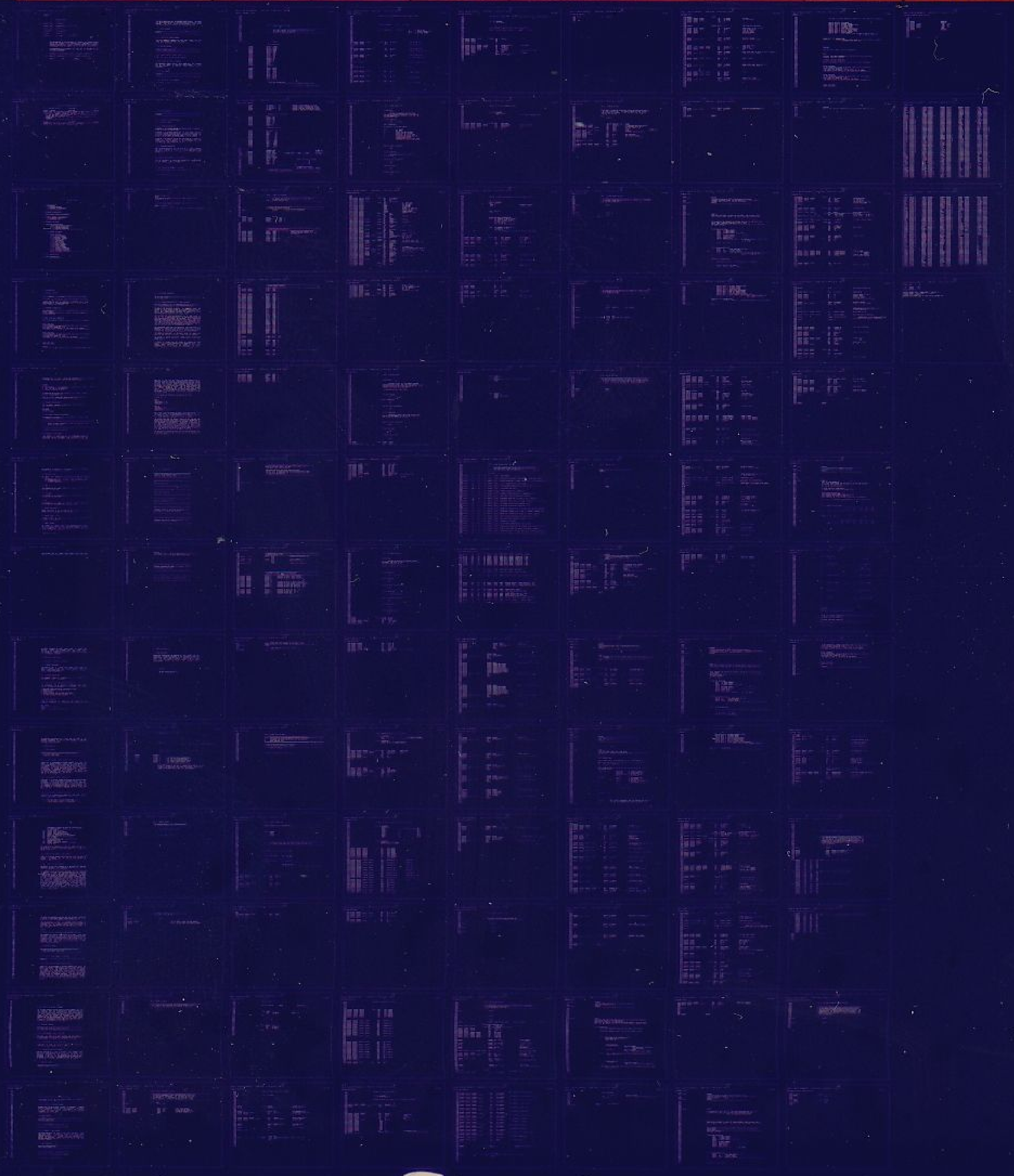
11/21+
KMV11A

KMV11A LINE CTRL DIAG
CNKMBB0

AH-T845B-MC
FICHE 01 OF 01

OCT 1984
digital
Made In USA



```
        .NLIST TOC
        .REM @


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

        PRODUCT CODE:   AC-T844B-MC

        PRODUCT NAME:   CNKMBB0 KMV11A LINE CNT DIAG

        PRODUCT DATE:   APRIL 1984

        MAINTAINER:     ISS DIAGNOSTICS

        AUTHOR:         MICHELET GUY

        MODIFIED BY:    JAKI BERG       9-APR-1984



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

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

                COPYRIGHT (C) 1982,1984 BY DIGITAL EQUIPMENT CORPORATION

        THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

                DIGITAL         PDP             UNIBUS          MASSBUS
                DEC             DECUS           DECTAPE
```

```

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

              REV A: ORIGINAL RELEASE          GUY MICHELET                 14-JAN-81

              CVKMBA => CNKMBA                 JAKI BERG                     9-APR-84
              CHANGES WERE MADE TO CVKMBA TO PRODUCE CNDMBA FOR THE FALCON-PLUS PROJECT
              (SBC-11/21+).   CHANGES, MARKED BY ";JB REV A-0", ARE:
               - SET THE  ODT BREAK  VECTOR (LOCATION 140) TO THE STARTING ADDRESS OF
                 FALCON'S ODT ROM (170000-OCTAL).
               - CHANGE PRIORITY FROM LEVEL 7 TO LEVEL 6 TO ALLOW THE BREAK KEY TO
                 INTERRUPT.
               - LOWERED RATE FROM 72KB TO 64KB IN TEST 06.

              REV B0:                          JAKI BERG                     9-JUL-84
              LOWERED RATE FROM 72KB TO 64KB IN TEST 07.  DEVELOPING ENGINEER IN ANNECY
              ADVISED THAT TEST 6 AND TEST 7 SHOULD BE RUN AT 64KB.

```

TABLE OF CONTENTS

1.0 INTRODUCTION

1.1 PROGRAM ABSTRACT

THIS DIAGNOSTIC WAS DESIGNED TO TEST OUT THE KMV11 MODULE

THE PROGRAM WAS IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR.

THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM WILL ALLOW MODIFICATION OF DEVICE PARAMETERS, SUCH AS UNIBUS ADDRESS, VECTOR ADDRESS, AND PROCESSOR TYPE.

1.2 HARDWARE INTRODUCTION

THIS DIAGNOSTIC WILL TEST ALL THE HARDWARE PART OF THE KMV11 A MODULE (M7500).
TO TEST COMPLETELY THIS PART ,EXTERNAL LOOP BACK CONNECTOR MUST BE INSTALLED.

EXTERNAL LOOP BACK CONNECTOR:

KMV11 A CAN OPERATE EITHER IN RS422 OR RS 423 LEVEL CONVERTERS

RS422 LOOP BACK:
TO TEST COMPLETELY A KMV11 B IN RS422 MODE ,RUN THIS DIAGNOSTIC WITH LOOP BACK CONNECTOR PLUG :
-USE H3255 TO LOOP DIRECTLY AT THE OUTPUT OF THE MODULE
-USE H3251 PLUG AT THE END OF BC55U MODEM CABLE CONNECTOR ASSY.

RS423 LOOP BACK:
TO TEST COMPLETELY A KMV11-A IN RS423 MODE ,RUN THIS DIAGNOSTIC WITH LOOP BACK CONNECTOR PLUG :
-USE H3255 TO LOOP AT THE OUTPUT OF THE MODULE
-USE H3251 PLUG AT THE END OF BC55H MODEM CABLE CONNECTOR ASSY.

RS232 LOOP BACK:
SAME AS FOR RS423.

CAUTION:
USE OF H325 LOOP BACK CONNECTOR WILL CAUSE MESSAGES ERROR IN TEST 8.

```
DIAGNOSTIC WILL TEST KMV11 CLOCKS,LINE INTERRUPTS,TX AND RX FUNCTION
IN INTERNAL AND EXTERNAL LOOP BACK AND MODEM SIGNALS.



CAUTION:
*******
AT THE BEGINNING OF THE DIAGNOSTIC THE OPERATOR WILL ANSWER
BY "YES " OR "NO" AT THE QUESTION:
IS EXTERNAL CONNECTOR PLUGGED?


IF CONNECTOR NO PLUGGED THE DIAGNOSTIC WILL REPORT AN ERROR
AND EXIT CORRESPONDING TEST.

KMV11 A IS FULLY TESTED ONLY WHEN DIAGNOSTIC HAS BEEN RUN
SUCCESSFULLY IN BOTH RS422 AND RS423 LOOP BACK.


2.0 HARDWARE REQUIREMENTS

THE  FOLLOWING  HARDWARE  IS REQUIRED TO RUN THE KMV11 A
LINE CONTROLLER STATIC TESTS:

SBC-11/21+
16K MEMORY
CONSOLE TERMINAL


3.0 PRELIMINARY PROGRAM REQUIREMENTS

THE PROCESSOR AND MEMORY SHOULD BE THOROUGHLY  TESTED  PRIOR
TO RUNNING THIS DIAGNOSTIC.

*********************************************************************
*                                                                   *
*       NOTE: THE KMV11 DIAGNOSTICS NKMDA AND NKMBA SHOULD BE       *
*       BEFORE RUNNING NKMCA.                                       *
*                                                                   *
*********************************************************************

4.0 GENERAL PROGRAM CONSIDERATIONS


4.1 DIAGNOSTIC SUPERVISOR

THIS  PROGRAM  IS  COMPATIBLE WITH THE STANDALONE DIAGNOSTIC
SUPERVISOR, AND MUST BE LOADED TO BE  CO-RESIDENT  WITH  THE
SUPERVISOR,  OR  BE  PREVIOUSLY COMBINED WITH THE SUPERVISOR
```

AND LOADED AS A SINGLE FILE. IN EITHER CASE, THE COMBINED PROGRAM WILL NOT EXCEED 16K OF MEMORY.

4.2 EXECUTION TIME

THE TOTAL TIME REQUIRED TO RUN THE KMV11 LINE CNT DIAGNOSTIC IS ABOUT :

- 160 SECONDS FROM TEST 1 TO TEST 6 (TEST IN INTERNAL LOOP).
- 260 SECONDS FROM TEST 1 TO TEST 8 (COMPLETE TEST, WITH EXTERNAL CONNECTOR).

4.3 XXDP+

THIS PROGRAM MAY BE LOADED UNDER XXDP+, AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.4 ACT/SLIDE

THIS PROGRAM MAY BE LOADED UNDER ACT OR SLIDE AND MAY BE RUN IN DUMP MODE OR CHAIN MODE.

4.5 APT

THIS PROGRAM MAY BE LOADED BY THE APT SYSTEM (INCLUDING APT-RD) AND RUN IN PROGRAM MODE OR SCRIPT MODE.

4.6 MEMORY MANAGEMENT

MEMORY MANAGEMENT IS NOT UTILIZED IN THIS PROGRAM. IF IT IS INSTALLED, IT IS DISABLED BY THE PROGRAM.

4.7 MEMORY PARITY OPTION

IF PARITY MEMORY IS INSTALLED, MEMORY PARITY TRAPS ARE DISABLED BY THE PROGRAM.

4.8 ERROR LOGGING

THE NUMBER OF ERRORS WHICH HAVE OCCURRED ON EACH DEVICE UNDER TEST SINCE THE LAST START OR RESTART COMMAND IS KEPT IN AN ERROR LOG. THIS LOG MAY BE PRINTED BY USING THE "PRINT" COMMAND (SEE SECTION 6.3.8).

5.0 PROGRAM LOAD MEDIA

```

                                THIS PROGRAM  CAN  BE  LOADED  FROM  PAPER  TAPE  USING  THE
                                ABSOLUTE  LOADER OR FROM ACT, SLIDE, OR APT SYSTEMS, OR FROM
```

ANY MEDIA SUPPORTED BY XXDP+. WHEN USING THE PAPER TAPE ABSOLUTE LOADER, THE PROGRAM SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC SUPERVISOR. WHEN USING XXDP+, THE DIAGNOSTIC SUPERVISOR SHOULD BE LOADED FIRST, FOLLOWED BY THE DIAGNOSTIC PROGRAM.

6.0 OPERATING INSTRUCTIONS

6.1 LOADING AND STARTING PROCEDURES

6.1.1 LOADING PROCEDURES

THIS PROGRAM MAY BE LOADED FROM PAPER TAPE USING THE ABSOLUTE LOADER. IT MAY ALSO BE LOADED FROM ANY XXDP+ LOAD MEDIA. WHEN LOADED UNDER XXDP+, THE DIAGNOSTIC SUPERVISOR WILL BE LOADED AUTOMATICALLY.

6.1.2 STARTING PROCEDURES

THE PROGRAM STARTS AT LOCATION 200. USE STANDARD DEC PROCEDURES TO START THE PROGRAM.

6.1.3 STEPS FOR QUICK AND SIMPLE EXECUTION

THE DIAGNOSTIC CAN BE EXECUTED STANDALONE UNDER XXDP+ WITHOUT READING THE REMAINDER OF THIS DOCUMENT, AS FOLLOWS:

A) LOAD AND START DIAGNOSTIC USING RUN COMMAND
B) RECEIVE DIAGNOSTIC SUPERVISOR PROMPT   (DR>)
C) ENTER STA<CR>
D) ANSWER HARDWARE AND SOFTWARE QUESTIONS
E) GET END OF PASS MESSAGES OR ERROR MESSAGES
F) TO END EXECUTION, ENTER CONTROL/C

6.2 INITIAL DIALOGUE

AFTER THE PROGRAM AND THE SUPERVISOR ARE LOADED AND THE PROGRAM IS STARTED , THE FOLLOWING IDENTIFICATION IS TYPED:

```
DRS LOADED
DIAG.  RUN-TIME SERVICES
NKMBAO
KMV11 A  LINE CONTROLLER DIAGNOSTIC
DR>
```

THE OPERATOR THEN PROCEEDS BY TYPING ONE OR MORE OF THE COMMANDS DESCRIBED IN THE FOLLOWING SECTION 6.3.(FOR MORE DETAILED INFORMATION, REFER TO THE DIAGNOSTIC SUPERVISOR FUNCTIONAL SPECIFICATION).

6.3 PROGRAM OPTIONS

6.3.1 START COMMAND

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

6.3.1.1 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.) THAT SPECIFY THE TESTS TO BE EXECUTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS RANGE FROM 1 TO THE LARGEST TEST NUMBER IN THE DIAGNOSTIC. THEY MAY BE SPECIFIED IN ANY ORDER. 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 6.3.1.5.

6.3.1.2 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) AGAINST ALL UNITS SUBMITTED. THE DEFAULT IS NON-ENDING EXECUTION. IN THIS CASE EXIT FROM THE PROGRAM IS ACCOMPLISHED EITHER BY TYPING A CONTROL/C OR BY OCCURRENCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

6.3.1.3 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) CONTAIN-
         ING THE ERROR
   IER   INHIBIT ERROR REPORTING
   IBE   INHIBIT BASIC ERROR REPORTS
   IXE   INHIBIT EXTENDED ERROR REPORTS
   PRI   DIRECT ALL MESSAGES TO A LINE PRINTER
   PNT   PRINT NUMBER OF TEST BEING EXECUTED
   BOE   BELL ON ERROR
   UAM   RUN IN UNATTENDED MODE, BYPASSING MANUAL
         INTERVENTION TESTS
   ISR   INHIBIT STATISTICAL REPORTS
   IDU   INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
   LOT   LOOP ON TEST

THE FLAGS NAMED OR EQUATED TO 1 ARE SET, THOSE EQUATED TO  0
ARE CLEARED.  A FLAG NOT SPECIFIED IS CLEARED.  IF THE FLAGS
SWITCH IS NOT GIVEN ALL FLAGS ARE CLEARED.  SEE  EXAMPLE  AT
END OF 6.3.1.5.


6.3.1.4 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 6.3.1.5.


6.3.1.5 EFFECT OF START COMMAND

THE EFFECT OF THE START COMMAND IS TO INITIATE THE  HARDWARE
PARAMETER  DIALOGUE,  THE  SOFTWARE  PARAMETER DIALOGUE, AND
THEN THE DIAGNOSTIC TESTS THEMSELVES.

THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE  QUESTION
"# UNITS?"  TO  WHICH  THE  OPERATOR REPLIES WITH A DECIMAL
NUMBER N FROM 1 TO 16. THE TERM "UNIT" REFERS TO THE  DEVICE
TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED.  FOLLOWING
THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES  WILL
BE  BUILT.  EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING
ALL THE HARDWARE INFORMATION FOR  ONE  UNIT.   THE  OPERATOR
MUST  SUPPLY  N  (NUMBER OF UNITS) VALUES FOR EACH QUESTION.
HE MAY DO THIS BY GIVING ONE ANSWER  TO  EACH  QUESTION  (IN
WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR
BY GIVING N VALUES, SEPARATED BY COMMAS,  TO  EACH  QUESTION
(SERIES  WILL  BE POSED ONCE).  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.
```

FOLLOWING THE HARDWARE QUESTIONS ARE THE SOFTWARE QUESTIONS TO BUILD THE SOFTWARE TABLES, WHICH DEFINE THE MODE (QUICK VERIFY ETC.) THAT THE DIAGNOSTIC WILL EXECUTE IN.

WHEN THE QUESTION "# UNITS?" IS ANSWERED, MEMORY STORAGE IS ALLOCATED FOR THE P-TABLES, AND IF THERE IS NOT ENOUGH TO ACCOMMODATE THEM THE MESSAGE "TOO MANY UNITS" IS ISSUED. IN THIS CASE THE DIAGNOSTIC MUST BE EXECUTED MORE THAN ONCE TO TEST ALL UNITS.

EXAMPLE:

```
STA/TESTS:1:2-4:6:8-10/PASS:3/FLAGS:IER:HOE=1:UAM:LOE
```

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. 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.

6.3.2 RESTART COMMAND

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

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

6.3.2.2 UNITS SWITCH (/UNITS:<UNIT-LIST>)

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

6.3.2.3 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 GIVES THE ABILITY TO SELECT A SUBSET OF THESE. 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 C) A CONTROL/C WAS ENTERED BY THE OPERATOR.

6.3.3 CONTINUE COMMAND

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

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

<PASS-CNT> IS SAME AS IN START COMMAND, BUT THE DEFAULT IS THE UNSATISFIED PASS-CNT FROM THE PREVIOUS START OR RESTART. IF NONE REMAINS, THE DEFAULT IS NON-ENDING EXECUTION.

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

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

6.3.3.3 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.

6.3.4 PROCEED COMMAND

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

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

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

6.3.4.2 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.

6.3.5 ADD COMMAND

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

6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.5.2 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.

6.3.6 DROP COMMAND

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

6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.6.2 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.

6.3.7 PRINT COMMAND

```
**************************************************************
PRI(NT)
**************************************************************
```

6.3.7.1 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.

6.3.8 DISPLAY COMMAND

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

6.3.8.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)

<UNIT-LIST> IS AS IN THE RESTART COMMAND.

6.3.8.2 EFFECT OF DISPLAY COMMAND

THE HARDWARE P-TABLES FOR ALL UNITS UNDER TEST ARE PRINTED OUT IN THE FORMAT IN WHICH THEY WERE ENTERED. ANY UNITS THAT WERE DROPPED BY THE OPERATOR "DROP" COMMAND ARE SO DESIGNATED.

6.3.9 FLAGS COMMAND

```
**************************************************************
FLA(GS)
**************************************************************
```

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

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

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

A CONTROL C (C) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES A RETURN TO COMMAND MODE.

A CONTROL Z (Z) ENTERED DURING ONE OF THE THREE OPERATOR DIALOGUES- INITIAL DIALOGUE (SEE 6.2),HARDWARE DIALOGUE (SEE 6.3.1.5), OR SOFTWARE DIALOGUE (SEE 6.3.1.5) CAUSES THE DEFAULTS TO BE TAKEN FOR THE REMAINDER OF THAT DIALOGUE.

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES ALL TELETYPE OUTPUT TO BE SUPPRESSED FOR THE REMAINDER OF THE DIAGNOSTIC OR UNTIL ANOTHER O IS TYPED, WHICH RESTORES NORMAL TELETYPE OUTPUT.

6.3.12 HARDWARE PARAMETERS

THE FOLLOWING QUESTIONS WILL BE ASKED ON A START COMMAND. THE VALUE LOCATED TO THE LEFT OF THE QUESTION MARK IS THE DEFAULT VALUE THAT WILL BE TAKEN ON A CARRIAGE RETURN RESPONSE.

2. MICRO-CPU CSR ADDRESS: (0) 177000?

THIS IS THE ADDRESS AT WHICH THE CSR REGISTERS (SELO) RESIDE ON THE UNIBUS. THE ALLOWABLE RANGE IS 160000-177776 (OCTAL), AND THE DEFAULT IS 177000.

3. MICRO CPU VECTOR ADDRESS: (0) 300?

THE ALLOWABLE RANGE IS 300-770,AND DEFAULT VALUE IS 300

4. MICRO CPU PRIORITY LEVEL: (4) 7?

```

DEFAULT VALUE IS 4
NOTE:
M7500 AND M7501 MODULE MOUNTED WITH DC003 CHIPS CAN ONLY
INTERRUPT ON LEVEL 4


5. IS LOOP BACK CONNECTOR PLUGGED? 0=NO,1=YES   (0) 1 ?

DEFAULT VALUE IS 1 (YES)
NOTE :
        REFER TO CHAPTER 1.2 FOR LOOP BACK CONNECTOR DESCRIPTION.


```

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED BY PART 2 OF THE STATIC LOGIC TESTS.

6.3.14 EXTENDED DISCUSSION OF P-TABLE DIALOGUE

THE FULL CAPABILITY OF THE HARDWARE DIALOGUE IS REVEALED BY THE FOLLOWING DISCUSSION OF WHAT HAPPENS INTERNALLY.

AS SOON AS THE QUESTION "# UNITS?" IS ANSWERED (WITH THE NUMBER N, SAY) SPACE IN CORE IS ALLOCATED FOR N P-TABLES. ALL OF THE P-TABLES ARE OF THE SAME FORMAT, AND THERE IS A ONE-TO ONE CORRESPONDENCE BETWEEN THE HARDWARE PARAMETER QUESTIONS AND THE SLOTS IN THE P-TABLE FORMAT.

ON THE FIRST TRIP THRU THE QUESTIONS, ALL OF THE SLOTS IN ALL OF THE P-TABLES ARE FILLED. IF THE OPERATOR TYPES IN LESS THAN N EXPLICIT VALUES IN RESPONSE TO A PARTICULAR QUESTION, THESE VALUES ARE PLACED IN THE P-TABLES (ONE VALUE GOING INTO THE PROPER SLOT OF EACH P-TABLE BEGINNING WITH THE FIRST P-TABLE) UNTIL THE STRING OF VALUES IS EXHAUSTED. THE LAST VALUE IN THE STRING BECOMES THE NEW DEFAULT AND IS USED TO FILL THAT SLOT IN THE REMAINING P-TABLES.

ON SUBSEQUENT TRIPS THRU THE QUESTIONS, THE SAME PROCESS IS CARRIED OUT, EXCEPT THAT THE EARLIEST P-TABLE NOT TO HAVE RECEIVED AN EXPLICIT VALUE IN ANY OF ITS SLOTS NOW ASSUMES THE ROLE THAT TABLE NUMBER ONE PLAYED IN THE FIRST TRIP.

THE SERIES OF QUESTIONS IS REISSUED UNTIL AT LEAST ONE QUESTION HAS RECEIVED N EXPLICIT VALUES FROM THE OPERATOR.

IN GIVING A STRING OF VALUES, COMMAS WITHOUT INTERVENING VALUES MAY BE USED TO INDICATE A REPETITION OF THE LAST NAMED VALUE.

A STRING OF VALUES MAY BE GIVEN AS A RANGE (6-10 FOR EXAMPLE). IF THE VALUES REPRESENT PURE NUMERICAL DATA, THIS SAMPLE RANGE TRANSLATES TO THE STRING 6,7,8,9,10 (AN INCREMENT OF 1). IF THE VALUES ARE ADDRESSES, THE SAMPLE RANGE TRANSLATES TO THE STRING 6,8,10 (AN INCREMENT OF 2).

NOW LET US SEE HOW WE COULD USE THESE CAPABILITIES TO CONSTRUCT A SET OF P-TABLES. ASSUME THAT WE HAVE 16 UNITS, AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE SLOTS IN THE P-TABLE, THREE HARDWARE QUESTIONS IN THE DIALOGUE). LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE THE NUMBER 75 FOR ALL 16 TABLES. LET THE DESIRED VALUE FOR THE SECOND PARAMETER BE EQUAL TO THE UNIT NUMBER (0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE THE VALUE 11. LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE NUMBER 77 FOR THE LAST 9 UNITS.

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
# UNITS (D) ? 16

UNIT 1
<QUESTION 1> ? 75
<QUESTION 2> ? 0-6
<QUESTION 3> ? 76

UNIT 21
<QUESTION 1> ?
<QUESTION 2> ? 7-11,,13-15
<QUESTION 3> ? 77
```

THE FIRST TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75 IN ALL 16 TABLES. SLOT TWO RECEIVES THE VALUES 0,1,2,...,6 IN TABLES 0 THRU 6 AND A CONSTANT 6 IN TABLES 7 THRU 15. SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.

THE SECOND TIME THRU THE SERIES, TABLES 16 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE THE OPERATOR IN THE FORM "UNIT XX" AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU 15, SINCE NO NEW EXPLICIT VALUES ARE TYPED IN. SLOT TWO GETS THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND GETS A 11 IN SLOT 12, AND GETS THE VALUES 13,14,15 IN TABLES 13 THRU 15. SLOT THREE GETS THE VALUE 77 IN TABLES 7 THRU 15.

THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT 16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION (NAMELY QUESTION 2).

```
7.0  TEST DESCRIPTIONS


******************** TEST 1 *********************
*VERIFY THAT REFERENCED UNIBUS DEVICE REGISTERS
*DOES NOT CAUSE TIME OUT TRAP
**************************************************


********************* TEST 2 *********************
*
*PROM REVISION TEST
*
**************************************************


********************** TEST 3 ********************
*
*REAL TIME CLOCK TEST
*
**************************************************


********************** TEST 4 ********************
*
*BAUD RATE GENERATOR TEST
*
**************************************************


********************** TEST 5 ********************
*
*TRANSMIT FRAMES AT LOW SPEED IN INTERNAL LOOP
*ON CHANNEL A  WITHOUT ANY INTERRUPT
*
**************************************************


********************** TEST 6 ********************
*
*TRANSMIT AND RECEIVE FRAMES IN INTERNAL LOOP AT
*DIFFERENT SPEED  WITH INTERRUPT
*
**************************************************
```

```
CAUTION:
TEST NUMBER 7 AND 8 LOOP BACK CONNECTOR MUST BE INSTALLED.
REFER TO CHAPTER 1.2 FOR LOOP BACK DESCRIPTION






********************** TEST 7 ******************
*
*TRANSMIT AND RECEIVE FRAMES IN EXTERNAL LOOP BACK
*(WITH EXTERNAL LOOP BACK)
*
*************************************************



********************** TEST 8 ******************
*
*TEST ALL MODEM SIGNAL IN EXTERNAL LOOP BACK
*
*************************************************
```

8.0 ERROR INFORMATION

8.1 ERROR REPORTING

ERRORS ARE REPORTED BY THE PROGRAM AS THEY OCCUR (IF NOT INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR ERROR REPORT FORMAT, AND CONSISTS OF A DESCRIPTION OF THE ERROR, THE TEST NUMBER, SUBTEST NUMBER, PC OF THE ERROR CALL, DEVICE ADDRESS, AND BASIC AND EXTENDED ERROR INFORMATION.

9.0 HISTORY

- DESIGN STARTED ON MAY 82
- REVIEW ON DECEMBER 82

@

```
  966                                      .TITLE KMV11 A LINE CNT DIAGNOSTIC
  974          002000                              .=2000
  975
  976
  977
  978
  979
  980
  981                                              .MCALL  SVC
  982 002000                                       SVC                                     ; INITIALIZE SUPERVISOR MACROS
  983
  984
  985
  986
  987
  988 002000                                       BGNMOD  KMV11A
  989
  990
  991          000000                      $LSTIN= 0
  992          000000                      $LSTTAG= 0
  993          177777                      SVCINS= -1       ; LIST INSTRUCTIONS, SHIFTED RIGHT
  994          177777                       SVCTS= -1       ; LIST TEST TAGS, SHIFTED RIGHT
  995          177777                      SVCSUB= -1       ; LIST SUBTEST TAGS, SHIFTED RIGHT
  996          177777                      SVCGBL= -1       ; LIST GLOBAL TAGS, SHIFTED RIGHT
  997          177777                      SVCTAG= -1       ; LIST OTHER TAGS, SHIFTED RIGHT
  998
  999                                      ;     CHANGE THE VALUES OF THE SVC... SYMBOLS TO BE ZERO IF YOU WISH
 1000                                      ;     TO ALIGN THE MACRO CALLS AND THEIR EXPANSIONS.  CHANGE THE
 1001                                      ;     SYMBOLS TO BE MINUS-ONE TO NOT LIST THE EXPANSIONS.  YOU MAY
 1002                                      ;     CHANGE THE SYMBOLS AT ANY POINT IN YOUR PROGRAM.
 1003
 1004
```

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

002000                                    POINTER BGNSW,BGNDU,BGNSETUP




002000                                    HEADER  NKMBA0,A,0,240.,0

```

```

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

002122                                            BGNPROT

002122  000000                                    0               ;OFFSET INTO P-TABLE FOR CSR ADDRESS
002124  177777                                    -1              ;OFFSET INTO P-TABLE FOR MASSBUS ADDRESS
002126  177777                                    -1              ;OFFSET INTO P-TABLE FOR DRIVE NUMBER




002130                                            ENDPROT

```

```
                                .SBTTL  DISPATCH TABLE

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

002130                                  DISPATCH 8
```

```
                                .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 RUN-TIME P-TABLE.
                                ;/ AND IS USED AS A " TEMPLATE" FOR BUILDING THE P-TABLE
                                ;//////////////////////////////////////////////////////////////////////////////

                                .ENABL  AMA
002152                                  BGNHW   DFPTBL



002154    177000                        .WORD   177000          ;KMV11,CSRS ADDRESS
002156    000300                        .WORD   300             ;KMV11, VECTOR ADDRESS
002160    004000                        .WORD   4000            ;INTERRUPT PRIORITY LEVEL
002162    000001                        .WORD   1               ;LOOP BACK CONNECTOR?
002164                                  ENDHW
```

```









                                .SBTTL  GLOBAL EQUATES SECTION

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






002164                                  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
                                ;
                                ;                               ; BIT POSITION IN SECOND STATUS WORD
```

```
             000040                     EF.START==        32.              ; (100000) START COMMAND WAS ISSUED
             000037                     EF.RESTART==      31.              ; (040000) RESTART COMMAND WAS ISSUED
             000036                     EF.CONTINUE==     30.              ; (020000) CONTINUE COMMAND WAS ISSUED
             000035                     EF.NEW==          29.              ; (010000) A NEW PASS HAS BEEN STARTED
             000034                     EF.PWR==          28.              ; (004000) A POWER-FAIL/POWER-UP OCCURRED
                                        ;
                                        ;
                                        ; PRIORITY LEVEL DEFINITIONS
                                        ;
             000340                     PRI07== 340
             000300                     PRI06== 300
             000240                     PRI05== 240
             000200                     PRI04== 200
             000140                     PRI03== 140
             000100                     PRI02== 100
             000040                     PRI01== 40
             000000                     PRI00== 0
                                        ;
                                        ;OPERATOR FLAG BITS
                                        ;
             000004                     EVL==        4
             000010                     LOT==       10
             000020                     ADR==       20
             000040                     IDU==       40
             000100                     ISR==      100
             000200                     UAM==      200
             000400                     BOE==      400
             001000                     PNT==     1000
             002000                     PRI==     2000
             004000                     IXE==     4000
             010000                     IBE==    10000
             020000                     IER==    20000
             040000                     LOE==    40000
             100000                     HOE==   100000
   1170
   1171
   1172                                 ;MAXPRI==340                                                     ;JB REV A-0
   1173      000300                     MAXPRI==300                                                      ;JB REV A-0
   1174      054000                     MAINT0==54000          ;MASTER CLEAR = 1,MODE = 1 ,MAINT 1 = 1 ,T11=HOLD
   1175      044000                     MAINT1==44000          ;MASTER CLEAR = 1,MODE = 0 ,MAINT 1 = 0 ,T11=NOT HOLD
   1176      040000                     MCLR==40000
   1177      052525                     DATA1== 052525
   1178      125252                     DATA2== 125252
   1179      013224                     KB1.2== 5780.                                  ;OCTAL VALUE OF 1.2 KBAUDS
   1180      000154                     KB64==  108.                                   ;  "        "   " 64    "
   1181      000174                     KB56==  124.                                   ;  "        "     56    "
   1182      000146                     KB68==  102.                                   ;  "        "     68    "
   1183      000141                     KB72==  97.                                    ;  "        "     72    "
   1184
   1185                                                                                ;DIVIDER CALCULATION
   1186                                                                                ;DECIMAL VAUE = 6912:YYY KBAUDS
   1187                                 ;;**************************************************************************
   1188                                 ;* PROGRAM EVENT FLAG DEFINITIONS
   1189                                 ;;**************************************************************************
   1190
   1191
   1192
```

```
                                .SBTTL  GLOBAL DATA SECTION

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




                                ;;*********************************************************************
                                ;* STORAGE FOR DEVICE REGISTERS
                                ;;*********************************************************************
002164                                  DESCRIPT        <KMV11A LINE CNT DIAGNOSTIC>




002220                                  ERRTBL
002220  000000                  ERRTYP::        .WORD   0
002222  000000                  ERRNBR::        .WORD   0
002224  000000                  ERRMSG::        .WORD   0
002226  000000                  ERRBLK::        .WORD   0




                                ;*****************************************************************
                                ;* PROGRAM CONTROL PARAMETERS
                                ;*****************************************************************
002230  000000                  LOCK:   .WORD   0               ;ADDRESS FOR LOCK CURRENT DATA
002232  000005                  MAXERR: .WORD   5               ;MAX ERROR BEFORE DROPPING THE UNIT
002234  000000                  ERRCNT: .WORD   0               ;ERROR COUNT
002236  000000                  L$SW:   .WORD   0
002240  000015                  L$UIT:  .WORD   15                      ;MAX LINE UNIT
002242  000000                  LOGDEV: .WORD   0
002244  000000                  SAVPC:  .WORD   0
002246  000000                  PSTACK: .WORD   0
002250  000000                  FTIME:  .WORD   0
```

```
 1246                                  ;*******************************************************************
 1247                                  ;* MISCELLANEOUS STORAGE
 1248                                  ;*******************************************************************
 1249 002252  000000                   SAVE4:   .WORD   0
 1250 002254  000000                   SAVE6:   .WORD   0
 1251 002256  000000                   FLAG:    .WORD   0
 1252
 1253 002260  000000                   DELCT1: .WORD    0
 1254 002262  000000                   DELCT2: .WORD    0
 1255 002264  000000                   GOOD:    .WORD   0
 1256 002266  000000                   GOOD0:   .WORD   0
 1257 002270  000000                   GOOD1:   .WORD   0
 1258 002272  000000                   GOOD2:   .WORD   0
 1259 002274  000000                   GOOD4:   .WORD   0
 1260 002276  000000                   GOOD6:   .WORD   0
 1261 002300  000000                   GOOD10: .WORD    0
 1262 002302  000000                   GOOD12: .WORD    0
 1263 002304  000000                   GOOD14: .WORD    0
 1264 002306  000000                   GOOD16: .WORD    0
 1265 002310  000000                   SEL0:    .WORD   0
 1266 002312  000000                   SEL1:    .WORD   0
 1267 002314  000000                   SEL2:    .WORD   0
 1268 002316  000000                   SEL4:    .WORD   0
 1269 002320  000000                   SEL6:    .WORD   0
 1270 002322  000000                   SEL10:   .WORD   0
 1271 002324  000000                   SEL12:   .WORD   0
 1272 002326  000000                   SEL14:   .WORD   0
 1273 002330  000000                   SEL16:   .WORD   0
 1274 002332  000000                   BSEL1:   .WORD   0
 1275 002334  000000                   RANST:   .WORD   0
 1276 002336  000000                   RANSEL: .WORD    0
 1277 002340  000000                   RANMTA: .WORD    0
 1278 002342  000000                   RANDN:   .WORD   0
 1279 002344  000000                   SAVPC1: .WORD    0
 1280 002346  000000                   SAVSTA: .WORD    0
 1281 002350  000000                   COUNT:   .WORD   0
 1282 002352  000000                   NUMBER: .WORD    0
 1283 002354  000000                   ADDR:    .WORD   0
 1284 002356  000000                   GDDAT:   .WORD   0
 1285 002360  000000                   BDDAT:   .WORD   0
 1286
 1287 002362                           TTABLE: .BLKW    2000
 1288 006362                           RTABLE: .BLKW    2000
 1289
 1290 012362  000000                   EXADDR: .WORD    0
 1291 012364  000000                   INTFLG: .WORD    0
 1292 012366  000000                   BAD:     .WORD   0
 1293 012370  000000                   BSEL0:   .WORD   0
 1294 012372  000000                   DATA:    .WORD   0
 1295 012374  000000                   VECT:    .WORD   0
 1296
 1297
 1298 012376  000000                   KIND:    .WORD   0                         ;=0 IF KMV11A ,=1 IF KMV11B
 1299 012400  000000                   CHANEL: .WORD    0
 1300
 1301 012402  000000                   TXDATA: .WORD    0
 1302 012404  000000                   RXDATA: .WORD    0
```

```
   1303 012406   000000              TSPEED: .WORD    0
   1304 012410   000000              LENGTH: .WORD    0
   1305 012412   000000              NUB:    .WORD    0
   1306 012414   000000              RXCNT:  .WORD    0
   1307 012416   000000              STAERR: .WORD    0
   1308 012420   000000              WRDCNT: .WORD    0
   1309 012422   000000              UNIT:   .WORD    0
```

```

                                ;******************************************************************
                                ;LOAD IN LOCATION "GDREV" THE PROM VERSION NUMBER THAT IS        *
                                ;COMPATIBLE WHITH THIS DIAGNOSTIC                                 *
                                ;                                                                 *
                                ;EACH PROM CONTAIN A REV LEVEL AND A ECO LEVEL:                   *
                                ;THE REV LEVEL IS MODIFIED EACH TIME A MODIFICATION IS DONE       *
                                ;THE ECO LEVEL IS MODIFIED WHEN THE PROM MODIFICATION NEED        *
                                ;A DIAGNOSTIC MODIFICATION                                        *
                                ;******************************************************************





012424  000001                  GDREV:  .WORD   1







```

```
                                  ;*******************************************************************
                                  ;* PROGRAM CONTROL FLAGS
                                  ;*******************************************************************
012426      000                   INIFLG: .BYTE   0               ;PROGRAM INITIALIZING FLAG
                                          .EVEN
012430      000                   LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
012431      000                   QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG
                                          .EVEN
012432   000000                   UUT:    .WORD   0               ;CURRENT UNIT UNDER TEST






                                  ;*******************************************************************
                                  ;* POINTERS TO KMV11 VECTORS AND REGISTERS
                                  ;*******************************************************************
012434   000000                   KMVV00: 0       ;POINTER TO KMV11  INTRPT VECTOR 0
012436   000000                   KMVLVL: 0       ;POINTER TO KMV11  INTRPT SERVICE
012440   000000                   KMVV04: 0       ;POINTER TO KMV11  INTRPT VECTOR 04
012442   000000                   KMVV02: 0       ;    "       "        "        "    02
012444   000000                   KMVV06: 0       ;    "       "        "        :   06
012446   000000                   KMTLVL: 0       ;POINTER TO KMV11 TX INTRPT SERVICE PS
012450   000000                   KMVCSR: 0       ;POINTER TO KMV11 CONTROL STATUS REGISTER
012452   000000                   KMVP02: 0       ;POINTER TO KMV11 PORT REGISTER - SEL2
012454   000000                   KMVP04: 0       ;POINTER TO KMV11 PORT REGISTER - SEL4
012456   000000                   KMVP06: 0       ;POINTER TO KMV11 PORT REGISTER - SEL6

012460   000000                   KMVP10: 0       ;POINTER TO KMV11 PORT REG  -SEL10
012462   000000                   KMVP12: 0       ;POINTER TO PORT REG  -SEL 14
012464   000000                   KMVP14: 0       ;POINTER TO PORT REG  -SEL14
012466   000000                   KMVP16: 0       ;POINTER TO PORT REG 16

012470   000000                   LOOP:   0       ;POINTER TO LOOP BACK CONNECTOR
```

```
   1371                                   ;****  PRIMARY REG ADRS STORAGE FOR THIS UNIT  ******
   1372                                   ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
   1373 012472                            REGADR:
   1374
   1375                                   ;;****  STACK USED FOR SUBROUTINE LINKAGE  *****
   1376 012472                                    .BLKW   100
   1377 012672                            SSTACK:
```

```

                        .SBTTL  GLOBAL TEXT SECTION

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

                        ;;****************************************************************************
                        ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
                        ;;****************************************************************************
012672                          DEVTYP  <KMV11A>


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

```
                                        .SBTTL  GLOBAL SUBROUTINES

                                        ;----------------------------------------------------------------------------
                                        ;MACRO'S NEEDED TO CALL SUBROUTINES
                                        ;----------------------------------------------------------------------------



                                        .MACRO  CLRMAR
                                                ROMCLK
                                                004000
                                        .ENDM   CLRMAR




                                        ;/////////////////////////////////////////////////////////////////////////////////
                                        ;/       THE GLOBAL SUBROUTINES ARE CALLED BY MORE THAN ONE TEST
                                        ;/////////////////////////////////////////////////////////////////////////////////




                                         ;ROUTINE TO WAIT FOR EVENT OR TIMEOUT



                                        ;CALLING SEQUENCE:       JSR   PC,WAIT1
                                        ;                        JSR   PC,WAIT2


                                        ;INPUTS PARAMETERS:      DELCT1,DELCT2


                                        ;                        INC DELCT1 UNTIL 0
                                        ;                        DEC DELCT2 UNTIL 0      DELCT2= NUMB OF WAIT1 PASSES







012702  005237  002260                  WAIT2:  INC     DELCT1
012706  001375                                  BNE     WAIT2

012710                                          BREAK


012712  005337  002262                          DEC     DELCT2
012716  001371                                  BNE     WAIT2

012720  000207                                  RTS     PC
```

GLOBAL SUBROUTINES

```
    1467
    1468
    1469 012722  005237  002260          WAIT1:  INC     DELCT1
    1470 012726  001375                          BNE     WAIT1
    1471
    1472 012730  000207                          RTS     PC
```

```
;MACRO TO WAIT A FEW MS


;CALLING SEQUENCE:       WAITA    X                  0<X<177777
;                        WAITB    X,Y                0<X OR Y<177777



.MACRO  WAITA   X
        MOV     #X,DELCT1                   ;LOAD COUNT
        JSR     PC,WAIT1                    ;WAIT
.ENDM







.MACRO  WAITB   X,Y
        MOV     #X,DELCT1
        MOV     #Y,DELCT2
        JSR     PC,WAIT2
.ENDM
```

```
                                        ;ROUTINE TO DROP UNIT AFTER 5 ERROR


                                        ;JSR    PC,CHKMAX








012732                          CHKMAX: INLOOP                                  ;LOOPING ON ERROR?
012734                                  BCOMPLETE       1$                      ;IF YES, EXIT


012736                                  RFLAGS  R0                              ;GET OPERATOR FLAG
012740  032700  000040                  BIT     #IDU,R0                         ;IS DROPPING INHIBITTED?
012744  001026                          BNE     1$                              ;IF YES EXIT


012746  005237  002234                  INC     ERRCNT                          ;UPDATE ERROR COUNT
012752  023737  002234  002232          CMP     ERRCNT,MAXERR                   ;TOO MANY ERROR?
012760  003420                          BLE     1$                              ;IF NOT JUMP


012762                                  PRINTF  #NERRS,MAXERR,UUT               ;TOO MANY ERROR!
013012                                  DODU    UUT                             ;DROP UNIT

013020                                  DOCLN                                   ;END THE SUBPASS

013022  000207                  1$:     RTS     PC




                                        .NLIST  BEX
013024     045     116     045  NERRS:  .ASCIZ  /%N%AMORE THAN %D3%A  ERRORS ON UNIT %D2/
                                        .LIST   BEX
                                        .EVEN
                                        ;ROUTINE TO CHECK REGISTER BSEL0 AND TO REPORT ERROR







                                        ;CALLING SEQUENCE:      JSR     PC,TSTERR



                                        ;OUTPUT PARAMETERS:     RETURN TO       PC      IF TEST IS OK
                                        ;                                       PC+2    IF TIMEOUT DURING TEST
                                        ;                                       PC+4    IF NO KMV11 ANSWER
                                        ;                                       PC+6    IF DATA CMP ERROR






013074  004537  013644                  TSTERR: JSR     R5,CBSEL0                       ;LOOK IF BSEL0=0
013100  000000                                  .WORD   0
013102  000411                                  BR      1$                              ;TEST IS OK ,RTS PC


013104  004537  013644                          JSR     R5,CBSEL0                       ;LOOK IF BSEL0=200
013110  000200                                  .WORD   200
013112  000406                                  BR      2$                              ;TIMEOUT DURING TEST,RTS PC+2


013114  004537  013644                          JSR     R5,CBSEL0                       ;LOOK IF BSEL0=100
013120  000100                                  .WORD   100
013122  000405                                  BR      3$                              ;DATA CMP ERROR,RTS PC+6



013124  000407                                  BR      4$                              ;NO KMV11 ANSWER ,RTS PC+4



013126  000207                          1$:     RTS     PC                              ;TEST OK


013130  062716  000002                  2$:     ADD     #2,(SP)
013134  000207                                  RTS     PC                              ;TIMEOUT ERROR


013136  062716  000006                  3$:     ADD     #6,(SP)
013142  000207                                  RTS     PC                              ;DATA CMP ERROR


013144  062716  000004                  4$:     ADD     #4,(SP)
013150  000207                                  RTS     PC                              ;NO KMV11 ANSWER

```

```
        .SBTTL  NUMBER GENERATOR
;
;
;
;       DESCRIPTION:
;
;               ROUTINE TO GENERATE DATA PATTERNS,
;       THE TYPE OF PATTERN IS SELECTED BY R3, AND THE
;       PATTERN GENERATED IS RETURNED IN LOCATION "DATA"
;       AND LOCATION "GOOD"
;
;       CALLING SEQUENCE:
;
;               JSR     PC,GENER
;
;       INPUT PARAMETERS:
;
;       R3 CONTAINS THE PATTERN NUMBER
;
;       R3=0            ALL ZEROES
;       1               ALL ONES
;       2               010101 ETC BIT PATTERN
;       3               101010 ETC BIT PATTERN
;       4               ROTATING 1 IN A ZERO WORD
;       5               ROTATING 0 IN AN ALL ONE WORD
;       6               PSEUDO RANDOM NUMBER
;       7               INCREMENTING DATA PATTERN, GOOD
;                       CONTAINS THE VALUE TO BE UPDATED
;
;
;
;
;
;       IMPLICIT INPUT PARAMETERS.
;
;               NONE
;
;
;       OUTPUT PARAMETERS:
;
;               THE NUMBER GENERATED IS HELD IN
;       DATA AND GOOD.
;
;
;       IMPLICIT OUTPUT PARAMETERS:
;
;               NONE
;
;
;       COMPLETION CODES:
;
;               NONE
;
;
;       POSSIBLE ERROR CODES:
;
;               NONE
;
```

```
                            ;
                            ;
013152 042703 177770        GENER:  BIC     #177770,R3
013156 004737 013452                JSR     PC,SAVREG
013162 006303                       ASL     R3
013164 000173 013170                JMP     @GENSEL(R3)
013170 013210               GENSEL: GEN0                            ;ALL ZERO WORD
013172 013214                       GEN1                            ;ALL ONE WORD
013174 013222                       GEN52                           ;52 PATTERN
013176 013230                       GEN25                           ;25 PATTERN
013200 013236                       GENR1                           ;ROTATE '1' EACH CALL
013202 013246                       GENR0                           ;ROTATE '0' EACH CALL
013204 013304                       GENRAN                          ;RANDOM NUMBER
013206 013424                       GENINC                          ;INCREMENTING COUNT
013210 005000               GEN0:   CLR     R0                      ;0>R0
013212 000507                       BR      GENEX
013214 005000               GEN1:   CLR     R0                      ;NOT0>R0
013216 005100                       COM     R0
013220 000504                       BR      GENEX
013222 012700 052525        GEN52:  MOV     #52525,R0               ;5252>R0
013226 000501                       BR      GENEX
013230 012700 125252        GEN25:  MOV     #125252,R0              ;125252>R0
013234 000476                       BR      GENEX
013236 000241               GENR1:  CLC
013240 004737 013260                JSR     PC,GENROT               ;SHIFT 1 > R0
013244 000472                       BR      GENEX
013246 000241               GENR0:  CLC
013250 004737 013260                JSR     PC,GENROT               ;
013254 005100                       COM     R0                      ;SHIFT 0 > R0
013256 000465                       BR      GENEX
013260 006037 013302        GENROT: ROR     GENISH                  ;ROTATE  1 PATTERN
013264 001003                       BNE     GENER1                  ;= 0?
013266 012737 100000 013302         MOV     #100000,GENISH          ;YES, SET MSB
013274 013700 013302        GENER1: MOV     GENISH,R0               ;PUT 1 IN R0
013300 000207                       RTS     PC                      ;AND EXIT
013302 000001               GENISH: 1
013304 012737 000005 002336 GENRAN: MOV     #5,RANSEL               ;SET SELECT VALUE TO 5
013312 004737 013324                JSR     PC,RANGEN               ;GENERATE RANDOM NUMBER IN R0
013316 013700 002342                MOV     RANDN,R0                ;
013322 000443                       BR      GENEX                   ;
013324 013702 002342        RANGEN: MOV     RANDN,R2                ;
013330 001002                       BNE     RAN1                    ;IS RANDOM = 0
013332 013702 002334                MOV     RANST,R2                ;YES, PUT RANDOM START VALUE IN
013336 032737 000777 002336 RAN1:   BIT     #777,RANSEL             ;NO;IS RANSEL SELECT VALUE = 0
013344 001003                       BNE     RAN2                    ;NO
013346 012737 000001 002336         MOV     #1,RANSEL               ;YES: SET RANSEL = 1
013354 013703 002336        RAN2:   MOV     RANSEL,R3               ;
013360 013702 002342                MOV     RANDN,R2
013364 033702 002340                BIT     RANMTA,R2               ;GET R2 <0 AND 1>
013370 001405                       BEQ     RANCLC                  ;
013372 005102                       COM     R2
013374 033702 002340                BIT     RANMTA,R2
013400 001401                       BEQ     RANCLC
013402 000402                       BR      RANSEC
013404 000241               RANCLC: CLC
013406 000401                       BR      RAN4
013410 000261               RANSEC: SEC
```

```
013412  006037  002342          RAN4:   ROR     RANDN           ;ROTATE C TO B15
013416  005303                          DEC     R3              ;IS THIS NUMBER REQUIRED?
013420  001357                          BNE     RAN2+4          ;NO, GET ANOTHER
013422  000207                  RANEX:  RTS     PC              ;YES, EXIT
013424  013700  002264          GENINC: MOV     GOOD,R0         ;INCREMENTS LOC. 'GOOD'
013430  005200                          INC     R0
013432  010037  002264          GENEX:  MOV     R0,GOOD
013436  004737  013532                  JSR     PC,RSTREG
013442  013737  002264  012372          MOV     GOOD,DATA
013450  000207                          RTS     PC

```

```
                                         .SBTTL  SAVE REGISTERS
                              ;
                              ;
                              ;          DESCRIPTION:
                              ;
                              ;
                              ;                  ROUTINE TO SAVE ALL THE GENERAL PURPOSE
                              ;          REGISTERS ON THE STACK, AND LEAVE THE ADDRESS OF THE
                              ;          CALLING ROUTINE ON THE STACK. THE ROUTINE WILL RUN AT
                              ;          PRIORITY 6 TO AVOID MOST INTERRUPTS
                              ;
                              ;          CAUTION:REGISTER R0 IS NOT SAVED
                              ;
                              ;
                              ;
                              ;          CALLING SEQUENCE:
                              ;
                              ;                  JSR     PC,SAVREG
                              ;
                              ;          INPUT PARAMETERS:
                              ;
                              ;                  NONE
                              ;
                              ;
                              ;          IMPLICIT INPUT PARAMETERS:
                              ;
                              ;                  NONE
                              ;
                              ;
                              ;          OUTPUT PARAMETERS:
                              ;
                              ;                  REGISTERS 0 THRU 5 ARE SAVED ON THE STACK
                              ;          AND THE RETURN ADDRESS OF THE CALLING ROUTINE IS
                              ;          SET AS THE LAST ENTRY ON THE STACK
                              ;
                              ;
                              ;
                              ;          IMPLICIT OUTPUT PARAMETERS:
                              ;
                              ;                  NONE
                              ;
                              ;
                              ;          COMPLETION CODES:
                              ;
                              ;                  NONE
                              ;
                              ;
                              ;          POSSIBLE ERROR CODES:
                              ;
                              ;
                              ;                  NONE
                              ;
                              ;
 013452                       SAVREG:   GETPRI  SAVSTA
 013460                                 SETPRI  MAXPRI
 013466  012637  002244                 MOV     (SP)+,SAVPC      ;SAVE PC FOR RETURN FROM THIS ROUTINE
```

```
   1788 013472  012637  002344                 MOV     (SP)+,SAVPC1
   1789 013476  010546                         MOV     R5,-(SP)
   1790 013500  010446                         MOV     R4,-(SP)
   1791 013502  010346                         MOV     R3,-(SP)
   1792 013504  010246                         MOV     R2,-(SP)
   1793 013506  010146                         MOV     R1,-(SP)
   1794 013510  010046                         MOV     R0,-(SP)
   1795 013512  013746  002344                 MOV     SAVPC1,-(SP)
   1796 013516  013746  002244                 MOV     SAVPC,-(SP)      ;PUT PC READY FOR
   1797 013522                                 SETPRI  SAVSTA
   1798 013530  000207                         RTS     PC               ;RETURN
   1799
   1800
   1801
```

```
                                        .SBTTL  RESTORE REGISTERS
                                ;
                                ;
                                ;
                                ;       DESCRIPTION:
                                ;
                                ;               RESTORE TO RESTORE THE GENERAL PURPOSE
                                ;       REGISTERS. THE STACK IS LEFT IN THE SAME STATE AS IT
                                ;       WAS WHEN SAVREG WAS CALLED.
                                ;
                                ;       CAUTION:  REGISTER R0 IS NOT SAVED
                                ;
                                ;
                                ;
                                ;
                                ;       CALLING SEQUENCE:
                                ;
                                ;               JSR     PC,RSTREG
                                ;
                                ;
                                ;       INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       IMPLICIT INPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       OUTPUT PARAMETERS:
                                ;
                                ;               R1 THRU R5 RESTORED
                                ;
                                ;
                                ;       IMPLICIT OUTPUT PARAMETERS:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       COMPLETION CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;       POSSIBLE ERROR CODES:
                                ;
                                ;               NONE
                                ;
                                ;
                                ;
013532                          RSTREG: GETPRI  SAVSTA
013540                                  SETPRI  MAXPRI
013546  012637  002244                  MOV     (SP)+,SAVPC
013552  012637  002344                  MOV     (SP)+,SAVPC1
013556  012600                          MOV     (SP)+,R0
013560  012601                          MOV     (SP)+,R1
```

RESTORE REGISTERS

```
    1860 013562  012602                       MOV     (SP)+,R2
    1861 013564  012603                       MOV     (SP)+,R3
    1862 013566  012604                       MOV     (SP)+,R4
    1863 013570  012605                       MOV     (SP)+,R5
    1864 013572  013746  002344               MOV     SAVPC1,-(SP)
    1865 013576  013746  002244               MOV     SAVPC,-(SP)      ;PUT PC READY FOR
    1866 013602                               SETPRI  SAVSTA
    1867 013610  000207                       RTS     PC
```

```
                                        ;CHECK CONTENT OF ONE OF THE 8 REGISTERS

                                        ; CALLING SEQUENCE
                                        ;       JSR     R5,CKSELN               ; N = REGISTER NUMBER
                                        ;       .WORD A                 A=EXPECTED CONTENT OF REGISTER N

                                        ;OUTPUT PARAMETER:
                                        ;       BRANCH IN PC+2 IF ERROR DETECTED
                                        ;       BRANCH IN PC IF NO ERROR DETECTED





013612  012537  002264                  CKSEL0: MOV     (R5)+,GOOD      ;WRITE GOOD
013616  017737  176626  002310                  MOV     @KMVCSR,SEL0    ;READ SEL 0
013624  023737  002310  002264                  CMP     SEL0,GOOD       ;CMP ?
013632  001001                                  BNE     1$
013634  000402                                  BR      2$
013636  062705  000002                  1$:     ADD     #2,R5
013642  000205                          2$:     RTS     R5







013644  005037  002264                  CBSEL0: CLR     GOOD
013650  012537  002264                          MOV     (R5)+,GOOD
013654  117737  176570  012370                  MOVB    @KMVCSR,BSEL0
013662  123737  012370  002264                  CMPB    BSEL0,GOOD
013670  001001                                  BNE     1$
013672  000402                                  BR      2$
013674  062705  000002                  1$:     ADD     #2,R5
013700  000205                          2$:     RTS     R5
```

```
                                        ;ROUTINE TO CHECK ALL REGISTER FROM SEL0 TO SEL16


                                        ;CALLING SEQUENCE:
                                        ;       JSR R5,CKALL
                                        ;       .WORD A                 A = EXPECTED VALUE FOR SEL0
                                        ;       .WORD B                 B       "       "     SEL2
                                        ;       .WORD C                 C       "       "     SEL4
                                        ;       .WORD D                 D       "       "     SEL6
                                        ;       .WORD E                 E       "       "     SEL10
                                        ;       .WORD F                 F       "       "     SEL12
                                        ;       .WORD G                 G       "       "     SEL14
                                        ;       .WORD H                 H       "       "     SEL16


                                        ;OUTPUT PARAMETER:
                                        ;       BRANCH IN PC+2 IF ERROR
                                        ;       BRANCH IN PC IF NO ERROR



013702  012537  002266                  CKALL:  MOV     (R5)+,GOOD0
013706  012537  002272                          MOV     (R5)+,GOOD2
013712  012537  002274                          MOV     (R5)+,GOOD4
013716  012537  002276                          MOV     (R5)+,GOOD6
013722  012537  002300                          MOV     (R5)+,GOOD10
013726  012537  002302                          MOV     (R5)+,GOOD12
013732  012537  002304                          MOV     (R5)+,GOOD14
013736  012537  002306                          MOV     (R5)+,GOOD16

013742  017737  176502  002310                  MOV     @KMVCSR,SEL0            ;READ SEL0
013750  000240                                  NOP
013752  017737  176474  002314                  MOV     @KMVP02,SEL2            ;READ SEL2
013760  000240                                  NOP
013762  017737  176466  002316                  MOV     @KMVP04,SEL4            ;READ SEL4
013770  000240                                  NOP
013772  017737  176460  002320                  MOV     @KMVP06,SEL6            ;READ SEL6
014000  000240                                  NOP
014002  017737  176452  002322                  MOV     @KMVP10,SEL10           ;READ SEL10
014010  000240                                  NOP
014012  017737  176444  002324                  MOV     @KMVP12,SEL12           ;READ SEL12
014020  000240                                  NOP
014022  017737  176436  002326                  MOV     @KMVP14,SEL14           ;READ SEL14
014030  000240                                  NOP
014032  017737  176430  002330                  MOV     @KMVP16,SEL16           ;READ SEL16

014040  023737  002310  002266                  CMP     SEL0,GOOD0
014046  001035                                  BNE     1$
014050  023737  002314  002272                  CMP     SEL2,GOOD2
014056  001031                                  BNE     1$
014060  023737  002316  002274                  CMP     SEL4,GOOD4
014066  001025                                  BNE     1$
014070  023737  002320  002276                  CMP     SEL6,GOOD6
014076  001021                                  BNE     1$
014100  023737  002322  002300                  CMP     SEL10,GOOD10
014106  001015                                  BNE     1$
014110  023737  002324  002302                  CMP     SEL12,GOOD12
```

```
   1963 014116  001011                                BNE     1$
   1964 014120  023737  002326  002304                CMP     SEL14,GOOD14
   1965 014126  001005                                BNE     1$
   1966 014130  023737  002330  002306                CMP     SEL16,GOOD16
   1967 014136  001001                                BNE     1$
   1968
   1969 014140  000402                                BR      2$
   1970 014142  062705  000002               1$:      ADD     #2,R5
   1971 014146  000205                       2$:      RTS     R5
```

```
                                             ;ROUTINE TO CHECK SEL2 TO SEL16





014150  012537  002272                      CKREG:  MOV     (R5)+,GOOD2
014154  012537  002274                              MOV     (R5)+,GOOD4
014160  012537  002276                              MOV     (R5)+,GOOD6
014164  012537  002300                              MOV     (R5)+,GOOD10
014170  012537  002302                              MOV     (R5)+,GOOD12
014174  012537  002304                              MOV     (R5)+,GOOD14
014200  012537  002306                              MOV     (R5)+,GOOD16


014204  017737  176242  002314                      MOV     @KMVP02,SEL2
014212  000240                                      NOP
014214  017737  176234  002316                      MOV     @KMVP04,SEL4
014222  000240                                      NOP
014224  017737  176226  002320                      MOV     @KMVP06,SEL6
014232  000240                                      NOP
014234  017737  176220  002322                      MJV     @KMVP10,SEL10
014242  000240                                      NOP
014244  017737  176212  002324                      MOV     @KMVP12,SEL12
014252  000240                                      NOP
014254  017737  176204  002326                      MOV     @KMVP14,SEL14
014262  000240                                      NOP
014264  017737  176176  002330                      MOV     @KMVP16,SEL16




014272  023737  002314  002272                      CMP     SEL2,GOOD2
014300  001031                                      BNE     1$
014302  023737  002316  002274                      CMP     SEL4,GOOD4
014310  001025                                      BNE     1$
014312  023737  002320  002276                      CMP     SEL6,GOOD6
014320  001021                                      BNE     1$
014322  023737  002322  002300                      CMP     SEL10,GOOD10
014330  001015                                      BNE     1$
014332  023737  002324  002302                      CMP     SEL12,GOOD12
014340  001011                                      BNE     1$
014342  023737  002326  002304                      CMP     SEL14,GOOD14
014350  001005                                      BNE     1$
014352  023737  002330  002306                      CMP     SEL16,GOOD16
014360  001001                                      BNE     1$
014362  000402                                      BR      2$

014364  062705  000002                      1$:     ADD     #2,R5
014370  000205                              2$:     RTS     R5
```

```
                                ;ROUTINE TO CLEAR KMV11 MODULE


                                ;CALLING SEQUENCE:
                                ;       JSR PC,CLRKMV

                                ;ROUTINE DESCRIPTION: CLEAR ALL CSR'S REGISTERS AND CHECK IF = 0



014372  005077  176052          CLRKMV: CLR     @KMVCSR
014376  012777  054000  176044          MOV     #MAINTO,@KMVCSR
014404                                  WAITA   0




014416  012702  000010                  MOV     #10,R2
014422  013701  012450                  MOV     KMVCSR,R1               ;LOAD ADDRESS
014426  005021                  1$:     CLR     (R1)+                   ;CLEAR
014430  005302                          DEC     R2                      ;ALL DONE
014432  001375                          BNE     1$                      ;NO
014434  004537  013702                  JSR     R5,CKALL                ;CHECK ALL REG = 0
014440  000000                          .WORD   0
014442  000000                          .WORD   0
014444  000000                          .WORD   0
014446  000000                          .WORD   0
014450  000000                          .WORD   0
014452  000000                          .WORD   0
014454  000000                          .WORD   0
014456  000000                          .WORD   0
014460  000404                          BR      2$                      ;OK BRANCH AT END
014462                                  ERRHRD 1,EM0002,PRALL           ;CSR'S REGISTERS CAN'T BE CLEARED
014472  000207                  2$:     RTS     PC

```

```
                                        ;ROUTINE TO SET MAINT MODE 1 AND CHECK DCT11 CLEAR SELO AFTER HAVING DECODED



                                        ;CALLING SEQUENCE:
                                        ;       JSR PC,MAINM1



                                        ;GIVE AN ERROR IF MASTER CLEAR IS NOT CLEAR BY DCT11
                                        ;
                                        ;MAINT1= MASTER CLEAR=1 + MAINT 1 =0 + MODE = 1 : T11=HOLD






014474  005077  175750                  MAINM1: CLR     @KMVCSR
014500  000240                                  NOP
014502  000240                                  NOP
014504  000240                                  NOP

014506  012777  044000  175734                  MOV     #MAINT1,@KMVCSR         ;LOAD ADDRESS
014514  012737  000000  002260                  MOV     #0,DELCT1
014522  012737  000001  002262                  MOV     #1,DELCT2
014530  004737  012702                          JSR     PC,WAIT2
014534  004537  013612                          JSR     R5,CKSELO               ;CHECK SELO=0 BUT MODE BIT =1
014540  004000                                  .WORD   4000
014542  000404                                  BR      1$                      ;OK BRANCH
014544                                          ERRHRD  2,EM0001,PRSELO
014554  000207                          1$:     RTS     PC
```

```
                                        ;ROUTINE TO SET TEST NUMBER ON BSELO




                                        ;CALLING SEQUENCE:
                                        ;       JSR R5,TSTNUB
                                        ;       .WORD   A                       A=TEST MICRO PROGRAM NUMBER






 014556  012537  012412                 TSTNUB: MOV     (R5)+,NUB
 014562  053777  012412  175660                 BIS     NUB,@KMVCSR             ;LOAD TEST NUMBER
 014570  012737  000000  002260                 MOV     #0000,DELCT1
 014576  004737  012722                         JSR     PC,WAIT1                ;WAIT
 014602  000205                                 RTS     R5
```

```

                                        ;ROUTINE TO WRITE OR READ ONE OF THE KMV11 REGISTERS



                                        ;CALLING SEQUENCE:
                                        ;JSR      R5,WRITE
                                        ;.WORD    A                                       A=ADDRESS TO WRITE
                                        ;.WORD    B                                       B=DATA TO WRITE
                                        ;
                                        ;
                                        ;
                                        ;JSR      R5,READ
                                        ;.WORD    A                                       A=ADDRESS TO READ
                                        ;
                                        ;
                                        ;
                                        ;MICRO DIAG NB 47 DESCRIPTION:
                                        ;WRITE:   PUT ADDRESS TO WRITE IN SEL2
                                        ;         PUT DATA TO WRITE IN SEL4
                                        ;         SET BIT 0 OF SEL6(WRITE BIT)
                                        ;         SET TEST NB 44
                                        ;         KMV11 CLEAR BSEL0 WHEN DONE
                                        ;
                                        ;
                                        ;READ:    PUT ADDRESS TO READ IN SEL2
                                        ;         CLEAR BIT 0 IN SEL6
                                        ;         SET TEST 47
                                        ;         KMV11 READ ADDRESS IN SEL2 AND CLEAR BSEL0 WHEN DONE




014604    012577    175642              WRITE:    MOV       (R5)+,@KMVP02                 ;WRITE ADDRESS
014610    012577    175640                        MOV       (R5)+,@KMVP04                 ;  "   DATA
014614    012777    000001    175634              MOV       #1,@KMVP06                    ;BIT WRITE

014622    004537    014556                        JSR       R5,TSTNUB                     ;SEND TEST NB 44
014626    000047                                  .WORD     47

014630    000205                                  RTS       R5                            ;RETURN






014632    012577    175614              READ:     MOV       (R5)+,@KMVP02                 ;SET ADDRESS TO READ
014636    005077    175612                        CLR       @KMVP04
014642    005077    175610                        CLR       @KMVP06

014646    004537    014556                        JSR       R5,TSTNUB                     ;SEND TEST NB 44
014652    000047                                  .WORD     47


014654    004737    013074                        JSR       PC,TSTERR                     ;CHECK BSEL 0
014660    000412                                  BR        1$                            ;OK
```

```
014662  000402                           BR      2$
014664  000401                           BR      2$
014666  000400                           BR      2$

014670                          2$:      ERRHRD  4,EM0004              ;NO KMV ANSWER
014700  004737  012732                   JSR     PC,CHKMAX
014704  000205                           RTS     R5

014706  017737  175542  012366  1$:      MOV     @KMVP04,BAD           ;READ DATA IN BAD
014714  000205                           RTS     R5
```

```
.MACRO  ED$CALL XY
        .LIST
;********************** TEST'XY' ************************
        .NLIST
        .ENDM









.MACRO  BADHEAD
        .RADIX  10
        ED$CALL \T$TESTNUM+1
        .RADIX  8
        .ENDM


```

```
                                        .SBTTL  GLOBAL ERROR REPORT SECTION

                                        ;//////////////////////////////////////////////////////////////////////
                                        ;/        THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
                                        ;/        THAT ARE USED IN MORE THAN ONE TEST.
                                        ;//////////////////////////////////////////////////////////////////////

                                        .NLIST BEX

 014716      040      102      125  TIM:     .ASCIZ  / BUS TIMEOUT/

 014733      045      116      045  TFM36:   .ASCIZ  /%N%AREGISTER ADDRESS ERROR,ADDRESS = %06%A,UNIT = %02/

 015021      115      101      123  EM0001:  .ASCIZ  /MASTER CLEAR FAIL TO RESET: DCT11 CAN'T CLEAR MASTER CLEAR /

 015115      040      113      115  EM0002:  .ASCIZ  / KMV11 REGISTERS CAN'T BE CLEARED /

 015160      040      104      101  EM0003:  .ASCIZ  / DATA COMPARE ERROR ON KMV11 REGISTER (SEL2 TO SEL16)/

 015246      040      116      117  EM0004:  .ASCIZ  / NO ANSWER FROM KMV11 /

 015275      124      111      115  EM0006:  .ASCIZ  /TIMEOUT DURING KMV11 MICRO TEST /

 015336      111      116      124  EM0007:  .ASCIZ  /INTERUPT OCCURED ON KMV11  AT INCORRECT VECTOR /

 015416      113      115      126  EM0011:  .ASCIZ  /KMV11 REAL TIME CLOCK FAILED TO INTERUPT /

 015470      107      105      116  EM0012:  .ASCIZ  /GENERATOR COUNT CAN'T BE READ OR WRITE CORRECTLY /

 015552      107      105      116  EM0013:  .ASCIZ  /GENERATOR OUTPUT ISN'T IN A GOOD STATE(NO ACTION ON OUTPUT)/

 015646      116      117      040  EM0033:  .ASCIZ  /NO CHANGE IN BAUD RATE GENERATOR COUNT /

 015716      116      117      040  EM0014:  .ASCIZ  /NO ACTION ON BAUD RATE GENERATOR OUTPUT /

 015767      105      122      122  EM0015:  .ASCIZ  /ERROR WHEN TRANSMITTING IN INTERNAL LOOP WITHOUT INTERUPTS /

 016063      105      122      122  EM0016:  .ASCIZ  /ERROR WHEN TRANSMITTING  FRAMES IN INTERNAL LOOPBACK MODE /

 016156      105      122      122  EM0017:  .ASCIZ  /ERROR WHEN TRANSMITTING FRAMES IN EXTERNAL LOOPBACK /

 016243      105      122      122  EM0022:  .ASCIZ  /ERROR DURING TRANSMISSION AND RECEPTION OF FRAMES /

 016326      122      105      101  EM0023:  .ASCIZ  /REAL TIME CLOCK INTERUPT OCCURED TOO EARLY /

 016402      111      116      103  EM0024:  .ASCIZ  /INCORRECT KMV11 REPLY /

 016431      116      117      040  EM0027:  .ASCIZ  /NO LOOP BACK CONNECTOR,TEST NOT EXECUTED /

 016503      105      122      122  EM0031:  .ASCIZ  /ERROR WHEN TRANSMITTING IN INTERNAL LOOP WITHOUT INTERUPTS /

 016577      115      117      104  EM0032:  .ASCIZ  /MODEM SIGNAL ERROR ON CHANNEL IN EXTERNAL LOOPBACK /

 016663      040      120      122  EM0035:  .ASCIZ  / PROM REVISION IS NOT COMPATIBLE WITH DIAGNOSTIC REVISION/

 016755      040      111      116  EM0036:  .ASCIZ  / INTERUPT OCCURED ON DCT11 WHEN REAL TIME CLOCK IS DISABLE/

```

```
017050     045     116     045  MSEL0:   .ASCIZ  /%N%A SEL0 = %06%A   SHOULD BE  = %06%N/

017116     045     116     045  MREG0:   .ASCIZ  /%N%A SEL0 = %06%A   SHOULD BE  = %06/
017162     045     116     045  MREG2:   .ASCIZ  /%N%A SEL2 = %06%A   SHOULD BE  = %06/
017226     045     116     045  MREG4:   .ASCIZ  /%N%A SEL4 = %06%A   SHOULD BE  = %06/
017272     045     116     045  MREG6:   .ASCIZ  /%N%A SEL6 = %06%A   SHOULD BE  = %06/
017336     045     116     045  MREG10:  .ASCIZ  /%N%A SEL10= %06%A   SHOULD BE  = %06/
017402     045     116     045  MREG12:  .ASCIZ  /%N%A SEL12= %06%A   SHOULD BE  = %06/
017446     045     116     045  MREG14:  .ASCIZ  /%N%A SEL14= %06%A   SHOULD BE  = %06/
017512     045     116     045  MREG16:  .ASCIZ  /%N%A SEL16= %06%A   SHOULD BE  = %06/


017556     045     116     045  MINT:    .ASCIZ  /%N%A GOOD = %06%A  BAD =%06/

017612     045     116     045  MBSEL0:  .ASCIZ  /%N%A BSEL0 =%06%A  SHOULD BE =%06/


017654     045     116     045  MVECT:   .ASCIZ  /%N%A RECEIVE BAD VECT =%06%A SHOULD BE =%06/






017730     045     116     045  MT11V:   .ASCIZ  /%N%A   RECEIVE VECTOR  =%06%A SHOULD BE = %06/
020006     045     116     045  MFRAM1:  .ASCIZ  /%N%A   RECEIVE FRAME IS =%06%A SHOULD BE =%06/
020064     045     116     045  MFRAM2:  .ASCIZ  /%N%A   TRANSMIT SPEED IS =%06%A FRAME LENGTH =%06/


020146     045     116     045  MSTER1:  .ASCIZ  /%N%A   ERROR STATUS  =%06/
020200     045     116     045  MSTER2:  .ASCIZ  /%N%A   WORD COUNT DISCREPANCY =%06/

020243     045     116     045  MODEM1:  .ASCIZ  /%N%A   TESTED MODEM SIGNAL IS  =%06/
020307     045     116     045  MODEM2:  .ASCIZ  /%N%A   RESULT OF TEST IS  =%06/
020346     045     116     045  MODEM3:  .ASCIZ  /%N%A   MODEM SIGNAL STATE IS  =%06/
020411     045     116     045  MODEM4:  .ASCIZ  /%N%A   SEE TEST HEADER FOR SIGNAL DESCRIPTION /

020466     045     116     045  MRAMEF:  .ASCIZ  /%N%A   TXDATA = %06%A  ,    RXDATA = %06/

020537     045     116     045  MLOOP:   .ASCIZ  /%N%A   NO LOOP BACK CONNECTOR,TEST NOT EXECUTED/
                                .EVEN
```

```
020616                                  BGNMSG  PRSELO                  ;REPORT SEL0
020616                                  PRINTB  #MSEL0,SEL0,GOOD
020646  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
020652                                  BREAK
020654                                  ENDMSG



020656                          BGNMSG  PRINT
020656                          PRINTB  #MINT,GOOD,BAD
020706  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
020712                          BREAK
020714                          ENDMSG


020716                          BGNMSG  PRALL                           ;REPORT CONTENT OF ALL CSR'S
020716                          PRINTB  #MREG0,SEL0,GOOD0
020746                          PRINTB  #MREG2,SEL2,GOOD2
020776                          PRINTB  #MREG4,SEL4,GOOD4
021026                          PRINTB  #MREG6,SEL6,GOOD6
021056                          PRINTB  #MREG10,SEL10,GOOD10
021106                          PRINTB  #MREG12,SEL12,GOOD12
021136                          PRINTB  #MREG14,SEL14,GOOD14
021166                          PRINTB  #MREG16,SEL16,GOOD16
021216  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
021222                          BREAK
021224                          ENDMSG






021226                          BGNMSG  PRREG                           ;REPORT ALL CSR'S BUT SEL0
021226                          PRINTB  #MREG2,SEL2,GOOD2
021256                          PRINTB  #MREG4,SEL4,GOOD4
021306                          PRINTB  #MREG6,SEL6,GOOD6
021336                          PRINTB  #MREG10,SEL10,GOOD10
021366                          PRINTB  #MREG12,SEL12,GOOD12
021416                          PRINTB  #MREG14,SEL14,GOOD14
021446                          PRINTB  #MREG16,SEL16,GOOD16
021476  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
021502                          BREAK
021504                          ENDMSG




021506                          BGNMSG  PADFLT                          ;ADDRESS TEST
021506                          PRINTB  #TFM36,ADDR,UNIT
021536  004737  012732                  JSR     PC,CHKMAX
021542                          ENDMSG





```

```



021544                          BGNMSG  PBSELO                          ;REPORT BSELO
021544                          PRINTB  #MBSELO,BSELO,GOOD
021574  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
021600                          BREAK
021602                          ENDMSG








021604                          BGNMSG  PVECT                           ;REPORT VECTOR
021604                          PRINTB  #MVECT,VECT,GOOD
021634  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
021640                          BREAK
021642                          ENDMSG




021644                          BGNMSG  PRT11V
021644                          PRINTB  #MT11V,VECT,GOOD
021674  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
021700                          BREAK
021702                          ENDMSG




021704                          BGNMSG  PFRAME                          ;REPORT FRAME ERROR
021704                          PRINTB  #MFRAM1,RXDATA,TXDATA
021734                          PRINTB  #MFRAM2,TSPEED,LENGTH
021764  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
021770                          BREAK
021772                          ENDMSG





021774                          BGNMSG  PMODEM                          ;REPORT MODEM SIGNAL  ERROR
021774                          PRINTB  #MODEM1,GOOD
022020                          PRINTB  #MODEM2,BAD
022044                          PRINTB  #MODEM3,DATA
022070                          PRINTB  #MODEM4
022110  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
022114                          BREAK
022116                          ENDMSG




```

```



022120                          BGNMSG  PRAMEF
022120                          PRINTB  #MRAMEF,TXDATA,RXDATA           ;SHORT REPORT FOR FRAME ERROR
022150                          BREAK
022152                          ENDMSG




022154                          BGNMSG  PRSTER                                    ;REPORT ERROR STATUS ,WORD CNT
022154                          PRINTB  #MSTER1,STAERR
022200                          PRINTB  #MSTER2,WRDCNT
022224  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
022230                          BREAK
022232                          ENDMSG
```

```

                                .SBTTL  REPORT CODING SECTION


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

        022234                          BGNRPT


        022234                          EXIT    RPT


        022240                          ENDRPT


```

```
                                .SBTTL  INITIALIZE SECTION

                                ;///////////////////////////////////////////////////////////////////////
                                ;/ THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;/ AT THE BEGINNING OF EACH PASS.
                                ;///////////////////////////////////////////////////////////////////////

022242                                  BGNINIT



                                        .EVEN


                                        .EVEN

022242                                  SETVEC  #140,#170000,#340       ;ODT ROM ADDRESS                ;JB REV A-0

                                ;INITIALIZE SUBROUTINE STACK
022270  012705  012672                  MOV     #SSTACK,R5
                                ;STORE BASE LEVEL PROGRAM STACK POINTER
022274  010637  002246                  MOV     SP,PSTACK
022300  005737  002250                  TST     FTIME
022304  001011                          BNE     1$
022306  013737  000004  002252          MOV     @#4,SAVE4
022314  013737  000006  002254          MOV     @#6,SAVE6
022322  012737  000001  002250          MOV     #1,FTIME
022330  013737  002252  000004  1$:     MOV     SAVE4,@#4
022336  013737  002254  000006          MOV     SAVE6,@#6

022344                                  READEF  #EF.START                       ;START COMMAND?
022352                                  BCOMPLETE       SETUP                   ;IF YES BRANCH

022354                                  READEF  #EF.CONTINUE                    ;CONTINUE COMMAND?
022362                                  BCOMPLETE       END

022364                                  READEF  #EF.NEW                         ;NEW PASS?
022372                                  BNCOMPLETE      NEXT                    ;IF NOT EXIT SETUP

022374  012737  177777  012432  SETUP:  MOV     #-1,UUT                         ;INITIALISE UNIT NUMBER

022402  005237  012432          NEXT:   INC     UUT                             ;POINT NEXT UNIT
022406  023737  012432  002240          CMP     UUT,L$UIT                       ;ALL DONE?
022414  001521                          BEQ     ABORT                           ;IF YES END OF PASS

022416  013701  012432                  MOV     UUT,R1
022422                                  PRINTF  #RUNNING,R1                     ;PRINT RUNNING MESSAGE
                                        .EVEN


022444                                  GPHARD  UUT,R1                          ;GET P TABLE
022454                                  BNCOMPLETE      NEXT                    ;IF NOT AVAILABLE GET NEXT


022456                          GETPRM:
                                                                 ;GET ADDRESS OF KMV11
022456  011137  012450                  MOV      (R1),KMVCSR
```

```

022462  011137  012452                  MOV     (R1),KMVP02             ;GET POINTER TO KMV11 SEL02 REG
022466  062737  000002  012452          ADD     #2,KMVP02

022474  011137  012454                  MOV     (R1),KMVP04             ;GET POINTER TO KMV11 PORT REG - SEL 4
022500  062737  000004  012454          ADD     #4,KMVP04

022506  011137  012456                  MOV     (R1),KMVP06             ;GET POINTER TO KMV11 PORT REG - SEL 6
022512  062737  000006  012456          ADD     #6,KMVP06

022520  011137  012460                  MOV     (R1),KMVP10             ;GET POINTER TO KMV11 REG 10
022524  062737  000010  012460          ADD     #10,KMVP10

022532  011137  012462                  MOV     (R1),KMVP12             ;GET POINTER TO KMV11 REG 12
022536  062737  000012  012462          ADD     #12,KMVP12

022544  011137  012464                  MOV     (R1),KMVP14             ;GET POINTER TO KMV11 REG 14
022550  062737  000014  012464          ADD     #14,KMVP14

022556  012137  012466                  MOV     (R1)+,KMVP16            ;GET POINTER TO KMV11 REG 16
022562  062737  000016  012466          ADD     #16,KMVP16

022570  011137  012434                  MOV     (R1),KMVV00             ;GET POINTER TO  VECTOR 0

022574  011137  012442                  MOV     (R1),KMVV02             ;GET POINTER TO  VECTOR 2
022600  062737  000002  012442          ADD     #2,KMVV02

022606  011137  012440                  MOV     (R1),KMVV04             ;GET POINTER TO VECTOR 4
022612  062737  000004  012440          ADD     #4,KMVV04

022620  012137  012444                  MOV     (R1)+,KMVV06            ;GET POINTER TO VECTOR 6
022624  062737  000006  012444          ADD     #6,KMVV06

022632  012137  012436                  MOV     (R1)+,KMVLVL            ;GET POINTER TO TX PRIORITY LEVEL
022636  062737  000006  012446          ADD     #6,KMTLVL

022644  011137  012470                  MOV     (R1),LOOP               ;GET LOOPBACK PARAMETERS:

022650  005037  002234                  CLR     ERRCNT                  ;CLEAR ERROR COUNT
022654                                  EXIT    INIT



022660                          ABORT:  DOCLN                                   ;CLEAN UP AND ABORT PASS
022662                                  EXIT INIT                               ;EXIT





                                        .NLIST BEX
022666     045     116     045  RUNNING:        .ASCIZ  /%N%A RUNNING ON UNIT %D2%A   /
                                        .LIST BEX
                                        .EVEN




```

```
  2630 022724                        END:     ENDINIT
  2631
  2632
  2633
  2634
```

```
                                    .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.
                                    ;--
                                    .EVEN
022726                                      BGNAUTO




                                            ;DEVICE DOES NOT HAVE A "READY"
022726 013701 012450                        MOV     KMVCSR,R1       ;R1 CONTAINS BASE KMV11 ADDRESS
022732 012705 000007                        MOV     #7,R5           ;7 REGISTERS TO BE TESTED
022736 012737 022770 000004                 MOV     #2$,4           ;SET OUT TIMEOUT TRAP
                                    ;       MOV     #340,6          ;LEVEL 7                      ;JB REV A-0
022744 012737 000300 000006                 MOV     #300,6          ;LEVEL 6                      ;JB REV A-0
022752 005711                       1$:     TST     (R1)            ;REFERENCE DEVICE REGISTERS
022754 000240                               NOP
022756 062701 000002                        ADD     #2,R1           ;NEXT REGISTER
022762 005305                               DEC     R5              ;DEC REGISTER COUNT
022764 001372                               BNE     1$              ;BR IF NOT LAST REGISTER
022766 000405                               BR      3$

022770 062706 000004                2$:     ADD     #4,SP
022774                                      DODU    LOGDEV

023002 013737 002252 000004         3$:     MOV     SAVE4,4
023010 013737 002254 000006                 MOV     SAVE6,6
023016                                      ENDAUTO
```

```
                                .SBTTL  CLEANUP CODING SECTION

                                ;//////////////////////////////////////////////////////////////////////
                                ;/ THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ;/ AT THE END OF EACH PASS.
                                ;//////////////////////////////////////////////////////////////////////

023020                                  BGNCLN



023020                                  BRESET

023022                                  ENDCLN
```

```

                                        .SBTTL  DROP UNIT SECTION

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

023024                                          BGNDU





                                                .EVEN

023024                                          PRINTF  #DROPD,R0       ;UNIT DROPPED

023046                                          EXIT    DU





023052          045     116     045     DROPD:  .NLIST  BEX
                                                .ASCIZ  /%N%A UNIT %D2%A DROPPED/
                                                .LIST   BEX
                                                .EVEN

023102                                          ENDDU





```

```
                                          .SBTTL  ADD UNIT SECTION

                                          ;////////////////////////////////////////////////////////////////////
                                          ;/ THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                          ;/  TO BE (A) TESTED FOR THE FIRST TIME, OR (B) RESUMED IN TESTING.  IF
                                          ;/  "EF.AUNIT" IS SET, THE UNIT WILL BE TESTED AS A NEW UNIT.
                                          ;////////////////////////////////////////////////////////////////////




  023104                                          BGNAU
  023104                                          ENDAU
  2799                                          .SBTTL  HARDWARE TESTS
  2800
  2801
  2802
  2803                                          ;START OF CODE BLOCK WHICH IS USED AS DATA
  2804 023106                                   ROMMAP:;++
  2805                                          ; TEST TO ...
  2806                                          ;--
  2807
  2814
  2820
  2821                                          ;       BGNTST
  2822
  2828
  2829                                          ;       EXIT    TST
  2830
  2842
  2843                                                  .EVEN
  2844                                          ;       ENDTST
  2845
  2851
  2853 023106                                     BADHEAD
                                          ;********************** TEST1 *************************
  2854                                            ;*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
  2855                                            ;*DOES NOT CAUSE A TIME OUT TRAP
  2856 023106                                     BADHEAD
                                          ;********************** TEST1 *************************
  2857
  2858 023106                             BGNTST
  2859 023106  013701  012450                     MOV     KMVCSR,R1         ;R1 CONTAINS KMV11 ADDRESSES
  2860 023112  012705  000007                     MOV     #7,R5             ;7 REGISTERS TO BE TESTED
  2861 023116  012737  023154  000004             MOV     #2$,4             ;SET OUT TIMEOUT TRAP
  2862                                    ;       MOV     #340,6            ;LEVEL 7                        ;JB REV A-0
  2863 023124  012737  000300  000006             MOV     #300,6            ;LEVEL 6                        ;JB REV A-0
  2864 023132  005711                     1$:     TST     (R1)              ;REFERENCE DEVICE REGISTERS
  2865 023134  000240                             NOP
  2866 023136                                     ESCAPE  TST
  2867 023142  062701  000002                     ADD     #2,R1             ;NEXT REGISTER
  2868 023146  005305                             DEC     R5                ;DEC REGISTER COUNT
  2869 023150  001370                             BNE     1$                ;BR IF NOT LAST REGISTER
  2870 023152  000413                             BR      3$
  2871
  2872 023154  062706  000004             2$:     ADD     #4,SP
  2873 023160  010137  002354                     MOV     R1,ADDR
  2874 023164  013737  012432  012422             MOV     UUT,UNIT
  2875 023172                                     ERRHRD  0,TIM,PADFLT      ;TIME OUT ERROR
  2876
  2877 023202  013737  002252  000004     3$:     MOV     SAVE4,4
  2878 023210  013737  002254  000006             MOV     SAVE6,6
  2879 023216                                     ESCAPE  TST
  2880
  2881 023222                             ENDTST
  2882                                    .EVEN
  2883
```

```
2885 023224                          BADHEAD
                                     ;********************** TEST2 *************************
2886                                 ;CHECK PROM REVISION TO SEE IF COMPATIBLE WHITH DIAGNOSTIC
2887 023224                          BADHEAD
                                     ;********************** TEST2 *************************
2888
2889
2890
2891
2892 023224                          STARS 1
2893                                 ;READ LOCATION 2 OF THE PROM (ADDRESS 160002) WHICH CONTAINS PROM VERSION
2894                                 ;       NUMBER
2895                                 ;CHECK IF DIAGNOSTIC AND PROM ARE COMPATIBLE AND GIVE AN ERROR IF NOT
2896 023224                          STARS 1
2897
2898
2899
2900
2901
2902 023224                          BGNTST
2903 023224 004737 014372                    JSR     PC,CLRKMV               ;CLEAR ALL REGISTERS
2904 023230 004737 014474                    JSR     PC,MAINM1               ;SET MAINT MODE
2905
2906
2907 023234 004537 014632            REVPRO: JSR     R5,READ                 ;READ LOCATION 160002
2908 023240 160002                           .WORD   160002
2909
2910
2911 023242 023737 012424 012366             CMP     GDREV,BAD               ;LOOK IF COMPATIBLE
2912 023250 001410                           BEQ     1$                      ;YES
2913
2914 023252                                  ERRHRD  7,EM0035                ;REPORT THE ERROR
2915 023262 004737 012732                    JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
2916 023266                                  ESCAPE TST
2917 023272                          1$:
2918 023272                          ENDTST
```

```


023274                        BADHEAD
                              ;********************** TEST3 ************************
                              ;REAL TIME CLOCK TEST
023274                        BADHEAD
                              ;********************** TEST3 ************************






023274                        STARS 1
                              ;THIS TEST CHECK KMV11 REAL TIME CLOCK.
                              ;THE DCT11 FULLY EXECUTE THIS MICRO TEST AND GIVE A RESULT VIA CSR'S
                              ;TO THE HOST.  (TIMING IN CHECKED BY DCT11)
                              ;
                              ;
                              ;TEST DESCRIPTION:
                              ;
                              ;DCT11 ENABLE KMV11 CLOCK,AND THEN SET UP A 80 MS PERIODE CLOCK.
                              ;
                              ;DCT11 WAIT FOR AT LEAST 80 MS AND CHECK IF AN INTERUPT OCCUR
                              ;ON DCT11 CHIP AT VECTOR 130
                              ;
                              ;
                              ;DCT11 TURN OF CLOCK, WAIT AGAIN FOR MORE THAN 80 MS AND CHECK THAT
                              ;NO INTERUPT OCCUR
                              ;
                              ;ERROR REPORTING:        BSEL0=200       IF TIMEOUT DURING TEST
                              ;                        BSEL0=100       IF ERROR DURING TEST
                              ;                        BSEL0=TEST NUB  IF NO KMV11 ANSWER
                              ;                        BSEL0=0         IF TEST IS OK
                              ;
                              ;IF ERROR                SEL6=1          IF NO INTERUPT OCCUR
                              ;                        SEL6=2          IF BAD VECTOR
                              ;                        SEL6=4          IF INTERUPT OCCUR WHEN CLOCK
                              ;                                        IS NOT ENABLE
                              ;                        SEL6=10         INTERUPT OCCUR TOO EARLY
                              ;
                              ;
                              ;
                              ;                        SEL2=EXPECTED VECTOR
                              ;
                              ;
                              ;
                              ; MICRO TEST NB= 27
                              ;
                              ;
                              ;
                              ;
                              ;
                              ;CAUTION:          KMV11 CRISTAL FREQUENCY CAN'T BE CHECKED WITH THIS TEST;
                              ;                  FOR THAT THE OPERATOR MUST SCOPE THE CRISTAL SIGNAL
                              ;                  DIRECTLY ON THE MODULE ON IC Y2 (13824 KHZ)
023274                        STARS   1
```

```

023274                          BGNTST
023274  004737  014372                  JSR     PC,CLRKMV               ;CLR REG
023300  004737  014474                  JSR     PC,MAINM1               ;SET MAINT MODE
023304  004537  014556          RTCLK:  JSR     R5,TSTNUB
023310  000027                          .WORD   27

023312                                  WAITB   0,2                     ;WAIT FOR TEST EXECUTION


023332  004737  013074                  JSR     PC,TSTERR               ;CHECK BSEL0
023336  000522                          BR      1$                      ;TEST OK
023340  000423                          BR      2$                      ;TIMEOUT ERROR
023342  000432                          BR      3$                      ;NO KMV ANSWER


023344  022777  000001  167104          CMP     #1,@KMVP06              ;ERROR DURING TEST ,SEE WHICH ONE
023352  001436                          BEQ     4$                      ;NO INTERUPT OCCUR

023354  022777  000002  167074          CMP     #2,@KMVP06
023362  001442                          BEQ     5$                      ;INT ON BAD VECTOR

023364  022737  000004  012456          CMP     #4,KMVP06
023372  001454                          BEQ     6$                      ;INT OCCUR WHEN CLOCK IS DESABLE



023374  022737  000010  012456          CMP     #10,KMVP06              ;INTERUPT OCCUR TOO EARLY
023402  001460                          BEQ     7$

023404  000137  023564                  JMP     10$                     ;WRONG KMV11 ANSWER




023410                          2$:     ERRHRD  8,EM0006                ;TIMEOUT ERROR
023420  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023424                                  ESCAPE  TST



023430                          3$:     ERRHRD  9,EM0004                ;NO KMV11 ANSWER
023440  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023444                                  ESCAPE  TST


023450                          4$:     ERRHRD  10,EM0011               ;NO INTERUPT OCCUR
023460  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023464                                  ESCAPE  TST



023470  017737  166760  012374  5$:     MOV     @KMVP04,VECT            ;READ BAD VECT
023476  012737  000130  002264          MOV     #130,GOOD
023504                                  ERRHRD  11,EM0007               ;INTERUPT OCCUR AT A BAD VECTOR
023514  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023520                                  ESCAPE  TST
```

```


023524                          6$:     ERRHRD  12,EM0036               ;INT OCCUR WHEN CHIP IS DESABLE
023534  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023540                                  ESCAPE TST




023544                          7$:     ERRHRD  13,EM0023               ;INTERUPT OCCUR TOO EARLY ON KMV11
023554  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023560                                  ESCAPE  TST






023564                          10$:    ERRHRD  14,EM0024               ;INCORRECT KMV11 RESULT
023574  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023600                                  ESCAPE  TST




023604  000240                  1$:     NOP
023606                          ENDTST



```

```
023610                          BADHEAD
                                ;********************** TEST4 ************************
                                ;BAUD RATE GENERATOR TEST
023610                          BADHEAD
                                ;********************** TEST4 ************************








023610                          STARS 1
                                ;THIS TEST READ THE STATUS AND THE OUTPUT OF THE BAUD RATE GENERATOR
                                ;DURING EACH PHASE OF THE CLOCK PULSE.
                                ;NOTE:THIS TEST AND ALL THE VERIFICATIONS ARE MADE BY THE DCT11 WHICH
                                ;ONLY GIVE TEST RESULT VIA CSR'S TO THE HOST(TIMING IS CHECKED BY DCT11)
                                ;
                                ;
                                ;TEST DESCRIPTION:
                                ;       -DCT11 LOAD GENERATOR COUNT WITH MAX COUNT (=↑4.74 MSEC)
                                ;
                                ;       -READ BACK GENERATOR COUNT ,STATUS AND VALIDATE REPONSE.
                                ;
                                ;
                                ;
                                ;
                                ;       STEP 1:READ COUNT AFTER STARTING CLOCK
                                ;              CLOCK COUNT MUST BE NEGATIVE
                                ;              OUTPUT MUST BE = 1
                                ;
                                ;
                                ;
                                ;               ERROR REPORTING:
                                ;               IF COUNT=POSITIVE       BSEL0=100=ERROR
                                ;                                       SEL6 =1  =GENE COUNT CAN'T BE READ OR
                                ;                                                 WRITEN CORRECTLY
                                ;               IF OUTPUT=0             BSEL0=100=ERROR
                                ;                                       SEL6 =2  =GENE OUTPUT IS NOT CORRECT
                                ;
                                ;       STEP 2:  WAIT 2,5MSEC AND READ BACK AGAIN GENERATOR COUNT AND STATUS
                                ;                OUTPUT MUST BE = 0
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;               ERROR REPORTING:
                                ;               IF OUTPUT =1            BSEL0=100= ERROR
                                ;                                       SEL6=10  =OUTPUT ISN'T IN A GOOD STATE
                                ;
                                ;       STEP3:WAIT 2.5 MSEC MORE AND READ BACK AGAIN GENERATOR COUNT AND STATUS
                                ;             OUTPUT MUST BE = 1
                                ;
                                ;
                                ;
                                ;
                                ;       ERROR REPORTING:
                                ;
                                ;               IF OUTPUT=0             BSEL0=100=ERROR
                                ;                                       SEL6=40   =NO ACTION ON GENERATOR OUTPUT
                                ;       -ELSE EXIT
                                ;
                                ;
                                ;
                                ;
                                ;TEST 30=       TEST GENERATOR A
023610                          STARS 1
```

```
023610                                  BGNTST
023610  004737  014372                          JSR     PC,CLRKMV               ;CLR REG
023614  004737  014474                          JSR     PC,MAINM1               ;SET MAINT MODE
023620  004537  014556                          JSR     R5,TSTNUB
023624  000030                                  .WORD   30


023626                                  BDRGEN: WAITB   0,1                     ;WAIT FOR TEST EXECUTION

023646  004737  013074                          JSR     PC,TSTERR               ;CHECK BSEL0 TO SEE IF ERROR
023652  000137  024052                          JMP     BDROKO                  ;TEST OK BR AT END
023656  000402                                  BR      2$                      ;TIME OUT ERROR
023660  000401                                  BR      2$                      ;NO KMV11 ANSWER
023662  000410                                  BR      3$                      ;ERROR DURING TEST



023664                                  2$:     ERRHRD  15,EM0004               ;NO KMV11 ANSWER
023674  004737  012732                          JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023700                                          ESCAPE TST



023704                                  3$:                                     ;LOOK WHICH ERROR
023704  017737  166546  002320                  MOV     @KMVP06,SEL6            ;READ SEL6
023712  022737  000001  002320                  CMP     #1,SEL6                 ;LOOK IF ERROR 1
023720  001010                                  BNE     4$                      ;NO

023722                                          ERRHRD  16,EM0012               ;GENE COUNT CAN'T BE READ OR WRITTE CORRECTLY
023732  004737  012732                          JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023736                                          ESCAPE TST


023742  022737  000002  002320  4$:     CMP     #2,SEL6                 ;LOOK IF ERROR 2
023750  001010                                  BNE     5$                      ;NO



023752                                          ERRHRD  17,EM0013               ;GENE OUTPUT ISN'T IN A GOOD STATE
023762  004737  012732                          JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023766                                          ESCAPE TST



023772  022737  000010  002320  5$:     CMP     #10,SEL6                ;EROR10?
024000  001414                                  BEQ     GENOUT
024002  022737  000040  002320                  CMP     #40,SEL6
024010  001410                                  BEQ     GENOUT

024012                                          ERRHRD  18,EM0024               ;WRONG KMV11 ANSWER
024022  004737  012732                          JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
024026                                          ESCAPE  TST





```

```
    3188
    3189 024032                          GENOUT: ERRHRD  19.EM0014                  ;NO ACTION ON GENERATOR OUTPUT
    3190 024042  004737  012732                  JSR     PC,CHKMAX                  ;CHECK IF TOO MANY ERROR
    3191 024046                                  ESCAPE TST
    3192
    3193
    3194
    3195 024052                          BDROKO:
    3196 024052                          ENDTST
```

```

024054          BADHEAD
                ;********************** TEST5 ************************
                ;TRANSMIT DIFFERENT FRAMES (OF 500 WORDS) AT 1,2 KBAUDS SPEED IN
                ;INTERNAL MODE WITHOUT ANY INTERUPT ON CHANNEL A .
024054          BADHEAD
                ;********************** TEST5 ************************







024054          STARS 1
                ;QBUS WRITE DIFFERENT TX TABLE OF 500 WORDS, LOAD IN KMV11 CSR'S
                ;THE TX AND RX TABLE ADDRESS ,THE TABLE LENGTH AND TRANSFER SPEED
                ;
                ;
                ;
                ;DCT11 EXECUTE THE TRANSFER IN INTERNAL MODE ON CHA  AND WRITE BACK
                ;IN RX TABLE (TRANSFER FROM QBUS TO KMV11 =DMA)
                ;QBUS CHECK BSELO TO SEE THE STATUS OF THE TEST AND IF TEST DONE CHECK IF
                ;RX TABLE =TX TABLE
                ;
                ;
                ;PARAMETERS SELECTION:
                ;       SEL2=   TX TABLE ADDRESS
                ;       SEL4=   TX TABLE LENGTH
                ;       BSEL6=  EXTENDED ADDRESS OF TX TABLE
                ;       BSEL7=      "       "       RX   "
                ;       SEL12=  RX TABLE ADDRESS
                ;       SEL14=  SPEED SELECTION
                ;       BSEL16= ERROR STATUS
                ;       SEL10=  RECEIVED BYTE COUNT      DIFFERENCE BETWEEN RX AND TX TABLE
                ;                                        >0  IF TX>RX
                ;                                        <0  IF TX<RX
                ;       BSELO=  TEST STATUS
                ;
                ;
                ;
                ;TEST STATUS DESCRIPTION:
                ;       BSELO=  0       =TEST DONE CHECK RX TABLE
                ;       BSELO=  200     =TIMEOUT ERROR
                ;       BSELO=  TSTNB   =NO KMV11 ANSWER
                ;       BSELO=  100     =ERROR DURING TEST ,IN THAT CASE SEE WHICH KIND OF
                ;                                        ERROR BY TESTING BSEL16.
                ;
                ;
                ;
                ;ERROR STATUS DESCRIPTION:
                ;
                ;       WHEN BSELO=100,GIVE STATUS AND WORD COUNT DISCREPANCY
                ;
                ;
                ;       BSEL16= BIT14=1 =FCS ERROR
                ;       BSEL16= BIT13=1 =OVERRUN ERROR
```

```
                        ;       BSEL16= BIT8 =1 =ILLEGAL INTERUPT ERROR
                        ;       BSEL16= BIT7 =1 =RX ABORT ERROR
                        ;       BSEL16= BIT6 =1 =UNDERRUN ERROR
                        ;       BSEL16= BIT5 =1 =WORD COUNT DISCREPANCY
                        ;       BSEL16= BIT4 =1 =DMA IN TIMEOUT ERROR
                        ;       BSEL16= BIT3 =1 =DMA OUT TIMEOUT ERROR
                        ;       BSEL16= BIT2 =1 =CLOCK PROBLEM (NO BUFFER EMPTY)
                        ;       BSEL16= BIT1 =1 =DATA COMPARE ERROR BETWEEN TX AND RX TABLE (USED
                        ;                                        ONLY DURING SELF TEST)
                        ;
                        ;MICRO DIAG TEST DESCRIPTION:
                        ;TEST 36        =TRANSMIT  FRAMES AT 1,2KB SPEED ON CHANNEL A WITHOUT INTERUPT
                        ;
                        ;
 024054                 STARS 1
```

```
024054                                    BGNTST
024054  004737  014372                            JSR     PC,CLRKMV               ;CLR REG
024060  005037  012400                            CLR     CHANEL
024064  005037  002256                            CLR     FLAG
024070  004737  014474                            JSR     PC,MAINM1               ;SET MAINT MODE
024074  012737  000500  012410                    MOV     #500,LENGTH             ;SELECT LENGTH

024102  012737  013224  012406                    MOV     #KB1.2,TSPEED           ;SELECT SPEED

024110  012703  000001                    INTTX:  MOV     #1,R3                   ;SELECT A PATTERN


024114  005203                            TXSTAR: INC     R3                      ;NEW ONE
024116                                            BREAK
024120  013704  012410                            MOV     LENGTH,R4               ;LOAD LENGTH
024124  012702  002362                            MOV     #TTABLE,R2              ;TX TABLE ADDRESS
024130  004737  013152                    10$:    JSR     PC,GENER                ;WRITE TX TABLE
024134  013722  012372                            MOV     DATA,(R2)+
024140  005304                                    DEC     R4                      ;ALL DONE?
024142  001372                                    BNE     10$



024144  013704  012410                            MOV     LENGTH,R4
024150  012702  006362                            MOV     #RTABLE,R2              ;CLEAR RX TABLE
024154  005022                            11$:    CLR     (R2)+
024156  005304                                    DEC     R4
024160  001375                                    BNE     11$




024162  013777  012406  166274                    MOV     TSPEED,@KMVP14          ;SEND TX SPEED
024170  012777  002362  166254                    MOV     #TTABLE,@KMVP02         ;SEND TX TABLE ADDRESS
024176  013777  012410  166250                    MOV     LENGTH,@KMVP04          ;LOAD TX TABLE ADDRESS
024204  012777  006362  166250                    MOV     #RTABLE,@KMVP12         ;LOAD RX TABLE ADDRESS
024212  005077  166240                            CLR     @KMVP06





024216  004537  014556                            JSR     R5,TSTNUB
024222  000036                                    .WORD   36                      ;DO TEST 36= CHA TEST



024224                                            WAITB   0,20                    ;WAIT FOR TEST EXECUTION


024244  004737  013074                            JSR     PC,TSTERR               ;CHECK BSEL0

024250  000427                                    BR      6$                      ;TEST OK CHECK RX TABLE
024252  000402                                    BR      3$                      ;TIMEOUT ERROR
024254  000401                                    BR      3$                      ;NO KMV11 ANSWER
024256  000410                                    BR      4$                      ;CHECK SEL16 TO SEE WHICH ONE

```

```

024260                          3$:     ERRHRD  25,EM0004               ;NO KMV11 ANSWER
024270  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
024274                                  ESCAPE  TST



024300                          4$:                                     ;ERROR DURING TEST READ ERROR STATUS
                                                                        ;TO CHECK WHICH ONE

024300  017737  166162  012416          MOV     @KMVP16,STAERR          ;READ ERROR STATUS

024306  017737  166146  012420          MOV     @KMVP10,WRDCNT          ;READ WORD COUNT DISCREPANCY

024314                                  ERRHRD  26,EM0031,PRSTER        ;ERROR WHILE TX,RX FRAMES,GIVE ERROR
                                                                        ;GIVE ERROR STATUS,WORD CNT DISCREPANCY
024324                                  ESCAPE  TST








024330  012702  002362          6$:     MOV     #TTABLE,R2              ;LOAD TXTABLE ADDRESS
024334  012705  006362                  MOV     #RTABLE,R5              ;  "  RXTABLE ADDRESS
024340  013704  012410                  MOV     LENGTH,R4               ;TABLE LENGTH

024344  022225                  RXCK:   CMP     (R2)+,(R5)+             ;CHECK RX AND TX TABLE
024346  001007                          BNE     RXERR
024350  005304                          DEC     R4                      ;ALL CHECK?
024352  001374                          BNE     RXCK                    ;NO BRANCH



024354  022703  000005                  CMP     #5,R3                   ;ALL KIND OF PATTERN DONE?
024360  001255                          BNE     TXSTAR                  ;NO TRY WHITH NEW ONE

024362  000137  024504                  JMP     RXEND

024366  162705  000002          RXERR:  SUB     #2,R5
024372  162702  000002                  SUB     #2,R2

024376  011237  012402                  MOV     (R2),TXDATA
024402  011537  012404                  MOV     (R5),RXDATA

024406  005737  002256                  TST     FLAG                    ;LOOK IF 1ST ERROR
024412  001014                          BNE     7$

024414                                  ERRHRD  27,EM0015,PFRAME        ;DATA CMP ERROR
024424  005237  002256                  INC     FLAG
024430  062702  000002                  ADD     #2,R2                   ;POINT NEXT ADDRESS
024434  062705  000002                  ADD     #2,R5
024440  000137  024344                  JMP     RXCK

024444                          7$:     ERRHRD  27,0,PRAMEF             ;SHORT REPORT
```

```
024454  005237  002256                          INC     FLAG
024460  062702  000002                          ADD     #2,R2
024464  062705  000002                          ADD     #2,R5                   ;POINT NEXT ADDRESS
024470  022737  000010  002256                  CMP     #10,FLAG                ;LOOK IF 10 REPORT
024476  001322                                  BNE     RXCK

024500                                          ESCAPE  TST


024504                                  RXEND:



024504                                  ENDTST
```

```



024506                          BADHEAD
                                ;********************** TEST6 ************************
                                ;TRANSMIT DIFFERENT  FRAME OF VARIOUS LENGTH (FROM 2BYTES TO 2K BYTES)
                                ;AT 64 KBAUDS IN INTERNAL  MODE ON CHANNEL A  (TRANSMISSION WITH INTERUPT)
024506                          BADHEAD
                                ;********************** TEST6 ************************







024506                          STARS 1
                                ;QBUS WRITE DIFFERENT TX TABLE OF VARIOUS LENGTH, LOAD IN KMV11 CSR'S
                                ;THE TX AND RX TABLE ADDRESS ,THE TABLE LENGTH AND TRANSFER SPEED
                                ;
                                ;
                                ;
                                ;DCT11 EXECUTE THE TRANSFER IN INTERNAL MODE ON CHA  AND WRITTE BACK
                                ;IN RX TABLE
                                ;QBUS CHECK BSELO TO SEE THE STATUS OF THE TEST AND IF TEST DONE CHECK IF
                                ;RX TABLE =TX TABLE
                                ;SPEED=64 KBAUDS
                                ;
                                ;
                                ;PARAMETERS SELECTION:
                                ;       SEL2=   TX TABLE ADDRESS
                                ;       SEL4=   TX TABLE LENGTH
                                ;       BSEL6=  EXTENDED ADDRESS OF TX TABLE
                                ;       BSEL7=     "         "          RX    "
                                ;       SEL12=  RX TABLE ADDRESS
                                ;       SEL14=  SPEED SELECTION           (= 154      IF 64KBAUDS)
                                ;       BSEL16= ERROR STATUS
                                ;       BSELO=  TEST STATUS
                                ;       SEL10=  BYTE COUNT DESCREPANCY         >0   IF TX>RX
                                ;                                              <0   IF TX<RX
                                ;
                                ;
                                ;TEST STATUS DESCRIPTION:
                                ;       BSELO=  0        =TEST DONE CHECK RX TABLE
                                ;       BSELO=  200      =TIMEOUT ERROR
                                ;       BSELO=  TSTNB    =NO KMV11 ANSWER
                                ;       BSELO=  100      =ERROR DURING TEST ,LOOK WHICH ONE BY TESTING BSEL16
                                ;
                                ;
                                ;
                                ;ERROR STATUS DESCRIPTION:
                                ;
                                ;       WHEN BSELO=100,GIVE CONTAINT OF ERROR STATUS AND WORD COUNT DISCREPANCY
                                ;
                                ;
                                ;       BSEL16= BIT14=1 =FCS ERROR
                                ;       BSEL16= BIT13=1 =OVERRUN ERROR
                                ;       BSEL16= BIT8 =1 =ILLEGAL INTERUPT ERROR
                                ;       BSEL16= BIT7 =1 =RX ABORT ERROR
                                ;       BSEL16= BIT6 =1 =UNDERRUN ERROR
                                ;       BSEL16= BIT5 =1 =BYTE COUNT DISCREPANCY
                                ;       BSEL16= BIT4 =1 =DMA IN TIMEOUT ERROR
                                ;       BSEL16= BIT3 =1 =DMA OUT TIMEOUT ERROR
                                ;       BSEL16= BIT2 =1 =CLOCK PROBLEM (NO BUFFER EMPTY)
                                ;       BSEL16= BIT1 =1 =DATA COMPARE ERROR BETWEEN TX AND RX TABLE (USE
                                ;                                               ONLY DURING SELF TEST)
                                ;
                                ;
                                ;MICRO DIAG TEST DESCRIPTION:
                                ;TEST 40        =TRANSMIT VARIOUS LENGTH FRAME  AT 64 KBAUDS ON CHANNEL A
                                ;
                                ;
      024506                    STARS 1
```

```
024506                          BGNTST
024506  004737  014372                    JSR     PC,CLRKMV                 ;CLR REG
024512  005037  012400                    CLR     CHANEL
024516  004737  014474                    JSR     PC,MAINM1                 ;SET MAINT MODE
024522  005037  002256                    CLR     FLAG


024526  012703  000005                    MOV     #5,R3                     ;SELECT RANDOM PATTERN
                                ; THE FOLLOWING RATE WAS CHANGED FROM 72KB TO 64KB.
                                ;         MOV     #KB72,TSPEED              ;SELECT SPEED   ;JB REV A-0
024532  012737  000154  012406            MOV     #KB64,TSPEED              ;SELECT SPEED   ;JB REV A-0

024540  012737  000001  012410  TXLTAR:   MOV     #1,LENGTH                 ;START WITH 2 CHARACTERS

024546  013704  012410          TXLBGN:   MOV     LENGTH,R4
024552  012702  002362                    MOV     #TTABLE,R2
024556  004737  013152          10$:      JSR     PC,GENER                  ;WRITE TX TABLE
024562  013722  012372                    MOV     DATA,(R2)+
024566  005304                            DEC     R4
024570  001372                            BNE     10$

024572                                    BREAK

024574  013704  012410                    MOV     LENGTH,R4                 ;CLEAR RX TABLE
024600  012702  006362                    MOV     #RTABLE,R2
024604  005022                  20$:      CLR     (R2)+
024606  005304                            DEC     R4
024610  001375                            BNE     20$






024612  013777  012406  165644            MOV     TSPEED,@KMVP14            ;SEND TX SPEED
024620  012777  002362  165624            MOV     #TTABLE,@KMVP02           ;  "  TX TABLE ADDRESS
024626  013777  012410  165620            MOV     LENGTH,@KMVP04            ;  "   "     "    LENGTH
024634  012777  006362  165620            MOV     #RTABLE,@KMVP12           ;SEND RX TABLE ADDRESS
024642  005077  165610                    CLR     @KMVP06                   ;CLR EXTENDED ADDRESS



024646  004537  014556                    JSR     R5,TSTNUB
024652  000040                            .WORD   40                        ;DO TEST 40= CHA TEST

024654                                    WAITB   0,2                       ;WAIT FOR TEST EXECUTION


024674  004737  013074                    JSR     PC,TSTERR                 ;CHECK BSELO

024700  000427                            BR      6$                        ;TEST OK CHECK RX TABLE
024702  000402                            BR      3$                        ;TIMEOUT ERROR
024704  000401                            BR      3$                        ;NO KMV11 ANSWER
024706  000410                            BR      4$                        ;CHECK SEL16 TO SEE WHICH ONE


024710                          3$:       ERRHRD  28,EM0004                 ;NO KMV11 ANSWER
```

```
024720  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
024724                                  ESCAPE  TST

024730                          4$:                                     ;ERROR DURING TEST; READ ERROR STATUS
                                                                        ;TO SEE WHICH ONE

024730  017737  165532  012416          MOV     @KMVP16,STAERR          ;READ ERROR STATUS

024736  017737  165516  012420          MOV     @KMVP10,WRDCNT          ;READ WORD COUNT DISCREPANCY

024744                                  ERRHRD  29,EM0022,PRSTER                ;ERROR WHILE TX,RX FRAMES,GIVE ERROR
                                                                        ;GIVE ERROR STATUS,WORD CNT DISCREPANCY
024754                                  ESCAPE  TST




024760  012702  002362          6$:     MOV     #TTABLE,R2              ;LOAD TX TABLE ADDRESS
024764  012705  006362                  MOV     #RTABLE,R5              ; "    RX    "      "
024770  013704  012410                  MOV     LENGTH,R4               ; "    TX TABLE LENGTH


024774  022522                  RXLCK:  CMP     (R5)+,(R2)+             ;CMP TX AND RX TABLE
024776  001015                          BNE     RXLERR                  ;BR IF ERROR
025000  005304                          DEC     R4                      ;ALL DONE
025002  001374                          BNE     RXLCK                   ;NO

025004  062737  000400  012410          ADD     #400,LENGTH             ;CHANGE LENGTH
025012  022737  002000  012410          CMP     #2000,LENGTH            ;IS IT MAX?
025020  100252                          BPL     TXLBGN                  ;NO DO TEST AGAIN WHITH NEW TABLE
                                                                        ;                         LENGTH

025022  005303                          DEC     R3                      ;SELECT OTHER PATERNS
025024  001245                          BNE     TXLTAR

025026  000137  025150                  JMP     RXLEND



025032  162705  000002          RXLERR: SUB     #2,R5
025036  162702  000002                  SUB     #2,R2

025042  011237  012402                  MOV     (R2),TXDATA
025046  011537  012404                  MOV     (R5),RXDATA

025052  005737  002256                  TST     FLAG                    ;LOOK IF 1ST ERROR
025056  001014                          BNE     30$

025060                                  ERRHRD  30,EM0016,PFRAME        ;DATA CMP ERROR
025070  005237  002256                  INC     FLAG
025074  062702  000002                  ADD     #2,R2                   ;POINT NEXT ADDRESS
025100  062705  000002                  ADD     #2,R5
025104  000137  024344                  JMP     RXCK

025110                          30$:    ERRHRD  30,0,PRAMEF             ;SHORT REPORT
025120  005237  002256                  INC     FLAG
025124  062702  000002                  ADD     #2,R2
```

```
025130  062705  000002                  ADD     #2,R5                   ;POINT NEXT ADDRESS
025134  022737  000010  002256          CMP     #10,FLAG                ;LOOK IF 10 REPORT
025142  001314                          BNE     RXLCK

025144                                  ESCAPE  TST




025150                          RXLEND:
025150                          ENDTST
```

```

025152                          BADHEAD
                                ;********************** TEST7 ************************
                                ;TRANSMIT DIFFERENT FRAMES OF VARIOUS LENGTH IN EXTERNAL LOOP BACK
                                ;MODE ON CHANNEL A   AT 64KB
025152                          BADHEAD
                                ;********************** TEST7 ************************






025152                          STARS 1
                                ;
                                ;AT BEGINNING OF TEST ,CHECK IF LOOP BACK CONNECTORS ARE INSTALLED
                                ;OR NOT:IF NOT INSTALLED = EXIT TEST AND GIVE ERROR MESSAGE
                                ;                         ********************************
                                ;
                                ;
                                ;
                                ;

                                ;QBUS WRITE DIFFERENT TX TABLE OF VARIOUS LENGTH, LOAD IN KMV11 CSR'S
                                ;THE TX AND RX TABLE ADDRESS ,THE TABLE LENGTH AND TRANSFER SPEED (64KB)
                                ;
                                ;
                                ;
                                ;DCT11 EXECUTE THE TRANSFER IN EXTERNAL MODE ON CHA  AND WRITE BACK
                                ;IN RX TABLE
                                ;QBUS CHECK BSELO TO SEE THE STATUS OF THE TEST AND IF TEST DONE CHECK IF
                                ;RX TABLE =TX TABLE
                                ;
                                ;
                                ;PARAMETERS SELECTION:
                                ;       SEL2=   TX TABLE ADDRESS
                                ;       SEL4=   TX TABLE LENGTH
                                ;       BSEL6=  EXTENDED ADDRESS OF TX TABLE
                                ;       BSEL7=     "        "        RX    "
                                ;       SEL12=  RX TABLE ADDRESS
                                ;       SEL14=  SPEED SELECTION         (=154 IF 64KB)
                                ;       BSEL16= ERROR STATUS
                                ;       BSELO=  TEST STATUS
                                ;       SEL10=  RECEIVE BYTE COUNT      >0 IF TX>RX
                                ;                                       <0 IF TX<RX
                                ;
                                ;
                                ;
                                ;TEST STATUS DESCRIPTION:
                                ;       BSELO=  0       =TEST DONE CHECK RX TABLE
                                ;       BSELO=  200     =TIMEOUT ERROR
                                ;       BSELO=  TSTNB   =NO KMV11 ANSWER
                                ;       BSELO=  100     =ERROR DURING TEST ,LOOK WHICH ONE BY TESTING BSEL16
                                ;
                                ;
                                ;
                                ;ERROR STATUS DESCRIPTION:
```

```
;
;       WHEN BSEL0=100,GIVE CONTAINT OF ERROR STATUS AND WORD COUNT DISCREPANCY
;
;
;       BSEL16= BIT14=1 =FCS ERROR
;       BSEL16= BIT13=1 =OVERRUN ERROR
;       BSEL16= BIT8 =1 =ILLEGAL INTERUPT ERROR
;       BSEL16= BIT7 =1 =RX ABORT ERROR
;       BSEL16= BIT6 =1 =UNDERRUN ERROR
;       BSEL16= BIT5 =1 =BYTE COUNT DISCREPANCY
;       BSEL16= BIT4 =1 =DMA IN TIMEOUT ERROR
;       BSEL16= BIT3 =1 =DMA OUT TIMEOUT ERROR
;       BSEL16= BIT2 =1 =CLOCK PROBLEM
;       BSEL16= BIT1 =1 =DATA COMPARE ERROR BETWEEN TX AND RX TABLE (USE
;                                           ONLY DURING SELF TEST)
;
;
;MICRO DIAG TEST DESCRIPTION:
;TEST 42         =TRANSMIT VARIOUS LENGTH FRAME AT 64 KBAUDS SPEED ON CHANNEL A
;                         IN EXTERNAL LOOP BACK MODE
;
;
;
;
;CAUTION:
;--------
;RUN ONLY WITH EXTERNAL LOOP BACK CONNECTOR:
;
;
;NOTE:
;
;TO FULLY TEST KMV11 DIAGNOSTIC MUST BE RUN WITH RS422 AND RS423
;EXTERNAL LOOP BACK CONECTOR
;
;EXTERNAL LOOP BACK CONNECTOR:
;----------------------------
;KMV11 A CAN OPERATE EITHER IN RS422 OR RS 423 LEVEL CONVERTERS
;
;
;RS422 LOOP BACK:
;TO TEST COMPLETELY A KMV11 B IN RS422 MODE ,RUN THIS DIAGNOSTIC
;WHITH LOOP BACK CONNECTOR  PLUG :
;-USE H3255 TO LOOP DIRECTLY AT THE OUTPUT OF THE MODULE
;-USE H3251 PLUG AT THE END OF BC55U MODEM CABLE CONNECTOR ASSY.
;
;
;RS423 LOOP BACK:
;TO TEST COMPLETELY A KMV11-A IN RS423 MODE ,RUN THIS DIAGNOSTIC
;WHITH LOOP BACK CONNECTOR  PLUG :
;-USE H3255 TO LOOP AT THE OUTPUT OF THE MODULE
;-USE H3251 PLUG AT THE END OF BC55H MODEM CABLE CONNECTOR ASSY.
;
;
;
;RS232 LOOP BACK:
;SAME AS FOR RS423.
;
```

```
   3708                                  ;CAUTION:
   3709                                  ;USE OF H325 LOOP BACK CONNECTOR WILL CAUSE MESSAGES ERROR IN TEST 8.
   3710                                  ;
   3711                                  ;
   3712                                  ;
   3713                                  ;
   3714                                  ;
   3715                                  ;
   3716 025152                           STARS 1
```

```
025152                                  BGNTST
025152  004737  014372                          JSR     PC,CLRKMV               ;CLEAR REGISTERS
025156  005737  012470                          TST     LOOP                    ;IS LOOP BIT=1?
025162  001012                                  BNE     BGNTXA                  ;YES GO ON TEST
025164                                          PRINTF  #MLOOP                  ;NO LOOP BACK CONNECTOR
                                                                                ;TEST NOT EXECUTED
025204                                          EXIT    TST


025210  004737  014474                  BGNTXA: JSR     PC,MAINM1               ;SET MAINT MODE
025214  005037  002256                          CLR     FLAG

025220  012703  000005                          MOV     #5,R3                   ;SELECT RANDOM PATTERN
                                        ; THE FOLLOWING RATE WAS CHANGED FROM 72KB TO 64KB.
                                        ;       MOV     #KB72,TSPEED            ;SELECT SPEED   ;JB REV A-0
025224  012737  000154  012406                  MOV     #KB64,TSPEED            ;SELECT SPEED   ;JB REV A-0


025232  012737  000001  012410          TXATAR: MOV     #1,LENGTH               ;1ST TABLE LENGTH(1 WORD)

025240  013704  012410                  TXABGN: MOV     LENGTH,R4
025244                                          BREAK

025246  012702  002362                          MOV     #TTABLE,R2
025252  004737  013152                  10$:    JSR     PC,GENER                ;WRITE TABLE
025256  013722  012372                          MOV     DATA,(R2)+
025262  005304                                  DEC     R4
025264  001372                                  BNE     10$



025266  013704  012410                          MOV     LENGTH,R4               ;CLEAR RX TABLE
025272  012702  006362                          MOV     #RTABLE,R2
025276  005022                          20$:    CLR     (R2)+
025300  005304                                  DEC     R4
025302  001375                                  BNE     20$







025304  013777  012406  165152                  MOV     TSPEED,@KMVP14          ;SEND TX SPEED
025312  012777  002362  165132                  MOV     #TTABLE,@KMVP02         ;  "  TX TABLE ADDRESS
025320  013777  012410  165126                  MOV     LENGTH,@KMVP04          ;  "   "     "   LENGTH
025326  012777  006362  165126                  MOV     #RTABLE,@KMVP12         ;SEND RX TABLE ADDRESS
025334  005077  165116                          CLR     @KMVP06                 ;CLR EXTENDED ADDRESS





025340  004537  014556                  1$:     JSR     R5,TSTNUB
025344  000042                                  .WORD   42                      ;DO TEST 42= CHB TEST
```

```

025346                          2$:     WAITB   0,3                             ;WAIT FOR TEST EXECUTION


025366  004737  013074                  JSR     PC,TSTERR                       ;CHECK BSELO

025372  000427                          BR      6$                              ;TEST OK CHECK RX TABLE
025374  000402                          BR      3$                              ;TIMEOUT ERROR
025376  000401                          BR      3$                              ;NO KMV11 ANSWER
025400  000410                          BR      4$                              ;CHECK SEL16 TO SEE WHICH ONE


025402                          3$:     ERRHRD  32,EM0004                       ;NO KMV11 ANSWER
025412  004737  012732                  JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
025416                                  ESCAPE  TST


025422                          4$:                                             ;ERROR DURING TEST READ ERROR STATUS
                                                                                ;TO CHECK WHICH ONE

025422  017737  165040  012416          MOV     @KMVP16,STAERR                  ;READ ERROR STATUS

025430  017737  165024  012420          MOV     @KMVP10,WRDCNT                  ;READ WORD COUNT DISCREPANCY

025436                                  ERRHRD  33,EM0022,PRSTER                        ;ERROR WHILE TX,RX FRAMES,GIVE ERROR
                                                                                ,GIVE ERROR STATUS,WORD CNT DISCREPANCY
025446                                  ESCAPE  TST





025452  012702  002362          6$:     MOV     #TTABLE,R2                      ;LOAD TABLE PARAMETERS
025456  012705  006362                  MOV     #RTABLE,R5
025462  013704  012410                  MOV     LENGTH,R4

025466  022225                  RXACK:  CMP     (R2)+,(R5)+                     ;CHECK TX AND RX TABLE
025470  001015                          BNE     RXAERR
025472  005304                          DEC     R4
025474  001374                          BNE     RXACK

025476  062737  000400  012410          ADD     #400,LENGTH                     ;CHANGE LENGTH
025504  022737  002000  012410          CMP     #2000,LENGTH
025512  100252                          BPL     TXABGN
025514  005303                          DEC     R3                              ;SELECT NEW PATERN
025516  001245                          BNE     TXATAR                          ;ALL DONE
025520  000137  025642                  JMP     RXAEND



025524  162705  000002          RXAERR: SUB     #2,R5
025530  162702  000002                  SUB     #2,R2

025534  011237  012402                  MOV     (R2),TXDATA
025540  011537  012404                  MOV     (R5),RXDATA

025544  005737  002256                  TST     FLAG                            ;LOOK IF 1ST ERROR
```

```
025550  001014                                  BNE     30$

025552                                          ERRHRD  34,EM0015,PFRAME        ;DATA CMP ERROR
025562  005237  002256                          INC     FLAG
025566  062702  000002                          ADD     #2,R2                   ;POINT NEXT ADDRESS
025572  062705  000002                          ADD     #2,R5
025576  000137  024774                          JMP     RXLCK

025602                                  30$:    ERRHRD  34,0,PRAMEF             ;SHORT REPORT
025612  005237  002256                          INC     FLAG
025616  062702  000002                          ADD     #2,R2
025622  062705  000002                          ADD     #2,R5                   ;POINT NEXT ADDRESS
025626  022737  000010  002256                  CMP     #10,FLAG                ;LOOK IF 10 REPORT
025634  001314                                  BNE     RXACK

025636                                          ESCAPE  TST





025642                                  RXAEND:
025642                                  ENDTST
```

```

025644          BADHEAD
                ;********************** TEST8 ************************
                ;TEST MODEM SIGNALS IN EXTERNAL LOOP BACK
025644          BADHEAD
                ;********************** TEST8 ************************






025644          STARS 1
                ;HOST SET TEST NUMBER 45
                ;DCT11 TEST MODEM SIGNAL 105,106,109,111,112,107,108,125,140,141
                ;BY SETTING AND CLEARING BIT 105,108,111,141,TIS AND TESTING
                ;BIT 106,109,125,107,112,142.
                ;
                ;IF TEST =OK,DCT11 CLEAR BSELO
                ;IF ERROR SET 100 IN BSELO AND REPORT ERROR
                ;
                ;
                ;ERROR REPORT DESCRIPTION:
                ;SEL2 INDICATE WHICH MODEM SIGNAL IS TESTED
                ;SEL4 INDICATE THE RESULT OF THE TEST
                ;SEL10 INDICATE IF IT WAS DURING A CLEAR OR A SET OPERATION
                ;
                ;SEL 10 BIT 1=0 INDICATE A CLEAR OPERATION ON TESTED MODEM SIGNAL
                ;            =1     "        SET     "      "    "    "     "
                ;
                ;
                ;
                ;SEL2  FORMAT (TESTED SIGNAL):
                ;      ------------------------------------------------------------
                ;      /       /       /       / 141   / TIS   /  111  / 108   / 105   /
                ;      ------------------------------------------------------------
                ;       BIT 7   BIT 6   BIT5    BIT4    BIT3    BIT2    BIT1    BIT0
                ;
                ;
                ;
                ;RESULT OF TEST (SEL4):
                ;
                ;
                ;      ------------------------------------------------------------
                ;      /       /       / 106   / 125   / 109   /  142  / 112   / 107   /
                ;      ------------------------------------------------------------
                ;       BIT 7   BIT 6   BIT5    BIT4    BIT3    BIT2    BIT1    BIT0
                ;
                ;
                ;
```

```

;MODEM SIGNAL LINK:
;
;          SEND                                         RECEIVE
;
;
;
;       CCITT 105 (RTS)                            CCITT 106 (CTS)
;       ---------------------------------------------------------------
;                                       /
;                                       /
;                                       /          CCITT 109 (CD)
;                                       -------------------------------
;
;
;
;        CCITT 108 (DTR)                           CCITT 107 (DSR)
;        --------------------------------------------------------------
;
;
;
;
;        CCITT 111 (DTE)                           CCITT 112 (DCE)
;         ------------------------------------------------------------
;
;
;
;
;        CCITT 141 (LL)                            CCITT 142 (TM)
;        -------------------------------------------------------------
;
;
;
;
;        TIS                                        CCITT 125 (RING)
;        -------------------------------------------------------------
;
;
;
;
;
;
;CAUTION:
;---------
;RUN ONLY WITH EXTERNAL LOOP BACK CONNECTOR:
;

;
;TO BE FULLY TESTED ,KMV11 DIAGNOSTIC MUST BE RUN WITH RS422 AND RS423
;EXTERNAL LOOP BACK CONECTOR
;
;EXTERNAL LOOP BACK CONNECTOR:
;-----------------------------
;KMV11 A CAN OPERATE EITHER IN RS422 OR RS 423 LEVEL CONVERTERS
;
;
;RS422 LOOP BACK:
```

```
                                ;TO TEST COMPLETELY A KMV11 B IN RS422 MODE ,RUN THIS DIAGNOSTIC
                                ;WHITH LOOP BACK CONNECTOR  PLUG :
                                ;-USE H3255 TO LOOP DIRECTLY AT THE OUTPUT OF THE MODULE
                                ;-USE H3251 PLUG AT THE END OF BC55U MODEM CABLE CONNECTOR ASSY.
                                ;
                                ;
                                ;RS423 LOOP BACK:
                                ;TO TEST COMPLETELY A KMV11-A IN RS423 MODE ,RUN THIS DIAGNOSTIC
                                ;WHITH LOOP BACK CONNECTOR  PLUG :
                                ;-USE H3255 TO LOOP AT THE OUTPUT OF THE MODULE
                                ;-USE H3251 PLUG AT THE END OF BC55H MODEM CABLE CONNECTOR ASSY.
                                ;
                                ;
                                ;
                                ;RS232 LOOP BACK:
                                ;SAME AS FOR RS423.
                                ;
                                ;CAUTION:
                                ;USE OF H325 LOOP BACK CONNECTOR WILL CAUSE MESSAGES ERROR IN TEST 8.
                                ;
                                ;
                                ;
                                ;
025644                          STARS 1


```

```
025644                            BGNTST
025644    004737  014372                  JSR     PC,CLRKMV               ;CLEAR ALL REGISTERS

025650    005737  012470                  TST     LOOP
025654    001012                          BNE     MODSIG                  ;LOOP BACK PRESENT GO ON

025656                                    PRINTF  #MLOOP                  ;NO LOOP BACK CONNECTOR
                                                                          ;TEST NOT EXECUTED

025676                                    EXIT    TST                     ;GO TO FOLLOWING TEST



025702    004737  014474          MODSIG: JSR     PC,MAINM1               ;SET MAINTENANCE MODE
025706    004537  014556                  JSR     R5,TSTNUB
025712    000045                          .WORD   45                      ;SEND TEST 45

025714                                    WAITB   0,4

025734    004737  013074                  JSR     PC,TSTERR               ;CHECK TEST RESULT
025740    000430                          BR      3$                      ;TEST OK GO ON
025742    000402                          BR      4$                      ;TIMEOUT
025744    000401                          BR      4$                      ;NO TEST ANSWER
025746    000406                          BR      5$                      ;ERROR DURING TEST ,LOOK WHICH ONE



025750                            4$:     ERRHRD  36,EM0004               ;NO ANSWER
025760                                    ESCAPE TST

025764    017737  164462  002264  5$:     MOV     @KMVP02,GOOD            ;READ WHICH SIGNAL WAS TESTED
025772    017737  164456  012366          MOV     @KMVP04,BAD             ;   "     "    IS THE RESULT OF TEST
026000    017737  164454  012372          MOV     @KMVP10,DATA            ;READ SIGAL VALUE

026006                                    ERRHRD  37,EM0032,PMODEM        ;REPORT ERROR
026016                                    ESCAPE  TST

026022                            3$:
026022                            MODEND:


026022                            ENDTST
```

```


                                    .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.
                                    ;//////////////////////////////////////////////////////////////////////////

026024                                      BGNHRD

026026                                      GPRMA   ADDRES,0,0,160000,177776,YES
026036                                      GPRMA   VECTOR,2,0,0,674,YES
026046                                      GPRMD   PRIRTY,4,0,7000,4,7,YES
026060                                      GPRMD   LOOPBK,6,0,1,0,1,YES
026072                                      ENDHRD



026072      115     111     103     ADDRES: .ASCIZ  /MICRO-CPU  CSR ADDRESS : /
026075      122     117     055
026100      103     120     125
026103      040     040     103
026106      123     122     040
026111      101     104     104
026114      122     105     123
026117      123     040     072
026122      040     000
026124      115     111     103     VECTOR: .ASCIZ  /MICRO-CPU VECTOR ADDRESS : /
026127      122     117     055
026132      103     120     125
026135      040     126     105
026140      103     124     117
026143      122     040     101
026146      104     104     122
026151      105     123     123
026154      040     072     040
026157      000
026160      115     111     103     PRIRTY: .ASCIZ  /MICRO-CPU PRIORITY LEVEL : /
026163      122     117     055
026166      103     120     125
026171      040     120     122
026174      111     117     122
026177      111     124     131
026202      040     114     105
026205      126     105     114
026210      040     072     040
```

```
        026213     000
   4070 026214     111       123       040   LOOPBK: .ASCIZ  /IS LOOP BACK CONNECTOR PLUGGED? 0=NO,1=YES: /
        026217     114       117       117
        026222     120       040       102
        026225     101       103       113
        026230     040       103       117
        026233     116       116       105
        026236     103       124       117
        026241     122       040       120
        026244     114       125       107
        026247     107       105       104
        026252     077       040       060
        026255     075       116       117
        026260     054       061       075
        026263     131       105       123
        026266     072       040       000
   4071                                      .EVEN
   4072
   4073
   4074
   4075
   4076
   4077
```

```
                                 .SBTTL  SOFTWARE PARAMETER CODING SECTION


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

026272                                  BGNSFT



026274                                  ENDSFT




```

```
   4115
   4116 026274                              $PATCH::
   4117 026274                                      .BLKW   50
   4118
   4125
   4126 026414                                      LASTAD
        026420                              L$LAST::
   4127 026420                                      ENDMOD
   4128
   4129
```

```
  4131
  4132
  4145
  4146 026420                               BGNSETUP        1
  4147 026420                               BGNPTAB
  4148 026424  177000                       .WORD   177000
  4149 026426  000300                       .WORD   300
  4150 026430  004000                       .WORD   4000
  4151 026432  000001                       .WORD   1
  4152 026434                               ENDPTAB
  4153 026434                               ENDSETUP
  4154
  4155
  4156
  4157
  4158
  4159         000001               .END
```

SYMBOL TABLE

```
ABORT   022660          C$CVEC= 000036          EM0002  015115          GENER1  013274          I$DU  = 000041
ADDR    002354          C$DCLN= 000044          EM0003  015160          GENEX   013432          I$HRD = 000041
ADDRES  026072          C$DODU= 000051          EM0004  015246          GENINC  013424          I$INIT= 000041
ADR   = 000020 G        C$DRPT= 000024          EM0006  015275          GENISH  013302          I$MOD = 000041
ASSEMB= 000010          C$DU  = 000053          EM0007  015336          GENOUT  024032          I$MSG = 000041
BAD     012366          C$EDIT= 000003          EM0011  015416          GENRAN  013304          I$PROT= 000040
BDDAT   002360          C$ERDF= 000055          EM0012  015470          GENROT  013260          I$PTAB= 000041
BDRGEN  023626          C$ERHR= 000056          EM0013  015552          GENR0   013246          I$PWR = 000041
BDROKC  024052          C$ERRO= 000060          EM0014  015716          GENR1   013236          I$RPT = 000041
BGNTXA  025210          C$ERSF= 000054          EM0015  015767          GENSEL  013170          I$SEG = 000041
BIT0  = 000001 G        C$ERSO= 000057          EM0016  016063          GEN0    013210          I$SETU= 000041
BIT00 = 000001 G        C$ESCA= 000010          EM0017  016156          GEN1    013214          I$SFT = 000041
BIT01 = 000002 G        C$ESEG= 000005          EM0022  016243          GEN25   013230          I$SRV = 000041
BIT02 = 000004 G        C$ESUB= 000003          EM0023  016326          GEN52   013222          I$SUB = 000041
BIT03 = 000010 G        C$ETST= 000001          EM0024  016402          GETPRM  022456          I$TST = 000041
BIT04 = 000020 G        C$EXIT= 000032          EM0027  016431          GOOD    002264          J$JMP = 000167
BIT05 = 000040 G        C$GETB= 000026          EM0031  016503          GOOD0   002266          KB1.2 = 013224 G
BIT06 = 000100 G        C$GETW= 000027          EM0032  016577          GOOD1   002270          KB56  = 000174 G
BIT07 = 000200 G        C$GMAN= 000043          EM0033  015646          GOOD10  002300          KB64  = 000154 G
BIT08 = 000400 G        C$GPHR= 000042          EM0035  016663          GOOD12  002302          KB68  = 000146 G
BIT09 = 001000 G        C$GPLO= 000030          EM0036  016755          GOOD14  002304          KB72  = 000141 G
BIT1  = 000002 G        C$GPRI= 000040          END     022724          GOOD16  002306          KIND    012376
BIT10 = 002000 G        C$INIT= 000011          ERRBLK  002226 G        GOOD2   002272          KMTLVL  012446
BIT11 = 004000 G        C$INLP= 000020          ERRCNT  002234          GOOD4   002274          KMVCSR  012450
BIT12 = 010000 G        C$MANI= 000050          ERRMSG  002224 G        GOOD6   002276          KMVLVL  012436
BIT13 = 020000 G        C$MEM = 000031          ERRNBR  002222 G        G$CNTO= 000200          KMVP02  012452
BIT14 = 040000 G        C$MSG = 000023          ERRTYP  002220 G        G$DELM= 000372          KMVP04  012454
BIT15 = 100000 G        C$OPEN= 000034          EVL   = 000004 G        G$DISP= 000003          KMVP06  012456
BIT2  = 000004 G        C$PNTB= 000014          EXADDR  012362          G$EXCP= 000400          KMVP10  012460
BIT3  = 000010 G        C$PNTF= 000017          E$END = 002100          G$HILI= 000002          KMVP12  012462
BIT4  = 000020 G        C$PNTS= 000016          E$LOAD= 000035          G$LOLI= 000001          KMVP14  012464
BIT5  = 000040 G        C$PNTX= 000015          FLAG    002256          G$NO  = 000000          KMVP16  012466
BIT6  = 000100 G        C$QIO = 000377          FTIME   002250          G$OFFS= 000400          KMVV00  012434
BIT7  = 000200 G        C$RDBU= 000007          F$AU  = 000015          G$OFSI= 000376          KMVV02  012442
BIT8  = 000400 G        C$REFG= 000047          F$AUTO= 000020          G$PRMA= 000001          KMVV04  012440
BIT9  = 001000 G        C$RESE= 000033          F$BGN = 000040          G$PRMD= 000002          KMVV06  012444
BOE   = 000400 G        C$REVI= 000003          F$CLEA= 000007          G$PRML= 000000          KMV11A  002000 G
BSEL0   012370          C$RFLA= 000021          F$DU  = 000016          G$RADA= 000140          LENGTH  012410
BSEL1   002332          C$RPT = 000025          F$END = 000041          G$RADB= 000000          LOCK    002230
CBSEL0  013644          C$SEFG= 000046          F$HARD= 000004          G$RADD= 000040          LOE   = 040000 G
CHANEL  012400          C$SPRI= 000041          F$HW  = 000013          G$RADL= 000120          LOGDEV  002242
CHKMAX  012732          C$SVEC= 000037          F$INIT= 000006          G$RADO= 000020          LOKFLG  012430
CKALL   013702          C$TPRI= 000013          F$JMP = 000050          G$XFER= 000004          LOOP    012470
CKREG   014150          DATA    012372          F$MOD = 000000          G$YES = 000010          LOOPBK  026214
CKSEL0  013612          DATA1 = 052525 G        F$MSG = 000011          HELP  = 000000          LOT   = 000010 G
CLRKMV  014372          DATA2 = 125252 G        F$PROT= 000021          HOE   = 100000 G        L$ACP   002110 G
COUNT   002350          DELCT1  002260          F$PWR = 000017          IBE   = 010000 G        L$APT   002036 G
C$AU  = 000052          DELCT2  002262          F$RPT = 000012          IDU   = 000040 G        L$AU    023104 G
C$AUTO= 000061          DFPTBL  002154 G        F$SEG = 000003          IER   = 020000 G        L$AUT   002070 G
C$BRK = 000022          DIAGMC= 000000          F$SOFT= 000005          INIFLG  012426          L$AUTO  022726 G
C$BSEG= 000004          DROPD   023052          F$SRV = 000010          INTFLG  012364          L$CCP   002106 G
C$BSUB= 000002          EF.CON= 000036 G        F$SUB = 000002          INTTX   024110          L$CLEA  023020 G
C$CEFG= 000045          EF.NEW= 000035 G        F$SW  = 000014          ISR   = 000100 G        L$CO    002032 G
C$CLCK= 000062          EF.PWR= 000034 G        F$TEST= 000001          IXE   = 004000 G        L$DEPO  002011 G
C$CLEA= 000012          EF.RES= 000037 G        GDDAT   002356          I$AU  = 000041          L$DESC  002164 G
C$CLOS= 000035          EF.STA= 000040 G        GDREV   012424          I$AUTO= 000041          L$DESP  002076 G
C$CLP1= 000006          EM0001  015021          GENER   013152          I$CLN = 000041          L$DEVP  002060 G
```

SYMBOL TABLE

```
L$DISP  002132 G    L10016  022240      O$BGNS= 000000      RXEND   024504      T$NS0 = 000000
L$DLY   002116 G    L10017  022724      O$DU  = 000001      RXERR   024366      T$NS1 = 000005
L$DTP   002040 G    L10020  023016      O$ERRT= 000000      RXLCK   024774      T$PCNT= 000000
L$DTYP  002034 G    L10021  023022      O$GNSW= 000001      RXLEND  025150      T$PTAB= 010037
L$DU    023024 G    L10022  023102      O$POIN= 000001      RXLERR  025032      T$PTHV= 000001
L$DUT   002072 G    L10023  023104      O$SETU= 000001      SAVE4   002252      T$PTNU= 000001
L$DVTY  012672 G    L10024  023222      PADFLT  021506 G    SAVE6   002254      T$SAVL= 177777
L$EF    002052 G    L10025  023272      PBSEL0  021544 G    SAVPC   002244      T$SEGL= 177777
L$ENVI  002044 G    L10026  023606      PFRAME  021704 G    SAVPC1  002344      T$SIZE= 000006
L$ERRT  002220 G    L10027  024052      PMODEM  021774 G    SAVREG  013452      T$SUBN= 000000
L$ETP   002102 G    L10030  024504      PNT   = 001000 G    SAVSTA  002346      T$TAGL= 177777
L$EXP1  002046 G    L10031  025150      PRALL   020716 G    SEL0    002310      T$TAGN= 010041
L$EXP4  002064 G    L10032  025642      PRAMEF  022120 G    SEL1    002312      T$TEMP= 000000
L$EXP5  002066 G    L10033  026022      PRI   = 002000 G    SEL10   002322      T$TEST= 000010
L$HARD  026026 G    L10034  026072      PRINT   020656 G    SEL12   002324      T$TSTM= 177777
L$HIME  002120 G    L10035  026274      PRIRTY  026160      SEL14   002326      T$TSTS= 000001
L$HPCP  002016 G    L10036  026424      PRI00 = 000000 G    SEL16   002330      T$$AU = 010023
L$HPTP  002022 G    L10040  026434      PRI01 = 000040 G    SEL2    002314      T$$AUT= 010020
L$HW    002154 G    MAINM1  014474      PRI02 = 000100 G    SEL4    002316      T$$CLE= 010021
L$ICP   002104 G    MAINT0= 054000 G    PRI03 = 000140 G    SEL6    002320      T$$DAT= 010040
L$INIT  022242 G    MAINT1= 044000 G    PRI04 = 000200 G    SETUP   022374      T$$DU = 010022
L$LADP  002026 G    MAXERR  002232      PRI05 = 000240 G    SSTACK  012672      T$$HAR= 010034
L$LAST  026420 G    MAXPRI= 000300 G    PRI06 = 000300 G    STAERR  012416      T$$HW = 010001
L$LOAD  002100 G    MBSEL0  017612      PRI07 = 000340 G    SVCGBL= 000000      T$$INI= 010017
L$LUN   002074 G    MCLR  = 040000 G    PRREG   021226 G    SVCINS= 177777      T$$MSG= 010015
L$MREV  002050 G    MFRAM1  020006      PRSEL0  020616 G    SVCSUB= 177777      T$$PC = 000001
L$NAME  002000 G    MFRAM2  020064      PRSTER  022154 G    SVCTAG= 177777      T$$PRO= 010000
L$PRIO  002042 G    MINT    017556      PRT11V  021644 G    SVCTS = 177777      T$$PTA= 010037
L$PROT  002122 G    MLOOP   020537      PSTACK  002246      SVCTST= 177777      T$$RPT= 010016
L$PRT   002112 G    MODEM1  020243      PVECT   021604 G    S$LSYM= 010000      T$$SOF= 010035
L$REPP  002062 G    MODEM2  020307      QV.FLG  012431      TFM36   014733      T$$TES= 010033
L$REV   002010 G    MODEM3  020346      RANCLC  013404      TIM     014716      T1      023106 G
L$RPT   022234 G    MODEM4  020411      RANDN   002342      TSPEED  012406      T2      023224 G
L$SOFT  026274 G    MODEND  026022      RANEX   013422      TSTERR  013074      T3      023274 G
L$SPC   002056 G    MODSIG  025702      RANGEN  013324      TSTNUB  014556      T4      023610 G
L$SPCP  002020 G    MRAMEF  020466      RANMTA  002340      TTABLE  002362      T5      024054 G
L$SPTP  002024 G    MREG0   017116      RANSEC  013410      TXABGN  025240      T6      024506 G
L$STA   002030 G    MREG10  017336      RANSEL  002336      TXATAR  025232      T7      025152 G
L$SW    002236      MREG12  017402      RANST   002334      TXDATA  012402      T8      025644 G
L$TEST  002114 G    MREG14  017446      RAN1    013336      TXLBGN  024546      UAM   = 000200 G
L$TIML  002014 G    MREG16  017512      RAN2    013354      TXLTAR  024540      UNIT    012422
L$UIT   002240      MREG2   017162      RAN4    013412      TXSTAR  024114      UUT     012432
L$UNIT  002012 G    MREG4   017226      READ    014632      T$ARGC= 000001      VECT    012374
L10001  002164      MREG6   017272      REGADR  012472      T$CODE= 003032      VECTOR  026124
L10002  020654      MSEL0   017050      REVPRO  023234      T$ERRN= 000045      WAIT1   012722
L10003  020714      MSTER1  020146      ROMMAP  023106      T$EXCP= 000000      WAIT2   012702
L10004  021224      MSTER2  020200      RSTREG  013532      T$FLAG= 000040      WRDCNT  012420
L10005  021504      MT11V   017730      RTABLE  006362      T$FREE= 026434      WRITE   014604
L10006  021542      MVECT   017654      RTCLK   023304      T$GMAN= 000000      X$ALWA= 000000
L10007  021602      NERRS   013024      RUNNIN  022666      T$HILI= 000001      X$FALS= 000040
L10010  021642      NEXT    022402      RXACK   025466      T$LAST= 000001      X$OFFS= 000400
L10011  021702      NUB     012412      RXAEND  025642      T$LOLI= 000000      X$TRUE= 000020
L10012  021772      NUMBER  002352      RXAERR  025524      T$LSYM= 010000      $LSTIN= 000000
L10013  022116      O$APTS= 000000      RXCK    024344      T$LTNO= 000010      $LSTTA= 000000
L10014  022152      O$AU  = 000000      RXCNT   012414      T$NEST= 177777      $PATCH  026274 G
L10015  022232      O$BGNR= 000000      RXDATA  012404
```

```
SYMBOL TABLE

. ABS.  026434      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  28944 WORDS  ( 114 PAGES)
DYNAMIC MEMORY:  20060 WORDS  ( 77 PAGES)
ELAPSED TIME:  00:04:12
CNKMBB0.BIC,CNKMBB0.SEQ/-SP=SVC34.MLB/ML,CNKMBB0.MAC
```