

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DCKBP-8-D
PRODUCT NAME:	11/45 POWER FAIL TEST
DATE CREATED:	1-NOV-72
MAINTAINER:	DIAGNOSTIC GROUP
AUTHORS:	BOB BRAIN

COPYRIGHT (C) 1972
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS.

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 11/45. 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. Initially power fails are tried in all processor 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 occurring and a memory volatility 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 POWDOWN and POWUP

These routines are used to save and restore vital registers and test the time allowed for power fall 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 fall 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 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 failing 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

H|t 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 initially set to 500

44		SETUP AND SIZING ROUTINES
93	TST1	SIMPLE DOWN/UP TEST (KERNAL)
120	TST2	SIMPLE DOWN/UP TEST (SUPERVISOR)
141	TST3	SIMPLE DOWN/UP TEST (USER)
165	TST4	POWER FAIL WITH ODD ADDRESS
180	TST5	POWER FAIL IN THE RED ZONE
208	TST6	POWER FAIL WITH TIME OUT (KERNAL)
224	TST7	POWER FAIL IN THE YELLOW ZONE (KERNAL)
253	TST10	POWER FAIL WITH RESETS
269	TST11	POWER FAIL WITH ODD ADDRESS (SUPERVISOR)
288	TST12	POWER FAIL WITH TIME OUT (SUPERVISOR)
305	TST13	POWER FAIL WITH ODD ADDRESS (USER)
324	TST14	POWER FAIL WITH TIME OUT (USER)
341	TST15	KT11C ABORT TEST
366	TST16	MEMORY VOLATILITY TEST
383		BELL AND SCOPE ROUTINE
471		POWER FAIL ROUTINE
529		OCTAL DUMP OF A WORD
580		TYPE ROUTINE

```

1          ,TITLE MAINDEC-11-DCKBP-B      11/45 POWER FAIL
2          ,COPYRIGHT 1972, DIGITAL EQUIPMENT CORP., MAYNARD, MASS
3          ,PROGRAM BY BOB BRAIN
4
5
6
7
8          ; SWITCH
9          ;
10         SW15= 100000
11         SW14= 40000
12         SW13= 20000
13         SW12= 10000
14         SW11= 4000
15         SW10= 2000
16         SW9= 1000
17         SW8= 400
18
19
20         N= 1
21         SCOPE= TRAP
22         HLT= EMT
23         TYPE= 10T
24         PS= 177776
25         SWR= 177570
26         DISPLAY=SWR
27         BELL= 7
28         R0= X0
29         R1= X1
30         R2= X2
31         R3= X3
32         R4= X4
33         R5= X5
34         TTY= X5
35         SP= X6
36         PC= X7
37         SCOPE= NOP
38
39         ,* 0
40         ;TRAP CATCHER FROM 0 = 776
41         ,* 200
42
43 000200 000137 001002 JMP ##BEGIN ;JUMP TO BEGINNING ADDRESS OF PROGRAM

```

```

44      001000      ,#      1000
45
46      001000      002000      FACTOR1 2000      ;CORE=2000 ;BIPOLAR=3300 ;MOS=1700
47
48      001002      012706      000500      BEGINI  MOV      #500,SP      ;** STACK AT 500 **
49      001006      012737      004172      000020      MOV      #IOTS,#20      ;SET IOT VECTOR
50      001014      012737      000006      000016      MOV      #6,#16      ;SET TRACE TRAP RETURN
51      001022      012777      003512      002640      MOV      #POWDN,#DVEC      ;SET UP POWER DOWN VECTOR
52      001030      012737      001120      000004      MOV      #DOCORE,#4      ;SET FOR TIMEOUT
53      001036      005777      002642      TST      @SR0      ;CHECK FOR KTI1
54      001042      004767      003056      DOSEGI  JSR      PC,HAP      ;SETUP MEMORY MANAGEMENT REGISTERS
55      001046      005277      002632      INC      @SR0      ;TURN ON MEMORY MANAGEMENT
56      001052      012737      001102      000004      MOV      #25,#4      ;SET TIMEOUT ADDRESS FOR CORE CALCULATIONS
57      001060      005737      157776      1$;      TST      @#157776      ;TRAP ON NON EX MEM
58      001064      062777      000200      002620      ADD      #200,@KIPAR6      ;GO TO NEXT BANK
59      001072      022777      007600      002612      CMP      #7600,@KIPAR6      ;LAST ONE?
60      001100      003367      BGT      1$      ;TRY NEXT
61      001102      017701      002604      2$;      MOV      @KIPAR6,R1      ;SAVE ASR6 IN R1
62      001106      072127      177771      ASH      #7,R1      ;PUT INTO POSITION
63      001112      042701      177740      BIC      #177740,R1      ;CLEAR JUNK
64      001116      000415      BR       TYPEIT      ;TYPE THE NAME
65
66      001120      012737      001152      000004      DOCOREI MOV      #TYPEIT,#4      ;SET FOR NEM
67      001126      012702      017776      MOV      #17776,R2      ;SET UP ADDRESS
68      001132      005001      CLR      R1      ;SET UP BANK COUNT
69      001134      062702      020000      1$;      ADD      #20000,R2      ;MOVE TO NEXT BANK
70      001140      005201      INC      R1      ;INC THE BANK COUNT
71      001142      005712      TST      (2)      ;TIMEOUT?
72      001144      022702      177776      CMP      #177776,R2      ;END?
73      001150      001371      BNE      1$      ;LOOP IF NOT AT THE END
74
75      001152      005301      TYPEIT: DEC      R1      ;DROPP BACK
76      001154      010167      002520      MOV      R1,LIMIT      ;SAVE THE TOP OF CORE
77      001160      012737      000006      000004      MOV      #6,#4      ;SET FOR NEM
78      001166      012706      000500      MOV      #500,SP      ;CLEAR STACK **500**
79      001172      005227      177777      INC      #1      ;TYPE THE OPTION ONLY ONCE
80      001176      001122      BNE      TST1      ;FIRST TIME?
81      001200      000004      TYPE     ,,*2      ;,ASCIZ <15><12>"MAINDEC-11-DCKBP-B"
82      001232      000004      TYPE     ,,*2      ;,ASCIZ <15><12><12>"BANKS 0 "
83      001254      010105      MOV      R1,TTY      ;TYPE R1 IN OCTAL
84      001256      004767      002470      JSR      PC,PRINTS      ;AND SUPPRESS LEADING ZERO'S
85      001262      000004      TYPE     ,,*2      ;,ASCIZ " EXIST"
86      001276      022701      000006      CMP      #6,R1      ;WHICH OPTION?
87      001302      100010      BPL      MES      ;SKIP IF NO KTI1C
88      001304      000004      001310      TYPE     ,,*2      ;,ASCIZ " WITH KTI1C"
89      001324
90      001324      000004      001330      MES:     TYPE     ,,*2      ;,ASCIZ <15><12><12>"INTERRUPT THE POWER AFTER THE TES"
91      001376      000004      001402      TYPE     ,,*2      ;,ASCIZ " NUMBER APPEARS IN THE DISPLAY"<15><12>

```

```

92      ;*****
93      ;TEST 1      SIMPLE DOWN/UP TEST (KERNAL)
94      ;*****
95      001444      TST1:
96      001444      012737      000001      177570      MOV      #1,@DISPLAY      ;SET TEST NUMBER
97      001452      005037      177776      CLR      #PS      ;SET KERNAL MODE
98      001456      012703      001464      MOV      #25,R3      ;SET POWER UP RETURN
99      001462      000001      WAIT     ;WAIT FOR THE POWER FAIL
100     001464      010600      2$;      MOV      SP,R0      ;GET SP
101     001466      022702      000474      CMP      #474,R0      ;CHECK SP
102     001472      001401      BEQ     ,+4      ;SKIP IF OK
103     001474      000000      HALT    ;SP NOT 474
104     001476      012706      000500      MOV      #500,SP      ;RESET SP
105     001502      013700      000474      MOV      @#474,R0      ;GET RETURN ADDRESS
106     001506      022700      001464      CMP      #25,R0      ;CHECK ADDRESS
107     001512      001401      BEQ     ,+4      ;SKIP IF OK
108     001514      000000      HALT    ;ADDRESS ON STACK IS WRONG
109     001516      013700      000476      MOV      @#476,R0      ;GET OLD PS
110     001522      022700      000000      CMP      #0,R0      ;CHECK OLD PS
111     001526      001401      BEQ     ,+4      ;SKIP IF OK
112     001530      000000      HALT    ;OLD PS IS WRONG
113     001532      1$;
114     001532      032737      040000      177570      BIT      #SW14,@SWR      ;LOOP ON TEST?
115     001540      001341      BNE     TST1      ;LOOP TO TST1
116
117
118      ;*****
119      ;TEST 2      SIMPLE DOWN/UP TEST (SUPERVISOR)
120      ;*****
121     001542      TST2:
122     001542      012737      000002      177570      MOV      #2,@DISPLAY      ;SET TEST NUMBER
123     001550      012737      040000      177776      MOV      #40000,@PS      ;SET SUPERVISOR MODE
124     001556      012703      001564      MOV      #25,R3      ;SET POWER UP RETURN
125     001562      000001      WAIT     ;WAIT FOR THE POWER FAIL
126     001564      012706      000500      2$;      MOV      #500,SP      ;RESET SP
127     001570      013700      000474      MOV      @#474,R0      ;GET RETURN ADDRESS
128     001574      022700      001564      CMP      #25,R0      ;CHECK ADDRESS
129     001600      001401      BEQ     ,+4      ;SKIP IF OK
130     001602      000000      HALT    ;ADDRESS ON STACK IS WRONG
131     001604      013700      000476      MOV      @#476,R0      ;GET OLD PS
132     001610      022700      040000      CMP      #40000,R0      ;CHECK OLD PS
133     001614      001401      BEQ     ,+4      ;SKIP IF OK
134     001616      000000      HALT    ;OLD PS IS WRONG
135     001620      1$;
136     001620      032737      040000      177570      BIT      #SW14,@SWR      ;LOOP ON TEST?
137     001626      001345      BNE     TST2      ;LOOP TO TST2

```



```

217
218
219
220 002176
221 002176 012737 000007 177570 MOV #7,0#DISPLAY ISET TEST NUMBER
222 002204 005037 177776 CLR #0#PS ISET KERNAL MODE
223 002210 005067 001462 CLR FLAG ICLEAR THE FLAG
224 002214 012737 002240 000004 MOV #25,0#4 ISET SICK TPAP ADDRESS
225 002222 012706 000400 MOV #400,SP ISET STACK TO YELLOW ZONE
226 002226 012703 002234 MOV #15,R3 ISET RETURN ADDRESS FOR POWER FAIL
227 002232 000001 WAIT IWAIT FOR POWER FAIL
228 002234 000000 15i HALT IPOWER FAIL RETURNED TOO SOON
229 002236 000422 BR 45 ISKIP SP CHECK
230 002240 012737 000006 000004 25i MOV #6,0#4 IRESET 4
231 002246 005767 001424 TST FLAG IIS THE FIRST INSTRUCTION FLAG SET?
232 002252 001010 BNE 55 IYFS
233 002254 012777 002262 001402 MOV #75,0#UVEC ISET UVEC TO HALT
234 002262 000000 75i HALT INOT ENOUGH OR TOO MANY INSTR, EXEC,
235 002264 012777 003512 001376 MOV #POWDN,0#DVEC ISET DVEC
236 002272 000404 BR 45 ISET OUT
237 002274 012703 002304 55i MOV #45,R3 ISET RETURN
238 002300 000002 RTI IGO TO THE POWER FAIL ROUTINE
239 002302 000000 HALT ISHOULD NOT RETURN HERE
240 002304
241 002304 032737 040000 177570 45i BIT #SW14,0#SWR ILOOP ON TEST?
242 002312 001331 BNE TST7 ILOOP TO TST7
243
244
245
246
247
248 002314
249 002314 012737 000010 177570 MOV #10,0#DISPLAY ISET TEST NUMBER
250 002322 005037 177776 CLR #0#PS ISET KERNAL MODE
251 002326 012703 002346 MOV #15,R3 ISET RETURN ADDRESS
252 002332 012706 000500 MOV #500,SP IRESET STACK
253 002336 000005 35i RESET IRESETS
254 002340 000005 RESET ITO WAIT
255 002342 000005 RESET IIN
256 002344 000774 BR 35 ILOOP
257 002346 012706 000500 15i MOV #500,SP IRESET STACK
258 002352 032737 040000 177570 BIT #SW14,0#SWR ILOOP ON TEST?
259 002360 001355 BNE TST10 ILOOP TO TST10
    
```

```

260
261
262
263 002362
264 002362 012737 000011 177570 MOV #11,0#DISPLAY ISET TEST NUMBER
265 002370 012737 002402 000004 MOV #35,0#4 ISET TRAP VECTOR
266 002376 012703 002426 MOV #15,R3 ISET RETURN ADDRESS FOR POWER FAIL
267 002402 012706 000500 35i MOV #500,SP IRESET STACK
268 002406 012737 040000 177776 MOV #40000,0#PS ISET SUPERVISOR MODE
269 002414 005737 000003 TST #03 ICAUSE ODD ADDRESS TRAP
270 002420 005037 177776 CLR #0#PS ISET KERNAL MODE
271 002424 000000 HALT IODD ADDRESS TRAP FAILED
272 002426 012706 000500 15i MOV #500,SP IRESET STACK POINTER
273 002432 012737 000006 000004 MOV #6,0#4 IRESET 4
274 002440 032737 040000 177570 BIT #SW14,0#SWR ILOOP ON TEST?
275 002446 001345 BNE TST11 ILOOP TO TST11
276
277
278
279
280
281 002450
282 002450 012737 000012 177570 MOV #12,0#DISPLAY ISET TEST NUMBER
283 002456 012737 002470 000004 MOV #35,0#4 ISET TRAP VECTOR
284 002464 012703 002514 MOV #15,R3 ISET UP RETURN ADDRESS FOR POWER FAIL
285 002470 012706 000500 35i MOV #500,SP IRESET STACK
286 002474 012737 040000 177776 MOV #40000,0#PS ISET SUPERVISOR MODE
287 002502 010037 173000 MOV R0,#173000 ICAUSE A TIMEOUT
288 002506 005037 177776 CLR #0#PS ISET KERNAL MODE
289 002512 000000 HALT ITIMEOUT FAILED
290 002514 012706 000500 15i MOV #500,SP IRESET STACK
291 002520 012737 000006 000004 MOV #6,0#4 IRESET 4
292 002526 032737 040000 177570 BIT #SW14,0#SWR ILOOP ON TEST?
293 002534 001345 BNE TST12 ILOOP TO TST12
    
```

```

294
295
296
297 002536
298 002536 312737 000013 177570
299 002544 012737 002556 000004
300 002552 012703 002602
301 002556 012706 000500
302 002562 012737 140000 177776
303 002570 005737 000003
304 002574 005037 177776
305 002600 000000
306 002602 012706 000500 15I
307 002606 012737 000006 000004
308 002614 032737 040000 177570
309 002622 001345
310
311
312
313
314
315 002624
316 002624 012737 000014 177570
317 002632 012737 002644 000004
318 002640 012703 002670
319 002644 012706 000500
320 002650 012737 140000 177776
321 002656 010037 173000
322 002662 005037 177776
323 002666 000000
324 002670 012706 000500 15I
325 002674 012737 000006 000004
326 002702 032737 040000 177570
327 002710 001345

```

 ITEST 13 POWER FAIL WITH ODD ADDRESS (USER)

 TST13I
 MOV #13,0#DISPLAY ISET TEST NUMBER
 MOV #35,0#4 ISET TRAP VECTOR
 MOV #15,R3 ISET RETURN ADDRESS FOR POWER FAIL
 35I MOV #500,SP IRESET STACK
 MOV #140000,0#PS ISET USER MODE
 TST #3 ICAUSE ODD ADDRESS TRAP
 CLR #0#PS ISET KERNAL MODE
 HALT IODD ADDRESS TRAP FAILED
 15I MOV #500,SP IRESET SP
 MOV #6,0#4 IRESET 4
 BIT #SW14,0#SWR ILOOP ON TEST?
 BNE TST13 ILOOP TO TST13

 ITEST 14 POWER FAIL WITH TIME OUT (USER)

 TST14I
 MOV #14,0#DISPLAY ISET TEST NUMBER
 MOV #35,0#4 ISET TRAP VECTOR
 MOV #15,R3 ISET UP RETURN ADDRESS FOR POWER FAIL
 35I MOV #500,SP IRESET STACK
 MOV #140000,0#PS ISET USER MODE
 MOV #0,0#173000 ICAUSE A TIMEOUT
 CLR #0#PS ISET KERNAL MODE
 HALT ITIMEOUT FAILED
 15I MOV #500,SP IRESET STACK
 MOV #6,0#4 IRESET 4
 BIT #SW14,0#SWR ILOOP ON TEST?
 BNE TST14 ILOOP TO TST14

```

328
329
330
331 002712
332 002712 012737 000015 177570
333 002720 012737 003010 000004
334 002726 005037 177776
335 002732 005777 000746
336 002736 012737 003002 000004
337 002744 004767 021154
338 002750 005077 000740
339 002754 012737 002772 000250
340 002762 012703 003004
341 002766 005277 000712
342 002772 012706 000500 35I
343 002776 005237 140000
344 003002 000000 45I
345
346 003004 005077 020674 15I
347 003010 012706 000500 25I
348 003014 012737 000006 000004
349 003022 032737 040000 177570
350 003030 001330
351
352
353
354
355 003032
356 003032 005037 177776
357 003036 012702 000010
358 003042 004767 000120 45I
359 003046 012703 003066
360 003052 012737 000016 177570
361 003060 004767 000246 25I
362 003064 000775
363 003066 012706 000500 15I
364 003072 004767 000234
365 003076 077217
366 003100 032737 040000 177570
367 003106 001351

```

 ITEST 15 KT11C ABORT TEST

 TST15I
 MOV #15,0#DISPLAY ISET TEST NUMBER
 MOV #25,0#4 ISET FOR TIMEOUT
 CLR #0#PS ISET KERNAL MODE
 TST #SR0 IIS THERE KT11C?
 MOV #45,0#4 IRESET 4
 JSR PC,MAP IMAP THE WORLD
 CLR #KIPDR6 IMAP UA TO 6
 MOV #35,0#257 ISET KT11C VECTOR
 MOV #15,R3 ILOAD PF RETURN
 #SR0 ITURN KT11C ON
 35I MOV #500,SP IZAP STACK
 INC #0#140000 IACCESS VIOLATION
 45I HALT INO VIOLATION OR TRAP TO 4
 15I CLR #SR0 ITURN OFF KT11C
 25I MOV #500,SP IMAKE A NEW STACK
 MOV #6,0#4 IRESET 4
 BIT #SW14,0#SWR ILOOP ON TEST?
 BNE TST15 ILOOP TO TST15

 ITEST 16 MEMORY VOLATILITY TEST

 TST16I
 CLR #0#PS ISET KERNAL MODE
 MOV #10,R2 ILOAD COUNT OF TEST ITERATIONS
 45I JSR PC,LOAD ILOAD ALL MEMORY WITH 52525
 MOV #15,R3 IPOWER FAIL RETURN ADDRESS
 MOV #16,0#DISPLAY ISET TEST NUMBER
 25I JSR PC,CHECK ICHECK FOR THE 52525
 BR 25 ILOOP FOR EVER OR POWER FAIL
 15I MOV #500,SP IZAP THE STACK
 JSR PC,CHECK ICHECK ALL MEMORY
 SOB R2,45 IDO IT 10 TIMES
 BIT #SW14,0#SWR ILOOP ON TEST?
 BNE TST16 ILOOP TO TST16

```

368 003110          DONE:
369 003110 062767 000001 000044 ADD #1,PCNT+2 ;ADD 1 TO THE PASS COUNT
370 003116 005567 000036 ADC PCNT ;MAKE IT DOUBLE PREC,
371 003122 032737 002000 177570 BIT #SW10,#SWR ;RING THE BELL?
372 003130 001002 BNE 45 ;NO!
373 003132 000004 000007 TYPE ,BELL ;RING THE BELL
374 003136 013700 000042 45: MOV ##42,H0 ;GET MONITOR ADDRESS
375 003142 001404 BEQ 35 ;IF NONE
376 003144 004710 JSR 7,(0) ;GO TO MONITOR
377 003146 000240 NOP ;SAVE ROOM
378 003150 000240 NOP ;FOR
379 003152 000240 NOP ;ACT11
380 003154 000137 001002 35: JMP ##BEGIN ;RETURN
381
382 003160 000000 000000 PCNT: R,0 ;PASS COUNT
383 003164 000000 ,TBIT: 0 ;T BIT FLAG
384
385
386 003166 016704 000506 LOAD: MOV LIMIT,R4 ;GET BANK COUNT
387 003172 022704 000006 CMP #6,R4 ;IS IT > 6?
388 003176 100002 BPL 15 ;SKIP IF > 6
389 003200 012704 000006 MOV #6,R4 ;FUDGE IN A 6
390 003204 072427 000015 15: ASH #13,,R4 ;MAKE IT AN ADDRESS
391 003210 062704 017500 ADD #17500,R4 ;MAKE IT ABS LOADER ADDRESS
392 003214 012700 004300 MOV #END,H0 ;LOAD LAST ADDRESS
393 003220 016720 000456 25: MOV DATA,(0)+ ;LOAD THE DATA
394 003224 020004 CMP R0,R4 ;IS IT THE END YET?
395 003226 001374 BNE 25 ;LOOP UNTIL DONE
396 003230 016704 000444 MOV LIMIT,R4 ;GET BANK COUNT AGAIN
397 003234 022704 000006 CMP #6,R4 ;CHECK AGAIN
398 003240 100401 BMI 35 ;YES = SKIP IF KT11C
399 003242 000207 RTS PC ;NO = EXIT
400 003244 004767 000654 35: JSR PC,MAP ;MAP THE WORLD
401 003250 005277 000430 INC #SR0 ;TURN ON KT11C
402 003254 005204 INC R4 ;GET TO RIGHT ONE
403 003256 072427 000007 ASH #7,R4 ;SHIFT IT INTO POSITION
404 003262 010446 MOV R4,-(6) ;SAVE IT
405 003264 012704 001600 MOV #1600,R4 ;SET TO BANK 7
406 003270 010477 000416 45: MOV R4,#KIPAR6 ;SET THE BANK
407 003274 012700 140000 MOV #140000,R0 ;GET FIRST ADDRESS
408 003300 016720 000376 55: MOV DATA,(0)+ ;LOAD THE DATA
409 003304 022700 160000 CMP #160000,R0 ;IS IT THE END?
410 003310 001373 BNE 55 ;LOOP UNTIL DONE
411 003312 062704 000200 ADD #200,R4 ;BUMP TO NEXT BANK
412 003316 020416 CMP R4,(6) ;END YET
413 003320 001363 BNE 45 ;NO = LOOP
414 003322 005726 TST (6)+ ;CLEAR STACK
415 003324 005077 000354 CLR #SR0 ;TURN KT11C OFF
416 003330 000207 RTS PC ;RETURN

```

```

417 003332 016704 000342 CHECK: MOV LIMIT,R4 ;GET BANK COUNT
418 003336 022704 000006 CMP #6,R4 ;IS IT > 6?
419 003342 100002 BPL 15 ;SKIP IF > 6
420 003344 012704 000006 MOV #6,R4 ;FUDGE IN A 6
421 003350 072427 000015 15: 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 25: 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 25 ;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 35 ;YES = SKIP IF KT11C
433 003414 000207 RTS PC ;NO = EXIT
434 003416 004767 000502 35: 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 45: MOV R4,#KIPAR6 ;SET THE BANK
441 003446 012700 140000 MOV #140000,R0 ;GET FIRST ADDRESS
442 003452 026710 000224 55: 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 55 ;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 45 ;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,OVVEC ;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,=(6) ;PUT
462 003552 010146 MOV R1,=(6) ;THE
463 003554 010246 MOV R2,=(6) ;REGISTERS
464 003556 010346 MOV R3,=(6) ;ON
465 003560 010446 MOV R4,=(6) ;THE
466 003562 010546 MOV R5,=(6) ;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,OVVEC ;RESET THE UP VECTOR
471 003604 000000 HALT ;WAIT FOR POWER DOWN
472
473 003606 012777 003656 000054 POWUPI MOV #ILLDWN,OVVEC ;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 (6)+,R5 ;TAKE
478 003630 012604 MOV (6)+,R4 ;THE
479 003632 012603 MOV (6)+,R3 ;REGISTERS
480 003634 012602 MOV (6)+,R2 ;FROM
481 003636 012601 MOV (6)+,R1 ;THE
482 003640 012600 MOV (6)+,R0 ;STACK
483 003642 012777 003512 000020 MOV #POWDOWN,OVVEC ;RESET THE DOWN VECTOR
484 003650 000113 JMP (R3) ;JUMP INDIRECT TO R3
485
486 003652 000000 ILLUPI 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

```

```

500 003704 177572 SR0: 177572 ;KT11C - MEMORY MANAGEMENT
501 003706 172340 KIPAR0: 172340
502 003710 172300 KIPDR0: 172300
503 003712 172354 KIPAR6: 172354
504 003714 172314 KIPDR6: 172314
505 003716 172356 KIPAR7: 172356
506 003720 172316 KIPDR7: 172316
507 003722 177640 UIPAR0: 177640
508 003724 177600 UIPDR0: 177600
509 003726 177656 UIPAR7: 177656
510 003730 177616 UIPDR7: 177616
511 ;
512 ; OCTAL TYPEOUT ROUTINE
513 ;
514 ; THIS ROUTINE IS USED TO TYPE AN OCTAL NUMBER ON THE TTY, IT WILL TYPE
515 ; ALL 6 CHARACTERS, SUPPRESS LEADING ZEROES, TYPE AN 18 BIT ADDRESS, OR TYPE
516 ; THE 16 BITS, IT IS CALLED VIA THE DUMP, SDUMP, DUMP16, OR BITYPE MACRO'S.
517 003732 012767 170101 000140 BITYPS: MOV #170101,,PR ;SET BIT FLAG AND 16, CHARACTER COUNT
518 003740 000411 BR ,PTIT ;NOW TYPE IT IN BIT FORM
519 003742 112767 000001 000130 PRINTR: MOVB #1,,PR ;SET ZERO FILL SWITCH
520 003750 000402 BR ,+6 ;SKIP
521 003752 005067 000122 PRINTS: CLR ,PR ;SUPPRESS LEADING ZERO'S
522 003756 112767 177772 000115 MOVB #=6,,PR+1 ;SET COUNT
523 003764 010446 ,PTIT: MOV R4,=(6) ;SAVE R4
524 003766 012704 004102 MOV #,PR+2,R4 ;SET POINTER TO FIRST ASCII CHAR,
525 003772 105014 CLR (4) ;CLEAR FIRST BYTE
526 003774 000411 BR ,PRF ;ROTATE FIRST BIT
527 003776 105014 ,PRL: CLR (4) ;CLEAR BYTE OF CHARACTER
528 004000 032767 000100 000072 BIT #100,,PR ;BIT TYPING MODE?
529 004006 001004 BNE ,PRF ;YES - SKIP 2 ROTATES
530 004010 006105 ROL TTY ;ROTATE BIT INTO C
531 004012 106114 ROLB (4) ;PACK IT
532 004014 006105 ROL TTY ;ROTATE BIT INTO C
533 004016 106114 ROLB (4) ;PACK IT
534 004020 006105 ,PRF: ROL TTY ;ROTATE BIT INTO C
535 004022 106114 ROLB (4) ;PACK IT
536 004024 105714 TSTB (4) ;IS IT ZERO?
537 004026 001402 BEQ ,+6 ;SKIP INC
538 004030 105267 000044 INCB ,PR ;SET FILL SWITCH
539 004034 105767 000040 TSTB ,PR ;CHECK FILL SWITCH
540 004040 001402 BEQ ,+6 ;SKIP BITSET
541 004042 152724 000060 BISO #'0,(4)+ ;MAKE INTO ASCII CHAR
542 004046 105267 000027 INCB ,PR+1 ;INC COUNT
543 004052 001351 BNE ,PRL ;REPEAT
544 004054 022704 004102 CMP #,PR+2,R4 ;EMPTY BUFFER?
545 004060 001002 BNE ,+6 ;SKIP IF NOT
546 004062 112724 000060 MOVB #'0,(4)+ ;LOAD 1 ZERO
547 004066 105014 CLR (4) ;NULL TERMINATOR
548 004070 000004 004102 TYPE ,,PR+2 ;TYPE IT
549 004074 012604 MOV (6)+,R4 ;RESTORE R4
550 004076 000207 RTS PC ;RETURN
551 004100 000012 ,PRI ,BLKW 12 ;COUNT, SWITCH, AND OUTPUT BUFFER

```

```

552 004124 012777 000000 177554 MAP: MOV #0,KIPAR0
553 004132 012777 077406 177550 MOV #77406,KIPDR0
554 004140 012777 000200 177544 MOV #200,KIPAR6
555 004146 012777 077406 177540 MOV #77406,KIPDR6
556 004154 012777 007600 177534 MOV #7600,KIPAR7
557 004162 012777 077406 177530 MOV #77406,KIPDR7
558 004170 000207 RTS PC
559
560

```

MESSAGE TYPEOUT ROUTINE

```

561
562
563 THIS ROUTINE IS USE TO TYPE ASCII MESSAGES ON THE TTY, THE
564 ICALL CAN BE IN ONE OF 3 FORMS: 1) "TYPE ,ADR" - TYPES THE
565 MESSAGE STARTING IN LOCATION "ADR", 2) "TYPE ,CHAR" - TYPES
566 THE ASCII "CHAR", AND 3) "PRINT <<1><2>"MESSAGE" - TYPES
567 THE MESSAGE WHICH IS INLINE ASCII.
568
569 004172 010546 IOTS: MOV TTY,=(6) ;SAVE TTY
570 004174 017605 000002 MOV #2(6),TTY ;GET ADDRESS TO BE TYPED
571 004200 032705 177400 BIT #177400,TTY ;IS IT A TYPED?
572 004204 001004 BNE 15 ;NO
573 004206 010567 000064 MOV TTY,TYPE ;GET THE CHARACTER
574 004212 012705 004276 MOV #,TYPE,TTY ;FUDGE THE ADDRESS
575 004216 105715 1S: TSTB (TTY) ;TERMINATOR?
576 004220 001406 BEQ 25 ;GET OUT IF SO
577 004222 112537 177566 MOVB (TTY)+,#177566 ;LOAD AND TYPE THE CHARACTER
578 004226 105737 177564 TSTB #177564 ;IS THE PRINTER READY
579 004232 106375 BPL ,=4 ;WAIT UNTIL IT IS
580 004234 006770 BR 15 ;GET THE NEXT CHARACTER
581 004236 017646 000002 2S: MOV #2(6),=(6) ;GET ADDRESS TO BE TYPED
582 004242 062766 000002 000004 ADD #2,4(6) ;ADD 2 TO THE ADDRESS
583 004250 022666 000002 CMP (6)+,2(6) ;IS IT ,+2?
584 004254 001006 BNE 35 ;NO
585 004256 062705 000002 ADD #2,TTY ;ADD 2 TO THE ADDRESS
586 004262 042705 000001 BIC #1,TTY ;BACK UP TO AN EVEN BYTE
587 004266 010566 000002 MOV TTY,2(6) ;RESTORE ADDRESS
588 004272 012605 3S: MOV (6)+,TTY ;RESTORE TTY
589 004274 000002 RTI ;RETURN
590 004276 000000 ,TYPE: 0 ;CHARACTER TYPE LOCATION
591 004300 000000 END: 0
592
593 000001 ,END

```

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	ILLDOWN	003656
ILLUP	003652	IOTS	004172	KIPAR0	003706	KIPAR6	003712
KIPAR7	003716	KIPDR0	003710	KIPDR6	003714	KIPDR7	003720
LIMIT	003700	LOAD	003166	MAP	004124	MES	001324
N	= 000017	PC	=X000007	PCNT	003160	POWDOWN	003512
POWUP	003606	PRINTR	003742	PRINTS	003752	PS	= 177776
QQ	= 000001	R0	=X000000	R1	=X000001	P2	=X000002
R3	=X000003	R4	=X000004	R5	=X000005	SAV6	003674
SCOPE	= 000240	SP	=X000006	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	=X000005	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