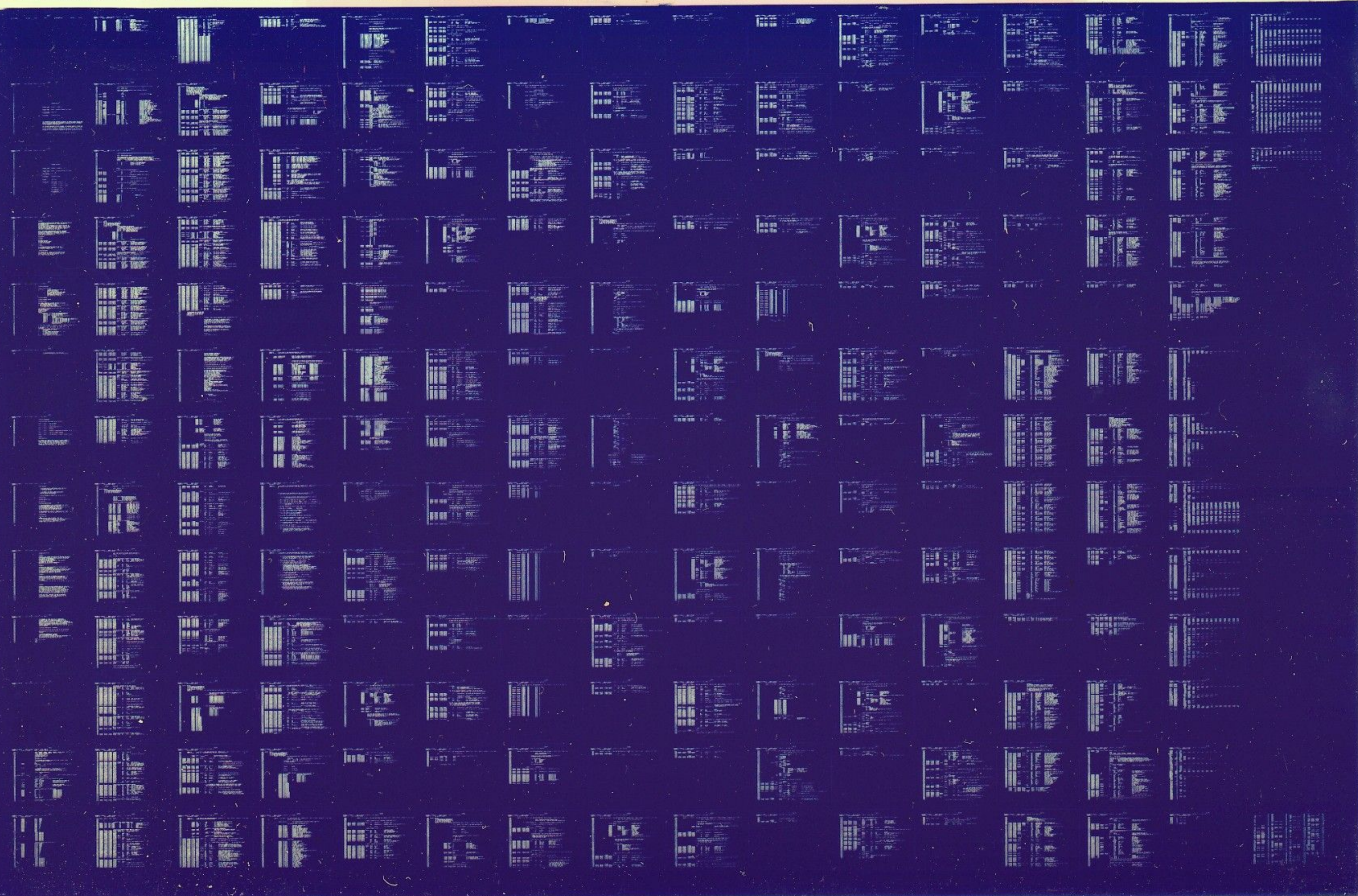


BM873

RESTART ROM LOADER
MD-11-DZBMD-J

EP-DZBMD-J-DL-C JUN 1977
COPYRIGHT 1977 **digital**
FICHE 1 OF 1 MADE IN USA



B01

EOF1DVOVASEG

00010000

770629

POP10 411

DVHOR1DZBMOJSEG

00010000

770629

53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98

PROGRAM HISTORY

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

PRODUCT NAME: BM873 - 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
RICH MURATORI	10/76	YG
RICH MURATORI	10/76	YH
FITZCARL JOHNSON		
ED RYAN	3/77	YJ

99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138

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 OR DISKETTE BY THE USE OF ROM BOOT LOADER.

139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192

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 BM873Y-B,C,D,F,G, H OR J, 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.
(11/34 PRESS CNTRL START SIMULTANEOUSLY)

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

MAINDEC-11-DZBMD-J
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))
BM873-YG KL10 (PDP-11) 256 BOOTSTRAP ROM
BM873-YH KL10 (PDP-11) 256 BOOTSTRAP ROM
BM873-YJ DECSYSTEM20 SECONDARY FRONT END (PDP-1134A)
256 BOOTSTRAP ROM

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 BM873Y-B,C,D,F,G,H OR J)

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

GO1

MAINDEC-11-DZBMD-J
DZBMD.P11

MACY11 27(663) 2-MAY-77 11:46 PAGE 5

193
194

IS YOUR RESPONSIBILITY. THAT DATA WAS
PLACED INTO SOFTWARE TABLE FOR TEST COMPARISON.

195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223

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 ... RESERVED

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 .

224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263

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 BM873Y-B,C,D,F,G,H OR J)
- 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 BM873Y-B,C,D,F,G,H OR J)

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

264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315

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 BMB73Y-B,C,D,F,G,H OR J 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 BMB73Y-B,C,D,F,G,H OR J 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.

K01

MAINDEC-11-DZBMD-J
DZBMD.P11

MACY11 27(663) 2-MAY-77 11:46 PAGE 9

316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341

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
BMB73YA OR YB, YC, YD, YF, YG, YH OR YJ BY COMPARING THE 1ST
WORD IN ROM WITH THE EXPECTED WORD. THE DIAGNOSTIC THEN RUNS
PROGRAM 1 AND PROGRAM 4 BEFORE ENTERING THE MONITOR.
(FOR ROM VERSIONS THAT HAVE THE SAME FIRST WORD AND ADDITIONAL
WORD IS CHECKED)

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 BMB73 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

MAINDEC-11-DZBMD-J
DZBMD.P11

MACY11 27(663) 2-MAY-77 11:46 PAGE 10

L01

342

%

MO1

MAINDEC-11-DZBMD-J MACY11 27(663)
 DZBMD.P11 MAINDEC-11-DZBMD-J

2-MAY-77 11:46 PAGE 11
 BM873 UNIVERSAL RESTART LOADER DIAGNOSTIC.

343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396

```
.MCALL .HEADER, SWRHI, SWRLO, EQUATE, SETUP, STRAP, SCATCH, SCMTAG
.MCALL .RDLIN, $SCOPE, $ERROR, $ERRTYP, $RDOCT
.TITLE MAINDEC-11-DZBMD-J
;*COPYRIGHT (C) 1977
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;*
;*PROGRAM BY E. RYAN
;*
;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
;*PACKAGE (MAINDEC-11-DZQAC-A1).
;*
$TN=1
$SWR=160000 ;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYP0UT

.SBTTL TRAP CATCHER

      =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)
      =200
000200 000137 012000      JMP      @#RESTRT      ;JUMP TO STARTING ADDRESS OF PROGRAM

.SBTTL BASIC DEFINITIONS
;*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
PS= 177776 ;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMT= 177774 ;STACK LIMIT REGISTER
PIRQ= 177772 ;PROGRAM INTERRUPT REQUEST REGISTER
SWR= 177570 ;SWITCH REGISTER
DISPLAY=SWR

;*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0 ;GENERAL REGISTER
R1= %1 ;GENERAL REGISTER
R2= %2 ;GENERAL REGISTER
R3= %3 ;GENERAL REGISTER
R4= %4 ;GENERAL REGISTER
R5= %5 ;GENERAL REGISTER
R6= %6 ;GENERAL REGISTER
R7= %7 ;GENERAL REGISTER
.EQUIV R6,SP ;STACK POINTER
.EQUIV R7,PC ;PROGRAM COUNTER

;*"SWITCH REGISTER" SWITCH DEFINITIONS
$W15= 100000
```

000001
160000

000000

000200

000200 000137 012000

001100

177776

177774

177772

177570

177570

000000

000001

000002

000003

000004

000005

000006

000007

100000

397 040000
 398 020000
 399 010000
 400 004000
 401 002000
 402 001000
 403 000400
 404 000200
 405 000100
 406 000040
 407 000020
 408 000010
 409 000004
 410 000002
 411 000001
 412
 413
 414
 415
 416
 417
 418
 419
 420
 421
 422
 423
 424 100000
 425 040000
 426 020000
 427 010000
 428 004000
 429 002000
 430 001000
 431 000400
 432 000200
 433 000100
 434 000040
 435 000020
 436 000010
 437 000004
 438 000002
 439 000001
 440
 441
 442
 443
 444
 445
 446
 447
 448
 449
 450

SW14= 40000
 SW13= 20000
 SW12= 10000
 SW11= 4000
 SW10= 2000
 SW09= 1000
 SW08= 400
 SW07= 200
 SW06= 100
 SW05= 40
 SW04= 20
 SW03= 10
 SW02= 4
 SW01= 2
 SW00= 1
 .EQUIV SW09, SW9
 .EQUIV SW08, SW8
 .EQUIV SW07, SW7
 .EQUIV SW06, SW6
 .EQUIV SW05, SW5
 .EQUIV SW04, SW4
 .EQUIV SW03, SW3
 .EQUIV SW02, SW2
 .EQUIV SW01, SW1
 .EQUIV SW00, SW0

.*DATA BIT DEFINITIONS (BIT00 TO BIT15)

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


```

464 ;*****
465
466 .SBTTL COMMON TAGS
467
468 ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
469 ;*USED IN THE PROGRAM.
470
471 000046 000046      .=46
472 000046 020002      $ENDAD                      ;LOGICAL END OF PROGRAM
473
474 000052 000052      .=52
475 000052 000000      .WORD 0
476
477 001100 001100      .=1100
478
479 001100 000000      $CMTAG:                      ;START OF COMMON TAGS
480 001100 000000      $PASS: .WORD 0                ;CONTAINS PASS COUNT
481 001102 000      $TSTNM: .BYTE 00                ;CONTAINS THE TEST NUMBER
482 001103 000      $ERFLG: .BYTE 00                ;CONTAINS ERROR FLAG
483 001104 000000      $ICNT: .WORD 00              ;CONTAINS SUBTEST ITERATION COUNT
484 001106 000000      $LPADR: .WORD 00             ;CONTAINS SCOPE LOOP
485 001110 000000      $LPERR: .WORD 00            ;CONTAINS SCOPE RETURN FOR ERRORS
486 001112 000000      $ERTTL: .WORD 00           ;CONTAINS TOTAL ERRORS DETECTED
487 001114 000      $ITEMB: .BYTE 00              ;CONTAINS ITEM CONTROL BYTE
488 001115 001      $ERMAX: .BYTE 1                ;CONTAINS MAX. ERRORS PER TEST
489 001116 000000      $ERRPC: .WORD 00           ;CONTAINS PC OF LAST ERROR INSTRUCTION
490 001120 000000      $GDADR: .WORD 00           ;CONTAINS OF 'GOOD' DATA
491 001122 000000      $BDADR: .WORD 00           ;CONTAINS OF 'BAD' DATA
492 001124 000000      $GDOAT: .WORD 00          ;CONTAINS 'GOOD' DATA
493 001126 000000      $BDOAT: .WORD 0           ;CONTAINS 'BAD' DATA
494 001130 000000 000000 000000      $RESV: .WORD 0,0,0 ;RESERVED--NOT TO BE USED
495 001136 177560      $TKS: 177560                ;TTY KBD STATUS
496 001140 177562      $TKB: 177562                ;TTY KBD BUFFER
497 001142 177564      $TPS: 177564                ;TTY PRINTER STATUS REG.
498 001144 177566      $TPB: 177566                ;TTY PRINTER BUFFER REG.
499 001146 000      $NULL: .BYTE 0                 ;CONTAINS NULL CHARACTER FOR FILLS
500 001147 002      $FILLS: .BYTE 2                ;CONTAINS # OF FILLER CHARACTERS REQUIRED
501 001150 012      $FILLC: .BYTE 12              ;INSERT FILL CHARS. AFTER A "LINE FEED"
502 001151 000      $TPFLG: .BYTE 0               ;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
503 001152 077      $QUES: .ASCII /?/            ;QUESTION MARK
504 001153 015      $CRLF: .ASCII <15>           ;CARRIAGE RETURN
505 001154 000012      $LF: .ASCIZ <12>          ;LINE FEED

```


506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522 001156
523
524
525
526 001156 021216
527 001160 021416
528 001162 021614
529 001164 000000
530
531
532
533 001166 021260
534 001170 021515
535 001172 021630
536 001174 000000
537
538
539
540 001176 021316
541 001200 021547
542 001202 021636
543 001204 000000
544
545
546 001206 021356
547 001210 021515
548 001212 021630
549 001214 000000
550
551 001216 000000
552 001220 000000
553 001222 000000
554 001224 000000
555 001226 000000
556 001230 000000
557 001232 000000
558 001234 000000
559 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 SITEMB, THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;#NOTE1:  IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
;#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
SAVR0:  0
SAVR1:  0
SAVR4:  0
SAVR5:  0

```

```

560
561      . =1400
562 001400 MAP.YA:
563      ; THE FOLLOWING IS A REPRODUCTION
564      ; OF THE ROM PROGRAM FOR BM873YA.
565      ; IT IS HERE FOR COMPARISON TO
566      ; ACTUAL ROM AND FOR REFERENCE.
567      ; 173000 . =173000
568      ; STARTING ADDRESS FOR BOOTSTRAP
569      ; THIS LOADER IS DESIGNED FOR THE RESTART MODULE M873.
570      ; IT FUNCTIONALLY REPLACES THE FOLLOWING ROMS:
571      ; M792-YA - PAPER TAPE BOOTSTRAP FOR PC11, KL11
572      ; MR11-DB BULK STORAGE BOOTSTRAP ROM
573      ; M792-YH TAI1 CASSETTE BOOTSTRAP ROM
574      ; REGISTER DEFINITIONS
575      ;
576      ;
577      ;
578      ;
579      ;
580      ;
581      ;
582      ;
583      ;
584      ;
585      ;
586      ;
587      ;
588      ;
589      ;
590      ;
591      ;
592      ;
593      ;
594      ;
595      ;
596      ;
597      ;
598      ;
599      ;
600      ;
601      ;
602      ;
603      ;
604      ;
605      ;
606      ;
607      ;
608      ;
609      ;
610      ;
611      ;
612      ;
613      ;

```

001400	010702	:173000	010702	RF11: MOV PC,R2	; SET POINTER TO PARAMETER LISTS
001402	000464	:173002	000464	BR OTHER	; TRANSFER TO SERVICE ROUTINE
001404	177462	:173004	177462	.WORD 177462	; DEVICE WORD COUNT ADDRESS
001406	000005	:173006	000005	.WORD 5	; DEVICE READ INSTRUCTION
001410	010702	:173010	010702	RK11: MOV PC,R2	; SET POINTER TO PARAMETER LIST
001412	000460	:173012	000460	BR OTHER	; TRANSFER TO SERVICE ROUTINE
001414	177406	:173014	177406	.WORD 177406	; DEVICE WORD COUNT REGISTER
001416	000005	:173016	000005	.WORD 5	; DEVICE READ INSTRUCTION
001420	013707	:173020	013707	TRANSR: MOV @SR,PC	; GO TO INDICATED LOCATION
001422	177570	:173022	177570		
					ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
001424	173000	:173024	173000	POWER: .WORD RF11	; ADDRESS OF FIRST LOCATION IN ROM
001426	000340	:173026	000340	.WORD 340	; PROCESSOR STATUS LEVEL 7
001430	010702	:173030	010702	TC11: MOV PC,R2	; SET UP POINTER TO PARAMETER LIST
001432	000426	:173032	000426	BR TAPES	; AND TRANSFER TO FIRST ROUTINE
001434	177344	:173034	177344	.WORD 177344	; DEVICE WORD COUNT ADDRESS
001436	004003	:173036	004003	.WORD 4003	; FIND PREVIOUS BLOCK COMMAND
001440	100000	:173040	100000	.WORD 100000	; USED AS DONE INDICATOR
001442	024000	:173042	024000	.WORD 24000	; USED AS ERROR INDICATOR/TEST FLAG
001444	000445	:173044	000445	BR OTHERX	; THEN TRANSFER TO NEXT ROUTINE
001446	000005	:173046	000005	.WORD 5	; DEVICE READ COMMAND

614						
615					;	THIS IS THE START LOCATION FOR TM11 MAGTAPE CONTROLLER
616	001450	010702	:173050	010702	TM11: MOV PC,R2	;SET POINTER TO PARAMETER LIST
617	001452	000416	:173052	000416	BR TAPES	;AND TRANSFER TO FIRST ROUTINE
618	001454	172524	:173054	172524	.WORD 172524	;DEVICE BYTE/RECORD COUNT REGISTER
619	001456	060017	:173056	060017	.WORD 60017	;DEVICE REWIND COMMAND
620	001460	000200	:173060	000200	.WORD 200	;DEVICE DONE FLAG
621	001462	100000	:173062	100000	.WORD 100000	;DEVICE ERROR FLAG BIT
622	001464	000413	:173064	000413	BR TAPESX	;THEN TRANSFER TO NEXT SERVICE RTN
623	001466	060011	:173066	060011	.WORD 60011	;DEVICE FORWARD SPACE COMMAND
624	001470	000200	:173070	000200	.WORD 200	;SAME AS ABOVE
625	001472	100000	:173072	100000	.WORD 100000	;SAME AS ABOVE
626	001474	000431	:173074	000431	BR OTHERX	;THEN TRANSFER TO READ/TRANSFER ROUTINE
627	001476	060003	:173076	060003	.WORD 60003	;DEVICE READ COMMAND
628						
629					;	THIS IS THE START LOCATION FOR THE RP11 CONTROLLER
630	001500	010702	:173100	010702	RP11: MOV PC,R2	;SET POINTER TO PARAMETER LIST
631	001502	000424	:173102	000424	BR OTHER	;TRANSFER TO TRANSFER ROUTINE
632	001504	176716	:173104	176716	.WORD 176716	;DEVICE WORD COUNT REGISTER
633	001506	000005	:173106	000005	.WORD 5	;DEVICE READ COMMAND
634						
635					;	THIS IS THE TAPE DEVICE SERVICE ROUTINE.
636	001510	010200	:173110	010200	TAPES: MOV R2,R0	;GET ADDRESS OF PARAMETER LIST
637	001512	005720	:173112	005720	TST (R0)+	;SKIP TWO WORDS FIRST TIME
638	001514	000005	:173114	000005	TAPESX: RESET	;RESET ALL DEVICES
639	001516	005720	:173116	005720	TST (R0)+	;SKIP OVER BRANCH INSTRUCTION
640	001520	016201	:173120	016201	MOV 2(R2),R1	;THEN GET DEVICE WORD/BYTE COUNT ADDRES
641	001522	000002	:173122	000002		
642	001524	005311	:173124	005311	DEC 2R1	;AND SET TO -1
643	001526	012041	:173126	012041	MOV (R0)+,-(R1)	;AND THEN ISSUE COMMAND TO DEVICE
644	001530	031011	:173130	031011	TAPWAT: BIT 2R0,2R1	;WAIT FOR DEVICE COMPLETION
645	001532	001776	:173132	001776	BEQ TAPWAT	;BY HANGING IN LOOP
646	001534	005720	:173134	005720	TST (R0)+	;AND THEN SKIP DONE FLAG
647	001536	032041	:173136	032041	BIT (R0)+,-(R1)	;THEN TEST FOR ERROR
648	001540	001063	:173140	001063	BNE ERROR	;THERE IS ONE
649	001542	000110	:173142	000110	RETURN: JMP 2R0	;AND TRANSFER TO FOLLOWING INSTRUCTION
650						
651					;	THIS IS THE STARTING ADDRESS FOR RC11 DISK CONTROLLERS
652	001544	010702	:173144	010702	RC11: MOV PC,R2	;SET UP POINTER TO PARAMETER LIST
653	001546	000402	:173146	000402	BR OTHER	;TRANSFER TO SERVICE RTN
654	001550	177450	:173150	177450	.WORD 177450	;DEVICE WORD COUNT REGISTER
655	001552	000005	:173152	000005	.WORD 5	;DEVICE READ INSTRUCTION
656						
657					;	THIS ROUTINE PERFORMS THE ACTUAL TRANSFER TO MEMORY OF DATA
658	001554	010200	:173154	010200	OTHER: MOV R2,R0	;SET POINTER TO LIST IN R0
659	001556	005720	:173156	005720	TST (R0)+	;SKIP TWO WORDS FIRST TIME.
660	001560	005720	:173160	005720	OTHERX: TST (R0)+	;SKIP PAST BR INSTRUCTION
661	001562	000005	:173162	000005	RESET	;REST THE WORLD
662	001564	016201	:173164	016201	MOV 2(R2),R1	;OBTAIN DEVICE WORD COUNT ADDRESS
663	001566	000002	:173166	000002		
664	001570	012711	:173170	012711	MOV 8-1000,2R1	;THEN OBTAIN LARGE WORD COUNT
665	001572	177000	:173172	177000		
666	001574	011041	:173174	011041	OTHWAT: MOV 2R0,-(R1)	;AND PUT COMMAND TO DEVICE
667	001576	105711	:173176	105711	TSTB 2R1	;WAIT FOR DONE FLAG

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 18
 DZBMD.P11 ROM CONTENTS TABLES

```

668 001600 100376 ;173200 100376      BPL OTHWAT      ;BY HANGING IN LOOP
669 001602 005711 ;173202 005711      TST R1         ;THEN TEST FOR ERROR
670 001604 100441 ;173204 100441      BMI ERROR      ;GOT PROBLEMS
671 001606 005007 ;173206 005007      CLR PC         ;AND TRANSFER TO ZERO
672                                     ;
673                                     ;THIS IS THE STARTING ADDRESS FOR THE PC11 PAPER TAPE CONTROLLER
674 001610 012704 ;173210 012704      KL11: MOV #177560,R4 ;OBTAIN DEVICE ADDRESS
675 001612 177560 ;173212 177560
676 001614 000440 ;173214 000440      BR CKDEV       ;AND TRANSFER TO READER SERVICE ROUTINE
677                                     ;
678                                     ;THIS IS THE CASSETTE DEVICE COMMAND TABLE
679 001616 017640 ;173216      240      TABLE: .BYTE 240      ;COMPARE WORD NOT A COMMAND
680                                     ;173217      037      .BYTE 37          ;ILBS+RWD+GO
681 001620 002415 ;173220      015      .BYTE 15          ;SPACE FORWARD BLOCK+GO
682                                     ;173221      005      .BYTE 5           ;READ+GO
683 001622 112024 ;173222      024      .BYTE 24          ;READ+ILBS
684                                     ;173223      224      .BYTE 224         ;READ+ILBS+END FLAG
685                                     ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
686                                     ;
687                                     ;THIS IS AN ADDITIONAL POWER VECTOR ADDRESS REQUIRED BY DEVICE
688 001624 173000 ;173224 173000      POWER2: .WORD RF11 ;ADDRESS OF BEGINNING OF BOOTSTRAP
689 001626 000340 ;173226 000340      .WORD 340        ;PRIORITY LEVEL 7
690                                     ;
691                                     ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE DEVICE #0
692 001630 005004 ;173230 005004      CBOOT: CLR R4     ;LOAD DEVICE NUMBER 0 IN R4
693 001632 012700 ;173232 012700      RESTX: MOV #177500,R0 ;GET DEVICE ADDRESS
694 001634 177500 ;173234 177500
695 001636 000005 ;173236 000005      RESTRT: RESET     ;ISSUE RESET INSTRUCTION
696 001640 010410 ;173240 010410      MOV R4,R0        ;LOAD DEVICE WITH UNIT NUMBER
697 001642 012701 ;173242 012701      MOV #TABLE,R1    ;GET FUNNY TABLE OF INSTRUCTIONS
698 001644 173216 ;173244 173216
699 001646 012702 ;173246 012702      MOV #375,R2      ;AND LOAD UP TRANSFER COUNTER
700 001650 000375 ;173250 000375
701 001652 112103 ;173252 112103      MOVB (R1)+,R3    ;THE LOAD UP COMPARATOR
702 001654 112110 ;173254 112110      LOOP1: MOVB (R1)+,R0 ;LOAD DEVICE REGISTER WITH COMMAND
703 001656 100407 ;173256 100407      BMI DONE
704 001660 130310 ;173260 130310      LOOP2: BITB R3,R0 ;HAS COMMAND COMPLETED
705 001662 001776 ;173262 001776      BEQ LOOP2        ;NO, WAIT
706 001664 105202 ;173264 105202      INCB R2          ;THEN INCREMENT ADDRESS CTR
707 001666 100772 ;173266 100772      BMI LOOP1        ;IF NEGATIVE, GET COMMAND
708 001670 116012 ;173270 116012      MOVB 2(R0),R2    ;AND STORE DATA AWAY
709 001672 000002 ;173272 000002
710 001674 000771 ;173274 000771
711 001676 005710 ;173276 005710      DONE: BR LOOP2   ;GO GET ANOTHER BYTE
712 001700 100756 ;173300 100756      TST R0           ;ANY DEVICE ERRORS
713 001702 005002 ;173302 005002      BMI RESTRT       ;YES, RETRY
714 001704 120312 ;173304 120312      CLR R2           ;CLEAR COMPARE ADDRESS AND TRANSFER ADDRESS
715 001706 001377 ;173306 001377      CMPB R3,R2       ;IT MUST BE 240
716 001710 000112 ;173310 000112      BNE +0           ;NO, THERE WAS AN ERROR
717                                     ERROR: JMP R2      ;NORMAL CASSETTE AND ERROR FOR BULK STORAGE
718                                     ;
719                                     ;THIS IS THE STARTING LOCATION FOR THE PC11 CONTROLLER
720 001712 012704 ;173312 012704      PC11: MOV #177550,R4 ;LOAD DEVICE ADDRESS
721 001714 177550 ;173314 177550

```

722	001716	000005	::173316	000005	CKDEV: RESET	;KILL ALL DEVICE ACTION
723	001720	012701	::173320	012701	MOV #160000,R1	;THEN SET UP MEMORY TEST LIMITS
724	001722	160000	::173322	160000		
725	001724	012702	::173324	012702	MOV #6,R2	;AND SET UP POINTER TO TIMEOUT LOCATION
726	001726	000006	::173326	000006		
727	001730	012712	::173330	012712	MOV #340,R2	;AND SET UP VECTOR TO RETURN TO NEXT
728	001732	000340	::173332	000340		
729	001734	010742	::173334	010742	MOV PC, -(R2)	;SAVE THE PC
730	001736	012706	::173336	012706	MOV #24,SP	;AND LOAD UP STACK POINTER
731	001740	000024	::173340	000024		
732	001742	010441	::173342	010441	MOV R4, -(R1)	;AND LOOK FOR END OF MEMORY
733	001744	040601	::173344	040601	BIC SP,R1	;THEN DROP TO XX7752
734	001746	010111	::173346	010111	MOV R1,R1	;AND STORE IN ITSELF
735	001750	011102	::173350	011102	MOV R1,R2	;THEN LOAD ADDRESS FOR DATA INSERTION
736	001752	005214	::173352	005214	INC R4	;AND START DEVICE
737	001754	105714	::173354	105714	RDRWAT: TSTB R4	;THEN WAIT FOR CHARACTER AVAILABLE
738	001756	100376	::173356	100376	BPL RDRWAT	;HANGING THERE IF NECESSARY
739	001760	116412	::173360	116412	MOVB 2(R4),R2	;STORE AWAY DATA BYTE
740	001762	000002	::173362	000002		
741	001764	005211	::173364	005211	INC R1	
742	001766	120227	::173366	120227	CMPB R2,#375	;HAS BRANCH OFFSET BEEN STORED
743	001770	000375	::173370	000375		
744	001772	001366	::173372	001366	BNE LOOP	;NO
745	001774	105222	::173374	105222	INCB (R2)+	;YES, ALL DONE
746	001776	END.YA:				
747	001776	000142	::173376	000142	JMP -(R2)	;THEN TRANSFER TO RTN

748 : BMB738 BOOTSTRAP MACY11 27(655) 1-OCT-74 14:50 PAGE 1

749 :
 750 : ;DATE: AUG 23, 1974

002000

MAP.YB:
 : THE FOLLOWING IS A REPRODUCTION
 : OF THE ROM PROGRAM FOR BMB73YB.
 : 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 TAI1 CASSETTE BOOTSTRAP ROM
 : RMB73A COMBINATION OF ABOVE ROMS

: PREPHERIAL EXTERNAL PAGE REGISTERS ASSIGNMENTS:

767	177462	RFMC=	177462	:	WORD COUNT REG. FOR RF1
768	177406	RKMC=	177406	:	WORD COUNT REG. FOR RK1
769	177344	TCMC=	177344	:	WORD COUNT REG. FOR TC1
770	172524	TMNC=	172524	:	BYTE/RECORD COUNT FOR T
771	176716	RPWC=	176716	:	WORD COUNT REG. FOR RP1
772	177450	RCMC=	177450	:	WORD COUNT REG. FOR RC1
773	177560	KLCS=	177560	:	CONTROL REG. FOR KL11
774	177500	TACS=	177500	:	CONTROL REG. FOR TAI1 C
775	177550	PCCS=	177550	:	CONTROL REG. FOR PC11
776	172440	TUCS=	172440	:	CONTROL STATUS REG. 1
777	172442	TUMC=	TUCS+2	:	TU16 WORD COUNT REG.

778					
779	176300	RHCSA=	176300	:	CONTROLLER REG. 1 FOR R
780	176302	RHWCA=	RHCSA+2		
781	172040	RSCSA=	172040	:	CONTROLLER REG.1 FOR RH
782	172042	RSWCA=	RSCSA+2		
783	176700	RPCSA=	176700	:	CONTROLLER REG. 1 FOR R
784	176702	RPWCA=	RPCSA+2		

: FUNCTION VALUE FOR PREPHERALS:

786	000005	RFREAD=	5	:	READ FUNCTION
787	004003	RNUM=	4003	:	REVERSE AND IDENTIFY BL
788	060017	TRMND=	60017	:	REWIND AND SET 800 BPI
789	060011	TMFRD=	60011	:	FORWARD RECORD COMMAND
790	060003	TTHREAD=	60003	:	TM11 READ
791	000011	DRCLR=	11	:	DRIVE CLEAR
792	000071	RHREAD=	71	:	RH11 READ COMMAND
793	000021	RHPRST=	21	:	READ IN PRESET
794	000031	TUSPAC=	31	:	SPACE FORWARD COMMAND F
795	040000	TUTAPE=	40000	:	TAPE BIT IN RH11/RHDT R
796	001300	TUMODE=	1300	:	800 BPI NORMAL MODE FOR
797	001000	FCE=	1000	:	FRAME COUNT ERROR BIT
798				:	CONSOLE SWITCH REG.
799	177570	CSW=	177570		

800
 801

```

802      ; ONLY THE LOW BYTE OF CONSOL SWITCH REGISTER IS
803      ; SELECT THE UNIT NUMBER OF THE DEVICE TO BOOT FR
804      ;
805      173000  .=173000
806      ;
807      ;
808      ; THIS IS THE STARTING ADDRESS FOR RH11/RS03/04 D
809      002000 000405 ; 173000 000405 RHRSA: BR 1$ ; ENTRY FOR SELECTING UNI
810      002002 010703 ; 173002 010703 RHRSB: MOV PC,R3 ; ENTRY TO SELECT UNITS
811      002004 113737 ; 173004 113737 MOVB 2#CSW,2#RSCSA+10;LOAD UNIT # INS
812      002006 177570 ; 173006 177570
813      002010 172050 ; 173010 172050
814      002012 000401 ; 173012 000401 BR 2$
815      002014 010703 ; 173014 010703 1$: MOV PC,R3
816      002016 012700 ; 173016 012700 2$: MOV #RSCSA,RD;SET CONTROL STATUS REG
817      002020 172040 ; 173020 172040
818      002022 000526 ; 173022 000526 BR RHCOMN
819      ;
820      ; THIS IS THE AUTO LOAD VECTOR
821      002024 173000 ; 173024 173000 .WORD RHRSA
822      002026 000340 ; 173026 000340 .WORD 340
823      ;
824      ; THIS IS THE STARTING ADDRESS FOR RK11 CONTROLLE
825      002030 000412 ; 173030 000412 RK11A: BR 2$ ; ENTRY TO SELECT UNIT 0
826      002032 010703 ; 173032 010703 RK11B: MOV PC,R3 ; ENTRY TO SELECT ALL UNI
827      ; SAVE ERROR RETRY ADDRES
828      002034 113705 ; 173034 113705 MOVB 2#CSW,R5;SET POINTER TO PARAMETE
829      002036 177570 ; 173036 177570
830      002040 052705 ; 173040 052705 BIS #10,R5 ;SET POSITION BIT
831      002042 000010 ; 173042 000010
832      002044 006105 ; 173044 006105 1$: ROL R5 ;SHIFT UNIT # TO BIT 13-
833      002046 103376 ; 173046 103376 BCC 1$ ;KEEP GOING
834      002050 010537 ; 173050 010537 MOV R5,2#RKWC+4;MOVE IN TO RKDA REGI
835      002052 177412 ; 173052 177412
836      002054 000401 ; 173054 000401 BR 3$ ;SKIP NEXT INSTRUCTION
837      002056 010703 ; 173056 010703 2$: MOV PC,R3 ;SAVE ERROR RETRY ADDRES
838      002060 010702 ; 173060 010702 3$: MOV PC,R2
839      002062 000546 ; 173062 000546 BR OTHERA
840      002064 177406 ; 173064 177406 .WORD RKWC
841      002066 000005 ; 173066 000005 .WORD RFREAD
842      ;
843      ; THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE)
844      002070 010703 ; 173070 010703 TC11: MOV PC,R3 ;SAVE ERROR RETRY ADDRES
845      002072 010702 ; 173072 010702 MOV PC,R2
846      002074 000570 ; 173074 000570 BR TAPES
847      002076 177344 ; 173076 177344 .WORD TCWC
848      002100 000005 ; 173100 000005 .WORD RFREAD
849      002102 004003 ; 173102 004003 .WORD RNUM
850      002104 100000 ; 173104 100000 .WORD 100000 ;DONE MASK
851      002106 024000 ; 173106 024000 .WORD 24000 ;ERROR MASK
852      ;
853      ;
854      ;
855      ; TM11 STARTING ADDRESS
    
```

K02

```

856 002110 010703 :173110 010703 TM11: MOV PC,R3 ;SAVE ERROR RETRY ADDRES
857 002112 012737 :173112 012737 MOV #TMRWD,0;#TMC-2;REWIND TAPE
858 002114 060017 :173114 060017
859 002116 172522 :173116 172522
860 002120 010702 :173120 010702 MOV PC,R2
861 002122 000555 :173122 000555 BR TAPES
862 002124 172524 :173124 172524 .WORD TMC
863 002126 060003 :173126 060003 .WORD TMRWD ;TM11 READ COMMAND
864 002130 060011 :173130 060011 .WORD TMRWD ;TM11 FORWARD RECORD COM
865 002132 000200 :173132 000200 .WORD 200 ;DONE MASK
866 002134 100000 :173134 100000 .WORD 100000 ;ERROR MASK
867
868 ;
869 ;THIS IS THE STARTING ADDRESS FOR RF11 CONTROLLE
870 002136 010703 :173136 010703 RF11: MOV PC,R3 ;SAVE ERROR RETRY ADDRES
871 002140 010702 :173140 010702 MOV PC,R2 ;SET POINTER TO PARAMETE
872 002142 000516 :173142 000516 BR OTHERA ;GO TO COMMON SERVICE RO
873 ;ASSUME UNIT 0
874 002144 177462 :173144 177462 .WORD RFWC ;DEVICE WORD COUNT REGIS
875 002146 000005 :173146 000005 .WORD RFREAD ;READ COMMAND
876 ;
877 ;THIS IS THE STARTING ADDRESS FOR RH/TU16/TM02
878 002150 010703 :173150 010703 TU16: MOV PC,R3 ;SAVE ERROR RETRY ADDRES
879 002152 012700 :173152 012700 MOV #TUCS,RO;GET CONTROL STATUS WORD
880 002154 172440 :173154 172440
881 002156 012710 :173156 012710 TU16RE: MOV #RHRST,(RO);REWIND TAPE CLEAR E
882 002160 000021 :173160 000021
883 002162 012760 :173162 012760 MOV #TUMODE,32(RO);SET 800 BPI NORMA
884 002164 001300 :173164 001300
885 002166 000032 :173166 000032
886 002170 012760 :173170 012760 MOV #-1,6(RO);LOAD FRAME COUNT
887 002172 177777 :173172 177777
888 002174 000006 :173174 000006
889 002176 012710 :173176 012710 MOV #TUSPAC,(RO);SPACE FORWARD
890 002200 000031 :173200 000031
891 002202 105760 :173202 105760 IS: TSTB 12(RO)
892 002204 000012 :173204 000012
893 002206 100375 :173206 100375 BPL IS ;KEEP LOOPING
894 002210 000433 :173210 000433 BR RHCOMN
895 ;
896 ;THIS IS THE STARTING ADDRESS FOR RC11 CONTROLLE
897 002212 010703 :173212 010703 RC11: MOV PC,R3
898 002214 010702 :173214 010702 MOV PC,R2 ;ASSUME UNIT 0
899 002216 000470 :173216 000470 BR OTHERA
900 002220 177450 :173220 177450 .WORD RCWC
901 002222 000005 :173222 000005 .WORD RFREAD
902 ;
903 ;THIS IS THE AUTO LOAD VECTOR
904 002224 173000 :173224 173000 .WORD RHSA
905 002226 000340 :173226 000340 .WORD 340
906 ;
907 ;THIS IS THE STARTING ADDRESS FOR RH11 DEVICE CO
908 ;
909 ;NOTE: IF TM02/TU16 SHOULD BE SELECTED. THE VAL
;IN CONSOL SWITCH REGISTER IS THE POSITIO
  
```



```

910      ;
911      ;
912 002230 000405 ;173230 000405 RH11A: BR 1$ ;ENTRY TO SELECT UNIT 0
913 002232 010703 ;173232 010703 RH11B: MOV PC,R3 ;ENTRY TO SELECT ALL UNI
914 002234 113737 ;173234 113737          MOVB 2#CSW,2#RHCSA+10;LOAD UNIT # INS
915 002236 177570 ;173236 177570
916 002240 176310 ;173240 176310
917 002242 000401 ;173242 000401          BR 2$
918 002244 010703 ;173244 010703 1$: MOV PC,R3
919 002246 012700 ;173246 012700 2$: MOV #RHCSA,RO
920 002250 176300 ;173250 176300
921 002252 032760 ;173252 032760 RPCOMN: BIT #TUTAPE,26(RO);TAPE UNIT?
922 002254 040000 ;173254 040000
923 002256 000026 ;173256 000026
924 002260 001336 ;173260 001336          BNE TUI6RE ;YES. GO TO TAPE LOGIC
925 002262 012710 ;173262 012710          MOV #RHRST,(RO);RESET DRIVE
926 002264 000021 ;173264 000021
927 002266 012760 ;173266 012760          MOV #14000,32(RO);SET 16 BIT FORMAT
928 002270 014000 ;173270 014000
929 002272 000032 ;173272 000032
930 002274 012710 ;173274 012710          MOV #DRCLR,(RO);CLEAR DRIVE ERROR
931 002276 000011 ;173276 000011
932      ;
933 002300 005720 ;173300 005720 RHCOMN: TST (RO)+ ;(GENERATED IF RS03/04
934 002302 010037 ;173302 010037          MOV RO,2#2 ;MOVE TO WORD COUNT ADDR
935 002304 000002 ;173304 000002          ;FAKE CALLING SEQUENCE
936 002306 012737 ;173306 012737          MOV #RHREAD,2#4
937 002310 000071 ;173310 000071
938 002312 000004 ;173312 000004
939 002314 005002 ;173314 005002          CLR R2 ;FOR FLAG AND POINTER TO
940 002316 000430 ;173316 000430          BR OTHERA
941      ;
942      ;
943      ;
944 002320 000405 ;173320 000405 RHRPA: BR 1$ ;THIS IS THE STARTING ADDRESS FOR RH11/RP04 DISK
945 002322 010703 ;173322 010703 RHRPB: MOV PC,R3 ;ENTRY FOR SELECT UNIT 0
946 002324 113737 ;173324 113737          MOVB 2#CSW,2#RPPSA+10;LOAD UNIT # INS
947 002326 177570 ;173326 177570
948 002330 176710 ;173330 176710
949 002332 000401 ;173332 000401          BR 2$
950 002334 010703 ;173334 010703 1$: MOV PC,R3
951 002336 012700 ;173336 012700 2$: MOV #RPPSA,RO
952 002340 176700 ;173340 176700
953 002342 000743 ;173342 000743          BR RPCOMN
954      ;
955      ;
956 002344 013707 ;173344 013707          ;ENTRY TO BRANCH TO THE PC SELECTED BY CONSOL SW
957 002346 177570 ;173346 177570          CSRGO: MOV 2#CSW,PC
958      ;
959      ;
960      ;
961      ;
962 002350 000405 ;173350 000405          ;THIS IS THE STARTING ADDRESS FOR RP11 CONTROLLE
963 002352 010703 ;173352 010703 RP11A: BR 1$ ;ENTRY TO SELECT UNIT 0
          RP11B: MOV PC,R3 ;ENTRY TO SELECT ALL UNI
    
```

964	002354	113705	:173354	113705		MOVB	2#CSW,R5	
965	002356	177570	:173356	177570				
966	002360	000305	:173360	000305		SWAB	R5	;GET UNIT # INTO HIGH BY
967	002362	000402	:173362	000402		BR	3\$	
968	002364	010703	:173364	010703	1\$:	MOV	PC,R3	
969	002366	005005	:173366	005005		CLR	R5	
970	002370	010702	:173370	010702	3\$:	MOV	PC,R2	
971	002372	000403	:173372	000403		BR	OTHER	
972	002374	176716	:173374	176716		.WORD	RPWC	
973	002376	000005	:173376	000005		.WORD	RFREAD	
974								
975	002400	005005	:173400	005005	OTHERA:	CLR	R5	;SET TO UNIT 0
976	002402	010200	:173402	010200	OTHER:	MOV	R2,R0	;R0 POINT AT WORD COUNT
977	002404	005720	:173404	005720		TST	(R0)+	;POINT TO PARAMETER LIST
978	002406	012001	:173406	012001		MOV	(R0)+,R1	;MOVE WORD COUNT ADDRESS
979	002410	012711	:173410	012711		MOV	#-256.*2,(R1)	;LOAD WORD COUNT
980	002412	177000	:173412	177000				
981	002414	051005	:173414	051005		BIS	(R0),R5	;COMBINE UNIT # WITH COM
982	002416	010541	:173416	010541		MOV	R5,-(R1)	;LOAD READ COMMAND
983	002420	032711	:173420	032711		BIT	#100200,(R1)	;CHECK FOR ERROR AND
984	002422	100200	:173422	100200				
985	002424	001775	:173424	001775		BEQ	.-4	;WAIT UNTIL COMPLETE
986	002426	100012	:173426	100012		BPL	1\$;NO ERROR
987	002430	005702	:173430	005702		TST	R2	;WAS IT CALLED BY MASS B
988	002432	001024	:173432	001024		BNE	AGAIN	;NO ERROR
989	002434	032761	:173434	032761		BIT	#TUTAPE,26(R1)	;IS TU16?
990	002436	040000	:173436	040000				
991	002440	000026	:173440	000026				
992	002442	001420	:173442	001420		BEQ	AGAIN	;NO. ERROR
993	002444	022761	:173444	022761		CMP	#FCE,14(R1)	;ARE WE READ A SHORT
994	002446	001000	:173446	001000				
995	002450	000014	:173450	000014				
996	002452	001014	:173452	001014		BNE	AGAIN	;SOME OTHER ERROR
997	002454	005007	:173454	005007	1\$:	CLR	PC	;O.K.
998								
999								
1000	002456	010200	:173456	010200				
1001	002460	005720	:173460	005720	TAPES:	MOV	R2,R0	;GET THE ADDRESS OF THE
1002	002462	012001	:173462	012001		TST	(R0)+	;STEP TO LAST COMMAND
1003	002464	005311	:173464	005311		MOV	(R0)+,R1	;GET THE WORD COUNT ADDR
1004	002466	005720	:173466	005720		DEC	(R1)	;SET UP TO ADVANCE 1 REC
1005	002470	012041	:173470	012041		TST	(R0)+	;MOVE R0 TO FIRST COMMAN
1006	002472	031011	:173472	031011		MOV	(R0)+,-(R1)	;LOAD COMMAND REG.
1007	002474	001776	:173474	001776		BIT	(R0),(R1)	;DONE?
1008	002476	005720	:173476	005720		BEQ	.-2	;NO. KEEP LOOPING
1009	002500	031041	:173500	031041		TST	(R0)+	;YES. CHECK FOR ERROR
1010	002502	001736	:173502	001736		BIT	(R0)-,(R1)	;ANY ERROR?
1011	002504	000005	:173504	000005	AGAIN:	BEQ	OTHERA	;NO ERROR- TRY TO READ
1012						RESET		
1013	002506	000113	:173506	000113		JMP	(R3)	;ERROR RETURN
1014								
1015								
1016	002510	012704	:173510	012704	KL11:	MOV	#KLCS,R4	;OBTAIN CONTROL REG.
1017	002512	177560	:173512	177560				

```

1018 002514 000443 ;173514 000443 BR CKDEV ;AND TRANSFER TO READER
1019
1020
1021
1022
1023 002516 .BYTE 240 ;173516 240 TABLE: .BYTE 240 ;COMPARE WORD NOT A COMM
1024 002517 .BYTE 037 ;173517 037 .BYTE 37 ;ILBS+RWD+GO
1025 002520 .BYTE 015 ;173520 015 .BYTE 15 ;SPACE FORWARD BLOCK+GO
1026 002521 .BYTE 005 ;173521 005 .BYTE 5 ;READ
1027 002522 .BYTE 024 ;173522 024 .BYTE 24 ;READ +ILBS
1028 002523 .BYTE 224 ;173523 224 .BYTE 224 ;READ+ILBS+END FLAG
1029
1030
1031 002524 000404 ;173524 000404 CBOOTA: BR 15 ;SELECT UNIT 0
1032 002526 113704 ;173526 113704 CBOOTB: MOVB #CSW,R4;SELECT UNITS
1033 002530 177570 ;173530 177570
1034 002532 000304 ;173532 000304 SWAB R4
1035 002534 000401 ;173534 000401 BR RESETX
1036 002536 005004 ;173536 005004 15: CLR R4
1037 002540 012700 ;173540 012700 RESETX: MOV #TACS,R0;GET CONTROL REG.
1038 002542 177500 ;173542 177500
1039 002544 000005 ;173544 000005 RESTRT: RESET
1040 002546 010410 ;173546 010410 MOV R4,(R0) ;SELECT UNIT
1041 002550 012701 ;173550 012701 MOV #TABLE,R1
1042 002552 173516 ;173552 173516
1043 002554 012702 ;173554 012702 MOV #375,R2 ;LOAD TRANSFER COUNTER
1044 002556 000375 ;173556 000375
1045 002560 112103 ;173560 112103 MOVB (R1)+,R3;LOAD COMPARATOR
1046 002562 112110 ;173562 112110 LOOP1: MOVB (R1)+,(R0);LOAD COMMAND
1047 002564 100407 ;173564 100407 BMI DONE
1048 002566 130310 ;173566 130310 LOOP2: BITB R3,(R0) ;COMMAND COMPLETE?
1049 002570 001776 ;173570 001776 BEQ LOOP2 ;NO. WAIT
1050 002572 105202 ;173572 105202 INCB R2 ;INCREMENT ADDRESS CTR.
1051 002574 100772 ;173574 100772 BMI LOOP1 ;IF (-) GET COMMAND
1052 002576 116012 ;173576 116012 MOVB 2(R0),(R2);STORE DATA
1053 002600 000002 ;173600 000002
1054 002602 000771 ;173602 000771 BR LOOP2 ;GET ANOTHER BYTE
1055 002604 005710 ;173604 005710 DONE: TST (R0) ;ANY ERROR?
1056 002606 100756 ;173606 100756 BMI RESTRT ;YES, RETRY
1057 002610 005002 ;173610 005002 CLR R2 ;CLEAR COMPARE ADDRESS
1058 002612 120312 ;173612 120312 CMPB R3,(R2) ;IT MUST BE 240
1059 002614 001377 ;173614 001377 BNE
1060 002616 000112 ;173616 000112 ERROR: JMP (R2)
1061
1062
1063 002620 012704 ;173620 012704 PC11: MOV #PCCS,R4
1064 002622 177550 ;173622 177550
1065 002624 000005 ;173624 000005 CKDEV: RESET
1066 002626 012701 ;173626 012701 MOV #160000,R1;SET UP MEMORY TEST LI
1067 002630 160000 ;173630 160000
1068 002632 012702 ;173632 012702 MOV #6,R2 ;SET UP POINTER TO TIME0
1069 002634 000006 ;173634 000006
1070 002636 012712 ;173636 012712 MOV #340,(R2);SET UP VECTOR TO RETUR
1071 002640 000340 ;173640 000340
  
```

1072	002642	010742	::173642	010742	MOV	PC, -(R2);	SAVE PC
1073	002644	012706	::173644	012706	MOV	R24, SP	;LOAD UP STACK POINTER
1074	002646	000024	::173646	000024			
1075	002650	010441	::173650	010441	MOV	R4, -(R1);	LOOK FOR END OF MEMORY
1076	002652	040601	::173652	040601	BIC	SP, R1	;THEN DROP TO XX752
1077	002654	010111	::173654	010111	MOV	R1, (R1)	;AND STORE IN ITSELF
1078	002656	011102	::173656	011102	LOOP: MOV	(R1), R2	
1079	002660	005214	::173660	005214	INC	(R4)	;START DEVICE
1080	002662	105714	::173662	105714	FORMAT: TSTB	(R4)	;WAIT
1081	002664	100376	::173664	100376	BPL	FORMAT	
1082	002666	116412	::173666	116412	MOVSB	2(R4), (R2);	SAVE THE DATA
1083	002670	000002	::173670	000002			
1084	002672	005211	::173672	005211	INC	(R1)	
1085	002674	120227	::173674	120227	CMPB	R2, R375	
1086	002676	000375	::173676	000375			
1087	002700	001366	::173700	001366	BNE	LOOP	:NO
1088	002702	105222	::173702	105222	INCB	(R2)+	:YES
1089	002704	000142	::173704	000142	JMP	-(R2)	
1090	002706	000000	::173706	000000			:THIS AREA IS UNUSED
1091	002710	000000	::173710	000000			:THIS AREA IS UNUSED
1092	002712	000000	::173712	000000			:THIS AREA IS UNUSED
1093	002714	000000	::173714	000000			:THIS AREA IS UNUSED
1094	002716	000000	::173716	000000			:THIS AREA IS UNUSED
1095	002720	000000	::173720	000000			:THIS AREA IS UNUSED
1096	002722	000000	::173722	000000			:THIS AREA IS UNUSED
1097	002724	000000	::173724	000000			:THIS AREA IS UNUSED
1098	002726	000000	::173726	000000			:THIS AREA IS UNUSED
1099	002730	000000	::173730	000000			:THIS AREA IS UNUSED
1100	002732	000000	::173732	000000			:THIS AREA IS UNUSED
1101	002734	000000	::173734	000000			:THIS AREA IS UNUSED
1102	002736	000000	::173736	000000			:THIS AREA IS UNUSED
1103	002740	000000	::173740	000000			:THIS AREA IS UNUSED
1104	002742	000000	::173742	000000			:THIS AREA IS UNUSED
1105	002744	000000	::173744	000000			:THIS AREA IS UNUSED
1106	002746	000000	::173746	000000			:THIS AREA IS UNUSED
1107	002750	000000	::173750	000000			:THIS AREA IS UNUSED
1108	002752	000000	::173752	000000			:THIS AREA IS UNUSED
1109	002754	000000	::173754	000000			:THIS AREA IS UNUSED
1110	002756	000000	::173756	000000			:THIS AREA IS UNUSED
1111	002760	000000	::173760	000000			:THIS AREA IS UNUSED
1112	002762	000000	::173762	000000			:THIS AREA IS UNUSED
1113	002764	000000	::173764	000000			:THIS AREA IS UNUSED
1114	002766	000000	::173766	000000			:THIS AREA IS UNUSED
1115	002770	000000	::173770	000000			:THIS AREA IS UNUSED
1116	002772	000000	::173772	000000			:THIS AREA IS UNUSED
1117	002774	000000	::173774	000000			:THIS AREA IS UNUSED
1118	002776	END. YB:					
1119	002776	000000	::173776	000000			:THIS AREA IS UNUSED

```

1120 003000 MAP.YC:
1121 :THE FOLLOWING 1000 LOCATIONS ARE
1122 :A REPRODUCTION OF THE ROM PROGRAM
1123 :FOR THE BMB73YC. THE FIRST 400 LOCATIONS
1124 :ARE AN EXACT COPY OF THE BMB73YA. THE
1125 :REMAINING 400 LOCATIONS ARE
1126 :THE DDCMP BOOTSTRAP ROM PROGRAM.
1127 :IT IS HERE FOR COMPARISON TO
1128 :ACTUAL ROM AND FOR REFERENCE.
1129 :173000 .=173000 ;STARTING ADDRESS FOR BOOTSTRAP
1130 :THIS LOADER IS DESIGNED FOR THE RESTART MODULE M873.
1131 :IT FUNCTIONALLY REPLACES THE FOLLOWING ROMS:
1132 :M792-YA - PAPER TAPE BOOTSTRAP FOR PC11,KL11
1133 :MR11-DB BULK STORAGE BOOTSTRAP ROM
1134 :M792-YH TAIL CASSETTE BOOTSTRAP ROM
1135 :
1136 :000000 R0= %0 ;REGISTER DEFINITIONS
1137 :000001 R1= %1
1138 :000002 R2= %2
1139 :000003 R3= %3
1140 :000004 R4= %4
1141 :000005 R5= %5
1142 :000006 SP= %6
1143 :000007 PC= %7
1144 :177570 SR= 177570 ;PROCESSOR SWITCH REGISTER
1145 :
1146 003000 010702 :173000 010702 ;STARTING LOCATION FOR RF11 DISK
1147 003002 000464 :173002 000464 RF11: MOV PC,R2 ;SET POINTER TO PARAMETER LISTS
1148 003004 177462 :173004 177462 BR OTHER ;TRANSFER TO SERVICE ROUTINE
1149 003006 000005 :173006 000005 .WORD 177462 ;DEVICE WORD COUNT ADDRESS
1150 :.WORD 5 ;DEVICE READ INSTRUCTION
1151 :
1152 003010 010702 :173010 010702 ;THIS IS THE STARTING LOCATION FOR THE RK11 CONTROLLER
1153 003012 000460 :173012 000460 RK11: MOV PC,R2 ;SET POINTER TO PARAMETER LIST
1154 003014 177406 :173014 177406 BR OTHER ;TRANSFER TO SERVICE ROUTINE
1155 003016 000005 :173016 000005 .WORD 177406 ;DEVICE WORD COUNT REGISTER
1156 :.WORD 5 ;DEVICE READ INSTRUCTION
1157 :
1158 :THIS IS A SPARE STARTING LOCATION. IT TRANSFERS TO ADDRESS
1159 003020 013707 :173020 013707 TRANSR: MOV @SR,PC ;GO TO INDICATED LOCATION
1160 003022 177570 :173022 177570
1161 :NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
1162 :
1163 :THIS IS THE POWER UP VECTOR REQUIRED FOR DEVICE AND
1164 003024 173000 :173024 173000 POWER: .WORD RF11 ;ADDRESS OF FIRST LOCATION IN ROM
1165 003026 000340 :173026 000340 .WORD 340 ;PROCESSOR STATUS LEVEL 7
1166 :
1167 :THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE) CONTROLLER.
1168 003030 010702 :173030 010702 TC11: MOV PC,R2 ;SET UP POINTER TO PARAMETER LIST
1169 003032 000426 :173032 000426 BR TAPES ;AND TRANSFER TO FIRST ROUTINE
1170 003034 177344 :173034 177344 .WORD 177344 ;DEVICE WORD COUNT ADDRESS
1171 003036 004003 :173036 004003 .WORD 4003 ;FIND PREVIOUS BLOCK COMMAND
1172 003040 100000 :173040 100000 .WORD 100000 ;USED AS DONE INDICATOR
1173 003042 024000 :173042 024000 .WORD 24000 ;USED AS ERROR INDICATOR/TEST FLAG

```

1174	003044	000445	:173044	000445	BR OTHERX	:THEN TRANSFER TO NEXT ROUTINE
1175	003046	000005	:173046	000005	.WORD 5	:DEVICE READ COMMAND
1176						
1177						
1178	003050	010702	:173050	010702	TM11: MOV PC,R2	:SET POINTER TO PARAMETER LIST
1179	003052	000416	:173052	000416	BR TAPES	:AND TRANSFER TO FIRST ROUTINE
1180	003054	172524	:173054	172524	.WORD 172524	:DEVICE BYTE/RECORD COUNT REGISTER
1181	003056	060017	:173056	060017	.WORD 60017	:DEVICE REWIND COMMAND
1182	003060	000200	:173060	000200	.WORD 200	:DEVICE DONE FLAG
1183	003062	100000	:173062	100000	.WORD 100000	:DEVICE ERROR FLAG BIT
1184	003064	000413	:173064	000413	BR TAPESX	:THEN TRANSFER TO NEXT SERVICE RTN
1185	003066	060011	:173066	060011	.WORD 60011	:DEVICE FORWARD SPACE COMMAND
1186	003070	000200	:173070	000200	.WORD 200	:SAME AS ABOVE
1187	003072	100000	:173072	100000	.WORD 100000	:SAME AS ABOVE
1188	003074	000431	:173074	000431	BR OTHERX	:THEN TRANSFER TO READ/TRANSFER ROUTINE
1189	003076	060003	:173076	060003	.WORD 60003	:DEVICE READ COMMAND
1190						
1191						
1192	003100	010702	:173100	010702	RP11: MOV PC,R2	:SET POINTER TO PARAMETER LIST
1193	003102	000424	:173102	000424	BR OTHER	:TRANSFER TO TRANSFER ROUTINE
1194	003104	176716	:173104	176716	.WORD 176716	:DEVICE WORD COUNT REGISTER
1195	003106	000005	:173106	000005	.WORD 5	:DEVICE READ COMMAND
1196						
1197						
1198	003110	010200	:173110	010200	TAPES: MOV R2,R0	:GET ADDRESS OF PARAMETER LIST
1199	003112	005720	:173112	005720	TST (R0)+	:SKIP TWO WORDS FIRST TIME
1200	003114	000005	:173114	000005	TAPESX: RESET	:RESET ALL DEVICES
1201	003116	005720	:173116	005720	TST (R0)+	:SKIP OVER BRANCH INSTRUCTION
1202	003120	016201	:173120	016201	MOV 2(R2),R1	:THEN GET DEVICE WORD/BYTE COUNT ADDRES
1203	003122	000002	:173122	000002		
1204	003124	005311	:173124	005311	DEC 2R1	:AND SET TO -1
1205	003126	012041	:173126	012041	MOV (R0)+,-(R1)	:AND THEN ISSUE COMMAND TO DEVICE
1206	003130	031011	:173130	031011	TAPMAT: BIT 2R0,2R1	:WAIT FOR DEVICE COMPLETION
1207	003132	001776	:173132	001776	BEG TAPMAT	:BY HANGING IN LOOP
1208	003134	005720	:173134	005720	TST (R0)+	:AND THEN SKIP DONE FLAG
1209	003136	032041	:173136	032041	BIT (R0)+,-(R1)	:THEN TEST FOR ERROR
1210	003140	001063	:173140	001063	BNE ERROR	:THERE IS ONE
1211	003142	000110	:173142	000110	RETURN: JMP 2R0	:AND TRANSFER TO FOLLOWING INSTRUCTION
1212						
1213						
1214	003144	010702	:173144	010702	RC11: MOV PC,R2	:SET UP POINTER TO PARAMETER LIST
1215	003146	000402	:173146	000402	BR OTHER	:TRANSFER TO SERVICE RTN
1216	003150	177450	:173150	177450	.WORD 177450	:DEVICE WORD COUNT REGISTER
1217	003152	000005	:173152	000005	.WORD 5	:DEVICE READ INSTRUCTION
1218						
1219						
1220	003154	010200	:173154	010200	OTHER: MOV R2,R0	:SET POINTER TO LIST IN R0
1221	003156	005720	:173156	005720	TST (R0)+	:SKIP TWO WORDS FIRST TIME.
1222	003160	005720	:173160	005720	OTHERX: TST (R0)+	:SKIP PAST BR INSTRUCTION
1223	003162	000005	:173162	000005	RESET	:REST THE WORLD
1224	003164	016201	:173164	016201	MOV 2(R2),R1	:OBTAIN DEVICE WORD COUNT ADDRESS
1225	003166	000002	:173166	000002		
1226	003170	012711	:173170	012711	MOV #-1000,2R1	:THEN OBTAIN LARGE WORD COUNT
1227	003172	177000	:173172	177000		

1228	003174	011041	::173174	011041	OTHWAT: MOV R0, -(R1)	: AND PUT COMMAND TO DEVICE
1229	003176	105711	::173176	105711	TST R1	: WAIT FOR DONE FLAG
1230	003200	100376	::173200	100376	BPL OTHWAT	: BY HANGING IN LOOP
1231	003202	005711	::173202	005711	TST R1	: THEN TEST FOR ERROR
1232	003204	100441	::173204	100441	BMI ERROR	: GOT PROBLEMS
1233	003206	005007	::173206	005007	CLR PC	: AND TRANSFER TO ZERO
1234						
1235						: THIS IS THE STARTING ADDRESS FOR THE PC11 PAPER TAPE CONTROLLER
1236	003210	012704	::173210	012704	KL11: MOV #177560, R4	: OBTAIN DEVICE ADDRESS
1237	003212	177560	::173212	177560		
1238	003214	000440	::173214	000440	BR CKDEV	: AND TRANSFER TO READER SERVICE ROUTINE
1239						
1240						
1241						: THIS IS THE CASSETTE DEVICE COMMAND TABLE
1242	003216	017640	::173216	240	TABLE: .BYTE 240	: COMPARE WORD NOT A COMMAND
1243			::173217	037	.BYTE 37	: ILBS+RMD+GO
1244	003220	002415	::173220	015	.BYTE 15	: SPACE FORWARD BLOCK+GO
1245			::173221	005	.BYTE 5	: READ+GO
1246	003222	112024	::173222	024	.BYTE 24	: READ+ILBS
1247			::173223	224	.BYTE 224	: READ+ILBS+END FLAG
1248						: NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
1249						
1250						: THIS IS AN ADDITIONAL POWER VECTOR ADDRESS REQUIRED BY DEVICE
1251	003224	173000	::173224	173000	POWER2: .WORD R11	: ADDRESS OF BEGINNING OF BOOTSTRAP
1252	003226	000340	::173226	000340	.WORD 340	: PRIORITY LEVEL 7
1253						
1254						: THIS IS THE STARTING ADDRESS FOR THE CASSETTE DEVICE #0
1255	003230	005004	::173230	005004	CBOOT: CLR R4	: LOAD DEVICE NUMBER 0 IN R4
1256	003232	012700	::173232	012700	RESTX: MOV #177500, R0	: GET DEVICE ADDRESS
1257	003234	177500	::	177500		
1258	003236	000005	::173236	000005	RESTRT: RESET	: ISSUE RESET INSTRUCTION
1259	003240	010410	::173240	010410	MOV R4, R0	: LOAD DEVICE WITH UNIT NUMBER
1260	003242	012701	::173242	012701	MOV #TABLE, R1	: GET FUNNY TABLE OF INSTRUCTIONS
1261	003244	173216	::173244	173216		
1262	003246	012702	::173246	012702	MOV #375, R2	: AND LOAD UP TRANSFER COUNTER
1263	003250	000375	::173250	000375		
1264	003252	112103	::173252	112103	MOV B (R1)+, R3	: THE LOAD UP COMPARATOR
1265	003254	112110	::173254	112110	MOV B (R1)+, R0	: LOAD DEVICE REGISTER WITH COMMAND
1266	003256	100407	::173256	100407	BMI DONE	
1267	003260	130310	::173260	130310	LOOP1: BIT B R3, R0	: HAS COMMAND COMPLETED
1268	003262	001776	::173262	001776	BEQ LOOP2	: NO, WAIT
1269	003264	105202	::173264	105202	INCB R2	: THEN INCREMENT ADDRESS CTR
1270	003266	100772	::173266	100772	BMI LOOP1	: IF NEGATIVE, GET COMMAND
1271	003270	116012	::173270	116012	MOV B 2(R0), R2	: AND STORE DATA AWAY
1272	003272	000002	::173272	000002		
1273	003274	000771	::173274	000771	BR LOOP2	: GO GET ANOTHER BYTE
1274	003276	005710	::173276	005710	DONE: TST R0	: ANY DEVICE ERRORS
1275	003300	100756	::173300	100756	BMI RESTRT	: YES, RETRY
1276	003302	005002	::173302	005002	CLR R2	: CLEAR COMPARE ADDRESS AND TRANSFER ADDRESS
1277	003304	120312	::173304	120312	CMP B R3, R2	: IT MUST BE 240
1278	003306	001377	::173306	001377	BNE .+0	: NO, THERE WAS AN ERROR
1279	003310	000112	::173310	000112	ERROR: JMP R2	: NORMAL CASSETTE AND ERROR FOR BULK STORAGE
1280						
1281						: THIS IS THE STARTING LOCATION FOR THE PC11 CONTROLLER

1282	003312	012704	:173312	012704	PC11:	MOV #177550,R4	;LOAD DEVICE ADDRESS
1283	003314	177550	:173314	177550			
1284	003316	000005	:173316	000005	CKDEV:	RESET	;KILL ALL DEVICE ACTION
1285	003320	012701	:173320	012701		MOV #160000,R1	;THEN SET UP MEMORY TEST LIMITS
1286	003322	160000	:173322	160000			
1287	003324	012702	:173324	012702		MOV #6,R2	;AND SET UP POINTER TO TIMEOUT LOCATION
1288	003326	000006	:173326	000006			
1289	003330	012712	:173330	012712		MOV #340,AR2	;AND SET UP VECTOR TO RETURN TO NEXT
1290	003332	000340	:173332	000340			
1291	003334	010742	:173334	010742		MOV PC,-(R2)	;SAVE THE PC
1292	003336	012706	:173336	012706		MOV #24,SP	;AND LOAD UP STACK POINTER
1293	003340	000024	:173340	000024			
1294	003342	010441	:173342	010441		MOV R4,-(R1)	;AND LOOK FOR END OF MEMORY
1295	003344	040601	:173344	040601		BIC SP,R1	;THEN DROP TO XX7752
1296	003346	010111	:173346	010111		MOV R1,AR1	;AND STORE IN ITSELF
1297	003350	011102	:173350	011102	LOOP:	MOV AR1,R2	;THEN LOAD ADDRESS FOR DATA INSERTION
1298	003352	005214	:173352	005214		INC AR4	;AND START DEVICE
1299	003354	105714	:173354	105714	RDRWAT:	TSTB AR4	;THEN WAIT FOR CHARACTER AVAILABLE
1300	003356	100376	:173356	100376		BPL RDRWAT	;HANGING THERE IF NECESSARY
1301	003360	116412	:173360	116412		MOVB 2(R4),AR2	;STORE AWAY DATA BYTE
1302	003362	000002	:173362	000002			
1303	003364	005211	:173364	005211		INC AR1	
1304	003366	120227	:173366	120227		CMPB R2,#375	;HAS BRANCH OFFSET BEEN STORED
1305	003370	000375	:173370	000375			
1306	003372	001366	:173372	001366		BNE LOOP	;NO
1307	003374	105222	:173374	105222		INCB (R2)+	;YES, ALL DONE
1308	003376	000142	:173376	000142		JMP -(R2)	;THEN TRANSFER TO RTN

1309
 1310 THE FOLLOWING 400 LOCATIONS ARE
 1311 A REPRODUCTION OF THE DDCMP BOOT-
 1312 STRAP ROM. IT IS HERE FOR COM-
 1313 PARISON TO THE ACTUAL ROM AND
 1314 FOR REFERENCE.
 1315

1316
 1317
 1318
 1319
 1320
 1321
 1322
 1323
 1324
 1325
 1326
 1327
 1328
 1329
 1330
 1331
 1332
 1333
 1334
 1335

.....

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

1336
 1337
 1338
 1339
 1340
 1341
 1342
 1343
 1344
 1345
 1346
 1347
 1348
 1349
 1350
 1351
 1352
 1353
 1354
 1355
 1356
 1357
 1358
 1359
 1360
 1361
 1362
 1363
 1364
 1365
 1366
 1367
 1368
 1369
 1370
 1371
 1372
 1373
 1374
 1375
 1376
 1377
 1378
 1379
 1380
 1381
 1382
 1383
 1384
 1385
 1386
 1387
 1388
 1389

 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=%0

; BLOCK LOAD ADDR

1390	:	000001	R1=%1	: DEVICE CSR ADDRESS
1391	:	000002	R2=%2	: CRC CALC TEMP
1392	:	000003	R3=%3	: SOFTWARE CRC
1393	:	000004	R4=%4	: BLOCK CHAR COUNT
1394	:	000005	R5=%5	: CRC CALC TEMP
1395	:	000006	SP=%6	: STACK ADDR
1396	:	000007	PC=%7	: LOCATION COUNTER
1397	:			
1398	:			
1399	:			
1400	:	000001	\$STADR=1	: STATION ADDR
1401	:	177570	\$SWR=177570	: SWITCH REGISTER ADDR
1402	:	000226	\$SYN=226	: SYNC CHARACTER
1403	:	000220	\$DLE=220	: DDCMP DLE CHARACTER
1404	:	000400	\$STRIP=400	
1405	:			
1406	:			
1407	:			
1408	:			
1409	:			
1410	:			
1411	:			
1412	:			
1413	:			
1414	:			
1415	:			
1416	:	173400	=173400	
1417	003400	012700	:173400 012700	START1: MOV (PC)+,R0 ;NON ZERO VALUE TO R0
1418	003402	005000	:173402 005000	START2: CLR R0 ;CLEAR R0
1419	003404	000005	:173404 000005	RESET ;RESET SYS, MEM MGT, ETC...
1420	003406	012706	:173406 012706	MOV #17776,SP ;STACK AT 4K-2
1421	003410	017776	:173410 017776	
1422	:			
1423	:			
1424	:			
1425	003412	010702	:173412 010702	MOV PC,R2 ;CURRENT PC
1426	003414	062702	:173414 062702	ADD #DEV TAB-,R2 ;DEVICE TABLE ADDR
1427	003416	000360	:173416 000360	
1428	003420	012703	:173420 012703	MOV #6,R3 ;TRAP PS ADDR
1429	003422	000006	:173422 000006	
1430	003424	005013	:173424 005013	CLR (R3) ;CLEAR NEW PS
1431	003426	010243	:173426 010243	MOV R2,-(R3) ;TABLE ADDR TO LOC 4
1432	003430	160313	:173430 160313	SUB R3,(R3) ;SUB TO TRAP RTN
1433	003432	005303	:173432 005303	DEC R3 ;LEAVE CNT 3 FOR LOOP
1434	003434	012701	:173434 012701	MOV #160010,R1 ;START SEARCH ADDR
1435	003436	160010	:173436 160010	
1436	003440	005711	:173440 005711	DEVELOP: TST (R1) ;IS DEVICE THERE
1437	003442	111204	:173442 111204	MOV# (R2),R4 ;DEVICE INCREMENT TO R3
1438	003444	060401	:173444 060401	ADD R4,R1 ;UPDATE TO NEXT DEVICE
1439	003446	005201	:173446 005201	INC R1 ;INCREMENT MODULO
1440	003450	040401	:173450 040401	BIC R4,R1 ;CLEAR EXCESS
1441	003452	005703	:173452 005703	TST R3 ;TEST FOR DONE
1442	003454	001371	:173454 001371	BNE DEVELOP ;NOT YET
1443	003456	005700	:173456 005700	TST R0 ;TEST SWITCH REG USE

LITERALS

THE STACK IS USED AS FOLLOWS:
STACK-2:FOR JSR TO GET ROUTINE
STACK-4:TEMP FOR CRC CALCULATION

START OF BOOT PROGRAM

START1-DEVICE UNIT 0-NORMAL CONFIGURATION
START2-USE SWITCH REG AS DEVICE DISPLACEMENT
I.E. #0-0, #1-10, #2-20

FIND THE DU-11 IN THE FLOATING ADDRESS SPACE

1444	003460	001002	::173460	001002	BNE	SNOREQ	;NO SWITCH REG
1445	003462	063701	::173462	063701	ADD	#SSWR,R1	;ADD SWR VALUE
1446	003464	177570	::173464	177570			
1447							
1448							
1449							
1450			::173466				
1451	003466	012711	::173466	012711	SNOREQ:	MOV	#6,(R1) ;DATA TERM RDY AND REQ TO SEND
1452	003470	000006	::173470	000006			
1453	003472	012761	::173472	012761	MOV	#36000+\$SYN,2(R1) ;SET SYNC REGISTER	
1454	003474	036226	::173474	036226			
1455	003476	000002	::173476	000002			
1456	003500	032711	::173500	032711	L3:	BIT	#20000,(R1) ;TEST CLEAR TO SEND
1457	003502	020000	::173502	020000			
1458	003504	001775	::173504	001775	BEQ	L3	;NOT YET
1459	003506	022121	::173506	022121	CMP	(R1)+,(R1)+	;MOVE PTR TO XMIT TSR
1460	003510	052711	::173510	052711	BIS	#20,(R1)	;TURN SEND ON
1461	003512	000020	::173512	000020			
1462							
1463							
1464							
1465	003514	010700	::173514	010700	MOV	PC,R0	;CURRENT PC
1466	003516	062700	::173516	062700	ADD	#RQMSG-. ,R0	;REQUEST MSG ADDR
1467	003520	000230	::173520	000230			
1468	003522	012704	::173522	012704	MOV	#RQMSGE-RQMSG,R4 ;COUNT	
1469	003524	000026	::173524	000026			
1470	003526	112061	::173526	112061	L4:	MOVB	(R0)+,2(R1) ;CHAR TO XMIT REGISTER
1471	003530	000002	::173530	000002			
1472	003532	105711	::173532	105711	L5:	TSTB	(R1) ;DONE YET ?
1473	003534	100376	::173534	100376	BPL	L5	;NO
1474	003536	005304	::173536	005304	DEC	R4	;DECREMENT COUNT
1475	003540	001372	::173540	001372	BNE	L4	;ONCE MORE
1476	003542	042711	::173542	042711	BIC	#20,(R1)	;DROP SEND
1477	003544	000020	::173544	000020			
1478	003546	024141	::173546	024141	CMP	-(R1),-(R1)	;RESET PTR TO RCV CSR
1479							
1480							
1481							
1482			::173550				
1483	003550	042711	::173550	042711	GETPGM:	BIC	#20,(R1) ;CLEAR SEARCH SYNC
1484	003552	000020	::173552	000020			
1485	003554	012711	::173554	012711	MOV	#422,(R1)	;SET FOR CLEAR AND STRIP SYNC
1486	003556	000422	::173556	000422			
1487	003560	005003	::173560	005003	CLR	R3	;CLEAR CRC VALUE
1488							
1489							
1490							
1491	003562	012700	::173562	012700	GET MESSAGE HEADER AND CHECK IT	MOV	#1,R0 ;LOAD HDR AT LOC. 1
1492	003564	000001	::173564	000001			
1493	003566	012704	::173566	012704	MOV	#8.,R4	;BLOCK COUNT
1494	003570	000010	::173570	000010			
1495	003572	004767	::173572	004767	JSR	PC,GET	;GET HEADER
1496	003574	000060	::173574	000060			
1497	003576	005703	::173576	005703	TST	R3	;CHECK HEADER CRC

J03

1498	003600	001363	:173600	001363	BNE	GETPGM	;NO GOOD
1499	003602	123727	:173602	123727	CMPB	2#6,#\$STADR	;CHECK FOR MY ADDR
1500	003604	000006	:173604	000006			
1501	003606	000001	:173606	000001			
1502	003610	001357	:173610	001357	BNE	GETPGM	;NOT MINE
1503	003612	123727	:173612	123727	CMPB	2#1,#\$DLE	;IS THIS A DLE MSG
1504	003614	000001	:173614	000001			
1505	003616	000220	:173616	000220			
1506	003620	001322	:173620	001322	BNE	SNDREQ	;NO, ASK FOR ONE
1507							
1508							
1509							
1510	003622	013704	:173622	013704	MOV	2#2,R4	;DATA FIELD LENGTH
1511	003624	000002	:173624	000002			
1512	003626	042704	:173626	042704	BIC	#140000,R4	;MASK OFF S,F BITS
1513	003630	140000	:173630	140000			
1514	003632	122424	:173632	122424	CMPB	(R4)+,(R4)+	;ADD 2 FOR CRC
1515	003634	005000	:173634	005000	CLR	R0	;LOAD INTO LOCATION 0
1516	003636	004767	:173636	004767	JSR	PC,GET1	;GET DATA BLOCK
1517	003640	000014	:173640	000014			
1518	003642	005703	:173642	005703	TST	R3	;CHECK DATA FIELD CRC
1519	003644	001310	:173644	001310	BNE	SNDREQ	;NO GOOD
1520	003646	105713	:173646	105713	TSTB	(R3)	;CHECK CODE IN LOC 0
1521	003650	001306	:173650	001306	BNE	SNDREQ	;NOT PROGRAM LOAD
1522	003652	000137	:173652	000137	JMP	2#6	;TRANSFER TO SECONDARY PGM
1523	003654	000006	:173654	000006			
1524							
1525							
1526							
1527			:173656				
1528			:173656				
1529	003656	105711	:173656	105711	TSTB	(R1)	;IS DEVICE DONE YET
1530	003660	100376	:173660	100376	BPL	GET	;NOT YET
1531	003662	042711	:173662	042711	BIC	#\$STRIP,(R1)	;NO STRIP SYNC
1532	003664	000400	:173664	000400			
1533	003666	116110	:173666	116110	MOVB	2(R1),(R0)	;STORE IT
1534	003670	000002	:173670	000002			
1535							
1536							
1537							
1538			:120001				
1539							
1540	003672	012705	:173672	012705	MOV	#8.,R5	;BYTE LENGTH
1541	003674	000010	:173674	000010			
1542	003676	112002	:173676	112002	MOVB	(R0)+,R2	;CHARACTER TO ADD TO CRC
1543	003700	000241	:173700	000241	CRCLOP:	CLC	;CLEAR CARRY
1544	003702	006003	:173702	006003	ROR	R3	;SHIFT OLD PARTIAL
1545	003704	103003	:173704	103003	BCC	L10	;IF CLEAR CHECK CHAR
1546	003706	006002	:173706	006002	ROR	R2	;SHIFT CHARACTER
1547	003710	103003	:173710	103003	BCC	L11	;XOR POLY
1548	003712	000410	:173712	000410	BR	L12	;NEXT BIT
1549	003714	006002	:173714	006002	L10:	ROR	;SHIFT CHARACTER
1550	003716	103006	:173716	103006	BCC	L12	;NEXT BIT
1551	003720	012746	:173720	012746	L11:	MOV	#\$POLY,-(SP)

K03

1552	003722	120001	;	173722	120001			
1553	003724	040316	;	173724	040316	BIC	R3,(SP)	;NOT PARTIAL AND POLY
1554	003726	042703	;	173726	042703	BIC	#POLY,R3	;NOT POLY AND PARTIAL
1555	003730	120001	;	173730	120001			
1556	003732	052603	;	173732	052603	BIS	(SP)+,R3	;POLY XOR PARTIAL
1557	003734	005305	;	173734	005305	DEC	R5	;DECREMENT BIT COUNT
1558	003736	001360	;	173736	001360	BNE	CRCLOP	;ONCE MORE
1559	003740	005304	;	173740	005304	DEC	R4	;DECREMENT COUNT
1560	003742	001345	;	173742	001345	BNE	GET	;ONCE MORE
1561	003744	000207	;	173744	000207	RTS	PC	;RETURN
1562			;					
1563			;					
1564			;					
1565	003746	113226	;	173746	113226	RQMSG:	.BYTE	\$SYN,\$SYN,\$SYN,\$SYN
1566	003750	113226	;	173750	113226			
1567	003752	002220	;	173752	002220		.BYTE	\$DLE,4,0,0,0,1
1568	003754	000000	;	173754	000000			
1569	003756	000400	;	173756	000400			
1570	003760	050055	;	173760	050055		.BYTE	55,120
1571	003762	001010	;	173762	001010		.BYTE	10
1572			;				.BYTE	2
1573	003764	000001	;	173764	000001		.BYTE	\$STADR
1574			;				.BYTE	0
1575	003766	030242	;	173766	030242		.BYTE	242,60
1576			;					
1577			;					
1578			;					
1579			;					
1580	003770	122243	;	173770	122243	NODEV:	.EVEN	
1581	003772	000002	;	173772	000002		CMPB	(R2)+,-(R3)
1582			;				RTI	;INC PTR-DEC CNT
1583			;					;RETURN FROM TRAP
1584	003774	007407	;	173774	007407	RQMSG:		
1585			;			DEVTAB:	.BYTE	7
1586	003776	END.YC:	;				.BYTE	17
1587	003776	003407	;	173776	003407		.BYTE	7
1588			;				.BYTE	7
1589			;	174000		END:		
1590			;		173400		.END	START1

1591 004000 MAP.YD:
 1592 :THE FOLLOWING IS A REPRODUCTION
 1593 :OF THE ROM PROGRAM FOR BM873YD.
 1594 :IT IS HERE FOR COMPARISON TO THE
 1595 :ACTUAL ROM AND FOR REFERENCE
 1596 :BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 1
 1597 :BM873-YD.P11

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

1603	:	000000	RO=%0	:GENERAL PURPOSE REGISTER 0
1604	:	000001	R1=%1	:GENERAL PURPOSE REGISTER 1
1605	:	000002	R2=%2	:GENERAL PURPOSE REGISTER 2
1606	:	000003	R3=%3	:GENERAL PURPOSE REGISTER 3
1607	:	000004	R4=%4	:GENERAL PURPOSE REGISTER 4
1608	:	000005	R5=%5	:GENERAL PURPOSE REGISTER 5
1609	:	000006	SP=%6	:STACK POINTER (REGISTER R6)
1610	:	000007	PC=%7	:PROGRAM COUNTER (REGISTER R7)

;SYMBOL DEFINITIONS

1615	:	177776	PS=177776	:PROCESSOR STATUS REGISTER
1616	:	177570	SMR=177570	:FRONT PANEL SWITCH REGISTER
1617	:	000000	PRO=0#40	:PRIORITY LEVEL 0
1618	:	000040	PR1=1#40	:PRIORITY LEVEL 1
1619	:	000100	PR2=2#40	:PRIORITY LEVEL 2
1620	:	000140	PR3=3#40	:PRIORITY LEVEL 3
1621	:	000200	PR4=4#40	:PRIORITY LEVEL 4
1622	:	000240	PR5=5#40	:PRIORITY LEVEL 5
1623	:	000300	PR6=6#40	:PRIORITY LEVEL 6
1624	:	000340	PR7=7#40	:PRIORITY LEVEL 7
1625	:	000001	BIT0=000001	
1626	:	000002	BIT1=000002	
1627	:	000004	BIT2=000004	
1628	:	000010	BIT3=000010	
1629	:	000020	BIT4=000020	
1630	:	000040	BIT5=000040	
1631	:	000100	BIT6=000100	
1632	:	000200	BIT7=000200	
1633	:	000400	BIT8=000400	
1634	:	001000	BIT9=001000	
1635	:	002000	BIT10=002000	
1636	:	004000	BIT11=004000	
1637	:	010000	BIT12=010000	
1638	:	020000	BIT13=020000	
1639	:	040000	BIT14=040000	
1640	:	100000	BIT15=100000	
1641	:			

M03

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 37
 DZBMD.P11 ROM CONTENTS TABLES

```

1642      ;BM873-YD      - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17)  MACY11 27(657) 18-DEC-74 11:59 PAGE 3
1643      ;BM873-YD.P11  BUTTON #1 - BOOTSTRAP USING THE PDP-11 SWITCH REGISTER
1644
1645
1646      ;           173000  ROMORG  =           173000      ;SET ROM ORIGIN TO 773000
1647      ;           173000      .ROMORG      ;BM873-YD OCCUPIES 773000-773777
1648
1649  004000  033727 ;173000  033727  BUTON1: BIT      @#SWR,#BIT0      ;IS RIGHTMOST BIT ON?
1650  004002  177570 ;173002  177570
1651  004004  000001 ;173004  000001
1652  004006  001010 ;173006  001010      BNE      LOWBIT      ;IF THE BIT IS ON, BRANCH
1653  004010  013707 ;173010  013707      MOV      @#SWR,PC      ;JUMP TO THE ADDRESS IN THE SWITCH REGISTER
1654  004012  177570 ;173012  177570
1655
1656
1657  004014  111704 ;173014  111704  BUTON3: MOVB      (PC),R4      ;R4 = 1 INDICATES THAT BUTTON #3 WAS PRESSED
1658  004016  005001 ;173016  005001      CLR      R1      ;SET UNIT NUMBER TO ZERO
1659  004020  005005 ;173020  005005      CLR      R5      ;CLEAR "LOGICAL SWITCH REGISTER"
1660  004022  000424 ;173022  000424      BR       TCBOOT      ;DO A DEFAULT BOOT STRAP FROM DECTAPE
1661
1662  004024  173000 ;173024  173000      .WORD   ROMORG,PR7
1663  004026  000340 ;173026  000340
1664
1665  004030  013701 ;173030  013701  LOWBIT: MOV      @#SWR,R1      ;R1 IS A COPY OF THE SWITCH REGISTER
1666  004032  177570 ;173032  177570
1667  004034  106301 ;173034  106301      ASLB    R1      ;LEFT-ALIGN SPEED FIELD IN RIGHT BYTE
1668  004036  122701 ;173036  122701      CMPB   #16*20,R1      ;IS THE SPEED 16 OR 17?
1669  004040  000340 ;173040  000340
1670  004042  101404 ;173042  101404      BLOS   UNITNO      ;IF SPEED IS 16 OR 17, BRANCH
1671  004044  122701 ;173044  122701      CMPB   #3*20,R1      ;IS THE SPEED 0, 1, OR 2?
1672  004046  000060 ;173046  000060
1673  004050  101001 ;173050  101001      BHI    UNITNO      ;IF THE SPEED IS 0, 1, OR 2, BRANCH
1674  004052  005001 ;173052  005001      CLR    R1      ;SPEED WAS 3-15; SET UNIT NUMBER = 0
1675  004054  000301 ;173054  000301  UNITNO: SWAB    R1      ;MOVE UNIT NUMBER TO BITS 0-2
1676
1677
1678      ; IT IS POSSIBLE TO MANUALLY SET THE DESIRED BOOTSTRAP UNIT NUMBER
1679      ; INTO THE RIGHTMOST THREE BITS OF R1, SET THE PDP-11 FRONT PANEL
1680      ; SWITCH REGISTER, AND THEN JUMP INTO THE ROM CODE AT THIS POINT.
1681  004056  042701 ;173056  042701      BIC    #1C7,R1      ;ISOLATE UNIT NUMBER IN R1
1682  004060  177770 ;173060  177770
1683  004062  013705 ;173062  013705      MOV    @#SWR,R5      ;R5 IS NOW THE "LOGICAL SWITCH REGISTER"
1684  004064  177570 ;173064  177570
1685  004066  005004 ;173066  005004      CLR    R4      ;R4 = 0 INDICATES THAT BUTTON #1 WAS PRESSED
1686  004070  105705 ;173070  105705      TSTB   R5      ;SHOULD WE BOOT FROM DECTAPE OR RH11/RP04?
1687  004072  100507 ;173072  100507      BMI    RPBOOT      ;IF BIT 7 WAS ONE, BRANCH OFF TO THE RH11/RP04
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
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
  
```

Address	Label	Value	Symbol	Description
1689	:BM673-YD	- KL10 (PDP-11) 256 WORD BOOTSTRAP ROM	VERSION 2(17)	MACY11 27(657) 18-DEC-74 11:59 PAGE 4
1690	:BM873-YD.P11	DECTAPE BOOTSTRAP AND DUMP ROUTINES		
1691				
1692				
1693	:	177344	TCWC =	177344 ;TC11 DECTAPE WORD COUNT REGISTER
1694	:	000001	TCGO =	1 ;TC11 "GO" BIT
1695	:	000002	TCRNUM =	1*2 ;TC11 "READ BLOCK NUMBER" FUNCTION
1696	:	000004	TCCREAD =	2*2 ;TC11 "READ DATA" FUNCTION
1697	:	000014	TCWRIT =	6*2 ;TC11 "WRITE DATA" FUNCTION
1698	:	004000	TCREV =	4000 ;MOVE DECTAPE IN REVERSE DIRECTION
1699				
1700				: BOOTSTRAP (FROM DECTAPE) PARAMETERS
1701	:	000400	TCBWDC =	↓D256 ;WORD COUNT FOR THE SECONDARY BOOTSTRAP
1702	:	000000	TCBEND =	0 ;WHICH END OF THE DECTAPE (0 = FRONT; 1 = BACK)
1703				
1704				: DUMP (TO DECTAPE) PARAMETERS
1705	:	070000	TCDWDC =	↓D28672 ;WORD COUNT FOR THE CORE DUMP TO DECTAPE
1706	:	000001	TCDEND =	1 ;WHICH END OF THE DECTAPE (0 = FRONT; 1 = BACK)
1707				
1708				: GENERAL (BOOTSTRAP AND DUMP) DECTAPE PARAMETER
1709	:	000024	TCRTRY =	↓D20 ;NUMBER OF RETRIES IN CASE OF ERROR
1710				
1711	004074 012700	:173074 012700	TCBOOT: MOV	#:TCBEND*TCREV>!TCCREAD!TCGO,R0 ;SET UP DATA-TRANSFER COMMAND
1712	004076 000005	:173076 000005		
1713	004100 012702	:173100 012702	MOV	#:TCBWDC,R2 ;SET WORD COUNT TO 256 (512 BYTES)
1714	004102 177400	:173102 177400		
1715	004104 012703	:173104 012703	MOV	#:((1-TCBEND)*TCREV)!TCRNUM!TCGO,R3 ;SET UP POSITION COMMAND
1716	004106 004003	:173106 004003		
1717	004110 000301	:173110 000301	SWAB	R1 ;BRING UNIT NUMBER INTO THE LEFT BYTE
1718	004112 050103	:173112 050103	BIS	R1,R3 ;PUT UNIT NUMBER INTO POSITIONING COMMAND
1719	004114 050100	:173114 050100	BIS	R1,R0 ;PUT UNIT NUMBER INTO DATA-TRANSFER COMMAND
1720	004116 012701	:173116 012701	TCSTRT: MOV	!TCWC,R1 ;R1 NOW POINTS TO TC11 WORD COUNT REGISTER
1721	004120 177344	:173120 177344		
1722	004122 012706	:173122 012706	TCLOOP: MOV	!TCRTRY,SP ;INITIALIZE RETRY COUNT IN SP
1723	004124 000024	:173124 000024		
1724	004126 005705	:173126 005705	TCBGIN: TST	R5 ;TEST "INDEFINITE RETRY" BIT
1725	004130 100404	:173130 100404	BMI	TCRSET ;BRANCH IF "INDEFINITE RETRY" IS ENABLED
1726	004132 005306	:173132 005306	DEC	SP ;DECREMENT RETRY COUNT
1727	004134 100002	:173134 100002	BPL	TCRSET ;BRANCH IF RETRY COUNT NOT EXHAUSTED
1728	004136 000000	:173136 000000	TCHALT: HALT	RETRY COUNT IS EXHAUSTED FOR DECTAPE OPERATION
1729	004140 000770	:173140 000770	BR	TCLOOP ;HE PRESSED "CONTINUE" SC TRY AGAIN
1730	004142 000005	:173142 000005	TCRSET: RESET	TCLOOP ;STOP ANYTHING IN PROGRESS, FOR NEXT TRY
1731	004144 010341	:173144 010341	TCWAIT: MOV	R3,-(R1) ;INITIATE DECTAPE POSITIONING OPERATION
1732	004146 005711	:173146 005711	TST	(R1) ;TEST FOR AN "ERROR"
1733	004150 100376	:173150 100376	BPL	TCWAIT ;LOOP UNTIL AN "ERROR" IS DETECTED
1734	004152 005721	:173152 005721	TST	(R1)+ ;MAKE R1 POINT TO THE WORD COUNT REGISTER
1735	004154 005761	:173154 005761	TST	-4(R1) ;IS THE ERROR "ENDZONE"?
1736	004156 177774	:173156 177774		
1737	004160 100362	:173160 100362	BPL	TCBGIN ;IF NOT, BRANCH BACK TO TRY AGAIN
1738	004162 010211	:173162 010211	MOV	R2,(R1) ;SET UP WORD COUNT FOR DATA-TRANSFER
1739	004164 010041	:173164 010041	MOV	R0,-(R1) ;INITIATE THE DATA-TRANSFER OPERATION
1740	004166 105711	:173166 105711	TCDONE: TSTB	(R1) ;TEST FOR "DONE"
1741	004170 100376	:173170 100376	BPL	TCDONE ;LOOP UNTIL THE "DONE" BIT SETS
1742	004172 005721	:173172 005721	TST	(R1)+ ;WAS AN "ERROR" DETECTED?

1743	004174	100754	::	173174	100754	BMI	TCBGIN	: IF SO, BRANCH BACK AND TRY AGAIN
1744	004176	005741	::	173176	005741	TST	-(R1)	: MAKE R1 POINT TO THE COMMAND REGISTER
1745	004200	105011	::	173200	105011	CLRB	(R1)	: STOP ALL DECTAPE MOTION
1746	004202	122700	::	173202	122700	CHPB	@TCREAD:TCGO,R0	: WAS THIS A "NORMAL READ" OPERATION?
1747	004204	000005	::	173204	000005			
1748	004206	001001	::	173206	001001	BNE	TCSTOP	: IF NOT, GO STOP
1749	004210	000137	::	173210	000137	GOTO0: JMP	2(PC)+	: JUMP TO PDP-11 LOCATION ZERO
1750	004212	000000	::	173212	000000	TCSTOP: HALT		: SUCCESSFUL COMPLETION OF A "NON-READ" OPERATION
1751	004214	000776	::	173214	000776	BR	TCSTOP	: SO THAT PRESSING "CONTINUE" WON'T GO ANYWHERE

MAINDEC-11-DZBND-J MACY11 27(663) 2-MAY-77 11:46 PAGE 40
 DZBND.P11 ROM CONTENTS TABLES

```

1752           ;BM873-YD           - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM    VERSION 2(17)    MACY11 27(657)    18-DEC-74    11:59    PAGE 5
1753           ;BM873-YD.P11       DECTAPE BOOTSTRAP AND DUMP ROUTINES
1754
1755   004216   010037   ;173216   010037   TCDUMP: MOV     RO,#R0TOR7           ;SAVE RO IN PDP-11 MEMORY LOCATION 40
1756   004220   000040   ;173220   000040                                            ;
1757   004222   000402   ;173222   000402                                            ;BR     TCCONT                               ;BRANCH AROUND REQUIRED INTERRUPT VECTOR
1758
1759   004224   173000   ;173224   173000                                            ;.WORD   ROMORG,PR7
1760   004226   000340   ;173226   000340
1761
1762   004230   010700   ;173230   010700   TCCONT: MOV     PC,RO                       ;USE RO FOR A SUBROUTINE RETURN ADDRESS
1763   004232   000410   ;173232   000410                                            ;BR     REGSAV                              ;GO TO THE "REGISTER SAVING" SUBROUTINE
1764   004234   012700   ;173234   012700                                            ;MOV    #(TCDEND*TCREV)!TCWRIT!TCGO,RO   ;SET UP (WRITE) TRANSFER COMMAND
1765   004236   004015   ;173236   004015                                            ;
1766   004240   012702   ;173240   012702                                            ;MOV    #-TCDWDC,R2                       ;SET WORD-COUNT TO 28K WORDS
1767   004242   110000   ;173242   110000                                            ;
1768   004244   012703   ;173244   012703                                            ;MOV    #(<<1-TCDEND)*TCREV)!TCRNUM!TCGO,R3   ;SET UP POSITION COMMAND
1769   004246   000003   ;173246   000003                                            ;
1770   004250   005005   ;173250   005005                                            ;CLR    R5                                 ;CLEAR "INDEFINITE RETRY" BIT
1771   004252   000721   ;173252   000721                                            ;BR     TCSTRT                             ;BRANCH INTO DECTAPE ROUTINE
1772
1773
1774
1775
1776                                            ; THE FOLLOWING SUBROUTINE IS USED TO SAVE THE PDP-11 GENERAL REGISTERS
1777                                            ; IN PDP-11 MEMORY LOCATIONS 40-57.
1778
1779                                            ; THE CALLING SEQUENCE IS AS FOLLOWS:                                            MOV     RO,#R0TOR7
1780                                            ;                                                                                    MOV     PC,RO
1781                                            ;                                                                                    BR     REGSAV
1782                                            ;                                                                                    RETURN HERE
1783
1784   004254   010137   ;173254   010137   REGSAV: MOV     R1,#R0TOR7+2           ;SAVE R1 IN MEMORY LOCATION 42
1785   004256   000042   ;173256   000042                                            ;
1786   004260   012701   ;173260   012701                                            ;MOV    #R0TOR7+4,R1                      ;R1 NOW POINTS TO MEMORY LOCATION 44
1787   004262   000044   ;173262   000044                                            ;
1788   004264   010221   ;173264   010221                                            ;MOV    R2,(R1)+                          ;SAVE R2 IN MEMORY LOCATION 44
1789   004266   010321   ;173266   010321                                            ;MOV    R3,(R1)+                          ;SAVE R3 IN MEMORY LOCATION 46
1790   004270   010421   ;173270   010421                                            ;MOV    R4,(R1)+                          ;SAVE R4 IN MEMORY LOCATION 50
1791   004272   010521   ;173272   010521                                            ;MOV    R5,(R1)+                          ;SAVE R5 IN MEMORY LOCATION 52
1792   004274   010621   ;173274   010621                                            ;MOV    SP,(R1)+                         ;SAVE SP IN MEMORY LOCATION 54
1793   004276   010021   ;173276   010021                                            ;MOV    RO,(R1)+                         ;SAVE PC IN MEMORY LOCATION 56
1794   004300   000160   ;173300   000160                                            ;JMP    2(RO)                             ;RETURN TO THE CALLING ROUTINE
1795   004302   000002   ;173302   000002
1796

```

1797 :BMB73-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 6
 1798 :BMB73-YD.P11 RH11/RP04 BOOTSTRAP AND DUMP ROUTINES

1801	:	176700	RPCS1	=	176700	:	ADDRESS OF RH11/RP04 CONTROL & STATUS REGISTER 1
1802	:	000002	RPWC	=	2	:	OFFSET TO RH11/RP04 WORD COUNT REGISTER
1803	:	000006	RPDA	=	6	:	OFFSET TO RH11/RP04 TRACK & SECTOR ADDRESS REGISTER
1804	:	000010	RPCS2	=	10	:	OFFSET TO RH11/RP04 CONTROL & STATUS REGISTER 2
1805	:	000012	RPDS	=	12	:	OFFSET TO RH11/RP04 DRIVE STATUS REGISTER
1806	:	000032	RPOF	=	32	:	OFFSET TO RH11/RP04 OFFSET REGISTER (CONTAINING FMT22)
1807	:	000034	RPDC	=	34	:	OFFSET TO RH11/RP04 DESIRED CYLINDER REGISTER
1808	:						
1809	:	040000	RPTRE	=	BIT14	:	"TRANSFER ERROR" BIT IN RPCS1
1810	:	020000	RPWCPE	=	BIT13	:	"MASSBUS CONTROL BUS PARITY ERROR" BIT IN RPCS1
1811	:	004000	RPDVA	=	BIT11	:	"DRIVE AVAILABLE" BIT IN RPCS1
1812	:	100000	RPATA	=	BIT15	:	"ATTENTION ACTIVE" BIT IN RPDS
1813	:	040000	RPERR	=	BIT14	:	"COMPOSITE ERROR" BIT IN RPDS
1814	:	010000	RPFMT	=	BIT12	:	"FMT22" (16-BIT WORDS) BIT IN RPOF
1815	:						
1816	:	000021	RPPRST	=	21	:	READ-IN PRESET
1817	:	000061	RPWRIT	=	61	:	WRITE DATA
1818	:	000071	RPREAD	=	71	:	READ DATA
1819	:						
1820	:	000000	RPBFMT	=	0	:	BOOTSTRAP FORMAT (0 = 18-BIT WORDS; 2 = 16-BIT WORDS)
1821	:	000400	RPBWC	=	↑D256	:	WORD COUNT FOR THE SECONDARY BOOTSTRAP FROM THE RP04
1822	:	000626	RPBCYL	=	↑D406	:	BOOTSTRAP CYLINDER NUMBER
1823	:	000000	RPBTRK	=	0	:	BOOTSTRAP TRACK NUMBER
1824	:	000000	RPBSCT	=	0	:	BOOTSTRAP SECTOR NUMBER
1825	:						
1826	:	000000	RPDFMT	=	0	:	DUMP FORMAT (0 = 18-BIT WORDS; 2 = 16-BIT WORDS)
1827	:	070000	RPDWC	=	↑D28672	:	WORD COUNT FOR THE CORE DUMP TO THE RP04
1828	:	000631	RPDCYL	=	↑D409	:	DUMP CYLINDER NUMBER
1829	:					:	THE FOLLOWING TWO ASSIGNMENTS PUT THE DUMP AT THE VERY END OF THE CYLINDER
1830	:	000015	RPDTRK	=	↑D18-((RPDWC-1)/((↑D20+RPDFMT)*↑D256))	:	DUMP TRACK NUMBER
1831	:	000010	RPDSCT	=	↑D19+RPDFMT-(((RPDWC-1)/↑D256)-((↑D18-RPDTRK)*((↑D20+RPDFMT)))	:	
1832	:						
1833	:						
1834	:						
1835	004304	111704	;173304	111704	BUTON2: MOV	(PC),R4	;R4 = 5 INDICATES THAT BUTTON #2 WAS PRESSED
1836	004306	005005	;173306	005005	CLR	R5	;CLEAR "LOGICAL SWITCH REGISTER"
1837	004310	005001	;173310	005001	CLR	R1	;SET UNIT NUMBER TO ZERO
1838	:						
1839	004312	012700	;173312	012700	RPBOOT: MOV	#(RPREAD*400)!(RPBSCT*10),R0	
1840	004314	034400	;173314	034400			
1841	004316	012702	;173316	012702	MOV	#-RPBWC,R2	
1842	004320	177400	;173320	177400			
1843	004322	012703	;173322	012703	MOV	#(RPBFMT*40000)!(RPBTRK*2000)!RPBCYL,R3	
1844	004324	000626	;173324	000626			
1845	004326	050100	;173326	050100	RPSTRT: BIS	R1,R0	;PUT THE UNIT NUMBER INTO R0
1846	004330	012701	;173330	012701	MOV	#RPCS1,R1	;SET R1 TO THE LOWEST ADDRESS USED BY THE RH11
1847	004332	176700	;173332	176700			

E04

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 42
 DZBMD.P11 ROM CONTENTS TABLES

VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 7

1848	:BM873-YD			- KL10 (PDP-11) 256 WORD BOOTSTRAP ROM		
1849	:BM873-YD.P11			RH11/RP04 BOOTSTRAP AND DUMP ROUTINES		
1850						
1851	004334	000005	:173334	000005	RLOOP: RESET	:RESET IN CASE OF RETRY
1852	004336	010006	:173336	010006	MOV R0,SP	:GET THE UNIT NUMBER INTO SP
1853	004340	042706	:173340	042706	BIC #1C7,SP	:ISOLATE THE UNIT NUMBER
1854	004342	177770	:173342	177770		
1855	004344	010661	:173344	010661	MOV SP,RPCS2(R1)	:TELL THE RH11 THE UNIT NUMBER
1856	004346	000010	:173346	000010		
1857	004350	032711	:173350	032711	BIT #RPDVA,(R1)	:TRY TO SEIZE THIS RP04 UNIT
1858	004352	004000	:173352	004000		
1859	004354	001767	:173354	001767	BEQ RLOOP	:BRANCH IF WE HAVEN'T SEIZED IT
1860	004356	012721	:173356	012721	MOV #RPPRST,(R1)+	:DO A "READ-IN PRESET" FUNCTION
1861	004360	000021	:173360	000021		
1862	004362	010306	:173362	010306	MOV R3,SP	:GET THE CYLINDER NUMBER INTO SP
1863	004364	042706	:173364	042706	BIC #1C1777,SP	:ISOLATE THE CYLINDER NUMBER
1864	004366	176000	:173366	176000		
1865	004370	010661	:173370	010661	MOV SP,RPDC-2(R1)	:TELL THE RP04 THE CYLINDER NUMBER
1866	004372	000032	:173372	000032		
1867	004374	010306	:173374	010306	MOV R3,SP	:GET THE FORMAT BIT AND TRACK NUMBER INTO SP
1868	004376	100003	:173376	100003	BPL RPCONT	:BRANCH IF 20 SECTOR (18-BIT WORDS) FORMAT
1869	004400	012761	:173400	012761	MOV #RPFMT,RPOF-2(R1)	:ESTABLISH 22 SECTOR (16-BIT WORDS) FORMAT
1870	004402	010000	:173402	010000		
1871	004404	000030	:173404	000030		
1872	004406	006206	:173406	006206	RPCONT: ASR SP	:RIGHT ALIGN THE TRACK
1873	004410	006206	:173410	006206	ASR SP	NUMBER IN THE LEFT BYTE
1874	004412	105006	:173412	105006	CLRB SP	:CLEAR THE RIGHT BYTE
1875	004414	150006	:173414	150006	BISB R0,SP	:PUT THE SECTOR NUMBER INTO THE RIGHT BYTE
1876	004416	106006	:173416	106006	RORB SP	:RIGHT ALIGN THE
1877	004420	106206	:173420	106206	ASRB SP	SECTOR NUMBER IN
1878	004422	106206	:173422	106206	ASRB SP	THE RIGHT BYTE
1879	004424	010661	:173424	010661	MOV SP,RPDA-2(R1)	:TELL THE RH11 THE TRACK AND SECTOR NUMBERS
1880	004426	000004	:173426	000004		
1881	004430	010211	:173430	010211	MOV R2,(R1)	:TELL THE RH11 THE WORD COUNT
1882	004432	010006	:173432	010006	MOV R0,SP	:GET THE FUNCTION CODE INTO SP
1883	004434	105006	:173434	105006	CLRB SP	:CLEAR THE RIGHT BYTE
1884	004436	000306	:173436	000306	SWAB SP	:RIGHT ALIGN THE FUNCTION CODE
1885	004440	010641	:173440	010641	MOV SP, -(R1)	:TELL THE RP04 THE FUNCTION CODE
1886	004442	105711	:173442	105711	RPDONE: TSTB (R1)	:TEST FOR RH11 "READY"
1887	004444	100376	:173444	100376	BPL RPDONE	:LOOP WAITING FOR RH11 "READY"
1888	004446	032711	:173446	032711	BIT #RPTRE!RPMCPE,(R1)	:TEST FOR RH11 ERROR BITS
1889	004450	060000	:173450	060000		
1890	004452	001330	:173452	001330	BNE RLOOP	:IF ERROR, BRANCH BACK FOR RETRY
1891	004454	032761	:173454	032761	BIT #RPATA!RPERR,RPDS(R1)	:TEST FOR RP04 ERROR BITS
1892	004456	140000	:173456	140000		
1893	004460	000012	:173460	000012		
1894	004462	001324	:173462	001324	BNE RLOOP	:IF ERROR, BRANCH BACK FOR RETRY
1895	004464	022706	:173464	022706	CMP #RPREAD,SP	:WAS THE FUNCTION A "NORMAL READ"?
1896	004466	000071	:173466	000071		
1897	004470	001250	:173470	001250	BNE TCSTOP	:IF NOT, BRANCH TO A HALT INSTRUCTION
1898	004472	022737	:173472	022737	CMP #000240,#0	:WAS "000240" READ INTO LOCATION ZERO?
1899	004474	000240	:173474	000240		
1900	004476	000000	:173476	000000		
1901	004500	001643	:173500	001643	BEQ GOT00	:IF SO, BRANCH TO LOCATION ZERO

F04

1902	004502	000000 ;173502	000000	HALT		;"000240" WAS NOT READ INTO LOCATION ZERO
1903	004504	000641 ;173504	000641	BR	GOTO0	;BRANCH TO LOCATION ZERO
1904						
1905						
1906	004506	010037 ;173506	010037	RPDUMP: MOV	RD,2#ROTOR7	;SAVE RD IN PDP-11 MEMORY LOCATION "ROTOR7"
1907	004510	000040 ;173510	000040			
1908	004512	010700 ;173512	010700	MOV	PC,RO	;USE RD FOR A SUBROUTINE RETURN ADDRESS
1909	004514	000657 ;173514	000657	BR	REGSAV	;GO TO THE "REGISTER SAVING" SUBROUTINE
1910	004516	012700 ;173516	012700	MOV	#<RPWRIT*400>!<RPDSCT*10>,RO	
1911	004520	030500 ;173520	030500			
1912	004522	012702 ;173522	012702	MOV	#-RPDWDG,R2	
1913	004524	110000 ;173524	110000			
1914	004526	012703 ;173526	012703	MOV	#<RPDFMT*40000>!<RPDTRK*2000>!RPDCYL,R3	
1915	004530	032631 ;173530	032631			
1916	004532	000676 ;173532	000676	BR	RPSTRT	
1917						

```

1918 ;BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 8
1919 ;BM873-YD.P11 DTE20 DEVICE REGISTER AND BIT DEFINITIONS
1920
1921
1922 ; 174400 DTEBAS=174400 ;BASE OF (FIRST) DTE20 DEVICE REGISTER BLOCK
1923 ; 000040 DTESIZ=000040 ;SPACING BETWEEN CONSECUTIVE DTE20'S
1924 ; 000004 DTEMAX=4 ;MAXIMUM NUMBER OF DTE20'S ON ONE PDP-11
1925
1926 ;
1927 ; OFFSETS FROM THE BASE OF THE DTE20 DEVICE REGISTER BLOCK
1928 ; TO SPECIFIC 10/11 INTERFACE RAM LOCATIONS AND REGISTERS.
1929
1930 ; THE FIRST 12 REGISTERS ARE NOT INITIALIZED BY "INIT" (BECAUSE THEY ARE IN RAMS
1931 ; 000000 DLYCNT=00 ;DELAY COUNT (ADDRESS XXXX00)
1932 ; 000002 DEXWD3=02 ;DEPOSIT OR EXAMINE WORD 3 (ADDRESS XXXX02)
1933 ; 000004 DEXWD2=04 ;DEPOSIT OR EXAMINE WORD 2 (ADDRESS XXXX04)
1934 ; 000006 DEXWD1=06 ;DEPOSIT OR EXAMINE WORD 1 (ADDRESS XXXX06)
1935 ; 000010 TENAD1=10 ;10 ADDRESS WORD 1 FOR DEX (ADDRESS XXXX10)
1936 ; 000012 TENAD2=12 ;10 ADDRESS WORD 2 FOR DEX (ADDRESS XXXX12)
1937 ; 000014 T010BC=14 ;T010 BYTE COUNT (ADDRESS XXXX14)
1938 ; 000016 T011BC=16 ;T011 BYTE COUNT (ADDRESS XXXX16)
1939 ; 000020 T010AD=20 ;T010 PDP11 MEMORY ADDRESS (ADDRESS XXXX20)
1940 ; 000022 T011AD=22 ;T011 PDP11 MEMORY ADDRESS (ADDRESS XXXX22)
1941 ; 000024 T010DT=24 ;T010 PDP11 DATA WORD (ADDRESS XXXX24)
1942 ; 000026 T011DT=26 ;T011 PDP11 DATA WORD (ADDRESS XXXX26)
1943
1944 ; THE LAST 4 REGISTERS ARE INITIALIZED BY "INIT" (BECAUSE THEY ARE IN FLIP-FLOPS
1945 ; 000030 DIAG1=30 ;DIAGNOSTIC WORD 1 (ADDRESS XXXX30)
1946 ; 000032 DIAG2=32 ;DIAGNOSTIC WORD 2 (ADDRESS XXXX32)
1947 ; 000034 STATUS=34 ;10/11 INTERFACE STATUS WORD (ADDRESS XXXX34)
1948 ; 000036 DIAG3=36 ;DIAGNOSTIC WORD 3 (ADDRESS XXXX36)
1949
1950
1951 ; THE FOLLOWING ARE THE ADDRESSES OF THE DTE20 INTERRUPT VECTORS
1952
1953 ; 000774 DTEIV1=774 ;INTERRUPT VECTOR FOR DTE20 #1
1954 ; 000770 DTEIV2=770 ;INTERRUPT VECTOR FOR DTE20 #2
1955 ; 000764 DTEIV3=764 ;INTERRUPT VECTOR FOR DTE20 #3
1956 ; 000760 DTEIV4=760 ;INTERRUPT VECTOR FOR DTE20 #4
1957
1958
1959 ; BIT ASSIGNMENTS FOR VARIOUS DTE20 REGISTERS USED BY THIS ROM CODE
1960
1961 ;
1962 ;BIT ASSIGNMENTS FOR T010BC
1963
1964 ; 100000 INT11=BIT15 ;SET DONE AND INTERRUPT BOTH 10 AND 11
1965
1966 ;BIT ASSIGNMENTS FOR T011BC
1967
1968 ; 100000 INT10=BIT15 ;SET DONE AND INTERRUPT BOTH 10 AND 11
1969 ; 040000 ZSTOP=BIT14 ;STOP ON NULL (ZERO) CHARACTER
1970 ; 020000 T011BM=BIT13 ;BYTE SIZE FOR T0-11 BYTE TRANSFERS
1971

```

H04

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 45
 DZBMD.P11 ROM CONTENTS TABLES

```

1972 ;BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 9
1973 ;BM873-YD.P11 DTE20 DEVICE REGISTER AND BIT DEFINITIONS
1974
1975 ;BIT ASSIGNMENTS FOR DIAG2 (WRITE)
1976 ; 000100 DRESET=BIT6 ;PERFORM DIAGNOSTIC CLEAR
1977
1978 ;BIT ASSIGNMENTS FOR DIAG3 (READ)
1979 ; 000020 DUPE=BIT4 ;DATO UNIBUS PARITY ERROR
1980 ; 000004 DURE=BIT2 ;DATO UNIBUS RECEIVE ERROR
1981 ; 000002 NUPE=BIT1 ;NPR UNIBUS PARITY ERROR
1982
1983 ;BIT ASSIGNMENTS FOR DIAG3 (WRITE)
1984
1985 ; 000020 CDD=BIT4 ;CLEAR DUPE AND DURE ERROR FLAGS
1986 ; 000002 CNUPE=BIT1 ;CLEAR NUPE ERROR FLAG
1987 ; 000001 T010BM=BIT0 ;BYTE SIZE FOR T0-10 BYTE TRANSFER
1988
1989 ;BIT ASSIGNMENTS FOR STATUS (WRITE)
1990
1991 ; 100000 DON10S=BIT15 ;SET T010 DONE
1992 ; 040000 DON10C=BIT14 ;CLEAR T010 DONE
1993 ; 020000 ERR10S=BIT13 ;SET T010 ERROR
1994 ; 010000 ERR10C=BIT12 ;CLEAR T010 ERROR
1995 ; 004000 INT11S=BIT11 ;RING THE PDP-11'S DOORBELL (INTERRUPTS THE -11)
1996 ; 002000 INT11C=BIT10 ;STOP RINGING THE PDP-11'S DOORBELL
1997 ; 001000 PERCLR=BIT9 ;CLEAR -11 MEMORY PARITY ERROR
1998 ; 000400 INT10S=BIT8 ;RING THE PDP-10'S DOORBELL (INTERRUPTS THE -10)
1999 ; 000200 DON11S=BIT7 ;SET T011 DONE
2000 ; 000100 DON11C=BIT6 ;CLEAR T011 DONE
2001 ; 000040 INTRON=BIT5 ;ENABLE DTE20 INTERRUPTS TO THE -11
2002 ; 000020 EBUSPC=BIT4 ;CLEAR "EBUS PARITY ERROR"
2003 ; 000010 INTR0F=BIT3 ;DISABLE THE PDP-11 INTERRUPTS
2004 ; 000004 EBUSPS=BIT2 ;SET "EBUS PARITY ERROR"
2005 ; 000002 ERR11S=BIT1 ;SET T011 ERROR
2006 ; 000001 ERR11C=BIT0 ;CLEAR T011 ERROR
2007
2008 ;BIT ASSIGNMENTS FOR STATUS (READ)
2009
2010 ; 100000 T010DN=BIT15 ;T010 DONE
2011 ; 020000 T010ER=BIT13 ;TO 10 ERROR (NPR TIMEOUT OR BUS ERROR)
2012 ; 010000 RAMISO=BIT12 ;RAM WORD READ IS ALL ZEROS
2013 ; 004000 T011DB=BIT11 ;1 = THE PDP11'S DOORBELL IS RINGING
2014 ; 002000 DXMRD1=BIT10 ;DEPOSIT OR EXAMINE WORD ONE
2015 ; 001000 MPE11=BIT9 ;PARITY ERROR WITHIN PDP-11 MEMORY
2016 ; 000400 T010DB=BIT8 ;1 = THE PDP-10'S DOORBELL IS RINGING
2017 ; 000200 T011DN=BIT7 ;T011 DONE
2018 ; 000100 EBSEL=BIT6 ;E BUFFER SELECT
2019 ; 000040 NULSTP=BIT5 ;NULL STOP
2020 ; 000020 BPARER=BIT4 ;EBUS PARITY ERROR
2021 ; 000010 RSTRACT=BIT3 ;THIS PDP-11 IS "RESTRICTED"
2022 ; 000004 DEXDON=BIT2 ;DEPOSIT OR EXAMINE DONE
2023 ; 000002 T011ER=BIT1 ;TO 11 ERROR (NPR TIMEOUT OR BUS ERROR)
2024 ; 000001 INTSON=BIT0 ;DTE20 INTERRUPTS (TO THE -11) ARE ENABLED
  
```

2025 ;BMB73-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 1
 2026 ;BMB73-YD.P11 PROCEDURE BY WHICH THE PDP-10 BOOTSTRAPS AND/OR DUMPS THE PDP-11

2027
 2028
 2029
 2030
 2031
 2032
 2033 THE FOLLOWING IS THE PROCEDURE WHICH THE KL10 EXECUTES IN ORDER
 2034 TO DUMP AND/OR BOOTSTRAP THE PDP-11 THROUGH THE DTE20:

- 2035
 2036
 2037 1. CLEAR THE DTE20 AND INITIATE A BMB73 BUTTON #4 BOOTSTRAP OPERATION
 2038 - CONO [SR11B!CL11PT!CLT011!CLT010!PILDEN]
 2039
 2040 2. WAIT TO SEE PDP-11 POWER FAIL (AC LOW = TRUE) - CONI [DEAD11] = 1
 2041
 2042 3. WAIT TO SEE PDP-11 POWER RECOVER (AC LOW = FALSE) - CONI [DEAD11] = 0
 2043
 2044 4. WAIT AT LEAST ANOTHER 150 MILLISECONDS AND THEN CLEAR THE RELOAD -11 BUTTON
 2045 - CONO [CR11B]
 2046
 2047 5. SET BYTE COUNTER TO A SPECIAL CODE (1365 OCTAL) - DATA0 [1365]
 2048
 2049 6. RING PDP-11'S DOORBELL - CONO[TO11DB]
 2050
 2051 7. WAIT UNTIL "-10 RINGING -11'S DOORBELL" IS TURNED OFF BY THE -11
 2052 (I.E. UNTIL CONI[TO11DB] BECOMES ZERO).
 2053
 2054 8. ENABLE THE DTE20 TO USE PI 0 INTERRUPTS
 2055 (I.E. SET CONO[PILDEN!PI0ENB]).
 2056
 2057 9. SET UP THE TO-10 BYTE POINTER (IN THE EPT) FOR THE FIRST 3.5K.
 2058
 2059 10. SET UP THE BYTE COUNTER FOR THE FIRST 3.5K, INDICATING
 2060 "INTERRUPT -10 ONLY" - DATA0 [1000]
 2061
 2062 11. WAIT FOR "TO-10 DONE" OR "TO-10 ERROR" - CONI [TO10DN!TO10ER]
 2063
 2064 12. NOTE WHETHER THERE WAS AN ERROR (CONI [TO10ER]) AND THEN TURN OFF
 2065 TO10DN AND TO10ER - CONO [CLT010]. IF ERROR, GO TO STEP 17.
 2066
 2067 13. IF END OF 28K, GO TO STEP 17.
 2068
 2069 14. SET UP TO-10 BYTE POINTER (IN THE EPT) FOR THE NEXT 3.5K.
 2070
 2071 15. SET UP THE BYTE COUNTER FOR THE NEXT 3.5K INDICATING
 2072 "INTERRUPT -10 ONLY" (DATA0 [1000]), UNLESS THIS IS THE
 2073 LAST 3.5K (OF 28K), IN WHICH CASE INDICATE "INTERRUPT
 2074 BOTH PROCESSORS" (DATA0 [TO10IB!1000]).
 2075
 2076 16. GO TO STEP 11.
 2077
 2078

J04

2079 ;BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 1
2080 ;BM873-YD.P11 PROCEDURE BY WHICH THE PDP-10 BOOTSTRAPS AND/OR DUMPS THE PDP-11
2081
2082
2083 17. SET UP TO-11 BYTE POINTER (IN THE EPT) FOR "PDP-11 BOOTSTRAP".
2084 NOTE THAT THE FIRST WORD OF THIS "PDP-11 BOOTSTRAP" MUST
2085 BE THE BIT PATTERN 000240 (A PDP-11 NOP INSTRUCTION).
2086
2087 18. RING THE PDP-11'S DOORBELL - CONO [T011DB]
2088
2089 19. WAIT FOR EITHER T011DB TO GO OFF (CONI[T011DB] = 0),
2090 OR T010DB TO COME ON (CONI[T010DB] = 1).
2091
2092 20. IF NO ERROR WAS NOTED IN STEP 12, T011DB SHOULD GO OFF
2093 (T010DB COMING ON INDICATES A MASSIVE SCREWUP).
2094 IF AN ERROR WAS NOTED IN STEP 12, T011DB GOING OFF INDICATES
2095 THAT THE ERROR WAS "NON-FATAL" (NON-EX-MEM OR -11 MEMORY
2096 PARITY) AND THE -11 IS PROCEEDING. T010DB COMING ON INDICATES
2097 THAT THE ERROR WAS "FATAL" AND THE -11 IS HALTED AT LOCATION 173714.
2098 IN THIS LATTER CASE THE -10 MUST RESTART FROM STEP 1.
2099
2100 21. IF T011DB WENT OFF, WAIT FOR "TO-11 DONE" OR "TO-11 ERROR"
2101 - CONI [T011DN:T011ER]
2102
2103 22. NOTE WHETHER THERE WAS AN ERROR - CONI [T011ER]
2104
2105 23. TURN OFF T011DN AND T011ER AND RING THE PDP-11'S DOORBELL
2106 - CONO [T011DB:CLT011]
2107
2108 24. WAIT FOR EITHER T011DB TO GO OFF (CONI[T011DB] = 0),
2109 OR T010DB TO COME ON (CONI[T010DB] = 1).
2110
2111 25. T011DB GOING OFF INDICATES THAT THE PDP-11 FOUND NO ERRORS
2112 AND IS TRANSFERRING CONTROL TO THE CODE WHICH WAS JUST
2113 RECEIVED FROM THE -10. IN THIS CASE THE -10 SHOULD START
2114 FOLLOWING THE PROTOCOL OF THIS CODE.
2115
2116 26. T010DB COMING ON INDICATES THAT THE PDP-11 HAS FOUND AN
2117 ERROR (OR THAT THE FIRST WORD TRANSMITTED WASN'T THE
2118 BIT PATTERN 000240), AND THE PDP-11 IS HALTED AT LOCATION 173766.
2119 IN THIS CASE THE -10 MUST RESTART FROM STEP 1.
2120
2121
2122
2123
2124
2125

K04

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 48
 DZBMD.P11 ROM CONTENTS TABLES

```

2126 ;BMB73-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 1
2127 ;BMB73-YD.P11 BUTTON #4 - BOOTSTRAP INITIATED BY THE PDP-10 (THROUGH DTE20)
2128
2129
2130 ; 000130 DTECOR = 130 ;CORE ADDRESS INTO WHICH TO STORE DTE20 REGS.
2131 ; 000014 DTEREG = 1D12 ;NUMBER OF DTE20 REGISTERS TO STORE
2132 ; 000400 DTEWDC = 1D256 ;WORD COUNT FOR SECONDARY BOOTSTRAP FROM THE -10
2133 ; ; ENTER HERE WHEN THE DTE20 PRESSES BUTTON #4 (BOOTSTRAP INITIATED
2134 ; BY THE PDP-10, THROUGH THE DTE20)
2135 004534 010037 ;173534 010037 BUTON4: MOV RO,#ROTOR7 ;SAVE RO IN PDP-11 MEMORY LOCATION "ROTOR7"
2136 004536 000040 ;173536 000040
2137 004540 010700 ;173540 010700 MOV PC,RO ;USE RO FOR A SUBROUTINE RETURN ADDRESS
2138 004542 000644 ;173542 000644 BR REGSAV ;GO TO THE "REGISTER SAVING" SUBROUTINE
2139 004544 005005 ;173544 005005 CLR R5 ;SET R5 = 0
2140 004546 012501 ;173546 012501 MOV (R5)+,R1 ;SAVE LOCATION 0 IN R1
2141 004550 012503 ;173550 012503 MOV (R5)+,R3 ;SAVE LOCATION 2 IN R3
2142 004552 012504 ;173552 012504 MOV (R5)+,R4 ;SAVE LOCATION 4 IN R4
2143 004554 011500 ;173554 011500 MOV (R5),RO ;SAVE LOCATION 6 IN RO
2144 004556 012715 ;173556 012715 MOV #PR7,(R5) ;SET UP PRIORITY FOR NON-EX-MEM TRAP
2145 004560 000340 ;173560 000340
2146 004562 005745 ;173562 005745
2147 004564 012702 ;173564 012702 10$: TST -(R5) ;SET R5 = 4
2148 004566 174340 ;173566 174340 MOV #DTEBAS-DTESIZ,R2
2149 004570 010715 ;173570 010715 MOV PC,(R5) ;STORE ADDRESS FOR NON-EX-MEM TRAP
2150 004572 010506 ;173572 010506 MOV R5,SP ;SET STACK POINTER = 4
2151 004574 062702 ;173574 062702 11$: ADD #DTESIZ,R2 ;R2 POINTS TO THE NEXT DTE20
2152 004576 000040 ;173576 000040
2153 004600 105702 ;173600 105702 TSTB R2
2154 004602 100770 ;173602 100770 BMI 10$ ;START LOOKING FROM THE BEGINNING AGAIN
2155 004604 032762 ;173604 032762 BIT #T011DB,STATUS(R2) ;IS THIS -10 RINGING THE -11'S DOORBELL?
2156 004606 004000 ;173606 004000
2157 004610 000034 ;173610 000034
2158 004612 001770 ;173612 001770 BEQ 11$ ;IF IT IS NOT, GO LOOK FOR ANOTHER -10
2159 004614 026217 ;173614 026217 CMP T010BC(R2),(PC) ;CHECK FOR A CODE (1365) FROM THE PDP-10
2160 004616 000014 ;173616 000014
2161 ; ;INDICATING THAT IT WANTS TO BOOTSTRAP THE -11
2162 004620 001365 ;173620 001365 BNE 11$
2163 ; NOTE THAT AT THIS POINT R2 CONTAINS THE ADDRESS OF THE DEVICE REGISTER
2164 ; BLOCK FOR THIS DTE20, THAT R5 = 4, AND THAT SP = 4
2165 004622 005725 ;173622 005725 TST (R5)+ ;SET R5 = 6
2166 004624 010015 ;173624 010015 MOV RO,(R5) ;RESTORE THE CONTENTS OF LOCATION 6
2167 004626 010445 ;173626 010445 MOV R4,-(R5) ;RESTORE THE CONTENTS OF LOCATION 4
2168 004630 010345 ;173630 010345 MOV R3,-(R5) ;RESTORE THE CONTENTS OF LOCATION 2
2169 004632 010145 ;173632 010145 MOV R1,-(R5) ;RESTORE THE CONTENTS OF LOCATION 0
2170 ; NOTE: AT THIS TIME R5 = 0. THIS FACT WILL BE USED LATER.
2171 004634 012700 ;173634 012700 MOV #DTECOR,RO ;RO = CORE ADDRESS FOR STORING DTE20 REGISTERS
2172 004636 000130 ;173636 000130
2173 004640 010204 ;173640 010204
2174 004642 012420 ;173642 012420 7$: MOV (R4)+,(RO)+ ;SAVE THE NEXT DTE20 REGISTER IN CORE
2175 004644 022700 ;173644 022700 CMP #<DTEREG*2>+DTECOR,RO ;HAVE WE FINISHED YET?
2176 004646 000160 ;173646 000160
2177 004650 101374 ;173650 101374 BHI 7$ ;LOOP UNTIL WE HAVE FINISHED

```

```

2178 ;BM873-YD - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 2(17) MACY11 27(657) 18-DEC-74 11:59 PAGE 1
2179 ;BM873-YD.P11 BUTTON #4 - BOOTSTRAP INITIATED BY THE PDP-10 (THROUGH DTE20)
2180
2181 004652 010201 ;173652 010201 MOV R2,R1 ;R1 = DTE20 DEVICE REGISTER BLOCK
2182 004654 062701 ;173654 062701 ADD #DIAG2,R1
2183 004656 000032 ;173656 000032
2184 004660 012721 ;173660 012721 MOV #DRESET,(R1)+ ;DO A "DIAGNOSTIC CLEAR" OF THE DTE20,
2185 004662 000100 ;173662 000100
2186 ; THE ABOVE OPERATION IS NECESSARY TO CLEAR THE "BYTE COUNT LOADED" FLAG
2187 ; AND SIMULTANEOUSLY TO TURN OFF "-10 RINGING -11'S DOORBELL".
2188 004664 005012 ;173664 005012 CLR (R2) ;SET DTE20 FOR NO DELAY
2189 004666 005062 ;173666 005062 CLR T010AD(R2) ;START WRITING -11 MEMORY INTO THE -10.
2190 004670 000020 ;173670 000020
2191 004672 032711 ;173672 032711 6S: BIT #T011DB,(R1) ;HAS THE -10 RUNG THE -11'S DOORBELL?
2192 004674 004000 ;173674 004000
2193 004676 001775 ;173676 001775 BEQ 6S ;LOOP UNTIL IT HAS.
2194 004700 032762 ;173700 032762 BIT #DUPE!DURE!NUPE,DIAG3(R2) ;"FATAL" ERROR?
2195 004702 000026 ;173702 000026
2196 004704 000036 ;173704 000036
2197 004706 001403 ;173706 001403 BEQ 8S ;BRANCH IF NO "FATAL" ERROR
2198 004710 012711 ;173710 012711 MOV #T010DB,(R1) ;SIGNAL "FATAL" ERROR TO THE PDP-10
2199 004712 000400 ;173712 000400
2200 004714 000000 ;173714 000000 2S: HALT ;HALT DUE TO "FATAL" ERROR
2201 004716 012762 ;173716 012762 8S: MOV #DRESET,DIAG2(R2) ;RESET AFTER POSSIBLE PDP-11
2202 004720 000100 ;173720 000100
2203 004722 000032 ;173722 000032
2204 ; MEMORY PARITY ERROR OR NON-EX-MEM ERROR, AND ALSO TURN OFF
2205 ; "-10 RINGING -11'S DOORBELL".
2206 004724 005062 ;173724 005062 3S: CLR T011AD(R2) ;START INPUTTING AT LOCATION 0
2207 004726 000022 ;173726 000022
2208 004730 012762 ;173730 012762 MOV #INT10!<<-DTEWDC>&7777>,T011BC(R2) ;READ IN 256 WORDS
2209 004732 107400 ;173732 107400
2210 004734 000016 ;173734 000016
2211 004736 032711 ;173736 032711 1S: BIT #T011DB,(R1) ;HAS THE -10 RUNG THE -11'S DOORBELL?
2212 004740 004000 ;173740 004000
2213 004742 001775 ;173742 001775 BEQ 1S ;LOOP UNTIL IT HAS.
2214 004744 132711 ;173744 132711 4S: BITB #T011DN!T011ER,(R1) ;IS THE TRANSMISSION FINISHED?
2215 004746 000202 ;173746 000202
2216 004750 001775 ;173750 001775 BEQ 4S ;LOOP UNTIL IT IS FINISHED
2217 004752 100003 ;173752 100003 BPL 5S ;IF "T011DN" ISN'T ON, "T011ER" MUST BE ON
2218 004754 022715 ;173754 022715 CMP #000240,(R5) ;CHECK FOR BIT PATTERN IN LOCATION ZERO
2219 004756 000240 ;173756 000240
2220 004760 001403 ;173760 001403 BEQ 9S ;UNLESS THERE IS A "NOP" IT IS AN ERROR
2221 004762 012711 ;173762 012711 5S: MOV #T010DB,(R1) ;SIGNAL THE -10 THAT THERE WAS AN ERROR
2222 004764 000400 ;173764 000400
2223 004766 000000 ;173766 000000 12S: HALT ;THIS ERROR HALT IS BECAUSE EITHER "T011ER"
2224 ; IS ON, OR BECAUSE THE BIT PATTERN READ INTO LOCATION ZERO WASN'T "000240".
2225 004770 012762 ;173770 012762 9S: MOV #DRESET,DIAG2(R2) ;SIGNAL THE -10 THAT EVERYTHING IS OK
2226 004772 000100 ;173772 000100
2227 004774 000032 ;173774 000032
2228 004776 END.YD:
2229 004776 000115 ;173776 000115 JMP (R5) ;JUMP TO LOCATION ZERO
2230 ; 000001 .END
    
```

M04

2231 005000 MAP.YF:

: 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

2255	177776	PS=177776
2256	177570	SWR=177570
2257	000000	PRO=0*40
2258	000040	PR1=1*40
2259	000100	PR2=2*40
2260	000140	PR3=3*40
2261	000200	PR4=4*40
2262	000240	PR5=5*40
2263	000300	PR6=6*40
2264	000340	PR7=7*40
2265	000001	BIT0=000001
2266	000002	BIT1=000002
2267	000004	BIT2=000004
2268	000010	BIT3=000010
2269	000020	BIT4=000020
2270	000040	BIT5=000040
2271	000100	BIT6=000100
2272	000200	BIT7=000200
2273	000400	BIT8=000400
2274	001000	BIT9=001000
2275	002000	BIT10=002000
2276	004000	BIT11=004000
2277	010000	BIT12=010000
2278	020000	BIT13=020000
2279	040000	BIT14=040000
2280	100000	BIT15=100000
2281	177400	HIBYTE=177400

: PROCESSOR STATUS REGISTER
: FRONT PANEL SWITCH REGISTER
: PRIORITY LEVEL 0
: PRIORITY LEVEL 1
: PRIORITY LEVEL 2
: PRIORITY LEVEL 3
: PRIORITY LEVEL 4
: PRIORITY LEVEL 5
: PRIORITY LEVEL 6
: PRIORITY LEVEL 7

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

DIRECTIVE FUNCTION CODES

000001 DR.DTE=1. ;DTE EXAMINE/DEPOSIT/INITALIZE/DOORBELL FUNCTIONS

DTE FUNCTION CODES (LOW ORDER BY BYTE)

000001 DF.DOR=1 ;DOOR BELL FUNCTION CODE
 000002 DF.OFF=2 ;DTE OFF FUNCTION
 000003 DF.ON=3 ;DTE ON FUNCTION
 000004 DF.DMG=4 ;DEPOSIT MY GENERAL FUNCTION
 000005 DF.EMG=5 ;EXAMINE MY GENERAL FUNCTION
 000006 DF.EMN=6 ;EXAMINE MY FOR N FUNCTION
 000007 DF.DMN=7 ;DEPOSIT MY FOR N FUNCTION
 000010 DF.EHG=10 ;EXAMINE HIS GEN SECTION FUNCTION
 000011 DF.EHM=11 ;EXAMINE HIS SECTION FOR ME FUNCTION
 000012 DF.KLR=12 ;DIAGNOSTIC KL READ
 000013 DF.KLW=13 ;DIAGNOSTIC KL WRITE (FUNCTION 13)
 000014 DF.KLX=14 ;DIAGNOSTIC KL EXECUTE (FUNCTION 14)
 000015 DF.PEX=15 ;PRIVILEGED EXAMINE (FUNCTION 15)
 000016 DF.PDP=16 ;PRIVILEGED DEPOSIT (FUNCTION 16)

CRASH CODES

000001 CC.ILD=1 ;ILLEGAL DIRECTIVE
 000002 CC.EMT=2 ;ILLEGAL EMT
 000003 CC.IDI=3 ;ILLEGAL DTE INTERRUPT
 000004 CC.IOT=4 ;IOT TRAP
 000005 CC.RES=5 ;RESERVED INSTRUCTION TRAP
 000006 CC.TBT=6 ;T BIT OR BPT TRAP
 000007 CC.TRP=7 ;TRAP INSTRUCTION TRAP
 000010 CC.TO4=10 ;TRAP TO 4
 000011 CC.UNT=11 ;ILLEGAL TRAP (UNKNOWN TRAP)
 000012 CC.MPE=12 ;MEMORY PARITY ERROR
 000013 CC.NPF=13 ;RESTRICTED FRONT CAN'T EXECUTE BOOT PROTOCOL
 000014 CC.PTB=14 ;PROTOCOL (PRIMARY) BROKEN
 000015 CC.CST=15 ;CLOCK STOPPED
 000016 CC.ILC=16 ;ILLEGAL COMMAND
 000017 CC.IPO=17 ;INPUT TTY OVERFLOW
 000020 CC.IAS=20 ;INCORRECT VALUE IN .SERFG
 000021 CC.NCE=21 ;NOT ENOUGH ENTRIES IN CLOCK QUEUE
 000022 CC.PIT=22 ;CAN'T EXIT PERMANENT TASK
 000023 CC.UMP=23 ;LOAD REQUEST NOT IMPL YET
 000024 CC.EPE=24 ;E BUS PARITY ERROR
 000025 CC.NDE=25 ;NOT ENOUGH ENTRYS FOR DTE20
 000026 CC.DEX=26 ;DEXDONE TIMEOUT
 000027 CC.TET=27 ;TO TEN ERROR
 000030 CC.ETE=30 ;TO ELEVEN ERROR
 000031 CC.MTF=31 ;MARK TIME FAILURE
 000032 CC.NON=32 ;NOT ENOUGH NODES
 000033 CC.TSP=33 ;TEN STOPPED
 000034 CC.UIE=34 ;UNIMPLEMENTED FUNCTION
 000035 CC.ILQ=35 ;ILLEGAL QUEUE

2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
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
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335

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

2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389

GENERAL PROCESSOR DEFINITIONS
 : 000340 PRI7=340 ;PROCESSOR PRIORITY 7
 DTE20 REGISTER DEFINITIONS
 THESE LABELS ARE THOSE USED IN THE FRONT END INTERFACE SPEC
 EXCEPT STATUS WHICH CONFLICTS WITH PROTOCOL SPEC
 PDM# 200-200-012-00

174400	DLYCNT=174400	: DELAY COUNT WORD
174402	DEXM03=174402	: DEPOSIT OR EXAMINE WORD 3
174404	DEXM02=174404	: DEPOSIT OR EXAMINE WORD 2
174406	DEXM01=174406	: DEPOSIT OR EXAMINE WORD 1
174410	TENAD1=174410	: TEN ADDRESS WORD 1
174412	TENAD2=174412	: TEN ADDRESS WORD 2
174414	TO10BC=174414	: TO-10 PDP-11 MEMORY ADDRESS
174416	TO11BC=174416	: TO-11 BYTE COUNT
174420	TO10AD=174420	: TO-10 PDP-11 MEMORY ADDRESS
174422	TO11AD=174422	: TO-11 PDP-11 MEMORY ADDRESS
174424	TO10DT=174424	: TO-10 PDP-11 DATA WORD
174426	TO11DT=174426	: TO-11 PDP-11 DATA WORD
174430	DIAG1=174430	: DIAGNOSTIC WORD 1
174432	DIAG2=174432	: DIAGNOSTIC WORD 2
174434	STAT=174434	: STATUS WORD
174436	DIAG3=174436	: DIAGNOSTIC WORD 3

EXTERNAL PAGE DEFINITIONS (DEVICE DEFINITIONS)

DTE DEFINITIONS

REGISTER BIT DEFINITIONS

TENAD1 DEFINITIONS

010000	DEP=010000	: DEPOSIT (BIT 12)
004000	PRTOFF=004000	: EXAMINE/DEPOSIT PROTECT OFF
100000	PHYS=100000	: PHYSICAL EXAMINE

TO11BC DEFINITIONS

100000	IFLOP=100000	: I FLIPFLOP BIT
040000	ZSTOP=040000	: ZSTOP
020000	TO11BM=020000	: TO 11 BYTE MODE

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

DIAG1 DEFINITIONS

```

004000 DS04=004000 ;KL CLOCK ERROR STOP
002000 DS05=002000 ;RUN
001000 DS06=001000 ;HALT
000400 DEX=000400 ;DEPOSIT OR EXAMINE MAJOR STATE
000200 T010=000200 ;TO 10
000200 DFUNC=000200
000100 T011=000100 ;T0-11 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 DCONST=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

```

2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
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
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474

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 PROBE
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
;APGE 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.

```


E05

2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528

```

;BMB73F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 1
CONTROL RAM LOAD FUNCTIONS
057000 .LCRM1=57*1000 ;EBUS CRAM
      :08-11 00-03
      :14-17 04-07
      :20-23 08-11
      :26-29 12-15
      :32-35 16-19
056000 .LCRM2=56*1000 ;08-11 20-23
      :14-17 24-27
      :20-23 28-31
      :26-29 32-35
      :32-35 36-39
055000 .LCRM3=55*1000 ;08-11 40-43
      :14-17 44-47
      :20-23 48-51
      :26-29 52-55
      :32-35 56-59
054000 .LCRM4=54*1000 ;08 60
      :10 62
      :14 64
      :16 66
      :20 68
      :22 70
      :26 72
      :28 74
      :32 76
      :34 78
053000 .LCRM5=53*1000 ;01-05 DISP 00-04
052000 .LCRDL=52*1000 ;01-05 CRAM DIAG ADDRES 00-04
051000 .LCRDR=51*1000 ;00-05 CRAM DIAG ADR 05-10

DRAM LOAD FUNCTIONS
060000 .LDRM1=60*1000 ;12-14 DRAM A00-02, EVEN ADDRESSES
      :15-17 DRAM B00-02, EVEN ADDRESSES
061000 .LDRM2=61*1000 ;12-14 DRAM A00-02, ODD ADDRESSES
      :15-17 DRAM B00-02, ODD ADDRESSES
062000 .LDRM3=62*1000 ;14-17 COMMON J01-04
063000 .LDRJV=63*1000 ;15-17 J08-10, EVEN ADDRESSES
      :12 PARIT BIT, EVEN ADDRESSES
064000 .LDRJD=64*1000 ;14 COMMON J07 (NOTE -- J05,6 DO NOT EXIST)
      :15-17 J08-10, ODD ADDRESSES
      :12 PARITY BIT, ODD ADDRESSES

IR, DRAM CONTROL FUNCTIONS
065000 .DSIOJ=65*1000 ;DISABLES SPECIAL DECODE OF OPCODES 254,7XX
066000 .DSACF=66*1000 ;DISABLE IR AC OUTPUTS
067000 .EIOJA=67*1000 ;ENABEL KL STYLE DECODING OF CODES AND AC'S

070000 .INICL=70*1000 ;INIT CHANNELS
071000 .WRMBX=71*1000 ;WRITE M-BOX
  
```

F05

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 56
 DZBMD.P11 ROM CONTENTS TABLES

MACY11 27(657) 22-AUG-75 10:30 PAGE 1

2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563
2564
2565
2566
2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581

```

;BMB73F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23)
;
; 076000 .MEMRS=76*1000 ;SET KL10 MEM RESET FLOP
;
; 147000 .RCRM1=147*1000 ;READ C-RAM BITS 0-19
; 146000 .RCRM2=146*1000 ;READ C-RAM BITS 20-39
; 145000 .RCRM3=145*1000 ;READ C-RAM BITS 40-59
; 144000 .RCRM4=144*1000 ;READ C-RAM BITS 60-79
;
; 141000 .RCSPF=141*1000 ;READ SPEC FIELD OF C-RAM
;
; 135000 .RDJ71=135*1000 ;READ J07-J10 OF D-RAM
; 134000 .RDJ14=134*1000 ;READ J01-J04 OF D-RAM
; 133000 .RDMAB=133*1000 ;READ A & B FIELD OF D-RAM
;
; 164000 .CSHRG=164*1000
; 102000 .GFNR=102*1000
;
; NOTE CONSOLE SOFTWARE MUST PERFORM THIS AS A PART OF
; MASTER RESET CODE
;
; LOAD AR FUNCTION
;
; 077000 .LDAR=77*1000 ;LOAD THE AR FROM EBUS 0-35
;
; 150000 .PCAB1=150*1000 ;PC-ADDRESS BREAK REGISTERS
; 151000 .PCAB2=151*1000
; 152000 .PCAB3=152*1000
; 153000 .PCAB4=153*1000
;
; DIAG3 DEFINITIONS
;
; 100000 SWSLLT=100000 ;SWAP SELECT LEFT
; 040000 DPS4=040000 ;PARITY
; 000040 SCD=000040 ;SHIFT CAPTURED DATA
; 000020 DUPE= 000020 ;DATO UNIBUS PARITY ERROR
; 000020 CDD=000020 ;CLEAR DUPE AND DURE ERROR FLAGS
; 000010 MEP=000010 ;WRITE EVEN (BAD) PARITY
; 000004 DURE=000004 ;DATO UNIBUS RECEIVE ERROR
; 000002 NUPE=000002 ;NPR UNIBUS PARITY ERROR
; 000002 CNUPE=000002 ;CLEAR NUPE
; 000001 T010BM=000001 ;T0-10 BYTE TRANSFER MODE
;
; DIAG2 DEFINITIONS
;
; 100000 RFMAD0=100000 ;RFM ADDRESS BIT 0
; 040000 RFMAD1=040000 ;RFM ADDRESS BIT 1
; 040000 EDONES=040000 ;EBUS DONE
; 020000 RFMAD2=020000 ;RFM ADDRESS BIT 2
; 010000 RFMAD3=010000 ;RFM ADDRESS BIT 3
; 000100 DRESET=000100 ;DTE RESET
  
```

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

2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623
2624
2625
2626
2627
2628
2629
2630
2631
2632
2633
2634
2635

```

:
:
:          STAT DEFINITIONS
:
:          100000 TO10DN=100000 :TO-10 NORMAL TERMINATION
:          100000 DON10S=100000 :NORMAL TERMINATION (DONE) TO 10
:          040000 DON10C=040000 :TO-10 NORMAL TERMINATION STATUS
:          020000 TO10ER=020000 :TO-10 ERROR TERMINATION
:          020000 ERR10S=020000 :ERROR TERMINATION STATUS
:          010000 RAM1S0=010000 :RAM IS ZEROS
:          010000 ERR10C=010000 :CLEAR TO-10 ERROR TERMINATION
:          004000 TO11DB=004000 :-10 REQUESTED -11 INTERRUPT
:          004000 INT11S=004000 :REQ 11 STATUS
:          002000 DXWRD1=002000 :DEXWORD 1
:          002000 INT11C=002000 :-10 REQUESTS -11 INTERRUPT STATUS
:          001000 MPE11=001000 :-11 MEMORY PARITY ERROR
:          001000 PERCLR=001000 :CLEAR -11 MEMORY PARITY ERROR FLAG STATUS
:          000400 TO10DB=000400 :-11 REQUEST -10 INTERRUPT
:          000400 INT10S=000400 :REQUEST -10 INTERRUPT STATUS
:          000200 TO11DN=000200 :TO-11 TRANSFER DONE
:          000200 DON11S=000200 :TO-11 NORMAL TERMINATION FLAG STATUS
:          000100 EBSEL=000100 :E BUFFER SELECT
:          000100 DON11C=000100 :TO-11 NORMAL TERMINATION FLAG STATUS
:          000040 NULSTP=000040 :NULL STOP
:          000040 INTRON=000040 :11 INTERRUPT ENABLE
:          000020 BPARER=000020 :EBUS PARITY ERROR
:          000020 EBUSPC=000020 :EBUS PARIT ERROR
:          000010 RM=000010 :RESTRICTED MODE
:          000010 INTR0F=000010 :DISABLE PDP11 INTERRUPT
:          000004 DEXDON=000004 :DEPOSIT/EXAMINE DONE
:          000004 EBUSPS=000004 :EBUS PARITY ERROR SET
:          000002 TO11ER=000002 :TO-11 BYTE ERROR TERMINATION
:          000002 ERR11S=000002 :TO-11 ERROR TERMINATION FLAG STATUS
:          000001 INTSON=000001 :INTERRUPTS ON
:          000001 ERR11C=000001 :CLEAR TO-11 ERROR TERMINATION FLAG STATUS
    
```

DTE20 COMMUNICATION AREA OFFSETS (WORD NAMES)

```

:
:          000000 PIDENT=0 :PROCESSOR IDENTIFICATION WORD
:          000001 CHNPNT=1 :POINTER TO COMM AREA OF NEXT PROCESSOR (CIRC LIST)
:          000002 CYCLS=2 :CLOCK CPS COUNT
:          000003 T00=3 :TIME OF DAY
:          000004 DATE=4 :DATE
:          000005 PSW1=5 :PROCESSOR STATUS WORD1
:          000006 PSW2=6 :PROCESSOR STATUS WORD2
:          000007 PSW3=7 :PROCESSOR STATUS WORD3
:          000010 PSW4=10 :PROCESSOR STATUS WORD4
:          000011 PSW5=11 :PROCESSOR STATUS WORD5
:          000012 PSW6=12 :PROCESSOR STATUS WORD6
:          000013 PSW7=13 :PROCESSOR STATUS WORD7
:          000014 PSW10=14 :PROCESSOR STATUS WORD10
:          000015 PSW11=15 :PROCESSOR STATUS WORD11
:          000016 PSW12=16 :PROCESSOR STATUS WORD12
    
```

H05

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 58
 DZBMD.P11 ROM CONTENTS TABLES

```

2636 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 1
2637 ;
2638 ; 000017 PSHW13=17 ; PROCESSOR STATUS WORD13
2639 ; 000020 FORPRO=20 ; FOR PROCESSOR IDENTIFICATION WORD
2640 ; 000021 PROPNT=21 ; POINTER TO COMM AREA OF THE PROCESSOR ASSOC WITH THIS BLOCK
2641 ; 000022 STATUS=22 ; COMMUNICATION STATUS WORD
2642 ; 000023 QSIZE=23 ; QUEUE SIZE WORD
2643 ; ; CTY0CW=24 ; CTY #0 COMMAND WORD
2644 ; ; CTY0RW=25 ; CTY #0 RESPONSE WORD
2645 ; ; CTY1CW=26 ; CTY #1 COMMAND WORD
2646 ; ; CTY1RW=27 ; CTY #1 RESPONSE WORD
2647 ; ; MISCH=30 ; MISCELLANEOUS COMMAND WORD FOR NON-QUEUE PROTOCOL
2648 ; ; MISRW=31 ; MISCELLANEOUS RESPONSE WORD
2649 ; ; 000032 UNASG1=32 ; UNASSIGNED WORD1
2650 ; ; 000033 UNASG2=33 ; UNASSIGNED WORD2
2651 ; ; 000034 UNASG3=34 ; UNASSIGNED WORD3
2652 ; ; 000035 UNASG4=35 ; UNASSIGNED WORD4
2653 ; ; 000036 UNASG5=36 ; UNASSIGNED WORD5
2654 ; ; 000037 UNASG6=37 ; UNASSIGNED WORD6
2655 ;
2656 ; ; EPT ADDRESSES AS DEFINED IN BOOTS FOR USE IN THE
2657 ; ; SECONDARY PROTOCOL
2658 ;
2659 ; ; 000444 DTEFLG=444 ; OPERATION COMPLETE FLAG
2660 ; ; 000450 DTEF11=450 ; PDP-10 FROM PDP-11 ARGUMENT
2661 ; ; 000451 DTECMD=451 ; PDP-10 TO PDP-11 COMMAND WORD
2662 ; ; 000455 DTEMTD=455 ; MONITOR TTY OUTPUT COMPLETE FLAG
2663 ; ; 000456 DTEMTI=456 ; MONITOR TTY INPUT FLAG
2664 ;
2665 ; ; STATUS DEFINITIONS
2666 ;
2667 ; ; 000001 TOIT=1 ; IN PROGRESS OF PROCESSING QUEUE
2668 ; ; 000002 TOIP=2 ; TO HIM INDIRECT IN PROGRESS
2669 ;
  
```

```

2670 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 2
2671 ;
2672 ;
2673 ;
2674 ;
2675 ;
2676 ; 000040 ROTOR7= 40 ;SAVE R0 TO R7 IN 40 TO 56
2677 ;
2678 ; 000130 DTESAV= 130 ;SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
2679 ; ; IN LOCATIONS 130-156
2680 ;
2681 ; 000012 RETRY= 10. ;DO 10 RETRIES BEFORE HALTING
2682 ;
2683 ; 173000 ROMORG= 173000 ;ROM STARTS AT 773000
2684 ;
2685 ; ESTABLISH ROM ORIGIN
2686 ;
2687 ; 173000 .=ROMORG
  
```

```

2688 ;BMB73F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 3
2689 ;
2690 ; EXTERNAL BUTTONS #1, #2, #3
2691 ;
2692 ; BUTTON #1 -- LOAD USING SWITCH REGISTER
2693 ;
2694 005000 010037 ;:173000 010037 BUTON1: MOV RO,ROTOR7+0 ;SAVE RO IN LOCATION 40
2695 005002 000040 ;:173002 000040
2696 005004 013700 ;:173004 013700 MOV SWR,RO ;GET SWITCH REGISTER
2697 005006 177570 ;:173006 177570
2698 005010 032700 ;:173010 032700 BIT #BIT0,RO ;IS LOW-ORDER BIT SET?
2699 005012 000001 ;:173012 000001
2700 005014 001007 ;:173014 001007 BNE BUTONX ;YES-- LOOK AT CONTENTS
2701 005016 000557 ;:173016 000557 BR REGSAV ;NO-- SAVE R1-R7 IN 42-56, GO TO ADDRESS IN RO (FROM SWR
2702 ;
2703 ; BUTTON #3 -- LOAD BOOT FROM RX11 FLOPPY DISK
2704 ;
2705 005020 005000 ;:173020 005000 BUTON3: CLR RO ;SAY LOAD FROM FLOPPY, UNIT 0
2706 005022 000404 ;:173022 000404 BR BUTONX ;GO TO COMMON CODE FOR 3 BUTTONS
2707 ;
2708 ; REQUIRED POWER-FAIL VECTOR
2709 ;
2710 005024 173000 ;:173024 173000 .WORD ROMORG,PR7
2711 005026 000340 ;:173026 000340
2712 ;
2713 ; BUTTON #2 -- LOAD BOOT FROM RPO4 DISK
2714 ;
2715 005030 012700 ;:173030 012700 BUTON2: MOV #BIT7,RO ;BIT 7 MEANS LOAD FROM RPO4
2716 005032 000200 ;:173032 000200 BR BUTONX ;FALL INTO COMMON CODE
2717 ;
2718 ; RO IS SAVED IN R5 AS THE PARAMETER WORD PASSED TO BOOT
2719 ; AND CONTAINS ONE OF THE FOLLOWING:
2720 ;
2721 ; BIT 0 = 1 IF FROM SWITCH REGISTER
2722 ; BIT 7 = 0 LOAD FROM RX11 FLOPPY DISK
2723 ; BIT 7 = 1 LOAD FROM RPO4 DISK
2724 ; BIT 15 = 1 INDEFINITE RETRY
2725 ;
2726 ; NOTE THAT IF BUTTON #4 IS PRESSED, R5 WILL CONTAIN BIT 0 = 0, BIT 15 = 1
2727 ;
2728 ;
2729 005034 010005 ;:173034 010005 BUTONX: MOV RO,R5 ;SAVE PARAMETER FOR BOOT
2730 005036 106300 ;:173036 106300 ASLB RO ;LEFT-ALIGN SPEED FIELD IN LOW BYTE
2731 005040 122700 ;:173040 122700 CMPB #16*BIT4,RO ;IS SPEED 16 OR 17?
2732 005042 000340 ;:173042 000340
2733 005044 101404 ;:173044 101404 BLOS 10$ ;YES-- UNIT FIELD IS UNIT # TO BOOT FROM
2734 005046 122700 ;:173046 122700 CMPB #3*BIT4,RO ;IS SPEED 0, 1, OR 2?
2735 005050 000060 ;:173050 000060
2736 005052 101001 ;:173052 101001 BHI 10$ ;YES-- UNIT IS UNIT TO USE
2737 005054 005000 ;:173054 005000 CLR RO ;NO-- USE UNIT #0
2738 ;
2739 005056 000300 ;:173056 000300 ios: SWAB RO ;GET UNIT # IN LOW BYTE
2740 005060 042700 ;:173060 042700 BIC #1C7,RO ;TRIM TO 3 BITS 2, 1, 0
2741 005062 177770 ;:173062 177770

```

K05

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 61
DZBMD.P11 ROM CONTENTS TABLES

2742										
2743										
2744										
2745	005064	105705	:	173064	105705		TSTB	RS		:WHERE SHOULD WE BOOT FROM?
2746	005066	100550	:	173066	100550		BMI	RPBOOT		:BIT 7 = 1 -- BOOT FROM RPO4 DISK
2747							BR	RXBOOT		:BIT 7 = 0 -- BOOT FROM RX11 FLOPPY DISK

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

;BM873F - 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
  
```

2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793

M05

2794				
2795				
2796				
2797			:173070	
2798	005070	012703	:173070	012703
2799	005072	000401	:173072	000401
2800	005074	005700	:173074	005700
2801	005076	001402	:173076	001402
2802	005100	012700	:173100	012700
2803	005102	000020	:173102	000020
2804	005104	052700	:173104	052700
2805	005106	000007	:173106	000007
2806				

;; 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

```

N05

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 64
 DZBMD.P11 ROM CONTENTS TABLES

```

2807      ;BMB73F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23)  MACY11 27(657) 22-AUG-75 10:30 PAGE 4
2808      ;
2809      ;
2810      ; HERE TO START RX11 ON A TRANSFER, EITHER DUMP OR BOOT
2811      ;
2812      005110 012706 ;173110 012706 RXSTRT: MOV      #RETRY,SP      ;SET RETRY COUNT
2813      005112 000012 ;173112 000012
2814      005114 012701 ;173114 012701          MOV      #RXEPA+RXCS,R1 ;ADDRESS CONTROL STATUS REGISTER FOR RX11
2815      005116 177170 ;173116 177170
2816      ;
2817      ;
2818      ; BR      RXRTRY      ;FALL THROUGH RETRY CHECK
2819      ;
2820      ; HERE ON ERROR TO RETRY
2821      ;
2822      005120 005705 ;173120 005705 RXRTRY: TST      R5      ;INDEFINITE RETRY?
2823      005122 100402 ;173122 100402          BMI      10$      ;YES-- TRY FAITHFULLY
2824      005124 005306 ;173124 005306          DEC      SP      ;NO-- DECREMENT RETRY COUNT
2825      005126 002475 ;173126 002475          BLT      RXEHLT    ;GIVE UP IF RUN OUT
2826      ;
2827      10$:      RESET      ;CLEAR THE WORLD
2828      005130 000005 ;173130 000005          CLR      R4      ;ALWAYS START TRANSFER AT LOCATION ZERO
2829      005132 005004 ;173132 005004          MOV      R3,R2   ;GET START TRACK AND SECTOR
2830      005134 010302 ;173134 010302          BIT      #RXDONE,(R1) ;WAIT UNTIL READY FOR FUNCTION
2831      005136 032711 ;173136 032711
2832      005138 000040 ;173138 000040          BEQ      20$      ;NOT YET-- WAIT
2833      005140 001775 ;173140 001775          TST      R0      ;THIS WRITE?
2834      005142 005700 ;173142 005700          BMI      RXFLSL   ;YES-- FILL SILO BEFORE WRITE
2835      005144 100454 ;173144 100454          BR       RXPERF   ;NO-- JUST DO FIRST READ
2836      ;
2837      ; HERE TO PERFORM READ OR WRITE, AS SPECIFIED IN R0
2838      ;
2839      005150 110011 ;173150 110011 RXPERF: MOVB     R0,(R1) ;DO READ OR WRITE
2840      005152 105711 ;173152 105711 10$:      TSTB     (R1)      ;READY?
2841      005154 100376 ;173154 100376          BPL      10$      ;NO-- WAIT
2842      005156 110261 ;173156 110261          MOVB     R2,RXDB(R1) ;SET SECTOR #
2843      005158 000002 ;173158 000002
2844      005160 105711 ;173160 105711 20$:      TSTB     (R1)      ;READY FOR TRACK?
2845      005162 100376 ;173162 100376          BPL      20$      ;NO-- WAIT
2846      005164 000302 ;173164 000302          SWAB     R2      ;YES-- GET TRACK #
2847      005166 110261 ;173166 110261          MOVB     R2,RXDB(R1) ;SET IT
2848      005168 000002 ;173168 000002
2849      005170 000302 ;173170 000302
2850      005172 173174 ;173172 173174 30$:      SWAB     R2      ;RESTORE HIGH TRACK, LOW SECTOR
2851      005174 000302 ;173174 000302          BIT      #RXERR!RXDONE,(R1) ;DONE OR ERROR?
2852      005176 032711 ;173176 032711
2853      005178 100040 ;173178 100040          BEQ      30$      ;NO-- WAIT
2854      005180 001775 ;173180 001775          BMI      RXRTRY   ;YES-- ERROR IN FUNCTION
2855      005182 100745 ;173182 100745
  
```

2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903

```

005206 005700 :173206 005700
005210 100421 :173210 100421

005212 012711 :173212 012711
005214 000003 :173214 000003

005216 132711 :173216 132711
005220 000240 :173220 000240
005222 000402 :173222 000402

005224 173000 :173224 173000
005226 000340 :173226 000340

005230 001772 :173230 001772
005232 100003 :173232 100003
005234 116124 :173234 116124
005236 000002 :173236 000002
005240 000766 :173240 000766

:173242
005242 122222 :173242 122222
005244 022704 :173244 022704
005246 001000 :173246 001000
005250 101337 :173250 101337
005252 005007 :173252 005007

:173254
005254 005202 :173254 005202
005256 122702 :173256 122702
005260 000032 :173260 000032
005262 103003 :173262 103003
005264 105002 :173264 105002
005266 062702 :173266 062702
005270 000401 :173270 000401
005272 022704 :173272 022704
005274 160000 :173274 160000
005276 101516 :173276 101516
  
```

```

: DISK TRANSFER COMPLETE WITH NO ERRORS
:
: TST R0 ; THIS A WRITE?
: BMI RXMDON ; YES-- SEE IF DONE WITH DUMP
: BR RXMSL ; NO-- READ-- EMPTY SILO

: READ COMPLETED-- EMPTY SILO TO MEMORY
RXMSL: MOV #RXEMPT+RXGO,(R1) ; START EMPTY

10S: BITB #RXTREQ!RXDONE,(R1) ; READY FOR WORD, OR TRANSFER DONE?
: BR 20S ; BRANCH AROUND VECTOR

: REQUIRED POWER-FAIL VECTOR
: .WORD ROMORG,PR7

20S: BEQ 10S ; NOT READY-- WAIT SOME MORE
: BPL RXRDON ; DONE-- GET ANOTHER SECTOR
: MOVB RXDB(R1),(R4)+ ; NOT DONE-- GET A BYTE FROM SILO TO MEMORY
: BR 10S ; WAIT FOR NEXT BYTE

: SILO EMPTIED-- SEE IF WE ARE DONE WITH BOOTING
RXRDON:
SSS=.
: CMPB (R2)+,(R2)+
: CMP #256. #2,R4 ; HAVE WE READ ENOUGH?

: BHI RXPERF ; NO-- READ SOME MORE
CLRPC: CLR PC ; YES-- GO TO LOCATION ZERO

: WRITE COMPLETED-- SEE IF DONE DUMPING
RXMDON:
SSS=.
: INC R2
: CMPB #26.,R2 ; THIS LAST SECTOR ON TRACK?

: BHIS 10S ; NO-- KEEP ON GOING
: CLRB R2 ; YES-- CLEAR SECTOR ADDRESS
: ADD #BITB!BIT0,R2 ; BUMP TO NEXT TRACK, SECTOR 1

10S: CMP #1024.*28.*2,R4 ; ARE WE DONE WITH 28 K?

: BLOS HALT ; YES-- GO HALT WITH R0= 0 IN DISPLAY
: BR RXFLSL ; NO-- FILL SILO WITH NEXT SECTOR
  
```

```

2904
2905
2906
2907 005300 012711 ;173300 012711
2908 005302 000001 ;173302 000001
2909
2910 005304 132711 ;173304 132711
2911 005306 000240 ;173306 000240
2912 005310 001775 ;173310 001775
2913 005312 100316 ;173312 100316
2914 005314 112461 ;173314 112461
2915 005316 000002 ;173316 000002
2916 005320 000771 ;173320 000771
2917
2918
2919
2920 005322 012711 ;173322 012711
2921 005324 000017 ;173324 000017
2922 005326 032711 ;173326 032711
2923 005330 000040 ;173330 000040
2924 005332 001775 ;173332 001775
2925 005334 016100 ;173334 016100
2926 005336 000002 ;173336 000002
2927 005340 000476 ;173340 000476
2928
2929
2930
2931
2932
2933
2934
2935
2936 005342 012703 ;173342 012703
2937 005344 035401 ;173344 035401
2938
2939 005346 012700 ;173346 012700
2940 005350 100005 ;173350 100005
2941 005352 005005 ;173352 005005
2942 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
:            MOVB   (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            ;GET ERROR REGISTER
:            MOV    RXDB(R1),RO   ;
:            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#BIT8>!<RXDSCT#BIT0>,R3
:
:            MOV    #BIT15!RXWRIT+RXGO,RO ;DO A WRITE
:            CLR    R5            ;CLEAR INDEFINITE RETRY BIT
:            BR     RXSTRT        ;START DUMP GOING

```

D06

: BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 5

: REGISTER SAVE ROUTINE

: REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS R0-R7
: IN MEMORY AT 40-56 (LOCATION ROTOR7).

: CALLING SEQUENCE:

: MOV R0, ROTOR7+0
: MOV #RET, R0
: BR REGSAV
: RET: <RETURN HERE>

: ALL REGISTERS RESTORED

2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022
3023
3024
3025
3026
3027
3028
3029
3030
3031
3032
3033
3034
3035
3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064
3065
3066
3067
3068
3069
3070
3071

005356 010037 :: 173356 010037
005360 000056 :: 173360 000056
005362 012700 :: 173362 012700
005364 000056 :: 173364 000056
005366 010640 :: 173366 010640
005370 010540 :: 173370 010540
005372 010440 :: 173372 010440
005374 010340 :: 173374 010340
005376 010240 :: 173376 010240
005400 010140 :: 173400 010140
005402 014000 :: 173402 014000
005404 000177 :: 173404 000177
005406 004446 :: 173406 004446

```
REGSAV: MOV      R0, ROTOR7+16      ; SAVE R0 AS PC IN 56
         MOV      #ROTOR7+16, R0    ; R0 NOW POINTS TO 56

         MOV      SP, -(R0)          ; SAVE SP IN 54
         MOV      R5, -(R0)          ; SAVE R5 IN 52
         MOV      R4, -(R0)          ; SAVE R4 IN 50
         MOV      R3, -(R0)          ; SAVE R3 IN 48
         MOV      R2, -(R0)          ; SAVE R2 IN 46
         MOV      R1, -(R0)          ; SAVE R1 IN 44
         MOV      -(R0), R0          ; RESTORE R0 FROM 40
         JMP      @ROTOR7+16        ; GO TO SAVED PC
```

```

2972 ;BMB73F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 6
2973 ;
2974 ; RPO4 DISK BOOTSTRAP AND DUMP ROUTINES
2975 ;
2976 ; RPO4 REGISTER DEFINITIONS
2977 ;
2978 ; 176700 RPEPA= 176700 ;EXTERNAL PAGE ADDRESS OF RPO4 REGISTERS
2979 ;
2980 ; 000000 RPCS1= 0 ;OFFSET FOR CSR #1
2981 ; 040000 RPTRE= BIT14 ;TRANSFER ERROR
2982 ; 020000 RPMCPE= BIT13 ;MASSBUS CONTROL PARITY ERROR
2983 ; 004000 RPDVA= BIT11 ;DRIVE AVAILABLE (TO -11)
2984 ; 000200 RPRDY= BIT7 ;FUNCTION COMPLETE
2985 ; 000076 RPFUNC= BITS!BIT4!BIT3!BIT2!BIT1 ;FUNCTION:
2986 ; 000020 RPPRST= 20 ; READ-IN PRESET
2987 ; 000060 RPWRIT= 60 ; WRITE DATA
2988 ; 000070 RPREAD= 70 ; READ DATA
2989 ; 000001 RPGO= BIT0 ;GO
2990 ; 000002 RPWC= 2 ;WORD COUNT REGISTER
2991 ; 000006 RPDA= 6 ;TRACK (HIGH BYTE) SECTOR (LOW BYTE)
2992 ; 000010 RPCS2= 10 ;CONTROL AND STATUS REGISTER #2
2993 ; 000007 RPUNIT= BIT2!BIT1!BIT0 ;UNIT #
2994 ; 000012 RPDS= 12 ;DRIVE STATUS REGISTER
2995 ; 100000 RPATA= BIT15 ;ATTENTION ACTIVE
2996 ; 040000 RPERR= BIT14 ;DRIVE ERROR
2997 ; 000034 RPDC= 34 ;DESIRED CYLINDER
2998 ;
2999 ;
3000 ; PARAMETERS
3001 ;
3002 ; 000000 RPBCYL= 0. ;BOOT FROM CYLINDER 0
3003 ; 000000 RPBTRK= 0. ; TRACK 0
3004 ; 000000 RPBSC= 0. ; SECTOR 0
3005 ;
3006 ; 000631 RPDCYL= 409. ;DUMP TO CYLINDER 409
3007 ; 000015 RPDTRK= 13. ; TRACK 13
3008 ; 000010 RPDSC= 8. ; SECTOR 8
3009 ;
3010 ;
3011 ; REGISTER USAGE:
3012 ; R0 -- FUNCTION CODE (HIGH BYTE) UNIT # (LOW BYTE)
3013 ; BIT 15 SET IF WRITE
3014 ; R1 -- ADDRESS OF RPCS1
3015 ; R2 -- CYLINDER #
3016 ; R3 -- TRACK (HIGH BYTE) SECTOR (LOW BYTE)
3017 ; R4 -- WORD COUNT
3018 ; R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
3019 ; SP -- RETRY COUNTER
3020 ;
    
```

F06

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 69
DZBMD.P11 ROM CONTENTS TABLES

3021				
3022				
3023				
3024			:173410	
3025	005410	005002	:173410	005002
3026	005412	005003	:173412	005003
3027	005414	052700	:173414	052700
3028	005416	034400	:173416	034400

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

RPBOOT:

CLR
CLR
BIS

R2
R3

*<RPREAD+RPGO>*BIT8,RO ;SET READ HIGH BYTE, UNIT # LOW BYTE

G06

```

3029 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 6
3030 ;
3031 005420 012704 ;173420 012704 MOV # -256.,R4 ;READ 256 WORDS TO BOOT
3032 005422 177400 ;173422 177400
3033 ;
3034 ; BR RPSTRT ;START BOOT GOING
3035 ;
3036 ; START RPO4 GOING ON EITHER DUMP OR BOOT
3037 005424 012706 ;173424 012706 RPSTRT: MOV #RETRY,SP ;RETRY RETRY TIMES
3038 005426 000012 ;173426 000012
3039 005430 012701 ;173430 012701 MOV #RPEPA+RPCS1,R1 ;ADDRESS RPCS1 IN R1
3040 005432 176700 ;173432 176700
3041 ;
3042 ; BR RPRTRY ;FALL THROUGH RETRY CODE
3043 ;
3044 ; HERE ON ERROR TO RETRY
3045 005434 005705 ;173434 005705 RPRTRY: TST R5 ;INFINITE RETRY?
3046 005436 100402 ;173436 100402 BMI 10$ ;YES-- TRY AGAIN
3047 005440 005306 ;173440 005306 DEC SP ;RETRY COUNT EXHAUSTED?
3048 005442 002437 ;173442 002437 BLT RPEHLT ;YES-- GIVE UP
3049 ;
3050 005444 000005 ;173444 000005 10$: RESET ;ZAP!!
3051 005446 110061 ;173446 110061 MOV# RO,RPCS2(R1) ;SELECT PROPER UNIT #
3052 005450 000010 ;173450 000010
3053 005452 032711 ;173452 032711 BIT #RPDVA,(R1) ;IS DRIVE AVAILABLE TO US?
3054 005454 004000 ;173454 004000
3055 005456 001766 ;173456 001766 BEQ RPRTRY ;NO-- TRY AGAIN
3056 005460 012711 ;173460 012711 MOV #RPPRST+RPGO,(R1) ;DO 'READ-IN PRESET' FUNCTION
3057 005462 000021 ;173460 000021
3058 005464 010261 ;173464 010261 MOV R2,RPDC(R1) ;SELECT PROPER CYLINDER
3059 005466 000034 ;173466 000034
3060 005470 010361 ;173470 010361 MOV R3,RPDA(R1) ; AND TRACK AND SECTOR
3061 005472 000006 ;173472 000006
3062 005474 010461 ;173474 010461 MOV R4,RPWC(R1) ;SET UP WORD COUNT TO PROPER VALUE
3063 005476 000002 ;173476 000002
3064 ;
3065 ; ;NOTE THAT IT IS NOT NECCESARY TO SET UP BUS
3066 005500 000300 ;173500 000300 SWAB RO ; ADDRESS, SINCE IT IS 0 AFTER READ-IN PRESET
3067 005502 110011 ;173502 110011 MOV# RO,(R1) ;GET FUNCTION CODE IN LOW BYTE
3068 005504 000300 ;173504 000300 SWAB RO ;START FUNCTION GOING
3069 ; ;RESTORE RO
3070 005506 105711 ;173506 105711 20$: TSTB (R1) ;READY?
3071 005510 100376 ;173510 100376 BPL 20$ ;NO-- WAIT UNTIL IT IS
3072 005512 032711 ;173512 032711 BIT #RPTRE!RPMCPE,(R1) ;TRANSFER OR MBC PARITY ERROR?
3073 005514 060000 ;173514 060000
3074 005516 001346 ;173516 001346 BNE RPRTRY ;YES-- ERROR-- TRY AGAIN
3075 005520 032761 ;173520 032761 BIT #RPATA!RPERR,RPDS(R1) ;ATTN OR OTHER ERROR?
3076 005522 140000 ;173522 140000
3077 005524 000012 ;173524 000012
3078 005526 001342 ;173526 001342 BNE RPRTRY ;YES-- ERROR-- TRY AGAIN
3079 005530 005700 ;173530 005700 TST RO ;READ FUNCTION?
3080 005532 100247 ;173532 100247 BPL CLRPC ;YES-- BOOT-- GO TO LOCATION 0
3081 ; ;NO-- DUMP-- HALT WITH RO= 0 IN DISPLAY

```


H06

```

3082
3083
3084
3085 005534 005000 ;173534 005000 HALTO: CLR RO ;DISPLAY RO= 0 IF NO ERRORS
3086
3087 005536 000000 ;173536 000000 HALTED: HALT ;DIE
3088 005540 000776 ;173540 000776 BR HALTED ;STAY DEAD
3089
3090
3091
3092 005542 016100 ;173542 016100 RPEHLT: MOV RPDS(R1),RO ;DISPLAY DRIVE STATUS
3093 005544 000012 ;173544 000012
3094 005546 000773 ;173546 000773 BR HALTED ;R.I.P.
3095
3096
3097
3098
3099
3100
3101
3102
3103 005550 012702 ;173550 012702
3104 005552 000631 ;173552 000631
3105 005554 012703 ;173554 012703
3106 005556 006410 ;173556 006410
3107 005560 012700 ;173560 012700
3108 005562 130400 ;173562 130400
3109 005564 012704 ;173564 012704
3110 005566 110000 ;173566 110000
3111 005570 005005 ;173570 005005
3112 005572 000714 ;173572 000714

```

```

: HERE TO HALT AFTER A DUMP-- DISPLAY RO= 0 IF NO ERRORS
:
: HERE ON ERROR FROM RPO4 AFTER RETRYING-- DISPLAY DRIVE STATUS IN RO
:
: START -11 HERE TO DUMP TO RPO4 DISK
:
: NOTE THAT RO-R7 HAVE ALREADY BEEN SAVED IN 40-56
: BY PRESSING BUTTON #1.
RPDUMP:
MOV #RPDCYL,R2
MOV #<RPDTRK*BIT8>!<RPDSCT*BIT0>,R3
MOV #BIT15!<<RPWRIT+RPG0>*BIT8>,RO ;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

```

```

3113 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 7
3114 ;
3115 ; INTERNAL BUTTON #4 -- DUMP AND BOOTSTRAP THROUGH DTE20
3116 ;
3117 ; DTE20 DEFINITIONS
3118 ;
3119 ; NOTE THAT ALL DTE20 REGISTER DEFINITIONS AND BIT DEFINITIONS
3120 ; ARE IN $DEF IN SYSMAC.SML
3121 ;
3122 ; 000040 DTESIZ= 40 ; EACH DTE OCCUPIES 20 WORDS IN EXTERNAL PAGE
3123 ; 000004 DTEMAX= 4 ; MAX OF 4 DTE'S ON A PDP-11
3124 ;
3125 ;
3126 ; BUTTON #4 -- INITIATED BY '-10 RELOAD -11' BIT
3127 ;
3128 005574 010037 ;173574 010037 BUTON4: MOV RO,ROTOR7+0 ;SAVE RO IN 40
3129 005576 000040 ;173576 000040
3130 005600 012700 ;173600 012700 MOV #10$,RO ;SET RETURN ADDRESS IN RO
3131 005602 173606 ;173602 173606
3132 005604 000664 ;173604 000664 BR REGSAV ;SAVE R1-R7
3133 ;
3134 ; REGISTERS SAVED-- LOOK FOR THE DTE20 WHICH PUSHED THE BUTTON
3135 ;
3136 ; THE DTE WHICH PUSHED THE BUTTON SHOULD HAVE THE DOORBELL
3137 ; RINGING AND HAVE THE VALUE 1365 (OCTAL) IN IT'S
3138 ; TO -10 BYTE COUNT TO10BC.
3139 ;
3140 ; NXM (TIME-OUT) TRAP IS USED TO SKIP NON-EXISTANT DTE20'S.
3141 ;
3142 005606 005005 ;173606 005005 10$: CLR R5 ;ADDRESS LOCATION ZERO
3143 005610 012500 ;173610 012500 MOV (R5)+,RO ;SAVE 0 IN RO
3144 005612 012501 ;173612 012501 MOV (R5)+,R1 ;SAVE 2 IN R1
3145 005614 011502 ;173614 011502 MOV (R5),R2 ;SAVE 4 IN R2
3146 005616 012725 ;173616 012725 MOV #21$, (R5)+ ;SET NXM TRAP ADDRESS IN 4
3147 005620 173634 ;173620 173634
3148 005622 011503 ;173622 011503 MOV (R5),R3 ;SAVE 6 IN R3
3149 005624 012715 ;173624 012715 MOV #PR7,(R5) ;SET PRIORITY FOR NXM TRAP
3150 005626 000340 ;173626 000340
3151 ;
3152 ; LOOP THROUGH ALL DTE'S
3153 ;
3154 005630 012704 ;173630 012704 20$: MOV #DLYCNT-DTESIZ,R4 ;POINT TO DTE # -1'S DELAY COUNT REGISTER
3155 005632 174340 ;173632 174340
3156 ; (WILL BUMP TO # 0)
  
```

J06

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 73
DZBMD.P11 ROM CONTENTS TABLES

```
3157  
3158  
3159  
3160 005634 012706 :173634 012706  
3161 005636 000004 :173636 000004  
3162  
3163 005640 062704 :173640 062704  
3164 005642 000040 :173642 000040  
3165 005644 105704 :173644 105704  
3166  
3167  
3168 005646 100770 :173646 100770  
3169  
3170 005650 032764 :173650 032764  
3171 005652 004000 :173652 004000  
3172 005654 000034 :173654 000034  
3173 005656 001770 :173656 001770  
3174 005660 026417 :173660 026417  
3175 005662 000014 :173662 000014  
3176  
3177 005664 001365 :173664 001365  
3178
```

;
; 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 #DTESIZ,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 ;YES-- START ALL OVER, UNTIL A DTE
; SAYS HE PUSHED THE BUTTON
BIT #T011DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?
BEQ 22\$;NO-- TRY NEXT DTE
CMP T010BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365
; IN IT'S TO -10 BYTE COUNT?
BNE 22\$;NO-- TRY ANOTHER DTE
;

K06

```

3179      ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23)  MACY11 27(657) 22-AUG-75 10:30 PAGE 7
3180      ;
3181      ; WE HAVE FOUND THE DTE WHICH PUSHED THE BUTTON
3182      ;
3183      ; ADDRESS OF DLYCNT REGISTER IS IN R4
3184      ;
3185 005666 010315 ;173666 010315      MOV      R3,(R5)      ;RESTORE LOCATION 6
3186 005670 010245 ;173670 010245      MOV      R2,-(R5)      ; 4
3187 005672 010145 ;173672 010145      MOV      R1,-(R5)      ; 2
3188 005674 010045 ;173674 010045      MOV      R0,-(R5)      ; 0
3189      ;
3190      ; SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
3191      ; IN LOCATIONS 130-156
3192      ;
3193 005676 012700 ;173676 012700      MOV      #DTESAV,R0      ;POINT TO SAVE AREA
3194 005700 000130 ;173700 000130
3195 005702 012420 ;173702 012420 29$: MOV      (R4)+,(R0)+      ;SAVE A REGISTER
3196 005704 022700 ;173704 022700      CMP      #T011DT-DLYCNT+DTESAV,R0 ;FINISHED?
3197 005706 000156 ;173706 000156
3198 005710 103374 ;173710 103374      BHS      29$      ;NO-- SAVE SOME MORE
3199      ;
3200      ; R4= T011DT+2
3201      ;
3202      ; SET R1= STATUS REGISTER
3203      ; R4= DIAG2 REGISTER
3204      ;
3205      ; DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
3206      ; LOADED FLAG
3207      ;
3208      ; $$$=
3209 005712 005724 ;173712 005724      TST      (R4)+
3210 005714 010401 ;173714 010401      MOV      R4,R1      ; SO DOES R1
3211 005716 012700 ;173716 012700      MOV      #DRESET,R0      ;SETUP R0 FOR 'DIAGNOSTIC RESET'
3212 005720 000100 ;173720 000100
3213 005722 010021 ;173722 010021      MOV      R0,(R1)+      ;R1 POINTS TO STATUS REGISTER

```

```

3214
3215
3216
3217
3218
3219
3220
3221
3222
3223 005724 005061 :173724 005061
3224 005726 177744 :173726 177744
3225 005730 005061 :173730 005061
3226 005732 177764 :173732 177764
3227
3228 005734 032711 :173734 032711
3229 005736 004000 :173736 004000
3230 005740 001775 :173740 001775
3231 005742 010014 :173742 010014
3232
3233
3234
3235
3236
3237
3238 005744 005061 :173744 005061
3239 005746 177766 :173746 177766
3240 005750 012761 :173750 012761
3241 005752 107400 :173752 107400
3242 005754 177762 :173754 177762
3243
3244 005756 032711 :173756 032711
3245 005760 004000 :173760 004000
3246 005762 001775 :173762 001775
3247 005764 010014 :173764 010014
3248 005766 012705 :173766 012705
3249 005770 100000 :173770 100000
3250
3251 005772 005007 :173772 005007
3252

```

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
30S: BIT #T011DB,(R1) ;IS DOORBELL RINGING (TRANSFER COMPLETE)?

```

```

BEQ 30S ;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

```

```

40S: BIT #T011DB,(R1) ; -10 WHEN DONE ;DOORBELL RINGING (LOAD FINISHED)?

```

```

BEQ 40S ;NO-- WAIT UNTIL DONE
MOV R0,(R4) ;CLEAR DOORBELL RINGING
MOV #BIT15,R5 ;SET R5: BIT15= 1, BIT0= 0

```

```

CLR PC ; TO SAY BUTTON #4 PRESSED ;GO TO LOADED CODE, STARTING AT ; LOCATION 0

```

M06

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 76
DZBMD.P11 ROM CONTENTS TABLES

```
3253 ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(657) 22-AUG-75 10:30 PAGE 8
3254 ;
3255 ;
3256 ;
3257 ;
3258 ;173774 000004 .PRINT <1000>-<.-ROMORG> ;FREE BYTES AT 1000
3259 005774 000000 ;173774 000 .BYTE 0
3260 ;173775 000 .BYTE 0
3261 005776 END.YF: ;173776 000 .BYTE 0
3262 005776 000000 ;173777 000 .BYTE 0
3263 ;
3264 ;
3265 ;
3266 ;174000 000001 PASS2: .END
3267 ;
3268 ;
3269 ;
```

```

3270 006000 MAP.YG:
3271 :THE FOLLOWING IS A REPRODUCTION
3272 :OF THE ROM PROGRAM FOR BM873YG.
3273 :IT IS HERE FOR COMPARISON TO THE
3274 :ACTUAL ROM AND FOR REFERENCE
3275 ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76 09:14 PAGE 1
3276
3277 .TITLE BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM
3278 ;
3279 ; THIS IS THE CODE TO BE ENCODED IN THE BOOTSTRAP ROM ON THE BM873-YG BOARD
3280
3281
3282 MODULE: BM873G
3283
3284 DATE: JUNE 1976
3285
3286 AUTHOR: RICH MURATORI
3287
3288
3289 COPYRIGHT (C) 1976 DIGITAL EQUIPMENT CORPORATION
3290 ALL RIGHTS RESERVED
3291
3292
3293 .ENABLE ABS,AMA
3294 ;
3295
3296 ASCII CHARACTER DEFINITIONS
3297
3298 000040 SPACE= 40 ;ASCII SPACE
3299 000001 SYN= 1 ;ASCII SYNC
3300 000012 LF= 12 ;ASCII LINE FEED
3301 000015 CR= 15 ;ASCII CARRIAGE RETURN
3302 000054 COMMA= 54 ;ASCII COMMA
3303 000006 ACK= 6 ;ASCII ACKNOWLEDGE
3304 000025 NAK= 25 ;ASCII NEG ACKNOWLEDGE
3305
3306
3307 BUFFER AREAS
3308
3309 002100 LINBUF= 2100 ;LINE INPUT BUFFER
3310 002310 DEABUF= 2310 ;DEASCIIIZED INPUT BUFFER
3311
3312 DL11E REGISTER DEFINITIONS
3313
3314 176000 DLRCR= 176000 ;DL11E RECEIVER STATUS REGISTER
3315 176002 DLRBUF= 176002 ;DL11E RECEIVER BUFFER
3316 176004 DLXCSR= 176004 ;DL11E TRANSMITTER STATUS REGISTER
3317 176006 DLXBUF= 176006 ;DL11E TRANSMISSION BUFFER
3318
3319 100000 BIT15=100000
3320 000340 PR7=7*40 ;PRIORITY LEVEL 7
3321
3322 DTE20 REGISTER DEFINITIONS
3323
3324 174400 DLYCNT=174400 ;DELAY COUNT WORD
3325 174414 T010BC=174414 ;T0-10 PDP-11 MEMORY ADDRESS
  
```

3324	:	174416	T011BC=174416	:T0-11	BYTE COUNT
3325	:	174420	T010AD=174420	:T0-10	POP-11 MEMORY ADDRESS
3326	:	174422	T011AD=174422	:T0-11	POP-11 MEMORY ADDRESS
3327	:	174426	T011DT=174426	:T0-11	POP-11 DATA WORD
3328	:	174434	STAT=174434	:STATUS	WORD
3329	:				
3330	:				

T011BC REGISTER BIT DEFINITIONS

C07

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 79
DZBMD.P11 ROM CONTENTS TABLES

```
3331 ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76 09:14 PAGE 1-1
3332
3333 ; 100000 IFLOP=100000 ;I FLIPFLOP BIT
3334
3335
3336
3337
3338 ; 000100 DRESET=000100 ;DTE RESET
3339
3340
3341
3342 ; 004000 T011DB=004000 ;-10 REQUESTED -11 INTERRUPT
3343
3344
3345
3346
3347 ; 000040 ROTOR7= 40 ;SAVE R0 TO R7 IN 40 TO 56
3348
3349 ; 000130 DTESAV= 130 ;SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
3350 ; ; IN LOCATIONS 130-156
3351
3352 ; 173000 ROMORG= 173000 ;ROM STARTS AT 773000
3353
3354 ; ESTABLISH ROM ORIGIN
3355
3356 ; 173000 .=ROMORG
```

3357 ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76 09:14 PAGE 2

3358 :
3359 : .SBTTL BOOT FROM MASTER FRONT END
3360 :

3361 : EXTERNAL BUTTON #1 -- LOAD FROM MASTER FRONT END VIA DL11E
3362 :

3363 : DEPRESSING BUTTON #1 CAUSES A LOADER UTILITY PROGRAM (DGQDE) TO BE LOADED
3364 :

3365 : TO ACCOMPLISH THIS, THE FOLLOWING CONDITIONS MUST EXIST:
3366 :

3367 : 1) THE LOADER UTILITY PROGRAM DGQDD MUST BE RUNNING UNDER KLDCP
3368 : IN THE MASTER FRONT END.

3369 : 2) THE '.A11' FILE FOR DGQDE MUST BE AVAILABLE ON THE SELECTED LOAD
3370 : MEDIUM IN THE MASTER FRONT END.

3371 : 3) THE MASTER FRONT END AND THE SECONDARY FRONT END MUST BE CONNECTED
3372 : THROUGH DL11E'S AND A NULL MODEM.

3373 : 4) THE SECONDARY FRONT END MUST HAVE A TTY CONNECTED TO IT.
3374 :

3375 : ; SEND BOOTSTRAP REQUEST TO MASTER FRONT END VIA THE DL11E. THE
3376 : REQUEST IS IN THE FORM 'B <CR><LF>', WHICH CALLS FOR THE
3377 : BOOTING OVER OF DGQDE.A11.

3378 006000 000005 ; 173000 000005 BUTON1: RESET ; CLEAR THE WORLD
3379 006002 012706 ; 173002 012706 MOV #2000,SP ; SETUP STACK POINTER

3380 006004 002000 ; 173004 002000 MOV #DGQDE,R1 ; ADDRESS OF SYNC + BOOT REQUEST
3381 006006 012701 ; 173006 012701

3382 006010 173374 ; 173010 173374 SENDIT: TSTB @DLXCSR ; IS DL11E READY TO TRANSMIT
3383 006012 105737 ; 173012 105737

3384 006014 176004 ; 173014 176004 BPL SENDIT ; LOOP UNTIL IT IS
3385 006016 100375 ; 173016 100375 BR IS ; BRANCH AROUND POWER -FAIL VECTOR

3386 006020 000403 ; 173020 000403 ; *****
3387 : REQUIRED POWER-FAIL VECTOR - MUST BE AT 173024
3388 :

3389 006022 000000 ; 173022 000000 .WORD 0 ; FILLER
3390 006024 173000 ; 173024 173000

3391 006026 000340 ; 173026 000340 .WORD ROMORG,PR7
3392 :
3393 :
3394 : *****

3395 006030 112137 ; 173030 112137 1S: MOV (R1)+,@DLXBUF ; LOAD A CHAR INTO OUTPUT BUFFER
3396 006032 176006 ; 173032 176006

3397 006034 105711 ; 173034 105711 TSTB (R1) ; ANY MORE CHARS TO SEND?
3398 006036 001365 ; 173036 001365 BNE SENDIT ; BRANCH IF MORE CHARS TO SEND

3399 006040 005005 ; 173040 005005 CLR RS ; CLEAR SYNC RECEIVED FLAG
3400 : ; WAIT TO RECEIVE BOOT PROGRAM (DGQDE.A11), ONE ASCIIZED CHAR AT
3401 : A TIME, ONE LINE AT A TIME.

3402 006042 012701 ; 173042 012701 NXTLIN: MOV @LINBUF,R1 ; ADDRESS OF LINE INPUT BUFFER
3403 006044 002100 ; 173044 002100

3404 006046 105737 ; 173046 105737 NXTCHR: TSTB @DLRCSR ; CHAR RECEIVED YET?
3405 006050 176000 ; 173050 176000

3406 006052 100375 ; 173052 100375 BPL NXTCHR ; BRANCH IF STILL TO WAIT
3407 :
3408 : ; PROCESS THE RECEIVED ASCIIZED CHAR. IGNORE ALL CHARS UNTIL A SYNC
3409 : SIGNAL IS RECEIVED. A LINE FEED MARKS THE END OF A LINE. THE MAX
3410 : NUMBER OF CHARS PER LINE IS 131, MORE THAN THAT IS AN ERROR.

E07

3411	006054	113711	::173054	113711	2S:	MOVB	3#DLRBUF, (R1)	;READ CHAR INTO LINE INPUT BUFFER
3412	006056	176002	::173056	176002		BICB	#200, (R1)	;CLEAR HIGH ORDER BIT OF CHAR
3413	006060	142711	::173060	142711		BEQ	NXTCHR	;BRANCH IF YES, IGNORE NULLS
3414	006062	000200	::173062	000200		CMPB	(R1), #SYN	;IS CHAR THE SYNC SIGNAL
3415	006064	001770	::173064	001770		BEQ	3\$;BRANCH IF YES
3416	006066	121127	::173066	121127		TST	R5	;HAS SYNC ALREADY BEEN RECEIVED?
3417	006070	000001	::173070	000001		BEQ	NXTCHR	;BRANCH IF NOT, IGNORE CHAR
3418	006072	001413	::173072	001413		CMPB	(R1)+, #LF	;IS CHAR A LINE FEED?
3419	006074	005705	::173074	005705		BEQ	PACKIT	;BRANCH IF YES, END OF LINE
3420	006076	001763	::173076	001763				
3421	006100	122127	::173100	122127				
3422	006102	000012	::173102	000012				
3423	006104	001410	::173104	001410				

```

3424 ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76 09:14 PAGE 2-1
3425
3426 006106 020127 ;173106 020127 CMP R1,#LINBUF+132. ;HAVE WE REACHED THE END OF THE BUFFER?
3427 006110 002304 ;173110 002304
3428 006112 003755 ;173112 003755 BLE NXTCHR ;BRANCH IF NOT, GET REST OF LINE
3429 006114 004737 ;173114 004737 JSR PC,NAKSND ;SEND A NEG ACKNOWLEDGE
3430 006116 173346 ;173116 173346
3431 006120 000000 ;173120 000000 HALT ;TOO MANY CHARS IN A LINE
3432
3433 006122 005205 ;173122 005205 3$: INC R5 ;SET SYNC RECEIVED FLAG
3434 006124 000750 ;173124 000750 BR NXTCHR ;GO GET NEXT CHAR
3435 ;
3436 ;UNSCRAMBLE THE ASCIIZED CHARS INTO 16-BIT WORDS. THE FORMAT OF A
3437 ;LINE IS E WRDCNT,LADDR,DATA,DATA,...,DATA,CHKSUM<CR><LF>
3438 ;WHERE WRDCNT IS THE WORD COUNT
3439 ; LADDR IS THE LOAD ADDRESS
3440 ; DATA IS LOAD DATA
3441 ; CHKSUM IS THE CHECKSUM
3442 ;
3443 006126 012703 ;173126 012703 PACKIT: MOV #DEABUF,R3 ;GET ADDRESS OF DE-ASCIIZED BUFFER
3444 006130 002310 ;173130 002310
3445 006132 012701 ;173132 012701 MOV #LINBUF,R1 ;GET ADDRESS OF INPUT BUFFER
3446 006134 002100 ;173134 002100
3447 006136 122127 ;173136 122127 CMPB (R1)+,#'E ;FIRST CHAR IN LINE SHOULD BE AN 'E'
3448 006140 000105 ;173140 000105
3449 006142 001403 ;173142 001403 BEQ 1$ ;BRANCH IF IT IS
3450 006144 004737 ;173144 004737 JSR PC,NAKSND ;SEND A NEG ACKNOWLEDGE
3451 006146 173346 ;173146 173346
3452 006150 000000 ;173150 000000 HALT ;INCORRECT LINE SYNTAX, FIRST CHAR NOT AN E
3453 ;
3454 006152 122127 ;173152 122127 1$: CMPB (R1)+,#SPACE ;SECOND CHAR SHOULD BE A SPACE
3455 006154 000040 ;173154 000040
3456 006156 001403 ;173156 001403 BEQ NXTWRD ;BRANCH IF IT IS
3457 006160 004737 ;173160 004737 JSR PC,NAKSND ;SEND A NEG ACKNOWLEDGE
3458 006162 173346 ;173162 173346
3459 006164 000000 ;173164 000000 HALT ;INCORRECT LINE SYNTAX, 2ND CHAR NOT A SPACE
3460 ;
3461 006166 005002 ;173166 005002 NXTWRD: CLR R2 ;CLEAR WORD FORMER
3462 006170 112100 ;173170 112100 1$: MOVB (R1)+,R0 ;READ CHAR FROM LINE BUFFER
3463 006172 122700 ;173172 122700 CMPB #CR,R0 ;IS CHAR A CARRIAGE RETURN
3464 006174 000015 ;173174 000015
3465 006176 001774 ;173176 001774 BEQ 1$ ;BRANCH IF YES
3466 006200 122700 ;173200 122700 CMPB #LF,R0 ;IS CHAR A LINE FEED
3467 006202 000012 ;173202 000012
3468 006204 001422 ;173204 001422 BEQ 3$ ;BRANCH IF IT IS
3469 006206 122700 ;173206 122700 CMPB #COMMA,R0 ;IS CHAR A COMMA
3470 006210 000054 ;173210 000054
3471 006212 001415 ;173212 001415 BEQ 2$ ;BRANCH IF IT IS
3472 006214 006302 ;173214 006302 ASL R2 ;SHIFT WORD OVER TO MAKE ROOM FOR
3473 006216 006302 ;173216 006302 ASL R2 ;NEXT CHAR
3474 006220 006302 ;173220 006302 ASL R2
3475 006222 000402 ;173222 000402 BR 4$ ;BRANCH AROUND POWER-FAIL VECTOR
3476 ;*****
3477 ; REQUIRED POWER-FAIL VECTOR - MUST BE AT 173224
    
```

G07

3478		.							
3479	006224	173000	;	173224	173000		.WORD	ROMORG,PR7	
3480	006226	000340	;	173226	000340				
3481		.							
3482		.					;*****		
3483		.							
3484	006230	006302	;	173230	006302	4\$:	ASL	R2	
3485	006232	006302	;	173232	006302		ASL	R2	
3486	006234	006302	;	173234	006302		ASL	R2	
3487	006236	042700	;	173236	042700		BIC	#100,R0	;CLEAR ASCIIIZED BIT
3488	006240	000100	;	173240	000100				
3489	006242	050002	;	173242	050002		BIS	R0,R2	;INSERT NEW CHAR INTO WORD
3490	006244	000751	;	173244	000751		BR	1\$;GO GET NEXT CHAR
3491		.							
3492	006246	010223	;	173246	010223	2\$:	MOV	R2,(R3)+	;STORE WORD IN BUFFER

H07

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 84
 DZBMD.P11 ROM CONTENTS TABLES

```

3493 ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76 09:14 PAGE 2-2
3494
3495 006250 000746 ;173250 000746 BR NXTWRD ;GO FORM NEXT WORD
3496
3497 006252 010223 ;173252 010223 3$: MOV R2,(R3)+ ;STORE CHECKSUM IN BUFFER
3498
3499 ;
3500 006254 012702 ;173254 012702 CHCKIT: MOV #DEABUF,R2 ;ADDRESS OF BUFFER
3501 006256 002310 ;173256 002310
3502 006260 005000 ;173260 005000
3503 006262 062200 ;173262 062200 1$: CLR R0 ;CLEAR CHECKSUM
3504 006264 020203 ;173264 020203 ADD (R2)+,R0 ;ADD NEXT WORD TO CHECKSUM
3505 006266 002775 ;173266 002775 CMP R2,R3 ;REACHED END OF BUFFER YET
3506 006270 005700 ;173270 005700 BLT 1$ ;BRANCH IF NOT
3507 006272 001403 ;173272 001403 TST R0 ;IS CHECKSUM = 0?
3508 006274 004737 ;173274 004737 BEQ LOADIT ;BRANCH IF YES
3509 006276 173346 ;173276 173346 JSR PC,NAKSND ;SEND A NEG ACKNOWLEDGE
3510 006300 000000 ;173300 000000 HALT ;CHECKSUM ERROR
3511
3512 ;
3513 ;LOAD THE RECEIVED DATA WORDS INTO THE DESIGNATED PLACE IN MEMORY.
3514 ;A WORD COUNT OF ZERO INDICATES A TRANSFER BLOCK. START EXECUTING
3515 006302 013700 ;173302 013700 ;THE LOADED PROGRAM AT THE SPECIFIED ADDRESS.
3516 006304 002310 ;173304 002310 LOADIT: MOV DEABUF,R0 ;GET LOAD WORD COUNT
3517 006306 001413 ;173306 001413 BEQ 2$ ;BRANCH IF IT'S ZERO (A TRANSFER BLOCK)
3518 006310 012702 ;173310 012702 MOV #DEABUF+4,R2 ;ADDRESS OF FIRST DATA WORD
3519 006312 002314 ;173312 002314
3520 006314 013701 ;173314 013701 MOV DEABUF+2,R1 ;GET LOAD ADDRESS
3521 006316 002312 ;173316 002312
3522 006320 112221 ;173320 112221 1$: MOVB (R2)+,(R1)+ ;MOVE DATA FROM BUFFER TO MEMORY
3523 006322 112221 ;173322 112221 MOVB (R2)+,(R1)+ ;MOVE DATA FROM BUFFER TO MEMORY
3524 006324 005300 ;173324 005300 DEC R0 ;DECREMENT WORD COUNT
3525 006326 003374 ;173326 003374 BGT 1$ ;BRANCH UNTIL ALL DATA IS LOADED
3526 006330 004737 ;173330 004737 JSR PC,ACKSND ;GO SEND AN ACK
3527 006332 173354 ;173332 173354
3528 006334 000642 ;173334 000642 BR NXTLIN ;GO GET NEXT LINE
3529
3530 006336 004737 ;173336 004737 2$: JSR PC,ACKSND ;GO SEND AN ACK
3531 006340 173354 ;173340 173354
3532 006342 013707 ;173342 013707 MOV DEABUF+2,PC ;START ADDRESS OF LOADED PROGRAM
3533 006344 002312 ;173344 002312
3534
3535 ;
3536 ;NAKSND IS USED TO SEND A NEG ACK BACK TO THE MASTER FRONT END.
3537 ;ACKSND IS USED TO SEND AN ACK.
3538 006346 012700 ;173346 012700 NAKSND: MOV #NAK,R0 ;SETUP ASCII NEG ACK
3539 006350 000025 ;173350 000025
3540 006352 000402 ;173352 000402 BR RESPND ;GO SEND IT
3541 006354 112700 ;173354 112700 ACKSND: MOVB #ACK,R0 ;SETUP ASCII ACK
3542 006356 000006 ;173356 000006
3543 006360 105737 ;173360 105737 RESPND: TSTB @DLXCSR ;IS TRANSMITTER READY?
3544 006362 176004 ;173362 176004
3545 006364 100375 ;173364 100375 BPL RESPND ;WAIT TIL IT IS
3546 006366 110037 ;173366 110037 MOVB R0,@DLXBUF ;SEND ACK/NAK

```

3547	006370	176006	;173370	176006				
3548	006372	000207	;173372	000207		RTS	PC	;RETURN TO CALLING ROUTINE
3549		.						
3550	006374	041001	;173374	001	DGQDE:	.BYTE	SYN	
3551		.	;173375	102		.ASCIZ	/B/<CR><LF>	
3552	006376	005015	;173376	005015				
3553	006400	000000	;173400	000				
3554		.	;173401	000		.BYTE	0	
3555	006402	000000	;173402	000		.BYTE	00	
3556		.	;173403	000		.BYTE	00	
3557	006404	000000	;173404	000		.BYTE	00	
3558		.	;173405	000		.BYTE	00	
3559	006406	000000	;173406	000		.BYTE	00	
3560		.	;173407	000		.BYTE	00	
3561	006410	000000	;173410	000		.BYTE	0	

J07

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 86
DZBMD.P11 ROM CONTENTS TABLES

3562 ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76 09:14 PAGE 2-3

3563			:173411	000	.BYTE	0
3564			:173412	000	.BYTE	0
3565	006412	000000	:173413	000	.BYTE	0
3566			:173414	000	.BYTE	0
3567	006414	000000	:173415	000	.BYTE	0
3568			:173416	000	.BYTE	0
3569	006416	000000	:173417	000	.BYTE	0
3570			:173420	000	.BYTE	0
3571	006420	000000	:173421	000	.BYTE	0
3572			:173422	000	.BYTE	0
3573	006422	000000	:173423	000	.BYTE	0
3574			:173424	000	.BYTE	0
3575	006424	000000	:173425	000	.BYTE	0
3576			:173426	000	.BYTE	0
3577	006426	000000	:173427	000	.BYTE	0
3578			:173430	000	.BYTE	0
3579	006430	000000	:173431	000	.BYTE	0
3580			:173432	000	.BYTE	0
3581	006432	000000	:173433	000	.BYTE	0
3582			:173434	000	.BYTE	0
3583	006434	000000	:173435	000	.BYTE	0
3584			:173436	000	.BYTE	0
3585	006436	000000	:173437	000	.BYTE	0
3586			:173440	000	.BYTE	0
3587	006440	000000	:173441	000	.BYTE	0
3588			:173442	000	.BYTE	0
3589	006442	000000	:173443	000	.BYTE	0
3590			:173444	000	.BYTE	0
3591	006444	000000	:173445	000	.BYTE	0
3592			:173446	000	.BYTE	0
3593	006446	000000	:173447	000	.BYTE	0
3594			:173450	000	.BYTE	0
3595	006450	000000	:173451	000	.BYTE	0
3596			:173452	000	.BYTE	0
3597	006452	000000	:173453	000	.BYTE	0
3598			:173454	000	.BYTE	0
3599	006454	000000	:173455	000	.BYTE	0
3600			:173456	000	.BYTE	0
3601	006456	000000	:173457	000	.BYTE	0
3602			:173460	000	.BYTE	0
3603	006460	000000	:173461	000	.BYTE	0
3604			:173462	000	.BYTE	0
3605	006462	000000	:173463	000	.BYTE	0
3606			:173464	000	.BYTE	0
3607	006464	000000	:173465	000	.BYTE	0
3608			:173466	000	.BYTE	0
3609	006466	000000	:173467	000	.BYTE	0
3610			:173470	000	.BYTE	0
3611	006470	000000	:173471	000	.BYTE	0
3612			:173472	000	.BYTE	0
3613	006472	000000	:173473	000	.BYTE	0
3614			:173474	000	.BYTE	0
3615	006474	000000				

K07

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 87
DZBMD.P11 ROM CONTENTS TABLES

3616			:173475	000	.BYTE	0
3617	006476	000000	:173476	000	.BYTE	0

L07

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 88
DZBMD.P11 ROM CONTENTS TABLES

3618 ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76 09:14 PAGE 2-4
3619

3620			:173477	000	.BYTE	0
3621	006500	000000	:173500	000	.BYTE	0
3622			:173501	000	.BYTE	0
3623	006502	000000	:173502	000	.BYTE	0
3624			:173503	000	.BYTE	0
3625	006504	000000	:173504	000	.BYTE	0
3626			:173505	000	.BYTE	0
3627	006506	000000	:173506	000	.BYTE	0
3628			:173507	000	.BYTE	0
3629	006510	000000	:173510	000	.BYTE	0
3630			:173511	000	.BYTE	0
3631	006512	000000	:173512	000	.BYTE	0
3632			:173513	000	.BYTE	0
3633	006514	000000	:173514	000	.BYTE	0
3634			:173515	000	.BYTE	0
3635	006516	000000	:173516	000	.BYTE	0
3636			:173517	000	.BYTE	0
3637	006520	000000	:173520	000	.BYTE	0
3638			:173521	000	.BYTE	0
3639	006522	000000	:173522	000	.BYTE	0
3640			:173523	000	.BYTE	0
3641	006524	000000	:173524	000	.BYTE	0
3642			:173525	000	.BYTE	0
3643	006526	000000	:173526	000	.BYTE	0
3644			:173527	000	.BYTE	0
3645	006530	000000	:173530	000	.BYTE	0
3646			:173531	000	.BYTE	0
3647	006532	000000	:173532	000	.BYTE	0
3648			:173533	000	.BYTE	0
3649	006534	000000	:173534	000	.BYTE	0
3650			:173535	000	.BYTE	0
3651	006536	000000	:173536	000	.BYTE	0
3652			:173537	000	.BYTE	0
3653					.EVEN	

M07

```

3654 ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76 09:14 PAGE 3
3655 ;
3656 ;
3657 ;
3658 ;
3659 ;
3660 ;
3661 ;
3662 ;
3663 ;
3664 ;
3665 ;
3666 ;
3667 ;
3668 ;
3669 ;
3670 ;
3671 ;
3672 ;
3673 006540 010037 ;173540 010037 REGSAV: MOV RO, ROTOR7+16 ;SAVE RO AS PC IN 56
3674 006542 000056 ;173542 000056
3675 ;
3676 006544 012700 ;173544 012700 MOV #ROTOR7+16,RO ;RO NOW POINTS TO 56
3677 006546 000056 ;173546 000056
3678 ;
3679 006550 010640 ;173550 010640 MOV SP, -(RO) ;SAVE SP IN 54
3680 006552 010540 ;173552 010540 MOV R5, -(RO) ;SAVE R5 IN 52
3681 006554 010440 ;173554 010440 MOV R4, -(RO) ;SAVE R4 IN 50
3682 006556 010340 ;173556 010340 MOV R3, -(RO) ;SAVE R3 IN 46
3683 006560 010240 ;173560 010240 MOV R2, -(RO) ;SAVE R2 IN 44
3684 006562 010140 ;173562 010140 MOV R1, -(RO) ;SAVE R1 IN 42
3685 006564 014000 ;173564 014000 MOV -(RO), RO ;RESTORE RO FROM 40
3686 006566 000177 ;173566 000177 JMP @ROTOR7+16 ;GO TO SAVED PC
3687 006570 004264 ;173570 004264
3688 ;
3689 006572 000000 ;173572 000000 .WORD 0 ;FILLER WORD
  
```

SBTTL REGISTER SAVE ROUTINE
 REGISTER SAVE ROUTINE

REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS RO-R7
 IN MEMORY AT 40-56 (LOCATION ROTOR7).

CALLING SEQUENCE:
 MOV RO, ROTOR7+0
 MOV #RET RO
 BR REGSAV
 RET: <RETURN HERE>

ALL REGISTERS RESTORED

.=ROMORG+540

NO7

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 90
 DZBMD.P11 ROM CONTENTS TABLES

```

3690 ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76 09:14 PAGE 4
3691 ;
3692 ; SBTTL DUMP AND BOOTSTRAP THROUGH DTE20
3693 ; INTERNAL BUTTON #4 -- DUMP AND BOOTSTRAP THROUGH DTE20
3694 ;
3695 ; 000040 DTESIZ= 40 ;EACH DTE OCCUPIES 20 WORDS IN EXTERNAL PAGE
3696 ;
3697 ; BUTTON #4 -- INITIATED BY '-10 RELOAD -11' BIT
3698 ;
3699 ; 173574 . =ROMORG+574
3700 ;
3701 006574 010037 ;173574 010037 BUTON4: MOV RO,ROTOR7+0 ;SAVE RO IN 40
3702 006576 000040 ;173576 000040
3703 006600 012700 ;173600 012700 MOV #10$,RO ;SET RETURN ADDRESS IN RO
3704 006602 173606 ;173602 173606
3705 006604 000755 ;173604 000755 BR REGSAV ;SAVE R1-R7
3706 ;
3707 ; REGISTERS SAVED-- LOOK FOR THE DTE20 WHICH PUSHED THE BUTTON
3708 ;
3709 ; THE DTE WHICH PUSHED THE BUTTON SHOULD HAVE THE DOORBELL
3710 ; RINGING AND HAVE THE VALUE 1365 (OCTAL) IN IT'S
3711 ; TO -10 BYTE COUNT TO10BC.
3712 ;
3713 ; NXM (TIME-OUT) TRAP IS USED TO SKIP NON-EXISTANT DTE20'S.
3714 ;
3715 006606 005005 ;173606 005005 10$: CLR R5 ;ADDRESS LOCATION ZERO
3716 006610 012500 ;173610 012500 MOV (R5)+,RO ;SAVE 0 IN RO
3717 006612 012501 ;173612 012501 MOV (R5)+,R1 ;SAVE 2 IN R1
3718 006614 011502 ;173614 011502 MOV (R5),R2 ;SAVE 4 IN R2
3719 006616 012725 ;173616 012725 MOV #21$, (R5)+ ;SET NXM TRAP ADDRESS IN 4
3720 006620 173634 ;173620 173634
3721 006622 011503 ;173622 011503 MOV (R5),R3 ;SAVE 6 IN R3
3722 006624 012715 ;173624 012715 MOV #PR7, (R5) ;SET PRIORITY FOR NXM TRAP
3723 006626 000340 ;173626 000340
3724 ;
3725 ; LOOP THROUGH ALL DTE'S
3726 ;
3727 006630 012704 ;173630 012704 20$: MOV #DLYCNT-DTESIZ,R4 ;POINT TO DTE # -1'S DELAY COUNT REGISTER
3728 006632 174340 ;173632 174340
3729 ; (WILL BUMP TO # 0)
3730 ;
3731 ; HERE ON NXM TRAP-- RESET SP AND TRY NEXT DTE
3732 ;
3733 006634 012706 ;173634 012706 21$: MOV #4,SP ;SET SP TO 4, STACK IS LOCATIONS 2 AND 0
3734 006636 000004 ;173636 000004
3735 ;
3736 006640 062704 ;173640 062704 22$: ADD #DTESIZ,R4 ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS
3737 006642 000040 ;173642 000040
3738 006644 105704 ;173644 105704 TSTB R4 ;IS THIS THE END OF THE DTE'S?
3739 ; ; NOTE THAT THE LAST DTE IS AT 774540
3740 ; ; AND THAT NOW R4= 774600 IF END
3741 006646 100770 ;173646 100770 BMI 20$ ;YES-- START ALL OVER, UNTIL A DTE
3742 ; ; SAYS HE PUSHED THE BUTTON
3743 006650 032764 ;173650 032764 BIT #TO11DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?

```

B08

3744	006652	004000	:	173652	004000
3745	006654	000034	:	173654	000034
3746	006656	001770	:	173656	001770
3747	006660	026417	:	173660	026417
3748	006662	000014	:	173662	000014
3749		.			
3750	006664	001365	:	173664	001365
3751		:			

BEQ 22\$;NO-- TRY NEXT DTE
CMP 1010BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365

BNE 22\$; IN IT'S TO -10 BYTE COUNT?
;NO-- TRY ANOTHER DTE

;BMB73G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76 09:14 PAGE 5

3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787

006666 010315 ;173666 010315
 006670 010245 ;173670 010245
 006672 010145 ;173672 010145
 006674 010045 ;173674 010045

 006676 012700 ;173676 012700
 006700 000130 ;173670 000130
 006702 012420 ;173702 012420
 006704 022700 ;173704 022700
 006706 000156 ;173706 000156
 006710 103374 ;173710 103374

 006712 005724 ;173712 005724
 006714 010401 ;173714 010401
 006716 012700 ;173716 012700
 006720 000100 ;173720 000100
 006722 010021 ;173722 010021

```

: WE HAVE FOUND THE DTE WHICH PUSHED THE BUTTON
: ADDRESS OF DLYCNT REGISTER IS IN R4
:
MOV R3,(R5) ;RESTORE LOCATION 6
MOV R2,-(R5) ; 4
MOV R1,-(R5) ; 2
MOV R0,-(R5) ; 0
:
SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
IN LOCATIONS 130-156
:
MOV #DTESAV,R0 ;POINT TO SAVE AREA
295: MOV (R4)+,(R0)+ ;SAVE A REGISTER
CMP #T011DT-DLYCNT+DTESAV,R0 ;FINISHED?
BHS 295 ;NO-- SAVE SOME MORE
:
R4= T011DT+2
:
SET R1= STATUS REGISTER
R4= DIAG2 REGISTER
:
DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
LOADED FLAG
:
TST (R4)+
MOV R4,R1 ; SO DOES R1
MOV #DRESET,R0 ;SETUP R0 FOR 'DIAGNOSTIC RESET'
MOV R0,(R1)+ ;R1 POINTS TO STATUS REGISTER
  
```

;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76 09:14 PAGE 6

3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838

```

006724 005061 ;173724 005061
006726 177744 ;173726 177744
006730 005061 ;173730 005061
006732 177764 ;173732 177764
006734 032711 ;173734 032711
006736 004000 ;173736 004000
006740 001775 ;173740 001775
006742 010014 ;173742 010014
006744 005061 ;173744 005061
006746 177766 ;173746 177766
006750 012761 ;173750 012761
006752 107400 ;173752 107400
006754 177762 ;173754 177762
006756 032711 ;173756 032711
006760 004000 ;173760 004000
006762 001775 ;173762 001775
006764 010014 ;173764 010014
006766 012705 ;173766 012705
006770 100000 ;173770 100000
006772 005007 ;173772 005007
000000 ;173774 000
006776 END.YG: ;173776 000
006776 000000 ;173777 000
; 000001
  
```

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 TO10AD-STAT(R1) ;START DUMPING -11 MEMORY TO -10

30S: BIT #TO11DB,(R1) ; STARTING AT LOCATION 0
 ; IS DOORBELL RINGING (TRANSFER COMPLETE)?

BEQ 30S ;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 TO11AD-STAT(R1) ;START INPUT TO LOCATION 0

MOV #IFLOP!<<-256.>&7777>,TO11BC-STAT(R1) ;256 WORDS, INTERRUPT

40S: BIT #TO11DB,(R1) ; -10 WHEN DONE
 ; DOORBELL RINGING (LOAD FINISHED)?

BEQ 40S ;NO-- WAIT UNTIL DONE

MOV R0,(R4) ;CLEAR DOORBELL RINGING

MOV #BIT15,R5 ;SET R5: BIT15= 1, BIT0= 0

CLR PC ; TO SAY BUTTON #4 PRESSED
 ;GO TO LOADED CODE, STARTING AT
 ; LOCATION 0

FILL TO END OF ROM

.BYTE 0

.BYTE 0

.BYTE 0

.BYTE 0

.END

E08

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 94
DZBMD.P11 ROM CONTENTS TABLES

3839 007000 MAP.YH:

3840 :THE FOLLOWING IS A REPRODUCTION
3841 :OF THE ROM PROGRAM FOR BM873YH.
3842 :IT IS HERE FOR COMPARISON TO THE
3843 :ACTUAL ROM AND FOR REFERENCE

3844 :BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 1

3845

3846

3847

3848

3849

3850

3851

3852

3853

3854

3855

3856

3857

3858

3859

3860

3861

3862

3863

3864

3865

;

.SBTTL TITLE PAGE

.TITLE BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23)

.....
COPYRIGHT (C) 1975, 1976 DIGITAL EQUIPMENT CORPORATION
ALL RIGHTS RESERVED

.....
THIS IS THE CODE TO BE ENCODED IN THE BOOTSTRAP ROM ON THE BM873-YH BOARD

.....
MODULE: BM873H

.....
DATE: 10-MAR-76

.....
AUTHOR: TOM PORCHER

.....
.ENABLE ABS,AMA

.LIST MEB

.MCALL SDEF

SDEF

.....
;000000

3866
 3867
 3868
 3869
 3870
 3871
 3872
 3873
 3874
 3875
 3876
 3877
 3878
 3879
 3880
 3881
 3882
 3883
 3884
 3885
 3886
 3887
 3888
 3889
 3890
 3891
 3892
 3893
 3894
 3895
 3896
 3897
 3898
 3899
 3900
 3901
 3902
 3903
 3904
 3905
 3906
 3907
 3908
 3909
 3910
 3911
 3912
 3913
 3914
 3915
 3916
 3917
 3918
 3919

;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 2

.SBTTL MACROS AND DEFINITIONS

MACRO TO FILL TO A LOCATION (RELATIVE TO ROM ORIGIN) WITH ZERO BYTES

```

.MACRO FILLTO LOC
.IFGE <LOC>-<.-ROMORG>
.IFG <LOC>-<.-ROMORG>
.IFDF PASS2
.PRINT <LOC>-<.-ROMORG> ;FREE BYTES AT LOC
.ENDC
.ENDC
.REPT <LOC>-<.-ROMORG>
.BYTE 0
.ENDR
.IFF
.ERROR <.-ROMORG>-<LOC> ;BOUNDARY EXCEEDED AT LOC
.ENDC
.ENDM FILLTO
  
```

MACRO TO DO 'MOV #XXX,DEST' OR 'CLR DEST' IF XXX IS ZERO

```

.MACRO MOVD XXX,DEST
.IFEQ XXX
CLR DEST
.IFF
MOV #XXX,DEST
.ENDC
.ENDM MOVD
  
```

MACRO TO ADD A SMALL NUMBER TO A REGISTER
 GENERATES ONE OF THE FOLLOWING:

```

CMP -(REG),-(REG) ; -4
TST -(REG) ; -2
CMPB -(REG),-(REG) ; -2 (REGISTER MAY BE ODD)
DEC REG ; -1
<NOTHING> ; 0
INC REG ; 1
TST (REG)+ ; 2
CMPB (REG)+,(REG)+ ; 2 (REGISTER MAY BE ODD)
CMP (REG)+,(REG)+ ; 4
ADD #XXX,REG ; ANYTHING ELSE
  
```

USE THIS MACRO WITH CAUTION, SINCE IT REFERENCES MEMORY
 AND ALSO DOES NOT SET THE CONDITION CODES PROPERLY

```

.MACRO ADDX XXX,REG,ODD
SSS=
.IFEQ XXX+4
.IF B <ODD>
  
```

MAINDEC-11-DZBMD-J
DZBMD.P11

MACY11 27(663)
ROM CONTENTS TABLES

2-MAY-77 11:46 PAGE 96

G08

3920
3921

⋮
⋮

.ENDC CMP -(REG),-(REG)

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 98
DZBMD.P11 ROM CONTENTS TABLES

3976 : 001 .IF DF TESTVR
3977 : .ROMORG-2

J08

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 99
DZBMD.P11 ROM CONTENTS TABLES

3978 ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 2-2
3979
3980 :
3981 :
3982 : 173000 .IFF HALT
3983 : 000 .=ROMORG
 : .ENDC

```

3984 ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 3
3985 ;
3986 ; .SBTTL EXTERNAL BUTTONS #1, #2, #3
3987 ;
3988 ; BUTTON #1 -- LOAD USING SWITCH REGISTER
3989 ;
3990 ; BUTON1:
3991 007000 010037 ;:173000 010037 MOV RO,ROTOR7+0 ;SAVE RO IN LOCATION 40
3992 007002 000040 ;:173002 000040 MOV SWR,RO ;GET SWITCH REGISTER
3993 007004 013700 ;:173004 013700 BIT #BIT0,RO ;IS LOW-ORDER BIT SET?
3994 007006 177570 ;:173006 177570 BNE BUTONX ;YES-- LOOK AT CONTENTS
3995 007010 032700 ;:173010 032700 BR REGSAV ;NO-- SAVE R1-R7 IN 42-56, GO TO ADDRESS IN RO (FROM SWR
3996 007012 000001 ;:173012 000001
3997 007014 001007 ;:173014 001007
3998 007016 000513 ;:173016 000513
3999 ;
4000 ; BUTTON #3 -- LOAD BOOT FROM RX11 FLOPPY DISK
4001 ;
4002 ; BUTON3:
4003 007020 005000 ;:173020 005000 CLR RO ;SAY LOAD FROM FLOPPY, UNIT 0
4004 007022 000404 ;:173022 000404 BR BUTONX ;GO TO COMMON CODE FOR 3 BUTTONS
4005 ;
4006 ; REQUIRED POWER-FAIL VECTOR
4007 ;
4008 ; FILLTO 24
4009 007024 173000 ;:173024 173000 .WORD ROMORG,PR7
4010 007026 000340 ;:173026 000340
4011 ;
4012 ; BUTTON #2 -- LOAD BOOT FROM RPO4 DISK
4013 ;
4014 ; BUTON2:
4015 007030 012700 ;:173030 012700 MOV #BIT7,RO ;BIT 7 MEANS LOAD FROM RPO4
4016 007032 000200 ;:173032 000200 BR BUTONX ;FALL INTO COMMON CODE
4017 ;
4018 ; RO IS SAVED IN R5 AS THE PARAMETER WORD PASSED TO BOOT
4019 ; AND CONTAINS ONE OF THE FOLLOWING:
4020 ;
4021 ; BIT 0 = 1 IF FROM SWITCH REGISTER
4022 ; BIT 7 = 0 LOAD FROM RX11 FLOPPY DISK (OR TC11 DECTAPE)
4023 ; BIT 7 = 1 LOAD FROM RPO4 DISK
4024 ; BIT 15 = 1 INDEFINITE RETRY
4025 ;
4026 ;
4027 ; BUTONX:
4028 007034 010005 ;:173034 010005 MOV RO,R5 ;SAVE PARAMETER FOR BOOT
4029 007036 106300 ;:173036 106300 ASLB RO ;LEFT-ALIGN SPEED FIELD IN LOW BYTE
4030 007040 122700 ;:173040 122700 CMPB #3*BIT4,RO ;IS SPEED 0, 1, OR 2?
4031 007042 000060 ;:173042 000060
4032 007044 101001 ;:173044 101001 BHI IOS ;YES-- UNIT IS UNIT TO USE
4033 007046 005000 ;:173046 005000 CLR RO ;NO-- USE UNIT #0
4034 ;
4035 ; IOS:
4035 007050 000300 ;:173050 000300 SWAB RO ;GET UNIT # IN LOW BYTE
4036 007052 042700 ;:173052 042700 BIC #1C7,RO ;TRIM TO 3 BITS 2, 1, 0
4037 007054 177770 ;:173054 177770

```

```

4038
4039
4040
4041 007056 105705 ;173056 105705
4042 007060 100553 ;173060 100553
4043
4044
4045
4046 007062 012737 ;173062 012737
4047 007064 173304 ;173064 173304
4048 007066 000004 ;173066 000004

```

: UNIT # IS IN RO-- CALL PROPER BOOT DEPENDING ON BIT 7
: TSTB R5 ;WHERE SHOULD WE BOOT FROM?
: BMI RPBOOT ;BIT 7 = 1 -- BOOT FROM RPO4 DISK
: BIT 7 = 0 -- BOOT FROM RX11 IF IT EXISTS, ELSE TC11
: MOV #TCBOOT,4 ;SET TIMEOUT TRAP TO TRY DECTAPE

M08

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 102
DZBMD.P11 ROM CONTENTS TABLES

4049 ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 3-1
4050
4051 007070 005037 ;173070 005037 CLR 6 ; . . .
4052 007072 000006 ;173072 000006 ; BR RXBOOT ;TRY FLOPPY FIRST
4053

4054 ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 4

```

4055 ;
4056 ;
4057 ;
4058 ;
4059 ;
4060 ;
4061 ;
4062 ;
4063 ;
4064 ;
4065 ;
4066 ;
4067 ;
4068 ;
4069 ;
4070 ;
4071 ;
4072 ;
4073 ;
4074 ;
4075 ;
4076 ;
4077 ;
4078 ;
4079 ;
4080 ;
4081 ;
4082 ;
4083 ;
4084 ;
4085 ;
4086 ;
4087 ;
4088 ;
4089 ;
4090 ;
4091 ;
4092 ;
4093 ;
4094 ;
4095 ;
4096 ;
4097 ;
4098 ;
4099 ;
4100 ;
4101 ;
4102 ;
4103 ;
4104 ;
4105 ;
4106 ;
4107 ;

```

```

;SBTTL RX11 FLOPPY DISK BOOTSTRAP 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

```

NOTE THAT THE BOOTSTRAP IS WRITTEN IN LOGICAL BLOCK 0 WHICH IS TRACK 1, SECTORS 1, 3, 5, 7. ONLY SECTOR 1 IS READ BY THE ROM.

REGISTER USAGE:
 R0 -- READ FUNCTION WITH UNIT SELECT SET
 R1 -- ADDRESS OF RXCS
 R2 -- ADDRESS OF RXDB
 R3 -- UNIT #
 R4 -- DATA ADDRESS (TO READ OR WRITE)
 R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
 SP -- RETRY COUNTER

HERE TO BOOT FROM RX11 FLOPPY DISK-- UNIT # IN R0

```

4094 ;
4095 007074 012706 ;173074 012706
4096 007076 000014 ;173076 000014
4097 007100 012701 ;173100 012701
4098 007102 177170 ;173102 177170
4099 007104 010003 ;173104 010003
4100 ;
4101 ;
4102 ;
4103 ;
4104 ;
4105 007106 005705 ;173106 005705
4106 007110 100402 ;173110 100402
4107 007112 005306 ;173112 005306

```

```

RXBOOT:
MOV #RETRY,SP ;SET RETRY COUNT
MOV #RXEPA+RXCS,R1 ;ADDRESS CONTROL STATUS REGISTER FOR RX11
MOV R0,R3 ;COPY UNIT #
BR RXRTRY ;FALL THROUGH RETRY CHECK
;
; HERE ON ERROR TO RETRY
RXRTRY:
TST R5 ;INDEFINITE RETRY?
BMI RXRSET ;YES-- TRY AGAIN FULLY
DEC SP ;NO-- DECREMENT RETRY COUNT

```

809

MAINDEC-11-DZBND-J MACY11 27(663) 2-MAY-77 11:46 PAGE 104
DZBND.P11 ROM CONTENTS TABLES

4108 007114 002445 ;173114 002445 BLT RXEHLT ;GIVE UP IF RUN OUT
4109
4110 :
4111 : HERE TO START TRANSFER
:

```

4112 ;BMB73H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 4-1
4113
4114 ;173116
4115 007116 000005 ;173116 000005 RXRSET: RESET ;CLEAR THE WORLD
4116 ;173120 20S: BIT #RXDONE,(R1) ;WAIT UNTIL READY FOR FUNCTION
4117 007120 032711 ;173120 032711
4118 007122 000040 ;173122 000040
4119 007124 001775 ;173124 001775
4120 ;
4121 ;
4122 ;
4123 ;
4124 ;173126 RXPERF:
4125 007126 010300 ;173126 010300 MOV R3,R0 ;GET UNIT #
4126 007130 001402 ;173130 001402 BEQ SS ;ZERO-- USE ZERO
4127 007132 012700 ;173132 012700 MOV #RXUNIT,R0 ;NON-ZERO-- ASSUME UNIT #1
4128 007134 000020 ;173134 000020
4129 ;173136 5S:
4130 007136 052700 ;173136 052700 BIS #RXREAD+RXGO,R0 ;SET READ FUNCTION
4131 007140 000007 ;173140 000007
4132 ;
4133 007142 010102 ;173142 010102 ;
4134 007144 010022 ;173144 010022 MOV R1,R2 ;COPY ADDRESS OF RXCS
4135 ;173146 10S: MOV RO,(R2)+ ;START READ FUNCTION, R2 NOW POINTS TO RXDB
4136 007146 105711 ;173146 105711 TSTB (R1) ;READY?
4137 007150 100376 ;173150 100376 BPL 10S ;NO-- WAIT
4138 007152 012712 ;173152 012712 MOV #1,(R2) ;SET SECTOR #
4139 007154 000001 ;173154 000001
4140 ;173156 20S:
4141 007156 105711 ;173156 105711 TSTB (R1) ;READY FOR TRACK?
4142 007160 100376 ;173160 100376 BPL 20S ;NO-- WAIT
4143 007162 012712 ;173162 012712 MOV #1,(R2) ;SET TRACK #
4144 007164 000001 ;173164 000001
4145 ;173166 30S:
4146 007166 032711 ;173166 032711 BIT #RXERR!RXDONE,(R1) ;DONE OR ERROR?
4147 007170 100040 ;173170 100040
4148 007172 001775 ;173172 001775 BEQ 30S ;NO-- WAIT
4149 007174 100744 ;173174 100744 BMI RXRTRY ;YES-- ERROR IN FUNCTION
4150 ;
4151 ; READ COMPLETED-- EMPTY SILO TO MEMORY
4152 ;
4153 ;173176 RXEMSL:
4154 007176 012711 ;173176 012711 MOV #RXEMPT+RXGO,(R1) ;START EMPTY
4155 007200 000003 ;173170 000003
4156 007202 005004 ;173202 005004 CLR R4 ;ALWAYS START TRANSFER AT LOCATION ZERO
4157 ;173204 10S:
4158 007204 132711 ;173204 132711 BITB #RXTREQ!RXDONE,(R1) ;READY FOR WORD, OR TRANSFER DONE?
4159 007206 000240 ;173206 000240
4160 007210 001775 ;173210 001775 BEQ 10S ;NOT READY-- WAIT SOME MORE
4161 007212 100153 ;173212 100153 BPL CLRPC ;DONE-- GO TO LOCATION 0
4162 007214 111224 ;173214 111224 MOVB (R2),(R4)+ ;NOT DONE-- GET A BYTE FROM SILO TO MEMORY
4163 007216 000772 ;173216 000772 BR 10S ;WAIT FOR NEXT BYTE
4164 ;
4165 ; REQUIRED POWER-FAIL VECTOR

```

009

4166									
4167			;173220			FILLTO	224		
4168	007220	000000	;173220	000		.BYTE	0		
4169			;173221	000		.BYTE	0		
4170	007222	000000	;173222	000		.BYTE	0		
4171			;173223	000		.BYTE	0		
4172	007224	173000	;173224	173000		.WORD	ROMORG,PR7		
4173	007226	000340	;173226	000340					
4174									
4175									

; HERE ON ERROR AFTER RETRYING -- DISPLAY ERROR REGISTER AND HALT

E09

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 107
DZBMD.P11 ROM CONTENTS TABLES

;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 4-2

4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187

007230 012711 :173230
007232 000017 :173232
007234 032711 :173234
007236 000040 :173236
007240 001775 :173240
007242 011200 :173242
007244 000541 :173244

;
RXEHLT:

MOV #RXRERR+RXGO, (R1) ;DO A READ ERROR REGISTER FUNCTION

10\$:

BIT #RXDONE, (R1) ;WAIT UNTIL ERROR ASSEMBLED

BEG 10\$;
MOV (R2), R0 ;GET ERROR REGISTER
BR HALTED ;HALT AND DISPLAY ERRORS

F09

M3INDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 108
DZBMD.P11 ROM CONTENTS TABLES

;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 5

; .SBTTL REGISTER SAVE ROUTINE

REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS R0-R7
IN MEMORY AT 40-56 (LOCATION ROTOR7).

CALLING SEQUENCE:

MOV R0,ROTOR7+0
MOV #RET,RO
BR REGSAV

RET: <RETURN HERE>

ALL REGISTERS RESTORED

4188				
4189				
4190				
4191				
4192				
4193				
4194				
4195				
4196				
4197				
4198				
4199				
4200				
4201				
4202				
4203				
4204			:173246	
4205	007246	010037	:173246	010037
4206	007250	000056	:173250	000056
4207	007252	012700	:173252	012700
4208	007254	000056	:173254	000056
4209	007256	010640	:173256	010640
4210	007260	010540	:173260	010540
4211	007262	010440	:173262	010440
4212	007264	010340	:173264	010340
4213	007266	010240	:173266	010240
4214	007270	010140	:173270	010140
4215	007272	014000	:173272	014000
4216	007274	000177	:173274	000177
4217	007276	004556	:173276	004556

REGSAV:

MOV	R0,ROTOR7+16	;SAVE R0 AS PC IN 56
MOV	#ROTOR7+16,R0	;R0 NOW POINTS TO 56
MOV	SP,-(R0)	;SAVE SP IN 54
MOV	R5,-(R0)	;SAVE R5 IN 52
MOV	R4,-(R0)	;SAVE R4 IN 50
MOV	R3,-(R0)	;SAVE R3 IN 46
MOV	R2,-(R0)	;SAVE R2 IN 44
MOV	R1,-(R0)	;SAVE R1 IN 42
MOV	-(R0),R0	;RESTORE R0 FROM 40
JMP	ROTOR7+16	;GO TO SAVED PC

;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 6

```

4218 ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 6
4219 ;
4220 ;.SBTTL TC11 DECTAPE BOOTSTRAP ROUTINES
4221 ;
4222 ; TC11 REGISTER DEFINITIONS
4223 ;
4224 ; 177340 TCEPA= 177340 ;EXTERNAL PAGE ADDRESS OF TC-11
4225 ;
4226 ; 000000 TCST= 0 ;STATUS REGISTER
4227 ; 100000 TCENDZ= BIT15 ;END-ZONE DETECTED
4228 ; 000002 TCCM= 2 ;COMMAND REGISTER
4229 ; 100000 TCERR= BIT15 ;ERROR
4230 ; 004000 TCREV= BIT11 ;REVERSE DIRECTION (TOWARD FORWARD END-ZONE)
4231 ; 003400 TCUNIT= BIT10!BIT9!BIT8 ;UNIT SELECT
4232 ;
4233 ; 000200 TCRDY= BIT7 ;READY
4234 ; 000016 TCFUNC= BIT3!BIT2!BIT1 ;FUNCTION:
4235 ; 000000 TCSATM= 0*BIT1 ; STOP ALL TAPE MOTION
4236 ; 000002 TCRNUM= 1*BIT1 ; READ BLOCK NUMBER
4237 ; 000004 TCREAD= 2*BIT1 ; READ DATA
4238 ; 000001 TCGO= BIT0 ;START FUNCTION
4239 ; 000004 TCWC= 4 ;WORD COUNT REGISTER
4240 ; 000006 TCBA= 6 ;BUS ADDRESS REGISTER
4241 ;
4242 ; REGISTER USAGE:
4243 ; R0 -- UNIT #
4244 ; R1 -- ADDRESS OF TCCM
4245 ; R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
4246 ;
4247 ; HERE TO START ROM TO BOOT FROM DECTAPE # 0, AS IF
4248 ; DECTAPE BUTTON WERE PUSHED, IN CASE FLOPPY EXISTS.
4249 ;
4250 ; 173300
4251 007300 005000 ;173300 005000
4252 007302 005005 ;173302 005005
4253 ;
4254 ;
4255 ;
4256 ;
4257 ; 173304
4258 007304 012706 ;173304 012706
4259 007306 000014 ;173306 000014
4260 007310 012701 ;173310 012701
4261 007312 177342 ;173312 177342
4262 ;
4263 ;
4264 ; HERE ON ERROR TO RETRY
4265 ;
4266 ; 173314
4267 007314 005705 ;173314 005705
4268 007316 100402 ;173316 100402
4269 007320 005306 ;173320 005306
4270 007322 002427 ;173322 002427
4271 ;173324
    
```

```

TCBOTO:
    CLR    R0          ;HERE TO START WITH A FLOPPY, FROM UNIT 0
    CLR    R5
    BR     TCBOOT     ;GO BOOT

; HERE TO BOOT FROM BLOCK 0 OF DECTAPE, UNIT # IN R0
TCBOOT:
    MOV    #RETRY,SP   ;INIT RETRY COUNTER
    MOV    #TCEPA+TCCM,R1 ;POINT TO COMMAND REGISTER
    BR     TCRTRY      ;TRY IT

; HERE ON ERROR TO RETRY
TCRTRY:
    TST    R5          ;INDEFINITE RETRY?
    BMI    10$         ;YES-- TRY HARDER
    DEC    SP          ;NO-- DECREMENT COUNT
    BLT    TCEHLT     ;TOO MANY-- GIVE UP
10$:
    
```

H09

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 110
DZBMD.P11 ROM CONTENTS TABLES

4272	007324	000005	;173324	000005
4273	007326	010003	;173326	010003
4274	007330	000303	;173330	000303
4275	007332	010304	;173332	010304

RESET
MOV R0,R3
SWAB R3
MOV R3,R4

;CLEAR TC11
;GET UNIT NUMBER
;TO BITS 10-8
;COPY FOR READ FUNCTION


```

4276 ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 6-1
4277
4278 007334 052703 ;173334 052703 BIS #TCREV+TCRNUM+TCGO,R3 ;START TAPE TOWARD FOWARD END-ZONE (BLOCK 0)
4279 007336 004003 ;173336 004003
4280 007340 010311 ;173340 010311 MOV R3,(R1) ;. . .
4281 ;173342 20$: TST (R1) ;ERROR?
4282 007342 005711 ;173342 005711 BPL 20$ ;NO-- WAIT FOR END-ZONE ERROR
4283 007344 100376 ;173344 100376 TST TCST-TCCM(R1) ;END-ZONE UP YET?
4284 007346 005761 ;173346 005761
4285 007350 177776 ;173350 177776
4286 007352 100360 ;173352 100360 BPL TCRTRY ;NO-- MUST BE OTHER ERROR
4287 ;
4288 007354 012761 ;173354 012761 MOV #-256.,TCWC-TCCM(R1) ;SET WORD COUNT
4289 007356 177400 ;173356 177400
4290 007360 000002 ;173360 000002
4291 ;
4292 007362 052704 ;173362 052704 BIS #TCREAD+TCGO,R4 ;NOTE THAT "RESET" CLEARS BUS ADDRESS REGISTER.
4293 007364 000005 ;173364 000005 ;START READ, FORWARD
4294 007366 010411 ;173366 010411 MOV R4,(R1) ;. . .
4295 ;173370 30$: TSTB (R1) ;TRANSFER DONE?
4296 007370 105711 ;173370 105711 BPL 30$ ;NO-- WAIT SOME MORE
4297 007372 100376 ;173372 100376 TST (R1) ;YES-- ERROR?
4298 007374 005711 ;173374 005711 BMI TCRTRY ;YES-- RETRY
4299 007376 100746 ;173376 100746 BR CLRPC ;NO-- DONE-- GOTO LOCATION 0
4300 007400 000460 ;173400 000460
4301 ;
4302 ; HERE ON TC11 ERROR
4303 ;
4304 ;
4305 007402 016100 ;173402 016100 TCEHLT: MOV TCST-TCCM(R1),R0 ;GET STATUS REGISTER
4306 007404 177776 ;173404 177776
4307 007406 000460 ;173406 000460 BR HALTED ;AND STOP
  
```

;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 7

4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361

```

;
; .SBTTL RPO4 DISK BOOTSTRAP ROUTINES
;
; RPO4 REGISTER DEFINITIONS
;
; 176700 RPEPA= 176700 ;EXTERNAL PAGE ADDRESS OF RPO4 REGISTERS
;
; 000000 RPCS1= 0 ;OFFSET FOR CSR #1
; 040000 RPTRE= BIT14 ;TRANSFER ERROR
; 020000 RPMCPE= BIT13 ;MASSBUS CONTROL PARITY ERROR
; 004000 RPDVA= BIT11 ;DRIVE AVAILABLE (TO -11)
; 000200 RPRDY= BIT7 ;FUNCTION COMPLETE
; 000076 RPFUNC= BIT5!BIT4!BIT3!BIT2!BIT1 ;FUNCTION:
; 000020 RPPRST= 20 ; READ-IN PRESET
; 000060 RPWRIT= 60 ; WRITE DATA
; 000070 RPREAD= 70 ; READ DATA
; 000001 RPWC= 2 ;GO
; 000002 RPDA= 6 ;WORD COUNT REGISTER
; 000006 RPCS2= 10 ;TRACK (HIGH BYTE) SECTOR (LOW BYTE)
; 000010 RPUNIT= BIT2!BIT1!BIT0 ;CONTROL AND STATUS REGISTER #2
; 000007 RPDS= 12 ;UNIT #
; 000012 RPATA= BIT15 ;DRIVE STATUS REGISTER
; 040000 RPERR= BIT14 ;ATTENTION ACTIVE
; 000014 RPER1= 14 ;DRIVE ERROR
; 000020 RPFER= BIT4 ;ERROR REGISTER #1
; 000032 RPOF= 32 ;FORMAT ERROR
; 010000 RPFM22= BIT12 ;OFFSET REGISTER
; 004000 RPECCI= BIT11 ;22 SECTOR (16 BIT) FORMAT
; 000034 RPDC= 34 ;INHIBIT ECC CORRECTION
; ;DESIRED CYLINDER

```

REGISTER USAGE:

```

RO -- UNIT #
R1 -- ADDRESS OF RPCS1
R2 -- DATA FOR RPOF: RPECCI (ECC INHIBIT), RPFM22 (22 SECTOR FORMAT)
R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
SP -- RETRY COUNTER

```

HERE TO BOOT FROM RPO4-- UNIT # IN RO

START RPO4 GOING ON BOOT

```

RPBOOT:
MOV #RETRY,SP ;RETRY RETRY TIMES
MOV #RPEPA+RPCS1,R1 ;ADDRESS RPCS1 IN R1
MOV #RPECCI,R2 ;SET ECC INHIBIT, 20 SECTOR MODE
BR RPRTRY ;FALL THROUGH RETRY CODE

```

```

; 173410
; 173410 012706
; 173412 000014
; 173414 012701
; 173416 176700
; 173420 012702
; 173422 004000

```

K09

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 113
DZBMD.P11 ROM CONTENTS TABLES

4362
4363
4364
4365 007424 005705 ;173424
4366 007426 100402 ;173426 100402

; HERE ON ERROR TO RETRY

RPRTY:

TST R5
BMI 10\$

; INFINITE RETRY?
; YES-- TRY AGAIN

4367 ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 7-1

```

4368
4369 007430 005306 :173430 005306 DEC SP ;RETRY COUNT EXHAUSTED?
4370 007432 002444 :173432 002444 BLT RPEHLT ;YES-- GIVE UP
4371 :173434 10S:
4372 007434 000005 :173434 000005 RESET ;ZAP!!
4373 007436 110061 :173436 110061 MOVB RO,RPCS2(R1) ;SELECT PROPER UNIT #
4374 007440 000010 :173440 000010
4375 007442 032711 :173442 032711 BIT #RPDVA,(R1) ;IS DRIVE AVAILABLE TO US?
4376 007444 004000 :173444 004000
4377 007446 001766 :173446 001766 BEQ RPRTRY ;NO-- TRY AGAIN
4378 007450 012711 :173450 012711 MOV #RPPRST+RPGO,(R1) ;DO 'READ-IN PRESET' FUNCTION
4379 007452 000021 :173452 000021
4380 007454 005061 :173454 005061 CLR RPDC(R1) ;SET CYLINDER 0
4381 007456 000034 :173456 000034
4382 007460 005061 :173460 005061 CLR RPDA(R1) ; TRACK 0, SECTOR 0
4383 007462 000006 :173462 000006
4384 007464 050261 :173464 050261 BIS R2,RPOF(R1) ;SET INHIBIT ECC, 22-SECTOR FORMAT (IF FORMAT ERROR)
4385 007466 000032 :173466 000032
4386 007470 012761 :173470 012761 MOV #-256.,RPWC(R1) ;SET UP WORD COUNT TO PROPER VALUE
4387 007472 177400 :173472 177400
4388 007474 000002 :173474 000002
4389 : ;NOTE THAT IT IS NOT NECESSARY TO SET UP BUS
4390 : ;ADDRESS, SINCE IT IS 0 AFTER READ-IN PRESET
4391 007476 012711 :173476 012711 MOV #RPREAD+RPGO,(R1) ;START READ FUNCTION
4392 007500 000071 :173470 000071
4393 :173502 20S:
4394 007502 105711 :173502 105711 TSTB (R1) ;READY?
4395 007504 100376 :173504 100376 BPL 20S ;NO-- WAIT UNTIL IT IS
4396 007506 032761 :173506 032761 BIT #RPFER,RPER1(R1) ;FORMAT ERROR?
4397 007510 000020 :173510 000020
4398 007512 000014 :173512 000014
4399 007514 001403 :173514 001403 BEQ 30S ;NO-- TRY AGAIN
4400 007516 052702 :173516 052702 BIS #RPFM22,R2 ;YES-- TRY FOR 22 SECTOR FLAVOR
4401 007520 010000 :173520 010000
4402 007522 000740 :173522 000740 BR RPRTRY ;TRY AGAIN
4403
4404 :173524 30S:
4405 007524 032711 :173524 032711 BIT #RPTRE!RPMCPE,(R1) ;TRANSFER OR MBC PARITY ERROR?
4406 007526 060000 :173526 060000
4407 007530 001335 :173530 001335 BNE RPRTRY ;YES-- ERROR-- TRY AGAIN
4408 007532 032761 :173532 032761 BIT #RPATA!RPERR,RPDS(R1) ;ATTN OR OTHER ERROR?
4409 007534 140000 :173534 140000
4410 007536 000012 :173536 000012
4411 007540 001331 :173540 001331 BNE RPRTRY ;YES-- ERROR-- TRY AGAIN
4412 : BR CLRPC ;NO ERRORS-- GO TO LOCATION 0
4413 :
4414 : HERE TO GO TO 0
4415 :
4416 :173542 CLRPC:
4417 007542 005007 :173542 005007 CLR PC ;JMP 0
4418 :
4419 : HERE ON ERROR FROM RPO4 AFTER RETRYING-- DISPLAY DRIVE STATUS IN RO
4420 :
    
```

M09

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 115
DZBMD.P11 ROM CONTENTS TABLES

4421			:173544	RPEHLT:			
4422	007544	016100	:173544 016100		MOV	RPDS(R1),RO	;DISPLAY DRIVE STATUS
4423	007546	000012	:173546 000012				
4424				:	BR	HALTED	;R.I.P.
4425				:			
4426			:173550	HALTED:			
4427	007550	000000	:173550 000000		HALT		;DIE
4428	007552	000776	:173552 000776		BR	HALTED	;STAY DEAD

;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 8

; .SBTTL INTERNAL BUTTON #4 -- DUMP AND BOOTSTRAP THROUGH DTE20

;; DTE20 DEFINITIONS

NOTE THAT ALL DTE20 REGISTER DEFINITIONS AND BIT DEFINITIONS
 ARE IN \$DEF IN SYSMAC.SML

000040 DTESIZ= 40 ;EACH DTE OCCUPIES 20 WORDS IN EXTERNAL PAGE
 000004 DTEMAX= 4 ;MAX OF 4 DTE'S ON A PDP-11

;; BUTTON #4 -- INITIATED BY '-10 RELOAD -11' BIT

BUTTON4:

```

MOV    R0,ROTOR7+0    ;SAVE R0 IN 40
MOV    #10$,R0        ;SET RETURN ADDRESS IN R0
BR     REGSAV          ;SAVE R1-R7
  
```

;; REGISTERS SAVED-- LOOK FOR THE DTE20 WHICH PUSHED THE BUTTON

THE DTE WHICH PUSHED THE BUTTON SHOULD HAVE THE DOORBELL
 RINGING AND HAVE THE VALUE 1365 (OCTAL) IN IT'S
 TO -10 BYTE COUNT TO10BC.

;; NXM (TIME-OUT) TRAP IS USED TO SKIP NON-EXISTANT DTE20'S.

10\$:

```

CLR    R5              ;ADDRESS LOCATION ZERO
MOV    (R5)+,R0        ;SAVE 0 IN R0
MOV    (R5)+,R1        ;SAVE 2 IN R1
MOV    (R5),R2         ;SAVE 4 IN R2
MOV    #21$, (R5)+    ;SET NXM TRAP ADDRESS IN 4
MOV    (R5),R3         ;SAVE 6 IN R3
CLR    (R5)            ;SET PS FOR TRAP
  
```

;; LOOP THROUGH ALL DTE'S

20\$:

```

MOV    #DLYCNT-DTESIZ,R4 ;POINT TO DTE # -1'S DELAY COUNT REGISTER
                          ; (WILL BUMP TO # 0)
  
```

;; 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    #DTESIZ,R4      ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS
  
```

```

4429 ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 8
4430 ;
4431 ;
4432 ;
4433 ;
4434 ;
4435 ;
4436 ;
4437 ;
4438 ;
4439 ;
4440 ;
4441 ;
4442 ;
4443 ;
4444 ;
4445 007554 010037 ;173554 010037
4446 007556 000040 ;173556 000040
4447 007560 012700 ;173560 012700
4448 007562 173566 ;173562 173566
4449 007564 000630 ;173564 000630
4450 ;
4451 ;
4452 ;
4453 ;
4454 ;
4455 ;
4456 ;
4457 ;
4458 ;
4459 ;
4460 007566 005005 ;173566 005005
4461 007570 012500 ;173570 012500
4462 007572 012501 ;173572 012501
4463 007574 011502 ;173574 011502
4464 007576 012725 ;173576 012725
4465 007600 173612 ;173570 173612
4466 007602 011503 ;173602 011503
4467 007604 005015 ;173604 005015
4468 ;
4469 ;
4470 ;
4471 ;
4472 007606 012704 ;173606 012704
4473 007610 174340 ;173610 174340
4474 ;
4475 ;
4476 ;
4477 ;
4478 ;
4479 007612 012706 ;173612 012706
4480 007614 000004 ;173614 000004
4481 ;
4482 007616 062704 ;173616 062704
  
```

B10

MAINDEC-11-DZBND-J MACY11 27(663) 2-MAY-77 11:46 PAGE 117
DZBND.P11 ROM CONTENTS TABLES

4483	007620	000040	:173620	000040
4484	007622	105704	:173622	105704
4485		:		
4486		:		
4487	007624	100770	:173624	100770
4488		:		
4489	007626	032764	:173626	032764
4490	007630	004000	:173630	004000
4491	007632	000034	:173632	000034
4492	007634	001770	:173634	001770

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
: YES-- START ALL OVER, UNTIL A DTE
: SAYS HE PUSHED THE BUTTON
: DOORBELL RINGING?

BMI 205

BIT #T011DB,STAT-DLYCNT(R4)

BEQ 225

;NO-- TRY NEXT DTE

C10

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 118
 DZBMD.P11 ROM CONTENTS TABLES

```

4493 ;BMB73H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 8-1
4494
4495 007636 026417 ;173636 026417 CMP T010BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365
4496 007640 000014 ;173640 000014
4497 ; IN IT'S TO -10 BYTE COUNT?
4498 007642 001365 ;173642 001365 BNE 22$ ;NO-- TRY ANOTHER DTE
4499
4500 ; WE HAVE FOUND THE DTE WHICH PUSHED THE BUTTON
4501 ; ADDRESS OF DLYCNT REGISTER IS IN R4
4502
4503
4504 007644 010315 ;173644 010315 MOV R3,(R5) ;RESTORE LOCATION 6
4505 007646 010245 ;173646 010245 MOV R2,-(R5) ; 4
4506 007650 010145 ;173650 010145 MOV R1,-(R5) ; 2
4507 007652 010045 ;173652 010045 MOV R0,-(R5) ; 0
4508
4509 ; SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
4510 ; IN LOCATIONS 130-156
4511
4512 007654 012700 ;173654 012700 MOV #DTESAV,R0 ;POINT TO SAVE AREA
4513 007656 000130 ;173656 000130
4514 ; 29$:
4515 007660 012420 ;173660 012420 MOV (R4)+,(R0)+ ;SAVE A REGISTER
4516 007662 022700 ;173662 022700 CMP #T011DT-DLYCNT+DTESAV,R0 ;FINISHED?
4517 007664 000156 ;173664 000156
4518 007666 103374 ;173666 103374 BHS 29$ ;NO-- SAVE SOME MORE
4519
4520 ; R4= T011DT+2
4521 ; SET R1= STATUS REGISTER
4522 ; R4= DIAG2 REGISTER
4523
4524 ; DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
4525 ; LOADED FLAG
4526
4527
4528 ; ADDX DIAG2-T011DT-2,R4 ;R4 POINTS TO DIAG2 REGISTER
4529 007670 005724 ;173670 005724 TST (R4)+
4530 007672 010401 ;173672 010401 MOV R4,R1 ; SO DOES R1
4531 007674 012700 ;173674 012700 MOV #DRESET,R0 ;SETUP R0 FOR 'DIAGNOSTIC RESET'
4532 007676 000100 ;173676 000100
4533 007700 010021 ;173700 010021 MOV R0,(R1)+ ;R1 POINTS TO STATUS REGISTER
4534
4535 ; REGISTERS:
4536 ; R0 -- DRESET (DIAGNOSTIC RESET FUNCTION)
4537 ; R1 -- STAT (STATUS REGISTER)
4538 ; R4 -- DIAG2 (DIAGNOSTIC REGISTER #2, WHERE DRESET IS)
4539
4540 ; THE -10 WILL NOW START READING -11 MEMORY, AS SOON AS WE SET
4541 ; THE TO -10 ADDRESS. WHEN FINISHED, THE -10 WILL RING OUR DOORBELL.
4542
4543 007702 005061 ;173702 005061 CLR DLYCNT-STAT(R1) ;SET DTE20 FOR MAXIMUM DELAY (ZERO)
4544 007704 177744 ;173704 177744
4545 007706 005061 ;173706 005061 CLR T010AD-STAT(R1) ;START DUMPING -11 MEMORY TO -10
4546 007710 177764 ;173710 177764

```


D10

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 119
DZBMD.P11 ROM CONTENTS TABLES

```
4547 ; ; STARTING AT LOCATION 0
4548 ;:173712
4549 007712 032711 ;:173712 032711 30$: BIT #T011DB,(R1) ;IS DOORBELL RINGING (TRANSFER COMPLETE)?
4550 007714 004000 ;:173714 004000
4551 007716 001775 ;:173716 001775 BEQ 30$ ;NO-- WAIT FOR DOORBELL
4552 007720 010014 ;:173720 010014 MOV R0,(R4) ;YES-- CLEAR DOORBELL AND ERROR FLAGS
4553 ;
4554 ; NOW THE -10 WILL GIVE US A 256 WORD BOOTSTRAP TO BE READ
4555 ; INTO -11 MEMORY STARTING AT LOCATION 0. WHEN FINISHED,
```

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 120
 DZBMD.P11 ROM CONTENTS TABLES

```

4556 ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 8-2
4557
4558 ; THE -10 WILL RING OUR DOORBELL, AND WE WILL START EXECUTION
4559 ; OF THE LOADED CODE AT LOCATION 0.
4560 ;
4561 007722 005061 ;:173722 005061 CLR T011AD-STAT(R1) ;START INPUT TO LOCATION 0
4562 007724 177766 ;:173724 177766
4563 007726 012761 ;:173726 012761 MOV #IFLOP!<<-256.>&7777>,T011BC-STAT(R1) ;256 WORDS, INTERRUPT
4564 007730 107400 ;:173730 107400
4565 007732 177762 ;:173732 177762
4566 ; ; -10 WHEN DONE
4567 ;:173734 40S: BIT #T011DN,(R1) ;TRANSFER COMPLETE?
4568 007734 032711 ;:173734 032711
4569 007736 000200 ;:173736 000200
4570 007740 001775 ;:173740 001775 BEQ 40S ;NO-- WAIT UNTIL DONE
4571 007742 005007 ;:173742 005007 CLR PC ;GO TO LOADED CODE, STARTING AT
4572 ; ; LOCATION 0

```

```

4573 ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM VERSION 3(23) MACY11 27(666) 17-AUG-76 16:19 PAGE 9
4574 ;
4575 ; .SBTTL FILL TO END OF ROM
4576 ;
4577 ;
4578 ;173744 FILLTO 1000
4579 007744 000000 ;173744 000 .BYTE 0
4580 ;173745 000 .BYTE 0
4581 007746 000000 ;173746 000 .BYTE 0
4582 ;173747 000 .BYTE 0
4583 007750 000000 ;173750 000 .BYTE 0
4584 ;173751 000 .BYTE 0
4585 007752 000000 ;173752 000 .BYTE 0
4586 ;173753 000 .BYTE 0
4587 007754 000000 ;173754 000 .BYTE 0
4588 ;173755 000 .BYTE 0
4589 007756 000000 ;173756 000 .BYTE 0
4590 ;173757 000 .BYTE 0
4591 007760 000000 ;173760 000 .BYTE 0
4592 ;173761 000 .BYTE 0
4593 007762 000000 ;173762 000 .BYTE 0
4594 ;173763 000 .BYTE 0
4595 007764 000000 ;173764 000 .BYTE 0
4596 ;173765 000 .BYTE 0
4597 007766 000000 ;173766 000 .BYTE 0
4598 ;173767 000 .BYTE 0
4599 007770 000000 ;173770 000 .BYTE 0
4600 ;173771 000 .BYTE 0
4601 007772 000000 ;173772 000 .BYTE 0
4602 ;173773 000 .BYTE 0
4603 007774 000000 ;173774 000 .BYTE 0
4604 ;173775 000 .BYTE 0
4605 007776 END.YH: ;173776 000 .BYTE 0
4606 007776 000000 ;173777 000 .BYTE 0
4607 ;
4608 ;
4609 ;
4610 ;
4611 ;174000 PASS2:
4612 ; 000001 .END

```

G10

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 122
DZBMD.P11 ROM CONTENTS TABLES

```
4613 010000 MAP.YJ:  
4614 :THE FOLLOWING IS A REPRODUCTION  
4615 :OF THE ROM PROGRAM FOR BM873YJ.  
4616 :IT IS HERE FOR COMPARISON TO THE  
4617 :ACTUAL ROM AND FOR REFERENCE  
4618 :BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 1  
4619  
4620 :  
4621 : .SBTTL TITLE PAGE  
4622 : .TITLE BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  
4623 :  
4624 : COPYRIGHT (C) 1975, 1976, 1977 DIGITAL EQUIPMENT CORPORATION  
4625 : ALL RIGHTS RESERVED  
4626 : THIS IS THE CODE TO BE ENCODED IN THE BOOTSTRAP ROM ON THE BM873-YJ BOARD  
4627 :  
4628 :  
4629 : MODULE: BM873J  
4630 :  
4631 : DATE: 3-SEP-76  
4632 :  
4633 : AUTHOR: TOM PORCHER  
4634 :
```

H10

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 123
DZBMD.P11 ROM CONTENTS TABLES

```
4635 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 2
4636 ;
4637 ; .SBTTL PARAMETERS AND MACROS
4638 ;
4639 ;: PARAMETERS
4640 ;:
4641 ; 173000 B$$873= 173000 ; START OF ROM
4642 ; * D$$BUG= 73000 ; DEBUG VERSION IF DEFINED
4643 ; * P$$ROM= 0 ; WRITE-ME-ON-DISK "ROM" IF DEFINED
4644 ; * E$$DSP= 0 ; DISPLAY ERROR STATUS IN RO FOR ERRORS
4645 ; 000020 R$$TRY= 20 ; RETRY COUNTER *** MUST BE EVEN & .GE. 20 ***
4646 ; 000040 R$$GSV= 40 ; SAVE GENERAL REGISTERS
4647 ; 000000 H$$ALT= 0 ; HALT WHEN "SWR" BUTTON PUSHED
4648 ;
4649 ; 174400 D$$T20= 174400 ; DTE20 SUPPORT
4650 ; 000130 S$$DTE=130 ; SAVE DTE20 REGISTERS DLYCNT TO T011DT
4651 ; 176700 R$$P04= 176700 ; RPO4/RPO6 SUPPORT
4652 ; 000000 F$$M20=0 ; 20 SECTOR (18-BIT) FORMAT
4653 ; 000000 F$$M22=0 ; 22 SECTOR (16-BIT) FORMAT
4654 ; 177170 R$$X11= 177170 ; RX11 (FLOPPY DISK) SUPPORT
4655 ; 177340 T$$C11= 177340 ; TC11 (DECTAPE) SUPPORT
4656 ; 177560 D$$L11= 177560 ; DL11 (REMOTE LOAD LINE) SUPPORT
4657 ; 000000 D$$CMP=0 ; DDCMP STYLE (WHOLE HEADER) FOR DATA
4658 ;
4659 ;: DEFINE ASSEMBLY CONDITIONS
4660 ;:
4661 ;: .ENABLE AMA
4662 ;: .DSABLE GBL
4663 ;: .LIST MEB
4664 ;:
4665 ;: MISC. MCALL'S
4666 ;:
4667 ;: .MCALL CALL,RETURN
4668 ;:
4669 ;: MACROS
4670 ;:
4671 ;: MACRO TO FILL TO A LOCATION (RELATIVE TO ROM ORIGIN) WITH ZERO BYTES
4672 ;:
4673 ;: .MACRO FILLTO LOC
4674 ;: .IF DF B$$873
4675 ;: .IF GE <LOC>-<.-BM873>
4676 ;: .IF G <LOC>-<.-BM873>
4677 ;: .IF DF BM873E
4678 ;: .PRINT <LOC>-<.-BM873> ;FREE BYTES AT LOC
4679 ;: .ENDC
4680 ;: .ENDC
4681 ;: .REPT <LOC>-<.-BM873>
4682 ;: .BYTE 0
4683 ;: .ENDR
4684 ;: .IFF
4685 ;: .ERROR <.-BM873>-<LOC> ;BOUNDARY EXCEEDED AT LOC
4686 ;: .ENDC
4687 ;: .ENDC ; .IF DF B$$873
4688 ;: .ENDM FILLTO
```

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 124
DZBMD.P11 ROM CONTENTS TABLES

4689
4690

;
;

4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744

;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 2-1

; MACRO TO DO 'MOV #XXX,DEST' OR 'CLR DEST' IF XXX IS ZERO

```

;
; .MACRO MOVO XXX,DEST
; .IFEQ XXX
;   CLR DEST
; .IFF
;   MOV #XXX,DEST
; .ENDC
; .ENDM MOVO

```

MACRO TO ADD A SMALL NUMBER TO A REGISTER
GENERATES ONE OF THE FOLLOWING:

```

;   CMP -(REG),-(REG) ; -4
;   TST -(REG) ; -2
;   CMPB -(REG),-(REG) ; -2 (REGISTER MAY BE ODD)
;   DEC REG ; -1
;   <NOTHING> ; 0
;   INC REG ; +1
;   TST (REG)+ ; +2
;   CMPB (REG)+,(REG)+ ; +2 (REGISTER MAY BE ODD)
;   CMP (REG)+,(REG)+ ; +4
;   ADD #XXX,REG ; ANYTHING ELSE

```

USE THIS MACRO WITH CAUTION, SINCE IT REFERENCES MEMORY
AND ALSO DOES NOT SET THE CONDITION CODES PROPERLY

.MACRO ADDX XXX,REG,ODD

```

$$$=.
; .IFEQ XXX+4
;   .IF B <ODD>
;     CMP -(REG),-(REG)
;   .ENDC
; .ENDC
; .IFEQ XXX+2
;   .IF B <ODD>
;     TST -(REG)
;   .IFF
;     CMPB -(REG),-(REG)
;   .ENDC
; .ENDC
; .IFEQ XXX+1
;   DEC REG
; .ENDC
; .IFEQ XXX
;   $$$=$$$+2
; .ENDC
; .IFEQ XXX-1
;   INC REG
; .ENDC
; .IFEQ XXX-2
;   .IF B <ODD>

```

K10

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 126
DZBMD.P11 ROM CONTENTS TABLES

4745 ;
4746 ; .IFF TST (REG)+

4747 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 2-2

```

4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785

```

```

      CMPB      (REG)+,(REG)+
      .ENDC
      .IFEQ XXX-4
      .IF B <00D>
      .ENDC
      CMP      (REG)+,(REG)+
      .ENDC
      .IFEQ $$$-
      .ENDC
      ADD      #XXX,REG
      .ENDC
      .ENDM    ADDX

```

;; GENERAL BIT DEFINITIONS

```

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

```

;; CPU REGISTER DEFINITIONS

```

177570 SWR= 177570 ;SWITCH REGISTER
177776 PS= 177776 ;PROCESSOR STATUS WORD
000340 PR7= 7*BITS ;PRIORITY 7

```

M10

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 128
 DZBMD.P11 ROM CONTENTS TABLES

;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 3

```

4786 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 3
4787 ;
4788 ; .SBTTL EXTERNAL BUTTONS #1, #2, #3
4789 ;
4790 ; ESTABLISH ROM ORIGIN
4791 ;
4792 001 .IF DF BSS873
4793 002 .IF DF DSSBUG
4794 .=.+DSSBUG-2
4795 START:
4796 HALT
4797 .IFF ; .IF DF DSSBUG
4798 .=.+BSS873
4799 ;173000'
4800 001 .ENDC ; .IF DF DSSBUG
4801 .IFF ; .IF DF BSS873
4802 002 .IF DF PSSROM
4803 START:
4804 .=.+PSSROM
4805 MOV PSSROM,RO ;START AT ROM START
4806 MOV (RO)+,2000-2(RO) ;MOVE A WORD
4807 CMP RO,#BM873E ;AT END YET?
4808 BLO 10$ ;NO-- BACK FOR MORE
4809 JMP BM873+2000 ;GO TO NEW CODE
4810 .IFF ; .IF DF PSSROM
4811 START:
4812 001 .ENDC ; .IF DF PSSROM
4813 000 .ENDC ; .IF DF BSS873
4814 ;
4815 ;173000'
4816 ; BM873:
4817 ;
4818 ; BUTTON #1 -- LOAD USING SWITCH REGISTER
4819 ;
4820 ;173000'
4821 001 .IF DF RSSGSV
4822 010000 010037 ;173000' 010037 MOV RO,RSSGSV+0 ;SAVE RO IN LOCATION 40
4823 010002 000040 ;173002 000040
4824 .IFTF ; .IF DF RSSGSV
4825 010004 013700 ;173004' 013700 MOV #SWR,RO ;GET SWITCH REGISTER
4826 010006 177570 ;173006 177570
4827 010010 032700 ;173010' 032700 BIT #BIT0,RO ;IS LOW-ORDER BIT SET?
4828 010012 000001 ;173012 000001
4829 010014 001007 ;173014' 001007 BNE BUTONX ;YES-- LOOK AT CONTENTS
4830 010016 000510 ;173016' 000510 BR REGSAV ;NO-- SAVE R1-R7 IN 42-56, GO TO ADDRESS IN RO (
4831 .IFT ; .IF DF RSSGSV
4832 .IFF ; .IF DF RSSGSV
4833 JMP (RO) ;GO TO ADDRESS FROM SWR
4834 000 .ENDC ; .IF DF RSSGSV
4835 ;
4836 ; BUTTON #3 -- LOAD BOOT FROM RX11 FLOPPY DISK, OR TC11 DECTAPE, OR DL11
4837 ;
4838 ;173020'
4839 010020 005000 ;173020' 005000 CLR RO ;SAY LOAD FROM FLOPPY, UNIT 0
010022 000404 ;173022' 000404 BR BUTONX ;GO TO COMMON CODE FOR 3 BUTTONS

```

N10

4840				:	REQUIRED POWER-FAIL VECTOR		
4841				:			
4842				:			
4843			:173024'	:		FILLTO 24	
4844	010024	173000	:173024'	:	173000'	.WORD	BM873,PR7
4845	010026	000340	:173026 000340	:			

;BMB73J - KL10 (POP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 3-1

```

4846
4847
4848
4849
4850
4851
4852 010030 012700 ;173030'
4853 010032 000200 ;173030'
4854 ;173032 000200
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865 010034 010005 ;173034'
4866 010036 106300 ;173034'
4867 010040 122700 ;173036'
4868 010042 000060 ;173040'
4869 010044 101001 ;173042 000060
4870 010046 005000 ;173044'
4871 ;173046'
4872 010050 000300 ;173050'
4873 010052 042700 ;173050'
4874 010054 177770 ;173052'
4875 ;173054 177770
4876
4877
4878
4879 010056 105705 ;173056' 001
4880 010060 100551 ;173060' 105705
4881 ; 000 100551
4882
4883
4884
4885
4886
4887 010062 012737 ;173062' 001
4888 010064 173274 ;173064 173274 002
4889 010066 000004 ;173066'
4890 010070 005037 ;173070' 005037
4891 010072 000006 ;173072 000006
4892
4893
4894
4895
4896
4897
4898
4899

; BUTTON #2 -- LOAD BOOT FROM RPO4 DISK
;
; BUTON2:
; MOV #BIT7,RO ;BIT 7 MEANS LOAD FROM RPO4
;
; BR BUTONX ;FALL INTO COMMON CODE
;
; RO IS SAVED IN R5 AS THE PARAMETER WORD PASSED TO BOOT
; AND CONTAINS ONE OF THE FOLLOWING:
;
; BIT 0 = 1 IF FROM SWITCH REGISTER
; BIT 7 = 0 LOAD FROM RX11 FLOPPY DISK (OR TC11 DECTAPE)
; BIT 7 = 1 LOAD FROM RPO4 DISK
; BIT 15 = 1 INDEFINITE RETRY
;
; BUTONX:
; MOV RO,R5 ;SAVE PARAMETER FOR BOOT
; ASLB RO ;LEFT-ALIGN SPEED FIELD IN LOW BYTE
; CMPB #3*BIT4,RO ;IS SPEED 0, 1, OR 2?
;
; BHI 105 ;YES-- UNIT IS UNIT TO USE
; CLR RO ;NO-- USE UNIT #0
;
; 105:
; SWAB RO ;GET UNIT # IN LOW BYTE
; BIC #1C7,RO ;TRIM TO 3 BITS 2, 1, 0
;
; UNIT # IS IN RO-- CALL PROPER BOOT DEPENDING ON BIT 7
;
; IF DF RSSP04
; TSTB R5 ;WHERE SHOULD WE BOOT FROM?
; BMI RPBOOT ;BIT 7 = 1 -- BOOT FROM RPO4 DISK
;
; .ENDC ; .IF DF RSSP04
;
; BIT 7 = 0 -- BOOT FROM RX11 IF IT EXISTS, ELSE TC11, ELSE DL11
;
; IF DF RSSX11
; IF DF TSSC11
; MOV #TCBOOT,#4 ;BOOT FROM DECTAPE IF NO FLOPPY
;
; CLR #6 ;SET PS OF TIMEOUT TRAP
;
; .IIF NDF RSSTRY, MOV #40,SP ;SET UP TEMP STACK
; .IFF ; .IF DF TSSC11
; .IF DF DSSL11
; MOV #DLBOOT,#4 ;BOOT FROM DL11 IF NO FLOPPY
; CLR #6 ;SET PS OF TIMEOUT TRAP
; .IIF NDF RSSTRY, MOV #40,SP ;SET UP TEMP STACK
; .ENDC ; .IF DF DSSL11
; .ENDC ; .IF DF TSSC11

```

```

4900      ; BR RXBOOT ;BOOT FROM FLOPPY TO START
4901      ;.IFF ;.IF DF RSSX11
4902      002 .IF DF TSSC11
4903      003 .IF DF DSSL11
4904      MOV #DLBOOT,2#4 ;BOOT FROM DL11 IF NO DECTAPE
4905      CLR 2#6 ;SET PS OF TIMEOUT TRAP
4906      .IIF NDF RSSTRY, MOV #40,SP ;SET UP TEMP STACK
4907      002 .ENDC ;.IF DF DSSL11

```

D11

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 132
DZBMD.P11 ROM CONTENTS TABLES

```

4908 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 3-2
4909
4910 ; BR TCBOOT ;BOOT FROM DECTAPE TO START
4911 ; .IFF ; .IF DF TSSC11
4912 003 ; .IF DF DSSL11
4913 ; BR DLBOOT ;BOOT FROM DL11
4914 ; .IFF ; .IF DF DSSL11
4915 ; .ERROR ; MUST HAVE EITHER RX11, TC11 OR DL11
4916 002 .ENDC ; .IF DF DSSL11
4917 001 .ENDC ; .IF DF TSSC11
4918 000 .ENDC ; .IF DF RSSX11

```

;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 4

4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972

.SBTTL RX11 FLOPPY DISK BOOTSTRAP ROUTINE

001 .IF DF RSSX11

RX11 REGISTER DEFINITIONS

```

RXCS= 0 ; OFFSET FOR CSR
      100000 ; ERROR
      000200 ; TRANSFER REQUEST
      000040 ; TRANSFER DONE
      000020 ; UNIT NUMBER 1
      000016 ; FUNCTION:
      000000 ; FILL SILO
      000002 ; EMPTY SILO
      000004 ; WRITE SECTOR
      000006 ; READ SECTOR
      000016 ; READ ERROR REGISTER
      000001 ; GO BIT
      000002 RXDB= 2 ; MULTI-PURPOSE DATA BUFFER REGISTER
    
```

NOTE THAT THE BOOTSTRAP IS WRITTEN IN LOGICAL BLOCK 0 WHICH IS TRACK 1, SECTORS 1, 3, 5, 7. ONLY SECTOR 1 IS READ BY THE ROM.

REGISTER USAGE:

```

R0 -- UNIT #
R1 -- ADDRESS OF RXCS
R2 -- READ FUNCTION WITH UNIT SELECT SET
R3 -- ADDRESS OF RXDB
R4 -- DATA ADDRESS (TO READ OR WRITE)
R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
SP -- RETRY COUNTER
    
```

HERE TO BOOT FROM RX11 FLOPPY DISK-- UNIT # IN R0

RXBOOT:

```

;173074'
010074 012706 ;173074' 002 .IF DF RSSTRY
010076 000020 ;173076 000020 012706 MOV #RSSTRY,SP ;SET RETRY COUNT
;
010100 012701 ;173100' 012701 .IF DF RSSTRY
010102 177170 ;173102 177170 012701 MOV #RSSX11+RXCS,R1 ;ADDRESS CONTROL STATUS REGISTER FOR RX11
;
010104 005711 ;173104' 003 .IF DF TSSC11!DSSL11
; 005711 TST (R1) ;RX11 EXIST?
; 002 .ENDC ; .IF DF TSSC11!DSSL11
; BR RXRTRY ;FALL THROUGH RETRY CHECK
;
; HERE ON ERROR TO RETRY
    
```

F11

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 134
DZBMD.P11 ROM CONTENTS TABLES

4973			;173106'						
4974									
4975	010106	005705	;173106'	005705	.IF DF RS	TST	RS		; INDEFINITE RETRY?
4976	010110	100402	;173110'	100402		BMI	RXRSET		; YES-- TRY FAITHFULLY


```

4977 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 4-1
4978
4979 010112 005306 ;173112' 005306 DEC SP ;NO-- DECREMENT RETRY COUNT
4980 010114 002531 ;173114' 002531 BLT RXEHLT ;GIVE UP IF RUN OUT
4981 ; 001 .ENDC ; .IF OF RS$TRY
4982 ;
4983 ; HERE TO START TRANSFER
4984 ;
4985 ;173116' RXRSET:
4986 010116 000005 ;173116' 000005 RESET ;CLEAR THE WORLD
4987 ;173120' 20$:
4988 010120 032711 ;173120' 032711 BIT #RXDONE,(R1) ;WAIT UNTIL READY FOR FUNCTION
4989 010122 000040 ;173122' 000040 BEQ 20$ ;NOT YET-- WAIT
4990 010124 001775 ;173124' 001775
4991 ;
4992 ; HERE TO PERFORM READ, UNIT # IN R0
4993 ;
4994 010126 111704 ;173126' 111704 MOVB (PC),R4 ;SET TRACK/SECTOR LOOP COUNT TO 2 *** MOV R0,R2
4995 010130 010002 ;173130' 010002 MOV RO,R2 ;GET UNIT # *** DO NOT MOVE THIS LINE ***
4996 010132 001402 ;173132' 001402 BEQ 30$ ;ZERO-- USE ZERO
4997 010134 012702 ;173134' 012702 MOV #RXUNIT,R2 ;NON-ZERO-- ASSUME UNIT #1
4998 010136 000020 ;173136' 000020
4999 ;173140' 30$:
5000 010140 052702 ;173140' 052702 BIS #RXREAD+RXGO,R2 ;SET READ FUNCTION
5001 010142 000007 ;173142' 000007
5002 ;
5003 010144 010103 ;173144' 010103 MOV R1,R3 ;COPY ADDRESS OF RXCS
5004 010146 010223 ;173146' 010223 MOV R2,(R3)+ ;START READ FUNCTION, R3 NOW POINTS TO RXDB
5005 ;173150' 40$:
5006 010150 105711 ;173150' 105711 TSTB (R1) ;READY?
5007 010152 100376 ;173152' 100376 BPL 40$ ;NO-- WAIT
5008 010154 012713 ;173154' 012713 MOV #1,(R3) ;SET SECTOR #, TRACK # SECOND TIME THROUGH
5009 010156 000001 ;173156' 000001
5010 010160 005304 ;173160' 005304 DEC R4 ;COUNT DOWN SECTOR (1) TRACK (0)
5011 010162 001372 ;173162' 001372 BNE 40$ ;TRACK TO SET STILL-- DO IT
5012 ;173164' 50$:
5013 010164 032711 ;173164' 032711 BIT #RXERR!RXDONE,(R1) ;DONE OR ERROR?
5014 010166 100040 ;173166' 100040
5015 010170 001775 ;173170' 001775 BEQ 50$ ;NO-- WAIT
5016 010172 100745 ;173172' 100745 BMI RXRTRY ;YES-- ERROR IN FUNCTION
5017 ;
5018 ; READ COMPLETED-- EMPTY SILO TO MEMORY
5019 ;
5020 ; NOTE THAT R4 = 0 FROM ABOVE
5021 ;
5022 010174 012711 ;173174' 012711 MOV #RXEMPT+RXGO,(R1) ;START EMPTY
5023 010176 000003 ;173176' 000003
5024 ;173200' 60$:
5025 010200 132711 ;173200' 132711 BITB #RXTREQ!RXDONE,(R1) ;READY FOR WORD, OR TRANSFER DONE?
5026 010202 000240 ;173202' 000240
5027 010204 001775 ;173204' 001775 BEQ 60$ ;NOT READY-- WAIT SOME MORE
5028 010206 100147 ;173206' 100147 BPL CLRPC ;DONE-- GO TO LOCATION 0
5029 010210 111324 ;173210' 111324 MOVB (R3),(R4)+ ;NOT DONE-- GET A BYTE FROM SILO TO MEMORY
5030 010212 000772 ;173212' 000772 BR 60$ ;WAIT FOR NEXT BYTE
  
```

H11

5031
5032
5033
5034
5035
5036
5037
5038
5039

;
;173214'
;173214'
;173216'
;

002 .IF DF TSSC11
BUTONS:
005000 CLR RO
000425 BR TCBOTO
001 .ENDC ; .IF DF TSSC11

HERE TO START ROM TO BOOT FROM DECTAPE # 0, AS IF
DECTAPE BUTTON WERE PUSHED, IN CASE FLOPPY EXISTS.

;HERE TO START WITH A FLOPPY, FROM UNIT 0
;BOOT FROM TAPE

```

5040 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 4-2
5041 ;
5042 ; .IFTF ; .IF DF RSSX11
5043 ; 002 .IF DF HSSALT
5044 ;
5045 ; BUTTON #1 (OPTIONAL) -- HALT, THEN LOAD USING SWITCH REGISTER
5046 ;
5047 ; BUTONO:
5048 010220 000000 ;173220' 000000 HALT ;HALT NOW
5049 010222 000666 ;173222' 000666 BR BUTON1 ;BUT LOOK AT SWR LATER
5050 ; 001 .ENDC ; .IF DF HSSALT
5051 ;
5052 ; REQUIRED POWER-FAIL VECTOR
5053 ;
5054 ; 010224 173230 ;173224' 173000' FILLTO 224
5055 010226 000340 ;173226 000340 .WORD BM873,PR7
5056 ;
5057 ; .IFT ; .IF DF RSSX11
5058 ;
5059 ; HERE ON ERROR AFTER RETRYING -- DISPLAY ERROR REGISTER AND HALT
5060 ;
5061 ; 002 .IF DF ESSDSP
5062 ; RXEHLT:
5063 ; MOV #RXRERR+RXGO,(R1) ;DO A READ ERROR REGISTER FUNCTION
5064 ; 10$:
5065 ; BIT #RXDONE,(R1) ;WAIT UNTIL ERROR ASSEMBLED
5066 ; BEQ 10$ ;
5067 ; MOV (R3),R0 ;GET ERROR REGISTER
5068 ; BR HALTED ;HALT AND DISPLAY ERRORS
5069 ;
5070 ; 001 .ENDC ; .IF DF ESSDSP
5071 ; 000 .ENDC ; .IF DF RSSX11

```

J11

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 138
DZBMD.P11 ROM CONTENTS TABLES

```
5071 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 5
5072 ;
5073 ; .SBTTL INTERNAL BUTTON #4 -- DUMP AND BOOTSTRAP THROUGH DTE20
5074 ;
5075 ; BUTTON #4 -- INITIATED BY '-10 RELOAD -11' BIT
5076 ;
5077 ; 001 .IF DF DSST20&RSSGSV
5078 ; ;173230' BUTON4:
5079 010230 010037 ;173230' 010037 MOV RO,RSSGSV+0 ;SAVE RO IN 40
5080 010232 C00040 ;173232 000040
5081 010234 012700 ;173234' 012700 MOV #DTE20,RO ;SET RETURN ADDRESS IN RO
5082 010236 173622 ;173236 173622
5083 ; BR REGSAV ;SAVE R1-R7
5084 ; 000 .ENDC ; .IF DF DSST20&RSSGSV
```

;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 6

; .SBTTL REGISTER SAVE ROUTINE

; 001 .IF DF R\$SGSV

REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS R0-R7
IN MEMORY AT 40-56 (LOCATION R\$SGSV).

CALLING SEQUENCE:

MOV R0,R\$SGSV+0
MOV #RET,R0
BR REGSAV

RET: <RETURN HERE>

ALL REGISTERS RESTORED

5103			:173240'		REGSAV:			
5104	010240	010037	:173240'	010037	MOV	R0,R\$SGSV+16		;SAVE R0 AS PC IN 56
5105	010242	000056	:173242	000056				
5106	010244	012700	:173244'	012700	MOV	#R\$SGSV+16,R0		;R0 NOW POINTS TO 56
5107	010246	000056	:173246	000056				
5108	010250	010640	:173250'	010640	MOV	SP,-(R0)		;SAVE SP IN 54
5109	010252	010540	:173252'	010540	MOV	R5,-(R0)		;SAVE R5 IN 52
5110	010254	010440	:173254'	010440	MOV	R4,-(R0)		;SAVE R4 IN 50
5111	010256	010340	:173256'	010340	MOV	R3,-(R0)		;SAVE R3 IN 46
5112	010260	010240	:173260'	010240	MOV	R2,-(R0)		;SAVE R2 IN 44
5113	010262	010140	:173262'	010140	MOV	R1,-(R0)		;SAVE R1 IN 42
5114	010264	014000	:173264'	014000	MOV	-(R0),R0		;RESTORE R0 FROM 40
5115	010266	000177	:173266'	000177	JMP	@R\$SGSV+16		;GO TO SAVED PC
5116	010270	004564	:173270	000056				
5117			000		.ENDC ; .IF DF R\$SGSV			

```

5118 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 7
5119 ;
5120 ; .SBTTL TC11 DECTAPE BOOTSTRAP ROUTINE
5121 ;
5122 ; 001 .IF DF TSSC11
5123 ;
5124 ; TC11 REGISTER DEFINITIONS
5125 ;
5126 ; 000000 TCST= 0 ;STATUS REGISTER
5127 ; 100000 TCENDZ= BIT15 ;END-ZONE DETECTED
5128 ; 000002 TCCM= 2 ;COMMAND REGISTER
5129 ; 100000 TCERR= BIT15 ;ERROR
5130 ; 004000 TCREV= BIT11 ;REVERSE DIRECTION (TOWARD FORWARD END-ZONE)
5131 ; 003400 TCUNIT= BIT10!BIT9!BIT8 ;UNIT SELECT
5132 ;
5133 ; 000200 TCRDY= BIT7 ;READY
5134 ; 000016 TCFUNC= BIT3!BIT2!BIT1 ;FUNCTION:
5135 ; 000000 TCSATM= 0*BIT1 ;STOP ALL TAPE MOTION
5136 ; 000002 TCRNUM= 1*BIT1 ;READ BLOCK NUMBER
5137 ; 000004 TCREAD= 2*BIT1 ;READ DATA
5138 ; 000001 TCGO= BIT0 ;START FUNCTION
5139 ; 000004 TCWC= 4 ;WORD COUNT REGISTER
5140 ; 000006 TCBA= 6 ;BUS ADDRESS REGISTER
5141 ;
5142 ; REGISTER USAGE:
5143 ; RO -- UNIT #
5144 ; R1 -- ADDRESS OF TCCM
5145 ; R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
5146 ; SP -- RETRY COUNTER
5147 ;
5148 ;
5149 ; HERE TO CONTINUE WITH BOOT FROM UNIT 0 IF FLOPPY EXISTS, UNIT # IN RO
5150 ;
5151 ; 002 .IF DF RSSX11
5152 ; TCBOOT:
5153 010272 005005 ;173272' 005005 CLR R5 ;SET SWR PARAMETER TO ZERO, ALSO
5154 ; BR TCBOOT ;ALL SET!
5155 ; 001 .ENDC ; .IF DF RSSX11
5156 ;
5157 ; HERE TO BOOT FROM BLOCK 0 OF DECTAPE, UNIT # IN RO
5158 ;
5159 ; 002 TCBOOT:
5160 ; .IF DF DSSL11
5161 010274 012737 ;173274' 012737 000004 MOV #DLBOOT,2#4 ;IN CASE NO DTA, TRY DL11
5162 010276 173530 ;173276 173530
5163 ; 001 .ENDC ; .IF DF DSSL11
5164 ; 002 .IF DF RSSTRY
5165 010300 000004 ;173300
5166 010302 012706 ;173302' 012706 MOV #RSSTRY,SP ;INIT RETRY COUNTER
5167 010304 000020 ;173304 000020
5168 ; .IFTF ; .IF DF RSSTRY
5169 010306 012701 ;173306' 012701 MOV #TSSC11+TCCM,R1 ;POINT TO COMMAND REGISTER
5170 010310 177342 ;173310 177342
5171 ; BR TCRTY ;TRY IT

```

M11

```

5172
5173
5174
5175
5176
5177 010312 105011 ;173312' 003 .IF DF RSSTRY:DSSL11
                                     TCRTRY:
                                     105011 CLR8 (R1)
                                     ;STOP ALL TAPE MOTION, TRAP IF NO TC11

```

```

5178 ;BMB73J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 7-1
5179
5180 ; 002 .ENDC ; .IF DF RSSTRY!DSSL11
5181 ; IFT ; .IF DF RSSTRY
5182 010314 005705 ;173314' 005705 TST R5 ; INDEFINITE RETRY?
5183 010316 100402 ;173316' 100402 BMI 10S ; YES-- TRY HARDER
5184 010320 005306 ;173320' 005306 DEC SP ; NO-- DECREMENT COUNT
5185 010322 002426 ;173322' 002426 BLT TCEHLT ; TOO MANY-- GIVE UP
5186 ;
5187 ; 001 .ENDC ; .IF DF RSSTRY
5188 010324 000005 ;173324' 000005 10S: RESET ; CLEAR TC11
5189 010326 110061 ;173326' 110061 MOVB RO,1(R1) ; SELECT PROPER UNIT
5190 010330 000001 ;173330' 000001
5191 010332 052711 ;173332' 052711 BIS #TCREV+TCRNUM+TCGO,(R1) ; START TAPE TOWARD FOWARD END-ZONE (BLOC
5192 010334 004003 ;173334' 004003
5193 ; 20S:
5194 010336 005711 ;173336' 005711 TST (R1) ; ERROR?
5195 010340 100376 ;173340' 100376 BPL 20S ; NO-- WAIT FOR END-ZONE ERROR
5196 010342 005761 ;173342' 005761 TST TCST-TCCM(R1) ; END-ZONE UP YET?
5197 010344 177776 ;173344' 177776
5198 010346 100361 ;173346' 100361 BPL TCRTRY ; NO-- MUST BE OTHER ERROR
5199 ;
5200 010350 012761 ;173350' 012761 MOV #-256.,TCWC-TCCM(R1) ; SET WORD COUNT
5201 010352 177400 ;173352' 177400
5202 010354 000002 ;173354' 000002
5203 ;
5204 010356 042711 ;173356' 042711 BIC #TCREV,(R1) ; NOTE THAT "RESET" CLEARS BUS ADDRESS REGISTER.
5205 010360 004000 ;173360' 004000 ; SET FORWARD MODE
5206 010362 112711 ;173362' 112711 MOVB #TCREAD+TCGO,(R1) ; START READ, FORWARD
5207 010364 000005 ;173364' 000005
5208 ; 30S:
5209 010366 105711 ;173366' 105711 TSTB (R1) ; TRANSFER DONE?
5210 010370 100376 ;173370' 100376 BPL 30S ; NO-- WAIT SOME MORE
5211 010372 005711 ;173372' 005711 TST (R1) ; YES-- ERROR?
5212 010374 100746 ;173374' 100746 BMI TCRTRY ; YES-- RETRY
5213 010376 005007 ;173376' 005007 CLR PC ; NO-- DONE-- GOTO LOCATION 0
5214 ;
5215 ; HERE ON TC11 ERROR
5216 ;
5217 ; 002 .IF DF ESSDSP
5218 ; TCEHLT:
5219 ; MOV TCST-TCCM(R1),RO ; GET STATUS REGISTER
5220 ; BR HALTED ; AND STOP
5221 ; 001 .ENDC ; .IF DF ESSDSP
5222 ; 000 .ENDC ; .IF DF TSSC11

```



```

5223 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 8
5224 ;
5225 ; .SBTTL ERROR HANDLING
5226 ;
5227 ;
5228 ; 001 ; .IF DF RSSTRY
5229 ; 002 ; .IF DF ESSDSP
5230 ; .IFF
5231 ; .IIF DF RSSP04, RPEHLT:
5232 ; .IIF DF RSSX11, RXEHLT:
5233 ; .IIF DF TSSC11, TCEHLT:
5234 ; .IIF DF DSSL11, DLEHLT:
5235 ;
5236 ; 001 .ENDC ; .IF DF ESSDSP
5237 ;
5238 ; HALTED:
5239 ; 000000 HALT ;DIE
5240 ; 000776 BR HALTED ;STAY DEAD
5241 ; .IFF ; .IF DF RSSTRY
5242 ; .IIF DF ESSDSP, .ERROR ; CANNOT HAVE ESSDSP WITHOUT RSSTRY
5243 ; 000 .ENDC ; .IF DF RSSTRY

```

;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 9

.SBTTL RPO4 DISK BOOTSTRAP ROUTINE

001 .IF DF RSSPO4

RPO4 REGISTER DEFINITIONS

```

000000 RPCS1= 0 ; OFFSET FOR CSR #1
040000 RPTRE= BIT14 ; TRANSFER ERROR
020000 RPMCPE= BIT13 ; MASSBUS CONTROL PARITY ERROR
004000 RPDVA= BIT11 ; DRIVE AVAILABLE (TO -11)
000200 RPRDY= BIT7 ; FUNCTION COMPLETE
000076 RPFUNC= BITS!BIT4!BIT3!BIT2!BIT1 ; FUNCTION:
000020 RPPRST= 20 ; READ-IN PRESET
000060 RPWRIT= 60 ; WRITE DATA
000070 RPREAD= 70 ; READ DATA
000001 RPGO= BIT0 ; GO
000002 RPWC= 2 ; WORD COUNT REGISTER
000006 RPDA= 6 ; TRACK (HIGH BYTE) SECTOR (LOW BYTE)
000010 RPCS2= 10 ; CONTROL AND STATUS REGISTER #2
000007 RPUNIT= BIT2!BIT1!BIT0 ; UNIT #
000012 RPDS= 12 ; DRIVE STATUS REGISTER
100000 RPATA= BIT15 ; ATTENTION ACTIVE
040000 RPERR= BIT14 ; DRIVE ERROR
000014 RPER1= 14 ; ERROR REGISTER #1
000020 RPFER= BIT4 ; FORMAT ERROR
000032 RPOF= 32 ; OFFSET REGISTER
010000 RPFM22= BIT12 ; 22 SECTOR (16 BIT) FORMAT
004000 RPECCI= BIT11 ; INHIBIT ECC CORRECTION
000034 RPDC= 34 ; DESIRED CYLINDER

```

REGISTER USAGE:

```

R0 -- UNIT #
R1 -- ADDRESS OF RPCS1
R2 -- DATA FOR RPOF: RPECCI (ECC INHIBIT), RPFM22 (22 SECTOR FORMAT)
R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
SP -- RETRY COUNTER

```

HERE TO BOOT FROM RPO4-- UNIT # IN R0

START RPO4 GOING ON BOOT

RPBOOT:

```

5242 ;173404'
5243
5244 ;
5245
5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256
5257
5258
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287 ;173404'
5288 ; 002
5289 010404 012706 ;173404' 012706 MOV #RSSTRY,SP ;RETRY RETRY TIMES
5290 010406 000020 ;173406 000020
5291 ;
5292 010410 012701 ;173410' 012701 .IFTF ; .IF DF RSSTRY MOV #RSSPO4+RPCS1,R1 ;ADDRESS RPCS1 IN R1
5293 010412 176700 ;173412 176700
5294 ; 003
5295 010414 012702 ;173414' 012702 MOV #RPECCI,R2 ;SET ECC INHIBIT, 20 SECTOR MODE

```

D12

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 145
DZBMD.P11 ROM CONTENTS TABLES

```
5296 010416 004000 ;173416 004000  
5297 ; 002 .ENDC ; .IF DF FSSM20&FSSM22  
5298 ; BR RPRTRY ;FALL THROUGH RETRY CODE  
5299 ;  
5300 ; HERE ON ERROR TO RETRY
```

```

5301 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 9-1
5302
5303
5304 ;173420' ;
5305 .IFT ; .IF DF RS$TRY
5306 010420 005705 ;173420' 005705 TST R5 ;INFINITE RETRY?
5307 010422 100402 ;173422' 100402 BMI RPRSET ;YES-- TRY AGAIN
5308 010424 005306 ;173424' 005306 DEC SP ;RETRY COUNT EXHAUSTED?
5309 010426 002764 ;173426' 002764 BLT RPEHLT ;YES-- GIVE UP
5310 ;
5311 001 .ENDC ; .IF DF RS$TRY
5312 ;
5313 ;
5314 ;
5315 010430 000005 ;173430' 000005 RPRSET: RESET ;ZAP!!
5316 010432 110061 ;173432' 110061 MOVB R0,RPCS2(R1) ;SELECT PROPER UNIT #
5317 010434 000010 ;173434' 000010 MOV #RPPRST+RPGO,(R1) ;DO 'READ-IN PRESET' FUNCTION
5318 010436 012711 ;173436' 012711 CLR RPDC(R1) ;SET CYLINDER 0
5319 010440 000021 ;173440' 000021 CLR RPDA(R1) ; TRACK 0, SECTOR 0
5320 010442 005061 ;173442' 005061
5321 010444 000034 ;173444' 000034
5322 010446 005061 ;173446' 005061
5323 010450 000006 ;173450' 000006
5324 ;
5325 010452 050261 ;173452' 050261 .IF DF FSSM20&FSSM22 BIS R2,RPOF(R1) ;SET INHIBIT ECC, 22-SECTOR FORMAT (IF FORMAT ER
5326 010454 000032 ;173454' 000032
5327 ;
5328 003 .IFF
5329 ;
5330 002 .IF DF FSSM20 BIS #RPECCI,RPOF(R1) ;SET ECC INHIBIT, 20 SECTOR FORMAT
5331 003 .ENDC ; .IF DF FSSM20
5332 ;
5333 002 .IF DF FSSM22 BIS #RPECCI+RPFM22,RPOF(R1) ;SET ECC INHIBIT, 22 SECTOR FORMAT
5334 ;
5335 010456 012761 ;173456' 012761 .ENDC ; .IF DF FSSM22
5336 010460 177400 ;173460' 177400 .IFTF ; .IF DF FSSM20&FSSM22 MOV #-256.,RPWC(R1) ;SET UP WORD COUNT TO PROPER VALUE
5337 010462 000002 ;173462' 000002
5338 ;
5339 ;
5340 010464 012711 ;173464' 012711 MOV #RPREAD+RPGO,(R1) ;START READ FUNCTION
5341 010466 000071 ;173466' 000071
5342 ;
5343 010470 105711 ;173470' 105711 20$: TSTB (R1) ;READY?
5344 010472 100376 ;173472' 100376 BPL 20$ ;NO-- WAIT UNTIL IT IS
5345 ;
5346 010474 032761 ;173474' 032761 .IFT ; .IF DF FSSM20&FSSM22 BIT #RPFER,RPER1(R1) ;FORMAT ERROR?
5347 010476 000020 ;173476' 000020
5348 010500 000014 ;173470' 000014
5349 010502 001402 ;173502' 001402 BEQ 30$ ;NO-- TRY AGAIN
5350 010504 052702 ;173504' 052702 BIS #RPFM22,R2 ;YES-- TRY FOR 22 SECTOR FLAVOR
5351 010506 010000 ;173506' 010000
5352 ;
5353 ;
5354 010510 032711 ;173510' 032711 .ENDC ; .IF DF FSSM20&FSSM22
5355 30$: BIT #RPTRE!RPMCPE,(R1) ;TRANSFER OR MBC PARITY ERROR?

```

```

5355 010512 060000 ;173512 060000
5356 010514 001341 ;173514' 001341 BNE RPRTRY ;YES-- ERROR-- TRY AGAIN
5357 010516 032761 ;173516' 032761 BIT #RPATA!RPERR,RPDS(R1) ;ATTN OR OTHER ERROR?
5358 010520 140000 ;173520 140000
5359 010522 000012 ;173522 000012
5360 010524 001335 ;173524' 001335 BNE RPRTRY ;YES-- ERROR-- TRY AGAIN
5361 ; 000 ; BR CLRPC ;NO ERRORS-- GO TO LOCATION 0
5362 ; .ENDC ; .IF DF R$SP04
5363 ;
5364 ; HERE TO GO TO 0
5365 ;
5366 ;173526' CLRPC:
5367 010526 005007 ;173526' 005007 CLR PC ;JMP 0
5368 ;
5369 ; HERE ON ERROR FROM RPO4 AFTER RETRYING-- DISPLAY DRIVE STATUS IN RO
5370 ;

```

G12

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 148
DZBMD.P11 ROM CONTENTS TABLES

;BMB73J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 9-2

5371
5372
5373
5374
5375
5376
5377

:
:
:
:
:
:
:

001 .IF DF ESSDSP
RPEHLT:
MOV RPDS(R1),R0 ;DISPLAY DRIVE STATUS
BR HALTED ;R.I.P.
000 .ENDC ; .IF DF ESSDSP

```

5378 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 10
5379 ;
5380 ; .SBTTL DL11 ASYNCHONOUS LINE BOOTSTRAP ROUTINE
5381 ;
5382 ; 001 ; IF DF DSSL11
5383 ;
5384 ; DL11 REGISTER DEFINITIONS
5385 ;
5386 ; 000000 DLRCR= 0 ;RECIEVER CONTROL/STATUS REGISTER
5387 ; 000200 DLRDON= BIT7 ;DONE
5388 ; 000002 DLRBUF= 2 ;RECIEVER DATA BUFFER
5389 ;
5390 ; DL11 BOOT IS OF THE FOLLOWING FORMAT:
5391 ;
5392 ; ... <DLE><CC1><CC2><FILL><... DATA ...>
5393 ;
5394 ; WHERE <DLE> IS 220(8).
5395 ; <CC1>, <CC2> FORM THE LOW AND HIGH BYTES OF THE BYTE COUNT FOR THE
5396 ; FOLLOWING DATA, RESPECTIVELY. THE TWO HIGH-ORDER
5397 ; BITS (QSYNC & SELECT) ARE IGNORED (DSSCMP DEFINED).
5398 ; <FILL> IS 5 BYTES OF IGNORED DATA (DSSCMP DEFINED).
5399 ; <DATA> IS A BYTE STREAM OF 8-BIT DATA OF LENGTH DETERMINED BY
5400 ; <CC2><CC1>.
5401 ;
5402 ; REGISTER USAGE:
5403 ; R1 -- ADDRESS OF DLRCR
5404 ; R2 -- ADDRESS OF "DLCHAR" ROUTINE
5405 ; R3 -- BYTE COUNT
5406 ; R4 -- MEMORY ADDRESS
5407 ; R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
5408 ; SP -- STACK POINTER (STACK AT 5004-)
5409 ;
5410 ;
5411 ; START HERE TO BOOT FROM DL11
5412 ;
5413 ; BUTON6:
5414 ; DLBOOT:
5415 010530 012701 ;173530' 012701 MOV #DSSL11+DLRCR,R1 ;GET DL11 EXTERNAL PAGE ADDRESS
5416 010532 177560 ;173532' 177560
5417 010534 011706 ;173534' 011706 MOV (PC),SP ;SET TEMP STACK AT 5004- *** CLR R4 MUST BE NEXT
5418 010536 005004 ;173536' 005004 CLR R4 ;RESET MEMORY ADDRESS FOR DLCHAR *** DO NOT MOV
5419 010540 012702 ;173540' 012702 MOV #DLCHAR,R2 ;SET ADDRESS OF GET CHARACTER ROUTINE
5420 010542 173610 ;173542' 173610
5421 ;173544'
5422 ;173544' 10S:
5423 010544 004712 ;173544' 004712 CALL (R2) ;GET A CHARACTER
5424 010546 124427 ;173546' 124427 JSR PC,(R2)
5425 010550 000220 ;173550' 000220 CMPB -(R4),#220 ;DLE?
5426 010552 001374 ;173552' 001374 BNE 10S ;NO-- KEEP ON LOOKING
5427 ; ; 002 .IF DF DSSCMP
5428 ;173554' CALL -(R2) ;GET TWO BYTES
5429 010554 004742 ;173554' 004742 JSR PC,-(R2)
5430 ; .IFF ; .IF DF DSSCMP
5431 ; CALL (R2) ;GET TWO
    
```

5432		:		CALL (R2)		; BYTES
5433		:		.IFTF ; .IF DF DSSCMP		
5434	010556	014403	;173556'	014403	MOV	-(R4),R3 ;GET BYTE COUNT
5435		:		.IFT ; .IF DF DSSCMP		
5436	010560	042703	;173560'	042703	BIC	#BIT15!BIT14,R3 ;CLEAR QSYNC AND SELECT BITS
5437	010562	140000	;173562 140000			


```

5438 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 10-1
5439
5440 ;173564' CALL (R2) ;SKIP TWO MORE BYTES
5441 010564 004712 ;173564' 004712 JSR PC,(R2)
5442 ;173566' CALL (R2) ; AND TWO MORE
5443 010566 004712 ;173566' 004712 JSR PC,(R2)
5444 010570 005722 ;173570' 005722 TST (R2)+ ;SKIP THE CALL (PC)
5445 ;173572' CALL (R2) ; +1 MAKES 5 BYTES TO SKIP
5446 010572 004712 ;173572' 004712 JSR PC,(R2)
5447 010574 005004 ;173574' 005004 CLR R4 ;RESET BACK TO 0
5448 ; .IFTF ; .IF DF DSSCMP
5449 ; 20$:
5450 ;173576' CALL (R2) ;GET A CHARACTER
5451 010576 004712 ;173576' 004712 JSR PC,(R2)
5452 010600 005303 ;173600' 005303 DEC R3 ;REDUCE COUNT
5453 010602 003375 ;173602' 003375 BGT 20$ ;BACK IF MORE
5454 010604 005007 ;173604' 005007 CLR PC ;ELSE GO TO LOADED CODE
5455
5456 ; DLCHAR -- GET CHARACTER FROM DL11
5457 ; R1 -- ADDRESS OF DL11
5458 ; R4 -- ADDRESS OF WHERE TO STORE CHARACTER, INCREMENTED
5459
5460 ; CALL DLCHAR-2 FOR TWO BYTES (DSSCMP ONLY)
5461
5462 ; .IFT ; .IF DF DSSCMP
5463 ;173606' CALL (PC) ;GET A BYTE, THEN ANOTHER
5464 010606 004717 ;173606' 004717 JSR PC,(PC)
5465 ; .ENDC ; .IF DF DSSCMP
5466 ; DLCHAR:
5467 010610 105711 ;173610' 105711 TSTB (R1) ;READY WITH A CHARACTER?
5468 010612 100376 ;173612' 100376 BPL DLCHAR ;NO-- WAIT SOME MORE
5469 010614 116124 ;173614' 116124 MOVB DLRBUF-DLRCSR(R1),(R4)+ ;YES-- STORE THE CHARACTER
5470 010616 000002 ;173616' 000002
5471 ;173620' RETURN ;AND RETURN FROM DLCHAR
5472 010620 000207 ;173620' 000207 RTS PC
5473 ; .ENDC ; .IF DF DSSL11
  
```

```

5474 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 11
5475 ;
5476 ; .SBTTL DTE20 DUMP AND BOOTSTRAP ROUTINE
5477 ;
5478 ; 001 .IF DF D$S$T20
5479 ;
5480 ; DTE20 REGISTER DEFINITIONS
5481 ;
5482 ; 000000 DLYCNT= 0 ; DELAY COUNTER
5483 ; 000002 DEXWD3= 2 ; DEPOSIT/EXAMINE WORD 3
5484 ; 000004 DEXWD2= 4 ; . . . 2
5485 ; 000006 DEXWD1= 6 ; . . . 1
5486 ; 000010 TENAD1= 10 ; KL-10 MEMORY ADDRESS 1
5487 ; 000012 TENAD2= 12 ; . . . 2
5488 ; 000014 T010BC= 14 ; T0 -10 BYTE (WORD) COUNT
5489 ; 000016 T011BC= 16 ; T0 -11 BYTE (WORD) COUNT
5490 ; 100000 IFLOP= BIT15 ; INTERRUPT BOTH -10 AND -11 WHEN DONE
5491 ; 040000 T011BM= BIT14 ; T0 -11 BYTE MODE
5492 ; 007777 BCOUNT= 7777 ; BYTE (WORD) COUNT
5493 ; 000020 T010AD= 20 ; T0 -10 ADDRESS
5494 ; 000022 T011AD= 22 ; T0 -11 ADDRESS
5495 ; 000024 T010DT= 24 ; T0 -10 DATA WORD
5496 ; 000026 T011DT= 26 ; T0 -11 DATA WORD
5497 ; 000030 DIAG1= 30 ; DIAGNOSTIC/CONTROL REGISTER 1
5498 ; 000032 DIAG2= 32 ; DIAGNOSTIC REGISTER 2
5499 ; 000100 DRESET= BIT6 ; DIAGNOSTIC DTE20 RESET
5500 ; 000034 STAT= 34 ; CONTROL/STATUS REGISTER
5501 ; 004000 T011DB= BIT11 ; T0 -11 DOORBELL
5502 ; 000200 T011DN= BIT7 ; T0 -11 TRANSFER DONE
5503 ; 000036 DIAG3= 36 ; . . . 3
5504 ;
5505 ; 000040 DTESIZ= 40 ; EACH DTE OCCUPIES 20 WORDS IN EXTERNAL PAGE
5506 ; 000004 DTEMAX= 4 ; MAX OF 4 DTE'S ON A PDP11
5507 ;
5508 ;
5509 ; HERE TO DUMP -11 MEMORY AND/OR BOOT -11 THROUGH THE DTE20
5510 ;
5511 ; 002 .IF DF R$S$G$V
5512 ; ;173622' DTEDMP:
5513 ; .IFF ; .IF DF R$S$G$V
5514 ; BUTON4:
5515 ; 001 .ENDC ; .IF DF R$S$G$V
5516 ;
5517 ; REGISTERS SAVED-- LOOK FOR THE DTE20 WHICH PUSHED THE BUTTON
5518 ;
5519 ; THE DTE WHICH PUSHED THE BUTTON SHOULD HAVE THE DOORBELL
5520 ; RINGING AND HAVE THE VALUE 1365 (OCTAL) IN IT'S
5521 ; T0 -10 BYTE COUNT T010BC.
5522 ;
5523 ; NXM (TIME-OUT) TRAP IS USED TO SKIP NON-EXISTANT DTE20'S.
5524 ;
5525 ; 002 .IF DF S$S$DTE
5526 ; 003 .IF LT <S$S$DTE-400>
5527 010622 005005 ;173622' 005005 CLR R5 ;POINT TO LOCATION 0

```

L12

5528	010624	012500	;173624'	012500	MOV	(R5)+,R0	:SAVE LOCATION 0 (FTL STK VIOLATION STACK)
5529	010626	012501	;173626'	012501	MOV	(R5)+,R1	: AND LOCATION 2 (. . .)

```

5530 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 11-1
5531
5532 .IFF ; .IF LT <SSSDTE-400>
5533 :
5534 002 .ENDC ; .IF LT <SSSDTE-400> ;SET TO LOCATION 4
5535 .IFF ; .IF DF SSSDTE
5536 :
5537 :
5538 :
5539 :
5540 :
5541 010630 011502 ;173630' 011502 ; .IFTF ; .IF DF SSSDTE
5542 010632 012725 ;173632' 012725 MOV (R5),R2 ;SAVE 4 IN R2
5543 010634 173646 ;173634 173646 MOV #21$, (R5)+ ;SET NXM TRAP ADDRESS IN 4
5544 010636 011503 ;173636' 011503 MOV (R5),R3 ;SAVE 6 IN R3
5545 010640 005015 ;173640' 005015 CLR (R5) ;SET PS FOR TRAP
5546 :
5547 :
5548 :
5549 :
5550 010642 012704 ;173642' 012704 20$: MOV #D$ST20+DLYCNT-DTESIZ,R4 ;POINT TO DTE # -1'S DELAY COUNT REGIST
5551 010644 174340 ;173644 174340 ; (WILL BUMP TO # 0)
5552 :
5553 :
5554 :
5555 :
5556 ;173646'
5557 :
5558 :
5559 010646 012706 ;173646' 003 012706 21$: .IFT ; .IF DF SSSDTE
5560 010650 000140 ;173650 000140 .IF LT <SSSDTE-400> MOV #SSSDTE+10,SP ;SET STACK TO SAVE AREA, WITH ROOM FOR FTL STK T
5561 :
5562 :
5563 002 .ENDC ; .IF LT <SSSDTE-400> ;SET STACK TO REGISTER SAVE AREA
5564 :
5565 :
5566 :
5567 :
5568 010652 062704 ;173652' 062704 22$: .IFF ; .IF LT <SSSDTE-400>
5569 010654 000040 ;173654 000040 MOV #SSSDTE+4,SP ;SET STACK TO REGISTER SAVE AREA
5570 010656 105704 ;173656' 105704 .IFF ; .IF DF SSSDTE
5571 :
5572 :
5573 010660 100770 ;173660' 100770 .IFTF ; .IF DF SSSDTE
5574 :
5575 010662 032764 ;173662' 032764 22$: ADD #DTESIZ,R4 ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS
5576 010664 004000 ;173664 004000 TSTB R4 ;IS THIS THE END OF THE DTE'S?
5577 010666 000034 ;173666 000034 ; NOTE THAT THE LAST DTE IS AT 774540
5578 010670 001770 ;173670' 001770 BMI 20$ ; AND THAT NOW R4= 774600 IF END
5579 010672 026417 ;173672' 026417 BIT #T011DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?
5580 010674 000014 ;173674 000014 BEQ 22$ ;YES-- START ALL OVER, UNTIL A DTE
5581 :
5582 :
5583 :
; *** THIS NEXT INSTRUCTION BETTER BE A 1365 !!! ***
; IN IT'S TO -10 BYTE COUNT?

```

5584				:				
5585	010676	001365	;173676'	:	001365	BNE	225	;NO-- TRY ANOTHER DTE
5586				:				
5587				:				
5588				:				
5589				:				
5590				:				
5591	010700	010315	;173700'	:	010315	MOV	R3,(R5)	;RESTORE LOCATION 6
5592	010702	010245	;173702'	:	010245	MOV	R2,-(R5)	; 4

```

5593 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 11-2
5594
5595 .IFF ; .IF DF SSSDTE
5596     MOV R1,-(SP) ; AND 1776
5597     MOV R0,-(SP) ; . . . 1774
5598
5599 .IFT ; .IF DF SSSDTE
5600 010704 010145 ;173704' 003 .IF LT <SSSDTE-400> 010145 MOV R1,-(R5) ;RESTORE 2
5601 010706 010045 ;173706' 010045 MOV R0,-(R5) ; AND 0
5602
5603 ; SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
5604 ; IN LOCATIONS 130-156
5605
5606 010710 012700 ;173710' 012700 MOV #SSSDTE,R0 ;POINT TO SAVE AREA
5607 010712 000130 ;173712' 000130
5608 ;173714' 29$:
5609 010714 012420 ;173714' 012420 MOV (R4)+(R0)+ ;SAVE A REGISTER
5610 010716 022700 ;173716' 022700 CMP #T011DT-DLYCNT+SSSDTE,R0 ;FINISHED?
5611 010720 000156 ;173720' 000156
5612 .IFF ; .IF LT <SSSDTE-400>
5613     CMP -(SP),-(SP) ;BACK DOWN TO START OF SAVE AREA
5614
5615 29$:
5616     MOV (R4)+(SP)+ ;SAVE A REGISTER
5617     CMP #T011DT-DLYCNT+SSSDTE,SP ;FINISHED?
5618 010722 103374 ;173722' 002 .ENDC ; .IF LT <SSSDTE-400>
5619     BHIS 29$ ;NO-- SAVE SOME MORE
5620
5621 ; R4= T011DT+2
5622 ; SET R1= STATUS REGISTER
5623 ; R4= DIAG2 REGISTER
5624
5625 ; DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
5626 ; LOADED FLAG
5627
5628     ADDX DIAG2-<T011DT+2>,R4 ;R4 POINTS TO DIAG2 REGISTER
5629 010724 005724 ;173724' 005724 TST (R4)+
5630 .IFF ; .IF DF SSSDTE
5631     ADDX DIAG2-DLYCNT,R4 ;POINT R4 TO DIAG2 REGISTER
5632 .ENDC ; .IF DF SSSDTE
5633 010726 010401 ;173726' 001 010401 MOV R4,R1 ; SO DOES R1
5634 010730 012700 ;173730' 012700 MOV #DRESET,R0 ;SETUP R0 FOR 'DIAGNOSTIC RESET'
5635 010732 000100 ;173732' 000100
5636 010734 010021 ;173734' 010021 MOV R0,(R1)+ ;R1 POINTS TO STATUS REGISTER
5637
5638 ; REGISTERS:
5639     R0 -- DRESET (DIAGNOSTIC RESET FUNCTION)
5640     R1 -- STAT (STATUS REGISTER)
5641     R4 -- DIAG2 (DIAGNOSTIC REGISTER #2, WHERE DRESET IS)
5642
5643 ; THE -10 WILL NOW START READING -11 MEMORY, AS SOON AS WE SET
5644 ; THE TO -10 ADDRESS. WHEN FINISHED, THE -10 WILL RING OUR DOORBELL.
5645
5646 010736 005061 ;173736' 005061 CLR DLYCNT-STAT(R1) ;SET DTE20 FOR MAXIMUM DELAY (ZERO)

```

5647	010740	177744	:173740	177744				
5648	010742	005061	::173742'		005061	CLR	T010AD-STAT(R1)	;START DUMPING -11 MEMORY TO -10
5649	010744	177764	::173744	177764				
5650								; STARTING AT LOCATION 0
5651			:173746'					
5652	010746	032711	::173746'		032711	30\$:	BIT	#T011DB,(R1) ;IS DOORBELL RINGING (TRANSFER COMPLETE)?
5653	010750	004000	::173750	004000				
5654	010752	001775	::173752'		001775	BEG	30\$;NO-- WAIT FOR DOORBELL

```

5655 ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 11-3
5656
5657 010754 010014 ;173754' 010014 MOV R0,(R4) ;YES-- CLEAR DOORBELL AND ERROR FLAGS
5658
5659 ; NOW THE -10 WILL GIVE US A 256 WORD BOOTSTRAP TO BE READ
5660 ; INTO -11 MEMORY STARTING AT LOCATION 0. WHEN FINISHED,
5661 ; WE WILL START EXECUTION OF THE LOADED CODE AT LOCATION 0.
5662
5663 010756 005061 ;173756' 005061 CLR T011AD-STAT(R1) ;START INPUT TO LOCATION 0
5664 010760 177766 ;173760 177766
5665 010762 012761 ;173762' 012761 MOV #IFLOP!<<-256.>&BCOUNT>,T011BC-STAT(R1) ;256 WORDS, INTERRUPT
5666 010764 107400 ;173764 107400
5667 010766 177762 ;173766 177762 ; -10 WHEN DONE
5668 ;
5669 ;173770' 40$:
5670 010770 105711 ;173770' 105711 TSTB (R1) ;TRANSFER COMPLETE?
5671 010772 100376 ;173772' 100376 BPL 40$ ;NO-- WAIT SOME MORE
5672 010774 005007 ;173774' 005007 CLR PC ;GO TO LOADED CODE, STARTING AT
5673 ; ; LOCATION 0
5674 ; 000 .ENDC ; .IF DF DS$T20

```


MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 159
DZBMD.P11 ROM CONTENTS TABLES

```

5675 ;BM873J -KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 20-JAN-77 11:51 PAGE 12
5676 ;
5677 ; .SBTTL FILL TO END OF ROM
5678 ;
5679 ;
5680 010776 END.YJ:
5681 ;:173776'
5682 ;:173776' 000002 FILL TO 1000
5683 010776 000000 ;:173776' .PRINT <1000>-<.-BM873> ;FREE BYTES AT 1000
5684 ;:173777' ;:000 .BYTE 0
5685 ;:000 .BYTE 0
5686 ;
5687 ;:174000' BM873E:
5688 ; 173000' .END START

```

5689	011000	000177	MAP.Y.: .BLKW	127.
5690	011376	000001	END.Y.: .BLKW	1
5691	011400	000177	MAP.YX: .BLKW	127.
5692	011776	000001	END.YX: .BLKW	1

```

5693
5694
5695
5696
5697 012000
5698 012000 012706 001100
5699 012004 005026
5700 012006 022706 001136
5701 012012 001374
5702 012014 012706 001100
5703 012020 012737 020534 000020
5704 012026 012737 000340 000022
5705 012034 012737 020630 000030
5706 012042 012737 000340 000032
5707 012050 012737 021160 000034
5708 012056 012737 000340 000036
5709 012064 012767 012064 167014
5710 012072 005067 001372
5711 012076 005067 001356
5712 012102 012706 001100
5713 012106 005067 167104
5714 012112 005037 000000
5715 012116 012767 012102 005546
5716 012124 012737 000006 000004
5717 012132 005037 000006
5718 012136 005067 005052
5719 012142 005737 000042
5720 012146 001002
5721 012150 000167 000616
5722 012154 013746 000004
5723 012160 012737 013474 000004
5724 012166 005737 173000
5725 012172 000240
5726 012174 012637 000004
5727 012200 026737 167174 173000
5728 012206 001034
5729 012210 013746 000004
5730 012214 012737 012236 000004
5731 012222 005737 173400
5732 012226 000240
5733 012230 012637 000004
5734 012234 000421
5735 012236 022626
5736 012240 012637 000004
5737 012244 012767 001400 001206
5738 012252 012767 001776 001202
5739 012260 012767 173376 001204
5740 012266 012767 000101 005520
5741 012274 000167 001416
5742 012300
5743 012300 026737 167474 173000
5744 012306 001016
5745 012310 012767 002000 001142
5746 012316 012767 002776 001136

```

```

*****
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    #.SLPADR       ;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

```

H13

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 162
 DZBMD.P11 ROM CONTENTS TABLES

5747	012324	012767	173776	001140		MOV	#173776, LASTA	: LAST ROM ADDR
5748	012332	012767	000102	005454		MOV	#000102, Verson	: SET ROM TYPE
5749	012340	000167	001352			JMP	PRG1	: START TEST 1
5750	012344				69\$:			
5751	012344	026737	170430	173000		CMP	MAP.YC, a#173000	: DOES 1ST WORD COMPARE?
5752	012352	001036				BNE	74\$: CHECK NEXT MAP
5753	012354	013746	000004			MOV	a#4, -(SP)	: SAVE LOC 4
5754	012360	012737	012404	000004		MOV	#76\$, a#4	: SET FOR TIMEOUT
5755	012366	026737	171006	173400		CMP	MAP.YC+400, a#173400	: IS IT YC?
5756	012374	001004				BNE	77\$: BR IF NOT YC
5757	012376	012637	000004			MOV	(SP)+, a#4	: RESTORE LOC 4
5758	012402	000404				BR	78\$: YES IT IS A YC
5759	012404	022626			76\$:	CMP	(SP)+, (SP)+	: ADJUST STACK
5760	012406	012637	000004		77\$:	MOV	(SP)+, a#4	: RESTORE LOC 4
5761	012412	000416				BR	74\$: CHECK NEXT MAP
5762	012414				78\$:			
5763	012414	012767	003000	001036		MOV	#MAP.YC, TABLE	: 1ST MAP ADDR
5764	012422	012767	003776	001032		MOV	#END.YC, ALLEND	: LAST MAP ADDR
5765	012430	012767	173776	001034		MOV	#173776, LASTA	: LAST ROM ADDR
5766	012436	012767	000103	005350		MOV	#000103, Verson	: SET ROM TYPE
5767	012444	000167	001246			JMP	PRG1	: START TEST 1
5768	012450				74\$:			
5769	012450	026737	171324	173000		CMP	MAP.YD, a#173000	: DOES 1ST WORD COMPARE?
5770	012456	001016				BNE	79\$: CHECK NEXT MAP
5771	012460	012767	004000	000772		MOV	#MAP.YD, TABLE	: 1ST MAP ADDR
5772	012466	012767	004776	000766		MOV	#END.YD, ALLEND	: LAST MAP ADDR
5773	012474	012767	173776	000770		MOV	#173776, LASTA	: LAST ROM ADDR
5774	012502	012767	000104	005304		MOV	#000104, Verson	: SET ROM TYPE
5775	012510	000167	001202			JMP	PRG1	: START TEST 1
5776	012514				79\$:			
5777	012514	026737	172260	173000		CMP	MAP.YF, a#173000	: DOES 1ST WORD COMPARE?
5778	012522	001022				BNE	84\$: CHECK NEXT MAP
5779	012524	026737	172550	173300		CMP	MAP.YF+300, a#173300	: IS IT YF?
5780	012532	001016				BNE	84\$: CHECK NEXT MAP
5781	012534	012767	005000	000716		MOV	#MAP.YF, TABLE	: 1ST MAP ADDR
5782	012542	012767	005776	000712		MOV	#END.YF, ALLEND	: LAST MAP ADDR
5783	012550	012767	173776	000714		MOV	#173776, LASTA	: LAST ROM ADDR
5784	012556	012767	000106	005230		MOV	#000106, Verson	: SET ROM TYPE
5785	012564	000167	001126			JMP	PRG1	: START TEST 1
5786	012570				84\$:			
5787	012570	026737	173204	173000		CMP	MAP.YG, a#173000	: DOES 1ST WORD COMPARE?
5788	012576	001016				BNE	89\$: CHECK NEXT MAP
5789	012600	012767	006000	000652		MOV	#MAP.YG, TABLE	: 1ST MAP ADDR
5790	012606	012767	006776	000646		MOV	#END.YG, ALLEND	: LAST MAP ADDR
5791	012614	012767	173776	000650		MOV	#173776, LASTA	: LAST ROM ADDR
5792	012622	012767	000107	005164		MOV	#000107, Verson	: SET ROM TYPE
5793	012630	000167	001062			JMP	PRG1	: START TEST 1
5794	012634				89\$:			
5795	012634	026737	174140	173000		CMP	MAP.YH, a#173000	: DOES 1ST WORD COMPARE?
5796	012642	001022				BNE	94\$: CHECK NEXT MAP
5797	012644	026737	174430	173300		CMP	MAP.YH+300, a#173300	: IS IT YH?
5798	012652	001016				BNE	94\$: CHECK NEXT MAP
5799	012654	012767	007000	000576		MOV	#MAP.YH, TABLE	: 1ST MAP ADDR
5800	012662	012767	007776	000572		MOV	#END.YH, ALLEND	: LAST MAP ADDR

5801	012670	012767	173776	000574		MOV	#173776, LASTA	; LAST ROM ADDR
5802	012676	012767	000110	005110		MOV	#000110, Verson	; SET ROM TYPE
5803	012704	000167	001006			JMP	PRG1	; START TEST 1
5804	012710				94\$:			
5805	012710	026737	175064	173000		CMP	MAP.YJ, @#173000	; DOES 1ST WORD COMPARE?
5806	012716	001022				BNE	99\$; CHECK NEXT MAP
5807	012720	026737	175354	173300		CMP	MAP.YJ+300, @#173300	; IS IT YJ?
5808	012726	001016				BNE	99\$; CHECK NEXT MAP
5809	012730	012767	010000	000522		MOV	#MAP.YJ, TABLE	; 1ST MAP ADDR
5810	012736	012767	010776	000516		MOV	#END.YJ, ALLEND	; LAST MAP ADDR
5811	012744	012767	173776	000520		MOV	#173776, LASTA	; LAST ROM ADDR
5812	012752	012767	000112	005034		MOV	#000112, Verson	; SET ROM TYPE
5813	012760	000167	000732			JMP	PRG1	; START TEST 1
5814	012764				99\$:			
5815	012764	104400	013574			TYPE	, NMATCH	; NOT BM873YA OR B OR C OR D OR F OR G OR H OR J
5816	012770	000000				HALT		
5817	012772	005767	000472		CONT:	TST	INITFG	; IS THIS THE FIRST TIME START UP?
5818	012776	001173				BNE	3\$; BR IF NOT FIRST TIME HERE.
5819	013000	005167	000464			COM	INITFG	; SET THE FLAG
5820	013004	104400	014310		2\$:	TYPE	, BM873X	; TYPE THE QUESTION.
5821	013010	104412				ROL IN		
5822	013012	012602				MOV	(SP)+, R2	
5823	013014	011202				MOV	(R2), R2	; PLACE CHARACTER INTO R2.
5824	013016	022702	000052			CMP	#52, R2	; WAS * HIT??
5825	013022	001011				BNE	64\$; BR IF NO
5826	013024	012767	011000	000426		MOV	#MAP.Y., TABLE	; SET FOR START OF TABLE
5827	013032	012767	011376	000422		MOV	#END.Y., ALLEND	; SET END OF TABLE
5828	013040	012767	173376	000424		MOV	#173376, LASTA	; SET LAST ROM ADDR
5829	013046				64\$:			
5830	013046	022702	000101			CMP	#101, R2	; WAS A HIT??
5831	013052	001011				BNE	65\$; BR IF NO
5832	013054	012767	001400	000376		MOV	#MAP.YA, TABLE	; SET FOR START OF TABLE
5833	013062	012767	001776	000372		MOV	#END.YA, ALLEND	; SET END OF TABLE
5834	013070	012767	173376	000374		MOV	#173376, LASTA	; SET LAST ROM ADDR
5835	013076				65\$:			
5836	013076	022702	000102			CMP	#102, R2	; WAS B HIT??
5837	013102	001011				BNE	66\$; BR IF NO
5838	013104	012767	002000	000346		MOV	#MAP.YB, TABLE	; SET FOR START OF TABLE
5839	013112	012767	002776	000342		MOV	#END.YB, ALLEND	; SET END OF TABLE
5840	013120	012767	173776	000344		MOV	#173776, LASTA	; SET LAST ROM ADDR
5841	013126				66\$:			
5842	013126	022702	000103			CMP	#103, R2	; WAS C HIT??
5843	013132	001011				BNE	67\$; BR IF NO
5844	013134	012767	003000	000316		MOV	#MAP.YC, TABLE	; SET FOR START OF TABLE
5845	013142	012767	003776	000312		MOV	#END.YC, ALLEND	; SET END OF TABLE
5846	013150	012767	173776	000314		MOV	#173776, LASTA	; SET LAST ROM ADDR
5847	013156				67\$:			
5848	013156	022702	000104			CMP	#104, R2	; WAS D HIT??
5849	013162	001011				BNE	68\$; BR IF NO
5850	013164	012767	004000	000266		MOV	#MAP.YD, TABLE	; SET FOR START OF TABLE
5851	013172	012767	004776	000262		MOV	#END.YD, ALLEND	; SET END OF TABLE
5852	013200	012767	173776	000264		MOV	#173776, LASTA	; SET LAST ROM ADDR
5853	013206				68\$:			
5854	013206	022702	000106			CMP	#106, R2	; WAS F HIT??

5855	013212	001011			BNE	69\$;BR IF NO
5856	013214	012767	005000	000236	MOV	#MAP.YF, TABLE		;SET FOR START OF TABLE
5857	013222	012767	005776	000232	MOV	#END.YF, ALLEND		;SET END OF TABLE
5858	013230	012767	173776	000234	MOV	#173776, LASTA		;SET LAST ROM ADDR
5859	013236							
5860	013236	022702	000107		69\$:	CMP	#107, R2	; WAS G HIT??
5861	013242	001011				BNE	70\$; BR IF NO
5862	013244	012767	006000	000206	MOV	#MAP.YG, TABLE		;SET FOR START OF TABLE
5863	013252	012767	006776	000202	MOV	#END.YG, ALLEND		;SET END OF TABLE
5864	013260	012767	173776	000204	MOV	#173776, LASTA		;SET LAST ROM ADDR
5865	013266				70\$:			
5866	013266	022702	000110			CMP	#110, R2	; WAS H HIT??
5867	013272	001011				BNE	71\$; BR IF NO
5868	013274	012767	007000	000156	MOV	#MAP.YH, TABLE		;SET FOR START OF TABLE
5869	013302	012767	007776	000152	MOV	#END.YH, ALLEND		;SET END OF TABLE
5870	013310	012767	173776	000154	MOV	#173776, LASTA		;SET LAST ROM ADDR
5871	013316				71\$:			
5872	013316	022702	000112			CMP	#112, R2	; WAS J HIT??
5873	013322	001011				BNE	72\$; BR IF NO
5874	013324	012767	010000	000126	MOV	#MAP.YJ, TABLE		;SET FOR START OF TABLE
5875	013332	012767	010776	000122	MOV	#END.YJ, ALLEND		;SET END OF TABLE
5876	013340	012767	173776	000124	MOV	#173776, LASTA		;SET LAST ROM ADDR
5877	013346				72\$:			
5878	013346	010267	004442			MOV	R2, VERSION	; STORE VERSION TYPE..
5879	013352	005767	000102			TST	TABLE	; HAS A MAP BEEN SELECTED?
5880	013356	001003				BNE	3\$; BR IF OK...
5881	013360	104400	014365			TYPE	, BM.ERR	; TYPE ERROR
5882	013364	000607				BR	2\$; GO AND GET CORRECT MAP.
5883	013366	104400	017336		3\$:	TYPE	, MSG3	; TYPE MESSAGE FOR TEST NUMBER
5884	013372	104412			X.X.:	RDLIN		
5885	013374	012602				MOV	(SP)+, R2	
5886	013376	011203				MOV	(R2), R3	; MOV THE CHAR TO R3
5887	013400	022703	000061		2\$:	CMP	#61, R3	; WAS 1 HIT??
5888	013404	001002				BNE	4\$; BR IF NO
5889	013406	000167	000304			JMP	PRG1	; GOTO PRG 1
5890	013412	022703	000062		4\$:	CMP	#62, R3	; WAS 2 HIT??
5891	013416	001002				BNE	5\$; BR IF NO
5892	013420	000167	001020			JMP	PRG2	; GOTO PRG 2
5893	013424	022703	000063		5\$:	CMP	#63, R3	; WAS 3 HIT??
5894	013430	001002				BNE	6\$; BR IF NO
5895	013432	000167	001746			JMP	PRG3	; GOTO PRG3
5896	013436	022703	000064		6\$:	CMP	#64, R3	; WAS 4 HIT??
5897	013442	001002				BNE	3\$; BR IF NO
5898	013444	000167	002740			JMP	PRG4	; GOTO PRG 4
5899	013450	104400	017542		3\$:	TYPE	, M.QM	; NEITHER 1 OR 2 OR 3 OR 4 WAS HIT
5900	013454	000167	176320			JMP	RESTRT	; TYPE "??" GO TO THE BEGINING.
5901	013460	000000			TABLE:	0		
5902	013462	000000			ALLEND:	0		
5903	013464	011400			EXTMAP:	MAP.YX		
5904	013466	011776			EXTEND:	END.YX		
5905	013470	000000			INITFG:	0		
5906	013472	000000			LASTA:	0		
5907	013474	104400	013504		NOROM:	TYPE	, NOROMS	; TYPE CAN'T FIND A RESPONSE
5908	013500	000000			HALT			; NO LOADER INSTALLED?

K13

MAINDEC-11-DZBMD-J MACY11 27(663) 2-MAY-77 11:46 PAGE 165
DZBMD.P11 ROM CONTENTS TABLES

5909	013502	000776					
5910	013504	005015	051124	050101	NOROMS:	.ASCII	<15><12>/TRAP TO 4 ON 1ST READ OF 173000/
	013545	015	044412	020123		.ASCIIZ	<15><12>/IS LOADER INSTALLED?/
	013574	005015	040503	023516	NMATCH:	.ASCII	<15><12>/CAN'T IDENTIFY LOADER AS YA, YB, YC, YD, YF, YG, YH OR YJ AFTER/
	013667	015	041412	050115		.ASCIIZ	<15><12>/CMP WITH LOC 173000/
	013716				.EVEN		

```

5911 :PROGRAM 1
5912 :THE PURPOSE OF PROGRAM 1 IS TO READ THE ROM AND
5913 :VERIFY THAT THE DATA IS CORRECT. ALL ADDRESSES
5914 :ARE READ, EXCEPT THE TRAP VECTOR, FIVE TIMES.
5915
5916 :THE SECOND PART OF THIS TEST VERIFIES THAT TRYING
5917 :TO WRITE THE ROM RESULTS IN A TIME OUT TRAP.
5918 :ALL ADDRESS ARE WRITTEN WITH A -1
5919 :AND ARE EXPECTED TO TRAP.
5920
5921 013716 012767 013716 003746 PRG1: MOV #PRG1,PRG.NO ;SET FOR PWR FAIL
5922 013724 012767 000500 165266 MOV #500,COUNT ;DO THIS TEST 500(8) TIMES.
5923 013732 012737 017674 000004 PRG.1: MOV #NO.TRAP,2#4 ;SET FOR UNEXPECTED TRAP.
5924 013740 012700 173000 MOV #173000,R0 ;SET BEGGING ADDRESS
5925 013744 012767 013770 165134 MOV 2#,$LPADR ;IF SW14=1; GOTO 2$ WHEN SCOPE IS HIT
5926 013752 016704 177502 MOV TABLE,R4 ;SET START OF MAP
5927 013756 016767 177510 000322 MOV LAST,LAST ;SET LAST ADDRESS
5928 013764 012703 000005 1$: MOV #5,R3 ;DO EACH ADDRESS 5 TIMES.
5929 013770 022700 173024 2$: CMP #173024,R0 ;DON'T DO THE VECTOR ADD.
5930 013774 001001 BNE 20$ ;BR IF NOT THE VECTOR ADD.
5931 013776 022024 CMP (R0)+,(R4)+ ;UPDATE TO NEXT ADDRESS
5932 014000 022700 173224 20$: CMP #173224,R0 ;DON'T DO THE TRAP VECTORS
5933 014004 001001 BNE 21$ ;NO THIS ISN'T A TRAP VECTOR.
5934 014006 022024 CMP (R0)+,(R4)+ ;UPDATE THE POINTERS..
5935 014010 010467 165110 21$: MOV R4,$GDDAT ;READ THE ADDRESS
5936 014014 010067 165106 MOV R0,$BDDAT ;READ THE SOFTWARE ADDRESS
5937 014020 011067 165202 MOV (R0),TEMP4
5938 014024 011467 165174 MOV (R4),TEMP3
5939 014030 026767 165170 165170 CMP TEMP3,TEMP4
5940 014036 001401 BEQ 22$ ;BR IF GOOD
5941 014040 104001 ERROR 1 ;INCORRECT COMPARISON.
5942 014042 032767 004000 163520 22$: BIT #BIT11,SWR ;QUICK PASS.?
5943 014050 001002 BNE 23$ ;BR IF YES
5944 014052 005303 DEC R3 ;HAS THAT ADD BEEN READ 5 TIMES?
5945 014054 001345 BNE 2$ ;BR IF NOT 5 TIMES
5946
5947 014056 026700 000224 23$: CMP LAST,R0 ;WAS LAST ADDRESS CHECKED?
5948 014062 001403 BEQ 10$ ;BR IF YES
5949 014064 000004 SCOPE ;LOCK ON THIS ADDRESS IF SW14=1
5950 014066 022024 CMP (R0)+,(R4)+ ;UPDATE THE POINTERS.
5951 014070 000735 BR 1$ ;CONTINUE THE TEST.
5952
5953 014072 032767 000001 163470 10$: BIT #BIT0,SWR ;EXTENDED WORD TO BE CHECKED?
5954 014100 001413 BEQ 3$ ;BR IF NO CHECKING.
5955 014102 022767 173776 000176 CMP #173776,LAST ;IS ALL THE TEST DONE?
5956 014110 001407 BEQ 3$ ;BR IF YES.
5957 014112 012767 173776 000166 MOV #173776,LAST ;SET LAST ADDRESS.
5958 014120 016704 177340 MOV EXTMAP,R4 ;SET EXTENDED MAP.
5959 014124 005720 TST (R0)+ ;POP POINTER
5960 014126 000716 BR 1$ ;GO DO THE TEST.
    
```



```

5961 ;TEST THAT WRITTING ROM RESULTS IN A TIME OUT
5962 ;TRAP.
5963
5964 014130 012767 014154 164750 3$: MOV #5$,SLPADR ;IF SW14=1 GOTO 5$ WHEN SCOPE IS HIT
5965 014136 012700 173000 MOV #173000,RO ;SET RO WITH BASE ADDRESS OF ROM
5966 014142 012737 014210 000004 MOV #6$,R#4 ;SET FOR TIME OUT TRAP
5967 014150 012703 000005 4$: MOV #5,R3 ;DO EACH ADD 5 TIMES
5968 014154 022700 173024 5$: CMP #173024,RO ;CHECK FOR A TRAP VECTOR
5969 014160 001001 BNE 24$ ;BR IF NOT VECTOR
5970 014162 005720 TST (RO)+ ;UPDATE THE REGISTER POINTER
5971 014164 022700 173224 24$: CMP #173224,RO ;CHECK FOR THE OTHER VECTOR
5972 014170 001001 BNE 25$ ;BR IF NOT THE VECTOR
5973 014172 005720 TST (RO)+ ;UPDATE THE POINTER
5974 014174 012710 177777 25$: MOV #-1,(RO) ;WRITE ROM WITH A -1
5975 014200 000240 NOP ;WAIT ONE INSTR. TIME
5976 014202 010067 165020 MOV RO,TEMP4
5977 014206 104002 ERROR 2 ;WRITING ROM DIDN'T TIME OUT.
5978 014210 012706 001100 6$: MOV #STACK,SP ;RESTORE STACK
5979 014214 032767 004000 163346 BIT #BIT11,SWR ;QUICK PASS?
5980 014222 001002 BNE 30$
5981 014224 005303 DEC R3 ;DO EACH ADD 5 TIMES
5982 014226 001352 BNE 5$ ;NOT DONE WITH THIS ONE YET.
5983
5984 014230 032767 000001 163332 30$: BIT #BIT0,SWR ;EXTENDED 128. WORDS TO BE CHECKED?
5985 014236 001404 BEQ 31$ ;BR IF NO
5986 014240 022700 173776 CMP #173776,RO ;HAVE ALL 256. WORDS BEEN CHECKED?
5987 014244 001407 BEQ 7$ ;BR IF ALL DONE
5988 014246 000403 BR 32$ ;KEEP GOING
5989 014250 026700 177216 31$: CMP LASTA,RO ;ALL DONE??
5990 014254 001403 BEQ 7$ ;HAVE ALL 128. WORDS DONE?
5991 014256 000004 32$: SCOPE ;CHECK SW14 FOR FREEZE!!
5992 014260 005720 TST (RO)+ ;UPDATE TO NEXT ADDRESS
5993 014262 000732 BR 4$ ;GO DO IT AGAIN
5994 014264 005367 164730 7$: DEC ICOUNT ;ITERATION COUNT DONE?
5995 014270 001004 BNE 8$ ;BR IF NOT DONE.
5996 014272 004767 003436 JSR PC EOP ;TYPE END MESSAGE
5997 014276 000167 177414 JMP PRG1 ;GO DO IT AGAIN.
5998 014302 000167 177424 8$: JMP PRG.1 ;GO RESTART.
5999 014306 000000 LAST: 0
6000
6001 014310 005015 040515 047111 BM873X: .ASCII <15><12>/MAINDEC-11-DZBMDJ/
014333 015 042012 053105 .ASCII <15><12>/DEVICE VERSION/
014353 015 041012 034115 .ASCIZ <15><12>/BM873-Y/
014365 015 025012 040454 BM.ERR: .ASCIZ <15><12>/*,A,B,C,D,F,G,H,J ONLY./
014417 040 020040 042526 VERS: .ASCIZ / VERSION: BM873-Y/
014444 .EVEN
    
```

```

6002 ;PROGRAM 2
6003 ;BLIND READ FROM ROM.
6004 ;THIS PROGRAM WILL DUMP THE CONTENTS OF THE ROM OUT
6005 ;PERFORMING NO CHECKING AT ALL.
6006 ;PLEASE NOTE: NO CHECKING IS DONE.
6007
6008 014444 012767 014444 003220 PRG2: MOV #PRG2,PRG,NO ;SET FOR POWER FAIL
6009 014452 012737 017674 000004 MOV #NO.TRAP,2#4 ;SET FOR UNEXPECTED TRAP TO 4
6010 014460 016767 177006 177620 MOV LASTA,LAST
6011 014466 062767 000002 177612 ADD #2,LAST
6012 014474 012700 173000 21$: MOV #173000,RO ;SET RO WITH THE STARTING ROM ADD.
6013 014500 016703 176754 MOV TABLE,R3 ;SET POINTER.
6014 014504 104400 015024 TYPE ,DH.2 ;TYPE MESSAGE
6015 014510 104400 015106 TYPE ,DH.2B ;TYPE THE HEADER
6016 014514 012767 000007 164500 1$: MOV #7,TEMPS ;SET COUNTER
6017 014522 011001 MOV (RO),R1 ;READ THE ROM
6018 014524 010067 164474 MOV RO,TEMP3 ;STORE RO
6019 014530 010167 164472 MOV R1,TEMP4 ;STORE R1
6020 014534 022767 011000 176716 CMP #MAP.Y.,TABLE ;IF BM873.Y* SELECTED; FILL TABLE
6021 014542 001001 BNE 22$ ;BR IF NOT BM873.Y*
6022 014544 011023 MOV (RO),(R3)+ ;FILL THE TABLE..
6023 014546 005720 22$: TST (RO)+ ;POP THE POINTER
6024 014550 104400 017556 TYPE ,MCRLF
6025
6026 014554 016746 164444 MOV TEMP3,-(SP)
6027 014560 104402 TYPOC
6028 014562 104400 017547 TYPE ,MSPACE ;TYPE THREE SPACES.
6029
6030
6031 014566 016746 164434 MOV TEMP4,-(SP)
6032 014572 104402 TYPOC
6033 014574 011001 7$: MOV (RO),R1 ;STORE ROM DATA
6034 014576 010067 164422 MOV RO,TEMP3 ;STORE ROM ADDRESS
6035 014602 010167 164420 MOV R1,TEMP4 ;PREPARE DATA FOR TYPE OUT
6036 014606 022767 011000 176644 CMP #MAP.Y.,TABLE ;IS BM873.Y* SELECTED?
6037 014614 001001 BNE 23$ ;BR IF NO..
6038 014616 011023 MOV (RO),(R3)+ ;FILL THE DATA TABLE
6039 014620 005720 23$: TST (RO)+ ;POP THE POINTER
6040
6041 014622 104400 017547 TYPE ,MSPACE
6042
6043 014626 016746 164374 MOV TEMP4,-(SP)
6044 014632 104402 TYPOC
6045
6046 014634 026700 177446 CMP LAST,RO ;HAS THE HIGHEST LIMIT BEEN HIT?
6047 014640 001404 BEQ 2$ ;BR IF ALL DONE.
6048 014642 005367 164354 DEC TEMPS ;DECREASE COUNTER
6049 014646 001352 BNE 7$ ;BR IF NOT 0; KEEP GOING
6050 014650 000721 BR 1$ ;GO TYPE ADDRESS NOW
6051
6052 014652 032767 000001 162710 2$: BIT #BIT0,SWR ;IS THE EXTENDED 128. WORDS TO BE CHECKED??
6053 014660 001455 BEQ 3$ ;BR IF NO.
6054 014662 012700 173400 MOV #173400,RO ;RESET POINTER OF ROM
6055 014666 016703 176572 MOV EXTMAP,R3 ;SET SOFTWARE MAP POINTER
    
```

6056	014672	104400	015220		TYPE	,DH.2A	;TYPE NEW HEADER
6057	014676	104400	015106		TYPE	,DH.2B	;TYPE ADDRESS AND +XX
6058	014702	012767	000007	164312	6S:	MOV #7,TEMP5	;SET TYPE OUT COUNTER
6059	014710	011001			MOV	(R0),R1	;READ THE ROM
6060	014712	010067	164306		MOV	R0,TEMP3	;STORE R0
6061	014716	010167	164304		MOV	R1,TEMP4	;STORE R1
6062	014722	012023			MOV	(R0)+,(R3)+	;STORE THE DATA IN SOFTWARE MAP
6063	014724	104400	017556		TYPE	MCRLF	
6064	014730	016746	164270		MOV	TEMP3,-(SP)	
6065	014734	104402			TYPOC		
6066							
6067	014736	104400	017547		TYPE	MSPACE	
6068	014742	016746	164260		MOV	TEMP4,-(SP)	
6069	014746	104402			TYPOC		
6070							
6071	014750	011001			8S:	MOV (R0),R1	;SAVE THE ROM DATA
6072	014752	010067	164246		MOV	R0,TEMP3	;SAVE THE ROM ADDRESS
6073	014756	010167	164244		MOV	R1,TEMP4	;SET DATA FOR TYPE OUT
6074							
6075	014762	104400	017547		TYPE	,MSPACE	
6076							
6077	014766	016746	164234		MOV	TEMP4,-(SP)	
6078	014772	104402			TYPOC		
6079							
6080	014774	012023			MOV	(R0)+,(R3)+	;STORE THE DATA IN SOFTWARE TABLE
6081	014776	022730	174000		CMP	#174000,R0	;HAS THE HIGHEST LIMIT BEEN HIT?
6082	015002	001404			BEQ	3S	;BR IF ALL DONE.
6083	015004	005367	164212		DEC	TEMP5	;DEC TABLE COUNTER
6084	015010	001357			BNE	8S	;BR TO JUST TYPE DATA
6085	015012	000733			BR	6S	;BR TO TYPE ADDRESS
6086	015014	005000			3S:	CLR R0	;CLEAR DATA LIGHTS
6087	015016	000000			HALT		;HIT CONTINUE TO PROCEED.
6088	015020	000167	177420		JMP	PRG2	;GOTO PRG 2
6089	015024	006414	005012	016412	DH.2:	.ASCII <14><15><12><12><12><35><37><177><177><177>/BLIND READ OF ROM/	
	015057	015	006412	077577		.ASCIZ <15><12><15><177><177>/NOTE: NO CHECKING/	
	015106	005015	040412	042104	DH.2B:	.ASCII <15><12><12>/ADDRESS ADD+00 ADD+02 ADD+04/	
	015147	040	040440	042104		.ASCIZ / ADD+06 ADD+10 ADD+12 ADD+14 ADD+16/	
	015220	005015	042412	052130	DH.2A:	.ASCII <15><12><12>/EXTENDED 128. WORD ROM DUMP./	
	015257	015	041412	047117		.ASCII <15><12>/CONTENTS DUMPED IS PLACED IN THE SOFTWARE/	
	015332	005015	040515	027120		.ASCII <15><12>/MAP. DATA SHOULD BE VISUALLY INSPECTED!/	
		015404			.EVEN		

6090
6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102
6103
6104
6105
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124
6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143

015404 012767 015404 002260
015412 016701 176042
015416 010167 000764
015422 104400 017451
015426 104412
015430 012602
015432 011202
015434 022702 000114
015440 001464
015442 022702 000104
015446 001413
015450 022702 000122
015454 001002
015456 000167 000342
015462 022702 000101
015466 001444
015470 104400 017542
015474 000752
015476 016767 000704 163522
015504 104400 017556
015510 016746 163512
015514 016701 163506
015520 104402
015522 104400 017547
015526 104414
015530 012611
015532 005721
015534 026701 175726
015540 103413
015542 010167 163460
015546 104400 017556
015552 016746 163450
015556 104402

```
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.

PRG3: MOV #PRG3,PRG.NO ;SET FOR POWER FAIL
MOV TABLE,R1 ;DEFAULT STARTING ADDRESS TO MAP
MOV R1,ADDRESS ;SAVE THE SOFTWARE ADDRESS
XHOLD: TYPE ,MASTER ;TYPE AN "*"
RDLIN
MOV (SP)+,R2
MOV (R2),R2
CMP #114,R2 ;WAS AN "L" (LIST) HIT?
BEQ SRV.L

15: CMP #104,R2 ;WAS A "D" (DATA) HIT?
BEQ SRV.D
CMP #122,R2 ;WAS AN "R" (RUN) HIT?
BNE 10$
JMP SRV.R
10$: CMP #101,R2 ;WAS AN "A" (ADDRESS) HIT?
BEQ SRV.A
TYPE ,M.QM ;TYPE A "?"
BR XHOLD ;NEITHER A "L","P","D","R","A",OR CR WAS HIT.

163522 SRV.D: MOV ADDRESS,TEMP4 ;RESET ADDRESS POINTER.
TYPE MCRLF
MOV TEMP4,-(SP)
MOV TEMP4,R1
TYPOC

TYPE ,MSPACE

RDOCT
MOV (SP)+,(R1) ;STORE DATA

TST (R1)+ ;UPDATE THE SOFTWARE ADDRESS
CMP EXTEND,R1 ;IS THE LIMIT EXCEEDED
BLO 7$ ;INPUT LIMIT EXCEEDED!! ERROR.
MOV R1,TEMP4 ;SAVE THE ADDRESS.
TYPE MCRLF
MOV TEMP4,-(SP)
TYPOC
```

6144	015560	010167	000622		MOV	R1, ADDRESS	;SAVE THE ADDRESS FOR GOOD	
6145	015564	000167	177632		JMP	XHOLD		
6146	015570	104400	017542	7\$:	TYPE	,M,GM	;TYPE A "?"	
6147	015574	000167	177622		JMP	XHOLD		
6148								
6149								
6150								
6151								
6152								
6153								
6154	015600	104414			SRV.A:	RDOCT	;READ THE ADDRESS HE WANTS TO MODIFY.	
6155	015602	012667	000600		MOV	(SP)+, ADDRESS		
6156	015606	000167	177610	4\$:	JMP	XHOLD		
6157								
6158								
6159								
6160								
6161								
6162								
6163	015612				SRV.L:			
6164	015612	016700	175642		MOV	TABLE, R0	;GET SOFTWARE MAP	
6165	015616	016767	175640	000176	MOV	ALLEN0, DEAD	;SET DEAD END POINTER	
6166	015624	104400	017370		TYPE	,MSG4	;TYPE HEADER	
6167	015630	104400	015106		TYPE	,DH,28	;TYPE ADDRESS ADD+XX	
6168	015634	012767	000007	163360	1\$:	MOV	#7, TEMP5	;SET COUNTER FOR ACCROSS PAGE
6169	015642	011067	163360		MOV	(R0), TEMP4	;GET DATA	
6170	015646	010067	163352		MOV	R0, TEMP3	;GET ADDRESS	
6171	015652	005720			TST	(R0)+	;UPDATE ADDRESS POINTER	
6172	015654	104400	017556		TYPE	,MCRLF		
6173								
6174	015660	016746	163340		MOV	TEMP3, -(SP)		
6175	015664	104402			TYPOC			
6176								
6177	015666	104400	017547		TYPE	,MSPACE		
6178								
6179	015672	016746	163330		MOV	TEMP4, -(SP)		
6180	015676	104402			TYPOC			
6181								
6182	015700	104400	017547		TYPE	,MSPACE		
6183								
6184	015704	011067	163316	2\$:	MOV	(R0), TEMP4	;GET DATA	
6185	015710	010067	163310		MOV	R0, TEMP3	;GET ADDRESS	
6186	015714	005720			TST	(R0)+	;UPDATE POINTER	
6187								
6188	015716	016746	163304		MOV	TEMP4, -(SP)		
6189	015722	104402			TYPOC			
6190	015724	104400	017547		TYPE	,MSPACE		
6191								
6192	015730	016703	000066	3\$:	MOV	DEAD, R3		
6193	015734	005723			TST	(R3)+	;UPDATE POINTER	
6194	015736	020003			CMP	R0, R3	;LIMIT DONE ??	
6195	015740	001404			BEQ	5\$;BR IF YES	
6196	015742	005367	163254	4\$:	DEC	TEMP5	;DEC DATA COUNTER	
6197	015746	001356			BNE	2\$;BR IF MORE DATA TO GO	

```

; YOU ARE HERE BECAUSE YOU HIT AN "A"
; YOU TOLD ME YOU WERE GOING TO INPUT AN ADDRESS.
; SO INPUT THE ADDRESS AND TERMINATE WITH A CARRAGE RETURN.
; OK??

```

```

; YOU ENTERED HERE BECAUSE YOU HIT "L"
; YOU TOLD ME YOU WANTED A LISTING OF THE SOFTWARE MAP
; SO HERE IT IS.

```

6198	015750	000731				BR	1\$;TYPE THE ADDRESS
6199	015752				5\$:				
6200	015752	032767	000001	161610		BIT	#BIT0,SWR		;EXTENDED SOFTWARE DUMP?
6201	015760	001416				BEQ	6\$;BR IF NO DUMP
6202	015762	005743				TST	-(R3)		;PUSH POINTER
6203	015764	026703	175476			CMP	EXTEND,R3		
6204	015770	001412				BEQ	6\$;BR IF ALL DONE
6205	015772	104400	017416			TYPE	,MSG5		;TYPE EXTENDED MAP:
6206	015776	104400	015106			TYPE	,DH.2B		
6207	016002	016700	175456			MOV	EXTMAP,RO		;SET POINTER
6208	016006	016767	175454	000006		MOV	EXTEND,DEAD		;SET DEAD END POINTER
6209	016014	000707				BR	1\$;DO IT AGAIN SAM.
6210	016016	000167	177400		6\$:	JMP	XHOLD		
6211	016022	000000			DEAD:	0			
6212									
6213									
6214									
6215									
6216									
6217									
6218	016024								
6219	016024	012737	017674	000004	SRV.R:	MOV	#NO.TRAP,3#4		;GET READY FOR UNEXPECTED TRAP
6220	016032	012767	000500	163160	RUN3:	MOV	#500,ICOUNT		;DO TEST 500(8) TIMES
6221	016040	012700	173000		RUN.3:	MOV	#173000,RO		;SET BEGGING ADDRESS
6222	016044	012767	016070	163034		MOV	#2\$,SLPADR		;IF SW14=1; GOTO 2\$ WHEN I HIT "SCOPE"
6223	016052	016704	175402			MOV	TABLE,R4		;SET SOFTWARE RESULTS
6224	016056	016767	175410	176222		MOV	LASTA,LAST		;SET LAST ADDRESS
6225	016064	012703	000005		1\$:	MOV	#5,R3		;DO EACH ADDRESS 5 TIMES.
6226	016070	022700	173024		2\$:	CMP	#173024,RO		;DON'T DO THE VECTOR ADD.
6227	016074	001001				BNE	30\$;BR IF NOT THE VECTOR ADD.
6228	016076	022024				CMP	(RO)+,(R4)+		;UPDATE TO NEXT ADDRESS
6229	016100	022700	173224		30\$:	CMP	#173224,RO		;IS THIS THE SECOND TRAP VECTOR??
6230	016104	001001				BNE	10\$;BR IF NOT VECTOR
6231	016106	022024				CMP	(RO)+,(R4)+		;UPDATE THE POINTERS !!
6232	016110	010467	163010		10\$:	MOV	R4,\$GDDAT		
6233	016114	010067	163006			MOV	RO,\$BDDAT		
6234	016120	011067	163102			MOV	(RO),TEMP4		;READ THE ADDRESS
6235	016124	011467	163074			MOV	(R4),TEMP3		;READ THE SOFTWARE ADDRESS
6236	016130	026767	163070	163070		CMP	TEMP3,TEMP4		
6237	016136	001401				BEQ	11\$;BRANCH IF OK
6238	016140	104001				ERROR	1		;INCORRECT COMPARISON.
6239	016142	032767	004000	161420	11\$:	BIT	#BIT11,SWR		;QUICK PASS.
6240	016150	001002				BNE	12\$;BR IF YES
6241	016152	005303				DEC	R3		;HAS THAT ADD BEEN READ 10 TIMES?
6242	016154	001345				BNE	2\$;BR IF NOT 10 TIMES
6243	016156	026700	176124		12\$:	CMP	LAST,RO		;WAS LAST ADDRESS CHECKED?
6244	016162	001403				BEQ	15\$;BR IF YES
6245	016164	000004				SCOPE			;LOCK ON THIS ADDRESS?
6246	016166	022024				CMP	(RO)+,(R4)+		;UPDATE THE POINTERS.
6247	016170	000735				BR	1\$;CONTINUE THE TEST.
6248	016172	032767	000001	161370	15\$:	BIT	#BIT0,SWR		;EXTENDED WORD TO BE CHECKED?
6249	016200	001413				BEQ	3\$;BR IF NO CHECKING.
6250	016202	022767	173776	176076		CMP	#173776,LAST		;IS ALL THE TEST DONE?
6251	016210	001407				BEQ	3\$;BR IF YES.

;NOW YOU ARE HERE BECAUSE YOU WANT TO RUN THE PROGRAM
 ;REMEMBER NOW, YOU SET UP THE MAP.
 ;ARE YOU SURE YOU TYPED IN THE CORRECT DATA.???
 ;HERE WE GO

6252	016212	012767	173776	176066
6253	016220	016704	175240	
6254	016224	005720		
6255	016226	000716		

MOV	#173776, LAST	;SET LAST ADDRESS.
MOV	EXTMAP, R4	;SET EXTENDED MAP.
TST	(R0)+	;POP POINTER
BR	IS	;GO DO THE TEST.

6256											
6257											
6258											
6259	016230	012700	173000		3\$:	MOV	#173000,RO				
6260	016234	012767	016254	162644		MOV	#5\$,SLPADR				
6261	016242	012737	016310	000004		MOV	#6\$,R#4				
6262	016250	012703	000012		4\$:	MOV	#10,R3				
6263	016254	022700	173024		5\$:	CMP	#173024,RO				
6264	016260	001001				BNE	20\$				
6265	016262	005720				TST	(RO)+				
6266	016264	022700	173224		20\$:	CMP	#173224,RO				
6267	016270	001001				BNE	21\$				
6268	016272	005720				TST	(RO)+				
6269	016274	012710	177777		21\$:	MOV	#-1,(RO)				
6270	016300	000240				NOP					
6271	016302	010067	162720			MOV	RO,TEMP4				
6272	016306	104002				ERROR	2				
6273	016310	012706	001100		6\$:	MOV	#STACK,SP				
6274	016314	032767	004000	161246		BIT	#BIT11,SWR				
6275	016322	001002				BNE	22\$				
6276	016324	005303				DEC	R3				
6277	016326	001352				BNE	5\$				
6278	016330	032767	000001	161232	22\$:	BIT	#BIT0,SWR				
6279	016336	001404				BEQ	23\$				
6280	016340	022700	173776			CMP	#173776,RO				
6281	016344	001407				BEQ	7\$				
6282	016346	000403				BR	24\$				
6283	016350	026700	175116		23\$:	CMP	LASTA,RO				
6284	016354	001403				BEQ	7\$				
6285	016356	000004			24\$:	SCOPE					
6286	016360	005720				TST	(RO)+				
6287	016362	000732				BR	4\$				
6288	016364	005367	162630		7\$:	DEC	ICOUNT				
6289	016370	001004				BNE	8\$				
6290	016372	004767	001336			JSR	PC,EOP				
6291	016376	000167	177422			JMP	RUN3				
6292	016402	000167	177432		8\$:	JMP	RUN.3				
6293											
6294	016406	000000			ADDRESS:		0				

```

;TEST THAT WRITING ROM RESULTS IN A TIME OUT
;TRAP.

:SET BASE ADDRESS
:IF SW14=1; GOTO 5$ AT SCOPE
:TIME OUT TRAP; GOTO 6$
:DO EACH ADD 10 TIMES
:IS THIS AT THE TRAP VECTOR
:BR IF NO
:UPDATE POINTER
:IS THIS AT THE SECOND TRAP VECTOR
:BR IF NO
:UPDATE THE POINTER
:WRITE ROM WITH A -1
:WAIT ONE INSTR. TIME

:WRITING ROM DIDN'T TIME OUT.
:RESTORE STACK
:QUICK PASS?
:BR IF YES
:DO EACH ADD 10 TIMES
:NOT DONE WITH THIS ONE YET.
:IS THE EXTENDED 128. WORDS TO BE TESTED??
:BR IF NO
:IS THE EXTENDED LIMIT BEEN TESTED?
:IF YES; GOTO 7$
:IF NO; KEEP GOING.
:ALL DONE??
:IF YES; GOTO 7$
:GO CHECK SW14; (FREEZE !!)
:UPDATE TO NEXT ADDRESS
:GO DO IT AGAIN
:CHECK ITERATION COUNT
:MORE TO GO
:GO TO END OF PASS ROUTINE
:GO DO TEST AGAIN

```


6295
6296
6297
6298
6299
6300
6301
6302
6303
6304
6305
6306
6307
6308
6309
6310
6311
6312
6313
6314
6315
6316
6317
6318
6319
6320
6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331
6332
6333
6334
6335
6336
6337
6338
6339
6340
6341
6342
6343
6344
6345
6346
6347
6348

016410 012767 016410 001254
016416 005067 162574
016422 012706 001100
016426 012767 020000 162564
016434 005767 000554
016440 001106
016442 005167 000546
016446 012705 000002
016452 012704 017204
016456 012737 017674 000004
016464 012767 000001 162532
016472 104400 017216
016476 104400 017547

016502 016746 162516
016506 104402
016510 104400 017547

016514 104400 017232
016520 104400 017245
016524 012700 173024
016530 005037 173024
016534 010537 173024
016540 000240
016542 012706 001100
016546 012700 173024
016552 012737 017674 000004
016560 013767 173024 162440

016566 104400 017547
016572 016746 162430
016576 104402
016600 013724 173024
016604 104400 017301
016610 012700 173224
016614 013767 173224 162404
016622 104400 017547

016626 016746 162374
016632 104402

```

;PROGRAM 4
;PROGRAM 4 CHECKS THE TRAP VECTOR ADDRESS.
;THE PROGRAM SIMULATES ACTIVATING THE BUTTON
;FOR EACH CHANNEL AND THEN READS
;THE CONTENTS OF THE ADDRESS.
;ON THE FIRST PASS THE CONTENTS WILL
;BE TYPED OUT FOR YOU THE
;USER TO VERIFY. AFTER THIS THE PROGRAM
;DOES A COMPARE TO THE PREVIOUSLY FOUND DATA
;AND REPORTS AN ERROR IF DIFFERENT THAN
;WHAT WAS FOUND BEFORE.
    
```

```

PRG4:  MOV    #PRG4,PRG.NO    ;SET FOR POWER FAIL
        CLR    LSTERR        ;PREPARE ERROR CONDITIONS
        MOV    #STACK,SP     ;SET THE STACK POINTER
        MOV    #20000,ICOUNT ;SET ITERATION COUNT TO 20000(8)
        TST   FLAG4         ;HAVE I BEEN HERE BEFOR??
        BNE   TAG.A         ;BR IF NOT FIRST TIME HERE.
        COM   FLAG4         ;SET THE FLAG
        MOV    #2,R5         ;SET R5 FOR SWITCH 1
        MOV    #LOC1,R4      ;SET STORAGE LOCATION
        MOV    #NO.TRAP,#4   ;SET FOR TIME OUT TRAP
        MOV    #1,TEMP3      ;SET FOR MESSAGE ON CHANNEL NO.
1$:     TYPE   ,MCHAN        ;TYPE MESSAGE ABOUT CHANNEL
        TYPE   ,MSPACE
        MOV    TEMP3,-(SP)
        TYPOC
        TYPE   ,MSPACE
        TYPE   ,MACTV       ;TYPE REST OF MESSAGE
2$:     TYPE   ,MADD1       ;TYPE ADDRESS MESSAGE
        MOV    #173024,RO
        CLR   #173024
        MOV   R5,#173024    ;WRITE ROM WITH SWITCH
        NOP   ;WAIT ONE INSTR. TIME
3$:     MOV    #STACK,SP     ;SET THE STACK POINTER
        MOV    #173024,RO    ;SET FOR ERROR MESSAGE
        MOV    #NO.TRAP,#4   ;SET FOR NO MORE TRAPS
        MOV    #173024,TEMP4 ;READ THE ADDRESS
        TYPE   MSPACE
        MOV    TEMP4,-(SP)
        TYPOC
        MOV    #173024,(R4)+ ;STORE THE INFORMATION FOUND
        TYPE   ,MADD2       ;TYPE THE SECOND ADDRESS MSG
        MOV    #173224,RO    ;SET FOR ERROR CONDITION.
        MOV    #173224,TEMP4 ;STORE ROM DATA
        TYPE   ,MSPACE
        MOV    TEMP4,-(SP)
        TYPOC
    
```

6349	016634	005267	162364		INC	TEMP3			;GET READY FOR NEXT SWITCH SETTING
6350	016640	000241			CLC				;CLEAR THE CARRY BIT
6351	016642	006105			ROL	R5			;UPDATE R5
6352	016644	022705	000040		CMP	#40,R5			;ALL SIMULATED SWITCHS DONE?
6353	016650	001310			BNE	1\$;BR IF NOT ALL DONE
6354	016652	000167	177532		JMP	PRG4			;JMP AND DO TEST AGAIN WITH OUT TYPE OUT
6355									
6356	016656	012703	000002		MOV	#2 R3			;SIMULATE SWITCH 1
6357	016662	012704	017204		MOV	#LOC1,R4			;GET LOCATION WHERE DATA IS STORED
6358	016666	012737	017674	000004	1\$:	MOV	#NO.TRAP,#4		;PREPARE FOR TIME OUT TRAP
6359	016674	005037	173024		CLR	#173024			
6360	016700	010337	173024		MOV	R3,#173024			;WRITE THE ROM
6361	016704	000240			NOP				;WAIT ONE INSTR. TIME
6362	016706	012706	001100		MOV	#STACK,SP			;SET THE STACK POINTER.
6363	016712	012737	017674	000004	2\$:	MOV	#NO.TRAP,#4		;SET FOR NO MORE TRAPS.
6364	016720	012700	173024		MOV	#173024,R0			;SET FOR ERROR MESSAGE
6365	016724	011401			MOV	(R4),R1			;SET FOR COMPARISON
6366	016726	013705	173024		MOV	#173024,R5			;GET THE DATA FROM THE ROM
6367	016732	012767	017204	162164	MOV	#LOC1,\$GDDAT			
6368	016740	012767	173024	162160	MOV	#173024,\$BDDAT			
6369	016746	016767	000232	162250	MOV	LOC1,TEMP3			
6370	016754	013767	173024	162244	MOV	#173024,TEMP4			
6371	016762	020105			CMP	R1,R5			;IS THE DATA THE SAME??
6372	016764	001401			BEQ	30\$;BR IF GOOD DATA.
6373	016766	104001			ERROR	1			;ERROR. DATA READ FIRST TIME NOT THE SAME
6374	016770	012700	173224		30\$:	MOV	#173224,R0		;SET FOR ERROR MESSAGE
6375	016774	013705	173224		MOV	#173224,R5			;READ THE ROM
6376	017000	012767	173224	162120	MOV	#173224,\$BDDAT			
6377	017006	013767	173224	162212	MOV	#173224,TEMP4			
6378	017014	020105			CMP	R1,R5			;IS THE DATA THE SAME?
6379	017016	001401			BEQ	31\$;BR IF GOOD DATA
6380	017020	104001			ERROR	1			;ERROR. DATA NOT THE SAME AS BEFORE.
6381	017022	005724			31\$:	TST	(R4)+		;UPDATE DATA POINTER.
6382	017024	000241			CLC				;CLEAR THE CARRY BIT
6383	017026	006103			ROL	R3			;UPDATE THE SIMULATED SWITCH SETTING
6384	017030	022703	000040		CMP	#40,R3			;HAVE ALL SETTING BEEN DONE
6385	017034	001314			BNE	1\$;BR IF NOT DONE
6386	017036	005367	162156		DEC	ICOUNT			;ITERATION COUNT DONE
6387	017042	001305			BNE	TAG.A			;BR IF NOT DONE
6388	017044	122767	000112	000742	CMPB	#112,VERSON			;TEST FOR J VERSION
6389	017052	001450			BEQ	33\$;IF J VERSION SKIP TEST FOR DEFAULT OPTION
6390	017054	012737	177777	173224	MOV	#-1,#173224			;WRITE SECOND TRAP VECTOR WITH -1
6391	017062	005037	173024		CLR	#173024			;ZERO THE FIRST VECTOR
6392	017066	012700	173024		MOV	#173024,R0			;SET FOR TYPE OUT IF ERROR
6393	017072	016701	000106		MOV	LOC1,R1			;SET FOR TYPE OUT ROUTINE
6394	017076	013705	173024		MOV	#173024,R5			;SAME AS ABOVE
6395	017102	012767	173024	162016	MOV	#173024,\$BDDAT			
6396	017110	013767	173024	162110	MOV	#173024,TEMP4			
6397	017116	020105			CMP	R1,R5			;IS DEFAULT LINE SELECTED =TO LINE 1
6398	017120	001401			BEQ	32\$;BR IF DEFAULT EQUALS LINE 1
6399	017122	104001			ERROR	1			;DATA NOT EQUAL TO LINE 1
6400	017124	012737	177777	173024	32\$:	MOV	#-1,#173024		;WRITE A -1 TO FIRST VECTOR
6401	017132	005037	173224		CLR	#173224			;ZERO SECOND VECTOR
6402	017136	012700	173224		MOV	#173224,R0			;SET FOR TYPE OUT IF ERROR

```

6403 017142 016701 000036          MOV    LOC1,R1          ;GET DATA
6404 017146 013705 173224          MOV    @#173224,R5     ;READ ROM
6405 017152 012767 173224 161746   MOV    @#173224,$BDDAT
6406 017160 013767 173224 162040   MOV    @#173224,TEMP4
6407
6408 017166 020105          CMP    R1,R5           ;IS LINE 1 DEFAULT LINE
6409 017170 001401          BEQ   33$             ;BR IF OK
6410 017172 104001          ERROR 1              ;ERROR LINE 1 NOT DEFAULT LINE
6411 017174 004767 000534          JSR   PC,EOP          ;TYPE END MESSAGE.
6412 017200 000167 177204          JMP   PRG4            ;GOTO PROGRAM 4 AGAIN
6413
6414 017204 000000          LOC1: 0
6415 017206 000000          LOC2: 0
6416 017210 000000          LOC3: 0
6417 017212 000000          LOC4: 0
6418 017214 000000          FLAG4: 0

```

6419	017216	005015	041412	040510	MCHAN:	.ASCIZ <15><12><12>/CHANNEL /
	017232	041501	044524	040526	MACTV:	.ASCIZ /ACTIVATED./
	017245	015	040412	042104	MADD1:	.ASCIZ <15><12>/ADDRESS 773024 CONTAINS: /
	017301	015	040412	042104	MADD2:	.ASCIZ <15><12>/ADDRESS 773224 CONTAINS: /
		017336			.EVEN	
6420						
6421	017336	005015	051120	043517	MSG3:	.ASCIZ <15><12>/PROGRAM NO. (1,2,3,4) /
	017370	006414	016412	077437	MSG4:	.ASCIZ <14><15><12><35><37><177><177><177>/SOFTWARE MAP:/
	017416	005015	020012	054105	MSG5:	.ASCIZ <15><12><12>/ EXTENDED SOFTWARE MAP:/
	017451	015	025012	000	MASTER:	.ASCIZ <15><12>*/
	017455	007	006407	042412	M.END:	.ASCIZ <7><7><15><12>/END PASS BM873-Y/
	017502				MFAIL:	
	017502	005015	053520	020122		.ASCII <15><12>/PWR UP AFTER/
	017520	005015	042522	046101		.ASCIZ <15><12>/REAL PWR FAIL/
	017540	000044			M.DOL:	.ASCIZ /S/
	017542	005015	037477	000	M.QM:	.ASCIZ <15><12>??/
	017547	040	000040		MSPACE:	.ASCIZ / /
	017552	020040	000040		SPACE3:	.ASCIZ / /
	017556	005015	000		MCRLF:	.ASCIZ <15><12>
	017561	012	000		MLF:	.ASCIZ <12>
		017564			.EVEN	
6422						

6423	017564	005067	161426		.PFAIL:	CLR	LSTERR	
6424	017570	013746	000004			MOV	2#4, -(SP)	
6425	017574	012737	017624	000004		MOV	1\$, 2#4	
6426	017602	005737	173000			TST	2#173000	: IS THIS PF REAL?
6427	017606	000240				NOP		: TRAP IS CAUSED BY LOADER
6428	017610	012737	017634	000024		MOV	2#PWR.UP, 2#24	; ITS REAL. PREPARE FOR PWR UP
6429	017616	012637	000004			MOV	(SP)+, 2#4	
6430	017622	000000				HALT		
6431	017624	005726			1\$:	TST	(SP)+	: POP THE STACK.
6432	017626	012637	000004			MOV	(SP)+, 2#4	
6433	017632	000000				HALT		: HARDWARE ERROR. BOOT DIDN'T FORCE
6434								: HIGH ADDR LINES AND LOAD BUTTON WAS ACTIVATED
6435	017634	012737	017564	000024	PWR.UP:	MOV	2#PFAIL, 2#24	
6436	017642	012706	001100			MOV	2#STACK, SP	
6437	017646	005000				CLR	RO	: SET DELAY
6438	017650	062700	000001		1\$:	ADD	1, RO	: WAIT FOR TTY
6439	017654	001375				BNE	1\$	
6440	017656	104400	017502			TYPE	2#MFAIL	: TYPE FAILED.
6441	017662	005067	160110			CLR	PS	: SET STATUS TO ZERO
6442	017666	000177	000000			JMP	2#PRG.NO	
6443	017672	000000			PRG.NO:	0		
6444	017674				NO.TRAP:			
6445	017674	011667	000032			MOV	(SP) XSTORE	
6446	017700	032716	100000			BIT	2#BIT15, (SP)	
6447	017704	001410				BEQ	1\$	
6448	017706	011600				MOV	(SP), RO	
6449	017710	104004				ERROR	4	
6450	017712	012706	001100			MOV	2#STACK, SP	
6451	017716	005067	160054			CLR	PS	
6452	017722	000177	177744			JMP	2#PRG.NO	
6453	017726	104003			1\$:	ERROR	3	
6454	017730	000002				RTI		
6455	017732	000000			XSTORE:	0		
6456								
6457	017734	005067	161256		EOP:	CLR	LSTERR	
6458	017740	104400	017455			TYPE	2#M.END	
6459	017744	104400	020014			TYPE	2#VERSION	
6460	017750	013701	000042			MOV	2#42, R1	
6461	017754	001416				BEQ	X1	
6462	017756	022767	013716	177706		CMP	2#PRG1, PRG.NO	
6463	017764	001002				BNE	2#+6	
6464	017766	000167	176416			JMP	PRG4	
6465	017772	013701	000042			MOV	2#42, R1	
6466	017776	001405				BEQ	X1	
6467	020000	000005				RESET		
6468	020002				SENDAD:			
6469	020002	004711			LOGIC:	JSR	PC, (R1)	
6470	020004	000240				NOP		
6471	020006	000240				NOP		
6472	020010	000240				NOP		
6473	020012	000207			X1:	RTS	PC	
6474	020014	000101			VERSION:	101		: SEVEN BIT ASCII FOR DEFAULT "A"

```

6475 020016 005015 041520 020072 MERRPC: .ASCIZ <15><12>/PC: /
6476 020024 000
6477 020026
6478 .EVEN
6479 .MCALL .SEOP, .STYPE, .STYPOCT, .SPOWER, .SREAD
6480 ;;*****
6481 .SBTTL TYPE ROUTINE
6482
6483 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
6484 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
6485 ;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
6486 ;*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
6487 ;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
6488 ;*
6489 ;*CALL:
6490 ;*1) USING A TRAP INSTRUCTION
6491 ;* TYPE ,MESADR ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
6492 ;*OR
6493 ;* TYPE
6494 ;* MESADR
6495 ;*
6496 ;*2) USING A JSR INSTRUCTION
6497 ;* MOV PS,-(SP) ;PUSH PROCESSOR STATUS WORD ON THE STACK
6498 ;* JSR PC,$TYPE ;CALL TYPE ROUTINE
6499 ;* MESADDR ;FIRST ADDRESS OF MESSAGE
6500
6501 020026 105767 161117 $TYPE: TSTB $TPFLG ;IS THERE A TERMINAL?
6502 020032 100002 BPL 1$ ;BR IF YES
6503 020034 000000 HALT ;HALT HERE IF NO TERMINAL
6504 020036 000407 BR 3$ ;LEAVE
6505 020040 010046 1$: MOV RO,-(SP) ;SAVE RO
6506 020042 017600 000002 MOV @2(SP),RO ;GET ADDRESS OF ASCIZ STRING
6507 020046 112046 2$: MOVB (RO)+,-(SP) ;PUSH CHARACTER TO BE TYPED ONTO STACK
6508 020050 001005 BNE 4$ ;BR IF IT ISN'T THE TERMINATOR
6509 020052 005726 TST (SP)+ ;IF TERMINATOR POP IT OFF THE STACK
6510 020054 012600 MOV (SP)+,RO ;RESTORE RO
6511 020056 062716 000002 3$: ADD #2,(SP) ;ADJUST RETURN PC
6512 020062 000002 RTI ;RETURN
6513 020064 004767 000026 4$: JSR PC,7$ ;GO TYPE THIS CHARACTER
6514 020070 126726 161054 5$: CMPB $FILLC,(SP)+ ;IS IT TIME FOR FILLER CHARS.?
6515 020074 001364 BNE 2$ ;IF NO GO GET NEXT CHAR.
6516 020076 016746 161044 MOV $NULL,-(SP) ;GET # OF FILLER CHARS. NEEDED
6517 ;AND THE NULL CHAR.
6518 020102 105366 000001 6$: DECB 1(SP) ;DOES A NULL NEED TO BE TYPED?
6519 020104 002770 BLT 5$ ;BR IF NO--GO POP THE NULL OFF OF STACK
6520 020110 004767 000002 JSR PC,7$ ;GO TYPE A NULL
6521 020114 000772 BR 5$ ;LOOP
6522 020116 105777 161020 7$: TSTB @STPS ;WAIT UNTIL PRINTER IS READY
6523 020122 100375 BPL 7$
6524 020124 116677 000002 161012 MOVB 2(SP),@STPB ;LOAD CHAR TO BE TYPED INTO DATA REG.
6525 020132 000207 RTS PC
6526 ;;*****
6527
6528 .SBTTL TTY INPUT ROUTINE

```

```

6529
6530 ;*INPUT A SINGLE CHARACTER FROM THE TTY
6531 ;*CALL:
6532 ;* RDCHR ;INPUT A SINGLE CHARACTER FROM THE TTY
6533 ;* RETURN HERE ;CHARACTER IS ON THE STACK
6534
6535
6536 020134 011646 000004 000002 SRDCHR: MOV (SP),-(SP) ;PUSH DOWN THE PC
6537 020136 016666 000004 000002 1$: MOV 4(SP),2(SP) ;SAVE THE PS
6538 020144 105777 160766 1$: TSTB @STKS ;WAIT FOR
6539 020150 100375 1$: BPL 1$ ;A CHARACTER
6540 020152 117766 160762 000004 1$: MOVB @STKB,4(SP) ;READ THE TTY
6541 020160 042766 177600 000004 BIC #1C<177>,4(SP) ;GET RID OF JUNK IF ANY
6542 020166 000002 RTI ;GO BACK TO USER
6543
6544 ;*****
6545 ;*INPUT A STRING FROM THE TTY
6546 ;*CALL:
6547 ;* RDLIN ;INPUT A STRING FROM THE TTY
6548 ;* RETURN HERE ;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
6549 ;* ;TERMINATOR WILL BE A BYTE OF ALL 0'S
6550 020170 010346 SRDLIN: MOV R3, -(SP) ;SAVE R3
6551 020172 012703 020276 1$: MOV #STTYIN, R3 ;GET ADDRESS
6552 020176 022703 020306 2$: CMP #STTYIN+8., R3 ;BUFFER FULL?
6553 020202 101405 BLOS 4$ ;BR IF YES
6554 020204 104410 RDCHR ;GO READ ONE CHARACTER FROM THE TTY
6555 020206 112613 MOV (SP)+, (R3) ;GET CHARACTER
6556 020210 122713 000177 CMPB #177, (R3) ;IS IT A RUBOUT
6557 020214 001003 BNE 3$ ;SKIP IF NOT
6558 020216 104400 001152 4$: TYPE $QUES ;TYPE A '?'
6559 020222 000763 BR 1$ ;CLEAR THE BUFFER AND LOOP
6560 020224 111367 000044 3$: MOV (R3), 8$ ;ECHO THE CHARACTER
6561 020230 104400 020274 TYPE 8$
6562 020234 122723 000015 CMPB 15, (R3)+ ;CHECK FOR RETURN
6563 020240 001356 BNE 2$ ;LOOP IF NOT RETURN
6564 020242 105063 177777 CLRB -1(R3) ;CLEAR RETURN (THE 15)
6565 020246 104400 001154 TYPE $LF ;TYPE A LINE FEED
6566 020252 012603 MOV (SP)+, R3 ;RESTORE R3
6567 020254 011646 MOV (SP), -(SP) ;ADJUST THE STACK AND PUT ADDRESS OF THE
6568 020256 016666 000004 000002 MOV 4(SP), 2(SP) ; FIRST ASCII CHARACTER ON IT
6569 020264 012766 020276 000004 MOV #STTYIN, 4(SP)
6570 020272 000002 RTI ;RETURN
6571 020274 000 8$: .BYTE 0 ;STORAGE FOR ASCII CHAR. TO TYPE
6572 020275 000 .BYTE 0 ;TERMINATOR
6573 020276 000010 $TTYIN: .BLKB 8. ;RESERVE 8 BYTES FOR TTY INPUT
6574 ;*****
6575
6576 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
6577
6578 ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
6579 ;*CALL:
6580 ;* MOV NUM, -(SP) ;NUMBER TO BE TYPED
6581 ;* TYPOS ;CALL FOR TYPEOUT
6582 ;* .BYTE N ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE

```

```

6583      *      .BYTE      M      ;M=1 OR 0
6584      *
6585      *
6586      *
6587      *STYPON-----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
6588      *STYPOS OR STYPOC
6589      *CALL:
6590      *      MOV      NUM,-(SP)      ;NUMBER TO BE TYPED
6591      *      TYPON
6592      *
6593      *STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
6594      *CALL:
6595      *      MOV      NUM,-(SP)      ;NUMBER TO BE TYPED
6596      *      TYPOC
6597      *
6598      020306 017646 000000      STYPOS: MOV      2(SP),-(SP)      ;PICKUP THE MODE
6599      020312 116667 000001 000211      MOVVB    1(SP),SOFILL      ;LOAD ZERO FILL SWITCH
6600      020320 112667 000207      MOVVB    (SP)+,SOMODE+1    ;NUMBER OF DIGITS TO TYPE
6601      020324 062716 000002      ADD      #2,(SP)          ;ADJUST RETURN ADDRESS
6602      020330 000406      BR      STYPON
6603      020332 112767 000001 000171      STYPOC: MOVVB    #1,SOFILL      ;SET THE ZERO FILL SWITCH
6604      020340 112767 000006 000135      MOVVB    #6,SOMODE+1      ;SET FOR SIX(6) DIGITS
6605      020346 112767 000005 000154      STYPON: MOVVB    #5,SOCNT      ;SET THE ITERATION COUNT
6606      020354 010346      MOV      R3,-(SP)          ;SAVE R3
6607      020356 010446      MOV      R4,-(SP)          ;SAVE R4
6608      020360 010546      MOV      R5,-(SP)          ;SAVE R5
6609      020362 116704 000145      MOVVB    SOMODE+1,R4      ;GET THE NUMBER OF DIGITS TO TYPE
6610      020366 005404      NEG      R4
6611      020370 062704 000006      ADD      #6,R4            ;SUBTRACT IT FOR MAX. ALLOWED
6612      020374 110467 000132      MOVVB    R4,SOMODE        ;SAVE IT FOR USE
6613      020400 116704 000125      MOVVB    SOFILL,R4        ;GET THE ZERO FILL SWITCH
6614      020404 016605 000012      MOV      12(SP),R5        ;PICKUP THE INPUT NUMBER
6615      020410 005003      CLR      R3                ;CLEAR THE OUTPUT WORD
6616      020412 006105      1$:    ROL      R5          ;ROTATE MSB INTO "C"
6617      020414 000404      BR      3$                ;GO DO MSB
6618      020416 006105      2$:    ROL      R5          ;FORM THIS DIGIT
6619      020420 006105      ROL      R5
6620      020422 006105      ROL      R5
6621      020424 010503      MOV      R5,R3
6622      020426 006103      3$:    ROL      R3          ;GET LSB OF THIS DIGIT
6623      020430 105367 000076      DECB    SOMODE            ;TYPE THIS DIGIT?
6624      020434 100016      BPL     7$                ;BR IF NO
6625      020436 042703 177770      BIC     #177770,R3        ;GET RID OF JUNK
6626      020442 001002      BNE     4$                ;TEST FOR 0
6627      020444 005704      TST     R4                ;SUPPRESS THIS 0?
6628      020446 001403      BEQ     5$                ;BR IF YES
6629      020450 005204      4$:    INC     R4            ;DON'T SUPPRESS ANYMORE 0'S
6630      020452 052703 000060      BIS     #'0,R3            ;MAKE THIS DIGIT ASCII
6631      020456 052703 000040      5$:    BIS     #' ,R3        ;MAKE ASCII IF NOT ALREADY
6632      020462 110367 000040      MOVVB    R3,8$            ;SAVE FOR TYPING
6633      020466 104400 020526      TYPE    8$                ;GO TYPE THIS DIGIT
6634      020472 105367 000032      7$:    DECB    $OCNT        ;COUNT BY 1
6635      020476 003347      BGT     2$                ;BR IF MORE TO DO
6636      020500 002402      BLT     6$                ;BR IF DONE

```



```

6637 020502 005204          INC      R4          ;INSURE LAST DIGIT ISN'T A BLANK
6638 020504 000744          BR       2$          ;GO DO THE LAST DIGIT
6639 020506 012605          6$:     MOV      (SP)+,R5      ;RESTORE R5
6640 020510 012604          MOV      (SP)+,R4      ;RESTORE R4
6641 020512 012603          MOV      (SP)+,R3      ;RESTORE R3
6642 020514 016666 000002 000004  MOV      2(SP),4(SP)    ;SET THE STACK FOR RETURNING
6643 020522 012616          MOV      (SP)+,(SP)
6644 020524 000002          RTI          ;RETURN
6645 020526 000          8$:     .BYTE   0          ;STORAGE FOR ASCII DIGIT
6646 020527 000          .BYTE   0          ;TERMINATOR FOR TYPE ROUTINE
6647 020530 000          $OCNT:  .BYTE   0          ;OCTAL DIGIT COUNTER
6648 020531 000          $OFILL: .BYTE   0          ;ZERO FILL SWITCH
6649 020532 000000          $OMODE: 0          ;NUMBER OF DIGITS TO TYPE
6650                                     ;*****
6651                                     ;*****
6652                                     .SBTTL  SCOPE HANDLER ROUTINE
6653
6654                                     ;*SW14=1          LOOP ON TEST
6655                                     ;*THE TEST NUMBER ($STNM) IS INCREMENTED AND DISPLAYED IN DISPLAY<7:0>
6656                                     ;*AND THE ERROR FLAG ($ERFLG) IS DISPLAYED IN DISPLAY<15:08>
6657
6658 020534          $SCOPE:
6659 020534 006137 177570          ROL     @#SWR          ;LOOP ON PRESENT TEST?
6660 020540 100425          BMI     $OVER          ;YES IF SW14=1
6661                                     ;*****START OF CODE FOR THE XOR TESTER*****
6662 020542 000416          $XTSTR: BR      6$          ;IF RUNNING ON THE "XOR" TESTER CHANGE
6663                                     ;THIS INSTRUCTION TO A "NOP" (NOP=240)
6664 020544 013746 000004          MOV     @#ERRVEC,-(SP)  ;SAVE THE CONTENTS OF THE ERROR VECTOR
6665 020550 012737 020570 000004  MOV     #5$,@#ERRVEC    ;SET FOR TIMEOUT
6666 020556 005737 177060          TST    @#177060        ;TIME OUT ON XOR?
6667 020562 012637 000004          MOV     (SP)+,@#ERRVEC ;RESTORE THE ERROR VECTOR
6668 020566 000404          BR     $$VLAD          ;GO TO THE NEXT TEST
6669 020570 022626          5$:     CMP     (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
6670 020572 012637 000004          MOV     (SP)+,@#ERRVEC ;RESTORE THE ERROR VECTOR
6671 020576 000406          BR     $OVER          ;LOOP ON THE PRESENT TEST
6672 020600          6$:     ;*****END OF CODE FOR THE XOR TESTER*****
6673 020600 105267 160276          $$VLAD: INCB    $STNM      ;COUNT TEST NUMBERS
6674 020604 011667 160276          MOV     (SP),$LPADR    ;SAVE SCOPE LOOP ADDRESS
6675 020610 105067 160267          CLRB   $ERFLG         ;ZERO THE ERROR FLAG
6676 020614 016737 160262 177570  $OVER:  MOV     $STNM,@#DISPLAY ;DISPLAY TEST NUMBER
6677 020622 016716 160260          MOV     $LPADR,(SP)   ;FUDGE RETURN ADDRESS
6678 020626 000002          RTI          ;FIXES PS
6679                                     ;*****
6680                                     ;*****
6681                                     .SBTTL  ERROR HANDLER ROUTINE
6682
6683                                     ;*SW15=1          HALT ON ERROR
6684                                     ;*SW13=1          INHIBIT ERROR TYPEOUTS
6685                                     ;*GO TO $ERRTYP ON ERROR
6686
6687 020630          $ERROR:
6688 020630 105267 160247          7$:     INCB    $ERFLG    ;SET THE ERROR FLAG
6689 020634 001775          BEQ    7$            ;DON'T LET THE FLAG GO TO ZERO
6690 020636 016737 160240 177570  MOV     $STNM,@#DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
    
```

```

6691 020644 005267 160242
6692 020650 011667 160242
6693 020654 162767 000002 160234
6694 020662 117767 160230 160224
6695 020670 032737 020000 177570
6696 020676 001004
6697 020700 004737 020722
6698 020704 104400 001153
6699 020710 005737 177570
6700 020714 100001
6701 020716 000000
6702 020720 000002
6703
6704
6705
6706
6707
6708
6709
6710
6711 020722
6712 020722 104400 001153
6713 020726 010046
6714 020730 005000
6715 020732 153700 001114
6716 020736 001004
6717
6718 020740 016746 160152
6719
6720 020744 104402
6721 020746 000426
6722 020750 005300
6723 020752 006300
6724 020754 006300
6725 020756 006300
6726 020760 062700 001156
6727 020764 012067 000004
6728 020770 001404
6729 020772 104400
6730 020774 000000
6731 020776 104400 001153
6732 021002 012067 000004
6733 021006 001404
6734 021010 104400
6735 021012 000000
6736 021014 104400 001153
6737 021020 011000
6738 021022 001004
6739 021024 012600
6740 021026 104400 001153
6741 021032 000207
6742 021034
6743 021034 013046
6744 021036 104402

```

```

INC SERTTL ;INC THE ERROR COUNT
MOV (SP), SERRPC ;GET ADDRESS OF ERROR INSTRUCTION
SUB #2, SERRPC
MOVB @SERRPC, $ITEMB ;STRIP AND SAVE THE ERROR ITEM CODE
BIT #SW13, @SWR ;SKIP TYPEOUT IF SET
BNE 2$ ;SKIP TYPEOUTS
JSR PC, @SERRTYP ;GO TO USER ERROR ROUTINE
TYPE $CRLF
2$: TST @SWR ;HALT ON ERROR
BPL 3$ ;SKIP IF CONTINUE
HALT ;HALT ON ERROR!
3$: RTI ;RETURN
;*****
.SBTTL ERROR MESSAGE TYPEOUT ROUTINE
; *THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
; *ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
; *AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
SERRTYP:
TYPE $CRLF ;"CARRIAGE RETURN" & "LINE FEED"
MOV RO, -(SP) ;SAVE RO
CLR RO ;PICKUP THE ITEM INDEX
BISB @SITEMB, RO
BNE 1$ ;IF ITEM NUMBER IS ZERO, JUST
;TYPE THE PC OF THE ERROR
;SAVE SERRPC FOR TYPEOUT
;ERROR ADDRESS
;GO TYPE--OCTAL ASCII(ALL DIGITS)
;GET OUT
;ADJUST THE INDEX SO THAT IT WILL
;WORK FOR THE ERROR TABLE
1$: DEC RO
ASL RO
ASL RO
ASL RO
ADD #SERRTB, RO ;FORM TABLE POINTER
MOV (RO)+, 2$ ;PICKUP "ERROR MESSAGE" POINTER
BEQ 3$ ;SKIP TYPEOUT IF NO POINTER
TYPE ;TYPE THE "ERROR MESSAGE"
2$: .WORD 0 ;"ERROR MESSAGE" POINTER GOES HERE
TYPE $CRLF ;"CARRIAGE RETURN" & "LINE FEED"
3$: MOV (RO)+, 4$ ;PICKUP "DATA HEADER" POINTER
BEQ 5$ ;SKIP TYPEOUT IF 0
TYPE ;TYPE THE "DATA HEADER"
4$: .WORD 0 ;"DATA HEADER" POINTER GOES HERE
TYPE $CRLF ;"CARRIAGE RETURN" & "LINE FEED"
5$: MOV (RO), RO ;PICKUP "DATA TABLE" POINTER
BNE 7$ ;GO TYPE THE DATA
6$: MOV (SP)+, RO ;RESTORE RO
TYPE $CRLF ;"CARRIAGE RETURN" & "LINE FEED"
RTS PC ;RETURN
7$: MOV @RO+, -(SP) ;SAVE @RO+ FOR TYPEOUT
TYPOC ;GO TYPE--OCTAL ASCII(ALL DIGITS)

```

```

6745 021040 005710          TST      (RO)          ; IS THERE ANOTHER NUMBER?
6746 021042 001770          BEQ      6$             ; BR IF NO
6747 021044 104400 021052   TYPE     ,8$           ; TYPE TWO(2) SPACES
6748 021050 000771          BR       7$             ; LOOP
6749 021052 020040 000     BS:      .ASCIZ  / /         ; TWO(2) SPACES
6750          021056          .EVEN
6751          ;*****
6752          .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
6753
6754
6755          ;*CALL:
6756          ;*      RDOCT          ; READ AN OCTAL NUMBER
6757          ;*      RETURN HERE   ; LOW ORDER BITS ARE ON TOP OF THE STACK
6758          ;*
6759          ;*
6760 021056 011646          $RDOCT: MOV     (SP),-(SP)      ; PROVIDE SPACE FOR THE
6761 021060 016666 000004 000002  MOV     4(SP),2(SP)      ; INPUT NUMBER
6762 021066 010046          MOV     RO,-(SP)         ; PUSH RO ON STACK
6763 021070 010146          MOV     R1,-(SP)         ; PUSH R1 ON STACK
6764 021072 010246          MOV     R2,-(SP)         ; PUSH R2 ON STACK
6765 021074 104412          1$:     RDLIN          ; READ AN ASCIZ LINE
6766 021076 012600          MOV     (SP)+,RO        ; GET ADDRESS OF 1ST CHARACTER
6767 021100 005001          CLR     R1              ; CLEAR DATA WORD
6768 021102 005002          CLR     R2
6769 021104 112046          2$:     MOVVB   (RO)+,-(SP) ; PICKUP THIS CHARACTER
6770 021106 001412          BEQ     3$             ; IF ZERO GET OUT
6771 021110 006301          ASL    R1              ; *2
6772 021112 006102          ROL    R2
6773 021114 006301          ASL    R1              ; *4
6774 021116 006102          ROL    R2
6775 021120 006301          ASL    R1              ; *8
6776 021122 006102          ROL    R2
6777 021124 042716 177770   BIC    #1C7,(SP)        ; STRIP THE ASCII JUNK
6778 021130 062601          ADD    (SP)+,R1        ; ADD IN THIS DIGIT
6779 021132 000764          BR     2$             ; LOOP
6780 021134 005726          3$:     TST     (SP)+      ; CLEAN TERMINATOR FROM STACK
6781 021136 010166 000012   MOV     R1,12(SP)       ; SAVE THE RESULT
6782 021142 010267 000010   MOV     R2,$HIOCT
6783 021146 012602          MOV     (SP)+,R2
6784 021150 012601          MOV     (SP)+,R1
6785 021152 012600          MOV     (SP)+,RO
6786 021154 000002          RTI
6787 021156 000000          $HIOCT: .WORD 0        ; HIGH ORDER BITS GO HERE
6788          ;*****
6789          .SBTTL  TRAP DECODER
6790
6791
6792          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
6793          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
6794          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
6795          ;*GO TO THAT ROUTINE.
6796
6797 021160 010046          $TRAP: MOV     RO,-(SP)   ; SAVE RO
6798 021162 016600 000002   MOV     2(SP),RO       ; GET TRAP ADDRESS

```

```

6799 021166 005740          TST      -(RO)          ;BACKUP BY 2
6800 021170 111000          MOV      (RO),RO        ;GET RIGHT BYTE OF TRAP
6801 021172 016000 021200  MOV      $TRPAD(RO),RO ;INDEX TO TABLE
6802 021176 000200          RTS      RO             ;GO TO ROUTINE
    
```

.SBTTL TRAP TABLE

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

```

;          ROUTINE
;          -----
6812 021200          $TRPAD:
6813 021200 020026          $TYPE          ;CALL=TYPE          TRAP+0(104400) TTY TYPEOUT ROUTINE
6814 021202 020332          $TYPC          ;CALL=TYPC          TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING
6815 021204 020306          $TYPOS          ;CALL=TYPOS          TRAP+4(104404) TYPE OCTAL NUMBER (NO LEADING ZE
6816 021206 020346          $TYPON          ;CALL=TYPON          TRAP+6(104406) TYPE OCTAL NUMBER (AS PER LAST C
6817 021210 020134          $RDCHR          ;CALL=RDCHR          TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
6818 021212 020170          $RDLIN          ;CALL=RDLIN          TRAP+12(104412) TTY TYPEIN STRING ROUTINE
6819 021214 021056          $RDOCT          ;CALL=RDOCT          TRAP+14(104414) READ AN OCTAL NUMBER FROM TTY
6820 021216 005015 047522 020115 EM1:  .ASCIZ  <15><12>/ROM READ DATA COMPARISON ERROR./
        021260 005015 051127 052111 EM2:  .ASCIZ  <15><12>/WRITING ROM FAILED TO TRAP./
        021316 005015 047125 054105 EM3:  .ASCIZ  <15><12>/UNEXP TRAP WHILE READING ROM./
        021356 005015 040506 040524 EM4:  .ASCIZ  <15><12>/FATAL TRAP. ROM PC ON STACK./
        021416 005015 041520 020040 DH1:  .ASCII  <15><12>/PC          SOFT  ROM/
        021443          015 040412 042104 .ASCIZ  <15><12>/ADDRESS ADDRESS ADDRESS EXPECTED FOUND /
        021515          015 050012 004503 DH2:  .ASCII  <15><12>/PC          ROM/
        021525          015 040412 042104 .ASCIZ  <15><12>/ADDRESS ADDRESS/
        021547          015 050012 020103 DH3:  .ASCII  <15><12>/PC OF  PROGRAM /
        021571          015 052012 040522 .ASCIZ  <15><12>/TRAP  ADDRESS/
        021614          .EVEN
6821 021614 001116 001124 001126 DT1:  .WORD  $ERRPC,$GDDAT,$BDDAT,TEMP3,TEMP4,0
6822 021622 001224 001226 000000
6823
6824 021630 001116 001226 000000 DT2:  .WORD  $ERRPC,TEMP4,0
6825 021636 001116 017732 000000 DT3:  .WORD  $ERRPC,XSTORE,0
6826          021704          .=. +40
6827 021704
6828          000001          CORMAX:
        .END
    
```


PC	=%000007	393#	5996*	6290*	6411*	6469*	6473*	6513*	6520*	6525*	6697*	6741*		
PIRQ	= 177772	379#												
PIRQVE	= 000240	463#												
PRG.NO	017672	5715*	5921*	6008*	6105*	6307*	6442	6443#	6452	6462				
PRG.1	013732	5923#	5998											
PRG1	013716	5741	5749	5767	5775	5785	5793	5803	5813	5889	5921#	5997	6462	
PRG2	014444	5892	6008#	6088										
PRG3	015404	5895	6105#											
PRG4	016410	5898	6307#	6354	6412	6464								
PS	= 177776	376#	377	6441*	6451*									
PSW	= 177776	377#												
PWRVEC	= 000024	458#												
PWR.UP	017634	6428	6435#											
RDCHR	= 104410	6554	6817#											
RDLIN	= 104412	5821	5884	6109	6765	6818#								
ROCT	= 104414	6133	6154	6819#										
RESTR	012000	368	5697#	5900										
RESVEC	= 000010	453#												
RUN.3	016040	6221#	6292											
RUN3	016024	6219#	6291											
RO	=%000000	384#	5924*	5929	5931	5932	5934	5936	5937	5947	5950	5959	5965*	5968
		5970	5971	5973	5974*	5976	5986	5989	5992	6012*	6017	6018	6022	6023
		6033	6034	6038	6039	6046	6054*	6059	6060	6062	6071	6072	6080	6081
		6086*	6164*	6169	6170	6171	6184	6185	6186	6194	6207*	6221*	6226	6228
		6229	6231	6233	6234	6243	6246	6254	6259*	6263	6265	6266	6268	6269*
		6271	6280	6283	6286	6328*	6333*	6342*	6364*	6374*	6392*	6402*	6437*	6438*
		6448*	6505	6506*	6507	6510*	6713	6714*	6715*	6722*	6723*	6724*	6725*	6726*
		6727	6732	6737*	6739*	6743	6745	6762	6766*	6769	6785*	6797	6798*	6799
		6800*	6801*	6802*										
R1	=%000001	385#	6017*	6019	6033*	6035	6059*	6061	6071*	6073	6106*	6107	6128*	6134*
		6136	6137	6139	6144	6365*	6371	6378	6393*	6397	6403*	6408	6460*	6465*
		6469	6763	6767*	6771*	6773*	6775*	6778*	6781	6784*				
R2	=%000002	386#	5822*	5823*	5824	5830	5836	5842	5848	5854	5860	5866	5872	5879
		5885*	5886	6110*	6111*	6112	6115	6117	6120	6764	6768*	6772*	6774*	6776*
		6782	6783*											
R3	=%000003	387#	5886*	5887	5890	5893	5896	5928*	5944*	5967*	5981*	6013*	6022*	6038*
		6055*	6062*	6080*	6192*	6193	6194	6202	6203	6225*	6241*	6262*	6276*	6356*
		6360	6383*	6384	6550	6551*	6552	6555*	6556	6560	6562	6564*	6566*	6606
		6615*	6621*	6622*	6625*	6630*	6631*	6632	6641*					
R4	=%000004	388#	5926*	5931	5934	5935	5938	5950	5958*	6223*	6228	6231	6232	6235
		6246	6253*	6315*	6340*	6357*	6365	6381	6607	6609*	6610*	6611*	6612	6613*
		6627	6629*	6637*	6640*									
R5	=%000005	389#	6314*	6330	6351*	6352	6366*	6371	6375*	6378	6394*	6397	6404*	6408
		6608	6614*	6616*	6618*	6619*	6620*	6621	6639*					
		390#	392	5698*	5699*	5700								
R6	=%000006	391#	393											
R7	=%000007	556#												
SAVR0	001230	557#												
SAVR1	001232	558#												
SAVR4	001234	559#												
SAVR5	001236	392#	5702*	5712*	5722*	5726	5729*	5733	5735	5736	5753*	5757	5759	5760
SP	=%000006	5822	5885	5978*	6026*	6031*	6043*	6064*	6068*	6077*	6110	6127*	6134	6141*
		6155	6174*	6179*	6188*	6273*	6309*	6321*	6332*	6338*	6346*	6362*	6424*	6429
		6431	6432	6436*	6445	6446	6448	6450*	6505*	6506	6507*	6509	6510	6511*

.SSCOP	10	3440	6650
.SSIZE	10		
.SSUPR	10		
.STRAP	10	3430	6788
.STYPB	10		
.STYPD	10		
.STYPE	10	64780	6479
.STYPO	10	64780	6574

ADD	6011	6438	6511	6601	6611	6726	6778								
ASL	6723	6724	6725	6771	6773	6775									
BEQ	5940	5948	5954	5956	5985	5987	5990	6047	6053	6082	6113	6116	6121	6195	6201
	6204	6237	6244	6249	6251	6279	6281	6284	6372	6379	6389	6398	6409	6447	6461
	6466	6628	6689	6728	6733	6746	6770								
BGT	6635														
BIC	6541	6625	6777												
BIS	6630	6631													
BISB	6715														
BIT	5942	5953	5979	5984	6052	6200	6239	6248	6274	6278	6446	6695			
BLO	6138														
BLOS	6553														
BLT	6519	6636													
BMI	6660														
BNE	5701	5720	5728	5744	5752	5756	5770	5778	5780	5788	5796	5798	5806	5808	5818
	5825	5831	5837	5843	5849	5855	5861	5867	5873	5880	5888	5891	5894	5897	5930
	5933	5943	5945	5969	5972	5980	5982	5995	6021	6037	6049	6084	6118	6197	6227
	6230	6240	6242	6264	6267	6275	6277	6289	6312	6353	6385	6387	6439	6463	6508
	6515	6557	6563	6626	6696	6716	6738								
BPL	6502	6523	6539	6624	6700										
BR	5734	5758	5761	5882	5909	5951	5960	5988	5993	6050	6085	6123	6198	6209	6247
	6255	6282	6287	6504	6521	6559	6602	6617	6638	6662	6668	6671	6721	6748	6779
CLC	6350	6382													
CLR	5699	5710	5711	5713	5714	5717	5718	6086	6308	6329	6359	6391	6401	6423	6437
	6441	6451	6457	6615	6714	6767	6768								
CLRB	6564	6675													
CMP	5700	5727	5735	5743	5751	5755	5759	5769	5777	5779	5787	5795	5797	5805	5807
	5824	5830	5836	5842	5848	5854	5860	5866	5872	5887	5890	5893	5896	5929	5931
	5932	5934	5939	5947	5950	5955	5968	5971	5986	5989	6020	6036	6046	6081	6112
	6115	6117	6120	6137	6194	6203	6226	6228	6229	6231	6236	6243	6246	6250	6263
	6266	6280	6283	6352	6371	6378	6384	6397	6408	6462	6552	6669			
CMPB	6388	6514	6556	6562											
COM	5819	6313													
DEC	5944	5981	5994	6048	6083	6196	6241	6276	6288	6386	6722				
DECB	6518	6623	6634												
EMT	374														
HALT	364	5816	5908	6087	6430	6433	6503	6701							
INC	6349	6629	6637	6691											
INCB	6673	6688													
IOT	375														
JMP	368	5721	5741	5749	5767	5775	5785	5793	5803	5813	5889	5892	5895	5898	5900
	5997	5998	6088	6119	6145	6147	6156	6210	6291	6292	6354	6412	6442	6452	6464
JSR	5996	6290	6411	6469	6513	6520	6697								
MOV	5698	5702	5703	5704	5705	5706	5707	5708	5709	5712	5715	5716	5722	5723	5726
	5729	5730	5733	5736	5737	5738	5739	5740	5745	5746	5747	5748	5753	5754	5757
	5760	5763	5764	5765	5766	5771	5772	5773	5774	5781	5782	5783	5784	5789	5790
	5791	5792	5799	5800	5801	5802	5809	5810	5811	5812	5822	5823	5826	5827	5828
	5832	5833	5834	5838	5839	5840	5844	5845	5846	5850	5851	5852	5856	5857	5858
	5862	5863	5864	5868	5869	5870	5874	5875	5876	5878	5885	5886	5921	5922	5923
	5924	5925	5926	5927	5928	5935	5936	5937	5938	5957	5958	5964	5965	5966	5967
	5974	5976	5978	6008	6009	6010	6012	6013	6016	6017	6018	6019	6022	6026	6031
	6033	6034	6035	6038	6043	6054	6055	6058	6059	6060	6061	6062	6064	6068	6071
	6072	6073	6077	6080	6105	6106	6107	6110	6111	6125	6127	6128	6134	6139	6141
	6144	6155	6164	6165	6168	6169	6170	6174	6179	6184	6185	6198	6192	6207	6208

.MCALL	343	344	464	6478											
.NLIST	1	343	364	464	503	561	5689	5698	5910	6101	6089	6419	6421	6543	6804
	6813	6814	6815	6816	6817	6818	6819	6820	6828						
.PAGE	464	506	748	1120	1591	1642	1689	1752	1797	1848	1918	1972	2025	2079	2126
	2178	2231	3270	3839	5693	5961	6256	6419							
.REM	1	2032	2082												
.REPT	364														
.SBTTL	343	358	365	370	466	508	561	6481	6528	6576	6652	6681	6705	6753	6790
	6805														
.TITLE	345														
.WORD	364	475	480	483	484	485	486	489	490	491	492	493	494	6730	6735
	6787	6821	6824	6825											

ERRORS DETECTED: 0

*.DZBMDJ.SEQ/SOL/CRF/NL:TOC=DZBMD.SML,DZBMD.P11
RUN-TIME: 13 20 1 SECONDS
CORE USED: 20K

10			...B1	2345			...B5				...B9	5602			...B13
62P111-DZBMD-J			...C1	2399			...C5	4121			...C9				...C13
108			...D1	2446			...D5	4175			...D9	5664	010760	177766	...D13
148			...E1	2484			...E5	4185	007240	001775	...E9	5684			...E13
			...F1	2538			...F5	4197			...F9				...F13
			...G1	2591			...G5	4227			...G9	5702	012014	012706	...G13
204			...H1	2645			...H5				...H9	5756	012374	001004	...H13
233			...I1	2679			...I5	4285	007350	177776	...I9	5810	012736	012767	...I13
273			...J1	2697	005006	177570	...J5	4317			...J9	5864	013260	012767	...J13
325			...K1				...K5				...K9				...K13
			...L1	2757			...L5	4376	007444	004000	...L9	5920			...L13
352			...M1	2803	005102	000020	...M5				...M9	5970	014162	005720	...M13
406		000040	...N1	2816			...N5	4438			...N9	6011	014466	062767	...N13
			...B2	2861	005212	012711	...B6	4492	007634	001770	...B10	6065	014734	104402	...B14
460		000034	...C2	2913	005312	100316	...C6	4502			...C10	6099			...C14
473	000046	020002	...D2	2952			...D6				...D10	6153			...D14
515			...E2	2981			...E6	4565	007732	177762	...E10	6207	016002	016700	...E14
569			...F2				...F6	4582			...F10				...F14
623	001466	060011	...G2	3038	005426	000012	...G6	4622			...G10	6265	016262	005720	...G14
677			...H2	3091			...H6	4644			...H10	6304			...H14
731	001740	000024	...I2	3122			...I6				...I10	6358	016666	012737	...I14
757		:ACTUAL	...J2	3166			...J6	4700			...J10	6412	017200	000167	...J14
811	002004	113737	...K2	3188	005674	010045	...K6				...K10		017451	015	...K14
865	002132	000200	...L2	3223	005724	005061	...L6	4756			...L10	6432	017626	012637	...L14
919	002246	012700	...M2	3262	005776	000000	...M6	4795			...M10	6484			...M14
973	002376	000005	...N2	3279			...N6				...N10	6538	020144	105777	...N14
1027	002522	.BYTE													
			...B3				...B7	4855			...B11	6592			...B15
1081	002664	100376	...C3	3340			...C7				...C11	6646	020527	000	...C15
1129		:173000	...D3	3366			...D7	4917			...D11	6700	020714	100001	...D15
1183	003062	100000	...E3	3420	006076	001763	...E7	4928			...E11	6754			...E15
1237	003212	177560	...F3	3433	006122	005205	...F7				...F11	6808			...F15
1291	003334	010742	...G3	3487	006236	042700	...G7	4986	010116	000005	...G11	BIT05 =	000040		...G15
1345			...H3	3502	006260	005000	...H7				...H11	EXTMAP =	013464		...H15
1399			...I3	3556			...I7	5049	010222	000666	...I11	PS =	177776		...I15
1453	003472	012761	...J3	3571	006420	000000	...J7	5080	010232	000040	...J11	SRV.R =	016024		...J15
1507			...K3				...K7	5094			...K11	TYPON =	104406		...K15
1561	003744	000207	...L3	3627	006506	000000	...L7	5127			...L11	\$TPFLG	001151		...L15
1600			...M3	3663			...M7				...M11	SETTRA	6804#	6814	...M15
1651	004004	000001	...N3	3699			...N7	5187			...N11				...N15
1698															
			...B4				...B8	5232			...B12	BIT	5942	5953	...B16
1761	004226	000340	...C4	3761	006672	010145	...C8	5251			...C12	MOV8	6507	6524	...C16
1806			...D4	3797			...D8				...D12	.TITLE	345	358	...D16
1857	004350	032711	...E4	3848			...E8	5310			...E12				
1911	004520	030500	...F4	3875			...F8	5364			...F12				
1927			...G4				...G8				...G12				
1981			...H4	3931			...H8	5387			...H12				
2034			...I4				...I8				...I12				
2088			...J4				...J8	5447	010574	005004	...J12				
2135	004534	010037	...K4	3993	007004	013700	...K8	5483			...K12				
2187			...L4	4047	007064	173304	...L8				...L12				
2240			...M4				...M8	5539			...M12				
2291			...N4	4063			...N8				...N12				