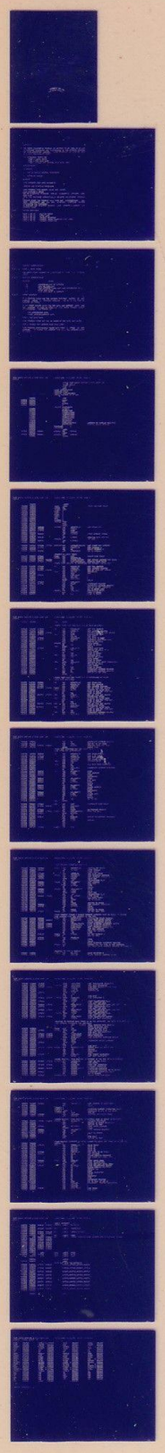
BM792YB

DECTAPE + DISK BOOTSTRAP
MD-11-DZBMB-A
LOADER

EP-DZBMB-A-DL

FICHE 1 OF 1

MAY 1978
digital
MADE IN USA



IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DZHMB-A-D REPLACES: MAINDEC-11-D1KA-D
PRODUCT NAME:	PM792YB DECTAPE & DISK BOOTSTRAP LOADER
DATE CREATED:	JUNE 30, 1972
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	J. ADAMS

A1

1. ABSTRACT

THE DZSMB DIAGNOSTIC PROGRAM IS WRITTEN TO BE USED AS AN AID TO HARDWARE DEBUGGING AND MAINTENANCE OF THE BM792YB DECYAPE AND DISK BOOTSTRAP LOADER). THESE PROGRAMS MAY ALSO BE USED AS A DATA RELIABILITY TEST.

THE AVAILABLE TESTS ARE
PRG0 = LOGIC TESTS
PRG1 = ROM DATA DUMP
PRG2 = SINGLE ROM ADDRESS READ DATA LOOP

2. REQUIREMENTS

2.1 EQUIPMENT

A. PDP 11 FAMILY CENTRAL PROCESSOR

B. BM792-YB MODULE

2.2 STORAGE

THIS PROGRAM USES CORE 0-4100(8)

3. LOADING AND STARTING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ABS LOADER.
LOAD ADDRESS = 00200
SET SR = DESIRED STANDARD PDP-11 DIAGNOSTIC OPTIONS (SEE SECT 6.0)
NOTE: ALL SWITCHES AUTOMATICALLY SELECTS AND STARTS PROGRAM 0.
DEPRESS START THE PROGRAM WILL TYPE OUT INSTRUCTIONS. ALL USER RESPONSES ARE VIA THE KEYBOARD (CARRIAGE RETURN TERMINATES THE RESPONSE)
TO RESTART THE SELECTED PROGRAM LOAD ADDRESS = 002212 AND DEPRESS START

4.0 SWITCH SETTINGS

SW15 1 OR UP HALT ON ERROR
SW14 1 OR UP SCOPE LOOP
SW13 1 OR UP INHIBIT PRINTOUT
SW12 1 OR UP INHIBIT TRACE TRAPPING (NOT USED)
SW11 1 OR UP INHIBIT ITERATION

5. PROGRAM DESCRIPTIONS

5.1 PRGB • LOGIC TESTS

THE LOGIC TESTS CONSIST OF 4 ROUTINES TO TEST THE BM792YB LOGIC

5.1.1 ROUTINE DESCRIPTIONS

ROUTINE	TESTS
T1	ADDRESSABILITY OF BM792YB
T2	DATA RELIABILITY
T3	THAT BM792YB TIMES OUT WHEN REFERENCED BY A DATIP BUS CYCLE
T4	THAT DATA READ IS CORRECT

5.1.2 ERROR PRINTOUT

IF A ROUTINE FAILS AND THE INHIBIT PRINTOUT SWITCH IS NOT ENABLED (SR13) A PRINTOUT RESULTS, THE PC AT THE TIME OF FAILURE IS TYPED.

IF AN ERROR OCCURS IN T4 THE ROM DATA AND CORRECT DATA AND THE ADDRESS OF EACH IS TYPED OUT (THE ERROR TYPEOUT CANNOT BE DISABLED), THE FORMAT IS

ROM ADDRESS/ROM DATA
IMAGE ADDRESS•CORRECT DATA

5.2 PRG1 • ROM DATA DUMP

THIS PROGRAM TYPES OUT THE 32 WORDS OF ROM DATA AND HALTS.

5.3 PRG2 • SINGLE ROM ADDRESS READ DATA LOOP

THIS PROGRAM CONTINUOUSLY READS DATA FROM A TYPED IN ROM ADDRESS. TO CHANGE THE ADDRESS TYPE IN A NEW ADDRESS.(MUST BE EVEN)

```
                        .TITLE TEST DZBMB DECTAPE & DISK BOOT LDR
                        .NLIST SEQ,MC
                        .LIST ME
                        .ABS
                    ;LOAD ADDRESS=P200
                    ;DEPRESS START
                    ;RESTART ADDRESS=0210
                    ;STACK POINTER IS AT 500

                        .=30
P000030 P002362         ERROR
P000032 P000340         S4P
P000034 002272          SCOPEC
P000036 P000200         P
                    ;EQUATE STATEMENTS
                        MLTSEMT
                        SCOPE=TRAP
                        TKCSR=177560
                        TKDBR=177562
                        TPCSR=177564
                        TPDBR=177566
                        PSW=177776
                        SR=177570
                        DISPLAY=177570          ;ADDRESS OF DISPLAY REGISTER
                        STKPTR=500              ;INITIAL STACK SETTING
                    TKINT=.+60
                    TKINT=.+62
                        .=200
P000200 P000167 P000024 START1: JMP     PRMTRS
                        .=210
P000210 P000167 000694  START1: JMP     RESTART
                        .=1000
```

```
        001000                          .=1000
001000  000000                  ICNT:   .WORD   0               ;CONTAINS PASS COUNT
001002  000040                  WORDS:  32.
001004  004000                  IMAGE:  4000
001006  000000                  DUMP:   0
001010  000000                  LAST:   0
001012  000000                  CHAR:   0
001014  000000                  TERM:   0
001016  000000                  SRT:    0
001020  000000                  ROMADD: 0
001022  001110                  PRGTAB: PRG0
001024  001640                          PRG1
001026  001766                          PRG2
001030  012706  000500          PRMTRS: MOV     #STKPTR,%6      ;SET STACK PTR
001034  005067  000020                  CLR     PRGNUM
001040  005737  177570                  TST     @#SR
001044  001411                          BEQ     RESTART
001046  004567  001012                  JSR     5,TYPEM         ;TYPE MESSAGE (PRG#)
001052  002542                          M6
001054  004567  001122                  JSR     5,RECD          ;RECEIVE DATA AND PUT
001060  000000                  PRGNUM: 0                       ;IT HERE
001062  004567  000776                  JSR     5,TYPEM
001066  002566                          M8

001070  012767  173100  177722  RESTART:MOV     #173100,ROMADD  ;GET FIRST ROM ADDRESS
001076  016700  177756                  MOV     PRGNUM,%0       ;GET PROGRAM #
001102  006300                          ASL     %0              ;SHIFT PROGRAM #
001104  000170  001022                  JMP     @PRGTAB(0)      ;GO TO PROGRAM

                                ;PROGRAM 0 LOGIC TESTS
001110  005067  177664          PRG0:   CLR     ICNT            ;CLEAR PASS COUNT
001114  012706  000500          PRG0R:  MOV     #STKPTR,%6
001120  012767  001114  001220          MOV     #PRG0R,RETURN   ;SET RETURN ADDRESS FOR SCOPE
001126  016737  177646  177570          MOV     ICNT,@#DISPLAY  ;DISPLAY PASS COUNT
                                ;TEST1 TEST ABILITY TO REFERENCE ROM WITHOUT TIMING OUT

001134  016700  177660          T1:     MOV     ROMADD,%0       ;GET ROM ADDRESS
001140  016701  177636                  MOV     WORDS,%1        ;GET ADDRESS COUNTER
001144  012767  001206  176632          MOV     #ERROR1,4       ;SET UP TIME OUT VECTOR
001152  011003                  T1A:    MOV     (0),%3          ;REFERENCE
001154  005720                          TST     (0)+            ;ROM
001156  064067  177624                  ADD     -(0),DUMP       ;
001162  021010                          CMP     (0),(0)         ;
001164  132020                          BITB    (0)+,(0)+       ;
001166  000005                          RESET                   ;DELAY
001170  164067  177612                  SUB     -(0),DUMP
001174  062700  000002                  ADD     #2,%0           ;INCREMENT POINTER
001200  005301                          DEC     %1              ;DECREMENT ADDRESS COUNTER
001202  001363                          BNE     T1A             ;BRANCH IF NOT FINISHED
001204  000423                          BR      T1B             ;GO TO SCOPE LOOP
001206  022626                  ERROR1: CMP     (6)+,(6)+       ;REPOSITION STACK
001210  104000                          HLT                     ;HERE IF ERROR
001212  000757                          BR      T1A             ;LOOP ON ERROR
```

```
001214  104400                  T1$:    SCOPE

                                ;TEST2  TEST THAT ROM DATA CAN BE READ RELIABLY.

001216  016700  177976            T2:     MOV     ROMADD,R0       ;GET ROM ADDRESS
001222  016701  177954                    MOV     WORDS,R1        ;GET ADDRESS COUNTER
001226  012767  000006  176558            MOV     #6,4            ;INITIALIZE TIME OUT VECTOR
001234  005067  177946            T2A:    CLR     DUMP            ;INITIALIZE DUMP
001240  011003                            MOV     (R0),R3         ;GET DATA
001242  062067  177940                    ADD     (R0)+,DUMP      ;ADD DATA TO DUMP
001246  166703  177934                    SUB     DUMP,R3         ;SUBTRACT DATA FROM DATA
001252  001402                            BEQ     T2B             ;BRANCH IF EQUAL
001254  104000                ERROR2: HLT                     ;DATA ERROR
001256  000766                            BR      T2A             ;LOOP ON ERROR
001260  000005                T2B:    RESET                   ;DELAY
001262  042067  177928                    BIC     -(R0),DUMP      ;CLEAR DUMP BITS
001266  001402                            BEQ     T2C             ;BRANCH IF EQUAL TO 0
001270  104000                            HLT                     ;DATA ERROR
001272  000772                            BR      T2B             ;LOOP ON ERROR
001274  021010                T2C:    CMP     (R0),(R0)       ;COMPARE DATA
001276  001402                            BEQ     T2D             ;BRANCH IF EQUAL
001300  104000                            HLT                     ;DATA ERROR
001302  000774                            BR      T2C             ;LOOP ON ERROR
001304  122048                T2D:    CMPB    (R0)+,-(R0)     ;COMPARE DATA (BYTE OPERATION)
001306  001402                            BEQ     T2E             ;BRANCH IF EQUAL
001310  104000                            HLT                     ;DATA ERROR
001312  000774                            BR      T2D             ;LOOP ON ERROR
001314  005720                T2E:    TST     (R0)+           ;INCREMENT ADDRESS POINTER
001316  005301                            DEC     R1              ;DECREMENT ADDRESS COUNTER
001320  001345                            BNE     T2A             ;RETURN IF NOT DONE
001322  104400                            SCOPE

                                ;TEST3 TEST THAT ROM TIMES OUT IF REFERENCED BY OTHER
                                ;THAN DATI BUS CYCLE

001324  012706  000500            T3:     MOV     #STKPTR,R6      ;SET STACK PTR
001330  016700  177464                    MOV     ROMADD,R0       ;SET ROM ADDRESS
001334  016701  177442                    MOV     WORDS,R1        ;GET ADDRESS COUNTER
001340  012767  001364  176436    T3AA1:  MOV     #T3B,4          ;SET UP TIME OUT VECTOR
001346  010010                T3A:    MOV     R0,(R0)         ;ATTEMPT TO ALTER DATA
001350  104000                            HLT                     ;HERE IF DID NOT TIME OUT
001352  000775                            BR      T3A             ;LOOP ON ERROR
001354  012767  001372  176422    T3B:    MOV     #T3D,4          ;SET UP TIME OUT VECTOR
001362  022626                            CMP     (SP)+,(SP)+     ;REPOSITION STACK
001364  005210                T3C:    INC     (R0)            ;ATTEMPT TO ALTER DATA
001366  104000                            HLT                     ;HERE IF DID NOT TIME OUT
001370  000775                            BR      T3C             ;LOOP ON ERROR
001372  012767  001412  176404    T3D:    MOV     #T3F,4          ;SET UP TIME OUT VECTOR
001400  022626                            CMP     (SP)+,(SP)+     ;REPOSITION STACK
001402  005077  177412            T3E:    CLR     @ROMADD         ;ATTEMPT TO ALTER DATA
001406  104000                            HLT                     ;HERE IF DID NOT TIME OUT
001410  000774                            BR      T3E             ;LOOP ON ERROR
001412  005720                T3F:    TST     (R0)+           ;INCREMENT ADDRESS POINTER
001414  022626                            CMP     (SP)+,(SP)+     ;REPOSITION STACK
```

```
001416  005301                  DEC     %1              ;DECREMENT ADDRESS COUNTER
001420  001347                  BNE     T3AA            ;RETURN IF NOT DONE
001422  012737  000004  000004  MOV     #4,@#4          ;RSTSTOPE TIME OUT TRAP
001430  104420                  SCOPE                   ;SCOPE LOOP
                                ;THIS TEST COMPARES ROM AND IMAGE DATA
                                ;AND TYPES OUT DIFFERENCES

001432  012706  000500  T4:     MOV     #STKPTR,%6      ;SET STACK PTR
001436  016701  177340          MOV     WORDS,%1        ;GET # OF WORDS
001442  016720  177352          MOV     ROMADD,%0       ;GET ROM ADDRESS
001446  016703  177332          MOV     IMAGE,%3        ;GET IMAGE ADDRESS
001452  021013          T4B:    CMP     (0),(3)         ;COMPARE DATA
001454  001004                  BNE     T4D
001456  005301          T4C:    DEC     %1              ;ALL DATA BEEN COMPARED
001460  001437                  BEQ     T4E
001462  022023                  CMP     (0)+,(3)+       ;INCREMENT ADDRESS POINTERS
001464  000772                  BR      T4B
001466  010067  000732  T4D:    MOV     %0,DZBTYP       ;TYPE
001472  004767  000730          JSR     7,02A           ;ROM ADDRESS
001476  004567  000362          JSR     5,TYPEM         ;TYPE
001522  002002                  .+10                    ;SEPARATOR
001524  011067  000714          MOV     (0),DZBTYP      ;TYPE
001530  004767  000712          JSR     7,02A           ;ROM DATA
001534  004567  000344          JSR     5,TYPEM         ;TYPE
001540  002006                  .+8                     ;CR/LF
001542  010367  000676          MOV     %3,DZBTYP       ;TYPE
001546  004767  000674          JSR     7,02A           ;IMAGE ADDRESS
001552  004567  000326          JSR     5,TYPEM         ;TYPE
001556  002010                  .+12                    ;SEPARATOR
001560  011367  000660          MOV     (3),DZBTYP      ;TYPE
001564  004767  000656          JSR     7,02A           ;IMAGE DATA
001570  004567  000320          JSR     5,TYPEM         ;TYPE
001574  002004                  .+6                     ;CR/LF
001576  000737                  BR      T4C             ;GO TO T4C
001600  104400          T4E:    SCOPE

001602  005267  177212  END:    INC     ICNT            ;INCREMENT PASS COUNT
001606  026727  177206  000100  CMP     ICNT,#100
001614  001402                  BEQ     DONE
001616  000167  177312          JMP     PRGBR           ;GO RESTART PROGRAM
001622  012737  000207  177566  DONE:   MOV     #207,@#TPDBR    ;RING THE BELL
001630  105737  177564          TSTB    @#TPCSR
001634  100375                  BPL     .-4
001636  013700  000042          MOV     @#42,%0         ;RETURN TO DECTAPE MONITOR
001642  001404                  BEQ     DONE1
001644  004710                  JSR     7,(0)           ;RETURN!
001646  000240                  NOP
001650  000240                  NOP
001652  000240                  NOP
001654  000167  177250  DONE1: JMP     PRGB
```

```
                                        ;THIS PROGRAM TYPES OUT ROM DATA

001040  012706  000500          PRG1:   MOV     #STKPTR,SP      ;INITIALIZE STACK
001044  004567  000214                  JSR     R,TYPEM         ;TYPE MESSAGE
001050  002552                          .ASCIZ  /ROM DATA/
001052  016701  177124                  MOV     WORDS,R1        ;GET # OF WORDS
001056  016700  177136          PRG1A:  MOV     ROMADD,R0       ;GET STARTING ADDRESS
001062  012702  000012                  MOV     #12,R2          ;GET ADDRESS INDICATOR
001066  105767  175672                  TSTB    TPCSR           ;WAIT FOR
001072  100375                          BPL     .-4             ;TELEPRINTER FLAG
001074  010067  000524          PRG1B:  MOV     R0,D2BTYP       ;GET ADDRESS
001100  004767  000522                  JSR     R,O2A           ;AND TYPE IT
001104  004567  000194                  JSR     R,TYPEM         ;TYPE
001110  002566                          .BYTE   15,12           ;CR/LF
001112  012067  000526          PRG1C:  MOV     (R)+,D2BTYP     ;TYPE
001116  004767  000504                  JSR     R,O2A           ;DATA
001122  105767  175636                  TSTB    TPCSR           ;WAIT FOR
001126  100375                          BPL     .-4             ;TELEPRINTER FLAG
001130  012767  000040  175638          MOV     #' ,TPDBR       ;TYPE SPACE
001136  005301                          DEC     R1              ;ALL DATA TYPED
001140  001410                          BEQ     PRG1D           ;GO TO FINISH
001142  005302                          DEC     R2
001144  001362                          BNE     PRG1C           ;RETURN TO PRG1B
001146  012702  000012                  MOV     #12,R2          ;GET ADDRESS INDICATOR
001152  004567  000100                  JSR     R,TYPEM         ;TYPE
001156  002566                          .BYTE   15,12           ;CR/LF
001160  000745                          BR      PRG1B           ;RETURN TO PRG1B
001162  000167  177842          PRG1D:  JMP     PRMTRS          ;GO GET NEXT TEST

                                        ;THIS PROGRAM CYCLES A SINGLE ADDRESS (ADDRESS MUST BE EVEN) TO CHANGE
                                        ;THE ADDRESS TYPE NEW ADDRESS ON THE TTY.
001166  012706  000500          PRG2:   MOV     #STKPTR,SP      ;INITIALIZE STACK POINTER
001172  012737  002060  000004          MOV     #PRG2C,@#4      ;LOAD TRAP ERROR VECTOR
002000  005067  175772                  CLR     PSW             ;CLEAR PROCESSOR STATUS
002004  012767  002036  176046          MOV     #PRG2A,TKINTA   ;LOAD KEYBOARD INTERRUPT VECTOR
002012  012767  000340  176042          MOV     #340,TKINTP     ;LOAD KEYBOARD PRIORITY
002020  012767  000100  175532          MOV     #100,TKCSR      ;SET INTERRUPT ENABLE BIT
002026  016700  176766                  MOV     ROMADD,R0       ;GET ROM ADDRESS
002032  005710                          TST     (R)             ;READ ROM ADDRESS
002034  000776                          BR      .-2             ;LOOP
002036  004567  000140          PRG2A:  JSR     R,RECD          ;GO GET ADDRESS &
002042  000000          PRG2B:  .WORD   0                       ;PUT IT HERE
002044  016700  177772                  MOV     PRG2B,R0
002050  004567  000010                  JSR     R,TYPEM         ;TYPE
002054  002566                          .BYTE   15,12           ;CR/LF
002056  000002                          RTI                     ;EXIT KEYBOARD INTERRUPT SERVICE
002060  104000          PRG2C:  HLT     0                       ;ERROR! DID YOU TYPE AN ODD ADDRESS?
002062  000777                          BR      .               ;SIT HERE UNTIL CORRECT ADDRESS IS TYPED IN


                                        ;ROUTINE TO TYPE A MESSAGE
002064  010026          TYPEM:  MOV     R0,(6)+         ;SAVE REGISTER 0
002066  012500                  MOV     (5)+,R0         ;PLACE MESSAGE ADDRESS IN R0
```

```
002070  112067  176720                  MOVB    (0)+,TER-       ;GET TERMINATOR CHARACTER
002074  112067  176712          TYPE-A: MOVB    (0)+,CHAR       ;GET NEXT CHARACTER
002100  126767  176706  176776          CMPB    CHAR,TER-       ;WAS NEXT CHARACTER THE TERM
002106  001005                          BNE     TYPE-B          ;CHARACTER
002110  014600                          MOV     -(6),R0         ;RESTORE R0
002112  105767  175446                  TSTB    TPCSR
002116  100375                          BPL     .-4
002120  000205                          RTS     5               ;AND EXIT
002122  126727  176664  000045  TYPE-B: CMPB    CHAR,#'%        ;WAS CHARACTER %
002130  001015                          BNE     TYPE-C
002132  105767  175426                  TSTB    TPCSR           ;TEST TELEPRINTER FLAG
002136  100375                          BPL     .-4             ;AND WAIT FOR DONE
002140  012767  000215  175420          MOV     #215,TPDBR      ;LOAD TELEPRINTER WITH CAR. RET
002146  105767  175412                  TSTB    TPCSR           ;TEST TELEPRINTER FLAG
002152  100375                          BPL     .-4             ;AND WAIT FOR DONE
002154  012767  000212  175404          MOV     #212,TPDBR      ;LOAD TELEPRINTER WITH LINE FEED
002162  000744                          BR      TYPE-A          ;GET NEXT CHARACTER
002164  105767  175394          TYPE-C: TSTB    TPCSR           ;TEST TELEPRINTER FLAG
002170  100375                          BPL     .-4             ;AND WAIT FOR DONE
002172  016767  176614  175366          MOV     CHAR,TPDBR      ;LOAD TELEPRINTER BUFFER
002200  000735                          BR      TYPE-A          ;AND GET NEXT CHARACTER

                                ;ROUTINE TO RECEIVE DATA TYPED IN ON THE KEYBOARD, THE DATA IS PLACED IN
                                ;THE ADDRESS FOLLOWING THE JSR CALL!
                                ;               JSR     5,RECD  ;CALL RECEIVE DATA ROUTINE
                                ;               0               ;DATA IS PLACED HERE
002202  005015          RECD:   CLR     (5)             ;CLEAR OUT OLD DATA
002204  105767  175358  RECDA:  TSTB    TKCSR           ;TEST KEYBOARD FLAG
002210  100375                  BPL     .-4             ;AND WAIT FOR CHARACTER
002212  116767  175344  176572  MOVB    TKDBR,CHAR      ;GET CHARACTER
002220  116767  176566  175340  MOV     CHAR,TPDBR      ;ECHO CHARACTER
002226  126727  176560  000215  CMPB    CHAR,#215       ;WAS CHARACTER CARRIAGE RETURN
002234  001005                  BNE     RECDB
002236  005725                  TST     (5)+            ;INCREMENT RETURN ADDRESS
002240  105767  175328          TSTB    TPCSR
002244  100375                  BPL     .-4
002246  000205                  RTS     5               ;AND EXIT
002250  042767  177770  176534  RECDB:  BIC     #177770,CHAR    ;STRIP AWAY ALL BUT 3 LSB
002256  006315                  ASL     (5)             ;ROTATE
002260  006315                  ASL     (5)             ;PREVIOUS
002262  006315                  ASL     (5)             ;DATA
002264  056715  176522          BIS     CHAR,(5)        ;AND INSERT CHARACTER
002270  000745                  BR      RECDA           ;GET NEXT CHARACTER

                                ;SCOPE ROUTINE, THIS ROUTINE IS ENTERED AT THE END OF EACH SUBTEST.
002272  032767  040000  175270  SCOPEC: BIT     #40000,SR       ;TEST SR FOR SCOPE
002300  001023                  BNE     SCOPEB          ;YES SCOPE
002302  032767  004000  175260          BIT     #4000,SR        ;TEST FOR ITERATION
002310  001007                  BNE     SCOPEG          ;INHIBIT ITERATION
002312  026767  000026  000022          CMP     SCOPEF,ICOUNT   ;ITERATION COMPLETE
002320  001403                  BEQ     SCOPEG          ;ITERATION COMPLETE GO TO SCOPEG
002322  005267  000016                  INC     SCOPEF          ;INCREMENT ITERATION COUNT
002326  000410                  BR      SCOPEB          ;GO TO SCOPEB
002330  005067  000010          SCOPEG: CLR     SCOPEF          ;CLEAR ITERATION COUNT
```

```
002334  012667  000006          MOV     #%6,RETURN      ;GET ADDRESS OF NEXT TEST
002340  000002                  RTI                     ;EXIT
002342  000005          ;COUNT; 5
002344  000000          SCOPEF; 0                       ;CONTAINS SUBTEST ITERATION COUNT
002346  000000          RETURN; .WORD   0               ;CONTAINS RETURN PC FOR SCOPE
002350  005726          SCOPEB; TST     (6)+            ;POP PC
002352  012667  175420          MOV     (6)+,PSW        ;RESTORE CONDITION CODES
002356  000177  177764          JMP     @RETURN

                        ;ERROR ROUTINE; THIS ROUTINE IS ENTERED WHEN AN ERROR IS DETECTED;
002362  036727  175202  000200 ERROR;  BIT     SR,#200200      ;;INHIBIT PRINTOUT;
002370  001401                  BEQ     .+4             ;BRANCH IF ERROR PRINT OUT
002372  000002                  RTI                     ;RETURN TO TEST
002374  004567  177464          JSR     %5,TYPEM        ;TYPE ERROR MESSAGE
002400  002532                  ERRORM                  ;/PC=      ,
002402  011667  000016          MOV     (6),D2BTYP      ;TYPE PROGRAM COUNTER
002406  004767  000014          JSR     7,02A
002412  005767  175152          TST     SR              ;HALT ON ERROR?
002416  100001                  BPL     .+4             ;
002420  000000                  HALT                    ;YES HALT
002422  000002                  RTI                     ;RETURN TO TEST

                        ;THIS ROUTINE CONVERTS AN OCTAL NUMBER TO ASCII AND TYPES IT ON THE TTY.
002424  000000          D2BTYP; 0
002426  016746  175132  02A;    MOV     TPCSR,-(6)      ;SAVE TPCSR
002432  010246                  MOV     %2,-(6)         ;SAVE R2
002434  010146                  MOV     %1,-(6)         ;SAVE R1
002436  010046                  MOV     %0,-(6)         ;SAVE R0
002440  016700  177760          MOV     D2BTYP,%0       ;GET DATA TO BE TYPED
002444  012701  000006          MOV     #6,%1           ;GET COUNTER
002450  005002                  CLR     %2              ;CLEAR WORKING REGISTER
002452  006100                  ROL     %0              ;MOV FIRST BIT (MSB) INTO
002454  006102                  ROL     %2              ;R2
002456  062702  000260  02AA;    ADD     #260,%2         ;FORM ASCII CODE
002462  105767  175076          TSTB    TPCSR           ;TEST TELEPRINTER
002466  100375                  BPL     .-4             ;FLAG AND WAIT UNTIL DONE
002470  010267  175072          MOV     %2,TPDBR        ;LOAD TELEPRINTER BUFFER
002474  005002                  CLR     %2              ;CLEAR WORKING REGISTER
002476  006100                  ROL     %0              ;ROTATE THE
002500  006102                  ROL     %2              ;NEXT
002502  006100                  ROL     %0              ;OCTAL CHARACTER
002504  006102                  ROL     %2              ;INTO
002506  006100                  ROL     %0              ;REGISTER
002510  006102                  ROL     %2              ;TWO
002512  005301                  DEC     %1              ;DECREMENT COUNTER
002514  001360                  BNE     02AA            ;GO TO 02AA IF NOT 0
002516  012600                  MOV     (6)+,%0         ;FINISHED;  RESTORE REGISTERS
002520  012601                  MOV     (6)+,%1         ;
002522  012602                  MOV     (6)+,%2         ;
002524  012667  175034          MOV     (6)+,TPCSR      ;AND TPCSR
002530  000207                  RTS     7               ;AND EXIT
```

```
                                        ;ASCII MESSAGES
002532  022520  050040  036503  ERRORM: .ASCII '0% PC= 0'
002540  040040
002542  022520  051120  021507  M6:     .ASCII  '0%PRG#=0'
002550  040075
002552  022500  047522  020115  M7:     .ASCII  '0%ROM DATA%='
002560  040504  040524  040045
002566  022500    100           M8:     .ASCII  '0%0'
002571    100   051045  046517  M9:     .ASCII  '0%ROM ADDRESS/IMAGE ADDRESS ROM DATA=IMAGE DATA%='
002576  040440  042104  042522
002604  051523  044457  040515
002612  042507  040440  042104
002620  042522  051523  051040
002626  046517  042040  052101
002634  025101  046511  043501
002642  020105  040504  040524
002650  040045
002652  027500    100           M10:    .ASCII  '0/0'
002655    100   040040          M11:    .ASCII  '0 0'
002660  025100    100           M12:    .ASCII  '0=0'

        003776                          .=3776
003776  000000                          .WORD
                                        ;DATA CUT INTO THE BM792-YB
004000  013701  177570  000005          013701,177570,000005,010100
004006  010100
004010  012710  177400  020027          012710,177400,020027,177544
004016  177544
004020  001007  012740  004002          001007,012740,004002,005710
004026  005710
004030  100376  005740  100363          100376,005740,100363,022020
004036  022020
004040  012740  000005  105710          012740,000005,105710,100376
004046  100376
004050  005710  100754  105010          005710,100754,105010,000137
004056  000137
004060  000000  000001  177777          000000,000001,177777,177777
004066  177777
004070  177777  177777  177777          177777,177777,177777,177777
004076  177777

        000001                          .END
```

K1

D2BMBA SYMBOL TABLE

```
CHAR    001012      DISPLA = 177570      DONE    001602      DONE1   001634
DUMP    001006      D2BTYP  002424      END     001562      ERROR   002362
ERRORM  002332      ERROR1  001206      ERROR2  001254      HL?     = 104000
ICNT    001000      ICOUNT  002342      IMAGE   001704      LAST    001010
M10     002652      M11     002655      M12     002660      M6      002542
M7      002552      M8      002566      M9      002571      02A     002426
02AA    002456      PRGNUM  001060      PRGTAB  001022      PRGB    001110
PRG0R   001114      PRG1    001640      PRG1A   001656      PRG1B   001674
PRG1C   001712      PRG1D   001762      PRG2    001766      PRG2A   002036
PRG2B   002042      PRG2C   002060      PRMTRS  001030      PSW     = 177776
RECD    002202      RECDA   002204      RECDB   002250      RESTAR  001070
RETURN  002346      ROMADD  001020      SCOPE   = 104400      SCOPEB  002350
SCOPEC  002272      SCOPEF  002344      SCOPEG  002330      SR      = 177570
SRT     001016      START1  002200      START3  002210      STKPTR  = 000000
TERM    001014      TKCSR   = 177560      TKDBR   = 177562      TKINTA  = 000060
TKINTP  = 000062      TPCSR   = 177564      TPDBR   = 177566      TYPEM   002064
TYPEMA  002074      TYPEMB  002122      TYPEMC  002164      T1      001134
T1A     001152      T1B     001214      T2      001216      T2I     001234
T2B     001260      T2C     001274      T2D     001304      T2E     001314
T3      001324      T3A     001346      T3AA    001340      T3B     001354
T3C     001364      T3D     001372      T3E     001402      T3F     001412
T4      001432      T4B     001452      T4C     001456      T4D     001466
T4E     001560      WORDS   001002      .       = 004100
```

ERRORS DETECTED: 0