BM873

RESTART ROM LOADER

MD-11-DZBMD-G

EP-DZBMD-G-DL-A

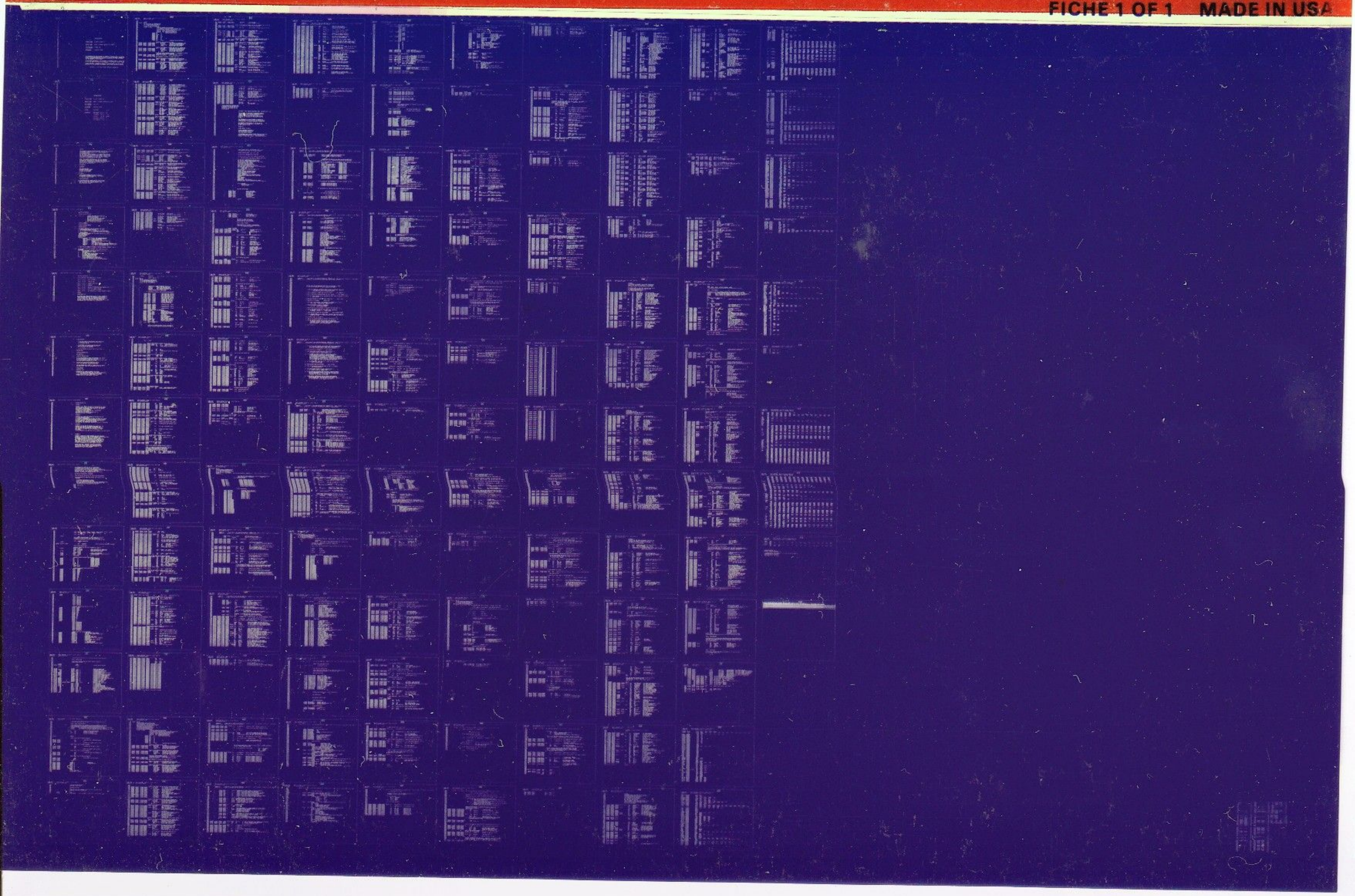
COPYRIGHT © 1976

FICHE 1 OF 1

NOV 1976

digital

MADE IN USA



.REM %

IDENTIFICATION

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

PRODUCT NAME: BM873 - UNIVERSAL RESTART ROM LOADER

DATE RELEASED: OCTOBER 1976

MAINTAINER: DIAGNOSTIC GROUP

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

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

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

COPYRIGHT (C) 1973,1976 DIGITAL EQUIPMENT CORPORATION

```
                              PROGRAM HISTORY
                              ---------------

                  PRODUCT CODE:   MAINDEC-11-DZBMD-G-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
```

```
                        1.      ABSTRACT

                                THIS MAINDEC CONSISTS OF FOUR PROGRAMS. THE TWO MAIN PROGRAMS
                                ARE PROGRAM ONE AND PROGRAM FOUR. THESE PROGRAMS WILL
                                BE DISCUSSED LATER.
                                THE PURPOSE OF THIS DIAGNOSTIC IS TO VERIFY THE DATA IN
                                THE ROM, MAKE SURE ALL ADDRESS WILL CAUSE A TIME OUT TRAP
                                WHEN WRITTEN INTO (EXCEPT THE TRAP VECTORS: 173024,173224)
                                ,AND ALERT THE OPERATOR AS TO WHAT THE OFFSET ADDRESS
                                WOULD BE IF A SELECTED BUTTON IS PUSHED.

                                NOTE:  FOR NORMAL CONFIGURATIONS; THE ONLY PROGRAMS
                                NECESSARY FOR ACCEPTANCE OF THE BM873 ARE PROGRAMS
                                ONE AND FOUR. PROGRAM TWO IS NECESSARY FOR "NON-STANDARD"
                                SETUPS AND IS A MAINTAINCE TOOL. PROGRAM THREE IS
                                ALSO JUST FOR MAINTAINCE AID.

                        2.      REQUIRMENTS

                        2.1     EQUIPMENT

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

                        2.2     STORAGE

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

                        3.      LOADING PROCEDURE

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

```
                                4.      STARTING PROCEDURE

                                4.1     CONTROL SWITCH SETTINGS

                                        SWITCH 00       CLEARED INDICATES ONLY FIRST 128
                                                        WORDS TO BE CHECKED.
                                                        SET INDICATES EXTENDED 128. WORDS
                                                        ARE TO BE CHECKED IN WHICH CASE
                                                        PROGRAM 2 MUST BE RUN FIRST.
                                                        WHEN RUNNING ON BM873Y-B,C,D,F OR G, 256 WORDS
                                                        ARE AUTOMATICALLY CHECKED.

                                4.2     STARTING ADDRESS

                                        STARTING ADDRESS 000200

                                4.3     OPERATOR ACTION

                                4.3.1   FOR NORMAL OPERATION (WITHOUT EXTENDED 128 WORDS)
                                        1. LOAD STARTING ADDRESS (000200)
                                        2. SET SWITCHES AS PER 5.1.1 (NORMAL ALL SWITCHES DOWN)
                                        T                       3. PRESS START SWITCH AND RELEASE.
                                        4. DEVICE VERSION.
                                         WHEN PROGRAM IS STARTED FOR THE FIRST TIME THE FOLLOWING
                                         WILL BE PRINTED OUT:
                                        MAINDEC-11-DZBMDG
                                        DEVICE VERSION
                                        BM873-Y
                                         THE OPERATOR WILL THEN SPECIFY THE VERSION BEING RUN.
                                                BM873-Y*        IS ANY NON-STANDARD VERSION.
                                                NOTE: PROGRAM TWO MUST BE RUN FIRST.
                                                BM873-YA        REPLACES M792-YA,MR11-DB,M792-YH
                                                BM873-YB        MASSBUS
                                                BM873-YC        DDCMP BOOTSTRAP ROM
                                                BM873-YD        KL10 (PDP-11) 256 BOOTSTRAP ROM (VERSION 2(17))
                                                BM873-YF        KL10 (PDP-11) 256 BOOTSTRAP ROM (VERSION 3(23))
                                                BM873-YG        KL10 (PDP-11) 256 BOOTSTRAP ROM

                                        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 OR G)
                                        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 OR G)

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 OR G)

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 OR G 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 OR G 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 OR YG BY COMPARING THE 1ST WORD IN ROM WITH
                                        THE EXPECTED WORD.  THE DIAGNOSTIC THEN RUNS
                                        PROGRAM 1 AND PROGRAM 4 BEFORE ENTERING THE MONITOR.

                                9.6     ELECTRICAL PREQUISITES  (HARDWARE)

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

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

                                9.7.3    IF FURTHER INFORMATION IS NEEDED
                                        CONSULT THE BM873 MANUAL FOR HELP.
                                        NOTE: THE DIAGNOSTIC RUNNING WITHOUT ANY INTERFERANCE FROM
                                        THE USER HAS NO WAY OF CHECKING THE PRESENTS OF THE
                                        "ACLO" AND "DCLO" SIGNALS ON THE OPTION.
                                .NLIST
                                .LIST   SEQ,LOC,BIN
                                .LIST
                                .PAGE
                                .ENDM   HELLO
```

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

                                .SBTTL  TRAP CATCHER

                        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  010000                  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
```

```
                        SW01=   2
000002                  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
000001                  .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  015550                  $ENDAD                          ;LOGICAL END OF PROGRAM

001100                          .=1100

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

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

                        .SBTTL  ERROR POINTER TABLE

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

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


001156                  $ERRTB:

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 0

001156  016764                 EM1             ;"ROM READ DATA COMPARISON ERROR"
001160  017164                 DH1             ;*
001162  017362                 DT1             ;*
001164  000000                 0               ;* PRINT ALL NUMERIC DATA IN OCTAL

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 1

001166  017026                 EM2             ;"WRITTING ROM FAILED TO TRAP"
001170  017263                 DH2
001172  017376                 DT2
001174  000000                 0               ;PRINT ALL NUMERIC DATA IN OCTAL.

                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 2

001176  017064                 EM3             ;"UNEXPECTED TRAP WHILE READING ROM"
001200  017315                 DH3
001202  017404                 DT3
001204  000000                 0
                        ;ERROR TABLE ITEM FOR ERROR MESSAGE 3

001206  017124                 EM4             ;"FATAL TRAP. ROM PC ON STACK."
001210  017263                 DH2
001212  017376                 DT2
001214  000000                 0

001216  000000          LSTERR: 0               ;ERROR FLAG
001220  000000          ICOUNT: 0               ;ITERATION COUNT.
001222  000000          TEMP5:  0
001224  000000          TEMP3:  0
001226  000000          TEMP4:  0
001230  000000          SAVR0:  0
001232  000000          SAVR1:  0
001234  000000          SAVR4:  0
001236  000000          SAVR5:  0
                        .TITLE  JUNE 1976
                        ;*COPYRIGHT (C) BM873 YX
```

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

```
                                .=1400
001400                          MAP.YA:
                                ;THE FOLLOWING IS A REPRODUCTION
                                ;OF THE ROM PROGRAM FOR BM873YA.
                                ;IT IS HERE FOR COMPARISON TO
                                ;ACTUAL ROM AND FOR REFERENCE.
                                ;173000 .=173000        ;STARTING ADDRESS FOR BOOTSTRAP
                                                        ;THIS LOADER IS DESIGNED FOR THE RESTART MODULE M873.
                                                        ;IT FUNCTIONALLY REPLACES THE FOLLOWING ROMS:
                                                        ;M792-YA - PAPER TAPE BOOTSTRAP FOR PC11,KL11
                                                        ;MR11-DB   BULK STORAGE BOOTSTRAP ROM
                                                        ;M792-YH   TA11 CASSETTE BOOTSTRAP ROM
                                ;               000000  R0=     %0      ;REGISTER DEFINITIONS
                                ;               000001  R1=     %1
                                ;               000002  R2=     %2
                                ;               000003  R3=     %3
                                ;               000004  R4=     %4
                                ;               000005  R5=     %5
                                ;               000006  SP=     %6
                                ;               000007  PC=     %7
                                ;               177570  SR=     177570          ;PROCESSOR SWITCH REGISTER
                                                        ;
                                                        ;STARTING LOCATION FOR RF11 DISK
001400  010702  ;173000 010702  RF11:   MOV PC,R2       ;SET POINTER TO PARAMETER LISTS
001402  000464  ;173002 000464          BR OTHER        ;TRANSFER TO SERVICE ROUTINE
001404  177462  ;173004 177462          .WORD 177462    ;DEVICE WORD COUNT ADDRESS
001406  000005  ;173006 000005          .WORD 5         ;DEVICE READ INSTRUCTION
                                                        ;
                                                        ;THIS IS THE STARTING LOCATION FOR THE RK11 CONTROLLER
001410  010702  ;173010 010702  RK11:   MOV PC,R2       ;SET POINTER TO PARAMETER LIST
001412  000460  ;173012 000460          BR OTHER        ;TRANSFER TO SERVICE ROUTINE
001414  177406  ;173014 177406          .WORD 177406    ;DEVICE WORD COUNT REGISTER
001416  000005  ;173016 000005          .WORD 5         ;DEVICE READ INSTRUCTION
                                                        ;
                                                        ;THIS IS A SPARE STARTING LOCATION. IT TRANSFERS TO ADDRESS
                                                        ;CONTAINED IN THE SWITCH REGISTER.
001420  013707  ;173020 013707  TRANSR: MOV @#SR,PC     ;GO TO INDICATED LOCATION
001422  177570  ;173022 177570
                ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
                                                        ;
                                                        ;THIS IS THE POWER UP VECTOR REQUIRED FOR DEVICE AND
001424  173000  ;173024 173000  POWER:  .WORD RF11       ;ADDRESS OF FIRST LOCATION IN ROM
001426  000340  ;173026 000340          .WORD 340       ;PROCESSOR STATUS LEVEL 7
                                                        ;
                                                        ;THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE) CONTROLLER.
001430  010702  ;173030 010702  TC11:   MOV PC,R2       ;SET UP POINTER TO PARAMETER LIST
001432  000426  ;173032 000426          BR TAPES        ;AND TRANSFER TO FIRST ROUTINE
001434  177344  ;173034 177344          .WORD 177344    ;DEVICE WORD COUNT ADDRESS
001436  004003  ;173036 004003          .WORD 4003      ;FIND PREVIOUS BLOCK COMMAND
001440  100000  ;173040 100000          .WORD 100000    ;USED AS DONE INDICATOR
001442  024000  ;173042 024000          .WORD 24000     ;USED AS ERROR INDICATOR/TEST FLAG
001444  000445  ;173044 000445          BR OTHERX       ;THEN TRANSFER TO NEXT ROUTINE
001446  000005  ;173046 000005          .WORD 5         ;DEVICE READ COMMAND
                                                        ;
                                                        ;THIS IS THE START LOCATION FOR TM11 MAGTAPE CONTROLLER
```

```
001450  010702  ;173050 010702  TM11:   MOV PC,R2         ;SET POINTER TO PARAMETER LIST
001452  000416  ;173052 000416          BR TAPES          ;AND TRANSFER TO FIRST ROUTINE
001454  172524  ;173054 172524          .WORD 172524      ;DEVICE BYTE/RECORD COUNT REGISTER
001456  060017  ;173056 060017          .WORD 60017       ;DEVICE REWIND COMMAND
001460  000200  ;173060 000200          .WORD 200         ;DEVICE DONE FLAG
001462  100000  ;173062 100000          .WORD 100000      ;DEVICE ERROR FLAG BIT
001464  000413  ;173064 000413          BR TAPESX         ;THEN TRANSFER TO NEXT SERVICE RTN
001466  060011  ;173066 060011          .WORD 60011       ;DEVICE FORWARD SPACE COMMAND
001470  000200  ;173070 000200          .WORD 200         ;SAME AS ABOVE
001472  100000  ;173072 100000          .WORD 100000      ;SAME AS ABOVE
001474  000431  ;173074 000431          BR OTHERX         ;THEN TRANSFER TO READ/TRANSFER ROUTINE
001476  060003  ;173076 060003          .WORD 60003       ;DEVICE READ COMMAND
                                ;
                                ;THIS IS THE START LOCATION FOR THE RP11 CONTROLLER
001500  010702  ;173100 010702  RP11:   MOV PC,R2         ;SET POINTER TO PARAMETER LIST
001502  000424  ;173102 000424          BR OTHER          ;TRANSFER TO TRANSFER ROUTINE
001504  176716  ;173104 176716          .WORD 176716      ;DEVICE WORD COUNT REGISTER
001506  000005  ;173106 000005          .WORD 5           ;DEVICE READ COMMAND
                                ;
                                ;THIS IS THE TAPE DEVICE SERVICE ROUTINE.
001510  010200  ;173110 010200  TAPES:  MOV R2,R0         ;GET ADDRESS OF PARAMETER LIST
001512  005720  ;173112 005720          TST (R0)+         ;SKIP TWO WORDS FIRST TIME
001514  000005  ;173114 000005  TAPESX: RESET             ;RESET ALL DEVICES
001516  005720  ;173116 005720          TST (R0)+         ;SKIP OVER BRANCH INSTRUCTION
001520  016201  ;173120 016201          MOV 2(R2),R1      ;THEN GET DEVICE WORD/BYTE COUNT ADDRES
001522  000002  ;173122 000002
001524  005311  ;173124 005311          DEC @R1           ;AND SET TO -1
001526  012041  ;173126 012041          MOV (R0)+,-(R1)   ;AND THEN ISSUE COMMAND TO DEVICE
001530  031011  ;173130 031011  TAPWAT: BIT @R0,@R1       ;WAIT FOR DEVICE COMPLETION
001532  001776  ;173132 001776          BEQ TAPWAT        ;BY HANGING IN LOOP
001534  005720  ;173134 005720          TST (R0)+         ;AND THEN SKIP DONE FLAG
001536  032041  ;173136 032041          BIT (R0)+,-(R1)   ;THEN TEST FOR ERROR
001540  001063  ;173140 001063          BNE ERROR         ;THERE IS ONE
001542  000110  ;173142 000110  RETURN: JMP @R0           ;AND TRANSFER TO FOLLOWING INSTRUCTION
                                ;
                                ;THIS IS THE STARTING ADDRESS FOR RC11 DISK CONTROLLERS
001544  010702  ;173144 010702  RC11:   MOV PC,R2         ;SET UP POINTER TO PARAMETER LIST
001546  000402  ;173146 000402          BR OTHER          ;TRANSFER TO SERVICE RTN
001550  177450  ;173150 177450          .WORD 177450      ;DEVICE WORD COUNT REGISTER
001552  000005  ;173152 000005          .WORD 5           ;DEVICE READ INSTRUCTION
                                ;
                                ;THIS ROUTINE PERFORMS THE ACTUAL TRANSFER TO MEMORY OF DATA
001554  010200  ;173154 010200  OTHER:  MOV R2,R0         ;SET POINTER TO LIST IN R0
001556  005720  ;173156 005720          TST (R0)+         ;SKIP TWO WORDS FIRST TIME.
001560  005720  ;173160 005720  OTHERX: TST (R0)+         ;SKIP PAST BR INSTRUCTION
001562  000005  ;173162 000005          RESET             ;REST THE WORLD
001564  016201  ;173164 016201          MOV 2(R2),R1      ;OBTAIN DEVICE WORD COUNT ADDRESS
001566  000002  ;173166 000002
001570  012711  ;173170 012711          MOV #-1000,@R1    ;THEN OBTAIN LARGE WORD COUNT
001572  177000  ;173172 177000
001574  011041  ;173174 011041          MOV @R0,-(R1)     ;AND PUT COMMAND TO DEVICE
001576  105711  ;173176 105711  OTHWAT: TSTB @R1          ;WAIT FOR DONE FLAG
001600  100376  ;173200 100376          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
```

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

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

```
                        ;
                        ;       173000  .=173000
                        ;
                        ;                       ;THIS IS THE STARTING ADDRESS FOR RH11/RS03/04 D
002000  000405 ;173000  000405  RHRSA:  BR      1$      ;ENTRY FOR SELECTING UNI
002002  010703 ;173002  010703  RHRSB:  MOV     PC,R3   ;ENTRY TO SELECT UNITS
002004  113737 ;173004  113737          MOVB    @#CSW,@#RSCSA+10;LOAD UNIT # INS
002006  177570 ;173006  177570
002010  172050 ;173010  172050
002012  000401 ;173012  000401          BR      2$
002014  010703 ;173014  010703  1$:     MOV     PC,R3
002016  012700 ;173016  012700  2$:     MOV     #RSCSA,R0;SET CONTROL STATUS REG
002020  172040 ;173020  172040
002022  000526 ;173022  000526          BR      RHCOMN
                        ;
                        ;THIS IS THE AUTO LOAD VECTOR
002024  173000 ;173024  173000          .WORD   RHRSA
002026  000340 ;173026  000340          .WORD   340
                        ;
                        ;THIS IS THE STARTING ADDRESS FOR RK11 CONTROLLE
002030  000412 ;173030  000412  RK11A:  BR      2$      ;ENTRY TO SELECT UNIT 0
002032  010703 ;173032  010703  RK11B:  MOV     PC,R3   ;ENTRY TO SELECT ALL UNI
                        ;                               ;SAVE ERROR RETRY ADDRES
002034  113705 ;173034  113705          MOVB    @#CSW,R5;SET POINTER TO PARAMETE
002036  177570 ;173036  177570
002040  052705 ;173040  052705          BIS     #10,R5  ;SET POSITION BIT
002042  000010 ;173042  000010
002044  006105 ;173044  006105  1$:     ROL     R5      ;SHIFT UNIT # TO BIT 13-
002046  103376 ;173046  103376          BCC     1$      ;KEEP GOING
002050  010537 ;173050  010537          MOV     R5,@#RKWC+4;MOVE IN TO RKDA REGI
002052  177412 ;173052  177412
002054  000401 ;173054  000401          BR      3$      ;SKIP NEXT INSTRUCTION
002056  010703 ;173056  010703  2$:     MOV     PC,R3   ;SAVE ERROR RETRY ADDRES
002060  010702 ;173060  010702  3$:     MOV     PC,R2
002062  000546 ;173062  000546          BR      OTHERA
002064  177406 ;173064  177406          .WORD   RKWC
002066  000005 ;173066  000005          .WORD   RFREAD
                        ;
                        ;THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE)
002070  010703 ;173070  010703  TC11:   MOV     PC,R3   ;SAVE ERROR RETRY ADDRES
002072  010702 ;173072  010702          MOV     PC,R2
002074  000570 ;173074  000570          BR      TAPES
002076  177344 ;173076  177344          .WORD   TCWC
002100  000005 ;173100  000005          .WORD   RFREAD
002102  004003 ;173102  004003          .WORD   RNUM
002104  100000 ;173104  100000          .WORD   100000  ;DONE MASK
002106  024000 ;173106  024000          .WORD   24000   ;ERROR MASK
                        ;
                        ;
                        ;
                        ;TM11 STARTING ADDRESS
002110  010703 ;173110  010703  TM11:   MOV     PC,R3   ;SAVE ERROR RETRY ADDRES
002112  012737 ;173112  012737          MOV     #TMRWND,@#TMWC-2;REWIND TAPE
002114  060017 ;173114  060017
002116  172522 ;173116  172522
```

```
002120  010702 ;173120  010702           MOV     PC,R2
002122  000555 ;173122  000555           BR      TAPES
002124  172524 ;173124  172524           .WORD   TMWC
002126  060003 ;173126  060003           .WORD   TMREAD  ;TM11 READ COMMAND
002130  060011 ;173130  060011           .WORD   TMFWRD  ;TM11 FORWARD RECORD COM
002132  000200 ;173132  000200           .WORD   200     ;DONE MASK
002134  100000 ;173134  100000           .WORD   100000  ;ERROR MASK
               ;                ;
               ;                ;THIS IS THE STARTING ADDRESS FOR RF11 CONTROLLE
002136  010703 ;173136  010703  RF11:    MOV     PC,R3   ;SAVE ERROR RETRY ADDRES
002140  010702 ;173140  010702           MOV     PC,R2   ;SET POINTER TO PARAMETE
002142  000516 ;173142  000516           BR      OTHERA  ;GO TO COMMON SERVICE RO
               ;                                 ;ASSUME UNIT 0
002144  177462 ;173144  177462           .WORD   RFWC    ;DEVICE WORD COUNT REGIS
002146  000005 ;173146  000005           .WORD   RFREAD  ;READ COMMAND
               ;                ;
               ;                ;THIS IS THE STARTING ADDRESS FOR RH/TU16/TM02
002150  010703 ;173150  010703  TU16:    MOV     PC,R3   ;SAVE ERROR RETRY ADDRES
002152  012700 ;173152  012700           MOV     #TUCS,R0;GET CONTROL STATUS WORD
002154  172440 ;173154  172440
002156  012710 ;173156  012710  TU16RE:  MOV     #RHPRST,(R0);REWIND TAPE CLEAR E
002160  000021 ;173160  000021
002162  012760 ;173162  012760           MOV     #TUMODE,32(R0);SET 800 BPI NORMA
002164  001300 ;173164  001300
002166  000032 ;173166  000032
002170  012760 ;173170  012760           MOV     #-1,6(R0);LOAD FRAME COUNT
002172  177777 ;173172  177777
002174  000006 ;173174  000006
002176  012710 ;173176  012710           MOV     #TUSPAC,(R0);SPACE FORWARD
002200  000031 ;173200  000031
002202  105760 ;173202  105760  1$:      TSTB    12(R0)
002204  000012 ;173204  000012
002206  100375 ;173206  100375           BPL     1$      ;KEEP LOOPING
002210  000433 ;173210  000433           BR      RHCOMN
               ;                ;
               ;                ;THIS IS THE STARTING ADDRESS FOR RC11 CONTROLLE
002212  010703 ;173212  010703  RC11:    MOV     PC,R3
002214  010702 ;173214  010702           MOV     PC,R2   ;ASSUME UNIT 0
002216  000470 ;173216  000470           BR      OTHERA
002220  177450 ;173220  177450           .WORD   RCWC
002222  000005 ;173222  000005           .WORD   RFREAD
               ;                ;
               ;                ;THIS IS THE AUTO LOAD VECTOR
002224  173000 ;173224  173000           .WORD   RHRSA
002226  000340 ;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,RO
002250 176300 ;173250 176300
002252 032760 ;173252 032760  RPCOMN:  BIT     #TUTAPE,26(RO);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,(RO);RESET DRIVE
002264 000021 ;173264 000021
002266 012760 ;173266 012760           MOV     #14000,32(RO);SET 16 BIT FORMAT
002270 014000 ;173270 014000
002272 000032 ;173272 000032
002274 012710 ;173274 012710           MOV     #DRCLR,(RO);CLEAR DRIVE ERROR
002276 000011 ;173276 000011
              ;                                ; (GENERATED IF RS03/04
002300 005720 ;173300 005720  RHCOMN:  TST     (RO)+   ;MOVE TO WORD COUNT ADDR
002302 010037 ;173302 010037           MOV     RO,@#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,RO
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
002354 113705 ;173354 113705           MOVB    @#CSW,R5
002356 177570 ;173356 177570
002360 000305 ;173360 000305           SWAB    R5      ;GET UNIT # INTO HIGH BY
002362 000402 ;173362 000402           BR      3$
002364 010703 ;173364 010703  1$:      MOV     PC,R3
002366 005005 ;173366 005005           CLR     R5
002370 010702 ;173370 010702  3$:      MOV     PC,R2
002372 000403 ;173372 000403           BR      OTHER
```

```
002374  176716 ;173374  176716              .WORD   RPWC
002376  000005 ;173376  000005              .WORD   RFREAD
        ;               ;
002400  005005 ;173400  005005  OTHERA: CLR     R5        ;SET TO UNIT 0
002402  010200 ;173402  010200  OTHER:  MOV     R2,R0     ;R0 POINT AT WORD COUNT
002404  005720 ;173404  005720          TST     (R0)+     ;POINT TO PARAMETER LIST
002406  012001 ;173406  012001          MOV     (R0)+,R1;MOVE WORD COUNT ADDRESS
002410  012711 ;173410  012711          MOV     #-256.*2,(R1);LOAD WORD COUNT
002412  177000 ;173412  177000
002414  051005 ;173414  051005          BIS     (R0),R5 ;COMBINE UNIT # WITH COM
002416  010541 ;173416  010541          MOV     R5,-(R1);LOAD READ COMMAND
002420  032711 ;173420  032711          BIT     #100200,(R1);CHECK FOR ERROR AND
002422  100200 ;173422  100200
002424  001775 ;173424  001775          BEQ     .-4       ;WAIT UNTIL COMPLETE
002426  100012 ;173426  100012          BPL     1$        ;NO ERROR
002430  005702 ;173430  005702          TST     R2        ;WAS IT CALLED BY MASS B
002432  001024 ;173432  001024          BNE     AGAIN     ;NO. ERROR
002434  032761 ;173434  032761          BIT     #TUTAPE,26(R1);IS TU16?
002436  040000 ;173436  040000
002440  000026 ;173440  000026
002442  001420 ;173442  001420          BEQ     AGAIN     ;NO. ERROR
002444  022761 ;173444  022761          CMP     #FCE,14(R1);ARE WE READ A SHORT
002446  001000 ;173446  001000
002450  000014 ;173450  000014
002452  001014 ;173452  001014          BNE     AGAIN     ;SOME OTHER ERROR
002454  005007 ;173454  005007  1$:     CLR     PC        ;O.K.
        ;               ;
        ;               ;THIS IS THE TAPE DEVICE SERVICE ROUTINE
002456  010200 ;173456  010200  TAPES:  MOV     R2,R0     ;GET THE ADDRESS OF THE
002460  005720 ;173460  005720          TST     (R0)+     ;STEP TO LAST COMMAND
002462  012001 ;173462  012001          MOV     (R0)+,R1;GET THE WORD COUNT ADDR
002464  005311 ;173464  005311          DEC     (R1)      ;SET UP TO ADVANCE 1 REC
002466  005720 ;173466  005720          TST     (R0)+     ;MOVE R0 TO FIRST COMMAN
002470  012041 ;173470  012041          MOV     (R0)+,-(R1);LOAD COMMAND REG.
002472  031011 ;173472  031011          BIT     (R0),(R1);DONE?
002474  001776 ;173474  001776          BEQ     .-2       ;NO. KEEP LOOPING
002476  005720 ;173476  005720          TST     (R0)+     ;YES. CHECK FOR ERROR
002500  031041 ;173500  031041          BIT     (R0),-(R1);ANY ERROR?
002502  001736 ;173502  001736          BEQ     OTHERA    ;NO ERROR- TRY TO READ
002504  000005 ;173504  000005  AGAIN:  RESET
        ;
002506  000113 ;173506  000113          JMP     (R3)      ;ERROR RETURN
        ;               ;
        ;               ;THIS IS THE STARTING ADDRESS FOR PC11 PAPER TAP
002510  012704 ;173510  012704  KL11:   MOV     #KLCS,R4;OBTAIN CONTROL REG.
002512  177560 ;173512  177560
002514  000443 ;173514  000443          BR      CKDEV     ;AND TRANSFER TO READER
        ;
        ;               ;
        ;
        ;               ;CASSETTE TAPE DEVICE COMMAND TABLE
002516  .BYTE   240 ;173516     240  TABLE:  .BYTE   240     ;COMPARE WORD NOT A COMM
002517  .BYTE   037 ;173517     037          .BYTE   37      ;ILBS+RWD+GO
002520  .BYTE   015 ;173520     015          .BYTE   15      ;SPACE FORWARD BLOCK+GO
002521  .BYTE   005 ;173521     005          .BYTE   5       ;READ
002522  .BYTE   024 ;173522     024          .BYTE   24      ;READ +ILBS
```

```
1017  002523  .BYTE   224 ;173523        224          .BYTE   224     ;READ+ILBS+END FLAG
1018          ;                          ;
1019          ;                          ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE D
1020  002524  000404 ;173524  000404  CBOOTA: BR      1$          ;SELECT UNIT 0
1021  002526  113704 ;173526  113704  CBOOTB: MOVB    @#CSW,R4;SELECT UNITS
1022  002530  177570 ;173530  177570
1023  002532  000304 ;173532  000304          SWAB    R4
1024  002534  000401 ;173534  000401          BR      RESETX
1025  002536  005004 ;173536  005004  1$:     CLR     R4
1026  002540  012700 ;173540  012700  RESETX: MOV     #TACS,R0;GET CONTROL REG.
1027  002542  177500 ;173542  177500
1028  002544  000005 ;173544  000005  RESTRT: RESET
1029  002546  010410 ;173546  010410          MOV     R4,(R0) ;SELECT UNIT
1030  002550  012701 ;173550  012701          MOV     #TABLE,R1
1031  002552  173516 ;173552  173516
1032  002554  012702 ;173554  012702          MOV     #375,R2 ;LOAD TRANSFER COUNTER
1033  002556  000375 ;173556  000375
1034  002560  112103 ;173560  112103          MOVB    (R1)+,R3;LOAD COMPARATOR
1035  002562  112110 ;173562  112110  LOOP1:  MOVB    (R1)+,(R0);LOAD COMMAND
1036  002564  100407 ;173564  100407          BMI     DONE
1037  002566  130310 ;173566  130310  LOOP2:  BITB    R3,(R0) ;COMMAND COMPLETE?
1038  002570  001776 ;173570  001776          BEQ     LOOP2   ;NO. WAIT
1039  002572  105202 ;173572  105202          INCB    R2      ;INCREMENT ADDRESS CTR.
1040  002574  100772 ;173574  100772          BMI     LOOP1   ;IF (-), GET COMMAND
1041  002576  116012 ;173576  116012          MOVB    2(R0),(R2);STORE DATA
1042  002600  000002 ;173600  000002
1043  002602  000771 ;173602  000771          BR      LOOP2   ;GET ANOTHER BYTE
1044  002604  005710 ;173604  005710  DONE:   TST     (R0)    ;ANY ERROR?
1045  002606  100756 ;173606  100756          BMI     RESTRT  ;YES, RETRY
1046  002610  005002 ;173610  005002          CLR     R2      ;CLEAR COMPARE ADDRESS
1047  002612  120312 ;173612  120312          CMPB    R3,(R2) ;IT MUST BE 240
1048  002614  001377 ;173614  001377          BNE     .
1049  002616  000112 ;173616  000112  ERROR:  JMP     (R2)
1050          ;                          ;
1051          ;                          ;THIS IS THE STARTING ADDRESS FOR THE PC11 CONTR
1052  002620  012704 ;173620  012704  PC11:   MOV     #PCCS,R4
1053  002622  177550 ;173622  177550
1054  002624  000005 ;173624  000005  CKDEV:  RESET
1055  002626  012701 ;173626  012701          MOV     #160000,R1;SET UP MEMORY TEST LI
1056  002630  160000 ;173630  160000
1057  002632  012702 ;173632  012702          MOV     #6,R2   ;SET UP POINTER TO TIMEO
1058  002634  000006 ;173634  000006
1059  002636  012712 ;173636  012712          MOV     #340,(R2);SET UP VECTOR TO RETUR
1060  002640  000340 ;173640  000340
1061  002642  010742 ;173642  010742          MOV     PC,-(R2);SAVE PC
1062  002644  012706 ;173644  012706          MOV     #24,SP  ;LOAD UP STACK POINTER
1063  002646  000024 ;173646  000024
1064  002650  010441 ;173650  010441          MOV     R4,-(R1);LOOK FOR END OF MEMORY
1065  002652  040601 ;173652  040601          BIC     SP,R1   ;THEN DROP TO XX752
1066  002654  010111 ;173654  010111          MOV     R1,(R1) ;AND STORE IN ITSELF
1067  002656  011102 ;173656  011102  LOOP:   MOV     (R1),R2
1068  002660  005214 ;173660  005214          INC     (R4)    ;START DEVICE
1069  002662  105714 ;173662  105714  RDRWAT: TSTB    (R4)    ;WAIT
1070  002664  100376 ;173664  100376          BPL     RDRWAT
1071  002666  116412 ;173666  116412          MOVB    2(R4),(R2);SAVE THE DATA
1072  002670  000002 ;173670  000002
```

```
1073  002672  005211 ;173672  005211          INC     (R1)
1074  002674  120227 ;173674  120227          CMPB    R2,#375
1075  002676  000375 ;173676  000375
1076  002700  001366 ;173700  001366          BNE     LOOP    ;NO
1077  002702  105222 ;173702  105222          INCB    (R2)+   ;YES
1078  002704  000142 ;173704  000142          JMP     -(R2)
1079  002706  000000 ;173706  000000  ;THIS AREA IS UNUSED
1080  002710  000000 ;173710  000000  ;THIS AREA IS UNUSED
1081  002712  000000 ;173712  000000  ;THIS AREA IS UNUSED
1082  002714  000000 ;173714  000000  ;THIS AREA IS UNUSED
1083  002716  000000 ;173716  000000  ;THIS AREA IS UNUSED
1084  002720  000000 ;173720  000000  ;THIS AREA IS UNUSED
1085  002722  000000 ;173722  000000  ;THIS AREA IS UNUSED
1086  002724  000000 ;173724  000000  ;THIS AREA IS UNUSED
1087  002726  000000 ;173726  000000  ;THIS AREA IS UNUSED
1088  002730  000000 ;173730  000000  ;THIS AREA IS UNUSED
1089  002732  000000 ;173732  000000  ;THIS AREA IS UNUSED
1090  002734  000000 ;173734  000000  ;THIS AREA IS UNUSED
1091  002736  000000 ;173736  000000  ;THIS AREA IS UNUSED
1092  002740  000000 ;173740  000000  ;THIS AREA IS UNUSED
1093  002742  000000 ;173742  000000  ;THIS AREA IS UNUSED
1094  002744  000000 ;173744  000000  ;THIS AREA IS UNUSED
1095  002746  000000 ;173746  000000  ;THIS AREA IS UNUSED
1096  002750  000000 ;173750  000000  ;THIS AREA IS UNUSED
1097  002752  000000 ;173752  000000  ;THIS AREA IS UNUSED
1098  002754  000000 ;173754  000000  ;THIS AREA IS UNUSED
1099  002756  000000 ;173756  000000  ;THIS AREA IS UNUSED
1100  002760  000000 ;173760  000000  ;THIS AREA IS UNUSED
1101  002762  000000 ;173762  000000  ;THIS AREA IS UNUSED
1102  002764  000000 ;173764  000000  ;THIS AREA IS UNUSED
1103  002766  000000 ;173766  000000  ;THIS AREA IS UNUSED
1104  002770  000000 ;173770  000000  ;THIS AREA IS UNUSED
1105  002772  000000 ;173772  000000  ;THIS AREA IS UNUSED
1106  002774  000000 ;173774  000000  ;THIS AREA IS UNUSED
1107  002776  END.YB:
1108  002776  000000 ;173776  000000  ;THIS AREA IS UNUSED
```

```
1109  003000  MAP.YC:
1110          ;THE FOLLOWING 1000 LOCATIONS ARE
1111          ;A REPRODUCTION OF THE ROM PROGRAM
1112          ;FOR THE BM873YC. THE FIRST 400 LOCATIONS
1113          ;ARE AN EXACT COPY OF THE BM873YA. THE
1114          ;REMAINING 400 LOCATIONS ARE
1115          ;THE DDCMP BOOTSTRAP ROM PROGRAM.
1116          ;IT IS HERE FOR COMPARISON TO
1117          ;ACTUAL ROM AND FOR REFERENCE.
1118          ;173000 .=173000          ;STARTING ADDRESS FOR BOOTSTRAP
1119                                    ;THIS LOADER IS DESIGNED FOR THE RESTART MODULE M873.
1120                                    ;IT FUNCTIONALLY REPLACES THE FOLLOWING ROMS:
1121                                    ;M792-YA - PAPER TAPE BOOTSTRAP FOR PC11,KL11
1122                                    ;MR11-DB   BULK STORAGE BOOTSTRAP ROM
1123                                    ;M792-YH   TA11 CASSETTE BOOTSTRAP ROM
1124          ;               000000  R0=     %0      ;REGISTER DEFINITIONS
1125          ;               000001  R1=     %1
1126          ;               000002  R2=     %2
1127          ;               000003  R3=     %3
1128          ;               000004  R4=     %4
1129          ;               000005  R5=     %5
1130          ;               000006  SP=     %6
1131          ;               000007  PC=     %7
1132          ;               177570  SR=     177570          ;PROCESSOR SWITCH REGISTER
1133                                    ;
1134                                    ;STARTING LOCATION FOR RF11 DISK
1135  003000  010702  ;173000 010702  RF11:   MOV PC,R2        ;SET POINTER TO PARAMETER LISTS
1136  003002  000464  ;173002 000464          BR OTHER        ;TRANSFER TO SERVICE ROUTINE
1137  003004  177462  ;173004 177462          .WORD 177462    ;DEVICE WORD COUNT ADDRESS
1138  003006  000005  ;173006 000005          .WORD 5         ;DEVICE READ INSTRUCTION
1139                                    ;
1140                                    ;THIS IS THE STARTING LOCATION FOR THE RK11 CONTROLLER
1141  003010  010702  ;173010 010702  RK11:   MOV PC,R2        ;SET POINTER TO PARAMETER LIST
1142  003012  000460  ;173012 000460          BR OTHER        ;TRANSFER TO SERVICE ROUTINE
1143  003014  177406  ;173014 177406          .WORD 177406    ;DEVICE WORD COUNT REGISTER
1144  003016  000005  ;173016 000005          .WORD 5         ;DEVICE READ INSTRUCTION
1145                                    ;
1146                                    ;THIS IS A SPARE STARTING LOCATION. IT TRANSFERS TO ADDRESS
1147                                    ;CONTAINED IN THE SWITCH REGISTER.
1148  003020  013707  ;173020 013707  TRANSR: MOV @#SR,PC     ;GO TO INDICATED LOCATION
1149  003022  177570  ;173022 177570
1150          ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
1151                                    ;
1152                                    ;THIS IS THE POWER UP VECTOR REQUIRED FOR DEVICE AND
1153  003024  173000  ;173024 173000  POWER:  .WORD RF11       ;ADDRESS OF FIRST LOCATION IN ROM
1154  003026  000340  ;173026 000340          .WORD 340       ;PROCESSOR STATUS LEVEL 7
1155                                    ;
1156                                    ;THIS IS THE STARTING ADDRESS FOR TC11 (DECTAPE) CONTROLLER.
1157  003030  010702  ;173030 010702  TC11:   MOV PC,R2        ;SET UP POINTER TO PARAMETER LIST
1158  003032  000426  ;173032 000426          BR TAPES        ;AND TRANSFER TO FIRST ROUTINE
1159  003034  177344  ;173034 177344          .WORD 177344    ;DEVICE WORD COUNT ADDRESS
1160  003036  004003  ;173036 004003          .WORD 4003      ;FIND PREVIOUS BLOCK COMMAND
1161  003040  100000  ;173040 100000          .WORD 100000    ;USED AS DONE INDICATOR
1162  003042  024000  ;173042 024000          .WORD 24000     ;USED AS ERROR INDICATOR/TEST FLAG
1163  003044  000445  ;173044 000445          BR OTHERX       ;THEN TRANSFER TO NEXT ROUTINE
1164  003046  000005  ;173046 000005          .WORD 5         ;DEVICE READ COMMAND
```

```
1165
1166
1167  003050  010702  ;173050 010702  TM11:   MOV PC,R2          ;SET POINTER TO PARAMETER LIST
1168  003052  000416  ;173052 000416          BR TAPES           ;AND TRANSFER TO FIRST ROUTINE
1169  003054  172524  ;173054 172524          .WORD 172524       ;DEVICE BYTE/RECORD COUNT REGISTER
1170  003056  060017  ;173056 060017          .WORD 60017        ;DEVICE REWIND COMMAND
1171  003060  000200  ;173060 000200          .WORD 200          ;DEVICE DONE FLAG
1172  003062  100000  ;173062 100000          .WORD 100000       ;DEVICE ERROR FLAG BIT
1173  003064  000413  ;173064 000413          BR TAPESX          ;THEN TRANSFER TO NEXT SERVICE RTN
1174  003066  060011  ;173066 060011          .WORD 60011        ;DEVICE FORWARD SPACE COMMAND
1175  003070  000200  ;173070 000200          .WORD 200          ;SAME AS ABOVE
1176  003072  100000  ;173072 100000          .WORD 100000       ;SAME AS ABOVE
1177  003074  000431  ;173074 000431          BR OTHERX          ;THEN TRANSFER TO READ/TRANSFER ROUTINE
1178  003076  060003  ;173076 060003          .WORD 60003        ;DEVICE READ COMMAND
1179
1180                                          ;
                                              ;THIS IS THE START LOCATION FOR THE RP11 CONTROLLER
1181  003100  010702  ;173100 010702  RP11:   MOV PC,R2          ;SET POINTER TO PARAMETER LIST
1182  003102  000424  ;173102 000424          BR OTHER           ;TRANSFER TO TRANSFER ROUTINE
1183  003104  176716  ;173104 176716          .WORD 176716       ;DEVICE WORD COUNT REGISTER
1184  003106  000005  ;173106 000005          .WORD 5            ;DEVICE READ COMMAND
1185
1186                                          ;
                                              ;THIS IS THE TAPE DEVICE SERVICE ROUTINE.
1187  003110  010200  ;173110 010200  TAPES:  MOV R2,R0          ;GET ADDRESS OF PARAMETER LIST
1188  003112  005720  ;173112 005720          TST (R0)+          ;SKIP TWO WORDS FIRST TIME
1189  003114  000005  ;173114 000005  TAPESX: RESET              ;RESET ALL DEVICES
1190  003116  005720  ;173116 005720          TST (R0)+          ;SKIP OVER BRANCH INSTRUCTION
1191  003120  016201  ;173120 016201          MOV 2(R2),R1       ;THEN GET DEVICE WORD/BYTE COUNT ADDRES
1192  003122  000002  ;173122 000002
1193  003124  005311  ;173124 005311          DEC @R1            ;AND SET TO -1
1194  003126  012041  ;173126 012041          MOV (R0)+,-(R1)    ;AND THEN ISSUE COMMAND TO DEVICE
1195  003130  031011  ;173130 031011  TAPWAT: BIT @R0,@R1        ;WAIT FOR DEVICE COMPLETION
1196  003132  001776  ;173132 001776          BEQ TAPWAT         ;BY HANGING IN LOOP
1197  003134  005720  ;173134 005720          TST (R0)+          ;AND THEN SKIP DONE FLAG
1198  003136  032041  ;173136 032041          BIT (R0)+,-(R1)    ;THEN TEST FOR ERROR
1199  003140  001063  ;173140 001063          BNE ERROR          ;THERE IS ONE
1200  003142  000110  ;173142 000110  RETURN: JMP @R0            ;AND TRANSFER TO FOLLOWING INSTRUCTION
1201                                          ;
1202                                          ;THIS IS THE STARTING ADDRESS FOR RC11 DISK CONTROLLERS
1203  003144  010702  ;173144 010702  RC11:   MOV PC,R2          ;SET UP POINTER TO PARAMETER LIST
1204  003146  000402  ;173146 000402          BR OTHER           ;TRANSFER TO SERVICE RTN
1205  003150  177450  ;173150 177450          .WORD 177450       ;DEVICE WORD COUNT REGISTER
1206  003152  000005  ;173152 000005          .WORD 5            ;DEVICE READ INSTRUCTION
1207
1208                                          ;
                                              ;THIS ROUTINE PERFORMS THE ACTUAL TRANSFER TO MEMORY OF DATA
1209  003154  010200  ;173154 010200  OTHER:  MOV R2,R0          ;SET POINTER TO LIST IN R0
1210  003156  005720  ;173156 005720          TST (R0)+          ;SKIP TWO WORDS FIRST TIME.
1211  003160  005720  ;173160 005720  OTHERX: TST (R0)+          ;SKIP PAST BR INSTRUCTION
1212  003162  000005  ;173162 000005          RESET              ;REST THE WORLD
1213  003164  016201  ;173164 016201          MOV 2(R2),R1       ;OBTAIN DEVICE WORD COUNT ADDRESS
1214  003166  000002  ;173166 000002
1215  003170  012711  ;173170 012711          MOV #-1000,@R1     ;THEN OBTAIN LARGE WORD COUNT
1216  003172  177000  ;173172 177000
1217  003174  011041  ;173174 011041          MOV @R0,-(R1)      ;AND PUT COMMAND TO DEVICE
1218  003176  105711  ;173176 105711  OTHWAT: TSTB @R1           ;WAIT FOR DONE FLAG
1219  003200  100376  ;173200 100376          BPL OTHWAT         ;BY HANGING IN LOOP
1220  003202  005711  ;173202 005711          TST @R1            ;THEN TEST FOR ERROR
```

```
1221  003204  100441  ;173204 100441          BMI ERROR       ;GOT PROBLEMS
1222  003206  005007  ;173206 005007          CLR PC          ;AND TRANSFER TO ZERO
1223                                          ;
1224                                          ;THIS IS THE STARTING ADDRESS FOR THE PC11 PAPER TAPE CONTROLLER
1225  003210  012704  ;173210 012704  KL11:   MOV #177560,R4  ;OBTAIN DEVICE ADDRESS
1226  003212  177560  ;173212 177560
1227  003214  000440  ;173214 000440          BR CKDEV        ;AND TRANSFER TO READER SERVICE ROUTINE
1228                                          ;
1229
1230                                          ;THIS IS THE CASSETTE DEVICE COMMAND TABLE
1231  003216  017640  ;173216    240  TABLE:   .BYTE 240       ;COMPARE WORD NOT A COMMAND
1232                  ;173217    037          .BYTE 37        ;ILBS+RWD+GO
1233  003220  002415  ;173220    015          .BYTE 15        ;SPACE FORWARD BLOCK+GO
1234                  ;173221    005          .BYTE 5         ;READ+GO
1235  003222  112024  ;173222    024          .BYTE 24        ;READ+ILBS
1236                  ;173223    224          .BYTE 224       ;READ+ILBS+END FLAG
1237          ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
1238                                          ;
1239                                          ;THIS IS AN ADDITIONAL POWER VECTOR ADDRESS REQUIRED BY DEVICE
1240  003224  173000  ;173224 173000  POWER2: .WORD RF11       ;ADDRESS OF BEGINNING OF BOOTSTRAP
1241  003226  000340  ;173226 000340          .WORD 340       ;PRIORITY LEVEL 7
1242                                          ;
1243                                          ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE DEVICE #0
1244  003230  005004  ;173230 005004  CBOOT:   CLR R4          ;LOAD DEVICE NUMBER 0 IN R4
1245  003232  012700  ;173232 012700  RESTX:   MOV #177500,R0  ;GET DEVICE ADDTESS
1246  003234  177500  ;        177500
1247  003236  000005  ;173236 000005  RESTRT: RESET           ;ISSUE RESET INSTRUCTION
1248  003240  010410  ;173240 010410          MOV R4,@R0      ;LOAD DEVICE WITH UNIT NUMBER
1249  003242  012701  ;173242 012701          MOV #TABLE,R1   ;GET FUNNY TABLE OF INSTRUCTIONS
1250  003244  173216  ;173244 173216
1251  003246  012702  ;173246 012702          MOV #375,R2     ;AND LOAD UP TRANSFER COUNTER
1252  003250  000375  ;173250 000375
1253  003252  112103  ;173252 112103          MOVB (R1)+,R3   ;THE LOAD UP COMPARATOR
1254  003254  112110  ;173254 112110  LOOP1:  MOVB (R1)+,@R0  ;LOAD DEVICE REGISTER WITH COMMAND
1255  003256  100407  ;173256 100407          BMI DONE
1256  003260  130310  ;173260 130310  LOOP2:  BITB R3,@R0     ;HAS COMMAND COMPLETED
1257  003262  001776  ;173262 001776          BEQ LOOP2       ;NO, WAIT
1258  003264  105202  ;173264 105202          INCB R2         ;THEN INCREMENT ADDRESS CTR
1259  003266  100772  ;173266 100772          BMI LOOP1       ;IF NEGATIVE, GET COMMAND
1260  003270  116012  ;173270 116012          MOVB 2(R0),@R2  ;AND STORE DATA AWAY
1261  003272  000002  ;173272 000002
1262  003274  000771  ;173274 000771          BR LOOP2        ;GO GET ANOTHER BYTE
1263  003276  005710  ;173276 005710  DONE:   TST @R0         ;ANY DEVICE ERRORS
1264  003300  100756  ;173300 100756          BMI RESTRT      ;YES, RETRY
1265  003302  005002  ;173302 005002          CLR R2          ;CLEAR COMPARE ADDRESS AND TRANSFER ADDRESS
1266  003304  120312  ;173304 120312          CMPB R3,@R2     ;IT MUST BE 240
1267  003306  001377  ;173306 001377          BNE .+0         ;NO, THERE WAS AN ERROR
1268  003310  000112  ;173310 000112  ERROR:   JMP @R2         ;NORMAL CASSETTE AND ERROR FOR BULK STORAGE
1269                                          ;
1270                                          ;THIS IS THE STARTING LOCATION FOR THE PC11 CONTROLLER
1271  003312  012704  ;173312 012704  PC11:    MOV #177550,R4  ;LOAD DEVICE ADDRESS
1272  003314  177550  ;173314 177550
1273  003316  000005  ;173316 000005  CKDEV:   RESET           ;KILL ALL DEVICE ACTION
1274  003320  012701  ;173320 012701          MOV #160000,R1  ;THEN SET UP MEMORY TEST LIMITS
1275  003322  160000  ;173322 160000
1276  003324  012702  ;173324 012702          MOV #6,R2       ;AND SET UP POINTER TO TIMEOUT LOCATION
```

```
1277  003326  000006  ;173326 000006
1278  003330  012712  ;173330 012712          MOV #340,@R2     ;AND SET UP VECTOR TO RETURN TO NEXT
1279  003332  000340  ;173332 000340
1280  003334  010742  ;173334 010742          MOV PC,-(R2)     ;SAVE THE PC
1281  003336  012706  ;173336 012706          MOV #24,SP       ;AND LOAD UP STACK POINTER
1282  003340  000024  ;173340 000024
1283  003342  010441  ;173342 010441          MOV R4,-(R1)     ;AND LOOK FOR END OF MEMORY
1284  003344  040601  ;173344 040601          BIC SP,R1        ;THEN DROP TO XX7752
1285  003346  010111  ;173346 010111          MOV R1,@R1       ;AND STORE IN ITSELF
1286  003350  011102  ;173350 011102  LOOP:   MOV @R1,R2       ;THEN LOAD ADDRESS FOR DATA INSERTION
1287  003352  005214  ;173352 005214          INC @R4          ;AND START DEVICE
1288  003354  105714  ;173354 105714  RDRWAT: TSTB @R4         ;THEN WAIT FOR CHARACTER AVAILABLE
1289  003356  100376  ;173356 100376          BPL RDRWAT       ;HANGING THERE IF NECCESSARY
1290  003360  116412  ;173360 116412          MOVB 2(R4),@R2   ;STORE AWAY DATA BYTE
1291  003362  000002  ;173362 000002
1292  003364  005211  ;173364 005211          INC @R1
1293  003366  120227  ;173366 120227          CMPB R2,#375     ;HAS BRANCH OFFSET BEEN STORED
1294  003370  000375  ;173370 000375
1295  003372  001366  ;173372 001366          BNE LOOP         ;NO
1296  003374  105222  ;173374 105222          INCB (R2)+       ;YES, ALL DONE
1297  003376  000142  ;173376 000142          JMP -(R2)        ;THEN TRANSFER TO RTN
1298
1299          ;THE FOLLOWING 400 LOCATIONS ARE
1300          ;A REPRODUCTION OF THE DDCMP BOOT-
1301          ;STRAP ROM. IT IS HERE FOR COM-
1302          ;PARISON TO THE ACTUAL ROM AND
1303          ;FOR REFERENCE.
1304                                   ;
1305                                   ;       COPYRIGHT  1975, DIGITAL EQUIPMENT CORP., MAYNARD,MASS. 01754
1306                                   ;
1307                                   ;       THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A
1308                                   ;       LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND
1309                                   ;       CAN BE COPIED (WITH INCLUSION OF DEC'S
1310                                   ;       COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM,
1311                                   ;       EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING
1312                                   ;       BY DEC.
1313                                   ;
1314                                   ;       THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO
1315                                   ;       CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED
1316                                   ;       AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
1317                                   ;
1318                                   ;       DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR
1319                                   ;       RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH
1320                                   ;       IS NOT SUPPLIED BY DEC.
1321                                   ;
1322                                   ;       VERSION 01
1323                                   ;
1324                                   ;       STUART WECKER  01/22/75
1325                                   ;
1326                                   ;       DIGITAL EQUIPMENT CORPORATION
1327                                   ;       COMPUTER NETWORK FACILITIES
1328                                   ;       DOWN-LINE LOADING PROGRAM
1329                                   ;
1330                                   ;       THIS PROGRAM LOADS COMPUTER MEMORY FROM DATA SENT OVER
1331                                   ;       A DATA COMMUNICATIONS LINK. IT SENDS AND RECEIVES
1332                                   ;       MESSAGES IN DDCMP BOOT FORMAT.  THE PRIMARY BOOT ONLY
```

```
1333                            ;       LOADS A SINGLE BLOCK, THE SECONDARY BOOT,  WHICH
1334                            ;       THEN REQUESTS AND LOADS THE DESIRED PROGRAM.
1335                            ;
1336                            ;       CURRENT VERSION DDCMP: 3.0 - MAY 7,1974
1337                            ;
1338                            ;       THE BOOTSTRAP MESSAGES ARE OF THE FORM:
1339                            ;
1340                            ;       SYN,SYN,DLE,CNT,F,S,FILL,FILL,ADDR,CRC1,DATA,CRC2
1341                            ;
1342                            ;       ALL ITEMS ARE 8-BITS LONG UNLESS OTHERWISE SPECIFIED
1343                            ;
1344                            ;       SYN-THE SYNC CHARACTER-SYNC-226,ASYNC-377
1345                            ;       DLE-THE BOOT HEADER CHARACTER-OCTAL 220
1346                            ;       CNT-THE 14-BIT COUNT FIELD-LENGTH OF DATA FIELD
1347                            ;       F-THE FINAL BIT-LINK CONTROL
1348                            ;       S-THE SELECT BIT-LINK CONTROL
1349                            ;       FILL-A FILL CHARACTER-OCTAL 000
1350                            ;       ADDR-THE STATION ADDR-FOR PT. TO PT.=1
1351                            ;       CRC1-THE 16-BIT CRC-16 COMPUTED ON DLE THROUGH ADDR
1352                            ;       DATA-THE BOOT DATA AS FOLLOWS:
1353                            ;               CODE,INFO
1354                            ;       ONLY THE FOLLOWING CODES ARE USED BY THE
1355                            ;       PRIMARY BOOT
1356                            ;       CODE=10  REQUEST SECONDARY PROGRAM
1357                            ;       INFO=DEVICE TYPE,STATION ADDRESS
1358                            ;       DEVICE TYPE-DP=0,DU=2,DL=4,DQ=6
1359                            ;       STATION ADDRESS=1
1360                            ;       CODE=0  PROGRAM LOAD WITH TRANSFER ADDRESS
1361                            ;       INFO=BLKNO,BLK LDADDR,IMAGE DATA,TRANS ADDR
1362                            ;       BLKNO=0
1363                            ;       BLOCK LDADDR=6
1364                            ;       TRANS ADDR=6
1365                            ;       HEADER COUNT > OR = TO 10.
1366                            ;       ADDRESSES ARE 4 BYTES-32 BITS-LOW BIT FIRST
1367                            ;       CRC2-THE 16-BIT CRC-16 COMPUTED ON THE DATA FIELD ONLY
1368                            ;
1369                            ;
1370                            ;       OPTION SWITCHES:
1371                            ;               DEVICE-DP11,DU11,DL11
1372                            ;               CRC-KG11,SCRC
1373                            ;
1374                            ;
1375                            ;
1376                            ;       REGISTER DEFINITIONS
1377                            ;
1378            000000  R0=%0                   ;BLOCK LOAD ADDR
1379            000001  R1=%1                   ;DEVICE CSR ADDRESS
1380            000002  R2=%2                   ;CRC CALC TEMP
1381            000003  R3=%3                   ;SOFTWARE CRC
1382            000004  R4=%4                   ;BLOCK CHAR COUNT
1383            000005  R5=%5                   ;CRC CALC TEMP
1384            000006  SP=%6                   ;STACK ADDR
1385            000007  PC=%7                   ;LOCATION COUNTER
1386                            ;
1387                            ;       LITERALS
1388                            ;
```

```
1389                            ;       000001  $STADR=1                ;STATION ADDR
1390                            ;       177570  $SWR=177570             ;SWITCH REGISTER ADDR
1391                            ;       000226  $SYN=226                ;SYNC CHARACTER
1392                            ;       000220  $DLE=220                ;DDCMP DLE CHARACTER
1393                            ;       000400  $STRIP=400
1394                                            ;
1395                                            ;       THE STACK IS USED AS FOLLOWS:
1396                                            ;       STACK-2:FOR JSR TO GET ROUTINE
1397                                            ;       STACK-4:TEMP FOR CRC CALCULATION
1398                                            ;
1399                                            ;       START OF BOOT PROGRAM
1400                                            ;
1401                                            ;       START1-DEVICE UNIT 0-NORMAL CONFIGURATION
1402                                            ;       START2-USE SWITCH REG AS DEVICE DISPLACEMENT
1403                                            ;               I.E. #0-0,#1-10,#2-20
1404                                            ;
1405                            ;       173400  .=173400
1406  003400  012700  ;173400 012700 START1: MOV     (PC)+,R0        ;NON ZERO VALUE TO R0
1407  003402  005000  ;173402 005000 START2: CLR     R0              ;CLEAR R0
1408  003404  000005  ;173404 000005         RESET                   ;RESET SYS,MEM MGT, ETC...
1409  003406  012706  ;173406 012706         MOV     #17776,SP       ;STACK AT 4K-2
1410  003410  017776  ;173410 017776
1411                                            ;
1412                                            ;       FIND THE DU-11 IN THE FLOATING ADDRESS SPACE
1413                                            ;
1414  003412  010702  ;173412 010702         MOV     PC,R2           ;CURRENT PC
1415  003414  062702  ;173414 062702         ADD     #DEVTAB-.,R2    ;DEVICE TABLE ADDR
1416  003416  000360  ;173416 000360
1417  003420  012703  ;173420 012703         MOV     #6,R3           ;TRAP PS ADDR
1418  003422  000006  ;173422 000006
1419  003424  005013  ;173424 005013         CLR     (R3)            ;CLEAR NEW PS
1420  003426  010243  ;173426 010243         MOV     R2,-(R3)        ;TABLE ADDR TO LOC 4
1421  003430  160313  ;173430 160313         SUB     R3,(R3)         ;SUB TO TRAP RTN
1422  003432  005303  ;173432 005303         DEC     R3              ;LEAVE CNT 3 FOR LOOP
1423  003434  012701  ;173434 012701         MOV     #160010,R1      ;START SEARCH ADDR
1424  003436  160010  ;173436 160010
1425  003440  005711  ;173440 005711 DEVLOP: TST     (R1)            ;IS DEVICE THERE
1426  003442  111204  ;173442 111204         MOVB    (R2),R4         ;DEVICE INCREMENT TO R3
1427  003444  060401  ;173444 060401         ADD     R4,R1           ;UPDATE TO NEXT DEVICE
1428  003446  005201  ;173446 005201         INC     R1              ;INCREMENT MODULO
1429  003450  040401  ;173450 040401         BIC     R4,R1           ;CLEAR EXCESS
1430  003452  005703  ;173452 005703         TST     R3              ;TEST FOR DONE
1431  003454  001371  ;173454 001371         BNE     DEVLOP          ;NOT YET
1432  003456  005700  ;173456 005700         TST     R0              ;TEST SWITCH REG USE
1433  003460  001002  ;173460 001002         BNE     SNDREQ          ;NO SWITCH REG
1434  003462  063701  ;173462 063701         ADD     @#$SWR,R1       ;ADD SWR VALUE
1435  003464  177570  ;173464 177570
1436                                          ;
1437                                          ;       SET UP DEVICE FOR OUTPUT
1438                                          ;
1439                            ;173466 SNDREQ:
1440  003466  012711  ;173466 012711         MOV     #6,(R1)         ;DATA TERM RDY AND REQ TO SEND
1441  003470  000006  ;173470 000006
1442  003472  012761  ;173472 012761         MOV     #36000+$SYN,2(R1) ;SET SYNC REGISTER
1443  003474  036226  ;173474 036226
1444  003476  000002  ;173476 000002
```

```
1445  003500  032711  ;173500 032711  L3:     BIT     #20000,(R1)      ;TEST CLEAR TO SEND
1446  003502  020000  ;173502 020000
1447  003504  001775  ;173504 001775          BEQ     L3               ;NOT YET
1448  003506  022121  ;173506 022121          CMP     (R1)+,(R1)+      ;MOVE PTR TO XMIT TSR
1449  003510  052711  ;173510 052711          BIS     #20,(R1)         ;TURN SEND ON
1450  003512  000020  ;173512 000020
1451                                  ;
1452                                  ;       SEND SECONDARY PGM REQUEST MESSAGE
1453                                  ;
1454  003514  010700  ;173514 010700          MOV     PC,R0            ;CURRENT PC
1455  003516  062700  ;173516 062700          ADD     #RQMSG-.,R0      ;REQUEST MSG ADDR
1456  003520  000230  ;173520 000230
1457  003522  012704  ;173522 012704          MOV     #RQMSGE-RQMSG,R4 ;COUNT
1458  003524  000026  ;173524 000026
1459  003526  112061  ;173526 112061  L4:     MOVB    (R0)+,2(R1)      ;CHAR TO XMIT REGISTER
1460  003530  000002  ;173530 000002
1461  003532  105711  ;173532 105711  L5:     TSTB    (R1)             ;DONE YET ?
1462  003534  100376  ;173534 100376          BPL     L5               ;NO
1463  003536  005304  ;173536 005304          DEC     R4               ;DECREMENT COUNT
1464  003540  001372  ;173540 001372          BNE     L4               ;ONCE MORE
1465  003542  042711  ;173542 042711          BIC     #20,(R1)         ;DROP SEND
1466  003544  000020  ;173544 000020
1467  003546  024141  ;173546 024141          CMP     -(R1),-(R1)      ;RESET PTR TO RCV CSR
1468                                  ;
1469                                  ;       GET SECONDARY PROGRAM
1470                                  ;
1471                  ;173550         GETPGM:
1472  003550  042711  ;173550 042711          BIC     #20,(R1)         ;CLEAR SEARCH SYNC
1473  003552  000020  ;173552 000020
1474  003554  012711  ;173554 012711          MOV     #422,(R1)        ;SET FOR CLEAR AND STRIP SYNC
1475  003556  000422  ;173556 000422
1476  003560  005003  ;173560 005003          CLR     R3               ;CLEAR CRC VALUE
1477                                  ;
1478                                  ;       GET MESSAGE HEADER AND CHECK IT
1479                                  ;
1480  003562  012700  ;173562 012700          MOV     #1,R0            ;LOAD HDR AT LOC. 1
1481  003564  000001  ;173564 000001
1482  003566  012704  ;173566 012704          MOV     #8.,R4           ;BLOCK COUNT
1483  003570  000010  ;173570 000010
1484  003572  004767  ;173572 004767          JSR     PC,GET           ;GET HEADER
1485  003574  000060  ;173574 000060
1486  003576  005703  ;173576 005703          TST     R3               ;CHECK HEADER CRC
1487  003600  001363  ;173600 001363          BNE     GETPGM           ;NO GOOD
1488  003602  123727  ;173602 123727          CMPB    @#6,#$STADR      ;CHECK FOR MY ADDR
1489  003604  000006  ;173604 000006
1490  003606  000001  ;173606 000001
1491  003610  001357  ;173610 001357          BNE     GETPGM           ;NOT MINE
1492  003612  123727  ;173612 123727          CMPB    @#1,#$DLE        ;IS THIS A DLE MSG
1493  003614  000001  ;173614 000001
1494  003616  000220  ;173616 000220
1495  003620  001322  ;173620 001322          BNE     SNDREQ           ;NO, ASK FOR ONE
1496                                  ;
1497                                  ;       GET DATA BLOCK
1498                                  ;
1499  003622  013704  ;173622 013704          MOV     @#2,R4           ;DATA FIELD LENGTH
1500  003624  000002  ;173624 000002
```

```
 1501  003626  042704  ;173626 042704          BIC     #140000,R4      ;MASK OFF S,F BITS
 1502  003630  140000  ;173630 140000
 1503  003632  122424  ;173632 122424          CMPB    (R4)+,(R4)+     ;ADD 2 FOR CRC
 1504  003634  005000  ;173634 005000          CLR     R0              ;LOAD INTO LOCATION 0
 1505  003636  004767  ;173636 004767          JSR     PC,GET1         ;GET DATA BLOCK
 1506  003640  000014  ;173640 000014
 1507  003642  005703  ;173642 005703          TST     R3              ;CHECK DATA FIELD CRC
 1508  003644  001310  ;173644 001310          BNE     SNDREQ          ;NO GOOD
 1509  003646  105713  ;173646 105713          TSTB    (R3)            ;CHECK CODE IN LOC 0
 1510  003650  001306  ;173650 001306          BNE     SNDREQ          ;NOT PROGRAM LOAD
 1511  003652  000137  ;173652 000137          JMP     @#6             ;TRANSFER TO SECONDARY PGM
 1512  003654  000006  ;173654 000006
 1513                                  ;
 1514                                  ;       GET A BLOCK AND COMPUTE CRC
 1515                                  ;
 1516                  ;173656         GET:
 1517                  ;173656         GET1:
 1518  003656  105711  ;173656 105711          TSTB    (R1)            ;IS DEVICE DONE YET
 1519  003660  100376  ;173660 100376          BPL     GET             ;NOT YET
 1520  003662  042711  ;173662 042711          BIC     #$STRIP,(R1)    ;NO STRIP SYNC
 1521  003664  000400  ;173664 000400
 1522  003666  116110  ;173666 116110          MOVB    2(R1),(R0)      ;STORE IT
 1523  003670  000002  ;173670 000002
 1524                                  ;
 1525                                  ;       CRC CALCULATION ROUTINE
 1526                                  ;
 1527                  ;       120001  POLY=120001                     ;CRC-16 POLYNOMIAL
 1528                                  ;
 1529  003672  012705  ;173672 012705          MOV     #8.,R5          ;BYTE LENGTH
 1530  003674  000010  ;173674 000010
 1531  003676  112002  ;173676 112002          MOVB    (R0)+,R2        ;CHARACTER TO ADD TO CRC
 1532  003700  000241  ;173700 000241  CRCLOP: CLC                     ;CLEAR CARRY
 1533  003702  006003  ;173702 006003          ROR     R3              ;SHIFT OLD PARTIAL
 1534  003704  103003  ;173704 103003          BCC     L10             ;IF CLEAR CHECK CHAR
 1535  003706  006002  ;173706 006002          ROR     R2              ;SHIFT CHARACTER
 1536  003710  103003  ;173710 103003          BCC     L11             ;XOR POLY
 1537  003712  000410  ;173712 000410          BR      L12             ;NEXT BIT
 1538  003714  006002  ;173714 006002  L10:    ROR     R2              ;SHIFT CHARACTER
 1539  003716  103006  ;173716 103006          BCC     L12             ;NEXT BIT
 1540  003720  012746  ;173720 012746  L11:    MOV     #POLY,-(SP)     ;POLY TO STACK
 1541  003722  120001  ;173722 120001
 1542  003724  040316  ;173724 040316          BIC     R3,(SP)         ;NOT PARTIAL AND POLY
 1543  003726  042703  ;173726 042703          BIC     #POLY,R3        ;NOT POLY AND PARTIAL
 1544  003730  120001  ;173730 120001
 1545  003732  052603  ;173732 052603          BIS     (SP)+,R3        ;POLY XOR PARTIAL
 1546  003734  005305  ;173734 005305  L12:    DEC     R5              ;DECREMENT BIT COUNT
 1547  003736  001360  ;173736 001360          BNE     CRCLOP          ;ONCE MORE
 1548  003740  005304  ;173740 005304          DEC     R4              ;DECREMENT COUNT
 1549  003742  001345  ;173742 001345          BNE     GET             ;ONCE MORE
 1550  003744  000207  ;173744 000207          RTS     PC              ;RETURN
 1551                                  ;
 1552                                  ;       SECONDARY PROGRAM REQEST MSG
 1553                                  ;
 1554  003746  113226  ;173746 113226  RQMSG:  .BYTE   $SYN,$SYN,$SYN,$SYN
 1555  003750  113226  ;173750 113226
 1556  003752  002220  ;173752 002220          .BYTE   $DLE,4,0,0,0,1
```

```
1557  003754  000000  ;173754 000000
1558  003756  000400  ;173756 000400
1559  003760  050055  ;173760 050055          .BYTE   55,120
1560  003762  001010  ;173762 001010          .BYTE   10              ;REQ SEC PGM CODE
1561                  ;                       .BYTE   2               ;DEVICE CODE
1562  003764  000001  ;173764 000001          .BYTE   $STADR          ;STATION ADDR
1563                  ;                       .BYTE   0               ;FILL
1564  003766  030242  ;173766 030242          .BYTE   242,60          ;FOR STADR=1
1565                                  ;       NOTE:NODEV AND DEVTAB MUST BE IN THIS ORDER
1566                                  ;       DO NOT SEPARATE THEM
1567                                  ;
1568                                  ;       .EVEN
1569  003770  122243  ;173770 122243  NODEV:  CMPB    (R2)+,-(R3)     ;INC PTR-DEC CNT
1570  003772  000002  ;173772 000002          RTI                     ;RETURN FROM TRAP
1571                                  ;
1572                  ;173774         RQMSGE:                         ;END OF MSG-USE JUNK AS PADS
1573  003774  007407  ;173774 007407  DEVTAB: .BYTE   7               ;DJ-11
1574                  ;                       .BYTE   17              ;DH-11
1575  003776  END.YC:
1576  003776  003407  ;173776 003407          .BYTE   7               ;DQ-11
1577                  ;                       .BYTE   7               ;DU-11
1578                  ;174000         END:
1579                  ;       173400          .END    START1
```

```
1580  004000  MAP.YD:
1581          ;THE FOLLOWING IS A REPRODUCTION
1582          ;OF THE ROM PROGRAM FOR BM873YD.
1583          ;IT IS HERE FOR COMPARISON TO THE
1584          ;ACTUAL ROM AND FOR REFERENCE
1585          ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
1586          ;BM873-YD.P11
1587
1588                                  ;       THIS CODE IS TO BE BLASTED INTO PROMS ON THE BM873-YD BOARD.
1589                                  ;       WRITTEN BY DAVID M. ROSENBERG           OCTOBER 1974
1590                                  ;REGISTER DEFINITIONS
1591
1592          ;       000000  R0=%0           ;GENERAL PURPOSE REGISTER 0
1593          ;       000001  R1=%1           ;GENERAL PURPOSE REGISTER 1
1594          ;       000002  R2=%2           ;GENERAL PURPOSE REGISTER 2
1595          ;       000003  R3=%3           ;GENERAL PURPOSE REGISTER 3
1596          ;       000004  R4=%4           ;GENERAL PURPOSE REGISTER 4
1597          ;       000005  R5=%5           ;GENERAL PURPOSE REGISTER 5
1598          ;       000006  SP=%6           ;STACK POINTER (REGISTER R6)
1599          ;       000007  PC=%7           ;PROGRAM COUNTER (REGISTER R7)
1600
1601
1602                                  ;SYMBOL DEFINITIONS
1603
1604          ;       177776  PS=177776       ;PROCESSOR STATUS REGISTER
1605          ;       177570  SWR=177570      ;FRONT PANEL SWITCH REGISTER
1606          ;       000000  PR0=0*40        ;PRIORITY LEVEL 0
1607          ;       000040  PR1=1*40        ;PRIORITY LEVEL 1
1608          ;       000100  PR2=2*40        ;PRIORITY LEVEL 2
1609          ;       000140  PR3=3*40        ;PRIORITY LEVEL 3
1610          ;       000200  PR4=4*40        ;PRIORITY LEVEL 4
1611          ;       000240  PR5=5*40        ;PRIORITY LEVEL 5
1612          ;       000300  PR6=6*40        ;PRIORITY LEVEL 6
1613          ;       000340  PR7=7*40        ;PRIORITY LEVEL 7
1614          ;       000001  BIT0=000001
1615          ;       000002  BIT1=000002
1616          ;       000004  BIT2=000004
1617          ;       000010  BIT3=000010
1618          ;       000020  BIT4=000020
1619          ;       000040  BIT5=000040
1620          ;       000100  BIT6=000100
1621          ;       000200  BIT7=000200
1622          ;       000400  BIT8=000400
1623          ;       001000  BIT9=001000
1624          ;       002000  BIT10=002000
1625          ;       004000  BIT11=004000
1626          ;       010000  BIT12=010000
1627          ;       020000  BIT13=020000
1628          ;       040000  BIT14=040000
1629          ;       100000  BIT15=100000
1630
```

J03


```
1631            ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 3
1632            ;BM873-YD.P11   BUTTON #1  -  BOOTSTRAP USING THE PDP-11 SWITCH REGISTER
1633
1634
1635            ;       173000  ROMORG  =       173000          ;SET ROM ORIGIN TO 773000
1636            ;       173000          .=ROMORG                ;BM873-YD OCCUPIES 773000-773777
1637
1638  004000  033727 ;173000  033727  BUTON1: BIT     @#SWR,#BIT0     ;IS RIGHTMOST BIT ON?
1639  004002  177570 ;173002  177570
1640  004004  000001 ;173004  000001
1641  004006  001010 ;173006  001010          BNE     LOWBIT          ;IF THE BIT IS ON, BRANCH
1642  004010  013707 ;173010  013707          MOV     @#SWR,PC        ;JUMP TO THE ADDRESS IN THE SWITCH REGISTER
1643  004012  177570 ;173012  177570
1644                                                                  ;WITHOUT HAVING TOUCHED ANY OF R0 - R6
1645
1646  004014  111704 ;173014  111704  BUTON3: MOVB    (PC),R4         ;R4 = 1 INDICATES THAT BUTTON #3 WAS PRESSED
1647  004016  005001 ;173016  005001          CLR     R1              ;SET UNIT NUMBER TO ZERO
1648  004020  005005 ;173020  005005          CLR     R5              ;CLEAR "LOGICAL SWITCH REGISTER"
1649  004022  000424 ;173022  000424          BR      TCBOOT          ;DO A DEFAULT BOOT STRAP FROM DECTAPE
1650
1651  004024  173000 ;173024  173000          .WORD   ROMORG,PR7
1652  004026  000340 ;173026  000340
1653
1654  004030  013701 ;173030  013701  LOWBIT: MOV     @#SWR,R1        ;R1 IS A COPY OF THE SWITCH REGISTER
1655  004032  177570 ;173032  177570
1656  004034  106301 ;173034  106301          ASLB    R1              ;LEFT-ALIGN SPEED FIELD IN RIGHT BYTE
1657  004036  122701 ;173036  122701          CMPB    #16*20,R1       ;IS THE SPEED 16 OR 17?
1658  004040  000340 ;173040  000340
1659  004042  101404 ;173042  101404          BLOS    UNITNO          ;IF SPEED IS 16 OR 17, BRANCH
1660  004044  122701 ;173044  122701          CMPB    #3*20,R1        ;IS THE SPEED 0, 1, OR 2?
1661  004046  000060 ;173046  000060
1662  004050  101001 ;173050  101001          BHI     UNITNO          ;IF THE SPEED IS 0, 1, OR 2, BRANCH
1663  004052  005001 ;173052  005001          CLR     R1              ;SPEED WAS 3-15; SET UNIT NUMBER = 0
1664  004054  000301 ;173054  000301  UNITNO: SWAB    R1              ;MOVE UNIT NUMBER TO BITS 0-2
1665
1666                                          ; IT IS POSSIBLE TO MANUALLY SET THE DESIRED BOOTSTRAP UNIT NUMBER
1667                                          ; INTO THE RIGHTMOST THREE BITS OF R1, SET THE PDP-11 FRONT PANEL
1668                                          ; SWITCH REGISTER, AND THEN JUMP INTO THE ROM CODE AT THIS POINT.
1669
1670  004056  042701 ;173056  042701          BIC     #^C7,R1         ;ISOLATE UNIT NUMBER IN R1
1671  004060  177770 ;173060  177770
1672  004062  013705 ;173062  013705          MOV     @#SWR,R5        ;R5 IS NOW THE "LOGICAL SWITCH REGISTER"
1673  004064  177570 ;173064  177570
1674  004066  005004 ;173066  005004          CLR     R4              ;R4 = 0 INDICATES THAT BUTTON #1 WAS PRESSED
1675  004070  105705 ;173070  105705          TSTB    R5              ;SHOULD WE BOOT FROM DECTAPE OR RH11/RP04?
1676  004072  100507 ;173072  100507          BMI     RPBOOT          ;IF BIT 7 WAS ONE, BRANCH OFF TO THE RH11/RP04
1677                                                                  ;OTHERWISE, FALL THROUGH TO THE DECTAPE
```

```
1678            ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 4
1679            ;BM873-YD.P11   DECTAPE BOOTSTRAP AND DUMP ROUTINES
1680
1681
1682            ;       177344  TCWC    =       177344          ;TC11 DECTAPE WORD COUNT REGISTER
1683            ;       000001  TCGO    =       1               ;TC11 "GO" BIT
1684            ;       000002  TCRNUM  =       1*2             ;TC11 "READ BLOCK NUMBER" FUNCTION
1685            ;       000004  TCREAD  =       2*2             ;TC11 "READ DATA" FUNCTION
1686            ;       000014  TCWRIT  =       6*2             ;TC11 "WRITE DATA" FUNCTION
1687            ;       004000  TCREV   =       4000            ;MOVE DECTAPE IN REVERSE DIRECTION
1688
1689                                            ; BOOTSTRAP (FROM DECTAPE) PARAMETERS
1690            ;       000400  TCBWDC  =       ^D256           ;WORD COUNT FOR THE SECONDARY BOOTSTRAP
1691            ;       000000  TCBEND  =       0               ;WHICH END OF THE DECTAPE (0 = FRONT; 1 = BACK)
1692
1693                                            ; DUMP (TO DECTAPE) PARAMETERS
1694            ;       070000  TCDWDC  =       ^D28672         ;WORD COUNT FOR THE CORE DUMP TO DECTAPE
1695            ;       000001  TCDEND  =       1               ;WHICH END OF THE DECTAPE (0 = FRONT; 1 = BACK)
1696
1697                                            ; GENERAL (BOOTSTRAP AND DUMP) DECTAPE PARAMETER
1698            ;       000024  TCRTRY  =       ^D20            ;NUMBER OF RETRIES IN CASE OF ERROR
1699
1700  004074  012700 ;173074  012700  TCBOOT: MOV     #<TCBEND*TCREV>!TCREAD!TCGO,R0  ;SET UP DATA-TRANSFER COMMAND
1701  004076  000005 ;173076  000005
1702  004100  012702 ;173100  012702          MOV     #-TCBWDC,R2     ;SET WORD COUNT TO 256 (512 BYTES)
1703  004102  177400 ;173102  177400
1704  004104  012703 ;173104  012703          MOV     #<<1-TCBEND>*TCREV>!TCRNUM!TCGO,R3      ;SET UP POSITION COMMAND
1705  004106  004003 ;173106  004003
1706  004110  000301 ;173110  000301          SWAB    R1              ;BRING UNIT NUMBER INTO THE LEFT BYTE
1707  004112  050103 ;173112  050103          BIS     R1,R3           ;PUT UNIT NUMBER INTO POSITIONING COMMAND
1708  004114  050100 ;173114  050100          BIS     R1,R0           ;PUT UNIT NUMBER INTO DATA-TRANSFER COMMAND
1709  004116  012701 ;173116  012701  TCSTRT: MOV     #TCWC,R1        ;R1 NOW POINTS TO TC11 WORD COUNT REGISTER
1710  004120  177344 ;173120  177344
1711  004122  012706 ;173122  012706  TCLOOP: MOV     #TCRTRY,SP      ;INITIALIZE RETRY COUNT IN SP
1712  004124  000024 ;173124  000024
1713  004126  005705 ;173126  005705  TCBGIN: TST     R5              ;TEST "INDEFINITE RETRY" BIT
1714  004130  100404 ;173130  100404          BMI     TCRSET          ;BRANCH IF "INDEFINITE RETRY" IS ENABLED
1715  004132  005306 ;173132  005306          DEC     SP              ;DECREMENT RETRY COUNT
1716  004134  100002 ;173134  100002          BPL     TCRSET          ;BRANCH IF RETRY COUNT NOT EXHAUSTED
1717  004136  000000 ;173136  000000  TCHALT: HALT                    ;RETRY COUNT IS EXHAUSTED FOR DECTAPE OPERATION
1718  004140  000770 ;173140  000770          BR      TCLOOP          ;HE PRESSED "CONTINUE", SO TRY AGAIN
1719  004142  000005 ;173142  000005  TCRSET: RESET                   ;STOP ANYTHING IN PROGRESS, FOR NEXT TRY
1720  004144  010341 ;173144  010341          MOV     R3,-(R1)        ;INITIATE DECTAPE POSIONING OPERATION
1721  004146  005711 ;173146  005711  TCWAIT: TST     (R1)            ;TEST FOR AN "ERROR"
1722  004150  100376 ;173150  100376          BPL     TCWAIT          ;LOOP UNTIL AN "ERROR" IS DETECTED
1723  004152  005721 ;173152  005721          TST     (R1)+           ;MAKE R1 POINT TO THE WORD COUNT REGISTER
1724  004154  005761 ;173154  005761          TST     -4(R1)          ;IS THE ERROR "ENDZONE"?
1725  004156  177774 ;173156  177774
1726  004160  100362 ;173160  100362          BPL     TCBGIN          ;IF NOT, BRANCH BACK TO TRY AGAIN
1727  004162  010211 ;173162  010211          MOV     R2,(R1)         ;SET UP WORD COUNT FOR DATA-TRANSFER
1728  004164  010041 ;173164  010041          MOV     R0,-(R1)        ;INITIATE THE DATA-TRANSFER OPERATION
1729  004166  105711 ;173166  105711  TCDONE: TSTB    (R1)            ;TEST FOR "DONE"
1730  004170  100376 ;173170  100376          BPL     TCDONE          ;LOOP UNTIL THE "DONE" BIT SETS
1731  004172  005721 ;173172  005721          TST     (R1)+           ;WAS AN "ERROR" DETECTED?
1732  004174  100754 ;173174  100754          BMI     TCBGIN          ;IF SO, BRANCH BACK AND TRY AGAIN
1733  004176  005741 ;173176  005741          TST     -(R1)           ;MAKE R1 POINT TO THE COMMAND REGISTER
```

```
1734  004200  105011 ;173200  105011          CLRB    (R1)                ;STOP ALL DECTAPE MOTION
1735  004202  122700 ;173202  122700          CMPB    #TCREAD!TCGO,RD     ;WAS THIS A "NORMAL READ" OPERATION?
1736  004204  000005 ;173204  000005
1737  004206  001001 ;173206  001001          BNE     TCSTOP              ;IF NOT, GO STOP
1738  004210  000137 ;173210  000137  GOTOO:  JMP     @(PC)+              ;JUMP TO PDP-11 LOCATION ZERO
1739  004212  000000 ;173212  000000  TCSTOP: HALT                        ;SUCCESSFUL COMPLETION OF A "NON-READ" OPERATION
1740  004214  000776 ;173214  000776          BR      TCSTOP              ;SO THAT PRESSING "CONTINUE" WON'T GO ANYWHERE
```

```
1741            ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 5
1742            ;BM873-YD.P11   DECTAPE BOOTSTRAP AND DUMP ROUTINES
1743
1744  004216  010037 ;173216  010037  TCDUMP: MOV     R0,@#ROTOR7     ;SAVE R0 IN PDP-11 MEMORY LOCATION 40
1745  004220  000040 ;173220  000040
1746  004222  000402 ;173222  000402          BR      TCCONT          ;BRANCH AROUND REQUIRED INTERRUPT VECTOR
1747
1748  004224  173000 ;173224  173000          .WORD   ROMORG,PR7
1749  004226  000340 ;173226  000340
1750
1751  004230  010700 ;173230  010700  TCCONT: MOV     PC,R0           ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
1752  004232  000410 ;173232  000410          BR      REGSAV          ;GO TO THE "REGISTER SAVING" SUBROUTINE
1753  004234  012700 ;173234  012700          MOV     #<TCDEND*TCREV>!TCWRIT!TCGO,R0  ;SET UP (WRITE) TRANSFER COMMAND
1754  004236  004015 ;173236  004015
1755  004240  012702 ;173240  012702          MOV     #-TCDWDC,R2     ;SET WORD-COUNT TO 28K WORDS
1756  004242  110000 ;173242  110000
1757  004244  012703 ;173244  012703          MOV     #<<1-TCDEND>*TCREV>!TCRNUM!TCGO,R3      ;SET UP POSITION COMMAND
1758  004246  000003 ;173246  000003
1759  004250  005005 ;173250  005005          CLR     R5              ;CLEAR "INDEFINITE RETRY" BIT
1760  004252  000721 ;173252  000721          BR      TCSTRT          ;BRANCH INTO DECTAPE ROUTINE
1761
1762
1763
1764
1765                                    ; THE FOLLOWING SUBROUTINE IS USED TO SAVE THE PDP-11 GENERAL REGISTERS
1766                                    ; IN PDP-11 MEMORY LOCATIONS 40-57.
1767
1768                                    ; THE CALLING SEQUENCE IS AS FOLLOWS:           MOV     R0,@#ROTOR7
1769                                    ;                                               MOV     PC,R0
1770                                    ;                                               BR      REGSAV
1771                                    ;                                               RETURN HERE
1772
1773  004254  010137 ;173254  010137  REGSAV: MOV     R1,@#ROTOR7+2   ;SAVE R1 IN MEMOTY LOCATION 42
1774  004256  000042 ;173256  000042
1775  004260  012701 ;173260  012701          MOV     #ROTOR7+4,R1    ;R1 NOW POINTS TO MEMORY LOCATION 44
1776  004262  000044 ;173262  000044
1777  004264  010221 ;173264  010221          MOV     R2,(R1)+        ;SAVE R2 IN MEMORY LOCATION 44
1778  004266  010321 ;173266  010321          MOV     R3,(R1)+        ;SAVE R3 IN MEMORY LOCATION 46
1779  004270  010421 ;173270  010421          MOV     R4,(R1)+        ;SAVE R4 IN MEMORY LOCATION 50
1780  004272  010521 ;173272  010521          MOV     R5,(R1)+        ;SAVE R5 IN MEMORY LOCATION 52
1781  004274  010621 ;173274  010621          MOV     SP,(R1)+        ;SAVE SP IN MEMORY LOCATION 54
1782  004276  010021 ;173276  010021          MOV     R0,(R1)+        ;SAVE PC IN MEMORY LOCATION 56
1783  004300  000160 ;173300  000160          JMP     2(R0)           ;RETURN TO THE CALLING ROUTINE
1784  004302  000002 ;173302  000002
1785
```

```
1786                    ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 6
1787                    ;BM873-YD.P11   RH11/RP04 BOOTSTRAP AND DUMP ROUTINES
1788
1789
1790                    ;       176700  RPCS1   =       176700  ;ADDRESS OF RH11/RP04 CONTROL & STATUS REGISTER 1
1791                    ;       000002  RPWC    =       2       ;OFFSET TO RH11/RP04 WORD COUNT REGISTER
1792                    ;       000006  RPDA    =       6       ;OFFSET TO RH11/RP04 TRACK & SECTOR ADDRESS REGISTER
1793                    ;       000010  RPCS2   =       10      ;OFFSET TO RH11/RP04 CONTROL & STATUS REGISTER 2
1794                    ;       000012  RPDS    =       12      ;OFFSET TO RH11/RP04 DRIVE STATUS REGISTER
1795                    ;       000032  RPOF    =       32      ;OFFSET TO RH11/RP04 OFFSET REGISTER (CONTAINING FMT22)
1796                    ;       000034  RPDC    =       34      ;OFFSET TO RH11/RP04 DESIRED CYLINDER REGISTER
1797
1798                    ;       040000  RPTRE   =       BIT14   ;"TRANSFER ERROR" BIT IN RPCS1
1799                    ;       020000  RPMCPE  =       BIT13   ;"MASSBUS CONTROL BUS PARITY ERROR" BIT IN RPCS1
1800                    ;       004000  RPDVA   =       BIT11   ;"DRIVE AVAILABLE" BIT IN RPCS1
1801                    ;       100000  RPATA   =       BIT15   ;"ATTENTION ACTIVE" BIT IN RPDS
1802                    ;       040000  RPERR   =       BIT14   ;"COMPOSITE ERROR" BIT IN RPDS
1803                    ;       010000  RPFMT   =       BIT12   ;"FMT22" (16-BIT WORDS) BIT IN RPOF
1804
1805                    ;       000021  RPPRST  =       21      ;READ-IN PRESET
1806                    ;       000061  RPWRIT  =       61      ;WRITE DATA
1807                    ;       000071  RPREAD  =       71      ;READ DATA
1808
1809                    ;       000000  RPBFMT  =       0       ;BOOTSTRAP FORMAT (0 = 18-BIT WORDS; 2 = 16-BIT WORDS)
1810                    ;       000400  RPBWDC  =       ^D256   ;WORD COUNT FOR THE SECONDARY BOOTSTRAP FROM THE RP04
1811                    ;       000626  RPBCYL  =       ^D406   ;BOOTSTRAP CYLINDER NUMBER
1812                    ;       000000  RPBTRK  =       0       ;BOOTSTRAP TRACK NUMBER
1813                    ;       000000  RPBSCT  =       0       ;BOOTSTRAP SECTOR NUMBER
1814
1815                    ;       000000  RPDFMT  =       0       ;DUMP FORMAT (0 = 18-BIT WORDS; 2 = 16-BIT WORDS)
1816                    ;       070000  RPDWDC  =       ^D28672 ;WORD COUNT FOR THE CORE DUMP TO THE RP04
1817                    ;       000631  RPDCYL  =       ^D409   ;DUMP CYLINDER NUMBER
1818                                                    ; THE FOLLOWING TWO ASSIGNMENTS PUT THE DUMP AT THE VERY END OF THE CYLINDER
1819                    ;       000015  RPDTRK  =       ^D18-<<RPDWDC-1>/<<^D20+RPDFMT>*^D256>>  ;DUMP TRACK NUMBER
1820                    ;       000010  RPDSCT  =       ^D19+RPDFMT-<<<RPDWDC-1>/^D256>-<<^D18-RPDTRK>*<^D20+RPDFMT>>>
1821
1822
1823
1824  004304  111704 ;173304  111704  BUTON2: MOVB    (PC),R4         ;R4 = 5 INDICATES THAT BUTTON #2 WAS PRESSED
1825  004306  005005 ;173306  005005          CLR     R5              ;CLEAR "LOGICAL SWITCH REGISTER"
1826  004310  005001 ;173310  005001          CLR     R1              ;SET UNIT NUMBER TO ZERO
1827
1828  004312  012700 ;173312  012700  RPBOOT: MOV     #<RPREAD*400>!<RPBSCT*10>,R0
1829  004314  034400 ;173314  034400
1830  004316  012702 ;173316  012702          MOV     #-RPBWDC,R2
1831  004320  177400 ;173320  177400
1832  004322  012703 ;173322  012703          MOV     #<RPBFMT*40000>!<RPBTRK*2000>!RPBCYL,R3
1833  004324  000626 ;173324  000626
1834  004326  050100 ;173326  050100          BIS     R1,R0           ;PUT THE UNIT NUMBER INTO R0
1835  004330  012701 ;173330  012701  RPSTRT: MOV     #RPCS1,R1       ;SET R1 TO THE LOWEST ADDRESS USED BY THE RH11
1836  004332  176700 ;173332  176700
```

```
1837                    ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 7
1838                    ;BM873-YD.P11   RH11/RP04 BOOTSTRAP AND DUMP ROUTINES
1839
1840  004334  000005 ;173334  000005  RPLOOP: RESET                           ;RESET IN CASE OF RETRY
1841  004336  010006 ;173336  010006          MOV     R0,SP                   ;GET THE UNIT NUMBER INTO SP
1842  004340  042706 ;173340  042706          BIC     #^C7,SP                 ;ISOLATE THE UNIT NUMBER
1843  004342  177770 ;173342  177770
1844  004344  010661 ;173344  010661          MOV     SP,RPCS2(R1)            ;TELL THE RH11 THE UNIT NUMBER
1845  004346  000010 ;173346  000010
1846  004350  032711 ;173350  032711          BIT     #RPDVA,(R1)             ;TRY TO SEIZE THIS RP04 UNIT
1847  004352  004000 ;173352  004000
1848  004354  001767 ;173354  001767          BEQ     RPLOOP                  ;BRANCH IF WE HAVEN'T SEIZED IT
1849  004356  012721 ;173356  012721          MOV     #RPPRST,(R1)+           ;DO A "READ-IN PRESET" FUNCTION
1850  004360  000021 ;173360  000021
1851  004362  010306 ;173362  010306          MOV     R3,SP                   ;GET THE CYLINDER NUMBER INTO SP
1852  004364  042706 ;173364  042706          BIC     #^C1777,SP              ;ISOLATE THE CYLINDER NUMBER
1853  004366  176000 ;173366  176000
1854  004370  010661 ;173370  010661          MOV     SP,RPDC-2(R1)           ;TELL THE RP04 THE CYLINDER NUMBER
1855  004372  000032 ;173372  000032
1856  004374  010306 ;173374  010306          MOV     R3,SP                   ;GET THE FORMAT BIT AND TRACK NUMBER INTO SP
1857  004376  100003 ;173376  100003          BPL     RPCONT                  ;BRANCH IF 20 SECTOR (18-BIT WORDS) FORMAT
1858  004400  012761 ;173400  012761          MOV     #RPFMT,RPOF-2(R1) ;ESTABLISH 22 SECTOR (16-BIT WORDS) FORMAT
1859  004402  010000 ;173402  010000
1860  004404  000030 ;173404  000030
1861  004406  006206 ;173406  006206  RPCONT: ASR     SP                      ;RIGHT ALIGN THE TRACK
1862  004410  006206 ;173410  006206          ASR     SP                      ;                       NUMBER IN THE LEFT BYTE
1863  004412  105006 ;173412  105006          CLRB    SP                      ;CLEAR THE RIGHT BYTE
1864  004414  150006 ;173414  150006          BISB    R0,SP                   ;PUT THE SECTOR NUMBER INTO THE RIGHT BYTE
1865  004416  106006 ;173416  106006          RORB    SP                      ;RIGHT ALIGN THE
1866  004420  106206 ;173420  106206          ASRB    SP                      ;                       SECTOR NUMBER IN
1867  004422  106206 ;173422  106206          ASRB    SP                      ;                                       THE RIGHT BYTE
1868  004424  010661 ;173424  010661          MOV     SP,RPDA-2(R1)           ;TELL THE RH11 THE TRACK AND SECTOR NUMBERS
1869  004426  000004 ;173426  000004
1870  004430  010211 ;173430  010211          MOV     R2,(R1)                 ;TELL THE RH11 THE WORD COUNT
1871  004432  010006 ;173432  010006          MOV     R0,SP                   ;GET THE FUNCTION CODE INTO SP
1872  004434  105006 ;173434  105006          CLRB    SP                      ;CLEAR THE RIGHT BYTE
1873  004436  000306 ;173436  000306          SWAB    SP                      ;RIGHT ALIGN THE FUNCTION CODE
1874  004440  010641 ;173440  010641          MOV     SP,-(R1)                ;TELL THE RP04 THE FUNCTION CODE
1875  004442  105711 ;173442  105711  RPDONE: TSTB    (R1)                    ;TEST FOR RH11 "READY"
1876  004444  100376 ;173444  100376          BPL     RPDONE                  ;LOOP, WAITING FOR RH11 "READY"
1877  004446  032711 ;173446  032711          BIT     #RPTRE!RPMCPE,(R1) ;TEST FOR RH11 ERROR BITS
1878  004450  060000 ;173450  060000
1879  004452  001330 ;173452  001330          BNE     RPLOOP                  ;IF ERROR, BRANCH BACK FOR RETRY
1880  004454  032761 ;173454  032761          BIT     #RPATA!RPERR,RPDS(R1) ;TEST FOR RP04 ERROR BITS
1881  004456  140000 ;173456  140000
1882  004460  000012 ;173460  000012
1883  004462  001324 ;173462  001324          BNE     RPLOOP                  ;IF ERROR, BRANCH BACK FOR RETRY
1884  004464  022706 ;173464  022706          CMP     #RPREAD,SP              ;WAS THE FUNCTION A "NORMAL READ"?
1885  004466  000071 ;173466  000071
1886  004470  001250 ;173470  001250          BNE     TCSTOP                  ;IF NOT, BRANCH TO A HALT INSTRUCTION
1887  004472  022737 ;173472  022737          CMP     #000240,@#0             ;WAS "000240" READ INTO LOCATION ZERO?
1888  004474  000240 ;173474  000240
1889  004476  000000 ;173476  000000
1890  004500  001643 ;173500  001643          BEQ     GOTOO                   ;IF SO, BRANCH TO LOCATION ZERO
1891  004502  000000 ;173502  000000          HALT                            ;"000240" WAS NOT READ INTO LOCATION ZERO
1892  004504  000641 ;173504  000641          BR      GOTOO                   ;BRANCH TO LOCATION ZERO
```

```
 1893
 1894
 1895  004506  010037 ;173506  010037  RPDUMP: MOV     R0,@#ROTOR7     ;SAVE R0 IN PDP-11 MEMORY LOCATION "ROTOR7"
 1896  004510  000040 ;173510  000040
 1897  004512  010700 ;173512  010700          MOV     PC,R0           ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
 1898  004514  000657 ;173514  000657          BR      REGSAV          ;GO TO THE "REGISTER SAVING" SUBROUTINE
 1899  004516  012700 ;173516  012700          MOV     #<RPWRIT*400>!<RPDSCT*10>,R0
 1900  004520  030500 ;173520  030500
 1901  004522  012702 ;173522  012702          MOV     #-RPDWDC,R2
 1902  004524  110000 ;173524  110000
 1903  004526  012703 ;173526  012703          MOV     #<RPDFMT*40000>!<RPDTRK*2000>!RPDCYL,R3
 1904  004530  032631 ;173530  032631
 1905  004532  000676 ;173532  000676          BR      RPSTRT
 1906
```

```
1907            ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(  )  18-DEC-74  11:59  PAGE 8
1908            ;BM873-YD.P11   DTE20 DEVICE REGISTER AND BIT DEFINITIONS
1909
1910
1911            ;       174400  DTEBAS=174400           ;BASE OF (FIRST) DTE20 DEVICE REGISTER BLOCK
1912            ;       000040  DTESIZ=000040           ;SPACING BETWEEN CONSECUTIVE DTE20'S
1913            ;       000004  DTEMAX=4                ;MAXIMUM NUMBER OF DTE20'S ON ONE PDP-11
1914
1915
1916                                                    ;OFFSETS FROM THE BASE OF THE DTE20 DEVICE REGISTER BLOCK
1917                                                    ;TO SPECIFIC 10/11 INTERFACE RAM LOCATIONS AND REGISTERS.
1918
1919                                            ; THE FIRST 12 REGISTERS ARE NOT INITIALIZED BY "INIT" (BECAUSE THEY ARE IN RAMS
1920            ;       000000  DLYCNT=00               ;DELAY COUNT                        (ADDRESS XXXX00)
1921            ;       000002  DEXWD3=02               ;DEPOSIT OR EXAMINE WORD 3          (ADDRESS XXXX02)
1922            ;       000004  DEXWD2=04               ;DEPOSIT OR EXAMINE WORD 2          (ADDRESS XXXX04)
1923            ;       000006  DEXWD1=06               ;DEPOSIT OR EXAMINE WORD 1          (ADDRESS XXXX06)
1924            ;       000010  TENAD1=10               ;10 ADDRESS WORD 1 FOR DEX          (ADDRESS XXXX10)
1925            ;       000012  TENAD2=12               ;10 ADDRESS WORD 2 FOR DEX          (ADDRESS XXXX12)
1926            ;       000014  TO10BC=14               ;TO10 BYTE COUNT                    (ADDRESS XXXX14)
1927            ;       000016  TO11BC=16               ;TO11 BYTE COUNT                    (ADDRESS XXXX16)
1928            ;       000020  TO10AD=20               ;TO10 PDP11 MEMORY ADDRESS          (ADDRESS XXXX20)
1929            ;       000022  TO11AD=22               ;TO11 PDP11 MEMORY ADDRESS          (ADDRESS XXXX22)
1930            ;       000024  TO10DT=24               ;TO10 PDP11 DATA WORD               (ADDRESS XXXX24)
1931            ;       000026  TO11DT=26               ;TO11 PDP11 DATA WORD               (ADDRESS XXXX26)
1932
1933                                            ; THE LAST 4 REGISTERS ARE INITIALIZED BY "INIT" (BECAUSE THEY ARE IN FLIP-FLOPS
1934            ;       000030  DIAG1=30                ;DIAGNOSTIC WORD 1                  (ADDRESS XXXX30)
1935            ;       000032  DIAG2=32                ;DIAGNOSTIC WORD 2                  (ADDRESS XXXX32)
1936            ;       000034  STATUS=34               ;10/11 INTERFACE STATUS WORD        (ADDRESS XXXX34)
1937            ;       000036  DIAG3=36                ;DIAGNOSTIC WORD 3                  (ADDRESS XXXX36)
1938
1939
1940                                            ;  THE FOLLOWING ARE THE ADDRESSES OF THE DTE20 INTERRUPT VECTORS
1941
1942            ;       000774  DTEIV1=774              ;INTERRUPT VECTOR FOR DTE20 #1
1943            ;       000770  DTEIV2=770              ;INTERRUPT VECTOR FOR DTE20 #2
1944            ;       000764  DTEIV3=764              ;INTERRUPT VECTOR FOR DTE20 #3
1945            ;       000760  DTEIV4=760              ;INTERRUPT VECTOR FOR DTE20 #4
1946
1947
1948                                            ;  BIT ASSIGNMENTS FOR VARIOUS DTE20 REGISTERS USED BY THIS ROM CODE
1949
1950
1951                                                    ;BIT ASSIGNMENTS FOR TO10BC
1952
1953            ;       100000  INT11=BIT15             ;SET DONE AND INTERRUPT BOTH 10 AND 11
1954
1955                                                    ;BIT ASSIGNMENTS FOR TO11BC
1956
1957            ;       100000  INT10=BIT15             ;SET DONE AND INTERRUPT BOTH 10 AND 11
1958            ;       040000  ZSTOP=BIT14             ;STOP ON NULL (ZERO) CHARACTER
1959            ;       020000  TO11BM=BIT13            ;BYTE SIZE FOR TO-11 BYTE TRANSFERS
1960
```

```
                ;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  CDD=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  TO11ER=BIT1             ;TO 11 ERROR (NPR TIMEOUT OR BUS ERROR)
;       000001  INTSON=BIT0             ;DTE20 INTERRUPTS (TO THE -11) ARE ENABLED
```

```
2014          :BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
2015          :BM873-YD.P11   PROCEDURE BY WHICH THE PDP-10 BOOTSTRAPS AND/OR DUMPS THE PDP-11
2016
2017
2018
2019
2020
2021
2022                          THE FOLLOWING IS THE PROCEDURE WHICH THE KL10 EXECUTES IN ORDER
2023                          TO DUMP AND/OR BOOTSTRAP THE PDP-11 THROUGH THE DTE20:
2024
2025
2026                          1. CLEAR THE DTE20 AND INITIATE A BM873 BUTTON #4 BOOTSTRAP OPERATION
2027                                -   CONO [SR11B!CL11PT!CLTO11!CLTO10!PILDEN]
2028
2029                          2. WAIT TO SEE PDP-11 POWER FAIL (AC LOW = TRUE)   -   CONI [DEAD11] = 1
2030
2031                          3. WAIT TO SEE PDP-11 POWER RECOVER (AC LOW = FALSE)   -   CONI [DEAD11] = 0
2032
2033                          4. WAIT AT LEAST ANOTHER 150 MILLISECONDS AND THEN CLEAR THE RELOAD -11 BUTTON
2034                                -   CONO [CR11B]
2035
2036                          5. SET BYTE COUNTER TO A SPECIAL CODE (1365 OCTAL)   -   DATAO [1365]
2037
2038                          6. RING PDP-11'S DOORBELL   -   CONO[TO11DB]
2039
2040                          7. WAIT UNTIL "-10 RINGING -11'S DOORBELL" IS TURNED OFF BY THE -11
2041                                (I.E. UNTIL   CONI[TO11DB]   BECOMES ZERO).
2042
2043                          8. ENABLE THE DTE20 TO USE PI 0 INTERRUPTS
2044                                (I.E. SET   CONO[PILDEN!PIOENB]).
2045
2046                          9. SET UP THE TO-10 BYTE POINTER (IN THE EPT) FOR THE FIRST 3.5K.
2047
2048                          10. SET UP THE BYTE COUNTER FOR THE FIRST 3.5K, INDICATING
2049                                "INTERRUPT -10 ONLY"   -   DATAO [1000]
2050
2051                          11. WAIT FOR "TO-10 DONE" OR "TO-10 ERROR"   -   CONI [TO10DN!TO10ER]
2052
2053                          12. NOTE WHETHER THERE WAS AN ERROR (CONI [TO10ER]) AND THEN TURN OFF
2054                                TO10DN AND TO10ER   -   CONO [CLTO10]. IF ERROR, GO TO STEP 17.
2055
2056                          13. IF END OF 28K, GO TO STEP 17.
2057
2058                          14. SET UP TO-10 BYTE POINTER (IN THE EPT) FOR THE NEXT 3.5K.
2059
2060                          15. SET UP THE BYTE COUNTER FOR THE NEXT 3.5K INDICATING
2061                                "INTERRUPT -10 ONLY" (DATAO [1000]), UNLESS THIS IS THE
2062                                LAST 3.5K (OF 28K), IN WHICH CASE INDICATE "INTERRUPT
2063                                BOTH PROCESSORS" (DATAO [TO10IB!1000]).
2064
2065                          16. GO TO STEP 11.
2066
2067
```

```
2068            ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
2069            ;BM873-YD.P11   PROCEDURE BY WHICH THE PDP-10 BOOTSTRAPS AND/OR DUMPS THE PDP-11
2070
2071
2072                            17. SET UP TO-11 BYTE POINTER (IN THE EPT) FOR "PDP-11 BOOTSTRAP".
2073                                NOTE THAT THE FIRST WORD OF THIS "PDP-11 BOOTSTRAP" MUST
2074                                BE THE BIT PATTERN 000240 (A PDP-11 NOP INSTRUCTION).
2075
2076                            18. RING THE PDP-11'S DOORBELL   -   CONO [TO11DB]
2077
2078                            19. WAIT FOR EITHER TO11DB TO GO OFF (CONI[TO11DB] = 0),
2079                                OR TO10DB TO COME ON (CONI[TO10DB] = 1).
2080
2081                            20. IF NO ERROR WAS NOTED IN STEP 12, TO11DB SHOULD GO OFF
2082                                (TO10DB COMING ON INDICATES A MASSIVE SCREWUP).
2083                                IF AN ERROR WAS NOTED IN STEP 12, TO11DB GOING OFF INDICATES
2084                                THAT THE ERROR WAS "NON-FATAL" (NON-EX-MEM OR -11 MEMORY
2085                                PARITY) AND THE -11 IS PROCEEDING. TO10DB COMING ON INDICATES
2086                                THAT THE ERROR WAS "FATAL" AND THE -11 IS HALTED AT LOCATION 173714.
2087                                IN THIS LATTER CASE THE -10 MUST RESTART FROM STEP 1.
2088
2089                            21. IF TO11DB WENT OFF, WAIT FOR "TO-11 DONE" OR "TO-11 ERROR"
2090                                -   CONI [TO11DN!TO11ER]
2091
2092                            22. NOTE WHETHER THERE WAS AN ERROR   -   CONI [TO11ER]
2093
2094                            23. TURN OFF TO11DN AND TO11ER AND RING THE PDP-11'S DOORBELL
2095                                -   CONO [TO11DB!CLTO11]
2096
2097                            24. WAIT FOR EITHER TO11DB TO GO OFF (CONI[TO11DB] = 0),
2098                                OR TO10DB TO COME ON (CONI[TO10DB] = 1).
2099
2100                            25. TO11DB GOING OFF INDICATES THAT THE PDP-11 FOUND NO ERRORS
2101                                AND IS TRANSFERRING CONTROL TO THE CODE WHICH WAS JUST
2102                                RECEIVED FROM THE -10. IN THIS CASE THE -10 SHOULD START
2103                                FOLLOWING THE PROTOCOL OF THIS CODE.
2104
2105                            26. TO10DB COMING ON INDICATES THAT THE PDP-11 HAS FOUND AN
2106                                ERROR (OR THAT THE FIRST WORD TRANSMITTED WASN'T THE
2107                                BIT PATTERN 000240), AND THE PDP-11 IS HALTED AT LOCATION 173766.
2108                                IN THIS CASE THE -10 MUST RESTART FROM STEP 1.
2109
2110
2111
2112
2113
2114
```

```
2115                    ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
2116                    ;BM873-YD.P11   BUTTON #4  -  BOOTSTRAP INITIATED BY THE PDP-10 (THROUGH DTE20)
2117
2118
2119                    ;       000130  DTECOR  =       130             ;CORE ADDRESS INTO WHICH TO STORE DTE20 REGS.
2120                    ;       000014  DTEREG  =       ↑D12            ;NUMBER OF DTE20 REGISTERS TO STORE
2121                    ;       000400  DTEWDC  =       ↑D256           ;WORD COUNT FOR SECONDARY BOOTSTRAP FROM THE -10
2122                                                    ; ENTER HERE WHEN THE DTE20 PRESSES BUTTON #4 (BOOTSTRAP INITIATED
2123                                                    ;       BY THE PDP-10, THROUGH THE DTE20)
2124  004534  010037 ;173534  010037  BUTON4: MOV     R0,@#ROTOR7     ;SAVE R0 IN PDP-11 MEMORY LOCATION "ROTOR7"
2125  004536  000040 ;173536  000040
2126  004540  010700 ;173540  010700          MOV     PC,R0           ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
2127  004542  000644 ;173542  000644          BR      REGSAV          ;GO TO THE "REGISTER SAVING" SUBROUTINE
2128  004544  005005 ;173544  005005          CLR     R5              ;SET R5 = 0
2129  004546  012501 ;173546  012501          MOV     (R5)+,R1        ;SAVE LOCATION 0 IN R1
2130  004550  012503 ;173550  012503          MOV     (R5)+,R3        ;SAVE LOCATION 2 IN R3
2131  004552  012504 ;173552  012504          MOV     (R5)+,R4        ;SAVE LOCATION 4 IN R4
2132  004554  011500 ;173554  011500          MOV     (R5),R0         ;SAVE LOCATION 6 IN R0
2133  004556  012715 ;173556  012715          MOV     #PR7,(R5)       ;SET UP PRIORITY FOR NON-EX-MEM TRAP
2134  004560  000340 ;173560  000340
2135  004562  005745 ;173562  005745          TST     -(R5)           ;SET R5 = 4
2136  004564  012702 ;173564  012702  10$:    MOV     #DTEBAS-DTESIZ,R2
2137  004566  174340 ;173566  174340
2138  004570  010715 ;173570  010715          MOV     PC,(R5)         ;STORE ADDRESS FOR NON-EX-MEM TRAP
2139  004572  010506 ;173572  010506          MOV     R5,SP           ;SET STACK POINTER = 4
2140  004574  062702 ;173574  062702  11$:    ADD     #DTESIZ,R2      ;R2 POINTS TO THE NEXT DTE20
2141  004576  000040 ;173576  000040
2142  004600  105702 ;173600  105702          TSTB    R2
2143  004602  100770 ;173602  100770          BMI     10$             ;START LOOKING FROM THE BEGINNING AGAIN
2144  004604  032762 ;173604  032762          BIT     #TO11DB,STATUS(R2)      ;IS THIS -10 RINGING THE -11'S DOORBELL?
2145  004606  004000 ;173606  004000
2146  004610  000034 ;173610  000034
2147  004612  001770 ;173612  001770          BEQ     11$             ;IF IT IS NOT, GO LOOK FOR ANOTHER -10
2148  004614  026217 ;173614  026217          CMP     TO10BC(R2),(PC) ;CHECK FOR A CODE (1365) FROM THE PDP-10
2149  004616  000014 ;173616  000014
2150                                                                  ;INDICATING THAT IT WANTS TO BOOTSTRAP THE -11
2151  004620  001365 ;173620  001365          BNE     11$
2152                                                  ; NOTE THAT AT THIS POINT R2 CONTAINS THE ADDRESS OF THE DEVICE REGISTER
2153                                                  ;       BLOCK FOR THIS DTE20, THAT R5 = 4, AND THAT SP = 4
2154  004622  005725 ;173622  005725          TST     (R5)+           ;SET R5 = 6
2155  004624  010015 ;173624  010015          MOV     R0,(R5)         ;RESTORE THE CONTENTS OF LOCATION 6
2156  004626  010445 ;173626  010445          MOV     R4,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 4
2157  004630  010345 ;173630  010345          MOV     R3,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 2
2158  004632  010145 ;173632  010145          MOV     R1,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 0
2159                                                  ; NOTE: AT THIS TIME R5 = 0. THIS FACT WILL BE USED LATER.
2160  004634  012700 ;173634  012700          MOV     #DTECOR,R0      ;R0 = CORE ADDRESS FOR STORING DTE20 REGISTERS
2161  004636  000130 ;173636  000130
2162  004640  010204 ;173640  010204          MOV     R2,R4
2163  004642  012420 ;173642  012420  7$:     MOV     (R4)+,(R0)+     ;SAVE THE NEXT DTE20 REGISTER IN CORE
2164  004644  022700 ;173644  022700          CMP     #<DTEREG*2>+DTECOR,R0   ;HAVE WE FINISHED YET?
2165  004646  000160 ;173646  000160
2166  004650  101374 ;173650  101374          BHI     7$              ;LOOP UNTIL WE HAVE FINISHED
```

```
                        ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
                        ;BM873-YD.P11   BUTTON #4  -  BOOTSTRAP INITIATED BY THE PDP-10 (THROUGH DTE20)
                        ;
004652  010201 ;173652  010201          MOV     R2,R1                   ;R1 = DTE20 DEVICE REGISTER BLOCK
004654  062701 ;173654  062701          ADD     #DIAG2,R1
004656  000032 ;173656  000032
004660  012721 ;173660  012721          MOV     #DRESET,(R1)+           ;DO A "DIAGNOSTIC CLEAR" OF THE DTE20,
004662  000100 ;173662  000100
                                        ; THE ABOVE  OPERATION IS NECESSARY TO CLEAR THE "BYTE COUNT LOADED" FLAG
                                        ; AND SIMULTANEOUSLY TO TURN OFF "-10 RINGING -11'S DOORBELL".
004664  005012 ;173664  005012          CLR     (R2)                    ;SET DTE20 FOR NO DELAY
004666  005062 ;173666  005062          CLR     TO10AD(R2)      ;START WRITING -11 MEMORY INTO THE -10.
004670  000020 ;173670  000020
004672  032711 ;173672  032711  6$:     BIT     #TO11DB,(R1)    ;HAS THE -10 RUNG THE -11'S DOORBELL?
004674  004000 ;173674  004000
004676  001775 ;173676  001775          BEQ     6$              ;LOOP UNTIL IT HAS.
004700  032762 ;173700  032762          BIT     #DUPE!DURE!NUPE,DIAG3(R2)       ;"FATAL" ERROR?
004702  000026 ;173702  000026
004704  000036 ;173704  000036
004706  001403 ;173706  001403          BEQ     8$              ;BRANCH IF NO "FATAL" ERROR
004710  012711 ;173710  012711          MOV     #TO10DB,(R1)    ;SIGNAL "FATAL" ERROR TO THE PDP-10
004712  000400 ;173712  000400
004714  000000 ;173714  000000  2$:     HALT                    ;HALT DUE TO "FATAL" ERROR
004716  012762 ;173716  012762  8$:     MOV     #DRESET,DIAG2(R2)       ;RESET AFTER POSSIBLE PDP-11
004720  000100 ;173720  000100
004722  000032 ;173722  000032
                                        ; MEMORY PARITY ERROR OR NON-EX-MEM ERROR, AND ALSO TURN OFF
                                        ; "-10 RINGING -11'S DOORBELL".
004724  005062 ;173724  005062  3$:     CLR     TO11AD(R2)      ;START INPUTTING AT LOCATION 0
004726  000022 ;173726  000022
004730  012762 ;173730  012762          MOV     #INT10!<<-DTEWDC>&7777>,TO11BC(R2)      ;READ IN 256 WORDS
004732  107400 ;173732  107400
004734  000016 ;173734  000016
004736  032711 ;173736  032711  1$:     BIT     #TO11DB,(R1)    ;HAS THE -10 RUNG THE -11'S DOORBELL?
004740  004000 ;173740  004000
004742  001775 ;173742  001775          BEQ     1$              ;LOOP UNTIL IT HAS.
004744  132711 ;173744  132711  4$:     BITB    #TO11DN!TO11ER,(R1)     ;IS THE TRANSMISSION FINISHED?
004746  000202 ;173746  000202
004750  001775 ;173750  001775          BEQ     4$              ;LOOP UNTIL IT IS FINISHED
004752  100003 ;173752  100003          BPL     5$              ;IF "TO11DN" ISN'T ON, "TO11ER" MUST BE ON
004754  022715 ;173754  022715          CMP     #000240,(R5)    ;CHECK FOR BIT PATTERN IN LOCATION ZERO
004756  000240 ;173756  000240
004760  001403 ;173760  001403          BEQ     9$              ;UNLESS THERE IS A "NOP" IT IS AN ERROR
004762  012711 ;173762  012711  5$:     MOV     #TO1CDB,(R1)    ;SIGNAL THE -10 THAT THERE WAS AN ERROR
004764  000400 ;173764  000400
004766  000000 ;173766  000000  12$:    HALT                    ;THIS ERROR HALT IS BECAUSE EITHER "TO11ER"
                                        ; IS ON, OR BECAUSE THE BIT PATTERN READ INTO LOCATION ZERO WASN'T "000240".
004770  012762 ;173770  012762  9$:     MOV     #DRESET,DIAG2(R2)       ;SIGNAL THE -10 THAT EVERYTHING IS OK
004772  000100 ;173772  000100
004774  000032 ;173774  000032
004776  END.YD:
004776  000115 ;173776  000115          JMP     (R5)            ;JUMP TO LOCATION ZERO
                        ;       000001          .END
```

```
2220  005000  MAP.YF:
2221          ;THE FOLLOWING IS A REPRODUCTION
2222          ;OF THE ROM PROGRAM FOR BM873YF.
2223          ;IT IS HERE FOR COMPARISON TO THE
2224          ;ACTUAL ROM AND FOR REFERENCE
2225                  ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
2226                  ;
2227                                  ;       TITLE PAGE
2228                                  ;       BM873F  - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)
2229                                  ;
2230                                  ; COPYRIGHT (C) 1975 DIGITAL EQUIPMENT CORPORATION
2231                                  ; ALL RIGHTS RESERVED
2232                                  ;
2233                                  ; THIS IS THE CODE TO BE ENCODED IN THE BOOTSTRAP ROM ON THE BM873-YF BOARD
2234                                  ;
2235                                  ;
2236                                  ; MODULE:       BM873F
2237                                  ;
2238                                  ; DATE:         17-JUN-75
2239                                  ;
2240                                  ; AUTHOR:       TOM PORCHER
2241                                  ;
2242                                  ;
2243                  ;                       .ENABLE ABS,AMA
2244                  ;       177776  PS=177776               ;PROCESSOR STATUS REGISTER
2245                  ;       177570  SWR=177570              ;FRONT PANEL SWITCH REGISTER
2246                  ;       000000  PR0=0*40                ;PRIORITY LEVEL 0
2247                  ;       000040  PR1=1*40                ;PRIORITY LEVEL 1
2248                  ;       000100  PR2=2*40                ;PRIORITY LEVEL 2
2249                  ;       000140  PR3=3*40                ;PRIORITY LEVEL 3
2250                  ;       000200  PR4=4*40                ;PRIORITY LEVEL 4
2251                  ;       000240  PR5=5*40                ;PRIORITY LEVEL 5
2252                  ;       000300  PR6=6*40                ;PRIORITY LEVEL 6
2253                  ;       000340  PR7=7*40                ;PRIORITY LEVEL 7
2254                  ;       000001  BIT0=000001
2255                  ;       000002  BIT1=000002
2256                  ;       000004  BIT2=000004
2257                  ;       000010  BIT3=000010
2258                  ;       000020  BIT4=000020
2259                  ;       000040  BIT5=000040
2260                  ;       000100  BIT6=000100
2261                  ;       000200  BIT7=000200
2262                  ;       000400  BIT8=000400
2263                  ;       001000  BIT9=001000
2264                  ;       002000  BIT10=002000
2265                  ;       004000  BIT11=004000
2266                  ;       010000  BIT12=010000
2267                  ;       020000  BIT13=020000
2268                  ;       040000  BIT14=040000
2269                  ;       100000  BIT15=100000
2270                  ;       177400  HIBYTE=177400
```

```
                ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                ;
                                ;       DIRECTIVE FUNCTION CODES
                                ;
                ;       000001  DR.DTE=1.       ;DTE EXAMINE/DEPOSIT/INITALIZE/DOORBELL FUNCTIONS
                                ;
                                ;       DTE FUNCTION CODES (LOW ORDER BY BYTE)
                                ;
                ;       000001  DF.DOR=1        ;DOOR BELL FUNCTION CODE
                ;       000002  DF.OFF=2        ;DTE OFF FUNCTION
                ;       000003  DF.ON=3         ;DTE ON FUNCTION
                ;       000004  DF.DMG=4        ;DEPOSIT MY GENERAL FUNCTION
                ;       000005  DF.EMG=5        ;EXAMINE MY GENERAL FUNCTION
                ;       000006  DF.EMN=6        ;EXAMINE MY FOR N FUNCTION
                ;       000007  DF.DMN=7        ;DEPOSIT MY FOR N FUNCTION
                ;       000010  DF.EHG=10       ;EXAMINE HIS GEN SECTION FUNCTION
                ;       000011  DF.EHM=11       ;EXAMINE HIS SECTION FOR ME FUNCTION
                ;       000012  DF.KLR=12       ;DIAGNOSTIC KL READ
                ;       000013  DF.KLW=13       ;DIAGNOSTIC KL WRITE (FUNCTION 13)
                ;       000014  DF.KLX=14       ;DIAGNOSTIC KL EXECUTE (FUNCTION 14)
                ;       000015  DF.PEX=15       ;PRIVILEDGED EXAMINE (FUNCTION 15)
                ;       000016  DF.PDP=16       ;PRIVILEDGED DEPOSIT (FUNCTION 16)
                                ;
                                ;       CRASH CODES
                                ;
                ;       000001  CC.ILD=1        ;ILLEGAL DIRECTIVE
                ;       000002  CC.EMT=2        ;ILLEGAL EMT
                ;       000003  CC.IDI=3        ;ILLEGAL DTE INTERRUPT
                ;       000004  CC.IOT=4        ;IOT TRAP
                ;       000005  CC.RES=5        ;RESERVED INSTRUCTION TRAP
                ;       000006  CC.TBT=6        ;T BIT OR BPT TRAP
                ;       000007  CC.TRP=7        ;TRAP INSTRUCTION TRAP
                ;       000010  CC.TO4=10       ;TRAP TO 4
                ;       000011  CC.UNT=11       ;ILLEGAL TRAP (UNKNOWN TRAP)
                ;       000012  CC.MPE=12       ;MEMORY PARITY ERROR
                ;       000013  CC.NPF=13       ;RESTRICTED FRONT CAN'T EXECUTE BOOT PROTOCOL
                ;       000014  CC.PTB=14       ;PROTOCOL (PRIMARY) BROKEN
                ;       000015  CC.CST=15       ;CLOCK STOPPED
                ;       000016  CC.ILC=16       ;ILLEGAL COMMAND
                ;       000017  CC.IPO=17       ;INPUT TTY OVERFLOW
                ;       000020  CC.IAS=20       ;INCORRECT VALUE IN .SERFG
                ;       000021  CC.NCE=21       ;NOT ENOUGH ENTRIES IN CLOCK QUEUE
                ;       000022  CC.PIT=22       ;CAN'T EXIT PERMANENT TASK
                ;       000023  CC.UMP=23       ;LOAD REQUEST NOT IMPL YET
                ;       000024  CC.EPE=24       ;E BUS PARITY ERROR
                ;       000025  CC.NDE=25       ;NOT ENOUGH ENTRYS FOR DTE20
                ;       000026  CC.DEX=26       ;DEXDONE TIMEOUT
                ;       000027  CC.TET=27       ;TO TEN ERROR
                ;       000030  CC.ETE=30       ;TO ELEVEN ERROR
                ;       000031  CC.MTF=31       ;MARK TIME FAILURE
                ;       000032  CC.NON=32       ;NOT ENOUGH NODES
                ;       000033  CC.TSP=33       ;TEN STOPPED
                ;       000034  CC.UIE=34       ;UNIMPLEMENTED FUNCTION
                ;       000035  CC.ILQ=35       ;ILLEGAL QUEUE
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;
                                ;
                                ;       GENERAL PROCESSOR DEFINITIONS
                                ;
                        ;       000340  PRI7=340        ;PROCESSOR PRIORITY 7
                                ;
                                ;
                                ;       DTE20 REGISTER DEFINITIONS
                                ;
                                ;       THESE LABELS ARE THOSE USED IN THE FRONT END INTERFACE SPEC
                                ;       EXCEPT STATUS WHICH CONFLICTS WITH PROTOCOL SPEC
                                ;
                                ;       PDM# 200-200-012-00
                                ;
                        ;       174400  DLYCNT=174400   ;DELAY COUNT WORD
                        ;       174402  DEXWD3=174402   ;DEPOSIT OR EXAMINE WORD 3
                        ;       174404  DEXWD2=174404   ;DEPOSIT OR EXAMINE WORD 2
                        ;       174406  DEXWD1=174406   ;DEPOSIT OR EXAMINE WORD 1
                        ;       174410  TENAD1=174410   ;TEN ADDRESS WORD 1
                        ;       174412  TENAD2=174412   ;TEN ADDRESS WORD 2
                        ;       174414  TO10BC=174414   ;TO-10 PDP-11 MEMORY ADDRESS
                        ;       174416  TO11BC=174416   ;TO-11 BYTE COUNT
                        ;       174420  TO10AD=174420   ;TO-10 PDP-11 MEMORY ADDRESS
                        ;       174422  TO11AD=174422   ;TO-11 PDP-11 MEMORY ADDRESS
                        ;       174424  TO10DT=174424   ;TO-10 PDP-11 DATA WORD
                        ;       174426  TO11DT=174426   ;TO-11 PDP-11 DATA WORD
                        ;       174430  DIAG1=174430    ;DIAGNOSTIC WORD 1
                        ;       174432  DIAG2=174432    ;DIAGNOSTIC WORD 2
                        ;       174434  STAT=174434     ;STATUS WORD
                        ;       174436  DIAG3=174436    ;DIAGNOSTIC WORD 3
                                ;
                                ;
                                ;       EXTERNAL PAGE DEFINITIONS (DEVICE DEFINITIONS)
                                ;
                                ;
                                ;       DTE DEFINITIONS
                                ;
                                ;
                                ;       REGISTER BIT DEFINITIONS
                                ;
                                ;
                                ;       TENAD1 DEFINITIONS
                                ;
                        ;       010000  DEP=010000      ;DEPOSIT (BIT 12)
                        ;       004000  PRTOFF=004000   ;EXAMINE/DEPOSIT PROTECT OFF
                        ;       100000  PHYS=100000     ;PHYSICAL EXAMINE
                                ;
                                ;       TO11BC DEFINITIONS
                                ;
                        ;       100000  IFLOP=100000    ;I FLIPFLOP BIT
                        ;       040000  ZSTOP=040000    ;ZSTOP
                        ;       020000  TO11BM=020000   ;TO 11 BYTE MODE
```

```
2379                    ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
2380                    ;
2381                    ;
2382                    ;       DIAG1 DEFINITIONS
2383                    ;
2384                    ;       004000  DS04=004000     ;KL CLOCK ERROR STOP
2385                    ;       002000  DS05=002000     ;RUN
2386                    ;       001000  DS06=001000     ;HALT
2387                    ;       000400  DEX=000400      ;DEPOSIT OR EXAMINE MAJOR STATE
2388                    ;       000200  TO10=000200     ;TO 10
2389                    ;       000200  DFUNC=000200
2390                    ;       000100  TO11=000100     ;TO-11 TRANSFER MAJOR STATE
2391                    ;       000040  D1011=000040    ;DIAGNOSE 10/11 INTERFACE
2392                    ;       000020  PULSE=000020    ;SINGLE CLOCK CYCLE
2393                    ;       000010  DIKL10=00010    ;DIAGNOSTIC MODE SWITCH
2394                    ;       000004  DSEND=000004    ;SEND DATA
2395                    ;       000001  DCOMST=000001   ;DIAGNOSTIC COMMAND START
2396                    ;
2397                    ;       DIAG1 FUNCTIONS
2398                    ;
2399                    ;       000000  .STPCL=0                ;STOP THE KL CLOCK
2400                    ;       001000  .STRCL=01*1000          ;START THE KL CLOCK
2401                    ;       002000  .SSCLK=02*1000          ;SINGLE STEP THE M BOX CLOCK
2402                    ;       003000  .SECLK=03*1000          ;SINGLE STEP THE  EBOX CLOCK. LEAVES THE
2403                    ;                       ;EBOX CLOCK FALSE AND EBOX SYNC TRUE.
2404                    ;                       ;CAUSES (2,3) MBOX CLOCKS DEPENDING ON
2405                    ;                       ;EBOX CLOCK INITIALLY (FALSE,TRUE).
2406                    ;                       ;DOES NOT DEPEND ON 'T' FIELD OR MB WAIT.
2407                    ;       004000  .CECLK=04*1000          ;CONDITIONALLY ISSUE AN EBOX CLOCK IF THE EBOX
2408                    ;                       ;CLOCK IS TRUE.  MAKES EBOX CLOCK FALSE.
2409                    ;                       ;IF ISSUED IN THE MASTER RESET STATE.
2410                    ;                       ;LEAVES EBOX SYNC TRUE.
2411                    ;       005000  .BRCLK=05*1000          ;ISSUE A BURST OF THE CLOCKS.  THE NUMBER
2412                    ;                       ;OF MBOX CLOCKS DESIRED (1-255) HAS BEEN
2413                    ;                       ;BEEN LOADED PREVIOUSLY BY FUNCTIONS LDBRR,LDBRL
2414                    ;                       ;(42,43)
2415                    ;       006000  .CLRMR=06*1000          ;CLEAR MASTER RESET STATE
2416                    ;       007000  .SETMR=07*1000          ;SET MASTER RESET STATE. RUNNING THE CLOCK WHILE IN THIS
2417                    ;                       ;STATE 'CLEARS' THE KL10.
2418                    ;       010000  .CLRUN=10*1000          ;CLEAR THE RUN FLOP.  MAKE THE MICRO CODE GO TO
2419                    ;                       ;THE HALT-LOOP.
2420                    ;       011000  .SETRN=11*1000          ;SET THE RUN FLOP.  ALLOW REPEATED INSTRUCTION EXECUTION
2421                    ;       012000  .CONBT=12*1000          ;SET THE CONTINUE FLOP (MOMENTARY). ALLOW THE
2422                    ;                       ;MICRO CODE TO LEAVE THE HALT LOOP
2423                    ;       014000  .IRLTC=14*1000          ;UNLATCH THE IR AND LOAD IT FROM THE AD.
2424                    ;       015000  .DRLTC=15*1000          ;UNLATCH THE DRAM REGISTER AND ALLOW IT TO LOAD FROM THE
2425                    ;                       ;RAMS
```

```
                        :BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;
                        ;
                        ;       CLOCK LOAD FUNCTIONS
                        ;
                042000  .LDBRR=42*1000  ;LOA D THE RIGHT HAND 4 BITS OF THE 8 BIT
                        ;BURST COUNTER FROM EBUS BITS 32-35
                043000  .LDBRL=43*1000          ;LOAD THE LEFT HAND 4 BITS OF THE BURST CTR.
                044000  .LDSEL=44*1000          ;LOAD THE CLOCK SOURCE AND RATE SELECT
                        ;REGISTER: 32,33        34,35
                        ;       SOURCE          RATE
                        ;        00 NORM XTL    00
                        ;        01 FAST XTL    01 /2
                        ;        10 EXT         10 /4
                        ;        11 UNDEF       11 /8
                045000  .LDDIS=45*1000          ;LOAD THE REGISTER WHICH CONTROLS THE EBOX CLOCK
                        ;DISTRIBUTION.
                        ;BIT    ACTION
                        ;33     DISABLE CONTROL LOGIC CLOCK
                        ;34     DISABLE CONTROL RAM CLOCK
                        ;35     DISABLE DATA PATHS CLOCK
                046000  .LDCK1=46*1000          ;LOAD THE CONDITION-CHECKING ENABLE REGISTER.
                        ;THESE ALL ENABLE THE CLOCK TO STOP AND SHOULD
                        ;BE USED IN CONJUNCTION WITH BIT 35 OF FUNCTION 47
                        ;BIT    FUNCTION
                        ;32     CHECK FM PARITY
                        ;33     CHECK CRAM PARITY
                        ;34     CHECK DRAM PARITY
                        ;35     CHECK FIELD SERVICE PROBE
                047000  .LDCK2=47*1000          ;LOAD THE ENABLE/DISABLE FUNCTION REGISTER
                        ;       BIT     FUNCTION
                        ;32     DISABLE EBOX REQUESTS TO MBOX
                        ;33     SIMULATE AN MB RESP FOR EACH MB WAIT
                        ;34     CHECK AR AND ARX PARITY AND CAUSE A
                        ;APGE FAIL UCODE TRAP IF ERROR
                        ;35     MUST BE SET TO PERFORM DESIRED ACTION OF
                        ;FUNCTION 46 (ABOVE).  STOPS ALL CLOCKS IF AN ERROR
                        ;IS DETECTED.
```

```
                        :BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;               CONTROL RAM LOAD FUNCTIONS
;       057000  .LCRM1=57*1000                  ;EBUS   CRAM
;                       ;08-11  00-03
;                       ;14-17  04-07
;                       ;20-23  08-11
;                       ;26-29  12-15
;                       ;32-35  16-19
;       056000  .LCRM2=56*1000                  ;08-11  20-23
;                       ;14-17  24-27
;                       ;20-23  28-31
;                       ;26-29  32-35
;                       ;32-35  36-39
;       055000  .LCRM3=55*1000                  ;08-11  40-43
;                       ;14-17  44-47
;                       ;20-23  48-51
;                       ;26-29  52-55
;                       ;32-35  56-59
;       054000  .LCRM4=54*1000                  ;08     60
;                       ;10     62
;                       ;14     64
;                       ;16     66
;                       ;20     68
;                       ;22     70
;                       ;26     72
;                       ;28     74
;                       ;32     76
;                       ;34     78
;       053000  .LCRM5=53*1000                  ;01-05  DISP 00-04
;       052000  .LCRDL=52*1000                  ;01-05  CRAM DIAG ADDRES 00-04
;       051000  .LCRDR=51*1000                  ;00-05  CRAM DIAG ADR 05-10
                        ;
                        ;       DRAM LOAD FUNCTIONS
                        ;
;       060000  .LDRM1=60*1000                  ;12-14  DRAM A00-02, EVEN ADDRESSES
                        ;15-17  DRAM B00-02, EVEN ADDRESSES
;       061000  .LDRM2=61*1000                  ;12-14  DRAM A00-02, ODD ADDRESSES
                        ;15-17  DRAM B00-02, ODD ADDRESSES
;       062000  .LDRM3=62*1000                  ;14-17  COMMON J01-04
;       063000  .LDRJV=63*1000                  ;15-17  J08-10, EVEN ADDRESSES
                        ;12     PARIT BIT, EVEN ADDRESSES
;       064000  .LDRJD=64*1000                  ;14     COMMON J07 (NOTE -- J05,6 DO NOT EXIST)
                        ;15-17  J08-10, ODD ADDRESSES
                        ;12     PARITY BIT, ODD ADDRESSES
                        ;
                        ;       IR, DRAM CONTROL FUNCTIONS
                        ;
;       065000  .DSIOJ=65*1000                  ;DISABLES SPECIAL DECODE OF OPCODES 254,7XX
;       066000  .DSACF=66*1000                  ;DISABLE IR AC OUTPUTS
;       067000  .EIOJA=67*1000                  ;ENABEL KL STYLE DECODING OF CODES AND AC'S
                        ;
;       070000  .INICL=70*1000  ;INIT CHANNELS
                        ;
;       071000  .WRMBX=71*1000  ;WRITE M-BOX
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;
                        ;       076000  .MEMRS=76*1000  ;SET KL10 MEM RESET FLOP
                                        ;
                        ;       147000  .RCRM1=147*1000 ;READ C-RAM BITS 0-19
                        ;       146000  .RCRM2=146*1000 ;READ C-RAM BITS 20-39
                        ;       145000  .RCRM3=145*1000 ;READ C-RAM BITS 40-59
                        ;       144000  .RCRM4=144*1000 ;READ C-RAM BITS 60-79
                                        ;
                        ;       141000  .RCSPF=141*1000 ;READ SPEC FIELD OF C-RAM
                                        ;
                        ;       135000  .RDJ71=135*1000 ;READ J07-J10 OF D-RAM
                        ;       134000  .RDJ14=134*1000 ;READ J01-J04 OF D-RAM
                        ;       133000  .RDMAB=133*1000 ;READ A & B FIELD OF D-RAM
                                        ;
                        ;       164000  .CSHRG=164*1000
                        ;       102000  .GFNR=102*1000
                                        ;
                        ;               ;NOTE CONSOLE SOFTWARE MUST PERFORM THIS AS A PART OF
                        ;               ;MASTER RESET CODE
                                ;
                                ;       LOAD AR FUNCTION
                                ;
                        ;       077000  .LDAR=77*1000           ;LOAD THE AR FROM EBUS 0-35
                                ;
                        ;       150000  .PCAB1=150*1000         ;PC-ADDRESS BREAK REGISTERS
                        ;       151000  .PCAB2=151*1000
                        ;       152000  .PCAB3=152*1000
                        ;       153000  .PCAB4=153*1000
                                ;
                                ;
                                ;       DIAG3 DEFINITIONS
                                ;
                        ;       100000  SWSLLT=100000   ;SWAP SELECT LEFT
                        ;       040000  DPS4=040000     ;PARITY
                        ;       000040  SCD=000040      ;SHIFT CAPTURED DATA
                        ;       000020  DUPE= 000020    ;DATO UNIBUS PARITY ERROR
                        ;       000020  CDD=000020      ;CLEAR DUPE AND DURE ERROR FLAGS
                        ;       000010  WEP=000010      ;WRITE EVEN (BAD) PARITY
                        ;       000004  DURE=000004     ;DATO UNIBUS RECEIVE ERROR
                        ;       000002  NUPE=000002     ;NPR UNIBUS PARITY ERROR
                        ;       000002  CNUPE=000002    ;CLEAR NUPE
                        ;       000001  TO10BM=000001   ;TO-10 BYTE TRANSFER MODE
                                ;
                                ;
                                ;       DIAG2 DEFINITIONS
                                ;
                        ;       100000  RFMAD0=100000   ;RFM ADDRESS BIT 0
                        ;       040000  RFMAD1=040000   ;RFM ADDRESS BIT 1
                        ;       040000  EDONES=040000   ;EBUS DONE
                        ;       020000  RFMAD2=020000   ;RFM ADDRESS BIT 2
                        ;       010000  RFMAD3=010000   ;RFM ADDRESS BIT 3
                        ;       000100  DRESET=000100   ;DTE RESET
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;
                                ;
                                ;       STAT DEFINITIONS
                                ;
                        ;  100000  TO10DN=100000   ;TO-10 NORMAL TERMINATION
                        ;  100000  DON10S=100000   ;NORMAL TERMINATION (DONE) TO 10
                        ;  040000  DON10C=040000   ;TO-10 NORMAL TERMINATION STATUS
                        ;  020000  TO10ER=020000   ;TO-10 ERROR TERMINATION
                        ;  020000  ERR10S=020000   ;ERROR TERMINATION STATUS
                        ;  010000  RAMIS0=010000   ;RAM IS ZEROS
                        ;  010000  ERR10C=010000   ;CLEAR TO-10 ERROR TERMINATION
                        ;  004000  TO11DB=004000   ;-10 REQUESTED -11 INTERRUPT
                        ;  004000  INT11S=004000   ;REQ 11 STATUS
                        ;  002000  DXWRD1=002000   ;DEXWORD 1
                        ;  002000  INT11C=002000   ;-10 REQUESTS -11 INTERRUPT STATUS
                        ;  001000  MPE11=001000    ;-11 MEMORY PARITY ERROR
                        ;  001000  PERCLR=001000   ;CLEAR -11 MEMORY PARITY ERROR FLAG STATUS
                        ;  000400  TO10DB=000400   ;-11 REQUEST -10 INTERRUPT
                        ;  000400  INT10S=000400   ;REQUEST -10 INTERRUPT STATUS
                        ;  000200  TO11DN=000200   ;TO-11 TRANSFER DONE
                        ;  000200  DON11S=000200   ;TO-11 NORMAL TERMINATION FLAG STATUS
                        ;  000100  EBSEL=000100    ;E BUFFER SELECT
                        ;  000100  DON11C=000100   ;TO-11 NORMAL TERMINATION FLAG STATUS
                        ;  000040  NULSTP=000040   ;NULL STOP
                        ;  000040  INTRON=000040   ;11 INTERRUPT ENABLE
                        ;  000020  BPARER=000020   ;EBUS PARITY ERROR
                        ;  000020  EBUSPC=000020   ;EBUS PARIT ERROR
                        ;  000010  RM=000010       ;RESTRICTED MODE
                        ;  000010  INTROF=000010   ;DISABLE PDP11 INTERRUPT
                        ;  000004  DEXDON=000004   ;DEPOSIT/EXAMINE DONE
                        ;  000004  EBUSPS=000004   ;EBUS PARITY ERROR SET
                        ;  000002  TO11ER=000002   ;TO-11 BYTE ERROR TERMINATION
                        ;  000002  ERR11S=000002   ;TO-11 ERROR TERMINATION FLAG STATUS
                        ;  000001  INTSON=000001   ;INTERRUPTS ON
                        ;  000001  ERR11C=000001   ;CLEAR TO-11 ERROR TERMINATION FLAG STATUS
                                ;
                                ;       DTE20 COMMUNICATION AREA OFFSETTS (WORD NAMES)
                                ;
                        ;  000000  PIDENT=0        ;PROCESSOR IDENTIFICATION WORD
                        ;  000001  CHNPNT=1        ;POINTER TO COMM AREA OF NEXT PROCESSOR (CIRC LIST)
                        ;  000002  CYCLS=2         ;CLOCK CPS COUNT
                        ;  000003  TOD=3           ;TIME OF DAY
                        ;  000004  DATE=4          ;DATE
                        ;  000005  PSWW1=5         ;PROCESSOR STATUS WORD1
                        ;  000006  PSWW2=6         ;PROCESSOR STATUS WORD2
                        ;  000007  PSWW3=7         ;PROCESSOR STATUS WORD3
                        ;  000010  PSWW4=10        ;PROCESSOR STATUS WORD4
                        ;  000011  PSWW5=11        ;PROCESSOR STATUS WORD5
                        ;  000012  PSWW6=12        ;PROCESSOR STATUS WORD6
                        ;  000013  PSWW7=13        ;PROCESSOR STATUS WORD7
                        ;  000014  PSWW10=14       ;PROCESSOR STATUS WORD10
                        ;  000015  PSWW11=15       ;PROCESSOR STATUS WORD11
                        ;  000016  PSWW12=16       ;PROCESSOR STATUS WORD12
```

```
                        ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 1
                        ;
                        ;       000017  PSWW13=17       ;PROCESSOR STATUS WORD13
                        ;       000020  FORPRO=20       ;FOR PROCESSOR IDENTIFICATON WORD
                        ;       000021  PROPNT=21       ;POINTER TO COMM AREA OF THE PROCESSOR ASSOC WITH THIS BLOCK
                        ;       000022  STATUS=22       ;COMMUNICATION STATUS WORD
                        ;       000023  QSIZE=23        ;QUEUE SIZE WORD
                                        ;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  DTEMTD=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
```

```
2659                    ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 2
2660                    ;
2661                    ;
2662                    ;
2663                    ; DEFINITIONS . . .
2664                    ;
2665                    ;       000040  ROTOR7= 40      ;SAVE R0 TO R7 IN 40 TO 56
2666                    ;
2667                    ;       000130  DTESAV= 130     ;SAVE FIRST 12 DTE REGISTERS DLYCNT TO TO11DT
2668                    ;                               ; IN LOCATIONS 130-156
2669                    ;
2670                    ;       000012  RETRY=  10.     ;DO 10 RETRIES BEFORE HALTING
2671                    ;
2672                    ;       173000  ROMORG= 173000  ;ROM STARTS AT 773000
2673                    ;
2674                    ; ESTABLISH ROM ORIGIN
2675                    ;
2676                    ;       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,R0TOR7+0     ;SAVE R0 IN LOCATION 40
005002  000040  ;173002 000040
005004  013700  ;173004 013700          MOV     SWR,R0          ;GET SWITCH REGISTER
005006  177570  ;173006 177570
005010  032700  ;173010 032700          BIT     #BIT0,R0        ;IS LOW-ORDER BIT SET?
005012  000001  ;173012 000001
005014  001007  ;173014 001007          BNE     BUTONX          ;YES-- LOOK AT CONTENTS
005016  000557  ;173016 000557          BR      REGSAV          ;NO-- SAVE R1-R7 IN 42-56, GO TO ADDRESS IN R0 (FROM SWR
                                        ;
                                        ; BUTTON #3 -- LOAD BOOT FROM RX11 FLOPPY DISK
                                        ;
005020  005000  ;173020 005000  BUTON3: CLR     R0              ;SAY LOAD FROM FLOPPY, UNIT 0
005022  000404  ;173022 000404          BR      BUTONX          ;GO TO COMMON CODE FOR 3 BUTTONS
                                        ;
                                        ; REQUIRED POWER-FAIL VECTOR
                                        ;
005024  173000  ;173024 173000          .WORD   ROMORG,PR7
005026  000340  ;173026 000340
                                        ;
                                        ; BUTTON #2 -- LOAD BOOT FROM RP04 DISK
                                        ;
005030  012700  ;173030 012700  BUTON2: MOV     #BIT7,R0        ;BIT 7 MEANS LOAD FROM RP04
005032  000200  ;173032 000200
                                        ;       BR      BUTONX          ;FALL INTO COMMON CODE
                                        ;
                                        ; R0 IS SAVED IN R5 AS THE PARAMETER WORD PASSED TO BOOT
                                        ; AND CONTAINS ONE OF THE FOLLOWING:
                                        ;
                                        ;       BIT 0 = 1       IF FROM SWITCH REGISTER
                                        ;       BIT 7 = 0       LOAD FROM RX11 FLOPPY DISK
                                        ;       BIT 7 = 1       LOAD FROM RP04 DISK
                                        ;       BIT 15 = 1      INDEFINITE RETRY
                                        ;
                                        ; NOTE THAT IF BUTTON #4 IS PRESSED, R5 WILL CONTAIN BIT 0 = 0, BIT 15 = 1
                                        ;
005034  010005  ;173034 010005  BUTONX: MOV     R0,R5           ;SAVE PARAMETER FOR BOOT
005036  106300  ;173036 106300          ASLB    R0              ;LEFT-ALIGN SPEED FIELD IN LOW BYTE
005040  122700  ;173040 122700          CMPB    #16*BIT4,R0     ;IS SPEED 16 OR 17?
005042  000340  ;173042 000340
005044  101404  ;173044 101404          BLOS    10$             ;YES-- UNIT FIELD IS UNIT # TO BOOT FROM
005046  122700  ;173046 122700          CMPB    #3*BIT4,R0      ;IS SPEED 0, 1, OR 2?
005050  000060  ;173050 000060
005052  101001  ;173052 101001          BHI     10$             ;YES-- UNIT IS UNIT TO USE
005054  005000  ;173054 005000          CLR     R0              ;NO-- USE UNIT #0
                                        ;
005056  000300  ;173056 000300  10$:    SWAB    R0              ;GET UNIT # IN LOW BYTE
005060  042700  ;173060 042700          BIC     #^C7,R0         ;TRIM TO 3 BITS 2, 1, 0
005062  177770  ;173062 177770
```

```
2731                                    ;
2732                                    ; UNIT # IS IN R0-- CALL PROPER BOOT DEPENDING ON BIT 7
2733                                    ;
2734  005064  105705  ;173064 105705          TSTB    R5              ;WHERE SHOULD WE BOOT FROM?
2735  005066  100550  ;173066 100550          BMI     RPBOOT          ;BIT 7 = 1 -- BOOT FROM RP04 DISK
2736                                    ;     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
                                ;
```

```
2783                                    ;
2784                                    ; HERE TO BOOT FROM RX11 FLOPPY DISK-- UNIT # IN R0
2785                                    ;
2786                    ;173070         RXBOOT:
2787  005070  012703    ;173070 012703          MOV     #<RXBTRK*BIT8>!<RXBSCT*BIT0>,R3
2788  005072  000401    ;173072 000401
2789  005074  005700    ;173074 005700          TST     R0              ;IS THIS UNIT # 0?
2790  005076  001402    ;173076 001402          BEQ     10$             ;YES-- USE 0
2791  005100  012700    ;173100 012700          MOV     #RXUNIT,R0      ;NO-- USE UNIT # 1
2792  005102  000020    ;173102 000020
2793  005104  052700    ;173104 052700  10$:    BIS     #RXREAD+RXGO,R0 ;SET READ FUNCTION IN R0
2794  005106  000007    ;173106 000007
2795                                    ;       BR      RXSTRT          ;FALL INTO START-UP
```

```
2796                    ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 4
2797                    ;
2798                                    ;
2799                                    ; HERE TO START RX11 ON A TRANSFER, EITHER DUMP OR BOOT
2800                                    ;
2801  005110  012706  ;173110 012706  RXSTRT: MOV     #RETRY,SP       ;SET RETRY COUNT
2802  005112  000012  ;173112 000012
2803  005114  012701  ;173114 012701         MOV     #RXEPA+RXCS,R1  ;ADDRESS CONTROL STATUS REGISTER FOR RX11
2804  005116  177170  ;173116 177170
2805                                    ;        BR      RXRTRY          ;FALL THROUGH RETRY CHECK
2806                                    ;
2807                                    ; HERE ON ERROR TO RETRY
2808                                    ;
2809  005120  005705  ;173120 005705  RXRTRY: TST     R5              ;INDEFINITE RETRY?
2810  005122  100402  ;173122 100402         BMI     10$             ;YES-- TRY FAITHFULLY
2811  005124  005306  ;173124 005306         DEC     SP              ;NO-- DECREMENT RETRY COUNT
2812  005126  002475  ;173126 002475         BLT     RXEHLT          ;GIVE UP IF RUN OUT
2813                                    ;
2814  005130  000005  ;173130 000005  10$:    RESET   ;CLEAR THE WORLD
2815  005132  005004  ;173132 005004         CLR     R4              ;ALWAYS START TRANSFER AT LOCATION ZERO
2816  005134  010302  ;173134 010302         MOV     R3,R2           ;GET START TRACK AND SECTOR
2817  005136  032711  ;173136 032711  20$:    BIT     #RXDONE,(R1)    ;WAIT UNTIL READY FOR FUNCTION
2818  005140  000040  ;173140 000040
2819  005142  001775  ;173142 001775         BEQ     20$             ;NOT YET-- WAIT
2820  005144  005700  ;173144 005700         TST     R0              ;THIS WRITE?
2821  005146  100454  ;173146 100454         BMI     RXFLSL          ;YES-- FILL SILO BEFORE WRITE
2822                                    ;        BR      RXPERF          ;NO-- JUST DO FIRST READ
2823                                    ;
2824                                    ; HERE TO PERFORM READ OR WRITE, AS SPECIFIED IN R0
2825                                    ;
2826  005150  110011  ;173150 110011  RXPERF: MOVB    R0,(R1)         ;DO READ OR WRITE
2827  005152  105711  ;173152 105711  10$:    TSTB    (R1)            ;READY?
2828  005154  100376  ;173154 100376         BPL     10$             ;NO-- WAIT
2829  005156  110261  ;173156 110261         MOVB    R2,RXDB(R1)     ;SET SECTOR #
2830  005160  000002  ;173160 000002
2831  005162  105711  ;173162 105711  20$:    TSTB    (R1)            ;READY FOR TRACK?
2832  005164  100376  ;173164 100376         BPL     20$             ;NO-- WAIT
2833  005166  000302  ;173166 000302         SWAB    R2              ;YES-- GET TRACK #
2834  005170  110261  ;173170 110261         MOVB    R2,RXDB(R1)     ;SET IT
2835  005172  000002  ;173172 000002
2836  005174  000302  ;173174 000302         SWAB    R2              ;RESTORE HIGH TRACK, LOW SECTOR
2837  005176  032711  ;173176 032711  30$:    BIT     #RXERR!RXDONE,(R1) ;DONE OR ERROR?
2838  005200  100040  ;173200 100040
2839  005202  001775  ;173202 001775         BEQ     30$             ;NO-- WAIT
2840  005204  100745  ;173204 100745         BMI     RXRTRY          ;YES-- ERROR IN FUNCTION
```

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

```
                                        ;
                                        ; WRITE ANOTHER BLOCK-- FILL SILO
                                        ;
005300  012711  ;173300 012711  RXFLSL: MOV     #RXFILL+RXGO,(R1) ;SET TO FILL SILO
005302  000001  ;173302 000001
                                        ;
005304  132711  ;173304 132711  10$:    BITB    #RXTREQ!RXDONE,(R1) ;READY FOR ANOTHER BYTE?
005306  000240  ;173306 000240
005310  001775  ;173310 001775          BEQ     10$             ;NO-- WAIT SOME MORE
005312  100316  ;173312 100316          BPL     RXPERF          ;DONE-- GO PERFORM WRITE
005314  112461  ;173314 112461          MOVB    (R4)+,RXDB(R1)  ;YES-- STORE ANOTHER BYTE IN SILO
005316  000002  ;173316 000002
005320  000771  ;173320 000771          BR      10$             ;WAIT UNTIL READY FOR ANOTHER
                                        ;
                                        ; HERE ON ERROR AFTER RETRYING -- DISPLAY ERROR REGISTER AND HALT
                                        ;
005322  012711  ;173322 012711  RXEHLT: MOV     #RXRERR+RXGO,(R1) ;DO A READ ERROR REGISTER FUNCTION
005324  000017  ;173324 000017
005326  032711  ;173326 032711  10$:    BIT     #RXDONE,(R1)    ;WAIT UNTIL ERROR ASSEMBLED
005330  000040  ;173330 000040
005332  001775  ;173332 001775          BEQ     10$             ;
005334  016100  ;173334 016100          MOV     RXDB(R1),R0     ;GET ERROR REGISTER
005336  000002  ;173336 000002
005340  000476  ;173340 000476          BR      HALTED          ;HALT AND DISPLAY ERRORS
                                        ;
                                        ;
                                        ; START -11 HERE TO DO A DUMP TO RX11 FLOPPY DISK
                                        ;
                                        ; NOTE THAT R0-R7 HAVE ALREADY BEEN SAVED IN 40-56
                                        ; WHEN BUTTON #1 WAS PUSHED
                                        ;
                        ;173342         RXDUMP:
005342  012703  ;173342 012703          MOV     #<RXDTRK*BIT8>!<RXDSCT*BIT0>,R3
005344  035401  ;173344 035401
                        ;
005346  012700  ;173346 012700          MOV     #BIT15!RXWRIT+RXGO,R0 ;DO A WRITE
005350  100005  ;173350 100005
005352  005005  ;173352 005005          CLR     R5              ;CLEAR INDEFINITE RETRY BIT
005354  000655  ;173354 000655          BR      RXSTRT          ;START DUMP GOING
```

```
                        ;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
                                ;
```

```
3010                                    ;
3011                                    ; HERE TO BOOT FROM RP04-- UNIT # IN R0
3012                                    ;
3013                    ;173410         RPBOOT:
3014  005410  005002    ;173410 005002          CLR     R2
3015  005412  005003    ;173412 005003          CLR     R3
3016  005414  052700    ;173414 052700          BIS     #(RPREAD+RPGO)*BIT8,R0 ;SET READ HIGH BYTE, UNIT # LOW BYTE
3017  005416  034400    ;173416 034400
```

```
3018                    ;BM873F - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 3(23)   MACY11 27(657)  22-AUG-75  10:30  PAGE 6
3019                    ;
3020  005420  012704    ;173420 012704          MOV     #-256.,R4       ;READ 256 WORDS TO BOOT
3021  005422  177400    ;173422 177400
3022                            ;       BR      RPSTRT          ;START BOOT GOING
3023                            ;
3024                            ; START RPO4 GOING ON EITHER DUMP OR BOOT
3025                            ;
3026  005424  012706    ;173424 012706  RPSTRT: MOV     #RETRY,SP       ;RETRY RETRY TIMES
3027  005426  000012    ;173426 000012
3028  005430  012701    ;173430 012701          MOV     #RPEPA+RPCS1,R1 ;ADDRESS RPCS1 IN R1
3029  005432  176700    ;173432 176700
3030                            ;       BR      RPRTRY          ;FALL THROUGH RETRY CODE
3031                            ;
3032                            ; HERE ON ERROR TO RETRY
3033                            ;
3034  005434  005705    ;173434 005705  RPRTRY: TST     R5              ;INFINITE RETRY?
3035  005436  100402    ;173436 100402          BMI     10$             ;YES-- TRY AGAIN
3036  005440  005306    ;173440 005306          DEC     SP              ;RETRY COUNT EXHAUSTED?
3037  005442  002437    ;173442 002437          BLT     RPEHLT          ;YES-- GIVE UP
3038                            ;
3039  005444  000005    ;173444 000005  10$:    RESET   ;ZAP!!
3040  005446  110061    ;173446 110061          MOVB    R0,RPCS2(R1)    ;SELECT PROPER UNIT #
3041  005450  000010    ;173450 000010
3042  005452  032711    ;173452 032711          BIT     #RPDVA,(R1)     ;IS DRIVE AVAILABLE TO US?
3043  005454  004000    ;173454 004000
3044  005456  001766    ;173456 001766          BEQ     RPRTRY          ;NO-- TRY AGAIN
3045  005460  012711    ;173460 012711          MOV     #RPPRST+RPGO,(R1) ;DO 'READ-IN PRESET' FUNCTION
3046  005462  000021    ;173460 000021
3047  005464  010261    ;173464 010261          MOV     R2,RPDC(R1)     ;SELECT PROPER CYLINDER
3048  005466  000034    ;173466 000034
3049  005470  010361    ;173470 010361          MOV     R3,RPDA(R1)     ; AND TRACK AND SECTOR
3050  005472  000006    ;173472 000006
3051  005474  010461    ;173474 010461          MOV     R4,RPWC(R1)     ;SET UP WORD COUNT TO PROPER VALUE
3052  005476  000002    ;173476 000002
3053                            ;               ;NOTE THAT IT IS NOT NECCESARY TO SET UP BUS
3054                            ;               ; ADDRESS, SINCE IT IS 0 AFTER READ-IN PRESET
3055  005500  000300    ;173500 000300          SWAB    R0              ;GET FUNCTION CODE IN LOW BYTE
3056  005502  110011    ;173502 110011          MOVB    R0,(R1)         ;START FUNCTION GOING
3057  005504  000300    ;173504 000300          SWAB    R0              ;RESTORE R0
3058                            ;
3059  005506  105711    ;173506 105711  20$:    TSTB    (R1)            ;READY?
3060  005510  100376    ;173510 100376          BPL     20$             ;NO-- WAIT UNTIL IT IS
3061  005512  032711    ;173512 032711          BIT     #RPTRE!RPMCPE,(R1) ;TRANSFER OR MBC PARITY ERROR?
3062  005514  060000    ;173514 060000
3063  005516  001346    ;173516 001346          BNE     RPRTRY          ;YES-- ERROR-- TRY AGAIN
3064  005520  032761    ;173520 032761          BIT     #RPATA!RPERR,RPDS(R1) ;ATTN OR OTHER ERROR?
3065  005522  140000    ;173522 140000
3066  005524  000012    ;173524 000012
3067  005526  001342    ;173526 001342          BNE     RPRTRY          ;YES-- ERROR-- TRY AGAIN
3068  005530  005700    ;173530 005700          TST     R0              ;READ FUNCTION?
3069  005532  100247    ;173532 100247          BPL     CLRPC           ;YES-- BOOT-- GO TO LOCATION 0
3070                            ;       BR      HALT0           ;NO-- DUMP-- HALT WITH R0= 0 IN DISPLAY
```

```
3071                                    ;
3072                                    ; HERE TO HALT AFTER A DUMP-- DISPLAY R0= 0 IF NO ERRORS
3073                                    ;
3074  005534  005000  ;173534 005000    HALTO:  CLR     R0              ;DISPLAY R0= 0 TO INDICATE NO ERRORS
3075                                    ;
3076  005536  000000  ;173536 000000    HALTED: HALT    ;DIE
3077  005540  000776  ;173540 000776            BR      HALTED          ;STAY DEAD
3078                                    ;
3079                                    ; HERE ON ERROR FROM RP04 AFTER RETRYING-- DISPLAY DRIVE STATUS IN R0
3080                                    ;
3081  005542  016100  ;173542 016100    RPEHLT: MOV     RPDS(R1),R0     ;DISPLAY DRIVE STATUS
3082  005544  000012  ;173544 000012
3083  005546  000773  ;173546 000773            BR      HALTED          ;R.I.P.
3084                                    ;
3085                                    ;
3086                                    ; START -11 HERE TO DUMP TO RP04 DISK
3087                                    ;
3088                                    ; NOTE THAT R0-R7 HAVE ALREADY BEEN SAVED IN 40-56
3089                                    ;  BY PRESSING BUTTON #1.
3090                                    ;
3091            ;173550                 RPDUMP:
3092  005550  012702  ;173550 012702            MOV     #RPDCYL,R2
3093  005552  000631  ;173552 000631
3094  005554  012703  ;173554 012703            MOV     #<RPDTRK*BIT8>!<RPDSCT*BIT0>,R3
3095  005556  006410  ;173556 006410
3096  005560  012700  ;173560 012700            MOV     #BIT15!<<RPWRIT+RPGO>*BIT8>,R0 ;DO A WRITE, UNIT # 0
3097  005562  130400  ;173562 130400
3098  005564  012704  ;173564 012704            MOV     #-<1024.*28.>,R4 ;SET TO DUMP 28 K
3099  005566  110000  ;173566 110000
3100  005570  005005  ;173570 005005            CLR     R5              ;CLEAR INDEFINITE RETRY BIT
3101  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 THROUCH 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)
```

```
3146                                    ;
3147                                    ; HERE ON NXM TRAP-- RESET SP AND TRY NEXT DTE
3148                                    ;
3149  005634  012706  ;173634 012706  21$:    MOV     #4,SP           ;SET SP TO 4, STACK IS LOCATIONS 2 AND 0
3150  005636  000004  ;173636 000004
3151                                    ;
3152  005640  062704  ;173640 062704  22$:    ADD     #DTESIZ,R4      ;BUMP TO NEXT DTE'S EXTERNAL PAGE ADDRESS
3153  005642  000040  ;173642 000040
3154  005644  105704  ;173644 105704          TSTB    R4              ;IS THIS THE END OF THE DTE'S?
3155                  ;                       ; NOTE THAT THE LAST DTE IS AT 774540
3156                  ;                       ; AND THAT NOW R4= 774600 IF END
3157  005646  100770  ;173646 100770          BMI     20$             ;YES-- START ALL OVER, UNTIL A DTE
3158                  ;                       ; SAYS HE PUSHED THE BUTTON
3159  005650  032764  ;173650 032764          BIT     #T011DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?
3160  005652  004000  ;173652 004000
3161  005654  000034  ;173654 000034
3162  005656  001770  ;173656 001770          BEQ     22$             ;NO-- TRY NEXT DTE
3163  005660  026417  ;173660 026417          CMP     T010BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365
3164  005662  000014  ;173662 000014
3165                  ;                               ; IN IT'S TO -10 BYTE COUNT?
3166  005664  001365  ;173664 001365          BNE     22$             ;NO-- TRY ANOTHER DTE
3167                                  ;
```

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

```
                                        ;
                                        ; REGISTERS:
                                        ;       R0 --   DRESET (DIAGNOSTIC RESET FUNCTION)
                                        ;       R1 --   STAT (STATUS REGISTER)
                                        ;       R4 --   DIAG2 (DIAGNOSTIC REGISTER #2, WHERE DRESET IS)
                                        ;
                                        ; THE -10 WILL NOW START READING -11 MEMORY, AS SOON AS WE SET
                                        ;  THE TO -10 ADDRESS.  WHEN FINISHED, THE -10 WILL RING OUR DOORBELL.
                                        ;
005724  005061  ;173724 005061                CLR     DLYCNT-STAT(R1) ;SET DTE20 FOR MAXIMUM DELAY (ZERO)
005726  177744  ;173726 177744
005730  005061  ;173730 005061                CLR     TO10AD-STAT(R1) ;START DUMPING -11 MEMORY TO -10
005732  177764  ;173732 177764
                ;                               ; STARTING AT LOCATION 0
005734  032711  ;173734 032711  30$:    BIT     #TO11DB,(R1)    ;IS DOORBELL RINGING (TRANSFER COMPLETE)?
005736  004000  ;173736 004000
005740  001775  ;173740 001775                BEQ     30$             ;NO-- WAIT FOR DOORBELL
005742  010014  ;173742 010014                MOV     R0,(R4)         ;YES-- CLEAR DOORBELL AND ERROR FLAGS
                                        ;
                                        ; NOW THE -10 WILL GIVE US A 256 WORD BOOTSTRAP TO BE READ
                                        ;  INTO -11 MEMORY STARTING AT LOCATION 0.  WHEN FINISHED,
                                        ;  THE -10 WILL RING OUR DOORBELL, AND WE WILL START EXECUTION
                                        ;  OF THE LOADED CODE AT LOCATION 0.
                                        ;
005744  005061  ;173744 005061                CLR     TO11AD-STAT(R1) ;START INPUT TO LOCATION 0
005746  177766  ;173746 177766
005750  012761  ;173750 012761                MOV     #IFLOP!<<-256.>&7777>,TO11BC-STAT(R1) ;256 WORDS, INTERRUPT
005752  107400  ;173752 107400
005754  177762  ;173754 177762
                                                        ; -10 WHEN DONE
005756  032711  ;173756 032711  40$:    BIT     #TO11DB,(R1)    ;DOORBELL RINGING (LOAD FINISHED)?
005760  004000  ;173760 004000
005762  001775  ;173762 001775                BEQ     40$             ;NO-- WAIT UNTIL DONE
005764  010014  ;173764 010014                MOV     R0,(R4)         ;CLEAR DOORBELL RINGING
005766  012705  ;173766 012705                MOV     #BIT15,R5       ;SET R5: BIT15= 1, BIT0= 0
005770  100000  ;173770 100000
                ;                               ; TO SAY BUTTON #4 PRESSED
005772  005007  ;173772 005007                CLR     PC              ;GO TO LOADED CODE, STARTING AT
                                                                        ; LOCATION 0
```

```
                        ;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
                        ;
                        ;
```

```
3259  006000  MAP.YG:
3260          ;THE FOLLOWING IS A REPRODUCTION
3261          ;OF THE ROM PROGRAM FOR BM873YG.
3262          ;IT IS HERE FOR COMPARISON TO THE
3263          ;ACTUAL ROM AND FOR REFERENCE
3264          ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 1
3265
3266          ;                               .TITLE  BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM
3267          ;
3268                                  ; THIS IS THE CODE TO BE ENCODED IN THE BOOTSTRAP ROM ON THE BM873-YG BOARD
3269                                  ;
3270                                  ;
3271                                  ;       MODULE:         BM873G
3272                                  ;
3273                                  ;       DATE:           JUNE 1976
3274                                  ;
3275                                  ;       AUTHOR:         RICH MURATORI
3276                                  ;
3277                                  ;
3278                                  ;       COPYRIGHT (C) 1976 DIGITAL EQUIPMENT CORPORATION
3279                                  ;       ALL RIGHTS RESERVED
3280                                  ;
3281                                  ;
3282          ;                               .ENABLE ABS,AMA
3283                                  ;
3284                                  ;       ASCII CHARACTER DEFINTIONS
3285                                  ;
3286          ;               000040  SPACE=  40                      ;ASCII SPACE
3287          ;               000001  SYN=    1                       ;ASCII SYNC
3288          ;               000012  LF=     12                      ;ASCII LINE FEED
3289          ;               000015  CR=     15                      ;ASCII CARRIAGE RETURN
3290          ;               000054  COMMA=  54                      ;ASCII COMMA
3291          ;               000006  ACK=    6                       ;ASCII ACKNOWLEDGE
3292          ;               000025  NAK=    25                      ;ASCII NEG ACKNOWLEDGE
3293                                  ;
3294                                  ;       BUFFER AREAS
3295                                  ;
3296          ;               002100  LINBUF= 2100                    ;LINE INPUT BUFFER
3297          ;               002310  DEABUF= 2310                    ;DEASCIIZED INPUT BUFFER
3298                                  ;
3299                                  ;       DL11E REGISTER DEFINITIONS
3300                                  ;
3301          ;               176000  DLRCSR= 176000                  ;DL11E RECEIVER STATUS REGISTER
3302          ;               176002  DLRBUF= 176002                  ;DL11E RECEIVER BUFFER
3303          ;               176004  DLXCSR= 176004                  ;DL11E TRANSMITTER STATUS REGISTER
3304          ;               176006  DLXBUF= 176006                  ;DL11E TRANSMISSION BUFFER
3305                                  ;
3306          ;               100000  BIT15=100000
3307          ;               000340  PR7=7*40                ;PRIORITY LEVEL 7
3308                                  ;
3309                                  ;       DTE20 REGISTER DEFINITIONS
3310                                  ;
3311          ;               174400  DLYCNT=174400   ;DELAY COUNT WORD
3312          ;               174414  TO10BC=174414   ;TO-10 PDP-11 MEMORY ADDRESS
3313          ;               174416  TO11BC=174416   ;TO-11 BYTE COUNT
3314          ;               174420  TO10AD=174420   ;TO-10 PDP-11 MEMORY ADDRESS
```

```
                        174422  TO11AD=174422   ;TO-11 PDP-11 MEMORY ADDRESS
                        174426  TO11DT=174426   ;TO-11 PDP-11 DATA WORD
                        174434  STAT=174434     ;STATUS WORD
                                ;       TO11BC REGISTER BIT DEFINITIONS
                                ;
```

```
3320          ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 1-1
3321
3322          ;                100000  IFLOP=100000    ;I FLIPFLOP BIT
3323                                   ;
3324                                   ;
3325                                   ;       DIAG2 DEFINITIONS
3326                                   ;
3327          ;                000100  DRESET=000100   ;DTE RESET
3328                                   ;
3329                                   ;       STAT REGISTER DEFINITIONS
3330                                   ;
3331          ;                004000  T011DB=004000   ;-10 REQUESTED -11 INTERRUPT
3332                                   ;
3333                                   ;
3334                                   ; DEFINITIONS . . .
3335                                   ;
3336          ;                000040  ROTOR7= 40      ;SAVE R0 TO R7 IN 40 TO 56
3337                                   ;
3338          ;                000130  DTESAV= 130     ;SAVE FIRST 12 DTE REGISTERS DLYCNT TO T011DT
3339          ;                                        ; IN LOCATIONS 130-156
3340                                   ;
3341          ;                173000  ROMORG= 173000  ;ROM STARTS AT 773000
3342                                   ;
3343                                   ; ESTABLISH ROM ORIGIN
3344                                   ;
3345          ;                173000          .=ROMORG
```

```
3346            ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 2
3347
3348            ;                               .SBTTL  BOOT FROM MASTER FRONT END
3349            ;
3350                                    ;EXTERNAL BUTTON #1 -- LOAD FROM MASTER FRONT END VIA DL11E
3351                                    ;
3352                                    ;DEPRESSING BUTTON #1 CAUSES A LOADER UTILITY PROGRAM (DGQDE) TO BE LOADED
3353                                    ;
3354                                    ;TO ACCOMPLISH THIS, THE FOLLOWING CONDITIONS MUST EXIST:
3355                                    ;       1) THE LOADER UTILITY PROGRAM DGQDD MUST BE RUNNING UNDER KLDCP
3356                                    ;          IN THE MASTER FRONT END.
3357                                    ;       2) THE '.A11' FILE FOR DGQDE MUST BE AVAILABLE ON THE SELECTED LOAD
3358                                    ;          MEDIUM IN THE MASTER FRONT END.
3359                                    ;       3) THE MASTER FRONT END AND THE SECONDARY FRONT END MUST BE CONNECTED
3360                                    ;          THROUGH DL11E'S AND A NULL MODEM.
3361                                    ;       4) THE SECONDARY FRONT END MUST HAVE A TTY CONNECTED TO IT.
3362            ;
3363                                    ;SEND BOOTSTRAP REQUEST TO MASTER FRONT END VIA THE DL11E.  THE
3364                                    ;REQUEST IS IN THE FORM 'B <CR><LF>', WHICH CALLS FOR THE
3365                                    ;BOOTING OVER OF DGQDE.A11.
3366  006000  000005  ;173000 000005  BUTON1: RESET                   ;CLEAR THE WORLD
3367  006002  012706  ;173002 012706         MOV     #2000,SP        ;SETUP STACK POINTER
3368  006004  002000  ;173004 002000
3369  006006  012701  ;173006 012701         MOV     #DGQDE,R1       ;ADDRESS OF SYNC + BOOT REQUEST
3370  006010  173374  ;173010 173374
3371  006012  105737  ;173012 105737  SENDIT: TSTB    @#DLXCSR        ;IS DL11E READY TO TRANSMIT
3372  006014  176004  ;173014 176004
3373  006016  100375  ;173016 100375         BPL     SENDIT          ;LOOP UNTIL IT IS
3374  006020  000403  ;173020 000403         BR      1$              ;BRANCH AROUND POWER -FAIL VECTOR
3375                                    ;*****************************************************************
3376                                    ;       REQUIRED POWER-FAIL VECTOR   - MUST BE AT 173024
3377            ;
3378  006022  000000  ;173022 000000         .WORD   0               ;FILLER
3379  006024  173000  ;173024 173000         .WORD   ROMORG,PR7
3380  006026  000340  ;173026 000340
3381            ;
3382                                    ;*****************************************************************
3383            ;
3384  006030  112137  ;173030 112137  1$:     MOVB    (R1)+,@#DLXBUF  ;LOAD A CHAR INTO OUTPUT BUFFER
3385  006032  176006  ;173032 176006
3386  006034  105711  ;173034 105711         TSTB    (R1)            ;ANY MORE CHARS TO SEND?
3387  006036  001365  ;173036 001365         BNE     SENDIT          ;BRANCH IF MORE CHARS TO SEND
3388  006040  005005  ;173040 005005         CLR     R5              ;CLEAR SYNC RECEIVED FLAG
3389                                    ;WAIT TO RECEIVE BOOT PROGRAM (DGQDE.A11), ONE ASCIIZED CHAR AT
3390                                    ;A TIME, ONE LINE AT A TIME.
3391  006042  012701  ;173042 012701  NXTLIN: MOV     #LINBUF,R1      ;ADDRESS OF LINE INPUT BUFFER
3392  006044  002100  ;173044 002100
3393  006046  105737  ;173046 105737  NXTCHR: TSTB    @#DLRCSR        ;CHAR RECEIVED YET?
3394  006050  176000  ;173050 176000
3395  006052  100375  ;173052 100375         BPL     NXTCHR          ;BRANCH IF STILL TO WAIT
3396            ;
3397                                    ;PROCESS THE RECEIVED ASCIIZED CHAR.  IGNORE ALL CHARS UNTIL A SYNC
3398                                    ;SIGNAL IS RECEIVED.  A LINE FEED MARKS THE END OF A LINE.  THE MAX
3399                                    ;NUMBER OF CHARS PER LINE IS 131, MORE THAN THAT IS AN ERROR.
3400  006054  113711  ;173054 113711  2$:     MOVB    @#DLRBUF,(R1)   ;READ CHAR INTO LINE INPUT BUFFER
3401  006056  176002  ;173056 176002
```

```
3402  006060  142711  ;173060 142711          BICB    #200,(R1)       ;CLEAR HIGH ORDER BIT OF CHAR
3403  006062  000200  ;173062 000200
3404  006064  001770  ;173064 001770          BEQ     NXTCHR          ;BRANCH IF YES, IGNORE NULLS
3405  006066  121127  ;173066 121127          CMPB    (R1),#SYN       ;IS CHAR THE SYNC SIGNAL
3406  006070  000001  ;173070 000001
3407  006072  001413  ;173072 001413          BEQ     3$              ;BRANCH IF YES
3408  006074  005705  ;173074 005705          TST     R5              ;HAS SYNC ALREADY BEEN RECEIVED?
3409  006076  001763  ;173076 001763          BEQ     NXTCHR          ;BRANCH IF NOT, IGNORE CHAR
3410  006100  122127  ;173100 122127          CMPB    (R1)+,#LF       ;IS CHAR A LINE FEED?
3411  006102  000012  ;173102 000012
3412  006104  001410  ;173104 001410          BEQ     PACKIT          ;BRANCH IF YES, END OF LINE
```

```
3413            ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM MACY11 27(663) 1-JUN-76  09:14  PAGE 2-1
3414
3415  006106 020127  ;173106 020127          CMP     R1,#LINBUF+132. ;HAVE WE REACHED THE END OF THE BUFFER?
3416  006110 002304  ;173110 002304
3417  006112 003755  ;173112 003755          BLE     NXTCHR          ;BRANCH IF NOT, GET REST OF LINE
3418  006114 004737  ;173114 004737          JSR     PC,NAKSND       ;SEND A NEG ACKNOWLEDGE
3419  006116 173346  ;173116 173346
3420  006120 000000  ;173120 000000          HALT                    ;TOO MANY CHARS IN A LINE
3421          ;
3422  006122 005205  ;173122 005205  3$:     INC     R5              ;SET SYNC RECEIVED FLAG
3423  006124 000750  ;173124 000750          BR      NXTCHR          ;GO GET NEXT CHAR
3424          ;
3425                                 ;UNSCRAMBLE THE ASCIIZED CHARS INTO 16-BIT WORDS.  THE FORMAT OF A
3426                                 ;LINE IS    E WRDCNT,LADDR,DATA,DATA,...,DATA,CHKSUM<CR><LF>
3427                                 ;WHERE WRDCNT IS THE WORD COUNT
3428                                 ;       LADDR IS THE LOAD ADDRESS
3429                                 ;       DATA IS LOAD DATA
3430                                 ;       CHKSUM IS THE CHECKSUM
3431          ;
3432  006126 012703  ;173126 012703  PACKIT: MOV     #DEABUF,R3      ;GET ADDRESS OF DE-ASCIIZED BUFFER
3433  006130 002310  ;173130 002310
3434  006132 012701  ;173132 012701          MOV     #LINBUF,R1      ;GET ADDRESS OF INPUT BUFFER
3435  006134 002100  ;173134 002100
3436  006136 122127  ;173136 122127          CMPB    (R1)+,#'E       ;FIRST CHAR IN LINE SHOULD BE AN 'E'
3437  006140 000105  ;173140 000105
3438  006142 001403  ;173142 001403          BEQ     1$              ;BRANCH IF IT IS
3439  006144 004737  ;173144 004737          JSR     PC,NAKSND       ;SEND A NEG ACKNOWLEDGE
3440  006146 173346  ;173146 173346
3441  006150 000000  ;173150 000000          HALT                    ;INCORRECT LINE SYNTAX, FIRST CHAR NOT AN E
3442          ;
3443  006152 122127  ;173152 122127  1$:     CMPB    (R1)+,#SPACE    ;SECOND CHAR SHOULD BE A SPACE
3444  006154 000040  ;173154 000040
3445  006156 001403  ;173156 001403          BEQ     NXTWRD          ;BRANCH IF IT IS
3446  006160 004737  ;173160 004737          JSR     PC,NAKSND       ;SEND A NEG ACKNOWLEDGE
3447  006162 173346  ;173162 173346
3448  006164 000000  ;173164 000000          HALT                    ;INCORRECT LINE SYNTAX, 2ND CHAR NOT A SPACE
3449          ;
3450  006166 005002  ;173166 005002  NXTWRD: CLR     R2              ;CLEAR WORD FORMER
3451  006170 112100  ;173170 112100  1$:     MOVB    (R1)+,R0        ;READ CHAR FROM LINE BUFFER
3452  006172 122700  ;173172 122700          CMPB    #CR,R0          ;IS CHAR A CARRIAGE RETURN
3453  006174 000015  ;173174 000015
3454  006176 001774  ;173176 001774          BEQ     1$              ;BRANCH IF YES
3455  006200 122700  ;173200 122700          CMPB    #LF,R0          ;IS CHAR A LINE FEED
3456  006202 000012  ;173202 000012
3457  006204 001422  ;173204 001422          BEQ     3$              ;BRANCH IF IT IS
3458  006206 122700  ;173206 122700          CMPB    #COMMA,R0       ;IS CHAR A COMMA
3459  006210 000054  ;173210 000054
3460  006212 001415  ;173212 001415          BEQ     2$              ;BRANCH IF IT IS
3461  006214 006302  ;173214 006302          ASL     R2              ;SHIFT  WORD OVER TO MAKE ROOM FOR
3462  006216 006302  ;173216 006302          ASL     R2              ;NEXT CHAR
3463  006220 006302  ;173220 006302          ASL     R2
3464  006222 000402  ;173222 000402          BR      4$              ;BRANCH AROUND POWER-FAIL VECTOR
3465                                 ;*****************************************************************
3466                                 ;       REQUIRED POWER-FAIL VECTOR   - MUST BE AT 173224
3467          ;
3468  006224 173000  ;173224 173000          .WORD   ROMORG,PR7
```

```
3469  006226  000340  ;173226 000340
3470                  :
3471                  ;*****************************************************************
3472                  .
3473  006230  006302  ;173230 006302  4$:     ASL     R2
3474  006232  006302  ;173232 006302          ASL     R2
3475  006234  006302  ;173234 006302          ASL     R2
3476  006236  042700  ;173236 042700          BIC     #100,R0         ;CLEAR ASCIIZED BIT
3477  006240  000100  ;173240 000100
3478  006242  050002  ;173242 050002          BIS     R0,R2           ;INSERT NEW CHAR INTO WORD
3479  006244  000751  ;173244 000751          BR      1$              ;GO GET NEXT CHAR
3480                  .
3481  006246  010223  ;173246 010223  2$:     MOV     R2,(R3)+        ;STORE WORD IN BUFFER
```

```
3482                    ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 2-2
3483
3484  006250  000746    ;173250 000746          BR      NXTWRD          ;GO FORM NEXT WORD
3485                    ;
3486  006252  010223    ;173252 010223  3$:     MOV     R2,(R3)+        ;STORE CHECKSUM IN BUFFER
3487                    ;
3488                                    ;VERIFY THAT THE CHECKSUM IS ZERO.
3489  006254  012702    ;173254 012702  CHCKIT: MOV     #DEABUF,R2      ;ADDRESS OF BUFFER
3490  006256  002310    ;173256 002310
3491  006260  005000    ;173260 005000          CLR     R0              ;CLEAR CHECKSUM
3492  006262  062200    ;173262 062200  1$:     ADD     (R2)+,R0        ;ADD NEXT WORD TO CHECKSUM
3493  006264  020203    ;173264 020203          CMP     R2,R3           ;REACHED END OF BUFFER YET
3494  006266  002775    ;173266 002775          BLT     1$              ;BRANCH IF NOT
3495  006270  005700    ;173270 005700          TST     R0              ;IS CHECKSUM = 0?
3496  006272  001403    ;173272 001403          BEQ     LOADIT          ;BRANCH IF YES
3497  006274  004737    ;173274 004737          JSR     PC,NAKSND       ;SEND A NEG ACKNOWLEDGE
3498  006276  173346    ;173276 173346
3499  006300  000000    ;173300 000000          HALT                    ;CHECKSUM ERROR
3500                    ;
3501                                    ;LOAD THE RECEIVED DATA WORDS INTO THE DESIGNATED PLACE IN MEMORY.
3502                                    ;A WORD COUNT OF ZERO INDICATES A TRANSFER BLOCK.  START EXECUTING
3503                                    ;THE LOADED PROGRAM AT THE SPECIFIED ADDRESS.
3504  006302  013700    ;173302 013700  LOADIT: MOV     DEABUF,R0       ;GET LOAD WORD COUNT
3505  006304  002310    ;173304 002310
3506  006306  001413    ;173306 001413          BEQ     2$              ;BRANCH IF IT'S ZERO (A TRANSFER BLOCK)
3507  006310  012702    ;173310 012702          MOV     #DEABUF+4,R2    ;ADDRESS OF FIRST DATA WORD
3508  006312  002314    ;173312 002314
3509  006314  013701    ;173314 013701          MOV     DEABUF+2,R1     ;GET LOAD ADDRESS
3510  006316  002312    ;173316 002312
3511  006320  112221    ;173320 112221  1$:     MOVB    (R2)+,(R1)+     ;MOVE DATA FROM BUFFER TO MEMORY
3512  006322  112221    ;173322 112221          MOVB    (R2)+,(R1)+     ;MOVE DATA FROM BUFFER TO MEMORY
3513  006324  005300    ;173324 005300          DEC     R0              ;DECREMENT WORD COUNT
3514  006326  003374    ;173326 003374          BGT     1$              ;BRANCH UNTIL ALL DATA IS LOADED
3515  006330  004737    ;173330 004737          JSR     PC,ACKSND       ;GO SEND AN ACK
3516  006332  173354    ;173332 173354
3517  006334  000642    ;173334 000642          BR      NXTLIN          ;GO GET NEXT LINE
3518                    ;
3519  006336  004737    ;173336 004737  2$:     JSR     PC,ACKSND       ;GO SEND AN ACK
3520  006340  173354    ;173340 173354
3521  006342  013707    ;173342 013707          MOV     DEABUF+2,PC     ;START ADDRESS OF LOADED PROGRAM
3522  006344  002312    ;173344 002312
3523                    ;
3524                    ;
3525                                    ;NAKSND IS USED TO SEND A NEG ACK BACK TO THE MASTER FRONT END.
3526                                    ;ACKSND IS USED TO SEND AN ACK.
3527  006346  012700    ;173346 012700  NAKSND: MOV     #NAK,R0         ;SETUP ASCII NEG ACK
3528  006350  000025    ;173350 000025
3529  006352  000402    ;173352 000402          BR      RESPND          ;GO SEND IT
3530  006354  112700    ;173354 112700  ACKSND: MOVB    #ACK,R0         ;SETUP ASCII ACK
3531  006356  000006    ;173356 000006
3532  006360  105737    ;173360 105737  RESPND: TSTB    @#DLXCSR        ;IS TRANSMITTER READY?
3533  006362  176004    ;173362 176004
3534  006364  100375    ;173364 100375          BPL     RESPND          ;WAIT TIL IT IS
3535  006366  110037    ;173366 110037          MOVB    R0,@#DLXBUF     ;SEND ACK/NAK
3536  006370  176006    ;173370 176006
3537  006372  000207    ;173372 000207          RTS     PC              ;RETURN TO CALLING ROUTINE
```

```
3538
3539  006374  041001  ;173374    001  DGQDE:  .BYTE   SYN
3540                  ;173375    102          .ASCIZ  /B/<CR><LF>
3541  006376  005015  ;173376 005015
3542  006400  000000  ;173400    000
3543                  ;173401    000          .BYTE   0
3544  006402  000000  ;173402    000          .BYTE   0
3545                  ;173403    000          .BYTE   0
3546  006404  000000  ;173404    000          .BYTE   0
3547                  ;173405    000          .BYTE   0
3548  006406  000000  ;173406    000          .BYTE   0
3549                  ;173407    000          .BYTE   0
3550  006410  000000  ;173410    000          .BYTE   0
```

```
3551                    ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 2-3
3552
3553                            ;173411    000          .BYTE   0
3554   006412  000000           ;173412    000          .BYTE   0
3555                            ;173413    000          .BYTE   0
3556   006414  000000           ;173414    000          .BYTE   0
3557                            ;173415    000          .BYTE   0
3558   006416  000000           ;173416    000          .BYTE   0
3559                            ;173417    000          .BYTE   0
3560   006420  000000           ;173420    000          .BYTE   0
3561                            ;173421    000          .BYTE   0
3562   006422  000000           ;173422    000          .BYTE   0
3563                            ;173423    000          .BYTE   0
3564   006424  000000           ;173424    000          .BYTE   0
3565                            ;173425    000          .BYTE   0
3566   006426  000000           ;173426    000          .BYTE   0
3567                            ;173427    000          .BYTE   0
3568   006430  000000           ;173430    000          .BYTE   0
3569                            ;173431    000          .BYTE   0
3570   006432  000000           ;173432    000          .BYTE   0
3571                            ;173433    000          .BYTE   0
3572   006434  000000           ;173434    000          .BYTE   0
3573                            ;173435    000          .BYTE   0
3574   006436  000000           ;173436    000          .BYTE   0
3575                            ;173437    000          .BYTE   0
3576   006440  000000           ;173440    000          .BYTE   0
3577                            ;173441    000          .BYTE   0
3578   006442  000000           ;173442    000          .BYTE   0
3579                            ;173443    000          .BYTE   0
3580   006444  000000           ;173444    000          .BYTE   0
3581                            ;173445    000          .BYTE   0
3582   006446  000000           ;173446    000          .BYTE   0
3583                            ;173447    000          .BYTE   0
3584   006450  000000           ;173450    000          .BYTE   0
3585                            ;173451    000          .BYTE   0
3586   006452  000000           ;173452    000          .BYTE   0
3587                            ;173453    000          .BYTE   0
3588   006454  000000           ;173454    000          .BYTE   0
3589                            ;173455    000          .BYTE   0
3590   006456  000000           ;173456    000          .BYTE   0
3591                            ;173457    000          .BYTE   0
3592   006460  000000           ;173460    000          .BYTE   0
3593                            ;173461    000          .BYTE   0
3594   006462  000000           ;173462    000          .BYTE   0
3595                            ;173463    000          .BYTE   0
3596   006464  000000           ;173464    000          .BYTE   0
3597                            ;173465    000          .BYTE   0
3598   006466  000000           ;173466    000          .BYTE   0
3599                            ;173467    000          .BYTE   0
3600   006470  000000           ;173470    000          .BYTE   0
3601                            ;173471    000          .BYTE   0
3602   006472  000000           ;173472    000          .BYTE   0
3603                            ;173473    000          .BYTE   0
3604   006474  000000           ;173474    000          .BYTE   0
3605                            ;173475    000          .BYTE   0
3606   006476  000000           ;173476    000          .BYTE   0
```

```
3607                    ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 2-4
3608
3609                            ;173477     000             .BYTE   0
3610    006500  000000          ;173500     000             .BYTE   0
3611                            ;173501     000             .BYTE   0
3612    006502  000000          ;173502     000             .BYTE   0
3613                            ;173503     000             .BYTE   0
3614    006504  000000          ;173504     000             .BYTE   0
3615                            ;173505     000             .BYTE   0
3616    006506  000000          ;173506     000             .BYTE   0
3617                            ;173507     000             .BYTE   0
3618    006510  000000          ;173510     000             .BYTE   0
3619                            ;173511     000             .BYTE   0
3620    006512  000000          ;173512     000             .BYTE   0
3621                            ;173513     000             .BYTE   0
3622    006514  000000          ;173514     000             .BYTE   0
3623                            ;173515     000             .BYTE   0
3624    006516  000000          ;173516     000             .BYTE   0
3625                            ;173517     000             .BYTE   0
3626    006520  000000          ;173520     000             .BYTE   0
3627                            ;173521     000             .BYTE   0
3628    006522  000000          ;173522     000             .BYTE   0
3629                            ;173523     000             .BYTE   0
3630    006524  000000          ;173524     000             .BYTE   0
3631                            ;173525     000             .BYTE   0
3632    006526  000000          ;173526     000             .BYTE   0
3633                            ;173527     000             .BYTE   0
3634    006530  000000          ;173530     000             .BYTE   0
3635                            ;173531     000             .BYTE   0
3636    006532  000000          ;173532     000             .BYTE   0
3637                            ;173533     000             .BYTE   0
3638    006534  000000          ;173534     000             .BYTE   0
3639                            ;173535     000             .BYTE   0
3640    006536  000000          ;173536     000             .BYTE   0
3641                            ;173537     000             .BYTE   0
3642                    ;                                   .EVEN
```

```
3643          ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 3
3644
3645          ;                                       .SBTTL  REGISTER SAVE ROUTINE
3646                                                ;       REGISTER SAVE ROUTINE
3647                                                ;
3648                                                ; REGSAV IS CALLED TO SAVE THE GENERAL REGISTERS R0-R7
3649                                                ; IN MEMORY AT 40-56 (LOCATION ROTOR7).
3650                                                ;
3651                                                ; CALLING SEQUENCE:
3652                                                ;       MOV     R0,ROTOR7+0
3653                                                ;       MOV     #RET,R0
3654                                                ;       BR      REGSAV
3655                                                ; RET:  <RETURN HERE>
3656                                                ;
3657                                                ; ALL REGISTERS RESTORED
3658                                                ;
3659                                                ;
3660          ;                 173540                .=ROMORG+540
3661          ;
3662  006540  010037  ;173540 010037  REGSAV: MOV     R0,ROTOR7+16    ;SAVE R0 AS PC IN 56
3663  006542  000056  ;173542 000056
3664          ;
3665  006544  012700  ;173544 012700          MOV     #ROTOR7+16,R0   ;R0 NOW POINTS TO 56
3666  006546  000056  ;173546 000056
3667          ;
3668  006550  010640  ;173550 010640          MOV     SP,-(R0)        ;SAVE SP IN 54
3669  006552  010540  ;173552 010540          MOV     R5,-(R0)        ;SAVE R5 IN 52
3670  006554  010440  ;173554 010440          MOV     R4,-(R0)        ;SAVE R4 IN 50
3671  006556  010340  ;173556 010340          MOV     R3,-(R0)        ;SAVE R3 IN 46
3672  006560  010240  ;173560 010240          MOV     R2,-(R0)        ;SAVE R2 IN 44
3673  006562  010140  ;173562 010140          MOV     R1,-(R0)        ;SAVE R1 IN 42
3674  006564  014000  ;173564 014000          MOV     -(R0),R0        ;RESTORE R0 FROM 40
3675  006566  000177  ;173566 000177          JMP     @ROTOR7+16      ;GO TO SAVED PC
3676  006570  004264  ;173570 004264
3677          ;
3678  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     #T011DB,STAT-DLYCNT(R4) ;DOORBELL RINGING?
006652  004000  ;173652 004000
006654  000034  ;173654 000034
```

```
 3735  006656  001770  ;173656 001770          BEQ     22$                 ;NO-- TRY NEXT DTE
 3736  006660  026417  ;173660 026417          CMP     T010BC-DLYCNT(R4),(PC) ;DOES THIS ONE HAVE 1365
 3737  006662  000014  ;173662 000014
 3738                  ;                                                   ; IN IT'S TO -10 BYTE COUNT?
 3739  006664  001365  ;173664 001365          BNE     22$                 ;NO-- TRY ANOTHER DTE
 3740                  ;
```

```
3741            ;BM873G - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  MACY11 27(663)  1-JUN-76  09:14  PAGE 5
3742
3743                                    ;
3744                                    ; WE HAVE FOUND THE DTE WHICH PUSHED THE BUTTON
3745                                    ;
3746                                    ; ADDRESS OF DLYCNT REGISTER IS IN R4
3747                                    ;
3748  006666  010315  ;173666 010315           MOV     R3,(R5)         ;RESTORE LOCATION 6
3749  006670  010245  ;173670 010245           MOV     R2,-(R5)        ; 4
3750  006672  010145  ;173672 010145           MOV     R1,-(R5)        ; 2
3751  006674  010045  ;173674 010045           MOV     R0,-(R5)        ; 0
3752                                    ;
3753                                    ; SAVE FIRST 12 DTE REGISTERS DLYCNT TO TO11DT
3754                                    ;  IN LOCATIONS 130-156
3755                                    ;
3756  006676  012700  ;173676 012700           MOV     #DTESAV,R0      ;POINT TO SAVE AREA
3757  006700  000130  ;173670 000130
3758  006702  012420  ;173702 012420   29$:    MOV     (R4)+,(R0)+     ;SAVE A REGISTER
3759  006704  022700  ;173704 022700           CMP     #TO11DT-DLYCNT+DTESAV,R0 ;FINISHED?
3760  006706  000156  ;173706 000156
3761  006710  103374  ;173710 103374           BHIS    29$             ;NO-- SAVE SOME MORE
3762                                    ;
3763                                    ; R4= TO11DT+2
3764                                    ;
3765                                    ; SET R1= STATUS REGISTER
3766                                    ;     R4= DIAG2 REGISTER
3767                                    ;
3768                                    ; DO 'DIAGNOSTIC RESET' TO CLEAR DOORBELL AND BYTE COUNT
3769                                    ;  LOADED FLAG
3770                                    ;
3771            ;
3772  006712  005724  ;173712 005724           TST     (R4)+
3773  006714  010401  ;173714 010401           MOV     R4,R1           ; SO DOES R1
3774  006716  012700  ;173716 012700           MOV     #DRESET,R0      ;SETUP R0 FOR 'DIAGNOSTIC RESET'
3775  006720  000100  ;173720 000100
3776  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
```

```
  3828  007000  000177                  MAP.Y.: .BLKW   127.
  3829  007376  000001                  END.Y.: .BLKW   1
  3830  007400  000177                  MAP.YX: .BLKW   127.
  3831  007776  000001                  END.YX: .BLKW   1
```

```
3832                                    ;*****************************************************************
3833                                    ;       INITIALIZATION AND START UP OF PROGRAM.
3834                                    ;*****************************************************************
3835
3836  010000                            RESTRT:
3837  010000  012706  001100                   MOV     #$CMTAG,R6              ;FIRST LOCATION TO BE CLEARED
3838  010004  005026                            CLR     (R6)+                   ;CLEAR MEMORY LOCATION
3839  010006  022706  001136                   CMP     #$TKS,R6                ;DONE?
3840  010012  001374                            BNE     .-6                     ;LOOP BACK IF NO
3841  010014  012706  001100                   MOV     #STACK,SP               ;SETUP THE STACK POINTER
3842  010020  012737  016302  000020           MOV     #$SCOPE,@#IOTVEC        ;IOT VECTOR FOR SCOPE ROUTINE
3843  010026  012737  000340  000022           MOV     #340,@#IOTVEC+2         ;LEVEL 7
3844  010034  012737  016376  000030           MOV     #$ERROR,@#EMTVEC        ;EMT VECTOR FOR ERROR ROUTINE
3845  010042  012737  000340  000032           MOV     #340,@#EMTVEC+2         ;LEVEL 7
3846  010050  012737  016726  000034           MOV     #$TRAP,@#TRAPVEC        ;TRAP VECTOR FOR TRAP CALLS
3847  010056  012737  000340  000036           MOV     #340,@#TRAPVEC+2        ;LEVEL 7
3848  010064  012767  010064  171014           MOV     #.,$LPADR               ;INITIALIZE THE LOOP ADDRESS FOR SCOPE
3849  010072  005067  001156                   CLR     INITFG          ;INITIALIZE TO ASK WHICH TYPE
3850  010076  005067  001142                   CLR     TABLE           ;INITIALIZE TO ASK WHICH TYPE
3851  010102  012706  001100           START:  MOV     #STACK,SP       ;SET THE STACK POINTER
3852  010106  005067  171104                   CLR     LSTERR          ;CLEAR ERROR FLG REPORT
3853  010112  005037  000000                   CLR     @#0             ;SET FOR UNEXPECTED TRAP TO ADD 0
3854  010116  012767  010102  005314           MOV     #START,PRG.NO   ;GET READY FOR PWR FAIL BEFORE FIRST TEST.
3855  010124  012737  000006  000004           MOV     #6,@#4          ;SET TIME OUT TRAP VECTOR
3856  010132  005037  000006                   CLR     @#6             ;SET TIME OUT STATUS TO 0
3857  010136  005067  004620                   CLR     FLAG4           ;CLEAR TEST 4 INITIAL FLAG
3858  010142  005737  000042                   TST     @#42            ;AM I RUNNING UNDER ACT-11??
3859  010146  001002                           BNE     .+6             ;BR IF *WE ARE* UNDER ACT-11!!
3860  010150  000167  000456                   JMP     CONT            ;JUMP IF NOT ACT-11
3861  010154  013746  000004                   MOV     @#4,-(SP)       ;SAV TRAP POINTER
3862  010160  012737  011260  000004           MOV     #NOROM,@#4      ;PUT IN A NEW ONE
3863  010166  005737  173000                   TST     @#173000        ;TRY TO READ THE ROM
3864  010172  000240                           NOP                     ;WAIT FOR POSSIBLE TRAP
3865  010174  012637  000004                   MOV     (SP)+,@#4       ;IF NO TRAP RESTORE POINTER
3866  010200  026737  171174  173000           CMP     MAP.YA,@#173000 ;DOES 1ST WORD COMPARE?
3867  010206  001034                           BNE     64$             ;CHECK NEXT MAP
3868  010210  013746  000004                   MOV     @#4,-(SP)       ;SAVE LOC 4
3869  010214  012737  010236  000004           MOV     #65$,@#4        ;SET FOR TIMEOUT
3870  010222  005737  173400                   TST     @#173400        ;READ FROM 173400
3871  010226  000240                           NOP                     ;IF NO TIMEOUT, NOT YA
3872  010230  012637  000004                   MOV     (SP)+,@#4       ;RESTORE LOC 4
3873  010234  000421                           BR      64$
3874  010236  022626                   65$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
3875  010240  012637  000004                   MOV     (SP)+,@#4       ;RESTORE LOC 4
3876  010244  012767  001400  000772           MOV     #MAP.YA,TABLE   ;1ST MAP ADDR
3877  010252  012767  001776  000766           MOV     #END.YA,ALLEND  ;LAST MAP ADDR
3878  010260  012767  173376  000770           MOV     #173376,LASTA   ;LAST ROM ADDR
3879  010266  012767  000101  005266           MOV     #000101,VERSON  ;SET ROM TYPE
3880  010274  000167  001174                   JMP     PRG1            ;START TEST 1
3881  010300                           64$:
3882  010300  026737  171474  173000           CMP     MAP.YB,@#173000 ;DOES 1ST WORD COMPARE?
3883  010306  001016                           BNE     69$             ;CHECK NEXT MAP
3884  010310  012767  002000  000726           MOV     #MAP.YB,TABLE   ;1ST MAP ADDR
3885  010316  012767  002776  000722           MOV     #END.YB,ALLEND  ;LAST MAP ADDR
3886  010324  012767  173776  000724           MOV     #173776,LASTA   ;LAST ROM ADDR
3887  010332  012767  000102  005222           MOV     #000102,VERSON  ;SET ROM TYPE
```

```
3888  010340  000167  001130              JMP     PRG1            ;START TEST 1
3889  010344                      69$:
3890  010344  026737  172430  173000      CMP     MAP.YC,@#173000 ;DOES 1ST WORD COMPARE?
3891  010352  001036                      BNE     74$             ;CHECK NEXT MAP
3892  010354  013746  000004              MOV     @#4,-(SP)       ;SAVE LOC 4
3893  010360  012737  010404  000004      MOV     #76$,@#4        ;SET FOR TIMEOUT
3894  010366  026737  173006  173400      CMP     MAP.YC+400,@#173400     ;IS IT YC?
3895  010374  001004                      BNE     77$             ;BR IF NOT YC
3896  010376  012637  000004              MOV     (SP)+,@#4       ;RESTORE LOC 4
3897  010402  000404                      BR      78$             ;YES IT IS A YC
3898  010404  022626              76$:    CMP     (SP)+,(SP)+     ;ADJUST STACK
3899  010406  012637  000004      77$:    MOV     (SP)+,@#4       ;RESTORE LOC 4
3900  010412  000416                      BR      74$             ;CHECK NEXT MAP
3901  010414                      78$:
3902  010414  012767  003000  000622      MOV     #MAP.YC,TABLE   ;1ST MAP ADDR
3903  010422  012767  003776  000616      MOV     #END.YC,ALLEND  ;LAST MAP ADDR
3904  010430  012767  173776  000620      MOV     #173776,LASTA   ;LAST ROM ADDR
3905  010436  012767  000103  005116      MOV     #000103,VERSON  ;SET ROM TYPE
3906  010444  000167  001024              JMP     PRG1            ;START TEST 1
3907  010450                      74$:
3908  010450  026737  173324  173000      CMP     MAP.YD,@#173000 ;DOES 1ST WORD COMPARE?
3909  010456  001016                      BNE     79$             ;CHECK NEXT MAP
3910  010460  012767  004000  000556      MOV     #MAP.YD,TABLE   ;1ST MAP ADDR
3911  010466  012767  004776  000552      MOV     #END.YD,ALLEND  ;LAST MAP ADDR
3912  010474  012767  173776  000554      MOV     #173776,LASTA   ;LAST ROM ADDR
3913  010502  012767  000104  005052      MOV     #000104,VERSON  ;SET ROM TYPE
3914  010510  000167  000760              JMP     PRG1            ;START TEST 1
3915  010514                      79$:
3916  010514  026737  174260  173000      CMP     MAP.YF,@#173000 ;DOES 1ST WORD COMPARE?
3917  010522  001016                      BNE     84$             ;CHECK NEXT MAP
3918  010524  012767  005000  000512      MOV     #MAP.YF,TABLE   ;1ST MAP ADDR
3919  010532  012767  005776  000506      MOV     #END.YF,ALLEND  ;LAST MAP ADDR
3920  010540  012767  173776  000510      MOV     #173776,LASTA   ;LAST ROM ADDR
3921  010546  012767  000106  005006      MOV     #000106,VERSON  ;SET ROM TYPE
3922  010554  000167  000714              JMP     PRG1            ;START TEST 1
3923  010560                      84$:
3924  010560  026737  175214  173000      CMP     MAP.YG,@#173000 ;DOES 1ST WORD COMPARE?
3925  010566  001016                      BNE     89$             ;CHECK NEXT MAP
3926  010570  012767  006000  000446      MOV     #MAP.YG,TABLE   ;1ST MAP ADDR
3927  010576  012767  006776  000442      MOV     #END.YG,ALLEND  ;LAST MAP ADDR
3928  010604  012767  173776  000444      MOV     #173776,LASTA   ;LAST ROM ADDR
3929  010612  012767  000107  004742      MOV     #000107,VERSON  ;SET ROM TYPE
3930  010620  000167  000650              JMP     PRG1            ;START TEST 1
3931  010624                      89$:
3932  010624  104400  011360              TYPE    ,NMATCH         ;NOT BM873YA OR B OR C OR D OR F OR G
3933  010630  000000                      HALT
3934  010632  005767  000416      CONT:   TST     INITFG          ;IS THIS THE FIRST TIME START UP?
3935  010636  001145                      BNE     3$              ;BR IF NOT FIRST TIME HERE.
3936  010640  005167  000410              COM     INITFG          ;SET THE FLAG
3937  010644  104400  012066      2$:     TYPE    ,BM873X         ;TYPE THE QUESTION.
3938  010650  104412                      RDLIN
3939  010652  012602                      MOV     (SP)+,R2
3940  010654  011202                      MOV     (R2),R2         ;PLACE CHARACTER INTO R2.
3941  010656  022702  000052              CMP     #52,R2          ;WAS * HIT??
3942  010662  001011                      BNE     64$             ;BR IF NO
3943  010664  012767  007000  000352      MOV     #MAP.Y.,TABLE   ;SET FOR START OF TABLE
```

```
3944  010672  012767  007376  000346          MOV     #END.Y.,ALLEND  ;SET END OF TABLE
3945  010700  012767  173376  000350          MOV     #173376,LASTA   ;SET LAST ROM ADDR
3946  010706                          64$:
3947  010706  042702  000040                  BIC     #40,R2          ;CLEAR LOWER CASE BIT JUST IN CASE
3948  010712  022702  000101                  CMP     #101,R2         ;WAS A HIT??
3949  010716  001011                          BNE     65$             ;BR IF NO
3950  010720  012767  001400  000316          MOV     #MAP.YA,TABLE   ;SET FOR START OF TABLE
3951  010726  012767  001776  000312          MOV     #END.YA,ALLEND  ;SET END OF TABLE
3952  010734  012767  173376  000314          MOV     #173376,LASTA   ;SET LAST ROM ADDR
3953  010742                          65$:
3954  010742  022702  000102                  CMP     #102,R2         ;WAS B HIT??
3955  010746  001011                          BNE     66$             ;BR IF NO
3956  010750  012767  002000  000266          MOV     #MAP.YB,TABLE   ;SET FOR START OF TABLE
3957  010756  012767  002776  000262          MOV     #END.YB,ALLEND  ;SET END OF TABLE
3958  010764  012767  173776  000264          MOV     #173776,LASTA   ;SET LAST ROM ADDR
3959  010772                          66$:
3960  010772  022702  000103                  CMP     #103,R2         ;WAS C HIT??
3961  010776  001011                          BNE     67$             ;BR IF NO
3962  011000  012767  003000  000236          MOV     #MAP.YC,TABLE   ;SET FOR START OF TABLE
3963  011006  012767  003776  000232          MOV     #END.YC,ALLEND  ;SET END OF TABLE
3964  011014  012767  173776  000234          MOV     #173776,LASTA   ;SET LAST ROM ADDR
3965  011022                          67$:
3966  011022  022702  000104                  CMP     #104,R2         ;WAS D HIT??
3967  011026  001011                          BNE     68$             ;BR IF NO
3968  011030  012767  004000  000206          MOV     #MAP.YD,TABLE   ;SET FOR START OF TABLE
3969  011036  012767  004776  000202          MOV     #END.YD,ALLEND  ;SET END OF TABLE
3970  011044  012767  173776  000204          MOV     #173776,LASTA   ;SET LAST ROM ADDR
3971  011052                          68$:
3972  011052  022702  000106                  CMP     #106,R2         ;WAS F HIT??
3973  011056  001011                          BNE     69$             ;BR IF NO
3974  011060  012767  005000  000156          MOV     #MAP.YF,TABLE   ;SET FOR START OF TABLE
3975  011066  012767  005776  000152          MOV     #END.YF,ALLEND  ;SET END OF TABLE
3976  011074  012767  173776  000154          MOV     #173776,LASTA   ;SET LAST ROM ADDR
3977  011102                          69$:
3978  011102  022702  000107                  CMP     #107,R2         ;WAS G HIT??
3979  011106  001011                          BNE     70$             ;BR IF NO
3980  011110  012767  006000  000126          MOV     #MAP.YG,TABLE   ;SET FOR START OF TABLE
3981  011116  012767  006776  000122          MOV     #END.YG,ALLEND  ;SET END OF TABLE
3982  011124  012767  173776  000124          MOV     #173776,LASTA   ;SET LAST ROM ADDR
3983  011132                          70$:
3984  011132  010267  004424                  MOV     R2,VERSON       ;STORE VERSION TYPE..
3985  011136  005767  000102                  TST     TABLE           ;HAS A MAP BEEN SELECTED?
3986  011142  001003                          BNE     3$              ;BR IF OK...
3987  011144  104400  012143                  TYPE    ,BM.ERR         ;TYPE ERROR
3988  011150  000635                          BR      2$              ;GO AND GET CORRECT MAP.
3989  011152  104400  015104          3$:     TYPE    ,MSG3           ;TYPE MESSAGE FOR TEST NUMBER
3990  011156  104412                  X.X.:   RDLIN
3991  011160  012602                          MOV     (SP)+,R2
3992  011162  011203                          MOV     (R2),R3         ;MOV THE CHAR TO R3
3993  011164  022703  000061          2$:     CMP     #61,R3          ;WAS 1 HIT??
3994  011170  001002                          BNE     4$              ;BR IF NO
3995  011172  000167  000276                  JMP     PRG1            ;GOTO PRG 1
3996  011176  022703  000062          4$:     CMP     #62,R3          ;WAS 2 HIT??
3997  011202  001002                          BNE     5$              ;BR IF NO
3998  011204  000167  001006                  JMP     PRG2            ;GOTO PRG 2
3999  011210  022703  000063          5$:     CMP     #63,R3          ;WAS 3 HIT??
```

```
 4000  011214  001002                        BNE     6$              ;BR IF NO
 4001  011216  000167  001734                JMP     PRG3            ;GOTO PRG3
 4002  011222  022703  000064        6$:     CMP     #64,R3          ;WAS 4 HIT??
 4003  011226  001002                        BNE     3$              ;BR IF NO
 4004  011230  000167  002732                JMP     PRG4            ;GOTO PRG 4
 4005  011234  104400  015310        3$:     TYPE    ,M.QM           ;NEITHER 1 OR 2 OR 3 OR 4 WAS HIT
 4006  011240  000167  176534                JMP     RESTRT          ;TYPE "??" GO TO THE BEGINING.
 4007  011244  000000                TABLE:  0
 4008  011246  000000                ALLEND: 0
 4009  011250  007400                EXTMAP: MAP.YX
 4010  011252  007776                EXTEND: END.YX
 4011  011254  000000                INITFG: 0
 4012  011256  000000                LASTA:  0
 4013  011260  104400  011270        NOROM:  TYPE    ,NOROMS ;TYPE   CAN'T FIND A RESPONSE
 4014  011264  000000                        HALT            ;NO LOADER INSTALLED?
 4015  011266  000776                        BR      .-2
 4016  011270  005015  051124  050101 NOROMS: .ASCII  <15><12>/TRAP TO 4 ON 1ST READ OF 173000/
       011331     015  044412  020123        .ASCIZ  <15><12>/IS LOADER INSTALLED?/
       011360  005015  040503  023516 NMATCH: .ASCII  <15><12>/CAN'T IDENTIFY LOADER AS YA,YB,YC,YD,YF OR YG AFTER/
       011445     015  041412  050115        .ASCIZ  <15><12>/CMP WITH LOC 173000/
                011474                .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.

011474 012767 011474 003736  PRG1:   MOV   #PRG1,PRG.NO    ;SET FOR PWR FAIL
011502 012767 000500 167510          MOV   #500,ICOUNT     ;DO THIS TEST 500(8) TIMES.
011510 012737 015442 000004  PRG.1:  MOV   #NO.TRAP,@#4    ;SET FOR UNEXPECTED TRAP.
011516 012700 173000                 MOV   #173000,R0      ;SET BEGGINING ADDRESS
011522 012767 011546 167356          MOV   #2$,$LPADR      ;IF SW14=1; GOTO 2$ WHEN SCOPE IS HIT
011530 016704 177510                 MOV   TABLE,R4        ;SET START OF MAP
011534 016767 177516 000322          MOV   LASTA,LAST      ;SET LAST ADDRESS
011542 012703 000005         1$:     MOV   #5.,R3          ;DO EACH ADDRESS 5 TIMES.
011546 022700 173024         2$:     CMP   #173024,R0      ;DON'T DO THE VECTOR ADD.
011552 001001                        BNE   20$             ;BR IF NOT THE VECTOR ADD.
011554 022024                        CMP   (R0)+,(R4)+     ;UPDATE TO NEXT ADDRESS
011556 022700 173224         20$:    CMP   #173224,R0      ;DON'T DO THE TRAP VECTORS
011562 001001                        BNE   21$             ;NO THIS ISN'T A TRAP VECTOR.
011564 022024                        CMP   (R0)+,(R4)+     ;UPDATE THE POINTERS..
011566 010467 167332         21$:    MOV   R4,$GDDAT
011572 010067 167330                 MOV   R0,$BDDAT
011576 011067 167424                 MOV   (R0),TEMP4      ;READ THE ADDRESS
011602 011467 167416                 MOV   (R4),TEMP3      ;READ THE SOFTWARE ADDRESS
011606 026767 167412 167412          CMP   TEMP3,TEMP4
011614 001401                        BEQ   22$             ;BR IF GOOD
011616 104001                        ERROR 1               ;INCORRECT COMPARISON.
011620 032767 004000 165742  22$:    BIT   #BIT11,SWR      ;QUICK PASS.?
011626 001002                        BNE   23$             ;BR IF YES
011630 005303                        DEC   R3              ;HAS THAT ADD BEEN READ 5 TIMES?
011632 001345                        BNE   2$              ;BR IF NOT 5 TIMES

011634 026700 000224         23$:    CMP   LAST,R0         ;WAS LAST ADDRESS CHECKED?
011640 001403                        BEQ   10$             ;BR IF YES
011642 000004                        SCOPE                 ;LOCK ON THIS ADDRESS IF SW14=1
011644 022024                        CMP   (R0)+,(R4)+     ;UPDATE THE POINTERS.
011646 000735                        BR    1$              ;CONTINUE THE TEST.

011650 032767 000001 165712  10$:    BIT   #BIT0,SWR       ;EXTENDED WORD TO BE CHECKED?
011656 001413                        BEQ   3$              ;BR IF NO CHECKING.
011660 022767 173776 000176          CMP   #173776,LAST    ;IS ALL THE TEST DONE?
011666 001407                        BEQ   3$              ;BR IF YES.
011670 012767 173776 000166          MOV   #173776,LAST    ;SET LAST ADDRESS.
011676 016704 177346                 MOV   EXTMAP,R4       ;SET EXTENDED MAP.
011702 005720                        TST   (R0)+           ;POP POINTER
011704 000716                        BR    1$              ;GO DO THE TEST.
```

```
                                        ;TEST THAT WRITTING ROM RESULTS IN A TIME OUT
                                        ;TRAP.
                                
012706  012767  011732  167172  3$:     MOV     #5$,$LPADR      ;IF SW14=1 GOTO 5$ WHEN SCOPE IS HIT
012714  012700  173000                   MOV     #173000,R0      ;SET R0 WITH BASE ADDRESS OF ROM
011720  012737  011766  000004          MOV     #6$,@#4         ;SET FOR TIME OUT TRAP
011726  012703  000005          4$:     MOV     #5.,R3          ;DO EACH ADD 5 TIMES
011732  022700  173024          5$:     CMP     #173024,R0      ;CHECK FOR A TRAP VECTOR
011736  001001                          BNE     24$             ;BR IF NOT VECTOR
011740  005720                          TST     (R0)+           ;UPDATE THE REGISTER POINTER
011742  022700  173224          24$:    CMP     #173224,R0      ;CHECK FOR THE OTHER VECTOR
011746  001001                          BNE     25$             ;BR IF NOT THE VECTOR
011750  005720                          TST     (R0)+           ;UPDATE THE POINTER
011752  012710  177777          25$:    MOV     #-1,(R0)        ;WRITE ROM WITH A -1
011756  000240                          NOP                     ;WAIT ONE INSTR. TIME
011760  010067  167242                   MOV     R0,TEMP4
011764  104002                          ERROR   2               ;WRITING ROM DIDN'T TIME OUT.
011766  012706  001100          6$:     MOV     #STACK,SP       ;RESTORE STACK
011772  032767  004000  165570          BIT     #BIT11,SWR      ;QUICK PASS?
012000  001002                          BNE     30$
012002  005303                          DEC     R3              ;DO EACH ADD 5 TIMES
012004  001352                          BNE     5$              ;NOT DONE WITH THIS ONE YET.

012006  032767  000001  165554  30$:    BIT     #BIT0,SWR       ;EXTENDED 128. WORDS TO BE CHECKED?
012014  001404                          BEQ     31$             ;BR IF NO
012016  022700  173776                   CMP     #173776,R0      ;HAVE ALL 256. WORDS BEEN CHECKED?
012022  001407                          BEQ     7$              ;BR IF ALL DONE
012024  000403                          BR      32$             ;KEEP GOING
012026  026700  177224          31$:    CMP     LASTA,R0        ;ALL DONE??
012032  001403                          BEQ     7$              ;HAVE ALL 128. WORDS DONE?
012034  000004          32$:    SCOPE                           ;CHECK SW14 FOR FREEZE!!
012036  005720                          TST     (R0)+           ;UPDATE TO NEXT ADDRESS
012040  000732                          BR      4$              ;GO DO IT AGAIN
012042  005367  167152          7$:     DEC     ICOUNT          ;ITERATION COUNT DONE?
012046  001004                          BNE     8$              ;BR IF NOT DONE.
012050  004767  003426                   JSR     PC,EOP          ;TYPE END MESSAGE
012054  000167  177414                   JMP     PRG1            ;GO DO IT AGAIN.
012060  000167  177424          8$:     JMP     PRG.1           ;GO RESTART.
012064  000000          LAST:   0

012066  005015  040515  047111  BM873X: .ASCII  <15><12>/MAINDEC-11-DZBMDG/
012111     015  042012  053105          .ASCII  <15><12>/DEVICE VERSION/
012131     015  041012  034115          .ASCIZ  <15><12>/BM873-Y/
012143     015  025012  040454  BM.ERR: .ASCIZ  <15><12>/*,A,B,C,D,F,G ONLY./
012171     040  020040  042526  VERS:   .ASCIZ  /   VERSION: BM873-Y/
           012216                  .EVEN
```

```
4108                                        ;PROGRAM 2
4109                                        ;BLIND READ FROM ROM.
4110                                        ;THIS PROGRAM WILL DUMP THE CONTENTS OF THE ROM OUT
4111                                        ;PERFORMING NO CHECKING AT ALL.
4112                                        ;PLEASE NOTE: NO CHECKING IS DONE.
4113
4114  012216 012767 012216 003214 PRG2:   MOV    #PRG2,PRG.NO    ;SET FOR POWER FAIL
4115  012224 012737 015442 000004         MOV    #NO.TRAP,@#4    ;SET FOR UNEXPECTED TRAP TO 4
4116  012232 016767 177020 177624         MOV    LASTA,LAST
4117  012240 062767 000002 177616         ADD    #2,LAST
4118  012246 012700 173000        21$:    MOV    #173000,R0      ;SET R0 WITH THE STARTING ROM ADD.
4119  012252 016703 176766                MOV    TABLE,R3        ;SET POINTER.
4120  012256 104400 012576                TYPE   ,DH.2           ;TYPE MESSAGE
4121  012262 104400 012660                TYPE   ,DH.2B          ;TYPE THE HEADER
4122  012266 012767 000007 166726 1$:     MOV    #7,TEMP5        ;SET COUNTER
4123  012274 011001                       MOV    (R0),R1         ;READ THE ROM
4124  012276 010067 166722                MOV    R0,TEMP3        ;STORE R0
4125  012302 010167 166720                MOV    R1,TEMP4        ;STORE R1
4126  012306 022767 007000 176730         CMP    #MAP.Y.,TABLE   ;IF BM873.Y* SELECTED; FILL TABLE
4127  012314 001001                       BNE    22$             ;BR IF NOT BM873.Y*
4128  012316 011023                       MOV    (R0),(R3)+      ;FILL THE TABLE..
4129  012320 005720               22$:    TST    (R0)+           ;POP THE POINTER
4130  012322 104400 015324                TYPE   ,MCRLF
4131
4132  012326 016746 166672                MOV    TEMP3,-(SP)
4133  012332 104402                       TYPOC
4134  012334 104400 015315                TYPE   ,MSPACE         ;TYPE THREE SPACES.
4135
4136
4137  012340 016746 166662                MOV    TEMP4,-(SP)
4138  012344 104402                       TYPOC
4139  012346 011001               7$:     MOV    (R0),R1         ;STORE ROM DATA
4140  012350 010067 166650                MOV    R0,TEMP3        ;STORE ROM ADDRESS
4141  012354 010167 166646                MOV    R1,TEMP4        ;PREPARE DATA FOR TYPE OUT
4142  012360 022767 007000 176656         CMP    #MAP.Y.,TABLE   ;IS BM873.Y* SELECTED?
4143  012366 001001                       BNE    23$             ;BR IF NO..
4144  012370 011023                       MOV    (R0),(R3)+      ;FILL THE DATA TABLE
4145  012372 005720               23$:    TST    (R0)+           ;POP THE POINTER
4146
4147  012374 104400 015315                TYPE   ,MSPACE
4148
4149  012400 016746 166622                MOV    TEMP4,-(SP)
4150  012404 104402                       TYPOC
4151
4152  012406 026700 177452                CMP    LAST,R0 ;HAS THE HIGHEST LIMIT BEEN HIT?
4153  012412 001404                       BEQ    2$              ;BR IF ALL DONE.
4154  012414 005367 166602                DEC    TEMP5           ;DECREASE COUNTER
4155  012420 001352                       BNE    7$              ;BR IF NOT 0; KEEP GOING
4156  012422 000721                       BR     1$              ;GO TYPE ADDRESS NOW
4157
4158  012424 032767 000001 165136 2$:     BIT    #BIT0,SWR       ;IS THE EXTENDED 128. WORDS TO BE CHECKED??
4159  012432 001455                       BEQ    3$              ;BR IF NO.
4160  012434 012700 173400                MOV    #173400,R0      ;RESET POINTER OF ROM
4161  012440 016703 176604                MOV    EXTMAP,R3       ;SET SOFTWARE MAP POINTER
4162  012444 104400 012772                TYPE   ,DH.2A          ;TYPE NEW HEADER
4163  012450 104400 012660                TYPE   ,DH.2B          ;TYPE ADDRESS AND +XX
```

```
4164  012454  012767  000007  166540  6$:     MOV     #7,TEMP5        ;SET TYPE OUT COUNTER
4165  012462  011001                          MOV     (R0),R1         ;READ THE ROM
4166  012464  010067  166534                  MOV     R0,TEMP3        ;STORE R0
4167  012470  010167  166532                  MOV     R1,TEMP4        ;STORE R1
4168  012474  012023                          MOV     (R0)+,(R3)+     ;STORE THE DATA IN SOFTWARE MAP
4169  012476  104400  015324                  TYPE    ,MCRLF
4170  012502  016746  166516                  MOV     TEMP3,-(SP)
4171  012506  104402                          TYPOC
4172
4173  012510  104400  015315                  TYPE    ,MSPACE
4174  012514  016746  166506                  MOV     TEMP4,-(SP)
4175  012520  104402                          TYPOC
4176
4177  012522  011001                  8$:     MOV     (R0),R1         ;SAVE THE ROM DATA
4178  012524  010067  166474                  MOV     R0,TEMP3        ;SAVE THE ROM ADDRESS
4179  012530  010167  166472                  MOV     R1,TEMP4        ;SET DATA FOR TYPE OUT
4180
4181  012534  104400  015315                  TYPE    ,MSPACE
4182
4183  012540  016746  166462                  MOV     TEMP4,-(SP)
4184  012544  104402                          TYPOC
4185
4186  012546  012023                          MOV     (R0)+,(R3)+     ;STORE THE DATA IN SOFTWARE TABLE
4187  012550  022700  174000                  CMP     #174000,R0      ;HAS THE HIGHEST LIMIT BEEN HIT?
4188  012554  001404                          BEQ     3$              ;BR IF ALL DONE.
4189  012556  005367  166440                  DEC     TEMP5           ;DEC TABLE COUNTER
4190  012562  001357                          BNE     8$              ;BR TO JUST TYPE DATA
4191  012564  000733                          BR      6$              ;BR TO TYPE ADDRESS
4192  012566  005000                  3$:     CLR     R0              ;CLEAR DATA LIGHTS
4193  012570  000000                          HALT                    ;HIT CONTINUE TO PROCEED.
4194  012572  000167  177420                  JMP     PRG2            ;GOTO PRG 2
4195  012576  006414  005012  016412  DH.2:   .ASCII <14><15><12><12><12><35><37><177><177><177>/BLIND READ OF ROM/
      012631     015  006412  077577          .ASCIZ <15><12><15><177><177>/NOTE: NO CHECKING/
      012660  005015  040412  042104  DH.2B:  .ASCII  <15><12><12>/ADDRESS ADD+00  ADD+02  ADD+04/
      012721     040  040440  042104          .ASCIZ /  ADD+06  ADD+10  ADD+12  ADD+14  ADD+16/
      012772  005015  042412  052130  DH.2A:  .ASCII  <15><12><12>/EXTENDED 128. WORD ROM DUMP./
      013031     015  041412  047117          .ASCII  <15><12>/CONTENTS DUMPED IS PLACED IN THE SOFTWARE/
      013104  005015  040515  027120          .ASCII  <15><12>/MAP. DATA SHOULD BE VISUALLY INSPECTED!/
              013156                  .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.
                                        ;
                                        ;

013156  012767  013156  002254  PRG3:   MOV     #PRG3,PRG.NO    ;SET FOR POWER FAIL
013164  016701  176054                  MOV     TABLE,R1        ;DEFAULT STARTING ADDRESS TO MAP
013170  010167  000770                  MOV     R1,ADDRESS      ;SAVE THE SOFTWARE ADDRESS
013174  104400  015217          XHOLD:  TYPE    ,MASTER         ;TYPE AN "*"
013200  104412                          RDLIN
013202  012602                          MOV     (SP)+,R2
013204  011202                          MOV     (R2),R2
013206  042702  000040                  BIC     #40,R2          ;CLEAR LOWER CASE BIT JUST IN CASE
013212  022702  000114                  CMP     #114,R2         ;WAS AN "L" (LIST) HIT?
013216  001464                          BEQ     SRV.L

013220  022702  000104          1$:     CMP     #104,R2         ;WAS A  "D" (DATA) HIT?
013224  001413                          BEQ     SRV.D
013226  022702  000122                  CMP     #122,R2         ;WAS AN "R" (RUN) HIT?
013232  001002                          BNE     10$
013234  000167  000342                  JMP     SRV.R
013240  022702  000101          10$:    CMP     #101,R2         ;WAS AN "A" (ADDRESS) HIT?
013244  001444                          BEQ     SRV.A
013246  104400  015310                  TYPE    ,M.QM           ;TYPE A "?"
013252  000750                          BR      XHOLD           ;NEITHER A "L","P","D","R","A",OR CR WAS HIT.

013254  016767  000704  165744  SRV.D:  MOV     ADDRESS,TEMP4   ;RESET ADDRESS POINTER.
013262  104400  015324                  TYPE    ,MCRLF
013266  016746  165734                  MOV     TEMP4,-(SP)
013272  016701  165730                  MOV     TEMP4,R1
013276  104402                          TYPOC

013300  104400  015315                  TYPE    ,MSPACE

013304  104414                          RDOCT
013306  012611                          MOV     (SP)+,(R1)      ;STORE DATA

013310  005721                          TST     (R1)+           ;UPDATE THE SOFTWARE ADDRESS
013312  026701  175734                  CMP     EXTEND,R1       ;IS THE LIMIT EXCEEDED
013316  103413                          BLO     7$              ;INPUT !LIMIT EXCEEDED!! ERROR.
013320  010167  165702                  MOV     R1,TEMP4        ;SAVE THE ADDRESS.
013324  104400  015324                  TYPE    ,MCRLF
013330  016746  165672                  MOV     TEMP4,-(SP)
013334  104402                          TYPOC

013336  010167  000622                  MOV     R1,ADDRESS      ;SAVE THE ADDRESS FOR GOOD
```

```
4252  013342  000167  177626                  JMP     XHOLD
4253  013346  104400  015310          7$:     TYPE    ,M.QM           ;TYPE A "?"
4254  013352  000167  177616                  JMP     XHOLD
4255
4256                                  ;YOU ARE HERE BECAUSE YOU HIT AN "A"
4257                                  ;YOU TOLD ME YOU WERE GOING TO INPUT AN ADDRESS.
4258                                  ;SO INPUT THE ADDRESS AND TERMINATE WITH A CARRAGE RETURN.
4259                                  ;OK??
4260
4261  013356  104414                  SRV.A:  RDOCT                   ;READ THE ADDRESS HE WANTS TO MODIFY.
4262  013360  012667  000600                  MOV     (SP)+,ADDRESS
4263  013364  000167  177604          4$:     JMP     XHOLD
4264
4265                                  ;YOU ENTERED HERE BECAUSE YOU HIT "L"
4266                                  ;YOU TOLD ME YOU WANTED A LISTING OF THE SOFTWARE MAP
4267                                  ;SO HERE IT IS.
4268
4269
4270  013370                          SRV.L:
4271  013370  016700  175650                  MOV     TABLE,R0        ;GET SOFTWARE MAP
4272  013374  016767  175646  000176          MOV     ALLEND,DEAD     ;SET DEAD END POINTER
4273  013402  104400  015136                  TYPE    ,MSG4           ;TYPE HEADER
4274  013406  104400  012660                  TYPE    ,DH.2B          ;TYPE ADDRESS ADD+XX
4275  013412  012767  000007  165602  1$:     MOV     #7,TEMP5        ;SET COUNTER FOR ACCROSS PAGE
4276  013420  011067  165602                  MOV     (R0),TEMP4      ;GET DATA
4277  013424  010067  165574                  MOV     R0,TEMP3        ;GET ADDRESS
4278  013430  005720                          TST     (R0)+           ;UPDATE ADDRESS POINTER
4279  013432  104400  015324                  TYPE    ,MCRLF
4280
4281  013436  016746  165562                  MOV     TEMP3,-(SP)
4282  013442  104402                          TYPOC
4283
4284  013444  104400  015315                  TYPE    ,MSPACE
4285
4286  013450  016746  165552                  MOV     TEMP4,-(SP)
4287  013454  104402                          TYPOC
4288
4289  013456  104400  015315                  TYPE    ,MSPACE
4290
4291  013462  011067  165540          2$:     MOV     (R0),TEMP4      ;GET DATA
4292  013466  010067  165532                  MOV     R0,TEMP3        ;GET ADDRESS
4293  013472  005720                          TST     (R0)+           ;UPDATE POINTER
4294
4295  013474  016746  165526                  MOV     TEMP4,-(SP)
4296  013500  104402                          TYPOC
4297  013502  104400  015315                  TYPE    ,MSPACE
4298
4299  013506  016703  000066          3$:     MOV     DEAD,R3
4300  013512  005723                          TST     (R3)+           ;UPDATE POINTER
4301  013514  020003                          CMP     R0,R3           ;LIMIT DONE ??
4302  013516  001404                          BEQ     5$              ;BR IF YES
4303  013520  005367  165476          4$:     DEC     TEMP5           ;DEC DATA COUNTER
4304  013524  001356                          BNE     2$              ;BR IF MORE DATA TO GO
4305  013526  000731                          BR      1$              ;TYPE THE ADDRESSS
4306  013530                          5$:
4307  013530  032767  000001  164032          BIT     #BIT0,SWR       ;EXTENDED SOFTWARE DUMP?
```

```
4308  013536  001416                          BEQ     6$              ;BR IF NO DUMP
4309  013540  005743                          TST     -(R3)           ;PUSH POINTER
4310  013542  026703  175504                  CMP     EXTEND,R3
4311  013546  001412                          BEQ     6$              ;BR IF ALL DONE
4312  013550  104400  015164                  TYPE    ,MSG5           ;TYPE EXTENDED MAP:
4313  013554  104400  012660                  TYPE    ,DH.2B
4314  013560  016700  175464                  MOV     EXTMAP,R0       ;SET POINTER
4315  013564  016767  175462  000006          MOV     EXTEND,DEAD     ;SET DEAD END POINTER
4316  013572  000707                          BR      1$              ;DO IT AGAIN SAM.
4317  013574  000167  177374          6$:     JMP     XHOLD
4318  013600  000000                  DEAD:   0
4319
4320                                  ;NOW YOU ARE HERE BECAUSE YOU WANT TO RUN THE PROGRAM
4321                                  ;REMEMBER NOW, YOU SET UP THE MAP.
4322                                  ;ARE YOU SURE YOU TYPED IN THE CORRECT DATA.???
4323                                  ;HERE WE GO
4324
4325  013602                          SRV.R:
4326  013602  012737  015442  000004  RUN3:   MOV     #NO.TRAP,@#4    ;GET READY FOR UNEXPECTED TRAP
4327  013610  012767  000500  165402          MOV     #500,ICOUNT     ;DO TEST 500(8) TIMES
4328  013616  012700  173000          RUN.3:  MOV     #173000,R0      ;SET BEGGINING ADDRESS
4329  013622  012767  013646  165256          MOV     #2$,$LPADR      ;IF SW14=1; GOTO 2$ WHEN I HIT "SCOPE"
4330  013630  016704  175410                  MOV     TABLE,R4        ;SET SOFTWARE RESUTS
4331  013634  016767  175416  176222          MOV     LASTA,LAST      ;SET LAST ADDRESS
4332  013642  012703  000005          1$:     MOV     #5.,R3          ;DO EACH ADDRESS 5 TIMES.
4333  013646  022700  173024          2$:     CMP     #173024,R0      ;DON'T DO THE VECTOR ADD.
4334  013652  001001                          BNE     30$             ;BR IF NOT THE VECTOR ADD.
4335  013654  022024                          CMP     (R0)+,(R4)+     ;UPDATE TO NEXT ADDRESS
4336  013656  022700  173224          30$:    CMP     #173224,R0      ;IS THIS THE SECOND TRAP VECTOR??
4337  013662  001001                          BNE     10$             ;BR IF NOT VECTOR
4338  013664  022024                          CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS !!
4339  013666  010467  165232          10$:    MOV     R4,$GDDAT
4340  013672  010067  165230                  MOV     R0,$BDDAT
4341  013676  011067  165324                  MOV     (R0),TEMP4      ;READ THE ADDRESS
4342  013702  011467  165316                  MOV     (R4),TEMP3      ;READ THE SOFTWARE ADDRESS
4343  013706  026767  165312  165312          CMP     TEMP3,TEMP4
4344  013714  001401                          BEQ     11$             ;BRANCH IF OK
4345  013716  104001                          ERROR   1               ;INCORRECT COMPARISON.
4346  013720  032767  004000  163642  11$:    BIT     #BIT11,SWR      ;QUICK PASS.
4347  013726  001002                          BNE     12$             ;BR IF YES
4348  013730  005303                          DEC     R3              ;HAS THAT ADD BEEN READ 10 TIMES?
4349  013732  001345                          BNE     2$              ;BR IF NOT 10 TIMES
4350  013734  026700  176124          12$:    CMP     LAST,R0         ;WAS LAST ADDRESS CHECKED?
4351  013740  001403                          BEQ     15$             ;BR IF YES
4352  013742  000004                          SCOPE                   ;LOCK ON THIS ADDRESS?
4353  013744  022024                          CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS.
4354  013746  000735                          BR      1$              ;CONTINUE THE TEST.
4355  013750  032767  000001  163612  15$:    BIT     #BIT0,SWR       ;EXTENDED WORD TO BE CHECKED?
4356  013756  001413                          BEQ     3$              ;BR IF NO CHECKING.
4357  013760  022767  173776  176076          CMP     #173776,LAST    ;IS ALL THE TEST DONE?
4358  013766  001407                          BEQ     3$              ;BR IF YES.
4359  013770  012767  173776  176066          MOV     #173776,LAST    ;SET LAST ADDRESS.
4360  013776  016704  175246                  MOV     EXTMAP,R4       ;SET EXTENDED MAP.
4361  014002  005720                          TST     (R0)+           ;POP POINTER
4362  014004  000716                          BR      1$              ;GO DO THE TEST.
```

```
                                        ;TEST THAT WRITING ROM RESULTS IN A TIME OUT
                                        ;TRAP.
                                        
014006  012700  173000          3$:   MOV   #173000,R0      ;SET BASE ADDRESS
014012  012767  014032  165066        MOV   #5$,$LPADR      ;IF SW14=1; GOTO 5$ AT SCOPE
014020  012737  014066  000004        MOV   #6$,@#4         ;TIME OUT TRAP; GOTO 6$
014026  012703  000012          4$:   MOV   #10.,R3         ;DO EACH ADD 10 TIMES
014032  022700  173024          5$:   CMP   #173024,R0      ;IS THIS AT THE TRAP VECTOR
014036  001001                        BNE   20$             ;BR IF NO
014040  005720                        TST   (R0)+           ;UPDATE POINTER
014042  022700  173224         20$:   CMP   #173224,R0      ;IS THIS AT THE SECOND TRAP VECTOR
014046  001001                        BNE   21$             ;BR IF NO
014050  005720                        TST   (R0)+           ;UPDATE THE POINTER
014052  012710  177777         21$:   MOV   #-1,(R0)        ;WRITE ROM WITH A -1
014056  000240                        NOP                   ;WAIT ONE INSTR. TIME
014060  010067  165142                MOV   R0,TEMP4
014064  104002                        ERROR 2               ;WRITING ROM DIDN'T TIME OUT.
014066  012706  001100          6$:   MOV   #STACK,SP       ;RESTORE STACK
014072  032767  004000  163470        BIT   #BIT11,SWR      ;QUICK PASS?
014100  001002                        BNE   22$             ;BR IF YES
014102  005303                        DEC   R3              ;DO EACH ADD 10 TIMES
014104  001352                        BNE   5$              ;NOT DONE WITH THIS ONE YET.
014106  032767  000001  163454  22$:   BIT   #BIT0,SWR      ;IS THE EXTENDED 128. WORDS TO BE TESTED??
014114  001404                        BEQ   23$             ;BR IF NO
014116  022700  173776                CMP   #173776,R0      ;IS THE EXTENDED LIMIT BEEN TESTED?
014122  001407                        BEQ   7$              ;IF YES; GOTO 7$
014124  000403                        BR    24$             ;IF NO; KEEP GOING.
014126  026700  175124         23$:   CMP   LASTA,R0        ;ALL DONE??
014132  001403                        BEQ   7$              ;IF YES; GOTO 7$
014134  000004                 24$:   SCOPE                 ;GO CHECK SW14; (FREEZE !!)
014136  005720                        TST   (R0)+           ;UPDATE TO NEXT ADDRESS
014140  000732                        BR    4$              ;GO DO IT AGAIN
014142  005367  165052          7$:   DEC   ICOUNT          ;CHECK ITERATION COUNT
014146  001004                        BNE   8$              ;MORE TO GO
014150  004767  001326                JSR   PC,EOP          ;GO TO END OF PASS ROUTINE
014154  000167  177422                JMP   RUN3            ;GO DO TEST AGAIN
014160  000167  177432          8$:   JMP   RUN.3           ;

014164  000000                 ADDRESS:      0
```

```
4402                                    ;PROGRAM 4
4403                                    ;PROGRAM 4 CHECKS THE TRAP VECTOR ADDRESS.
4404                                    ;THE PROGRAM SIMULATES ACTIVATING THE BUTTON
4405                                    ;FOR EACH CHANNEL AND THEN READS
4406                                    ;THE CONTENTS OF THE ADDRESS.
4407                                    ;ON THE FIRST PASS THE CONTENTS WILL
4408                                    ;BE TYPED OUT FOR YOU THE
4409                                    ;USER TO VERIFY. AFTER THIS THE PROGRAM
4410                                    ;DOES A COMPARE TO THE PREVIOUSLY FOUND DATA
4411                                    ;AND REPORTS AN ERROR IF DIFFERENT THAN
4412                                    ;WHAT WAS FOUND BEFORE.
4413
4414  014166  012767  014166  001244  PRG4:   MOV     #PRG4,PRG.NO    ;SET FOR POWER FAIL
4415  014174  005067  165016           CLR     LSTERR          ;PREPARE ERROR CONDITIONS
4416  014200  012706  001100           MOV     #STACK,SP       ;SET THE STACK POINTER
4417  014204  012767  020000  165006   MOV     #20000,ICOUNT   ;SET ITERATION COUNT TO 20000(8)
4418  014212  005767  000544           TST     FLAG4           ;HAVE I BEEN HERE BEFOR??
4419  014216  001106                   BNE     TAG.A           ;BR IF NOT FIRST TIME HERE.
4420  014220  005167  000536           COM     FLAG4           ;SET THE FLAG
4421  014224  012705  000002           MOV     #2,R5           ;SET R5 FOR SWITCH 1
4422  014230  012704  014752           MOV     #LOC1,R4        ;SET STORAGE LOCATION
4423  014234  012737  015442  000004   MOV     #NO.TRAP,@#4    ;SET FOR TIME OUT TRAP
4424  014242  012767  000001  164754   MOV     #1,TEMP3        ;SET FOR MESSAGE ON CHANNEL NO.
4425  014250  104400  014764   1$:     TYPE    .MCHAN          ;TYPE MESSAGE ABOUT CHANNEL
4426  014254  104400  015315           TYPE    .MSPACE
4427
4428  014260  016746  164740           MOV     TEMP3,-(SP)
4429  014264  104402                   TYPOC
4430  014266  104400  015315           TYPE    ,MSPACE
4431
4432
4433  014272  104400  015000           TYPE    ,MACTV          ;TYPE REST OF MESSAGE
4434  014276  104400  015013   2$:     TYPE    ,MADD1          ;TYPE ADDRESS MESSAGE
4435  014302  012700  173024           MOV     #173024,R0
4436  014306  005037  173024           CLR     @#173024
4437  014312  010537  173024           MOV     R5,@#173024     ;WRITE ROM WITH SWITCH
4438  014316  000240                   NOP                     ;WAIT ONE INSTR. TIME
4439  014320  012706  001100   3$:     MOV     #STACK,SP       ;SET THE STACK POINTER
4440  014324  012700  173024           MOV     #173024,R0      ;SET FOR ERROR MESSAGE
4441  014330  012737  015442  000004   MOV     #NO.TRAP,@#4    ;SET FOR NO MORE TRAPS
4442  014336  013767  173024  164662   MOV     @#173024,TEMP4  ;READ THE ADDRESS
4443
4444  014344  104400  015315           TYPE    ,MSPACE
4445  014350  016746  164652           MOV     TEMP4,-(SP)
4446  014354  104402                   TYPOC
4447  014356  013724  173024           MOV     @#173024,(R4)+  ;STORE THE INFORMATION FOUND
4448  014362  104400  015047           TYPE    ,MADD2          ;TYPE THE SECOND ADDRESS MSG
4449  014366  012700  173224           MOV     #173224,R0      ;SET FOR ERROR CONDITION.
4450  014372  013767  173224  164626   MOV     @#173224,TEMP4  ;STORE ROM DATA
4451  014400  104400  015315           TYPE    ,MSPACE
4452
4453  014404  016746  164616           MOV     TEMP4,-(SP)
4454  014410  104402                   TYPOC
4455
4456  014412  005267  164606           INC     TEMP3           ;GET READY FOR NEXT SWITCH SETTING
4457  014416  000241                   CLC                     ;CLEAR THE CARRY BIT
```

```
4458  014420  006105                        ROL     R5              ;UPDATE R5
4459  014422  022705  000040                CMP     #40,R5          ;ALL SIMULATED SWITCHS DONE?
4460  014426  001310                        BNE     1$              ;BR IF NOT ALL DONE
4461  014430  000167  177532                JMP     PRG4            ;JMP    AND DO TEST AGAIN WITH OUT TYPE OUT
4462
4463  014434  012703  000002        TAG.A:  MOV     #2,R3           ;SIMULATE SWITCH 1
4464  014440  012704  014752                MOV     #LOC1,R4        ;GET LOCATION WHERE DATA IS STORED
4465  014444  012737  015442  000004  1$:   MOV     #NO.TRAP,@#4    ;PREPARE FOR TIME OUT TRAP
4466  014452  005037  173024                CLR     @#173024
4467  014456  010337  173024                MOV     R3,@#173024     ;WRITE THE ROM
4468  014462  000240                        NOP                     ;WAIT ONE INSTR. TIME
4469  014464  012706  001100        2$:     MOV     #STACK,SP       ;SET THE STACK POINTER.
4470  014470  012737  015442  000004        MOV     #NO.TRAP,@#4    ;SET FOR NO MORE TRAPS.
4471  014476  012700  173024                MOV     #173024,R0      ;SET FOR ERROR MESSAGE
4472  014502  011401                        MOV     (R4),R1         ;SET FOR COMPARISON
4473  014504  013705  173024                MOV     @#173024,R5     ;GET THE DATA FROM THE ROM
4474  014510  012767  014752  164406        MOV     #LOC1,$GDDAT
4475  014516  012767  173024  164402        MOV     #173024,$BDDAT
4476  014524  016767  000222  164472        MOV     LOC1,TEMP3
4477  014532  013767  173024  164466        MOV     @#173024,TEMP4
4478  014540  020105                        CMP     R1,R5           ;IS THE DATA THE SAME??
4479  014542  001401                        BEQ     30$             ;BR IF GOOD DATA.
4480  014544  104001                        ERROR   1               ;ERROR. DATA READ FIRST TIME NOT THE SAME
4481  014546  012700  173224        30$:    MOV     #173224,R0      ;SET FOR ERROR MESSAGE
4482  014552  013705  173224                MOV     @#173224,R5     ;READ THE ROM
4483  014556  012767  173224  164342        MOV     #173224,$BDDAT
4484  014564  013767  173224  164434        MOV     @#173224,TEMP4
4485  014572  020105                        CMP     R1,R5           ;IS THE DATA THE SAME?
4486  014574  001401                        BEQ     31$             ;BR IF GOOD DATA
4487  014576  104001                        ERROR   1               ;ERROR. DATA NOT THE SAME AS BEFORE.
4488  014600  005724                31$:    TST     (R4)+           ;UPDATE DATA POINTER.
4489  014602  000241                        CLC                     ;CLEAR THE CARRY BIT
4490  014604  006103                        ROL     R3              ;UPDATE THE SIMULATED SWITHCH SETTING
4491  014606  022703  000040                CMP     #40,R3          ;HAVE ALL SETTING BEEN DONE
4492  014612  001314                        BNE     1$              ;BR IF NOT DONE
4493  014614  005367  164400                DEC     ICOUNT          ;ITERATION COUNT DONE
4494  014620  001305                        BNE     TAG.A           ;BR IF NOT DONE
4495  014622  012737  177777  173224        MOV     #-1,@#173224    ;WRITE SECOND TRAP VECTOR WITH -1
4496  014630  005037  173024                CLR     @#173024        ;ZERO THE FIRST VECTOR
4497  014634  012700  173024                MOV     #173024,R0      ;SET FOR TYPE OUT IF ERROR
4498  014640  016701  000106                MOV     LOC1,R1         ;SET FOR TYPE OUT ROUTINE
4499  014644  013705  173024                MOV     @#173024,R5     ;SAME AS ABOVE
4500  014650  012767  173024  164250        MOV     #173024,$BDDAT
4501  014656  013767  173024  164342        MOV     @#173024,TEMP4
4502  014664  020105                        CMP     R1,R5           ;IS DEFAULT LINE SELECTED =TO LINE 1
4503  014666  001401                        BEQ     32$             ;BR IF DEFAULT EQUALS LINE 1
4504  014670  104001                        ERROR   1               ;DATA NOT EQUAL TO LINE I
4505  014672  012737  177777  173024  32$:  MOV     #-1,@#173024    ;WRITE A -1 TO FIRST VECTOR
4506  014700  005037  173224                CLR     @#173224        ;ZERO SECOND VECTOR
4507  014704  012700  173224                MOV     #173224,R0      ;SET FOR TYPE OUT IF ERROR
4508  014710  016701  000036                MOV     LOC1,R1         ;GET DATA
4509  014714  013705  173224                MOV     @#173224,R5     ;READ ROM
4510  014720  012767  173224  164200        MOV     #173224,$BDDAT
4511  014726  013767  173224  164272        MOV     @#173224,TEMP4
4512
4513  014734  020105                        CMP     R1,R5           ;IS LINE 1 DEFAULT LINE
```

```
 4514  014736  001401                        BEQ     33$             ;BR IF OK
 4515  014740  104001                        ERROR   1               ;ERROR LINE 1 NOT DEFAULT LINE
 4516  014742  004767  000534        33$:    JSR     PC,EOP          ;TYPE END MESSAGE.
 4517  014746  000167  177214                JMP     PRG4            ;GOTO PROGRAM 4 AGAIN
 4518
 4519  014752  000000                LOC1:   0
 4520  014754  000000                LOC2:   0
 4521  014756  000000                LOC3:   0
 4522  014760  000000                LOC4:   0
 4523  014762  000000                FLAG4:  0
```

```
4524  014764  005015  041412  040510  MCHAN:   .ASCIZ <15><12><12>/CHANNEL /
      015000  041501  044524  040526  MACTV:   .ASCIZ/ACTIVATED./
      015013     015  040412  042104  MADD1:   .ASCIZ  <15><12>/ADDRESS 773024 CONTAINS: /
      015047     015  040412  042104  MADD2:   .ASCIZ  <15><12>/ADDRESS 773224 CONTAINS: /
              015104                  .EVEN
4525
4526  015104  005015  051120  043517  MSG3:    .ASCIZ <15><12>/PROGRAM NO. (1,2,3,4)  /
      015136  006414  016412  077437  MSG4:    .ASCIZ <14><15><12><35><37><177><177><177>/SOFTWARE MAP:/
      015164  005015  020012  054105  MSG5:    .ASCIZ  <15><12><12>/ EXTENDED SOFTWARE MAP:/
      015217     015  025012     000  MASTER:  .ASCIZ <15><12>/*/
      015223     007  006407  042412  M.END:   .ASCIZ <7><7><15><12>/END PASS BM873-Y/
      015250                          MFAIL:
      015250  005015  053520  020122          .ASCII  <15><12>/PWR UP AFTER/
      015266  005015  042522  046101          .ASCIZ  <15><12>/REAL PWR FAIL/
      015306  000044                  M.DOL:   .ASCIZ /$/
      015310  005015  037477     000  M.QM:    .ASCIZ <15><12>/??/
      015315     040  000040          MSPACE:  .ASCIZ  /  /
      015320  020040  000040          SPACE3:  .ASCIZ  /  /
      015324  005015     000          MCRLF:   .ASCIZ <15><12>
      015327     012     000          MLF:     .ASCIZ <12>
              015332                  .EVEN
4527
```

```
 4528  015332  005067  163660          .PFAIL: CLR     LSTERR
 4529  015336  013746  000004                  MOV     @#4,-(SP)
 4530  015342  012737  015372  000004          MOV     #1$,@#4
 4531  015350  005737  173000                  TST     @#173000        ;IS THIS PF REAL?
 4532  015354  000240                          NOP     ;TRAP IS CAUSED BY LOADER
 4533  015356  012737  015402  000024          MOV     #PWR.UP,@#24    ;ITS REAL. PREPARE FOR PWR UP
 4534  015364  012637  000004                  MOV     (SP)+,@#4
 4535  015370  000000                          HALT
 4536  015372  005726                  1$:     TST     (SP)+   ;POP THE STACK.
 4537  015374  012637  000004                  MOV     (SP)+,@#4
 4538  015400  000000                          HALT    ;HARDWARE ERROR.  BOOT DIDN'T FORCE
 4539                          ;HIGH ADDR LINES AND LOAD BUTTON WAS ACTIVATED
 4540  015402  012737  015332  000024  PWR.UP: MOV     #.PFAIL,@#24
 4541  015410  012706  001100                  MOV     #STACK,SP
 4542  015414  005000                          CLR     R0              ;SET DELAY
 4543  015416  062700  000001          1$:     ADD     #1,R0           ;WAIT FOR TTY
 4544  015422  001375                          BNE     1$
 4545  015424  104400  015250                  TYPE    ,MFAIL          ;TYPE FAILED.
 4546  015430  005067  162342                  CLR     PS              ;SET STATUS TO ZERO
 4547  015434  000177  000000                  JMP     @PRG.NO
 4548  015440  000000                  PRG.NO: 0
 4549  015442                          NO.TRAP:
 4550  015442  011667  000032                  MOV     (SP),XSTORE
 4551  015446  032716  100000                  BIT     #BIT15,(SP)
 4552  015452  001410                          BEQ     1$
 4553  015454  011600                          MOV     (SP),R0
 4554  015456  104004                          ERROR   4
 4555  015460  012706  001100                  MOV     #STACK,SP
 4556  015464  005067  162306                  CLR     PS
 4557  015470  000177  177744                  JMP     @PRG.NO
 4558  015474  104003                  1$:     ERROR   3
 4559  015476  000002                          RTI
 4560  015500  000000                  XSTORE: 0
 4561
 4562  015502  005067  163510          EOP:    CLR     LSTERR
 4563  015506  104400  015223                  TYPE    ,M.END
 4564  015512  104400  015562                  TYPE    ,VERSON
 4565  015516  013701  000042                  MOV     @#42,R1
 4566  015522  001416                          BEQ     X1
 4567  015524  022767  011474  177706          CMP     #PRG1,PRG.NO
 4568  015532  001002                          BNE     .+6
 4569  015534  000167  176426                  JMP     PRG4
 4570  015540  013701  000042                  MOV     @#42,R1
 4571  015544  001405                          BEQ     X1
 4572  015546  000005                          RESET
 4573  015550                          $ENDAD:
 4574  015550  004711                  LOGIC:  JSR     PC,(R1)
 4575  015552  000240                          NOP
 4576  015554  000240                          NOP
 4577  015556  000240                          NOP
 4578  015560  000207                  X1:     RTS     PC
 4579  015562  000101                  VERSON: 101                     ;SEVEN BIT ASCII FOR DEFAULT "A"
```

```
4580  015564  005015  041520  020072  MERRPC: .ASCIZ  <15><12>/PC: /
4581  015572     000
4582          015574                  .EVEN
4583                                  .MCALL .$EOP, .$TYPE, .$TYPOCT, .$POWER, .$READ
4584                                  ;;*****************************************************************
4585
4586                                  .SBTTL  TYPE ROUTINE
4587
4588                                  ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
4589                                  ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
4590                                  ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
4591                                  ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
4592                                  ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
4593                                  ;*
4594                                  ;*CALL:
4595                                  ;*1) USING A TRAP INSTRUCTION
4596                                  ;*      TYPE    ,MESADR                 ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
4597                                  ;*OR
4598                                  ;*      TYPE
4599                                  ;*      MESADR
4600                                  ;*
4601                                  ;*2) USING A JSR INSTRUCTION
4602                                  ;*      MOV     PS,-(SP)                ;PUSH PROCESSOR STATUS WORD ON THE STACK
4603                                  ;*      JSR     PC,$TYPE                ;CALL TYPE ROUTINE
4604                                  ;*      MESADDR                         ;FIRST ADRESS OF MESSAGE
4605
4606  015574  105767  163351  $TYPE:  TSTB    $TPFLG                  ;IS THERE A TERMINAL?
4607  015600  100002                  BPL     1$                      ;BR IF YES
4608  015602  000000                  HALT                            ;HALT HERE IF NO TERMINAL
4609  015604  000407                  BR      3$                      ;LEAVE
4610  015606  010046          1$:     MOV     R0,-(SP)                ;SAVE R0
4611  015610  017600  000002          MOV     @2(SP),R0               ;GET ADDRESS OF ASCIZ STRING
4612  015614  112046          2$:     MOVB    (R0)+,-(SP)             ;PUSH CHARACTER TO BE TYPED ONTO STACK
4613  015616  001005                  BNE     4$                      ;BR IF IT ISN'T THE TERMINATOR
4614  015620  005726                  TST     (SP)+                   ;IF TERMINATOR POP IT OFF THE STACK
4615  015622  012600                  MOV     (SP)+,R0                ;RESTORE R0
4616  015624  062716  000002  3$:     ADD     #2,(SP)                 ;ADJUST RETURN PC
4617  015630  000002                  RTI                             ;RETURN
4618  015632  004767  000026  4$:     JSR     PC,7$                   ;GO TYPE THIS CHARACTER
4619  015636  126726  163306  5$:     CMPB    $FILLC,(SP)+            ;IS IT TIME FOR FILLER CHARS.?
4620  015642  001364                  BNE     2$                      ;IF NO GO GET NEXT CHAR.
4621  015644  016746  163276          MOV     $NULL,-(SP)             ;GET # OF FILLER CHARS. NEEDED
4622                                                                  ;AND THE NULL CHAR.
4623  015650  105366  000001  6$:     DECB    1(SP)                   ;DOES A NULL NEED TO BE TYPED?
4624  015654  002770                  BLT     5$                      ;BR IF NO--GO POP THE NULL OFF OF STACK
4625  015656  004767  000002          JSR     PC,7$                   ;GO TYPE A NULL
4626  015662  000772                  BR      6$                      ;LOOP
4627  015664  105777  163252  7$:     TSTB    @$TPS                   ;WAIT UNTIL PRINTER IS READY
4628  015670  100375                  BPL     7$
4629  015672  116677  000002  163244  MOVB    2(SP),@$TPB             ;LOAD CHAR TO BE TYPED INTO DATA REG.
4630  015700  000207                  RTS     PC
4631                                  ;;*****************************************************************
4632
4633                                  .SBTTL  TTY INPUT ROUTINE
4634
4635                                  ;*INPUT A SINGLE CHARACTER FROM THE TTY
```

```
4636                                    ;*CALL:
4637                                    ;*      RDCHR                   ;INPUT A SINGLE CHARACTER FROM THE TTY
4638                                    ;*      RETURN HERE             ;CHARACTER IS ON THE STACK
4639
4640
4641  015702  011646                    $RDCHR: MOV     (SP),-(SP)      ;PUSH DOWN THE PC
4642  015704  016666  000004  000002            MOV     4(SP),2(SP)     ;SAVE THE PS
4643  015712  105777  163220            1$:     TSTB    @$TKS           ;WAIT FOR
4644  015716  100375                            BPL     1$              ;A CHARACTER
4645  015720  117766  163214  000004            MOVB    @$TKB,4(SP)     ;READ THE TTY
4646  015726  042766  177600  000004            BIC     #^C<177>,4(SP)  ;GET RID OF JUNK IF ANY
4647  015734  000002                            RTI                     ;GO BACK TO USER
4648                                    ;;**************************************************************
4649                                    ;*INPUT A STRING FROM THE TTY
4650                                    ;*CALL:
4651                                    ;*      RDLIN                   ;INPUT A STRING FROM THE TTY
4652                                    ;*      RETURN HERE             ;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
4653                                    ;*                              ;TERMINATOR WILL BE A BYTE OF ALL 0'S
4654
4655  015736  010346                    $RDLIN: MOV     R3,-(SP)        ;SAVE R3
4656  015740  012703  016044            1$:     MOV     #$TTYIN,R3      ;GET ADDRESS
4657  015744  022703  016054            2$:     CMP     #$TTYIN+8.,R3   ;BUFFER FULL?
4658  015750  101405                            BLOS    4$              ;BR IF YES
4659  015752  104410                            RDCHR                   ;GO READ ONE CHARACTER FROM THE TTY
4660  015754  112613                            MOVB    (SP)+,(R3)      ;GET CHARACTER
4661  015756  122713  000177                    CMPB    #177,(R3)       ;IS IT A RUBOUT
4662  015762  001003                            BNE     3$              ;SKIP IF NOT
4663  015764  104400  001152            4$:     TYPE    ,$QUES          ;TYPE A '?'
4664  015770  000763                            BR      1$              ;CLEAR THE BUFFER AND LOOP
4665  015772  111367  000044            3$:     MOVB    (R3),8$         ;ECHO THE CHARACTER
4666  015776  104400  016042                    TYPE    ,8$
4667  016002  122723  000015                    CMPB    #15,(R3)+       ;CHECK FOR RETURN
4668  016006  001356                            BNE     2$              ;LOOP IF NOT RETURN
4669  016010  105063  177777                    CLRB    -1(R3)          ;CLEAR RETURN (THE 15)
4670  016014  104400  001154                    TYPE    ,$LF            ;TYPE A LINE FEED
4671  016020  012603                            MOV     (SP)+,R3        ;RESTORE R3
4672  016022  011646                            MOV     (SP),-(SP)      ;ADJUST THE STACK AND PUT ADDRESS OF THE
4673  016024  016666  000004  000002            MOV     4(SP),2(SP)     ;       FIRST ASCII CHARACTER ON IT
4674  016032  012766  016044  000004            MOV     #$TTYIN,4(SP)
4675  016040  000002                            RTI                     ;RETURN
4676  016042     000            8$:     .BYTE   0                       ;STORAGE FOR ASCII CHAR. TO TYPE
4677  016043     000                    .BYTE   0                       ;TERMINATOR
4678  016044  000010            $TTYIN: .BLKB   8.                      ;RESERVE 8 BYTES FOR TTY INPUT
4679                                    ;;**************************************************************
4680
4681                                    .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
4682
4683                                    ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
4684                                    ;*CALL:
4685                                    ;*      MOV     NUM,-(SP)       ;NUMBER TO BE TYPED
4686                                    ;*      TYPOS                   ;CALL FOR TYPEOUT
4687                                    ;*      .BYTE   N               ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
4688                                    ;*      .BYTE   M               ;M=1 OR 0
4689                                    ;*                                      ;1=TYPE LEADING ZEROS
4690                                    ;*                                      ;0=SUPPRESS LEADING ZEROS
4691                                    ;*
```

```
                                        ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                        ;*$TYPOS OR $TYPOC
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
                                        ;*      TYPON                           ;CALL FOR TYPEOUT
                                        ;*
                                        ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)               ;NUMBER TO BE TYPED
                                        ;*      TYPOC                           ;CALL FOR TYPEOUT

016054  017646  000000          $TYPOS: MOV     @(SP),-(SP)             ;PICKUP THE MODE
016060  116667  000001  000211          MOVB    1(SP),$OFILL            ;LOAD ZERO FILL SWITCH
016066  112667  000207                  MOVB    (SP)+,$OMODE+1          ;NUMBER OF DIGITS TO TYPE
016072  062716  000002                  ADD     #2,(SP)                 ;ADJUST RETURN ADDRESS
016076  000406                          BR      $TYPON
016100  112767  000001  000171  $TYPOC: MOVB    #1,$OFILL               ;SET THE ZERO FILL SWITCH
016106  112767  000006  000165          MOVB    #6,$OMODE+1             ;SET FOR SIX(6) DIGITS
016114  112767  000005  000154  $TYPON: MOVB    #5,$OCNT                ;SET THE ITERATION COUNT
016122  010346                          MOV     R3,-(SP)                ;SAVE R3
016124  010446                          MOV     R4,-(SP)                ;SAVE R4
016126  010546                          MOV     R5,-(SP)                ;SAVE R5
016130  116704  000145                  MOVB    $OMODE+1,R4             ;GET THE NUMBER OF DIGITS TO TYPE
016134  005404                          NEG     R4
016136  062704  000006                  ADD     #6,R4                   ;SUBTRACT IT FOR MAX. ALLOWED
016142  110467  000132                  MOVB    R4,$OMODE               ;SAVE IT FOR USE
016146  116704  000125                  MOVB    $OFILL,R4               ;GET THE ZERO FILL SWITCH
016152  016605  000012                  MOV     12(SP),R5               ;PICKUP THE INPUT NUMBER
016156  005003                          CLR     R3                      ;CLEAR THE OUTPUT WORD
016160  006105                  1$:     ROL     R5                      ;ROTATE MSB INTO "C"
016162  000404                          BR      3$                      ;GO DO MSB
016164  006105                  2$:     ROL     R5                      ;FORM THIS DIGIT
016166  006105                          ROL     R5
016170  006105                          ROL     R5
016172  010503                          MOV     R5,R3
016174  006103                  3$:     ROL     R3                      ;GET LSB OF THIS DIGIT
016176  105367  000076                  DECB    $OMODE                  ;TYPE THIS DIGIT?
016202  100016                          BPL     7$                      ;BR IF NO
016204  042703  177770                  BIC     #177770,R3              ;GET RID OF JUNK
016210  001002                          BNE     4$                      ;TEST FOR 0
016212  005704                          TST     R4                      ;SUPPRESS THIS 0?
016214  001403                          BEQ     5$                      ;BR IF YES
016216  005204                  4$:     INC     R4                      ;DON'T SUPPRESS ANYMORE 0'S
016220  052703  000060                  BIS     #'0,R3                  ;MAKE THIS DIGIT ASCII
016224  052703  000040          5$:     BIS     #' ,R3                  ;MAKE ASCII IF NOT ALREADY
016230  110367  000040                  MOVB    R3,8$                   ;SAVE FOR TYPING
016234  104400  016274                  TYPE    ,8$                     ;GO TYPE THIS DIGIT
016240  105367  000032          7$:     DECB    $OCNT                   ;COUNT BY 1
016244  003347                          BGT     2$                      ;BR IF MORE TO DO
016246  002402                          BLT     6$                      ;BR IF DONE
016250  005204                          INC     R4                      ;INSURE LAST DIGIT ISN'T A BLANK
016252  000744                          BR      2$                      ;GO DO THE LAST DIGIT
016254  012605                  6$:     MOV     (SP)+,R5                ;RESTORE R5
016256  012604                          MOV     (SP)+,R4                ;RESTORE R4
016260  012603                          MOV     (SP)+,R3                ;RESTORE R3
016262  016666  000002  000004          MOV     2(SP),4(SP)             ;SET THE STACK FOR RETURNING
```

```
4748  016270  012616                          MOV     (SP)+,(SP)
4749  016272  000002                          RTI                             ;RETURN
4750  016274     000                  8$:     .BYTE   0                       ;STORAGE FOR ASCII DIGIT
4751  016275     000                          .BYTE   0                       ;TERMINATOR FOR TYPE ROUTINE
4752  016276     000                  $OCNT:  .BYTE   0                       ;OCTAL DIGIT COUNTER
4753  016277     000                  $OFILL: .BYTE   0                       ;ZERO FILL SWITCH
4754  016300  000000                  $OMODE: 0                               ;NUMBER OF DIGITS TO TYPE
4755                                  ;;**************************************************************
4756
4757                                  .SBTTL  SCOPE HANDLER ROUTINE
4758
4759                                  ;*SW14=1        LOOP ON TEST
4760                                  ;*THE TEST NUMBER ($TSTNM) IS INCREMENTED AND DISPLAYED IN DISPLAY<7:0>
4761                                  ;*AND THE ERROR FLAG ($ERFLG) IS DISPLAYED IN DISPLAY<15:08>
4762
4763  016302                          $SCOPE:
4764  016302  006137  177570                  ROL     @#SWR                   ;LOOP ON PRESENT TEST?
4765  016306  100425                          BMI     $OVER                   ;YES IF SW14=1
4766                                          ;#####START OF CODE FOR THE XOR TESTER#####
4767  016310  000416                  $XTSTR: BR      6$                      ;IF RUNNING ON THE "XOR" TESTER CHANGE
4768                                                                          ;THIS INSTRUCTION TO A "NOP" (NOP=240)
4769  016312  013746  000004                  MOV     @#ERRVEC,-(SP)          ;SAVE THE CONTENTS OF THE ERROR VECTOR
4770  016316  012737  016336  000004          MOV     #5$,@#ERRVEC            ;SET FOR TIMEOUT
4771  016324  005737  177060                  TST     @#177060                ;TIME OUT ON XOR?
4772  016330  012637  000004                  MOV     (SP)+,@#ERRVEC          ;RESTORE THE ERROR VECTOR
4773  016334  000404                          BR      $SVLAD                  ;GO TO THE NEXT TEST
4774  016336  022626                  5$:     CMP     (SP)+,(SP)+             ;CLEAR THE STACK AFTER A TIME OUT
4775  016340  012637  000004                  MOV     (SP)+,@#ERRVEC          ;RESTORE THE ERROR VECTOR
4776  016344  000406                          BR      $OVER                   ;LOOP ON THE PRESENT TEST
4777  016346                          6$:     ;#####END OF CODE FOR THE XOR TESTER#####
4778  016346  105267  162530          $SVLAD: INCB    $TSTNM                  ;COUNT TEST NUMBERS
4779  016352  011667  162530                  MOV     (SP),$LPADR             ;SAVE SCOPE LOOP ADDRESS
4780  016356  105067  162521                  CLRB    $ERFLG                  ;ZERO THE ERROR FLAG
4781  016362  016737  162514  177570  $OVER:  MOV     $TSTNM,@#DISPLAY        ;DISPLAY TEST NUMBER
4782  016370  016716  162512                  MOV     $LPADR,(SP)             ;FUDGE RETURN ADDRESS
4783  016374  000002                          RTI                             ;FIXES PS
4784                                  ;;**************************************************************
4785
4786                                  .SBTTL  ERROR HANDLER ROUTINE
4787
4788                                  ;*SW15=1        HALT ON ERROR
4789                                  ;*SW13=1        INHIBIT ERROR TYPEOUTS
4790                                  ;*GO TO $ERRTYP ON ERROR
4791
4792  016376                          $ERROR:
4793  016376  105267  162501          7$:     INCB    $ERFLG                  ;SET THE ERROR FLAG
4794  016402  001775                          BEQ     7$                      ;DON'T LET THE FLAG GO TO ZERO
4795  016404  016737  162472  177570          MOV     $TSTNM,@#DISPLAY        ;DISPLAY TEST NUMBER AND ERROR FLAG
4796  016412  005267  162474                  INC     $ERTTL                  ;INC THE ERROR COUNT
4797  016416  011667  162474                  MOV     (SP),$ERRPC             ;GET ADDRESS OF ERROR INSTRUCTION
4798  016422  162767  000002  162466          SUB     #2,$ERRPC
4799  016430  117767  162462  162456          MOVB    @$ERRPC,$ITEMB          ;STRIP AND SAVE THE ERROR ITEM CODE
4800  016436  032737  020000  177570          BIT     #SW13,@#SWR             ;SKIP TYPEOUT IF SET
4801  016444  001004                          BNE     2$                      ;SKIP TYPEOUTS
4802  016446  004737  016470                  JSR     PC,@#$ERRTYP            ;GO TO USER ERROR ROUTINE
4803  016452  104400  001153                  TYPE    ,$CRLF
```

```
 4804  016456  005737  177570          2$:     TST     @#SWR                   ;HALT ON ERROR
 4805  016462  100001                          BPL     3$                      ;SKIP IF CONTINUE
 4806  016464  000000                          HALT                            ;HALT ON ERROR!
 4807  016466  000002                  3$:     RTI                             ;RETURN
 4808                                  ;;**************************************************************
 4809
 4810                                  .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE
 4811
 4812                                  ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
 4813                                  ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
 4814                                  ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
 4815
 4816  016470                          $ERRTYP:
 4817  016470  104400  001153                TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
 4818  016474  010046                          MOV     R0,-(SP)                ;SAVE R0
 4819  016476  005000                          CLR     R0                      ;PICKUP THE ITEM INDEX
 4820  016500  153700  001114                BISB    @#$ITEMB,R0
 4821  016504  001004                          BNE     1$                      ;IF ITEM NUMBER IS ZERO, JUST
 4822                                                                          ;TYPE THE PC OF THE ERROR
 4823  016506  016746  162404                MOV     $ERRPC,-(SP)            ;SAVE $ERRPC FOR TYPEOUT
 4824                                                                          ;ERROR ADDRESS
 4825  016512  104402                          TYPOC                           ;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4826  016514  000426                          BR      6$                      ;GET OUT
 4827  016516  005300                  1$:     DEC     R0                      ;ADJUST THE INDEX SO THAT IT WILL
 4828  016520  006300                          ASL     R0                      ;       WORK FOR THE ERROR TABLE
 4829  016522  006300                          ASL     R0
 4830  016524  006300                          ASL     R0
 4831  016526  062700  001156                ADD     #$ERRTB,R0              ;FORM TABLE POINTER
 4832  016532  012067  000004                MOV     (R0)+,2$                ;PICKUP "ERROR MESSAGE" POINTER
 4833  016536  001404                          BEQ     3$                      ;SKIP TYPEOUT IF NO POINTER
 4834  016540  104400                          TYPE                            ;TYPE THE "ERROR MESSAGE"
 4835  016542  000000                  2$:     .WORD   0                       ;"ERROR MESSAGE" POINTER GOES HERE
 4836  016544  104400  001153                TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
 4837  016550  012067  000004                MOV     (R0)+,4$                ;PICKUP "DATA HEADER" POINTER
 4838  016554  001404                          BEQ     5$                      ;SKIP TYPEOUT IF 0
 4839  016556  104400                          TYPE                            ;TYPE THE "DATA HEADER"
 4840  016560  000000                  4$:     .WORD   0                       ;"DATA HEADER" POINTER GOES HERE
 4841  016562  104400  001153                TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
 4842  016566  011000                  5$:     MOV     (R0),R0                 ;PICKUP "DATA TABLE" POINTER
 4843  016570  001004                          BNE     7$                      ;GO TYPE THE DATA
 4844  016572  012600                  6$:     MOV     (SP)+,R0                ;RESTORE R0
 4845  016574  104400  001153                TYPE    ,$CRLF                  ;"CARRIAGE RETURN" & "LINE FEED"
 4846  016600  000207                          RTS     PC                      ;RETURN
 4847  016602                          7$:
 4848  016602  013046                          MOV     @(R0)+,-(SP)            ;SAVE @(R0)+ FOR TYPEOUT
 4849  016604  104402                          TYPOC                           ;GO TYPE--OCTAL ASCII(ALL DIGITS)
 4850  016606  005710                          TST     (R0)                    ;IS THERE ANOTHER NUMBER?
 4851  016610  001770                          BEQ     6$                      ;BR IF NO
 4852  016612  104400  016620                TYPE    ,8$                     ;TYPE TWO(2) SPACES
 4853  016616  000771                          BR      7$                      ;LOOP
 4854  016620  020040      000         8$:     .ASCIZ  /  /                    ;TWO(2) SPACES
 4855          016624                          .EVEN
 4856                                  ;;**************************************************************
 4857
 4858                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
 4859
```

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

016624  011646                  $RDOCT: MOV     (SP),-(SP)              ;PROVIDE SPACE FOR THE
016626  016666  000004  000002          MOV     4(SP),2(SP)             ;INPUT NUMBER
016634  010046                          MOV     R0,-(SP)                ;PUSH R0 ON STACK
016636  010146                          MOV     R1,-(SP)                ;PUSH R1 ON STACK
016640  010246                          MOV     R2,-(SP)                ;PUSH R2 ON STACK
016642  104412                  1$:     RDLIN                           ;READ AN ASCIZ LINE
016644  012600                          MOV     (SP)+,R0                ;GET ADDRESS OF 1ST CHARACTER
016646  005001                          CLR     R1                      ;CLEAR DATA WORD
016650  005002                          CLR     R2
016652  112046                  2$:     MOVB    (R0)+,-(SP)             ;PICKUP THIS CHARACTER
016654  001412                          BEQ     3$                      ;IF ZERO GET OUT
016656  006301                          ASL     R1                      ;*2
016660  006102                          ROL     R2
016662  006301                          ASL     R1                      ;*4
016664  006102                          ROL     R2
016666  006301                          ASL     R1                      ;*8
016670  006102                          ROL     R2
016672  042716  177770                  BIC     #^C7,(SP)               ;STRIP THE ASCII JUNK
016676  062601                          ADD     (SP)+,R1                ;ADD IN THIS DIGIT
016700  000764                          BR      2$                      ;LOOP
016702  005726                  3$:     TST     (SP)+                   ;CLEAN TERMINATOR FROM STACK
016704  010166  000012                  MOV     R1,12(SP)               ;SAVE THE RESULT
016710  010267  000010                  MOV     R2,$HIOCT
016714  012602                          MOV     (SP)+,R2                ;POP STACK INTO R2
016716  012601                          MOV     (SP)+,R1                ;POP STACK INTO R1
016720  012600                          MOV     (SP)+,R0                ;POP STACK INTO R0
016722  000002                          RTI                             ;RETURN
016724  000000                  $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.

016726  010046                  $TRAP:  MOV     R0,-(SP)                ;SAVE R0
016730  016600  000002                  MOV     2(SP),R0                ;GET TRAP ADDRESS
016734  005740                          TST     -(R0)                   ;BACKUP BY 2
016736  111000                          MOVB    (R0),R0                 ;GET RIGHT BYTE OF TRAP
016740  016000  016746                  MOV     $TRPAD(R0),R0           ;INDEX TO TABLE
016744  000200                          RTS     R0                      ;GO TO ROUTINE


                                        .SBTTL  TRAP TABLE

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

                                        ;       ROUTINE
```

```
4916                                        ;       -------
4917  016746                                $TRPAD:
4918  016746  015574                                $TYPE           ;CALL=TYPE      TRAP+0(104400)  TTY TYPEOUT ROUTINE
4919  016750  016100                                $TYPOC          ;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING
4920  016752  016054                                $TYPOS          ;CALL=TYPOS     TRAP+4(104404)  TYPE OCTAL NUMBER (NO LEADING ZE
4921  016754  016114                                $TYPON          ;CALL=TYPON     TRAP+6(104406)  TYFE OCTAL NUMBER (AS PER LAST C
4922  016756  015702                                $RDCHR          ;CALL=RDCHR     TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
4923  016760  015736                                $RDLIN          ;CALL=RDLIN     TRAP+12(104412) TTY TYPEIN STRING ROUTINE
4924  016762  016624                                $RDOCT          ;CALL=RDOCT     TRAP+14(104414) READ AN OCTAL NUMBER FROM TTY
4925  016764  005015  047522  020115  EM1:    .ASCIZ  <15><12>/ROM READ DATA COMPARISON ERROR./
      017026  005015  051127  052111  EM2:    .ASCIZ  <15><12>/WRITING ROM FAILED TO TRAP./
      017064  005015  047125  054105  EM3:    .ASCIZ  <15><12>/UNEXP TRAP WHILE READING ROM./
      017124  005015  040506  040524  EM4:    .ASCIZ  <15><12>/FATAL TRAP.  ROM PC ON STACK./
      017164  005015  041520  020040  DH1:    .ASCII  <15><12>/PC      SOFT    ROM/
      017211     015  040412  042104          .ASCIZ  <15><12>/ADDRESS ADDRESS ADDRESS EXPECTED FOUND /
      017263     015  050012  004503  DH2:    .ASCII  <15><12>/PC     ROM/
      017273     015  040412  042104          .ASCIZ  <15><12>/ADDRESS ADDRESS/
      017315     015  050012  020103  DH3:    .ASCII  <15><12>/PC OF   PROGRAM /
      017337     015  052012  040522          .ASCIZ  <15><12>/TRAP    ADDRESS/
                017362                       .EVEN
4926  017362  001116  001124  001126  DT1:    .WORD   $ERRPC,$GDDAT,$BDDAT,TEMP3,TEMP4,0
4927  017370  001224  001226  000000
4928
4929  017376  001116  001226  000000  DT2:    .WORD   $ERRPC,TEMP4,0
4930  017404  001116  015500  000000  DT3:    .WORD   $ERRPC,XSTORE,0
4931             017452                       .=.+40
4932  017452                             CORMAX:
4933             000001                       .END
```

```
ADDRES  014164          4213*  4232   4251*  4262*  4401#
ALLEND  011246          3877*  3885*  3903*  3911*  3919*  3927*  3944*  3951*  3957*  3963*  3969*  3975*  3981*
                        4008#  4272
BIT0   = 000001          429#  4059   4090   4158   4307   4355   4385
BIT00  = 000001          419#   429
BIT01  = 000002          418#   428
BIT02  = 000004          417#   427
BIT03  = 000010          416#   426
BIT04  = 000020          415#   425
BIT05  = 000040          414#   424
BIT06  = 000100          413#   423
BIT07  = 000200          412#   422
BIT08  = 000400          411#   421
BIT09  = 001000          410#   420
BIT1   = 000002          428#
BIT10  = 002000          409#
BIT11  = 004000          408#  4048   4085   4346   4381
BIT12  = 010000          407#
BIT13  = 020000          406#
BIT14  = 040000          405#
BIT15  = 100000          404#  4551
BIT2   = 000004          427#
BIT3   = 000010          426#
BIT4   = 000020          425#
BIT5   = 000040          424#
BIT6   = 000100          423#
BIT7   = 000200          422#
BIT8   = 000400          421#
BIT9   = 001000          420#
BM.ERR  012143          3987   4107#
BMB73X  012066          3937   4107#
BPTVEC= 000014           436#
CONT    010632          3860   3934#
CORMAX  017452          4932#  4933
DEAD    013600          4272*  4299   4315*  4318#
DH.2    012576          4120   4195#
DH.2A   012772          4162   4195#
DH.2B   012660          4121   4163   4195#  4274   4313
DH1     017164           504   4925#
DH2     017263           511    524   4925#
DH3     017315           518   4925#
DISPLA= 177570           361#  4781*  4795*
DT1     017362           505   4926#
DT2     017376           512    525   4929#
DT3     017404           519   4930#
EMTVEC= 000030           439#  3844*  3845*
EM1     016764           503   4925#
EM2     017026           510   4925#
EM3     017064           517   4925#
EM4     017124           523   4925#
END.YA  001776           735#  3877   3951
END.YB  002776          1107#  3885   3957
END.YC  003776          1575#  3903   3963
END.YD  004776          2217#  3911   3969
END.YF  005776          3250#  3919   3975
END.YG  006776          3824#  3927   3981
```

```
END.YX  007776          3831#  4010
END.Y.  007376          3829#  3944
EOP     015502          4102   4397   4516   4562#
ERRVEC= 000004           432#  4769   4770*  4772*  4775*
EXTEND  011252          4010#  4244   4310   4315
EXTMAP  011250          4009#  4064   4161   4314   4360
FLAG4   014762          3857*  4418   4420*  4523#
GNS   = ****** U         344   4918   4919   4920   4921   4922-  4923   4924
ICOUNT  001220           529#  4028*  4100*  4327*  4395*  4417*  4493*
INITFG  011254          3849*  3934   3936*  4011#
IOTVEC= 000020           437#  3842*  3843*
LAST    012064          4033*  4053   4061   4063*  4105#  4116*  4117*  4152  4331*  4350  4357  4359*
LASTA   011256          3878*  3886*  3904*  3912*  3920*  3928*  3945*  3952*  3958*  3964*  3970*  3976*  3982*
                        4012#  4033   4095   4116   4331   4390
LOC1    014752          4422   4464   4474   4476   4498   4508   4519#
LOC2    014754          4520#
LOC3    014756          4521#
LOC4    014760          4522#
LOGIC   015550          4574#
LSTERR  001216           528#  3852*  4415*  4528*  4562*
MACTV   015000          4433   4524#
MADD1   015013          4434   4524#
MADD2   015047          4448   4524#
MAP.YA  001400           551#  3866   3876   3950
MAP.YB  002000           740#  3882   3884   3956
MAP.YC  003000          1109#  3890   3894   3902   3962
MAP.YD  004000          1580#  3908   3910   3968
MAP.YF  005000          2220#  3916   3918   3974
MAP.YG  006000          3259#  3924   3926   3980
MAP.YX  007400          3830#  4009
MAP.Y.  007000          3828#  3943   4126   4142
MASTER  015217          4214   4526#
MCHAN   014764          4425   4524#
MCRLF   015324          4130   4169   4233   4247   4279   4526#
MERRPC  015564          4580#
MFAIL   015250          4526#  4545
MLF     015327          4526#
MSG3    015104          3989   4526#
MSG4    015136          4273   4526#
MSG5    015164          4312   4526#
MSPACE  015315          4134   4147   4173   4181   4238   4284   4289   4297   4426  4430  4444  4451  4526#
M.DOL   015306          4526#
M.END   015223          4526#  4563
M.QM    015310          4005   4229   4253   4526#
NMATCH  011360          3932   4016#
NOROM   011260          3862   4013#
NOROMS  011270          4013   4016#
NO.TRA  015442          4029   4115   4326   4423   4441   4465   4470   4549#
PC    =%000007           373#  4102*  4397*  4516*  4574*  4578*  4618*  4625*  4630*  4802*  4846*
PIRQ  = 177772           359#
PIRQVE= 000240           443#
PRG.NO  015440          3854*  4027*  4114*  4211*  4414*  4547   4548#  4557   4567
PRG.1   011510          4029#  4104
PRG1    011474          3880   3888   3906   3914   3922   3930   3995   4027#  4103   4567
PRG2    012216          3998   4114#  4194
PRG3    013156          4001   4211#
```

```
FRG4    014166          4004    4414#   4461    4517    4569
PS    = 177776           356#     357   4546*   4556*
PSW   = 177776           357#
PWRVEC= 000024           438#
PWR.UP  015402          4533    4540#
RDCHR = 104410          4659    4922#
RDLIN = 104412          3938    3990    4215    4870    4923#
RDOCT = 104414          4240    4261    4924#
RESTRT  010000           348    3836#   4006
RESVEC= 000010           433#
RUN.3   013616          4328#   4399
RUN3    013602          4326#   4398
R0    =%000000           364#   4030*   4035    4037    4038    4040    4042    4043    4053    4056    4065    4071*   4074
                        4076    4077    4079    4080*   4082    4092    4095    4098    4118*   4123    4124    4128    4129
                        4139    4140    4144    4145    4152    4160*   4165    4166    4168    4177    4178    4186    4187
                        4192*   4271*   4276    4277    4278    4291    4292    4293    4301    4314*   4328*   4333    4335
                        4336    4338    4340    4341    4350    4353    4361    4366*   4370    4372    4373    4375    4376*
                        4378    4387    4390    4393    4435*   4440*   4449*   4471*   4481*   4497*   4507*   4542*   4543*
                        4553*   4610    4611*   4612    4615*   4818    4819*   4820*   4827*   4828*   4829*   4830*   4831*
                        4832    4837    4842*   4844*   4848    4850    4867    4871*   4874    4890*   4902    4903*   4904
                        4905*   4906*   4907*
R1    =%000001           365#   4123*   4125    4139*   4141    4165*   4167    4177*   4179    4212*   4213    4235*   4241*
                        4243    4244    4246    4251    4472*   4478    4485    4498*   4502    4508*   4513    4565*   4570*
                        4574    4868    4872*   4876*   4878*   4880*   4883*   4886    4889*
R2    =%000002           366#   3939*   3940*   3941    3947*   3948    3954    3960    3966    3972    3978    3984    3991*
                        3992    4216*   4217*   4218*   4219    4222    4224    4227    4869    4873*   4877*   4879*   4881*
                        4887    4888*
R3    =%000003           367#   3992*   3993    3996    3999    4002    4034*   4050*   4073*   4087*   4119*   4128*   4144*
                        4161*   4168*   4186*   4299*   4300    4301    4309    4310    4332*   4348*   4369*   4383*   4463*
                        4467    4490*   4491    4655    4656*   4657    4660*   4661    4665    4667    4669*   4671*   4711
                        4720*   4726*   4727*   4730*   4735*   4736*   4737    4746*
R4    =%000004           368#   4032*   4037    4040    4041    4044    4056    4064*   4330*   4335    4338    4339    4342
                        4353    4360*   4422*   4447*   4464*   4472    4488    4712    4714*   4715*   4716*   4717    4718*
                        4732    4734*   4742*   4745*
R5    =%000005           369#   4421*   4437    4458*   4459    4473*   4478    4482*   4485    4499*   4502    4509*   4513
                        4713    4719*   4721*   4723*   4724*   4725*   4726    4744*
R6    =%000006           370#    372    3837*   3838*   3839
R7    =%000007           371#    373
SAVR0   001230           533#
SAVR1   001232           534#
SAVR4   001234           535#
SAVR5   001236           536#
SP    =%000006           372#   3841*   3851*   3861*   3865    3868*   3872    3874    3875    3892*   3896    3898    3899
                        3939    3991    4084*   4132*   4137*   4149*   4170*   4174*   4183*   4216    4234*   4241    4248*
                        4262    4281*   4286*   4295*   4380*   4416*   4428*   4439*   4445*   4453*   4469*   4529*   4534
                        4536    4537    4541*   4550    4551    4553    4555*   4610*   4611    4612*   4614    4615    4616*
                        4619    4621*   4623*   4629    4641*   4642*   4645*   4646*   4655*   4660    4671    4672*   4673*
                        4674*   4703*   4704    4705    4706*   4711*   4712*   4713*   4719    4744    4745    4746    4747*
                        4748*   4769*   4772    4774    4775    4779    4782*   4797    4818*   4823*   4844    4848*   4865*
                        4866*   4867*   4868*   4869*   4871    4874*   4882*   4883    4885    4886*   4888    4889    4890
                        4902*   4903
SPACE3  015320          4526#
SRV.A   013356          4228    4261#
SRV.D   013254          4223    4232#
SRV.L   013370          4220    4270#
SRV.R   013602          4226    4325#
```

```
STACK = 001100          353#  3841  3851  4084  4380  4416  4439  4469  4541  4555
START   010102         3851#  3854
STKLMT= 177774          358#
SWR   = 177570          360#   361  4048  4059  4085  4090  4158  4307  4346  4355  4381  4385  4764*
                       4800   4804
SW0   = 000001          401#
SW00  = 000001          391#   401
SW01  = 000002          390#   400
SW02  = 000004          389#   399
SW03  = 000010          388#   398
SW04  = 000020          387#   397
SW05  = 000040          386#   396
SW06  = 000100          385#   395
SW07  = 000200          384#   394
SW08  = 000400          383#   393  4760  4778
SW09  = 001000          382#   392   480  3848  4760  4776  4778  4780  4784  4790  4807  4808
SW1   = 000002          400#
SW10  = 002000          381#   480  4790  4796  4808
SW11  = 004000          380#   480  3848  4760  4778  4780  4784
SW12  = 010000          379#
SW13  = 020000          378#  4789  4800
SW14  = 040000          377#  4759  4764
SW15  = 100000          376#  4788  4804
SW2   = 000004          399#
SW3   = 000010          398#
SW4   = 000020          397#
SW5   = 000040          396#
SW6   = 000100          395#
SW7   = 000200          394#
SW8   = 000400          393#
SW9   = 001000          392#
TABLE   011244         3850*  3876*  3884*  3902*  3910*  3918*  3926*  3943*  3950*  3956*  3962*  3968*  3974*
                       3980*  3985  4007#  4032  4119  4126  4142  4212  4271  4330
TAG.A   014434         4419  4463#  4494
TBITVE= 000014          434#
TEMP3   001224          531#  4044*  4045  4124*  4132  4140*  4166*  4170  4178*  4277*  4281  4292*  4342*
                       4343  4424*  4428  4456*  4476*  4926
TEMP4   001226          532#  4043*  4045  4082*  4125*  4137  4141*  4149  4167*  4174  4179*  4183  4232*
                       4234  4235  4246*  4248  4276*  4286  4291*  4295  4341*  4343  4378*  4442*  4445
                       4450*  4453  4477*  4484*  4501*  4511*  4926  4929
TEMP5   001222          530#  4122*  4154*  4164*  4189*  4275*  4303*
TKVEC = 000060          441#
TPVEC = 000064          442#
TRAPVE= 000034          440#  3846*  3847*
TRTVEC= 000014          435#
TYPE  = 104400         3932  3937  3987  3989  4005  4013  4120  4121  4130  4134  4147  4162  4163
                       4169  4173  4181  4214  4229  4233  4238  4247  4253  4273  4274  4279  4284
                       4289  4297  4312  4313  4425  4426  4430  4433  4434  4444  4448  4451  4545
                       4563  4564  4663  4666  4670  4738  4803  4817  4834  4836  4839  4841  4845
                       4852  4918#
TYPOC = 104402         4133  4138  4150  4171  4175  4184  4236  4249  4282  4287  4296  4429  4446
                       4454  4825  4849  4919#
TYPON = 104406         4921#
TYPOS = 104404         4920#
VERS    012171         4107#
VERSON  015562         3879*  3887*  3905*  3913*  3921*  3929*  3984*  4564  4579#
```

```
XHOLD   013174          4214#  4230   4252   4254   4263   4317
XSTORE  015500          4550*  4560#  4930
X.X.    011156          3990#
X1      015560          4566   4571   4578#
$BDADR  001122           468#
$BDDAT  001126           470#  4042*  4340*  4475*  4483*  4500*  4510*  4926
$CMTAG  001100           456#  3837   3844   3848
$CM3  =  000000           480#
$CRLF   001153           481#  4679   4803   4808   4817   4836   4841   4845
$ENDAD  015550           452   4573#
$ERFLG  001103           459#  4761   4780*  4784   4793*  4808
$ERMAX  001115           465#  4784
$ERROR  016376          3844   4792#
$ERRPC  001116           466#  4797*  4798*  4799   4808   4823   4926   4929   4930
$ERRTB  001156           499#  4831
$ERRTY  016470          4802   4816#
$ERTTL  001112           463#  4796*  4808
$FILLC  001150           478#  4619   4631
$FILLS  001147           477#  4631
$GDADR  001120           467#
$GDDAT  001124           469#  4041*  4339*  4474*  4926
$HD   =  000003           547    548
$HIOCT  016724          4887*  4892#
$ICNT   001104           460#
$ITEMB  001114           464#  4799*  4808   4820
$LF     001154           482#  4670   4679   4808
$LPADR  001106           461#  3848*  4031*  4070*  4329*  4367*  4779*  4782   4784
$LPERR  001110           462#
$NULL   001146           476#  4621   4631
$OCNT   016276          4710*  4739*  4752#
$OMODE  016300          4705*  4709*  4714   4717*  4728*  4754#
$OVER   016362          4765   4776   4781#
$PASS   001100           457#
$QUES   001152           480#  4663   4679   4808
$RDCHR  015702          4641#  4922
$RDDEC=  ******  U      4925
$RDLIN  015736          4655#  4923
$RDOCT  016624          4865#  4924
$RDSZ =  000010          4648#
$SAVRE=  ******  U      4925
$SCOPE  016302          3842   4763#
$SETUP=  000007          3837#  3842   3844   3846   3848
$STUP =  177777          3837#
$SVLAD  016346          4773   4778#
$SWR  =  160000           480    547    548#  3848   4759   4760   4764   4776   4778   4780   4784   4788   4789
                        4790   4796   4800   4804   4807   4808
$TKB    001140           473#  4640   4645
$TKS    001136           472#  3839   4640   4643
$TN   =  000001           547#
$TPB    001144           475#  4629*  4631
$TPFLG  001151           479#  4606   4631
$TPS    001142           474#  4627   4631
$TRAP   016726          3846   4902#
$TRP  =  000016          4909#  4919#  4920#  4921#  4922#  4923#  4924#  4925#
$TRPAD  016746          4906   4917#
$TSTNM  001102           458#  4761   4778*  4781   4784   4795   4808
```

```
$TTYIN  016044          4656    4657    4674    4678#
$TYPBN= ****** U        4922
$TYPDS= ****** U        4922
$TYPE   015574          4606#   4909    4918
$TYPOC  016100          4708#   4919
$TYPON  016114          4707    4710#   4921
$TYPOS  016054          4703#   4920
$XTSTR  016310          4767#
$OFILL  016277          4704*   4708*   4718    4753#
.     = 017452           340#    344     346#    451#    454#    483     550#    3828#   3829#   3830#   3831#   3840    3848
                        3859    4015    4016#   4107#   4195#   4524#   4526#   4568    4582#   4631    4640    4678#   4679
                        4784    4808    4855#   4925#   4931#
.PFAIL  015332          4528#   4540
```

F10


```
COMMEN      1#    444#
ENDCOM      1#    444#
ERROR     354#   4047   4083   4345   4379   4480   4487   4504   4515   4554   4558
ESCAPE      1#    444#
MULT        1#    444#
NEWTST      1#    444#
POP         1#    444#   4888
PUSH        1#    444#   4867
SCOPE     355#   4055   4097   4352   4392
SETTRA   4909#   4919   4920   4921   4922   4923   4924
SETUP       1#    444#   3837
SKIP        1#    444#
SLASH       1#    444#
SPACE     444#
STARS       1#    444#    483   4584   4631   4648   4679   4755   4784   4808   4856   4893
TRMTRP   4909#
TYPBIN      1#    444#
TYPDEC      1#    444#
TYPNUM      1#    444#
TYPOCS      1#    444#
TYPOCT      1#    444#   4823   4847
TYPTXT      1#    444#
$HOW      549#   3941   3948   3954   3960   3966   3972   3978
$ROMTY    549#   3866   3882   3890   3908   3916   3924
$$CMRE    444#
$$CMTM    444#
$$ESCA      1#    444#
$$NEWT      1#    444#
$$SET    4909#   4919   4920   4921   4922   4923   4924
$$SKIP      1#    444#
.EQUAT      1#    335#    349
.HEADE      1#    335#    537
.KT11       1#
.RDLIN    336#
.SETUP      1#    335#   3836
.SWRHI      1#    335#
.SWRLO    335#
.$CATC      1#    335#    337
.$CMTA      1#    335#    444
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#   4583#
.$ERRO      1#    336#   4784
.$ERRT      1#    336#   4808
.$MULT      1#
.$POWE      1#   4583#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#    336#   4856
.$READ      1#   4583#   4631
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#    336#   4755
.$SIZE      1#
```

```
.$SUPR      1#
.$TRAP      1#    335#   4893
.$TYPB      1#
.$TYPD      1#
.$TYPE      1#   4583#   4584
.$TYPO      1#   4583#   4679
```

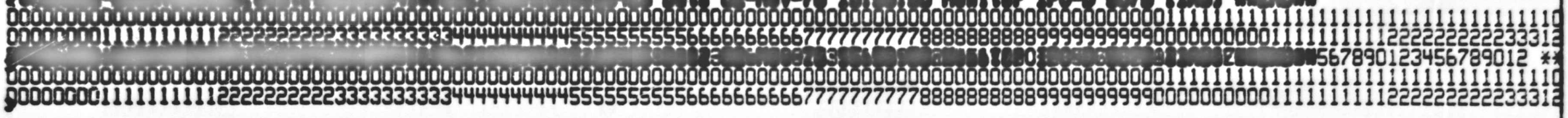
```
ADD     4117  4543  4616  4706  4716  4831  4883
ASL     4828  4829  4830  4876  4878  4880
BEQ     4046  4054  4060  4062  4091  4093  4096  4153  4159  4188  4220  4223  4228  4302  4308
        4311  4344  4351  4356  4358  4386  4388  4391  4479  4486  4503  4514  4552  4566  4571
        4733  4794  4833  4838  4851  4875
BGT     4740
BIC     3947  4218  4646  4730  4882
BIS     4735  4736
BISB    4820
BIT     4048  4059  4085  4090  4158  4307  4346  4355  4381  4385  4551  4800
BLO     4245
BLOS    4658
BLT     4624  4741
BMI     4765
BNE     3840  3859  3867  3883  3891  3895  3909  3917  3925  3935  3942  3949  3955  3961  3967
        3973  3979  3986  3994  3997  4000  4003  4036  4039  4049  4051  4075  4078  4086  4088
        4101  4127  4143  4155  4190  4225  4304  4334  4337  4347  4349  4371  4374  4382  4384
        4396  4419  4460  4492  4494  4544  4568  4613  4620  4662  4668  4731  4801  4821  4843
BPL     4607  4628  4644  4729  4805
BR      3873  3897  3900  3988  4015  4057  4066  4094  4099  4156  4191  4230  4305  4316  4354
        4362  4389  4394  4609  4626  4664  4707  4722  4743  4767  4773  4776  4826  4853  4884
CLC     4457  4489
CLR     3838  3849  3850  3852  3853  3856  3857  4192  4415  4436  4466  4496  4506  4528  4542
        4546  4556  4562  4720  4819  4872  4873
CLRB    4669  4780
CMP     3839  3866  3874  3882  3890  3894  3898  3908  3916  3924  3941  3948  3954  3960  3966
        3972  3978  3993  3996  3999  4002  4035  4037  4038  4040  4045  4053  4056  4061  4074
        4077  4092  4095  4126  4142  4152  4187  4219  4222  4224  4227  4244  4301  4310  4333
        4335  4336  4338  4343  4350  4353  4357  4370  4373  4387  4390  4459  4478  4485  4491
        4502  4513  4567  4657  4774
CMPB    4619  4661  4667
COM     3936  4420
DEC     4050  4087  4100  4154  4189  4303  4348  4383  4395  4493  4827
DECB    4623  4728  4739
EMT      354
HALT     344  3933  4014  4193  4535  4538  4608  4806
INC     4456  4734  4742  4796
INCB    4778  4793
IOT      355
JMP      348  3860  3880  3888  3906  3914  3922  3930  3995  3998  4001  4004  4006  4103  4104
        4194  4226  4252  4254  4263  4317  4398  4399  4461  4517  4547  4557  4569
JSR     4102  4397  4516  4574  4618  4625  4802
MOV     3837  3841  3842  3843  3844  3845  3846  3847  3848  3851  3854  3855  3861  3862  3865
        3868  3869  3872  3875  3876  3877  3878  3879  3884  3885  3886  3887  3892  3893  3896
        3899  3902  3903  3904  3905  3910  3911  3912  3913  3918  3919  3920  3921  3926  3927
        3928  3929  3939  3940  3943  3944  3945  3950  3951  3952  3956  3957  3958  3962  3963
        3964  3968  3969  3970  3974  3975  3976  3980  3981  3982  3984  3991  3992  4027  4028
        4029  4030  4031  4032  4033  4034  4041  4042  4043  4044  4063  4064  4070  4071  4072
        4073  4080  4082  4084  4114  4115  4116  4118  4119  4122  4123  4124  4125  4128  4132
        4137  4139  4140  4141  4144  4149  4160  4161  4164  4165  4166  4167  4168  4170  4174
        4177  4178  4179  4183  4186  4211  4212  4213  4216  4217  4232  4234  4235  4241  4246
        4248  4251  4262  4271  4272  4275  4276  4277  4281  4286  4291  4292  4295  4299  4314
        4315  4326  4327  4328  4329  4330  4331  4332  4339  4340  4341  4342  4359  4360  4366
        4367  4368  4369  4376  4378  4380  4414  4416  4417  4421  4422  4423  4424  4428  4435
        4437  4439  4440  4441  4442  4445  4447  4449  4450  4453  4463  4464  4465  4467  4469
        4470  4471  4472  4473  4474  4475  4476  4477  4481  4482  4483  4484  4495  4497  4498
```

```
                4499  4500  4501  4505  4507  4508  4509  4510  4511  4529  4530  4533  4534  4537  4540
                4541  4550  4553  4555  4565  4570  4610  4611  4615  4621  4641  4642  4655  4656  4671
                4672  4673  4674  4703  4711  4712  4713  4719  4726  4744  4745  4746  4747  4748  4769
                4770  4772  4775  4779  4781  4782  4795  4797  4818  4823  4832  4837  4842  4844  4848
                4865  4866  4867  4868  4869  4871  4886  4887  4888  4889  4890  4902  4903  4906
MOVB            4612  4629  4645  4660  4665  4704  4705  4708  4709  4710  4714  4717  4718  4737  4799
                4874  4905
NEG             4715
NOP             3864  3871  4081  4377  4438  4468  4532  4575  4576  4577
RESET           4572
ROL             4458  4490  4721  4723  4724  4725  4727  4764  4877  4879  4881
RTI             4559  4617  4647  4675  4749  4783  4807  4891
RTS             4578  4630  4846  4907
SUB             4798
TRAP            4909  4919  4920  4921  4922  4923  4924
TST             3858  3863  3870  3934  3985  4065  4076  4079  4098  4129  4145  4243  4278  4293  4300
                4309  4361  4372  4375  4393  4418  4488  4531  4536  4614  4732  4771  4804  4850  4885
                4904
TSTB            4606  4627  4643
.ASCII           480   481  4016  4107  4195  4526  4925
.ASCIZ           482  4016  4107  4195  4524  4526  4580  4854  4925
.BLKB           4678
.BLKW           3828  3829  3830  3831
.BYTE            458   459   464   465   476   477   478   479  1012  1013  1014  1015  1016  1017  4676
                4677  4750  4751  4752  4753
.ENABL              1   335
.END            4933
.ENDC            349   354   430   444   445   455   480   483   484   542  3837  3841  3842  3844  3846
                3848  3876  3884  3892  3902  3910  3918  3926  4585  4632  4648  4649  4656  4658  4663
                4678  4679  4680  4756  4762  4764  4766  4777  4778  4780  4784  4785  4791  4793  4797
                4803  4804  4808  4809  4827  4856  4857  4893  4894  4903  4906  4918  4919  4920  4921
                4922  4923  4924  4925  4933
.EQUIV           354   355   357   372   373   392   393   394   395   396   397   398   399   400   401
                 420   421   422   423   424   425   426   427   428   429
.EVEN           4016  4107  4195  4524  4526  4582  4855  4925
.IF              344   352   402   430   444   454   480   483   538  3837  3841  3842  3844  3846  3848
                3868  3876  3884  3892  3910  3918  3926  4584  4631  4648  4656  4657  4662  4678  4679
                4755  4761  4764  4776  4778  4780  4784  4790  4793  4796  4800  4802  4803  4804  4807
                4808  4826  4842  4856  4872  4893  4902  4906  4909  4919  4920  4921  4922  4923  4924
                4925  4933
.IFF             352   445   454   480   484  3841  4585  4632  4648  4649  4658  4662  4679  4680  4756
                4776  4778  4784  4785  4790  4797  4807  4808  4809  4827  4856  4857  4894  4903
.IFT            4778  4803  4876  4892  4893
.IFTF           4778  4802  4872  4876  4892
.IIF             344   483   537   542   547   548  3842  3844  3848  4631  4640  4671  4679  4759  4760
                4780  4784  4788  4789  4790  4808  4824  4849  4918  4919  4920  4921  4922  4923  4924
.IRP            3837  4867  4888
.LIST              1   335   344   444   480   550  3828  3837  4016  4107  4195  4524  4526  4648  4909
                4918  4919  4920  4921  4922  4923  4924  4925  4933
.MACRO             1   444   549  4909
.MCALL           335   336   444  4583
.NLIST             1   335   344   444   480   550  3828  3837  4016  4107  4195  4524  4526  4648  4909
                4918  4919  4920  4921  4922  4923  4924  4925  4933
.PAGE            444   483   737  1109  1580  1631  1678  1741  1786  1837  1907  1961  2014  2068  2115
                2167  2220  3259  3828  3832  4067  4363  4524
.REM               1  2021  2071
```

```
.REPT     344
.SBTTL    335     338     345     350     446     485     550    4586    4633    4681    4757    4786    4810    4858    4895
         4910
.TITLE    537
.WORD     344     457     460     461     462     463     466     467     468     469     470     471    4835    4840    4892
         4926    4929    4930
```

```
ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

*,DZBMDG.SEQ/SOL/CRF/PAGNUM/NL:TOC/DS:ERFZZ=SYSMAC.A,DZBMDG.P11
RUN-TIME:  29 45 3 SECONDS
RUN-TIME RATIO: 144/77=1.8
CORE USED:  21K  (41 PAGES)
```



K10

```
Spooler runtime 15 Seconds, 66 KCS, 449 disk reads, 3 disk writes, 126 pages
/[redacted] Date [illegible]-Nov-76 15:[illegible]:50 Monitor IPC-D [illegible] [illegible] [illegible]
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000111111111111111111111111111111110
000000000111111111122222222223333333333444444444455555555556666666666777777777788888888889999999999000000000011111111112222222222333312
[redacted]567890123456789012 **
000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000111111111111111111111111111111110
000000000111111111122222222223333333333444444444455555555556666666666777777777788888888889999999999000000000011111111112222222222333312
```