BM873

RESTART ROM LOADER

MD-11-DZBMD-J

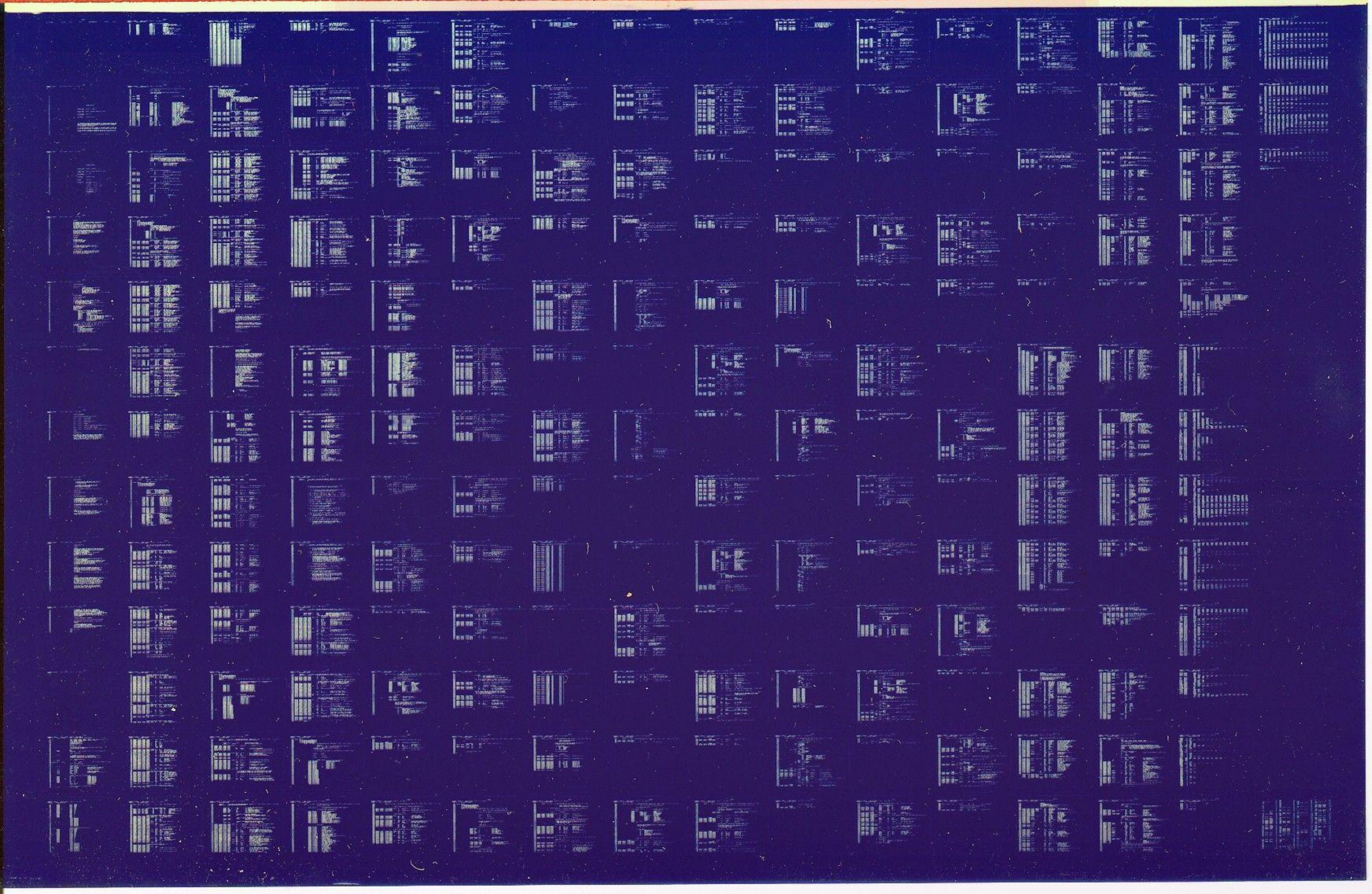
EP-DZBMD-J-DL-C

FICHE 1 OF 1

JUN 1977

digital

MADE IN USA



B01

EOF1DVDVARSEQ	00010000	770629	PDP10 411	DVHDR1DZBMDJSEQ	00010000	770629

```
                                .REM%
```

IDENTIFICATION

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

PRODUCT NAME: BM873 - UNIVERSAL RESTART ROM LOADER

DATE RELEASED: MARCH 1977.

MAINTAINER: DIAGNOSTIC GROUP

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 FITZCARL JOHNSON	10/76	YH
	ED RYAN	3/77	YJ

```
                                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 RUM, 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 OR DISKETTE
                                        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, H OR J, 256 WORDS
                                                        ARE AUTOMATICALLY CHECKED.

                                4.2     STARTING ADDRESS

                                        STARTING ADDRESS 000200

                                4.3     OPERATOR ACTION

                                4.3.1   FOR NORMAL OPERATION (WITHOUT EXTENDED 128 WORDS)
                                        1. LOAD STARTING ADDRESS (000200)
                                        2. SET SWITCHES AS PER 5.1.1 (NORMAL ALL SWITCHES DOWN)
                                        3. PRESS START SWITCH AND RELEASE.
                                           (11/34 PRESS CNTRL START SIMULTANEOUSLY)

                                        WHEN PROGRAM IS STARTED FOR THE FIRST TIME THE FOLLOWING
                                        WILL BE PRINTED OUT:
                                        MAINDEC-11-DZBMD-J
                                        DEVICE VERSION
                                        BM873-Y
                                         THE OPERATOR WILL THEN SPECIFY THE VERSION BEING RUN.
                                                BM873-Y*        IS ANY NON-STANDARD VERSION.
                                                NOTE: PROGRAM TWO MUST BE RUN FIRST.
                                                BM873-YA        REPLACES M792-YA,MR11-DB,M792-YH
                                                BM873-YB        MASSBUS
                                                BM873-YC        DDCMP BOOTSTRAP ROM
                                                BM873-YD        KL10 (PDP-11) 256 BOOTSTRAP ROM (VERSION 2(17))
                                                BM873-YF        KL10 (PDP-11) 256 BOOTSTRAP ROM (VERSION 3(23))
                                                BM873-YG        KL10 (PDP-11) 256 BOOTSTRAP ROM
                                                BM873-YH        KL10 (PDP-11) 256 BOOTSTRAP ROM
                                               BM873-YJ        DECSYSTEM20 SECONDARY FRONT END (PDP-1134A)
                                                               256 BOOTSTRAP ROM

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

                                4.3.2   IF YOU WISH TO TEST THE EXTENDED 128. WORDS THIS IS THE
                                        PROCEDURE:
                                        (NOT NEEDED FOR NORMAL TESTING OF BM873Y-B,C,D,F,G,H OR J)
                                        1.      LOAD STARTING ADD. 000200
                                        2.      SET SW00=1
                                        3.      SET HALT ENABLE SW AND SINGLE CYCLE SW UP
                                        4.      HIT START SWITCH AND RELEASE.
                                        5.      RUN PROGRAM 2 FOR ONE PASS.
                                        6.      NOW ANY PROGRAM MAY BE RUN.
                                        NOTE:   VISUAL INSPECTION OF EXTENDED DUMP
```

```
                                        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 ERROR

                                6.      ERRORS

                                6.1     ERROR PRINT OUT

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

```
                        6.2     ERROR RECOVERY

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

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

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

                        7.      RESTRICTIONS

                        7.1     STARTING RESTRICTIONS

                                SEE SECTION 4.

                        7.2     OPERATING RESTRICTIONS

                        7.2.1   IF YOU WISH PROGRAM TO TEST YOUR EXTENDED 128. WORDS;
                                YOU MUST START AS PER SECTION 4 AND THEN
                                ****** RUN PROGRAM 2 FIRST AND VISUALLY VERIFY DATA.****
                                (NOT APPLICABLE TO BM873Y-B,C,D,F,G,H OR J)
                        7.2.2   YOU MAY NOT ALTER THE SOFTWARE MAP UNLESS--
                                ****** YOU KNOW WHAT YOU ARE DOING *****

                        7.2.3   THE ROM ADDRESS MUST START AT 173000 AND BE AT  LEAST
                                128 WORDS LONG. (256 FOR THE BM873Y-B,C,D,F,G,H OR J)

                        8.      MISCELLANEOUS

                        8.1     EXECUTION TIME

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

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,H OR J IS TESTED.

9.2 PROGRAM 2

PROGRAM 2 WILL DUMP THE CONTENTS OF THE ROM ONTO THE
TTY. NOTE NO VERIFICATION
OF ANY KIND IS PERFORMED ON THE DATA.(AN ERROR WILL OCCUR
IF A TRAP IS ENCOUNTERED WHILE READING) YOU MUST INSPECT
THE DATA YOUR SELF. IF SW00 WAS UP WHEN START WAS HIT
THE EXTENDED 128. WORDS WILL BE PRINTED.
256 WORDS WILL BE PRINTED IF BM873Y-B,C,D,F,G,H OR J IS SELECTED.

9.3 PROGRAM 3

PROGRAM 3 IS THE SAME AS PROGRAM ONE EXCEPT THAT THE
USER HAS THE ABILITY TO ALTER THE SOFTWARE MAP,
LIST OR PRINT THE SOFTWARE MAP, AND RUN THE PROGRAM.
NOTE THAT IF YOU ALTER THE MAP BE CAREFULL OF WHAT
YOU CHANGE.
FOR THE COMMANDS TO BE USED SEE TOP OF PROGRAM 3 IN THIS LISTING

9.4 PROGRAM 4

PROGRAM 4 CHECKS THE OFFSET ADDRESS WHEN THE SIMULATED
PUSHING OF A BUTTON IS DONE BY THE SOFTWARE. ON THE FIRST PASS
THE OFFSET IS TYPED OUT FOR YOU TO VERIFY (NOTE:
THE PROGRAM HAS NO WAY OF KNOWING WHAT THE OFFSET WILL
BE). AFTER THE DATA IS TYPED OUT IT IS STORED AWAY IN
CORE. WHEN THE FIRST PASS IS FINISHED THE PROCESS IS
REPEATED ONLY NO TYPE OUT IS PERFORMED, AND THE DATA
IN CORE IS COMPARED TO THE DATA FOUND AT THE ROM.

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

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

                        9.6     ELECTRICAL 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
```

342 %

```
                                .MCALL .HEADER, .SWRHI, .SWRLO, .EQUATE, .SETUP, .STRAP, .SCATCH, .SCMTAG
                                .MCALL .RDLIN, .$SCOPE, .$ERROR, .$ERRTYP, .$RDOCT
                                .TITLE  MAINDEC-11-DZBMD-J
                                ;*COPYRIGHT (C) 1977
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY E. RYAN
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-A1).
                                ;*
000001                          $TN=1
160000                          $SWR=160000     ;HALT ON ERROR, LOOP ON TEST, INHIBIT ERROR TYPOUT

                                .SBTTL  TRAP CATCHER

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

                                .SBTTL  STARTING ADDRESS(ES)
000200                                  .=200

000200  000137  012000                  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
000064                          TPVEC=  64              ;TTY PRINTER VECTOR
000240                          PIRQVEC=240             ;PROGRAM INTERRUPT REQUEST VECTOR
```

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

                                .SBTTL  COMMON TAGS

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

000046                                  .=46
000046  020002                  $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 SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:         IF SITEMB 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  021216                         EM1             ;"ROM READ DATA COMPARISON ERROR"
001160  021416                         DH1             ;*
001162  021614                         DT1             ;*
001164  000000                         0               ;* PRINT ALL NUMERIC DATA IN OCTAL

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 1

001166  021260                         EM2             ;"WRITTING ROM FAILED TO TRAP"
001170  021515                         DH2
001172  021630                         DT2
001174  000000                         0               ;PRINT ALL NUMERIC DATA IN OCTAL.

                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 2

001176  021316                         EM3             ;"UNEXPECTED TRAP WHILE READING ROM"
001200  021547                         DH3
001202  021636                         DT3
001204  000000                         0
                                ;ERROR TABLE ITEM FOR ERROR MESSAGE 3

001206  021356                         EM4             ;"FATAL TRAP. ROM PC ON STACk."
001210  021515                         DH2
001212  021630                         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                 SAVP5:  0
```

```
                        .=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
001602  005711  ;173202 005711          TST @R1         ;THEN TEST FOR ERROR
001604  100441  ;173204 100441          BMI ERROR       ;GOT PROBLEMS
001606  005007  ;173206 005007          CLR PC          ;AND TRANSFER TO ZERO
                                        ;
                                        ;THIS IS THE STARTING ADDRESS FOR THE PC11 PAPER TAPE CONTROLLER
001610  012704  ;173210 012704  KL11:   MOV #177560,R4  ;OBTAIN DEVICE ADDRESS
001612  177560  ;173212 177560
001614  000440  ;173214 000440          BR CKDEV        ;AND TRANSFER TO READER SERVICE ROUTINE
                                        ;
                                        ;THIS IS THE CASSETTE DEVICE COMMAND TABLE
001616  017640  ;173216    240  TABLE:   .BYTE 240       ;COMPARE WORD NOT A COMMAND
                ;173217    037          .BYTE 37        ;ILBS+RWD+GO
001620  002415  ;173220    015          .BYTE 15        ;SPACE FORWARD BLOCK+GO
                ;173221    005          .BYTE 5         ;READ+GO
001622  112024  ;173222    024          .BYTE 24        ;READ+ILBS
                ;173223    224          .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
001624  173000  ;173224 173000  POWER2: .WORD RF11      ;ADDRESS OF BEGINNING OF BOOTSTRAP
001626  000340  ;173226 000340          .WORD 340       ;PRIORITY LEVEL 7
                                        ;
                                        ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE DEVICE #0
001630  005004  ;173230 005004  CBOOT:   CLR R4          ;LOAD DEVICE NUMBER 0 IN R4
001632  012700  ;173232 012700  RESTX:   MOV #177500,R0  ;GET DEVICE ADDTESS
001634  177500  ;       177500
001636  000005  ;173236 000005  RESTRT:  RESET           ;ISSUE RESET INSTRUCTION
001640  010410  ;173240 010410          MOV R4,@R0      ;LOAD DEVICE WITH UNIT NUMBER
001642  012701  ;173242 012701          MOV #TABLE,R1   ;GET FUNNY TABLE OF INSTRUCTIONS
001644  173216  ;173244 173216
001646  012702  ;173246 012702          MOV #375,R2     ;AND LOAD UP TRANSFER COUNTER
001650  000375  ;173250 000375
001652  112103  ;173252 112103          MOVB (R1)+,R3   ;THE LOAD UP COMPARATOR
001654  112110  ;173254 112110  LOOP1:   MOVB (R1)+,@R0  ;LOAD DEVICE REGISTER WITH COMMAND
001656  100407  ;173256 100407          BMI DONE
001660  130310  ;173260 130310  LOOP2:   BITB R3,@R0     ;HAS COMMAND COMPLETED
001662  001776  ;173262 001776          BEQ LOOP2       ;NO, WAIT
001664  105202  ;173264 105202          INCB R2         ;THEN INCREMENT ADDRESS CTR
001666  100772  ;173266 100772          BMI LOOP1       ;IF NEGATIVE, GET COMMAND
001670  116012  ;173270 116012          MOVB 2(R0),@R2  ;AND STORE DATA AWAY
001672  000002  ;173272 000002
001674  000771  ;173274 000771          BR LOOP2        ;GO GET ANOTHER BYTE
001676  005710  ;173276 005710  DONE:    TST @R0         ;ANY DEVICE ERRORS
001700  100756  ;173300 100756          BMI RESTRT      ;YES, RETRY
001702  005002  ;173302 005002          CLR R2          ;CLEAR COMPARE ADDRESS AND TRANSFER ADDRESS
001704  120312  ;173304 120312          CMPB R3,@R2     ;IT MUST BE 240
001706  001377  ;173306 001377          BNE .+0         ;NO, THERE WAS AN ERROR
001710  000112  ;173310 000112  ERROR:   JMP @R2         ;NORMAL CASSETTE AND ERROR FOR BULK STORAGE
                                        ;
                                        ;THIS IS THE STARTING LOCATION FOR THE PC11 CONTROLLER
001712  012704  ;173312 012704  PC11:    MOV #177550,R4  ;LOAD DEVICE ADDRESS
001714  177550  ;173314 177550
```

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

```
                        ; BM8738        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
                        ;               ;BM873A 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      TAPE$
                                ;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
```

```
  964  002354  113705 ;173354  113705          MOVB    @#CSW,R5
  965  002356  177570 ;173356  177570
  966  002360  000305 ;173360  000305          SWAB    R5      ;GET UNIT # INTO HIGH BY
  967  002362  000402 ;173362  000402          BR      3$
  968  002364  010703 ;173364  010703  1$:     MOV     PC,R3
  969  002366  005005 ;173366  005005          CLR     R5
  970  002370  010702 ;173370  010702  3$:     MOV     PC,R2
  971  002372  000403 ;173372  000403          BR      OTHER
  972  002374  176716 ;173374  176716          .WORD   RPWC
  973  002376  000005 ;173376  000005          .WORD   RFREAD
  974                  ;                       ;
  975  002400  005005 ;173400  005005  OTHERA: CLR     R5      ;SET TO UNIT 0
  976  002402  010200 ;173402  010200  OTHER:  MOV     R2,R0   ;R0 POINT AT WORD COUNT
  977  002404  005720 ;173404  005720          TST     (R0)+   ;POINT TO PARAMETER LIST
  978  002406  012001 ;173406  012001          MOV     (R0)+,R1;MOVE WORD COUNT ADDRESS
  979  002410  012711 ;173410  012711          MOV     #-256.*2,(R1);LOAD WORD COUNT
  980  002412  177000 ;173412  177000
  981  002414  051005 ;173414  051005          BIS     (R0),R5 ;COMBINE UNIT # WITH COM
  982  002416  010541 ;173416  010541          MOV     R5,-(R1);LOAD READ COMMAND
  983  002420  032711 ;173420  032711          BIT     #100200,(R1);CHECK FOR ERROR AND
  984  002422  100200 ;173422  100200
  985  002424  001775 ;173424  001775          BEQ     .-4     ;WAIT UNTIL COMPLETE
  986  002426  100012 ;173426  100012          BPL     1$      ;NO ERROR
  987  002430  005702 ;173430  005702          TST     R2      ;WAS IT CALLED BY MASS B
  988  002432  001024 ;173432  001024          BNE     AGAIN   ;NO, ERROR
  989  002434  032761 ;173434  032761          BIT     #TUTAPE,26(R1);IS TU16?
  990  002436  040000 ;173436  040000
  991  002440  000026 ;173440  000026
  992  002442  001420 ;173442  001420          BEQ     AGAIN   ;NO. ERROR
  993  002444  022761 ;173444  022761          CMP     #FCE,14(R1);ARE WE READ A SHORT
  994  002446  001000 ;173446  001000
  995  002450  000014 ;173450  000014
  996  002452  001014 ;173452  001014          BNE     AGAIN   ;SOME OTHER ERROR
  997  002454  005007 ;173454  005007  1$:     CLR     PC      ;O.K.
  998                  ;                       ;
  999                  ;                       ;THIS IS THE TAPE DEVICE SERVICE ROUTINE
 1000  002456  010200 ;173456  010200  TAPES:  MOV     R2,R0   ;GET THE ADDRESS OF THE
 1001  002460  005720 ;173460  005720          TST     (R0)+   ;STEP TO LAST COMMAND
 1002  002462  012001 ;173462  012001          MOV     (R0)+,R1;GET THE WORD COUNT ADDR
 1003  002464  005311 ;173464  005311          DEC     (R1)    ;SET UP TO ADVANCE 1 REC
 1004  002466  005720 ;173466  005720          TST     (R0)+   ;MOVE R0 TO FIRST COMMAN
 1005  002470  012041 ;173470  012041          MOV     (R0)+,-(R1);LOAD COMMAND REG.
 1006  002472  031011 ;173472  031011          BIT     (R0),(R1);DONE?
 1007  002474  001776 ;173474  001776          BEQ     .-2     ;NO. KEEP LOOPING
 1008  002476  005720 ;173476  005720          TST     (R0)+   ;YES. CHECK FOR ERROR
 1009  002500  031041 ;173500  031041          BIT     (R0),-(R1);ANY ERROR?
 1010  002502  001736 ;173502  001736          BEQ     OTHERA  ;NO ERROR- TRY TO READ
 1011  002504  000005 ;173504  000005  AGAIN:  RESET
 1012                  ;                       ;
 1013  002506  000113 ;173506  000113          JMP     (R3)    ;ERROR RETURN
 1014                  ;                       ;
 1015                  ;                       ;THIS IS THE STARTING ADDRESS FOR PC11 PAPER TAP
 1016  002510  012704 ;173510  012704  KL11:   MOV     #KLCS,R4;OBTAIN CONTROL REG.
 1017  002512  177560 ;173512  177560
```

```
1018  002514  000443 ;173514  000443          BR      CKDEV   ;AND TRANSFER TO READER
1019          ;
1020          ;                 ;
1021          ;
1022          ;                 ;CASSETTE TAPE DEVICE COMMAND TABLE
1023  002516  .BYTE   240 ;173516        240  TABLE:  .BYTE   240     ;COMPARE WORD NOT A COMM
1024  002517  .BYTE   037 ;173517        037          .BYTE   37      ;ILBS+RWD+GO
1025  002520  .BYTE   015 ;173520        015          .BYTE   15      ;SPACE FORWARD BLOCK+GO
1026  002521  .BYTE   005 ;173521        005          .BYTE   5       ;READ
1027  002522  .BYTE   024 ;173522        024          .BYTE   24      ;READ +ILBS
1028  002523  .BYTE   224 ;173523        224          .BYTE   224     ;READ+ILBS+END FLAG
1029          ;                 ;
1030          ;                 ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE D
1031  002524  000404 ;173524  000404  CBOOTA: BR      1$      ;SELECT UNIT 0
1032  002526  113704 ;173526  113704  CBOOTB: MOVB    @#CSW,R4;SELECT UNITS
1033  002530  177570 ;173530  177570
1034  002532  000304 ;173532  000304          SWAB    R4
1035  002534  000401 ;173534  000401          BR      RESETX
1036  002536  005004 ;173536  005004  1$:     CLR     R4
1037  002540  012700 ;173540  012700  RESETX: MOV     #TACS,R0;GET CONTROL REG.
1038  002542  177500 ;173542  177500
1039  002544  000005 ;173544  000005  RESTRT: RESET
1040  002546  010410 ;173546  010410          MOV     R4,(R0) ;SELECT UNIT
1041  002550  012701 ;173550  012701          MOV     #TABLE,R1
1042  002552  173516 ;173552  173516
1043  002554  012702 ;173554  012702          MOV     #375,R2 ;LOAD TRANSFER COUNTER
1044  002556  000375 ;173556  000375
1045  002560  112103 ;173560  112103          MOVB    (R1)+,R3;LOAD COMPARATOR
1046  002562  112110 ;173562  112110  LOOP1:  MOVB    (R1)+,(R0);LOAD COMMAND
1047  002564  100407 ;173564  100407          BMI     DONE
1048  002566  130310 ;173566  130310  LOOP2:  BITB    R3,(R0) ;COMMAND COMPLETE?
1049  002570  001776 ;173570  001776          BEQ     LOOP2   ;NO. WAIT
1050  002572  105202 ;173572  105202          INCB    R2      ;INCREMENT ADDRESS CTR.
1051  002574  100772 ;173574  100772          BMI     LOOP1   ;IF (-), GET COMMAND
1052  002576  116012 ;173576  116012          MOVB    2(R0),(R2);STORE DATA
1053  002600  000002 ;173600  000002
1054  002602  000771 ;173602  000771          BR      LOOP2   ;GET ANOTHER BYTE
1055  002604  005710 ;173604  005710  DONE:   TST     (R0)    ;ANY ERROR?
1056  002606  100756 ;173606  100756          BMI     RESTRT  ;YES, RETRY
1057  002610  005002 ;173610  005002          CLR     R2      ;CLEAR COMPARE ADDRESS
1058  002612  120312 ;173612  120312          CMPB    R3,(R2) ;IT MUST BE 240
1059  002614  001377 ;173614  001377          BNE     .
1060  002616  000112 ;173616  000112  ERROR:  JMP     (R2)
1061          ;                 ;
1062          ;                 ;THIS IS THE STARTING ADDRESS FOR THE PC11 CONTR
1063  002620  012704 ;173620  012704  PC11:   MOV     #PCCS,R4
1064  002622  177550 ;173622  177550
1065  002624  000005 ;173624  000005  CKDEV:  RESET
1066  002626  012701 ;173626  012701          MOV     #160000,R1;SET UP MEMORY TEST LI
1067  002630  160000 ;173630  160000
1068  002632  012702 ;173632  012702          MOV     #6,R2   ;SET UP POINTER TO TIMEO
1069  002634  000006 ;173634  000006
1070  002636  012712 ;173636  012712          MOV     #340,(R2);SET UP VECTOR TO RETUR
1071  002640  000340 ;173640  000340
```

```
1072  002742  010742  ;173642  010742          MOV     PC,-(R2);SAVE PC
1073  002744  012706  ;173644  012706          MOV     #24,SP  ;LOAD UP STACK POINTER
1074  002746  000024  ;173646  000024
1075  002750  010441  ;173650  010441          MOV     R4,-(R1);LOOK FOR END OF MEMORY
1076  002752  040601  ;173652  040601          BIC     SP,R1   ;THEN DROP TO XX752
1077  002754  010111  ;173654  010111          MOV     R1,(R1) ;AND STORE IN ITSELF
1078  002756  011102  ;173656  011102  LOOP:   MOV     (R1),R2
1079  002760  005214  ;173660  005214          INC     (R4)    ;START DEVICE
1080  002762  105714  ;173662  105714  FORMAT: TSTB    (R4)    ;WAIT
1081  002764  100376  ;173664  100376          BPL     FORMAT
1082  002766  116412  ;173666  116412          MOVB    2(R4),(R2);SAVE THE DATA
1083  002770  000002  ;173670  000002
1084  002772  005211  ;173672  005211          INC     (R1)
1085  002774  120227  ;173674  120227          CMPB    R2,#375
1086  002776  000375  ;173676  000375
1087  002700  001366  ;173700  001366          BNE     LOOP    ;NO
1088  002702  105222  ;173702  105222          INCB    (R2)+   ;YES
1089  002704  000142  ;173704  000142          JMP     -(R2)
1090  002706  000000  ;173706  000000  ;THIS AREA IS UNUSED
1091  002710  000000  ;173710  000000  ;THIS AREA IS UNUSED
1092  002712  000000  ;173712  000000  ;THIS AREA IS UNUSED
1093  002714  000000  ;173714  000000  ;THIS AREA IS UNUSED
1094  002716  000000  ;173716  000000  ;THIS AREA IS UNUSED
1095  002720  000000  ;173720  000000  ;THIS AREA IS UNUSED
1096  002722  000000  ;173722  000000  ;THIS AREA IS UNUSED
1097  002724  000000  ;173724  000000  ;THIS AREA IS UNUSED
1098  002726  000000  ;173726  000000  ;THIS AREA IS UNUSED
1099  002730  000000  ;173730  000000  ;THIS AREA IS UNUSED
1100  002732  000000  ;173732  000000  ;THIS AREA IS UNUSED
1101  002734  000000  ;173734  000000  ;THIS AREA IS UNUSED
1102  002736  000000  ;173736  000000  ;THIS AREA IS UNUSED
1103  002740  000000  ;173740  000000  ;THIS AREA IS UNUSED
1104  002742  000000  ;173742  000000  ;THIS AREA IS UNUSED
1105  002744  000000  ;173744  000000  ;THIS AREA IS UNUSED
1106  002746  000000  ;173746  000000  ;THIS AREA IS UNUSED
1107  002750  000000  ;173750  000000  ;THIS AREA IS UNUSED
1108  002752  000000  ;173752  000000  ;THIS AREA IS UNUSED
1109  002754  000000  ;173754  000000  ;THIS AREA IS UNUSED
1110  002756  000000  ;173756  000000  ;THIS AREA IS UNUSED
1111  002760  000000  ;173760  000000  ;THIS AREA IS UNUSED
1112  002762  000000  ;173762  000000  ;THIS AREA IS UNUSED
1113  002764  000000  ;173764  000000  ;THIS AREA IS UNUSED
1114  002766  000000  ;173766  000000  ;THIS AREA IS UNUSED
1115  002770  000000  ;173770  000000  ;THIS AREA IS UNUSED
1116  002772  000000  ;173772  000000  ;THIS AREA IS UNUSED
1117  002774  000000  ;173774  000000  ;THIS AREA IS UNUSED
1118  002776  END.YB:
1119  002776  000000  ;173776  000000  ;THIS AREA IS UNUSED
```

```
1120  003000  MAP.YC:
1121          ;THE FOLLOWING 1000 LOCATIONS ARE
1122          ;A REPRODUCTION OF THE ROM PROGRAM
1123          ;FOR THE BM873YC. THE FIRST 400 LOCATIONS
1124          ;ARE AN EXACT COPY OF THE BM873YA. THE
1125          ;REMAINING 400 LOCATIONS ARE
1126          ;THE DDCMP BOOTSTRAP ROM PROGRAM.
1127          ;IT IS HERE FOR COMPARISON TO
1128          ;ACTUAL ROM AND FOR REFERENCE.
1129          ;173000 .=173000          ;STARTING ADDRESS FOR BOOTSTRAP
1130                                    ;THIS LOADER IS DESIGNED FOR THE RESTART MODULE M873.
1131                                    ;IT FUNCTIONALLY REPLACES THE FOLLOWING ROMS:
1132                                    ;M792-YA - PAPER TAPE BOOTSTRAP FOR PC11,KL11
1133                                    ;MR11-DB   BULK STORAGE BOOTSTRAP ROM
1134                                    ;M792-YH   TA11 CASSETTE BOOTSTRAP ROM
1135          ;               000000  R0=     %0        ;REGISTER DEFINITIONS
1136          ;               000001  R1=     %1
1137          ;               000002  R2=     %2
1138          ;               000003  R3=     %3
1139          ;               000004  R4=     %4
1140          ;               000005  R5=     %5
1141          ;               000006  SP=     %6
1142          ;               000007  PC=     %7
1143          ;               177570  SR=     177570          ;PROCESSOR SWITCH REGISTER
1144                                    ;
1145                                    ;STARTING LOCATION FOR RF11 DISK
1146  003000  010702  ;173000 010702  RF11:   MOV PC,R2        ;SET POINTER TO PARAMETER LISTS
1147  003002  000464  ;173002 000464         BR OTHER        ;TRANSFER TO SERVICE ROUTINE
1148  003004  177462  ;173004 177462         .WORD 177462    ;DEVICE WORD COUNT ADDRESS
1149  003006  000005  ;173006 000005         .WORD 5         ;DEVICE READ INSTRUCTION
1150                                    ;
1151                                    ;THIS IS THE STARTING LOCATION FOR THE RK11 CONTROLLER
1152  003010  010702  ;173010 010702  RK11:   MOV PC,R2        ;SET POINTER TO PARAMETER LIST
1153  003012  000460  ;173012 000460         BR OTHER        ;TRANSFER TO SERVICE ROUTINE
1154  003014  177406  ;173014 177406         .WORD 177406    ;DEVICE WORD COUNT REGISTER
1155  003016  000005  ;173016 000005         .WORD 5         ;DEVICE READ INSTRUCTION
1156                                    ;
1157                                    ;THIS IS A SPARE STARTING LOCATION. IT TRANSFERS TO ADDRESS
1158                                    ;CONTAINED IN THE SWITCH REGISTER.
1159  003020  013707  ;173020 013707  TRANSR. MOV @#SR,PC     ;GO TO INDICATED LOCATION
1160  003022  177570  ;173022 177570
1161          ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
1162                                    ;
1163                                    ;THIS IS THE POWER UP VECTOR REQUIRED FOR DEVICE AND
1164  003024  173000  ;173024 173000  POWER:  .WORD RF11       ;ADDRESS OF FIRST LOCATION IN ROM
1165  003026  000340  ;173026 000340         .WORD 340       ;PROCESSOR STATUS LEVEL 7
1166                                    ;
1167                                    ;THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE) CONTROLLER.
1168  003030  010702  ;173030 010702  TC11:   MOV PC,R2        ;SET UP POINTER TO PARAMETER LIST
1169  003032  000426  ;173032 000426         BR TAPES        ;AND TRANSFER TO FIRST ROUTINE
1170  003034  177344  ;173034 177344         .WORD 177344    ;DEVICE WORD COUNT ADDRESS
1171  003036  004003  ;173036 004003         .WORD 4003      ;FIND PREVIOUS BLOCK COMMAND
1172  003040  100000  ;173040 100000         .WORD 100000    ;USED AS DONE INDICATOR
1173  003042  024000  ;173042 024000         .WORD 24000     ;USED AS ERROR INDICATOR/TEST FLAG
```

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

```
                ;173174 011041          MOV @R0,-(R1)   ;AND PUT COMMAND TO DEVICE
                ;173176 105711  OTHWAT: TSTB @R1        ;WAIT FOR DONE FLAG
                ;173200 100376          BPL OTHWAT      ;BY HANGING IN LOOP
                ;173202 005711          TST @R1         ;THEN TEST FOR ERROR
                ;173204 100441          BMI ERROR       ;GOT PROBLEMS
                ;173206 005007          CLR PC          ;AND TRANSFER TO ZERO
                                        ;
                                        ;THIS IS THE STARTING ADDRESS FOR THE PC11 PAPER TAPE CONTROLLER
                ;173210 012704  KL11:    MOV #177560,R4 ;OBTAIN DEVICE ADDRESS
                ;173212 177560
                ;173214 000440          BR CKDEV        ;AND TRANSFER TO READER SERVICE ROUTINE
                                        ;
                                        ;THIS IS THE CASSETTE DEVICE COMMAND TABLE
                ;173216    240  TABLE:   .BYTE 240       ;COMPARE WORD NOT A COMMAND
                ;173217    037           .BYTE 37        ;ILBS+RWD+GO
                ;173220    015           .BYTE 15        ;SPACE FORWARD BLOCK+GO
                ;173221    005           .BYTE 5         ;READ+GO
                ;173222    024           .BYTE 24        ;READ+ILBS
                ;173223    224           .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
                ;173224 173000  POWER2: .WORD RF11       ;ADDRESS OF BEGINNING OF BOOTSTRAP
                ;173226 000340          .WORD 340        ;PRIORITY LEVEL 7
                                        ;
                                        ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE DEVICE #0
                ;173230 005004  CBOOT:  CLR R4           ;LOAD DEVICE NUMBER 0 IN R4
                ;173232 012700  RESTX:  MOV #177500,R0   ;GET DEVICE ADDTESS
                ;       177500
                ;173236 000005  RESTRT: RESET            ;ISSUE RESET INSTRUCTION
                ;173240 010410          MOV R4,@R0       ;LOAD DEVICE WITH UNIT NUMBER
                ;173242 012701          MOV #TABLE,R1    ;GET FUNNY TABLE OF INSTRUCTIONS
                ;173244 173216
                ;173246 012702          MOV #375,R2      ;AND LOAD UP TRANSFER COUNTER
                ;173250 000375
                ;173252 112103          MOVB (R1)+,R3    ;THE LOAD UP COMPARATOR
                ;173254 112110  LOOP1:  MOVB (R1)+,@R0   ;LOAD DEVICE REGISTER WITH COMMAND
                ;173256 100407          BMI DONE
                ;173260 130310  LOOP2:  BITB R3,@R0      ;HAS COMMAND COMPLETED
                ;173262 001776          BEQ LOOP2        ;NO, WAIT
                ;173264 105202          INCB R2          ;THEN INCREMENT ADDRESS CTR
                ;173266 100772          BMI LOOP1        ;IF NEGATIVE, GET COMMAND
                ;173270 116012          MOVB 2(R0),@R2   ;AND STORE DATA AWAY
                ;173272 000002
                ;173274 000771          BR LOOP2         ;GO GET ANOTHER BYTE
                ;173276 005710  DONE:   TST @R0          ;ANY DEVICE ERRORS
                ;173300 100756          BMI RESTRT       ;YES, RETRY
                ;173302 005002          CLR R2           ;CLEAR COMPARE ADDRESS AND TRANSFER ADDRESS
                ;173304 120312          CMPB R3,@R2      ;IT MUST BE 240
                ;173306 001377          BNE .+0          ;NO, THERE WAS AN ERROR
                ;173310 000112  ERROR:  JMP @R2          ;NORMAL CASSETTE AND ERROR FOR BULK STORAGE
                                        ;
                                        ;THIS IS THE STARTING LOCATION FOR THE PC11 CONTROLLER
```

```
1282  003312  012704  ;173312 012704  PC11:   MOV #177550,R4  ;LOAD DEVICE ADDRESS
1283  003314  177550  ;173314 177550
1284  003316  000005  ;173316 000005  CKDEV:  RESET           ;KILL ALL DEVICE ACTION
1285  003320  012701  ;173320 012701          MOV #160000,R1  ;THEN SET UP MEMORY TEST LIMITS
1286  003322  160000  ;173322 160000
1287  003324  012702  ;173324 012702          MOV #6,R2       ;AND SET UP POINTER TO TIMEOUT LOCATION
1288  003326  000006  ;173326 000006
1289  003330  012712  ;173330 012712          MOV #340,@R2    ;AND SET UP VECTOR TO RETURN TO NEXT
1290  003332  000340  ;173332 000340
1291  003334  010742  ;173334 010742          MOV PC,-(R2)    ;SAVE THE PC
1292  003336  012706  ;173336 012706          MOV #24,SP      ;AND LOAD UP STACK POINTER
1293  003340  000024  ;173340 000024
1294  003342  010441  ;173342 010441          MOV R4,-(R1)    ;AND LOOK FOR END OF MEMORY
1295  003344  040601  ;173344 040601          BIC SP,R1       ;THEN DROP TO XX7752
1296  003346  010111  ;173346 010111          MOV R1,@R1      ;AND STORE IN ITSELF
1297  003350  011102  ;173350 011102  LOOP:   MOV @R1,R2      ;THEN LOAD ADDRESS FOR DATA INSERTION
1298  003352  005214  ;173352 005214          INC @R4         ;AND START DEVICE
1299  003354  105714  ;173354 105714  RDRWAT: TSTB @R4        ;THEN WAIT FOR CHARACTER AVAILABLE
1300  003356  100376  ;173356 100376          BPL RDRWAT      ;HANGING THERE IF NECCESSARY
1301  003360  116412  ;173360 116412          MOVB 2(R4),@R2  ;STORE AWAY DATA BYTE
1302  003362  000002  ;173362 000002
1303  003364  005211  ;173364 005211          INC @R1
1304  003366  120227  ;173366 120227          CMPB R2,#375    ;HAS BRANCH OFFSET BEEN STORED
1305  003370  000375  ;173370 000375
1306  003372  001366  ;173372 001366          BNE LOOP        ;NO
1307  003374  105222  ;173374 105222          INCB (R2)+      ;YES, ALL DONE
1308  003376  000142  ;173376 000142          JMP -(R2)       ;THEN TRANSFER TO RTN
1309
1310           THE FOLLOWING 400 LOCATIONS ARE
1311           A REPRODUCTION OF THE DDCMP BOOT-
1312          ;STRAP ROM. IT IS HERE FOR COM-
1313          ;PARISON TO THE ACTUAL ROM AND
1314          ;FOR REFERENCE.
1315
1316                                  ;       COPYRIGHT  1975, DIGITAL EQUIPMENT CORP., MAYNARD,MASS. 01754
1317                                  ;
1318                                  ;       THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A
1319                                  ;       LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND
1320                                  ;       CAN BE COPIED (WITH INCLUSION OF DEC'S
1321                                  ;       COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM,
1322                                  ;       EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING
1323                                  ;       BY DEC.
1324                                  ;
1325                                  ;       THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO
1326                                  ;       CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED
1327                                  ;       AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
1328                                  ;
1329                                  ;       DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR
1330                                  ;       RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH
1331                                  ;       IS NOT SUPPLIED BY DEC.
1332                                  ;
1333                                  ;       VERSION 01
1334                                  ;
1335                                  ;       STUART WECKER  01/22/75
```

```
1336
1337                                    ;       DIGITAL EQUIPMENT CORPORATION
1338                                    ;       COMPUTER NETWORK FACILITIES
1339                                    ;       DOWN-LINE LOADING PROGRAM
1340                                    ;
1341                                    ;       THIS PROGRAM LOADS COMPUTER MEMORY FROM DATA SENT OVER
1342                                    ;       A DATA COMMUNICATIONS LINK. IT SENDS AND RECEIVES
1343                                    ;       MESSAGES IN DDCMP BOOT FORMAT.  THE PRIMARY BOOT ONLY
1344                                    ;       LOADS A SINGLE BLOCK, THE SECONDARY BOOT,  WHICH
1345                                    ;       THEN REQUESTS AND LOADS THE DESIRED PROGRAM.
1346                                    ;
1347                                    ;       CURRENT VERSION DDCMP: 3.0 - MAY 7,1974
1348                                    ;
1349                                    ;       THE BOOTSTRAP MESSAGES ARE OF THE FORM:
1350                                    ;
1351                                    ;       SYN,SYN,DLE,CNT,F.S,FILL,FILL,ADDR,CRC1,DATA,CRC2
1352                                    ;
1353                                    ;       ALL ITEMS ARE 8-BITS LONG UNLESS OTHERWISE SPECIFIED
1354                                    ;
1355                                    ;       SYN-THE SYNC CHARACTER-SYNC-226,ASYNC-377
1356                                    ;       DLE-THE BOOT HEADER CHARACTER-OCTAL 220
1357                                    ;       CNT-THE 14-BIT COUNT FIELD-LENGTH OF DATA FIELD
1358                                    ;       F-THE FINAL BIT-LINK CONTROL
1359                                    ;       S-THE SELECT BIT-LINK CONTROL
1360                                    ;       FILL-A FILL CHARACTER-OCTAL 000
1361                                    ;       ADDR-THE STATION ADDR-FOR PT. TO PT.=1
1362                                    ;       CRC1-THE 16-BIT CRC-16 COMPUTED ON DLE THROUGH ADDR
1363                                    ;       DATA-THE BOOT DATA AS FOLLOWS:
1364                                    ;               CODE,INFO
1365                                    ;       ONLY THE FOLLOWING CODES ARE USED BY THE
1366                                    ;       PRIMARY BOOT
1367                                    ;       CODE=10  REQUEST SECONDARY PROGRAM
1368                                    ;       INFO=DEVICE TYPE,STATION ADDRESS
1369                                    ;       DEVICE TYPE-DP=0,DU=2,DL=4,DQ=6
1370                                    ;       STATION ADDRESS=1
1371                                    ;       CODE=0  PROGRAM LOAD WITH TRANSFER ADDRESS
1372                                    ;       INFO=BLKNO,BLK LDADDR,IMAGE DATA,TRANS ADDR
1373                                    ;       BLKNO=0
1374                                    ;       BLOCK LDADDR=6
1375                                    ;       TRANS ADDR=6
1376                                    ;       HEADER COUNT > OR = TO 10.
1377                                    ;       ADDRESSES ARE 4 BYTES-32 BITS-LOW BIT FIRST
1378                                    ;       CRC2-THE 16-BIT CRC-16 COMPUTED ON THE DATA FIELD ONLY
1379                                    ;
1380                                    ;
1381                                    ;       OPTION SWITCHES:
1382                                    ;               DEVICE-DP11,DU11,DL11
1383                                    ;               CRC-KG11,SCRC
1384                                    ;
1385                                    ;
1386                                    ;
1387                                    ;       REGISTER DEFINITIONS
1388                                    ;
1389                    ;       000000  R0=%0                   ;BLOCK LOAD ADDR
```

```
1390                    ;       000001  R1=%1                   ;DEVICE CSR ADDRESS
1391                    ;       000002  R2=%2                   ;CRC CALC TEMP
1392                    ;       000003  R3=%3                   ;SOFTWARE CRC
1393                    ;       000004  R4=%4                   ;BLOCK CHAR COUNT
1394                    ;       000005  R5=%5                   ;CRC CALC TEMP
1395                    ;       000006  SP=%6                   ;STACK ADDR
1396                    ;       000007  PC=%7                   ;LOCATION COUNTER
1397                                    ;
1398                                    ;       LITERALS
1399                                    ;
1400                    ;       000001  $STADR=1                ;STATION ADDR
1401                    ;       177570  $SWR=177570             ;SWITCH REGISTER ADDR
1402                    ;       000226  $SYN=226                ;SYNC CHARACTER
1403                    ;       000220  $DLE=220                ;DDCMP DLE CHARACTER
1404                    ;       000400  $STRIP=400
1405                                    ;
1406                                    ;       THE STACK IS USED AS FOLLOWS:
1407                                    ;       STACK-2:FOR JSR TO GET ROUTINE
1408                                    ;       STACK-4:TEMP FOR CRC CALCULATION
1409                                    ;
1410                                    ;       START OF BOOT PROGRAM
1411                                    ;
1412                                    ;       START1-DEVICE UNIT 0 NORMAL CONFIGURATION
1413                                    ;       START2-USE SWITCH REG AS DEVICE DISPLACEMENT
1414                                    ;               I.E. #0-0,#1-10,#2-20
1415                                    ;
1416                    ;       173400  .=173400
1417    003400  012700  ;173400 012700 START1: MOV     (PC)+,R0        ;NON ZERO VALUE TO R0
1418    003402  005000  ;173402 005000 START2: CLR     R0              ;CLEAR R0
1419    003404  000005  ;173404 000005         RESET                   ;RESET SYS,MEM MGT, ETC...
1420    003406  012706  ;173406 012706         MOV     #17776,SP       ;STACK AT 4K-2
1421    003410  017776  ;173410 017776
1422                                    ;
1423                                    ;       FIND THE DU-11 IN THE FLOATING ADDRESS SPACE
1424                                    ;
1425    003412  010702  ;173412 010702         MOV     PC,R2           ;CURRENT PC
1426    003414  062702  ;173414 062702         ADD     #DEVTAB-.,R2    ;DEVICE TABLE ADDR
1427    003416  000360  ;173416 000360
1428    003420  012703  ;173420 012703         MOV     #6,R3           ;TRAP PS ADDR
1429    003422  000006  ;173422 000006
1430    003424  005013  ;173424 005013         CLR     (R3)            ;CLEAR NEW PS
1431    003426  010243  ;173426 010243         MOV     R2,-(R3)        ;TABLE ADDR TO LOC 4
1432    003430  160313  ;173430 160313         SUB     R3,(R3)         ;SUB TO TRAP RTN
1433    003432  005303  ;173432 005303         DEC     R3              ;LEAVE CNT 3 FOR LOOP
1434    003434  012701  ;173434 012701         MOV     #160010,R1      ;START SEARCH ADDR
1435    003436  160010  ;173436 .60010
1436    003440  005711  ;173440 005711 DEVLOP: TST     (R1)            ;IS DEVICE THERE
1437    003442  111204  ;173442 111204         MOVB    (R2),R4         ;DEVICE INCREMENT TO R3
1438    003444  060401  ;173444 060401         ADD     R4,R1           ;UPDATE TO NEXT DEVICE
1439    003446  005201  ;173446 005201         INC     R1              ;INCREMENT MODULO
1440    003450  040401  ;173450 040401         BIC     R4,R1           ;CLEAR EXCESS
1441    003452  005703  ;173452 005703         TST     R3              ;TEST FOR DONE
1442    003454  001371  ;173454 001371         BNE     DEVLOP          ;NOT YET
1443    003456  005700  ;173456 005700         TST     R0              ;TEST SWITCH REG USE
```

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

```
1498  003600  001363  ;173600 001363          BNE     GETPGM          ;NO GOOD
1499  003602  123727  ;173602 123727          CMPB    @#6,#$STADR     ;CHECK FOR MY ADDR
1500  003604  000006  ;173604 000006
1501  003606  000001  ;173606 000001
1502  003610  001357  ;173610 001357          BNE     GETPGM          ;NOT MINE
1503  003612  123727  ;173612 123727          CMPB    @#1,#$DLE       ;IS THIS A DLE MSG
1504  003614  000001  ;173614 000001
1505  003616  000220  ;173616 000220
1506  003620  001322  ;173620 001322          BNE     SNDREQ          ;NO, ASK FOR ONE
1507
1508                                  ;       GET DATA BLOCK
1509                                  ;
1510  003622  013704  ;173622 013704          MOV     @#2,R4          ;DATA FIELD LENGTH
1511  003624  000002  ;173624 000002
1512  003626  042704  ;173626 042704          BIC     #140000,R4      ;MASK OFF S,F BITS
1513  003630  140000  ;173630 140000
1514  003632  122424  ;173632 122424          CMPB    (R4)+,(R4)+     ;ADD 2 FOR CRC
1515  003634  005000  ;173634 005000          CLR     R0              ;LOAD INTO LOCATION 0
1516  003636  004767  ;173636 004767          JSR     PC,GET1         ;GET DATA BLOCK
1517  003640  000014  ;173640 000014
1518  003642  005703  ;173642 005703          TST     R3              ;CHECK DATA FIELD CRC
1519  003644  001310  ;173644 001310          BNE     SNDREQ          ;NO GOOD
1520  003646  105713  ;173646 105713          TSTB    (R3)            ;CHECK CODE IN LOC 0
1521  003650  001306  ;173650 001306          BNE     SNDREQ          ;NOT PROGRAM LOAD
1522  003652  000137  ;173652 000137          JMP     @#6             ;TRANSFER TO SECONDARY PGM
1523  003654  000006  ;173654 000006
1524                                  ;
1525                                  ;       GET A BLOCK AND COMPUTE CRC
1526                                  ;
1527                  ;173656         GET:
1528                  ;173656         GET1:
1529  003656  105711  ;173656 105711          TSTB    (R1)            ;IS DEVICE DONE YET
1530  003660  100376  ;173660 100376          BPL     GET             ;NOT YET
1531  003662  042711  ;173662 042711          BIC     #$STRIP,(R1)    ;NO STRIP SYNC
1532  003664  000400  ;173664 000400
1533  003666  116110  ;173666 116110          MOVB    2(R1),(R0)      ;STORE IT
1534  003670  000002  ;173670 000002
1535                                  ;
1536                                  ;       CRC CALCULATION ROUTINE
1537                                  ;
1538                  ;       120001  POLY=120001                     ;CRC-16 POLYNOMIAL
1539                                  ;
1540  003672  012705  ;173672 012705          MOV     #8.,R5          ;BYTE LENGTH
1541  003674  000010  ;173674 000010
1542  003676  112002  ;173676 112002          MOVB    (R0)+,R2        ;CHARACTER TO ADD TO CRC
1543  003700  000241  ;173700 000241  CRCLOP: CLC                     ;CLEAR CARRY
1544  003702  006003  ;173702 006003          ROR     R3              ;SHIFT OLD PARTIAL
1545  003704  103003  ;173704 103003          BCC     L10             ;IF CLEAR CHECK CHAR
1546  003706  006002  ;173706 006002          ROR     R2              ;SHIFT CHARACTER
1547  003710  103003  ;173710 103003          BCC     L11             ;XOR POLY
1548  003712  000410  ;173712 000410          BR      L12             ;NEXT BIT
1549  003714  006002  ;173714 006002  L10:    ROR     R2              ;SHIFT CHARACTER
1550  003716  103006  ;173716 103006          BCC     L12             ;NEXT BIT
1551  003720  012746  ;173720 012746  L11:    MOV     #POLY,-(SP)     ;POLY TO STACK
```

```
1552  003722  120001  ;173722 120001
1553  003724  040316  ;173724 040316          BIC     R3,(SP)         ;NOT PARTIAL AND POLY
1554  003726  042703  ;173726 042703          BIC     #POLY,R3        ;NOT POLY AND PARTIAL
1555  003730  120001  ;173730 120001
1556  003732  052603  ;173732 052603          BIS     (SP)+,R3        ;POLY XOR PARTIAL
1557  003734  005305  ;173734 005305  L12:     DEC     R5              ;DECREMENT BIT COUNT
1558  003736  001360  ;173736 001360          BNE     CRCLOP          ;ONCE MORE
1559  003740  005304  ;173740 005304          DEC     R4              ;DECREMENT COUNT
1560  003742  001345  ;173742 001345          BNE     GET             ;ONCE MORE
1561  003744  000207  ;173744 000207          RTS     PC              ;RETURN
1562                                  ;
1563                                  ;       SECONDARY PROGRAM REQEST MSG
1564                                  ;
1565  003746  113226  ;173746 113226  RQMSG:   .BYTE   $SYN,$SYN,$SYN,$SYN
1566  003750  113226  ;173750 113226
1567  003752  002220  ;173752 002220          .BYTE   $DLE,4,0,0,0,1
1568  003754  000000  ;173754 000000
1569  003756  000400  ;173756 000400
1570  003760  050055  ;173760 050055          .BYTE   55,120
1571  003762  001010  ;173762 001010          .BYTE   10              ;REQ SEC PGM CODE
1572                  ;                       .BYTE   2               ;DEVICE CODE
1573  003764  000001  ;173764 000001          .BYTE   $STADR          ;STATION ADDR
1574                  ;                       .BYTE   0               ;FILL
1575  003766  030242  ;173766 030242          .BYTE   242,60          ;FOR STADR=1
1576                                  ;       NOTE:NODEV AND DEVTAB MUST BE IN THIS ORDER
1577                                  ;       DO NOT SEPARATE THEM
1578                                  ;
1579                                  ;       .EVEN
1580  003770  122243  ;173770 122243  NODEV:   CMPB    (R2)+,-(R3)     ;INC PTR-DEC CNT
1581  003772  000002  ;173772 000002          RTI                     ;RETURN FROM TRAP
1582                                  ;
1583                  ;173774          RQMSGE:                         ;END OF MSG-USE JUNK AS PADS
1584  003774  007407  ;173774 007407  DEVTAB:  .BYTE   7               ;DJ-11
1585                  ;                       .BYTE   17              ;DH-11
1586  003776  END.YC:
1587  003776  003407  ;173776 003407          .BYTE   7               ;DQ-11
1588                  ;                       .BYTE   7               ;DU-11
1589                  ;174000          END:
1590                  ;       173400          .END    START1
```

```
1591  004000  MAP.YD:
1592          ;THE FOLLOWING IS A REPRODUCTION
1593          ;OF THE ROM PROGRAM FOR BM873YD.
1594          ;IT IS HERE FOR COMPARISON TO THE
1595          ;ACTUAL ROM AND FOR REFERENCE
1596          ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
1597          ;BM873-YD.P11
1598
1599                                  ;       THIS CODE IS TO BE BLASTED INTO PROMS ON THE BM873-YD BOARD.
1600                                  ;       WRITTEN BY DAVID M. ROSENBERG           OCTOBER 1974
1601                                  ;REGISTER DEFINITIONS
1602
1603          ;       000000  R0=%0           ;GENERAL PURPOSE REGISTER 0
1604          ;       000001  R1=%1           ;GENERAL PURPOSE REGISTER 1
1605          ;       000002  R2=%2           ;GENERAL PURPOSE REGISTER 2
1606          ;       000003  R3=%3           ;GENERAL PURPOSE REGISTER 3
1607          ;       000004  R4=%4           ;GENERAL PURPOSE REGISTER 4
1608          ;       000005  R5=%5           ;GENERAL PURPOSE REGISTER 5
1609          ;       000006  SP=%6           ;STACK POINTER (REGISTER R6)
1610          ;       000007  PC=%7           ;PROGRAM COUNTER (REGISTER R7)
1611
1612
1613                                  ;SYMBOL DEFINITIONS
1614
1615          ;       177776  PS=177776       ;PROCESSOR STATUS REGISTER
1616          ;       177570  SWR=177570      ;FRONT PANEL SWITCH REGISTER
1617          ;       000000  PR0=0*40        ;PRIORITY LEVEL 0
1618          ;       000040  PR1=1*40        ;PRIORITY LEVEL 1
1619          ;       000100  PR2=2*40        ;PRIORITY LEVEL 2
1620          ;       000140  PR3=3*40        ;PRIORITY LEVEL 3
1621          ;       000200  PR4=4*40        ;PRIORITY LEVEL 4
1622          ;       000240  PR5=5*40        ;PRIORITY LEVEL 5
1623          ;       000300  PR6=6*40        ;PRIORITY LEVEL 6
1624          ;       000340  PR7=7*40        ;PRIORITY LEVEL 7
1625          ;       000001  BIT0=000001
1626          ;       000002  BIT1=000002
1627          ;       000004  BIT2=000004
1628          ;       000010  BIT3=000010
1629          ;       000020  BIT4=000020
1630          ;       000040  BIT5=000040
1631          ;       000100  BIT6=000100
1632          ;       000200  BIT7=000200
1633          ;       000400  BIT8=000400
1634          ;       001000  BIT9=001000
1635          ;       002000  BIT10=002000
1636          ;       004000  BIT11=004000
1637          ;       010000  BIT12=010000
1638          ;       020000  BIT13=020000
1639          ;       040000  BIT14=040000
1640          ;       100000  BIT15=100000
1641
```

```
1642                    ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 3
1643                    ;BM873-YD.P11   BUTTON #1  -  BOOTSTRAP USING THE PDP-11 SWITCH REGISTER
1644
1645
1646                    ;       173000  ROMORG  =       173000          ;SET ROM ORIGIN TO 773000
1647                    ;       173000          .=ROMORG                ;BM873-YD OCCUPIES 773000-773777
1648
1649  004000  033727 ;173000  033727  BUTON1: BIT     @#SWR,#BIT0     ;IS RIGHTMOST BIT ON?
1650  004002  177570 ;173002  177570
1651  004004  000001 ;173004  000001
1652  004006  001010 ;173006  001010          BNE     LOWBIT          ;IF THE BIT IS ON, BRANCH
1653  004010  013707 ;173010  013707          MOV     @#SWR,PC        ;JUMP TO THE ADDRESS IN THE SWITCH REGISTER
1654  004012  177570 ;173012  177570
1655                                                                  ;WITHOUT HAVING TOUCHED ANY OF R0 - R6
1656
1657  004014  111704 ;173014  111704  BUTON3: MOVB    (PC),R4         ;R4 = 1 INDICATES THAT BUTTON #3 WAS PRESSED
1658  004016  005001 ;173016  005001          CLR     R1              ;SET UNIT NUMBER TO ZERO
1659  004020  005005 ;173020  005005          CLR     R5              ;CLEAR "LOGICAL SWITCH REGISTER"
1660  004022  000424 ;173022  000424          BR      TCBOOT          ;DO A DEFAULT BOOT STRAP FROM DECTAPE
1661
1662  004024  173000 ;173024  173000          .WORD   ROMORG,PR7
1663  004026  000340 ;173026  000340
1664
1665  004030  013701 ;173030  013701  LOWBIT: MOV     @#SWR,R1        ;R1 IS A COPY OF THE SWITCH REGISTER
1666  004032  177570 ;173032  177570
1667  004034  106301 ;173034  106301          ASLB    R1              ;LEFT-ALIGN SPEED FIELD IN RIGHT BYTE
1668  004036  122701 ;173036  122701          CMPB    #16*20,R1       ;IS THE SPEED 16 OR 17?
1669  004040  000340 ;173040  000340
1670  004042  101404 ;173042  101404          BLOS    UNITNO          ;IF SPEED IS 16 OR 17, BRANCH
1671  004044  122701 ;173044  122701          CMPB    #3*20,R1        ;IS THE SPEED 0, 1, OR 2?
1672  004046  000060 ;173046  000060
1673  004050  101001 ;173050  101001          BHI     UNITNO          ;IF THE SPEED IS 0, 1, OR 2, BRANCH
1674  004052  005001 ;173052  005001          CLR     R1              ;SPEED WAS 3-15; SET UNIT NUMBER = 0
1675  004054  000301 ;173054  000301  UNITNO: SWAB    R1              ;MOVE UNIT NUMBER TO BITS 0-2
1676
1677                                          ; IT IS POSSIBLE TO MANUALLY SET THE DESIRED BOOTSTRAP UNIT NUMBER
1678                                          ; INTO THE RIGHTMOST THREE BITS OF R1, SET THE PDP-11 FRONT PANEL
1679                                          ; SWITCH REGISTER, AND THEN JUMP INTO THE ROM CODE AT THIS POINT.
1680
1681  004056  042701 ;173056  042701          BIC     #^C7,R1         ;ISOLATE UNIT NUMBER IN R1
1682  004060  177770 ;173060  177770
1683  004062  013705 ;173062  013705          MOV     @#SWR,R5        ;R5 IS NOW THE "LOGICAL SWITCH REGISTER"
1684  004064  177570 ;173064  177570
1685  004066  005004 ;173066  005004          CLR     R4              ;R4 = 0 INDICATES THAT BUTTON #1 WAS PRESSED
1686  004070  105705 ;173070  105705          TSTB    R5              ;SHOULD WE BOOT FROM DECTAPE OR RH11/RP04?
1687  004072  100507 ;173072  100507          BMI     RPBOOT          ;IF BIT 7 WAS ONE, BRANCH OFF TO THE RH11/RP04
1688                                                                  ;OTHERWISE, FALL THROUGH TO THE DECTAPE
```

```
                        ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 4
                        ;BM873-YD.P11   DECTAPE BOOTSTRAP AND DUMP ROUTINES

                        ;       177344  TCWC    =       177344          ;TC11 DECTAPE WORD COUNT REGISTER
                        ;       000001  TCGO    =       1               ;TC11 "GO" BIT
                        ;       000002  TCRNUM  =       1*2             ;TC11 "READ BLOCK NUMBER" FUNCTION
                        ;       000004  TCREAD  =       2*2             ;TC11 "READ DATA" FUNCTION
                        ;       000014  TCWRIT  =       6*2             ;TC11 "WRITE DATA" FUNCTION
                        ;       004000  TCREV   =       4000            ;MOVE DECTAPE IN REVERSE DIRECTION

                                                        ; BOOTSTRAP (FROM DECTAPE) PARAMETERS
                        ;       000400  TCBWDC  =       ^D256           ;WORD COUNT FOR THE SECONDARY BOOTSTRAP
                        ;       000000  TCBEND  =       0               ;WHICH END OF THE DECTAPE (0 = FRONT; 1 = BACK)

                                                        ; DUMP (TO DECTAPE) PARAMETERS
                        ;       070000  TCDWDC  =       ^D28672         ;WORD COUNT FOR THE CORE DUMP TO DECTAPE
                        ;       000001  TCDEND  =       1               ;WHICH END OF THE DECTAPE (0 = FRONT; 1 = BACK)

                                                        ; GENERAL (BOOTSTRAP AND DUMP) DECTAPE PARAMETER
                        ;       000024  TCRTRY  =       ^D20            ;NUMBER OF RETRIES IN CASE OF ERROR

004074  012700 ;173074  012700  TCBOOT: MOV     #<TCBEND*TCREV>!TCREAD!TCGO,R0  ;SET UP DATA-TRANSFER COMMAND
004076  000005 ;173076  000005
004100  012702 ;173100  012702          MOV     #-TCBWDC,R2     ;SET WORD COUNT TO 256 (512 BYTES)
004102  177400 ;173102  177400
004104  012703 ;173104  012703          MOV     #<<1-TCBEND>*TCREV>!TCRNUM!TCGO,R3      ;SET UP POSITION COMMAND
004106  004003 ;173106  004003
004110  000301 ;173110  000301          SWAB    R1              ;BRING UNIT NUMBER INTO THE LEFT BYTE
004112  050103 ;173112  050103          BIS     R1,R3           ;PUT UNIT NUMBER INTO POSITIONING COMMAND
004114  050100 ;173114  050100          BIS     R1,R0           ;PUT UNIT NUMBER INTO DATA-TRANSFER COMMAND
004116  012701 ;173116  012701  TCSTRT: MOV     #TCWC,R1        ;R1 NOW POINTS TO TC11 WORD COUNT REGISTER
004120  177344 ;173120  177344
004122  012706 ;173122  012706  TCLOOP: MOV     #TCRTRY,SP      ;INITIALIZE RETRY COUNT IN SP
004124  000024 ;173124  000024
004126  005705 ;173126  005705  TCBGIN. TST     R5              ;TEST "INDEFINITE RETRY" BIT
004130  100404 ;173130  100404          BMI     TCRSET          ;BRANCH IF "INDEFINITE RETRY" IS ENABLED
004132  005306 ;173132  005306          DEC     SP              ;DECREMENT RETRY COUNT
004134  100002 ;173134  100002          BPL     TCRSET          ;BRANCH IF RETRY COUNT NOT EXHAUSTED
004136  000000 ;173136  000000  TCHALT: HALT                    ;RETRY COUNT IS EXHAUSTED FOR DECTAPE OPERATION
004140  000770 ;173140  000770          BR      TCLOOP          ;HE PRESSED "CONTINUE", SO TRY AGAIN
004142  000005 ;173142  000005  TCRSET: RESET                   ;STOP ANYTHING IN PROGRESS, FOR NEXT TRY
004144  010341 ;173144  010341          MOV     R3,-(R1)        ;INITIATE DECTAPE POSIONING OPERATION
004146  005711 ;173146  005711  TCWAIT: TST     (R1)            ;TEST FOR AN "ERROR"
004150  100376 ;173150  100376          BPL     TCWAIT          ;LOOP UNTIL AN "ERROR" IS DETECTED
004152  005721 ;173152  005721          TST     (R1)+           ;MAKE R1 POINT TO THE WORD COUNT REGISTER
004154  005761 ;173154  005761          TST     -4(R1)          ;IS THE ERROR "ENDZONE"?
004156  177774 ;173156  177774
004160  100362 ;173160  100362          BPL     TCBGIN          ;IF NOT, BRANCH BACK TO TRY AGAIN
004162  010211 ;173162  010211          MOV     R2,(R1)         ;SET UP WORD COUNT FOR DATA-TRANSFER
004164  010041 ;173164  010041          MOV     R0,-(R1)        ;INITIATE THE DATA-TRANSFER OPERATION
004166  105711 ;173166  105711  TCDONE: TSTB    (R1)            ;TEST FOR "DONE"
004170  100376 ;173170  100376          BPL     TCDONE          ;LOOP UNTIL THE "DONE" BIT SETS
004172  005721 ;173172  005721          TST     (R1)+           ;WAS AN "ERROR" DETECTED?
```

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

```
1752                    ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 5
1753                    ;BM873-YD.P11   DECTAPE BOOTSTRAP AND DUMP ROUTINES
1754
1755  004216  010037 ;173216  010037  TCDUMP: MOV     R0,@#ROTOR7     ;SAVE R0 IN PDP-11 MEMORY LOCATION 40
1756  004220  000040 ;173220  000040
1757  004222  000402 ;173222  000402          BR      TCCONT          ;BRANCH AROUND REQUIRED INTERRUPT VECTOR
1758
1759  004224  173000 ;173224  173000          .WORD   ROMORG,PR7
1760  004226  000340 ;173226  000340
1761
1762  004230  010700 ;173230  010700  TCCONT: MOV     PC,R0           ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
1763  004232  000410 ;173232  000410          BR      REGSAV          ;GO TO THE "REGISTER SAVING" SUBROUTINE
1764  004234  012700 ;173234  012700          MOV     #<TCDEND*TCREV>!TCWRIT!TCGO,R0  ;SET UP (WRITE) TRANSFER COMMAND
1765  004236  004015 ;173236  004015
1766  004240  012702 ;173240  012702          MOV     #-TCDWDC,R2     ;SET WORD-COUNT TO 28K WORDS
1767  004242  110000 ;173242  110000
1768  004244  012703 ;173244  012703          MOV     #<(1-TCDEND)*TCREV>!TCRNUM!TCGO,R3      ;SET UP POSITION COMMAND
1769  004246  000003 ;173246  000003
1770  004250  005005 ;173250  005005          CLR     R5              ;CLEAR "INDEFINITE RETRY" BIT
1771  004252  000721 ;173252  000721          BR      TCSTRT          ;BRANCH INTO DECTAPE ROUTINE
1772
1773
1774
1775
1776                                    ; THE FOLLOWING SUBROUTINE IS USED TO SAVE THE PDP-11 GENERAL REGISTERS
1777                                    ; IN PDP-11 MEMORY LOCATIONS 40-57.
1778
1779                                    ; THE CALLING SEQUENCE IS AS FOLLOWS:           MOV     R0,@#ROTOR7
1780                                    ;                                               MOV     PC,R0
1781                                    ;                                               BR      REGSAV
1782                                    ;                                               RETURN HERE
1783
1784  004254  010137 ;173254  010137  REGSAV: MOV     R1,@#ROTOR7+2   ;SAVE R1 IN MEMOTY LOCATION 42
1785  004256  000042 ;173256  000042
1786  004260  012701 ;173260  012701          MOV     #ROTOR7+4,R1    ;R1 NOW POINTS TO MEMORY LOCATION 44
1787  004262  000044 ;173262  000044
1788  004264  010221 ;173264  010221          MOV     R2,(R1)+        ;SAVE R2 IN MEMORY LOCATION 44
1789  004266  010321 ;173266  010321          MOV     R3,(R1)+        ;SAVE R3 IN MEMORY LOCATION 46
1790  004270  010421 ;173270  010421          MOV     R4,(R1)+        ;SAVE R4 IN MEMORY LOCATION 50
1791  004272  010521 ;173272  010521          MOV     R5,(R1)+        ;SAVE R5 IN MEMORY LOCATION 52
1792  004274  010621 ;173274  010621          MOV     SP,(R1)+        ;SAVE SP IN MEMORY LOCATION 54
1793  004276  010021 ;173276  010021          MOV     R0,(R1)+        ;SAVE PC IN MEMORY LOCATION 56
1794  004300  000160 ;173300  000160          JMP     2(R0)           ;RETURN TO THE CALLING ROUTINE
1795  004302  000002 ;173302  000002
1796
```

DO4


```
1797            ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 6
1798            ;BM873-YD.P11   RH11/RP04 BOOTSTRAP AND DUMP ROUTINES
1799
1800
1801            ;       176700  RPCS1   =       176700  ;ADDRESS OF RH11/RP04 CONTROL & STATUS REGISTER 1
1802            ;       000002  RPWC    =       2       ;OFFSET TO RH11/RP04 WORD COUNT REGISTER
1803            ;       000006  RPDA    =       6       ;OFFSET TO RH11/RP04 TRACK & SECTOR ADDRESS REGISTER
1804            ;       000010  RPCS2   =       10      ;OFFSET TO RH11/RP04 CONTROL & STATUS REGISTER 2
1805            ;       000012  RPDS    =       12      ;OFFSET TO RH11/RP04 DRIVE STATUS REGISTER
1806            ;       000032  RPOF    =       32      ;OFFSET TO RH11/RP04 OFFSET REGISTER (CONTAINING FMT22)
1807            ;       000034  RPDC    =       34      ;OFFSET TO RH11/RP04 DESIRED CYLINDER REGISTER
1808
1809            ;       040000  RPTRE   =       BIT14   ;"TRANSFER ERROR" BIT IN RPCS1
1810            ;       020000  RPMCPE  =       BIT13   ;"MASSBUS CONTROL BUS PARITY ERROR" BIT IN RPCS1
1811            ;       004000  RPDVA   =       BIT11   ;"DRIVE AVAILABLE" BIT IN RPCS1
1812            ;       100000  RPATA   =       BIT15   ;"ATTENTION ACTIVE" BIT IN RPDS
1813            ;       040000  RPERR   =       BIT14   ;"COMPOSITE ERROR" BIT IN RPDS
1814            ;       010000  RPFMT   =       BIT12   ;"FMT22" (16-BIT WORDS) BIT IN RPOF
1815
1816            ;       000021  RPPRST  =       21      ;READ-IN PRESET
1817            ;       000061  RPWRIT  =       61      ;WRITE DATA
1818            ;       000071  RPREAD  =       71      ;READ DATA
1819
1820            ;       000000  RPBFMT  =       0       ;BOOTSTRAP FORMAT (0 = 18-BIT WORDS; 2 = 16-BIT WORDS)
1821            ;       000400  RPBWDC  =       ^D256   ;WORD COUNT FOR THE SECONDARY BOOTSTRAP FROM THE RP04
1822            ;       000626  RPBCYL  =       ^D406   ;BOOTSTRAP CYLINDER NUMBER
1823            ;       000000  RPBTRK  =       0       ;BOOTSTRAP TRACK NUMBER
1824            ;       000000  RPBSCT  =       0       ;BOOTSTRAP SECTOR NUMBER
1825
1826            ;       000000  RPDFMT  =       0       ;DUMP FORMAT (0 = 18-BIT WORDS; 2 = 16-BIT WORDS)
1827            ;       070000  RPDWDC  =       ^D28672 ;WORD COUNT FOR THE CORE DUMP TO THE RP04
1828            ;       000631  RPDCYL  =       ^D409   ;DUMP CYLINDER NUMBER
1829                                                    ; THE FOLLOWING TWO ASSIGNMENTS PUT THE DUMP AT THE VERY END OF THE CYLINDER
1830            ;       000015  RPDTRK  =       ^D18-((RPDWDC-1)/((^D20+RPDFMT)*^D256)) ;DUMP TRACK NUMBER
1831            ;       000010  RPDSCT  =       ^D19+RPDFMT-(((RPDWDC-1)/^D256)-((^D18-RPDTRK)*(^D20+RPDFMT)))
1832
1833
1834
1835  004304  111704 ;173304  111704  BUTON2: MOVB    (PC),R4         ;R4 = 5 INDICATES THAT BUTTON #2 WAS PRESSED
1836  004306  005005 ;173306  005005          CLR     R5              ;CLEAR "LOGICAL SWITCH REGISTER"
1837  004310  005001 ;173310  005001          CLR     R1              ;SET UNIT NUMBER TO ZERO
1838
1839  004312  012700 ;173312  012700  RPBOOT: MOV     #(RPREAD*400)!(RPBSCT*10),R0
1840  004314  034400 ;173314  034400
1841  004316  012702 ;173316  012702          MOV     #-RPBWDC,R2
1842  004320  177400 ;173320  177400
1843  004322  012703 ;173322  012703          MOV     #(RPBFMT*40000)!(RPBTRK*2000)!RPBCYL,R3
1844  004324  000626 ;173324  000626
1845  004326  050100 ;173326  050100          BIS     R1,R0           ;PUT THE UNIT NUMBER INTO R0
1846  004330  012701 ;173330  012701  RPSTRT: MOV     #RPCS1,R1       ;SET R1 TO THE LOWEST ADDRESS USED BY THE RH11
1847  004332  176700 ;173332  176700
```

E04


```
                        ;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/RP04 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 RP04 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 RP04 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 RP04 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 RP04 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
```

F04


```
1902  004502  000000 ;173502  000000          HALT                    ;"000240" WAS NOT READ INTO LOCATION ZERO
1903  004504  000641 ;173504  000641          BR      GOTOO           ;BRANCH TO LOCATION ZERO
1904
1905
1906  004506  010037 ;173506  010037  RPDUMP: MOV     R0,@#ROTOR7     ;SAVE R0 IN PDP-11 MEMORY LOCATION "ROTOR7"
1907  004510  000040 ;173510  000040
1908  004512  010700 ;173512  010700          MOV     PC,R0           ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
1909  004514  000657 ;173514  000657          BR      REGSAV          ;GO TO THE "REGISTER SAVING" SUBROUTINE
1910  004516  012700 ;173516  012700          MOV     #<RPWRIT*400>!<RPDSCT*10>,R0
1911  004520  030500 ;173520  030500
1912  004522  012702 ;173522  012702          MOV     #-RPDWDC,R2
1913  004524  110000 ;173524  110000
1914  004526  012703 ;173526  012703          MOV     #<RPDFMT*40000>!<RPDTRK*2000>!RPDCYL,R3
1915  004530  032631 ;173530  032631
1916  004532  000676 ;173532  000676          BR      RPSTRT
1917
```

```
                ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 8
                ;BM873-YD.P11   DTE20 DEVICE REGISTER AND BIT DEFINITIONS

;       174400  DTEBAS=174400           ;BASE OF (FIRST) DTE20 DEVICE REGISTER BLOCK
;       000040  DTESIZ=000040           ;SPACING BETWEEN CONSECUTIVE DTE20'S
;       000004  DTEMAX=4                ;MAXIMUM NUMBER OF DTE20'S ON ONE PDP-11


                                        ;OFFSETS FROM THE BASE OF THE DTE20 DEVICE REGISTER BLOCK
                                        ;TO SPECIFIC 10/11 INTERFACE RAM LOCATIONS AND REGISTERS.

                                ; THE FIRST 12 REGISTERS ARE NOT INITIALIZED BY "INIT" (BECAUSE THEY ARE IN RAMS
;       000000  DLYCNT=00               ;DELAY COUNT                    (ADDRESS XXXX00)
;       000002  DEXWD3=02               ;DEPOSIT OR EXAMINE WORD 3      (ADDRESS XXXX02)
;       000004  DEXWD2=04               ;DEPOSIT OR EXAMINE WORD 2      (ADDRESS XXXX04)
;       000006  DEXWD1=06               ;DEPOSIT OR EXAMINE WORD 1      (ADDRESS XXXX06)
;       000010  TENAD1=10               ;10 ADDRESS WORD 1 FOR DEX      (ADDRESS XXXX10)
;       000012  TENAD2=12               ;10 ADDRESS WORD 2 FOR DEX      (ADDRESS XXXX12)
;       000014  T010BC=14               ;T010 BYTE COUNT                (ADDRESS XXXX14)
;       000016  T011BC=16               ;T011 BYTE COUNT                (ADDRESS XXXX16)
;       000020  T010AD=20               ;T010 PDP11 MEMORY ADDRESS      (ADDRESS XXXX20)
;       000022  T011AD=22               ;T011 PDP11 MEMORY ADDRESS      (ADDRESS XXXX22)
;       000024  T010DT=24               ;T010 PDP11 DATA WORD           (ADDRESS XXXX24)
;       000026  T011DT=26               ;T011 PDP11 DATA WORD           (ADDRESS XXXX26)

                                ; THE LAST 4 REGISTERS ARE INITIALIZED BY "INIT" (BECAUSE THEY ARE IN FLIP-FLOPS
;       000030  DIAG1=30                ;DIAGNOSTIC WORD 1              (ADDRESS XXXX30)
;       000032  DIAG2=32                ;DIAGNOSTIC WORD 2              (ADDRESS XXXX32)
;       000034  STATUS=34               ;10/11 INTERFACE STATUS WORD    (ADDRESS XXXX34)
;       000036  DIAG3=36                ;DIAGNOSTIC WORD 3              (ADDRESS XXXX36)


                                ;  THE FOLLOWING ARE THE ADDRESSES OF THE DTE20 INTERRUPT VECTORS

;       000774  DTEIV1=774              ;INTERRUPT VECTOR FOR DTE20 #1
;       000770  DTEIV2=770              ;INTERRUPT VECTOR FOR DTE20 #2
;       000764  DTEIV3=764              ;INTERRUPT VECTOR FOR DTE20 #3
;       000760  DTEIV4=760              ;INTERRUPT VECTOR FOR DTE20 #4


                                ;  BIT ASSIGNMENTS FOR VARIOUS DTE20 REGISTERS USED BY THIS ROM CODE


                                        ;BIT ASSIGNMENTS FOR T010BC

:       100000  INT11=BIT15             ;SET DONE AND INTERRUPT BOTH 10 AND 11

                                        ;BIT ASSIGNMENTS FOR T011BC

;       100000  INT10=BIT15             ;SET DONE AND INTERRUPT BOTH 10 AND 11
;       040000  ZSTOP=BIT14             ;STOP ON NULL (ZERO) CHARACTER
;       020000  T011BM=BIT13            ;BYTE SIZE FOR TO-11 BYTE TRANSFERS
```

```
                ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 9
                ;BM873-YD.P11   DTE20 DEVICE REGISTER AND BIT DEFINITIONS

                                                        ;BIT ASSIGNMENTS FOR DIAG2 (WRITE)
;       000100  DRESET=BIT6             ;PERFORM DIAGNOSTIC CLEAR

                                                        ;BIT ASSIGNMENTS FOR DIAG3 (READ)
;       000020  DUPE=BIT4               ;DATO UNIBUS PARITY ERROR
;       000004  DURE=BIT2               ;DATO UNIBUS RECEIVE ERROR
;       000002  NUPE=BIT1               ;NPR UNIBUS PARITY ERROR

                                                        ;BIT ASSIGNMENTS FOR DIAG3 (WRITE)

;       000020  CDO=BIT4                ;CLEAR DUPE AND DURE ERROR FLAGS
;       000002  CNUPE=BIT1              ;CLEAR NUPE ERROR FLAG
;       000001  TO10BM=BIT0             ;BYTE SIZE FOR TO-10 BYTE TRANSFER

                                                        ;BIT ASSIGNMENTS FOR STATUS (WRITE)

;       100000  DON10S=BIT15            ;SET TO10 DONE
;       040000  DON10C=BIT14            ;CLEAR TO10 DONE
;       020000  ERR10S=BIT13            ;SET TO10 ERROR
;       010000  ERR10C=BIT12            ;CLEAR TO10 ERROR
;       004000  INT11S=BIT11            ;RING THE PDP-11'S DOORBELL (INTERRUPTS THE -11)
;       002000  INT11C=BIT10            ;STOP RINGING THE PDP-11'S DOORBELL
;       001000  PERCLR=BIT9             ;CLEAR -11 MEMORY PARITY ERROR
;       000400  INT10S=BIT8             ;RING THE PDP-10'S DOORBELL (INTERRUPTS THE -10)
;       000200  DON11S=BIT7             ;SET TO11 DONE
;       000100  DON11C=BIT6             ;CLEAR TO11 DONE
;       000040  INTRON=BIT5             ;ENABLE DTE20 INTERRUPTS TO THE -11
;       000020  EBUSPC=BIT4             ;CLEAR "EBUS PARITY ERROR"
;       000010  INTROF=BIT3             ;DISABLE THE PDP-11 INTERUPTS
;       000004  EBUSPS=BIT2             ;SET "EBUS PARITY ERROR"
;       000002  ERR11S=BIT1             ;SET TO11 ERROR
;       000001  ERR11C=BIT0             ;CLEAR TO11 ERROR

                                                        ;BIT ASSIGNMENTS FOR STATUS (READ)

;       100000  TO10DN=BIT15            ;TO10 DONE
;       020000  TO10ER=BIT13            ;TO 10 ERROR (NPR TIMEOUT OR BUS ERROR)
;       010000  RAMISO=BIT12            ;RAM WORD READ IS ALL ZEROS
;       004000  TO11DB=BIT11            ;1 = THE PDP11'S DOORBELL IS RINGING
;       002000  DXWRD1=BIT10            ;DEPOSIT OR EXAMINE WORD ONE
;       001000  MPE11=BIT9              ;PARITY ERROR WITHIN PDP-11 MEMORY
;       000400  TO10DB=BIT8             ;1 = THE PDP-10'S DOORBELL IS RINGING
;       000200  TO11DN=BIT7             ;TO11 DONE
;       000100  EBSEL=BIT6              ;E BUFFER SELECT
;       000040  NULSTP=BIT5             ;NULL STOP
;       000020  BPARER=BIT4             ;EBUS PARITY ERROR
;       000010  RSTRCT=BIT3             ;THIS PDP-11 IS "RESTRICTED"
;       000004  DEXDON=BIT2             ;DEPOSIT OR EXAMINE DONE
;       000002  TC11ER=BIT1             ;TO 11 ERROR (NPR TIMEOUT OR BUS ERROR)
;       000001  INTSON=BIT0             ;DTE20 INTERRUPTS (TO THE -11) ARE ENABLED
```

```
                ;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   PROCEDURE BY WHICH THE PDP-10 BOOTSTRAPS AND/OR DUMPS THE PDP-11
```

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

1. CLEAR THE DTE20 AND INITIATE A BM873 BUTTON #4 BOOTSTRAP OPERATION
   - CONO [SR11B!CL11PT!CLTO11!CLTO10!PILDEN]

2. WAIT TO SEE PDP-11 POWER FAIL (AC LOW = TRUE) - CONI [DEAD11] = 1

3. WAIT TO SEE PDP-11 POWER RECOVER (AC LOW = FALSE) - CONI [DEAD11] = 0

4. WAIT AT LEAST ANOTHER 150 MILLISECONDS AND THEN CLEAR THE RELOAD -11 BUTTON
   - CONO [CR11B]

5. SET BYTE COUNTER TO A SPECIAL CODE (1365 OCTAL) - DATAO [1365]

6. RING PDP-11'S DOORBELL - CONO[TO11DB]

7. WAIT UNTIL "-10 RINGING -11'S DOORBELL" IS TURNED OFF BY THE -11
   (I.E. UNTIL CONI[TO11DB] BECOMES ZERO).

8. ENABLE THE DTE20 TO USE PI 0 INTERRUPTS
   (I.E. SET CONO[PILDEN!PIOENB]).

9. SET UP THE TO-10 BYTE POINTER (IN THE EPT) FOR THE FIRST 3.5K.

10. SET UP THE BYTE COUNTER FOR THE FIRST 3.5K, INDICATING
    "INTERRUPT -10 ONLY" - DATAO [1000]

11. WAIT FOR "TO-10 DONE" OR "TO-10 ERROR" - CONI [TO10DN!TO10ER]

12. NOTE WHETHER THERE WAS AN ERROR (CONI [TO10ER]) AND THEN TURN OFF
    TO10DN AND TO10ER - CONO [CLTO10]. IF ERROR, GO TO STEP 17.

13. IF END OF 28K, GO TO STEP 17.

14. SET UP TO-10 BYTE POINTER (IN THE EPT) FOR THE NEXT 3.5K.

15. SET UP THE BYTE COUNTER FOR THE NEXT 3.5K INDICATING
    "INTERRUPT -10 ONLY" (DATAO [1000]), UNLESS THIS IS THE
    LAST 3.5K (OF 28K), IN WHICH CASE INDICATE "INTERRUPT
    BOTH PROCESSORS" (DATAO [TO10IB!1000]).

16. GO TO STEP 11.

```
2079          ;BM873-YD       - KL10 (PDP-11, 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
2080          ;BM873-YD.P11   PROCEDURE BY WHICH THE PDP-10 BOOTSTRAPS AND/OR DUMPS THE PDP-11
2081
2082
2083                          17. SET UP TO-11 BYTE POINTER (IN THE EPT) FOR "PDP-11 BOOTSTRAP".
2084                              NOTE THAT THE FIRST WORD OF THIS "PDP-11 BOOTSTRAP" MUST
2085                              BE THE BIT PATTERN 000240 (A PDP-11 NOP INSTRUCTION).
2086
2087                          18. RING THE PDP-11'S DOORBELL   -   CONO [TO11DB]
2088
2089                          19. WAIT FOR EITHER TO11DB TO GO OFF (CONI[TO11DB] = 0),
2090                              OR TO10DB TO COME ON (CONI[TO10DB] = 1).
2091
2092                          20. IF NO ERROR WAS NOTED IN STEP 12, TO11DB SHOULD GO OFF
2093                              (TO10DB COMING ON INDICATES A MASSIVE SCREWUP).
2094                              IF AN ERROR WAS NOTED IN STEP 12, TO11DB GOING OFF INDICATES
2095                              THAT THE ERROR WAS "NON-FATAL" (NON-EX-MEM OR -11 MEMORY
2096                              PARITY) AND THE -11 IS PROCEEDING. TO10DB COMING ON INDICATES
2097                              THAT THE ERROR WAS "FATAL" AND THE -11 IS HALTED AT LOCATION 173714.
2098                              IN THIS LATTER CASE THE -10 MUST RESTART FROM STEP 1.
2099
2100                          21. IF TO11DB WENT OFF, WAIT FOR "TO-11 DONE" OR "TO-11 ERROR"
2101                              -   CONI [TO11DN!TO11ER]
2102
2103                          22. NOTE WHETHER THERE WAS AN ERROR   -   CONI [TO11ER]
2104
2105                          23. TURN OFF TO11DN AND TO11ER AND RING THE PDP-11'S DOORBELL
2106                              -   CONO [TO11DB!CLTO11]
2107
2108                          24. WAIT FOR EITHER TO11DB TO GO OFF (CONI[TO11DB] = 0),
2109                              OR TO10DB TO COME ON (CONI[TO10DB] = 1).
2110
2111                          25. TO11DB GOING OFF INDICATES THAT THE PDP-11 FOUND NO ERRORS
2112                              AND IS TRANSFERRING CONTROL TO THE CODE WHICH WAS JUST
2113                              RECEIVED FROM THE -10. IN THIS CASE THE -10 SHOULD START
2114                              FOLLOWING THE PROTOCOL OF THIS CODE.
2115
2116                          26. TO10DB COMING ON INDICATES THAT THE PDP-11 HAS FOUND AN
2117                              ERROR (OR THAT THE FIRST WORD TRANSMITTED WASN'T THE
2118                              BIT PATTERN 000240), AND THE PDP-11 IS HALTED AT LOCATION 173766.
2119                              IN THIS CASE THE -10 MUST RESTART FROM STEP 1.
2120
2121
2122
2123
2124
2125
```

```
                        ;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)
                        ;
                        ;       000130  DTECOR  =       130             ;CORE ADDRESS INTO WHICH TO STORE DTE20 REGS.
                        ;       000014  DTEREG  =       ^D12            ;NUMBER OF DTE20 REGISTERS TO STORE
                        ;       000400  DTEWDC  =       ^D256           ;WORD COUNT FOR SECONDARY BOOTSTRAP FROM THE -10
                                                        ; ENTER HERE WHEN THE DTE20 PRESSES BUTTON #4 (BOOTSTRAP INITIATED
                                                        ;       BY THE PDP-10, THROUGH THE DTE20)
004534  010037  ;173534  010037  BUTON4: MOV     R0,@#ROTOR7     ;SAVE R0 IN PDP-11 MEMORY LOCATION "ROTOR7"
004536  000040  ;173536  000040
004540  010700  ;173540  010700          MOV     PC,R0           ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
004542  000644  ;173542  000644          BR      REGSAV          ;GO TO THE "REGISTER SAVING" SUBROUTINE
004544  005005  ;173544  005005          CLR     R5              ;SET R5 = 0
004546  012501  ;173546  012501          MOV     (R5)+,R1        ;SAVE LOCATION 0 IN R1
004550  012503  ;173550  012503          MOV     (R5)+,R3        ;SAVE LOCATION 2 IN R3
004552  012504  ;173552  012504          MOV     (R5)+,R4        ;SAVE LOCATION 4 IN R4
004554  011500  ;173554  011500          MOV     (R5),R0         ;SAVE LOCATION 6 IN R0
004556  012715  ;173556  012715          MOV     #PR7,(R5)       ;SET UP PRIORITY FOR NON-EX-MEM TRAP
004560  000340  ;173560  000340
004562  005745  ;173562  005745          TST     -(R5)           ;SET R5 = 4
004564  012702  ;173564  012702  10$:    MOV     #DTEBAS-DTESIZ,R2
004566  174340  ;173566  174340
004570  010715  ;173570  010715          MOV     PC,(R5)         ;STORE ADDRESS FOR NON-EX-MEM TRAP
004572  010506  ;173572  010506          MOV     R5,SP           ;SET STACK POINTER = 4
004574  062702  ;173574  062702  11$:    ADD     #DTESIZ,R2      ;R2 POINTS TO THE NEXT DTE20
004576  000040  ;173576  000040
004600  105702  ;173600  105702          TSTB    R2
004602  100770  ;173602  100770          BMI     10$             ;START LOOKING FROM THE BEGINNING AGAIN
004604  032762  ;173604  032762          BIT     #T011DB,STATUS(R2)      ;IS THIS -10 RINGING THE -11'S DOORBELL?
004606  004000  ;173606  004000
004610  000034  ;173610  000034
004612  001770  ;173612  001770          BEQ     11$             ;IF IT IS NOT, GO LOOK FOR ANOTHER -10
004614  026217  ;173614  026217          CMP     T010BC(R2),(PC) ;CHECK FOR A CODE (1365) FROM THE PDP-10
004616  000014  ;173616  000014
                                                                ;INDICATING THAT IT WANTS TO BOOTSTRAP THE -11
004620  001365  ;173620  001365          BNE     11$
                                        ; NOTE THAT AT THIS POINT R2 CONTAINS THE ADDRESS OF THE DEVICE REGISTER
                                        ;       BLOCK FOR THIS DTE20, THAT R5 = 4, AND THAT SP = 4
004622  005725  ;173622  005725          TST     (R5)+           ;SET R5 = 6
004624  010015  ;173624  010015          MOV     R0,(R5)         ;RESTORE THE CONTENTS OF LOCATION 6
004626  010445  ;173626  010445          MOV     R4,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 4
004630  010345  ;173630  010345          MOV     R3,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 2
004632  010145  ;173632  010145          MOV     R1,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 0
                                        ; NOTE: AT THIS TIME R5 = 0. THIS FACT WILL BE USED LATER.
004634  012700  ;173634  012700          MOV     #DTECOR,R0      ;R0 = CORE ADDRESS FOR STORING DTE20 REGISTERS
004636  000130  ;173636  000130
004640  010204  ;173640  010204          MOV     R2,R4
004642  012420  ;173642  012420  7$:     MOV     (R4)+,(R0)+     ;SAVE THE NEXT DTE20 REGISTER IN CORE
004644  022700  ;173644  022700          CMP     #<DTEREG*2>+DTECOR,R0   ;HAVE WE FINISHED YET?
004646  000160  ;173646  000160
004650  101374  ;173650  101374          BHI     7$              ;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     T010AD(R2)      ;START WRITING -11 MEMORY INTO THE -10.
004670  000020 ;173670  000020
004672  032711 ;173672  032711  6$:     BIT     #T011DB,(R1)    ;HAS THE -10 RUNG THE -11'S DOORBELL?
004674  004000 ;173674  004000
004676  001775 ;173676  001775          BEQ     6$              ;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     #T010DB,(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$:     CLR     T011AD(R2)      ;START INPUTTING AT LOCATION 0
004726  000022 ;173726  000022
004730  012762 ;173730  012762          MOV     #INT10!<<-DTEWDC>&7777>,T011BC(R2)      ;READ IN 256 WORDS
004732  107400 ;173732  107400
004734  000016 ;173734  000016
004736  032711 ;173736  032711  1$:     BIT     #T011DB,(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$:     BITB    #T011DN!T011ER,(R1)     ;IS THE TRANSMISSION FINISHED?
004746  000202 ;173746  000202
004750  001775 ;173750  001775          BEQ     4$              ;LOOP UNTIL IT IS FINISHED
004752  100003 ;173752  100003          BPL     5$              ;IF "T011DN" ISN'T ON, "T011ER" 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$:     MOV     #T010DB,(R1)    ;SIGNAL THE -10 THAT THERE WAS AN ERROR
004764  000400 ;173764  000400
004766  000000 ;173766  000000  12$:    HALT                    ;THIS ERROR HALT IS BECAUSE EITHER "T011ER"
                                        ; 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 .
                ;
                                ;
                                ;       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
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;
                                ;
                                ;       DIAG1 DEFINITIONS
                                ;
                ;       004000  DS04=004000     ;KL CLOCK ERROR STOP
                ;       002000  DS05=002000     ;RUN
                ;       001000  DS06=001000     ;HALT
                ;       000400  DEX=000400      ;DEPOSIT OR EXAMINE MAJOR STATE
                ;       000200  TO10=000200     ;TO 10
                ;       000200  DFUNC=000200
                ;       000100  TO11=000100     ;TO-11 TRANSFER MAJOR STATE
                ;       000040  DI011=000040    ;DIAGNOSE 10/11 INTERFACE
                ;       000020  PULSE=000020    ;SINGLE CLOCK CYCLE
                ;       000010  DIKL10=00010    ;DIAGNOSTIC MODE SWITCH
                ;       000004  DSEND=000004    ;SEND DATA
                ;       000001  DCOMST=000001   ;DIAGNOSTIC COMMAND START
                ;
                ;       DIAG1 FUNCTIONS
                ;
                ;       000000  .STPCL=0                ;STOP THE KL CLOCK
                ;       001000  .STRCL=01*1000          ;START THE KL CLOCK
                ;       002000  .SSCLK=02*1000          ;SINGLE STEP THE M BOX CLOCK
                ;       003000  .SECLK=03*1000          ;SINGLE STEP THE  EBOX CLOCK. LEAVES THE
                                ;EBOX CLOCK FALSE AND EBOX SYNC TRUE.
                                ;CAUSES (2 3) MBOX CLOCKS DEPENDING ON
                                ;EBOX CLOCK INITIALLY (FALSE,TRUE).
                                ;DOES NOT DEPEND ON 'T' FIELD OR MB WAIT.
                ;       004000  .CECLK=04*1000          ;CONDITIONALLY ISSUE AN EBOX CLOCK IF THE EBOX
                                ;CLOCK IS TRUE.  MAKES EBOX CLOCK FALSE.
                                ;IF ISSUED IN THE MASTER RESET STATE.
                                ;LEAVES EBOX SYNC TRUE.
                ;       005000  .BRCLK=05*1000          ;ISSUE A BURST OF THE CLOCKS.  THE NUMBER
                                ;OF MBOX CLOCKS DESIRED (1-255) HAS BEEN
                                ;BEEN LOADED PREVIOUSLY BY FUNCTIONS LDBRR,LDBRL
                                ;(42,43)
                ;       006000  .CLRMR=06*1000          ;CLEAR MASTER RESET STATE
                ;       007000  .SETMR=07*1000          ;SET MASTER RESET STATE. RUNNING THE CLOCK WHILE IN THIS
                                ;STATE 'CLEARS' THE KL10.
                ;       010000  .CLRUN=10*1000          ;CLEAR THE RUN FLOP.  MAKE THE MICRO CODE GO TO
                                ;THE HALT-LOOP.
                ;       011000  .SETRN=11*1000          ;SET THE RUN FLOP.  ALLOW REPEATED INSTRUCTION EXECUTION
                ;       012000  .CONBT=12*1000          ;SET THE CONTINUE FLOP (MOMENTARY). ALLOW THE
                                ;MICRO CODE TO LEAVE THE HALT LOOP
                ;       014000  .IRLTC=14*1000          ;UNLATCH THE IR AND LOAD IT FROM THE AD.
                ;       015000  .DRLTC=15*1000          ;UNLATCH THE DRAM REGISTER AND ALLOW IT TO LOAD FROM THE
                                ;RAMS
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;
                                ;
                                ;       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  TO1GDB=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
                                ;CTYOCW=24      ;CTY #0 COMMAND WORD
                                ;CTYORW=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
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 2
                        ;
                                        ;
                                        ;
                                        ; DEFINITIONS . . .
                                        ;
;       000040  ROTOR7= 40      ;SAVE R0 TO R7 IN 40 TO 56
                                ;
;       000130  DTESAV= 130     ;SAVE FIRST 12 DTE REGISTERS DLYCNT TO TO11DT
;                               ; IN LOCATIONS 130-156
                                ;
;       000012  RETRY=  10.     ;DO 10 RETRIES BEFORE HALTING
                                ;
;       173000  ROMORG= 173000  ;ROM STARTS AT 773000
                                ;
                                ; ESTABLISH ROM ORIGIN
                                ;
;       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
005025  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
```

```
                                        ;
                                        ; UNIT # IS IN R0-- CALL PROPER BOOT DEPENDING ON BIT 7
                                        ;
005064  105705  ;173064 105705           TSTB    RS              ;WHERE SHOULD WE BOOT FROM?
005066  100550  ;173066 100550           BMI     RPBOOT          ;BIT 7 = 1 -- BOOT FROM RP04 DISK
                                        ;       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
                                ;
```

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

```
2807                    ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 4
2808                    ;
2809                                    ;
2810                                    ; HERE TO START RX11 ON A TRANSFER, EITHER DUMP OR BOOT
2811                                    ;
2812  005110  012706  ;173110 012706  RXSTRT: MOV     #RETRY,SP       ;SET RETRY COUNT
2813  005112  000012  ;173112 000012
2814  005114  012701  ;173114 012701          MOV     #RXEPA+RXCS,R1  ;ADDRESS CONTROL STATUS REGISTER FOR RX11
2815  005116  177170  ;173116 177170
2816                                    ;       BR      RXRTRY          ;FALL THROUGH RETRY CHECK
2817                                    ;
2818                                    ; HERE ON ERROR TO RETRY
2819                                    ;
2820  005120  005705  ;173120 005705  RXRTRY: TST     R5              ;INDEFINITE RETRY?
2821  005122  100402  ;173122 100402          BMI     10$             ;YES-- TRY FAITHFULLY
2822  005124  005306  ;173124 005306          DEC     SP              ;NO-- DECREMENT RETRY COUNT
2823  005126  002475  ;173126 002475          BLT     RXEHLT          ;GIVE UP IF RUN OUT
2824                                    ;
2825  005130  000005  ;173130 000005  10$:     RESET   ;CLEAR THE WORLD
2826  005132  005004  ;173132 005004          CLR     R4              ;ALWAYS START TRANSFER AT LOCATION ZERO
2827  005134  010302  ;173134 010302          MOV     R3,R2           ;GET START TRACK AND SECTOR
2828  005136  032711  ;173136 032711  20$:     BIT     #RXDONE,(R1)    ;WAIT UNTIL READY FOR FUNCTION
2829  005140  000040  ;173140 000040
2830  005142  001775  ;173142 001775          BEQ     20$             ;NOT YET-- WAIT
2831  005144  005700  ;173144 005700          TST     R0              ;THIS WRITE?
2832  005146  100454  ;173146 100454          BMI     RXFLSL          ;YES-- FILL SILO BEFORE WRITE
2833                                    ;       BR      RXPERF          ;NO-- JUST DO FIRST READ
2834                                    ;
2835                                    ; HERE TO PERFORM READ OR WRITE, AS SPECIFIED IN R0
2836                                    ;
2837  005150  110011  ;173150 110011  RXPERF: MOVB    R0,(R1)         ;DO READ OR WRITE
2838  005152  105711  ;173152 105711  10$:     TSTB    (R1)            ;READY?
2839  005154  100376  ;173154 100376          BPL     10$             ;NO-- WAIT
2840  005156  110261  ;173156 110261          MOVB    R2,RXDB(R1)     ;SET SECTOR #
2841  005160  000002  ;173160 000002
2842  005162  105711  ;173162 105711  20$:     TSTB    (R1)            ;READY FOR TRACK?
2843  005164  100376  ;173164 100376          BPL     20$             ;NO-- WAIT
2844  005166  000302  ;173166 000302          SWAB    R2              ;YES-- GET TRACK #
2845  005170  110261  ;173170 110261          MOVB    R2,RXDB(R1)     ;SET IT
2846  005172  000002  ;173172 000002
2847  005174  000302  ;173174 000302          SWAB    R2              ;RESTORE HIGH TRACK, LOW SECTOR
2848  005176  032711  ;173176 032711  30$:     BIT     #RXERR!RXDONE,(R1) ;DONE OR ERROR?
2849  005200  100040  ;173200 100040
2850  005202  001775  ;173202 001775          BEQ     30$             ;NO-- WAIT
2851  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$:    BITB    #RX.REQ!RXDONE,(R1) ;READY FOR WORD, OR TRANSFER DONE?
005220  000240  ;173220 000240
005222  000402  ;173222 000402                        BR      20$             ;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
```

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

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 6
                        ;
                                        ;       RP04 DISK BOOTSTRAP AND DUMP ROUTINES
                                        ;
                                        ; RP04 REGISTER DEFINITIONS
                                        ;
                        ;       176700  RPEPA=  176700  ;EXTERNAL PAGE ADDRESS OF RP04 REGISTERS
                                        ;
                        ;       000000  RPCS1=  0       ;OFFSET FOR CSR #1
                        ;       040000          RPTRE=  BIT14           ;TRANFER ERRROR
                        ;       020000          RPMCPE= BIT13           ;MASSBUS CONTROL PARITY ERROR
                        ;       004000          RPDVA=  BIT11           ;DRIVE AVAILABLE (TO -11)
                        ;       000200          RPRDY=  BIT7            ;FUNCTION COMPLETE
                        ;       000076          RPFUNC= BIT5!BIT4!BIT3!BIT2!BIT1 ;FUNCTION:
                        ;       000020                  RPPRST= 20      ; READ-IN PRESET
                        ;       000060                  RPWRIT= 60      ; WRITE DATA
                        ;       000070                  RPREAD= 70      ; READ DATA
                        ;       000001          RPGO=   BIT0            ;GO
                        ;       000002  RPWC=   2       ;WORD COUNT REGISTER
                        ;       000006  RPDA=   6       ;TRACK (HIGH BYTE) SECTOR (LOW BYTE)
                        ;       000010  RPCS2=  10      ;CONTROL AND STATUS REGISTER #2
                        ;       000007          RPUNIT= BIT2!BIT1!BIT0  ;UNIT #
                        ;       000012  RPDS=   12      ;DRIVE STATUS REGISTER
                        ;       100000          RPATA=  BIT15           ;ATTENTION ACTIVE
                        ;       040000          RPERR=  BIT14           ;DRIVE ERROR
                        ;       000034  RPDC=   34      ;DESIRED CYLINDER
                                        ;
                                        ; PARAMETERS
                                        ;
                        ;       000000  RPBCYL= 0.      ;BOOT FROM CYLINDER 0
                        ;       000000  RPBTRK= 0.      ; TRACK 0
                        ;       000000  RPBSCT= 0.      ; SECTOR 0
                                        ;
                        ;       000631  RPDCYL= 409.    ;DUMP TO CYLINDER 409
                        ;       000015  RPDTRK= 13.     ; TRACK 13
                        ;       000010  RPDSCT= 8.      ; SECTOR 8
                                        ;
                                        ; REGISTER USAGE:
                                        ;       R0 --   FUNCTION CODE (HIGH BYTE) UNIT # (LOW BYTE)
                                        ;               BIT 15 SET IF WRITE
                                        ;       R1 --   ADDRESS OF RPCS1
                                        ;       R2 --   CYLINDER #
                                        ;       R3 --   TRACK (HIGH BYTE) SECTOR (LOW BYTE)
                                        ;       R4 --   WORD COUNT
                                        ;       R5 --   PARAMETER WORD SAVED FROM INITIALIZATION
                                        ;       SP --   RETRY COUNTER
                                        ;
```

```
3021                                            ;
3022                                            ; HERE TO BOOT FROM RP04-- UNIT # IN R0
3023                                            ;
3024                        ;173410             RPBOOT:
3025  005410  005002        ;173410 005002              CLR     R2
3026  005412  005003        ;173412 005003              CLR     R3
3027  005414  052700        ;173414 052700              BIS     #<RPREAD+RPGO>*BIT8,R0 ;SET READ HIGH BYTE, UNIT # LOW BYTE
3028  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
```

```
3082                                    ;
3083                                    ; HERE TO HALT AFTER A DUMP-- DISPLAY R0= 0 IF NO ERRORS
3084                                    ;
3085  005534  005000  ;173534 005000  HALTO:  CLR     R0              ;DISPLAY R0= 0 TO INDICATE NO ERRORS
3086                                    ;
3087  005536  000000  ;173536 000000  HALTED: HALT    ;DIE
3088  005540  000776  ;173540 000776          BR      HALTED          ;STAY DEAD
3089                                    ;
3090                                    ; HERE ON ERROR FROM RP04 AFTER RETRYING-- DISPLAY DRIVE STATUS IN R0
3091                                    ;
3092  005542  016100  ;173542 016100  RPEHLT: MOV     RPDS(R1),R0     ;DISPLAY DRIVE STATUS
3093  005544  000012  ;173544 000012
3094  005546  000773  ;173546 000773          BR      HALTED          ;R.I.P.
3095                                    ;
3096                                    ;
3097                                    ; START -11 HERE TO DUMP TO RP04 DISK
3098                                    ;
3099                                    ; NOTE THAT R0-R7 HAVE ALREADY BEEN SAVED IN 40-56
3100                                    ;  BY PRESSING BUTTON #1.
3101                                    ;
3102            ;173550               RPDUMP:
3103  005550  012702  ;173550 012702          MOV     #RPDCYL,R2
3104  005552  000631  ;173552 000631
3105  005554  012703  ;173554 012703          MOV     #<RPDTRK*BIT8>!<RPDSCT*BIT0>,R3
3106  005556  006410  ;173556 006410
3107  005560  012700  ;173560 012700          MOV     #BIT15!<<RPWRIT+RPGO>*BIT8>,R0 ;DO A WRITE, UNIT # 0
3108  005562  130400  ;173562 130400
3109  005564  012704  ;173564 012704          MOV     #-<1024.*28.>,R4 ;SET TO DUMP 28 K
3110  005566  110000  ;173566 110000
3111  005570  005005  ;173570 005005          CLR     R5              ;CLEAR INDEFINITE RETRY BIT
3112  005572  000714  ;173572 000714          BR      RPSTRT          ;START DUMP GOING
```

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

```
3157                                            ;
3158                                            ; HERE ON NXM TRAP-- RESET SP AND TRY NEXT DTE
3159                                            ;
3160  005634  012706  ;173634 012706  21$:    MOV     #4,SP           ;SET SP TO 4, STACK IS LOCATIONS 2 AND 0
3161  005636  000004  ;173636 000004
3162                                            ;
3163  005640  062704  ;173640 062704  22$:    ADD     #DTESIZ,R4      ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS
3164  005642  000040  ;173642 000040
3165  005644  105704  ;173644 105704          TSTB    R4              ;IS THIS THE END OF THE DTE'S?
3166                  ;                               ; NOTE THAT THE LAST DTE IS AT 774540
3167                  ;                               ; AND THAT NOW R4= 774600 IF END
3168  005646  100770  ;173646 100770          BMI     20$             ;YES-- START ALL OVER, UNTIL A DTE
3169                  ;                               ; SAYS HE PUSHED THE BUTTON
3170  005650  032764  ;173650 032764          BIT     #T011DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?
3171  005652  004000  ;173652 004000
3172  005654  000034  ;173654 000034
3173  005656  001770  ;173656 001770          BEQ     22$             ;NO-- TRY NEXT DTE
3174  005660  026417  ;173660 026417          CMP     T010BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365
3175  005662  000014  ;173662 000014
3176                  ;                               ; IN IT'S TO -10 BYTE COUNT?
3177  005664  001365  ;173664 001365          BNE     22$             ;NO-- TRY ANOTHER DTE
3178                                          ;
```

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

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 8
                        ;
                                                ;       FILL TO END OF ROM
                                                ;
                                                ;
                        ;173774 000004          .PRINT  <1000>-<.-ROMORG> ;FREE BYTES AT 1000
005774  000000          ;173774    000          .BYTE   0
                        ;173775    000          .BYTE   0
005776  END.YF:
005776  000000          ;173776    000          .BYTE   0
                        ;173777    000          .BYTE   0
                                                ;
                                                ;
                                                ;
                        ;174000 000001  PASS2:  .END
                        ;
                        ;
```

```
                        MAP.YG:
                        ;THE FOLLOWING IS A REPRODUCTION
                        ;OF THE ROM PROGRAM FOR BM873YG.
                        ;IT IS HERE FOR COMPARISON TO THE
                        ;ACTUAL ROM AND FOR REFERENCE
                        ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 1

                        ;                               .TITLE  BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM
                        ;
                                                ; THIS IS THE CODE TO BE ENCODED IN THE BOOTSTRAP ROM ON THE BM873-YG BOARD
                                                ;
                                                ;       MODULE:         BM873G
                                                ;
                                                ;       DATE:           JUNE 1976
                                                ;
                                                ;       AUTHOR:         RICH MURATORI
                                                ;
                                                ;       COPYRIGHT (C) 1976 DIGITAL EQUIPMENT CORPORATION
                                                ;       ALL RIGHTS RESERVED
                                                ;
                                                ;
                        ;                               .ENABLE ABS,AMA
                                                ;
                                                ;       ASCII CHARACTER DEFINTIONS
                                                ;
                        ;       000040  SPACE=  40                      ;ASCII SPACE
                        ;       000001  SYN=    1                       ;ASCII SYNC
                        ;       000012  LF=     12                      ;ASCII LINE FEED
                        ;       000015  CR=     15                      ;ASCII CARRIAGE RETURN
                        ;       000054  COMMA=  54                      ;ASCII COMMA
                        ;       000006  ACK=    6                       ;ASCII ACKNOWLEDGE
                        ;       000025  NAK=    25                      ;ASCII NEG ACKNOWLEDGE
                                                ;
                                                ;       BUFFER AREAS
                                                ;
                        ;       002100  LINBUF= 2100                    ;LINE INPUT BUFFER
                        ;       002310  DEABUF= 2310                    ;DEASCIIZED INPUT BUFFER
                                                ;
                                                ;       DL11E REGISTER DEFINITIONS
                                                ;
                        ;       176000  DLRCSR= 176000                  ;DL11E RECEIVER STATUS REGISTER
                        ;       176002  DLRBUF= 176002                  ;DL11E RECEIVER BUFFER
                        ;       176004  DLXCSR= 176004                  ;DL11E TRANSMITTER STATUS REGISTER
                        ;       176006  DLXBUF= 176006                  ;DL11E TRANSMISSION BUFFER
                                                ;
                        ;       100000  BIT15=100000
                        ;       000340  PR7=7*40                ;PRIORITY LEVEL 7
                                                ;
                                                ;       DTE20 REGISTER DEFINITIONS
                                                ;
                        ;       174400  DLYCNT=174400   ;DELAY COUNT WORD
                        ;       174414  TO10BC=174414   ;TO-10 PDP-11 MEMORY ADDRESS
```

```
3324                    ;               174416  T011BC=174416   ;TO-11 BYTE COUNT
3325                    ;               174420  T010AD=174420   ;TO-10 PDP-11 MEMORY ADDRESS
3326                    ;               174422  T011AD=174422   ;TO-11 PDP-11 MEMORY ADDRESS
3327                    ;               174426  T011DT=174426   ;TO-11 PDP-11 DATA WORD
3328                    ;               174434  STAT=174434     ;STATUS WORD
3329                                            ;       T011BC REGISTER BIT DEFINITIONS
333C                                            ;
```

```
3331          ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 1-1
3332
3333          ;               100000  IFLOP=100000    ;I FLIPFLOP BIT
3334                                  ;
3335                                  ;
3336                                  ;       DIAG2 DEFINITIONS
3337                                  ;
3338          ;               000100  DRESET=000100   ;DTE RESET
3339                                  ;
3340                                  ;       STAT REGISTER DEFINITIONS
3341                                  ;
3342          ;               004000  T011DB=004000   ;-10 REQUESTED -11 INTERRUPT
3343                                  ;
3344                                  ;
3345                                  ; DEFINITIONS . . .
3346                                  ;
3347          ;               000040  ROTOR7= 40      ;SAVE R0 TO R7 IN 40 TO 56
3348                                  ;
3349          ;               000130  DTESAV= 130     ;SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
3350          ;                                       ; IN LOCATIONS 130-156
3351                                  ;
3352          ;               173000  ROMORG= 173000  ;ROM STARTS AT 773000
3353                                  ;
3354                                  ; ESTABLISH ROM ORIGIN
3355                                  ;
3356          ;               173000          .=ROMORG
```

```
3357            ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 2
3358
3359            ;                               .SBTTL  BOOT FROM MASTER FRONT END
3360            ;
3361                                    ;EXTERNAL BUTTON #1 -- LOAD FROM MASTER FRONT END VIA DL11E
3362                                    ;
3363                                    ;DEPRESSING BUTTON #1 CAUSES A LOADER UTILITY PROGRAM (DGQDE) TO BE LOADED
3364                                    ;
3365                                    ;TO ACCOMPLISH THIS, THE FOLLOWING CONDITIONS MUST EXIST:
3366                                    ;       1) THE LOADER UTILITY PROGRAM DGQDO MUST BE RUNNING UNDER KLDCP
3367                                    ;          IN THE MASTER FRONT END.
3368                                    ;       2) THE '.A11' FILE FOR DGQDE MUST BE AVAILABLE ON THE SELECTED LOAD
3369                                    ;          MEDIUM IN THE MASTER FRONT END.
3370                                    ;       3) THE MASTER FRONT END AND THE SECONDARY FRONT END MUST BE CONNECTED
3371                                    ;          THROUGH DL11E'S AND A NULL MODEM.
3372                                    ;       4) THE SECONDARY FRONT END MUST HAVE A TTY CONNECTED TO IT.
3373            ;
3374                                    ;SEND BOOTSTRAP REQUEST TO MASTER FRONT END VIA THE DL11E.  THE
3375                                    ;REQUEST IS IN THE FORM 'B <CR><LF>', WHICH CALLS FOR THE
3376                                    ;BOOTING OVER OF DGQDE.A11.
3377  006000  000005  ;173000 000005  BUTON1: RESET                   ;CLEAR THE WORLD
3378  006002  012706  ;173002 012706         MOV     #2000,SP        ;SETUP STACK POINTER
3379  006004  002000  ;173004 002000
3380  006006  012701  ;173006 012701         MOV     #DGQDE,R1       ;ADDRESS OF SYNC + BOOT REQUEST
3381  006010  173374  ;173010 173374
3382  006012  105737  ;173012 105737  SENDIT: TSTB    @#DLXCSR        ;IS DL11E READY TO TRANSMIT
3383  006014  176004  ;173014 176004
3384  006016  100375  ;173016 100375         BPL     SENDIT          ;LOOP UNTIL IT IS
3385  006020  000403  ;173020 000403         BR      1$              ;BRANCH AROUND POWER -FAIL VECTOR
3386                                    ;*****************************************************************
3387                                    ;       REQUIRED POWER-FAIL VECTOR   - MUST BE AT 173024
3388            .
3389  006022  000000  ;173022 000000         .WORD   0               ;FILLER
3390  006024  173000  ;173024 173000         .WORD   ROMORG,PR7
3391  006026  000340  ;173026 000340
3392            ;
3393                                    ;*****************************************************************
3394            .
3395  006030  112137  ;173030 112137  1$:     MOVB    (R1)+,@#DLXBUF  ;LOAD A CHAR INTO OUTPUT BUFFER
3396  006032  176006  ;173032 176006
3397  006034  105711  ;173034 105711         TSTB    (R1)            ;ANY MORE CHARS TO SEND?
3398  006036  001365  ;173036 001365         BNE     SENDIT          ;BRANCH IF MORE CHARS TO SEND
3399  006040  005005  ;173040 005005         CLR     R5              ;CLEAR SYNC RECEIVED FLAG
3400                                    ;WAIT TO RECEIVE BOOT PROGRAM (DGQDE.A11), ONE ASCIIZED CHAR AT
3401                                    ;A TIME, ONE LINE AT A TIME.
3402  006042  012701  ;173042 012701  NXTLIN: MOV     #LINBUF,R1      ;ADDRESS OF LINE INPUT BUFFER
3403  006044  002100  ;173044 002100
3404  006046  105737  ;173046 105737  NXTCHR: TSTB    @#DLRCSR        ;CHAR RECEIVED YET?
3405  006050  176000  ;173050 176000
3406  006052  100375  ;173052 100375         BPL     NXTCHR          ;BRANCH IF STILL TO WAIT
3407            ;
3408                                    ;PROCESS THE RECEIVED ASCIIZED CHAR.  IGNORE ALL CHARS UNTIL A SYNC
3409                                    ;SIGNAL IS RECEIVED.  A LINE FEED MARKS THE END OF A LINE.  THE MAX
3410                                    ;NUMBER OF CHARS PER LINE IS 131, MORE THAN THAT IS AN ERROR.
```

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

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

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

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

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

```
3562            ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 2-3
3563
3564                    ;173411    000          .BYTE   0
3565  006412  000000    ;173412    000          .BYTE   0
3566                    ;173413    000          .BYTE   0
3567  006414  000000    ;173414    000          .BYTE   0
3568                    ;173415    000          .BYTE   0
3569  006416  000000    ;173416    000          .BYTE   0
3570                    ;173417    000          .BYTE   0
3571  006420  000000    ;173420    000          .BYTE   0
3572                    ;173421    000          .BYTE   0
3573  006422  000000    ;173422    000          .BYTE   0
3574                    ;173423    000          .BYTE   0
3575  006424  000000    ;173424    000          .BYTE   0
3576                    ;173425    000          .BYTE   0
3577  006426  000000    ;173426    000          .BYTE   0
3578                    ;173427    000          .BYTE   0
3579  006430  000000    ;173430    000          .BYTE   0
3580                    ;173431    000          .BYTE   0
3581  006432  000000    ;173432    000          .BYTE   0
3582                    ;173433    000          .BYTE   0
3583  006434  000000    ;173434    000          .BYTE   0
3584                    ;173435    000          .BYTE   0
3585  006436  000000    ;173436    000          .BYTE   0
3586                    ;173437    000          .BYTE   0
3587  006440  000000    ;173440    000          .BYTE   0
3588                    ;173441    000          .BYTE   0
3589  006442  000000    ;173442    000          .BYTE   0
3590                    ;173443    000          .BYTE   0
3591  006444  000000    ;173444    000          .BYTE   0
3592                    ;173445    000          .BYTE   0
3593  006446  000000    ;173446    000          .BYTE   0
3594                    ;173447    000          .BYTE   0
3595  006450  000000    ;173450    000          .BYTE   0
3596                    ;173451    000          .BYTE   0
3597  006452  000000    ;173452    000          .BYTE   0
3598                    ;173453    000          .BYTE   0
3599  006454  000000    ;173454    000          .BYTE   0
3600                    ;173455    000          .BYTE   0
3601  006456  000000    ;173456    000          .BYTE   0
3602                    ;173457    000          .BYTE   0
3603  006460  000000    ;173460    000          .BYTE   0
3604                    ;173461    000          .BYTE   0
3605  006462  000000    ;173462    000          .BYTE   0
3606                    ;173463    000          .BYTE   0
3607  006464  000000    ;173464    000          .BYTE   0
3608                    ;173465    000          .BYTE   0
3609  006466  000000    ;173466    000          .BYTE   0
3610                    ;173467    000          .BYTE   0
3611  006470  000000    ;173470    000          .BYTE   0
3612                    ;173471    000          .BYTE   0
3613  006472  000000    ;173472    000          .BYTE   0
3614                    ;173473    000          .BYTE   0
3615  006474  000000    ;173474    000          .BYTE   0
```

```
  3616                    ;173475    000          .BYTE   0
  3617  006476  000000    ;173476    000          .BYTE   0
```

```
                        ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 2-4
                        ;
                        ;173477    000          .BYTE   0
006500  000000          ;173500    000          .BYTE   0
                        ;173501    000          .BYTE   0
006502  000000          ;173502    000          .BYTE   0
                        ;173503    000          .BYTE   0
006504  000000          ;173504    000          .BYTE   0
                        ;173505    000          .BYTE   0
006506  000000          ;173506    000          .BYTE   0
                        ;173507    000          .BYTE   0
006510  000000          ;173510    000          .BYTE   0
                        ;173511    000          .BYTE   0
006512  000000          ;173512    000          .BYTE   0
                        ;173513    000          .BYTE   0
006514  000000          ;173514    000          .BYTE   0
                        ;173515    000          .BYTE   0
006516  000000          ;173516    000          .BYTE   0
                        ;173517    000          .BYTE   0
006520  000000          ;173520    000          .BYTE   0
                        ;173521    000          .BYTE   0
006522  000000          ;173522    000          .BYTE   0
                        ;173523    000          .BYTE   0
006524  000000          ;173524    000          .BYTE   0
                        ;173525    000          .BYTE   0
006526  000000          ;173526    000          .BYTE   0
                        ;173527    000          .BYTE   0
006530  000000          ;173530    000          .BYTE   0
                        ;173531    000          .BYTE   0
006532  000000          ;173532    000          .BYTE   0
                        ;173533    000          .BYTE   0
006534  000000          ;173534    000          .BYTE   0
                        ;173535    000          .BYTE   0
006536  000000          ;173536    000          .BYTE   0
                        ;173537    000          .BYTE   0
                        ;                       .EVEN
```

```
                        ;
                                                .SBTTL  REGISTER SAVE ROUTINE
                                                ;       REGISTER SAVE ROUTINE
                                                ;
                                                ; REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS R0-R7
                                                ; IN MEMORY AT 40-56 (LOCATION ROTOR7).
                                                ;
                                                ; CALLING SEQUENCE:
                                                ;       MOV     R0,ROTOR7+0
                                                ;       MOV     #RET,R0
                                                ;       BR      REGSAV
                                                ; RET:  <RETURN HERE>
                                                ;
                                                ; ALL REGISTERS RESTORED
                                                ;
                        ;               173540          .=ROMORG+540
                        ;
006540  010037  ;173540 010037  REGSAV: MOV     R0,ROTOR7+16    ;SAVE R0 AS PC IN 56
006542  000056  ;173542 000056
                        ;
006544  012700  ;173544 012700          MOV     #ROTOR7+16,R0   ;R0 NOW POINTS TO 56
006546  000056  ;173546 000056
                        ;
006550  010640  ;173550 010640          MOV     SP,-(R0)        ;SAVE SP IN 54
006552  010540  ;173552 010540          MOV     R5,-(R0)        ;SAVE R5 IN 52
006554  010440  ;173554 010440          MOV     R4,-(R0)        ;SAVE R4 IN 50
006556  010340  ;173556 010340          MOV     R3,-(R0)        ;SAVE R3 IN 46
006560  010240  ;173560 010240          MOV     R2,-(R0)        ;SAVE R2 IN 44
006562  010140  ;173562 010140          MOV     R1,-(R0)        ;SAVE R1 IN 42
006564  014000  ;173564 014000          MOV     -(R0),R0        ;RESTORE R0 FROM 40
006566  000177  ;173566 000177          JMP     @ROTOR7+16      ;GO TO SAVED PC
006570  004264  ;173570 004264
                        ;
006572  000000  ;173572 000000          .WORD   0               ;FILLER WORD
```

```
                        ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 4
                        
                        ;                               .SBTTL  DUMP AND BOOTSTRAP THROUGH DTE20
                                                ;       INTERNAL BUTTON #4 -- DUMP AND BOOTSTRAP THROUGH DTE20
                                                ;
                        ;               000040  DTESIZ= 40      ;EACH DTE OCCUPIES 20 WORDS IN EXTERNAL PAGE
                                                ;
                                                ; BUTTON #4 -- INITIATED BY '-10 RELOAD -11' BIT
                                                ;
                        ;               173574          .=ROMORG+574
                        
006574  010037  ;173574 010037  BUTON4: MOV     R0,ROTOR7+0     ;SAVE R0 IN 40
006576  000040  ;173576 000040
006600  012700  ;173600 012700          MOV     #10$,R0         ;SET RETURN ADDRESS IN R0
006602  173606  ;173602 173606
006604  000755  ;173604 000755          BR      REGSAV          ;SAVE R1-R7
                                                ;
                                                ; REGISTERS SAVED-- LOOK FOR THE DTE20 WHICH PUSHED THE BUTTON
                                                ;
                                                ; THE DTE WHICH PUSHED THE BUTTON SHOULD HAVE THE DOORBELL
                                                ;  RINGING AND HAVE THE VALUE 1365 (OCTAL) IN IT'S
                                                ;  TO -10 BYTE COUNT TO10BC.
                                                ;
                                                ; NXM (TIME-OUT) TRAP IS USED TO SKIP NON-EXISTANT DTE20'S.
                                                ;
006606  005005  ;173606 005005  10$:    CLR     R5              ;ADDRESS LOCATION ZERO
006610  012500  ;173610 012500          MOV     (R5)+,R0        ;SAVE 0 IN R0
006612  012501  ;173612 012501          MOV     (R5)+,R1        ;SAVE 2 IN R1
006614  011502  ;173614 011502          MOV     (R5),R2         ;SAVE 4 IN R2
006616  012725  ;173616 012725          MOV     #21$,(R5)+      ;SET NXM TRAP ADDRESS IN 4
006620  173634  ;173620 173634
006622  011503  ;173622 011503          MOV     (R5),R3         ;SAVE 6 IN R3
006624  012715  ;173624 012715          MOV     #PR7,(R5)       ;SET PRIORITY FOR NXM TRAP
006626  000340  ;173626 000340
                                                ;
                                                ; LOOP THROUGH ALL DTE'S
                                                ;
006630  012704  ;173630 012704  20$:    MOV     #DLYCNT-DTESIZ,R4 ;POINT TO DTE # -1'S DELAY COUNT REGISTER
006632  174340  ;173632 174340
                                                ; (WILL BUMP TO # 0)
                                                ;
                                                ; HERE ON NXM TRAP-- RESET SP AND TRY NEXT DTE
                                                ;
006634  012706  ;173634 012706  21$:    MOV     #4,SP           ;SET SP TO 4, STACK IS LOCATIONS 2 AND 0
006636  000004  ;173636 000004
                                                ;
006640  062704  ;173640 062704  22$:    ADD     #DTESIZ,R4      ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS
006642  000040  ;173642 000040
006644  105704  ;173644 105704          TSTB    R4              ;IS THIS THE END OF THE DTE'S?
                        ;                                       ; NOTE THAT THE LAST DTE IS AT 774540
                                                                ; AND THAT NOW R4= 774600 IF END
006646  100770  ;173646 100770          BMI     20$             ;YES-- START ALL OVER, UNTIL A DTE
                        ;                                       ; SAYS HE PUSHED THE BUTTON
006650  032764  ;173650 032764          BIT     #TO11DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?
```

```
3744  006652  004000  ;173652 004000
3745  006654  000034  ;173654 000034
3746  006656  001770  ;173656 001770          BEQ     22$                 ;NO-- TRY NEXT DTE
3747  006660  026417  ;173660 026417          CMP     T010BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365
3748  006662  000014  ;173662 000014                                      ; IN IT'S TO -10 BYTE COUNT?
3749                  ;
3750  006664  001365  ;173664 001365          BNE     22$                 ;NO-- TRY ANOTHER DTE
3751                  ;
```

```
3752            ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 5
3753
3754                                    ;
3755                                    ; WE HAVE FOUND THE DTE WHICH PUSHED THE BUTTON
3756                                    ;
3757                                    ; ADDRESS OF DLYCNT REGISTER IS IN R4
3758                                    ;
3759  006666  010315  ;173666 010315           MOV     R3,(R5)         ;RESTORE LOCATION 6
3760  006670  010245  ;173670 010245           MOV     R2,-(R5)        ; 4
3761  006672  010145  ;173672 010145           MOV     R1,-(R5)        ; 2
3762  006674  010045  ;173674 010045           MOV     R0,-(R5)        ; 0
3763                                    ;
3764                                    ; SAVE FIRST 12 DTE REGISTERS DLYCNT TO TO11DT
3765                                    ;  IN LOCATIONS 130-156
3766                                    ;
3767  006676  012700  ;173676 012700           MOV     #DTESAV,R0      ;POINT TO SAVE AREA
3768  006700  000130  ;173670 000130
3769  006702  012420  ;173702 012420  29$:    MOV     (R4)+,(R0)+     ;SAVE A REGISTER
3770  006704  022700  ;173704 022700           CMP     #TO11DT-DLYCNT+DTESAV,R0 ;FINISHED?
3771  006706  000156  ;173706 000156
3772  006710  103374  ;173710 103374           BHIS    29$             ;NO-- SAVE SOME MORE
3773                                    ;
3774                                    ; R4= TO11DT+2
3775                                    ;
3776                                    ; SET R1= STATUS REGISTER
3777                                    ;     R4= DIAG2 REGISTER
3778                                    ;
3779                                    ; DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
3780                                    ;  LOADED FLAG
3781                                    ;
3782
3783  006712  005724  ;173712 005724           TST     (R4)+
3784  006714  010401  ;173714 010401           MOV     R4,R1           ; SO DOES R1
3785  006716  012700  ;173716 012700           MOV     #DRESET,R0      ;SETUP R0 FOR 'DIAGNOSTIC RESET'
3786  006720  000100  ;173720 000100
3787  006722  010021  ;173722 010021           MOV     R0,(R1)+        ;R1 POINTS TO STATUS REGISTER
```

```
                ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 6
                                        ;
                                        ; 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.
                                        ;
006724  005061  ;173724 005061          CLR     DLYCNT-STAT(R1) ;SET DTE20 FOR MAXIMUM DELAY (ZERO)
006726  177744  ;173726 177744
006730  005061  ;173730 005061          CLR     TO10AD-STAT(R1) ;START DUMPING -11 MEMORY TO -10
006732  177764  ;173732 177764
                                                                ; STARTING AT LOCATION 0
006734  032711  ;173734 032711  30$:    BIT     #TO11DB,(R1)    ;IS DOORBELL RINGING (TRANSFER COMPLETE)?
006736  004000  ;173736 004000
006740  001775  ;173740 001775          BEQ     30$             ;NO-- WAIT FOR DOORBELL
006742  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.
                                        ;
006744  005061  ;173744 005061          CLR     TO11AD-STAT(R1) ;START INPUT TO LOCATION 0
006746  177766  ;173746 177766
006750  012761  ;173750 012761          MOV     #IFLOP!<<-256.>&7777>,TO11BC-STAT(R1) ;256 WORDS, INTERRUPT
006752  107400  ;173752 107400
006754  177762  ;173754 177762
                                                                ; -10 WHEN DONE
006756  032711  ;173756 032711  40$:    BIT     #TO11DB,(R1)    ;DOORBELL RINGING (LOAD FINISHED)?
006760  004000  ;173760 004000
006762  001775  ;173762 001775          BEQ     40$             ;NO-- WAIT UNTIL DONE
006764  010014  ;173764 010014          MOV     R0,(R4)         ;CLEAR DOORBELL RINGING
006766  012705  ;173766 012705          MOV     #BIT15,R5       ;SET R5: BIT15= 1, BIT0= 0
006770  100000  ;173770 100000
                                                                ; TO SAY BUTTON #4 PRESSED
006772  005007  ;173772 005007          CLR     PC              ;GO TO LOADED CODE, STARTING AT
                ;                                               ; LOCATION 0
                                        ;
                                        ;       FILL TO END OF ROM
                                        ;
006774  000000; ;173774     000         .BYTE   0
                ;173775     000         .BYTE   0
006776  END.YG:
006776  000000; ;173776     000         .BYTE   0
                ;173777     000         .BYTE   0
                ;           000001      .END
```

```
3839  007000  MAP.YH:
3840          ;THE FOLLOWING IS A REPRODUCTION
3841          ;OF THE ROM PROGRAM FOR BM873YH.
3842          ;IT IS HERE FOR COMPARISON TO THE
3843          ;ACTUAL ROM AND FOR REFERENCE
3844          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY'1 27(666)  17-AUG-76  16:19  PAGE 1
3845
3846          ;                               .SBTTL  TITLE PAGE
3847          ;                               .TITLE  BM873H  - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)
3848
3849                                  ;
3850                                  ; COPYRIGHT (C) 1975, 1976 DIGITAL EQUIPMENT CORPORATION
3851                                  ; ALL RIGHTS RESERVED
3852                                  ;
3853                                  ; THIS IS THE CODE TO BE ENCODED IN THE BOOTSTRAP ROM ON THE BM873-YH BOARD
3854                                  ;
3855                                  ;
3856                                  ; MODULE:       BM873H
3857                                  ;
3858                                  ; DATE:         10-MAR-76
3859                                  ;
3860                                  ; AUTHOR:       TOM PORCHER
3861                                  ;
3862          ;                               .ENABLE ABS,AMA
3863          ;                               .LIST   MEB
3864          ;                               .MCALL  $DEF
3865                  ;000000                 $DEF
```

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

;                               .SBTTL  MACROS AND DEFINITIONS
;
;
; MACRO TO FILL TO A LOCATION (RELATIVE TO ROM ORIGIN) WITH ZERO BYTES
;
;               .MACRO  FILLTO  LOC
;.IFGE <LOC>-<.-ROMORG>
;  .IFG <LOC>-<.-ROMORG>
;    .IFDF PASS2
;               .PRINT  <LOC>-<.-ROMORG> ;FREE BYTES AT LOC
;    .ENDC
;  .ENDC
;  .REPT <LOC>-<.-ROMORG>
;               .BYTE   0
;  .ENDR
;.IFF
;               .ERROR  <.-ROMORG>-<LOC> ;BOUNDARY EXCEEDED AT LOC
;.ENDC
;               .ENDM   FILLTO
;
;
; MACRO TO DO 'MOV #XXX,DEST' OR 'CLR DEST' IF XXX IS ZERO
;
;               .MACRO  MOVO    XXX,DEST
;.IFEQ XXX
;               CLR     DEST
;.IFF
;               MOV     #XXX,DEST
;.ENDC
;               .ENDM   MOVO
;
;
; MACRO TO ADD A SMALL NUMBER TO A REGISTER
;  GENERATES ONE OF THE FOLLOWING:
;               CMP     -(REG),-(REG)   ; -4
;               TST     -(REG)          ; -2
;               CMPB    -(REG),-(REG)   ; -2 (REGISTER MAY BE ODD)
;               DEC     REG             ; -1
;               <NOTHING>               ; 0
;               INC     REG             ; 1
;               TST     (REG)+          ; 2
;               CMPB    (REG)+,(REG)+   ; 2 (REGISTER MAY BE ODD)
;               CMP     (REG)+,(REG)+   ; 4
;               ADD     #XXX,REG        ; ANYTHING ELSE
;
; USE THIS MACRO WITH CAUTION, SINCE IT REFERENCES MEMORY
;  AND ALSO DOES NOT SET THE CONDITION CODES PROPERLY
;
;               .MACRO  ADDX    XXX,REG,ODD
;$$$=.
;.IFEQ XXX+4
;  .IF B <ODD>
```

```
3920            ;                               CMP     -(REG),-(REG)
3921            ;                           .ENDC
```

```
3922          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 2-1
3923
3924          ;                       .ENDC
3925          ;                       .IFEQ XXX+2
3926          ;                         .IF B <0DD>
3927          ;                               TST     -(REG)
3928          ;                         .IFF
3929          ;                               CMPB    -(REG),-(REG)
3930          ;                         .ENDC
3931          ;                       .ENDC
3932          ;                       .IFEQ XXX+1
3933          ;                               DEC     REG
3934          ;                       .ENDC
3935          ;                       .IFEQ XXX
3936          ;                         SSS=SSS+2
3937          ;                       .ENDC
3938          ;                       .IFEQ XXX-1
3939          ;                               INC     REG
3940          ;                       .ENDC
3941          ;                       .IFEQ XXX-2
3942          ;                         .IF B <0DD>
3943          ;                               TST     (REG)+
3944          ;                         .IFF
3945          ;                               CMPB    (REG)+,(REG)+
3946          ;                         .ENDC
3947          ;                       .ENDC
3948          ;                       .IFEQ XXX-4
3949          ;                         .IF B <0DD>
3950          ;                               CMP     (REG)+,(REG)+
3951          ;                         .ENDC
3952          ;                       .ENDC
3953          ;                       .IFEQ SSS-.
3954          ;                               ADD     #XXX,REG
3955          ;                       .ENDC
3956          ;                               .ENDM   ADDX
3957                                ;
3958                                ;
3959                                ; DEFINITIONS . . .
3960                                ;
3961          ;             000040  ROTOR7= 40                      ;SAVE R0 TO R7 IN 40 TO 56
3962                                ;
3963          ;             000130  DTESAV= 130                     ;SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
3964          ;                                                     ; IN LOCATIONS 130-156
3965                                ;
3966          ;             000014  RETRY=  12.                     ;DO 12 RETRIES BEFORE HALTING
3967                                ;
3968          ;                001  .IFNDF  TESTVR
3969          ;             173000  ROMORG= 173000                  ;ROM STARTS AT 773000
3970          ;                     .IFF
3971          ;                     ROMORG= 073000                  ;IF TEST, START AT 73000 INSTEAD
3972          ;                000  .ENDC
3973                                ;
3974                                ; ESTABLISH ROM ORIGIN
3975                                ;
```

```
3976            ;                       001  .IF DF TESTVR
3977            ;                            .=ROMORG-2
```

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

                ;                                       HALT
                ;                               .IFF
                ;                       173000  .=ROMORG
                ;                          000  .ENDC
```

```
  3984          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 3
  3985
  3986          ;                               .SBTTL  EXTERNAL BUTTONS #1, #2, #3
  3987                                  ;
  3988                                  ; BUTTON #1 -- LOAD USING SWITCH REGISTER
  3989                                  ;
  3990                        ;173000J  BUTON1:
  3991  007000  010037       ;173000 010037          MOV     R0,ROTOR7+0     ;SAVE R0 IN LOCATION 40
  3992  007002  000040       ;173002 000040
  3993  007004  013700       ;173004 013700          MOV     SWR,R0          ;GET SWITCH REGISTER
  3994  007006  177570       ;173006 177570
  3995  007010  032700       ;173010 032700          BIT     #BIT0,R0        ;IS LOW-ORDER BIT SET?
  3996  007012  000001       ;173012 000001
  3997  007014  001007       ;173014 001007          BNE     BUTONX          ;YES-- LOOK AT CONTENTS
  3998  007016  000513       ;173016 000513          BR      REGSAV          ;NO-- SAVE R1-R7 IN 42-56, GO TO ADDRESS IN R0 (FROM SWR
  3999                                  ;
  4000                                  ; BUTTON #3 -- LOAD BOOT FROM RX11 FLOPPY DISK
  4001                                  ;
  4002                        ;173020   BUTON3:
  4003  007020  005000       ;173020 005000          CLR     R0              ;SAY LOAD FROM FLOPPY, UNIT 0
  4004  007022  000404       ;173022 000404          BR      BUTONX          ;GO TO COMMON CODE FOR 3 BUTTONS
  4005
  4006                                  ; REQUIRED POWER-FAIL VECTOR
  4007                                  ;
  4008                        ;173024
  4009  007024  173000       ;173024 173000          FILLTO  24
  4010  007026  000340       ;173026 000340          .WORD   ROMORG,PR7
  4011
  4012                                  ; BUTTON #2 -- LOAD BOOT FROM RP04 DISK
  4013                                  ;
  4014                        ;173030   BUTON2:
  4015  007030  012700       ;173030 012700          MOV     #BIT7,R0        ;BIT 7 MEANS LOAD FROM RP04
  4016  007032  000200       ;173032 000200
  4017                                  ;       BR      BUTONX          ;FALL INTO COMMON CODE
  4018                                  ;
  4019                                  ; R0 IS SAVED IN R5 AS THE PARAMETER WORD PASSED TO BOOT
  4020                                  ; AND CONTAINS ONE OF THE FOLLOWING:
  4021                                  ;
  4022                                  ;       BIT 0 = 1       IF FROM SWITCH REGISTER
  4023                                  ;       BIT 7 = 0       LOAD FROM RX11 FLOPPY DISK (OR TC11 DECTAPE)
  4024                                  ;       BIT 7 = 1       LOAD FROM RP04 DISK
  4025                                  ;       BIT 15 = 1      INDEFINITE RETRY
  4026                                  ;
  4027                        ;173034   BUTONX:
  4028  007034  010005       ;173034 010005          MOV     R0,R5           ;SAVE PARAMETER FOR BOOT
  4029  007036  106300       ;173036 106300          ASLB    R0              ;LEFT-ALIGN SPEED FIELD IN LOW BYTE
  4030  007040  122700       ;173040 122700          CMPB    #3*BIT4,R0      ;IS SPEED 0, 1, OR 2?
  4031  007042  000060       ;173042 000060
  4032  007044  101001       ;173044 101001          BHI     10$             ;YES-- UNIT IS UNIT TO USE
  4033  007046  005000       ;173046 005000          CLR     R0              ;NO-- USE UNIT #0
  4034                        ;173050   10$:
  4035  007050  000300       ;173050 000300          SWAB    R0              ;GET UNIT # IN LOW BYTE
  4036  007052  042700       ;173052 042700          BIC     #^C7,R0         ;TRIM TO 3 BITS 2, 1, 0
  4037  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
```

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

007070  005037  ;173070 005037          CLR     6               ; . .
007072  000006  ;173072 000006
                ;       BR      RXBOOT          ;TRY FLOPPY FIRST
```

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

                        ;
                        ;                               .SBTTL  RX11 FLOPPY DISK BOOTSTRAP ROUTINES
                                                ;
                                                ; RX11 REGISTER DEFINITIONS
                                                ;
                        ;               177170  RXEPA=  177170                  ;EXTERNAL PAGE ADDR OF FLOPPY
                                                ;
                        ;               000000  RXCS=   0                       ;OFFSET FOR CSR
                        ;               100000          RXERR=  BIT15           ;ERROR
                        ;               000200          RXTREQ= BIT7            ;TRANSFER REQUEST
                        ;               000040          RXDONE= 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
                                                ;
                                                ;
                                                ;  NOTE THAT THE BOOTSTRAP IS WRITTEN IN LOGICAL BLOCK 0
                                                ;  WHICH IS TRACK 1, SECTORS 1, 3, 5, 7.  ONLY SECTOR 1
                                                ;  IS READ BY THE ROM.
                                                ;
                                                ;
                                                ; REGISTER USAGE:
                                                ;       R0 --   READ FUNCTION WITH UNIT SELECT SET
                                                ;       R1 --   ADDRESS OF RXCS
                                                ;       R2 --   ADDRESS OF RXDB
                                                ;       R3 --   UNIT #
                                                ;       R4 --   DATA ADDRESS (TO READ OR WRITE)
                                                ;       R5 --   PARAMETER WORD SAVED FROM INITIALIZATION
                                                ;       SP --   RETRY COUNTER
                                                ;
                                                ;
                                                ; HERE TO BOOT FROM RX11 FLOPPY DISK-- UNIT # IN R0
                                                ;
                                                RXBOOT:
                        ;173074
 007074  012706  ;173074 012706                         MOV     #RETRY,SP       ;SET RETRY COUNT
 007076  000014  ;173076 000014
 007100  012701  ;173100 012701                         MOV     #RXEPA+RXCS,R1  ;ADDRESS CONTROL STATUS REGISTER FOR RX11
 007102  177170  ;173102 177170
 007104  010003  ;173104 010003                         MOV     R0,R3           ;COPY UNIT #
                                                        BR      RXRTRY          ;FALL THROUGH RETRY CHECK
                                                ;
                                                ; HERE ON ERROR TO RETRY
                                                ;
                                                RXRTRY:
                        ;173106
 007106  005705  ;173106 005705                         TST     R5              ;INDEFINITE RETRY?
 007110  100402  ;173110 100402                         BMI     RXRSET          ;YES-- TRY FAITHFULLY
 007112  005306  ;173112 005306                         DEC     SP              ;NO-- DECREMENT RETRY COUNT
```

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

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

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

```
 4166                                        ;
 4167                     ;173220                    FILLTO  224
 4168  007220  000000     ;173220      000           .BYTE   0
 4169                     ;173221      000           .BYTE   0
 4170  007222  000000     ;173222      000           .BYTE   0
 4171                     ;173223      000           .BYTE   0
 4172  007224  173000     ;173224  173000            .WORD   ROMORG,PR7
 4173  007226  000340     ;173226  000340
 4174                                        ;
 4175                                        ; HERE ON ERROR AFTER RETRYING -- DISPLAY ERROR REGISTER AND HALT
```

E09


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

                                         ;
                                         RXEHLT:
007230  012711   ;173230 012711                  MOV     #RXERR+RXGO,(R1) ;DO A READ ERROR REGISTER FUNCTION
007232  000017   ;173232 000017
                 ;173234                 10$:
007234  032711   ;173234 032711                  BIT     #RXDONE,(R1)     ;WAIT UNTIL ERROR ASSEMBLED
007236  000040   ;173236 000040
007240  001775   ;173240 001775                  BEQ     10$              ;
007242  011200   ;173242 011200                  MOV     (R2),R0          ;GET ERROR REGISTER
007244  000541   ;173244 000541                  BR      HALTED           ;HALT AND DISPLAY ERRORS
```

```
  4188          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 5
  4189
  4190          ;                                       .SBTTL  REGISTER SAVE ROUTINE
  4191                                                ;
  4192                                                ; REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS R0-R7
  4193                                                ; IN MEMORY AT 40-56 (LOCATION ROTOR7).
  4194                                                ;
  4195                                                ; CALLING SEQUENCE:
  4196                                                ;       MOV     R0,ROTOR7+0
  4197                                                ;       MOV     #RET,R0
  4198                                                ;       BR      REGSAV
  4199                                                ; RET:  <RETURN HERE>
  4200                                                ;
  4201                                                ; ALL REGISTERS RESTORED
  4202                                                ;
  4203                                                ;
  4204                  ;173246                       REGSAV:
  4205  007246  010037  ;173246 010037                       MOV     R0,ROTOR7+16    ;SAVE R0 AS PC IN 56
  4206  007250  000056  ;173250 000056
  4207  007252  012700  ;173252 012700                       MOV     #ROTOR7+16,R0   ;R0 NOW POINTS TO 56
  4208  007254  000056  ;173254 000056
  4209  007256  010640  ;173256 010640                       MOV     SP,-(R0)        ;SAVE SP IN 54
  4210  007260  010540  ;173260 010540                       MOV     R5,-(R0)        ;SAVE R5 IN 52
  4211  007262  010440  ;173262 010440                       MOV     R4,-(R0)        ;SAVE R4 IN 50
  4212  007264  010340  ;173264 010340                       MOV     R3,-(R0)        ;SAVE R3 IN 46
  4213  007266  010240  ;173266 010240                       MOV     R2,-(R0)        ;SAVE R2 IN 44
  4214  007270  010140  ;173270 010140                       MOV     R1,-(R0)        ;SAVE R1 IN 42
  4215  007272  014000  ;173272 014000                       MOV     -(R0),R0        ;RESTORE R0 FROM 40
  4216  007274  000177  ;173274 000177                       JMP     @ROTOR7+16      ;GO TO SAVED PC
  4217  007276  004556  ;173276 004556
```

```
4218          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 6
4219
4220          ;                               .SBTTL  TC11 DECTAPE BOOTSTRAP ROUTINES
4221                                  ;
4222                                  ; TC11 REGISTER DEFINITIONS
4223                                  ;
4224          ;               177340  TCEPA=  177340                  ;EXTERNAL PAGE ADDRESS OF TC-11
4225                                  ;
4226          ;               000000  TCST=   0                       ;STATUS REGISTER
4227          ;               100000          TCENDZ= BIT15           ;END-ZONE DETECTED
4228          ;               000002  TCCM=   2                       ;COMMAND REGISTER
4229          ;               100000          TCERR=  BIT15           ;ERROR
4230          ;               004000          TCREV=  BIT11           ;REVERSE DIRECTION (TOWARD FORWARD END-ZONE)
4231          ;               003400          TCUNIT= BIT10!BIT9!BIT8 ;UNIT SELECT
4232                                  ;
4233          ;               000200          TCRDY=  BIT7            ;READY
4234          ;               000016          TCFUNC= BIT3!BIT2!BIT1  ;FUNCTION:
4235          ;               000000                  TCSATM= 0*BIT1  ; STOP ALL TAPE MOTION
4236          ;               000002                  TCRNUM= 1*BIT1  ; READ BLOCK NUMBER
4237          ;               000004                  TCREAD= 2*BIT1  ; READ DATA
4238          ;               000001          TCGO=   BIT0            ;START FUNCTION
4239          ;               000004  TCWC=   4                       ;WORD COUNT REGISTER
4240          ;               000006  TCBA=   6                       ;BUS ADDRESS REGISTER
4241                                  ;
4242                                  ; REGISTER USAGE:
4243                                  ;       R0 -- UNIT #
4244                                  ;       R1 -- ADDRESS OF TCCM
4245                                  ;       R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
4246                                  ;
4247                                  ; HERE TO START ROM TO BOOT FROM DECTAPE # 0, AS IF
4248                                  ;  DECTAPE BUTTON WERE PUSHED, IN CASE FLOPPY EXISTS.
4249                                  ;
4250                  ;173300         TCBOT0:
4251  007300  005000  ;173300 005000         CLR     R0              ;HERE TO START WITH A FLOPPY, FROM UNIT 0
4252  007302  005005  ;173302 005005         CLR     R5              ;
4253                                  ;       BR      TCBOOT          ;GO BOOT
4254                                  ;
4255                                  ; HERE TO BOOT FROM BLOCK 0 OF DECTAPE, UNIT # IN R0
4256                                  ;
4257                  ;173304         TCBOOT:
4258  007304  012706  ;173304 012706         MOV     #RETRY,SP       ;INIT RETRY COUNTER
4259  007306  000014  ;173306 000014
4260  007310  012701  ;173310 012701         MOV     #TCEPA+TCCM,R1  ;POINT TO COMMAND REGISTER
4261  007312  177342  ;173312 177342
4262                                  ;       BR      TCRTRY          ;TRY IT
4263                                  ;
4264                                  ; HERE ON ERROR TO RETRY
4265                                  ;
4266                  ;173314         TCRTRY:
4267  007314  005705  ;173314 005705         TST     R5              ;INDEFINITE RETRY?
4268  007316  100402  ;173316 100402         BMI     10$             ;YES-- TRY HARDER
4269  007320  005306  ;173320 005306         DEC     SP              ;NO-- DECREMENT COUNT
4270  007322  002427  ;173322 002427         BLT     TCEHLT          ;TOO MANY-- GIVE UP
4271                  ;173324         10$:
```

```
 4272  007324  000005  ;173324 000005          RESET                   ;CLEAR TC11
 4273  007326  010003  ;173326 010003          MOV     R0,R3           ;GET UNIT NUMBER
 4274  007330  000303  ;173330 000303          SWAB    R3              ;TO BITS 10-8
 4275  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
                                ;
                ;173402         TCEHLT:
007402  016100  ;173402 016100          MOV     TCST-TCCM(R1),R0 ;GET STATUS REGISTER
007404  177776  ;173404 177776
007406  000460  ;173406 000460          BR      HALTED          ;AND STOP
```

```
4308          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 7
4309
4310          ;                               .SBTTL  RP04 DISK BOOTSTRAP ROUTINES
4311                                          ;
4312                                          ; RP04 REGISTER DEFINITIONS
4313                                          ;
4314          ;                 176700  RPEPA=  176700                  ;EXTERNAL PAGE ADDRESS OF RP04 REGISTERS
4315                                          ;
4316          ;                 000000  RPCS1=  0                       ;OFFSET FOR CSR #1
4317          ;                 040000          RPTRE=  BIT14           ;TRANFER ERRROR
4318          ;                 020000          RPMCPE= BIT13           ;MASSBUS CONTROL PARITY ERROR
4319          ;                 004000          RPDVA=  BIT11           ;DRIVE AVAILABLE (TO -11)
4320          ;                 000200          RPRDY=  BIT7            ;FUNCTION COMPLETE
4321          ;                 000076          RPFUNC= BIT5!BIT4!BIT3!BIT2!BIT1 ;FUNCTION:
4322          ;                 000020                  RPPRST= 20      ; READ-IN PRESET
4323          ;                 000060                  RPWRIT= 60      ; WRITE DATA
4324          ;                 000070                  RPREAD= 70      ; READ DATA
4325          ;                 000001          RPGO=   BIT0            ;GO
4326          ;                 000002  RPWC=   2                       ;WORD COUNT REGISTER
4327          ;                 000006  RPDA=   6                       ;TRACK (HIGH BYTE) SECTOR (LOW BYTE)
4328          ;                 000010  RPCS2=  10                      ;CONTROL AND STATUS REGISTER #2
4329          ;                 000007          RPUNIT= BIT2!BIT1!BIT0  ;UNIT #
4330          ;                 000012  RPDS=   12                      ;DRIVE STATUS REGISTER
4331          ;                 100000          RPATA=  BIT15           ;ATTENTION ACTIVE
4332          ;                 040000          RPERR=  BIT14           ;DRIVE ERROR
4333          ;                 000014  RPER1=  14                      ;ERROR REGISTER #1
4334          ;                 000020          RPFER=  BIT4            ;FORMAT ERROR
4335          ;                 000032  RPOF=   32                      ;OFFSET REGISTER
4336          ;                 010000          RPFM22= BIT12           ;22 SECTOR (16 BIT) FORMAT
4337          ;                 004000          RPECCI= BIT11           ;INHIBIT ECC CORRECTION
4338          ;                 000034  RPDC=   34                      ;DESIRED CYLINDER
4339                                          ;
4340                                          ;
4341                                          ; REGISTER USAGE:
4342                                          ;       R0 --   UNIT #
4343                                          ;       R1 --   ADDRESS OF RPCS1
4344                                          ;       R2 --   DATA FOR RPOF: RPECCI (ECC INHIBIT), RPFM22 (22 SECTOR FORMAT)
4345                                          ;       R5 --   PARAMETER WORD SAVED FROM INITIALIZATION
4346                                          ;       SP --   RETRY COUNTER
4347                                          ;
4348                                          ;
4349                                          ; HERE TO BOOT FROM RP04-- UNIT # IN R0
4350                                          ;
4351                                          ; START RP04 GOING ON BOOT
4352                                          ;
4353                          ;173410         RPBOOT:
4354  007410  012706          ;173410 012706          MOV     #RETRY,SP       ;RETRY RETRY TIMES
4355  007412  000014          ;173412 000014
4356  007414  012701          ;173414 012701          MOV     #RPEPA+RPCS1,R1 ;ADDRESS RPCS1 IN R1
4357  007416  176700          ;173416 176700
4358  007420  012702          ;173420 012702          MOV     #RPECCI,R2      ;SET ECC INHIBIT, 20 SECTOR MODE
4359  007422  004000          ;173422 004000
4360                                          ;       BR      RPRTRY          ;FALL THROUGH RETRY CODE
4361                                          ;
```

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

```
                        ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 7-1
                        ;173430 005306          DEC     SP              ;RETRY COUNT EXHAUSTED?
                        ;173432 002444          BLT     RPEHLT          ;YES-- GIVE UP
                        ;173434                 10$:
                        ;173434 000005          RESET                   ;ZAP!!
                        ;173436 110061          MOVB    R0,RPCS2(R1)    ;SELECT PROPER UNIT #
                        ;173440 000010
                        ;173442 032711          BIT     #RPDVA,(R1)     ;IS DRIVE AVAILABLE TO US?
                        ;173444 004000
                        ;173446 001766          BEQ     RPRTRY          ;NO-- TRY AGAIN
                        ;173450 012711          MOV     #RPPRST+RPGO,(R1) ;DO 'READ-IN PRESET' FUNCTION
                        ;173452 000021
                        ;173454 005061          CLR     RPDC(R1)        ;SET CYLINDER 0
                        ;173456 000034
                        ;173460 005061          CLR     RPDA(R1)        ; TRACK 0, SECTOR 0
                        ;173462 000006
                        ;173464 050261          BIS     R2,RPOF(R1)     ;SET INHIBIT ECC, 22-SECTOR FORMAT (IF FORMAT ERROR)
                        ;173466 000032
                        ;173470 012761          MOV     #-256.,RPWC(R1) ;SET UP WORD COUNT TO PROPER VALUE
                        ;173472 177400
                        ;173474 000002
                        ;                                               ;NOTE THAT IT IS NOT NECCESARY TO SET UP BUS
                        ;                                               ; ADDRESS, SINCE IT IS 0 AFTER READ-IN PRESET
                        ;173476 012711          MOV     #RPREAD+RPGO,(R1) ;START READ FUNCTION
                        ;173470 000071
                        ;173502                 20$:
                        ;173502 105711          TSTB    (R1)            ;READY?
                        ;173504 100376          BPL     20$             ;NO-- WAIT UNTIL IT IS
                        ;173506 032761          BIT     #RPFER,RPER1(R1) ;FORMAT ERROR?
                        ;173510 000020
                        ;173512 000014
                        ;173514 001403          BEQ     30$             ;NO-- TRY AGAIN
                        ;173516 052702          BIS     #RPFM22,R2      ;YES-- TRY FOR 22 SECTOR FLAVOR
                        ;173520 010000
                        ;173522 000740          BR      RPRTRY          ;TRY AGAIN
                        ;173524                 30$:
                        ;173524 032711          BIT     #RPTRE!RPMCPE,(R1) ;TRANSFER OR MBC PARITY ERROR?
                        ;173526 060000
                        ;173530 001335          BNE     RPRTRY          ;YES-- ERROR-- TRY AGAIN
                        ;173532 032761          BIT     #RPATA!RPERR,RPDS(R1) ;ATTN OR OTHER ERROR?
                        ;173534 140000
                        ;173536 000012
                        ;173540 001331          BNE     RPRTRY          ;YES-- ERROR-- TRY AGAIN
                                                BR      CLRPC           ;NO ERRORS-- GO TO LOCATION 0
                                                ;
                                                ; HERE TO GO TO 0
                                                ;
                        ;173542                 CLRPC:
                        ;173542 005007          CLR     PC              ;JMP 0
                                                ;
                                                ; HERE ON ERROR FROM RP04 AFTER RETRYING-- DISPLAY DRIVE STATUS IN R0
                                                ;
```

```
 4421                           ;173544          RPEHLT:
 4422  007544  016100           ;173544 016100           MOV     RPDS(R1),R0     ;DISPLAY DRIVE STATUS
 4423  007546  000012           ;173546 000012
 4424                                            ;       BR      HALTED          ;R.I.P.
 4425                                            ;
 4426                           ;173550          HALTED:
 4427  007550  000000           ;173550 000000           HALT                    ;DIE
 4428  007552  000776           ;173552 000776           BR      HALTED          ;STAY DEAD
```

```
4429          ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 8
4430
4431          ;                               .SBTTL  INTERNAL BUTTON #4 -- DUMP AND BOOTSTRAP THROUGH DTE20
4432                                        ;
4433                                        ; DTE20 DEFINITIONS
4434                                        ;
4435                                        ; NOTE THAT ALL DTE20 REGISTER DEFINITIONS AND BIT DEFINITIONS
4436                                        ;  ARE IN $DEF IN SYSMAC.SML
4437                                        ;
4438          ;                 000040  DTESIZ= 40                      ;EACH DTE OCCUPIES 20 WORDS IN EXTERNAL PAGE
4439          ;                 000004  DTEMAX= 4                       ;MAX OF 4 DTE'S ON A PDP-11
4440                                        ;
4441                                        ; BUTTON #4 -- INITIATED BY '-10 RELOAD -11' BIT
4442                                        ;
4443                                        ;
4444                    ;173554             BUTON4:
4445  007554  010037   ;173554 010037              MOV     R0,ROTOR7+0     ;SAVE R0 IN 40
4446  007556  000040   ;173556 000040
4447  007560  012700   ;173560 012700              MOV     #10$,R0         ;SET RETURN ADDRESS IN R0
4448  007562  173566   ;173562 173566
4449  007564  000630   ;173564 000630              BR      REGSAV          ;SAVE R1-R7
4450                                        ;
4451                                        ; REGISTERS SAVED-- LOOK FOR THE DTE20 WHICH PUSHED THE BUTTON
4452                                        ;
4453                                        ; THE DTE WHICH PUSHED THE BUTTON SHOULD HAVE THE DOORBELL
4454                                        ;  RINGING AND HAVE THE VALUE 1365 (OCTAL) IN IT'S
4455                                        ;  TO -10 BYTE COUNT TO10BC.
4456                                        ;
4457                                        ; NXM (TIME-OUT) TRAP IS USED TO SKIP NON-EXISTANT DTE20'S.
4458                                        ;
4459                    ;173566             10$:
4460  007566  005005   ;173566 005005              CLR     R5              ;ADDRESS LOCATION ZERO
4461  007570  012500   ;173570 012500              MOV     (R5)+,R0        ;SAVE 0 IN R0
4462  007572  012501   ;173572 012501              MOV     (R5)+,R1        ;SAVE 2 IN R1
4463  007574  011502   ;173574 011502              MOV     (R5),R2         ;SAVE 4 IN R2
4464  007576  012725   ;173576 012725              MOV     #21$,(R5)+      ;SET NXM TRAP ADDRESS IN 4
4465  007600  173612   ;173570 173612
4466  007602  011503   ;173602 011503              MOV     (R5),R3         ;SAVE 6 IN R3
4467  007604  005015   ;173604 005015              CLR     (R5)            ;SET PS FOR TRAP
4468                                        ;
4469                                        ; LOOP THROUGH ALL DTE'S
4470                                        ;
4471                    ;173606             20$:
4472  007606  012704   ;173606 012704              MOV     #DLYCNT-DTESIZ,R4 ;POINT TO DTE # -1'S DELAY COUNT REGISTER
4473  007610  174340   ;173610 174340
4474          ;                                                   ; (WILL BUMP TO # 0)
4475                                        ;
4476                                        ; HERE ON NXM TRAP-- RESET SP AND TRY NEXT DTE
4477                                        ;
4478                    ;173612             21$:
4479  007612  012706   ;173612 012706              MOV     #4,SP           ;SET SP TO 4, STACK IS LOCATIONS 2 AND 0
4480  007614  000004   ;173614 000004
4481                    ;173616             22$:
4482  007616  062704   ;173616 062704              ADD     #DTESIZ,R4      ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS
```

```
4483  007620  000040  ;173620 000040
4484  007622  105704  ;173622 105704          TSTB    R4              ;IS THIS THE END OF THE DTE'S?
4485                  ;                                               ; NOTE THAT THE LAST DTE IS AT 774540
4486                  ;                                               ; AND THAT NOW R4= 774600 IF END
4487  007624  100770  ;173624 100770          BMI     20$             ;YES-- START ALL OVER, UNTIL A DTE
4488                  ;                                               ; SAYS HE PUSHED THE BUTTON
4489  007626  032764  ;173626 032764          BIT     #TO11DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?
4490  007630  004000  ;173630 004000
4491  007632  000034  ;173632 000034
4492  007634  001770  ;173634 001770          BEQ     22$             ;NO-- TRY NEXT DTE
```

```
  4493                  ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 8-1
  4494
  4495  007636  026417  ;173636 026417          CMP     T010BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365
  4496  007640  000014  ;173640 000014                                         ; IN IT'S TO -10 BYTE COUNT?
  4497                  ;
  4498  007642  001365  ;173642 001365          BNE     22$             ;NO-- TRY ANOTHER DTE
  4499                                        ;
  4500                                        ; WE HAVE FOUND THE DTE WHICH PUSHED THE BUTTON
  4501                                        ;
  4502                                        ; ADDRESS OF DLYCNT REGISTER IS IN R4
  4503                                        ;
  4504  007644  010315  ;173644 010315          MOV     R3,(R5)         ;RESTORE LOCATION 6
  4505  007646  010245  ;173646 010245          MOV     R2,-(R5)        ; 4
  4506  007650  010145  ;173650 010145          MOV     R1,-(R5)        ; 2
  4507  007652  010045  ;173652 010045          MOV     R0,-(R5)        ; 0
  4508                                        ;
  4509                                        ; SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
  4510                                        ;  IN LOCATIONS 130-156
  4511                                        ;
  4512  007654  012700  ;173654 012700          MOV     #DTESAV,R0      ;POINT TO SAVE AREA
  4513  007656  000130  ;173656 000130
  4514                  ;173660               29$:
  4515  007660  012420  ;173660 012420          MOV     (R4)+,(R0)+     ;SAVE A REGISTER
  4516  007662  022700  ;173662 022700          CMP     #T011DT-DLYCNT+DTESAV,R0 ;FINISHED?
  4517  007664  000156  ;173664 000156
  4518  007666  103374  ;173666 103374          BHIS    29$             ;NO-- SAVE SOME MORE
  4519                                        ;
  4520                                        ; R4= T011DT+2
  4521                                        ;
  4522                                        ; SET R1= STATUS REGISTER
  4523                                        ;     R4= DIAG2 REGISTER
  4524                                        ;
  4525                                        ; DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
  4526                                        ;  LOADED FLAG
  4527                                        ;
  4528                  ;173670               ADDX    DIAG2-T011DT-2,R4 ;R4 POINTS TO DIAG2 REGISTER
  4529  007670  005724  ;173670 005724          TST     (R4)+
  4530  007672  010401  ;173672 010401          MOV     R4,R1           ; SO DOES R1
  4531  007674  012700  ;173674 012700          MOV     #DRESET,R0      ;SETUP R0 FOR 'DIAGNOSTIC RESET'
  4532  007676  000100  ;173676 000100
  4533  007700  010021  ;173700 010021          MOV     R0,(R1)+        ;R1 POINTS TO STATUS REGISTER
  4534                                        ;
  4535                                        ; REGISTERS:
  4536                                        ;       R0 --   DRESET (DIAGNOSTIC RESET FUNCTION)
  4537                                        ;       R1 --   STAT (STATUS REGISTER)
  4538                                        ;       R4 --   DIAG2 (DIAGNOSTIC REGISTER #2, WHERE DRESET IS)
  4539                                        ;
  4540                                        ; THE -10 WILL NOW START READING -11 MEMORY, AS SOON AS WE SET
  4541                                        ;  THE TO -10 ADDRESS.  WHEN FINISHED, THE -10 WILL RING OUR DOORBELL.
  4542                                        ;
  4543  007702  005061  ;173702 005061          CLR     DLYCNT-STAT(R1) ;SET DTE20 FOR MAXIMUM DELAY (ZERO)
  4544  007704  177744  ;173704 177744
  4545  007706  005061  ;173706 005061          CLR     T010AD-STAT(R1) ;START DUMPING -11 MEMORY TO -10
  4546  007710  177764  ;173710 177764
```

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

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

F10


```
                        ;BM873H - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(666)  17-AUG-76  16:19  PAGE 9

                        ;                                       .SBTTL  FILL TO END OF ROM
                                                        ;
                                                        ;
                                ;173744                         FILLTO  1000
007744  000000            ;173744         000             .BYTE   0
                                ;173745         000             .BYTE   0
007746  000000            ;173746         000             .BYTE   0
                                ;173747         000             .BYTE   0
007750  000000            ;173750         000             .BYTE   0
                                ;173751         000             .BYTE   0
007752  000000            ;173752         000             .BYTE   0
                                ;173753         000             .BYTE   0
007754  000000            ;173754         000             .BYTE   0
                                ;173755         000             .BYTE   0
007756  000000            ;173756         000             .BYTE   0
                                ;173757         000             .BYTE   0
007760  000000            ;173760         000             .BYTE   0
                                ;173761         000             .BYTE   0
007762  000000            ;173762         000             .BYTE   0
                                ;173763         000             .BYTE   0
007764  000000            ;173764         000             .BYTE   0
                                ;173765         000             .BYTE   0
007766  000000            ;173766         000             .BYTE   0
                                ;173767         000             .BYTE   0
007770  000000            ;173770         000             .BYTE   0
                                ;173771         000             .BYTE   0
007772  000000            ;173772         000             .BYTE   0
                                ;173773         000             .BYTE   0
007774  000000            ;173774         000             .BYTE   0
                                ;173775         000             .BYTE   0
007776  END.YH:
007776  000000            ;173776         000             .BYTE   0
                                ;173777         000             .BYTE   0
                                                        ;
                                                        ;
                                                        ;
                                ;174000                 PASS2:
                        ;                       000001          .END
```

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

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

                        ;                               .SBTTL  PARAMETERS AND MACROS
                                                ;
                                                ; PARAMETERS
                                                ;
                        ;               173000          B$$873= 173000          ;START OF ROM
                                        ;*      D$$BUG= 73000           ;DEBUG VERSION IF DEFINED
                                        ;*      P$$ROM= 0               ;WRITE-ME-ON-DISK "ROM" IF DEFINED
                                        ;*      E$$DSP= 0               ;DISPLAY ERROR STATUS IN R0 FOR ERRORS
                        ;               000020          R$$TRY= 20              ;RETRY COUNTER *** MUST BE EVEN & .GE. 20 ***
                        ;               000040          R$$GSV= 40              ;SAVE GENERAL REGISTERS
                        ;               000000          H$$ALT= 0               ;HALT WHEN "SWR" BUTTON PUSHED
                                                ;
                        ;               174400          D$$T20= 174400          ;DTE20 SUPPORT
                        ;               000130           S$$DTE=130             ;SAVE DTE20 REGISTERS DLYCNT TO TO11DT
                        ;               176700          R$$P04= 176700          ;RP04/RP06 SUPPORT
                        ;               000000           F$$M20=0               ;20 SECTOR (18-BIT) FORMAT
                        ;               000000           F$$M22=0               ;22 SECTOR (16-BIT) FORMAT
                        ;               177170          R$$X11= 177170          ;RX11 (FLOPPY DISK) SUPPORT
                        ;               177340          T$$C11= 177340          ;TC11 (DECTAPE) SUPPORT
                        ;               177560          D$$L11= 177560          ;DL11 (REMOTE LOAD LINE) SUPPORT
                        ;               000000           D$$CMP=0               ;DOCMP STYLE (WHOLE HEADER) FOR DATA
                                                ;
                                                ; DEFINE ASSEMBLY CONDITIONS
                                                ;
                                                        .ENABLE AMA
                        ;                               .DSABLE GBL
                        ;                               .LIST   MEB
                                                ;
                                                ; MISC. MCALL'S
                                                ;
                        ;                               .MCALL   CALL,RETURN
                                                ;
                                                ; MACROS
                                                ;
                                                ; MACRO TO FILL TO A LOCATION (RELATIVE TO ROM ORIGIN) WITH ZERO BYTES
                                                ;
                        ;                               .MACRO  FILLTO  LOC
                        ;                       .IF DF B$$873
                        ;                       .IF GE <LOC>-<.-BM873>
                        ;                         .IF G <LOC>-<.-BM873>
                        ;                           .IF DF BM873E
                        ;                             .PRINT  <LOC>-<.-BM873> ;FREE BYTES AT LOC
                        ;                           .ENDC
                        ;                         .ENDC
                        ;                         .REPT <LOC>-<.-BM873>
                        ;                             .BYTE   0
                        ;                         .ENDR
                        ;                       .IFF
                        ;                             .ERROR  <.-BM873>-<LOC> ;BOUNDARY EXCEEDED AT LOC
                        ;                       .ENDC
                        ;                       .ENDC ; .IF DF B$$873
                        ;                             .ENDM   FILLTO
```

```
4689                                    ;
4690                                    ;
```

```
4691          ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 2-1
4692
4693                                  ; MACRO TO DO 'MOV #XXX,DEST' OR 'CLR DEST' IF XXX IS ZERO
4694                                  ;
4695          ;                               .MACRO  MOVO    XXX,DEST
4696          ;                       .IFEQ XXX
4697          ;                               CLR     DEST
4698          ;                       .IFF
4699          ;                               MOV     #XXX,DEST
4700          ;                       .ENDC
4701          ;                               .ENDM   MOVO
4702                                  ;
4703                                  ;
4704                                  ; MACRO TO ADD A SMALL NUMBER TO A REGISTER
4705                                  ;  GENERATES ONE OF THE FOLLOWING:
4706                                  ;       CMP     -(REG),-(REG)   ; -4
4707                                  ;       TST     -(REG)          ; -2
4708                                  ;       CMPB    -(REG),-(REG)   ; -2 (REGISTER MAY BE ODD)
4709                                  ;       DEC     REG             ; -1
4710                                  ;       <NOTHING>               ; 0
4711                                  ;       INC     REG             ; 1
4712                                  ;       TST     (REG)+          ; 2
4713                                  ;       CMPB    (REG)+,(REG)+   ; 2 (REGISTER MAY BE ODD)
4714                                  ;       CMP     (REG)+,(REG)+   ; 4
4715                                  ;       ADD     #XXX,REG        ; ANYTHING ELSE
4716                                  ;
4717                                  ; USE THIS MACRO WITH CAUTION, SINCE IT REFERENCES MEMORY
4718                                  ;  AND ALSO DOES NOT SET THE CONDITION CODES PROPERLY
4719                                  ;
4720          ;                               .MACRO  ADDX    XXX,REG,ODD
4721          ;                       SSS=.
4722          ;                       .IFEQ XXX+4
4723          ;                         .IF B <ODD>
4724          ;                               CMP     -(REG),-(REG)
4725          ;                         .ENDC
4726          ;                       .ENDC
4727          ;                       .IFEQ XXX+2
4728          ;                         .IF B <ODD>
4729          ;                               TST     -(REG)
4730          ;                         .IFF
4731          ;                               CMPB    -(REG),-(REG)
4732          ;                         .ENDC
4733          ;                       .ENDC
4734          ;                       .IFEQ XXX+1
4735          ;                               DEC     REG
4736          ;                       .ENDC
4737          ;                       .IFEQ XXX
4738          ;                         SSS=SSS+2
4739          ;                       .ENDC
4740          ;                       .IFEQ XXX-1
4741          ;                               INC     REG
4742          ;                       .ENDC
4743          ;                       .IFEQ XXX-2
4744          ;                         .IF B <ODD>
```

```
4745                ;                                   TST     (REG)+
4746                ;                             .IFF
```

```
  4747          ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 2-2
  4748
  4749          ;                               CMPB    (REG)+,(REG)+
  4750          ;                             .ENDC
  4751          ;                           .ENDC
  4752          ;                           .IFEQ XXX-4
  4753          ;                             .IF B <ODD>
  4754          ;                               CMP     (REG)+,(REG)+
  4755          ;                             .ENDC
  4756          ;                           .ENDC
  4757          ;                           .IFEQ SSS-.
  4758          ;                               ADD     #XXX,REG
  4759          ;                           .ENDC
  4760          ;                               .ENDM   ADDX
  4761                                      ;
  4762                                      ; GENERAL BIT DEFINITIONS
  4763                                      ;
  4764          ;               000001  BIT0=   000001
  4765          ;               000002  BIT1=   000002
  4766          ;               000004  BIT2=   000004
  4767          ;               000010  BIT3=   000010
  4768          ;               000020  BIT4=   000020
  4769          ;               000040  BIT5=   000040
  4770          ;               000100  BIT6=   000100
  4771          ;               000200  BIT7=   000200
  4772          ;               000400  BIT8=   000400
  4773          ;               001000  BIT9=   001000
  4774          ;               002000  BIT10=  002000
  4775          ;               004000  BIT11=  004000
  4776          ;               010000  BIT12=  010000
  4777          ;               020000  BIT13=  020000
  4778          ;               040000  BIT14=  040000
  4779          ;               100000  BIT15=  100000
  4780                                      ;
  4781                                      ; CPU REGISTER DEFINITIONS
  4782                                      ;
  4783          ;               177570  SWR=    177570                  ;SWITCH REGISTER
  4784          ;               177776  PS=     177776                  ;PROCESSOR STATUS WORD
  4785          ;               000340          PR7=    7*8IT5          ;PRIORITY 7
```

```
                        ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 3
                        ;                               .SBTTL  EXTERNAL BUTTONS #1, #2, #3
                                                        ;
                                                        ; ESTABLISH ROM ORIGIN
                                                        ;
                        ;               001     .IF DF B$$873
                        ;               002     .IF DF D$$BUG
                        ;                       .=.+D$$BUG-2
                        ;                       START:
                        ;                               HALT
                        ;                       .IFF ; .IF DF D$$BUG
                        ;           173000'     .=.+B$$873
                        ;       ;173000'                START:
                        ;               001     .ENDC ; .IF DF D$$BUG
                        ;                       .IFF ; .IF DF B$$873
                        ;               002     .IF DF P$$ROM
                        ;                       START:
                        ;                       .=.+P$$ROM
                        ;                               MOVO    P$$ROM,R0       ;START AT ROM START
                        ;                               MOV     (R0)+,2000-2(R0) ;MOVE A WORD
                        ;                               CMP     R0,#BM873E      ;AT END YET?
                        ;                               BLO     10$             ;NO-- BACK FOR MORE
                        ;                               JMP     BM873+2000      ;GO TO NEW CODE
                        ;                       .IFF ; .IF DF P$$ROM
                        ;                       START:
                        ;               001     .ENDC ; .IF DF P$$ROM
                        ;               000     .ENDC ; .IF DF B$$873
                                                        ;
                                ;173000'                BM873:
                                                        ;
                                                        ; BUTTON #1 -- LOAD USING SWITCH REGISTER
                                                        ;
                                ;173000'                BUTON1:
                        ;               001     .IF DF R$$GSV
 010000  010037 ;173000'                        010037          MOV     R0,R$$GSV+0     ;SAVE R0 IN LOCATION 40
 010002  000040 ;173002 000040
                        ;                       .IFTF ; .IF DF R$$GSV
 010004  013700 ;173004'                        013700          MOV     @#SWR,R0        ;GET SWITCH REGISTER
 010006  177570 ;173006 177570
 010010  032700 ;173010'                        032700          BIT     #BIT0,R0        ;IS LOW-ORDER BIT SET?
 010012  000001 ;173012 000001
 010014  001007 ;173014'                        001007          BNE     BUTONX          ;YES-- LOOK AT CONTENTS
                        ;                       .IFT ; .IF DF R$$GSV
 010016  000510 ;173016'                        000510          BR      REGSAV          ;NO-- SAVE R1-R7 IN 42-56, GO TO ADDRESS IN R0 (
                        ;                       .IFF ; .IF DF R$$GSV
                        ;                               JMP     (R0)            ;GO TO ADDRESS FROM SWR
                        ;               000     .ENDC ; .IF DF R$$GSV
                                                        ;
                                                        ; BUTTON #3 -- LOAD BOOT FROM RX11 FLOPPY DISK, OR TC11 DECTAPE, OR DL11
                                                        ;
                                ;173020'                BUTON3:
 010020  005000 ;173020'                        005000          CLR     R0              ;SAY LOAD FROM FLOPPY, UNIT 0
 010022  000404 ;173022'                        000404          BR      BUTONX          ;GO TO COMMON CODE FOR 3 BUTTONS
```

```
                                        ;
                                        ; REQUIRED POWER-FAIL VECTOR
                                        ;
                                                FILLTO  24
                        ;173024'        173000'         .WORD   BM873,PR7
 010024  173000         ;173024'
 010026  000340         ;173026 000340
```

```
                ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 3-1

                                                ;
                                                ; BUTTON #2 -- LOAD BOOT FROM RP04 DISK
                                                ;
                        ;173030'                        BUTON2:
        010030  012700  ;173030'                012700          MOV     #BIT7,R0        ;BIT 7 MEANS LOAD FROM RP04
        010032  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 (OR TC11 DECTAPE)
                                                ;       BIT 7 = 1       LOAD FROM RP04 DISK
                                                ;       BIT 15 = 1      INDEFINITE RETRY
                                                ;
                        ;173034'                        BUTONX:
        010034  010005  ;173034'                010005          MOV     R0,R5           ;SAVE PARAMETER FOR BOOT
        010036  106300  ;173036'                106300          ASLB    R0              ;LEFT-ALIGN SPEED FIELD IN LOW BYTE
        010040  122700  ;173040'                122700          CMPB    #3*BIT4,R0      ;IS SPEED 0, 1, OR 2?
        010042  000060  ;173042 000060
        010044  101001  ;173044'                101001          BHI     10$             ;YES-- UNIT IS UNIT TO USE
        010046  005000  ;173046'                005000          CLR     R0              ;NO-- USE UNIT #0
                        ;173050'                        10$:
        010050  000300  ;173050'                000300          SWAB    R0              ;GET UNIT # IN LOW BYTE
        010052  042700  ;173052'                042700          BIC     #^C7,R0         ;TRIM TO 3 BITS 2, 1, 0
        010054  177770  ;173054 177770
                                                ;
                                                ; UNIT # IS IN R0-- CALL PROPER BOOT DEPENDING ON BIT 7
                                                ;
                ;                       001     .IF DF R$$P04
        010056  105705  ;173056'                105705          TSTB    R5              ;WHERE SHOULD WE BOOT FROM?
        010060  100551  ;173060'                100551          BMI     RPBOOT          ;BIT 7 = 1 -- BOOT FROM RP04 DISK
                ;                       000     .ENDC ; .IF DF R$$P04
                                                ;
                                                ; BIT 7 = 0 -- BOOT FROM RX11 IF IT EXISTS, ELSE TC11, ELSE DL11
                                                ;
                ;                       001     .IF DF R$$X11
                ;                       002      .IF DF T$$C11
        010062  012737  ;173062'                012737  000004          MOV     #TCBOOT,@#4     ;BOOT FROM DECTAPE IF NO FLOPPY
        010064  173274  ;173064 173274
        010066  000004  ;173066
        010070  005037  ;173070'                005037          CLR     @#6             ;SET PS OF TIMEOUT TRAP
        010072  000006  ;173072 000006
                ;                               .IIF NDF R$$TRY, MOV #40,SP     ;SET UP TEMP STACK
                ;                                .IFF ; .IF DF T$$C11
                ;                       003       .IF DF D$$L11
                ;                                       MOV     #DLBOOT,@#4     ;BOOT FROM DL11 IF NO FLOPPY
                ;                                       CLR     @#6             ;SET PS OF TIMEOUT TRAP
                ;                               .IIF NDF R$$TRY, MOV #40,SP     ;SET UP TEMP STACK
                ;                       002       .ENDC ; .IF DF D$$L11
                ;                       001      .ENDC ; .IF DF T$$C11
```

```
4900                                    ;       BR      RXBOOT          ;BOOT FROM FLOPPY TO START
4901                                    .IFF ; .IF DF R$$X11
4902            ;               002      .IF DF T$$C11
4903            ;               003      .IF DF D$$L11
4904            ;                               MOV     #DLBOOT,@#4     ;BOOT FROM DL11 IF NO DECTAPE
4905            ;                               CLR     @#6             ;SET PS OF TIMEOUT TRAP
4906            ;                       .IIF NDF R$$TRY, MOV #40,SP     ;SET UP TEMP STACK
4907            ;               002      .ENDC ; .IF DF D$$L11
```

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

  4908          ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 3-2
  4909
  4910                                  ;       BR      TCBOOT          ;BOOT FROM DECTAPE TO START
  4911          ;                        .IFF ; .IF DF T$$C11
  4912          ;                 003     .IF DF D$$L11
  4913          ;                         BR      DLBOOT          ;BOOT FROM DL11
  4914          ;                        .IFF ; .IF DF D$$L11
  4915          ;                               .ERROR  ; MUST HAVE EITHER RX11, TC11 OR DL11
  4916          ;                 002     .ENDC ; .IF DF D$$L11
  4917          ;                 001   .ENDC ; .IF DF T$$C11
  4918          ;                 000  .ENDC ; .IF DF R$$X11
```

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

                  ;                         .SBTTL  RX11 FLOPPY DISK BOOTSTRAP ROUTINE
                                          ;
                  ;                  001  .IF DF R$$X11
                                          ;
                                          ; RX11 REGISTER DEFINITIONS
                                          ;
                  ;               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
                                          ;
                                          ;
                                          ;  NOTE THAT THE BOOTSTRAP IS WRITTEN IN LOGICAL BLOCK 0
                                          ;  WHICH IS TRACK 1, SECTORS 1, 3, 5, 7.  ONLY SECTOR 1
                                          ;  IS READ BY THE ROM.
                                          ;
                                          ;
                                          ; REGISTER USAGE:
                                          ;       R0 --   UNIT #
                                          ;       R1 --   ADDRESS OF RXCS
                                          ;       R2 --   READ FUNCTION WITH UNIT SELECT SET
                                          ;       R3 --   ADDRESS OF RXDB
                                          ;       R4 --   DATA ADDRESS (TO READ OR WRITE)
                                          ;       R5 --   PARAMETER WORD SAVED FROM INITIALIZATION
                                          ;       SP --   RETRY COUNTER
                                          ;
                                          ;
                                          ; HERE TO BOOT FROM RX11 FLOPPY DISK-- UNIT # IN R0
                                          ;
                          ;173074'                 RXBOOT:
                  ;                  002  .IF DF R$$TRY
 010074   012706  ;173074'                012706          MOV     #R$$TRY,SP      ;SET RETRY COUNT
 010076   000020  ;173076 000020
                  ;                       .IFTF ; .IF DF R$$TRY
 010100   012701  ;173100'                012701          MOV     #R$$X11+RXCS,R1 ;ADDRESS CONTROL STATUS REGISTER FOR RX11
 010102   177170  ;173102 177170
                  ;                  003  .IF DF T$$C11!D$$L11
 010104   005711  ;173104'                005711          TST     (R1)            ;RX11 EXIST?
                  ;                  002  .ENDC ; .IF DF T$$C11!D$$L11
                                          ;       BR      RXRTRY          ;FALL THROUGH RETRY CHECK
                                          ;
                                          ; HERE ON ERROR TO RETRY
                                          ;
```

```
4973                    ;173106'              RXRTRY:
4974            .                       .IFT ;  .IF DF R$$TRY
4975  010106  005705  ;173106'        005705            TST     R5              ;INDEFINITE RETRY?
4976  010110  100402  ;173110'        100402            BMI     RXRSET          ;YES-- TRY FAITHFULLY
```

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

```
5031                                    ;
5032                                    ; HERE TO START ROM TO BOOT FROM DECTAPE # 0, AS IF
5033                                    ;  DECTAPE BUTTON WERE PUSHED, IN CASE FLOPPY EXISTS.
5034                                    ;
5035            ;                   002 .IF DF T$$C11
5036                        ;173214'        BUTONS:
5037  010214    005000    ;173214'        005000          CLR     R0              ;HERE TO START WITH A FLOPPY, FROM UNIT 0
5038  010216    000425    ;173216'        000425          BR      TCBOTO          ;BOOT FROM TAPE
5039            ;           001 .ENDC ; .IF DF T$$C11
```

```
                        ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 4-2
                        
                        .IFTF ; .IF DF R$$X11
                    002 .IF DF H$$ALT
                        ;
                        ; BUTTON #1 (OPTIONAL) -- HALT, THEN LOAD USING SWITCH REGISTER
                        ;
                ;173220'        BUTONO:
010220  000000  ;173220'        000000          HALT                    ;HALT NOW
010222  000666  ;173222'        000666          BR      BUTON1          ;BUT LOOK AT SWR LATER
                    001 .ENDC ; .IF DF H$$ALT
                        ;
                        ; REQUIRED POWER-FAIL VECTOR
                        ;
                ;173224'                        FILLTO  224
010224  173230  ;173224'        173000'         .WORD   BM873,PR7
010226  000340  ;173226 000340
                        .IFT ; .IF DF R$$X11
                        ;
                        ; HERE ON ERROR AFTER RETRYING -- DISPLAY ERROR REGISTER AND HALT
                        ;
                    002 .IF DF E$$DSP
                        RXEHLT:
                                MOV     #RXRERR+RXGO,(R1) ;DO A READ ERROR REGISTER FUNCTION
                        10$:
                                BIT     #RXDONE,(R1)    ;WAIT UNTIL ERROR ASSEMBLED
                                BEQ     10$             ;
                                MOV     (R3),R0         ;GET ERROR REGISTER
                                BR      HALTED          ;HALT AND DISPLAY ERRORS
                    001 .ENDC ; .IF DF E$$DSP
                    000 .ENDC ; .IF DF R$$X11
```

```
5071            ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 5
5072
5073            ;                               .SBTTL  INTERNAL BUTTON #4 -- DUMP AND BOOTSTRAP THROUGH DTE20
5074                                    ;
5075                                    ; BUTTON #4 -- INITIATED BY '-10 RELOAD -11' BIT
5076                                    ;
5077            ;                   001 .IF DF D$$T20&R$$GSV
5078                                        BUTON4:
5079  010230  010037  ;173230'     010037          MOV     R0,R$$GSV+0     ;SAVE R0 IN 40
5080  010232  000040  ;173232 000040
5081  010234  012700  ;173234'     012700          MOV     #DTEDMP,R0      ;SET RETURN ADDRESS IN R0
5082  010236  173622  ;173236 173622
5083                                    ;       BR      REGSAV          ;SAVE R1-R7
5084            ;                   000 .ENDC ; .IF DF D$$T20&R$$GSV
```

```
5085            ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 6
5086
5087            ;                               .SBTTL  REGISTER SAVE ROUTINE
5088                                    ;
5089            ;                   001 .IF DF R$$GSV
5090                                    ;
5091                                    ; REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS R0-R7
5092                                    ; IN MEMORY AT 40-56 (LOCATION R$$GSV).
5093                                    ;
5094                                    ; CALLING SEQUENCE:
5095                                    ;       MOV     R0,R$$GSV+0
5096                                    ;       MOV     #RET,R0
5097                                    ;       BR      REGSAV
5098                                    ; RET:  <RETURN HERE>
5099                                    ;
5100                                    ; ALL REGISTERS RESTORED
5101                                    ;
5102                                    ;
5103                    ;173240'        REGSAV:
5104    010240  010037  ;173240'        010037          MOV     R0,R$$GSV+16    ;SAVE R0 AS PC IN 56
5105    010242  000056  ;173242 000056
5106    010244  012700  ;173244'        012700          MOV     #R$$GSV+16,R0   ;R0 NOW POINTS TO 56
5107    010246  000056  ;173246 000056
5108    010250  010640  ;173250'        010640          MOV     SP,-(R0)        ;SAVE SP IN 54
5109    010252  010540  ;173252'        010540          MOV     R5,-(R0)        ;SAVE R5 IN 52
5110    010254  010440  ;173254'        010440          MOV     R4,-(R0)        ;SAVE R4 IN 50
5111    010256  010340  ;173256'        010340          MOV     R3,-(R0)        ;SAVE R3 IN 46
5112    010260  010240  ;173260'        010240          MOV     R2,-(R0)        ;SAVE R2 IN 44
5113    010262  010140  ;173262'        010140          MOV     R1,-(R0)        ;SAVE R1 IN 42
5114    010264  014000  ;173264'        014000          MOV     -(R0),R0        ;RESTORE R0 FROM 40
5115    010266  000177  ;173266'        000177          JMP     @R$$GSV+16      ;GO TO SAVED PC
5116    010270  004564  ;173270 000056
5117            ;                   000 .ENDC ; .IF DF R$$GSV
```

```
                        ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(563)  20-JAN-77  11:51  PAGE 7

                        ;                               .SBTTL  TC11 DECTAPE BOOTSTRAP ROUTINE
                        ;
                        ;                   001         .IF DF T$$C11
                        ;
                        ;                               ; TC11 REGISTER DEFINITIONS
                        ;
                        ;               000000          TCST=   0                       ;STATUS REGISTER
                        ;               100000                  TCENDZ= BIT15           ;END-ZONE DETECTED
                        ;               000002          TCCM=   2                       ;COMMAND REGISTER
                        ;               100000                  TCERR=  BIT15           ;ERROR
                        ;               004000                  TCREV=  BIT11           ;REVERSE DIRECTION (TOWARD FORWARD END-ZONE)
                        ;               003400                  TCUNIT= BIT10!BIT9!BIT8 ;UNIT SELECT
                        ;
                        ;               000200                  TCRDY=  BIT7            ;READY
                        ;               000016                  TCFUNC= BIT3!BIT2!BIT1  ;FUNCTION:
                        ;               000000                          TCSATM= 0*BIT1  ; STOP ALL TAPE MOTION
                        ;               000002                          TCRNUM= 1*BIT1  ; READ BLOCK NUMBER
                        ;               000004                          TCREAD= 2*BIT1  ; READ DATA
                        ;               000001                  TCGO=   BIT0            ;START FUNCTION
                        ;               000004          TCWC=   4                       ;WORD COUNT REGISTER
                        ;               000006          TCBA=   6                       ;BUS ADDRESS REGISTER
                        ;                               ; REGISTER USAGE:
                        ;                                       R0 -- UNIT #
                        ;                                       R1 -- ADDRESS OF TCCM
                        ;                                       R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
                        ;                                       SP -- RETRY COUNTER
                        ;
                        ;                               ; HERE TO CONTINUE WITH BOOT FROM UNIT 0 IF FLOPPY EXISTS, UNIT # IN R0
                        ;
                        ;                   002         .IF DF R$$X11
                        ;               ;173272'                TCBOT0:
010272  005005  ;173272'                005005                  CLR     R5              ;SET SWR PARAMETER TO ZERO, ALSO
                        ;                               ;       BR      TCBOOT          ;ALL SET!
                        ;                   001         .ENDC ; .IF DF R$$X11
                        ;
                        ;                               ; HERE TO BOOT FROM BLOCK 0 OF DECTAPE, UNIT # IN R0
                        ;
                        ;               ;173274'                TCBOOT:
                        ;                   002         .IF DF D$$L11
010274  012737  ;173274'                012737  000004          MOV     #DLBOOT,@#4     ;IN CASE NO DTA, TRY DL11
010276  173530  ;173276 173530
                        ;                   001         .ENDC ; .IF DF D$$L11
                        ;                   002         .IF DF R$$TRY
010300  000004  ;173300
010302  012706  ;173302'                012706                  MOV     #R$$TRY,SP      ;INIT RETRY COUNTER
010304  000020  ;173304 000020
                        ;                               .IFTF ; .IF DF R$$TRY
010306  012701  ;173306'                012701                  MOV     #T$$C11+TCCM,R1 ;POINT TO COMMAND REGISTER
010310  177342  ;173310 177342
                        ;               ;       BR      TCRTRY          ;TRY IT
```

```
5172                                    ;
5173                                    ; HERE ON ERROR TO RETRY
5174                                    ;
5175                    ;173312'        TCRTRY:
5176                                003 .IF DF RSSTRY!DSSL11
5177  010312  105011    ;173312'        105011          CLRB    (R1)            ;STOP ALL TAPE MOTION, TRAP IF NO TC11
```

```
                        ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 7-1
                        ;
                        ;                   002  .ENDC ; .IF DF R$$TRY!D$$L11
                        ;                        .IFT ; .IF DF R$$TRY
010314  005705  ;173314'          005705          TST     R5              ;INDEFINITE RETRY?
010316  100402  ;173316'          100402          BMI     10$             ;YES-- TRY HARDER
010320  005306  ;173320'          005306          DEC     SP              ;NO-- DECREMENT COUNT
010322  002426  ;173322'          002426          BLT     TCEHLT          ;TOO MANY-- GIVE UP
                        ;                   001  .ENDC ; .IF DF R$$TRY
                        ;173324'                 10$:
010324  000005  ;173324'          000005          RESET                   ;CLEAR TC11
010326  110061  ;173326'          110061          MOVB    R0,1(R1)        ;SELECT PROPER UNIT
010330  000001  ;173330 000001
010332  052711  ;173332'          052711          BIS     #TCREV+TCRNUM+TCGO,(R1) ;START TAPE TOWARD FOWARD END-ZONE (BLOC
010334  004003  ;173334 004003
                ;173336'                 20$:
010336  005711  ;173336'          005711          TST     (R1)            ;ERROR?
010340  100376  ;173340'          100376          BPL     20$             ;NO-- WAIT FOR END-ZONE ERROR
010342  005761  ;173342'          005761          TST     TCST-TCCM(R1)   ;END-ZONE UP YET?
010344  177776  ;173344 177776
010346  100361  ;173346'          100361          BPL     TCRTRY          ;NO-- MUST BE OTHER ERROR
                ;                        .
010350  012761  ;173350'          012761          MOV     #-256.,TCWC-TCCM(R1) ;SET WORD COUNT
010352  177400  ;173352 177400
010354  000002  ;173354 000002
                ;                                                ;NOTE THAT "RESET" CLEARS BUS ADDRESS REGISTER.
010356  042711  ;173356'          042711          BIC     #TCREV,(R1)     ;SET FORWARD MODE
010360  004000  ;173360 004000
010362  112711  ;173362'          112711          MOVB    #TCREAD+TCGO,(R1) ;START READ, FORWARD
010364  000005  ;173364 000005
                ;173366'                    30$:
010366  105711  ;173366'          105711          TSTB    (R1)            ;TRANSFER DONE?
010370  100376  ;173370'          100376          BPL     30$             ;NO-- WAIT SOME MORE
010372  005711  ;173372'          005711          TST     (R1)            ;YES-- ERROR?
010374  100746  ;173374'          100746          BMI     TCRTRY          ;YES-- RETRY
010376  005007  ;173376'          005007          CLR     PC              ;NO-- DONE-- GOTO LOCATION 0
                                                 ;
                                                 ; HERE ON TC11 ERROR
                                                 ;
                        ;                   002  .IF DF E$$DSP
                        ;                        TCEHLT:
                        ;                                MOV     TCST-TCCM(R1),R0 ;GET STATUS REGISTER
                        ;                                BR      HALTED          ;AND STOP
                        ;                   001  .ENDC ; .IF DF E$$DSP
                        ;                   000  .ENDC ; .IF DF T$$C11
```

```
                        ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 8
                        ;
                        ;                                       .SBTTL  ERROR HANDLING
                                                                ;
                        ;                               001     .IF DF RSSTRY
                        ;                               002     .IF DF ESSDSP
                        ;                                       .IFF
                                ;173400'                        .IIF DF RSSPO4, RPEHLT:
                                ;173400'                        .IIF DF RSSX11, RXEHLT:
                                ;173400'                        .IIF DF TSSC11, TCEHLT:
                                ;173400'                        .IIF DF DSSL11, DLEHLT:
                        ;                               001     .ENDC ; .IF DF ESSDSP
                                                                ;
                                ;173400'                        HALTED:
010400  000000          ;173400'                000000          HALT                    ;DIE
010402  000776          ;173402'                000776          BR      HALTED          ;STAY DEAD
                        ;                                       .IFF ; .IF DF RSSTRY
                        ;                                       .IIF DF ESSDSP, .ERROR ; CANNOT HAVE ESSDSP WITHOUT RSSTRY
                        ;                               000     .ENDC ; .IF DF RSSTRY
```

```
                          ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  .1:51  PAGE 9

                          ;                                  .SBTTL  RPO4 DISK BOOTSTRAP ROUTINE
                                                   ;
                          ;                       001  .IF DF R$$PO4
                                                   ;
                                                   ; RPO4 REGISTER DEFINITIONS
                                                   ;
                          ;                    000000  RPCS1=  0                            ;OFFSET FOR CSR #1
                          ;                    040000          RPTRE=  BIT14                ;TRANFER ERRROR
                          ;                    020000          RPMCPE= BIT13                ;MASSBUS CONTROL PARITY ERROR
                          ;                    004000          RPDVA=  BIT11                ;DRIVE AVAILABLE (TO -11)
                          ;                    000200          RPRDY=  BIT7                 ;FUNCTION COMPLETE
                          ;                    000076          RPFUNC= BIT5!BIT4!BIT3!BIT2!BIT1 ;FUNCTION:
                          ;                    000020                  RPPRST= 20           ; READ-IN PRESET
                          ;                    000060                  RPWRIT= 60           ; WRITE DATA
                          ;                    000070                  RPREAD= 70           ; READ DATA
                          ;                    000001          RPGO=   BIT0                 ;GO
                          ;                    000002  RPWC=   2                            ;WORD COUNT REGISTER
                          ;                    000006  RPDA=   6                            ;TRACK (HIGH BYTE) SECTOR (LOW BYTE)
                          ;                    000010  RPCS2=  10                           ;CONTROL AND STATUS REGISTER #2
                          ;                    000007          RPUNIT= BIT2!BIT1!BIT0       ;UNIT #
                          ;                    000012  RPDS=   12                           ;DRIVE STATUS REGISTER
                          ;                    100000          RPATA=  BIT15                ;ATTENTION ACTIVE
                          ;                    040000          RPERR=  BIT14                ;DRIVE ERROR
                          ;                    000014  RPER1=  14                           ;ERROR REGISTER #1
                          ;                    000020          RPFER=  BIT4                 ;FORMAT ERROR
                          ;                    000032  RPOF=   32                           ;OFFSET REGISTER
                          ;                    010000          RPFM22= BIT12                ;22 SECTOR (16 BIT) FORMAT
                          ;                    004000          RPECCI= BIT11                ;INHIBIT ECC CORRECTION
                          ;                    000034  RPDC=   34                           ;DESIRED CYLINDER
                                                   ;
                                                   ;
                                                   ; REGISTER USAGE:
                                                   ;       R0 --   UNIT #
                                                   ;       R1 --   ADDRESS OF RPCS1
                                                   ;       R2 --   DATA FOR RPOF: RPECCI (ECC INHIBIT), RPFM22 (22 SECTOR FORMAT)
                                                   ;       R5 --   PARAMETER WORD SAVED FROM INITIALIZATION
                                                   ;       SP --   RETRY COUNTER
                                                   ;
                                                   ;
                                                   ; HERE TO BOOT FROM RPO4-- UNIT # IN R0
                                                   ;
                                                   ; START RPO4 GOING ON BOOT
                                                   ;
                                    ;173404'               RPBOOT:
                          ;                       002  .IF DF R$$TRY
 010404   012706            ;173404'               012706          MOV     #R$$TRY,SP      ;RETRY RETRY TIMES
 010406   000020            ;173406 000020
                          ;                            .IFTF ; .IF DF R$$TRY
 010410   012701            ;173410'               012701          MOV     #R$$PO4+RPCS1,R1        ;ADDRESS RPCS1 IN R1
 010412   176700            ;173412 176700
                          ;                       003  .IF DF F$$M20&F$$M22
 010414   012702            ;173414'               012702          MOV     #RPECCI,R2      ;SET ECC INHIBIT, 20 SECTOR MODE
```

```
  5296  010416  004000  ;173416 004000
  5297                  ;               002  .ENDC ; .IF DF F$$M20&F$$M22
  5298                                       ;       BR      RPRTRY          ;FALL THROUGH RETRY CODE
  5299                                       ;
  5300                                       ; HERE ON ERROR TO RETRY
```

```
5301            ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 9-1
5302
5303                                    ;
5304                    ;173420'                RPRTRY:
5305                    .                       .IFT ; .IF DF R$$TRY
5306    010420  005705  ;173420'                005705          TST     R5              ;INFINITE RETRY?
5307    010422  100402  ;173422'                100402          BMI     RPRSET          ;YES-- TRY AGAIN
5308    010424  005306  ;173424'                005306          DEC     SP              ;RETRY COUNT EXHAUSTED?
5309    010426  002764  ;173426'                002764          BLT     RPEHLT          ;YES-- GIVE UP
5310                    ;                   001 .ENDC ; .IF DF R$$TRY
5311                                            ;
5312                                            ; HERE TO GO ON WITH RP04
5313                                            ;
5314                    ;173430'                RPRSET:
5315    010430  000005  ;173430'                000005          RESET                   ;ZAP!!
5316    010432  110061  ;173432'                110061          MOVB    R0,RPCS2(R1)    ;SELECT PROPER UNIT #
5317    010434  000010  ;173434 000010
5318    010436  012711  ;173436'                012711          MOV     #RPPRST+RPGO,(R1) ;DO 'READ-IN PRESET' FUNCTION
5319    010440  000021  ;173440 000021
5320    010442  005061  ;173442'                005061          CLR     RPDC(R1)        ;SET CYLINDER 0
5321    010444  000034  ;173444 000034
5322    010446  005061  ;173446'                005061          CLR     RPDA(R1)        ; TRACK 0, SECTOR 0
5323    010450  000006  ;173450 000006
5324                    ;                   002 .IF DF F$$M20&F$$M22
5325    010452  050261  ;173452'                050261          BIS     R2,RPOF(R1)     ;SET INHIBIT ECC, 22-SECTOR FORMAT (IF FORMAT ER
5326    010454  000032  ;173454 000032
5327                    ;                       .IFF
5328                    ;                   003 .IF DF F$$M20
5329                    ;                       BIS     #RPECCI,RPOF(R1) ;SET ECC INHIBIT, 20 SECTOR FORMAT
5330                    ;                   002 .ENDC ; .IF DF F$$M20
5331                    ;                   003 .IF DF F$$M22
5332                    ;                       BIS     #RPECCI+RPFM22,RPOF(R1) ;SET ECC INHIBIT, 22 SECTOR FORMAT
5333                    ;                   002 .ENDC ; .IF DF F$$M22
5334                    .                       .IFTF ; .IF DF F$$M20&F$$M22
5335    010456  012761  ;173456'                012761          MOV     #-256.,RPWC(R1) ;SET UP WORD COUNT TO PROPER VALUE
5336    010460  177400  ;173460 177400
5337    010462  000002  ;173462 000002
5338                    ;                                                       ;NOTE THAT IT IS NOT NECCESARY TO SET UP BUS
5339                    ;                                                       ; ADDRESS, SINCE IT IS 0 AFTER READ-IN PRESET
5340    010464  012711  ;173464'                012711          MOV     #RPREAD+RPGO,(R1)';START READ FUNCTION
5341    010466  000071  ;173466 000071
5342                    ;173470'                20$:
5343    010470  105711  ;173470'                105711          TSTB    (R1)            ;READY?
5344    010472  100376  ;173472'                100376          BPL     20$             ;NO-- WAIT UNTIL IT IS
5345                    .                       .IFT ; .IF DF F$$M20&F$$M22
5346    010474  032761  ;173474'                032761          BIT     #RPFER,RPER1(R1) ;FORMAT ERROR?
5347    010476  000020  ;173476 000020
5348    010500  000014  ;173470 000014
5349    010502  001402  ;173502'                001402          BEQ     30$             ;NO-- TRY AGAIN
5350    010504  052702  ;173504'                052702          BIS     #RPFM22,R2      ;YES-- TRY FOR 22 SECTOR FLAVOR
5351    010506  010000  ;173506 010000
5352                    ;                   001 .ENDC ; .IF DF F$$M20&F$$M22
5353                    ;173510'                30$:
5354    010510  032711  ;173510'                032711          BIT     #RPTRE!RPMCPE,(R1) ;TRANSFER OR MBC PARITY ERROR?
```

```
5355  010512  060000  ;173512 060000
5356  010514  001341  ;173514'          001341          BNE     RPRTRY          ;YES-- ERROR-- TRY AGAIN
5357  010516  032761  ;173516'          032761          BIT     #RPATA!RPERR,RPDS(R1) ;ATTN OR OTHER ERROR?
5358  010520  140000  ;173520 140000
5359  010522  000012  ;173522 000012
5360  010524  001335  ;173524'          001335          BNE     RPRTRY          ;YES-- ERROR-- TRY AGAIN
5361                                    ;       BR      CLRPC           ;NO ERRORS-- GO TO LOCATION 0
5362          ;                   000   .ENDC ; .IF DF R$$P04
5363                                    ;
5364                                    ; HERE TO GO TO 0
5365                                    ;
5366                  ;173526'          CLRPC:
5367  010526  005007  ;173526'          005007          CLR     PC              ;JMP 0
5368                                    ;
5369                                    ; HERE ON ERROR FROM RP04 AFTER RETRYING-- DISPLAY DRIVE STATUS IN R0
5370                                    ;
```

```
5371            ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 9-2
5372
5373            ;                   001  .IF DF ESSDSP
5374            ;                        RPEHLT:
5375            ;                                MOV     RPDS(R1),R0     ;DISPLAY DRIVE STATUS
5376            ;                                BR      HALTED          ;R.I.P.
5377            ;                   000  .ENDC ; .IF DF ESSDSP
```

```
                        ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 10

                        ;                               .SBTTL  DL11 ASYNCHONOUS LINE BOOTSTRAP ROUTINE
                        ;
                        ;                   001         .IF DF D$$L11
                        ;
                                                        ; DL11 REGISTER DEFINITIONS
                        ;
                        ;               000000          DLRCSR= 0                       ;RECIEVER CONTROL/STATUS REGISTER
                        ;               000200                  DLRDON= BIT7            ;DONE
                        ;               000002          DLRBUF= 2                       ;RECIEVER DATA BUFFER
                                                        ;
                                                        ; DL11 BOOT IS OF THE FOLLOWING FORMAT:
                                                        ;
                                                        ;       ... <DLE><CC1><CC2><FILL><... DATA ...>
                                                        ;
                                                        ; WHERE <DLE> IS 220(8).
                                                        ;       <CC1>, <CC2> FORM THE LOW AND HIGH BYTES OF THE BYTE COUNT FOR THE
                                                        ;               FOLLOWING DATA, RESPECTIVELY.  THE TWO HIGH-ORDER
                                                        ;               BITS (QSYNC & SELECT) ARE IGNORED (D$$CMP DEFINED).
                                                        ;       <FILL> IS 5 BYTES OF IGNORED DATA (D$$CMP DEFINED).
                                                        ;       <DATA> IS A BYTE STREAM OF 8-BIT DATA OF LENGTH DETERMINED BY
                                                        ;               <CC2><CC1>.
                                                        ;
                                                        ; REGISTER USAGE:
                                                        ;       R1 -- ADDRESS OF DLRCSR
                                                        ;       R2 -- ADDRESS OF "DLCHAR" ROUTINE
                                                        ;       R3 -- BYTE COUNT
                                                        ;       R4 -- MEMORY ADDRESS
                                                        ;       R5 -- PARAMETER WORD SAVED FROM INITIALIZATION
                                                        ;       SP -- STACK POINTER (STACK AT 5004-)
                                                        ;
                                                        ;
                                                        ; START HERE TO BOOT FROM DL11
                                                        ;
                        ;173530'                        BUTON6:
                        ;173530'                        DLBOOT:
010530  012701  ;173530'                012701          MOV     #D$$L11+DLRCSR,R1 ;GET DL11 EXTERNAL PAGE ADDRESS
010532  177560  ;173532 177560
010534  011706  ;173534'                011706          MOV     (PC),SP         ;SET TEMP STACK AT 5004- *** CLR R4 MUST BE NEXT
010536  005004  ;173536'                005004          CLR     R4              ;RESET MEMORY ADDRESS FOR DLCHAR  *** DO NOT MOV
010540  012702  ;173540'                012702                  MOV     #DLCHAR,R2      ;SET ADDRESS OF GET CHARACTER ROUTINE
010542  173610  ;173542 173610
                ;173544'                        10$:
                ;173544'                                CALL    (R2)            ;GET A CHARACTER
010544  004712  ;173544'                004712          JSR PC,(R2)
010546  124427  ;173546'                124427          CMPB    -(R4),#220      ;DLE?
010550  000220  ;173550 000220
010552  001374  ;173552'                001374          BNE     10$             ;NO-- KEEP ON LOOKING
                ;                               002     .IF DF D$$CMP
                ;173554'                                CALL    -(R2)           ;GET TWO BYTES
010554  004742  ;173554'                004742          JSR PC,-(R2)
                ;                               .IFF ;  .IF DF D$$CMP
                ;                                       CALL    (R2)            ;GET TWO
```

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

```
                        ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 10-1
                        ;173564'                        CALL    (R2)            ;SKIP TWO MORE BYTES
010564  004712          ;173564'        004712          JSR PC,(R2)
                        ;173566'                        CALL    (R2)            ; AND TWO MORE
010566  004712          ;173566'        004712          JSR PC,(R2)
010570  005722          ;173570'        005722          TST     (R2)+           ;SKIP THE CALL (PC)
                        ;173572'                        CALL    (R2)            ; +1 MAKES 5 BYTES TO SKIP
010572  004712          ;173572'        004712          JSR PC,(R2)
010574  005004          ;173574'        005004          CLR     R4              ;RESET BACK TO 0
                        ;                               .IFTF ; .IF DF D$$CMP
                                                      20$:
                        ;173576'                        CALL    (R2)            ;GET A CHARACTER
                        ;173576'
010576  004712          ;173576'        004712          JSR PC,(R2)
010600  005303          ;173600'        005303          DEC     R3              ;REDUCE COUNT
010602  003375          ;173602'        003375          BGT     20$             ;BACK IF MORE
010604  005007          ;173604'        005007          CLR     PC              ;ELSE GO TO LOADED CODE
                                        ;
                                        ; DLCHAR -- GET CHARACTER FROM DL11
                                        ;       R1 -- ADDRESS OF DL11
                                        ;       R4 -- ADDRESS OF WHERE TO STORE CHARACTER, INCREMENTED
                                        ;
                                        ; CALL DLCHAR-2 FOR TWO BYTES (D$$CMP ONLY)
                                        ;
                        ;               .IFT ; .IF DF D$$CMP
                        ;173606'                        CALL    (PC)            ;GET A BYTE, THEN ANOTHER
010606  004717          ;173606'        004717          JSR PC,(PC)
                        ;                       001     .ENDC ; .IF DF D$$CMP
                        ;173610'                      DLCHAR:
010610  105711          ;173610'        105711          TSTB    (R1)            ;READY WITH A CHARACTER?
010612  100376          ;173612'        100376          BPL     DLCHAR          ;NO-- WAIT SOME MORE
010614  116124          ;173614'        116124          MOVB    DLRBUF-DLRCSR(R1),(R4)+ ;YES-- STORE THE CHARACTER
010616  000002          ;173616 000002
                        ;173620'                        RETURN                  ;AND RETURN FROM DLCHAR
010620  000207          ;173620'        000207          RTS PC
                        ;                       000     .ENDC ; .IF DF D$$L11
```

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

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

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

                ;                       .IFF ; .IF LT <S$$DTE-400>
                ;                            MOV     #4,R5              ;SET TO LOCATION 4
                ;                 002  .ENDC ; .IF LT <S$$DTE-400>
                ;                       .IFF ; .IF DF S$$DTE
                ;                            MOV     #2000-4,SP         ;POINT TO STACK WE'LL USE
                ;                            MOV     (SP)+,R0           ;SAVE 1774 IN R0
                ;                            MOV     (SP)+,R1           ;SAVE 1776 IN R1
                ;                            MOV     #4,R5              ;POINT TO TIME-OUT TRAP VECTOR
                ;                       .IFTF ; .IF DF S$$DTE
010630  011502  ;173630'              011502        MOV     (R5),R2            ;SAVE 4 IN R2
010632  012725  ;173632'              012725        MOV     #21$,(R5)+         ;SET NXM TRAP ADDRESS IN 4
010634  173646  ;173634 173646
010636  011503  ;173636'              011503        MOV     (R5),R3            ;SAVE 6 IN R3
010640  005015  ;173640'              005015        CLR     (R5)               ;SET PS FOR TRAP

                                      ;
                                      ; LOOP THROUGH ALL DTE'S
                                      ;
                ;173642'                    20$:
010642  012704  ;173642'              012704        MOV     #D$$T20+DLYCNT-DTESIZ,R4 ;POINT TO DTE # -1'S DELAY COUNT REGIST
010644  174340  ;173644 174340
                ;                                                              ; (WILL BUMP TO # 0)

                                      ;
                                      ; HERE ON NXM TRAP-- RESET SP AND TRY NEXT DTE
                                      ;
                ;173646'                    21$:
                ;                     .IFT ; .IF DF S$$DTE
                ;                 003 .IF LT <S$$DTE-400>
010646  012706  ;173646'              012706        MOV     #S$$DTE+10,SP   ;SET STACK TO SAVE AREA, WITH ROOM FOR FTL STK T
010650  000140  ;173650 000140
                ;                     .IFF ; .IF LT <S$$DTE-400>
                ;                            MOV     #S$$DTE+4,SP    ;SET STACK TO REGISTER SAVE AREA
                ;                 002 .ENDC ; .IF LT <S$$DTE-400>
                ;                     .IFF ; .IF DF S$$DTE
                ;                            MOV     #2000,SP        ;SET TEMP STACK (SAVED ABOVE)
                ;                     .IFTF ; .IF DF S$$DTE
                ;173652'                    22$:
010652  062704  ;173652'              062704        ADD     #DTESIZ,R4      ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS
010654  000040  ;173654 000040
010656  105704  ;173656'              105704        TSTB    R4              ;IS THIS THE END OF THE DTE'S?
                ;                                                     ; NOTE THAT THE LAST DTE IS AT 774540
                ;                                                     ; AND THAT NOW R4= 774600 IF END
010660  100770  ;173660'              100770        BMI     20$             ;YES-- START ALL OVER, UNTIL A DTE
                ;                                                     ; SAYS HE PUSHED THE BUTTON
010662  032764  ;173662'              032764        BIT     #T011DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?
010664  004000  ;173664 004000
010666  000034  ;173666 000034
010670  001770  ;173670'              001770        BEQ     22$             ;NO-- TRY NEXT DTE
010672  026417  ;173672'              026417        CMP     T010BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365
010674  000014  ;173674 000014
                ;                                                     ; IN IT'S TO -10 BYTE COUNT?

                                      ;
                                      ; *** THIS NEXT INSTRUCTION BETTER BE A 1365 !!! ***
```

```
5584
5585  010676  001365  ;173676'        001365          BNE     22$             ;NO-- TRY ANOTHER DTE
5586                                   ;
5587                                   ; WE HAVE FOUND THE DTE WHICH PUSHED THE BUTTON
5588                                   ;
5589                                   ; ADDRESS OF DLYCNT REGISTER IS IN R4
5590                                   ;
5591  010700  010315  ;173700'        010315          MOV     R3,(R5)         ;RESTORE LOCATION 6
5592  010702  010245  ;173702'        010245          MOV     R2,-(R5)        ; 4
```

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

                ;                       .IFF ; .IF DF S$$DTE
                ;                             MOV     R1,-(SP)        ; AND 1776
                ;                             MOV     R0,-(SP)        ; . . . 1774
                ;                       .IFT ; .IF DF S$$DTE
                ;                  003  .IF LT <S$$DTE-400>
000704  010145  ;173704'                010145          MOV     R1,-(R5)        ;RESTORE 2
000706  010045  ;173706'                010045          MOV     R0,-(R5)        ; AND 0
                                        ;
                                        ; SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
                                        ;  IN LOCATIONS 130-156
                                        ;
000710  012700  ;173710'                012700          MOV     #S$$DTE,R0      ;POINT TO SAVE AREA
000712  000130  ;173712 000130
                ;173714'                        29$:
000714  012420  ;173714'                012420          MOV     (R4)+,(R0)+     ;SAVE A REGISTER
000716  022700  ;173716'                022700          CMP     #T011DT-DLYCNT+S$$DTE,R0 ;FINISHED?
000720  000156  ;173720 000156
                ;                       .IFF ; .IF LT <S$$DTE-400>
                ;                             CMP     -(SP),-(SP)     ;BACK DOWN TO START OF SAVE AREA
                ;                       29$:
                ;                             MOV     (R4)+,(SP)+     ;SAVE A REGISTER
                ;                             CMP     #T011DT-DLYCNT+S$$DTE,SP ;FINISHED?
                ;                  002  .ENDC ; .IF LT <S$$DTE-400>
000722  103374  ;173722'                103374          BHIS    29$             ;NO-- SAVE SOME MORE
                                        ;
                                        ; R4= T011DT+2
                                        ;
                                        ; SET R1= STATUS REGISTER
                                        ;     R4= DIAG2 REGISTER
                                        ;
                                        ; DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
                                        ;  LOADED FLAG
                                        ;
                ;173724'                                ADDX    DIAG2-<T011DT+2>,R4 ;R4 POINTS TO DIAG2 REGISTER
000724  005724  ;173724'                005724          TST     (R4)+
                ;                       .IFF ; .IF DF S$$DTE
                ;                             ADDX    DIAG2-DLYCNT,R4 ;POINT R4 TO DIAG2 REGISTER
                ;                  001  .ENDC ; .IF DF S$$
000726  010401  ;173726'                010401          MOV     R4,R1           ; SO DOES R1
000730  012700  ;173730'                012700          MOV     #DRESET,R0      ;SETUP R0 FOR 'DIAGNOSTIC RESET'
000732  000100  ;173732 000100
000734  010021  ;173734'                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.
                                        ;
000736  005061  ;173736'                005061          CLR     DLYCNT-STAT(R1) ;SET DTE20 FOR MAXIMUM DELAY (ZERO)
```

```
5647  010740  177744  ;173740 177744
5648  010742  005061  ;173742'          005061          CLR     TO10AD-STAT(R1) ;START DUMPING -11 MEMORY TO -10
5649  010744  177764  ;173744 177764
5650                  ;                                       ; STARTING AT LOCATION 0
5651                  ;173746'                  30$:
5652  010746  032711  ;173746'          032711          BIT     #TO11DB,(R1)    ;IS DOORBELL RINGING (TRANSFER COMPLETE)?
5653  010750  004000  ;173750 004000
5654  010752  001775  ;173752'          001775          BEQ     30$             ;NO-- WAIT FOR DOORBELL
```

```
                        ;BM873J - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  20-JAN-77  11:51  PAGE 11-3
                        ;
010754  010014  ;173754'          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,
                        ;  WE WILL START EXECUTION OF THE LOADED CODE AT LOCATION 0.
                        ;
010756  005061  ;173756'          005061          CLR     TO11AD-STAT(R1) ;START INPUT TO LOCATION 0
010760  177766  ;173760 177766
010762  012761  ;173762'          012761          MOV     #IFLOP!<<-256.>&8COUNT>,TO11BC-STAT(R1) ;256 WORDS, INTERRUPT
010764  107400  ;173764 107400
010766  177762  ;173766 177762
                        ;                                                               ; -10 WHEN DONE
                        ;173770'                40$:
010770  105711  ;173770'          105711          TSTB    (R1)            ;TRANSFER COMPLETE?
010772  100376  ;173772'          100376          BPL     40$             ;NO-- WAIT SOME MORE
010774  005007  ;173774'          005007          CLR     PC              ;GO TO LOADED CODE, STARTING AT
                        ;                                               ; LOCATION 0
                        ;                   000 .ENDC ; .IF DF D$$T20
```

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

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

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

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

012000                                  RESTRT:
012000  012706  001100                          MOV     #$CMTAG,R6              ;FIRST LOCATION TO BE CLEARED
012004  005026                                  CLR     (R6)+                   ;CLEAR MEMORY LOCATION
012006  022706  001136                          CMP     #$TKS,R6                ;DONE?
012012  001374                                  BNE     .-6                     ;LOOP BACK IF NO
012014  012706  001100                          MOV     #STACK,SP               ;SETUP THE STACK POINTER
012020  012737  020534  000020                  MOV     #$SCOPE,@#IOTVEC        ;IOT VECTOR FOR SCOPE ROUTINE
012026  012737  000340  000022                  MOV     #340,@#IOTVEC+2         ;LEVEL 7
012034  012737  020630  000030                  MOV     #$ERROR,@#EMTVEC        ;EMT VECTOR FOR ERROR ROUTINE
012042  012737  000340  000032                  MOV     #340,@#EMTVEC+2         ;LEVEL 7
012050  012737  021160  000034                  MOV     #$TRAP,@#TRAPVEC        ;TRAP VECTOR FOR TRAP CALLS
012056  012737  000340  000036                  MOV     #340,@#TRAPVEC+2        ;LEVEL 7
012064  012767  012064  167014                  MOV     #.,$LPADR               ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
012072  005067  001372                          CLR     INITFG          ;INITIALIZE TO ASK WHICH TYPE
012076  005067  001356                          CLR     TABLE           ;INITIALIZE TO ASK WHICH TYPE
012102  012706  001100          START:  MOV     #STACK,SP       ;SET THE STACK POINTER
012106  005067  167104                  CLR     LSTERR          ;CLEAR ERROR FLG REPORT
012112  005037  000000                  CLR     @#0             ;SET FOR UNEXPECTED TRAP TO ADD 0
012116  012767  012102  005546          MOV     #START,PRG.NO   ;GET READY FOR PWR FAIL BEFORE FIRST TEST.
012124  012737  000006  000004          MOV     #6,@#4          ;SET TIME OUT TRAP VECTOR
012132  005037  000006                  CLR     @#6             ;SET TIME OUT STATUS TO 0
012136  005067  005052                  CLR     FLAG4           ;CLEAR TEST 4 INITIAL FLAG
012142  005737  000042                  TST     @#42            ;AM I RUNNING UNDER ACT-11??
012146  001002                          BNE     .+6             ;BR IF *WE ARE* UNDER ACT-11!!
012150  000167  000616                  JMP     CONT            ;JUMP IF NOT ACT-11
012154  013746  000004                  MOV     @#4,-(SP)       ;SAV TRAP POINTER
012160  012737  013474  000004          MOV     #NOROM,@#4      ;PUT IN A NEW ONE
012166  005737  173000                  TST     @#173000        ;TRY TO READ THE ROM
012172  000240                          NOP                     ;WAIT FOR POSSIBLE TRAP
012174  012637  000004                  MOV     (SP)+,@#4       ;IF NO TRAP RESTORE POINTER
012200  026737  167174  173000          CMP     MAP.YA,@#173000 ;DOES 1ST WORD COMPARE?
012206  001034                          BNE     64$             ;CHECK NEXT MAP
012210  013746  000004                  MOV     @#4,-(SP)       ;SAVE LOC 4
012214  012737  012236  000004          MOV     #65$,@#4        ;SET FOR TIMEOUT
012222  005737  173400                  TST     @#173400        ;READ FROM 173400
012226  000240                          NOP                     ;IF NO TIMEOUT, NOT YA
012230  012637  000004                  MOV     (SP)+,@#4       ;RESTORE LOC 4
012234  000421                          BR      64$
012236  022626                  65$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
012240  012637  000004                  MOV     (SP)+,@#4       ;RESTORE LOC 4
012244  012767  001400  001206          MOV     #MAP.YA,TABLE   ;1ST MAP ADDR
012252  012767  001776  001202          MOV     #END.YA,ALLEND  ;LAST MAP ADDR
012260  012767  173376  001204          MOV     #173376,LASTA   ;LAST ROM ADDR
012266  012767  000101  005520          MOV     #000101,VERSON  ;SET ROM TYPE
012274  000167  001416                  JMP     PRG1            ;START TEST 1
012300                                  64$:
012300  026737  167474  173000          CMP     MAP.YB,@#173000 ;DOES 1ST WORD COMPARE?
012306  001016                          BNE     69$             ;CHECK NEXT MAP
012310  012767  002000  001142          MOV     #MAP.YB,TABLE   ;1ST MAP ADDR
012316  012767  002776  001136          MOV     #END.YB,ALLEND  ;LAST MAP ADDR
```

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

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

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

```
 5909  013502  000776                          BR      .-2
 5910  013504  005015  051124  050101  NOROMS: .ASCII  <15><12>/TRAP TO 4 ON 1ST READ OF 173000/
       013545     015  044412  020123          .ASCIZ  <15><12>/IS LOADER INSTALLED?/
       013574  005015  040503  023516  NMATCH: .ASCII  <15><12>/CAN'T IDENTIFY LOADER AS YA,YB,YC,YD,YF,YG,YH OR YJ AFTER/
       013667     015  041412  050115          .ASCIZ  <15><12>/CMP WITH LOC 173000/
                013716                  .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.

013716 012767 013716 003746 PRG1:   MOV     #PRG1,PRG.NO    ;SET FOR PWR FAIL
013724 012767 000500 165266         MOV     #500,ICOUNT     ;DO THIS TEST 500(8) TIMES.
013732 012737 017674 000004 PRG.1:  MOV     #NO.TRAP,@#4    ;SET FOR UNEXPECTED TRAP.
013740 012700 173000                MOV     #173000,R0      ;SET BEGGINING ADDRESS
013744 012767 013770 165134         MOV     #2$,$LPADR      ;IF SW14=1; GOTO 2$ WHEN SCOPE IS HIT
013752 016704 177502                MOV     TABLE,R4        ;SET START OF MAP
013756 016767 177510 000322         MOV     LASTA,LAST      ;SET LAST ADDRESS
013764 012703 000005         1$:    MOV     #5.,R3          ;DO EACH ADDRESS 5 TIMES.
013770 022700 173024         2$:    CMP     #173024,R0      ;DON'T DO THE VECTOR ADD.
013774 001001                       BNE     20$             ;BR IF NOT THE VECTOR ADD.
013776 022024                       CMP     (R0)+,(R4)+     ;UPDATE TO NEXT ADDRESS
014000 022700 173224         20$:   CMP     #173224,R0      ;DON'T DO THE TRAP VECTORS
014004 001001                       BNE     21$             ;NO THIS ISN'T A TRAP VECTOR.
014006 022024                       CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS..
014010 010467 165110         21$:   MOV     R4,$GDDAT
014014 010067 165106                MOV     R0,$BDDAT
014020 011067 165202                MOV     (R0),TEMP4      ;READ THE ADDRESS
014024 011467 165174                MOV     (R4),TEMP3      ;READ THE SOFTWARE ADDRESS
014030 026767 165170 165170         CMP     TEMP3,TEMP4
014036 001401                       BEQ     22$             ;BR IF GOOD
014040 104001                       ERROR   1               ;INCORRECT COMPARISON.
014042 032767 004000 163520 22$:    BIT     #BIT11,SWR      ;QUICK PASS.?
014050 001002                       BNE     23$             ;BR IF YES
014052 005303                       DEC     R3              ;HAS THAT ADD BEEN READ 5 TIMES?
014054 001345                       BNE     2$              ;BR IF NOT 5 TIMES

014056 026700 000224         23$:   CMP     LAST,R0         ;WAS LAST ADDRESS CHECKED?
014062 001403                       BEQ     10$             ;BR IF YES
014064 000004                       SCOPE                   ;LOCK ON THIS ADDRESS IF SW14=1
014066 022024                       CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS.
014070 000735                       BR      1$              ;CONTINUE THE TEST.

014072 032767 000001 163470 10$:    BIT     #BIT0,SWR       ;EXTENDED WORD TO BE CHECKED?
014100 001413                       BEQ     3$              ;BR IF NO CHECKING.
014102 022767 173776 000176         CMP     #173776,LAST    ;IS ALL THE TEST DONE?
014110 001407                       BEQ     3$              ;BR IF YES.
014112 012767 173776 000166         MOV     #173776,LAST    ;SET LAST ADDRESS.
014120 016704 177340                MOV     EXTMAP,R4       ;SET EXTENDED MAP.
014124 005720                       TST     (R0)+           ;POP POINTER
014126 000716                       BR      1$              ;GO DO THE TEST.
```

```
                                        ;TEST THAT WRITTING ROM RESULTS IN A TIME OUT
                                        ;TRAP.
                                        
014130  012767  014154  164750  3$:     MOV     #5$,$LPADR      ;IF SW14=1 GOTO 5$ WHEN SCOPE IS HIT
014136  012700  173000                  MOV     #173000,R0      ;SET R0 WITH BASE ADDRESS OF ROM
014142  012737  014210  000004          MOV     #6$,@#4         ;SET FOR TIME OUT TRAP
014150  012703  000005          4$:     MOV     #5.,R3          ;DO EACH ADD 5 TIMES
014154  022700  173024          5$:     CMP     #173024,R0      ;CHECK FOR A TRAP VECTOR
014160  001001                          BNE     24$             ;BR IF NOT VECTOR
014162  005720                          TST     (R0)+           ;UPDATE THE REGISTER POINTER
014164  022700  173224          24$:    CMP     #173224,R0      ;CHECK FOR THE OTHER VECTOR
014170  001001                          BNE     25$             ;BR IF NOT THE VECTOR
014172  005720                          TST     (R0)+           ;UPDATE THE POINTER
014174  012710  177777          25$:    MOV     #-1,(R0)        ;WRITE ROM WITH A -1
014200  000240                          NOP                     ;WAIT ONE INSTR. TIME
014202  010067  165020                  MOV     R0,TEMP4
014206  104002                          ERROR   2               ;WRITING ROM DIDN'T TIME OUT.
014210  012706  001100          6$:     MOV     #STACK,SP       ;RESTORE STACK
014214  032767  004000  163346          BIT     #BIT11,SWR      ;QUICK PASS?
014222  001002                          BNE     30$
014224  005303                          DEC     R3              ;DO EACH ADD 5 TIMES
014226  001352                          BNE     5$              ;NOT DONE WITH THIS ONE YET.

014230  032767  000001  163332  30$:     BIT     #BIT0,SWR       ;EXTENDED 128. WORDS TO BE CHECKED?
014236  001404                          BEQ     31$             ;BR IF NO
014240  022700  173776                  CMP     #173776,R0      ;HAVE ALL 256. WORDS BEEN CHECKED?
014244  001407                          BEQ     7$              ;BR IF ALL DONE
014246  000403                          BR      32$             ;KEEP GOING
014250  026700  177216          31$:    CMP     LASTA,R0        ;ALL DONE??
014254  001403                          BEQ     7$              ;HAVE ALL 128. WORDS DONE?
014256  000004                  32$:    SCOPE                   ;CHECK SW14 FOR FREEZE!!
014260  005720                          TST     (R0)+           ;UPDATE TO NEXT ADDRESS
014262  000732                          BR      4$              ;GO DO IT AGAIN
014264  005367  164730          7$:     DEC     ICOUNT          ;ITERATION COUNT DONE?
014270  001004                          BNE     8$              ;BR IF NOT DONE.
014272  004767  003436                  JSR     PC,EOP          ;TYPE END MESSAGE
014276  000167  177414                  JMP     PRG1            ;GO DO IT AGAIN.
014302  000167  177424          8$:     JMP     PRG.1           ;GO RESTART.
014306  000000          LAST:   0

014310  005015  040515  047111  BM873X: .ASCII  <15><12>/MAINDEC-11-DZBMDJ/
014333     015  042012  053105          .ASCII  <15><12>/DEVICE VERSION/
014353     015  041012  034115          .ASCIZ  <15><12>/BM873-Y/
014365     015  025012  040454  BM.ERR: .ASCIZ  <15><12>/*,A,B,C,D,F,G,H,J ONLY./
014417     040  020040  042526  VERS:   .ASCIZ  /   VERSION: BM873-Y/
          014444                  .EVEN
```

```
                                                ;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.

014444  012767  014444  003220  PRG2:   MOV     #PRG2,PRG.NO    ;SET FOR POWER FAIL
014452  012737  017674  000004          MOV     #NO.TRAP,@#4    ;SET FOR UNEXPECTED TRAP TO 4
014460  016767  177006  177620          MOV     LASTA,LAST
014466  062767  000002  177612          ADD     #2,LAST
014474  012700  173000          21$:    MOV     #173000,R0      ;SET R0 WITH THE STARTING ROM ADD.
014500  016703  176754                  MOV     TABLE,R3        ;SET POINTER.
014504  104400  015024                  TYPE    ,DH.2           ;TYPE MESSAGE
014510  104400  015106                  TYPE    ,DH.2B          ;TYPE THE HEADER
014514  012767  000007  164500  1$:     MOV     #7,TEMP5        ;SET COUNTER
014522  011001                          MOV     (R0),R1         ;READ THE ROM
014524  010067  164474                  MOV     R0,TEMP3        ;STORE R0
014530  010167  164472                  MOV     R1,TEMP4        ;STORE R1
014534  022767  011000  176716          CMP     #MAP.Y.,TABLE   ;IF BM873.Y* SELECTED; FILL TABLE
014542  001001                          BNE     22$             ;BR IF NOT BM873.Y*
014544  011023                          MOV     (R0),(R3)+      ;FILL THE TABLE..
014546  005720                  22$:    TST     (R0)+           ;POP THE POINTER
014550  104400  017556                  TYPE    ,MCRLF

014554  016746  164444                  MOV     TEMP3,-(SP)
014560  104402                          TYPOC
014562  104400  017547                  TYPE    ,MSPACE         ;TYPE THREE SPACES.


014566  016746  164434                  MOV     TEMP4,-(SP)
014572  104402                          TYPOC
014574  011001                  7$:     MOV     (R0),R1         ;STORE ROM DATA
014576  010067  164422                  MOV     R0,TEMP3        ;STORE ROM ADDRESS
014602  010167  164420                  MOV     R1,TEMP4        ;PREPARE DATA FOR TYPE OUT
014606  022767  011000  176644          CMP     #MAP.Y.,TABLE   ;IS BM873.Y* SELECTED?
014614  001001                          BNE     23$             ;BR IF NO..
014616  011023                          MOV     (R0),(R3)+      ;FILL THE DATA TABLE
014620  005720                  23$:    TST     (R0)+           ;POP THE POINTER

014622  104400  017547                  TYPE    ,MSPACE

014626  016746  164374                  MOV     TEMP4,-(SP)
014632  104402                          TYPOC

014634  026700  177446                  CMP     LAST,R0 ;HAS THE HIGHEST LIMIT BEEN HIT?
014640  001404                          BEQ     2$              ;BR IF ALL DONE.
014642  005367  164354                  DEC     TEMP5           ;DECREASE COUNTER
014646  001352                          BNE     7$              ;BR IF NOT 0; KEEP GOING
014650  000721                          BR      1$              ;GO TYPE ADDRESS NOW

014652  032767  000001  162710  2$:     BIT     #BIT0,SWR       ;IS THE EXTENDED 128. WORDS TO BE CHECKED??
014660  001455                          BEQ     3$              ;BR IF NO.
014662  012700  173400                  MOV     #173400,R0      ;RESET POINTER OF ROM
014666  016703  176572                  MOV     EXTMAP,R3       ;SET SOFTWARE MAP POINTER
```

```
 6056  014672  104400  015220                        TYPE    ,DH.2A          ;TYPE NEW HEADER
 6057  014676  104400  015106                        TYPE    ,DH.2B          ;TYPE ADDRESS AND +XX
 6058  014702  012767  000007  164312  6$:           MOV     #7,TEMP5        ;SET TYPE OUT COUNTER
 6059  014710  011001                                MOV     (R0),R1         ;READ THE ROM
 6060  014712  010067  164306                        MOV     R0,TEMP3        ;STORE R0
 6061  014716  010167  164304                        MOV     R1,TEMP4        ;STORE R1
 6062  014722  012023                                MOV     (R0)+,(R3)+     ;STORE THE DATA IN SOFTWARE MAP
 6063  014724  104400  017556                        TYPE    ,MCRLF
 6064  014730  016746  164270                        MOV     TEMP3,-(SP)
 6065  014734  104402                                TYPOC
 6066
 6067  014736  104400  017547                        TYPE    ,MSPACE
 6068  014742  016746  164260                        MOV     TEMP4,-(SP)
 6069  014746  104402                                TYPOC
 6070
 6071  014750  011001                        8$:     MOV     (R0),R1         ;SAVE THE ROM DATA
 6072  014752  010067  164246                        MOV     R0,TEMP3        ;SAVE THE ROM ADDRESS
 6073  014756  010167  164244                        MOV     R1,TEMP4        ;SET DATA FOR TYPE OUT
 6074
 6075  014762  104400  017547                        TYPE    ,MSPACE
 6076
 6077  014766  016746  164234                        MOV     TEMP4,-(SP)
 6078  014772  104402                                TYPOC
 6079
 6080  014774  012023                                MOV     (R0)+,(R3)+     ;STORE THE DATA IN SOFTWARE TABLE
 6081  014776  022700  174000                        CMP     #174000,R0      ;HAS THE HIGHEST LIMIT BEEN HIT?
 6082  015002  001404                                BEQ     3$              ;BR IF ALL DONE.
 6083  015004  005367  164212                        DEC     TEMP5           ;DEC TABLE COUNTER
 6084  015010  001357                                BNE     8$              ;BR TO JUST TYPE DATA
 6085  015012  000733                                BR      6$              ;BR TO TYPE ADDRESS
 6086  015014  005000                        3$:     CLR     R0              ;CLEAR DATA LIGHTS
 6087  015016  000000                                HALT                    ;HIT CONTINUE TO PROCEED.
 6088  015020  000167  177420                        JMP     PRG2            ;GOTO PRG 2
 6089  015024  006414  005012  016412  DH.2:         .ASCII <14><15><12><12><12><35><37><177><177><177>/BLIND READ OF ROM/
       015057     015  006412  077577                .ASCIZ <15><12><15><177><177>/NOTE: NO CHECKING/
       015106  005015  040412  042104  DH.2B:        .ASCII  <15><12><12>/ADDRESS ADD+00  ADD+02  ADD+04/
       015147     040  040440  042104                .ASCIZ /  ADD+06  ADD+10  ADD+12  ADD+14  ADD+16/
       015220  005015  042412  052130  DH.2A:        .ASCII  <15><12><12>/EXTENDED 128. WORD ROM DUMP./
       015257     015  041412  047117                .ASCII  <15><12>/CONTENTS DUMPED IS PLACED IN THE SOFTWARE/
       015332  005015  040515  027120                .ASCII  <15><12>/MAP. DATA SHOULD BE VISUALLY INSPECTED!/
               015404                        .EVEN
```

```
                                        ;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.
                                        ;

015404  012767  015404  002260  PRG3:   MOV     #PRG3,PRG.NO    ;SET FOR POWER FAIL
015412  012701  176042                  MOV     TABLE,R1        ;DEFAULT STARTING ADDRESS TO MAP
015416  010167  000764                  MOV     R1,ADDRESS      ;SAVE THE SOFTWARE ADDRESS
015422  104400  017451          XHOLD:  TYPE    ,MASTER         ;TYPE AN "*"
015426  104412                          RDLIN
015430  012602                          MOV     (SP)+,R2
015432  011202                          MOV     (R2),R2
015434  022702  000114                  CMP     #114,R2         ;WAS AN "L" (LIST) HIT?
015440  001464                          BEQ     SRV.L

015442  022702  000104          1$:     CMP     #104,R2         ;WAS A  "D" (DATA) HIT?
015446  001413                          BEQ     SRV.D
015450  022702  000122                  CMP     #122,R2         ;WAS AN "R" (RUN) HIT?
015454  001002                          BNE     10$
015456  000167  000342                  JMP     SRV.R
015462  022702  000101          10$:    CMP     #101,R2         ;WAS AN "A" (ADDRESS) HIT?
015466  001444                          BEQ     SRV.A
015470  104400  017542                  TYPE    ,M.QM           ;TYPE A "?"
015474  000752                          BR      XHOLD           ;NEITHER A "L","P","D","R","A",OR CR WAS HIT.

015476  016767  000704  163522  SRV.D:  MOV     ADDRESS,TEMP4   ;RESET ADDRESS POINTER.
015504  104400  017556                  TYPE    ,MCRLF
015510  016746  163512                  MOV     TEMP4,-(SP)
015514  016701  163506                  MOV     TEMP4,R1
015520  104402                          TYPOC

015522  104400  017547                  TYPE    ,MSPACE

015526  104414                          RDOCT
015530  012611                          MOV     (SP)+,(R1)      ;STORE DATA

015532  005721                          TST     (R1)+           ;UPDATE THE SOFTWARE ADDRESS
015534  026701  175726                  CMP     EXTEND,R1       ;IS THE LIMIT EXCEEDED
015540  103413                          BLO     7$              ;INPUT LIMIT EXCEEDED!! ERROR.
015542  010167  163460                  MOV     R1,TEMP4        ;SAVE THE ADDRESS.
015546  104400  017556                  TYPE    ,MCRLF
015552  016746  163450                  MOV     TEMP4,-(SP)
015556  104402                          TYPOC
```

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

```
                                                BR      1$              ;TYPE THE ADDRESSS
                                        5$:
                                                BIT     #BIT0,SWR       ;EXTENDED SOFTWARE DUMP?
                                                BEQ     6$              ;BR IF NO DUMP
                                                TST     -(R3)           ;PUSH POINTER
                                                CMP     EXTEND,R3
                                                BEQ     6$              ;BR IF ALL DONE
                                                TYPE    ,MSG5           ;TYPE EXTENDED MAP:
                                                TYPE    ,DH.2B
                                                MOV     EXTMAP,R0       ;SET POINTER
                                                MOV     EXTEND,DEAD     ;SET DEAD END POINTER
                                                BR      1$              ;DO IT AGAIN SAM.
                                        6$:     JMP     XHOLD
                                        DEAD:   0

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

                                        SRV.R:
                                        RUN3.   MOV     #NO.TRAP,@#4    ;GET READY FOR UNEXPECTED TRAP
                                                MOV     #500,ICOUNT     ;DO TEST 500(8) TIMES
                                        RUN.3:  MOV     #173000,R0      ;SET BEGGINING ADDRESS
                                                MOV     #2$,$LPADR      ;IF SW14=1; GOTO 2$ WHEN I HIT "SCOPE"
                                                MOV     TABLE,R4        ;SET SOFTWARE RESUTS
                                                MOV     LASTA,LAST      ;SET LAST ADDRESS
                                        1$:     MOV     #5.,R3          ;DO EACH ADDRESS 5 TIMES.
                                        2$:     CMP     #173024,R0      ;DON'T DO THE VECTOR ADD.
                                                BNE     30$             ;BR IF NOT THE VECTOR ADD.
                                                CMP     (R0)+,(R4)+     ;UPDATE TO NEXT ADDRESS
                                        30$:    CMP     #173224,R0      ;IS THIS THE SECOND TRAP VECTOR??
                                                BNE     10$             ;BR IF NOT VECTOR
                                                CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS !!
                                        10$:    MOV     R4,$GDDAT
                                                MOV     R0,$BDDAT
                                                MOV     (R0),TEMP4      ;READ THE ADDRESS
                                                MOV     (R4),TEMP3      ;READ THE SOFTWARE ADDRESS
                                                CMP     TEMP3,TEMP4
                                                BEQ     11$             ;BRANCH IF OK
                                                ERROR   1               ;INCORRECT COMPARISON.
                                        11$:    BIT     #BIT11,SWR      ;QUICK PASS.
                                                BNE     12$             ;BR IF YES
                                                DEC     R3              ;HAS THAT ADD BEEN READ 10 TIMES?
                                                BNE     2$              ;BR IF NOT 10 TIMES
                                        12$:    CMP     LAST,R0         ;WAS LAST ADDRESS CHECKED?
                                                BEQ     15$             ;BR IF YES
                                                SCOPE                   ;LOCK ON THIS ADDRESS?
                                                CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS.
                                                BR      1$              ;CONTINUE THE TEST.
                                        15$:    BIT     #BIT0,SWR       ;EXTENDED WORD TO BE CHECKED?
                                                BEQ     3$              ;BR IF NO CHECKING.
                                                CMP     #173776,LAST    ;IS ALL THE TEST DONE?
                                                BEQ     3$              ;BR IF YES.
```

F14


```
  6252  016212  012767  173776  176066          MOV     #173776,LAST    ;SET LAST ADDRESS.
  6253  016220  016704  175240                  MOV     EXTMAP,R4       ;SET EXTENDED MAP.
  6254  016224  005720                          TST     (R0)+           ;POP POINTER
  6255  016226  000716                          BR      1$              ;GO DO THE TEST.
```

```
                                                ;TEST THAT WRITING ROM RESULTS IN A TIME OUT
                                                ;TRAP.
                                                
016230  012700  173000          3$:     MOV     #173000,R0      ;SET BASE ADDRESS
016234  012767  016254  162644          MOV     #5$,$LPADR      ;IF SW14=1; GOTO 5$ AT SCOPE
016242  012737  016310  000004          MOV     #6$,@#4         ;TIME OUT TRAP; GOTO 6$
016250  012703  000012          4$:     MOV     #10.,R3         ;DO EACH ADD 10 TIMES
016254  022700  173024          5$:     CMP     #173024,R0      ;IS THIS AT THE TRAP VECTOR
016260  001001                          BNE     20$             ;BR IF NO
016262  005720                          TST     (R0)+           ;UPDATE POINTER
016264  022700  173224          20$:    CMP     #173224,R0      ;IS THIS AT THE SECOND TRAP VECTOR
016270  001001                          BNE     21$             ;BR IF NO
016272  005720                          TST     (R0)+           ;UPDATE THE POINTER
016274  012710  177777          21$:    MOV     #-1,(R0)        ;WRITE ROM WITH A -1
016300  000240                          NOP                     ;WAIT ONE INSTR. TIME
016302  010067  162720                  MOV     R0,TEMP4
016306  104002                          ERROR   2               ;WRITING ROM DIDN'T TIME OUT.
016310  012706  001100          6$:     MOV     #STACK,SP       ;RESTORE STACK
016314  032767  004000  161246          BIT     #BIT11,SWR      ;QUICK PASS?
016322  001002                          BNE     22$             ;BR IF YES
016324  005303                          DEC     R3              ;DO EACH ADD 10 TIMES
016326  001352                          BNE     5$              ;NOT DONE WITH THIS ONE YET.
016330  032767  000001  161232  22$:    BIT     #BIT0,SWR       ;IS THE EXTENDED 128. WORDS TO BE TESTED??
016336  001404                          BEQ     23$             ;BR IF NO
016340  022700  173776                  CMP     #173776,R0      ;IS THE EXTENDED LIMIT BEEN TESTED?
016344  001407                          BEQ     7$              ;IF YES; GOTO 7$
016346  000403                          BR      24$             ;IF NO; KEEP GOING.
016350  026700  175116          23$:    CMP     LASTA,R0        ;ALL DONE??
016354  001403                          BEQ     7$              ;IF YES; GOTO 7$
016356  000004                  24$:    SCOPE                   ;GO CHECK SW14; (FREEZE !!)
016360  005720                          TST     (R0)+           ;UPDATE TO NEXT ADDRESS
016362  000732                          BR      4$              ;GO DO IT AGAIN
016364  005367  162630          7$:     DEC     ICOUNT          ;CHECK ITERATION COUNT
016370  001004                          BNE     8$              ;MORE TO GO
016372  004767  001336                  JSR     PC,EOP          ;GO TO END OF PASS ROUTINE
016376  000167  177422                  JMP     RUN3            ;GO DO TEST AGAIN
016402  000167  177432          8$:     JMP     RUN.3           ;
                                
016406  000000          ADDRESS:        0
```

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

016410  012767  016410  001254  PRG4:   MOV     #PRG4,PRG.NO    ;SET FOR POWER FAIL
016416  005067  162574                  CLR     LSTERR          ;PREPARE ERROR CONDITIONS
016422  012706  001100                  MOV     #STACK,SP       ;SET THE STACK POINTER
016426  012767  020000  162564          MOV     #20000,ICOUNT   ;SET ITERATION COUNT TO 20000(8)
016434  005767  000554                  TST     FLAG4           ;HAVE I BEEN HERE BEFOR??
016440  001106                          BNE     TAG.A           ;BR IF NOT FIRST TIME HERE.
016442  005167  000546                  COM     FLAG4           ;SET THE FLAG
016446  012705  000002                  MOV     #2,R5           ;SET R5 FOR SWITCH 1
016452  012704  017204                  MOV     #LOC1,R4        ;SET STORAGE LOCATION
016456  012737  017674  000004          MOV     #NO.TRAP,@#4    ;SET FOR TIME OUT TRAP
016464  012767  000001  162532          MOV     #1,TEMP3        ;SET FOR MESSAGE ON CHANNEL NO.
016472  104400  017216          1$:     TYPE    ,MCHAN          ;TYPE MESSAGE ABOUT CHANNEL
016476  104400  017547                  TYPE    ,MSPACE

016502  016746  162516                  MOV     TEMP3,-(SP)
016506  104402                          TYPOC
016510  104400  017547                  TYPE    ,MSPACE


016514  104400  017232                  TYPE    ,MACTV          ;TYPE REST OF MESSAGE
016520  104400  017245          2$:     TYPE    ,MADD1          ;TYPE ADDRESS MESSAGE
016524  012700  173024                  MOV     #173024,R0
016530  005037  173024                  CLR     @#173024
016534  010537  173024                  MOV     R5,@#173024     ;WRITE ROM WITH SWITCH
016540  000240                          NOP                     ;WAIT ONE INSTR. TIME
016542  012706  001100          3$:     MOV     #STACK,SP       ;SET THE STACK POINTER
016546  012700  173024                  MOV     #173024,R0      ;SET FOR ERROR MESSAGE
016552  012737  017674  000004          MOV     #NO.TRAP,@#4    ;SET FOR NO MORE TRAPS
016560  013767  173024  162440          MOV     @#173024,TEMP4  ;READ THE ADDRESS

016566  104400  017547                  TYPE    ,MSPACE
016572  016746  162430                  MOV     TEMP4,-(SP)
016576  104402                          TYPOC
016600  013724  173024                  MOV     @#173024,(R4)+  ;STORE THE INFORMATION FOUND
016604  104400  017301                  TYPE    ,MADD2          ;TYPE THE SECOND ADDRESS MSG
016610  012700  173224                  MOV     #173224,R0      ;SET FOR ERROR CONDITION.
016614  013767  173224  162404          MOV     @#173224,TEMP4  ;STORE ROM DATA
016622  104400  017547                  TYPE    ,MSPACE

016626  016746  162374                  MOV     TEMP4,-(SP)
016632  104402                          TYPOC
```

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

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

```
 6419  017216  005015  041412  040510  MCHAN:  .ASCIZ <15><12><12>/CHANNEL /
       017232  041501  044524  040524  MACTV:  .ASCIZ/ACTIVATED./
       017245     015  040412  042104  MADD1:  .ASCIZ  <15><12>/ADDRESS 773024 CONTAINS: /
       017301     015  040412  042104  MADD2:  .ASCIZ  <15><12>/ADDRESS 773224 CONTAINS: /
                017336                  .EVEN
 6420
 6421  017336  005015  051120  043517  MSG3:   .ASCIZ <15><12>/PROGRAM NO. (1,2,3,4)  /
       017370  006414  016412  077437  MSG4:   .ASCIZ <14><15><12><35><37><177><177><177>/SOFTWARE MAP:/
       017416  005015  020012  054105  MSG5:   .ASCIZ  <15><12><12>/ EXTENDED SOFTWARE MAP:/
       017451     015  025012     000  MASTER: .ASCIZ <15><12>/*/
       017455     007  006407  042412  M.END:  .ASCIZ <7><7><15><12>/END PASS BM873-Y/
       017502                          MFAIL:
       017502  005015  053520  020122          .ASCII  <15><12>/PWR UP AFTER/
       017520  005015  042522  046101          .ASCIZ  <15><12>/REAL PWR FAIL/
       017540  000044                  M.DOL:  .ASCIZ /$/
       017542  005015  037477     000  M.QM:   .ASCIZ <15><12>/??/
       017547     040  000040          MSPACE: .ASCIZ  /  /
       017552  020040  000040          SPACE3: .ASCIZ  /  /
       017556  005015     000          MCRLF:  .ASCIZ <15><12>
       017561     012     000          MLF:    .ASCIZ <12>
                017564                  .EVEN
 6422
```

```
  6423  017564  005067  161426          .PFAIL: CLR     LSTERR
  6424  017570  013746  000004                  MOV     @#4,-(SP)
  6425  017574  012737  017624  000004          MOV     #1$,@#4
  6426  017602  005737  173000                  TST     @#173000        ;IS THIS PF REAL?
  6427  017606  000240                          NOP     ;TRAP IS CAUSED BY LOADER
  6428  017610  012737  017634  000024          MOV     #PWR.UP,@#24    ;ITS REAL. PREPARE FOR PWR UP
  6429  017616  012637  000004                  MOV     (SP)+,@#4
  6430  017622  000000                          HALT
  6431  017624  005726                  1$:     TST     (SP)+   ;POP THE STACK.
  6432  017626  012637  000004                  MOV     (SP)+,@#4
  6433  017632  000000                          HALT    ;HARDWARE ERROR.  BOOT DIDN'T FORCE
  6434                          ;HIGH ADDR LINES AND LOAD BUTTON WAS ACTIVATED
  6435  017634  012737  017564  000024  PWR.UP: MOV     #.PFAIL,@#24
  6436  017642  012706  001100                  MOV     #STACK,SP
  6437  017646  005000                          CLR     R0              ;SET DELAY
  6438  017650  062700  000001          1$:     ADD     #1,R0           ;WAIT FOR TTY
  6439  017654  001375                          BNE     1$
  6440  017656  104400  017502                  TYPE    ,MFAIL          ;TYPE FAILED.
  6441  017662  005067  160110                  CLR     PS              ;SET STATUS TO ZERO
  6442  017666  000177  000000                  JMP     @PRG.NO
  6443  017672  000000          PRG.NO: 0
  6444  017674                  NO.TRAP:
  6445  017674  011667  000032                  MOV     (SP),XSTORE
  6446  017700  032716  100000                  BIT     #BIT15,(SP)
  6447  017704  001410                          BEQ     1$
  6448  017706  011600                          MOV     (SP),R0
  6449  017710  104004                          ERROR   4
  6450  017712  012706  001100                  MOV     #STACK,SP
  6451  017716  005067  160054                  CLR     PS
  6452  017722  000177  177744                  JMP     @PRG.NO
  6453  017726  104003                  1$:     ERROR   3
  6454  017730  000002                          RTI
  6455  017732  000000          XSTORE: 0
  6456
  6457  017734  005067  161256          EOP:    CLR     LSTERR
  6458  017740  104400  017455                  TYPE    ,M.END
  6459  017744  104400  020014                  TYPE    ,VERSON
  6460  017750  013701  000042                  MOV     @#42,R1
  6461  017754  001416                          BEQ     X1
  6462  017756  022767  013716  177706          CMP     #PRG1,PRG.NO
  6463  017764  001002                          BNE     .+6
  6464  017766  000167  176416                  JMP     PRG4
  6465  017772  013701  000042                  MOV     @#42.R1
  6466  017776  001405                          BEQ     X1
  6467  020000  000005                          RESET
  6468  020002                  $ENDAD:
  6469  020002  004711          LOGIC:  JSR     PC,(R1)
  6470  020004  000240                  NOP
  6471  020006  000240                  NOP
  6472  020010  000240                  NOP
  6473  020012  000207          X1:     RTS     PC
  6474  020014  000101          VERSON: 101                     ;SEVEN BIT ASCII FOR DEFAULT "A"
```

```
  6475  020016  005015  041520  020072  MERRPC: .ASCIZ  <15><12>/PC: /
  6476  020024     000
  6477          020026                  .EVEN
  6478                                  .MCALL .SEOP, .STYPE, .STYPOCT, .SPOWER, .SREAD
  6479                                  ;;*****************************************************************
  6480
  6481                                  .SBTTL  TYPE ROUTINE
  6482
  6483                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
  6484                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
  6485                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
  6486                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
  6487                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
  6488                                  ;*
  6489                                  ;*CALL:
  6490                                  ;*1) USING A TRAP INSTRUCTION
  6491                                  ;*      TYPE    ,MESADR                 ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
  6492                                  ;*OR
  6493                                  ;*      TYPE
  6494                                  ;*      MESADR
  6495                                  ;*
  6496                                  ;*2) USING A JSR INSTRUCTION
  6497                                  ;*      MOV     PS,-(SP)                ;PUSH PROCESSOR STATUS WORD ON THE STACK
  6498                                  ;*      JSR     PC,$TYPE                ;CALL TYPE ROUTINE
  6499                                  ;*      MESADDR                         ;FIRST ADRESS OF MESSAGE
  6500
  6501  020026  105767  161117  $TYPE:  TSTB    $TPFLG                  ;IS THERE A TERMINAL?
  6502  020032  100002                  BPL     1$                      ;BR IF YES
  6503  020034  000000                  HALT                            ;HALT HERE IF NO TERMINAL
  6504  020036  000407                  BR      3$                      ;LEAVE
  6505  020040  010046          1$:     MOV     R0,-(SP)                ;SAVE R0
  6506  020042  017600  000002          MOV     @2(SP),R0               ;GET ADDRESS OF ASCIZ STRING
  6507  020046  112046          2$:     MOVB    (R0)+,-(SP)             ;PUSH CHARACTER TO BE TYPED ONTO STACK
  6508  020050  001005                  BNE     4$                      ;BR IF IT ISN'T THE TERMINATOR
  6509  020052  005726                  TST     (SP)+                   ;IF TERMINATOR POP IT OFF THE STACK
  6510  020054  012600                  MOV     (SP)+,R0                ;RESTORE R0
  6511  020056  062716  000002          ADD     #2,(SP)                 ;ADJUST RETURN PC
  6512  020062  000002                  RTI                             ;RETURN
  6513  020064  004767  000026          JSR     PC,7$                   ;GO TYPE THIS CHARACTER
  6514  020070  126726  161054          CMPB    $FILLC,(SP)+            ;IS IT TIME FOR FILLER CHARS.?
  6515  020074  001364                  BNE     2$                      ;IF NO GO GET NEXT CHAR.
  6516  020076  016746  161044          MOV     $NULL,-(SP)             ;GET # OF FILLER CHARS. NEEDED
  6517                                                                  ;AND THE NULL CHAR.
  6518  020102  105366  000001          DECB    1(SP)                   ;DOES A NULL NEED TO BE TYPED?
  6519  020106  002770                  BLT     5$                      ;BR IF NO--GO POP THE NULL OFF OF STACK
  6520  020110  004767  000002          JSR     PC,7$                   ;GO TYPE A NULL
  6521  020114  000772                  BR      6$                      ;LOOP
  6522  020116  105777  161020  7$:     TSTB    @$TPS                   ;WAIT UNTIL PRINTER IS READY
  6523  020122  100375                  BPL     7$
  6524  020124  116677  000002  161012  MOVB    2(SP),@$TPB             ;LOAD CHAR TO BE TYPED INTO DATA REG.
  6525  020132  000207                  RTS     PC
  6526                                  ;;*****************************************************************
  6527
  6528                                  .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

020134  011646          $RDCHR: MOV     (SP),-(SP)              ;PUSH DOWN THE PC
020136  016666  000004  000002          MOV     4(SP),2(SP)             ;SAVE THE PS
020144  105777  160766          1$:     TSTB    @$TKS                   ;WAIT FOR
020150  100375                          BPL     1$                      ;A CHARACTER
020152  117766  160762  000004          MOVB    @$TKB,4(SP)             ;READ THE TTY
020160  042766  177600  000004          BIC     #^C<177>,4(SP)          ;GET RID OF JUNK IF ANY
020166  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

020170  010346          $RDLIN: MOV     R3,-(SP)                ;SAVE R3
020172  012703  020276  1$:     MOV     #$TTYIN,R3              ;GET ADDRESS
020176  022703  020306  2$:     CMP     #$TTYIN+8.,R3           ;BUFFER FULL?
020202  101405                          BLOS    4$                      ;BR IF YES
020204  104410                          RDCHR                           ;GO READ ONE CHARACTER FROM THE TTY
020206  112613                          MOVB    (SP)+,(R3)              ;GET CHARACTER
020210  122713  000177                  CMPB    #177,(R3)               ;IS IT A RUBOUT
020214  001003                          BNE     3$                      ;SKIP IF NOT
020216  104400  001152          4$:     TYPE    ,$QUES                  ;TYPE A '?'
020222  000763                          BR      1$                      ;CLEAR THE BUFFER AND LOOP
020224  111367  000044          3$:     MOVB    (R3),8$                 ;ECHO THE CHARACTER
020230  104400  020274                  TYPE    ,8$
020234  122723  000015                  CMPB    #15,(R3)+               ;CHECK FOR RETURN
020240  001356                          BNE     2$                      ;LOOP IF NOT RETURN
020242  105063  177777                  CLRB    -1(R3)                  ;CLEAR RETURN (THE 15)
020246  104400  001154                  TYPE    ,$LF                    ;TYPE A LINE FEED
020252  012603                          MOV     (SP)+,R3                ;RESTORE R3
020254  011646                          MOV     (SP),-(SP)              ;ADJUST THE STACK AND PUT ADDRESS OF THE
020256  016666  000004  000002          MOV     4(SP),2(SP)             ;       FIRST ASCII CHARACTER ON IT
020264  012766  020276  000004          MOV     #$TTYIN,4(SP)
020272  000002                          RTI                             ;RETURN
020274     000          8$:     .BYTE   0                       ;STORAGE FOR ASCII CHAR. TO TYPE
020275     000                  .BYTE   0                       ;TERMINATOR
020276  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
```

```
6583                                    ;*      .BYTE   M                       ;M=1 OR 0
6584                                    ;*                                      ;1=TYPE LEADING ZEROS
6585                                    ;*                                      ;0=SUPPRESS LEADING ZEROS
6586                                    ;*
6587                                    ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
6588                                    ;*$TYPOS OR $TYPOC
6589                                    ;*CALL:
6590                                    ;*      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
6591                                    ;*      TYPON                           ;CALL FOR TYPEOUT
6592                                    ;*
6593                                    ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
6594                                    ;*CALL:
6595                                    ;*      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
6596                                    ;*      TYPOC                           ;CALL FOR TYPEOUT
6597
6598  020306  017646  000000            $TYPOS: MOV     @(SP),-(SP)             ;PICKUP THE MODE
6599  020312  116667  000001  000211            MOVB    1(SP),$OFILL            ;LOAD ZERO FILL SWITCH
6600  020320  112667  000207                    MOVB    (SP)+,$OMODE+1          ;NUMBER OF DIGITS TO TYPE
6601  020324  062716  000002                    ADD     #2,(SP)                 ;ADJUST RETURN ADDRESS
6602  020330  000406                            BR      $TYPON
6603  020332  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL               ;SET THE ZERO FILL SWITCH
6604  020340  112767  000006  000155          MOVB    #6,$OMODE+1             ;SET FOR SIX(6) DIGITS
6605  020346  112767  000005  000154  $TYPON: MOVB    #5,$OCNT                ;SET THE ITERATION COUNT
6606  020354  010346                            MOV     R3,-(SP)                ;SAVE R3
6607  020356  010446                            MOV     R4,-(SP)                ;SAVE R4
6608  020360  010546                            MOV     R5,-(SP)                ;SAVE R5
6609  020362  116704  000145                    MOVB    $OMODE+1,R4             ;GET THE NUMBER OF DIGITS TO TYPE
6610  020366  005404                            NEG     R4
6611  020370  062704  000006                    ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
6612  020374  110467  000132                    MOVB    R4,$OMODE               ;SAVE IT FOR USE
6613  020400  116704  000125                    MOVB    $OFILL,R4               ;GET THE ZERO FILL SWITCH
6614  020404  016605  000012                    MOV     12(SP),R5               ;PICKUP THE INPUT NUMBER
6615  020410  005003                            CLR     R3                      ;CLEAR THE OUTPUT WORD
6616  020412  006105                    1$:     ROL     R5                      ;ROTATE MSB INTO "C"
6617  020414  000404                            BR      3$                      ;GO DO MSB
6618  020416  006105                    2$:     ROL     R5                      ;FORM THIS DIGIT
6619  020420  006105                            ROL     R5
6620  020422  006105                            ROL     R5
6621  020424  010503                            MOV     R5,R3
6622  020426  006103                    3$:     ROL     R3                      ;GET LSB OF THIS DIGIT
6623  020430  105367  000076                    DECB    $OMODE                  ;TYPE THIS DIGIT?
6624  020434  100016                            BPL     7$                      ;BR IF NO
6625  020436  042703  177770                    BIC     #177770,R3              ;GET RID OF JUNK
6626  020442  001002                            BNE     4$                      ;TEST FOR 0
6627  020444  005704                            TST     R4                      ;SUPPRESS THIS 0?
6628  020446  001403                            BEQ     5$                      ;BR IF YES
6629  020450  005204                    4$:     INC     R4                      ;DON'T SUPPRESS ANYMORE 0'S
6630  020452  052703  000060                    BIS     #'0,R3                  ;MAKE THIS DIGIT ASCII
6631  020456  052703  000040            5$:     BIS     #' ,R3                  ;MAKE ASCII IF NOT ALREADY
6632  020462  110367  000040                    MOVB    R3,8$                   ;SAVE FOR TYPING
6633  020466  104400  020526                    TYPE    ,8$                     ;GO TYPE THIS DIGIT
6634  020472  105367  000032            7$:     DECB    $OCNT                   ;COUNT BY 1
6635  020476  003347                            BGT     2$                      ;BR IF MORE TO DO
6636  020500  002402                            BLT     6$                      ;BR IF DONE
```

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

```
 6691  020644  005267  160242                 INC     $ERTTL                  ;INC THE ERROR COUNT
 6692  020650  011667  160242                 MOV     (SP),$ERRPC             ;GET ADDRESS OF ERROR INSTRUCTION
 6693  020654  162767  000002  160234         SUB     #2,$ERRPC
 6694  020662  117767  160230  160224         MOVB    @$ERRPC,$ITEMB          ;STRIP AND SAVE THE ERROR ITEM CODE
 6695  020670  032737  020000  177570         BIT     #SW13,@#SWR             ;SKIP TYPEOUT IF SET
 6696  020676  001004                         BNE     2$                      ;SKIP TYPEOUTS
 6697  020700  004737  020722                 JSR     PC,@#$ERRTYP            ;GO TO USER ERROR ROUTINE
 6698  020704  104400  001153                 TYPE    ,$CRLF
 6699  020710  005737  177570         2$:     TST     @#SWR                   ;HALT ON ERROR
 6700  020714  100001                         BPL     3$                      ;SKIP IF CONTINUE
 6701  020716  000000                         HALT                            ;HALT ON ERROR!
 6702  020720  000002         3$:             RTI                             ;RETURN
 6703                                 ;;*******************************************************************
 6704
 6705                                 .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
 6706
 6707                                 ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 6708                                 ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
 6709                                 ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
 6710
 6711  020722                         $ERRTYP:
 6712  020722  104400  001153                 TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
 6713  020726  010046                         MOV     R0,-(SP)                ;SAVE R0
 6714  020730  005000                         CLR     R0                      ;PICKUP THE ITEM INDEX
 6715  020732  153700  001114                 BISB    @#$ITEMB,R0
 6716  020736  001004                         BNE     1$                      ;IF ITEM NUMBER IS ZERO, JUST
 6717                                                                         ;TYPE THE PC OF THE ERROR
 6718  020740  016746  160152                 MOV     $ERRPC,-(SP)            ;SAVE $ERRPC FOR TYPEOUT
 6719                                                                         ;ERROR ADDRESS
 6720  020744  104402                         TYPOC                           ;GO TYPE--OCTAL ASCII(ALL DIGITS)
 6721  020746  000426                         BR      6$                      ;GET OUT
 6722  020750  005300         1$:             DEC     R0                      ;ADJUST THE INDEX SO THAT IT WILL
 6723  020752  006300                         ASL     R0                      ;       WORK FOR THE ERROR TABLE
 6724  020754  006300                         ASL     R0
 6725  020756  006300                         ASL     R0
 6726  020760  062700  001156                 ADD     #$ERRTB,R0              ;FORM TABLE POINTER
 6727  020764  012067  000004                 MOV     (R0)+,2$                ;PICKUP "ERROR MESSAGE" POINTER
 6728  020770  001404                         BEQ     3$                      ;SKIP TYPEOUT IF NO POINTER
 6729  020772  104400                         TYPE                            ;TYPE THE "ERROR MESSAGE"
 6730  020774  000000         2$:             .WORD   0                       ;"ERROR MESSAGE" POINTER GOES HERE
 6731  020776  104400  001153                 TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
 6732  021002  012067  000004         3$:     MOV     (R0)+,4$                ;PICKUP "DATA HEADER" POINTER
 6733  021006  001404                         BEQ     5$                      ;SKIP TYPEOUT IF 0
 6734  021010  104400                         TYPE                            ;TYPE THE "DATA HEADER"
 6735  021012  000000         4$:             .WORD   0                       ;"DATA HEADER" POINTER GOES HERE
 6736  021014  104400  001153                 TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
 6737  021020  011000         5$:             MOV     (R0),R0                 ;PICKUP "DATA TABLE" POINTER
 6738  021022  001004                         BNE     7$                      ;GO TYPE THE DATA
 6739  021024  012600         6$:             MOV     (SP)+,R0                ;RESTORE R0
 6740  021026  104400  001153                 TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
 6741  021032  000207                         RTS     PC                      ;RETURN
 6742  021034                         7$:
 6743  021034  013046                         MOV     @(R0)+,-(SP)            ;SAVE @(R0)+ FOR TYPEOUT
 6744  021036  104402                         TYPOC                           ;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
                                        TST     (R0)                    ;IS THERE ANOTHER NUMBER?
                                        BEQ     6$                      ;BR IF NO
                                        TYPE    ,8$                     ;TYPE TWO(2) SPACES
                                        BR      7$                      ;LOOP
                                8$:     .ASCIZ  /  /                    ;TWO(2) SPACES
                                        .EVEN
                                ;;**********************************************************************

                                .SBTTL  READ AN OCTAL NUMBER FROM THE TTY

                                ;*CALL:
                                ;*      RDOCT                           ;READ AN OCTAL NUMBER
                                ;*      RETURN HERE                     ;LOW ORDER BITS ARE ON TOP OF THE STACK
                                ;*                                      ;HIGH ORDER BITS ARE IN $HIOCT

                                $RDOCT: MOV     (SP),-(SP)              ;PROVIDE SPACE FOR THE
                                        MOV     4(SP),2(SP)             ;INPUT NUMBER
                                        MOV     R0,-(SP)                ;PUSH R0 ON STACK
                                        MOV     R1,-(SP)                ;PUSH R1 ON STACK
                                        MOV     R2,-(SP)                ;PUSH R2 ON STACK
                                1$:     RDLIN                           ;READ AN ASCIZ LINE
                                        MOV     (SP)+,R0                ;GET ADDRESS OF 1ST CHARACTER
                                        CLR     R1                      ;CLEAR DATA WORD
                                        CLR     R2
                                2$:     MOVB    (R0)+,-(SP)             ;PICKUP THIS CHARACTER
                                        BEQ     3$                      ;IF ZERO GET OUT
                                        ASL     R1                      ;*2
                                        ROL     R2
                                        ASL     R1                      ;*4
                                        ROL     R2
                                        ASL     R1                      ;*8
                                        ROL     R2
                                        BIC     #^C7,(SP)               ;STRIP THE ASCII JUNK
                                        ADD     (SP)+,R1                ;ADD IN THIS DIGIT
                                        BR      2$                      ;LOOP
                                3$:     TST     (SP)+                   ;CLEAN TERMINATOR FROM STACK
                                        MOV     R1,12(SP)               ;SAVE THE RESULT
                                        MOV     R2,$HIOCT
                                        MOV     (SP)+,R2                ;POP STACK INTO R2
                                        MOV     (SP)+,R1                ;POP STACK INTO R1
                                        MOV     (SP)+,R0                ;POP STACK INTO R0
                                        RTI                             ;RETURN
                                $HIOCT: .WORD   0                       ;HIGH ORDER BITS GO HERE
                                ;;**********************************************************************

                                .SBTTL  TRAP DECODER

                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                ;*GO TO THAT ROUTINE.

                                $TRAP:  MOV     R0,-(SP)                ;SAVE R0
                                        MOV     2(SP),R0                ;GET TRAP ADDRESS
```

```
                                        TST     -(R0)                   ;BACKUP BY 2
                                        MOVB    (R0),R0                 ;GET RIGHT BYTE OF TRAP
                                        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
                                ;       -------
                                $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
021216  005015  047522  020115  EM1:    .ASCIZ  <15><12>/ROM READ DATA COMPARISON ERROR./
021260  005015  051127  052111  EM2:    .ASCIZ  <15><12>/WRITING ROM FAILED TO TRAP./
021316  005015  047125  054105  EM3:    .ASCIZ  <15><12>/UNEXP TRAP WHILE READING ROM./
021356  005015  040506  040524  EM4:    .ASCIZ  <15><12>/FATAL TRAP.  ROM PC ON STACK./
021416  005015  041520  020040  DH1:    .ASCII  <15><12>/PC      SOFT    ROM/
021443     015  040412  042104          .ASCIZ  <15><12>/ADDRESS ADDRESS ADDRESS EXPECTED FOUND /
021515     015  050012  004503  DH2:    .ASCII  <15><12>/PC     ROM/
021525     015  040412  042104          .ASCIZ  <15><12>/ADDRESS ADDRESS/
021547     015  050012  020103  DH3:    .ASCII  <15><12>/PC OF   PROGRAM /
021571     015  052012  040522          .ASCIZ  <15><12>/TRAP    ADDRESS/
021614                                  .EVEN
021614  001116  001124  001126  DT1:    .WORD   $ERRPC,$GDDAT,$BDDAT,TEMP3,TEMP4,0
021622  001224  001226  000000
021630  001116  001226  000000  DT2:    .WORD   $ERRPC,TEMP4,0
021636  001116  017732  000000  DT3:    .WORD   $ERRPC,XSTORE,0
021704                                  .=.+40
021704                          CORMAX:
000001                                  .END
```

```
ADDRES  016406          6107*  6125   6144*  6155*  6294#
ALLEND  013462          5738*  5746*  5764*  5772*  5782*  5790*  5800*  5810*  5827*  5833*  5839*  5845*  5851*
                        5857*  5863*  5869*  5875*  5902#  6165
BIT0   = 000001          449#  5953   5984   6052   6200   6248   6278
BIT00  = 000001          439#   449
BIT01  = 000002          438#   448
BIT02  = 000004          437#   447
BIT03  = 000010          436#   446
BIT04  = 000020          435#   445
BIT05  = 000040          434#   444
BIT06  = 000100          433#   443
BIT07  = 000200          432#   442
BIT08  = 000400          431#   441
BIT09  = 001000          430#   440
BIT1   = 000002          448#
BIT10  = 002000          429#
BIT11  = 004000          428#  5942   5979   6239   6274
BIT12  = 010000          427#
BIT13  = 020000          426#
BIT14  = 040000          425#
BIT15  = 100000          424#  6446
BIT2   = 000004          447#
BIT3   = 000010          446#
BIT4   = 000020          445#
BIT5   = 000040          444#
BIT6   = 000100          443#
BIT7   = 000200          442#
BIT8   = 000400          441#
BIT9   = 001000          440#
BM.ERR  014365          5881   6001#
BM873X  014310          5820   6001#
BPTVEC= 000014           456#
CONT    012772          5721   5817#
CORMAX  021704          6827#  6828
DEAD    016022          6165*  6192   6208*  6211#
DH.2    015024          6014   6089#
DH.2A   015220          6056   6089#
DH.2B   015106          6015   6057   6089#  6167   6206
DH1     021416           527   6820#
DH2     021515           534    547   6820#
DH3     021547           541   6820#
DISPLA= 177570           381#  6676*  6690*
DT1     021614           528   6821#
DT2     021630           535    548   6824#
DT3     021636           542   6825#
EMTVEC= 000030           459#  5705*  5706*
EM1     021216           526   6820#
EM2     021260           533   6820#
EM3     021316           540   6820#
EM4     021356           546   6820#
END.YA  001776           746#  5738   5833
END.YB  002776          1118#  5746   5839
END.YC  003776          1586#  5764   5845
END.YD  004776          2228#  5772   5851
```

DZBMD.P11 CROSS REFERENCE TABLE

```
END.YF  005776          3261#  5782   5857
END.YG  006776          3835#  5790   5863
END.YH  007776          4605#  5800   5869
END.YJ  010776          5680#  5810   5875
END.YX  011776          5692#  5904
END.Y.  011376          5690#  5827
EOP     017734          5996   6290   6411   6457#
ERRVEC= 000004           452#  6664   6665*  6667*  6670*
EXTEND  013466          5904#  6137   6203   6208
EXTMAP  013464          5903#  5958   6055   6207   6253
FLAG4   017214          5718*  6311   6313*  6418#
GNS   = ****** U         364   6813   6814   6815   6816   6817   6818   6819
ICOUNT  001220           552#  5922*  5994*  6220*  6288*  6310*  6386*
INITFG  013470          5710*  5817   5819*  5905#
IOTVEC= 000020           457#  5703*  5704*
LAST    014306          5927*  5947   5955   5957*  5999#  6010*  6011*  6046   6224*  6243   6250   6252*
LASTA   013472          5739*  5747*  5765*  5773*  5783*  5791*  5801*  5811*  5828*  5834*  5840*  5846*  5852*
                        5858*  5864*  5870*  5876*  5906#  5927   5989   6010   6224   6283
LOC1    017204          6315   6357   6367   6369   6393   6403   6414#
LOC2    017206          6415#
LOC3    017210          6416#
LOC4    017212          6417#
LOGIC   020002          6469#
LSTERR  001216           551#  5713*  6308*  6423*  6457*
MACTV   017232          6326   6419#
MADD1   017245          6327   6419#
MADD2   017301          6341   6419#
MAP.YA  001400           562#  5727   5737   5832
MAP.YB  002000           751#  5743   5745   5838
MAP.YC  003000          1120#  5751   5755   5763   5844
MAP.YD  004000          1591#  5769   5771   5850
MAP.YF  005000          2231#  5777   5779   5781   5856
MAP.YG  006000          3270#  5787   5789   5862
MAP.YH  007000          3839#  5795   5797   5799   5868
MAP.YJ  010000          4613#  5805   5807   5809   5874
MAP.YX  011400          5691#  5903
MAP.Y.  011000          5689#  5826   6020   6036
MASTER  017451          6108   6421#
MCHAN   017216          6318   6419#
MCRLF   017556          6024   6063   6126   6140   6172   6421#
MERRPC  020016          6475#
MFAIL   017502          6421#  6440
MLF     017561          6421#
MSG3    017336          5883   6421#
MSG4    017370          6166   6421#
MSG5    017416          6205   6421#
MSPACE  017547          6028   6041   6067   6075   6131   6177   6182   6190   6319   6323   6337   6344   6421#
M.DOL   017540          6421#
M.END   017455          6421#  6458
M.QM    017542          5899   6122   6146   6421#
NMATCH  013574          5815   5910#
NOROM   013474          5723   5907#
NOROMS  013504          5907   5910#
NC.TRA  017674          5923   6009   6219   6316   6334   6358   6363   6444#
```

```
PC     =%000007   393#  5996*  6290*  6411*  6469*  6473*  6513*  6520*  6525*  6697*  6741*
PIRQ   = 177772   379#
PIRQVE= 000240   463#
PRG.NO  017672  5715*  5921*  6008*  6105*  6307*  6442  6443#  6452  6462
PRG.1   013732  5923#  5998
PRG1    013716  5741  5749  5767  5775  5785  5793  5803  5813  5889  5921#  5997  6462
PRG2    014444  5892  6008#  6088
PRG3    015404  5895  6105#
PRG4    016410  5898  6307#  6354  6412  6464
PS     = 177776   376#  377  6441*  6451*
PSW    = 177776   377#
PWRVEC= 000024   458#
PWR.UP  017634  6428  6435#
RDCHR  = 104410  6554  6817#
RDLIN  = 104412  5821  5884  6109  6765  6818#
RDOCT  = 104414  6133  6154  6819#
RESTRT  012000   368  5697#  5900
RESVEC= 000010   453#
RUN.3   016040  6221#  6292
RUN3    016024  6219#  6291
R0     =%000000   384#  5924*  5929  5931  5932  5934  5936  5937  5947  5950  5959  5965*  5968
                  5970  5971  5973  5974*  5976  5986  5989  5992  6012*  6017  6018  6022  6023
                  6033  6034  6038  6039  6046  6054*  6059  6060  6062  6071  6072  6080  6081
                  6086*  6164*  6169  6170  6171  6184  6185  6186  6194  6207*  6221*  6226  6228
                  6229  6231  6233  6234  6243  6246  6254  6259*  6263  6265  6266  6268  6269*
                  6271  6280  6283  6286  6328*  6333*  6342*  6364*  6374*  6392*  6402*  6437*  6438*
                  6448*  6505  6506*  6507  6510*  6713  6714*  6715*  6722*  6723*  6724*  6725*  6726*
                  6727  6732  6737*  6739*  6743  6745  6762  6766*  6769  6785*  6797  6798*  6799
                  6800*  6801*  6802*
R1     =%000001   385#  6017*  6019  6033*  6035  6059*  6061  6071*  6073  6106*  6107  6128*  6134*
                  6136  6137  6139  6144  6365*  6371  6378  6393*  6397  6403*  6408  6460*  6465*
                  6469  6763  6767*  6771*  6773*  6775*  6778*  6781  6784*
R2     =%000002   386#  5822*  5823*  5824  5830  5836  5842  5848  5854  5860  5866  5872  5879
                  5885*  5886  6110*  6111*  6112  6115  6117  6120  6764  6768*  6772*  6774*  6776*
                  6782  6783*
R3     =%000003   387#  5886*  5887  5890  5893  5896  5928*  5944*  5967*  5981*  6013*  6022*  6038*
                  6055*  6062*  6080*  6192*  6193  6194  6202  6203  6225*  6241*  6262*  6276*  6356*
                  6360  6383*  6384  6550  6551*  6552  6555*  6556  6560  6562  6564*  6566*  6606
                  6615*  6621*  6622*  6625*  6630*  6631*  6632  6641*
R4     =%000004   388#  5926*  5931  5934  5935  5938  5950  5958*  6223*  6228  6231  6232  6235
                  6246  6253*  6315*  6340*  6357*  6365  6381  6607  6609*  6610*  6611*  6612  6613*
                  6627  6629*  6637*  6640*
R5     =%000005   389#  6314*  6330  6351*  6352  6366*  6371  6375*  6378  6394*  6397  6404*  6408
                  6608  6614*  6616*  6618*  6619*  6620*  6621  6639*
R6     =%000006   390#  392  5698*  5699*  5700
R7     =%000007   391#  393
SAVR0    001230   556#
SAVR1    001232   557#
SAVR4    001234   558#
SAVR5    001236   559#
SP     =%000006   392#  5702*  5712*  5722*  5726  5729*  5733  5735  5736  5753*  5757  5759  5760
                  5822  5885  5978*  6026*  6031*  6043*  6064*  6068*  6077*  6110  6127*  6134  6141*
                  6155  6174*  6179*  6188*  6273*  6309*  6321*  6332*  6338*  6346*  6362*  6424*  6429
                  6431  6432  6436*  6445  6446  6448  6450*  6505*  6506  6507*  6509  6510  6511*
```

```
                        6514    6516*   6518*   6524    6536*   6537*   6540*   6541*   6550*   6555    6566    6567*   6568*
                        6569*   6598*   6599    6600    6601*   6606*   6607*   6608*   6614    6639    6640    6641    6642*
                        6643*   6664*   6667    6669    6670    6674    6677*   6692    6713*   6718*   6739    6743*   6760*
                        6761*   6762*   6763*   6764*   6766    6769*   6777*   6778    6780    6781*   6783    6784    6785
                        6797*   6798
SPACE3  017552          6421#
SRV.A   015600          6121    6154#
SRV.D   015476          6116    6125#
SRV.L   015612          6113    6163#
SRV.R   016024          6119    6218#
STACK = 001100           373#   5702    5712    5978    6273    6309    6332    6362    6436    6450
START   012102          5712#   5715
STKLMT= 177774           378#
SWR   = 177570           380#    381    5942    5953    5979    5984    6052    6200    6239    6248    6274    6278    6659*
                        6695    6699
SW0   = 000001           421#
SW00  = 000001           411#    421
SW01  = 000002           410#    420
SW02  = 000004           409#    419
SW03  = 000010           408#    418
SW04  = 000020           407#    417
SW05  = 000040           406#    416
SW06  = 000100           405#    415
SW07  = 000200           404#    414
SW08  = 000400           403#    413    6655    6673
SW09  = 001000           402#    412     503    5709    6655    6671    6673    6675    6679    6685    6702    6703
SW1   = 000002           420#
SW10  = 002000           401#     503    6685    6691    6703
SW11  = 004000           400#     503    5709    6655    6673    6675    6679
SW12  = 010000           399#
SW13  = 020000           398#   6684    6695
SW14  = 040000           397#   6654    6659
SW15  = 100000           396#   6683    6699
SW2   = 000004           419#
SW3   = 000010           418#
SW4   = 000020           417#
SW5   = 000040           416#
SW6   = 000100           415#
SW7   = 000200           414#
SW8   = 000400           413#
SW9   = 001000           412#
TABLE   013460          5711*   5737*   5745*   5763*   5771*   5781*   5789*   5799*   5809*   5826*   5832*   5838*   5844*
                        5850*   5856*   5862*   5868*   5874*   5879    5901#   5926    6013    6020    6036    6106    6164
                        6223
TAG.A   016656          6312    6356#   6387
TBITVE= 000014           454#
TEMP3   001224           554#   5938*   5939    6018*   6026    6034*   6060*   6064    6072*   6170*   6174    6185*   6235*
                        6236    6317*   6321    6349*   6369*   6821
TEMP4   001226           555#   5937*   5939    5976*   6019*   6031    6035*   6043    6061*   6068    6073*   6077    6125*
                        6127    6128    6139*   6141    6169*   6179    6184*   6188    6234*   6236    6271*   6335*   6338
                        6343*   6346    6370*   6377*   6396*   6406*   6821    6824
TEMP5   001222           553#   6016*   6048*   6058*   6083*   6168*   6196*
TKVEC = 000060           461#
TPVEC = 000064           462#
```

```
TRAPVE= 000034            460*  5707*  5708*
TRTVEC= 000014            455*
TYPE  = 104400           5815  5820  5881  5883  5899  5907  6014  6015  6024  6028  6041  6056  6057
                         6063  6067  6075  6108  6122  6126  6131  6140  6146  6166  6167  6172  6177
                         6182  6190  6205  6206  6318  6319  6323  6326  6327  6337  6341  6344  6440
                         6458  6459  6558  6561  6565  6633  6698  6712  6729  6731  6734  6736  6740
                         6747  6813*
TYPOC = 104402           6027  6032  6044  6065  6069  6078  6129  6142  6175  6180  6189  6322  6339
                         6347  6720  6744  6814*
TYPON = 104406           6816*
TYPOS = 104404           6815*
VERS    014417           6001*
VERSON  020014           5740*  5748*  5766*  5774*  5784*  5792*  5802*  5812*  5878*  6388  6459  6474*
XHOLD   015422           6108*  6123  6145  6147  6156  6210
XSTORE  017732           6445*  6455*  6825
X.X.    013372           5884*
X1      020012           6461  6466  6473*
$BOADR  001122            491*
$BODAT  001126            493*  5936*  6233*  6368*  6376*  6395*  6405*  6821
$CMTAG  001100            479*  5698  5705  5709
$CM3  = 000000            503*
$CRLF   001153            504*  6574  6698  6703  6712  6731  6736  6740
$ENDAD  020002            472  6468*
$ERFLG  001103            482*  6656  6675*  6679  6688*  6703
$ERMAX  001115            488*  6679
$ERROR  020630           5705  6687*
$ERRPC  001116            489*  6692*  6693*  6694  6703  6718  6821  6824  6825
$ERRTB  001156            522*  6726
$ERRTY  020722           6697  6711*
$ERTTL  001112            486*  6691*  6703
$FILLC  001150            501*  6514  6526
$FILLS  001147            500*  6526
$GDADR  001120            490*
$GDDAT  001124            492*  5935*  6232*  6367*  6821
$HD   = 000003            355  356
$HIOCT  021156           6782*  6787*
$ICNT   001104            483*
$ITEMB  001114            487*  6694*  6703  6715
$LF     001154            505*  6565  6574  6703
$LPADR  001106            484*  5709*  5925*  5964*  6222*  6260*  6674*  6677  6679
$LPERR  001110            485*
$NULL   001146            499*  6516  6526
$OCNT   020530           6605*  6634*  6647*
$OMODE  020532           6600*  6604*  6609  6612*  6623*  6649*
$OVER   020614           6660  6671  6676*
$PASS   001100            480*
$QUES   001152            503*  6558  6574  6703
$RDCHR  020134           6536*  6817
$RDDEC= ****** U         6820
$RDLIN  020170           6550*  6818
$RDOCT  021056           6760*  6819
$RDSZ = 000010           6543*
$SAVRE= ****** U         6820
$SCOPE  020534           5703  6658*
```

DZBMD.P11 CROSS REFERENCE TABLE

```
$SETUP= 000007          5698#  5703   5705   5707   5709
$STUP  = 177777         5698#
$SVLAD  020600          6668   6673#
$SWR   = 160000          355    356#   503   5709   6654   6655   6659   6671   6673   6675   6679   6683   6684
                        6685   6691   6695   6699   6702   6703
$TKB    001140           496#  6535   6540
$TKS    001136           495#  5700   6535   6538
$TN    = 000001          355#
$TPB    001144           498#  6524#  6526
$TPFLG  001151           502#  6501   6526
$TPS    001142           497#  6522   6526
$TRAP   021160          5707   6797#
$TRP   = 000016         6804#  6814#  6815#  6816#  6817#  6818#  6819#  6820#
$TRPAD  021200          6801   6812#
$TSTNM  001102           481#  6656   6673*  6676   6679   6690   6703
$TTYIN  020276          6551   6552   6569   6573#
$TYPBN= ****** U        6817
$TYPDS= ****** U        6817
$TYPE   020026          6501#  6804   6813
$TYPOC  020332          6603#  6814
$TYPON  020346          6602   6605#  6816
$TYPOS  020306          6598#  6815
$XTSTR  020542          6662#
$OFILL  020531          6599*  6603*  6613   6648#
.      = 021704          360#   364    366#   471#   474#   477#   506    561#  5689#  5690#  5691#  5692#  5701
                        5709   5720   5909   5910#  6001#  6089#  6419#  6421#  6463   6477#  6526   6535   6573#
                        6574   6679   6703   6750#  6820#  6826#
..FAIL  017564          6423#  6435
```

```
COMMEN      1#    464#
ENDCOM      1#    464#
ERROR     374#   5941    5977    6238    6272    6373    6380    6399    6410    6449    6453
ESCAPE      1#    464#
MULT        1#    464#
NEWTST      1#    464#
POP         1#    464#   6783
PUSH        1#    464#   6762
SCOPE     375#   5949    5991    6245    6285
SETTRA   6804#   6814    6815    6816    6817    6818    6819
SETUP       1#    464#   5698
SKIP        1#    464#
SLASH       1#    464#
SPACE     464#
STARS       1#    464#    506    6479    6526    6543    6574    6650    6679    6703    6751    6788
TRMTRP   6804#
TYPBIN      1#    464#
TYPDEC      1#    464#
TYPNUM      1#    464#
TYPOCS      1#    464#
TYPOCT      1#    464#   6718    6742
TYPTXT      1#    464#
$HOW      560#   5824    5830    5836    5842    5848    5854    5860    5866    5872
$ROMTY    560#   5727    5743    5751    5769    5777    5787    5795    5805
$SCMRE    464#
$SCMTM    464#
$SESCA      1#    464#
$SNEWT      1#    464#
$SSET    6804#   6814    6815    6816    6817    6818    6819
$SSKIP      1#    464#
.EQUAT      1#    343#    369
.HEADE      1#    343#    345
.KTI1       1#
.RDLIN    344#
.SETUP      1#    343#   5697
.SWRHI      1#    343#
.SWRLO    343#
.$CATC      1#    343#    357
.$CMTA      1#    343#    464
.$D82D      1#
.$D820      1#
.$D1.       1#
.$EOP       1#   6478#
.$ERRO      1#    344#   6679
.$ERRT      1#    344#   6703
.$MULT      1#
.$POWE      1#   6478#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#    344#   6751
.$READ      1#   6478#   6526
.$SAVE      1#
.$SB2D      1#
.$SB20      1#
```

ADD	6011	6438	6511	6601	6611	6726	6778								
ASL	6723	6724	6725	6771	6773	6775									
BEQ	5940	5948	5954	5956	5985	5987	5990	6047	6053	6082	6113	6116	6121	6195	6201
	6204	6237	6244	6249	6251	6279	6281	6284	6372	6379	6389	6398	6409	6447	6461
	6466	6628	6689	6728	6733	6746	6770								
BGT	6635														
BIC	6541	6625	6777												
BIS	6630	6631													
BISB	6715														
BIT	5942	5953	5979	5984	6052	6200	6239	6248	6274	6278	6446	6695			
BLO	6138														
BLOS	6553														
BLT	6519	6636													
BMI	6660														
BNE	5701	5720	5728	5744	5752	5756	5770	5778	5780	5788	5796	5798	5806	5808	5818
	5825	5831	5837	5843	5849	5855	5861	5867	5873	5880	5888	5891	5894	5897	5930
	5933	5943	5945	5969	5972	5980	5982	5995	6021	6037	6049	6084	6118	6197	6227
	6230	6240	6242	6264	6267	6275	6277	6289	6312	6353	6385	6387	6439	6463	6508
	6515	6557	6563	6626	6696	6716	6738								
BPL	6502	6523	6539	6624	6700										
BR	5734	5758	5761	5882	5909	5951	5960	5988	5993	6050	6085	6123	6198	6209	6247
	6255	6282	6287	6504	6521	6559	6602	6617	6638	6662	6668	6671	6721	6748	6779
CLC	6350	6382													
CLR	5699	5710	5711	5713	5714	5717	5718	6086	6308	6329	6359	6391	6401	6423	6437
	6441	6451	6457	6615	6714	6767	6768								
CLRB	6564	6675													
CMP	5700	5727	5735	5743	5751	5755	5759	5769	5777	5779	5787	5795	5797	5805	5807
	5824	5830	5836	5842	5848	5854	5860	5866	5872	5887	5890	5893	5896	5929	5931
	5932	5934	5939	5947	5950	5955	5968	5971	5986	5989	6020	6036	6046	6081	6112
	6115	6117	6120	6137	6194	6203	6226	6228	6229	6231	6236	6243	6246	6250	6263
	6266	6280	6283	6352	6371	6378	6384	6397	6408	6462	6552	6669			
CMPB	6388	6514	6556	6562											
COM	5819	6313													
DEC	5944	5981	5994	6048	6083	6196	6241	6276	6288	6386	6[illegible]2				
DECB	6518	6623	6634												
EMT	374														
HALT	364	5816	5908	6087	6430	6433	6503	6701							
INC	6349	6629	6637	6691											
INCB	6673	6688													
IOT	375														
JMP	368	5721	5741	5749	5767	5775	5785	5793	5803	5813	5889	5892	5895	5898	5900
	5997	5998	6088	6119	6145	6147	6156	6210	6291	6292	6354	6412	6442	6452	6464
JSR	5996	6290	6411	6469	6513	6520	6697								
MOV	5698	5702	5703	5704	5705	5706	5707	5708	5709	5712	5715	5716	5722	5723	5726
	5729	5730	5733	5736	5737	5738	5739	5740	5745	5746	5747	5748	5753	5754	5757
	5760	5763	5764	5765	5766	5771	5772	5773	5774	5781	5782	5783	5784	5789	5790
	5791	5792	5799	5800	5801	5802	5809	5810	5811	5812	5822	5823	5826	5827	5828
	5832	5833	5834	5838	5839	5840	5844	5845	5846	5850	5851	5852	5856	5857	5858
	5862	5863	5864	5868	5869	5870	5874	5875	5876	5878	5885	5886	5921	5922	5923
	5924	5925	5926	5927	5928	5935	5936	5937	5938	5957	5958	5964	5965	5966	5967
	5974	5976	5978	6008	6009	6010	6012	6013	6016	6017	6018	6019	6022	6026	6031
	6033	6034	6035	6038	6043	6054	6055	6058	6059	6060	6061	6062	6064	6068	6071
	6072	6073	6077	6080	6105	6106	6107	6110	6111	6125	6127	6128	6134	6139	6141
	6144	6155	6164	6165	6168	6169	6170	6174	6179	6184	6185	6198	6192	6207	6208

DZBMD.P11 CROSS REFERENCE TABLE

```
                6219  6220  6221  6222  6223  6224  6225  6232  6233  6234  6235  6252  6253  6259  6260
                6261  6262  6269  6271  6273  6307  6309  6310  6314  6315  6316  6317  6321  6328  6330
                6332  6333  6334  6335  6338  6340  6342  6343  6346  6356  6357  6358  6360  6362  6363
                6364  6365  6366  6367  6368  6369  6370  6374  6375  6376  6377  6390  6392  6393  6394
                6395  6396  6400  6402  6403  6404  6405  6406  6424  6425  6428  6429  6432  6435  6436
                6445  6448  6450  6460  6465  6505  6506  6510  6516  6536  6537  6550  6551  6566  6567
                6568  6569  6598  6606  6607  6608  6614  6621  6639  6640  6641  6642  6643  6664  6665
                6667  6670  6674  6676  6677  6690  6692  6713  6718  6727  6732  6737  6739  6743  6760
                6761  6762  6763  6764  6766  6781  6782  6783  6784  6785  6797  6798  6801
MOVB            6507  6524  6540  6555  6560  6599  6600  6603  6604  6605  6609  6612  6613  6632  6694
                6769  6800
NEG             6610
NOP             5725  5732  5975  6270  6331  6361  6427  6470  6471  6472
RESET           6467
ROL             6351  6383  6616  6618  6619  6620  6622  6659  6772  6774  6776
RTI             6454  6512  6542  6570  6644  6678  6702  6786
RTS             6473  6525  6741  6802
SUB             6693
TRAP            6804  6814  6815  6816  6817  6818  6819
TST             5719  5724  5731  5817  5879  5959  5970  5973  5992  6023  6039  6136  6171  6186  6193
                6202  6254  6265  6268  6286  6311  6381  6426  6431  6509  6627  6666  6699  6745  6780
                6799
TSTB            6501  6522  6538
.ASCII           503   504  5910  6001  6089  6421  6820
.ASCIZ           505  5910  6001  6089  6419  6421  6475  6749  6820
.BLKB           6573
.BLKW           5689  5690  5691  5692
.BYTE            481   482   487   488   499   500   501   502  1023  1024  1025  1026  1027  1028  6571
                6572  6645  6646  6647  6648
.ENABL             1   343
.END            6828
.ENDC            350   369   374   450   464   465   478   503   506   507  5698  5702  5703  5705  5707
                5709  5737  5745  5753  5763  5771  5779  5781  5789  5797  5799  5807  5809  6480  6527
                6543  6544  6551  6553  6558  6573  6574  6575  6651  6657  6659  6661  6672  6673  6675
                6679  6680  6686  6688  6692  6698  6699  6703  6704  6722  6751  6752  6788  6789  6798
                6801  6813  6814  6815  6816  6817  6818  6819  6820  6828
.EQUIV           374   375   377   392   393   412   413   414   415   416   417   418   419   420   421
                 440   441   442   443   444   445   446   447   448   449
.EVEN           5910  6001  6089  6419  6421  6477  6750  6820
.IF              346   364   372   422   450   464   477   503   506  5698  5702  5703  5705  5707  5709
                5729  5737  5745  5753  5763  5771  5779  5781  5789  5797  5799  5807  6479  6526  6543
                6551  6552  6557  6573  6574  6650  6656  6659  6671  6673  6675  6679  6685  6688  6691
                6695  6697  6698  6699  6702  6703  6721  6737  6751  6767  6788  6797  6801  6804  6814
                6815  6816  6817  6818  6819  6820  6828
.IFF             372   465   477   503   507  5702  6480  6527  6543  6544  6553  6557  6574  6575  6651
                6671  6673  6679  6680  6685  6692  6702  6703  6704  6722  6751  6752  6789  6798
.IFT            6673  6698  6771  6787  6788
.IFTF           6673  6697  6767  6771  6787
.IIF             345   350   355   356   364   506  5703  5705  5709  6526  6535  6566  6574  6654  6655
                6675  6679  6683  6684  6685  6703  6719  6744  6813  6814  6815  6816  6817  6818  6819
.IRP            5698  6762  6783
.LIST              1   343   364   464   503   561  5689  5698  5910  6001  6089  6419  6421  6543  6804
                6813  6814  6815  6816  6817  6818  6819  6820  6828
.MACRO             1   464   560  6804
```

DZBMD.P11 CROSS REFERENCE TABLE

ERRORS DETECTED: 0

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

```
                                ...B1   2345                    ...B5                           ...B9   5602                    ...B13
  10                            ...C1   2399                    ...C5   4121                    ...C9                           ...C13
  62P111-DZBMD-J                ...D1   2446                    ...D5   4175                    ...D9   5664   010760  177766   ...D13
 108                            ...E1   2484                    ...E5   4185   007240  001775   ...E9   5684                    ...E13
 148                            ...F1   2538                    ...F5   4197                    ...F9                           ...F13
                                ...G1   2591                    ...G5   4227           ;        ...G9   5702   012014  012706   ...G13
 204                            ...H1   2645                    ...H5                           ...H9   5756   012374  001004   ...H13
 233                            ...I1   2679                    ...I5   4285   007350  177776   ...I9   5810   012736  012767   ...I13
 273                            ...J1   2697   005006  177570   ...J5   4317           ;        ...J9   5864   013260  012767   ...J13
 325                            ...K1                           ...K5                           ...K9                           ...K13
                                ...L1   2757                    ...L5   4376   007444  004000   ...L9   5920                    ...L13
 352                            ...M1   2803   005102  000020   ...M5                           ...M9   5970   014162  005720   ...M13
 406            000040          ...N1   2816                    ...N5   4438           ;        ...N9   6011   014466  062767   ...N13

 460            000034          ...B2   2861   005212  012711   ...B6   4492   007634  001770   ...B10  6065   014734  104402   ...B14
 473    000046  020002          ...C2   2913   005312  100316   ...C6   4502                    ...C10  6099                    ...C14
 515                            ...D2   2952                    ...D6                           ...D10  6153                    ...D14
 569                            ...E2   2981                    ...E6   4565   007732  177762   ...E10  6207   016002  016700   ...E14
 623    001466  060011          ...F2                           ...F6   4582                    ...F10                          ...F14
 677                            ...G2   3038   005426  000012   ...G6   4622                    ...G10  6265   016262  005720   ...G14
 731    001740  000024          ...H2   3091                    ...H6   4644                    ...H10  6304                    ...H14
 757            ;ACTUAL         ...I2   3122                    ...I6                           ...I10  6358   016666  012737   ...I14
 811    002004  113737  ;       ...J2   3166                    ...J6   4700           ;        ...J10  6412   017200  000167   ...J14
 865    002132  000200  ;       ...K2   3188   005674  010045   ...K6                           ...K10         017451     015   ...K14
 919    002246  012700  ;       ...L2   3223   005724  005061   ...L6   4756           ;        ...L10  6432   017626  012637   ...L14
 973    002376  000005  ;       ...M2   3262   005776  000000   ...M6   4795           ;        ...M10  6484                    ...M14
1027    002522  .BYTE           ...N2   3279                    ...N6                           ...N10  6538   020144  105777   ...N14

1081    002664  100376  ;       ...B3                           ...B7   4855                    ...B11  6592                    ...B15
1129            ;173000'        ...C3   3340                    ...C7                           ...C11  6646   020527     000   ...C15
1183    003062  100000          ...D3   3366                    ...D7   4917           ;        ...D11  6700   020714  100001   ...D15
1237    003212  177560          ...E3   3420   006076  001763   ...E7   4928           ;        ...E11  6754                    ...E15
1291    003334  010742          ...F3   3433   006122  005205   ...F7                           ...F11  6808                    ...F15
1345                            ...G3   3487   006236  042700   ...G7   4986   010116  000005   ...G11  BITOS  = 000040         ...G15
1399                            ...H3   3502   006260  005000   ...H7                           ...H11  EXTMAP   013464         ...H15
1453    003472  012761          ...I3   3556                    ...I7   5049   010222  000666   ...I11  PS     = 177776         ...I15
1507                            ...J3   3571   006420  000000   ...J7   5080   010232  000040   ...J11  SRV.R    016024         ...J15
1561    003744  000207          ...K3                           ...K7   5094                    ...K11  TYPON  = 104406         ...K15
1600                            ...L3   3627   006506  000000   ...L7   5127           ;        ...L11  $TPFLG   001151         ...L15
1651    004004  000001  ;       ...M3   3663                    ...M7                           ...M11  SETTRA    6804#   6814  ...M15
1638            ;               ...N3   3699           ;        ...N7   5187                    ...N11                          ...N15

                                ...B4                           ...B8   5232                    ...B12  BIT       5942    5953  ...B16
1761    004226  000340  ;       ...C4   3761   006672  010145   ...C8   5251           ;        ...C12  MOVB      6507    6524  ...C16
1806            ;               ...D4   3797                    ...D8                           ...D12  .TITLE     345     358  ...D16
1857    004350  032711  ;       ...E4   3848                    ...E8   5310           ;        ...E12
1911    004520  030500  ;       ...F4   3875           ;        ...F8   5364                    ...F12
1927                            ...G4                           ...G8                           ...G12
1981            ;               ...H4   3931           ;        ...H8   5387           ;        ...H12
2034                            ...I4                           ...I8                           ...I12
2088                            ...J4                           ...J8   5447   010574  005004   ...J12
2135    004534  010037  ;       ...K4   3993   007004  013700   ...K8   5483           ;        ...K12
2187                            ...L4   4047   007064  173304   ...L8                           ...L12
2240                            ...M4                           ...M8   5539           ;        ...M12
2291                            ...N4   4063           ;        ...N8                           ...N12
```