PDP11/70

POWER FAIL TEST

MD-11-DEKBG-A

EP-DEKBG-A-DL-A

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DEKBG-A-D
PRODUCT NAME:	PDP-11/70 POWER FAIL TEST
DATE CREATED:	1-MAR-75
MAINTAINER:	DIAGNOSTIC ENGINEERING
AUTHORS:	JIM LACEY

CONTENTS

1. ABSTRACT
2. REQUIREMENTS
2.1 Equipment
2.2 Storage
2.3 Preliminary programs
3. LOADING PROCEDURE
4. STARTING PROCEDURE
4.1 Control switch settings
4.2 Starting address
4.3 Program and/or operator action
5. OPERATING PROCEDURE
5.1 Operational switch settings
5.2 Subroutine abstract
6. ERRORS
7. RESTRICTIONS
8. MISCELLANEOUS
8.1 Execution time
8.2 Stack pointer

1. Abstract

This program is made up of 16 subtests to check out the power fail on the 11/70. The 2 msec. power down and power up time is checked on each power fail. Initally power fails are tried in all prosessor modes then error conditions like red zone, yellow zone, time out, and odd address in all the processor modes. Finally a power fail is done with memory management aborts occuring and a memory volitility test is run on all available memory.

2. Requirements

2.1 Equipment

PDP11/70 standard computer with up to 1M words of core memory.

2.2 Storage

Program Storage - the routines use memory 0 - 4472

2.3 Preliminary programs

All processor diagnostics

3. Loading procedure

Use standard procedure for "XXDP" media.

4. Starting Procedure

4.1 Control switch settings

See 5.1.1

4.2 Starting address

Load Address 200 and Start.

5. Operating procedure

Load Address 200 and START. A message will be typed which is the name of the program, the size of memory, and running instructions. Turn the DISPLAY switch to DISPLAY REGISTER

and power down then up when the test number appears in the lights. Do this for each test until the count recycles to 1. Each subtest is executed once except test 16 which runs 8 times before continuing. SW14 causes the program loop on the current test..blank 2

5.1 Operational switch settings

At SA 200 ... all switches down will run through each test and HALT on error. SW14 should be used to loop on the current test.

5.1.1 Switch settings are:

sw<14> = 1 ..... scope loop

5.2 Subroutine Abstracts

5.2.1 POWDWN and POWUP

These poutines are used to save and restore vital registers and test the time allowed for power fail by the processor. A SOB loop is used to check the timing. LOC 1100 contains the timing factor. Control is returned to the program via. JMP (R3) so the power fail return address is put in R3. ILLUP and ILLDWN are used for reporting not enough time to power down and up.

6. Errors

6.1 Error printout

None

6.2 Error HALTs

The program will HALT on error. The DISPLAY switch should be turned to the DATA PATHS position for the failing data. R0, which is displayed on a HALT, contains the bad data or bad address (see listing) in most of the tests.

If an error occurs in test 16, the data can be examined by turning the MODE switch to KERNAL I, load address with the address in R0 and examine. To calculate the failing address, examine KIPAR5 (17772352) and use that for the offset to the address in R0. To do this, use KIPAR5 as bits <21:6> and add R0<12:0> to it. This is the physical address of the bad data.

If the processor HALT's at ILLUP, the power down routine did

not have enough time to complete. If it HALT's at ILLDWN, the processor powered down before the up routine completed. In both cases, 2 msec is the minimum time allowed by the processor. The program must be restarted at 200 after these errors. The address of the power failed routine is in location ERROR.

6.3 Error recovery

Hit continue or Restart at 200

7. Restrictions

None

8. Miscellaneous

8.1 Execution time

N/A

8.2 Stack Pointer

Stack is initally set to 1100

%

```
                                .TITLE  MAINDEC-11-DEKBG-A      PDP-11/70 POWER FAIL
                                ;*COPYRIGHT (C) 1975
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY JIM LACEY
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-A3).
                                ;*


                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        14            LOOP ON TEST
                                ;*        10            INHIBIT BELL AT END-OF-PASS


                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100            ;;FIRST ADDRESS OF THE STACK
001100                          KERSTK= STACK           ;;KERNEL STACK
000700                          SUPSTK= STACK-200       ;;SUPERVISOR STACK
```

```
000600          USESTK= STACK-300       ;;USER STACK
                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL
177776          PS=     177776          ;;PROCESSOR STATUS WORD
```

```
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          SWR=    177570          ;;SWITCH REGISTER
177570                          DISPLAY=SWR

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE LINE FEED
000015                          CR=     15              ;;CODE CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
                                .EQUIV  R0,R10          ;;GENERAL REGISTER
                                .EQUIV  R1,R11          ;;GENERAL REGISTER
                                .EQUIV  R2,R12          ;;GENERAL REGISTER
                                .EQUIV  R3,R13          ;;GENERAL REGISTER
                                .EQUIV  R4,R14          ;;GENERAL REGISTER
                                .EQUIV  R5,R15          ;;GENERAL REGISTER
                                .EQUIV  R6,SP           ;;STACK POINTER
                                .EQUIV  SP,KSP          ;;KERNEL STACK POINTER
                                .EQUIV  SP,SSP          ;;SUPERVISOR STACK POINTER
                                .EQUIV  SP,USP          ;;USER STACK POINTER
                                .EQUIV  R7,PC           ;;PROGRAM COUNTER

                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
000140                          PR3=    140             ;;PRIORITY LEVEL 3
000200                          PR4=    200             ;;PRIORITY LEVEL 4
000240                          PR5=    240             ;;PRIORITY LEVEL 5
000300                          PR6=    300             ;;PRIORITY LEVEL 6
000340                          PR7=    340             ;;PRIORITY LEVEL 7

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13=   20000
010000                          SW12=   10000
004000                          SW11=   4000
002000                          SW10=   2000
001000                          SW09=   1000
000400                          SW08=   400
000200                          SW07=   200
000100                          SW06=   100
000040                          SW05=   40
000020                          SW04=   20
```

```
                        SW03=   10
                        SW02=   4
                        SW01=   2
                        SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0

                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
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
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000114                  CACHVEC=114             ;;CACHE ERROR INTERRUPT VECTOR
```

```
                        
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
000250                  MMVEC=  250             ;;MEMORY MANAGEMENT VECTOR
                        
                        .SBTTL  CACHE   REGISTER DEFINITIONS
                        
                        
177740                  LOADRS = 177740         ;;LOWER 16 BITS OF ADDRESS THAT CAUSED ERROR
177742                  HIADRS = 177742         ;;UPPER SIX BITS OF ADDRESS THAT CAUSED ERROR
177744                  MEMERR = 177744         ;;CACHE ERROR REGISTER
177746                  CONTRL = 177746         ;;MEMORY CONTROL REGISTER
177750                  MAINT  = 177750         ;;MEMORY MAINTENENCE REGISTER
177752                  HITMIS = 177752         ;;HIT MISS REGISTER "1" IMPLIES HIT IN CACHE
                        
                        
                        .SBTTL  CPU REGISTER DEFINITIONS
                        
                        
177760                  SIZELO = 177760         ;;MEMORY SIZE REGISTER NUMBER TO PUT INTO A PAR
                                                ;;TO GET TO THE LAST 32 WORDS OF MEMORY
177762                  SIZEHI = 177762         ;;HIGH SIZE REGISTER, RESERVED FOR FUTURE USE
                                                ;;CURRENTLY ALL ZERO
177764                  SYSTID = 177764         ;;SYSTEM ID REGISTER
177766                  CPUERR = 177766         ;;CPU ERROR REGISTER HOLDS CONDITION THAT CAUSED
                                                ;;THE TRAP TO ERRVEC (000004)
                        
                        
                        
                        .SBTTL  MEMORY MANAGEMENT DEFINITIONS
                        
                        ;*MEMORY MANAGEMENT STATUS REGISTER ADDRESSES
                        
177572                  MMR0=   177572
177574                  MMR1=   177574
177576                  MMR2=   177576
172516                  MMR3=   172516
                        .EQUIV  MMR0,SR0
                        .EQUIV  MMR1,SR1
                        .EQUIV  MMR2,SR2
                        .EQUIV  MMR3,SR3
                        
                        ;*USER "I" PAGE DESCRIPTOR REGISTERS
                        
177600                  UIPDR0= 177600
177602                  UIPDR1= 177602
177604                  UIPDR2= 177604
177606                  UIPDR3= 177606
177610                  UIPDR4= 177610
177612                  UIPDR5= 177612
177614                  UIPDR6= 177614
177616                  UIPDR7= 177616
                        
                        ;*USER "D" PAGE DESCRIPTOR REGISTORS
                        
177620                  UDPDR0= 177620
```

```
                                UDPDR1= 177622
                                UDPDR2= 177624
                                UDPDR3= 177626
                                UDPDR4= 177630
                                UDPDR5= 177632
                                UDPDR6= 177634
                                UDPDR7= 177636

                                ;*USER "I" PAGE ADDRESS REGISTERS

                                UIPAR0= 177640
                                UIPAR1= 177642
                                UIPAR2= 177644
                                UIPAR3= 177646
                                UIPAR4= 177650
                                UIPAR5= 177652
                                UIPAR6= 177654
                                UIPAR7= 177656

                                ;*USER "D" PAGE ADDRESS REGISTERS

                                UDPAR0= 177660
                                UDPAR1= 177662
                                UDPAR2= 177664
                                UDPAR3= 177666
                                UDPAR4= 177670
                                UDPAR5= 177672
                                UDPAR6= 177674
                                UDPAR7= 177676

                                ;*SUPERVISOR "I" PAGE DESCRIPTOR REGISTERS

                                SIPDR0= 172200
                                SIPDR1= 172202
                                SIPDR2= 172204
                                SIPDR3= 172206
                                SIPDR4= 172210
                                SIPDR5= 172212
                                SIPDR6= 172214
                                SIPDR7= 172216

                                ;*SUPERVISOR "D" PAGE DESCRIPTOR REGISTERS

                                SDPDR0= 172220
                                SDPDR1= 172222
                                SDPDR2= 172224
                                SDPDR3= 172226
                                SDPDR4= 172230
                                SDPDR5= 172232
                                SDPDR6= 172234
                                SDPDR7= 172236

                                ;*SUPERVISOR "I" PAGE ADDRESS REGISTERS

                                SIPAR0= 172240
                                SIPAR1= 172242
```

```
                        172244          SIPAR2= 172244
                        172246          SIPAR3= 172246
                        172250          SIPAR4= 172250
                        172252          SIPAR5= 172252
                        172254          SIPAR6= 172254
                        172256          SIPAR7= 172256

                                        ;*SUPERVISOR "D" PAGE ADDRESS REGISTERS

                        172260          SDPAR0= 172260
                        172262          SDPAR1= 172262
                        172264          SDPAR2= 172264
                        172266          SDPAR3= 172266
                        172270          SDPAR4= 172270
                        172272          SDPAR5= 172272
                        172274          SDPAR6= 172274
                        172276          SDPAR7= 172276

                                        ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

                        172300          KIPDR0= 172300
                        172302          KIPDR1= 172302
                        172304          KIPDR2= 172304
                        172306          KIPDR3= 172306
                        172310          KIPDR4= 172310
                        172312          KIPDR5= 172312
                        172314          KIPDR6= 172314
                        172316          KIPDR7= 172316

                                        ;*KERNEL "D" PAGE DESCRIPTOR REGISTERS

                        172320          KDPDR0= 172320
                        172322          KDPDR1= 172322
                        172324          KDPDR2= 172324
                        172326          KDPDR3= 172326
                        172330          KDPDR4= 172330
                        172332          KDPDR5= 172332
                        172334          KDPDR6= 172334
                        172336          KDPDR7= 172336

                                        ;*KERNEL "I" PAGE ADDRESS REGISTERS

                        172340          KIPAR0= 172340
                        172342          KIPAR1= 172342
                        172344          KIPAR2= 172344
                        172346          KIPAR3= 172346
                        172350          KIPAR4= 172350
                        172352          KIPAR5= 172352
                        172354          KIPAR6= 172354
                        172356          KIPAR7= 172356

                                        ;*KERNEL "D" PAGE ADDRESS REGISTERS

                        172360          KDPAR0= 172360
                        172362          KDPAR1= 172362
                        172364          KDPAR2= 172364
```

```
                        KDPAR3= 172366
                        KDPAR4= 172370
                        KDPAR5= 172372
                        KDPAR6= 172374
                        KDPAR7= 172376


                        .SBTTL  UNIBUS MAP REGISTER DEFINITIONS

                        ;*THE LOWER 16 BITS OF THE MAP REGISTERS ARE LABELED 'MAPLXX'
                        ;*THE UPPER 6 BITS OF THE MAP REGISTERS ARE LABELED 'MAPHXX'

                        MAPL00 = 170200
                        MAPH00 = 170202
                        MAPL01 = 170204
                        MAPH01 = 170206
                        MAPL02 = 170210
                        MAPH02 = 170212
                        MAPL03 = 170214
                        MAPH03 = 170216
                        MAPL04 = 170220
                        MAPH04 = 170222
                        MAPL05 = 170224
                        MAPH05 = 170226
                        MAPL06 = 170230
                        MAPH06 = 170232
                        MAPL07 = 170234
                        MAPH07 = 170236
                        MAPL10 = 170240
                        MAPH10 = 170242
                        MAPL11 = 170244
                        MAPH11 = 170246
                        MAPL12 = 170250
                        MAPH12 = 170252
                        MAPL13 = 170254
                        MAPH13 = 170256
                        MAPL14 = 170260
                        MAPH14 = 170262
                        MAPL15 = 170264
                        MAPH15 = 170266
                        MAPL16 = 170270
                        MAPH16 = 170272
                        MAPL17 = 170274
                        MAPH17 = 170276
                        MAPL20 = 170300
                        MAPH20 = 170302
                        MAPL21 = 170304
                        MAPH21 = 170306
                        MAPL22 = 170310
                        MAPH22 = 170312
                        MAPL23 = 170314
                        MAPH23 = 170316
```

```
                170320          MAPL24 = 170320
                170320          MAPH24 = 170320
                170324          MAPL25 = 170324
                170326          MAPH25 = 170326
                170330          MAPL26 = 170330
                170332          MAPH26 = 170332
                170334          MAPL27 = 170334
                170336          MAPH27 = 170336
                170340          MAPL30 = 170340
                170342          MAPH30 = 170342
                170344          MAPL31 = 170344
                170346          MAPH31 = 170346
                170350          MAPL32 = 170350
                170352          MAPH32 = 170352
                170354          MAPL33 = 170354
                170356          MAPH33 = 170356
                170360          MAPL34 = 170360
                170362          MAPH34 = 170362
                170364          MAPL35 = 170364
                170366          MAPH35 = 170366
                170370          MAPL36 = 170370
                170372          MAPH36 = 170372
                170374          MAPL37 = 170374
                170376          MAPH37 = 170376
                                .EQUIV  MAPL00,MAPL0
                                .EQUIV  MAPH00,MAPH0
                                .EQUIV  MAPL01,MAPL1
                                .EQUIV  MAPH01,MAPH1
                                .EQUIV  MAPL02,MAPL2
                                .EQUIV  MAPH02,MAPH2
                                .EQUIV  MAPL03,MAPL3
                                .EQUIV  MAPH03,MAPH3
                                .EQUIV  MAPL04,MAPL4
                                .EQUIV  MAPH04,MAPH4
                                .EQUIV  MAPL05,MAPL5
                                .EQUIV  MAPH05,MAPH5
                                .EQUIV  MAPL06,MAPL6
                                .EQUIV  MAPH06,MAPH6
                                .EQUIV  MAPL07,MAPL7
                                .EQUIV  MAPH07,MAPH7
```

```
                                .SBTTL  TRAP CATCHER

        000000                          .=0
                                ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
                                ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
                                ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS

                                .SBTTL  STARTING ADDRESS(ES)
        000200                          .=200

000200  000137  001102                  JMP     @#BEGIN         ;;JUMP TO STARTING ADDRESS OF PROGRAM
                                ;;*****************************************************************

                                .SBTTL          ACT11 HOOKS

                                ;*THE FOLLOWING LOCATIONS ARE SETUP TO BE USED WITH ACT11
                                ;*
                                ;*LOCATION 46 WILL CONTAIN THE ADDRESS OF THE LOCICAL
                                ;*END OF THE PROGRAM.
                                ;*LOCATION 52 IS USED TO SPECIFY PROGRAM OPERATING REQUIREMENTS
                                ;*AND/OR RESTRICTIONS. THIS IS ACCOMPLISHED BY SETTING VARIOUS BITS
                                ;*TO A ONE OR A ZERO. THE BITS USED AND THERE MEANING ARE:
                                ;*
                                ;*      BIT 15=1 PROGRAM SHOULD BE POWER FAILED WHILE RUNNING
                                ;*           =0 NO POWER FAIL DESIRED
                                ;*
                                ;*      BIT 14=1 PROGRAM RUN TIME IS MEMORY SIZE DEPENDENT
                                ;*           =0 RUN TIME IS NOT MEMORY SIZE DEPENDENT
                                ;*
                                ;*      BITS 13-0  MUST BE ZERO'S

        000204                  $SVPC=.                 ;;SAVE LOCATION COUNTER
        000046                  .=46                    ;;SET LOCATION COUNTER
000046  003132                  .WORD   $ENDAD          ;;SET LOC.46 TO ADDRESS $ENDAD
        000052                  .=52                    ;;SET LOCATION COUNTER
000052  040000                  .WORD   BIT14           ;;SET LOC.52 TO BIT14
        000204                  .=$SVPC                 ;; RESTORE LOCATION COUNTER
```

```
                001100                          .=1100
                
001100  006405          FACTOR: .WORD   20000./6        ;20000 E-7/6 E-7 = 2MS/.6NS =
                                                        ;POWER DOWN TIME DIVIDED BY "SOB" EXECUTION TIME
001102                  BEGIN:
001102  012706  001100          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
001106  012737  004440  000034  MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001114  012737  000340  000036  MOV     #340,@#TRAPVEC+2;LEVEL 7
001122  005067  002020          CLR     $PASS           ;;CLEAR THE PASS COUNT
001126  016767  001764  001754  MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
001134  012777  003472  002506  MOV     #POWDWN,@DVEC   ;SET UP POWER DOWN VECTOR
001142  012737  003372  000114  MOV     #PARERR,@#CACHVEC ;SET UP PARITY ERROR VECTOR
001150  012737  000001  003374  MOV     #1,@#PARFLG     ;INITIALIZE THE MULTI PARITY ERROR INDICATOR

                ;DETERMINE THE NUMBER OF 4K BLOCKS MEMORY IS MADE UP OF.

001156  005000          CLR     R0              ;MAKE MSB'S = 0
001160  013701  177760  MOV     @#SIZELO,R1     ;GET PAR VALUE FOR LAST 32 WORDS OF
001164  005201          INC     R1              ;MEMORY AND ROUND UP TO A 4K BOUNDARY
001166  071027  000200  DIV     #200,R0         ;FORM THE BLOCK NUMBER OF
001172  005300          DEC     R0              ;THE LAST 4K BLOCK AND
001174  010067  002460  MOV     R0,LIMIT        ;SAVE IT.

                ;IDENTIFY THE PROGRAM AND TYPE INSTRUCTIONS

001200  005227  177777  INC     #-1             ;;FIRST TIME?
001204  001116          BNE     TST1            ;;BRANCH IF NO
001206  022737  003132  000042  CMP     #$ENDAD,@#42    ;;ACT-11?
001214  001512          BEQ     TST1            ;;BRANCH IF YES
001216  104400  001224  TYPE    ,65$            ;;TYPE ASCIZ STRING
001222  000413          BR      64$             ;;GET OVER THE ASCIZ
                ;;65$:   .ASCIZ  <CRLF>*MAINDEC-11-DEKBG-A*<CRLF>
001252                  64$:
001252  104400  001260  TYPE    ,67$            ;;TYPE ASCIZ STRING
001256  000406          BR      66$             ;;GET OVER THE ASCIZ
                ;;67$:   .ASCIZ  <CRLF>*BANKS 0 - *
001274                  66$:
001274  010046          MOV     R0,-(SP)        ;;SAVE R0 FOR TYPEOUT
001276  104404          TYPOS                   ;;GO TYPE--OCTAL ASCII
001300     006          .BYTE   6               ;;TYPE 6 DIGITS
001301     000          .BYTE   0               ;;SUPPRESS LEADING ZEROS
001302  104400  001310  TYPE    ,69$            ;;TYPE ASCIZ STRING
001306  000404          BR      68$             ;;GET OVER THE ASCIZ
                ;;69$:   .ASCIZ  * EXIST*
001320                  68$:
001320  104400  001326  TYPE    ,71$            ;;TYPE ASCIZ STRING
001324  000423          BR      70$             ;;GET OVER THE ASCIZ
                ;;71$:   .ASCIZ  <CRLF><LF>*INTERRUPT THE POWER AFTER THE TEST*
001374                  70$:
001374  104400  001402  TYPE    ,73$            ;;TYPE ASCIZ STRING
001400  000420          BR      72$             ;;GET OVER THE ASCIZ
                ;;73$:   .ASCIZ  * NUMBER APPEARS IN THE DISPLAY*<CRLF>
001442                  72$:
```

```
                                        ;;*****************************************************************
                                        ;*TEST 1        SIMPLE DOWN/UP TEST (KERNAL)
                                        ;;*****************************************************************
001442                                  TST1:
001442  012737  000001  177570                  MOV     #1,@#DISPLAY    ;SET TEST NUMBER
001450  005037  177776                          CLR     @#PS            ;SET KERNAL MODE
001454  012703  001462                          MOV     #2$,R3          ;SET POWER UP RETURN
001460  000001                                  WAIT                    ;WAIT FOR THE POWER FAIL
001462  010600                          2$:     MOV     SP,R0           ;GET SP
001464  022700  001074                          CMP     #STACK-4,R0     ;CHECK SP
001470  001401                                  BEQ     .+4             ;SKIP IF OK
001472  000000                                  HALT                    ;SP NOT "STACK-4"
001474  012706  001100                          MOV     #STACK,SP       ;RESET SP
001500  013700  001074                          MOV     @#STACK-4,R0    ;GET RETURN ADDRESS
001504  022700  001462                          CMP     #2$,R0          ;CHECK ADDRESS
001510  001401                                  BEQ     .+4             ;SKIP IF OK
001512  000000                                  HALT                    ;ADDRESS ON STACK IS WRONG
001514  013700  001076                          MOV     @#STACK-2,R0    ;GET OLD PS
001520  022700  000000                          CMP     #0,R0           ;CHECK OLD PS
001524  001401                                  BEQ     .+4             ;SKIP IF OK
001526  000000                                  HALT                    ;OLD PS IS WRONG
001530                                  1$:
001530  032737  040000  177570                  BIT     #SW14,@#SWR     ;LOOP ON TEST?
001536  001341                                  BNE     TST1            ;LOOP TO TST1


                                        ;;*****************************************************************
                                        ;*TEST 2        SIMPLE DOWN/UP TEST (SUPERVISOR)
                                        ;;*****************************************************************
001540                                  TST2:
001540  012737  000002  177570                  MOV     #2,@#DISPLAY    ;SET TEST NUMBER
001546  012737  040000  177776                  MOV     #40000,@#PS     ;SET SUPERVISOR MODE
001554  012703  001562                          MOV     #2$,R3          ;SET POWER UP RETURN
001560  000001                                  WAIT                    ;WAIT FOR THE POWER FAIL
001562  012706  001100                  2$:     MOV     #STACK,SP       ;RESET SP
001566  013700  001074                          MOV     @#STACK-4,R0    ;GET RETURN ADDRESS
001572  022700  001562                          CMP     #2$,R0          ;CHECK ADDRESS
001576  001401                                  BEQ     .+4             ;SKIP IF OK
001600  000000                                  HALT                    ;ADDRESS ON STACK IS WRONG
001602  013700  001076                          MOV     @#STACK-2,R0    ;GET OLD PS
001606  022700  040000                          CMP     #40000,R0       ;CHECK OLD PS
001612  001401                                  BEQ     .+4             ;SKIP IF OK
001614  000000                                  HALT                    ;OLD PS IS WRONG
001616                                  1$:
001616  032737  040000  177570                  BIT     #SW14,@#SWR     ;LOOP ON TEST?
001624  001345                                  BNE     TST2            ;LOOP TO TST2
```

```
                                        ;;*****************************************************************
                                        ;*TEST 3         SIMPLE DOWN/UP TEST (USER)
                                        ;;*****************************************************************
001626                                  TST3:
001626  012737  000003  177570                  MOV     #3,@#DISPLAY    ;SET TEST NUMBER
001634  012737  140000  177776                  MOV     #140000,@#PS    ;SET USER MODE
001642  012703  001650                          MOV     #2$,R3          ;SET POWER UP RETURN
001646  000001                                  WAIT                    ;WAIT FOR THE POWER FAIL
001650  012706  001100                2$:       MOV     #STACK,SP       ;RESET SP
001654  013700  001074                          MOV     @#STACK-4,R0    ;GET RETURN ADDRESS
001660  022700  001650                          CMP     #2$,R0          ;CHECK ADDRESS
001664  001401                                  BEQ     .+4             ;SKIP IF OK
001666  000000                                  HALT                    ;ADDRESS ON STACK IS WRONG
001670  013700  001076                          MOV     @#STACK-2,R0    ;GET OLD PS
001674  022700  140000                          CMP     #140000,R0      ;CHECK OLD PS
001700  001401                                  BEQ     .+4             ;SKIP IF OK
001702  000000                                  HALT                    ;OLD PS IS WRONG
001704                                  1$:
001704  032737  040000  177570                  BIT     #SW14,@#SWR     ;LOOP ON TEST?
001712  001345                                  BNE     TST3            ;LOOP TO TST3


                                        ;;*****************************************************************
                                        ;*TEST 4         POWER FAIL WITH ODD ADDRESS
                                        ;;*****************************************************************
001714                                  TST4:
001714  012737  000004  177570                  MOV     #4,@#DISPLAY    ;SET TEST NUMBER
001722  005037  177776                          CLR     @#PS            ;SET KERNAL MODE
001726  012737  001740  000004                  MOV     #3$,@#ERRVEC    ;SET TRAP VECTOR
001734  012703  001752                          MOV     #1$,R3          ;SET RETURN ADDRESS FOR POWER FAIL
001740  012706  001100                3$:       MOV     #STACK,SP       ;RESET STACK
001744  005737  000003                          TST     @#3             ;CAUSE ODD ADDRESS TRAP
001750  000000                                  HALT                    ;ODD ADDRESS TRAP FAILED
001752  012737  000006  000004        1$:       MOV     #6,@#ERRVEC     ;RESET 4
001760  032737  040000  177570                  BIT     #SW14,@#SWR     ;LOOP ON TEST?
001766  001352                                  BNE     TST4            ;LOOP TO TST4
```

```
                                ;;*****************************************************************
                                ;*TEST 5        POWER FAIL IN THE RED ZONE
                                ;;*****************************************************************
001770                          TST5:
001770  012737  000005  177570          MOV     #5,@#DISPLAY    ;SET TEST NUMBER
001776  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002002  012737  002022  000004          MOV     #2$,@#ERRVEC    ;SET TRAP REGISTER
002010  012703  002040                  MOV     #1$,R3          ;SET POWER UP RETURN
002014  012706  000002                  MOV     #2,SP           ;SET STACK TO RED ZONE
002020  000001                          WAIT                    ;WAIT FOR POWER FAIL TRAP
002022  012777  002030  001614  2$:     MOV     #7$,@UVEC       ;SET UVEC TO HALT
002030  000000                  7$:     HALT                    ;ILLEGAL TRAP TO 4
002032  012777  003472  001610          MOV     #POWDWN,@DVEC   ;RESET DVEC
002040  012706  001100          1$:     MOV     #STACK,SP       ;RESET STACK
002044  012737  000006  000004          MOV     #6,@#ERRVEC     ;RESET 4
002052  013700  000002                  MOV     @#2,R0          ;GET FOR TYPING
002056  005737  000002                  TST     @#2             ;IS 2 OK?
002062  001401                          BEQ     .+4             ;SKIP IF OK
002064  000000                          HALT                    ;NO!
002066  013700  000000                  MOV     @#0,R0          ;GET FOR TYPING
002072  022737  003472  000000          CMP     #POWDWN,@#0     ;IS 0 OK?
002100  001401                          BEQ     .+4             ;SKIP IF OK
002102  000000                          HALT                    ;0 IS WRONG!
002104  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002112  001326                          BNE     TST5            ;LOOP TO TST5

                                ;;*****************************************************************
                                ;*TEST 6        POWER FAIL WITH TIME OUT (KERNAL)
                                ;;*****************************************************************
002114                          TST6:
002114  012737  000006  177570          MOV     #6,@#DISPLAY    ;SET TEST NUMBER
002122  012737  002134  000004          MOV     #3$,@#ERRVEC    ;SET TRAP VECTOR
002130  012703  002152                  MOV     #1$,R3          ;SET UP RETURN ADDRESS FOR POWER FAIL
002134  012706  001100          3$:     MOV     #STACK,SP       ;SET STACK
002140  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002144  010037  173000                  MOV     R0,@#173000     ;CAUSE A TIMEOUT
002150  000000                          HALT                    ;TIMEOUT FAILED
002152  012706  001100          1$:     MOV     #STACK,SP       ;SET STACK
002156  012737  000006  000004          MOV     #6,@#ERRVEC     ;RESET 4
002164  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002172  001350                          BNE     TST6            ;LOOP TO TST6
```

```
                                ;;*****************************************************************
                                ;*TEST 7         POWER FAIL IN THE YELLOW ZONE (KERNAL)
                                ;;*****************************************************************
002174                          TST7:
002174  012737  000007  177570          MOV     #7,@#DISPLAY    ;SET TEST NUMBER
002202  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002206  005067  001444                  CLR     FLAG            ;CLEAR THE FLAG
002212  012737  002236  000004          MOV     #2$,@#ERRVEC    ;SET SICK TRAP ADDRESS
002220  012706  000400                  MOV     #400,SP         ;SET STACK TO YELLOW ZONE
002224  012703  002232                  MOV     #1$,R3          ;SET RETURN ADDRESS FOR POWER FAIL
002230  000001                          WAIT                    ;WAIT FOR POWER FAIL
002232  000000                  1$:     HALT                    ;POWER FAIL RETURNED TOO SOON
002234  000422                          BR      4$              ;SKIP SP CHECK
002236  012737  000006  000004  2$:     MOV     #6,@#ERRVEC     ;RESET 4
002244  005767  001406                  TST     FLAG            ;IS THE FIRST INSTRUCTION FLAG SET?
002250  001010                          BNE     5$              ;YES
002252  012777  002260  001364          MOV     #7$,@UVEC       ;SET UVEC TO HALT
002260  000000                  7$:     HALT                    ;NOT ENOUGH OR TOO MANY INSTR. EXEC.
002262  012777  003472  001360          MOV     #POWDWN,@DVEC   ;SET DVEC
002270  000404                          BR      4$              ;GET OUT
002272  012703  002302          5$:     MOV     #4$,R3          ;SET RETURN
002276  000002                          RTI                     ;GO TO THE POWER FAIL ROUTINE
002300  000000                          HALT                    ;SHOULD NOT RETURN HERE
002302                          4$:
002302  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002310  001331                          BNE     TST7            ;LOOP TO TST7

                                ;;*****************************************************************
                                ;*TEST 10        POWER FAIL WITH RESETS
                                ;;*****************************************************************
002312                          TST10:
002312  012737  000010  177570          MOV     #10,@#DISPLAY   ;SET TEST NUMBER
002320  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002324  012703  002344                  MOV     #1$,R3          ;SET RETURN ADDRESS
002330  012706  001100                  MOV     #STACK,SP       ;RESET STACK
002334  000005                  3$:     RESET                   ;RESETS
002336  000005                          RESET                   ;TO WAIT
002340  000005                          RESET                   ;IN
002342  000774                          BR      3$              ;LOOP
002344  012706  001100          1$:     MOV     #STACK,SP       ;RESET STACK
002350  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002356  001355                          BNE     TST10           ;LOOP TO TST10
```

```
                                ;;*****************************************************************
                                ;*TEST 11        POWER FAIL WITH ODD ADDRESS (SUPERVISOR)
                                ;;*****************************************************************
002360                          TST11:
002360  012737  000011  177570          MOV     #11,@#DISPLAY   ;SET TEST NUMBER
002366  012737  002400  000004          MOV     #3$,@#ERRVEC    ;SET TRAP VECTOR
002374  012703  002424                  MOV     #1$,R3          ;SET RETURN ADDRESS FOR POWER FAIL
002400  012706  001100          3$:     MOV     #STACK,SP       ;RESET STACK
002404  012737  040000  177776          MOV     #40000,@#PS     ;SET SUPERVISOR MODE
002412  005737  000003                  TST     @#3             ;CAUSE ODD ADDRESS TRAP
002416  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002422  000000                          HALT                    ;ODD ADDRESS TRAP FAILED
002424  012706  001100          1$:     MOV     #STACK,SP       ;RESET STACK POINTER
002430  012737  000006  000004          MOV     #6,@#ERRVEC     ;RESET 4
002436  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002444  001345                          BNE     TST11           ;LOOP TO TST11

                                ;;*****************************************************************
                                ;*TEST 12        POWER FAIL WITH TIME OUT (SUPERVISOR)
                                ;;*****************************************************************
002446                          TST12:
002446  012737  000012  177570          MOV     #12,@#DISPLAY   ;SET TEST NUMBER
002454  012737  002466  000004          MOV     #3$,@#ERRVEC    ;SET TRAP VECTOR
002462  012703  002512                  MOV     #1$,R3          ;SET UP RETURN ADDRESS FOR POWER FAIL
002466  012706  001100          3$:     MOV     #STACK,SP       ;RESET STACK
002472  012737  040000  177776          MOV     #40000,@#PS     ;SET SUPERVISOR MODE
002500  010037  173000                  MOV     R0,@#173000     ;CAUSE A TIMEOUT
002504  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002510  000000                          HALT                    ;TIMEOUT FAILED
002512  012706  001100          1$:     MOV     #STACK,SP       ;RESET STACK
002516  012737  000006  000004          MOV     #6,@#ERRVEC     ;RESET 4
002524  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002532  001345                          BNE     TST12           ;LOOP TO TST12
```

```
                                ;;*****************************************************************
                                ;*TEST 13       POWER FAIL WITH ODD ADDRESS (USER)
                                ;;*****************************************************************
002534                          TST13:
002534  012737  000013  177570          MOV     #13,@#DISPLAY   ;SET TEST NUMBER
002542  012737  002554  000004          MOV     #3$,@#ERRVEC    ;SET TRAP VECTOR
002550  012703  002600                  MOV     #1$,R3          ;SET RETURN ADDRESS FOR POWER FAIL
002554  012706  001100          3$:     MOV     #STACK,SP       ;RESET STACK
002560  012737  140000  177776          MOV     #140000,@#PS    ;SET USER MODE
002566  005737  000003                  TST     @#3             ;CAUSE ODD ADDRESS TRAP
002572  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002576  000000                          HALT                    ;ODD ADDRESS TRAP FAILED
002600  012706  001100          1$:     MOV     #STACK,SP       ;RESET SP
002604  012737  000006  000004          MOV     #6,@#ERRVEC     ;RESET 4
002612  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002620  001345                          BNE     TST13           ;LOOP TO TST13


                                ;;*****************************************************************
                                ;*TEST 14       POWER FAIL WITH TIME OUT (USER)
                                ;;*****************************************************************
002622                          TST14:
002622  0127    000014  177570          MOV     #14,@#DISPLAY   ;SET TEST NUMBER
002630  0127    002642  000004          MOV     #3$,@#ERRVEC    ;SET TRAP VECTOR
002636  012703  002666                  MOV     #1$,R3          ;SET UP RETURN ADDRESS FOR POWER FAIL
002642  012706  001100          3$:     MOV     #STACK,SP       ;RESET STACK
002646  012737  140000  177776          MOV     #140000,@#PS    ;SET USER MODE
002654  010037  173000                  MOV     R0,@#173000     ;CAUSE A TIMEOUT
002660  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002664  000000                          HALT                    ;TIMEOUT FAILED
002666  012706  001100          1$:     MOV     #STACK,SP       ;RESET STACK
002672  012737  000006  000004          MOV     #6,@#ERRVEC     ;RESET 4
002700  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002706  001345                          BNE     TST14           ;LOOP TO TST14
```

```
                                        ;;*****************************************************************
                                        ;*TEST 15       MEMORY MANAGEMENT ABORT TEST
                                        ;;*****************************************************************
002710                                  TST15:
002710  012737  000015  177570                MOV     #15,@#DISPLAY   ;SET TEST NUMBER
002716  005037  177776                        CLR     @#PS            ;SET KERNAL MODE
002722  012737  002762  000004                MOV     #4$,@#ERRVEC    ;SET FOR TIMEOUT
002730  004767  000730                        JSR     PC,MAP          ;MAP THE WORLD
002734  012737  002752  000250                MOV     #3$,@#MMVEC     ;SET MEMORY MANAGEMENT VECTOR
002742  012703  002764                        MOV     #1$,R3          ;LOAD PF RETURN
002746  005237  177572                        INC     @#MMR0          ;TURN MEMORY MANAGEMENT ON
002752  012706  001100            3$:         MOV     #STACK,SP       ;ZAP STACK
002756  005237  140000                        INC     @#140000        ;ACCESS VIOLATION
002762  000000                    4$:         HALT                    ;NO VIOLATION OR TRAP TO 4
                                        
002764  005037  177572            1$:         CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT
002770  012706  001100            2$:         MOV     #STACK,SP       ;MAKE A NEW STACK
002774  012737  000006  000004                MOV     #6,@#ERRVEC     ;RESET 4
003002  032737  040000  177570                BIT     #SW14,@#SWR     ;LOOP ON TEST?
003010  001337                                BNE     TST15           ;LOOP TO TST15
                                        
                                        ;;*****************************************************************
                                        ;*TEST 16       MEMORY VOLATILITY TEST
                                        ;;*****************************************************************
003012  000240                          TST16:  NOP
003014  005037  177776                        CLR     @#PS            ;SET KERNAL MODE
003020  012702  000010                        MOV     #10,R2          ;LOAD COUNT OF TEST ITERATIONS
003024  004767  000120            4$:         JSR     PC,LOAD         ;LOAD ALL MEMORY WITH 52525
003030  012703  003050                        MOV     #1$,R3          ;POWER FAIL RETURN ADDRESS
003034  012737  000016  177570                MOV     #16,@#DISPLAY   ;SET TEST NUMBER
003042  004767  000206            2$:         JSR     PC,CHECK        ;CHECK FOR THE 52525
003046  000775                                BR      2$              ;LOOP FOR EVER OR POWER FAIL
003050  012706  001100            1$:         MOV     #STACK,SP       ;ZAP THE STACK
003054  004767  000174                        JSR     PC,CHECK        ;CHECK ALL MEMORY
003060  077217                                SOB     R2,4$           ;DO IT 10 TIMES
003062  032737  040000  177570                BIT     #SW14,@#SWR     ;LOOP ON TEST?
003070  001350                                BNE     TST16           ;LOOP TO TST16
```

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

                                        .SBTTL  END OF PASS ROUTINE

                                        ;*INCREMENT THE PASS NUMBER ($PASS)
                                        ;*IF THERES A MONITOR GO TO IT
                                        ;*IF THERE ISN'T JUMP TO 200

003072                                  $EOP:
003072  000240                                  NOP
003074  005267  000046                          INC     $PASS           ;;INCREMENT THE PASS NUMBER
003100  042767  100000  000040                  BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
003106  005327                                  DEC     (PC)+           ;;LOOP?
003110  000001                          $EOPCT: .WORD   1
003112  003013                                  BGT     $DOAGN          ;;YES
003114  012737                                  MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
003116  000001                          $ENDCT: .WORD   1
003120  003110                                  $EOPCT
003122  013700  000042                  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
003126  001405                                  BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
003130  000005                                  RESET                   ;;CLEAR THE WORLD
003132  004710                          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
003134  000240                                  NOP                     ;;SAVE ROOM
003136  000240                                  NOP                     ;;FOR
003140  000240                                  NOP                     ;;ACT11
003142                                  $DOAGN:
003142  000137  000200                          JMP     @#200           ;;RETURN
003146  000000                          $PASS:  .WORD   0               ;;NUMBER OF PASSES
```

```
1042
1043                                        .SBTTL  LOAD MEMORY WITH DATA PATTERN
1044  003150  004767  000510                LOAD:   JSR     PC,MAP          ;SET UP MEMORY MANAGEMENT REGISTERS
1045  003154  016704  000500                        MOV     LIMIT,R4        ;GET BANK COUNT
1046  003160  016705  000476                        MOV     DATA,R5         ;GET THE DATA PATTERN
1047  003164  012737  003214  000250              MOV     #2$,@#MMVEC     ;SET UP FOR MEMORY MANAGEMENT ABORTS
1048  003172  012700  004470                        MOV     #END,R0         ;FORM ADDRESS OF WHERE TO START WRITING THE DATA
1049  003176  062700  120000                        ADD     #120000,R0      ;USE KIPAR5 FOR ADDRESSING
1050  003202  012737  000001  177572              MOV     #1,@#MMR0       ;TURN ON MEMORY MANAGEMENT
1051  003210  010520                      1$:     MOV     R5,(R0)+        ;WRITE THE DATA
1052  003212  000776                              BR      1$
1053  003214  005304                      2$:     DEC     R4              ;LAST BANK WRITTEN?
1054  003216  100411                              BMI     3$              ;BRANCH IF YES
1055  003220  012700  120000                        MOV     #120000,R0      ;RESET ADDRESSING REGISTER
1056  003224  062737  000200  172352              ADD     #200,@#KIPAR5   ;MAP NEXT 4K BANK
1057  003232  012737  000001  177572              MOV     #1,@#MMR0       ;CLEAR MEMORY MANAGEMENT ERRORS
1058  003240  000002                              RTI                     ;RETURN TO CONTINUE WRITING
1059  003242  005037  177572              3$:     CLR     @#MMR0          ;TURN OFF MEMORY MANAGEMENT
1060  003246  062706  000004                        ADD     #4,SP           ;CLEAN UP THE STACK
1061  003252  000207                              RTS     PC              ;RETURN TO CALLER
```

```
                                        .SBTTL  CHECK MEMORY FOR CORRECT DATA PATTERN
                                
003254  004767  000404          CHECK:  JSR     PC,MAP          ;SET UP MEMORY MANAGEMENT REGISTERS
003260  016704  000374                  MOV     LIMIT,R4        ;GET BANK COUNT
003264  016705  000372                  MOV     DATA,R5         ;GET DATA PATTERN
003270  012737  003332  000250          MOV     #2$,@#MMVEC     ;SET UP FOR MEMORY MANAGEMENT ABORTS
003276  012700  004470                  MOV     #END,R0         ;ADDRESS OF WHERE DATA PATTERN STARTS
003302  062700  120000                  ADD     #120000,R0      ;USE KIPAR5 FOR ADDRESSING
003306  012737  000001  177572          MOV     #1,@#MMR0       ;TURN ON MEM. MANAGEMENT
003314  012001                  1$:     MOV     (R0)+,R1        ;READ THE DATA
003316  020501                          CMP     R5,R1           ;IS IT GOOD?
003320  001775                          BEQ     1$              ;BRANCH IF YES
003322  000000                          HALT                    ;BAD DATA IS IN R1
003324  010560  177776                  MOV     R5,-2(R0)       ;WRITE GOOD DATA
003330  000771                          BR      1$              ;GO READ THE NEXT WORD
003332  005304                  2$:     DEC     R4              ;LAST BANK COMPARED?
003334  100411                          BMI     3$              ;BRANCH IF YES
003336  012700  120000                  MOV     #120000,R0      ;RESET ADDRESSING REGISTER
003342  062737  000200  172352          ADD     #200,@#KIPAR5   ;MAP NEXT 4K BANK
003350  012737  000001  177572          MOV     #1,@#MMR0       ;CLEAR MEMORY MANAGEMENT ERRORS
003356  000002                          RTI                     ;CONTINUE TESTING
003360  005037  177572          3$:     CLR     @#MMR0          ;TURN OFF MEM. MANAGEMENT
003364  062706  000004                  ADD     #4,SP           ;CLEAN UP THE STACK
003370  000207                          RTS     PC              ;RETURN FOR CALL
```

```
1086                                        .SBTTL  PARITY ERROR HANDLER
1087
1088   003372  005327                       PARERR: DEC     (PC)+           ;FIRST TIME IN?
1089   003374  000001                       PARFLG: .WORD   1
1090   003376  002002                               BGE     2$              ;BRANCH IF YES
1091   003400  000000                       1$:     HALT                    ;NO--ANOTHER PARITY ERROR OCCURRED WHILE
1092   003402  000776                               BR      1$              ;PROCESSING THE FIRST ONE.
1093   003404  032767  000077  174330       2$:     BIT     #77,HIADRS      ;MSB'S OF PARITY ERROR ADDRESS=0?
1094   003412  001006                               BNE     4$              ;BRANCH IF NO
1095   003414  023727  177740  004470               CMP     @#LOADRS,#END   ;LSB'S ABOVE THE PROGRAM?
1096   003422  101002                               BHI     4$              ;BRANCH IF YES
1097   003424  000000                       3$:     HALT                    ;PARITY ERROR IS IN THE PROGRAM
1098   003426  000776                               BR      3$
1099   003430  000000                       4$:     HALT                    ;DATA PARITY ERROR OCCURRED
1100   003432  005737  177744                       TST     @#MEMERR        ;DID AN ABORT OCCUR?
1101   003436  100405                               BMI     5$              ;BRANCH IF YES
1102   003440  162700  000002                       SUB     #2,R0           ;BACKUP BY ONE WORD
1103   003444  012705  000002                       MOV     #BIT01,R5
1104   003450  074500                               XOR     R5,R0           ;
1105   003452  010320                       5$:     MOV     R3,(R0)+        ;REWRITE THE BAD DATA
1106   003454  013737  177744  177744               MOV     @#MEMERR,@#MEMERR ;CLEAR ERROR INDICATORS
1107   003462  012767  000001  177704               MOV     #1,PARFLG       ;INITIALIZE PARITY ERROR FLAG
1108   003470  000002                               RTI                     ;CONTINUE TESTING
```

```
1109
1110                                        .SBTTL  POWER FAIL ROUTINE
1111
1112  003472  012767  177777  000156  POWDWN: MOV     #-1,FLAG        ;FIRST INSTRUCTION FLAG
1113  003500  005067  000152                CLR     FLAG            ;NOW CLEAR IT
1114  003504  012777  003632  000132        MOV     #ILLUP,@UVEC    ;IF TOO FAST
1115  003512  011667  000124                MOV     (SP),ERRAD      ;SET THE ERROR ADDRESS
1116  003516  022706  000440                CMP     #440,SP         ;YELLOW OR RED?
1117  003522  100402                        BMI     1$              ;NO
1118  003524  012706  001100                MOV     #STACK,SP       ;SET EMERGENCY STACK
1119  003530  010046                1$:     MOV     R0,-(6)         ;PUT
1120  003532  010146                        MOV     R1,-(6)         ;THE
1121  003534  010246                        MOV     R2,-(6)         ;REGISTERS
1122  003536  010346                        MOV     R3,-(6)         ;ON
1123  003540  010446                        MOV     R4,-(6)         ;THE
1124  003542  010546                        MOV     R5,-(6)         ;STACK
1125  003544  010667  000104                MOV     SP,SAV6         ;SAVE THE STACK POINTER
1126  003550  016700  175324                MOV     FACTOR,R0       ;SET TIME FACTOR
1127  003554  077001                        SOB     R0,.            ;NOW WAIT
1128  003556  012777  003566  000060        MOV     #POWUP,@UVEC    ;RESET THE UP VECTOR
1129  003564  000000                        HALT                    ;WAIT FOR POWER DOWN
1130
1131  003566  012777  003636  000054  POWUP: MOV     #ILLDWN,@DVEC   ;SET TOO FAST DOWN VECTOR
1132  003574  016706  000054                MOV     SAV6,SP         ;RESET SP
1133  003600  016700  175274                MOV     FACTOR,R0       ;SET TIME FACTOR
1134  003604  077001                        SOB     R0,.            ;WAIT
1135  003606  012605                        MOV     (6)+,R5         ;TAKE
1136  003610  012604                        MOV     (6)+,R4         ;THE
1137  003612  012603                        MOV     (6)+,R3         ;REGISTERS
1138  003614  012602                        MOV     (6)+,R2         ;FROM
1139  003616  012601                        MOV     (6)+,R1         ;THE
1140  003620  012600                        MOV     (6)+,R0         ;STACK
1141  003622  012777  003472  000020        MOV     #POWDWN,@DVEC   ;RESET THE DOWN VECTOR
1142  003630  000113                        JMP     (R3)            ;JUMP INDIRECT TO R3
1143
1144  003632  000000                ILLUP:  HALT                    ;POWER UP BEFORE POWER DOWN COMPLETE
1145  003634  000776                        BR      .-2             ;LOCK UP THE HALT
1146
1147  003636  000000                ILLDWN: HALT                    ;POWERED DOWN BEFORE UP COMPLETE
1148  003640  000776                        BR      .-2             ;LOCK UP THE HALT
1149
1150  003642  000000                ERRAD:  0                       ;RETURN ADDRESS FROM POWER FAIL
1151
1152  003644  000024  000026        UVEC:   24,26                   ;UP ADDRESS PAIR
1153  003650  000024  000026        DVEC:   24,26                   ;DOWN ADDRESS PAIR
1154  003654  000000                SAV6:   0                       ;SOME PLACE TO PUT THE SP
1155  003656  000000                FLAG:   0                       ;1 INSTRUCTION DOWN FLAG
1156  003660  000000                LIMIT:  0                       ;TOP OF MEMORY
1157  003662  052525                DATA:   52525                   ;WHAT IS TO BE WRITTEN INTO MEMORY
```

```
1158                                            .SBTTL  SETUP MEMORY MANAGMENT REGISTERS
1159
1160  003664  012737  000000  172340  MAP:      MOV     #0,@#KIPAR0     ;SETUP PAR0 FOR 1ST 4K
1161  003672  012737  077406  172300            MOV     #77406,@#KIPDR0 ;4K, R/W, EXPAND UP
1162  003700  012737  000000  172352            MOV     #0,@#KIPAR5     ;SET UP PAR5 FOR 1ST 4K
1163  003706  012737  077406  172312            MOV     #77406,@#KIPDR5 ;4K, R/W, ED=UP
1164  003714  012737  000200  172354            MOV     #200,@#KIPAR6   ;SET UP PAR6 FOR 2ND 4K
1165  003722  012737  000000  172314            MOV     #0,@#KIPDR6     ;ABORT ALL REFERENCES
1166  003730  012737  177600  172356            MOV     #177600,@#KIPAR7  ;SET UP PAR7 FOR I/O PAGE
1167  003736  012737  077406  172316            MOV     #77406,@#KIPDR7 ;4K, R/W, ED=UP
1168  003744  012737  000020  172516            MOV     #BIT04,@#MMR3   ;SET UP FOR 22-BIT MAPPING
1169  003752  012737  003762  000250            MOV     #MMERR,@#MMVEC  ;SET UP MEMORY MANAGEMENT VECTOR
1170  003760  000207                            RTS     PC              ;RETURN FROM CALL
1171  003762  000000            MMERR:  HALT                            ;MEMORY MANAGEMENT ERROR
1172  003764  000776                    BR      MMERR
1173
1174
```

```
1175                                    ;;**********************************************************************
1176
1177                                    .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
1178
1179                                    ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
1180                                    ;*OCTAL (ASCII) NUMBER AND TYPE IT.
1181                                    ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
1182                                    ;*CALL:
1183                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
1184                                    ;*      TYPOS                   ;;CALL FOR TYPEOUT
1185                                    ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
1186                                    ;*      .BYTE   M               ;;M=1 OR 0
1187                                    ;*                                      ;;1=TYPE LEADING ZEROS
1188                                    ;*                                      ;;0=SUPPRESS LEADING ZEROS
1189                                    ;*
1190                                    ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
1191                                    ;*$TYPOS OR $TYPOC
1192                                    ;*CALL:
1193                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
1194                                    ;*      TYPON                   ;;CALL FOR TYPEOUT
1195                                    ;*
1196                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
1197                                    ;*CALL:
1198                                    ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
1199                                    ;*      TYPOC                   ;;CALL FOR TYPEOUT
1200
1201  003766  017646  000000            $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
1202  003772  116667  000001  000211            MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
1203  004000  112667  000207                    MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
1204  004004  062716  000002                    ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
1205  004010  000406                            BR      $TYPON
1206  004012  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
1207  004020  112767  000006  000165            MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
1208  004026  112767  000005  000154  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
1209  004034  010346                            MOV     R3,-(SP)        ;;SAVE R3
1210  004036  010446                            MOV     R4,-(SP)        ;;SAVE R4
1211  004040  010546                            MOV     R5,-(SP)        ;;SAVE R5
1212  004042  116704  000145                    MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
1213  004046  005404                            NEG     R4
1214  004050  062704  000006                    ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
1215  004054  110467  000132                    MOVB    R4,$OMODE       ;;SAVE IT FOR USE
1216  004060  116704  000125                    MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
1217  004064  016605  000012                    MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
1218  004070  005003                            CLR     R3              ;;CLEAR THE OUTPUT WORD
1219  004072  006105                    1$:     ROL     R5              ;;ROTATE MSB INTO "C"
1220  004074  000404                            BR      3$              ;;GO DO MSB
1221  004076  006105                    2$:     ROL     R5              ;;FORM THIS DIGIT
1222  004100  006105                            ROL     R5
1223  004102  006105                            ROL     R5
1224  004104  010503                            MOV     R5,R3
1225  004106  006103                    3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
1226  004110  105367  000076                    DECB    $OMODE          ;;TYPE THIS DIGIT?
1227  004114  100016                            BPL     7$              ;;BR IF NO
1228  004116  042703  177770                    BIC     #177770,R3      ;;GET RID OF JUNK
1229  004122  001002                            BNE     4$              ;;TEST FOR 0
1230  004124  005704                            TST     R4              ;;SUPPRESS THIS 0?
```

```
1231   004126  001403                          BEQ     5$              ;;BR IF YES
1232   004130  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
1233   004132  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
1234   004136  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
1235   004142  110367  000040                  MOVB    R3,8$           ;;SAVE FOR TYPING
1236   004146  104400  004206                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
1237   004152  105367  000032          7$:     DECB    $OCNT           ;;COUNT BY 1
1238   004156  003347                          BGT     2$              ;;BR IF MORE TO DO
1239   004160  002402                          BLT     6$              ;;BR IF DONE
1240   004162  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
1241   004164  000744                          BR      2$              ;;GO DO THE LAST DIGIT
1242   004166  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
1243   004170  012604                          MOV     (SP)+,R4        ;;RESTORE R4
1244   004172  012603                          MOV     (SP)+,R3        ;;RESTORE R3
1245   004174  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
1246   004202  012616                          MOV     (SP)+,(SP)
1247   004204  000002                          RTI                     ;;RETURN
1248   004206     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
1249   004207     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
1250   004210     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
1251   004211     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
1252   004212  000000                  $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
1253                                  ;;**************************************************************
1254
1255                                  .SBTTL  TYPE ROUTINE
1256
1257                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
1258                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
1259                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
1260                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
1261                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
1262                                  ;*
1263                                  ;*CALL:
1264                                  ;*1) USING A TRAP INSTRUCTION
1265                                  ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
1266                                  ;*OR
1267                                  ;*      TYPE
1268                                  ;*      MESADR
1269                                  ;*
1270                                  ;*2) USING A JSR INSTRUCTION
1271                                  ;*      MOV     PS,-(SP)        ;;PUSH PROCESSOR STATUS WORD ON THE STACK
1272                                  ;*      JSR     PC,$TYPE        ;;CALL TYPE ROUTINE
1273                                  ;*      MESADDR                 ;;FIRST ADRESS OF MESSAGE
1274
1275   004214  105767  000213  $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
1276   004220  100002                  BPL     1$              ;;BR IF YES
1277   004222  000000                  HALT                    ;;HALT HERE IF NO TERMINAL
1278   004224  000407                  BR      3$              ;;LEAVE
1279   004226  010046          1$:     MOV     R0,-(SP)        ;;SAVE R0
1280   004230  017600  000002          MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
1281   004234  112046          2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
1282   004236  001005                  BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
1283   004240  005726                  TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
1284   004242  012600                  MOV     (SP)+,R0        ;;RESTORE R0
1285   004244  062716  000002  3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
1286   004250  000002                  RTI                     ;;RETURN
```

```
1287  004252  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
1288  004256  001424                          BEQ     8$
1289  004260  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT
1290  004264  001004                          BNE     5$
1291  004266  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
1292  004270  104400  004434                  TYPE    ,$CRLF
1293  004274  000757                          BR      2$              ;;GET NEXT CHARACTER
1294  004276  004767  000052          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
1295  004302  126726  000124          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
1296  004306  001352                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
1297  004310  016746  000114                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
1298                                                                  ;;AND THE NULL CHAR.
1299  004314  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
1300  004320  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
1301  004322  004767  000026                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
1302  004326  000772                          BR      7$              ;;LOOP
1303
1304                                  ;;HORIZONTAL TAB PROCESSOR
1305
1306  004330  112716  000040          8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
1307  004334  004767  000014          9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
1308  004340  132767  000007  000052          BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
1309  004346  001372                          BNE     9$              ;;TAB STOP
1310  004350  005726                          TST     (SP)+           ;;POP SPACE OFF STACK
1311  004352  000730                          BR      2$              ;;GET NEXT CHARACTER
1312  004354  105777  000044          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
1313  004360  100375                          BPL     $TYPEC
1314  004362  116677  000002  000036          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
1315  004370  122766  000015  000002          CMPB    #CR,2(SP)       ;;BRANCH IF
1316  004376  001003                          BNE     1$              ;;NOT <CR>
1317  004400  105067  000014                  CLRB    $CHARCNT        ;;
1318  004404  000406                          BR      $TYPEX          ;;EXIT
1319  004406  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;BRANCH IF
1320  004414  001402                          BEQ     $TYPEX          ;;<LF>
1321  004416  105227                          INCB    (PC)+           ;;INC SPACE
1322  004420  000000                  $CHARCNT:.WORD  0               ;;COUNT
1323  004422  000207                  $TYPEX: RTS     PC
1324
1325  004424  177564                  $TPS:   .WORD   177564          ;;TTY PRINTER STATUS REG. ADDRESS
1326  004426  177566                  $TPB:   .WORD   177566          ;;TTY PRINTER BUFFER REG. ADDRESS
1327  004430     000                  $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
1328  004431     002                  $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
1329  004432     012                  $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
1330  004433     000                  $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
1331  004434     015                  $CRLF:  .ASCII  <15>            ;;CARRAIGE RETURN
1332  004435     012     000          $LF:    .ASCIZ  <12>            ;;LINEFEED
1333        004440                           .EVEN
1334                                  ;;*****************************************************************
1335
1336                                  .SBTTL  TRAP DECODER
1337
1338                                  ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
1339                                  ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
1340                                  ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
1341                                  ;*GO TO THAT ROUTINE.
1342
```

```
1343   004440  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
1344   004442  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
1345   004446  005740                         TST     -(R0)           ;;BACKUP BY 2
1346   004450  111000                         MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
1347   004452  016000  004460                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
1348   004456  000200                         RTS     R0              ;;GO TO ROUTINE
1349
1350
1351                                  .SBTTL  TRAP TABLE
1352
1353                                  ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
1354                                  ;*BY THE "TRAP" INSTRUCTION.
1355
1356                                  ;       ROUTINE
1357                                  ;       -------
1358   004460                         $TRPAD:
1359   004460  004214                         $TYPE   ;;CALL=TYPE     TRAP+0(104400)  TTY TYPEOUT ROUTINE
1360   004462  004012                         $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
1361   004464  003766                         $TYPOS  ;;CALL=TYPOS    TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
1362   004466  004026                         $TYPON  ;;CALL=TYPON    TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST CALL)
1363   004470  000000                 END:    0
1364
1365           000001                  .END
```

DEKBGA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
BEGIN   001102            661     692#
BIT0  = 000001            366#
BIT00 = 000001            356#    366
BIT01 = 000002            355#    365     1103
BIT02 = 000004            354#    364
BIT03 = 000010            353#    363
BIT04 = 000020            352#    362     1168
BIT05 = 000040            351#    361
BIT06 = 000100            350#    360
BIT07 = 000200            349#    359
BIT08 = 000400            348#    358
BIT09 = 001000            347#    357
BIT1  = 000002            365#
BIT10 = 002000            346#
BIT11 = 004000            345#
BIT12 = 010000            344#
BIT13 = 020000            343#
BIT14 = 040000            342#    686
BIT15 = 100000            341#
BIT2  = 000004            364#
BIT3  = 000010            363#
BIT4  = 000020            362#
BIT5  = 000040            361#
BIT6  = 000100            360#
BIT7  = 000200            359#
BIT8  = 000400            358#
BIT9  = 001000            357#
BPTVEC= 000014            373#
CACHVE= 000114            380#    699*
CHECK   003254           1007    1010    1064#
CONTRL= 177746            390#
CPUERR= 177766            403#
CR    = 000015            278#    1315    1325
CRLF  = 000200            279#     720     724     736     740     1289    1325
DATA    003662           1046    1066    1157#
DISPLA= 177570            273#     745*    771*    791*    814*    828*    855*    870*    898*    913*    931*    947*    965*
                          981*    1006*
DVEC    003650            698*     836*    884*   1131*   1141*   1153#
EMTVEC= 000030            376#
END     004470           1048    1068    1095    1363#
ERRAD   003642           1115*   1150#
ERRVEC= 000004            369#     816*    821*    830*    838*    856*    863*    873*    879*    914*    922*    932*    940*
                          948*     956*    966*    974*    983*    994*
FACTOR  001100            690#    1126    1133
FLAG    003656            872*     880    1112*   1113*   1155#
GNS   = ****** U          657      719     723     731     735     739    1359    1360    1361    1362
HIADRS= 177742            388#    1093
HITMIS= 177752            392#
HT    = 000011            276#    1287    1325
ILLDWN  003636           1131    1147#
ILLUP   003632           1114    1144#
IOTVEC= 000020            374#
KDPAR0= 172360            546#
KDPAR1= 172362            547#
KDPAR2= 172364            548#
KDPAR3= 172366            549#
```

```
KDPAR4= 172370          550#
KDPAR5= 172372          551#
KDPAR6= 172374          552#
KDPAR7= 172376          553#
KDPDR0= 172320          524#
KDPDR1= 172322          525#
KDPDR2= 172324          526#
KDPDR3= 172326          527#
KDPDR4= 172330          528#
KDPDR5= 172332          529#
KDPDR6= 172334          530#
KDPDR7= 172336          531#
KERSTK= 001100          263#
KIPAR0= 172340          535#   1160*
KIPAR1= 172342          536#
KIPAR2= 172344          537#
KIPAR3= 172346          538#
KIPAR4= 172350          539#
KIPAR5= 172352          540#   1056*   1080*   1162*
KIPAR6= 172354          541#   1164*
KIPAR7= 172356          542#   1166*
KIPDR0= 172300          513#   1161*
KIPDR1= 172302          514#
KIPDR2= 172304          515#
KIPDR3= 172306          516#
KIPDR4= 172310          517#
KIPDR5= 172312          518#   1163*
KIPDR6= 172314          519#   1165*
KIPDR7= 172316          520#   1167*
KSP    =%000006         297#
LF     = 000012         277#    736    1319    1325
LIMIT    003660         709*   1045    1065    1156#
LOAD     003150        1004    1044#
LOADRS= 177740          387#   1095
MAINT = 177750          391#
MAP      003664         984    1044    1064    1160#
MAPH0 = 170202          630#
MAPH00= 170202          566#    630
MAPH01= 170206          568#    632
MAPH02= 170212          570#    634
MAPH03= 170216          572#    636
MAPH04= 170222          574#    638
MAPH05= 170226          576#    640
MAPH06= 170232          578#    642
MAPH07= 170236          580#    644
MAPH1 = 170206          632#
MAPH10= 170242          582#
MAPH11= 170246          584#
MAPH12= 170252          586#
MAPH13= 170256          588#
MAPH14= 170262          590#
MAPH15= 170266          592#
MAPH16= 170272          594#
MAPH17= 170276          596#
MAPH2 = 170212          634#
MAPH20= 170302          598#
```

```
MAPH21= 170306          600#
MAPH22= 170312          602#
MAPH23= 170316          604#
MAPH24= 170320          606#
MAPH25= 170326          608#
MAPH26= 170332          610#
MAPH27= 170336          612#
MAPH3 = 170216          636#
MAPH30= 170342          614#
MAPH31= 170346          616#
MAPH32= 170352          618#
MAPH33= 170356          620#
MAPH34= 170362          622#
MAPH35= 170366          624#
MAPH36= 170372          626#
MAPH37= 170376          628#
MAPH4 = 170222          638#
MAPH5 = 170226          640#
MAPH6 = 170232          642#
MAPH7 = 170236          644#
MAPL0 = 170200          629#
MAPL00= 170200          565#    629
MAPL01= 170204          567#    631
MAPL02= 170210          569#    633
MAPL03= 170214          571#    635
MAPL04= 170220          573#    637
MAPL05= 170224          575#    639
MAPL06= 170230          577#    641
MAPL07= 170234          579#    643
MAPL1 = 170204          631#
MAPL10= 170240          581#
MAPL11= 170244          583#
MAPL12= 170250          585#
MAPL13= 170254          587#
MAPL14= 170260          589#
MAPL15= 170264          591#
MAPL16= 170270          593#
MAPL17= 170274          595#
MAPL2 = 170210          633#
MAPL20= 170300          597#
MAPL21= 170304          599#
MAPL22= 170310          601#
MAPL23= 170314          603#
MAPL24= 170320          605#
MAPL25= 170324          607#
MAPL26= 170330          609#
MAPL27= 170334          611#
MAPL3 = 170214          635#
MAPL30= 170340          613#
MAPL31= 170344          615#
MAPL32= 170350          617#
MAPL33= 170354          619#
MAPL34= 170360          621#
MAPL35= 170364          623#
MAPL36= 170370          625#
MAPL37= 170374          627#
```

DEKBGA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
MAPL4 = 170220          637#
MAPL5 = 170224          639#
MAPL6 = 170230          641#
MAPL7 = 170234          643#
MEMERR= 177744          389#  1100  1106*
MMERR   003762         1169   1171#  1172
MMR0  = 177572          414#   418   987*  992*  1050*  1057*  1059*  1070*  1081*  1083*
MMR1  = 177574          415#   419
MMR2  = 177576          416#   420
MMR3  = 172516          417#   421  1168*
MMVEC = 000250          382#   985* 1047* 1067* 1169*
PARERR  003372          699   1088#
PARFLG  003374          700*  1089# 1107*
PC    =%000007          300#   984* 1004* 1007* 1010* 1026* 1029* 1035* 1044* 1061* 1064* 1085* 1088*
                       1170*  1294* 1301* 1307* 1321* 1323*
PIRQ  = 177772          271#
PIRQVE= 000240          381#
POWDWN  003472          698    836   844   884  1112#  1141
POWUP   003566         1128   1131#
PR0   = 000000          303#
PR1   = 000040          304#
PR2   = 000100          305#
PR3   = 000140          306#
PR4   = 000200          307#
PR5   = 000240          308#
PR6   = 000300          309#
PR7   = 000340          310#
PS    = 177776          268#   269   746*  772*  792*  815*  829*  859*  871*  899*  917*  919*  935*
                        937*   951*  953*  969*  971*  982* 1002*
PSW   = 177776          269#
PWRVEC= 000024          375#
RESVEC= 000010          370#
R0    =%000000          282#   290   704*  707*  708*  709   725   749*  750   754*  755   758*  759
                        776*   777   780*  781   796*  797   800*  801   839*  843*  860   936   970
                       1032*  1035  1048* 1049* 1051* 1055* 1068* 1069* 1071  1075* 1079* 1102* 1104*
                       1105*  1119  1126* 1127* 1133* 1134* 1140* 1279  1280* 1281  1284* 1343  1344*
                       1345   1346* 1347* 1348*
R1    =%000001          283#   291   705*  706*  1071*  1072  1120  1139*
R10   =%000000          290#
R11   =%000001          291#
R12   =%000002          292#
R13   =%000003          293#
R14   =%000004          294#
R15   =%000005          295#
R2    =%000002          284#   292  1003* 1011* 1121  1138*
R3    =%000003          285#   293   747*  773*  793*  817*  831*  857*  875*  886*  900*  915*  933*
                        949*   967*  986*  1005* 1105  1122  1137* 1142  1209  1218* 1224* 1225* 1228*
                       1233*  1234* 1235  1244*
R4    =%000004          286#   294  1045* 1053* 1065* 1077* 1123  1136* 1210  1212* 1213* 1214* 1215
                       1216*  1230 1232* 1240* 1243*
R5    =%000005          287#   295  1046* 1051  1066* 1072  1075  1103* 1104  1124  1135* 1211  1217*
                       1219*  1221* 1222* 1223* 1224  1242*
R6    =%000006          288#   296
R7    =%000007          289#   300
SAV6    003654         1125*  1132  1154#
SDPAR0= 172260          502#
```

DEKBGA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

```
SDPAR1= 172262          503#
SDPAR2= 172264          504#
SDPAR3= 172266          505#
SDPAR4= 172270          506#
SDPAR5= 172272          507#
SDPAR6= 172274          508#
SDPAR7= 172276          509#
SDPDR0= 172220          480#
SDPDR1= 172222          481#
SDPDR2= 172224          482#
SDPDR3= 172226          483#
SDPDR4= 172230          484#
SDPDR5= 172232          485#
SDPDR6= 172234          486#
SDPDR7= 172236          487#
SIPAR0= 172240          491#
SIPAR1= 172242          492#
SIPAR2= 172244          493#
SIPAR3= 172246          494#
SIPAR4= 172250          495#
SIPAR5= 172252          496#
SIPAR6= 172254          497#
SIPAR7= 172256          498#
SIPDR0= 172200          469#
SIPDR1= 172202          470#
SIPDR2= 172204          471#
SIPDR3= 172206          472#
SIPDR4= 172210          473#
SIPDR5= 172212          474#
SIPDR6= 172214          475#
SIPDR7= 172216          476#
SIZEHI= 177762          400#
SIZELO= 177760          398#    705
SP    =%000006          296#    297     298     299     693*    725*    749     753*    775*    795*    818*    832*    837*
                        858*    862*    874*    901*    906*    916*    921*    934*    939*    950*    955*    968*    973*
                        988*    993*   1009*   1060*   1084*   1115    1116   1118*   1125    1132*   1201*   1202    1203
                       1204*   1209*   1210*   1211*   1217    1242    1243    1244    1245*   1246*   1279*   1280    1281*
                       1283    1284    1285*   1287    1289    1291    1295   1297*   1299*   1306*   1310    1314    1315
                       1319    1343*   1344
SR0   = 177572          418#
SR1   = 177574          419#
SR2   = 177576          420#
SR3   = 172516          421#
SSP   =%000006          298#
STACK = 001100          262#    263     264     265     693     750     753     754     758     775     776     780     795
                        796     800     818     837     858     862     901     906     916     921     934     939     950
                        955     968     973     988     993    1009    1118
STKLMT= 177774          270#
SUPSTK= 000700          264#
SWR   = 177570          272#    273     763     785     805     822     847     864     890     907     923     941     957
                        975     995    1012
SW0   = 000001          338#
SW00  = 000001          328#    338
SW01  = 000002          327#    337
SW02  = 000004          326#    336
SW03  = 000010          325#    335
```

```
SW04  = 000020          324#    334
SW05  = 000040          323#    333
SW06  = 000100          322#    332
SW07  = 000200          321#    331
SW08  = 000400          320#    330
SW09  = 001000          319#    329
SW1   = 000002          337#
SW10  = 002000          318#
SW11  = 004000          317#
SW12  = 010000          316#
SW13  = 020000          315#
SW14  = 040000          314#    763     785     805     822     847     864     890     907     923     941     957     975
                        995    1012
SW15  = 100000          313#
SW2   = 000004          336#
SW3   = 000010          335#
SW4   = 000020          334#
SW5   = 000040          333#
SW6   = 000100          332#
SW7   = 000200          331#
SW8   = 000400          330#
SW9   = 001000          329#
SYSTID= 177764          402#
TBITVE= 000014          371#
TKVEC = 000060          378#
TPVEC = 000064          379#
TRAPVE= 000034          377#    694*    695*
TRTVEC= 000014          372#
TST1    001442          714     716     744#    764
TST10   002312          897#    908
TST11   002360          912#    924
TST12   002446          930#    942
TST13   002534          946#    958
TST14   002622          964#    976
TST15   002710          980#    996
TST16   003012         1001#   1013
TST2    001540          770#    786
TST3    001626          790#    806
TST4    001714          813#    823
TST5    001770          827#    848
TST6    002114          854#    865
TST7    002174          869#    891
TYPE  = 104400          717     721     729     733     737     1236    1292    1359#
TYPOC = 104402         1360#
TYPON = 104406         1362#
TYPOS = 104404          726    1361#
UDPAR0= 177660          458#
UDPAR1= 177662          459#
UDPAR2= 177664          460#
UDPAR3= 177666          461#
UDPAR4= 177670          462#
UDPAR5= 177672          463#
UDPAR6= 177674          464#
UDPAR7= 177676          465#
UDPDR0= 177620          436#
UDPDR1= 177622          437#
```

```
UDPDR2= 177624            438#
UDPDR3= 177626            439#
UDPDR4= 177630            440#
UDPDR5= 177632            441#
UDPDR6= 177634            442#
UDPDR7= 177636            443#
UIPAR0= 177640            447#
UIPAR1= 177642            448#
UIPAR2= 177644            449#
UIPAR3= 177646            450#
UIPAR4= 177650            451#
UIPAR5= 177652            452#
UIPAR6= 177654            453#
UIPAR7= 177656            454#
UIPDR0= 177600            425#
UIPDR1= 177602            426#
UIPDR2= 177604            427#
UIPDR3= 177606            428#
UIPDR4= 177610            429#
UIPDR5= 177612            430#
UIPDR6= 177614            431#
UIPDR7= 177616            432#
USESTK= 000600            265#
USP   =%000006            299#
UVEC    003644            834*    882*   1114*   1128*   1152#
$CHARC  004420           1308    1317*   1322#
$CMTAG= ****** U          693     696
$CRLF   004434           1292    1331#
$DOAGN  003142           1028    1033    1039#
$ENDAD  003132            684     715    1035#
$ENDCT  003116            697    1030#
$EOP    003072           1022#
$EOPCT  003110            697*   1027#   1031
$FILLC  004432           1295    1329#
$FILLS  004431           1328#
$GET42  003122           1032#
$HD   = 000000            248
$LF     004435           1332#
$NULL   004430           1297    1327#
$NWTST= 000001            741#    767#    787#    810#    824#    851#    866#    894#    909#    927#    943#    961#    977#
                          998#
$OCNT   004210           1208*   1237*   1250#
$OMODE  004212           1203*   1207*   1212    1215*   1226*   1252#
$PASS   003146            696*   1024*   1025*   1041#
$RDCHR= ****** U         1363
$RDDEC= ****** U         1363
$RDLIN= ****** U         1363
$RDOCT= ****** U         1363
$SAVRE= ****** U         1363
$SETUP= 000024             692#    694     696     698     715    1024
$STUP = 177777             692#
$SVPC = 000204             682#    687
$SWR  = 040000             238#    248     255     256     257     698     746     772     792     815     829     856     871
                           899     914     932     948     966     982    1002    1019    1024    1034    1040    1042
$TN   = 000017             238#    248     741     745     746#    763     767     771     772#    785     787     791     792#
                           805     810     814     815#    822     824     828     829#    847     851     855     856#    864
```

```
                        866     870     871*    890     894     898     899*    907     909     913     914*    923     927
                        931     932*    941     943     947     948*    957     961     965     966*    975     977     981
                        982*    995     998     1002*   1006    1012
$TPB    004426         1314*   1326*
$TPFLG  004433         1275    1330*
$TPS    004424         1312    1325*
$TRAP   004440          694    1343*
$TRP  = 000010         1350*   1360*   1361*   1362*   1363*
$TRPAD  004460         1347    1358*
$TYPBN= ****** U       1363
$TYPDS= ****** U       1363
$TYPE   004214         1275*   1350    1359
$TYPEC  004354         1294    1301    1307    1312*   1313
$TYPEX  004422         1318    1320    1323*
$TYPOC  004012         1206*   1360
$TYPON  004026         1205    1208*   1362
$TYPOS  003766         1201*   1361
$$GET4= 000000         1034*
$$TRP = 000002         1349*   1360    1361    1362    1363
$DFILL  004211         1202*   1206*   1216    1251*
.     = 004472          653*    657     659*    682     683*    685*    687*    688*    720*    732*    736*    751     756
                        760     778     782     798     802     841     845    1041    1042    1127    1134    1145    1148
                       1325    1326    1327    1328    1329    1330    1331    1332    1333*
```

```
COMMEN    648#
ENDCOM    648#
ERROR     266#
ESCAPE    648#
LOOP      258#   762   784   804   822   847   864   889   907   923   941   957   975   995  1012
LOOPX     258#   763   785   805   822   847   864   890   907   923   941   957   975   995  1012
MULT      648#
NEWTST    648#   741   767   787   810   824   851   866   894   909   927   943   961   977   998
POP       648#
PUSH      648#
SCOPE     267#
SETK      258#   746   815   829   859   871   899   919   937   953   971   982  1002
SETS      258#   772   917   935
SETTRA   1350#  1360  1361  1362
SETU      258#   792   951   969
SETUP     648#   692
SKIR      648#
SLASH     648#
SPACE     648#
STARS     648#   662   741   743   767   769   787   789   810   812   824   826   851   853   866
          868    894   896   909   911   927   929   943   945   961   963   977   979   998  1000
         1014   1175  1253  1334
TEST      258#   744   770   790   813   827   854   869   897   912   930   946   964   980  1006
TRMTRP   1350#
TSTNUM    258#   745   771   791   814   828   855   870   898   913   931   947   965   981  1006
TYPBIN    648#
TYPDEC    648#
TYPNAM    648#   713
TYPNUM    648#
TYPOCS    648#   725
TYPOCT    648#
TYPTXT    648#   721   729   733   737
$$ESCA    648#
$$NEWT    648#   741   767   787   810   824   851   866   894   909   927   943   961   977   998
$$SET    1350#  1360  1361  1362
$$SKIP    648#
.HEADE    238#
.SETUP    238#   692
.SWRHI    238#   250
.SWRLO    257#
.$ACT1    238#   662
.$CATC    238#   650
.$EOP     238#  1014
.$TRAP    238#  1334
.$TYPE    238#  1253
.$TYPO    238#  1175
.1170     238#   258
```

DEKBGA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

```
ADD     1049  1056  1060  1069  1080  1084  1204  1214  1285
BEQ      716   751   756   760   778   782   798   802   841   945  1033  1073  1231  1288  1320
BGE     1090
BGT     1028  1238
BHI     1096
BIC     1025  1228
BIS     1233  1234
BIT      763   785   805   822   847   864   890   907   923   941   957   975   995  1012  1093
BITB    1308
BLT     1239  1300
BMI     1054  1078  1101  1117
BNE      714   764   786   806   823   848   865   881   891   908   924   942   958   976   996
        1013  1094  1229  1282  1290  1296  1309  1316
BPL     1227  1276  1313
BR       718   722   730   734   738   878   885   905  1008  1052  1076  1092  1098  1145  1148
        1172  1205  1220  1241  1278  1293  1302  1311  1318
CLR      696   704   746   815   829   859   871   872   899   919   937   953   971   982   992
        1002  1059  1083  1113  1218
CLRB    1317
CMP      715   750   755   759   777   781   797   801   844  1072  1095  1116
CMPB    1287  1289  1295  1315  1319
DEC      708  1026  1053  1077  1088
DECB    1226  1237  1299
DIV      707
EMT      266
HALT     657   752   757   761   779   783   799   803   820   835   842   846   861   877   883
         888   920   938   954   972   990  1074  1091  1097  1099  1129  1144  1147  1171  1277
INC      706   713   987   989  1024  1232  1240
INCB    1321
IOT      267
JMP      661  1040  1142
JSR      984  1004  1007  1010  1035  1044  1064  1294  1301  1307
MOV      693   694   695   697   698   699   700   705   709   725   745   747   749   753   754
         758   771   772   773   775   776   780   791   792   793   795   796   800   814   816
         817   818   821   828   830   831   832   834   836   837   838   839   843   855   856
         857   858   860   862   863   870   873   874   875   879   882   884   886   898   900
         901   906   913   914   915   916   917   921   922   931   932   933   934   935   936
         939   940   947   948   949   950   951   955   956   965   966   967   968   969   970
         973   974   981   983   985   986   988   993   994  1003  1005  1006  1009  1029  1032
        1045  1046  1047  1048  1050  1051  1055  1057  1065  1066  1067  1068  1070  1071  1075
        1079  1081  1103  1105  1106  1107  1112  1114  1115  1118  1119  1120  1121  1122  1123
        1124  1125  1126  1128  1131  1132  1133  1135  1136  1137  1138  1139  1140  1141  1160
        1161  1162  1163  1164  1165  1166  1167  1168  1169  1201  1209  1210  1211  1217  1224
        1242  1243  1244  1245  1246  1279  1280  1284  1297  1343  1344  1347
MOVB    1202  1203  1206  1207  1208  1212  1215  1216  1235  1281  1306  1314  1346
NEG     1213
NOP     1001  1023  1036  1037  1038
RESET    902   903   904  1034
ROL     1219  1221  1222  1223  1225
RTI      887  1058  1082  1108  1247  1286
RTS     1061  1085  1170  1323  1348
SOB     1011  1127  1134
SUB     1102
TRAP    1350  1360  1361  1362
TST      819   840   880   918   952  1100  1230  1283  1291  1310  1345
TSTB    1275  1312
```

```
WAIT     748   774   794   833   876
XOR     1104
.ASCII  1331
.ASCIZ   720   724   732   736   740  1332
.BYTE    727   728  1248  1249  1250  1251  1327  1328  1329  1330
.ENABL   238
.END    1365
.ENDC    243   256   257   258   647   662   663   685   687   692   693   694   696   698   715
         717   720   724   728   729   732   736   740   742   743   744   746   768   769   770
         772   788   789   790   792   811   812   813   815   825   826   827   829   852   853
         854   856   867   868   869   871   895   896   897   899   910   911   912   914   928
         929   930   932   944   945   946   948   962   963   964   966   978   979   980   982
         999  1000  1001  1002  1015  1018  1019  1021  1024  1028  1031  1032  1034  1040  1041
        1042  1176  1254  1335  1344  1347  1349  1359  1360  1361  1362  1363
.EQUIV   266   267   269   290   291   292   293   294   295   296   297   298   299   300   329
         330   331   332   333   334   335   336   337   338   357   358   359   360   361   362
         363   364   365   366   418   419   420   421   629   630   631   632   633   634   635
         636   637   638   639   640   641   642   643   644
.EVEN    720   724   732   736   740  1333
.IF      239   256   257   258   647   657   662   684   686   692   693   694   696   698   714
         715   716   719   723   727   728   731   735   739   741   743   746   767   769   772
         787   789   792   810   812   815   824   826   829   851   853   856   866   868   871
         894   896   899   909   911   914   927   929   932   943   945   948   961   963   966
         977   979   982   998  1000  1002  1014  1018  1019  1020  1021  1023  1027  1030  1032
        1034  1040  1042  1175  1253  1334  1343  1347  1349  1350  1360  1361  1362  1363
.IFF     256   257   663   685   686   693   715   717   727   728   741   742   743   744   746
         767   768   769   770   772   787   788   789   790   792   810   811   812   813   815
         824   825   826   827   829   851   852   853   854   856   866   867   868   869   871
         894   895   896   897   899   909   910   911   912   914   927   928   929   930   932
         943   944   945   946   948   961   962   963   964   966   977   978   979   980   982
         998   999  1000  1001  1002  1015  1021  1023  1028  1031  1040  1176  1254  1335  1344
.IFT     720   724   732   736   740
.IFTF    720   724   732   736   740
.IIF     238   243   248   255   256   657   694   696   715   726  1024  1041  1042  1325  1326
        1327  1328  1329  1330  1331  1332  1333  1359  1360  1361  1362
.IRP     692   741   767   787   810   824   851   866   894   909   927   943   961   977   998
        1023
.LIST    238   648   657   692   715   720   724   732   736   740   741   746   767   772   787
         792   810   815   824   829   851   856   866   871   894   899   909   914   927   932
         943   948   961   966   977   982   998  1002  1024  1034  1349  1350  1359  1360  1361
        1362  1363
.MACRO   257   258  1350
.MCALL   238   648
.NLIST   238   648   657   692   715   720   724   732   736   740   741   746   767   772   787
         792   810   815   824   829   851   856   866   871   894   899   909   914   927   932
         943   948   961   966   977   982   998  1002  1024  1034  1349  1350  1359  1360  1361
        1362  1363
.PAGE    650   688   741   787   824   866   909   943   977  1014  1042  1062  1086  1109  1158
        1175
.REM        1
.REPT    657
.SBTTL   251   259   384   395   409   558   651   658   664   741   767   787   810   824   851
         866   894   909   927   943   961   977   998  1016  1043  1062  1086  1110  1158  1177
        1255  1336  1351
.TITLE   238
.WORD    657   684   686   690  1027  1030  1041  1089  1252  1322  1325  1326
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DEKBGA.SEQ/SOL/CRF/PAGNUM=DEKBGA
RUN-TIME: 25 13 3 SECONDS
RUN-TIME RATIO: 162/42=3.8
CORE USED:  15K  (29 PAGES)
```

G04

Spooler runtime 5 Seconds, 26 KCS, 142 disk reads, 0 disk writes, 44 pages