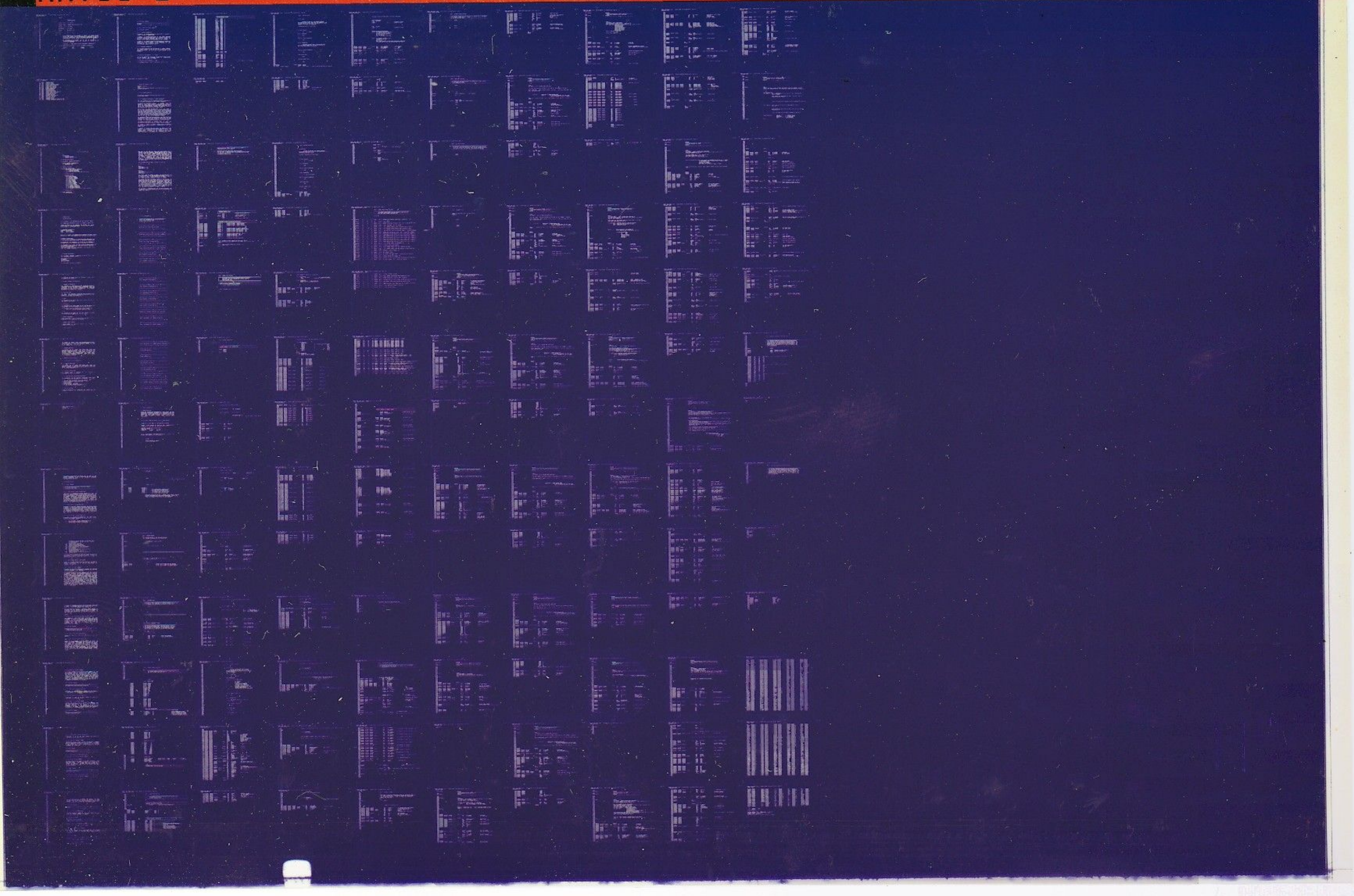
KMV11-A,
KMV11-B

LOGIC DIAG
CVKMAAO

AH-T370A-MC
FICHE 1 OF 1



```
                        .REM @

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

                        PRODUCT CODE:   AC-T369A-MC

                        PRODUCT NAME:   CVKMAA0 KMV11A/B LOGIC DIAG

                        PRODUCT DATE:   JAN 1983

                        MAINTAINER:     CSS ANNECY

                        AUTHOR:         MICHELET, GUY
```

TABLE OF CONTENTS

TABLE OF CONTENTS

```
                                1.0 INTRODUCTION

                                1.1 PROGRAM ABSTRACT


                                THIS  DIAGNOST.  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 PRIORITY LEVEL.


                                1.2 HARDWARE INTRODUCTION
                                HARDWARE DESCRIPTION:
                                M7500   = KMV11-A MODULE
                                M7501   = KMV11-B MODULE


                                KMV11-A IS A SINGLE LINE COMMUNICATION CONTROLER FOR QBUS SYSTEMS
                                KMV11-B IS A DUAL LINE COMMUNICATION CONTROLER FOR QBUS SYSTEMS


                                DIAGNOSTIC DECRIPTION:
                                THE KMV11 STATIC DIAGNOSTIC IS COMPATIBLE WITH BOTH KMV11 A/B
                                IT WILL RUN IN STAND ALONE WITHOUT ANY OPERATOR INTERVENTIONS

                                THE PURPOSE OF THIS DIAGNOSTIC IS TO TEST ALL THE HARDWARE OF
                                THE QBUS PART OF THE INTERFACE AND THE RAM PART OF THE KMV11.

                                THIS DIAGNOSTIC WILL FIRST TEST QBUS ACCESS ON KMV11A(M7500) AND
                                KMV11B(M7501) CSR'S REGISTERS, THEN DATA TRANSFER FROM QBUS
                                TO DCT11 MICROPROCESSOR.
                                AFTER THAT IT WILL TEST KMV11 RAM MEMORY, DMA TRANSFERS IN/OUT
                                KMV11 AND INTERUPT CAPABILITY.


                                2.0 HARDWARE REQUIREMENTS

                                THE  FOLLOWING  HARDWARE  IS REQUIRED TO RUN THE STATIC LOGIC
                                TESTS ON MODULES M7500 OR M7501:

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

                                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 M7500 OR M7501 STATIC DIAGNOSTIC IS ABOUT 120 SECONDS PER PASS FOR EACH UNIT (ON PDP11/23A WITH SUPERVISOR VERSION C4).

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 (FOR THAT DIAGNOSTIC MUST BE SETUP FIRST).

CAUTION:UNDER SLIDE THE OPERATOR MUST ALWAYS ANSWER "YES"
(THE FIRST TIME)FOR HARDWARE PRAMETERS CHANGE.

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 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
                                VKMAA0-A-0
                                KMV11A/B LOGIC 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
                                BOE BELL ON ERROR
                                UAM RUN IN UNATTENDED MODE, BYPASSING MANUAL
                                    INTERVENTION TESTS
                                ISR INHIBIT STATISTICAL REPORTS
                                IDU INHIBIT DROPPING OF UNITS BY DIAGNOSTIC
                                LOT LOOP ON TEST

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


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

                            <INCR> IS A DECIMAL NUMBER INDICATING HOW OFTEN (IN TERMS OF
                            PASSES) IT IS DESIRED  THAT  THE  END  OF  PASS  MESSAGE  BE
                            PRINTED.  THE  DEFAULT  IS  AT  THE END OF EVERY PASS.  SEE
                            EXAMPLE AT END OF 6.3.1.5.


                            6.3.1.5 EFFECT OF START COMMAND

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

                            THE HARDWARE PARAMETER DIALOGUE COMMENCES WITH THE  QUESTION
                            "#  UNITS?"  TO  WHICH  THE  OPERATOR REPLIES WITH A DECIMAL
                            NUMBER N FROM 1 TO 16. THE TERM "UNIT" REFERS TO THE  DEVICE
                            TO WHICH THIS SERIES OF DIAGNOSTICS IS DEDICATED.  FOLLOWING
                            THIS ARE THE QUESTIONS WHEREBY THE P-TABLES THEMSELVES  WILL
                            BE  BUILT.  EACH P-TABLE IS A CORE-RESIDENT TABLE CONTAINING
                            ALL THE HARDWARE INFORMATION FOR  ONE  UNIT.   THE  OPERATOR
                            MUST  SUPPLY  N  (NUMBER OF UNITS) VALUES FOR EACH QUESTION.
                            HE MAY DO THIS BY GIVING ONE ANSWER  TO  EACH  QUESTION  (IN
                            WHICH CASE THE SERIES OF QUESTIONS WILL BE POSED N TIMES) OR
                            BY GIVING N VALUES, SEPARATED BY COMMAS,  TO  EACH  QUESTION
                            (SERIES  WILL  BE POSED ONCE).  EACH QUESTION IS FOLLOWED BY
                            THE RESPONSE RADIX (D FOR  DECIMAL,  B  FOR  BINARY,  O  FOR
                            OCTAL,  L  FOR  YES/NO) IN PARENTHESES AND THE DEFAULT VALUE
                            AFTER THE PARENTHESES.
```

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

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

EXAMPLE:

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

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAP 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 FPOM 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>
****

```
495
496
497
498
499
500                             6.3.4.1 FLAGS SWITCH (/FLAGS:<FLAG-LIST>)
501
502                             <FLAG-LIST>  IS  AS  IN  THE  START COMMAND, BUT UNSPECIFIED
503                             FLAGS RETAIN THEIR CURRENT VALUE.
504
505
506                             6.3.4.2 EFFECT OF PROCEED COMMAND
507
508                             PROCEED MUST FOLLOW A START, RESTART, OR CONTINUE.   COMMAND
509                             MODE MUST HAVE BEEN ENTERED VIA A HALT ON ERROR.  THE EFFECT
510                             OF THE  COMMAND  IS  TO  BEGIN  EXECUTION  AT  THE  LOCATION
511                             FOLLOWING  THE  ERROR  CALL.   NEITHER HARDWARE NOR SOFTWARE
512                             PARAMETERS MAY BE ALTERED.
513
514
515                             6.3.5 ADD COMMAND
516
517                             ****************************************************************
518                             ADD/UNITS:<UNIT-LIST>
519                             ****************************************************************
520
521
522                             6.3.5.1 UNITS SWITCH (/UNITS:<UNIT-LIST>
523
524                             <UNIT-LIST> IS AS IN THE RESTART COMMAND.
525
526
527                             6.3.5.2 EFFECT OF ADD COMMAND
528
529                             THE UNITS SPECIFIED ARE ADDED TO THE  TEST  SEQUENCE.   EACH
530                             UNIT  MUST  HAVE  A  P-TABLE  IN  MEMORY  DUE  TO AN EARLIER
531                             HARDWARE DIALOGUE.  THIS  COMMAND  MUST  BE  FOLLOWED  BY  A
532                             RESTART  OR  CONTINUE.   THE UNITS SWITCH MUST BE SPECIFIED.
533                             THE ADD COMMAND IS  MEANINGFUL  ONLY  FOR  UNITS  THAT  WERE
534                             PREVIOUSLY DROPPED.
535
536
537                             6.3.6 DROP COMMAND
538
539                             ****************************************************************
540                             DRO(P)/UNITS:<UNIT-LIST>
541                             ****************************************************************
542
543
544                             6.3.6.1 UNITS SWITCH (/UNITS:<UNIT-LIST>)
545
546                             <UNIT-LIST> IS AS IN THE RESTART COMMAND.
547
548
549                             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?

DEFFAULT VALYE IS 4
NOTE:
M7500 AND M7501 MODULE MOUNTED WITH DC003 CHIPS CAN ONLY
INTERUPT ON LEVEL 4

6.3.13 SOFTWARE PARAMETERS

NO SOFTWARE PARAMETER QUESTIONS ARE ASKED IN THAT
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 OFERATOR.

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

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

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

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
# UNITS (D) ? 16

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

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

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

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

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

```
                                7.0  TEST DESCRIPTIONS


                        ********************* TEST 1 *********************
                        *VERIFY THAT REFERENCED QBUS DEVICE REGISTERS
                        *DO NOT CAUSE TIME OUT TRAP
                        ****************************************************



                        ********************** TEST 2 ********************
                        *
                        *CLEAR ALL KMV11 REGISTERS AND CHECK
                        *
                        ****************************************************



                        ********************** TEST 3 ********************
                        *
                        *CHECK QBUS ACCESS ON KMV11 REGISTERS (FROM SEL2 TO SEL16)
                        *
                        ****************************************************



                        ********************** TEST 4 ********************
                        *
                        *CHECK Q BUS ACCESS ON REGISTER SEL0
                        *
                        ****************************************************

                        ********************** TEST 5 ********************
                        *
                        *CHECK Q BUS BYTE ACCESS ON  ALL KMV11 REGISTERS
                        *
                        ****************************************************



                        ********************** TEST 6 ********************
                        *
                        *DATA TRANSFER TO REGISTER SEL 2
                        *
                        ****************************************************



                        ********************** TEST 7 ********************
                        *
                        *DATA TRANSFER TO REGISTER SEL 4
                        *
                        ****************************************************
```

```
                        ********************** TEST 8 ********************
                        *
                        *DATA TRANSFER TO REGISTER SEL 6
                        *
                        ************************************************


                        ********************** TEST 9 ********************
                        *
                        *DATA TRANSFER TO REGISTER SEL 10
                        *
                        ************************************************


                        ********************** TEST 10 *******************
                        *
                        *DATA TRANSFER TO REGISTER SEL 12
                        *
                        ************************************************


                        ********************** TEST 11 *******************
                        *
                        *DATA TRANSFER TO REGISTER SEL 14
                        *
                        ************************************************


                        ********************** TEST 12 *******************
                        *
                        *DATA TRANSFER TO REGISTER SEL 16
                        *
                        ************************************************


                        ********************** TEST 13 *******************
                        *
                        *CHECK DATA TRASFERS USING ALL CSR'S REGISTERS
                        *
                        ************************************************


                        ********************** TEST 14 *******************
                        *
                        *KMV11 RAM MEMORY TEST: MEMORY PATERN TEST
                        *
                        ************************************************


                        ********************** TEST 15 *******************
                        *
                        *KMV11 RAM MEMORY TEST: MEMORY ADDRESS TEST
                        *
                        ************************************************
```

```
                        ********************** TEST 16 ********************
                        *
                        *KMV11 RAM MEMORY TEST: MEMORY ADDRESS COMPLEMENT TEST
                        *
                        ***************************************************


                        ********************** TEST 17 ********************
                        *
                        *CHECK PROM REVISION
                        *
                        ***************************************************


                        ********************** TEST 18 ********************
                        *
                        *PROM CHECKSUM TEST
                        *
                        ***************************************************


                        ********************** TEST 19 ********************
                        *
                        *DMA TRANSFER INTO KMV11
                        *
                        ***************************************************


                        ********************** TEST 20 ********************
                        *
                        *TEST DMA TRANSFERS OUT KMV11
                        *
                        ***************************************************


                        ********************** TEST 21 ********************
                        *
                        *TEST DMA TRANSFERS IN BOTH DIRECTION
                        *
                        ***************************************************


                        ********************** TEST 22 ********************
                        *
                        *TEST INTERUPT CAPABILITY OF KMV11 MODULE ON QBUS
                        *
                        ***************************************************


                        ********************** TEST 23 ********************
                        *
                        *TEST INTERUPT ON DCT11 MICROPROCESSOR
                        *
                        ***************************************************
```

```
                                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.


                                THE FOLLOWING EXAMPLES PROVIDE TYPICAL ERROR REPORTS:

                                ;CZDMQ DVC FTL ERR 00045 TST 027 SUB 000 PC:022572

                                ;MASTER   CLEAR  FAILED  TO  CLEAR  PC  REG,  CONTENTS=000624
                                ;CZDMQ DVC FTL ERR 00015 TST 042 SUB 000 PC:027234

                                ;UNIT=00, FAILING UNIT ADDRESS=160170
                                ;JUMP TEST ERROR
                                ;FROM ADDR      TO ADDR       BAD ADDR
                                ;000402         000000        000114


                                FOR ALL OTHER ERRORS, THE REPORT MAY BE MORE  EXTENSIVE  AND
                                REQUIRE ADDITIONAL DATA TO BE REPORTED.


                                9.0     HISTORY

                                   -    DESIGN STARTED ON JANUARY 82
                                   -    REVIEW ON DECEMBER 82

                                @
```

```
                                .TITLE KMV11 A/B LOGIC DIAG
002000                          .=2000



                                .MCALL  SVC
002000                          SVC                                     ; INITIALIZE SUPERVISOR MACROS


002000                          BGNMOD  KMV11A.B

000000                  $LSTIN= 0
000000                  $LSTTAG= 0
177777                  SVCINS= -1      ; LIST INSTRUCTIONS, SHIFTED RIGHT
177777                  SVCTST= -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  VKMAA0,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 23



                                ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX



                                .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
002210                                  BGNHW   DFPTBL


002212  177000                          .WORD   177000          ;KMV11,CSRS ADDRESS
002214  000300                          .WORD   300             ;KMV11, VECTOR ADDRESS
002216  004000                          .WORD   4000            ;INTERRUPT PRIORITY LEVEL (4)
002220  000001                          .WORD   1

002222                                  ENDHW
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

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

002222                                  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                          HUE==   100000
1207
1208
1209    000340                  MAXPRI==340
1210
1211    054000                  MAINT0==54000           ;MASTER CLEAR = 1,MODE = 1 ,MAINT 1 = 1 ,T11=HOLD
1212    044000                  MAINT1==44000           ;MASTER CLEAR = 1,MODE = 0 ,MAINT 1 = 0 ,T11=NOT HOLD
1213    040000                  MCLR==40000
1214    052525                  DATA1== 052525
1215    125252                  DATA2== 125252
1216
1217
1218                            ;;*****************************************************************************
1219                            ;* PROGRAM EVENT FLAG DEFINITIONS
1220                            ;;*****************************************************************************
1221
1222
1223
```

```
                                .SBTTL  GLOBAL DATA SECTION

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



                                ;;******************************************************************************
                                ;* STORAGE FOR DEVICE REGISTERS
                                ;;******************************************************************************
002222                                  DESCRIPT        <KMV11A.B  LOGIC DIAGNOSTIC>



002256                                  ERRTBL
002256  000000                  ERRTYP::        .WORD   0
002260  000000                  ERRNBR::        .WORD   0
002262  000000                  ERRMSG::        .WORD   0
002264  000000                  ERRBLK::        .WORD   0


                                ;******************************************************************************
                                ;* PROGRAM CONTROL PARAMETERS
                                ;******************************************************************************
002266  000000                  L$SW:   .WORD   0
002270  000015                  L$UIT:  .WORD   15
002272  000000                  UNIT:   .WORD   0
002274  000000                  LOCK:   .WORD   0               ;ADDRESS FOR LOCK CURRENT DATA
002276  000005                  MAXERR: .WORD   5               ;MAX ERROR CNT BEFORE DROPPING UNIT
002300  000000                  ERRCNT: .WORD   0               ;ERROR CNT
002302  000000                  LOGDEV: .WORD   0               ;LOGICAL DEVICE NUMBER
002304  000000                  PSTACK: .WORD   0               ;BASE LEVEL PROGRAM STACK POINTER
002306  000000                  SAVSP:  .WORD   0               ;STACK POINTER STORAGE
002310  000000                  SAVPC:  .WORD   0               ;PROGRAM COUNTER STORAGE
002312  000000                  SAVE4:  .WORD   0
002314  000000                  SAVE6:  .WORD   0
002316  000000                  FTIME:  .WORD   0
```

```
1281                                    ;*****************************************************************************
1282                                    ;* MISCELLANEOUS STORAGE
1283                                    ;*****************************************************************************
1284 002320  000000                     FLAG:    .WORD   0
1285 002322  000000                     DH1:     .WORD   0
1286 002324  000000                     DELCT1:  .WORD   0
1287 002326  000000                     DELCT2:  .WORD   0
1288 002330  000000                     GOOD:    .WORD   0
1289 002332  000000                     GOOD0:   .WORD   0
1290 002334  000000                     GOOD1:   .WORD   0
1291 002336  000000                     GOOD2:   .WORD   0
1292 002340  000000                     GOOD4:   .WORD   0
1293 002342  000000                     GOOD6:   .WORD   0
1294 002344  000000                     GOOD10:  .WORD   0
1295 002346  000000                     GOOD12:  .WORD   0
1296 002350  000000                     GOOD14:  .WORD   0
1297 002352  000000                     GOOD16:  .WORD   0
1298 002354  000000                     SEL0:    .WORD   0
1299 002356  000000                     SEL1:    .WORD   0
1300 002360  000000                     SEL2:    .WORD   0
1301 002362  000000                     SEL4:    .WORD   0
1302 002364  000000                     SEL6:    .WORD   0
1303 002366  000000                     SEL10:   .WORD   0
1304 002370  000000                     SEL12:   .WORD   0
1305 002372  000000                     SEL14:   .WORD   0
1306 002374  000000                     SEL16:   .WORD   0
1307 002376  000000                     BSEL1:   .WORD   0
1308 002400  000000                     RANST:   .WORD   0
1309 002402  000000                     RANSEL:  .WORD   0
1310 002404  000000                     RANMTA:  .WORD   0
1311 002406  000000                     RANDN:   .WORD   0
1312 002410  000000                     SAVPC1:  .WORD   0
1313 002412  000000                     SAVSTA:  .WORD   0
1314 002414  000000                     COUNT:   .WORD   0
1315 002416  000000                     NUMBER:  .WORD   0
1316 002420  000000                     ADDR:    .WORD   0
1317 002422  000000                     GDDAT:   .WORD   0
1318 002424  000000                     BDDAT:   .WORD   0
1319
1320 002426                             TTABLE:  .BLKW   2000
1321 006426                             RTABLE:  .BLKW   2000
1322
1323 012426  000000                     EXADDR:  .WORD   0
1324 012430  000000                     INTFLG:  .WORD   0
1325 012432  000000                     BAD:     .WORD   0
1326 012434  000000                     BSEL0:   .WORD   0
1327 012436  000000                     DATA:    .WORD   0
1328 012440  000000                     VECT:    .WORD   0
1329
1330 012442  000000                     KIND:    .WORD   0                      ;=0 IF KMV11A ,=1 IF KMV11B
1331 012444  000000                     CHANEL:  .WORD   0
1332
1333 012446  000000                     TXDATA:  .WORD   0
1334 012450  000000                     RXDATA:  .WORD   0
1335 012452  000000                     TSPEED:  .WORD   0
1336 012454  000000                     LENGTH:  .WORD   0
1337 012456  000000                     NUB:     .WORD   0
```

GLOBAL DATA SECTION

```
1338 012460  000000                RXCNT:  .WORD   0
1339 012462  000000                MAXCNT: .WORD   0
1340
```

```
1342
1343                                    ;*****************************************************************
1344                                    ;LOAD IN LOCATION "GDREV" THE PROM VERSION NUMBER THAT IS       *
1345                                    ;COMPATIBLE WHITH THIS DIAGNOSTIC                              *
1346                                    ;                                                              *
1347                                    ;EACH PROM CONTAIN A REV LEVEL AND A ECO LEVEL:                *
1348                                    ;THE REV LEVEL IS MODIFIED EACH TIME A MODIFICATION IS DONE    *
1349                                    ;THE ECO LEVEL IS MODIFIED WHEN THE PROM MODIFICATION NEED     *
1350                                    ;A DIAGNOSTIC MODIFICATION                                     *
1351                                    ;*****************************************************************
1352
1353
1354
1355
1356
1357
1358 012464  000001                    GDREV:  .WORD   1
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
```

```
1370                            ;*****************************************************************************
1371                            ;* PROGRAM CONTROL FLAGS
1372                            ;*****************************************************************************
1373 012466     000            INIFLG: .BYTE   0               ;PROGRAM INITIALIZING FLAG
1374                                    .EVEN
1375 012470     000            LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
1376 012471     000            QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG
1377                                    .EVEN
1378 012472  000000            UUT:    .WORD   0               ;CURRENT UNIT UNDER TEST
1379
1380
1381                            ;*****************************************************************************
1382                            ;* POINTERS TO KMV11 VECTORS AND REGISTERS
1383                            ;*****************************************************************************
1384 012474  000000            KMVV00: 0       ;POINTER TO KMV11  INTRPT VECTOR 0
1385 012476  000000            KMVLVL: 0       ;POINTER TO KMV11  INTRPT SERVICE
1386 012500  000000            KMVV04: 0       ;POINTER TO KMV11  INTRPT VECTOR 04
1387 012502  000000            KMVV02: 0       ;    ''       ''       ''       ''     02
1388 012504  000000            KMVV06: 0       ;    ''       ''       ''       :     06
1389 012506  000000            KMTLVL: 0       ;POINTER TO KMV11 TX INTRPT SERVICE PS
1390 012510  000000            KMVCSR: 0       ;POINTER TO KMV11 CONTROL STATUS REGISTER
1391 012512  000000            KMVP02: 0       ;POINTER TO KMV11 PORT REGISTER - SEL2
1392 012514  000000            KMVP04: 0       ;POINTER TO KMV11 PORT REGISTER - SEL4
1393 012516  000000            KMVP06: 0       ;POINTER TO KMV11 PORT REGISTER - SEL6
1394
1395 012520  000000            KMVP10: 0       ;POINTER TO KMV11 PORT REG  -SEL10
1396 012522  000000            KMVP12: 0       ;POINTER TO PORT REG  -SEL 14
1397 012524  000000            KMVP14: 0       ;POINTER TO PORT REG  -SEL14
1398 012526  000000            KMVP16: 0       ;POINTER TO PORT REG 16
1399
1400 012530  000000            LOOP:   0       ;POINTER TO LOOP BACK CONNECTOR
1401
1402
1403                            ;;****  PRIMARY REG ADRS STORAGE FOR THIS UNIT  ******
1404                            ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
1405 012532                     REGADR:
1406
1407                            ;;****  STACK USED FOR SUBROUTINE LINKAGE  *****
1408 012532                             .BLKW   100
1409 012732                     SSTACK:
1410
1411
1412
1413
1414
1415
1416
```

```
   1418                                 .SBTTL  GLOBAL TEXT SECTION
   1419
   1420                                 ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   1421                                 ;%      THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
   1422                                 ;%      MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
   1423                                 ;%      MORE THAN ONE TEST.
   1424                                 ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
   1425
   1426                                 ;;****************************************************************************
   1427                                 ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
   1428                                 ;;****************************************************************************
   1429 012732                                  DEVTYP  <M7500 OR M7501 MODULE>
   1430
   1431
   1432                                 ;
   1433                                 ; FORMAT STATEMENTS USED IN PRINT CALLS
   1434                                 ;
   1435
   1442
   1443
   1444
   1445
   1446
```

```
                .SBTTL  GLOBAL SUBROUTINES





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

                .MACRO  CLRMAR
                        ROMCLK
                        004000
                .ENDM   CLRMAR
```

```
                                ;ROUTINE TO WAIT FOR EVENT OR TIMECJT


                                ;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


012760  005237  002324         WAIT2:  INC     DELCT1
012764  001375                         BNE     WAIT2

012766                                  BREAK

012770  005337  002326                 DEC     DELCT2
012774  001371                         BNE     WAIT2

012776  000207                         RTS     PC


013000  005237  002324         WAIT1:  INC     DELCT1
013004  001375                         BNE     WAIT1

013006  000207                         RTS     PC
```

```
1503                                    ;MACRO TO WAIT A FEW MS
1504
1505
1506                                    ;CALLING SEQUENCE:      WAITA   X               0<X<177777
1507                                    ;                       WAITB   X,Y             0<X OR Y<177777
1508
1509
1510
1511                                    .MACRO  WAITA   X
1512                                            MOV     #X,DELCT1               ;LOAD COUNT
1513                                            JSR     PC,WAIT1                ;WAIT
1514                                    .ENDM
1515
1516
1517
1518
1519
1520
1521
1522                                    .MACRO  WAITB   X,Y
1523                                            MOV     #X,DELCT1
1524                                            MOV     #Y,DELCT2
1525                                            JSR     PC,WAIT2
1526                                    .ENDM
1527
1528
1529
1530
```

```
1532                                    ;ROUTINE TO DROP UNIT AFTER 5 ERROR
1533
1534
1535                                    ;JSR   PC,CHKMAX
1536
1537
1538
1539
1540
1541
1542
1543
1544 013010                             CHKMAX: INLOOP                          ;LOOPING ON ERROR?
1545 013012                                     BCOMPLETE       1$              ;IF YES, EXIT
1546
1547
1548 013014                                     RFLAGS  R0                      ;GET OPERATOR FLAG
1549 013016  032700  000040                     BIT     #IDU,R0                 ;IS DROPPING INHIBITTED?
1550 013022  001026                             BNE     1$                      ;IF YES EXIT
1551
1552
1553 013024  005237  002300                     INC     ERRCNT                  ;UPDATE ERROR COUNT
1554 013030  023737  002300  002276             CMP     ERRCNT,MAXERR           ;TOO MANY ERROR?
1555 013036  003420                             BLE     1$                      ;IF NOT JUMP
1556
1557
1558 013040                                     PRINTF  #NERRS,MAXERR,UUT       ;TOO MANY ERROR!
1559 013070                                     DODU    UUT                     ;DROP UNIT
1560
1561 013076                                     DOCLN                           ;END THE SUBPASS
1562
1563 013100  000207                     1$:     RTS     PC
1564
1565
1566
1567
1568                                            .NLIST  BEX
1569 013102      045     116     045    NERRS:  .ASCIZ  /%N%AMORE THAN %D3%A  ERRORS ON UNIT %D2/
1570                                            .LIST   BEX
1571                                            .EVEN
1572
1573
1574
```

```
                                ;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
                                ;
                                ;
                                ;
                                ;
013152  004537  013722  TSTERR: JSR     R5,CBSEL0               ;LOOK IF BSEL0=0
013156  000000                 .WORD   0
013160  000411                 BR      1$                      ;TEST IS OK ,RTS PC
                                ;
013162  122737  000200  012434  CMPB    #200,BSEL0              ;LOOK IF BSEL0=200
013170  001406                 BEQ     2$                      ;TIMEOUT DURING TEST,RTS PC+2
                                ;
013172  122737  000100  012434  CMPB    #100,BSEL0              ;LOOK IF BSEL0=100
013200  001405                 BEQ     3$                      ;DATA CMP ERROR,RTS PC+6
                                ;
                                ;
013202  000407                 BR      4$                      ;NO KMV11 ANSWER ,RTS PC+4
                                ;
                                ;
013204  000207          1$:    RTS     PC                      ;TEST OK
                                ;
013206  062716  000002  2$:    ADD     #2,(SP)
013212  000207                 RTS     PC                      ;TIMEOUT ERROR
                                ;
013214  062716  000006  3$:    ADD     #6,(SP)
013220  000207                 RTS     PC                      ;DATA CMP ERROR
                                ;
013222  062716  000004  4$:    ADD     #4,(SP)
013226  000207                 RTS     PC                      ;NO KMV11 ANSWER
```

```
1629                                    .SBTTL  NUMBER GENERATOR
1630
1631
1632
1633                                    DESCRIPTION:
1634
1635                                            ROUTINE TO GENERATE DATA PATTERNS,
1636                                    THE TYPE OF PATTERN IS SELECTED BY R3, AND THE
1637                                    PATTERN GENERATED IS RETURNED IN LOCATION "DATA"
1638                                    AND LOCATION "GOOD"
1639
1640                                    CALLING SEQUENCE:
1641
1642                                            JSR     PC,GENER
1643
1644                                    INPUT PARAMETERS:
1645
1646                                    R3 CONTAINS THE PATTERN NUMBER
1647
1648                                    R3=0            ALL ZEROES
1649                                    1               ALL ONES
1650                                    2               010101 ETC BIT PATTERN
1651                                    3               101010 ETC BIT PATTERN
1652                                    4               ROTATING 1 IN A ZERO WORD
1653                                    5               ROTATING 0 IN AN ALL ONE WORD
1654                                    6               PSEUDO RANDOM NUMBER
1655                                    7               INCREMENTING DATA PATTERN, GOOD
1656                                                    CONTAINS THE VALUE TO BE UPDATED
1657
1658
1659
1660
1661                                    IMPLICIT INPUT PARAMETERS:
1662
1663                                            NONE
1664
1665
1666                                    OUTPUT PARAMETERS:
1667
1668                                            THE NUMBER GENERATED IS HELD IN
1669                                    DATA AND GOOD.
1670
1671
1672                                    IMPLICIT OUTPUT PARAMETERS:
1673
1674                                            NONE
1675
1676
1677                                    COMPLETION CODES:
1678
1679                                            NONE
1680
1681
1682                                    POSSIBLE ERROR CODES:
1683
1684                                            NONE
1685
```

```
                                        ;
                                        ;
013230  042703  177770        GENER:  BIC     #177770,R3
013234  004737  013530                JSR     PC,SAVREG
013240  006303                        ASL     R3
013242  000173  013246                JMP     @GENSEL(R3)
013246  013266                GENSEL: GEN0                    ;ALL ZERO WORD
013250  013272                        GEN1                    ;ALL ONE WORD
013252  013300                        GEN52                   ;52 PATTERN
013254  013306                        GEN25                   ;25 PATTERN
013256  013314                        GENR1                   ;ROTATE '1' EACH CALL
013260  013324                        GENR0                   ;ROTATE '0' EACH CALL
013262  013362                        GENRAN                  ;RANDOM NUMBER
013264  013502                        GENINC                  ;INCREMENTING COUNT
013266  005000                GEN0:   CLR     R0              ;0>R0
013270  000507                        BR      GENEX
013272  005000                GEN1:   CLR     R0              ;NOT0>R0
013274  005100                        COM     R0
013276  000504                        BR      GENEX
013300  012700  052525        GEN52:  MOV     #52525,R0       ;5252>R0
013304  000501                        BR      GENEX
013306  012700  125252        GEN25:  MOV     #125252,R0      ;125252>R0
013312  000476                        BR      GENEX
013314  000241                GENR1:  CLC
013316  004737  013336                JSR     PC,GENROT       ;SHIFT 1 > R0
013322  000472                        BR      GENEX
013324  000241                GENR0:  CLC
013326  004737  013336                JSR     PC,GENROT       ;
013332  005100                        COM     R0              ;SHIFT 0 > R0
013334  000465                        BR      GENEX
013336  006037  013360        GENROT: ROR     GENISH          ;ROTATE  1 PATTERN
013342  001003                        BNE     GENER1          ;= 0?
013344  012737  100000  013360        MOV     #100000,GENISH  ;YES, SET MSB
013352  013700  013360        GENER1: MOV     GENISH,R0       ;PUT 1 IN R0
013356  000207                        RTS     PC              ;AND EXIT
013360  000001                GENISH: 1
013362  012737  000005  002402 GENRAN: MOV     #5,RANSEL       ;SET SELECT VALUE TO 5
013370  004737  013402                JSR     PC,RANGEN       ;GENERATE RANDOM NUMBER IN R0
013374  013700  002406                MOV     RANDN,R0        ;
013400  000443                        BR      GENEX           ;
013402  013702  002406        RANGEN: MOV     RANDN,R2        ;
013406  001002                        BNE     RAN1            ;IS RANDOM = 0
013410  013702  002400                MOV     RANST,R2        ;YES, PUT RANDOM START VALUE IN
013414  032737  000777  002402 RAN1:   BIT     #777,RANSEL     ;NO;IS RANSEL SELECT VALUE = 0
013422  001003                        BNE     RAN2            ;NO
013424  012737  000001  002402        MOV     #1,RANSEL       ;YES: SET RANSEL = 1
013432  013703  002402        RAN2:   MOV     RANSEL,R3       ;
013436  013702  002406                MOV     RANDN,R2
013442  033702  002404                BIT     RANMTA,R2       ;GET R2 <0 AND 1>
013446  001405                        BEQ     RANCLC          ;
013450  005102                        COM     R2
013452  033702  002404                BIT     RANMTA,R2
013456  001401                        BEQ     RANCLC
013460  000402                        BR      RANSEC
013462  000241                RANCLC: CLC
013464  000401                        BR      RAN4
013466  000261                RANSEC: SEC
```

```
1743 013470  006037  002406          RAN4:   ROR     RANDN           ;ROTATE C TO B15
1744 013474  005303                          DEC     R3              ;IS THIS NUMBER REQUIRED?
1745 013476  001357                          BNE     RAN2+4          ;NO, GET ANOTHER
1746 013500  000207                  RANEX:  RTS     PC              ;YES, EXIT
1747 013502  013700  002330          GENINC: MOV     GOOD,R0         ;INCREMENTS LOC. 'GOOD'
1748 013506  005200                          INC     R0
1749 013510  010037  002330          GENEX:  MOV     R0,GOOD
1750 013514  004737  013610                  JSR     PC,RSTREG
1751 013520  013737  002330  012436          MOV     GOOD,DATA
1752 013526  000207                          RTS     PC
1753
```

```
                                .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
                        ;
                        ;
013530                  SAVREG: GETPRI  SAVSTA
013536                          SETPRI  MAXPRI
013544  012637  002310          MOV     (SP)+,SAVPC     ;SAVE PC FOR RETURN FROM THIS ROUTINE
```

SAVE REGISTERS

```
1812 013550  012637  002410                MOV     (SP)+,SAVPC1
1813 013554  010546                        MOV     R5,-(SP)
1814 013556  010446                        MOV     R4,-(SP)
1815 013560  010346                        MOV     R3,-(SP)
1816 013562  010246                        MOV     R2,-(SP)
1817 013564  010146                        MOV     R1,-(SP)
1818 013566  010046                        MOV     R0,-(SP)
1819 013570  013746  002410                MOV     SAVPC1,-(SP)
1820 013574  013746  002310                MOV     SAVPC,-(SP)     ;PUT PC READY FOR
1821 013600                                SETPRI  SAVSTA
1822 013606  000207                        RTS     PC              ;RETURN
1823
1824
1825
```

```
                                .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
                        ;
                        ;
013610                  RSTREG: GETPRI  SAVSTA
013616                          SETPRI  MAXPRI
013624  012637  002310          MOV     (SP)+,SAVPC
013630  012637  002410          MOV     (SP)+,SAVPC1
013634  012600                  MOV     (SP)+,R0
013636  012601                  MOV     (SP)+,R1
```

```
1884 013640  012602                        MOV     (SP)+,R2
1885 013642  012603                        MOV     (SP)+,R3
1886 013644  012604                        MOV     (SP)+,R4
1887 013646  012605                        MOV     (SP)+,R5
1888 013650  013746  002410                MOV     SAVPC1,-(SP)
1889 013654  013746  002310                MOV     SAVPC,-(SP)     ;PUT PC READY FOR
1890 013660                                SETPRI  SAVSTA
1891 013666  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
                                        ;
                                        ;
                                        ;
                                        ;
013670  012537  002330          CKSELO: MOV     (R5)+,GOOD      ;WRITE GOOD
013674  017737  176610  002354          MOV     @KMVCSR,SELO    ;READ SEL 0
013702  023737  002354  002330          CMP     SELO,GOOD       ;CMP ?
013710  001001                          BNE     1$
013712  000402                          BR      2$
013714  062705  000002          1$:     ADD     #2,R5
013720  000205                  2$:     RTS     R5
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
                                        ;
013722  005037  002330          CBSELO: CLR     GOOD
013726  012537  002330                  MOV     (R5)+,GOOD
013732  117737  176552  012434          MOVB    @KMVCSR,BSELO
013740  123737  012434  002330          CMPB    BSELO,GOOD
013746  001001                          BNE     1$
013750  000402                          BR      2$
013752  062705  000002          1$:     ADD     #2,R5
013756  000205                  2$:     RTS     R5
```

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


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


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


013760  012537  002332          CKALL:  MOV     (R5)+,GOOD0
013764  012537  002336                  MOV     (R5)+,GOOD2
013770  012537  002340                  MOV     (R5)+,GOOD4
013774  012537  002342                  MOV     (R5)+,GOOD6
014000  012537  002344                  MOV     (R5)+,GOOD10
014004  012537  002346                  MOV     (R5)+,GOOD12
014010  012537  002350                  MOV     (R5)+,GOOD14
014014  012537  002352                  MOV     (R5)+,GOOD16

014020  017737  176464  002354          MOV     @KMVCSR,SEL0            ;READ SEL0
014026  000240                          NOP
014030  000240                          NOP
014032  000240                          NOP
014034  017737  176452  002360          MOV     @KMVP02,SEL2            ;READ SEL2
014042  000240                          NOP
014044  000240                          NOP
014046  000240                          NOP
014050  017737  176440  002362          MOV     @KMVP04,SEL4            ;READ SEL4
014056  000240                          NOP
014060  000240                          NOP
014062  000240                          NOP
014064  017737  176426  002364          MOV     @KMVP06,SEL6            ;READ SEL6
014072  000240                          NOP
014074  000240                          NOP
014076  000240                          NOP
014100  017737  176414  002366          MOV     @KMVP10,SEL10           ;READ SEL10
014106  000240                          NOP
014110  000240                          NOP
014112  000240                          NOP
014114  017737  176402  002370          MOV     @KMVP12,SEL12           ;READ SEL12
014122  000240                          NOP
014124  000240                          NOP
014126  000240                          NOP
014130  017737  176370  002372          MOV     @KMVP14,SEL14           ;READ SEL14
014136  000240                          NOP
014140  000240                          NOP
```

```
1987 014142  000240                          NOP
1988 014144  017737  176356  002374          MOV     @KMVP16,SEL16           ;READ SEL16
1989
1990
1991 014152  023737  002354  002332          CMP     SEL0,GOOD0
1992 014160  001035                          BNE     1$
1993 014162  023737  002360  002336          CMP     SEL2,GOOD2
1994 014170  001031                          BNE     1$
1995 014172  023737  002362  002340          CMP     SEL4,GOOD4
1996 014200  001025                          BNE     1$
1997 014202  023737  002364  002342          CMP     SEL6,GOOD6
1998 014210  001021                          BNE     1$
1999 014212  023737  002366  002344          CMP     SEL10,GOOD10
2000 014220  001015                          BNE     1$
2001 014222  023737  002370  002346          CMP     SEL12,GOOD12
2002 014230  001011                          BNE     1$
2003 014232  023737  002372  002350          CMP     SEL14,GOOD14
2004 014240  001005                          BNE     1$
2005 014242  023737  002374  002352          CMP     SEL16,GOOD16
2006 014250  001001                          BNE     1$
2007
2008 014252  000402                          BR      2$
2009 014254  062705  000002          1$:     ADD     #2,R5
2010 014260  000205                  2$:     RTS     R5
```

```
2012                                    ;ROUTINE TO CHECK SEL2 TO SEL16
2013
2014
2015
2016
2017
2018 014262  012537  002336          CKREG:  MOV     (R5)+,GOOD2
2019 014266  012537  002340                  MOV     (R5)+,GOOD4
2020 014272  012537  002342                  MOV     (R5)+,GOOD6
2021 014276  012537  002344                  MOV     (R5)+,GOOD10
2022 014302  012537  002346                  MOV     (R5)+,GOOD12
2023 014306  012537  002350                  MOV     (R5)+,GOOD14
2024 014312  012537  002352                  MOV     (R5)+,GOOD16
2025
2026
2027 014316  017737  176170  002360          MOV     @KMVP02,SEL2
2028 014324  000240                          NOP
2029 014326  000240                          NOP
2030 014330  000240                          NOP
2031 014332  000240                          NOP
2032 014334  017737  176154  002362          MOV     @KMVP04,SEL4
2033 014342  000240                          NOP
2034 014344  000240                          NOP
2035 014346  000240                          NOP
2036 014350  000240                          NOP
2037 014352  017737  176140  002364          MOV     @KMVP06,SEL6
2038 014360  000240                          NOP
2039 014362  000240                          NOP
2040 014364  000240                          NOP
2041 014366  000240                          NOP
2042 014370  017737  176124  002366          MOV     @KMVP10,SEL10
2043 014376  000240                          NOP
2044 014400  000240                          NOP
2045 014402  000240                          NOP
2046 014404  000240                          NOP
2047 014406  017737  176110  002370          MOV     @KMVP12,SEL12
2048 014414  000240                          NOP
2049 014416  000240                          NOP
2050 014420  000240                          NOP
2051 014422  000240                          NOP
2052 014424  017737  176074  002372          MOV     @KMVP14,SEL14
2053 014432  000240                          NOP
2054 014434  000240                          NOP
2055 014436  000240                          NOP
2056 014440  000240                          NOP
2057 014442  017737  176060  002374          MOV     @KMVP16,SEL16
2058
2059
2060
2061
2062 014450  023737  002360  002336          CMP     SEL2,GOOD2
2063 014456  001031                          BNE     1$
2064 014460  023737  002362  002340          CMP     SEL4,GOOD4
2065 014466  001025                          BNE     1$
2066 014470  023737  002364  002342          CMP     SEL6,GOOD6
2067 014476  001021                          BNE     1$
2068 014500  023737  002366  002344          CMP     SEL10,GOOD10
```

```
2069 014506  001015                          BNE     1$
2070 014510  023737  002370  002346          CMP     SEL12,GOOD12
2071 014516  001011                          BNE     1$
2072 014520  023737  002372  002350          CMP     SEL14,GOOD14
2073 014526  001005                          BNE     1$
2074 014530  023737  002374  002352          CMP     SEL16,GOOD16
2075 014536  001001                          BNE     1$
2076 014540  000402                          BR      2$
2077
2078 014542  062705  000002          1$:     ADD     #2,R5
2079 014546  000205                  2$:     RTS     R5
```

```
                                        ;ROUTINE TO CLEAR KMV11 MODULE
                                        ;
                                        ;CALLING SEQUENCE:
                                        ;       JSR PC,CLRKMV
                                        ;
                                        ;ROUTINE DESCRIPTION: CLEAR ALL CSR'S REGISTERS AND CHECK IF = 0
                                        ;
                                        ;
014550  005077  175734          CLRKMV: CLR     @KMVCSR
014554  012777  054000  175726          MOV     #MAINTO,@KMVCSR         ;SET MAINTENANCE MODE
014562                                  WAITA   0
                                        ;
014574  012702  000010                  MOV     #10,R2
014600  013701  012510                  MOV     KMVCSR,R1               ;LOAD ADDRESS
014604  005021                  1$:     CLR     (R1)+                   ;CLEAR
014606  000240                          NOP
014610  000240                          NOP
014612  000240                          NOP
014614  005302                          DEC     R2                      ;ALL DONE
014616  001372                          BNE     1$                      ;NO
014620  004537  013760                  JSR     R5,CKALL                ;CHECK ALL REG = 0
014624  000000                          .WORD   0
014626  000000                          .WORD   0
014630  000000                          .WORD   0
014632  000000                          .WORD   0
014634  000000                          .WORD   0
014636  000000                          .WORD   0
014640  000000                          .WORD   0
014642  000000                          .WORD   0
014644  000404                          BR      2$                      ;OK BRANCH AT END
014646                                  ERRHRD 2,EM0002,PRALL           ;CSR'S REGISTERS CAN'T BE CLEARED
014656  000207                  2$:     RTS     PC
                                        ;
```

```
                                ;ROUTINE TO SET MAINTENANCE MODE C ON KMV11



                                ;CALLING SEQUENCE:
                                ;       JSR PC,MAINMO



                                ;MAINTO = MASTER CLEAR=1 +MAINT1=1 +MODE = 1  ;DCT11 = HOLD


                                ;TEST DESCRIPTION:SET MAINTENANCE MODE 0 AND CHECK THAT MASTER CLEAR
                                ;                       IS RESET BY DCT11 MICRO PROCESSOR
                                ;
                                ;                       GIVE AN ERROR IF NOT RESET


014660  012777  054000  175622  MAINMO: MOV     #MAINTO,@KMVCSR         ;LOAD MAINTO
014666  012737  177000  002324          MOV     #177000,DELCT1
014674  012737  000001  002326          MOV     #1,DELCT2
014702  004737  012760                  JSR     PC,WAIT2                        ;WAIT
014706  004537  013670                  JSR     R5,CKSEL0               ;CHECK SEL0=0 BUT MODE BIT + MAINT1 BIT
014712  014000                          .WORD   14000
014714  000404                          BR      1$
014716                                  ERRHRD  3,EM0001,PRSEL0
014726  000207                  1$:     RTS     PC
```

```
                                        ;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=NOT IN HOLD

014730  005077  175554        MAINM1: CLR     @KMVCSR
014734  000240                        NOP
014736  000240                        NOP
014740  012777  044000  175542        MOV     #MAINT1,@KMVCSR         ;LOAD ADDRESS
014746  012737  177700  002324        MOV     #177700,DELCT1
014754  012737  000001  002326        MOV     #1,DELCT2
014762  004737  012760                JSR     PC,WAIT2
014766  004537  013670                JSR     R5,CKSEL0               ;CHECK SEL0=0 BUT MODE BIT =1
014772  004000                        .WORD   4000
014774  000404                        BR      1$                      ;OK BRANCH
014776                                ERRHRD  4,EM0001,PRSEL0
015006  000207                1$:     RTS     PC
```

```
 2186                                   ;ROUTINE TO SET TEST NUMBER ON BSEL0
 2187
 2188
 2189
 2190
 2191                                   ;CALLING SEQUENCE:
 2192                                   ;       JSR R5,TSTNUB
 2193                                   ;       .WORD   A                       A=TEST MICRO PROGRAM NUMBER
 2194
 2195
 2196
 2197
 2198
 2199
 2200 015010  012537  012456           TSTNUB: MOV     (R5)+,NUB
 2201 015014  053777  012456  175466           BIS     NUB,@KMVCSR             ;LOAD TEST NUMBER
 2202 015022  012737  170000  002324           MOV     #170000,DELCT1
 2203 015030  012737  000001  002326           MOV     #1,DELCT2
 2204 015036  004737  012760                   JSR     PC,WAIT2                ;WAIT
 2205 015042  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
                                ;       THE READ DATA IS LOAD IN LOCATION 'BAD' AND 'DATA'


015044  012577  175442          WRITE:  MOV     (R5)+,@KMVP02           ;WRITE ADDRESS
015050  012577  175440                  MOV     (R5)+,@KMVP04           ;  ''    DATA
015054  012777  000001  175434          MOV     #1,@KMVP06              ;BIT WRITE

015062  004537  015010                  JSR     R5,TSTNUB               ;SEND TEST NB 44
015066  000047                          .WORD   47

015070  000205                          RTS     R5                      ;RETURN



015072  012577  175414          READ:   MOV     (R5)+,@KMVP02           ;SET ADDRESS TO READ
015076  005077  175412                  CLR     @KMVP04
015102  005077  175410                  CLR     @KMVP06

015106  004537  015010                  JSR     R5,TSTNUB               ;SEND TEST NB 44
015112  000047                          .WORD   47

015114  000240                          NOP
```

```
2264 015116  000240                          NOP
2265
2266
2267 015120  004737  013152                  JSR     PC,TSTERR               ;CHECK BSEL 0
2268 015124  000410                          BR      1$                      ;OK
2269 015126  000402                          BR      2$
2270 015130  000401                          BR      2$
2271 015132  000400                          BR      2$
2272
2273 015134                          2$:     ERRHRD  5,EM0024                ;NO KMV ANSWER
2274 015144  000205                          RTS     R5
2275
2276 015146  017737  175342  012432  1$:     MOV     @KMVP04,BAD             ;READ DATA IN BAD
2277 015154  013737  012432  012436          MOV     BAD,DATA                ;READ DATA IN "DATA" LOCATION
2278
2279 015162  000205                          RTS     R5
2280
2281
2282
2283
```

```
                                .MACRO  ROMCLK
                                .LIST
                                        JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
                                .NLIST
                                .ENDM

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



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

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

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

                                .NLIST BEX


015164  045  116  045  TFM36:  .ASCIZ  /%N%AREGISTER ADDRESS ERROR,ADDR = %06%A,UNIT = %02/

015247  040  102  125  TIM:    .ASCIZ  / BUS TIMEOUT /

015265  115  101  123  EM0001: .ASCIZ  /MASTER CLEAR FAIL TO RESET: DCT11 CAN'T CLEAR MASTER CLEAR /

015361  040  113  115  EM0002: .ASCIZ  / KMV11 REGISTERS CAN'T BE CLEARED /

015424  040  104  101  EM0003: .ASCIZ  / DATA COMPARE ERROR ON KMV11 REGISTER (SEL2 TO SEL16)/

015512  040  104  101  EM0004: .ASCIZ  / DATA COMPARE ERROR ON BSEL0 WHEN ACCESSED BY QBUS/

015575  040  122  105  EM0005: .ASCIZ  / REGISTER SEL2 CAN'T BE ACCESSED CORRECTLY BY MICRO PROGRAM/

015671  105  122  122  EM0006: .ASCIZ  /ERROR WHEN TESTING SEL4,DCT11 CAN'T  ACCESS SEL4 CORRECTLY/

015764  105  122  122  EM0007: .ASCIZ  /ERROR WHEN TESTING SEL6,DCT11 CAN'T ACCESS SEL6 CORRECTLY/

016056  105  122  122  EM0010: .ASCIZ  /ERROR WHEN TESTING SEL10,DCT11 CAN'T ACCESS SEL10 CORRECTLY/

016152  105  122  122  EM0011: .ASCIZ  /ERROR WHEN TESTING SEL12,DCT11 CAN'T ACCESS SEL12 CORRECTLY/

016246  105  122  122  EM0012: .ASCIZ  /ERROR WHEN TESTING SEL14,DCT11 CAN'T ACCESS SEL14 CORRECTLY/

016342  105  122  122  EM0013: .ASCIZ  /ERROR WHEN TESTING SEL16,DCT11 CAN'T ACCESS SEL16 CORRECTLY/

016436  040  104  101  EM0015: .ASCIZ  / DATA COMPARE ERROR IN RAM MEMORY TEST /

016506  040  124  111  EM0016: .ASCIZ  / TIMEOUT DURING DMA TRANSFER /

016544  040  104  101  EM0020: .ASCIZ  / DATA COMPARE ERROR AFTER DMA  TRANSFER INTO KMV11 /

016630  040  104  101  EM0021: .ASCIZ  / DATA COMPARE ERROR AFTER DMA TRANSFER IN BOTH DIRECTION /

016722  111  116  124  EM0022: .ASCIZ  /INTERUPT  OCCUR AT WRONG LEVEL /

016762  116  117  040  EM0023: .ASCIZ  /NO INTERUPT OCCUR /

017005  116  117  040  EM0024: .ASCIZ  /NO ANSWER FROM KMV11 MODULE ,  MICRO TEST NOT EXECUTED /

017075  124  111  115  EM0025: .ASCIZ  /TIMEOUT DURING KMV11 MICRO TEST /

017136  111  116  124  EM0026: .ASCIZ  /INTERUPT ON DCT11 WITH ILLEGAL VECTOR WHEN ACCESSING BSEL2 /

017232  116  117  040  EM0027: .ASCIZ  /NO KMV11 ANSWER ,DCT11  RECEIVE NO INTERUPT /
```

```
2365 017307     111     114     114  EM0028: .ASCIZ  /ILLEGAL INTERUPT OCCURED /
2366
2367 017341     104     101     124  EM0030: .ASCIZ  /DATA COMPARE ERROR DURING DMA TRANSFER OUT KMV11 /
2368
2369 017423     105     122     122  EM0031: .ASCIZ  /ERROR DURING BYTE ACCES ON KMV11 REGISTERS /
2370
2371 017477     111     116     124  EM0032: .ASCIZ  /INTERUPT ON DCT11 WITH ILLEGAL VECTOR WHEN ACCESSING BSELO /
2372
2373 017573     122     101     115  EM0033: .ASCIZ  /RAM MEMORY ERROR WHEN TRANSFERING BUFFER IN DMA /
2374
2375 017654     120     122     117  EM0034: .ASCIZ  /PROM REVISION IS NOT COMPATIBLE WITH DIAGNOSTIC REVISION /
2376
2377 017746     040     103     110  EM0134: .ASCIZ  / CHECK PROM AND DIAGNOSTIC REVISION /
2378
2379 020013     040     040     120  EM0035: .ASCIZ  /  PROM CHECKSUM ERROR /
2380
2381
2382
```

```
2384 020042  045  116  045  MSEL0:  .ASCIZ  /%N%A SEL0 = %06%A  SHOUD BE  = %06%N/
2385
2386 020107  045  116  045  MREG0:  .ASCIZ  /%N%A SEL0 = %06%A  SHOUD BE  = %06/
2387 020152  045  116  045  MREG2:  .ASCIZ  /%N%A SEL2 = %06%A  SHOUD BE  = %06/
2388 020215  045  116  045  MREG4:  .ASCIZ  /%N%A SEL4 = %06%A  SHOUD BE  = %06/
2389 020260  045  116  045  MREG6:  .ASCIZ  /%N%A SEL6 = %06%A  SHOUD BE  = %06/
2390 020323  045  116  045  MREG10: .ASCIZ  /%N%A SEL10= %06%A  SHOUD BE  = %06/
2391 020366  045  116  045  MREG12: .ASCIZ  /%N%A SEL12= %06%A  SHOUD BE  = %06/
2392 020431  045  116  045  MREG14: .ASCIZ  /%N%A SEL14= %06%A  SHOUD BE  = %06/
2393 020474  045  116  045  MREG16: .ASCIZ  /%N%A SEL16= %06%A  SHOUD BE  = %06/
2394
2395
2396 020537  045  116  045  MSEL2:  .ASCIZ  /%N%A SEL2 = %06%A  SHOUD BE  = %06/
2397
2398 020602  045  116  045  MSEL4:  .ASCIZ  /%N%A SEL4 = %06%A  SHOUD BE  = %06/
2399
2400 020645  045  116  045  MSEL10: .ASCIZ  /%N%A SEL10 = %06%A SHOUD BE  =%06/
2401
2402 020707  045  116  045  MRAM1:  .ASCIZ  /%N%A  RAM ADDRESS  =%06%A,EXTENDED ADDRESS   =%06/
2403 020771  045  116  045  MRAM2:  .ASCIZ  /%N%A          BDDAT = %06%A SHOUD CONTENT  = %06/
2404
2405 021051  045  116  045  MINT:   .ASCIZ  /%N%A GOOD= %06%A  BAD=%06/
2406
2407 021103  045  116  045  MBSEL0: .ASCIZ  /%N%A BSEL0 =%06%A  SHOUD BE  =%06/
2408
2409 021145  045  116  045  MINTR:  .ASCIZ  /%N%A DCT11 ILLEGAL INTERUPT WHEN ADDRESS =%06%A IS WRITTEN/
2410
2411 021240  045  116  045  MDMA1:  .ASCIZ  /%N%A   DMA ERROR AT ADDRESS =%06%A EXTADDRESS =%06/
2412 021323  045  116  045  MDMA2:  .ASCIZ  /%N%A       BDDAT=%06%A SHOULD BE =%06/
2413
2414 021371  045  116  045  MBYTE:  .ASCIZ  /%N%A   AT ADDRESS ADDR =%06%A, GOOD =%06%A,BAD =%06/
2415
2416 021453  045  116  045  MDMAR1: .ASCIZ  /%N%A    RAM MEMORY LOCATION  =%06%A  IS MODIFIED DURING /
2417 021542  045  116  045  MDMAR2: .ASCIZ  /%N%A    DMA TRANSFER IN BOTH DIRECTION /
2418 021610  045  116  045  MDMAR3: .ASCIZ  /%N%A        READ DATA  =%06%A  SHOULD BE  =%06/
2419
2420 021666  045  116  045  MCHECK: .ASCIZ  /%N%A    CHECKSUM IS =%06%A SHOULD BE ZERO /
2421
2422 021737  045  116  045  MDMAF:  .ASCIZ  /%N%A   ADDR = %06%A  ,GDDAT = %06%A  ,BDDAT  = %06/
2423
2424
2425
```

```
2427
2428                                    .EVEN
2429
2430
2431
2432                                    ;-------------------------------------------------------------------------
2433                                    ; MACRO'S NEEDED TO REPORT ERRORS
2434                                    ;-------------------------------------------------------------------------
2435
2436 022022                                    BGNMSG  PRSEL0                                  ;REPORT CONTENT OF SEL0
2437 022022                                    PRINTB  #MSEL0,SEL0,GOOD
2438 022052  004737  013010                     JSR     PC,CHKMAX                               ;CHECK IF MAX ERROR?
2439 022056                                    ENDMSG
2440
2441
2442 022060                                BGNMSG  PRRAM                                       ;RAM ERROR REPORT
2443 022060                                PRINTB  #MRAM1,ADDR,EXADDR
2444 022110                                PRINTB  #MRAM2,BDDAT,GOOD
2445 022140  004737  013010                     JSR     PC,CHKMAX                               ;CHECK IF MAX ERROR?
2446 022144                                ENDMSG
2447
2448
2449
2450 022146                                BGNMSG  PRBYTE                                      ;BYTE ACCES REPORT
2451 022146                                PRINTB  #MBYTE,ADDR,GOOD,BAD
2452 022202  004737  013010                     JSR     PC,CHKMAX                               ;CHECK IF MAX ERROR?
2453 022206                                ENDMSG
2454
2455
2456 022210                                BGNMSG  PDMARA                                      ;DMA IN RAM ERROR REPORT
2457 022210                                PRINTB  #MDMAR1,ADDR
2458 022234                                PRINTB  #MDMAR2
2459 022254                                PRINTB  #MDMAR3,BDDAT,GOOD
2460 022304                                        BREAK
2461 022306  004737  013010                     JSR     PC,CHKMAX                               ;CHECK IF MAX ERROR?
2462 022312                                ENDMSG
2463
2464
2465 022314                                BGNMSG  PCHECK                                      ;CHECKSUM ERROR REPORT
2466 022314                                PRINTB  #MCHECK,BAD
2467 022340  004737  013010                     JSR     PC,CHKMAX                               ;CHECK IF MAX ERROR?
2468 022344                                ENDMSG
2469
2470
2471
2472
2473 022346                                BGNMSG  PADFLT                                      ;ADDRESS TEST
2474 022346                                PRINTB  #TFM36,ADDR,UNIT
2475 022376  004737  013010                     JSR     PC,CHKMAX
2476 022402                                ENDMSG
2477
2478
2479
2480
2481
```

```
2483 022404                          BGNMSG  PRALL                          ;CSR'S CONTENT REPORT
2484 022404                          PRINTB  #MREG0,SEL0,GOOD0
2485 022434                          PRINTB  #MREG2,SEL2,GOOD2
2486 022464                          PRINTB  #MREG4,SEL4,GOOD4
2487 022514                          PRINTB  #MREG6,SEL6,GOOD6
2488 022544                          PRINTB  #MREG10,SEL10,GOOD10
2489 022574                          PRINTB  #MREG12,SEL12,GOOD12
2490 022624                          PRINTB  #MREG14,SEL14,GOOD14
2491 022654                          PRINTB  #MREG16,SEL16,GOOD16
2492 022704                                  BREAK
2493 022706  004737  013010                  JSR     PC,CHKMAX              ;CHECK IF MAX ERROR?
2494 022712                          ENDMSG
2495
2496
2497
2498
2499
2500
2501
2502
2503 022714                          BGNMSG  PRREG                          ;CSR'S REPORT BUT SEL0
2504 022714                          PRINTB  #MREG2,SEL2,GOOD2
2505 022744                          PRINTB  #MREG4,SEL4,GOOD4
2506 022774                          PRINTB  #MREG6,SEL6,GOOD6
2507 023024                          PRINTB  #MREG10,SEL10,GOOD10
2508 023054                          PRINTB  #MREG12,SEL12,GOOD12
2509 023104                          PRINTB  #MREG14,SEL14,GOOD14
2510 023134                          PRINTB  #MREG16,SEL16,GOOD16
2511 023164                                  BREAK
2512 023166  004737  013010                  JSR     PC,CHKMAX              ;CHECK IF MAX ERROR?
2513 023172                          ENDMSG
2514
2515
2516
2517
2518
2519
2520
2521 023174                          BGNMSG  PBSEL0                         ;BSEL0 REPORT
2522 023174                          PRINTB  #MBSEL0,BSEL0,GOOD
2523 023224  004737  013010                  JSR     PC,CHKMAX              ;CHECK IF MAX ERROR?
2524 023230                          ENDMSG
2525
2526
2527
2528
2529
2530 023232                          BGNMSG  PINTR                          ;INTERUPT REPORT
2531 023232                          PRINTB  #MINTR,ADDR
2532 023256  004737  013010                  JSR     PC,CHKMAX              ;CHECK IF MAX ERROR?
2533 023262                          ENDMSG
2534
2535
2536
```

```
2538
2539 023264                          BGNMSG  PRDMA                          ;DMA ERROR REPORT
2540 023264                          PRINTB  #MDMA1,ADDR,EXADDR
2541 023314                          PRINTB  #MDMA2,BDDAT,GOOD
2542 023344  004737  013010                  JSR     PC,CHKMAX              ;CHECK IF MAX ERROR?
2543 023350                                  BREAK
2544 023352                          ENDMSG
2545
2546
2547 023354                          BGNMSG  PDMAF                          ;DMA SHORT REPORT
2548 023354                          PRINTB  #MDMAF,ADDR,GDDAT,BDDAT
2549 023410                          ENDMSG
2550
2551
2552
2553
```

```
2555                                    .SBTTL  REPORT CODING SECTION
2556
2557
2558                                    ;++
2559                                    ; THE REPORT CODING SECTION CONTAINS THE
2560                                    ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
2561                                    ;--
2562
2563 023412                                    BGNRPT
2564
2570
2571 023412                                    EXIT    RPT
2572
2579
2580 023416                                    ENDRPT
2581
2582
```

```
                                .SBTTL  INITIALIZE SECTION

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

023420                                  BGNINIT



                                        .EVEN



                                ;INITIALIZE SUBROUTINE STACK
023420  012705  012732                  MOV     #SSTACK,R5
                                ;STORE BASE LEVEL PROGRAM STACK POINTER
023424  010637  002304                  MOV     SP,PSTACK
023430  005737  002316                  TST     FTIME
023434  001011                          BNE     1$
023436  013737  000004  002312          MOV     @#4,SAVE4
023444  013737  000006  002314          MOV     @#6,SAVE6
023452  012737  000001  002316          MOV     #1,FTIME
023460  013737  002312  000004  1$:     MOV     SAVE4,@#4
023466  013737  002314  000006          MOV     SAVE6,@#6

023474                                  READEF  #EF.START                       ;START COMMAND?
023502                                  BCOMPLETE       SETUP                   ;IF YES BRANCH

023504                                  READEF  #EF.CONTINUE                    ;CONTINUE COMMAND?
023512                                  BCOMPLETE       END


023514                                  READEF  #EF.NEW                         ;NEW PASS?
023522                                  BNCOMPLETE      NEXT                    ;IF NOT EXIT SETUP

023524  012737  177777  012472  SETUP:  MOV     #-1,UUT                         ;INITIALISE UNIT NUMBER

023532  005237  012472          NEXT:   INC     UUT                             ;POINT NEXT UNIT
023536  023737  012472  002270          CMP     UUT,L$UIT                       ;ALL DONE?
023544  001521                          BEQ     ABORT                           ;IF YES END OF PASS

023546  013701  012472                  MOV     UUT,R1
023552                                  PRINTF  #RUNNING,R1
                                        .EVEN

023574                                  GPHARD  UUT,R1                          ;GET P TABLE
023604                                  BNCOMPLETE      NEXT                    ;IF NOT AVAILABLE GET NEXT


023606                          GETPRM:
                                                                        ;GET ADDRESS OF KMV11
023606  011137  012510                  MOV     (R1),KMVCSR
                                                                        ;GET POINTER TO KMV11 SEL 2 REG
023612  011137  012512                  MOV     (R1),KMVP02
```

```
2675 023616  062737  000002  012512          ADD     #2,KMVP02
2676                                                                ;GET POINTER TO KMV11 PORT REG - SEL 4
2677 023624  011137  012514                  MOV     (R1),KMVP04
2678 023630  062737  000004  012514          ADD     #4,KMVP04
2679                                                                ;GET POINTER TO KMV11 PORT REG - SEL 6
2680 023636  011137  012516                  MOV     (R1),KMVP06
2681 023642  062737  000006  012516          ADD     #6,KMVP06
2682                                                                ;GET POINTER TO KMV11 REG 10
2683 023650  011137  012520                  MOV     (R1),KMVP10
2684 023654  062737  000010  012520          ADD     #10,KMVP10
2685                                                                ;GET POINTER TO KMV11 REG 12
2686 023662  011137  012522                  MOV     (R1),KMVP12
2687 023666  062737  000012  012522          ADD     #12,KMVP12
2688                                                                ;GET POINTER TO KMV11 REG 14
2689 023674  011137  012524                  MOV     (R1),KMVP14
2690 023700  062737  000014  012524          ADD     #14,KMVP14
2691                                                                ;GET POINTER TO KMV11 REG 16
2692 023706  012137  012526                  MOV     (R1)+,KMVP16
2693 023712  062737  000016  012526          ADD     #16,KMVP16
2694                                                                ;GET POINTER TO  VECTOR 0
2695 023720  011137  012474                  MOV     (R1),KMVV00
2696                                                                ;GET POINTER TO  VECTOR 2
2697 023724  011137  012502                  MOV     (R1),KMVV02
2698 023730  062737  000002  012502          ADD     #2,KMVV02
2699                                                                ;GET POINTER TO VECTOR 4
2700 023736  011137  012500                  MOV     (R1),KMVV04
2701 023742  062737  000004  012500          ADD     #4,KMVV04
2702                                                                ;GET POINTER TO VECTOR 6
2703 023750  012137  012504                  MOV     (R1)+,KMVV06
2704 023754  062737  000006  012504          ADD     #6,KMVV06
2705                                                                ;GET POINTER TO TX PRIORITY LEVEL
2706 023762  012137  012476                  MOV     (R1)+,KMVLVL
2707 023766  062737  000006  012506          ADD     #6,KMTLVL
2708                                                                ;GET LOOPBACK PARAMETERS:
2709 023774  011137  012530                  MOV     (R1),LOOP
2710
2711 024000  005037  002300                  CLR     ERRCNT          ;CLEAR ERROR COUNT
2712 024004                                  EXIT    INIT
2713
2714
2715
2716 024010                          ABORT:  DOCLN                           ;CLEAN UP AND ABORT PASS
2717 024012                                  EXIT INIT                       ;EXIT
2718
2719                                         .NLIST  BEX
2720 024016     045     116     045  RUNNING:        .ASCIZ  /%N%A RUNNING ON UNIT %D2%A  PASS TIME = 2 MINUTES/
2721                                         .LIST   BEX
2722                                         .EVEN
2723
2724
2725
2726 024100                          END:    ENDINIT
2727
2728
2729
2730
```

```
                                .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
024102                                  BGNAUTO


                                        ;CHECK IF EXISTING DEVICE

024102  013701  012510                  MOV     KMVCSR,R1       ;R1 CONTAINS BASE KMV11 ADDRESS
024106  012705  000007                  MOV     #7,R5           ;7 REGISTERS TO BE TESTED
024112  012737  024144  000004          MOV     #2$,4           ;SET OUT TIMEOUT TRAP
024120  012737  000340  000006          MOV     #340,6          ;LEVEL 7
024126  005711                  1$:     TST     (R1)            ;REFERENCE DEVICE REGISTERS
024130  000240                          NOP
024132  062701  000002                  ADD     #2,R1           ;NEXT REGISTER
024136  005305                          DEC     R5              ;DEC REGISTER COUNT
024140  001372                          BNE     1$              ;BR IF NOT LAST REGISTER
024142  000405                          BR      3$

024144  062706  000004          2$:     ADD     #4,SP
024150                                  DODU    LOGDEV

024156  013737  002312  000004  3$:     MOV     SAVE4,4
024164  013737  002314  000006          MOV     SAVE6,6


024172                                  ENDAUTO
```

```
                                .SBTTL  CLEANUP CODING SECTION

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

024174                                  BGNCLN



024174                                  BRESET

024176                                  ENDCLN
```

```
                                .SBTTL  DROP UNIT SECTION

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

024200                                  BGNDU


                                        .EVEN

024200                                  PRINTF  #DROPD,R0       ;UNIT DROPPED

024222                                  EXIT    DU


                                        .NLIST  BEX
024226  045  116  045  DROPD:  .ASCIZ  /%N%A UNIT %D2%A DROPPED/
                                        .LIST   BEX
                                        .EVEN

024256                                  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.
                                ;////////////////////////////////////////////////////////////////

024260                                  BGNAU
024260                                  ENDAU
```

```
2904
2905                                    .SBTTL  HARDWARE TESTS
2906
2907
2908
2909                                    ;START OF CODE BLOCK WHICH IS USED AS DATA
2910 024262                             ROMMAP::++
2911                                    ; TEST TO ...
2912                                    ;--
2913
2920
2926
2927                                    ;       BGNTST
2928
2934
2935                                    ;       EXIT    TST
2936
2948
2949                                            .EVEN
2950                                    ;       ENDTST
2951
2957
```

```
                                        BADHEAD
                                ;********************* TEST1 ******************
                                        ;*VERIFY THAT REFERENCING QBUS DEVICE REGISTERS
                                        ;*DOES NOT CAUSE A TIME OUT TRAP
                                        BADHEAD
                                ;********************* TEST1 ******************

                                BGNTST
024262  013701  012510          1$:     MOV     KMVCSR,R1       ;R1 CONTAINS KMV11 ADDRESSES
024266  012705  000007                  MOV     #7,R5           ;7 REGISTERS TO BE TESTED
024272  012737  024330  000004          MOV     #2$,4           ;SET OUT TIMEOUT TRAP
024300  012737  000340  000006          MOV     #340,6          ;LEVEL 7
024306  005711                          TST     (R1)            ;REFERENCE DEVICE REGISTERS
024310  000240                          NOP
024312                                  ESCAPE  TST
024316  062701  000002                  ADD     #2,R1           ;NEXT REGISTER
024322  005305                          DEC     R5              ;DEC REGISTER COUNT
024324  001370                          BNE     1$              ;BP IF NOT LAST REGISTER
024326  000413                          BR      3$

024330  062706  000004          2$:     ADD     #4,SP
024334  010137  002420                  MOV     R1,ADDR         ;REPORT ADDRESS LOCATION
024340  013737  012472  002272          MOV     UUT,UNIT        ;REPORT UNIT NUMBER
024346                                  ERRHRD  0,TIM,PADFLT    ;BUS TIMEOUT,ADDRESS PROBLEM ON THIS UNIT

024356  013737  002312  000004  3$:     MOV     SAVE4,4
024364  013737  002314  000006          MOV     SAVE6,6
024372                                  ESCAPE  TST

024376                          ENDTST
                                .EVEN
```

```
                                BADHEAD
                                ;********************* TEST2 ******************
                                ;CLEAR ALL KMV11 REGISTERS AND CHECK
                                BADHEAD
                                ;********************* TEST2 ******************


                                BGNTST
                                BGNSUB

                                RESTST:
                                        CLR     @KMVCSR
                                        MOV     #MAINTO,@KMVCSR                 ;SET MASTER CLEAR TO EXIT
                                                                                ;SELF TEST IF RUNNING
                                        WAITA   0


                                        MOV     #10,R2                          ;LOAD NUMBER OF REGISTER
                                        MOV     KMVCSR,R1
                                3$:     CLR     (R1)+                           ;CLR KMV11 REGISTERS
                                        NOP
                                        NOP
                                        DEC     R2
                                        BNE     3$                              ;ALL DONE?

                                        JSR     R5,CKALL                        ;CHECK ALL REGISTERS = 0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        .WORD   0
                                        BR      2$                                      ;OK BRANCH
                                        ERRHRD  1,EM0002,PRALL          ;REGISTERS FAIL TO RESET
                                        ESCAPE  SUB

                                2$:     NOP
                                ENDSUB


                                BGNSUB
                                        JSR     PC,CLRKMV                       ;CLEAR REGISTERS

                                        MOV     #MAINTO,@KMVCSR                 ;SET MASTER CLEAR,MODE BIT AND MAINT

                                        WAITA   0

                                        JSR     R5,CKSEL0                       ;CHECK MASTER CLR IS RESET BY DCT11
                                        .WORD   14000

                                        BR      1$                              ;YES
```

```
3045 024554                             ERRHRD  6,EM0001,PRSEL0         ;MASTER CLR FAIL TO RESET
3046 024564                             ESCAPE  SUB
3047
3048 024570                         1$:
3049 024570                         ENDSUB
3050 024572                         ENDTST
3051
3052
3053
3054
```

```
                                BADHEAD
                                ;********************* TEST3 ******************
                                ;CHECK QBUS ACCESS ON KMV11 REGISTERS (FROM SEL2 TO SEL16)
                                BADHEAD
                                ;********************* TEST3 ******************


                                STARS 1
                                ;SET MAINT MODE =0  ;DCT11 DECODE AND GOES IN HOLD
                                ;PROCESSOR SEND ROTATING PATTERN TO EACH REGISTERS AND CHECK
                                STARS 1


                                BGNTST
                        TSTREG: JSR     PC,CLRKMV               ;CLEAR REGISTERS
                                JSR     PC,MAINMO               ;SET MAINT MODE 0
                                MOV     #7,COUNT                ;NUMBER OF REG
                                MOV     #CHECK,R4
                                ADD     #4,R4                   ;POINT GOOD VALUE OF SEL2
                                MOV     KMVP02,R1               ;LOAD SEL2 ADDRESS

                        TSELA:  CLR     R3                      ;SELECT FIRST PATTERN

                                BREAK

                        TSELB:  JSR     PC,GENER                ;GENER PATTERN


                        1$:     MOV     DATA,(R1)                       ;LOAD PATTERN IN REG
                                MOV     DATA,(R4)                       ;LOAD GOOD VALUE
                                NOP

                        CHECK:  JSR     R5,CKREG                ;CHECK ALL REGISTER BUT SELO
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                .WORD   0
                                BR      1$                      ;IF GOOD BR
                                ERRHRD  7,EM0003,PRREG
                                ESCAPE  TST

                        1$:     INC     R3                      ;NEW PATTERN
                                CMP     #7,R3                   ;ALL DONE
                                BNE     TSELB                   ;NO BR

                                CLR     (R1)+                   ;SELECT NEW REG
                                CLR     (R4)+                   ;POINT NEW GOOD VALUE
                                DEC     COUNT                   ;ALL REG TESTED
                                BNE     TSELA                   ;NO BR
```

```
   3111 024732                          ENDTST
```

```
3113 024734                          BADHEAD
                                     ;********************* TEST4 ******************
3114                                 ;CHECK QBUS ACCES ON SEL0 REGISTER
3115 024734                          BADHEAD
                                     ;********************* TEST4 ******************
3116
3117
3118
3119 024734                          STARS 1
3120                                 ;SET MAINT MODE 0     ;DCT11=HOLD
3121                                 ;SEND ROTATING PATTERN IN SEL0 (EXCEPT BIT 11,12,14) AND CHECK
3122 024734                          STARS 1
3123
3124
3125
3126
3127 024734                          BGNTST
3128 024734  004737  014550                 JSR     PC,CLRKMV               ;CLEAR REG
3129 024740  004737  014660                 JSR     PC,MAINM0               ;LOAD MAINT MODE 0
3130 024744  005003                         CLR     R3                      ;FIRST PATTERN
3131 024746  012704  025014                 MOV     #CHECK1,R4              ;POINT SEL 0
3132 024752  062704  000004                 ADD     #4,R4
3133 024756  012737  000015  002414  TCSRNB: MOV     #15,COUNT               ;SELECT NB 0 PATTERN
3134
3135 024764                                 BREAK
3136
3137 024766  004737  013230          TCSR:  JSR     PC,GENER                ;GERERATE A  PATTERN
3138 024772  042737  054000  012436         BIC     #54000,DATA             ;MASK   MCLR,MODE,MAINT1
3139 025000  013714  012436                 MOV     DATA,(R4)
3140 025004  013777  012436  165476         MOV     DATA,@KMVCSR            ;WRITE PATTERN
3141 025012  000240                         NOP
3142 025014  004537  013760          CHECK1: JSR     R5,CKALL                ;CHECK
3143 025020  000000                         .WORD   0
3144 025022  000000                         .WORD   0
3145 025024  000000                         .WORD   0
3146 025026  000000                         .WORD   0
3147 025030  000000                         .WORD   0
3148 025032  000000                         .WORD   0
3149 025034  000000                         .WORD   0
3150 025036  000000                         .WORD   0
3151 025040  000411                         BR      1$
3152 025042                                 ERRHRD  8,EM0004,PRALL
3153 025052                                 ESCAPE  TST
3154
3155 025056  005337  002414                 DEC     COUNT                   ;DONE ENOUGH
3156 025062  001341                         BNE     TCSR
3157
3158
3159 025064  005203                  1$:    INC     R3                      ;NEW PATTERN
3160 025066  022703  000007                 CMP     #7,R3                   ;ALL DONE
3161 025072  001331                         BNE     TCSRNB                  ;NO BR
3162 025074                          ENDTST
```

```
                                BADHEAD
                                ;********************** TEST5 ***********************
                                ;CHECK QBUS BYTE ACCES ON ALL KMV11 REGISTERS
                                BADHEAD
                                ;********************** TEST5 ***********************


                                STARS 1
                                ;SET MAINT MODE 0  ;DCT11=HOLD
                                ;WRITE PATTERN IN EACH BYTE ON KMV11 REGISTERS AND CHECK
                                ;QBUS SEND VARIOUS PATTERN IN ALL BYTE ADDRESS
                                STARS 1


                                BGNTST
                                        JSR     PC,CLRKMV
                                TBYTE:  JSR     PC,MAINM0               ;SET MAINT MODE0

                                        MOV     KMVCSR,R1               ;LOAD KMV CSR ADDRESS
                                        MOV     #15,R4                  ;LOAD NUMBER OF REGISTERS

                                1$:     MOV     #377,GOOD               ;SELECT A PATTERN
                                        BICB    #130,GOOD
                                        BISB    GOOD,(R1)               ;WRITE 1ST BYTE
                                        CLR     BAD

                                        WAITA   177700

                                        BREAK

                                        MOVB    (R1),BAD                ;READ REG
                                        BICB    #130,BAD                ;MASK UNUSED BITS

                                        CMPB    GOOD,BAD                ;COMPARE
                                        BEQ     3$                      ;IF = BRANCH


                                        MOV     R1,ADDR                 ;PREPARE ERROR REPORT
                                        ERRHRD  9,EM0031,PRBYTE         ;DATA CMP ERROR WHEN ACCESSING A BYTE
                                        ESCAPE  TST


                                3$:     INC     R1                      ;SELECT NEW REGISTER
                                        DEC     R4                      ;DONE ALL?
                                        BNE     1$
```

```
3219 025226                         ENDTST
3220
```

```
 3222 025230                        BADHEAD
                                    ;********************* TEST6 ******************
 3223                               ;DATA TRANSFER ON REGISTER SEL 2
 3224 025230                        BADHEAD
                                    ;********************* TEST6 ******************
 3225
 3226
 3227 025230                        STARS 1
 3228                               ;SET MAINT1  ;DCT11 DECODE ,CLEAR SEL0 AND WAIT FOR TEST NUMBER
 3229                               ;THE HOST WRITE A PATTERN IN SEL2
 3230                               ;THE HOST WRITE A TEST NUMBER IN BSEL0
 3231                               ;
 3232                               ;IF DCT11 READ CORRECT VALUE ,IT CLEAR BSEL0
 3233                               ;IF ERROR       SET 100 IN BSEL0 IF DATA CMP ERROR
 3234                               ;               BSEL0 =TST NUMBER IF NO KMV11 ANSWER
 3235                               ;
 3236                               ;
 3237                               ;BSEL0=1 = MICRO DIAG TEST 1  ;DCT11 MUST READ 052525 IN SEL2
 3238                               ;BSEL0=2 = MICRO DIAG TEST 2  ;DCT11 MUST READ 125252 IN SEL2
 3239 025230                        STARS 1
 3240
 3241
 3242
 3243
 3244
 3245 025230                        BGNTST
 3246 025230  004737  014550                JSR     PC,CLRKMV               ;CLEAR REG
 3247 025234  004737  014730                JSR     PC,MAINM1               ;SET MAINT MODE 1
 3248
 3249 025240                        BGNSUB
 3250 025242  012777  052525  165242        MOV     #DATA1,@KMVP02          ;SEND 052525
 3251 025250  004537  015010                JSR     R5,TSTNUB               ;SEND   TEST NUMB 1
 3252 025254  000001                        .WORD   1
 3253 025256  004537  013722                JSR     R5,CBSEL0               ;CHECK BSEL0 = 0
 3254 025262  000000                        .WORD   0
 3255 025264  000425                        BR      1$                      ;TEST OK BR AT END
 3256 025266  004537  013722                JSR     R5,CBSEL0               ;CHECK BSEL0=100
 3257 025272  000100                        .WORD   100
 3258 025274  000401                        BR      2$
 3259 025276  000410                        BR      3$
 3260
 3261
 3262
 3263 025300                        2$:     ERRHRD  10,EM0005               ;DATA CMP ERROR
 3264 025310  004737  013010                JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
 3265 025314                                ESCAPE  SUB
 3266
 3267 025320                        3$:     ERRHRD  11,EM0024               ;NO KMV11 ANSWER
 3268 025330  004737  013010                JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
 3269 025334                                ESCAPE SUB
 3270 025340  000240                1$:     NOP
 3271 025342                        ENDSUB
 3272
 3273
 3274 025344                        BGNSUB
 3275 025346  004737  014730                JSR     PC,MAINM1               ;SET MAINT MODE 1
 3276 025352  012777  125252  165132        MOV     #DATA2,@KMVP02          ;SEND   125252
```

```
3277 025360  004537  015010              JSR     R5,TSTNUB          ;SEND TEST NUB 2
3278 025364  000002                      .WORD   2
3279 025366  004537  013722              JSR     R5,CBSEL0          ;CHECK DCT11 HAS ANSWERED
3280 025372  000000                      .WORD   0                  ;BY CLEARING SEL0
3281 025374  000425                      BR      1$                 ;OK BR
3282 025376  004537  013722              JSR     R5,CBSEL0          ;CHECK IF =100
3283 025402  000100                      .WORD   100
3284 025404  000401                      BR      2$
3285 025406  000410                      BR      3$
3286
3287
3288
3289 025410                       2$:    ERRHRD  12,EM0005          ;DATA CMP ERROR ON SEL2
3290 025420  004737  013010              JSR     PC,CHKMAX          ;CHECK IF TOO MANY ERROR
3291 025424                              ESCAPE  SUB
3292
3293 025430                       3$:    ERRHRD  13,EM0024          ;NO KMV11 ANSWER
3294 025440  004737  013010              JSR     PC,CHKMAX          ;CHECK IF TOO MANY ERROR
3295 025444                              ESCAPE  SUB
3296
3297 025450  000240                1$:   NOP
3298 025452                        ENDSUB
3299 025454                        ENDTST
```

```
3301 025456                          BADHEAD
                                     ;********************* TEST7 ********************
3302                                 ; CHECK DATA TRANSFER ON REGISTER SEL4
3303 025456                          BADHEAD
                                     ;********************* TEST7 ********************
3304
3305
3306
3307 025456                          STARS 1
3308                                 ;SET MAINT1  ;DCT11 DECODE AND CLEAR SEL0
3309                                 ;
3310                                 ;THE HOST SEND ROTATING PATTERN IN SEL4,AND SET TEST NUMBER 3 IN BSEL0
3311                                 ;
3312                                 ;DCT11 READ SEL4 , WRITE CONTENT OF SEL4 INTO SEL2 , CLEAR SEL0 WHEN DONE, AND
3313                                 ;WAIT FOR NEW PATTERN
3314                                 ;
3315                                 ;
3316                                 ;
3317                                 ;      AFTER TEST       BSEL0=100 IF ERROR DURING TEST
3318                                 ;                       BSEL0=0    IF TEST HAS BEEN EXECUTED (IN THAT CASE
3319                                 ;                                        CHECK IF CONTENT OF SEL2 IS CORRECT)
3320                                 ;
3321                                 ;BSEL 0 = 3 ;MICRO DIAG NB 3 ;DCT11 TAKE CONTENT OF SEL4 AND PUT IT IN SEL 2.
3322 025456                          STARS 1
3323
3324
3325 025456                          BGNTST
3326 025456                          BGNSUB
3327 025460  004737  014550                  JSR     PC,CLRKMV               ;CLEAR REG
3328 025464  004737  014730                  JSR     PC,MAINM1               ;SET MAINT MODE 1
3329 025470  005003                          CLR     R3                      ;SELECT FIRST PATTERN
3330
3331 025472  012737  000005  002414  TGENE1: MOV     #5,COUNT                ;NB OF PATTERN
3332 025500                          BREAK
3333
3334 025502  004737  013230          GENE1:  JSR     PC,GENER                ;GENER 1ST PATTERN
3335 025506  013777  012436  165000          MOV     DATA,@KMVP04            ;LOAD SEL4
3336 025514  004537  015010                  JSR     R5,TSTNUB               ;SET TEST NUMB  3
3337 025520  000003                          .WORD   3
3338 025522  004537  013722                  JSR     R5,CBSEL0               ;LOOK IF ANSWER
3339 025526  000000                          .WORD   0
3340 025530  000404                          BR      1$                      ;OK BR
3341 025532                                  ERRHRD  14,EM0024,PBSEL0        ;BSEL0 NOT CLEARED ,NO  ANSWER
3342 025542  000240          1$:             NOP
3343 025544                          ENDSUB
3344
3345
3346 025546                          BGNSUB
3347 025550  012704  025570                  MOV     #TSEL4,R4               ;POINT GOOD VALUE
3348 025554  013764  012436  000006          MOV     DATA,6(R4)
3349 025562  013764  012436  000004          MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL4
3350
3351 025570  004537  014262          TSEL4:  JSR     R5,CKREG                ;CHECK SEL2 = SEL4
3352 025574  000000                          .WORD   0
3353 025576  000000                          .WORD   0
3354 025600  000000                          .WORD   0
3355 025602  000000                          .WORD   0
```

```
3356 025604  000000                        .WORD   0
3357 025606  000000                        .WORD   0
3358 025610  000000                        .WORD   0
3359 025612  000406                        BR      2$
3360 025614                                ERRHRD  15,EM0006,PRREG
3361 025624                                ESCAPE  SUB
3362
3363
3364 025630  005337  002414        2$:     DEC     COUNT                   ;DONE ENOUGH?
3365 025634  001322                        BNE     GENE1
3366
3367 025636  005203                        INC     R3                      ;NEW PATTERN
3368 025640  022703  000007                CMP     #7,R3                   ;ALL DONE
3369 025644  001312                        BNE     TGENE1                  ;NO BR
3370 025646                        ENDSUB
3371 025650                        ENDTST
```

```
3373 025652                          BADHEAD
                                     ;********************* TEST8 ******************
3374                                 ; CHECK DATA TRANSFER ON REGISTER SEL6
3375 025652                          BADHEAD
                                     ;********************* TEST8 ******************
3376
3377
3378 025652                          STARS 1
3379                                 ;SET MAINT1  ;DCT11 DECODE AND CLEAR SEL0
3380                                 ;
3381                                 ;THE HOST SENDS A ROTATING PATTERN IN SEL6,AND SET TEST NUMBER 4 IN BSEL0
3382                                 ;
3383                                 ;DCT11 READ SEL6 , WRITE CONTENT OF SEL6 IN SEL2 , CLEAR SEL0 WHEN DONE, AND
3384                                 ;WAIT FOR NEW PATTERN
3385                                 ;
3386                                 ;
3387                                 ;
3388                                 ;      AFTER TEST       BSEL0=100 IF ERROR DURING TEST
3389                                 ;                       BSEL0=0    IF TEST HAS BEEN EXECUTED (IN THAT CASE
3390                                 ;                                        CHECK IF CONTENT OF SEL4 IS CORRECT)
3391                                 ;
3392                                 ;BSEL 0 = 4 ;MICRO DIAG NB 4 ;DCT11 TAKE CONTENT OF SEL6 AND PUT IT IN SEL 2.
3393 025652                          STARS 1
3394
3395
3396
3397 025652                          BGNTST
3398 025652                          BGNSUB
3399 025654  004737  014550                  JSR     PC,CLRKMV               ;CLEAR REG
3400 025660  004737  014730                  JSR     PC,MAINM1               ;SET MAINT MODE 1
3401 025664  012703  000001                  MOV     #1,R3                   ;SELECT 1ST PATTERN
3402 025670  012737  000005  002414  TGENE2: MOV     #5,COUNT                ;     ''   NUMBER OF PATTERN
3403 025676                                  BREAK
3404
3405
3406 025700  004737  013230          GENE2:  JSR     PC,GENER                ;GENERATE 1ST PATTERN
3407 025704  013777  012436  164604          MOV     DATA,@KMVP06            ;LOAD SEL6
3408 025712  004537  015010                  JSR     R5,TSTNUB               ;SET TEST NUMB  4
3409 025716  000004                          .WORD   4
3410 025720  004537  013722                  JSR     R5,CBSEL0               ;LOOK IF ANSWER
3411 025724  000000                          .WORD   0
3412 025726  000404                          BR      1$                      ;OK BR
3413 025730                                  ERRHRD  16,EM0024,PBSEL0        ;NO KMV11 ANSWER,BSEL0 NOT = 0
3414 025740  000240                  1$:     NOP
3415 025742                          ENDSUB
3416
3417
3418 025744                          BGNSUB
3419 025746  012704  025766                  MOV     #TSEL6,R4               ;POINT GOOD VALUE
3420 025752  013764  012436  000010          MOV     DATA,10(R4)
3421 025760  013764  012436  000004          MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL6
3422
3423 025766  004537  014262          TSEL6:  JSR     R5,CKREG                ;CHECK SEL2 = SEL6
3424 025772  000000                          .WORD   0
3425 025774  000000                          .WORD   0
3426 025776  000000                          .WORD   0
3427 026000  000000                          .WORD   0
```

```
3428 026002  000000                        .WORD   0
3429 026004  000000                        .WORD   0
3430 026006  000000                        .WORD   0
3431 026010  000406                        BR      2$
3432 026012                                ERRHRD  17,EM0007,PRREG
3433 026022                                ESCAPE  SUB
3434 026026  005337  002414          2$:   DEC     COUNT
3435 026032  001322                        BNE     GENE2
3436
3437 026034  005203                        INC     R3                      ;NEW PATTERN
3438 026036  022703  000006                CMP     #6,R3                   ;ALL DONE
3439 026042  001312                        BNE     TGENE2                  ;NO BR
3440 026044                        ENDSUB
3441 026046                        ENDTST
```

```
3443 026050                          BADHEAD
                                     ;********************* TEST9 ********************
3444                                 ;TEST TO CHECK DATA TRANSFER ON REGISTER SEL10
3445 026050                          BADHEAD
                                     ;********************* TEST9 ********************
3446
3447
3448 026050                          STARS 1
3449                                 ;SET MAINT1  ;DCT11 DECODE AND CLEAR SEL0
3450                                 ;
3451                                 ;THE HOST SENDS A ROTATING PATTERN IN SEL10,AND SET TEST NUMBER 5 IN BSEL0
3452                                 ;
3453                                 ;DCT11 READ SEL10 , WRITE CONTENT OF SEL10 IN SEL2 , CLEAR SEL0 WHEN DONE, AND
3454                                 ;WAIT FOR NEW PATTERN
3455                                 ;
3456                                 ;
3457                                 ;
3458                                 ;      AFTER TEST       BSEL0=100 IF ERROR DURING TEST
3459                                 ;                       BSEL0=0    IF TEST HAS BEEN EXECUTED (IN THAT CASE
3460                                 ;                                        CHECK IF CONTENT OF SEL6 IS CORRECT)
3461                                 ;
3462                                 ;BSEL 0 = 5 ;MICRO DIAG NB 5 ;DCT11 TAKE CONTENT OF SEL10 AND PUT IT IN SEL 2.
3463 026050                          STARS 1
3464
3465
3466
3467 026050                          BGNTST
3468 026050                          BGNSUB
3469 026052  004737  014550                JSR     PC,CLRKMV               ;CLEAR REG
3470 026056  004737  014730                JSR     PC,MAINM1               ;SET MAINT MODE 1
3471 026062  012703  000001                MOV     #1,R3                   ;SELECT FIRST PATTERN
3472
3473 026066  012737  000005  002414  TGENE3: MOV     #5,COUNT
3474 026074                                BREAK
3475
3476
3477 026076  004737  013230          GENE3:  JSR     PC,GENER                ;GENERATE A 1ST PATTERN
3478 026102  042737  040000  012436          BIC     #40000,DATA             ;MASK BIT 14
3479 026110  013777  012436  164402          MOV     DATA,@KMVP10            ;LOAD SEL10
3480 026116  004537  015010                JSR     R5,TSTNUB               ;SET TEST NUMB  5
3481 026122  000005                        .WORD   5
3482 026124  004537  013722                JSR     R5,CBSEL0               ;LOOK IF ANSWER
3483 026130  000000                        .WORD   0
3484 026132  000406                        BR      1$                      ;OK BR
3485 026134                                ERRHRD  20,EM0024,PBSEL0        ;NO KMV11 ANSWER
3486 026144                                ESCAPE  SUB
3487 026150  000240                  1$:   NOP
3488 026152                          ENDSUB
3489
3490
3491 026154                          BGNSUB
3492 026156  012704  026176                MOV     #TSEL10,R4              ;POINT GOOD VALUE
3493 026162  013764  012436  000012        MOV     DATA,12(R4)
3494 026170  013764  012436  000004        MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL10
3495
3496 026176  004537  014262          TSEL10: JSR     R5,CKREG                ;CHECK SEL2 = SEL10
3497 026202  000000                        .WORD   0
```

```
3498 026204  000000                        .WORD   0
3499 026206  000000                        .WORD   0
3500 026210  000000                        .WORD   0
3501 026212  000000                        .WORD   0
3502 026214  000000                        .WORD   0
3503 026216  000000                        .WORD   0
3504 026220  000406                        BR      2$
3505 026222                                ERRHRD  21,EM0010,PRREG         ;DATA CMP ERROR IN SEL10
3506 026232                                ESCAPE  SUB
3507
3508 026236  005337  002414        2$:     DEC     COUNT
3509 026242  001315                        BNE     GENE3
3510
3511 026244  005203                        INC     R3                      ;NEW PATTERN
3512 026246  022703  000006                CMP     #6,R3                   ;ALL DONE
3513 026252  001305                        BNE     TGENE3                  ;NO BR
3514 026254                        ENDSUE
3515 026256                        ENDTST
```

```
                                BADHEAD
                                ;********************* TEST10 ******************
                                ;TEST TO CHECK DATA TRANSFER ON REGISTER SEL12
                                BADHEAD
                                ;********************* TEST10 ******************


                                STARS 1
                                ;SET MAINT1  ;DCT11 DECODE AND CLEAR SEL0
                                ;
                                ;THE HOST SENDS A ROTATING PATTERN IN SEL12,AND SET TEST NUMBER 6 IN BSEL0
                                ;
                                ;DCT11 READ SEL12 , WRITE CONTENT OF SEL12 IN SEL 2 , CLEAR SEL0 WHEN DONE, AND
                                ;WAIT FOR NEW PATTERN
                                ;
                                ;
                                ;
                                ;       AFTER TEST      BSEL0=100 IF ERROR DURING TEST
                                ;                       BSEL0=0    IF TEST HAS BEEN EXECUTED (IN THAT CASE
                                ;                                        CHECK IF CONTENT OF SEL10 IS CORRECT)
                                ;
                                ;BSEL 0 = 6 ;MICRO DIAG NB 6 ;DCT11 TAKE CONTENT OF SEL12 AND PUT IT IN SEL 2.
                                STARS 1


                                BGNTST
                                BGNSUB
                                        JSR     PC,CLRKMV               ;CLEAR REG
                                        JSR     PC,MAINM1               ;SET MAINT MODE 1
                                        MOV     #1,R3

                                TGENE4: MOV     #5,COUNT
                                        BREAK


                                GENE4:  JSR     PC,GENER                ;GENERATE A 1ST PATTERN
                                        MOV     DATA,@KMVP12            ;LOAD SEL12
                                        JSR     R5,TSTNUB               ;SET TEST NUMB  6
                                        .WORD   6
                                        JSR     R5,CBSEL0               ;LOOK IF ANSWER
                                        .WORD   0
                                        BR      1$                      ;OK BR
                                        ERRHRD  22,EM0024,PBSEL0                ;NO KMV11 ANSWER
                                        ESCAPE  SUB
                                1$:     NOP
                                ENDSUB


                                BGNSUB
                                        MOV     #TSEL12,R4              ;POINT GOOD VALUE
                                        MOV     DATA,14(R4)
                                        MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL12

                                TSEL12: JSR     R5,CKREG                ;CHECK SEL2 = SEL12
                                        .WORD   0
                                        .WORD   0
```

```
3572 026410  000000                          .WORD   0
3573 026412  000000                          .WORD   0
3574 026414  000000                          .WORD   0
3575 026416  000000                          .WORD   0
3576 026420  000000                          .WORD   0
3577 026422  000406                          BR      2$
3578 026424                                  ERRHRD  23,EM0011,PRREG         ;DATA CMP ERROR IN SEL12
3579 026434                                  ESCAPE  SUB
3580
3581
3582 026440  005337  002414          2$:     DEC     COUNT
3583 026444  001320                          BNE     GENE4
3584
3585 026446  005203                          INC     R3                      ;NEW PATTERN
3586 026450  022703  000006                  CMP     #6,R3                   ;ALL DONE
3587 026454  001310                          BNE     TGENE4                  ;NO BR
3588 026456                          ENDSUB
3589 026460                          ENDTST
```

```
3591 026462                          BADHEAD
                                     ;********************* TEST11 ******************
3592                                 ; CHECK DATA TRANSFER ON REGISTER SEL14
3593 026462                          BADHEAD
                                     ;********************* TEST11 ******************
3594
3595
3596 026462                          STARS 1
3597                                 ;SET MAINT1  ;DCT11 DECODE AND CLEAR SEL0
3598                                 ;
3599                                 ;THE HOST SEND ROTATING PATTERN IN SEL14,AND SET TEST NUMBER 6 IN BSEL0
3600                                 ;
3601                                 ;DCT11 READ SEL14 , WRITE CONTENT OF SEL14 IN SEL 2 , CLEAR SEL0 WHEN DONE, AND
3602                                 ;WAIT FOR NEW PATTERN
3603                                 ;
3604                                 ;
3605                                 ;
3606                                 ;      AFTER TEST       BSEL0=100 IF ERROR DURING TEST
3607                                 ;                       BSEL0=0    IF TEST HAS BEEN EXECUTED (IN THAT CASE
3608                                 ;                                        CHECK IF CONTENT OF SEL12 IS CORRECT)
3609                                 ;
3610                                 ;BSEL 0 = 6 ;MICRO DIAG NB 6 ;DCT11 TAKE CONTENT OF SEL14 AND PUT IT IN SEL 12.
3611 026462                          STARS 1
3612
3613
3614 026462                          BGNTST
3615 026462                          BGNSUB
3616 026464  004737  014550                  JSR     PC,CLRKMV               ;CLEAR REG
3617 026470  004737  014730                  JSR     PC,MAINM1               ;SET MAINT MODE 1
3618 026474  012703  000001                  MOV     #1,R3                   ;SELECT FIRST PATTERN
3619
3620 026500  012737  000005  002414  TGENE5: MOV     #5,COUNT                ;SELECT NB OF PATTERN
3621 026506                                  BREAK
3622
3623
3624 026510  004737  013230          GENE5:  JSR     PC,GENER                ;GENERATE A  1ST PATTERN
3625 026514  013777  012436  164002          MOV     DATA,@KMVP14            ;LOAD SEL14
3626 026522  004537  015010                  JSR     R5,TSTNUB               ;SET TEST NUMB  7
3627 026526  000007                          .WORD   7
3628 026530  004537  013722                  JSR     R5,CBSEL0               ;LOOK IF ANSWER
3629 026534  000000                          .WORD   0
3630 026536  000406                          BR      1$                      ;OK BR
3631 026540                                  ERRHRD  24,EM0024,PBSEL0                ;NO KMV11 ANSWER
3632 026550                                  ESCAPE  SUB
3633
3634 026554  000240                  1$:     NOP
3635 026556                          ENDSUB
3636
3637
3638 026560                          BGNSUB
3639 026562  012704  026602                  MOV     #TSEL14,R4              ;POINT GOOD VALUE
3640 026566  013764  012436  000016          MOV     DATA,16(R4)
3641 026574  013764  012436  000004          MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL4
3642
3643 026602  004537  014262          TSEL14: JSR     R5,CKREG                ;CHECK SEL2 = SEL14
3644 026606  000000                          .WORD   0
3645 026610  000000                          .WORD   0
```

```
3646 026612  000000                        .WORD   0
3647 026614  000000                        .WORD   0
3648 026616  000000                        .WORD   0
3649 026620  000000                        .WORD   0
3650 026622  000000                        .WORD   0
3651 026624  000406                        BR      2$
3652 026626                                ERRHRD  25,EM0012,PRREG          ;DATA CMP ERROR IN SEL14
3653 026636                                ESCAPE  SUB
3654
3655
3656 026642  005337  002414        2$:     DEC     COUNT                     ;DONE ENOUGH?
3657 026646  001320                        BNE     GENE5
3658
3659 026650  005203                        INC     R3                        ;NEW PATTERN
3660 026652  022703  000006                CMP     #6,R3                     ;ALL DONE
3661 026656  001310                        BNE     TGENE5                    ;NO BR
3662 026660                          ENDSUB
3663 026662                          ENDTST
```

```
                                BADHEAD
                                ;********************* TEST12 ********************
                                ; CHECK DATA TRANSFER ON REGISTER SEL16
                                BADHEAD
                                ;********************* TEST12 ********************


                                STARS 1
                                ;SET MAINT1  ;DCT11 DECODE AND CLEAR SEL0
                                ;
                                ;THE HOST SENDS A ROTATING PATTERN IN SEL16,AND SET TEST NUMBER 7 IN BSEL0
                                ;
                                ;DCT11 READ SEL16 , WRITE CONTENT OF SEL16 IN SEL 2 , CLEAR SEL0 WHEN DONE, AND
                                ;WAIT FOR NEW PATTERN
                                ;
                                ;
                                ;
                                ;       AFTER TEST      BSEL0=100 IF ERROR DURING TEST
                                ;                       BSEL0=0    IF TEST HAS BEEN EXECUTED (IN THAT CASE
                                ;                                     CHECK IF CONTENT OF SEL14 IS CORRECT)
                                ;
                                ;BSEL 0 = 10 ;MICRO DIAG NB 10 ;DCT11 TAKE CONTENT OF SEL16 AND PUT IT IN SEL 2


                                BGNTST
                                BGNSUB
026666  004737  014550                JSR     PC,CLRKMV               ;CLEAR REG
026672  004737  014730                JSR     PC,MAINM1               ;SET MAINT MODE 1
026676  012703  000001                MOV     #1,R3                   ;SELECT FIRST PATTERN

026702  012737  000005  002414  TGENE6: MOV     #5,CCUNT
026710                                BREAK


026712  004737  013230          GENE6:  JSR     PC,GENER                ;GENERATE 1ST PATTERN
026716  013777  012436  163602        MOV     DATA,@KMVP16            ;LOAD SEL16
026724  004537  015010                JSR     R5,TSTNUB               ;SET TEST NUMB  10
026730  000010                        .WORD   10
026732  004537  013722                JSR     R5,CBSEL0               ;LOOK IF ANSWFR
026736  000000                        .WORD   0
026740  000406                        BR      1$                      ;OK BR
026742                                ERRHRD  26,EM0024,PBSEL0                ;NO KMV11 ANSWER
026752                                ESCAPE  SUB

026756  000240                  1$:   NOP
026760                          ENDSUB


026762                          BGNSUB
026764  012704  027004                MOV     #TSEL16,R4              ;POINT GOOD VALUE
026770  013764  012436  000020        MOV     DATA,20(R4)
026776  013764  012436  000004        MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL16

027004  004537  014262          TSEL16: JSR     R5,CKREG                ;CHECK SEL2 = SEL16
027010  000000                        .WORD   0
027012  000000                        .WORD   0
027014  000000                        .WORD   0
```

```
3720 027016  000000                        .WORD   0
3721 027020  000000                        .WORD   0
3722 027022  000000                        .WORD   0
3723 027024  000000                        .WORD   0
3724 027026  000406                        BR      2$
3725 027030                                ERRHRD  27,EM0013,PRREG
3726 027040                                ESCAPE  SUB
3727
3728
3729 027044  005337  002414          2$:   DEC     COUNT                   ;DONE ENOUGH?
3730 027050  001320                        BNE     GENE6
3731
3732 027052  005203                        INC     R3                      ;NEW PATTERN
3733 027054  022703  000006                CMP     #6,R3                   ;ALL DONE
3734 027060  001310                        BNE     TGENE6                  ;NO BR
3735 027062                          ENDSUB
3736 027064                          ENDTST
3737
3738
3739
```

```
3741 027066                          BADHEAD
                                     ;********************** TEST13 ********************
3742                                 ; CHECK DATA TRANSFER ON ALL REGISTERS
3743 027066                          BADHEAD
                                     ;********************** TEST13 ********************
3744
3745
3746
3747
3748
3749 027066                          STARS 1
3750                                 ;SET MAINT1  DCT11 CLEAR BSEL0
3751                                 ;SEND DIFFERENT PATTERN IN SEL2,  SEND TEST 11
3752                                 ;DCT11 READ SEL2 AND WRITTE A CALCULATED VALUE IN SEL4 TO SEL16
3753                                 ;                (SEL4)=SEL2+SEL2
3754                                 ;                (SEL6)=SEL4+SEL2
3755                                 ;                (SEL10)=SEL6+SEL2
3756                                 ;                (SEL12)=SEL10+SEL2
3757                                 ;                (SEL14)=SEL12+SEL2
3758                                 ;                (SEL16)=SEL14+SEL2
3759                                 ;DCT11 CLEAR BSEL0 WHEN DONE
3760                                 ;
3761                                 ;=MICRO DIAG NUMBER 11
3762                                 ;
3763                                 ;
3764
3765
3766 027066                          STARS 1
3767
3768
3769
3770
3771 027066                          BGNTST
3772 027066  004737  014550                  JSR     PC,CLRKMV               ;CLEAR REG
3773 027072  004737  014730                  JSR     PC,MAINM1               ;SET MAINT1
3774
3775
3776 027076  012703  000004                  MOV     #4,R3                   ;PREPARE INCREMENTING PATTERN
3777 027102  012737  000007  002416          MOV     #7,NUMBER               ;SELECT NUMBER FOR DIFFERENT PATTERN
3778
3779
3780
3781 027110  004737  013230          RGALL:  JSR     PC,GENER                ;PREPARE ONE RANDOM PETTERN
3782
3783 027114  013777  012436  163370          MOV     DATA,@KMVP02            ;WRITE PATTERN IN SEL2
3784 027122  013737  012436  002336          MOV     DATA,GOOD2
3785 027130  004537  015010                  JSR     R5,TSTNUB               ;SEND TEST NB11
3786 027134  000011                          .WORD   11
3787
3788
3789 027136  012737  177700  002324          MOV     #177700,DELCT1
3790 027144  004737  013000                  JSR     PC,WAIT1
3791
3792
3793 027150                                   BREAK
3794
3795 027152  004537  013722                  JSR     R5,CBSEL0               ;LOOK IF TEST DONE
```

```
3796 027156  000000                        .WORD   0
3797 027160  000406                        BR      1$                      ;YES BRANCH
3798 027162                                ERRHRD  28,EM0024,PBSEL0        ;NO KMV11 ANSWER
3799 027172                                ESCAPE  TST
3800
3801
3802 027176  017737  163310  002360  1$:   MOV     @KMVP02,SEL2            ;READ SEL2 TO SEL16
3803 027204  017737  163304  002362        MOV     @KMVP04,SEL4
3804 027212  017737  163300  002364        MOV     @KMVP06,SEL6
3805 027220  017737  163274  002366        MOV     @KMVP10,SEL10
3806 027226  017737  163270  002370        MOV     @KMVP12,SEL12
3807 027234  017737  163264  002372        MOV     @KMVP14,SEL14
3808 027242  017737  163260  002374        MOV     @KMVP16,SEL16
3809
3810
3811 027250  013737  002336  002340        MOV     GOOD2,GOOD4
3812 027256  063737  002336  002340        ADD     GOOD2,GOOD4
3813
3814 027264  013737  002340  002342        MOV     GOOD4,GOOD6             ;WHAT IS GOOD6
3815 027272  063737  002336  002342        ADD     GOOD2,GOOD6
3816
3817 027300  013737  002342  002344        MOV     GOOD6,GOOD10            ;WHAT IS GOOD10
3818 027306  063737  002336  002344        ADD     GOOD2,GOOD10
3819
3820 027314  013737  002344  002346        MOV     GOOD10,GOOD12           ;  "   " GOOD12
3821 027322  063737  002336  002346        ADD     GOOD2,GOOD12
3822
3823 027330  013737  002346  002350        MOV     GOOD12,GOOD14           ;  "   "  GOOD14
3824 027336  063737  002336  002350        ADD     GOOD2,GOOD14
3825
3826 027344  013737  002350  002352        MOV     GOOD14,GOOD16           ;   "  "  GOOD16
3827 027352  063737  002336  002352        ADD     GOOD2,GOOD16
3828
3829
3830 027360  023737  002336  002360        CMP     GOOD2,SEL2
3831 027366  001031                        BNE     2$
3832 027370  023737  002340  002362        CMP     GOOD4,SEL4
3833 027376  001025                        BNE     2$
3834 027400  023737  002342  002364        CMP     GOOD6,SEL6
3835 027406  001021                        BNE     2$
3836 027410  023737  002344  002366        CMP     GOOD10,SEL10
3837 027416  001015                        BNE     2$
3838 027420  023737  002346  002370        CMP     GOOD12,SEL12
3839 027426  001011                        BNE     2$
3840 027430  023737  002350  002372        CMP     GOOD14,SEL14
3841 027436  001005                        BNE     2$
3842 027440  023737  002352  002374        CMP     GOOD16,SEL16
3843 027446  001001                        BNE     2$
3844 027450  000410                        BR      3$
3845
3846
3847 027452                          2$:   BREAK
3848 027454                                ERRHRD  29,EM0003,PRREG
3849 027464                                BREAK
3850 027466                                ESCAPE  TST
3851
3852
```

```
3853 027472  005337  002416          3$:     DEC     NUMBER                  ;ALL PATTERN DONE?
3854 027476                                  BREAK
3855 027500  001203                          BNE     RGALL
3856 027502                          ENDTST
```

```
                                BADHEAD
                                ;********************* TEST14 ******************
                                ;KMV11 RAM MEMORY TEST:  MEMORY PATTERN TEST
027504                          BADHEAD
                                ;********************* TEST14 ******************
```

```
027504                          STARS 1
                                ;SET MAINT MODE 1   ;DCT11 DECODE AND CLEAR SEL 0
                                ;THE HOST WRITES IN SEL2 THE PATTERN TO BE WRTTEN IN ALL MEMORY
                                ;AND SETS TEST NUMBER TO 13
                                ;
                                ;DCT11 WRITE ALL THE MEMORY WITH THIS VALUE,CHECK IF OK AND
                                ;WHEN DONE CLEAR BSEL0 IF TEST OK
                                ;
                                ;IF ERROR SEL2 = EXPECTED VALUE OF LOCATION IN ERROR
                                ;         SEL4 = READ VALUE OF LOCATION IN ERROR
                                ;         SEL6 = ADDRESS IN ERROR
                                ;
                                ;
                                ;
                                ;
                                ;BSEL0 = 13 ,  MICRO DIAGNOSTIC TEST NUMBER 13
                                ;               DCT11 SEND PATTERN IN RAM MEMORY AND CHECK
                                ;
                                ;               PATTERN DESCRIPTION:
                                ;               .       ALL     ZERO
                                ;               .       ALL     ONE
                                ;               .       10101010 PATERN
                                ;               .       01010101 PATERN
                                ;               .       ROTATING 1
                                ;               .       ROTATING 0
027504                          STARS 1
```

```
027504                          BGNTST
027504  004737  014550        RAMPAT: JSR     PC,CLRKMV               ;CLEAR REG
027510  004737  014730                JSR     PC,MAINM1               ;SET MAINT1

027514  005003                        CLR     R3                      ;SELECT 1ST PATTERN
027516  005037  002414                CLR     COUNT
027522  004737  013230        4$:     JSR     PC,GENER                ;MAKE PATTERN
027526  013777  012436  162756        MOV     DATA,@KMVP02            ;WRITE PATTERN IN SEL2

027534  004537  015010                JSR     R5,TSTNUB               ;SET TEST NB 13
027540  000013                        .WORD   13

027542                                WAITB   0,1

027562  004737  013152                JSR     PC,TSTERR                       ;CHECK BSEL0=WHICH ERROR?
027566  000441                        BR      1$                      ;TEST OK
```

```
3913 027570  000420                        BR      2$
3914                                                                ;TIME OUT ERROR
3915 027572  000427                        BR      3$               ;NO KMV11 ANSWER
3916
3917
3918
3919
3920
3921 027574  017737  162716  002420        MOV     @KMVP06,ADDR     ;READ   ADDRESS OF RAM
3922 027602  017737  162704  002422        MOV     @KMVP02,GDDAT    ;READ EXPECTED DATA (GDDAT)
3923 027610  017737  162700  002424        MOV     @KMVP04,BDDAT    ;READ BAD VALUE OF DATA (BDDAT)
3924
3925 027616                                ERRHRD  30,EM0015,PRRAM  ;DATA CMP ERROR ON ONE RAM LOCATION
3926 027626                                ESCAPE  TST
3927
3928
3929
3930
3931
3932 027632  005037  002330          2$:   CLR GOOD
3933 027636                                ERRHRD  31,EM0025,PBSEL0 ;TIMEOUT ERROR
3934 027646                                ESCAPE TST
3935
3936
3937
3938
3939
3940 027652  005037  002330          3$:   CLR     GOOD
3941 027656                                ERRHRD  32,EM0024,PBSEL0 ;NO KMV11 ANSWER
3942 027666                                ESCAPE  TST
3943
3944
3945
3946
3947
3948 027672  005237  002414          1$:   INC     COUNT
3949 027676  022737  000015  002414        CMP     #15,COUNT        ;SEND 1 WORDS IN THE SAME PATTERN
3950 027704  001306                        BNE     4$               ;15 WORDS DONE BR
3951
3952 027706  005037  002414                CLR     COUNT
3953 027712  005203                        INC     R3               ;TRY WITH A NEW PATTERN
3954 027714  022703  000006                CMP     #6,R3            ;ALL DONE ?
3955 027720  001300                        BNE     4$               ;NO BR
3956 027722                          ENDTST
```

```
                                BADHEAD
                                ;********************** TEST15 ********************
                                ;KMV11 RAM MEMORY TEST:  MEMORY ADDRESS TEST
                                BADHEAD
                                ;********************** TEST15 ********************
                                STARS 1
                                ;SET MAINT MODE 1   ;DCT11 DECODE AND CLEAR SEL 0
                                ;
                                ;THE HOST SETS TEST NB 14 IN BSEL10
                                ;DCT11 WRITE ADDRESS VALUE IN EACH ADDRESS LOCATIONS FOR ALL
                                ;THE KMV11 RAM.(EXEMPLE: 1000=1000,1002=1002.....).
                                ;DCT11 CLEAR BSEL0 IF TEST IS OK
                                ;             BSEL0= 100 IF DATA COMPARE ERROR DURING CHECK
                                ;
                                ;IF ERROR SEL2 = EXPECTED VALUEOF LOCATION IN ERROR
                                ;         SEL4 = READ VALUE OF LOCATION IN ERROR
                                ;         SEL6 = ADDRESS IN ERROR
                                ;
                                ;
                                ;
                                ;
                                ;BSEL0 = 14 ,  MICRO DIAGNOSTIC TEST NUMBER 14
                                ;              DCT11 WRITE ADDRESS VALUE IN ADDRESS LOCATION AND CHECK
                                STARS 1


                                BGNTST
027724  004737  014550          RAMADD: JSR     PC,CLRKMV               ;CLEAR REG
027730  004737  014730                  JSR     PC,MAINM1               ;SET MAINT1

027734  004537  015010                  JSR     R5,TSTNUB               ;SET TEST NB 14
027740  000014                          .WORD   14

027742                                  WAITB   0,1


027762  004737  013152                  JSR     PC,TSTERR               ;CHECK BSEL0
027766  000441                          BR      1$                      ;TEST OK
027770  000420                          BR      2$                      ;TIMEOUT ERROR
027772  000427                          BR      3$                      ;NO KMV11 ANSWER


027774  017737  162516  002420          MOV     @KMVP06,ADDR            ;READ   ADDRESS OF RAM
030002  017737  162504  002330          MOV     @KMVP02,GOOD            ;READ EXPECTED DATA (GDDAT)
030010  017737  162500  002424          MOV     @KMVP04,BDDAT           ;READ BAD VALUE OF DATA (BDDAT)

030016                                  ERRHRD  33,EM0015,PRRAM         ;DATA CMP ERROR ON ONE RAM LOCATION
030026                                  ESCAPE  TST
```

```
4013
4014
4015
4016
4017 030032  005037  002330          2$:     CLR     GOOD
4018 030036                                  ERRHRD  34,EM0025,PBSEL0        ;TIMEOUT ERROR
4019 030046                                  ESCAPE TST
4020
4021
4022
4023
4024 030052  005037  002330          3$:     CLR     GOOD
4025 030056                                  ERRHRD  35,EM0024,PBSEL0        ;NO KMV11 ANSWER
4026 030066                                  ESCAPE TST
4027 030072  000240                  1$:     NOP
4028 030074                          ENDTST
```

```
                                BADHEAD
                                ;********************* TEST16 *******************
                                ;KMV11 RAM MEMORY TEST:  MEMORY ADDRESS COMPLEMENT TEST
                                BADHEAD
                                ;********************* TEST16 *******************
                                STARS 1
                                ;SET MAINT MODE 1   ;DCT11 DECODE AND CLEAR SEL 0
                                ;SET TEST NUMBER 15   ;DCT11 EXECUTE TST
                                ;DCT11 WRITE COMPLEMENT ADDRESS VALUE IN EACH ADDRESS LOCATION AND CHECK.
                                ;
                                ;DCT11 CLEAR BSEL0 IF TEST OK AND PUT 100 IN BSEL0 IF DATA COMPARE ERROR
                                ;
                                ;IF ERROR SEL2 = EXPECTED VALUE OF LOCATION IN ERROR
                                ;         SEL4 = READ VALUE OF LOCATION IN ERROR
                                ;         SEL6 = ADDRESS IN ERROR
                                ;
                                ;
                                ;
                                ;BSEL0 = 15 ,  MICRO DIAGNOSTIC TEST NUMBER 15
                                ;              DCT11 WRITE COMPL. ADDRESS IN ADDRESS IN RAM MEMORY AND CHECK
                                STARS 1


                                BGNTST
030076  004737  014550          RAMCAD: JSR     PC,CLRKMV               ;CLEAR REG
030102  004737  014730                  JSR     PC,MAINM1               ;SET MAINT1

030106  004537  015010                  JSR     R5,TSTNUB               ;SET TEST NB 15
030112  000015                          .WORD   15

030114                                  WAITB   0,1


030134  004737  013152                  JSR     PC,TSTERR               ;CHECK  BSEL0,WHICH ERROR
030140  000441                          BR      1$                      ;TEST OK
030142  000420                          BR      2$                      ;TIMEOUT ERROR
030144  000427                          BR      3$                      ;NO KMV11 ANSWER

030146  017737  162344  002420          MOV     @KMVP06,ADDR            ;READ   ADDRESS OF RAM
030154  017737  162332  002330          MOV     @KMVP02,GOOD            ;READ EXPECTED DATA (GDDAT)
030162  017737  162326  002424          MOV     @KMVP04,BDDAT           ;READ BAD VALUE OF DATA (BDDAT)

030170                                  ERRHRD  36,EM0015,PRRAM         ;DATA CMP ERROR ON ONE RAM LOCATION
030200                                  ESCAPE  TST
```

```
4085 030204  005037  002330          2$:     CLR     GOOD
4086 030210                                  ERRHRD  37,EM0025,PBSEL0        ;TIMEOUT ERROR
4087 030220                                  ESCAPE  TST
4088
4089
4090
4091
4092
4093 030224  005037  002330          3$:     CLP     GOOD
4094 030230                                  ERRHRD  38,EM0024,PBSEL0        ;NO KMV11 ANSWER
4095 030240                                  ESCAPE  TST
4096
4097
4098
4099
4100 030244  000240                  1$:     NOP
4101 030246                          ENDTST
4102
4103
```

```
4105 030250                          BADHEAD
                                     ;********************** TEST17 ******  ***********
4106                                 ;CHECK PROM REVISION TO SEE IF COMPAT.BLE WHITH DIAGNOSTIC
4107 030250                          BADHEAD
                                     ;********************** TEST17 **********.*********
4108
4109
4110
4111
4112 030250                          STARS 1
4113                                 ;READ LOCATION 2 OF THE PROM (ADDRESS 160002) WHICH CONTENTS PROM VERSION
4114                                 ;       NUMBER
4115                                 ;CHECK IF DIAGNOSTIC AND PROM ARE COMPATIBLE AND GIVE AN ERROR IF NOT
4116 030250                          STARS 1
4117
4118
4119
4120
4121
4122 030250                          BGNTST
4123 030250  004737  014550                 JSR     PC,CLRKMV               ;CLEAR ALL REGISTERS
4124 030254  004737  014730                 JSR     PC,MAINM1               ;SET MAINT MODE
4125
4126
4127 030260  004537  015072          REVPRO: JSR     R5,READ                 ;READ LOCATION 160002
4128 030264  160002                         .WORD   160002
4129
4130
4131 030266  023737  012464  012432         CMP     GDREV,BAD               ;LOOK IF COMPATIBLE
4132 030274  001406                         BEQ     1$                      ;YES
4133
4134 030276                                 ERRHRD  39,EM0034               ;REPORT THE ERROR
4135 030306                                 ESCAPE TST
4136 030312                          1$:
4137 030312                          ENDTST
```

```
                                BADHEAD
                                ;********************** TEST18 ********************
                                ;PROM CHECKSUM TEST
                                BADHEAD
                                ;********************** TEST18 ********************
                                STARS 1
                                ;DIAGNOSIC READS ALL PROM'S LOCATIONS AND ADDS THEN TOGETHER
                                ;RESULT MUST BE ZERO
                                ;
                                ;
                                ;TEST 33 DESCRIPTION:
                                ;DCT11 ADD ALL PROMS LOCATIONS ,IF RESULT IS ZERO=CLEAR BSELO
                                ;                               IF CHECKSUM ERROR =SET 100 IN BSELO
                                STARS 1
                                BGNTST
030314  004737  014550          JSR     PC,CLRKMV               ;CLEAR REGISTERS
030320  004737  014730          JSR     PC,MAINM1               ;SET MAINTENANCE MODE

030324  004537  015010  PROMCK: JSR     R5,TSTNUB
030330  000033                  .WORD   33                      ;SET TEST 33

030332                          WAITB   0,1

030352  004737  013152          JSR     PC,TSTERR               ;TEST IF ERROR
030356  000427                  BR      1$                      ;TEST OK
030360  000412                  BR      2$                      ;TIMEOUT ERROR
030362  000417                  BR      3$                      ;NO ANSWER FROM KMV11

030364  017737  162122  012432  MOV     @KMVP02,BAD             ;CHECKSUM ERROR
030372                          ERRHRD  40,EM0035,PCHECK
030402                          ESCAPE  TST

030406                  2$:     ERRHRD  41,EM0025               ;TIMEOUT DURING TEST
030416                          ESCAPE  TST

030422                  3$:     ERRHRD  42,EM0024               ;NO KMV ANSWER
030432                          ESCAPE  TST

030436  000240          1$:     NOP
```

```
4194
4195 030440                          ENDTST
4196
4197
4198
4199
4200
4201
4202
4203
```

```
                                BADHEAD
                                ;********************** TEST19 ********************
                                ;TEST DMA TRANSFER IN KMV11
                                BADHEAD
                                ;********************** TEST19 ********************




                                STARS 1
                                ;SET MAINT1   ;DCT11 DECODE AND CLEAR BSEL0
                                ;LOAD FIRST ADDRESS OF TX TABLE IN SEL12 , TABLE LENGTH IN SEL14,
                                ;TX TABLE EXTENDED ADDRESS IN BSEL 10.
                                ;
                                ;SET TEST NUMBER (16 OR 17)IN BSEL0
                                ;DCT11 EXECUTE THE DMA TRANSFER OF THE TABLE IN KMV11 RAM AND CHECK.
                                ;WHEN DONE CLEAR BSEL0 IF TEST OK
                                ;SET 200 IN BSEL0 IF TIMEOUT DURING TEST
                                ;SET 100 IN BSEL0 IF ERROR DURING TRANSFER
                                ;IN THAT CASE            SEL2=EXPECTED VALUE
                                ;                        SEL4=READ VALUE
                                ;                        SEL6=ADDRESS LOCATION OF ERROR
                                ;                        BSEL10=EXTENDED ADDRESS
                                ;TEST DESCRIPTION: PDP GENERATE AN INCREMENTING PATTERN TABLE OF 1K WORDS
                                ;SEND STARTING ADDRESS AND TABLE LENGTH TO KMV11
                                ;KMV11 START DMA TRANSFER AND CHECK
                                ;
                                ;TEST 16 = TABLE CONTENT INCREMENTING PATTERN FROM 0
                                ;TEST 17 = EACH LOCATION CONTENT ADDRESS VALUE OF LOCATION
                                STARS 1



                                BGNTST
                                DMAIN:  JSR     PC,CLRKMV               ;CLEAR REG
                                        JSR     PC,MAINM1               ;SET MAINT 1
                                BGNSUB
                                        MOV     #TTABLE,R1              ;POINT TX TABLE
                                        CLR     R2                      ;CLR TABLE
                                1$:     MOV     R2,(R1)+                ;MAKE AN INCREMENTING PATTERN FROM 0
                                        INC     R2
                                        CMP     #2000,R2                ;TABLE LENGTH=1K WORDS
                                        BNE     1$


                                        MOV     #TTABLE,@KMVP12         ;SET TX TABLE ADDRESS
                                        MOV     #2000,@KMVP14           ;SET TABLE LENGTH
                                        CLR     @KMVP10                 ;CLEAR EXTENDED ADDRESS
                                        JSR     R5,TSTNUB               ;SEND TEST NB 16
                                        .WORD   16


                                        WAITB   0,1                     ;WAIT FOR TEST EXCUTION
```

```
4260
4261
4262
4263 030542  004737  013152                JSR     PC,TSTERR               ;CHECK BSEL0 ,WHICH ERROR
4264 030546  000444                        BR      2$                      ;TEST OK
4265 030550  000423                        BR      3$                      ;TIMEOUT ERROR
4266 030552  000432                        BR      4$                      ;NO KMV11 ANSWER
4267
4268
4269
4270
4271
4272 030554  017737  161732  002330        MOV     @KMVP02,GOOD            ;READ GOOD DATA
4273 030562  017737  161726  002424        MOV     @KMVP04,BDDAT           ;READ BAD DATA
4274 030570  017737  161722  002420        MOV     @KMVP06,ADDR            ;READ ERROR ADDRESS
4275 030576  117737  161716  012426        MOVB    @KMVP10,EXADDR          ;READ EXTENDED ADDRESS
4276 030604                                ERRHRD  43,EM0020,PRDMA         ;DATA CMP ERROR DURING DMA IN TX
4277 030614                                ESCAPE  SUB
4278
4279
4280
4281
4282
4283 030620  005037  002330          3$:   CLR     GOOD
4284 030624                                ERRHRD  44,EM0016,PBSEL0        ;TIMEOUT ERROR
4285 030634                                ESCAPE  SUB
4286
4287
4288 030640  005037  002330          4$:   CLR     GOOD
4289 030644                                ERRHRD  45,EM0024,PBSEL0        ;NO KMV ANSWER
4290 030654                                ESCAPE  SUB
4291
4292
4293 030660  000240                  2$:   NOP
4294 030662                          ENDSUB
4295
4296
4297
4298 030664                          BGNSUB
4299 030666  004737  014730                JSR     PC,MAINM1
4300 030672  012704  002426                MOV     #TTABLE,R4              ;POINT TTABLE
4301 030676  012702  002000                MOV     #2000,R2                ;TABLE LENGTH
4302 030702  010401                  1$:   MOV     R4,R1
4303 030704  010124                        MOV     R1,(R4)+        ;TABLE LOCATION CONTENT TABLE LOCATION ADDRESS
4304 030706  005302                        DEC     R2
4305 030710  001374                        BNE     1$
4306
4307
4308 030712  012777  002426  161602        MOV     #TTABLE,@KMVP12         ;SEND TABLE ADDRESS
4309 030720  012777  002000  161576        MOV     #2000,@KMVP14           ;   "     "  LENGTH
4310 030726  004537  015010                JSR     R5,TSTNUB               ;SET TEST NB 17
4311 030732  000017                        .WORD   17
4312
4313
4314 030734                                WAITB   0,1                     ;WAIT FOR TEST EXECUTION
4315
4316
```

```
4317 030754  004737  013152                JSR     PC,TSTERR           ;CHECK BSEL0
4318 030760  000444                        BR      2$                  ;TEST OK
4319 030762  000423                        BR      3$                  ;TIMEOUT ERROR
4320 030764  000432                        BR      4$                  ;NO KMV ANSWER
4321
4322
4323
4324
4325 030766  017737  161520  002330        MOV     @KMVP02,GOOD        ;READ GOOD DATA
4326 030774  017737  161514  002424        MOV     @KMVP04,BDDAT       ;  ''   BAD    ''
4327 031002  017737  161510  002420        MOV     @KMVP06,ADDR        ;  ''   ERROR ADDRESS
4328 031010  117737  161504  012426        MOVB    @KMVP10,EXADDR      ;  ''   EXTENDED ADDRESS
4329
4330 031016                                ERRHRD  46,EM0020,PRDMA     ;DATA CMP ERROR
4331 031026                                ESCAPE  SUB
4332
4333
4334
4335 031032  005037  002330          3$:   CLR     GOOD
4336 031036                                ERRHRD  47,EM0016,PBSEL0    ;TIMEOUT ERROR
4337 031046                                ESCAPE  SUB
4338
4339
4340
4341
4342 031052  005037  002330          4$:   CLR     GOOD
4343 031056                                ERRHRD  48,EM0024,PBSEL0    ;NO KMV ANSWER
4344 031066                                ESCAPE  SUB
4345
4346
4347
4348
4349 031072  000240                  2$:   NOP
4350 031074                                ENDSUB
4351 031076                                ENDTST
```

```
                                BADHEAD
                                ;********************* TEST20 ******************
                                ;TEST DMA TRANSFER OUT KMV11
031100                          BADHEAD
                                ;********************* TEST20 ******************
                                
                                
                                
                                
                                
031100                          STARS 1
                                ;SET MAINT1   ;DCT11 DECODE AND CLEAR BSEL0
                                ;LOAD FIRST ADDRESS OF RX TABLE IN SEL12 AND RX TABLE LENGTH IN SEL14
                                ;EXTENDED ADDRESS IN BSEL10
                                
                                ;SET TEST NUMBER 20,21 ;DCT11 EXECUTE TEST
                                ;WHEN DONE CLEAR BSEL0 IF TEST OK
                                ;SET 200 IN BSEL0 IF TIMEOUT DURING TEST
                                ;
                                ;
                                ;
                                ;TEST 20 DESCRIPTION: DCT11 SEND IN DMA AN INCREMENTING PATTERN (OF 1K WORDS)
                                ;                     IN HOST MEMORY. THIS PATTERN STARS AT ADDRESS FOUND
                                ;                     IN SEL12 (RX TABLE)
                                ;                     WHEN DONE CLEAR BSEL0
                                ;                     HOST CHECK IF THE RECEIVE TABLE IS CORRECT
                                
                                
                                ;TEST 21 DESCRIPTION: IDEM BUT TABLE CONTENT ADDRESS VALUE OF EACH LOCATION
031100                          STARS 1
                                
                                
                                
                                
031100                          BGNTST
031100                          BGNSUB
031102  004737  014550          DMAOUT: JSR     PC,CLRKMV               ;CLEAR REG
031106  004737  014730                  JSR     PC,MAINM1               ;SET MAINT 1
031112  005037  002320                  CLR     FLAG
031116  012701  006426                  MOV     #RTABLE,R1              ;POINT RX TABLE
031122  012702  002000                  MOV     #2000,R2                        ;CLR RX TABLE
031126  005021                  1$:     CLR     (R1)+
031130  005302                          DEC     R2
031132  001375                          BNE     1$
                                
                                
                                
031134  012777  006426  161360          MOV     #RTABLE,@KMVP12         ;SET RX TABLE ADDRESS
031142  012777  002000  161354          MOV     #2000,@KMVP14           ;SET TABLE LENGTH
031150  105077  161344                  CLRB    @KMVP10                 ;CLEAR EXTENDED ADDRESS
031154  004537  015010                  JSR     R5,TSTNUB               ;SEND TEST NB 20
031160  000020                          .WORD   20
                                
                                
                                
031162                          WAITB   0,1                     ;WAIT FOR TEST EXECUTION
                                
                                
```

```
4408 031202  004737  013152              JSR     PC,TSTERR           ;CHECK  BSEL0;WHICH ERROR
4409 031206  000423                      BR      2$                  ;TEST OK
4410 031210  000402                      BR      5$                  ;TIMEOUT ERROR
4411 031212  000411                      BR      6$                  ;NO KMV ANSWER
4412 031214  000420                      BR      2$
4413
4414
4415 031216  005037  002330        5$:   CLR     GOOD
4416 031222                              ERRHRD  49,EM0016,PBSEL0     ;TIMEOUT ERROR
4417 031232                              ESCAPE  SUB
4418
4419
4420
4421 031236  005037  002330        6$:   CLR     GOOD
4422 031242                              ERRHRD  50,EM0024,PBSEL0     ;NO KMV ANSWER
4423 031252                              ESCAPE  SUB
4424
4425
4426
4427
4428
4429 031256  012701  006426        2$:   MOV     #RTABLE,R1          ;CHECK RX TABLE
4430 031262  005037  002422              CLR     GDDAT               ;1ST WORD
4431
4432 031266  010137  002420        3$:   MOV     R1,ADDR
4433 031272  023711  002422              CMP     GDDAT,(R1)          ;COMPARE
4434 031276  001431                      BEQ     4$                  ;GOOD BR
4435
4436 031300  011137  002424              MOV     (R1),BDDAT
4437
4438 031304  005737  002320              TST     FLAG
4439 031310  001007                      BNE     7$                  ;LOOK IF 1ST MESAGE OR EXTENDED ONE
4440 031312                              ERRHRD  51,EM0030,PRDMA
4441
4442 031322  005237  002320              INC     FLAG
4443 031326  000415                      BR      4$
4444
4445 031330                        7$:   ERRHRD  51,0,PDMAF          ;DATA CMP ERROR
4446 031340                              BREAK
4447 031342  005237  002320              INC     FLAG
4448 031346  022737  000010  002320      CMP     #10,FLAG            ;REPORT 10 FIRST ERROR
4449 031354  001002                      BNE     4$
4450 031356                              ESCAPE  SUB
4451
4452
4453
4454
4455 031362  005237  002422        4$:   INC     GDDAT               ;CHECK NEW LOCATION
4456 031366  062701  000002              ADD     #2,R1
4457 031372  022737  002000  002422      CMP     #2000,GDDAT         ;ALL DONE
4458 031400  001332                      BNE     3$
4459 031402                        ENDSUB
4460
4461
4462
4463
4464
```

```
                                        BGNSUB
031404                                          
031406  005037  002320                          CLR     FLAG
031412  004737  014730                          JSR     PC,MAINM1
031416  012777  006426  161076                  MOV     #RTABLE,@KMVP12         ;LOAD RX TABLE IN SEL12
031424  005077  161070                          CLR     @KMVP10
031430  012777  002000  161066                  MOV     #2000,@KMVP14           ;LOAD TABLE LENGTH
                                                
                                                
031436  012702  002000                          MOV     #2000,R2                ;TABLE LENGTH
031442  012701  006426                          MOV     #RTABLE,R1
031446  005021                          10$:    CLR     (R1)+                   ;CLEAR RX TABLE
031450  005302                                  DEC     R2
031452  001375                                  BNE     10$
                                                
                                                
                                                
                                                
031454  004537  015010                          JSR     R5,TSTNUB               ;LOAD TEST NB21
031460  000021                                  .WORD   21
                                                
031462                                          WAITB   0,1                     ;WAIT FOR TEST EXECUTION
                                                
                                                
031502  004737  013152                          JSR     PC,TSTERR               ;CHECK BSEL0;WHICH ERROR
031506  000423                                  BR      2$                      ;TEST OK
031510  000402                                  BR      5$                      ;TIMEOUT ERROR
031512  000411                                  BR      6$                      ;NO ANSWER
031514  000420                                  BR      2$                      ;DATA CMP ERROR
                                                
                                                
                                                
031516  005037  002330                  5$:     CLR     GOOD
031522                                          ERRHRD  52,EM0016,PBSEL0        ;TIMEOUT ERROR
031532                                          ESCAPE  SUB
                                                
                                                
                                                
031536  005037  002330                  6$:     CLR     GOOD
031542                                          ERRHRD  53,EM0024,PBSEL0        ;NO KMV11 ANSWER
031552                                          ESCAPE  SUB
                                                
                                                
                                                
                                                
                                                
                                                
031556  012702  002000                  2$:     MOV     #2000,R2
031562  012737  006426  002420                  MOV     #RTABLE,ADDR            ;VERIFY RX TABLE
031570  012737  006426  002422                  MOV     #RTABLE,GDDAT
                                                
031576  023737  002422  002420          3$:     CMP     GDDAT,ADDR              ;CMP TABLE
031604  001432                                  BEQ     4$
031606  017737  150606  002424                  MOV     @ADDR,BDDAT             ;READ BAD DATA
                                                
                                                
031614  005737  002320                          TST     FLAG
031620  001007                                  BNE     1$                      ;LOOK IF 1ST REPORT
```

```
4522
4523
4524 031622                                    ERRHRD  54,EM0030,PRDMA         ;DATA CMP ERROR IN RX TABLE
4525 031632  005237  002320                    INC     FLAG
4526 031636  000415                            BR      4$
4527
4528 031640                            1$:     ERRHRD  54,0,PDMAF              ;SHORT ERROR REPORT
4529 031650                                    BREAK
4530 031652  005237  002320                    INC     FLAG
4531 031656  022737  000010  002320            CMP     #10,FLAG                ;REPORT 10 ERROR
4532 031664  001002                            BNE     4$
4533 031666                                    ESCAPE  SUB
4534
4535
4536 031672  062737  000002  002422  4$:     ADD     #2,GDDAT                ;VERIFY NEXT LOCATION
4537 031700  062737  000002  002420            ADD     #2,ADDR
4538 031706  005302                            DEC     R2                      ;ALL DONE:
4539 031710  001332                            BNE     3$                      ;NO BRANCH
4540
4541 031712                            ENDSUB
4542 031714                            ENDTST
4543
```

```
                                BADHEAD
                                ;********************* TEST21 ******************
                                ;TEST DMA TRANSFER IN BOTH DIRECTION
                                BADHEAD
                                ;********************* TEST21 ******************
                                STARS 1
                                ;SET MAINT1    ;  DCT11 DECODE AND CLEAR BSEL0
                                ;THE HOST SET ALL THE PARAMETERS IN CSR'S
                                ;LOAD TX TABLE ADDRESS IN SEL12,TABLE LENGTH IN SEL14,EXTENDED ADDRESS IN BSEL10
                                ;EXTENDED ADDRESS OF RX TABLE IN BSEL2 , ADDRESS OF RX TABLE IN SEL4 AND
                                ;RAM STARTING ADDRESS FOR TRANSFER IN SEL6.
                                ;
                                ;
                                ;LOAD TEST NUMBER 22  ;  DCT11 EXECUTE TEST
                                ;  WHEN DONE CLEAR BSEL0 IF TEST OK OR SET 200 IN BSEL0 IF TIMEOUT DURING DMA.
                                ;
                                ;TEST DESCRIPTION:
                                ; HOST COMPUTER GENERATES DIFFERENT 1K WORD TABLES ,GIVES ALL PARAMETERS IN
                                ; THE CSR'S AND SET TEST 22 IN BSEL0
                                ; DCT11 TAKES SEL6 AS THE STARTING ADDRESS FOR THE DIFFERENT TRANSFERS IN KMV11
                                ; RAM MEMORY (DMA INTO KMV11) AND TRANSFER THIS TABLE IN DMA BACK TO HOST
                                ; MEMORY (DMA OUT).
                                ;
                                ; DATA TRANSFER ARE MADE IN DIFFERENT AREAS IN RAM AND  DCT11 CHECKS
                                ; THAT THE UNUSED PART OF THE RAM IS NOT MODIFIED
                                ;
                                ; WHEN TRANSFER IN BOTH DIRECTION HAS BEEN DONE ,DCT11 CLEAR BSEL0 AND
                                ; HOST COMPARES RX AND TX TABLE
                                ;
                                ;ERROR REPORT IN BSEL0:          200=TIMEOUT DURING DMA
                                ;                                100=UNUSED MEMORY MODIFIED DURING TRANSFER
                                ;                                        IN THAT CASE SEL2 =GOOD
                                ;                                                     SEL4 =BAD
                                ;                                                     SEL6 = ADDRESS
                                ;
                                ;
                                STARS 1
                                BGNTST
031716  012737  065000  012462          MOV     #65000,MAXCNT           ;RAM MEMORY MAX LENGTH
031724  005037  002320                  CLR     FLAG
                                ;
031730  012703  000002                  MOV     #2,R3                   ;SELECT 1ST PATTERN
031734  004737  014550          DMATWO: JSR     PC,CLRKMV               ;CLEAR REG
031740  005037  002414                  CLR     COUNT                   ;SELECT RAM STARTING ADDRESS FOR TX
031744                                  BREAK
```

```
4600 031746  004737  014750          TWODMA: JSR     PC,MAINM1               ;SET MAINT 1
4601
4602
4603
4604 031752  012702  002000                  MOV     #2000,R2
4605 031756  012701  006426                  MOV     #RTABLE,R1
4606 031762  005021                  10$:    CLR     (R1)+                   ;CLEAR RX TABLE
4607 031764  005302                          DEC     R2
4608 031766  001375                          BNE     10$
4609
4610
4611
4612
4613
4614
4615 031770  012702  002000                  MOV     #2000,R2
4616 031774  012701  002426                  MOV     #TTABLE,R1
4617 032000  004737  013230          1$:     JSR     PC,GENER                ;MAKE A PATTERN
4618
4619 032004  013721  012436                  MOV     DATA,(R1)+              ;WRITE ONE TABLE LOCATION
4620 032010  005302                          DEC     R2                      ;ALL LOCATION DONE?
4621 032012  001372                          BNE     1$                      ;NO
4622
4623 032014  005077  160500                  CLR     @KMVP10                 ;CLEAR EXTENDED ADDRESS
4624 032020  013777  002414  160500          MOV     COUNT,@KMVP16           ;LOAD STATING ADDRESS IN RAM
4625
4626 032026  012777  002426  160466          MOV     #TTABLE,@KMVP12         ;SEND TX TABLE ADDRESS
4627 032034  012777  002000  160462          MOV     #2000,@KMVP14           ;SEND TABLE LENGTH
4628 032042  012777  006426  160444          MOV     #RTABLE,@KMVP04         ;SEND RX TABLE IN SEL4
4629 032050  005077  160436                  CLR     @KMVP02                 ;CLR RX TABLE EXT ADDRESS
4630 032054  004537  015010                  JSR     R5,TSTNUB               ;LOAD TEST NB 22
4631 032060  000022                          .WORD   22
4632
4633 032062  012737  070000  002324          MOV     #70000,DELCT1           ; SET DELAY COUNTER
4634 032070  117700  160414          11$:    MOVB    @KMVCSR,R0              ; GET BSEL0
4635 032074  105700                          TSTB    R0                      ; SEE IF TEST DONE OR CSR/DMA INTERFERENCE
4636 032076  001452                          BEQ     3$                      ; CHECK XMT/RCV BUFFER
4637 032100  005237  002324                  INC     DELCT1                  ; UPDATE TIMEOUT COUNTER
4638 032104  001371                          BNE     11$                     ; BR IF NOT TIMED OUT
4639 032106  000407                          BR      6$                      ; TIME-OUT
4640
4641 032110                                  BREAK
4642
4643
4644 032112  004737  013152                  JSR     PC,TSTERR               ;CHECK BSEL0;WHICH ERROR
4645 032116  000442                          BR      3$                      ;TEST OK
4646 032120  000402                          BR      6$                      ;TIME OUT
4647 032122  000411                          BR      7$                      ;NO KMV11 ANSWER
4648 032124  000420                          BR      20$                     ;PROBLEM IN THE UNUSED PART OF RAM:
4649                                                                         ;DMA TRANSFER MODIFY UNUSED RAM
4650                                                                         ; LOCATIONS.
4651
4652
4653
4654 032126  005037  002330          6$:     CLR     GOOD
4655 032132                                  ERRHRD  55,EM0016,PBSEL0        ;TIMEOUT ERROR
4656 032142                                  ESCAPE  TST
```

```
4657
4658
4659
4660 032146  005037  002330          7$:     CLR     GOOD
4661 032152                                  ERRHRD  56,EM0024,PBSEL0        ;NO KMV11 ANSWER
4662 032162                                  ESCAPE  TST
4663
4664
4665
4666 032166  017737  160324  002420  20$:    MOV     @KMVP06,ADDR            ;READ ADD IN ERROR
4667 032174  017737  160312  002330          MOV     @KMVP02,GOOD            ;GOOD VALUE
4668 032202  017737  160306  002424          MOV     @KMVP04,BDDAT           ;READ WRONG VALUE
4669 032210                                  ERRHRD  57,EM0033,PDMARA        ;DATA  ERROR IN RAM DURING TRANSFER
4670 032220                                  ESCAPE  TST
4671
4672
4673
4674
4675 032224  005077  160260          3$:     CLR     @KMVCSR
4676 032230  000240                          NOP
4677 032232  012777  044000  160250          MOV     #MAINT1,@KMVCSR         ;STOP TEST 22 IN KMV
4678 032240  012701  002426                  MOV     #TTABLE,R1              ;TX TABLE ADDRESS
4679 032244  012704  006426                  MOV     #RTABLE,R4              ;RX     "       "
4680 032250  012702  002000                  MOV     #2000.R2                ;TABLE LENGHT
4681
4682
4683
4684 032254  021114                  4$:     CMP     (R1),(R4)               ;CMP RX TABLE AND TX TABLE
4685 032256  001437                          BEQ     5$                      ;OK TEST NEXT LOCATION
4686
4687
4688 032260  011137  002422                  MOV     (R1),GDDAT              ;PREPARE ERROR REPORT
4689 032264  011437  002424                  MOV     (R4),BDDAT
4690 032270  010437  002420                  MOV     R4,ADDR
4691 032274  005037  012426                  CLR     EXADDR
4692
4693 032300  005737  002320                  TST     FLAG
4694 032304  001007                          BNE     2$
4695 032306                                  ERRHRD  58,EM0021,PRDMA         ;DATA CMP ERROR IN TABLE
4696 032316  005237  002320                  INC     FLAG
4697 032322  000415                          BR      5$
4698
4699
4700 032324                          2$:     ERRHRD   8,0,PDMAF              ;REPORT 10 FIRST ERROR
4701 032334                                  BREAK
4702 032336  005237  002320                  INC     FLAG
4703 032342  022737  000010  002320          CMP     #10,FLAG
4704 032350  001002                          BNE     5$
4705 032352                                  ESCAPE  TST
4706
4707
4708
4709
4710 032356  005721                  5$:     TST     (R1)+
4711 032360  005724                          TST     (R4)+
4712 032362  005302                          DEC     R2                      ;ALL MEMORY TESTED?
4713 032364  001333                          BNE     4$                      ;NO BRANCH
```

```
4714
4715
4716 032366  062737  002000  002414          ADD     #2000,COUNT             ;USE OTHER PART OF RAM
4717 032374  023737  002414  012462          CMP     COUNT,MAXCNT            ;IS ALL RAM USED?
4718 032402  100002                          BPL     30$
4719 032404  000137  031746                  JMP     TWODMA
4720
4721
4722
4723 032410  005203                  30$:    INC     R3                      ;SELECT NEW KIND OF PATTERN
4724 032412  022703  000005                  CMP     #5,R3                   ;ALL DONE?
4725 032416  001402                          BEQ     40$                     ;NO BRANCH
4726 032420  000137  031746                  JMP     TWODMA
4727 032424                          40$:
4728 032424                          ENDTST
4729
```

```
                                BADHEAD
                                ;********************* TEST22 ******************
                                ;TEST INTERUPT CAPABILITY OF KMV11 MODULE ON QBUS
                                BADHEAD
                                ;********************* TEST22 ******************



                                STARS 1
                                ;SET MAINT1   ; KMV11 DECODE AND CLEAR BSEL0
                                ;HOST PREPARES VECTOR AREA
                                ;SEND TEST NUMBER (23 OR 24)
                                ;DCT11 INTERUPTS THE HOST BY SETTING BITS 5 OR 6 IN ADDRESS 140000 OF
                                ;KMV11 MICRO BUS ;DCT11 CLEAR BSEL0 WHEN TEST COMPLETED.
                                ;
                                ;HOST TESTS IF THE INTERUPT HAS BEEN RECEIVED WITH CORRECT VECTOR
                                ;
                                ;
                                ;
                                ;
                                ;MICRO TEST 23 =INTERUPT ON LOW VECTOR
                                ;MICRO TEST 24 =INTERUPT ON HIGH VECTOR
                                STARS 1



                                BGNTST
032426  004737  014550          JSR     PC,CLRKMV               ;CLR REG
032432  004737  014730          JSR     PC,MAINM1
                                BGNSUB
032440  005037  012430          CLR     INTFLG
032444  013702  012476          MOV     KMVLVL,R2               ;READ KMV PRIORITY
032450  012777  032534  160016  MOV     #INT1,@KMVV00           ;SET UP VECTOR 0
032456  006202                  ASR     R2
032460  006202                  ASR     R2
032462  006202                  ASR     R2
032464  006202                  ASR     R2
032466  012777  000340  160006  MOV     #340,@KMVV02            ;SET KMV PRIORITY 7 FOR INTERUPT



032474  012703  000340          MOV     #340,R3                 ;TRY PRIORITY 7 FOR PROCESSOR

032500  106403          SETPR1: MTPS    R3                      ;LOAD PRIORITY
032502  004537  015010          JSR     R5,TSTNUB               ;SEND TEST 23
032506  000023                  .WORD   23

032510  000240                  NOP
032512  000240                  NOP
032514  000240                  NOP
032516  000240                  NOP
032520  000240                  NOP
```

```
4786 032522  000240                        NOP
4787 032524  000240                        NOP
4788
4789 032526                                BREAK
4790 032530  000137  032544                JMP     VECTO
4791
4792
4793                              ;***************INTERUPT  ROUTINE  ******************************
4794
4795 032534  052737  000001  012430  INT1:  BIS     #1,INTFLG              ;SET INT FLAG
4796 032542  000002                        RTI
4797
4798
4799                              ;*************************************************************************
4800
4801
4802
4803 032544  004537  013722          VECTO: JSR     R5,CBSELO              ;CHECK IF KMV11 ANSWER
4804 032550  000000                        .WORD   0
4805 032552  000410                        BR      2$
4806 032554                                ERRHRD  59,EM0024              ;NO KMV11 ANSWER
4807 032564  004737  013010                JSR     PC,CHKMAX              ;CHECK IF TOO MANY ERROR
4808 032570                                ESCAPE  SUB
4809
4810
4811 032574  005737  012430          2$:   TST     INTFLG                 ;TEST IF INTERUPT ?
4812 032600  001454                        BEQ     3$
4813
4814 032602  010237  002330                MOV     R2,GOOD                ;GOOD INTERUPT LEVEL
4815
4816
4817
4818 032606  062703  000040                ADD     #40,R3                 ;WAS IT LEGAL
4819 032612  010337  012432                MOV     R3,BAD
4820 032616  023737  012432  002330        CMP     BAD,GOOD
4821 032624  001461                        BEQ     4$                     ;YES BRANCH
4822
4823 032626  106237  002330                ASRB    GOOD
4824 032632  106237  002330                ASRB    GOOD
4825 032636  106237  002330                ASRB    GOOD
4826 032642  106237  002330                ASRB    GOOD
4827 032646  106237  002330                ASRB    GOOD
4828 032652  042737  177770  002330        BIC     #177770,GOOD
4829
4830
4831 032660  106237  012432                ASRB    BAD
4832 032664  106237  012432                ASRB    BAD
4833 032670  106237  012432                ASRB    BAD
4834 032674  106237  012432                ASRB    BAD
4835 032700  106237  012432                ASRB    BAD
4836 032704  042737  177770  012432        BIC     #177770,BAD
4837 032712                                ERRHRD  60,EM0022
4838 032722  004737  013010                JSR     PC,CHKMAX
4839 032726                                ESCAPE  SUB
4840
4841
4842
```

```
4843 032732  122703  000140          3$:     CMPB    #140,R3             ;IS PROCESSOR AT LEVEL 3
4844 032736  001404                          BEQ     5$
4845 032740  162703  000040                  SUB     #40,R3              ;DECREASE PRIORITY
4846 032744  000137  032500                  JMP     SETPR1              ;TRY WITH NEW ONE
4847
4848
4849
4850 032750                          5$:     ERRHRD  61,EM0023           ;NO INTERUPT OCCUR
4851 032760  004737  013010                  JSR     PC,CHKMAX           ;CHECK IF TOO MANY ERROR
4852 032764                                  ESCAPE  SUB
4853
4854 032770                          4$:
4855 032770                          ENDSUB
4856
4857
4858 032772                          BGNSUB
4859 032774  004737  014730                  JSR     PC,MAINM1
4860 033000  005037  012430                  CLR     INTFLG
4861 033004  013702  012476                  MOV     KMVLVL,R2           ;SET PRIORITY LEVEL
4862 033010  012777  000340  157466          MOV     #340,@KMVV06
4863 033016  012777  033070  157454          MOV     #INT2,@KMVV04       ;SET UP VECTOR 4
4864 033024  006202                          ASR     R2
4865 033026  006202                          ASR     R2
4866 033030  006202                          ASR     R2
4867 033032  006202                          ASR     R2
4868 033034  012703  000340                  MOV     #340,R3             ;START WITH PRIORITY 7 FOR PROCESSOR
4869
4870
4871
4872
4873 033040  106403                  INTPR2: MTPS    R3                  ;LOAD PRIORITY
4874 033042  004537  015010                  JSR     R5,TSTNUB
4875 033046  000024                          .WORD   24                  ;SET TEST NB 24
4876 033050  000240                          NOP
4877 033052  000240                          NOP
4878 033054  000240                          NOP
4879 033056  000240                          NOP
4880 033060  000240                          NOP
4881
4882 033062                                  BREAK
4883 033064  000137  033100                  JMP     VECT4
4884
4885
4886                                 ;****************INTERUPT ROUTINE ***********************************
4887
4888
4889 033070  052737  000001  012430  INT2:   BIS     #1,INTFLG           ;SET FLAG
4890 033076  000002                          RTI
4891
4892
4893                                 ;*************************************************************************
4894
4895
4896
4897 033100  004537  013722          VECT4:  JSR     R5,CBSEL0           ;IS THERE KMV11 ANSWER ?
4898 033104  000000                          .WORD   0
4899 033106  000410                          BR      2$
```

```
4900 033110                                    ERRHRD  62,EM0024               ;NO KMV11 ANSWER
4901 033120  004737  013010                    JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
4902 033124                                    ESCAPE SUB
4903
4904
4905
4906 033130  032737  000001  012430  2$:      BIT     #1,INTFLG               ;TEST IF INTERUPT OCCUR
4907 033136  001454                            BEQ     3$                      ;NO INTERUPT
4908
4909 033140  010237  002330                    MOV     R2,GOOD                 ;GOOD INTERUPT LEVEL
4910
4911
4912 033144  062703  000040                    ADD     #40,R3                  ;WAS IT LEGAL ?
4913 033150  010337  012432                    MOV     R3,BAD
4914
4915 033154  023737  012432  002330            CMP     BAD,GOOD
4916 033162  001461                            BEQ     4$                      ;YES BRANCH
4917 033164  106237  002330                    ASRB    GOOD
4918 033170  106237  002330                    ASRB    GOOD
4919 033174  106237  002330                    ASRB    GOOD
4920 033200  106237  002330                    ASRB    GOOD
4921 033204  106237  002330                    ASRB    GOOD
4922 033210  042737  177770  002330            BIC     #177770,GOOD            ;GET ACTUAL LEVEL
4923
4924 033216  106237  012432                    ASRB    BAD
4925 033222  106237  012432                    ASRB    BAD
4926 033226  106237  012432                    ASRB    BAD
4927 033232  106237  012432                    ASRB    BAD
4928 033236  106237  012432                    ASRB    BAD
4929 033242  042737  177770  012432            BIC     #177770,BAD
4930 033250                                    ERRHRD  63,EM0022               ;INT OCCUR AT BAD LEVEL
4931 033260  004737  013010                    JSR     PC,CHKMAX
4932 033264                                    ESCAPE SUB
4933
4934
4935
4936
4937 033270  122703  000040          3$:      CMPB    #40,R3                  ;IS PROCESSOR AT PRIORITY 3
4938 033274  001404                            BEQ     5$                      ;YES,NO INTERUPT OCCURED
4939 033276  162703  000040                    SUB     #40,R3                  ;DECRESE PRIORITY LEVEL
4940 033302  000137  033040                    JMP     INTPR2
4941
4942
4943 033306                          5$:       ERRHRD  64,EM0023
4944 033316  004737  013010                    JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
4945 033322                                    ESCAPE  SUB
4946 033326                          4$:
4947 033326                          ENDSUB
4948 033330                          ENDTST
```

```
                                ;********************** TEST23 ********************
                                ;TEST INTERUPT ON DCT11 MICROPROCESSOR
                                ;********************** TEST23 ********************
                                STARS 1
                                ;CHECKS THAT QBUS ACCESS ON BSEL0 AND BSEL10 CAUSE AN INTERUPT ON DCT11
                                ;CHECKS THAT ACCESSES ON ALL THE OTHER CSR'S DOES NOT CAUSE ANY INTERUPTS.
                                ;
                                ;
                                ;TEST DESCRIPTION:
                                ;TEST NUMB 25:  DCT11 INITIALIZE VECTOR 60 ON DCT11 BUS CORRESPONDING TO
                                ;               BSEL0 INTERUPT
                                ;
                                ;               QBUS ACCESS ALL REGISTERS BUT BSEL0  AND CHECK THAT NO
                                ;               INTERUPT OCCUR ON DCT11
                                ;
                                ;
                                ;
                                ;               CHECK THAT QBUS ACCESS ON BSEL0 GIVE AN INTERUPT ON VECTOR 60
                                ;
                                ;
                                ;TEST NUMB 26:  DCT11 INITIALIZE VECTOR 70 CORRESPONDING TO BSEL2
                                ;               INTERUPT
                                ;
                                ;               QBUS ACCESS ALL REGISTERS BUT BSEL2 AND CHECK NO INTERUPT
                                ;               OCCUR ON DCT11
                                ;
                                ;               CHECK THAT QBUS ACCES ON BSEL2 INTERUPT ON VECTOR 70
                                ;
                                ;ERROR REPORTING:       BSEL0=0         IF INTERUPT OCCUR
                                ;                       BSEL0=100       IF ILLEGAL VECTOR
                                ;                       BSEL0=TST NB    IF NO INTERUPT
                                ;                       SEL2 =          EXPECTED VECTOR
                                STARS 1
```

```
                                BGNTST
                                BGNSUB
033334  004737  014550                 JSR     PC,CLRKMV       ;CLEAR REG
033340  004737  014730                 JSR     PC,MAINM1       ;SET MAINT MODE
033344  004537  015010                 JSR     R5,TSTNUB       ;SET TEST NB 25
033350  000025                         .WORD   25

033352                                 WAITB   0,1

033372  013701  012512                 MOV     KMVP02,R1       ;LOAD CSR ADDR
033376  012702  000012                 MOV     #12,R2          ;ACCES BSEL2 TO BSEL16

033402  152721  000207         1$:     BISB    #207,(R1)+      ;WRITE ALL REG BUT BSEL0

033406                                 WAITB   0,1             ;WAIT FOR TEST EXECUTION

033426  004537  013722                 JSR     R5,CBSEL0       ;LOOK IF INTERUPT OCCUR
033432  000000                         .WORD   0

033434  000404                         BR      3$              ;YES SEE WHICH ERROR
033436  005302                         DEC     R2              ;ALL REG DONE ?
033440  001360                         BNE     1$              ;NO BR


033442  000137  033474                 JMP     GOON1           ;OK NO ACCESS INTERUPT THE DC111 ;GO ON


033446  010137  002420         3$:     MOV     R1,ADDR         ;SEE WHICH ADDRESS CAUSE INTERUPT
033452  162737  000001  002420         SUB     #1,ADDR
033460                                 ERRHRD  65,EM0026,PINTR ;WRONG INTERUPT OCCURED ON DCT11
                                                               ;WHILE ADDRESSING KMV11 REGISTERS
033470                                 ESCAPE  SUB


033474  052777  004025  157006 GOON1:  BIS     #4025,@KMVCSR   ;ACCESS BSEL0

033502                                 WAITB   0,1

033522  004537  013722                 JSR     R5,CBSEL0
033526  000000                         .WORD   0
```

```
5054 033530  000424                        BR      5$              ;TEST OK ,INTRUPT OCCURED AT GOOD VECTOR
5055
5056 033532  004537  013722                JSR     R5,CBSEL0
5057 033536  000100                        .WORD   100
5058 033540  000410                        BR      6$              ;INT ON ILLEGAL VECTOR
5059 033542                                ERRHRD  66,EM0027       ;NO KMV11 ANSWER, DCT11 DOES NOT RECEIVE ANY
5060 033552  004737  013010                JSR     PC,CHKMAX       ;CHECK IF TOO MANY ERROR
5061 033556                                ESCAPE  SUB             ;INTERUPT WHEN QBUS ADDRESS CSR'S
5062
5063
5064
5065
5066 033562                        6$:     ERRHRD  67,EM0032       ;INT ON ILLEGAL VECTOR WHEN ACCESSING BSEL0
5067 033572  004737  013010                JSR     PC,CHKMAX       ;CHECK IF TOO MANY ERROR
5068 033576                                ESCAPE  SUB
5069
5070 033602  000240                5$:     NOP
5071 033604                        ENDSUB
5072
5073
5074
5075
5076
5077 033606                        BGNSUB
5078 033610  004737  014730                JSR     PC,MAINM1       ;SET MAINT MODE
5079 033614  004537  015010                JSR     R5,TSTNUB       ;SET TEST NB 26
5080 033620  000026                        .WORD   26
5081
5082
5083 033622                                WAITB   0,1
5084
5085 033642  052777  000026  156640        BIS     #26,@KMVCSR     ;WRITE SEL0
5086
5087 033650  013701  012514                MOV     KMVP04,R1       ;LOAD CSR ADDR
5088 033654  012702  000010                MOV     #10,R2          ;ACCES BSEL3 TO BSEL11
5089
5090 033660  152721  000207          1$:   BISB    #207,(R1)+      ;WRITE ALL REG BUT BSEL2
5091
5092 033664                                WAITB   0,1             ;WAIT FOR TEST EXECUTION
5093
5094 033704  004537  013722                JSR     R5,CBSEL0       ;LOOK IF INTERUPT OCCUR
5095 033710  000000                        .WORD   0
5096
5097 033712  000404                        BR      3$              ;YES SEE WHICH ERROR
5098 033714  005302                        DEC     R2              ;ALL REG DONE ?
5099 033716  001360                        BNE     1$              ;NO BR
5100
5101
5102
5103
5104
5105 033720  000137  033756                JMP     GOON2           ;OK NO ACCESS INTERUPT THE DCT11 ;GO ON
5106
5107
5108 033724  017737  156564  012440  3$:   MOV     @KMVP04,VECT    ;READ RECEIVE VECTOR
5109 033732  010137  002420                MOV     R1,ADDR         ;SEE WHICH ADDRESS MAKE INTERUPT
5110 033736  005337  002420                DEC     ADDR
```

```
5111 033742                                ERRHRD  68,EM0026,PINTR ;WRONG INTERUPT OCCUR WHILE ACCESSING REGISTERS
5112 033752                                ESCAPE  SUB
5113
5114
5115 033756  000240              GOON2:    NOP
5116
5117 033760  052777  017777  156524       BIS     #17777,@kMVP02  ;ACCESS BSEL2
5118
5119 033766                                WAITB   0,1
5120
5121 034006  004537  013722               JSR     R5,CBSEL0
5122 034012  000000                       .WORD   0
5123 034014  000424                       BR      5$              ;TEST OK ,INTRUPT OCCUR AT GOOD VECTOR
5124
5125 034016  004537  013722               JSR     R5,CBSEL0
5126 034022  000100                       .WORD   100
5127 034024  000410                       BR      6$              ;INT ON ILLEGAL VECTOR
5128 034026                                ERRHRD  69,EM0027       ;NO KMV11 ANSWER
5129 034036  004737  013010               JSR     PC,CHKMAX       ;CHECK IF TOO MANY ERROR
5130 034042                                ESCAPE  SUB
5131
5132
5133
5134
5135
5136 034046                        6$:     ERRHRD  70,EM0026               ;INT ON ILLEGAL VECTOR
5137 034056  004737  013010               JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
5138 034062                                ESCAPE  SUB
5139
5140 034066  000240                5$:     NOP
5141 034070                                ENDSUB
5142
5143
5144
5145 034072                                ENDTST
5146
```

```
                                              .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.
                                              ;//////////////////////////////////////////////////////////////////////

034074                                                BGNHRD

034076                                                GPRMA   ADDRES,0,0,60000,177776,YES
034106                                                GPRMA   VECTOR,2,0,0,674,YES
034116                                                GPRMD   PRIRTY,4,0,7000,4,7,YES
034130                                                ENDHRD



034130     115     111     103        ADDRES: .ASCIZ  /MICRO-CPU  CSR ADDRESS : /
034133     122     117     055
034136     103     120     125
034141     040     040     103
034144     123     122     040
034147     101     104     104
034152     122     105     123
034155     123     040     072
034160     040     000
034162     115     111     103        VECTOR: .ASCIZ  /MICRO-CPU VECTOR ADDRESS : /
034165     122     117     055
034170     103     120     125
034173     040     126     105
034176     103     124     117
034201     122     040     101
034204     104     104     122
034207     105     123     123
034212     040     072     040
034215     000
034216     115     111     103        PRIRTY: .ASCIZ  /MICRO-CPU PRIORITY LEVEL : /
034221     122     117     055
034224     103     120     125
034227     040     120     122
034232     111     117     122
034235     111     124     131
034240     040     114     105
034243     126     105     114
034246     040     072     040
034251     000

                                              .EVEN
```

```
5185
5186
5187
```

```
                                  .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.
                                  ;//////////////////////////////////////////////////////////////////

 034252                                   BGNSFT



 034254                                   ENDSFT




```

```
,225
5226 034254                             $PATCH::
5227 034254                                     .BLKW   50
5228
5235
5236 034374                                     LASTAD
     034400                             LSLAST::
5237 034400                                     ENDMOD
5238
5239
```

```
5241
5242
5255
5256 034400                                  BGNSETUP        1
5257 034400                                  BGNPTAB
5258 034404  177000                          .WORD   177000
5259 034406  000300                          .WORD   300
5260 034410  004000                          .WORD   4000
5261 034412  000001                          .WORD   1
5262 034414                                  ENDPTAB
5263 034414                                  ENDSETUP
5264
5265
5266
5267
5268
5269         000001              .END
```

SYMBOL TABLE

```
ABORT   024010          C$DCLN= 000044          EF.RES= 000037 G        F$RPT = 000012          G$RADL= 000120
ADDR    002420          C$DODU= 000051          EF.STA= 000040 G        F$SEG = 000003          G$RADO= 000020
ADDRES  034130          C$DRPT= 000024          EM0001  015265          F$SOFT= 000005          G$XFER= 000004
ADR     = 000020 G      C$DU  = 000053          EM0002  015361          F$SRV = 000010          G$YES = 000010
ASSEMB= 000010          C$EDIT= 000003          EM0003  015424          F$SUB = 000002          HELP  = 000000
BAD     012432          C$ERDF= 000055          EM0004  015512          F$SW  = 000014          HOE   = 100000 G
BDDAT   002424          C$ERHR= 000056          EM0005  015575          F$TEST= 000001          IBE   = 010000 G
BIT0    = 000001 G      C$ERRO= 000060          EM0006  015671          GDDAT   002422          IDU   = 000040 G
BIT00   = 000001 G      C$ERSF= 000054          EM0007  015764          GDREV   012464          IER   = 020000 G
BIT01   = 000002 G      C$ERSO= 000057          EM0010  016056          GENER   013230          INIFLG  012466
BIT02   = 000004 G      C$ESCA= 000010          EM0011  016152          GENER1  013352          INTFLG  012430
BIT03   = 000010 G      C$ESEG= 000005          EM0012  016246          GENEX   013510          INTPR2  033040
BIT04   = 000020 G      C$ESUB= 000003          EM0013  016342          GENE1   025502          INT1    032534
BIT05   = 000040 G      C$ETST= 000001          EM0015  016436          GENE2   025700          INT2    033070
BIT06   = 000100 G      C$EXIT= 000032          EM0016  016506          GENE3   026076          ISR   = 000100 G
BIT07   = 000200 G      C$GETB= 000026          EM0020  016544          GENE4   026306          IXE   = 004000 G
BIT08   = 000400 G      C$GETW= 000027          EM0021  016630          GENE5   026510          I$AU  = 000041
BIT09   = 001000 G      C$GMAN= 000043          EM0022  016722          GENE6   026712          I$AUTO= 000041
BIT1    = 000002 G      C$GPHR= 000042          EM0023  016762          GENINC  013502          I$CLN = 000041
BIT10   = 002000 G      C$GPLO= 000030          EM0024  017005          GENISH  013360          I$DU  = 000041
BIT11   = 004000 G      C$GPRI= 000040          EM0025  017075          GENRAN  013362          I$HRD = 000041
BIT12   = 010000 G      C$INIT= 000011          EM0026  017136          GENROT  013336          I$INIT= 000041
BIT13   = 020000 G      C$INLP= 000020          EM0027  017232          GENR0   013324          I$MOD = 000041
BIT14   = 040000 G      C$MANI= 000050          EM0028  017307          GENR1   013314          I$MSG = 000041
BIT15   = 100000 G      C$MEM = 000031          EM0030  017341          GENSEL  013246          I$PROT= 000040
BIT2    = 000004 G      C$MSG = 000023          EM0031  017423          GEN0    013266          I$PTAB= 000041
BIT3    = 000010 G      C$OPEN= 000034          EM0032  017477          GEN1    013272          I$PWR = 000041
BIT4    = 000020 G      C$PNTB= 000014          EM0033  017573          GEN25   013306          I$RPT = 000041
BIT5    = 000040 G      C$PNTF= 000017          EM0034  017654          GEN52   013300          I$SEG = 000041
BIT6    = 000100 G      C$PNTS= 000016          EM0035  020013          GETPRM  023606          I$SETU= 000041
BIT7    = 000200 G      C$PNTX= 000015          EM0134  017746          GOOD    002330          I$SFT = 000041
BIT8    = 000400 G      C$QIO = 000377          END     024100          GOOD0   002332          I$SRV = 000041
BIT9    = 001000 G      C$RDBU= 000007          ERRBLK  002264 G        GOOD1   002334          I$SUB = 000041
BOE     = 000400 G      C$REFG= 000047          ERRCNT  002300          GOOD10  002344          I$TST = 000041
BSEL0   012434          C$RESE= 000033          ERRMSG  002262 G        GOOD12  002346          J$JMP = 000167
BSEL1   002376          C$REVI= 000003          ERRNBR  002260 G        GOOD14  002350          KIND    012442
CBSEL0  013722          C$RFLA= 000021          ERRTYP  002256 G        GOOD16  002352          KMTLVL  012506
CHANEL  012444          C$RPT = 000025          EVL   = 000004 G        GOOD2   002336          KMVCSR  012510
CHECK   024650          C$SEFG= 000046          EXADDR  012426          GOOD4   002340          KMVLVL  012476
CHECK1  025014          C$SPRI= 000041          E$END = 002100          GOOD6   002342          KMVP02  012512
CHKMAX  013010          C$SVEC= 000037          E$LOAD= 000035          GOON1   033474          KMVP04  012514
CKALL   013760          C$TPRI= 000013          FLAG    002320          GOON2   033756          KMVP06  012516
CKREG   014262          DATA    012436          FTIME   002316          G$CNTO= 000200          KMVP10  012520
CKSEL0  013670          DATA1 = 052525 G        F$AU  = 000015          G$DELM= 000372          KMVP12  012522
CLRKMV  014550          DATA2 = 125252 G        F$AUTO= 000020          G$DISP= 000003          KMVP14  012524
COUNT   002414          DELCT1  002324          F$BGN = 000040          G$EXCP= 000400          KMVP16  012526
C$AU  = 000052          DELCT2  002326          F$CLEA= 000007          G$HILI= 000002          KMVV00  012474
C$AUTO= 000061          DFPTBL  002212 G        F$DU  = 000016          G$LOLI= 000001          KMVV02  012502
C$BRK = 000022          DH1     002322          F$END = 000041          G$NO  = 000000          KMVV04  012500
C$BSEG= 000004          DIAGMC= 000000          F$HARD= 000004          G$OFFS= 000400          KMVV06  012504
C$BSUB= 000002          DMAIN   030442          F$HW  = 000013          G$OFSI= 000376          KMV11A  002000 G
C$CEFG= 000045          DMAOUT  031102          F$INIT= 000006          G$PRMA= 000001          LENGTH  012454
C$CLCK= 000062          DMATWO  031734          F$JMP = 000050          G$PRMD= 000002          LOCK    002274
C$CLEA= 000012          DROPD   024226          F$MOD = 000000          G$PRML= 000000          LOE   = 040000 G
C$CLOS= 000035          EF.CON= 000036 G        F$MSG = 000011          G$RADA= 000140          LOGDEV  002302
C$CLP1= 000006          EF.NEW= 000035 G        F$PROT= 000021          G$RADB= 000000          LOKFLG  012470
C$CVEC= 000036          EF.PWR= 000034 G        F$PWR = 000017          G$RADD= 000040          LOOP    012530
```

SYMBOL TABLE

```
LOT    = 000010 G      L10002  022056      L10073  031712      O$SETU= 000001        SAVSP   002306
L$ACP    002110 G      L10003  022144      L10074  032424      PADFLT  022346 G      SAVSTA  002412
L$APT    002036 G      L10004  022206      L10075  033330      PBSEL0  023174 G      SEL0    002354
L$AU     024260 G      L10005  022312      L10076  032770      PCHECK  022314 G      SEL1    002356
L$AUT    002070 G      L10006  022344      L10077  033326      PDMAF   023354 G      SEL10   002366
L$AUTO   024102 G      L10007  022402      L10100  034072      PDMARA  022210 G      SEL12   002370
L$CCP    002106 G      L10010  022712      L10101  033604      PINTR   023232 G      SEL14   002372
L$CLEA   024174 G      L10011  023172      L10102  034070      PNT    = 001000 G     SEL16   002374
L$CO     002032 G      L10012  023230      L10103  034130      PRALL   022404 G      SEL2    002360
L$DEPO   002011 G      L10013  023262      L10104  034254      PRBYTE  022146 G      SEL4    002362
L$DESC   002222 G      L10014  023352      L10105  034404      PRDMA   023264 G      SEL6    002364
L$DESP   002076 G      L10015  023410      L10107  034414      PRI    = 002000 G     SETPR1  032500
L$DEVP   002060 G      L10016  023416      MAINM0  014660      PRIRTY  034216        SETUP   023524
L$DISP   002132 G      L10017  024100      MAINM1  014730      PRI00  = 000000 G     SSTACK  012732
L$DLY    002116 G      L10020  024172      MAINT0= 054000 G     PRI01  = 000040 G     SVCGBL= 000000
L$DTP    002040 G      L10021  024176      MAINT1= 044000 G     PRI02  = 000100 G     SVCINS= 177777
L$DTYP   002034 G      L10022  024256      MAXCNT  012462      PRI03  = 000140 G     SVCSUB= 177777
L$DU     024200 G      L10023  024260      MAXERR  002276      PRI04  = 000200 G     SVCTAG= 177777
L$DUT    002072 G      L10024  024376      MAXPRI= 000340 G     PRI05  = 000240 G     SVCTST= 177777
L$DVTY   012732 G      L10025  024572      MBSEL0  021103      PRI06  = 000300 G     S$LSYM= 010000
L$EF     002052 G      L10026  024514      MBYTE   021371      PRI07  = 000340 G     TBYTE   025102
L$ENVI   002044 G      L10027  024570      MCHECK  021666      PROMCK  030324        TCSR    024766
L$ERRT   002256 G      L10030  024732      MCLR   = 040000 G     PRRAM   022C60 G      TCSRNB  024756
L$ETP    002102 G      L10031  025074      MDMAF   021737      PRREG   022714 G      TFM36   015164
L$EXP1   002046 G      L10032  025226      MDMAR1  021453      PRSEL0  022022 G      TGENE1  025472
L$EXP4   002064 G      L10033  025454      MDMAR2  021542      PSTACK  002304        TGENE2  025670
L$EXP5   002066 G      L10034  025342      MDMAR3  021610      QV.FLG  012471        TGENE3  026066
L$HARD   034076 G      L10035  025452      MDMA1   021240      RAMADD  027724        TGENE4  026276
L$HIME   002120 G      L10036  025650      MDMA2   021323      RAMCAD  030076        TGENE5  026500
L$HPCP   002016 G      L10037  025544      MINT    021051      RAMPAT  027504        TGENE6  026702
L$HPTP   002022 G      L10040  025646      MINTR   021145      RANCLC  013462        TIM     015247
L$HW     002212 G      L10041  026046      MRAM1   020707      RANDN   002406        TSELA   024626
L$ICP    002104 G      L10042  025742      MRAM2   020771      RANEX   013500        TSELB   024632
L$INIT   023420 G      L10043  026044      MREG0   020107      RANGEN  013402        TSEL10  026176
L$LADP   002026 G      L10044  026256      MREG10  020323      RANMTA  002404        TSEL12  026400
L$LAST   034400 G      L10045  026152      MREG12  020366      RANSEC  013466        TSEL14  026602
L$LOAD   002100 G      L10046  026254      MREG14  020431      RANSEL  002402        TSEL16  027004
L$LUN    002074 G      L10047  026460      MREG16  020474      RANST   002400        TSEL4   025570
L$MREV   002050 G      L10050  026354      MREG2   020152      RAN1    013414        TSEL6   025766
L$NAME   002000 G      L10051  026456      MREG4   020215      RAN2    013432        TSPEED  012452
L$PRIO   002042 G      L10052  026662      MREG6   020260      RAN4    013470        TSTERR  013152
L$PROT   002122 G      L10053  026556      MSEL0   020042      READ    015072        TSTNUB  015010
L$PRT    002112 G      L10054  026660      MSEL10  020645      REGADR  012532        TSTREG  024574
L$REPP   002062 G      L10055  027064      MSFL2   020537      RESTST  024402        TTABLE  002426
L$REV    002010 G      L10056  026760      MSEL4   020602      REVPRO  030260        TWODMA  031746
L$RPT    023412 G      L10057  027062      NERRS   013102      RGALL   027110        TXDATA  012446
L$SOFT   034254 G      L10060  027502      NEXT    023532      ROMMAP  024262        T$ARGC= 000002
L$SPC    002056 G      L10061  027722      NUB     012456      RSTREG  013610        T$CODE= 002032
L$SPCP   002020 G      L10062  030074      NUMBER  002416      RTABLE  006426        T$ERRN= 000106
L$SPTP   002024 G      L10063  030246      O$APTS= 000000      RUNNIN  024016        T$EXCP= 000000
L$STA    002030 G      L10064  030312      O$AU   = 000000      RXCNT   012460        T$FLAG= 000040
L$SW     002266        L10065  030440      O$BGNR= 000000      RXDATA  012450        T$FREE= 034414
L$TEST   002114 G      L10066  031076      O$BGNS= 000000      SAVE4   002312        T$GMAN= 000000
L$TIML   002014 G      L10067  030662      O$DU   = 000001      SAVE6   002314        T$HILI= 000007
L$UIT    002270        L10070  031074      O$ERRT= 000000      SAVPC   002310        T$LAST= 000001
L$UNIT   002012 G      L10071  031714      O$GNSW= 000001      SAVPC1  002410        T$LOLI= 000004
L10001   002222        L10072  031402      O$POIN= 000001      SAVREG  013530        T$LSYM= 010000
```

```
KMV11 A/B LOGIC DIAG    MACRO M1200  06-JAN-83 09:39  PAGE 81-3
SYMBOL TABLE

T$LTNO= 000027          T$$AUT= 010020          T11.1   026462          T21     031716 G        T9.1    026050
T$NEST= 177777          T$$CLE= 010021          T11.2   026560          T22     032426 G        T9.2    026154
T$NS0 = 000000          T$$DAT= 010107          T12     026664 G        T22.1   032436          UAM   = 000200 G
T$NS1 = 000005          T$$DU = 010022          T12.1   026664          T22.2   032772          UNIT    002272
T$NS2 = 000002          T$$HAR= 010103          T12.2   026762          T23     033332 G        UUT     012472
T$PCNT= 000000          T$$HW = 010001          T13     027066 G        T23.1   033332          VECT    012440
T$PTAB= 010106          T$$INI= 010017          T14     027504 G        T23.2   033606          VECTOR  034162
T$PTHV= 000001          T$$MSG= 010015          T15     027724 G        T3      024574 G        VECTO   032544
T$PTNU= 000001          T$$PC = 000001          T16     030076 G        T4      024734 G        VECT4   033100
T$SAVL= 177777          T$$PRO= 010000          T17     030250 G        T5      025076 G        WAIT1   013000
T$SEGL= 177777          T$$PTA= 010106          T18     030314 G        T6      025230 G        WAIT2   012760
T$SIZE= 000006          T$$RPT= 010016          T19     030442 G        T6.1    025240          WRITE   015044
T$SUBN= 000002          T$$SOF= 010104          T19.1   030452          T6.2    025344          X$ALWA= 000000
T$TAGL= 177777          T$$SUB= 010102          T19.2   030664          T7      025456 G        X$FALS= 000040
T$TAGN= 010110          T$$TES= 010100          T2      024400 G        T7.1    025456          X$OFFS= 000400
T$TEMP= 000000          T1      024262 G        T2.1    024400          T7.2    025546          X$TRUE= 000020
T$TEST= 000027          T10     026260 G        T2.2    024516          T8      025652 G        $LSTIN= 000000
T$TSTM= 177777          T10.1   026260          T20     031100 G        T8.1    025652          $LSTTA= 000000
T$TSTS= 000001          T10.2   026356          T20.1   031100          T8.2    025744          $PATCH  034254 G
T$$AU = 010023          T11     026462 G        T20.2   031404          T9      026050 G

. ABS.  034414      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  29088 WORDS  ( 114 PAGES)
DYNAMIC MEMORY:  21924 WORDS  ( 84 PAGES)
ELAPSED TIME:  00:29:54
VKMAA0.BIN,VKMAA0=[64,3]LIBA.MLB/ML,[64,5]VKMAA0
```