

# BM873

RESTART ROM LOADER  
MD-11-DZBMD-F

EP-DZBMD-F-DL-B  
COPYRIGHT © 1975  
FICHE 1 OF 1

OCT 1975  
digital  
MADE IN USA

DZBMD-F  
SEQ

The microfiche grid contains 56 frames of text, arranged in 7 rows and 8 columns. Each frame displays a page of text from the document 'DZBMD-F SEQ'. The text is printed in a monospaced font and appears to be a sequence of data or instructions. The first frame in the top-left corner contains the title 'DZBMD-F' and 'SEQ'. The text in the frames is too small to read accurately but appears to be a sequence of data or instructions.

CTA



801

ASCII art patterns using characters D, Z, B, M, and F. The patterns are arranged in a grid-like structure with varying lengths and spacing.

ASCII art patterns using characters S, E, Q, and O. The patterns are arranged in a grid-like structure with varying lengths and spacing.

LP: SPL Version 100(2025) Running on MTA260
\*START\* User: WELINGHAM, A (400,3722) Job DZBMDF Seq. 696 Date 09-Oct-75 09:03:25 Monitor RK213C KL10 SYS#1026 \*START\*
Request created: 09-Oct-75 09:02:46
Bill E BGKN: DZBMDF / SIDE 408, 372200216511d / SBAC006-75 / DDM00: 982<058RMSPNORNA0: 09-Oct-75 09:03:26

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZBMD-F-D

PRODUCT NAME: BMB73 - UNIVERSAL RESTART ROM LOADER

DATE RELEASED: 21 NOVEMBER 1975

MAINTAINER: DIAGNOSTIC GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH A LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIBILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975 DIGITAL EQUIPMENT CORPORATION

PROGRAM HISTORY  
-----

PRODUCT CODE: MAINDEC-11-DZBMD-F-D  
-----

PRODUCT NAME: BMB73 - UNIVERSAL RESTART ROM LOADER  
-----

DATE CREATED: JULY 1973  
-----

MAINTAINER: DIAGNOSTIC GROUP  
-----

AUTHOR: JOHN EGOLF Y\*, YA  
-----

REVISED BY: BOB MISNER 10/21/74 YB  
FAY BASHAW 3/21/75 YC, YD  
JIM KELLY 7/21/75 SYSMAC  
JOHN EGOLF 11/21/75 YF

COPYRIGHT (C) MARCH, 1974  
DIGITAL EQUIPMENT CORPORATION  
MAYNARD MASSACHUSETTS 01754

## 1. ABSTRACT

THIS MAINDEC CONSISTS OF FOUR PROGRAMS. THE TWO MAIN PROGRAMS ARE PROGRAM ONE AND PROGRAM FOUR. THESE PROGRAMS WILL BE DISCUSSED LATER.

THE PURPOSE OF THIS DIAGNOSTIC IS TO VERIFY THE DATA IN THE ROM, MAKE SURE ALL ADDRESS WILL CAUSE A TIME OUT TRAP WHEN WRITTEN INTO (EXCEPT THE TRAP VECTORS: 173024, 173224) AND ALERT THE OPERATOR AS TO WHAT THE OFFSET ADDRESS WOULD BE IF A SELECTED BUTTON IS PUSHED.

NOTE: FOR NORMAL CONFIGURATIONS; THE ONLY PROGRAMS NECESSARY FOR ACCEPTANCE OF THE BMB73 ARE PROGRAMS ONE AND FOUR. PROGRAM TWO IS NECESSARY FOR "NON-STANDARD" SETUPS AND IS A MAINTAINCE TOOL. PROGRAM THREE IS ALSO JUST FOR MAINTAINCE AID.

## 2. REQUIRMENTS

## 2.1 EQUIPMENT

ANY PDP-11/40 CPU  
UNIVERSAL RESTART LOADER  
TELETYPE OR EQUIVALENT  
AT LEAST 4K OF MEMORY.

## 2.2 STORAGE

THIS PROGRAM RESERVES THE RIGHT TO USE ALL OF THE FIRST 4K EXCEPT WHERE BOOTSTRAP LOADER AND ABSOLUTE LOADER RESIDE.

## 3. LOADING PROCEDURE

THE PROGRAM MAY BE LOADED LIKE ANY OTHER PROGRAM SUCH AS: PAPER TAPE, DECTAPE MAGTAPE, DISK, ETC. MOST COMMON WILL BE THROUGH DECTAPE BY THE USE OF ROM BOOT LOADER.

## 4. STARTING PROCEDURE

## 4.1 CONTROL SWITCH SETTINGS

SWITCH 00 CLEARED INDICATES ONLY FIRST 128 WORDS TO BE CHECKED.  
 SET INDICATES EXTENDED 128. WORDS ARE TO BE CHECKED IN WHICH CASE PROGRAM 2 MUST BE RUN FIRST.  
 WHEN RUNNING ON BM873YB, BM873YC, BM873YD OR BM873YF, 256 WORDS ARE AUTOMATICALLY CHECKED.

## 4.2 STARTING ADDRESS

STARTING ADDRESS 000200

## 4.3 OPERATOR ACTION

## 4.3.1 FOR NORMAL OPERATION (WITHOUT EXTENDED 128 WORDS)

1. LOAD STARTING ADDRESS (000200)
2. SET SWITCHES AS PER 5.1.1 (NORMAL ALL SWITCHES DOWN)
3. PRESS START SWITCH AND RELEASE.

## 4. DEVICE VERSION.

WHEN PROGRAM IS STARTED FOR THE FIRST TIME THE FOLLOWING WILL BE PRINTED OUT:

MAINDEC-11-DZBMOF  
 DEVICE VERSION  
 BM873-Y

THE OPERATOR WILL THEN SPECIFY THE VERSION BEING RUN.

BM873-Y\* IS ANY NON-STANDARD VERSION.

NOTE: PROGRAM TWO MUST BE RUN FIRST.

BM873-YA REPLACES M792-YA, MR11-DB, M792-YH

BM873-YB MASSBUS

BM873-YC DDCMP BOOTSTRAP ROM

BM873-YD KL10 (PDP-11) 256 BOOTSTRAP ROM (VERSION 2(17))

BM873-YF KL10 (PDP-11) 256 BOOTSTRAP ROM (VERSION 3(23))

5. THEN TYPE IN NUMBER OF PROGRAM TO BE RUN (NORMALLY PROGRAM 1 AND 4)

## 4.3.2 IF YOU WISH TO TEST THE EXTENDED 128. WORDS THIS IS THE PROCEDURE:

(NOT NEEDED FOR NORMAL TESTING OF BM873YB, BM873YC, BM873YD, OR BM873YF)

1. LOAD STARTING ADD. 000200
2. SET SW00=1
3. SET HALT ENABLE SW AND SINGLE CYCLE SW UP
4. HIT START SWITCH AND RELEASE.
5. RUN PROGRAM 2 FOR ONE PASS.
6. NOW ANY PROGRAM MAY BE RUN.

NOTE: VISUAL INSPECTION OF EXTENDED DUMP IS YOUR RESPONSIBILITY. THAT DATA WAS PLACED INTO SOFTWARE TABLE FOR TEST COMPARISON.

## 5. OPERATING PROCEDURE

## 5.1.1 SWITCH SETTINGS (APPLICABLE IN ALL PROGRAMS)

SW15 = 1 OR UP ... HALT ON ERROR

SW14 = 1 OR UP ... LOOP ON TEST

SW13 = 1 OR UP ... INHIBIT ERROR PRINT OUT

SW12 = 1 OR UP ... INHIBIT ALL PRINT OUT/ BELL ON ERROR.

SW11 = 1 OR UP ... INSTEAD OF EXERCISING EACH ADDRESS 10X DO IT 1X.

SW09 = 1 OR UP ... LOOP WITH CURRENT ADDRESS

SW08 = 1 OR UP ... GOTO BEGINNING OF CURRENT PROGRAM ON ERROR

## 6. ERRORS

## 6.1 ERROR PRINT OUT

ALL ERRORS WILL HAVE A PRINT OUT. IF IT WAS A COMPARISON ERROR: THE SOFT ADDRESS, ROM ADDRESS, EXPECTED DATA (FROM SOFTWARE MAP), AND THE FOUND DATA WILL BE PRINTED OUT. IF IT WAS A "NO TRAP WHEN WRITTEN" ERROR: THE ADDRESS WILL BE PRINTED OUT. IF IT WAS AN "UNEXPECTED TRAP" WHEN READING ROM THE ADDRESS WILL BE PRINTED.

## 6.2 ERROR RECOVERY

1. ITS A GOOD IDEA TO LEAVE SW15=1 WHILE TEST RUNS TO PREVENT A RUN AWAY ERROR FROM GOING WILD IF YOU LEAVE THE CPU.
2. IN AN ERROR; SET SW14=1(LOOP ON THIS ADDR.) AND SET SW 13=1(DELETE ERROR PRINT OUT). IF CPU IS HALTED; HIT CONTINUE.
3. NOW THE PROGRAM IS RUNNING AND YOU MAY SCOPE IT.

## 7. RESTRICTIONS

## 7.1 STARTING RESTRICTIONS

SEE SECTION 4.

## 7.2 OPERATING RESTRICTIONS

- 7.2.1 IF YOU WISH PROGRAM TO TEST YOUR EXTENDED 128. WORDS; YOU MUST START AS PER SECTION 4 AND THEN  
\*\*\*\*\* RUN PROGRAM 2 FIRST AND VISUALLY VERIFY DATA.\*\*\*\*  
(NOT APPLICABLE TO BM873YB, BM873YC, BM873YD, OR BM873YF)
- 7.2.2 YOU MAY NOT ALTER THE SOFTWARE MAP UNLESS--  
\*\*\*\*\* YOU KNOW WHAT YOU ARE DOING \*\*\*\*\*
- 7.2.3 THE ROM ADDRESS MUST START AT 173000 AND BE AT LEAST 128 WORDS LONG. (256 FOR THE BM873YB, BM873YC, BM873YD, OR BM873YF)

## 8. MISCELLANEOUS

## 8.1 EXECUTION TIME

PROGRAM ONE WILL PASS AT APPROX. FIVE MINS.  
PROGRAM TWO HAS NO END PASS; BUT WILL HALT AT COMPLETEION  
HIT CONTINUE TO PROCEED IN THIS PROGRAM.  
PROGRAM THREE (RUN) WILL PASS APPROX. FIVE MINS.  
PROGRAM FOUR WILL PASS APPROX. FIVE MINS



## 9. PROGRAM DESCRIPTION

## 9.1 PROGRAM 1

PROGRAM 1 WILL VERIFY THE DATA IN THE ROM AND THE VERIFY THAT WRITING THE ROM WILL TRAP OUT (EXCEPT THE VECTORS) EACH ADDRESS IS REFERENCED FIVE TIMES IN A ROW BEFORE UPDATING TO THE NEXT ADDRESS.

IF SW00 WAS UP WHEN START WAS HIT. THE EXTENDED 128 WORDS WILL BE CHECKED.  
256 WORDS WILL BE CHECKED AUTOMATICALLY IF BMB73YB, BMB73YC, BMB73YD, OR BMB73YF IS TESTED.

## 9.2 PROGRAM 2

PROGRAM 2 WILL DUMP THE CONTENTS OF THE ROM ONTO THE TTY. NOTE NO VERIFICATION

OF ANY KIND IS PERFORMED ON THE DATA. (AN ERROR WILL OCCUR IF A TRAP IS ENCOUNTERED WHILE READING) YOU MUST INSPECT THE DATA YOUR SELF. IF SW00 WAS UP WHEN START WAS HIT THE EXTENDED 128. WORDS WILL BE PRINTED.

256 WORDS WILL BE PRINTED IF BMB73YB, BMB73YC, BMB73YD, OR BMB73YF IS SELECTED

## 9.3 PROGRAM 3

PROGRAM 3 IS THE SAME AS PROGRAM ONE EXCEPT THAT THE USER HAS THE ABILITY TO ALTER THE SOFTWARE MAP.

LIST OR PRINT THE SOFTWARE MAP, AND RUN THE PROGRAM.

NOTE THAT IF YOU ALTER THE MAP BE CAREFULL OF WHAT YOU CHANGE.

FOR THE COMMANDS TO BE USED SEE TOP OF PROGRAM 3 IN THIS LISTING

## 9.4 PROGRAM 4

PROGRAM 4 CHECKS THE OFFSET ADDRESS WHEN THE SIMULATED PUSHING OF A BUTTON IS DONE BY THE SOFTWARE. ON THE FIRST PASS

THE OFFSET IS TYPED OUT FOR YOU TO VERIFY (NOTE: THE PROGRAM HAS NO WAY OF KNOWING WHAT THE OFFSET WILL BE). AFTER THE DATA IS TYPED OUT IT IS STORED AWAY IN CORE. WHEN THE FIRST PASS IS FINISHED THE PROCESS IS REPEATED ONLY NO TYPE OUT IS PERFORMED, AND THE DATA IN CORE IS COMPARED TO THE DATA FOUND AT THE ROM.

DURING THIS TEST "WRITING" THE ROM IS PERFORMED. THE VECTORS (173024,173224) ARE "WRITTEN" AND ARE \*\*NOT\*\* EXPECTED TO TRAP. AN ERROR MESSAGE WILL BE REPORTED IF A TRAP IS DISCOVERED.

# J01

SEG 0008

9.5 THIS PROGRAM IS "XXDP AND ACT-11" COMPATIBLE;  
AT PRESENT TIME IF IN CHAIN MODE UNDER ACT-11 OR  
XXDP THE PROGRAM AUTOMATICALLY DETERMINES IF THE ROM IS  
BM873YA OR YB, YC, YD, OR YF BY COMPARING THE 1ST WORD IN ROM WITH THE  
EXPECTED WORD. THE DIAGNOSTIC THEN RUNS  
PROGRAM 1 AND PROGRAM 4 BEFORE ENTERING THE MONITOR.

9.6 ELECTRICAL PREREQUISITES (HARDWARE)

9.7.1 THIS OPTION MUST BE ON THE CPU SIDE OF ANY BUS BUFFERS.

9.7.2 NPR CYCLES ARE NOT PERMITTED DURING THE POWER UP TRAP  
SEQUENCE.

9.7.3 IF FURTHER INFORMATION IS NEEDED  
CONSULT THE BM873 MANUAL FOR HELP.  
NOTE: THE DIAGNOSTIC RUNNING WITHOUT ANY INTERFERENCE FROM  
THE USER HAS NO WAY OF CHECKING THE PRESENTS OF THE  
"ACLO" AND "DCLO" SIGNALS ON THE OPTION.

.NLIST  
.LIST SEQ,LOC,BIN  
.LIST  
.PAGE  
.ENDM HELLO



1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54

```

.MCALL .HEADER, .SWRHI, .SWRLO, .EQUATE, .SETUP, .STRAP, .SCATCH, .SCMTAG
.MCALL .RDLIN, .SSCOPE, .SEERROR, .SERRTYP, .SRDOCT

.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 007260      JMP      @#RESTRT      ;JUMP TO STARTING ADDRESS OF PROGRAM

.SBTTL BASIC DEFINITIONS

001100      ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
            STACK= 1100
            .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

            ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000      R0= X0      ;GENERAL REGISTER
000001      R1= X1      ;GENERAL REGISTER
000002      R2= X2      ;GENERAL REGISTER
000003      R3= X3      ;GENERAL REGISTER
000004      R4= X4      ;GENERAL REGISTER
000005      R5= X5      ;GENERAL REGISTER
000006      R6= X6      ;GENERAL REGISTER
000007      R7= X7      ;GENERAL REGISTER
            .EQUIV R6,SP      ;STACK POINTER
            .EQUIV R7,PC      ;PROGRAM COUNTER

            ;*"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
000010      SW03= 10

```



55 000004  
56 000002  
57 000001  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70 100000  
71 040000  
72 020000  
73 010000  
74 004000  
75 002000  
76 001000  
77 000400  
78 000200  
79 000100  
80 000040  
81 000020  
82 000010  
83 000004  
84 000002  
85 000001  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98 000004  
99 000010  
100 000014  
101 000014  
102 000014  
103 000020  
104 000024  
105 000030  
106 000034  
107 000060  
108 000064

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)

BIT15= 100000  
BIT14= 40000  
BIT13= 20000  
BIT12= 10000  
BIT11= 4000  
BIT10= 2000  
BIT09= 1000  
BIT08= 400  
BIT07= 200  
BIT06= 100  
BIT05= 40  
BIT04= 20  
BIT03= 10  
BIT02= 4  
BIT01= 2  
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

ERRVEC= 4 ; TIME OUT AND OTHER ERRORS  
RESVEC= 10 ; RESERVED AND ILLEGAL INSTRUCTIONS  
TBITVEC= 14 ; "T" BIT  
TRTVEC= 14 ; TRACE TRAP  
BPTVEC= 14 ; BREAKPOINT TRAP (BPT)  
IOTVEC= 20 ; INPUT/OUTPUT TRAP (IOT) \*\*SCOPE\*\*  
PMRVEC= 24 ; POWER FAIL  
EMTVEC= 30 ; EMULATOR TRAP (EMT) \*\*ERROR\*\*  
TRAPVEC= 34 ; "TRAP" TRAP  
TKVEC= 60 ; TTY KEYBOARD VECTOR  
TPVEC= 64 ; TTY PRINTER VECTOR

MO1

MAR 1975  
DZBMOF.P11

MACY11 27(657) 1-OCT-75 09:30 PAGE 3  
BASIC DEFINITIONS

SEQ 0011

109

000240

PIRQVEC=240

;PROGRAM INTERRUPT REQUEST VECTOR

```

110 ;*****
111
112 .SBTTL COMMON TAGS
113
114 ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
115 ;*USED IN THE PROGRAM.
116
117 000046 000046 .=46
118 000046 014774 $ENDAD ;LOGICAL END OF PROGRAM
119
120 001100 .=1100
121
122 001100 $CMTAG: ;START OF COMMON TAGS
123 001100 000000 $PASS: .WORD 0 ;CONTAINS PASS COUNT
124 001102 000 $STNM: .BYTE 0 ;CONTAINS THE TEST NUMBER
125 001103 000 $ERFLG: .BYTE 0 ;CONTAINS ERROR FLAG
126 001104 000000 $ICNT: .WORD 0 ;CONTAINS SUBTEST ITERATION COUNT
127 001106 000000 $LPADR: .WORD 0 ;CONTAINS SCOPE LOOP
128 001110 000000 $LPERR: .WORD 0 ;CONTAINS SCOPE RETURN FOR ERRORS
129 001112 000000 $ERTTL: .WORD 0 ;CONTAINS TOTAL ERRORS DETECTED
130 001114 000 $ITEMB: .BYTE 0 ;CONTAINS ITEM CONTROL BYTE
131 001115 001 $ERMAX: .BYTE 1 ;CONTAINS MAX. ERRORS PER TEST
132 001116 000000 $ERRPC: .WORD 0 ;CONTAINS PC OF LAST ERROR INSTRUCTION
133 001120 000000 $GDADR: .WORD 0 ;CONTAINS OF 'GOOD' DATA
134 001122 000000 $BDADR: .WORD 0 ;CONTAINS OF 'BAD' DATA
135 001124 000000 $GODAT: .WORD 0 ;CONTAINS 'GOOD' DATA
136 001126 000000 $BODAT: .WORD 0 ;CONTAINS 'BAD' DATA
137 001130 000000 000000 000000 .WORD 0,0,0 ;RESERVED--NOT TO BE USED
138 001136 177560 $TKS: 177560 ;TTY KBD STATUS
139 001140 177562 $TKB: 177562 ;TTY KBD BUFFER
140 001142 177564 $TPS: 177564 ;TTY PRINTER STATUS REG.
141 001144 177566 $TPB: 177566 ;TTY PRINTER BUFFER REG.
142 001146 000 $NULL: .BYTE 0 ;CONTAINS NULL CHARACTER FOR FILLS
143 001147 002 $FILLS: .BYTE 2 ;CONTAINS # OF FILLER CHARACTERS REQUIRED
144 001150 012 $FILLC: .BYTE 12 ;INSERT FILL CHARS. AFTER A "LINE FEED"
145 001151 000 $TPFLG: .BYTE 0 ;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
146 001152 077 $QUES: .ASCII ?? ;QUESTION MARK
147 001153 015 $CRLF: .ASCII <15> ;CARRIAGE RETURN
148 001154 000012 $LF: .ASCII <12> ;LINE FEED

```



```

149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165 001156
166
167
168
169 001156 016206
170 001160 016414
171 001162 016612
172 001164 000000
173
174
175
176 001166 016250
177 001170 016513
178 001172 016626
179 001174 000000
180
181
182
183 001176 016307
184 001200 016545
185 001202 016634
186 001204 000000
187
188
189 001206 016354
190 001210 016513
191 001212 016626
192 001214 000000
193
194 001216 000000
195 001220 000000
196 001222 000000
197 001224 000000
198 001226 000000
199 001230 000000
200 001232 000000
201 001234 000000
202 001236 000000

```

```

;*****
.SBTTL  ERROR POINTER TABLE

*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
*NOTE1:      IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
*NOTE2:      EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

*      EM      ;POINTS TO THE ERROR MESSAGE
*      DH      ;POINTS TO THE DATA HEADER
*      DT      ;POINTS TO THE DATA
*      DF      ;POINTS TO THE DATA FORMAT

SERRTB:

;ERROR TABLE ITEM FOR ERROR MESSAGE 0

      EM1      ;"ROM READ DATA COMPARISON ERROR"
      DH1
      DT1
      0        ;* PRINT ALL NUMERIC DATA IN OCTAL

;ERROR TABLE ITEM FOR ERROR MESSAGE 1

      EM2      ;"WRITTING ROM FAILED TO TRAP"
      DH2
      DT2
      0        ;PRINT ALL NUMERIC DATA IN OCTAL.

;ERROR TABLE ITEM FOR ERROR MESSAGE 2

      EM3      ;"UNEXPECTED TRAP WHILE READING ROM"
      DH3
      DT3
      0

;ERROR TABLE ITEM FOR ERROR MESSAGE 3

      EM4      ;"FATAL TRAP. ROM PC ON STACK."
      DH2
      DT2
      0

LSTERR: 0      ;ERROR FLAG
ICOUNT: 0      ;ITERATION COUNT.
TEMPS:  0
TEMP3:  0
TEMP4:  0
SAVRO:  0
SAVR1:  0
SAVR4:  0
SAVR5:  0

```

MAR 1975  
DZBMOF.P11

MACY11 27(657) 1-OCT-75 09:30 PAGE 6  
ERROR POINTER TABLE

SEQ 0014

203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214

000001  
160000

```
.TITLE MAR 1975
.*COPYRIGHT (C) BM873 YX
.*DIGITAL EQUIPMENT CORP.
.*MAYNARD, MASS. 01754
.*
.*PROGRAM BY DZBMD
.*
.*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
.*PACKAGE (MAINDEC-11-DZGAC-A1).
.*
$TN=1
$SWR=160000 :HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT
```

MAR 1975  
DZBMOF.P11MACY11 27(657) 1-OCT-75 09:30 PAGE 7  
ERROR POINTER TABLE

SEG 0015

```

215
216
217 001400  =1400
218      MAP.YA:
219      ;THE FOLLOWING IS A REPRODUCTION
220      ;OF THE ROM PROGRAM FOR BM873YA.
221      ;IT IS HERE FOR COMPARISON TO
222      ;ACTUAL ROM AND FOR REFERENCE.
223      ;173000 .=173000      ;STARTING ADDRESS FOR BOOTSTRAP
224      ;THIS LOADER IS DESIGNED FOR THE RESTART MODULE M873.
225      ;IT FUNCTIONALLY REPLACES THE FOLLOWING ROMS:
226      ;M792-YA - PAPER TAPE BOOTSTRAP FOR PC11,KL11
227      ;MR11-DB  BULK STORAGE BOOTSTRAP ROM
228      ;M792-YH  TAI1 CASSETTE BOOTSTRAP ROM
229      ;
230      ;000000  R0=    X0      ;REGISTER DEFINITIONS
231      ;000001  R1=    X1
232      ;000002  R2=    X2
233      ;000003  R3=    X3
234      ;000004  R4=    X4
235      ;000005  R5=    X5
236      ;000006  SP=    X6
237      ;000007  PC=    X7
238      ;177570  SR=    177570      ;PROCESSOR SWITCH REGISTER
239
240      ;STARTING LOCATION FOR RF11 DISK
241      RF11:  MOV PC,R2      ;SET POINTER TO PARAMETER LISTS
242      BR OTHER      ;TRANSFER TO SERVICE ROUTINE
243      .WORD 177462      ;DEVICE WORD COUNT ADDRESS
244      .WORD 5          ;DEVICE READ INSTRUCTION
245
246      ;THIS IS THE STARTING LOCATION FOR THE RK11 CONTROLLER
247      RK11:  MOV PC,R2      ;SET POINTER TO PARAMETER LIST
248      BR OTHER      ;TRANSFER TO SERVICE ROUTINE
249      .WORD 177406      ;DEVICE WORD COUNT REGISTER
250      .WORD 5          ;DEVICE READ INSTRUCTION
251
252      ;THIS IS A SPARE STARTING LOCATION. IT TRANSFERS TO ADDRESS
253      ;CONTAINED IN THE SWITCH REGISTER.
254      TRANSR: MOV @#SR,PC      ;GO TO INDICATED LOCATION
255
256      ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
257
258      ;THIS IS THE POWER UP VECTOR REQUIRED FOR DEVICE AND
259      POWER:  .WORD RF11      ;ADDRESS OF FIRST LOCATION IN ROM
260      .WORD 340      ;PROCESSOR STATUS LEVEL 7
261
262      ;THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE) CONTROLLER.
263      TC11:  MOV PC,R2      ;SET UP POINTER TO PARAMETER LIST
264      BR TAPES      ;AND TRANSFER TO FIRST ROUTINE
265      .WORD 177344      ;DEVICE WORD COUNT ADDRESS
266      .WORD 4003      ;FIND PREVIOUS BLOCK COMMAND
267      .WORD 100000      ;USED AS DONE INDICATOR
268      .WORD 24000      ;USED AS ERROR INDICATOR/TEST FLAG
269      BR OTHERX      ;THEN TRANSFER TO NEXT ROUTINE
270      .WORD 5          ;DEVICE READ COMMAND

```



```

269
270
271 001450 010702 ;173050 010702
272 001452 000416 ;173052 000416
273 001454 172524 ;173054 172524
274 001456 060017 ;173056 060017
275 001460 000200 ;173060 000200
276 001462 100000 ;173062 100000
277 001464 000413 ;173064 000413
278 001466 060011 ;173066 060011
279 001470 000200 ;173070 000200
280 001472 100000 ;173072 100000
281 001474 000431 ;173074 000431
282 001476 060003 ;173076 060003
283
284
285 001500 010702 ;173100 010702
286 001502 000424 ;173102 000424
287 001504 176716 ;173104 176716
288 001506 000005 ;173106 000005
289
290
291 001510 010200 ;173110 010200
292 001512 005720 ;173112 005720
293 001514 000005 ;173114 000005
294 001516 005720 ;173116 005720
295 001520 016201 ;173120 016201
296 001522 000002 ;173122 000002
297 001524 005311 ;173124 005311
298 001526 012041 ;173126 012041
299 001530 031011 ;173130 031011
300 001532 001776 ;173132 001776
301 001534 005720 ;173134 005720
302 001536 032041 ;173136 032041
303 001540 001063 ;173140 001063
304 001542 000110 ;173142 000110
305
306
307 001544 010702 ;173144 010702
308 001546 000402 ;173146 000402
309 001550 177450 ;173150 177450
310 001552 000005 ;173152 000005
311
312
313 001554 010200 ;173154 010200
314 001556 005720 ;173156 005720
315 001560 005720 ;173160 005720
316 001562 000005 ;173162 000005
317 001564 016201 ;173164 016201
318 001566 000002 ;173166 000002
319 001570 012711 ;173170 012711
320 001572 177000 ;173172 177000
321 001574 011041 ;173174 011041
322 001576 105711 ;173176 105711

```

; THIS IS THE START LOCATION FOR TM11 MAGTAPE CONTROLLER  
TM11: MOV PC,R2 ;SET POINTER TO PARAMETER LIST  
BR TAPES ;AND TRANSFER TO FIRST ROUTINE  
.WORD 172524 ;DEVICE BYTE/RECORD COUNT REGISTER  
.WORD 60017 ;DEVICE REWIND COMMAND  
.WORD 200 ;DEVICE DONE FLAG  
.WORD 100000 ;DEVICE ERROR FLAG BIT  
BR TAPESX ;THEN TRANSFER TO NEXT SERVICE RTN  
.WORD 60011 ;DEVICE FORWARD SPACE COMMAND  
.WORD 200 ;SAME AS ABOVE  
.WORD 100000 ;SAME AS ABOVE  
BR OTHERX ;THEN TRANSFER TO READ/TRANSFER ROUTINE  
.WORD 60003 ;DEVICE READ COMMAND

; THIS IS THE START LOCATION FOR THE RP11 CONTROLLER  
RP11: MOV PC,R2 ;SET POINTER TO PARAMETER LIST  
BR OTHER ;TRANSFER TO TRANSFER ROUTINE  
.WORD 176716 ;DEVICE WORD COUNT REGISTER  
.WORD 5 ;DEVICE READ COMMAND

; THIS IS THE TAPE DEVICE SERVICE ROUTINE.  
TAPES: MOV R2,R0 ;GET ADDRESS OF PARAMETER LIST  
TST (R0)+ ;SKIP TWO WORDS FIRST TIME  
TAPESX: RESET ;RESET ALL DEVICES  
TST (R0)+ ;SKIP OVER BRANCH INSTRUCTION  
MOV 2(R2),R1 ;THEN GET DEVICE WORD/BYTE COUNT ADDRES  
DEC 2R1 ;AND SET TO -1  
MOV (R0)+,-(R1) ;AND THEN ISSUE COMMAND TO DEVICE  
TAPMAT: BIT 2R0,2R1 ;WAIT FOR DEVICE COMPLETION  
BEQ TAPMAT ;BY HANGING IN LOOP  
TST (R0)+ ;AND THEN SKIP DONE FLAG  
BIT (R0)+,-(R1) ;THEN TEST FOR ERROR  
BNE ERROR ;THERE IS ONE  
RETURN: JMP 2R0 ;AND TRANSFER TO FOLLOWING INSTRUCTION

; THIS IS THE STARTING ADDRESS FOR RC11 DISK CONTROLLERS  
RC11: MOV PC,R2 ;SET JP POINTER TO PARAMETER LIST  
BR OTHER ;TRANSFER TO SERVICE RTN  
.WORD 177450 ;DEVICE WORD COUNT REGISTER  
.WORD 5 ;DEVICE READ INSTRUCTION

; THIS ROUTINE PERFORMS THE ACTUAL TRANSFER TO MEMORY OF DATA  
OTHER: MOV R2,R. ;SET POINTER TO LIST IN R0  
TST (R0)+ ;SKIP TWO WORDS FIRST TIME.  
OTHERX: TST (R0)+ ;SKIP PAST BR INSTRUCTION  
RESET ;REST THE WORLD  
MOV 2(R2),R1 ;OBTAIN DEVICE WORD COUNT ADDRESS  
MOV #-1000,2R1 ;THEN OBTAIN LARGE WORD COUNT  
MOV 2R0,-(R1) ;AND PUT COMMAND TO DEVICE  
OTHMAT: TSTB 2R1 ;WAIT FOR DONE FLAG

```

323 001600 100376 ;173200 100376      BPL OTHMAT      ;BY HANGING IN LOOP
324 001602 005711 ;173202 005711      TST R1         ;THEN TEST FOR ERROR
325 001604 100441 ;173204 100441      BMI ERROR     ;GOT PROBLEMS
326 001606 005007 ;173206 005007      CLR PC        ;AND TRANSFER TO ZERO
327
328 ;THIS IS THE STARTING ADDRESS FOR THE PC11 PAPER TAPE CONTROLLER
329 001610 012704 ;173210 012704      KL11: MOV #177560,R4 ;OBTAIN DEVICE ADDRESS
330 001612 177560 ;173212 177560
331 001614 000440 ;173214 000440      BR CKDEV      ;AND TRANSFER TO READER SERVICE ROUTINE
332
333 ;THIS IS THE CASSETTE DEVICE COMMAND TABLE
334 ;TABLE: .BYTE 240 ;COMPARE WORD NOT A COMMAND
335 001616 017640 ;173216 240
336 ;173217 037 ;ILBS+RND+GO
337 001620 002415 ;173220 015 ;SPACE FORWARD BLOCK+GO
338 ;173221 005 ;READ+GO
339 001622 112024 ;173222 024 ;READ+ILBS
340 ;173223 224 ;READ+ILBS+END FLAG
341 ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
342
343 ;THIS IS AN ADDITIONAL POWER VECTOR ADDRESS REQUIRED BY DEVICE
344 001624 173000 ;173224 173000      POWER2: .WORD R11 ;ADDRESS OF BEGINNING OF BOOTSTRAP
345 001626 000340 ;173226 000340      .WORD 340 ;PRIORITY LEVEL 7
346
347 ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE DEVICE #0
348 001630 005004 ;173230 005004      CBOOT: CLR R4 ;LOAD DEVICE NUMBER 0 IN R4
349 001632 012700 ;173232 012700      RESTX: MOV #177500,R0 ;GET DEVICE ADDRESS
350 001634 177500 ;173234 177500
351 001636 000005 ;173236 000005      RESTR1: RESET ;ISSUE RESET INSTRUCTION
352 001640 010410 ;173240 010410      MOV R4,R0 ;LOAD DEVICE WITH UNIT NUMBER
353 001642 012701 ;173242 012701      MOV #TABLE,R1 ;GET FUNNY TABLE OF INSTRUCTIONS
354 001644 173216 ;173244 173216
355 001646 012702 ;173246 012702      MOV #375,R2 ;AND LOAD UP TRANSFER COUNTER
356 001650 000375 ;173250 000375
357 001652 112103 ;173252 112103      MOVB (R1)+,R3 ;THE LOAD UP COMPARATOR
358 001654 112110 ;173254 112110      LOOP1: MOVB (R1)+,R0 ;LOAD DEVICE REGISTER WITH COMMAND
359 001656 100407 ;173256 100407      BMI DONE
360 001660 130310 ;173260 130310      LOOP2: BITB R3,R0 ;HAS COMMAND COMPLETED
361 001662 001776 ;173262 001776      BEQ LOOP2 ;NO WAIT
362 001664 105202 ;173264 105202      INCB R2 ;THEN INCREMENT ADDRESS CTR
363 001666 100772 ;173266 100772      BMI LOOP1 ;IF NEGATIVE, GET COMMAND
364 001670 116012 ;173270 116012      MOVB 2(R0),R2 ;AND STORE DATA AWAY
365 001672 000002 ;173272 000002
366 001674 000771 ;173274 000771      BR LOOP2 ;GO GET ANOTHER BYTE
367 001676 005710 ;173276 005710      DONE: TST R0 ;ANY DEVICE ERRORS
368 001700 100756 ;173300 100756      BMI RESTR1 ;YES, RETRY
369 001702 005002 ;173302 005002      CLR R2 ;CLEAR COMPARE ADDRESS AND TRANSFER ADDRESS
370 001704 120312 ;173304 120312      CMPB R3,R2 ;IT MUST BE 240
371 001706 001377 ;173306 001377      BNE +0 ;NO, THERE WAS AN ERROR
372 001710 000112 ;173310 000112      ERROR: JMP R2 ;NORMAL CASSETTE AND ERROR FOR BULK STORAGE
373
374 ;THIS IS THE STARTING LOCATION FOR THE PC11 CONTROLLER
375 001712 012704 ;173312 012704      PC11: MOV #177550,R4 ;LOAD DEVICE ADDRESS
376 001714 177550 ;173314 177550

```

377	001716	000005	:173316	000005	CKDEV: RESET	:KILL ALL DEVICE ACTION
378	001720	012701	:173320	012701	MOV #160000,R1	:THEN SET UP MEMORY TEST LIMITS
379	001722	160000	:173322	160000		
380	001724	012702	:173324	012702	MOV #6,R2	:AND SET UP POINTER TO TIMEOUT LOCATION
381	001726	000006	:173326	000006		
382	001730	012712	:173330	012712	MOV #340,R2	:AND SET UP VECTOR TO RETURN TO NEXT
383	001732	000340	:173332	000340		
384	001734	010742	:173334	010742	MOV PC, -(R2)	:SAVE THE PC
385	001736	012706	:173336	012706	MOV #24,SP	:AND LOAD UP STACK POINTER
386	001740	000024	:173340	000024		
387	001742	010441	:173342	010441	MOV R4, -(R1)	:AND LOOK FOR END OF MEMORY
388	001744	040601	:173344	040601	BIC SP,R1	:THEN DROP TO XX7752
389	001746	010111	:173346	010111	MOV R1,R1	:AND STORE IN ITSELF
390	001750	011102	:173350	011102	LOOP: MOV R1,R2	:THEN LOAD ADDRESS FOR DATA INSERTION
391	001752	005214	:173352	005214	INC R4	:AND START DEVICE
392	001754	105714	:173354	105714	RDRMAT: TSTB R4	:THEN WAIT FOR CHARACTER AVAILABLE
393	001756	100376	:173356	100376	BPL RDRMAT	:HANGING THERE IF NECESSARY
394	001760	116412	:173360	116412	MOVB 2(R4),R2	:STORE AWAY DATA BYTE
395	001762	000002	:173362	000002		
396	001764	005211	:173364	005211	INC R1	
397	001766	120227	:173366	120227	CMPB R2, #375	:HAS BRANCH OFFSET BEEN STORED
398	001770	000375	:173370	000375		
399	001772	001366	:173372	001366	BNE LOOP	:NO
400	001774	105222	:173374	105222	INCB (R2)+	:YES, ALL DONE
401	001776	END.YA:				
402	001776	000142	:173376	000142	JMP -(R2)	:THEN TRANSFER TO RTN

403  
404  
405  
406  
407  
408  
409  
410  
411  
412  
413  
414  
415  
416  
417  
418  
419  
420  
421  
422  
423  
424  
425  
426  
427  
428  
429  
430  
431  
432  
433  
434  
435  
436  
437  
438  
439  
440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456

002000

: BM8738            BOOTSTRAP            MACY11 27(655) 1-OCT-74 14:50 PAGE 1  
:                    ;DATE: AUG 23, 1974

MAP.YB:  
: THE FOLLOWING IS A REPRODUCTION  
: OF THE ROM PROGRAM FOR BM873YB.  
: IT IS HERE FOR COMPARISON TO THE  
: ACTUAL ROM AND FOR REFERENCE

: THIS IS THE LOADER TO REPLACE THE FOLLOW  
: M792-YA            PAPER TAPE BOOTSTRAP ROM  
: MR11-DB            BULK STORAGE BOOTSTRAP ROM  
: M792-YH            TAII CASSETTE BOOTSTRAP ROM  
: RM873A COMBINATION OF ABOVE ROMS

: PREPHERIAL EXTERNAL PAGE REGISTERS ASSIGNMENTS:

177462	RFMC=	177462	: WORD COUNT REG. FOR RF1
177406	RKMC=	177406	: WORD COUNT REG. FOR RK1
177344	TCMC=	177344	: WORD COUNT REG. FOR TC1
172524	TMCC=	172524	: BYTE/RECORD COUNT FOR T
176716	RPWC=	176716	: WORD COUNT REG. FOR RP1
177450	RCMC=	177450	: WORD COUNT REG. FOR RC1
177560	KLCS=	177560	: CONTROL REG. FOR KL11
177500	TACS=	177500	: CONTROL REG. FOR TAI1 C
177550	PCCS=	177550	: CONTROL REG. FOR PC11
172440	TUCS=	172440	: CONTROL STATUS REG. 1
172442	TUMC=	TUCS+2	: TU16 WORD COUNT REG.

176300	RHCSA=	176300	: CONTROLLER REG. 1 FOR R
176302	RHMCA=	RHCSA+2	
172040	RSCSA=	172040	: CONTROLLER REG. 1 FOR RH
172042	RSMCA=	RSCSA+2	
176700	RPCSA=	176700	: CONTROLLER REG. 1 FOR R
176702	RPWCA=	RPCSA+2	

: FUNCTION VALUE FOR PREPHERALS:

000005	RFREAD=	5	: READ FUNCTION
004003	RNUM=	4003	: REVERSE AND IDENTIFY BL
060017	TMRWD=	60017	: REMIND AND SET 800 BPI
060011	TMFWD=	60011	: FORWARD RECORD COMMAND
060003	TMREAD=	60003	: TM11 READ
000011	DRCLR=	11	: DRIVE CLEAR
000071	RHREAD=	71	: RH11 READ COMMAND
000021	RHPRST=	21	: READ IN PRESET
000031	TUSPAC=	31	: SPACE FORWARD COMMAND F
040000	TUTAPE=	40000	: TAPE BIT IN RH11/RHDT R
001300	TUMODE=	1300	: 800 BPI NORMAL MODE FOR
001000	FCE=	1000	: FRAME COUNT ERROR BIT

: CONSOLE SWITCH REG.  
177570 CSM= 177570

```

457      ; ONLY THE LOW BYTE OF CONSOL SWITCH REGISTER IS
458      ; SELECT THE UNIT NUMBER OF THE DEVICE TO BOOT FR
459      ;
460      ; 173000  .=173000
461      ;
462      ;
463      ; THIS IS THE STARTING ADDRESS FOR RH11/RS03/04 D
464 002000 000405 ;173000 000405 RHRSA: BR 1$ ;ENTRY FOR SELECTING UNI
465 002002 010703 ;173002 010703 RHR58: MOV PC,R3 ;ENTRY TO SELECT UNITS
466 002004 113737 ;173004 113737 MOVB @#CSM,@#RSCSA+10;LOAD UNIT # INS
467 002006 177570 ;173006 177570
468 002010 172050 ;173010 172050
469 002012 000401 ;173012 000401 BR 2$
470 002014 010703 ;173014 010703 1$: MOV PC,R3
471 002016 012700 ;173016 012700 2$: MOV @#RSCSA,R0;SET CONTROL STATUS REG
472 002020 172040 ;173020 172040
473 002022 000526 ;173022 000526 BR RHCOMN
474      ;
475      ; THIS IS THE AUTO LOAD VECTOR
476 002024 173000 ;173024 173000 .WORD RHRSA
477 002026 000340 ;173026 000340 .WORD 340
478      ;
479      ; THIS IS THE STARTING ADDRESS FOR RK11 CONTROLLE
480 002030 000412 ;173030 000412 RK11A: BR 2$ ;ENTRY TO SELECT UNIT 0
481 002032 010703 ;173032 010703 RK11B: MOV PC,R3 ;ENTRY TO SELECT ALL UNI
482      ; SAVE ERROR RETRY ADDRES
483 002034 113705 ;173034 113705 MOVB @#CSM,R5;SET POINTER TO PARAMETE
484 002036 177570 ;173036 177570
485 002040 052705 ;173040 052705 BIS #10,R5 ;SET POSITION BIT
486 002042 000010 ;173042 000010
487 002044 006105 ;173044 006105 1$: ROL R5 ;SHIFT UNIT # TO BIT 13-
488 002046 103376 ;173046 103376 BCC 1$ ;KEEP GOING
489 002050 010537 ;173050 010537 MOV R5,@#RKMC+4;MOVE IN TO RKDA REGI
490 002052 177412 ;173052 177412
491 002054 000401 ;173054 000401 BR 3$ ;SKIP NEXT INSTRUCTION
492 002056 010703 ;173056 010703 2$: MOV PC,R3 ;SAVE ERROR RETRY ADDRES
493 002060 010702 ;173060 010702 3$: MOV PC,R2
494 002062 000546 ;173062 000546 BR OTHERA
495 002064 177406 ;173064 177406 .WORD RKMC
496 002066 000005 ;173066 000005 .WORD RFREAD
497      ;
498      ; THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE)
499 002070 010703 ;173070 010703 TC11: MOV PC,R3 ;SAVE ERROR RETRY ADDRES
500 002072 010702 ;173072 010702 MOV PC,R2
501 002074 000570 ;173074 000570 BR TAPES
502 002076 177344 ;173076 177344 .WORD TCMC
503 002100 000005 ;173100 000005 .WORD RFREAD
504 002102 004003 ;173102 004003 .WORD RNUM
505 002104 100000 ;173104 100000 .WORD 100000 ;DONE MASK
506 002106 024000 ;173106 024000 .WORD 24000 ;ERROR MASK
507      ;
508      ;
509      ;
510      ; TM11 STARTING ADDRESS

```

# J02

MAR 1975  
DZB.MDF.P11

MACY11 27(657) 1-OCT-75 09:30 PAGE 13  
ERROR POINTER TABLE

SEQ 0021

```

511 002110 010703 :173110 010703 TM11: MOV PC,R3 ;SAVE ERROR RETRY ADDRES
512 002112 012737 :173112 012737 MOV #TMRAND,2#TMC-2;REWIND TAPE
513 002114 060017 :173114 060017
514 002116 172522 :173116 172522
515 002120 010702 :173120 010702 MOV PC,R2
516 002122 000555 :173122 000555 BR TAPES
517 002124 172524 :173124 172524 .WORD TMC
518 002126 060003 :173126 060003 .WORD TMRAD ;TM11 READ COMMAND
519 002130 060011 :173130 060011 .WORD TMRAD ;TM11 FORWARD RECORD COM
520 002132 000200 :173132 000200 .WORD 200 ;DONE MASK
521 002134 100000 :173134 100000 .WORD 100000 ;ERROR MASK
522
523 ;THIS IS THE STARTING ADDRESS FOR RF11 CONTROLLE
524 002136 010703 :173136 010703 RF11: MOV PC,R3 ;SAVE ERROR RETRY ADDRES
525 002140 010702 :173140 010702 MOV PC,R2 ;SET POINTER TO PARAMETE
526 002142 000516 :173142 000516 BR OTHERA ;GO TO COMMON SERVICE RO
527 ;ASSUME UNIT 0
528 002144 177462 :173144 177462 .WORD RFMC ;DEVICE WORD COUNT REGIS
529 002146 000005 :173146 000005 .WORD RFREAD ;READ COMMAND
530
531 ;THIS IS THE STARTING ADDRESS FOR RH/TU16/TM02
532 002150 010703 :173150 010703 TU16: MOV PC,R3 ;SAVE ERROR RETRY ADDRES
533 002152 012700 :173152 012700 MOV #TUCS,RO;GET CONTROL STATUS WORD
534 002154 172440 :173154 172440
535 002156 012710 :173156 012710 TU16RE: MOV #RHPRST,(RO);REWIND TAPE CLEAR E
536 002160 000021 :173160 000021
537 002162 012760 :173162 012760 MOV #TUMODE,32(RO);SET 800 BPI NORMA
538 002164 001300 :173164 001300
539 002166 000032 :173166 000032
540 002170 012760 :173170 012760 MOV #-1,6(RO);LOAD FRAME COUNT
541 002172 177777 :173172 177777
542 002174 000006 :173174 000006
543 002176 012710 :173176 012710 MOV #TUSPAC,(RO);SPACE FORWARD
544 002200 000031 :173200 000031
545 002202 105760 :173202 105760 IS: TSTB 12(RO)
546 002204 000012 :173204 000012
547 002206 100375 :173206 100375 BPL IS ;KEEP LOOPING
548 002210 000433 :173210 000433 BR RHCOMM
549
550 ;THIS IS THE STARTING ADDRESS FOR RC11 CONTROLLE
551 002212 010703 :173212 010703 RC11: MOV PC,R3
552 002214 010702 :173214 010702 MOV PC,R2 ;ASSUME UNIT 0
553 002216 000470 :173216 000470 BR OTHERA
554 002220 177450 :173220 177450 .WORD RCMC
555 002222 000005 :173222 000005 .WORD RFRAD
556
557 ;THIS IS THE AUTO LOAD VECTOR
558 002224 173000 :173224 173000 .WORD RHREA
559 002226 000340 :173226 000340 .WORD 340
560
561 ;THIS IS THE STARTING ADDRESS FOR RH11 DEVICE CO
562
563 ;NOTE: IF TM02/TU16 SHOULD BE SELECTED. THE VAL
564 ;IN CONSOL SWITCH REGISTER IS THE POSITIO

```



# K02

MAR 1975  
DZBMD.F.P11

MACY11 27(657) 1-OCT-75 09:30 PAGE 14  
ERROR POINTER TABLE

SEG 0022

```

565      ;
566      ;
567 002230 000405 ;173230 000405 RH11A: BR 1$ ;ENTRY TO SELECT UNIT 0
568 002232 010703 ;173232 010703 RH11B: MOV PC,R3 ;ENTRY TO SELECT ALL UNI
569 002234 113737 ;173234 113737 MOVB @#CSM,@#RHCSA+10;LOAD UNIT # INS
570 002236 177570 ;173236 177570
571 002240 176310 ;173240 176310
572 002242 000401 ;173242 000401 BR 2$
573 002244 010703 ;173244 010703 1$: MOV PC,R3
574 002246 012700 ;173246 012700 2$: MOV #RHCSA,R0
575 002250 176300 ;173250 176300
576 002252 032760 ;173252 032760 RPCOMN: BIT #TUTAPE,26(R0);TAPE UNIT?
577 002254 040000 ;173254 040000
578 002256 000026 ;173256 000026
579 002260 001336 ;173260 001336 BNE TU16RE ;YES. GO TO TAPE LOGIC
580 002262 012710 ;173262 012710 MOV #RHPRST,(R0);RESET DRIVE
581 002264 000021 ;173264 000021
582 002266 012760 ;173266 012760 MOV #14000,32(R0);SET 16 BIT FORMAT
583 002270 014000 ;173270 014000
584 002272 000032 ;173272 000032
585 002274 012710 ;173274 012710 MOV #DRCLR,(R0);CLEAR DRIVE ERROR
586 002276 000011 ;173276 000011
587      ;
588 002300 005720 ;173300 005720 RHCOMN: TST ;(GENERATED IF RS03/04
589 002302 010037 ;173302 010037 MOV (R0)+ ;MOVE TO WORD COUNT ADDR
590 002304 000002 ;173304 000002 MOV R0,@#2 ;FAKE CALLING SEQUENCE
591 002306 012737 ;173306 012737 MOV #RHREAD,@#4
592 002310 000071 ;173310 000071
593 002312 000004 ;173312 000004
594 002314 005002 ;173314 005002 CLR R2 ;FOR FLAG AND POINTER TO
595 002316 000430 ;173316 000430 BR OTHERA
596      ;
597      ;
598      ;
599 002320 000405 ;173320 000405 RHRPA: BR 1$ ;ENTRY FOR SELECT UNIT 0
600 002322 010703 ;173322 010703 RHRPB: MOV PC,R3 ;ENTRY TO SELECT ALL UNI
601 002324 113737 ;173324 113737 MOVB @#CSM,@#RPPSA+10;LOAD UNIT # INS
602 002326 177570 ;173326 177570
603 002330 176710 ;173330 176710
604 002332 000401 ;173332 000401 BR 2$
605 002334 010703 ;173334 010703 1$: MOV PC,R3
606 002336 012700 ;173336 012700 2$: MOV #RPPSA,R0
607 002340 176700 ;173340 176700
608 002342 000743 ;173342 000743 BR RPCOMN
609      ;
610      ;
611 002344 013707 ;173344 013707 ;ENTRY TO BRANCH TO THE PC SELECTED BY CONSOL SM
612 002346 177570 ;173346 177570 CSRG0: MOV @#CSM,PC
613      ;
614      ;
615      ;
616      ;
617 002350 000405 ;173350 000405 RP11A: BR 1$ ;ENTRY TO SELECT UNIT 0
618 002352 010703 ;173352 010703 RP11B: MOV PC,R3 ;ENTRY TO SELECT ALL UNI
    
```

```

619 002354 113705 ;173354 113705      MOVB      @#CSM,R5
620 002356 177570 ;173356 177570
621 002360 000305 ;173360 000305      SWAB      R5          ;GET UNIT # INTO HIGH BY
622 002362 000402 ;173362 000402      BR        3$
623 002364 010703 ;173364 010703      1$:      MOV      PC,R3
624 002366 005005 ;173366 005005      CLR      R5
625 002370 010702 ;173370 010702      3$:      MOV      PC,R2
626 002372 000403 ;173372 000403      BP      OTHER
627 002374 176716 ;173374 176716      .WORD   RPWC
628 002376 000005 ;173376 000005      .WORD   RFREAD
629
630 002400 005005 ;173400 005005      OTHERA: CLR      R5          ;SET TO UNIT 0
631 002402 010200 ;173402 010200      OTHER:  MOV      R2,R0      ;RD POINT AT WORD COUNT
632 002404 005720 ;173404 005720      TST      (R0)+          ;POINT TO PARAMETER LIST
633 002406 012001 ;173406 012001      MOV      (R0)+,R1      ;MOVE WORD COUNT ADDRESS
634 002410 012711 ;173410 012711      MOV      #-256,R2      ;LOAD WORD COUNT
635 002412 177000 ;173412 177000
636 002414 051005 ;173414 051005      BIS      (R0),R5      ;COMBINE UNIT # WITH COM
637 002416 010541 ;173416 010541      MOV      R5,-(R1)      ;LOAD READ COMMAND
638 002420 032711 ;173420 032711      BIT      #100200,(R1)  ;CHECK FOR ERROR AND
639 002422 100200 ;173422 100200
640 002424 001775 ;173424 001775      BEQ      .-4          ;WAIT UNTIL COMPLETE
641 002426 100012 ;173426 100012      BPL      1$          ;NO ERROR
642 002430 005702 ;173430 005702      TST      R2          ;WAS IT CALLED BY MASS B
643 002432 001024 ;173432 001024      BNE      AGAIN        ;NO ERROR
644 002434 032761 ;173434 032761      BIT      #TUTAPE,26(R1); IS TU16?
645 002436 040000 ;173436 040000
646 002440 000026 ;173440 000026
647 002442 001420 ;173442 001420      BEQ      AGAIN        ;NO. ERROR
648 002444 022761 ;173444 022761      CMP      #FCE,14(R1)  ;ARE WE READ A SHORT
649 002446 001000 ;173446 001000
650 002450 000014 ;173450 000014
651 002452 001014 ;173452 001014      BNE      AGAIN        ;SOME OTHER ERROR
652 002454 005007 ;173454 005007      1$:      CLR      PC          ;O.K.
653
654
655 002456 010200 ;173456 010200      ;THIS IS THE TAPE DEVICE SERVICE ROUTINE
656 002460 005720 ;173460 005720      TAPES:  MOV      R2,R0      ;GET THE ADDRESS OF THE
657 002462 012001 ;173462 012001      TST      (R0)+          ;STEP TO LAST COMMAND
658 002464 005311 ;173464 005311      MOV      (R0)+,R1      ;GET THE WORD COUNT ADDR
659 002466 005720 ;173466 005720      DEC      (R1)          ;SET UP TO ADVANCE 1 REC
660 002470 012041 ;173470 012041      TST      (R0)+          ;MOVE R0 TO FIRST COMMAN
661 002472 031011 ;173472 031011      MOV      (R0)+,-(R1)   ;LOAD COMMAND REG.
662 002474 001776 ;173474 001776      BIT      (R0),(R1)     ;DONE?
663 002476 005720 ;173476 005720      BEQ      .-2          ;NO. KEEP LOOPING
664 002500 031041 ;173500 031041      TST      (R0)+          ;YES. CHECK FOR ERROR
665 002502 001736 ;173502 001736      BIT      (R0),(R1)     ;ANY ERROR?
666 002504 000005 ;173504 000005      BEQ      OTHERA      ;NO ERROR.- TRY TO READ
667
668 002506 000113 ;173506 000113      AGAIN:  RESET
669
670
671 002510 012704 ;173510 012704      JMP      (R3)          ;ERROR RETURN
672 002512 177560 ;173512 177560      ;THIS IS THE STARTING ADDRESS FOR PC11 PAPER TAP
        KL11:  MOV      #KLCS,R4;OBTAIN CONTROL REG.

```

```

673 002514 000443 ;173514 000443      BR      CKDEV      ;AND TRANSFER TO READER
674      :
675      :
676      :
677      :
678 002516 .BYTE 240 ;173516 240      TABLE: .BYTE 240      ;COMPARE WORD NOT A COMM
679 002517 .BYTE 037 ;173517 037      .BYTE 37      ;ILBS+RWD+GO
680 002520 .BYTE 015 ;173520 015      .BYTE 15      ;SPACE FORWARD BLOCK+GO
681 002521 .BYTE 005 ;173521 005      .BYTE 5       ;READ
682 002522 .BYTE 024 ;173522 024      .BYTE 24      ;READ +ILBS
683 002523 .BYTE 224 ;173523 224      .BYTE 224     ;READ+ILBS+END FLAG
684      :
685      :
686 002524 000404 ;173524 000404      CBOOTA: BR      15      ;SELECT UNIT 0
687 002526 113704 ;173526 113704      CBOOTB: MOVB   2#CSW,R4;SELECT UNITS
688 002530 177570 ;173530 177570
689 002532 000304 ;173532 000304      SWAB      R4
690 002534 000401 ;173534 000401      BR      RESETX
691 002536 005004 ;173536 005004      15:      CLR      R4
692 002540 012700 ;173540 012700      RESETX: MOV    #TACS,R0;GET CONTROL REG.
693 002542 177500 ;173542 177500
694 002544 000005 ;173544 000005      RESTRT: RESET
695 002546 010410 ;173546 010410      MOV      R4,(R0);SELECT UNIT
696 002550 012701 ;173550 012701      MOV      #TABLE,R1
697 002552 173516 ;173552 173516
698 002554 012702 ;173554 012702      MOV      #375,R2 ;LOAD TRANSFER COUNTER
699 002556 000375 ;173556 000375
700 002560 112103 ;173560 112103      MOVB     (R1)+,R3;LOAD COMPARATOR
701 002562 112110 ;173562 112110      LOOP1:  MOVB     (R1)+,(R0);LOAD COMMAND
702 002564 100407 ;173564 100407      BMI     DONE
703 002566 130310 ;173566 130310      LOOP2:  BITB     R3,(R0);COMMAND COMPLETE?
704 002570 001776 ;173570 001776      BEQ     LOOP2 ;NO. WAIT
705 002572 105202 ;173572 105202      INCB     R2 ;INCREMENT ADDRESS CTR.
706 002574 100772 ;173574 100772      BMI     LOOP1 ;IF (-), GET COMMAND
707 002576 116012 ;173576 116012      MOVB     2(R0),(R2);STORE DATA
708 002600 000002 ;173600 000002
709 002602 000771 ;173602 000771
710 002604 005710 ;173604 005710      DONE:   BR      LOOP2 ;GET ANOTHER BYTE
711 002606 100756 ;173606 100756      TST     (R0);ANY ERROR?
712 002610 005002 ;173610 005002      BMI     RESTRT ;YES, RETRY
713 002612 120312 ;173612 120312      CLR     R2 ;CLEAR COMPARE ADDRESS
714 002614 001377 ;173614 001377      CMPB    R3,(R2);IT MUST BE 240
715 002616 000112 ;173616 000112      BNE     ERROR: JMP    (R2)
716      :
717      :
718 002620 012704 ;173620 012704      PC11:   MOV    #PCC5,R4
719 002622 177550 ;173622 177550
720 002624 000005 ;173624 000005      CKDEV:  RESET
721 002626 012701 ;173626 012701      MOV     #160000,R1;SET UP MEMORY TEST LI
722 002630 160000 ;173630 160000
723 002632 012702 ;173632 012702      MOV     #6,R2 ;SET UP POINTER TO TIME0
724 002634 000006 ;173634 000006
725 002636 012712 ;173636 012712      MOV     #340,(R2);SET UP VECTOR TO RETUR
726 002640 000340 ;173640 000340

```

```

727 002642 010742 ;173642 010742      MOV      PC,-(R2);SAVE PC
728 002644 012706 ;173644 012706      MOV      #24,SP ;LOAD UP STACK POINTER
729 002646 000024 ;173646 000024
730 002650 010441 ;173650 010441      MOV      R4,-(R1);LOOK FOR END OF MEMORY
731 002652 040601 ;173652 040601      BIC      SP,R1 ;THEN DROP TO XX752
732 002654 010111 ;173654 010111      MOV      R1,(R1);AND STORE IN ITSELF
733 002656 011102 ;173656 011102      LOOP:    MOV      (R1),R2
734 002660 005214 ;173660 005214      RDRMAT: INC      (R4) ;START DEVICE
735 002662 105714 ;173662 105714      TSTB    (R4) ;WAIT
736 002664 100376 ;173664 100376      BPL     RDRMAT
737 002666 116412 ;173666 116412      MOVVB   2(R4),(R2);SAVE THE DATA
738 002670 000002 ;173670 000002
739 002672 005211 ;173672 005211      INC     (R1)
740 002674 120227 ;173674 120227      CMPB   R2,#375
741 002676 000375 ;173676 000375
742 002700 001366 ;173700 001366      BNE    LOOP ;NO
743 002702 105222 ;173702 105222      INCB   (R2)+ ;YES
744 002704 000142 ;173704 000142      JMP    -(R2)
745 002706 000000 ;173706 000000 ;THIS AREA IS UNUSED
746 002710 000000 ;173710 000000 ;THIS AREA IS UNUSED
747 002712 000000 ;173712 000000 ;THIS AREA IS UNUSED
748 002714 000000 ;173714 000000 ;THIS AREA IS UNUSED
749 002716 000000 ;173716 000000 ;THIS AREA IS UNUSED
750 002720 000000 ;173720 000000 ;THIS AREA IS UNUSED
751 002722 000000 ;173722 000000 ;THIS AREA IS UNUSED
752 002724 000000 ;173724 000000 ;THIS AREA IS UNUSED
753 002726 000000 ;173726 000000 ;THIS AREA IS UNUSED
754 002730 000000 ;173730 000000 ;THIS AREA IS UNUSED
755 002732 000000 ;173732 000000 ;THIS AREA IS UNUSED
756 002734 000000 ;173734 000000 ;THIS AREA IS UNUSED
757 002736 000000 ;173736 000000 ;THIS AREA IS UNUSED
758 002740 000000 ;173740 000000 ;THIS AREA IS UNUSED
759 002742 000000 ;173742 000000 ;THIS AREA IS UNUSED
760 002744 000000 ;173744 000000 ;THIS AREA IS UNUSED
761 002746 000000 ;173746 000000 ;THIS AREA IS UNUSED
762 002750 000000 ;173750 000000 ;THIS AREA IS UNUSED
763 002752 000000 ;173752 000000 ;THIS AREA IS UNUSED
764 002754 000000 ;173754 000000 ;THIS AREA IS UNUSED
765 002756 000000 ;173756 000000 ;THIS AREA IS UNUSED
766 002760 000000 ;173760 000000 ;THIS AREA IS UNUSED
767 002762 000000 ;173762 000000 ;THIS AREA IS UNUSED
768 002764 000000 ;173764 000000 ;THIS AREA IS UNUSED
769 002766 000000 ;173766 000000 ;THIS AREA IS UNUSED
770 002770 000000 ;173770 000000 ;THIS AREA IS UNUSED
771 002772 000000 ;173772 000000 ;THIS AREA IS UNUSED
772 002774 000000 ;173774 000000 ;THIS AREA IS UNUSED
773 002776 END.YB:
774 002776 000000 ;173776 000000 ;THIS AREA IS UNUSED

```

```

775 003000 MAP.YC:
776 ;THE FOLLOWING 1000 LOCATIONS ARE
777 ;A REPRODUCTION OF THE ROM PROGRAM
778 ;FOR THE BMB73YC. THE FIRST 400 LOCATIONS
779 ;ARE AN EXACT COPY OF THE BMB73YA. THE
780 ;REMAINING 400 LOCATIONS ARE
781 ;THE DDCMP BOOTSTRAP ROM PROGRAM.
782 ;IT IS HERE FOR COMPARISON TO
783 ;ACTUAL ROM AND FOR REFERENCE.
784 ;173000 .=173000
785 ;STARTING ADDRESS FOR BOOTSTRAP
786 ;THIS LOADER IS DESIGNED FOR THE RESTART MODULE M873.
787 ;IT FUNCTIONALLY REPLACES THE FOLLOWING ROMS:
788 ;M792-YA - PAPER TAPE BOOTSTRAP FOR PC11,KL11
789 ;MR11-DB BULK STORAGE BOOTSTRAP ROM
790 ;M792-YH TAIL CASSETTE BOOTSTRAP ROM
791 ;REGISTER DEFINITIONS
792 RC= X0
793 R1= X1
794 R2= X2
795 R3= X3
796 R4= X4
797 R5= X5
798 SP= X6
799 PC= X7
800 SR= 177570 ;PROCESSOR SWITCH REGISTER
801 ;STARTING LOCATION FOR RF11 DISK
802 RF11: MOV PC,R2 ;SET POINTER TO PARAMETER LISTS
803 BR OTHER ;TRANSFER TO SERVICE ROUTINE
804 .WORD 177462 ;DEVICE WORD COUNT ADDRESS
805 .WORD 5 ;DEVICE READ INSTRUCTION
806 ;THIS IS THE STARTING LOCATION FOR THE RK11 CONTROLLER
807 RK11: MOV PC,R2 ;SET POINTER TO PARAMETER LIST
808 BR OTHER ;TRANSFER TO SERVICE ROUTINE
809 .WORD 177406 ;DEVICE WORD COUNT REGISTER
810 .WORD 5 ;DEVICE READ INSTRUCTION
811 ;THIS IS A SPARE STARTING LOCATION. IT TRANSFERS TO ADDRESS
812 ;CONTAINED IN THE SWITCH REGISTER.
813 TRANSR: MOV @SR,PC ;GO TO INDICATED LOCATION
814 ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
815 ;THIS IS THE POWER UP VECTOR REQUIRED FOR DEVICE AND
816 POWER: .WORD RF11 ;ADDRESS OF FIRST LOCATION IN ROM
817 .WORD 340 ;PROCESSOR STATUS LEVEL 7
818 ;THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE) CONTROLLER.
819 TC11: MOV PC,R2 ;SET UP POINTER TO PARAMETER LIST
820 BR TAPES ;AND TRANSFER TO FIRST ROUTINE
821 .WORD 177344 ;DEVICE WORD COUNT ADDRESS
822 .WORD 4003 ;FIND PREVIOUS BLOCK COMMAND
823 .WORD 190000 ;USED AS DONE INDICATOR
824 .WORD 24000 ;USED AS ERROR INDICATOR/TEST FLAG

```

829	003044	000445	:173044	003445	BR OTHERX	:THEN TRANSFER TO NEXT ROUTINE
830	003046	000005	:173046	000005	.WORD 5	:DEVICE READ COMMAND
831						
832						:THIS IS THE START LOCATION FOR TM11 MAGTAPE CONTROLLER
833	003050	010702	:173050	010702	TM11: MOV PC,R2	:SET POINTER TO PARAMETER LIST
834	003052	000416	:173052	000416	BR TAPES	:AND TRANSFER TO FIRST ROUTINE
835	003054	172524	:173054	172524	.WORD 172524	:DEVICE BYTE/RECORD COUNT REGISTER
836	003056	060017	:173056	060017	.WORD 60017	:DEVICE REWIND COMMAND
837	003060	000200	:173060	000200	.WORD 200	:DEVICE DONE FLAG
838	003062	100000	:173062	100000	.WORD 100000	:DEVICE ERROR FLAG BIT
839	003064	000413	:173064	000413	BR TAPESX	:THEN TRANSFER TO NEXT SERVICE RTN
840	003066	060011	:173066	060011	.WORD 60011	:DEVICE FORWARD SPACE COMMAND
841	003070	000200	:173070	000200	.WORD 200	:SAME AS ABOVE
842	003072	100000	:173072	100000	.WORD 100000	:SAME AS ABOVE
843	003074	000431	:173074	000431	BR OTHERX	:THEN TRANSFER TO READ/TRANSFER ROUTINE
844	003076	060003	:173076	060003	.WORD 60003	:DEVICE READ COMMAND
845						
846						:THIS IS THE START LOCATION FOR THE RP11 CONTROLLER
847	003100	010702	:173100	010702	RP11: MOV PC,R2	:SET POINTER TO PARAMETER LIST
848	003102	000424	:173102	000424	BR OTHER	:TRANSFER TO TRANSFER ROUTINE
849	003104	176716	:173104	176716	.WORD 176716	:DEVICE WORD COUNT REGISTER
850	003106	000005	:173106	000005	.WORD 5	:DEVICE READ COMMAND
851						
852						:THIS IS THE TAPE DEVICE SERVICE ROUTINE.
853	003110	010200	:173110	010200	TAPES: MOV R2,R0	:GET ADDRESS OF PARAMETER LIST
854	003112	005720	:173112	005720	TST (R0)+	:SKIP TWO WORDS FIRST TIME
855	003114	000005	:173114	000005	TAPESX: RESET	:RESET ALL DEVICES
856	003116	005720	:173116	005720	TST (R0)+	:SKIP OVER BRANCH INSTRUCTION
857	003120	016201	:173120	016201	MOV 2(R2),R1	:THEN GET DEVICE WORD/BYTE COUNT ADDRES
858	003122	000002	:173122	000002		
859	003124	005311	:173124	005311	DEC R1	:AND SET TO -1
860	003126	012041	:173126	012041	MOV (R0)+,-(R1)	:AND THEN ISSUE COMMAND TO DEVICE
861	003130	031011	:173130	031011	TAPMAT: BIT R0,R1	:WAIT FOR DEVICE COMPLETION
862	003132	001776	:173132	001776	BEQ TAPMAT	:BY HANGING IN LOOP
863	003134	005720	:173134	005720	TST (R0)+	:AND THEN SKIP DONE FLAG
864	003136	032041	:173136	032041	BIT (R0)+,-(R1)	:THEN TEST FOR ERROR
865	003140	001063	:173140	001063	BNE ERROR	:THERE IS ONE
866	003142	000110	:173142	000110	RETURN: JMP R0	:AND TRANSFER TO FOLLOWING INSTRUCTION
867						
868						:THIS IS THE STARTING ADDRESS FOR RC11 DISK CONTROLLERS
869	003144	010702	:173144	010702	RC11: MOV PC,R2	:SET UP POINTER TO PARAMETER LIST
870	003146	000402	:173146	000402	BR OTHER	:TRANSFER TO SERVICE RTN
871	003150	177450	:173150	177450	.WORD 177450	:DEVICE WORD COUNT REGISTER
872	003152	000005	:173152	000005	.WORD 5	:DEVICE READ INSTRUCTION
873						
874						:THIS ROUTINE PERFORMS THE ACTUAL TRANSFER TO MEMORY OF DATA
875	003154	010200	:173154	010200	OTHER: MOV R2,R0	:SET POINTER TO LIST IN R0
876	003156	005720	:173156	005720	TST (R0)+	:SKIP TWO WORDS FIRST TIME.
877	003160	005720	:173160	005720	OTHERX: TST (R0)+	:SKIP PAST BR INSTRUCTION
878	003162	000005	:173162	000005	RESET	:REST THE WORLD
879	003164	016201	:173164	016201	MOV 2(R2),R1	:OBTAIN DEVICE WORD COUNT ADDRESS
880	003166	000002	:173166	000002		
881	003170	012711	:173170	012711	MOV #-1000,R1	:THEN OBTAIN LARGE WORD COUNT
882	003172	177000	:173172	177000		



```

883 003174 011041 ;173174 011041      MOV 2R0,-(R1) ;AND PUT COMMAND TO DEVICE
884 003176 105711 ;173176 105711      OTHMAT: TSTB 2R1 ;WAIT FOR DONE FLAG
885 003200 100376 ;173200 100376      BPL OTHMAT ;BY HANGING IN LOOP
886 003202 005711 ;173202 005711      TST 2R1 ;THEN TEST FOR ERROR
887 003204 100441 ;173204 100441      BMI ERROR ;GOT PROBLEMS
888 003206 005007 ;173206 005007      CLR PC ;AND TRANSFER TO ZERO
889
890 ;THIS IS THE STARTING ADDRESS FOR THE PC11 PAPER TAPE CONTROLLER
891 003210 012704 ;173210 012704      KL11: MOV #177560,R4 ;OBTAIN DEVICE ADDRESS
892 003212 177560 ;173212 177560
893 003214 000440 ;173214 000440      BR CKDEV ;AND TRANSFER TO READER SERVICE ROUTINE
894
895 ;
896 ;THIS IS THE CASSETTE DEVICE COMMAND TABLE
897 003216 017640 ;173216 240      TABLE: .BYTE 240 ;COMPARE WORD NOT A COMMAND
898 ;173217 037 ;.BYTE 37 ;ILBS+RMD+GO
899 003220 002415 ;173220 015 ;.BYTE 15 ;SPACE FORWARD BLOCK+GO
900 ;173221 005 ;.BYTE 5 ;READ+GO
901 003222 112024 ;173222 024 ;.BYTE 24 ;READ+ILBS
902 ;173223 224 ;.BYTE 224 ;READ+ILBS+END FLAG
903 ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
904
905 ;THIS IS AN ADDITIONAL POWER VECTOR ADDRESS REQUIRED BY DEVICE
906 003224 173000 ;173224 173000      POWER2: .WORD RF11 ;ADDRESS OF BEGINNING OF BOOTSTRAP
907 003226 000340 ;173226 000340      .WORD 340 ;PRIORITY LEVEL 7
908
909 ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE DEVICE #0
910 003230 005004 ;173230 005004      CBOOT: CLR R4 ;LOAD DEVICE NUMBER 0 IN R4
911 003232 012700 ;173232 012700      RESTX: MOV #177500,R0 ;GET DEVICE ADDRESS
912 003234 177500 ;173234 177500
913 003236 000005 ;173236 000005      RESTRT: RESET ;ISSUE RESET INSTRUCTION
914 003240 010410 ;173240 010410      MOV R4,2R0 ;LOAD DEVICE WITH UNIT NUMBER
915 003242 012701 ;173242 012701      MOV #TABLE,R1 ;GET FUNNY TABLE OF INSTRUCTIONS
916 003244 173216 ;173244 173216
917 003246 012702 ;173246 012702      MOV #375,R2 ;AND LOAD UP TRANSFER COUNTER
918 003250 000375 ;173250 000375
919 003252 112103 ;173252 112103      LOOP1: MOV8 (R1)+,R3 ;THE LOAD UP COMPARATOR
920 003254 112110 ;173254 112110      MOV8 (R1)+,2R0 ;LOAD DEVICE REGISTER WITH COMMAND
921 003256 100407 ;173256 100407      BMI DONE
922 003260 130310 ;173260 130310      LOOP2: BITB R3,2R0 ;HAS COMMAND COMPLETED
923 003262 001776 ;173262 001776      BEQ LOOP2 ;NO, WAIT
924 003264 105202 ;173264 105202      INCB R2 ;THEN INCREMENT ADDRESS CTR
925 003266 100772 ;173266 100772      BMI LOOP1 ;IF NEGATIVE, GET COMMAND
926 003270 116012 ;173270 116012      MOV8 2(R0),2R2 ;AND STORE DATA AWAY
927 003272 000002 ;173272 000002
928 003274 000771 ;173274 000771      BR LOOP2 ;GO GET ANOTHER BYTE
929 003276 005710 ;173276 005710      DONE: TST 2R0 ;ANY DEVICE ERRORS
930 003300 100756 ;173300 100756      BMI RESTRT ;YES, RETRY
931 003302 005002 ;173302 005002      CLR R2 ;CLEAR COMPARE ADDRESS AND TRANSFER ADDRESS
932 003304 120312 ;173304 120312      CMPB R3,2R2 ;IT MUST BE 240
933 003306 001377 ;173306 001377      BNE +0 ;NO, THERE WAS AN ERROR
934 003310 000112 ;173310 000112      ERROR: JMP 2R2 ;NORMAL CASSETTE AND ERROR FOR BULK STORAGE
935
936 ;THIS IS THE STARTING LOCATION FOR THE PC11 CONTROLLER

```

# E03

MAR 1975  
DZBMD.F.P11

MACY11 27(657) 1-OCT-75 09:30 PAGE 21  
ERROR POINTER TABLE

SEQ 0029

```
937 003312 012704 :173312 012704 PC11: MOV #177550,R4 ;LOAD DEVICE ADDRESS
938 003314 177550 :173314 177550
939 003316 000005 :173316 000005 CKDEV: RESET ;KILL ALL DEVICE ACTION
940 003320 012701 :173320 012701 MOV #160000,R1 ;THEN SET UP MEMORY TEST LIMITS
941 003322 160000 :173322 160000
942 003324 012702 :173324 012702 MOV #6,R2 ;AND SET UP POINTER TO TIMEOUT LOCATION
943 003326 000006 :173326 000006
944 003330 012712 :173330 012712 MOV #340,R2 ;AND SET UP VECTOR TO RETURN TO NEXT
945 003332 000340 :173332 000340
946 003334 010742 :173334 010742 MOV PC, -(R2) ;SAVE THE PC
947 003336 012706 :173336 012706 MOV #24,SP ;AND LOAD UP STACK POINTER
948 003340 000024 :173340 000024
949 003342 010441 :173342 010441 MOV R4, -(R1) ;AND LOOK FOR END OF MEMORY
950 003344 040601 :173344 040601 BIC SP,R1 ;THEN DROP TO XX7752
951 003346 010111 :173346 010111 MOV R1,R1 ;AND STORE IN ITSELF
952 003350 011102 :173350 011102 LOOP: MOV R1,R2 ;THEN LOAD ADDRESS FOR DATA INSERTION
953 003352 005214 :173352 005214 INC R4 ;AND START DEVICE
954 003354 105714 :173354 105714 RDRMAT: TSTB R4 ;THEN WAIT FOR CHARACTER AVAILABLE
955 003356 100376 :173356 100376 BPL RDRMAT ;HANGING THERE IF NECESSARY
956 003360 116412 :173360 116412 MOVB 2(R4),R2 ;STORE AWAY DATA BYTE
957 003362 000002 :173362 000002
958 003364 005211 :173364 005211 INC R1
959 003366 120227 :173366 120227 CMPB R2,#375 ;HAS BRANCH OFFSET BEEN STORED
960 003370 000375 :173370 000375
961 003372 001366 :173372 001366 BNE LOOP ;NO
962 003374 105222 :173374 105222 INCB (R2)+ ;YES, ALL DONE
963 003376 000142 :173376 000142 JMP -(R2) ;THEN TRANSFER TO RTN
```

964  
965 ;THE FOLLOWING 400 LOCATIONS ARE  
966 ;A REPRODUCTION OF THE DDCMP BOOT-  
967 ;STRAP ROM. IT IS HERE FOR COM-  
968 ;PARISON TO THE ACTUAL ROM AND  
969 ;FOR REFERENCE.

970  
971  
972  
973  
974  
975  
976  
977  
978  
979  
980  
981  
982  
983  
984  
985  
986  
987  
988  
989  
990

.....  
COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A  
LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND  
CAN BE COPIED (WITH INCLUSION OF DEC'S  
COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM,  
EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING  
BY DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO  
CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED  
AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR  
RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH  
IS NOT SUPPLIED BY DEC.

VERSION 01

STUART WECKER 01/22/75

991  
992  
993  
994  
995  
996  
997  
998  
999  
1000  
1001  
1002  
1003  
1004  
1005  
1006  
1007  
1008  
1009  
1010  
1011  
1012  
1013  
1014  
1015  
1016  
1017  
1018  
1019  
1020  
1021  
1022  
1023  
1024  
1025  
1026  
1027  
1028  
1029  
1030  
1031  
1032  
1033  
1034  
1035  
1036  
1037  
1038  
1039  
1040  
1041  
1042  
1043  
1044

DIGITAL EQUIPMENT CORPORATION  
COMPUTER NETWORK FACILITIES  
DOWN-LINE LOADING PROGRAM

THIS PROGRAM LOADS COMPUTER MEMORY FROM DATA SENT OVER  
A DATA COMMUNICATIONS LINK. IT SENDS AND RECEIVES  
MESSAGES IN DDCMP BOOT FORMAT. THE PRIMARY BOOT ONLY  
LOADS A SINGLE BLOCK, THE SECONDARY BOOT, WHICH  
THEN REQUESTS AND LOADS THE DESIRED PROGRAM.

CURRENT VERSION DDCMP: 3.0 - MAY 7, 1974

THE BOOTSTRAP MESSAGES ARE OF THE FORM:

SYN, SYN, DLE, CNT, F, S, FILL, FILL, ADDR, CRC1, DATA, CRC2

ALL ITEMS ARE 8-BITS LONG UNLESS OTHERWISE SPECIFIED

SYN-THE SYNC CHARACTER-SYNC-226, ASYNC-377

DLE-THE BOOT HEADER CHARACTER-OCTAL 220

CNT-THE 14-BIT COUNT FIELD-LENGTH OF DATA FIELD

F-THE FINAL BIT-LINK CONTROL

S-THE SELECT BIT-LINK CONTROL

FILL-A FILL CHARACTER-OCTAL 000

ADDR-THE STATION ADDR-FOR PT. TO PT.=1

CRC1-THE 16-BIT CRC-16 COMPUTED ON DLE THROUGH ADDR

DATA-THE BOOT DATA AS FOLLOWS:

CODE, INFO

ONLY THE FOLLOWING CODES ARE USED BY THE  
PRIMARY BOOT

CODE=10 REQUEST SECONDARY PROGRAM

INFO=DEVICE TYPE, STATION ADDRESS

DEVICE TYPE-DP=0, DU=2, DL=4, DQ=6

STATION ADDRESS=1

CODE=0 PROGRAM LOAD WITH TRANSFER ADDRESS

INFO=BLKNO, BLK LDADDR, IMAGE DATA, TRANS ADDR

BLKNO=0

BLOCK LDADDR=6

TRANS ADDR=6

HEADER COUNT > OR = TO 10.

ADDRESSES ARE 4 BYTES-32 BITS-LOW BIT FIRST

CRC2-THE 16-BIT CRC-16 COMPUTED ON THE DATA FIELD ONLY

OPTION SWITCHES:

DEVICE-DP11, DU11, DL11

CRC-KG11, SCRC

REGISTER DEFINITIONS

; 000000 RO=X0

; BLOCK LOAD ADDR

```

1045      :      000001  R1=X1      ;DEVICE CSR ADDRESS
1046      :      000002  R2=X2      ;CRC CALC TEMP
1047      :      000003  R3=X3      ;SOFTWARE CRC
1048      :      000004  R4=X4      ;BLOCK CHAR COUNT
1049      :      000005  R5=X5      ;CRC CALC TEMP
1050      :      000006  SP=X6      ;STACK ADDR
1051      :      000007  PC=X7      ;LOCATION COUNTER
1052      :
1053      :      LITERALS
1054      :
1055      :      000001  $STADR=1    ;STATION ADDR
1056      :      177570  $SMR=177570 ;SWITCH REGISTER ADDR
1057      :      000226  $SYN=226   ;SYNC CHARACTER
1058      :      000220  $DLE=220   ;DDCMP DLE CHARACTER
1059      :      000400  $STRIP=400
1060      :
1061      :      THE STACK IS USED AS FOLLOWS:
1062      :      STACK-2:FOR JSR TO GET ROUTINE
1063      :      STACK-4:TEMP FOR CRC CALCULATION
1064      :
1065      :      START OF BOOT PROGRAM
1066      :
1067      :      START1-DEVICE UNIT 0-NORMAL CONFIGURATION
1068      :      START2-USE SWITCH REG AS DEVICE DISPLACEMENT
1069      :      I.E. #0-0,#1-10,#2-20
1070      :
1071      :      =173400
1072  003400  012700  :173400  012700  START1: MOV      (PC)+,R0      ;NON ZERO VALUE TO R0
1073  003402  005000  :173402  005000  START2: CLR      R0          ;CLEAR R0
1074  003404  000005  :173404  000005          RESET          ;RESET SYS, MEM MGT, ETC...
1075  003406  012706  :173406  012706          MOV      #17776,SP      ;STACK AT 4K-2
1076  003410  017776  :173410  017776
1077      :
1078      :      FIND THE DU-11 IN THE FLOATING ADDRESS SPACE
1079      :
1080  003412  010702  :173412  010702          MOV      PC,R2          ;CURRENT PC
1081  003414  062702  :173414  062702          ADD      #DEVTAB-. ,R2  ;DEVICE TABLE ADDR
1082  003416  000360  :173416  000360
1083  003420  012703  :173420  012703          MOV      #6,R3         ;TRAP PS ADDR
1084  003422  000006  :173422  000006
1085  003424  005013  :173424  005013          CLR      (R3)          ;CLEAR NEW PS
1086  003426  010243  :173426  010243          MOV      R2,-(R3)      ;TABLE ADDR TO LOC 4
1087  003430  160313  :173430  160313          SUB      R3,(R3)       ;SUB TO TRAP RTN
1088  003432  005303  :173432  005303          DEC      R3            ;LEAVE CNT 3 FOR LOOP
1089  003434  012701  :173434  012701          MOV      #160010,R1    ;START SEARCH ADDR
1090  003436  160010  :173436  160010
1091  003440  005711  :173440  005711  DEVELOP: TST      (R1)      ;IS DEVICE THERE
1092  003442  111204  :173442  111204          MOV      (R2),R4       ;DEVICE INCREMENT TO R3
1093  003444  060401  :173444  060401          ADD      R4,R1         ;UPDATE TO NEXT DEVICE
1094  003446  005201  :173446  005201          INC      R1            ;INCREMENT MODULO
1095  003450  040401  :173450  040401          BIC      R4,R1         ;CLEAR EXCESS
1096  003452  005703  :173452  005703          TST      R3            ;TEST FOR DONE
1097  003454  001371  :173454  001371          BNE     DEVELOP        ;NOT YET
1098  003456  005700  :173456  005700          TST      R0            ;TEST SWITCH REG USE

```

1099	003460	001002	:173460	001002	BNE	SNDREQ	;NO SWITCH REG
1100	003462	063701	:173462	063701	ADD	2*SSMR,R1	;ADD SMR VALUE
1101	003464	177570	:173464	177570			
1102							
1103							
1104							
1105			:173466				
1106	003466	012711	:173466	012711	SNDREQ: MOV	#6,(R1)	;DATA TERM RDY AND REQ TO SEND
1107	003470	000006	:173470	000006			
1108	003472	012761	:173472	012761	MOV	#36000+\$SYN,2(R1)	;SET SYNC REGISTER
1109	003474	036226	:173474	036226			
1110	003476	000002	:173476	000002			
1111	003500	032711	:173500	032711	L3: BIT	#20000,(R1)	;TEST CLEAR TO SEND
1112	003502	020000	:173502	020000			
1113	003504	001775	:173504	001775	BEQ	L3	;NOT YET
1114	003506	022121	:173506	022121	CMP	(R1)+,(R1)+	;MOVE PTR TO XMIT TSR
1115	003510	052711	:173510	052711	BIS	#20,(R1)	;TURN SEND ON
1116	003512	000020	:173512	000020			
1117							
1118							
1119							
1120	003514	010700	:173514	010700	MOV	PC,RO	;CURRENT PC
1121	003516	062700	:173516	062700	ADD	#RQMSG-,RO	;REQUEST MSG ADDR
1122	003520	000230	:173520	000230			
1123	003522	012704	:173522	012704	MOV	#RQMSGE-RQMSG,R4	;COUNT
1124	003524	000026	:173524	000026			
1125	003526	112061	:173526	112061	L4: MOVB	(RO)+,2(R1)	;CHAR TO XMIT REGISTER
1126	003530	000002	:173530	000002			
1127	003532	105711	:173532	105711	L5: TSTB	(R1)	;DONE YET ?
1128	003534	100376	:173534	100376	BPL	L5	;NO
1129	003536	005304	:173536	005304	DEC	R4	;DECREMENT COUNT
1130	003540	001372	:173540	001372	BNE	L4	;ONCE MORE
1131	003542	042711	:173542	042711	BIC	#20,(R1)	;DROP SEND
1132	003544	000020	:173544	000020			
1133	003546	024141	:173546	024141	CMP	-(R1),-(R1)	;RESET PTR TO RCV CSR
1134							
1135							
1136							
1137			:173550				
1138	003550	042711	:173550	042711	GETPGM: BIC	#20,(R1)	;CLEAR SEARCH SYNC
1139	003552	000020	:173552	000020			
1140	003554	012711	:173554	012711	MOV	#422,(R1)	;SET FOR CLEAR AND STRIP SYNC
1141	003556	000422	:173556	000422			
1142	003560	005003	:173560	005003	CLR	R3	;CLEAR CRC VALUE
1143							
1144							
1145							
1146	003562	012700	:173562	012700	MOV	#1,RO	;LOAD HDR AT LOC. 1
1147	003564	000001	:173564	000001			
1148	003566	012704	:173566	012704	MOV	#8.,R4	;BLOCK COUNT
1149	003570	000010	:173570	000010			
1150	003572	004767	:173572	004767	JSR	PC,GET	;GET HEADER
1151	003574	000060	:173574	000060			
1152	003576	005703	:173576	005703	TST	R3	;CHECK HEADER CRC

```

1153 003600 001363 ;173600 001363      BNE    GETPGM      ;NO GOOD
1154 003602 123727 ;173602 123727      CMPB   @#6, #SSTADR ;CHECK FOR MY ADDR
1155 003604 000006 ;173604 000006
1156 003606 000001 ;173606 000001
1157 003610 001357 ;173610 001357      BNE    GETPGM      ;NOT MINE
1158 003612 123727 ;173612 123727      CMPB   @#1, #SDLE  ;IS THIS A DLE MSG
1159 003614 000001 ;173614 000001
1160 003616 000220 ;173616 000220
1161 003620 001322 ;173620 001322      BNE    SNDREQ      ;NO, ASK FOR ONE
1162
1163 ;
1164 ;
1165 003622 013704 ;173622 013704      MOV    @#2, R4      ;DATA FIELD LENGTH
1166 003624 000002 ;173624 000002
1167 003626 042704 ;173626 042704      BIC    #140000, R4  ;MASK OFF S,F BITS
1168 003630 140000 ;173630 140000
1169 003632 122424 ;173632 122424      CMPB   (R4)+, (R4)+ ;ADD 2 FOR CRC
1170 003634 005000 ;173634 005000      CLR    R0           ;LOAD INTO LOCATION 0
1171 003636 004767 ;173636 004767      JSR    PC, GET1     ;GET DATA BLOCK
1172 003640 000014 ;173640 000014
1173 003642 005703 ;173642 005703      TST    R3           ;CHECK DATA FIELD CRC
1174 003644 001310 ;173644 001310      BNE    SNDREQ      ;NO GOOD
1175 003646 105713 ;173646 105713      TSTB   (R3)        ;CHECK CODE IN LOC 0
1176 003650 001306 ;173650 001306      BNE    SNDREQ      ;NOT PROGRAM LOAD
1177 003652 000137 ;173652 000137      JMP    @#6         ;TRANSFER TO SECONDARY PGM
1178 003654 000006 ;173654 000006
1179
1180 ;
1181 ;
1182 ;
1183 ;
1184 003656 105711 ;173656 105711      TSTB   (R1)        ;IS DEVICE DONE YET
1185 003660 100376 ;173660 100376      BPL    GET         ;NOT YET
1186 003662 042711 ;173662 042711      BIC    #SSTRIP, (R1) ;NO STRIP SYNC
1187 003664 000400 ;173664 000400
1188 003666 116110 ;173666 116110      MOVB   2(R1), (R0) ;STORE IT
1189 003670 000002 ;173670 000002
1190
1191 ;
1192 ;
1193 ;
1194 ;
1195 003672 012705 ;173672 012705      MOV    #8., R5      ;BYTE LENGTH
1196 003674 000010 ;173674 000010
1197 003676 112002 ;173676 112002
1198 003700 000241 ;173700 000241      CRCLOP: CLC         ;CHARACTER TO ADD TO CRC
1199 003702 006003 ;173702 006003      ROR    R3          ;CLEAR CARRY
1200 003704 103003 ;173704 103003      BCC    L10        ;SHIFT OLD PARTIAL
1201 003706 006002 ;173706 006002      ROR    R2          ;IF CLEAR CHECK CHAR
1202 003710 103003 ;173710 103003      BCC    L11        ;SHIFT CHARACTER
1203 003712 000410 ;173712 000410      BR     L12        ;XOR POLY
1204 003714 006002 ;173714 006002      L10:  ROR    R2    ;NEXT BIT
1205 003716 103006 ;173716 103006      BCC    L12        ;SHIFT CHARACTER
1206 003720 012746 ;173720 012746      L11:  MOV    #POLY, -(SP) ;NEXT BIT
;POLY TO STACK

```



```

1207 003722 120001 ;173722 120001
1208 003724 040316 ;173724 040316      BIC      R3,(SP)      ;NOT PARTIAL AND POLY
1209 003726 042703 ;173726 042703      BIC      #POLY,R3    ;NOT POLY AND PARTIAL
1210 003730 120001 ;173730 120001
1211 003732 052603 ;173732 052603      BIS      (SP)+,R3    ;POLY XOR PARTIAL
1212 003734 005305 ;173734 005305      L12:    DEC      R5      ;DECREMENT BIT COUNT
1213 003736 001360 ;173736 001360      BNE     CRCLOP      ;ONCE MORE
1214 003740 005304 ;173740 005304      DEC     R4          ;DECREMENT COUNT
1215 003742 001345 ;173742 001345      BNE     GET         ;ONCE MORE
1216 003744 000207 ;173744 000207      RTS     PC          ;RETURN
1217
1218 ;
1219 ;      SECONDARY PROGRAM REQUEST MSG
1220 003746 113226 ;173746 113226      RQMSG:  .BYTE    $SYN,$SYN,$SYN,$SYN
1221 003750 113226 ;173750 113226
1222 003752 002220 ;173752 002220      .BYTE    $DLE,4,0,0,0,1
1223 003754 000000 ;173754 000000
1224 003756 000400 ;173756 000400
1225 003760 050055 ;173760 050055      .BYTE    55,120
1226 003762 001010 ;173762 001010      .BYTE    10          ;REQ SEC PGM CODE
1227 ;      .BYTE    2          ;DEVICE CODE
1228 003764 000001 ;173764 000001      .BYTE    $STADR      ;STATION ADDR
1229 ;      .BYTE    0          ;FILL
1230 003766 030242 ;173766 030242      .BYTE    242,60     ;FOR STADR=1
1231 ;      NOTE: NODEV AND DEVTAB MUST BE IN THIS ORDER
1232 ;      DO NOT SEPARATE THEM
1233 ;
1234 ;      .EVEN
1235 003770 122243 ;173770 122243      NODEV:  C1PB      (R2)+,-(R3) ;INC PTR-DEC CNT
1236 003772 000002 ;173772 000002      RTI          ;RETURN FROM TRAP
1237 ;
1238 ;      RQMSG:
1239 003774 007407 ;173774 007407      DEVTAB: .BYTE    7          ;END OF MSG-USE JUNK AS PADS
1240 ;      .BYTE    17         ;DJ-11
1241 ;      .BYTE    17         ;DH-11
1241 003776 END.YC: ;
1242 003776 003407 ;173776 003407      .BYTE    7          ;DQ-11
1243 ;      .BYTE    7          ;DU-11
1244 ;      .BYTE    174000
1245 ;      .END    START1

```

1246 004000 MAP.YD:

1247 : THE FOLLOWING IS A REPRODUCTION  
 1248 : OF THE ROM PROGRAM FOR BM873YD.  
 1249 : IT IS HERE FOR COMPARISON TO THE  
 1250 : ACTUAL ROM AND FOR REFERENCE

1251 : BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 1  
 1252 : BM873-YD.P11

1253  
 1254 : THIS CODE IS TO BE BLASTED INTO PROMS ON THE BM873-YD BOARD.  
 1255 : WRITTEN BY DAVID M. ROSENBERG OCTOBER 1974  
 1256 : REGISTER DEFINITIONS

1257	:	000000	R0=X0	:	GENERAL PURPOSE REGISTER 0
1258	:	000001	R1=X1	:	GENERAL PURPOSE REGISTER 1
1259	:	000002	R2=X2	:	GENERAL PURPOSE REGISTER 2
1260	:	000003	R3=X3	:	GENERAL PURPOSE REGISTER 3
1261	:	000004	R4=X4	:	GENERAL PURPOSE REGISTER 4
1262	:	000005	R5=X5	:	GENERAL PURPOSE REGISTER 5
1263	:	000006	SP=X6	:	STACK POINTER (REGISTER R6)
1264	:	000007	PC=X7	:	PROGRAM COUNTER (REGISTER R7)

1265  
 1266  
 1267  
 1268 ;SYMBOL DEFINITIONS

1269	:	177776	PS=177776	:	PROCESSOR STATUS REGISTER
1270	:	177570	SMR=177570	:	FRONT PANEL SWITCH REGISTER
1271	:	000000	PR0=0*40	:	PRIORITY LEVEL 0
1272	:	000040	PR1=1*40	:	PRIORITY LEVEL 1
1273	:	000100	PR2=2*40	:	PRIORITY LEVEL 2
1274	:	000140	PR3=3*40	:	PRIORITY LEVEL 3
1275	:	000200	PR4=4*40	:	PRIORITY LEVEL 4
1276	:	000240	PR5=5*40	:	PRIORITY LEVEL 5
1277	:	000300	PR6=6*40	:	PRIORITY LEVEL 6
1278	:	000340	PR7=7*40	:	PRIORITY LEVEL 7
1279	:	000001	BIT0=000001		
1280	:	000002	BIT1=000002		
1281	:	000004	BIT2=000004		
1282	:	000010	BIT3=000010		
1283	:	000020	BIT4=000020		
1284	:	000040	BIT5=000040		
1285	:	000100	BIT6=000100		
1286	:	000200	BIT7=000200		
1287	:	000400	BIT8=000400		
1288	:	001000	BIT9=001000		
1289	:	002000	BIT10=002000		
1290	:	004000	BIT11=004000		
1291	:	010000	BIT12=010000		
1292	:	020000	BIT13=020000		
1293	:	040000	BIT14=040000		
1294	:	100000	BIT15=100000		
1295	:				
1296	:				

```

1297      ;BM873-YD      - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17)  MACY11 27(657) 18-DEC-74 11:59 PAGE 3
1298      ;BM873-YD.P11  BUTTON #1 - BOOTSTRAP USING THE PDP-11 SWITCH REGISTER
1299
1300
1301      ;           173000  ROMORG  =      173000      ;SET ROM ORIGIN TO 773000
1302      ;           173000      . =ROMORG      ;BM873-YD OCCUPIES 773000-773777
1303
1304  004000  033727 ;173000  033727  BUTON1: BIT      @*SWR,#BIT0      ;IS RIGHTMOST BIT ON?
1305  004002  177570 ;173002  177570
1306  004004  000001 ;173004  000001
1307  004006  001010 ;173006  001010      BNE      LOWBIT      ;IF THE BIT IS ON, BRANCH
1308  004010  013707 ;173010  013707      MOV      @*SWR,PC      ;JUMP TO THE ADDRESS IN THE SWITCH REGISTER
1309  004012  177570 ;173012  177570
1310
1311
1312  004014  111704 ;173014  111704  BUTON3: MOVB      (PC),R4      ;R4 = 1 INDICATES THAT BUTTON #3 WAS PRESSED
1313  004016  005001 ;173016  005001      CLR      R1      ;SET UNIT NUMBER TO ZERO
1314  004020  005005 ;173020  005005      CLR      R5      ;CLEAR "LOGICAL SWITCH REGISTER"
1315  004022  000424 ;173022  000424      BR       TCBOOT      ;DO A DEFAULT BOOT STRAP FROM DECTAPE
1316
1317  004024  173000 ;173024  173000      .WORD   ROMORG,PR7
1318  004026  000340 ;173026  000340
1319
1320  004030  013701 ;173030  013701  LOWBIT: MOV      @*SWR,R1      ;R1 IS A COPY OF THE SWITCH REGISTER
1321  004032  177570 ;173032  177570
1322  004034  106301 ;173034  106301      ASLB    R1      ;LEFT-ALIGN SPEED FIELD IN RIGHT BYTE
1323  004036  122701 ;173036  122701      CMPB    #16*20,R1      ;IS THE SPEED 16 OR 17?
1324  004040  000340 ;173040  000340
1325  004042  101404 ;173042  101404      BLOS    UNITNO      ;IF SPEED IS 16 OR 17, BRANCH
1326  004044  122701 ;173044  122701      CMPB    #3*20,R1      ;IS THE SPEED 0, 1, OR 2?
1327  004046  000060 ;173046  000060
1328  004050  101001 ;173050  101001      BHI     UNITNO      ;IF THE SPEED IS 0, 1, OR 2, BRANCH
1329  004052  005001 ;173052  005001      CLR     R1      ;SPEED WAS 3-15; SET UNIT NUMBER = 0
1330  004054  000301 ;173054  000301  UNITNO: SWAB    R1      ;MOVE UNIT NUMBER TO BITS 0-2
1331
1332      ; IT IS POSSIBLE TO MANUALLY SET THE DESIRED BOOTSTRAP UNIT NUMBER
1333      ; INTO THE RIGHTMOST THREE BITS OF R1, SET THE PDP-11 FRONT PANEL
1334      ; SWITCH REGISTER, AND THEN JUMP INTO THE ROM CODE AT THIS POINT.
1335
1336  004056  042701 ;173056  042701      BIC     #1C7,R1      ;ISOLATE UNIT NUMBER IN R1
1337  004060  177770 ;173060  177770
1338  004062  013705 ;173062  013705      MOV     @*SWR,R5      ;R5 IS NOW THE "LOGICAL SWITCH REGISTER"
1339  004064  177570 ;173064  177570
1340  004066  005004 ;173066  005004      CLR     R4      ;R4 = 0 INDICATES THAT BUTTON #1 WAS PRESSED
1341  004070  105705 ;173070  105705      TSTB   R5      ;SHOULD WE BOOT FROM DECTAPE OR RH11/RP04?
1342  004072  100507 ;173072  100507      BMI     RPBOOT      ;IF BIT 7 WAS ONE, BRANCH OFF TO THE RH11/RP04
1343
1344

```

```

1344 ;BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 4
1345 ;BM873-YD.P11 DECTAPE BOOTSTRAP AND DUMP ROUTINES
1346
1347
1348 ; 177344 TCWC = 177344 ;TC11 DECTAPE WORD COUNT REGISTER
1349 ; 000001 TCGO = 1 ;TC11 "GO" BIT
1350 ; 000002 TCRNUM = 1*2 ;TC11 "READ BLOCK NUMBER" FUNCTION
1351 ; 000004 TCREAD = 2*2 ;TC11 "READ DATA" FUNCTION
1352 ; 000014 TCWRIT = 6*2 ;TC11 "WRITE DATA" FUNCTION
1353 ; 004000 TCREV = 4000 ;MOVE DECTAPE IN REVERSE DIRECTION
1354
1355 ; BOOTSTRAP (FROM DECTAPE) PARAMETERS
1356 ; 000400 TCBWDC = 1D256 ;WORD COUNT FOR THE SECONDARY BOOTSTRAP
1357 ; 000000 TCBEND = 0 ;WHICH END OF THE DECTAPE (0 = FRONT; 1 = BACK)
1358
1359 ; DUMP (TO DECTAPE) PARAMETERS
1360 ; 070000 TCDWDC = 1D28672 ;WORD COUNT FOR THE CORE DUMP TO DECTAPE
1361 ; 000001 TCDEND = 1 ;WHICH END OF THE DECTAPE (0 = FRONT; 1 = BACK)
1362
1363 ; GENERAL (BOOTSTRAP AND DUMP) DECTAPE PARAMETER
1364 ; 000024 TCRTRY = 1D20 ;NUMBER OF RETRIES IN CASE OF ERROR
1365
1366 004074 012700 ;173074 012700 TCBOOT: MOV #<TCBEND*TCREV>!TCREAD!TCGO,R0 ;SET UP DATA-TRANSFER COMMAND
1367 004076 000005 ;173076 000005
1368 004100 012702 ;173100 012702 MOV #-TCBWD, R2 ;SET WORD COUNT TO 256 (512 BYTES)
1369 004102 177400 ;173102 177400
1370 004104 012703 ;173104 012703 MOV #<<1-TCBEND>*TCREV>!TCRNUM!TCGO,R3 ;SET UP POSITION COMMAND
1371 004106 004003 ;173106 004003
1372 004110 000301 ;173110 000301 SWAB R1 ;BRING UNIT NUMBER INTO THE LEFT BYTE
1373 004112 050103 ;173112 050103 BIS R1,R3 ;PUT UNIT NUMBER INTO POSITIONING COMMAND
1374 004114 050100 ;173114 050100 BIS R1,R0 ;PUT UNIT NUMBER INTO DATA-TRANSFER COMMAND
1375 004116 012701 ;173116 012701 TCSTRT: MOV #TCWC,R1 ;R1 NOW POINTS TO TC11 WORD COUNT REGISTER
1376 004120 177344 ;173120 177344
1377 004122 012706 ;173122 012706 TCLOOP: MOV #TCRTRY,SP ;INITIALIZE RETRY COUNT IN SP
1378 004124 000024 ;173124 000024
1379 004126 005705 ;173126 005705 TCBGIN: TST R5 ;TEST "INDEFINITE RETRY" BIT
1380 004130 100404 ;173130 100404 BMI TCRSET ;BRANCH IF "INDEFINITE RETRY" IS ENABLED
1381 004132 005306 ;173132 005306 DEC SP ;DECREMENT RETRY COUNT
1382 004134 100002 ;173134 100002 BPL TCRSET ;BRANCH IF RETRY COUNT NOT EXHAUSTED
1383 004136 000000 ;173136 000000 TCHALT: HALT ;RETRY COUNT IS EXHAUSTED FOR DECTAPE OPERATION
1384 004140 000770 ;173140 000770 BR TCLOOP ;HE PRESSED "CONTINUE", SO TRY AGAIN
1385 004142 000005 ;173142 000005 TCRSET: RESET ;STOP ANYTHING IN PROGRESS, FOR NEXT TRY
1386 004144 010341 ;173144 010341 MOV R3, -(R1) ;INITIATE DECTAPE POSITIONING OPERATION
1387 004146 005711 ;173146 005711 TCWAIT: TST (R1) ;TEST FOR AN "ERROR"
1388 004150 100376 ;173150 100376 BPL TCWAIT ;LOOP UNTIL AN "ERROR" IS DETECTED
1389 004152 005721 ;173152 005721 TST (R1)+ ;MAKE R1 POINT TO THE WORD COUNT REGISTER
1390 004154 005761 ;173154 005761 TST -4(R1) ;IS THE ERROR "ENDZONE"?
1391 004156 177774 ;173156 177774
1392 004160 100362 ;173160 100362 BPL TCBGIN ;IF NOT, BRANCH BACK TO TRY AGAIN
1393 004162 010211 ;173162 010211 MOV R2, (R1) ;SET UP WORD COUNT FOR DATA-TRANSFER
1394 004164 010041 ;173164 010041 MOV R0, -(R1) ;INITIATE THE DATA-TRANSFER OPERATION
1395 004166 105711 ;173166 105711 TCDONE: TSTB (R1) ;TEST FOR "DONE"
1396 004170 100376 ;173170 100376 BPL TCDONE ;LOOP UNTIL THE "DONE" BIT SETS
1397 004172 005721 ;173172 005721 TST (R1)+ ;WAS AN "ERROR" DETECTED?

```

1398	004174	100754 ;173174	100754	BMI	TCBGIN	; IF SO, BRANCH BACK AND TRY AGAIN
1399	004176	005741 ;173176	005741	TST	-(R1)	; MAKE R1 POINT TO THE COMMAND REGISTER
1400	004200	105011 ;173200	105011	CLRB	(R1)	; STOP ALL DECTAPE MOTION
1401	004202	122700 ;173202	122700	CMPB	#TCREAD!TCGO,R0	; WAS THIS A "NORMAL READ" OPERATION?
1402	004204	000005 ;173204	000005			
1403	004206	001001 ;173206	001001	BNE	TCSTOP	; IF NOT, GO STOP
1404	004210	000137 ;173210	000137	GOTO0: JMP	2(PC)+	; JUMP TO PDP-11 LOCATION ZERO
1405	004212	000000 ;173212	000000	TCSTOP: HALT		; SUCCESSFUL COMPLETION OF A "NON-READ" OPERATION
1406	004214	000776 ;173214	000776	BR	TCSTOP	; SO THAT PRESSING "CONTINUE" WON'T GO ANYWHERE

1407 :BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 5  
1408 ;BM873-YD.P11 DECTAPE BOOTSTRAP AND DUMP ROUTINES

```

1409
1410 004216 010037 :173216 010037 TCDUMP: MOV R0,#ROTOR7 ;SAVE R0 IN PDP-11 MEMORY LOCATION 40
1411 004220 000040 :173220 000040
1412 004222 000402 :173222 000402 BR TCCONT ;BRANCH AROUND REQUIRED INTERRUPT VECTOR
1413
1414 004224 173000 :173224 173000 .WORD ROMORG,PR7
1415 004226 000340 :173226 000340
1416
1417 004230 010700 :173230 010700 TCCONT: MOV PC,R0 ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
1418 004232 000410 :173232 000410 BR REGSAV ;GO TO THE "REGISTER SAVING" SUBROUTINE
1419 004234 012700 :173234 012700 MOV #<TCDEND*TCREV>!TCRIT!TCGO,R0 ;SET UP (WRITE) TRANSFER COMMAND
1420 004236 004015 :173236 004015
1421 004240 012702 :173240 012702 MOV #-TCDMDC,R2 ;SET WORD-COUNT TO 28K WORDS
1422 004242 110000 :173242 110000
1423 004244 012703 :173244 012703 MOV #<<1-TCDEND>*TCREV>!TCRNJM!TCGO,R3 ;SET UP POSITION COMMAND
1424 004246 000003 :173246 000003
1425 004250 005005 :173250 005005 CLR R5 ;CLEAR "INDEFINITE RETRY" BIT
1426 004252 000721 :173252 000721 BR TCSTRT ;BRANCH INTO DECTAPE ROUTINE
1427
1428
1429
1430

```

: THE FOLLOWING SUBROUTINE IS USED TO SAVE THE PDP-11 GENERAL REGISTERS  
: IN PDP-11 MEMORY LOCATIONS 40-57.

: THE CALLING SEQUENCE IS AS FOLLOWS:

```

MOV R0,#ROTOR7
MOV PC,R0
BR REGSAV
RETURN HERE

```

```

1431
1432
1433
1434
1435
1436
1437
1438
1439 004254 010137 :173254 010137 REJSAV: MOV R1,#ROTOR7+2 ;SAVE R1 IN MEMORY LOCATION 42
1440 004256 000042 :173256 000042
1441 004260 012701 :173260 012701 MOV #ROTOR7+4,R1 ;R1 NOW POINTS TO MEMORY LOCATION 44
1442 004262 000044 :173262 000044
1443 004264 010221 :173264 010221 MOV R2,(R1)+ ;SAVE R2 IN MEMORY LOCATION 44
1444 004266 010321 :173266 010321 MOV R3,(R1)+ ;SAVE R3 IN MEMORY LOCATION 46
1445 004270 010421 :173270 010421 MOV R4,(R1)+ ;SAVE R4 IN MEMORY LOCATION 50
1446 004272 010521 :173272 010521 MOV R5,(R1)+ ;SAVE R5 IN MEMORY LOCATION 52
1447 004274 010621 :173274 010621 MOV SP,(R1)+ ;SAVE SP IN MEMORY LOCATION 54
1448 004276 010021 :173276 010021 MOV R0,(R1)+ ;SAVE PC IN MEMORY LOCATION 56
1449 004300 000160 :173300 000160 JMP 2(R0) ;RETURN TO THE CALLING ROUTINE
1450 004302 000002 :173302 000002
1451

```



1452	;BM873-YD		- KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17)		MACY11 27(657)	18-DEC-74	11:59	PAGE 6
1453	;BM873-YD.P11		RH11/RP04 BOOTSTRAP AND DUMP ROUTINES					
1454								
1455								
1456	:	176700	RPCS1	=	176700	;ADDRESS OF RH11/RP04 CONTROL & STATUS REGISTER 1		
1457	:	000002	RPMC	=	2	;OFFSET TO RH11/RP04 WORD COUNT REGISTER		
1458	:	000006	RPDA	=	6	;OFFSET TO RH11/RP04 TRACK & SECTOR ADDRESS REGISTER		
1459	:	000010	RPCS2	=	10	;OFFSET TO RH11/RP04 CONTROL & STATUS REGISTER 2		
1460	:	000012	RPCS	=	12	;OFFSET TO RH11/RP04 DRIVE STATUS REGISTER		
1461	:	000032	RPOF	=	32	;OFFSET TO RH11/RP04 OFFSET REGISTER (CONTAINING FMT22)		
1462	:	000034	RPDC	=	34	;OFFSET TO RH11/RP04 DESIRED CYLINDER REGISTER		
1463								
1464	:	040000	RP*RE	=	BIT14	;"TRANSFER ERROR" BIT IN RPCS1		
1465	:	020000	RPMCPE	=	BIT13	;"MASSBUS CONTROL BUS PARITY ERROR" BIT IN RPCS1		
1466	:	004000	RPDVA	=	BIT11	;"DRIVE AVAILABLE" BIT IN RPCS1		
1467	:	100000	RPATA	=	BIT15	;"ATTENTION ACTIVE" BIT IN RPDS		
1468	:	040000	RPEER	=	BIT14	;"COMPOSITE ERROR" BIT IN RPDS		
1469	:	010000	RPFMT	=	BIT12	;"FMT22" (16-BIT WORDS) BIT IN RPOF		
1470								
1471	:	000021	RPPRST	=	21	;READ-IN PRESET		
1472	:	000061	RPWRIT	=	61	;WRITE DATA		
1473	:	000071	RPREAD	=	71	;READ DATA		
1474								
1475	:	000000	RPBFMT	=	0	;BOOTSTRAP FORMAT (0 = 18-BIT WORDS; 2 = 16-BIT WORDS)		
1476	:	000400	RPBMDC	=	↑D256	;WORD COUNT FOR THE SECONDARY BOOTSTRAP FROM THE RP04		
1477	:	000626	RPBCYL	=	↑D406	;BOOTSTRAP CYLINDER NUMBER		
1478	:	000000	RPBTRK	=	0	;BOOTSTRAP TRACK NUMBER		
1479	:	000000	RPBSCT	=	0	;BOOTSTRAP SECTOR NUMBER		
1480								
1481	:	000000	RPDFMT	=	0	;DUMP FORMAT (0 = 18-BIT WORDS; 2 = 16-BIT WORDS)		
1482	:	070000	RPDMDC	=	↑D28672	;WORD COUNT FOR THE CORE DUMP ↑D256		
1483	:	000631	RPDCYL	=	↑D409	;DUMP CYLINDER NUMBER		
1484	:			=		; THE FOLLOWING TWO ASSIGNMENTS PUT THE DUMP AT THE VERY END OF THE CYLINDER		
1485	:	000015	RPDTRK	=	↑D18-⟨⟨RPDMDC-1⟩⟩/⟨⟨↑D20+RPDFMT⟩⟩*↑D256⟩⟩	;DUMP TRACK NUMBER		
1486	:	000010	RPDSCT	=	↑D19+RPDFMT-⟨⟨RPDMDC-1⟩⟩/↑D256)-⟨⟨↑D18-RPDTRK⟩⟩*⟨⟨↑D20+RPDFMT⟩⟩⟩	;DUMP SECTOR NUMBER		
1487								
1488								
1489								
1490	004304	111704	;173304	111704	BUTON2: MOV	(PC),R4	;R4 = 5 INDICATES THAT BUTTON #2 WAS PRESSED	
1491	004306	005005	;173306	005005	CLR	RS	;CLEAR "LOGICAL SWITCH REGISTER"	
1492	004310	005001	;173310	005001	CLR	R1	;SET UNIT NUMBER TO ZERO	
1493								
1494	004312	012700	;173312	012700	RPBOOT: MOV	↑⟨RPREAD*400⟩!⟨RPBSCT*10⟩,R0		
1495	004314	034400	;173314	034400				
1496	004316	012702	;173316	012702	MOV	↑-RPBMDC,R2		
1497	004320	177400	;173320	177400				
1498	004322	012703	;173322	012703	MOV	↑⟨RPBFMT*40000⟩!⟨RPBTRK*2000⟩!RPBCYL,R3		
1499	004324	000626	;173324	000626				
1500	004326	050100	;173326	050100	RPSTR: BIS	R1,R0	;PUT THE UNIT NUMBER INTO R0	
1501	004330	012701	;173330	012701	MOV	↑RPCS1,R1	;SET R1 TO THE LOWEST ADDRESS USED BY THE RH11	
1502	004332	176700	;173332	176700				

```

1503      ;BM873-YD      - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 7
1504      ;BM873-YD.P11 RH11/RP04 BOOTSTRAP AND DUMP ROUTINES
1505
1506 004334 000005 ;173334 000005 RPL00P: RESET ;RESET IN CASE OF RETRY
1507 004336 010006 ;173336 010006 MOV RC,SP ;GET THE UNIT NUMBER INTO SP
1508 004340 042706 ;173340 042706 BIC #1C7,SP ;ISOLATE THE UNIT NUMBER
1509 004342 177770 ;173342 177770
1510 004344 010661 ;173344 010661 MOV SP,RPCS2(R1) ;TELL THE RH11 THE UNIT NUMBER
1511 004346 000010 ;173346 000010
1512 004350 032711 ;173350 032711 BIT #RPDVA,(R1) ;TRY TO SEIZE THIS RPO4 UNIT
1513 004352 004000 ;173352 004000
1514 004354 001767 ;173354 001767 BEQ RPL00P ;BRANCH IF WE HAVEN'T SEIZED IT
1515 004356 012721 ;173356 012721 MOV #RPPRST,(R1)+ ;DO A "READ-IN PRESET" FUNCTION
1516 004360 000021 ;173360 000021
1517 004362 010306 ;173362 010306 MOV R3,SP ;GET THE CYLINDER NUMBER INTO SP
1518 004364 042706 ;173364 042706 BIC #1C1777,SP ;ISOLATE THE CYLINDER NUMBER
1519 004366 176000 ;173366 176000
1520 004370 010661 ;173370 010661 MOV SP,RPDC-2(R1) ;TELL THE RPO4 THE CYLINDER NUMBER
1521 004372 000032 ;173372 000032
1522 004374 010306 ;173374 010306 MOV R3,SP ;GET THE FORMAT BIT AND TRACK NUMBER INTO SP
1523 004376 100003 ;173376 100003 BPL RPCONT ;BRANCH IF 20 SECTOR (18-BIT WORDS) FORMAT
1524 004400 012761 ;173400 012761 MOV #RPFMT,RPOF-2(R1) ;ESTABLISH 22 SECTOR (16-BIT WORDS) FORMAT
1525 004402 010000 ;173402 010000
1526 004404 000030 ;173404 000030
1527 004406 006206 ;173406 006206 RPCONT: ASR SP ;RIGHT ALIGN THE TRACK
1528 004410 006206 ;173410 006206 ASR SP ;NUMBER IN THE LEFT BYTE
1529 004412 105006 ;173412 105006 CLRB SP ;CLEAR THE RIGHT BYTE
1530 004414 150006 ;173414 150006 BISB R0,SP ;PUT THE SECTOR NUMBER INTO THE RIGHT BYTE
1531 004416 106006 ;173416 106006 RORB SP ;RIGHT ALIGN THE
1532 004420 106206 ;173420 106206 ASRB SP ;SECTOR NUMBER IN
1533 004422 106206 ;173422 106206 ASRB SP ;THE RIGHT BYTE
1534 004424 010661 ;173424 010661 MOV SP,RPDA-2(R1) ;TELL THE RH11 THE TRACK AND SECTOR NUMBERS
1535 004426 000004 ;173426 000004
1536 004430 010211 ;173430 010211 MOV R2,(R1) ;TELL THE RH11 THE WORD COUNT
1537 004432 010006 ;173432 010006 MOV R0,SP ;GET THE FUNCTION CODE INTO SP
1538 004434 105006 ;173434 105006 CLRB SP ;CLEAR THE RIGHT BYTE
1539 004436 000306 ;173436 000306 SWAB SP ;RIGHT ALIGN THE FUNCTION CODE
1540 004440 010641 ;173440 010641 MOV SP,-(R1) ;TELL THE RPO4 THE FUNCTION CODE
1541 004442 105711 ;173442 105711 RPDONE: TSTB (R1) ;TEST FOR RH11 "READY"
1542 004444 100376 ;173444 100376 BPL RPDONE ;LOOP, WAITING FOR RH11 "READY"
1543 004446 032711 ;173446 032711 BIT #PPTRE!RPMCPE,(R1) ;TEST FOR RH11 ERROR BITS
1544 004450 060000 ;173450 060000
1545 004452 001330 ;173452 001330 BNE RPL00P ;IF ERROR, BRANCH BACK FOR RETRY
1546 004454 032761 ;173454 032761 BIT #RPATA!RPERR,RPDS(R1) ;TEST FOR RPO4 ERROR BITS
1547 004456 140000 ;173456 140000
1548 004460 000012 ;173460 000012
1549 004462 001324 ;173462 001324 BNE RPL00P ;IF ERROR, BRANCH BACK FOR RETRY
1550 004464 022706 ;173464 022706 CMP #RPREAD,SP ;WAS THE FUNCTION A "NORMAL READ"?
1551 004466 000071 ;173466 000071
1552 004470 001250 ;173470 001250 BNE TCSTOP ;IF NOT, BRANCH TO A HALT INSTRUCTION
1553 004472 022737 ;173472 022737 CMP #000240,#0 ;WAS "000240" READ INTO LOCATION ZERO?
1554 004474 000240 ;173474 000240
1555 004476 000000 ;173476 000000
1556 004500 001643 ;173500 001643 BEQ GOT00 ;IF SO, BRANCH TO LOCATION ZERO

```

```

1557 004502 000000 :173502 000000      HALT
1558 004504 000641 ;173504 000641      BR      GOT00      ;"000240" WAS NOT READ INTO LOCATION ZERO
1559
1560
1561 004506 010037 :173506 010037  RPDUMP: MOV      RO,2*ROTOR7      ;SAVE RO IN PDP-11 MEMORY LOCATION "ROTOR7"
1562 004510 000040 ;173510 000040
1563 004512 010700 ;173512 010700      MOV      PC,RO      ;USE RO FOR A SUBROUTINE RETURN ADDRESS
1564 004514 000657 ;173514 000657      BR      REGSAV      ;GO TO THE "REGISTER SAVING" SUBROUTINE
1565 004516 012700 ;173516 012700      MOV      *(<RPMRIT*400>!<RPDSC*10>),RO
1566 004520 030500 ;173520 030500
1567 004522 012702 ;173522 012702      MOV      *-RPDMDC,R2
1568 004524 110000 ;173524 110000
1569 004526 012703 ;173526 012703      MOV      *(<RPFMT*40000>!<RPDTRK*2000>!RPDCYL,R3
1570 004530 032631 ;173530 032631
1571 004532 000676 ;173532 000676      BR      RPSTRT
1572

```

```

1573 :BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 8
1574 :BM873-YD.P11 DTE20 DEVICE REGISTER AND BIT DEFINITIONS
1575
1576
1577 : 174400 DTEBAS=174400 ;BASE OF (FIRST) DTE20 DEVICE REGISTER BLOCK
1578 : 000040 DTESIZ=000040 ;SPACING BETWEEN CONSECUTIVE DTE20'S
1579 : 000004 DTEMAX=4 ;MAXIMUM NUMBER OF DTE20'S ON ONE PDP-11
1580
1581
1582 : ;OFFSETS FROM THE BASE OF THE DTE20 DEVICE REGISTER BLOCK
1583 : ;TO SPECIFIC 10/11 INTERFACE RAM LOCATIONS AND REGISTERS.
1584
1585 ; THE FIRST 12 REGISTERS ARE NOT INITIALIZED BY "INIT" (BECAUSE THEY ARE IN RAMS
1586 : 000000 DLYCNT=00 ;DELAY COUNT (ADDRESS XXXX00)
1587 : 000002 DEXWD3=02 ;DEPOSIT OR EXAMINE WORD 3 (ADDRESS XXXX02)
1588 : 000004 DEXWD2=04 ;DEPOSIT OR EXAMINE WORD 2 (ADDRESS XXXX04)
1589 : 000006 DEXWD1=06 ;DEPOSIT OR EXAMINE WORD 1 (ADDRESS XXXX06)
1590 : 000010 TENAD1=10 ;10 ADDRESS WORD 1 FOR DEX (ADDRESS XXXX10)
1591 : 000012 TENAD2=12 ;10 ADDRESS WORD 2 FOR DEX (ADDRESS XXXX12)
1592 : 000014 T010BC=14 ;T010 BYTE COUNT (ADDRESS XXXX14)
1593 : 000016 T011BC=16 ;T011 BYTE COUNT (ADDRESS XXXX16)
1594 : 000020 T010AD=20 ;T010 PDP11 MEMORY ADDRESS (ADDRESS XXXX20)
1595 : 000022 T011AD=22 ;T011 PDP11 MEMORY ADDRESS (ADDRESS XXXX22)
1596 : 000024 T010DT=24 ;T010 PDP11 DATA WORD (ADDRESS XXXX24)
1597 : 000026 T011DT=26 ;T011 PDP11 DATA WORD (ADDRESS XXXX26)
1598
1599 ; THE LAST 4 REGISTERS ARE INITIALIZED BY "INIT" (BECAUSE THEY ARE IN FLIP-FLOPS
1600 : 000030 DIAG1=30 ;DIAGNOSTIC WORD 1 (ADDRESS XXXX30)
1601 : 000032 DIAG2=32 ;DIAGNOSTIC WORD 2 (ADDRESS XXXX32)
1602 : 000034 STATUS=34 ;10/11 INTERFACE STATUS WORD (ADDRESS XXXX34)
1603 : 000036 DIAG3=36 ;DIAGNOSTIC WORD 3 (ADDRESS XXXX36)
1604
1605
1606 ; THE FOLLOWING ARE THE ADDRESSES OF THE DTE20 INTERRUPT VECTORS
1607
1608 : 000774 DTEIV1=774 ;INTERRUPT VECTOR FOR DTE20 #1
1609 : 000770 DTEIV2=770 ;INTERRUPT VECTOR FOR DTE20 #2
1610 : 000764 DTEIV3=764 ;INTERRUPT VECTOR FOR DTE20 #3
1611 : 000760 DTEIV4=760 ;INTERRUPT VECTOR FOR DTE20 #4
1612
1613
1614 ; BIT ASSIGNMENTS FOR VARIOUS DTE20 REGISTERS USED BY THIS ROM CODE
1615
1616 ;BIT ASSIGNMENTS FOR T010BC
1617
1618 ;
1619 : 100000 INT11=BIT15 ;SET DONE AND INTERRUPT BOTH 10 AND 11
1620
1621 ;BIT ASSIGNMENTS FOR T011BC
1622
1623 : 100000 INT10=BIT15 ;SET DONE AND INTERRUPT BOTH 10 AND 11
1624 : 040000 ZSTOP=BIT14 ;STOP ON NULL (ZERO) CHARACTER
1625 : 020000 T011BM=BIT13 ;BYTE SIZE FOR 10-11 BYTE TRANSFERS
1626

```

```

1627 ;BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 9
1628 ;BM873-YD.P11 DTE20 DEVICE REGISTER AND BIT DEFINITIONS
1629
1630 ;BIT ASSIGNMENTS FOR DIAG2 (WRITE)
1631 ; 000100 DRESET=BIT6 ;PERFORM DIAGNOSTIC CLEAR
1632
1633 ;BIT ASSIGNMENTS FOR DIAG3 (READ)
1634 ; 000020 DUPE=BIT4 ;DATO UNIBUS PARITY ERROR
1635 ; 000004 DURE=BIT2 ;DATO UNIBUS RECEIVE ERROR
1636 ; 000002 NUPE=BIT1 ;NPR UNIBUS PARITY ERROR
1637
1638 ;BIT ASSIGNMENTS FOR DIAG3 (WRITE)
1639
1640 ; 000020 CDC=BIT4 ;CLEAR DUPE AND DURE ERROR FLAGS
1641 ; 000002 CNUPE=BIT1 ;CLEAR NUPE ERROR FLAG
1642 ; 000001 T010BM=BIT0 ;BYTE SIZE FOR T0-10 BYTE TRANSFER
1643
1644 ;BIT ASSIGNMENTS FOR STATUS (WRITE)
1645
1646 ; 100000 DON10S=BIT15 ;SET T010 DONE
1647 ; 040000 DON10C=BIT14 ;CLEAR T010 DONE
1648 ; 020000 ERR10S=BIT13 ;SET T010 ERROR
1649 ; 010000 ERR10C=BIT12 ;CLEAR T010 ERROR
1650 ; 004000 INT11S=BIT11 ;RING THE PDP-11'S DOORBELL (INTERRUPTS THE -11)
1651 ; 002000 INT11C=BIT10 ;STOP RINGING THE PDP-11'S DOORBELL
1652 ; 001000 PERCLR=E. '9 ;CLEAR -11 MEMORY PARITY ERROR
1653 ; 000400 INT10S=BIT8 ;RING THE PDP-10'S DOORBELL (INTERRUPTS THE -10)
1654 ; 000200 DON11S=BIT7 ;SET T011 DONE
1655 ; 000100 DON11C=BIT6 ;CLEAR T011 DONE
1656 ; 000040 INTRON=BITS ;ENABLE DTE20 INTERRUPTS TO THE -11
1657 ; 000020 EBUSPC=BIT4 ;CLEAR "EBUS PARITY ERROR"
1658 ; 000010 INTR0F=BIT3 ;DISABLE THE PDP-11 INTERRUPTS
1659 ; 000004 EBUSPS=BIT2 ;SET "EBUS PARITY ERROR"
1660 ; 000002 ERR11S=BIT1 ;SET T011 ERROR
1661 ; 000001 ERR11C=BIT0 ;CLEAR T011 ERROR
1662
1663 ;BIT ASSIGNMENTS FOR STATUS (READ)
1664
1665 ; 100000 T010DN=BIT15 ;T010 DONE
1666 ; 020000 T010ER=BIT13 ;TO 10 ERROR (NPR TIMEOUT OR BUS ERROR)
1667 ; 010000 RAMISO=BIT12 ;RAM WORD READ IS ALL ZEROS
1668 ; 004000 T011DB=BIT11 ;1 = THE PDP11'S DOORBELL IS RINGING
1669 ; 002000 DXMRD1=BIT10 ;DEPOSIT OR EXAMINE WORD ONE
1670 ; 001000 MPE11=BIT9 ;PARITY ERROR WITHIN PDP-11 MEMORY
1671 ; 000400 T010DB=BIT8 ;1 = THE PDP-10'S DOORBELL IS RINGING
1672 ; 000200 T011DN=BIT7 ;T011 DONE
1673 ; 000100 EBSEL=BIT6 ;E BUFFER SELECT
1674 ; 000040 NULSTP=BITS ;NULL STOP
1675 ; 000020 BPARER=BIT4 ;EBUS PARITY ERROR
1676 ; 000010 RSTRCT=BIT3 ;THIS PDP-11 IS "RESTRICTED"
1677 ; 000004 DEXDON=BIT2 ;DEPOSIT OR EXAMINE DONE
1678 ; 000002 T011ER=BIT1 ;TO 11 ERROR (NPR TIMEOUT OR BUS ERROR)
1679 ; 000001 INTSON=BIT0 ;DTE20 INTERRUPTS (TO THE -11) ARE ENABLED

```

1680 ;BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 1  
 1681 ;BM873-YD.P11 PROCEDURE BY WHICH THE PDP-10 BOOTSTRAPS AND/OR DUMPS THE PDP-11

1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696  
1697  
1698  
1699  
1700  
1701  
1702  
1703  
1704  
1705  
1706  
1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733

THE FOLLOWING IS THE PROCEDURE WHICH THE KL10 EXECUTES IN ORDER TO DUMP AND/OR BOOTSTRAP THE PDP-11 THROUGH THE DTE20:

1. CLEAR THE DTE20 AND INITIATE A BM873 BUTTON #4 BOOTSTRAP OPERATION  
- CONO [SR11B!CL1IPT!CLT011!CLT010!PILDEN]
2. WAIT TO SEE PDP-11 POWER FAIL (AC LOW = TRUE) - CONI [DEAD11] = 1
3. WAIT TO SEE PDP-11 POWER RECOVER (AC LOW = FALSE) - CONI [DEAD11] = 0
4. WAIT AT LEAST ANOTHER 150 MILLISECONDS AND THEN CLEAR THE RELOAD -11 BUTTON  
- CONO [CR11B]
5. SET BYTE COUNTER TO A SPECIAL CODE (1365 OCTAL) - DATA0 [1365]
6. RING PDP-11'S DOORBELL - CONO[TO11DB]
7. WAIT UNTIL "-10 RINGING -11'S DOORBELL" IS TURNED OFF BY THE -11  
(I.E. UNTIL CONI[TO11DB] BECOMES ZERO).
8. ENABLE THE DTE20 TO USE PI 0 INTERRUPTS  
(I.E. SET CONO[PILDEN!PIDENB]).
9. SET UP THE TO-10 BYTE POINTER (IN THE EPT) FOR THE FIRST 3.5K.
10. SET UP THE BYTE COUNTER FOR THE FIRST 3.5K, INDICATING  
"INTERRUPT -10 ONLY" - DATA0 [1000]
11. WAIT FOR "TO-10 DONE" OR "TO-10 ERROR" - CONI [TO10DN!TO10ER]
12. NOTE WHETHER THERE WAS AN ERROR (CONI [TO10ER]) AND THEN TURN OFF  
TO10DN AND TO10ER - CONO [CLT010]. IF ERROR, GO TO STEP 17.
13. IF END OF 28K, GO TO STEP 17.
14. SET UP TO-10 BYTE POINTER (IN THE EPT) FOR THE NEXT 3.5K.
15. SET UP THE BYTE COUNTER FOR THE NEXT 3.5K INDICATING  
"INTERRUPT -10 ONLY" (DATA0 [1000]), UNLESS THIS IS THE  
LAST 3.5K (OF 28K), IN WHICH CASE INDICATE "INTERRUPT  
BOTH PROCESSORS" (DATA0 [TO10IB!1000]).
16. GO TO STEP 11.

1734 ;BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 1  
 1735 ;BM873-YD.P11 PROCEDURE BY WHICH THE PDP-10 BOOTSTRAPS AND/OR DUMPS THE PDP-11  
 1736  
 1737  
 1738 17. SET UP TO-11 BYTE POINTER (IN THE EPT) FOR "PDP-11 BOOTSTRAP".  
 1739 NOTE THAT THE FIRST WORD OF THIS "PDP-11 BOOTSTRAP" MUST  
 1740 BE THE BIT PATTERN 00C240 (A PDP-11 NOP INSTRUCTION).  
 1741  
 1742 18. RING THE PDP-11'S DOORBELL - CONO [TO11DB]  
 1743  
 1744 19. WAIT FOR EITHER TO11DB TO GO OFF (CONI[TO11DB] = 0),  
 1745 OR TO10DB TO COME ON (CONI[TO10DB] = 1).  
 1746  
 1747 20. IF NO ERROR WAS NOTED IN STEP 12, TO11DB SHOULD GO OFF  
 1748 (TO10DB COMING ON INDICATES A MASSIVE SCREWUP).  
 1749 IF AN ERROR WAS NOTED IN STEP 12, TO11DB GOING OFF INDICATES  
 1750 THAT THE ERROR WAS "NON-FATAL" (NON-EX-MEM OR -11 MEMORY  
 1751 PARITY) AND THE -11 IS PROCEEDING. TO10DB COMING ON INDICATES  
 1752 THAT THE ERROR WAS "FATAL" AND THE -11 IS HALTED AT LOCATION 173714.  
 1753 IN THIS LATTER CASE THE -10 MUST RESTART FROM STEP 1.  
 1754  
 1755 21. IF TO11DB WENT OFF, WAIT FOR "TO-11 DONE" OR "TO-11 ERROR"  
 1756 - CONI [TO11DN!TO11ER]  
 1757  
 1758 22. NOTE WHETHER THERE WAS AN ERROR - CONI [TO11ER]  
 1759  
 1760 23. TURN OFF TO11DN AND TO11ER AND RING THE PDP-11'S DOORBELL  
 1761 - CONO [TO11DB!CLTO11]  
 1762  
 1763 24. WAIT FOR EITHER TO11DB TO GO OFF (CONI[TO11DB] = 0),  
 1764 OR TO10DB TO COME ON (CONI[TO10DB] = 1).  
 1765  
 1766 25. TO11DB GOING OFF INDICATES THAT THE PDP-11 FOUND NO ERRORS  
 1767 AND IS TRANSFERRING CONTROL TO THE CODE WHICH WAS JUST  
 1768 RECEIVED FROM THE -10. IN THIS CASE THE -10 SHOULD START  
 1769 FOLLOWING THE PROTOCOL OF THIS CODE.  
 1770  
 1771 26. TO10DB COMING ON INDICATES THAT THE PDP-11 HAS FOUND AN  
 1772 ERROR (OR THAT THE FIRST WORD TRANSMITTED WASN'T THE  
 1773 BIT PATTERN 00C240), AND THE PDP-11 IS HALTED AT LOCATION 173766.  
 1774 IN THIS CASE THE -10 MUST RESTART FROM STEP 1.  
 1775  
 1776  
 1777  
 1778  
 1779  
 1780



```

1781 :BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 1
1782 ;BM873-YD.P11 BUTTON #4 - BOOTSTRAP INITIATED BY THE PDP-10 (THROUGH DTE20)
1783
1784
1785 ; 000130 DTECOR = 130 ;CORE ADDRESS INTO WHICH TO STORE DTE20 REGS.
1786 ; 000014 DTEREG = 1012 ;NUMBER OF DTE20 REGISTERS TO STORE
1787 ; 000400 DTEWDC = 10256 ;WORD COUNT FOR SECONDARY BOOTSTRAP FROM THE -10
1788 ; ENTER HERE WHEN THE DTE20 PRESSES BUTTON #4 (BOOTSTRAP INITIATED
1789 ; BY THE PDP-10, THROUGH THE DTE20)
1790 004534 010037 ;173534 010037 BUTON#4: MOV RO,#ROTOR7 ;SAVE RO IN PDP-11 MEMORY LOCATION "ROTOR7"
1791 004536 000040 ;173536 000040
1792 004540 010700 ;173540 010700 MOV PC,RO ;USE RO FOR A SUBROUTINE RETURN ADDRESS
1793 004542 000644 ;173542 000644 BR REGSAV ;GO TO THE "REGISTER SAVING" SUBROUTINE
1794 004544 005005 ;173544 005005 CLR R5 ;SET R5 = 0
1795 004546 012501 ;173546 012501 MOV (R5)+,R1 ;SAVE LOCATION 0 IN R1
1796 004550 012503 ;173550 012503 MOV (R5)+,R3 ;SAVE LOCATION 2 IN R3
1797 004552 012504 ;173552 012504 MOV (R5)+,R4 ;SAVE LOCATION 4 IN R4
1798 004554 011500 ;173554 011500 MOV (R5),R0 ;SAVE LOCATION 6 IN R0
1799 004556 012715 ;173556 012715 MOV #PR7,(R5) ;SET UP PRIORITY FOR NON-EX-MEM TRAP
1800 004560 000340 ;173560 000340
1801 004562 005745 ;173562 005745 TST -(R5) ;SET R5 = 4
1802 004564 012702 ;173564 012702 10$: MOV #DTEBAS-DTESIZ,R2
1803 004566 174340 ;173566 174340
1804 004570 010715 ;173570 010715 MOV PC,(R5) ;STORE ADDRESS FOR NON-EX-MEM TRAP
1805 004572 010506 ;173572 010506 MOV R5,SP ;SET STACK POINTER = 4
1806 004574 062702 ;173574 062702 11$: ADD #DTESIZ,R2 ;R2 POINTS TO THE NEXT DTE20
1807 004576 000040 ;173576 000040
1808 004600 105702 ;173600 105702 TSTB R2
1809 004602 100770 ;173602 100770 BMI 10$ ;START LOOKING FROM THE BEGINNING AGAIN
1810 004604 032762 ;173604 032762 BIT #T011DB,STATUS(R2) ;IS THIS -10 RINGING THE -11'S DOORBELL?
1811 004606 004000 ;173606 004000
1812 004610 000034 ;173610 000034
1813 004612 001770 ;173612 001770 BEQ 11$ ;IF IT IS NOT, GO LOOK FOR ANOTHER -10
1814 004614 026217 ;173614 026217 CMP T010BC(R2),(PC) ;CHECK FOR A CODE (1365) FROM THE PDP-10
1815 004616 000014 ;173616 000014
1816 ;INDICATING THAT IT WANTS TO BOOTSTRAP THE -11
1817 004620 001365 ;173620 001365 BNE 11$
1818 ; NOTE THAT AT THIS POINT R2 CONTAINS THE ADDRESS OF THE DEVICE REGISTER
1819 ; BLOCK FOR THIS DTE20, THAT R5 = 4, AND THAT SP = 4
1820 004622 005725 ;173622 005725 TST (R5)+ ;SET R5 = 6
1821 004624 010015 ;173624 010015 MOV RO,(R5) ;RESTORE THE CONTENTS OF LOCATION 6
1822 004626 010445 ;173626 010445 MOV R4,-(R5) ;RESTORE THE CONTENTS OF LOCATION 4
1823 004630 010345 ;173630 010345 MOV R3,-(R5) ;RESTORE THE CONTENTS OF LOCATION 2
1824 004632 010145 ;173632 010145 MOV R1,-(R5) ;RESTORE THE CONTENTS OF LOCATION 0
1825 ; NOTE: AT THIS TIME R5 = 0. THIS FACT WILL BE USED LATER.
1826 004634 012700 ;173634 012700 MOV #DTECOR,RO ;RO = CORE ADDRESS FOR STORING DTE20 REGISTERS
1827 004636 000130 ;173636 000130
1828 004640 010204 ;173640 010204
1829 004642 012420 ;173642 012420 7$: MOV R2,R4
1830 004644 022700 ;173644 022700 MOV (R4)+,(RO)+ ;SAVE THE NEXT DTE20 REGISTER IN CORE
1831 004646 000160 ;173646 000160 CMP #<DTEREG*2>+DTECOR,RO ;HAVE WE FINISHED YET?
1832 004650 101374 ;173650 101374 BHI 7$ ;LOOP UNTIL WE HAVE FINISHED

```

```

1833      ;BM873-YD          - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17)  MACY11 27(657) 18-DEC-74 11:59 PAGE 1
1834      ;BM873-YD.P11    BUTTON #4 - BOOTSTRAP INITIATED BY THE PDP-10 (THROUGH DTE20)
1835
1836      004652 010201 ;173652 010201      MOV      R2,R1          ;R1 = DTE20 DEVICE REGISTER BLOCK
1837      004654 062701 ;173654 062701      ADD      #DIAG2,R1
1838      004656 000032 ;173656 000032
1839      004660 012721 ;173660 012721      MOV      #DRESET,(R1)+ ;DO A "DIAGNOSTIC CLEAR" OF THE DTE20,
1840      004662 000100 ;173662 000100
1841      ; THE ABOVE OPERATION IS NECESSARY TO CLEAR THE "BYTE COUNT LOADED" FLAG
1842      ; AND SIMULTANEOUSLY TO TURN OFF "-10 RINGING -11'S DOORBELL".
1843      004664 005012 ;173664 005012      CLR      (R2)          ;SET DTE20 FOR NO DELAY
1844      004666 005062 ;173666 005062      CLR      T010AD(R2)   ;START WRITING -11 MEMORY INTO THE -10.
1845      004670 000020 ;173670 000020
1846      004672 032711 ;173672 032711      6$:     BIT      #T011DB,(R1) ;HAS THE -10 RUNG THE -11'S DOORBELL?
1847      004674 004000 ;173674 004000
1848      004676 001775 ;173676 001775      BEQ      6$           ;LOOP UNTIL IT HAS.
1849      004700 032762 ;173700 032762      BIT      #DUPE!DURE!NUPE,DIAG3(R2) ;"FATAL" ERROR?
1850      004702 000026 ;173702 000026
1851      004704 000036 ;173704 000036
1852      004706 001403 ;173706 001403      BEQ      8$           ;BRANCH IF NO "FATAL" ERROR
1853      004710 012711 ;173710 012711      MOV      #T010DB,(R1) ;SIGNAL "FATAL" ERROR TO THE PDP-10
1854      004712 000400 ;173712 000400
1855      004714 000000 ;173714 000000      2$:     HALT     ;HALT DUE TO "FATAL" ERROR
1856      004716 012762 ;173716 012762      8$:     MOV      #DRESET,DIAG2(R2) ;RESET AFTER POSSIBLE PDP-11
1857      004720 000100 ;173720 000100
1858      004722 000032 ;173722 000032
1859      ; MEMORY PARITY ERROR OR NON-EX-MEM ERROR, AND ALSO TURN OFF
1860      ; "-10 RINGING -11'S DOORBELL".
1861      004724 005062 ;173724 005062      3$:     CLR      T011AD(R2) ;START INPUTTING AT LOCATION 0
1862      004726 000022 ;173726 000022
1863      004730 012762 ;173730 012762      MOV      #INT10!<<-DTEWDC>&7777>,T011BC(R2) ;READ IN 256 WORDS
1864      004732 107400 ;173732 107400
1865      004734 000016 ;173734 000016
1866      004736 032711 ;173736 032711      1$:     BIT      #T011DB,(R1) ;HAS THE -10 RUNG THE -11'S DOORBELL?
1867      004740 004000 ;173740 004000
1868      004742 001775 ;173742 001775      BEQ      1$           ;LOOP UNTIL IT HAS.
1869      004744 132711 ;173744 132711      4$:     BITB     #T011DN:T011ER,(R1) ;IS THE TRANSMISSION FINISHED?
1870      004746 000202 ;173746 000202
1871      004750 001775 ;173750 001775      BEQ      4$           ;LOOP UNTIL IT IS FINISHED
1872      004752 100003 ;173752 100003      BPL      5$           ;IF "T011DN" ISN'T ON, "T011ER" MUST BE ON
1873      004754 022715 ;173754 022715      CMP      #000240,(R5) ;CHECK FOR BIT PATTERN IN LOCATION ZERO
1874      004756 000240 ;173756 000240
1875      004760 001403 ;173760 001403      BEQ      9$           ;UNLESS THERE IS A "NOP" IT IS AN ERROR
1876      004762 012711 ;173762 012711      5$:     MOV      #T010DB,(R1) ;SIGNAL THE -10 THAT THERE WAS AN ERROR
1877      004764 000400 ;173764 000400
1878      004766 000000 ;173766 000000      12$:    HALT     ;THIS ERROR HALT IS BECAUSE EITHER "T011ER"
1879      ; IS ON, OR BECAUSE THE BIT PATTERN READ INTO LOCATION ZERO WASN'T "000240".
1880      004770 012762 ;173770 012762      9$:     MOV      #DRESET,DIAG2(R2) ;SIGNAL THE -10 THAT EVERYTHING IS OK
1881      004772 000100 ;173772 000100
1882      004774 000032 ;173774 000032
1883      004776 END.YD:
1894      004776 000115 ;173776 000115      JMP      (R5)         ;JUMP TO LOCATION ZERO
1885      ; 000001 .END

```

1886 005000 MAP.YF:  
1887  
1888  
1889  
1890  
1891  
1892  
1893  
1894  
1895  
1896  
1897  
1898  
1899  
1900  
1901  
1902  
1903  
1904  
1905  
1906  
1907  
1908  
1909  
1910  
1911  
1912  
1913  
1914  
1915  
1916  
1917  
1918  
1919  
1920  
1921  
1922  
1923  
1924  
1925  
1926  
1927  
1928  
1929  
1930  
1931  
1932  
1933  
1934  
1935  
1936

THE FOLLOWING IS A REPRODUCTION  
OF THE ROM PROGRAM FOR BM873YF.  
IT IS HERE FOR COMPARISON TO THE  
ACTUAL ROM AND FOR REFERENCE  
;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 1

..... TITLE PAGE  
BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23)  
..... COPYRIGHT (C) 1975 DIGITAL EQUIPMENT CORPORATION  
ALL RIGHTS RESERVED  
..... THIS IS THE CODE TO BE ENCODED IN THE BOOTSTRAP ROM ON THE BM873-YF BOARD  
.....  
MODULE: BM873F  
DATE: 17-JUN-75  
AUTHOR: TOM PORCHER  
.....

ENABLE ABS,AMA

..... 177776 PS=177776 ; PROCESSOR STATUS REGISTER  
177570 SWR=177570 ; FRONT PANEL SWITCH REGISTER  
000000 PR0=0\*40 ; PRIORITY LEVEL 0  
000040 PR1=1\*40 ; PRIORITY LEVEL 1  
000100 PR2=2\*40 ; PRIORITY LEVEL 2  
000140 PR3=3\*40 ; PRIORITY LEVEL 3  
000200 PR4=4\*40 ; PRIORITY LEVEL 4  
000240 PR5=5\*40 ; PRIORITY LEVEL 5  
000300 PR6=6\*40 ; PRIORITY LEVEL 6  
000340 PR7=7\*40 ; PRIORITY LEVEL 7  
000001 BIT0=000001  
000002 BIT1=000002  
000004 BIT2=000004  
000010 BIT3=000010  
000020 BIT4=000020  
000040 BIT5=000040  
000100 BIT6=000100  
000200 BIT7=000200  
000400 BIT8=000400  
001000 BIT9=001000  
002000 BIT10=002000  
004000 BIT11=004000  
010000 BIT12=010000  
020000 BIT13=020000  
040000 BIT14=040000  
100000 BIT15=100000  
177400 HIBYTE=177400

```

1937 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 1
1938 ;
1939 ; DIRECTIVE FUNCTION CODES
1940 ;
1941 ; 000001 DR.DTE=1. ;DTE EXAMINE/DEPOSIT/INITALIZE/DOORBELL FUNCTIONS
1942 ;
1943 ; DTE FUNCTION CODES (LOW ORDER BY BYTE)
1944 ;
1945 ; 000001 DF.DOR=1 ;DOOR BELL FUNCTION CODE
1946 ; 000002 DF.OFF=2 ;DTE OFF FUNCTION
1947 ; 000003 DF.ON=3 ;DTE ON FUNCTION
1948 ; 000004 DF.DMG=4 ;DEPOSIT MY GENERAL FUNCTION
1949 ; 000005 DF.EMG=5 ;EXAMINE MY GENERAL FUNCTION
1950 ; 000006 DF.EMN=6 ;EXAMINE MY FOR N FUNCTION
1951 ; 000007 DF.DMN=7 ;DEPOSIT MY FOR N FUNCTION
1952 ; 000010 DF.EHG=10 ;EXAMINE HIS GEN SECTION FUNCTION
1953 ; 000011 DF.EHM=11 ;EXAMINE HIS SECTION FOR ME FUNCTION
1954 ; 000012 DF.KLR=12 ;DIAGNOSTIC KL READ
1955 ; 000013 DF.KLW=13 ;DIAGNOSTIC KL WRITE (FUNCTION 13)
1956 ; 000014 DF.KLX=14 ;DIAGNOSTIC KL EXECUTE (FUNCTION 14)
1957 ; 000015 DF.PEX=15 ;PRIVILEGED EXAMINE (FUNCTION 15)
1958 ; 000016 DF.PDP=16 ;PRIVILEGED DEPOSIT (FUNCTION 16)
1959 ;
1960 ; CRASH CODES
1961 ;
1962 ; 000001 CC.ILD=1 ;ILLEGAL DIRECTIVE
1963 ; 000002 CC.EMT=2 ;ILLEGAL EMT
1964 ; 000003 CC.IDI=3 ;ILLEGAL DTE INTERRUPT
1965 ; 000004 CC.IOT=4 ;IOT TRAP
1966 ; 000005 CC.RES=5 ;RESERVED INSTRUCTION TRAP
1967 ; 000006 CC.TBT=6 ;T BIT OR BPT TRAP
1968 ; 000007 CC.TRP=7 ;TRAP INSTRUCTION TRAP
1969 ; 000010 CC.TO4=10 ;TRAP TO 4
1970 ; 000011 CC.UNT=11 ;ILLEGAL TRAP (UNKNOWN TRAP)
1971 ; 000012 CC.MPE=12 ;MEMORY PARITY ERROR
1972 ; 000013 CC.NPF=13 ;RESTRICTED FRONT CAN'T EXECUTE BOOT PROTOCOL
1973 ; 000014 CC.PTB=14 ;PROTOCOL (PRIMARY) BROKEN
1974 ; 000015 CC.CST=15 ;CLOCK STOPPED
1975 ; 000016 CC.ILC=16 ;ILLEGAL COMMAND
1976 ; 000017 CC.IPO=17 ;INPUT TTY OVERFLOW
1977 ; 000020 CC.IAS=20 ;INCORRECT VALUE IN .SERFG
1978 ; 000021 CC.NCE=21 ;NOT ENOUGH ENTRIES IN CLOCK QLEUE
1979 ; 000022 CC.PIT=22 ;CAN'T EXIT PERMANENT TASK
1980 ; 000023 CC.UMP=23 ;LOAD REQUEST NOT IMPL YET
1981 ; 000024 CC.EPE=24 ;E BUS PARITY ERROR
1982 ; 000025 CC.NOE=25 ;NOT ENOUGH ENTRYS FOR DTE20
1983 ; 000026 CC.DEX=26 ;DEXDONE TIMEOUT
1984 ; 000027 CC.TET=27 ;TO TEN ERROR
1985 ; 000030 CC.ETE=30 ;TO ELEVEN ERROR
1986 ; 000031 CC.MTF=31 ;MARK TIME FAILURE
1987 ; 000032 CC.NON=32 ;NOT ENOUGH NODES
1988 ; 000033 CC.TSP=33 ;TEN STOPPED
1989 ; 000034 CC.UIE=34 ;UNIMPLEMENTED FUNCTION
1990 ; 000035 CC.ILQ=35 ;ILLEGAL QUEUE

```

```

1991 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 1
1992 ;
1993 ;
1994 ; GENERAL PROCESSOR DEFINITIONS
1995 ;
1996 ; 000340 PRI7=340 ;PROCESSOR PRIORITY 7
1997 ;
1998 ;
1999 ; DTE20 REGISTER DEFINITIONS
2000 ;
2001 ; THESE LABELS ARE THOSE USED IN THE FRONT END INTERFACE SPEC
2002 ; EXCEPT STATUS WHICH CONFLICTS WITH PROTOCOL SPEC
2003 ;
2004 ; PDM# 200-200-012-00
2005 ;
2006 ;
2007 ; 174400 DLYCNT=174400 ;DELAY COUNT WORD
2008 ; 174402 DEXWD3=174402 ;DEPOSIT OR EXAMINE WORD 3
2009 ; 174404 DEXWD2=174404 ;DEPOSIT OR EXAMINE WORD 2
2010 ; 174406 DEXWD1=174406 ;DEPOSIT OR EXAMINE WORD 1
2011 ; 174410 TENAD1=174410 ;TEN ADDRESS WORD 1
2012 ; 174412 TENAD2=174412 ;TEN ADDRESS WORD 2
2013 ; 174414 T010BC=174414 ;TO-10 PDP-11 MEMORY ADDRESS
2014 ; 174416 T011BC=174416 ;TO-11 BYTE COUNT
2015 ; 174420 T010AD=174420 ;TO-10 PDP-11 MEMORY ADDRESS
2016 ; 174422 T011AD=174422 ;TO-11 PDP-11 MEMORY ADDRESS
2017 ; 174424 T010DT=174424 ;TO-10 PDP-11 DATA WORD
2018 ; 174426 T011DT=174426 ;TO-11 PDP-11 DATA WORD
2019 ; 174430 DIAG1=174430 ;DIAGNOSTIC WORD 1
2020 ; 174432 DIAG2=174432 ;DIAGNOSTIC WORD 2
2021 ; 174434 STAT=174434 ;STATUS WORD
2022 ; 174436 DIAG3=174436 ;DIAGNOSTIC WORD 3
2023 ;
2024 ;
2025 ; EXTERNAL PAGE DEFINITIONS (DEVICE DEFINITIONS)
2026 ;
2027 ;
2028 ; DTE DEFINITIONS
2029 ;
2030 ;
2031 ; REGISTER BIT DEFINITIONS
2032 ;
2033 ;
2034 ; TENAD1 DEFINITIONS
2035 ;
2036 ; 010000 DEP=010000 ;DEPOSIT (BIT 12)
2037 ; 004000 PRTOFF=004000 ;EXAMINE/DEPOSIT PROTECT OFF
2038 ; 100000 PHYS=100000 ;PHYSICAL EXAMINE
2039 ;
2040 ; T011BC DEFINITIONS
2041 ;
2042 ; 100000 IFLOP=100000 ;I FLIPFLOP BIT
2043 ; 040000 ZSTOP=040000 ;ZSTOP
2044 ; 020000 T011BM=020000 ;TO 11 BYTE MODE

```

2045 :BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 1

2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091

DIAG1 DEFINITIONS

```

004000 DS04=004000 ;KL CLOCK ERROR STOP
002000 DS05=002000 ;RJM
001000 DS06=001000 ;HALT
000400 DEX=000400 ;DEPOSIT OR EXAMINE MAJOR STATE
000200 T010=000200 ;TO 10
000200 DFUNC=000200
000100 T011=000100 ;TO 11 TRANSFER MAJOR STATE
000040 D1011=000040 ;DIAGNOSE 10/11 INTERFACE
000020 PULSE=000020 ;SINGLE CLOCK CYCLE
000010 DIKL10=000010 ;DIAGNOSTIC MODE SWITCH
000004 DSEND=000004 ;SEND DATA
000001 DCOMST=000001 ;DIAGNOSTIC COMMAND START
    
```

DIAG1 FUNCTIONS

```

000000 .STPCL=0 ;STOP THE KL CLOCK
001000 .STRCL=01*1000 ;START THE KL CLOCK
002000 .SSCLK=02*1000 ;SINGLE STEP THE M BOX CLOCK
003000 .SECLK=03*1000 ;SINGLE STEP THE EBOX CLOCK. LEAVES THE
;EBOX CLOCK FALSE AND EBOX SYNC TRUE.
;CAUSES (2,3) MBOX CLOCKS DEPENDING ON
;EBOX CLOCK INITIALLY (FALSE TRUE).
;DOES NOT DEPEND ON 'T' FIELD OR MB WAIT.
004000 .CECLK=04*1000 ;CONDITIONALLY ISSUE AN EBOX CLOCK IF THE EBOX
;CLOCK IS TRUE. MAKES EBOX CLOCK FALSE.
;IF ISSUED IN THE MASTER RESET STATE.
;LEAVES EBOX SYNC TRUE.
005000 .BRCLK=05*1000 ;ISSUE A BURST OF THE CLOCKS. THE NUMBER
;OF MBOX CLOCKS DESIRED (1-255) HAS BEEN
;BEEN LOADED PREVIOUSLY BY FUNCTIONS LDBRR,LDBRL
;(42,43)
006000 .CLRMR=06*1000 ;CLEAR MASTER RESET STATE
007000 .SETMR=07*1000 ;SET MASTER RESET STATE. RUNNING THE CLOCK WHILE IN THIS
;STATE 'CLEARS' THE KL10.
010000 .CLRUN=10*1000 ;CLEAR THE RUN FLOP. MAKE THE MICRO CODE GO TO
;THE HALT-LOOP.
011000 .SETRN=11*1000 ;SET THE RUN FLOP. ALLOW REPEATED INSTRUCTION EXECUTION
012000 .CONBT=12*1000 ;SET THE CONTINUE FLOP (MOMENTARY). ALLOW THE
;MICRO CODE TO LEAVE THE HALT LOOP
014000 .IRLTC=14*1000 ;UNLATCH THE IR AND LOAD IT FROM THE AD.
015000 .DRLTC=15*1000 ;UNLATCH THE DRAM REGISTER AND ALLOW IT TO LOAD FROM THE
;RAMS
    
```

:BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 1

2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129

CLOCK LOAD FUNCTIONS

```

042000 .LDBRR=42*1000 ;LOAD THE RIGHT HAND 4 BITS OF THE 8 BIT
;BURST COUNTER FROM EBUS BITS 32-35
043000 .LDBRL=43*1000 ;LOAD THE LEFT HAND 4 BITS OF THE BURST CTR.
044000 .LDSEL=44*1000 ;LOAD THE CLOCK SOURCE AND RATE SELECT
;REGISTER: 32 33 34 35
;SOURCE RATE
;00 NORM XTL 00
;01 FAST XTL 01 /2
;10 EXT 10 /4
;11 UNDEF 11 /8
045000 .LDDIS=45*1000 ;LOAD THE REGISTER WHICH CONTROLS THE EBOX CLOCK
;DISTRIBUTION.
;BIT ACTION
;33 DISABLE CONTROL LOGIC CLOCK
;34 DISABLE CONTROL RAM CLOCK
;35 DISABLE DATA PATHS CLOCK
046000 .LDCK1=46*1000 ;LOAD THE CONDITION-CHECKING ENABLE REGISTER.
;THESE ALL ENABLE THE CLOCK TO STOP AND SHOULD
;BE USED IN CONJUNCTION WITH BIT 35 OF FUNCTION 47
;BIT FUNCTION
;32 CHECK FM PARITY
;33 CHECK CRAM PARITY
;34 CHECK DRAM PARITY
;35 CHECK FIELD SERVICE PPOBE
047000 .LDCK2=47*1000 ;LOAD THE ENABLE/DISABLE FUNCTION REGISTER
;BIT FUNCTION
;32 DISABLE EBOX REQUESTS TO MBOX
;33 SIMULATE AN MB RESP FOR EACH MB WAIT
;34 CHECK AR AND ARX PARITY AND CAUSE A
;PAGE FAIL UCODE TRAP IF ERROR
;35 MUST BE SET TO PERFORM DESIRED ACTION OF
;FUNCTION 46 (ABOVE). STOPS ALL CLOCKS IF AN ERROR
;IS DETECTED.

```



```

2130 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 1
2131 ; CONTROL RAM LOAD FUNCTIONS
2132 057000 .LCRM1=57*1000 ;EBUS CRAM
2133 ;:08-11 00-03
2134 ;:14-17 04-07
2135 ;:20-23 08-11
2136 ;:26-29 12-15
2137 ;:32-35 16-19
2138 056000 .LCRM2=56*1000 ;08-11 20-23
2139 ;:14-17 24-27
2140 ;:20-23 28-31
2141 ;:26-29 32-35
2142 ;:32-35 36-39
2143 055000 .LCRM3=55*1000 ;08-11 40-43
2144 ;:14-17 44-47
2145 ;:20-23 48-51
2146 ;:26-29 52-55
2147 ;:32-35 56-59
2148 054000 .LCRM4=54*1000 ;08 60
2149 ;:10 62
2150 ;:14 64
2151 ;:16 66
2152 ;:20 68
2153 ;:22 70
2154 ;:26 72
2155 ;:28 74
2156 ;:32 76
2157 ;:34 78
2158 053000 .LCRM5=53*1000 ;01-05 DISP 00-04
2159 052000 .LCRDL=52*1000 ;01-05 CRAM DIAG ADDRES 00-04
2160 051000 .LCRDR=51*1000 ;00-05 CRAM DIAG ADR 05-10
2161 ;
2162 ; DRAM LOAD FUNCTIONS
2163 ;
2164 060000 .LDRM1=60*1000 ;12-14 DRAM A00-02, EVEN ADDRESSES
2165 ;:15-17 DRAM B00-02, EVEN ADDRESSES
2166 061000 .LDRM2=61*1000 ;12-14 DRAM A00-02, ODD ADDRESSES
2167 ;:15-17 DRAM B00-02, ODD ADDRESSES
2168 062000 .LDRM3=62*1000 ;14-17 COMMON J01-04
2169 063000 .LDRJV=63*1000 ;15-17 J08-10, EVEN ADDRESSES
2170 ;:12 PARIT BIT, EVEN ADDRESSES
2171 064000 .LDRJD=64*1000 ;14 COMMON J07 (NOTE -- J05,6 DO NOT EXIST)
2172 ;:15-17 J08-10, ODD ADDRESSES
2173 ;:12 PARITY BIT, ODD ADDRESSES
2174 ;
2175 ; IR, DRAM CONTROL FUNCTIONS
2176 ;
2177 065000 .DSIOJ=65*1000 ;DISABLES SPECIAL DECODE OF OPCODES 254,7XX
2178 066000 .DSACF=66*1000 ;DISABLE IR AC OUTPUTS
2179 067000 .EIOJA=67*1000 ;ENABEL KL STYLE DECODING OF CODES AND AC'S
2180 ;
2181 070000 .INICL=70*1000 ;INIT CHANNELS
2182 ;
2183 071000 .WRMBX=71*1000 ;WRITE M-BOX

```

```

2184 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 1
2185 ;
2186 ; 076000 .MEMRS=76*1000 ;SET KL10 MEM RESET FLOP
2187 ;
2188 ; 147000 .RCRM1=147*1000 ;READ C-RAM BITS 0-19
2189 ; 146000 .RCRM2=146*1000 ;READ C-RAM BITS 20-39
2190 ; 145000 .RCRM3=145*1000 ;READ C-RAM BITS 40-59
2191 ; 144000 .RCRM4=144*1000 ;READ C-RAM BITS 60-79
2192 ;
2193 ; 141000 .RCSPF=141*1000 ;READ SPEC FIELD OF C-RAM
2194 ;
2195 ; 135000 .RDJ71=135*1000 ;READ J07-J10 OF D-RAM
2196 ; 134000 .RDJ14=134*1000 ;READ J01-J04 OF D-RAM
2197 ; 133000 .ROMAB=133*1000 ;READ A & B FIELD OF D-RAM
2198 ;
2199 ; 164000 .CSHRG=164*1000
2200 ; 102000 .GFNR=102*1000
2201 ;
2202 ; ;NOTE CONSOLE SOFTWARE MUST PERFORM THIS AS A PART OF
2203 ; ;MASTER RESET CODE
2204 ;
2205 ; ;LOAD AR FUNCTION
2206 ;
2207 ; 077000 .LDAR=77*1000 ;LOAD THE AR FROM EBUS 0-35
2208 ;
2209 ; 150000 .PCAB1=150*1000 ;PC-ADDRESS BREAK REGISTERS
2210 ; 151000 .PCAB2=151*1000
2211 ; 152000 .PCAB3=152*1000
2212 ; 153000 .PCAB4=153*1000
2213 ;
2214 ; ;DIAG3 DEFINITIONS
2215 ;
2216 ;
2217 ; 100000 SMSLLT=100000 ;SWAP SELECT LEFT
2218 ; 040000 DPS4=040000 ;PARITY
2219 ; 000040 SCD=000040 ;SHIFT CAPTURED DATA
2220 ; 000020 DUPE= 000020 ;DATO UNIBUS PARITY ERROR
2221 ; 000020 CDD=000020 ;CLEAR DUPE AND DURE ERROR FLAGS
2222 ; 000010 MEP=000010 ;WRITE EVEN (BAD) PARITY
2223 ; 000004 DUPE=000004 ;DATO UNIBUS RECEIVE ERROR
2224 ; 000002 NJPE=000002 ;NPR UNIBUS PARITY ERROR
2225 ; 000002 CNUPE=000002 ;CLEAR NUPE
2226 ; 000001 TO10BM=000001 ;TO-10 BYTE TRANSFER MODE
2227 ;
2228 ; ;DIAG2 DEFINITIONS
2229 ;
2230 ;
2231 ; 100000 RFMA00=100000 ;RFM ADDRESS BIT 0
2232 ; 040000 RFMA01=040000 ;RFM ADDRESS BIT 1
2233 ; 040000 EDONES=040000 ;EBUS DONE
2234 ; 020000 RFMA02=020000 ;RFM ADDRESS BIT 2
2235 ; 010000 RFMA03=010000 ;RFM ADDRESS BIT 3
2236 ; 000100 DRESET=000100 ;DTE RESET

```

2237 ; BMB73F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 1

STAT DEFINITIONS

2238	100000	TO10DN=100000	; TO-10 NORMAL TERMINATION
2239	100000	DON10S=100000	; NORMAL TERMINATION (DONE) TO 10
2240	040000	DON10C=040000	; TO-10 NORMAL TERMINATION STATUS
2241	020000	TO10ER=020000	; TO-10 ERROR TERMINATION
2242	020000	ERR10S=020000	; ERROR TERMINATION STATUS
2243	010000	RAMIS0=010000	; RAM IS ZEROS
2244	010000	ERR10C=010000	; CLEAR TO-10 ERROR TERMINATION
2245	004000	TO1108=004000	; -10 REQUESTED -11 INTERRUPT
2246	004000	INT11S=004000	; REQ 11 STATUS
2247	002000	DEXWRD1=002000	; DEXWRD 1
2248	002000	INT11C=002000	; -10 REQUESTS -11 INTERRUPT STATUS
2249	001000	MPE11=001000	; -11 MEMORY PARITY ERROR
2250	001000	PEPCLR=001000	; CLEAR -11 MEMORY PARITY ERROR FLAG STATUS
2251	000400	TO1008=000400	; -11 REQUEST -10 INTERRUPT
2252	000400	INT10S=000400	; REQUEST -10 INTERRUPT STATUS
2253	000200	TO11CN=000200	; TO-11 TRANSFER DONE
2254	000200	DON11S=000200	; TO-11 NORMAL TERMINATION FLAG STATUS
2255	000100	EBSEL=000100	; E BUFFER SELECT
2256	000100	DON11C=000100	; TO-11 NORMAL TERMINATION FLAG STATUS
2257	000040	NULSTP=000040	; NULL STOP
2258	000040	INTRON=000040	; !! INTERRUPT ENABLE
2259	000020	BPARER=000020	; EBUS PARITY ERROR
2260	000020	EBUSPC=000020	; EBUS PARIT ERROR
2261	000010	RM=000010	; RESTRICTED MODE
2262	000010	INTROF=000010	; DISABLE PDP11 INTERRUPT
2263	000004	DEXDON=000004	; DEPOSIT/EXAMINE DONE
2264	000004	EBUSPS=000004	; EBUS PARITY ERROR SET
2265	000002	TO11ER=000002	; TO-11 BYTE ERROR TERMINATION
2266	000002	ERR11S=000002	; TO-11 ERROR TERMINATION FLAG STATUS
2267	000001	INTSON=000001	; INTERRUPTS ON
2268	000001	ERR11C=000001	; CLEAR TO-11 ERROR TERMINATION FLAG STATUS

DTE20 COMMUNICATION AREA OFFSETS (WORD NAMES)

2273	000000	PIDENT=0	; PROCESSOR IDENTIFICATION WORD
2274	000001	CHNPNT=1	; POINTER TO COMM AREA OF NEXT PROCESSOR (CIRC LIST)
2275	000002	CYCLS=2	; CLOCK CPS COUNT
2276	000003	TOO=3	; TIME OF DAY
2277	000004	DATE=4	; DATE
2278	000005	PSM1=5	; PROCESSOR STATUS WORD1
2279	000006	PSM2=6	; PROCESSOR STATUS WORD2
2280	000007	PSM3=7	; PROCESSOR STATUS WORD3
2281	000010	PSM4=10	; PROCESSOR STATUS WORD4
2282	000011	PSM5=11	; PROCESSOR STATUS WORD5
2283	000012	PSM6=12	; PROCESSOR STATUS WORD6
2284	000013	PSM7=13	; PROCESSOR STATUS WORD7
2285	000014	PSM10=14	; PROCESSOR STATUS WORD10
2286	000015	PSM11=15	; PROCESSOR STATUS WORD11
2287	000016	PSM12=16	; PROCESSOR STATUS WORD12

```

2291 :BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 1
2292 :
2293 :
2294 :
2295 :
2296 :
2297 :
2298 :
2299 :
2300 :
2301 :
2302 :
2303 :
2304 :
2305 :
2306 :
2307 :
2308 :
2309 :
2310 :
2311 :
2312 :
2313 :
2314 :
2315 :
2316 :
2317 :
2318 :
2319 :
2320 :
2321 :
2322 :
2323 :
2324 :

```

000017	PSM13=17	; PROCESSOR STATUS WORD13
000020	FORPRO=20	; FOR PROCESSOR IDENTIFICATION WORD
000021	PROPNT=21	; POINTER TO COMM AREA OF THE PROCESSOR ASSOC WITH THIS BLOCK
000022	STATUS=22	; COMMUNICATION STATUS WORD
000023	QSIZE=23	; QUEUE SIZE WORD
	:CTY0CM=24	:CTY #0 COMMAND WORD
	:CTY0RM=25	:CTY #0 RESPONSE WORD
	:CTY1CM=26	:CTY #1 COMMAND WORD
	:CTY1RM=27	:CTY #1 RESPONSE WORD
	:MISCM=30	: MISCELLANEOUS COMMAND WORD FOR NON-QUEUE PROTOCOL
	:MISRM=31	: MISCELLANEOUS RESPONSE WORD
000032	UNASG1=32	: UNASSIGNED WORD1
000033	UNASG2=33	: UNASSIGNED WORD2
000034	UNASG3=34	: UNASSIGNED WORD3
000035	UNASG4=35	: UNASSIGNED WORD4
000036	UNASG5=36	: UNASSIGNED WORD5
000037	UNASG6=37	: UNASSIGNED WORD6

EPT ADDRESSES AS DEFINED IN BOOTS FOR USE IN THE  
SECONDARY PROTOCOL

000444	DTEFLG=444	; OPERATION COMPLETE FLAG
000450	DTEF11=450	; PDP-10 FROM PDP-11 ARGUMENT
000451	DTECMD=451	; PDP-10 TO PDP-11 COMMAND WORD
000455	DTEMTD=455	; MONITOR TTY OUTPUT COMPLETE FLAG
000456	DTEMTI=456	; MONITOR TTY INPUT FLAG

STATUS DEFINITIONS

000001	TOIT=1	; IN PROGRESS OF PROCESSING QUEUE
000002	TOIP=2	; TO HIM INDIRECT IN PROGRESS

```

2325 ;BM873F - KL10 (PDP-1!) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 2
2326 ;
2327 ;
2328 ;
2329 ;
2330 ;
2331 ; 000040 ROTOR7= 40 ;SAVE R0 TO R7 IN 40 TO 56
2332 ;
2333 ; 000130 DTESAV= 130 ;SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
2334 ; ; IN LOCATIONS 130-156
2335 ;
2336 ; 000012 RETRY= 10. ;DO 10 RETRIES BEFORE HALTING
2337 ;
2338 ; 173000 ROMORG= 173000 ;ROM STARTS AT 773000
2339 ;
2340 ; ESTABLISH ROM ORIGIN
2341 ;
2342 ; 173000 .=ROMORG

```

```

2343 ;BM873F - KL10 (POP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 3
2344 ;
2345 ; EXTERNAL BUTTONS #1, #2, #3
2346 ;
2347 ; BUTTON #1 -- LOAD USING SWITCH REGISTER
2348 ;
2349 005000 010037 ;173000 010037 BUTON1: MOV RO,ROTOR7+0 ;SAVE RO IN LOCATION 40
2350 005002 000040 ;173002 000040
2351 005004 013700 ;173004 013700 MOV SWR,RO ;GET SWITCH REGISTER
2352 005006 177570 ;173006 177570
2353 005010 032700 ;173010 032700 BIT #BIT0,RO ;IS LOW-ORDER BIT SET?
2354 005012 000001 ;173012 000001
2355 005014 001007 ;173014 001007 BNE BUTONX ;YES-- LOOK AT CONTENTS
2356 005016 000557 ;173016 000557 BR REGSAV ;NO-- SAVE R1-R7 IN 42-56, GO TO ADDRESS IN RO (FROM SWR
2357 ;
2358 ; BUTTON #3 -- LOAD BOOT FROM RX11 FLOPPY DISK
2359 ;
2360 005020 005000 ;173020 005000 BUTON3: CLR RO ;SAY LOAD FROM FLOPPY, UNIT 0
2361 005022 000404 ;173022 000404 BR BUTONX ;GO TO COMMON CODE FOR 3 BUTTONS
2362 ;
2363 ; REQUIPED POWER-FAIL VECTOR
2364 ;
2365 005024 173000 ;173024 173000 .WORD ROMORG,PR7
2366 005026 000340 ;173026 000340
2367 ;
2368 ; BUTTON #2 -- LOAD BOOT FROM RPO4 DISK
2369 ;
2370 005030 012700 ;173030 012700 BUTON2: MOV #BIT7,RO ;BIT 7 MEANS LOAD FROM RPO4
2371 005032 000200 ;173032 000200 BR BUTONX ;FALL INTO COMMON CODE
2372 ;
2373 ;
2374 ; RO IS SAVED IN R5 AS THE PARAMETER WORD PASSED TO BOOT
2375 ; AND CONTAINS ONE OF THE FOLLOWING:
2376 ;
2377 ; BIT 0 = 1 IF FROM SWITCH REGISTER
2378 ; BIT 7 = 0 LOAD FROM RX11 FLOPPY DISK
2379 ; BIT 7 = 1 LOAD FROM RPO4 DISK
2380 ; BIT 15 = 1 INDEFINITE RETRY
2381 ;
2382 ; NOTE THAT IF BUTTON #4 IS PRESSED, R5 WILL CONTAIN BIT 0 = 0, BIT 15 = 1
2383 ;
2384 005034 010005 ;173034 010005 BUTONX: MOV RO,R5 ;SAVE PARAMETER FOR BOOT
2385 005036 106300 ;173036 106300 ASLB RO ;LEFT-ALIGN SPEED FIELD IN LOW BYTE
2386 005040 122700 ;173040 122700 CMPB #16*BIT4,RO ;IS SPEED 16 OR 17?
2387 005042 000340 ;173042 000340
2388 005044 101404 ;173044 101404 BLOS 10$ ;YES-- UNIT FIELD IS UNIT # TO BOOT FROM
2389 005046 122700 ;173046 122700 CMPB #3*BIT4,RO ;IS SPEED 0, 1, OR 2?
2390 005050 000060 ;173050 000060
2391 005052 101001 ;173052 101001 BHI 10$ ;YES-- UNIT IS UNIT TO USE
2392 005054 005000 ;173054 005000 CLR RO ;NO-- USE UNIT #0
2393 ;
2394 005056 000300 ;173056 000300 ios: SWAB RO ;GET UNIT # IN LOW BYTE
2395 005060 042700 ;173060 042700 BIC #1C7,RO ;TRIM TO 3 BITS 2, 1, 0
2396 005062 177770 ;173062 177770

```

2397  
2398  
2399  
2400  
2401  
2402

005064 105705 ;173064 105705  
005066 100550 ;173066 100550

;  
;  
;  
;  
;

UNIT # IS IN RD-- CALL PROPER BOOT DEPENDING ON BIT 7

TSTB RS  
BMI RFBOOT  
BR RXBOOT

;WHERE SHOULD WE BOOT FROM?  
;BIT 7 = 1 -- BOOT FROM RPO4 DISK  
;BIT 7 = 0 -- BOOT FROM RX11 FLOPPY DISK



2403  
2404  
2405  
2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416  
2417  
2418  
2419  
2420  
2421  
2422  
2423  
2424  
2425  
2426  
2427  
2428  
2429  
2430  
2431  
2432  
2433  
2434  
2435  
2436  
2437  
2438  
2439  
2440  
2441  
2442  
2443  
2444  
2445  
2446  
2447  
2448

;BMB73F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 4

RX11 FLOPPY DISK BOOTSTRAP AND DUMP ROUTINES

RX11 REGISTER DEFINITIONS

```

; 177170 RXEPA= 177170 ;EXTERNAL PAGE ADDR OF FLOPPY
;
; 000000 RXCS= 0 ;OFFSET FOR CSR
; 100000 RXERR= BIT15 ;ERROR
; 000200 RXTREQ= BIT7 ;TRANSFER REQUEST
; 000040 RXDONE= BITS ;TRANSFER DONE
; 000020 RXUNIT= BIT4 ;UNIT NUMBER 1
; 000016 RXFUNC= BIT3!BIT2!BIT1 ;FUNCTION:
; 000000 RXFILL= 0 ; FILL SILO
; 000002 RXEMPT= 2 ; EMPTY SILO
; 000004 RXWRIT= 4 ; WRITE SECTOR
; 000006 RXREAD= 6 ; READ SECTOR
; 000016 RXRERR= 16 ; READ ERROR REGISTER
; 000001 RXGO= BIT0 ;GO BIT
; 000002 RXDB= 2 ;MULTI-PURPOSE DATA BUFFER REGISTER

```

PARAMETERS

```

; 000001 RXBTRK= 1. ;BOOTSTRAP FROM TRACK 1
; 000001 RXBSCT= 1. ; SECTOR 1 (LOGICAL BLOCK 0)
;
; 000073 RXDTRK= 59. ;DUMP TO TRACK 59
; 000001 RXDSCT= 1. ; SECTOR 1

```

NOTE THAT THE BOOTSTRAP IS WRITTEN IN LOGICAL BLOCK 0 WHICH IS TRACK 1, SECTORS 1, 3, 5, 7. THE DUMP IS WRITTEN STARTING WITH TRACK 59, SECTOR 1, IN EVERY SECTOR (PHYSICAL SECTORS, NOT INTERLEAVED OR SKEWED).

REGISTER USAGE:

```

R0 -- READ OR WRITE FUNCTION. BIT 15 SET IF WRITE
R1 -- ADDRESS OF RXCS
R2 -- CURRENT TRACK (HIGH BYTE) SECTOR (LOW BYTE)
R3 -- TRACK (HIGH BYTE) SECTOR (LOW BYTE)
R4 -- DATA ADDRESS (TO READ OR WRITE)
R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
SP -- RETRY COUNTER

```

```

2449
2450
2451
2452
2453 005070 012703 ;173070
2454 005072 000401 ;173072 012703
2455 005074 005700 ;173074 000401
2456 005076 001402 ;173076 005700
2457 005100 012700 ;173100 001402
2458 005102 000020 ;173102 012700
2459 005104 052700 ;173104 000020
2460 005106 000007 ;173106 052700
2461

```

; HERE TO BOOT FROM RX11 FLOPPY DISK-- UNIT # IN R0  
RXBOOT: MOV #<RXBTRK#BIT8>!<RXBSCT#BIT0>,R3  
TST R0 ;IS THIS UNIT # 0?  
BEQ 10\$ ;YES-- USE 0  
MOV #RXUNIT,R0 ;NO-- USE UNIT # 1  
10\$: BIS #RXREAD+RXGO,R0 ;SET READ FUNCTION IN R0  
; BR RXSTRT ;FALL INTO START-UP

# M05

MAR 1975  
DZBMD.F.P11

MACY11 27(657) 1-OCT-75 09:30 PAGE 55  
ERROR POINTER TABLE

SEQ 0063

```

2462 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 4
2463 ;
2464 ;
2465 ; HERE TO START RX11 ON A TRANSFER, EITHER DUMP OR BOOT
2466 ;
2467 005110 012706 ;173110 012706 RXSTRT: MOV #RETRY,SP ;SET RETRY COUNT
2468 005112 000012 ;173112 000012
2469 005114 012701 ;173114 012701 MOV #RXEPA+RXCS,R1 ;ADDRESS CONTROL STATUS REGISTER FOR RX11
2470 005116 177170 ;173116 177170
2471 ; BR RXRTRY ;FALL THROUGH RETRY CHECK
2472 ;
2473 ; HERE ON ERROR TO RETRY
2474 ;
2475 005120 005705 ;173120 005705 RXRTRY: TST R5 ;INDEFINITE RETRY?
2476 005122 100402 ;173122 100402 BMI 10$ ;YES-- TRY FAITHFULLY
2477 005124 005306 ;173124 005306 DEC SP ;NO-- DECREMENT RETRY COUNT
2478 005126 002475 ;173126 002475 BLT RXEHLT ;GIVE UP IF RUN OUT
2479 ;
2480 005130 000005 ;173130 000005 10$: RESET ;CLEAR THE WORLD
2481 005132 005004 ;173132 005004 CLR R4 ;ALWAYS START TRANSFER AT LOCATION ZERO
2482 005134 010302 ;173134 010302 MOV R3,R2 ;GET START TRACK AND SECTOR
2483 005136 032711 ;173136 032711 20$: BIT #RXDONE,(R1) ;WAIT UNTIL READY FOR FUNCTION
2484 005140 000040 ;173140 000040
2485 005142 001775 ;173142 001775 BEQ 20$ ;NOT YET-- WAIT
2486 005144 005700 ;173144 005700 TST R0 ;THIS WRITE?
2487 005146 100454 ;173146 100454 BMI RXFLSL ;YES-- FILL SILO BEFORE WRITE
2488 ; BR RXPERF ;NO-- JUST DO FIRST READ
2489 ;
2490 ; HERE TO PERFORM READ OR WRITE, AS SPECIFIED IN R0
2491 ;
2492 005150 110011 ;173150 110011 RXPERF: MOVB R0,(R1) ;DO READ OR WRITE
2493 005152 105711 ;173152 105711 10$: TSTB (R1) ;READY?
2494 005154 100376 ;173154 100376 BPL 10$ ;NO-- WAIT
2495 005156 110261 ;173156 110261 MOVB R2,RXDB(R1) ;SET SECTOR #
2496 005160 000002 ;173160 000002
2497 005162 105711 ;173162 105711 20$: TSTB (R1) ;READY FOR TRACK?
2498 005164 100376 ;173164 100376 BPL 20$ ;NO-- WAIT
2499 005166 000302 ;173166 000302 SWAB R2 ;YES-- GET TRACK #
2500 005170 110261 ;173170 110261 MOVB R2,RXDB(R1) ;SET IT
2501 005172 000002 ;173172 000002
2502 005174 000302 ;173174 000302 30$: SWAB R2 ;RESTORE HIGH TRACK, LOW SECTOR
2503 005176 032711 ;173176 032711 BIT #RXERR!RXDONE,(R1) ;DONE OR ERROR?
2504 005200 100040 ;173200 100040
2505 005202 001775 ;173202 001775 BEQ 30$ ;NO-- WAIT
2506 005204 100745 ;173204 100745 BMI RXRTRY ;YES-- ERROR IN FUNCTION

```

```

2507
2508      ; DISK TRANSFER COMPLETE WITH NO ERRORS
2509
2510 005206 005700 ;173206 005700      TST      R0      ;THIS A WRITE?
2511 005210 100421 ;173210 100421      BMI      RXWON   ;YES-- SEE IF DONE WITH DUMP
2512      ;      BR      RXEMSL ;NO-- READ-- EMPTY SILO
2513
2514      ; READ COMPLETED-- EMPTY SILO TO MEMORY
2515
2516 005212 012711 ;173212 012711  RXEMSL: MOV      #RXEMPT+RXGO,(R1) ;START EMPTY
2517 005214 000003 ;173214 000003
2518
2519 005216 132711 ;173216 132711 10$: BITB      #RXTREQ!RXDONE,(R1) ;READY FOR WORD, OR TRANSFER DONE?
2520 005220 000240 ;173220 000240
2521 005222 000402 ;173222 000402      BR      20$      ;BRANCH AROUND VECTOR
2522
2523      ; REQUIRED POWER-FAIL VECTOR
2524
2525 005224 173000 ;173224 173000      .WORD    ROMORG,PR7
2526 005226 000340 ;173226 000340
2527
2528 005230 001772 ;173230 001772 20$: BEQ      10$      ;NOT READY-- WAIT SOME MORE
2529 005232 100003 ;173232 100003      BPL      RXRDON  ;DONE-- GET ANOTHER SECTOR
2530 005234 116124 ;173234 116124      MOVB     RXDB(R1),(R4)+ ;NOT DONE-- GET A BYTE FROM SILO TO MEMORY
2531 005236 000002 ;173236 000002
2532 005240 000766 ;173240 000766      BR      10$      ;WAIT FOR NEXT BYTE
2533
2534      ; SILO EMPTIED-- SEE IF WE ARE DONE WITH BOOTING
2535
2536      ;173242      RXRDON:
2537      $$$=.      173242
2538 005242 122222 ;173242 122222      CMPB     (R2)+,(R2)+
2539 005244 022704 ;173244 022704      CMP      #256.*2,R4      ;HAVE WE READ ENOUGH?
2540 005246 001000 ;173246 001000
2541 005250 101337 ;173250 101337
2542 005252 005007 ;173252 005007  CLRPC: BHI      RXPERF ;NO-- READ SOME MORE
2543      CLR      PC      ;YES-- GO TO LOCATION ZERO
2544
2545      ; WRITE COMPLETED-- SEE IF DONE DUMPING
2546
2547      ;173254      RXWON:
2548      $$$=.      173254
2549 005254 005202 ;173254 005202      INC      R2
2550 005256 122702 ;173256 122702      CMPB     #26.,R2      ;THIS LAST SECTOR ON TRACK?
2551 005260 000032 ;173260 000032
2552 005262 103003 ;173262 103003      BHS      10$      ;NO-- KEEP ON GOING
2553 005264 105002 ;173264 105002      CLRB     R2      ;YES-- CLEAR SECTOR ADDRESS
2554 005266 062702 ;173266 062702      ADD      #BIT8!BIT0,R2 ;BUMP TO NEXT TRACK, SECTOR 1
2555 005270 000401 ;173270 000401
2556 005272 022704 ;173272 022704 10$: CMP      #1024.*28.*2,R4 ;ARE WE DONE WITH 28 K?
2557 005274 160000 ;173274 160000
2558 005276 101516 ;173276 101516      BLOS     HALTO    ;YES-- GO HALT WITH R0= 0 IN DISPLAY
2559      BR      RXFLSL ;NO-- FILL SILO WITH NEXT SECTOR

```

```

2559
2560
2561
2562 005300 012711 :173300 012711
2563 005302 000001 :173302 000001
2564
2565 005304 132711 :173304 132711
2566 005306 000240 :173306 000240
2567 005310 001775 :173310 001775
2568 005312 100316 :173312 100316
2569 005314 112461 :173314 112461
2570 005316 000002 :173316 000002
2571 005320 000771 :173320 000771
2572
2573
2574
2575 005322 012711 :173322 012711
2576 005324 000017 :173324 000017
2577 005326 032711 :173326 032711
2578 005330 000040 :173330 000040
2579 005332 001775 :173332 001775
2580 005334 016100 :173334 016100
2581 005336 000002 :173336 000002
2582 005340 000476 :173340 000476
2583
2584
2585
2586
2587
2588
2589
2590
2591 005342 012703 :173342 012703
2592 005344 035401 :173344 035401
2593
2594 005346 012700 :173346 012700
2595 005350 100005 :173350 100005
2596 005352 005005 :173352 005005
2597 005354 000655 :173354 000655

```

```

: WRITE ANOTHER BLOCK-- FILL SILO
RXFLSL: MOV #RXFILL+RXGO,(R1) ;SET TO FILL SILO
:
IOS: BITB #RXTREQ!RXDONE,(R1) ;READY FOR ANOTHER BYTE?
      BEQ IOS ;NO-- WAIT SOME MORE
      BPL RXPERF ;DONE-- GO PERFORM WRITE
      MOV#B (R4)+,RXDB(R1) ;YES-- STORE ANOTHER BYTE IN SILO
      BR IOS ;WAIT UNTIL READY FOR ANOTHER
:
: HERE ON ERROR AFTER RETRYING -- DISPLAY ERROR REGISTER AND HALT
RXEHLT: MOV #RXRERR+RXGO,(R1) ;DO A READ ERROR REGISTER FUNCTION
:
IOS: BIT #RXDONE,(R1) ;WAIT UNTIL ERROR ASSEMBLED
      BEQ IOS
      MOV RXDB(R1),R0 ;GET ERROR REGISTER
      BR HALTED ;HALT AND DISPLAY ERRORS
:
: START -11 HERE TO DO A DUMP TO RX11 FLOPPY DISK
: NOTE THAT R0-R7 HAVE ALREADY BEEN SAVED IN 40-56
: WHEN BUTTON #1 WAS PUSHED
RXDUMP: MOV #<RXDTRK*BITB>!<RXDSCT*BITD>,R3
:
      MOV #BIT15!RXWRIT+RXGO,R0 ;DO A WRITE
      CLR R5 ;CLEAR INDEFINITE RETRY BIT
      BR RXSTRT ;START DUMP GOING

```

```

2598 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 5
2599 ;
2600 REGISTER SAVE ROUTINE
2601
2602 REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS R0-R7
2603 IN MEMORY AT 40-56 (LOCATION ROTOR7).
2604
2605 CALLING SEQUENCE:
2606     MOV     R0,ROTOR7+0
2607     MOV     #RET,R0
2608     BR     REGSAV
2609 RET: <RETURN HERE>
2610
2611 ALL REGISTERS RESTORED
2612
2613
2614 005356 010037 ;173356 010037 REGSAV: MOV     R0,ROTOR7+16 ;SAVE R0 AS PC IN 56
2615 005360 000056 ;173360 000056
2616 005362 012700 ;173362 012700     MOV     #ROTOR7+16,R0 ;R0 NOW POINTS TO 56
2617 005364 000056 ;173364 000056
2618 005366 010640 ;173366 010640     MOV     SP,-(R0) ;SAVE SP IN 54
2619 005370 010540 ;173370 010540     MOV     R5,-(R0) ;SAVE R5 IN 52
2620 005372 010440 ;173372 010440     MOV     R4,-(R0) ;SAVE R4 IN 50
2621 005374 010340 ;173374 010340     MOV     R3,-(R0) ;SAVE R3 IN 46
2622 005376 010240 ;173376 010240     MOV     R2,-(R0) ;SAVE R2 IN 44
2623 005400 010140 ;173400 010140     MOV     R1,-(R0) ;SAVE R1 IN 42
2624 005402 014000 ;173402 014000     MOV     -(R0),R0 ;RESTORE R0 FROM 40
2625 005404 000177 ;173404 000177     JMP     @ROTOR7+16 ;GO TO SAVED PC
2626 005406 004446 ;173406 004446

```

```

2627      ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 6
2628      ;
2629      ;           RPO4 DISK BOOTSTRAP AND DUMP ROUTINES
2630      ;
2631      ;           RPO4 REGISTER DEFINITIONS
2632      ;
2633      ;           176700 RPEPA= 176700 ;EXTERNAL PAGE ADDRESS OF RPO4 REGISTERS
2634      ;
2635      ;           000020 RPCS1= 0 ;OFFSET FOR CSR #1
2636      ;           040000 RPTRE= BIT14 ;TRANSFER ERROR
2637      ;           020000 RPMCPE= BIT13 ;MASSBUS CONTROL PARITY ERROR
2638      ;           004000 RPDVA= BIT11 ;DRIVE AVAILABLE (TO -11)
2639      ;           000200 RPRDY= BIT7 ;FUNCTION COMPLETE
2640      ;           000076 RPFUNC= BITS!BIT4!BIT3!BIT2!BIT1 ;FUNCTION:
2641      ;           000020 RPPRST= 20 ; READ-IN PRESET
2642      ;           000060 RPMWRIT= 60 ; WRITE DATA
2643      ;           000070 RPREAD= 70 ; READ DATA
2644      ;           000001 RPGO= BIT0 ;GO
2645      ;           000002 RPWC= 2 ;WORD COUNT REGISTER
2646      ;           000006 RPCA= 6 ;TRACK (HIGH BYTE) SECTOR (LOW BYTE)
2647      ;           000010 RPCS2= 10 ;CONTROL AND STATUS REGISTER #2
2648      ;           000007 RPUNIT= BIT2!BIT1!BIT0 ;UNIT #
2649      ;           000012 RPDS= 12 ;DRIVE STATUS REGISTER
2650      ;           100000 RPATA= BIT15 ;ATTENTION ACTIVE
2651      ;           040000 RPERR= BIT14 ;DRIVE ERROR
2652      ;           000034 RPDC= 34 ;DESIRED CYLINDER
2653      ;
2654      ;           ;
2655      ;           ;
2656      ;           ;
2657      ;           ;
2658      ;           ;
2659      ;           ;
2660      ;           ;
2661      ;           ;
2662      ;           ;
2663      ;           ;
2664      ;           ;
2665      ;           ;
2666      ;           ;
2667      ;           ;
2668      ;           ;
2669      ;           ;
2670      ;           ;
2671      ;           ;
2672      ;           ;
2673      ;           ;
2674      ;           ;
2675      ;           ;

```

PARAMETERS

```

000000 RPBCYL= 0. ;BOOT FROM CYLINDER 0
000000 RPBTRK= 0. ; TRACK 0
000000 RPBSC= 0. ; SECTOR 0
000631 RPDCYL= 409. ;DUMP TO CYLINDER 409
000015 RPDRK= 13. ; TRACK 13
000010 RPDSC= 8. ; SECTOR 8

```

REGISTER USAGE:

```

R0 -- FUNCTION CODE (HIGH BYTE) UNIT # (LOW BYTE)
      BIT 15 SET IF WRITE
R1 -- ADDRESS OF RPCS1
R2 -- CYLINDER #
R3 -- TRACK (HIGH BYTE) SECTOR (LOW BYTE)
R4 -- WORD COUNT
R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
SP -- RETRY COUNTER

```

2676				
2677				
2678				
2679			:173410	
2680	005410	005002	:173410	005002
2681	005412	005003	:173412	005003
2682	005414	052700	:173414	052700
2683	005416	034400	:173416	034400

... HERE TO BOOT FROM RP04-- UNIT # IN RO

RPBOOT:

CLR	R2
CLR	R3
BIS	*(RPREAD+RPGO)*BITB,RO ;SET READ HIGH BYTE, UNIT # LOW BYTE



# F06

MAR 1975  
DZBMDP.P11

MACY11 27(657) 1-OCT-75 09:30 PAGE 61  
ERROR POINTER TABLE

SEG 0069

```

2684 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 6
2685
2686 005420 012704 ;173420 012704 MOV #256.,R4 ;READ 256 WORDS TO BOOT
2687 005422 177400 ;173422 177400
2688 ;
2689 ; BR RPSTRT ;START BOOT GOING
2690 ;
2691 ; START RPO4 GOING ON EITHER DUMP OR BOOT
2692 005424 012706 ;173424 012706 RPSTRT: MOV #RETRY,SP ;RETRY RETRY TIMES
2693 005426 000012 ;173426 000012
2694 005430 012701 ;173430 012701 MOV #RPEPA+RPCS1,R1 ;ADDRESS RPCS1 IN R1
2695 005432 176700 ;173432 176700
2696 ;
2697 ; BR RPRTRY ;FALL THROUGH RETRY CODE
2698 ;
2699 ; HERE ON ERROR TO RETRY
2700 005434 005705 ;173434 005705 RPRTRY: TST R5 ;INFINITE RETRY?
2701 005436 100402 ;173436 100402 BMI 10$ ;YES-- TRY AGAIN
2702 005440 005306 ;173440 005306 DEC SP ;RETRY COUNT EXHAUSTED?
2703 005442 002437 ;173442 002437 BLT RPEHLT ;YES-- GIVE UP
2704 ;
2705 005444 000005 ;173444 000005 10$: RESET .ZAP!!
2706 005446 110061 ;173446 110061 MOV# R0,RPCS2(R1) ;SELECT PROPER UNIT #
2707 005450 000010 ;173450 000010
2708 005452 032711 ;173452 032711 BIT #RPDVA,(R1) ;IS DRIVE AVAILAABLE TO US?
2709 005454 004000 ;173454 004000
2710 005456 001766 ;173456 001766 BEQ RPRTRY ;NO-- TRY AGAIN
2711 005460 012711 ;173460 012711 MOV #RPPRST+RPGO,(R1) ;DO 'READ-IN PRESET' FUNCTION
2712 005462 000021 ;173460 000021
2713 005464 010261 ;173464 010261 MOV R2,RPDC(R1) ;SELECT PROPER CYLINDER
2714 005466 000034 ;173466 000034
2715 005470 010361 ;173470 010361 MOV R3,RPDA(R1) ; AND TRACK AND SECTOR
2716 005472 000006 ;173472 000006
2717 005474 010461 ;173474 010461 MOV R4,RPMC(R1) ;SET UP WORD COUNT TO PROPER VALUE
2718 005476 000002 ;173476 000002
2719 ;
2720 ; NOTE THAT IT IS NOT NECESSARY TO SET UP BUS
2721 005500 000300 ;173500 000300 SWAB R0 ; ADDRESS, SINCE IT IS 0 AFTER READ-IN PRESET
2722 005502 110011 ;173502 110011 MOV# R0,(R1) ;GET FUNCTION CODE IN LOW BYTE
2723 005504 000300 ;173504 000300 SWAB R0 ;START FUNCTION GOING
2724 ; ;RESTORE R0
2725 005506 105711 ;173506 105711 20$: TSTB (R1) ;READY?
2726 005510 100376 ;173510 100376 BPL 20$ ;NO-- WAIT UNTIL IT IS
2727 005512 032711 ;173512 032711 BIT #RPTRE!RPMCPE,(R1) ;TRANSFER OR MBC PARITY ERROR?
2728 005514 060000 ;173514 060000
2729 005516 001346 ;173516 001346 BNE RPRTRY ;YES-- ERROR-- TRY AGAIN
2730 005520 032761 ;173520 032761 BIT #RPATA!RPERR,RPCS(R1) ;ATTN OR OTHER ERROR?
2731 005522 140000 ;173522 140000
2732 005524 000012 ;173524 000012
2733 005526 001342 ;173526 001342 BNE RPRTRY ;YES-- ERROR-- TRY AGAIN
2734 005530 005700 ;173530 005700 TST R0 ;READ FUNCTION?
2735 005532 100247 ;173532 100247 BPL CLRPC ;YES-- BOOT-- GO TO LOCATION 0
2736 ; BR HALTO ;NO-- DUMP-- HALT WITH RC= 0 IN DISPLAY

```

```

2737
2738
2739
2740 005534 005000 ;173534 005000 HALTO: CLR R0 ;DISPLAY R0= 0 IF NO ERRORS
2741
2742 005536 000000 ;173536 000000 HALTED: HALT ;DIE
2743 005540 000776 ;173540 000776 BR HALTED ;STAY DEAD
2744
2745
2746
2747 005542 016100 ;173542 016100 RPEHLT: MOV R0S(R1),R0 ;DISPLAY DRIVE STATUS
2748 005544 000012 ;173544 000012
2749 005546 000773 ;173546 000773 BR HALTED ;R.I.P.
2750
2751
2752
2753
2754
2755
2756
2757
2758 005550 012702 ;173550 012702
2759 005552 000631 ;173552 000631
2760 005554 012703 ;173554 012703
2761 005556 006410 ;173556 006410
2762 005560 012700 ;173560 012700
2763 005562 130400 ;173562 130400
2764 005564 012704 ;173564 012704
2765 005566 110000 ;173566 110000
2766 005570 005005 ;173570 005005
2767 005572 000714 ;173572 000714

```

: HERE TO HALT AFTER A DUMP-- DISPLAY R0= 0 IF NO ERRORS  
: HERE ON ERROR FROM RPO4 AFTER RETRYING-- DISPLAY DRIVE STATUS IN R0  
: START -11 HERE TO DUMP TO RPO4 DISK  
: NOTE THAT R0-R7 HAVE ALREADY BEEN SAVED IN 40-56  
: BY PRESSING BUTTON #1.

```

RPDUMP:
MOV #RFD CYL,R2
MOV #<RPOTRK*BIT8>!<RPODCT*BIT0>,R3
MOV #BIT15!<<RPMRIT+RPG0>*BIT8>,R0 ;DO A WRITE, UNIT # 0
MOV #-<1024.*28.>,R4 ;SET TO DUMP 28 K
CLR R5 ;CLEAR INDEFINITE RETRY BIT
BR RPSTRT ;START DUMP GOING

```

```

2768 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 7
2769 ;
2770 ; INTERNAL BUTTON #4 -- DUMP AND BOOTSTRAP THROUGH DTE20
2771 ;
2772 ; DTE20 DEFINITIONS
2773 ;
2774 ; NOTE THAT ALL DTE20 REGISTER DEFINITIONS AND BIT DEFINITIONS
2775 ; ARE IN $DEF IN SYSMAC.SML
2776 ;
2777 ; 000040 DTESIZ= 40 ;EACH DTE OCCUPIES 20 WORDS IN EXTERNAL PAGE
2778 ; 000004 DTEMAX= 4 ;MAX OF 4 DTE'S ON A PDP-11
2779 ;
2780 ;
2781 ; BUTTON #4 -- INITIATED BY '-10 RELOAD -11' BIT
2782 ;
2783 005574 010037 ;173574 010037 BUTON4: MOV R0,ROTOR7+0 ;SAVE R0 IN 40
2784 005576 000040 ;173576 000040
2785 005600 012700 ;173600 012700 MOV #10$,R0 ;SET RETURN ADDRESS IN R0
2786 005602 173606 ;173602 173606
2787 005604 000664 ;173604 000664 BR REGSAV ;SAVE R1-R7
2788 ;
2789 ; REGISTERS SAVED-- LOOK FOR THE DTE20 WHICH PUSHED THE BUTTON
2790 ;
2791 ; THE DTE WHICH PUSHED THE BUTTON SHOULD HAVE THE DOORBELL
2792 ; RINGING AND HAVE THE VALUE 1365 (OCTAL) IN IT'S
2793 ; TO -10 BYTE COUNT TO10BC.
2794 ;
2795 ; NXM (TIME-OUT) TRAP IS USED TO SKIP NON-EXISTANT DTE20'S.
2796 ;
2797 005606 005005 ;173606 005005 10$: CLR R5 ;ADDRESS LOCATION ZERO
2798 005610 012500 ;173610 012500 MOV (R5)+,R0 ;SAVE 0 IN R0
2799 005612 012501 ;173612 012501 MOV (R5)+,R1 ;SAVE 2 IN R1
2800 005614 011502 ;173614 011502 MOV (R5),R2 ;SAVE 4 IN R2
2801 005616 012725 ;173616 012725 MOV #21$, (R5)+ ;SET NXM TRAP ADDRESS IN 4
2802 005620 173634 ;173620 173634
2803 005622 011503 ;173622 011503 MOV (R5),R3 ;SAVE 6 IN R3
2804 005624 012715 ;173624 012715 MOV #PR7,(R5) ;SET PRIORITY FOR NXM TRAP
2805 005626 000340 ;173626 000340
2806 ;
2807 ; LOOP THROUGH ALL DTE'S
2808 ;
2809 005630 012704 ;173630 012704 20$: MOV #DLYCNT-DTESIZ,R4 ;POINT TO DTE # -1'S DELAY COUNT REGISTER
2810 005632 174340 ;173632 174340
2811 ; (WILL BUMP TO # 0)

```

```

2812
2813
2814
2815 005634 012706 ;173634 012706
2816 005636 000004 ;173636 000004
2817
2818 005640 062704 ;173640 062704
2819 005642 000040 ;173642 000040
2820 005644 105704 ;173644 105704
2821
2822
2823 005646 100770 ;173646 100770
2824
2825 005650 032764 ;173650 032764
2826 005652 004000 ;173652 004000
2827 005654 000034 ;173654 000034
2828 005656 001770 ;173656 001770
2829 005660 026417 ;173660 026417
2830 005662 000014 ;173662 000014
2831
2832 005664 001365 ;173664 001365
2833

```

; HERE ON NXM TRAP-- RESET SP AND TRY NEXT DTE  
21\$: MOV #4,SP ;SET SP TO 4, STACK IS LOCATIONS 2 AND 0  
;
22\$: ADD #DTE\$IZ,R4 ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS  
TSTB R4 ;IS THIS THE END OF THE DTE'S?  
; NOTE THAT THE LAST DTE IS AT 774540  
; AND THAT NOW R4= 774600 IF END  
20\$: BMI 20\$ ;YES-- START ALL OVER, UNTIL A DTE  
; SAYS HE PUSHED THE BUTTON  
BIT #TO11DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?  
;
22\$: BEQ 22\$ ;NO-- TRY NEXT DTE  
CMP TO10BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365  
; IN IT'S TO -10 BYTE COUNT?  
22\$: BNE 22\$ ;NO-- TRY ANOTHER DTE  
;

```

2834 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 7
2835 ;
2836 ; WE HAVE FOUND THE DTE WHICH PUSHED THE BUTTON
2837 ;
2838 ; ADDRESS OF DLYCNT REGISTER IS IN R4
2839 ;
2840 005666 010315 ;173666 010315      MOV      R3,(R5)      ;RESTORE LOCATION 6
2841 005670 010245 ;173670 010245      MOV      R2,-(R5)     ; 4
2842 005672 010145 ;173672 010145      MOV      R1,-(R5)     ; 2
2843 005674 010045 ;173674 010045      MOV      R0,-(R5)     ; 0
2844 ;
2845 ; SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
2846 ; IN LOCATIONS 130-156
2847 ;
2848 005676 012700 ;173676 012700      MOV      #DTESAV,R0   ;POINT TO SAVE AREA
2849 005700 000130 ;173700 000130
2850 005702 012420 ;173702 012420 29$: MOV      (R4)+(R0)+ ;SAVE A REGISTER
2851 005704 022700 ;173704 022700      CMP      #T011DT-DLYCNT+DTESAV,R0 ;FINISHED?
2852 005706 000156 ;173706 000156
2853 005710 103374 ;173710 103374      BHS     29$          ;NO-- SAVE SOME MORE
2854 ;
2855 ; R4= T011DT+2
2856 ;
2857 ; SET R1= STATUS REGISTER
2858 ; R4= DIAG2 REGISTER
2859 ;
2860 ; DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
2861 ; LOADED FLAG
2862 ;
2863 ; $$$=
2864 005712 005724 ;173712 005724      TST     (R4)+
2865 005714 010401 ;173714 010401      MOV     R4,R1        ; SO DOES R1
2866 005716 012700 ;173716 012700      MOV     #DRESET,R0   ;SETUP R0 FOR 'DIAGNOSTIC RESET'
2867 005720 000100 ;173720 000100
2868 005722 010021 ;173722 010021      MOV     R0,(R1)+     ;R1 POINTS TO STATUS REGISTER

```

```

2869
2870
2871
2872
2873
2874
2875
2876
2877
2878 005724 005061 ;173724 005061
2879 005726 177744 ;173726 177744
2880 005730 005061 ;173730 005061
2881 005732 177764 ;173732 177764
2882
2883 005734 032711 ;173734 032711
2884 005736 004000 ;173736 004000
2885 005740 001775 ;173740 001775
2886 005742 010014 ;173742 010014
2887
2888
2889
2890
2891
2892
2893 005744 005061 ;173744 005061
2894 005746 177766 ;173746 177766
2895 005750 012761 ;173750 012761
2896 005752 107400 ;173752 107400
2897 005754 177762 ;173754 177762
2898
2899 005756 032711 ;173756 032711
2900 005760 004000 ;173760 004000
2901 005762 001775 ;173762 001775
2902 005764 010014 ;173764 010014
2903 005766 012705 ;173766 012705
2904 005770 100000 ;173770 100000
2905
2906 005772 005007 ;173772 005007
2907
    
```

```

: REGISTERS:
: R0 -- DRESET (DIAGNOSTIC RESET FUNCTION)
: R1 -- STAT (STATUS REGISTER)
: R4 -- DIAG2 (DIAGNOSTIC REGISTER #2, WHERE DRESET IS)
:
: THE -10 WILL NOW START READING -11 MEMORY, AS SOON AS WE SET
: THE TO -10 ADDRESS. WHEN FINISHED, THE -10 WILL RING OUR DOORBELL.
:
: CLR DLYCNT-STAT(R1) ;SET DTE20 FOR MAXIMUM DELAY (ZERO)
:
: CLR T010AD-STAT(R1) ;START DUMPING -11 MEMORY TO -10
:
: STARTING AT LOCATION 0
30$: BIT #T011DB,(R1) ;IS DOORBELL RINGING (TRANSFER COMPLETE)?
:
: BEQ 30$ ;NO-- WAIT FOR DOORBELL
: MOV R0,(R4) ;YES-- CLEAR DOORBELL AND ERROR FLAGS
:
: NOW THE -10 WILL GIVE US A 256 WORD BOOTSTRAP TO BE READ
: INTO -11 MEMORY STARTING AT LOCATION 0. WHEN FINISHED,
: THE -10 WILL RING OUR DOORBELL, AND WE WILL START EXECUTION
: OF THE LOADED CODE AT LOCATION 0.
:
: CLR T011AD-STAT(R1) ;START INPUT TO LOCATION 0
:
: MOV #IFLOP!<<-256.>&7777>,T011BC-STAT(R1) ;256 WORDS, INTERRUPT
:
: BIT #T011DB,(R1) ; -10 WHEN DONE
40$: ;DOORBELL RINGING (LOAD FINISHED)?
:
: BEQ 40$ ;NO-- WAIT UNTIL DONE
: MOV R0,(R4) ;CLEAR DOORBELL RINGING
: MOV #BIT15,R5 ;SET R5: BIT15= 1, BIT0= 0
:
: TO SAY BUTTON #4 PRESSED
: CLR PC ;GO TO LOADED CODE, STARTING AT
: ; LOCATION 0
    
```

```

2908 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 8
2909 ;
2910 ; FILL TO END OF ROM
2911 ;
2912 ;
2913 ;173774 000004 .PRINT <1000>-<.-ROMORG> ;FREE BYTES AT 1000
2914 005774 000000 ;173774 000 .BYTE 0
2915 ;173775 000 .BYTE 0
2916 005776 END.YF: ;173776 000 .BYTE 0
2917 005776 000000 ;173777 000 .BYTE 0
2918 ;
2919 ;
2920 ;
2921 ;
2922 ;174000 000001 PASS2: .END
2923 ;
2924 ;
2925 006000 000127 MAP.YG: .BLKW 127
2926 006256 000001 END.YG: .BLKW 1
2927 006260 000177 MAP.Y.: .BLKW 127.
2928 006656 000001 END.Y.: .BLKW 1
2929 006660 000177 MAP.YX: .BLKW 127.
2930 007256 000001 END.YX: .BLKW 1

```

```

2931
2932
2933
2934
2935 007260
2936 007260 012706 001100
2937 007264 005026
2938 007266 022706 001136
2939 007272 001374
2940 007274 012706 001100
2941 007300 012737 015524 000020
2942 007306 012737 000340 000022
2943 007314 012737 015620 000030
2944 007322 012737 000340 000032
2945 007330 012737 016150 000034
2946 007336 012737 000340 000036
2947 007344 012767 007344 171534
2948 007352 005067 001056
2949 007356 005067 001042
2950 007362 012706 001100
2951 007366 005067 171624
2952 007372 005037 000000
2953 007376 012767 007362 005270
2954 007404 012737 000006 000004
2955 007412 005037 000006
2956 007416 005067 004546
2957 007422 005737 000042
2958 007426 001002
2959 007430 000167 000412
2960 007434 013746 000004
2961 007440 012737 010440 000004
2962 007446 005737 173000
2963 007452 000240
2964 007454 012637 000004
2965 007460 026737 171714 173000
2966 007466 001034
2967 007470 013746 000004
2968 007474 012737 007516 000004
2969 007502 005737 173400
2970 007506 000240
2971 007510 012637 000004
2972 007514 000421
2973 007516 022626
2974 007520 012637 000004
2975 007524 012767 001400 000672
2976 007532 012767 001776 000666
2977 007540 012767 173376 000670
2978 007546 012767 000101 005230
2979 007554 000167 001070
2980 007560
2981 007560 026737 172214 173000
2982 007566 001016
2983 007570 012767 002000 000626
2984 007576 012767 002776 000622

```

```

*****
INITIALIZATION AND START UP OF PROGRAM.
*****
RESTRT:
MOV   #SCMTAG,R6      ;FIRST LOCATION TO BE CLEARED
CLR   (R6)+           ;CLEAR MEMORY LOCATION
CMP   #STKS,R6       ;DONE?
BNE   .-6             ;LOOP BACK IF NO
MOV   #STACK,SP      ;SETUP THE STACK POINTER
MOV   #SCOPE,@#IOTVEC ;IOT VECTOR FOR SCOPE ROUTINE
MOV   #340,@#IOTVEC+2 ;LEVEL 7
MOV   #ERROR,@#EMTVEC ;EMT VECTOR FOR ERROR ROUTINE
MOV   #340,@#EMTVEC+2 ;LEVEL 7
MOV   #STRAP,@#TRAPVEC ;TRAP VECTOR FOR TRAP CALLS
MOV   #340,@#TRAPVEC+2 ;LEVEL 7
MOV   #,$LPADR       ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
CLR   INITFG         ;INITIALIZE TO ASK WHICH TYPE
CLR   TABLE        ;INITIALIZE TO ASK WHICH TYPE
START:
MOV   #STACK,SP     ;SET THE STACK POINTER
CLR   LSTERR        ;CLEAR ERROR FLG REPORT
CLR   @#0           ;SET FOR UNEXPECTED TRAP TO ADD 0
MOV   #START,PRG.NO ;GET READY FOR PWR FAIL BEFORE FIRST TEST.
MOV   #6,@#4        ;SET TIME OUT TRAP VECTOR
CLR   @#6          ;SET TIME OUT STATUS TO 0
CLR   FLAG4        ;CLEAR TEST 4 INITIAL FLAG
TST   @#42         ;AM I RUNNING UNDER ACT-11??
BNE   .+6          ;BR IF *WE ARE* UNDER ACT-11!!
JMP   CONT         ;JUMP IF NOT ACT-11
MOV   @#4,-(SP)    ;SAV TRAP POINTER
MOV   #NOROM,@#4  ;PUT IN A NEW ONE
TST   @#173000    ;TRY TO READ THE ROM
NOP                    ;WAIT FOR POSSIBLE TRAP
MOV   (SP)+,@#4   ;IF NO TRAP RESTORE POINTER
CMP   MAP.YA,@#173000 ;DOES 1ST WORD COMPARE?
BNE   64$         ;CHECK NEXT MAP
MOV   @#4,-(SP)   ;SAVE LOC 4
MOV   #65$,@#4   ;SET FOR TIMEOUT
TST   @#173400   ;READ FROM 173400
NOP                    ;IF NO TIMEOUT, NOT YA
MOV   (SP)+,@#4  ;RESTORE LOC 4
BR   64$
65$:
CMP   (SP)+,(SP)+ ;ADJUST STACK
MOV   (SP)+,@#4  ;RESTORE LOC 4
MOV   #MAP.YA, TABLE ;1ST MAP ADDR
MOV   #END.YA,ALLEND ;LAST MAP ADDR
MOV   #173376,LASTA ;LAST ROM ADDR
MOV   #000101,VERSON ;SET ROM TYPE
JMP   PRG1       ;START TEST 1
64$:
CMP   MAP.YB,@#173000 ;DOES 1ST WORD COMPARE?
BNE   69$         ;CHECK NEXT MAP
MOV   #MAP.YB, TABLE ;1ST MAP ADDR
MOV   #END.YB,ALLEND ;LAST MAP ADDR

```



2985	007604	012767	173776	000624	MOV	#173776, LASTA	; LAST ROM ADDR
2986	007612	012767	000102	005164	MOV	#000102, VERSON	; SET ROM TYPE
2987	007620	000167	001024		JMP	PRG1	; START TEST 1
2988	007624						
2989	007624	026737	173150	173000	69\$: CMP	MAP.YC, @#173000	; DOES 1ST WORD COMPARE?
2990	007632	001036			BNE	74\$	; CHECK NEXT MAP
2991	007634	013746	000004		MOV	@#4, -(SP)	; SAVE LOC 4
2992	007640	012737	007664	000004	MOV	#76\$, @#4	; SET FOR TIMEOUT
2993	007646	026737	173526	173400	CMP	MAP.YC+400, @#173400	; IS IT YC?
2994	007654	001004			BNE	77\$	; BR IF NOT YC
2995	007656	012637	000004		MOV	(SP)+, @#4	; RESTORE LOC 4
2996	007662	000404			BR	78\$	; YES IT IS A YC
2997	007664	022626			76\$: CMP	(SP)+, (SP)+	; ADJUST STACK
2998	007666	012637	000004		77\$: MOV	(SP)+, @#4	; RESTORE LOC 4
2999	007672	000416			BR	74\$	; CHECK NEXT MAP
3000	007674				78\$:		
3001	007674	012767	003000	000522	MOV	#MAP.YC, TABLE	; 1ST MAP ADDR
3002	007702	012767	003776	000516	MOV	#END.YC, ALLEND	; LAST MAP ADDR
3003	007710	012767	173776	000520	MOV	#173776, LASTA	; LAST ROM ADDR
3004	007716	012767	000103	005060	MOV	#000103, VERSON	; SET ROM TYPE
3005	007724	000167	000720		JMP	PRG1	; START TEST 1
3006	007730				74\$:		
3007	007730	026737	174044	173000	CMP	MAP.YD, @#173000	; DOES 1ST WORD COMPARE?
3008	007736	001016			BNE	79\$	; CHECK NEXT MAP
3009	007740	012767	004000	000456	MOV	#MAP.YD, TABLE	; 1ST MAP ADDR
3010	007746	012767	004776	000452	MOV	#END.YD, ALLEND	; LAST MAP ADDR
3011	007754	012767	173776	000454	MOV	#173776, LASTA	; LAST ROM ADDR
3012	007762	012767	000104	005014	MOV	#000104, VERSON	; SET ROM TYPE
3013	007770	000167	000654		JMP	PRG1	; START TEST 1
3014	007774				79\$:		
3015	007774	026737	175000	173000	CMP	MAP.YF, @#173000	; DOES 1ST WORD COMPARE?
3016	010002	001016			BNE	84\$	; CHECK NEXT MAP
3017	010004	012767	005000	000412	MOV	#MAP.YF, TABLE	; 1ST MAP ADDR
3018	010012	012767	005776	000406	MOV	#END.YF, ALLEND	; LAST MAP ADDR
3019	010020	012767	173776	000410	MOV	#173776, LASTA	; LAST ROM ADDR
3020	010026	012767	000106	004750	MOV	#000106, VERSON	; SET ROM TYPE
3021	010034	000167	000610		JMP	PRG1	; START TEST 1
3022	010040				84\$:		
3023	010040	104400	010540		TYPE	, NMATCH	; NOT BMB73YA OR B OR C OR D
3024	010044	000000			HALT		
3025	010046	005767	000362		CONT: TST	INITFG	; IS THIS THE FIRST TIME START UP?
3026	010052	001127			BNE	3\$	; BR IF NOT FIRST TIME HERE.
3027	010054	005167	000354		COM	INITFG	; SET THE FLAG
3028	010060	104400	011242		2\$: TYPE	, BMB73X	; TYPE THE QUESTION.
3029	010064	104412			RDLIN		
3030	010066	012602			MOV	(SP)+, R2	
3031	010070	011202			MOV	(R2), R2	; PLACE CHARACTER INTO R2.
3032	010072	022702	000052		CMP	#52, R2	; WAS * HIT??
3033	010076	001011			BNE	64\$	; BR IF NO
3034	010100	012767	006260	000316	MOV	#MAP.Y., TABLE	; SET FOR START OF TABLE
3035	010106	012767	006656	000312	MOV	#END.Y., ALLEND	; SET END OF TABLE
3036	010114	012767	173376	000314	MOV	#173376, LASTA	; SET LAST ROM ADDR
3037	010122				64\$:		
3038	010122	022702	000101		CMP	#101, R2	; WAS A HIT??

```

3039 010126 000111 BNE 65$ ;BR IF NO
3040 010130 012767 001400 000266 MOV #MAP.YA, TABLE ;SET FOR START OF TABLE
3041 010136 012767 001776 000262 MOV #END.YA, ALLEND ;SET END OF TABLE
3042 010144 012767 173376 000264 MOV #173376, LASTA ;SET LAST ROM ADDR
3043 010152 65$: CMP #102, R2 ;WAS B HIT??
3044 010152 022702 000102 BNE 65$ ;BR IF NO
3045 010156 001011 65$
3046 010160 012767 002000 000236 MOV #MAP.YB, TABLE ;SET FOR START OF TABLE
3047 010166 012767 002776 000232 MOV #END.YB, ALLEND ;SET END OF TABLE
3048 010174 012767 173776 000234 MOV #173776, LASTA ;SET LAST ROM ADDR
3049 010202 66$: CMP #103, R2 ;WAS C HIT??
3050 010202 022702 000103 BNE 67$ ;BR IF NO
3051 010206 001011 67$
3052 010210 012767 003000 000206 MOV #MAP.YC, TABLE ;SET FOR START OF TABLE
3053 010216 012767 003776 000202 MOV #END.YC, ALLEND ;SET END OF TABLE
3054 010224 012767 173776 000204 MOV #173776, LASTA ;SET LAST ROM ADDR
3055 010232 67$: CMP #104, R2 ;WAS D HIT??
3056 010232 022702 000104 BNE 68$ ;BR IF NO
3057 010236 001011 68$
3058 010240 012767 004000 000156 MOV #MAP.YD, TABLE ;SET FOR START OF TABLE
3059 010246 012767 004776 000152 MOV #END.YD, ALLEND ;SET END OF TABLE
3060 010254 012767 173776 000154 MOV #173776, LASTA ;SET LAST ROM ADDR
3061 010262 68$: CMP #106, R2 ;WAS F HIT??
3062 010262 022702 000106 BNE 69$ ;BR IF NO
3063 010266 001011 69$
3064 010270 012767 005000 000126 MOV #MAP.YF, TABLE ;SET FOR START OF TABLE
3065 010276 012767 005776 000122 MOV #END.YF, ALLEND ;SET END OF TABLE
3066 010304 012767 173776 000124 MOV #173776, LASTA ;SET LAST ROM ADDR
3067 010312 69$: MOV R2, VERSION ;STORE VERSION TYPE.
3068 010312 010267 004466 TS# TABLE ;HAS A MAP BEEN SELECTED?
3069 010316 005767 000102 BNE 3$ ;BR IF OK...
3070 010322 001003 3$
3071 010324 104400 011317 TYPE #M.ERR ;TYPE ERROR
3072 010330 000653 BR 2$ ;GO AND GET CORRECT MAP.
3073 010332 104400 014312 3$: TYPE #MSG3 ;TYPE MESSAGE FOR TEST NUMBER
3074 010336 104412 X.X.: RDLIN
3075 010340 012602 MOV (SP)+, R2
3076 010342 011203 MOV (R2), R3 ;MOV THE CHAR TO R3
3077 010344 022703 000061 2$: CMP #61, R3 ;WAS 1 HIT??
3078 010350 001002 BNE 4$ ;BR IF NO
3079 010352 000167 000272 JMP PRG1 ;GOTO PRG 1
3080 010356 022703 000062 4$: CMP #62, R3 ;WAS 2 HIT??
3081 010362 001002 BNE 5$ ;BR IF NO
3082 010364 000167 001000 JMP PRG2 ;GOTO PRG 2
3083 010370 022703 000063 5$: CMP #63, R3 ;WAS 3 HIT??
3084 010374 001002 BNE 6$ ;BR IF NO
3085 010376 000167 001766 JMP PRG3 ;GOTO PRG 3
3086 010402 022703 000064 6$: CMP #64, R3 ;WAS 4 HIT??
3087 010406 001002 3$ ;BR IF NO
3088 010410 000167 002760 JMP PRG4 ;GOTO PRG 4
3089 010414 104400 014544 3$: TYPE #M.QM ;NEITHER 1 OR 2 OR 3 OR 4 WAS HIT
3090 010420 000167 176634 JMP RESTART ;TYPE "??" GO TO THE BEGINING.
3091 010424 000000
3092 010426 000000
TABLE: 0
ALLEND: 0

```

3093	010430	006660			EXTMAP:	MAP.YX		
3094	010432	007256			EXTEND:	END.YX		
3095	010434	000000			INITFG:	0		
3096	010436	000000			LASTA:	0		
3097	010440	104400	010450		NOROM:	TYPE	,NOROMS ;TYPE CAN'T FIND A RESPONSE	
3098	010444	000000				HALT	;NO LOADER INSTALLED?	
3099	010446	000776				BR		
3100	010450	005015	051124	050101	NOROMS:	.ASCII	<15><12>/TRAP TO 4 ON 1ST READ OF 173000/	
	010511	015	044412	020123		.ASCIZ	<15><12>/IS LOADER INSTALLED?/	
	010540	005015	040503	023516	NMATCH:	.ASCII	<15><12>/CAN'T IDENTIFY LOADER AS YA,YB,YC,YD OR YF AFTER/	
	010622	005015	046503	020120		.ASCIZ	<15><12>/CMP WITH LOC 173000/	
						.EVEN		

```

3101 :PROGRAM 1
3102 :THE PURPOSE OF PROGRAM 1 IS TO READ THE ROM AND
3103 :VERIFY THAT THE DATA IS CORRECT. ALL ADDRESSES
3104 :ARE READ, EXCEPT THE TRAP VECTOR, FIVE TIMES.
3105
3106 :THE SECOND PART OF THIS TEST VERIFIES THAT TRYING
3107 :TO WRITE THE ROM RESULTS IN A TIME OUT TRAP.
3108 :ALL ADDRESS ARE WRITTEN WITH A -1
3109 :AND ARE EXPECTED TO TRAP.
3110
3111 010650 012767 010650 004016 PRG1:  MOV      #PRG1,PRG.NO      ;SET FOR PWR FAIL
3112 010656 012767 000500 170334      MOV      #500,COUNT      ;DO THIS TEST 500(8) TIMES.
3113 010664 012737 014676 000004 PRG.1:  MOV      #NO.TRAP,2#4    ;SET FOR UNEXPECTED TRAP.
3114 010672 012700 173000      MOV      #173000,R0      ;SET BEGGING ADDRESS
3115 010676 012767 010722 170202      MOV      #25,$LPADR      ;IF SW14=1; GOTO 25 WHEN SCOPE IS HIT
3116 010704 016704 177514      MOV      TABLE,R4      ;SET START OF MAP
3117 010710 016767 177522 000322      MOV      LASTA, LAST    ;SET LAST ADDRESS
3118 010716 012703 000005      15:    MOV      #5, R3        ;DO EACH ADDRESS 5 TIMES.
3119 010722 022700 173024      25:    CMP      #173024, R0    ;DON'T DO THE VECTOR ADD.
3120 010726 001001      BNE     20$             ;BR IF NOT THE VECTOR ADD.
3121 010730 022024      CMP      (R0)+, (R4)+    ;UPDATE TO NEXT ADDRESS
3122 010732 022700 173224      20$:   CMP      #173224, R0    ;DON'T DO THE TRAP VECTORS
3123 010736 001001      BNE     21$             ;NO THIS ISN'T A TRAP VECTOR.
3124 010740 022024      CMP      (R0)+, (R4)+    ;UPDATE THE POINTERS..
3125 010742 010467 170156      21$:   MOV      R4, $GDDAT
3126 010746 010067 170154      MOV      R0, $BDDAT
3127 010752 011067 170250      MOV      (R0), TEMP4
3128 010756 011467 170242      MOV      (R4), TEMP3
3129 010762 026767 170236 170236      CMP      TEMP3, TEMP4
3130 010770 001401      BEQ     22$
3131 010772 104001      ERROR  1
3132 010774 032767 004000 166566 22$:   BIT      #BIT11, SWR
3133 011002 001002      BNE     23$
3134 011004 005307      DEC     R3
3135 011006 001345      BNE     25$
3136
3137 011010 026700 000224      23$:   CMP      LAST, R0
3138 011014 001403      BEQ     10$
3139 011016 000004      SCOPE
3140 011020 022024      CMP      (R0)+, (R4)+
3141 011022 000735      BR     15$
3142
3143 011024 032767 000001 166536 10$:   BIT      #BIT0, SWR
3144 011032 001413      BEQ     35$
3145 011034 022767 173776 000176      CMP      #173776, LAST
3146 011042 001407      BEQ     35$
3147 011044 012767 173776 000166      MOV      #173776, LAST
3148 011052 016704 177352      MOV      EXTMAP, R4
3149 011056 005720      TST     (R0)+
3150 011060 000716      BR     15$

```

```

;READ THE ADDRESS
;READ THE SOFTWARE ADDRESS
;BR IF GOOD
;INCORRECT COMPARISON.
;QUICK PASS.?
;BR IF YES
;HAS THAT ADD BEEN READ 5 TIMES?
;BR IF NOT 5 TIMES
;WAS LAST ADDRESS CHECKED?
;BR IF YES
;LOCK ON THIS ADDRESS IF SW14=1
;UPDATE THE POINTERS.
;CONTINUE THE TEST.
;EXTENDED WORD TO BE CHECKED?
;BR IF NO CHECKING.
;IS ALL THE TEST DONE?
;BR IF YES.
;SET LAST ADDRESS.
;SET EXTENDED MAP.
;POP POINTER
;GO DO THE TEST.

```

```

3151                                     ;TEST THAT WRITTING ROM RESULTS IN A TIME OUT
3152                                     ;TRAP.
3153
3154 011062 012767 011106 170016 3$: MOV #5$,SLPADR ;IF SW14=1 GOTO 5$ WHEN SCOPE IS HIT
3155 011070 012700 173000 MOV #173000,RO ;SET RO WITH BASE ADDRESS OF ROM
3156 011074 012737 011142 000004 MOV #6$,R#4 ;SET FOR TIME OUT TRAP
3157 011102 012703 000005 4$: MOV #5$,R3 ;DO EACH ADD 5 TIMES
3158 011106 022700 173024 5$: CMP #173024,RO ;CHECK FOR A TRAP VECTOR
3159 011112 001001 BNE 24$ ;BR IF NOT VECTOR
3160 011114 005720 TST (RO)+ ;UPDATE THE REGISTER POINTER
3161 011116 022700 173224 24$: CMP #173224,RO ;CHECK FOR THE OTHER VECTOR
3162 011122 001001 BNE 25$ ;BR IF NOT THE VECTOR
3163 011124 005720 TST (R3)+ ;UPDATE THE POINTER
3164 011126 012710 177777 25$: MOV #-1,(RO) ;WRITE ROM WITH A -1
3165 011132 000240 NOP ;WAIT ONE INSTR. TIME
3166 011134 010067 170066 MOV RO,TEMP4
3167 011140 104002 ERROR 2 ;WRITING ROM DIDN'T TIME OUT.
3168 011142 012706 001100 6$: MOV #STACK,SP ;RESTORE STACK
3169 011146 032767 004000 166414 BIT #BIT11,SWR ;QUICK PASS?
3170 011154 001002 BNE 30$
3171 011156 005303 DEC R3 ;DO EACH ADD 5 TIMES
3172 011160 001352 BNE 5$ ;NOT DONE WITH THIS ONE YET.
3173
3174 011162 032767 000001 166400 30$: BIT #BIT0,SWR ;EXTENDED 128. WORDS TO BE CHECKED?
3175 011170 001404 BEQ 31$ ;BR IF NO
3176 011172 022700 173776 CMP #173776,RO ;HAVE ALL 256. WORDS BEEN CHECKED?
3177 011176 001407 BEQ 7$ ;BR IF ALL DONE
3178 011200 000403 BR 32$ ;KEEP GOING
3179 011202 026700 177230 31$: CMP LASTA,RO ;ALL DONE??
3180 011206 001403 BEQ 7$ ;HAVE ALL 128. WORDS DONE?
3181 011210 000004 32$: SCOPE ;CHECK SW14 FOR FREEZE!!
3182 011212 005720 TST (RO)+ ;UPDATE TO NEXT ADDRESS
3183 011214 000732 BR 4$ ;GO DO IT AGAIN
3184 011216 005367 167776 7$: DEC ICOUNT ;ITERATION COUNT DONE?
3185 011222 001004 BNE 8$ ;BR IF NOT DONE.
3186 011224 004767 003506 JSR PC,EOP ;TYPE END MESSAGE
3187 011230 000167 177414 JMP PRG1 ;GO DO IT AGAIN.
3188 011234 000167 177424 8$: JMP PRG.1 ;GO RESTART.
3189 011240 000000 LAST: 0
3190
3191 011242 005015 040515 047111 BMB73X: .ASCII <15><12>/MAINDEC-11-DZBMOF/
011265 015 042012 053105 .ASCII <15><12>/DEVICE VERSION/
011305 015 041012 034115 .ASCIZ <15><12>/BMB73-Y/
011317 015 025012 040454 BM.ERR: .ASCIZ <15><12>/A,B,C,D,F ONLY./
011343 040 020040 042526 VERS: .ASCIZ / VERSION: BMB73-Y/
011370 .EVEN

```

```

3192 ;PROGRAM 2
3193 ;BLIND READ FROM ROM.
3194 ;THIS PROGRAM WILL DUMP THE CONTENTS OF THE ROM OUT
3195 ;PERFORMING NO CHECKING AT ALL.
3196 ;PLEASE NOTE: NO CHECKING IS DONE.
3197
3198 011370 012767 011370 003276 PRG2:  MOV    #PRG2,PRG,NO    ;SET FOR POWER FAIL
3199 011376 012737 014676 000004      MOV    #NO,TRAP,2#4    ;SET FOR UNEXPECTED TRAP TO 4
3200 011404 016767 177026 177626      MOV    LASTA, LAST
3201 011412 062767 000002 177620      ADD    #2, LAST
3202 011420 012700 173000 21$:      MOV    #173000,RO     ;SET RO WITH THE STARTING ROM ADD.
3203 011424 016703 176774      MOV    TABLE,R3     ;SET POINTER.
3204 011430 104400 011750      TYPE   ,DH.2         ;TYPE MESSAGE
3205 011434 104400 012050      TYPE   ,DH.2B        ;TYPE THE HEADER
3206 011440 012767 000007 167554 1$:      MOV    #7,TEMPS      ;SET COUNTER
3207 011446 011001      MOV    (RO),R1       ;READ THE ROM
3208 011450 010067 167550      MOV    RO,TEMP3      ;STORE RO
3209 011454 010167 167546      MOV    R1,TEMP4      ;STORE R1
3210 011460 022767 006260 176736      CMP    #MAP.Y.,TABLE ;IF BM873.Y* SELECTED; FILL TABLE
3211 011466 001001      BNE    22$           ;BR IF NOT BM873.Y*
3212 011470 011023      MOV    (RO),(R3)+    ;FILL THE TABLE..
3213 011472 005720 22$:      TST   (RO)+         ;POP THE POINTER
3214 011474 104400 014560      TYPE   ,MCRLF
3215
3216 011500 016746 167520      MOV    TEMP3,-(SP)
3217 011504 104402      TYPOC
3218 011506 104400 014551      TYPE   ,MSPACE      ;TYPE THREE SPACES.
3219
3220
3221 011512 016746 167510      MOV    TEMP4,-(SF)
3222 011516 104402      TYPOC
3223 011520 011001 7$:      MOV    (RO),R1       ;STORE ROM DATA
3224 011522 010067 167476      MOV    RO,TEMP3      ;STORE ROM ADDRESS
3225 011526 010167 167474      MOV    R1,TEMP4      ;PREPARE DATA FOR TYPE OUT
3226 011532 022767 006260 176664      CMP    #MAP.Y.,TABLE ;IS BM873.Y* SELECTED?
3227 011540 001001      BNE    23$           ;BR IF NO.
3228 011542 011023      MOV    (RO),(R3)+    ;FILL THE DATA TABLE
3229 011544 005720 23$:      TST   (RO)+         ;POP THE POINTER
3230
3231 011546 104400 014551      TYPE   ,MSPACE
3232
3233 011552 016746 167450      MOV    TEMP4,-(SP)
3234 011556 104402      TYPOC
3235
3236 011560 026700 177454      CMP    LAST,RO ;HAS THE HIGHEST LIMIT BEEN HIT?
3237 011564 001404 2$:      BEQ    2$           ;BR IF ALL DONE.
3238 011566 005367 167430      DEC    TEMPS        ;DECREASE COUNTER
3239 011572 001352 7$:      BNE    7$           ;BR IF NOT 0; KEEP GOING
3240 011574 000721 1$:      BR     1$           ;GO TYPE ADDRESS NOW
3241
3242 011576 032767 000001 165764 2$:      BIT    #BIT0,SWR     ;IS THE EXTENDED 128. WORDS TO BE CHECKED??
3243 011604 001455 3$:      BEQ    3$           ;BR IF NO.
3244 011606 012700 173400      MOV    #173400,RO    ;RESET POINTER OF ROM
3245 011612 016703 176612      MOV    EXTMAP,R3     ;SET SOFTWARE MAP POINTER

```

```

3246 011616 104400 012162          TYPE      ,DH.2A          ;TYPE NEW HEADER
3247 011622 104400 012050          TYPE      ,DH.2B          ;TYPE ADDRESS AND +XX
3248 011626 012767 000007 167366 6$: MOV      #7,TEMP5      ;SET TYPE OUT COUNTER
3249 011634 011001                MOV      (R0),R1      ;READ THE ROM
3250 011636 010067 167362          MOV      R0,TEMP3     ;STORE R0
3251 011642 010167 167360          MOV      R1,TEMP4     ;STORE R1
3252 011646 012023                MOV      (R0)+,(R3)+  ;STORE THE DATA IN SOFTWARE MAP
3253 011650 104400 014560          TYPE      MCRLF
3254 011654 016746 167344          MOV      TEMP3,-(SP)
3255 011660 104402                TYPOC
3256
3257 011662 104400 014551          TYPE      MSPACE
3258 011666 016746 167334          MOV      TEMP4,-(SP)
3259 011672 104402                TYPOC
3260
3261 011674 011001                8$: MOV      (R0),R1      ;SAVE THE ROM DATA
3262 011676 010067 167322          MOV      R0,TEMP3     ;SAVE THE ROM ADDRESS
3263 011702 010167 167320          MOV      R1,TEMP4     ;SET DATA FOR TYPE OUT
3264
3265 011706 104400 014551          TYPE      ,MSPACE
3266
3267 011712 016746 167310          MOV      TEMP4,-(SP)
3268 011716 104402                TYPOC
3269
3270 011720 012023                MOV      (R0)+,(R3)+  ;STORE THE DATA IN SOFTWARE TABLE
3271 011722 022700 174000          CMP      #174000,R0   ;HAS THE HIGHEST LIMIT BEEN HIT?
3272 011726 001404                BEQ      3$           ;BR IF ALL DONE.
3273 011730 005367 167266          DEC      TEMPS        ;DEC TABLE COUNTER
3274 011734 001357                BNE      8$           ;BR TO JUST TYPE DATA
3275 011736 000733                BR       6$           ;BR TO TYPE ADDRESS
3276 011740 005000                3$: CLR      R0        ;CLEAR DATA LIGHTS
3277 011742 000000                HALT
3278 011744 000167 177420          JMP      PRG2        ;HIT CONTINUE TO PROCEED.
3279 011750 006414 005012 016412 DH.2:  .ASCII <14><15><12><12><12><35><37><177><177><177>/BLIND READ OF ROM/
012003 015 006412 077577          .ASCIZ <15><12><15><177><177>/NOTE: NO CHECKING IS PERFORMED./
012050 005015 040412 042104 DH.2B: .ASCII <15><12><12>/ADDRESS ADD+00 ADD+02 ADD+04/
012111 040 040440 042104          .ASCIZ / ADD+06 ADD+10 ADD+12 ADD+14 ADD+16/
012162 005015 042412 052130 DH.2A: .ASCII <15><12><12>/EXTENDED 128. WORD ROM DUMP./
012221 015 041412 047117          .ASCII <15><12>/CONTENTS DUMPED IS PLACED IN THE SOFTWARE/
012274 005015 040515 027120          .ASCII <15><12>/MAP. VISUAL INSPECTION OF DATA IS/
012337 015 054412 052517          .ASCIZ <15><12>/YOUR RESPONSIBILITY!!/
012370                .EVEN

```

```

3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295 012370 012767 012370 002276 PRG3:  MOV    #PRG3,PRG.NO    ;SET FOR POWER FAIL
3296 012376 016701 176022          MOV    TABLE,R1      ;DEFAULT STARTING ADDRESS TO MAP
3297 012402 010167 000764          MOV    R1,ADDRESS     ;SAVE THE SOFTWARE ADDRESS
3298 012406 104400 014453          XHOLD: TYPE    ,MASTER ;TYPE AN "*"
3299 012412 104412
3300 012414 012602          MOV    (SP)+,R2
3301 012416 011202          MOV    (R2),R2
3302 012420 022702 000114          CMP    #114,R2        ;WAS AN "L" (LIST) HIT?
3303 012424 001464          BEQ    SRV.L
3304
3305 012426 022702 000104          1$:    CMP    #104,R2   ;WAS A "D" (DATA) HIT?
3306 012432 001413          BEQ    SRV.D
3307 012434 022702 000122          CMP    #122,R2        ;WAS AN "R" (RUN) HIT?
3308 012440 001002          BNE    10$
3309 012442 000167 000342          JMP    SRV.R
3310 012446 022702 000101          10$:   CMP    #101,R2        ;WAS AN "A" (ADDRESS) HIT?
3311 012452 001444          BEQ    SRV.A
3312 012454 104400 014544          TYPE    ,M.QM        ;TYPE A "?"
3313 012460 000752          BR     XHOLD          ;NEITHER A "L","P","D","R","A",OR CR WAS HIT.
3314
3315 012462 016767 000704 166536 SRV.D:  MOV    ADDRESS,TEMP4 ;RESET ADDRESS POINTER.
3316 012470 104400 014560          TYPE    MCRLF
3317 012474 016746 166526          MOV    TEMP4,-(SP)
3318 012500 016701 166522          MOV    TEMP4,R1
3319 012504 104402          TYPOC
3320
3321 012506 104400 014551          TYPE    ,MSPACE
3322
3323 012512 104414          RDOCT
3324 012514 012611          MOV    (SP)+,(R1)    ;STORE DATA
3325
3326 012516 005721          TST    (R1)+
3327 012520 026701 175706          CMP    EXTEND,R1     ;UPDATE THE SOFTWARE ADDRESS
3328 012524 103413          BLO    7$            ;IS THE LIMIT EXCEEDED
3329 012526 010167 166474          MOV    R1,TEMP4     ;INPUT LIMIT EXCEEDED!! ERROR.
3330 012532 104400 014560          TYPE    MCRLF        ;SAVE THE ADDRESS.
3331 012536 016746 166464          MOV    TEMP4,-(SP)
3332 012542 104402          TYPOC
3333

```

```

:PROGRAM 3
:PROGRAM 3 IS THE SAME AS PROGRAM 1 ONLY YOU THE
:USER HAS THE CHANCE TO ALTER THE MAP WHICH IS
:COMPARED TO THE DATA IN THE ROM ADDRESSES
:NOTE THE FOLLOWING COMMANDS:

```

```

: *D DATA INSERT DATA; HIT LINE FEED TO ESCAPE.
: *R RUN RUN THE PROGRAM
: *L LIST LIST THE SOFTWARE TABLE ON TTY.
: *A ADDRESS INPUT THE ADDRESS OF THE DATA YOU WANT TO ALTER.
: CR CARRAGE RETURN- WHEN IN THE DATA INPUT MODE A CARRAGE RETURN
: WAITS FOR NEW DATA.

```



MAR 1975  
DZBMD.F.P11MACY11 27(657) 1-OCT-75 09:30 PAGE 77  
ERROR POINTER TABLE

SEQ 0085

```

3334 012544 010157 000622          MOV    R1,ADDRESS      ;SAVE THE ADDRESS FOR GOOD
3335 012550 000167 177632          JMP    XHOLD
3336 012554 104400 014544          7$:   TYPE    ,M.OM      ;TYPE A "?"
3337 012560 000167 177622          JMP    XHOLD
3338
3339          ;YOU ARE HERE BECAUSE YOU HIT AN "A"
3340          ;YOU TOLD ME YOU WERE GOING TO INPUT AN ADDRESS.
3341          ;SO INPUT THE ADDRESS AND TERMINATE WITH A CARRAGE RETURN.
3342          ;OK??
3343
3344 012564 104414          SRV.A: RDOCT          ;READ THE ADDRESS HE WANTS TO MODIFY.
3345 012566 012667 000600          MOV    (SP)+,ADDRESS
3346 012572 000167 177610          4$:   JMP    XHOLD
3347
3348          ;YOU ENTERED HERE BECAUSE YOU HIT "L"
3349          ;YOU TOLD ME YOU WANTED A LISTING OF THE SOFTWARE MAP
3350          ;SO HERE IT IS.
3351
3352
3353          SRV.L:
3354 012576 016700 175622          MOV    TABLE,RO      ;GET SOFTWARE MAP
3355 012602 016767 175620 000176  MOV    ALLEND,DEAD    ;SET DEAD END POINTER
3356 012610 104400 014344          TYPE    ,MSG4        ;TYPE HEADER
3357 012614 104400 012050          TYPE    ,DH.28      ;TYPE ADDRESS ADD+XX
3358 012620 012767 000007 166374 1$:   MOV    #7,TEMP5      ;SET COUNTER FOR ACCROSS PAGE
3359 012626 011067 166374          MOV    (RO),TEMP4    ;GET DATA
3360 012632 010067 166366          MOV    RO,TEMP3      ;GET ADDRESS
3361 012636 005720          TST    (RO)+         ;UPDATE ADDRESS POINTER
3362 012640 104400 014560          TYPE    ,MCRLF
3363
3364 012644 016746 166354          MOV    TEMP3,-(SP)
3365 012650 104402          TYPOC
3366
3367 012652 104400 014551          TYPE    ,MSPACE
3368
3369 012656 016746 166344          MOV    TEMP4,-(SP)
3370 012662 104402          TYPOC
3371
3372 012664 104400 014551          TYPE    ,MSPACE
3373
3374 012670 011067 166332          2$:   MOV    (RO),TEMP4    ;GET DATA
3375 012674 010067 166324          MOV    RO,TEMP3      ;GET ADDRESS
3376 012700 005720          TST    (RO)+         ;UPDATE POINTER
3377
3378 012702 016746 166320          MOV    TEMP4,-(SP)
3379 012706 104402          TYPOC
3380 012710 104400 014551          TYPE    ,MSPACE
3381
3382 012714 016703 000066          3$:   MOV    DEAD,R3
3383 012720 005723          TST    (R3)+         ;UPDATE POINTER
3384 012722 020003          CMP    RO,R3         ;LIMIT DONE ??
3385 012724 001404          BEQ    $$           ;BR IF YES
3386 012726 005367 166270          4$:   DEC    TEMPS      ;DEC DATA COUNTER
3387 012732 001356          BNE    2$          ;BR IF MORE DATA TO GO

```

```

3388 012734 000731          BR      1$          ;TYPE THE ADDRESS$
3389 012736          5$:          BIT      #BIT0,SWR      ;EXTENDED SOFTWARE DUMP?
3390 012736 032767 000001 164624  BEQ     6$          ;BR IF NO DUMP
3391 012744 001416          TST     -(R3)       ;PUSH POINTER
3392 012746 005743          CMP     EXTEND,R3
3393 012750 026703 175456          BEQ     6$          ;BR IF ALL DONE
3394 012754 001412          TYPE   ,MSG5       ;TYPE EXTENDED MAP:
3395 012756 104400 014410          TYPE   ,DH.28
3396 012762 104400 012050          MOV     EXTMAP,R0   ;SET POINTER
3397 012766 016700 175436          MOV     EXTEND,DEAD ;SET DEAD END POINTER
3398 012772 016767 175434 000006  BR      1$          ;DO IT AGAIN SAM.
3399 013000 000707          JMP     XHOLD
3400 013002 000167 177400  6$:          XHOLD
3401 013006 000000  DEAD:      0
3402
3403          ;NOW YOU ARE HERE BECAUSE YOU WANT TO RUN THE PROGRAM
3404          ;REMEMBER NOW, YOU SET UP THE MAP.
3405          ;ARE YOU SURE YOU TYPED IN THE CORRECT DATA.??
3406          ;HERE WE GO
3407
3408 013010          SRV.R:
3409 013010 012737 014676 000004  RUN3:  MOV     #NO.TRAP,2#4 ;GET READY FOR UNEXPECTED TRAP
3410 013016 012767 000500 166174  MOV     #500,ICOUNT ;DO TEST 500(8) TIMES
3411 013024 012700 173000          RUN.3: MOV     #173000,R0 ;SET BEGGING ADDRESS
3412 013030 012767 013054 166050  MOV     #2$, $LPADR ;IF SW14=1: GOTO 2$ WHEN I HIT "SCOPE"
3413 013036 016704 175362          MOV     TABLE,R4 ;SET SOFTWARE RESULTS
3414 013042 016767 175370 176170  MOV     LAST, LAST ;SET LAST ADDRESS
3415 013050 012703 000005  1$:      MOV     #5, R3 ;DO EACH ADDRESS 5 TIMES.
3416 013054 022700 173024  2$:      CMP     #173024,R0 ;DON'T DO THE VECTOR ADD.
3417 013060 001001          BNE    30$         ;BR IF NOT THE VECTOR ADD.
3418 013052 022024          CMP     (R0)+,(R4)+ ;UPDATE TO NEXT ADDRESS
3419 013064 022700 173224  30$:     CMP     #173224,R0 ;IS THIS THE SECOND TRAP VECTOR??
3420 013070 001001          BNE    10$         ;BR IF NOT VECTOR
3421 013072 022024          CMP     (R0)+,(R4)+ ;UPDATE THE POINTERS !!
3422 013074 010467 166024  10$:     MOV     R4,$GCDAT
3423 013100 010067 166022          MOV     R0,$BDDAT
3424 013104 011067 166116          MOV     (R0),TEMP4 ;READ THE ADDRESS
3425 013110 011467 166110          MOV     (R4),TEMP3 ;READ THE SOFTWARE ADDRESS
3426 013114 026767 166104 166104  CMP     TEMP3,TEMP4
3427 013122 001401          BEQ    11$         ;BRANCH IF OK
3428 013124 104001          ERROR 1 ;INCORRECT COMPARISON.
3429 013126 032767 004000 164434  11$:     BIT     #BIT11,SWR ;QUICK PASS.
3430 013134 001002          BNE    12$         ;BR IF YES
3431 013136 005303          DEC    R3 ;HAS THAT ADD BEEN READ 10 TIMES?
3432 013140 001345          BNE    2$          ;BR IF NOT 10 TIMES
3433 013142 026700 176072  12$:     CMP     LAST,R0 ;WAS LAST ADDRESS CHECKED?
3434 013146 001403          BEQ    15$         ;BR IF YES
3435 013150 000004          SCOPE ;LOCK ON THIS ADDRESS?
3436 013152 022024          CMP     (R0)+,(R4)+ ;UPDATE THE POINTERS.
3437 013154 000735          BR     1$          ;CONTINUE THE TEST.
3438 013156 032767 000001 164404  15$:     BIT     #BIT0,SWR ;EXTENDED WORD TO BE CHECKED?
3439 013164 001413          BEQ    3$          ;BR IF NO CHECKING.
3440 013166 022767 173776 176044  CMP     #173776, LAST ;IS ALL THE TEST DONE?
3441 013174 001407          BEQ    3$          ;BR IF YES.

```

3442	013176	012767	173776	176034	MOV	#173776, LAST	;SET LAST ADDRESS.
3443	013204	016704	175220		MOV	EXTMAP, R4	;SET EXTENDED MAP.
3444	013210	005720			TST	(R0)+	;POP POINTER
3445	013212	000716			BR	1\$	;GO DO THE TEST.

```

3446                                     ;TEST THAT WRITING ROM RESULTS IN A TIME OUT
3447                                     ;TRAP.
3448
3449 013214 012700 173000 3$: MOV #173000,RO ;SET BASE ADDRESS
3450 013220 012767 013240 165660 MOV #5$, $LPADR ;IF SW14=1; GOTO 5$ AT SCOPE
3451 013226 012737 013274 000004 MOV #6$, 2#4 ;TIME OUT TRAP; GOTO 6$
3452 013234 012703 000012 4$: MOV #10, R3 ;DO EACH ADD 10 TIMES
3453 013240 022700 173024 5$: CMP #173024,RO ;IS THIS AT THE TRAP VECTOR
3454 013244 001001 BNE 20$ ;BR IF NO
3455 013246 005720 TST (RO)+ ;UPDATE POINTER
3456 013250 022700 173224 20$: CMP #173224,RO ;IS THIS AT THE SECOND TRAP VECTOR
3457 013254 001001 BNE 21$ ;BR IF NO
3458 013256 005720 TST (RO)+ ;UPDATE THE POINTER
3459 013260 012710 177777 21$: MOV #-1, (RO) ;WRITE ROM WITH A -1
3460 013264 000240 NOP ;WAIT ONE INSTR. TIME
3461 013266 010067 165734 MOV RO,TEMP4
3462 013272 104002 ERROR 2 ;WRITING ROM DIDN'T TIME OUT.
3463 013274 012706 001100 6$: MOV #STACK,SP ;RESTORE STACK
3464 013300 032767 004000 164262 BIT #BIT11,SWR ;QUICK PASS?
3465 013306 001002 BNE 22$ ;BR IF YES
3466 013310 005303 DEC R3 ;DO EACH ADD 10 TIMES
3467 013312 001352 BNE 5$ ;NOT DONE WITH THIS ONE YET.
3468 013314 032767 000001 164246 22$: BIT #BIT0,SWR ;IS THE EXTENDED 128. WORDS TO BE TESTED??
3469 013322 001404 BEQ 23$ ;BR IF NO
3470 013324 022700 173776 CMP #173776,RO ;IS THE EXTENDED LIMIT BEEN TESTED?
3471 013330 001407 BEQ 7$ ;IF YES; GOTO 7$
3472 013332 000403 BR 24$ ;IF NO; KEEP GOING.
3473 013334 026700 175076 23$: CMP LASTA,RO ;ALL DONE??
3474 013340 001403 BEQ 7$ ;IF YES; GOTO 7$
3475 013342 000004 24$: SCOPE ;GO CHECK SW14; (FREEZE !!)
3476 013344 005720 TST (RO)+ ;UPDATE TO NEXT ADDRESS
3477 013346 000732 BR 4$ ;GO DO IT AGAIN
3478 013350 005367 165644 7$: DEC ICOUNT ;CHECK ITERATION COUNT
3479 013354 001004 BNE 8$ ;MORE TO GO
3480 013356 004767 001354 JSR PC,EOP ;GO TO END OF PASS ROUTINE
3481 013362 000167 177422 JMP RUN3 ;GO DO TEST AGAIN
3482 013366 000167 177432 8$: JMP RUN.3
3483
3484 013372 000000 ADDRESS: 0

```

```

3485 ;PROGRAM 4
3486 ;PROGRAM 4 CHECKS THE TRAP VECTOR ADDRESS.
3487 ;THE PROGRAM SIMULATES ACTIVATING THE BUTTON
3488 ;FOR EACH CHANNEL AND THEN READS
3489 ;THE CONTENTS OF THE ADDRESS.
3490 ;ON THE FIRST PASS THE CONTENTS WILL
3491 ;BE TYPED OUT FOR YOU THE
3492 ;USER TO VERIFY. AFTER THIS THE PROGRAM
3493 ;DOES A COMPARE TO THE PREVIOUSLY FOUND DATA
3494 ;AND REPORTS AN ERROR IF DIFFERENT THAN
3495 ;WHAT WAS FOUND BEFORE.
3496
3497 013374 012767 013374 001272 PRG4:  MOV    #PRG4,PRG.NO    ;SET FOR POWER FAIL
3498 013402 005067 165610          CLR    LSTERR        ;PREPARE ERROR CONDITIONS
3499 013406 012706 001100          MOV    #STACK,SP    ;SET THE STACK POINTER
3500 013412 012767 020000 165600  MOV    #20000,ICOUNT ;SET ITERATION COUNT TO 20000(8)
3501 013420 005767 000544          TST    FLAG4        ;HAVE I BEEN HERE BEFOR??
3502 013424 001106          BNE    TAG.A        ;BR IF NOT FIRST TIME HERE.
3503 013426 005167 000536          COM    FLAG4        ;SET THE FLAG
3504 013432 012705 000002          MOV    #2,R5        ;SET R5 FOR SWITCH 1
3505 013436 012704 014160          MOV    #LOC1,R4     ;SET STORAGE LOCATION
3506 013442 012737 014676 000004  MOV    #NO.TRAP,#4  ;SET FOR TIME OUT TRAP
3507 013450 012767 000001 165546  MOV    #1,TEMP3     ;SET FOR MESSAGE ON CHANNEL NO.
3508 013456 104400 014172          TYPE  ,MCHAN        ;TYPE MESSAGE ABOUT CHANNEL
3509 013462 104400 014551          TYPE  ,MSPACE
3510
3511 013466 016746 165532          MOV    TEMP3,-(SP)
3512 013472 104402          TYPOC
3513 013474 104400 014551          TYPE  ,MSPACE
3514
3515
3516 013500 104400 014206          TYPE  ,MACTV        ;TYPE REST OF MESSAGE
3517 013504 104400 014221          TYPE  ,MADD1        ;TYPE ADDRESS MESSAGE
3518 013510 012700 173024          MOV    #173024,RO
3519 013514 005037 173024          CLR    #173024
3520 013520 010537 173024          MOV    R5,#173024  ;WRITE ROM WITH SWITCH
3521 013524 000240          NOP                ;WAIT ONE INSTR. TIME
3522 013526 012706 001100          MOV    #STACK,SP   ;SET THE STACK POINTER
3523 013532 012700 173024          MOV    #173024,RO  ;SET FOR ERROR MESSAGE
3524 013536 012737 014676 000004  MOV    #NO.TRAP,#4 ;SET FOR NO MORE TRAPS
3525 013544 013767 173024 165454  MOV    #173024,TEMP4 ;READ THE ADDRESS
3526
3527 013552 104400 014551          TYPE  ,MSPACE
3528 013556 016746 165444          MOV    TEMP4,-(SP)
3529 013562 104402          TYPOC
3530 013564 013724 173024          MOV    #173024,(R4)+ ;STORE THE INFORMATION FOUND
3531 013570 104400 014255          TYPE  ,MADD2        ;TYPE THE SECOND ADDRESS MSG
3532 013574 012700 173224          MOV    #173224,RO  ;SET FOR ERROR CONDITION.
3533 013600 013767 173224 165420  MOV    #173224,TEMP4 ;STORE ROM DATA
3534 013606 104400 014551          TYPE  ,MSPACE
3535
3536 013612 016746 165410          MOV    TEMP4,-(SP)
3537 013616 104402          TYPOC
3538

```

3539	013620	005267	165400		INC	TEMP3			; GET READY FOR NEXT SWITCH SETTING
3540	013624	000241			CLC				; CLEAR THE CARRY BIT
3541	013626	006105			ROL	R5			; UPDATE R5
3542	013630	022705	000040		CMP	#40, R5			; ALL SIMULATED SWITCHES DONE?
3543	013634	001310			BNE	1\$			; BR IF NOT ALL DONE
3544	013636	000167	177532		JMP	PRG4			; JMP AND DO TEST AGAIN WITH OUT TYPE OUT
3545									
3546	013642	012703	000002		TAG.A:	MOV	#2, R3		; SIMULATE SWITCH 1
3547	013646	012704	014160			MOV	#LOC1, R4		; GET LOCATION WHERE DATA IS STORED
3548	013652	012737	014676	000004	1\$:	MOV	#NO. TRAP, #4		; PREPARE FOR TIME OUT TRAP
3549	013660	005037	173024			CLR	#173024		
3550	013664	010337	173024			MOV	R3, #173024		; WRITE THE ROM
3551	013670	000240				NOP			; WAIT ONE INSTR. TIME
3552	013672	012706	001100		2\$:	MOV	#STACK, SP		; SET THE STACK POINTER.
3553	013676	012737	014676	000004		MOV	#NO. TRAP, #4		; SET FOR NO MORE TRAPS.
3554	013704	012700	173024			MOV	#173024, R0		; SET FOR ERROR MESSAGE
3555	013710	011401				MOV	(R4), R1		; SET FOR COMPARISON
3556	013712	013705	173024			MOV	#173024, R5		; GET THE DATA FROM THE ROM
3557	013716	012767	014160	165200		MOV	#LOC1, \$GDDAT		
3558	013724	012767	173024	165174		MOV	#173024, \$BDDAT		
3559	013732	016767	000222	165264		MOV	LOC1, TEMP3		
3560	013740	013767	173024	165260		MOV	#173024, TEMP4		
3561	013746	020105				CMP	R1, R5		; IS THE DATA THE SAME??
3562	013750	001401				BEQ	30\$		; BR IF GOOD DATA.
3563	013752	104001				ERROR	1		; ERROR. DATA READ FIRST TIME NOT THE SAME
3564	013754	012700	173224		30\$:	MOV	#173224, R0		; SET FOR ERROR MESSAGE
3565	013760	013705	173224			MOV	#173224, R5		; READ THE ROM
3566	013764	012767	173224	165134		MOV	#173224, \$BDDAT		
3567	013772	013767	173224	165226		MOV	#173224, TEMP4		
3568	014000	020105				CMP	R1, R5		; IS THE DATA THE SAME?
3569	014002	001401				BEQ	31\$		; BR IF GOOD DATA
3570	014004	104001				ERROR	1		; ERROR. DATA NOT THE SAME AS BEFORE.
3571	014006	005724			31\$:	TST	(R4)+		; UPDATE DATA POINTER.
3572	014010	000241				CLC			; CLEAR THE CARRY BIT
3573	014012	006103				ROL	R3		; UPDATE THE SIMULATED SWITCH SETTING
3574	014014	022703	000040			CMP	#40, R3		; HAVE ALL SETTING BEEN DONE
3575	014020	001314				BNE	1\$		; BR IF NOT DONE
3576	014022	005367	165172			DEC	ICOUNT		; ITERATION COUNT DONE
3577	014026	001305				BNE	TAG.A		; BR IF NOT DONE
3578	014030	012737	177777	173224		MOV	#-1, #173224		; WRITE SECOND TRAP VECTOR WITH -1
3579	014036	005037	173024			CLR	#173024		; ZERO THE FIRST VECTOR
3580	014042	012700	173024			MOV	#173024, R0		; SET FOR TYPE OUT IF ERROR
3581	014046	016701	000106			MOV	LOC1, R1		; SET FOR TYPE OUT ROUTINE
3582	014052	013705	173024			MOV	#173024, R5		; SAME AS ABOVE
3583	014056	012767	173024	165042		MOV	#173024, \$BDDAT		
3584	014064	013767	173024	165134		MOV	#173024, TEMP4		
3585	014072	020105				CMP	R1, R5		; IS DEFAULT LINE SELECTED =TO LINE 1
3586	014074	001401				BEQ	32\$		; BR IF DEFAULT EQUALS LINE 1
3587	014076	104001				ERROR	1		; DATA NOT EQUAL TO LINE 1
3588	014100	012737	177777	173024	32\$:	MOV	#-1, #173024		; WRITE A -1 TO FIRST VECTOR
3589	014106	005037	173224			CLR	#173224		; ZERO SECOND VECTOR
3590	014112	012700	173224			MOV	#173224, R0		; SET FOR TYPE OUT IF ERROR
3591	014116	016701	000036			MOV	LOC1, R1		; GET DATA
3592	014122	013705	173224			MOV	#173224, R5		; READ ROM

3593	014126	012767	173224	164772	MOV	#173224, \$BDDAT	
3594	014134	013767	173224	165064	MOV	@173224, TEMP4	
3595							
3596	014142	020105			CMP	R1, R5	: IS LINE 1 DEFAULT LINE
3597	014144	001401			BEQ	33\$	: BR IF OK
3598	014146	104001			ERROR	1	: ERROR LINE 1 NOT DEFAULT LINE
3599	014150	004767	000562		JSR	PC, EOP	: TYPE END MESSAGE.
3600	014154	000167	177214	33\$:	JMP	PRG4	: GOTO PROGRAM 4 AGAIN
3601							
3602	014160	000000			LOC1:	0	
3603	014162	000000			LOC2:	00	
3604	014164	000000			LOC3:	00	
3605	014166	000000			LOC4:	00	
3606	014170	000000			FLAG4:	0	

3607	014172	005015	041412	040510	MCHAN:	.ASCIZ <15><12><12>/CHANNEL /
	014206	041501	044524	040526	MACTV:	.ASCIZ/ACTIVATED./
	014221	015	040412	042104	MADD1:	.ASCIZ <15><12>/ADDRESS 773024 CONTAINS: /
	014255	015	040412	042104	MADD2:	.ASCIZ <15><12>/ADDRESS 773224 CONTAINS: /
		014312			.EVEN	
3608						
3609	014312	005015	051120	043517	MSG3:	.ASCIZ <15><12>/PROGRAM NO. (1,2,3,4) /
	014344	006414	016412	077437	MSG4:	.ASCIZ <14><15><12><35><37><177><177><177>/SOFTWARE MAP IS AS FOLLOWS:/
	014410	005015	020012	054105	MSG5:	.ASCIZ <15><12><12>/ EXTENDED SOFTWARE MAP FOLLOWS./
	014453	015	025012	000	MASTER:	.ASCIZ <15><12>/**/
	014457	007	006407	042412	M.END:	.ASCIZ <7><7><15><12>/END PASS BM873-Y/
	014504				MFAIL:	
	014504	005015	053520	020122		.ASCII <15><12>/PWR UP AFTER/
	014522	005015	042522	046101		.ASCIZ <15><12>/REAL PWR FAIL/
	014542	000044			M.DOL:	.ASCIZ /S/
	014544	005015	037477	000	M.QM:	.ASCIZ <15><12>??/
	014551	040	000040		MSPACE:	.ASCIZ / /
	014554	020040	000040		SPACE3:	.ASCIZ / /
	014560	005015	000		MCRLF:	.ASCIZ <15><12>
	014563	012	000		MLF:	.ASCIZ <12>
		014566			.EVEN	

3610



3611	014566	005067	164424		.PFAIL: CLR	LSTERR	
3612	014572	013746	000004		MOV	#4, -(SP)	
3613	014576	012737	014626	000004	MOV	#15, #4	
3614	014604	005737	173000		TST	#173000	: IS THIS PF REAL?
3615	014610	000240			NOP		: TRAP IS CAUSED BY LOADER
3616	014612	012737	014636	000024	MOV	#PWR.UP, #24	; ITS REAL. PREPARE FOR PWR UP
3617	014620	012637	000004		MOV	(SP)+, #4	
3618	014624	000000			HALT		
3619	014626	005726			IS: TST	(SP)+	: POP THE STACK.
3620	014630	012637	000004		MOV	(SP)+, #4	
3621	014634	000000			HALT		: HARDWARE ERROR. BOOT DIDN'T FORCE
3622					: HIGH ADDR LINES AND LOAD BUTTON WAS ACTIVATED		
3623	014636	012737	014566	000024	PWR.UP: MOV	#.PFAIL, #24	
3624	014644	012706	001100		MOV	#STACK, SP	
3625	014650	005000			CLR	RO	: SET DELAY
3626	014652	062700	000001		IS: ADD	#1, RO	: WAIT FOR TTY
3627	014656	001375			BNE	IS	
3628	014660	104400	014504		TYPE	.MFAIL	: TYPE FAILED.
3629	014664	005067	163106		CLR	PS	: SET STATUS TO ZERO
3630	014670	000177	000000		JMP	#PRG.NO	
3631	014674	000000			PRG.NO: 0		
3632	014676				NO.TRAP:		
3633	014676	011667	000032		MOV	(SP), XSTORE	
3634	014702	032716	100000		BIT	#BIT15, (SP)	
3635	014706	001410			BEQ	IS	
3636	014710	011600			MOV	(SP), RO	
3637	014712	104004			ERROR	4	
3638	014714	012706	001100		MOV	#STACK, SP	
3639	014720	005067	163052		CLR	PS	
3640	014724	000177	177744		JMP	#PRG.NO	
3641	014730	104003			IS: ERROR	3	
3642	014732	000002			RTI		
3643	014734	000000			XSTORE: 0		
3644							
3645	014736	005067	164254		EOP: CLR	LSTERR	
3646	014742	104400	014457		TYPE	.M.END	
3647	014746	104400	015004		TYPE	.VERSION	
3648	014752	013701	000042		MOV	#42, R1	
3649	014756	001411			BEQ	X1	
3650	014760	022767	010650	177706	CMP	#PRG1, PRG.NO	
3651	014766	001002			BNE	.+6	
3652	014770	000167	176400		JMP	PRG4	
3653	014774				SENDAD:		
3654	014774	004711			LOGIC: JSR	PC, (R1)	
3655	014776	000240			NOP		
3656	015000	000240			NOP		
3657	015002	000207			X1: RTS	PC	
3658	015004	000101			VERSION: 101		: SEVEN BIT ASCII FOR DEFAULT "A"

```

3659 015006 005015 041520 020072 MERRPC: .ASCIZ <15><12>/PC: /
3660 015014 000
3661 015016 .EVEN
3662 .MCALL .SEOP, .STYPE, .STYPOCT, .SPOWER, .SREAD
3663 ;*****
3664 .SBTTL TYPE ROUTINE
3665
3666 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
3667 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
3668 ;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
3669 ;*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
3670 ;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
3671 *
3672 ;*CALL:
3673 ;*1) USING A TRAP INSTRUCTION
3674 * TYPE ,MESADR ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
3675 *
3676 ;*OR
3677 * TYPE
3678 * MESADR
3679 *
3680 ;*2) USING A JSR INSTRUCTION
3681 * MOV PS,-(SP) ;PUSH PROCESSOR STATUS WORD ON THE STACK
3682 * JSR PC,$TYPE ;CALL TYPE ROUTINE
3683 * $MESADDR ;FIRST ADDRESS OF MESSAGE
3684
3685 015016 105767 164127 $TYPE: TSTB $TFPLG ;IS THERE A TERMINAL?
3686 015022 100002 BPL $S ;BR IF YES
3687 015024 000000 HALT ;HALT HERE IF NO TERMINAL
3688 015026 000407 BR $S ;LEAVE
3689 015030 010046 $S: MOV RO,-(SP) ;SAVE RO
3690 015032 017600 000002 MOV $2(SP),RO ;GET ADDRESS OF ASCIZ STRING
3691 015036 112046 $S: MOVB (RO)+,-(SP) ;PUSH CHARACTER TO BE TYPED ONTO STACK
3692 015040 001005 BNE $S ;BR IF IT ISN'T THE TERMINATOR
3693 015042 005726 TST (SP)+ ;IF TERMINATOR POP IT OFF THE STACK
3694 015044 012600 MOV (SP)+,RO ;RESTORE RO
3695 015046 062716 000002 $S: ADD $2,(SP) ;ADJUST RETURN PC
3696 015052 000002 RTI ;RETURN
3697 015054 004767 000026 $S: JSR PC,$S ;GO TYPE THIS CHARACTER
3698 015060 126726 164064 $S: CMPB $FILLC,(SP)+ ;IS IT TIME FOR FILLER CHARS.?
3699 015064 001364 BNE $S ;IF NO GO GET NEXT CHAR.
3700 015066 016746 164054 MOV $NULL,-(SP) ;GET # OF FILLER CHARS. NEEDED
3701 ;AND THE NULL CHAR.
3702 015072 105366 000001 $S: DECB 1(SP) ;DOES A NULL NEED TO BE TYPED?
3703 015076 002770 BLT $S ;BR IF NO--GO POP THE NULL OFF OF STACK
3704 015100 004767 000002 JSR PC,$S ;GO TYPE A NULL
3705 015104 000772 BR $S ;LOOP
3706 015106 105777 164030 $S: TSTB $STPS ;WAIT UNTIL PRINTER IS READY
3707 015112 100375 BPL $S
3708 015114 116677 000002 164022 MOVB 2(SP),$STPB ;LOAD CHAR TO BE TYPED INTO DATA REG.
3709 015122 000207 RTS PC
3710 ;*****
3711 .SBTTL TTY INPUT ROUTINE
3712

```

```

3713
3714      ;*INPUT A SINGLE CHARACTER FROM THE TTY
3715      ;*CALL:
3716      ;*      RDCHR                      ;INPUT A SINGLE CHARACTER FROM THE TTY
3717      ;*      RETURN HERE                ;CHARACTER IS ON THE STACK
3718
3719
3720 015124 011646 SRDCHR: MOV      (SP), -(SP)      ;PUSH DOWN THE PC
3721 015126 016666 000004 000002      MOV      4(SP), 2(SP)      ;SAVE THE PS
3722 015134 105777 163776      15:      TSTB      2(STKB)      ;WAIT FOR
3723 015140 103375      BPL      15              ;A CHARACTER
3724 015142 117766 163772 000004      MOVB     2(STKB), 4(SP)    ;READ THE TTY
3725 015150 042766 177600 000004      BIC      #177, 4(SP)     ;GET RID OF JUNK IF ANY
3726 015156 000002      RTI                      ;GO BACK TO USER
3727      ;*****
3728      ;*INPUT A STRING FROM THE TTY
3729      ;*CALL:
3730      ;*      RDLIN                      ;INPUT A STRING FROM THE TTY
3731      ;*      RETURN HERE                ;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
3732      ;*                                  ;TERMINATOR WILL BE A BYTE OF ALL 0'S
3733
3734 015160 010346 SRDLIN: MOV      R3, -(SP)      ;SAVE R3
3735 015162 012703 015266 15:      MOV      #177, R3        ;GET ADDRESS
3736 015166 022703 015276 25:      CMP      #177, R3        ;BUFFER FULL?
3737 015172 101405      BLOS     45              ;BR IF YES
3738 015174 104410      RDCHR     ;GO READ ONE CHARACTER FROM THE TTY
3739 015176 112613      MOVB     (SP)+, (R3)     ;GET CHARACTER
3740 015200 122713 000177      CMPB     #177, (R3)     ;IS IT A RUBOUT
3741 015204 001303      BNE      35              ;SKIP IF NOT
3742 015206 104400 001152 45:      TYPE     $QUES         ;TYPE A '?'
3743 015212 000763      BR       15              ;CLEAR THE BUFFER AND LOOP
3744 015214 111367 000044 35:      MOVB     (R3), B5        ;ECHO THE CHARACTER
3745 015220 104400 015264      TYPE     B5
3746 015224 122723 000015      CMPB     #15, (R3)+     ;CHECK FOR RETURN
3747 015230 001356      BNE      25              ;LOOP IF NOT RETURN
3748 015232 105063 177777      CLRB     -1(R3)         ;CLEAR RETURN (THE 15)
3749 015236 104400 001154      TYPE     $LF            ;TYPE A LINE FEED
3750 015242 012603      MOV      (SP)+, R3      ;RESTORE R3
3751 015244 011646      MOV      (SP), -(SP)    ;ADJUST THE STACK AND PUT ADDRESS OF THE
3752 015246 016666 000004 000002      MOV      4(SP), 2(SP)    ;FIRST ASCII CHARACTER ON IT
3753 015254 012766 015266 000004      MOV      #177, 4(SP)
3754 015262 000002      RTI                      ;RETURN
3755 015264      000      B5:      .BYTE     0              ;STORAGE FOR ASCII CHAR. TO TYPE
3756 015265      000      .BYTE     0              ;TERMINATOR
3757 015266 000010      STTYIN: .BLKB     8      ;RESERVE 8 BYTES FOR TTY INPUT
3758      ;*****
3759
3760      .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3761
3762      ;*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3763      ;*CALL:
3764      ;*      MOV      NUM, -(SP)          ;NUMBER TO BE TYPED
3765      ;*      TYPOS     ;CALL FOR TYPEOUT
3766      ;*      .BYTE     N                  ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE

```

```

3767      *      .BYTE      M      ;M=1 OR 0
3768      *      ;          ;1=TYPE LEADING ZEROS
3769      *      ;          ;0=SUPPRESS LEADING ZEROS
3770      *
3771      *$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3772      *$TYPOS OR $TYPOC
3773      *CALL:
3774      *      MOV      NUM,-(SP)      ;NUMBER TO BE TYPED
3775      *      TYPON      ;CALL FOR TYPEOUT
3776      *
3777      *$TYPJC----ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3778      *CALL:
3779      *      MOV      NUM,-(SP)      ;NUMBER TO BE TYPED
3780      *      TYPOC      ;CALL FOR TYPEOUT
3781
3782 015276 017646 000000      $TYPOS: MOV      2(SP),-(SP)      ;PICKUP THE MODE
3783 015302 116667 000001 000211      MOVB     1(SP),$OFILL      ;LOAD ZERO FILL SWITCH
3784 015310 112567 000207      MOVB     (SP)+,$OMODE+1      ;NUMBER OF DIGITS TO TYPE
3785 015314 062716 000002      ADD      #2,(SP)      ;ADJUST RETURN ADDRESS
3786 015320 000406      BR      $TYPON
3787 015322 112767 000001 000171      $TYPOC: MOVB     #1,$OFILL      ;SET THE ZERO FILL SWITCH
3788 015330 112767 000006 000165      MOVB     #6,$OMODE+1      ;SET FOR SIX(6) DIGITS
3789 015336 112767 000005 000154      $TYPON: MOVB     #5,$OCNT      ;SET THE ITERATION COUNT
3790 015344 010346      MOV      R3,-(SP)      ;SAVE R3
3791 015346 010446      MOV      R4,-(SP)      ;SAVE R4
3792 015350 010546      MOV      R5,-(SP)      ;SAVE R5
3793 015352 116704 000145      MOVB     $OMODE+1,R4      ;GET THE NUMBER OF DIGITS TO TYPE
3794 015356 005404      NEG      R4
3795 015360 062704 000006      ADD      #6,R4      ;SUBTRACT IT FOR MAX. ALLOWED
3796 015364 110467 000132      MOVB     R4,$OMODE      ;SAVE IT FOR USE
3797 015370 116704 000125      MOVB     $OFILL,R4      ;GET THE ZERO FILL SWITCH
3798 015374 016605 000012      MOV      12(SP),R5      ;PICKUP THE INPUT NUMBER
3799 015400 005003      CLR      R3      ;CLEAR THE OUTPUT WORD
3800 015402 006105      1$:     ROL      R5      ;ROTATE MSB INTO "C"
3801 015404 000404      BR      3$      ;GO DO MSB
3802 015406 006105      2$:     ROL      R5      ;FORM THIS DIGIT
3803 015410 006105      ROL      R5
3804 015412 006105      ROL      R5
3805 015414 010503      MOV      R5,R3
3806 015416 006103      3$:     ROL      R3      ;GET LSB OF THIS DIGIT
3807 015420 105367 000076      DECB     $OMODE      ;TYPE THIS DIGIT?
3808 015424 100016      BPL      7$      ;BR IF NO
3809 015426 042703 177770      BIC      #177770,R3      ;GET RID OF JUNK
3810 015432 001002      BNE      4$      ;TEST FOR 0
3811 015434 005704      TST      R4      ;SUPPRESS THIS 0?
3812 015436 001403      BEQ      5$      ;BR IF YES
3813 015440 005204      4$:     INC      R4      ;DON'T SUPPRESS ANYMORE 0'S
3814 015442 052703 000060      BIS      #'0,R3      ;MAKE THIS DIGIT ASCII
3815 015446 052703 000040      5$:     BIS      #' ,R3      ;MAKE ASCII IF NOT ALREADY
3816 015452 110367 000040      MOVB     R3,8$      ;SAVE FOR TYPING
3817 015456 104400 015516      TYPE     8$      ;GO TYPE THIS DIGIT
3818 015462 105367 000032      7$:     DECB     $OCNT      ;COUNT BY 1
3819 015466 003347      BGT      2$      ;BR IF MORE TO DO
3820 015470 002402      BLT      6$      ;BR IF DONE

```

```

3821 015472 005204          INC      R4          ;INSURE LAST DIGIT ISN'T A BLANK
3822 015474 000744          BR       2$          ;GO DO THE LAST DIGIT
3823 015476 012605          6$:     MOV      (SP)+,R5 ;RESTORE R5
3824 015500 012604          MOV      (SP)+,R4 ;RESTORE R4
3825 015502 012603          MOV      (SP)+,R3 ;RESTORE R3
3826 015504 016666 000002 000004 MOV      2(SP),4(SP) ;SET THE STACK FOR RETURNING
3827 015512 012616          MOV      (SP)+,(SP)
3828 015514 000002          RTI          ;RETURN
3829 015516 000          8$:     .BYTE   0          ;STORAGE FOR ASCII DIGIT
3830 015517 000          .BYTE   0          ;TERMINATOR FOR TYPE ROUTINE
3831 015520 000          $OCNT:  .BYTE   0          ;OCTAL DIGIT COUNTER
3832 015521 000          $OFILL: .BYTE   0          ;ZERO FILL SWITCH
3833 015522 000000          $OMODE: 0          ;NUMBER OF DIGITS TO TYPE
3834 ;;*****
3835
3836 .SBTTL SCOPE HANDLER ROUTINE
3837
3838 ;*SW14=1          LOOP ON TEST
3839 ;*THE TEST NUMBER ($TSTNM) IS INCREMENTED AND DISPLAYED IN DISPLAY<7:C>
3840 ;*AND THE ERROR FLAG ($ERFLG) IS DISPLAYED IN DISPLAY<15:08>
3841
3842 015524          $$SCOPE:
3843 015524 006137 177570          ROL     2,$SWR          ;LOOP ON PRESENT TEST?
3844 015530 100425          BMI     $OVER          ;YES IF SW14=1
3845 ;*****START OF CODE FOR THE XOR TESTER*****
3846 015532 000416          $XTSTR: BR       6$          ;IF RUNNING ON THE "XOR" TESTER CHANGE
3847 ;THIS INSTRUCTION TO A "NOP" (NOP=240)
3848 015534 013746 000004          MOV     2,$ERRVEC,-(SP) ;SAVE THE CONTENTS OF THE ERROR VECTOR
3849 015540 012737 015560 000004 MOV     $5,$ERRVEC          ;SET FOR TIMEOUT
3850 015546 005737 177060          TST    2,$177060          ;TIME OUT ON XOR?
3851 015552 012637 000004          MOV     (SP)+,2,$ERRVEC ;RESTORE THE ERROR VECTOR
3852 015556 000404          BR     $$VLAD          ;GO TO THE NEXT TEST
3853 015560 022626          5$:     CMP     (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
3854 015562 012637 000004          MOV     (SP)+,2,$ERRVEC ;RESTORE THE ERROR VECTOR
3855 015566 000406          BR     $OVER          ;LOOP ON THE PRESENT TEST
3856 015570          6$:     ;*****END OF CODE FOR THE XOR TESTER*****
3857 015570 105267 163306          $$VLAD: INCB    $TSTNM          ;COUNT TEST NUMBERS
3858 015574 011667 163306          MOV     (SP),$LPADR          ;SAVE SCOPE LOOP ADDRESS
3859 015600 105067 163277          CLRB   $ERFLG          ;ZERO THE ERROR FLAG
3860 015604 016737 163272 177570 $OVER:  MOV     $TSTNM,2,$DISPLAY ;DISPLAY TEST NUMBER
3861 015612 016716 163270          MOV     $LPADR,(SP) ;FUDGE RETURN ADDRESS
3862 015616 000002          RTI          ;FIXES PS
3863 ;;*****
3864
3865 .SBTTL ERROR HANDLER ROUTINE
3866
3867 ;*SW15=1          HALT ON ERROR
3868 ;*SW13=1          INHIBIT ERROR TYPEOUTS
3869 ;*GO TO $ERRTYP ON ERROR
3870
3871 015620          $ERROR:
3872 015620 105267 163257          7$:     INCB    $ERFLG          ;SET THE ERROR FLAG
3873 015624 001775          BEQ    7$          ;DON'T LET THE FLAG GO TO ZERO
3874 015626 016737 163250 177570          MOV     $TSTNM,2,$DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG

```

```

3875 015634 005267 163252      INC      SERTTL      ;INC THE ERROR COUNT
3876 015640 011667 163252      MOV      (SP),SERRPC ;GET ADDRESS OF ERROR INSTRUCTION
3877 015644 162767 000002 163244      SUB      #2,SERRPC
3878 015652 117767 163240 163234      MOV      @SERRPC,$ITEMB ;STRIP AND SAVE THE ERROR ITEM CODE
3879 015660 032737 020000 177570      BIT      #SW13,@SWR
3880 015666 001004          BNE      2$         ;SKIP TYPEOUT IF SET
3881 015670 004737 015712      JSR      PC,@SERRTYF ;SKIP TYPEOUTS
3882 015674 104400 001153          ;GO TO USER ERROR ROUTINE
3883 015700 005737 177570      2$:      TST      @SWR
3884 015704 100001          BPL      3$         ;HALT ON ERROR
3885 015706 000000          HALT
3886 015710 000002          3$:      RTI         ;SKIP IF CONTINUE
3887          ;*****
3888          ;*****
3889          ;*****
3890          ;*****
3891          ;*****
3892          ;*****
3893          ;*****
3894          ;*****
3895          ;*****
3896 015712 104400 001153      TYPE    $CRLF      ;"CARRIAGE RETURN" & "LINE FEED"
3897 015716 010046          MOV      RO,-(SP)   ;SAVE RO
3898 015720 005000          CLR      RO         ;PICKUP THE ITEM INDEX
3899 015722 153700 001114      BISB    @,$ITEMB,RO
3900 015726 001004          BNE      1$         ;IF ITEM NUMBER IS ZERO, JUST
3901          ;TYPE THE PC OF THE ERROR
3902 015730 016746 163162      MOV      SERRPC,-(SP) ;SAVE SERRPC FOR TYPEOUT
3903          ;ERROR ADDRESS
3904 015734 104402          TYPCC
3905 015736 000426          BR      6$         ;GO TYPE--OCTAL ASCII(ALL DIGITS)
3906 015740 005300          1$:      DEC      RO         ;GET OUT
3907 015742 006300          ASL      RO         ;ADJUST THE INDEX SO THAT IT WILL
3908 015744 006300          ASL      RO         ;WORK FOR THE ERROR TABLE
3909 015746 006300          ASL      RO
3910 015750 062700 001156      ADD      #SERRTB,RO ;FORM TABLE POINTER
3911 015754 012067 000004      MOV      (RO)+,2$   ;PICKUP "ERROR MESSAGE" POINTER
3912 015760 001404          BEQ      3$         ;SKIP TYPEOUT IF NO POINTER
3913 015762 104400          TYPE
3914 015764 000000          2$:      .WORD    0      ;TYPE THE "ERROR MESSAGE"
3915 015766 104400 001153      TYPE    $CRLF      ;"ERROR MESSAGE" POINTER GOES HERE
3916 015772 012067 000004      3$:      MOV      (RO)+,4$   ;"CARRIAGE RETURN" & "LINE FEED"
3917 015776 001404          BEQ      5$         ;PICKUP "DATA HEADER" POINTER
3918 016000 104400          TYPE
3919 016002 000000          4$:      .WORD    0      ;SKIP TYPEOUT IF 0
3920 016004 104400 001153      TYPE    $CRLF      ;TYPE THE "DATA HEADER"
3921 016010 011000          5$:      MOV      (RO),RO   ;"DATA HEADER" POINTER GOES HERE
3922 016012 001004          BNE      7$         ;"CARRIAGE RETURN" & "LINE FEED"
3923 016014 012600          6$:      MOV      (SP)+,RO  ;PICKUP "DATA TABLE" POINTER
3924 016016 104400 001153      TYPE    $CRLF      ;GO TYPE THE DATA
3925 016022 000207          RTS      PC        ;RESTORE RO
3926 016024          7$:
3927 016024 013046          MOV      @2(RO)+,-(SP) ;SAVE @2(RO)+ FOR TYPEOUT
3928 016026 104402          TYPCC ;GO TYPE--OCTAL ASCII(ALL DIGITS)

```

```

3929 016030 005710          TST      (R0)          ; IS THERE ANOTHER NUMBER?
3930 016032 001770          BEQ      6$           ; BR IF NO
3931 016034 104400 016042  TYPE      8$           ; TYPE TWO(2) SPACES
3932 016040 000771          BR       7$           ; LOOP
3933 016042 020040 000      8$: .ASCIZ  / /          ; TWO(2) SPACES
3934 016046 016046          .EVEN
3935 ;*****
3936
3937 .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
3938
3939 ;*CALL:
3940 ;*      RDOCT          ; READ AN OCTAL NUMBER
3941 ;*      RETURN HERE   ; LOW ORDER BITS ARE ON TOP OF THE STACK
3942 ;*
3943
3944 016046 011646 000004 000002 $RDOCT: MOV      (SP), -(SP)      ; PROVIDE SPACE FOR THE
3945 016050 016666          MOV      4(SP), 2(SP)      ; INPUT NUMBER
3946 016056 010046          MOV      RO, -(SP)        ; PUSH RO ON STACK
3947 016060 010146          MOV      R1, -(SP)        ; PUSH R1 ON STACK
3948 016062 010246          MOV      R2, -(SP)        ; PUSH R2 ON STACK
3949 016064 104412          1$:  RDLIN          ; READ AN ASCIZ LINE
3950 016066 012600          MOV      (SP)+, RO        ; GET ADDRESS OF 1ST CHARACTER
3951 016070 005001          CLR      R1              ; CLEAR DATA WORD
3952 016072 005002          CLR      R2
3953 016074 112046          2$:  MOVB      (R0)+, -(SP)  ; PICKUP THIS CHARACTER
3954 016076 001412          BEQ      3$           ; IF ZERO GET OUT
3955 016100 006301          ASL      R1              ; *2
3956 016102 006102          ROL      R2
3957 016104 006301          ASL      R1              ; *4
3958 016106 006102          ROL      R2
3959 016110 006301          ASL      R1              ; *8
3960 016112 006102          ROL      R2
3961 016114 042716 177770  BIC      #1C7, (SP)      ; STRIP THE ASCII JUNK
3962 016120 062601          ADD      (SP)+, R1      ; ADD IN THIS DIGIT
3963 016122 000764          BR       2$           ; LOOP
3964 016124 005726          3$:  TST      (SP)+        ; CLEAN TERMINATOR FROM STACK
3965 016126 010166 000012  MOV      R1, 12(SP)      ; SAVE THE RESULT
3966 016132 010267 000010  MOV      R2, $HIOCT
3967 016136 012602          MOV      (SP)+, R2
3968 016140 012601          MOV      (SP)+, R1
3969 016142 012600          MOV      (SP)+, RO
3970 016144 000002          RTI
3971 016146 000000          $HIOCT: .WORD  0        ; HIGH ORDER BITS GO HERE
3972 ;*****
3973
3974 .SBTTL  TRAP DECODER
3975
3976 ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
3977 ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
3978 ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
3979 ;*GO TO THAT ROUTINE.
3980
3981 016150 010046          $TRAP: MOV      RO, -(SP)      ; SAVE RO
3982 016152 016500 000002  MOV      2(SP), RO      ; GET TRAP ADDRESS

```

```

3983 016156 005740          TST      -(RO)          ;BACKUP BY 2
3984 016160 111000          MOVB     (RO),RO        ;GET RIGHT BYTE OF TRAP
3985 016162 016000 016170  MOV      $TRPAD(RO),RO ;INDEX TO TABLE
3986 016166 000200          RTS      RO             ;GO TO ROUTINE

```

.SBTTL TRAP TABLE

;\*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED  
;\*BY THE "TRAP" INSTRUCTION.

ROUTINE  
-----

```

3996 016170          $TRPAD:
3997 016170 015016          $TYPE          ;CALL=TYPE          TRAP+0(104400) TTY TYPEOUT ROUTINE
3998 016172 015322          $TYPOC         ;CALL=TYPOC         TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING
3999 016174 015276          $TYPOS         ;CALL=TYPOS         TRAP+4(104404) TYPE OCTAL NUMBER (NO LEADING ZE
4000 016176 015336          $TYPON         ;CALL=TYPON         TRAP+6(104406) TYPE OCTAL NUMBER (AS PER LAST C
4001 016200 015124          $RDCHR         ;CALL=RDCHR         TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
4002 016202 015160          $RDLIN         ;CALL=RDLIN         TRAP+12(104412) TTY TYPEIN STRING ROUTINE
4003 016204 016046          $RDOCT         ;CALL=RDOCT         TRAP+14(104414) READ AN OCTAL NUMBER FROM TTY
4004 016206 005015 047522 020115 EM1:  .ASCIZ <15><12>/ROM READ DATA COMPARISON ERROR./
      016250 005015 051127 052111 EM2:  .ASCIZ <15><12>/WRITTING ROM FAILED TO TRAP./
      016307 015 052412 042516 EM3:  .ASCIZ <15><12>/UNEXPECTED TRAP WHILE READING ROM./
      016354 005015 040506 040524 EM4:  .ASCIZ <15><12>/FATAL TRAP. ROM PC ON STACK./
      016414 005015 041520 020040 DH1:  .ASCII <15><12>/PC SOFT ROM/
      016441 015 040412 042134          .ASCIZ <15><12>/ADDRESS ADDRESS ADDRESS EXPECTED FOUND /
      016513 015 050012 004503 DH2:  .ASCII <15><12>/PC ROM/
      016523 015 040412 042104          .ASCIZ <15><12>/ADDRESS ADDRESS/
      016545 015 050012 020103 DH3:  .ASCII <15><12>/PC OF PROGRAM /
      016567 015 052012 040522          .ASCIZ <15><12>/TRAP ADDRESS/

      016612          .EVEN
4005 016612 001116 001124 001126 DT1:  .WORD  $ERRPC, $GDDAT, $BDDAT, TEMP3, TEMP4, 0
4006 016620 001224 001226 000000
4007
4008 016626 001116 001226 000000 DT2:  .WORD  $ERRPC, TEMP4, 0
4009 016634 001116 014734 000000 DT3:  .WORD  $ERRPC, XSTORE, 0
4010 016702          .=. +40
4011
4012 000001          CORMAX:
      .END

```

















# F09

MAR 1975  
CZBMOF.P11

MACY11 27(657) 1-OCT-75 09:30 PAGE 100  
CROSS REFERENCE TABLE

SEQ 0108

ADD	3201	3626	3695	3785	3795	3910	3962								
ASL	3907	3938	3909	3955	3957	3959									
BEO	3130	3138	3144	3146	3175	3177	3180	3237	3243	3272	3303	3306	3311	3385	3391
	3394	3427	3434	3439	3441	3469	3471	3474	3562	3569	3586	3597	3635	3643	3812
	3873	3912	3917	3930	3954										
BGT	3819														
BIC	3725	3809	3961												
BIS	3814	3815													
BISB	3899														
BIT	3132	3143	3169	3174	3242	3390	3429	3438	3464	3468	3634	3879			
BLO	3328														
BLOS	3737														
BLT	3703	3820													
BMI	3844														
BNE	2939	2958	2966	2982	2990	2994	3008	3016	3026	3033	3039	3045	3051	3057	3063
	3070	3078	3081	3084	3087	3120	3123	3133	3135	3159	3162	3170	3172	3185	3211
	3227	3239	3274	3308	3387	3417	3420	3430	3432	3454	3457	3465	3467	3479	3502
	3543	3575	3577	3627	3651	3692	3699	3741	3747	3810	3880	3900	3922		
BPL	3686	3707	3723	3808	3884										
BR	2972	2996	2999	3072	3099	3141	3150	3178	3183	3240	3275	3313	3388	3399	3437
	3445	3472	3477	3688	3705	3743	3786	3801	3822	3846	3852	3855	3905	3932	3963
CLC	3540	3572													
CLR	2937	2948	2949	2951	2952	2955	2956	3276	3498	3519	3549	3579	3589	3611	3625
	3529	3639	3645	3799	3898	3951	3952								
CLRB	3748	3859													
CMP	2938	2965	2973	2981	2989	2993	2997	3007	3015	3032	3038	3044	3050	3056	3062
	3077	3080	3083	3086	3119	3121	3122	3124	3129	3137	3140	3145	3158	3161	3176
	3179	3210	3226	3236	3271	3302	3305	3307	3310	3327	3384	3393	3416	3418	3419
	3421	3426	3433	3436	3440	3453	3456	3470	3473	3542	3561	3568	3574	3585	3596
	3650	3736	3853												
CMPB	3698	3740	3746												
COM	3027	3503													
DEC	3134	3171	3184	3238	3273	3386	3431	3466	3478	3576	3906				
DECB	3702	3807	3818												
EMT	20														
HALT	10	3024	3098	3277	3618	3621	3667	3885							
INC	3539	3813	3821	3875											
INCB	3857	3872													
IOT	21														
JMP	14	2959	2979	2987	3005	3013	3021	3079	3082	3085	3088	3090	3187	3188	3278
	3309	3335	3337	3346	3400	3481	3482	3544	3600	3630	3640	3652			
JSR	3186	3480	3599	3654	3697	3704	3881								
MOV	2936	2940	2941	2942	2943	2944	2945	2946	2947	2950	2953	2954	2960	2961	2964
	2967	2968	2971	2974	2975	2976	2977	2978	2983	2984	2985	2986	2991	2992	2995
	2998	3001	3002	3003	3004	3009	3010	3011	3012	3017	3018	3019	3020	3030	3031
	3034	3035	3036	3040	3041	3042	3046	3047	3048	3052	3053	3054	3058	3059	3060
	3064	3065	3066	3068	3075	3076	3111	3112	3113	3114	3115	3116	3117	3118	3125
	3126	3127	3128	3147	3148	3154	3155	3156	3157	3164	3166	3168	3198	3199	3200
	3202	3203	3206	3207	3208	3209	3212	3216	3221	3223	3224	3225	3228	3233	3244
	3245	3248	3249	3250	3251	3252	3254	3258	3261	3262	3263	3267	3270	3295	3296
	3297	3300	3301	3315	3317	3318	3324	3329	3331	3334	3345	3354	3355	3358	3359
	3360	3364	3369	3374	3375	3378	3382	3397	3398	3409	3410	3411	3412	3413	3414
	3415	3422	3423	3424	3425	3442	3443	3449	3450	3451	3452	3459	3461	3463	3497
	3499	3500	3504	3505	3506	3507	3511	3518	3520	3522	3523	3524	3525	3528	3530



	3532	3533	3536	3546	3547	3548	3550	3552	3553	3554	3555	3556	3557	3558	3559
	3560	3564	3565	3566	3567	3578	3580	3581	3582	3583	3584	3588	3590	3591	3592
	3593	3594	3612	3613	3616	3617	3620	3623	3624	3633	3636	3638	3648	3689	3690
	3694	3700	3720	3721	3734	3735	3750	3751	3752	3753	3782	3790	3791	3792	3798
	3805	3823	3824	3825	3826	3827	3848	3849	3851	3854	3858	3860	3861	3874	3876
	3897	3902	3911	3916	3921	3923	3927	3944	3945	3946	3947	3948	3950	3965	3966
	3967	3968	3969	3981	3982	3985									
MOV8	3691	3708	3724	3739	3744	3783	3784	3787	3788	3789	3793	3796	3797	3816	3878
	3953	3984													
NEG	3794														
NOP	2963	2970	3165	3460	3521	3551	3615	3655	3656						
ROL	3541	3573	3800	3802	3803	3804	3806	3843	3956	3958	3960				
RTI	3642	3696	3726	3754	3828	3862	3886	3970							
RTS	3657	3709	3925	3986											
SUB	3877														
TRAP	3988	3998	3999	4000	4001	4002	4003								
TST	2957	2962	2969	3025	3069	3149	3160	3163	3182	3213	3229	3326	3361	3376	3383
	3392	3444	3455	3458	3476	3501	3571	3614	3619	3693	3811	3850	3883	3929	3964
	3983														
TSTB	3685	3706	3722												
.ASCII	146	147	3100	3191	3279	3609	4004								
.ASCIZ	148	3100	3191	3279	3607	3609	3659	3933	4004						
.BLKB	3757														
.BLKM	2925	2926	2927	2928	2929	2930									
.BYTE	124	125	130	131	142	143	144	145	678	679	680	681	682	683	3755
	3756	3829	3830	3831	3832										
.ENABL	1														
.END	4012														
.ENDC	15	20	96	110	111	121	146	149	150	208	2936	2940	2941	2943	2945
	2947	2975	2983	2991	3001	3009	3017	3664	3711	3727	3728	3735	3737	3742	3757
	3758	3759	3835	3841	3843	3845	3856	3857	3859	3863	3864	3870	3872	3876	3882
	3883	3887	3888	3906	3935	3936	3972	3973	3982	3985	3997	3998	3999	4000	4001
	4002	4003	4004	4012											
.EQUIV	20	21	23	38	39	58	59	60	61	62	63	64	65	66	67
	86	87	88	89	90	91	92	93	94	95					
.EVEN	3100	3191	3279	3607	3609	3661	3934	4004							
.IF	10	18	68	96	110	120	146	149	204	2936	2940	2941	2943	2945	2947
	2967	2975	2983	2991	3009	3017	3663	3710	3727	3735	3736	3741	3757	3758	3834
	3840	3843	3855	3857	3859	3863	3869	3872	3875	3879	3881	3882	3883	3886	3887
	3905	3921	3935	3951	3972	3981	3985	3988	3998	3999	4000	4001	4002	4003	4004
	4012														
.IFF	18	111	120	146	150	2940	3664	3711	3727	3728	3737	3741	3758	3759	3835
	3855	3857	3863	3864	3869	3876	3886	3887	3888	3906	3935	3936	3973	3982	
.IFT	3857	3882	3955	3971	3972										
.IFTF	3857	3881	3951	3955	3971										
.IIF	10	149	203	208	213	214	2941	2943	2947	3710	3719	3750	3758	3838	3839
	3859	3863	3867	3868	3869	3887	3903	3928	3997	3998	3999	4000	4001	4002	4003
.IRP	2936	3946	3967												
.LIST	1	10	110	146	216	2925	2936	3100	3191	3279	3607	3609	3727	3988	3997
	3998	3999	4000	4001	4002	4003	4004	4012							
.MACRO	110	215	3988												
.MCALL	1	2	110	3662											
.NLIST	1	10	110	146	216	2925	2936	3100	3191	3279	3607	3609	3727	3988	3997
	3998	3999	4000	4001	4002	4003	4004	4012							

MAR 1975  
DZBMDF.P11

MACY11 27(657) 1-OCT-75 09:30 PAGE 102  
CROSS REFERENCE TABLE

SEQ 0110

.PAGE	110	149	403	775	1246	1297	1344	1407	1452	1503	1573	1627	1680	1734	1781
	1833	1886	2931	3151	3446	3607									
.REM	1687	1737													
.REPT	10														
.SBTTL	1	4	11	16	112	151	3665	3712	3760	3836	3865	3889	3937	3974	3989
.TITLE	203														
.WORD	10	123	126	127	128	129	132	133	134	135	136	137	3914	3919	3971
	4005	4008	4009												

ERRORS DETECTED: 0

\*DZBMDF DZBMDF/SOL/CRF+DZBMDF.P11  
RUN-TIME: 41 31 3 SECONDS  
CORE USED: 16K

