXENBM       ;SAVE PREVIOUS TX ENABLE STATES IN STORAGE.
```

```
 6809 027424  012605                          MOV    (SP)+,R5         ;RESTORE GPR R5.
 6810 027426  012767  100000  153052          MOV    #BIT15,TXDBLF    ;PREVENT TX FROM BEING DISABLED AGAIN.
 6811
 6812 027434  026727  153256  000100  8$:     CMP    RXBCNT,#RXBFUL   ;CHECK FOR BUFFER FULL CONDITION.
 6813 027442  103723                          BLO    2$               ;LOOP TO READ ANOTHER CHAR IF BUFFER NOT FULL.
 6814
 6815 027444  004767  175116                  JSR    PC,RXIE0         ;BUFFER IS FULL, DISABLE RX INTERRUPTS.
 6816
 6817 027450  012602                  60$:    MOV    (SP)+,R2         ;RESTORE R2 TO ITS SAVED VALUE.
 6818 027452  000002                          RTI
```

```
                                .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.
                                ;**************************************************************************

027454  021627  017412          TP4BRT:: CMP    (SP),@TRPAD2    ;COMPARE EXPECTED ADDR WITH TRAP RET PC.
027460  001402                           BEQ    2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
027462  000177  152570                   JMP    @TP4VEC         ;IF NOT, JUMP TO NORMAL 004 TRAP SERVICE RTN.
027466  052767  100000  152560  2$:      BIS    #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
027474  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 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.
                                ;*******************************************************************************

027476  021627  017362        TP4RTN:: CMP     (SP),@ADRPTR    ;COMPARE EXPECTED ADR AGAINST TRAP RET PC.
027502  001402                        BEQ     2$              ;IF THEY MATCH, CONTINUE THIS ROUTINE.
027504  000177  152546                JMP     @TP4VEC         ;IF NOT,JUMP TO NORMAL 004 TRAP SERVICE RTN.
027510  052767  100000  152536  2$:   BIS     #BIT15,TP4FLG   ;SET THE 004 TRAP OCCURED FLAG.
027516  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.
                                ;-- ************************************************************************

027520                          TXDMA:: SAVE                    ;SAVE CONTENTS OF GPRS R0 THRU R5.
027520  004567  155600                        JSR     R5,PREGOS       ;CALL REGISTER SAVE SUBRT.
027524  017701  152450                 MOV     @CSRA,R1        ;READ THE CONTENTS OF THE DUT CSR.
027530  010100                         MOV     R1,R0           ;SAVE INITIAL CONTENTS OF IND.ADR.REG FIELD.
027532  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.
                                ;-
027534  017701  152440          2$:      MOV     @CSRA,R1        ;READ THE CONTENTS OF THE DUT CSR.
027540  100033                 4$:     BPL     60$             ;EXIT ROUTINE IF TX.ACTION IS CLEAR.
027542  000301                          SWAB    R1              ;CALCULATE THE LINE NUMBER OF THE LINE WHICH IS
027544  042701  177760                  BIC     #177760,R1      ;  ASSOCIATED WITH THE TX.ACTION.
027550  010104                          MOV     R1,R4           ;CALCULATE AN OFFSET FOR USE IN ACCESSING
027552  006304                          ASL     R4              ;  LINE COUNTER AND POINTER IN TABLES.
027554  016405  002364                  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.
                                ;-
027560  026464  003502  003442          CMP     TXCNTB(R4),CHCNTB(R4) ;COMPARE # CHARS SENT AND TX COUNT.
027566  103403                        BLO     6$              ;GO TO SEND A CHAR IF NOT ALL CHARS SENT.
```

```
6937 027570  050567  152706                  BIS     R5,TXDONF       ;SET THIS LINE'S TX DONE FLAG.
6938 027574  000757                          BR      2$              ;LOOP TO CHECK TX.ACTION AGAIN.
6939                                ;+
6940                                ; START THE DMA OF THE NEXT BUFFER (DATA PATTERN) ON THIS LINE.
6941                                ;    GET THE DATA PATTERN LENGTH FOR THIS LINE.
6942                                ;    GET THE START ADDRESS OF THE DATA PATTERN.
6943                                ;-
6944 027576  016403  003202        6$:     MOV     DPLENB(R4),R3   ;PASS DATA PATTERN LENGTH FOR LINE TO DODMA.
6945 027602  016402  003342                MOV     TXPTRB(R4),R2   ;PASS THE TX START ADR TO DODMA.
6946                                    ;+
6947                                    ; WRITE DMA PARAMETERS TO THE DUT.
6948                                    ;
6949 027606  004767  170172                JSR     PC,DODMA
6950 027612  103403                        BCS     8$              ;SKIP ERROR IF DODMA WAS SUCCESSFUL.
6951                                    ;+
6952                                    ; SET THE PROPER BIT OF THE TX INTERRUPT FLAGS TO INDICATE THE LINE ERROR.
6953                                    ;-
6954 027614  050567  152444                BIS     R5,TXINTF       ;INDICATE THE ERROR.
6955 027620  000402                        BR      10$             ;SKIP UPDATING POINTERS AND COUNTERS.
6956                                    ;+
6957                                    ; UPDATE THE TX CHARACTER FOR THIS LINE.
6958                                    ; UPDATE THE TX BUFFER POINTER FOR THIS LINE.
6959                                    ;-
6960 027622  060364  003502        8$:     ADD     R3,TXCNTB(R4)   ;ADD THE DATA PAT LENGTH TO THE TX COUNT.
6961                                ;+
6962                                ; LOOP TO CHECK THE TX.ACTION BIT FOR ANOTHER LINE.
6963                                ;-
6964 027626  000742                10$:    BR      2$              ;LOOP BACK TO CHECK TX.ACTION BIT AGAIN.
6965
6966 027630  016701  152400        60$:    MOV     IESTAT,R1       ;GET THE PRESENT STATES OF TX.IE & RX.IE BITS.
6967 027634  042700  177760                BIC     #177760,R0      ;GET SAVED IND.ADR.REG FIELD BITS.
6968 027640  050001                        BIS     R0,R1           ;COMBINE IND.ADR.REG FIELD BITS WITH IE BITS.
6969 027642  010177  152332                MOV     R1,@CSRA        ;RESTORE THE DUT CSR IND.ADR.REG FIELD.
6970 027646                                PASS                    ;RESTORE GPRS.
     027646  004736                                JSR     PC,@(SP)+       ;RETURN TO PREGOS SUBRT.
6971 027650  000002                        RTI
```

```
   6973
   6974                         ;**********************************************************************
**
   6975                         ;
   6976                         ;               FVTA.RPT
   6977                         ;
   6978                         ;**********************************************************************
   6979
   6980
   6981
   6982                         .SBTTL  REPORT CODING SECTION
   6983
   6984                         ;++
   6985                         ; THE REPORT CODING SECTION CONTAINS THE
   6986                         ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
   6987                         ;--
   6988
   6989 027652                           BGNRPT
        027652                                                          L$RPT::
   6990
   6991 027652                           EXIT    RPT
        027652  000167                                                          .WORD   J$JMP
        027654  000000                                                          .WORD   L10016-2-.
   6992
   6993                                  .EVEN
   6994
   6995 027656                           ENDRPT
        027656                                                          L10016:
        027656  104425                                                          TRAP    C$RPT
```

```
                                .SBTTL  PROTECTION TABLE

                                ;**********************************************************************
*
                                ;
                                ;               FVTSKL4.P11
                                ;
                                ;**********************************************************************



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

027660                                  BGNPROT
027660                                                                  L$PROT::

027660  177777                          1               ;OFFSET INTO P-TABLE FOR CSR ADDRESS
027662  177777                          -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
027664  177777                          -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER

027666                                  ENDPROT
```

```
                        ;*****************************************************************************
                        ;
                        ;               FVTC.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.
                        ;*
                        ;*****************************************************************************
                        ;--
027666                          BGNINIT
027666                                                                  L$INIT::
                        ;SEE IF PROGRAM JUST STARTED, BR IF YES
                                READEF  #EF.START
027666  012700  000040                                                  MOV     #EF.START,R0
027672  104447                                                          TRAP    C$REFG
027674                          BCOMPLETE       NEWSTA
027674  103416                                                          BCS     NEWSTA
                        ;SEE IF PROGRAM JUST RESTARTED, BR IF YES
                                READEF  #EF.RESTART
027676  012700  000037                                                  MOV     #EF.RESTART,R0
027702  104447                                                          TRAP    C$REFG
027704                          BCOMPLETE       NEWRES
027704  103556                                                          BCS     NEWRES
                        ;SEE IF THIS IS A NEW PASS, BR IF YES
                                READEF  #EF.NEW
027706  012700  000035                                                  MOV     #EF.NEW,R0
027712  104447                                                          TRAP    C$REFG
027714                          BCOMPLETE       NEWPAS
027714  103555                                                          BCS     NEWPAS
                        ;SEE IF PROGRAM WAS JUST CONTINUED
                                READEF  #EF.CONTINUE
027716  012700  000036                                                  MOV     #EF.CONTINUE,R0
027722  104447                                                          TRAP    C$REFG
027724                          BNCOMPLETE      GETPRM
027724  103161                                                          BCC     GETPRM
027726  000167  000544         JMP     ENDIT
027732                  NEWSTA:
027732                          BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
027732  104433                                                          TRAP    C$RESET
                        ;+
                        ; SET UP FOR LINE TIME CLOCK INTERRUPTS.
                        ;-
027734                          CLOCK   L,R1            ;GET THE CLOCK PARAMETERS.
027734  012700  000114                                                  MOV     #'L,R0
027740  104462                                                          TRAP    C$CLCK
```

```
     027742  010001                                                                 MOV     R0,R1
7075 027744  012167  152320                MOV     (R1)+,CLKCSR            ;STORE CLOCK CSR ADDRESS.
7076 027750  012167  152316                MOV     (R1)+,CLKBRL            ;STORE CLOCK BUS REQ INT LEVEL.
7077 027754  012167  152314                MOV     (R1)+,CLKVEC            ;STORE CLOCK INTERRUPT VECTOR.
7078 027760  012167  152312                MOV     (R1)+,CLKHRZ            ;STORE CLOCK FREQUENCY.
7079 027764  026727  152306  000062        CMP     CLKHRZ,#50.             ;TEST FOR 50HZ LINE FREQUENCY.
7080 027772  001004                        BNE     2$                      ;BRANCH IF CLOCK IS NOT 50HZ.
7081 027774  012767  000024  152306        MOV     #20.,MSTICK             ;INDICATE 20MS PER CLOCK TICK.
7082 030002  000403                        BR      4$
7083 030004  012767  000021  152276  2$:   MOV     #17.,MSTICK             ;INDICATE 17 MS PER CLOCK TICK.
7084 030012                        4$:     SETVEC  CLKVEC,#CLKINT,PRI06    ;INITIALIZE CLOCK INTERRUPT VECTOR.
     030012  016746  150262                                                        MOV     PRI06,-(SP)
     030016  012746  027240                                                        MOV     #CLKINT,-(SP)
     030022  016746  152246                                                        MOV     CLKVEC,-(SP)
     030026  012746  000003                                                        MOV     #3,-(SP)
     030032  104437                                                               TRAP    C$SVEC
     030034  062706  000010                                                        ADD     #10,SP
7085 030040  016700  152232                MOV     CLKHRZ,R0               ;INITIALIZE THE BREAK COUNT
7086 030044  006300                        ASL     R0                      ; TO CAUSE A BREAK
7087 030046  010067  152234                MOV     R0,BCOUNT               ; EVERY 2 SECONDS.
7088 030052                                SETPRI  #PRI05                  ;ALLOW CLOCK INTERRUPTS DISABLE OTHERS.
     030052  012700  000240                                                        MOV     #PRI05,R0
     030056  104441                                                               TRAP    C$SPRI
7089                                ;+
7090                                ; ENABLE THE LINE TIME CLOCK (LTC) CHECKING TO MAKE SURE THAT THE CSR
7091                                ; IS ACCESSABLE.
7092                                ;    FIRST SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
7093                                ;-
7094 030060  016767  147720  152170        MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
7095 030066  012767  027476  147710        MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
7096                                ;+
7097                                ;    ENABLE LTC CHECKING FOR 004 TRAP IN CASE CSR IS NOT THERE.
7098                                ;-
7099 030074  005067  152154                CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
7100 030100  012767  000100  152160        MOV     #BIT6,WORD1     ;SET UP TO SET BIT6 OF THE LTC CSR.
7101 030106  012700  002266                MOV     #WORD1,R0       ;SET UP WORD1 AS THE CKTRAP MOVE SOURCE.
7102 030112  016701  152152                MOV     CLKCSR,R1       ;SET UP LTC CSR AS DESTINATION FOR CKTRAP MOVE.
7103 030116  004767  167226                JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
7104 030122  016767  152130  147654        MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
7105 030130  103403                        BCS     6$              ;IF NO TRAP, LTC IS THERE SO CONTINUE.
7106 030132  005067  152140                CLR     CLKHRZ          ;CLEAR LTC FREQUENCY WORD TO INDICATE NO LTC.
7107 030136  000402                        BR      8$              ;BYPASS THE FOLLOWING CALIBRATION PROCEDURES.
7108                                ;+
7109                                ; CALIBRATE THE DELAY ROUTINE MILLI-SECOND DELAY COUNT VALUE.
7110                                ;-
7111 030140  004767  166020        6$:     JSR     PC,CALMSL
7112                                ;+
7113                                ; CHECK FOR MEMMORY MANAGEMENT PRESENT ON THIS MACHINE.
7114                                ; IF MEM MGT IS PRESENT, DISABLE IT.
7115                                ;-
7116 030144  016767  147634  152104  8$:       MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
7117 030152  012767  027476  147624        MOV     #TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
7118 030160  005067  152070                CLR     TP4FLG          ;CLEAR THE 004 TRAP FLAG.
7119 030164  005067  152076                CLR     WORD1           ;PREPARE TO CLEAR THE MEM MGT SR0 REGISTER.
7120 030170  012700  002266                MOV     #WORD1,R0       ;SELECT CLEARED WORD AS CKTRAP RTN SOURCE.
7121 030174  016701  152114                MOV     MMSR0,R1        ;SELECT MEM MGT SR0 REGISTER AS DESTINATION.
7122 030200  005067  152114                CLR     MMPRES          ;INDICATE NO MEM MGT PRESENT IN CASE IT ISN'T.
```

```
7123 030204  005067  152112                    CLR     MMENAB          ;INDICATE MEM MGT IS NOT ENABLED.
7124 030210  004767  167134                    JSR     PC,CKTRAP       ;CLEAR THE MEM MCT SR0 REG AND CHECK FOR TRAP.
7125 030214  016767  152036  147562            MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
7126 030222  103003                            BCC     10$             ;SKIP INDICATING MEM MGT PRESENT IF IT ISN'T.
7127 030224  012767  000001  152066            MOV     #1,MMPRES       ;INDICATE THAT MEM MGT IS PRESENT.
7128 030232  005067  152004          10$:      CLR     PASCNT          ;CLR COUNTER USED IN REPORTING ROM VERSION #.
7129 030236  000167  000006                    JMP     NEWPAS          ;SKIP AROUND THE BUS RESET, IT'S BEEN DONE.
7130
7131 030242                          NEWRES:   BRESET                  ;RESET THE BUS TO PREVENT ILLEGAL INTERRUPTS.
     030242  104433                                                                TRAP    C$RESET
7132 030244  005067  151772                    CLR     PASCNT          ; LR COUNTER USED IN REPORTING ROM VERSION #.
7133
7134 030250                          NEWPAS:
7135 030250  012767  177777  151720            MOV     #-1,UNITN       ;RESET LOGICAL DEVICE TO -1
7136                                           ;+
7137                                           ; INCREMENT THE PASS COUNTER, CORRECT FOR ANY OVERFLOW.
7138                                           ;  THIS COUNTER IS USED IN THE ROM VERSION TEST.
7139                                           ;-
7140 030256  005267  151760                    INC     PASCNT          ;INCREMENT THE PASS COUNTER.
7141 030262  001002                            BNE     GETPRM          ;BRANCH IF WE HAVE NOT YET! OVERFLOWED.
7142 030264  005367  151752                    DEC     PASCNT          ;SET PASS COUNT TO 177777 OCTAL.
7143
7144                                           ; GET THE HARDWARE PARAMETERS FOR THIS UNIT.
7145 030270                          GETPRM:
7146 030270  005267  151702                    INC     UNITN           ;INCREMENT LOGICAL DEVICE NUMBER
7147 030274  026767  151676  151510            CMP     UNITN,L$UNIT    ;SEE IF MAXIMUM UNIT NO. EXCEEDED
7148 030302  002362                            BGE     NEWPAS          ;BR IF YES
7149
7150 030304                                    GPHARD  UNITN,R1        ;GET P-TABLE POINTER INTO R1
     030304  016700  151666                                                        MOV     UNITN,R0
     030310  104442                                                                TRAP    C$GPHRD
     030312  010001                                                                MOV     R0,R1
7151 030314                                    BCOMPLETE       30$     ;BR IF DEVICE AVAILABLE
     030314  103401                                                                BCS     30$
7152 030316  000764                            BR      GETPRM          ;SKIP THIS DEVICE
7153
7154
7155                                           ;************ HARDWARE PARAMETER MOVING CODE ****************
7156 030320  012167  151654          30$:      MOV     (R1)+,CSRA      ;STORE DHU-11 CSR ADDRESS IN DEV.REG.ADDRESS TABLE
7157 030324  012102                            MOV     (R1)+,R2        ;GET THE RX INTERRUPT VECTOR ADDRESS.
7158 030326  010267  151634                    MOV     R2,RXVECA       ;STORE RX INT VECTOR ADDRESS.
7159 030332  062702  000004                    ADD     #4,R2           ;CALCULATE TX INTERRUPT VECTOR ADDRESS.
7160 030336  010267  151626                    MOV     R2,TXVECA       ;STORE TX INT VECTOR ADDRESS.
7161 030342  012167  151624                    MOV     (R1)+,ACTLNS    ;STORE DHU-11 ACTIVE LINE BIT MAP
7162 030346  112167  151622                    MOVB    (R1)+,LOPBCK    ;STORE DHU-11 LOOPBACK MODE
7163 030352  111167  151617                    MOVB    (R1),BRLEVL     ;STORE DHU-11 INTERUPT BUS REQUEST LEVEL
7164                                           ;+
7165                                           ;       CALCULTATE DEVICE REGISTER ADDRESSES,AND PUT THEM IN THE
7166                                           ;       DEVICE REGISTER ADDRESS TABLE.
7167                                           ;-
7168 030356  016701  151616                    MOV     CSRA,R1         ;COPY CSR ADDRESS
7169 030362  005201                            INC     R1              ;INCREMENT CSR ADDRESS
7170 030364  005201                            INC     R1              ; COPY BY 2.
7171 030366  012703  000007                    MOV     #7,R3           ;SET UP REGISTER COUNT
7172 030372  012702  002202                    MOV     #RBUFA,R2       ;GET LOCATION WHERE RBUF ADDRESS GOES IN TABLE
7173 030376  010122                  12$:      MOV     R1,(R2)+        ;STORE REGISTER ADDRESS IN TABLE
7174 030400  005201                            INC     R1              ;INCREMENT REGISTER ADDRESS
```

```
7175 030402  005201                        INC     R1              ; BY 2,FOR THE NEXT DEVICE REGISTER.
7176 030404  005303                        DEC     R3              ;DECREMENT REGISTER COUNT
7177 030406  001373                        BNE     12$             ;LOOP IF NOT DONE
7178
7179                                ;+
7180                                ; INITIALISE THE BMP CODE QUEUE.
7181                                ;-
7182 030410  012700  002512                 MOV     #BMPCQB,R0      ;GET THE START ADDRESS OF THE QUEUE.
7183 030414  012701  002712                 MOV     #BMPCQE,R1      ;GET THE END ADDRESS OF THE QUEUE.
7184 030420  010067  152064                 MOV     R0,BMPCQP       ;SET THE POINTER TO THE START OF THE QUEUE.
7185 030424  005020                 14$:    CLR     (R0)+           ;CLEAR OUT THE CONTENTS OF THE QUEUE.
7186 030426  020001                         CMP     R0,R1           ;CHECK IF END OF QUEUE HAS BEEN REACHED.
7187 030430  103775                        BLO     14$             ;LOOP IF NOT ALL DONE.
7188
7189                                ;+
7190                                ; REPORT THE UNIT NUMBER IF THE SOFTWARE P-TABLE QUESTION WAS ANSWERED YES,
7191                                ;  AND THE MAXIMUM UNIT NUMBER IS GREATER THAN 1.
7192 030432  032767  000020  151522 ;-
                                           BIT     #BIT4,OPTION    ;CHECK IF THE QUESTION WAS ANSWERED YES.
7193 030440  001416                        BEQ     16$             ;SKIP REPORTING UNIT NUMBER IF IT IS DISABLED.
7194 030442  026727  151344  000001        CMP     L$UNIT,#1       ;CHECK MAXIMUM NUMBER OF UNITS SELECTED.
7195 030450  003412                        BLE     16$             ;DO NOT REPORT UNIT NUMBER IF MAX NUMBER < 1.
7196 030452                                PRINTF  #MFUNIT,UNITN   ;REPORT UNIT NUMBER.
     030452  016746  151520                                                       MOV     UNITN,-(SP)
     030456  012746  005422                                                       MOV     #MFUNIT,-(SP)
     030462  012746  000002                                                       MOV     #2,-(SP)
     030466  010600                                                               MOV     SP,R0
     030470  104417                                                               TRAP    C$PNTF
     030472  062706  000006                                                       ADD     #6,SP
7197 030476                         16$:
7198
7199 030476                         ENDIT:
7200                                ;+
7201                                ; SET THE PROCESSOR PRIORITY TO DISABLE ALL INTERRUPTS.
7202                                ;-
7203 030476                                SETPRI  #PRI07                  ;SET PROCESSOR PRIORITY TO 7.
     030476  012700  000340                                                       MOV     #PRI07,R0
     030502  104441                                                               TRAP    C$SPRI
7204
7205 030504                                ENDINIT
     030504                                                             L10020:
     030504  104411                                                               TRAP    C$INIT
7206
7207            000000                     TNUM == 0               ;INITIALIZE THE ASSEMBLER TEST NUMBER VARIABLE.
```

```
                        ;*****************************************************************************
                        ;
                        ;               FVTA.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.
                        ;--

030506                          BGNAUTO
030506                                                          L$AUTO::

030506                          ENDAUTO
030506                                                          L10021:
030506  104461                                                          TRAP    C$AUTO
```

```
7239
7240                            ;**********************************************************************************************
7241                            ;
7242                            ;               FVT.CUC
7243                            ;
7244                            ;**********************************************************************************************
7245
7246
7247
7248                            .SBTTL  CLEANUP CODING SECTION
7249
7250                            ;++
7251                            ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
7252                            ; AFTER THE HARDWARE TESTS HAVE BEEN PERFORMED.
7253                            ;--
7254
7255 030510                             BGNCLN
    030510                                                                      L$CLEAN::
7256
7265
7266 030510  005767  151504              TST     CTRLCF          ;DID WE GET HERE BY CTRL-C FROM TEST?
7267 030514  001401                      BEQ     2$              ;CTRL-C FROM TEST?  NO, SKIP BUS RESET.
7268 030516                              BRESET                  ;YES, CLR ANY DMAS OR OUTSTANDING INTERRUPTS.
    030516  104433                                                              TRAP    C$RESET
7269 030520                     2$:
7270 030520                              EXIT    CLN
    030520  104432                                                             TRAP    C$EXIT
    030522  000002                                                             .WORD   L10022-.
7271
7283
7284                                    .EVEN
7285
7286 030524                             ENDCLN
    030524                                                                  L10022:
    030524  104412                                                             TRAP    C$CLEAN
```

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



                                        .SBTTL  DROP UNIT SECTION

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

 030526                                         BGNDU
 030526                                                                                  L$DU::

                                        ;*******************************************************************
                                        ;       INSERT DROP CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
                                        ;       A "DROP" COMMAND OR A "DODU" MACRO EXECUTION.  THE PURPOSE
                                        ;       OF THIS CODE IS TO DO ANY NECESSARY HOUSEKEEPING AFTER A
                                        ;       UNIT HAS BEEN DROPPED.  THIS SECTION IS OPTIONAL.
                                        ;*******************************************************************
 030526                                          PRINTF #DROP,R0        ;REPORT UNIT THAT HAS BEEN DROPPED.
 030526 010046                                                                           MOV     R0,-(SP)
 030530 012746 030552                                                                    MOV     #DROP,-(SP)
 030534 012746 000002                                                                    MOV     #2,-(SP)
 030540 010600                                                                           MOV     SP,R0
 030542 104417                                                                           TRAP    C$PNTF
 030544 062706 000006                                                                    ADD     #6,SP
 030550 000427                                   BR      EDROP          ;BRANCH AROUND THE MESSAGE.

 030552    045    101    040    DROP:   .ASCIZ/%A UNIT%D6%A DROPPED FROM FURTHER TESTING.%N/
 030555    125    116    111
 030560    124    045    104
 030563    066    045    101
 030566    040    104    122
 030571    117    120    120
 030574    105    104    040
 030577    106    122    117
 030602    115    040    106
 030605    125    122    124
 030610    110    105    122
 030613    040    124    105
 030616    123    124    111
 030621    116    107    056
 030624    045    116    000
                                        .EVEN
 030630                                 EDROP:

 030630                                         EXIT    DU
 030630 000167                                                                           .WORD   J$JMP
 030632 000000                                                                           .WORD   L10023-2-.


```

```
 7322 030634                                ENDDU
      030634                                                L10023:
      030634  104453                                              TRAP    C$DU
```

```
7324
7325                            ;*****************************************************************************************
7326                            ;
7327                            ;               FVTA.ADD
7328                            ;
7329                            ;*****************************************************************************************
7330
7331
7332
7333                            .SBTTL  ADD UNIT SECTION
7334
7335                            ;++
7336                            ; THE ADD-UNIT SECTION CONTAINS ANY CODE THE PROGRAMMER WISHES
7337                            ;  TO BE EXECUTED IN CONJUNCTION WITH THE ADDING OF A UNIT BACK
7338                            ;  TO THE TEST CYCLE.
7339                            ;--
7340
7341 030636                     BGNAU
    030636                                                              L$AU::
7342
7343                            ;*****************************************************************************************
7344                            ;       INSERT ADD CODE HERE.  THIS CODE WILL BE EXECUTED AFTER
7345                            ;       AN "ADD" COMMAND.  THE PURPOSE OF THIS CODE IS TO DO ANY
7346                            ;       HOUSEKEEPING THAT MAY BE NECESSARY AFTER A UNIT HAS BEEN ADDED.
7347                            ;       THIS SECTION IS OPTIONAL.
7348                            ;*****************************************************************************************
7349
7350 030636                     EXIT    AU
    030636  000167                                                      .WORD   J$JMP
    030640  000000                                                      .WORD   L10024-2-.
7351
7352
7353                            .EVEN
7354
7355 030642                     ENDAU
    030642                                                              L10024:
    030642  104452                                                             TRAP    C$AU
7356
```

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

030644                                          BGNTST
030644                                                                          T1::
                000001                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
030644  012767  000001  151406                   MOV     @TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (1)
030652  012767  177777  151340                   MOV     @-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
030660  012767  000145  154430                   MOV     @101.,ERRNBR    ;SET THE TEST ERROR NUMBER IN THE TABLE.
030666  012767  010041  154424                   MOV     @EM0103,ERRMSG  ;SET UP THE TEST FAILURE MESSAGE IN THE TABLE.
030674  012767  013572  154420                   MOV     @ER0101,ERRBLK  ;SET-UP THE ERROR ROUTINE IN THE ERROR TABLE.
                                        ;+
                                        ; SET UP TO CATCH ANY 004 TRAPS WHICH OCCUR:
                                        ;-
030702  016767  147076  151346                   MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR.
030710  012767  027476  147066                   MOV     @TP4RTN,4       ;SET 004 TRAP VECTOR TO OUR SERVICE RTN ADR.
030716  005005                                  CLR     R5              ;CLEAR THE ERROR FLAGS.

                                        ;+
                                        ; HERE BEGINS THE LOOP TO TEST THE REGISTERS FOR A LINE.
                                        ;    FIRST TEST THE CSR AND SET THE IND.ADR.REG (I.A.R) FIELD.
                                        ;-
030720  016700  151254                          MOV     CSRA,R0         ;SET UP CSR AS THE CKTRAP MOVE SOURCE.
030724  012701  031116                          MOV     @52$,R1         ;SET UP DESTINATION LOCATION FOR CKTRAP MOVE.
030730  004767  166414                          JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
030734  103402                                  BCS     4$              ;IF NO TRAP, BYPASS ERROR.
030736  052705  100001                          BIS     @100001,R5      ;SET FATAL READ ERROR FLAGS.
030742  042767  000017  000146  4$:             BIC     @17,52$         ;CLEAR THE I.A.R FIELD OF THE CSR DATA.
030750  010100                                  MOV     R1,R0           ;USE OLD DESTINATION FOR SOURCE OF CKTRAP MOVE.
030752  016701  151222                          MOV     CSRA,R1         ;SET UP CSR AS THE CKTRAP MOVE DESTINATION.
030756  004767  166366                          JSR     PC,CKTRAP       ;MOVE AND CHECK FOR TRAP.
030762  103403                                  BCS     6$              ;IF NO TRAP, BYPASS ERROR.
030764  052705  100002                          BIS     @100002,R5      ;SET FATAL WRITE ERROR FLAGS.
030770  000434                                  BR      40$             ;EXIT AND REPORT FATAL ERROR.
                                        ;+
                                        ;    NOW, WE TEST EACH REGISTER FOR THIS LINE.
                                        ;-
030772  012702  000010                  6$:     MOV     @8.,R2          ;INIT REGISTER COUNTER TO 8.
030776  016767  151176  000110                  MOV     CSRA,50$        ;INITIALIZE THE REGISTER POINTER.
031004  012700  031114                  8$:     MOV     @50$,R0         ;SET UP REGISTER AS THE SOURCE FOR CKTRAP MOVE.
031010  012701  031116                          MOV     @52$,R1         ;SET UP LOCAL STORAGE AS THE DES FOR CKTRAP.
031014  004767  166330                          JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
031020  103402                                  BCS     10$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
031022  052705  100001                          BIS     @100001,R5      ;SET FATAL READ ERROR FLAGS.
031026  010100                  10$:    MOV     R1,R0           ;USE OLD DEST AS SRC FOR CKTRAP MOVE.
031030  012701  031114                          MOV     @50$,R1         ;SET UP REGISTER AS THE DEST FOR CKTRAP MOVE
```

```
7414 031034  004767  166310                JSR     PC,CKTRAP       ;PERFORM THE MOVE, CHECK FOR TRAP.
7415 031040  103402                        BCS     12$             ;IF NO TRAP, BYPASS THE SETTING OF ERROR FLAGS.
7416 031042  052705  100002                 BIS     #100002,R5      ;SET FATAL WRITE ERROR FLAGS.
7417 031046  005267  000042          12$:    INC     50$             ;INCREMENT THE REGISTER
7418 031052  005267  000036                 INC     50$             ; POINTER BY 2.
7419 031056  005302                        DEC     R2              ;COUNT THE REGISTER.
7420 031060  001351                        BNE     8$              ;LOOP TO TEST THE NEXT REGISTER ADDRESS.
7421
7422                                ;+
7423                                ; DONE CHECKING DEVICE REGISTER ADDRESSES.
7424                                ;    REPORT ANY ERRORS AND EXIT.
7425                                ;-
7426
7427 031062  016767  151170  146714  40$:     MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR.
7428 031070  005705                         TST     R5              ;CHECK THE ERROR FLAGS.
7429 031072  100012                        BPL     60$             ;EXIT ROUTINE IF NO ERRORS.
7430                                ;+
7431                                ; REPORT "DEVICE REGISTER ACCESS TEST FAILED"
7432                                ;-
7433 031074                                ERROR
     031074  104460                                                                TRAP    C$ERROR
7434
7435
7436 031076                                DODU    UNITN           ;DROP THIS UNIT FROM FUTHER TESTING.
     031076  016700  151074                                                        MOV     UNITN,R0
     031102  104451                                                                TRAP    C$DODU
7437 031104  005067  151110                 CLR     CTRLCF          ;INDICATE NO CTRL-C ABORT FROM TEST.
7438 031110                                DOCLN                   ;ABORT THIS SUB PASS.
     031110  104444                                                                TRAP    C$DCLN
7439 031112  000402                        BR      60$             ;
7440
7441
7442 031114  000000                ;****************** LOCAL STORAGE. ***********************************************
7443 031116  000000                50$:    .WORD   0               ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
7444                                52$:    .WORD   0               ;STORAGE FOR THE SOURCE OR DEST OF THE CKTRAP MOVE.
7445                                ;******************** END ********************************************************
7446 031120  005067  151074          60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
7447 031124                                 ENDTST
     031124                                                                L10025:
     031124  104401                                                                TRAP    C$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.  THIS TEST SETS UP THE LINES
                                ;*      FOR: 1 STOP BIT, NO PARITY AND  8 BITS/CHARACTER.
                                ;*      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.
                                ;*
                                ;-- ************************************************************************
031126                                  BGNTST
031126                                                                          T2::
031126                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
031126  012700  000240                                                        MOV     #PRIO5,R0
031132  104441                                                                TRAP    C$SPRI
        000002                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031134  012767  000002  151116           MOV    #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (94)
                                ;*
                                ; VERIFY THAT THE TEST SHOULD BE PERFORMED.  MUST HAVE THE FOLLOWING:
                                ;       KEYBOARD ECHO LOOPBACK SELECTED.
                                ;       MANUAL INTERVENTION ALLOWED.
                                ;-
031142  126727  151026  000005           CMPB   LOPBCK,#5        ;TEST THE LOOPBACK TYPE INDICATOR.
031150  001402                          BEQ    2$               ;K8D ECHO LPBCK SELECTED? YES, CONTINUE TEST.
031152                                  EXIT   TST              ;NO, ABORT THE TEST.
031152  104432                                                                TRAP    C$EXIT
031154  000212                                                                .WORD   L10026 .
031156                          2$:     MANUAL                  ;CHECK FOR MANUAL INTERVENTION ALLOWED.
031156  104450                                                                TRAP    C$MANI
031160                                  BCOMPLETE       4$      ;MANUAL INTERVENTION ALLOWED? YES, DO TEST.
031160  103402                                                                BCS     4$
031162                                  EXIT    TST             ;NO, ABORT THE TEST.
031162  104432                                                                TRAP    C$EXIT
031164  000202                                                                .WORD   L10026-.

031166  012767  177777  151024   4$:      MOV    #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
031174  012767  000001  154112           MOV    #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
031202  012767  022271  154106           MOV    #9401.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
031210  012767  013065  154102           MOV    #EM9401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
031216  005067  151256                   CLR    ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 9401 <<<<<.
                                ;-
031222  004767  166202                  JSR    PC,CLNRST       ;RESET THE DMU-11, REPORT ANY ERRORS FOUND.
031226  103402                          BCS    .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
031230  000167  000120                  JMP    60$             ;RESET FAILURE, ABORT THIS TEST.
                                ;*
                                ; PRINT THE TEST NAME.
                                ;-
```

```
031234                          PRINTF  #EF0503,#EM9401
031234  012746  013065                                                          MOV     #EM9401,-(SP)
031240  012746  005453                                                          MOV     #EF0503, (SP)
031244  012746  000002                                                          MOV     #2,-(SP)
031250  010600                                                                  MOV     SP,R0
031252  104417                                                                  TRAP    C$PNTF
031254  062706  000006                                                          ADD     #6,SP

                        ;+
                        ; SET UP THE DUT UARTS WITH THE PROPER LINE PARAMETERS.
                        ;    GET THE DESIRED BAUDRATE FROM THE OPERATOR.
                        ;    CALCULATE PROPER DUT LPR CONTENTS.
                        ;    SET UP THE DUT LPR REGISTERS.
                        ;    GET THE PROPER DUT LNCTRL REGISTER CONTENTS.
                        ;    SET UP THE DUT LNCTRL REGISTERS.
                        ;-
031260  004767  167134          JSR     PC,GETBDR
031264  010100                  MOV     R1,R0
031266  006301                  ASL     R1
031270  006301                  ASL     R1
031272  006301                  ASL     R1              ;GET DUPLICATE COPIES OF BAUDRATE CODE
031274  006301                  ASL     R1              ;  IN THE UPPER BYTE OF THE NEW
031276  050100                  BIS     R1,R0           ;  LPR CONTENTS.
031300  000300                  SWAB    R0
031302  042700  000377          BIC     #377,R0         ;SET UP 1 STOP BIT, NO PARITY, 8 BITS/CHAR
031306  052700  000030          BIS     #30,R0          ;  IN THE LPR CONTENTS.

031312  016705  150654          MOV     ACTLNS,R5       ;GET THE ACTIVE LINES BIT MAP.
031316  004767  175666          JSR     PC,WTWLPR       ;SET UP THE DUT LPR REGISTERS FOR ACTIVE LINES.

031322  012700  011304          MOV     #11304,R0       ;SET UP DTR, RTS, REMOTE LPBK, AND RX ENABLE.
031326  004767  175626          JSR     PC,WTWLNC       ;SET UP THE DUT LNCTRL REGS FOR ACTIVE LINES.

                        ;+
                        ; WAIT FOR THE OPERATOR TO TERMINATE THE TEST.
                        ;   PROMPT "TYPE <CR> TO TERMINATE THE TEST:"
                        ;-
031332  012767  000001  150670  MOV     #1,GMANWD       ;SET UP DEFAULT ANSWER TO YES.
031340                          GMANIL  TERMSG,GMANWD,1,YES
031340  104443                                                          TRAP    C$GMAN
031342  000404                                                          BR      10000$
031344  002230                                                          .WORD   GMANWD
031346  000130                                                          .WORD   T$CODE
031350  013531                                                          .WORD   TERMSG
031352  000001                                                          .WORD   1
031354                                                          10000$:

                        ;+
                        ; WE GOT A RESPONSE FROM THE OPERATOR, SO TERMINATE THE TEST.
                        ;-
031354                  60$:    SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
031354  012700  000340                                                  MOV     #PRIO7,R0
031360  104441                                                          TRAP    C$SPRI
031362  005067  150632          CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
031366                          ENDTST
031366                                                          L10026:
031366  104401                                                          TRAP    C$ETST
```

```
                        .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.
                        ;*
                        ;-- ************************************************************************
031370                          BGNTST
031370                                                                          T3::
031370                          SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
031370  012700  000240                                                        MOV     #PRIO5,R0
031374  104441                                                                TRAP    C$SPRI
        000003
                                 TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
031376  012767  000003  150654    MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (89)
                        ;*
                        ; VERIFY THAT THE TEST SHOULD BE PERFORMED.  MUST HAVE THE FOLLOWING:
                        ;       MODEM LOOPBACK SELECTED.
                        ;       MANUAL INTERVENTION ALLOWED.
                        ;-
031404  126727  150564  000004    CMPB    LOPBCK,#4      ;TEST THE LOOPBACK TYPE INDICATOR.
031412  001402                  BEQ     2$             ;MODEM LOOPBACK SELECTED? YES, CONTINUE TEST.
031414                          EXIT    TST            ;NO, ABORT THE TEST.
031414  104432                                                                TRAP    C$EXIT
031416  000702                                                                .WORD   L10027-.
031420                  2$:     MANUAL                 ;CHECK FOR MANUAL INTERVENTION ALLOWED.
031420  104450                                                                TRAP    C$MANI
031422                          BCOMPLETE       4$     ;MANUAL INTERVENTION ALLOWED? YES, DO TEST.
031422  103402                                                                BCS     4$
031424                          EXIT    TST            ;NO, ABORT THE TEST.
031424  104432                                                                TRAP    C$EXIT
031426  000672                                                                .WORD   L10027 .
031430  012767  177777  150562  4$:     MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
```

```
 7586 031436  012767  000001  153650          MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
 7587 031444  012767  021305  153644          MOV     #8901 ,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
 7588 031452  012767  011211  153640          MOV     #EM8901,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
 7589 031460  005067  151014                  CLR     ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
 7590
 7591                                 ;+
 7592                                 ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
 7593                                 ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
 7594                                 ; THIS SUBROUTINE REPORTS ERROR >>>>> 8901 <<<<<.
 7595                                 ;-
 7596 031464  004767  165740                  JSR     PC,CLNRST       ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
 7597 031470  103402                          BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
 7598 031472  000167  000610                  JMP     60$             ;RESET FAILURE, ABORT THIS TEST.
 7599
 7600                                 ;+
 7601                                 ; SET UP FOR TRANSMIT AND RECEIVE INTERRUPTS.
 7602                                 ;-
 7603 031476                                  SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      031476  012700  000340                                                  MOV     #PRI07,R0
      031502  104441                                                          TRAP    C$SPRI
 7604 031504                                  SETVEC  TXVECA,#TXDMA,#PRI06    ;SELECT DMA TX INT SERVICE RTN.
      031504  012746  000300                                                  MOV     #PRI06,-(SP)
      031510  012746  027520                                                  MOV     #TXDMA,-(SP)
      031514  016746  150450                                                  MOV     TXVECA,-(SP)
      031520  012746  000003                                                  MOV     #3,-(SP)
      031524  104437                                                          TRAP    C$SVEC
      031526  062706  000010                                                  ADD     #10,SP
 7605 031532                                  SETVEC  RXVECA,#RXCHRS,#PRI06   ;SELECT RX INT SERVICE RTN.
      031532  012746  000300                                                  MOV     #PRI06,-(SP)
      031536  012746  027310                                                  MOV     #RXCHRS,-(SP)
      031542  016746  150420                                                  MOV     RXVECA,-(SP)
      031546  012746  000003                                                  MOV     #3,-(SP)
      031552  104437                                                          TRAP    C$SVEC
      031554  062706  000010                                                  ADD     #10,SP
 7606 031560                                  SETPRI  #PRI04          ;ALLOW INTERRUPTS.
      031560  012700  000200                                                  MOV     #PRI04,R0
      031564  104441                                                          TRAP    C$SPRI
 7607                                 ;+
 7608                                 ; CLEAR THE CUMULATIVE ERROR COUNTERS (ONE FOR EACH LINE).
 7609                                 ;-
 7610 031566  012700  003302                   MOV     #ERCNTB,R0
 7611 031572  004767  165654                  JSR     PC,CLR16W       ;CLEAR THE RX ERROR COUNTERS TABLE.
 7612                                 ;+
 7613                                 ; PRINT THE THE TEST NAME.
 7614                                 ;-
 7615 031576                                  PRINTF  #EF0503,#EM8901
      031576  012746  011211                                                  MOV     #EM8901,-(SP)
      031602  012746  005453                                                  MOV     #EF0503,-(SP)
      031606  012746  000002                                                  MOV     #2,-(SP)
      031612  010600                                                          MOV     SP,R0
      031614  104417                                                          TRAP    C$PNTF
      031616  062706  000006                                                  ADD     #6,SP
 7616                                 ;+
 7617                                 ; PREPARE TO CALL THE SET UP ROUTINE.
 7618                                 ;    GET THE DESIRED BAUDRATE FROM THE OPERATOR.
 7619                                 ;    CALCULATE PROPER DUT LPR CONTENTS.
 7620                                 ;    CALCULATE THE PROPER RX TIME-OUT VALUE FOR THIS SPEED.
```

```
7621                                    ;    SET UP THE BIT MAP OF UNUSED TX/RX BITS.
7622                                    ;-
7623 031622  004767  166572                 JSR     PC,GETBDR
7624 031626  010100                         MOV     R1,R0
7625 031630  006301                         ASL     R1
7626 031632  006301                         ASL     R1
7627 031634  006301                         ASL     R1              ;GET DUPLICATE COPIES OF BAUDRATE CODE
7628 031636  006301                         ASL     R1              ;  IN THE UPPER BYTE OF THE NEW
7629 031640  050001                         BIS     R0,R1           ;  LPR CONTENTS.
7630 031642  000301                         SWAB    R1
7631 031644  042701  000377                BIC     #377,R1         ;SET UP 1 STOP BIT, NO PARITY, 8 BITS/CHAR
7632 031650  052701  000030                BIS     #30,R1          ;  IN THE LPR CONTENTS.
7633
7634 031654  004767  166750                JSR     PC,GETTIM       ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
7635 031660  012767  177400  150344        MOV     #177400,IBM     ;FORM BIT MAP OF UNUSED TX/RX BITS.
7636                                    ;+
7637                                    ;    SET UP A 256 BYTE DATA PATTERN.
7638                                    ;-
7639 031666  005003                         CLR     R3              ;PREPARE TO START DATA PATTERN AT 255.
7640 031670  012702  003602                MOV     #BUFBAS,R2      ;GET THE BASE OF THE DATA PATTERN BUFFER.
7641 031674  010204                         MOV     R2,R4
7642 031676  105303                 6$:     DECB    R3              ;GET THE NEXT BYTE OF THE DATA PATTERN.
7643 031700  110324                         MOVB    R3,(R4)+        ;WRITE A BYTE OF THE DATA PATTERN.
7644 031702  105703                         TSTB    R3              ;CHECK FOR DONE WRITING DATA PATTERN.
7645 031704  001374                        BNE     6$              ;DATA PATTERN DONE?  NO, LOOP TO DO NEXT BYTE.
7646                                                                ;YES, WRITE 32 BYTE OVERFLOW REGION.
7647 031706  010205                         MOV     R2,R5           ;PREPARE SOURCE POINTER.
7648 031710  012700  000020                MOV     #16.,R0         ;PREPARE LOOP COUNTER.
7649 031714  012524                 8$:     MOV     (R5)+,(R4)+     ;WRITE 2 BYTES OF THE OVERFLOW PATTERN.
7650 031716  005300                         DEC     R0              ;COUNT THESE 2 BYTES.
7651 031720  001375                        BNE     8$              ;16 WORDS WRITTEN?  NO, LOOP TO WRITE ANOTHER.
7652                                                                ;YES, COMPLETE DATA PATTERN IS DONE.
7653 031722  012703  000400                MOV     #256.,R3        ;SET DATA PATTERN LENGTH TO 256.
7654                                    ;+
7655                                    ; SET THE DU1 RTS AND DTR BITS FOR THE ACTIVE LINES.
7656                                    ;-
7657 031726  012700  011000                MOV     #11000,R0       ;SPECIFY TO SET RTS AND DTR.
7658 031732  016705  150234                MOV     ACTLNS,R5       ;SPECIFY ACTIVE LINES.
7659 031736  004767  175216                JSR     PC,WTWLNC       ;SET DUT RTS AND DTR ON ALL ACTIVE LINES.
7660                                    ;+
7661                                    ; WAIT FOR THE OPERATOR TO ESTABLISH THE MODEM CONNECTION.
7662                                    ;   PROMPT "TYPE <CR> WHEN MODEM LINK ESTABLISHED:"
7663                                    ;-
7664 031742  012767  000001  150260        MOV     #1,GMANWD       ;SET UP DEFAULT ANSWER TO YES.
7665 031750                             GMANIL  EMLMSG,GMANWD,1,YES
     031750  104443                                                                TRAP    C$GMAN
     031752  000404                                                                BR      10000$
     031754  002230                                                                .WORD   GMANWD
     031756  000130                                                                .WORD   T$CODE
     031760  013166                                                                .WORD   EMLMSG
     031762  000001                                                                .WORD   1
     031764                                                                 10000$:
7666                                    ;+
7667                                    ; REPORT THE STATE OF THE MODEM STATUS SIGNALS.
7668                                    ; SET DEFAULT OF PRINTING MODEM STATUS AFTER EVERY DATA PATTERN.
7669                                    ;-
7670 031764  004767  167326                JSR     PC,MSSRPT
```

```
 7671 031770  012767  000001  150246          MOV     #1,PMSFLG
 7672                                 ;+
 7673                                 ; ASK OPERATOR FOR THE NUMBER OF DATA PATTERNS TO SEND.
 7674                                 ;   PROMPT: "NUMBER OF 256 BYTE DATA PATTERNS TO SEND ON EACH SELECTED LINE
 7675                                 ;                          (1-255, 0=SEND UNTIL ^C): (D)  1 ?"
 7676                                 ;-
 7677 031776  012767  000001  150224  10$:     MOV     #1,GMANWD       ;SET DEFAULT NUMBER OF PATTERNS TO 1.
 7678 032004                          GMANID  NDPMSG,GMANWD,D,377,0,255,YES
      032004  104443                                                                  TRAP    C$GMAN
      032006  000406                                                                  BR      10001$
      032010  002230                                                                  .WORD   GMANWD
      032012  000052                                                                  .WORD   T$CODE
      032014  013316                                                                  .WORD   NDPMSG
      032016  000377                                                                  .WORD   377
      032020  000000                                                                  .WORD   T$LOLIM
      032022  000255                                                                  .WORD   T$HILIM
      032024                                                                  10001$:
 7679 032024  016704  150200              MOV     GMANWD,R4
 7680 032030  005005                      CLR     R5              ;CLEAR THE DATA PATTERN COUNTER.
 7681 032032  005067  150166              CLR     FERROR          ;CLEAR THE "AT LEAST ONE ERROR" FLAG
 7682                                 ;+
 7683                                 ; ASK IF MODEM STATUS SIGNALS SHOULD BE REPORTED AFTER EACH DATA PATTERN.
 7684                                 ;   PROMPT: "PRINT MODEM STATUS SIGNAL REPORT AFTER EACH PATTERN: (L)  Y ?"
 7685                                 ;    USE LAST RESPONSE AS DEFAULT (DEFAULT OF YES THE FIRST TIME).
 7686                                 ;-
 7687 032036                          GMANIL  PMSMSG,PMSFLG,1,YES
      032036  104443                                                                  TRAP    C$GMAN
      032040  000404                                                                  BR      10002$
      032042  002244                                                                  .WORD   PMSFLG
      032044  000130                                                                  .WORD   T$CODE
      032046  013444                                                                  .WORD   PMSMSG
      032050  000001                                                                  .WORD   1
      032052                                                                  10002$:
 7688
 7689                                 ;+
 7690                                 ; SET UP THE DUT AND TX/RX VARIABLES.
 7691                                 ;       R1 - TX, RX LPR CONTENTS.
 7692                                 ;       R2 - START ADDRESS OF DATA PATTERN TO TX/RX.
 7693                                 ;       R3 - LENGTH OF DATA PATTERN.
 7694                                 ; SEND THE DATA.
 7695                                 ;-
 7696 032052  005205                  12$:     INC     R5              ;COUNT THIS DATA PATTERN.
 7697 032054  004767  166746          JSR     PC,MODSUP       ;SET UP THE DUT AND TX/RX VARIABLES.
 7698
 7699 032060  004767  170656          JSR     PC,PUFIFO       ;PURGE THE DUT RECEIVE CHARACTER FIFO.
 7700 032064  103110                  BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
 7701
 7702 032066  004767  171134          JSR     PC,PURRXB       ;PURGE THE RX CHAR BUFFER IN MEMORY.
 7703 032072  004767  166610          JSR     PC,INIDMA       ;SEND THE FIRST BATCH OF DATA PATTERNS.
 7704 032076  012767  021306  153212   MOV     #8902.,ERRNBR   ;SET ERROR NUMBER TO 8905.
 7705                                 ;+
 7706                                 ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 8902 THRU 8907 <<<<<.
 7707                                 ;-
 7708 032104  004767  171152              JSR     PC,RDCHRS       ;READ AND VERIFY THE RX CHARACTERS.
 7709
 7710 032110  005767  150110              TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
 7711 032114  001404                      BEQ     13$             ;BRANCH IF IT HASN'T.
```

```
7712 032116  032767  000100  150036          BIT     @BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
7713 032124  001430                          BEQ     16$             ;BRANCH TO THE "EXIT TEST ?" QUESTION IF NOT.
7714
7715 032126  012767  021314  153162  13$:     MOV     @8908.,ERRNBR   ;SET ERROR NUMBER TO 8908.
7716                                        ;+
7717                                        ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 8908 THRU 8911 <<<<<.
7718                                        ;-
7719 032134  004767  174200                  JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM RX/RX.
7720                                        ;+
7721                                        ; REPORT END OF DATA PATTERN IF ALLOWED.
7722                                        ;    "MODEM LOOPBACK TEST STATUS REPORT:  PATTERN @NNN (D) COMPLETED."
7723                                        ; REPORT THE MODEM STATUS SIGNAL STATES IF REQUESTED.
7724                                        ;-
7725 032140                                  PRINTX  @EDPFMT,R5
     032140  010546                                                                 MOV     R5,-(SP)
     032142  012746  007733                                                         MOV     @EDPFMT,-(SP)
     032146  012746  000002                                                         MOV     @2,-(SP)
     032152  010600                                                                 MOV     SP,R0
     032154  104415                                                                 TRAP    C$PNTX
     032156  062706  000006                                                         ADD     @6,SP
7726 032162  005767  150056                      TST     PMSFLG          ;CHECK THE "PRINT MODEM STATUS" FLAG.
7727 032166  001402                              BEQ     14$             ;PRINT MODEM STATUS?  NO, SKIP PRINTING.
7728 032170  004767  167122                      JSR     PC,MSSRPT       ;REPORT THE MODEM STATUS.
7729                                        ;+
7730                                        ; IF THERE ARE MORE DATA PATTERNS TO SEND, LOOP BACK TO SEND AGAIN.
7731                                        ;-
7732 032174  005304                      14$:     DEC     R4              ;COUNT THIS DATA PATTERN.
7733 032176  001403                              BEQ     16$             ;LAST DATA PAT SENT?  YES, PROMPT FOR EXIT.
7734 032200  100324                              BPL     12$             ;NO, CONTINOUS SENDING?  NO, SEND NEXT PAT.
7735 032202  005204                              INC     R4              ;YES, RESTORE PATTERN COUNTER.
7736 032204  000722                              BR      12$             ;GO TO SEND NEXT DATA PATTERN.
7737                                        ;+
7738                                        ; PROMPT FOR EXIT OF THE TEST OR SENDING OF MORE DATA PATTERNS.
7739                                        ;   PROMPT: "EXIT THE TEST (N = LOOP BACK TO SEND MORE DATA): (L) Y ?"
7740                                        ;-
7741 032206  012767  000001  150014  16$:     MOV     @1,GMANWD       ;SET DEFAULT ANSWER TO YES.
7742 032214                                  GMANIL  EXTMSG,GMANWD,1,YES
     032214  104443                                                                 TRAP    C$GMAN
     032216  000404                                                                 BR      10003$
     032220  002230                                                                 .WORD   GMANWD
     032222  000130                                                                 .WORD   T$CODE
     032224  013235                                                                 .WORD   EXTMSG
     032226  000001                                                                 .WORD   1
     032230                                                                 10003$:
7743 032230  026727  147774  000001              CMP     GMANWD,@1       ;CHECK OPERATOR RESPONSE.
7744 032236  001257                              BNE     10$             ;EXIT RESPONSE?  NO, LOOP TO SEND MORE DATA.
7745                                                                     ;NO, EXIT ROUTINE.
7746                                        ;+
7747                                        ; ALL DONE.  HAVE BEEN TOLD TO EXIT.
7748                                        ;    CLEAR DEVICE DTR AND RTS SIGNALS.
7749                                        ;    DISABLE INTERRUPTS.
7750                                        ;    CLEAR THE INTERRUPT VECTORS.
7751                                        ;    REPORT ANY NECESSARY ERROR SUMMARIES.
7752                                        ;-
7753 032240  005000                              CLR     R0              ;INDICATE TO CLEAR ALL LNCTRL BITS.
7754 032242  012705  177777                      MOV     @MAPLNS,R5      ;INDICATE TO CLEAR FOR ALL LINES.
7755 032246  004767  174706                      JSR     PC,WTWLNC       ;CLEAR ALL THE RTS AND DTR SIGNALS
```

```
 7756
 7757 032252                                SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
      032252  012700  000340                                                MOV     #PRIO7,R0
      032256  104441                                                       TRAP    C$SPRI
 7758 032260                                CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
      032260  016700  147704                                                MOV     TXVEC,R0
      032264  104436                                                       TRAP    C$CVEC
 7759 032266                                CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
      032266  016700  147674                                                MOV     RXVECA,R0
      032272  104436                                                       TRAP    C$CVEC
 7760
 7761 032274  012767  021320  153014          MOV     #8912.,ERRNBR   ;SELECT NUMBER 8912 FOR THE NEXT ERROR REPORT.
 7762 032302  004767  171614                JSR     PC,REPSMR       ;REPORT ERROR SUMMARIES IF CALLED FOR.
 7763 032306                          60$:   SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
      032306  012700  000340                                                MOV     #PRIO7,R0
      032312  104441                                                       TRAP    C$SPRI
 7764 032314  005067  147700                 CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 7765 032320                                ENDTST
      032320                                                         L10027:
      032320  104401                                                       TRAP    C$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 18 BIT OR 16 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, IF REQUESTED.
                                        ;*
                                        ;*
                                        ;-- ******************************************************************
 032322                                         BGNTST
 032322                                                                                 T4::

 032322                                         SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS
 032322  012700  000240                                                                         MOV     #PRIO5,R0
 032326  104441                                                                                 TRAP    C$SPRI

         000004                                   TNUM ==       TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER
 032330  012767  000004  147722                   MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER
 032336  012767  177777  147654                   MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST
 032344  012767  000001  152742                   MOV     #1,ERRTYP       ;SET ERROR TYPE AS FATAL IN ERROR TABLE
 032352  012767  010461  152736                   MOV     #4401.,ERRNBR   ;SET ERROR NUMBER TO 4401
 032360  012767  010166  152732                   MOV     #EM4401,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN TABLE
 032366  012767  014124  152726                   MOV     #ER0503,ERRBLK  ;SELECT THE CORRECT ERROR REPORTING ROUTINE

                                        ;+
                                        ; CLEAR THE SUCCESS FLAG TO INDICATE TEST FAILURE IN CASE IT DOES
                                        ;-

 032374  005067  001374                           CLR     SUCSS           ;INDICATE FAILURE , IN CASE THE DUT FAILS

                                        ;+
                                        ; RESET THE DUT TO A KNOWN STATE,REMOVE THE STATUS CODES FROM THE FIFO.
                                        ; CLEAR TX AND RX INTERUPT ENABLE BITS IN THE CSR
                                        ;-

 032400  004767  165024                           JSR     PC,CLNRST       ;RESET THE DHU , REPORT ANY ERRORS
 032404  103402                                   BCS     .+6             ;SKIP EXIT OF TEST IF NO FATAL ERROR FOUND.
 032406  000167  001324                           JMP     60$             ;EXIT THE TEST, FATAL ERROR WAS FOUND.

                                        ;+
                                        ; SET UP THE 004 TRAP VECTOR TO POINT TO OUR TRAP SERVICE ROUTINE.
                                        ;-
 032412  016767  145366  147636                   MOV     4,TP4VEC        ;SAVE THE EXISTING 004 TRAP VECTOR
 032420  012767  027476  145356                   MOV     #TP4RTN,4       ;POINT THE VECTOR AT OUR SERVICE ROUTINE.

                                        ;+
                                        ; DETERMINE WHETHER MEMORY MANAGEMENT IS PRESENT
                                        ;-

 032426  005767  147666                           TST     MMPRES          ;IF MEM MGT IS PRESENT THEN
 032432  001007                                   BNE     1$              ;AVOID SETTING THE DMA TEST ADDR FOR
                                                                          ;A 16 BIT MACHINE.
 032434  012767  001252  147560                   MOV     #1252,DMTSTA    ;SET UP THE FIRST DMA TEST ADDR FOR
```

```
7821
7822 032442  012767  000021  001316          MOV     #17.,BITSTD     ;A 16 BIT MACHINE
7823 032450  000442                          BR      10$             ;SET THE BITS TESTED TO 16 + 1
7824                                                                 ;SINCE MEM MGT ISN'T PRESENT
7825                                                                 ;THERE'S NO NEED TO SET UP THE MEM MGT RLGS.
7826                                 ;+
7827                                 ; SET UP THE HIGHEST POSSIBLE TEST ADDRESS IN DMTSTA
7828                                 ;-
7829 032452  012767  005252  147542  1$:     MOV     #5252,DMTSTA    ;SET UP THE FIRST DMA TEST ADDRESS FOR THE
7830                                                                 ; 18 BIT MACHINE.
7831 032460  012767  000023  001300          MOV     #19.,BITSTD     ;SET THE BITS TESTED TO 18 BITS + 1.
7832                                 ;+
7833                                 ; SET UP THE PARS 0 THROUGH 6 TO RELOCATE TO THE SAME ADDRESS
7834                                 ;-
7835 032466  005000                          CLR     R0              ;SET THE PAGE BASE ADDRESS TO ZERO
7836 032470  012701  002324                  MOV     #PARATB,R1      ;POINT AT THE START OF THE PAR ADDRESS TABLE
7837 032474  010031                  2$:     MOV     R0,@(R1)+       ;LOAD THE PAR
7838 032476  062700  000200                  ADD     #200,R0         ;CALCULATE THE NEXT PAGE ADDRESS
7839 032502  022701  002342                  CMP     #PAR7A,R1       ;LOOP UNTIL PARS 0 THROUGH 6
7840 032506  001372                          BNE     2$              ;ARE LOADED.
7841
7842                                 ;+
7843                                 ; SET UP THE INPUT/OUTPUT PAGE PAR TO THE TOP 4KW OF PHYSICAL MEMORY
7844                                 ;-
7845
7846 032510  012777  177600  147624          MOV     #177600,@PAR7A  ;SET PAR #7 TO POINT AT THE TOP 4KW
7847
7848                                 ;+
7849                                 ; SET UP THE PDRS FOR , NO ABORT/TRAP,UPWARD EXPANSION,128 BLOCKS PER PAGE
7850                                 ;-
7851
7852 032516  012700  077406                  MOV     #77406,R0       ;BIT PATTERN FOR THE PDRS
7853 032522  012701  002344                  MOV     #PDRATB,R1      ;POINT AT START OF PDR ADDR TABLE
7854 032526  010031                  4$:     MOV     R0,@(R1)+       ;
7855 032530  022701  002364                  CMP     #PDRATE,R1      ;LOOP UNTIL ALL PDRS HAVE
7856 032534  001374                          BNE     4$              ;BEEN SET UP.
7857
7858                                 ;+
7859                                 ; SET THE MEM MGT STATUS REG #3 FOR, 18 BIT ADDRESSSING,
7860                                 ;  NO UNIBUS MAPPING, NO D SPACE, IN CASE THIS REG EXISTS.
7861                                 ; NOT ALL UNIBUS MACHINES HAVE MEM MGT STATUS REG #3, SO THE CKTRAP ROUTINE
7862                                 ; MUST BE USED TO CATCH AN 004 TRAPS THAT OCCUR IF THE REG DOES NOT EXIST.
7863                                 ;-
7864 032536  005067  001220                  CLR     70$             ;SET UP THE SOURCE OPERAND
7865 032542  012700  033762                  MOV     #70$,R0         ;SET UP THE SOURCE ADDR FOR THE MOVE.
7866 032546  016701  147544                  MOV     MMSR3,R1        ;SET UP THE DESTINATION ADDR FOR THE MOVE.
7867 032552  004767  164572                  JSR     PC,CKTRAP       ;PERFORM THE MOVE AND CATCH ANY 004 TRAPS.
7868
7869                                 ;+
7870                                 ; TRY AND FIND A COMPLEMENTARY PAIR OF ADDRESSES WITHIN THE MEMORY AND SAVE
7871                                 ;  THE CONTENTS OF THE TWO AREAS. THE TEST IS ABANDONED IF A COMPLEMENTARY
7872                                 ;  PAIR HAS NOT BEEN FOUND BEFORE THE AREA OF MEMORY CONTAINING THE
7873                                 ;  DIAGNOSTIC IS ENCOUNTERED.
7874                                 ;-
7875 032556  012767  027454  145220  10$:     MOV     #TP4BRT,4       ;CHANGE THE 004 TRAP VECTOR TO POINT TO
7876                                                                 ;TP4BRT SINCE THIS IS THE ROUTINE ASSOCIATED
7877                                                                 ;WITH THE BYTE SUBROUTINE CKTRPB.
```

```
7878
7879 032564                                    MEMORY  FFREM           ;GET THE ADDRESS OF THE FIRST FREE WORD
     032564  104431                                                     TRAP    C$MEM
     032566  010067  147434                                             MOV     R0,FFREM
7880                                                                    ;OF MEMORY ABOVE THE DIAGNOSTIC.
7881
7882 032572  012701  003602                      MOV     #BUFBAS,R1     ;POINT AT THE BUFFER WHERE THE CONTENTS OF
7883                                                                    ;THE MEMORY BEING READ ARE TO BE SAVED.
7884 032576  005004                              CLR     R4             ;CLEAR THE COMPLEMENTARY PAIR INDICATOR (CPI)
7885
7886
7887 032600  005204                  12$:        INC     R4             ;INCREMENT THE CPI
7888 032602  005005                              CLR     R5             ;INDICATE THAT A SAVE OF THE DATA AT
7889                                                                    ;(DMTSTA) IS REQUIRED
7890 032604  012703  000020                      MOV     #16.,R3        ;SET THE NUMBER OF BYTES TO BE READ
7891 032610  004767  165046                      JSR     PC,DMRW        ;SAVE THE DATA CONTAINED AT ADDRESS DMTSTA.
7892 032614  012701  004202                      MOV     #BUFMID,R1     ;POINT AT SECOND STORAGE AREA
7893 032620  005767  147430                      TST     TP4FLG         ;IF WE HAVE VALID MEMORY THEN AVOID CLEARING
7894 032624  001403                              BEQ     14$            ;THE CPI AND RESETING THE SAVE AREA ADDR
7895
7896 032626  005004                              CLR     R4             ;CLEAR THE CPI.
7897 032630  012701  003602                      MOV     #BUFBAS,R1     ;RESET THE ADDR FOR THE SAVED DATA STORE
7898 032634  022704  000002          14$:        CMP     #2,R4          ;IF A PAIR OF COMPLEMENTARY ADDRESSSES HAVE
7899                                                                    ;BEEN FOUND THEN
7900 032640  001447                              BEQ     17$            ;GO AND WRITE THE TEST DATA TO THESE ADDRS.
7901 032642  016767  147354  001122              MOV     DMTSTA,ODTSTA  ;SAVE THE OLD DMTSTA
7902 032650  000241                              CLC                    ;CLEAR CARRY READY FOR THE ROTATION
7903 032652  006067  147344                      ROR     DMTSTA         ;COMPLEMENT THE DMTSTA TO PRODUCE THE NEXT
7904                                                                    ; DMA TEST ADDR.
7905 032656  005367  001104                      DEC     BITSTD         ;DECREMENT THE NUMBER OF BITS TESTED COUNT
7906
7907                                             ;+
7908                                             ; CHECK THAT THE NEW DMTSTA IS NOT INSIDE THE DIAGNOSTIC PROGRAM
7909                                             ;-
7910
7911 032662  032767  176000  147332              BIT     #176000,DMTSTA ;IS THE DMTSTA > 1252 . IF IT IS THEN WE'RE
7912                                                                    ; SAFE SO,
7913 032670  001343                              BNE     12$            ;BRANCH AND CONTINUE WITH THE SEARCH
7914 032672  004767  164712                      JSR     PC,DM16B       ;CONVERT THE DMTSTA TO A PHYSICAL ADDR.
7915 032676  020067  147324                      CMP     R0,FFREM       ;ARE WE INSIDE THE DIAGNOSTIC REGION ?
7916 032702  103336                              BHIS    12$            ;NO . THEN BRANCH AND CONTINUE WITH THE SEARCH
7917
7918                                             ;+
7919                                             ;SINCE WE ARE NOW INSIDE THE DIAGNOSTIC, WE INCREMENT BIT #14 OF THE DMTSTA
7920                                             ;PHYSICAL ADDRESSS AND IF WE'RE STILL INSIDE THE DIAGNOSTIC WE ABANDON THE
7921                                             ;TEST. ONCE WE ARE IN THIS REGION WE ARE ONLY ABLE TO TEST THE LOWEST 14 BITS.
7922                                             ;-
7923
7924
7925 032704  022767  000252  147310              CMP     #252,DMTSTA    ;IF THE BIT HAS ALREADY BEEN SET THEN
7926 032712  001014                              BNE     15$            ;ABANDON THE TEST,AFTER REPORTING THE ERROR ,
7927                                                                    ; BECAUSE NO SUITABLE MEMORY HAS BEEN FOUND.
7928 032714  012767  000652  147300              MOV     #652,DMTSTA    ;SET THE BIT
7929 032722  062700  040000                      ADD     #40000,R0      ;ADD THE BIT INTO THE PHYSICAL ADDR
7930 032726  020067  147274                      CMP     R0,FFREM       ;IF WE'RE NOW STILL INSIDE THE DIAGNOSTIC THEN
7931 032732  103404                              BLO     15$            ;REPORT ERROR AND ABANDON THE TEST.
7932 032734  012767  000016  001024              MOV     #14.,BITSTD    ;OTHERWISE SET THE BITS TESTED TO 14 BITS.
```

```
7933 032742  000716                        BR      12$             ;CONTINUE WITH THE SEARCH.
7934
7935
7936 032744  005267  152346          15$:   INC     ERRNBR          ;SET THE ERROR NUMBER TO 4402
7937 032750  012701  010216                 MOV     #EM4402,R1      ;SELECT MESSAGE TO BE REPORTED.
7938                                                                ; " NO SUITABLE ADDR FOUND. DMA TEST ABORTED "
7939
7940
7941 032754  000167  000754          16$:   JMP     34$             ;JUMP TO THE ERROR.
7942
7943
7944                                ;+
7945                                ; WRITE THE TEST DATA INTO THE TWO AREAS JUST FOUND. IF A TRAP OCCURS WHILE
7946                                ; WE ARE WRITING DATA INTO THESE AREAS THEN THE HOST MACHINE IS AT FAULT.
7947                                ;-
7948
7949 032760  012700  005130          17$:   MOV     #SDPBAS,R0      ;SET UP THE SOURCE ADDR FOR THE MOVE AS OUR
7950                                                                ;TEST DATA PATTERN.
7951 032764  016767  147232  000776        MOV     DMTSTA,DUMY     ;SAVE THE LOWER DMTSTA
7952 032772  016767  000774  147222        MOV     ODTSTA,DMTSTA   ;START WITH THE HIGHER OF THE TWO
7953                                                                ; COMPLEMENTARY ADDRESSES.
7954 033000  012703  000020                MOV     #16.,R3         ;SET THE NUMBER OF DATA BYTES TO BE WRITTEN
7955 033004  012705  000001                MOV     #1,P5           ;INDICATE TO WRITE TO DMTSTA
7956
7957
7958 033010  012701  000340                MOV     #340,R1         ;SET PRIORITY 7 TO DISABLE THE CLOCK
7959 033014  004767  172256                JSR     PC,STPSW        ;
7960
7961 033020  005267  152272                INC     ERRNBR          ;SET THE ERROR NUMBER TO 4403
7962 033024  012701  010264                MOV     #EM4403,R1      ;SELECT THE MESSAGE.
7963                                                                ; "HOST FAILURE. WRITE FAILED TO AN ADDR WHICH
7964                                                                ;HAD BEEN SUCCESSFULLY READ, TEST ABANDONED "
7965
7966 033030  004767  164626                JSR     PC,DMRW         ;PERFORM THE TRANSFER
7967 033034  005767  147214                 TST    TP4FLG          ;EXIT IF HOST FAILURE
7968 033040  001345                        BNE     16$             ;AND REPORT ERROR.
7969 033042  016767  000722  147152         MOV    DUMY,DMTSTA     ;SELECT THE LOWER DMA TEST ADDR.
7970 033050  012700  005154                 MOV    #SDP2B,R0       ;SELECT THE NEXT DATA PATTERN
7971 033054  004767  164602                JSR     PC,DMRW         ;PERFORM THE TRANSFER
7972 033060  005767  147170                 TST    TP4FLG          ;EXIT IF HOST FAILURE
7973 033064  001333                        BNE     16$             ;
7974
7975                                ;+
7976                                ; SET UP THE DHU TO PERFORM THE DMA.
7977                                ;-
7978
7979                                ;+
7980                                ; SET INTERNAL LOOPBACK, ENABLE THE RECIEVER FUNCTION ON THE LINE.
7981                                ; SET THE LPR ON THE LINE TO 38.4K BAUD, 8 BITS PER CHARACTER, ODD PARITY,
7982                                ; 2 STOP BITS. ENABLE THE TRANSMITTER ON THE LINE.
7983                                ;-
7984
7985 033066  005267  152224                 INC    ERRNBR          ;SET THE ERRNBR TO 4404
7986
7987
7988 033072  004767  164776                JSR     PC,FINACT       ;FIRST FIND AN ACTIVE LINE ON WHICH TO PERFORM
7989                                                                ; THE DMA.
```

```
7990 033076  010102                          MOV     R1,R2           ;SAVE THE LINE NUMBER ON WHICH THE DMA WILL OCCUR
7991 033100  012701  010361                  MOV     #EM4404,R1      ;SELECT THE MESSAGE,
7992                                                                 ; "NO ACTIVE LINES , TEST ABANDONED"
7993 033104  103402                          BCS     .+6             ;EXIT IF A LINE COULD NOT BE FOUND ,AFTER FIRST
7994 033106  000167  000424                  JMP     30$             ;RESTORING THE CONTENTS OF MEMORY.
7995 033112  010201                          MOV     R2,R1           ;RESTORE THE ACTIVE LINE NUMBER.
7996                                 ;+
7997                                 ; AN ACTIVE LINE HAS BEEN FOUND
7998                                 ;-
7999
8000 033114  012700  000204                  MOV     #204,R0         ;PASS THE LNCTRL CONTENTS
8001 033120  004767  174034                  JSR     PC,WTWLNC       ;INITIALISE THE LNCTRL REGISTER
8002 033124  012700  177670                  MOV     #177670,R0      ;PASS THE LPR CONTENTS
8003 033130  004767  174054                  JSR     PC,WTWLPR       ;INITIALISE THE LPR REGISTER
8004 033134  004767  172362                  JSR     PC,TXENBL       ;ENABLE TRANSMITTER ON THE LINE
8005
8006                                 ;+
8007                                 ; INITIATE THE DMA
8008                                 ;
8009
8010 033140  016 05  000626                  MOV     ODTSTA,R5       ;START FROM THE HIGHER OF THE PAIR OF ADDR.
8011 033144  012704  005130                  MOV     #SDPBAS,R4      ;SET UP THE ADDR OF THE DATA PATTERN
8012 033150  010167  000610                  MOV     R1,80$          ;SAVE THE LINE NUMBER FOR THE DMA
8013 033154  012767  000002  000600          MOV     #2,70$          ;INITIALISE THE LOOP COUNT
8014
8015 033162  012700  000052          18$:    MOV     #52,R0          ;SET UP THE LSB'S
8016 033166  005003                          CLR     R3              ;CLEAR THE REG THAT WILL HOLD THE 6 MSB'S
8017 033170  012702  000006                  MOV     #6,R2           ;CONVERT THE DMTSTA INTO
8018 033174  006305                  20$:    ASL     R5              ;A PHYSICAL ADDRESS WITH
8019 033176  006103                          ROL     R3              ;THE MSB'S IN REG #3.
8020 033200  005302                          DEC     R2              ;
8021 033202  001374                          BNE     20$             ;
8022 033204  032705  000100                  BIT     #100,R5         ;TEST BIT #6 OF THE DMTSTA
8023 033210  001402                          BEQ     22$             ;
8024 033212  012700  000025                  MOV     #25,R0          ;ALTER THE LSB'S IF BIT #6 WAS SET.
8025 033216  060005                  22$:    ADD     R0,R5           ,ADD IN THE LSB'S
8026 033220  052703  000200                  BIS     #200,R3         ;SET BIT #7.
8027
8028 033224  016777  000534  146746          MOV     80$,@CSRA       ;SELECT THE LINE ON WHICH TO PERFORM THE DMA.
8029
8030 033232  012767  010465  152056          MOV     #4405.,ERRNBR   ;SET ERROR NUMBER 4405
8031 033240  012701  010420                  MOV     #EM4405,R1      ;SELECT THE MESSAGE,
8032                                                                 ; " DMA_START BIT FOUND SET BEFORE DMA INIT.
8033                                                                 ;TEST ABANDONED"
8034 033244  105777  146744                  TSTB    @TXAD2A         ;TEST THE DUT DMA-START BIT
8035 033250  100532                          BMI     30$             ;EXIT WITH ERROR IF SET ,AFTER FIRST RESTORING
8036                                                                 ;THE CONTENTS OF MEMORY.
8037 033252  012777  000020  146736          MOV     #16.,@TXBFCA    ;SET UP CHARACTER COUNT
8038 033260  010577  146726                  MOV     R5,@TXAD1A      ;SET UP BITS 0 TO 15 OF THE PHYISCAL ADDR.
8039 033264  110377  146724                  MOVB    R3,@TXAD2A      ;SET UP BITS 16 TP 21 , AND INITIATE THE DMA.
8040
8041                                 ;+
8042                                 ; WAIT FOR THE DMA TO COMPLETE AND THE LAST CHARACTER TO BE RECIEVED
8043                                 ;-
8044
8045 033270  012701  170144                  MOV     #170144,R1      ;TEST BIT 15, TIME-OUT OF 100 MS.
8046 033274  016702  146700                  MOV     CSRA,R2         ;PASS THE ADDR OF THE REG TO TEST.
```

```
8047
8048 033300  005267  152012                    INC     ERRNBR          ;SET ERROR NUMBER TO 4406
8049
8050 033304  004767  173574                    JSR     PC,WAIBIS       ;WAIT FOR BIT TO SET
8051 033310  012701  010514                    MOV     #EM4406,R1      ;SELECT THE MESSAGE ,
8052                                                                    ;  " TIME-OUT OCCURED WAITING FOR DMA TO
8053                                                                    ;COMPLETE. TEST ABANDONED"
8054 033314  103110                            BCC     30$             ;EXIT IF TIME-OUT OCCCURED. AFTER FIRST ,
8055                                                                    ;RESTORING THE CONTENTS OF MEMORY.
8056 033316  010402                            MOV     R4,R2           ;SAVE R4
8057 033320  012704  000005                    MOV     #5,R4           ;SET 5 MS DELAY
8058 033324  004767  164220                    JSR     PC,DELAY        ;DELAY TO ALLOW LAST CHARACTER TO BE RECIEVED
8059 033330  010204                            MOV     R2,R4           ;RESTORE R4
8060
8061                                    ;+
8062                                    ; READ THE CONTENTS OF THE RXFIFO AND COMPARE THEM WITH THE CORRECT DATA
8063                                    ;-
8064
8065 033332  005003                            CLR     R3              ;CLEAR THE READ DATA COUNTER
8066 033334  012705  000200                    MOV     #128.,R5        ;SET THE MAX BMP CODE READ COUNT
8067
8068 033340  012767  010467  151750  24$:      MOV     #4407.,ERRNBR   ;SET THE ERRNBR TO 4407
8069 033346  012701  010570                    MOV     #EM4407,R1      ;SELECT THE MESSAGE ,
8070                                                                    ;  " RXFIFO EMPTY TOO SOON, DMA FAILED
8071                                                                    ;TEST ABANDONED"
8072
8073 033352  017702  146624                    MOV     @RBUFA,R2       ;READ THE CHARACTER FROM THE FIFO
8074 033356  100067                            BPL     30$             ;BRANCH TO REPORT ERROR IF FIFO EMPTY TOO SOON,
8075                                                                    ;AFTER FIRST RESTORING THE CONTENTS OF MEMORY.
8076 033360  012700  170301                    MOV     #170301,R0      ;SET UP BIT MASK OF A BMP CODE
8077 033364  040200                            BIC     R2,R0           ;TRY TO CLEAR THE BMP CODE MASK
8078 033366  001011                            BNE     28$             ;BRANCH IF NOT A BMP CODE
8079 033370  004767  171256                    JSR     PC,SAVBMP       ;SAVE THE BMP CODE ON THE QUEUE
8080
8081 033374  005267  151716                    INC     ERRNBR          ;SET THE ERRNBR TO 4408
8082 033400  012701  010652                    MOV     #EM4408,R1      ;SELECT THE MESSAGE.
8083                                                                    ;  " TOO MANY BMP CODES FOUND IN THE RXFIFO.
8084                                                                    ;TEST ABANDONED'
8085
8086 033404  005305                            DEC     R5              ;DEC THE MAX BMP CODE READ COUNT
8087 033406  001453                            BEQ     30$             ;GO REPORT ERROR IF TOO MANY BMP CODES FOUND .
8088                                                                    ;AFTER FIRST RESTORING THE CONTENTS OF MEMORY.
8089 033410  000753                            BR      24$             ;DON'T COUNT THE BMP CODE AS A VALID CHARACTER
8090
8091
8092 033412  012767  010471  151676  28$:      MOV     #4409.,ERRNBR   ;SET THE ERRNBR TO 4409
8093 033420  010201                            MOV     R2,R1           ;SAVE THE CHARACTER FROM THE FIFO
8094 033422  012702  010715                    MOV     #EM4409,R2      ;SELECT THE MESSAGE ,
8095                                                                    ;  " BAD BIT BETWEEN BITS 0 AND "
8096 033426  012767  015536  151666            MOV     #ER9101,ERRBLK  ;SELECT THE ERROR ROUTINE.
8097 033434  012767  177777  000332            MOV     #-1,SUCSS       ;INDICATE 'BAD BITS' FAILURE
8098
8099 033442  122401                            CMPB    (R4)+,R1        ;COMPARE CHAR FROM FIFO WITH THE CORRECT DATA.
8100 033444  001034                            BNE     30$             ;BRANCH IF INCORRECT AND RESTORE MEM CONT'S.
8101 033446  005067  000322                    CLR     SUCSS           ;INDICATE NON TEST SPECIFIC FAILURE  E.G. TIME-OUTS
8102 033452  005203                            INC     R3              ;COUNT THIS CHARACTER.
8103 033454  022703  000020                    CMP     #16.,R3         ;HAVE WE RECIEVED ALL THE CHARACTERS ?
```

```
8104 033460  001327                          BNE     24$             ;LOOP UNTIL ALL CHARACTERS (NON-BMP) ARE READ.
8105 033462  005367  000274                   DEC     70$             ;DECREMENT THE LOOP COUNT
8106 033466  001420                          BEQ     29$             ;BRANCH IF BOTH DMA'S ARE COMPLETED
8107 033470  012704  005154                   MOV     #SDP2B,R4       ;SET UP THE SECOND DATA PATTERN
8108 033474  016705  146522                   MOV     DMTSTA,R5       ;SET UP THE OTHER DMA TEST ADDRESS
8109
8110 033500  012767  010472  151610           MOV     #4410.,ERRNBR   ;SET ERRNBR TO 4410
8111 033506  012701  010752                   MOV     #EM4410,R1      ;SELECT THE MESSAGE ,
8112                                                                  ; " RXFIFO FAILED TO PURGE. TEST ABANDONED "
8113 033512  012767  014124  151602           MOV     #ERC503,ERRBLK  ;SELECT THE ERROR ROUTINE
8114
8115 033520  004767  167216                   JSR     PC,PUFIFO       ;PURGE THE RXFIFO
8116 033524  103004                          BCC     30$             ;EXIT WITH ERROR IF FIFO WOULD NOT PURGE ,
8117                                                                  ;AFTER FIRST RESTORING THE CONTENTS OF MEMORY.
8118 033526  000615                          BR      18$             ;OTHERWISE REPEAT.
8119
8120 033530  012767  000001  000236  29$:     MOV     #1,SUCSS        ;INDICATE THAT WE HAVE BEEN ABLE TO TEST,
8121                                                                  ;SOME OF THE BITS.
8122
8123
8124                                 ;+
8125                                 ; RESTORE THE ORIGINAL DATA IN THE MEMORY
8126                                 ;-
8127
8128 033536  016767  146460  000224  30$:     MOV     DMTSTA,DUMY     ;START WITH THE HIGHER OF THE PAIR OF DMTSTA
8129 033544  016767  000222  146450           MOV     ODTSTA,DMTSTA   ;
8130 033552  012700  003602                   MOV     #BUFBAS,R0      ;POINT AT THE START OF THE SAVED DATA AREA
8131 033556  012705  000001                   MOV     #1,R5           ;SELECT WRITE TO (DMTSTA)
8132 033562  012703  000020                   MOV     #16.,R3         ;PASS NUMBER OF BYTES TO BE WRITTEN
8133 033566  004767  164070                   JSR     PC,DMRW         ;RESTORE THE DATA
8134 033572  005767  146456                   TST     TP4FLG          ;GO REPORT ERROR IF A TRAP OCCURED
8135 033576  001012                          BNE     31$             ;
8136 033600  016767  000164  146414           MOV     DUMY,DMTSTA     ;NOW RESTORE THE DATA FROM THE LOWER
8137                                                                  ;OF THE PAIR OF TEST ADDRESSES.
8138 033606  012700  004202                   MOV     #BUFMID,R0      ;POINT AT THE START OF THE SAVED DATA AREA
8139 033612  004767  164044                   JSR     PC,DMRW         ;RESTORE THE DATA
8140
8141 033616  005767  146432                   TST     TP4FLG          ;GO REPORT ANY ERRORS IF A NO TRAP
8142 033622  001411                          BEQ     32$             ; OCCURED DURING THE RESTORE.
8143
8144 033624  012767  010473  151464  31$:     MOV     #4411.,ERRNBR   ;SET THE ERROR NUMBER TO 4411
8145 033632  012701  011001                   MOV     #EM4411,R1      ;SELECT THE MESSAGE ,
8146                                                                  ; " HOST FAILURE. WRITE FAILURE TO AN ADDR
8147                                                                  ;WHICH HAD PREVIOUSLY BEEN SUCCESSFULLY
8148                                                                  ;WRITTEN TO. "
8149 033636  012767  014124  151456           MOV     #ERC503,ERRBLK  ;SELECT THE ERROR ROUTINE
8150 033644  000433                          BR      34$             ;REPORT THE ERROR
8151
8152                                 ;+
8153                                 ; HAS THE TEST BEEN SUCCESSFUL, PRINT THE BITS TESTED IF IT HAS,
8154                                 ; REPORT THE ERRORS OTHERWISE.
8155                                 ;-
8156
8157
8158 033646  005767  000122          32$:     TST     SUCSS           ;IF THE ERROR IS NON TEST SPECIFIC THEN
8159 033652  001430                          BEQ     34$             ;BRANCH TO REPORT ERRORS
8160 033654  016701  000106                   MOV     BITSTD,R1       ;LOAD THE NUMBER OF BITS TESTED
```

```
8161 033660  005301                        DEC     R1              ;DEC TO GIVE THE BIT POSITION OF THE MSB TESTED.
8162 033662  022767  000001  000104        CMP     #1,SUCSS        ;IF THE BITS TESTED ARE BAD THEN
8163 033670  001021                      BNE     34$             ;BRANCH AND REPORT ERRORS.
8164
8165                                    ;+
8166                                    ; OTHERWISE DETERMINE IF PRINTING OF THE SUCCESSFULLY TESTED BITS WAS REQUESTED.
8167                                    ;
8168
8169
8170 033672  032767  000040  146262        BIT     #BIT05,OPTION   ; PRINT THE BITS TESTED IF THE SOFTWARE
8171 033700  001416                      BEQ     60$             ;OPTION HAS REQUESTED IT
8172 033702  010102                       MOV     R1,R2           ;CALCULATE THE NUMBER OF BITS WHICH HAVE
8173 033704  005202                       INC     R2              ;  BEEN TESTED SUCCESSFULLY.
8174 033706                              PRINTB  #EF4401,R1,R2   ;PRINT THE NUMBER OF BITS TESTED MESSSAGE.
     033706  010246                                                                  MOV     R2,-(SP)
     033710  010146                                                                  MOV     R1,-(SP)
     033712  012746  005554                                                         MOV     #EF4401,-(SP)
     033716  012746  000003                                                         MOV     #3,-(SP)
     033722  010600                                                                 MOV     SP,R0
     033724  104414                                                                 TRAP    C$PNTB
     033726  062706  000010                                                         ADD     #10,SP
8175 033732  000401                      BR      60$             ;EXIT THE TEST
8176
8177
8178 033734                        34$:  ERROR                   ;  REPORT ERRORS
     033734  104460                                                                TRAP    C$ERROR
8179
8180
8181 033736                        60$:  SETPRI  #PRI05           ;ENABLE THE CLOCK
     033736  012700  000240                                                         MOV     #PRI05,R0
     033742  104441                                                                 TRAP    C$SPRI
8182 033744  016767  146306  144032        MOV     TP4VEC,4        ;RESTORE THE NORMAL 004 TRAP VECTOR
8183 033752  005067  146242                CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST
8184
8185 033756                              EXIT    TST
     033756  104432                                                                TRAP    C$EXIT
     033760  000016                                                                .WORD   L10030-.
8186
8187
8188                                    ;+
8189                                    ; ********************** LOCAL VARIBLE AREA *********************************
8190                                    ;-
8191
8192 033762  000000                    70$:    .WORD   0               ;COUNTER FOR THE NUMBER OF DMA'S COMPLETED
8193 033764  000000                    80$:    .WORD   0               ;SAVE AREA FOR THE ACTIVE LINE NUMBER
8194 033766  000000                    BITSTD: .WORD   0               ;NUMBER OF BITS TESTED
8195 033770  000000                    DUMY:   .WORD   0               ;DUMMY VARIABLE
8196 033772  000000                    ODTSTA: .WORD   0               ;HIGHER OF THE PAIR OF `OMPLEMENTARY ADDR.
8197 033774  000000                    SUCSS:  .WORD   0               ;SUCCESS INDICATOR, -1   ERROR DUE TO BAD BITS
8198                                                                    ;                    1 - SUCCESSFUL TEST
8199                                                                    ;                    0   OTHER ERRORS
8200
8201                                    ;+
8202                                    ;****************************** END ***********************************
8203                                    ;-
8204
8205
```

```
8206
8207 033776                      ENDTST
     033776                                                    L10030:
     033776  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 DHU11.
                                ;*      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.
                                ;*
                                ;--**********************************************************************
034000                                  BGNTST
034000                                                                          T5::
                                ;+
                                ; EXECUTE THIS TEST IN  STAGGARED LOOPBACK MODE ONLY.
                                ;-
034000  126727  146170  000002          CMPB    LOPBCK,#2       ;CHECK MODE SELECTED.
034006  001402                          BEQ     .+6             ;AVOID EXITING THE TEST IF STAGGERED LOOPBACK
                                                                ;MODE IS SELECTED.
034010  000167  000372                  JMP     60$             ; EXIT THE TEST.
034014                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
034014  012700  000240                                                          MOV     #PRI05,R0
034020  104441                                                                  TRAP    C$SPRI
034022  012767  177777  146170          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
000005                                  TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
034030  012767  000005  146222          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (62)
034036  012767  000001  151250          MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
034044  012767  014071  151244          MOV     #6201.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
034052  012767  011117  151240          MOV     #EM6201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
034060  005067  146414                  CLR     ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
034064  005067  146134                  CLR     FERROR          ;CLEAR THE "AT LEAST ONE ERROR" INDICATOR.
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND  RX INTERRUPT ENABLE BITS.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 6201 <<<<<.
                                ;-
034070  004767  163334                  JSR     PC,CLNRST       ;RESET THE DUT.
034074  103144                          BCC     60$             ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; DISABLE ALL INTERRUPTS.
                                ; SET UP  DMA TX AND RX INTERRUPT SERVICE ROUTINES.
                                ;-
034076                                  SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
034076  012700  000340                                                          MOV     #PRI07,R0
034102  104441                                                                  TRAP    C$SPRI
034104                                  SETVEC  TXVECA,#TXDMA,#PRI06    ;SELECT DMA TX INT SERVICE RTN.
034104  012746  000300                                                          MOV     #PRI06,-(SP)
034110  012746  027520                                                          MOV     #TXDMA,-(SP)
034114  016746  146050                                                          MOV     TXVECA,-(SP)
034120  012746  000003                                                          MOV     #3,-(SP)
034124  104437                                                                  TRAP    C$SVEC
034126  062706  000010                                                          ADD     #10,SP
034132                                  SETPRI  #PRI04          ;ALLOW INTERRUPTS.
```

```
        034132  012700  000200                                                  MOV     #PRIO4,R0
        034136  104441                                                          TRAP    C$SPRI
   8255                                 ;+
   8256                                 ; CLEAR TX, RX, AND DMA_START ERROR FLAGS.
   8257                                 ;-
   8258 034140  005067  146336                  CLR     TXDONF          ;CLEAR TX DONE FLAGS FOR ALL LINES.
   8259 034144  005067  146334                  CLR     RXDONF          ;CLEAR RX DONE FLAGS FOR ALL LINES.
   8260 034150  005067  146110                  CLR     TXINTF          ;CLEAR TX ERROR FLAGS FOR ALL LINES.
   8261                                 ;+
   8262                                 ; SET UP ERROR TABLE AND DATA PATTERN TABLE.
   8263                                 ;  THE NUMERICAL VALUE OF THE CHARACTER INDICATES THE NUMBER OF THE LINE
   8264                                 ;  THAT TRANSMITTED IT.
   8265                                 ;-
   8266 034154  012700  003302                  MOV     #ERCNTB,R0      ;PASS THE ADDRESS OF THE TABLE TO BE CLEARED.
   8267 034160  004767  163266                 JSR     PC,CLR16W       ;CLEAR THE RX ERROR COUNTERS TABLE.
   8268 034164  005067  147412                  CLR     BUFBAS          ;SET SINGLE CHAR DATA TO BE A NULL.
   8269                                 ;+
   8270                                 ; INITIALISE DMA PARAMETERS IN THE CONTROL BLOCK.
   8271                                 ;  TRANSMISSION ON LINE GROUP 1 AT 8 BITS/CHAR,1 STOP BITS,ODD PARITY.
   8272                                 ;  RECEPTION ON LINE GROUP 2 AT 5 BITS/CHAR,1 STOP,ODD PARITY.
   8273                                 ;-
   8274 034170  012700  156470                  MOV     #156470,R0      ;PASS LPR PARAMETER FOR 8 BITS/CHAR.
   8275 034174  012701  156440                  MOV     #156440,R1      ;PASS LPR PARAMETER FOR 5 BITS/CHAR.
   8276 034200  004767  164424                 JSR     PC,GETTIM       ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
   8277 034204  012702  003602                  MOV     #BUFBAS,R2      ;PASS START ADDRESS OF DATA PATTERN.
   8278 034210  012703  000001                  MOV     #1,R3           ;PASS LENGTH OF DATA PATTERN.
   8279 034214  016704  146016                  MOV     LGRP1M,R4       ;PASS LINE GROUP OF LINES THAT ARE TO TX.
   8280 034220  004767  163730                 JSR     PC,FRPSUP       ;SET UP DUT FOR TRANSMISSION AND RECEPTION.
   8281
   8282                                 ;+
   8283                                 ; PURGE THE FIFO OF ANY UN-WANTED CHARACTERS. THIS ROUTINE REPORTS ERRORS
   8284                                 ;  WITH WITH ERROR NUMBERS FROM >>>>> 6202 THRU 6204 <<<<<.
   8285                                 ; PERFORM TRANSMISSION AND RECEPTION AT 9600 BAUD.
   8286                                 ;  REPORT ANY ERRORS FOUND, IE. FRAMING ERROR BIT CLEAR OR PARITY ERROR SET.
   8287                                 ;-
   8288 034224  005267  151066                   INC     ERRNBR          ;SET THE ERROR REPORT NUMBER TO 6202.
   8289 034230  004767  166570                  JSR     PC,PUFIFR       ;CLEAN OUT THE FIFO.
   8290 034234  103064                          BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
   8291 034236  012767  014075  151052           MOV     #6205.,ERRNBR   ;SET THE ERROR NUMBER TO 6205.
   8292 034244  004767  171346                  JSR     PC,TXFRPR       ;TX DATA PATTERN ON SELECTED ACTIVE LINES.
   8293 034250  012705  100000                   MOV     #100000,R5      ;PASS FRAMING ERROR TEST FLAG.
   8294                                 ;+
   8295                                 ; THIS SUBROUTINE REPORTS ERROR NUMBER >>>>> 6205 <<<<<.
   8296                                 ;-
   8297 034254  004767  162550                  JSR     PC,CKFRPR       ;READ CHARACTERS, REPORT ANY ERRORS FOUND.
   8298
   8299 034260  005767  145740                   TST     FERROR          ;HAS AN ERROR BEEN DETECTED?
   8300 034264  001404                          BEQ     2$              ;BRANCH IF NO ERROR
   8301 034266  032767  000100  145666           BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN ENBL'D?
   8302 034274  001436                          BEQ     54$             ;BRANCH IF IT HASN'T AND EXIT THE TEST. THE
   8303                                                                 ;TEST FAILURE MESSAGE HAS BEEN PRINTED.
   8304                                 ;+
   8305                                 ; REVERSE TRANSMISSION/RECEPTION ROLES ON ALL ACTIVE LINES, AND REPEAT TEST.
   8306                                 ;-
   8307 034276  005104                  2$:     COM     R4              ;REVERSE ROLES FOR TRANSMISSION AND RECEPTION.
   8308 034300  004767  163650                  JSR     PC,FRPSUP       ;SET UP DUT FOR TRANSMISSION AND RECEPTION.
   8309 034304  005267  151006                   INC     ERRNBR          ;SET ERROR NUMBER TO 6206.
```

```
8310                                    ;*
8311                                    ; THIS ROUTNE REPORTS ERRORS WITH NUMBERS >>>>> 6206 THRU 6208 <<<<<.
8312                                    ;-
8313 034310  004767  166510                    JSR     PC,PUFIFR       ;CLEAN OUT THE FIFO.
8314 034314  103034                            BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
8315 034316  012767  014101  150772            MOV     #6209.,ERRNBR   ;SET ERROR NUMBER TO 6209.
8316 034324  004767  171266                    JSR     PC,TXFRPR       ;TX DATA PATTERN ON SELECTED ACTIVE LINES.
8317 034330  012705  100000                    MOV     #100000,R5      ;PASS FRAMING ERROR TEST FLAG.
8318                                    ;*
8319                                    ; THIS SUBROUTINE REPORTS ERRORS >>>>> 6209 <<<<<.
8320                                    ;-
8321 034334  004767  162470                    JSR     PC,CKFRPR       ;READ CHARACTERS, REPORT ANY ERRORS FOUND.
8322
8323 034340  005767  145660                    TST     FERROR          ;HAS AN ERROR BEEN DETECTED?
8324 034344  001404                            BEQ     4$              ;BRANCH IF NO ERROR
8325 034346  032767  000100  145606            BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN ENBL'D?
8326 034354  001406                            BEQ     54$             ;BRANCH IF IT HASN'T AND EXIT THE TEST. THE
8327                                                                    ;TEST FAILURE MESSAGE HAS BEEN PRINTED.
8328
8329 034356  005267  150734            4$:     INC     ERRNBR          ;SET ERROR NUMBER TO 6210.
8330                                    ;*
8331                                    ;    DISABLE INTERRUPTS.
8332                                    ;    CLEAR THE INTERRUPT VECTORS.
8333                                    ;    UPDATE THE ACTIVE LINES BIT MAP TO REFLECT LINES REMOVED FROM TESTING.
8334                                    ;     THIS SUBROUTINE REPORTS ERRORS >>>>> 6210 THRU 6212 <<<<<.
8335                                    ;-
8336 034362  004767  171324                    JSR     PC,TXIEO        ;DISABLE ALL TX INTERRUPTS.
8337 034366  004767  171746                    JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM TX/RX.
8338
8339 034372                            54$:    SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
     034372  012700  000340                                                    MOV     #PRIO7,R0
     034376  104441                                                            TRAP    C$SPRI
8340 034400                                    CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
     034400  016700  145564                                                    MOV     TXVECA,R0
     034404  104436                                                            TRAP    C$CVEC
8341
8342 034406  005067  145606            60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
8343
8344 034412                                    ENDTST
     034412                                                         L10031:
     034412  104401                                                            TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - PARERR
                                ;++*************************************************************************
                                ;*                          - PARITY ERROR GENERATION TEST -
                                ;*
                                ;*      THIS TEST IS USED TO VERIFY THE PARITY ERROR DETECTION AND REPORT
                                ;*      CAPABILITIES OF THE DUT.
                                ;*      WHEN STAGGARED LOOPBACK MODE IS SELECTED, DATA IS TRANSMITTED
                                ;*      ON ALL ACTIVE LINES IN LINE GROUP 1 WITH ODD PARITY SELECTED,
                                ;*      AND RECEIVED ON LINES IN GROUP 2 WITH EVEN PARITY SELECTED.
                                ;*      THIS WILL GENERATE A PARITY ERROR FOR EACH CHARACTER RECEIVED.
                                ;*      THE PARITY SELECTION IS THEN REVERSED ON THE LINES IN EACH GROUP
                                ;*      AND THE TEST IS REPEATED.
                                ;*      THIS TEST WILL ONLY EXECUTE IF THE STAGGARED LOOPBACK MODE IS SELECTED.
                                ;*      THE SPECIAL STAGGARED LOOPBACK BERG CONNECTOR MUST BE FITTED.
                                ;*
                                ;--*************************************************************************
034414                                  BGNTST
034414                                                                          T6::
                                ;+
                                ; EXECUTE THIS TEST IN  STAGGARED LOOPBACK MODE ONLY.
                                ;-
034414  126727  145554  000002          CMPB    LOPBCK,#2       ;CHECK MODE SELECTED.
034422  001402                          BEQ     .+6             ;AVOID EXITING THE TEST IF STAGGERED LOOPBACK
                                                                ;MODE IS SELECTED.
034424  000167  000434                  JMP     60$             ; EXIT THE TEST.

034430                                  SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
034430  012700  000240                                                  MOV     #PRI05,R0
034434  104441                                                          TRAP    C$SPRI
034436  012767  177777  145554          MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
        000006                          TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
034444  012767  000006  145606          MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (63)
034452  012767  000001  150634          MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
034460  012767  014235  150630          MOV     #6301.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
034466  012767  011160  150624          MOV     #EM6301,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
034474  005067  146000                  CLR     ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
034500  005067  145520                  CLR     FERROR          ;CLEAR THE "AT LEAST ONE ERROR" INDICATOR
                                ;+
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND  RX INTERRUPT ENABLE BITS.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 6301 <<<<<.
                                ;-
034504  004767  162720                  JSR     PC,CLNRST       ;RESET THE DUT.
034510  103165                          BCC     60$             ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
                                ;+
                                ; DISABLE ALL INTERRUPTS.
                                ; SET UP  DMA TX AND RX INTERRUPT SERVICE ROUTINES.
                                ;-
034512                                  SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
034512  012700  000340                                                  MOV     #PRI07,R0
034516  104441                                                          TRAP    C$SPRI
034520                                  SETVEC  TXVECA,#TXDMA,#PRI06    ;SELECT DMA TX INT SERVICE RTN.
034520  012746  000300                                                  MOV     #PRI06,-(SP)
034524  012746  027520                                                  MOV     #TXDMA,-(SP)
034530  016746  145434                                                  MOV     TXVECA,-(SP)
034534  012746  000003                                                  MOV     #3,-(SP)
034540  104437                                                          TRAP    C$SVEC
```

```
     034542 062706 000010                                                                     ADD     #10,SP
     034546                          SETPRI  #PRIO4          ;ALLOW INTERRUPTS.
     034546 012700 000200                                                                     MOV     #PRIO4,R0
     034552 104441                                                                            TRAP    C$SPRI
                              ;+
                              ; CLEAR TX/RX  FLAGS.
                              ;
     034554 005067 145722            CLR     TXDONF          ;CLEAR TX DONE FLAGS FOR ALL LINES.
     034560 005067 145720            CLR     RXDONF          ;CLEAR RX DONE FLAGS FOR ALL LINES.
     034564 005067 145474            CLR     TXINTF          ;CLEAR TX ERROR FLAGS FOR ALL LINES.
                              ;+
                              ; SET UP ERROR COUNTER TABLE.
                              ;
     034570 012700 003302            MOV     #ERCNTB,R0      ;PASS THE ADDRESS OF THE TABLE TO BE CLEARED.
     034574 004767 162652            JSR     PC,CLR16W       ;CLEAR THE RX ERROR COUNTERS TABLE.

                              ;+
                              ; INITIALISE DMA PARAMETERS IN THE CONTROL BLOCK.
                              ;-
     034600 012700 156470            MOV     #156470,R0      ;PASS LPR PARAMETER WITH ODD PARITY.
     034604 012701 156570            MOV     #156570,R1      ;PASS LPR PARAMETER WITH EVEN PARITY.
     034610 004767 164014            JSR     PC,GETTIM       ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
     034614 012702 005154            MOV     #SDP2B,R2       ;PASS START ADDRESS OF DATA PATTERN.
     034620 012703 000020            MOV     #16.,R3         ;PASS LENGTH OF DATA PATTERN.
     034624 016704 145406            MOV     LGRP1M,R4       ;PASS BIT MAP OF LINES TO BE SET WITH ODD PAR.
     034630 004767 163320            JSR     PC,FRPSUP       ;SET UP DUT FOR TRANSMISSION AND RECEPTION.

                              ;+
                              ; PURGE THE FIFO OF ANY UN-WANTED CHARACTERS.
                              ; PERFORM TRANSMISSION AND RECEPTION OF THE 16 BYTE DATA PATTERN AT 9600 BAUD.
                              ;  TRANSMISSION ON LINE IN  GROUP 1, 8 BITS/CHAR, 1 STOP BITS, ODD PARITY.
                              ;  RECEPTION ON LINES IN  GROUP 2 AT 8 BITS/CHAR, 1 STOP, EVEN PARITY.
                              ;  REMOVE CHARACTERS FROM THE FIFO AND LOOK FOR THE PARITY ERROR BIT BEING SET.
                              ;  REPORT ANY ERRORS FOUND, IE. FRAMMING ERROR BIT SET OR PARITY ERROR CLEAR.
                              ;-
                                ;+
                                ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 6302 THRU 6304 <<<<<.
                                ;-
     034634 004767 166164            JSR     PC,PUFIFR       ;CLEAN OUT THE FIFO.
     034640 103111                   BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
     034642 012767 014241 150446     MOV     #6305.,ERRNBR   ;SET ERROR NUMBER TO 6305
     034650 004767 164032            JSR     PC,INIDMA       ;TX DATA PATTERN ON ALL ACTIVE LINES.
     034654 005005                   CLR     R5              ;PASS PARITY ERROR TEST FLAG.
                              ;+
                              ;THIS SUBROUTINE REPORTS ERROR NUMBER  >>>>> 6305 <<<<<.
                              ;-
     034656 004767 162146            JSR     PC,CKFRPR       ;READ CHARACTERS, REPORT ANY ERRORS FOUND.

     034662 005767 145336            TST     FERROR          ;HAS AN ERROR BEEN FOUND ?
     034666 001404                   BEQ     2$              ;BRANCH TO CONTINUE IF IT HASN'T.
     034670 032767 000100 145264     BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
     034676 001457                   BEQ     54$             ;EXIT THE TEST IF IT HASN'T. THE TEST FAILURE
                                                             ;MESSAGE HAS ALREADY BEEN REPORTED.

     034700 005267 150412    2$:     INC     ERRNBR          ;SET ERROR NUMBER TO 6306.
                              ;+
                              ;  THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 6306 THRU 6309 <<<<<
```

```
8447                                        ;-
8448 034704  004767  171430                        JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM TX/RX.
8449 034710  005767  145310                         TST     FERROR          ;HAS AN ERROR BEEN FOUND ?
8450 034714  001404                               BEQ     4$              ;BRANCH TO CONTINUE IF IT HASN'T.
8451 034716  032767  000100  145236                BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
8452 034724  001457                               BEQ     60$             ;EXIT THE TEST IF IT HASN'T. THE TEST FAILURE
8453                                                                       ;MESSAGE HAS ALREADY BEEN REPORTED.
8454
8455 034726  012767  014246  150362  4$:          MOV     #6310.,ERRNBR   ;SET ERROR NUMBER TO 6310.
8456 034734  005067  145542                         CLR     TXDONF          ;CLEAR TX DONE FLAGS FOR ALL LINES.
8457 034740  005067  145540                         CLR     RXDONF          ;CLEAR RX DONE FLAGS FOR ALL LINES.
8458 034744  005067  145314                         CLR     TXINTF          ;CLEAR TX DMA HANDOVER ERROR FLAGS.
8459                                        ;+
8460                                        ; REVERSE TRANSMISSION/RECEPTION ROLES ON ALL ACTIVE LINES, AND REPEAT TEST.
8461                                        ;-
8462 034750  00510-                                COM     R4              ;REVERSE ROLES FOR TRANSMISSION AND RECEPTION.
8463 034752  004767  163176                        JSR     PC,FRPSUP       ;SET UP DUT FOR TRANSMISSION AND RECEPTION.
8464                                        ;+
8465                                        ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 6310 THRU 6311 <<<<<.
8466                                        ;-
8467 034756  004767  166042                        JSR     PC,PUFIFR       ;CLEAN OUT THE FIFO.
8468 034762  103040                               BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
8469 034764  012767  014250  150324                MOV     #6312.,ERRNBR   ;SET ERROR NUMBER TO 6312.
8470 034772  004767  163710                        JSR     PC,INIDMA       ;TX DATA PATTERN ON SELECTED ACTIVE LINES.
8471                                        ;+
8472                                        ; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 6312 THRU 6316 <<<<<.
8473                                        ;-
8474 034776  004767  162026                        JSR     PC,CKFRPR       ;READ CHARACTERS, REPORT ANY ERRORS FOUND.
8475 035002  005767  145216                         TST     FERROR          ;HAS AN ERROR BEEN FOUND ?
8476 035006  001404                               BEQ     6$              ;BRANCH TO CONTINUE IF IT HASN'T.
8477 035010  032767  000100  145144                BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
8478 035016  001407                               BEQ     54$             ;EXIT THE TEST IF IT HASN'T. THE TEST FAILURE
8479                                                                       ;MESSAGE HAS ALREADY BEEN REPORTED.
8480
8481 035020  012767  014255  150270  6$:          MOV     #6317.,ERRNBR   ;SET ERROR NUMBER TO 6317.
8482                                        ;+
8483                                        ;    DISABLE INTERRUPTS.
8484                                        ;    CLEAR THE INTERRUPT VECTORS.
8485                                        ;    UPDATE THE ACTIVE LINES BIT MAP TO REFLECT LINES REMOVED FROM TESTING.
8486                                        ;-
8487 035026  004767  170660                        JSR     PC,TXIEO        ;DISABLE ALL TX INTERRUPTS.
8488                                        ;+
8489                                        ;     THIS SUBROUTINE REPORTS ERRORS >>>>> 6317 THRU 6320 <<<<<.
8490                                        ;-
8491 035032  004767  171302                        JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM TX/RX.
8492
8493 035036                                 54$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     035036  012700  000340                                                                MOV     #PRI07,R0
     035042  104441                                                                      TRAP    C$SPRI
8494 035044                                        CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
     035044  016700  145120                                                                MOV     TXVECA,R0
     035050  104436                                                                      TRAP    C$CVEC
8495
8496                                        ;+
8497                                        ; THIS SUBROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 6321 <<<<<.
8498                                        ;
8499 035052  012767  014261  150236                MOV     #6321.,ERRNBR   ;SET ERROR NUMBER TO 6321.
```

```
 8500 035060  004767  167036                JSR     PC,REPSMR       ;REPORT ERROR SUMMARIES IF CALLED FOR.
 8501
 8502 035064  005067  145130        60$:     CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
 8503 035070                                 ENDTST
      035070                                                                     L10032:
      035070  104401                                                                 TRAP    C$ETST
```

```
                                .SBTTL  HARDWARE TEST           - DMA -
                                ;++ ************************************************************************
                                ;*                      - DMA MODE TEST -
                                ;*      THIS TEST VERIFIES THAT THE DEVICE UNDER TEST (DUT) WILL PERFORM
                                ;*      TRANSMISSION AND RECEPTION CORRECTLY USING THE DMA MODE TRANSMISSION.
                                ;*      THE TEST IS PERFORMED AT ALL BAUDRATES (EXCEPT 50 BAUD), 8 BITS PER
                                ;*      CHARACTER, 1 STOP BIT, AND WITH PARITY CHECKING (BOTH ODD AND EVEN).
                                ;*      A HIGH SPEED TEST IS ALSO PERFORMED AT THE HIGHEST 3 BAUDRATES AT
                                ;*      BOTH 5 AND 8 BITS PER CHARACTER, 1 STOP BIT, AND NO PARITY CHECKING.
                                ;*      THIS TEST IS PERFORMED WITH THE TYPE OF LOOPBACK WHICH WAS SPECIFIED
                                ;*      IN THE DUT HARDWARE P TABLE ON ALL ACTIVE LINES.
                                ;*
                                ;   ************************************************************************
035072                                  BGNTST
035072                                                                          T7::
035072                                  SETPRI  #PRIO5          ;ALLOW LTC INTERRUPTS.
035072  012700  000240                                                        MOV     #PRIO5,R0
035076  104441                                                                TRAP    C$SPRI
        000007                           TNUM == TNUM + 1       ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
035100  012767  000007  145152          MOV     #TNUM,TSTNUM   ;SET UP THE TEST NUMBER.        (91)
035106  012767  177777  145104          MOV     #-1,CTRLCF     ;INDICATE THAT WE ARE IN A TEST.
035114  012767  000001  150172          MOV     #1,ERRTYP      ;SET ERROR TYPE AS FATAL IN ERROR TABLE.
035122  012767  021615  150166          MOV     #9101.,ERRNBR  ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
035130  012767  012400  150162          MOV     #EM9101,ERRMSG ;SET ERROR MESSAGE ADDRESS IN ERRTBL.
035136  005067  145336                  CLR     ERSMRF         ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
035142  005067  145056                  CLR     FERROR         ;CLEAR THE "AT LEAST ONE ERROR' INDICATOR.
                                ;*
                                ; RESET THE DUT TO A KNOWN STATE, REMOVE THE STATUS CODES FROM THE FIFO.
                                ;  CLEAR TX AND RX INTERRUPT ENABLE BITS IN THE CSR.
                                ; THIS SUBROUTINE REPORTS ERROR >>>>> 9101 <<<<<.
                                ;-
035146  004767  162256                  JSR     PC,CLNRST      ;RESET THE DHU-11, REPORT ANY ERRORS FOUND.
035152  103402                          BCS     2$             ;SKIP AROUND TEST EXIT IF NO FATAL ERROR FOUND.
035154  000167  000664                  JMP     60$            ;RESET FAILURE, ABORT THIS TEST.
                                ;*
                                ; SET UP FOR TRANSMIT INTERRUPTS.
                                ;-
035160                          2$:     SETPRI  #PRIO7         ;DISABLE ALL INTERRUPTS.
035160  012700  000340                                                        MOV     #PRIO7,R0
035164  104441                                                                TRAP    C$SPRI
035166                                  SETVEC  TXVECA,#TXDMA,#PRIO6    ;SELECT DMA TX INT SERVICE RTN.
035166  012746  000300                                                        MOV     #PRIO6,-(SP)
035172  012746  027520                                                        MOV     #TXDMA,-(SP)
035176  016746  144766                                                        MOV     TXVECA,-(SP)
035202  012746  000003                                                        MOV     #3,-(SP)
035206  104437                                                                TRAP    C$SVEC
035210  062706  000010                                                        ADD     #10,SP
035214                                  SETVEC  RXVECA,#RXCHRS,#PRIO6   ;SELECT RX INT SERVICE RTN.
035214  012746  000300                                                        MOV     #PRIO6,-(SP)
035220  012746  027310                                                        MOV     #RXCHRS,-(SP)
035224  016746  144736                                                        MOV     RXVECA,-(SP)
035230  012746  000003                                                        MOV     #3,-(SP)
035234  104437                                                                TRAP    C$SVEC
035236  062706  000010                                                        ADD     #10,SP
035242                                  SETPRI  #PRIO4         ;ALLOW INTERRUPTS.
035242  012700  000200                                                        MOV     #PRIO4,R0
035246  104441                                                                TRAP    C$SPRI
```

```
8543                                    ;*
8544                                    ; TRANSMIT AND RECEIVE SHORT DATA PATTERN AT ALL BAUDRATES,
8545                                    ;   WITH 8 BITS PER CHARACTER, 1 STOP BIT, AND BOTH TYPES OF PARITY,
8546                                    ;   BOTH LINE GROUPS (LGPRS) TX AND RX WITH THE SAME PARAMETERS.
8547                                    ;-
8548 035250  012700  003302                    MOV     #ERCNTB,R0
8549 035254  004767  162172                   JSR     PC,CLR16W       ;CLEAR THE RX ERROR COUNTERS TABLE.
8550 035260  012701  010470                    MOV     #10470,R1       ;SET UP LPR CONTENTS FOR TX/RX AT 75 BAUD.
8551 035264  004767  163340            4$:     JSR     PC,GETTIM       ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
8552 035270  012702  005154                    MOV     #SDP2B,R2       ;SET UP THE START ADR OF THE DATA PATTERN.
8553 035274  012703  000020                    MOV     #SDP2E-SDP2B,R3 ;SET UP THE DATA PATTERN LENGTH.
8554 035300  012704  000001                    MOV     #1,R4           ;SPECIFY TO SEND 1 DATA PATTERN TO EACH LINE.
8555 035304  004767  171376                   JSR     PC,VANSUP       ;SET UP "VANILLA FLAVORED" TX/RX.
8556 035310  012767  177400  144714           MOV     #177400,IBM     ;FORM BIT MAP OF UNUSED TX/RX BITS.
8557 035316  012767  021616  147772           MOV     #9102.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 9102.
8558                                    ;*
8559                                    ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 9102 THRU 9104 <<<<<.
8560                                    ;-
8561 035324  004767  165474                   JSR     PC,PUFIFR       ;PURGE THE DUT RECEIVE CHARACTER FIFO.
8562 035330  103402                    BCS     .+6
8563 035332  000167  000452                   JMP     50$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
8564
8565 035336  004767  165664                   JSR     PC,PURRXB       ;PURGE THE RX CHAR BUFFER IN MEMORY.
8566 035342  004767  163340                   JSR     PC,INIDMA       ;SEND THE FIRST BATCH OF DATA PATTERNS.
8567 035346  012767  021621  147742           MOV     #9105.,ERRNBR   ;SET ERROR NUMBER TO 9105.
8568                                    ;*
8569                                    ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 9105 THRU 9110 <<<<<.
8570                                    ;-
8571 035354  004767  165702                   JSR     PC,RDCHRS       ;READ AND VERIFY THE RX CHARACTERS.
8572 035360  005767  144640                   TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
8573 035364  001406                    BEQ     5$              ;NO, THEN BRANCH.
8574 035366  032767  000100  144566           BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
8575 035374  001002                    BNE     .+6
8576 035376  000167  000406                   JMP     50$             ;NO, THEN EXIT THE TEST.
8577
8578 035402  012767  021627  147706    5$:     MOV     #9111.,ERRNBR   ;SET ERROR NUMBER TO 9111.
8579                                    ;*
8580                                    ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 9111 THRU 9114 <<<<<.
8581                                    ;-
8582 035410  004767  170724                   JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM RX/RX.
8583 035414  005767  144604                   TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
8584 035420  001404                    BEQ     6$              ;NO, THEN BRANCH.
8585 035422  032767  000100  144532           BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
8586 035430  001567                    BEQ     50$             ;NO, THEN EXIT THE TEST.
8587                                    ;*
8588                                    ; TOGGLE THE PARITY TYPE BIT SPECIFIER IN THE TX/RX SETUP PARAMETERS.
8589                                    ; SELECT THE NEXT BAUDRATE AND PERFORM THE TEST AGAIN IF NOT DONE.
8590                                    ;-
8591 035432  010100                6$:     MOV     R1,R0           ;COMPLEMENT THE PARITY TYPE
8592 035434  042701  000100                   BIC     #100,R1         ;  BIT IN THE TX/RX LPR SETUP
8593 035440  005100                    COM     R0              ;  PARAMETER LEAVING THE
8594 035442  042700  177677                   BIC     #177677,R0      ;  OTHER LPR PARAMETER
8595 035446  050001                    BIS     R0,R1           ;  BITS UNCHANGED.
8596 035450  062701  010400                   ADD     #10400,R1       ;SELECT THE NEXT BAUDRATE.
8597 035454  103303                    BCC     4$              ;LOOP TO TX/RX AGAIN IF NOT PAST LAST BAUDRATE.
8598                                    ;*
8599                                    ; PERFORM WIDE OPEN DMA TEST.
```

```
                                        ; TRANSMIT AND RECEIVE 512 BYTE DATA PATTERNS AT ALL COMBINATIONS OF 9.6K,
                                        ; 19.2K, AND 38.4K BUADRATES AND 5 AND 8 BITS PER CHARACTER.  USE 1 STOP BIT
                                        ; AND NO PARITY GENERATION OR DETECTION.
                                        ;
                                            ;+
                                            ; INITIALIZE THE 512 BYTE PATTERN AND THE VARIOUS DATA PATTERN POINTERS.
                                            ;-
035456  005001                                CLR     R1              ;CLEAR THE DATA BYTE COUNTER.
035460  012702  003602                        MOV     #BUFBAS,R2      ;GET THE BASE OF THE DATA PATTERN BUFFER.
035464  110122                        7$:     MOVB    R1,(R2)+        ;WRITE A BYTE OF THE DATA PATTERN.
035466  105201                                INCB    R1              ;GET THE NEXT BYTE FOR THE DATA PATTERN.
035470  001375                                BNE     7$              ;LOOP UNTIL FIRST 1/2 OF PATTERN IS DONE.
035472  105301                        8$:     DECB    R1              ;GET THE NEXT BYTE FOR THE DATA PATTERN.
035474  110122                                MOVB    R1,(R2)+        ;WRITE A BYTE OF THE DATA PATTERN.
035476  105701                                TSTB    R1              ;CHECK FOR DONE WRITING DATA PATTERN.
035500  001374                                BNE     8$              ;LOOP IF DATA PATTERN IS NOT DONE.
035502  110122                        10$:    MOVB    R1,(R2)+        ;WRITE A BYTE OF THE 32 BYTE OVERFLOW REGION.
035504  005201                                INC     R1              ;COUNT THIS BYTE.
035506  020127  000040                        CMP     R1,#32.         ;TEST FOR 32 BYTES WRITTEN.
035512  001373                               BNE     10$             ;LOOP UNTIL 32 BYTES ARE WRITTEN.
                                            ;+
                                            ; PREPARE TO LOOP ON THE 3 DIFFERENT BAUDRATES (9.6K, 19.2K, AND 38.4K).
                                            ;-
035514  012705  005072                        MOV     #DLPRTB,R5      ;GET THE BASE ADR OF THE DMA BAUDRATE TABLE.
                                            ;+
                                            ; SPECIFY THE PROPER BAUDRATE.
                                            ; SPECIFY 8 BITS PER CHARACTER.
                                            ; PERFORM DMA TRANSMISSION AND RECEPTION OF 512 BYTE DATA PATTERN.
                                            ;-
                                                ;+
                                                ; THE FOLLOWING ROUTINE REPORTS THE ERROR WITH NUMBERS 914 THRU 921.
                                                ;    LPR CHANGE BIT ERROR FLAGS MAY BE SET BY THIS SUBROUTINE.
                                                ;-
035520  012501                        12$:     MOV     (R5)+,R1        ;SET UP LPR PARAM AT NEXT BAUD, 8 BITS/CHAR.
035522  004767  163102                        JSR     PC,GETTIM       ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
035526  012702  003602                         MOV     #BUFBAS,R2      ;SET UP THE START ADR OF THE DATA PATTERN.
035532  012703  001000                         MOV     #512.,R3        ;SET UP THE DATA PATTERN LENGTH.
035536  012704  000001                         MOV     #1,R4           ;SPECIFY TO SEND 1 DATA PATTERN TO EACH LINE.
035542  004767  171140                        JSR     PC,VANSUP       ;SET UP "VANILLA FLAVORED" TX/RX.
035546  012767  177400  144456                 MOV     #177400,IBM     ;FORM BIT MAP OF UNUSED BITS FOR 8 BITS/CHAR.
035554  012767  021633  147534                 MOV     #9115.,ERRNBR   ;SET ERROR NUMBER TO 9115.
                                                ;+
                                                ; THIS ROUTINE REPORTS ERROS WITH NUMBERS >>>>> 9115 THRU 9117 <<<<<.
                                                ;-
035562  004767  165236                        JSR     PC,PUFIFR       ;PURGE THE DUT RECEIVE CHARACTER FIFO.
035566  103126                               BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
035570  012767  021636  147520                 MOV     #9118.,ERRNBR   ;SET ERROR NUMBER TO 9118.
035576  004767  165424                        JSR     PC,PURRXB       ;PURGE THE RX CHAR BUFFER IN MEMORY.
035602  004767  163100                        JSR     PC,INIDMA       ;SEND THE FIRST BATCH OF DATA PATTERNS.
                                            ;+
                                            ; THIS ROUTINE REPORTS THE ERROR WITH NUMBERS >>>>> 9118 THRU 9123 <<<<<.
                                            ;-
035606  004767  165450                        JSR     PC,RDCHRS       ;READ AND VERIFY THE RX CHARACTERS.
035612  005767  144406                         TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
035616  001407                               BEQ     14$             ;NO, THEN BRANCH.
```

```
8657 035620  032767  000100  144334          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
8658 035626  001470                          BEQ     50$             ;NO, THEN EXIT THE TEST.
8659
8660 035630  012767  021644  147460          MOV     #9124.,ERRNBR   ;SET ERROR NUMBER TO 9124.
8661                                        ;+
8662                                        ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 9124 THRU 9127 <<<<<.
8663                                        ;-
8664 035636  004767  170476          14$:   JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM RX/RX.
8665 035642  005767  144356                  TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
8666 035646  001404                          BEQ     16$             ;NO, THEN BRANCH.
8667 035650  032767  000100  144304          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
8668 035656  001454                          BEQ     50$             ;NO, THEN EXIT THE TEST.
8669
8670 035660  012767  021650  147430  16$:    MOV     #9128.,ERRNBR   ;SET ERROR NUMBER TO 9128.
8671                                        ;+
8672                                        ; SPECIFY 5 BITS PER CHARACTER.
8673                                        ; PERFORM DMA TRANSMISSION AND RECEPTION OF 512 BYTE DATA PATTERN.
8674                                        ;-
8675 035666  042701  000030                  BIC     #30,R1          ;SET UP CHAR LENGTH PARAM TO 5 BITS/CHAR.
8676 035672  004767  171010                  JSR     PC,VANSUP       ;SET UP "VANILLA FLAVORED" TX/RX.
8677 035676  012767  177740  144326          MOV     #177740,IBM     ;FORM BIT MAP OF UNUSED BITS FOR 5 BITS/CHAR.
8678                                      ;+
8679                                      ; THIS  ROUTINE REPORTS THE ERROR WITH NUMBERS >>> 9128 THRU 9131 <<<.
8680                                      ;-
8681 035704  004767  165114                JSR     PC,PUFIFR       ;PURGE THE DUT RECEIVE CHARACTER FIFO.
8682 035710  103055                        BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
8683 035712  012767  021654  147376          MOV     #9132.,ERRNBR   ;SET THE ERROR REPORT NUMBER TO 9132.
8684
8685 035720  004767  165302                JSR     PC,PURRXB       ;PURGE THE RX CHAR BUFFER IN MEMORY.
8686 035724  004767  162756                JSR     PC,INIDMA       ;SEND THE FIRST BATCH OF DATA PATTERNS.
8687                                      ;+
8688                                      ; THIS ROUTINE REPORTS THE ERROR WITH NUMBERS >>>>> 9132 THRU 9137 <<<<<.
8689                                      ;-
8690 035730  004767  165326                JSR     PC,RDCHRS       ;READ AND VERIFY THE RX CHARACTERS.
8691 035734  005767  144264                TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
8692 035740  001404                        BEQ     18$             ;NO, THEN BRANCH.
8693 035742  032767  000100  144212          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
8694 035750  001417                        BEQ     50$             ;NO, THEN EXIT THE TEST.
8695
8696 035752  012767  021662  147336  18$:    MOV     #9138.,ERRNBR   ;SET ERROR NUMBER TO 9138.
8697                                      ;+
8698                                      ; THIS ROUTINE REPORTS THE ERROR WITH NUMBERS >>>>> 9138 THRU 9141 <<<<<.
8699                                      ;-
8700 035760  004767  170354                JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM RX/RX.
8701 035764  005767  144234                TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
8702 035770  001404                        BEQ     20$             ;NO, THEN BRANCH.
8703 035772  032767  000100  144162          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
8704 036000  001403                        BEQ     50$             ;NO, THEN EXIT THE TEST.
8705
8706 036002  020527  005100          20$:    CMP     R5,#DLPRTE      ;COMPARE DMA BAUDRATE TABLE PTR WITH TABLE END.
8707 036006  103644                        BLO     12$             ;LOOP IF NOT ALL BAUDRATES DONE YET.
8708                                    ;+
8709                                    ; ALL DONE.  HAVE EITHER RUN OUT OF ACTIVE LINES, OR COMPLETED THE TEST.
8710                                    ;    DISABLE INTERRUPTS.
8711                                    ;    CLEAR THE INTERRUPT VECTORS.
8712                                    ;-
8713 036010                          50$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
```

E2


```
        036010  012700  000340                                                          MOV     #PRIO7,R0
        036014  104441                                                                  TRAP    C$SPRI
   8714 036016                          CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
        036016  016700  144146                                                          MOV     TXVECA,R0
        036022  104436                                                                  TRAP    C$CVEC
   8715 036024                          CLRVEC  RXVECA          ;RETURN RX INT VECTOR TO UNUSED POOL.
        036024  016700  144136                                                          MOV     RXVECA,R0
        036030  104436                                                                  TRAP    C$CVEC
   8716
   8717 036032  012767  021666  147256          MOV     #9142.,ERRNBR   ;SELECT NUMBER 9142 FOR THE NEXT ERROR REPORT.
   8718 036040  004767  166056                  JSR     PC,REPSMR       ;REPORT ERROR SUMMARIES IF CALLED FOR.
   8719 036044                          60$:    SETPRI  #PRIO7          ;DISABLE ALL INTERRUPTS.
        036044  012700  000340                                                          MOV     #PRIO7,R0
        036050  104441                                                                  TRAP    C$SPRI
   8720 036052  005067  144142                  CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
   8721 036056                          ENDTST
        036056                                                                  L10033:
        036056  104401                                                                  TRAP    C$ETST
```

```
 8723                                   .SBTTL  HARDWARE TEST           - SPLSPD -
 8724                                   ;+**************************************************************************
 8725                                   ;*                          - SPLIT SPEED TEST -
 8726                                   ;*      THIS TEST IS USED TO VERIFY THE SPLIT SPEED CAPABILITIES OF THE DHU11,
 8727                                   ;*      AND THE CORRECT OPERATION OF THE A & B BAUD RATE GROUP SELECTION.
 8728                                   ;*      THE TEST USES THREE SETS OF BAUD RATES (38.4,50; 1200,75; 2000,2400).
 8729                                   ;*      THIS TEST WILL ONLY EXECUTE IF THE STAGGARED LOOPBACK MODE IS SELECTED.
 8730                                   ;*      THE SPECIAL STAGGARED LOOPBACK BERG CONNECTOR MUST BE FITTED.
 8731                                   ;*
 8732                                   ;--**************************************************************************
 8733 036060                                    BGNTST
      036060                                                                                    T8::
 8734 036060  126727  144110  000002             CMPB    LOPBCK,#2       ;CHECK MODE SELECTED.
 8735 036066  001402                            BEQ     2$              ;DO NOT EXIT IF STAGGERD LOPBCK MODE SELECTED.
 8736 036070  000167  000516                    JMP     60$             ;EXIT THIS TEST.
 8737 036074                            2$:     SETPRI  #PRI05          ;ALLOW LTC INTERRUPTS.
      036074  012700  000240                                                            MOV     #PRI05,R0
      036100  104441                                                                   TRAP    C$SPRI
 8738            000010                         TNUM == TNUM + 1        ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
 8739 036102  012767  000010  144150            MOV     #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (92)
 8740 036110  012767  177777  144102            MOV     #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
 8741 036116  012767  000001  147170            MOV     #1,ERRTYP       ;SET ERROR TYPE IN ERROR TABLE.
 8742 036124  012767  021761  147164            MOV     #9201.,ERRNBR   ;SET THE FIRST ERROR NUMBER IN ERROR TABLE.
 8743 036132  012767  012605  147160            MOV     #EM9201,ERRMSG  ;SET ERROR MESSAGE ADDRESS IN ERROR TABLE.
 8744 036140  005067  144334                    CLR     ERSMRF          ;INITIALIZE THE "REPORT ERROR SUMMARY" FLAGS.
 8745 036144  005067  144054                    CLR     FERROR          ;CLEAR THE "AT LEAST ONE ERROR" FLAG.
 8746                                   ;+
 8747                                   ; RESET THE DUT TO A KNOWN STATE, REMOVE STATUS CODES FROM THE FIFO.
 8748                                   ;  CLEAR TX AND  RX INTERRUPT ENABLE BITS.
 8749                                   ; THIS SUBROUTINE REPORTS ERROR >>>>> 9201 <<<<<.
 8750                                   ;-
 8751 036150  004767  161254                    JSR     PC,CLNRST       ;RESET THE DUT.
 8752 036154  103402                            BCS     .+6
 8753 036156  000167  000430                    JMP     60$             ;ABORT THE TEST IF FATAL ERROR FOUND IN RESET.
 8754                                   ;+
 8755                                   ; DISABLE ALL INTERRUPTS.
 8756                                   ; SET UP  DMA TX AND RX INTERRUPT SERVICE ROUTINES.
 8757                                   ;-
 8758 036162                                    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
      036162  012700  000340                                                            MOV     #PRI07,R0
      036166  104441                                                                    TRAP    C$SPRI
 8759 036170                                    SETVEC  TXVECA,#TXDMA,#PRI06    ;SELECT DMA TX INT SERVICE RTN.
      036170  012746  000300                                                            MOV     #PRI06,-(SP)
      036174  012746  027520                                                            MOV     #TXDMA,-(SP)
      036200  016746  143764                                                            MOV     TXVECA,-(SP)
      036204  012746  000003                                                            MOV     #3,-(SP)
      036210  104437                                                                    TRAP    C$SVEC
      036212  062706  000010                                                            ADD     #10,SP
 8760 036216                                    SETVEC  RXVECA,#RXCHRS,#PRI06   ;SELECT RX INT SERVICE RTN.
      036216  012746  000300                                                            MOV     #PRI06,-(SP)
      036222  012746  027310                                                            MOV     #RXCHRS,-(SP)
      036226  016746  143734                                                            MOV     RXVECA,-(SP)
      036232  012746  000003                                                            MOV     #3,-(SP)
      036236  104437                                                                    TRAP    C$SVEC
      036240  062706  000010                                                            ADD     #10,SP
 8761 036244                                    SETPRI  #PRI04          ;ALLOW INTERRUPTS.
      036244  012700  000200                                                            MOV     #PRI04,R0
```

```
036250  104441                                                  TRAP    C$SPRI
                                 ;+
                                 ; ENABLE TRANSMITTERS ON ALL LINES.
                                 ;
036252  012705  177777                  MOV     #MAPLNS,R5      ;PASS ACTIVE LINE BIT MAP.
036256  004767  167240                  JSR     PC,TXENBL       ;ENABLE TRANSMISSIONS ON ALL LINES.

                                 ;+
                                 ; CLEAR ERROR TABLE PRIOR TO PERFOMING TX/RX TEST.
                                 ;-
036262  012700  003302                  MOV     #ERCNTB,R0      ;GET THE BASE ADDRESS OF THE ERROR COUNTER TBL.
036266  004767  161160                  JSR     PC,CLR16W       ;CLEAR THE RX ERROR COUNTERS TABLE.

                                 ;+
                                 ; PERFORM SPLIT SPEED DMA TX AND RX ON ALL SELECTED LINES AT THE FOLLOWING
                                 ; BAUD RATES.
                                 ; 38.4K, 50 ; 1200, 75 ; 2000, 2400.
                                 ;-
                                  ;+
                                  ; INITIALISE DMA TX/RX PARAMETERS IN THE CONTROL BLOCK FR EACH OF THE BAUD
                                  ; RATES MENTIONED ABOVE.
                                  ;  8 BITS/CHAR,1 STOP BITS,ODD PARITY.
                                  ;-
036272  012705  005100                  MOV     #SPLPRB,R5      ;GET BASE ADDRESS OF LPR PARAMETER TABLE.
036276  012500                   4$:    MOV     (R5)+,R0        ;GET LPR CONTENTS FOR LINGRP II.
036300  012501                          MOV     (R5)+,R1        ;GET LPR CONTENTS FOR LINGRP I.
036302  004767  162322                  JSR     PC,GETTIM       ;GET TIME-OUT BASED ON MINIMUM BAUDRATE IN USE.
036306  012702  005154                  MOV     #SDP2B,R2       ;SET UP THE START ADR OF THE DATA PATTERN.
036312  012503                          MOV     (R5)+,R3        ;GET NUMBER OF REPEAT TRANSMISSION ON LINGRPII.
036314  012504                          MOV     (R5)+,R4        ;GET NUMBER OF REPEAT TRANSMISSION ON LINGRPI.
036316  004767  166454                  JSR     PC,SPLSUP       ;SET UP CONTROL BLOCK ETC. FOR TX/RX.
036322  012767  021762  146766          MOV     #9202.,ERRNBR   ;SET THE ERROR NUMBER TO 9202.
                                   ;+
                                   ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 9202 THRU 9204 <<<<<.
                                   ;-
036330  004767  164470                  JSR     PC,PUFIFR       ;PURGE THE DUT RECEIVE CHARACTER FIFO.
036334  103126                          BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
036336  012767  021765  146752          MOV     #9205.,ERRNBR   ;SET ERROR NUMBER TO 9205.

036344  004767  164656                  JSR     PC,PURRXB       ;PURGE THE RX CHAR BUFFER IN MEMORY.
036350  004767  162332                  JSR     PC,INIDMA       ;SEND THE FIRST BATCH OF DATA PATTERNS.
                                     ;+
                                     ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 9205 THRU 9210 <<<<<.
                                     ;-
036354  004767  164702                  JSR     PC,RDCHRS       ;READ AND VERIFY THE RX CHARACTERS.
036360  005767  143640                  TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
036364  001404                          BEQ     6$              ;NO, THEN BRANCH.
036366  032767  000100  143566          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
036374  001473                          BEQ     50$             ;NO, THEN EXIT THE TEST.

036376  012767  021773  146712   6$:    MOV     #9211.,ERRNBR   ;SET THE ERROR NUMBER TO 9211.
                                     ;+
                                     ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 9211 THRU 9214 <<<<<.
                                     ;-
036404  004767  167730                  JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM RX/RX.
036410  005767  143610                  TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
036414  001404                          BEQ     8$              ;NO, THEN BRANCH.
```

```
8818 036416  032767  000100  143536          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
8819 036424  001457                          BEQ     50$             ;NO, THEN EXIT THE TEST.
8820
8821 036426  012767  021777  146662  8$:     MOV     #9215.,ERRNBR   ;SET ERROR NUMBER TO 9215.
8822                                        ;+
8823                                        ; SWAP PARAMETERS TO ALLOW FOR BOTH CHANNELS TO BE EXERCISED.
8824                                        ;
8825 036434  010246                          MOV     R2, (SP)        ;PUSH THE START ADDRESS ONTO THE STACK.
8826 036436  010002                          MOV     R0,R2           ;
8827 036440  010100                          MOV     R1,R0           ;
8828 036442  010201                          MOV     R2,R1           ;SWAP THE TWO SETS OF
8829 036444  010302                          MOV     R3,R2           ; PARAMETERS OVER.
8830 036446  010403                          MOV     R4,R3           ;
8831 036450  010204                          MOV     R2,R4           ;
8832 036452  012602                          MOV     (SP)+,R2        ;RESTORE THE START ADDRESS.
8833 036454  004767  166316                  JSR     PC,SPLSUP       ;SET UP CONTROL BLOCK ETC, FOR TX/RX.
8834
8835                                      ;+
8836                                      ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 9215 THRU 9217 <<<<<.
8837                                      ;-
8838 036460  004767  164340                  JSR     PC,PUFIFR       ;PURGE THE DUT RECEIVE CHARACTER FIFO.
8839 036464  103052                          BCC     60$             ;ABORT THIS TEST IF FIFO WOULD NOT PURGE.
8840 036466  012767  022002  146622          MOV     #9218.,ERRNBR   ;SET ERROR NUMBER TO 9218.
8841
8842 036474  004767  164526                  JSR     PC,PURRXB       ;PURGE THE RX CHAR BUFFER IN MEMORY.
8843 036500  004767  162202                  JSR     PC,INIDMA       ;SEND THE FIRST BATCH OF DATA PATTERNS.
8844                                      ;+
8845                                      ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 9218 THRU 9223 <<<<<.
8846                                      ;-
8847 036504  004767  164552                  JSR     PC,RDCHRS       ;READ AND VERIFY THE RX CHARACTERS.
8848 036510  005767  143510                   TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
8849 036514  001404                          BEQ     10$             ;NO, THEN BRANCH.
8850 036516  032767  000100  143436          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
8851 036524  001417                          BEQ     50$             ;NO, THEN EXIT THE TEST.
8852
8853 036526  012767  022010  146562  10$:     MOV     #9224.,ERRNBR   ;SET ERROR NUMBER TO 9224.
8854                                      ;+
8855                                      ; THIS ROUTINE REPORTS ERRORS WITH NUMBERS >>>>> 9224 THRU 9227 <<<<<.
8856                                      ;-
8857 036534  004767  167600                  JSR     PC,TXRREP       ;REPORT FINAL ERRORS FROM RX/RX.
8858 036540  005767  143460                   TST     FERROR          ;HAS AN ERROR BEEN DETECTED ?
8859 036544  001404                          BEQ     12$             ;NO, THEN BRANCH.
8860 036546  032767  000100  143406          BIT     #BIT06,OPTION   ;HAS EXTENDED ERROR REPORTING BEEN REQUESTED ?
8861 036554  001403                          BEQ     50$             ;NO, THEN EXIT THE TEST.
8862
8863 036556  020527  005130          12$:     CMP     R5,#SPLPRE      ;CHECK IF ALL PARAMETERS HAVE BEEN DONE.
8864 036562  103645                          BLO     4$              ;IF NOT DONE LOOP TO SELECT THE NEXT PARAMETER.
8865                                      ;+
8866                                      ;    DISABLE INTERRUPTS.
8867                                      ;    CLEAR THE INTERRUPT VECTORS.
8868                                      ;-
8869 036564                          50$:    SETPRI  #PRI07          ;DISABLE ALL INTERRUPTS.
     036564  012700  000340                                                          MOV     #PRI07,R0
     036570  104441                                                                  TRAP    C$SPRI
8870 036572                                  CLRVEC  TXVECA          ;RETURN TX INT VECTOR TO UNUSED POOL.
     036572  016700  143372                                                          MOV     TXVECA,R0
     036576  104436                                                                  TRAP    C$CVEC
```

```
8871
8872 036600  012767  022014  146510          MOV     #9228.,ERRNBR   ;SELECT NUMBER 9228 FOR THE NEXT ERROR REPORT.
8873 036606  004767  165310                  JSR     PC,REPSMR       ;REPORT ERROR SUMMARIES IF CALLED FOR.
8874 036612  005067  143402          60$:    CLR     CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
8875 036616                                  ENDTST
     036616                                                                  L10034:
     036616  104401                                                                  TRAP    C$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.
                                        ;*
                                        ;-- ***********************************************************************
036620                                          BGNTST
036620                                                                                  T9::
        000011                                    TNUM == TNUM + 1      ;INCREMENT THE ASSEMBLY TIME TEST COUNTER.
036620  012767  000011  143432                    MOV   #TNUM,TSTNUM    ;SET UP THE TEST NUMBER.        (93)
036626  012767  177777  143364                    MOV   #-1,CTRLCF      ;INDICATE THAT WE ARE IN A TEST.
036634  016702  143650                            MOV   BMPCQP,R2       ;GET THE CONTENTS OF THE POINTER.
036640  012703  002512                            MOV   #BMPCQB,R3      ;GET THE START ADDRESS OF THE QUEUE.
036644  020203                                    CMP   R2,R3           ;SEE IF THE POINTER HAS MOVED FROM THE BASE.
036646  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"

036650  012701  013011                          MOV     #EM9304,R1      ;PASS THE FIRST MESSAGE TO BE REORTED.
036654                                          ERRDF   9301,EM9301,ER9301 ;            >>>>> ERROR #9301 <<<<<.
036654  104455                                                                                  TRAP    C$ERDF
036656  022125                                                                                  .WORD   9301
036660  012635                                                                                  .WORD   EM9301
036662  015720                                                                                  .WORD   ER9301

036664  012767  002512  143616                    MOV   #BMPCQB,BMPCQP  ;SET POINTER BACK TO THE BEGINING OF THE QUE.

036672  005067  143322                  60$:      CLR   CTRLCF          ;INDICATE THAT WE ARE NOT WITHIN A TEST.
036676                                          ENDTST
036676                                                                                  L10035:
036676  104401                                                                                  TRAP    C$ETST
```

```
                                ;*******************************************************************************
                                ;
                                ;               FVTC.HWQ
                                ;
                                ;*******************************************************************************


                                .SBTTL  HARDWARE PARAMETER CODING SECTION



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

 036700                                 BGNHRD
 036700  000027                                                                          .WORD L10036-L$HARD/2
 036702                                                                         L$HARD::

                                ;DEVICE CSR ADDRESS QUESTION:
 036702                                 GPRMA   HWPTQ1,0,0,160000,177776,YES
 036702  000031                                                                          .WORD   T$CODE
 036704  036760                                                                          .WORD   HWPTQ1
 036706  160000                                                                          .WORD   T$LOLIM
 036710  177776                                                                          .WORD   T$HILIM
                                ;DEVICE INTERRUPT VECTOR QUESTION:
 036712                                 GPRMA   HWPTQ2,2,0,40,776,YES
 036712  001031                                                                          .WORD   T$CODE
 036714  036776                                                                          .WORD   HWPTQ2
 036716  000040                                                                          .WORD   T$LOLIM
 036720  000776                                                                          .WORD   T$HILIM
                                ;ACTIVE LINES BIT MAP QUESTION:
 036722                                 GPRMD   HWPTQ3,4,0,MAPLNS,0,MAPLNS,YES
 036722  002032                                                                          .WORD   T$CODE
 036724  037031                                                                          .WORD   HWPTQ3
 036726  177777                                                                          .WORD   MAPLNS
 036730  000000                                                                          .WORD   T$LOLIM
 036732  177777                                                                          .WORD   T$HILIM
                                ;TYPE OF LOOPBACK QUESTION:
 036734                                 GPRMD   HWPTQ4,6,0,377,1,5,YES
 036734  003032                                                                          .WORD   T$CODE
 036736  037057                                                                          .WORD   HWPTQ4
 036740  000377                                                                          .WORD   377
 036742  000001                                                                          .WORD   T$LOLIM
 036744  000005                                                                          .WORD   T$HILIM
                                ;INTERRUPT BR LEVEL QUESTION:
 036746                                 GPRMD   HWPTQ5,6,0,177400,0,6,YES
 036746  003032                                                                          .WORD   T$CODE
 036750  037216                                                                          .WORD   HWPTQ5
 036752  177400                                                                          .WORD   177400
 036754  000000                                                                          .WORD   T$LOLIM
 036756  000006                                                                          .WORD   T$HILIM
```

```
8951
8952
8953 036760                                 ENDHRD
                                                                                .EVEN
     036760                                                               L10036:
8954
8961
8962 036760    103    123    122            HWPTQ1: .ASCIZ  /CSR ADDRESS: /
     036763    040    101    104
     036766    104    122    105
     036771    123    123    072
     036774    040    000
8963 036776    111    116    124            HWPTQ2: .ASCIZ  /INTERRUPT VECTOR ADDRESS: /
     037001    105    122    122
     037004    125    120    124
     037007    040    126    105
     037012    103    124    117
     037015    122    040    101
     037020    104    104    122
     037023    105    123    123
     037026    072    040    000
8964 037031    101    103    124            HWPTQ3: .ASCIZ  /ACTIVE LINE BIT MAP: /
     037034    111    126    105
     037037    040    114    111
     037042    116    105    040
     037045    102    111    124
     037050    040    115    101
     037053    120    072    040
     037056    000
8965 037057    124    131    120            HWPTQ4: .ASCII  /TYPE OF LOOPBACK (1=INTERNAL, 2=H3277, 3=H325/<15><12>
     037062    105    040    117
     037065    106    040    114
     037070    117    117    120
     037073    102    101    103
     037076    113    040    050
     037101    061    075    111
     037104    116    124    105
     037107    122    116    101
     037112    114    054    040
     037115    062    075    110
     037120    063    062    067
     037123    067    054    040
     037126    063    075    110
     037131    063    062    065
     037134    015    012
8966 037136    040    040    040                    .ASCIZ  /                    4=MODEM, 5=KEYBOARD ECHO): /
     037141    040    040    040
     037144    040    040    040
     037147    040    040    040
     037152    040    040    040
     037155    040    040    040
     037160    040    040    064
     037163    075    115    117
     037166    104    105    115
     037171    054    040    065
     037174    075    113    105
     037177    131    102    117
```

M2


```
            037202      101     122     104
            037205      040     105     103
            037210      110     117     051
            037213      072     040     000
    8967    037216      111     116     124         HWPTQ5: .ASCIZ  /INTERRUPT BR LEVEL: /
            037221      105     122     122
            037224      125     120     124
            037227      040     102     122
            037232      040     114     105
            037235      126     105     114
            037240      072     040     000
    8968
    8969                                            .EVEN
```

```
8972                            ;*****************************************************************************
*
8973                            ;
8974                            ;               FVTA.SWQ
8975                            ;
8976                            ;*****************************************************************************
*
8978
8979
8980                            .SBTTL  SOFTWARE PARAMETER CODING SECTION
8981
8982                            ;++
8983                            ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
8984                            ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
8985                            ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
8986                            ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
8987                            ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
8988                            ; WITH THE OPERATOR.
8989                            ; -
8990
8991 037244                             BGNSFT
     037244  000017                                                    .WORD L1003. -.SOFT/2
     037246                                                         L$SOFT::
8992
9001                            ;UNIT NUMBER PRINTOUT QUESTION:
9002 037246                             GPRML   SWPTQ1,0,20,YES
     037246  000130                                                    .WORD   T$CODE
     037250  037304                                                    .WORD   SWPTQ1
     037252  000020                                                    .WORD   20
9003
9004 037254                     ;REPORT NUMB OF BITS TESTED IN DMA ADDR TEST QUESTION:
                                        GPRML   SWPTQ2,0,40,YES
     037254  000130                                                    .WORD   T$CODE
     037256  037360                                                    .WORD   SWPTQ2
     037260  000040                                                    .WORD   40
9005                            ;EXTENDED ERROR REPORTING QUESTION:
9006 037262                             GPRML   SWPTQ3,0,100,YES
     037262  000130                                                    .WORD   T$CODE
     037264  037440                                                    .WORD   SWPTQ3
     037266  000100                                                    .WORD   100
9007                              ;+
9008                              ; IF EXTENDED ERROR REPORTING IS NOT REQUIRED THEN SKIP THE NEXT QUESTION.
9009                              ;-
9010 037270                             XFERF   ENDD
     037270  006044                                                    .WORD   T$CODE
9011
9012                            ;NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE QUESTION:
9013 037272                             GPRMD   SWPTQ4,2,D.177777,0,177777,YES
     037272  001052                                                    .WORD   T$CODE
     037274  037473                                                    .WORD   SWPTQ4
     037276  177777                                                    .WORD   177777
     037300  000000                                                    .WORD   T$LOLIM
     037302  177777                                                    .WORD   T$HILIM
9014                                    .EVEN
9015
9016 037304                     ENDD:   ENDSFT
                                                                       .EVEN
     037304                                                         L10037:
9017
9018
```

```
 9025 037304  122  105  120        SWPTQ1: .ASCIZ  /REPORT UNIT NUMBER AS EACH UNIT IS TESTED: /
      037307  117  122  124
      037312  040  125  116
      037315  111  124  040
      037320  116  125  115
      037323  102  105  122
      037326  040  101  123
      037331  040  105  101
      037334  103  110  040
      037337  125  116  111
      037342  124  040  111
      037345  123  040  124
      037350  105  123  124
      037353  105  104  072
      037356  040  000
 9026 037360  122  105  120        SWPTQ2: .ASCIZ  /REPORT NUMBER OF BITS TESTED IN DMA ADDR TEST: /
      037363  117  122  124
      037366  040  116  125
      037371  115  102  105
      037374  122  040  117
      037377  106  040  102
      037402  111  124  123
      037405  040  124  105
      037410  123  124  105
      037413  104  040  111
      037416  116  040  104
      037421  115  101  040
      037424  101  104  104
      037427  122  040  124
      037432  105  123  124
      037435  072  040  000
 9027 037440  105  130  124        SWPTQ3: .ASCIZ  /EXTENDED ERROR REPORTING: /
      037443  105  116  104
      037446  105  104  040
      037451  105  122  122
      037454  117  122  040
      037457  122  105  120
      037462  117  122  124
      037465  111  116  107
      037470  072  040  000
 9028 037473  116  125  115        SWPTQ4: .ASCIZ  /NUMBER OF INDIVIDUAL DATA ERRORS TO REPORT ON A LINE: /
      037476  102  105  122
      037501  040  117  106
      037504  040  111  116
      037507  104  111  126
      037512  111  104  125
      037515  101  114  040
      037520  104  101  124
      037523  101  040  105
      037526  122  122  117
      037531  122  123  040
      037534  124  117  040
      037537  122  105  120
      037542  117  122  124
      037545  040  117  116
      037550  040  101  040
      037553  114  111  116
```

```
        037556      105     072     040
        037561      000
   9029
   9030                                         .EVEN
```

```
 9032
 9033                                   ;**********************************************************************************************
 9034                                   ;
 9035                                   ;               FVTSKL6.P11
 9036                                   ;
 9037                                   ;**********************************************************************************************
 9038
 9039
 9040
 9041 037562                            $PATCH::
 9042 037562                                    .BLKW   24
 9043
 9050
 9051
 9052
 9053
 9054 037632                                    LASTAD
                                                                                .EVEN
      037632  000000                                                           .WORD   0
      037634  000000                                                           .WORD   0
      037636                            L$LAST::
 9055 037636                                    ENDMOD
 9056
 9057
 9058
 9059
 9060
 9061
 9062
 9063          000001                           .END
```

SYMBOL TABLE

```
ACTLNS  002172 G        CHCNTB  003442 G        C$OPEN= 000034          EF9012  007001 G        EM9401  013065 G
ADDR    025312          CHKEXT  016410 G        C$PNTB= 000014          EF9013  007115 G        ENDD    037304
ADR   = 000020 G        CHKLOS  016510 G        C$PNTF= 000017          EF9019  007162 G        ENDETB  004602 G
ADRPTR  017362 G        CHRTOT  002476 G        C$PNTS= 000016          EF9020  007201 G        ENDIT   030476
ALTFLD  016112 G        CKCHR   016612 G        C$PNTX= 000015          EF9101  007262 G        ERCNTB  003302 G
ASSEMB= 000010          CKFRPR  017030 G        C$QIO = 000377          EF9103  007265 G        ERLTBL  003602 G
BCOUNT  002306 G        CKINAC  017242 G        C$RDBU= 000007          EF9301  007333 G        ERRBLK  005322 G
BDRMSG  013137 G        CKTRAP  017350 G        C$REFG= 000047          EF9302  007401 G        ERRMSG  005320 G
BITSTD  033766          CKTRPB  017400 G        C$RESE= 000033          EMLMSG  013166 G        ERRNBR  005316 G
BITTBL  002364 G        CLKBRL  002272 G        C$REVI= 000003          EM0101  022162 G        ERRTYP  005314 G
BIT0  = 000001 G        CLKCSR  002270 G        C$RFLA= 000021          EM0102  022246 G        ERSMRF  002500 G
BIT00 = 000001 G        CLKHRZ  002276 G        C$RPT = 000025          EM0103  010041 G        ER0101  013572 G
BIT01 = 000002 G        CLKINT  027240 G        C$SEFG= 000046          EM0509  010077 G        ER0503  014124 G
BIT02 = 000004 G        CLKVEC  002274 G        C$SPRI= 000041          EM1601  010103 G        ER1603  014162 G
BIT03 = 000010 G        CLNRST  017430 G        C$SVEC= 000037          EM4401  010166 G        ER6201  014254 G
BIT04 = 000020 G        CLR16W  017452 G        C$TPRI= 000013          EM4402  010216 G        ER9001  014512 G
BIT05 = 000040 G        CONMAP  017474 G        DELAY   017550 G        EM4403  010264 G        ER9002  014612 G
BIT06 = 000100 G        CSRA    002200 G        DFPTBL  002150 G        EM4404  010361 G        ER9003  014770 G
BIT07 = 000200 G        CSRO  = 000000 G        DIAGMC= 000000          EM4405  010420 G        ER9004  015162 G
BIT08 = 000400 G        CTRLCF  002220 G        DLPRTB  005072 G        EM4406  010514 G        ER9005  015276 G
BIT09 = 001000 G        C$AU  = 000052          DLPRTE  005100 G        EM4407  010570 G        ER9101  015536 G
BIT1  = 000002 G        C$AUTO= 000061          DMRW    017662 G        EM4408  010652 G        ER9102  015576 G
BIT10 = 002000 G        C$BRK = 000022          DMTSTA  002222 G        EM4409  010715 G        ER9301  015720 G
BIT11 = 004000 G        C$BSEG= 000004          DM16B   017610 G        EM4410  010752 G        EVL   = 000004 G
BIT12 = 010000 G        C$BSUB= 000002          DODMA   020004 G        EM4411  011001 G        EXCNTB  003242 G
BIT13 = 020000 G        C$CEFG= 000045          DPENDB  003142 G        EM5303  011046 G        EXTMSG  013235 G
BIT14 = 040000 G        C$CLCK= 000062          DPLENB  003202 G        EM6201  011117 G        E$END = 002100
BIT15 = 100000 G        C$CLEA= 000012          DPRSQB  004642 G        EM6202  011151 G        E$LOAD= 000035
BIT2  = 000004 G        C$CLOS= 000035          DPRSQE  005042 G        EM6301  011160 G        FDATA   002206 G
BIT3  = 000010 G        C$CLP1= 000006          DRADRT  002200 G        EM8901  011211 G        FDATO = 000006 G
BIT4  = 000020 G        C$CVEC= 000036          DROP    030552          EM9003  011236 G        FERROR  002224 G
BIT5  = 000040 G        C$DCLN= 000044          DUMY    033770          EM9004  011260 G        FFREM   002226 G
BIT6  = 000100 G        C$DODU= 000051          EDPFMT  007733 G        EM9006  011276 G        FINACT  020074 G
BIT7  = 000200 G        C$DRPT= 000024          EDROP   030630          EM9007  011351 G        FRPSUP  020154 G
BIT8  = 000400 G        C$DU  = 000053          EF.CON= 000036 G        EM9008  011434 G        FSLSA   002206 G
BIT9  = 001000 G        C$EDIT= 000003          EF.NEW= 000035 G        EM9009  011515 G        FSLSO = 000006 G
BMPCQB  002512 G        C$ERDF= 000055          EF.PWR= 000034 G        EM9010  011541 G        F$AU  = 000015
BMPCQE  002712 G        C$ERHR= 000056          EF.RES= 000037 G        EM9011  011565 G        F$AUTO= 000020
BMPCQP  002510 G        C$ERRO= 000060          EF.STA= 000040 G        EM9012  011575 G        F$BGN = 000040
BOE   = 000400 G        C$ERSF= 000054          EF0503  005453 G        EM9013  011605 G        F$CLEA= 000007
BRLEVL  002175 G        C$ERSO= 000057          EF1601  005460 G        EM9014  011614 G        F$DU  = 000016
BRTBLB  002424 G        C$ESCA= 000010          EF1603  005512 G        EM9015  011710 G        F$END = 000041
BRTBLE  002464 G        C$ESEG= 000005          EF4401  005554 G        EM9016  011724 G        F$HARD= 000004
BUFBAS  003602 G        C$ESUB= 000003          EF6201  005671 G        EM9017  011733 G        F$HW  = 000013
BUFEND  004602 G        C$ETST= 000001          EF6202  006004 G        EM9025  012044 G        F$INIT= 000006
BUFMID  004202 G        C$EXIT= 000032          EF6203  006102 G        EM9026  012140 G        F$JMP = 000050
BUF3QT  004402 G        C$GETB= 000026          EF7801  006177 G        EM9027  012164 G        F$MOD = 000000
CALMSL  016164 G        C$GETW= 000027          EF9001  006235 G        EM9028  012244 G        F$MSG = 000011
CBB     003122 G        C$GMAN= 000043          EF9002  006317 G        EM9030  012323 G        F$PROT= 000021
CBDPAA  003126 G        C$GPHR= 000042          EF9003  006371 G        EM9101  012400 G        F$PWR = 000017
CBDPLA  003130 G        C$GPLO= 000030          EF9004  006420 G        EM9102  012442 G        F$RPT = 000012
CBDPNA  003132 G        C$GPRI= 000040          EF9005  006450 G        EM9104  012531 G        F$SEG = 000003
CBLNCA  003124 G        C$INIT= 000011          EF9006  006501 G        EM9201  012605 G        F$SOFT= 000005
CBLPBA  003136 G        C$INLP= 000020          EF9007  006520 G        EM9301  012635 G        F$SRV = 000010
CBLPRA  003122 G        C$MANI= 000050          EF9008  006614 G        EM9302  012714 G        F$SUB = 000002
CBMAPA  003134 G        C$MEM = 000031          EF9009  006653 G        EM9303  012744 G        F$SW  = 000014
CBOFSA  003140 G        C$MSG = 000023          EF9010  006712 G        EM9304  013011 G        F$TEST= 000001
```

SYMBOL TABLE

```
GETBDR  020420 G        I$TST = 000041          L$RPT   027652 G        MSTICK  002310 G        PRIO7 = 000340 G
GETCHR  020546 G        J$JMP = 000167          L$SOFT  037246 G        MUL16U  021532 G        PROTBL  005214 G
GETPRM  030270          LGRP1M  002236 G       L$SPC   002056 G        NDERPT  002164 G        PRPARE  022450 G
GETTIM  020630 G        LGRP2M  002240 G       L$SPCP  002020 G        NDPMSG  013316 G        PRTLPR  022660 G
GMANWD  002230 G        LINBIT  021000 G       L$SPTP  002024 G        NEWCHR  021606 G        PUFIFO  022742 G
GPRSOB  002464 G        LNCTRA  002210 G       L$STA   002030 G        NEWPAS  030250          PUFIFR  023024 G
G$CNTO= 000200          LNCTRO= 000010 G       L$SW    002162 G        NEWRES  030242          PURRXB  023226 G
G$DELM= 000372          LOE   = 040000 G       L$TEST  002114 G        NEWSTA  027732          RBUFA   002202 G
G$DISP= 000003          LOPBCK  002174 G       L$TIML  002014 G        NUMLNS= 000020 G        RBUFO = 000002 G
G$EXCP= 000400          LOT   = 000010 G       L$UNIT  002012 G        ODTSTA  033772          RDCHRS  023262 G
G$HILI= 000002          LPCSLT= 000036 G       L10000  002160          OOPS    022116 G        RDMAST  023722 G
G$LOLI= 000001          LPRA    002204 G       L10001  002166          OPTION  002162 G        REPCOD  023762 G
G$NO  = 000000          LPRO  = 000004 G       L10002  013706          O$APTS= 000000          REPSMR  024122 G
G$OFFS= 000400          L$ACP   002110 G       L10003  014160          O$AU  = 000000          RESETT  024150 G
G$OFSI= 000376          L$APT   002036 G       L10004  014252          O$BGNR= 000001          RRXNDN  024262 G
G$PRMA= 000001          L$AU    030636 G       L10005  014510          O$BGNS= 000001          RTXNDN  024330 G
G$PRMD= 000002          L$AUT   002070 G       L10006  014610          O$DU  = 000001          RXBCNT  002716 G
G$PRML= 000000          L$AUTO  030506 G       L10007  014766          O$ERRT= 000001          RXBDTX= 000030 G
G$RADA= 000140          L$CCP   002106 G       L10010  015160          O$GNSW= 000001          RXBEND  003120 G
G$RADB= 000000          L$CLEA  030510 G       L10011  015274          O$POIN= 000001          RXBETX= 000020 G
G$RADD= 000040          L$CO    002032 G       L10012  015534          O$SETU= 000000          RXBFUL= 000100 G
G$RADL= 000120          L$DEPO  002011 G       L10013  015574          PARATB  002324 G        RXBIPT  002714 G
G$RADO= 000020          L$DESC  005374 G       L10014  015716          PARATE  002344 G        RXBOPT  002712 G
G$XFER= 000004          L$DESP  002076 G       L10015  016110          PAROA   002324 G        RXBSTA  002720 G
G$YES = 000010          L$DEVP  002060 G       L10016  027656          PAR1A   002326 G        RXCHRS  027310 G
HELP  = 000000          L$DISP  002124 G       L10020  030504          PAR2A   002330 G        RXCNTB  003542 G
HOE   = 100000 G        L$DLY   002116 G       L10021  030506          PAR3A   002332 G        RXDONF  002504 G
HWPTQ1  036760          L$DTP   002040 G       L10022  030524          PAR4A   002334 G        RXDSBL  024376 G
HWPTQ2  036776          L$DTYP  002034 G       L10023  030634          PAR5A   002336 G        RXENBL  024472 G
HWPTQ3  037031          L$DU    030526 G       L10024  030642          PAR6A   002340 G        RXIE0   024566 G
HWPTQ4  037057          L$DUT   002072 G       L10025  031124          PAR7A   002342 G        RXIE1   024626 G
HWPTQ5  037216          L$DVTY  005364 G       L10026  031366          PASCNT  002242 G        RXPTRB  003402 G
IBE   = 010000 G        L$EF    002052 G       L10027  032320          PCSLOT= 000016 G        RXTIMO= 000002 G
IBM     002232 G        L$ENVI  002044 G       L10030  033776          PDRATB  002344 G        RXTMA   002202 G
IDU   = 000040 G        L$ERRT  005314 G       L10031  034412          PDRATE  002364 G        RXTOUT  002246 G
IER   = 020000 G        L$ETP   002102 G       L10032  035070          PDR0A   002344 G        RXVECA  002166 G
IESTAT  002234 G        L$EXP1  002046 G       L10033  036056          PDR1A   002346 G        R0SLOT= 000002 G
INIDMA  020706 G        L$EXP4  002064 G       L10034  036616          PDR2A   002350 G        R1SLOT= 000004 G
ISR   = 000100 G        L$EXP5  002066 G       L10035  036676          PDR3A   002352 G        R2SLOT= 000006 G
IXE   = 004000 G        L$HARD  036702 G       L10036  036760          PDR4A   002354 G        R3SLOT= 000010 G
I$AU  = 000041          L$HIME  002120 G       L10037  037304          PDR5A   002356 G        R4SLOT= 000012 G
I$AUTO= 000041          L$HPCP  002016 G       MAPLNS= 177777 G        PDR6A   002360 G        R5SLOT= 000014 G
I$CLN = 000041          L$HPTP  002022 G       MFUNIT  005422 G        PDR7A   002362 G        SAVBMP  024652 G
I$DU  = 000041          L$HW    002150 G       MMENAB  002322 G        PMSFLG  002244 G        SAVPRI  002250 G
I$HRD = 000041          L$ICP   002104 G       MMPRES  002320 G        PMSMSG  013444 G        SAVTEN  002252 G
I$INIT= 000041          L$INIT  027666 G       MMSR0   002314 G        PNT   = 001000 G        SCBCTB  005042 G
I$MOD = 000041          L$LADP  002026 G       MMSR3   002316 G        PREGRT  005346 G        SCBCTE  005052 G
I$MSG = 000041          L$LAST  037636 G       MODSUP  021026 G        PREG05  005324          SCBRTB  005052 G
I$PROT= 000040          L$LOAD  002100 G       MSFMT1  007577 G        PRFRME  022344 G        SCBRTE  005060 G
I$PTAB= 000041          L$LUN   002074 G       MSFMT2  007637 G        PRI   = 002000 G        SCNSTB  005060 G
I$PWR = 000041          L$MREV  002050 G       MSG1    013710 G        PRI00 = 000000 G        SCNSTE  005064 G
I$RPT = 000041          L$NAME  002000 G       MSG2    013766 G        PRI01 = 000040 G        SCTPTB  005064 G
I$SEG = 000041          L$PRIO  002042 G       MSG3    014045 G        PRI02 = 000100 G        SCTPTE  005072 G
I$SETU= 000041          L$PROT  027660 G       MSLCNT  002312 G        PRI03 = 000140 G        SDPBAS  005130 G
I$SFT = 000041          L$PRT   002112 G       MSLGET  021166 G        PRI04 = 000200 G        SDPEND  005150 G
I$SRV = 000041          L$REPP  002062 G       MSLOOP  021302 G        PRI05 = 000240 G        SDP2B   005154 G
I$SUB = 000041          L$REV   002010 G       MSSRPT  021316 G        PRI06 = 000300 G        SDP2E   005174 G
```

SYMBOL TABLE

```
SFPTBL  002162 G        TP4FLG  002254 G        TXPTRB  003342 G        T$SAVL= 177777          T2      031126 G
SKPSTS  024720 G        TP4RTN  027476 G        TXRINI  025776 G        T$SEGL= 177777          T3      031370 G
SPLPRB  005100 G        TP4VEC  002256 G        TXROFF  026252 G        T$SUBN= 000000          T4      032322 G
SPLPRE  005130 G        TRPAD2  017412 G        TXRON   026312 G        T$TAGL= 177777          T5      034000 G
SPLSUP  024776 G        TSTNUM  002260 G        TXRREP  026340 G        T$TAGN= 010040          T6      034414 G
STGTRB  005274 G        TXAD1A  002212 G        TXRXLB  005234 G        T$TEMP= 000000          T7      035072 G
STPSW   025276 G        TXAD10= 000012 G        TXRXLE  005274 G        T$TEST= 000011          T8      036060 G
SUCSS   033774           TXAD2A  002214 G        TXVECA  002170 G        T$TSTM= 177777          T9      036620 G
SVCGBL= 000000           TXAD20= 000014 G        T$ARGC= 000003          T$TSTS= 000001          UAM   = 000200 G
SVCINS= 000001           TXBFCA  002216 G        T$CODE= 001052          T$$AU = 010024          UBRFMT  007501 G
SVCSUB= 000001           TXBFCO= 000016 G        T$ERRN= 022125          T$$AUT= 010021          UNITN   002176 G
SVCTAG= 000001           TXCNTB  003502 G        T$EXCP= 000000          T$$CLE= 010022          UNSDIV  026454 G
SVCTST= 000001           TXDBLF  002506 G        T$FLAG= 000040          T$$DU = 010023          UPDCHR  026610 G
SWPTQ1  037304           TXDMA   027520 G        T$GMAN= 000000          T$$HAR= 010036          VANSUP  026706 G
SWPTQ2  037360           TXDONE  025316 G        T$HILI= 177777          T$$HW = 010000          WAIBIS  027104 G
SWPTQ3  037440           TXDONF  002502 G        T$LAST= 000001          T$$INI= 010020          WORD1   002266 G
SWPTQ4  037473           TXDSBL  025426 G        T$LOLI= 000000          T$$MSG= 010015          WTWLNC  027160 G
S$LSYM= 010000           TXENBL  025522 G        T$LSYM= 010000          T$$PRO= 010017          WTWLPR  027210 G
TERMSG  013531 G         TXENBM  002262 G        T$LTNO= 000011          T$$RPT= 010016          X$ALWA= 000000
TIMER1  002300 G         TXFRPR  025616 G        T$NEST= 177777          T$$SOF= 010037          X$FALS= 000040
TIMER2  002302 G         TXIE0   025712 G        T$NS0 = 000000          T$$SW = 010001          X$OFFS= 000400
TIMER3  002304 G         TXIE1   025752 G        T$NS1 = 000005          T$$TES= 010035          X$TRUE= 000020
TNUM  = 000011 G         TXINTF  002264 G        T$PTNU= 000000          T1      030644 G        $PATCH  037562 G
TP4BRT  027454 G

. ABS.  037636      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  30240 WORDS  ( 119 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:03:59
PARTC.BIN,PARTC.LST/-SP=SVC34R/ML,PARTC.P11
```