6.8.75

IDENTIFICATION

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

PRODUCT NAME: BM873 UNIVERSAL RESTART LOADER

DATE CREATED: APRIL 21, 1975

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: JOHN EGOLF

YA-version

REVISED BY: BOB MISNER 10/21/74
FAY BASHAW 3/21/75

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 FAMILY CPU
UNIVERSAL RESTART LOADER
TELETYPE OR EQUIVALENT
LINE PRINTER (OPTIONAL)
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 PAPER TAPE BY THE
USE OF ABSOLUTE LOADER.

4. STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

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

4.2 STARTING ADDRESS

STARTING ADDRESS 000200
RESTART ADDRESS 000210

4.3 OPERATOR ACTION

4.3.1 FOR NORMAL OPERATION (WITHOUT EXTENDED 128 WORDS)
1. LOAD STARTING ADDRESS (000200)
2. SET SWITCHES AS PER 5.1.1 (NORMAL ALL SWITCHES DOWN)
3. PRESS START SWITCH AND RELEASE.
4. DEVICE VERSION.
WHEN PROGRAM IS STARTED FOR THE FIRST TIME THE FOLLOWING
WILL BE PRINTED OUT:
"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

5. THEN TYPE IN NUMBER OF PROGRAM TO BE RUN (NORMALLY PROGRAM 1 AND 4)
6. HITTING CONTROL "G" WHILE PROGRAM IS RUNNING WILL CAUSE A RESTART.
YOU MAY THEN INPUT DIFFERENT PROGRAM NUMBER.

NOTE: RESTARTS AT ADDRESS 200 WILL GO TO STEP 5 ABOVE.
RESTARTS AT ADDRESS 210 WILL GO TO STEP 3 ABOVE.

4.3.2 IF YOU WISH TO TEST THE EXTENDED 128. WORDS THIS IS THE
PROCEDURE:
(NOT NEEDED FOR NORMAL TESTING OF BM873YB, BM873YC, OR BM873YD)
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

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

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

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

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

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

6. ERRORS

6.1 ERROR PRINT OUT

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

6.2 ERROR RECOVERY

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

2. IN AN ERROR; SET SW09=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 OPERATING ON 11/45; AND JUMPER HAS BEEN CUT FOR
A POWER FAIL TO GOTO ADDRESS 173200;
****** A POWER FAIL MUST NOT HAPPEN WHILE PRG IS RUNNING *****

7.2.2 IF YOU WISH PROGRAM TO TEST YOUR EXTENDED 128. WORDS;
YOU MUST START AS PER SECTION 4 AND THEN
****** RUN PROGRAM 2 FIRST AND VISUALLY VERIFY DATA.****
(NOT APPLICABLE TO BM873YB, BM873YC, OR BM873YD)

7.2.3 YOU MAY NOT ALTER THE SOFTWARE MAP UNLESS--
****** YOU KNOW WHAT YOU ARE DOING *****

7.2.4 THE ROM ADDRESS MUST START AT 173000 AND BE AT LEAST
128 WORDS LONG. (256 FOR THE BM873YB, BM873YC, OR BM873YD)

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.
PROGRAN 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
BM873YB, BM873YC, OR BM873YD IS TESTED.

9.2 PROGRAM 2

PROGRAM 2 WILL DUMP THE CONTENTS OF THE ROM ONTO THE
TTY (OR LINE PRINTER IF SW07=1), 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 BM873YB, BM873YC, OR BM873YD
IS SELECTED
NOTE: IF SW 07=1 PRINTING WILL BE DONE ON THE LINE PRINTER
IF IT EXISTS.

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
REPETED 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 CONTROL "G"

HITTING CONTROL "G" WILL GO TO THE BEGINING OF THE PROGRAM AND ASK FOR PROGRAM NUMBER.

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

9.7 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 ACLO AND DCLO MUST BE AVAILABLE AT BACKPLANE PINS "CV1" AND "CN1" RESPECTIVELY. THIS WIRING IS NOT PROVIDED ON THE "SPC" SLOTS OF THE 11/15, 11/20, 11/35, 11/40, AND THE DD11-A; IT IS PROVIDED ON THE 11/05, AND 11/45, ALSO ON THE DD11-B. 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.

```
                                ;BM873 YX
                                ;COPYRIGHT MAR 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

                                        ;STARTING PROCEDURE
                                        ;LOAD PROGRAM
                                        ;LOAD ADDRESS 000200
                                        ;PRESS START
                                        ;PROGRAM WILL TYPE "DEVICE VERSION
                                        ;                   BM873-Y "
                                        ;THE OPERATOR WILL THEN SPECIFY THE VERSION.
                                        ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
                                        ;AND THEN RESUME TESTING


                                ;SWITCH REGISTER OPTIONS

100000                          SW15=100000             ;=1,HALT ON ERROR
040000                          SW14=40000
020000                          SW13=20000              ;=1, INHIBIT ERROR PRINTOUT
010000                          SW12=10000              ;=1,DELETE TYPEOUT/BELL ON ERROR,
004000                          SW11=4000               ;=1 DO EACH ADDRESS 1 TIME INSTEAD OF 10 TIMES.
002000                          SW10=2000
001000                          SW09=1000               ;=1,LOOP WITH CURRENT DATA
000400                          SW08=400                ;=1 RETURN TO TOP OF CURRENT PROGRAM ON ERROR.
000100                          SW06=100
000040                          SW05=40
000020                          SW04=20
000010                          SW03=10
000004                          SW02=4
000002                          SW01=2
000001                          SW00=1                  ;=1 TEST THE EXPANDED 128 WORDS
```



```
                                ;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                          SP=%6           ;PROCESSOR STACK POINTER
000007                          PC=%7           ;PROGRAM COUNTER

                                ;LOCATION EQUIVALENCIES

177570                          SWR=177570      ;CONSOLE SWITCH REGISTER
177570                          LIGHTS=177570   ;PDP-11/45 DISPLAY REGISTER
177776                          PS=177776       ;PROCESSOR STATUS WORD
001200                          STACK=1200      ;START OF PROCESSOR STACK

                                ;INSTRUCTION DEFINITIONS

005746                          PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
005726                          POP1SP=5726     ;INCREMENT PROCESSOR STACK 1 WORD
010046                          PUSHR0=10046    ;SAVE R0 ON STACK
012600                          POPR0=12600     ;RESTORE R0 FROM STACK
024646                          PUSH2SP=24646   ;DECREMENT STACK TWICE
022626                          POP,SP=22626    ;INCREMENT STACK TWICE
                                .EQUIV  EMT,HLT ;BASIC DEFINITION OF ERROR CALL


100000                          BIT15=100000
040000                          BIT14=40000
020000                          BIT13=20000
010000                          BIT12=10000
004000                          BIT11=4000
002000                          BIT10=2000
001000                          BIT9=1000
000400                          BIT8=400
000200                          BIT7=200
000100                          BIT6=100
000040                          BIT5=40
000020                          BIT4=20
000010                          BIT3=10
000004                          BIT2=4
000002                          BIT1=2
000001                          BIT0=1
```

```
                                        ;TRAPCATCAER FOR ILLEGAL INTERRUPTS
                    000000          .=0
000000  000002                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000002  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000004  000006                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000006  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000010  000012                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000012  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000014  000016                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000016  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000020  000022                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000022  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000024  000026                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000026  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000030  000032                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000032  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000034  000036                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000036  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000040  000042                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000042  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000044  000046                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000046  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000050  000052                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000052  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000054  000056                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000056  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000060  000062                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000062  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000064  000066                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000066  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000070  000072                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000072  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000074  000076                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000076  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000100  000102                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000102  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000104  000106                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000106  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000110  000112                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000112  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000114  000116                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000116  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000120  000122                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000122  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000124  000126                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000126  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000130  000132                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000132  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000134  000136                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000136  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000140  000142                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000142  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000144  000146                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000146  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
```



```
000150  000152                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000152  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000154  000156                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000156  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000160  000162                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000162  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000164  000166                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000166  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000170  000172                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000172  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000174  000176                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000176  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000200  000202                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000202  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000204  000206                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000206  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000210  000212                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000212  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000214  000216                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000216  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000220  000222                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000222  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000224  000226                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000226  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000230  000232                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000232  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000234  000236                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000236  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000240  000242                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000242  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000244  000246                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000246  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000250  000252                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000252  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000254  000256                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000256  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000260  000262                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000262  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000264  000266                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000266  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000270  000272                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000272  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000274  000276                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000276  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000300  000302                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000302  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000304  000306                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000306  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000310  000312                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000312  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000314  000316                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000316  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
000320  000322                      .+2             ;UNEXPECTED TRAP TO THIS LOCATION
000322  000000                      HALT            ;EXAMINE STACK TO FIND CAUSE
```

```
509  000324  000326                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
510  000326  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
511  000330  000332                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
512  000332  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
513  000334  000336                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
514  000336  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
515  000340  000342                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
516  000342  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
517  000344  000346                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
518  000346  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
519  000350  000352                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
520  000352  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
521  000354  000356                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
522  000356  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
523  000360  000362                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
524  000362  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
525  000364  000366                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
526  000366  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
527  000370  000372                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
528  000372  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
529  000374  000376                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
530  000376  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
531  000400  000402                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
532  000402  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
533  000404  000406                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
534  000406  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
535  000410  000412                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
536  000412  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
537  000414  000416                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
538  000416  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
539  000420  000422                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
540  000422  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
541  000424  000426                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
542  000426  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
543  000430  000432                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
544  000432  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
545  000434  000436                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
546  000436  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
547  000440  000442                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
548  000442  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
549  000444  000446                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
550  000446  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
551  000450  000452                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
552  000452  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
553  000454  000456                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
554  000456  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
555  000460  000462                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
556  000462  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
557  000464  000466                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
558  000466  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
559  000470  000472                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
560  000472  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
561  000474  000476                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
562  000476  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
```



```
563  000500  000502                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
564  000502  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
565  000504  000506                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
566  000506  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
567  000510  000512                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
568  000512  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
569  000514  000516                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
570  000516  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
571  000520  000522                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
572  000522  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
573  000524  000526                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
574  000526  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
575  000530  000532                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
576  000532  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
577  000534  000536                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
578  000536  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
579  000540  000542                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
580  000542  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
581  000544  000546                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
582  000546  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
583  000550  000552                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
584  000552  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
585  000554  000556                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
586  000556  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
587  000560  000562                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
588  000562  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
589  000564  000566                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
590  000566  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
591  000570  000572                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
592  000572  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
593  000574  000576                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
594  000576  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
595  000600  000602                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
596  000602  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
597  000604  000606                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
598  000606  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
599  000610  000612                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
600  000612  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
601  000614  000616                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
602  000616  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
603  000620  000622                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
604  000622  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
605  000624  000626                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
606  000626  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
607  000630  000632                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
608  000632  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
609  000634  000636                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
610  000636  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
611  000640  000642                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
612  000642  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
613  000644  000646                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
614  000646  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
615  000650  000652                  .+2             ;UNEXPECTED TRAP TO THIS LOCATION
616  000652  000000                  HALT            ;EXAMINE STACK TO FIND CAUSE
```

```
   617  000654  000656                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   618  000656  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   619  000660  000662                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   620  000662  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   621  000664  000666                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   622  000666  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   623  000670  000672                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   624  000672  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   625  000674  000676                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   626  000676  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   627  000700  000702                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   628  000702  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   629  000704  000706                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   630  000706  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   631  000710  000712                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   632  000712  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   633  000714  000716                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   634  000716  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   635  000720  000722                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   636  000722  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   637  000724  000726                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   638  000726  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   639  000730  000732                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   640  000732  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   641  000734  000736                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   642  000736  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   643  000740  000742                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   644  000742  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   645  000744  000746                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   646  000746  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   647  000750  000752                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   648  000752  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   649  000754  000756                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   650  000756  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   651  000760  000762                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   652  000762  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   653  000764  000766                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   654  000766  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   655  000770  000772                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   656  000772  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   657  000774  000776                          .+2             ;UNEXPECTED TRAP TO THIS LOCATION
   658  000776  000000                          HALT            ;EXAMINE STACK TO FIND CAUSE
   659
   660          000024                  .=24
   661  000024  014510                          .PFAIL
   662  000026  000340                          340
   663  000030  015116                          .HLT
   664  000032  000000                          0
   665  000034  015022                          .TRPSR
   666  000036  000000                          0
   667          000200                  .=200
   668  000200  000137  007150                  JMP     @START
   669          000210                  .=210
   670  000210  000137  007140                  JMP     @RESTRT
```



```
   671          001200                  .=1200
   672
   673                                  ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
   674
   675  001200  177514                  LPTCSR: 177514
   676  001202  177516                  LPTDBR: 177516
   677  001204  177560                  TKCSR:  177560          ;TELETYPE KEYBOARD CONTROL REGISTER
   678  001206  177562                  TKDBR:  177562          ;TELETYPE KEYBOARD DATA BUFFER
   679  001210  177564                  TPCSR:  177564          ;TELEPRINTER CONTROL REGISTER
   680  001212  177566                  TPDBR:  177566          ;TELEPRINTER DATA BUFFER
   681  001214  000060                  TKVEC:  60              ;TELETYPE KEYBOARD VECTOR
   682  001216  000200                  TKLVL:  200             ;PRIORTIY =4
   683  001220  000064                  TPVEC:  64              ;TELETYPE PRINTER VECTOR
   684  001222  000200                  TPLVL:  200             ;PRIORITY =4
   685
   686                                  ;PROGRAM CONTROL PARAMETERS
   687
   688  001224  000000                  RETURN: 0               ;SCOPE ADDRESS FOR LOOP ON TEST
   689  001226  000000                  NEXT:   0               ;ADDRESS OF NEXT TEST TO BE EXECUTED
   690  001230  000000                  LOCK:   0               ;ADDRESS FOR LOCK ON CURRENT DATA
   691  001232  000000                  ICOUNT: 0               ;NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
   692  001234  000000                  LPCNT:  0               ;NUMBER OF ITEREATIONS COMPLETED
   693  001236  000000                  TSTNO:  0               ;NUMBER OF TEST IN PROGRESS
   694  001240  000000                  PASCNT: 0               ;NUMBER OF PASSES COMPLETED
   695  001242  000000                  ERRCNT: 0               ;TOTAL NUMBER OF ERRORS
   696  001244  000000                  LSTERR: 0               ;PC OF LAST ERROR CALL
   697
   698                                  ;PROGRAM VARIABLES
   699
   700  001246  000000                  TEMP1:  0               ;TEMPORARY STORAGE
   701  001250  000000                  TEMP2:  0               ;TEMPORARY STORAGE
   702  001252  000000                  TEMP3:  0               ;TEMPORARY STORAGE
   703  001254  000000                  TEMP4:  0               ;TEMPORARY STORAGE
   704  001256  000000                  TEMP5:  0               ;TEMPORARY STORAGE
   705  001260  000000                  SAVR0:  0               ;R0 STORAGE
   706  001262  000000                  SAVR1:  0               ;R1 STORAGE
   707  001264  000000                  SAVR2:  0               ;R2 STORAGE
   708  001266  000000                  SAVR3:  0               ;R3 STORAGE
   709  001270  000000                  SAVR4:  0               ;R4 STORAGE
   710  001272  000000                  SAVR5:  0               ;R5 STORAGE
   711  001274  000000                  SAVSP:  0               ;STACK POINTER STORAGE
   712  001276  000000                  SAVPC:  0               ;PROGRAM COUNTER STORAGE
```

```
                                        ;PROGRAM CONTROL FLAGS
001300      000                 INIFLG: .BYTE   0               ;PROGRAM INITIALIZATION FLAG
001301      000                 STFLG:  .BYTE   0               ;TEST START FLAG
001302      000                 ERRFLG: .BYTE   0               ;ERROR OCCURED FLAG
001303      000                 LOKFLG: .BYTE   0               ;LOCK ON CURRENT TEST FLAG
            000000              $Y=0

                                        ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
                                        ;POINTERS TO SUBROUTINES CAN BE FOUND
                                        ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS

                                ;*********************************************************************
                                ;*********************************************************************
001304                          .TRPTAB:
            104400              SCOP1=TRAP+0            ;CALL TO LOOP ON CURRENT DATA HANDLER
001304  014740                          ,SCOP1
            104401              TYPE=TRAP+1             ;CALL TO TELETYPE OUTPUT ROUTINE
001306  014756                          ,TYPE
            104402              SAV05=TRAP+2            ;CALL TO REGISTER SAVE ROUTINE
001310  015730                          ,SAV05
            104403              RES05=TRAP+3            ;CALL TO REGISTER RESTORE ROUTINE
001312  015770                          ,RES05
            104404              CONVRT=TRAP+4           ;CALL TO DATA OUTPUT ROUTINE
001314  015424                          ,CONVRT
            104405              CNVRT=TRAP+5            ;CALL TO DATA OUTPUT ROUNTINE WITHOUT CR/LF,
001316  015430                          ,CNVRT
            104406              KEY,TO,R2=TRAP+6                ;CALL TO GET CHAR, ECHO IT, PUT IT IN R2,CLR BIT 7
001320  014444                          ,KEY,TO,R2

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



```
            .=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
```

```
  908  001716  000005  ;173316 000005  CKDEV:  RESET           ;KILL ALL DEVICE ACTION
  909  001720  012701  ;173320 012701         MOV #160000,R1  ;THEN SET UP MEMORY TEST LIMITS
  910  001722  160000  ;173322 160000
  911  001724  012702  ;173324 012702         MOV #6,R2       ;AND SET UP POINTER TO TIMEOUT LOCATION
  912  001726  000006  ;173326 000006
  913  001730  012712  ;173330 012712         MOV #340,@R2    ;AND SET UP VECTOR TO RETURN TO NEXT
  914  001732  000340  ;173332 000340
  915  001734  010742  ;173334 010742         MOV PC,-(R2)    ;SAVE THE PC
  916  001736  012706  ;173336 012706         MOV #24,SP      ;AND LOAD UP STACK POINTER
  917  001740  000024  ;173340 000024
  918  001742  010441  ;173342 010441         MOV R4,-(R1)    ;AND LOOK FOR END OF MEMORY
  919  001744  040601  ;173344 040601         BIC SP,R1       ;THEN DROP TO XX7752
  920  001746  010111  ;173346 010111         MOV R1,@R1      ;AND STORE IN ITSELF
  921  001750  011102  ;173350 011102  LOOP:  MOV @R1,R2      ;THEN LOAD ADDRESS FOR DATA INSERTION
  922  001752  005214  ;173352 005214         INC @R4         ;AND START DEVICE
  923  001754  105714  ;173354 105714  RDRWAT: TSTB @R4        ;THEN WAIT FOR CHARACTER AVAILABLE
  924  001756  100376  ;173356 100376         BPL RDRWAT      ;HANGING THERE IF NECCESSARY
  925  001760  116412  ;173360 116412         MOVB 2(R4),@R2  ;STORE AWAY DATA BYTE
  926  001762  000002  ;173362 000002
  927  001764  005211  ;173364 005211         INC @R1
  928  001766  120227  ;173366 120227         CMPB R2,#375    ;HAS BRANCH OFFSET BEEN STORED
  929  001770  000375  ;173370 000375
  930  001772  001366  ;173372 001366         BNE LOOP        ;NO
  931  001774  105222  ;173374 105222         INCB (R2)+      ;YES, ALL DONE
  932  001776  END,YA:
  933  001776  000142  ;173376 000142         JMP *(R2)       ;THEN TRANSFER TO RTN
```



```
  934          ; BM873B        BOOTSTRAP       MACY11 27(655)  1-OCT-74  14:50  PAGE 1
  935          ;
  936          ;               ;DATE:  AUG 23, 1974
  937  002000  MAP,YB:
  938          ;THE FOLLOWING IS A REPRODUCTION
  939          ;OF THE ROM PROGRAM FOR BM873YB,
  940          ;IT IS HERE FOR COMPARISON TO THE
  941          ;ACTUAL ROM AND FOR REFERENCE
  942          ;
  943          ;
  944          ;
  945          ;               ;THIS IS THE LOADER TO REPLACE THE FOLLOW
  946          ;               ;M792-YA        PAPER TAPE BOOTSTRAP ROM
  947          ;               ;MR11-DB        BULK STORAGE BOOTSTRAP ROM
  948          ;               ;M792-YH        TA11 CASSETTE BOOTSTRAP ROM
  949          ;               ;RM873A COMBINATION OF ABOVE ROMS
  950          ;
  951          ;               ;PREPHERIAL EXTERNAL PAGE REGISTERS ASSIGNMENTS:
  952          ;
  953          ;       177462  RFWC=   177462          ;WORD COUNT REG, FOR RF1
  954          ;       177406  RKWC=   177406          ;WORD COUNT REG, FOR RK1
  955          ;       177344  TCWC=   177344          ;WORD COUNT REG, FOR TC1
  956          ;       172524  TMWC=   172524          ;BYTE/RECORD COUNT FOR T
  957          ;       176716  RPWC=   176716          ;WORD COUNT REG, FOR RP1
  958          ;       177450  RCWC=   177450          ;WORD COUNT REG, FOR RC1
  959          ;       177560  KLCS=   177560          ;CONTROL REG, FOR KL11
  960          ;       177500  TACS=   177500          ;CONTROL REG, FOR TA11 C
  961          ;       177550  PCCS=   177550          ;CONTROL REG, FOR PC11
  962          ;       172440  TUCS=   172440          ;CONTROL STATUS REG, 1
  963          ;       172442  TUWC=   TUCS+2          ;TU16 WORD COUNT REG,
  964          ;
  965          ;       176300  RHCSA=  176300          ;CONTROLLER REG, 1 FOR R
  966          ;       176302  RHWCA=  RHCSA+2
  967          ;       172040  RSCSA=  172040          ;CONTROLLER REG,1 FOR RH
  968          ;       172042  RSWCA=  RSCSA+2
  969          ;       176700  RPCSA=  176700          ;CONTROLLER REG, 1 FOR R
  970          ;       176702  RPWCA=  RPCSA+2
  971          ;               ;FUNCTION VALUE FOR PREPHERALS:
  972          ;       000005  RFREAD= 5               ;READ FUNCTION
  973          ;       004003  RNUM=   4003            ;REVERSE AND IDENTIFY BL
  974          ;       060017  TMRWND= 60017           ;REWIND AND SET 800 BPI
  975          ;       060011  TMFWRD= 60011           ;FORWARD RECORD COMMAND
  976          ;       060003  TMREAD= 60003           ;TM11 READ
  977          ;       000011  DRCLR=  11              ;DRIVE CLEAR
  978          ;       000071  RHREAD= 71              ;RH11 READ COMMAND
  979          ;       000021  RHPRST= 21              ;READ IN PRESET
  980          ;       000031  TUSPAC= 31              ;SPACE FORWARD COMMAND F
  981          ;       040000  TUTAPE= 40000           ;TAPE BIT IN RH11/RHDT R
  982          ;       001300  TUMODE= 1300            ;800 BPI NORMAL MODE FOR
  983          ;       001000  FCE=    1000            ;FRAME COUNT ERROR BIT
  984          ;               ;CONSOLE SWITCH REG,
  985          ;       177570  CSW=    177570
  986          ;
  987          ;
```

```
                        ;               ;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    @#C$W,@#RSCSA+10;LOAD UNIT # INS
002006  177570 ;173006  177570
002010  172050 ;173010  172050
002012  000401 ;173012  000401          BR      2$
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    @#C$W,R5;SET POINTER TO PARAMETE
002036  177570 ;173036  177570
002040  052705 ;173040  052705          BIS     #10,R5  ;SET POSITION BIT
002042  000010 ;173042  000010
002044  006105 ;173044  006105  1$:     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+1B;LOAD UNIT # INS
002236  177570 ;173236  177570
002240  176310 ;173240  176310
002242  000401 ;173242  000401          BR      2$
002244  010703 ;173244  010703  1$:     MOV     PC,R3
002246  012700 ;173246  012700  2$:     MOV     #RHCSA,R0
002250  176300 ;173250  176300
002252  032760 ;173252  032760  RPCOMN: BIT     #TUTAPE,26(R0);TAPE UNIT?
002254  040000 ;173254  040000
002256  000026 ;173256  000026
002260  001336 ;173260  001336          BNE     TU16RE  ;YES, GO TO TAPE LOGIC
002262  012710 ;173262  012710          MOV     #RHPRST,(R0);RESET DRIVE
002264  000021 ;173264  000021
002266  012760 ;173266  012760          MOV     #14000,32(R0);SET 16 BIT FORMAT
002270  014000 ;173270  014000
002272  000032 ;173272  000032
002274  012710 ;173274  012710          MOV     #DRCLR,(R0);CLEAR DRIVE ERROR
002276  000011 ;173276  000011
                        ;                       ; (GENERATED IF RS03/04
002300  005720 ;173300  005720  RHCOMN: TST     (R0)+   ;MOVE TO WORD COUNT ADDR
002302  010037 ;173302  010037          MOV     R0,@#2  ;FAKE CALLING SEQUENCE
002304  000002 ;173304  000002
002306  012737 ;173306  012737          MOV     #RHREAD,@#4
002310  000071 ;173310  000071
002312  000004 ;173312  000004
002314  005002 ;173314  005002          CLR     R2      ;FOR FLAG AND POINTER TO
002316  000430 ;173316  000430          BR      OTHERA
                        ;               .DSABL  LSB
                        ;               ;
                        ;               ;THIS IS THE STARTING ADDRESS FOR RH11/RP04 DISK
002320  000405 ;173320  000405  RHRPA:  BR      1$      ;ENTRY FOR SELECT UNIT 0
002322  010703 ;173322  010703  RHRPB:  MOV     PC,R3   ;ENTRY TO SELECT ALL UNI
002324  113737 ;173324  113737          MOVB    @#CSW,@#RPCSA+1B;LOAD UNIT # INS
002326  177570 ;173326  177570
002330  176710 ;173330  176710
002332  000401 ;173332  000401          BR      2$
002334  010703 ;173334  010703  1$:     MOV     PC,R3
002336  012700 ;173336  012700  2$:     MOV     #RPCSA,R0
002340  176700 ;173340  176700
002342  000743 ;173342  000743          BR      RPCOMN
                        ;               ;
                        ;               ;ENTRY TO BRANCH TO THE PC SELECTED BY CONSOL SW
002344  013707 ;173344  013707  CSRGO:  MOV     @#CSW,PC
002346  177570 ;173346  177570
                        ;               ;
                        ;               ;
                        ;               ;
                        ;               ;THIS IS THE STARTING ADDRESS FOR RP11 CONTROLLE
002350  000405 ;173350  000405  RP11A:  BR      1$      ;ENTRY TO SELECT UNIT 0
002352  010703 ;173352  010703  RP11B:  MOV     PC,R3   ;ENTRY TO SELECT ALL UNI
```



```
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
                        ;               ;
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
```

```
1204  002514  000443 ;173514  000443          BR      CKDEV   ;AND TRANSFER TO READER
1205          ;
1206          ;               ;
1207          ;
1208          ;               ;CASSETTE TAPE DEVICE COMMAND TABLE
1209  002516  .BYTE   240 ;173516        240   TABLE:  .BYTE   240     ;COMPARE WORD NOT A COMM
1210  002517  .BYTE   037 ;173517        037           .BYTE   37      ;ILBS+RWD+GO
1211  002520  .BYTE   015 ;173520        015           .BYTE   15      ;SPACE FORWARD BLOCK+GO
1212  002521  .BYTE   005 ;173521        005           .BYTE   5       ;READ
1213  002522  .BYTE   024 ;173522        024           .BYTE   24      ;READ +ILBS
1214  002523  .BYTE   224 ;173523        224           .BYTE   224     ;READ+ILBS+END FLAG
1215          ;
1216          ;               ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE D
1217  002524  000404 ;173524  000404  CBOOTA: BR      1$      ;SELECT UNIT 0
1218  002526  113704 ;173526  113704  CBOOTB: MOVB    @#CSW,R4;SELECT UNITS
1219  002530  177570 ;173530  177570
1220  002532  000304 ;173532  000304          SWAB    R4
1221  002534  000401 ;173534  000401          BR      RESETX
1222  002536  005004 ;173536  005004  1$:     CLR     R4
1223  002540  012700 ;173540  012700  RESETX: MOV     #TACS,R0;GET CONTROL REG.
1224  002542  177500 ;173542  177500
1225  002544  000005 ;173544  000005  RESTRT: RESET
1226  002546  010410 ;173546  010410          MOV     R4,(R0) ;SELECT UNIT
1227  002550  012701 ;173550  012701          MOV     #TABLE,R1
1228  002552  173516 ;173552  173516
1229  002554  012702 ;173554  012702          MOV     #375,R2 ;LOAD TRANSFER COUNTER
1230  002556  000375 ;173556  000375
1231  002560  112103 ;173560  112103          MOVB    (R1)+,R3;LOAD COMPARATOR
1232  002562  112110 ;173562  112110  LOOP1:  MOVB    (R1)+,(R0);LOAD COMMAND
1233  002564  100407 ;173564  100407          BMI     DONE
1234  002566  130310 ;173566  130310  LOOP2:  BITB    R3,(R0) ;COMMAND COMPLETE?
1235  002570  001776 ;173570  001776          BEQ     LOOP2   ;NO, WAIT
1236  002572  105202 ;173572  105202          INCB    R2      ;INCREMENT ADDRESS CTR,
1237  002574  100772 ;173574  100772          BMI     LOOP1   ;IF (=), GET COMMAND
1238  002576  116012 ;173576  116012          MOVB    2(R0),(R2);STORE DATA
1239  002600  000002 ;173600  000002
1240  002602  000771 ;173602  000771          BR      LOOP2   ;GET ANOTHER BYTE
1241  002604  005710 ;173604  005710  DONE:   TST     (R0)    ;ANY ERROR?
1242  002606  100756 ;173606  100756          BMI     RESTRT  ;YES, RETRY
1243  002610  005002 ;173610  005002          CLR     R2      ;CLEAR COMPARE ADDRESS
1244  002612  120312 ;173612  120312          CMPB    R3,(R2) ;IT MUST BE 240
1245  002614  001377 ;173614  001377          BNE     .
1246  002616  000112 ;173616  000112  ERROR:  JMP     (R2)
1247          ;
1248          ;               ;THIS IS THE STARTING ADDRESS FOR THE PC11 CONTR
1249  002620  012704 ;173620  012704  PC11:   MOV     #PCCS,R4
1250  002622  177550 ;173622  177550
1251  002624  000005 ;173624  000005  CKDEV:  RESET
1252  002626  012701 ;173626  012701          MOV     #160000,R1;SET UP MEMORY TEST LI
1253  002630  160000 ;173630  160000
1254  002632  012702 ;173632  012702          MOV     #6,R2   ;SET UP POINTER TO TIMEO
1255  002634  000006 ;173634  000006
1256  002636  012712 ;173636  012712          MOV     #340,(R2);SET UP VECTOR TO RETUR
1257  002640  000340 ;173640  000340
```



```
1258  002642  010742 ;173642  010742          MOV     PC,-(R2);SAVE PC
1259  002644  012706 ;173644  012706          MOV     #24,SP  ;LOAD UP STACK POINTER
1260  002646  000024 ;173646  000024
1261  002650  010441 ;173650  010441          MOV     R4,-(R1);LOOK FOR END OF MEMORY
1262  002652  040601 ;173652  040601          BIC     SP,R1   ;THEN DROP TO XX752
1263  002654  010111 ;173654  010111          MOV     R1,(R1) ;AND STORE IN ITSELF
1264  002656  011102 ;173656  011102  LOOP:   MOV     (R1),R2
1265  002660  005214 ;173660  005214          INC     (R4)    ;START DEVICE
1266  002662  105714 ;173662  105714  RDRWAT: TSTB    (R4)    ;WAIT
1267  002664  100376 ;173664  100376          BPL     RDRWAT
1268  002666  116412 ;173666  116412          MOVB    2(R4),(R2);SAVE THE DATA
1269  002670  000002 ;173670  000002
1270  002672  005211 ;173672  005211          INC     (R1)
1271  002674  120227 ;173674  120227          CMPB    R2,#375
1272  002676  000375 ;173676  000375
1273  002700  001366 ;173700  001366          BNE     LOOP    ;NO
1274  002702  105222 ;173702  105222          INCB    (R2)+   ;YES
1275  002704  000142 ;173704  000142          JMP     -(R2)
1276  002706  000000 ;173706  000000  ;THIS AREA IS UNUSED
1277  002710  000000 ;173710  000000  ;THIS AREA IS UNUSED
1278  002712  000000 ;173712  000000  ;THIS AREA IS UNUSED
1279  002714  000000 ;173714  000000  ;THIS AREA IS UNUSED
1280  002716  000000 ;173716  000000  ;THIS AREA IS UNUSED
1281  002720  000000 ;173720  000000  ;THIS AREA IS UNUSED
1282  002722  000000 ;173722  000000  ;THIS AREA IS UNUSED
1283  002724  000000 ;173724  000000  ;THIS AREA IS UNUSED
1284  002726  000000 ;173726  000000  ;THIS AREA IS UNUSED
1285  002730  000000 ;173730  000000  ;THIS AREA IS UNUSED
1286  002732  000000 ;173732  000000  ;THIS AREA IS UNUSED
1287  002734  000000 ;173734  000000  ;THIS AREA IS UNUSED
1288  002736  000000 ;173736  000000  ;THIS AREA IS UNUSED
1289  002740  000000 ;173740  000000  ;THIS AREA IS UNUSED
1290  002742  000000 ;173742  000000  ;THIS AREA IS UNUSED
1291  002744  000000 ;173744  000000  ;THIS AREA IS UNUSED
1292  002746  000000 ;173746  000000  ;THIS AREA IS UNUSED
1293  002750  000000 ;173750  000000  ;THIS AREA IS UNUSED
1294  002752  000000 ;173752  000000  ;THIS AREA IS UNUSED
1295  002754  000000 ;173754  000000  ;THIS AREA IS UNUSED
1296  002756  000000 ;173756  000000  ;THIS AREA IS UNUSED
1297  002760  000000 ;173760  000000  ;THIS AREA IS UNUSED
1298  002762  000000 ;173762  000000  ;THIS AREA IS UNUSED
1299  002764  000000 ;173764  000000  ;THIS AREA IS UNUSED
1300  002766  000000 ;173766  000000  ;THIS AREA IS UNUSED
1301  002770  000000 ;173770  000000  ;THIS AREA IS UNUSED
1302  002772  000000 ;173772  000000  ;THIS AREA IS UNUSED
1303  002774  000000 ;173774  000000  ;THIS AREA IS UNUSED
1304  002776  END,YR:
1305  002776  000000 ;173776  000000  ;THIS AREA IS UNUSED
```

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



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

```
1414  003174  011041  ;173174 011041          MOV @R0,-(R1)    ;AND PUT COMMAND TO DEVICE
1415  003176  105711  ;173176 105711  OTHWAT: TSTB @R1          ;WAIT FOR DONE FLAG
1416  003200  100376  ;173200 100376          BPL OTHWAT       ;BY HANGING IN LOOP
1417  003202  005711  ;173202 005711          TST @R1          ;THEN TEST FOR ERROR
1418  003204  100441  ;173204 100441          BMI ERROR        ;GOT PROBLEMS
1419  003206  005007  ;173206 005007          CLR PC           ;AND TRANSFER TO ZERO
1420                                          ;
1421                                          ;THIS IS THE STARTING ADDRESS FOR THE PC11 PAPER TAPE CONTROLLER
1422  003210  012704  ;173210 012704  KL11:   MOV #177560,R4   ;OBTAIN DEVICE ADDRESS
1423  003212  177560  ;173212 177560
1424  003214  000440  ;173214 000440          BR CKDEV         ;AND TRANSFER TO READER SERVICE ROUTINE
1425                                          ;
1426
1427                                          ;THIS IS THE CASSETTE DEVICE COMMAND TABLE
1428  003216  017640  ;173216    240  TABLE:  .BYTE 240        ;COMPARE WORD NOT A COMMAND
1429                  ;173217    037          .BYTE 37         ;ILBS+RWD+GO
1430  003220  002415  ;173220    015          .BYTE 15         ;SPACE FORWARD BLOCK+GO
1431                  ;173221    005          .BYTE 5          ;READ+GO
1432  003222  112024  ;173222    024          .BYTE 24         ;READ+ILBS
1433                  ;173223    224          .BYTE 224        ;READ+ILBS+END FLAG
1434          ;NOTE 773024 AND 773224 ARE DEPENDENT ON OFFSET IN DIODES FOR LINE 1
1435                                          ;
1436                                          ;THIS IS AN ADDITIONAL POWER VECTOR ADDRESS REQUIRED BY DEVICE
1437  003224  173000  ;173224 173000  POWER2: .WORD RF11        ;ADDRESS OF BEGINNING OF BOOTSTRAP
1438  003226  000340  ;173226 000340          .WORD 340        ;PRIORITY LEVEL 7
1439                                          ;
1440                                          ;THIS IS THE STARTING ADDRESS FOR THE CASSETTE DEVICE #0
1441  003230  005004  ;173230 005004  CBOOT:  CLR R4            ;LOAD DEVICE NUMBER 0 IN R4
1442  003232  012700  ;173232 012700  RESTX:  MOV #177500,R0    ;GET DEVICE ADDTESS
1443  003234  177500  ;       177500
1444  003236  000005  ;173236 000005  RESTRT: RESET            ;ISSUE RESET INSTRUCTION
1445  003240  010410  ;173240 010410          MOV R4,@R0       ;LOAD DEVICE WITH UNIT NUMBER
1446  003242  012701  ;173242 012701          MOV #TABLE,R1    ;GET FUNNY TABLE OF INSTRUCTIONS
1447  003244  173216  ;173244 173216
1448  003246  012702  ;173246 012702          MOV #375,R2      ;AND LOAD UP TRANSFER COUNTER
1449  003250  000375  ;173250 000375
1450  003252  112103  ;173252 112103          MOVB (R1)+,R3    ;THE LOAD UP COMPARATOR
1451  003254  112110  ;173254 112110  LOOP1:  MOVB (R1)+,@R0   ;LOAD DEVICE REGISTER WITH COMMAND
1452  003256  100407  ;173256 100407          BMI DONE
1453  003260  130310  ;173260 130310  LOOP2:  BITB R3,@R0      ;HAS COMMAND COMPLETED
1454  003262  001776  ;173262 001776          BEQ LOOP2        ;NO, WAIT
1455  003264  105202  ;173264 105202          INCB R2          ;THEN INCREMENT ADDRESS CTR
1456  003266  100772  ;173266 100772          BMI LOOP1        ;IF NEGATIVE, GET COMMAND
1457  003270  116012  ;173270 116012          MOVB 2(R0),@R2   ;AND STORE DATA AWAY
1458  003272  000002  ;173272 000002
1459  003274  000771  ;173274 000771          BR LOOP2         ;GO GET ANOTHER BYTE
1460  003276  005710  ;173276 005710  DONE:   TST @R0          ;ANY DEVICE ERRORS
1461  003300  100756  ;173300 100756          BMI RESTRT       ;YES, RETRY
1462  003302  005002  ;173302 005002          CLR R2           ;CLEAR COMPARE ADDRESS AND TRANSFER ADDRESS
1463  003304  120312  ;173304 120312          CMPB R3,@R2      ;IT MUST BE 240
1464  003306  001377  ;173306 001377          BNE .+0          ;NO, THERE WAS AN ERROR
1465  003310  000112  ;173310 000112  ERROR:  JMP @R2          ;NORMAL CASSETTE AND ERROR FOR BULK STORAGE
1466                                          ;
1467                                          ;THIS IS THE STARTING LOCATION FOR THE PC11 CONTROLLER
```



```
1468  003312  012704  ;173312 012704  PC11:   MOV #177550,R4  ;LOAD DEVICE ADDRESS
1469  003314  177550  ;173314 177550
1470  003316  000005  ;173316 000005  CKDEV:  RESET            ;KILL ALL DEVICE ACTION
1471  003320  012701  ;173320 012701          MOV #160000,R1   ;THEN SET UP MEMORY TEST LIMITS
1472  003322  160000  ;173322 160000
1473  003324  012702  ;173324 012702          MOV #6,R2        ;AND SET UP POINTER TO TIMEOUT LOCATION
1474  003326  000006  ;173326 000006
1475  003330  012712  ;173330 012712          MOV #340,@R2     ;AND SET UP VECTOR TO RETURN TO NEXT
1476  003332  000340  ;173332 000340
1477  003334  010742  ;173334 010742          MOV PC,-(R2)     ;SAVE THE PC
1478  003336  012706  ;173336 012706          MOV #24,SP       ;AND LOAD UP STACK POINTER
1479  003340  000024  ;173340 000024
1480  003342  010441  ;173342 010441          MOV R4,-(R1)     ;AND LOOK FOR END OF MEMORY
1481  003344  040601  ;173344 040601          BIC SP,R1        ;THEN DROP TO XX7752
1482  003346  010111  ;173346 010111          MOV R1,@R1       ;AND STORE IN ITSELF
1483  003350  011102  ;173350 011102  LOOP:   MOV @R1,R2       ;THEN LOAD ADDRESS FOR DATA INSERTION
1484  003352  005214  ;173352 005214          INC @R4          ;AND START DEVICE
1485  003354  105714  ;173354 105714  RDRWAT: TSTB @R4         ;THEN WAIT FOR CHARACTER AVAILABLE
1486  003356  100376  ;173356 100376          BPL RDRWAT       ;HANGING THERE IF NECCESSARY
1487  003360  116412  ;173360 116412          MOVB 2(R4),@R2   ;STORE AWAY DATA BYTE
1488  003362  000002  ;173362 000002
1489  003364  005211  ;173364 005211          INC @R1
1490  003366  120227  ;173366 120227          CMPB R2,#375     ;HAS BRANCH OFFSET BEEN STORED
1491  003370  000375  ;173370 000375
1492  003372  001366  ;173372 001366          BNE LOOP         ;NO
1493  003374  105222  ;173374 105222          INCB (R2)+       ;YES, ALL DONE
1494  003376  000142  ;173376 000142          JMP -(R2)        ;THEN TRANSFER TO RTN
1495
1496          ;THE FOLLOWING 400 LOCATIONS ARE
1497          ;A REPRODUCTION OF THE DDCMP BOOT-
1498          ;STRAP ROM, IT IS HERE FOR COM-
1499          ;PARISON TO THE ACTUAL ROM AND
1500          ;FOR REFERENCE.
1501                                          ;
1502                                          ;       COPYRIGHT  1975, DIGITAL EQUIPMENT CORP., MAYNARD,MASS. 01754
1503                                          ;
1504                                          ;       THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A
1505                                          ;       LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND
1506                                          ;       CAN BE COPIED (WITH INCLUSION OF DEC'S
1507                                          ;       COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM,
1508                                          ;       EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING
1509                                          ;       BY DEC.
1510                                          ;
1511                                          ;       THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO
1512                                          ;       CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED
1513                                          ;       AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.
1514                                          ;
1515                                          ;       DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR
1516                                          ;       RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH
1517                                          ;       IS NOT SUPPLIED BY DEC.
1518                                          ;
1519                                          ;       VERSION 01
1520                                          ;
1521                                          ;       STUART WECKER  01/22/75
```

```
1522                                    ;
1523                                    ;       DIGITAL EQUIPMENT CORPORATION
1524                                    ;       COMPUTER NETWORK FACILITIES
1525                                    ;       DOWN-LINE LOADING PROGRAM
1526                                    ;
1527                                    ;       THIS PROGRAM LOADS COMPUTER MEMORY FROM DATA SENT OVER
1528                                    ;       A DATA COMMUNICATIONS LINK, IT SENDS AND RECEIVES
1529                                    ;       MESSAGES IN DDCMP BOOT FORMAT.  THE PRIMARY BOOT ONLY
1530                                    ;       LOADS A SINGLE BLOCK, THE SECONDARY BOOT,  WHICH
1531                                    ;       THEN REQUESTS AND LOADS THE DESIRED PROGRAM.
1532                                    ;
1533                                    ;       CURRENT VERSION DDCMP: 3.0 - MAY 7,1974
1534                                    ;
1535                                    ;       THE BOOTSTRAP MESSAGES ARE OF THE FORM:
1536                                    ;
1537                                    ;       SYN,SYN,DLE,CNT,F,S,FILL,FILL,ADDR,CRC1,DATA,CRC2
1538                                    ;
1539                                    ;       ALL ITEMS ARE 8-BITS LONG UNLESS OTHERWISE SPECIFIED
1540                                    ;
1541                                    ;       SYN=THE SYNC CHARACTER=SYNC-226,ASYNC-377
1542                                    ;       DLE=THE BOOT HEADER CHARACTER=OCTAL 220
1543                                    ;       CNT=THE 14-BIT COUNT FIELD=LENGTH OF DATA FIELD
1544                                    ;       F=THE FINAL BIT=LINK CONTROL
1545                                    ;       S=THE SELECT BIT=LINK CONTROL
1546                                    ;       FILL=A FILL CHARACTER=OCTAL 000
1547                                    ;       ADDR=THE STATION ADDR=FOR PT. TO PT.=1
1548                                    ;       CRC1=THE 16-BIT CRC=16 COMPUTED ON DLE THROUGH ADDR
1549                                    ;       DATA=THE BOOT DATA AS FOLLOWS:
1550                                    ;               CODE,INFO
1551                                    ;       ONLY THE FOLLOWING CODES ARE USED BY THE
1552                                    ;       PRIMARY BOOT
1553                                    ;       CODE=10  REQUEST SECONDARY PROGRAM
1554                                    ;       INFO=DEVICE TYPE,STATION ADDRESS
1555                                    ;       DEVICE TYPE=DP=0,DU=2,DL=4,DQ=6
1556                                    ;       STATION ADDRESS=1
1557                                    ;       CODE=0  PROGRAM LOAD WITH TRANSFER ADDRESS
1558                                    ;       INFO=BLKNO,BLK LDADDR,IMAGE DATA,TRANS ADDR
1559                                    ;       BLKNO=0
1560                                    ;       BLOCK LDADDR=6
1561                                    ;       TRANS ADDR=6
1562                                    ;       HEADER COUNT > OR = TO 10.
1563                                    ;       ADDRESSES ARE 4 BYTES=32 BITS=LOW BIT FIRST
1564                                    ;       CRC2=THE 16-BIT CRC=16 COMPUTED ON THE DATA FIELD ONLY
1565                                    ;
1566                                    ;
1567                                    ;       OPTION SWITCHES:
1568                                    ;               DEVICE=DP11,DU11,DL11
1569                                    ;               CRC=KG11,SCRC
1570                                    ;
1571
1572                                    ;
1573                                    ;       REGISTER DEFINITIONS
1574                                    ;
1575                    ;       000000  R0=%0                   ;BLOCK LOAD ADDR
```



```
1576                    ;       000001  R1=%1                   ;DEVICE CSR ADDRESS
1577                    ;       000002  R2=%2                   ;CRC CALC TEMP
1578                    ;       000003  R3=%3                   ;SOFTWARE CRC
1579                    ;       000004  R4=%4                   ;BLOCK CHAR COUNT
1580                    ;       000005  R5=%5                   ;CRC CALC TEMP
1581                    ;       000006  SP=%6                   ;STACK ADDR
1582                    ;       000007  PC=%7                   ;LOCATION COUNTER
1583                                    ;
1584                                    ;       LITERALS
1585                                    ;
1586                    ;       000001  $STADR=1                ;STATION ADDR
1587                    ;       177570  $SWR=177570             ;SWITCH REGISTER ADDR
1588                    ;       000226  $SYN=226                ;SYNC CHARACTER
1589                    ;       000220  $DLE=220                ;DDCMP DLE CHARACTER
1590                    ;       000400  $STRIP=400
1591                                    ;
1592                                    ;       THE STACK IS USED AS FOLLOWS:
1593                                    ;       STACK-2:FOR JSR TO GET ROUTINE
1594                                    ;       STACK-4:TEMP FOR CRC CALCULATION
1595                                    ;
1596                                    ;       START OF BOOT PROGRAM
1597                                    ;
1598                                    ;       START1=DEVICE UNIT 0=NORMAL CONFIGURATION
1599                                    ;       START2=USE SWITCH REG AS DEVICE DISPLACEMENT
1600                                    ;               I.E. #0=0,#1=10,#2=20
1601                                    ;
1602                    ;       173400  .=173400
1603  003400  012700  ;173400 012700   START1: MOV     (PC)+,R0        ;NON ZERO VALUE TO R0
1604  003402  005000  ;173402 005000   START2: CLR     R0              ;CLEAR R0
1605  003404  000005  ;173404 000005           RESET                   ;RESET SYS,MEM MGT, ETC...
1606  003406  012706  ;173406 012706           MOV     #17776,SP       ;STACK AT 4K-2
1607  003410  017776  ;173410 017776
1608                                    ;
1609                                    ;       FIND THE DU-11 IN THE FLOATING ADDRESS SPACE
1610                                    ;
1611  003412  010702  ;173412 010702           MOV     PC,R2           ;CURRENT PC
1612  003414  062702  ;173414 062702           ADD     #DEVTAB-.,R2    ;DEVICE TABLE ADDR
1613  003416  000360  ;173416 000360
1614  003420  012703  ;173420 012703           MOV     #6,R3           ;TRAP PS ADDR
1615  003422  000006  ;173422 000006
1616  003424  005013  ;173424 005013           CLR     (R3)            ;CLEAR NEW PS
1617  003426  010243  ;173426 010243           MOV     R2,-(R3)        ;TABLE ADDR TO LOC 4
1618  003430  160313  ;173430 160313           SUB     R3,(R3)         ;SUB TO TRAP RTN
1619  003432  005303  ;173432 005303           DEC     R3              ;LEAVE CNT 3 FOR LOOP
1620  003434  012701  ;173434 012701           MOV     #160010,R1      ;START SEARCH ADDR
1621  003436  160010  ;173436 160010
1622  003440  005711  ;173440 005711   DEVLOP: TST     (R1)            ;IS DEVICE THERE
1623  003442  111204  ;173442 111204           MOVB    (R2),R4         ;DEVICE INCREMENT TO R3
1624  003444  060401  ;173444 060401           ADD     R4,R1           ;UPDATE TO NEXT DEVICE
1625  003446  005201  ;173446 005201           INC     R1              ;INCREMENT MODULO
1626  003450  040401  ;173450 040401           BIC     R4,R1           ;CLEAR EXCESS
1627  003452  005703  ;173452 005703           TST     R3              ;TEST FOR DONE
1628  003454  001371  ;173454 001371           BNE     DEVLOP          ;NOT YET
1629  003456  005700  ;173456 005700           TST     R0              ;TEST SWITCH REG USE
```

```
1630  003460  001002  ;173460 001002          BNE     SNDREQ          ;NO SWITCH REG
1631  003462  063701  ;173462 063701          ADD     @#SWR,R1        ;ADD SWR VALUE
1632  003464  177570  ;173464 177570
1633                                  ;
1634                                  ;       SET UP DEVICE FOR OUTPUT
1635                                  ;
1636                  ;173466         SNDREQ:
1637  003466  012711  ;173466 012711          MOV     #6,(R1)         ;DATA TERM RDY AND REQ TO SEND
1638  003470  000006  ;173470 000006
1639  003472  012761  ;173472 012761          MOV     #36000+#SYN,2(R1) ;SET SYNC REGISTER
1640  003474  036226  ;173474 036226
1641  003476  000002  ;173476 000002
1642  003500  032711  ;173500 032711  L3:     BIT     #20000,(R1)     ;TEST CLEAR TO SEND
1643  003502  020000  ;173502 020000
1644  003504  001775  ;173504 001775          BEQ     L3              ;NOT YET
1645  003506  022121  ;173506 022121          CMP     (R1)+,(R1)+     ;MOVE PTR TO XMIT TSR
1646  003510  052711  ;173510 052711          BIS     #20,(R1)        ;TURN SEND ON
1647  003512  000020  ;173512 000020
1648                                  ;
1649                                  ;       SEND SECONDARY PGM REQUEST MESSAGE
1650                                  ;
1651  003514  010700  ;173514 010700          MOV     PC,R0           ;CURRENT PC
1652  003516  062700  ;173516 062700          ADD     #RQMSG-.,R0     ;REQUEST MSG ADDR
1653  003520  000230  ;173520 000230
1654  003522  012704  ;173522 012704          MOV     #RQMSGE-RQMSG,R4 ;COUNT
1655  003524  000026  ;173524 000026
1656  003526  112061  ;173526 112061  L4:     MOVB    (R0)+,2(R1)     ;CHAR TO XMIT REGISTER
1657  003530  000002  ;173530 000002
1658  003532  105711  ;173532 105711  L5:     TSTB    (R1)            ;DONE YET ?
1659  003534  100376  ;173534 100376          BPL     L5              ;NO
1660  003536  005304  ;173536 005304          DEC     R4              ;DECREMENT COUNT
1661  003540  001372  ;173540 001372          BNE     L4              ;ONCE MORE
1662  003542  042711  ;173542 042711          BIC     #20,(R1)        ;DROP SEND
1663  003544  000020  ;173544 000020
1664  003546  024141  ;173546 024141          CMP     -(R1),-(R1)     ;RESET PTR TO RCV CSR
1665                                  ;
1666                                  ;       GET SECONDARY PROGRAM
1667                                  ;
1668                  ;173550         GETPGM:
1669  003550  042711  ;173550 042711          BIC     #20,(R1)        ;CLEAR SEARCH SYNC
1670  003552  000020  ;173552 000020
1671  003554  012711  ;173554 012711          MOV     #422,(R1)       ;SET FOR CLEAR AND STRIP SYNC
1672  003556  000422  ;173556 000422
1673  003560  005003  ;173560 005003          CLR     R3              ;CLEAR CRC VALUE
1674                                  ;
1675                                  ;       GET MESSAGE HEADER AND CHECK IT
1676                                  ;
1677  003562  012700  ;173562 012700          MOV     #1,R0           ;LOAD HDR AT LOC. 1
1678  003564  000001  ;173564 000001
1679  003566  012704  ;173566 012704          MOV     #8.,R4          ;BLOCK COUNT
1680  003570  000010  ;173570 000010
1681  003572  004767  ;173572 004767          JSR     PC,GET          ;GET HEADER
1682  003574  000060  ;173574 000060
1683  003576  005703  ;173576 005703          TST     R3              ;CHECK HEADER CRC
```



```
1684  003600  001363  ;173600 001363          BNE     GETPGM          ;NO GOOD
1685  003602  123727  ;173602 123727          CMPB    @#6,@#STADR     ;CHECK FOR MY ADDR
1686  003604  000006  ;173604 000006
1687  003606  000001  ;173606 000001
1688  003610  001357  ;173610 001357          BNE     GETPGM          ;NOT MINE
1689  003612  123727  ;173612 123727          CMPB    @#1,@#DLE       ;IS THIS A DLE MSG
1690  003614  000001  ;173614 000001
1691  003616  000220  ;173616 000220
1692  003620  001322  ;173620 001322          BNE     SNDREQ          ;NO, ASK FOR ONE
1693                                  ;
1694                                  ;       GET DATA BLOCK
1695                                  ;
1696  003622  013704  ;173622 013704          MOV     @#2,R4          ;DATA FIELD LENGTH
1697  003624  000002  ;173624 000002
1698  003626  042704  ;173626 042704          BIC     #140000,R4      ;MASK OFF S,F BITS
1699  003630  140000  ;173630 140000
1700  003632  122424  ;173632 122424          CMPB    (R4)+,(R4)+     ;ADD 2 FOR CRC
1701  003634  005000  ;173634 005000          CLR     R0              ;LOAD INTO LOCATION 0
1702  003636  004767  ;173636 004767          JSR     PC,GET1         ;GET DATA BLOCK
1703  003640  000014  ;173640 000014
1704  003642  005703  ;173642 005703          TST     R3              ;CHECK DATA FIELD CRC
1705  003644  001310  ;173644 001310          BNE     SNDREQ          ;NO GOOD
1706  003646  105713  ;173646 105713          TSTB    (R3)            ;CHECK CODE IN LOC 0
1707  003650  001306  ;173650 001306          BNE     SNDREQ          ;NOT PROGRAM LOAD
1708  003652  000137  ;173652 000137          JMP     @#6             ;TRANSFER TO SECONDARY PGM
1709  003654  000006  ;173654 000006
1710                                  ;
1711                                  ;       GET A BLOCK AND COMPUTE CRC
1712                                  ;
1713                  ;173656         GET:
1714                  ;173656         GET1:
1715  003656  105711  ;173656 105711          TSTB    (R1)            ;IS DEVICE DONE YET
1716  003660  100376  ;173660 100376          BPL     GET             ;NOT YET
1717  003662  042711  ;173662 042711          BIC     @#STRIP,(R1)    ;NO STRIP SYNC
1718  003664  000400  ;173664 000400
1719  003666  116110  ;173666 116110          MOVB    2(R1),(R0)      ;STORE IT
1720  003670  000002  ;173670 000002
1721                                  ;
1722                                  ;       CRC CALCULATION ROUTINE
1723                                  ;
1724                  ;       120001  POLY=120001                     ;CRC-16 POLYNOMIAL
1725                                  ;
1726  003672  012705  ;173672 012705          MOV     #8.,R5          ;BYTE LENGTH
1727  003674  000010  ;173674 000010
1728  003676  112002  ;173676 112002          MOVB    (R0)+,R2        ;CHARACTER TO ADD TO CRC
1729  003700  000241  ;173700 000241  CRCLOP: CLC                     ;CLEAR CARRY
1730  003702  006003  ;173702 006003          ROR     R3              ;SHIFT OLD PARTIAL
1731  003704  103003  ;173704 103003          BCC     L10             ;IF CLEAR CHECK CHAR
1732  003706  006002  ;173706 006002          ROR     R2              ;SHIFT CHARACTER
1733  003710  103003  ;173710 103003          BCC     L11             ;XOR POLY
1734  003712  000410  ;173712 000410          BR      L12             ;NEXT BIT
1735  003714  006002  ;173714 006002  L10:    ROR     R2              ;SHIFT CHARACTER
1736  003716  103006  ;173716 103006          BCC     L12             ;NEXT BIT
1737  003720  012746  ;173720 012746  L11:    MOV     #POLY,-(SP)     ;POLY TO STACK
```

```
1738  003722  120001  ;173722 120001
1739  003724  040316  ;173724 040316          BIC     R3,(SP)         ;NOT PARTIAL AND POLY
1740  003726  042703  ;173726 042703          BIC     #POLY,R3        ;NOT POLY AND PARTIAL
1741  003730  120001  ;173730 120001
1742  003732  052603  ;173732 052603          BIS     (SP)+,R3        ;POLY XOR PARTIAL
1743  003734  005305  ;173734 005305  L12:    DEC     R5              ;DECREMENT BIT COUNT
1744  003736  001360  ;173736 001360          BNE     CRCLOP          ;ONCE MORE
1745  003740  005304  ;173740 005304          DEC     R4              ;DECREMENT COUNT
1746  003742  001345  ;173742 001345          BNE     GET             ;ONCE MORE
1747  003744  000207  ;173744 000207          RTS     PC              ;RETURN
1748                                  ;
1749                                  ;       SECONDARY PROGRAM REQEST MSG
1750                                  ;
1751  003746  113226  ;173746 113226  RQMSG:  .BYTE   $SYN,$SYN,$SYN,$SYN
1752  003750  113226  ;173750 113226
1753  003752  002220  ;173752 002220          .BYTE   $DLE,4,0,0,0,1
1754  003754  000000  ;173754 000000
1755  003756  000400  ;173756 000400
1756  003760  050055  ;173760 050055          .BYTE   55,120
1757  003762  001010  ;173762 001010          .BYTE   10              ;REQ SEC PGM CODE
1758                  ;                       .BYTE   2               ;DEVICE CODE
1759  003764  000001  ;173764 000001          .BYTE   $STADR          ;STATION ADDR
1760                  ;                       .BYTE   0               ;FILL
1761  003766  030242  ;173766 030242          .BYTE   242,60          ;FOR STADR=1
1762                                  ;       NOTE:NODEV AND DEVTAB MUST BE IN THIS ORDER
1763                                  ;       DO NOT SEPARATE THEM
1764                                  ;
1765                                  ;       .EVEN
1766  003770  122243  ;173770 122243  NODEV:  CMPB    (R2)+,-(R3)     ;INC PTR=DEC CNT
1767  003772  000002  ;173772 000002          RTI                     ;RETURN FROM TRAP
1768                                  ;
1769                  ;173774         RQMSGE:                         ;END OF MSG-USE JUNK AS PADS
1770  003774  007407  ;173774 007407  DEVTAB: .BYTE   7               ;DJ=11
1771                  ;                       .BYTE   17              ;DH=11
1772  003776  END,YC:
1773  003776  003407  ;173776 003407          .BYTE   7               ;DQ=11
1774                  ;                       .BYTE   7               ;DU=11
1775                  ;174000         END:
1776                  ;       173400          .END    START1
```



```
1777  004000  MAP,YD:
1778          ;THE FOLLOWING IS A REPRODUCTION
1779          ;OF THE ROM PROGRAM FOR BM873YD,
1780          ;IT IS HERE FOR COMPARISON TO THE
1781          ;ACTUAL ROM AND FOR REFERENCE
1782          ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
1783          ;BM873-YD,P11
1784
1785                                          ;       This code is to be blasted into PROMs on the BM873-YD board.
1786                                          ;       Written by David M. Rosenberg           October 1974
1787                                          ;REGISTER DEFINITIONS
1788
1789          ;       000000  R0=%0                   ;GENERAL PURPOSE REGISTER 0
1790          ;       000001  R1=%1                   ;GENERAL PURPOSE REGISTER 1
1791          ;       000002  R2=%2                   ;GENERAL PURPOSE REGISTER 2
1792          ;       000003  R3=%3                   ;GENERAL PURPOSE REGISTER 3
1793          ;       000004  R4=%4                   ;GENERAL PURPOSE REGISTER 4
1794          ;       000005  R5=%5                   ;GENERAL PURPOSE REGISTER 5
1795          ;       000006  SP=%6                   ;STACK POINTER (REGISTER R6)
1796          ;       000007  PC=%7                   ;PROGRAM COUNTER (REGISTER R7)
1797
1798
1799                                          ;SYMBOL DEFINITIONS
1800
1801          ;       177776  PS=177776               ;PROCESSOR STATUS REGISTER
1802          ;       177570  SWR=177570             ;FRONT PANEL SWITCH REGISTER
1803          ;       000000  PR0=0*40                ;PRIORITY LEVEL 0
1804          ;       000040  PR1=1*40                ;PRIORITY LEVEL 1
1805          ;       000100  PR2=2*40                ;PRIORITY LEVEL 2
1806          ;       000140  PR3=3*40                ;PRIORITY LEVEL 3
1807          ;       000200  PR4=4*40                ;PRIORITY LEVEL 4
1808          ;       000240  PR5=5*40                ;PRIORITY LEVEL 5
1809          ;       000300  PR6=6*40                ;PRIORITY LEVEL 6
1810          ;       000340  PR7=7*40                ;PRIORITY LEVEL 7
1811          ;       000001  BIT0=000001
1812          ;       000002  BIT1=000002
1813          ;       000004  BIT2=000004
1814          ;       000010  BIT3=000010
1815          ;       000020  BIT4=000020
1816          ;       000040  BIT5=000040
1817          ;       000100  BIT6=000100
1818          ;       000200  BIT7=000200
1819          ;       000400  BIT8=000400
1820          ;       001000  BIT9=001000
1821          ;       002000  BIT10=002000
1822          ;       004000  BIT11=004000
1823          ;       010000  BIT12=010000
1824          ;       020000  BIT13=020000
1825          ;       040000  BIT14=040000
1826          ;       100000  BIT15=100000
1827
```

```
                ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 3
                ;BM873-YD,P11   BUTTON #1  -  BOOTSTRAP USING THE PDP-11 SWITCH REGISTER


                ;       173000  ROMORG  =       173000          ;Set ROM origin to 773000
                ;       173000          ,=ROMORG                ;BM873-YD occupies 773000-773777

004000 033727 ;173000  033727  BUTON1: BIT     @#SWR,#BIT0     ;Is rightmost bit on?
004002 177570 ;173002  177570
004004 000001 ;173004  000001
004006 001010 ;173006  001010          BNE     LOWBIT          ;If the bit is on, branch
004010 013707 ;173010  013707          MOV     @#SWR,PC        ;Jump to the address in the switch register
004012 177570 ;173012  177570                                          ;without having touched any of R0 - R6

004014 111704 ;173014  111704  BUTON3: MOVB    (PC),R4         ;R4 = 1 indicates that button #3 was pressed
004016 005001 ;173016  005001          CLR     R1              ;Set unit number to zero
004020 005005 ;173020  005005          CLR     R5              ;Clear "logical switch register"
004022 000424 ;173022  000424          BR      TCBOOT          ;Do a default boot strap from DECtape

004024 173000 ;173024  173000          .WORD   ROMORG,PR7
004026 000340 ;173026  000340

004030 013701 ;173030  013701  LOWBIT: MOV     @#SWR,R1        ;R1 is a copy of the switch register
004032 177570 ;173032  177570
004034 106301 ;173034  106301          ASLB    R1              ;Left-align speed field in right byte
004036 122701 ;173036  122701          CMPB    #16*20,R1       ;Is the speed 16 or 17?
004040 000340 ;173040  000340
004042 101404 ;173042  101404          BLOS    UNITNO          ;If speed is 16 or 17, branch
004044 122701 ;173044  122701          CMPB    #3*20,R1        ;Is the speed 0, 1, or 2?
004046 000060 ;173046  000060
004050 101001 ;173050  101001          BHI     UNITNO          ;If the speed is 0, 1, or 2, branch
004052 005001 ;173052  005001          CLR     R1              ;Speed was 3-15; set unit number = 0
004054 000301 ;173054  000301  UNITNO: SWAB    R1              ;Move unit number to bits 0-2

                                       ; It is possible to manually set the desired bootstrap unit number
                                       ; into the rightmost three bits of R1, set the PDP-11 front panel
                                       ; switch register, and then jump into the ROM code at this point.

004056 042701 ;173056  042701          BIC     #^C7,R1         ;Isolate unit number in R1
004060 177770 ;173060  177770
004062 013705 ;173062  013705          MOV     @#SWR,R5        ;R5 is now the "logical switch register"
004064 177570 ;173064  177570
004066 005004 ;173066  005004          CLR     R4              ;R4 = 0 indicates that button #1 was pressed
004070 105705 ;173070  105705          TSTB    R5              ;Should we boot from DECtape or RH11/RP04?
004072 100507 ;173072  100507          BMI     RPBOOT          ;If bit 7 was one, branch off to the RH11/RP04
                                                                       ;Otherwise, fall through to the DECtape
```



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


                ;       177344  TCWC    =       177344          ;TC11 DECtape word count register
                ;       000001  TCGO    =       1               ;TC11 "GO" bit
                ;       000002  TCRNUM  =       1*2             ;TC11 "read block number" function
                ;       000004  TCREAD  =       2*2             ;TC11 "read data" function
                ;       000014  TCWRIT  =       6*2             ;TC11 "write data" function
                ;       004000  TCREV   =       4000            ;Move DECtape in reverse direction

                                       ; Bootstrap (from DECtape) Parameters
                ;       000400  TCBWDC  =       ^D256           ;Word count for the secondary bootstrap
                ;       000000  TCBEND  =       0               ;Which end of the DECtape (0 = front; 1 = back)

                                       ; Dump (to DECtape) Parameters
                ;       070000  TCDWDC  =       ^D28672         ;Word count for the core dump to DECtape
                ;       000001  TCDEND  =       1               ;Which end of the DECtape (0 = front; 1 = back)

                                       ; General (Bootstrap and Dump) DECtape Parameter
                ;       000024  TCRTRY  =       ^D20            ;Number of retries in case of error

004074 012700 ;173074  012700  TCBOOT: MOV     #<TCBEND*TCREV>!TCREAD!TCGO,R0  ;Set up data-transfer command
004076 000005 ;173076  000005
004100 012702 ;173100  012702          MOV     #-TCBWDC,R2     ;Set word count to 256 (512 bytes)
004102 177400 ;173102  177400
004104 012703 ;173104  012703          MOV     #<<1-TCBEND>*TCREV>!TCRNUM!TCGO,R3       ;Set up position command
004106 004003 ;173106  004003
004110 000301 ;173110  000301          SWAB    R1              ;Bring unit number into the left byte
004112 050103 ;173112  050103          BIS     R1,R3           ;Put unit number into positioning command
004114 050100 ;173114  050100          BIS     R1,R0           ;Put unit number into data-transfer command
004116 012701 ;173116  012701  TCSTRT: MOV     #TCWC,R1        ;R1 now points to TC11 word count register
004120 177344 ;173120  177344
004122 012706 ;173122  012706  TCLOOP: MOV     #TCRTRY,SP      ;Initialize retry count in SP
004124 000024 ;173124  000024
004126 005705 ;173126  005705  TCBGIN: TST     R5              ;Test "indefinite retry" bit
004130 100404 ;173130  100404          BMI     TCRSET          ;Branch if "indefinite retry" is enabled
004132 005306 ;173132  005306          DEC     SP              ;Decrement retry count
004134 100002 ;173134  100002          BPL     TCRSET          ;Branch if retry count not exhausted
004136 000000 ;173136  000000  TCHALT: HALT                    ;Retry count is exhausted for DECtape operation
004140 000770 ;173140  000770          BR      TCLOOP          ;He pressed "CONTINUE", so try again
004142 000005 ;173142  000005  TCRSET: RESET                   ;Stop anything in progress, for next try
004144 010341 ;173144  010341          MOV     R3,-(R1)        ;Initiate DECtape positioning operation
004146 005711 ;173146  005711  TCWAIT: TST     (R1)            ;Test for an "ERROR"
004150 100376 ;173150  100376          BPL     TCWAIT          ;Loop until an "ERROR" is detected
004152 005721 ;173152  005721          TST     (R1)+           ;Make R1 point to the word count register
004154 005761 ;173154  005761          TST     -4(R1)          ;Is the error "ENDZONE"?
004156 177774 ;173156  177774
004160 100362 ;173160  100362          BPL     TCBGIN          ;If not, branch back to try again
004162 010211 ;173162  010211          MOV     R2,(R1)         ;Set up word count for data-transfer
004164 010041 ;173164  010041          MOV     R0,-(R1)        ;Initiate the data-transfer operation
004166 105711 ;173166  105711  TCDONE: TSTB    (R1)            ;Test for "DONE"
004170 100376 ;173170  100376          BPL     TCDONE          ;Loop until the "DONE" bit sets
004172 005721 ;173172  005721          TST     (R1)+           ;Was an "ERROR" detected?
```

```
 1929  004174  100754 ;173174  100754          BMI     TCBGIN          ;If so, branch back and try again
 1930  004176  005741 ;173176  005741          TST     -(R1)           ;Make R1 point to the command register
 1931  004200  105011 ;173200  105011          CLRB    (R1)            ;Stop all DECtape motion
 1932  004202  122700 ;173202  122700          CMPB    #TCREAD!TCGO,R0 ;Was this a "normal read" operation?
 1933  004204  000005 ;173204  000005
 1934  004206  001001 ;173206  001001          BNE     TCSTOP          ;If not, go stop
 1935  004210  000137 ;173210  000137  GOTO@0:  JMP     @(PC)+          ;Jump to PDP-11 location zero
 1936  004212  000000 ;173212  000000  TCSTOP: HALT                    ;Successful completion of a "non-read" operation
 1937  004214  000776 ;173214  000776          BR      TCSTOP          ;So that pressing "CONTINUE" won't go anywhere
```



```
 1938          ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 5
 1939          ;BM873-YD,P11   DECTAPE BOOTSTRAP AND DUMP ROUTINES
 1940
 1941  004216  010037 ;173216  010037  TCDUMP: MOV     R0,@#R0TOR7     ;Save R0 in PDP-11 memory location 40
 1942  004220  000040 ;173220  000040
 1943  004222  000402 ;173222  000402          BR      TCCONT          ;Branch around required interrupt vector
 1944
 1945  004224  173000 ;173224  173000          .WORD   ROMORG,PR7
 1946  004226  000340 ;173226  000340
 1947
 1948  004230  010700 ;173230  010700  TCCONT: MOV     PC,R0           ;Use R0 for a subroutine return address
 1949  004232  000410 ;173232  000410          BR      REGSAV          ;Go to the "Register Saving" subroutine
 1950  004234  012700 ;173234  012700          MOV     #<TCDEND-TCREV>!TCWRIT!TCGO,R0  ;Set up (write) transfer command
 1951  004236  004015 ;173236  004015
 1952  004240  012702 ;173240  012702          MOV     #-TCDWDC,R2     ;Set word-count to 28K words
 1953  004242  110000 ;173242  110000
 1954  004244  012703 ;173244  012703          MOV     #<<1-TCDEND>-TCREV>!TCRNUM!TCGO,R3      ;Set up position command
 1955  004246  000003 ;173246  000003
 1956  004250  005005 ;173250  005005          CLR     R5              ;Clear "indefinite retry" bit
 1957  004252  000721 ;173252  000721          BR      TCSTRT          ;Branch into DECtape routine
 1958
 1959
 1960
 1961
 1962                                          ; The following subroutine is used to save the PDP-11 general registers
 1963                                          ; in PDP-11 memory locations 40-57.
 1964
 1965                                          ; The calling sequence is as follows:           MOV     R0,@#R0TOR7
 1966                                          ;                                               MOV     PC,R0
 1967                                          ;                                               BR      REGSAV
 1968                                          ;                                               return here
 1969
 1970  004254  010137 ;173254  010137  REGSAV: MOV     R1,@#R0TOR7+2   ;Save R1 in memoty location 42
 1971  004256  000042 ;173256  000042
 1972  004260  012701 ;173260  012701          MOV     #R0TOR7+4,R1    ;R1 now points to memory location 44
 1973  004262  000044 ;173262  000044
 1974  004264  010221 ;173264  010221          MOV     R2,(R1)+        ;Save R2 in memory location 44
 1975  004266  010321 ;173266  010321          MOV     R3,(R1)+        ;Save R3 in memory location 46
 1976  004270  010421 ;173270  010421          MOV     R4,(R1)+        ;Save R4 in memory location 50
 1977  004272  010521 ;173272  010521          MOV     R5,(R1)+        ;Save R5 in memory location 52
 1978  004274  010621 ;173274  010621          MOV     SP,(R1)+        ;Save SP in memory location 54
 1979  004276  010021 ;173276  010021          MOV     R0,(R1)+        ;Save PC in memory location 56
 1980  004300  000160 ;173300  000160          JMP     2(R0)           ;Return to the calling routine
 1981  004302  000002 ;173302  000002
 1982
```

```
                ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 6
                ;BM873-YD,P11   RH11/RP04 BOOTSTRAP AND DUMP ROUTINES

                ;       176700  RPCS1   =       176700  ;Address of RH11/RP04 Control & Status register 1
                ;       000002  RPWC    =       2       ;Offset to RH11/RP04 Word Count register
                ;       000006  RPDA    =       6       ;Offset to RH11/RP04 Track & Sector Address register
                ;       000010  RPCS2   =       10      ;Offset to RH11/RP04 Control & Status register 2
                ;       000012  RPDS    =       12      ;Offset to RH11/RP04 Drive Status register
                ;       000032  RPOF    =       32      ;Offset to RH11/RP04 Offset register (CONTAINING FMT22)
                ;       000034  RPDC    =       34      ;Offset to RH11/RP04 Desired Cylinder register

                ;       040000  RPTRE   =       BIT14   ;"Transfer Error" bit in RPCS1
                ;       020000  RPMCPE  =       BIT13   ;"Massbus Control Bus Parity Error" bit in RPCS1
                ;       004000  RPDVA   =       BIT11   ;"Drive Available" bit in RPCS1
                ;       100000  RPATA   =       BIT15   ;"Attention Active" bit in RPDS
                ;       040000  RPERR   =       BIT14   ;"Composite Error" bit in RPDS
                ;       010000  RPFMT   =       BIT12   ;"FMT22" (16-bit words) bit in RPOF

                ;       000021  RPPRST  =       21      ;Read-in Preset
                ;       000061  RPWRIT  =       61      ;Write Data
                ;       000071  RPREAD  =       71      ;Read Data

                ;       000000  RPBFMT  =       0       ;Bootstrap format (0 = 18-bit words; 2 = 16-bit words)
                ;       000400  RPBWDC  =       ^D256   ;Word count for the secondary bootstrap from the RP04
                ;       000626  RPBCYL  =       ^D406   ;Bootstrap cylinder number
                ;       000000  RPBTRK  =       0       ;Bootstrap track number
                ;       000000  RPBSCT  =       0       ;Bootstrap sector number

                ;       000000  RPDFMT  =       0       ;Dump format (0 = 18-bit words; 2 = 16-bit words)
                ;       070000  RPDWDC  =       ^D28672 ;Word count for the core dump to the RP04
                ;       000631  RPDCYL  =       ^D409   ;Dump cylinder number
                                                ; The following two assignments put the dump at the very end of the cylinder
                ;       000015  RPDTRK  =       ^D18-<<RPDWDC-1>/<<^D20+RPDFMT>*^D256>> ;Dump track number
                ;       000010  RPDSCT  =       ^D19+RPDFMT-<<<RPDWDC-1>/^D256>-<<^D18-RPDTRK>*<^D20+RPDFMT>>>


004304  111704 ;173304  111704  BUTON2: MOVB    (PC),R4         ;R4 = 5 indicates that button #2 was pressed
004306  005005 ;173306  005005          CLR     R5              ;Clear "logical switch register"
004310  005001 ;173310  005001          CLR     R1              ;Set unit number to zero

004312  012700 ;173312  012700  RPBOOT: MOV     #<RPREAD*400>!<RPBSCT*10>,R0
004314  034400 ;173314  034400
004316  012702 ;173316  012702          MOV     #-RPBWDC,R2
004320  177400 ;173320  177400
004322  012703 ;173322  012703          MOV     #<RPBFMT*40000>!<RPBTRK*2000>!RPBCYL,R3
004324  000626 ;173324  000626
004326  050100 ;173326  050100          BIS     R1,R0           ;Put the unit number into R0
004330  012701 ;173330  012701  RPSTRT: MOV     #RPCS1,R1       ;Set R1 to the lowest address used by the RH11
004332  176700 ;173332  176700
```



```
                ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 7
                ;BM873-YD,P11   RH11/RP04 BOOTSTRAP AND DUMP ROUTINES

004334  000005 ;173334  000005  RPLOOP: RESET                   ;Reset in case of retry
004336  010006 ;173336  010006          MOV     R0,SP           ;Get the unit number into SP
004340  042706 ;173340  042706          BIC     #^C7,SP         ;Isolate the unit number
004342  177770 ;173342  177770
004344  010661 ;173344  010661          MOV     SP,RPCS2(R1)    ;Tell the RH11 the unit number
004346  000010 ;173346  000010
004350  032711 ;173350  032711          BIT     #RPDVA,(R1)     ;Try to seize this RP04 unit
004352  004000 ;173352  004000
004354  001767 ;173354  001767          BEQ     RPLOOP          ;Branch if we haven't seized it
004356  012721 ;173356  012721          MOV     #RPPRST,(R1)+   ;Do a "Read-In Preset" function
004360  000021 ;173360  000021
004362  010306 ;173362  010306          MOV     R3,SP           ;Get the cylinder number into SP
004364  042706 ;173364  042706          BIC     #^C1777,SP      ;Isolate the cylinder number
004366  176000 ;173366  176000
004370  010661 ;173370  010661          MOV     SP,RPDC-2(R1)   ;Tell the RP04 the cylinder number
004372  000032 ;173372  000032
004374  010306 ;173374  010306          MOV     R3,SP           ;Get the format bit and track number into SP
004376  100003 ;173376  100003          BPL     RPCONT          ;Branch if 20 sector (18-bit words) format
004400  012761 ;173400  012761          MOV     #RPFMT,RPOF-2(R1) ;Establish 22 sector (16-bit words) format
004402  010000 ;173402  010000
004404  000030 ;173404  000030
004406  006206 ;173406  006206  RPCONT: ASR     SP              ;Right align the track
004410  006206 ;173410  006206          ASR     SP              ;               number in the left byte
004412  105006 ;173412  105006          CLRB    SP              ;Clear the right byte
004414  150006 ;173414  150006          BISB    R0,SP           ;Put the sector number into the right byte
004416  106006 ;173416  106006          RORB    SP              ;Right align the
004420  106206 ;173420  106206          ASRB    SP              ;               sector number in
004422  106206 ;173422  106206          ASRB    SP              ;                     the right byte
004424  010661 ;173424  010661          MOV     SP,RPDA-2(R1)   ;Tell the RH11 the track and sector numbers
004426  000004 ;173426  000004
004430  010211 ;173430  010211          MOV     R2,(R1)         ;Tell the RH11 the word count
004432  010006 ;173432  010006          MOV     R0,SP           ;Get the function code into SP
004434  105006 ;173434  105006          CLRB    SP              ;Clear the right byte
004436  000306 ;173436  000306          SWAB    SP              ;Right align the function code
004440  010641 ;173440  010641          MOV     SP,-(R1)        ;Tell the RP04 the function code
004442  105711 ;173442  105711  RPDONE: TSTB    (R1)            ;Test for RH11 "ready"
004444  100376 ;173444  100376          BPL     RPDONE          ;Loop, waiting for RH11 "ready"
004446  032711 ;173446  032711          BIT     #RPTRE!RPMCPE,(R1) ;Test for RH11 error bits
004450  060000 ;173450  060000
004452  001330 ;173452  001330          BNE     RPLOOP          ;If error, branch back for retry
004454  032761 ;173454  032761          BIT     #RPATA!RPERR,RPDS(R1) ;Test for RP04 error bits
004456  140000 ;173456  140000
004460  000012 ;173460  000012
004462  001324 ;173462  001324          BNE     RPLOOP          ;If error, branch back for retry
004464  022706 ;173464  022706          CMP     #RPREAD,SP      ;Was the function a "normal read"?
004466  000071 ;173466  000071
004470  001250 ;173470  001250          BNE     TCSTOP          ;If not, branch to a HALT instruction
004472  022737 ;173472  022737          CMP     #000240,@#0     ;Was "000240" read into location zero?
004474  000240 ;173474  000240
004476  000000 ;173476  000000
004500  001643 ;173500  001643          BEQ     GOTO0           ;If so, branch to location zero
```

```
 2088  004502  000000 ;173502  000000          HALT                    ;"000240" was not read into location zero
 2089  004504  000641 ;173504  000641          BR      GOTO0           ;Branch to location zero
 2090
 2091
 2092  004506  010037 ;173506  010037  RPDUMP: MOV     R0,@#R0TOR7     ;SAVE R0 IN PDP-11 MEMORY LOCATION "R0TOR7"
 2093  004510  000040 ;173510  000040
 2094  004512  010700 ;173512  010700          MOV     PC,R0           ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
 2095  004514  000657 ;173514  000657          BR      REGSAV          ;GO TO THE "REGISTER SAVING" SUBROUTINE
 2096  004516  012700 ;173516  012700          MOV     #<RPWRIT*400>!<RPDSCT*10>,R0
 2097  004520  030500 ;173520  030500
 2098  004522  012702 ;173522  012702          MOV     #-RPDWDC,R2
 2099  004524  110000 ;173524  110000
 2100  004526  012703 ;173526  012703          MOV     #<RPDFMT*40000>!<RPDTRK*2000>!RPDCYL,R3
 2101  004530  032631 ;173530  032631
 2102  004532  000676 ;173532  000676          BR      RPSTRT
 2103
```



```
 2104          ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 8
 2105          ;BM873-YD,P11   DTE20 DEVICE REGISTER AND BIT DEFINITIONS
 2106
 2107
 2108          ;       174400  DTEBAS=174400           ;BASE OF (FIRST) DTE20 DEVICE REGISTER BLOCK
 2109          ;       000040  DTESIZ=000040           ;SPACING BETWEEN CONSECUTIVE DTE20'S
 2110          ;       000004  DTEMAX=4                ;MAXIMUM NUMBER OF DTE20'S ON ONE PDP-11
 2111
 2112
 2113                                                  ;OFFSETS FROM THE BASE OF THE DTE20 DEVICE REGISTER BLOCK
 2114                                                  ;TO SPECIFIC 10/11 INTERFACE RAM LOCATIONS AND REGISTERS.
 2115
 2116                                                ; THE FIRST 12 REGISTERS ARE NOT INITIALIZED BY "INIT" (BECAUSE THEY ARE IN RAMS
 2117          ;       000000  DLYCNT=00               ;DELAY COUNT                    (ADDRESS XXXX00)
 2118          ;       000002  DEXWD3=02               ;DEPOSIT OR EXAMINE WORD 3      (ADDRESS XXXX02)
 2119          ;       000004  DEXWD2=04               ;DEPOSIT OR EXAMINE WORD 2      (ADDRESS XXXX04)
 2120          ;       000006  DEXWD1=06               ;DEPOSIT OR EXAMINE WORD 1      (ADDRESS XXXX06)
 2121          ;       000010  TENAD1=10               ;10 ADDRESS WORD 1 FOR DEX      (ADDRESS XXXX10)
 2122          ;       000012  TENAD2=12               ;10 ADDRESS WORD 2 FOR DEX      (ADDRESS XXXX12)
 2123          ;       000014  TO10BC=14               ;TO10 BYTE COUNT                (ADDRESS XXXX14)
 2124          ;       000016  TO11BC=16               ;TO11 BYTE COUNT                (ADDRESS XXXX16)
 2125          ;       000020  TO10AD=20               ;TO10 PDP11 MEMORY ADDRESS      (ADDRESS XXXX20)
 2126          ;       000022  TO11AD=22               ;TO11 PDP11 MEMORY ADDRESS      (ADDRESS XXXX22)
 2127          ;       000024  TO10DT=24               ;TO10 PDP11 DATA WORD           (ADDRESS XXXX24)
 2128          ;       000026  TO11DT=26               ;TO11 PDP11 DATA WORD           (ADDRESS XXXX26)
 2129
 2130                                                ; THE LAST 4 REGISTERS ARE INITIALIZED BY "INIT" (BECAUSE THEY ARE IN FLIP-FLOPS
 2131          ;       000030  DIAG1=30                ;DIAGNOSTIC WORD 1              (ADDRESS XXXX30)
 2132          ;       000032  DIAG2=32                ;DIAGNOSTIC WORD 2              (ADDRESS XXXX32)
 2133          ;       000034  STATUS=34               ;10/11 INTERFACE STATUS WORD    (ADDRESS XXXX34)
 2134          ;       000036  DIAG3=36                ;DIAGNOSTIC WORD 3              (ADDRESS XXXX36)
 2135
 2136
 2137                                                ;  THE FOLLOWING ARE THE ADDRESSES OF THE DTE20 INTERRUPT VECTORS
 2138
 2139          ;       000774  DTEIV1=774              ;INTERRUPT VECTOR FOR DTE20 #1
 2140          ;       000770  DTEIV2=770              ;INTERRUPT VECTOR FOR DTE20 #2
 2141          ;       000764  DTEIV3=764              ;INTERRUPT VECTOR FOR DTE20 #3
 2142          ;       000760  DTEIV4=760              ;INTERRUPT VECTOR FOR DTE20 #4
 2143
 2144
 2145                                                ;  Bit assignments for various DTE20 registers used by this ROM code
 2146
 2147
 2148                                                        ;BIT ASSIGNMENTS FOR TO10BC
 2149
 2150          ;       100000  INT11=BIT15             ;SET DONE AND INTERRUPT BOTH 10 AND 11
 2151
 2152                                                        ;BIT ASSIGNMENTS FOR TO11BC
 2153
 2154          ;       100000  INT10=BIT15             ;SET DONE AND INTERRUPT BOTH 10 AND 11
 2155          ;       040000  ZSTOP=BIT14             ;STOP ON NULL (ZERO) CHARACTER
 2156          ;       020000  TO11BM=BIT13            ;Byte size for To-11 byte transfers
 2157
```

```
2158          ;BM873-YD       = KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 9
2159          ;BM873-YD,P11   DTE20 DEVICE REGISTER AND BIT DEFINITIONS
2160
2161                                                        ;BIT ASSIGNMENTS FOR DIAG2 (WRITE)
2162          ;       000100  DRESET=BIT6             ;PERFORM DIAGNOSTIC CLEAR
2163
2164                                                        ;BIT ASSIGNMENTS FOR DIAG3 (READ)
2165          ;       000020  DUPE=BIT4               ;DATO UNIBUS parity error
2166          ;       000004  DURE=BIT2               ;DATO UNIBUS receive error
2167          ;       000002  NUPE=BIT1               ;NPR UNIBUS parity error
2168
2169                                                        ;BIT ASSIGNMENTS FOR DIAG3 (WRITE)
2170
2171          ;       000020  CDD=BIT4                ;Clear DUPE and DURE error flags
2172          ;       000002  CNUPE=BIT1              ;Clear NUPE error flag
2173          ;       000001  TO10BM=BIT0             ;Byte size for To-10 byte transfer
2174
2175                                                        ;BIT ASSIGNMENTS FOR STATUS (WRITE)
2176
2177          ;       100000  DON10S=BIT15            ;SET TO10 DONE
2178          ;       040000  DON10C=BIT14            ;CLEAR TO10 DONE
2179          ;       020000  ERR10S=BIT13            ;SET TO10 ERROR
2180          ;       010000  ERR10C=BIT12            ;CLEAR TO10 ERROR
2181          ;       004000  INT11S=BIT11            ;Ring the PDP-11's doorbell (interrupts the -11)
2182          ;       002000  INT11C=BIT10            ;Stop ringing the PDP-11's doorbell
2183          ;       001000  PERCLR=BIT9             ;CLEAR -11 MEMORY PARITY ERROR
2184          ;       000400  INT10S=BIT8             ;Ring the PDP-10's doorbell (interrupts the -10)
2185          ;       000200  DON11S=BIT7             ;SET TO11 DONE
2186          ;       000100  DON11C=BIT6             ;CLEAR TO11 DONE
2187          ;       000040  INTRON=BIT5             ;Enable DTE20 interrupts to the -11
2188          ;       000020  EBUSPC=BIT4             ;Clear "EBUS parity error"
2189          ;       000010  INTROF=BIT3             ;Disable the PDP-11 interupts
2190          ;       000004  EBUSPS=BIT2             ;SET "EBUS parity error"
2191          ;       000002  ERR11S=BIT1             ;SET TO11 ERROR
2192          ;       000001  ERR11C=BIT0             ;CLEAR TO11 ERROR
2193
2194                                                        ;BIT ASSIGNMENTS FOR STATUS (READ)
2195
2196          ;       100000  TO10DN=BIT15            ;TO10 DONE
2197          ;       020000  TO10ER=BIT13            ;TO 10 ERROR (NPR TIMEOUT OR BUS ERROR)
2198          ;       010000  RAMISO=BIT12            ;RAM word read is all zeros
2199          ;       004000  TO11DB=BIT11            ;1 = the PDP11's doorbell is ringing
2200          ;       002000  DXWRD1=BIT10            ;DEPOSIT OR EXAMINE WORD ONE
2201          ;       001000  MPE11=BIT9              ;Parity error within PDP-11 memory
2202          ;       000400  TO10DB=BIT8             ;1 = the PDP-10's doorbell is ringing
2203          ;       000200  TO11DN=BIT7             ;TO11 DONE
2204          ;       000100  EBSEL=BIT6              ;E BUFFER SELECT
2205          ;       000040  NULSTP=BIT5             ;NULL STOP
2206          ;       000020  BPARER=BIT4             ;EBUS parity error
2207          ;       000010  RSTRCT=BIT3             ;This PDP-11 is "RESTRICTED"
2208          ;       000004  DEXDON=BIT2             ;DEPOSIT OR EXAMINE DONE
2209          ;       000002  TO11ER=BIT1             ;TO 11 ERROR (NPR TIMEOUT OR BUS ERROR)
2210          ;       000001  INTSON=BIT0             ;DTE20 interrupts (to the -11) are enabled
```



```
2211          ;BM873-YD       = KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
2212          ;BM873-YD,P11   PROCEDURE BY WHICH THE PDP-10 BOOTSTRAPS AND/OR DUMPS THE PDP-11
2213
2214
2215
2216
2217
2218
2219                          The following is the procedure which the KL10 executes in order
2220                          to dump and/or bootstrap the PDP-11 through the DTE20:
2221
2222
2223                          1, Clear the DTE20 and initiate a BM873 button #4 bootstrap operation
2224                                  -   CONO [SR11B!CL11PT!CLTO11!CLTO10!PILDEN]
2225
2226                          2, Wait to see PDP-11 power fail (AC LOW = true)   -   CONI [DEAD11] = 1
2227
2228                          3, Wait to see PDP-11 power recover (AC LOW = false)   -   CONI [DEAD11] = 0
2229
2230                          4, Wait at least another 150 milliseconds and then clear the reload -11 button
2231                                  -   CONO [CR11B]
2232
2233                          5, Set byte counter to a special code (1365 octal)   -   DATAO [1365]
2234
2235                          6, Ring PDP-11's doorbell   -   CONO[TO11DB]
2236
2237                          7, Wait until "-10 ringing -11's doorbell" is turned off by the -11
2238                                (i.e, until   CONI[TO11DB]   becomes zero),
2239
2240                          8, Enable the DTE20 to use PI 0 interrupts
2241                                (i.e, set   CONO[PILDEN!PI0ENB]),
2242
2243                          9, Set up the To-10 byte pointer (in the EPT) for the first 3.5K,
2244
2245                          10, Set up the byte counter for the first 3.5K, indicating
2246                                "interrupt -10 only"   -   DATAO [1000]
2247
2248                          11, Wait for "To-10 done" or "To-10 error"   -   CONI [TO10DN!TO10ER]
2249
2250                          12, Note whether there was an error (CONI [TO10ER]) and then turn off
2251                                TO10DN and TO10ER   -   CONO [CLTO10], If error, go to step 17,
2252
2253                          13, If end of 28K, go to step 17,
2254
2255                          14, Set up To-10 byte pointer (in the EPT) for the next 3.5K,
2256
2257                          15, Set up the byte counter for the next 3.5K indicating
2258                                "interrupt -10 only" (DATAO [1000]), unless this is the
2259                                last 3.5K (of 28K), in which case indicate "interrupt
2260                                both processors" (DATAO [TO10IB!1000]),
2261
2262                          16, Go to step 11,
2263
2264
```

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


                                17. Set up To-11 byte pointer (in the EPT) for "PDP-11 bootstrap".
                                        Note that the first word of this "PDP-11 bootstrap" must
                                        be the bit pattern 000240 (a PDP-11 NOP instruction).

                                18. Ring the PDP-11's doorbell   -   CONO [TO11DB]

                                19. Wait for either TO11DB to go off (CONI[TO11DB] = 0),
                                        or TO10DB to come on (CONI[TO10DB] = 1).

                                20. If no error was noted in step 12, TO11DB should go off
                                        (TO10DB coming on indicates a massive screwup).
                                        If an error was noted in step 12, TO11DB going off indicates
                                        that the error was "non-fatal" (non-ex-mem or -11 memory
                                        parity) and the -11 is proceeding. TO10DB coming on indicates
                                        that the error was "fatal" and the -11 is HALTED AT LOCATION 173714.
                                        In this latter case the -10 must restart from step 1.

                                21. If TO11DB went off, wait for "To-11 done" or "To-11 error"
                                        -   CONI [TO11DN!TO11ER]

                                22. Note whether there was an error   -   CONI [TO11ER]

                                23. Turn off TO11DN and TO11ER and ring the PDP-11's doorbell
                                        -   CONO [TO11DB!CLTO11]

                                24. Wait for either TO11DB to go off (CONI[TO11DB] = 0),
                                        or TO10DB to come on (CONI[TO10DB] = 1).

                                25. TO11DB going off indicates that the PDP-11 found no errors
                                        and is transferring control to the code which was just
                                        received from the -10. In this case the -10 should start
                                        following the protocol of this code.

                                26. TO10DB coming on indicates that the PDP-11 has found an
                                        error (or that the first word transmitted wasn't the
                                        bit pattern 000240), and the PDP-11 is HALTED AT LOCATION 173766.
                                        In this case the -10 must restart from step 1.
```



```
                        ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
                        ;BM873-YD,P11   BUTTON #4  -  BOOTSTRAP INITIATED BY THE PDP-10 (THROUGH DTE20)


                        ;       000130  DTECOR  =       130             ;Core address into which to store DTE20 regs.
                        ;       000014  DTEREG  =       "D12            ;Number of DTE20 registers to store
                        ;       000400  DTEWDC  =       "D256           ;Word count for secondary bootstrap from the -10
                                                        ; ENTER HERE WHEN THE DTE20 PRESSES BUTTON #4 (BOOTSTRAP INITIATED
                                                        ;       BY THE PDP-10, THROUGH THE DTE20)
004534  010037 ;173534  010037  BUTON4: MOV     R0,@#R0TOR7     ;SAVE R0 IN PDP-11 MEMORY LOCATION "R0TOR7"
004536  000040 ;173536  000040
004540  010700 ;173540  010700          MOV     PC,R0           ;USE R0 FOR A SUBROUTINE RETURN ADDRESS
004542  000644 ;173542  000644          BR      REGSAV          ;GO TO THE "REGISTER SAVING" SUBROUTINE
004544  005005 ;173544  005005          CLR     R5              ;SET R5 = 0
004546  012501 ;173546  012501          MOV     (R5)+,R1        ;SAVE LOCATION 0 IN R1
004550  012503 ;173550  012503          MOV     (R5)+,R3        ;SAVE LOCATION 2 IN R3
004552  012504 ;173552  012504          MOV     (R5)+,R4        ;SAVE LOCATION 4 IN R4
004554  011500 ;173554  011500          MOV     (R5),R0         ;SAVE LOCATION 6 IN R0
004556  012715 ;173556  012715          MOV     #PR7,(R5)       ;SET UP PRIORITY FOR NON-EX-MEM TRAP
004560  000340 ;173560  000340
004562  005745 ;173562  005745          TST     -(R5)           ;SET R5 = 4
004564  012702 ;173564  012702  10$:    MOV     #DTEBAS-DTESIZ,R2
004566  174340 ;173566  174340
004570  010715 ;173570  010715          MOV     PC,(R5)         ;STORE ADDRESS FOR NON-EX-MEM TRAP
004572  010506 ;173572  010506          MOV     R5,SP           ;SET STACK POINTER = 4
004574  062702 ;173574  062702  11$:    ADD     #DTESIZ,R2      ;R2 POINTS TO THE NEXT DTE20
004576  000040 ;173576  000040
004600  105702 ;173600  105702          TSTB    R2
004602  100770 ;173602  100770          BMI     10$             ;START LOOKING FROM THE BEGINNING AGAIN
004604  032762 ;173604  032762          BIT     #TO11DB,STATUS(R2)      ;Is this -10 ringing the -11's doorbell?
004606  004000 ;173606  004000
004610  000034 ;173610  000034
004612  001770 ;173612  001770          BEQ     11$             ;If it is not, go look for another -10
004614  026217 ;173614  026217          CMP     TO10BC(R2),(PC) ;CHECK FOR A CODE (1365) FROM THE PDP-10
004616  000014 ;173616  000014
                                                                ;INDICATING THAT IT WANTS TO BOOTSTRAP THE -11
004620  001365 ;173620  001365          BNE     11$
                                                        ; NOTE THAT AT THIS POINT R2 CONTAINS THE ADDRESS OF THE DEVICE REGISTER
                                                        ;       BLOCK FOR THIS DTE20, THAT R5 = 4, AND THAT SP = 4
004622  005725 ;173622  005725          TST     (R5)+           ;SET R5 = 6
004624  010015 ;173624  010015          MOV     R0,(R5)         ;RESTORE THE CONTENTS OF LOCATION 6
004626  010445 ;173626  010445          MOV     R4,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 4
004630  010345 ;173630  010345          MOV     R3,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 2
004632  010145 ;173632  010145          MOV     R1,-(R5)        ;RESTORE THE CONTENTS OF LOCATION 0
                                                        ; Note: At this time R5 = 0, This fact will be used later.
004634  012700 ;173634  012700          MOV     #DTECOR,R0      ;R0 = core address for storing DTE20 registers
004636  000130 ;173636  000130
004640  010204 ;173640  010204          MOV     R2,R4
004642  012420 ;173642  012420  7$:     MOV     (R4)+,(R0)+     ;Save the next DTE20 register in core
004644  022700 ;173644  022700          CMP     #<DTEREG*2>+DTECOR,R0   ;Have we finished yet?
004646  000160 ;173646  000160
004650  101374 ;173650  101374          BHI     7$              ;Loop until we have finished
```

```
                ;BM873-YD       - KL10 (PDP-11) 256 WORD BOOTSTRAP ROM  VERSION 2(17)   MACY11 27(657)  18-DEC-74  11:59  PAGE 1
                ;BM873-YD.P11   BUTTON #4  -  BOOTSTRAP INITIATED BY THE PDP-10 (THROUGH DTE20)
                ;
004652  010201  ;173652  010201          MOV     R2,R1                   ;R1 = DTE20 DEVICE REGISTER BLOCK
004654  062701  ;173654  062701          ADD     #DIAG2,R1
004656  000032  ;173656  000032
004660  012721  ;173660  012721          MOV     #DRESET,(R1)+           ;DO A "DIAGNOSTIC CLEAR" OF THE DTE20,
004662  000100  ;173662  000100
                                        ; The above  operation is necessary to clear the "byte count loaded" flag
                                        ; and simultaneously to turn off "-10 ringing -11's doorbell",
004664  005012  ;173664  005012          CLR     (R2)                    ;SET DTE20 FOR NO DELAY
004666  005062  ;173666  005062          CLR     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$:     MOV     #TO10DB,(R1)    ;Signal the -10 that there was an error
004764  000400  ;173764  000400
004766  000000  ;173766  000000  12$:    HALT                    ;This error halt is because either "TO11ER"
                                        ; is on, or because the bit pattern read into location zero wasn't "000240",
004770  012762  ;173770  012762  9$:     MOV     #DRESET,DIAG2(R2)       ;Signal the -10 that everything is OK
004772  000100  ;173772  000100
004774  000032  ;173774  000032
004776  END,YD!
004776  000115  ;173776  000115          JMP     (R5)            ;Jump to location zero
                ;       000001          .END
```



```
005000  000177          MAP.YE! ,BLKW   127.
005376  000001          END.YE! ,BLKW   1
005400  000127          MAP.YF! ,BLKW   127
005656  000001          END.YF! ,BLKW   1
005660  000127          MAP.YG! ,BLKW   127
006136  000001          END.YG! ,BLKW   1
006140  000177          MAP.Y.! ,BLKW   127.
006536  000001          END.Y.! ,BLKW   1
006540  000177          MAP.YX! ,BLKW   127.
007136  000001          END.YX! ,BLKW   1
```

```
2427
2428                                    ;*********************************************************
2429                                    ;       INITIALIZATION AND START UP OF PROGRAM.
2430                                    ;*********************************************************
2431
2432  007140  005037  010156   RESTRT: CLR     INITFG          ;INITIALIZE TO ASK WHICH TYPE
2433  007144  005037  010156           CLR     TABLE           ;INITIALIZE TO ASK WHICH TYPE
2434  007150  012706  001200   START:  MOV     #STACK,SP       ;SET THE STACK POINTER
2435  007154  013737  177570  010154           MOV     SWR,SAVSR1      ;SAVE THE SWITH REGISTER
2436  007162  005037  015114           CLR     LPTFLG          ;SET FOR NON- LINE PRINTER OUTPUT
2437  007166  005037  001244           CLR     LSTERR          ;CLEAR ERROR FLG REPORT
2438  007172  005037  000000           CLR     @#0             ;SET FOR UNEXPECTED TRAP TO ADD 0
2439  007176  012737  007150  014616           MOV     #START,PRG,NO   ;GET READY FOR PWR FAIL BEFORE FIRST TEST.
2440  007204  012737  000006  000004           MOV     #6,@#4          ;SET TIME OUT TRAP VECTOR
2441  007212  005037  000006           CLR     @#6             ;SET TIME OUT STATUS TO 0
2442  007216  005037  014042           CLR     FLAG4           ;CLEAR TEST 4 INITIAL FLAG
2443  007222  005737  000042           TST     @#42            ;AM I RUNNING UNDER ACT-11??
2444  007226  001563           BEQ     CONT            ;BR IF *NOT* UNDER ACT-11!!
2445  007230  013746  000004           MOV     @#4,-(SP)       ;SAV TRAP POINTER
2446  007234  012737  010172  000004           MOV     #NOROM,@#4      ;PUT IN A NEW ONE
2447  007242  005737  173000           TST     @#173000        ;TRY TO READ THE ROM
2448  007246  000240           NOP             ;WAIT FOR POSSIBLE TRAP
2449  007250  012637  000004           MOV     (SP)+,@#4       ;IF NO TRAP RESTORE POINTER
2450  007254  023737  001400  173000           CMP     MAP.YA,@#173000 ;DOES 1ST WORD COMPARE?
2451  007262  001034           BNE     648             ;CHECK NEXT MAP
2452  007264  013746  000004           MOV     @#4,-(SP)       ;SAVE LOC 4
2453  007270  012737  007312  000004           MOV     #658,@#4        ;SET FOR TIMEOUT
2454  007276  005737  173400           TST     @#173400        ;READ FROM 173400
2455  007302  000240           NOP             ;IF NO TIMEOUT, NOT YA
2456  007304  012637  000004           MOV     (SP)+,@#4       ;RESTORE LOC 4
2457  007310  000421           BR      648
2458  007312  022626           658:    CMP     (SP)+,(SP)+     ;ADJUST STACK
2459  007314  012637  000004           MOV     (SP)+,@#4       ;RESTORE LOC 4
2460  007320  012737  001400  010156           MOV     #MAP,YA,TABLE   ;1ST MAP ADDR
2461  007326  012737  001776  010160           MOV     #END,YA,ALLEND  ;LAST MAP ADDR
2462  007334  012737  173376  010170           MOV     #173376,LASTA   ;LAST ROM ADDR
2463  007342  012737  000101  014736           MOV     #000101,VERSON  ;SET ROM TYPE
2464  007350  000137  010400           JMP     PRG1            ;START TEST 1
2465  007354                   648:
2466  007354  023737  002000  173000           CMP     MAP.YB,@#173000 ;DOES 1ST WORD COMPARE?
2467  007362  001016           BNE     698             ;CHECK NEXT MAP
2468  007364  012737  002000  010156           MOV     #MAP,YB,TABLE   ;1ST MAP ADDR
2469  007372  012737  002776  010160           MOV     #END,YB,ALLEND  ;LAST MAP ADDR
2470  007400  012737  173776  010170           MOV     #173776,LASTA   ;LAST ROM ADDR
2471  007406  012737  000102  014736           MOV     #000102,VERSON  ;SET ROM TYPE
2472  007414  000137  010400           JMP     PRG1            ;START TEST 1
2473  007420                   698:
2474  007420  023737  003000  173000           CMP     MAP.YC,@#173000 ;DOES 1ST WORD COMPARE?
2475  007426  001036           BNE     748             ;CHECK NEXT MAP
2476  007430  013746  000004           MOV     @#4,-(SP)       ;SAVE LOC 4
2477  007434  012737  007460  000004           MOV     #768,@#4        ;SET FOR TIMEOUT
2478  007442  023737  003400  173400           CMP     MAP.YC+400,@#173400     ;IS IT YC?
2479  007450  001004           BNE     778             ;BR IF NOT YC
2480  007452  012637  000004           MOV     (SP)+,@#4       ;RESTORE LOC 4
```



```
2481  007456  000404           BR      788             ;YES IT IS A YC
2482  007460  022626           768:    CMP     (SP)+,(SP)+     ;ADJUST STACK
2483  007462  012637  000004           778:    MOV     (SP)+,@#4       ;RESTORE LOC 4
2484  007466  000416           BR      748             ;CHECK NEXT MAP
2485  007470                   788:
2486  007470  012737  003000  010156           MOV     #MAP,YC,TABLE   ;1ST MAP ADDR
2487  007476  012737  003776  010160           MOV     #END,YC,ALLEND  ;LAST MAP ADDR
2488  007504  012737  173776  010170           MOV     #173776,LASTA   ;LAST ROM ADDR
2489  007512  012737  000103  014736           MOV     #000103,VERSON  ;SET ROM TYPE
2490  007520  000137  010400           JMP     PRG1            ;START TEST 1
2491  007524                   748:
2492  007524  023737  004000  173000           CMP     MAP.YD,@#173000 ;DOES 1ST WORD COMPARE?
2493  007532  001016           BNE     798             ;CHECK NEXT MAP
2494  007534  012737  004000  010156           MOV     #MAP,YD,TABLE   ;1ST MAP ADDR
2495  007542  012737  004776  010160           MOV     #END,YD,ALLEND  ;LAST MAP ADDR
2496  007550  012737  173776  010170           MOV     #173776,LASTA   ;LAST ROM ADDR
2497  007556  012737  000104  014736           MOV     #000104,VERSON  ;SET ROM TYPE
2498  007564  000137  010400           JMP     PRG1            ;START TEST 1
2499  007570                   798:
2500  007570  104401  010272           TYPE    ,NMATCH         ;NOT BM873YA OR B OR C OR D
2501  007574  000000           HALT
2502  007576  005737  010166   CONT:   TST     INITFG          ;IS THIS THE FIRST TIME START UP?
2503  007602  001114           BNE     3$              ;BR IF NOT FIRST TIME HERE.
2504  007604  005137  010166           COM     INITFG          ;SET THE FLAG
2505  007610  104401  010762   2$:     TYPE    ,BM873X         ;TYPE THE QUESTION.
2506  007614  105777  171364           TSTB    @TKCSR          ;WAIT FOR KEY TO BE HIT
2507  007620  100375           BPL     .-4             ;KEEP WAITING
2508  007622  104406           KEY.TO,R2               ;ECHO AND STRIP BIT 7
2509  007624  022702  000052           CMP     #52,R2          ;WAS * HIT??
2510  007630  001011           BNE     648             ;BR IF NO
2511  007632  012737  006140  010156           MOV     #MAP,Y,,TABLE   ;SET FOR START OF TABLE
2512  007640  012737  006536  010160           MOV     #END,Y,,ALLEND  ;SET END OF TABLE
2513  007646  012737  173376  010170           MOV     #173376,LASTA   ;SET LAST ROM ADDR
2514  007654                   648:
2515  007654  022702  000101           CMP     #101,R2         ;WAS A HIT??
2516  007660  001011           BNE     658             ;BR IF NO
2517  007662  012737  001400  010156           MOV     #MAP,YA,TABLE   ;SET FOR START OF TABLE
2518  007670  012737  001776  010160           MOV     #END,YA,ALLEND  ;SET END OF TABLE
2519  007676  012737  173376  010170           MOV     #173376,LASTA   ;SET LAST ROM ADDR
2520  007704                   658:
2521  007704  022702  000102           CMP     #102,R2         ;WAS B HIT??
2522  007710  001011           BNE     668             ;BR IF NO
2523  007712  012737  002000  010156           MOV     #MAP,YB,TABLE   ;SET FOR START OF TABLE
2524  007720  012737  002776  010160           MOV     #END,YB,ALLEND  ;SET END OF TABLE
2525  007726  012737  173776  010170           MOV     #173776,LASTA   ;SET LAST ROM ADDR
2526  007734                   668:
2527  007734  022702  000103           CMP     #103,R2         ;WAS C HIT??
2528  007740  001011           BNE     678             ;BR IF NO
2529  007742  012737  003000  010156           MOV     #MAP,YC,TABLE   ;SET FOR START OF TABLE
2530  007750  012737  003776  010160           MOV     #END,YC,ALLEND  ;SET END OF TABLE
2531  007756  012737  173776  010170           MOV     #173776,LASTA   ;SET LAST ROM ADDR
2532  007764                   678:
2533  007764  022702  000104           CMP     #104,R2         ;WAS D HIT??
2534  007770  001011           BNE     688             ;BR IF NO
```

```
2535  007772  012737  004000  010156          MOV     #MAP,YD,TABLE   ;SET FOR START OF TABLE
2536  010000  012737  004776  010160          MOV     #END,YD,ALLEND  ;SET END OF TABLE
2537  010006  012737  173776  010170          MOV     #173776,LASTA   ;SET LAST ROM ADDR
2538  010014                                  68$:
2539  010014  010237  014736                  MOV     R2,VERSON       ;STORE VERSION TYPE..
2540  010020  005737  010156                  TST     TABLE           ;HAS A MAP BEEN SELECTED?
2541  010024  001003                          BNE     3$              ;BR IF OK...
2542  010026  104401  011014                  TYPE    ,BM,ERR         ;TYPE ERROR
2543  010032  000666                          BR      2$              ;GO AND GET CORRECT MAP.
2544  010034  104401  014200          3$:     TYPE    ,MSG3           ;TYPE MESSAGE FOR TEST NUMBER
2545  010040  012704  000002                  MOV     #2,R4           ;SET FOR MAX 2 INPUTS FROM TTY
2546  010044  005003                  X,X,1:  CLR     R3              ;CLEAR CHAR STORAGE
2547  010046  105777  171132          1$:     TSTB    @TKCSR          ;WAIT FOR THE TTY KEY TO BE HIT
2548  010052  100375                          BPL     1$              ;KEEP WAITING.....
2549  010054  104406                          KEY.TO,R2               ;GO ECHO IT ;STRIP BIT 7 FROM IT
2550  010056  022702  000015                  CMP     #15,R2          ;WAS IT A CR???
2551  010062  001404                          BEQ     2$              ;BR IF CR
2552  010064  005304                          DEC     R4              ;ONE MORE CHAR TO GO BEFOR ERROR
2553  010066  001426                          BEQ     3$              ;IF R4=0 THEN ERROR
2554  010070  010203                          MOV     R2,R3           ;MOV THE CHAR TO R3
2555  010072  000765                          BR      1$              ;GO WAIT FOR ANOTHER CHAR
2556  010074  022703  000061          2$:     CMP     #61,R3          ;WAS 1 HIT??
2557  010100  001002                          BNE     .+6             ;BR IF NO
2558  010102  000137  010400                  JMP     PRG1            ;GOTO PRG 1
2559  010106  022703  000062                  CMP     #62,R3          ;WAS 2 HIT??
2560  010112  001002                          BNE     .+6             ;BR IF NO
2561  010114  000137  011062                  JMP     PRG2            ;GOTO PRG 2
2562  010120  022703  000063                  CMP     #63,R3          ;WAS 3 HIT??
2563  010124  001002                          BNE     .+6             ;BR IF NO
2564  010126  000137  012116                  JMP     PRG3            ;GOTO PRG3
2565  010132  022703  000064                  CMP     #64,R3          ;WAS 4 HIT??
2566  010136  001002                          BNE     .+6             ;BR IF NO
2567  010140  000137  013354                  JMP     PRG4            ;GOTO PRG 4
2568  010144  104401  014432          3$:     TYPE    ,M,QM           ;NEITHER 1 OR 2 OR 3 OR 4 WAS HIT
2569  010150  000137  007140                  JMP     RESTRT          ;TYPE "??" GO TO THE BEGINING.
2570  010154  000000                  SAVSR1: 0                       ;SAVE THE SWR HERE.
2571  010156  000000                  TABLE:  0
2572  010160  000000                  ALLEND: 0
2573  010162  006540                  EXTMAP: MAP,YX
2574  010164  007136                  EXTEND: END,YX
2575  010166  000000                  INITFG: 0
2576  010170  000000                  LASTA:  0
2577  010172  104401  010202          NOROM:  TYPE    ,NOROMS ;TYPE   CAN'T FIND A RESPONSE
2578  010176  000000                          HALT            ;NO LOADER INSTALLED?
2579  010200  000776                          BR      .-2
2580  010202  005015  051124  050101  NOROMS: .ASCII  <15><12>/TRAP TO 4 ON 1ST READ OF 173000/
      010243     015  044412  020123          .ASCIZ  <15><12>/IS LOADER INSTALLED?/
      010272  005015  040503  023516  NMATCH: .ASCII  <15><12>/CAN'T IDENTIFY LOADER AS YA,YB,YC, OR YD AFTER/
      010352  005015  046503  020120          .ASCIZ  <15><12>/CMP WITH LOC 173000/
                                              .EVEN
```



```
2581                                    ;PROGRAM 1
2582                                    ;THE PURPOSE OF PROGRAM 1 IS TO READ THE ROM AND
2583                                    ;VERIFY THAT THE DATA IS CORRECT. ALL ADDRESSES
2584                                    ;ARE READ, EXCECPT THE TRAP VECTOR, FIVE TIMES.
2585                                    ;
2586                                    ;THE SECOND PART OF THIS TEST VERIFIES THAT TRYING
2587                                    ;TO WRITE THE ROM RESUTS IN A TIME OUT TRAP.
2588                                    ;ALL ADDRESS ARE WRITTEN WITH A =1
2589                                    ;,AND ARE EXPECTED TO TRAP.
2590
2591  010400  012777  014444  170606  PRG1:   MOV     #.KEY.TO,R2,@TKVEC
2592  010406  012777  000100  170570          MOV     #100,@TKCSR     ;SET INTERRUPT ENABLE
2593  010414  012737  010400  014616          MOV     #PRG1,PRG,NO    ;SET FOR PWR FAIL
2594  010422  012737  000500  001232          MOV     #500,ICOUNT     ;DO THIS TEST 500(8) TIMES.
2595  010430  012737  014620  000004  PRG.1:  MOV     #NO.TRAP,@#4    ;SET FOR UNEXPECTED TRAP.
2596  010436  012700  173000                  MOV     #173000,R0      ;SET BEGGINING ADDRESS
2597  010442  012737  010466  001230          MOV     #2$,LOCK        ;IF SW09=1; GOTO 2$ WHEN SCOP1 IS HIT
2598  010450  013704  010156                  MOV     TABLE,R4        ;SET START OF MAP
2599  010454  013737  010170  010760          MOV     LASTA,LAST      ;SET LAST ADDRESS
2600  010462  012703  000005          1$:     MOV     #5.,R3          ;DO EACH ADDRESS 5 TIMES.
2601  010466  022700  173024          2$:     CMP     #173024,R0      ;DON'T DO THE VECTOR ADD.
2602  010472  001001                          BNE     .+4             ;BR IF NOT THE VECTOR ADD.
2603  010474  022024                          CMP     (R0)+,(R4)+     ;UPDATE TO NEXT ADDRESS
2604  010476  022700  173224                  CMP     #173224,R0      ;DON'T DO THE TRAP VECTORS
2605  010502  001001                          BNE     .+4             ;NO THIS ISN'T A TRAP VECTOR.
2606  010504  022024                          CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS..
2607  010506  011005                          MOV     (R0),R5         ;READ THE ADDRESS
2608  010510  011401                          MOV     (R4),R1         ;READ THE SOFTWARE ADDRESS
2609  010512  020105                          CMP     R1,R5           ;DO THEY MATCH?
2610  010514  001401                          BEQ     .+4             ;BR IF GOOD
2611  010516  104001                          HLT     1               ;INCORRECT COMPARISON.
2612  010520  032737  004000  177570          BIT     #BIT11,SWR      ;QUICK PASS,?
2613  010526  001002                          BNE     .+6             ;BR IF YES
2614  010530  005303                          DEC     R3              ;HAS THAT ADD BEEN READ 5 TIMES?
2615  010532  001355                          BNE     2$              ;BR IF NOT 5 TIMES
2616  010534  023700  010760                  CMP     LAST,R0         ;WAS LAST ADDRESS CHECKED?
2617  010540  001403                          BEQ     .+10            ;BR IF YES
2618  010542  104400                          $COP1                   ;LOCK ON THIS ADDRESS?
2619  010544  022024                          CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS.
2620  010546  000745                          BR      1$              ;CONTINUE THE TEST.
2621  010550  032737  000001  010154          BIT     #BIT0,SAVSR1    ;EXTENDED WORD TO BE CHECKED?
2622  010556  001413                          BEQ     3$              ;BR IF NO CHECKING.
2623  010560  022737  173776  010760          CMP     #173776,LAST    ;IS ALL THE TEST DONE?
2624  010566  001407                          BEQ     3$              ;BR IF YES.
2625  010570  012737  173776  010760          MOV     #173776,LAST    ;SET LAST ADDRESS.
2626  010576  013704  010162                  MOV     EXTMAP,R4       ;SET EXTENDED MAP.
2627  010602  005720                          TST     (R0)+           ;POP POINTER
2628  010604  000726                          BR      1$              ;GO DO THE TEST.
```

```
                                        ;TEST THAT WRITTING ROM RESUTS IN A TIME OUT
                                        ;TRAP,
010606  012737  010632  001230  3$:     MOV     #5$,LOCK        ;IF SW09=1 GOTO 5$ WHEN SCOP1 IS HIT
010614  012700  173000                  MOV     #173000,R0      ;SET R0 WITH BASE ADDRESS OF ROM
010620  012737  010662  000004          MOV     #6$,@#4         ;SET FOR TIME OUT TRAP
010626  012703  000005          4$:     MOV     #5.,R3          ;DO EACH ADD 5 TIMES
010632  022700  173024          5$:     CMP     #173024,R0      ;CHECK FOR A TRAP VECTOR
010636  001001                          BNE     .+4             ;BR IF NOT VECTOR
010640  005720                          TST     (R0)+           ;UPDATE THE REGISTER POINTER
010642  022700  173224                  CMP     #173224,R0      ;CHECK FOR THE OTHER VECTOR
010646  001001                          BNE     .+4             ;BR IF NOT THE VECTOR
010650  005720                          TST     (R0)+           ;UPDATE THE POINTER
010652  012710  177777                  MOV     #-1,(R0)        ;WRITE ROM WITH A -1
010656  000240                          NOP                     ;WAIT ONE INSTR, TIME
010660  104002                          HLT     2               ;WRITING ROM DIDN'T TIME OUT,
010662  012706  001200          6$:     MOV     #STACK,SP       ;RESTORE STACK
010666  032737  004000  177570          BIT     #BIT11,SWR      ;QUICK PASS?
010674  001002                          BNE     .+6
010676  005303                          DEC     R3              ;DO EACH ADD 5 TIMES
010700  001354                          BNE     5$              ;NOT DONE WITH THIS ONE YET,
010702  032737  000001  010154          BIT     #BIT0,SAVSR1    ;EXTENDED 128. WORDS TO BE CHECKED?
010710  001404                          BEQ     .+12            ;BR IF NO
010712  022700  173776                  CMP     #173776,R0      ;HAVE ALL 256. WORDS BEEN CHECKED?
010716  001407                          BEQ     7$              ;BR IF ALL DONE
010720  000403                          BR      .+10            ;KEEP GOING
010722  023700  010170                  CMP     LASTA,R0        ;ALL DONE??
010726  001403                          BEQ     7$              ;HAVE ALL 128. WORDS DONE?
010730  104400                          SCOP1                   ;CHECK SW09 FOR FREEZE!!
010732  005720                          TST     (R0)+           ;UPDATE TO NEXT ADDRESS
010734  000734                          BR      4$              ;GO DO IT AGAIN
010736  005337  001232          7$:     DEC     ICOUNT          ;ITERATION COUNT DONE?
010742  001004                          BNE     8$              ;BR IF NOT DONE,
010744  004737  014660                  JSR     PC,EOP          ;TYPE END MESSAGE
010750  000137  010400                  JMP     PRG1            ;GO DO IT AGAIN,
010754  000137  010430          8$:     JMP     PRG.1           ;GO RESTART,
010760  000000          LAST:   0
                        ;
010762  005015  042504  044526  BM873X: .ASCII  <15><12>/DEVICE VERSION/
011002  005015  046502  033470          .ASCII  <15><12>/BM873-Y/
011014  005015  026052  026101  BM.ERR: .ASCII  <15><12>/*,A,B,C,D ONLY,/
011036  020040  053040  051105  VERS:   .ASCII  /   VERSION: BM873-Y/
                                .EVEN
```



```
                                        ;PROGRAM 2
                                        ;BLIND READ FROM ROM,
                                        ;THIS PROGRAM WILL DUMP THE CONTENTS OF THE ROM OUT
                                        ;PERFORMING NO CHECKING AT ALL,
                                        ;PLEASE NOTE: NO CHECKING IS DONE,
                                        ;NOTE:  HITTING "CONTROL "C"" RESTARTS AT TOP OF PROG,
                                        ;NOTE:  IF YOU LEAVE SW07=1; THE DUMP WILL
                                                ;BE MADE ON THE LINE PRINTER
                                                ;IF IT EXISTS,
011062  012737  011062  014616  PRG2:   MOV     #PRG2,PRG.NO    ;SET FOR POWER FAIL
011070  012737  014620  000004          MOV     #NO.TRAP,@#4    ;SET FOR UNEXPECTED TRAP TO 4
011076  013737  010170  010760          MOV     LASTA,LAST
011104  062737  000002  010760          ADD     #2,LAST
011112  105737  177570                  TSTB    SWR             ;IF SW07=1; OUTPUT TABLE ON LPT
011116  100004                          BPL     .+12            ;IF IT EXISTS.  BR IF SW07=0
011120  012737  177777  015114          MOV     #-1,LPTFLG      ;SET THE LPT FLAG
011126  000402                          BR      .+6             ;BR TO TEST
011130  005037  015114                  CLR     LPTFLG          ;CLEAR THE LPT FLAG
011134  012700  173000                  MOV     #173000,R0      ;SET R0 WITH THE STARTING ROM ADD,
011140  012777  011430  170046          MOV     #4$,@TKVEC      ;SET TTY VECTOR FOR CONTROL C
011146  012777  000100  170030          MOV     #100,@TKCSR     ;SET TTY KYBD INTERRUPT EN,
011154  013703  010156                  MOV     TABLE,R3        ;SET POINTER,
011160  104401  011456                  TYPE    ,DH,2           ;TYPE MESSAGE
011164  104401  011556                  TYPE    ,DH,2B          ;TYPE THE HEADER
011170  012737  000007  001256  1$:     MOV     #7,TEMP5        ;SET COUNTER
011176  011001                          MOV     (R0),R1         ;READ THE ROM
011200  010037  001252                  MOV     R0,TEMP3        ;STORE R0
011204  010137  001254                  MOV     R1,TEMP4        ;STORE R1
011210  022737  006140  010156          CMP     #MAP.Y,.TABLE   ;IF BM873,Y* SELECTED; FILL TABLE
011216  001001                          BNE     .+4             ;BR IF NOT BM873,Y*
011220  011023                          MOV     (R0),(R3)+      ;FILL THE TABLE,.
011222  005720                          TST     (R0)+           ;POP THE POINTER
011224  104404                          CONVRT                  ;TYPE AND CONVERT NUMBERS
011226  012076                          DT,2                    ;IN DATA TABLE 2
011230  011001                  7$:     MOV     (R0),R1         ;STORE ROM DATA
011232  010037  001252                  MOV     R0,TEMP3        ;STORE ROM ADDRESS
011236  010137  001254                  MOV     R1,TEMP4        ;PREPARE DATA FOR TYPE OUT
011242  022737  006140  010156          CMP     #MAP.Y,.TABLE   ;IS BM873,Y* SELECTED?
011250  001001                          BNE     .+4             ;BR IF NO,.
011252  011023                          MOV     (R0),(R3)+      ;FILL THE DATA TABLE
011254  005720                          TST     (R0)+           ;POP THE POINTER
011256  104405                          CNVRT                   ;TYPE OUT MAP
011260  012110                          DT,2A                   ;USE DATA TABLE 2A
011262  023700  010760                  CMP     LAST,R0 ;HAS THE HIGHEST LIMIT BEEN HIT?
011266  001404                          BEQ     2$              ;BR IF ALL DONE,
011270  005337  001256                  DEC     TEMP5           ;DECREASE COUNTER
011274  001355                          BNE     7$              ;BR IF NOT 0; KEEP GOING
011276  000734                          BR      1$              ;GO TYPE ADDRESS NOW
011300  032737  000001  010154  2$:     BIT     #BIT0,SAVSR1    ;IS THE EXTENDED 128. WORDS TO BE CHECKED??
011306  001442                          BEQ     3$              ;BR IF NO,
011310  012700  173400                  MOV     #173400,R0      ;RESET POINTER OF ROM
```

```
 2722  011314  013703  010162                  MOV     EXTMAP,R3       ;SET SOFTWARE MAP POINTER
 2723  011320  104401  011671                  TYPE    ,DH.2A          ;TYPE NEW HEADER
 2724  011324  104401  011556                  TYPE    ,DH.2B          ;TYPE ADDRESS AND +XX
 2725  011330  012737  000007  001256  6$:     MOV     #7,TEMP5        ;SET TYPE OUT COUNTER
 2726  011336  011001                          MOV     (R0),R1         ;READ THE ROM
 2727  011340  010037  001252                  MOV     R0,TEMP3        ;STORE R0
 2728  011344  010137  001254                  MOV     R1,TEMP4        ;STORE R1
 2729  011350  012023                          MOV     (R0)+,(R3)+     ;STORE THE DATA IN SOFTWARE MAP
 2730  011352  104404                          CONVRT                  ;TYPE AND CONVERT NUMBERS
 2731  011354  012076                          DT,2                    ;IN DATA TABLE 2
 2732  011356  011001                  8$:     MOV     (R0),R1         ;SAVE THE ROM DATA
 2733  011360  010037  001252                  MOV     R0,TEMP3        ;SAVE THE ROM ADDRESS
 2734  011364  010137  001254                  MOV     R1,TEMP4        ;SET DATA FOR TYPE OUT
 2735  011370  104405                          CNVRT                   ;TYPE OUT THE TABLE.
 2736  011372  012110                          DT,2A                   ;FIND DATA THROUGH DATA TABLE 2A
 2737  011374  012023                          MOV     (R0)+,(R3)+     ;STORE THE DATA IN SOFTWARE TABLE
 2738  011376  022700  174000                  CMP     #174000,R0      ;HAS THE HIGHEST LIMIT BEEN HIT?
 2739  011402  001404                          BEQ     3$              ;BR IF ALL DONE.
 2740  011404  005337  001256                  DEC     TEMP5           ;DEC TABLE COUNTER
 2741  011410  001362                          BNE     8$              ;BR TO JUST TYPE DATA
 2742  011412  000746                          BR      6$              ;BR TO TYPE ADDRESS
 2743  011414  104401  014437          3$:     TYPE    ,MCRLF          ;DUMP LAST LINE FROM LPT.
 2744  011420  005000                          CLR     R0              ;CLEAR DATA LIGHTS
 2745  011422  000000                          HALT                    ;HIT CONTINUE TO PROCEED.
 2746  011424  000137  011062                  JMP     PRG2            ;GOTO PRG 2
 2747                                          ;ENTER BY KYBD INTERUPT.
 2748  011430  104406                  4$:     KEY,TO,R2               ;ECHO KEY
 2749  011432  022702  000003                  CMP     #3,R2           ;WAS CONTROL C HIT
 2750  011436  001006                          BNE     5$              ;BR IF NO
 2751  011440  005037  015114                  CLR     LPTFLG          ;STOP PRINTING ON LPT
 2752  011444  104401  013326                  TYPE    ,MC             ;TYPE "C
 2753  011450  012716  011062                  MOV     #PRG2,(SP)      ;SET FOR RETURN
 2754  011454  000002                  5$:     RTI                     ;LEAVE THIS ROUTINE
 2755
 2756  011456  006414  005012  016412  DH.2:  .ASCII <14><15><12><12><12><35><37><177><177><177>/BLIND READ OF ROM/
       011511     015  006412  077577          .ASCIZ <15><12><15><177><177>/NOTE: NO CHECKING IS PERFORMED./
       011556  005015  040412  042104  DH.2B:  .ASCII  <15><12><12>/ADDRESS  ADD+00  ADD+02  ADD+04/
       011620  020040  042101  025504          .ASCIZ /  ADD+06  ADD+10  ADD+12  ADD+14  ADD+16/
       011671     015  005012  054105  DH.2A:  .ASCII  <15><12><12>/EXTENDED 128. WORD ROM DUMP./
       011730  005015  047503  052116          .ASCII  <15><12>/CONTENTS DUMPED IS PLACED IN THE SOFTWARE/
       012003     015  046412  050101          .ASCII  <15><12>/MAP. VISUAL INSPECTION OF DATA IS/
       012046  005015  047531  051125          .ASCIZ  <15><12>/YOUR RESPONSIBILITY!!/
                                          .EVEN
 2757  012076  000002                  DT,2:   2
 2758  012100     206     003                  .BYTE   206,3
 2759  012102  001252                          TEMP3
 2760  012104     006     002                  .BYTE   6,2
 2761  012106  001254                          TEMP4
 2762
 2763  012110  000001                  DT,2A:  1
 2764  012112     006     002                  .BYTE   6,2
 2765  012114  001254                          TEMP4
```



```
 2766
 2767
 2768                                     ;PROGRAM 3
 2769                                     ;PROGRAM 3 IS THE SAME AS PROGRAM 1 ONLY YOU THE
 2770                                     ;USER HAS THE CHANCE TO ALTER THE MAP WHICH IS
 2771                                     ;COMPARED TO THE DATA IN THE ROM ADDRESSES
 2772                                     ;NOTE THE FOLLOWING COMMANDS:
 2773                                     ;
 2774                                     ;*D     DATA    INSERT DATA; HIT LINE FEED TO ESCAPE.
 2775                                     ;*R     RUN     RUN THE PROGRAM
 2776                                     ;*L     LIST    LIST THE SOFTWARE TABLE ON TTY.
 2777                                     ;*P     PRINT   PRINT LISTING OF SOFTWARE TABLE ON LINE PRINTER.
 2778                                     ;*A     ADDRESS INPUT THE ADDRESS OF THE DATA YOU WANT TO ALTER.
 2779                                     ;LINE FEED      WHEN IN THE DATA INPUT MODE HIT LINE FEED TO
 2780                                     ;               ESCAPE TO COMMAND MODE
 2781                                     ;CR     CARRAGE RETURN=  WHEN IN THE DATA INPUT MODE A CARRAGE RETURN
 2782                                     ;               ZEROS NEXT ADDRESS AND WAITS FOR NEW DATA.
 2783                                     ;
 2784
 2785
 2786  012116  005037  015114          PRG3:   CLR     LPTFLG          ;SET FOR TTY OUTPUT
 2787  012122  012737  012116  014616          MOV     #PRG3,PRG,NO    ;SET FOR POWER FAIL
 2788  012130  012777  012170  167056          MOV     #ST,VEC,@TKVEC  ;SET FOR KEY BOARD INTERUPT
 2789  012136  013701  010156                  MOV     TABLE,R1        ;DEFAULT STARTING ADDRESS TO MAP
 2790  012142  010137  013352                  MOV     R1,ADDRESS      ;SAVE THE SOFTWARE ADDRESS
 2791  012146  104401  014341          XHOLD:  TYPE    ,MASTER         ;TYPE AN "*"
 2792  012152  012777  000100  167024          MOV     #100,@TKCSR     ;SET INTERUPT ENABLE FOR TTY
 2793  012160  000001                          WAIT                    ;HIT A KEY
 2794  012162  000771                          BR      XHOLD           ;GO BACK AN TYPE AN "*"
 2795  012164  000001                  HOLD:   WAIT                    ;HIT A KEY !!
 2796  012166  000776                          BR      HOLD            ;GO BACK AND HIT AGAIN !!
 2797  012170  104406                  ST,VEC: KEY,TO,R2               ;ECHO CHAR AND STRIP BIT 7
 2798  012172  104401  014437                  TYPE    ,MCRLF          ;TYPE A CARRAGE RETURN + LINE FEED
 2799  012176  022702  000114                  CMP     #114,R2         ;WAS AN "L" (LIST) HIT?
 2800  012202  001571                          BEQ     SRV,L
 2801  012204  022702  000120                  CMP     #120,R2         ;WAS A  "P" (PRINT) HIT?
 2802  012210  001563                          BEQ     SRV,P
 2803  012212  022702  000104                  CMP     #104,R2         ;WAS A  "D" (DATA) HIT?
 2804  012216  001417                          BEQ     SRV,D
 2805  012220  022702  000122                  CMP     #122,R2         ;WAS AN "R" (RUN) HIT?
 2806  012224  001002                          BNE     .+6
 2807  012226  000137  012754                  JMP     SRV,R
 2808  012232  022702  000101                  CMP     #101,R2         ;WAS AN "A" (ADDRESS) HIT?
 2809  012236  001501                          BEQ     SRV,A
 2810  012240  022702  000015                  CMP     #15,R2          ;WAS CARRAGE RETURN HIT?
 2811  012244  001001                          BNE     .+4
 2812  012246  000002                          RTI                     ;GO WAIT FOR VALID CHAR.
 2813  012250  104401  014432                  TYPE    ,M,QM           ;TYPE A "?"
 2814  012254  000002                          RTI                     ;NEITHER A "L","P","D","R","A",OR CR WAS HIT.
 2815  012256                           SRV,D:
 2816  012256  012777  012310  166730  2$:     MOV     #4$,@TKVEC      ;SET VECTOR FOR DATA INPUT.
 2817  012264  013701  013352                  MOV     ADDRESS,R1      ;RESET ADDRESS POINTER.
 2818  012270  010137  001254                  MOV     R1,TEMP4        ;SET FOR TYPE OUT ROUTINE
 2819  012274  104404                          CONVRT                  ;TYPE THE SOFTWARE ADDRESS
```

```
2820  012276  013332                          DT,3A                   ;FIND TABLE 3A
2821  012300  005011                          CLR     (R1)            ;ZERO THE ADDRESS AND WAIT FOR DATA
2822  012302  012716  012164                  MOV     #HOLD,(SP)      ;SET RETURN (NO "*")
2823  012306  000002                          RTI                     ;GO BACK WAIT FOR KEY TO BE HIT
2824  012310  104406                  48:     KEY,TO,R2               ;ECHO CHAR AND STRIP BIT 7
2825  012312  022702  000015                  CMP     #15,R2          ;WAS CARRAGE RETURN HIT?
2826  012316  001016                          BNE     5$              ;NO GO TO 5$
2827  012320  104401  014442                  TYPE    ,NLF            ;TYPE A LINE FEED
2828  012324  005721                          TST     (R1)+           ;UPDATE THE SOFTWARE ADDRESS
2829  012326  023701  010164                  CMP     EXTEND,R1       ;IS THE LIMIT EXCEEDED
2830  012332  101440                          BLOS    7$              ;INPUT LIMIT EXCEEDED!! ERROR.
2831  012334  005011                          CLR     (R1)            ;ZERO ADDRESS WAIT FOR DATA
2832  012336  010137  001254                  MOV     R1,TEMP4        ;SAVE THE ADDRESS.
2833  012342  104404                          CONVRT                  ;TYPE THE ADDRESS
2834  012344  013332                          DT,3A                   ;FROM DATA TABLE 3A
2835  012346  010137  013352                  MOV     R1,ADDRESS      ;SAVE THE ADDRESS FOR GOOD
2836  012352  000002                          RTI                     ;LEAVE HERE.
2837  012354  022702  000012          58:     CMP     #12,R2          ;WAS A LINE FEED HIT??
2838  012360  001006                          BNE     6$              ;NO LINE FEED NOT HIT.
2839  012362  012777  012170  166624          MOV     #ST,VEC,@TKVEC  ;SET THE KEYBOADD VECTOR
2840  012370  012716  012146                  MOV     #XHOLD,(SP)     ;WHEN I RETURN TYPE AN "*"
2841  012374  000002                          RTI                     ;LEAVE HERE
2842  012376  032702  000110          68:     BIT     #110,R2         ;WAS AN OCTAL CHAR HIT?
2843  012402  001014                          BNE     7$              ;BR IF INVALID CHAR
2844  012404  042702  000260                  BIC     #260,R2         ;CLEAR ALL BUT LAST 3 BITS
2845  012410  000241                          CLC                     ;CLEAR CPU CARRY BIT
2846  012412  006111                          ROL     (R1)            ;SHIFT DATA
2847  012414  103406                          BCS     8$              ;ERROR ??
2848  012416  006111                          ROL     (R1)            ;SHIFT AGAIN.
2849  012420  103404                          BCS     8$              ;ERROR ??
2850  012422  006111                          ROL     (R1)            ;SHIFT ONCE MORE
2851  012424  103402                          BCS     8$              ;ERROR ??
2852  012426  050211                          BIS     R2,(R1)         ;SET NEW NUMBER IN ADDRESS
2853  012430  000002                          RTI                     ;LEAVE HERE
2854  012432  005011                  88:     CLR     (R1)            ;CLEAR THE ADDRESS.
2855  012434  104401  014432          78:     TYPE    ,M,QM           ;TYPE A "?"
2856  012440  000002                          RTI
2857
2858                                  ;YOU ARE HERE BECAUSE YOU HIT AN "A"
2859                                  ;YOU TOLD ME YOU WERE GOING TO INPUT AN ADDRESS.
2860                                  ;SO INPUT THE ADDRESS AND TERMINATE WITH A CARRAGE RETURN.
2861                                  ;OK??
2862
2863  012442  012777  012460  166544  SRV,A:  MOV     #1$,@TKVEC      ;SET KEY BOADR VECTOR
2864  012450  005003                          CLR     R3              ;R3 WILL HOLD THE SOFTWARE ADDRESS
2865  012452  012716  012164                  MOV     #HOLD,(SP)      ;SET RTURN FOR NO "*"
2866  012456  000002                          RTI                     ;LEAVE HERE
2867  012460  104406                  1$:     KEY,TO,R2               ;ECHO KEY AND CLEAR BIT 7
2868  012462  022702  000015                  CMP     #15,R2          ;CARRAGE RETURN ?
2869  012466  001416                          BEQ     2$              ;BR IF YES
2870  012470  032702  000110                  BIT     #110,R2         ;WAS AN OCTAL NUMBER HIT?
2871  012474  001023                          BNE     3$              ;INVALID ANSWER
2872  012476  000241                          CLC                     ;CLEAR CPU CARRY BIT
2873  012500  006103                          ROL     R3              ;SHIFT THE ADDRESS
```



```
2874  012502  103420                          BCS     3$              ;ERROR ??
2875  012504  006103                          ROL     R3              ;SHIFT AGAIN
2876  012506  103416                          BCS     3$              ;ERROR ??
2877  012510  006103                          ROL     R3              ;ONE LAST TIME
2878  012512  103414                          BCS     3$              ;ERROR ??
2879  012514  042702  000260                  BIC     #260,R2         ;CLEAR UNWANTED BITS
2880  012520  050203                          BIS     R2,R3           ;PLACE DATA IN R3
2881  012522  000002                          RTI                     ;LEAVE THIS ROUTINE
2882  012524  012716  012146          2$:     MOV     #XHOLD,(SP)     ;WHEN RETURNING TYPE AN "*"
2883  012530  012777  012170  166456          MOV     #ST,VEC,@TKVEC  ;SET INITAL VECTOR
2884  012536  010337  013352                  MOV     R3,ADDRESS      ;SAVE GOOD ADDRESS FOR FUTURE USE
2885  012542  000002                          RTI                     ;LEAVE HERE
2886  012544  005003                  3$:     CLR     R3              ;ZERO BAD ADDRESS
2887  012546  104401  014437                  TYPE    ,MCRLF          ;TYPE A CR LF
2888  012552  104401  014432                  TYPE    ,M,QM           ;TYPE "??"
2889  012556  000002                          RTI                     ;LEAVE !!
2890
2891  012560  012737  177777  015114  SRV,P:  MOV     #-1,LPTFLG
2892                                  ;YOU ENTERED HERE BECAUSE YOU HIT "L"
2893                                  ;YOU TOLD ME YOU WANTED A LISTING OF THE SOFTWARE MAP
2894                                  ;SO HERE IT IS.
2895
2896
2897  012566                          SRV,L:
2898  012566  013700  010156                  MOV     TABLE,R0        ;GET SOFTWARE MAP
2899  012572  013737  010160  012752          MOV     ALLEND,DEAD     ;SET DEAD END POINTER
2900  012600  104401  014232                  TYPE    ,MSG4           ;TYPE HEADER
2901  012604  104401  011556                  TYPE    ,DH,2B          ;TYPE ADDRESS ADD+XX
2902  012610  012737  000007  001256  1$:     MOV     #7,TEMP5        ;SET COUNTER FOR ACCROSS PAGE
2903  012616  011001                          MOV     (R0),R1         ;GET DATA
2904  012620  010037  001252                  MOV     R0,TEMP3        ;GET ADDRESS
2905  012624  010137  001254                  MOV     R1,TEMP4        ;SET DATA FOR TYPE OUT
2906  012630  005720                          TST     (R0)+           ;UPDATE ADDRESS POINTER
2907  012632  104404                          CONVRT                  ;TYPE ADDRESS ADD+XX
2908  012634  013340                          DT,3B                   ;USE DATA TABLE 3B
2909  012636  011001                  2$:     MOV     (R0),R1         ;GET DATA
2910  012640  010037  001252                  MOV     R0,TEMP3        ;GET ADDRESS
2911  012644  010137  001254                  MOV     R1,TEMP4        ;SAVE DATA
2912  012650  005720                          TST     (R0)+           ;UPDATE POINTER
2913  012652  104405                          CNVRT                   ;TYPE DATA
2914  012654  012110                          DT,2A                   ;USE DATA TABLE 2A
2915  012656  013703  012752          3$:     MOV     DEAD,R3
2916  012662  005723                          TST     (R3)+           ;UPDATE POINTER
2917  012664  020003                          CMP     R0,R3           ;LIMIT DONE ??
2918  012666  001404                          BEQ     5$              ;BR IF YES
2919  012670  005337  001256          4$:     DEC     TEMP5           ;DEC DATA COUNTER
2920  012674  001360                          BNE     2$              ;BR IF MORE DATA TO GO
2921  012676  000744                          BR      1$              ;TYPE THE ADDRESSS
2922  012700                          5$:
2923  012700  032737  000001  010154          BIT     #BIT0,SAVSR1    ;EXTENDED SOFTWARE DUMP?
2924  012706  001416                          BEQ     6$              ;BR IF NO DUMP
2925  012710  005743                          TST     -(R3)           ;PUSH POINTER
2926  012712  023703  010164                  CMP     EXTEND,R3
2927  012716  001412                          BEQ     6$              ;BR IF ALL DONE
```

```
2928  012720  104401  014276          TYPE    ,MSG5           ;TYPE EXTENDED MAP:
2929  012724  104401  011556          TYPE    ,DH,2B
2930  012730  013700  010162          MOV     EXTMAP,R0       ;SET POINTER
2931  012734  013737  010164  012752  MOV     EXTEND,DEAD     ;SET DEAD END POINTER
2932  012742  000722                  BR      1$              ;DO IT AGAIN SAM.
2933  012744  005037  015114  6$:     CLR     LPTFLG          ;SET FOR TTY OUTPUT
2934  012750  000002                  RTI                     ;GO BACK HOME.
2935  012752  000000          DEAD:   0
2936
2937                          ;NOW YOU ARE HERE BECAUSE YOU WANT TO RUN THE PROGRAM
2938                          ;REMEMBER NOW, YOU SET UP THE MAP.
2939                          ;ARE YOU SURE YOU TYPED IN THE CORRECT DATA.???
2940                          ;HERE WE GO
2941
2942  012754                  $RV,R:
2943  012754  012777  014444  166232  RUN3:   MOV     #,KEY,TO,R2,@TKVEC
2944  012762  012777  000100  166214          MOV     #100,@TKCSR     ;SET KEYBOARD INTERUPT ENABLE
2945  012770  012737  014620  000004          MOV     #NO.TRAP,@#4    ;GET READY FOR UNEXPECTED TRAP
2946  012776  012737  000500  001232          MOV     #500,ICOUNT     ;DO TEST 500(8) TIMES
2947  013004  012700  173000          RUN.3:  MOV     #173000,R0      ;SET BEGGINING ADDRESS
2948  013010  012737  013034  001230          MOV     #2$,LOCK        ;IF SW09=1; GOTO 2$ WHEN I HIT "SCOP1"
2949  013016  013704  010156                  MOV     TABLE,R4        ;SET SOFTWARE RESUTS
2950  013022  013737  010170  010760          MOV     LASTA,LAST      ;SET LAST ADDRESS
2951  013030  012703  000005          1$:     MOV     #5,,R3          ;DO EACH ADDRESS 5 TIMES.
2952  013034  022700  173024          2$:     CMP     #173024,R0      ;DON'T DO THE VECTOR ADD.
2953  013040  001001                          BNE     .+4             ;BR IF NOT THE VECTOR ADD.
2954  013042  022024                          CMP     (R0)+,(R4)+     ;UPDATE TO NEXT ADDRESS
2955  013044  022700  173224                  CMP     #173224,R0      ;IS THIS THE SECOND TRAP VECTOR??
2956  013050  001001                          BNE     .+4             ;BR IF NOT VECTOR
2957  013052  022024                          CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS !!
2958  013054  011005                          MOV     (R0),R5         ;READ THE ADDRESS
2959  013056  011401                          MOV     (R4),R1         ;READ THE SOFTWARE ADDRESS
2960  013060  020105                          CMP     R1,R5           ;DO THEY MATCH?
2961  013062  001401                          BEQ     .+4             ;BRANCH IF OK
2962  013064  104001                          HLT     1               ;INCORRECT COMPARISON.
2963  013066  032737  004000  177570          BIT     #BIT11,SWR      ;QUICK PASS.
2964  013074  001002                          BNE     .+6             ;BR IF YES
2965  013076  005303                          DEC     R3              ;HAS THAT ADD BEEN READ 10 TIMES?
2966  013100  001355                          BNE     2$              ;BR IF NOT 10 TIMES
2967  013102  023700  010760                  CMP     LAST,R0         ;WAS LAST ADDRESS CHECKED?
2968  013106  001403                          BEQ     .+10            ;BR IF YES
2969  013110  104400                          $COP1                   ;LOCK ON THIS ADDRESS?
2970  013112  022024                          CMP     (R0)+,(R4)+     ;UPDATE THE POINTERS.
2971  013114  000745                          BR      1$              ;CONTINUE THE TEST.
2972  013116  032737  000001  010154          BIT     #BIT0,$AVSR1    ;EXTENDED WORD TO BE CHECKED?
2973  013124  001413                          BEQ     3$              ;BR IF NO CHECKING.
2974  013126  022737  173776  010760          CMP     #173776,LAST    ;IS ALL THE TEST DONE?
2975  013134  001407                          BEQ     3$              ;BR IF YES.
2976  013136  012737  173776  010760          MOV     #173776,LAST    ;SET LAST ADDRESS.
2977  013144  013704  010162                  MOV     EXTMAP,R4       ;SET EXTENDED MAP.
2978  013150  005720                          TST     (R0)+           ;POP POINTER
2979  013152  000726                          BR      1$              ;GO DO THE TEST.
```



```
2980                          ;TEST THAT WRITING ROM RESULTS IN A TIME OUT
2981                          ;TRAP.
2982
2983  013154  012700  173000  3$:     MOV     #173000,R0      ;SET BASE ADDRESS
2984  013160  012737  013200  001230          MOV     #5$,LOCK        ;IF SW09=1; GOTO 5$ AT SCOP1
2985  013166  012737  013230  000004          MOV     #6$,@#4         ;TIME OUT TRAP; GOTO 6$
2986  013174  012703  000012  4$:     MOV     #10,,R3         ;DO EACH ADD 10 TIMES
2987  013200  022700  173024  5$:     CMP     #173024,R0      ;IS THIS AT THE TRAP VECTOR
2988  013204  001001                  BNE     .+4             ;BR IF NO
2989  013206  005720                  TST     (R0)+           ;UPDATE POINTER
2990  013210  022700  173224          CMP     #173224,R0      ;IS THIS AT THE SECOND TRAP VECTOR
2991  013214  001001                  BNE     .+4             ;BR IF NO
2992  013216  005720                  TST     (R0)+           ;UPDATE THE POINTER
2993  013220  012710  177777          MOV     #-1,(R0)        ;WRITE ROM WITH A -1
2994  013224  000240                  NOP                     ;WAIT ONE INSTR. TIME
2995  013226  104000                  HLT                     ;WRITING ROM DIDN'T TIME OUT.
2996  013230  012706  001200  6$:     MOV     #STACK,SP       ;RESTORE STACK
2997  013234  032737  004000  177570          BIT     #BIT11,SWR      ;QUICK PASS?
2998  013242  001002                  BNE     .+6             ;BR IF YES
2999  013244  005303                  DEC     R3              ;DO EACH ADD 10 TIMES
3000  013246  001354                  BNE     5$              ;NOT DONE WITH THIS ONE YET.
3001  013250  032737  000001  010154          BIT     #BIT0,$AVSR1    ;IS THE EXTENDED 128. WORDS TO BE TESTED??
3002  013256  001404                  BEQ     .+12            ;BR IF NO
3003  013260  022700  173776          CMP     #173776,R0      ;IS THE EXTENDED LIMIT BEEN TESTED?
3004  013264  001407                  BEQ     7$              ;IF YES; GOTO 7$
3005  013266  000403                  BR      .+10            ;IF NO; KEEP GOING.
3006  013270  023700  010170          CMP     LASTA,R0        ;ALL DONE??
3007  013274  001403                  BEQ     7$              ;IF YES; GOTO 7$
3008  013276  104400                  $COP1                   ;GO CHECK SW09; (FREEZE !!)
3009  013300  005720                  TST     (R0)+           ;UPDATE TO NEXT ADDRESS
3010  013302  000734                  BR      4$              ;GO DO IT AGAIN
3011  013304  005337  001232  7$:     DEC     ICOUNT          ;CHECK ITERATION COUNT
3012  013310  001004                  BNE     8$              ;MORE TO GO
3013  013312  004737  014660          JSR     PC,EOP          ;GO TO END OF PASS ROUTINE
3014  013316  000137  012754          JMP     RUN3            ;GO DO TEST AGAIN
3015  013322  000137  013004  8$:     JMP     RUN.3           ;
3016
3017  013326  041536    000   MC:     .ASCIZ  /"C/
3018          013332          .EVEN
3019  013332  000001  DT.3A:  1
3020  013334     006    002             .BYTE   6,2
3021  013336  001254                  TEMP4
3022  013340  000002  DT.3B:  2
3023  013342     006    003             .BYTE   6,3
3024  013344  001252                  TEMP3
3025  013346     006    002             .BYTE   6,2
3026  013350  001254                  TEMP4
3027  013352  000000  ADDRESS:        0
3028
```

```
3029
3030                                    ;PROGRAM 4
3031                                    ;PROGRAM 4 CHECKS THE TRAP VECTOR ADDRESS.
3032                                    ;THE PROGRAM SIMULATES ACTIVATING THE BUTTON
3033                                    ;FOR EACH CHANNEL AND THEN READS
3034                                    ;THE CONTENTS OF THE ADDRESS.
3035                                    ;ON THE FIRST PASS THE CONTENTS WILL
3036                                    ;BE TYPED OUT FOR YOU THE
3037                                    ;USER TO VERIFY. AFTER THIS THE PROGRAM
3038                                    ;DOES A COMPARE TO THE PREVIOUSLY FOUND DATA
3039                                    ;AND REPORTS AN ERROR IF DIFFERENT THAN
3040                                    ;WHAT WAS FOUND BEFORE.
3041
3042  013354  012777  014444  165632  PRG4:   MOV     #.KEY.TO,R2,@TKVEC
3043  013362  012777  000100  165614          MOV     #100,@TKCSR
3044  013370  012737  013354  014616          MOV     #PRG4,PRG.NO    ;SET FOR POWER FAIL
3045  013376  005037  001244                  CLR     LSTERR          ;PREPARE ERROR CONDITIONS
3046  013402  012706  001200                  MOV     #STACK,SP       ;SET THE STACK POINTER
3047  013406  012737  020000  001232          MOV     #20000,ICOUNT   ;SET ITERATION COUNT TO 20000(8)
3048  013414  005737  014042                  TST     FLAG4           ;HAVE I BEEN HERE BEFOR??
3049  013420  001073                          BNE     TAG,A           ;BR IF NOT FIRST TIME HERE.
3050  013422  005137  014042                  COM     FLAG4           ;SET THE FLAG
3051  013426  012705  000002                  MOV     #2,R5           ;SET R5 FOR SWITCH 1
3052  013432  012704  014032                  MOV     #LOC1,R4        ;SET STORAGE LOCATION
3053  013436  012737  014620  000004          MOV     #NO.TRAP,@#4    ;SET FOR TIME OUT TRAP
3054  013444  012737  000001  001252          MOV     #1,TEMP3        ;SET FOR MESSAGE ON CHANNEL NO.
3055  013452  104401  014044          1$:     TYPE    ,MCHAN          ;TYPE MESSAGE ABOUT CHANNEL
3056  013456  104405                          CNVRT                   ;GOTO TYPE AND CONVERT ROUTINE
3057  013460  014164                          DT,4A                   ;CONVERT DATA TABLE 4A
3058  013462  104401  014060                  TYPE    ,MACTV          ;TYPE REST OF MESSAGE
3059  013466  104401  014073          2$:     TYPE    ,MADD1          ;TYPE ADDRESS MESSAGE
3060  013472  012700  173024                  MOV     #173024,R0
3061  013476  005037  173024                  CLR     @#173024
3062  013502  010537  173024                  MOV     R5,@#173024     ;WRITE ROM WITH SWITCH
3063  013506  000240                          NOP                     ;WAIT ONE INSTR, TIME
3064  013510  012706  001200          3$:     MOV     #STACK,SP       ;SET THE STACK POINTER
3065  013514  012700  173024                  MOV     #173024,R0      ;SET FOR ERROR MESSAGE
3066  013520  012737  014620  000004          MOV     #NO.TRAP,@#4    ;SET FOR NO MORE TRAPS
3067  013526  013737  173024  001254          MOV     @#173024,TEMP4  ;READ THE ADDRESS
3068  013534  104405                          CNVRT                   ;TYPE OUT AND CONVERT
3069  013536  014172                          DT,4B                   ;DATA TABLE 4B
3070  013540  013724  173024                  MOV     @#173024,(R4)+  ;STORE THE INFORMATION FOUND
3071  013544  104401  014127                  TYPE    ,MADD2          ;TYPE THE SECOND ADDRESS MSG
3072  013550  012700  173224                  MOV     #173224,R0      ;SET FOR ERROR CONDITION.
3073  013554  013737  173224  001254          MOV     @#173224,TEMP4  ;STORE ROM DATA
3074  013562  104405                          CNVRT                   ;CONVERT AND TYPE
3075  013564  014172                          DT,4B                   ;DATA TABLE 4B
3076  013566  005237  001252                  INC     TEMP3           ;GET READY FOR NEXT SWITCH SETTING
3077  013572  000241                          CLC                     ;CLEAR THE CARRY BIT
3078  013574  006105                          ROL     R5              ;UPDATE R5
3079  013576  022705  000040                  CMP     #40,R5          ;ALL SIMULATED SWITCHS DONE?
3080  013602  001323                          BNE     1$              ;BR IF NOT ALL DONE
3081  013604  000137  013354                  JMP     PRG4            ;JMP    AND DO TEST AGAIN WITH OUT TYPE OUT
3082
```



```
3083  013610  012703  000002          TAG,A:  MOV     #2,R3           ;SIMULATE SWITCH 1
3084  013614  012704  014032                  MOV     #LOC1,R4        ;GET LOCATION WHERE DATA IS STORED
3085  013620  012737  014620  000004  1$:     MOV     #NO,TRAP,@#4    ;PREPARE FOR TIME OUT TRAP
3086  013626  005037  173024                  CLR     @#173024
3087  013632  010337  173024                  MOV     R3,@#173024     ;WRITE THE ROM
3088  013636  000240                          NOP                     ;WAIT ONE INSTR, TIME
3089  013640  012706  001200          2$:     MOV     #STACK,SP       ;SET THE STACK POINTER.
3090  013644  012737  014620  000004          MOV     #NO,TRAP,@#4    ;SET FOR NO MORE TRAPS.
3091  013652  012700  173024                  MOV     #173024,R0      ;SET FOR ERROR MESSAGE
3092  013656  011401                          MOV     (R4),R1         ;SET FOR COMPARISON
3093  013660  013705  173024                  MOV     @#173024,R5     ;GET THE DATA FROM THE ROM
3094  013664  020105                          CMP     R1,R5           ;IS THE DATA THE SAME??
3095  013666  001401                          BEQ     .+4             ;BR IF GOOD DATA.
3096  013670  104001                          HLT     1               ;ERROR, DATA READ FIRST TIME NOT THE SAME
3097  013672  012700  173224                  MOV     #173224,R0      ;SET FOR ERROR MESSAGE
3098  013676  013705  173224                  MOV     @#173224,R5     ;READ THE ROM
3099  013702  020105                          CMP     R1,R5           ;IS THE DATA THE SAME?
3100  013704  001401                          BEQ     .+4             ;BR IF GOOD DATA
3101  013706  104001                          HLT     1               ;ERROR, DATA NOT THE SAME AS BEFORE.
3102  013710  005724                          TST     (R4)+           ;UPDATE DATA POINTER.
3103  013712  000241                          CLC                     ;CLEAR THE CARRY BIT
3104  013714  006103                          ROL     R3              ;UPDATE THE SIMULATED SWITHCH SETTING
3105  013716  022703  000040                  CMP     #40,R3          ;HAVE ALL SETTING BEEN DONE
3106  013722  001336                          BNE     1$              ;BR IF NOT DONE
3107  013724  005337  001232                  DEC     ICOUNT          ;ITERATION COUNT DONE
3108  013730  001327                          BNE     TAG,A           ;BR IF NOT DONE
3109  013732  012737  177777  173224          MOV     #-1,@#173224    ;WRITE SECOND TRAP VECTOR WITH -1
3110  013740  005037  173024                  CLR     @#173024        ;ZERO THE FIRST VECTOR
3111  013744  012700  173024                  MOV     #173024,R0      ;SET FOR TYPE OUT IF ERROR
3112  013750  013701  014032                  MOV     LOC1,R1         ;SET FOR TYPE OUT ROUTINE
3113  013754  013705  173024                  MOV     @#173024,R5     ;SAME AS ABOVE
3114  013760  020105                          CMP     R1,R5           ;IS DEFAULT LINE SELECTED =TO LINE 1
3115  013762  001401                          BEQ     .+4             ;BR IF DEFAULT EQUALS LINE 1
3116  013764  104001                          HLT     1               ;DATA NOT EQUAL TO LINE I
3117  013766  012737  177777  173024          MOV     #-1,@#173024    ;WRITE A -1 TO FIRST VECTOR
3118  013774  005037  173224                  CLR     @#173224        ;ZERO SECOND VECTOR
3119  014000  012700  173224                  MOV     #173224,R0      ;SET FOR TYPE OUT IF ERROR
3120  014004  013701  014032                  MOV     LOC1,R1         ;GET DATA
3121  014010  013705  173224                  MOV     @#173224,R5     ;READ ROM
3122  014014  020105                          CMP     R1,R5           ;IS LINE 1 DEFAULT LINE
3123  014016  001401                          BEQ     .+4             ;BR IF OK
3124  014020  104001                          HLT     1               ;ERROR LINE 1 NOT DEFAULT LINE
3125  014022  004737  014660                  JSR     PC,EOP          ;TYPE END MESSAGE,
3126  014026  000137  013354                  JMP     PRG4            ;GOTO PROGRAM 4 AGAIN
3127
3128  014032  000000          LOC1:   0
3129  014034  000000          LOC2:   0
3130  014036  000000          LOC3:   0
3131  014040  000000          LOC4:   0
3132  014042  000000          FLAG4:  0
```

```
3133  014044  005015  041412  040510  MCHAN:  .ASCIZ <15><12><12>/CHANNEL /
      014060  041501  044524  040526  MACTV:  .ASCIZ/ACTIVATED./
      014073     015  040412  042104  MADD1:  .ASCIZ  <15><12>/ADDRESS 773024 CONTAINS: /
      014127     015  040412  042104  MADD2:  .ASCIZ  <15><12>/ADDRESS 773224 CONTAINS: /
              014164                  .EVEN
3134  014164  000001                  DT.4A:  1
3135  014166     002     002                  .BYTE   2,2
3136  014170  001252                          TEMP3
3137  014172  000001                  DT.4B:  1
3138  014174     006     002                  .BYTE   6,2
3139  014176  001254                          TEMP4
3140
3141  014200  005015  051120  043517  MSG3:   .ASCIZ <15><12>/PROGRAM NO. (1,2,3,4)  /
      014232  006414  016412  077437  MSG4:   .ASCIZ <14><15><12><35><37><177><177><177>/SOFTWARE MAP IS AS FOLLOWS:/
      014276  005015  020012  054105  MSG5:   .ASCIZ  <15><12><12>/ EXTENDED SOFTWARE MAP FOLLOWS./
      014341     015  025012     000  MASTER: .ASCIZ <15><12>/*/
      014345     007  006407  042412  M.END:  .ASCIZ <7><7><15><12>/END PASS BM873-Y/
      014372                          MFAIL:  .
      014372  005015  053520  020122          .ASCIZ  <15><12>/PWR UP AFTER/
      014410  005015  042522  046101          .ASCIZ  <15><12>/REAL PWR FAIL/
      014430  000044                  M.DOL:  .ASCIZ /$/
      014432  005015  037477     000  M.QM:   .ASCIZ <15><12>/??/
      014437     015  000012          MCRLF:  .ASCIZ <15><12>
      014442  000012                  MLF:    .ASCIZ <12>
                                      .EVEN
3142  014444                          .KEY.TO.R2:
3143  014444  017702  164536                  MOV     @TKDBR,R2       ;READ THE TTY KYBD BUFFER
3144  014450  105777  164534                  TSTB    @TPCSR          ;PRINTER READY??
3145  014454  100375                          BPL     .-4             ;BR IF NOT READY.
3146  014456  010277  164530                  MOV     R2,@TPDBR       ;ECHO INPUTTED CHARACTER
3147  014462  042702  000200                  BIC     #BIT7,R2        ;CLEAR BIT 7 FOR A SEVEN BIT CHAR.
3148  014466  022702  000007                  CMP     #7,R2           ;WAS CNTR "G" HIT?
3149  014472  001005                          BNE     1$              ;NO!
3150  014474  000005                          RESET                   ;CLEAR THE WORLD
3151  014476  005037  177776                  CLR     PS
3152  014502  000137  007150                  JMP     START           ;GO TO TEST
3153  014506  000002                  1$:     RTI                     ;LEAVE HERE
3154
```



```
3155  014510  005037  001244          .PFAIL: CLR     LSTERR
3156  014514  013746  000004                  MOV     @#4,-(SP)
3157  014520  012737  014550  000004          MOV     #1$,@#4
3158  014526  005737  173000                  TST     @#173000        ;IS THIS PF REAL?
3159  014532  000240                          NOP     ;TRAP IS CAUSED BY LOADER
3160  014534  012737  014560  000024          MOV     #PWR.UP,@#24    ;ITS REAL, PREPARE FOR PWR UP
3161  014542  012637  000004                  MOV     (SP)+,@#4
3162  014546  000000                          HALT
3163  014550  022626                  1$:     POP,SP
3164  014552  012637  000004                  MOV     (SP)+,@#4
3165  014556  000000                          HALT    ;HARDWARE ERROR,  BOOT DIDN'T FORCE
3166                                  ;HIGH ADDR LINES AND LOAD BUTTON WAS ACTIVATED
3167  014560  012737  014510  000024  PWR.UP: MOV     #,PFAIL,@#24
3168  014566  012706  001200                  MOV     #STACK,SP
3169  014572  005000                          CLR     R0              ;SET DELAY
3170  014574  062700  000001                  ADD     #1,R0           ;WAIT FOR TTY
3171  014600  001375                          BNE     .-4             ;
3172  014602  104401  014372                  TYPE    ,MFAIL          ;TYPE FAILED,
3173  014606  005037  177776                  CLR     PS              ;SET STATUS TO ZERO
3174  014612  000177  000000                  JMP     @PRG.NO
3175  014616  000000          PRG.NO: 0
3176  014620                  NO.TRAP:
3177  014620  011637  014656                  MOV     (SP),XSTORE
3178  014624  032716  100000                  BIT     #BIT15,(SP)
3179  014630  001410                          BEQ     1$
3180  014632  011600                          MOV     (SP),R0
3181  014634  104004                          HLT     4
3182  014636  012706  001200                  MOV     #STACK,SP
3183  014642  005037  177776                  CLR     PS
3184  014646  000177  177744                  JMP     @PRG.NO
3185  014652  104003                  1$:     HLT     3
3186  014654  000002                          RTI
3187  014656  000000          XSTORE: 0
3188
3189  014660  005037  001244          EOP:    CLR     LSTERR
3190  014664  005037  015114                  CLR     LPTFLG
3191  014670  104401  014345                  TYPE    ,M.END
3192  014674  104401  014736                  TYPE    ,VERSON
3193  014700  013701  000042                  MOV     @#42,R1
3194  014704  001413                          BEQ     X1
3195  014706  022737  010400  014616          CMP     #PRG1,PRG.NO
3196  014714  001002                          BNE     .+6
3197  014716  000137  013354                  JMP     PRG4
3198  014722  004711          LOGIC:  JSR     PC,(R1)
3199  014724  000240                          NOP
3200  014726  000240                          NOP
3201  014730  000240                          NOP
3202  014732  000240                          NOP
3203  014734  000207                  X1:     RTS     PC
3204  014736  000101          VERSON: 101                     ;SEVEN BIT ASCII FOR DEFAULT "A"
```

```
                                                        ;CHECK FOR FREEZE ON CURRENT DATA
014740  032737  001000  177570  .SCOP1: BIT     #SW09,SWR
014746  001402                          BEQ     1$
014750  013716  001230                  MOV     LOCK,(SP)
014754  000002                  1$:     RTI
                                        ;TELETYPE OUTPUT ROUTINE
014756  013737  000004  001000  .TYPE:  MOV     @#4,@#1000
014764  010537  001002                  MOV     R5,@#1002
014770  017605  000000                  MOV     @(SP),R5
014774  062716  000002                  ADD     #2,(SP)
015000  032737  010000  177570  1$:     BIT     #SW12,SWR
015006  001013                          BNE     3$
015010  105715                          TSTB    (R5)
015012  001411                          BEQ     3$
015014  005737  015114                  TST     LPTFLG
015020  001014                          BNE     4$
015022  105777  164162          2$:     TSTB    @TPCSR
015026  100375                          BPL     2$
015030  112577  164156                  MOVB    (R5)+,@TPDBR
015034  000761                          BR      1$
015036  013737  001000  000004  3$:     MOV     @#1000,@#4
015044  013705  001002                  MOV     @#1002,R5
015050  000002                          RTI
015052  012737  015070  000004  4$:     MOV     #5$,@#4
015060  005777  164114                  TST     @LPTCSR
015064  000240                          NOP
015066  000404                          BR      6$
015070  005037  015114          5$:     CLR     LPTFLG
015074  022626                          POP,SP
015076  000740                          BR      1$
015100  105777  164074          6$:     TSTB    @LPTCSR
015104  100375                          BPL     6$
015106  112577  164070                  MOVB    (R5)+,@LPTDBR
015112  000732                          BR      1$
015114  000000                  LPTFLG: 0
                                        ;ERROR HANDLER
015116  005037  015114          .HLT:   CLR     LPTFLG
015122  032737  010000  177570          BIT     #SW12,SWR
015130  001406                          BEQ     XBX
015132  105777  164052                  TSTB    @TPCSR
015136  100003                          BPL     XBX
015140  112777  000207  164044          MOVB    #207,@TPDBR
015146  032737  020000  177570  XBX:    BIT     #SW13,SWR
015154  001075                          BNE     HALTS
015156  021637  001244                  CMP     (SP),LSTERR
015162  001404                          BEQ     1$
```



```
015164  011637  001244                  MOV     (SP),LSTERR
015170  105037  001302                  CLRB    ERRFLG
015174  104402                  1$:     SAV05
015176  011605                          MOV     (SP),R5
015200  162705  000002                  SUB     #2,R5
015204  011504                          MOV     (R5),R4
015206  006304                          ASL     R4
015210  061504                          ADD     (R5),R4
015212  006304                          ASL     R4
015214  042704  177001                  BIC     #177001,R4
015220  062704  016064                  ADD     #,ERRTAB,R4
015224  012437  015320                  MOV     (R4)+,ERRMSG
015230  012437  015332                  MOV     (R4)+,DATAHD
015234  011437  015344                  MOV     (R4),DATABP
015240  105737  001302                  TSTB    ERRFLG
015244  001403                          BEQ     TYPMSG
015246  005737  015344                  TST     DATABP
015252  001030                          BNE     TYPDAT
015254  104401                  TYPMSG: TYPE
015256  016054                          MERRPC
015260  104405                          CNVRT
015262  015416                          ERTAB0
015264  104401                          TYPE
015266  011036                          VERS
015270  105777  163714                  TSTB    @TPCSR
015274  100375                          BPL     .-4
015276  113777  014736  163706          MOVB    VERSON,@TPDBR
015304  104401                          TYPE
015306  014437                          MCRLF
015310  112737  177777  001302          MOVB    #-1,ERRFLG
015316  104401                          TYPE
015320  000000                  ERRMSG: 0
015322  005737  015332                  TST     DATAHD
015326  001402                          BEQ     TYPDAT
015330  104401                          TYPE
015332  000000                  DATAHD: 0
015334  005737  015344          TYPDAT: TST     DATABP
015340  001402                          BEQ     RESREG
015342  104404                          CONVRT
015344  000000                  DATABP: 0
015346  104403                  RESREG: RES05
015350  005737  177570          HALTS:  TST     SWR
015354  100005                          BPL     EXITER
015356  010046                          PUSHR0
015360  016600  000002                  MOV     2(SP),R0
015364  000000                          HALT
015366  012600                          POPR0
015370  005237  001242          EXITER: INC     ERRCNT
015374  032737  000400  177570          BIT     #BIT8,SWR
015402  001404                          BEQ     1$
015404  012706  001200                  MOV     #STACK,SP
015410  000177  177202                  JMP     @PRG,NO
015414  000002                  1$:     RTI
015416  000001                  ERTAB0: 1
```

```
015420     006      002                      .BYTE   6,2
015422  001276                               SAVPC

                                             ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER

015424  104401                      .CONVR:  TYPE
015426  014437                               MCRLF
015430  017637  000000  001250      .CNVRT:  MOV     @(SP),TEMP2
015436  013737  016530  001246               MOV     TEMP,TEMP1
015444  062716  000002                       ADD     #2,(SP)
015450  010046                               MOV     R0,-(SP)
015452  010146                               MOV     R1,-(SP)
015454  010246                               MOV     R2,-(SP)
015456  010346                               MOV     R3,-(SP)
015460  010446                               MOV     R4,-(SP)
015462  010546                               MOV     R5,-(SP)
015464  013701  001250                       MOV     TEMP2,R1
015470  012137  015720                       MOV     (R1)+,WRDCNT
015474  112137  015722              1$:      MOVB    (R1)+,CHRCNT
015500  112137  015723                       MOVB    (R1)+,SPACNT
015504  013137  015724                       MOV     @(R1)+,BINWRD
015510  105737  015722                       TSTB    CHRCNT
015514  100007                               BPL     7$
015516  042737  000200  015722               BIC     #BIT7,CHRCNT
015524  012737  177777  015726               MOV     #-1,XFLAG
015532  000402                               BR      2$
015534  005037  015726              7$:      CLR     XFLAG
015540  013704  015724              2$:      MOV     BINWRD,R4
015544  113705  015722                       MOVB    CHRCNT,R5
015550  012700  016530                       MOV     #TEMP,R0
015554  010403                      3$:      MOV     R4,R3
015556  042703  177770                       BIC     #177770,R3
015562  062703  000260                       ADD     #260,R3
015566  110320                               MOVB    R3,(R0)+
015570  000241                               CLC
015572  006004                               ROR     R4
015574  000241                               CLC
015576  006004                               ROR     R4
015600  000241                               CLC
015602  006004                               ROR     R4
015604  005305                               DEC     R5
015606  001362                               BNE     3$
015610  012703  016572                       MOV     #MDATA,R3
015614  114023                      4$:      MOVB    -(R0),(R3)+
015616  105337  015722                       DECB    CHRCNT
015622  001374                               BNE     4$
015624  005737  015726                       TST     XFLAG
015630  001403                               BEQ     .+10
015632  052737  000006  016572               BIS     #6,MDATA
015640  105737  015723                       TSTB    SPACNT
015644  001405                               BEQ     6$
015646  112723  000240              5$:      MOVB    #240,(R3)+
015652  105337  015723                       DECB    SPACNT
015656  001373                               BNE     5$
```



```
015660  105013                      6$:      CLRB    (R3)
015662  104401                               TYPE
015664  016572                               MDATA
015666  005337  015720                       DEC     WRDCNT
015672  001300                               BNE     1$
015674  013737  001246  016530               MOV     TEMP1,TEMP
015702  012605                               MOV     (SP)+,R5
015704  012604                               MOV     (SP)+,R4
015706  012603                               MOV     (SP)+,R3
015710  012602                               MOV     (SP)+,R2
015712  012601                               MOV     (SP)+,R1
015714  012600                               MOV     (SP)+,R0
015716  000002                               RTI
015720  000000                      WRDCNT:  0
015722  000000                      CHRCNT:  0
        015723                      SPACNT=CHRCNT+1
015724  000000                      BINWRD:  0
015726  000000                      XFLAG:   0

                                             ;SAVE PC OF TEST THAT FAILED AND R0-R5

015730  016637  000004  001276      .SAV05:  MOV     4(SP),SAVPC

                                             ;SAVE R0-R5

015736  010537  001272              SV05:    MOV     R5,SAVR5
015742  010437  001270                       MOV     R4,SAVR4
015746  010337  001266                       MOV     R3,SAVR3
015752  010237  001264                       MOV     R2,SAVR2
015756  010137  001262                       MOV     R1,SAVR1
015762  010037  001260                       MOV     R0,SAVR0
015766  000002                               RTI

                                             ;RESTORE R0-R5

015770  013700  001260              .RES05:  MOV     SAVR0,R0
015774  013701  001262                       MOV     SAVR1,R1
016000  013702  001264                       MOV     SAVR2,R2
016004  013703  001266                       MOV     SAVR3,R3
016010  013704  001270                       MOV     SAVR4,R4
016014  013705  001272                       MOV     SAVR5,R5
016020  000002                               RTI
                                             ;TRAP DISPATCH SERVICE
                                             ;ARGUMENT OF TRAP IS EXTRACTED
                                             ;AND USED AS OFFSET TO OBTAIN POINTER
                                             ;TO SELECTED SUBROUTINE

016022  011646                      .TRPSR:  MOV     (SP),-(SP)          ;GET PC OF RETURN
016024  162716  000002                       SUB     #2,(SP)             ;=PC OF TRAP
016030  017616  000000                       MOV     @(SP),(SP)          ;GET TRP
016034  006316                      TRPOK:   ASL     (SP)                ;MULTIPLY TRAP ARG BY 2
016036  042716  177001                       BIC     #177001,(SP)        ;CLEAR UNWANTED BITS
016042  062716  001304                       ADD     #.TRPTAB,(SP)       ;POINTER TO SUBROUTINE ADDRESS
016046  017616  000000                       MOV     @(SP),(SP)          ;SUBROUTINE ADDRESS
```

```
  3421  016052  000136                          JMP     @(SP)+                  ;GO TO SUBROUTINE
  3422  016054  005015  041520  020072  MERRPC: .ASCIZ  <15><12>/PC: /
  3423  016062     000
  3424          016064                  .EVEN
  3425  016064                          .ERRTAB:
  3426  016064  000000                          0
  3427  016066  000000                          0
  3428  016070  000000                          0
  3429  016072  016122                          EM0
  3430  016074  016330                          DH0
  3431  016076  016466                          DT0
  3432  016100  016164                          EM1
  3433  016102  016415                          DH1
  3434  016104  016510                          DT1
  3435  016106  016223                          EM2
  3436  016110  016430                          DH2
  3437  016112  016516                          DT2
  3438  016114  016270                          EM3
  3439  016116  016415                          DH1
  3440  016120  016510                          DT1
  3441  016122  005015  047522  020115  EM0:    .ASCIZ  <15><12>/ROM READ DATA COMPARISON ERROR./
        016164  005015  051127  052111  EM1:    .ASCIZ  <15><12>/WRITTING ROM FAILED TO TRAP./
        016223     015  052412  042516  EM2:    .ASCIZ  <15><12>/UNEXPECTED TRAP WHILE READING ROM./
        016270  005015  040506  040524  EM3:    .ASCIZ  <15><12>/FATAL TRAP.  ROM PC ON STACK./
        016330  005015  051440  043117  DH0:    .ASCII  <15><12>/ SOFT      ROM/
        016350  005015  042101  051104          .ASCII  <15><12>/ADDRESS  ADDRESS  EXPECTED  FOUND /
        016415     015  040412  042104  DH1:    .ASCII  <15><12>/ADDRESS /
        016430  005015  041520  047440  DH2:    .ASCII  <15><12>/PC OF TRAP    ROM ADDRESS /
                016466                  .EVEN
  3442  016466  000004                  DT0:    4
  3443  016470     006     003                  .BYTE   6,3
  3444  016472  001270                  $AVR4
  3445  016474     206     003                  .BYTE   206,3
  3446  016476  001260                  $AVR0
  3447  016500     006     004                  .BYTE   6,4
  3448  016502  001262                  $AVR1
  3449  016504     006     002                  .BYTE   6,2
  3450  016506  001272                  $AVR5
  3451
  3452  016510  000001                  DT1:    1
  3453  016512     206     002                  .BYTE   206,2
  3454  016514  001260                  $AVR0
  3455
  3456  016516  000002                  DT2:    2
  3457  016520     006     010                  .BYTE   6,8.
  3458  016522  014656                  XSTORE
  3459  016524     206     002                  .BYTE   206,2
  3460  016526  001260                  $AVR0
  3461
  3462  016530  000000                  TEMP:   0
  3463          016572                  .=.+40
  3464  016572  000000                  MDATA:  0
  3465          016634                  .=.+40
  3466  016634                          CORMAX:
```



```
  3467          000001                  .END
```

```
ADDRES  013352          2790*  2817   2835*  2884*  3027#
ALLEND  010160          2461*  2469*  2487*  2495*  2512*  2518*  2524*  2530*  2536*  2572#  2899
BINWRD  015724          3333*  3340   3383#
BIT0  = 000001           398#  2621   2658   2719   2923   2972   3001
BIT1  = 000002           397#
BIT10 = 002000           388#
BIT11 = 004000           387#  2612   2646   2963   2997
BIT12 = 010000           386#
BIT13 = 020000           385#
BIT14 = 040000           384#
BIT15 = 100000           383#  3178
BIT2  = 000004           396#
BIT3  = 000010           395#
BIT4  = 000020           394#
BIT5  = 000040           393#
BIT6  = 000100           392#
BIT7  = 000200           391#  3147   3336
BIT8  = 000400           390#  3307
BIT9  = 001000           389#
BM,ERR  011014          2542   2667#
BM873X  010762          2505   2667#
CHRCNT  015722          3331*  3334   3336*  3341   3357*  3381#  3382
CNVRT = 104405           739#  2712   2735   2913   3056   3068   3074   3279
CONT    007576          2444   2502#
CONVRT= 104404           737#  2703   2730   2819   2833   2987   3297
CORMAX  016634          3466#  3467
DATABP  015344          3272*  3275   3295   3298#
DATAHD  015332          3271*  3291   3294#
DEAD    012752          2899*  2915   2931*  2938#
DH,2    011456          2693   2756#
DH,2A   011671          2723   2756#
DH,2B   011556          2694   2724   2756#  2901   2929
DH0     016330          3430   3441#
DH1     016415          3433   3439   3441#
DH2     016430          3436   3441#
DT,2    012076          2704   2731   2757#
DT,2A   012110          2713   2736   2763#  2914
DT,3A   013332          2820   2834   3019#
DT,3B   013340          2908   3022#
DT,4A   014164          3057   3134#
DT,4B   014172          3069   3075   3137#
DT0     016466          3431   3442#
DT1     016510          3434   3440   3452#
DT2     016516          3437   3456#
EM0     016122          3429   3441#
EM1     016164          3432   3441#
EM2     016223          3435   3441#
EM3     016270          3438   3441#
END,YA  001776           932#  2461   2518
END,YB  002776          1304#  2469   2524
END,YC  003776          1772#  2487   2530
END,YD  004776          2414#  2495   2536
END,YE  005376          2418#
END,YF  005656          2420#
```



```
END,YG  006136          2422#
END,YX  007136          2426#  2574
END,Y,  006536          2424#  2512
EOP     014660          2662   3013   3125   3189#
ERRCNT  001242           695#  3306*
ERRFLG  001302           710#  3268*  3273   3288*
ERRMSG  015320          3270*  3290#
ERTAB0  015416          3280   3312#
EXITER  015370          3301   3306#
EXTEND  010164          2574#  2829   2926   2931
EXTMAP  010162          2573#  2626   2722   2930   2977
FLAG4   014042          2442*  3048   3058*  3132#
HALTS   015350          3256   3300#
HOLD    012164          2795#  2796   2822   2865
ICOUNT  001232           691#  2594*  2660*  2946*  3011*  3047*  3187*
INIFLG  001300           716#
INITFG  010166          2432*  2502   2504*  2575#
KEY,TO= 104406           741#  2508   2549   2748   2797   2824   2867
LAST    010760          2599*  2616   2623   2625*  2665#  2682*  2683*  2714   2950*  2967   2974   2976*
LASTA   010170          2462*  2470*  2488*  2496*  2513*  2519*  2525*  2531*  2537*  2576#  2599   2655   2682
                        2950   3006
LIGHTS= 177570           368#
LOCK    001230           690#  2597*  2632*  2948*  2984*  3211
LOC1    014032          3052   3084   3112   3120   3128#
LOC2    014034          3129#
LOC3    014036          3130#
LOC4    014040          3131#
LOGIC   014722          3198#
LOKFLG  001303           719#
LPCNT   001234           692#
LPTCSR  001200           675#  3235   3241
LPTDBR  001202           676#  3243*
LPTFLG  015114          2436*  2686*  2688*  2751*  2786*  2891*  2933*  3190*  3224   3238*  3245#  3249*
LSTERR  001244           696#  2437*  3045*  3155*  3189*  3257   3259*
MACTV   014060          3056   3133#
MADD1   014073          3059   3133#
MADD2   014127          3071   3133#
MAP,YA  001400           748#  2450   2460   2517
MAP,YB  002000           937#  2466   2468   2523
MAP,YC  003000          1306#  2474   2478   2486   2529
MAP,YD  004000          1777#  2492   2494   2535
MAP,YE  005000          2417#
MAP,YF  005400          2419#
MAP,YG  005660          2421#
MAP,YX  006540          2425#  2573
MAP,Y,  006140          2423#  2511   2699   2708
MASTER  014341          2791   3141#
MC      013326          2752   3017#
MCHAN   014044          3055   3133#
MCRLF   014437          2743   2798   2887   3141#  3287   3319
MDATA   016572          3355   3361*  3369   3464#
MERRPC  016054          3278   3422#
MFAIL   014372          3141#  3172
MLF     014442          2827   3141#
```

```
                        3067*   3073*   3139
TEMP5   001256           704#   2695*   2716*   2725*   2740*   2902*   2919*
TKCSR   001204           677#   2506    2547    2592*   2691*   2792*   2944*   3043*
TKDBR   001206           678#   3143
TKLVL   001216           682#
TKVEC   001214           681#   2891*   2690*   2788*   2816*   2839*   2863*   2883*   2943*   3042*
TPCSR   001210           679#   3144    3226    3252    3283
TPDBR   001212           680#   3146*   3228*   3254*   3285*
TPLVL   001222           684#
TPVEC   001220           683#
TRPOK   016034          3417#
TSTNO   001236           693#
TYPDAT  015334          3276    3292    3295#
TYPE  = 104401           731#   2500    2505    2542    2544    2560    2577    2693    2694    2723    2724    2743    2752
                        2791    2798    2813    2827    2855    2887    2888    2900    2901    2928    2929    3055    3058
                        3059    3071    3172    3191    3192    3277    3281    3286    3289    3293    3318    3368
TYPMSG  015254          3274    3277#
VERS    011036          2667#   3282
VERSON  014736          2463*   2471*   2489*   2497*   2539*   3192    3284#   3285
WRDCNT  015720          3330*   3370*   3380#
XBX     015146          3251    3253    3255#
XFLAG   015726          3337*   3339*   3359    3384#
XHOLD   012146          2791#   2794    2840    2882
XSTORE  014656          3177*   3187#   3458
X,X.    010044          2546#
X1      014734          3194    3203#
$Y    = 000007             1#    720#    729     731#    733#    735#    737#    739#    741#    743#
.     = 016634           402#    403     405     407     409     411     413     415     417     419     421     423     425
                         427     429     431     433     435     437     439     441     443     445     447     449     451
                         453     455     457     459     461     463     465     467     469     471     473     475     477
                         479     481     483     485     487     489     491     493     495     497     499     501     503
                         505     507     509     511     513     515     517     519     521     523     525     527     529
                         531     533     535     537     539     541     543     545     547     549     551     553     555
                         557     559     561     563     565     567     569     571     573     575     577     579     581
                         583     585     587     589     591     593     595     597     599     601     603     605     607
                         609     611     613     615     617     619     621     623     625     627     629     631     633
                         635     637     639     641     643     645     647     649     651     653     655     657     660#
                         667#    669#    671#    747#   2417#   2418#   2419#   2420#   2421#   2422#   2423#   2424#   2425#
                        2426#   2507    2557    2560    2563    2566    2579    2602    2605    2610    2613    2617    2637
                        2640    2647    2651    2654    2685    2687    2700    2709    2806    2811    2953    2956    2961
                        2964    2968    2988    2991    2998    3002    3005    3018#   3095    3100    3115    3123    3133#
                        3145    3171    3196    3284    3360    3424#   3441#   3463#   3465#
,CNVRT  015430           740    3320#
,CONVR  015424           738    3318#
,ERRTA  016064          3269    3425#
,HLT    015116           663    3249#
,KEY,T  014444           742    2591    2943    3042    3142#
,PFAIL  014510           661    3155#   3167
,RESET  015770           736    3402#
,SAVES  015730           734    3388#
,SCOP1  014740           730    3209#
,TRPSR  016022           665    3414#
,TRPTA  001304           728#   3419
,TYPE   014756           732    3216#
```



```
HELLO      1#
HLT      380#   2611    2644    2962    2995    3096    3101    3116    3124    3181    3185
$BUFFE     1#
$CATCH     1#    401
$CONVR     1#   3315
$HEADE     1#    319
$HLT       1#   3246
$HOW     746#   2509    2515    2521    2527    2533
$INSTR     1#
$PARAM     1#
$REG       1#   3385
$ROMTY   746#   2450    2466    2474    2492
$SCOPE     1#
$SCOP1     1#   3206
$SYMBO     1#    333
$TRAPS     1#    720
$TRPDE     1#    729     731     733     735     737     739     741
$TRPSR     1#   3409
$TYPE      1#   3213
$VARIA     1#    659
```

```
ADD      2683  3170  3219  3266  3269  3322  3345  3419
ASL      3265  3267  3417
BCS      2847  2849  2851  2874  2876  2878
BEQ      2444  2551  2553  2610  2617  2622  2624  2651  2653  2656  2715  2720  2739  2800  2802
         2804  2809  2869  2918  2924  2927  2961  2968  2973  2975  3002  3004  3007  3095  3100
         3115  3123  3179  3194  3210  3223  3251  3258  3274  3292  3296  3308  3360  3363
BIC      2844  2879  3147  3268  3336  3344  3418
BIS      2852  2880  3361
BIT      2612  2621  2646  2650  2719  2842  2870  2923  2963  2972  2997  3001  3178  3209  3220
         3250  3255  3307
BLOS     2830
BNE      2451  2467  2475  2479  2493  2503  2510  2516  2522  2528  2534  2541  2557  2560  2563
         2566  2602  2605  2613  2615  2637  2640  2647  2649  2661  2700  2709  2717  2741  2750
         2806  2811  2826  2838  2843  2871  2920  2953  2956  2964  2966  2988  2991  2998  3000
         3012  3049  3080  3106  3108  3149  3171  3196  3221  3225  3256  3276  3354  3358  3366
         3371
BPL      2507  2548  2685  3145  3227  3242  3253  3284  3301  3335
BR       2457  2481  2484  2543  2555  2579  2620  2628  2654  2659  2687  2718  2742  2794  2796
         2921  2932  2971  2979  3005  3010  3229  3237  3240  3244  3338
CLC      2845  2872  3077  3103  3347  3349  3351
CLR      2432  2433  2436  2437  2438  2441  2442  2546  2688  2744  2751  2786  2821  2831  2854
         2864  2886  2933  3045  3061  3086  3110  3118  3151  3155  3169  3173  3183  3189  3190
         3238  3249  3339
CLRB     3260  3367
CMP      2450  2458  2466  2474  2478  2482  2492  2509  2515  2521  2527  2533  2550  2556  2559
         2562  2565  2601  2603  2604  2606  2609  2616  2619  2623  2636  2639  2652  2655  2699
         2708  2714  2738  2749  2799  2801  2803  2805  2808  2810  2825  2829  2837  2868  2917
         2926  2952  2954  2955  2957  2960  2967  2970  2974  2987  2990  3003  3006  3079  3094
         3099  3105  3114  3122  3148  3195  3257
COM      2504  3050
DEC      2552  2614  2648  2660  2716  2740  2919  2965  2999  3011  3107  3353  3370
DECB     3357  3365
EMT       380
HALT      404   406   408   410   412   414   416   418   420   422   424   426   428   430   432
          434   436   438   440   442   444   446   448   450   452   454   456   458   460   462
          464   466   468   470   472   474   476   478   480   482   484   486   488   490   492
          494   496   498   500   502   504   506   508   510   512   514   516   518   520   522
          524   526   528   530   532   534   536   538   540   542   544   546   548   550   552
          554   556   558   560   562   564   566   568   570   572   574   576   578   580   582
          584   586   588   590   592   594   596   598   600   602   604   606   608   610   612
          614   616   618   620   622   624   626   628   630   632   634   636   638   640   642
          644   646   648   650   652   654   656   658  2501  2578  2745  3162  3165  3304
INC      3076  3306
JMP       668   670  2464  2472  2490  2498  2558  2561  2564  2567  2569  2663  2664  2746  2807
         3014  3015  3081  3126  3152  3174  3184  3197  3310  3421
JSR      2662  3013  3125  3198
MOV      2434  2435  2439  2440  2445  2446  2449  2452  2453  2456  2459  2460  2461  2462  2463
         2468  2469  2470  2471  2476  2477  2480  2483  2486  2487  2488  2489  2494  2495  2496
         2497  2511  2512  2513  2517  2518  2519  2523  2524  2525  2529  2530  2531  2535  2536
         2537  2539  2545  2554  2591  2592  2593  2594  2595  2596  2597  2598  2599  2600  2607
         2608  2625  2626  2632  2633  2634  2635  2642  2645  2680  2681  2682  2686  2689  2690
         2691  2692  2695  2696  2697  2698  2701  2705  2706  2707  2710  2721  2722  2725  2726
         2727  2728  2729  2732  2733  2734  2737  2753  2787  2788  2789  2790  2792  2816  2817
         2818  2822  2832  2835  2839  2840  2863  2865  2882  2883  2884  2891  2898  2899  2902
```



```
         2903  2904  2905  2909  2910  2911  2915  2930  2931  2943  2944  2945  2946  2947  2948
         2949  2950  2951  2958  2959  2976  2977  2983  2984  2985  2986  2993  2996  3042  3043
         3044  3046  3047  3051  3052  3053  3054  3060  3062  3064  3065  3066  3067  3070  3072
         3073  3083  3084  3085  3087  3089  3090  3091  3092  3093  3097  3098  3109  3111  3112
         3113  3117  3119  3120  3121  3143  3146  3156  3157  3160  3161  3164  3167  3168  3177
         3180  3182  3193  3211  3216  3217  3218  3230  3231  3233  3259  3262  3264  3270  3271
         3272  3303  3309  3320  3321  3323  3324  3325  3326  3327  3328  3329  3330  3333  3337
         3340  3342  3343  3355  3372  3373  3374  3375  3376  3377  3378  3388  3392  3393  3394
         3395  3396  3397  3402  3403  3404  3405  3406  3407  3414  3416  3420
MOVB     3228  3243  3254  3285  3288  3331  3332  3341  3346  3356  3364
NOP      2448  2455  2643  2994  3063  3088  3159  3199  3200  3201  3202  3236
RESET    3150
ROL      2846  2848  2850  2873  2875  2877  3078  3104
ROR      3348  3350  3352
RTI      2754  2812  2814  2823  2836  2841  2853  2856  2866  2881  2885  2889  2934  3153  3186
         3212  3232  3311  3379  3398  3400
RTS      3203
SUB      3263  3415
TRAP      729   731   733   735   737   739   741
TST      2443  2447  2454  2502  2540  2627  2638  2641  2658  2702  2711  2828  2906  2912  2916
         2925  2978  2989  2992  3009  3048  3102  3158  3224  3235  3275  3291  3295  3300  3359
TSTB     2506  2547  2684  3144  3222  3226  3241  3252  3273  3283  3334  3362
WAIT     2793  2795
.ASCII   2580  2667  2756  3141  3441
.ASCIZ   2580  2667  2756  3017  3133  3141  3422  3441
.BLKW    2417  2418  2419  2420  2421  2422  2423  2424  2425  2426
.BYTE     716   717   718   719  1209  1210  1211  1212  1213  1214  2758  2760  2764  3020  3023
         3025  3135  3138  3313  3443  3445  3447  3449  3453  3457  3459
.ENABL      1   333
.END     3467
.ENDC    2460  2468  2476  2486  2494  3467
.EQUIV    380
.EVEN    2580  2667  2756  3018  3133  3141  3424  3441
.IF      2452  2460  2468  2476  2494  3467
.IRP      729   731   733   735   737   739   741
.LIST       1   319   333   731   733   735   737   739   741   743   747  2417  2580  2667  2756
         3133  3141  3441  3467
.MACRO      1   746
.NLIST      1   319   333   731   733   735   737   739   741   743   747  2417  2580  2667  2756
         3133  3141  3441  3467
.PAGE      51    95   153   183   231   319   352   401   713   934  1306  1777  1828  1875  1938
         1983  2034  2104  2158  2211  2265  2312  2364  2427  2581  2629  2980  3133
.REM        1    52    96   154   184   232   289  2218  2268
.REPT     403
.SBTTL      1
.TITLE    333
```

ERRORS DETECTED: 0

*DZBMD,DZBMD/SOL/CRF_BMDOCD,BMUTLD,BM873D
RUN-TIME: 17 30 2 SECONDS

CORE USED: 11K