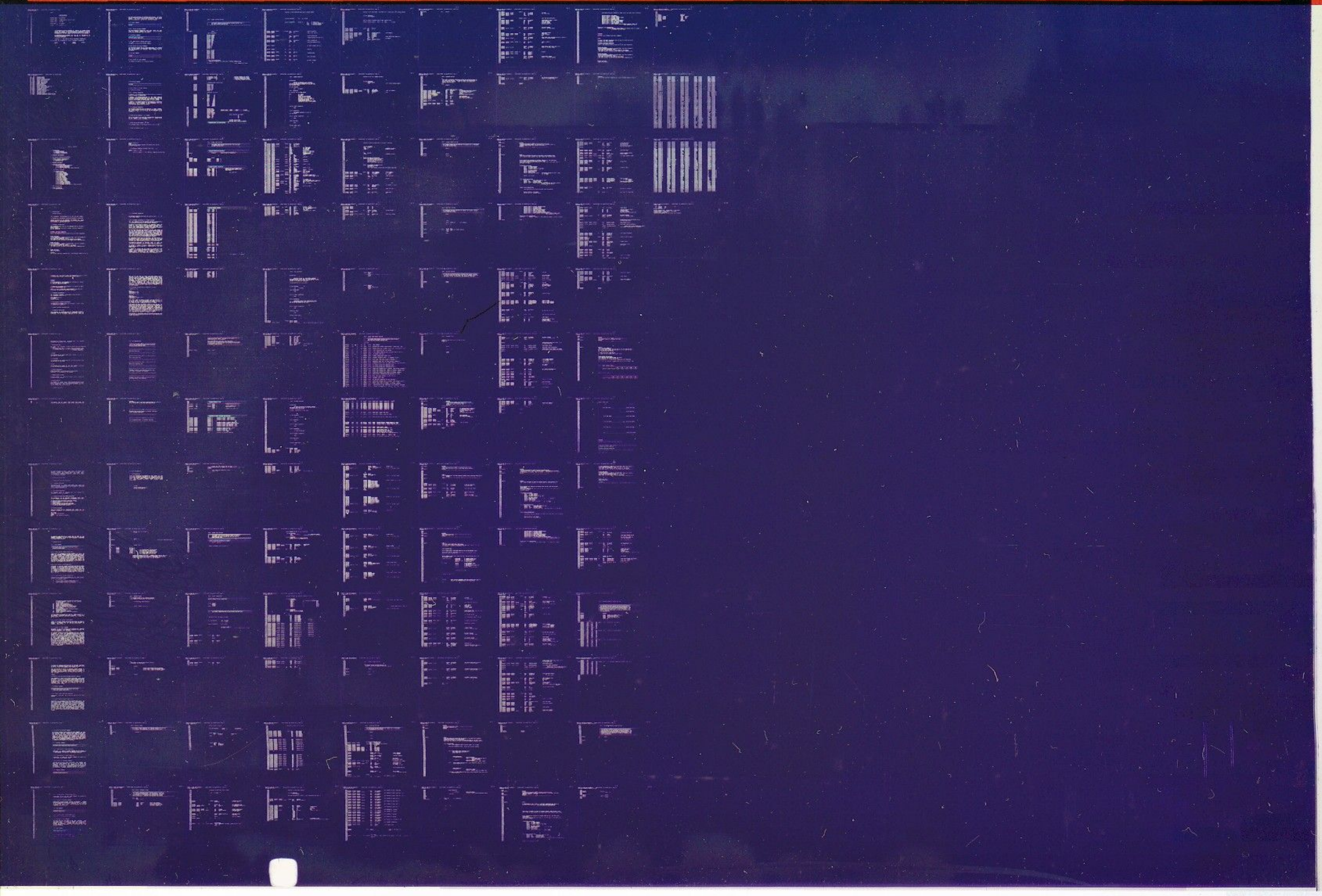
KMV11-A

LINE CNT DIAG
CVKMBAO

AH-T372A-MC
FICHE 1 OF 1



```
                        .REM @

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

                        PRODUCT CODE:   AC-T371A-MC

                        PRODUCT NAME:   CVKMBA0 KMV11A LINE CNT DIAG

                        PRODUCT DATE:   JAN 1983

                        MAINTAINER:     CSS ANNECY

                        AUTHOR:         MICHELET, GUY
```

KMV11 A LINE CNT DIAGNOSTIC MACRO M1200 06-JAN-83 09:40
TABLE OF CONTENTS

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
                                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.
```

```
                                DIAGNOSTIC WILL TEST KMV11 CLOCKS,LINE INTERUPTS,TX AND RX FUCTION
                                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
                                SUCCESFULLY IN BOTH RS422 AND RS423 LOOP BACK.


                                2.0 HARDWARE REQUIREMENTS

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

                                PDP-11/03,23,23 PLUS
                                16K MEMORY
                                CONSOLE TERMINAL


                                3.0 PRELIMINARY PROGRAM REQUIREMENTS

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


                                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 :
- 85 SECONDS FROM TEST 1 TO TEST 6 (TEST IN INTERNAL LOOP).
- 125 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
VKMBA0
KMV11 A  LINE CONTROLER 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 OCCURANCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 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
                            BCE 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.  SEF  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 DIAGLOGUE. THE UNITS WHICH ARE SELECTED MUST NOT HAVE BEEN DROPPED BY THE DROP COMMAND. SEE THE DISCUSSION OF ADD AND DROP COMMANDS BELOW. DEFAULT IS TO TEST ALL UNITS WHICH HAVE NOT BEEN DROPPED BY A DROP COMMAND.

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>
****

```
542
543
544
545
546
547                         6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)
548
549                         <FLAG-LIST>  IS  AS  IN  THE  START COMMAND, BUT UNSPECIFIED
550                         FLAGS RETAIN THEIR CURRENT VALUE.
551
552
553                         6.3.4.2 EFFECT OF PROCEED COMMAND
554
555                         PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE.   COMMAND
556                         MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR.  THE EFFECT
557                         OF THE  COMMAND  IS  TO  BEGIN  EXECUTION  AT  THE  LOCATION
558                         FOLLOWING  THE  ERROR  CALL.   NEITHER HARDWARE NOR SOFTWARE
559                         PARAMETERS MAY BE ALTERED.
560
561
562                         6.3.5 ADD COMMAND
563
564                         ************************************************************
565                         ADD/UNITS:<UNIT-LIST>
566                         ************************************************************
567
568
569                         6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>
570
571                         <UNIT-LIST> IS AS IN THE RESTART COMMAND.
572
573
574                         6.3.5.2 EFFECT OF ADD COMMAND
575
576                         THE UNITS SPECIFIED ARE ADDED TO THE  TEST  SEQUENCE.   EACH
577                         UNIT  MUST  HAVE  A  P-TABLE  IN  MEMORY  DUE  TO AN EARLIER
578                         HARDWARE DIALOGUE.  THIS  COMMAND  MUST  BE  FOLLOWED  BY  A
579                         RESTART  OR  CONTINUE.   THE UNITS SWITCH MUST BE SPECIFIED.
580                         THE ADD COMMAND IS  MEANINGFUL  ONLY  FOR  UNITS  THAT  WERE
581                         PREVIOUSLY DROPPED.
582
583
584                         6.3.6 DROP COMMAND
585
586                         ************************************************************
587                         DRO(P)/UNITS:<UNIT-LIST>
588                         ************************************************************
589
590
591                         6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)
592
593                         <UNIT-LIST> IS AS IN THE RESTART COMMAND.
594
595
596                         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 SURPRESSED 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: (O) 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: (O) 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
                                INTERUPT 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).

```
778
779
780
781
782
783                     NOW  LET  US  SEE  HOW  WE  COULD  USE THESE CAPABILITIES TO
784                     CONSTRUCT A SET OF P-TABLES.  ASSUME THAT WE HAVE 16  UNITS,
785                     AND THAT THERE ARE THREE HARDWARE PARAMETERS FOR EACH (THREE
786                     SLOTS IN  THE  P-TABLE,  THREE  HARDWARE  QUESTIONS  IN  THE
787                     DIALOGUE).  LET THE DESIRED VALUE FOR THE FIRST PARAMETER BE
788                     THE NUMBER 75 FOR ALL 16 TABLES.  LET THE DESIRED VALUE  FOR
789                     THE   SECOND   PARAMETER   BE   EQUAL   TO   THE   UNIT   NUMBER
790                     (0,1,2,...,15) EXCEPT FOR UNIT 12, WHICH SHOULD RECEIVE  THE
791                     VALUE  11.  LET THE DESIRED VALUE FOR THE THIRD PARAMETER BE
792                     THE NUMBER 76 FOR THE FIRST 7 UNITS AND THE  NUMBER  77  FOR
793                     THE LAST 9 UNITS.
794
795                     THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:
796
797                     # UNITS (D) ? 16
798
799                     UNIT 1
800                     <QUESTION 1> ? 75
801                     <QUESTION 2> ? 0-6
802                     <QUESTION 3> ? 76
803
804                     UNIT 21
805                     <QUESTION 1> ?
806                     <QUESTION 2> ? 7-11,,13-15
807                     <QUESTION 3> ? 77
808
809                     THE  FIRST  TIME THE SERIES IS ASKED, SLOT ONE RECEIVES A 75
810                     IN ALL 16 TABLES.  SLOT TWO RECEIVES THE VALUES  0,1,2,...,6
811                     IN  TABLES  0  THRU  6 AND A CONSTANT 6 IN TABLES 7 THRU 15.
812                     SLOT THREE RECEIVES A CONSTANT 76 IN ALL 16 TABLES.
813
814                     THE SECOND TIME THRU THE SERIES, TABLES 16 THRU THE END  ARE
815                     GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS
816                     PRINTED OUT FOR THE THE OPERATOR IN THE FORM  "UNIT  XX"  AT
817                     THE  BEGINNING  OF EACH SERIES).  QUESTION 1 IS RESPONDED TO
818                     BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU
819                     15, SINCE NO NEW EXLICIT VALUES ARE TYPED IN.  SLOT TWO GETS
820                     THE VALUES 7,8,9,10,11 IN TABLES 7 THRU 11, AND GETS A 11 IN
821                     SLOT  12, AND GETS THE VALUES 13,14,15 IN TABLES 13 THRU 15.
822                     SLOT THREE GETS THE VALUE 77 IN TABLES 7 THRU 15.
823
824                     THE DIALOGUE IS TERMINATED WHEN THE SOFTWARE RECOGNIZES THAT
825                     16 EXPLICIT VALUES HAVE BEEN GIVEN FOR AT LEAST ONE QUESTION
826                     (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 CHANEL A  WHITHOUT ANY INTERUPT
                                *
                                ****************************************************


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

```
                        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

a

```
                                .TITLE KMV11 A LINE CNT DIAGNOSTIC
002000                                  .=2000



                                        .MCALL  SVC
002000                                  SVC                                     ; INITIALIZE SUPERVISOR MACROS


002000                                  BGNMOD  KMV11A

000000                          $LSTIN= 0
000000                          $LSTTAG= 0
177777                          SVCINS= -1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
177777                           SVCTS= -1      ; LIST TEST TAGS, SHIFTED RIGHT
177777                          SVCSUB= -1      ; LIST SUBTEST TAGS, SHIFTED RIGHT
177777                          SVCGBL= -1      ; LIST GLOBAL TAGS, SHIFTED RIGHT
177777                          SVCTAG= -1      ; LIST OTHER TAGS, SHIFTED RIGHT

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

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

002000                                  POINTER BGNSW,BGNDU,BGNSETUP


002000                                  HEADER  VKMBA0,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
                                ;
000040                          EF.START==      32.                             ; START COMMAND WAS ISSUED
```

```
000037                  EF.RESTART==    31.             ; RESTART COMMAND WAS ISSUED
000036                  EF.CONTINUE==   30.             ; CONTINUE COMMAND WAS ISSUED
000035                  EF.NEW==        29.             ; A NEW PASS HAS BEEN STARTED
000034                  EF.PWR==        28.             ; A POWER-FAIL/POWER-UP OCCURRED
                        ;
                        ;
                        ; PRIORITY LEVEL DEFINITIONS
                        ;
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
000140                  PRI03== 140
000100                  PRI02== 100
000040                  PRI01== 40
000000                  PRI00== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==     4
000010                  LOT==    10
000020                  ADR==    20
000040                  IDU==    40
000100                  ISR==   100
000200                  UAM==   200
000400                  BOE==   400
001000                  PNT==  1000
002000                  PRI==  2000
004000                  IXE==  4000
010000                  IBE== 10000
020000                  IER== 20000
040000                  LOE== 40000
100000                  HOE== 100000
1143
1144
1145    000340          MAXPRI==340
1146    054000          MAINT0==54000           ;MASTER CLEAR = 1,MODE = 1 ,MAINT 1 = 1 ,T11=HOLD
1147    044000          MAINT1==44000           ;MASTER CLEAR = 1,MODE = 0 ,MAINT 1 = 0 ,T11=NOT HOLD
1148    040000          MCLR==40000
1149    052525          DATA1== 052525
1150    125252          DATA2== 125252
1151    013224          KB1.2== 5780.                   ;OCTAL VALUE OF 1,2 KBAUDS
1152    000154          KB64==  108.                    ;  ''      ''   '' 64   ''
1153    000174          KB56==  124.                    ;  ''      ''   56   ''
1154    000146          KB68==  102.                    ;  ''      ''   68   ''
1155    000141          KB72==  97.                     ;  ''      ''   72   ''
1156
1157                                                    ;DIVIDER CALCULATION
1158                                                    ;DECIMAL VAUE = 6912:YYY KBAUDS
1159                    ;;*************************************************************************
1160                    ;* PROGRAM EVENT FLAG DEFINITIONS
1161                    ;;*************************************************************************
1162
1163
1164
```

```
                                .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>



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



                                ;**************************************************************************
                                ;* 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
```

```
1218                            ;*****************************************************************************
1219                            ;* MISCELLANEOUS STORAGE
1220                            ;*****************************************************************************
1221 002252  000000            SAVE4:  .WORD   0
1222 002254  000000            SAVE6:  .WORD   0
1223 002256  000000            FLAG:   .WORD   0
1224
1225 002260  000000            DELCT1: .WORD   0
1226 002262  000000            DELCT2: .WORD   0
1227 002264  000000            GOOD:   .WORD   0
1228 002266  000000            GOOD0:  .WORD   0
1229 002270  000000            GOOD1:  .WORD   0
1230 002272  000000            GOOD2:  .WORD   0
1231 002274  000000            GOOD4:  .WORD   0
1232 002276  000000            GOOD6:  .WORD   0
1233 002300  000000            GOOD10: .WORD   0
1234 002302  000000            GOOD12: .WORD   0
1235 002304  000000            GOOD14: .WORD   0
1236 002306  000000            GOOD16: .WORD   0
1237 002310  000000            SEL0:   .WORD   0
1238 002312  000000            SEL1:   .WORD   0
1239 002314  000000            SEL2:   .WORD   0
1240 002316  000000            SEL4:   .WORD   0
1241 002320  000000            SEL6:   .WORD   0
1242 002322  000000            SEL10:  .WORD   0
1243 002324  000000            SEL12:  .WORD   0
1244 002326  000000            SEL14:  .WORD   0
1245 002330  000000            SEL16:  .WORD   0
1246 002332  000000            BSEL1:  .WORD   0
1247 002334  000000            RANST:  .WORD   0
1248 002336  000000            RANSEL: .WORD   0
1249 002340  000000            RANMTA: .WORD   0
1250 002342  000000            RANDN:  .WORD   0
1251 002344  000000            SAVPC1: .WORD   0
1252 002346  000000            SAVSTA: .WORD   0
1253 002350  000000            COUNT:  .WORD   0
1254 002352  000000            NUMBER: .WORD   0
1255 002354  000000            ADDR:   .WORD   0
1256 002356  000000            GDDAT:  .WORD   0
1257 002360  000000            BDDAT:  .WORD   0
1258
1259 002362                     TTABLE: .BLKW   2000
1260 006362                     RTABLE: .BLKW   2000
1261
1262 012362  000000            EXADDR: .WORD   0
1263 012364  000000            INTFLG: .WORD   0
1264 012366  000000            BAD:    .WORD   0
1265 012370  000000            BSEL0:  .WORD   0
1266 012372  000000            DATA:   .WORD   0
1267 012374  000000            VECT:   .WORD   0
1268
1269
1270 012376  000000            KIND:   .WORD   0                       ;=0 IF KMV11A ,=1 IF KMV11B
1271 012400  000000            CHANEL: .WORD   0
1272
1273 012402  000000            TXDATA: .WORD   0
1274 012404  000000            RXDATA: .WORD   0
```

GLOBAL DATA SECTION

```
1275 012406  000000                TSPEED: .WORD   0
1276 012410  000000                LENGTH: .WORD   0
1277 012412  000000                NUB:    .WORD   0
1278 012414  000000                RXCNT:  .WORD   0
1279 012416  000000                STAERR: .WORD   0
1280 012420  000000                WRDCNT: .WORD   0
1281 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
```

```
1307                            ;*************************************************************************
1308                            ;* PROGRAM CONTROL FLAGS
1309                            ;*************************************************************************
1310 012426     000            INIFLG: .BYTE   0               ;PROGRAM INITIALIZING FLAG
1311                                    .EVEN
1312 012430     000            LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
1313 012431     000            QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG
1314                                    .EVEN
1315 012432  000000            UUT:    .WORD   0               ;CURRENT UNIT UNDER TEST
1316
1317
1318
1319
1320
1321
1322                            ;*************************************************************************
1323                            ;* POINTERS TO KMV11 VECTORS AND REGISTERS
1324                            ;*************************************************************************
1325 012434  000000            KMVV00: 0       ;POINTER TO KMV11  INTRPT VECTOR 0
1326 012436  000000            KMVLVL: 0       ;POINTER TO KMV11  INTRPT SERVICE
1327 012440  000000            KMVV04: 0       ;POINTER TO KMV11  INTRPT VECTOR 04
1328 012442  000000            KMVV02: 0       ;    ''       ''       ''       ''       02
1329 012444  000000            KMVV06: 0       ;    ''       ''       ''       :        06
1330 012446  000000            KMTLVL: 0       ;POINTER TO KMV11 TX INTRPT SERVICE PS
1331 012450  000000            KMVCSR: 0       ;POINTER TO KMV11 CONTROL STATUS REGISTER
1332 012452  000000            KMVP02: 0       ;POINTER TO KMV11 PORT REGISTER - SEL2
1333 012454  000000            KMVP04: 0       ;POINTER TO KMV11 PORT REGISTER - SEL4
1334 012456  000000            KMVP06: 0       ;POINTER TO KMV11 PORT REGISTER - SEL6
1335
1336 012460  000000            KMVP10: 0       ;POINTER TO KMV11 PORT REG  -SEL10
1337 012462  000000            KMVP12: 0       ;POINTER TO PORT REG  -SEL 14
1338 012464  000000            KMVP14: 0       ;POINTER TO PORT REG  -SEL14
1339 012466  000000            KMVP16: 0       ;POINTER TO PORT REG 16
1340
1341 012470  000000            LOOP:   0       ;POINTER TO LOOP BACK CONNECTOR
```

GLOBAL DATA SECTION

```
                                ;****  PRIMARY REG ADRS STORAGE FOR THIS UNIT  ******
                                ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
012472                          REGADR:

                                ;;****  STACK USED FOR SUBROUTINE LINKAGE  *****
012472                                  .BLKW   100
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 SOBROUTINES
                                ;-----------------------------------------------------------------------

                                .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
```

```
1439
1440
1441 012722  005237  002260          WAIT1:  INC     DELCT1
1442 012726  001375                          BNE     WAIT1
1443
1444 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


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

                                                RFLAGS  R0                      ;GET OPERATOR FLAG
                                                BIT     #IDU,R0                 ;IS DROPPING INHIBITTED?
                                                BNE     1$                      ;IF YES EXIT

                                                INC     ERRCNT                  ;UPDATE ERROR COUNT
                                                CMP     ERRCNT,MAXERR           ;TOO MANY ERROR?
                                                BLE     1$                      ;IF NOT JUMP

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

                                                DOCLN                           ;END THE SUBPASS

                                        1$:     RTS     PC


                                                .NLIST  BEX
                                        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
```

```
1577                                    .SBTTL  NUMBER GENERATOR
1578
1579
1580
1581                                    DESCRIPTION:
1582
1583                                            ROUTINE TO GENERATE DATA PATTERNS,
1584                                    THE TYPE OF PATTERN IS SELECTED BY R3, AND THE
1585                                    PATTERN GENERATED IS RETURNED IN LOCATION "DATA"
1586                                    AND LOCATION "GOOD"
1587
1588                                    CALLING SEQUENCE:
1589
1590                                            JSR     PC,GENER
1591
1592                                    INPUT PARAMETERS:
1593
1594                                    R3 CONTAINS THE PATTERN NUMBER
1595
1596                                    R3=0            ALL ZEROES
1597                                    1               ALL ONES
1598                                    2               010101 ETC BIT PATTERN
1599                                    3               101010 ETC BIT PATTERN
1600                                    4               ROTATING 1 IN A ZERO WORD
1601                                    5               ROTATING 0 IN AN ALL ONE WORD
1602                                    6               PSEUDO RANDOM NUMBER
1603                                    7               INCREMENTING DATA PATTERN, GOOD
1604                                                    CONTAINS THE VALUE TO BE UPDATED
1605
1606
1607
1608
1609                                    IMPLICIT INPUT PARAMETERS:
1610
1611                                            NONE
1612
1613
1614                                    OUTPUT PARAMETERS:
1615
1616                                            THE NUMBER GENERATED IS HELD IN
1617                                    DATA AND GOOD.
1618
1619
1620                                    IMPLICIT OUTPUT PARAMETERS:
1621
1622                                            NONE
1623
1624
1625                                    COMPLETION CODES:
1626
1627                                            NONE
1628
1629
1630                                    POSSIBLE ERROR CODES:
1631
1632                                            NONE
1633
```

```
1634                                    ;
1635                                    ;
1636 013152  042703  177770      GENER:  BIC     #177770,R3
1637 013156  004737  013452              JSR     PC,SAVREG
1638 013162  006303                      ASL     R3
1639 013164  000173  013170              JMP     @GENSEL(R3)
1640 013170  013210              GENSEL: GEN0                    ;ALL ZERO WORD
1641 013172  013214                      GEN1                    ;ALL ONE WORD
1642 013174  013222                      GEN52                   ;52 PATTERN
1643 013176  013230                      GEN25                   ;25 PATTERN
1644 013200  013236                      GENR1                   ;ROTATE '1' EACH CALL
1645 013202  013246                      GENR0                   ;ROTATE '0' EACH CALL
1646 013204  013304                      GENRAN                  ;RANDOM NUMBER
1647 013206  013424                      GENINC                  ;INCREMENTING COUNT
1648 013210  005000              GEN0:   CLR     R0              ;0>R0
1649 013212  000507                      BR      GENEX
1650 013214  005000              GEN1:   CLR     R0              ;NOT0>R0
1651 013216  005100                      COM     R0
1652 013220  000504                      BR      GENEX
1653 013222  012700  052525      GEN52:  MOV     #52525,R0       ;5252>R0
1654 013226  000501                      BR      GENEX
1655 013230  012700  125252      GEN25:  MOV     #125252,R0      ;125252>R0
1656 013234  000476                      BR      GENEX
1657 013236  000241              GENR1:  CLC
1658 013240  004737  013260              JSR     PC,GENROT       ;SHIFT 1 > R0
1659 013244  000472                      BR      GENEX
1660 013246  000241              GENR0:  CLC
1661 013250  004737  013260              JSR     PC,GENROT       ;
1662 013254  005100                      COM     R0              ;SHIFT 0 > R0
1663 013256  000465                      BR      GENEX
1664 013260  006037  013302      GENROT: ROR     GENISH          ;ROTATE  1 PATTERN
1665 013264  001003                      BNE     GENER1          ;= 0?
1666 013266  012737  100000  013302      MOV     #100000,GENISH  ;YES, SET MSB
1667 013274  013700  013302      GENER1: MOV     GENISH,R0       ;PUT 1 IN R0
1668 013300  000207                      RTS     PC              ;AND EXIT
1669 013302  000001              GENISH: 1
1670 013304  012737  000005  002336 GENRAN: MOV     #5,RANSEL       ;SET SELECT VALUE TO 5
1671 013312  004737  013324              ISR     PC,RANGEN       ;GENERATE RANDOM NUMBER IN R0
1672 013316  013700  002342              MOV     RANDN,R0        ;
1673 013322  000443                      BR      GENEX           ;
1674 013324  013702  002342      RANGEN: MOV     RANDN,R2        ;
1675 013330  001002                      BNE     RAN1            ;IS RANDOM = 0
1676 013332  013702  002334              MOV     RANST,R2        ;YES, PUT RANDOM START VALUE IN
1677 013336  032737  000777  002336 RAN1:   BIT     #777,RANSEL     ;NO;IS RANSEL SELECT VALUE = 0
1678 013344  001003                      BNE     RAN2            ;NO
1679 013346  012737  000001  002336      MOV     #1,RANSEL       ;YES: SET RANSEL = 1
1680 013354  013703  002336      RAN2:   MOV     RANSEL,R3       ;
1681 013360  013702  002342              MOV     RANDN,R2
1682 013364  033702  002340              BIT     RANMTA,R2       ;GET R2 <0 AND 1>
1683 013370  001405                      BEQ     RANCLC          ;
1684 013372  005102                      COM     R2
1685 013374  033702  002340              BIT     RANMTA,R2
1686 013400  001401                      BEQ     RANCLC
1687 013402  000402                      BR      RANSEC
1688 013404  000241              RANCLC: CLC
1689 013406  000401                      BR      RAN4
1690 013410  000261              RANSEC: SEC
```

```
1691 013412  006037  002342          RAN4:   ROR     RANDN           ;ROTATE C TO B15
1692 013416  005303                          DEC     R3              ;IS THIS NUMBER REQUIRED?
1693 013420  001357                          BNE     RAN2+4          ;NO, GET ANOTHER
1694 013422  000207                  RANEX:  RTS     PC              ;YES, EXIT
1695 013424  013700  002264          GENINC: MOV     GOOD,R0         ;INCREMENTS LOC. 'GOOD'
1696 013430  005200                          INC     R0
1697 013432  010037  002264          GENEX:  MOV     R0,GOOD
1698 013436  004737  013532                  JSR     PC,RSTREG
1699 013442  013737  002264  012372          MOV     GOOD,DATA
1700 013450  000207                          RTS     PC
1701
```

```
                                        .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 7 TO AVOID ANY 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
                                ;
013402  SAVREG: GETPRI  SAVSTA
013460          SETPRI  MAXPRI
013466  012637  002244          MOV     (SP)+,SAVPC     ;SAVE PC FOR RETURN FROM THIS ROUTINE
```

SAVE REGISTERS

```
1760 013472  012637  002344                MOV     (SP)+,SAVPC1
1761 013476  010546                        MOV     R5,-(SP)
1762 013500  010446                        MOV     R4,-(SP)
1763 013502  010346                        MOV     R3,-(SP)
1764 013504  010246                        MOV     R2,-(SP)
1765 013506  010146                        MOV     R1,-(SP)
1766 013510  010046                        MOV     R0,-(SP)
1767 013512  013746  002344                MOV     SAVPC1,-(SP)
1768 013516  013746  002244                MOV     SAVPC,-(SP)     ;PUT PC READY FOR
1769 013522                                SETPRI  SAVSTA
1770 013530  000207                        RTS     PC              ;RETURN
1771
1772
1773
```

RESTORE REGISTERS

```
                        .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

```
1832 013562  012602                        MOV     (SP)+,R2
1833 013564  012603                        MOV     (SP)+,R3
1834 013566  012604                        MOV     (SP)+,R4
1835 013570  012605                        MOV     (SP)+,R5
1836 013572  013746  002344                MOV     SAVPC1,-(SP)
1837 013576  013746  002244                MOV     SAVPC,-(SP)     ;PUT PC READY FOR
1838 013602                                SETPRI  SAVSTA
1839 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          CKSELO: MOV     (R5)+,GOOD      ;WRITE GOOD
013616  017737  176626  002310          MOV     @KMVCSR,SELO    ;READ SEL 0
013624  023737  002310  002264          CMP     SELO,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          CBSELO: CLR     GOOD
013650  012537  002264                  MOV     (R5)+,GOOD
013654  117737  176570  012370          MOVB    @KMVCSR,BSELO
013662  123737  012370  002264          CMPB    BSELO,GOOD
013670  001001                          BNE     1$
013672  000402                          BR      2$
013674  062705  000002          1$:     ADD     #2,R5
013700  000205                  2$:     RTS     R5
```

```
1878                                    ;ROUTINE TO CHECK ALL REGISTER FROM SEL0 TO SEL16
1879
1880
1881                                    ;CALLING SEQUENCE:
1882                                    ;       JSR R5,CKALL
1883                                    ;       .WORD A                 A = EXPECTED VALUE FOR SEL0
1884                                    ;       .WORD B                 B       "       "       SEL2
1885                                    ;       .WORD C                 C       "       "       SEL4
1886                                    ;       .WORD D                 D       "       "       SEL6
1887                                    ;       .WORD E                 E       "       "       SEL10
1888                                    ;       .WORD F                 F       "       "       SEL12
1889                                    ;       .WORD G                 G       "       "       SEL14
1890                                    ;       .WORD H                 H       "       "       SEL16
1891
1892
1893                                    ;OUTPUT PARAMETER:
1894                                    ;       BRANCH IN PC+2 IF ERROR
1895                                    ;       BRANCH IN PC IF NO ERROR
1896
1897
1898
1899 013702  012537  002266            CKALL:  MOV     (R5)+,GOOD0
1900 013706  012537  002272                    MOV     (R5)+,GOOD2
1901 013712  012537  002274                    MOV     (R5)+,GOOD4
1902 013716  012537  002276                    MOV     (R5)+,GOOD6
1903 013722  012537  002300                    MOV     (R5)+,GOOD10
1904 013726  012537  002302                    MOV     (R5)+,GOOD12
1905 013732  012537  002304                    MOV     (R5)+,GOOD14
1906 013736  012537  002306                    MOV     (R5)+,GOOD16
1907
1908 013742  017737  176502  002310            MOV     @KMVCSR,SEL0            ;READ SEL0
1909 013750  000240                    NOP
1910 013752  017737  176474  002314    MOV     @KMVP02,SEL2            ;READ SEL2
1911 013760  000240                    NOP
1912 013762  017737  176466  002316    MOV     @KMVP04,SEL4            ;READ SEL4
1913 013770  000240                    NOP
1914 013772  017737  176460  002320    MOV     @KMVP06,SEL6            ;READ SEL6
1915 014000  000240                    NOP
1916 014002  017737  176452  002322    MOV     @KMVP10,SEL10           ;READ SEL10
1917 014010  000240                    NOP
1918 014012  017737  176444  002324    MOV     @KMVP12,SEL12           ;READ SEL12
1919 014020  000240                    NOP
1920 014022  017737  176436  002326    MOV     @KMVP14,SEL14           ;READ SEL14
1921 014030  000240                    NOP
1922 014032  017737  176430  002330    MOV     @KMVP16,SEL16           ;READ SEL16
1923
1924 014040  023737  002310  002266    CMP     SEL0,GOOD0
1925 014046  001035                    BNE     1$
1926 014050  023737  002314  002272    CMP     SEL2,GOOD2
1927 014056  001031                    BNE     1$
1928 014060  023737  002316  002274    CMP     SEL4,GOOD4
1929 014066  001025                    BNE     1$
1930 014070  023737  002320  002276    CMP     SEL6,GOOD6
1931 014076  001021                    BNE     1$
1932 014100  023737  002322  002300    CMP     SEL10,GOOD10
1933 014106  001015                    BNE     1$
1934 014110  023737  002324  002302    CMP     SEL12,GOOD12
```

RESTORE REGISTERS

```
   1935 014116  001011                          BNE     1$
   1936 014120  023737  002326  002304          CMP     SEL14,GOOD14
   1937 014126  001005                          BNE     1$
   1938 014130  023737  002330  002306          CMP     SEL16,GOOD16
   1939 014136  001001                          BNE     1$
   1940
   1941 014140  000402                          BR      2$
   1942 014142  062705  000002          1$:     ADD     #2,R5
   1943 014146  000205                  2$:     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          MOV     @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
```

RESTORE REGISTERS

```
                                        ;ROUTINE TO SET MAINT MODE 1 AND CHECK DCT11 CLEAR SEL0 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,CKSEL0               ;CHECK SEL0=0 BUT MODE BIT =1
014540  004000                       .WORD   4000
014542  000404                       BR      1$                      ;OK BRANCH
014544                               ERRHRD  2,EM0001,PRSEL0
014554  000207                1$:    RTS     PC
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
```

```
                                        ;ROUTINE TO SET TEST NUMBER ON BSEL0
                                        ;
                                        ;
                                        ;
                                        ;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
```

RESTORE REGISTERS

```
2147 014662  000402                          BR      2$
2148 014664  000401                          BR      2$
2149 014666  000400                          BR      2$
2150
2151 014670                          2$:     ERRHRD  4,EM0004                 ;NO KMV ANSWER
2152 014700  004737  012732                  JSR     PC,CHKMAX
2153 014704  000205                          RTS     R5
2154
2155 014706  017737  175542  012366  1$:     MOV     @KMVP04,BAD             ;READ DATA IN BAD
2156 014714  000205                          RTS     R5
2157
2158
2159
2160
2161
```

```
2163
2164                            .MACRO  ED$CALL XY
2165                                    .LIST
2166                            ;********************* TEST'XY' ************************
2167                                    .NLIST
2168                                    .ENDM
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178                            .MACRO  BADHEAD
2179                                    .RADIX  10
2180                                    ED$CALL \T$TESTNUM+1
2181                                    .RADIX  8
2182                                    .ENDM
2183
2184
```

```
                                .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/
```

```
2244 017050  045  116  045  MSEL0:  .ASCIZ  /%N%A SEL0 = %O6%A   SHOULD BE  = %O6%N/
2245
2246 017116  045  116  045  MREG0:  .ASCIZ  /%N%A SEL0 = %O6%A   SHOULD BE  = %O6/
2247 017162  045  116  045  MREG2:  .ASCIZ  /%N%A SEL2 = %O6%A   SHOULD BE  = %O6/
2248 017226  045  116  045  MREG4:  .ASCIZ  /%N%A SEL4 = %O6%A   SHOULD BE  = %O6/
2249 017272  045  116  045  MREG6:  .ASCIZ  /%N%A SEL6 = %O6%A   SHOULD BE  = %O6/
2250 017336  045  116  045  MREG10: .ASCIZ  /%N%A SEL10= %O6%A   SHOULD BE  = %O6/
2251 017402  045  116  045  MREG12: .ASCIZ  /%N%A SEL12= %O6%A   SHOULD BE  = %O6/
2252 017446  045  116  045  MREG14: .ASCIZ  /%N%A SEL14= %O6%A   SHOULD BE  = %O6/
2253 017512  045  116  045  MREG16: .ASCIZ  /%N%A SEL16= %O6%A   SHOULD BE  = %O6/
2254
2255
2256 017556  045  116  045  MINT:   .ASCIZ  /%N%A GOOD = %O6%A  BAD =%O6/
2257
2258 017612  045  116  045  MBSEL0: .ASCIZ  /%N%A BSEL0 =%O6%A  SHOULD BE =%O6/
2259
2260
2261 017654  045  116  045  MVECT:  .ASCIZ  /%N%A RECEIVE BAD VECT =%O6%A SHOULD BE =%O6/
2262
2263
2264
2265
2266
2267 017730  045  116  045  MT11V:  .ASCIZ  /%N%A   RECEIVE VECTOR  =%O6%A SHOULD BE = %O6/
2268 020006  045  116  045  MFRAM1: .ASCIZ  /%N%A   RECEIVE FRAME IS =%O6%A SHOULD BE =%O6/
2269 020064  045  116  045  MFRAM2: .ASCIZ  /%N%A   TRANSMIT SPEED IS =%O6%A FRAME LENGTH =%O6/
2270
2271
2272 020146  045  116  045  MSTER1: .ASCIZ  /%N%A   ERROR STATUS  =%O6/
2273 020200  045  116  045  MSTER2: .ASCIZ  /%N%A   WORD COUNT DISCREPANCY =%O6/
2274
2275 020243  045  116  045  MODEM1: .ASCIZ  /%N%A   TESTED MODEM SIGNAL IS  =%O6/
2276 020307  045  116  045  MODEM2: .ASCIZ  /%N%A   RESULT OF TEST IS  =%O6/
2277 020346  045  116  045  MODEM3: .ASCIZ  /%N%A   MODEM SIGNAL STATE IS  =%O6/
2278 020411  045  116  045  MODEM4: .ASCIZ  /%N%A   SEE TEST HEADER FOR SIGNAL DESCRIPTION /
2279
2280 020466  045  116  045  MRAMEF: .ASCIZ  /%N%A   TXDATA = %O6%A  ,    RXDATA = %O6/
2281
2282 020537  045  116  045  MLOOP:  .ASCIZ  /%N%A   NO LOOP BACK CONNECTOR,TEST NOT EXECUTED/
2283                          .EVEN
```

```
2285 020616                                BGNMSG  PRSEL0                  ;REPORT SEL0
2286 020616                                PRINTB  #MSEL0,SEL0,GOOD
2287 020646  004737  012732                JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
2288 020652                                BREAK
2289 020654                                ENDMSG
2290
2291
2292
2293 020656                          BGNMSG  PRINT
2294 020656                          PRINTB  #MINT,GOOD,BAD
2295 020706  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
2296 020712                          BREAK
2297 020714                          ENDMSG
2298
2299
2300 020716                          BGNMSG  PRALL                           ;REPORT CONTENT OF ALL CSR'S
2301 020716                          PRINTB  #MREG0,SEL0,GOOD0
2302 020746                          PRINTB  #MREG2,SEL2,GOOD2
2303 020776                          PRINTB  #MREG4,SEL4,GOOD4
2304 021026                          PRINTB  #MREG6,SEL6,GOOD6
2305 021056                          PRINTB  #MREG10,SEL10,GOOD10
2306 021106                          PRINTB  #MREG12,SEL12,GOOD12
2307 021136                          PRINTB  #MREG14,SEL14,GOOD14
2308 021166                          PRINTB  #MREG16,SEL16,GOOD16
2309 021216  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
2310 021222                          BREAK
2311 021224                          ENDMSG
2312
2313
2314
2315
2316
2317
2318 021226                          BGNMSG  PRREG                           ;REPORT ALL CSR'S BUT SEL0
2319 021226                          PRINTB  #MREG2,SEL2,GOOD2
2320 021256                          PRINTB  #MREG4,SEL4,GOOD4
2321 021306                          PRINTB  #MREG6,SEL6,GOOD6
2322 021336                          PRINTB  #MREG10,SEL10,GOOD10
2323 021366                          PRINTB  #MREG12,SEL12,GOOD12
2324 021416                          PRINTB  #MREG14,SEL14,GOOD14
2325 021446                          PRINTB  #MREG16,SEL16,GOOD16
2326 021476  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
2327 021502                          BREAK
2328 021504                          ENDMSG
2329
2330
2331
2332
2333 021506                          BGNMSG  PADFLT                          ;ADDRESS TEST
2334 021506                          PRINTB  #TFM36,ADDR,UNIT
2335 021536  004737  012732                  JSR     PC,CHKMAX
2336 021542                          ENDMSG
2337
2338
2339
2340
2341
```

```
2342
2343
2344
2345 021544                          BGNMSG  PBSEL0                          ;REPORT BSEL0
2346 021544                          PRINTB  #MBSEL0,BSEL0,GOOD
2347 021574  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
2348 021600                          BREAK
2349 021602                          ENDMSG
2350
2351
2352
2353
2354
2355
2356
2357
2358 021604                          BGNMSG  PVECT                           ;REPORT VECTOR
2359 021604                          PRINTB  #MVECT,VECT,GOOD
2360 021634  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
2361 021640                          BREAK
2362 021642                          ENDMSG
2363
2364
2365
2366
2367 021644                          BGNMSG  PRT11V
2368 021644                          PRINTB  #MT11V,VECT,GOOD
2369 021674  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
2370 021700                          BREAK
2371 021702                          ENDMSG
2372
2373
2374
2375
2376 021704                          BGNMSG  PFRAME                          ;REPORT FRAME ERROR
2377 021704                          PRINTB  #MFRAM1,RXDATA,TXDATA
2378 021734                          PRINTB  #MFRAM2,TSPEED,LENGTH
2379 021764  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
2380 021770                          BREAK
2381 021772                          ENDMSG
2382
2383
2384
2385
2386
2387 021774                          BGNMSG  PMODEM                          ;REPORT MODEM SIGNAL  ERROR
2388 021774                          PRINTB  #MODEM1,GOOD
2389 022020                          PRINTB  #MODEM2,BAD
2390 022044                          PRINTB  #MODEM3,DATA
2391 022070                          PRINTB  #MODEM4
2392 022110  004737  012732          JSR     PC,CHKMAX                       ;CHECK IF TOO MANY ERROR
2393 022114                          BREAK
2394 022116                          ENDMSG
2395
2396
2397
2398
```

```
2399
2400
2401
2402 022120                          BGNMSG  PRAMEF
2403 022120                          PRINTB  #MRAMEF,TXDATA,RXDATA          ;SHORT REPORT FOR FRAME ERROR
2404 022150                          BREAK
2405 022152                          ENDMSG
2406
2407
2408
2409
2410 022154                          BGNMSG  PRSTER                                 ;REPORT ERROR STATUS ,WORD CNT
2411 022154                          PRINTB  #MSTER1,STAERR
2412 022200                          PRINTB  #MSTER2,WRDCNT
2413 022224  004737  012732          JSR     PC,CHKMAX                      ;CHECK IF TOO MANY ERROR
2414 022230                          BREAK
2415 022232                          ENDMSG
2416
2417
2418
2419
2420
2421
2422
```

```
2424
2425                                    .SBTTL  REPORT CODING SECTION
2426
2427
2428                                    ;++
2429                                    ; THE REPORT CODING SECTION CONTAINS THE
2430                                    ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
2431                                    ;--
2432
2433 022234                                    BGNRPT
2434
2440
2441 022234                                    EXIT    RPT
2442
2449
2450 022240                                    ENDRPT
2451
2452
```

```
                                .SBTTL  INITIALIZE SECTION

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

022242                                  BGNINIT


                                        .EVEN


                                        .EVEN


                                ;INITIALIZE SUBROUTINE STACK
022242  012705  012672                  MOV     #SSTACK,R5
                                ;STORE BASE LEVEL PROGRAM STACK POINTER
022246  010637  002246                  MOV     SP,PSTACK
022252  005737  002250                  TST     FTIME
022256  001011                          BNE     1$
022260  013737  000004  002252          MOV     @#4,SAVE4
022266  013737  000006  002254          MOV     @#6,SAVE6
022274  012737  000001  002250          MOV     #1,FTIME
022302  013737  002252  000004  1$:     MOV     SAVE4,@#4
022310  013737  002254  000006          MOV     SAVE6,@#6

022316                                  READEF  #EF.START                       ;START COMMAND?
022324                                  BCOMPLETE       SETUP                   ;IF YES BRANCH

022326                                  READEF  #EF.CONTINUE                    ;CONTINUE COMMAND?
022334                                  BCOMPLETE       END

022336                                  READEF  #EF.NEW                         ;NEW PASS?
022344                                  BNCOMPLETE      NEXT                    ;IF NOT EXIT SETUP

022346  012737  177777  012432  SETUP:  MOV     #-1,UUT                         ;INITIALISE UNIT NUMBER

022354  005237  012432          NEXT:   INC     UUT                             ;POINT NEXT UNIT
022360  023737  012432  002240          CMP     UUT,L$UIT                       ;ALL DONE?
022366  001521                          BEQ     ABORT                           ;IF YES END OF PASS

022370  013701  012432                  MOV     UUT,R1
022374                                  PRINTF  #RUNNING,R1                     ;PRINT RUNNING MESSAGE
                                        .EVEN


022416                                  GPHARD  UUT,R1                          ;GET P TABLE
022426                                  BNCOMPLETE      NEXT                    ;IF NOT AVAILABLE GET NEXT

022430                          GETPRM:
                                                                        ;GET ADDRESS OF KMV11
022430  011137  012450                  MOV     (R1),KMVCSR
```

```
2545
2546 022434  011137  012452                MOV     (R1),KMVP02          ;GET POINTER TO KMV11 SEL02 REG
2547 022440  062737  000002  012452         ADD     #2,KMVP02
2548
2549 022446  011137  012454                MOV     (R1),KMVP04          ;GET POINTER TO KMV11 PORT REG - SEL 4
2550 022452  062737  000004  012454         ADD     #4,KMVP04
2551
2552 022460  011137  012456                MOV     (R1),KMVP06          ;GET POINTER TO KMV11 PORT REG - SEL 6
2553 022464  062737  000006  012456         ADD     #6,KMVP06
2554
2555 022472  011137  012460                MOV     (R1),KMVP10          ;GET POINTER TO KMV11 REG 10
2556 022476  062737  000010  012460         ADD     #10,KMVP10
2557
2558 022504  011137  012462                MOV     (R1),KMVP12          ;GET POINTER TO KMV11 REG 12
2559 022510  062737  000012  012462         ADD     #12,KMVP12
2560
2561 022516  011137  012464                MOV     (R1),KMVP14          ;GET POINTER TO KMV11 REG 14
2562 022522  062737  000014  012464         ADD     #14,KMVP14
2563
2564 022530  012137  012466                MOV     (R1)+,KMVP16         ;GET POINTER TO KMV11 REG 16
2565 022534  062737  000016  012466         ADD     #16,KMVP16
2566
2567 022542  011137  012434                MOV     (R1),KMVV00          ;GET POINTER TO  VECTOR 0
2568
2569 022546  011137  012442                MOV     (R1),KMVV02          ;GET POINTER TO  VECTOR 2
2570 022552  062737  000002  012442         ADD     #2,KMVV02
2571
2572 022560  011137  012440                MOV     (R1),KMVV04          ;GET POINTER TO VECTOR 4
2573 022564  062737  000004  012440         ADD     #4,KMVV04
2574
2575 022572  012137  012444                MOV     (R1)+,KMVV06         ;GET POINTER TO VECTOR 6
2576 022576  062737  000006  012444         ADD     #6,KMVV06
2577
2578 022604  012137  012436                MOV     (R1)+,KMVLVL         ;GET POINTER TO TX PRIORITY LEVEL
2579 022610  062737  000006  012446         ADD     #6,KMTLVL
2580
2581 022616  011137  012470                MOV     (R1),LOOP            ;GET LOOPBACK PARAMETERS:
2582
2583 022622  005037  002234                CLR     ERRCNT               ;CLEAR ERROR COUNT
2584 022626                                EXIT    INIT
2585
2586
2587
2588 022632                          ABORT: DOCLN                        ;CLEAN UP AND ABORT PASS
2589 022634                                 EXIT INIT                    ;EXIT
2590
2591
2592
2593
2594                                       .NLIST BEX
2595 022640     045     116     045  RUNNING:        .ASCIZ  /%N%A RUNNING ON UNIT %D2%A   PASS TIME =  2 MINUTES/
2596                                       .LIST BEX
2597                                       .EVEN
2598
2599
2600
2601
```

```
 2602 022724                        END:    ENDINIT
 2603
 2604
 2605
 2606
```

```
                                        .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$,4           ;SET OUT TIMEOUT TRAP
022744  012737  000340  000006          MOV     #340,6          ;LEVEL 7
022752  005711                  1$:     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
```

CLEANUP CODING SECTION

```
                                .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



                                        .NLIST  BEX
023052  045  116  045  DROPD:  .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
```

```
                                .SBTTL  HARDWARE TESTS



                                ;START OF CODE BLOCK WHICH IS USED AS DATA
023106                          ROMMAP::++
                                ; TEST TO ...
                                ;--



                                ;       BGNTST

                                ;       EXIT    TST

                                        .EVEN
                                ;       ENDTST
```

```
2824 023106                                     BADHEAD
                                        ;********************* TEST1 ************************
2825                                            ;*VERIFY THAT REFERENCING UNIBUS DEVICE REGISTERS
2826                                            ;*DOES NOT CAUSE A TIME OUT TRAP
2827 023106                                     BADHEAD
                                        ;********************* TEST1 ************************
2828
2829 023106                             BGNTST
2830 023106  013701  002450                        MOV     KMVCSR,R1       ;R1 CONTAINS KMV11 ADDRESSES
2831 023112  012705  000007                        MOV     #7,R5           ;7 REGISTERS TO BE TESTED
2832 023116  012737  023154  000004                MOV     #2$,4           ;SET OUT TIMEOUT TRAP
2833 023124  012737  000340  000006                MOV     #340,6          ;LEVEL 7
2834 023132  005711                     1$:        TST     (R1)            ;REFERENCE DEVICE REGISTERS
2835 023134  000240                                NOP
2836 023136                                        ESCAPE  TST
2837 023142  062701  000002                        ADD     #2,R1           ;NEXT REGISTER
2838 023146  005305                                DEC     R5              ;DEC REGISTER COUNT
2839 023150  001370                                BNE     1$              ;BP IF NOT LAST REGISTER
2840 023152  000413                                BR      3$
2841
2842 023154  062706  000004             2$:        ADD     #4,SP
2843 023160  010137  002354                        MOV     R1,ADDR
2844 023164  013737  012432  012422                MOV     UUT,UNIT
2845 023172                                        ERRHRD  0,TIM,PADFLT    ;TIME OUT ERROR
2846
2847 023202  013737  002252  000004     3$:        MOV     SAVE4,4
2848 023210  013737  002254  000006                MOV     SAVE6,6
2849 023216                                        ESCAPE  TST
2850
2851 023222                             ENDTST
2852                                    .EVEN
2853
```

```
                                BADHEAD
                                ;********************** TEST2 ************************
                                ;CHECK PROM REVISION TO SEE IF COMPATIBLE WHITH DIAGNOSTIC
                                BADHEAD
                                ;********************** TEST2 ************************



                                STARS 1
                                ;READ LOCATION 2 OF THE PROM (ADDRESS 160002) WHICH CONTAINS PROM VERSION
                                ;       NUMBER
                                ;CHECK IF DIAGNOSTIC AND PROM ARE COMPATIBLE AND GIVE AN ERROR IF NOT
                                STARS 1



                                BGNTST
023224  004737  014372                 JSR     PC,CLRKMV               ;CLEAR ALL REGISTERS
023230  004737  014474                 JSR     PC,MAINM1               ;SET MAINT MODE

023234  004537  014632         REVPRO: JSR     R5,READ                 ;READ LOCATION 160002
023240  160002                         .WORD   160002

023242  023737  012424  012366         CMP     GDREV,BAD               ;LOOK IF COMPATIBLE
023250  001410                         BEQ     1$                      ;YES

023252                                 ERRHRD  7,EM0035                ;REPORT THE ERROR
023262  004737  012732                 JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023266                                 ESCAPE TST
023272                         1$:
023272                         ENDTST
```

```
2890
2891
2892 023274                    BADHEAD
                               ;********************* TEST3 ************************
2893                           ;REAL TIME CLOCK TEST
2894 023274                    BADHEAD
                               ;********************* TEST3 ************************
2895
2896
2897
2898
2899
2900
2901 023274                    STARS 1
2902                           ;THIS TEST CHECK KMV11 REAL TIME CLOCK.
2903                           ;THE DCT11 FULLY EXECUTE THIS MICRO TEST AND GIVE A RESULT VIA CSR'S
2904                           ;TO THE HOST.  (TIMING IN CHECKED BY DCT11)
2905                           ;
2906                           ;
2907                           ;TEST DESCRIPTION:
2908                           ;
2909                           ;DCT11 ENABLE KMV11 CLOCK,AND THEN SET UP A 80 MS PERIODE CLOCK.
2910                           ;
2911                           ;DCT11 WAIT FOR AT LEAST 80 MS AND CHECK IF AN INTERUPT OCCUR
2912                           ;ON DCT11 CHIP AT VECTOR 130
2913                           ;
2914                           ;
2915                           ;DCT11 TURN OF CLOCK, WAIT AGAIN FOR MORE THAN 80 MS AND CHECK THAT
2916                           ;NO INTERUPT OCCUR
2917                           ;
2918                           ;ERROR REPORTING:       BSEL0=200        IF TIMEOUT DURING TEST
2919                           ;                       BSEL0=100        IF ERROR DURING TEST
2920                           ;                       BSEL0=TEST NUB   IF NO KMV11 ANSWER
2921                           ;                       BSEL0=0          IF TEST IS OK
2922                           ;
2923                           ;IF ERROR               SEL6=1           IF NO INTERUPT OCCUR
2924                           ;                       SEL6=2           IF BAD VECTOR
2925                           ;                       SEL6=4           IF INTERUPT OCCUR WHEN CLOCK
2926                           ;                                        IS NOT ENABLE
2927                           ;                       SEL6=10          INTERUPT OCCUR TOO EARLY
2928                           ;
2929                           ;
2930                           ;
2931                           ;                       SEL2=EXPECTED VECTOR
2932                           ;
2933                           ;
2934                           ;
2935                           ; MICRO TEST NB= 27
2936                           ;
2937                           ;
2938                           ;
2939                           ;
2940                           ;
2941                           ;CAUTION:        KMV11 CRISTAL FREQUENCY CAN'T BE CHECKED WITH THIS TEST;
2942                           ;                FOR THAT THE OPERATOR MUST SCOPE THE CRISTAL SIGNAL
2943                           ;                DIRECTLY ON THE MODULE ON IC Y2 (13824 KHZ)
2944 023274                    STARS   1
```

```
2946
2947 023274                                BGNTST
2948 023274  004737  014372                       JSR     PC,CLRKMV              ;CLR REG
2949 023300  004737  014474                       JSR     PC,MAINM1              ;SET MAINT MODE
2950 023304  004537  014556                RTCLK: JSR     R5,TSTNUB
2951 023310  000027                              .WORD   27
2952
2953 023312                                      WAITB   0,2                    ;WAIT FOR TEST EXECUTION
2954
2955
2956 023332  004737  013074                      JSR     PC,TSTERR              ;CHECK BSEL0
2957 023336  000522                              BR      1$                     ;TEST OK
2958 023340  000423                              BR      2$                     ;TIMEOUT ERROR
2959 023342  000432                              BR      3$                     ;NO KMV ANSWER
2960
2961
2962 023344  022777  000001  167104              CMP     #1,@KMVP06             ;ERROR DURING TEST ,SEE WHICH ONE
2963 023352  001436                              BEQ     4$                     ;NO INTERUPT OCCUR
2964
2965 023354  022777  000002  167074              CMP     #2,@KMVP06
2966 023362  001442                              BEQ     5$                     ;INT ON BAD VECTOR
2967
2968 023364  022737  000004  012456              CMP     #4,KMVP06
2969 023372  001454                              BEQ     6$                     ;INT OCCUR WHEN CLOCK IS DESABLE
2970
2971
2972
2973 023374  022737  000010  012456              CMP     #10,KMVP06             ;INTERUPT OCCUR TOO EARLY
2974 023402  001460                              BEQ     7$
2975
2976 023404  000137  023564                      JMP     10$                    ;WRONG KMV11 ANSWER
2977
2978
2979
2980
2981 023410                                2$:   ERRHRD  8,EM0006               ;TIMEOUT ERROR
2982 023420  004737  012732                      JSR     PC,CHKMAX              ;CHECK IF TOO MANY ERROR
2983 023424                                      ESCAPE  TST
2984
2985
2986
2987 023430                                3$:   ERRHRD  9,EM0004               ;NO KMV11 ANSWER
2988 023440  004737  012732                      JSR     PC,CHKMAX              ;CHECK IF TOO MANY ERROR
2989 023444                                      ESCAPE  TST
2990
2991
2992 023450                                4$:   ERRHRD  10,EM0011              ;NO INTERUPT OCCUR
2993 023460  004737  012732                      JSR     PC,CHKMAX              ;CHECK IF TOO MANY ERROR
2994 023464                                      ESCAPE  TST
2995
2996
2997
2998 023470  017737  166760  012374        5$:   MOV     @KMVP04,VECT           ;READ BAD VECT
2999 023476  012737  000130  002264              MOV     #130,GOOD
3000 023504                                      ERRHRD  11,EM0007              ;INTERUPT OCCUR AT A BAD VECTOR
3001 023514  004737  012732                      JSR     PC,CHKMAX              ;CHECK IF TOO MANY ERROR
3002 023520                                      ESCAPE  TST
```

```
                3003
                3004
023524                          6$:     ERRHRD  12,EM0036              ;INT OCCUR WHEN CHIP IS DESABLE
023534  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023540                                  ESCAPE  TST
                3008
                3009
                3010
                3011
023544                          7$:     ERRHRD  13,EM0023              ;INTERUPT OCCUR TOO EARLY ON KMV11
023554  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023560                                  ESCAPE  TST
                3015
                3016
                3017
                3018
                3019
                3020
023564                          10$:    ERRHRD  14,EM0024              ;INCORRECT KMV11 RESULT
023574  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
023600                                  ESCAPE  TST
                3024
                3025
                3026
                3027
023604  000240                  1$:     NOP
023606                          ENDTST
                3030
                3031
                3032
```

```
3034 023610                     BADHEAD
                                ;********************** TEST4 ************************
3035                            ;BAUD RATE GENERATOR TEST
3036 023610                     BADHEAD
                                ;********************** TEST4 ************************
3037
3038
3039
3040
3041
3042
3043
3044
3045 023610                     STARS 1
3046                            ;THIS TEST READ THE STATUS AND THE OUTPUT OF THE BAUD RATE GENERATOR
3047                            ;DURING EACH PHASE OF THE CLOCK PULSE.
3048                            ;NOTE:THIS TEST AND ALL THE VERIFICATIONS ARE MADE BY THE DCT11 WHICH
3049                            ;ONLY GIVE TEST RESULT VIA CSR'S TO THE HOST(TIMING IS CHECKED BY DCT11)
3050                            ;
3051                            ;
3052                            ;TEST DESCRIPTION:
3053                            ;       -DCT11 LOAD GENERATOR COUNT WITH MAX COUNT (=^4.74 MSEC)
3054                            ;
3055                            ;       -READ BACK GENERATOR COUNT ,STATUS AND VALIDATE REPONSE.
3056                            ;
3057                            ;
3058                            ;
3059                            ;
3060                            ;       STEP 1:READ COUNT AFTER STARTING CLOCK
3061                            ;       STEP 1:READ COUNT AFTER STARTING CLOCK
3062                            ;       CLOCK COUNT MUST BE NEGATIVE
3063                            ;       OUTPUT MUST BE = 1
3064                            ;
3065                            ;
3066                            ;        ERROR REPORTING:
3067                            ;        IF COUNT=POSITIVE      BSEL0=100=ERROR
3068                            ;                               SEL6 =1  =GENE COUNT CAN'T BE READ OR
3069                            ;                                        WRITEN CORRECTLY
3070                            ;        IF OUTPUT=0            BSEL0=100=ERROR
3071                            ;                               SEL6 =2  =GENE OUTPUT IS NOT CORRECT
3072                            ;
3073                            ;STEP 2:  WAIT 2,5MSEC AND READ BACK AGAIN GENERATOR COUNT AND STATUS
3074                            ;        OUTPUT MUST BE = 0
3075                            ;
3076                            ;
3077                            ;
3078                            ;
3079                            ;        ERROR REPORTING:
3080                            ;        IF OUTPUT =1           BSEL0=100= ERROR
3081                            ;                               SEL6=10  =OUTPUT ISN'T IN A GOOD STATE
3082                            ;
3083                            ;STEP3:WAIT 2.5 MSEC MORE AND READ BACK AGAIN GENERATOR COUNT AND STATUS
3084                            ;      OUTPUT MUST BE = 1
3085                            ;
3086                            ;
3087                            ;
3088                            ;
```

```
                                ;       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
```

```
3101 023610                                BGNTST
3102 023610  004737  014372                       JSR     PC,CLRKMV          ;CLR REG
3103 023614  004737  014474                       JSR     PC,MAINM1          ;SET MAINT MODE
3104 023620  004537  014556                       JSR     R5,TSTNUB
3105 023624  000030                               .WORD   30
3106
3107
3108 023626                                BDRGEN: WAITB   0,1                ;WAIT FOR TEST EXECUTION
3109
3110 023646  004737  013074                       JSR     PC,TSTERR          ;CHECK BSEL0 TO SEE IF ERROR
3111 023652  000137  024052                       JMP     BDROK0             ;TEST OK BR AT END
3112 023656  000402                               BR      2$                 ;TIME OUT ERROR
3113 023660  000401                               BR      2$                 ;NO KMV11 ANSWER
3114 023662  000410                               BR      3$                 ;ERROR DURING TEST
3115
3116
3117
3118 023664                                2$:     ERRHRD  15,EM0004          ;NO KMV11 ANSWER
3119 023674  004737  012732                       JSR     PC,CHKMAX          ;CHECK IF TOO MANY ERROR
3120 023700                                       ESCAPE TST
3121
3122
3123
3124 023704                                3$:                                ;LOOK WHICH ERROR
3125 023704  017737  166546  002320               MOV     @KMVP06,SEL6       ;READ SEL6
3126 023712  022737  000001  002320               CMP     #1,SEL6            ;LOOK IF ERROR 1
3127 023720  001010                               BNE     4$                 ;NO
3128
3129 023722                                       ERRHRD  16,EM0012          ;GENE COUNT CAN'T BE READ OR WRITTE CORRECTLY
3130 023732  004737  012732                       JSR     PC,CHKMAX          ;CHECK IF TOO MANY ERROR
3131 023736                                       ESCAPE TST
3132
3133
3134 023742  022737  000002  002320  4$:         CMP     #2,SEL6            ;LOOK IF ERROR 2
3135 023750  001010                               BNE     5$                 ;NO
3136
3137
3138
3139 023752                                       ERRHRD  17,EM0013          ;GENE OUTPUT ISN'T IN A GOOD STATE
3140 023762  004737  012732                       JSR     PC,CHKMAX          ;CHECK IF TOO MANY ERROR
3141 023766                                       ESCAPE TST
3142
3143
3144
3145 023772  022737  000010  002320  5$:         CMP     #10,SEL6           ;EROR10?
3146 024000  001414                               BEQ     GENOUT
3147 024002  022737  000040  002320               CMP     #40,SEL6
3148 024010  001410                               BEQ     GENOUT
3149
3150 024012                                       ERRHRD  18,EM0024          ;WRONG KMV11 ANSWER
3151 024022  004737  012732                       JSR     PC,CHKMAX          ;CHECK IF TOO MANY ERROR
3152 024026                                       ESCAPE  TST
3153
3154
3155
3156
3157
```

```
   3158
   3159 024032                          GENOUT: ERRHRD  19,EM0014               ;NO ACTION ON GENERATOR OUTPUT
   3160 024042  004737  012732                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
   3161 024046                                  ESCAPE TST
   3162
   3163
   3164
   3165 024052                          BDROK0:
   3166 024052                          ENDTST
```

```
                        BADHEAD
                        ;********************** TEST5 ************************
                        ;TRANSMIT DIFFERENT FRAMES (OF 500 WORDS) AT 1,2 KBAUDS SPEED IN
                        ;INTERNAL MODE WITHOUT ANY INTERUPT ON CHANNEL A .
                        BADHEAD
                        ;********************** TEST5 ************************
                        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 BSEL0 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
                        ;       BSEL0=  TEST STATUS
                        ;
                        ;
                        ;
                        ;TEST STATUS DESCRIPTION:
                        ;       BSEL0=  0       =TEST DONE CHECK RX TABLE
                        ;       BSEL0=  200     =TIMEOUT ERROR
                        ;       BSEL0=  TSTNB   =NO KMV11 ANSWER
                        ;       BSEL0=  100     =ERROR DURING TEST ,IN THAT CASE SEE WHICH KIND OF
                        ;                                       ERROR BY TESTING BSEL16.
                        ;
                        ;
                        ;
                        ;ERROR STATUS DESCRIPTION:
                        ;
                        ;       WHEN BSEL0=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
```

```
3239 024054                                BGNTST
3240 024054  004737  014372                      JSR     PC,CLRKMV               ;CLR REG
3241 024060  005037  012400                      CLR     CHANEL
3242 024064  005037  002256                      CLR     FLAG
3243 024070  004737  014474                      JSR     PC,MAINM1               ;SET MAINT MODE
3244 024074  012737  000500  012410              MOV     #500,LENGTH             ;SELECT LENGTH
3245
3246 024102  012737  013224  012406              MOV     #KB1.2,TSPEED           ;SELECT SPEED
3247
3248 024110  012703  000001          INTTX:  MOV     #1,R3                   ;SELECT A PATTERN
3249
3250
3251 024114  005203                  TXSTAR: INC     R3                      ;NEW ONE
3252 024116                                  BREAK
3253 024120  013704  012410                  MOV     LENGTH,R4               ;LOAD LENGTH
3254 024124  012702  002362                  MOV     #TTABLE,R2              ;TX TABLE ADDRESS
3255 024130  004737  013152          10$:    JSR     PC,GENER                ;WRITE TX TABLE
3256 024134  013722  012372                  MOV     DATA,(R2)+
3257 024140  005304                          DEC     R4                      ;ALL DONE?
3258 024142  001372                          BNE     10$
3259
3260
3261
3262 024144  013704  012410                  MOV     LENGTH,R4
3263 024150  012702  006362                  MOV     #RTABLE,R2              ;CLEAR RX TABLE
3264 024154  005022          11$:    CLR     (R2)+
3265 024156  005304                          DEC     R4
3266 024160  001375                          BNE     11$
3267
3268
3269
3270
3271 024162  013777  012406  166274          MOV     TSPEED,@KMVP14          ;SEND TX SPEED
3272 024170  012777  002362  166254          MOV     #TTABLE,@KMVP02         ;SEND TX TABLE ADDRESS
3273 024176  013777  012410  166250          MOV     LENGTH,@KMVP04          ;LOAD TX TABLE ADDRESS
3274 024204  012777  006362  166250          MOV     #RTABLE,@KMVP12         ;LOAD RX TABLE ADDRESS
3275 024212  005077  166240                  CLR     @KMVP06
3276
3277
3278
3279
3280
3281 024216  004537  014556                  JSR     R5,TSTNUB
3282 024222  000036                          .WORD   36                      ;DO TEST 36= CHA TEST
3283
3284
3285
3286 024224                                  WAITB   0,20                    ;WAIT FOR TEST EXECUTION
3287
3288
3289 024244  004737  013074                  JSR     PC,TSTERR               ;CHECK BSEL0
3290
3291 024250  000427                          BR      6$                      ;TEST OK CHECK RX TABLE
3292 024252  000402                          BR      3$                      ;TIMEOUT ERROR
3293 024254  000401                          BR      3$                      ;NO KMV11 ANSWER
3294 024256  000410                          BR      4$                      ;CHECK SEL16 TO SEE WHICH ONE
3295
```

```
3296
3297 024260                          3$:    ERRHRD  25,EM0004              ;NO KMV11 ANSWER
3298 024270  004737  012732                 JSR     PC,CHKMAX              ;CHECK IF TOO MANY ERROR
3299 024274                                 ESCAPE  TST
3300
3301
3302
3303 024300                          4$:                                    ;ERROR DURING TEST READ ERROR STATUS
3304                                                                        ;TO CHECK WHICH ONE
3305
3306 024300  017737  166162  012416        MOV     @KMVP16,STAERR         ;READ ERROR STATUS
3307
3308 024306  017737  166146  012420        MOV     @KMVP10,WRDCNT         ;READ WORD COUNT DISCREPANCY
3309
3310 024314                                 ERRHRD  26,EM0031,PRSTER       ;ERROR WHILE TX,RX FRAMES,GIVE ERROR
3311                                                                        ;GIVE ERROR STATUS,WORD CNT DISCREPANCY
3312 024324                                 ESCAPE  TST
3313
3314
3315
3316
3317
3318
3319
3320
3321 024330  012702  002362          6$:    MOV     #TTABLE,R2             ;LOAD TXTABLE ADDRESS
3322 024334  012705  006362                 MOV     #RTABLE,R5             ;   "  RXTABLE ADDRESS
3323 024340  013704  012410                 MOV     LENGTH,R4              ;TABLE LENGTH
3324
3325 024344  022225                RXCK:    CMP     (R2)+,(R5)+            ;CHECK RX AND TX TABLE
3326 024346  001007                         BNE     RXERR
3327 024350  005304                         DEC     R4                     ;ALL CHECK?
3328 024352  001374                         BNE     RXCK                   ;NO BRANCH
3329
3330
3331
3332 024354  022703  000005                CMP     #5,R3                  ;ALL KIND OF PATTERN DONE?
3333 024360  001255                         BNE     TXSTAR                 ;NO TRY WHITH NEW ONE
3334
3335 024362  000137  024504                JMP     RXEND
3336
3337 024366  162705  000002          RXERR: SUB     #2,R5
3338 024372  162702  000002                 SUB     #2,R2
3339
3340 024376  011237  012402                MOV     (R2),TXDATA
3341 024402  011537  012404                MOV     (R5),RXDATA
3342
3343 024406  005737  002256                TST     FLAG                   ;LOOK IF 1ST ERROR
3344 024412  001014                         BNE     7$
3345
3346 024414                                 ERRHRD  27,EM0015,PFRAME       ;DATA CMP ERROR
3347 024424  005237  002256                INC     FLAG
3348 024430  062702  000002                ADD     #2,R2                  ;POINT NEXT ADDRESS
3349 024434  062705  000002                ADD     #2,R5
3350 024440  000137  024344                JMP     RXCK
3351
3352 024444                          7$:    ERRHRD  27,0,PRAMEF            ;SHORT REPORT
```

```
3353 024454  005237  002256                  INC     FLAG
3354 024460  062702  000002                  ADD     #2,R2
3355 024464  062705  000002                  ADD     #2,R5                   ;POINT NEXT ADDRESS
3356 024470  022737  000010  002256          CMP     #10,FLAG                ;LOOK IF 10 REPORT
3357 024476  001322                          BNE     RXCK
3358
3359 024500                                  ESCAPE  TST
3360
3361
3362 024504                          RXEND:
3363
3364
3365
3366 024504                          ENDTST
```

```
                                BADHEAD
                                ;********************** TEST6 ************************
                                ;TRANSMIT DIFFERENT  FRAME OF VARIOUS LENGTH (FROM 2BYTES TO 2K BYTES)
                                ;AT 72 KBAUDS IN INTERNAL  MODE ON CHANNEL A  (TRANSMISSION WITH INTERUPT)
                                BADHEAD
                                ;********************** TEST6 ************************

                                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 BSEL0 TO SEE THE STATUS OF THE TEST AND IF TEST DONE CHECK IF
                                ;RX TABLE =TX TABLE
                                ;SPEED=72 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         (= 141      IF 72KBAUDS)
                                ;       BSEL16= ERROR STATUS
                                ;       BSEL0=  TEST STATUS
                                ;       SEL10=  BYTE COUNT DESCREPANCY      >0    IF TX>RX
                                ;                                           <0    IF TX<RX
                                ;
                                ;TEST STATUS DESCRIPTION:
                                ;       BSEL0=  0       =TEST DONE CHECK RX TABLE
                                ;       BSEL0=  200     =TIMEOUT ERROR
                                ;       BSEL0=  TSTNB   =NO KMV11 ANSWER
                                ;       BSEL0=  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 (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 72 KBAUDS ON CHANNEL A
                                ;
                                ;
3438 024506                     STARS 1
```

```
3440 024506                                BGNTST
3441 024506  004737  014372                       JSR     PC,CLRKMV               ;CLR REG
3442 024512  004737  014474                       JSR     PC,MAINM1               ;SET MAINT MODE
3443 024516  005037  002256                       CLR     FLAG
3444
3445
3446 024522  012703  000005                       MOV     #5,R3                   ;SELECT RANDOM PATTERN
3447 024526  012737  000141  012406               MOV     #KB72,TSPEED            ;SELECT SPEED
3448
3449 024534  012737  000001  012410  TXLTAR: MOV     #1,LENGTH               ;START WITH 2 CHARACTERS
3450
3451 024542  013704  012410          TXLBGN: MOV     LENGTH,R4
3452 024546  012702  002362                   MOV     #TTABLE,R2
3453 024552  004737  013152          10$:    JSR     PC,GENER                ;WRITE TX TABLE
3454 024556  013722  012372                   MOV     DATA,(R2)+
3455 024562  005304                           DEC     R4
3456 024564  001372                           BNE     10$
3457
3458 024566                                    BREAK
3459
3460 024570  013704  012410                   MOV     LENGTH,R4               ;CLEAR RX TABLE
3461 024574  012702  006362                   MOV     #RTABLE,R2
3462 024600  005022                   20$:    CLR     (R2)+
3463 024602  005304                           DEC     R4
3464 024604  001375                           BNE     20$
3465
3466
3467
3468
3469
3470
3471 024606  013777  012406  165650           MOV     TSPEED,@KMVP14          ;SEND TX SPEED
3472 024614  012777  002362  165630           MOV     #TTABLE,@KMVP02         ;  "  TX TABLE ADDRESS
3473 024622  013777  012410  165624           MOV     LENGTH,@KMVP04          ;  "  "     "   LENGTH
3474 024630  012777  006362  165624           MOV     #RTABLE,@KMVP12         ;SEND RX TABLE ADDRESS
3475 024636  005077  165614                   CLR     @KMVP06                 ;CLR EXTENDED ADDRESS
3476
3477
3478
3479 024642  004537  014556                   JSR     R5,TSTNUB
3480 024646  000040                           .WORD   40                      ;DO TEST 40= CHA TEST
3481
3482 024650                                    WAITB   0,2                     ;WAIT FOR TEST EXECUTION
3483
3484
3485 024670  004737  013074                   JSR     PC,TSTERR               ;CHECK BSEL0
3486
3487 024674  000427                           BR      6$                      ;TEST OK CHECK RX TABLE
3488 024676  000402                           BR      3$                      ;TIMEOUT ERROR
3489 024700  000401                           BR      3$                      ;NO KMV11 ANSWER
3490 024702  000410                           BR      4$                      ;CHECK SEL16 TO SEE WHICH ONE
3491
3492
3493 024704                          3$:     ERRHRD  28,EM0004               ;NO KMV11 ANSWER
3494 024714  004737  012732                   JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
3495 024720                                    ESCAPE  TST
3496
```

```
3497 024724                          4$:                                ;ERROR DURING TEST; READ ERROR STATUS
3498                                                                    ;TO SEE WHICH ONE
3499
3500 024724  017737  165536  012416          MOV     @KMVP16,STAERR     ;READ ERROR STATUS
3501
3502 024732  017737  165522  012420          MOV     @KMVP10,WRDCNT     ;READ WORD COUNT DISCREPANCY
3503
3504 024740                          ERRHRD  29,EM0022,PRSTER                ;ERROR WHILE TX,RX FRAMES,GIVE ERROR
3505                                                                    ;GIVE ERROR STATUS,WORD CNT DISCREPANCY
3506 024750                          ESCAPE  TST
3507
3508
3509
3510
3511 024754  012702  002362          6$:     MOV     #TTABLE,R2         ;LOAD TX TABLE ADDRESS
3512 024760  012705  006362                  MOV     #RTABLE,R5         ; ''   RX    ''     ''
3513 024764  013704  012410                  MOV     LENGTH,R4          ; ''   TX TABLE LENGTH
3514
3515
3516 024770  022522                  RXLCK:  CMP     (R5)+,(R2)+        ;CMP TX AND RX TABLE
3517 024772  001015                          BNE     RXLERR             ;BR IF ERROR
3518 024774  005304                          DEC     R4                 ;ALL DONE
3519 024776  001374                          BNE     RXLCK              ;NO
3520
3521 025000  062737  000400  012410          ADD     #400,LENGTH        ;CHANGE LENGTH
3522 025006  022737  002000  012410          CMP     #2000,LENGTH       ;IS IT MAX?
3523 025014  100252                          BPL     TXLBGN             ;NO DO TEST AGAIN WHITH NEW TABLE
3524                                                                    ;                          LENGTH
3525
3526 025016  005303                          DEC     R3                 ;SELECT OTHER PATERNS
3527 025020  001245                          BNE     TXLTAR
3528
3529 025022  000137  025144                  JMP     RXLEND
3530
3531
3532
3533 025026  162705  000002          RXLERR: SUB     #2,R5
3534 025032  162702  000002                  SUB     #2,R2
3535
3536 025036  011237  012402                  MOV     (R2),TXDATA
3537 025042  011537  012404                  MOV     (R5),RXDATA
3538
3539 025046  005737  002256                  TST     FLAG               ;LOOK IF 1ST ERROR
3540 025052  001014                          BNE     30$
3541
3542 025054                          ERRHRD  30,EM0016,PFRAME           ;DATA CMP ERROR
3543 025064  005237  002256                  INC     FLAG
3544 025070  062702  000002                  ADD     #2,R2              ;POINT NEXT ADDRESS
3545 025074  062705  000002                  ADD     #2,R5
3546 025100  000137  024344                  JMP     RXCK
3547
3548 025104                          30$:    ERRHRD  30,0,PRAMEF        ;SHORT REPORT
3549 025114  005237  002256                  INC     FLAG
3550 025120  062702  000002                  ADD     #2,R2
3551 025124  062705  000002                  ADD     #2,R5              ;POINT NEXT ADDRESS
3552 025130  022737  000010  002256          CMP     #10,FLAG           ;LOOK IF 10 REPORT
3553 025136  001314                          BNE     RXLCK
```

```
3554
3555 025140                                ESCAPE  TST
3556
3557
3558
3559
3560 025144                            RXLEND:
3561 025144                            ENDTST
```

```
3563
3564 025146                     BADHEAD
                                ;********************** TEST7 ************************
3565                            ;TRANSMIT DIFFERENT FRAMES OF VARIOUS LENGTH IN EXTERNAL LOOP BACK
3566                            ;MODE ON CHANNEL A   AT 72KB
3567 025146                     BADHEAD
                                ;********************** TEST7 ************************
3568
3569
3570
3571
3572
3573
3574 025146                     STARS 1
3575                            ;
3576                            ;AT BEGINNING OF TEST ,CHECK IF LOOP BACK CONNECTORS ARE INSTALLED
3577                            ;OR NOT:IF NOT INSTALLED = EXIT TEST AND GIVE ERROR MESSAGE
3578                            ;                          ***********************************
3579                            ;
3580                            ;
3581                            ;
3582                            ;
3583                            ;
3584                            ;QBUS WRITE DIFFERENT TX TABLE OF VARIOUS LENGTH, LOAD IN KMV11 CSR'S
3585                            ;THE TX AND RX TABLE ADDRESS ,THE TABLE LENGTH AND TRANSFER SPEED (72KB)
3586                            ;
3587                            ;
3588                            ;
3589                            ;DCT11 EXECUTE THE TRANSFER IN EXTERNAL MODE ON CHA  AND WRITE BACK
3590                            ;IN RX TABLE
3591                            ;QBUS CHECK BSEL0 TO SEE THE STATUS OF THE TEST AND IF TEST DONE CHECK IF
3592                            ;RX TABLE =TX TABLE
3593                            ;
3594                            ;
3595                            ;PARAMETERS SELECTION:
3596                            ;       SEL2=   TX TABLE ADDRESS
3597                            ;       SEL4=   TX TABLE LENGTH
3598                            ;       BSEL6=  EXTENDED ADDRESS OF TX TABLE
3599                            ;       BSEL7=      "       "       RX   "
3600                            ;       SEL12=  RX TABLE ADDRESS
3601                            ;       SEL14=  SPEED SELECTION          (=141 IF 72KB)
3602                            ;       BSEL16= ERROR STATUS
3603                            ;       BSEL0=  TEST STATUS
3604                            ;       SEL10=  RECEIVE BYTE COUNT       >0 IF TX>RX
3605                            ;                                        <0 IF TX<RX
3606                            ;
3607                            ;
3608                            ;
3609                            ;TEST STATUS DESCRIPTION:
3610                            ;       BSEL0=  0       =TEST DONE CHECK RX TABLE
3611                            ;       BSEL0=  200     =TIMEOUT ERROR
3612                            ;       BSEL0=  TSTNB   =NO KMV11 ANSWER
3613                            ;       BSEL0=  100     =ERROR DURING TEST ,LOOK WHICH ONE BY TESTING BSEL16
3614                            ;
3615                            ;
3616                            ;
3617                            ;ERROR STATUS DESCRIPTION:
```

```
3618                            ;
3619                            ;       WHEN BSEL0=100,GIVE CONTAINT OF ERROR STATUS AND WORD COUNT DISCREPANCY
3620                            ;
3621                            ;
3622                            ;       BSEL16= BIT14=1 =FCS ERROR
3623                            ;       BSEL16= BIT13=1 =OVERRUN ERROR
3624                            ;       BSEL16= BIT8 =1 =ILLEGAL INTERUPT ERROR
3625                            ;       BSEL16= BIT7 =1 =RX ABORT ERROR
3626                            ;       BSEL16= BIT6 =1 =UNDERRUN ERROR
3627                            ;       BSEL16= BIT5 =1 =BYTE COUNT DISCREPANCY
3628                            ;       BSEL16= BIT4 =1 =DMA IN TIMEOUT ERROR
3629                            ;       BSEL16= BIT3 =1 =DMA OUT TIMEOUT ERROR
3630                            ;       BSEL16= BIT2 =1 =CLOCK PROBLEM
3631                            ;       BSEL16= BIT1 =1 =DATA COMPARE ERROR BETWEEN TX AND RX TABLE (USE
3632                            ;                                                   ONLY DURING SELF TEST)
3633                            ;
3634                            ;
3635                            ;MICRO DIAG TEST DESCRIPTION:
3636                            ;TEST 42        =TRANSMIT VARIOUS LENGTH FRAME AT 72 KBAUDS SPEED ON CHANNEL A
3637                            ;                       IN EXTERNAL LOOP BACK MODE
3638                            ;
3639                            ;
3640                            ;
3641                            ;
3642                            ;CAUTION:
3643                            ;--------
3644                            ;RUN ONLY WITH EXTERNAL LOOP BACK CONNECTOR:
3645                            ;
3646                            ;
3647                            ;NOTE:
3648                            ;
3649                            ;TO FULLY TEST KMV11 DIAGNOSTIC MUST BE RUN WITH RS422 AND RS423
3650                            ;EXTERNAL LOOP BACK CONECTOR
3651                            ;
3652                            ;EXTERNAL LOOP BACK CONNECTOR:
3653                            ;-----------------------------
3654                            ;KMV11 A CAN OPERATE EITHER IN RS422 OR RS 423 LEVEL CONVERTERS
3655                            ;
3656                            ;
3657                            ;RS422 LOOP BACK:
3658                            ;TO TEST COMPLETELY A KMV11 B IN RS422 MODE ,RUN THIS DIAGNOSTIC
3659                            ;WHITH LOOP BACK CONNECTOR  PLUG :
3660                            ;-USE H3255 TO LOOP DIRECTLY AT THE OUTPUT OF THE MODULE
3661                            ;-USE H3251 PLUG AT THE END OF BC55U MODEM CABLE CONNECTOR ASSY.
3662                            ;
3663                            ;
3664                            ;RS423 LOOP BACK:
3665                            ;TO TEST COMPLETELY A KMV11-A IN RS423 MODE ,RUN THIS DIAGNOSTIC
3666                            ;WHITH LOOP BACK CONNECTOR  PLUG :
3667                            ;-USE H3255 TO LOOP AT THE OUTPUT OF THE MODULE
3668                            ;-USE H3251 PLUG AT THE END OF BC55H MODEM CABLE CONNECTOR ASSY.
3669                            ;
3670                            ;
3671                            ;
3672                            ;RS232 LOOP BACK:
3673                            ;SAME AS FOR RS423.
3674                            ;
```

```
                                        ;CAUTION:
                                        ;USE OF H325 LOOP BACK CONNECTOR WILL CAUSE MESSAGES ERROR IN TEST 8.
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
 3683 025146                            STARS 1
```

```
3685 025146                                BGNTST
3686 025146  004737  014372                       JSR     PC,CLRKMV          ;CLEAR REGISTERS
3687 025152  005737  012470                       TST     LOOP               ;IS LOOP BIT=1?
3688 025156  001012                               BNE     BGNTXA             ;YES GO ON TEST
3689 025160                                       PRINTF  #MLOOP             ;NO LOOP BACK CONNECTOR
3690                                                                         ;TEST NOT EXECUTED
3691 025200                                       EXIT    TST
3692
3693
3694 025204  004737  014474          BGNTXA:      JSR     PC,MAINM1          ;SET MAINT MODE
3695 025210  005037  002256                       CLR     FLAG
3696
3697 025214  012703  000005                       MOV     #5,R3              ;SELECT RANDOM PATTERN
3698 025220  012737  000141  012406               MOV     #KB72,TSPEED       ;SELECT SPEED
3699
3700 025226  012737  000001  012410  TXATAR:      MOV     #1,LENGTH          ;1ST TABLE LENGTH(1 WORD)
3701
3702 025234  013704  012410          TXABGN:      MOV     LENGTH,R4
3703 025240                                       BREAK
3704
3705 025242  012702  002362                       MOV     #TTABLE,R2
3706 025246  004737  013152          10$:         JSR     PC,GENER           ;WRITE TABLE
3707 025252  013722  012372                       MOV     DATA,(R2)+
3708 025256  005304                               DEC     R4
3709 025260  001372                               BNE     10$
3710
3711
3712
3713 025262  013704  012410                       MOV     LENGTH,R4          ;CLEAR RX TABLE
3714 025266  012702  006362                       MOV     #RTABLE,R2
3715 025272  005022                  20$:         CLR     (R2)+
3716 025274  005304                               DEC     R4
3717 025276  001375                               BNE     20$
3718
3719
3720
3721
3722
3723
3724
3725 025300  013777  012406  165156               MOV     TSPEED,@KMVP14     ;SEND TX SPEED
3726 025306  012777  002362  165136               MOV     #TTABLE,@KMVP02    ;   "  TX TABLE ADDRESS
3727 025314  013777  012410  165132               MOV     LENGTH,@KMVP04     ;   "  "     "   LENGTH
3728 025322  012777  006362  165132               MOV     #RTABLE,@KMVP12    ;SEND RX TABLE ADDRESS
3729 025330  005077  165122                       CLR     @KMVP06            ;CLR EXTENDED ADDRESS
3730
3731
3732
3733
3734
3735 025334  004537  014556          1$:          JSR     R5,TSTNUB
3736 025340  000042                               .WORD   42                 ;DO TEST 42= CHB TEST
3737
3738
3739
3740 025342                          2$:          WAITB   0,3                ;WAIT FOR TEST EXECUTION
3741
```

```
3742
3743 025362  004737  013074              JSR     PC,TSTERR           ;CHECK BSEL0
3744
3745 025366  000427                      BR      6$                  ;TEST OK CHECK RX TABLE
3746 025370  000402                      BR      3$                  ;TIMEOUT ERROR
3747 025372  000401                      BR      3$                  ;NO KMV11 ANSWER
3748 025374  000410                      BR      4$                  ;CHECK SEL16 TO SEE WH'CH ONE
3749
3750
3751 025376                      3$:     ERRHRD  32,EM0004           ;NO KMV11 ANSWER
3752 025406  004737  012732              JSR     PC,CHKMAX           ;CHECK IF TOO MANY ERROR
3753 025412                              ESCAPE  TST
3754
3755
3756 025416                      4$:                                 ;ERROR DURING TEST READ ERROR STATUS
3757                                                                 ;TO CHECK WHICH ONE
3758
3759 025416  017737  165044  012416      MOV     @KMVP16,STAERR      ;READ ERROR STATUS
3760
3761 025424  017737  165030  012420      MOV     @KMVP10,WRDCNT      ;READ WORD COUNT DISCREPANCY
3762
3763 025432                              ERRHRD  33,EM0022,PRSTER            ;ERROR WHILE TX,RX FRAMES,GIVE ERROR
3764                                                                 ;GIVE FRROR STATUS,WORD CNT DISCREPANCY
3765 025442                              ESCAPE  TST
3766
3767
3768
3769
3770
3771 025446  012702  002362      6$:     MOV     #TTABLE,R2          ;LOAD TABLE PARAMETERS
3772 025452  012705  006362              MOV     #RTABLE,R5
3773 025456  013704  012410              MOV     LENGTH,R4
3774
3775 025462  022225              RXACK:  CMP     (R2)+,(R5)+         ;CHECK TX AND RX TABLE
3776 025464  001015                      BNE     RXAERR
3777 025466  005304                      DEC     R4
3778 025470  001374                      BNE     RXACK
3779
3780 025472  062737  000400  012410      ADD     #400,LENGTH         ;CHANGE LENGTH
3781 025500  022737  002000  012410      CMP     #2000,LENGTH
3782 025506  100252                      BPL     TXABGN
3783 025510  005303                      DEC     R3                  ;SELECT NEW PATERN
3784 025512  001245                      BNE     TXATAR              ;ALL DONE
3785 025514  000137  025636              JMP     RXAEND
3786
3787
3788
3789 025520  162705  000002      RXAERR: SUB     #2,R5
3790 025524  162702  000002              SUB     #2,R2
3791
3792 025530  011237  012402              MOV     (R2),TXDATA
3793 025534  011537  012404              MOV     (R5),RXDATA
3794
3795 025540  005737  002256              TST     FLAG                ;LOOK IF 1ST ERROR
3796 025544  001014                      BNE     30$
3797
3798 025546                              ERRHRD  34,EM0015,PFRAME    ;DATA CMP ERROR
```

```
3799 025556  005237  002256                    INC     FLAG
3800 025562  062702  000002                    ADD     #2,R2                   ;POINT NEXT ADDRESS
3801 025566  062705  000002                    ADD     #2,R5
3802 025572  000137  024770                    JMP     RXLCK
3803
3804 025576                            30$:    ERRHRD  34,0,PRAMEF             ;SHORT REPORT
3805 025606  005237  002256                    INC     FLAG
3806 025612  062702  000002                    ADD     #2,R2
3807 025616  062705  000002                    ADD     #2,R5                   ;POINT NEXT ADDRESS
3808 025622  022737  000010  002256            CMP     #10,FLAG                ;LOOK IF 10 REPORT
3809 025630  001314                            BNE     RXACK
3810
3811 025632                                    ESCAPE  TST
3812
3813
3814
3815
3816
3817 025636                            RXAEND:
3818 025636                            ENDTST
```

```
                                ;
025640                          BADHEAD
                                ;********************* TEST8 ************************
                                ;TEST MODEM SiGNALS IN EXTERNAL LOOP BACK
025640                          BADHEAD
                                ;********************* TEST8 ************************
                                ;
                                ;
                                ;
                                ;
                                ;
                                ;
025640                          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 BSEL0
                                ;IF ERROR SET 100 IN BSEL0 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
                                ;
                                ;
                                ;
```

```
3869
3870                    ;MODEM SIGNAL LINK:
3871                    ;
3872                    ;         SEND                                  RECEIVE
3873                    ;
3874                    ;
3875                    ;
3876                    ;       CCITT 105 (RTS)                       CCITT 106 (CTS)
3877                    ;       ---------------------------------------------------------
3878                    ;                                            /
3879                    ;                                           /
3880                    ;                                          /      CCITT 109 (CD)
3881                    ;                                          ------------------------
3882                    ;
3883                    ;
3884                    ;
3885                    ;       CCITT 108 (DTR)                       CCITT 107 (DSR)
3886                    ;       ---------------------------------------------------------
3887                    ;
3888                    ;
3889                    ;
3890                    ;
3891                    ;       CCITT 111 (DTE)                       CCITT 112 (DCE)
3892                    ;       ---------------------------------------------------------
3893                    ;
3894                    ;
3895                    ;
3896                    ;
3897                    ;       CCITT 141 (LL)                        CCITT 142 (TM)
3898                    ;       ---------------------------------------------------------
3899                    ;
3900                    ;
3901                    ;
3902                    ;
3903                    ;       TIS                                   CCITT 125 (RING)
3904                    ;       ---------------------------------------------------------
3905                    ;
3906                    ;
3907                    ;
3908                    ;
3909                    ;
3910                    ;
3911                    ;CAUTION:
3912                    ;--------
3913                    ;RUN ONLY WITH EXTERNAL LOOP BACK CONNECTOR:
3914                    ;
3915                    ;
3916                    ;
3917                    ;TO BE FULLY TESTED ,KMV11 DIAGNOSTIC MUST BE RUN WITH RS422 AND RS423
3918                    ;EXTERNAL LOOP BACK CONECTOR
3919                    ;
3920                    ;EXTERNAL LOOP BACK CONNECTOR:
3921                    ;-----------------------------
3922                    ;KMV11 A CAN OPERATE EITHER IN RS422 OR RS 423 LEVEL CONVERTERS
3923                    ;
3924                    ;
3925                    ;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.
                                ;
                                ;
                                ;
                                ;
 025640                         $TARS 1
```

```
3953 025640                          BGNTST
3954 025640  004737  014372                  JSR     PC,CLRKMV               ;CLEAR ALL REGISTERS
3955
3956 025644  005737  012470                  TST     LOOP
3957 025650  001012                          BNE     MODSIG                  ;LOOP BACK PRESENT GO ON
3958
3959 025652                                  PRINTF  #MLOOP                  ;NO LOOP BACK CONNECTOR
3960                                                                         ;TEST NOT EXECUTED
3961
3962 025672                                  EXIT    TST                     ;GO TO FOLLOWING TEST
3963
3964
3965
3966 025676  004737  014474          MODSIG: JSR     PC,MAINM1               ;SET MAINTENANCE MODE
3967 025702  004537  014556                  JSR     R5,TSTNUB
3968 025706  000045                          .WORD   45                      ;SEND TEST 45
3969
3970 025710                                  WAITB   0,4
3971
3972 025730  004737  013074                  JSR     PC,TSTERR               ;CHECK TEST RESULT
3973 025734  000430                          BR      3$                      ;TEST OK GO ON
3974 025736  000402                          BR      4$                      ;TIMEOUT
3975 025740  000401                          BR      4$                      ;NO TEST ANSWER
3976 025742  000406                          BR      5$                      ;ERROR DURING TEST ,LOOK WHICH ONE
3977
3978
3979
3980 025744                          4$:     ERRHRD  36,EM0004               ;NO ANSWER
3981 025754                                  ESCAPE  TST
3982
3983 025760  017737  164466  002264  5$:     MOV     @KMVP02,GOOD            ;READ WHICH SIGNAL WAS TESTED
3984 025766  017737  164462  012366          MOV     @KMVP04,BAD             ;   ''      ''   IS THE RESULT OF TEST
3985 025774  017737  164460  012372          MOV     @KMVP10,DATA            ;READ SIGAL VALUE
3986
3987 026002                                  ERRHRD  37,EM0032,PMODEM        ;REPORT ERROR
3988 026012                                  ESCAPE  TST
3989
3990 026016                          3$:
3991 026016                          MODEND:
3992
3993
3994 026016                          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.
                                ;////////////////////////////////////////////////////////////////////////

026020                                  BGNHRD

026022                                  GPRMA   ADDRES,0,0,160000,177776,YES
026032                                  GPRMA   VECTOR,2,0,0,674,YES
026042                                  GPRMD   PRIRTY,4,0,7000,4,7,YES
026054                                  GPRMD   LOOPBK,6,0,1,0,1,YES
026066                                  ENDHRD


026066  115  111  103  ADDRES: .ASCIZ  /MICRO-CPU  CSR ADDRESS : /
026071  122  117  055
026074  103  120  125
026077  040  040  103
026102  123  122  040
026105  101  104  104
026110  122  105  123
026113  123  040  072
026116  040  000
026120  115  111  103  VECTOR: .ASCIZ  /MICRO-CPU VECTOR ADDRESS : /
026123  122  117  055
026126  103  120  125
026131  040  126  105
026134  103  124  117
026137  122  040  101
026142  104  104  122
026145  105  123  123
026150  040  072  040
026153  000
026154  115  111  103  PRIRTY: .ASCIZ  /MICRO-CPU PRIORITY LEVEL : /
026157  122  117  055
026162  103  120  125
026165  040  120  122
026170  111  117  122
026173  111  124  131
026176  040  114  105
026201  126  105  114
026204  040  072  040
```

```
                026207  000
   4034         026210  111     123     040  LOOPBK: .ASCIZ  /IS LOOP BACK CONNECTOR PLUGGED? 0=NO,1=YES: /
                026213  114     117     117
                026216  120     040     102
                026221  101     103     113
                026224  040     103     117
                026227  116     116     105
                026232  103     124     117
                026235  122     040     120
                026240  114     125     107
                026243  107     105     104
                026246  077     040     060
                026251  075     116     117
                026254  054     061     075
                026257  131     105     123
                026262  072     040     000
   4035                                     .EVEN
   4036
   4037
   4038
   4039
   4040
   4041
```

```
                                .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.
                                ;////////////////////////////////////////////////////////////////
                                
026266                                  BGNSFT


                                
026270                                  ENDSFT
```

```
 4079
 4080 026270                            $PATCH::
 4081 026270                                    .BLKW   50
 4082
 4089
 4090 026410                                    LASTAD
      026414                            L$LAST::
 4091 026414                                    ENDMOD
 4092
 4093
```

```
   4095
   4096
   4109
   4110 026414                                  BGNSETUP        1
   4111 026414                                  BGNPTAB
   4112 026420  177000                          .WORD   177000
   4113 026422  000300                          .WORD   300
   4114 026424  004000                          .WORD   4000
   4115 026426  000001                          .WORD   1
   4116 026430                                  ENDPTAB
   4117 026430                                  ENDSETUP
   4118
   4119
   4120
   4121
   4122
   4123         000001                  .END
```

KMV11 A LINE CNT DIAGNOSTIC MACRO M1200 06-JAN-83 09:40 PAGE 74-1
SYMBOL TABLE

```
ABORT   022632          C$CVEC= 000036          EM0002  015115          GENER1  013274          I$DU   = 000041
ADDR    002354          C$DCLN= 000044          EM0003  015160          GENEX   013432          I$HRD  = 000041
ADDRES  026066          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
BDROK0  024052          C$ERRO= 000060          EM0014  015716          GENR1   013236          I$RPT  = 000041
BGNTXA  025204          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$ETST= 000001          EM0024  016402          GETPRM  022430          I$TST  = 000041
BIT04 = 000020 G        C$EXIT= 000032          EM0027  016431          GOOD    002264          J$JMP  = 000167
BIT05 = 0[illegible]0 G  C$GETB= 000026          EM0031  016503          GOOD0   002266          KB1.2  = 013224 G
BIT06 = 0 0100 G        C$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$YES = 000010          LOOPBK   026210
CKSEL0  013612          DATA1 = 052525 G        F$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   024770      T$PCNT= 000000
L$DTYP  002034 G    L10021  023022      O$GNSW= 000001      RXLEND  025144      T$PTAB= 010037
L$DU    023024 G    L10022  023102      O$POIN= 000001      RXLERR  025026      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$EXP1  002046 G    L10031  025144      PRALL   020716 G    SEL0    002310      T$TAGN= 010041
L$EXP4  002064 G    L10032  025636      PRAMEF  022120 G    SEL1    002312      T$TEMP= 000000
L$EXP5  002066 G    L10033  026016      PRI   = 002000 G    SEL10   002322      T$TEST= 000010
L$HARD  026022 G    L10034  026066      PRINT   020656 G    SEL12   002324      T$TSTM= 177777
L$HIME  002120 G    L10035  026270      PRIRTY  026154      SEL14   002326      T$TSTS= 000001
L$HPCP  002016 G    L10036  026420      PRI00 = 000000 G    SEL16   002330      T$$AU = 010023
L$HPTP  002022 G    L10040  026430      PRI01 = 000040 G    SEL2    002314      T$$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   022346      T$$DU = 010022
L$LADP  002026 G    MAXERR  002232      PRI05 = 000240 G    SSTACK  012672      T$$HAR= 010034
L$LAST  026414 G    MAXPRI= 000340 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$SOFT  026270 G    MODEND  026016      RANEX   013422      TSTERR  013074      T3      023274 G
L$SPC   002056 G    MODSIG  025676      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  025234      T6      024506 G
L$STA   002030 G    MREG10  017336      RANSEL  002336      TXATAR  025226      T7      025146 G
L$SW    002236      MREG12  017402      RANST   002334      TXDATA  012402      T8      025640 G
L$TEST  002114 G    MREG14  017446      RAN1    013336      TXLBGN  024542      UAM   = 000200 G
L$TIML  002014 G    MREG16  017512      RAN2    013354      TXLTAR  024534      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  026120
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= 026430      WRITE   014604
L10006  021542      MVECT   017654      RTCLK   023304      T$GMAN= 000000      X$ALWA= 000000
L10007  021602      NERRS   013024      RUNNIN  022640      T$HILI= 000001      X$FALS= 000040
L10010  021642      NEXT    022354      RXACK   025462      T$LAST= 000001      X$OFFS= 000400
L10011  021702      NUB     012412      RXAEND  025636      T$LOLI= 000000      X$TRUE= 000020
L10012  021772      NUMBER  002352      RXAERR  025520      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      $PATCH  026270 G
L10015  022232      O$BGNR= 000000      RXDATA  012404
```

SYMBOL TABLE

```
. ABS.  026430     000
        000000     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  29056 WORDS  ( 114 PAGES)
DYNAMIC MEMORY:  21924 WORDS  ( 84 PAGES)
ELAPSED TIME:  00:24:20
VKMBA0.BIN,VKMBA0=[64,3]LIBA.MLB/ML,[64,5]VKMBA0
```