BM873

RESTART ROM LOADER

MD-11-DZBMD-H

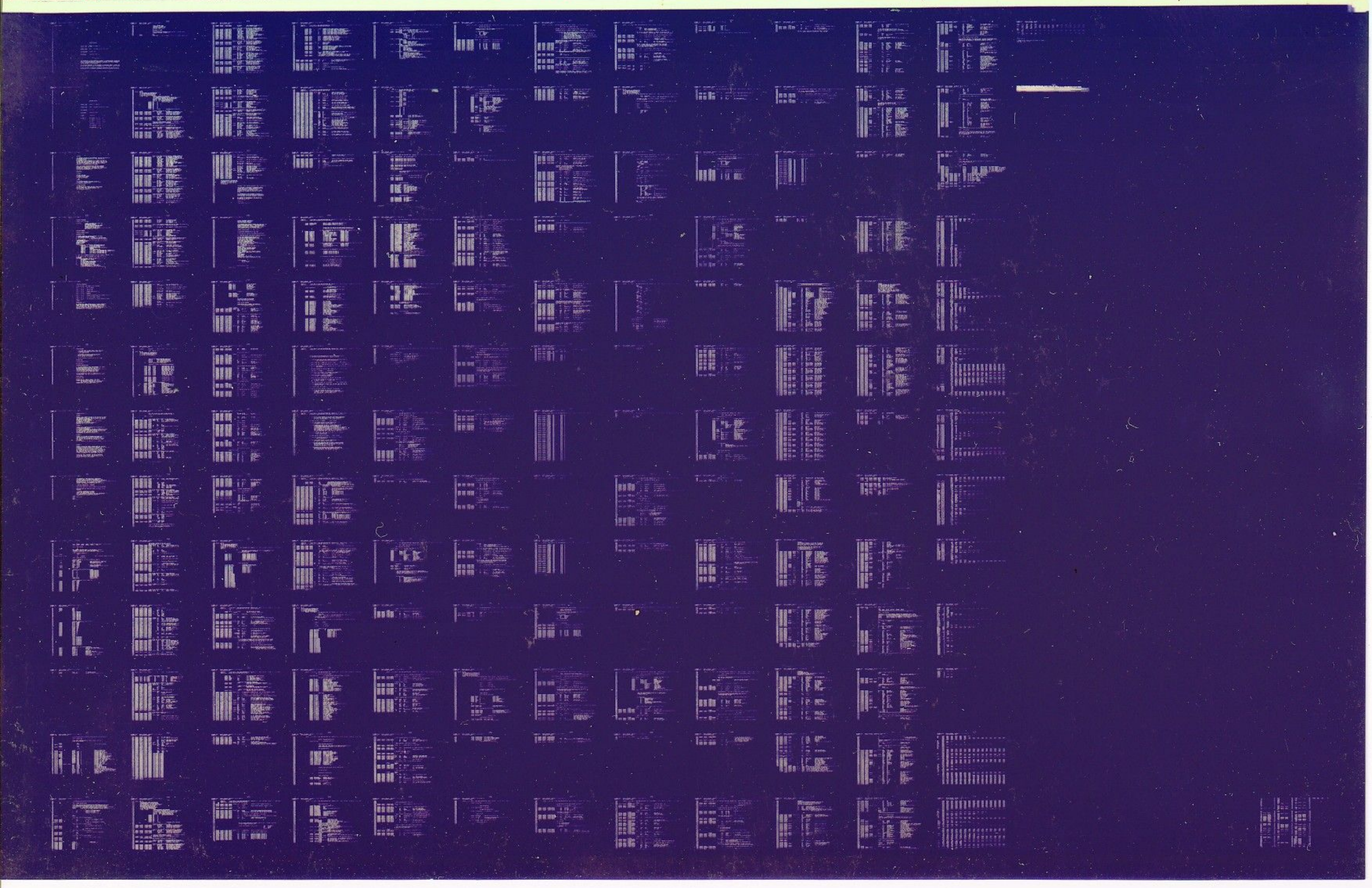
EP DZBMD H DL B

FICHE 1 OF 1

MAR 1977

digital

MADE IN USA



```
  1  000000  000000                .

                                   .REM \
```

IDENTIFICATION

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

PRODUCT NAME: BM873 - UNIVERSAL RESTART ROM LOADER

DATE RELEASED: JANUARY 1977

MAINTAINER: DIAGNOSTIC GROUP

```
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
```

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 FITZCARL JOHNSON	YH

```
                                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 BM873 ARE PROGRAMS
                                        ONE AND FOUR. PROGRAM TWO IS NECESSARY FOR "NON-STANDARD"
                                        SETUPS AND IS A MAINTAINCE TOOL. PROGRAM THREE IS
                                        ALSO JUST FOR MAINTAINCE AID.

                                2.      REQUIRMENTS

                                2.1     EQUIPMENT

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

                                2.2     STORAGE

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

                                3.      LOADING PROCEDURE

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

```
                                4.      STARTING PROCEDURE

                                4.1     CONTROL SWITCH SETTINGS

                                        SWITCH 00       CLEARED INDICATES ONLY FIRST 128
                                                        WORDS TO BE CHECKED.
                                                        SET INDICATES EXTENDED 128. WORDS
                                                        ARE TO BE CHECKED IN WHICH CASE
                                                        PROGRAM 2 MUST BE RUN FIRST.
                                                        WHEN RUNNING ON BM873Y-B,C,D,F,G OR H, 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)
                                        T                       3. PRESS START SWITCH AND RELEASE.
                                        4. DEVICE VERSION.
                                         WHEN PROGRAM IS STARTED FOR THE FIRST TIME THE FOLLOWING
                                         WILL BE PRINTED OUT:
                                        MAINDEC-11-DZBMDG
                                        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

                                        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 OR H)
                                        1.      LOAD STARTING ADD. 000200
                                        2.      SET SW00=1
                                        3.      SET HALT ENABLE SW AND SINGLE CYCLE SW UP
                                        4.      HIT START SWITCH AND RELEASE.
                                        5.      RUN PROGRAM 2 FOR ONE PASS.
                                        6.      NOW ANY PROGRAM MAY BE RUN.
                                        NOTE:   VISUAL INSPECTION OF EXTENDED DUMP
                                                IS YOUR RESPONSIBILITY. THAT DATA WAS
                                                PLACED INTO SOFTWARE TABLE FOR TEST COMPARISON.
```

5. OPERATING PROCEDURE

5.1.1 SWITCH SETTINGS (APPLICABLE IN ALL PROGRAMS)

SW15 = 1 OR UP ... HALT ON ERROR

SW14 = 1 OR UP ... LOOP ON TEST

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

SW12 = 1 OR UP ... 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 ERRCR

6. ERRORS

6.1 ERROR PRINT OUT

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

```
                                6.2     ERROR RECOVERY

                                        1. ITS A GOOD IDEA TO LEAVE SW15=1 WHILE TEST RUNS
                                           TO PREVENT A RUN AWAY ERROR FROM GOING WILD
                                           IF YOU LEAVE THE CPU.

                                        2. IN AN ERROR; SET SW14=1(LOOP ON THIS ADDR.)
                                           AND SET SW 13=1(DELETE ERROR PRINT OUT).
                                IF CPU IS HALTED; HIT CONTINUE.

                                        3. NOW THE PROGRAM IS RUNNING AND YOU MAY SCOPE IT.

                                7.      RESTRICTIONS

                                7.1     STARTING RESTRICTIONS

                                        SEE SECTION 4.

                                7.2     OPERATING RESTRICTIONS

                                7.2.1   IF YOU WISH PROGRAM TO TEST YOUR EXTENDED 128. WORDS;
                                        YOU MUST START AS PER SECTION 4 AND THEN
                                        ****** RUN PROGRAM 2 FIRST AND VISUALLY VERIFY DATA.****
                                        (NOT APPLICABLE TO BM873Y-B,C,D,F,G OR H)
                                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 OR H)

                                8.      MISCELLANEOUS

                                8.1     EXECUTION TIME

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

9. PROGRAM DESCRIPTION

9.1 PROGRAM 1

PROGRAM 1 WILL VERIFY THE DATA IN THE ROM AND THE VERIFY
THAT WRITING THE ROM WILL TRAP OUT (EXCEPT THE VECTORS)
EACH ADDRESS IS REFERENCED FIVE TIMES IN A ROW BEFORE
UPDATING TO THE NEXT ADDRESS.
IF SW00 WAS UP WHEN START WAS HIT, THE EXTENDED
128 WORDS WILL BE CHECKED.
256 WORDS WILL BE CHECKED AUTOMATICALLY IF
BM873Y-B,C,D,F,G OR H 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 BM873Y-B,C,D,F,G OR H 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.

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

                                9.6     ELECTRICAL PREQUISITES  (HARDWARE)

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

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

                                9.7.3    IF FURTHER INFORMATION IS NEEDED
                                        CONSULT THE BM873 MANUAL FOR HELP.
                                        NOTE: THE DIAGNOSTIC RUNNING WITHOUT ANY INTERFERANCE 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
```

```
                                        \
                                .MCALL .HEADER, .SWRHI, .SWRLO, .EQUATE, .SETUP, .$TRAP, .$CATCH, .$CMTAG
                                .MCALL .RDLIN, .$SCOPE, .$ERROR, .$ERRTYP, .$RDOCT

                                .SBTTL  TRAP CATCHER

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

                                .SBTTL  STARTING ADDRESS(ES)
000200                                  .=200

000200  000137  011000                  JMP     @#RESTRT                ;JUMP TO STARTING ADDRESS OF PROGRAM

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR               ;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE               ;BASIC DEFINITION OF SCOPE CALL
177776                          PS=     177776                  ;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774                  ;STACK LIMIT REGISTER
177772                          PIRQ=   177772                  ;PROGRAM INTERRUPT REQUEST REGISTER
177570                          SWR=    177570                  ;SWITCH REGISTER
177570                          DISPLAY=SWR

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0                      ;GENERAL REGISTER
000001                          R1=     %1                      ;GENERAL REGISTER
000002                          R2=     %2                      ;GENERAL REGISTER
000003                          R3=     %3                      ;GENERAL REGISTER
000004                          R4=     %4                      ;GENERAL REGISTER
000005                          R5=     %5                      ;GENERAL REGISTER
000006                          R6=     %6                      ;GENERAL REGISTER
000007                          R7=     %7                      ;GENERAL REGISTER
                                .EQUIV  R6,SP                   ;STACK POINTER
                                .EQUIV  R7,PC                   ;PROGRAM COUNTER

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

```
                    000010          SW03=   10
                    000004          SW02=   4
                    000002          SW01=   2
                    000001          SW00=   1
                                    .EQUIV  SW09,SW9
                                    .EQUIV  SW08,SW8
                                    .EQUIV  SW07,SW7
                                    .EQUIV  SW06,SW6
                                    .EQUIV  SW05,SW5
                                    .EQUIV  SW04,SW4
                                    .EQUIV  SW03,SW3
                                    .EQUIV  SW02,SW2
                                    .EQUIV  SW01,SW1
                                    .EQUIV  SW00,SW0

                                    ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                    100000          BIT15=  100000
                    040000          BIT14=  40000
                    020000          BIT13=  20000
                    010000          BIT12=  10000
                    004000          BIT11=  4000
                    002000          BIT10=  2000
                    001000          BIT09=  1000
                    000400          BIT08=  400
                    000200          BIT07=  200
                    000100          BIT06=  100
                    000040          BIT05=  40
                    000020          BIT04=  20
                    000010          BIT03=  10
                    000004          BIT02=  4
                    000002          BIT01=  2
                    000001          BIT00=  1
                                    .EQUIV  BIT09,BIT9
                                    .EQUIV  BIT08,BIT8
                                    .EQUIV  BIT07,BIT7
                                    .EQUIV  BIT06,BIT6
                                    .EQUIV  BIT05,BIT5
                                    .EQUIV  BIT04,BIT4
                                    .EQUIV  BIT03,BIT3
                                    .EQUIV  BIT02,BIT2
                                    .EQUIV  BIT01,BIT1
                                    .EQUIV  BIT00,BIT0

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

```
444  000064  TPVEC=  64  ;TTY PRINTER VECTOR
445  000240  PIRQVEC=240  ;PROGRAM INTERRUPT REQUEST VECTOR
```

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

                        .SBTTL  COMMON TAGS

                        ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                        ;*USED IN THE PROGRAM.

        000046                  .=46
000046  016640                  $ENDAD                          ;LOGICAL END OF PROGRAM

        000052                  .=52
000052  000000                  .WORD   0

        001100                  .=1100

001100                  $CMTAG:                                 ;START OF COMMON TAGS
001100  000000          $PASS:  .WORD   0                       ;CONTAINS PASS COUNT
001102     000          $TSTNM: .BYTE   0                       ;CONTAINS THE TEST NUMBER
001103     000          $ERFLG: .BYTE   0                       ;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0                       ;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0                       ;CONTAINS SCOPE LOOP
001110  000000          $LPERR: .WORD   0                       ;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0                       ;CONTAINS TOTAL ERRORS DETECTED
001114     000          $ITEMB: .BYTE   0                       ;CONTAINS ITEM CONTROL BYTE
001115     001          $ERMAX: .BYTE   1                       ;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0                       ;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0                       ;CONTAINS  OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0                       ;CONTAINS  OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0                       ;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0                       ;CONTAINS 'BAD' DATA
001130  000000  000000  000000          .WORD   0,0,0           ;RESERVED--NOT TO BE USED
001136  177560          $TKS:   177560                          ;TTY KBD STATUS
001140  177562          $TKB:   177562                          ;TTY KBD BUFFER
001142  177564          $TPS:   177564                          ;TTY PRINTER STATUS REG.
001144  177566          $TPB:   177566                          ;TTY PRINTER BUFFER REG.
001146     000          $NULL:  .BYTE   0                       ;CONTAINS NULL CHARACTER FOR FILLS
001147     002          $FILLS: .BYTE   2                       ;CONTAINS # OF FILLER CHARACTERS REQUIRED
001150     012          $FILLC: .BYTE   12                      ;INSERT FILL CHARS. AFTER A "LINE FEED"
001151     000          $TPFLG: .BYTE   0                       ;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001152     077          $QUES:  .ASCII  /?/                     ;QUESTION MARK
001153     015          $CRLF:  .ASCII  <15>                    ;CARRIAGE RETURN
001154  000012          $LF:    .ASCIZ  <12>                    ;LINE FEED
```

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

                        .SBTTL  ERROR POINTER TABLE

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

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


001156                  $ERRTB:

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 0

001156  020054                 EM1             ;"ROM READ DATA COMPARISON ERROR"
001160  020254                 DH1             ;*
001162  020452                 DT1             ;*
001164  000000                 0               ;* PRINT ALL NUMERIC DATA IN OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 1

001166  020116                 EM2             ;"WRITTING ROM FAILED TO TRAP"
001170  020353                 DH2
001172  020466                 DT2
001174  000000                 0               ;PRINT ALL NUMERIC DATA IN OCTAL.

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 2

001176  020154                 EM3             ;"UNEXPECTED TRAP WHILE READING ROM"
001200  020405                 DH3
001202  020474                 DT3
001204  000000                 0
                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 3

001206  020214                 EM4             ;"FATAL TRAP. ROM PC ON STACK."
001210  020353                 DH2
001212  020466                 DT2
001214  000000                 0

001216  000000          LSTERR: 0               ;ERROR FLAG
001220  000000          ICOUNT: 0               ;ITERATION COUNT.
001222  000000          TEMP5:  0
001224  000000          TEMP3:  0
001226  000000          TEMP4:  0
001230  000000          SAVR0:  0
001232  000000          SAVR1:  0
001234  000000          SAVR4:  0
001236  000000          SAVR5:  0
```

```
                                .TITLE  OCTOBER 1976
                                ;*COPYRIGHT (C) BM873 YX
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY DZBMD
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-A1).
                                ;*
000001                          $TN=1
160000                          $SWR=160000     ;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT
```

```
                        .=1400
001400                  MAP.YA:
                        ;THE FOLLOWING IS A REPRODUCTION
                        ;OF THE ROM PROGRAM FOR BM873YA.
                        ;IT IS HERE FOR COMPARISON TO
                        ;ACTUAL ROM AND FOR REFERENCE.
                        ;173000 .=173000        ;STARTING ADDRESS FOR BOOTSTRAP
                                                ;THIS LOADER IS DESIGNED FOR THE RESTART MODULE M873.
                                                ;IT FUNCTIONALLY REPLACES THE FOLLOWING ROMS:
                                                ;M792-YA - PAPER TAPE BOOTSTRAP FOR PC11,KL11
                                                ;MR11-DB   BULK STORAGE BOOTSTRAP ROM
                                                ;M792-YH   TA11 CASSETTE BOOTSTRAP ROM
                                000000  R0=     %0      ;REGISTER DEFINITIONS
                        ;               000001  R1=     %1
                        ;               000002  R2=     %2
                        ;               000003  R3=     %3
                        ;               000004  R4=     %4
                        ;               000005  R5=     %5
                        ;               000006  SP=     %6
                        ;               000007  PC=     %7
                        ;               177570  SR=     177570          ;PROCESSOR SWITCH REGISTER
                                                ;
                                                ;STARTING LOCATION FOR RF11 DISK
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
                                                ;
                                                ;THIS IS THE STARTING LOCATION FOR THE RK11 CONTROLLER
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
                                                ;
                                                ;THIS IS A SPARE STARTING LOCATION. IT TRANSFERS TO ADDRESS
                                                ;CONTAINED IN THE SWITCH REGISTER.
001420  013707  ;173020 013707  TRANSR: MOV @#SR,PC     ;GO TO INDICATED LOCATION
001422  177570  ;173022 177570
                        ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
                                                ;
                                                ;THIS IS THE POWER UP VECTOR REQUIRED FOR DEVICE AND
001424  173000  ;173024 173000  POWER:  .WORD RF11       ;ADDRESS OF FIRST LOCATION IN ROM
001426  000340  ;173026 000340          .WORD 340       ;PROCESSOR STATUS LEVEL 7
                                                ;
                                                ;THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE) CONTROLLER.
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
```

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

```
                                        BPL OTHWAT      ;BY HANGING IN LOOP
                                        TST @R1         ;THEN TEST FOR ERROR
                                        BMI ERROR       ;GOT PROBLEMS
                                        CLR PC          ;AND TRANSFER TO ZERO
                                ;
                                ;THIS IS THE STARTING ADDRESS FOR THE PC11 PAPER TAPE CONTROLLER
                                KL11:   MOV #177560,R4  ;OBTAIN DEVICE ADDRESS
                                        BR CKDEV        ;AND TRANSFER TO READER SERVICE ROUTINE
                                ;
                                ;THIS IS THE CASSETTE DEVICE COMMAND TABLE
                                TABLE:  .BYTE 240       ;COMPARE WORD NOT A COMMAND
                                        .BYTE 37        ;ILBS+RWD+GO
                                        .BYTE 15        ;SPACE FORWARD BLOCK+GO
                                        .BYTE 5         ;READ+GO
                                        .BYTE 24        ;READ+ILBS
                                        .BYTE 224       ;READ+ILBS+END FLAG
                        ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
                                ;
                                ;THIS IS AN ADDITIONAL POWER VECTOR ADDRESS REQUIRED BY DEVICE
                                POWER2: .WORD RF11      ;ADDRESS OF BEGINNING OF BOOTSTRAP
                                        .WORD 340       ;PRIORITY LEVEL 7
                                ;
                                ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE DEVICE #0
                                CBOOT:  CLR R4          ;LOAD DEVICE NUMBER 0 IN R4
                                RESTX:  MOV #177500,R0  ;GET DEVICE ADDTESS
                                RESTRT: RESET           ;ISSUE RESET INSTRUCTION
                                        MOV R4,@R0      ;LOAD DEVICE WITH UNIT NUMBER
                                        MOV #TABLE,R1   ;GET FUNNY TABLE OF INSTRUCTIONS

                                        MOV #375,R2     ;AND LOAD UP TRANSFER COUNTER

                                        MOVB (R1)+,R3   ;THE LOAD UP COMPARATOR
                                LOOP1:  MOVB (R1)+,@R0  ;LOAD DEVICE REGISTER WITH COMMAND
                                        BMI DONE
                                LOOP2:  BITB R3,@R0     ;HAS COMMAND COMPLETED
                                        BEQ LOOP2       ;NO, WAIT
                                        INCB R2         ;THEN INCREMENT ADDRESS CTR
                                        BMI LOOP1       ;IF NEGATIVE, GET COMMAND
                                        MOVB 2(R0),@R2  ;AND STORE DATA AWAY

                                        BR LOOP2        ;GO GET ANOTHER BYTE
                                DONE:   TST @R0         ;ANY DEVICE ERRORS
                                        BMI RESTRT      ;YES, RETRY
                                        CLR R2          ;CLEAR COMPARE ADDRESS AND TRANSFER ADDRESS
                                        CMPB R3,@R2     ;IT MUST BE 240
                                        BNE .+0         ;NO, THERE WAS AN ERROR
                                ERROR:  JMP @R2         ;NORMAL CASSETTE AND ERROR FOR BULK STORAGE
                                ;
                                ;THIS IS THE STARTING LOCATION FOR THE PC11 CONTROLLER
                                PC11:   MOV #177550,R4  ;LOAD DEVICE ADDRESS
```

```
  716  001716  000005  ;173316 000005  CKDEV:  RESET              ;KILL ALL DEVICE ACTION
  717  001720  012701  ;173320 012701          MOV #160000,R1     ;THEN SET UP MEMORY TEST LIMITS
  718  001722  160000  ;173322 160000
  719  001724  012702  ;173324 012702          MOV #6,R2          ;AND SET UP POINTER TO TIMEOUT LOCATION
  720  001726  000006  ;173326 000006
  721  001730  012712  ;173330 012712          MOV #340,@R2       ;AND SET UP VECTOR TO RETURN TO NEXT
  722  001732  000340  ;173332 000340
  723  001734  010742  ;173334 010742          MOV PC,-(R2)       ;SAVE THE PC
  724  001736  012706  ;173336 012706          MOV #24,SP         ;AND LOAD UP STACK POINTER
  725  001740  000024  ;173340 000024
  726  001742  010441  ;173342 010441          MOV R4,-(R1)       ;AND LOOK FOR END OF MEMORY
  727  001744  040601  ;173344 040601          BIC SP,R1          ;THEN DROP TO XX7752
  728  001746  010111  ;173346 010111          MOV R1,@R1         ;AND STORE IN ITSELF
  729  001750  011102  ;173350 011102  LOOP:   MOV @R1,R2         ;THEN LOAD ADDRESS FOR DATA INSERTION
  730  001752  005214  ;173352 005214          INC @R4            ;AND START DEVICE
  731  001754  105714  ;173354 105714  RDRWAT: TSTB @R4           ;THEN WAIT FOR CHARACTER AVAILABLE
  732  001756  100376  ;173356 100376          BPL RDRWAT         ;HANGING THERE IF NECCESSARY
  733  001760  116412  ;173360 116412          MOVB 2(R4),@R2     ;STORE AWAY DATA BYTE
  734  001762  000002  ;173362 000002
  735  001764  005211  ;173364 005211          INC @R1
  736  001766  120227  ;173366 120227          CMPB R2,#375       ;HAS BRANCH OFFSET BEEN STORED
  737  001770  000375  ;173370 000375
  738  001772  001366  ;173372 001366          BNE LOOP           ;NO
  739  001774  105222  ;173374 105222          INCB (R2)+         ;YES, ALL DONE
  740  001776  END.YA:
  741  001776  000142  ;173376 000142          JMP -(R2)          ;THEN TRANSFER TO RTN
```

```
                        ; BM873B        BOOTSTRAP       MACY11 27(655)  1-OCT-74  14:50  PAGE 1
                        ;
                        ;               ;DATE:  AUG 23, 1974
002000                  MAP.YB:
                        ;THE FOLLOWING IS A REPRODUCTION
                        ;OF THE ROM PROGRAM FOR BM873YB.
                        ;IT IS HERE FOR COMPARISON TO THE
                        ;ACTUAL ROM AND FOR REFERENCE
                        ;
                        ;
                        ;
                        ;               ;THIS IS THE LOADER TO REPLACE THE FOLLOW
                        ;               ;M792-YA        PAPER TAPE BOOTSTRAP ROM
                        ;               ;MR11-DB        BULK STORAGE BOOTSTRAP ROM
                        ;               ;M792-YH        TA11 CASSETTE BOOTSTRAP ROM
                        ;               ;RM873A COMBINATION OF ABOVE ROMS
                        ;
                        ;               ;PREPHERIAL EXTERNAL PAGE REGISTERS ASSIGNMENTS:
                        ;
                        ;       177462  RFWC=   177462          ;WORD COUNT REG. FOR RF1
                        ;       177406  RKWC=   177406          ;WORD COUNT REG. FOR RK1
                        ;       177344  TCWC=   177344          ;WORD COUNT REG. FOR TC1
                        ;       172524  TMWC=   172524          ;BYTE/RECORD COUNT FOR T
                        ;       176716  RPWC=   176716          ;WORD COUNT REG. FOR RP1
                        ;       177450  RCWC=   177450          ;WORD COUNT REG. FOR RC1
                        ;       177560  KLCS=   177560          ;CONTROL REG. FOR KL11
                        ;       177500  TACS=   177500          ;CONTROL REG. FOR TA11 C
                        ;       177550  PCCS=   177550          ;CONTROL REG. FOR PC11
                        ;       172440  TUCS=   172440          ;CONTROL STATUS REG. 1
                        ;       172442  TUWC=   TUCS+2          ;TU16 WORD COUNT REG.
                        ;
                        ;       176300  RHCSA=  176300          ;CONTROLLER REG. 1 FOR R
                        ;       176302  RHWCA=  RHCSA+2
                        ;       172040  RSCSA=  172040          ;CONTROLLER REG.1 FOR RH
                        ;       172042  RSWCA=  RSCSA+2
                        ;       176700  RPCSA=  176700          ;CONTROLLER REG. 1 FOR R
                        ;       176702  RPWCA=  RPCSA+2
                        ;               ;FUNCTION VALUE FOR PREPHERALS:
                        ;       000005  RFREAD= 5               ;READ FUNCTION
                        ;       004003  RNUM=   4003            ;REVERSE AND IDENTIFY BL
                        ;       060017  TMRWND= 60017           ;REWIND AND SET 800 BPI
                        ;       060011  TMFWRD= 60011           ;FORWARD RECORD COMMAND
                        ;       060003  TMREAD= 60003           ;TM11 READ
                        ;       000011  DRCLR=  11              ;DRIVE CLEAR
                        ;       000071  RHREAD= 71              ;RH11 READ COMMAND
                        ;       000021  RHPRST= 21              ;READ IN PRESET
                        ;       000031  TUSPAC= 31              ;SPACE FORWARD COMMAND F
                        ;       040000  TUTAPE= 40000           ;TAPE BIT IN RH11/RHDT R
                        ;       001300  TUMODE= 1300            ;800 BPI NORMAL MODE FOR
                        ;       001000  FCE=    1000            ;FRAME COUNT ERROR BIT
                        ;               ;CONSOLE SWITCH REG.
                        ;       177570  CSW=    177570
                        ;
                        ;
```

```
                        ;               ;ONLY THE LOW BYTE OF CONSOL SWITCH REGISTER IS
                        ;               ;SELECT THE UNIT NUMBER OF THE DEVICE TO BOOT FR
                        ;
                        ;       173000  .=173000
                        ;
                        ;
                        ;               ;THIS IS THE STARTING ADDRESS FOR RH11/RS03/04 D
002000  000405 ;173000  000405  RHRSA:  BR      1$      ;ENTRY FOR SELECTING UNI
002002  010703 ;173002  010703  RHRSB:  MOV     PC,R3   ;ENTRY TO SELECT UNITS
002004  113737 ;173004  113737          MOVB    @#CSW,@#RSCSA+10;LOAD UNIT # INS
002006  177570 ;173006  177570
002010  172050 ;173010  172050
002012  000401 ;173012  000401          BR      2$
002014  010703 ;173014  010703  1$:     MOV     PC,R3
002016  012700 ;173016  012700  2$:     MOV     #RSCSA,R0;SET CONTROL STATUS REG
002020  172040 ;173020  172040
002022  000526 ;173022  000526          BR      RHCOMN
                        ;               ;
                        ;               ;THIS IS THE AUTO LOAD VECTOR
002024  173000 ;173024  173000          .WORD   RHRSA
002026  000340 ;173026  000340          .WORD   340
                        ;               ;
                        ;               ;THIS IS THE STARTING ADDRESS FOR RK11 CONTROLLE
002030  000412 ;173030  000412  RK11A:  BR      2$      ;ENTRY TO SELECT UNIT 0
002032  010703 ;173032  010703  RK11B:  MOV     PC,R3   ;ENTRY TO SELECT ALL UNI
                        ;                               ;SAVE ERROR RETRY ADDRES
002034  113705 ;173034  113705          MOVB    @#CSW,R5;SET POINTER TO PARAMETE
002036  177570 ;173036  177570
002040  052705 ;173040  052705          BIS     #10,R5  ;SET POSITION BIT
002042  000010 ;173042  000010
002044  006105 ;173044  006105  1$:     ROL     R5      ;SHIFT UNIT # TO BIT 13-
002046  103376 ;173046  103376          BCC     1$      ;KEEP GOING
002050  010537 ;173050  010537          MOV     R5,@#RKWC+4;MOVE IN TO RKDA REGI
002052  177412 ;173052  177412
002054  000401 ;173054  000401          BR      3$      ;SKIP NEXT INSTRUCTION
002056  010703 ;173056  010703  2$:     MOV     PC,R3   ;SAVE ERROR RETRY ADDRES
002060  010702 ;173060  010702  3$:     MOV     PC,R2
002062  000546 ;173062  000546          BR      OTHERA
002064  177406 ;173064  177406          .WORD   RKWC
002066  000005 ;173066  000005          .WORD   RFREAD
                        ;               ;
                        ;               ;THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE)
002070  010703 ;173070  010703  TC11:   MOV     PC,R3   ;SAVE ERROR RETRY ADDRES
002072  010702 ;173072  010702          MOV     PC,R2
002074  000570 ;173074  000570          BR      TAPES
002076  177344 ;173076  177344          .WORD   TCWC
002100  000005 ;173100  000005          .WORD   RFREAD
002102  004003 ;173102  004003          .WORD   RNUM
002104  100000 ;173104  100000          .WORD   100000  ;DONE MASK
002106  024000 ;173106  024000          .WORD   24000   ;ERROR MASK
                        ;
                        ;
                        ;               ;
                        ;               ;TM11 STARTING ADDRESS
```

```
                                ;173110  010703  TM11:   MOV     PC,R3   ;SAVE ERROR RETRY ADDRES
                                ;173112  012737          MOV     #TMRWND,@#TMWC-2;REWIND TAPE
                                ;173114  060017
                                ;173116  172522
                                ;173120  010702          MOV     PC,R2
                                ;173122  000555          BR      TAPES
                                ;173124  172524          .WORD   TMWC
                                ;173126  060003          .WORD   TMREAD  ;TM11 READ COMMAND
                                ;173130  060011          .WORD   TMFWRD  ;TM11 FORWARD RECORD COM
                                ;173132  000200          .WORD   200     ;DONE MASK
                                ;173134  100000          .WORD   100000  ;ERROR MASK
                                ;
                                ;THIS IS THE STARTING ADDRESS FOR RF11 CONTROLLE
                                ;173136  010703  RF11:   MOV     PC,R3   ;SAVE ERROR RETRY ADDRES
                                ;173140  010702          MOV     PC,R2   ;SET POINTER TO PARAMETE
                                ;173142  000516          BR      OTHERA  ;GO TO COMMON SERVICE RO
                                ;                                        ;ASSUME UNIT 0
                                ;173144  177462          .WORD   RFWC    ;DEVICE WORD COUNT REGIS
                                ;173146  000005          .WORD   RFREAD  ;READ COMMAND
                                ;
                                ;THIS IS THE STARTING ADDRESS FOR RH/TU16/TM02
                                ;173150  010703  TU16:   MOV     PC,R3   ;SAVE ERROR RETRY ADDRES
                                ;173152  012700          MOV     #TUCS,R0;GET CONTROL STATUS WORD
                                ;173154  172440
                                ;173156  012710  TU16RE: MOV     #RHPRST,(R0);REWIND TAPE CLEAR E
                                ;173160  000021
                                ;173162  012760          MOV     #TUMODE,32(R0);SET 800 BPI NORMA
                                ;173164  001300
                                ;173166  000032
                                ;173170  012760          MOV     #-1,6(R0);LOAD FRAME COUNT
                                ;173172  177777
                                ;173174  000006
                                ;173176  012710          MOV     #TUSPAC,(R0);SPACE FORWARD
                                ;173200  000031
                                ;173202  105760  1$:     TSTB    12(R0)
                                ;173204  000012
                                ;173206  100375          BPL     1$      ;KEEP LOOPING
                                ;173210  000433          BR      RHCOMN
                                ;
                                ;THIS IS THE STARTING ADDRESS FOR RC11 CONTROLLE
                                ;173212  010703  RC11:   MOV     PC,R3
                                ;173214  010702          MOV     PC,R2   ;ASSUME UNIT 0
                                ;173216  000470          BR      OTHERA
                                ;173220  177450          .WORD   RCWC
                                ;173222  000005          .WORD   RFREAD
                                ;
                                ;THIS IS THE AUTO LOAD VECTOR
                                ;173224  173000          .WORD   RHRSA
                                ;173226  000340          .WORD   340
                                ;
                                ;THIS IS THE STARTING ADDRESS FOR RH11 DEVICE CO
                                ;
                                ;NOTE:  IF TM02/TU16 SHOULD BE SELECTED. THE VAL
                                ;IN CONSOL SWITCH REGISTER IS THE POSITIO
```

```
                        ;                       ;ON THE RH11 INSTEAD OF THE UNIT # ON TU1
                        ;                       ;THE SLAVE UNIT # (# ON TU16) SHOULD STIL
002230  000405 ;173230  000405  RH11A:  BR      1$      ;ENTRY TO SELECT UNIT 0
002232  010703 ;173232  010703  RH11B:  MOV     PC,R3   ;ENTRY TO SELECT ALL UNI
002234  113737 ;173234  113737          MOVB    @#CSW,@#RHCSA+10;LOAD UNIT # INS
002236  177570 ;173236  177570
002240  176310 ;173240  176310
002242  000401 ;173242  000401          BR      2$
002244  010703 ;173244  010703  1$:     MOV     PC,R3
002246  012700 ;173246  012700  2$:     MOV     #RHCSA,R0
002250  176300 ;173250  176300
002252  032760 ;173252  032760  RPCOMN: BIT     #TUTAPE,26(R0);TAPE UNIT?
002254  040000 ;173254  040000
002256  000026 ;173256  000026
002260  001336 ;173260  001336          BNE     TU16RE  ;YES. GO TO TAPE LOGIC
002262  012710 ;173262  012710          MOV     #RHPRST,(R0);RESET DRIVE
002264  000021 ;173264  000021
002266  012760 ;173266  012760          MOV     #14000,32(R0);SET 16 BIT FORMAT
002270  014000 ;173270  014000
002272  000032 ;173272  000032
002274  012710 ;173274  012710          MOV     #DRCLR,(R0);CLEAR DRIVE ERROR
002276  000011 ;173276  000011
                        ;                               ; (GENERATED IF RS03/04
002300  005720 ;173300  005720  RHCOMN: TST     (R0)+   ;MOVE TO WORD COUNT ADDR
002302  010037 ;173302  010037          MOV     R0,@#2  ;FAKE CALLING SEQUENCE
002304  000002 ;173304  000002
002306  012737 ;173306  012737          MOV     #RHREAD,@#4
002310  000071 ;173310  000071
002312  000004 ;173312  000004
002314  005002 ;173314  005002          CLR     R2      ;FOR FLAG AND POINTER TO
002316  000430 ;173316  000430          BR      OTHERA
                        ;               .DSABL  LSB
                        ;
                        ;               ;THIS IS THE STARTING ADDRESS FOR RH11/RP04 DISK
002320  000405 ;173320  000405  RHRPA:  BR      1$      ;ENTRY FOR SELECT UNIT 0
002322  010703 ;173322  010703  RHRPB:  MOV     PC,R3   ;ENTRY TO SELECT ALL UNI
002324  113737 ;173324  113737          MOVB    @#CSW,@#RPCSA+10;LOAD UNIT # INS
002326  177570 ;173326  177570
002330  176710 ;173330  176710
002332  000401 ;173332  000401          BR      2$
002334  010703 ;173334  010703  1$:     MOV     PC,R3
002336  012700 ;173336  012700  2$:     MOV     #RPCSA,R0
002340  176700 ;173340  176700
002342  000743 ;173342  000743          BR      RPCOMN
                        ;
                        ;               ;ENTRY TO BRANCH TO THE PC SELECTED BY CONSOL SW
002344  013707 ;173344  013707  CSRGO:  MOV     @#CSW,PC
002346  177570 ;173346  177570
                        ;
                        ;
                        ;
                        ;               ;THIS IS THE STARTING ADDRESS FOR RP11 CONTROLLE
002350  000405 ;173350  000405  RP11A:  BR      1$      ;ENTRY TO SELECT UNIT 0
002352  010703 ;173352  010703  RP11B:  MOV     PC,R3   ;ENTRY TO SELECT ALL UNI
```

```
                                ;173354  113705          MOVB    @#CSW,R5
                                ;173356  177570
                                ;173360  000305          SWAB    R5      ;GET UNIT # INTO HIGH BY
                                ;173362  000402          BR      3$
                                ;173364  010703  1$:     MOV     PC,R3
                                ;173366  005005          CLR     R5
                                ;173370  010702  3$:     MOV     PC,R2
                                ;173372  000403          BR      OTHER
                                ;173374  176716          .WORD   RPWC
                                ;173376  000005          .WORD   RFREAD
                                ;173400  005005  OTHERA: CLR     R5      ;SET TO UNIT 0
                                ;173402  010200  OTHER:  MOV     R2,R0   ;R0 POINT AT WORD COUNT
                                ;173404  005720          TST     (R0)+   ;POINT TO PARAMETER LIST
                                ;173406  012001          MOV     (R0)+,R1;MOVE WORD COUNT ADDRESS
                                ;173410  012711          MOV     #-256.*2,(R1);LOAD WORD COUNT
                                ;173412  177000
                                ;173414  051005          BIS     (R0),R5 ;COMBINE UNIT # WITH COM
                                ;173416  010541          MOV     R5,-(R1);LOAD READ COMMAND
                                ;173420  032711          BIT     #100200,(R1);CHECK FOR ERROR AND
                                ;173422  100200
                                ;173424  001775          BEQ     .-4     ;WAIT UNTIL COMPLETE
                                ;173426  100012          BPL     1$      ;NO ERROR
                                ;173430  005702          TST     R2      ;WAS IT CALLED BY MASS B
                                ;173432  001024          BNE     AGAIN   ;NO. ERROR
                                ;173434  032761          BIT     #TUTAPE,26(R1);IS TU16?
                                ;173436  040000
                                ;173440  000026
                                ;173442  001420          BEQ     AGAIN   ;NO. ERROR
                                ;173444  022761          CMP     #FCE,14(R1);ARE WE READ A SHORT
                                ;173446  001000
                                ;173450  000014
                                ;173452  001014          BNE     AGAIN   ;SOME OTHER ERROR
                                ;173454  005007  1$:     CLR     PC      ;O.K.
                                ;
                                ;THIS IS THE TAPE DEVICE SERVICE ROUTINE
                                ;173456  010200  TAPES:  MOV     R2,R0   ;GET THE ADDRESS OF THE
                                ;173460  005720          TST     (R0)+   ;STEP TO LAST COMMAND
                                ;173462  012001          MOV     (R0)+,R1;GET THE WORD COUNT ADDR
                                ;173464  005311          DEC     (R1)    ;SET UP TO ADVANCE 1 REC
                                ;173466  005720          TST     (R0)+   ;MOVE R0 TO FIRST COMMAN
                                ;173470  012041          MOV     (R0)+,-(R1);LOAD COMMAND REG.
                                ;173472  031011          BIT     (R0),(R1);DONE?
                                ;173474  001776          BEQ     .-2     ;NO. KEEP LOOPING
                                ;173476  005720          TST     (R0)+   ;YES. CHECK FOR ERROR
                                ;173500  031041          BIT     (R0),-(R1);ANY ERROR?
                                ;173502  001736          BEQ     OTHERA  ;NO ERROR- TRY TO READ
                                ;173504  000005  AGAIN:   RESET
                                ;
                                ;173506  000113          JMP     (R3)    ;ERROR RETURN
                                ;
                                ;THIS IS THE STARTING ADDRESS FOR PC11 PAPER TAP
                                ;173510  012704  KL11:    MOV     #KLCS,R4;OBTAIN CONTROL REG.
                                ;173512  177560
```

```
1012  002514  000443 ;173514  000443          BR      CKDEV   ;AND TRANSFER TO READER
1013          ;
1014          ;               ;
1015          ;
1016          ;               ;CASSETTE TAPE DEVICE COMMAND TABLE
1017  002516  .BYTE   240 ;173516        240  TABLE:  .BYTE   240     ;COMPARE WORD NOT A COMM
1018  002517  .BYTE   037 ;173517        037          .BYTE   37      ;ILBS+RWD+GO
1019  002520  .BYTE   015 ;173520        015          .BYTE   15      ;SPACE FORWARD BLOCK+GO
1020  002521  .BYTE   005 ;173521        005          .BYTE   5       ;READ
1021  002522  .BYTE   024 ;173522        024          .BYTE   24      ;READ +ILBS
1022  002523  .BYTE   224 ;173523        224          .BYTE   224     ;READ+ILBS+END FLAG
1023          ;               ;
1024          ;               ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE D
1025  002524  000404 ;173524  000404  CBOOTA: BR      1$      ;SELECT UNIT 0
1026  002526  113704 ;173526  113704  CBOOTB: MOVB    @#CSW,R4;SELECT UNITS
1027  002530  177570 ;173530  177570
1028  002532  000304 ;173532  000304          SWAB    R4
1029  002534  000401 ;173534  000401          BR      RESETX
1030  002536  005004 ;173536  005004  1$:     CLR     R4
1031  002540  012700 ;173540  012700  RESETX: MOV     #TACS,R0;GET CONTROL REG.
1032  002542  177500 ;173542  177500
1033  002544  000005 ;173544  000005  RESTRT: RESET
1034  002546  010410 ;173546  010410          MOV     R4,(R0) ;SELECT UNIT
1035  002550  012701 ;173550  012701          MOV     #TABLE,R1
1036  002552  173516 ;173552  173516
1037  002554  012702 ;173554  012702          MOV     #375,R2 ;LOAD TRANSFER COUNTER
1038  002556  000375 ;173556  000375
1039  002560  112103 ;173560  112103          MOVB    (R1)+,R3;LOAD COMPARATOR
1040  002562  112110 ;173562  112110  LOOP1:  MOVB    (R1)+,(R0);LOAD COMMAND
1041  002564  100407 ;173564  100407          BMI     DONE
1042  002566  130310 ;173566  130310  LOOP2:  BITB    R3,(R0) ;COMMAND COMPLETE?
1043  002570  001776 ;173570  001776          BEQ     LOOP2   ;NO. WAIT
1044  002572  105202 ;173572  105202          INCB    R2      ;INCREMENT ADDRESS CTR.
1045  002574  100772 ;173574  100772          BMI     LOOP1   ;IF (-), GET COMMAND
1046  002576  116012 ;173576  116012          MOVB    2(R0),(R2);STORE DATA
1047  002600  000002 ;173600  000002
1048  002602  000771 ;173602  000771          BR      LOOP2   ;GET ANOTHER BYTE
1049  002604  005710 ;173604  005710  DONE:   TST     (R0)    ;ANY ERROR?
1050  002606  100756 ;173606  100756          BMI     RESTRT  ;YES, RETRY
1051  002610  005002 ;173610  005002          CLR     R2      ;CLEAR COMPARE ADDRESS
1052  002612  120312 ;173612  120312          CMPB    R3,(R2) ;IT MUST BE 240
1053  002614  001377 ;173614  001377          BNE     .
1054  002616  000112 ;173616  000112  ERROR:  JMP     (R2)
1055          ;               ;
1056          ;               ;THIS IS THE STARTING ADDRESS FOR THE PC11 CONTR
1057  002620  012704 ;173620  012704  PC11:   MOV     #PCCS,R4
1058  002622  177550 ;173622  177550
1059  002624  000005 ;173624  000005  CKDEV:  RESET
1060  002626  012701 ;173626  012701          MOV     #160000,R1;SET UP MEMORY TEST LI
1061  002630  160000 ;173630  160000
1062  002632  012702 ;173632  012702          MOV     #6,R2   ;SET UP POINTER TO TIMEO
1063  002634  000006 ;173634  000006
1064  002636  012712 ;173636  012712          MOV     #340,(R2);SET UP VECTOR TO RETUR
1065  002640  000340 ;173640  000340
```

```
1066  002642  010742 ;173642  010742          MOV     PC,-(R2);SAVE PC
1067  002644  012706 ;173644  012706          MOV     #24,SP  ;LOAD UP STACK POINTER
1068  002646  000024 ;173646  000024
1069  002650  010441 ;173650  010441          MOV     R4,-(R1);LOOK FOR END OF MEMORY
1070  002652  040601 ;173652  040601          BIC     SP,R1   ;THEN DROP TO XX752
1071  002654  010111 ;173654  010111          MOV     R1,(R1) ;AND STORE IN ITSELF
1072  002656  011102 ;173656  011102  LOOP:   MOV     (R1),R2
1073  002660  005214 ;173660  005214          INC     (R4)    ;START DEVICE
1074  002662  105714 ;173662  105714  RDRWAT: TSTB    (R4)    ;WAIT
1075  002664  100376 ;173664  100376          BPL     RDRWAT
1076  002666  116412 ;173666  116412          MOVB    2(R4),(R2);SAVE THE DATA
1077  002670  000002 ;173670  000002
1078  002672  005211 ;173672  005211          INC     (R1)
1079  002674  120227 ;173674  120227          CMPB    R2,#375
1080  002676  000375 ;173676  000375
1081  002700  001366 ;173700  001366          BNE     LOOP    ;NO
1082  002702  105222 ;173702  105222          INCB    (R2)+   ;YES
1083  002704  000142 ;173704  000142          JMP     -(R2)
1084  002706  000000 ;173706  000000  ;THIS AREA IS UNUSED
1085  002710  000000 ;173710  000000  ;THIS AREA IS UNUSED
1086  002712  000000 ;173712  000000  ;THIS AREA IS UNUSED
1087  002714  000000 ;173714  000000  ;THIS AREA IS UNUSED
1088  002716  000000 ;173716  000000  ;THIS AREA IS UNUSED
1089  002720  000000 ;173720  000000  ;THIS AREA IS UNUSED
1090  002722  000000 ;173722  000000  ;THIS AREA IS UNUSED
1091  002724  000000 ;173724  000000  ;THIS AREA IS UNUSED
1092  002726  000000 ;173726  000000  ;THIS AREA IS UNUSED
1093  002730  000000 ;173730  000000  ;THIS AREA IS UNUSED
1094  002732  000000 ;173732  000000  ;THIS AREA IS UNUSED
1095  002734  000000 ;173734  000000  ;THIS AREA IS UNUSED
1096  002736  000000 ;173736  000000  ;THIS AREA IS UNUSED
1097  002740  000000 ;173740  000000  ;THIS AREA IS UNUSED
1098  002742  000000 ;173742  000000  ;THIS AREA IS UNUSED
1099  002744  000000 ;173744  000000  ;THIS AREA IS UNUSED
1100  002746  000000 ;173746  000000  ;THIS AREA IS UNUSED
1101  002750  000000 ;173750  000000  ;THIS AREA IS UNUSED
1102  002752  000000 ;173752  000000  ;THIS AREA IS UNUSED
1103  002754  000000 ;173754  000000  ;THIS AREA IS UNUSED
1104  002756  000000 ;173756  000000  ;THIS AREA IS UNUSED
1105  002760  000000 ;173760  000000  ;THIS AREA IS UNUSED
1106  002762  000000 ;173762  000000  ;THIS AREA IS UNUSED
1107  002764  000000 ;173764  000000  ;THIS AREA IS UNUSED
1108  002766  000000 ;173766  000000  ;THIS AREA IS UNUSED
1109  002770  000000 ;173770  000000  ;THIS AREA IS UNUSED
1110  002772  000000 ;173772  000000  ;THIS AREA IS UNUSED
1111  002774  000000 ;173774  000000  ;THIS AREA IS UNUSED
1112  002776  END.YB:
1113  002776  000000 ;173776  000000  ;THIS AREA IS UNUSED
```

```
                        MAP.YC:
                        ;THE FOLLOWING 1000 LOCATIONS ARE
                        ;A REPRODUCTION OF THE ROM PROGRAM
                        ;FOR THE BM873YC. THE FIRST 400 LOCATIONS
                        ;ARE AN EXACT COPY OF THE BM873YA. THE
                        ;REMAINING 400 LOCATIONS ARE
                        ;THE DDCMP BOOTSTRAP ROM PROGRAM.
                        ;IT IS HERE FOR COMPARISON TO
                        ;ACTUAL ROM AND FOR REFERENCE.
                        ;173000 .=173000          ;STARTING ADDRESS FOR BOOTSTRAP
                                                  ;THIS LOADER IS DESIGNED FOR THE RESTART MODULE M873.
                                                  ;IT FUNCTIONALLY REPLACES THE FOLLOWING ROMS:
                                                  ;M792-YA - PAPER TAPE BOOTSTRAP FOR PC11,KL11
                                                  ;MR11-DB   BULK STORAGE BOOTSTRAP ROM
                                                  ;M792-YH   TA11 CASSETTE BOOTSTRAP ROM
                        ;               000000  R0=     %0        ;REGISTER DEFINITIONS
                        ;               000001  R1=     %1
                        ;               000002  R2=     %2
                        ;               000003  R3=     %3
                        ;               000004  R4=     %4
                        ;               000005  R5=     %5
                        ;               000006  SP=     %6
                        ;               000007  PC=     %7
                        ;               177570  SR=     177570          ;PROCESSOR SWITCH REGISTER
                                                  ;
                                                  ;STARTING LOCATION FOR RF11 DISK
003000  010702  ;173000 010702  RF11:   MOV PC,R2         ;SET POINTER TO PARAMETER LISTS
003002  000464  ;173002 000464          BR OTHER          ;TRANSFER TO SERVICE ROUTINE
003004  177462  ;173004 177462          .WORD 177462      ;DEVICE WORD COUNT ADDRESS
003006  000005  ;173006 000005          .WORD 5           ;DEVICE READ INSTRUCTION
                                                  ;
                                                  ;THIS IS THE STARTING LOCATION FOR THE RK11 CONTROLLER
003010  010702  ;173010 010702  RK11:   MOV PC,R2         ;SET POINTER TO PARAMETER LIST
003012  000460  ;173012 000460          BR OTHER          ;TRANSFER TO SERVICE ROUTINE
003014  177406  ;173014 177406          .WORD 177406      ;DEVICE WORD COUNT REGISTER
003016  000005  ;173016 000005          .WORD 5           ;DEVICE READ INSTRUCTION
                                                  ;
                                                  ;THIS IS A SPARE STARTING LOCATION. IT TRANSFERS TO ADDRESS
                                                  ;CONTAINED IN THE SWITCH REGISTER.
003020  013707  ;173020 013707  TRANSR: MOV @#SR,PC      ;GO TO INDICATED LOCATION
003022  177570  ;173022 177570
                        ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
                                                  ;
                                                  ;THIS IS THE POWER UP VECTOR REQUIRED FOR DEVICE AND
003024  173000  ;173024 173000  POWER:  .WORD RF11        ;ADDRESS OF FIRST LOCATION IN ROM
003026  000340  ;173026 000340          .WORD 340         ;PROCESSOR STATUS LEVEL 7
                                                  ;
                                                  ;THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE) CONTROLLER.
003030  010702  ;173030 010702  TC11:   MOV PC,R2         ;SET UP POINTER TO PARAMETER LIST
003032  000426  ;173032 000426          BR TAPES          ;AND TRANSFER TO FIRST ROUTINE
003034  177344  ;173034 177344          .WORD 177344      ;DEVICE WORD COUNT ADDRESS
003036  004003  ;173036 004003          .WORD 4003        ;FIND PREVIOUS BLOCK COMMAND
003040  100000  ;173040 100000          .WORD 100000      ;USED AS DONE INDICATOR
003042  024000  ;173042 024000          .WORD 24000       ;USED AS ERROR INDICATOR/TEST FLAG
```

```
1168  003044  000445  ;173044 000445          BR OTHERX       ;THEN TRANSFER TO NEXT ROUTINE
1169  003046  000005  ;173046 000005          .WORD 5         ;DEVICE READ COMMAND
1170                                  ;
1171                                  ;THIS IS THE START LOCATION FOR TM11 MAGTAPE CONTROLLER
1172  003050  010702  ;173050 010702  TM11:    MOV PC,R2       ;SET POINTER TO PARAMETER LIST
1173  003052  000416  ;173052 000416          BR TAPES        ;AND TRANSFER TO FIRST ROUTINE
1174  003054  172524  ;173054 172524          .WORD 172524    ;DEVICE BYTE/RECORD COUNT REGISTER
1175  003056  060017  ;173056 060017          .WORD 60017     ;DEVICE REWIND COMMAND
1176  003060  000200  ;173060 000200          .WORD 200       ;DEVICE DONE FLAG
1177  003062  100000  ;173062 100000          .WORD 100000    ;DEVICE ERROR FLAG BIT
1178  003064  000413  ;173064 000413          BR TAPESX       ;THEN TRANSFER TO NEXT SERVICE RTN
1179  003066  060011  ;173066 060011          .WORD 60011     ;DEVICE FORWARD SPACE COMMAND
1180  003070  000200  ;173070 000200          .WORD 200       ;SAME AS ABOVE
1181  003072  100000  ;173072 100000          .WORD 100000    ;SAME AS ABOVE
1182  003074  000431  ;173074 000431          BR OTHERX       ;THEN TRANSFER TO READ/TRANSFER ROUTINE
1183  003076  060003  ;173076 060003          .WORD 60003     ;DEVICE READ COMMAND
1184                                  ;
1185                                  ;THIS IS THE START LOCATION FOR THE RP11 CONTROLLER
1186  003100  010702  ;173100 010702  RP11:    MOV PC,R2       ;SET POINTER TO PARAMETER LIST
1187  003102  000424  ;173102 000424          BR OTHER        ;TRANSFER TO TRANSFER ROUTINE
1188  003104  176716  ;173104 176716          .WORD 176716    ;DEVICE WORD COUNT REGISTER
1189  003106  000005  ;173106 000005          .WORD 5         ;DEVICE READ COMMAND
1190                                  ;
1191                                  ;THIS IS THE TAPE DEVICE SERVICE ROUTINE.
1192  003110  010200  ;173110 010200  TAPES:   MOV R2,R0       ;GET ADDRESS OF PARAMETER LIST
1193  003112  005720  ;173112 005720          TST (R0)+       ;SKIP TWO WORDS FIRST TIME
1194  003114  000005  ;173114 000005  TAPESX:  RESET           ;RESET ALL DEVICES
1195  003116  005720  ;173116 005720          TST (R0)+       ;SKIP OVER BRANCH INSTRUCTION
1196  003120  016201  ;173120 016201          MOV 2(R2),R1    ;THEN GET DEVICE WORD/BYTE COUNT ADDRES
1197  003122  000002  ;173122 000002
1198  003124  005311  ;173124 005311          DEC @R1         ;AND SET TO -1
1199  003126  012041  ;173126 012041          MOV (R0)+,-(R1) ;AND THEN ISSUE COMMAND TO DEVICE
1200  003130  031011  ;173130 031011  TAPWAT:  BIT @R0,@R1     ;WAIT FOR DEVICE COMPLETION
1201  003132  001776  ;173132 001776          BEQ TAPWAT      ;BY HANGING IN LOOP
1202  003134  005720  ;173134 005720          TST (R0)+       ;AND THEN SKIP DONE FLAG
1203  003136  032041  ;173136 032041          BIT (R0)+,-(R1) ;THEN TEST FOR ERROR
1204  003140  001063  ;173140 001063          BNE ERROR       ;THERE IS ONE
1205  003142  000110  ;173142 000110  RETURN:  JMP @R0         ;AND TRANSFER TO FOLLOWING INSTRUCTION
1206                                  ;
1207                                  ;THIS IS THE STARTING ADDRESS FOR RC11 DISK CONTROLLERS
1208  003144  010702  ;173144 010702  RC11:    MOV PC,R2       ;SET UP POINTER TO PARAMETER LIST
1209  003146  000402  ;173146 000402          BR OTHER        ;TRANSFER TO SERVICE RTN
1210  003150  177450  ;173150 177450          .WORD 177450    ;DEVICE WORD COUNT REGISTER
1211  003152  000005  ;173152 000005          .WORD 5         ;DEVICE READ INSTRUCTION
1212                                  ;
1213                                  ;THIS ROUTINE PERFORMS THE ACTUAL TRANSFER TO MEMORY OF DATA
1214  003154  010200  ;173154 010200  OTHER:   MOV R2,R0       ;SET POINTER TO LIST IN R0
1215  003156  005720  ;173156 005720          TST (R0)+       ;SKIP TWO WORDS FIRST TIME.
1216  003160  005720  ;173160 005720  OTHERX:  TST (R0)+       ;SKIP PAST BR INSTRUCTION
1217  003162  000005  ;173162 000005          RESET           ;REST THE WORLD
1218  003164  016201  ;173164 016201          MOV 2(R2),R1    ;OBTAIN DEVICE WORD COUNT ADDRESS
1219  003166  000002  ;173166 000002
1220  003170  012711  ;173170 012711          MOV #-1000,@R1  ;THEN OBTAIN LARGE WORD COUNT
1221  003172  177000  ;173172 177000
```

```
1222  003174  011041  ;173174 011041          MOV @R0,-(R1)   ;AND PUT COMMAND TO DEVICE
1223  003176  105711  ;173176 105711  OTHWAT: TSTB @R1        ;WAIT FOR DONE FLAG
1224  003200  100376  ;173200 100376          BPL OTHWAT      ;BY HANGING IN LOOP
1225  003202  005711  ;173202 005711          TST @R1         ;THEN TEST FOR ERROR
1226  003204  100441  ;173204 100441          BMI ERROR       ;GOT PROBLEMS
1227  003206  005007  ;173206 005007          CLR PC          ;AND TRANSFER TO ZERO
1228                                          ;
1229                                          ;THIS IS THE STARTING ADDRESS FOR THE PC11 PAPER TAPE CONTROLLER
1230  003210  012704  ;173210 012704  KL11:   MOV #177560,R4  ;OBTAIN DEVICE ADDRESS
1231  003212  177560  ;173212 177560
1232  003214  000440  ;173214 000440          BR CKDEV        ;AND TRANSFER TO READER SERVICE ROUTINE
1233                                          ;
1234
1235                                          ;THIS IS THE CASSETTE DEVICE COMMAND TABLE
1236  003216  017640  ;173216     240  TABLE:  .BYTE 240       ;COMPARE WORD NOT A COMMAND
1237                  ;173217     037          .BYTE 37        ;ILBS+RWD+GO
1238  003220  002415  ;173220     015          .BYTE 15        ;SPACE FORWARD BLOCK+GO
1239                  ;173221     005          .BYTE 5         ;READ+GO
1240  003222  112024  ;173222     024          .BYTE 24        ;READ+ILBS
1241                  ;173223     224          .BYTE 224       ;READ+ILBS+END FLAG
1242          ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
1243                                          ;
1244                                          ;THIS IS AN ADDITIONAL POWER VECTOR ADDRESS REQUIRED BY DEVICE
1245  003224  173000  ;173224 173000  POWER2: .WORD RF11       ;ADDRESS OF BEGINNING OF BOOTSTRAP
1246  003226  000340  ;173226 000340          .WORD 340       ;PRIORITY LEVEL 7
1247                                          ;
1248                                          ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE DEVICE #0
1249  003230  005004  ;173230 005004  CBOOT:  CLR R4           ;LOAD DEVICE NUMBER 0 IN R4
1250  003232  012700  ;173232 012700  RESTX:  MOV #177500,R0  ;GET DEVICE ADDTESS
1251  003234  177500  ;        177500
1252  003236  000005  ;173236 000005  RESTRT: RESET           ;ISSUE RESET INSTRUCTION
1253  003240  010410  ;173240 010410          MOV R4,@R0      ;LOAD DEVICE WITH UNIT NUMBER
1254  003242  012701  ;173242 012701          MOV #TABLE,R1   ;GET FUNNY TABLE OF INSTRUCTIONS
1255  003244  173216  ;173244 173216
1256  003246  012702  ;173246 012702          MOV #375,R2     ;AND LOAD UP TRANSFER COUNTER
1257  003250  000375  ;173250 000375
1258  003252  112103  ;173252 112103          MOVB (R1)+,R3   ;THE LOAD UP COMPARATOR
1259  003254  112110  ;173254 112110  LOOP1:  MOVB (R1)+,@R0  ;LOAD DEVICE REGISTER WITH COMMAND
1260  003256  100407  ;173256 100407          BMI DONE
1261  003260  130310  ;173260 130310  LOOP2:  BITB R3,@R0     ;HAS COMMAND COMPLETED
1262  003262  001776  ;173262 001776          BEQ LOOP2       ;NO, WAIT
1263  003264  105202  ;173264 105202          INCB R2         ;THEN INCREMENT ADDRESS CTR
1264  003266  100772  ;173266 100772          BMI LOOP1       ;IF NEGATIVE, GET COMMAND
1265  003270  116012  ;173270 116012          MOVB 2(R0),@R2  ;AND STORE DATA AWAY
1266  003272  000002  ;173272 000002
1267  003274  000771  ;173274 000771          BR LOOP2        ;GO GET ANOTHER BYTE
1268  003276  005710  ;173276 005710  DONE:   TST @R0         ;ANY DEVICE ERRORS
1269  003300  100756  ;173300 100756          BMI RESTRT      ;YES, RETRY
1270  003302  005002  ;173302 005002          CLR R2          ;CLEAR COMPARE ADDRESS AND TRANSFER ADDRESS
1271  003304  120312  ;173304 120312          CMPB R3,@R2     ;IT MUST BE 240
1272  003306  001377  ;173306 001377          BNE .+0         ;NO, THERE WAS AN ERROR
1273  003310  000112  ;173310 000112  ERROR:  JMP @R2         ;NORMAL CASSETTE AND ERROR FOR BULK STORAGE
1274                                          ;
1275                                          ;THIS IS THE STARTING LOCATION FOR THE PC11 CONTROLLER
```

```
1276  003312  012704  ;173312 012704  PC11:   MOV #177550,R4  ;LOAD DEVICE ADDRESS
1277  003314  177550  ;173314 177550
1278  003316  000005  ;173316 000005  CKDEV:  RESET            ;KILL ALL DEVICE ACTION
1279  003320  012701  ;173320 012701          MOV #160000,R1  ;THEN SET UP MEMORY TEST LIMITS
1280  003322  160000  ;173322 160000
1281  003324  012702  ;173324 012702          MOV #6,R2       ;AND SET UP POINTER TO TIMEOUT LOCATION
1282  003326  000006  ;173326 000006
1283  003330  012712  ;173330 012712          MOV #340,@R2    ;AND SET UP VECTOR TO RETURN TO NEXT
1284  003332  000340  ;173332 000340
1285  003334  010742  ;173334 010742          MOV PC,-(R2)    ;SAVE THE PC
1286  003336  012706  ;173336 012706          MOV #24,SP      ;AND LOAD UP STACK POINTER
1287  003340  000024  ;173340 000024
1288  003342  010441  ;173342 010441          MOV R4,-(R1)    ;AND LOOK FOR END OF MEMORY
1289  003344  040601  ;173344 040601          BIC SP,R1       ;THEN DROP TO XX7752
1290  003346  010111  ;173346 010111          MOV R1,@R1      ;AND STORE IN ITSELF
1291  003350  011102  ;173350 011102  LOOP:   MOV @R1,R2      ;THEN LOAD ADDRESS FOR DATA INSERTION
1292  003352  005214  ;173352 005214          INC @R4         ;AND START DEVICE
1293  003354  105714  ;173354 105714  RDRWAT: TSTB @R4        ;THEN WAIT FOR CHARACTER AVAILABLE
1294  003356  100376  ;173356 100376          BPL RDRWAT      ;HANGING THERE IF NECCESSARY
1295  003360  116412  ;173360 116412          MOVB 2(R4),@R2  ;STORE AWAY DATA BYTE
1296  003362  000002  ;173362 000002
1297  003364  005211  ;173364 005211          INC @R1
1298  003366  120227  ;173366 120227          CMPB R2,#375    ;HAS BRANCH OFFSET BEEN STORED
1299  003370  000375  ;173370 000375
1300  003372  001366  ;173372 001366          BNE LOOP        ;NO
1301  003374  105222  ;173374 105222          INCB (R2)+      ;YES, ALL DONE
1302  003376  000142  ;173376 000142          JMP -(R2)       ;THEN TRANSFER TO RTN
1303
1304          ;THE FOLLOWING 400 LOCATIONS ARE
1305          ;A REPRODUCTION OF THE DDCMP BOOT-
1306          ;STRAP ROM. IT IS HERE FOR COM-
1307          ;PARISON TO THE ACTUAL ROM AND
1308          ;FOR REFERENCE.
1309
1310                                  ;       COPYRIGHT  1975, DIGITAL EQUIPMENT CORP., MAYNARD,MASS. 01754
1311                                  ;
1312                                  ;       THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A
1313                                  ;       LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND
1314                                  ;       CAN BE COPIED (WITH INCLUSION OF DEC'S
1315                                  ;       COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM,
1316                                  ;       EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING
1317                                  ;       BY DEC.
1318                                  ;
1319                                  ;       THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO
1320                                  ;       CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED
1321                                  ;       AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
1322                                  ;
1323                                  ;       DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR
1324                                  ;       RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH
1325                                  ;       IS NOT SUPPLIED BY DEC.
1326                                  ;
1327                                  ;       VERSION 01
1328                                  ;
1329                                  ;       STUART WECKER  01/22/75
```

```
1330
1331                            ;       DIGITAL EQUIPMENT CORPORATION
1332                            ;       COMPUTER NETWORK FACILITIES
1333                            ;       DOWN-LINE LOADING PROGRAM
1334                            ;
1335                            ;       THIS PROGRAM LOADS COMPUTER MEMORY FROM DATA SENT OVER
1336                            ;       A DATA COMMUNICATIONS LINK. IT SENDS AND RECEIVES
1337                            ;       MESSAGES IN DDCMP BOOT FORMAT.  THE PRIMARY BOOT ONLY
1338                            ;       LOADS A SINGLE BLOCK, THE SECONDARY BOOT,  WHICH
1339                            ;       THEN REQUESTS AND LOADS THE DESIRED PROGRAM.
1340                            ;
1341                            ;       CURRENT VERSION DDCMP: 3.0 - MAY 7,1974
1342                            ;
1343                            ;       THE BOOTSTRAP MESSAGES ARE OF THE FORM:
1344                            ;
1345                            ;       SYN,SYN,DLE,CNT,F,S,FILL,FILL,ADDR,CRC1,DATA,CRC2
1346                            ;
1347                            ;       ALL ITEMS ARE 8-BITS LONG UNLESS OTHERWISE SPECIFIED
1348                            ;
1349                            ;       SYN-THE SYNC CHARACTER-SYNC-226,ASYNC-377
1350                            ;       DLE-THE BOOT HEADER CHARACTER-OCTAL 220
1351                            ;       CNT-THE 14-BIT COUNT FIELD-LENGTH OF DATA FIELD
1352                            ;       F-THE FINAL BIT-LINK CONTROL
1353                            ;       S-THE SELECT BIT-LINK CONTROL
1354                            ;       FILL-A FILL CHARACTER-OCTAL 000
1355                            ;       ADDR-THE STATION ADDR-FOR PT. TO PT.=1
1356                            ;       CRC1-THE 16-BIT CRC-16 COMPUTED ON DLE THROUGH ADDR
1357                            ;       DATA-THE BOOT DATA AS FOLLOWS:
1358                            ;               CODE,INFO
1359                            ;       ONLY THE FOLLOWING CODES ARE USED BY THE
1360                            ;       PRIMARY BOOT
1361                            ;       CODE=10  REQUEST SECONDARY PROGRAM
1362                            ;       INFO=DEVICE TYPE,STATION ADDRESS
1363                            ;       DEVICE TYPE-DP=0,DU=2,DL=4,DQ=6
1364                            ;       STATION ADDRESS=1
1365                            ;       CODE=0  PROGRAM LOAD WITH TRANSFER ADDRESS
1366                            ;       INFO=BLKNO,BLK LDADDR,IMAGE DATA,TRANS ADDR
1367                            ;       BLKNO=0
1368                            ;       BLOCK LDADDR=6
1369                            ;       TRANS ADDR=6
1370                            ;       HEADER COUNT > OR = TO 10.
1371                            ;       ADDRESSES ARE 4 BYTES-32 BITS-LOW BIT FIRST
1372                            ;       CRC2-THE 16-BIT CRC-16 COMPUTED ON THE DATA FIELD ONLY
1373                            ;
1374                            ;
1375                            ;       OPTION SWITCHES:
1376                            ;               DEVICE-DP11,DU11,DL11
1377                            ;               CRC-KG11,SCRC
1378                            ;
1379                            ;
1380                            ;
1381                            ;       REGISTER DEFINITIONS
1382                            ;
1383    ;       000000  R0=%0                   ;BLOCK LOAD ADDR
```

```
                                ;       000001  R1=%1                   ;DEVICE CSR ADDRESS
                                ;       000002  R2=%2                   ;CRC CALC TEMP
                                ;       000003  R3=%3                   ;SOFTWARE CRC
                                ;       000004  R4=%4                   ;BLOCK CHAR COUNT
                                ;       000005  R5=%5                   ;CRC CALC TEMP
                                ;       000006  SP=%6                   ;STACK ADDR
                                ;       000007  PC=%7                   ;LOCATION COUNTER
                                                ;
                                                ;       LITERALS
                                                ;
                                ;       000001  $STADR=1                ;STATION ADDR
                                ;       177570  $SWR=177570             ;SWITCH REGISTER ADDR
                                ;       000226  $SYN=226                ;SYNC CHARACTER
                                ;       000220  $DLE=220                ;DDCMP DLE CHARACTER
                                ;       000400  $STRIP=400
                                                ;
                                                ;       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
                                                ;
                                ;       173400  .=173400
003400  012700  ;173400 012700  START1: MOV     (PC)+,R0                ;NON ZERO VALUE TO R0
003402  005000  ;173402 005000  START2: CLR     R0                      ;CLEAR R0
003404  000005  ;173404 000005          RESET                           ;RESET SYS,MEM MGT, ETC...
003406  012706  ;173406 012706          MOV     #17776,SP               ;STACK AT 4K-2
003410  017776  ;173410 017776
                                                ;
                                                ;       FIND THE DU-11 IN THE FLOATING ADDRESS SPACE
                                                ;
003412  010702  ;173412 010702          MOV     PC,R2                   ;CURRENT PC
003414  062702  ;173414 062702          ADD     #DEVTAB-.,R2            ;DEVICE TABLE ADDR
003416  000360  ;173416 000360
003420  012703  ;173420 012703          MOV     #6,R3                   ;TRAP PS ADDR
003422  000006  ;173422 000006
003424  005013  ;173424 005013          CLR     (R3)                    ;CLEAR NEW PS
003426  010243  ;173426 010243          MOV     R2,-(R3)                ;TABLE ADDR TO LOC 4
003430  160313  ;173430 160313          SUB     R3,(R3)                 ;SUB TO TRAP RTN
003432  005303  ;173432 005303          DEC     R3                      ;LEAVE CNT 3 FOR LOOP
003434  012701  ;173434 012701          MOV     #160010,R1              ;START SEARCH ADDR
003436  160010  ;173436 160010
003440  005711  ;173440 005711  DEVLOP: TST     (R1)                    ;IS DEVICE THERE
003442  111204  ;173442 111204          MOVB    (R2),R4                 ;DEVICE INCREMENT TO R3
003444  060401  ;173444 060401          ADD     R4,R1                   ;UPDATE TO NEXT DEVICE
003446  005201  ;173446 005201          INC     R1                      ;INCREMENT MODULO
003450  040401  ;173450 040401          BIC     R4,R1                   ;CLEAR EXCESS
003452  005703  ;173452 005703          TST     R3                      ;TEST FOR DONE
003454  001371  ;173454 001371          BNE     DEVLOP                  ;NOT YET
003456  005700  ;173456 005700          TST     R0                      ;TEST SWITCH REG USE
```

```
1438  003460  001002  ;173460 001002          BNE     SNDREQ          ;NO SWITCH REG
1439  003462  063701  ;173462 063701          ADD     @#$SWR,R1       ;ADD SWR VALUE
1440  003464  177570  ;173464 177570
1441                                  ;
1442                                  ;       SET UP DEVICE FOR OUTPUT
1443                                  ;
1444                  ;173466         SNDREQ:
1445  003466  012711  ;173466 012711          MOV     #6,(R1)         ;DATA TERM RDY AND REQ TO SEND
1446  003470  000006  ;173470 000006
1447  003472  012761  ;173472 012761          MOV     #36000+$SYN,2(R1) ;SET SYNC REGISTER
1448  003474  036226  ;173474 036226
1449  003476  000002  ;173476 000002
1450  003500  032711  ;173500 032711  L3:     BIT     #20000,(R1)     ;TEST CLEAR TO SEND
1451  003502  020000  ;173502 020000
1452  003504  001775  ;173504 001775          BEQ     L3              ;NOT YET
1453  003506  022121  ;173506 022121          CMP     (R1)+,(R1)+     ;MOVE PTR TO XMIT TSR
1454  003510  052711  ;173510 052711          BIS     #20,(R1)        ;TURN SEND ON
1455  003512  000020  ;173512 000020
1456                                  ;
1457                                  ;       SEND SECONDARY PGM REQUEST MESSAGE
1458                                  ;
1459  003514  010700  ;173514 010700          MOV     PC,R0           ;CURRENT PC
1460  003516  062700  ;173516 062700          ADD     #RQMSG-.,R0     ;REQUEST MSG ADDR
1461  003520  000230  ;173520 000230
1462  003522  012704  ;173522 012704          MOV     #RQMSGE-RQMSG,R4 ;COUNT
1463  003524  000026  ;173524 000026
1464  003526  112061  ;173526 112061  L4:     MOVB    (R0)+,2(R1)     ;CHAR TO XMIT REGISTER
1465  003530  000002  ;173530 000002
1466  003532  105711  ;173532 105711  L5:     TSTB    (R1)            ;DONE YET ?
1467  003534  100376  ;173534 100376          BPL     L5              ;NO
1468  003536  005304  ;173536 005304          DEC     R4              ;DECREMENT COUNT
1469  003540  001372  ;173540 001372          BNE     L4              ;ONCE MORE
1470  003542  042711  ;173542 042711          BIC     #20,(R1)        ;DROP SEND
1471  003544  000020  ;173544 000020
1472  003546  024141  ;173546 024141          CMP     -(R1),-(R1)     ;RESET PTR TO RCV CSR
1473
1474                                  ;       GET SECONDARY PROGRAM
1475                                  ;
1476                  ;173550         GETPGM:
1477  003550  042711  ;173550 042711          BIC     #20,(R1)        ;CLEAR SEARCH SYNC
1478  003552  000020  ;173552 000020
1479  003554  012711  ;173554 012711          MOV     #422,(R1)       ;SET FOR CLEAR AND STRIP SYNC
1480  003556  000422  ;173556 000422
1481  003560  005003  ;173560 005003          CLR     R3              ;CLEAR CRC VALUE
1482                                  ;
1483                                  ;       GET MESSAGE HEADER AND CHECK IT
1484                                  ;
1485  003562  012700  ;173562 012700          MOV     #1,R0           ;LOAD HDR AT LOC. 1
1486  003564  000001  ;173564 000001
1487  003566  012704  ;173566 012704          MOV     #8.,R4          ;BLOCK COUNT
1488  003570  000010  ;173570 000010
1489  003572  004767  ;173572 004767          JSR     PC,GET          ;GET HEADER
1490  003574  000060  ;173574 000060
1491  003576  005703  ;173576 005703          TST     R3              ;CHECK HEADER CRC
```

```
1492  003600  001363  ;173600 001363          BNE     GETPGM          ;NO GOOD
1493  003602  123727  ;173602 123727          CMPB    @#6,#$STADR     ;CHECK FOR MY ADDR
1494  003604  000006  ;173604 000006
1495  003606  000001  ;173606 000001
1496  003610  001357  ;173610 001357          BNE     GETPGM          ;NOT MINE
1497  003612  123727  ;173612 123727          CMPB    @#1,#$DLE       ;IS THIS A DLE MSG
1498  003614  000001  ;173614 000001
1499  003616  000220  ;173616 000220
1500  003620  001322  ;173620 001322          BNE     SNDREQ          ;NO, ASK FOR ONE
1501                                  ;
1502                                  ;       GET DATA BLOCK
1503                                  ;
1504  003622  013704  ;173622 013704          MOV     @#2,R4          ;DATA FIELD LENGTH
1505  003624  000002  ;173624 000002
1506  003626  042704  ;173626 042704          BIC     #140000,R4      ;MASK OFF S,F BITS
1507  003630  140000  ;173630 140000
1508  003632  122424  ;173632 122424          CMPB    (R4)+,(R4)+     ;ADD 2 FOR CRC
1509  003634  005000  ;173634 005000          CLR     R0              ;LOAD INTO LOCATION 0
1510  003636  004767  ;173636 004767          JSR     PC,GET1         ;GET DATA BLOCK
1511  003640  000014  ;173640 000014
1512  003642  005703  ;173642 005703          TST     R3              ;CHECK DATA FIELD CRC
1513  003644  001310  ;173644 001310          BNE     SNDREQ          ;NO GOOD
1514  003646  105713  ;173646 105713          TSTB    (R3)            ;CHECK CODE IN LOC 0
1515  003650  001306  ;173650 001306          BNE     SNDREQ          ;NOT PROGRAM LOAD
1516  003652  000137  ;173652 000137          JMP     @#6             ;TRANSFER TO SECONDARY PGM
1517  003654  000006  ;173654 000006
1518                                  ;
1519                                  ;       GET A BLOCK AND COMPUTE CRC
1520                                  ;
1521                  ;173656         GET:
1522                  ;173656         GET1:
1523  003656  105711  ;173656 105711          TSTB    (R1)            ;IS DEVICE DONE YET
1524  003660  100376  ;173660 100376          BPL     GET             ;NOT YET
1525  003662  042711  ;173662 042711          BIC     #$STRIP,(R1)    ;NO STRIP SYNC
1526  003664  000400  ;173664 000400
1527  003666  116110  ;173666 116110          MOVB    2(R1),(R0)      ;STORE IT
1528  003670  000002  ;173670 000002
1529                                  ;
1530                                  ;       CRC CALCULATION ROUTINE
1531                                  ;
1532                  ;       120001  POLY=120001                     ;CRC-16 POLYNOMIAL
1533                                  ;
1534  003672  012705  ;173672 012705          MOV     #8.,R5          ;BYTE LENGTH
1535  003674  000010  ;173674 000010
1536  003676  112002  ;173676 112002          MOVB    (R0)+,R2        ;CHARACTER TO ADD TO CRC
1537  003700  000241  ;173700 000241  CRCLOP: CLC                     ;CLEAR CARRY
1538  003702  006003  ;173702 006003          ROR     R3              ;SHIFT OLD PARTIAL
1539  003704  103003  ;173704 103003          BCC     L10             ;IF CLEAR CHECK CHAR
1540  003706  006002  ;173706 006002          ROR     R2              ;SHIFT CHARACTER
1541  003710  103003  ;173710 103003          BCC     L11             ;XOR POLY
1542  003712  000410  ;173712 000410          BR      L12             ;NEXT BIT
1543  003714  006002  ;173714 006002  L10:    ROR     R2              ;SHIFT CHARACTER
1544  003716  103006  ;173716 103006          BCC     L12             ;NEXT BIT
1545  003720  012746  ;173720 012746  L11:    MOV     #POLY,-(SP)     ;POLY TO STACK
```

```
1546  003722  120001  ;173722 120001
1547  003724  040316  ;173724 040316          BIC     R3,(SP)         ;NOT PARTIAL AND POLY
1548  003726  042703  ;173726 042703          BIC     #POLY,R3        ;NOT POLY AND PARTIAL
1549  003730  120001  ;173730 120001
1550  003732  052603  ;173732 052603          BIS     (SP)+,R3        ;POLY XOR PARTIAL
1551  003734  005305  ;173734 005305  L12:    DEC     R5              ;DECREMENT BIT COUNT
1552  003736  001360  ;173736 001360          BNE     CRCLOP          ;ONCE MORE
1553  003740  005304  ;173740 005304          DEC     R4              ;DECREMENT COUNT
1554  003742  001345  ;173742 001345          BNE     GET             ;ONCE MORE
1555  003744  000207  ;173744 000207          RTS     PC              ;RETURN
1556                                  ;
1557                                  ;       SECONDARY PROGRAM REQEST MSG
1558                                  ;
1559  003746  113226  ;173746 113226  RQMSG:  .BYTE   $SYN,$SYN,$SYN,$SYN
1560  003750  113226  ;173750 113226
1561  003752  002220  ;173752 002220          .BYTE   $DLE,4,0,0,0,1
1562  003754  000000  ;173754 000000
1563  003756  000400  ;173756 000400
1564  003760  050055  ;173760 050055          .BYTE   55,120
1565  003762  001010  ;173762 001010          .BYTE   10              ;REQ SEC PGM CODE
1566                  ;                       .BYTE   2               ;DEVICE CODE
1567  003764  000001  ;173764 000001          .BYTE   $STADR          ;STATION ADDR
1568                  ;                       .BYTE   0               ;FILL
1569  003766  030242  ;173766 030242          .BYTE   242,60          ;FOR STADR=1
1570                                  ;       NOTE:NODEV AND DEVTAB MUST BE IN THIS ORDER
1571                                  ;       DO NOT SEPARATE THEM
1572                                  ;
1573                                  ;       .EVEN
1574  003770  122243  ;173770 122243  NODEV:  CMPB    (R2)+,-(R3)     ;INC PTR-DEC CNT
1575  003772  000002  ;173772 000002          RTI                     ;RETURN FROM TRAP
1576                                  ;
1577                  ;173774         RQMSGE:                         ;END OF MSG-USE JUNK AS PADS
1578  003774  007407  ;173774 007407  DEVTAB: .BYTE   7               ;DJ-11
1579                  ;                       .BYTE   17              ;DH-11
1580  003776  END.YC:
1581  003776  003407  ;173776 003407          .BYTE   7               ;DQ-11
1582                  ;                       .BYTE   7               ;DU-11
1583                  ;174000         END:
1584                  ;       173400          .END    START1
```

```
1585  004000  MAP.YD:
1586          ;THE FOLLOWING IS A REPRODUCTION
1587          ;OF THE ROM PROGRAM FOR BM873YD.
1588          ;IT IS HERE FOR COMPARISON TO THE
1589          ;ACTUAL ROM AND FOR REFERENCE
1590          ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
1591          ;BM873-YD.P11
1592
1593                                  ;       THIS CODE IS TO BE BLASTED INTO PROMS ON THE BM873-YD BOARD.
1594                                  ;       WRITTEN BY DAVID M. ROSENBERG           OCTOBER 1974
1595                                  ;REGISTER DEFINITIONS
1596
1597          ;       000000  R0=%0                   ;GENERAL PURPOSE REGISTER 0
1598          ;       000001  R1=%1                   ;GENERAL PURPOSE REGISTER 1
1599          ;       000002  R2=%2                   ;GENERAL PURPOSE REGISTER 2
1600          ;       000003  R3=%3                   ;GENERAL PURPOSE REGISTER 3
1601          ;       000004  R4=%4                   ;GENERAL PURPOSE REGISTER 4
1602          ;       000005  R5=%5                   ;GENERAL PURPOSE REGISTER 5
1603          ;       000006  SP=%6                   ;STACK POINTER (REGISTER R6)
1604          ;       000007  PC=%7                   ;PROGRAM COUNTER (REGISTER R7)
1605
1606
1607                                  ;SYMBOL DEFINITIONS
1608
1609          ;       177776  PS=177776               ;PROCESSOR STATUS REGISTER
1610          ;       177570  SWR=177570              ;FRONT PANEL SWITCH REGISTER
1611          ;       000000  PR0=0*40                ;PRIORITY LEVEL 0
1612          ;       000040  PR1=1*40                ;PRIORITY LEVEL 1
1613          ;       000100  PR2=2*40                ;PRIORITY LEVEL 2
1614          ;       000140  PR3=3*40                ;PRIORITY LEVEL 3
1615          ;       000200  PR4=4*40                ;PRIORITY LEVEL 4
1616          ;       000240  PR5=5*40                ;PRIORITY LEVEL 5
1617          ;       000300  PR6=6*40                ;PRIORITY LEVEL 6
1618          ;       000340  PR7=7*40                ;PRIORITY LEVEL 7
1619          ;       000001  BIT0=000001
1620          ;       000002  BIT1=000002
1621          ;       000004  BIT2=000004
1622          ;       000010  BIT3=000010
1623          ;       000020  BIT4=000020
1624          ;       000040  BIT5=000040
1625          ;       000100  BIT6=000100
1626          ;       000200  BIT7=000200
1627          ;       000400  BIT8=000400
1628          ;       001000  BIT9=001000
1629          ;       002000  BIT10=002000
1630          ;       004000  BIT11=004000
1631          ;       010000  BIT12=010000
1632          ;       020000  BIT13=020000
1633          ;       040000  BIT14=040000
1634          ;       100000  BIT15=100000
1635
```

```
1636            ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 3
1637            ;BM873-YD.P11   BUTTON #1  -  BOOTSTRAP USING THE PDP-11 SWITCH REGISTER
1638
1639
1640            ;       173000  ROMORG  =       173000          ;SET ROM ORIGIN TO 773000
1641            ;       173000          .=ROMORG                ;BM873-YD OCCUPIES 773000-773777
1642
1643  004000  033727 ;173000  033727  BUTON1: BIT     @#SWR,#BIT0     ;IS RIGHTMOST BIT ON?
1644  004002  177570 ;173002  177570
1645  004004  000001 ;173004  000001
1646  004006  001010 ;173006  001010          BNE     LOWBIT          ;IF THE BIT IS ON, BRANCH
1647  004010  013707 ;173010  013707          MOV     @#SWR,PC        ;JUMP TO THE ADDRESS IN THE SWITCH REGISTER
1648  004012  177570 ;173012  177570
1649                                                                  ;WITHOUT HAVING TOUCHED ANY OF R0 - R6
1650
1651  004014  111704 ;173014  111704  BUTON3: MOVB    (PC),R4         ;R4 = 1 INDICATES THAT BUTTON #3 WAS PRESSED
1652  004016  005001 ;173016  005001          CLR     R1              ;SET UNIT NUMBER TO ZERO
1653  004020  005005 ;173020  005005          CLR     R5              ;CLEAR "LOGICAL SWITCH REGISTER"
1654  004022  000424 ;173022  000424          BR      TCBOOT          ;DO A DEFAULT BOOT STRAP FROM DECTAPE
1655
1656  004024  173000 ;173024  173000          .WORD   ROMORG,PR7
1657  004026  000340 ;173026  000340
1658
1659  004030  013701 ;173030  013701  LOWBIT: MOV     @#SWR,R1        ;R1 IS A COPY OF THE SWITCH REGISTER
1660  004032  177570 ;173032  177570
1661  004034  106301 ;173034  106301          ASLB    R1              ;LEFT-ALIGN SPEED FIELD IN RIGHT BYTE
1662  004036  122701 ;173036  122701          CMPB    #16*20,R1       ;IS THE SPEED 16 OR 17?
1663  004040  000340 ;173040  000340
1664  004042  101404 ;173042  101404          BLOS    UNITNO          ;IF SPEED IS 16 OR 17, BRANCH
1665  004044  122701 ;173044  122701          CMPB    #3*20,R1        ;IS THE SPEED 0, 1, OR 2?
1666  004046  000060 ;173046  000060
1667  004050  101001 ;173050  101001          BHI     UNITNO          ;IF THE SPEED IS 0, 1, OR 2, BRANCH
1668  004052  005001 ;173052  005001          CLR     R1              ;SPEED WAS 3-15; SET UNIT NUMBER = 0
1669  004054  000301 ;173054  000301  UNITNO: SWAB    R1              ;MOVE UNIT NUMBER TO BITS 0-2
1670
1671                                          ; IT IS POSSIBLE TO MANUALLY SET THE DESIRED BOOTSTRAP UNIT NUMBER
1672                                          ; INTO THE RIGHTMOST THREE BITS OF R1, SET THE PDP-11 FRONT PANEL
1673                                          ; SWITCH REGISTER, AND THEN JUMP INTO THE ROM CODE AT THIS POINT.
1674
1675  004056  042701 ;173056  042701          BIC     #^C7,R1         ;ISOLATE UNIT NUMBER IN R1
1676  004060  177770 ;173060  177770
1677  004062  013705 ;173062  013705          MOV     @#SWR,R5        ;R5 IS NOW THE "LOGICAL SWITCH REGISTER"
1678  004064  177570 ;173064  177570
1679  004066  005004 ;173066  005004          CLR     R4              ;R4 = 0 INDICATES THAT BUTTON #1 WAS PRESSED
1680  004070  105705 ;173070  105705          TSTB    R5              ;SHOULD WE BOOT FROM DECTAPE OR RH11/RP04?
1681  004072  100507 ;173072  100507          BMI     RPBOOT          ;IF BIT 7 WAS ONE, BRANCH OFF TO THE RH11/RP04
1682                                                                  ;OTHERWISE, FALL THROUGH TO THE DECTAPE
```

```
1683            ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 4
1684            ;BM873-YD.P11   DECTAPE BOOTSTRAP AND DUMP ROUTINES
1685
1686
1687            ;       177344  TCWC    =       177344          ;TC11 DECTAPE WORD COUNT REGISTER
1688            ;       000001  TCGO    =       1               ;TC11 "GO" BIT
1689            ;       000002  TCRNUM  =       1*2             ;TC11 "READ BLOCK NUMBER" FUNCTION
1690            ;       000004  TCREAD  =       2*2             ;TC11 "READ DATA" FUNCTION
1691            ;       000014  TCWRIT  =       6*2             ;TC11 "WRITE DATA" FUNCTION
1692            ;       004000  TCREV   =       4000            ;MOVE DECTAPE IN REVERSE DIRECTION
1693
1694                                            ; BOOTSTRAP (FROM DECTAPE) PARAMETERS
1695            ;       000400  TCBWDC  =       ^D256           ;WORD COUNT FOR THE SECONDARY BOOTSTRAP
1696            ;       000000  TCBEND  =       0               ;WHICH END OF THE DECTAPE (0 = FRONT; 1 = BACK)
1697
1698                                            ; DUMP (TO DECTAPE) PARAMETERS
1699            ;       070000  TCDWDC  =       ^D28672         ;WORD COUNT FOR THE CORE DUMP TO DECTAPE
1700            ;       000001  TCDEND  =       1               ;WHICH END OF THE DECTAPE (0 = FRONT; 1 = BACK)
1701
1702                                            ; GENERAL (BOOTSTRAP AND DUMP) DECTAPE PARAMETER
1703            ;       000024  TCRTRY  =       ^D20            ;NUMBER OF RETRIES IN CASE OF ERROR
1704
1705  004074  012700 ;173074  012700  TCBOOT: MOV     #<TCBEND*TCREV>!TCREAD!TCGO,R0  ;SET UP DATA-TRANSFER COMMAND
1706  004076  000005 ;173076  000005
1707  004100  012702 ;173100  012702          MOV     #-TCBWDC,R2     ;SET WORD COUNT TO 256 (512 BYTES)
1708  004102  177400 ;173102  177400
1709  004104  012703 ;173104  012703          MOV     #<<1-TCBEND>*TCREV>!TCRNUM!TCGO,R3      ;SET UP POSITION COMMAND
1710  004106  004003 ;173106  004003
1711  004110  000301 ;173110  000301          SWAB    R1              ;BRING UNIT NUMBER INTO THE LEFT BYTE
1712  004112  050103 ;173112  050103          BIS     R1,R3           ;PUT UNIT NUMBER INTO POSITIONING COMMAND
1713  004114  050100 ;173114  050100          BIS     R1,R0           ;PUT UNIT NUMBER INTO DATA-TRANSFER COMMAND
1714  004116  012701 ;173116  012701  TCSTRT: MOV     #TCWC,R1        ;R1 NOW POINTS TO TC11 WORD COUNT REGISTER
1715  004120  177344 ;173120  177344
1716  004122  012706 ;173122  012706  TCLOOP: MOV     #TCRTRY,SP      ;INITIALIZE RETRY COUNT IN SP
1717  004124  000024 ;173124  000024
1718  004126  005705 ;173126  005705  TCBGIN: TST     R5              ;TEST "INDEFINITE RETRY" BIT
1719  004130  100404 ;173130  100404          BMI     TCRSET          ;BRANCH IF "INDEFINITE RETRY" IS ENABLED
1720  004132  005306 ;173132  005306          DEC     SP              ;DECREMENT RETRY COUNT
1721  004134  100002 ;173134  100002          BPL     TCRSET          ;BRANCH IF RETRY COUNT NOT EXHAUSTED
1722  004136  000000 ;173136  000000  TCHALT: HALT                    ;RETRY COUNT IS EXHAUSTED FOR DECTAPE OPERATION
1723  004140  000770 ;173140  000770          BR      TCLOOP          ;HE PRESSED "CONTINUE", SO TRY AGAIN
1724  004142  000005 ;173142  000005  TCRSET: RESET                   ;STOP ANYTHING IN PROGRESS, FOR NEXT TRY
1725  004144  010341 ;173144  010341          MOV     R3,-(R1)        ;INITIATE DECTAPE POSIONING OPERATION
1726  004146  005711 ;173146  005711  TCWAIT: TST     (R1)            ;TEST FOR AN "ERROR"
1727  004150  100376 ;173150  100376          BPL     TCWAIT          ;LOOP UNTIL AN "ERROR" IS DETECTED
1728  004152  005721 ;173152  005721          TST     (R1)+           ;MAKE R1 POINT TO THE WORD COUNT REGISTER
1729  004154  005761 ;173154  005761          TST     -4(R1)          ;IS THE ERROR "ENDZONE"?
1730  004156  177774 ;173156  177774
1731  004160  100362 ;173160  100362          BPL     TCBGIN          ;IF NOT, BRANCH BACK TO TRY AGAIN
1732  004162  010211 ;173162  010211          MOV     R2,(R1)         ;SET UP WORD COUNT FOR DATA-TRANSFER
1733  004164  010041 ;173164  010041          MOV     R0,-(R1)        ;INITIATE THE DATA-TRANSFER OPERATION
1734  004166  105711 ;173166  105711  TCDONE: TSTB    (R1)            ;TEST FOR "DONE"
1735  004170  100376 ;173170  100376          BPL     TCDONE          ;LOOP UNTIL THE "DONE" BIT SETS
1736  004172  005721 ;173172  005721          TST     (R1)+           ;WAS AN "ERROR" DETECTED?
```

```
 1737  004174  100754 ;173174  100754          BMI     TCBGIN          ;IF SO, BRANCH BACK AND TRY AGAIN
 1738  004176  005741 ;173176  005741          TST     -(R1)           ;MAKE R1 POINT TO THE COMMAND REGISTER
 1739  004200  105011 ;173200  105011          CLRB    (R1)            ;STOP ALL DECTAPE MOTION
 1740  004202  122700 ;173202  122700          CMPB    #TCREAD!TCGO,R0 ;WAS THIS A "NORMAL READ" OPERATION?
 1741  004204  000005 ;173204  000005
 1742  004206  001001 ;173206  001001          BNE     TCSTOP          ;IF NOT, GO STOP
 1743  004210  000137 ;173210  000137  GOTO0:  JMP     @(PC)+          ;JUMP TO PDP-11 LOCATION ZERO
 1744  004212  000000 ;173212  000000  TCSTOP: HALT                    ;SUCCESSFUL COMPLETION OF A "NON-READ" OPERATION
 1745  004214  000776 ;173214  000776          BR      TCSTOP          ;SO THAT PRESSING "CONTINUE" WON'T GO ANYWHERE
```

```
1746                    ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 5
1747                    ;BM873-YD.P11   DECTAPE BOOTSTRAP AND DUMP ROUTINES
1748
1749  004216  010037 ;173216  010037  TCDUMP: MOV     R0,@#ROTOR7     ;SAVE R0 IN PDP-11 MEMORY LOCATION 40
1750  004220  000040 ;173220  000040
1751  004222  000402 ;173222  000402          BR      TCCONT          ;BRANCH AROUND REQUIRED INTERRUPT VECTOR
1752
1753  004224  173000 ;173224  173000          .WORD   ROMORG,PR7
1754  004226  000340 ;173226  000340
1755
1756  004230  010700 ;173230  010700  TCCONT: MOV     PC,R0           ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
1757  004232  000410 ;173232  000410          BR      REGSAV          ;GO TO THE "REGISTER SAVING" SUBROUTINE
1758  004234  012700 ;173234  012700          MOV     #<TCDEND*TCREV>!TCWRIT!TCGO,R0  ;SET UP (WRITE) TRANSFER COMMAND
1759  004236  004015 ;173236  004015
1760  004240  012702 ;173240  012702          MOV     #-TCDWDC,R2     ;SET WORD-COUNT TO 28K WORDS
1761  004242  110000 ;173242  110000
1762  004244  012703 ;173244  012703          MOV     #<<1-TCDEND>*TCREV>!TCRNUM!TCGO,R3      ;SET UP POSITION COMMAND
1763  004246  000003 ;173246  000003
1764  004250  005005 ;173250  005005          CLR     R5              ;CLEAR "INDEFINITE RETRY" BIT
1765  004252  000721 ;173252  000721          BR      TCSTRT          ;BRANCH INTO DECTAPE ROUTINE
1766
1767
1768
1769
1770                                    ; THE FOLLOWING SUBROUTINE IS USED TO SAVE THE PDP-11 GENERAL REGISTERS
1771                                    ; IN PDP-11 MEMORY LOCATIONS 40-57.
1772
1773                                    ; THE CALLING SEQUENCE IS AS FOLLOWS:           MOV     R0,@#ROTOR7
1774                                    ;                                               MOV     PC,R0
1775                                    ;                                               BR      REGSAV
1776                                    ;                                               RETURN HERE
1777
1778  004254  010137 ;173254  010137  REGSAV: MOV     R1,@#ROTOR7+2   ;SAVE R1 IN MEMOTY LOCATION 42
1779  004256  000042 ;173256  000042
1780  004260  012701 ;173260  012701          MOV     #ROTOR7+4,R1    ;R1 NOW POINTS TO MEMORY LOCATION 44
1781  004262  000044 ;173262  000044
1782  004264  010221 ;173264  010221          MOV     R2,(R1)+        ;SAVE R2 IN MEMORY LOCATION 44
1783  004266  010321 ;173266  010321          MOV     R3,(R1)+        ;SAVE R3 IN MEMORY LOCATION 46
1784  004270  010421 ;173270  010421          MOV     R4,(R1)+        ;SAVE R4 IN MEMORY LOCATION 50
1785  004272  010521 ;173272  010521          MOV     R5,(R1)+        ;SAVE R5 IN MEMORY LOCATION 52
1786  004274  010621 ;173274  010621          MOV     SP,(R1)+        ;SAVE SP IN MEMORY LOCATION 54
1787  004276  010021 ;173276  010021          MOV     R0,(R1)+        ;SAVE PC IN MEMORY LOCATION 56
1788  004300  000160 ;173300  000160          JMP     2(R0)           ;RETURN TO THE CALLING ROUTINE
1789  004302  000002 ;173302  000002
1790
```

```
1791            ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 6
1792            ;BM873-YD.P11   RH11/RP04 BOOTSTRAP AND DUMP ROUTINES
1793
1794
1795            ;       176700  RPCS1   =       176700  ;ADDRESS OF RH11/RP04 CONTROL & STATUS REGISTER 1
1796            ;       000002  RPWC    =       2       ;OFFSET TO RH11/RP04 WORD COUNT REGISTER
1797            ;       000006  RPDA    =       6       ;OFFSET TO RH11/RP04 TRACK & SECTOR ADDRESS REGISTER
1798            ;       000010  RPCS2   =       10      ;OFFSET TO RH11/RP04 CONTROL & STATUS REGISTER 2
1799            ;       000012  RPDS    =       12      ;OFFSET TO RH11/RP04 DRIVE STATUS REGISTER
1800            ;       000032  RPOF    =       32      ;OFFSET TO RH11/RP04 OFFSET REGISTER (CONTAINING FMT22)
1801            ;       000034  RPDC    =       34      ;OFFSET TO RH11/RP04 DESIRED CYLINDER REGISTER
1802
1803            ;       040000  RPTRE   =       BIT14   ;"TRANSFER ERROR" BIT IN RPCS1
1804            ;       020000  RPMCPE  =       BIT13   ;"MASSBUS CONTROL BUS PARITY ERROR" BIT IN RPCS1
1805            ;       004000  RPDVA   =       BIT11   ;"DRIVE AVAILABLE" BIT IN RPCS1
1806            ;       100000  RPATA   =       BIT15   ;"ATTENTION ACTIVE" BIT IN RPDS
1807            ;       040000  RPERR   =       BIT14   ;"COMPOSITE ERROR" BIT IN RPDS
1808            ;       010000  RPFMT   =       BIT12   ;"FMT22" (16-BIT WORDS) BIT IN RPOF
1809
1810            ;       000021  RPPRST  =       21      ;READ-IN PRESET
1811            ;       000061  RPWRIT  =       61      ;WRITE DATA
1812            ;       000071  RPREAD  =       71      ;READ DATA
1813
1814            ;       000000  RPBFMT  =       0       ;BOOTSTRAP FORMAT (0 = 18-BIT WORDS; 2 = 16-BIT WORDS)
1815            ;       000400  RPBWDC  =       ^D256   ;WORD COUNT FOR THE SECONDARY BOOTSTRAP FROM THE RP04
1816            ;       000626  RPBCYL  =       ^D406   ;BOOTSTRAP CYLINDER NUMBER
1817            ;       000000  RPBTRK  =       0       ;BOOTSTRAP TRACK NUMBER
1818            ;       000000  RPBSCT  =       0       ;BOOTSTRAP SECTOR NUMBER
1819
1820            ;       000000  RPDFMT  =       0       ;DUMP FORMAT (0 = 18-BIT WORDS; 2 = 16-BIT WORDS)
1821            ;       070000  RPDWDC  =       ^D28672 ;WORD COUNT FOR THE CORE DUMP TO THE RP04
1822            ;       000631  RPDCYL  =       ^D409   ;DUMP CYLINDER NUMBER
1823                                                    ; THE FOLLOWING TWO ASSIGNMENTS PUT THE DUMP AT THE VERY END OF THE CYLINDER
1824            ;       000015  RPDTRK  =       ^D18-<<RPDWDC-1>/<<^D20+RPDFMT>*^D256>> ;DUMP TRACK NUMBER
1825            ;       000010  RPDSCT  =       ^D19+RPDFMT-<<<RPDWDC-1>/^D256>-<<^D18-RPDTRK>*<^D20+RPDFMT>>>
1826
1827
1828
1829  004304  111704  ;173304  111704  BUTON2: MOVB    (PC),R4         ;R4 = 5 INDICATES THAT BUTTON #2 WAS PRESSED
1830  004306  005005  ;173306  005005          CLR     R5              ;CLEAR "LOGICAL SWITCH REGISTER"
1831  004310  005001  ;173310  005001          CLR     R1              ;SET UNIT NUMBER TO ZERO
1832
1833  004312  012700  ;173312  012700  RPBOOT: MOV     #<RPREAD*400>!<RPBSCT*10>,R0
1834  004314  034400  ;173314  034400
1835  004316  012702  ;173316  012702          MOV     #-RPBWDC,R2
1836  004320  177400  ;173320  177400
1837  004322  012703  ;173322  012703          MOV     #<RPBFMT*40000>!<RPBTRK*2000>!RPBCYL,R3
1838  004324  000626  ;173324  000626
1839  004326  050100  ;173326  050100          BIS     R1,R0           ;PUT THE UNIT NUMBER INTO R0
1840  004330  012701  ;173330  012701  RPSTRT: MOV     #RPCS1,R1       ;SET R1 TO THE LOWEST ADDRESS USED BY THE RH11
1841  004332  176700  ;173332  176700
```

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

```
1896  004502  000000 ;173502  000000          HALT                    ;"000240" WAS NOT READ INTO LOCATION ZERO
1897  004504  000641 ;173504  000641          BR      GOTOO           ;BRANCH TO LOCATION ZERO
1898
1899
1900  004506  010037 ;173506  010037  RPDUMP: MOV     R0,@#ROTOR7     ;SAVE R0 IN PDP-11 MEMORY LOCATION "ROTOR7"
1901  004510  000040 ;173510  000040
1902  004512  010700 ;173512  010700          MOV     PC,R0           ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
1903  004514  000657 ;173514  000657          BR      REGSAV          ;GO TO THE "REGISTER SAVING" SUBROUTINE
1904  004516  012700 ;173516  012700          MOV     #<RPWRIT*400>!<RPDSCT*10>,R0
1905  004520  030500 ;173520  030500
1906  004522  012702 ;173522  012702          MOV     #-RPDWDC,R2
1907  004524  110000 ;173524  110000
1908  004526  012703 ;173526  012703          MOV     #<RPDFMT*40000>!<RPDTRK*2000>!RPDCYL,R3
1909  004530  032631 ;173530  032631
1910  004532  000676 ;173532  000676          BR      RPSTRT
1911
```

```
1912          ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 8
1913          ;BM873-YD.P11   DTE20 DEVICE REGISTER AND BIT DEFINITIONS
1914
1915
1916          ;       174400  DTEBAS=174400           ;BASE OF (FIRST) DTE20 DEVICE REGISTER BLOCK
1917          ;       000040  DTESIZ=000040           ;SPACING BETWEEN CONSECUTIVE DTE20'S
1918          ;       000004  DTEMAX=4                ;MAXIMUM NUMBER OF DTE20'S ON ONE PDP-11
1919
1920
1921                                                  ;OFFSETS FROM THE BASE OF THE DTE20 DEVICE REGISTER BLOCK
1922                                                  ;TO SPECIFIC 10/11 INTERFACE RAM LOCATIONS AND REGISTERS.
1923
1924                                          ; THE FIRST 12 REGISTERS ARE NOT INITIALIZED BY "INIT" (BECAUSE THEY ARE IN RAMS
1925          ;       000000  DLYCNT=00               ;DELAY COUNT                        (ADDRESS XXXX00)
1926          ;       000002  DEXWD3=02               ;DEPOSIT OR EXAMINE WORD 3          (ADDRESS XXXX02)
1927          ;       000004  DEXWD2=04               ;DEPOSIT OR EXAMINE WORD 2          (ADDRESS XXXX04)
1928          ;       000006  DEXWD1=06               ;DEPOSIT OR EXAMINE WORD 1          (ADDRESS XXXX06)
1929          ;       000010  TENAD1=10               ;10 ADDRESS WORD 1 FOR DEX          (ADDRESS XXXX10)
1930          ;       000012  TENAD2=12               ;10 ADDRESS WORD 2 FOR DEX          (ADDRESS XXXX12)
1931          ;       000014  TO10BC=14               ;TO10 BYTE COUNT                    (ADDRESS XXXX14)
1932          ;       000016  TO11BC=16               ;TO11 BYTE COUNT                    (ADDRESS XXXX16)
1933          ;       000020  TO10AD=20               ;TO10 PDP11 MEMORY ADDRESS          (ADDRESS XXXX20)
1934          ;       000022  TO11AD=22               ;TO11 PDP11 MEMORY ADDRESS          (ADDRESS XXXX22)
1935          ;       000024  TO10DT=24               ;TO10 PDP11 DATA WORD               (ADDRESS XXXX24)
1936          ;       000026  TO11DT=26               ;TO11 PDP11 DATA WORD               (ADDRESS XXXX26)
1937
1938                                          ; THE LAST 4 REGISTERS ARE INITIALIZED BY "INIT" (BECAUSE THEY ARE IN FLIP-FLOPS
1939          ;       000030  DIAG1=30                ;DIAGNOSTIC WORD 1                  (ADDRESS XXXX30)
1940          ;       000032  DIAG2=32                ;DIAGNOSTIC WORD 2                  (ADDRESS XXXX32)
1941          ;       000034  STATUS=34               ;10/11 INTERFACE STATUS WORD        (ADDRESS XXXX34)
1942          ;       000036  DIAG3=36                ;DIAGNOSTIC WORD 3                  (ADDRESS XXXX36)
1943
1944
1945                                          ;  THE FOLLOWING ARE THE ADDRESSES OF THE DTE20 INTERRUPT VECTORS
1946
1947          ;       000774  DTEIV1=774              ;INTERRUPT VECTOR FOR DTE20 #1
1948          ;       000770  DTEIV2=770              ;INTERRUPT VECTOR FOR DTE20 #2
1949          ;       000764  DTEIV3=764              ;INTERRUPT VECTOR FOR DTE20 #3
1950          ;       000760  DTEIV4=760              ;INTERRUPT VECTOR FOR DTE20 #4
1951
1952
1953                                          ;  BIT ASSIGNMENTS FOR VARIOUS DTE20 REGISTERS USED BY THIS ROM CODE
1954
1955
1956                                                  ;BIT ASSIGNMENTS FOR TO10BC
1957
1958          ;       100000  INT11=BIT15             ;SET DONE AND INTERRUPT BOTH 10 AND 11
1959
1960                                                  ;BIT ASSIGNMENTS FOR TO11BC
1961
1962          ;       100000  INT10=BIT15             ;SET DONE AND INTERRUPT BOTH 10 AND 11
1963          ;       040000  ZSTOP=BIT14             ;STOP ON NULL (ZERO) CHARACTER
1964          ;       020000  TO11BM=BIT13            ;BYTE SIZE FOR TO-11 BYTE TRANSFERS
1965
```

```
1966          ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 9
1967          ;BM873-YD.P11   DTE20 DEVICE REGISTER AND BIT DEFINITIONS
1968
1969                                                          ;BIT ASSIGNMENTS FOR DIAG2 (WRITE)
1970          ;       000100  DRESET=BIT6              ;PERFORM DIAGNOSTIC CLEAR
1971
1972                                                          ;BIT ASSIGNMENTS FOR DIAG3 (READ)
1973          ;       000020  DUPE=BIT4               ;DATO UNIBUS PARITY ERROR
1974          ;       000004  DURE=BIT2               ;DATO UNIBUS RECEIVE ERROR
1975          ;       000002  NUPE=BIT1               ;NPR UNIBUS PARITY ERROR
1976
1977                                                          ;BIT ASSIGNMENTS FOR DIAG3 (WRITE)
1978
1979          ;       000020  CDD=BIT4                ;CLEAR DUPE AND DURE ERROR FLAGS
1980          ;       000002  CNUPE=BIT1              ;CLEAR NUPE ERROR FLAG
1981          ;       000001  T010BM=BIT0             ;BYTE SIZE FOR TO-10 BYTE TRANSFER
1982
1983                                                          ;BIT ASSIGNMENTS FOR STATUS (WRITE)
1984
1985          ;       100000  DON10S=BIT15            ;SET TO10 DONE
1986          ;       040000  DON10C=BIT14            ;CLEAR TO10 DONE
1987          ;       020000  ERR10S=BIT13            ;SET TO10 ERROR
1988          ;       010000  ERR10C=BIT12            ;CLEAR TO10 ERROR
1989          ;       004000  INT11S=BIT11            ;RING THE PDP-11'S DOORBELL (INTERRUPTS THE -11)
1990          ;       002000  INT11C=BIT10            ;STOP RINGING THE PDP-11'S DOORBELL
1991          ;       001000  PERCLR=BIT9             ;CLEAR -11 MEMORY PARITY ERROR
1992          ;       000400  INT10S=BIT8             ;RING THE PDP-10'S DOORBELL (INTERRUPTS THE -10)
1993          ;       000200  DON11S=BIT7             ;SET TO11 DONE
1994          ;       000100  DON11C=BIT6             ;CLEAR TO11 DONE
1995          ;       000040  INTRON=BIT5             ;ENABLE DTE20 INTERRUPTS TO THE -11
1996          ;       000020  EBUSPC=BIT4             ;CLEAR "EBUS PARITY ERROR"
1997          ;       000010  INTROF=BIT3             ;DISABLE THE PDP-11 INTERUPTS
1998          ;       000004  EBUSPS=BIT2             ;SET "EBUS PARITY ERROR"
1999          ;       000002  ERR11S=BIT1             ;SET TO11 ERROR
2000          ;       000001  ERR11C=BIT0             ;CLEAR TO11 ERROR
2001
2002                                                          ;BIT ASSIGNMENTS FOR STATUS (READ)
2003
2004          ;       100000  T010DN=BIT15            ;TO10 DONE
2005          ;       020000  T010ER=BIT13            ;TO 10 ERROR (NPR TIMEOUT OR BUS ERROR)
2006          ;       010000  RAMISO=BIT12            ;RAM WORD READ IS ALL ZEROS
2007          ;       004000  T011DB=BIT11            ;1 = THE PDP11'S DOORBELL IS RINGING
2008          ;       002000  DXWRD1=BIT10            ;DEPOSIT OR EXAMINE WORD ONE
2009          ;       001000  MPE11=BIT9              ;PARITY ERROR WITHIN PDP-11 MEMORY
2010          ;       000400  T010DB=BIT8             ;1 = THE PDP-10'S DOORBELL IS RINGING
2011          ;       000200  T011DN=BIT7             ;TO11 DONE
2012          ;       000100  EBSEL=BIT6              ;E BUFFER SELECT
2013          ;       000040  NULSTP=BIT5             ;NULL STOP
2014          ;       000020  BPARER=BIT4             ;EBUS PARITY ERROR
2015          ;       000010  RSTRCT=BIT3             ;THIS PDP-11 IS "RESTRICTED"
2016          ;       000004  DEXDON=BIT2             ;DEPOSIT OR EXAMINE DONE
2017          ;       000002  T011ER=BIT1             ;TO 11 ERROR (NPR TIMEOUT OR BUS ERROR)
2018          ;       000001  INTSON=BIT0             ;DTE20 INTERRUPTS (TO THE -11) ARE ENABLED
```

```
2019          ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
2020          ;BM873-YD.P11   PROCEDURE BY WHICH THE PDP-10 BOOTSTRAPS AND/OR DUMPS THE PDP-11
2021
2022
2023
2024
2025
2026
2027                          THE FOLLOWING IS THE PROCEDURE WHICH THE KL10 EXECUTES IN ORDER
2028                          TO DUMP AND/OR BOOTSTRAP THE PDP-11 THROUGH THE DTE20:
2029
2030
2031                          1. CLEAR THE DTE20 AND INITIATE A BM873 BUTTON #4 BOOTSTRAP OPERATION
2032                                  -   CONO [SR11B!CL11PT!CLTO11!CLTO10!PILDEN]
2033
2034                          2. WAIT TO SEE PDP-11 POWER FAIL (AC LOW = TRUE)   -   CONI [DEAD11] = 1
2035
2036                          3. WAIT TO SEE PDP-11 POWER RECOVER (AC LOW = FALSE)   -   CONI [DEAD11] = 0
2037
2038                          4. WAIT AT LEAST ANOTHER 150 MILLISECONDS AND THEN CLEAR THE RELOAD -11 BUTTON
2039                                  -   CONO [CR11B]
2040
2041                          5. SET BYTE COUNTER TO A SPECIAL CODE (1365 OCTAL)   -   DATAO [1365]
2042
2043                          6. RING PDP-11'S DOORBELL   -   CONO[TO11DB]
2044
2045                          7. WAIT UNTIL "-10 RINGING -11'S DOORBELL" IS TURNED OFF BY THE -11
2046                                (I.E. UNTIL   CONI[TO11DB]   BECOMES ZERO).
2047
2048                          8. ENABLE THE DTE20 TO USE PI 0 INTERRUPTS
2049                                (I.E. SET   CONO[PILDEN!PIOENB]).
2050
2051                          9. SET UP THE TO-10 BYTE POINTER (IN THE EPT) FOR THE FIRST 3.5K.
2052
2053                          10. SET UP THE BYTE COUNTER FOR THE FIRST 3.5K, INDICATING
2054                                "INTERRUPT -10 ONLY"   -   DATAO [1000]
2055
2056                          11. WAIT FOR "TO-10 DONE" OR "TO-10 ERROR"   -   CONI [TO10DN!TO10ER]
2057
2058                          12. NOTE WHETHER THERE WAS AN ERROR (CONI [TO10ER]) AND THEN TURN OFF
2059                                TO10DN AND TO10ER   -   CONO [CLTO10]. IF ERROR, GO TO STEP 17.
2060
2061                          13. IF END OF 28K, GO TO STEP 17.
2062
2063                          14. SET UP TO-10 BYTE POINTER (IN THE EPT) FOR THE NEXT 3.5K.
2064
2065                          15. SET UP THE BYTE COUNTER FOR THE NEXT 3.5K INDICATING
2066                                "INTERRUPT -10 ONLY" (DATAO [1000]), UNLESS THIS IS THE
2067                                LAST 3.5K (OF 28K), IN WHICH CASE INDICATE "INTERRUPT
2068                                BOTH PROCESSORS" (DATAO [TO10IB!1000]).
2069
2070                          16. GO TO STEP 11.
2071
2072
```

```
2073          ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
2074          ;BM873-YD.P11   PROCEDURE BY WHICH THE PDP-10 BOOTSTRAPS AND/OR DUMPS THE PDP-11
2075
2076
2077                          17. SET UP TO-11 BYTE POINTER (IN THE EPT) FOR "PDP-11 BOOTSTRAP".
2078                              NOTE THAT THE FIRST WORD OF THIS "PDP-11 BOOTSTRAP" MUST
2079                              BE THE BIT PATTERN 000240 (A PDP-11 NOP INSTRUCTION).
2080
2081                          18. RING THE PDP-11'S DOORBELL   -   CONO [TO11DB]
2082
2083                          19. WAIT FOR EITHER TO11DB TO GO OFF (CONI[TO11DB] = 0),
2084                              OR TO10DB TO COME ON (CONI[TO10DB] = 1).
2085
2086                          20. IF NO ERROR WAS NOTED IN STEP 12, TO11DB SHOULD GO OFF
2087                              (TO10DB COMING ON INDICATES A MASSIVE SCREWUP).
2088                              IF AN ERROR WAS NOTED IN STEP 12, TO11DB GOING OFF INDICATES
2089                              THAT THE ERROR WAS "NON-FATAL" (NON-EX-MEM OR -11 MEMORY
2090                              PARITY) AND THE -11 IS PROCEEDING. TO10DB COMING ON INDICATES
2091                              THAT THE ERROR WAS "FATAL" AND THE -11 IS HALTED AT LOCATION 173714.
2092                              IN THIS LATTER CASE THE -10 MUST RESTART FROM STEP 1.
2093
2094                          21. IF TO11DB WENT OFF, WAIT FOR "TO-11 DONE" OR "TO-11 ERROR"
2095                              -   CONI [TO11DN!TO11ER]
2096
2097                          22. NOTE WHETHER THERE WAS AN ERROR   -   CONI [TO11ER]
2098
2099                          23. TURN OFF TO11DN AND TO11ER AND RING THE PDP-11'S DOORBELL
2100                              -   CONO [TO11DB!CLTO11]
2101
2102                          24. WAIT FOR EITHER TO11DB TO GO OFF (CONI[TO11DB] = 0),
2103                              OR TO10DB TO COME ON (CONI[TO10DB] = 1).
2104
2105                          25. TO11DB GOING OFF INDICATES THAT THE PDP-11 FOUND NO ERRORS
2106                              AND IS TRANSFERRING CONTROL TO THE CODE WHICH WAS JUST
2107                              RECEIVED FROM THE -10. IN THIS CASE THE -10 SHOULD START
2108                              FOLLOWING THE PROTOCOL OF THIS CODE.
2109
2110                          26. TO10DB COMING ON INDICATES THAT THE PDP-11 HAS FOUND AN
2111                              ERROR (OR THAT THE FIRST WORD TRANSMITTED WASN'T THE
2112                              BIT PATTERN 000240), AND THE PDP-11 IS HALTED AT LOCATION 173766.
2113                              IN THIS CASE THE -10 MUST RESTART FROM STEP 1.
2114
2115
2116
2117
2118
2119
```

```
2120            ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
2121            ;BM873-YD.P11   BUTTON #4  -  BOOTSTRAP INITIATED BY THE PDP-10 (THROUGH DTE20)
2122
2123
2124            ;       000130  DTECOR  =       130             ;CORE ADDRESS INTO WHICH TO STORE DTE20 REGS.
2125            ;       000014  DTEREG  =       ↑D12            ;NUMBER OF DTE20 REGISTERS TO STORE
2126            ;       000400  DTEWDC  =       ↑D256           ;WORD COUNT FOR SECONDARY BOOTSTRAP FROM THE -10
2127                                            ; ENTER HERE WHEN THE DTE20 PRESSES BUTTON #4 (BOOTSTRAP INITIATED
2128                                            ;       BY THE PDP-10, THROUGH THE DTE20)
2129  004534  010037 ;173534  010037  BUTON4: MOV     R0,@#ROTOR7     ;SAVE R0 IN PDP-11 MEMORY LOCATION "ROTOR7"
2130  004536  000040 ;173536  000040
2131  004540  010700 ;173540  010700          MOV     PC,R0           ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
2132  004542  000644 ;173542  000644          BR      REGSAV          ;GO TO THE "REGISTER SAVING" SUBROUTINE
2133  004544  005005 ;173544  005005          CLR     R5              ;SET R5 = 0
2134  004546  012501 ;173546  012501          MOV     (R5)+,R1        ;SAVE LOCATION 0 IN R1
2135  004550  012503 ;173550  012503          MOV     (R5)+,R3        ;SAVE LOCATION 2 IN R3
2136  004552  012504 ;173552  012504          MOV     (R5)+,R4        ;SAVE LOCATION 4 IN R4
2137  004554  011500 ;173554  011500          MOV     (R5),R0         ;SAVE LOCATION 6 IN R0
2138  004556  012715 ;173556  012715          MOV     #PR7,(R5)       ;SET UP PRIORITY FOR NON-EX-MEM TRAP
2139  004560  000340 ;173560  000340
2140  004562  005745 ;173562  005745          TST     -(R5)           ;SET R5 = 4
2141  004564  012702 ;173564  012702  10$:    MOV     #DTEBAS-DTESIZ,R2
2142  004566  174340 ;173566  174340
2143  004570  010715 ;173570  010715          MOV     PC,(R5)         ;STORE ADDRESS FOR NON-EX-MEM TRAP
2144  004572  010506 ;173572  010506          MOV     R5,SP           ;SET STACK POINTER = 4
2145  004574  062702 ;173574  062702  11$:    ADD     #DTESIZ,R2      ;R2 POINTS TO THE NEXT DTE20
2146  004576  000040 ;173576  000040
2147  004600  105702 ;173600  105702          TSTB    R2
2148  004602  100770 ;173602  100770          BMI     10$             ;START LOOKING FROM THE BEGINNING AGAIN
2149  004604  032762 ;173604  032762          BIT     #TO11DB,STATUS(R2)      ;IS THIS -10 RINGING THE -11'S DOORBELL?
2150  004606  004000 ;173606  004000
2151  004610  000034 ;173610  000034
2152  004612  001770 ;173612  001770          BEQ     11$             ;IF IT IS NOT, GO LOOK FOR ANOTHER -10
2153  004614  026217 ;173614  026217          CMP     TO10BC(R2),(PC) ;CHECK FOR A CODE (1365) FROM THE PDP-10
2154  004616  000014 ;173616  000014                                  ;INDICATING THAT IT WANTS TO BOOTSTRAP THE -11
2155
2156  004620  001365 ;173620  001365          BNE     11$
2157                                          ; NOTE THAT AT THIS POINT R2 CONTAINS THE ADDRESS OF THE DEVICE REGISTER
2158                                          ;       BLOCK FOR THIS DTE20, THAT R5 = 4, AND THAT SP = 4
2159  004622  005725 ;173622  005725          TST     (R5)+           ;SET R5 = 6
2160  004624  010015 ;173624  010015          MOV     R0,(R5)         ;RESTORE THE CONTENTS OF LOCATION 6
2161  004626  010445 ;173626  010445          MOV     R4,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 4
2162  004630  010345 ;173630  010345          MOV     R3,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 2
2163  004632  010145 ;173632  010145          MOV     R1,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 0
2164                                          ; NOTE: AT THIS TIME R5 = 0. THIS FACT WILL BE USED LATER.
2165  004634  012700 ;173634  012700          MOV     #DTECOR,R0      ;R0 = CORE ADDRESS FOR STORING DTE20 REGISTERS
2166  004636  000130 ;173636  000130
2167  004640  010204 ;173640  010204          MOV     R2,R4
2168  004642  012420 ;173642  012420  7$:     MOV     (R4)+,(R0)+     ;SAVE THE NEXT DTE20 REGISTER IN CORE
2169  004644  022700 ;173644  022700          CMP     #<DTEREG*2>+DTECOR,R0   ;HAVE WE FINISHED YET?
2170  004646  000160 ;173646  000160
2171  004650  101374 ;173650  101374          BHI     7$              ;LOOP UNTIL WE HAVE FINISHED
```

```
                        ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
                        ;BM873-YD.P11   BUTTON #4  -  BOOTSTRAP INITIATED BY THE PDP-10 (THROUGH DTE20)
                        ;
004652  010201 ;173652  010201          MOV     R2,R1                   ;R1 = DTE20 DEVICE REGISTER BLOCK
004654  062701 ;173654  062701          ADD     #DIAG2,R1
004656  000032 ;173656  000032
004660  012721 ;173660  012721          MOV     #DRESET,(R1)+           ;DO A "DIAGNOSTIC CLEAR" OF THE DTE20,
004662  000100 ;173662  000100
                                        ; THE ABOVE  OPERATION IS NECESSARY TO CLEAR THE "BYTE COUNT LOADED" FLAG
                                        ; AND SIMULTANEOUSLY TO TURN OFF "-10 RINGING -11'S DOORBELL".
004664  005012 ;173664  005012          CLR     (R2)                    ;SET DTE20 FOR NO DELAY
004666  005062 ;173666  005062          CLR     TO10AD(R2)      ;START WRITING -11 MEMORY INTO THE -10.
004670  000020 ;173670  000020
004672  032711 ;173672  032711  6$:     BIT     #TO11DB,(R1)    ;HAS THE -10 RUNG THE -11'S DOORBELL?
004674  004000 ;173674  004000
004676  001775 ;173676  001775          BEQ     6$              ;LOOP UNTIL IT HAS.
004700  032762 ;173700  032762          BIT     #DUPE!DURE!NUPE,DIAG3(R2)       ;"FATAL" ERROR?
004702  000026 ;173702  000026
004704  000036 ;173704  000036
004706  001403 ;173706  001403          BEQ     8$              ;BRANCH IF NO "FATAL" ERROR
004710  012711 ;173710  012711          MOV     #TC10DB,(R1)    ;SIGNAL "FATAL" ERROR TO THE PDP-10
004712  000400 ;173712  000400
004714  000000 ;173714  000000  2$:     HALT                    ;HALT DUE TO "FATAL" ERROR
004716  012762 ;173716  012762  8$:     MOV     #DRESET,DIAG2(R2)       ;RESET AFTER POSSIBLE PDP-11
004720  000100 ;173720  000100
004722  000032 ;173722  000032
                                        ; MEMORY PARITY ERROR OR NON-EX-MEM ERROR, AND ALSO TURN OFF
                                        ; "-10 RINGING -11'S DOORBELL".
004724  005062 ;173724  005062  3$:     CLR     TO11AD(R2)      ;START INPUTTING AT LOCATION 0
004726  000022 ;173726  000022
004730  012762 ;173730  012762          MOV     #INT10!<<-DTEWDC>&7777>,TO11BC(R2)      ;READ IN 256 WORDS
004732  107400 ;173732  107400
004734  000016 ;173734  000016
004736  032711 ;173736  032711  1$:     BIT     #TO11DB,(R1)    ;HAS THE -10 RUNG THE -11'S DOORBELL?
004740  004000 ;173740  004000
004742  001775 ;173742  001775          BEQ     1$              ;LOOP UNTIL IT HAS.
004744  132711 ;173744  132711  4$:     BITB    #TO11DN!TO11ER,(R1)     ;IS THE TRANSMISSION FINISHED?
004746  000202 ;173746  000202
004750  001775 ;173750  001775          BEQ     4$              ;LOOP UNTIL IT IS FINISHED
004752  100003 ;173752  100003          BPL     5$              ;IF "TO11DN" ISN'T ON, "TO11ER" MUST BE ON
004754  022715 ;173754  022715          CMP     #000240,(R5)    ;CHECK FOR BIT PATTERN IN LOCATION ZERO
004756  000240 ;173756  000240
004760  001403 ;173760  001403          BEQ     9$              ;UNLESS THERE IS A "NOP" IT IS AN ERROR
004762  012711 ;173762  012711  5$:     MOV     #TO10DB,(R1)    ;SIGNAL THE -10 THAT THERE WAS AN ERROR
004764  000400 ;173764  000400
004766  000000 ;173766  000000  12$:    HALT                    ;THIS ERROR HALT IS BECAUSE EITHER "TO11ER"
                                        ; IS ON, OR BECAUSE THE BIT PATTERN READ INTO LOCATION ZERO WASN'T "000240".
004770  012762 ;173770  012762  9$:     MOV     #DRESET,DIAG2(R2)       ;SIGNAL THE -10 THAT EVERYTHING IS OK
004772  000100 ;173772  000100
004774  000032 ;173774  000032
004776  END.YD:
004776  000115 ;173776  000115          JMP     (R5)            ;JUMP TO LOCATION ZERO
                        ;       000001          .END
```

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

```
                        ;BM873F - 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       ;PRIVILEDGED EXAMINE (FUNCTION 15)
;       000016  DF.PDP=16       ;PRIVILEDGED 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
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;
                                ;
                                ;       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  DEXWD3=174402   ;DEPOSIT OR EXAMINE WORD 3
;       174404  DEXWD2=174404   ;DEPOSIT OR EXAMINE WORD 2
;       174406  DEXWD1=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
```

```
2384                    ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
2385                    ;
2386
2387                                    ;
2388                                    ;       DIAG1 DEFINITIONS
2389                    ;               004000  DS04=004000      ;KL CLOCK ERROR STOP
2390                    ;               002000  DS05=002000      ;RUN
2391                    ;               001000  DS06=001000      ;HALT
2392                    ;               000400  DEX=000400       ;DEPOSIT OR EXAMINE MAJOR STATE
2393                    ;               000200  TO10=000200      ;TO 10
2394                    ;               000200  DFUNC=000200
2395                    ;               000100  TO11=000100      ;TO-11 TRANSFER MAJOR STATE
2396                    ;               000040  D1011=000040     ;DIAGNOSE 10/11 INTERFACE
2397                    ;               000020  PULSE=000020     ;SINGLE CLOCK CYCLE
2398                    ;               000010  DIKL10=00010     ;DIAGNOSTIC MODE SWITCH
2399                    ;               000004  DSEND=000004     ;SEND DATA
2400                    ;               000001  DCOMST=000001    ;DIAGNOSTIC COMMAND START
2401                                    ;
2402                                    ;       DIAG1 FUNCTIONS
2403                                    ;
2404                    ;       000000  .STPCL=0                ;STOP THE KL CLOCK
2405                    ;       001000  .STRCL=01*1000          ;START THE KL CLOCK
2406                    ;       002000  .SSCLK=02*1000          ;SINGLE STEP THE M BOX CLOCK
2407                    ;       003000  .SECLK=03*1000          ;SINGLE STEP THE  EBOX CLOCK. LEAVES THE
2408                    ;                       ;EBOX CLOCK FALSE AND EBOX SYNC TRUE.
2409                    ;                       ;CAUSES (2,3) MBOX CLOCKS DEPENDING ON
2410                    ;                       ;EBOX CLOCK INITIALLY (FALSE,TRUE).
2411                    ;                       ;DOES NOT DEPEND ON 'T' FIELD OR MB WAIT.
2412                    ;       004000  .CECLK=04*1000          ;CONDITIONALLY ISSUE AN EBOX CLOCK IF THE EBOX
2413                    ;                       ;CLOCK IS TRUE.  MAKES EBOX CLOCK FALSE.
2414                    ;                       ;IF ISSUED IN THE MASTER RESET STATE.
2415                    ;                       ;LEAVES EBOX SYNC TRUE.
2416                    ;       005000  .BRCLK=05*1000          ;ISSUE A BURST OF THE CLOCKS.  THE NUMBER
2417                    ;                       ;OF MBOX CLOCKS DESIRED (1-255) HAS BEEN
2418                    ;                       ;BEEN LOADED PREVIOUSLY BY FUNCTIONS LDBRR,LDBRL
2419                    ;                       ;(42,43)
2420                    ;       006000  .CLRMR=06*1000          ;CLEAR MASTER RESET STATE
2421                    ;       007000  .SETMR=07*1000          ;SET MASTER RESET STATE. RUNNING THE CLOCK WHILE IN THIS
2422                    ;                       ;STATE 'CLEARS' THE KL10.
2423                    ;       010000  .CLRUN=10*1000          ;CLEAR THE RUN FLOP.  MAKE THE MICRO CODE GO TO
2424                    ;                       ;THE HALT-LOOP.
2425                    ;       011000  .SETRN=11*1000          ;SET THE RUN FLOP.  ALLOW REPEATED INSTRUCTION EXECUTION
2426                    ;       012000  .CONBT=12*1000          ;SET THE CONTINUE FLOP (MOMENTARY). ALLOW THE
2427                    ;                       ;MICRO CODE TO LEAVE THE HALT LOOP
2428                    ;       014000  .IRLTC=14*1000          ;UNLATCH THE IR AND LOAD IT FROM THE AD.
2429                    ;       015000  .DRLTC=15*1000          ;UNLATCH THE DRAM REGISTER AND ALLOW IT TO LOAD FROM THE
2430                    ;                       ;RAMS
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;
                                ;
                                ;       CLOCK LOAD FUNCTIONS
                                ;
                ;       042000  .LDBRR=42*1000  ;LOA D 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.
```

```
                        ;BM873F - 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
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;
                        ;       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  WEP=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  TO10BM=000001   ;TO-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
                        ;
                                ;
                                ;       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  RAMIS0=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  INTROF=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 OFFSETTS (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  TOD=3           ;TIME OF DAY
;       000004  DATE=4          ;DATE
;       000005  PSWW1=5         ;PROCESSOR STATUS WORD1
;       000006  PSWW2=6         ;PROCESSOR STATUS WORD2
;       000007  PSWW3=7         ;PROCESSOR STATUS WORD3
;       000010  PSWW4=10        ;PROCESSOR STATUS WORD4
;       000011  PSWW5=11        ;PROCESSOR STATUS WORD5
;       000012  PSWW6=12        ;PROCESSOR STATUS WORD6
;       000013  PSWW7=13        ;PROCESSOR STATUS WORD7
;       000014  PSWW10=14       ;PROCESSOR STATUS WORD10
;       000015  PSWW11=15       ;PROCESSOR STATUS WORD11
;       000016  PSWW12=16       ;PROCESSOR STATUS WORD12
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;
                        ;       000017  PSWW13=17       ;PROCESSOR STATUS WORD13
                        ;       000020  FORPRO=20       ;FOR PROCESSOR IDENTIFICATON WORD
                        ;       000021  PROPNT=21       ;POINTER TO COMM AREA OF THE PROCESSOR ASSOC WITH THIS BLOCK
                        ;       000022  STATUS=22       ;COMMUNICATION STATUS WORD
                        ;       000023  QSIZE=23        ;QUEUE SIZE WORD
                                        ;CTY0CW=24      ;CTY #0 COMMAND WORD
                                        ;CTY0RW=25      ;CTY #0 RESPONSE WORD
                                        ;CTY1CW=26      ;CTY #1 COMMAND WORD
                                        ;CTY1RW=27      ;CTY #1 RESPONSE WORD
                                        ;MISCW=30       ;MISCELLANEOUS COMMAND WORD FOR NON-QUEUE PROTOCOL
                                        ;MISRW=31       ;MISCELLANEOUS RESPONSE WORD
                        ;       000032  UNASG1=32       ;UNASSIGNED WORD1
                        ;       000033  UNASG2=33       ;UNASSIGNED WORD2
                        ;       000034  UNASG3=34       ;UNASSIGNED WORD3
                        ;       000035  UNASG4=35       ;UNASSIGNED WORD4
                        ;       000036  UNASG5=36       ;UNASSIGNED WORD5
                        ;       000037  UNASG6=37       ;UNASSIGNED WORD6
                                        ;
                                        ;       EPT ADDRESSES AS DEFINED IN BOOTS FOR USE IN THE
                                        ;       SECONDARY PROTOCOL
                                        ;
                        ;       000444  DTEFLG=444      ;OPERATION COMPLETE FLAG
                        ;       000450  DTEF11=450      ;PDP-10 FROM PDP-11 ARGUMENT
                        ;       000451  DTECMD=451      ;PDP-10 TO PDP-11 COMMAND WORD
                        ;       000455  DTEMTO=455      ;MONITOR TTY OUTPUT COMPLETE FLAG
                        ;       000456  DTEMTI=456      ;MONITOR TTY INPUT FLAG
                                        ;
                                        ;
                                        ;       STATUS DEFINITONS
                                        ;
                        ;       000001  TOIT=1          ;IN PROGRESS OF PROCESSING QUEUE
                        ;       000002  TOIP=2          ;TO HIM INDIRECT IN PROGRESS
```

```
2664                    ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 2
2665                    ;
2666                                    ;
2667                                    ;
2668                                    ; DEFINITIONS . . .
2669                                    ;
2670                    ;       000040  ROTOR7= 40      ;SAVE R0 TO R7 IN 40 TO 56
2671                                    ;
2672                    ;       000130  DTESAV= 130     ;SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
2673                    ;                               ; IN LOCATIONS 130-156
2674                                    ;
2675                    ;       000012  RETRY=  10.     ;DO 10 RETRIES BEFORE HALTING
2676                                    ;
2677                    ;       173000  ROMORG= 173000  ;ROM STARTS AT 773000
2678                                    ;
2679                                    ; ESTABLISH ROM ORIGIN
2680                                    ;
2681                    ;       173000  .=ROMORG
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 3
                        ;
                                        ;       EXTERNAL BUTTONS #1, #2, #3
                                        ;
                                        ; BUTTON #1 -- LOAD USING SWITCH REGISTER
                                        ;
005000  010037  ;173000 010037  BUTON1: MOV     R0,ROTOR7+0     ;SAVE R0 IN LOCATION 40
005002  000040  ;173002 000040
005004  013700  ;173004 013700          MOV     SWR,R0          ;GET SWITCH REGISTER
005006  177570  ;173006 177570
005010  032700  ;173010 032700          BIT     #BIT0,R0        ;IS LOW-ORDER BIT SET?
005012  000001  ;173012 000001
005014  001007  ;173014 001007          BNE     BUTONX          ;YES-- LOOK AT CONTENTS
005016  000557  ;173016 000557          BR      REGSAV          ;NO-- SAVE R1-R7 IN 42-56, GO TO ADDRESS IN R0 (FROM SWR
                                        ;
                                        ; BUTTON #3 -- LOAD BOOT FROM RX11 FLOPPY DISK
                                        ;
005020  005000  ;173020 005000  BUTON3: CLR     R0              ;SAY LOAD FROM FLOPPY, UNIT 0
005022  000404  ;173022 000404          BR      BUTONX          ;GO TO COMMON CODE FOR 3 BUTTONS
                                        ;
                                        ; REQUIRED POWER-FAIL VECTOR
                                        ;
005024  173000  ;173024 173000          .WORD   ROMORG,PR7
005026  000340  ;173026 000340
                                        ;
                                        ; BUTTON #2 -- LOAD BOOT FROM RP04 DISK
                                        ;
005030  012700  ;173030 012700  BUTON2: MOV     #BIT7,R0        ;BIT 7 MEANS LOAD FROM RP04
005032  000200  ;173032 000200          BR      BUTONX          ;FALL INTO COMMON CODE
                                        ;
                                        ; R0 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
                                        ;       BIT 7 = 1       LOAD FROM RP04 DISK
                                        ;       BIT 15 = 1      INDEFINITE RETRY
                                        ;
                                        ; NOTE THAT IF BUTTON #4 IS PRESSED, R5 WILL CONTAIN BIT 0 = 0, BIT 15 = 1
                                        ;
005034  010005  ;173034 010005  BUTONX: MOV     R0,R5           ;SAVE PARAMETER FOR BOOT
005036  106300  ;173036 106300          ASLB    R0              ;LEFT-ALIGN SPEED FIELD IN LOW BYTE
005040  122700  ;173040 122700          CMPB    #16*BIT4,R0     ;IS SPEED 16 OR 17?
005042  000340  ;173042 000340
005044  101404  ;173044 101404          BLOS    10$             ;YES-- UNIT FIELD IS UNIT # TO BOOT FROM
005046  122700  ;173046 122700          CMPB    #3*BIT4,R0      ;IS SPEED 0, 1, OR 2?
005050  000060  ;173050 000060
005052  101001  ;173052 101001          BHI     10$             ;YES-- UNIT IS UNIT TO USE
005054  005000  ;173054 005000          CLR     R0              ;NO-- USE UNIT #0
                                        ;
005056  000300  ;173056 000300  10$:    SWAB    R0              ;GET UNIT # IN LOW BYTE
005060  042700  ;173060 042700          BIC     #^C7,R0         ;TRIM TO 3 BITS 2, 1, 0
005062  177770  ;173062 177770
```

```
2736                                    ;
2737                                    ; UNIT # IS IN R0-- CALL PROPER BOOT DEPENDING ON BIT 7
2738                                    ;
2739  005064  105705  ;173064 105705            TSTB    R5              ;WHERE SHOULD WE BOOT FROM?
2740  005066  100550  ;173066 100550            BMI     RPBOOT          ;BIT 7 = 1 -- BOOT FROM RP04 DISK
2741                                    ;       BR      RXBOOT          ;BIT 7 = 0 -- BOOT FROM RX11 FLOPPY DISK
```

```
                        ;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= BIT5            ;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
                                ;
```

```
2788                                    ;
2789                                    ; HERE TO BOOT FROM RX11 FLOPPY DISK-- UNIT # IN R0
2790                                    ;
2791                    ;173070         RXBOOT:
2792  005070  012703    ;173070 012703          MOV     #<RXBTRK*BIT8>!<RXBSCT*BIT0>,R3
2793  005072  000401    ;173072 000401
2794  005074  005700    ;173074 005700          TST     R0              ;IS THIS UNIT # 0?
2795  005076  001402    ;173076 001402          BEQ     10$             ;YES-- USE 0
2796  005100  012700    ;173100 012700          MOV     #RXUNIT,R0      ;NO-- USE UNIT # 1
2797  005102  000020    ;173102 000020
2798  005104  052700    ;173104 052700  10$:    BIS     #RXREAD+RXGO,R0 ;SET READ FUNCTION IN R0
2799  005106  000007    ;173106 000007
2800                                    ;       BR      RXSTRT          ;FALL INTO START-UP
```

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

```
                                        ;
                                        ; DISK TRANSFER COMPLETE WITH NO ERRORS
                                        ;
005206  005700  ;173206 005700                TST     R0              ;THIS A WRITE?
005210  100421  ;173210 100421                BMI     RXWDON          ;YES-- SEE IF DONE WITH DUMP
                                        ;       BR      RXEMSL          ;NO-- READ-- EMPTY SILO
                                        ;
                                        ; READ COMPLETED-- EMPTY SILO TO MEMORY
                                        ;
005212  012711  ;173212 012711  RXEMSL: MOV     #RXEMPT+RXGO,(R1) ;START EMPTY
005214  000003  ;173214 000003
                                        ;
005216  132711  ;173216 132711  10$:    BITB    #RXTREQ!RXDONE,(R1) ;READY FOR WORD, OR TRANSFER DONE?
005220  000240  ;173220 000240
005222  000402  ;173222 000402                BR      20$             ;BRANCH AROUND VECTOR
                                        ;
                                        ; REQUIRED POWER-FAIL VECTOR
                                        ;
005224  173000  ;173224 173000                .WORD   ROMORG,PR7
005226  000340  ;173226 000340
                                        ;
005230  001772  ;173230 001772  20$:    BEQ     10$             ;NOT READY-- WAIT SOME MORE
005232  100003  ;173232 100003            BPL     RXRDON          ;DONE-- GET ANOTHER SECTOR
005234  116124  ;173234 116124            MOVB    RXDB(R1),(R4)+  ;NOT DONE-- GET A BYTE FROM SILO TO MEMORY
005236  000002  ;173236 000002
005240  000766  ;173240 000766            BR      10$             ;WAIT FOR NEXT BYTE
                                        ;
                                        ; SILO EMPTIED-- SEE IF WE ARE DONE WITH BOOTING
                                        ;
                ;173242                 RXRDON:
                ;       173242          $$$=.
005242  122222  ;173242 122222            CMPB    (R2)+,(R2)+
005244  022704  ;173244 022704            CMP     #256.*2,R4      ;HAVE WE READ ENOUGH?
005246  001000  ;173246 001000
005250  101337  ;173250 101337            BHI     RXPERF          ;NO-- READ SOME MORE
005252  005007  ;173252 005007  CLRPC:   CLR     PC              ;YES-- GO TO LOCATION ZERO
                                        ;
                                        ; WRITE COMPLETED-- SEE IF DONE DUMPING
                                        ;
                ;173254                 RXWDON:
                ;       173254          $$$=.
005254  005202  ;173254 005202            INC     R2
005256  122702  ;173256 122702            CMPB    #26.,R2         ;THIS LAST SECTOR ON TRACK?
005260  000032  ;173260 000032
005262  103003  ;173262 103003            BHIS    10$             ;NO-- KEEP ON GOING
005264  105002  ;173264 105002            CLRB    R2              ;YES-- CLEAR SECTOR ADDRESS
005266  062702  ;173266 062702            ADD     #BIT8!BIT0,R2   ;BUMP TO NEXT TRACK, SECTOR 1
005270  000401  ;173270 000401
005272  022704  ;173272 022704  10$:    CMP     #1024.*28.*2,R4 ;ARE WE DONE WITH 28 K?
005274  160000  ;173274 160000
005276  101516  ;173276 101516            BLOS    HALT0           ;YES-- GO HALT WITH R0= 0 IN DISPLAY
                                        ;       BR      RXFLSL          ;NO-- FILL SILO WITH NEXT SECTOR
```

```
                                        ;
                                        ; WRITE ANOTHER BLOCK-- FILL SILO
                                        ;
005300  012711  ;173300 012711  RXFLSL: MOV     #RXFILL+RXGO,(R1) ;SET TO FILL SILO
005302  000001  ;173302 000001
                                        ;
005304  132711  ;173304 132711  10$:    BITB    #RXTREQ!RXDONE,(R1) ;READY FOR ANOTHER BYTE?
005306  000240  ;173306 000240
005310  001775  ;173310 001775          BEQ     10$             ;NO-- WAIT SOME MORE
005312  100316  ;173312 100316          BPL     RXPERF          ;DONE-- GO PERFORM WRITE
005314  112461  ;173314 112461          MOVB    (R4)+,RXDB(R1)  ;YES-- STORE ANOTHER BYTE IN SILO
005316  000002  ;173316 000002
005320  000771  ;173320 000771          BR      10$             ;WAIT UNTIL READY FOR ANOTHER
                                        ;
                                        ; HERE ON ERROR AFTER RETRYING -- DISPLAY ERROR REGISTER AND HALT
                                        ;
005322  012711  ;173322 012711  RXEHLT: MOV     #RXRERR+RXGO,(R1) ;DO A READ ERROR REGISTER FUNCTION
005324  000017  ;173324 000017
005326  032711  ;173326 032711  10$:    BIT     #RXDONE,(R1)    ;WAIT UNTIL ERROR ASSEMBLED
005330  000040  ;173330 000040
005332  001775  ;173332 001775          BEQ     10$             ;
005334  016100  ;173334 016100          MOV     RXDB(R1),R0     ;GET ERROR REGISTER
005336  000002  ;173336 000002
005340  000476  ;173340 000476          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
                                        ;
                ;173342         RXDUMP:
005342  012703  ;173342 012703          MOV     #<RXDTRK*BIT8>!<RXDSCT*BIT0>,R3
005344  035401  ;173344 035401
                                        ;
005346  012700  ;173346 012700          MOV     #BIT15!RXWRIT+RXGO,R0 ;DO A WRITE
005350  100005  ;173350 100005
005352  005005  ;173352 005005          CLR     R5              ;CLEAR INDEFINITE RETRY BIT
005354  000655  ;173354 000655          BR      RXSTRT          ;START DUMP GOING
```

```
2937                    ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 5
2938                    ;
2939                                    ;       REGISTER SAVE ROUTINE
2940                                    ;
2941                                    ; REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS R0-R7
2942                                    ; IN MEMORY AT 40-56 (LOCATION R0TOR7).
2943                                    ;
2944                                    ; CALLING SEQUENCE:
2945                                    ;       MOV     R0,R0TOR7+0
2946                                    ;       MOV     #RET,R0
2947                                    ;       BR      REGSAV
2948                                    ; RET:  <RETURN HERE>
2949                                    ;
2950                                    ; ALL REGISTERS RESTORED
2951                                    ;
2952                                    ;
2953  005356  010037  ;173356 010037   REGSAV: MOV     R0,R0TOR7+16    ;SAVE R0 AS PC IN 56
2954  005360  000056  ;173360 000056
2955  005362  012700  ;173362 012700           MOV     #R0TOR7+16,R0   ;R0 NOW POINTS TO 56
2956  005364  000056  ;173364 000056
2957  005366  010640  ;173366 010640           MOV     SP,-(R0)        ;SAVE SP IN 54
2958  005370  010540  ;173370 010540           MOV     R5,-(R0)        ;SAVE R5 IN 52
2959  005372  010440  ;173372 010440           MOV     R4,-(R0)        ;SAVE R4 IN 50
2960  005374  010340  ;173374 010340           MOV     R3,-(R0)        ;SAVE R3 IN 46
2961  005376  010240  ;173376 010240           MOV     R2,-(R0)        ;SAVE R2 IN 44
2962  005400  010140  ;173400 010140           MOV     R1,-(R0)        ;SAVE R1 IN 42
2963  005402  014000  ;173402 014000           MOV     -(R0),R0        ;RESTORE R0 FROM 40
2964  005404  000177  ;173404 000177           JMP     @R0TOR7+16      ;GO TO SAVED PC
2965  005406  004446  ;173406 004446
```

```
2966                    ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 6
2967                    ;
2968                            ;       RP04 DISK BOOTSTRAP AND DUMP ROUTINES
2969                            ;
2970                            ; RP04 REGISTER DEFINITIONS
2971                            ;
2972                    ;       176700  RPEPA=  176700  ;EXTERNAL PAGE ADDRESS OF RP04 REGISTERS
2973                            ;
2974                    ;       000000  RPCS1=  0       ;OFFSET FOR CSR #1
2975                    ;       040000          RPTRE=  BIT14           ;TRANFER ERRROR
2976                    ;       020000          RPMCPE= BIT13           ;MASSBUS CONTROL PARITY ERROR
2977                    ;       004000          RPDVA=  BIT11           ;DRIVE AVAILABLE (TO -11)
2978                    ;       000200          RPRDY=  BIT7            ;FUNCTION COMPLETE
2979                    ;       000076          RPFUNC= BIT5!BIT4!BIT3!BIT2!BIT1 ;FUNCTION:
2980                    ;       000020                  RPPRST= 20      ; READ-IN PRESET
2981                    ;       000060                  RPWRIT= 60      ; WRITE DATA
2982                    ;       000070                  RPREAD= 70      ; READ DATA
2983                    ;       000001          RPGO=   BIT0            ;GO
2984                    ;       000002  RPWC=   2       ;WORD COUNT REGISTER
2985                    ;       000006  RPDA=   6       ;TRACK (HIGH BYTE) SECTOR (LOW BYTE)
2986                    ;       000010  RPCS2=  10      ;CONTROL AND STATUS REGISTER #2
2987                    ;       000007          RPUNIT= BIT2!BIT1!BIT0  ;UNIT #
2988                    ;       000012  RPDS=   12      ;DRIVE STATUS REGISTER
2989                    ;       100000          RPATA=  BIT15           ;ATTENTION ACTIVE
2990                    ;       040000          RPERR=  BIT14           ;DRIVE ERROR
2991                    ;       000034  RPDC=   34      ;DESIRED CYLINDER
2992                            ;
2993                            ;
2994                            ; PARAMETERS
2995                            ;
2996                    ;       000000  RPBCYL= 0.      ;BOOT FROM CYLINDER 0
2997                    ;       000000  RPBTRK= 0.      ; TRACK 0
2998                    ;       000000  RPBSCT= 0.      ; SECTOR 0
2999                            ;
3000                    ;       000631  RPDCYL= 409.    ;DUMP TO CYLINDER 409
3001                    ;       000015  RPDTRK= 13.     ; TRACK 13
3002                    ;       000010  RPDSCT= 8.      ; SECTOR 8
3003                            ;
3004                            ;
3005                            ; REGISTER USAGE:
3006                            ;       R0 --   FUNCTION CODE (HIGH BYTE) UNIT # (LOW BYTE)
3007                            ;               BIT 15 SET IF WRITE
3008                            ;       R1 --   ADDRESS OF RPCS1
3009                            ;       R2 --   CYLINDER #
3010                            ;       R3 --   TRACK (HIGH BYTE) SECTOR (LOW BYTE)
3011                            ;       R4 --   WORD COUNT
3012                            ;       R5 --   PARAMETER WORD SAVED FROM INITIALIZATION
3013                            ;       SP --   RETRY COUNTER
3014                            ;
```

```
3015                                    ;
3016                                    ; HERE TO BOOT FROM RP04-- UNIT # IN R0
3017                                    ;
3018                    ;173410         RPBOOT:
3019  005410  005002    ;173410 005002          CLR     R2
3020  005412  005003    ;173412 005003          CLR     R3
3021  005414  052700    ;173414 052700          BIS     #<RPREAD+RPGO>*BIT8,R0 ;SET READ HIGH BYTE, UNIT # LOW BYTE
3022  005416  034400    ;173416 034400
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 6
                        ;
005420  012704  ;173420 012704          MOV     #-256.,R4       ;READ 256 WORDS TO BOOT
005422  177400  ;173422 177400
                ;       BR      RPSTRT          ;START BOOT GOING
                ;
                ; START RPO4 GOING ON EITHER DUMP OR BOOT
                ;
005424  012706  ;173424 012706  RPSTRT: MOV     #RETRY,SP       ;RETRY RETRY TIMES
005426  000012  ;173426 000012
005430  012701  ;173430 012701          MOV     #RPEPA+RPCS1,R1 ;ADDRESS RPCS1 IN R1
005432  176700  ;173432 176700
                ;       BR      RPRTRY          ;FALL THROUGH RETRY CODE
                ;
                ; HERE ON ERROR TO RETRY
                ;
005434  005705  ;173434 005705  RPRTRY: TST     R5              ;INFINITE RETRY?
005436  100402  ;173436 100402          BMI     10$             ;YES-- TRY AGAIN
005440  005306  ;173440 005306          DEC     SP              ;RETRY COUNT EXHAUSTED?
005442  002437  ;173442 002437          BLT     RPEHLT          ;YES-- GIVE UP
                ;
005444  000005  ;173444 000005  10$:    RESET   ;ZAP!!
005446  110061  ;173446 110061          MOVB    R0,RPCS2(R1)    ;SELECT PROPER UNIT #
005450  000010  ;173450 000010
005452  032711  ;173452 032711          BIT     #RPDVA,(R1)     ;IS DRIVE AVAILABLE TO US?
005454  004000  ;173454 004000
005456  001766  ;173456 001766          BEQ     RPRTRY          ;NO-- TRY AGAIN
005460  012711  ;173460 012711          MOV     #RPPRST+RPGO,(R1) ;DO 'READ-IN PRESET' FUNCTION
005462  000021  ;173460 000021
005464  010261  ;173464 010261          MOV     R2,RPDC(R1)     ;SELECT PROPER CYLINDER
005466  000034  ;173466 000034
005470  010361  ;173470 010361          MOV     R3,RPDA(R1)     ; AND TRACK AND SECTOR
005472  000006  ;173472 000006
005474  010461  ;173474 010461          MOV     R4,RPWC(R1)     ;SET UP WORD COUNT TO PROPER VALUE
005476  000002  ;173476 000002
                ;                       ;NOTE THAT IT IS NOT NECCESARY TO SET UP BUS
                ;                       ; ADDRESS, SINCE IT IS 0 AFTER READ-IN PRESET
005500  000300  ;173500 000300          SWAB    R0              ;GET FUNCTION CODE IN LOW BYTE
005502  110011  ;173502 110011          MOVB    R0,(R1)         ;START FUNCTION GOING
005504  000300  ;173504 000300          SWAB    R0              ;RESTORE R0
                ;
005506  105711  ;173506 105711  20$:    TSTB    (R1)            ;READY?
005510  100376  ;173510 100376          BPL     20$             ;NO-- WAIT UNTIL IT IS
005512  032711  ;173512 032711          BIT     #RPTRE!RPMCPE,(R1) ;TRANSFER OR MBC PARITY ERROR?
005514  060000  ;173514 060000
005516  001346  ;173516 001346          BNE     RPRTRY          ;YES-- ERROR-- TRY AGAIN
005520  032761  ;173520 032761          BIT     #RPATA!RPERR,RPDS(R1) ;ATTN OR OTHER ERROR?
005522  140000  ;173522 140000
005524  000012  ;173524 000012
005526  001342  ;173526 001342          BNE     RPRTRY          ;YES-- ERROR-- TRY AGAIN
005530  005700  ;173530 005700          TST     R0              ;READ FUNCTION?
005532  100247  ;173532 100247          BPL     CLRPC           ;YES-- BOOT-- GO TO LOCATION 0
                ;       BR      HALT0           ;NO-- DUMP-- HALT WITH R0= 0 IN DISPLAY
```

```
3076                                    ;
3077                                    ; HERE TO HALT AFTER A DUMP-- DISPLAY R0= 0 IF NO ERRORS
3078                                    ;
3079  005534  005000  ;173534 005000  HALTO:  CLR     R0              ;DISPLAY R0= 0 TO INDICATE NO ERRORS
3080                                    ;
3081  005536  000000  ;173536 000000  HALTED: HALT    ;DIE
3082  005540  000776  ;173540 000776          BR      HALTED          ;STAY DEAD
3083                                    ;
3084                                    ; HERE ON ERROR FROM RP04 AFTER RETRYING-- DISPLAY DRIVE STATUS IN R0
3085                                    ;
3086  005542  016100  ;173542 016100  RPEHLT: MOV     RPDS(R1),R0     ;DISPLAY DRIVE STATUS
3087  005544  000012  ;173544 000012
3088  005546  000773  ;173546 000773          BR      HALTED          ;R.I.P.
3089                                    ;
3090                                    ;
3091                                    ; START -11 HERE TO DUMP TO RP04 DISK
3092                                    ;
3093                                    ; NOTE THAT R0-R7 HAVE ALREADY BEEN SAVED IN 40-56
3094                                    ;  BY PRESSING BUTTON #1.
3095                                    ;
3096                    ;173550         RPDUMP:
3097  005550  012702  ;173550 012702          MOV     #RPDCYL,R2
3098  005552  000631  ;173552 000631
3099  005554  012703  ;173554 012703          MOV     #<RPDTRK*BIT8>!<RPDSCT*BIT0>,R3
3100  005556  006410  ;173556 006410
3101  005560  012700  ;173560 012700          MOV     #BIT15!<<RPWRIT+RPGO>*BIT8>,R0 ;DO A WRITE, UNIT # 0
3102  005562  130400  ;173562 130400
3103  005564  012704  ;173564 012704          MOV     #-<1024.*28.>,R4 ;SET TO DUMP 28 K
3104  005566  110000  ;173566 110000
3105  005570  005005  ;173570 005005          CLR     R5              ;CLEAR INDEFINITE RETRY BIT
3106  005572  000714  ;173572 000714          BR      RPSTRT          ;START DUMP GOING
```

```
3107                    ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 7
3108                    ;
3109                                    ;       INTERNAL BUTTON #4 -- DUMP AND BOOTSTRAP THROUGH DTE20
3110                                    ;
3111                                    ; DTE20 DEFINITIONS
3112                                    ;
3113                                    ; NOTE THAT ALL DTE20 REGISTER DEFINITIONS AND BIT DEFINITIONS
3114                                    ;  ARE IN $DEF IN SYSMAC.SML
3115                                    ;
3116                    ;       000040  DTESIZ= 40      ;EACH DTE OCCUPIES 20 WORDS IN EXTERNAL PAGE
3117                    ;       000004  DTEMAX= 4       ;MAX OF 4 DTE'S ON A PDP-11
3118                                    ;
3119                                    ;
3120                                    ; BUTTON #4 -- INITIATED BY '-10 RELOAD -11' BIT
3121                                    ;
3122  005574  010037  ;173574 010037  BUTON4: MOV     R0,ROTOR7+0     ;SAVE R0 IN 40
3123  005576  000040  ;173576 000040
3124  005600  012700  ;173600 012700          MOV     #10$,R0         ;SET RETURN ADDRESS IN R0
3125  005602  173606  ;173602 173606
3126  005604  000664  ;173604 000664          BR      REGSAV          ;SAVE R1-R7
3127                                    ;
3128                                    ; REGISTERS SAVED-- LOOK FOR THE DTE20 WHICH PUSHED THE BUTTON
3129                                    ;
3130                                    ; THE DTE WHICH PUSHED THE BUTTON SHOULD HAVE THE DOORBELL
3131                                    ;  RINGING AND HAVE THE VALUE 1365 (OCTAL) IN IT'S
3132                                    ;  TO -10 BYTE COUNT TO10BC.
3133                                    ;
3134                                    ; NXM (TIME-OUT) TRAP IS USED TO SKIP NON-EXISTANT DTE20'S.
3135                                    ;
3136  005606  005005  ;173606 005005  10$:    CLR     R5              ;ADDRESS LOCATION ZERO
3137  005610  012500  ;173610 012500          MOV     (R5)+,R0        ;SAVE 0 IN R0
3138  005612  012501  ;173612 012501          MOV     (R5)+,R1        ;SAVE 2 IN R1
3139  005614  011502  ;173614 011502          MOV     (R5),R2         ;SAVE 4 IN R2
3140  005616  012725  ;173616 012725          MOV     #21$,(R5)+      ;SET NXM TRAP ADDRESS IN 4
3141  005620  173634  ;173620 173634
3142  005622  011503  ;173622 011503          MOV     (R5),R3         ;SAVE 6 IN R3
3143  005624  012715  ;173624 012715          MOV     #PR7,(R5)       ;SET PRIORITY FOR NXM TRAP
3144  005626  000340  ;173626 000340
3145                                    ;
3146                                    ; LOOP THROUGH ALL DTE'S
3147                                    ;
3148  005630  012704  ;173630 012704  20$:    MOV     #DLYCNT-DTESIZ,R4 ;POINT TO DTE # -1'S DELAY COUNT REGISTER
3149  005632  174340  ;173632 174340
3150                                                    ; (WILL BUMP TO # 0)
```

```
3151                                    ;
3152                                    ; HERE ON NXM TRAP-- RESET SP AND TRY NEXT DTE
3153                                    ;
3154  005634  012706  ;173634 012706  21$:    MOV     #4,SP           ;SET SP TO 4, STACK IS LOCATIONS 2 AND 0
3155  005636  000004  ;173636 000004
3156                                    ;
3157  005640  062704  ;173640 062704  22$:    ADD     #DTESIZ,R4      ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS
3158  005642  000040  ;173642 000040
3159  005644  105704  ;173644 105704          TSTB    R4              ;IS THIS THE END OF THE DTE'S?
3160                  ;                               ; NOTE THAT THE LAST DTE IS AT 774540
3161                  ;                               ; AND THAT NOW R4= 774600 IF END
3162  005646  100770  ;173646 100770          BMI     20$             ;YES-- START ALL OVER, UNTIL A DTE
3163                  ;                               ; SAYS HE PUSHED THE BUTTON
3164  005650  032764  ;173650 032764          BIT     #TO11DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?
3165  005652  004000  ;173652 004000
3166  005654  000034  ;173654 000034
3167  005656  001770  ;173656 001770          BEQ     22$             ;NO-- TRY NEXT DTE
3168  005660  026417  ;173660 026417          CMP     TO10BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365
3169  005662  000014  ;173662 000014
3170                  ;                               ; IN IT'S TO -10 BYTE COUNT?
3171  005664  001365  ;173664 001365          BNE     22$             ;NO-- TRY ANOTHER DTE
3172                                    ;
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 7
                        ;
                                        ; WE HAVE FOUND THE DTE WHICH PUSHED THE BUTTON
                                        ;
                                        ; ADDRESS OF DLYCNT REGISTER IS IN R4
                                        ;
005666  010315  ;173666 010315                MOV     R3,(R5)         ;RESTORE LOCATION 6
005670  010245  ;173670 010245                MOV     R2,-(R5)        ; 4
005672  010145  ;173672 010145                MOV     R1,-(R5)        ; 2
005674  010045  ;173674 010045                MOV     R0,-(R5)        ; 0
                                        ;
                                        ; SAVE FIRST 12 DTE REGISTERS DLYCNT TO TO11DT
                                        ;  IN LOCATIONS 130-156
                                        ;
005676  012700  ;173676 012700                MOV     #DTESAV,R0      ;POINT TO SAVE AREA
005700  000130  ;173700 000130
005702  012420  ;173702 012420        29$:    MOV     (R4)+,(R0)+     ;SAVE A REGISTER
005704  022700  ;173704 022700                CMP     #TO11DT-DLYCNT+DTESAV,R0 ;FINISHED?
005706  000156  ;173706 000156
005710  103374  ;173710 103374                BHIS    29$             ;NO-- SAVE SOME MORE
                                        ;
                                        ; R4= TO11DT+2
                                        ;
                                        ; SET R1= STATUS REGISTER
                                        ;     R4= DIAG2 REGISTER
                                        ;
                                        ; DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
                                        ;  LOADED FLAG
                                        ;
                        ;173712        $$$=.
005712  005724  ;173712 005724                TST     (R4)+
005714  010401  ;173714 010401                MOV     R4,R1           ; SO DOES R1
005716  012700  ;173716 012700                MOV     #DRESET,R0      ;SETUP R0 FOR 'DIAGNOSTIC RESET'
005720  000100  ;173720 000100
005722  010021  ;173722 010021                MOV     R0,(R1)+        ;R1 POINTS TO STATUS REGISTER
```

```
                                        ;
                                        ; 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.
                                        ;
005724  005061  ;173724 005061                CLR     DLYCNT-STAT(R1) ;SET DTE20 FOR MAXIMUM DELAY (ZERO)
005726  177744  ;173726 177744
005730  005061  ;173730 005061                CLR     TO10AD-STAT(R1) ;START DUMPING -11 MEMORY TO -10
005732  177764  ;173732 177764
                ;                               ; STARTING AT LOCATION 0
005734  032711  ;173734 032711  30$:    BIT     #TO11DB,(R1)    ;IS DOORBELL RINGING (TRANSFER COMPLETE)?
005736  004000  ;173736 004000
005740  001775  ;173740 001775                BEQ     30$             ;NO-- WAIT FOR DOORBELL
005742  010014  ;173742 010014                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.
                                        ;
005744  005061  ;173744 005061                CLR     TO11AD-STAT(R1) ;START INPUT TO LOCATION 0
005746  177766  ;173746 177766
005750  012761  ;173750 012761                MOV     #IFLOP!<<-256.>&7777>,TO11BC-STAT(R1) ;256 WORDS, INTERRUPT
005752  107400  ;173752 107400
005754  177762  ;173754 177762
                                                        ; -10 WHEN DONE
005756  032711  ;173756 032711  40$:    BIT     #TO11DB,(R1)    ;DOORBELL RINGING (LOAD FINISHED)?
005760  004000  ;173760 004000
005762  001775  ;173762 001775                BEQ     40$             ;NO-- WAIT UNTIL DONE
005764  010014  ;173764 010014                MOV     R0,(R4)         ;CLEAR DOORBELL RINGING
005766  012705  ;173766 012705                MOV     #BIT15,R5       ;SET R5: BIT15= 1, BIT0= 0
005770  100000  ;173770 100000
                ;                               ; TO SAY BUTTON #4 PRESSED
005772  005007  ;173772 005007                CLR     PC              ;GO TO LOADED CODE, STARTING AT
                                                                        ; LOCATION 0
```

```
3247                    ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 8
3248                    ;
3249                                    ;       FILL TO END OF ROM
3250                                    ;
3251                                    ;
3252                    ;173774 000004          .PRINT  <1000>-<.-ROMORG> ;FREE BYTES AT 1000
3253  005774  000000    ;173774    000          .BYTE   0
3254                    ;173775    000          .BYTE   0
3255  005776  END.YF:
3256  005776  000000    ;173776    000          .BYTE   0
3257                    ;173777    000          .BYTE   0
3258                                    ;
3259                                    ;
3260                                    ;
3261                    ;174000 000001  PASS2:  .END
3262                    ;
3263                    ;
```

```
3264  006000  MAP.YG:
3265          ;THE FOLLOWING IS A REPRODUCTION
3266          ;OF THE ROM PROGRAM FOR BM873YG.
3267          ;IT IS HERE FOR COMPARISON TO THE
3268          ;ACTUAL ROM AND FOR REFERENCE
3269          ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 1
3270
3271          ;                                   .TITLE  BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM
3272          ;
3273                                      ; THIS IS THE CODE TO BE ENCODED IN THE BOOTSTRAP ROM ON THE BM873-YG BOARD
3274                                      ;
3275                                      ;
3276                                      ;       MODULE:         BM873G
3277                                      ;
3278                                      ;       DATE:           JUNE 1976
3279                                      ;
3280                                      ;       AUTHOR:         RICH MURATORI
3281                                      ;
3282                                      ;
3283                                      ;       COPYRIGHT (C) 1976 DIGITAL EQUIPMENT CORPORATION
3284                                      ;       ALL RIGHTS RESERVED
3285                                      ;
3286                                      ;
3287          ;                           ;       .ENABLE ABS,AMA
3288                                      ;
3289                                      ;       ASCII CHARACTER DEFINTIONS
3290                                      ;
3291          ;           000040  SPACE=  40                  ;ASCII SPACE
3292          ;           000001  SYN=    1                   ;ASCII SYNC
3293          ;           000012  LF=     12                  ;ASCII LINE FEED
3294          ;           000015  CR=     15                  ;ASCII CARRIAGE RETURN
3295          ;           000054  COMMA=  54                  ;ASCII COMMA
3296          ;           000006  ACK=    6                   ;ASCII ACKNOWLEDGE
3297          ;           000025  NAK=    25                  ;ASCII NEG ACKNOWLEDGE
3298                                      ;
3299                                      ;       BUFFER AREAS
3300                                      ;
3301          ;           002100  LINBUF= 2100                ;LINE INPUT BUFFER
3302          ;           002310  DEABUF= 2310                ;DEASCIIZED INPUT BUFFER
3303                                      ;
3304                                      ;       DL11E REGISTER DEFINITIONS
3305                                      ;
3306          ;           176000  DLRCSR= 176000              ;DL11E RECEIVER STATUS REGISTER
3307          ;           176002  DLRBUF= 176002              ;DL11E RECEIVER BUFFER
3308          ;           176004  DLXCSR= 176004              ;DL11E TRANSMITTER STATUS REGISTER
3309          ;           176006  DLXBUF= 176006              ;DL11E TRANSMISSION BUFFER
3310                                      ;
3311          ;           100000  BIT15=100000
3312          ;           000340  PR7=7*40                    ;PRIORITY LEVEL 7
3313                                      ;
3314                                      ;       DTE20 REGISTER DEFINITIONS
3315                                      ;
3316          ;           174400  DLYCNT=174400   ;DELAY COUNT WORD
3317          ;           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
                        174426  TO11DT=174426   ;TO-11 PDP-11 DATA WORD
                        174434  STAT=174434     ;STATUS WORD
                                ;       TO11BC REGISTER BIT DEFINITIONS
                                ;
```

```
3325          ;BM873G - K.10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 1-1
3326
3327          ;               100000  IFLOP=100000    ;I FLIPFLOP BIT
3328                                  ;
3329                                  ;
3330                                  ;       DIAG2 DEFINITIONS
3331                                  ;
3332          ;               000100  DRESET=000100   ;DTE RESET
3333                                  ;
3334                                  ;       STAT REGISTER DEFINITIONS
3335                                  ;
3336          ;               004000  TO11DB=004000   ;-10 REQUESTED -11 INTERRUPT
3337                                  ;
3338                                  ;
3339                                  ; DEFINITIONS . . .
3340                                  ;
3341          ;               000040  ROTOR7= 40      ;SAVE R0 TO R7 IN 40 TO 56
3342                                  ;
3343          ;               000130  DTESAV= 130     ;SAVE FIRST 12 DTE REGISTERS DLYCNT TO TO11DT
3344          ;                                       ; IN LOCATIONS 130-156
3345                                  ;
3346          ;               173000  ROMORG= 173000  ;ROM STARTS AT 773000
3347                                  ;
3348                                  ; ESTABLISH ROM ORIGIN
3349                                  ;
3350          ;               173000          .=ROMORG
```

```
3351            ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 2
3352
3353            ;                               .SBTTL  BOOT FROM MASTER FRONT END
3354            ;
3355                                    ;EXTERNAL BUTTON #1 -- LOAD FROM MASTER FRONT END VIA DL11E
3356                                    ;
3357                                    ;DEPRESSING BUTTON #1 CAUSES A LOADER UTILITY PROGRAM (DGQDE) TO BE LOADED
3358                                    ;
3359                                    ;TO ACCOMPLISH THIS, THE FOLLOWING CONDITIONS MUST EXIST:
3360                                    ;       1) THE LOADER UTILITY PROGRAM DGQDD MUST BE RUNNING UNDER KLDCP
3361                                    ;          IN THE MASTER FRONT END.
3362                                    ;       2) THE '.A11' FILE FOR DGQDE MUST BE AVAILABLE ON THE SELECTED LOAD
3363                                    ;          MEDIUM IN THE MASTER FRONT END.
3364                                    ;       3) THE MASTER FRONT END AND THE SECONDARY FRONT END MUST BE CONNECTED
3365                                    ;          THROUGH DL11E'S AND A NULL MODEM.
3366                                    ;       4) THE SECONDARY FRONT END MUST HAVE A TTY CONNECTED TO IT.
3367            ;
3368                                    ;SEND BOOTSTRAP REQUEST TO MASTER FRONT END VIA THE DL11E.  THE
3369                                    ;REQUEST IS IN THE FORM 'B <CR><LF>', WHICH CALLS FOR THE
3370                                    ;BOOTING OVER OF DGQDE.A11.
3371  006000  000005  ;173000 000005  BUTON1: RESET                   ;CLEAR THE WORLD
3372  006002  012706  ;173002 012706         MOV     #2000,SP        ;SETUP STACK POINTER
3373  006004  002000  ;173004 002000
3374  006006  012701  ;173006 012701         MOV     #DGQDE,R1       ;ADDRESS OF SYNC + BOOT REQUEST
3375  006010  173374  ;173010 173374
3376  006012  105737  ;173012 105737  SENDIT: TSTB    @#DLXCSR        ;IS DL11E READY TO TRANSMIT
3377  006014  176004  ;173014 176004
3378  006016  100375  ;173016 100375         BPL     SENDIT          ;LOOP UNTIL IT IS
3379  006020  000403  ;173020 000403         BR      1$              ;BRANCH AROUND POWER -FAIL VECTOR
3380                                    ;*****************************************************************
3381                                    ;       REQUIRED POWER-FAIL VECTOR   - MUST BE AT 173024
3382            ;
3383  006022  000000  ;173022 000000         .WORD   0               ;FILLER
3384  006024  173000  ;173024 173000         .WORD   ROMORG,PR7
3385  006026  000340  ;173026 000340
3386            ;
3387                                    ;*****************************************************************
3388            ;
3389  006030  112137  ;173030 112137  1$:     MOVB    (R1)+,@#DLXBUF  ;LOAD A CHAR INTO OUTPUT BUFFER
3390  006032  176006  ;173032 176006
3391  006034  105711  ;173034 105711         TSTB    (R1)            ;ANY MORE CHARS TO SEND?
3392  006036  001365  ;173036 001365         BNE     SENDIT          ;BRANCH IF MORE CHARS TO SEND
3393  006040  005005  ;173040 005005         CLR     R5              ;CLEAR SYNC RECEIVED FLAG
3394                                    ;WAIT TO RECEIVE BOOT PROGRAM (DGQDE.A11), ONE ASCIIZED CHAR AT
3395                                    ;A TIME, ONE LINE AT A TIME.
3396  006042  012701  ;173042 012701  NXTLIN: MOV     #LINBUF,R1      ;ADDRESS OF LINE INPUT BUFFER
3397  006044  002100  ;173044 002100
3398  006046  105737  ;173046 105737  NXTCHR: TSTB    @#DLRCSR        ;CHAR RECEIVED YET?
3399  006050  176000  ;173050 176000
3400  006052  100375  ;173052 100375         BPL     NXTCHR          ;BRANCH IF STILL TO WAIT
3401            ;
3402                                    ;PROCESS THE RECEIVED ASCIIZED CHAR.  IGNORE ALL CHARS UNTIL A SYNC
3403                                    ;SIGNAL IS RECEIVED.  A LINE FEED MARKS THE END OF A LINE.  THE MAX
3404                                    ;NUMBER OF CHARS PER LINE IS 131, MORE THAN THAT IS AN ERROR.
```

```
3405  006054  113711  ;173054 113711  2$:     MOVB    @#DLRBUF,(R1)   ;READ CHAR INTO LINE INPUT BUFFER
3406  006056  176002  ;173056 176002
3407  006060  142711  ;173060 142711          BICB    #200,(R1)       ;CLEAR HIGH ORDER BIT OF CHAR
3408  006062  000200  ;173062 000200
3409  006064  001770  ;173064 001770          BEQ     NXTCHR          ;BRANCH IF YES, IGNORE NULLS
3410  006066  121127  ;173066 121127          CMPB    (R1),#SYN       ;IS CHAR THE SYNC SIGNAL
3411  006070  000001  ;173070 000001
3412  006072  001413  ;173072 001413          BEQ     3$              ;BRANCH IF YES
3413  006074  005705  ;173074 005705          TST     R5              ;HAS SYNC ALREADY BEEN RECEIVED?
3414  006076  001763  ;173076 001763          BEQ     NXTCHR          ;BRANCH IF NOT, IGNORE CHAR
3415  006100  122127  ;173100 122127          CMPB    (R1)+,#LF       ;IS CHAR A LINE FEED?
3416  006102  000012  ;173102 000012
3417  006104  001410  ;173104 001410          BEQ     PACKIT          ;BRANCH IF YES, END OF LINE
```

```
3418                    ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76  09:14  PAGE 2-1
3419
3420  006106  020127    ;173106 020127            CMP     R1,#LINBUF+132. ;HAVE WE REACHED THE END OF THE BUFFER?
3421  006110  002304    ;173110 002304
3422  006112  003755    ;173112 003755            BLE     NXTCHR          ;BRANCH IF NOT, GET REST OF LINE
3423  006114  004737    ;173114 004737            JSR     PC,NAKSND       ;SEND A NEG ACKNOWLEDGE
3424  006116  173346    ;173116 173346
3425  006120  000000    ;173120 000000            HALT                    ;TOO MANY CHARS IN A LINE
3426                    ;
3427  006122  005205    ;173122 005205  3$:       INC     R5              ;SET SYNC RECEIVED FLAG
3428  006124  000750    ;173124 000750            BR      NXTCHR          ;GO GET NEXT CHAR
3429                    ;
3430                                    ;UNSCRAMBLE THE ASCIIZED CHARS INTO 16-BIT WORDS.  THE FORMAT OF A
3431                                    ;LINE IS   E WRDCNT,LADDR,DATA,DATA,...,DATA,CHKSUM<CR><LF>
3432                                    ;WHERE WRDCNT IS THE WORD COUNT
3433                                    ;      LADDR IS THE LOAD ADDRESS
3434                                    ;      DATA IS LOAD DATA
3435                                    ;      CHKSUM IS THE CHECKSUM
3436                    ;
3437  006126  012703    ;173126 012703  PACKIT:  MOV     #DEABUF,R3      ;GET ADDRESS OF DE-ASCIIZED BUFFER
3438  006130  002310    ;173130 002310
3439  006132  012701    ;173132 012701            MOV     #LINBUF,R1      ;GET ADDRESS OF INPUT BUFFER
3440  006134  002100    ;173134 002100
3441  006136  122127    ;173136 122127            CMPB    (R1)+,#'E       ;FIRST CHAR IN LINE SHOULD BE AN 'E'
3442  006140  000105    ;173140 000105
3443  006142  001403    ;173142 001403            BEQ     1$              ;BRANCH IF IT IS
3444  006144  004737    ;173144 004737            JSR     PC,NAKSND       ;SEND A NEG ACKNOWLEDGE
3445  006146  173346    ;173146 173346
3446  006150  000000    ;173150 000000            HALT                    ;INCORRECT LINE SYNTAX, FIRST CHAR NOT AN E
3447                    ;
3448  006152  122127    ;173152 122127  1$:       CMPB    (R1)+,#SPACE    ;SECOND CHAR SHOULD BE A SPACE
3449  006154  000040    ;173154 000040
3450  006156  001403    ;173156 001403            BEQ     NXTWRD          ;BRANCH IF IT IS
3451  006160  004737    ;173160 004737            JSR     PC,NAKSND       ;SEND A NEG ACKNOWLEDGE
3452  006162  173346    ;173162 173346
3453  006164  000000    ;173164 000000            HALT                    ;INCORRECT LINE SYNTAX, 2ND CHAR NOT A SPACE
3454                    ;
3455  006166  005002    ;173166 005002  NXTWRD:  CLR     R2              ;CLEAR WORD FORMER
3456  006170  112100    ;173170 112100  1$:      MOVB    (R1)+,R0        ;READ CHAR FROM LINE BUFFER
3457  006172  122700    ;173172 122700            CMPB    #CR,R0          ;IS CHAR A CARRIAGE RETURN
3458  006174  000015    ;173174 000015
3459  006176  001774    ;173176 001774            BEQ     1$              ;BRANCH IF YES
3460  006200  122700    ;173200 122700            CMPB    #LF,R0          ;IS CHAR A LINE FEED
3461  006202  000012    ;173202 000012
3462  006204  001422    ;173204 001422            BEQ     3$              ;BRANCH IF IT IS
3463  006206  122700    ;173206 122700            CMPB    #COMMA,R0       ;IS CHAR A COMMA
3464  006210  000054    ;173210 000054
3465  006212  001415    ;173212 001415            BEQ     2$              ;BRANCH IF IT IS
3466  006214  006302    ;173214 006302            ASL     R2              ;SHIFT  WORD OVER TO MAKE ROOM FOR
3467  006216  006302    ;173216 006302            ASL     R2              ;NEXT CHAR
3468  006220  006302    ;173220 006302            ASL     R2
3469  006222  000402    ;173222 000402            BR      4$              ;BRANCH AROUND POWER-FAIL VECTOR
3470                                    ;*************************************************************
3471                                    ;       REQUIRED POWER-FAIL VECTOR   - MUST BE AT 173224
```

```
3472                    ;
3473  006224  173000    ;173224 173000          .WORD   ROMORG,PR7
3474  006226  000340    ;173226 000340
3475                    ;
3476                    ;*****************************************************************
3477                    ;
3478  006230  006302    ;173230 006302  4$:     ASL     R2
3479  006232  006302    ;173232 006302          ASL     R2
3480  006234  006302    ;173234 006302          ASL     R2
3481  006236  042700    ;173236 042700          BIC     #100,R0         ;CLEAR ASCIIZED BIT
3482  006240  000100    ;173240 000100
3483  006242  050002    ;173242 050002          BIS     R0,R2           ;INSERT NEW CHAR INTO WORD
3484  006244  000751    ;173244 000751          BR      1$              ;GO GET NEXT CHAR
3485                    ;
3486  006246  010223    ;173246 010223  2$:     MOV     R2,(R3)+        ;STORE WORD IN BUFFER
```

```
3487                    ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 2-2
3488
3489  006250  000746    ;173250 000746          BR      NXTWRD          ;GO FORM NEXT WORD
3490                    ;
3491  006252  010223    ;173252 010223  3$:     MOV     R2,(R3)+        ;STORE CHECKSUM IN BUFFER
3492                    ;
3493                                    ;VERIFY THAT THE CHECKSUM IS ZERO.
3494  006254  012702    ;173254 012702  CHCKIT: MOV     #DEABUF,R2      ;ADDRESS OF BUFFER
3495  006256  002310    ;173256 002310
3496  006260  005000    ;173260 005000          CLR     R0              ;CLEAR CHECKSUM
3497  006262  062200    ;173262 062200  1$:     ADD     (R2)+,R0        ;ADD NEXT WORD TO CHECKSUM
3498  006264  020203    ;173264 020203          CMP     R2,R3           ;REACHED END OF BUFFER YET
3499  006266  002775    ;173266 002775          BLT     1$              ;BRANCH IF NOT
3500  006270  005700    ;173270 005700          TST     R0              ;IS CHECKSUM = 0?
3501  006272  001403    ;173272 001403          BEQ     LOADIT          ;BRANCH IF YES
3502  006274  004737    ;173274 004737          JSR     PC,NAKSND       ;SEND A NEG ACKNOWLEDGE
3503  006276  173346    ;173276 173346
3504  006300  000000    ;173300 000000          HALT                    ;CHECKSUM ERROR
3505                    ;
3506                                    ;LOAD THE RECEIVED DATA WORDS INTO THE DESIGNATED PLACE IN MEMORY.
3507                                    ;A WORD COUNT OF ZERO INDICATES A TRANSFER BLOCK.  START EXECUTING
3508                                    ;THE LOADED PROGRAM AT THE SPECIFIED ADDRESS.
3509  006302  013700    ;173302 013700  LOADIT: MOV     DEABUF,R0       ;GET LOAD WORD COUNT
3510  006304  002310    ;173304 002310
3511  006306  001413    ;173306 001413          BEQ     2$              ;BRANCH IF IT'S ZERO (A TRANSFER BLOCK)
3512  006310  012702    ;173310 012702          MOV     #DEABUF+4,R2    ;ADDRESS OF FIRST DATA WORD
3513  006312  002314    ;173312 002314
3514  006314  013701    ;173314 013701          MOV     DEABUF+2,R1     ;GET LOAD ADDRESS
3515  006316  002312    ;173316 002312
3516  006320  112221    ;173320 112221  1$:     MOVB    (R2)+,(R1)+     ;MOVE DATA FROM BUFFER TO MEMORY
3517  006322  112221    ;173322 112221          MOVB    (R2)+,(R1)+     ;MOVE DATA FROM BUFFER TO MEMORY
3518  006324  005300    ;173324 005300          DEC     R0              ;DECREMENT WORD COUNT
3519  006326  003374    ;173326 003374          BGT     1$              ;BRANCH UNTIL ALL DATA IS LOADED
3520  006330  004737    ;173330 004737          JSR     PC,ACKSND       ;GO SEND AN ACK
3521  006332  173354    ;173332 173354
3522  006334  000642    ;173334 000642          BR      NXTLIN          ;GO GET NEXT LINE
3523                    ;
3524  006336  004737    ;173336 004737  2$:     JSR     PC,ACKSND       ;GO SEND AN ACK
3525  006340  173354    ;173340 173354
3526  006342  013707    ;173342 013707          MOV     DEABUF+2,PC     ;START ADDRESS OF LOADED PROGRAM
3527  006344  002312    ;173344 002312
3528                    ;
3529                    ;
3530                                    ;NAKSND IS USED TO SEND A NEG ACK BACK TO THE MASTER FRONT END.
3531                                    ;ACKSND IS USED TO SEND AN ACK.
3532  006346  012700    ;173346 012700  NAKSND: MOV     #NAK,R0         ;SETUP ASCII NEG ACK
3533  006350  000025    ;173350 000025
3534  006352  000402    ;173352 000402          BR      RESPND          ;GO SEND IT
3535  006354  112700    ;173354 112700  ACKSND: MOVB    #ACK,R0         ;SETUP ASCII ACK
3536  006356  000006    ;173356 000006
3537  006360  105737    ;173360 105737  RESPND: TSTB    @#DLXCSR        ;IS TRANSMITTER READY?
3538  006362  176004    ;173362 176004
3539  006364  100375    ;173364 100375          BPL     RESPND          ;WAIT TIL IT IS
3540  006366  110037    ;173366 110037          MOVB    R0,@#DLXBUF     ;SEND ACK/NAK
```

```
3541  006370  176006  ;173370 176006
3542  006372  000207  ;173372 000207          RTS     PC              ;RETURN TO CALLING ROUTINE
3543                  .
3544  006374  041001  ;173374    001  DGQDE:  .BYTE   SYN
3545                  ;173375    102         .ASCIZ  /B/<CR><LF>
3546  006376  005015  ;173376 005015
3547  006400  000000  ;173400    000
3548                  ;173401    000         .BYTE   0
3549  006402  000000  ;173402    000         .BYTE   0
3550                  ;173403    000         .BYTE   0
3551  006404  000000  ;173404    000         .BYTE   0
3552                  ;173405    000         .BYTE   0
3553  006406  000000  ;173406    000         .BYTE   0
3554                  ;173407    000         .BYTE   0
3555  006410  000000  ;173410    000         .BYTE   0
```

```
3556                    ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 2-3
3557
3558                            ;173411    000             .BYTE   0
3559    006412  000000          ;173412    000             .BYTE   0
3560                            ;173413    000             .BYTE   0
3561    006414  000000          ;173414    000             .BYTE   0
3562                            ;173415    000             .BYTE   0
3563    006416  000000          ;173416    000             .BYTE   0
3564                            ;173417    000             .BYTE   0
3565    006420  000000          ;173420    000             .BYTE   0
3566                            ;173421    000             .BYTE   0
3567    006422  000000          ;173422    000             .BYTE   0
3568                            ;173423    000             .BYTE   0
3569    006424  000000          ;173424    000             .BYTE   0
3570                            ;173425    000             .BYTE   0
3571    006426  000000          ;173426    000             .BYTE   0
3572                            ;173427    000             .BYTE   0
3573    006430  000000          ;173430    000             .BYTE   0
3574                            ;173431    000             .BYTE   0
3575    006432  000000          ;173432    000             .BYTE   0
3576                            ;173433    000             .BYTE   0
3577    006434  000000          ;173434    000             .BYTE   0
3578                            ;173435    000             .BYTE   0
3579    006436  000000          ;173436    000             .BYTE   0
3580                            ;173437    000             .BYTE   0
3581    006440  000000          ;173440    000             .BYTE   0
3582                            ;173441    000             .BYTE   0
3583    006442  000000          ;173442    000             .BYTE   0
3584                            ;173443    000             .BYTE   0
3585    006444  000000          ;173444    000             .BYTE   0
3586                            ;173445    000             .BYTE   0
3587    006446  000000          ;173446    000             .BYTE   0
3588                            ;173447    000             .BYTE   0
3589    006450  000000          ;173450    000             .BYTE   0
3590                            ;173451    000             .BYTE   0
3591    006452  000000          ;173452    000             .BYTE   0
3592                            ;173453    000             .BYTE   0
3593    006454  000000          ;173454    000             .BYTE   0
3594                            ;173455    000             .BYTE   0
3595    006456  000000          ;173456    000             .BYTE   0
3596                            ;173457    000             .BYTE   0
3597    006460  000000          ;173460    000             .BYTE   0
3598                            ;173461    000             .BYTE   0
3599    006462  000000          ;173462    000             .BYTE   0
3600                            ;173463    000             .BYTE   0
3601    006464  000000          ;173464    000             .BYTE   0
3602                            ;173465    000             .BYTE   0
3603    006466  000000          ;173466    000             .BYTE   0
3604                            ;173467    000             .BYTE   0
3605    006470  000000          ;173470    000             .BYTE   0
3606                            ;173471    000             .BYTE   0
3607    006472  000000          ;173472    000             .BYTE   0
3608                            ;173473    000             .BYTE   0
3609    006474  000000          ;173474    000             .BYTE   0
```

```
3610                        ;173475    000          .BYTE   0
3611   006476   000000      ;173476    000          .BYTE   0
```

```
3612                    ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 2-4
3613
3614                            ;173477    000             .BYTE   0
3615  006500  000000            ;173500    000             .BYTE   0
3616                            ;173501    000             .BYTE   0
3617  006502  000000            ;173502    000             .BYTE   0
3618                            ;173503    000             .BYTE   0
3619  006504  000000            ;173504    000             .BYTE   0
3620                            ;173505    000             .BYTE   0
3621  006506  000000            ;173506    000             .BYTE   0
3622                            ;173507    000             .BYTE   0
3623  006510  000000            ;173510    000             .BYTE   0
3624                            ;173511    000             .BYTE   0
3625  006512  000000            ;173512    000             .BYTE   0
3626                            ;173513    000             .BYTE   0
3627  006514  000000            ;173514    000             .BYTE   0
3628                            ;173515    000             .BYTE   0
3629  006516  000000            ;173516    000             .BYTE   0
3630                            ;173517    000             .BYTE   0
3631  006520  000000            ;173520    000             .BYTE   0
3632                            ;173521    000             .BYTE   0
3633  006522  000000            ;173522    000             .BYTE   0
3634                            ;173523    000             .BYTE   0
3635  006524  000000            ;173524    000             .BYTE   0
3636                            ;173525    000             .BYTE   0
3637  006526  000000            ;173526    000             .BYTE   0
3638                            ;173527    000             .BYTE   0
3639  006530  000000            ;173530    000             .BYTE   0
3640                            ;173531    000             .BYTE   0
3641  006532  000000            ;173532    000             .BYTE   0
3642                            ;173533    000             .BYTE   0
3643  006534  000000            ;173534    000             .BYTE   0
3644                            ;173535    000             .BYTE   0
3645  006536  000000            ;173536    000             .BYTE   0
3646                            ;173537    000             .BYTE   0
3647                    ;                                   .EVEN
```

```
3648          ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 3
3649
3650          ;                                       .SBTTL  REGISTER SAVE ROUTINE
3651                                                ;       REGISTER SAVE ROUTINE
3652                                                ;
3653                                                ; REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS R0-R7
3654                                                ; IN MEMORY AT 40-56 (LOCATION R0TOR7).
3655                                                ;
3656                                                ; CALLING SEQUENCE:
3657                                                ;       MOV     R0,R0TOR7+0
3658                                                ;       MOV     #RET,R0
3659                                                ;       BR      REGSAV
3660                                                ; RET:  <RETURN HERE>
3661                                                ;
3662                                                ; ALL REGISTERS RESTORED
3663                                                ;
3664                                                ;
3665          ;                  173540                      .=ROMORG+540
3666          ;
3667   006540 010037  ;173540 010037  REGSAV: MOV     R0,R0TOR7+16    ;SAVE R0 AS PC IN 56
3668   006542 000056  ;173542 000056
3669          ;
3670   006544 012700  ;173544 012700          MOV     #R0TOR7+16,R0   ;R0 NOW POINTS TO 56
3671   006546 000056  ;173546 000056
3672          ;
3673   006550 010640  ;173550 010640          MOV     SP,-(R0)        ;SAVE SP IN 54
3674   006552 010540  ;173552 010540          MOV     R5,-(R0)        ;SAVE R5 IN 52
3675   006554 010440  ;173554 010440          MOV     R4,-(R0)        ;SAVE R4 IN 50
3676   006556 010340  ;173556 010340          MOV     R3,-(R0)        ;SAVE R3 IN 46
3677   006560 010240  ;173560 010240          MOV     R2,-(R0)        ;SAVE R2 IN 44
3678   006562 010140  ;173562 010140          MOV     R1,-(R0)        ;SAVE R1 IN 42
3679   006564 014000  ;173564 014000          MOV     -(R0),R0        ;RESTORE R0 FROM 40
3680   006566 000177  ;173566 000177          JMP     @R0TOR7+16      ;GO TO SAVED PC
3681   006570 004264  ;173570 004264
3682          ;
3683   006572 000000  ;173572 000000          .WORD   0               ;FILLER WORD
```

```
3684            ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 4
3685
3686            ;                                       .SBTTL  DUMP AND BOOTSTRAP THROUGH DTE20
3687                                                    ;       INTERNAL BUTTON #4 -- DUMP AND BOOTSTRAP THROUGH DTE20
3688                                                    ;
3689            ;               000040          DTESIZ= 40      ;EACH DTE OCCUPIES 20 WORDS IN EXTERNAL PAGE
3690                                                    ;
3691                                                    ; BUTTON #4 -- INITIATED BY '-10 RELOAD -11' BIT
3692                                                    ;
3693            ;               173574                  .=ROMORG+574
3694            ;
3695  006574  010037  ;173574 010037  BUTON4: MOV     R0,ROTOR7+0     ;SAVE R0 IN 40
3696  006576  000040  ;173576 000040
3697  006600  012700  ;173600 012700          MOV     #10$,R0         ;SET RETURN ADDRESS IN R0
3698  006602  173606  ;173602 173606
3699  006604  000755  ;173604 000755          BR      REGSAV          ;SAVE R1-R7
3700                                                  ;
3701                                                  ; REGISTERS SAVED-- LOOK FOR THE DTE20 WHICH PUSHED THE BUTTON
3702                                                  ;
3703                                                  ; THE DTE WHICH PUSHED THE BUTTON SHOULD HAVE THE DOORBELL
3704                                                  ;  RINGING AND HAVE THE VALUE 1365 (OCTAL) IN IT'S
3705                                                  ;  TO -10 BYTE COUNT TO10BC.
3706                                                  ;
3707                                                  ; NXM (TIME-OUT) TRAP IS USED TO SKIP NON-EXISTANT DTE20'S.
3708                                                  ;
3709  006606  005005  ;173606 005005  10$:    CLR     R5              ;ADDRESS LOCATION ZERO
3710  006610  012500  ;173610 012500          MOV     (R5)+,R0        ;SAVE 0 IN R0
3711  006612  012501  ;173612 012501          MOV     (R5)+,R1        ;SAVE 2 IN R1
3712  006614  011502  ;173614 011502          MOV     (R5),R2         ;SAVE 4 IN R2
3713  006616  012725  ;173616 012725          MOV     #21$,(R5)+      ;SET NXM TRAP ADDRESS IN 4
3714  006620  173634  ;173620 173634
3715  006622  011503  ;173622 011503          MOV     (R5),R3         ;SAVE 6 IN R3
3716  006624  012715  ;173624 012715          MOV     #PR7,(R5)       ;SET PRIORITY FOR NXM TRAP
3717  006626  000340  ;173626 000340
3718                                                  ;
3719                                                  ; LOOP THROUGH ALL DTE'S
3720                                                  ;
3721  006630  012704  ;173630 012704  20$:    MOV     #DLYCNT-DTESIZ,R4 ;POINT TO DTE # -1'S DELAY COUNT REGISTER
3722  006632  174340  ;173632 174340
3723                                                  ; (WILL BUMP TO # 0)
3724                                                  ;
3725                                                  ; HERE ON NXM TRAP-- RESET SP AND TRY NEXT DTE
3726                                                  ;
3727  006634  012706  ;173634 012706  21$:    MOV  *  #4,SP           ;SET SP TO 4, STACK IS LOCATIONS 2 AND 0
3728  006636  000004  ;173636 000004
3729                                                  ;
3730  006640  062704  ;173640 062704  22$:    ADD     #DTESIZ,R4      ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS
3731  006642  000040  ;173642 000040
3732  006644  105704  ;173644 105704          TSTB    R4              ;IS THIS THE END OF THE DTE'S?
3733            ;                                                     ;  NOTE THAT THE LAST DTE IS AT 774540
3734            ;                                                     ;  AND THAT NOW R4= 774600 IF END
3735  006646  100770  ;173646 100770          BMI     20$             ;YES-- START ALL OVER, UNTIL A DTE
3736            ;                                                     ;  SAYS HE PUSHED THE BUTTON
3737  006650  032764  ;173650 032764          BIT     #TO11DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?
```

```
                        ,
3738  006652  004000  ;173652 004000
3739  006654  000034  ;173654 000034
3740  006656  001770  ;173656 001770          BEQ     22$                 ;NO-- TRY NEXT DTE
3741  006660  026417  ;173660 026417          CMP     T010BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365
3742  006662  000014  ;173662 000014
3743                  ;                                                   ; IN IT'S TO -10 BYTE COUNT?
3744  006664  001365  ;173664 001365          BNE     22$                 ;NO-- TRY ANOTHER DTE
3745                  ;
```

```
3746            ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 5
3747
3748                                    ;
3749                                    ; WE HAVE FOUND THE DTE WHICH PUSHED THE BUTTON
3750                                    ;
3751                                    ; ADDRESS OF DLYCNT REGISTER IS IN R4
3752                                    ;
3753  006666  010315  ;173666 010315           MOV     R3,(R5)         ;RESTORE LOCATION 6
3754  006670  010245  ;173670 010245           MOV     R2,-(R5)        ; 4
3755  006672  010145  ;173672 010145           MOV     R1,-(R5)        ; 2
3756  006674  010045  ;173674 010045           MOV     R0,-(R5)        ; 0
3757                                    ;
3758                                    ; SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
3759                                    ;  IN LOCATIONS 130-156
3760                                    ;
3761  006676  012700  ;173676 012700           MOV     #DTESAV,R0      ;POINT TO SAVE AREA
3762  006700  000130  ;173670 000130
3763  006702  012420  ;173702 012420  29$:     MOV     (R4)+,(R0)+     ;SAVE A REGISTER
3764  006704  022700  ;173704 022700           CMP     #T011DT-DLYCNT+DTESAV,R0 ;FINISHED?
3765  006706  000156  ;173706 000156
3766  006710  103374  ;173710 103374           BHIS    29$             ;NO-- SAVE SOME MORE
3767
3768                                    ; R4= T011DT+2
3769                                    ;
3770                                    ; SET R1= STATUS REGISTER
3771                                    ;     R4= DIAG2 REGISTER
3772                                    ;
3773                                    ; DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
3774                                    ;  LOADED FLAG
3775                                    ;
3776
3777  006712  005724  ;173712 005724           TST     (R4)+
3778  006714  010401  ;173714 010401           MOV     R4,R1           ; SO DOES R1
3779  006716  012700  ;173716 012700           MOV     #DRESET,R0      ;SETUP R0 FOR 'DIAGNOSTIC RESET'
3780  006720  000100  ;173720 000100
3781  006722  010021  ;173722 010021           MOV     R0,(R1)+        ;R1 POINTS TO STATUS REGISTER
```

```
3782          ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 6
3783
3784                                  ;
3785                                  ; REGISTERS:
3786                                  ;       R0 --   DRESET (DIAGNOSTIC RESET FUNCTION)
3787                                  ;       R1 --   STAT (STATUS REGISTER)
3788                                  ;       R4 --   DIAG2 (DIAGNOSTIC REGISTER #2, WHERE DRESET IS)
3789                                  ;
3790                                  ; THE -10 WILL NOW START READING -11 MEMORY, AS SOON AS WE SET
3791                                  ;  THE TO -10 ADDRESS.  WHEN FINISHED, THE -10 WILL RING OUR DOORBELL.
3792                                  ;
3793  006724  005061  ;173724 005061          CLR     DLYCNT-STAT(R1) ;SET DTE20 FOR MAXIMUM DELAY (ZERO)
3794  006726  177744  ;173726 177744
3795  006730  005061  ;173730 005061          CLR     TO10AD-STAT(R1) ;START DUMPING -11 MEMORY TO -10
3796  006732  177764  ;173732 177764
3797                                                                  ; STARTING AT LOCATION 0
3798  006734  032711  ;173734 032711  30$:    BIT     #TO11DB,(R1)    ;IS DOORBELL RINGING (TRANSFER COMPLETE)?
3799  006736  004000  ;173736 004000
3800  006740  001775  ;173740 001775          BEQ     30$             ;NO-- WAIT FOR DOORBELL
3801  006742  010014  ;173742 010014          MOV     R0,(R4)         ;YES-- CLEAR DOORBELL AND ERROR FLAGS
3802                                  ;
3803                                  ; NOW THE -10 WILL GIVE US A 256 WORD BOOTSTRAP TO BE READ
3804                                  ;  INTO -11 MEMORY STARTING AT LOCATION 0.  WHEN FINISHED,
3805                                  ;  THE -10 WILL RING OUR DOORBELL, AND WE WILL START EXECUTION
3806                                  ;  OF THE LOADED CODE AT LOCATION 0.
3807                                  ;
3808  006744  005061  ;173744 005061          CLR     TO11AD-STAT(R1) ;START INPUT TO LOCATION 0
3809  006746  177766  ;173746 177766
3810  006750  012761  ;173750 012761          MOV     #IFLOP!<<-256.>&7777>,TO11BC-STAT(R1) ;256 WORDS, INTERRUPT
3811  006752  107400  ;173752 107400
3812  006754  177762  ;173754 177762
3813                  ;                                               ; -10 WHEN DONE
3814  006756  032711  ;173756 032711  40$:    BIT     #TO11DB,(R1)    ;DOORBELL RINGING (LOAD FINISHED)?
3815  006760  004000  ;173760 004000
3816  006762  001775  ;173762 001775          BEQ     40$             ;NO-- WAIT UNTIL DONE
3817  006764  010014  ;173764 010014          MOV     R0,(R4)         ;CLEAR DOORBELL RINGING
3818  006766  012705  ;173766 012705          MOV     #BIT15,R5       ;SET R5: BIT15= 1, BIT0= 0
3819  006770  100000  ;173770 100000
3820                  ;                                               ; TO SAY BUTTON #4 PRESSED
3821  006772  005007  ;173772 005007          CLR     PC              ;GO TO LOADED CODE, STARTING AT
3822                  ;                                               ; LOCATION 0
3823                                  ;
3824                                  ;       FILL TO END OF ROM
3825                                  ;
3826                                  ;
3827  006774  000000; ;173774    000         .BYTE   0
3828                  ;173775    000         .BYTE   0
3829  006776  END.YG:
3830  006776  000000; ;173776    000         .BYTE   0
3831                  ;173777    000         .BYTE   0
3832          ;               000001         .END
```

```
3833  007000  MAP.YH:
3834          ;THE FOLLOWING IS A REPRODUCTION
3835          ;OF THE ROM PROGRAM FOR BM873YH.
3836          ;IT IS HERE FOR COMPARISON TO THE
3837          ;ACTUAL ROM AND FOR REFERENCE
3838          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 1
3839
3840          ;                               .SBTTL  TITLE PAGE
3841          ;                               .TITLE  BM873H  - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)
3842                                ;
3843                                ; COPYRIGHT (C) 1975, 1976 DIGITAL EQUIPMENT CORPORATION
3844                                ; ALL RIGHTS RESERVED
3845                                ;
3846                                ; THIS IS THE CODE TO BE ENCODED IN THE BOOTSTRAP ROM ON THE BM873-YH BOARD
3847                                ;
3848                                ;
3849                                ; MODULE:       BM873H
3850                                ;
3851                                ; DATE:         10-MAR-76
3852                                ;
3853                                ; AUTHOR:       TOM PORCHER
3854                                ;
3855                                ;
3856          ;                               .ENABLE ABS,AMA
3857          ;                               .LIST   MEB
3858          ;                               .MCALL  $DEF
3859                  ;000000                 $DEF
```

```
3860            ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 2
3861
3862            ;                               .SBTTL  MACROS AND DEFINITIONS
3863                                    ;
3864                                    ;
3865                                    ; MACRO TO FILL TO A LOCATION (RELATIVE TO ROM ORIGIN) WITH ZERO BYTES
3866                                    ;
3867            ;                               .MACRO  FILLTO  LOC
3868            ;                       .IFGE <LOC>-<.-ROMORG>
3869            ;                         .IFG <LOC>-<.-ROMORG>
3870            ;                           .IFDF PASS2
3871            ;                               .PRINT  <LOC>-<.-ROMORG> ;FREE BYTES AT LOC
3872            ;                           .ENDC
3873            ;                         .ENDC
3874            ;                         .REPT <LOC>-<.-ROMORG>
3875            ;                               .BYTE   0
3876            ;                         .ENDR
3877            ;                       .IFF
3878            ;                               .ERROR  <.-ROMORG>-<LOC> ;BOUNDARY EXCEEDED AT LOC
3879            ;                       .ENDC
3880            ;                               .ENDM   FILLTO
3881                                    ;
3882                                    ;
3883                                    ; MACRO TO DO 'MOV #XXX,DEST' OR 'CLR DEST' IF XXX IS ZERO
3884                                    ;
3885            ;                               .MACRO  MOVO    XXX,DEST
3886            ;                       .IFEQ XXX
3887            ;                               CLR     DEST
3888            ;                       .IFF
3889            ;                               MOV     #XXX,DEST
3890            ;                       .ENDC
3891            ;                               .ENDM   MOVO
3892
3893
3894                                    ; MACRO TO ADD A SMALL NUMBER TO A REGISTER
3895                                    ;  GENERATES ONE OF THE FOLLOWING:
3896                                    ;       CMP     -(REG),-(REG)   ; -4
3897                                    ;       TST     -(REG)          ; -2
3898                                    ;       CMPB    -(REG),-(REG)   ; -2 (REGISTER MAY BE ODD)
3899                                    ;       DEC     REG             ; -1
3900                                    ;       <NOTHING>               ; 0
3901                                    ;       INC     REG             ; 1
3902                                    ;       TST     (REG)+          ; 2
3903                                    ;       CMPB    (REG)+,(REG)+   ; 2 (REGISTER MAY BE ODD)
3904                                    ;       CMP     (REG)+,(REG)+   ; 4
3905                                    ;       ADD     #XXX,REG        ; ANYTHING ELSE
3906                                    ;
3907                                    ; USE THIS MACRO WITH CAUTION, SINCE IT REFERENCES MEMORY
3908                                    ;  AND ALSO DOES NOT SET THE CONDITION CODES PROPERLY
3909                                    ;
3910            ;                               .MACRO  ADDX    XXX,REG,ODD
3911            ;                       $$$=.
3912            ;                       .IFEQ XXX+4
3913            ;                         .IF B <ODD>
```

```
3914            ;                               CMP     -(REG),-(REG)
3915            ;                       .ENDC
```

```
3916          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 2-1
3917
3918          ;                       .ENDC
3919          ;                       .IFEQ XXX+2
3920          ;                         .IF B <ODD>
3921          ;                               TST     -(REG)
3922          ;                         .IFF
3923          ;                               CMPB    -(REG),-(REG)
3924          ;                         .ENDC
3925          ;                       .ENDC
3926          ;                       .IFEQ XXX+1
3927          ;                               DEC     REG
3928          ;                       .ENDC
3929          ;                       .IFEQ XXX
3930          ;                         $$$=$$$+2
3931          ;                       .ENDC
3932          ;                       .IFEQ XXX-1
3933          ;                               INC     REG
3934          ;                       .ENDC
3935          ;                       .IFEQ XXX-2
3936          ;                         .IF B <ODD>
3937          ;                               TST     (REG)+
3938          ;                         .IFF
3939          ;                               CMPB    (REG)+,(REG)+
3940          ;                         .ENDC
3941          ;                       .ENDC
3942          ;                       .IFEQ XXX-4
3943          ;                         .IF B <ODD>
3944          ;                               CMP     (REG)+,(REG)+
3945          ;                         .ENDC
3946          ;                       .ENDC
3947          ;                       .IFEQ $$$-.
3948          ;                               ADD     #XXX,REG
3949          ;                       .ENDC
3950          ;                               .ENDM   ADDX
3951                                ;
3952                                ;
3953                                ; DEFINITIONS . . .
3954                                ;
3955          ;             000040  ROTOR7= 40                      ;SAVE R0 TO R7 IN 40 TO 56
3956                                ;
3957          ;             000130  DTESAV= 130                     ;SAVE FIRST 12 DTE REGISTERS DLYCNT TO TO11DT
3958          ;                                                     ; IN LOCATIONS 130-156
3959                                ;
3960          ;             000014  RETRY=  12.                     ;DO 12 RETRIES BEFORE HALTING
3961                                ;
3962          ;                 001 .IFNDF  TESTVR
3963          ;             173000  ROMORG= 173000                  ;ROM STARTS AT 773000
3964          ;                     .IFF
3965          ;                     ROMORG= 073000                  ;IF TEST, START AT 73000 INSTEAD
3966          ;                 000 .ENDC
3967                                ;
3968                                ; ESTABLISH ROM ORIGIN
3969                                ;
```

```
3970                ;                       001  .IF DF TESTVR
3971                ;                            .=ROMORG-2
```

```
3972            ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 2-2
3973
3974            ;                                   HALT
3975            ;                           .IFF
3976            ;                 173000    .=ROMORG
3977            ;                    000    .ENDC
```

```
3978            ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 3
3979
3980            ;                               .SBTTL  EXTERNAL BUTTONS #1, #2, #3
3981                                    ;
3982                                    ; BUTTON #1 -- LOAD USING SWITCH REGISTER
3983                                    ;
3984                    ;173000         BUTON1:
3985  007000  010037   ;173000 010037          MOV     R0,ROTOR7+0     ;SAVE R0 IN LOCATION 40
3986  007002  000040   ;173002 000040
3987  007004  013700   ;173004 013700          MOV     SWR,R0          ;GET SWITCH REGISTER
3988  007006  177570   ;173006 177570
3989  007010  032700   ;173010 032700          BIT     #BIT0,R0        ;IS LOW-ORDER BIT SET?
3990  007012  000001   ;173012 000001
3991  007014  001007   ;173014 001007          BNE     BUTONX          ;YES-- LOOK AT CONTENTS
3992  007016  000513   ;173016 000513          BR      REGSAV          ;NO-- SAVE R1-R7 IN 42-56, GO TO ADDRESS IN R0 (FROM SWR
3993                                    ;
3994                                    ; BUTTON #3 -- LOAD BOOT FROM RX11 FLOPPY DISK
3995                                    ;
3996                    ;173020         BUTON3:
3997  007020  005000   ;173020 005000          CLR     R0              ;SAY LOAD FROM FLOPPY, UNIT 0
3998  007022  000404   ;173022 000404          BR      BUTONX          ;GO TO COMMON CODE FOR 3 BUTTONS
3999                                    ;
4000                                    ; REQUIRED POWER-FAIL VECTOR
4001                                    ;
4002                    ;173024                 FILLTO  24
4003  007024  173000   ;173024 173000          .WORD   ROMORG,PR7
4004  007026  000340   ;173026 000340
4005                                    ;
4006                                    ; BUTTON #2 -- LOAD BOOT FROM RP04 DISK
4007                                    ;
4008                    ;173030         BUTON2:
4009  007030  012700   ;173030 012700          MOV     #BIT7,R0        ;BIT 7 MEANS LOAD FROM RP04
4010  007032  000200   ;173032 000200
4011                                    ;       BR      BUTONX          ;FALL INTO COMMON CODE
4012                                    ;
4013                                    ; R0 IS SAVED IN R5 AS THE PARAMETER WORD PASSED TO BOOT
4014                                    ; AND CONTAINS ONE OF THE FOLLOWING:
4015                                    ;
4016                                    ;       BIT 0 = 1       IF FROM SWITCH REGISTER
4017                                    ;       BIT 7 = 0       LOAD FROM RX11 FLOPPY DISK (OR TC11 DECTAPE)
4018                                    ;       BIT 7 = 1       LOAD FROM RP04 DISK
4019                                    ;       BIT 15 = 1      INDEFINITE RETRY
4020                                    ;
4021                    ;173034         BUTONX:
4022  007034  010005   ;173034 010005          MOV     R0,R5           ;SAVE PARAMETER FOR BOOT
4023  007036  106300   ;173036 106300          ASLB    R0              ;LEFT-ALIGN SPEED FIELD IN LOW BYTE
4024  007040  122700   ;173040 122700          CMPB    #3*BIT4,R0      ;IS SPEED 0, 1, OR 2?
4025  007042  000060   ;173042 000060
4026  007044  101001   ;173044 101001          BHI     10$             ;YES-- UNIT IS UNIT TO USE
4027  007046  005000   ;173046 005000          CLR     R0              ;NO-- USE UNIT #0
4028                    ;173050         10$:
4029  007050  000300   ;173050 000300          SWAB    R0              ;GET UNIT # IN LOW BYTE
4030  007052  042700   ;173052 042700          BIC     #^C7,R0         ;TRIM TO 3 BITS 2, 1, 0
4031  007054  177770   ;173054 177770
```

```
                                        ;
                                        ; UNIT # IS IN R0-- CALL PROPER BOOT DEPENDING ON BIT 7
                                        ;
007056  105705  ;173056 105705           TSTB    R5              ;WHERE SHOULD WE BOOT FROM?
007060  100553  ;173060 100553           BMI     RPBOOT          ;BIT 7 = 1 -- BOOT FROM RP04 DISK
                                        ;
                                        ; BIT 7 = 0 -- BOOT FROM RX11 IF IT EXISTS, ELSE TC11
                                        ;
007062  012737  ;173062 012737           MOV     #TCBOOT,4       ;SET TIMEOUT TRAP TO TRY DECTAPE
007064  173304  ;173064 173304
007066  000004  ;173066 000004
```

K08


```
4043                    ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 3-1
4044
4045  007070  005037    ;173070 005037          CLR     6               ; . .
4046  007072  000006    ;173072 000006
4047                                  ;         BR      RXBOOT          ;TRY FLOPPY FIRST
```

```
4048          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 4
4049
4050          ;                               .SBTTL  RX11 FLOPPY DISK BOOTSTRAP ROUTINES
4051                                  ;
4052                                  ; RX11 REGISTER DEFINITIONS
4053                                  ;
4054          ;               177170  RXEPA=  177170                  ;EXTERNAL PAGE ADDR OF FLOPPY
4055                                  ;
4056          ;               000000  RXCS=   0                       ;OFFSET FOR CSR
4057          ;               100000          RXERR=  BIT15           ;ERROR
4058          ;               000200          RXTREQ= BIT7            ;TRANSFER REQUEST
4059          ;               000040          RXDONE= BIT5            ;TRANSFER DONE
4060          ;               000020          RXUNIT= BIT4            ;UNIT NUMBER 1
4061          ;               000016          RXFUNC= BIT3!BIT2!BIT1  ;FUNCTION:
4062          ;               000000                  RXFILL= 0       ; FILL SILO
4063          ;               000002                  RXEMPT= 2       ; EMPTY SILO
4064          ;               000004                  RXWRIT= 4       ; WRITE SECTOR
4065          ;               000006                  RXREAD= 6       ; READ SECTOR
4066          ;               000016                  RXRERR= 16      ; READ ERROR REGISTER
4067          ;               000001          RXGO=   BIT0            ;GO BIT
4068          ;               000002  RXDB=   2                       ;MULTI-PURPOSE DATA BUFFER REGISTER
4069                                  ;
4070                                  ;
4071                                  ;  NOTE THAT THE BOOTSTRAP IS WRITTEN IN LOGICAL BLOCK 0
4072                                  ;  WHICH IS TRACK 1, SECTORS 1, 3, 5, 7.  ONLY SECTOR 1
4073                                  ;  IS READ BY THE ROM.
4074                                  ;
4075                                  ;
4076                                  ; REGISTER USAGE:
4077                                  ;       R0 --   READ FUNCTION WITH UNIT SELECT SET
4078                                  ;       R1 --   ADDRESS OF RXCS
4079                                  ;       R2 --   ADDRESS OF RXDB
4080                                  ;       R3 --   UNIT #
4081                                  ;       R4 --   DATA ADDRESS (TO READ OR WRITE)
4082                                  ;       R5 --   PARAMETER WORD SAVED FROM INITIALIZATION
4083                                  ;       SP --   RETRY COUNTER
4084                                  ;
4085                                  ;
4086                                  ; HERE TO BOOT FROM RX11 FLOPPY DISK-- UNIT # IN R0
4087                                  ;
4088                  ;173074         RXBOOT:
4089  007074  012706  ;173074 012706         MOV     #RETRY,SP       ;SET RETRY COUNT
4090  007076  000014  ;173076 000014
4091  007100  012701  ;173100 012701         MOV     #RXEPA+RXCS,R1  ;ADDRESS CONTROL STATUS REGISTER FOR RX11
4092  007102  177170  ;173102 177170
4093  007104  010003  ;173104 010003         MOV     R0,R3           ;COPY UNIT #
4094                                          BR      RXRTRY          ;FALL THROUGH RETRY CHECK
4095                                  ;
4096                                  ; HERE ON ERROR TO RETRY
4097                                  ;
4098                  ;173106         RXRTRY:
4099  007106  005705  ;173106 005705         TST     R5              ;INDEFINITE RETRY?
4100  007110  100402  ;173110 100402         BMI     RXRSET          ;YES-- TRY FAITHFULLY
4101  007112  005306  ;173112 005306         DEC     SP              ;NO-- DECREMENT RETRY COUNT
```

```
 4102  007114  002445  ;173114 002445          BLT     RXEHLT          ;GIVE UP IF RUN OUT
 4103                                  ;
 4104                                  ; HERE TO START TRANSFER
 4105                                  ;
```

```
4106                    ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 4-1
4107
4108                    ;173116                 RXRSET:
4109  007116  000005    ;173116 000005                  RESET                           ;CLEAR THE WORLD
4110                    ;173120                 20$:
4111  007120  032711    ;173120 032711                  BIT     #RXDONE,(R1)    ;WAIT UNTIL READY FOR FUNCTION
4112  007122  000040    ;173122 000040
4113  007124  001775    ;173124 001775                  BEQ     20$             ;NOT YET-- WAIT
4114                                                    BR      RXPERF          ;JUST DO FIRST READ
4115                                    ;
4116                                    ; HERE TO PERFORM READ, UNIT # IN R0
4117                                    ;
4118                    ;173126         RXPERF:
4119  007126  010300    ;173126 010300                  MOV     R3,R0           ;GET UNIT #
4120  007130  001402    ;173130 001402                  BEQ     5$              ;ZERO-- USE ZERO
4121  007132  012700    ;173132 012700                  MOV     #RXUNIT,R0      ;NON-ZERO-- ASSUME UNIT #1
4122  007134  000020    ;173134 000020
4123                    ;173136                 5$:
4124  007136  052700    ;173136 052700                  BIS     #RXREAD+RXGO,R0 ;SET READ FUNCTION
4125  007140  000007    ;173140 000007
4126                                    ;
4127  007142  010102    ;173142 010102                  MOV     R1,R2           ;COPY ADDRESS OF RXCS
4128  007144  010022    ;173144 010022                  MOV     R0,(R2)+        ;START READ FUNCTION, R2 NOW POINTS TO RXDB
4129                    ;173146                 10$:
4130  007146  105711    ;173146 105711                  TSTB    (R1)            ;READY?
4131  007150  100376    ;173150 100376                  BPL     10$             ;NO-- WAIT
4132  007152  012712    ;173152 012712                  MOV     #1,(R2)         ;SET SECTOR #
4133  007154  000001    ;173154 000001
4134                    ;173156                 20$:
4135  007156  105711    ;173156 105711                  TSTB    (R1)            ;READY FOR TRACK?
4136  007160  100376    ;173160 100376                  BPL     20$             ;NO-- WAIT
4137  007162  012712    ;173162 012712                  MOV     #1,(R2)         ;SET TRACK #
4138  007164  000001    ;173164 000001
4139                    ;173166                 30$:
4140  007166  032711    ;173166 032711                  BIT     #RXERR!RXDONE,(R1) ;DONE OR ERROR?
4141  007170  100040    ;173170 100040
4142  007172  001775    ;173172 001775                  BEQ     30$             ;NO-- WAIT
4143  007174  100744    ;173174 100744                  BMI     RXRTRY          ;YES-- ERROR IN FUNCTION
4144                                    ;
4145                                    ; READ COMPLETED-- EMPTY SILO TO MEMORY
4146                                    ;
4147                    ;173176         RXEMSL:
4148  007176  012711    ;173176 012711                  MOV     #RXEMPT+RXGO,(R1) ;START EMPTY
4149  007200  000003    ;173170 000003
4150  007202  005004    ;173202 005004                  CLR     R4              ;ALWAYS START TRANSFER AT LOCATION ZERO
4151                    ;173204                 10$:
4152  007204  132711    ;173204 132711                  BITB    #RXTREQ!RXDONE,(R1) ;READY FOR WORD, OR TRANSFER DONE?
4153  007206  000240    ;173206 000240
4154  007210  001775    ;173210 001775                  BEQ     10$             ;NOT READY-- WAIT SOME MORE
4155  007212  100153    ;173212 100153                  BPL     CLRPC           ;DONE-- GO TO LOCATION 0
4156  007214  111224    ;173214 111224                  MOVB    (R2),(R4)+      ;NOT DONE-- GET A BYTE FROM SILO TO MEMORY
4157  007216  000772    ;173216 000772                  BR      10$             ;WAIT FOR NEXT BYTE
4158                                    ;
4159                                    ; REQUIRED POWER-FAIL VECTOR
```

```
4160                                    ;
4161                        ;173220             FILLTO  224
4162  007220  000000      ;173220    000        .BYTE   0
4163                        ;173221    000        .BYTE   0
4164  007222  000000      ;173222    000        .BYTE   0
4165                        ;173223    000        .BYTE   0
4166  007224  173000      ;173224 173000        .WORD   ROMORG,PR7
4167  007226  000340      ;173226 000340
4168                                    ;
4169                                    ; HERE ON ERROR AFTER RETRYING -- DISPLAY ERROR REGISTER AND HALT
```

```
4170                    ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 4-2
4171
4172                                                ;
4173                            ;173230             RXEHLT:
4174  007230  012711            ;173230 012711              MOV     #RXRERR+RXGO,(R1) ;DO A READ ERROR REGISTER FUNCTION
4175  007232  000017            ;173232 000017
4176                            ;173234             10$:
4177  007234  032711            ;173234 032711              BIT     #RXDONE,(R1)    ;WAIT UNTIL ERROR ASSEMBLED
4178  007236  000040            ;173236 000040
4179  007240  001775            ;173240 001775              BEQ     10$             ;
4180  007242  011200            ;173242 011200              MOV     (R2),R0         ;GET ERROR REGISTER
4181  007244  000541            ;173244 000541              BR      HALTED          ;HALT AND DISPLAY ERRORS
```

```
4182            ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 5
4183
4184            ;                               .SBTTL  REGISTER SAVE ROUTINE
4185                                    ;
4186                                    ; REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS R0-R7
4187                                    ; IN MEMORY AT 40-56 (LOCATION ROTOR7).
4188                                    ;
4189                                    ; CALLING SEQUENCE:
4190                                    ;       MOV     R0,ROTOR7+0
4191                                    ;       MOV     #RET,R0
4192                                    ;       BR      REGSAV
4193                                    ; RET:  <RETURN HERE>
4194                                    ;
4195                                    ; ALL REGISTERS RESTORED
4196                                    ;
4197                                    ;
4198                    ;173246         REGSAV:
4199  007246  010037    ;173246 010037          MOV     R0,ROTOR7+16    ;SAVE R0 AS PC IN 56
4200  007250  000056    ;173250 000056
4201  007252  012700    ;173252 012700          MOV     #ROTOR7+16,R0   ;R0 NOW POINTS TO 56
4202  007254  000056    ;173254 000056
4203  007256  010640    ;173256 010640          MOV     SP,-(R0)        ;SAVE SP IN 54
4204  007260  010540    ;173260 010540          MOV     R5,-(R0)        ;SAVE R5 IN 52
4205  007262  010440    ;173262 010440          MOV     R4,-(R0)        ;SAVE R4 IN 50
4206  007264  010340    ;173264 010340          MOV     R3,-(R0)        ;SAVE R3 IN 46
4207  007266  010240    ;173266 010240          MOV     R2,-(R0)        ;SAVE R2 IN 44
4208  007270  010140    ;173270 010140          MOV     R1,-(R0)        ;SAVE R1 IN 42
4209  007272  014000    ;173272 014000          MOV     -(R0),R0        ;RESTORE R0 FROM 40
4210  007274  000177    ;173274 000177          JMP     @ROTOR7+16      ;GO TO SAVED PC
4211  007276  004556    ;173276 004556
```

```
4212            ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 6
4213
4214            ;                               .SBTTL  TC11 DECTAPE BOOTSTRAP ROUTINES
4215                                    ;
4216                                    ; TC11 REGISTER DEFINITIONS
4217                                    ;
4218            ;               177340  TCEPA=  177340                  ;EXTERNAL PAGE ADDRESS OF TC-11
4219                                    ;
4220            ;               000000  TCST=   0                       ;STATUS REGISTER
4221            ;               100000          TCENDZ= BIT15           ;END-ZONE DETECTED
4222            ;               000002  TCCM=   2                       ;COMMAND REGISTER
4223            ;               100000          TCERR=  BIT15           ;ERROR
4224            ;               004000          TCREV=  BIT11           ;REVERSE DIRECTION (TOWARD FORWARD END-ZONE)
4225            ;               003400          TCUNIT= BIT10!BIT9!BIT8 ;UNIT SELECT
4226                                    ;
4227            ;               000200          TCRDY=  BIT7            ;READY
4228            ;               000016          TCFUNC= BIT3!BIT2!BIT1  ;FUNCTION:
4229            ;               000000                  TCSATM= 0*BIT1  ; STOP ALL TAPE MOTION
4230            ;               000002                  TCRNUM= 1*BIT1  ; READ BLOCK NUMBER
4231            ;               000004                  TCREAD= 2*BIT1  ; READ DATA
4232            ;               000001          TCGO=   BIT0            ;START FUNCTION
4233            ;               000004  TCWC=   4                       ;WORD COUNT REGISTER
4234            ;               000006  TCBA=   6                       ;BUS ADDRESS REGISTER
4235                                    ;
4236                                    ; REGISTER USAGE:
4237                                    ;       R0 -- UNIT #
4238                                    ;       R1 -- ADDRESS OF TCCM
4239                                    ;       R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
4240                                    ;
4241                                    ; HERE TO START ROM TO BOOT FROM DECTAPE # 0, AS IF
4242                                    ;  DECTAPE BUTTON WERE PUSHED, IN CASE FLOPPY EXISTS.
4243                                    ;
4244                    ;173300         TCBOT0:
4245    007300  005000  ;173300 005000         CLR     R0              ;HERE TO START WITH A FLOPPY, FROM UNIT 0
4246    007302  005005  ;173302 005005         CLR     R5              ;
4247                                    ;       BR      TCBOOT          ;GO BOOT
4248                                    ;
4249                                    ; HERE TO BOOT FROM BLOCK 0 OF DECTAPE, UNIT # IN R0
4250                                    ;
4251                    ;173304         TCBOOT:
4252    007304  012706  ;173304 012706         MOV     #RETRY,SP       ;INIT RETRY COUNTER
4253    007306  000014  ;173306 000014
4254    007310  012701  ;173310 012701         MOV     #TCEPA+TCCM,R1  ;POINT TO COMMAND REGISTER
4255    007312  177342  ;173312 177342
4256                                    ;       BR      TCRTRY          ;TRY IT
4257                                    ;
4258                                    ; HERE ON ERROR TO RETRY
4259                                    ;
4260                    ;173314         TCRTRY:
4261    007314  005705  ;173314 005705         TST     R5              ;INDEFINITE RETRY?
4262    007316  100402  ;173316 100402         BMI     10$             ;YES-- TRY HARDER
4263    007320  005306  ;173320 005306         DEC     SP              ;NO-- DECREMENT COUNT
4264    007322  002427  ;173322 002427         BLT     TCEHLT          ;TOO MANY-- GIVE UP
4265                    ;173324         10$:
```

```
  4266  007324  000005  ;173324 000005          RESET                   ;CLEAR TC11
  4267  007326  010003  ;173326 010003          MOV     R0,R3           ;GET UNIT NUMBER
  4268  007330  000303  ;173330 000303          SWAB    R3              ;TO BITS 10-8
  4269  007332  010304  ;173332 010304          MOV     R3,R4           ;COPY FOR READ FUNCTION
```

```
                ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 6-1

007334  052703  ;173334 052703          BIS     #TCREV+TCRNUM+TCGO,R3 ;START TAPE TOWARD FOWARD END-ZONE (BLOCK 0)
007336  004003  ;173336 004003
007340  010311  ;173340 010311          MOV     R3,(R1)         ; . .
                ;173342         20$:
007342  005711  ;173342 005711          TST     (R1)            ;ERROR?
007344  100376  ;173344 100376          BPL     20$             ;NO-- WAIT FOR END-ZONE ERROR
007346  005761  ;173346 005761          TST     TCST-TCCM(R1)   ;END-ZONE UP YET?
007350  177776  ;173350 177776
007352  100360  ;173352 100360          BPL     TCRTRY          ;NO-- MUST BE OTHER ERROR
                ;
007354  012761  ;173354 012761          MOV     #-256.,TCWC-TCCM(R1) ;SET WORD COUNT
007356  177400  ;173356 177400
007360  000002  ;173360 000002
                ;                                               ;NOTE THAT "RESET" CLEARS BUS ADDRESS REGISTER.
007362  052704  ;173362 052704          BIS     #TCREAD+TCGO,R4 ;START READ, FORWARD
007364  000005  ;173364 000005
007366  010411  ;173366 010411          MOV     R4,(R1)         ; . .
                ;173370         30$:
007370  105711  ;173370 105711          TSTB    (R1)            ;TRANSFER DONE?
007372  100376  ;173372 100376          BPL     30$             ;NO-- WAIT SOME MORE
007374  005711  ;173374 005711          TST     (R1)            ;YES-- ERROR?
007376  100746  ;173376 100746          BMI     TCRTRY          ;YES-- RETRY
007400  000460  ;173400 000460          BR      CLRPC           ;NO-- DONE-- GOTO LOCATION 0
                ;
                ; HERE ON TC11 ERROR
                ;
                ;CEHLT:
                ;173402
007402  016100  ;173402 016100                  MOV     TCST-TCCM(R1),R0 ;GET STATUS REGISTER
007404  177776  ;173404 177776
007406  000460  ;173406 000460                  BR      HALTED          ;AND STOP
```

```
4302          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 7
4303
4304          ;                               .SBTTL  RP04 DISK BOOTSTRAP ROUTINES
4305                                        ;
4306                                        ; RP04 REGISTER DEFINITIONS
4307                                        ;
4308          ;                 176700  RPEPA=  176700                    ;EXTERNAL PAGE ADDRESS OF RP04 REGISTERS
4309                                        ;
4310          ;                 000000  RPCS1=  0                         ;OFFSET FOR CSR #1
4311          ;                 040000          RPTRE=  BIT14             ;TRANFER ERRROR
4312          ;                 020000          RPMCPE= BIT13             ;MASSBUS CONTROL PARITY ERROR
4313          ;                 004000          RPDVA=  BIT11             ;DRIVE AVAILABLE (TO -11)
4314          ;                 000200          RPRDY=  BIT7              ;FUNCTION COMPLETE
4315          ;                 000076          RPFUNC= BIT5!BIT4!BIT3!BIT2!BIT1 ;FUNCTION:
4316          ;                 000020                  RPPRST= 20        ; READ-IN PRESET
4317          ;                 000060                  RPWRIT= 60        ; WRITE DATA
4318          ;                 000070                  RPREAD= 70        ; READ DATA
4319          ;                 000001          RPGO=   BIT0              ;GO
4320          ;                 000002  RPWC=   2                         ;WORD COUNT REGISTER
4321          ;                 000006  RPDA=   6                         ;TRACK (HIGH BYTE) SECTOR (LOW BYTE)
4322          ;                 000010  RPCS2=  10                        ;CONTROL AND STATUS REGISTER #2
4323          ;                 000007          RPUNIT= BIT2!BIT1!BIT0    ;UNIT #
4324          ;                 000012  RPDS=   12                        ;DRIVE STATUS REGISTER
4325          ;                 100000          RPATA=  BIT15             ;ATTENTION ACTIVE
4326          ;                 040000          RPERR=  BIT14             ;DRIVE ERROR
4327          ;                 000014  RPER1=  14                        ;ERROR REGISTER #1
4328          ;                 000020          RPFER=  BIT4              ;FORMAT ERROR
4329          ;                 000032  RPOF=   32                        ;OFFSET REGISTER
4330          ;                 010000          RPFM22= BIT12             ;22 SECTOR (16 BIT) FORMAT
4331          ;                 004000          RPECCI= BIT11             ;INHIBIT ECC CORRECTION
4332          ;                 000034  RPDC=   34                        ;DESIRED CYLINDER
4333                                        ;
4334                                        ;
4335                                        ; REGISTER USAGE:
4336                                        ;       R0 --   UNIT #
4337                                        ;       R1 --   ADDRESS OF RPCS1
4338                                        ;       R2 --   DATA FOR RPOF: RPECCI (ECC INHIBIT), RPFM22 (22 SECTOR FORMAT)
4339                                        ;       R5 --   PARAMETER WORD SAVED FROM INITIALIZATION
4340                                        ;       SP --   RETRY COUNTER
4341                                        ;
4342                                        ;
4343                                        ; HERE TO BOOT FROM RP04-- UNIT # IN R0
4344                                        ;
4345                                        ; START RP04 GOING ON BOOT
4346                                        ;
4347                    ;173410             RPBOOT:
4348  007410  012706    ;173410 012706              MOV     #RETRY,SP       ;RETRY RETRY TIMES
4349  007412  000014    ;173412 000014
4350  007414  012701    ;173414 012701              MOV     #RPEPA+RPCS1,R1 ;ADDRESS RPCS1 IN R1
4351  007416  176700    ;173416 176700
4352  007420  012702    ;173420 012702              MOV     #RPECCI,R2      ;SET ECC INHIBIT, 20 SECTOR MODE
4353  007422  004000    ;173422 004000
4354                                        ;       BR      RPRTRY          ;FALL THROUGH RETRY CODE
4355                                        ;
```

```
                                        ; HERE ON ERROR TO RETRY
                                        ;
                        ;173424         RPRTRY:
007424  005705  ;173424 005705                TST     R5              ;INFINITE RETRY?
007426  100402  ;173426 100402                BMI     10$             ;YES-- TRY AGAIN
```

```
4361            ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 7-1
4362
4363  007430  005306  ;173430 005306          DEC     SP              ;RETRY COUNT EXHAUSTED?
4364  007432  002444  ;173432 002444          BLT     RPEHLT          ;YES-- GIVE UP
4365                  ;173434           10$:
4366  007434  000005  ;173434 000005          RESET                   ;ZAP!!
4367  007436  110061  ;173436 110061          MOVB    R0,RPCS2(R1)    ;SELECT PROPER UNIT #
4368  007440  000010  ;173440 000010
4369  007442  032711  ;173442 032711          BIT     #RPDVA,(R1)     ;IS DRIVE AVAILABLE TO US?
4370  007444  004000  ;173444 004000
4371  007446  001766  ;173446 001766          BEQ     RPRTRY          ;NO-- TRY AGAIN
4372  007450  012711  ;173450 012711          MOV     #RPPRST+RPGO,(R1) ;DO 'READ-IN PRESET' FUNCTION
4373  007452  000021  ;173452 000021
4374  007454  005061  ;173454 005061          CLR     RPDC(R1)        ;SET CYLINDER 0
4375  007456  000034  ;173456 000034
4376  007460  005061  ;173460 005061          CLR     RPDA(R1)        ; TRACK 0, SECTOR 0
4377  007462  000006  ;173462 000006
4378  007464  050261  ;173464 050261          BIS     R2,RPOF(R1)     ;SET INHIBIT ECC, 22-SECTOR FORMAT (IF FORMAT ERROR)
4379  007466  000032  ;173466 000032
4380  007470  012761  ;173470 012761          MOV     #-256.,RPWC(R1) ;SET UP WORD COUNT TO PROPER VALUE
4381  007472  177400  ;173472 177400
4382  007474  000002  ;173474 000002
4383                  ;                                       ;NOTE THAT IT IS NOT NECCESARY TO SET UP BUS
4384                  ;                                       ; ADDRESS, SINCE IT IS 0 AFTER READ-IN PRESET
4385  007476  012711  ;173476 012711          MOV     #RPREAD+RPGO,(R1) ;START READ FUNCTION
4386  007500  000071  ;173470 000071
4387                  ;173502           20$:
4388  007502  105711  ;173502 105711          TSTB    (R1)            ;READY?
4389  007504  100376  ;173504 100376          BPL     20$             ;NO-- WAIT UNTIL IT IS
4390  007506  032761  ;173506 032761          BIT     #RPFER,RPER1(R1) ;FORMAT ERROR?
4391  007510  000020  ;173510 000020
4392  007512  000014  ;173512 000014
4393  007514  001403  ;173514 001403          BEQ     30$             ;NO-- TRY AGAIN
4394  007516  052702  ;173516 052702          BIS     #RPFM22,R2      ;YES-- TRY FOR 22 SECTOR FLAVOR
4395  007520  010000  ;173520 010000
4396  007522  000740  ;173522 000740          BR      RPRTRY          ;TRY AGAIN
4397                  ;
4398                  ;173524           30$:
4399  007524  032711  ;173524 032711          BIT     #RPTRE!RPMCPE,(R1) ;TRANSFER OR MBC PARITY ERROR?
4400  007526  060000  ;173526 060000
4401  007530  001335  ;173530 001335          BNE     RPRTRY          ;YES-- ERROR-- TRY AGAIN
4402  007532  032761  ;173532 032761          BIT     #RPATA!RPERR,RPDS(R1) ;ATTN OR OTHER ERROR?
4403  007534  140000  ;173534 140000
4404  007536  000012  ;173536 000012
4405  007540  001331  ;173540 001331          BNE     RPRTRY          ;YES-- ERROR-- TRY AGAIN
4406                  ;               CLRPC:  BR      CLRPC           ;NO ERRORS-- GO TO LOCATION 0
4407                  ;
4408                  ; HERE TO GO TO 0
4409                  ;
4410                  ;173542           CLRPC:
4411  007542  005007  ;173542 005007          CLR     PC              ;JMP 0
4412
4413                  ; HERE ON ERROR FROM RP04 AFTER RETRYING-- DISPLAY DRIVE STATUS IN R0
4414                  ;
```

```
4415                    ;173544         RPEHLT:
4416  007544  016100    ;173544 016100          MOV     RPDS(R1),R0     ;DISPLAY DRIVE STATUS
4417  007546  000012    ;173546 000012
4418                                    ;       BR      HALTED          ;R.I.P.
4419                                    ;
4420                    ;173550         HALTED:
4421  007550  000000    ;173550 000000          HALT                    ;DIE
4422  007552  000776    ;173552 000776          BR      HALTED          ;STAY DEAD
```

```
4423          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 8
4424
4425          ;                                       .SBTTL  INTERNAL BUTTON #4 -- DUMP AND BOOTSTRAP THROUGH DTE20
4426                                                  ;
4427                                                  ; DTE20 DEFINITIONS
4428                                                  ;
4429                                                  ; NOTE THAT ALL DTE20 REGISTER DEFINITIONS AND BIT DEFINITIONS
4430                                                  ;  ARE IN $DEF IN SYSMAC.SML
4431                                                  ;
4432          ;                 000040  DTESIZ= 40                      ;EACH DTE OCCUPIES 20 WORDS IN EXTERNAL PAGE
4433          ;                 000004  DTEMAX= 4                       ;MAX OF 4 DTE'S ON A PDP-11
4434                                                  ;
4435                                                  ;
4436                                                  ; BUTTON #4 -- INITIATED BY '-10 RELOAD -11' BIT
4437                                                  ;
4438                        ;173554                   BUTON4:
4439  007554  010037  ;173554 010037                         MOV     R0,ROTOR7+0     ;SAVE R0 IN 40
4440  007556  000040  ;173556 000040
4441  007560  012700  ;173560 012700                         MOV     #10$,R0         ;SET RETURN ADDRESS IN R0
4442  007562  173566  ;173562 173566
4443  007564  000630  ;173564 000630                         BR      REGSAV          ;SAVE R1-R7
4444                                                  ;
4445                                                  ; REGISTERS SAVED-- LOOK FOR THE DTE20 WHICH PUSHED THE BUTTON
4446                                                  ;
4447                                                  ; THE DTE WHICH PUSHED THE BUTTON SHOULD HAVE THE DOORBELL
4448                                                  ;  RINGING AND HAVE THE VALUE 1365 (OCTAL) IN IT'S
4449                                                  ;  TO -10 BYTE COUNT TO10BC.
4450                                                  ;
4451                                                  ; NXM (TIME-OUT) TRAP IS USED TO SKIP NON-EXISTANT DTE20'S.
4452                                                  ;
4453                        ;173566                   10$:
4454  007566  005005  ;173566 005005                         CLR     R5              ;ADDRESS LOCATION ZERO
4455  007570  012500  ;173570 012500                         MOV     (R5)+,R0        ;SAVE 0 IN R0
4456  007572  012501  ;173572 012501                         MOV     (R5)+,R1        ;SAVE 2 IN R1
4457  007574  011502  ;173574 011502                         MOV     (R5),R2         ;SAVE 4 IN R2
4458  007576  012725  ;173576 012725                         MOV     #21$,(R5)+      ;SET NXM TRAP ADDRESS IN 4
4459  007600  173612  ;173570 173612
4460  007602  011503  ;173602 011503                         MOV     (R5),R3         ;SAVE 6 IN R3
4461  007604  005015  ;173604 005015                         CLR     (R5)            ;SET PS FOR TRAP
4462                                                  ;
4463                                                  ; LOOP THROUGH ALL DTE'S
4464                                                  ;
4465                        ;173606                   20$:
4466  007606  012704  ;173606 012704                         MOV     #DLYCNT-DTESIZ,R4 ;POINT TO DTE # -1'S DELAY COUNT REGISTER
4467  007610  174340  ;173610 174340
4468          ;                                                                   ; (WILL BUMP TO # 0)
4469                                                  ;
4470                                                  ; HERE ON NXM TRAP-- RESET SP AND TRY NEXT DTE
4471                                                  ;
4472                        ;173612                   21$:
4473  007612  012706  ;173612 012706                         MOV     #4,SP           ;SET SP TO 4, STACK IS LOCATIONS 2 AND 0
4474  007614  000004  ;173614 000004
4475                        ;173616                   22$:
4476  007616  062704  ;173616 062704                         ADD     #DTESIZ,R4      ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS
```

```
4477  007620  000040  ;173620 000040
4478  007622  105704  ;173622 105704          TSTB    R4                  ;IS THIS THE END OF THE DTE'S?
4479                  ;                                                   ; NOTE THAT THE LAST DTE IS AT 774540
4480                  ;                                                   ; AND THAT NOW R4= 774600 IF END
4481  007624  100770  ;173624 100770          BMI     20$                 ;YES-- START ALL OVER, UNTIL A DTE
4482                  ;                                                   ; SAYS HE PUSHED THE BUTTON
4483  007626  032764  ;173626 032764          BIT     #T011DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?
4484  007630  004000  ;173630 004000
4485  007632  000034  ;173632 000034
4486  007634  001770  ;173634 001770          BEQ     22$                 ;NO-- TRY NEXT DTE
```

```
4487          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 8-1
4488
4489  007636  026417  ;173636 026417          CMP     T010BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365
4490  007640  000014  ;173640 000014
4491                  ;                                                     ; IN IT'S TO -10 BYTE COUNT?
4492  007642  001365  ;173642 001365          BNE     22$             ;NO-- TRY ANOTHER DTE
4493                                        ;
4494                                        ; WE HAVE FOUND THE DTE WHICH PUSHED THE BUTTON
4495                                        ;
4496                                        ; ADDRESS OF DLYCNT REGISTER IS IN R4
4497                                        ;
4498  007644  010315  ;173644 010315          MOV     R3,(R5)         ;RESTORE LOCATION 6
4499  007646  010245  ;173646 010245          MOV     R2,-(R5)        ; 4
4500  007650  010145  ;173650 010145          MOV     R1,-(R5)        ; 2
4501  007652  010045  ;173652 010045          MOV     R0,-(R5)        ; 0
4502                                        ;
4503                                        ; SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
4504                                        ;  IN LOCATIONS 130-156
4505                                        ;
4506  007654  012700  ;173654 012700          MOV     #DTESAV,R0      ;POINT TO SAVE AREA
4507  007656  000130  ;173656 000130
4508                  ;173660         29$:
4509  007660  012420  ;173660 012420          MOV     (R4)+,(R0)+     ;SAVE A REGISTER
4510  007662  022700  ;173662 022700          CMP     #T011DT-DLYCNT+DTESAV,R0 ;FINISHED?
4511  007664  000156  ;173664 000156
4512  007666  103374  ;173666 103374          BHIS    29$             ;NO-- SAVE SOME MORE
4513                                        ;
4514                                        ; R4= T011DT+2
4515                                        ;
4516                                        ; SET R1= STATUS REGISTER
4517                                        ;     R4= DIAG2 REGISTER
4518                                        ;
4519                                        ; DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
4520                                        ;  LOADED FLAG
4521                                        ;
4522                  ;173670                 ADDX    DIAG2-T011DT-2,R4 ;R4 POINTS TO DIAG2 REGISTER
4523  007670  005724  ;173670 005724          TST     (R4)+
4524  007672  010401  ;173672 010401          MOV     R4,R1           ; SO DOES R1
4525  007674  012700  ;173674 012700          MOV     #DRESET,R0      ;SETUP R0 FOR 'DIAGNOSTIC RESET'
4526  007676  000100  ;173676 000100
4527  007700  010021  ;173700 010021          MOV     R0,(R1)+        ;R1 POINTS TO STATUS REGISTER
4528                                        ;
4529                                        ; REGISTERS:
4530                                        ;       R0 --   DRESET (DIAGNOSTIC RESET FUNCTION)
4531                                        ;       R1 --   STAT (STATUS REGISTER)
4532                                        ;       R4 --   DIAG2 (DIAGNOSTIC REGISTER #2, WHERE DRESET IS)
4533                                        ;
4534                                        ; THE -10 WILL NOW START READING -11 MEMORY, AS SOON AS WE SET
4535                                        ;  THE TO -10 ADDRESS.  WHEN FINISHED, THE -10 WILL RING OUR DOORBELL.
4536                                        ;
4537  007702  005061  ;173702 005061          CLR     DLYCNT-STAT(R1) ;SET DTE20 FOR MAXIMUM DELAY (ZERO)
4538  007704  177744  ;173704 177744
4539  007706  005061  ;173706 005061          CLR     T010AD-STAT(R1) ;START DUMPING -11 MEMORY TO -10
4540  007710  177764  ;173710 177764
```

```
4541                    ;                                                       ; STARTING AT LOCATION 0
4542                            ;173712         30$:
4543  007712  032711    ;173712 032711                  BIT     #T011DB,(R1)    ;IS DOORBELL RINGING (TRANSFER COMPLETE)?
4544  007714  004000    ;173714 004000
4545  007716  001775    ;173716 001775                  BEQ     30$             ;NO-- WAIT FOR DOORBELL
4546  007720  010014    ;173720 010014                  MOV     R0,(R4)         ;YES-- CLEAR DOORBELL AND ERROR FLAGS
4547                                            ;
4548                                            ; NOW THE -10 WILL GIVE US A 256 WORD BOOTSTRAP TO BE READ
4549                                            ;  INTO -11 MEMORY STARTING AT LOCATION 0.  WHEN FINISHED,
```

```
4550          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 8-2
4551
4552                                  ;  THE -10 WILL RING OUR DOORBELL, AND WE WILL START EXECUTION
4553                                  ;  OF THE LOADED CODE AT LOCATION 0.
4554                                  ;
4555  007722  005061  ;173722 005061          CLR     TO11AD-STAT(R1) ;START INPUT TO LOCATION 0
4556  007724  177766  ;173724 177766
4557  007726  012761  ;173726 012761          MOV     #IFLOP!<<-256.>&7777>,TO11BC-STAT(R1) ;256 WORDS, INTERRUPT
4558  007730  107400  ;173730 107400
4559  007732  177762  ;173732 177762
4560          ;                                                                       ; -10 WHEN DONE
4561                  ;173734         40$:
4562  007734  032711  ;173734 032711          BIT     #TO11DN,(R1)    ;TRANSFER COMPLETE?
4563  007736  000200  ;173736 000200
4564  007740  001775  ;173740 001775          BEQ     40$             ;NO-- WAIT UNTIL DONE
4565  007742  005007  ;173742 005007          CLR     PC              ;GO TO LOADED CODE, STARTING AT
4566          ;                                                       ; LOCATION 0
```

```
4567          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 9
4568
4569          ;                                   .SBTTL  FILL TO END OF ROM
4570                                        ;
4571                                        ;
4572                        ;173744                 FILLTO  1000
4573  007744  000000        ;173744     000         .BYTE   0
4574                        ;173745     000         .BYTE   0
4575  007746  000000        ;173746     000         .BYTE   0
4576                        ;173747     000         .BYTE   0
4577  007750  000000        ;173750     000         .BYTE   0
4578                        ;173751     000         .BYTE   0
4579  007752  000000        ;173752     000         .BYTE   0
4580                        ;173753     000         .BYTE   0
4581  007754  000000        ;173754     000         .BYTE   0
4582                        ;173755     000         .BYTE   0
4583  007756  000000        ;173756     000         .BYTE   0
4584                        ;173757     000         .BYTE   0
4585  007760  000000        ;173760     000         .BYTE   0
4586                        ;173761     000         .BYTE   0
4587  007762  000000        ;173762     000         .BYTE   0
4588                        ;173763     000         .BYTE   0
4589  007764  000000        ;173764     000         .BYTE   0
4590                        ;173765     000         .BYTE   0
4591  007766  000000        ;173766     000         .BYTE   0
4592                        ;173767     000         .BYTE   0
4593  007770  000000        ;173770     000         .BYTE   0
4594                        ;173771     000         .BYTE   0
4595  007772  000000        ;173772     000         .BYTE   0
4596                        ;173773     000         .BYTE   0
4597  007774  000000        ;173774     000         .BYTE   0
4598                        ;173775     000         .BYTE   0
4599  007776  END.YH:
4600  007776  000000        ;173776     000         .BYTE   0
4601                        ;173777     000         .BYTE   0
4602                                        ;
4603                                        ;
4604                                        ;
4605                        ;174000                 PASS2:
4606          ;                     000001          .END
```

```
  4607  010000  000177                MAP.Y.: .BLKW   127.
  4608  010376  000001                END.Y.: .BLKW   1
  4609  010400  000177                MAP.YX: .BLKW   127.
  4610  010776  000001                END.YX: .BLKW   1
```

```
                                ;*************************************************************
                                ;       INITIALIZATION AND START UP OF PROGRAM.
                                ;*************************************************************

                                RESTRT:
011000                                  MOV     #$CMTAG,R6              ;FIRST LOCATION TO BE CLEARED
011000  012706  001100                  CLR     (R6)+                   ;CLEAR MEMORY LOCATION
011004  005026                          CMP     #$TKS,R6                ;DONE?
011006  022706  001136                  BNE     .-6                     ;LOOP BACK IF NO
011012  001374                          MOV     #STACK,SP               ;SETUP THE STACK POINTER
011014  012706  001100                  MOV     #$SCOPE,@#IOTVEC        ;IOT VECTOR FOR SCOPE ROUTINE
011020  012737  017372  000020          MOV     #340,@#IOTVEC+2         ;LEVEL 7
011026  012737  000340  000022          MOV     #$ERROR,@#EMTVEC        ;EMT VECTOR FOR ERROR ROUTINE
011034  012737  017466  000030          MOV     #340,@#EMTVEC+2         ;LEVEL 7
011042  012737  000340  000032          MOV     #$TRAP,@#TRAPVEC        ;TRAP VECTOR FOR TRAP CALLS
011050  012737  020016  000034          MOV     #340,@#TRAPVEC+2        ;LEVEL 7
011056  012737  000340  000036          MOV     #.,$LPADR               ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
011064  012767  011064  170014          CLR     INITFG          ;INITIALIZE TO ASK WHICH TYPE
011072  005067  001246                  CLR     TABLE           ;INITIALIZE TO ASK WHICH TYPE
011076  005067  001232          START:  MOV     #STACK,SP       ;SET THE STACK POINTER
011102  012706  001100                  CLR     LSTERR          ;CLEAR ERROR FLG REPORT
011106  005067  170104                  CLR     @#0             ;SET FOR UNEXPECTED TRAP TO ADD 0
011112  005037  000000                  MOV     #START,PRG.NO   ;GET READY FOR PWR FAIL BEFORE FIRST TEST.
011116  012767  011102  005404          MOV     #6,@#4          ;SET TIME OUT TRAP VECTOR
011124  012737  000006  000004          CLR     @#6             ;SET TIME OUT STATUS TO 0
011132  005037  000006                  CLR     FLAG4           ;CLEAR TEST 4 INITIAL FLAG
011136  005067  004710                  TST     @#42            ;AM I RUNNING UNDER ACT-11??
011142  005737  000042                  BNE     .+6             ;BR IF *WE ARE* UNDER ACT-11!!
011146  001002                          JMP     CONT            ;JUMP IF NOT ACT-11
011150  000167  000522                  MOV     @#4,-(SP)       ;SAV TRAP POINTER
011154  013746  000004                  MOV     #NOROM,@#4      ;PUT IN A NEW ONE
011160  012737  012350  000004          TST     @#173000        ;TRY TO READ THE ROM
011166  005737  173000                  NOP                     ;WAIT FOR POSSIBLE TRAP
011172  000240                          MOV     (SP)+,@#4       ;IF NO TRAP RESTORE POINTER
011174  012637  000004                  CMP     MAP.YA,@#173000 ;DOES 1ST WORD COMPARE?
011200  026737  170174  173000          BNE     64$             ;CHECK NEXT MAP
011206  001034                          MOV     @#4,-(SP)       ;SAVE LOC 4
011210  013746  000004                  MOV     #65$,@#4        ;SET FOR TIMEOUT
011214  012737  011236  000004          TST     @#173400        ;READ FROM 173400
011222  005737  173400                  NOP                     ;IF NO TIMEOUT, NOT YA
011226  000240                          MOV     (SP)+,@#4       ;RESTORE LOC 4
011230  012637  000004                  BR      64$
011234  000421                  65$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
011236  022626                          MOV     (SP)+,@#4       ;RESTORE LOC 4
011240  012637  000004                  MOV     #MAP.YA,TABLE   ;1ST MAP ADDR
011244  012767  001400  001062          MOV     #END.YA,ALLEND  ;LAST MAP ADDR
011252  012767  001776  001056          MOV     #173376,LASTA   ;LAST ROM ADDR
011260  012767  173376  001060          MOV     #000101,VERSON  ;SET ROM TYPE
011266  012767  000101  005356          JMP     PRG1            ;START TEST 1
011274  000167  001266
011300                          64$:
011300  026737  170474  173000          CMP     MAP.YB,@#173000 ;DOES 1ST WORD COMPARE?
011306  001016                          BNE     69$             ;CHECK NEXT MAP
011310  012767  002000  001016          MOV     #MAP.YB,TABLE   ;1ST MAP ADDR
011316  012767  002776  001012          MOV     #END.YB,ALLEND  ;LAST MAP ADDR
```

```
 4665  011324  012767  173776  001014          MOV     #173776,LASTA   ;LAST ROM ADDR
 4666  011332  012767  000102  005312          MOV     #000102,VERSON  ;SET ROM TYPE
 4667  011340  000167  001222                  JMP     PRG1            ;START TEST 1
 4668  011344                          69$:
 4669  011344  026737  171430  173000          CMP     MAP.YC,@#173000 ;DOES 1ST WORD COMPARE?
 4670  011352  001036                          BNE     74$             ;CHECK NEXT MAP
 4671  011354  013746  000004                  MOV     @#4,-(SP)       ;SAVE LOC 4
 4672  011360  012737  011404  000004          MOV     #76$,@#4        ;SET FOR TIMEOUT
 4673  011366  026737  172006  173400          CMP     MAP.YC+400,@#173400     ;IS IT YC?
 4674  011374  001004                          BNE     77$             ;BR IF NOT YC
 4675  011376  012637  000004                  MOV     (SP)+,@#4       ;RESTORE LOC 4
 4676  011402  000404                          BR      78$             ;YES IT IS A YC
 4677  011404  022626                  76$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
 4678  011406  012637  000004          77$:    MOV     (SP)+,@#4       ;RESTORE LOC 4
 4679  011412  000416                          BR      74$             ;CHECK NEXT MAP
 4680  011414                          78$:
 4681  011414  012767  003000  000712          MOV     #MAP.YC,TABLE   ;1ST MAP ADDR
 4682  011422  012767  003776  000706          MOV     #END.YC,ALLEND  ;LAST MAP ADDR
 4683  011430  012767  173776  000710          MOV     #173776,LASTA   ;LAST ROM ADDR
 4684  011436  012767  000103  005206          MOV     #000103,VERSON  ;SET ROM TYPE
 4685  011444  000167  001116                  JMP     PRG1            ;START TEST 1
 4686  011450                          74$:
 4687  011450  026737  172324  173000          CMP     MAP.YD,@#173000 ;DOES 1ST WORD COMPARE?
 4688  011456  001016                          BNE     79$             ;CHECK NEXT MAP
 4689  011460  012767  004000  000646          MOV     #MAP.YD,TABLE   ;1ST MAP ADDR
 4690  011466  012767  004776  000642          MOV     #END.YD,ALLEND  ;LAST MAP ADDR
 4691  011474  012767  173776  000644          MOV     #173776,LASTA   ;LAST ROM ADDR
 4692  011502  012767  000104  005142          MOV     #000104,VERSON  ;SET ROM TYPE
 4693  011510  000167  001052                  JMP     PRG1            ;START TEST 1
 4694  011514                          79$:
 4695  011514  026737  173260  173000          CMP     MAP.YF,@#173000 ;DOES 1ST WORD COMPARE?
 4696  011522  001016                          BNE     84$             ;CHECK NEXT MAP
 4697  011524  012767  005000  000602          MOV     #MAP.YF,TABLE   ;1ST MAP ADDR
 4698  011532  012767  005776  000576          MOV     #END.YF,ALLEND  ;LAST MAP ADDR
 4699  011540  012767  173776  000600          MOV     #173776,LASTA   ;LAST ROM ADDR
 4700  011546  012767  000106  005076          MOV     #000106,VERSON  ;SET ROM TYPE
 4701  011554  000167  001006                  JMP     PRG1            ;START TEST 1
 4702  011560                          84$:
 4703  011560  026737  174214  173000          CMP     MAP.YG,@#173000 ;DOES 1ST WORD COMPARE?
 4704  011566  001016                          BNE     89$             ;CHECK NEXT MAP
 4705  011570  012767  006000  000536          MOV     #MAP.YG,TABLE   ;1ST MAP ADDR
 4706  011576  012767  006776  000532          MOV     #END.YG,ALLEND  ;LAST MAP ADDR
 4707  011604  012767  173776  000534          MOV     #173776,LASTA   ;LAST ROM ADDR
 4708  011612  012767  000107  005032          MOV     #000107,VERSON  ;SET ROM TYPE
 4709  011620  000167  000742                  JMP     PRG1            ;START TEST 1
 4710  011624                          89$:
 4711  011624  026737  175150  173000          CMP     MAP.YH,@#173000 ;DOES 1ST WORD COMPARE?
 4712  011632  001016                          BNE     94$             ;CHECK NEXT MAP
 4713  011634  012767  007000  000472          MOV     #MAP.YH,TABLE   ;1ST MAP ADDR
 4714  011642  012767  007776  000466          MOV     #END.YH,ALLEND  ;LAST MAP ADDR
 4715  011650  012767  173776  000470          MOV     #173776,LASTA   ;LAST ROM ADDR
 4716  011656  012767  000110  004766          MOV     #000110,VERSON  ;SET ROM TYPE
 4717  011664  000167  000676                  JMP     PRG1            ;START TEST 1
 4718  011670                          94$:
```

```
4719  011670  104400  012450                    TYPE    ,NMATCH         ;NOT BM873YA OR B OR C OR D OR F OR G OR H
4720  011674  000000                            HALT
4721  011676  005767  000442          CONT:     TST     INITFG          ;IS THIS THE FIRST TIME START UP?
4722  011702  001157                            BNE     3$              ;BR IF NOT FIRST TIME HERE.
4723  011704  005167  000434                    COM     INITFG          ;SET THE FLAG
4724  011710  104400  013160          2$:       TYPE    ,BM873X         ;TYPE THE QUESTION.
4725  011714  104412                            RDLIN
4726  011716  012602                            MOV     (SP)+,R2
4727  011720  011202                            MOV     (R2),R2         ;PLACE CHARACTER INTO R2.
4728  011722  022702  000052                    CMP     #52,R2          ;WAS * HIT??
4729  011726  001011                            BNE     64$             ;BR IF NO
4730  011730  012767  010000  000376            MOV     #MAP.Y.,TABLE   ;SET FOR START OF TABLE
4731  011736  012767  010376  000372            MOV     #END.Y.,ALLEND  ;SET END OF TABLE
4732  011744  012767  173376  000374            MOV     #173376,LASTA   ;SET LAST ROM ADDR
4733  011752                          64$:
4734  011752  022702  000101                    CMP     #101,R2         ;WAS A HIT??
4735  011756  001011                            BNE     65$             ;BR IF NO
4736  011760  012767  001400  000346            MOV     #MAP.YA,TABLE   ;SET FOR START OF TABLE
4737  011766  012767  001776  000342            MOV     #END.YA,ALLEND  ;SET END OF TABLE
4738  011774  012767  173376  000344            MOV     #173376,LASTA   ;SET LAST ROM ADDR
4739  012002                          65$:
4740  012002  022702  000102                    CMP     #102,R2         ;WAS B HIT??
4741  012006  001011                            BNE     66$             ;BR IF NO
4742  012010  012767  002000  000316            MOV     #MAP.YB,TABLE   ;SET FOR START OF TABLE
4743  012016  012767  002776  000312            MOV     #END.YB,ALLEND  ;SET END OF TABLE
4744  012024  012767  173776  000314            MOV     #173776,LASTA   ;SET LAST ROM ADDR
4745  012032                          66$:
4746  012032  022702  000103                    CMP     #103,R2         ;WAS C HIT??
4747  012036  001011                            BNE     67$             ;BR IF NO
4748  012040  012767  003000  000266            MOV     #MAP.YC,TABLE   ;SET FOR START OF TABLE
4749  012046  012767  003776  000262            MOV     #END.YC,ALLEND  ;SET END OF TABLE
4750  012054  012767  173776  000264            MOV     #173776,LASTA   ;SET LAST ROM ADDR
4751  012062                          67$:
4752  012062  022702  000104                    CMP     #104,R2         ;WAS D HIT??
4753  012066  001011                            BNE     68$             ;BR IF NO
4754  012070  012767  004000  000236            MOV     #MAP.YD,TABLE   ;SET FOR START OF TABLE
4755  012076  012767  004776  000232            MOV     #END.YD,ALLEND  ;SET END OF TABLE
4756  012104  012767  173776  000234            MOV     #173776,LASTA   ;SET LAST ROM ADDR
4757  012112                          68$:
4758  012112  022702  000106                    CMP     #106,R2         ;WAS F HIT??
4759  012116  001011                            BNE     69$             ;BR IF NO
4760  012120  012767  005000  000206            MOV     #MAP.YF,TABLE   ;SET FOR START OF TABLE
4761  012126  012767  005776  000202            MOV     #END.YF,ALLEND  ;SET END OF TABLE
4762  012134  012767  173776  000204            MOV     #173776,LASTA   ;SET LAST ROM ADDR
4763  012142                          69$:
4764  012142  022702  000107                    CMP     #107,R2         ;WAS G HIT??
4765  012146  001011                            BNE     70$             ;BR IF NO
4766  012150  012767  006000  000156            MOV     #MAP.YG,TABLE   ;SET FOR START OF TABLE
4767  012156  012767  006776  000152            MOV     #END.YG,ALLEND  ;SET END OF TABLE
4768  012164  012767  173776  000154            MOV     #173776,LASTA   ;SET LAST ROM ADDR
4769  012172                          70$:
4770  012172  022702  000110                    CMP     #110,R2         ;WAS H HIT??
4771  012176  001011                            BNE     71$             ;BR IF NO
4772  012200  012767  007000  000126            MOV     #MAP.YH,TABLE   ;SET FOR START OF TABLE
```

```
4773  012206  012767  007776  000122          MOV     #END.YH,ALLEND  ;SET END OF TABLE
4774  012214  012767  173776  000124          MOV     #173776,LASTA   ;SET LAST ROM ADDR
4775  012222                          71$:
4776  012222  010267  004424                  MOV     R2,VERSON       ;STORE VERSION TYPE..
4777  012226  005767  000102                  TST     TABLE           ;HAS A MAP BEEN SELECTED?
4778  012232  001003                          BNE     3$              ;BR IF OK...
4779  012234  104400  013235                  TYPE    ,BM.ERR         ;TYPE ERROR
4780  012240  000623                          BR      2$              ;GO AND GET CORRECT MAP.
4781  012242  104400  016174          3$:     TYPE    ,MSG3           ;TYPE MESSAGE FOR TEST NUMBER
4782  012246  104412                  X.X.:   RDLIN
4783  012250  012602                          MOV     (SP)+,R2
4784  012252  011203                          MOV     (R2),R3         ;MOV THE CHAR TO R3
4785  012254  022703  000061          2$:     CMP     #61,R3          ;WAS 1 HIT??
4786  012260  001002                          BNE     4$              ;BR IF NO
4787  012262  000167  000300                  JMP     PRG1            ;GOTO PRG 1
4788  012266  022703  000062          4$:     CMP     #62,R3          ;WAS 2 HIT??
4789  012272  001002                          BNE     5$              ;BR IF NO
4790  012274  000167  001012                  JMP     PRG2            ;GOTO PRG 2
4791  012300  022703  000063          5$:     CMP     #63,R3          ;WAS 3 HIT??
4792  012304  001002                          BNE     6$              ;BR IF NO
4793  012306  000167  001740                  JMP     PRG3            ;GOTO PRG3
4794  012312  022703  000064          6$:     CMP     #64,R3          ;WAS 4 HIT??
4795  012316  001002                          BNE     3$              ;BR IF NO
4796  012320  000167  002732                  JMP     PRG4            ;GOTO PRG 4
4797  012324  104400  016400          3$:     TYPE    ,M.QM           ;NEITHER 1 OR 2 OR 3 OR 4 WAS HIT
4798  012330  000167  176444                  JMP     RESTRT          ;TYPE "??" GO TO THE BEGINING.
4799  012334  000000                  TABLE:  0
4800  012336  000000                  ALLEND: 0
4801  012340  010400                  EXTMAP: MAP.YX
4802  012342  010776                  EXTEND: END.YX
4803  012344  000000                  INITFG: 0
4804  012346  000000                  LASTA:  0
4805  012350  104400  012360          NOROM:  TYPE    ,NOROMS ;TYPE   CAN'T FIND A RESPONSE
4806  012354  000000                          HALT            ;NO LOADER INSTALLED?
4807  012356  000776                          BR      .-2
4808  012360  005015  051124  050101  NOROMS: .ASCII  <15><12>/TRAP TO 4 ON 1ST READ OF 173000/
      012421     015  044412  020123          .ASCIZ  <15><12>/IS LOADER INSTALLED?/
      012450  005015  040503  023516  NMATCH: .ASCII  <15><12>/CAN'T IDENTIFY LOADER AS YA,YB,YC,YD,YF,YG OR YH AFTER/
      012540  005015  046503  020120          .ASCIZ  <15><12>/CMP WITH LOC 173000/
                                      .EVEN
```

```
                                        ;PROGRAM 1
                                        ;THE PURPOSE OF PROGRAM 1 IS TO READ THE ROM AND
                                        ;VERIFY THAT THE DATA IS CORRECT. ALL ADDRESSES
                                        ;ARE READ, EXCECPT THE TRAP VECTOR, FIVE TIMES.
                                        ;
                                        ;THE SECOND PART OF THIS TEST VERIFIES THAT TRYING
                                        ;TO WRITE THE ROM RESUTS IN A TIME OUT TRAP.
                                        ;ALL ADDRESS ARE WRITTEN WITH A -1
                                        ;,AND ARE EXPECTED TO TRAP.

012566  012767  012566  003734  PRG1:   MOV     #PRG1,PRG.NO    ;SET FOR PWR FAIL
012574  012767  000500  166416          MOV     #500,ICOUNT     ;DO THIS TEST 500(8) TIMES.
012602  012737  016532  000004  PRG.1:  MOV     #NO.TRAP,@#4    ;SET FOR UNEXPECTED TRAP.
012610  012700  173000                  MOV     #173000,R0      ;SET BEGGINING ADDRESS
012614  012767  012640  166264          MOV     #2$,$LPADR      ;IF SW14=1; GOTO 2$ WHEN SCOPE IS HIT
012622  016704  177506                  MOV     TABLE,R4        ;SET START OF MAP
012626  016767  177514  000322          MOV     LASTA,LAST      ;SET LAST ADDRESS
012634  012703  000005          1$:     MOV     #5.,R3          ;DO EACH ADDRESS 5 TIMES.
012640  022700  173024          2$:     CMP     #173024,R0      ;DON'T DO THE VECTOR ADD.
012644  001001                          BNE     20$             ;BR IF NOT THE VECTOR ADD.
012646  022024                          CMP     (R0)+,(R4)+     ;UPDATE TO NEXT ADDRESS
012650  022700  173224          20$:    CMP     #173224,R0      ;DON'T DO THE TRAP VECTORS
012654  001001                          BNE     21$             ;NO THIS ISN'T A TRAP VECTOR.
012656  022024                          CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS..
012660  010467  166240          21$:    MOV     R4,$GDDAT
012664  010067  166236                  MOV     R0,$BDDAT
012670  011067  166332                  MOV     (R0),TEMP4      ;READ THE ADDRESS
012674  011467  166324                  MOV     (R4),TEMP3      ;READ THE SOFTWARE ADDRESS
012700  026767  166320  166320          CMP     TEMP3,TEMP4
012706  001401                          BEQ     22$             ;BR IF GOOD
012710  104001                          ERROR   1               ;INCORRECT COMPARISON.
012712  032767  004000  164650  22$:    BIT     #BIT11,SWR      ;QUICK PASS.?
012720  001002                          BNE     23$             ;BR IF YES
012722  005303                          DEC     R3              ;HAS THAT ADD BEEN READ 5 TIMES?
012724  001345                          BNE     2$              ;BR IF NOT 5 TIMES

012726  026700  000224          23$:    CMP     LAST,R0         ;WAS LAST ADDRESS CHECKED?
012732  001403                          BEQ     10$             ;BR IF YES
012734  000004                          SCOPE                   ;LOCK ON THIS ADDRESS IF SW14=1
012736  022024                          CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS.
012740  000735                          BR      1$              ;CONTINUE THE TEST.

012742  032767  000001  164620  10$:    BIT     #BIT0,SWR       ;EXTENDED WORD TO BE CHECKED?
012750  001413                          BEQ     3$              ;BR IF NO CHECKING.
012752  022767  173776  000176          CMP     #173776,LAST    ;IS ALL THE TEST DONE?
012760  001407                          BEQ     3$              ;BR IF YES.
012762  012767  173776  000166          MOV     #173776,LAST    ;SET LAST ADDRESS.
012770  016704  177344                  MOV     EXTMAP,R4       ;SET EXTENDED MAP.
012774  005720                          TST     (R0)+           ;POP POINTER
012776  000716                          BR      1$              ;GO DO THE TEST.
```

```
                                        ;TEST THAT WRITTING ROM RESULTS IN A TIME OUT
                                        ;TRAP.
                                
013000  012767  013024  166100  3$:     MOV     #5$,$LPADR      ;IF SW14=1 GOTO 5$ WHEN SCOPE IS HIT
013006  012700  173000                  MOV     #173000,R0      ;SET R0 WITH BASE ADDRESS OF ROM
013012  012737  013060  000004          MOV     #6$,@#4         ;SET FOR TIME OUT TRAP
013020  012703  000005          4$:     MOV     #5.,R3          ;DO EACH ADD 5 TIMES
013024  022700  173024          5$:     CMP     #173024,R0      ;CHECK FOR A TRAP VECTOR
013030  001001                          BNE     24$             ;BR IF NOT VECTOR
013032  005720                          TST     (R0)+           ;UPDATE THE REGISTER POINTER
013034  022700  173224          24$:    CMP     #173224,R0      ;CHECK FOR THE OTHER VECTOR
013040  001001                          BNE     25$             ;BR IF NOT THE VECTOR
013042  005720                          TST     (R0)+           ;UPDATE THE POINTER
013044  012710  177777          25$:    MOV     #-1,(R0)        ;WRITE ROM WITH A -1
013050  000240                          NOP                     ;WAIT ONE INSTR. TIME
013052  010067  166150                  MOV     R0,TEMP4
013056  104002                          ERROR   2               ;WRITING ROM DIDN'T TIME OUT.
013060  012706  001100          6$:     MOV     #STACK,SP       ;RESTORE STACK
013064  032767  004000  164476          BIT     #BIT11,SWR      ;QUICK PASS?
013072  001002                          BNE     30$
013074  005303                          DEC     R3              ;DO EACH ADD 5 TIMES
013076  001352                          BNE     5$              ;NOT DONE WITH THIS ONE YET.

013100  032767  000001  164462  30$:     BIT     #BIT0,SWR       ;EXTENDED 128. WORDS TO BE CHECKED?
013106  001404                          BEQ     31$             ;BR IF NO
013110  022700  173776                  CMP     #173776,R0      ;HAVE ALL 256. WORDS BEEN CHECKED?
013114  001407                          BEQ     7$              ;BR IF ALL DONE
013116  000403                          BR      32$             ;KEEP GOING
013120  026700  177222          31$:    CMP     LASTA,R0        ;ALL DONE??
013124  001403                          BEQ     7$              ;HAVE ALL 128. WORDS DONE?
013126  000004          32$:    SCOPE                           ;CHECK SW14 FOR FREEZE!!
013130  005720                          TST     (R0)+           ;UPDATE TO NEXT ADDRESS
013132  000732                          BR      4$              ;GO DO IT AGAIN
013134  005367  166060          7$:     DEC     ICOUNT          ;ITERATION COUNT DONE?
013140  001004                          BNE     8$              ;BR IF NOT DONE.
013142  004767  003424                  JSR     PC,EOP          ;TYPE END MESSAGE
013146  000167  177414                  JMP     PRG1            ;GO DO IT AGAIN.
013152  000167  177424          8$:     JMP     PRG.1           ;GO RESTART.
013156  000000          LAST:   0

013160  005015  040515  047111  BM873X: .ASCII  <15><12>/MAINDEC-11-DZBMDH/
013203     015  042012  053105          .ASCII  <15><12>/DEVICE VERSION/
013223     015  041012  034115          .ASCIZ  <15><12>/BM873-Y/
013235     015  025012  040454  BM.ERR: .ASCIZ  <15><12>/*,A,B,C,D,F,G,H ONLY./
013265     040  020040  042526  VERS:   .ASCIZ  /   VERSION: BM873-Y/
           013312                  .EVEN
```

```
                                        ;PROGRAM 2
                                        ;BLIND READ FROM ROM.
                                        ;THIS PROGRAM WILL DUMP THE CONTENTS OF THE ROM OUT
                                        ;PERFORMING NO CHECKING AT ALL.
                                        ;PLEASE NOTE: NO CHECKING IS DONE.

013312  012767  013312  003210  PRG2:   MOV     #PRG2,PRG.NO    ;SET FOR POWER FAIL
013320  012737  016532  000004          MOV     #NO.TRAP,@#4    ;SET FOR UNEXPECTED TRAP TO 4
013326  016767  177014  177622          MOV     LASTA,LAST
013334  062767  000002  177614          ADD     #2,LAST
013342  012700  173000          21$:    MOV     #173000,R0      ;SET R0 WITH THE STARTING ROM ADD.
013346  016703  176762                  MOV     TABLE,R3        ;SET POINTER.
013352  104400  013672                  TYPE    ,DH.2           ;TYPE MESSAGE
013356  104400  013754                  TYPE    ,DH.2B          ;TYPE THE HEADER
013362  012767  000007  165632  1$:     MOV     #7,TEMP5        ;SET COUNTER
013370  011001                          MOV     (R0),R1         ;READ THE ROM
013372  010067  165626                  MOV     R0,TEMP3        ;STORE R0
013376  010167  165624                  MOV     R1,TEMP4        ;STORE R1
013402  022767  010000  176724          CMP     #MAP.Y.,TABLE   ;IF BM873.Y* SELECTED; FILL TABLE
013410  001001                          BNE     22$             ;BR IF NOT BM873.Y*
013412  011023                          MOV     (R0),(R3)+      ;FILL THE TABLE..
013414  005720                  22$:    TST     (R0)+           ;POP THE POINTER
013416  104400  016414                  TYPE    ,MCRLF

013422  016746  165576                  MOV     TEMP3,-(SP)
013426  104402                          TYPOC
013430  104400  016405                  TYPE    ,MSPACE         ;TYPE THREE SPACES.


013434  016746  165566                  MOV     TEMP4,-(SP)
013440  104402                          TYPOC
013442  011001                  7$:     MOV     (R0),R1         ;STORE ROM DATA
013444  010067  165554                  MOV     R0,TEMP3        ;STORE ROM ADDRESS
013450  010167  165552                  MOV     R1,TEMP4        ;PREPARE DATA FOR TYPE OUT
013454  022767  010000  176652          CMP     #MAP.Y.,TABLE   ;IS BM873.Y* SELECTED?
013462  001001                          BNE     23$             ;BR IF NO..
013464  011023                          MOV     (R0),(R3)+      ;FILL THE DATA TABLE
013466  005720                  23$:    TST     (R0)+           ;POP THE POINTER

013470  104400  016405                  TYPE    ,MSPACE

013474  016746  165526                  MOV     TEMP4,-(SP)
013500  104402                          TYPOC

013502  026700  177450                  CMP     LAST,R0 ;HAS THE HIGHEST LIMIT BEEN HIT?
013506  001404                          BEQ     2$              ;BR IF ALL DONE.
013510  005367  165506                  DEC     TEMP5           ;DECREASE COUNTER
013514  001352                          BNE     7$              ;BR IF NOT 0; KEEP GOING
013516  000721                          BR      1$              ;GO TYPE ADDRESS NOW

013520  032767  000001  164042  2$:     BIT     #BIT0,SWR       ;IS THE EXTENDED 128. WORDS TO BE CHECKED??
013526  001455                          BEQ     3$              ;BR IF NO.
013530  012700  173400                  MOV     #173400,R0      ;RESET POINTER OF ROM
013534  016703  176600                  MOV     EXTMAP,R3       ;SET SOFTWARE MAP POINTER
```

```
4954  013540  104400  014066                    TYPE    ,DH.2A          ;TYPE NEW HEADER
4955  013544  104400  013754                    TYPE    ,DH.2B          ;TYPE ADDRESS AND +XX
4956  013550  012767  000007  165444  6$:       MOV     #7,TEMP5        ;SET TYPE OUT COUNTER
4957  013556  011001                            MOV     (R0),R1         ;READ THE ROM
4958  013560  010067  165440                    MOV     R0,TEMP3        ;STORE R0
4959  013564  010167  165436                    MOV     R1,TEMP4        ;STORE R1
4960  013570  012023                            MOV     (R0)+,(R3)+     ;STORE THE DATA IN SOFTWARE MAP
4961  013572  104400  016414                    TYPE    ,MCRLF
4962  013576  016746  165422                    MOV     TEMP3,-(SP)
4963  013602  104402                            TYPOC
4964
4965  013604  104400  016405                    TYPE    ,MSPACE
4966  013610  016746  165412                    MOV     TEMP4,-(SP)
4967  013614  104402                            TYPOC
4968
4969  013616  011001                  8$:       MOV     (R0),R1         ;SAVE THE ROM DATA
4970  013620  010067  165400                    MOV     R0,TEMP3        ;SAVE THE ROM ADDRESS
4971  013624  010167  165376                    MOV     R1,TEMP4        ;SET DATA FOR TYPE OUT
4972
4973  013630  104400  016405                    TYPE    ,MSPACE
4974
4975  013634  016746  165366                    MOV     TEMP4,-(SP)
4976  013640  104402                            TYPOC
4977
4978  013642  012023                            MOV     (R0)+,(R3)+     ;STORE THE DATA IN SOFTWARE TABLE
4979  013644  022700  174000                    CMP     #174000,R0      ;HAS THE HIGHEST LIMIT BEEN HIT?
4980  013650  001404                            BEQ     3$              ;BR IF ALL DONE.
4981  013652  005367  165344                    DEC     TEMP5           ;DEC TABLE COUNTER
4982  013656  001357                            BNE     8$              ;BR TO JUST TYPE DATA
4983  013660  000733                            BR      6$              ;BR TO TYPE ADDRESS
4984  013662  005000                  3$:       CLR     R0              ;CLEAR DATA LIGHTS
4985  013664  000000                            HALT                    ;HIT CONTINUE TO PROCEED.
4986  013666  000167  177420                    JMP     PRG2            ;GOTO PRG 2
4987  013672  006414  005012  016412  DH.2:     .ASCII  <14><15><12><12><12><35><37><177><177><177>/BLIND READ OF ROM/
      013725     015  006412  077577            .ASCIZ  <15><12><15><177><177>/NOTE: NO CHECKING/
      013754  005015  040412  042104  DH.2B:    .ASCII   <15><12><12>/ADDRESS ADD+00  ADD+02  ADD+04/
      014015     040  040440  042104            .ASCIZ /  ADD+06  ADD+10  ADD+12  ADD+14  ADD+16/
      014066  005015  042412  052130  DH.2A:    .ASCII  <15><12><12>/EXTENDED 128. WORD ROM DUMP./
      014125     015  041412  047117            .ASCII  <15><12>/CONTENTS DUMPED IS PLACED IN THE SOFTWARE/
      014200  005015  040515  027120            .ASCII  <15><12>/MAP. DATA SHOULD BE VISUALLY INSPECTED!/
              014252                  .EVEN
```

```
                                        ;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.
                                        ;

014252  012767  014252  002250  PRG3:   MOV     #PRG3,PRG.NO    ;SET FOR POWER FAIL
014260  016701  176050          XHOLD:  MOV     TABLE,R1        ;DEFAULT STARTING ADDRESS TO MAP
014264  010167  000764                  MOV     R1,ADDRESS      ;SAVE THE SOFTWARE ADDRESS
014270  104400  016307                  TYPE    ,MASTER         ;TYPE AN "*"
014274  104412                          RDLIN
014276  012602                          MOV     (SP)+,R2
014300  011202                          MOV     (R2),R2
014302  022702  000114                  CMP     #114,R2         ;WAS AN "L" (LIST) HIT?
014306  001464                          BEQ     SRV.L

014310  022702  000104          1$:     CMP     #104,R2         ;WAS A  "D" (DATA) HIT?
014314  001413                          BEQ     SRV.D
014316  022702  000122                  CMP     #122,R2         ;WAS AN "R" (RUN) HIT?
014322  001002                          BNE     10$
014324  000167  000342                  JMP     SRV.R
014330  022702  000101          10$:    CMP     #101,R2         ;WAS AN "A" (ADDRESS) HIT?
014334  001444                          BEQ     SRV.A
014336  104400  016400                  TYPE    ,M.QM           ;TYPE A "?"
014342  000752                          BR      XHOLD           ;NEITHER A "L","P","D","R","A",OR CR WAS HIT.

014344  016767  000704  164654  SRV.D:  MOV     ADDRESS,TEMP4   ;RESET ADDRESS POINTER.
014352  104400  016414                  TYPE    ,MCRLF
014356  016746  164644                  MOV     TEMP4,-(SP)
014362  016701  164640                  MOV     TEMP4,R1
014366  104402                          TYPOC

014370  104400  016405                  TYPE    ,MSPACE

014374  104414                          RDOCT
014376  012611                          MOV     (SP)+,(R1)      ;STORE DATA

014400  005721                          TST     (R1)+           ;UPDATE THE SOFTWARE ADDRESS
014402  026701  175734                  CMP     EXTEND,R1       ;IS THE LIMIT EXCEEDED
014406  103413                          BLO     7$              ;INPUT LIMIT EXCEEDED!! ERROR.
014410  010167  164612                  MOV     R1,TEMP4        ;SAVE THE ADDRESS.
014414  104400  016414                  TYPE    ,MCRLF
014420  016746  164602                  MOV     TEMP4,-(SP)
014424  104402                          TYPOC
```

```
5042  014426  010167  000622                  MOV     R1,ADDRESS      ;SAVE THE ADDRESS FOR GOOD
5043  014432  000167  177632                  JMP     XHOLD
5044  014436  104400  016400          7$:     TYPE    ,M.QM           ;TYPE A "?"
5045  014442  000167  177622                  JMP     XHOLD
5046
5047                                  ;YOU ARE HERE BECAUSE YOU HIT AN "A"
5048                                  ;YOU TOLD ME YOU WERE GOING TO INPUT AN ADDRESS.
5049                                  ;SO INPUT THE ADDRESS AND TERMINATE WITH A CARRAGE RETURN.
5050                                  ;OK??
5051
5052  014446  104414                  SRV.A:  RDOCT                   ;READ THE ADDRESS HE WANTS TO MODIFY.
5053  014450  012667  000600                  MOV     (SP)+,ADDRESS
5054  014454  000167  177610          4$:     JMP     XHOLD
5055
5056                                  ;YOU ENTERED HERE BECAUSE YOU HIT "L"
5057                                  ;YOU TOLD ME YOU WANTED A LISTING OF THE SOFTWARE MAP
5058                                  ;SO HERE IT IS.
5059
5060
5061  014460                          SRV.L:
5062  014460  016700  175650                  MOV     TABLE,R0        ;GET SOFTWARE MAP
5063  014464  016767  175646  000176          MOV     ALLEND,DEAD     ;SET DEAD END POINTER
5064  014472  104400  016226                  TYPE    ,MSG4           ;TYPE HEADER
5065  014476  104400  013754                  TYPE    ,DH.2B          ;TYPE ADDRESS ADD+XX
5066  014502  012767  000007  164512  1$:     MOV     #7,TEMP5        ;SET COUNTER FOR ACCROSS PAGE
5067  014510  011067  164512                  MOV     (R0),TEMP4      ;GET DATA
5068  014514  010067  164504                  MOV     R0,TEMP3        ;GET ADDRESS
5069  014520  005720                          TST     (R0)+           ;UPDATE ADDRESS POINTER
5070  014522  104400  016414                  TYPE    ,MCRLF
5071
5072  014526  016746  164472                  MOV     TEMP3,-(SP)
5073  014532  104402                          TYPOC
5074
5075  014534  104400  016405                  TYPE    ,MSPACE
5076
5077  014540  016746  164462                  MOV     TEMP4,-(SP)
5078  014544  104402                          TYPOC
5079
5080  014546  104400  016405                  TYPE    ,MSPACE
5081
5082  014552  011067  164450          2$:     MOV     (R0),TEMP4      ;GET DATA
5083  014556  010067  164442                  MOV     R0,TEMP3        ;GET ADDRESS
5084  014562  005720                          TST     (R0)+           ;UPDATE POINTER
5085
5086  014564  016746  164436                  MOV     TEMP4,-(SP)
5087  014570  104402                          TYPOC
5088  014572  104400  016405                  TYPE    ,MSPACE
5089
5090  014576  016703  000066          3$:     MOV     DEAD,R3
5091  014602  005723                          TST     (R3)+           ;UPDATE POINTER
5092  014604  020003                          CMP     R0,R3           ;LIMIT DONE ??
5093  014606  001404                          BEQ     5$              ;BR IF YES
5094  014610  005367  164406          4$:     DEC     TEMP5           ;DEC DATA COUNTER
5095  014614  001356                          BNE     2$              ;BR IF MORE DATA TO GO
```

```
5096  014616  000731                          BR      1$              ;TYPE THE ADDRESSS
5097  014620                          5$:
5098  014620  032767  000001  162742          BIT     #BIT0,SWR       ;EXTENDED SOFTWARE DUMP?
5099  014626  001416                          BEQ     6$              ;BR IF NO DUMP
5100  014630  005743                          TST     -(R3)           ;PUSH POINTER
5101  014632  026703  175504                  CMP     EXTEND,R3
5102  014636  001412                          BEQ     6$              ;BR IF ALL DONE
5103  014640  104400  016254                  TYPE    ,MSG5           ;TYPE EXTENDED MAP:
5104  014644  104400  013754                  TYPE    ,DH.2B
5105  014650  016700  175464                  MOV     EXTMAP,R0       ;SET POINTER
5106  014654  016767  175462  000006          MOV     EXTEND,DEAD     ;SET DEAD END POINTER
5107  014662  000707                          BR      1$              ;DO IT AGAIN SAM.
5108  014664  000167  177400          6$:     JMP     XHOLD
5109  014670  000000                  DEAD:   0
5110
5111                                  ;NOW YOU ARE HERE BECAUSE YOU WANT TO RUN THE PROGRAM
5112                                  ;REMEMBER NOW, YOU SET UP THE MAP.
5113                                  ;ARE YOU SURE YOU TYPED IN THE CORRECT DATA.???
5114                                  ;HERE WE GO
5115
5116  014672                          SRV.R:
5117  014672  012737  016532  000004  RUN3:   MOV     #NO.TRAP,@#4    ;GET READY FOR UNEXPECTED TRAP
5118  014700  012767  000500  164312          MOV     #500,ICOUNT     ;DO TEST 500(8) TIMES
5119  014706  012700  173000          RUN.3:  MOV     #173000,R0      ;SET BEGGINING ADDRESS
5120  014712  012767  014736  164166          MOV     #2$,$LPADR      ;IF SW14=1; GOTO 2$ WHEN I HIT "SCOPE"
5121  014720  016704  175410                  MOV     TABLE,R4        ;SET SOFTWARE RESUTS
5122  014724  016767  175416  176224          MOV     LASTA,LAST      ;SET LAST ADDRESS
5123  014732  012703  000005          1$:     MOV     #5.,R3          ;DO EACH ADDRESS 5 TIMES.
5124  014736  022700  173024          2$:     CMP     #173024,R0      ;DON'T DO THE VECTOR ADD.
5125  014742  001001                          BNE     30$             ;BR IF NOT THE VECTOR ADD.
5126  014744  022024                          CMP     (R0)+,(R4)+     ;UPDATE TO NEXT ADDRESS
5127  014746  022700  173224          30$:    CMP     #173224,R0      ;IS THIS THE SECOND TRAP VECTOR??
5128  014752  001001                          BNE     10$             ;BR IF NOT VECTOR
5129  014754  022024                          CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS !!
5130  014756  010467  164142          10$:    MOV     R4,$GDDAT
5131  014762  010067  164140                  MOV     R0,$BDDAT
5132  014766  011067  164234                  MOV     (R0),TEMP4      ;READ THE ADDRESS
5133  014772  011467  164226                  MOV     (R4),TEMP3      ;READ THE SOFTWARE ADDRESS
5134  014776  026767  164222  164222          CMP     TEMP3,TEMP4
5135  015004  001401                          BEQ     11$             ;BRANCH IF OK
5136  015006  104001                          ERROR   1               ;INCORRECT COMPARISON.
5137  015010  032767  004000  162552  11$:    BIT     #BIT11,SWR      ;QUICK PASS.
5138  015016  001002                          BNE     12$             ;BR IF YES
5139  015020  005303                          DEC     R3              ;HAS THAT ADD BEEN READ 10 TIMES?
5140  015022  001345                          BNE     2$              ;BR IF NOT 10 TIMES
5141  015024  026700  176126          12$:    CMP     LAST,R0         ;WAS LAST ADDRESS CHECKED?
5142  015030  001403                          BEQ     15$             ;BR IF YES
5143  015032  000004                          SCOPE                   ;LOCK ON THIS ADDRESS?
5144  015034  022024                          CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS.
5145  015036  000735                          BR      1$              ;CONTINUE THE TEST.
5146  015040  032767  000001  162522  15$:    BIT     #BIT0,SWR       ;EXTENDED WORD TO BE CHECKED?
5147  015046  001413                          BEQ     3$              ;BR IF NO CHECKING.
5148  015050  022767  173776  1.6100          CMP     #173776,LAST    ;IS ALL THE TEST DONE?
5149  015056  001407                          BEQ     3$              ;BR IF YES.
```

```
 5150  015060  012767  173776  176070          MOV     #173776,LAST    ;SET LAST ADDRESS.
 5151  015066  016704  175246                  MOV     EXTMAP,R4       ;SET EXTENDED MAP.
 5152  015072  005720                          TST     (R0)+           ;POP POINTER
 5153  015074  000716                          BR      1$              ;GO DO THE TEST.
```

```
                                        ;TEST THAT WRITING ROM RESULTS IN A TIME OUT
                                        ;TRAP.
                                        
015076  012700  173000          3$:     MOV     #173000,R0      ;SET BASE ADDRESS
015102  012767  015122  163776          MOV     #5$,$LPADR      ;IF SW14=1; GOTO 5$ AT SCOPE
015110  012737  015156  000004          MOV     #6$,@#4         ;TIME OUT TRAP; GOTO 6$
015116  012703  000012          4$:     MOV     #10.,R3         ;DO EACH ADD 10 TIMES
015122  022700  173024          5$:     CMP     #173024,R0      ;IS THIS AT THE TRAP VECTOR
015126  001001                          BNE     20$             ;BR IF NO
015130  005720                          TST     (R0)+           ;UPDATE POINTER
015132  022700  173224          20$:    CMP     #173224,R0      ;IS THIS AT THE SECOND TRAP VECTOR
015136  001001                          BNE     21$             ;BR IF NO
015140  005720                          TST     (R0)+           ;UPDATE THE POINTER
015142  012710  177777          21$:    MOV     #-1,(R0)        ;WRITE ROM WITH A -1
015146  000240                          NOP                     ;WAIT ONE INSTR. TIME
015150  010067  164052                  MOV     R0,TEMP4
015154  104002                          ERROR   2               ;WRITING ROM DIDN'T TIME OUT.
015156  012706  001100          6$:     MOV     #STACK,SP       ;RESTORE STACK
015162  032767  004000  162400          BIT     #BIT11,SWR      ;QUICK PASS?
015170  001002                          BNE     22$             ;BR IF YES
015172  005303                          DEC     R3              ;DO EACH ADD 10 TIMES
015174  001352                          BNE     5$              ;NOT DONE WITH THIS ONE YET.
015176  032767  000001  162364  22$:    BIT     #BIT0,SWR       ;IS THE EXTENDED 128. WORDS TO BE TESTED??
015204  001404                          BEQ     23$             ;BR IF NO
015206  022700  173776                  CMP     #173776,R0      ;IS THE EXTENDED LIMIT BEEN TESTED?
015212  001407                          BEQ     7$              ;IF YES; GOTO 7$
015214  000403                          BR      24$             ;IF NO; KEEP GOING.
015216  026700  175124          23$:    CMP     LASTA,R0        ;ALL DONE??
015222  001403                          BEQ     7$              ;IF YES; GOTO 7$
015224  000004                  24$:    SCOPE                   ;GO CHECK SW14; (FREEZE !!)
015226  005720                          TST     (R0)+           ;UPDATE TO NEXT ADDRESS
015230  000732                          BR      4$              ;GO DO IT AGAIN
015232  005367  163762          7$:     DEC     ICOUNT          ;CHECK ITERATION COUNT
015236  001004                          BNE     8$              ;MORE TO GO
015240  004767  001326                  JSR     PC,EOP          ;GO TO END OF PASS ROUTINE
015244  000167  177422                  JMP     RUN3            ;GO DO TEST AGAIN
015250  000167  177432          8$:     JMP     RUN.3           ;

015254  000000          ADDRESS:        0
```

```
5193                                    ;PROGRAM 4
5194                                    ;PROGRAM 4 CHECKS THE TRAP VECTOR ADDRESS.
5195                                    ;THE PROGRAM SIMULATES ACTIVATING THE BUTTON
5196                                    ;FOR EACH CHANNEL AND THEN READS
5197                                    ;THE CONTENTS OF THE ADDRESS.
5198                                    ;ON THE FIRST PASS THE CONTENTS WILL
5199                                    ;BE TYPED OUT FOR YOU THE
5200                                    ;USER TO VERIFY. AFTER THIS THE PROGRAM
5201                                    ;DOES A COMPARE TO THE PREVIOUSLY FOUND DATA
5202                                    ;AND REPORTS AN ERROR IF DIFFERENT THAN
5203                                    ;WHAT WAS FOUND BEFORE.
5204
5205  015256  012767  015256  001244  PRG4:   MOV     #PRG4,PRG.NO    ;SET FOR POWER FAIL
5206  015264  005067  163726           CLR     LSTERR          ;PREPARE ERROR CONDITIONS
5207  015270  012706  001100           MOV     #STACK,SP       ;SET THE STACK POINTER
5208  015274  012767  020000  163716   MOV     #20000,ICOUNT   ;SET ITERATION COUNT TO 20000(8)
5209  015302  005767  000544           TST     FLAG4           ;HAVE I BEEN HERE BEFOR??
5210  015306  001106                   BNE     TAG.A           ;BR IF NOT FIRST TIME HERE.
5211  015310  005167  000536           COM     FLAG4           ;SET THE FLAG
5212  015314  012705  000002           MOV     #2,R5           ;SET R5 FOR SWITCH 1
5213  015320  012704  016042           MOV     #LOC1,R4        ;SET STORAGE LOCATION
5214  015324  012737  016532  000004   MOV     #NO.TRAP,@#4    ;SET FOR TIME OUT TRAP
5215  015332  012767  000001  163664   MOV     #1,TEMP3        ;SET FOR MESSAGE ON CHANNEL NO.
5216  015340  104400  016054           1$:     TYPE    ,MCHAN          ;TYPE MESSAGE ABOUT CHANNEL
5217  015344  104400  016405                   TYPE    ,MSPACE
5218
5219  015350  016746  163650           MOV     TEMP3,-(SP)
5220  015354  104402                   TYPOC
5221  015356  104400  016405           TYPE    ,MSPACE
5222
5223
5224  015362  104400  016070           TYPE    ,MACTV          ;TYPE REST OF MESSAGE
5225  015366  104400  016103   2$:     TYPE    ,MADD1          ;TYPE ADDRESS MESSAGE
5226  015372  012700  173024           MOV     #173024,R0
5227  015376  005037  173024           CLR     @#173024
5228  015402  010537  173024           MOV     R5,@#173024     ;WRITE ROM WITH SWITCH
5229  015406  000240                   NOP                     ;WAIT ONE INSTR. TIME
5230  015410  012706  001100   3$:     MOV     #STACK,SP       ;SET THE STACK POINTER
5231  015414  012700  173024           MOV     #173024,R0      ;SET FOR ERROR MESSAGE
5232  015420  012737  016532  000004   MOV     #NO.TRAP,@#4    ;SET FOR NO MORE TRAPS
5233  015426  013767  173024  163572   MOV     @#173024,TEMP4  ;READ THE ADDRESS
5234
5235  015434  104400  016405           TYPE    ,MSPACE
5236  015440  016746  163562           MOV     TEMP4,-(SP)
5237  015444  104402                   TYPOC
5238  015446  013724  173024           MOV     @#173024,(R4)+  ;STORE THE INFORMATION FOUND
5239  015452  104400  016137           TYPE    ,MADD2          ;TYPE THE SECOND ADDRESS MSG
5240  015456  012700  173224           MOV     #173224,R0      ;SET FOR ERROR CONDITION.
5241  015462  013767  173224  163536   MOV     @#173224,TEMP4  ;STORE ROM DATA
5242  015470  104400  016405           TYPE    ,MSPACE
5243
5244  015474  016746  163526           MOV     TEMP4,-(SP)
5245  015500  104402                   TYPOC
5246
```

```
5247  015502  005267  163516                  INC     TEMP3           ;GET READY FOR NEXT SWITCH SETTING
5248  015506  000241                          CLC                     ;CLEAR THE CARRY BIT
5249  015510  006105                          ROL     R5              ;UPDATE R5
5250  015512  022705  000040                  CMP     #40,R5          ;ALL SIMULATED SWITCHS DONE?
5251  015516  001310                          BNE     1$              ;BR IF NOT ALL DONE
5252  015520  000167  177532                  JMP     PRG4            ;JMP    AND DO TEST AGAIN WITH OUT TYPE OUT
5253
5254  015524  012703  000002          TAG.A:  MOV     #2,R3           ;SIMULATE SWITCH 1
5255  015530  012704  016042                  MOV     #LOC1,R4        ;GET LOCATION WHERE DATA IS STORED
5256  015534  012737  016532  000004  1$:     MOV     #NO.TRAP,@#4    ;PREPARE FOR TIME OUT TRAP
5257  015542  005037  173024                  CLR     @#173024
5258  015546  010337  173024                  MOV     R3,@#173024     ;WRITE THE ROM
5259  015552  000240                          NOP                     ;WAIT ONE INSTR. TIME
5260  015554  012706  001100          2$:     MOV     #STACK,SP       ;SET THE STACK POINTER.
5261  015560  012737  016532  000004          MOV     #NO.TRAP,@#4    ;SET FOR NO MORE TRAPS.
5262  015566  012700  173024                  MOV     #173024,R0      ;SET FOR ERROR MESSAGE
5263  015572  011401                          MOV     (R4),R1         ;SET FOR COMPARISON
5264  015574  013705  173024                  MOV     @#173024,R5     ;GET THE DATA FROM THE ROM
5265  015600  012767  016042  163316          MOV     #LOC1,$GDDAT
5266  015606  012767  173024  163312          MOV     #173024,$BDDAT
5267  015614  016767  000222  163402          MOV     LOC1,TEMP3
5268  015622  013767  173024  163376          MOV     @#173024,TEMP4
5269  015630  020105                          CMP     R1,R5           ;IS THE DATA THE SAME??
5270  015632  001401                          BEQ     30$             ;BR IF GOOD DATA.
5271  015634  104001                          ERROR   1               ;ERROR. DATA READ FIRST TIME NOT THE SAME
5272  015636  012700  173224          30$:    MOV     #173224,R0      ;SET FOR ERROR MESSAGE
5273  015642  013705  173224                  MOV     @#173224,R5     ;READ THE ROM
5274  015646  012767  173224  163252          MOV     #173224,$BDDAT
5275  015654  013767  173224  163344          MOV     @#173224,TEMP4
5276  015662  020105                          CMP     R1,R5           ;IS THE DATA THE SAME?
5277  015664  001401                          BEQ     31$             ;BR IF GOOD DATA
5278  015666  104001                          ERROR   1               ;ERROR. DATA NOT THE SAME AS BEFORE.
5279  015670  005724                  31$:    TST     (R4)+           ;UPDATE DATA POINTER.
5280  015672  000241                          CLC                     ;CLEAR THE CARRY BIT
5281  015674  006103                          ROL     R3              ;UPDATE THE SIMULATED SWITHCH SETTING
5282  015676  022703  000040                  CMP     #40,R3          ;HAVE ALL SETTING BEEN DONE
5283  015702  001314                          BNE     1$              ;BR IF NOT DONE
5284  015704  005367  163310                  DEC     ICOUNT          ;ITERATION COUNT DONE
5285  015710  001305                          BNE     TAG.A           ;BR IF NOT DONE
5286  015712  012737  177777  173224          MOV     #-1,@#173224    ;WRITE SECOND TRAP VECTOR WITH -1
5287  015720  005037  173024                  CLR     @#173024        ;ZERO THE FIRST VECTOR
5288  015724  012700  173024                  MOV     #173024,R0      ;SET FOR TYPE OUT IF ERROR
5289  015730  016701  000106                  MOV     LOC1,R1         ;SET FOR TYPE OUT ROUTINE
5290  015734  013705  173024                  MOV     @#173024,R5     ;SAME AS ABOVE
5291  015740  012767  173024  163160          MOV     #173024,$BDDAT
5292  015746  013767  173024  163252          MOV     @#173024,TEMP4
5293  015754  020105                          CMP     R1,R5           ;IS DEFAULT LINE SELECTED =TO LINE 1
5294  015756  001401                          BEQ     32$             ;BR IF DEFAULT EQUALS LINE 1
5295  015760  104001                          ERROR   1               ;DATA NOT EQUAL TO LINE I
5296  015762  012737  177777  173024  32$:    MOV     #-1,@#173024    ;WRITE A -1 TO FIRST VECTOR
5297  015770  005037  173224                  CLR     @#173224        ;ZERO SECOND VECTOR
5298  015774  012700  173224                  MOV     #173224,R0      ;SET FOR TYPE OUT IF ERROR
5299  016000  016701  000036                  MOV     LOC1,R1         ;GET DATA
5300  016004  013705  173224                  MOV     @#173224,R5     ;READ ROM
```

```
5301  016010  012767  173224  163110          MOV     #173224,$BDDAT
5302  016016  013767  173224  163202          MOV     @#173224,TEMP4
5303
5304  016024  020105                          CMP     R1,R5           ;IS LINE 1 DEFAULT LINE
5305  016026  001401                          BEQ     33$             ;BR IF OK
5306  016030  104001                          ERROR   1               ;ERROR LINE 1 NOT DEFAULT LINE
5307  016032  004767  000534          33$:    JSR     PC,EOP          ;TYPE END MESSAGE.
5308  016036  000167  177214                  JMP     PRG4            ;GOTO PROGRAM 4 AGAIN
5309
5310  016042  000000                  LOC1:   0
5311  016044  000000                  LOC2:   0
5312  016046  000000                  LOC3:   0
5313  016050  000000                  LOC4:   0
5314  016052  000000                  FLAG4:  0
```

```
5315  016054  005015  041412  040510  MCHAN:   .ASCIZ <15><12><12>/CHANNEL /
      016070  041501  044524  040526  MACTV:   .ASCIZ/ACTIVATED./
      016103     015  040412  042104  MADD1:   .ASCIZ  <15><12>/ADDRESS 773024 CONTAINS: /
      016137     015  040412  042104  MADD2:   .ASCIZ  <15><12>/ADDRESS 773224 CONTAINS: /
              016174                  .EVEN
5316
5317  016174  005015  051120  043517  MSG3:    .ASCIZ <15><12>/PROGRAM NO. (1,2,3,4)  /
      016226  006414  016412  077437  MSG4:    .ASCIZ <14><15><12><35><37><177><177><177>/SOFTWARE MAP:/
      016254  005015  020012  054105  MSG5:    .ASCIZ  <15><12><12>/ EXTENDED SOFTWARE MAP:/
      016307     015  025012     000  MASTER:  .ASCIZ <15><12>/*/
      016313     007  006407  042412  M.END:   .ASCIZ <7><7><15><12>/END PASS BM873-Y/
      016340                          MFAIL:
      016340  005015  053520  020122          .ASCII  <15><12>/PWR UP AFTER/
      016356  005015  042522  046101          .ASCIZ  <15><12>/REAL PWR FAIL/
      016376  000044                  M.DOL:   .ASCIZ /$/
      016400  005015  037477     000  M.QM:    .ASCIZ <15><12>/??/
      016405     040  000040          MSPACE:  .ASCIZ  /  /
      016410  020040  000040          SPACE3:  .ASCIZ  /   /
      016414  005015     000          MCRLF:   .ASCIZ <15><12>
      016417     012     000          MLF:     .ASCIZ <12>
              016422                  .EVEN
5318
```

```
5319  016422  005067  162570          .PFAIL: CLR     LSTERR
5320  016426  013746  000004                  MOV     @#4,-(SP)
5321  016432  012737  016462  000004          MOV     #1$,@#4
5322  016440  005737  173000                  TST     @#173000        ;IS THIS PF REAL?
5323  016444  000240                          NOP     ;TRAP IS CAUSED BY LOADER
5324  016446  012737  016472  000024          MOV     #PWR.UP,@#24    ;ITS REAL. PREPARE FOR PWR UP
5325  016454  012637  000004                  MOV     (SP)+,@#4
5326  016460  000000                          HALT
5327  016462  005726                  1$:     TST     (SP)+   ;POP THE STACK.
5328  016464  012637  000004                  MOV     (SP)+,@#4
5329  016470  000000                          HALT    ;HARDWARE ERROR.  BOOT DIDN'T FORCE
5330                          ;HIGH ADDR LINES AND LOAD BUTTON WAS ACTIVATED
5331  016472  012737  016422  000024  PWR.UP: MOV     #.PFAIL,@#24
5332  016500  012706  001100                  MOV     #STACK,SP
5333  016504  005000                          CLR     R0              ;SET DELAY
5334  016506  062700  000001          1$:     ADD     #1,R0           ;WAIT FOR TTY
5335  016512  001375                          BNE     1$
5336  016514  104400  016340                  TYPE    ,MFAIL          ;TYPE FAILED.
5337  016520  005067  161252                  CLR     PS              ;SET STATUS TO ZERO
5338  016524  000177  000000                  JMP     @PRG.NO
5339  016530  000000          PRG.NO: 0
5340  016532                  NO.TRAP:
5341  016532  011667  000032                  MOV     (SP),XSTORE
5342  016536  032716  100000                  BIT     #BIT15,(SP)
5343  016542  001410                          BEQ     1$
5344  016544  011600                          MOV     (SP),R0
5345  016546  104004                          ERROR   4
5346  016550  012706  001100                  MOV     #STACK,SP
5347  016554  005067  161216                  CLR     PS
5348  016560  000177  177744                  JMP     @PRG.NO
5349  016564  104003          1$:     ERROR   3
5350  016566  000002                          RTI
5351  016570  000000          XSTORE: 0
5352
5353  016572  005067  162420          EOP:    CLR     LSTERR
5354  016576  104400  016313                  TYPE    ,M.END
5355  016602  104400  016652                  TYPE    ,VERSON
5356  016606  013701  000042                  MOV     @#42,R1
5357  016612  001416                          BEQ     X1
5358  016614  022767  012566  177706          CMP     #PRG1,PRG.NO
5359  016622  001002                          BNE     .+6
5360  016624  000167  176426                  JMP     PRG4
5361  016630  013701  000042                  MOV     @#42,R1
5362  016634  001405                          BEQ     X1
5363  016636  000005                          RESET
5364  016640                  $ENDAD:
5365  016640  004711          LOGIC:  JSR     PC,(R1)
5366  016642  000240                  NOP
5367  016644  000240                  NOP
5368  016646  000240                  NOP
5369  016650  000207          X1:     RTS     PC
5370  016652  000101          VERSON: 101                     ;SEVEN BIT ASCII FOR DEFAULT "A"
```

```
5371  016654  005015  041520  020072  MERRPC: .ASCIZ  <15><12>/PC: /
5372  016662     000
5373          016664                  .EVEN
5374                                  .MCALL .$EOP, .$TYPE, .$TYPOCT, .$POWER, .$READ
5375                                  ;;*****************************************************************
5376
5377                                  .SBTTL  TYPE ROUTINE
5378
5379                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
5380                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
5381                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
5382                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
5383                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
5384                                  ;*
5385                                  ;*CALL:
5386                                  ;*1) USING A TRAP INSTRUCTION
5387                                  ;*      TYPE    ,MESADR                 ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
5388                                  ;*OR
5389                                  ;*      TYPE
5390                                  ;*      MESADR
5391                                  ;*
5392                                  ;*2) USING A JSR INSTRUCTION
5393                                  ;*      MOV     PS,-(SP)                ;PUSH PROCESSOR STATUS WORD ON THE STACK
5394                                  ;*      JSR     PC,$TYPE                ;CALL TYPE ROUTINE
5395                                  ;*      MESADDR                         ;FIRST ADRESS OF MESSAGE
5396
5397  016664  105767  162261         $TYPE:  TSTB    $TPFLG                  ;IS THERE A TERMINAL?
5398  016670  100002                  BPL     1$                      ;BR IF YES
5399  016672  000000                  HALT                            ;HALT HERE IF NO TERMINAL
5400  016674  000407                  BR      3$                      ;LEAVE
5401  016676  010046                  1$:     MOV     R0,-(SP)                ;SAVE R0
5402  016700  017600  000002          MOV     @2(SP),R0               ;GET ADDRESS OF ASCIZ STRING
5403  016704  112046                  2$:     MOVB    (R0)+,-(SP)             ;PUSH CHARACTER TO BE TYPED ONTO STACK
5404  016706  001005                  BNE     4$                      ;BR IF IT ISN'T THE TERMINATOR
5405  016710  005726                  TST     (SP)+                   ;IF TERMINATOR POP IT OFF THE STACK
5406  016712  012600                  MOV     (SP)+,R0                ;RESTORE R0
5407  016714  062716  000002          3$:     ADD     #2,(SP)                 ;ADJUST RETURN PC
5408  016720  000002                  RTI                             ;RETURN
5409  016722  004767  000026          4$:     JSR     PC,7$                   ;GO TYPE THIS CHARACTER
5410  016726  126726  162216          5$:     CMPB    $FILLC,(SP)+            ;IS IT TIME FOR FILLER CHARS.?
5411  016732  001364                  BNE     2$                      ;IF NO GO GET NEXT CHAR.
5412  016734  016746  162206          MOV     $NULL,-(SP)             ;GET # OF FILLER CHARS. NEEDED
5413                                                                  ;AND THE NULL CHAR.
5414  016740  105366  000001          6$:     DECB    1(SP)                   ;DOES A NULL NEED TO BE TYPED?
5415  016744  002770                  BLT     5$                      ;BR IF NO--GO POP THE NULL OFF OF STACK
5416  016746  004767  000002          JSR     PC,7$                   ;GO TYPE A NULL
5417  016752  000772                  BR      6$                      ;LOOP
5418  016754  105777  162162          7$:     TSTB    @$TPS                   ;WAIT UNTIL PRINTER IS READY
5419  016760  100375                  BPL     7$
5420  016762  116677  000002  162154  MOVB    2(SP),@$TPB             ;LOAD CHAR TO BE TYPED INTO DATA REG.
5421  016770  000207                  RTS     PC
5422                                  ;;*****************************************************************
5423
5424                                  .SBTTL  TTY INPUT ROUTINE
```

```
                                ;*INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*CALL:
                                ;*      RDCHR                           ;INPUT A SINGLE CHARACTER FROM THE TTY
                                ;*      RETURN HERE                     ;CHARACTER IS ON THE STACK

016772  011646                  $RDCHR: MOV     (SP),-(SP)              ;PUSH DOWN THE PC
016774  016666  000004  000002          MOV     4(SP),2(SP)             ;SAVE THE PS
017002  105777  162130          1$:     TSTB    @$TKS                   ;WAIT FOR
017006  100375                          BPL     1$                      ;A CHARACTER
017010  117766  162124  000004          MOVB    @$TKB,4(SP)             ;READ THE TTY
017016  042766  177600  000004          BIC     #^C<177>,4(SP)          ;GET RID OF JUNK IF ANY
017024  000002                          RTI                             ;GO BACK TO USER
                                ;;*****************************************************************
                                ;*INPUT A STRING FROM THE TTY
                                ;*CALL:
                                ;*      RDLIN                           ;INPUT A STRING FROM THE TTY
                                ;*      RETURN HERE                     ;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                ;*                                      ;TERMINATOR WILL BE A BYTE OF ALL 0'S

017026  010346                  $RDLIN: MOV     R3,-(SP)                ;SAVE R3
017030  012703  017134          1$:     MOV     #$TTYIN,R3              ;GET ADDRESS
017034  022703  017144          2$:     CMP     #$TTYIN+8.,R3           ;BUFFER FULL?
017040  101405                          BLOS    4$                      ;BR IF YES
017042  104410                          RDCHR                           ;GO READ ONE CHARACTER FROM THE TTY
017044  112613                          MOVB    (SP)+,(R3)              ;GET CHARACTER
017046  122713  000177                  CMPB    #177,(R3)               ;IS IT A RUBOUT
017052  001003                          BNE     3$                      ;SKIP IF NOT
017054  104400  001152          4$:     TYPE    $QUES                   ;TYPE A '?'
017060  000763                          BR      1$                      ;CLEAR THE BUFFER AND LOOP
017062  111367  000044          3$:     MOVB    (R3),8$                 ;ECHO THE CHARACTER
017066  104400  017132                  TYPE    ,8$
017072  122723  000015                  CMPB    #15,(R3)+               ;CHECK FOR RETURN
017076  001356                          BNE     2$                      ;LOOP IF NOT RETURN
017100  105063  177777                  CLRB    -1(R3)                  ;CLEAR RETURN (THE 15)
017104  104400  001154                  TYPE    $LF                     ;TYPE A LINE FEED
017110  012603                          MOV     (SP)+,R3                ;RESTORE R3
017112  011646                          MOV     (SP),-(SP)              ;ADJUST THE STACK AND PUT ADDRESS OF THE
017114  016666  000004  000002          MOV     4(SP),2(SP)             ;       FIRST ASCII CHARACTER ON IT
017122  012766  017134  000004          MOV     #$TTYIN,4(SP)
017130  000002                          RTI                             ;RETURN
017132     000          8$:     .BYTE   0                               ;STORAGE FOR ASCII CHAR. TO TYPE
017133     000                  .BYTE   0                               ;TERMINATOR
017134  000010          $TTYIN: .BLKB   8.                              ;RESERVE 8 BYTES FOR TTY INPUT
                                ;;*****************************************************************

                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
                                ;*      TYPOS                           ;CALL FOR TYPEOUT
                                ;*      .BYTE   N                       ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
```

```
5479                                    ;*      .BYTE   M                       ;M=1 OR 0
5480                                    ;*                                      ;1=TYPE LEADING ZEROS
5481                                    ;*                                      ;0=SUPPRESS LEADING ZEROS
5482                                    ;*
5483                                    ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
5484                                    ;*$TYPOS OR $TYPOC
5485                                    ;*CALL:
5486                                    ;*      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
5487                                    ;*      TYPON                           ;CALL FOR TYPEOUT
5488                                    ;*
5489                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
5490                                    ;*CALL:
5491                                    ;*      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
5492                                    ;*      TYPOC                           ;CALL FOR TYPEOUT
5493
5494  017144  017646  000000           $TYPOS: MOV     @(SP),-(SP)             ;PICKUP THE MODE
5495  017150  116667  000001  000211           MOVB    1(SP),$OFILL            ;LOAD ZERO FILL SWITCH
5496  017156  112667  000207                   MOVB    (SP)+,$OMODE+1          ;NUMBER OF DIGITS TO TYPE
5497  017162  062716  000002                   ADD     #2,(SP)                 ;ADJUST RETURN ADDRESS
5498  017166  000406                           BR      $TYPON
5499  017170  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL               ;SET THE ZERO FILL SWITCH
5500  017176  112767  000006  000165          MOVB    #6,$OMODE+1             ;SET FOR SIX(6) DIGITS
5501  017204  112767  000005  000154  $TYPON: MOVB    #5,$OCNT                ;SET THE ITERATION COUNT
5502  017212  010346                           MOV     R3,-(SP)                ;SAVE R3
5503  017214  010446                           MOV     R4,-(SP)                ;SAVE R4
5504  017216  010546                           MOV     R5,-(SP)                ;SAVE R5
5505  017220  116704  000145                   MOVB    $OMODE+1,R4             ;GET THE NUMBER OF DIGITS TO TYPE
5506  017224  005404                           NEG     R4
5507  017226  062704  000006                   ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
5508  017232  110467  000132                   MOVB    R4,$OMODE               ;SAVE IT FOR USE
5509  017236  116704  000125                   MOVB    $OFILL,R4               ;GET THE ZERO FILL SWITCH
5510  017242  016605  000012                   MOV     12(SP),R5               ;PICKUP THE INPUT NUMBER
5511  017246  005003                           CLR     R3                      ;CLEAR THE OUTPUT WORD
5512  017250  006105                   1$:     ROL     R5                      ;ROTATE MSB INTO "C"
5513  017252  000404                           BR      3$                      ;GO DO MSB
5514  017254  006105                   2$:     ROL     R5                      ;FORM THIS DIGIT
5515  017256  006105                           ROL     R5
5516  017260  006105                           ROL     R5
5517  017262  010503                           MOV     R5,R3
5518  017264  006103                   3$:     ROL     R3                      ;GET LSB OF THIS DIGIT
5519  017266  105367  000076                   DECB    $OMODE                  ;TYPE THIS DIGIT?
5520  017272  100016                           BPL     7$                      ;BR IF NO
5521  017274  042703  177770                   BIC     #177770,R3              ;GET RID OF JUNK
5522  017300  001002                           BNE     4$                      ;TEST FOR 0
5523  017302  005704                           TST     R4                      ;SUPPRESS THIS 0?
5524  017304  001403                           BEQ     5$                      ;BR IF YES
5525  017306  005204                   4$:     INC     R4                      ;DON'T SUPPRESS ANYMORE 0'S
5526  017310  052703  000060                   BIS     #'0,R3                  ;MAKE THIS DIGIT ASCII
5527  017314  052703  000040           5$:     BIS     #' ,R3                  ;MAKE ASCII IF NOT ALREADY
5528  017320  110367  000040                   MOVB    R3,8$                   ;SAVE FOR TYPING
5529  017324  104400  017364                   TYPE    ,8$                     ;GO TYPE THIS DIGIT
5530  017330  105367  000032           7$:     DECB    $OCNT                   ;COUNT BY 1
5531  017334  003347                           BGT     2$                      ;BR IF MORE TO DO
5532  017336  002402                           BLT     6$                      ;BR IF DONE
```

```
5533  017340  005204                          INC     R4                      ;INSURE LAST DIGIT ISN'T A BLANK
5534  017342  000744                          BR      2$                      ;GO DO THE LAST DIGIT
5535  017344  012605                  6$:     MOV     (SP)+,R5                ;RESTORE R5
5536  017346  012604                          MOV     (SP)+,R4                ;RESTORE R4
5537  017350  012603                          MOV     (SP)+,R3                ;RESTORE R3
5538  017352  016666  000002  000004          MOV     2(SP),4(SP)             ;SET THE STACK FOR RETURNING
5539  017360  012616                          MOV     (SP)+,(SP)
5540  017362  000002                          RTI                             ;RETURN
5541  017364    000                   8$:     .BYTE   0                       ;STORAGE FOR ASCII DIGIT
5542  017365    000                           .BYTE   0                       ;TERMINATOR FOR TYPE ROUTINE
5543  017366    000                   $OCNT:  .BYTE   0                       ;OCTAL DIGIT COUNTER
5544  017367    000                   $OFILL: .BYTE   0                       ;ZERO FILL SWITCH
5545  017370  000000                  $OMODE: 0                               ;NUMBER OF DIGITS TO TYPE
5546                                  ;;**************************************************************
5547
5548                                  .SBTTL  SCOPE HANDLER ROUTINE
5549
5550                                  ;*SW14=1        LOOP ON TEST
5551                                  ;*THE TEST NUMBER ($TSTNM) IS INCREMENTED AND DISPLAYED IN DISPLAY<7:0>
5552                                  ;*AND THE ERROR FLAG ($ERFLG) IS DISPLAYED IN DISPLAY<15:08>
5553
5554  017372                          $SCOPE:
5555  017372  006137  177570                  ROL     @#SWR                   ;LOOP ON PRESENT TEST?
5556  017376  100425                          BMI     $OVER                   ;YES IF SW14=1
5557                                          ;#####START OF CODE FOR THE XOR TESTER#####
5558  017400  000416                  $XTSTR: BR      6$                      ;IF RUNNING ON THE "XOR" TESTER CHANGE
5559                                                                          ;THIS INSTRUCTION TO A "NOP" (NOP=240)
5560  017402  013746  000004                  MOV     @#ERRVEC,-(SP)          ;SAVE THE CONTENTS OF THE ERROR VECTOR
5561  017406  012737  017426  000004          MOV     #5$,@#ERRVEC            ;SET FOR TIMEOUT
5562  017414  005737  177060                  TST     @#177060                ;TIME OUT ON XOR?
5563  017420  012637  000004                  MOV     (SP)+,@#ERRVEC          ;RESTORE THE ERROR VECTOR
5564  017424  000404                          BR      $SVLAD                  ;GO TO THE NEXT TEST
5565  017426  022626                  5$:     CMP     (SP)+,(SP)+             ;CLEAR THE STACK AFTER A TIME OUT
5566  017430  012637  000004                  MOV     (SP)+,@#ERRVEC          ;RESTORE THE ERROR VECTOR
5567  017434  000406                          BR      $OVER                   ;LOOP ON THE PRESENT TEST
5568  017436                          6$:     ;#####END OF CODE FOR THE XOR TESTER#####
5569  017436  105267  161440          $SVLAD: INCB    $TSTNM                  ;COUNT TEST NUMBERS
5570  017442  011667  161440                  MOV     (SP),$LPADR             ;SAVE SCOPE LOOP ADDRESS
5571  017446  105067  161431                  CLRB    $ERFLG                  ;ZERO THE ERROR FLAG
5572  017452  016737  161424  177570  $OVER:  MOV     $TSTNM,@#DISPLAY        ;DISPLAY TEST NUMBER
5573  017460  016716  161422                  MOV     $LPADR,(SP)             ;FUDGE RETURN ADDRESS
5574  017464  000002                          RTI                             ;FIXES PS
5575                                  ;;**************************************************************
5576
5577                                  .SBTTL  ERROR HANDLER ROUTINE
5578
5579                                  ;*SW15=1        HALT ON ERROR
5580                                  ;*SW13=1        INHIBIT ERROR TYPEOUTS
5581                                  ;*GO TO $ERRTYP ON ERROR
5582
5583  017466                          $ERROR:
5584  017466  105267  161411          7$:     INCB    $ERFLG                  ;SET THE ERROR FLAG
5585  017472  001775                          BEQ     7$                      ;DON'T LET THE FLAG GO TO ZERO
5586  017474  016737  161402  177570          MOV     $TSTNM,@#DISPLAY        ;DISPLAY TEST NUMBER AND ERROR FLAG
```

```
5587  017502  005267  161404              INC     $ERTTL                  ;INC THE ERROR COUNT
5588  017506  011667  161404              MOV     (SP),$ERRPC             ;GET ADDRESS OF ERROR INSTRUCTION
5589  017512  162767  000002  161376      SUB     #2,$ERRPC
5590  017520  117767  161372  161366      MOVB    @$ERRPC,$ITEMB          ;STRIP AND SAVE THE ERROR ITEM CODE
5591  017526  032737  020000  177570      BIT     #SW13,@#SWR             ;SKIP TYPEOUT IF SET
5592  017534  001004                      BNE     2$                      ;SKIP TYPEOUTS
5593  017536  004737  017560              JSR     PC,@#$ERRTYP            ;GO TO USER ERROR ROUTINE
5594  017542  104400  001153              TYPE    ,$CRLF
5595  017546  005737  177570      2$:     TST     @#SWR                   ;HALT ON ERROR
5596  017552  100001                      BPL     3$                      ;SKIP IF CONTINUE
5597  017554  000000                      HALT                            ;HALT ON ERROR!
5598  017556  000002              3$:     RTI                             ;RETURN
5599                                      ;;**************************************************************
5600
5601                                      .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
5602
5603                                      ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
5604                                      ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
5605                                      ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
5606
5607  017560                              $ERRTYP:
5608  017560  104400  001153                      TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
5609  017564  010046                              MOV     R0,-(SP)                ;SAVE R0
5610  017566  005000                              CLR     R0                      ;PICKUP THE ITEM INDEX
5611  017570  153700  001114                      BISB    @#$ITEMB,R0
5612  017574  001004                              BNE     1$                      ;IF ITEM NUMBER IS ZERO, JUST
5613                                                                              ;TYPE THE PC OF THE ERROR
5614  017576  016746  161314                      MOV     $ERRPC,-(SP)            ;SAVE $ERRPC FOR TYPEOUT
5615                                                                              ;ERROR ADDRESS
5616  017602  104402                              TYPOC                           ;GO TYPE--OCTAL ASCII(ALL DIGITS)
5617  017604  000426                              BR      6$                      ;GET OUT
5618  017606  005300              1$:             DEC     R0                      ;ADJUST THE INDEX SO THAT IT WILL
5619  017610  006300                              ASL     R0                      ;       WORK FOR THE ERROR TABLE
5620  017612  006300                              ASL     R0
5621  017614  006300                              ASL     R0
5622  017616  062700  001156                      ADD     #$ERRTB,R0              ;FORM TABLE POINTER
5623  017622  012067  000004                      MOV     (R0)+,2$                ;PICKUP "ERROR MESSAGE" POINTER
5624  017626  001404                              BEQ     3$                      ;SKIP TYPEOUT IF NO POINTER
5625  017630  104400                              TYPE                            ;TYPE THE "ERROR MESSAGE"
5626  017632  000000              2$:             .WORD   0                       ;"ERROR MESSAGE" POINTER GOES HERE
5627  017634  104400  001153                      TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
5628  017640  012067  000004      3$:             MOV     (R0)+,4$                ;PICKUP "DATA HEADER" POINTER
5629  017644  001404                              BEQ     5$                      ;SKIP TYPEOUT IF 0
5630  017646  104400                              TYPE                            ;TYPE THE "DATA HEADER"
5631  017650  000000              4$:             .WORD   0                       ;"DATA HEADER" POINTER GOES HERE
5632  017652  104400  001153                      TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
5633  017656  011000              5$:             MOV     (R0),R0                 ;PICKUP "DATA TABLE" POINTER
5634  017660  001004                              BNE     7$                      ;GO TYPE THE DATA
5635  017662  012600              6$:             MOV     (SP)+,R0                ;RESTORE R0
5636  017664  104400  001153                      TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
5637  017670  000207                              RTS     PC                      ;RETURN
5638  017672                      7$:
5639  017672  013046                              MOV     @(R0)+,-(SP)            ;SAVE @(R0)+ FOR TYPEOUT
5640  017674  104402                              TYPOC                           ;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
5641  017676  005710                          TST     (R0)                    ;IS THERE ANOTHER NUMBER?
5642  017700  001770                          BEQ     6$                      ;BR IF NO
5643  017702  104400  017710                  TYPE    ,8$                     ;TYPE TWO(2) SPACES
5644  017706  000771                          BR      7$                      ;LOOP
5645  017710  020040      000       8$:       .ASCIZ  /  /                    ;TWO(2) SPACES
5646          017714                          .EVEN
5647                                ;;*****************************************************************
5648
5649                                .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
5650
5651                                ;*CALL:
5652                                ;*      RDOCT                           ;READ AN OCTAL NUMBER
5653                                ;*      RETURN HERE                     ;LOW ORDER BITS ARE ON TOP OF THE STACK
5654                                ;*                                      ;HIGH ORDER BITS ARE IN $HIOCT
5655
5656  017714  011646                $RDOCT: MOV     (SP),-(SP)              ;PROVIDE SPACE FOR THE
5657  017716  016666  000004  000002        MOV     4(SP),2(SP)             ;INPUT NUMBER
5658  017724  010046                        MOV     R0,-(SP)                ;PUSH R0 ON STACK
5659  017726  010146                        MOV     R1,-(SP)                ;PUSH R1 ON STACK
5660  017730  010246                        MOV     R2,-(SP)                ;PUSH R2 ON STACK
5661  017732  104412                1$:     RDLIN                           ;READ AN ASCIZ LINE
5662  017734  012600                        MOV     (SP)+,R0                ;GET ADDRESS OF 1ST CHARACTER
5663  017736  005001                        CLR     R1                      ;CLEAR DATA WORD
5664  017740  005002                        CLR     R2
5665  017742  112046                2$:     MOVB    (R0)+,-(SP)             ;PICKUP THIS CHARACTER
5666  017744  001412                        BEQ     3$                      ;IF ZERO GET OUT
5667  017746  006301                        ASL     R1                      ;*2
5668  017750  006102                        ROL     R2
5669  017752  006301                        ASL     R1                      ;*4
5670  017754  006102                        ROL     R2
5671  017756  006301                        ASL     R1                      ;*8
5672  017760  006102                        ROL     R2
5673  017762  042716  177770              BIC     #^C7,(SP)               ;STRIP THE ASCII JUNK
5674  017766  062601                        ADD     (SP)+,R1                ;ADD IN THIS DIGIT
5675  017770  000764                        BR      2$                      ;LOOP
5676  017772  005726                3$:     TST     (SP)+                   ;CLEAN TERMINATOR FROM STACK
5677  017774  010166  000012              MOV     R1,12(SP)               ;SAVE THE RESULT
5678  020000  010267  000010              MOV     R2,$HIOCT
5679  020004  012602                        MOV     (SP)+,R2                ;POP STACK INTO R2
5680  020006  012601                        MOV     (SP)+,R1                ;POP STACK INTO R1
5681  020010  012600                        MOV     (SP)+,R0                ;POP STACK INTO R0
5682  020012  000002                        RTI                             ;RETURN
5683  020014  000000                $HIOCT: .WORD   0                       ;HIGH ORDER BITS GO HERE
5684                                ;;*****************************************************************
5685
5686                                .SBTTL  TRAP DECODER
5687
5688                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
5689                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
5690                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
5691                                ;*GO TO THAT ROUTINE.
5692
5693  020016  010046                $TRAP:  MOV     R0,-(SP)                ;SAVE R0
5694  020020  016600  000002              MOV     2(SP),R0                ;GET TRAP ADDRESS
```

```
                        TST     -(R0)                   ;BACKUP BY 2
                        MOVB    (R0),R0                 ;GET RIGHT BYTE OF TRAP
020036                  MOV     $TRPAD(R0),R0           ;INDEX TO TABLE
                        RTS     R0                      ;GO TO ROUTINE


                .SBTTL  TRAP TABLE

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

                ;       ROUTINE
                ;       -------
020036          $TRPAD:
                        $TYPE           ;CALL=TYPE      TRAP+0(104400)  TTY TYPEOUT ROUTINE
                        $TYPOC          ;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING
                        $TYPOS          ;CALL=TYPOS     TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZE
                        $TYPON          ;CALL=TYPON     TRAP+6(104406)  TYPE OCTAL NUMBER (AS PER LAST C
                        $RDCHR          ;CALL=RDCHR     TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
                        $RDLIN          ;CALL=RDLIN     TRAP+12(104412) TTY TYPEIN STRING ROUTINE
                        $RDOCT          ;CALL=RDOCT     TRAP+14(104414) READ AN OCTAL NUMBER FROM TTY
020054  005015  047522  020115  EM1:    .ASCIZ  <15><12>/ROM READ DATA COMPARISON ERROR./
020116  005015  051127  052111  EM2:    .ASCIZ  <15><12>/WRITING ROM FAILED TO TRAP./
020154  005015  047125  054105  EM3:    .ASCIZ  <15><12>/UNEXP TRAP WHILE READING ROM./
020214  005015  040506  040524  EM4:    .ASCIZ  <15><12>/FATAL TRAP.  ROM PC ON STACK./
020254  005015  041520  020040  DH1:    .ASCII  <15><12>/PC      SOFT    ROM/
020301     015  040412  042104          .ASCIZ  <15><12>/ADDRESS ADDRESS ADDRESS EXPECTED FOUND /
020353     015  050012  004503  DH2:    .ASCII  <15><12>/PC      ROM/
020363     015  040412  042104          .ASCIZ  <15><12>/ADDRESS ADDRESS/
020405     015  050012  020103  DH3:    .ASCII  <15><12>/PC OF   PROGRAM /
020427     015  052012  040522          .ASCIZ  <15><12>/TRAP    ADDRESS/
           020452                       .EVEN
020452  001116  001124  001126  DT1:    .WORD   $ERRPC,$GDDAT,$BDDAT,TEMP3,TEMP4,0
020460  001224  001226  000000
020466  001116  001226  000000  DT2:    .WORD   $ERRPC,TEMP4,0
020474  001116  016570  000000  DT3:    .WORD   $ERRPC,XSTORE,0
           020542                       .=.+40
020542                          CORMAX:
           000001                       .END
```

```
ADDRES  015254          5005*  5023   5042*  5053*  5192#
ALLEND  012336          4656*  4664*  4682*  4690*  4698*  4706*  4714*  4731*  4737*  4743*  4749*  4755*  4761*
                        4767*  4773*  4800#  5063
BIT0   = 000001          431*  4851   4882   4950   5098   5146   5176
BIT00  = 000001          421#   431
BIT01  = 000002          420#   430
BIT02  = 000004          419#   429
BIT03  = 000010          418#   428
BIT04  = 000020          417#   427
BIT05  = 000040          416#   426
BIT06  = 000100          415#   425
BIT07  = 000200          414#   424
BIT08  = 000400          413#   423
BIT09  = 001000          412#   422
BIT1   = 000002          430#
BIT10  = 002000          411#
BIT11  = 004000          410#   4840   4877   5137   5172
BIT12  = 010000          409#
BIT13  = 020000          408#
BIT14  = 040000          407#
BIT15  = 100000          406#   5342
BIT2   = 000004          429#
BIT3   = 000010          428#
BIT4   = 000020          427#
BIT5   = 000040          426#
BIT6   = 000100          425#
BIT7   = 000200          424#
BIT8   = 000400          423#
BIT9   = 001000          422#
BM.ERR  013235          4779   4899#
BM873X  013160          4724   4899#
BPTVEC= 000014           438#
CONT    011676          4639   4721#
CORMAX  020542          5723#  5724
DEAD    014670          5063*  5090   5106*  5109#
DH.2    013672          4912   4987#
DH.2A   014066          4954   4987#
DH.2B   013754          4913   4955   4987#  5065   5104
DH1     020254           509   5716#
DH2     020353           516    529   5716#
DH3     020405           523   5716#
DISPLA= 177570           363#  5572*  5586*
DT1     020452           510   5717#
DT2     020466           517    530   5720#
DT3     020474           524   5721#
EMTVEC= 000030           441#  4623*  4624*
EM1     020054           508   5716#
EM2     020116           515   5716#
EM3     020154           522   5716#
EM4     020214           528   5716#
END.YA  001776           740#  4656   4737
END.YB  002776          1112#  4664   4743
END.YC  003776          1580#  4682   4749
END.YD  004776          2222#  4690   4755
```

F12


```
END.YF  005776          3255#   4698    4761
END.YG  006776          3829#   4706    4767
END.YH  007776          4599#   4714    4773
END.YX  010776          4610#   4802
END.Y.  010376          4608#   4731
EOP     016572          4894    5188    5307    5353#
ERRVEC= 000004           434#   5560    5561*   5563*   5566*
EXTEND  012342          4802#   5035    5101    5106
EXTMAP  012340          4801#   4856    4953    5105    5151
FLAG4   016052          4636*   5209    5211*   5314#
GNS   = ****** U         346    5709    5710    5711    5712    5713    5714    5715
ICOUNT  001220           534#   4820*   4892*   5118*   5186*   5208*   5284*
INITFG  012344          4628*   4721    4723*   4803#
IOTVEC= 000020           439#   4621*   4622*
LAST    013156          4825*   4845    4853    4855*   4897#   4908*   4909*   4944    5122*   5141    5148    5150*
LASTA   012346          4657*   4665*   4683*   4691*   4699*   4707*   4715*   4732*   4738*   4744*   4750*   4756*   4762*
                        4768*   4774*   4804#   4825    4887    4908    5122    5181
LOC1    016042          5213    5255    5265    5267    5289    5299    5310#
LOC2    016044          5311#
LOC3    016046          5312#
LOC4    016050          5313#
LOGIC   016640          5365#
LSTERR  001216           533#   4631*   5206*   5319*   5353*
MACTV   016070          5224    5315#
MADD1   016103          5225    5315#
MADD2   016137          5239    5315#
MAP.YA  001400           556#   4645    4655    4736
MAP.YB  002000           745#   4661    4663    4742
MAP.YC  003000          1114#   4669    4673    4681    4748
MAP.YD  004000          1585#   4687    4689    4754
MAP.YF  005000          2225#   4695    4697    4760
MAP.YG  006000          3264#   4703    4705    4766
MAP.YH  007000          3833#   4711    4713    4772
MAP.YX  010400          4609#   4801
MAP.Y.  010000          4607#   4730    4918    4934
MASTER  016307          5006    5317#
MCHAN   016054          5216    5315#
MCRLF   016414          4922    4961    5024    5038    5070    5317#
MERRPC  016654          5371#
MFAIL   016340          5317#   5336
MLF     016417          5317#
MSG3    016174          4781    5317#
MSG4    016226          5064    5317#
MSG5    016254          5103    5317#
MSPACE  016405          4926    4939    4965    4973    5029    5075    5080    5088    5217    5221    5235    5242    5317#
M.DOL   016376          5317#
M.END   016313          5317#   5354
M.QM    016400          4797    5020    5044    5317#
NMATCH  012450          4719    4808#
NOROM   012350          4641    4805#
NOROMS  012360          4805    4808#
NO.TRA  016532          4821    4907    5117    5214    5232    5256    5261    5340#
PC    =%000007           375#   4894*   5188*   5307*   5365*   5369*   5409*   5416*   5421*   5593*   5637*
PIRQ  = 177772           361#
```

```
PIRQVE= 000240           445#
PRG.NO  016530          4633*  4819*  4906*  5003*  5205*  5338  5339#  5348  5358
PRG.1   012602          4821#  4896
PRG1    012566          4659   4667   4685   4693   4701   4709  4717   4787  4819#  4895  5358
PRG2    013312          4790   4906#  4986
PRG3    014252          4793   5003#
PRG4    015256          4796   5205#  5252   5308   5360
PS    = 177776           358#   359   5337*  5347*
PSW   = 177776           359#
PWRVEC= 000024           440#
PWR.UP  016472          5324   5331#
RDCHR = 104410          5450   5713#
RDLIN = 104412          4725   4782   5007   5661   5714#
RDOCT = 104414          5031   5052   5715#
RESTRT  011000           350   4615#  4798
RESVEC= 000010           435#
RUN.3   014706          5119#  5190
RUN3    014672          5117#  5189
R0    =%000000           366#  4822*  4827   4829   4830   4832  4834   4835  4845   4848  4857   4863*  4866
                        4868   4869   4871   4872*  4874   4884  4887   4890  4910*  4915  4916   4920   4921
                        4931   4932   4936   4937   4944   4952* 4957   4958  4960   4969  4970   4978   4979
                        4984*  5062*  5067   5068   5069   5082  5083   5084  5092   5105* 5119*  5124   5126
                        5127   5129   5131   5132   5141   5144  5152   5157* 5161   5163  5164   5166   5167*
                        5169   5178   5181   5184   5226*  5231* 5240*  5262* 5272*  5288* 5298*  5333*  5334*
                        5344*  5401   5402*  5403   5406*  5609  5610*  5611* 5618*  5619* 5620*  5621*  5622*
                        5623   5628   5633*  5635*  5639   5641  5658   5662* 5665   5681* 5693   5694*  5695
                        5696*  5697*  5698*
R1    =%000001           367#  4915*  4917   4931*  4933   4957* 4959   4969* 4971   5004* 5005   5026*  5032*
                        5034   5035   5037   5042   5263*  5269  5276   5289* 5293   5299* 5304   5356*  5361*
                        5365   5659   5663*  5667*  5669*  5671* 5674*  5677  5680*
R2    =%000002           368#  4726*  4727*  4728   4734   4740  4746   4752  4758   4764  4770   4776   4783*
                        4784   5008*  5009*  5010   5013   5015  5018   5660  5664*  5668* 5670*  5672*  5678
                        5679*
R3    =%000003           369#  4784*  4785   4788   4791   4794  4826*  4842* 4865*  4879* 4911*  4920*  4936*
                        4953*  4960*  4978*  5090*  5091   5092  5100   5101  5123*  5139* 5160*  5174*  5254*
                        5258   5281*  5282   5446   5447*  5448  5451*  5452  5456   5458  5460*  5462*  5502
                        5511*  5517*  5518*  5521*  5526*  5527* 5528   5537*
R4    =%000004           370#  4824*  4829   4832   4833   4836  4848   4856* 5121*  5126  5129   5130   5133
                        5144   5151*  5213*  5238*  5255*  5263  5279   5503  5505*  5506* 5507*  5508   5509*
                        5523   5525*  5533*  5536*
R5    =%000005           371#  5212*  5228   5249*  5250   5264* 5269   5273* 5276   5290* 5293   5300*  5304
                        5504   5510*  5512*  5514*  5515*  5516* 5517   5535*
R6    =%000006           372#   374   4616*  4617*  4618
R7    =%000007           373#   375
SAVR0     001230         538#
SAVR1     001232         539#
SAVR4     001234         540#
SAVR5     001236         541#
SP    =%000006           374#  4620*  4630*  4640*  4644   4647* 4651   4653  4654   4671* 4675   4677   4678
                        4726   4783   4876*  4924*  4929*  4941* 4962*  4966* 4975*  5008  5025*  5032   5039*
                        5053   5072*  5077*  5086*  5171*  5207* 5219*  5230* 5236*  5244* 5260*  5320*  5325
                        5327   5328   5332*  5341   5342   5344  5346*  5401* 5402   5403* 5405   5406   5407*
                        5410   5412*  5414*  5420   5432*  5433* 5436*  5437* 5446*  5451  5462   5463*  5464*
                        5465*  5494*  5495   5496   5497*  5502* 5503*  5504* 5510   5535  5536   5537   5538*
```

```
                    5539*  5560*  5563   5565   5566   5570   5573*  5588   5609*  5614*  5635   5639*  5656*
                    5657*  5658*  5659*  5660*  5662   5665*  5673*  5674   5676   5677*  5679   5680   5681
                    5693*  5694
SPACE3  016410      5317#
SRV.A   014446      5019   5052#
SRV.D   014344      5014   5023#
SRV.L   014460      5011   5061#
SRV.R   014672      5017   5116#
STACK = 001100       355#  4620   4630   4876   5171   5207   5230   5260   5332   5346
START   011102      4630#  4633
STKLMT= 177774       360#
SWR   = 177570       362#   363   4840   4851   4877   4882   4950   5098   5137   5146   5172   5176   5555*
                    5591   5595
SW0   = 000001       403#
SW00  = 000001       393#   403
SW01  = 000002       392#   402
SW02  = 000004       391#   401
SW03  = 000010       390#   400
SW04  = 000020       389#   399
SW05  = 000040       388#   398
SW06  = 000100       387#   397
SW07  = 000200       386#   396
SW08  = 000400       385#   395   5551   5569
SW09  = 001000       384#   394    485   4627   5551   5567   5569   5571   5575   5581   5598   5599
SW1   = 000002       402#
SW10  = 002000       383#   485   5581   5587   5599
SW11  = 004000       382#   485   4627   5551   5569   5571   5575
SW12  = 010000       381#
SW13  = 020000       380#  5580   5591
SW14  = 040000       379#  5550   5555
SW15  = 100000       378#  5579   5595
SW2   = 000004       401#
SW3   = 000010       400#
SW4   = 000020       399#
SW5   = 000040       398#
SW6   = 000100       397#
SW7   = 000200       396#
SW8   = 000400       395#
SW9   = 001000       394#
TABLE   012334      4629*  4655*  4663*  4681*  4689*  4697*  4705*  4713*  4730*  4736*  4742*  4748*  4754*
                    4760*  4766*  4772*  4777   4799#  4824   4911   4918   4934   5004   5062   5121
TAG.A   015524      5210   5254#  5285
TBITVE= 000014       436#
TEMP3   001224       536#  4836*  4837   4916*  4924   4932*  4958*  4962   4970*  5068*  5072   5083*  5133*
                    5134   5215*  5219   5247*  5267*  5717
TEMP4   001226       537#  4835*  4837   4874*  4917*  4929   4933*  4941   4959*  4966   4971*  4975   5023*
                    5025   5026   5037*  5039   5067*  5077   5082*  5086   5132*  5134   5169*  5233*  5236
                    5241*  5244   5268*  5275*  5292*  5302*  5717   5720
TEMP5   001222       535#  4914*  4946*  4956*  4981*  5066*  5094*
TKVEC = 000060       443#
TPVEC = 000064       444#
TRAPVE= 000034       442#  4625*  4626*
TRTVEC= 000014       437#
TYPE  = 104400      4719   4724   4779   4781   4797   4805   4912   4913   4922   4926   4939   4954   4955
```

```
                    4961    4965    4973    5006    5020    5024    5029    5038    5044    5064    5065    5070    5075
                    5080    5088    5103    5104    5216    5217    5221    5224    5225    5235    5239    5242    5336
                    5354    5355    5454    5457    5461    5529    5594    5608    5625    5627    5630    5632    5636
                    5643    5709#
TYPOC = 104402      4925    4930    4942    4963    4967    4976    5027    5040    5073    5078    5087    5220    5237
                    5245    5616    5640    5710#
TYPON = 104406      5712#
TYPOS = 104404      5711#
VERS    013265      4899#
VERSON  016652      4658*   4666*   4684*   4692*   4700*   4708*   4716*   4776*   5355    5370#
XHOLD   014270      5006#   5021    5043    5045    5054    5108
XSTORE  016570      5341*   5351#   5721
X.X.    012246      4782#
X1      016650      5357    5362    5369#
$BDADR  001122       473#
$BDDAT  001126       475#   4834*   5131*   5266*   5274*   5291*   5301*   5717
$CMTAG  001100       461#   4616    4623    4627
$CM3  = 000000       485#
$CRLF   001153       486#   5470    5594    5599    5608    5627    5632    5636
$ENDAD  016640       454    5364#
$ERFLG  001103       464#   5552    5571*   5575    5584*   5599
$ERMAX  001115       470#   5575
$ERROR  017466      4623    5583#
$ERRPC  001116       471#   5588*   5589*   5590    5599    5614    5717    5720    5721
$ERRTB  001156       504#   5622
$ERRTY  017560      5593    5607#
$ERTTL  001112       468#   5587*   5599
$FILLC  001150       483#   5410    5422
$FILLS  001147       482#   5422
$GDADR  001120       472#
$GDDAT  001124       474#   4833*   5130*   5265*   5717
$HD   = 000003       552     553
$HIOCT  020014      5678*   5683#
$ICNT   001104       465#
$ITEMB  001114       469#   5590*   5599    5611
$LF     001154       487#   5461    5470    5599
$LPADR  001106       466#   4627*   4823*   4862*   5120*   5158*   5570*   5573    5575
$LPERR  001110       467#
$NULL   001146       481#   5412    5422
$OCNT   017366      5501*   5530*   5543#
$OMODE  017370      5496*   5500*   5505    5508*   5519*   5545#
$OVER   017452      5556    5567    5572#
$PASS   001100       462#
$QUES   001152       485#   5454    5470    5599
$RDCHR  016772      5432#   5713
$RDDEC= ****** U    5716
$RDLIN  017026      5446#   5714
$RDOCT  017714      5656#   5715
$RDSZ = 000010      5439#
$SAVRE= ****** U    5716
$SCOPE  017372      4621    5554#
$SETUP= 000007      4616#   4621    4623    4625    4627
$STUP = 177777      4616#
$SVLAD  017436      5564    5569#
```

```
$SWR  = 160000          485    552    553#   4627   5550   5551   5555   5567   5569   5571   5575   5579   5580
                       5581   5587   5591   5595   5598   5599
$TKB    001140          478#   5431   5436
$TKS    001136          477#   4618   5431   5434
$TN   = 000001          552#
$TPB    001144          480#   5420*  5422
$TPFLG  001151          484#   5397   5422
$TPS    001142          479#   5418   5422
$TRAP   020016         4625    5693#
$TRP  = 000016         5700#   5710#  5711#  5712#  5713#  5714#  5715#  5716#
$TRPAD  020036         5697    5708#
$TSTNM  001102          463#   5552   5569*  5572   5575   5586   5599
$TTYIN  017134         5447    5448   5465   5469#
$TYPBN= ****** U       5713
$TYPDS= ****** U       5713
$TYPE   016664         5397#   5700   5709
$TYPOC  017170         5499#   5710
$TYPON  017204         5498    5501#  5712
$TYPOS  017144         5494#   5711
$XTSTR  017400         5558#
$OFILL  017367         5495*   5499*  5509   5544#
.     = 020542            1     342#   346    348#   453#   456#   459#   488    555#   4607#  4608#  4609#  4610#
                       4619    4627   4638   4807   4899#  4987#  5315#  5317#  5359   5373#  5422   5431   5469#
                       5470    5575   5599   5646#  5716#  5722#
.PFAIL  016422         5319#   5331
```

```
COMMEN      1#    446#
ENDCOM      1#    446#
ERROR     356#   4839   4875   5136   5170   5271   5278   5295   5306   5345   5349
ESCAPE      1#    446#
MULT        1#    446#
NEWTST      1#    446#
POP         1#    446#   5679
PUSH        1#    446#   5658
SCOPE     357#   4847   4889   5143   5183
SETTRA   5700#   5710   5711   5712   5713   5714   5715
SETUP       1#    446#   4616
SKIP        1#    446#
SLASH       1#    446#
SPACE     446#
STARS       1#    446#    488   5375   5422   5439   5470   5546   5575   5599   5647   5684
TRMTRP   5700#
TYPBIN      1#    446#
TYPDEC      1#    446#
TYPNUM      1#    446#
TYPOCS      1#    446#
TYPOCT      1#    446#   5614   5638
TYPTXT      1#    446#
$HOW      554#   4728   4734   4740   4746   4752   4758   4764   4770
$ROMTY    554#   4645   4661   4669   4687   4695   4703   4711
$$CMRE    446#
$$CMTM    446#
$$ESCA      1#    446#
$$NEWT      1#    446#
$$SET    5700#   5710   5711   5712   5713   5714   5715
$$SKIP      1#    446#
.EQUAT      1#    337#    351
.HEADE      1#    337#    542
.KT11       1#
.RDLIN    338#
.SETUP      1#    337#   4615
.SWRHI      1#    337#
.SWRLO    337#
.$CATC      1#    337#    339
.$CMTA      1#    337#    446
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#   5374#
.$ERRO      1#    338#   5575
.$ERRT      1#    338#   5599
.$MULT      1#
.$POWE      1#   5374#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#    338#   5647
.$READ      1#   5374#   5422
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
```

```
.$SCOP      1#    338#   5546
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#    337#   5684
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#   5374#   5375
.$TYPO      1#   5374#   5470
```

M12

```
          5130  5131  5132  5133  5150  5151  5157  5158  5159  5160  5167  5169  5171  5205  5207
          5208  5212  5213  5214  5215  5219  5226  5228  5230  5231  5232  5233  5236  5238  5240
          5241  5244  5254  5255  5256  5258  5260  5261  5262  5263  5264  5265  5266  5267  5268
          5272  5273  5274  5275  5286  5288  5289  5290  5291  5292  5296  5298  5299  5300  5301
          5302  5320  5321  5324  5325  5328  5331  5332  5341  5344  5346  5356  5361  5401  5402
          5406  5412  5432  5433  5446  5447  5462  5463  5464  5465  5494  5502  5503  5504  5510
          5517  5535  5536  5537  5538  5539  5560  5561  5563  5566  5570  5572  5573  5586  5588
          5609  5614  5623  5628  5633  5635  5639  5656  5657  5658  5659  5660  5662  5677  5678
          5679  5680  5681  5693  5694  5697
MOVB      5403  5420  5436  5451  5456  5495  5496  5499  5500  5501  5505  5508  5509  5528  5590
          5665  5696
NEG       5506
NOP       4643  4650  4873  5168  5229  5259  5323  5366  5367  5368
RESET     5363
ROL       5249  5281  5512  5514  5515  5516  5518  5555  5668  5670  5672
RTI       5350  5408  5438  5466  5540  5574  5598  5682
RTS       5369  5421  5637  5698
SUB       5589
TRAP      5700  5710  5711  5712  5713  5714  5715
TST       4637  4642  4649  4721  4777  4857  4868  4871  4890  4921  4937  5034  5069  5084  5091
          5100  5152  5163  5166  5184  5209  5279  5322  5327  5405  5523  5562  5595  5641  5676
          5695
TSTB      5397  5418  5434
.ASCII     485   486  4808  4899  4987  5317  5716
.ASCIZ     487  4808  4899  4987  5315  5317  5371  5645  5716
.BLKB     5469
.BLKW     4607  4608  4609  4610
.BYTE      463   464   469   470   481   482   483   484  1017  1018  1019  1020  1021  1022  5467
          5468  5541  5542  5543  5544
.ENABL       1   337
.END      5724
.ENDC      351   356   432   446   447   460   485   488   489   547  4616  4620  4621  4623  4625
          4627  4655  4663  4671  4681  4689  4697  4705  4713  5376  5423  5439  5440  5447  5449
          5454  5469  5470  5471  5547  5553  5555  5557  5568  5569  5571  5575  5576  5582  5584
          5588  5594  5595  5599  5600  5618  5647  5648  5684  5685  5694  5697  5709  5710  5711
          5712  5713  5714  5715  5716  5724
.EQUIV     356   357   359   374   375   394   395   396   397   398   399   400   401   402   403
           422   423   424   425   426   427   428   429   430   431
.EVEN     4808  4899  4987  5315  5317  5373  5646  5716
.IF        346   354   404   432   446   459   485   488   543  4616  4620  4621  4623  4625  4627
          4647  4655  4663  4671  4689  4697  4705  4713  5375  5422  5439  5447  5448  5453  5469
          5470  5546  5552  5555  5567  5569  5571  5575  5581  5584  5587  5591  5593  5594  5595
          5598  5599  5617  5633  5647  5663  5684  5693  5697  5700  5710  5711  5712  5713  5714
          5715  5716  5724
.IFF       354   447   459   485   489  4620  5376  5423  5439  5440  5449  5453  5470  5471  5547
          5567  5569  5575  5576  5581  5588  5598  5599  5600  5618  5647  5648  5685  5694
.IFT      5569  5594  5667  5683  5684
.IFTF     5569  5593  5663  5667  5683
.IIF       346   488   542   547   552   553  4621  4623  4627  5422  5431  5462  5470  5550  5551
          5571  5575  5579  5580  5581  5599  5615  5640  5709  5710  5711  5712  5713  5714  5715
.IRP      4616  5658  5679
.LIST        1   337   346   446   485   555  4607  4616  4808  4899  4987  5315  5317  5439  5700
          5709  5710  5711  5712  5713  5714  5715  5716  5724
.MACRO       1   446   554  5700
```

```
.MCALL    337     338     446    5374
.NLIST      1     337     346     446     485     555    4607    4616    4808    4899    4987    5315    5317    5439    5700
         5709    5710    5711    5712    5713    5714    5715    5716    5724
.PAGE     446     488     742    1114    1585    1636    1683    1746    1791    1842    1912    1966    2019    2073    2120
         2172    2225    3264    3833    4611    4859    5154    5315
.REM        4    2026    2076
.REPT     346
.SBTTL    337     340     347     352     448     490     555    5377    5424    5472    5548    5577    5601    5649    5686
         5701
.TITLE    542
.WORD     346     457     462     465     466     467     468     471     472     473     474     475     476    5626    5631
         5683    5717    5720    5721
```

ERRORS DETECTED: 0

*,DZBMDH.SEQ/SOL/CRF/NL:TOC=DSKZ:DZBMDH.SML,DZBMDH.P11
RUN-TIME: 12 18 1 SECONDS
CORE USED: 20K

Spooler runtime 5 Seconds, 45 KCS, 508 disk reads, 4 disk writes, 157 pages

```
  10                          ...B1   2440                     ...B5   4169                     ...B9   .TITLE     542      340      ...B13
  63.P1176       MACY11 2     ...C1   2478                     ...C5   4179    007240   001775  ...C9    **END** USER DAVIES,TOM   ...C13
 107                          ...D1   2532                     ...D5   4191                     ...D9
 147                          ...E1   2585                     ...E5   4221             ;       ...E9
 200                          ...F1   2639                     ...F5                            ...F9
 229                          ...G1   2673                     ...G5   4279    007350   177776  ...G9
 269                          ...H1   2691   005006   177570   ...H5   4311             ;       ...H9
 321                          ...I1                            ...I5                            ...I9
 345                          ...J1   2751                     ...J5   4370    007444   004000  ...J9
 399                          ...K1   2797   005102   000020   ...K5                            ...K9
                              ...L1   2810                     ...L5   4432             ;       ...L9
 455   000046   016640        ...M1   2855   005212   012711   ...M5   4486    007634   001770  ...M9
 497                          ...N1   2907   005312   100316   ...N5   4496                     ...N9

 551                          ...B2   2946                     ...B6                            ...B10
 563                          ...C2   2975                     ...C6   4559    007732   177762  ...C10
 617   001466   060011        ...D2                            ...D6   4576                     ...D10
 671                          ...E2   3032   005426   000012   ...E6                            ...E10
 725   001740   000024        ...F2   3085                     ...F6   4620    011014   012706  ...F10
 751            ;ACTUAL       ...G2   3116                     ...G6   4674    011374   001004  ...G10
 805   002004   113737 ;      ...H2   3160                     ...H6   4728    011722   022702  ...H10
 859   002132   000200 ;      ...I2   3182   005674   010045   ...I6   4782    012246   104412  ...I10
 913   002246   012700 ;      ...J2   3217   005724   005061   ...J6   4818                     ...J10
 967   002376   000005 ;      ...K2   3256   005776   000000   ...K6   4868    013032   005720  ...K10
1021   002522   .BYTE         ...L2   3273                     ...L6   4909    013334   062767  ...L10
1075   002664   100376 ;      ...M2                            ...M6   4963    013602   104402  ...M10
1123            ;173000       ...N2   3334                     ...N6   4997                     ...N10

1177   003062   100000        ...B3   3360                     ...B7   5051                     ...B11
1231   003212   177560        ...C3   3414   006076   001763   ...C7   5105    014650   016700  ...C11
1285   003334   010742        ...D3   3427   006122   005205   ...D7                            ...D11
1339                          ...E3   3481   006236   042700   ...E7   5163    015130   005720  ...E11
1393                          ...F3   3496   006260   005000   ...F7   5202                     ...F11
1447   003472   012761        ...G3   3550                     ...G7   5256    015534   012737  ...G11
1501                          ...H3   3565   006420   000000   ...H7   5310    016042   000000  ...H11
1555   003744   000207        ...I3                            ...I7           016307      015  ...I11
1594                          ...J3   3621   006506   000000   ...J7   5328    016464   012637  ...J11
1645   004004   000001 ;      ...K3   3657                     ...K7   5380                     ...K11
1692            ;             ...L3   3693              ;      ...L7   5434    017002   105777  ...L11
                              ...M3                            ...M7   5488                     ...M11
1755   004226   000340 ;      ...N3   3755   006672   010145   ...N7   5542    017365      000  ...N11

1800            ;             ...B4   3791                     ...B8   5596    017552   100001  ...B12
1851   004350   032711 ;      ...C4   3842                     ...C8   5650                     ...C12
1905   004520   030500 ;      ...D4   3869              ;      ...D8   5704                     ...D12
1921                          ...E4                            ...E8   BITO5 = 000040           ...E12
1975            ;             ...F4   3925              ;      ...F8   FLAG4   016052           ...F12
2028                          ...G4                            ...G8   PWRVEC= 000024           ...G12
2082                          ...H4                            ...H8   START   011102           ...H12
2129   004534   010037 ;      ...I4   3987   007004   013700   ...I8   VERSON  016652           ...I12
2181                          ...J4   4041   007064   173304   ...J8   $TRP  = 000016           ...J12
2234                          ...K4                            ...K8   SETTRA    5700#   5710   ...K12
2285                          ...L4   4057              ;      ...L8                            ...L12
2339                          ...M4                            ...M8   BIT       4840    4851   ...M12
2393                          ...N4   4115                     ...N8   MOVB      5403    5420   ...N12
```