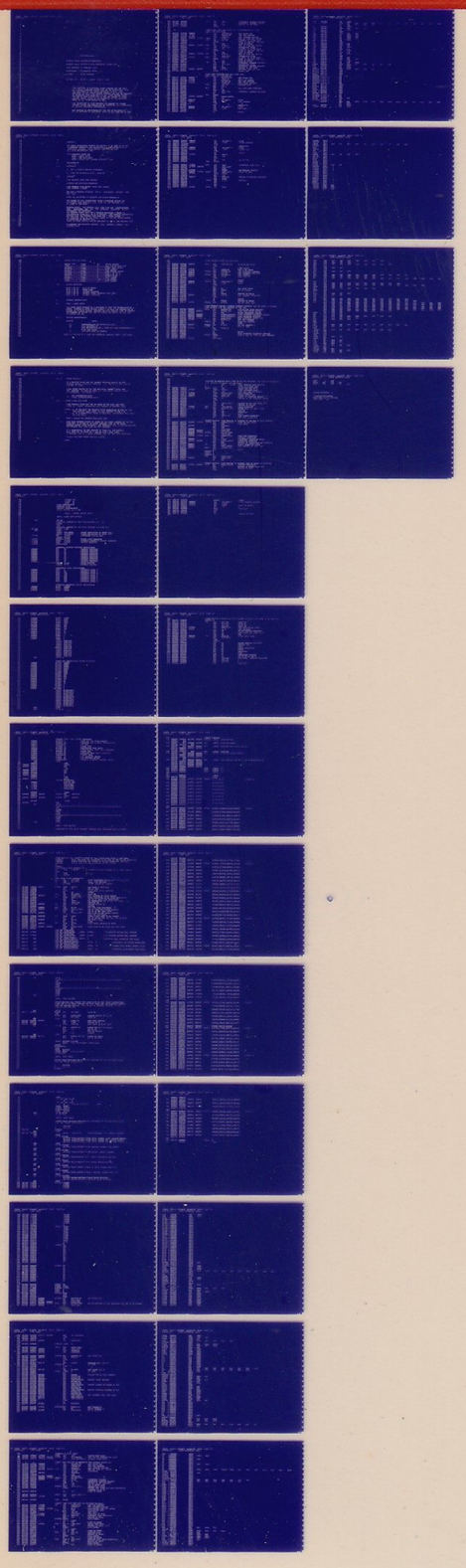
BM792YA-H,K,L

BOOTSTRAP LOADER TEST
MD-11-DZBMA-E

EP-DZBMA-E-DL

FICHE 1 OF 1

MAY 1978
digital
MADE IN USA



```
                                        .REPT   0
```

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZBMA-E-D

PRODUCT NAME: BM792YA-M,K,8L BOOTSTRAP LOADER TEST

DATE CREATED: 13 FEBRUARY 1973

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: BRUCE BURGESS

REVISED BY: DAVID L. ADAMS JUNE 3, 1975

```
                                1.      ABSTRACT

                                        THE DZBMA DIAGNOSTIC PROGRAM IS WRITTEN TO BE USED AS AN AID
                                        TO HARDWARE DEBUGGING AND MAINTENANCE OF THE BM792YA-M,K,BL
                                         BOOTSTRAP LOADERS.  THIS PROGRAM 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.  M792 AND BM792YA-M,K,OR L MODULES

                                2.2     STORAGE

                                        THIS PROGRAM USES CORE 0-ENDAD

                                3.      LOADING AND STARTING PROCEDURE

                                        LOAD PROGRAM INTO MEMORY USING ABS LOADER.
                                        LOAD ADDRESS = 000200

                                        SET SWR = DESIRED STANDARD  PDP-11  DIAGNOSTIC  OPTIONS  (SEE
                                        SECT 4.0)

                                        NOTE: ALL SWITCHES = 0 SELECTS AND STARTS PROGRAM 0

                                        THE NUMBER OF FILL CHARACTERS AFTER A CARRIAGE RETURN MAY
                                        BE PATCHED INTO LOCATION 1101. NINE (9) ARE REQUIRED FOR
                                        AN LA30S AT 300 BAUD.

                                        DEPRESS START.  THE PROGRAM WILL THEN TYPE OUT  INSTRUCTIONS.   ALL
                                        USER   RESPONSES  ARE  VIA  THE  KEYBOARD  (CARRIAGE  RETURN
                                        TERMINATES THE RESPONSE)
                                        THE PROGRAM FIRST ASKS FOR A PROGRAM SELECTION " PRG= ".
                                        THE OPERATOR TYPES 0,1, OR 2 AS DESIRED. (SEE PROGRAM DESCRIPTION
                                        SECTION 3.0) THE PROGRAM THEN ASKS FOR " IMAGE NUMBER ".
                                        THE OPERATOR TYPES IN THE IMAGE NUMBER OF THE ROM BEING TESTED
                                        AS INDICATED IN SECTION 3.1.
                                        THERE IS OTHER SPECIAL DIALOG REQUIRED IN PRG 2. SEE SECTION 5.2.

                                        TO RESTART THE SELECTED PROGRAM  LOAD  ADDRESS = 000210  AND
                                        DEPRESS START
```

```
                                3.1     MODULE TYPE AND IMAGE

                                        MODULE TYPE     IMAGE           NO.     OPTION BOOTED
                                        M792.....IMAGI.....0.....(ALL 1'S NO DATA OUT)
                                        MB792VA..... IMAGA.....1.....PAPER TAPE LOADER
                                        MB792VB..... IMAGB.....2.....DISK LOADER
                                        MB792VC..... IMAGC.....3.....CARD READER LOADER
                                        MB792VD..... IMAGD.....4.....CARD 1 OF MR 11
                                        MB792VE..... IMAGE.....5.....CARD 2 OF MR 11
                                        MB792VF..... IMAGF.....6.....DP588 LOADER
                                        MB792VM..... IMAGM.....7.....TA11 LOADER
                                        M792VK..... IMAGK.....10.....V128 LOADER
                                        M792VL..... IMAGL.....11.....RA11 LOADER

                                3.2       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


                                3.       PROGRAM DESCRIPTIONS

                                3.1      PRG0 - LOGIC TESTS

                                THE LOGIC TESTS CONSIST OF 4 ROUTINES TO TEST THE DM792VA-M,K,OR L
                                LOGIC.  PROGRAM 0 LOOPS WITHIN ITSELF . TO HALT THE PROGRAM OR TO
                                CHANGE THE PROGRAM NUMBER, DEPRESS THE HALT SWITCH, THEN TO RESTART
                                THE PROGRAM AT EITHER LOC. 200 TO CHANGE THE PRG 0, OR LOC. 210
                                TO RESTART PRG 00.

                        3.1.1   ROUTINE DESCRIPTIONS

                                ROUTINE                 TESTS

                                   T1           ADDRESSABILITY OF DM792VA-M,K,OR L
                                   T2           DATA RELIABILITY
                                   T3           THAT DM792VA-M,K,OR L TIMES OUT WHEN REFERENCED BY A
                                                DATIP BUS CYCLE
                                   T4           THAT DATA READ IS CORRECT

                              NOTE: AT THE END OF A PASS THE APPROPATE TERMINAL "BELL" WILL RING.
```

```
                        5.1.2     ERROR PRINTOUT

                                  IF A ROUTINE FAILS AND THE INHIBIT PRINTOUT SWITCH IS NOT
                                  ENABLED (SW13) A PRINTOUT RESULTS.  I.E. THE PC AT THE TIME OF
                                  FAILURE IS TYPED.

                                  IF AN ERROR OCCURS IN T4, THE ROM DATA, 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 THEN
                                  TYPES OUT 'PRG#' REQUESTING WHAT PROGRAM TO PERFORM NEXT.

                                 NOTES:  1. IF YOU HALT THE PROGRAM AFTER COMPLETION OF PRG #1, AND
                                         THEN RESTART (LOC. 210), IT WILL RESTART TO PRG # 0.
                                         2. IF A NONEXISTANT IMAGE IS SELECTED, THE PROGRAM WILL TRAP
                                         OUT AND HALT.

                        5.3       PRG2 - SINGLE RUN ADDRESS READ DATA LOOP

                                  WHEN THIS PROGRAM STARTS IT LOOPS ON THE FIRST ADDRESS OF THE RUN
                                  SELECTED. TO CHANGE TO AN OTHER ADDRESS, TYPE IN THE OCTAL
                                  ADDRESS WANTED (MUST BE EVEN), AND TERMINATE WITH A CARRIAGE RETURN.
                                  THE PROGRAM WILL THEN LOOP ON THAT ADDRESS.

                                  IF A NONEXISTANT OR ODD ADDRESS IS TYPED IN, THE PROGRAM
                                  WILL TYPE OUT AN ERROR PC # AND GO INTO A WAIT LOOP.
                                  TO RECOVER TYPE IN A LEGAL ADDRESS, AND THE TEST WILL CONTINUE.

                                  TO HALT THE TEST PRESS THE HALT SWITCH.

                                 .ENDR
```

```
                                        .ENABLE ABS
                                        .NLIST  MC
                                        .LIST   ME
                                ;LOAD ADDRESS=0200
                                ;DEPRESS START
                                ;RESTART ADDRESS=0210
                                ;STACK POINTER IS AT 500

                                        .MCALL  .STYPE,.SIRAP,.EQUAT

                                .SBTTL  BASIC DEFINITIONS

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

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

                                ;*PRIORITY LEVEL DEFINITIONS
                  000000        PR0=    0               ;PRIORITY LEVEL 0
                  000040        PR1=    40              ;PRIORITY LEVEL 1
                  000100        PR2=    100             ;PRIORITY LEVEL 2
                  000140        PR3=    140             ;PRIORITY LEVEL 3
                  000200        PR4=    200             ;PRIORITY LEVEL 4
                  000240        PR5=    240             ;PRIORITY LEVEL 5
                  000300        PR6=    300             ;PRIORITY LEVEL 6
                  000340        PR7=    340             ;PRIORITY LEVEL 7

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
                  100000        SW15=   100000
                  040000        SW14=   40000
                  020000        SW13=   20000
```

```
(1)             810000                  $M12=   10000
(1)             804000                  $M11=   4000
(1)             802000                  $M10=   2000
(1)             801000                  $M09=   1000
(1)             800400                  $M08=   400
(1)             800200                  $M07=   200
(1)             800100                  $M06=   100
(1)             800040                  $M05=   40
(1)             800020                  $M04=   20
(1)             800010                  $M03=   10
(1)             800004                  $M02=   4
(1)             800002                  $M01=   2
(1)             800001                  $M00=   1
(1)                                     .EQUIV  $M09,$M9
(1)                                     .EQUIV  $M08,$M8
(1)                                     .EQUIV  $M07,$M7
(1)                                     .EQUIV  $M06,$M6
(1)                                     .EQUIV  $M05,$M5
(1)                                     .EQUIV  $M04,$M4
(1)                                     .EQUIV  $M03,$M3
(1)                                     .EQUIV  $M02,$M2
(1)                                     .EQUIV  $M01,$M1
(1)                                     .EQUIV  $M00,$M0
(1)         001                         .IF B <>
(1)
(1)                                     ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)         1$0000                      BIT15=  100000
(1)         0$0000                      BIT14=  40000
(1)         0$0000                      BIT13=  20000
(1)         0$0000                      BIT12=  10000
(1)         0$4000                      BIT11=  4000
(1)         0$2000                      BIT10=  2000
(1)         0$1000                      BIT09=  1000
(1)         0$0400                      BIT08=  400
(1)         0$0200                      BIT07=  200
(1)         0$0100                      BIT06=  100
(1)         0$0040                      BIT05=  40
(1)         0$0020                      BIT04=  20
(1)         0$0010                      BIT03=  10
(1)         0$0004                      BIT02=  4
(1)         0$0002                      BIT01=  2
(1)         0$0001                      BIT00=  1
(1)                                     .EQUIV  BIT09,BIT9
(1)                                     .EQUIV  BIT08,BIT8
(1)                                     .EQUIV  BIT07,BIT7
(1)                                     .EQUIV  BIT06,BIT6
(1)                                     .EQUIV  BIT05,BIT5
(1)                                     .EQUIV  BIT04,BIT4
(1)                                     .EQUIV  BIT03,BIT3
(1)                                     .EQUIV  BIT02,BIT2
(1)                                     .EQUIV  BIT01,BIT1
(1)                                     .EQUIV  BIT00,BIT0
(1)         000                         .ENDC
(1)         001                         .IF B <>
```

```
(1)
(1)                                     ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)            000004                   EMRVEC= 4                ;TIME OUT AND OTHER ERRORS
(1)            000010                   RESVEC= 10               ;RESERVED AND ILLEGAL INSTRUCTIONS
(1)            000014                   TBITVEC=14               ;"T" BIT
(1)            000014                   TRTVEC= 14               ;TRACE TRAP
(1)            000014                   BPTVEC= 14               ;BREAKPOINT TRAP (BPT)
(1)            000020                   IOTVEC= 20               ;INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1)            000024                   PWRVEC= 24               ;POWER FAIL
(1)            000030                   EMTVEC= 30               ;EMULATOR TRAP (EMT) **ERROR**
(1)            000034                   TRAPVEC=34               ;"TRAP" TRAP
(1)            000060                   TKVEC=  60               ;TK KEYBOARD VECTOR
(1)            000064                   TPVEC=  64               ;TK PRINTER VECTOR
(1)            000240                   PIRQVEC=240              ;PROGRAM INTERRUPT REQUEST VECTOR
(1)               000                   .ENDC
209            000020                           .=20
210   000020   002606                           .SCOPE
211   000022   000240                           240
212            000030                           .=30
213   000030   002730                           .MLT
214   000032   000340                           340
215   000034   001000                           TYP
216   000036   000000                           0
217            000046                           .=46
218   000046   002310                           LOGIC
219                                     ;EQUATE STATEMENTS
220            000500                           STKPTR= 500
221            177564                           TPS=177564
222            177566                           TPB=177566
223            177560                           TKS=177560
224            177562                           TKB=177562
225
226            000200                           .=200
227   000200   000137  001374           START:  JMP     @#PHYTRS
228            000210                           .=210
229   000210   000167  001322           START3: JMP     RESTART
230
231            001000                           .=1000
232   001000                           TYP:
(2)               001                   .IF 0
(2)                                     ;*****************************************************
(2)                                     .IFF
(2)                                     .NLIST
(2)                                     .REPT
(2)                                     .LIST
(2)                                     ;*****************************************************
(2)                                     .NLIST
(2)                                     .ENDR
(2)                                     .LIST
(2)               000                   .ENDC
(1)
(1)                                     .SBTTL  TYPE ROUTINE
(1)
(1)                                     ;*ROUTINE TO TYPE ASCIZ MESSAGE, MESSAGE MUST TERMINATE WITH A 0 BYTE.
```

```
(1)                                     ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1)                                     ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1)                                     ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1)                                     ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1)                                     ;*
(1)                                     ;*CALL:
(1)                                     ;*1) USING A TRAP INSTRUCTION
(1)                                     ;*      TYPE    ,MESADR         ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
(1)                                     ;*OR
(1)                                     ;*      TYPE
(1)                                     ;*      MESADR
(1)                                     ;*
(1)                                     ;*2) USING A JSR INSTRUCTION
(1)                                     ;*      MOV     PS,-(SP)        ;PUSH PROCESSOR STATUS WORD ON THE STACK
(1)                                     ;*      JSR     PC,$TYPE        ;CALL TYPE ROUTINE
(1)                                     ;*      MESADDR                 ;FIRST ADRESS OF MESSAGE
(1)
(1)  001000  105767  000111             $TYPE:  TSTB    $TPFLG          ;IS THERE A TERMINAL?
(1)  001004  100002                             BPL     1$              ;BR IF YES
(1)  001006  000000                             HALT                    ;HALT HERE IF NO TERMINAL
(1)  001010  000407                             BR      3$              ;LEAVE
(1)  001012  010046                     1$:     MOV     R0,-(SP)        ;SAVE R0
(1)  001014  017600  000002                     MOV     @2(SP),R0       ;SET ADDRESS OF ASCIZ STRING
(1)  001020  112046                     2$:     MOVB    (R0)+,-(SP)     ;PUSH CHARACTER TO BE TYPED ONTO STACK
(1)  001022  001005                             BNE     4$              ;BR IF IT ISN'T THE TERMINATOR
(1)  001024  005726                             TST     (SP)+           ;IF TERMINATOR POP IT OFF THE STACK
(1)  001026  012600                             MOV     (SP)+,R0        ;RESTORE R0
(1)  001030  062716  000002             3$:     ADD     #2,(SP)         ;ADJUST RETURN PC
(1)  001034  000002                             RTI                     ;RETURN
(1)  001036  004767  000026             4$:     JSR     PC,$TS          ;GO TYPE THIS CHARACTER
(1)  001042  126726  000046             5$:     CMPB    $FILLC,(SP)+    ;IS IT TIME FOR FILLER CHARS.?
(1)  001046  001363                             BNE     2$              ;IF NO GO GET NEXT CHAR.
(1)  001050  016746  000036                     MOV     $NULL,-(SP)     ;GET # OF FILLER CHARS. NEEDED
(1)                                             ;AND THE NULL CHAR.
(1)  001054  105366  000001             6$:     DECB    1(SP)           ;DOES A NULL NEED TO BE TYPED?
(1)  001060  002770                             BLT     3$              ;BR IF NO--GO POP THE NULL OFF OF STACK
(1)  001062  004767  000002                     JSR     PC,$TS          ;GO TYPE A NULL
(1)  001066  000772                             BR      6$              ;LOOP
(1)  001070  105777  000012             7$:     TSTB    @$TPS           ;WAIT UNTIL PRINTER IS READY
(1)  001074  100375                             BPL     7$
(1)  001076  116677  000002  000004             MOVB    2(SP),@$TPB     ;LOAD CHAR TO BE TYPED INTO DATA REG.
(1)  001104  000207                             RTS     PC
(1)                                     .$IF NDF $TPS,$TPS:
(1)  001106  177564                     .$IF EQ .-$TPS,$TPS:    .WORD   177564          ;TTY PRINTER STATUS REG. ADDRESS
(1)                                     .$IF NDF $TPB,$TPB:
(1)  001110  177566                     .$IF EQ .-$TPB,$TPB:    .WORD   177566          ;TTY PRINTER BUFFER REG. ADDRESS
(1)                                     .$IF NDF $NULL,$NULL:
(1)  001112  000                        .$IF EQ .-$NULL,$NULL:  .BYTE   0               ;CONTAINS NULL CHARACTER FOR FILLS
(1)                                     .$IF NDF $FILLS,$FILLS:
(1)  001113  002                        .$IF EQ .-$FILLS,$FILLS:        .BYTE   2               ;CONTAINS # OF FILLER CHARACTERS
(1)                                     .$IF NDF $FILLC,$FILLC:
(1)  001114  012                        .$IF EQ .-$FILLC,$FILLC:        .BYTE   12              ;INSERT FILL CHARS. AFTER A "LIN
(1)                                     .$IF NDF $TPFLG,$TPFLG:
(1)  001115  000                        .$IF EQ .-$TPFLG,$TPFLG:        .BYTE   0               ;"TERMINAL AVAILABLE" FLAG (BIT<
```

```
                    001                 .IF B
(2)                                     ;;*******************************************************************
(2)                                     .IFF
(2)                                     .NLIST
(2)                                     .REPT
(2)                                     .LIST
(2)                                     ;;*******************************************************************
(2)                                     .NLIST
(2)                                     .ENDR
(2)                                     .LIST
(2)                     000             .ENDC
(1)
(1)                                     .SBTTL  TRAP DECODER
(1)
(1)                                     ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
(1)                                     ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
(1)                                     ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
(1)                                     ;*GO TO THAT ROUTINE.
(1)
(1)                     001             .IF B
(1)     001116  010046                  $TRAP:  MOV     R0,-(SP)        ;SAVE R0
(1)                                     .IFF
(1)                                     $TRAP:  MOV     2(SP),@#PS      ;ASSUME STATUS OF CALLER
(1)                                             MOV     R0,-(SP)        ;SAVE R0
(1)                     000             .ENDC
(1)     001120  016600  000002                  MOV     2(SP),R0        ;GET TRAP ADDRESS
(1)     001124  005740                          TST     -(R0)           ;BACKUP BY 2
(1)     001126  111000                          MOVB    (R0),R0         ;GET RIGHT BYTE OF TRAP
(1)                     001             .IF NB
(1)                                             CMP     #$TERM,R0       ;CHECK FOR OUT OF BOUNDS
(1)                                             BGT     .+6             ;BR IF OK
(1)                                             HALT                    ;OUT OF BOUNDS
(1)                                             BR      .-2             ;HANGUP
(1)                     000             .ENDC
(1)     001130  016000  001136                  MOV     $TRPAD(R0),R0   ;INDEX TO TABLE
(1)     001134  000200                          RTS     R0              ;GO TO ROUTINE
(1)
(1)                                     .MACRO  SETTRAP A,B,MSG
(1)                                             $$SET   A,B,\<TRAP+$TRP>,\$TRP,<MSG>
(1)                                     .NLIST
(1)                                     $TRP=$TRP+2
(1)                                     .LIST
(1)                                     .ENDM   SETTRAP
(1)                                     .MACRO  $$SET   A,B,C,D,CUMNT
(1)                                     .IF EQ $TRP
(1)
(1)                                     .SBTTL  TRAP TABLE
(1)
(1)                                     ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
(1)                                     ;*BY THE "TRAP" INSTRUCTION.
(1)
(1)                                     ;       ROUTINE
(1)                                     ;       -------
(1)                                     $TRPAD:
```

```
(1)                                     .ENDC
(1)                                     .IF NDF GNS,.NLIST
(1)                                             A=      C
(1)                                     .IF NDF GNS,.LIST
(1)                                             B       ;CALL=A TRAP=D(C)       COMNT
(1)                                     .ENDM   $$SET
(1)                                     .MACRO  TRMTRP
(1)                                     $!ERM=.-$TRPAD
(1)                                     .ENDM   TRMTRP
(1)             001                     .IF DF STYPE
(3)             002                     .IF EQ STRP
(3)
(3)                                     .SBTTL  TRAP TABLE
(3)
(3)                                     ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
(3)                                     ;*BY THE "TRAP" INSTRUCTION.
(3)
(3)                                     ;       ROUTINE
(3)                                     ;       -------
(3)  001136                             $!RPAD:
(3)             001                     .ENDC
(3)  001136  001000                             $TYPE   ;CALL=TYPE      TRAP=8(104408)  TTY TYPEOUT ROUTINE
(1)             000                     .ENDC
(1)             001                     .IF DF $TYPOC
(1)                                             $$ETTRAP TYPOC,$TYPOC,^/TYPE OCTAL NUMBER (WITH LEADING ZEROS)/
(1)                                             $$ETTRAP TYPOS,$TYPOS,^/TYPE OCTAL NUMBER (NO LEADING ZEROS)/
(1)                                             $$ETTRAP TYPON,$TYPON,^/TYPE OCTAL NUMBER (AS PER LAST CALL)/
(1)             000                     .ENDC
(1)             001                     .IF DF $TYPDS
(1)             000                     .ENDC
(1)             001                     .IF DF $TYPDS
(1)                                             $$ETTRAP TYPDS,$TYPDS,^/TYPE DECIMAL NUMBER (WITH SIGN)/
(1)             000                     .ENDC
(1)             001                     .IF DF $TYPBN
(1)                                             $$ETTRAP TYPBN,$TYPBN,^/TYPE BINARY (ASCII) NUMBER/
(1)             000                     .ENDC
(1)             001                     .IF DF $RDCHR
(1)                                             $$ETTRAP RDCHR,$RDCHR,^/TTY TYPEIN CHARACTER ROUTINE/
(1)             000                     .ENDC
(1)             001                     .IF DF $RDLIN
(1)                                             $$ETTRAP RDLIN,$RDLIN,^/TTY TYPEIN STRING ROUTINE/
(1)             000                     .ENDC
(1)             001                     .IF DF $RDOCT
(1)                                             $$ETTRAP RDOCT,$RDOCT,^/READ AN OCTAL NUMBER FROM TTY/
(1)             000                     .ENDC
(1)             001                     .IF DF $RDDEC
(1)                                             $$ETTRAP RDDEC,$RDDEC,^/READ A DECIMAL NUMBER FROM TTY/
(1)             000                     .ENDC
(1)             001                     .IF DF $$AVREG
(1)                                             $$ETTRAP $AVREG,$$AVREG,^/SAVE R0-R5 ROUTINE/
(1)                                             $$ETTRAP RESREG,$RESREG,^/RESTORE R0-R5 ROUTINE/
(1)             000                     .ENDC
234  001140  173000                    RUMADR: 173000
235  001142  173000                            173000
236  001144  173100                            173100
237  001146  173200                            173200
```

```
   238  001150  173100                          173100
   239  001152  173200                          173200
   240  001154  173200                          173200
   241  001156  173300                          173300
   242  001160  173000                          173000
   243  001162  173400                          173400
   244  001164  000000                          0
   245  001166  004000                IMAGI:    IMAGI
   246  001170  004100                          IMAGA
   247  001172  004200                          IMAGB
   248  001174  004300                          IMAGC
   249  001176  004400                          IMAGD
   250  001200  004500                          IMAGE
   251  001202  004600                          IMAGF
   252  001204  004676                          IMAGH
   253  001206  004776                          IMAGK
   254  001210  005076                          IMAGL
   255  001212  000000                          0
   256  001214  000040                LENGTH:   32.
   257  001216  000040                          32.
   258  001220  000040                          32.
   259  001222  000040                          32.
   260  001224  000040                          32.
   261  001226  000040                          32.
   262  001230  000037                          31.
   263  001232  000040                          32.
   264  001234  000040                          32.
   265  001236  000040                          32.
   266  001240  000000                          0
   267
   268  001242  000020                BLKBOX:   20
   269  001244  000021                          21
   270  001246  000022                          22
   271  001250  000023                          23
   272  001252  000024                          24
   273  001254  000025                          25
   274  001256  000026                          26
   275  001260  000027                          27
   276  001262  000030                          30
   277  001264  000031                          31
   278
   279  001266  000040                WORDS:    32.
   280  001270  004100                IMAGS:    IMAGA
   281  001272  173000                RUNADD:   173000
   282  001274  000000                TEMP:     0
   283  001276  000000                ICNT:     0
   284  001300  001550                PRGTAB:   PRG0
   285  001302  002324                          PRG1
   286  001304  002426                          PRG2
   287  001306  012706  000500        START:    MOV     #STKPTR,SP      ;SET STACK PTR
   288  001312  013746  000004                  MOV     @#4,-(SP)
   289  001316  012737  001356  000004          MOV     #XORA,@#4
   290  001324  012737  000000  177060  MASK:   MOV     #0,@#177060     ;#0 IS MODIFIED BY THE SELECTION AND SET UP OF IMAGES
   291  001332  012637  000004                  MOV     (SP)+,@#4
```

```
   292  001336  012737  177777  001354          MOV     #-1,@#XORFLG
   293  001344  104400                          TYPE
   294  001346  003151                          M3
   295  001350  000137  001536                  JMP     @#RESTART
   296
   297  001354  000000                  XORFLG: .WORD   0
   298
   299  001356  042626                  XORA:   CMP     (%6)+,(%6)+
   300  001360  012637  000004                  MOV     (%6)+,@#4
   301  001364  005037  001354                  CLR     @#XORFLG
   302  001370  000137  001536                  JMP     @#RESTART
   303
   304  001374  012706  000500          PRNTRS: MOV     #STK.PTR,%6     ;SET STACK PTR
   305  001400  005067  000010                  CLR     PRGNUM
   306  001404  104400                          TYPE
   307  001406  003077                          M6
   308  001410  004567  001112                  JSR     5,RECD          ;RECEIVE DATA AND PUT
   309  001414  000000                  PRGNUM: 0                       ;IT HERE
   310  001416  104400                          TYPE
   311  001420  003127                          M1
   312  001422  004567  001100                  JSR     %5,RECD         ;GET IMAGE AND PT
   313  001426  000000                  IMIG:   .WORD   0               ;IT HERE
   314  001430  000241                          CLC
   315  001432  006137  001426                  ROL     @#IMIG          ;ADJUST FOR AN EVEN ADDRESS
   316  001436  013700  001426                  MOV     @#IMIG,%0
   317  001442  062700  001166                  ADD     #IMAGE,%0
   318  001446  011037  001270                  MOV     (%0),@#IMAGE    ;SELECT IMAGE ADDRESS
   319  001452  013700  001426                  MOV     @#IMIG,%0
   320  001456  062700  001214                  ADD     #LENGTH,%0
   321  001462  011037  001266                  MOV     (%0),@#NOWORDS  ;SELECT NUMBER OF WORDES IN ROM
   322  001466  013700  001426                  MOV     @#IMIG,%0
   323  001472  062700  001140                  ADD     #ROMADD,%0
   324  001476  011037  001272                  MOV     (%0),@#ROMADD   ;SELECT STARTING ADDRESS OF ROM
   325  001502  013700  001426                  MOV     @#IMIG,%0
   326  001506  062700  001242                  ADD     #BLKBOX,%0
   327  001512  011067  177610                  MOV     (%0),MASK*2     ;SET CONSTANT FOR "XOR" MASK
   328  001516  011067  000110                  MOV     (%0),MASK1*2
   329  001522  011067  001112                  MOV     (%0),MASK2*2
   330  001526  104400                          TYPE
   331  001530  003124                          M8
   332  001532  000137  001306                  JMP     @#START2
   333
   334  001536  016700  177652          RESTART:MOV     PRGNUM,%0       ;SET PROGRAM #
   335  001542  006300                          ASL     %0              ;SHIFT PROGRAM #
   336  001544  000170  001300                  JMP     @PRGTAB(0)      ;GO TO PROGRAM
   337
```

```
                                        ;PROGRAM 8 LOGIC TEST8
001550  005067  177522          PRG8:   CLR     ICNT            ;CLEAR PASS COUNT
001554  012767  001570  001134  PRGBR:  MOV     #T1,LASTPC      ;SET RETURN ADDRESS FOR SCOPE
001562  016737  177510  177570          MOV     ICNT,@#DISPLAY  ;DISPLAY PASS COUNT

                                        ;TEST1 TEST ABILITY TO REFERENCE ROM WITHOUT TIMING OUT
001570  012706  000500          T1:     MOV     #STKPTR,SP      ;SET STACK PTR
001574  016700  177472                  MOV     ROMADD,R0       ;SET ROM ADDRESS
001600  016701  177462                  MOV     WORDS,R1        ;SET ADDRESS COUNTER
001604  012767  001650  176172  T1A:    MOV     #ERROR1,4       ;SET UP TIME OUT VECTOR
001612  011003                          MOV     (R0),R3         ;REFERENCE
001614  005737  001354                  TST     @#XORFLG
001620  100006                          BPL     A1B
001622  012737  001656  000004          MOV     #ERR2,@#4
001630  012737  000000  177060  MASK1:  MOV     #0,@#177060
001636  062700  000002          A1B:    ADD     #2,R0           ;INCREMENT POINTER
001642  005301                          DEC     R1              ;DECREMENT ADDRESS COUNTER
001644  001357                          BNE     T1A             ;BRANCH IF NOT FINISHED
001646  000406                          BR      T1B             ;GO TO SCOPE LOOP
001650  022626                  ERROR1: CMP     (SP)+,(SP)+     ;REPOSITION STACK
001652  104000                          ERROR                   ;ERROR1 ROM TIMED OUT WHEN REFERENCED
                                                                ;ADDRESS IS IN R0
001654  000753                          BR      T1A             ;LOOP ON ERROR

001656  022626          ERR2:   CMP     (SP)+,(SP)+
001660  000167  177720                  JMP     T1A

001664  000004          T1B:    SCOPE

                                        ;TEST2  TEST THAT ROM DATA CAN BE READ RELIABLY.
001666  016700  177480          T2:     MOV     ROMADD,R0       ;SET ROM ADDRESS
001672  016701  177370                  MOV     WORDS,R1        ;SET ADDRESS COUNTER
001676  012767  000006  176100          MOV     #6,4            ;INITIALIZE TIME OUT VECTOR
001704  005067  177364          T2A:    CLR     TEMP            ;INITIALIZE TEMP
001710  011003                          MOV     (R0),R3         ;SET DATA
001712  062067  177356                  ADD     (R0)+,TEMP      ;ADD DATA TO TEMP
001716  166703  177352                  SUB     TEMP,R3         ;SUBTRACT DATA FROM DATA
001722  001403                          BEQ     T2B             ;BRANCH IF EQUAL
001724  104000          ERROR2: ERROR                           ;DATA ERROR
001726  005740                          TST     -(R0)
001730  000765                          BR      T2A             ;LOOP ON ERROR
001732  046067  177336          T2B:    BIC     -(R0),TEMP      ;CLEAR TEMP BITS
001736  001403                          BEQ     T2C             ;BRANCH IF EQUAL TO 0
001740  104000                          ERROR                   ;DATA ERROR
001742  005720                          TST     (R0)+
001744  000772                          BR      T2B             ;LOOP ON ERROR
001746  021810          T2C:    CMP     (R0),(R0)               ;COMPARE DATA
001750  001402                          BEQ     T2D             ;BRANCH IF EQUAL
001752  104000                          ERROR                   ;DATA ERROR
001754  000774                          BR      T2C             ;LOOP ON ERROR
001756  142040          T2D:    CMPB    (R0)+,-(R0)             ;COMPARE DATA (BYTE OPERATION)
001760  001402                          BEQ     T2E             ;BRANCH IF EQUAL
001762  104000                          ERROR                   ;DATA ERROR
001764  000774                          BR      T2D             ;LOOP ON ERROR
```

```
001766  005720                    T2E:    TST     (0)+            ;INCREMENT ADDRESS POINTER
001770  005301                            DEC     %1              ;DECREMENT ADDRESS COUNTER
001772  001344                            BNE     T2A             ;RETURN IF NOT DONE
001774  000004                            SCOPE

                                  ;TEST3 TEST THAT ROM TIMES OUT IF REFERENCED BY OTHER
                                  ;THAN DATI BUS CYCLE
001776  012706  000500            T3:     MOV     #STKPTR,%6      ;SET STACK PTR
002002  016700  177264                    MOV     ROMADD,%0       ;GET ROM ADDRESS
002006  016701  177254                    MOV     WORDS,%1        ;GET ADDRESS COUNTER
002012  012767  002026  175764    T3AA:   MOV     #T30,4          ;SET UP TIME OUT VECTOR
002020  010010                    T3A:    MOV     %0,(0)          ;ATTEMPT TO ALTER DATA
002022  104000                            ERROR                   ;HERE IF DID NOT TIME OUT
002024  000775                            BR      T3A             ;LOOP ON ERROR
002026  012767  002044  175750    T3B:    MOV     #T3D,4          ;SET UP TIME OUT VECTOR
002034  022626                            CMP     (6)+,(6)+       ;REPOSITION STACK
002036  005210                    T3C:    INC     (0)             ;ATTEMPT TO ALTER DATA
002040  104000                            ERROR                   ;HERE IF DID NOT TIME OUT
002042  000775                            BR      T3C             ;LOOP ON ERROR
002044  012767  002064  175732    T3D:    MOV     #T3F,4          ;SET UP TIME OUT VECTOR
002052  022626                            CMP     (6)+,(6)+       ;REPOSITION STACK
002054  005077  177212            T3E:    CLR     @ROMADD         ;ATTEMPT TO ALTER DATA
002060  104000                            ERROR                   ;HERE IF DID NOT TIME OUT
002062  000774                            BR      T3E             ;LOOP ON ERROR
002064  005720                    T3F:    TST     (0)+            ;INCREMENT ADDRESS POINTER
002066  022626                            CMP     (6)+,(6)+       ;REPOSITION STACK
002070  005301                            DEC     %1              ;DECREMENT ADDRESS COUNTER
002072  001347                            BNE     T3AA            ;RETURN IF NOT DONE
002074  012737  000006  000004            MOV     @6,@#4          ;RESTORE TIME OUT TRAP
002102  000004                            SCOPE                   ;SCOPE LOOP

                                  ;THIS TEST COMPARES ROM AND IMAGE DATA
                                  ;AND TYPES OUT DIFFERENCES
002104  012706  000500            T4:     MOV     #STKPTR,%6      ;SET STACK PTR
002110  016701  177152                    MOV     WORDS,%1        ;GET # OF WORDS
002114  016700  177152                    MOV     ROMADD,%0       ;GET ROM ADDRESS
002120  016703  177144                    MOV     IMAGE,%3        ;GET IMAGE ADDRESS
002124  021013                    T4B:    CMP     (0),(3)         ;COMPARE DATA
002126  001004                            BNE     T4D
002130  005301                    T4C:    DEC     %1              ;ALL DATA BEEN COMPARED
002132  001444                            BEQ     T4E
002134  022023                            CMP     (0)+,(3)+       ;INCREMENT ADDRESS POINTERS
002136  000772                            BR      T4B
002140  104000                    T4D:    ERROR
002142  032737  177570  020000            BIT     @#SWR,#20000
002150  001035                            BNE     T4E
002152  010067  000616                    MOV     %0,D2BTYP       ;TYPE
002156  004767  000614                    JSR     7,O2A           ;ROM ADDRESS
002162  104400                            TYPE
002164  003206                            M10                     ;SEPARATOR
002166  011067  000602                    MOV     (0),D2BTYP      ;TYPE
002172  004767  000600                    JSR     7,O2A           ;ROM DATA
002176  104400                            TYPE
002200  003124                            M0                      ;CR/LF
```

```
   447  002202  010367  000566                  MOV     %3,DZBTYP       ;TYPE
   448  002206  004767  000564                  JSR     7,DZA           ;IMAGE ADDRESS
   449  002212  104400                          TYPE
   450  002214  003212                          M12                     ;SEPARATOR
   451  002216  011367  000552                  MOV     (3),DZBTYP      ;TYPE
   452  002222  004767  000550                  JSR     7,DZA           ;IMAGE DATA
   453  002226  104400                          TYPE
   454  002230  003124                          M8                      ;CR/LF
   455  002232  005737  177570                  TST     @#8WR
   456  002236  100334                          BPL     T4C
   457  002240  000000                          HALT
   458  002242  000732                          BR      T4C             ;GO TO T4C
   459  002244  000004              T4E:        SCOPE
   460
   461  002246  005267  177024      END:        INC     ICNT            ;INCREMENT PASS COUNT
   462  002252  026727  177020  000100          CMP     ICNT,#100
   463  002260  001402                          BEQ     DONE
   464  002262  000167  177266                  JMP     PRGBR           ;GO RESTART PROGRAM
   465  002266  012737  000207  177566  DONE:   MOV     #207,@#TPB      ;RING THE BELL
   466  002274  105737  177564                  TSTB    @#TPS
   467  002300  100375                          BPL     .-4
   468  002302  013700  000042                  MOV     @#42,%0         ;RETURN TO DECTAPE MONITOR?
   469  002306  001404                          BEQ     DONE1
   470  002310  004710              LOGIC:      JSR     7,(0)           ;RETURN!
   471  002312  000240                          NOP
   472  002314  000240                          NOP
   473  002316  000240                          NOP
   474  002320  000167  177224      DONE1:      JMP     PRGB
   475
```

```
                                        ;THIS PROGRAM TYPES OUT ROM DATA
                                        ;
002324  012706  000500          PRG1:   MOV     #STKPTR,SP      ;INITIALIZE STACK
002330  104400                          TYPE
002332  003107                          M7                      ;'ROM DATA'
002334  016701  176726                  MOV     WORDS,R1        ;GET # OF WORDS
002340  016700  176726          PRG1A:  MOV     ROMADD,R0       ;GET STARTING ADDRESS
002344  012702  000012                  MOV     #12,R2          ;GET ADDRESS INDICATOR
002350  010067  000420          PRG1B:  MOV     R0,D2BTYP       ;GET ADDRESS
002354  004767  000416                  JSR     7,D2A           ;AND TYPE IT
002360  104400                          TYPE
002362  003124                          M8                      ;CR/LF
002364  012067  000404          PRG1C:  MOV     (R0)+,D2BTYP    ;TYPE
002370  004767  000402                  JSR     7,D2A           ;DATA
002374  104400                          TYPE
002376  003210                          M11
002400  005301                          DEC     R1              ;ALL DATA TYPED
002402  001407                          BEQ     PRG1D           ;GO TO FINISH
002404  005302                          DEC     R2
002406  001366                          BNE     PRG1C           ;RETURN TO PRG1B
002410  012702  000012                  MOV     #12,R2          ;GET ADDRESS INDICATOR
002414  104400                          TYPE
002416  003124                          M8                      ;CR/LF
002420  000753                          BR      PRG1B           ;RETURN TO PRG1B
002422  000167  176746          PRG1D:  JMP     PRMTR8          ;GO GET NEXT TEST
                                        ;
                                        ;THIS PROGRAM CYCLES A SINGLE ADDRESS (ADDRESS MUST BE EVEN) TO CHANGE
                                        ;THE ADDRESS TYPE NEW ADDRESS ON THE TTY.
002426  012706  000500          PRG2:   MOV     #STKPTR,SP      ;INITIALIZE STACK POINTER
002432  012737  002516  000004          MOV     #PRG2C,#4       ;LOAD TRAP ERROR VECTOR
002440  005067  175332                  CLR     PSW             ;CLEAR PROCESSOR STATUS
002444  012737  002476  000060          MOV     #PRG2A,#KYVEC   ;LOAD KEYBOARD INTERRUPT VECTOR
002452  012737  000340  000062          MOV     #340,#KYVEC+2   ;LOAD KEYBOARD PRIORITY
002460  012737  000100  177560          MOV     #100,#KYCS      ;SET INTERRUPT ENABLE BIT
002466  016700  176600                  MOV     ROMADD,R0       ;GET ROM ADDRESS
002472  005710          PRG21:  TST     (R0)            ;READ ROM ADDRESS
002474  000776                  BR      .-2             ;LOOP
002476  004567  000024          PRG2A:  JSR     5,RECD          ;GO GET ADDRESS &
002502  000000          PRG2B:  0                       ;PUT IT HERE
002504  016700  177772                  MOV     PRG2B,R0
002510  104400                          TYPE
002512  003124                          M8                      ;CR/LF
002514  000002                          RTI                     ;EXIT KEYBOARD INTERRUPT SERVICE
002516  104000          PRG2C:  ERROR                   ;ERROR! DID YOU TYPE AN ODD ADDRESS?
002520  000001                  WAIT
002522  000167  177744                  JMP PRG21
```

```
                                        ;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
002526  005015                  RECD:   CLR     (5)             ;CLEAR OUT OLD DATA
002530  105737  177560          10:     TSTB    @#TKS           ;TEST KEYBOARD FLAG
002534  100375                          BPL     .-4             ;AND WAIT FOR CHARACTER
002536  117746  177562                  MOVB    @#TKB,-(SP)     ;GET CHARACTER
002542  042716  000200                  BIC     #200,(SP)       ;STRIP PARITY BIT
002546  122716  000015                  CMPB    #15,(SP)        ;CHECK IF CARRIAGE RETURN
002552  001004                          BNE     20              ;BRANCH IF NOT CARRIAGE RETURN
002554  104400                          TYPE
002556  003124                          MB
002560  062526                          CMP     (R5)+,(SP)+     ;ADJUST R5 AND THE STACK PTR
002562  000205                          RTS     R5              ;RETURN TO CALLER
002564  011637  177566          20:     MOV     (SP),@#TPB      ;ECHO CHARACTER
002570  042716  177770          30:     BIC     #177770,(SP)    ;STRIP AWAY ALL BUT 3 LSB
002574  006315                          ASL     (5)             ;ROTATE
002576  006315                          ASL     (5)             ;PREVIOUS
002600  006315                          ASL     (5)             ;DATA
002602  052615                          BIS     (SP)+,(5)       ;AND INSERT CHARACTER
002604  000751                          BR      10              ;GET NEXT CHARACTER

                                        ;@SCOPE ROUTINE. THIS ROUTINE IS ENTERED AT THE END OF EACH SUBTEST.
002606  032737  040000  177570  .SCOPE: BIT     #40000,@#SWR    ;TEST SR FOR SCOPE
002614  001033                          BNE     $28             ;YES SCOPE
002616  005737  001354                  TST     @#XORFLG
002622  100012                          BPL     A38
002624  013746  000004                  MOV     @#4,-(%6)
002630  012737  002720  000004          MOV     #XOR,@#4
002636  012737  000000  177060  MASK2:  MOV     #0,@#177060
002644  012637  000004                  MOV     (%6)+,@#4
002650  032737  004000  177570  A38:    BIT     #4000,@#SWR     ;TEST FOR ITERATION
002656  001006                          BNE     10              ;INHIBIT ITERATION
002660  005267  000030                  INC     ITCNT           ;INCREMENT ITERATION COUNT
002664  026767  000024  000020          CMP     ITCNT,ICOUNT    ;ITERATION COMPLETE
002672  003004                          BGT     $28             ;BRANCH IF ITERATIONS NOT COMPLETE
002674  005067  000014          10:     CLR     ITCNT           ;CLEAR ITERATION COUNT
002700  011667  000012                  MOV     (SP),LASTPC     ;GET ADDRESS OF NEXT TEST
002704  016716  000006          $28:    MOV     LASTPC,(SP)
002710  000002                          RTI                     ;EXIT
002712  000005                  ICOUNT: 5
002714  000000                  ITCNT:  0
002716  000000                  LASTPC: 0
002720  022626                  XOR:    CMP     (%6)+,(%6)+
002722  012637  000004                  MOV     (%6)+,@#4
002726  000766                          BR      $28

                                        ;ERROR ROUTINE. THIS ROUTINE IS ENTERED WHEN AN ERROR IS DETECTED.
002730  032737  177570  020000  .MLT:   BIT     @#SWR,#20000    ;INHIBIT PRINTOUT
002736  001401                          BEQ     .+4             ;BRANCH IF ERROR PRINT OUT
002740  000002                          RTI                     ;RETURN TO TEST
002742  104400                          TYPE
```

```
 578  002744  003070                  ERRORM                       ;'PC=      '
 579  002746  011667  000022          MOV     (6),DZBTYP           ;TYPE PROGRAM COUNTER
 580  002752  004767  000020          JSR     7,OZA
 581  002756  005737  177570          TST     @#SWR                ;HALT ON ERROR?
 582  002762  100001                  BPL     .+4                  ;
 583  002764  000000                  HALT                         ;YES HALT
 584  002766  104400                  TYPE
 585  002770  003124                  M0
 586  002772  000002                  RTI                          ;RETURN TO TEST
 587
```

```
                                        ;THIS ROUTINE CONVERTS AN OCTAL NUMBER TO ASCII AND TYPES IT ON THE TTY.
003774  000000                          D28TYP: 0
003776  010246                          D4A:    MOV     %2,-(6)         ;SAVE R2
004000  010146                                  MOV     %1,-(6)         ;SAVE R1
004002  010046                                  MOV     %0,-(6)         ;SAVE R0
004004  016700  177764                          MOV     D28TYP,%0       ;GET DATA TO BE TYPED
004010  012701  000006                          MOV     #6,%1           ;GET COUNTER
004014  005002                                  CLR     %2              ;CLEAR WORKING REGISTER
004016  006100                                  ROL     %0              ;MOV FIRST BIT (M*0) INTO
004020  006102                                  ROL     %2              ;R2
004022  062702  000260          10:     ADD     #260,%2         ;FORM ASCII CODE
004026  010267  177742                          MOV     R2,D28TYP
004032  104400                                  TYPE
003774                                          D28TYP
004036  005002                                  CLR     %2              ;CLEAR WORKING REGISTER
004040  006100                                  ROL     %0              ;ROTATE THE
004042  006102                                  ROL     %2              ;NEXT
004044  006100                                  ROL     %0              ;OCTAL CHARACTER
004046  006102                                  ROL     %2              ;INTO
004050  006100                                  ROL     %0              ;REGISTER
004052  006102                                  ROL     %2              ;TWO
004054  005301                                  DEC     %1              ;DECREMENT COUNTER
004056  001361                                  BNE     1B              ;GO TO 1B IF NOT 0
004060  012600                                  MOV     (6)+,%0         ;FINISHED,  RESTORE REGISTERS
004062  012601                                  MOV     (6)+,%1         ;
004064  012602                                  MOV     (6)+,%2         ;
004066  000207                                  RTS     7               ;AND EXIT
```

```
617                                             ;ASCII MESSAGES
618  003070  005015  041520  020075  ERRORM:  .ASCIZ  <15><12>"PC= "
     003076      000
619  003077      015  050012  043522  M0:      .ASCIZ  <15><12>"PRG#="
     003104  036443      000
620  003107      015  051012  046517  M7:      .ASCIZ  <15><12>"NOM DATA"<15><12>
     003114  042040  052101  006501
     003122  000012
621  003124  005015      000          M8:      .ASCIZ  <15><12>
622  003127      015  044412  040515  M1:      .ASCIZ  <15><12>"IMAGE NUMBER=  "
     003134  042507  047040  046525
     003142  042502  036522  020040
     003150      000
623  003151      015  054412  052517  M9:      .ASCIZ  <15><12>"YOU ARE ON AN XOR TESTER!"<15><12>
     003156  040440  042522  047440
     003164  020116  047101  054040
     003172  051117  052040  051505
     003200  042524  006522  000012
624  003206  000057                   M10:     .ASCIZ  "/"
625  003210  000040                   M11:     .ASCIZ  " "
626  003212  000052                   M12:     .ASCIZ  "*"
627
628          003776                           .=3776
629  003776  000000                   .WORD
630  004000  177777  177777  177777  IMAGE:   -1,-1,-1,-1                    ;IMAGE 0
     004006  177777
631  004010  177777  177777  177777           -1,-1,-1,-1
     004016  177777
632  004020  177777  177777  177777           -1,-1,-1,-1
     004026  177777
633  004030  177777  177777  177777           -1,-1,-1,-1
     004036  177777
634  004040  177777  177777  177777           -1,-1,-1,-1
     004046  177777
635  004050  177777  177777  177777           -1,-1,-1,-1
     004056  177777
636  004060  177777  177777  177777           -1,-1,-1,-1
     004066  177777
637  004070  177777  177777  177777           -1,-1,-1,-1
     004076  177777
638
639  004100  012701  160000  012702  IMAGA:   012701,160000,012702,000006    ;IMAGE 1
     004106  000006
640  004110  012703  173100  005012           012703,173100,005012,010742
     004116  010742
641  004120  110706  014304  005714           110706,014304,005714,100775
     004126  100775
642  004130  010712  012706  000024           010712,012706,000024,010441
     004136  010441
643  004140  040601  010111  011102           040601,010111,011102,005214
     004146  005214
644  004150  105714  100376  116412           105714,100376,116412,000002
     004156  000002
645  004160  005211  120227  000375           005211,120227,000375,001366
```

```
                004166  001366
646             004170  105222  000142  177560          105222,000142,177560,177550
                004176  177550
647             004200  013701  177570  000005  IMAGB1  013701,177570,000005,010100      ;IMAGE 2
                004206  010100
648             004210  012710  177400  020027          012710,177400,020027,177344
                004216  177344
649             004220  001007  012740  004002          001007,012740,004002,005710
                004226  005710
650             004230  100376  005740  100363          100376,005740,100363,022020
                004236  022020
651             004240  012740  000005  105710          012740,000005,105710,100376
                004246  100376
652             004250  005710  100754  105010          005710,100754,105010,000137
                004256  000137
653             004260  000000  000001  177777          000000,000001,177777,177777
                004266  177777
654             004270  177777  177777  177777          177777,177777,177777,177777
                004276  177777
655             004300  000005  012700  177160  IMAGC1  000005,012700,177160,010001      ;IMAGE 3
                004306  010001
656             004310  032721  001400  001371          032721,001400,001371,005210
                004316  005210
657             004320  005003  005004  031027          005003,005004,031027,040000
                004326  040000
658             004330  001372  105710  100373          001372,105710,100373,000303
                004336  000303
659             004340  151103  005104  100772          151103,005104,100772,121761
                004346  121761
660             004350  000001  001405  003002          000001,001405,003002,010302
                004356  010302
661             004360  000757  010322  000755          000757,010322,000755,031027
                004366  031027
662             004370  040000  001775  000005          040000,001775,000005,000113
                004376  000113
663             004400  010702  000451  177462  IMAGD1  010702,000451,177462,000005      ;IMAGE 4
                004406  000005
664             004410  010702  000445  177406          010702,000445,177406,000005
                004416  000005
665             004420  010702  000417  177344          010702,000417,177344,000005
                004426  000005
666             004430  004003  100000  024000          004003,100000,024000,010702
                004436  010702
667             004440  000410  172524  060003          000410,172524,060003,060011
                004446  060011
668             004450  000200  100000  010702          000200,100000,010702,000423
                004456  000423
669             004460  176716  000005  010200          176716,000005,010400,005720
                004466  005720
670             004470  012001  005311  005720          012001,005311,005720,012041
                004476  012041
671             004500  031011  001776  005720  IMAGE1  031011,001776,005720,031041      ;IMAGE 5
                004506  031041
672             004510  001406  000112  173100          001406,000112,173100,000340
```

```
                004516  000340
    673         004520  010702  000401  177450          010702,000401,177450,000005
                004526  000005
    674         004530  010200  005720  012001          010200,005720,012001,012711
                004536  012711
    675         004540  177000  011041  032711          177000,011041,032711,100200
                004546  100200
    676         004550  001775  100757  005007          001775,100757,005007,000000
                004556  000000
    677         004560  173110  000340  173220          173110,000340,173220,000340
                004566  000340
    678         004570  173154  000340  173120          173154,000340,173120,000340
                004576  000340
    679
    680         004600  012701  177406  000405  IMAGP1  012701,177406,000405,012701      ;IMAGE 6
                004606  012701
    681         004610  177462  000402  012701          177462,000402,012701,177344
                004616  177344
    682         004620  000005  010100  012710          000005,010100,012710,177400
                004626  177400
    683         004630  020027  177344  001007          020027,177344,001007,012740
                004636  012740
    684         004640  004002  005710  100376          004002,005710,100376,005740
                004646  005740
    685         004650  100363  022020  012740          100363,022020,012740,000005
                004656  000005
    686         004660  105710  100376  005710          105710,100376,005710,100754
                004666  100754
    687         004670  000005  000137  000000          000005,000137,000000
    688         004676  012700  177500  005010  IMAGM1  012700,177500,005010,010701      ;IMAGE 7
                004704  010701
    689         004706  062701  000052  012702          062701,000052,012702,000375
                004714  000375
    690         004716  112103  112110  100413          112103,112110,100413,130310
                004724  130310
    691         004726  001776  105202  100772          001776,105202,100772,116012
                004734  116012
    692         004736  000002  120337  000000          000002,120337,000000,001767
                004744  001767
    693         004746  000000  000755  005710          000000,000755,005710,100774
                004754  100774
    694         004756  005007  017640  002415          005007,017640,002415,112024
                004764  112024
    695         004766  000000  000000  173300          000000,000000,173300,000340
                004774  000340
    696         004776  012701  160000  012702  IMAGK1  012701,160000,012702,000006      ;IMAGE 10
                005004  000006
    697         005006  012703  173100  005012          012703,173100,005012,010742
                005014  010742
    698         005016  110706  014304  005714          110706,014304,005714,100775
                005024  100775
    699         005026  010712  012706  000024          010712,012706,000024,010441
                005034  010441
    700         005036  040601  010111  011102          040601,010111,011102,005214
```

```
        005044  005214
   701  005046  105714  100376  116412          105714,100376,116412,000002
        005054  000002
   702  005056  005211  120227  000375          005211,120227,000375,001366
        005064  001366
   703  005066  105222  000142  177560          105222,000142,177560,175610
        005074  175610
   704
   705  005076  005002  052702  100247  IMAGL1  005002,052702,100247,012701      ;IMAGE 11
        005104  012701
   706  005106  177170  130211  001776          177170,130211,001776,112703
        005114  112703
   707  005116  000007  010100  010220          000007,010100,010220,000402
        005124  000402
   708  005126  012710  000001  006203          012710,000001,006203,103402
        005134  103402
   709  005136  112711  111023  030211          112711,111023,030211,001776
        005144  001776
   710  005146  100756  103766  105711          100756,103766,105711,100771
        005154  100771
   711  005156  005000  022710  000240          005000,022710,000240,001347
        005164  001347
   712  005166  122702  000247  005500          122702,000247,005500,005007
        005174  005007
   713
   714  005176  000000                  ENDABS  0
   715          000001                          .END
```

```
A1B     001636              351     3540
A3B     002650              552     5570
BIT0  = 000001              2010
BIT00 = 000001              2010
BIT01 = 000002              2010
BIT02 = 000004              2010
BIT03 = 000010              2010
BIT04 = 000020              2010
BIT05 = 000040              2010
BIT06 = 000100              2010
BIT07 = 000200              2010
BIT08 = 000400              2010
BIT09 = 001000              2010
BIT1  = 000002              2010
BIT10 = 002000              2010
BIT11 = 004000              2010
BIT12 = 010000              2010
BIT13 = 020000              2010
BIT14 = 040000              2010
BIT15 = 100000              2010
BIT2  = 000004              2010
BIT3  = 000010              2010
BIT4  = 000020              2010
BIT5  = 000040              2010
BIT6  = 000100              2010
BIT7  = 000200              2010
BIT8  = 000400              2010
BIT9  = 001000              2010
BLKBOX  001242              2600    326
BPTVEC= 000014              2010
DISPLA= 177570              2010    342*
DONE    002266              463     4650
DONE1   002328              469     4740
D2BTYP  002774              439*    443*    447*    451*    485*    489*    579*    5980    594     688*    682
EMTVEC= 000030              2010
END     002246              4610
ENDAD   005176              7140
ERRORM  003070              570     6180
ERROR1  001650              348     3580
ERROR2  001724              3770
ERRVEC= 000004              2010
ERR2    001656              352     3630
GN8   = ****** U            233
ICNT    001276              2830    348*    342     461*    462
ICOUNT  002712              568     5660
IMAGA   004100              246     288     6390
IMAGB   004200              247     6470
IMAGC   004300              248     6550
IMAGD   004400              249     6630
IMAGE   004500              250     6710
IMAGP   004600              251     6880
IMAGM   004676              252     6880
IMAGI   004000              245     6300
IMAGK   004776              253     6960
```

```
IMAGL   000076            254     705#
IMAGB   001270            288#    318#    429
IMAG1   001166            245#    317
IMIG    001426            313#    315#    316     319     322     325
IOTVEC# 000020            281#
ITCNT   002714            559#    560     562#    567#
LASTPC  002716            341#    563#    564     568#
LENGTH  001214            256#    328
LOGIC   002310            218     478#
MASK    001324            298#    327#
MASK1   001630            328#    353#
MASK2   002636            329#    555#
M1      003127            31:     622#
M10     003206            442     624#
M11     003210            492     625#
M12     003212            458     626#
M3      003151            294     623#
M6      003077            387     619#
M7      003107            481     628#
M8      003124            331     446     454     488     499     518     537     585     621#
O2A     002776            448     444     448     452     486     498     588     591#
PC    =%000007            201#    232#
PIRQ  = 177772            201#
PIRQVE# 000240            201#
PRGNUM  001414            385#    389#    334
PRGTAB  001388            284#    336
PRG0    001550            284     348#    474
PRG0R   001554            341#    464
PRG1    002324            285     479#
PRG1A   002340            483#
PRG1B   002350            485#    588
PRG1C   002364            489#    496
PRG1D   002422            494     581#
PRG2    002426            286     585#
PRG2A   002476            588     514#
PRG2B   002502            515#    516
PRG2C   002516            586     528#
PRG21   002472            512#    522
PRMTRS  001374            227     384#    581
PR0   = 000000            201#
PR1   = 000240            201#
PR2   = 000300            201#
PR3   = 000140            201#
PR4   = 000200            201#
PR5   = 000240            201#
PR7   = 000340            201#
PS    = 177776            201#
PSW   = 177776            201#    587#
PWRVEC# 000024            201#
RECD    002526            388     312     514     529#
RESTAR  001536            229     295     382     334#
RESVEC# 000010            201#
ROMADD  001272            201#    324#    346     369     401     414#    428     483     511
ROMADB  001140            234#    323
```

```
RP6     = 000300            2010
R0      =%000000            2010    232#    233#
R1      =%000001            2010
R2      =%000002            2010    608
R3      =%000003            2010
R4      =%000004            2010
R5      =%000005            2010    538     539#
R6      =%000006            2010
R7      =%000007            2010
SP      =%000006            2010    232#    233#    345#    532#    533#    534     538     548     541#    545     563     564#
STACK   = 001100            2010
STARTZ  001306              2070    332
START1  000200              2270
START3  000210              2290
STKLMT= 177774              2010
STKPTR= 000500              2200    287     304     345     408     426     479     505
SWR     = 177570            2010    437     455     549     557     574     581
SW0     = 000001            2010
SW00    = 000001            2010
SW01    = 000002            2010
SW02    = 000004            2010
SW03    = 000010            2010
SW04    = 000020            2010
SW05    = 000040            2010
SW06    = 000100            2010
SW07    = 000200            2010
SW08    = 000400            2010
SW09    = 001000            2010
SW1     = 000002            2010
SW10    = 002000            2010
SW11    = 004000            2010
SW12    = 010000            2010
SW13    = 020000            2010
SW14    = 040000            2010
SW15    = 100000            2010
SW2     = 000004            2010
SW3     = 000010            2010
SW4     = 000020            2010
SW5     = 000040            2010
SW6     = 000100            2010
SW7     = 000200            2010
SW8     = 000400            2010
SW9     = 001000            2010
$28     002704              550     561     564#    571
TBITVE= 000014              2010
TEMP    001274              2020    372#    374#    375     388#
TKB     = 177562            2240    532
TKS     = 177560            2230    518#    538
TKVEC   = 000060            2010    508#    509#
TPB     = 177566            2220    465#    548#
TPS     = 177564            2210    466
TPVEC   = 000064            2010
TRAPVE= 000034              2010
TRTVEC= 000014              2010
```

```
TYP     001000#          215     232#
TYPE  = 104400#          233#    293     326     310     330     441     445     449     453     488     487     491     498
                         517     536     577     584     601
T1      00157#           341     345#
T1A     001604           348#    356     361     364
T1B     001664           357     366#
T2      001666           369#
T2A     001704           372#    379     395
T2B     001732           376     388#    384
T2C     001746           381     385#    388
T2D     001756           386     389#    392
T2E     001766           390     393#
T3      001776           400#
T3A     002020           404#    406
T3AA    002012           403#    420
T3B     002026           403     407#
T3C     002036           409#    411
T3D     002044           407     412#
T3E     002054           414#    416
T3F     002064           412     417#
T4      002104           426#
T4B     002124           438#    435
T4C     002130           432#    456     458
T4D     002140           431     436#
T4E     002244           433     438     459#
WORDS   001266           279#    321*    347     378     402     427     482
XOR     002720           554     569#
XORA    001356           289     299#
XORFLG  001354           292*    297#    301*    350     551
$FILLC  001114           232#
$FILLS  001113           232#
$NULL   001112           232#
$RDCHR* ****** U         233
$RDDEC* ****** U         233
$RDLIN* ****** U         233
$RDOCT* ****** U         233
$SAVRE* ****** U         233
$TPB    001110           232#*
$TPFLG  001115           232#
$TPS    001106           232#
$TRAP   001116           233#
$TRP  = 000002           233#
$TRPAD  001136           233#
$TYPBN* ****** U         233
$TYPDS* ****** U         233
$TYPE   001000           232#    233
$TYPOC* ****** U         233
.     = 005200           203#    207     209#    212#    217#    226#    228#    231#    232     467     513     531     575
                         502     628#
.HLT    002730           213     574#
.SCOPE  002606           210     549#
```

DZBMAE.P11 CROSS REFERENCE TABLE

```
COMMEN    2018#
ENDCOM    2018#
ERROR     2018#    359     377     382     387     391     405     410     415     436     520
ESCAPE    2018#
MULT      2018#
NEWTST    2018#
POP       2018#
PUSH      2018#
SCOPE     2018#    366     396     422     459
SETTRA    2338#
SETUP     2018#
SKIP      2018#
SLASH     2018#
SPACE     2018#
STARS     2018#    232     233
TRMTRP    2338#
TYPBIN    2018#
TYPDEC    2018#
TYPNUM    2018#
TYPOCS    2018#
TYPOCT    2018#
TYPTXT    2018#
$SESCA    2018#
$SNEWT    2018#
$SSET     2338#
$SSKIP    2018#
.EQUAT    2008#    201
.STRAP    2008#    233
.STYPE    2008#    232
```

DZBMAE.P11 CROSS REFERENCE TABLE

```
ADD       232   317   320   323   326   354   374   599
ASL       335   542   543   544
BEQ       376   381   386   390   433   463   469   494   575
BGT       561
BIC       388   533   541
BIS       545
BIT       437   549   557   574
BLT       232
BNE       232   356   395   428   431   438   496   535   550   558   611
BPL       232   351   456   467   531   552   582
BR        232   357   361   379   384   388   392   486   411   416   435   458   508   513   546
          571
CLC       314
CLR       301   305   340   372   414   507   529   562   596   603
CMP       299   358   363   385   408   413   418   430   434   462   538   560   569
CMPB      232   389   534
DEC       355   394   419   432   493   495   618
DECB      232
EMT       201
HALT      207   232   457   583
INC       409   461   559
IOT       201
JMP       227   229   295   302   332   336   364   464   474   501   522
JSR       232   308   312   440   444   448   452   470   486   490   514   588
MOV       232   233   287   288   289   290   291   292   300   304   316   318   319   321   322
          324   325   327   328   329   334   341   342   345   346   347   348   349   352   353
          369   370   371   373   400   401   402   403   404   407   412   421   426   427   428
          429   439   443   447   451   465   468   479   482   483   484   485   489   497   505
          506   508   509   510   511   516   540   553   554   555   556   563   564   570   579
          591   592   593   594   595   600   612   613   614
MOVB      232   233   532
NOP       471   472   473
ROL       315   597   598   604   605   606   607   608   609
RTI       232   519   565   576   586
RTS       232   233   539   615
SUB       375
TRAP      233
TST       232   233   350   370   383   393   417   455   512   551   581
TSTB      232   466   530
WAIT      521
.ASCIZ    618   619   620   621   622   623   624   625   626
.BYTE     232
.ENABL    192
.END      715
.ENDC     201   232   233
.EQUIV    201
.IF       201   232   233
.IFF      201   232   233
.IIF      232   233
.LIST     194   201   208   233
.MACRO    233
.MCALL    200   201
.NLIST    193   201   202   233
.PAGE     338   523   588
```

DZBMAE.P11 CROSS REFERENCE TABLE

```
.REPT        1     204
.SBTTL     201     232     233
.WORD      232     297     313     629



 ERRORS DETECTED:  0

 *,DZBMAE/CRF_DZBMAE
 RUN-TIME:  13  7  1 SECONDS
 CORE USED:  18K
```