PDP-11/45

POWER FAIL TEST

MD-11-DCKBP-B

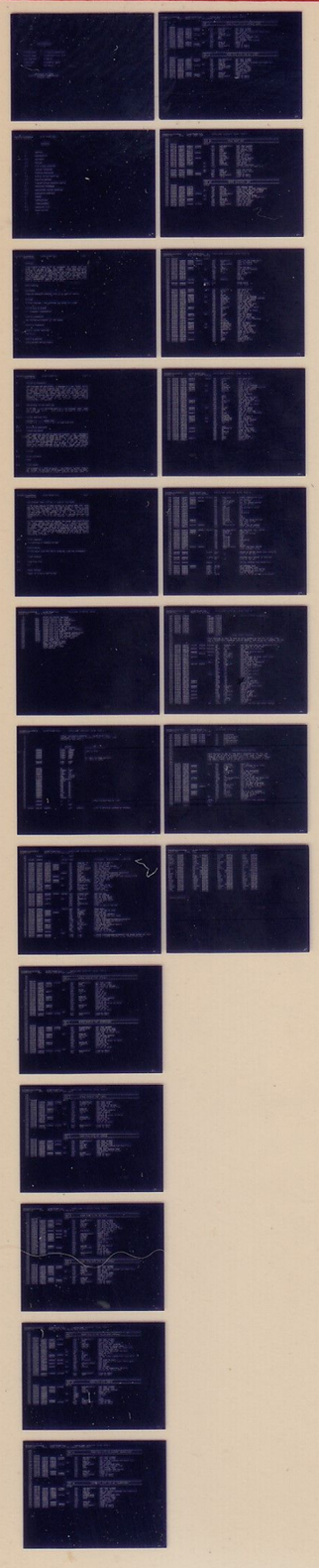
EP-DCKBP-B-DL

FICHE 1 OF 1

MAY 1978

digital

MADE IN USA



IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DCKBP-B-D
PRODUCT NAME:	11/45 POWER FAIL TEST
DATE CREATED:	1-NOV-72
MAINTAINER:	DIAGNOSTIC GROUP
AUTHORS:	BOB BRAIN

CONTENTS

1, Abstract

This program is made up of 16 subtests to check out the power fail on the 11/45, The 2 msec, power down and power up time is checked on each power fail, A constant has to be changed for use in BIPOLAR or MOS memories, 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 memory (up to 124K),

2, Requirements

2,1 Equipment

PDP11/45 standard computer with up to 124K of memory,

2,2 Storage

Program Storage - the routines use memory 0 - 4100

2,3 Preliminary programs

All processor diagnostics

3, Loading procedure

Use standard procedure for ABS tapes,

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, KT11C if it exists, 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 a bell rings and/or the count recycles to 1, Each subtest is executed once except 16 which runs 8 times before continuing, SW14 loops on the current test and SW10 inhibits bell at the end,

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
sw<10> = 1 ,,,,, inhibit bell on pass complete

5,2 Subroutine Abstracts

5,2,1 POWDWN and POWUP

These routines are use 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 1000 contains the timing factor for each memory, It is set initially for core and should be changed if 0 = 4K is MOS or BIPOLAR, Control is returned to the program via, JMP (3) so the power fail return address is put in R3, ILLUP and ILLDWN are use 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 and it is above 28K, 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 falling address, examine KIPAR6 (772354) and use that for the offset to the address in R0, To do this, move KIPAR6<11:0> into bits <17:6> of a zeroed word 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, LOC 1000 initially contains the timing factor for core memory, This must be changed to fit the type of memory you have from 0 - 4K, The address of the power failed routine is in ERROR,

6,3 Error recovery

Hit continue or Restart at 200

7, Restrictions

Do not power down the MOS or BIPOLAR, just the processor',

8, Miscellaneous

8,1 Execution time

N/A

8,2 Stack Pointer

Stack is initally set to 500

```
                                ;TITLE  MAINDEC-11-DCKBP-B     11/45 POWER FAIL
                                ;COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
                                ;PROGRAM BY BOB BRAIN



                                ;    SWITCH                 USE
                                ;    ------            ------------------
          100000                SW15=   100000        ;
          040000                SW14=   40000         ;LOOP ON TEST
          020000                SW13=   20000         ;
          010000                SW12=   10000         ;
          004000                SW11=   4000          ;
          002000                SW10=   2000          ;0 - BELL ON PASS COMPLETE
                                                      ;1 - BELL ON ERROR
          001000                SW9=    1000          ;
          000400                SW8=    400           ;


          000001                N=      1
          104400                SCOPE=  TRAP
          104000                HLT=    EMT
          000004                TYPE=   IOT
          177776                PS=     177776
          177570                SWR=    177570
          177570                DISPLAY=SWR
          000007                BELL=   7
          000000                R0=     %0
          000001                R1=     %1
          000002                R2=     %2
          000003                R3=     %3
          000004                R4=     %4
          000005                R5=     %5
          000005                TTY=    %5
          000006                SP=     %6
          000007                PC=     %7
          000240                SCOPE=  NOP

          000000                .=      0                       ;TRAP CATCHER FROM 0 - 776

          000200                .=      200

000200    000137  001002                JMP     @#BEGIN         ;JUMP TO BEGINING ADDRESS OF PROGRAM
```

```
                001000  -               .=      1000
                
001000  002000          FACTOR: 2000                    ;CORE=2000  ;BIPOLAR=3300  ;MOS=1700
                
001002  012706  000500  BEGIN:  MOV     #500,SP         ;** STACK AT 500 **
001006  012737  004172  000020          MOV     #IOTS,@#20      ;SET IOT VECTOR
001014  012737  000006  000016          MOV     #6,@#16         ;SET TRACE TRAP RETURN
001022  012777  003512  002640          MOV     #POWDWN,@DVEC   ;SET UP POWER DOWN VECTOR
001030  012737  001120  000004          MOV     #DOCORE,@#4     ;SET FOR TIMEOUT
001036  005777  002642                  TST     @SR0            ;CHECK FOR KT11
001042  004767  003056  DOSEG:  JSR     PC,MAP          ;SETUP MEMORY MANAGEMENT REGISTERS
001046  005277  002632                  INC     @SR0            ;TURN ON MEMORY MANAGEMENT
001052  012737  001102  000004          MOV     #2$,@#4         ;SET TIMEOUT ADDRESS FOR CORE CALCULATIONS
001060  005737  157776  1$:     TST     @#157776        ;TRAP ON NON EX MEM
001064  062777  000200  002620          ADD     #200,@KIPAR6    ;GO TO NEXT BANK
001072  022777  007600  002612          CMP     #7600,@KIPAR6   ;LAST ONE?
001100  003367                          BGT     1$              ;TRY NEXT
001102  017701  002604  2$:     MOV     @KIPAR6,R1      ;SAVE ASR6 IN R1
001106  072127  177771                  ASH     #-7,R1          ;PUT INTO POSITION
001112  042701  177740                  BIC     #177740,R1      ;CLEAR JUNK
001116  000415                          BR      TYPEIT          ;TYPE THE NAME
                
001120  012737  001152  000004  DOCORE: MOV     #TYPEIT,@#4     ;SET FOR NEM
001126  012702  017776                  MOV     #17776,R2       ;SET UP ADDRESS
001132  005001                          CLR     R1              ;SET UP BANK COUNT
001134  062702  020000          1$:     ADD     #20000,R2       ;MOVE TO NEXT BANK
001140  005201                          INC     R1              ;INC THE BANK COUNT
001142  005712                          TST     (2)             ;TIMEOUT?
001144  022702  177776                  CMP     #177776,R2      ;END?
001150  001371                          BNE     1$              ;LOOP IF NOT AT THE END
                
001152  005301  TYPEIT: DEC     R1              ;DROP BACK
001154  010167  002520                  MOV     R1,LIMIT        ;SAVE THE TOP OF CORE
001160  012737  000006  000004          MOV     #6,@#4          ;SET FOR NEM
001166  012706  000500                  MOV     #500,SP         ;CLEAR STACK **500**
001172  005227  177777                  INC     #-1             ;TYPE THE OPTION ONLY ONCE
001176  001122                          BNE     TST1            ;FIRST TIME?
001200  000004  001204                  TYPE    ,,.+2           ;,ASCIZ <15><12>"MAINDEC-11-DCKBP-B"
001232  000004  001236                  TYPE    ,,.+2           ;,ASCIZ <15><12><12>"BANKS 0 - "
001254  010105                          MOV     R1,TTY          ;TYPE R1 IN OCTAL
001256  004767  002470                  JSR     PC,PRINTS       ;AND SUPRESS LEADING ZERO'S
001262  000004  001266                  TYPE    ,,.+2           ;,ASCIZ " EXIST"
001276  022701  000006                  CMP     #6,R1           ;WHICH OPTION?
001302  100010                          BPL     MES             ;SKIP IF NO KT11C
001304  000004  001310                  TYPE    ,,.+2           ;,ASCIZ " WITH KT11C"
001324                          MES:
001324  000004  001330                  TYPE    ,,.+2           ;,ASCIZ <15><12><12>"INTERRUPT THE POWER AFTER THE TEST"
001376  000004  001402                  TYPE    ,,.+2           ;,ASCIZ " NUMBER APPEARS IN THE DISPLAY"<15><12>
```

```
                                        ;*****************************************************************
                                        ;TEST 1          SIMPLE DOWN/UP TEST (KERNAL)
                                        ;*****************************************************************
001444                                  TST1:
001444  012737  000001  177570                  MOV     #1,@#DISPLAY    ;SET TEST NUMBER
001452  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
001456  012703  001464                  MOV     #2$,R3          ;SET POWER UP RETURN
001462  000001                          WAIT                    ;WAIT FOR THE POWER FAIL
001464  010600                  2$:     MOV     SP,R0           ;GET SP
001466  022700  000474                  CMP     #474,R0         ;CHECK SP
001472  001401                          BEQ     .+4             ;SKIP IF OK
001474  000000                          HALT                    ;SP NOT 474
001476  012706  000500                  MOV     #500,SP         ;RESET SP
001502  013700  000474                  MOV     @#474,R0        ;GET RETURN ADDRESS
001506  022700  001464                  CMP     #2$,R0          ;CHECK ADDRESS
001512  001401                          BEQ     .+4             ;SKIP IF OK
001514  000000                          HALT                    ;ADDRESS ON STACK IS WRONG
001516  013700  000476                  MOV     @#476,R0        ;GET OLD PS
001522  022700  000000                  CMP     #0,R0           ;CHECK OLD PS
001526  001401                          BEQ     .+4             ;SKIP IF OK
001530  000000                          HALT                    ;OLD PS IS WRONG
001532                                  1$:
001532  032737  040000  177570                  BIT     #SW14,@#SWR     ;LOOP ON TEST?
001540  001341                          BNE     TST1            ;LOOP TO TST1

                                        ;*****************************************************************
                                        ;TEST 2          SIMPLE DOWN/UP TEST (SUPERVISOR)
                                        ;*****************************************************************
001542                                  TST2:
001542  012737  000002  177570                  MOV     #2,@#DISPLAY    ;SET TEST NUMBER
001550  012737  040000  177776                  MOV     #40000,@#PS     ;SET SUPERVISOR MODE
001556  012703  001564                  MOV     #2$,R3          ;SET POWER UP RETURN
001562  000001                          WAIT                    ;WAIT FOR THE POWER FAIL
001564  012706  000500                  2$:     MOV     #500,SP         ;RESET SP
001570  013700  000474                  MOV     @#474,R0        ;GET RETURN ADDRESS
001574  022700  001564                  CMP     #2$,R0          ;CHECK ADDRESS
001600  001401                          BEQ     .+4             ;SKIP IF OK
001602  000000                          HALT                    ;ADDRESS ON STACK IS WRONG
001604  013700  000476                  MOV     @#476,R0        ;GET OLD PS
001610  022700  040000                  CMP     #40000,R0       ;CHECK OLD PS
001614  001401                          BEQ     .+4             ;SKIP IF OK
001616  000000                          HALT                    ;OLD PS IS WRONG
001620                                  1$:
001620  032737  040000  177570                  BIT     #SW14,@#SWR     ;LOOP ON TEST?
001626  001345                          BNE     TST2            ;LOOP TO TST2
```

```
                                ;*****************************************************************
                                ;TEST 3          SIMPLE DOWN/UP TEST (USER)
                                ;*****************************************************************
001630                          TST3:
001630  012737  000003  177570          MOV     #3,@#DISPLAY    ;SET TEST NUMBER
001636  012737  140000  177776          MOV     #140000,@#PS    ;SET USER MODE
001644  012703  001652                  MOV     #2$,R3          ;SET POWER UP RETURN
001650  000001                          WAIT                    ;WAIT FOR THE POWER FAIL
001652  012706  000500          2$:     MOV     #500,SP         ;RESET SP
001656  013700  000474                  MOV     @#474,R0        ;GET RETURN ADDRESS
001662  022700  001652                  CMP     #2$,R0          ;CHECK ADDRESS
001666  001401                          BEQ     .+4             ;SKIP IF OK
001670  000000                          HALT                    ;ADDRESS ON STACK IS WRONG
001672  013700  000476                  MOV     @#476,R0        ;GET OLD PS
001676  022700  140000                  CMP     #140000,R0      ;CHECK OLD PS
001702  001401                          BEQ     .+4             ;SKIP IF OK
001704  000000                          HALT                    ;OLD PS IS WRONG
001706                          1$:
001706  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
001714  001345                          BNE     TST3            ;LOOP TO TST3


                                ;*****************************************************************
                                ;TEST 4          POWER FAIL WITH ODD ADDRESS
                                ;*****************************************************************
001716                          TST4:
001716  012737  000004  177570          MOV     #4,@#DISPLAY    ;SET TEST NUMBER
001724  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
001730  012737  001742  000004          MOV     #3$,@#4         ;SET TRAP VECTOR
001736  012703  001754                  MOV     #1$,R3          ;SET RETURN ADDRESS FOR POWER FAIL
001742  012706  000500          3$:     MOV     #500,SP         ;RESET STACK
001746  005737  000003                  TST     @#3             ;CAUSE ODD ADDRESS TRAP
001752  000000                          HALT                    ;ODD ADDRESS TRAP FAILED
001754  012737  000006  000004  1$:     MOV     #6,@#4          ;RESET 4
001762  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
001770  001352                          BNE     TST4            ;LOOP TO TST4
```

```
                                        ;*****************************************************************
                                        ;TEST 5          POWER FAIL IN THE RED ZONE
                                        ;*****************************************************************
001772                                  TST5:
001772  012737  000005  177570          MOV     #5,@#DISPLAY    ;SET TEST NUMBER
002000  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002004  012737  002024  000004          MOV     #2$,@#4         ;SET TRAP REGISTER
002012  012703  002042                  MOV     #1$,R3          ;SET POWER UP RETURN
002016  012706  000002                  MOV     #2,SP           ;SET STACK TO RED ZONE
002022  000001                          WAIT                    ;WAIT FOR POWER FAIL TRAP
002024  012777  002032  001632  2$:     MOV     #7$,@UVEC       ;SET UVEC TO HALT
002032  000000          7$:     HALT                    ;ILLEGAL TRAP TO 4
002034  012777  003512  001626          MOV     #POWDWN,@DVEC   ;RESET DVEC
002042  012706  000500          1$:     MOV     #500,SP         ;RESET STACK
002046  012737  000006  000004          MOV     #6,@#4          ;RESET 4
002054  013700  000002                  MOV     @#2,R0          ;GET FOR TYPING
002060  005737  000002                  TST     @#2             ;IS 2 OK?
002064  001401                          BEQ     .+4             ;SKIP IF OK
002066  000000                          HALT                    ;NO!
002070  013700  000000                  MOV     @#0,R0          ;GET FOR TYPING
002074  022737  003512  000000          CMP     #POWDWN,@#0     ;IS 0 OK?
002102  001401                          BEQ     .+4             ;SKIP IF OK
002104  000000                          HALT                    ;0 IS WRONG!
002106  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002114  001326                          BNE     TST5            ;LOOP TO TST5

                                        ;*****************************************************************
                                        ;TEST 6          POWER FAIL WITH TIME OUT (KERNAL)
                                        ;*****************************************************************
002116                                  TST6:
002116  012737  000006  177570          MOV     #6,@#DISPLAY    ;SET TEST NUMBER
002124  012737  002136  000004          MOV     #3$,@#4         ;SET TRAP VECTOR
002132  012703  002154                  MOV     #1$,R3          ;SET UP RETURN ADDRESS FOR POWER FAIL
002136  012706  000500          3$:     MOV     #500,SP         ;SET STACK
002142  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002146  010037  173000                  MOV     R0,@#173000     ;CAUSE A TIMEOUT
002152  000000                          HALT                    ;TIMEOUT FAILED
002154  012706  000500          1$:     MOV     #500,SP         ;SET STACK
002160  012737  000006  000004          MOV     #6,@#4          ;RESET 4
002166  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002174  001350                          BNE     TST6            ;LOOP TO TST6
```

```
                                ;*****************************************************************
                                ;TEST 7          POWER FAIL IN THE YELLOW ZONE (KERNAL)
                                ;*****************************************************************
002176                          TST7:
002176  012737  000007  177570          MOV     #7,@#DISPLAY    ;SET TEST NUMBER
002204  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002210  005067  001462                  CLR     FLAG            ;CLEAR THE FLAG
002214  012737  002240  000004          MOV     #2$,@#4         ;SET SICK TRAP ADDRESS
002222  012706  000400                  MOV     #400,SP         ;SET STACK TO YELLOW ZONE
002226  012703  002234                  MOV     #1$,R3          ;SET RETURN ADDRESS FOR POWER FAIL
002232  000001                          WAIT                    ;WAIT FOR POWER FAIL
002234  000000                  1$:     HALT                    ;POWER FAIL RETURNED TOO SOON
002236  000422                          BR      4$              ;SKIP SP CHECK
002240  012737  000006  000004  2$:     MOV     #6,@#4          ;RESET 4
002246  005767  001424                  TST     FLAG            ;IS THE FIRST INSTRUCTION FLAG SET?
002252  001010                          BNE     5$              ;YES
002254  012777  002262  001402          MOV     #7$,@UVEC       ;SET UVEC TO HALT
002262  000000                  7$:     HALT                    ;NOT ENOUGH OR TOO MANY INSTR. EXEC.
002264  012777  003512  001376          MOV     #POWDWN,@DVEC   ;SET DVEC
002272  000404                          BR      4$              ;GET OUT
002274  012703  002304          5$:     MOV     #4$,R3          ;SET RETURN
002300  000002                          RTI                     ;GO TO THE POWER FAIL ROUTINE
002302  000000                          HALT                    ;SHOULD NOT RETURN HERE
002304                          4$:
002304  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002312  001331                          BNE     TST7            ;LOOP TO TST7

                                ;*****************************************************************
                                ;TEST 10                 POWER FAIL WITH RESETS
                                ;*****************************************************************
002314                          TST10:
002314  012737  000010  177570          MOV     #10,@#DISPLAY   ;SET TEST NUMBER
002322  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002326  012703  002346                  MOV     #1$,R3          ;SET RETURN ADDRESS
002332  012706  000500                  MOV     #500,SP         ;RESET STACK
002336  000005          3$:     RESET                   ;RESETS
002340  000005                  RESET                   ;TO WAIT
002342  000005                  RESET                   ;IN
002344  000774                  BR      3$              ;LOOP
002346  012706  000500          1$:     MOV     #500,SP         ;RESET STACK
002352  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002360  001355                          BNE     TST10           ;LOOP TO TST10
```

```
                                ;*****************************************************************
                                ;TEST 11                POWER FAIL WITH ODD ADDRESS (SUPERVISOR)
                                ;*****************************************************************
002362                          TST11:
002362  012737  000011  177570          MOV     #11,@#DISPLAY   ;SET TEST NUMBER
002370  012737  002402  000004          MOV     #3$,@#4         ;SET TRAP VECTOR
002376  012703  002426                  MOV     #1$,R3          ;SET RETURN ADDRESS FOR POWER FAIL
002402  012706  000500          3$:     MOV     #500,SP         ;RESET STACK
002406  012737  040000  177776          MOV     #40000,@#PS     ;SET SUPERVISOR MODE
002414  005737  000003                  TST     @#3             ;CAUSE ODD ADDRESS TRAP
002420  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002424  000000                          HALT                    ;ODD ADDRESS TRAP FAILED
002426  012706  000500          1$:     MOV     #500,SP         ;RESET STACK POINTER
002432  012737  000006  000004          MOV     #6,@#4          ;RESET 4
002440  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002446  001345                          BNE     TST11           ;LOOP TO TST11

                                ;*****************************************************************
                                ;TEST 12                POWER FAIL WITH TIME OUT (SUPERVISOR)
                                ;*****************************************************************
002450                          TST12:
002450  012737  000012  177570          MOV     #12,@#DISPLAY   ;SET TEST NUMBER
002456  012737  002470  000004          MOV     #3$,@#4         ;SET TRAP VECTOR
002464  012703  002514                  MOV     #1$,R3          ;SET UP RETURN ADDRESS FOR POWER FAIL
002470  012706  000500          3$:     MOV     #500,SP         ;RESET STACK
002474  012737  040000  177776          MOV     #40000,@#PS     ;SET SUPERVISOR MODE
002502  010037  173000                  MOV     R0,@#173000     ;CAUSE A TIMEOUT
002506  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002512  000000                          HALT                    ;TIMEOUT FAILED
002514  012706  000500          1$:     MOV     #500,SP         ;RESET STACK
002520  012737  000006  000004          MOV     #6,@#4          ;RESET 4
002526  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002534  001345                          BNE     TST12           ;LOOP TO TST12
```

```
                                ;*****************************************************************
                                ;TEST 13                 POWER FAIL WITH ODD ADDRESS (USER)
                                ;*****************************************************************
002536                          TST13:
002536  012737  000013  177570          MOV     #13,@#DISPLAY   ;SET TEST NUMBER
002544  012737  002556  000004          MOV     #3$,@#4         ;SET TRAP VECTOR
002552  012703  002602                  MOV     #1$,R3          ;SET RETURN ADDRESS FOR POWER FAIL
002556  012706  000500          3$:     MOV     #500,SP         ;RESET STACK
002562  012737  140000  177776          MOV     #140000,@#PS    ;SET USER MODE
002570  005737  000003                  TST     @#3             ;CAUSE ODD ADDRESS TRAP
002574  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002600  000000                          HALT                    ;ODD ADDRESS TRAP FAILED
002602  012706  000500          1$:     MOV     #500,SP         ;RESET SP
002606  012737  000006  000004          MOV     #6,@#4          ;RESET 4
002614  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002622  001345                          BNE     TST13           ;LOOP TO TST13

                                ;*****************************************************************
                                ;TEST 14                 POWER FAIL WITH TIME OUT (USER)
                                ;*****************************************************************
002624                          TST14:
002624  012737  000014  177570          MOV     #14,@#DISPLAY   ;SET TEST NUMBER
002632  012737  002644  000004          MOV     #3$,@#4         ;SET TRAP VECTOR
002640  012703  002670                  MOV     #1$,R3          ;SET UP RETURN ADDRESS FOR POWER FAIL
002644  012706  000500          3$:     MOV     #500,SP         ;RESET STACK
002650  012737  140000  177776          MOV     #140000,@#PS    ;SET USER MODE
002656  010037  173000                  MOV     R0,@#173000     ;CAUSE A TIMEOUT
002662  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002666  000000                          HALT                    ;TIMEOUT FAILED
002670  012706  000500          1$:     MOV     #500,SP         ;RESET STACK
002674  012737  000006  000004          MOV     #6,@#4          ;RESET 4
002702  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
002710  001345                          BNE     TST14           ;LOOP TO TST14
```

```
                                ;*****************************************************************
                                ;TEST 15                KT11C ABORT TEST
                                ;*****************************************************************
002712                          TST15:
002712  012737  000015  177570          MOV     #15,@#DISPLAY   ;SET TEST NUMBER
002720  012737  003010  000004          MOV     #2$,@#4         ;SET FOR TIMEOUT
002726  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
002732  005777  000746                  TST     @SR0            ;IS THERE KT11C?
002736  012737  003002  000004          MOV     #4$,@#4         ;RESET 4
002744  004767  001154                  JSR     PC,MAP          ;MAP THE WORLD
002750  005077  000740                  CLR     @KIPDR6         ;MAP U6 TO 6
002754  012737  002772  000250          MOV     #3$,@#250       ;SET KT11C VECTOR
002762  012703  003004                  MOV     #1$,R3          ;LOAD PF RETURN
002766  005277  000712                  INC     @SR0            ;TURN KT11C ON
002772  012706  000500          3$:     MOV     #500,SP         ;ZAP STACK
002776  005237  140000                  INC     @#140000        ;ACCESS VIOLATION
003002  000000          4$:     HALT                            ;NO VIOLATION OR TRAP TO 4

003004  005077  000674          1$:     CLR     @SR0            ;TURN OFF KT11C
003010  012706  000500          2$:     MOV     #500,SP         ;MAKE A NEW STACK
003014  012737  000006  000004          MOV     #6,@#4          ;RESET 4
003022  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
003030  001330                  BNE     TST15                   ;LOOP TO TST15

                                ;*****************************************************************
                                ;TEST 16                MEMORY VOLATILITY TEST
                                ;*****************************************************************
003032                          TST16:
003032  005037  177776                  CLR     @#PS            ;SET KERNAL MODE
003036  012702  000010                  MOV     #10,R2          ;LOAD COUNT OF TEST ITERATIONS
003042  004767  000120          4$:     JSR     PC,LOAD         ;LOAD ALL MEMORY WITH 52525
003046  012703  003066                  MOV     #1$,R3          ;POWER FAIL RETURN ADDRESS
003052  012737  000016  177570          MOV     #16,@#DISPLAY   ;SET TEST NUMBER
003060  004767  000246          2$:     JSR     PC,CHECK        ;CHECK FOR THE 52525
003064  000775                  BR      2$                      ;LOOP FOR EVER OR POWER FAIL
003066  012706  000500          1$:     MOV     #500,SP         ;ZAP THE STACK
003072  004767  000234                  JSR     PC,CHECK        ;CHECK ALL MEMORY
003076  077217                  SOB     R2,4$                   ;DO IT 10 TIMES
003100  032737  040000  177570          BIT     #SW14,@#SWR     ;LOOP ON TEST?
003106  001351                  BNE     TST16                   ;LOOP TO TST16
```

```
                                        DONE:
003110  062767  000001  200044          ADD     #1,PCNT+2       ;ADD 1 TO THE PASS COUNT
003116  005567  000036                  ADC     PCNT            ;MAKE IT DOUBLE PREC.
003122  032737  002000  177570          BIT     #SW10,@#SWR     ;RING THE BELL?
003130  001002                          BNE     4$              ;NO!
003132  000004  000007                  TYPE    ,BELL           ;RING THE BELL
003136  013700  000042          4$:     MOV     @#42,R0         ;GET MONITOR ADDRESS
003142  001404                          BEQ     3$              ;IF NONE
003144  004710                          JSR     7,(0)           ;GO TO MONITOR
003146  000240                          NOP                     ;SAVE ROOM
003150  000240                          NOP                     ;FOR
003152  000240                          NOP                     ;ACT11
003154  000137  001002          3$:     JMP     @#BEGIN         ;RETURN

003160  000000  000000          PCNT:   0,0                     ;PASS COUNT
003164  000000                  .TBIT:  0                       ;T BIT FLAG


003166  016704  000506          LOAD:   MOV     LIMIT,R4        ;GET BANK COUNT
003172  022704  000006                  CMP     #6,R4           ;IS IT > 6?
003176  100002                          BPL     1$              ;SKIP IF > 6
003200  012704  000006                  MOV     #6,R4           ;FUDGE IN A 6
003204  072427  000015          1$:     ASH     #13.,R4         ;MAKE IT AN ADDRESS
003210  062704  017500                  ADD     #17500,R4       ;MAKE IT ABS LOADER ADDRESS
003214  012700  004300                  MOV     #END,R0         ;LOAD LAST ADDRESS
003220  016720  000456          2$:     MOV     DATA,(0)+       ;LOAD THE DATA
003224  020004                          CMP     R0,R4           ;IS IT THE END YET?
003226  001374                          BNE     2$              ;LOOP UNTIL DONE
003230  016704  000444                  MOV     LIMIT,R4        ;GET BANK COUNT AGAIN
003234  022704  000006                  CMP     #6,R4           ;CHECK AGAIN
003240  100401                          BMI     3$              ;YES - SKIP IF KT11C
003242  000207                          RTS     PC              ;NO - EXIT
003244  004767  000654          3$:     JSR     PC,MAP          ;MAP THE WORLD
003250  005277  000430                  INC     @#SR0           ;TURN ON KT11C
003254  005204                          INC     R4              ;GET TO RIGHT ONE
003256  072427  000007                  ASH     #7,R4           ;SHIFT IT INTO POSITION
003262  010446                          MOV     R4,-(6)         ;SAVE IT
003264  012704  001600                  MOV     #1600,R4        ;SET TO BANK 7
003270  010477  000416          4$:     MOV     R4,@#KIPAR6     ;SET THE BANK
003274  012700  140000                  MOV     #140000,R0      ;GET FIRST ADDRESS
003300  016720  000376          5$:     MOV     DATA,(0)+       ;LOAD THE DATA
003304  022700  160000                  CMP     #160000,R0      ;IS IT THE END?
003310  001373                          BNE     5$              ;LOOP UNTIL DONE
003312  062704  000200                  ADD     #200,R4         ;BUMP TO NEXT BANK
003316  020416                          CMP     R4,(6)          ;END YET
003320  001363                          BNE     4$              ;NO - LOOP
003322  005726                          TST     (6)+            ;CLEAR STACK
003324  005077  000354                  CLR     @#SR0           ;TURN KT11C OFF
003330  000207                          RTS     PC              ;RETURN
```

```
   417  003332  016704  000342          CHECK1  MOV     LIMIT,R4        ;GET BANK COUNT
   418  003336  022704  000006                  CMP     #6,R4           ;IS IT > 6?
   419  003342  100002                          BPL     1$              ;SKIP IF > 6
   420  003344  012704  000006                  MOV     #6,R4           ;FUDGE IN A 6
   421  003350  072427  000013          1$:     ASH     #13.,R4         ;MAKE IT AN ADDRESS
   422  003354  062704  017500                  ADD     #17500,R4       ;MAKE IT ABS LOADER ADDRESS
   423  003360  012700  004300                  MOV     #END,R0         ;LOAD LAST ADDRESS
   424  003364  026710  000312          2$:     CMP     DATA,(0)        ;CHECK THE DATA
   425  003370  001401                          BEQ     .+4             ;SKIP ! OK
   426  003372  000000                          HALT                    ;DATA IS WRONG
   427  003374  005720                          TST     (0)+            ;BUMP R0
   428  003376  020004                          CMP     R0,R4           ;IS IT THE END YET?
   429  003400  001371                          BNE     2$              ;LOOP UNTIL DONE
   430  003402  016704  000272                  MOV     LIMIT,R4        ;GET BANK COUNT AGAIN
   431  003406  022704  000006                  CMP     #6,R4           ;CHECK AGAIN
   432  003412  100401                          BMI     3$              ;YES - SKIP IF KT11C
   433  003414  000207                          RTS     PC              ;NO - EXIT
   434  003416  004767  000502          3$:     JSR     PC,MAP          ;MAP THE WORLD
   435  003422  005277  000256                  INC     @SR0            ;TURN ON KT11C
   436  003426  005204                          INC     R4              ;GET TO RIGHT ONE
   437  003430  072427  000007                  ASH     #7,R4           ;SHIFT IT INTO POSITION
   438  003434  010446                          MOV     R4,-(6)         ;SAVE IT
   439  003436  012704  001600                  MOV     #1600,R4        ;SET TO BANK 7
   440  003442  010477  000244          4$:     MOV     R4,@KIPAR6      ;SET THE BANK
   441  003446  012700  140000                  MOV     #140000,R0      ;GET FIRST ADDRESS
   442  003452  026710  000224          5$:     CMP     DATA,(0)        ;CHECK THE DATA
   443  003456  001401                          BEQ     .+4             ;SKIP ! OK
   444  003460  000000                          HALT                    ;DATA IS WRONG
   445  003462  005720                          TST     (0)+            ;BUMP R0
   446  003464  022700  160000                  CMP     #160000,R0      ;IS IT THE END?
   447  003470  001370                          BNE     5$              ;LOOP UNTIL DONE
   448  003472  062704  000200                  ADD     #200,R4         ;BUMP TO NEXT BANK
   449  003476  020416                          CMP     R4,(6)          ;END YET
   450  003500  001360                          BNE     4$              ;NO - LOOP
   451  003502  005726                          TST     (6)+            ;CLEAR STACK
   452  003504  005077  000174                  CLR     @SR0            ;TURN KT11C OFF
   453  003510  000207                          RTS     PC              ;RETURN
```

```
   454  003512  012767  177777  000156  POWDWN: MOV     #-1,FLAG        ;FIRST INSTRUCTION FLAG
   455  003520  005067  000152          CLR     FLAG            ;NOW CLEAR IT
   456  003524  012777  003652  000132  MOV     #ILLUP,@UVEC    ;IF TOO FAST
   457  003532  011667  000124          MOV     (SP),ERROR      ;SET THE ERROR ADDRESS
   458  003536  022706  000440          CMP     #440,SP         ;YELLOW OR RED?
   459  003542  100402                  BMI     .+6             ;NO
   460  003544  012706  000500          MOV     #500,SP         ;SET EMERGENCY STACK
   461  003550  010046                  MOV     R0,-(0)         ;PUT
   462  003552  010146                  MOV     R1,-(0)         ;THE
   463  003554  010246                  MOV     R2,-(0)         ;REGISTERS
   464  003556  010346                  MOV     R3,-(0)         ;ON
   465  003560  010446                  MOV     R4,-(0)         ;THE
   466  003562  010546                  MOV     R5,-(0)         ;STACK
   467  003564  010667  000104          MOV     SP,SAV6         ;SAVE THE STACK POINTER
   468  003570  016700  175204          MOV     FACTOR,R0       ;SET TIME FACTOR
   469  003574  077001                  SOB     R0,.            ;NOW WAIT
   470  003576  012777  003606  000060  MOV     #POWUP,@UVEC    ;RESET THE UP VECTOR
   471  003604  000000                  HALT                    ;WAIT FOR POWER DOWN
   472
   473  003606  012777  003656  000054  POWUP: MOV     #ILLDWN,@DVEC   ;SET TOO FAST DOWN VECTOR
   474  003614  016706  000054          MOV     SAV6,SP         ;RESET SP
   475  003620  016700  175154          MOV     FACTOR,R0       ;SET TIME FACTOR
   476  003624  077001                  SOB     R0,.            ;WAIT
   477  003626  012605                  MOV     (0)+,R5         ;TAKE
   478  003630  012604                  MOV     (0)+,R4         ;THE
   479  003632  012603                  MOV     (0)+,R3         ;REGISTERS
   480  003634  012602                  MOV     (0)+,R2         ;FROM
   481  003636  012601                  MOV     (0)+,R1         ;THE
   482  003640  012600                  MOV     (0)+,R0         ;STACK
   483  003642  012777  003512  000020  MOV     #POWDWN,@DVEC   ;RESET THE DOWN VECTOR
   484  003650  000113                  JMP     (R3)            ;JUMP INDIRECT TO R3
   485
   486  003652  000000          ILLUP:  HALT                    ;POWER UP BEFORE POWER DOWN COMPLETE
   487  003654  000776                  BR      .-2             ;LOCK UP THE HALT
   488
   489  003656  000000          ILLDWN: HALT                    ;POWERED DOWN BEFORE UP COMPLETE
   490  003660  000776                  BR      .-2             ;LOCK UP THE HALT
   491
   492  003662  000000          ERROR:  0                       ;RETURN ADDRESS FROM POWER FAIL
   493
   494  003664  000024  000026  UVEC:   24,26                   ;UP ADDRESS PAIR
   495  003670  000024  000026  DVEC:   24,26                   ;DOWN ADDRESS PAIR
   496  003674  000000          SAV6:   0                       ;SOME PLACE TO PUT THE SP
   497  003676  000000          FLAG:   0                       ;1 INSTRUCTION DOWN FLAG
   498  003700  000000          LIMIT:  0                       ;TOP OF MEMORY
   499  003702  052525          DATA:   52525                   ;WHAT IS TO BE WRITTEN INTO MEMORY
```

```
                                SR0:    177572                  ;KT11C - MEMORY MANAGEMENT
                                KIPAR0: 172340
                                KIPDR0: 172300
                                KIPAR6: 172354
                                KIPDR6: 172314
                                KIPAR7: 172356
                                KIPDR7: 172316
                                UIPAR0: 177640
                                UIPDR0: 177600
                                UIPAR7: 177656
                                UIPDR7: 177616
                                ;       $OCTAL          OCTAL TYPEOUT ROUTINE
                                ;THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY, IT WILL TYPE
                                ;ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 18 BIT ADDRESS, OR TYPE
                                ;THE 16 BITS, IT IS CALLED VIA THE DUMP, SDUMP, DUMP18, OR BITYPE MACRO'S,
003732  012767  170101  000140  BITYPS: MOV     #170101,,PR     ;SET BIT FLAG ANS 16, CHARACTER COUNT
003740  000411                  BR      ,PTIT           ;NOW TYPE IT IN BIT FORM
003742  112767  000001  000130  PRINTR: MOVB    #1,,PR          ;SET ZERO FILL SWITCH
003750  000402                  BR      ,.+6            ;SKIP
003752  005067  000122          PRINTS: CLR     ,PR             ;SUPPRESS LEADING ZERO'S
003756  112767  177772  000115          MOVB    #-6,,PR+1       ;SET COUNT
003764  010446                  ,PTIT:  MOV     R4,-(6)         ;SAVE R4
003766  012704  004102                  MOV     #,PR+2,R4       ;SET POINTER TO FIRST ASCII CHAR,
003772  105014                          CLRB    (4)             ;CLEAR FIRST BYTE
003774  000411                          BR      ,PRF            ;ROTATE FIRST BIT
003776  105014                  ,PRL:   CLRB    (4)             ;CLEAR BYTE OF CHARACTER
004000  032767  000100  000072          BIT     #100,,PR        ;BIT TYPING MODE?
004006  001004                          BNE     ,PRF            ;YES - SKIP 2 ROTATES
004010  006105                          ROL     TTY             ;ROTATE BIT INTO C
004012  106114                          ROLB    (4)             ;PACK IT
004014  006105                          ROL     TTY             ;ROTATE BIT INTO C
004016  106114                          ROLB    (4)             ;PACK IT
004020  006105                  ,PRF:   ROL     TTY             ;ROTATE BIT INTO C
004022  106114                          ROLB    (4)             ;PACK IT
004024  105714                          TSTB    (4)             ;IS IT ZERO?
004026  001402                          BEQ     ,.+6            ;SKIP INC
004030  105267  000044                  INCB    ,PR             ;SET FILL SWITCH
004034  105767  000040                  TSTB    ,PR             ;CHECK FILL SWITCH
004040  001402                          BEQ     ,.+6            ;SKIP BITSET
004042  152724  000060                  BISB    #'0,(4)+        ;MAKE INTO ASCII CHAR
004046  105267  000027                  INCB    ,PR+1           ;INC COUNT
004052  001351                          BNE     ,PRL            ;REPEAT
004054  022704  004102                  CMP     #,PR+2,R4       ;EMPTY BUFFER?
004060  001002                          BNE     ,.+6            ;SKIP IF NOT
004062  112724  000060                  MOVB    #'0,(4)+        ;LOAD 1 ZERO
004066  105014                          CLRB    (4)             ;NULL TERMINATOR
004070  000004  004102                  TYPE    ,,PR+2          ;TYPE IT
004074  012604                          MOV     (6)+,R4         ;RESTORE R4
004076  000207                          RTS     PC              ;RETURN
004100  000012                  ,PR:    ,BLKW   12              ;COUNT, SWITCH, AND OUTPUT BUFFER
```

```
952  004124  012777  000000  177554  MAP:    MOV     #0,@KIPAR0
953  004132  012777  077406  177550          MOV     #77406,@KIPDR0
954  004140  012777  000200  177544          MOV     #200,@KIPAR6
955  004146  012777  077406  177540          MOV     #77406,@KIPDR6
956  004154  012777  007600  177534          MOV     #7600,@KIPAR7
957  004162  012777  077406  177530          MOV     #77406,@KIPDR7
958  004170  000207                          RTS     PC
959
960
961                              ;       STYPE           MESSAGE TYPEOUT ROUTINE
962
963                              ;THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY, THE
964                              ;CALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
965                              ;MESSAGE STARTING IN LOCATION "ADR!", 2) "TYPE ,CHAR" - TYPES
966                              ;THE ASCII "CHAR", AND 3) "PRINT <<15><12>"MESSAGE"> - TYPES
967                              ;THE MESSAGE WHICH IS INLINE ASCII.
968
969  004172  210546                  10$:    MOV     TTY,-(6)        ;SAVE TTY
970  004174  017605  000002                  MOV     @2(6),TTY       ;GET ADDRESS TO BE TYPED
971  004200  032705  177400                  BIT     #177400,TTY     ;IS IT A TYPEM?
972  004204  001004                          BNE     1$              ;NO
973  004206  010567  000064                  MOV     TTY,,TYPE       ;GET THE CHARACTER
974  004212  012705  004276                  MOV     #,TYPE,TTY      ;FUDGE THE ADDRESS
975  004216  105715                  1$:     TSTB    (TTY)           ;TERMINATOR?
976  004220  001406                          BEQ     2$              ;GET OUT IF SO
977  004222  112537  177566                  MOVB    (TTY)+,@#177566 ;LOAD AND TYPE THE CHARACTER
978  004226  105737  177564                  TSTB    @#177564        ;IS THE PRINTER READY
979  004232  100375                          BPL     .-4             ;WAIT UNTIL IT IS
980  004234  000770                          BR      1$              ;GET THE NEXT CHARACTER
981  004236  017646  000002          2$:     MOV     @2(6),-(6)      ;GET ADDRESS TO BE TYPED
982  004242  062766  000002  000004          ADD     #2,4(6)         ;ADD 2 TO THE ADDRESS
983  004250  022666  000002                  CMP     (6)+,2(6)       ;IS IT .+2?
984  004254  001006                          BNE     3$              ;NO
985  004256  062705  000002                  ADD     #2,TTY          ;ADD 2 TO THE ADDRESS
986  004262  042705  000001                  BIC     #1,TTY          ;BACK UP TO AN EVEN BYTE
987  004266  010566  000002                  MOV     TTY,2(6)        ;RESTORE ADDRESS
988  004272  012605                  3$:     MOV     (6)+,TTY        ;RESTORE TTY
989  004274  000002                          RTI                     ;RETURN
990  004276  000000                  .TYPE:  0                       ;CHARACTER TYPE LOCATION
991  004300  000000                  END:    0
992
993          000001                  .END
```

DCKBPB.P11 SYMBOL TABLE

```
BEGIN    001002          BELL   = 000007          BITYPS   003732          CHECK    003332
DATA     003702          DISPLA = 177570          DOCORE   001120          DONE     003110
DOSEG    001042          DVEC     003670          END      004300          ERROR    003662
FACTOR   001000          FLAG     003676          HLT    = 104000          ILLDWN   003656
ILLUP    003652          IOTS     004172          KIPAR0   003706          KIPAR6   003712
KIPAR7   003716          KIPDR0   003710          KIPDR.   003714          KIPDR7   003720
LIMIT    003700          LOAD     003166          MAP      004124          MES      001324
N      = 000017          PC     =%000007          PCNT     003160          POWDWN   003912
POWUP    003606          PRINTR   003742          PRINTS   003752          PS     = 177776
QQ     = 000001          R0     =%000000          R1     =%000001          R2     =%000002
R3     =%000003          R4     =%000004          R5     =%000005          SAV6     003674
SCOPE  = 000240          SP     =%000006          SR0      003704          SWR    = 177570
SW10   = 002000          SW11   = 004000          SW12   = 010000          SW13   = 020000
SW14   = 040000          SW15   = 100000          SW8    = 000400          SW9    = 001000
TST1     001444          TST10    002314          TST11    002362          TST12    002450
TST13    002536          TST14    002624          TST15    002712          TST16    003032
TST2     001542          TST3     001630          TST4     001716          TST5     001772
TST6     002116          TST7     002176          TTY    =%000005          TYPE   = 000004
TYPEIT   001152          UIPAR0   003722          UIPAR7   003726          UIPDR0   003724
UIPDR7   003730          UVEC     003664          .BIT   = 042000          .PR      004100
.PRF     004020          .PRL     003776          .PTIT    003764          .TBIT    003164
.TYPE    004276          .      = 004302
```

ERRORS DETECTED: 0