```
                        .NLIST TOC
                        .REM @

                                IDENTIFICATION

                        PRODUCT CODE:   AC-1842A-MC

                        PRODUCT NAME:   CNKMDAO KMV11A/B LOGIC DIAG

                        PRODUCT DATE:   APRIL 1984

                        MAINTAINER:     ISS DIAGNOSTICS

                        AUTHOR:         MICHELET GUY

                        MODIFIED BY:    JAKI BERG       9-APR-1984
```

```
                        ***** MODIFICATION HISTORY *****

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

CVKMAA -> CNKMDA                     JAKI BERG                      9-APR-84
CHANGES WERE MADE TO CVKMAA TO PRODUCE CNDMDA FOR THE FALCON-PLUS PROJECT
(SBC 11/21+).   CHANGES, MARKED BY ";JB REV A-0", ARE:
 - SET THE  ODT BREAK  VECTOR (LOCATION 140) TO THE STARTING ADDRESS OF
   FALCON'S ODT ROM (170000-OCTAL).
 - CHANGE PRIORITY LEVEL 7 TO LEVEL 6 TO ALLOW THE BREAK KEY TO INTERRUPT.
```

TABLE OF CONTENTS

```
111
112
113
114
115
116                             1.0 INTRODUCTION
117
118                             1.1 PROGRAM ABSTRACT
119
120
121                             THIS  DIAGNOSTIC  WAS DESIGNED TO TEST OUT THE KMV11 MODULE
122
123                             THE PROGRAM WAS IMPLEMENTED USING THE DIAGNOSTIC SUPERVISOR.
124
125                             THROUGH DIALOGUE WITH THE OPERATOR, THE PROGRAM  WILL  ALLOW
126                             MODIFICATION  OF  DEVICE PARAMETERS, SUCH AS UNIBUS ADDRESS,
127                             VECTOR ADDRESS, AND PRIORITY LEVEL.
128
129
130                             1.2 HARDWARE INTRODUCTION
131                             HARDWARE DESCRIPTION:
132                             M7500    = KMV11-A MODULE
133                             M7501    = KMV11-B MODULE
134
135
136                             KMV11-A IS A SINGLE LINE COMMUNICATION CONTROLLER FOR QBUS SYSTEMS
137                             KMV11-B IS A DUAL LINE COMMUNICATION CONTROLLER FOR QBUS SYSTEMS
138
139
140                             DIAGNOSTIC DESCRIPTION:
141                             THE KMV11 STATIC DIAGNOSTIC IS COMPATIBLE WITH BOTH KMV11 A/B
142                             IT WILL RUN IN STAND ALONE WITHOUT ANY OPERATOR INTERVENTIONS
143
144                             THE PURPOSE OF THIS DIAGNOSTIC IS TO TEST ALL THE HARDWARE OF
145                             THE QBUS PART OF THE INTERFACE AND THE RAM PART OF THE KMV11.
146
147                             THIS DIAGNOSTIC WILL FIRST TEST QBUS ACCESS ON KMV11A(M7500) AND
148                             KMV11B(M7501) CSR'S REGISTERS, THEN DATA TRANSFER FROM QBUS
149                             TO DCT11 MICROPROCESSOR.
150                             AFTER THAT IT WILL TEST KMV11 RAM MEMORY, DMA TRANSFERS IN/OUT
151                             KMV11 AND INTERRUPT CAPABILITY.
152
153
154
155
156
157                             2.0 HARDWARE REQUIREMENTS
158
159                             THE  FOLLOWING  HARDWARE  IS REQUIRED TO RUN THE STATIC LOGIC
160                             TESTS ON MODULES M7500 OR M7501:
161
162                             SBC 11/21 +
163                             16K MEMORY
164                             CONSOLE TERMINAL
165                             REAL TIME CLOCK
166
167                             3.0 PRELIMINARY PROGRAM REQUIREMENTS
```

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

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

4.0 GENERAL PROGRAM CONSIDERATIONS

4.1 DIAGNOSTIC SUPERVISOR

THIS PROGRAM IS COMPATIBLE WITH THE STANDALONE DIAGNOSTIC SUPERVISOR, AND MUST BE LOADED TO BE CO-RESIDENT WITH THE SUPERVISOR, OR BE PREVIOUSLY COMBINED WITH THE SUPERVISOR AND LOADED AS A SINGLE FILE. IN EITHER CASE, THE COMBINED PROGRAM WILL NOT EXCEED 16K OF MEMORY.

4.2 EXECUTION TIME

THE TOTAL TIME REQUIRED TO RUN THE M7500 OR M7501 STATIC DIAGNOSTIC IS ABOUT 210 SECONDS PER PASS FOR EACH UNIT.

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 PARAMETERS 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
                        NKMDAO-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 OCCURENCE OF AN ERROR WITH THE HALT ON ERROR FLAG BEING SET. THE EXIT IS A RETURN TO COMMAND MODE. SEE EXAMPLE AT END OF 6.3.1.5.

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

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

HOE	HALT ON ERROR, CAUSING COMMAND MODE TO BE ENTERED WHEN AN ERROR IS ENCOUNTERED
LOE	LOOP ON ERROR, CAUSING THE DIAGNOSTIC TO LOOP

CONTINUOUSLY WITHIN THE SMALLEST DEFINED BLOCK
OF CODING (SEGMENT, SUBTEST, OR TEST) CONTAIN-
ING THE ERROR

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

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

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

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

6.3.1.5 EFFECT OF START COMMAND

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

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

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

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

EXAMPLE:

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

THIS COMMAND WILL CAUSE THREE PASSES TO BE MADE, EACH PASS CONSISTING OF TESTS 1,2,3,4,6,8,9, AND 10 EXECUTED AGAINST ALL UNITS. THERE IS NO DIFFERENCE BETWEEN SAYING <FLAG> AND SAYING <FLAG=1>. THE NOTATION <FLAG=0> IS MEANINGFUL ONLY ON A COMMAND OTHER THAN START TO CLEAR A FLAG THAT WAS PREVIOUSLY SET. NOTE THAT ON ALL COMMANDS ONLY THE FIRST THREE LETTERS ARE SCANNED.

6.3.2 RESTART COMMAND

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

6.3.2.1 TESTS, PASS, AND FLAGS SWITCHES

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

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

<UNIT-LIST> IS A SEQUENCE OF DECIMAL NUMBERS (0,1 ETC.) OR RANGES OF DECIMAL NUMBERS (0-5, 8-10 ETC.) THAT SPECIFY THE UNITS TO BE TESTED. THE NUMBERS ARE SEPARATED BY COLONS. THE NUMBERS MAY RANGE FROM 0 THRU N-1 (N IS THE NUMBER OF UNITS SPECIFIED IN THE PREVIOUS START COMMAND). THE NUMBER INDICATES THE POSITION OF THE P-TABLE AS THE DATA WAS ENTERED DURING THE HARDWARE 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>
                        ************************************************************
```

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

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

6.3.4.2 EFFECT OF PROCEED COMMAND

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

6.3.5 ADD COMMAND

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

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

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

6.3.5.2 EFFECT OF ADD COMMAND

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

6.3.6 DROP COMMAND

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

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

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

6.3.6.2 EFFECT OF DROP COMMAND

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

6.3.7 PRINT COMMAND

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

6.3.7.1 EFFECT OF PRINT COMMAND

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

6.3.8 DISPLAY COMMAND

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

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

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

6.3.8.2 EFFECT OF DISPLAY COMMAND

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

6.3.9 FLAGS COMMAND

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

6.3.9.1 EFFECT OF FLAGS COMMAND

THE CURRENT SETTINGS OF ALL FLAGS ARE PRINTED.

6.3.10 ZFLAGS COMMAND

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

6.3.10.1 EFFECT OF ZFLAGS COMMAND

ALL FLAGS ARE CLEARED.

6.3.11 CONTROL CHARACTERS

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

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

A CONTROL O (O) ENTERED DURING THE EXECUTION OF A DIAGNOSTIC CAUSES ALL TELETYPE OUTPUT TO BE 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 OPERATOR.

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

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

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

THE FOLLOWING DIALOGUE WOULD ACCOMPLISH THIS GOAL:

```
@ UNITS (D) ? 16

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

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

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

THE SECOND TIME THRU THE SERIES, TABLES 16 THRU THE END ARE GOING TO BE AFFECTED (NOTE THAT THIS PIECE OF INFORMATION IS PRINTED OUT FOR THE THE OPERATOR IN THE FORM "UNIT XX" AT THE BEGINNING OF EACH SERIES). QUESTION 1 IS RESPONDED TO BY A <CR>, SO SLOT ONE STAYS AT CONSTANT 75 IN TABLES 7 THRU 15, SINCE NO NEW 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
                        ;*
                        ;***************************************************
```

```
960
961
962                         8.0 ERROR INFORMATION
963
964
965                         8.1 ERROR REPORTING
966
967                         ERRORS ARE REPORTED BY THE PROGRAM AS  THEY  OCCUR  (IF  NOT
968                         INHIBITED). THE REPORT CONFORMS TO THE DIAGNOSTIC SUPERVISOR
969                         ERROR REPORT FORMAT, AND CONSISTS OF A  DESCRIPTION  OF  THE
970                         ERROR,  THE  TEST  NUMBER,  SUBTEST  NUMBER, PC OF THE ERROR
971                         CALL,  DEVICE  ADDRESS,  AND  BASIC   AND   EXTENDED   ERROR
972                         INFORMATION.
973
974
975
976                         THE FOLLOWING EXAMPLES PROVIDE TYPICAL ERROR REPORTS:
977
978
979                         ;CZDMQ DVC FTL ERR 00045 TST 027 SUB 000 PC:022572
980
981                         ;MASTER  CLEAR  FAILED  TO  CLEAR  PC  REG.  CONTENTS=000624
982                         ;CZDMQ DVC FTL ERR 00015 TST 042 SUB 000 PC:027234
983
984                         ;UNIT=00, FAILING UNIT ADDRESS=160170
985                         ;JUMP TEST ERROR
986                         ;FROM ADDR      TO ADDR       BAD ADDR
987                         ;000402         000000        000114
988
989
990
991                         FOR ALL OTHER ERRORS, THE REPORT MAY BE MORE  EXTENSIVE  AND
992                         REQUIRE ADDITIONAL DATA TO BE REPORTED.
993
994
995
996                         9.0     HISTORY
997
998                             -   DESIGN STARTED ON JANUARY 82
999                             -   REVIEW ON DECEMBER 82
1000
1001                        @
1002
1003
1004
1005
1006
```

```
                                .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  NKMDA0,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



                        ;*****************************************************************************



                        .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
                                ;
                                ;                                       ;  BIT POSITION IN SECOND STATUS WORD
000040                          EF.START=:      32.                     ; (100000) START COMMAND WAS ISSUED
000037                          EF.RESTART=:    31.                     ; (040000) RESTART COMMAND WAS ISSUED
000036                          EF.CONTINUE=:   30.                     ; (020000) CONTINUE COMMAND WAS ISSUED
000035                          EF.NEW=:        29.                     ; (010000) A NEW PASS HAS BEEN STARTED
000034                          EF.PWR=:        28.                     ; (004000) A POWER-FAIL/POWER-UP OCCURRED
```

```
                        ;
                        ;
                        ; PRIORITY LEVEL DEFINITIONS
                        ;
000340                  PRI07== 340
000300                  PRI06== 300
000240                  PRI05== 240
000200                  PRI04== 200
000140                  PRI03== 140
000100                  PRI02== 100
000040                  PRI01== 40
000000                  PRI00== 0
                        ;
                        ;OPERATOR FLAG BITS
                        ;
000004                  EVL==     4
000010                  LOT==    10
000020                  ADR==    20
000040                  IDU==    40
000100                  ISR==   100
000200                  UAM==   200
000400                  BOE==   400
001000                  PNT==  1000
002000                  PRI==  2000
004000                  IXE==  4000
010000                  IBE== 10000
020000                  IER== 20000
040000                  LOE== 40000
100000                  HOE== 100000
1230
1231
1232                    ;MAXPRI==340                                        ;JB REV A-0
1233    000300          MAXPRI==300                                         ;JB REV A-0
1234
1235    054000          MAINT0==54000       ;MASTER CLEAR = 1,MODE = 1 ,MAINT 1 = 1 ,T11=HOLD
1236    044000          MAINT1==44000       ;MASTER CLEAR = 1,MODE = 0 ,MAINT 1 = 0 ,T11=NOT HOLD
1237    040000          MCLR==40000
1238    052525          DATA1== 052525
1239    125252          DATA2== 125252
1240
1241
1242                    ;;*****************************************************************************
1243                    ;* PROGRAM EVENT FLAG DEFINITIONS
1244                    ;;*****************************************************************************
1245
1246
1247
```

```
                                .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>



                                        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
```

```
1305                                    ;*************************************************************************
1306                                    ;* MISCELLANEOUS STORAGE
1307                                    ;*************************************************************************
1308 002320  000000                     FLAG:    .WORD   0
1309 002322  000000                     DH1:     .WORD   0
1310 002324  000000                     DELCT1:  .WORD   0
1311 002326  000000                     DELCT2:  .WORD   0
1312 002330  000000                     GOOD:    .WORD   0
1313 002332  000000                     GOOD0:   .WORD   0
1314 002334  000000                     GOOD1:   .WORD   0
1315 002336  000000                     GOOD2:   .WORD   0
1316 002340  000000                     GOOD4:   .WORD   0
1317 002342  000000                     GOOD6:   .WORD   0
1318 002344  000000                     GOOD10:  .WORD   0
1319 002346  000000                     GOOD12:  .WORD   0
1320 002350  000000                     GOOD14:  .WORD   0
1321 002352  000000                     GOOD16:  .WORD   0
1322 002354  000000                     SEL0:    .WORD   0
1323 002356  000000                     SEL1:    .WORD   0
1324 002360  000000                     SEL2:    .WORD   0
1325 002362  000000                     SEL4:    .WORD   0
1326 002364  000000                     SEL6:    .WORD   0
1327 002366  000000                     SEL10:   .WORD   0
1328 002370  000000                     SEL12:   .WORD   0
1329 002372  000000                     SEL14:   .WORD   0
1330 002374  000000                     SEL16:   .WORD   0
1331 002376  000000                     BSEL1:   .WORD   0
1332 002400  000000                     RANST:   .WORD   0
1333 002402  000000                     RANSEL:  .WORD   0
1334 002404  000000                     RANMTA:  .WORD   0
1335 002406  000000                     RANDN:   .WORD   0
1336 002410  000000                     SAVPC1:  .WORD   0
1337 002412  000000                     SAVSTA:  .WORD   0
1338 002414  000000                     COUNT:   .WORD   0
1339 002416  000000                     NUMBER:  .WORD   0
1340 002420  000000                     ADDR:    .WORD   0
1341 002422  000000                     GDDAT:   .WORD   0
1342 002424  000000                     BDDAT:   .WORD   0
1343
1344 002426                             TTABLE:  .BLKW   2000
1345 006426                             RTABLE:  .BLKW   2000
1346
1347 012426  000000                     EXADDR:  .WORD   0
1348 012430  000000                     INTFLG:  .WORD   0
1349 012432  000000                     BAD:     .WORD   0
1350 012434  000000                     BSEL0:   .WORD   0
1351 012436  000000                     DATA:    .WORD   0
1352 012440  000000                     VECT:    .WORD   0
1353
1354 012442  000000                     KIND:    .WORD   0              ;=0 IF KMV11A ,=1 IF KMV11B
1355 012444  000000                     CHANEL:  .WORD   0
1356
1357 012446  000000                     TXDATA:  .WORD   0
1358 012450  000000                     RXDATA:  .WORD   0
1359 012452  000000                     TSPEED:  .WORD   0
1360 012454  000000                     LENGTH:  .WORD   0
1361 012456  000000                     NUB:     .WORD   0
```

GLOBAL DATA SECTION

```
1362 012460  000000          RXCNT:  .WORD   0
1363 012462  000000          MAXCNT: .WORD   0
1364
```

```
1366
1367                            ;*****************************************************************
1368                            ;LOAD IN LOCATION "GDREV" THE PROM VERSION NUMBER THAT IS       *
1369                            ;COMPATIBLE WHITH THIS DIAGNOSTIC                              *
1370                            ;                                                              *
1371                            ;EACH PROM CONTAIN A REV LEVEL AND A ECO LEVEL:                *
1372                            ;THE REV LEVEL IS MODIFIED EACH TIME A MODIFICATION IS DONE    *
1373                            ;THE ECO LEVEL IS MODIFIED WHEN THE PROM MODIFICATION NEED     *
1374                            ;A DIAGNOSTIC MODIFICATION                                     *
1375                            ;*****************************************************************
1376
1377
1378
1379
1380
1381
1382 012464  000001             GDREV:  .WORD   1
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
```

```
1394                            ;*************************************************************************
1395                            ;* PROGRAM CONTROL FLAGS
1396                            ;*************************************************************************
1397 012466     000            INIFLG: .BYTE   0               ;PROGRAM INITIALIZING FLAG
1398                                    .EVEN
1399 012470     000            LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
1400 012471     000            QV.FLG: .BYTE   0               ;QUICK VERIFY FLAG
1401                                    .EVEN
1402 012472  000000            UUT:    .WORD   0               ;CURRENT UNIT UNDER TEST
1403
1404
1405                            ;*************************************************************************
1406                            ;* POINTERS TO KMV11 VECTORS AND REGISTERS
1407                            ;*************************************************************************
1408 012474  000000            KMVV00: 0       ;POINTER TO KMV11  INTRPT VECTOR 0
1409 012476  000000            KMVLVL: 0       ;POINTER TO KMV11  INTRPT SERVICE
1410 012500  000000            KMVV04: 0       ;POINTER TO KMV11  INTRPT VECTOR 04
1411 012502  000000            KMVV02: 0       ;    "       "       "       "     02
1412 012504  000000            KMVV06: 0       ;    "       "       "       :     06
1413 012506  000000            KMTLVL: 0       ;POINTER TO KMV11 TX INTRPT SERVICE PS
1414 012510  000000            KMVCSR: 0       ;POINTER TO KMV11 CONTROL STATUS REGISTER
1415 012512  000000            KMVP02: 0       ;POINTER TO KMV11 PORT REGISTER - SEL2
1416 012514  000000            KMVP04: 0       ;POINTER TO KMV11 PORT REGISTER - SEL4
1417 012516  000000            KMVP06: 0       ;POINTER TO KMV11 PORT REGISTER - SEL6
1418
1419 012520  000000            KMVP10: 0       ;POINTER TO KMV11 PORT REG  -SEL10
1420 012522  000000            KMVP12: 0       ;POINTER TO PORT REG  -SEL 14
1421 012524  000000            KMVP14: 0       ;POINTER TO PORT REG  -SEL14
1422 012526  000000            KMVP16: 0       ;POINTER TO PORT REG 16
1423
1424 012530  000000            LOOP:   0       ;POINTER TO LOOP BACK CONNECTOR
1425
1426
1427                            ;;****  PRIMARY REG ADRS STORAGE FOR THIS UNIT  ******
1428                            ;THESE LOCATIONS WILL BE LOADED FOR THE CURRENT UNIT, IN INIT CODE
1429 012532                    REGADR:
1430
1431                            ;;****  STACK USED FOR SUBROUTINE LINKAGE  *****
1432 012532                            .BLKW   100
1433 012732                    SSTACK:
1434
1435
1436
1437
1438
1439
1440
```

```
                        .SBTTL  GLOBAL TEXT SECTION

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

                        ;;*************************************************************************
                        ;* NAMES OF DEVICES SUPPORTED BY PROGRAM
                        ;;*************************************************************************
012732                          DEVTYP  <M7500 OR M7501 MODULE>


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

```
1472                            .SBTTL  GLOBAL SUBROUTINES
1473
1474
1475
1476
1477
1478                            ;-----------------------------------------------------------------
1479                            ; MACRO'S NEEDED TO CALL SUBROUTINES
1480                            ;-----------------------------------------------------------------
1481
1482                            .MACRO  CLRMAR
1483                                    ROMCLK
1484                                    004000
1485                            .ENDM   CLRMAR
```

```
1487                                    ;ROUTINE TO WAIT FOR EVENT OR TIMEOUT
1488
1489
1490
1491                                    ;CALLING SEQUENCE:      JSR   PC,WAIT1
1492                                    ;                       JSR   PC,WAIT2
1493
1494
1495                                    ;INPUTS PARAMETERS:     DELCT1,DELCT2
1496
1497
1498                                    ;                       INC DELCT1 UNTIL 0
1499                                    ;                       DEC DELCT2 UNTIL 0      DELCT2= NUMB OF WAIT1 PASSES
1500
1501
1502
1503
1504
1505
1506
1507 012760  005237  002324          WAIT2:  INC     DELCT1
1508 012764  001375                          BNE     WAIT2
1509
1510 012766                                  BREAK
1511
1512 012770  005337  002326                  DEC     DELCT2
1513 012774  001371                          BNE     WAIT2
1514
1515 012776  000207                          RTS     PC
1516
1517
1518
1519
1520
1521
1522 013000  005237  002324          WAIT1:  INC     DELCT1
1523 013004  001375                          BNE     WAIT1
1524
1525 013006  000207                          RTS     PC
```

```
1527                            ;MACRO TO WAIT A FEW MS
1528
1529
1530                            ;CALLING SEQUENCE:      WAITA   X               0<X<177777
1531                            ;                       WAITB   X,Y             0<X OR Y<177777
1532
1533
1534
1535                            .MACRO  WAITA   X
1536                                    MOV     #X,DELCT1               ;LOAD COUNT
1537                                    JSR     PC,WAIT1                ;WAIT
1538                            .ENDM
1539
1540
1541
1542
1543
1544
1545
1546                            .MACRO  WAITB   X,Y
1547                                    MOV     #X,DELCT1
1548                                    MOV     #Y,DELCT2
1549                                    JSR     PC,WAIT2
1550                            .ENDM
1551
1552
1553
1554
```

```
                                ;ROUTINE TO DROP UNIT AFTER 5 ERROR

                                ;JSR   PC,CHKMAX

013010                          CHKMAX: INLOOP                          ;LOOPING ON ERROR?
013012                                  BCOMPLETE       1$              ;IF YES, EXIT

013014                                  RFLAGS  R0                      ;GET OPERATOR FLAG
013016  032700  000040                  BIT     #IDU,R0                 ;IS DROPPING INHIBITTED?
013022  001026                          BNE     1$                      ;IF YES EXIT

013024  005237  002300                  INC     ERRCNT                  ;UPDATE ERROR COUNT
013030  023737  002300  002276          CMP     ERRCNT,MAXERR           ;TOO MANY ERROR?
013036  003420                          BLE     1$                      ;IF NOT JUMP

013040                                  PRINTF  #NERRS,MAXERR,UUT       ;TOO MANY ERROR!
013070                                  DODU    UUT                     ;DROP UNIT

013076                                  DOCLN                           ;END THE SUBPASS

013100  000207                  1$:     RTS     PC

                                        .NLIST  BEX
013102      045     116     045 NERRS:  .ASCIZ  /%N%AMORE THAN %D3%A  ERRORS ON UNIT %D2/
                                        .LIST   BEX
                                        .EVEN
```

```
                                ;ROUTINE TO CHECK REGISTER BSELO 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,CBSELO               ;LOOK IF BSELO=0
013156  000000                          .WORD   0
013160  000411                          BR      1$                      ;TEST IS OK ,RTS PC
013162  122737  000200  012434          CMPB    #200,BSELO              ;LOOK IF BSELO=200
013170  001406                          BEQ     2$                      ;TIMEOUT DURING TEST,RTS PC+2
013172  122737  000100  012434          CMPB    #100,BSELO              ;LOOK IF BSELO=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
```

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

```
1710                                    ;
1711                                    ;
1712 013230  042703  177770      GENER:  BIC     #177770,R3
1713 013234  004737  013530              JSR     PC,SAVREG
1714 013240  006303                      ASL     R3
1715 013242  000173  013246              JMP     @GENSEL(R3)
1716 013246  013266              GENSEL: GEN0                    ;ALL ZERO WORD
1717 013250  013272                      GEN1                    ;ALL ONE WORD
1718 013252  013300                      GEN52                   ;52 PATTERN
1719 013254  013306                      GEN25                   ;25 PATTERN
1720 013256  013314                      GENR1                   ;ROTATE '1' EACH CALL
1721 013260  013324                      GENR0                   ;ROTATE '0' EACH CALL
1722 013262  013362                      GENRAN                  ;RANDOM NUMBER
1723 013264  013502                      GENINC                  ;INCREMENTING COUNT
1724 013266  005000              GEN0:   CLR     R0              ;0>R0
1725 013270  000507                      BR      GENEX
1726 013272  005000              GEN1:   CLR     R0              ;NOT0>R0
1727 013274  005100                      COM     R0
1728 013276  000504                      BR      GENEX
1729 013300  012700  052525      GEN52:  MOV     #52525,R0       ;5252>R0
1730 013304  000501                      BR      GENEX
1731 013306  012700  125252      GEN25:  MOV     #125252,R0      ;125252>R0
1732 013312  000476                      BR      GENEX
1733 013314  000241              GENR1:  CLC
1734 013316  004737  013336              JSR     PC,GENROT       ;SHIFT 1 > R0
1735 013322  000472                      BR      GENEX
1736 013324  000241              GENR0:  CLC
1737 013326  004737  013336              JSR     PC,GENROT       ;
1738 013332  005100                      COM     R0              ;SHIFT 0 > R0
1739 013334  000465                      BR      GENEX
1740 013336  006037  013360      GENROT: ROR     GENISH          ;ROTATE  1 PATTERN
1741 013342  001003                      BNE     GENER1          ;= 0?
1742 013344  012737  100000  013360      MOV     #100000,GENISH  ;YES, SET MSB
1743 013352  013700  013360      GENER1: MOV     GENISH,R0       ;PUT 1 IN R0
1744 013356  000207                      RTS     PC              ;AND EXIT
1745 013360  000001              GENISH: 1
1746 013362  012737  000005  002402 GENRAN: MOV     #5,RAN.L        ;SET SELECT VALUE TO 5
1747 013370  004737  013402              JSR     PC,RANGEN       ;GENERATE RANDOM NUMBER IN R0
1748 013374  013700  002406              MOV     RANDN,R0        ;
1749 013400  000443                      BR      GENEX           ;
1750 013402  013702  002406      RANGEN: MOV     RANDN,R2        ;
1751 013406  001002                      BNE     RAN1            ;IS RANDOM = 0
1752 013410  013702  002400              MOV     RANST,R2        ;YES, PUT RANDOM START VALUE IN
1753 013414  032737  000777  002402 RAN1:   BIT     #777,RANSEL     ;NO;IS RANSEL SELECT VALUE = 0
1754 013422  001003                      BNE     RAN2            ;NO
1755 013424  012737  000001  002402      MOV     #1,RANSEL       ;YES: SET RANSEL = 1
1756 013432  013703  002402      RAN2:   MOV     RANSEL,R3       ;
1757 013436  013702  002406              MOV     RANDN,R2
1758 013442  033702  002404              BIT     RANMTA,R2       ;GET R2 <0 AND 1>
1759 013446  001405                      BEQ     RANCLC          ;
1760 013450  005102                      COM     R2
1761 013452  033702  002404              BIT     RANMTA,R2
1762 013456  001401                      BEQ     RANCLC
1763 013460  000402                      BR      RANSEC
1764 013462  000241              RANCLC: CLC
1765 013464  000401                      BR      RAN4
1766 013466  000261              RANSEC: SEC
```

```
1767 013470  006037  002406          RAN4:   ROR     RANDN           ;ROTATE C TO B15
1768 013474  005303                  DEC     R3                      ;IS THIS NUMBER REQUIRED?
1769 013476  001357                  BNE     RAN2+4                  ;NO, GET ANOTHER
1770 013500  000207          RANEX:  RTS     PC                      ;YES, EXIT
1771 013502  013700  002330  GENINC: MOV     GOOD,R0                 ;INCREMENTS LOC. 'GOOD'
1772 013506  005200                  INC     R0
1773 013510  010037  002330  GENEX:  MOV     R0,GOOD
1774 013514  004737  013610          JSR     PC,RSTREG
1775 013520  013737  002330  012436  MOV     GOOD,DATA
1776 013526  000207                  RTS     PC
1777
```

SAVE REGISTERS

```
                        .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

```
1836 013550  012637  002410          MOV     (SP)+,SAVPC1
1837 013554  010546                  MOV     R5, -(SP)
1838 013556  010446                  MOV     R4, -(SP)
1839 013560  010346                  MOV     R3, -(SP)
1840 013562  010246                  MOV     R2, -(SP)
1841 013564  010146                  MOV     R1, -(SP)
1842 013566  010046                  MOV     R0, -(SP)
1843 013570  013746  002410          MOV     SAVPC1, -(SP)
1844 013574  013746  002310          MOV     SAVPC, -(SP)    ;PUT PC READY FOR
1845 013600                          SETPRI  SAVSTA
1846 013606  000207                  RTS     PC              ;RETURN
1847
1848
1849
```

RESTORE REGISTERS

```
                        .SBTTL  RESTORE REGISTERS
                ;
                ;
                ;       DESCRIPTION:
                ;
                ;               RESTORE TO RESTORE THE GENERAL PURPOSE
                ;       REGISTERS. THE STACK IS LEFT IN THE SAME STATE AS IT
                ;       WAS WHEN SAVREG WAS CALLED.
                ;
                ;       CAUTION:  REGISTER R0 IS NOT SAVED
                ;
                ;
                ;
                ;       CALLING SEQUENCE:
                ;
                ;               JSR     PC,RSTREG
                ;
                ;       INPUT PARAMETERS:
                ;
                ;               NONE
                ;
                ;       IMPLICIT INPUT PARAMETERS:
                ;
                ;               NONE
                ;
                ;       OUTPUT PARAMETERS:
                ;
                ;               R1 THRU R5 RESTORED
                ;
                ;       IMPLICIT OUTPUT PARAMETERS:
                ;
                ;               NONE
                ;
                ;       COMPLETION CODES:
                ;
                ;               NONE
                ;
                ;       POSSIBLE ERROR CODES:
                ;
                ;               NONE
                ;
                ;
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
```

RESTORE REGISTERS

```
1908 013640  012602                    MOV     (SP)+,R2
1909 013642  012603                    MOV     (SP)+,R3
1910 013644  012604                    MOV     (SP)+,R4
1911 013646  012605                    MOV     (SP)+,R5
1912 013650  013746  002410            MOV     SAVPC1,-(SP)
1913 013654  013746  002310            MOV     SAVPC,-(SP)     ;PUT PC READY FOR
1914 013660                            SETPRI  SAVSTA
1915 013666  000207                    RTS     PC
```

RESTORE REGISTERS

```
                                ;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
```

```
RESTORE REGISTERS

 2011 014142  000240                        NOP
 2012 014144  017737  176356  002374       MOV     @KMVP16,SEL16          ;READ SEL16
 2013
 2014
 2015 014152  023737  002354  002332       CMP     SEL0,GOOD0
 2016 014160  001035                        BNE     1$
 2017 014162  023737  002360  002336       CMP     SEL2,GOOD2
 2018 014170  001031                        BNE     1$
 2019 014172  023737  002362  002340       CMP     SEL4,GOOD4
 2020 014200  001025                        BNE     1$
 2021 014202  023737  002364  002342       CMP     SEL6,GOOD6
 2022 014210  001021                        BNE     1$
 2023 014212  023737  002366  002344       CMP     SEL10,GOOD10
 2024 014220  001015                        BNE     1$
 2025 014222  023737  002370  002346       CMP     SEL12,GOOD12
 2026 014230  001011                        BNE     1$
 2027 014232  023737  002372  002350       CMP     SEL14,GOOD14
 2028 014240  001005                        BNE     1$
 2029 014242  023737  002374  002352       CMP     SEL16,GOOD16
 2030 014250  001001                        BNE     1$
 2031
 2032 014252  000402                        BR      2$
 2033 014254  062705  000002          1$:   ADD     #2,R5
 2034 014260  000205                2$:   RTS     R5
```

```
                                        ;ROUTINE TO CHECK SEL2 TO SEL16


014262  012537  002336          CKREG:  MOV     (R5)+,GOOD2
014266  012537  002340                  MOV     (R5)+,GOOD4
014272  012537  002342                  MOV     (R5)+,GOOD6
014276  012537  002344                  MOV     (R5)+,GOOD10
014302  012537  002346                  MOV     (R5)+,GOOD12
014306  012537  002350                  MOV     (R5)+,GOOD14
014312  012537  002352                  MOV     (R5)+,GOOD16

014316  017737  176170  002360          MOV     @KMVP02,SEL2
014324  000240                          NOP
014326  000240                          NOP
014330  000240                          NOP
014332  000240                          NOP
014334  017737  176154  002362          MOV     @KMVP04,SEL4
014342  000240                          NOP
014344  000240                          NOP
014346  000240                          NOP
014350  000240                          NOP
014372  017737  176140  002364          MOV     @KMVP06,SEL6
014430  000240                          NOP
014432  000240                          NOP
014434  000240                          NOP
014436  000240                          NOP
014440  017737  176124  002366          MOV     @KMVP10,SEL10
014376  000240                          NOP
014400  000240                          NOP
014402  000240                          NOP
014404  000240                          NOP
014406  017737  176110  002370          MOV     @KMVP12,SEL12
014414  000240                          NOP
014416  000240                          NOP
014420  000240                          NOP
014422  000240                          NOP
014424  017737  176074  002372          MOV     @KMVP14,SEL14
014432  000240                          NOP
014434  000240                          NOP
014436  000240                          NOP
014440  000240                          NOP
014442  017737  176060  002374          MOV     @KMVP16,SEL16


014450  023737  002360  002336          CMP     SEL2,GOOD2
014456  001031                          BNE     1$
014460  023737  002362  002340          CMP     SEL4,GOOD4
014466  001025                          BNE     1$
014470  023737  002364  002342          CMP     SEL6,GOOD6
014476  001021                          BNE     1$
014500  023737  002366  002344          CMP     SEL10,GOOD10
```

RESTORE REGISTERS

```
 2093 014506  001015                          BNE     1$
 2094 014510  023737  002370  002346          CMP     SEL12,GOOD12
 2095 014516  001011                          BNE     1$
 2096 014520  023737  002372  002350          CMP     SEL14,GOOD14
 2097 014526  001005                          BNE     1$
 2098 014530  023737  002374  002352          CMP     SEL16,GOOD16
 2099 014536  001001                          BNE     1$
 2100 014540  000402                          BR      2$
 2101
 2102 014542  062705  000002          1$:     ADD     #2,R5
 2103 014546  000205                  2$:     RTS     R5
```

RESTORE REGISTERS

```
2105                                    ;ROUTINE TO CLEAR KMV11 MODULE
2106
2107
2108                                    ;CALLING SEQUENCE:
2109                                    ;       JSR PC,CLRKMV
2110
2111                                    ;ROUTINE DESCRIPTION: CLEAR ALL CSR'S REGISTERS AND CHECK IF = 0
2112
2113
2114
2115 014550  005077  175734  CLRKMV: CLR     @KMVCSR
2116 014554  012777  054000  175726          MOV     #MAINTO,@KMVCSR         ;SET MAINTENANCE MODE
2117 014562                             WAITA   0
2118
2119 014574  012702  000010          MOV     #10,R2
2120 014600  013701  012510          MOV     KMVCSR,R1               ;LOAD ADDRESS
2121 014604  005021          1$:     CLR     (R1)+                   ;CLEAR
2122 014606  000240          NOP
2123 014610  000240          NOP
2124 014612  000240          NOP
2125 014614  005302          DEC     R2                      ;ALL DONE
2126 014616  001372          BNE     1$                      ;NO
2127 014620  004537  013760          JSR     R5,CKALL                ;CHECK ALL REG = 0
2128 014624  000000                  .WORD   0
2129 014626  000000                  .WORD   0
2130 014630  000000                  .WORD   0
2131 014632  000000                  .WORD   0
2132 014634  000000                  .WORD   0
2133 014636  000000                  .WORD   0
2134 014640  000000                  .WORD   0
2135 014642  000000                  .WORD   0
2136 014644  000404                  BR      2$                      ;OK BRANCH AT END
2137 014646                          ERRHRD 2,EM0002,PRALL           ;CSR'S REGISTERS CAN'T BE CLEARED
2138 014656  000207          2$:     RTS     PC
2139
```

```
                                ;ROUTINE TO SET MAINTENANCE MODE 0 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,CKSELO               ;CHECK SELO=0 BUT MODE BIT + MAINT1 BIT
014712  014000                          .WORD   14000
014714  000404                          BR      1$
014716                                  ERRHRD  3,EM0001,PRSELO
014726  000207                  1$:     RTS     PC
```

RESTORE REGISTERS

```
2174                                    ;ROUTINE TO SET MAINT MODE 1 AND CHECK DCT11 CLEAR SEL0 AFTER HAVING DECODED
2175
2176
2177
2178                                    ;CALLING SEQUENCE:
2179                                    ;       JSR PC,MAINM1
2180
2181
2182
2183                                    ;GIVE AN ERROR IF MASTER CLEAR IS NOT CLEAR BY DCT11
2184                                    ;
2185                                    ;MAINT1= MASTER CLEAR=1 + MAINT 1 =0 + MODE = 1 ; T11=NOT IN HOLD
2186
2187
2188
2189
2190
2191
2192 014730  005077  175554            MAINM1: CLR     @KMVCSR
2193 014734  000240                            NOP
2194 014736  000240                            NOP
2195 014740  012777  044000  175542            MOV     #MAINT1,@KMVCSR         ;LOAD ADDRESS
2196 014746  012737  177700  002324            MOV     #177700,DELCT1
2197 014754  012737  000001  002326            MOV     #1,DELCT2
2198 014762  004737  012760                    JSR     PC,WAIT2
2199 014766  004537  013670                    JSR     R5,CKSELO               ;CHECK SEL0=0 BUT MODE BIT =1
2200 014772  004000                            .WORD   4000
2201 014774  000404                            BR      1$                      ;OK BRANCH
2202 014776                                    ERRHRD  4,EM0001,PRSELO
2203 015006  000207                    1$:     RTS     PC
2204
2205
2206
2207
2208
```

```
                                ;ROUTINE TO SET TEST NUMBER ON BSEL0
                                ;
                                ;
                                ;
                                ;CALLING SEQUENCE:
                                ;       JSR R5,TSTNUB
                                ;       .WORD   A                       A=TEST MICRO PROGRAM NUMBER
                                ;
                                ;
                                ;
                                ;
                                ;
                                TSTNUB: MOV     (R5)+,NUB
015010  012537  012456          TSTNUB: BIS     NUB,@KMVCSR             ;LOAD TEST NUMBER
015014  053777  012456  175466          MOV     #170000,DELCT1
015022  012737  170000  002324          MOV     #1,DELCT2
015030  012737  000001  002326          JSR     PC,WAIT2                ;WAIT
015036  004737  012760                  RTS     R5
015042  000205
```

RESTORE REGISTERS

```
2231
2232
2233                            ;ROUTINE TO WRITE OR READ ONE OF THE KMV11 REGISTERS
2234
2235
2236
2237                            ;CALLING SEQUENCE:
2238                            ;JSR     R5,WRITE
2239                            ;.WORD   A                              A=ADDRESS TO WRITE
2240                            ;.WORD   B                              B=DATA TO WRITE
2241                            ;
2242                            ;
2243                            ;
2244                            ;JSR     R5,READ
2245                            ;.WORD   A                              A=ADDRESS TO READ
2246                            ;
2247                            ;
2248                            ;
2249                            ;MICRO DIAG NB 47 DESCRIPTION:
2250                            ;WRITE: PUT ADDRESS TO WRITE IN SEL2
2251                            ;       PUT DATA TO WRITE IN SEL4
2252                            ;       SET BIT 0 OF SEL6(WRITE BIT)
2253                            ;       SET TEST NB 44
2254                            ;       KMV11 CLEAR BSEL0 WHEN DONE
2255                            ;
2256                            ;
2257                            ;READ:  PUT ADDRESS TO READ IN SEL2
2258                            ;       CLEAR BIT 0 IN SEL6
2259                            ;       SET TEST 47
2260                            ;       KMV11 READ ADDRESS IN SEL2 AND CLEAR BSEL0 WHEN DONE
2261                            ;       THE READ DATA IS LOAD IN LOCATION "BAD" AND "DATA"
2262
2263
2264
2265
2266 015044  012577  175442    WRITE:  MOV     (R5)+,@KMVP02           ;WRITE ADDRESS
2267 015050  012577  175440            MOV     (R5)+,@KMVP04           ;  "   DATA
2268 015054  012777  000001  175434    MOV     #1,@KMVP06              ;BIT WRITE
2269
2270 015062  004537  015010            JSR     R5,TSTNUB               ;SEND TEST NB 44
2271 015066  000047                    .WORD   47
2272
2273 015070  000205                    RTS     R5                      ;RETURN
2274
2275
2276
2277
2278
2279
2280 015072  012577  175414    READ:   MOV     (R5)+,@KMVP02           ;SET ADDRESS TO READ
2281 015076  005077  175412            CLR     @KMVP04
2282 015102  005077  175410            CLR     @KMVP06
2283
2284 015106  004537  015010            JSR     R5,TSTNUB               ;SEND TEST NB 44
2285 015112  000047                    .WORD   47
2286
2287 015114  000240                    NOP
```

RESTORE REGISTERS

```
2288 015116  000240                          NOP
2289
2290
2291 015120  004737  013152                  JSR     PC,TSTERR               ;CHECK BSEL 0
2292 015124  000410                          BR      1$                      ;OK
2293 015126  000402                          BR      2$
2294 015130  000401                          BR      2$
2295 015132  000400                          BR      2$
2296
2297 015134                          2$:     ERRHRD  5,EM0024                ;NO KMV ANSWER
2298 015144  000205                          RTS     R5
2299
2300 015146  017737  175342  012432  1$:     MOV     @KMVP04,BAD             ;READ DATA IN BAD
2301 015154  013737  012432  012436          MOV     BAD,DATA                ;READ DATA IN "DATA" LOCATION
2302
2303 015162  000205                          RTS     R5
2304
2305
2306
2307
```

RESTORE REGISTERS

```
2309                                    .MACRO  ROMCLK
2310                                    .LIST
2311                                            JSR     R5,.ROMCLK      ;CLOCK INSTRUCTION
2312                                    .NLIST
2313                                    .ENDM
2314
2315                                    .MACRO  ED$CALL XY
2316                                            .LIST
2317                                    ;********************* TEST'XY' ********************
2318                                            .NLIST
2319                                            .ENDM
2320
2321
2322
2323                                    .MACRO  BADHEAD
2324                                            .RADIX  10
2325                                            ED$CALL \T$TESTNUM+1
2326                                            .RADIX  8
2327                                            .ENDM
2328
2329
2330
```

GLOBAL ERROR REPORT SECTION

```
2332                                    .SBTTL  GLOBAL ERROR REPORT SECTION
2333
2334                                    ;////////////////////////////////////////////////////////////////
2335                                    ;/      THE GLOBAL ERROR REPORT SECTION CONTAINS ERROR MESSAGES
2336                                    ;/      THAT ARE USED IN MORE THAN ONE TEST.
2337                                    ;////////////////////////////////////////////////////////////////
2338
2339                                    .NLIST BEX
2340
2341
2342
2343 015164      045     116     045    TFM36:  .ASCIZ  /%N%AREGISTER ADDRESS ERROR,ADDR = %O6%A,UNIT = %O2/
2344
2345 015247      040     102     125    TIM:    .ASCIZ  / BUS TIMEOUT /
2346
2347 015265      115     101     123    EM0001: .ASCIZ  /MASTER CLEAR FAIL TO RESET: DCT11 CAN'T CLEAR MASTER CLEAR /
2348
2349 015361      040     113     115    EM0002: .ASCIZ  / KMV11 REGISTERS CAN'T BE CLEARED /
2350
2351 015424      040     104     101    EM0003: .ASCIZ  / DATA COMPARE ERROR ON KMV11 REGISTER (SEL2 TO SEL16)/
2352
2353 015512      040     104     101    EM0004: .ASCIZ  / DATA COMPARE ERROR ON BSEL0 WHEN ACCESSED BY QBUS/
2354
2355 015575      040     122     105    EM0005: .ASCIZ  / REGISTER SEL2 CAN'T BE ACCESSED CORRECTLY BY MICRO PROGRAM/
2356
2357 015671      105     122     122    EM0006: .ASCIZ  /ERROR WHEN TESTING SEL4,DCT11 CAN'T  ACCESS SEL4 CORRECTLY/
2358
2359 015764      105     122     122    EM0007: .ASCIZ  /ERROR WHEN TESTING SEL6,DCT11 CAN'T ACCESS SEL6 CORRECTLY/
2360
2361 016056      105     122     122    EM0010: .ASCIZ  /ERROR WHEN TESTING SEL10,DCT11 CAN'T ACCESS SEL10 CORRECTLY/
2362
2363 016152      105     122     122    EM0011: .ASCIZ  /ERROR WHEN TESTING SEL12,DCT11 CAN'T ACCESS SEL12 CORRECTLY/
2364
2365 016246      105     122     122    EM0012: .ASCIZ  /ERROR WHEN TESTING SEL14,DCT11 CAN'T ACCESS SEL14 CORRECTLY/
2366
2367 016342      105     122     122    EM0013: .ASCIZ  /ERROR WHEN TESTING SEL16,DCT11 CAN'T ACCESS SEL16 CORRECTLY/
2368
2369 016436      040     104     101    EM0015: .ASCIZ  / DATA COMPARE ERROR IN RAM MEMORY TEST /
2370
2371 016506      040     124     111    EM0016: .ASCIZ  / TIMEOUT DURING DMA TRANSFER /
2372
2373 016544      040     104     101    EM0020: .ASCIZ  / DATA COMPARE ERROR AFTER DMA  TRANSFER INTO KMV11 /
2374
2375 016630      040     104     101    EM0021: .ASCIZ  / DATA COMPARE ERROR AFTER DMA TRANSFER IN BOTH DIRECTION /
2376
2377 016722      111     116     124    EM0022: .ASCIZ  /INTERUPT  OCCUR AT WRONG LEVEL /
2378
2379 016762      116     117     040    EM0023: .ASCIZ  /NO INTERUPT OCCUR /
2380
2381 017005      116     117     040    EM0024: .ASCIZ  /NO ANSWER FROM KMV11 MODULE ,  MICRO TEST NOT EXECUTED /
2382
2383 017075      124     111     115    EM0025: .ASCIZ  /TIMEOUT DURING KMV11 MICRO TEST /
2384
2385 017136      111     116     124    EM0026: .ASCIZ  /INTERUPT ON DCT11 WITH ILLEGAL VECTOR WHEN ACCESSING BSEL2 /
2386
2387 017232      116     117     040    EM0027: .ASCIZ  /NO KMV11 ANSWER ,DCT11  RECEIVE NO INTERUPT /
2388
```

```
2389 017307     111     114     114  EM0028: .ASCIZ  /ILLEGAL INTERUPT OCCURED /
2390
2391 017341     104     101     124  EM0030: .ASCIZ  /DATA COMPARE ERROR DURING DMA TRANSFER OUT KMV11 /
2392
2393 017423     105     122     122  EM0031: .ASCIZ  /ERROR DURING BYTE ACCES ON KMV11 REGISTERS /
2394
2395 017477     111     116     124  EM0032: .ASCIZ  /INTERUPT ON DCT11 WITH ILLEGAL VECTOR WHEN ACCESSING BSEL0 /
2396
2397 017573     122     101     115  EM0033: .ASCIZ  /RAM MEMORY ERROR WHEN TRANSFERING BUFFER IN DMA /
2398
2399 017654     120     122     117  EM0034: .ASCIZ  /PROM REVISION IS NOT COMPATIBLE WITH DIAGNOSTIC REVISION /
2400
2401 017746     040     103     110  EM0134: .ASCIZ  / CHECK PROM AND DIAGNOSTIC REVISION /
2402
2403 020013     040     040     120  EM0035: .ASCIZ  /  PROM CHECKSUM ERROR /
2404
2405
2406
```

```
2408 020042  045  116  045  MSEL0:  .ASCIZ  /%N%A SEL0 = %O6%A  SHOUD BE  = %O6%N/
2409
2410 020107  045  116  045  MREG0:  .ASCIZ  /%N%A SEL0 = %O6%A  SHOUD BE  = %O6/
2411 020152  045  116  045  MREG2:  .ASCIZ  /%N%A SEL2 = %O6%A  SHOUD BE  = %O6/
2412 020215  045  116  045  MREG4:  .ASCIZ  /%N%A SEL4 = %O6%A  SHOUD BE  = %O6/
2413 020260  045  116  045  MREG6:  .ASCIZ  /%N%A SEL6 = %O6%A  SHOUD BE  = %O6/
2414 020323  045  116  045  MREG10: .ASCIZ  /%N%A SEL10= %O6%A  SHOUD BE  = %O6/
2415 020366  045  116  045  MREG12: .ASCIZ  /%N%A SEL12= %O6%A  SHOUD BE  = %O6/
2416 020431  045  116  045  MREG14: .ASCIZ  /%N%A SEL14= %O6%A  SHOUD BE  = %O6/
2417 020474  045  116  045  MREG16: .ASCIZ  /%N%A SEL16= %O6%A  SHOUD BE  = %O6/
2418
2419
2420 020537  045  116  045  MSEL2:  .ASCIZ  /%N%A SEL2 = %O6%A  SHOUD BE  = %O6/
2421
2422 020602  045  116  045  MSEL4:  .ASCIZ  /%N%A SEL4 = %O6%A  SHOUD BE  = %O6/
2423
2424 020645  045  116  045  MSEL10: .ASCIZ  /%N%A SEL10 = %O6%A SHOUD BE  =%O6/
2425
2426 020707  045  116  045  MRAM1:  .ASCIZ  /%N%A  RAM ADDRESS  =%O6%A,EXTENDED ADDRESS   =%O6/
2427 020771  045  116  045  MRAM2:  .ASCIZ  /%N%A          BDDAT = %O6%A SHOUD CONTENT  = %O6/
2428
2429 021051  045  116  045  MINT:   .ASCIZ  /%N%A GOOD= %O6%A  BAD=%O6/
2430
2431 021103  045  116  045  MBSEL0: .ASCIZ  /%N%A BSEL0 =%O6%A  SHOUD BE  =%O6/
2432
2433 021145  045  116  045  MINTR:  .ASCIZ  /%N%A DCT11 ILLEGAL INTERUPT WHEN ADDRESS =%O6%A IS WRITTEN/
2434
2435 021240  045  116  045  MDMA1:  .ASCIZ  /%N%A   DMA ERROR AT ADDRESS =%O6%A EXTADDRESS =%O6/
2436 021323  045  116  045  MDMA2:  .ASCIZ  /%N%A       BDDAT=%O6%A SHOULD BE =%O6/
2437
2438 021371  045  116  045  MBYTE:  .ASCIZ  /%N%A   AT ADDRESS ADDR =%O6%A, GOOD =%O6%A,BAD =%O6/
2439
2440 021453  045  116  045  MDMAR1: .ASCIZ  /%N%A    RAM MEMORY LOCATION  =%O6%A  IS MODIFIED DURING /
2441 021542  045  116  045  MDMAR2: .ASCIZ  /%N%A    DMA TRANSFER IN BOTH DIRECTION /
2442 021610  045  116  045  MDMAR3: .ASCIZ  /%N%A        READ DATA  =%O6%A  SHOULD BE  =%O6/
2443
2444 021666  045  116  045  MCHECK: .ASCIZ  /%N%A    CHECKSUM IS =%O6%A SHOULD BE ZERO /
2445
2446 021737  045  116  045  MDMAF:  .ASCIZ  /%N%A   ADDR = %O6%A  ,GDDAT = %O6%A  ,BDDAT  = %O6/
2447
2448
2449
```

```
2451
2452                                    .EVEN
2453
2454
2455
2456                                    ;-------------------------------------------------------------------------
2457                                    ; MACRO'S NEEDED TO REPORT ERRORS
2458                                    ;-------------------------------------------------------------------------
2459
2460 022022                                     BGNMSG  PRSELO                          ;REPORT CONTENT OF SELO
2461 022022                                     PRINTB  #MSELO,SELO,GOOD
2462 022052  004737  013010                     JSR     PC,CHKMAX                       ;CHECK IF MAX ERROR?
2463 022056                                     ENDMSG
2464
2465
2466 022060                             BGNMSG  PRRAM                                   ;RAM ERROR REPORT
2467 022060                             PRINTB  #MRAM1,ADDR,EXADDR
2468 022110                             PRINTB  #MRAM2,BDDAT,GOOD
2469 022140  004737  013010                     JSR     PC,CHKMAX                       ;CHECK IF MAX ERROR?
2470 022144                             ENDMSG
2471
2472
2473
2474 022146                             BGNMSG  PRBYTE                                  ;BYTE ACCES REPORT
2475 022146                             PRINTB  #MBYTE,ADDR,GOOD,BAD
2476 022202  004737  013010                     JSR     PC,CHKMAX                       ;CHECK IF MAX ERROR?
2477 022206                             ENDMSG
2478
2479
2480 022210                             BGNMSG  PDMARA                                  ;DMA IN RAM ERROR REPORT
2481 022210                             PRINTB  #MDMAR1,ADDR
2482 022234                             PRINTB  #MDMAR2
2483 022254                             PRINTB  #MDMAR3,BDDAT,GOOD
2484 022304                                     BREAK
2485 022306  004737  013010                     JSR     PC,CHKMAX                       ;CHECK IF MAX ERROR?
2486 022312                             ENDMSG
2487
2488
2489 022314                             BGNMSG  PCHECK                                  ;CHECKSUM ERROR REPORT
2490 022314                             PRINTB  #MCHECK,BAD
2491 022340  004737  013010                     JSR     PC,CHKMAX                       ;CHECK IF MAX ERROR?
2492 022344                             ENDMSG
2493
2494
2495
2496
2497 022346                             BGNMSG  PADFLT                                  ;ADDRESS TEST
2498 022346                             PRINTB  #TFM36,ADDR,UNIT
2499 022376  004737  013010                     JSR     PC,CHKMAX
2500 022402                             ENDMSG
2501
2502
2503
2504
2505
```

```
2507 022404                          BGNMSG  PRALL                          ;CSR'S CONTENT REPORT
2508 022404                          PRINTB  @MREG0,SEL0,GOOD0
2509 022434                          PRINTB  @MREG2,SEL2,GOOD2
2510 022464                          PRINTB  @MREG4,SEL4,GOOD4
2511 022514                          PRINTB  @MREG6,SEL6,GOOD6
2512 022544                          PRINTB  @MREG10,SEL10,GOOD10
2513 022574                          PRINTB  @MREG12,SEL12,GOOD12
2514 022624                          PRINTB  @MREG14,SEL14,GOOD14
2515 022654                          PRINTB  @MREG16,SEL16,GOOD16
2516 022704                                  BREAK
2517 022706  004737  013010                  JSR     PC,CHKMAX              ;CHECK IF MAX ERROR?
2518 022712                          ENDMSG
2519
2520
2521
2522
2523
2524
2525
2526
2527 022714                          BGNMSG  PRREG                          ;CSR'S REPORT BUT SEL0
2528 022714                          PRINTB  @MREG2,SEL2,GOOD2
2529 022744                          PRINTB  @MREG4,SEL4,GOOD4
2530 022774                          PRINTB  @MREG6,SEL6,GOOD6
2531 023024                          PRINTB  @MREG10,SEL10,GOOD10
2532 023054                          PRINTB  @MREG12,SEL12,GOOD12
2533 023104                          PRINTB  @MREG14,SEL14,GOOD14
2534 023134                          PRINTB  @MREG16,SEL16,GOOD16
2535 023164                                  BREAK
2536 023166  004737  013010                  JSR     PC,CHKMAX              ;CHECK IF MAX ERROR?
2537 023172                          ENDMSG
2538
2539
2540
2541
2542
2543
2544
2545 023174                          BGNMSG  PBSEL0                         ;BSEL0 REPORT
2546 023174                          PRINTB  @MBSEL0,BSEL0,GOOD
2547 023224  004737  013010                  JSR     PC,CHKMAX              ;CHECK IF MAX ERROR?
2548 023230                          ENDMSG
2549
2550
2551
2552
2553
2554 023232                          BGNMSG  PINTR                          ;INTERUPT REPORT
2555 023232                          PRINTB  @MINTR,ADDR
2556 023256  004737  013010                  JSR     PC,CHKMAX              ;CHECK IF MAX ERROR?
2557 023262                          ENDMSG
2558
2559
2560
```

GLOBAL ERROR REPORT SECTION

```
2562
2563 023264                          BGNMSG  PRDMA                          ;DMA ERROR REPORT
2564 023264                          PRINTB  #MDMA1,ADDR,EXADDR
2565 023314                          PRINTB  #MDMA2,BDDAT,GOOD
2566 023344  004737  013010                  JSR     PC,CHKMAX              ;CHECK IF MAX ERROR?
2567 023350                                  BREAK
2568 023352                          ENDMSG
2569
2570
2571 023354                          BGNMSG  PDMAF                          ;DMA SHORT REPORT
2572 023354                          PRINTB  #MDMAF,ADDR,GDDAT,BDDAT
2573 023410                          ENDMSG
2574
2575
2576
2577
```

```
                                .SBTTL  REPORT CODING SECTION


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

023412                                  BGNRPT


023412                                  EXIT    RPT


023416                                  ENDRPT
```

```
                                .SBTTL  INITIALIZE SECTION

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

023420                                  BGNINIT



                                        .EVEN

023420                                  SETVEC  #140,#170000,#340       ;ODT ROM ADDRESS                ;JB REV A-0


                                ;INITIALIZE SUBROUTINE STACK
023446  012705  012732                  MOV     #SSTACK,R5
                                ;STORE BASE LEVEL PROGRAM STACK POINTER
023452  010637  002304                  MOV     SP,PSTACK
023456  005737  002316                  TST     FTIME
023462  001011                          BNE     1$
023464  013737  000004  002312          MOV     @#4,SAVE4
023472  013737  000006  002314          MOV     @#6,SAVE6
023500  012737  000001  002316          MOV     #1,FTIME
023506  013737  002312  000004  1$:     MOV     SAVE4,@#4
023514  013737  002314  000006          MOV     SAVE6,@#6

023522                                  READEF  #EF.START                       ;START COMMAND?
023530                                  BCOMPLETE       SETUP                   ;IF YES BRANCH

023532                                  READEF  #EF.CONTINUE                    ;CONTINUE COMMAND?
023540                                  BCOMPLETE       END


023542                                  READEF  #EF.NEW                         ;NEW PASS?
023550                                  BNCOMPLETE      NEXT                    ;IF NOT EXIT SETUP

023552  012737  177777  012472  SETUP:  MOV     #-1,UUT                         ;INITIALISE UNIT NUMBER

023560  005237  012472          NEXT:   INC     UUT                             ;POINT NEXT UNIT
023564  023737  012472  002270          CMP     UUT,L$UIT                       ;ALL DONE?
023572  001521                          BEQ     ABORT                           ;IF YES END OF PASS

023574  013701  012472                  MOV     UUT,R1
023600                                  PRINTF  #RUNNING,R1
                                        .EVEN

023622                                  GPHARD  UUT,R1                          ;GET P TABLE
023632                                  BNCOMPLETE      NEXT                    ;IF NOT AVAILABLE GET NEXT


023634                          GETPRM:
                                                                        ;GET ADDRESS OF KMV11
023634  011137  012510                  MOV     (R1),KMVCSR
                                                                        ;GET POINTER TO KMV11 SEL02 REG
```

```
2699 023640  011137  012512                MOV     (R1),KMVP02
2700 023644  062737  000002  012512        ADD     #2,KMVP02
2701                                                ;GET POINTER TO KMV11 PORT REG - SEL 4
2702 023652  011137  012514                MOV     (R1),KMVP04
2703 023656  062737  000004  012514        ADD     #4,KMVP04
2704                                                ;GET POINTER TO KMV11 PORT REG - SEL 6
2705 023664  011137  012516                MOV     (R1),KMVP06
2706 023670  062737  000006  012516        ADD     #6,KMVP06
2707                                                ;GET POINTER TO KMV11 REG 10
2708 023676  011137  012520                MOV     (R1),KMVP10
2709 023702  062737  000010  012520        ADD     #10,KMVP10
2710                                                ;GET POINTER TO KMV11 REG 12
2711 023710  011137  012522                MOV     (R1),KMVP12
2712 023714  062737  000012  012522        ADD     #12,KMVP12
2713                                                ;GET POINTER TO KMV11 REG 14
2714 023722  011137  012524                MOV     (R1),KMVP14
2715 023726  062737  000014  012524        ADD     #14,KMVP14
2716                                                ;GET POINTER TO KMV11 REG 16
2717 023734  012137  012526                MOV     (R1)+,KMVP16
2718 023740  062737  000016  012526        ADD     #16,KMVP16
2719                                                ;GET POINTER TO  VECTOR 0
2720 023746  011137  012474                MOV     (R1),KMVV00
2721                                                ;GET POINTER TO  VECTOR 2
2722 023752  011137  012502                MOV     (R1),KMVV02
2723 023756  062737  000002  012502        ADD     #2,KMVV02
2724                                                ;GET POINTER TO VECTOR 4
2725 023764  011137  012500                MOV     (R1),KMVV04
2726 023770  062737  000004  012500        ADD     #4,KMVV04
2727                                                ;GET POINTER TO VECTOR 6
2728 023776  012137  012504                MOV     (R1)+,KMVV06
2729 024002  062737  000006  012504        ADD     #6,KMVV06
2730                                                ;GET POINTER TO TX PRIORITY LEVEL
2731 024010  012137  012476                MOV     (R1)+,KMVLVL
2732 024014  062737  000006  012506        ADD     #6,KMTLVL
2733                                                ;GET LOOPBACK PARAMETERS:
2734 024022  011137  012530                MOV     (R1),LOOP
2735
2736 024026  005037  002300                CLR     ERRCNT          ;CLEAR ERROR COUNT
2737 024032                                EXIT    INIT
2738
2739
2740
2741 024036                        ABORT:  DOCLN                   ;CLEAN UP AND ABORT PASS
2742 024040                                EXIT INIT               ;EXIT
2743
2744                                        .NLIST  BEX
2745 024044     045     116     045  RUNNING:       .ASCIZ  /%N%A RUNNING ON UNIT %D2%A  /
2746                                        .LIST   BEX
2747                                        .EVEN
2748
2749
2750
2751 024102                        END:    ENDINIT
2752
2753
2754
2755
```

```
                                .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
024104                                  BGNAUTO



                                        ;CHECK IF EXISTING DEVICE

024104  013701  012510                  MOV     KMVCSR,R1       ;R1 CONTAINS BASE KMV11 ADDRESS
024110  012705  000007                  MOV     #7,R5           ;7 REGISTERS TO BE TESTED
024114  012737  024146  000004          MOV     #2$,4           ;SET OUT TIMEOUT TRAP
                        ;               MOV     #340,6          ;LEVEL 7                        ;JB REV A-0
024122  012737  000300  000006          MOV     #300,6          ;LEVEL 6                        ;JB REV A-0
024130  005711                  1$:     TST     (R1)            ;REFERENCE DEVICE REGISTERS
024132  000240                          NOP
024134  062701  000002                  ADD     #2,R1           ;NEXT REGISTER
024140  005305                          DEC     R5              ;DEC REGISTER COUNT
024142  001372                          BNE     1$              ;BR IF NOT LAST REGISTER
024144  000405                          BR      3$

024146  062706  000004          2$:     ADD     #4,SP
024152                                  DODU    LOGDEV

024160  013737  002312  000004  3$:     MOV     SAVE4,4
024166  013737  002314  000006          MOV     SAVE6,6



024174                                  ENDAUTO
```

```
                        .SBTTL  CLEANUP CODING SECTION
                        ;///////////////////////////////////////////////////////////////////////////
                        ;/ THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS PERFORMED
                        ;/ AT THE END OF EACH PASS.
                        ;///////////////////////////////////////////////////////////////////////////
 024176                         BGNCLN
 024176                         BRESET
 024200                         ENDCLN
```

```
                        .SBTTL  DROP UNIT SECTION

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

024202                  BGNDU




                        .EVEN

024202                  PRINTF  #DROPD,R0       ;UNIT DROPPED

024224                  EXIT    DU



                        .NLIST  BEX
024230  045 116 045 DROPD: .ASCIZ  /%N%A UNIT %D2%A DROPPED/
                        .LIST   BEX
                        .EVEN

024260                  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.
                                ;//////////////////////////////////////////////////////////////////////


024262                                  BGNAU
024262                                  ENDAU
```

```
2930
2931                                    .SBTTL  HARDWARE TESTS
2932
2933
2934
2935                                    ;START OF CODE BLOCK WHICH IS USED AS DATA
2936 024264                            ROMMAP:;++
2937                                    ; TEST TO ...
2938                                    ;--
2939
2946
2952
2953                                    ;       BGNTST
2954
2960
2961                                    ;       EXIT    TST
2962
2974
2975                                            .EVEN
2976                                    ;       ENDTST
2977
2983
```

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

                                BGNTST
                                        MOV     KMVCSR,R1       ;R1 CONTAINS KMV11 ADDRESSES
                                        MOV     #7,R5           ;7 REGISTERS TO BE TESTED
                                        MOV     #2$,4           ;SET OUT TIMEOUT TRAP
                                :       MOV     #340,6          ;LEVEL 7                        ;JB REV A-0
                                        MOV     #300,6          ;LEVEL 6                        ;JB REV A-0
                                1$:     TST     (R1)            ;REFERENCE DEVICE REGISTERS
                                        NOP
                                ESCAPE  TST
                                        ADD     #2,R1           ;NEXT REGISTER
                                        DEC     R5              ;DEC REGISTER COUNT
                                        BNE     1$              ;BR IF NOT LAST REGISTER
                                        BR      3$

                                2$:     ADD     #4,SP
                                        MOV     R1,ADDR         ;REPORT ADDRESS LOCATION
                                        MOV     UUT,UNIT        ;REPORT UNIT NUMBER
                                        ERRHRD  0,TIM,PADFLT    ;BUS TIMEOUT,ADDRESS PROBLEM ON THIS UNIT

                                3$:     MOV     SAVE4,4
                                        MOV     SAVE6,6
                                        ESCAPE  TST

                                ENDTST
                                .EVEN
```

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


                                BGNTST
                                BGNSUB

                                RESTST:
024404  005077  166100              CLR     @KMVCSR
024410  012777  054000  166072      MOV     #MAINTO,@KMVCSR         ;SET MASTER CLEAR TO EXIT
                                                                    ;SELF TEST IF RUNNING
024416                              WAITA   0


024430  012702  000010              MOV     #10,R2                  ;LOAD NUMBER OF REGISTER
024434  013701  012510              MOV     KMVCSR,R1
024440  005021              3$:     CLR     (R1)+                   ;CLR KMV11 REGISTERS
024442  000240                      NOP
024444  000240                      NOP
024446  005302                      DEC     R2
024450  001373                      BNE     3$                      ;ALL DONE?

024452  004537  013760              JSR     R5,CKALL                ;CHECK ALL REGISTERS = 0
024456  000000                      .WORD   0
024460  000000                      .WORD   0
024462  000000                      .WORD   0
024464  000000                      .WORD   0
024466  000000                      .WORD   0
024470  000000                      .WORD   0
024472  000000                      .WORD   0
024474  000000                      .WORD   0
024476  000406                      BR      2$                      ;OK BRANCH
024500                              ERRHRD  1,EM0002,PRALL          ;REGISTERS FAIL TO RESET
024510                              ESCAPE  SUB

024514  000240              2$:     NOP
024516                      ENDSUB


024520                      BGNSUB
024522  004737  014550              JSR     PC,CLRKMV               ;CLEAR REGISTERS

024526  012777  054000  165754      MOV     #MAINTO,@KMVCSR         ;SET MASTER CLEAR,MODE BIT AND MAINT

024534                              WAITA   0

024546  004537  013670              JSR     R5,CKSELO               ;CHECK MASTER CLR IS RESET BY DCT11
024552  014000                      .WORD   14000

024554  000406                      BR      1$                      ;YES
```

```
3072 024556                             ERRHRD  6,EM0001,PRSELO         ;MASTER CLR FAIL TO RESET
3073 024566                             ESCAPE  SUB
3074
3075 024572                         1$:
3076 024572                         ENDSUB
3077 024574                         ENDTST
3078
3079
3080
3081
```

```
                                BADHEAD
                                ;********************* TEST3 ********************
                                ;CHECK QBUS ACCESS ON KMV11 REGISTERS (FROM SEL2 TO SEL16)
024576                          BADHEAD
                                ;********************** TEST3 ***********************


                                STARS 1
                                ;SET MAINT MODE =0  ;DCT11 DECODE AND GOES IN HOLD
                                ;PROCESSOR SEND ROTATING PATTERN TO EACH REGISTERS AND CHECK
024576                          STARS 1



                                BGNTST
024576  004737  014550          TSTREG: JSR     PC,CLRKMV               ;CLEAR REGISTERS
024602  004737  014660                  JSR     PC,MAINMO               ;SET MAINT MODE 0
024606  012737  000007  002414          MOV     #7,COUNT                ;NUMBER OF REG
024614  012704  024652                  MOV     #CHECK,R4
024620  062704  000004                  ADD     #4,R4                   ;POINT GOOD VALUE OF SEL2
024624  013701  012512                  MOV     KMVP02,R1               ;LOAD SEL2 ADDRESS

024630  005003                  TSELA:  CLR     R3                      ;SELECT FIRST PATTERN

024632                          BREAK

024634  004737  013230          TSELB:  JSR     PC,GENER                ;GENER PATTERN

024640  013711  012436          1$:     MOV     DATA,(R1)                       ;LOAD PATTERN IN REG
024644  013714  012436                  MOV     DATA,(R4)                       ;LOAD GOOD VALUE
024650  000240                          NOP

024652  004537  014262          CHECK:  JSR     R5,CKREG                ;CHECK ALL REGISTER BUT SEL0
024656  000000                          .WORD   0
024660  000000                          .WORD   0
024662  000000                          .WORD   0
024664  000000                          .WORD   0
024666  000000                          .WORD   0
024670  000000                          .WORD   0
024672  000000                          .WORD   0
024674  000406                          BR      1$                      ;IF GOOD BR
024676                                  ERRHRD  7,EM0003,PRREG
024706                                  ESCAPE  TST

024712  005203          1$:     INC     R3                      ;NEW PATTERN
024714  022703  000007                  CMP     #7,R3                   ;ALL DONE
024720  001345                          BNE     TSELB                   ;NO BR

024722  005021                          CLR     (R1)+                   ;SELECT NEW REG
024724  005024                          CLR     (R4)+                   ;POINT NEW GOOD VALUE
024726  005337  002414                  DEC     COUNT                   ;ALL REG TESTED
024732  001356                          BNE     TSELA                   ;NO BR
```

```
    3138 024734                         ENDTST
```

```
3140 024736                          BADHEAD
                                     ;********************* TEST4 ********************
3141                                 ;CHECK QBUS ACCES ON SEL0 REGISTER
3142 024736                          BADHEAD
                                     ;********************* TEST4 ********************
3143
3144
3145
3146 024736                          STARS 1
3147                                 ;SET MAINT MODE 0      ;DCT11 HOLD
3148                                 ;SEND ROTATING PATTERN IN SEL0 (EXCEPT BIT 11,12,14) AND CHECK
3149 024736                          STARS 1
3150
3151
3152
3153
3154 024736                          BGNTST
3155 024736  004737  014550                  JSR     PC,CLRKMV               ;CLEAR REG
3156 024742  004737  014660                  JSR     PC,MAINMO               ;LOAD MAINT MODE 0
3157 024746  005003                          CLR     R3                      ;FIRST PATTERN
3158 024750  012704  025016                  MOV     #CHECK1,R4              ;POINT SEL 0
3159 024754  062704  000004                  ADD     #4,R4
3160 024760  012737  000015  002414  TCSRNB: MOV     #15,COUNT               ;SELECT NB 0 PATTERN
3161
3162 024766                                  BREAK
3163
3164 024770  004737  013230          TCSR:   JSR     PC,GENER                ;GERERATE A  PATTERN
3165 024774  042737  054000  012436          BIC     #54000,DATA             ;MASK   MCLR,MODE,MAINT1
3166 025002  013714  012436                  MOV     DATA,(R4)
3167 025006  013777  012436  165474          MOV     DATA,@KMVCSR            ;WRITE PATTERN
3168 025014  000240                          NOP
3169 025016  004537  013760          CHECK1: JSR     R5,CKALL                ;CHECK
3170 025022  000000                          .WORD   0
3171 025024  000000                          .WORD   0
3172 025026  000000                          .WORD   0
3173 025030  000000                          .WORD   0
3174 025032  000000                          .WORD   0
3175 025034  000000                          .WORD   0
3176 025036  000000                          .WORD   0
3177 025040  000000                          .WORD   0
3178 025042  000411                          BR      1$
3179 025044                                  ERRHRD  8,EM0004,PRALL
3180 025054                                  ESCAPE  TST
3181
3182 025060  005337  002414                  DEC     COUNT                   ;DONE ENOUGH
3183 025064  001341                          BNE     TCSR
3184
3185
3186 025066  005203                  1$:     INC     R3                      ;NEW PATTERN
3187 025070  022703  000007                  CMP     #7,R3                   ;ALL DONE
3188 025074  001331                          BNE     TCSRNB                  ;NO BR
3189 025076                          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,MAINMO               ;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$:     INC     R1                      ;SELECT NEW REGISTER
                DEC     R4                      ;DONE ALL?
                BNE     1$
```

```
3246 025230                    ENDTST
3247
```

```
                                BADHEAD
                                ;********************* TEST6 ********************
                                ;DATA TRANSFER ON REGISTER SEL 2
                                BADHEAD
                                ;********************* TEST6 ********************


                                STARS 1
                                ;SET MAINT1  ;DCT11 DECODE ,CLEAR SELO AND WAIT FOR TEST NUMBER
                                ;THE HOST WRITE A PATTERN IN SEL2
                                ;THE HOST WRITE A TEST NUMBER IN BSELO
                                ;
                                ;IF DCT11 READ CORRECT VALUE ,IT CLEAR BSELO
                                ;IF ERROR       SET 100 IN BSELO IF DATA CMP ERROR
                                ;               BSELO =TST NUMBER IF NO KMV11 ANSWER
                                ;
                                ;
                                ;BSELO=1 = MICRO DIAG TEST 1  ;DCT11 MUST READ 052525 IN SEL2
                                ;BSELO=2 = MICRO DIAG TEST 2  ;DCT11 MUST READ 125252 IN SEL2
                                STARS 1



                                BGNTST
                                        JSR     PC,CLRKMV               ;CLEAR REG
                                        JSR     PC,MAINM1               ;SET MAINT MODE 1

                                BGNSUB
                                        MOV     #DATA1,@KMVP02          ;SEND 052525
                                        JSR     R5,TSTNUB               ;SEND   TEST NUMB 1
                                        .WORD   1
                                        JSR     R5,CBSELO               ;CHECK BSELO = 0
                                        .WORD   0
                                        BR      1$                      ;TEST OK BR AT END
                                        JSR     R5,CBSELO               ;CHECK BSELO=100
                                        .WORD   100
                                        BR      2$
                                        BR      3$


                                2$:     ERRHRD  10,EM0005               ;DATA CMP ERROR
                                        JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
                                        ESCAPE  SUB

                                3$:     ERRHRD  11,EM0024               ;NO KMV11 ANSWER
                                        JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
                                        ESCAPE SUB
                                1$:     NOP
                                ENDSUB


                                BGNSUB
                                        JSR     PC,MAINM1               ;SET MAINT MODE 1
                                        MOV     #DATA2,@KMVP02          ;SEND   125252
```

```
3304 025362  004537  015010              JSR     R5,TSTNUB       ;SEND TEST NUB 2
3305 025366  000002                      .WORD   2
3306 025370  004537  013722              JSR     R5,CBSEL0       ;CHECK DCT11 HAS ANSWERED
3307 025374  000000                      .WORD   0               ;BY CLEARING SEL0
3308 025376  000425                      BR      1$              ;OK BR
3309 025400  004537  013722              JSR     R5,CBSEL0       ;CHECK IF +100
3310 025404  000100                      .WORD   100
3311 025406  000401                      BR      2$
3312 025410  000410                      BR      3$
3313
3314
3315
3316 025412                      2$:     ERRHRD  12,EM0005       ;DATA CMP ERROR ON SEL2
3317 025422  004737  013010              JSR     PC,CHKMAX       ;CHECK IF TOO MANY ERROR
3318 025426                              ESCAPE  SUB
3319
3320 025432                      3$:     ERRHRD  13,EM0024       ;NO KMV11 ANSWER
3321 025442  004737  013010              JSR     PC,CHKMAX       ;CHECK IF TOO MANY ERROR
3322 025446                              ESCAPE  SUB
3323
3324 025452  000240                      1$:     NOP
3325 025454                      ENDSUB
3326 025456                      ENDTST
```

C 7


```
       025460                                    BADHEAD
                                                 ;********************* TEST7 ******************
                                                 ; CHECK DATA TRANSFER ON REGISTER SEL4
       025460                                    BADHEAD
                                                 ;********************* TEST7 ******************



       025460                                    STARS 1
                                                 ;SET MAINT1  ;DCT11 DECODE AND CLEAR SELO
                                                 ;
                                                 ;THE HOST SEND ROTATING PATTERN IN SEL4,AND SET TEST NUMBER 3 IN BSELO
                                                 ;
                                                 ;DCT11 READ SEL4 , WRITE CONTENT OF SEL4 INTO SEL2 , CLEAR SELO WHEN DONE, AND
                                                 ;WAIT FOR NEW PATTERN
                                                 ;
                                                 ;
                                                 ;
                                                 ;       AFTER TEST       BSELO=100 IF ERROR DURING TEST
                                                 ;                        BSELO=0    IF TEST HAS BEEN EXECUTED (IN THAT CASE
                                                 ;                                      CHECK IF CONTENT OF SEL2 IS CORRECT)
                                                 ;
                                                 ;BSEL 0 = 3 ;MICRO DIAG NB 3 ;DCT11 TAKE CONTENT OF SEL4 AND PUT IT IN SEL 2.
       025460                                    STARS 1


       025460                                    BGNTST
       025460                                    BGNSUB
       025462  004737  014550                            JSR     PC,CLRKMV               ;CLEAR REG
       025466  004737  014730                            JSR     PC,MAINM1               ;SET MAINT MODE 1
       025472  005003                                    CLR     R3                      ;SELECT FIRST PATTERN

       025474  012737  000005  002414            TGENE1: MOV     #5,COUNT                ;NB OF PATTERN
       025502                                            BREAK

       025504  004737  013230                    GENE1:  JSR     PC,GENER                ;GENER 1ST PATTERN
       025510  013777  012436  164776                    MOV     DATA,@KMVP04            ;LOAD SEL4
       025516  004537  015010                            JSR     R5,TSTNUB               ;SET TEST NUMB  3
       025522  000003                                    .WORD   3
       025524  004537  013722                            JSR     R5,CBSELO               ;LOOK IF ANSWER
       025530  000000                                    .WORD   0
       025532  000404                                    BR      1$                      ;OK BR
       025534                                            ERRHRD  14,EM0024,PBSELO        ;BSELO NOT CLEARED ,NO  ANSWER
       025544  000240                            1$:     NOP
       025546                                    ENDSUB


       025550                                    BGNSUB
       025552  012704  025572                            MOV     #TSEL4,R4               ;POINT GOOD VALUE
       025556  013764  012436  000006                    MOV     DATA,6(R4)
       025564  013764  012436  000004                    MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL4

       025572  004537  014262                    TSEL4:  JSR     R5,CKREG                ;CHECK SEL2 = SEL4
       025576  000000                                    .WORD   0
       025600  000000                                    .WORD   0
       025602  000000                                    .WORD   0
       025604  000000                                    .WORD   0
```

```
3383 025606  000000                        .WORD   0
3384 025610  000000                        .WORD   0
3385 025612  000000                        .WORD   0
3386 025614  000406                        BR      2$
3387 025616                                ERRHRD  15,EM0006,PRREG
3388 025626                                ESCAPE  SUB
3389
3390
3391 025632  005337  002414          2$:   DEC     COUNT                   ;DONE ENOUGH?
3392 025636  001322                        BNE     GENE1
3393
3394 025640  005203                        INC     R3                      ;NEW PATTERN
3395 025642  022703  000007                CMP     #7,R3                   ;ALL DONE
3396 025646  001312                        BNE     TGENE1                  ;NO BR
3397 025650                        ENDSUB
3398 025652                        ENDTST
```

```
3400 025654                          BADHEAD
                                     ;********************* TEST8 ********************
3401                                 ; CHECK DATA TRANSFER ON REGISTER SEL6
3402 025654                          BADHEAD
                                     ;********************* TEST8 ********************
3403
3404
3405 025654                          STARS 1
3406                                 ;SET MAINT1  ;DCT11 DECODE AND CLEAR SEL0
3407                                 ;
3408                                 ;THE HOST SENDS A ROTATING PATTERN IN SEL6,AND SET TEST NUMBER 4 IN BSEL0
3409                                 ;
3410                                 ;DCT11 READ SEL6 , WRITE CONTENT OF SEL6 IN SEL2 , CLEAR SEL0 WHEN DONE, AND
3411                                 ;WAIT FOR NEW PATTERN
3412                                 ;
3413                                 ;
3414                                 ;
3415                                 ;      AFTER TEST       BSEL0=100 IF ERROR DURING TEST
3416                                 ;                       BSEL0=0    IF TEST HAS BEEN EXECUTED (IN THAT CASE
3417                                 ;                                        CHECK IF CONTENT OF SEL4 IS CORRECT)
3418                                 ;
3419                                 ;BSEL 0 = 4 ;MICRO DIAG NB 4 ;DCT11 TAKE CONTENT OF SEL6 AND PUT IT IN SEL 2.
3420 025654                          STARS 1
3421
3422
3423
3424 025654                          BGNTST
3425 025654                          BGNSUB
3426 025656  004737  014550                  JSR     PC,CLRKMV               ;CLEAR REG
3427 025662  004737  014730                  JSR     PC,MAINM1               ;SET MAINT MODE 1
3428 025666  012703  000001                  MOV     #1,R3                   ;SELECT 1ST PATTERN
3429 025672  012737  000005  002414  TGENE2: MOV     #5,COUNT                ;    "   NUMBER OF PATTERN
3430 025700                                  BREAK
3431
3432
3433 025702  004737  013230          GENE2:  JSR     PC,GENER                ;GENERATE 1ST PATTERN
3434 025706  013777  012436  164602          MOV     DATA,@KMVP06            ;LOAD SEL6
3435 025714  004537  015010                  JSR     R5,TSTNUB               ;SET TEST NUMB  4
3436 025720  000004                          .WORD   4
3437 025722  004537  013722                  JSR     R5,CBSEL0               ;LOOK IF ANSWER
3438 025726  000000                          .WORD   0
3439 025730  000404                          BR      1$                      ;OK BR
3440 025732                                  ERRHRD  16,EM0024,PBSEL0        ;NO KMV11 ANSWER,BSEL0 NOT = 0
3441 025742  000240                  1$:     NOP
3442 025744                          ENDSUB
3443
3444
3445 025746                          BGNSUB
3446 025750  012704  025770                  MOV     #TSEL6,R4               ;POINT GOOD VALUE
3447 025754  013764  012436  000010          MOV     DATA,10(R4)
3448 025762  013764  012436  000004          MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL6
3449
3450 025770  004537  014262          TSEL6:  JSR     R5,CKREG                ;CHECK SEL2 = SEL6
3451 025774  000000                          .WORD   0
3452 025776  000000                          .WORD   0
3453 026000  000000                          .WORD   0
3454 026002  000000                          .WORD   0
```

```
3455 026004  000000                          .WORD   0
3456 026006  000000                          .WORD   0
3457 026010  000000                          .WORD   0
3458 026012  000406                          BR      2$
3459 026014                                  ERRHRD  17,EM0007,PRREG
3460 026024                                  ESCAPE  SUB
3461 026030  005337  002414          2$:     DEC     COUNT
3462 026034  00.322                         BNE     GENE2
3463
3464 026036  005203                          INC     R3                      ;NEW PATTERN
3465 026040  022703  000006                  CMP     #6,R3                   ;ALL DONE
3466 026044  001312                          BNE     TGENE2                  ;NO BR
3467 026046                          ENDSUB
3468 026050                          ENDTST
```

```
3470 026052                          BADHEAD
                                     ;********************* TEST9 ********************
3471                                 ;TEST TO CHECK DATA TRANSFER ON REGISTER SEL10
3472 026052                          BADHEAD
                                     ;********************* TEST9 ********************
3473
3474
3475 026052                          STARS 1
3476                                 ;SET MAINT1  ;DCT11 DECODE AND CLEAR SEL0
3477                                 ;
3478                                 ;THE HOST SENDS A ROTATING PATTERN IN SEL10,AND SET TEST NUMBER 5 IN BSEL0
3479                                 ;
3480                                 ;DCT11 READ SEL10 , WRITE CONTENT OF SEL10 IN SEL2 , CLEAR SEL0 WHEN DONE, AND
3481                                 ;WAIT FOR NEW PATTERN
3482                                 ;
3483                                 ;
3484                                 ;
3485                                 ;      AFTER TEST       BSEL0=100 IF ERROR DURING TEST
3486                                 ;                       BSEL0=0    IF TEST HAS BEEN EXECUTED (IN THAT CASE
3487                                 ;                                        CHECK IF CONTENT OF SEL6 IS CORRECT)
3488                                 ;
3489                                 ;BSEL 0 = 5 ;MICRO DIAG NB 5 ;DCT11 TAKE CONTENT OF SEL10 AND PUT IT IN SEL 2.
3490 026052                          STARS 1
3491
3492
3493
3494 026052                          BGNTST
3495 026052                          BGNSUB
3496 026054  004737  014550                  JSR     PC,CLRKMV               ;CLEAR REG
3497 026060  004737  014730                  JSR     PC,MAINM1               ;SET MAINT MODE 1
3498 026064  012703  000001                  MOV     #1,R3                   ;SELECT FIRST PATTERN
3499
3500 026070  012737  000005  002414  TGENE3: MOV     #5,COUNT
3501 026076                          BREAK
3502
3503
3504 026100  004737  013230          GENE3:  JSR     PC,GENER                ;GENERATE A 1ST PATTERN
3505 026104  042737  040000  012436          BIC     #40000,DATA             ;MASK BIT 14
3506 026112  013777  012436  164400          MOV     DATA,@KMVP10            ;LOAD SEL10
3507 026120  004537  015010                  JSR     R5,TSTNUB               ;SET TEST NUMB  5
3508 026124  000005                          .WORD   5
3509 026126  004537  013722                  JSR     R5,CBSEL0               ;LOOK IF ANSWER
3510 026132  000000                          .WORD   0
3511 026134  000406                          BR      1$                      ;OK BR
3512 026136                                  ERRHRD  20,EM0024,PBSEL0        ;NO KMV11 ANSWER
3513 026146                                  ESCAPE  SUB
3514 026152  000240                  1$:     NOP
3515 026154                          ENDSUB
3516
3517
3518 026156                          BGNSUB
3519 026160  012704  026200                  MOV     #TSEL10,R4              ;POINT GOOD VALUE
3520 026164  013764  012436  000012          MOV     DATA,12(R4)
3521 026172  013764  012436  000004          MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL10
3522
3523 026200  004537  014262          TSEL10: JSR     R5,CKREG                ;CHECK SEL2 = SEL10
3524 026204  000000                          .WORD   0
```

```
 3525 026206  000000                        .WORD   0
 3526 026210  000000                        .WORD   0
 3527 026212  000000                        .WORD   0
 3528 026214  000000                        .WORD   0
 3529 026216  000000                        .WORD   0
 3530 026220  000000                        .WORD   0
 3531 026222  000406                        BR      2$
 3532 026224                                ERRHRD  21,EM0010,PRREG         ;DATA CMP ERROR IN SEL10
 3533 026234                                ESCAPE  SUB
 3534
 3535 026240  005337  002414          2$:   DEC     COUNT
 3536 026244  001315                        BNE     GENE3
 3537
 3538 026246  005203                        INC     R3                      ;NEW PATTERN
 3539 026250  022703  000006                CMP     #6,R3                   ;ALL DONE
 3540 026254  001305                        BNE     TGENE3                  ;NO BR
 3541 026256                        ENDSUB
 3542 026260                        ENDTST
```

```
3544 026262                          BADHEAD
                                     ;********************* TEST10 ********************
3545                                 ;TEST TO CHECK DATA TRANSFER ON REGISTER SEL12
3546 026262                          BADHEAD
                                     ;********************* TEST10 ********************
3547
3548
3549 026262                          STARS 1
3550                                 ;SET MAINT1  ;DCT11 DECODE AND CLEAR SELO
3551                                 ;
3552                                 ;THE HOST SENDS A ROTATING PATTERN IN SEL12,AND SET TEST NUMBER 6 IN BSELO
3553                                 ;
3554                                 ;DCT11 READ SEL12 , WRITE CONTENT OF SEL12 IN SEL 2 , CLEAR SELO WHEN DONE, AND
3555                                 ;WAIT FOR NEW PATTERN
3556                                 ;
3557                                 ;
3558                                 ;
3559                                 ;      AFTER TEST       BSELO=100 IF ERROR DURING TEST
3560                                 ;                       BSELO=0    IF TEST HAS BEEN EXECUTED (IN THAT CASE
3561                                 ;                                        CHECK IF CONTENT OF SEL10 IS CORRECT)
3562                                 ;
3563                                 ;BSEL 0 = 6 ;MICRO DIAG NB 6 ;DCT11 TAKE CONTENT OF SEL12 AND PUT IT IN SEL 2.
3564 026262                          STARS 1
3565
3566
3567
3568 026262                          BGNTST
3569 026262                          BGNSUB
3570 026264  004737  014550                  JSR     PC,CLRKMV               ;CLEAR REG
3571 026270  004737  014730                  JSR     PC,MAIN#1               ;SET MAINT MODE 1
3572 026274  012703  000001                  MOV     #1,R3
3573
3574 026300  012737  000005  002414  TGENE4: MOV     #5,COUNT
3575 026306                          BREAK
3576
3577
3578 026310  004737  013230          GENE4:  JSR     PC,GENER                ;GENERATE A 1ST PATTERN
3579 026314  013777  012436  164200          MOV     DATA,@KMVP12            ;LOAD SEL12
3580 026322  004537  015010                  JSR     R5,TSTNUB               ;SET TEST NUMB  6
3581 026326  000006                          .WORD   6
3582 026330  004537  013722                  JSR     R5,CBSELO               ;LOOK IF ANSWER
3583 026334  000000                          .WORD   0
3584 026336  000406                          BR      1$                      ;OK BR
3585 026340                                  ERRHRD  22,EM0024,PBSELO                ;NO KMV11 ANSWER
3586 026350                                  ESCAPE  SUB
3587 026354  000240                  1$:     NOP
3588 026356                          ENDSUB
3589
3590
3591 026360                          BGNSUB
3592 026362  012704  026402                  MOV     #TSEL12,R4              ;POINT GOOD VALUE
3593 026366  013764  012436  000014          MOV     DATA,14(R4)
3594 026374  013764  012436  000004          MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL12
3595
3596 026402  004537  014262          TSEL12: JSR     R5,CKREG                ;CHECK SEL2 = SEL12
3597 026406  000000                          .WORD   0
3598 026410  000000                          .WORD   0
```

```
3599 026412  000000                          .WORD   0
3600 026414  000000                          .WORD   0
3601 026416  000000                          .WORD   0
3602 026420  000000                          .WORD   0
3603 026422  000000                          .WORD   0
3604 026424  000406                          BR      2$
3605 026426                                  ERRHRD  23,EM0011,PRREG         ;DATA CMP ERROR IN SEL 12
3606 026436                                  ESCAPE  SUB
3607
3608
3609 026442  005337  002414          2$:     DEC     COUNT
3610 026446  001320                          BNE     GENE4
3611
3612 026450  005203                          INC     R3                      ;NEW PATTERN
3613 026452  022703  000006                  CMP     #6,R3                   ;ALL DONE
3614 026456  001310                          BNE     TGENE4                  ;NO BR
3615 026460                          ENDSUB
3616 026462                          ENDTST
```

```
3618 026464                          BADHEAD
                                     ;********************* TEST11 *********************
3619                                 ; CHECK DATA TRANSFER ON REGISTER SEL14
3620 026464                          BADHEAD
                                     ;********************* TEST11 *********************
3621
3622
3623 026464                          STARS 1
3624                                 ;SET MAINT1  ;DCT11 DECODE AND CLEAR SELO
3625                                 ;
3626                                 ;THE HOST SEND ROTATING PATTERN IN SEL14,AND SET TEST NUMBER 6 IN BSELO
3627                                 ;
3628                                 ;DCT11 READ SEL14 , WRITE CONTENT OF SEL14 IN SEL 2 , CLEAR SELO WHEN DONE, AND
3629                                 ;WAIT FOR NEW PATTERN
3630                                 ;
3631                                 ;
3632                                 ;
3633                                 ;      AFTER TEST       BSELO=100 IF ERROR DURING TEST
3634                                 ;                       BSELO=0    IF TEST HAS BEEN EXECUTED (IN THAT CASE
3635                                 ;                                       CHECK IF CONTENT OF SEL12 IS CORRECT)
3636                                 ;
3637                                 ;BSEL 0 = 6 ;MICRO DIAG NB 6 ;DCT11 TAKE CONTENT OF SEL14 AND PUT IT IN SEL 12.
3638 026464                          STARS 1
3639
3640
3641 026464                          BGNTST
3642 026464                          BGNSUB
3643 026466  004737  014550                  JSR     PC,CLRKMV               ;CLEAR REG
3644 026472  004737  014730                  JSR     PC,MAINM1               ;SET MAINT MODE 1
3645 026476  012703  000001                  MOV     #1,R3                   ;SELECT FIRST PATTERN
3646
3647 026502  012737  000005  002414  TGENE5: MOV     #5,COUNT                ;SELECT NB OF PATTERN
3648 026510                                  BREAK
3649
3650
3651 026512  004737  013230          GENE5:  JSR     PC,GENER                ;GENERATE A  1ST PATTERN
3652 026516  013777  012436  164000          MOV     DATA,@KMVP14            ;LOAD SEL14
3653 026524  004537  015010                  JSR     R5,TSTNUB               ;SET TEST NUMB  7
3654 026530  000007                          .WORD   7
3655 026532  004537  013722                  JSR     R5,CBSELO               ;LOOK IF ANSWER
3656 026536  000000                          .WORD   0
3657 026540  000406                          BR      1$                      ;OK BR
3658 026542                                  ERRHRD  24,EM0024,PBSELO                ;NO KMV11 ANSWER
3659 026552                                  ESCAPE  SUB
3660
3661 026556  000240                  1$:     NOP
3662 026560                          ENDSUB
3663
3664
3665 026562                          BGNSUB
3666 026564  012704  026604                  MOV     #TSEL14,R4              ;POINT GOOD VALUE
3667 026570  013764  012436  000016          MOV     DATA,16(R4)
3668 026576  013764  012436  000004          MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL4
3669
3670 026604  004537  014262          TSEL14: JSR     R5,CKREG                ;CHECK SEL2 = SEL14
3671 026610  000000                          .WORD   0
3672 026612  000000                          .WORD   0
```

```
 3673 026614  000000                          .WORD   0
 3674 026616  000000                          .WORD   0
 3675 026620  000000                          .WORD   0
 3676 026622  000000                          .WORD   0
 3677 026624  000000                          .WORD   0
 3678 026626  000406                          BR      2$
 3679 026630                                  ERRHRD  25,EM0012,PRREG         ;DATA CMP ERROR IN SEL 14
 3680 026640                                  ESCAPE  SUB
 3681
 3682
 3683 026644  005337  002414          2$:     DEC     COUNT                   ;DONE ENOUGH?
 3684 026650  001320                          BNE     GENE5
 3685
 3686 026652  005203                          INC     R3                      ;NEW PATTERN
 3687 026654  022703  000006                  CMP     #6,R3                   ;ALL DONE
 3688 026660  001310                          BNE     TGENE5                  ;NO BR
 3689 026662                          ENDSUB
 3690 026664                          ENDTST
```

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


                                STARS 1
                                ;SET MAINT1  ;DCT11 DECODE AND CLEAR SELO
                                ;
                                ;THE HOST SENDS A ROTATING PATTERN IN SEL16,AND SET TEST NUMBER 7 IN BSELO
                                ;
                                ;DCT11 READ SEL16 , WRITE CONTENT OF SEL16 IN SEL 2 , CLEAR SELO WHEN DONE, AND
                                ;WAIT FOR NEW PATTERN
                                ;
                                ;
                                ;
                                ;       AFTER TEST      BSELO=100 IF ERROR DURING TEST
                                ;                       BSELO=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
026670  004737  014550                  JSR     PC,CLRKMV               ;CLEAR REG
026674  004737  014730                  JSR     PC,MAINM1               ;SET MAINT MODE 1
026700  012703  000001                  MOV     #1,R3                   ;SELECT FIRST PATTERN

026704  012737  000005  002414  TGENE6. MOV     #5 COUNT
026712                          BREAK


026714  004737  013230          GENE6:  JSR     PC,GENER                ;GENERATE 1ST PATTERN
026720  013777  012436  163600          MOV     DATA,@KMVP16            ;LOAD SEL16
026726  004537  015010                  JSR     R5,TSTNUB               ;SET TEST NUMB  10
026732  000010                          .WORD   10
026734  004537  013722                  JSR     R5,CBSELO               ;LOOK IF ANSWER
026740  000000                          .WORD   0
026742  000406                          BR      1$                      ;OK BR
026744                                  ERRHRD  26,EM0024,PBSELO                ;NO KMV11 ANSWER
026754                                  ESCAPE  SUB

026760  000240                  1$:     NOP
026762                          ENDSUB


026764                          BGNSUB
026766  012704  027006                  MOV     #TSEL16,R4              ;POINT GOOD VALUE
026772  013764  012436  000020          MOV     DATA,20(R4)
027000  013764  012436  000004          MOV     DATA,4(R4)              ;WRITE GOOD VALUE FOR SEL2 AND SEL16

027006  004537  014262          TSEL16: JSR     R5,CKREG                ;CHECK SEL2 = SEL16
027012  000000                          .WORD   0
027014  000000                          .WORD   0
027016  000000                          .WORD   0
```

```
3747 027020  000000                        .WORD   0
3748 027022  000000                        .WORD   0
3749 027024  000000                        .WORD   0
3750 027026  000000                        .WORD   0
3751 027030  000406                        BR      2$
3752 027032                                ERPHRD  27,EM0013,PRREG
3753 027042                                ESCAPE  SUB
3754
3755
3756 027046  005337  002414          2$:   DEC     COUNT                   ;DONE ENOUGH?
3757 027052  001320                        BNE     GENE6
3758
3759 027054  005203                        INC     R3                      ;NEW PATTERN
3760 027056  022703  000006                CMP     #6,R3                   ;ALL DONE
3761 027062  001310                        BNE     TGENE6                  ;NO BR
3762 027064                        ENDSUB
3763 027066                        ENDTST
3764
3765
3766
```

```
                                BADHEAD
                                ;********************* TEST13 ********************
                                ; CHECK DATA TRANSFER ON ALL REGISTERS
                                BADHEAD
                                ;********************* TEST13 ********************
                                STARS 1
                                ;SET MAINT1  DCT11 CLEAR BSEL0
                                ;SEND DIFFERENT PATTERN IN SEL2,  SEND TEST 11
                                ;DCT11 READ SEL2 AND WRITTE A CALCULATED VALUE  IN SEL4 TO SEL16
                                ;                (SEL4)=SEL2+SEL2
                                ;                (SEL6)=SEL4+SEL2
                                ;                (SEL10)=SEL6+SEL2
                                ;                (SEL12)=SEL10+SEL2
                                ;                (SEL14)=SEL12+SEL2
                                ;                (SEL16)=SEL14+SEL2
                                ;DCT11 CLEAR BSEL0 WHEN DONE
                                ;
                                ;-MICRO DIAG NUMBER 11
                                ;
                                ;
                                STARS 1
                                BGNTST
027070  004737  014550          BGNTST: JSR     PC,CLRKMV               ;CLEAR REG
027074  004737  014730                  JSR     PC,MAINM1               ;SET MAINT1
                                ;
027100  012703  000004                  MOV     #4,R3                   ;PREPARE INCREMENTING PATTERN
027104  012737  000007  002416          MOV     #7,NUMBER               ;SELECT NUMBER FOR DIFFERENT PATTERN
                                ;
027112  004737  013230          RGALL:  JSR     PC,GENER                ;PREPARE ONE RANDOM PETTERN
                                ;
027116  013777  012436  163366          MOV     DATA,@KMVP02            ;WRITE PATTERN IN SEL2
027124  013737  012436  002336          MOV     DATA,GOOD2
027132  004537  015010                  JSR     R5,TSTNUB               ;SEND TEST NB11
027136  000011                          .WORD   11
                                ;
027140  012737  177700  002324          MOV     #177700,DELCT1
027146  004737  013000                  JSR     PC,WAIT1
                                ;
027152                                  BREAK
                                ;
027154  004537  013722                  JSR     R5,CBSEL0               ;LOOK IF TEST DONE
```

```
3823 027160  000000                        .WORD   0
3824 027162  000406                        BR      1$                      ;YES BRANCH
3825 027164                                ERRHRD  28,EM0024,PBSEL0        ;NO KMV11 ANSWER
3826 027174                                ESCAPE  TST
3827
3828
3829 027200  017737  163306  002360  1$:   MOV     @KMVP02,SEL2            ;READ SEL2 TO SEL16
3830 027206  017737  163302  002362        MOV     @KMVP04,SEL4
3831 027214  017737  163276  002364        MOV     @KMVP06,SEL6
3832 027222  017737  163272  002366        MOV     @KMVP10,SEL10
3833 027230  017737  163266  002370        MOV     @KMVP12,SEL12
3834 027236  017737  163262  002372        MOV     @KMVP14,SEL14
3835 027244  017737  163256  002374        MOV     @KMVP16,SEL16
3836
3837
3838 027252  013737  002336  002340        MOV     GOOD2,GOOD4
3839 027260  063737  002336  002340        ADD     GOOD2,GOOD4
3840
3841 027266  013737  002340  002342        MOV     GOOD4,GOOD6             ;WHAT IS GOOD6
3842 027274  063737  002336  002342        ADD     GOOD2,GOOD6
3843
3844 027302  013737  002342  002344        MOV     GOOD6,GOOD10            ;WHAT IS GOOD10
3845 027310  063737  002336  002344        ADD     GOOD2,GOOD10
3846
3847 027316  013737  002344  002346        MOV     GOOD10,GOOD12           ;  "    " GOOD12
3848 027324  063737  002336  002346        ADD     GOOD2,GOOD12
3849
3850 027332  013737  002346  002350        MOV     GOOD12,GOOD14           ;  "    "  GOOD14
3851 027340  063737  002336  002350        ADD     GOOD2,GOOD14
3852
3853 027346  013737  002350  002352        MOV     GOOD14,GOOD16           ;    "  "  GOOD16
3854 027354  063737  002336  002352        ADD     GOOD2,GOOD16
3855
3856
3857 027362  023737  002336  002360        CMP     GOOD2,SEL2
3858 027370  001031                        BNE     2$
3859 027372  023737  002340  002362        CMP     GOOD4,SEL4
3860 027400  001025                        BNE     2$
3861 027402  023737  002342  002364        CMP     GOOD6,SEL6
3862 027410  001021                        BNE     2$
3863 027412  023737  002344  002366        CMP     GOOD10,SEL10
3864 027420  001015                        BNE     2$
3865 027422  023737  002346  002370        CMP     GOOD12,SEL12
3866 027430  001011                        BNE     2$
3867 027432  023737  002350  002372        CMP     GOOD14,SEL14
3868 027440  001005                        BNE     2$
3869 027442  023737  002352  002374        CMP     GOOD16,SEL16
3870 027450  001001                        BNE     2$
3871 027452  000410                        BR      3$
3872
3873
3874 027454                          2$:   BREAK
3875 027456                                ERRHRD  29,EM0003,PRREG
3876 027466                                BREAK
3877 027470                                ESCAPE  TST
3878
3879
```

```
 3880 027474  005337  002416          5$:     DEC     NUMBER              ;ALL PATTERN DONE?
 3881 027500                                  BREAK
 3882 027502  001203                          BNE     RGALL
 3883 027504                          ENDTST
```

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



                                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 ERR..
                                ;
                                ;
                                ;
                                ;
                                ;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
                                STARS 1



                                BGNTST
                                RAMPAT: JSR     PC,CLRKMV               ;CLEAR REG
                                        JSR     PC,MAINM1               ;SET MAINT1

                                        CLR     R3                      ;SELECT 1ST PATTERN
                                        CLR     COUNT
                                4$:     JSR     PC,GENER                ;MAKE PATTERN
                                        MOV     DATA,@KMVP02            ;WRITE PATTERN IN SEL2

                                        JSR     R5,TSTNUB               ;SET TEST NB 13
                                        .WORD   13

                                        WAITB   0,1

                                        JSR     PC,TSTERR                       ;CHECK BSEL0=WHICH ERROR?
                                        BR      1$                      ;TEST OK
```

```
3940 027572  000420                        BR      2$
3941                                                        ;TIME OUT ERROR
3942 027574  000427                        BR      3$      ;NO KMV11 ANSWER
3943
3944
3945
3946
3947
3948 027576  017737  162714  002420        MOV     @KMVP06,ADDR        ;READ   ADDRESS OF RAM
3949 027604  017737  162702  002422        MOV     @KMVP02,GDDAT       ;READ EXPECTED DATA (GDDAT)
3950 027612  017737  162676  002424        MOV     @KMVP04,BDDAT       ;READ BAD VALUE OF DATA (BDDAT)
3951
3952 027620                                ERRHRD  30,EM0015,PRRAM     ;DATA CMP ERROR ON ONE RAM LOCATION
3953 027630                                ESCAPE  TST
3954
3955
3956
3957
3958
3959 027634  005037  002330          2$:   CLR GOOD
3960 027640                                ERRHRD  31,EM0025,PBSELO    ;TIMEOUT ERROR
3961 027650                                ESCAPE TST
3962
3963
3964
3965
3966
3967 027654  005037  002330          3$:   CLR     GOOD
3968 027660                                ERRHRD  32,EM0024,PBSELO    ;NO KMV11 ANSWER
3969 027670                                ESCAPE  TST
3970
3971
3972
3973
3974
3975 027674  005237  002414          1$:   INC     COUNT
3976 027700  022737  000015  002414        CMP     #15,COUNT           ;SEND 1 WORDS IN THE SAME PATTERN
3977 027706  001306                        BNE     4$                  ;15 WORDS DONE BR
3978
3979 027710  005037  002414                CLR     COUNT
3980 027714  005203                        INC     R3                  ;TRY WITH A NEW PATTERN
3981 027716  022703  000006                CMP     #6,R3               ;ALL DONE ?
3982 027722  001300                        BNE     4$                  ;NO BR
3983 027724                          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 BSELO IF TEST IS OK
                                ;            BSELO= 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
                                ;
                                ;
                                ;
                                ;
                                ;BSELO = 14 ,  MICRO DIAGNOSTIC TEST NUMBER 14
                                ;              DCT11 WRITE ADDRESS VALUE IN ADDRESS LOCATION AND CHECK
                                STARS 1



                                BGNTST
                                RAMADD: JSR     PC,CLRKMV               ;CLEAR REG
                                        JSR     PC,MAINM1               ;SET MAINT1

                                        JSR     R5,TSTNUB               ;SET TEST NB 14
                                        .WORD   14

                                        WAITB   0,1



                                        JSR     PC,TSTERR               ;CHECK BSELO
                                        BR      1$                      ;TEST OK
                                        BR      2$                      ;TIMEOUT ERROR
                                        BR      3$                      ;NO KMV11 ANSWER



                                        MOV     @KMVP06,ADDR            ;READ   ADDRESS OF RAM
                                        MOV     @KMVP02,GOOD            ;READ EXPECTED DATA (GDDAT)
                                        MOV     @KMVP04,BDDAT           ;READ BAD VALUE OF DATA (BDDAT)

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

```
4040
4041
4042
4043
4044 030034  005037  002330          2$:     CLR     GOOD
4045 030040                                  ERRHRD  34,EM0025,PBSEL0        ;TIMEOUT ERROR
4046 030050                                  ESCAPE TST
4047
4048
4049
4050
4051 030054  005037  002330          3$:     CLR     GOOD
4052 030060                                  ERRHRD  35,EM0024,PBSEL0        ;NO KMV11 ANSWER
4053 030070                                  ESCAPE TST
4054 030074  000240                  1$:     NOP
4055 030076                          ENDTST
```

```
4057 030100                            BADHEAD
                                       ;********************* TEST16 *********************
4058                                   ;KMV11 RAM MEMORY TEST:  MEMORY ADDRESS COMPLEMENT TEST
4059 030100                            BADHEAD
                                       ;********************* TEST16 *********************
4060
4061
4062
4063
4064 030100                            STARS 1
4065                                   ;SET MAINT MODE 1    ;DCT11 DECODE AND CLEAR SEL 0
4066                                   ;SET TEST NUMBER 15   ;DCT11 EXECUTE TEST
4067                                   ;DCT11 WRITE COMPLEMENT ADDRESS VALUE IN EACH ADDRESS LOCATION AND CHECK.
4068                                   ;
4069                                   ;DCT11 CLEAR BSEL0 IF TEST OK AND PUT 100 IN BSEL0 IF DATA COMPARE ERROR
4070                                   ;
4071                                   ;IF ERROR SEL2 = EXPECTED VALUE OF LOCATION IN ERROR
4072                                   ;         SEL4 = READ VALUE OF LOCATION IN ERROR
4073                                   ;         SEL6 = ADDRESS IN ERROR
4074                                   ;
4075                                   ;
4076                                   ;
4077                                   ;
4078                                   ;BSEL0 = 15 ,  MICRO DIAGNOSTIC TEST NUMBER 15
4079                                   ;               DCT11 WRITE COMPL. ADDRESS IN ADDRESS IN RAM MEMORY AND CHECK
4080 030100                            STARS 1
4081
4082
4083
4084
4085 030100                            BGNTST
4086 030100  004737  014550            RAMCAD: JSR     PC,CLRKMV               ;CLEAR REG
4087 030104  004737  014730                    JSR     PC,MAINM1               ;SET MAINT1
4088
4089 030110  004537  015010                    JSR     R5,TSTNUB               ;SET TEST NB 15
4090 030114  000015                            .WORD   15
4091
4092 030116                                    WAITB   0,1
4093
4094
4095
4096 030136  004737  013152                    JSR     PC,TSTERR               ;CHECK  BSEL0,WHICH ERROR
4097 030142  000441                            BR      1$                      ;TEST OK
4098 030144  000420                            BR      2$                      ;TIMEOUT ERROR
4099 030146  000427                            BR      3$                      ;NO KMV11 ANSWER
4100
4101
4102 030150  017737  162342  002420            MOV     @KMVP06,ADDR            ;READ   ADDRESS OF RAM
4103 030156  017737  162330  002330            MOV     @KMVP02,GOOD            ;READ EXPECTED DATA (GDDAT)
4104 030164  017737  162324  002424            MOV     @KMVP04,BDDAT           ;READ BAD VALUE OF DATA (BDDAT)
4105
4106 030172                                    ERRHRD  36.EM0015,PRRAM         ;DATA CMP ERROR ON ONE RAM LOCATION
4107 030202                                    ESCAPE  TST
4108
4109
4110
4111
```

```
4112 030206  005037  002330          2$:     CLR     GOOD
4113 030212                                  ERRHRD  37,EM0025,PBSEL0        ;TIMEOUT ERROR
4114 030222                                  ESCAPE  TST
4115
4116
4117
4118
4119
4120 030226  005037  002330          3$:     CLR     GOOD
4121 030232                                  ERRHRD  38,EM0024,PBSEL0        ;NO KMV11 ANSWER
4122 030242                                  ESCAPE  TST
4123
4124
4125
4126
4127 030246  000240                  1$:     NOP
4128 030250                          ENDTST
4129
4130
```

```
                                BADHEAD
                                ;********************* TEST17 ********************
                                ;CHECK PROM REVISION TO SEE IF COMPATIBLE WHITH DIAGNOSTIC
                                BADHEAD
                                ;********************* TEST17 ********************



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



                                BGNTST
                                        JSR     PC,CLRKMV               ;CLEAR ALL REGISTERS
                                        JSR     PC,MAINM1               ;SET MAINT MODE

                                REVPRO: JSR     R5,READ                 ;READ LOCATION 160002
                                        .WORD   160002

                                        CMP     GDREV,BAD               ;LOOK IF COMPATIBLE
                                        BEQ     1$                      ;YES

                                        ERRHRD  39,EM0034               ;REPORT THE ERROR
                                        ESCAPE TST
                                1$:
                                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
                                        JSR     PC,CLRKMV               ;CLEAR REGISTERS
                                        JSR     PC,MAINM1               ;SET MAINTENANCE MODE

                        PROMCK: JSR     R5,TSTNUB
                                .WORD   33                              ;SET TEST 33

                                WAITB   0,1

                                JSR     PC,TSTERR               ;TEST IF ERROR
                                BR      1$                      ;TEST OK
                                BR      2$                      ;TIMEOUT ERROR
                                BR      3$                      ;NO ANSWER FROM KMV11

                                MOV     @KMVP02,BAD             ;CHECKSUM ERROR
                                ERRHRD  40,EM0035,PCHECK
                                ESCAPE  TST

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

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

                        1$:     NOP
```

M8


```
4221
4222 030442                        ENDTST
4223
4224
4225
4226
4227
4228
4229
4230
```

```
4232
4233 030444                            BADHEAD
                                       ;********************* TEST19 ********************
4234
4235 030444                            BADHEAD
                                       ;********************* TEST19 ********************
4236
4237
4238
4239
4240 030444                            STARS 1
4241                                   ;SET MAINT1   ;DCT11 DECODE AND CLEAR BSEL0
4242                                   ;LOAD FIRST ADDRESS OF TX TABLE IN SEL12 , TABLE LENGTH IN SEL14,
4243                                   ;TX TABLE EXTENDED ADDRESS IN BSEL 10.
4244                                   ;
4245                                   ;SET TEST NUMBER (16 OR 17)IN BSEL0
4246                                   ;DCT11 EXECUTE THE DMA TRANSFER OF THE TABLE IN KMV11 RAM AND CHECK.
4247                                   ;WHEN DONE CLEAR BSEL0 IF TEST OK
4248                                   ;SET 200 IN BSEL0 IF TIMEOUT DURING TEST
4249                                   ;SET 100 IN BSEL0 IF ERROR DURING TRANSFER
4250                                   ;IN THAT CASE          SEL2=EXPECTED VALUE
4251                                   ;                      SEL4=READ VALUE
4252                                   ;                      SEL6=ADDRESS LOCATION OF ERROR
4253                                   ;                      BSEL10=EXTENDED ADDRESS
4254                                   ;TEST DESCRIPTION: PDP GENERATE AN INCREMENTING PATTERN TABLE OF 1K WORDS
4255                                   ;SEND STARTING ADDRESS AND TABLE LENGTH TO KMV11
4256                                   ;KMV11 START DMA TRANSFER AND CHECK
4257                                   ;
4258
4259                                   ;TEST 16 = TABLE CONTENT INCREMENTING PATTERN FROM 0
4260                                   ;TEST 17 = EACH LOCATION CONTENT ADDRESS VALUE OF LOCATION
4261 030444                            STARS 1
4262
4263
4264
4265
4266 030444                            BGNTST
4267 030444  004737  014550            DMAIN:  JSR     PC,CLRKMV               ;CLEAR REG
4268 030450  004737  014730                    JSR     PC,MAINM1               ;SET MAINT 1
4269 030454                            BGNSUB
4270 030456  012701  002426                    MOV     #TTABLE,R1              ;POINT TX TABLE
4271 030462  005002                            CLR     R2                      ;CLR TABLE
4272 030464  010221                    1$:     MOV     R2,(R1)+                ;MAKE AN INCREMENTING PATTERN FROM 0
4273 030466  005202                            INC     R2
4274 030470  022702  002000                    CMP     #2000,R2                ;TABLE LENGTH=1K WORDS
4275 030474  001373                            BNE     1$
4276
4277
4278
4279 030476  012777  002426  162016            MOV     #TTABLE,@KMVP12         ;SET TX TABLE ADDRESS
4280 030504  012777  002000  162012            MOV     #2000,@KMVP14           ;SET TABLE LENGTH
4281 030512  005077  162002                    CLR     @KMVP10                 ;CLEAR EXTENDED ADDRESS
4282 030516  004537  015010                    JSR     R5,TSTNUB               ;SEND TEST NB 16
4283 030522  000016                            .WORD   16
4284
4285
4286 030524                                    WAITB   0,1                     ;WAIT FOR TEST EXCUTION
```

```
4287
4288
4289
4290 030544  004737  013152          JSR     PC,TSTERR               ;CHECK BSELO ,WHICH ERROR
4291 030550  000444                  BR      2$                      ;TEST OK
4292 030552  000423                  BR      3$                      ;TIMEOUT ERROR
4293 030554  000432                  BR      4$                      ;NO KMV11 ANSWER
4294
4295
4296
4297
4298
4299 030556  017737  161730  002330  MOV     @KMVP02,GOOD            ;READ GOOD DATA
4300 030564  017737  161724  002424  MOV     @KMVP04,BDDAT           ;READ BAD DATA
4301 030572  017737  161720  002420  MOV     @KMVP06,ADDR            ;READ ERROR ADDRESS
4302 030600  117737  161714  012426  MOVB    @KMVP10,EXADDR          ;READ EXTENDED ADDRESS
4303 030606                          ERRHRD  43,EM0020,PRDMA         ;DATA CMP ERROR DURING DMA IN TX
4304 030616                          ESCAPE  SUB
4305
4306
4307
4308
4309
4310 030622  005037  002330  3$:     CLR     GOOD
4311 030626                          ERRHRD  44,EM0016,PBSELO        ;TIMEOUT ERROR
4312 030636                          ESCAPE  SUB
4313
4314
4315 030642  005037  002330  4$:     CLR     GOOD
4316 030646                          ERRHRD  45,EM0024,PBSELO        ;NO KMV ANSWER
4317 030656                          ESCAPE  SUB
4318
4319
4320 030662  000240          2$:     NOP
4321 030664                  ENDSUB
4322
4323
4324
4325 030666                  BGNSUB
4326 030670  004737  014730          JSR     PC,MAINM1
4327 030674  012704  002426          MOV     #TTABLE,R4              ;POINT TTABLE
4328 030700  012702  002000          MOV     #2000,R2                ;TABLE LENGTH
4329 030704  010401          1$:     MOV     R4,R1
4330 030706  010124                  MOV     R1,(R4)+        ;TABLE LOCATION CONTENT TABLE LOCATION ADDRESS
4331 030710  005302                  DEC     R2
4332 030712  001374                  BNE     1$
4333
4334
4335 030714  012777  002426  161600  MOV     #TTABLE,@KMVP12         ;SEND TABLE ADDRESS
4336 030722  012777  002000  161574  MOV     #2000,@KMVP14           ;  "      "  LENGTH
4337 030730  004537  015010          JSR     R5,TSTNUB               ;SET TEST NB 17
4338 030734  000017                  .WORD   17
4339
4340
4341 030736                          WAITB   0,1                     ;WAIT FOR TEST EXECUTION
4342
4343
```

```
4344 030756  004737  013152          JSR     PC,ISTERR               ;CHECK BSEL0
4345 030762  000444                  BR      2$                      ;TEST OK
4346 030764  000423                  BR      3$                      ;TIMEOUT ERROR
4347 030766  000432                  BR      4$                      ;NO KMV ANSWER
4348
4349
4350
4351
4352 030770  017737  161516  002330  MOV     @KMVP02,GOOD            ;READ GOOD
4353 030776  017737  161512  002424  MOV     @KMVP04,BDDAT           ;   "   BAD
4354 031004  017737  161506  002420  MOV     @KMVP06,ADDR            ;   "   ERROR ADDRESS
4355 031012  117737  161502  012426  MOVB    @KMVP10,EXADDR          ;   "   EXTENDED ADDRESS
4356
4357 031020                          ERRHRD  46,EM0020,PRDMA         ;DATA CMP ERROR
4358 031030                          ESCAPE  SUB
4359
4360
4361
4362 031034  005037  002330      3$: CLR     GOOD
4363 031040                          ERRHRD  47,EM0016,PBSEL0        ;TIMEOUT ERROR
4364 031050                          ESCAPE  SUB
4365
4366
4367
4368
4369 031054  005037  002330      4$: CLR     GOOD
4370 031060                          ERRHRD  48,EM0024,PBSEL0        ;NO KMV ANSWER
4371 031070                          ESCAPE  SUB
4372
4373
4374
4375
4376 031074  000240              2$: NOP
4377 031076                          ENDSUB
4378 031100                          ENDTST
```

```
4380 031102                        BADHEAD
                                   ;********************* TEST20 ********************
4381                               ;TEST DMA TRANSFER OUT KMV11
4382 031102                        BADHEAD
                                   ;********************* TEST20 ********************
4383
4384
4385
4386
4387 031102                        STARS 1
4388                               ;SET MAINT1    ;DCT11 DECODE AND CLEAR BSEL0
4389                               ;LOAD FIRST ADDRESS OF RX TABLE IN SEL12 AND RX TABLE LENGTH IN SEL14
4390                               ;EXTENDED ADDRESS IN BSEL10
4391
4392                               ;SET TEST NUMBER 20,21 ;DCT11 EXECUTE TEST
4393                               ;WHEN DONE CLEAR BSEL0 IF TEST OK
4394                               ;SET 200 IN BSEL0 IF TIMEOUT DURING TEST
4395                               ;
4396                               ;
4397                               ;
4398                               ;TEST 20 DESCRIPTION: DCT11 SEND IN DMA AN INCREMENTING PATTERN (OF 1K WORDS)
4399                               ;                     IN HOST MEMORY. THIS PATTERN STARS AT ADDRESS FOUND
4400                               ;                     IN SEL12 (RX TABLE)
4401                               ;                     WHEN DONE CLEAR BSEL0
4402                               ;                     HOST CHECK IF THE RECEIVE TABLE IS CORRECT
4403
4404
4405                               ;TEST 21 DESCRIPTION, IDEM BUT TABLE CONTENT ADDRESS VALUE OF EACH LOCATION
4406 031102                        STARS 1
4407
4408
4409
4410
4411 031102                        BGNTST
4412 031102                        BGNSUB
4413 031104  004737  014550       DMAOUT: JSR     PC,CLRKMV              ;CLEAR REG
4414 031110  004737  014730               JSR     PC,MAINM1              ;SET MAINT 1
4415 031114  005037  002320               CLR     FLAG
4416 031120  012701  006426               MOV     #RTABLE,R1             ;POINT RX TABLE
4417 031124  012702  002000               MOV     #2000,R2                       ;CLR RX TABLE
4418 031130  005021                1$:     CLR     (R1)+
4419 031132  005302                       DEC     R2
4420 031134  001375                       BNE     1$
4421
4422
4423
4424 031136  012777  006426  161356       MOV     #RTABLE,@KMVP12        ;SET RX TABLE ADDRESS
4425 031144  012777  002000  161352       MOV     #2000,@KMVP14          ;SET TABLE LENGTH
4426 031152  105077  161342               CLRB    @KMVP10                ;CLEAR EXTENDED ADDRESS
4427 031156  004537  015010               JSR     R5,TSTNUB              ;SEND TEST NB 20
4428 031162  000020                       .WORD   20
4429
4430
4431
4432 031164                               WAITB   0,1                    ;WAIT FOR TEST EXECUTION
4433
4434
```

```
4435 031204  004737  013152            JSR     PC,TSTERR           ;CHECK  BSELO;WHICH ERROR
4436 031210  000423                    BR      2$                  ;TEST OK
4437 031212  000402                    BR      5$                  ;TIMEOUT ERROR
4438 031214  000411                    BR      6$                  ;NO KMV ANSWER
4439 031216  000420                    BR      2$
4440
4441
4442 031220  005037  002330      5$:   CLR     GOOD
4443 031224                            ERRHRD  49,EM0016,PBSELO     ;TIMEOUT ERROR
4444 031234                            ESCAPE  SUB
4445
4446
4447
4448 031240  005037  002330      6$:   CLR     GOOD
4449 031244                            ERRHRD  50,EM0024,PBSELO     ;NO KMV ANSWER
4450 031254                            ESCAPE  SUB
4451
4452
4453
4454
4455
4456 031260  012701  006426      2$:   MOV     #RTABLE,R1          ;CHECK Rx TABLE
4457 031264  005037  002422            CLR     GDDAT               ;1ST WORD
4458
4459 031270  010137  002420      3$:   MOV     R1,ADDR
4460 031274  023711  002422            CMP     GDDAT,(R1)          ;COMPARE
4461 031300  001431                    BEQ     4$                  ;GOOD BR
4462
4463 031302  011137  002424            MOV     (R1),BDDAT
4464
4465 031306  005737  002320            TST     FLAG
4466 031312  001007                    BNE     7$                  ;LOOK IF 1ST MESAGE OR EXTENDED ONE
4467 031314                            ERRHRD  51,EM0030,PRDMA
4468
4469 031324  005237  002320            INC     FLAG
4470 031330  000415                    BR      4$
4471
4472 031332                      7$:   ERRHRD  51,0,PDMAF          ;DATA CMP ERROR
4473 031342                            BREAK
4474 031344  005237  002320            INC     FLAG
4475 031350  022737  000010  002320    CMP     #10,FLAG            ;REPORT 10 FIRST ERROR
4476 031356  001002                    BNE     4$
4477 031360                            ESCAPE  SUB
4478
4479
4480
4481
4482 031364  005237  002422      4$:   INC     GDDAT               ;CHECK NEW LOCATION
4483 031370  062701  000002            ADD     #2,R1
4484 031374  022737  002000  002422    CMP     #2000,GDDAT         ;ALL DONE
4485 031402  001332                    BNE     3$
4486 031404                      ENDSUB
4487
4488
4489
4490
4491
```

```
4492 031406                                BGNSUB
4493 031410  005037  002320                       CLR     FLAG
4494 031414  004737  014730                       JSR     PC,MAINM1
4495 031420  012777  006426  161074               MOV     #RTABLE,@KMVP12         ;LOAD RX TABLE IN SEL12
4496 031426  005077  161066                       CLR     @KMVP10
4497 031432  012777  002000  161064               MOV     #2000,@KMVP14           ;LOAD TABLE LENGTH
4498
4499
4500 031440  012702  002000                       MOV     #2000,R2                ;TABLE LENGTH
4501 031444  012701  006426                       MOV     #RTABLE,R1
4502 031450  005021                     10$:      CLR     (R1)+                   ;CLEAR RX TABLE
4503 031452  005302                               DEC     R2
4504 031454  001375                               BNE     10$
4505
4506
4507
4508
4509 031456  004537  015010                       JSR     R5,TSTNUB               ;LOAD TEST NB21
4510 031462  000021                               .WORD   21
4511
4512 031464                                       WAITB   0,1                     ;WAIT FOR TEST EXECUTION
4513
4514
4515 031504  004737  013152                       JSR     PC,TSTERR               ;CHECK BSELO;WHICH ERROR
4516 031510  000423                               BR      2$                      ;TEST OK
4517 031512  000402                               BR      5$                      ;TIMEOUT ERROR
4518 031514  000411                               BR      6$                      ;NO ANSWER
4519 031516  000420                               BR      2$                      ;DATA CMP ERROR
4520
4521
4522
4523 031520  005037  002330             5$:       CLR     GOOD
4524 031524                                       ERRHRD  52,EM0016,PBSELO        ;TIMEOUT ERROR
4525 031534                                       ESCAPE  SUB
4526
4527
4528
4529 031540  005037  002330             6$:       CLR     GOOD
4530 031544                                       ERRHRD  53,EM0024,PBSELO        ;NO KMV11 ANSWER
4531 031554                                       ESCAPE  SUB
4532
4533
4534
4535
4536
4537
4538 031560  012702  002000             2$:       MOV     #2000,R2
4539 031564  012737  006426  002420               MOV     #RTABLE,ADDR            ;VERIFY RX TABLE
4540 031572  012737  006426  002422               MOV     #RTABLE,GDDAT
4541
4542 031600  023737  002422  002420     3$:       CMP     GDDAT,ADDR              ;CMP TABLE
4543 031606  001432                               BEQ     4$
4544 031610  017737  150604  002424               MOV     @ADDR,BDDAT             ;READ BAD DATA
4545
4546
4547 031616  005737  002320                       TST     FLAG
4548 031622  001007                               BNE     1$                      ;LOOK IF 1ST REPORT
```

```
4549
4550
4551 031624                                    ERRHRD  54,EM0030,PRDMA         ;DATA CMP ERROR IN R4 TABLE
4552 031634  005237  002320                    INC     FLAG
4553 031640  000415                            BR      4$
4554
4555 031642                            1$:     ERRHRD  54,0,PDMAF              ;SHORT ERROR REPORT
4556 031652                                    BREAK
4557 031654  005237  002320                    INC     FLAG
4558 031660  022737  000010  002320            CMP     #10,FLAG                ;REPORT 10 ERROR
4559 031666  001002                            BNE     4$
4560 031670                                    ESCAPE  SUB
4561
4562
4563 031674  062737  000002  002422  4$:     ADD     #2,GDDAT                ;VERIFY NEXT LOCATION
4564 031702  062737  000002  002420            ADD     #2,ADDR
4565 031710  005302                            DEC     R2                      ;ALL DONE:
4566 031712  001332                            BNE     3$                      ;NO BRANCH
4567
4568 031714                            ENDSUB
4569 031716                            ENDTST
4570
```

```
4572 031720                        BADHEAD
                                   ;********************* TEST21 *******************
4573                               ;TEST DMA TRANSFER IN BOTH DIRECTION
4574 031720                        BADHEAD
                                   ;********************* TEST21 *******************
4575
4576
4577
4578
4579
4580 031720                        STARS 1
4581                               ;SET MAINT1    ;  DCT11 DECODE AND CLEAR BSEL0
4582                               ;THE HOST SET ALL THE PARAMETERS IN CSR'S
4583                               ;LOAD TX TABLE ADDRESS IN SEL12,TABLE LENGTH IN SEL14,EXTENDED ADDRESS IN BSEL10
4584                               ;EXTENDED ADDRESS OF RX TABLE IN BSEL2 , ADDRESS OF RX TABLE IN SEL4 AND
4585                               ;RAM STARTING ADDRESS FOR TRANSFER IN SEL6.
4586                               ;
4587                               ;
4588                               ;LOAD TEST NUMBER 22  ;  DCT11 EXECUTE TEST
4589                               ;  WHEN DONE CLEAR BSEL0 IF TEST OK OR SET 200 IN BSEL0 IF TIMEOUT DURING DMA.
4590                               ;
4591                               ;TEST DESCRIPTION:
4592                               ; HOST COMPUTER GENERATES DIFFERENT 1K WORD TABLES ,GIVES ALL PARAMETERS IN
4593                               ; THE CSR'S AND SET TEST 22 IN BSEL0
4594                               ; DCT11 TAKES SEL6 AS THE STARTING ADDRESS FOR THE DIFFERENT TRANSFERS IN KMV11
4595                               ; RAM MEMORY (DMA INTO KMV11) AND TRANSFER THIS TABLE IN DMA BACK TO HOST
4596                               ; MEMORY (DMA OUT).
4597                               ;
4598                               ; DATA TRANSFER ARE MADE IN DIFFERENT AREAS IN RAM AND  DCT11 CHECKS
4599                               ; THAT THE UNUSED PART OF THE RAM IS NOT MODIFIED
4600                               ;
4601                               ; WHEN TRANSFER IN BOTH DIRECTION HAS BEEN DONE ,DCT11 CLEAR BSEL0 AND
4602                               ; HOST COMPARES RX AND TX TABLE
4603                               ;
4604                               ;
4605                               ;ERROR REPORT IN BSEL0:          200=TIMEOUT DURING DMA
4606                               ;                                100=UNUSED MEMORY MODIFIED DURING TRANSFER
4607                               ;                                        IN THAT CASE SEL2 =GOOD
4608                               ;                                                     SEL4 =BAD
4609                               ;                                                     SEL6 = ADDRESS
4610                               ;
4611                               ;
4612                               ;
4613 031720                        STARS 1
4614
4615
4616
4617
4618 031720                        BGNTST
4619 031720  012737  065000  012462          MOV     #65000,MAXCNT          ;RAM MEMORY MAX LENGTH
4620 031726  005037  002320                  CLR     FLAG
4621
4622 031732  012703  000002          DMATWO: MOV     #2,R3                  ;SELECT 1ST PATTERN
4623 031736  004737  014550                  JSR     PC,CLRKMV              ;CLEAR REG
4624 031742  005037  002414                  CLR     COUNT                  ;SELECT RAM STARTING ADDRESS FOR TX
4625 031746                                  BREAK
4626
```

```
4627 031750  004737  014730          TWODMA: JSR     PC,MAINM1               ;SET MAINT 1
4628
4629
4630
4631 031754  012702  002000                  MOV     #2000,R2
4632 031760  012701  006426                  MOV     #RTABLE,R1
4633 031764  005021                  10$:    CLR     (R1)+                   ;CLEAR RX TABLE
4634 031766  005302                          DEC     R2
4635 031770  001375                          BNE     10$
4636
4637
4638
4639
4640
4641
4642 031772  012702  002000                  MOV     #2000,R2
4643 031776  012701  002426                  MOV     #TTABLE,R1
4644 032002  004737  013230          1$:     JSR     PC,GENER                ;MAKE A PATTERN
4645
4646 032006  013721  012436                  MOV     DATA,(R1)+              ;WRITE ONE TABLE LOCATION
4647 032012  005302                          DEC     R2                      ;ALL LOCATION DONE?
4648 032014  001372                          BNE     1$                      ;NO
4649
4650 032016  005077  160476                  CLR     @#KMVP10                ;CLEAR EXTENDED ADDRESS
4651 032022  013777  002414  160476          MOV     COUNT,@#KMVP16          ;LOAD STATING ADDRESS IN RAM
4652
4653 032030  012777  002426  160464          MOV     #TTABLE,@#KMVP12        ;SEND TX TABLE ADDRESS
4654 032036  012777  002000  160460          MOV     #2000,@#KMVP14          ;SEND TABLE LENGTH
4655 032044  012777  006426  160442          MOV     #RTABLE,@#KMVP04        ;SEND RX TABLE IN SEL4
4656 032052  005077  160434                  CLR     @#KMVP02                ;CLR RX TABLE EXT ADDRESS
4657 032056  004537  015010                  JSR     R5,TSTNUB               ;LOAD TEST NB 22
4658 032062  000022                          .WORD   22
4659
4660 032064  012737  070000  002324          MOV     #70000,DELCT1           ; SET DELAY COUNTER
4661 032072  117700  160412          11$:    MOVB    @#KMVCSR,R0             ; GET BSELO
4662 032076  105700                          TSTB    R0                      ; SEE IF TEST DONE OR CSR/DMA INTERFERENCE
4663 032100  001452                          BEQ     3$                      ; CHECK XMT/RCV BUFFER
4664 032102  005237  002324                  INC     DELCT1                  ; UPDATE TIMEOUT COUNTER
4665 032106  001371                          BNE     11$                     ; BR IF NOT TIMED OUT
4666 032110  000407                          BR      6$                      ; TIME-OUT
4667
4668 032112                                  BREAK
4669
4670
4671 032114  004737  013152                  JSR     PC,TSTERR               ;CHECK BSELO;WHICH ERROR
4672 032120  000442                          BR      3$                      ;TEST OK
4673 032122  000402                          BR      6$                      ;TIME OUT
4674 032124  000411                          BR      7$                      ;NO KMV11 ANSWER
4675 032126  000420                          BR      20$                     ;PROBLEM IN THE UNUSED PART OF RAM:
4676                                                                         ;DMA TRANSFER MODIFY UNUSED RAM
4677                                                                         ; LOCATIONS.
4678
4679
4680
4681 032130  005037  002330          6$:     CLR     GOOD
4682 032134                                  ERRHRD  55,EM0016,PBSELO        ;TIMEOUT ERROR
4683 032144                                  ESCAPE  TST
```

```
4684
4685
4686
4687 032150  005037  002330          7$:     CLR     GOOD
4688 032154                                  ERRHRD  56,EM0024,PBSEL 0       ;NO KMV11 ANSWER
4689 032164                                  ESCAPE  TST
4690
4691
4692
4693 032170  017737  160322  002420  20$:    MOV     @#KMVP06,ADDR           ;READ ADD IN ERROR
4694 032176  017737  160310  002330          MOV     @#KMVP02,GOOD           ;GOOD VALUE
4695 032204  017737  160304  002424          MOV     @#KMVP04,BDDAT          ;READ WRONG VALUE
4696 032212                                  ERRHRD  57,EM0033,PDMARA        ;DATA  ERROR IN RAM DURING TRANSFER
4697 032222                                  ESCAPE  TST
4698
4699
4700
4701
4702 032226  005077  160256          3$:     CLR     @#KMVCSR
4703 032232  000240                          NOP
4704 032234  012777  044000  160246          MOV     #MAINT1,@#KMVCSR        ;STOP TEST 2 IN KMV
4705 032242  012701  002426                  MOV     #TTABLE,R1              ;TX TABLE ADDRESS
4706 032246  012704  006426                  MOV     #RTABLE,R4              ;RX     "       "
4707 032252  012702  002000                  MOV     #2000,R2                ;TABLE LENGHT
4708
4709
4710
4711 032256  021114                  4$:     CMP     (R1),(R4)               ;CMP RX TABLE AND TX TABLE
4712 032260  001437                          BEQ     5$                      ;OK TEST NEXT LOCATION
4713
4714
4715 0322c   011137  002422                  MOV     (R1),GDDAT              ;PREPARE ERROR REPORT
4716 032266  011437  002424                  MOV     (R4),BDDAT
4717 032272  010437  002420                  MOV     R4,ADDR
4718 032276  005037  012426                  CLR     EXADDR
4719
4720 032302  005737  002320                  TST     FLAG
4721 032306  001007                          BNE     2$
4722 032310                                  ERRHRD  58,EM0021,PRDMA         ;DATA CMP ERROR IN TABLE
4723 032320  005237  002320                  INC     FLAG
4724 032324  000415                          BR      5$
4725
4726
4727 032326                          2$:     ERRHRD  58,0,PDMAF              ;REPORT 10 FIRST ERROR
4728 032336                                  BREAK
4729 032340  005237  002320                  INC     FLAG
4730 032344  022737  000010  002320          CMP     #10,FLAG
4731 032352  001002                          BNE     5$
4732 032354                                  ESCAPE  TST
4733
4734
4735
4736
4737 032360  005721                  5$:     TST     (R1)+
4738 032362  005724                          TST     (R4)+
4739 032364  005302                          DEC     R2                      ;ALL MEMORY TESTED?
4740 032366  001333                          BNE     4$                      ;NO BRANCH
```

```
4741
4742
4743 032370  062737  002000  002414          ADD     #2000,COUNT             ;USE OTHER PART OF RAM
4744 032376  023737  002414  012462          CMP     COUNT,MAXCNT            ;IS ALL RAM USED?
4745 032404  100002                          BPL     30$
4746 032406  000137  031750                  JMP     TWODMA
4747
4748
4749
4750 032412  005203                  30$:    INC     R3                      ;SELECT NEW KIND OF PATTERN
4751 032414  022703  000005                  CMP     #5,R3                   ;ALL DONE?
4752 032420  001402                          BEQ     40$                     ;NO BRANCH
4753 032422  000137  031750                  JMP     TWODMA
4754 032426                          40$:
4755 032426                          ENDTST
4756
```

```
4758 032430                         BADHEAD
                                    ;********************* TEST22 ********************
4759                                ;TEST INTERUPT CAPABILITY OF KMV11 MODULE ON QBUS
4760 032430                         BADHEAD
                                    ;********************* TEST22 ********************
4761
4762
4763
4764 032430                         STARS 1
4765                                ;SET MAINT1   ; KMV11 DECODE AND CLEAR BSELO
4766                                ;HOST PREPARES VECTOR AREA
4767                                ;SEND TEST NUMBER (23 OR 24)
4768                                ;DCT11 INTERUPTS THE HOST BY SETTING BITS 5 OR 6 IN ADDRESS 140000 OF
4769                                ;KMV11 MICRO BUS ;DCT11 CLEAR BSELO WHEN TEST COMPLETED.
4770                                ;
4771                                ;HOST TESTS IF THE INTERUPT HAS BEEN RECEIVED WITH CORRECT VECTOR
4772                                ;
4773                                ;
4774                                ;
4775                                ;
4776                                ;MICRO TEST 23 =INTERUPT ON LOW VECTOR
4777                                ;MICRO TEST 24 =INTERUPT ON HIGH VECTOR
4778 032430                         STARS 1
4779
4780
4781
4782
4783
4784
4785
4786 032430                         BGNTST
4787 032430  004737  014550                 JSR     PC,CLRKMV               ;CLR REG
4788 032434  004737  014730                 JSR     PC,MAINM1
4789 032440                         BGNSUB
4790 032442  005037  012430                 CLR     INTFLG
4791 032446  013702  012476                 MOV     KMVLVL,R2               ;READ KMV PRIORITY
4792 032452  012777  032536  160014         MOV     #INT1,@KMVV00           ;SET UP VECTOR 0
4793 032460  006202                        ASR     R2
4794 032462  006202                        ASR     R2
4795 032464  006202                        ASR     R2
4796 032466  006202                        ASR     R2
4797                                ;       MOV     #340,@KMVV02            ;SET KMV PRIORITY 7 FOR INTERUPT        ;JB REV A-0
4798 032470  012777  000300  160004         MOV     #300,@KMVV02            ;SET KMV PRIORITY 6 FOR INTERUPT        ;JB REV A-0
4799
4800
4801
4802                                ;       MOV     #340,R3                 ;TRY PRIORITY 7 FOR PROCESSOR           ;JB REV A-0
4803 032476  012703  000300                 MOV     #300,R3                 ;TRY PRIORITY 6 FOR PROCESSOR           ;JB REV A-0
4804
4805
4806 032502  106403                 SETPR1: MTPS    R3                      ;LOAD PRIORITY
4807 032504  004537  015010                 JSR     R5,TSTNUB               ;SEND TEST 23
4808 032510  000023                         .WORD   23
4809
4810 032512  000240                         NOP
4811 032514  000240                         NOP
4812 032516  000240                         NOP
```

```
4813 032520  000240                        NOP
4814 032522  000240                        NOP
4815 032524  000240                        NOP
4816 032526  000240                        NOP
4817
4818 032530                                BREAK
4819 032532  000137  032546                JMP     VECTO
4820
4821
4822                          ;***************INTERUPT  ROUTINE  ******************************
4823
4824 032536  052737  000001  012430  INT1:  BIS     #1,INTFLG              ;SET INT FLAG
4825 032544  000002                        RTI
4826
4827
4828                          ;*************************************************************************
4829
4830
4831
4832 032546  004537  013722          VECTO: JSR     R5,CBSELO              ;CHECK IF KMV11 ANSWER
4833 032552  000000                        .WORD   0
4834 032554  000410                        BR      2$
4835 032556                                ERRHRD  59,EM0024              ;NO KMV11 ANSWER
4836 032566  004737  013010                JSR     PC,CHKMAX              ;CHECK IF TOO MANY ERROR
4837 032572                                ESCAPE  SUB
4838
4839
4840 032576  005737  012430          2$:   TST     INTFLG                 ;TEST IF INTERUPT ?
4841 032602  001454                        BEQ     3$
4842
4843 032604  010237  002330                MOV     R2,GOOD                ;GOOD INTERUPT LEVEL
4844
4845
4846
4847 032610  062703  000040                ADD     #40,R3                 ;WAS IT LEGAL
4848 032614  010337  012432                MOV     R3,BAD
4849 032620  023737  012432  002330        CMP     BAD,GOOD
4850 032626  001461                        BEQ     4$                     ;YES BRANCH
4851
4852 032630  106237  002330                ASRB    GOOD
4853 032634  106237  002330                ASRB    GOOD
4854 032640  106237  002330                ASRB    GOOD
4855 032644  106237  002330                ASRB    GOOD
4856 032650  106237  002330                ASRB    GOOD
4857 032654  042737  177770  002330        BIC     #177770,GOOD
4858
4859
4860 032662  106237  012432                ASRB    BAD
4861 032666  106237  012432                ASRB    BAD
4862 032672  106237  012432                ASRB    BAD
4863 032676  106237  012432                ASRB    BAD
4864 032702  106237  012432                ASRB    BAD
4865 032706  042737  177770  012432        BIC     #177770,BAD
4866 032714                                ERRHRD  60,EM0022
4867 032724  004737  013010                JSR     PC,CHKMAX
4868 032730                                ESCAPE  SUB
4869
```

```
032734  122703  000140          3$:     CMPB    #140,R3                 ;IS PROCESSOR AT LEVEL 3
032740  001404                          BEQ     5$
032742  162703  000040                  SUB     #40,R3                  ;DECREASE PRIORITY
032746  000137  032502                  JMP     SETPR1                  ;TRY WITH NEW ONE
032752                          5$:     ERRHRD  61,EM0023               ;NO INTERUPT OCCUR
032762  004737  013010                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
032766                                  ESCAPE  SUB
032772                          4$:
032772                          ENDSUB
032774                          BGNSUB
032776  004737  014730                  JSR     PC,MAINM1
033002  005037  012430                  CLR     INTFLG
033006  013702  012476                  MOV     KMVLVL,R2               ;SET PRIORITY LEVEL
                                ;       MOV     #340,@KMVV06                            ;JB REV A-0
033012  012777  000300  157464          MOV     #300,@KMVV06                            ;JB REV A-0
033020  012777  033072  157452          MOV     #INT2,@KMVV04           ;SET UP VECTOR 4
033026  006202                          ASR     R2
033030  006202                          ASR     R2
033032  006202                          ASR     R2
033034  006202                          ASR     R2
                                ;       MOV     #340,R3                 ;START WITH PRIORITY 7 FOR PROCESSOR    ;JB REV A-0
033036  012703  000300                  MOV     #300,R3                 ;START WITH PRIORITY 6 FOR PROCESSOR    ;JB REV A-0
033042  106403                  INTPR2: MTPS    R3                      ;LOAD PRIORITY
033044  004537  015010                  JSR     R5,TSTNUB
033050  000024                          .WORD   24                      ;SET TEST NB 24
033052  000240                          NOP
033054  000240                          NOP
033056  000240                          NOP
033060  000240                          NOP
033062  000240                          NOP
033064                                  BREAK
033066  000137  033102                  JMP     VECT4
                                ;****************INTERUPT ROUTINE **********************************
033072  052737  000001  012430  INT2:   BIS     #1,INTFLG               ;SET FLAG
033100  000002                          RTI
                                ;*******************************************************************
```

```
4927
4928 033102  004537  013722          VECT4:  JSR     R5,CBSEL0               ;IS THERE KMV11 ANSWER ?
4929 033106  000000                          .WORD   0
4930 033110  000410                          BR      2$
4931 033112                                  ERRHRD  62,EM0024               ;NO KMV11 ANSWER
4932 033122  004737  013010                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
4933 033126                                  ESCAPE  SUB
4934
4935
4936
4937 033132  032737  000001  012430  2$:     BIT     #1,INTFLG               ;TEST IF INTERUPT OCCUR
4938 033140  001454                          BEQ     3$                      ;NO INTERUPT
4939
4940 033142  010237  002330                  MOV     R2,GOOD                 ;GOOD INTERUPT LEVEL
4941
4942
4943 033146  062703  000040                  ADD     #40,R3                  ;WAS IT LEGAL ?
4944 033152  010337  012432                  MOV     R3,BAD
4945
4946 033156  023737  012432  002330          CMP     BAD,GOOD
4947 033164  001461                          BEQ     4$                      ;YES BRANCH
4948 033166  106237  002330                  ASRB    GOOD
4949 033172  106237  002330                  ASRB    GOOD
4950 033176  106237  002330                  ASRB    GOOD
4951 033202  106237  002330                  ASRB    GOOD
4952 033206  106237  002330                  ASRB    GOOD
4953 033212  042737  177770  002330          BIC     #177770,GOOD            ;GET ACTUAL LEVEL
4954
4955 033220  106237  012432                  ASRB    BAD
4956 033224  106237  012432                  ASRB    BAD
4957 033230  106237  012432                  ASRB    BAD
4958 033234  106237  012432                  ASRB    BAD
4959 033240  106237  012432                  ASRB    BAD
4960 033244  042737  177770  012432          BIC     #177770,BAD
4961 033252                                  ERRHRD  63,EM0022               ;INT OCCUR AT BAD LEVEL
4962 033262  004737  013010                  JSR     PC,CHKMAX
4963 033266                                  ESCAPE  SUB
4964
4965
4966
4967
4968 033272  122703  000040          3$:     CMPB    #40,R3                  ;IS PROCESSOR AT PRIORITY 3
4969 033276  001404                          BEQ     5$                      ;YES,NO INTERUPT OCCURED
4970 033300  162703  000040                  SUB     #40,R3                  ;DECRESE PRIORITY LEVEL
4971 033304  000137  033042                  JMP     INTPR2
4972
4973
4974 033310                          5$:     ERRHRD  64,EM0023
4975 033320  004737  013010                  JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
4976 033324                                  ESCAPE  SUB
4977 033330                          4$:
4978 033330                          ENDSUB
4979 033332                          ENDTST
```

```
                        BADHEAD
                        ;********************* TEST23 ******************
                        ;TEST INTERUPT ON DCT11 MICROPROCESSOR
                        BADHEAD
                        ;********************* 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
```

```
5028
5029
5030
5031
5032
5033
5034
5035
5036 033334                    BGNTST
5037
5038
5039 033334                    BGNSUB
5040 033356 004737 014550      JSR     PC,CLRKMV       ;CLEAR REG
5041 033342 004737 014730      JSR     PC,MAINM1       ;SET MAINT MODE
5042 033346 004537 015010      JSR     R5,TSTNUB       ;SET TEST NB 25
5043 033352 000025             ,WORD   25
5044
5045
5046 033354                    WAITB   0,1
5047
5048
5049
5050 033374 013701 012512      MOV     KMVP02,R1       ;LOAD CSR ADDR
5051 033400 012702 000012      MOV     #12,R2          ;ACCES BSELS TO BSEL16
5052
5053 033404 152721 000207      1$:     BISB    #207,(R1)+      ;WRITE ALL REG BUT BSELO
5054
5055 033410                    WAITB   0,1             ;WAIT FOR TEST EXECUTION
5056
5057 033430 004537 013722      JSR     R5,CBSELO       ;LOOK IF INTERUPT OCCUR
5058 033434 000000             ,WORD   0
5059
5060 033436 000404             BR      3$              ;YES SEE WHICH ERROR
5061 033440 005302             DEC     R2              ;ALL REG DONE ?
5062 033442 001360             BNE     1$              ;NO BR
5063
5064
5065
5066
5067 033444 000137 033476      JMP     GOON1           ;OK NO ACCESS INTERUPT THE DCT11 ;GO ON
5068
5069
5070
5071 033450 010137 002420      3$:     MOV     R1,ADDR         ;SEE WHICH ADDRESS CAUSE INTERUPT
5072 033454 162737 000001 002420       SUB     #1,ADDR
5073 033462                    ERRHRD  65,EM0026,PINTR ;WRONG INTERUPT OCCURED ON DCT11
5074                                                   ;WHILE ADDRESSING KMV11 REGISTERS
5075 033472                    ESCAPE  SUB
5076
5077
5078
5079 033476 052777 004025 157004 GOON1: BIS     #4025,@KMVCSR   ;ACCESS BSELO
5080
5081 033504                    WAITB   0,1
5082
5083 033524 004537 013722      JSR     R5,CBSELO
5084 033530 000000             ,WORD   0
```

```
5085 033532  000424                          BR      5$              ;TEST OK ,INTRUPT OCCURED AT GOOD VECTOR
5086
5087 033534  004537  013722                  JSR     R5,CBSEL0
5088 033540  000100                          .WORD   100
5089 033542  000410                          BR      6$              ;INT ON ILLEGAL VECTOR
5090 033544                                  ERRHRD  66,EM0027       ;NO KMV11 ANSWER, DCT11 DOES NOT RECEIVE ANY
5091 033554  004737  013010                  JSR     PC,CHKMAX       ;CHECK IF TOO MANY ERROR
5092 033560                                  ESCAPE  SUB             ;INTERUPT WHEN QBUS ADDRESS CSR'S
5093
5094
5095
5096
5097 033564                          6$:     ERRHRD  67,EM0032       ;INT ON ILLEGAL VECTOR WHEN ACCESSING BSEL0
5098 033574  004737  013010                  JSR     PC,CHKMAX       ;CHECK IF TOO MANY ERROR
5099 033600                                  ESCAPE  SUB
5100
5101 033604  000240                  5$:     NOP
5102 033606                          ENDSUB
5103
5104
5105
5106
5107
5108 033610                          BGNSUB
5109 033612  004737  014730                  JSR     PC,MAINM1       ;SET MAINT MODE
5110 033616  004537  015010                  JSR     R5,TSTNUB       ;SET TEST NB 26
5111 033622  000026                          .WORD   26
5112
5113
5114 033624                                  WAITB   0,1
5115
5116 033644  052777  000026  156636          BIS     #26,@KMVCSR     ;WRITE SEL0
5117
5118 033652  013701  012514                  MOV     KMVP04,R1       ;LOAD CSR ADDR
5119 033656  012702  000010                  MOV     #10,R2          ;ACCES BSEL3 TO BSEL11
5120
5121 033662  152721  000207          1$:     BISB    #207,(R1)+      ;WRITE ALL REG BUT BSEL2
5122
5123 033666                                  WAITB   0,1             ;WAIT FOR TEST EXECUTION
5124
5125 033706  004537  013722                  JSR     R5,CBSEL0       ;LOOK IF INTERUPT OCCUR
5126 033712  000000                          .WORD   0
5127
5128 033714  000404                          BR      3$              ;YES SEE WHICH ERROR
5129 033716  005302                          DEC     R2              ;ALL REG DONE ?
5130 033720  001360                          BNE     1$              ;NO BR
5131
5132
5133
5134
5135
5136 033722  000137  033760                  JMP     GOON2           ;OK NO ACCESS INTERUPT THE DCT11 ;GO ON
5137
5138
5139 033726  017737  156562  012440  3$:     MOV     @KMVP04,VECT    ;READ RECEIVE VECTOR
5140 033734  010137  002420                  MOV     R1,ADDR         ;SEE WHICH ADDRESS MAKE INTERUPT
5141 033740  005337  002420                  DEC     ADDR
```

```
5142 033744                                ERRHRD  68,EM0026,PINTR ;WRONG INTERUPT OCCUR WHILE ACCESSING REGISTERS
5143 033754                                ESCAPE  SUB
5144
5145
5146 033760  000240              GOON2:    NOP
5147
5148 033762  052777  017777  156522       BIS     #17777,@KMVP02  ;ACCESS BSEL2
5149
5150 033770                                WAITB   0,1
5151
5152 034010  004537  013722               JSR     R5,CBSEL0
5153 034014  000000                       .WORD   0
5154 034016  000424                       BR      5$              ;TEST OK ,INTRUPT OCCUR AT GOOD VECTOR
5155
5156 034020  004537  013722               JSR     R5,CBSEL0
5157 034024  000100                       .WORD   100
5158 034026  000410                       BR      6$              ;INT ON ILLEGAL VECTOR
5159 034030                                ERRHRD  69,EM0027       ;NO KMV11 ANSWER
5160 034040  004737  013010               JSR     PC,CHKMAX       ;CHECK IF TOO MANY ERROR
5161 034044                                ESCAPE  SUB
5162
5163
5164
5165
5166
5167 034050                        6$:     ERRHRD  70,EM0026               ;INT ON ILLEGAL VECTOR
5168 034060  004737  013010               JSR     PC,CHKMAX               ;CHECK IF TOO MANY ERROR
5169 034064                                ESCAPE  SUB
5170
5171 034070  000240                5$:     NOP
5172 034072                        ENDSUB
5173
5174
5175
5176 034074                        ENDTST
5177
```

```
5179
5180                                    .SBTTL HARDWARE PARAMETER CODING SECTION
5181
5182
5183
5184                                    ;////////////////////////////////////////////////////////////////////////////
5185                                    ;/ THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
5186                                    ;/ THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
5187                                    ;/ MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
5188                                    ;/ INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
5189                                    ;/ MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
5190                                    ;/ WITH THE OPERATOR.
5191                                    ;////////////////////////////////////////////////////////////////////////////
5192
5193 034076                                   BGNHRD
5194
5195 034100                                   GPRMA   ADDRES,0,0,60000,177776,YES
5196 034110                                   GPRMA   VECTOR,2,0,0,674,YES
5197 034120                                   GPRMD   PRIRTY,4,0,7000,4,7,YES
5198 034132                                   ENDHRD
5199
5206
5207
5208 034132     115     111     103  ADDRES: .ASCIZ  /MICRO-CPU  CSR ADDRESS : /
      034135     122     117     055
      034140     103     120     125
      034143     040     040     103
      034146     123     122     040
      034151     101     104     104
      034154     122     105     123
      034157     123     040     072
      034162     040     000
5209 034164     115     111     103  VECTOR: .ASCIZ  /MICRO-CPU VECTOR ADDRESS : /
      034167     122     117     055
      034172     103     120     125
      034175     040     126     105
      034200     103     124     117
      034203     122     040     101
      034206     104     104     122
      034211     105     123     123
      034214     040     072     040
      034217     000
5210 034220     115     111     103  PRIRTY: .ASCIZ  /MICRO-CPU PRIORITY LEVEL : /
      034223     122     117     055
      034226     103     120     125
      034231     040     120     122
      034234     111     117     122
      034237     111     124     131
      034242     040     114     105
      034245     126     105     114
      034250     040     072     040
      034253     000
5211
5212                                        .EVEN
5213
```

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

034254                                  BGNSFT



034256                                  ENDSFT
```

```
5251
5252 034256                          $PATCH::
5253 034256                                  .BLKW   50
5254
5261
5262 034376                                  LASTAD
     034402                          L$LAST::
5263 034402                                  ENDMOD
5264
5265
```

SOFTWARE PARAMETER CODING SECTION

```
5267
5268
5281
5282 034402                                  BGNSETUP        1
5283 034402                                  BGNPTAB
5284 034406  177000                          .WORD   177000
5285 034410  000300                          .WORD   300
5286 034412  004000                          .WORD   4000
5287 034414  000001                          .WORD   1
5288 034416                                  ENDPTAB
5289 034416                                  ENDSETUP
5290
5291
5292
5293
5294
5295         000001                  .END
```

SYMBOL TABLE

```
ABORT   024036          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  034132          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$ESEG= 000005          EM0012  016246          GENEX   013510          INTPR2  033042
BIT04   = 000020 G      C$ESUB= 000003          EM0013  016342          GENE1   025504          INT1    032536
BIT05   = 000040 G      C$ETST= 000001          EM0015  016436          GENE2   025702          INT2    033072
BIT06   = 000100 G      C$EXIT= 000032          EM0016  016506          GENE3   026100          ISR   = 000100 G
BIT07   = 000200 G      C$GETB= 000026          EM0020  016544          GENE4   026310          IXE   = 004000 G
BIT08   = 000400 G      C$GETW= 000027          EM0021  016630          GENE5   026512          I$AU   = 000041
BIT09   = 001000 G      C$GMAN= 000043          EM0022  016722          GENE6   026714          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$PUR  = 000041
BIT4    = 000020 G      C$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  023634          I$SETU= 000041
BIT7    = 000200 G      C$PNTX= 000015          EM0134  017746          GOOD    002330          I$SFT  = 000041
BIT8    = 000400 G      C$QIO  = 000377         END     024102          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   024652          C$SEFG= 000046          EXADDR  012426          GOOD4   002340          KMVLVL  012476
CHECK1  025016          C$SPRI= 000041          E$END  = 002100         GOOD6   002342          KMVP02  012512
CHKMAX  013010          C$SVEC= 000037          E$LOAD= 000035          GOON1   033476          KMVP04  012514
CKALL   013760          C$TPRI= 000013          FLAG    002320          GOON2   033760          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   030444          F$HW   = 000013         G$OFSI= 000376          KMV11A  002000 G
C$CEFG= 000045          DMAOUT  031104          F$INIT= 000006          G$PRMA= 000001          LENGTH  012454
C$CLCK= 000062          DMATWO  031736          F$JMP  = 000050         G$PRMD= 000002          LOCK    002274
C$CLEA= 000012          DROPD   024230          F$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  031714      O$SETU= 000001       SAVSP   002306
L$ACP    002110 G      L10003  022144      L10074  032426      PADFLT  022346 G     SAVSTA  002412
L$APT    002036 G      L10004  022206      L10075  033332      PBSEL0  023174 G     SEL0    002354
L$AU     024262 G      L10005  022312      L10076  032772      PCHECK  022314 G     SEL1    002356
L$AUT    002070 G      L10006  022344      L10077  033330      PDMAF   023354 G     SEL10   002366
L$AUTO   024104 G      L10007  022402      L10100  034074      PDMARA  022210 G     SEL12   002370
L$CCP    002106 G      L10010  022712      L10101  033606      PINTR   023232 G     SEL14   002372
L$CLEA   024176 G      L10011  023172      L10102  034072      PNT    = 001000 G    SEL16   002374
L$CO     002032 G      L10012  023230      L10103  034132      PRALL   022404 G     SEL2    002360
L$DEPO   002011 G      L10013  023262      L10104  034256      PRBYTE  022146 G     SEL4    002362
L$DESC   002222 G      L10014  023352      L10105  034406      PRDMA   023264 G     SEL6    002364
L$DESP   002076 G      L10015  023410      L10107  034416      PRI    = 002000 G    SETPRI  032502
L$DEVP   002060 G      L10016  023416      MAINM0  014660      PRIRTY  034220       SETUP   023552
L$DISP   002132 G      L10017  024102      MAINM1  014730      PRIO0  = 000000 G    SSTACK  012732
L$DLY    002116 G      L10020  024174      MAINT0= 054000 G     PRIO1  = 000040 G    SVCGBL= 000000
L$DTP    002040 G      L10021  024200      MAINT1= 044000 G     PRIO2  = 000100 G    SVCINS= 177777
L$DTYP   002034 G      L10022  024260      MAXCNT  012462       PRIO3  = 000140 G    SVCSUB= 177777
L$DU     024202 G      L10023  024262      MAXERR  002276       PRIO4  = 000200 G    SVCTAG= 177777
L$DUT    002072 G      L10024  024400      MAXPRI= 000300 G     PRIO5  = 000240 G    SVCTST= 177777
L$DVTY   012732 G      L10025  024574      MBSEL0  021103       PRIO6  = 000300 G    S$LSYM= 010000
L$EF     002052 G      L10026  024516      MBYTE   021371       PRIO7  = 000340 G    TBYTE   025104
L$ENVI   002044 G      L10027  024572      MCHECK  021666       PROMCK  030326       TCSR    024770
L$ERRT   002256 G      L10030  024734      MCLR   = 040000 G    PRRAM   022060 G     TCSRMB  024760
L$ETP    002102 G      L10031  025076      MDMAF   021737       PRREG   022714 G     TFM36   015164
L$EXP1   002046 G      L10032  025230      MDMAR1  021453       PRSEL0  022022 G     TGENE1  025474
L$EXP4   002064 G      L10033  025456      MDMAR2  021542       PSTACK  002304       TGENE2  025672
L$EXP5   002066 G      L10034  025344      MDMAR3  021610       QV.FLG  012471       TGENE3  026070
L$HARD   034100 G      L10035  025454      MDMA1   021240       RAMADD  027726       TGENE4  026300
L$HIME   002120 G      L10036  025652      MDMA2   021323       RAMCAD  030100       TGENE5  026502
L$HPCP   002016 G      L10037  025546      MINT    021051       RAMPAT  027506       TGENE6  026704
L$HPTP   002022 G      L10040  025650      MINTR   021145       RANCLC  013462       TIM     015247
L$HW     002212 G      L10041  026050      MRAM1   020707       RANDN   002406       TSELA   024630
L$ICP    002104 G      L10042  025744      MRAM2   020771       RANEX   013500       TSELB   024634
L$INIT   023420 G      L10043  026046      MREG0   020107       RANGEN  013402       TSEL10  026200
L$LADP   002026 G      L10044  026260      MREG10  020323       RANMTA  002404       TSEL12  026402
L$LAST   034402 G      L10045  026154      MREG12  020366       RANSEC  013466       TSEL14  026604
L$LOAD   002100 G      L10046  026256      MREG14  020431       RANSEL  002402       TSEL16  027006
L$LUN    002074 G      L10047  026462      MREG16  020474       RANST   002400       TSEL4   025572
L$MREV   002050 G      L10050  026356      MREG2   020152       RAN1    013414       TSEL6   025770
L$NAME   002000 G      L10051  026460      MREG4   020215       RAN2    013432       TSPEED  012452
L$PRIO   002042 G      L10052  026664      MREG6   020260       RAN4    013470       TSTERR  013152
L$PROT   002122 G      L10053  026560      MSEL0   020042       READ    015072       TSTNUB  015010
L$PRT    002112 G      L10054  026662      MSEL10  020645       REGADR  012532       TSTREG  024576
L$REPP   002062 G      L10055  027066      MSEL2   020537       RESTST  024404       TTABLE  002426
L$REV    002010 G      L10056  026762      MSEL4   020602       REVPRO  030262       TWODMA  031750
L$RPT    023412 G      L10057  027064      NERRS   013102       RGALL   027112       TXDATA  012446
L$SOFT   034256 G      L10060  027504      NEXT    023560       ROMMAP  024264       T$ARGC= 000002
L$SPC    002056 G      L10061  027724      NUB     012456       RSTREG  013610       T$CODE= 002032
L$SPCP   002020 G      L10062  030076      NUMBER  002416       RTABLE  006426       T$ERRN= 000106
L$SPTP   002024 G      L10063  030250      O$APTS= 000000       RUNNIN  024044       T$EXCP= 000000
L$STA    002030 G      L10064  030314      O$AU   = 000000      RXCNT   012460       T$FLAG= 000040
L$SW     002266        L10065  030442      O$BGNR= 000000       RXDATA  012450       T$FREE= 034416
L$TEST   002114 G      L10066  031100      O$BGNS= 000000       SAVE4   002312       T$GMAN= 000000
L$TIML   002014 G      L10067  030664      O$DU   = 000001      SAVE6   002314       T$HILI= 000007
L$UIT    002270        L10070  031076      O$ERRT= 000000       SAVPC   002310       T$LAST= 000001
L$UNIT   002012 G      L10071  031716      O$GNSW= 000001       SAVPC1  002410       T$LOLI= 000004
L10001   002222        L10072  031404      O$POIN= 000001       SAVREG  013530       T$LSYM= 010000
```

```
KMV11 A/B LOGIC DIAG    MACRO M1200  05-APR-84 11:23  PAGE 82-3                                                    SEQ 129
SYMBOL TABLE

T$LTNO= 000027          T$$AUT= 010020          T11.1   026464          T21     031720 G        T9.1    026052
T$NEST= 177777          T$$CLE= 010021          T11.2   026562          T22     032430 G        T9.2    026156
T$NSO = 000000          T$$DAT= 010107          T12     026666 G        T22.1   032440          UAM   = 000200 G
T$NS1 = 000005          T$$DU = 010022          T12.1   026666          T22.2   032774          UNIT    002272
T$NS2 = 000002          T$$HAR= 010103          T12.2   026764          T23     033334 G        UUT     012472
T$PCNT= 000000          T$$HW = 010001          T13     027070 G        T23.1   033334          VECT    012440
T$PTAB= 010106          T$$INI= 010017          T14     027506 G        T23.2   033610          VECTOR  034164
T$PTHV= 000001          T$$MSG= 010015          T15     027726 G        T3      024576 G        VECTO   032546
T$PTNU= 000001          T$$PC = 000001          T16     030100 G        T4      024736 G        VECT4   033102
T$SAVL= 177777          T$$PRO= 010000          T17     030252 G        T5      025100 G        WAIT1   013000
T$SEGL= 177777          T$$PTA= 010106          T18     030316 G        T6      025232 G        WAIT2   012760
T$SIZE= 000006          T$$RPT= 010016          T19     030444 G        T6.1    025242          WRITE   015044
T$SUBN= 000002          T$$SOF= 010104          T19.1   030454          T6.2    025346          X$ALWA= 000000
T$TAGL= 177777          T$$SUB= 010102          T19.2   030666          T7      025460 G        X$FALS= 000040
T$TAGN= 010110          T$$TES= 010100          T2      024402 G        T7.1    025460          X$OFFS= 000400
T$TEMP= 000000          T1      024264 G        T2.1    024402          T7.2    025550          X$TRUE= 000020
T$TEST= 000027          T10     026262 G        T2.2    024520          T8      025654 G        $LSTIN= 000000
T$TSTM= 177777          T10.1   026262          T20     031102 G        T8.1    025654          $LSTTA= 000000
T$TSTS= 000001          T10.2   026360          T20.1   031102          T8.2    025746          $PATCH  034256 G
T$$AU = 010023          T11     026464 G        T20.2   031406          T9      026052 G

. ABS.  034416      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  29232 WORDS  ( 115 PAGES)
DYNAMIC MEMORY:  19748 WORDS  ( 75 PAGES)
ELAPSED TIME:  00:19:17
ÇNKMDA.BIC,CNKMDA.SEQ/CRF/-SP=SVC34.MLB/ML,CNKMDA.MAC
```