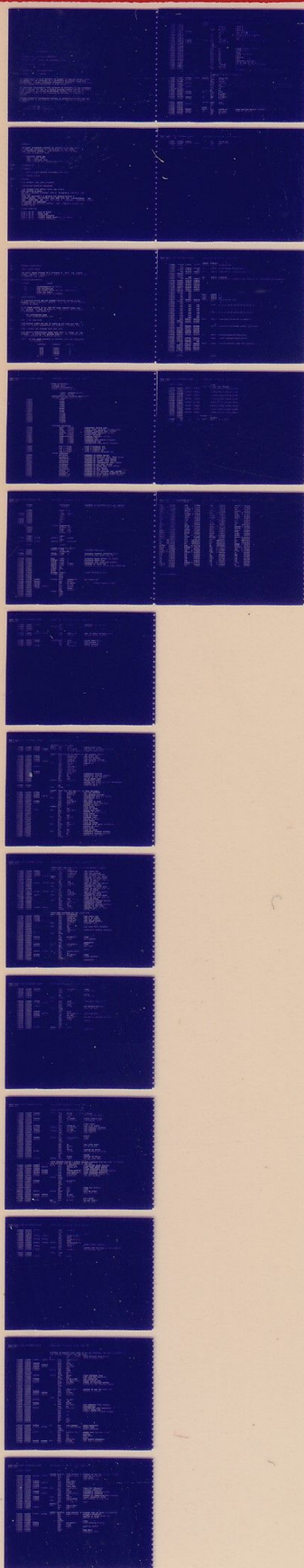


# VT20

BOOTSTRAP LOADER  
MD-11-DZBMK-B

EP-DZBMK-B-DL  
COPYRIGHT '73-74  
FICHE 1 OF 1

MAY 1978  
**digital**  
MADE IN USA





IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZBMK-R-D

PRODUCT NAME: M792YK VT20 BOOTSTRAP LOADER

DATE CREATED: JANUARY 11, 1974

MAINTAINERS: DIAGNOSTIC GROUP

AUTHORS: ED BADGER

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

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

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

COPYRIGHT © 1973, 1974  
DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

THE DZBMK DIAGNOSTIC PROGRAM IS WRITTEN TO BE USED AS AN AID TO HARDWARE DEBUGGING AND MAINTENANCE OF THE M792YK VI20 BOOTSTRAP LOADER). 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/WITH VT20
- B. M792-YK 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 SWR = DESIRED STANDARD PDP-11 DIAGNOSTIC OPTIONS (SEE SECT 4.0)

NOTE: ALL SWITCHES = 0 SELECTS AND STARTS PROGRAM 0

DEPRESS START. THE PROGRAM WILL THEN TYPE OUT INSTRUCTIONS. ALL USER RESPONSES ARE VIA THE "TUBE 0" KEYBOARD (CARRIAGE RETURN TERMINATES THE RESPONSE)

TO RESTART THE SELECTED PROGRAM LOAD ADDRESS = 000210 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 PRG0 - LOGIC TESTS

THE LOGIC TESTS CONSIST OF 4 ROUTINES TO TEST THE M792YK LOGIC. PROGRAM 0 LOOPS WITHIN ITSELF UNTIL A NEW PROGRAM NUMBER IS PROVIDED I.E. PRG# = ?

5.1.1 ROUTINE DESCRIPTIONS

ROUTINE	TESTS
T1	ADDRESSABILITY OF M792YK
T2	DATA RELIABILITY
T3	THAT M792YK 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 (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.

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) FOLLOWED BY A CARRIAGE RETURN.

NOTE: THE ROM WORDS STARTING AT LOCATION 4000 ARE DESIGNATED AS FOLLOWS -

LOCATION	CONTENTS	WORD NO.
4000	012701	0
4002	160000	2
4004	012702	4
4006	000006	6
4010	012703	10

ETC.



!LOAD ADDRESS=0200  
!DEPRESS START  
!RESTART ADDRESS=0210  
!STACK POINTER IS AT 500

.MCALL SCPREGS,SCPVECS  
.MCALL SCMTAG  
!DEFINITIONS AND REGISTER ASSIGNMENTS  
!GENERAL REGISTER ASSIGNMENTS

000000  
000001  
000002  
000003  
000004  
000005  
000006  
000007  
000000  
000001  
000002  
000003  
000004  
000005

R0=80  
R1=81  
R2=82  
R3=83  
R4=84  
R5=85  
SP=86  
PC=87  
R10=80  
R11=81  
R12=82  
R13=83  
R14=84  
R15=85

!REGISTER ADDRESSES

177776  
177774  
177772  
177770  
177570  
177570  
177560  
177562  
177564  
177566

PSW= 177776  
SLR= 177774  
PIRQ= 177772  
UBREAK= 177770  
SR= 177570  
DISPLAY=177570  
TKS= 177560  
TKB= 177562  
TPS= 177564  
TPB= 177566

!PROCESSOR STATUS WORD  
!STACK LIMIT REGISTER (11/40,11/45)  
!PROGRAM INTERRUPT REQ. (11/45)  
!MICRO-BREAK REGISTER (11/45)  
!SWITCH REGISTER  
!DISPLAY REGISTER (11/45)  
!KEYBOARD CSR  
!KEYBOARD DATA BUFFER REGISTER  
!TELEPRINTER CSP  
!TELEPRINTER DATA BUFFER REGISTER

170334  
170336  
170304  
170302

FKS = 170334  
FKB = 170336  
VTC = 170304  
VIS = 170302

!TUBE 0 KEYBOARD CSR  
!TUBE 0 KEYBOARD DATA  
!TUBE 0 DISPLAY CSR  
!TUBE 0 DISPLAY STARTING ADR.

!VECTOR ADDRESSES

000004  
000010  
000014  
000014  
000014  
000020  
000024  
000030  
000034  
000060  
000064  
000114  
000240

ERRVEC=4  
RESVEC=10  
TBITVEC=14  
TRTVEC=14  
BPTVEC=14  
IOTVEC=20  
PFVEC=24  
EMTVEC=30  
TRAPVEC=34  
TKVEC= 60  
TPVEC=64  
PARVEC= 114  
PIRVEC=240

!ADDRESS OF ERROR VECTOR  
!ADDRESS OF RESERVED INST. TRAP VECTOR  
!ADDRESS OF 'T' BIT TRAP VECTOR  
!ADDRESS OF 'TRACE' TRAP VECTOR  
!ADDRESS OF 'BREAKPOINT' TRAP VECTOR  
!ADDRESS OF IOT TRAP VECTOR  
!ADDRESS OF POWER FAIL TRAP VECTOR  
!ADDRESS OF EMT VECTOR  
!ADDRESS OF TRAP VECTOR  
!ADDRESS OF TTY KEYBOARD INT. VECTOR  
!ADDRESS OF TTY PRINTER INTERRUPT VECTOR  
!ADDRESS OF MA/MF PARITY ERROR VECTOR  
!ADDRESS OF PIRQ VECTOR

	000244		FPEVFC=244		;ADDRESS OF FLOATING POINT INT. VECTOR
	000250		MMVEC=250		;ADDRESS OF MEM MGMT ERROR TRAP VECTOR
	000320		FKVEC = 320		
	000020		.=20		
000020	003234		.WORD VTOUT		
000022	000240		.WORD 240		
	000030		.=30		
000030	002764		.SCOPE		
000032	000340		340		
000034	003074		.HLT		
000036	000000		0		
			;EQUATE STATEMENTS		
	000500		STKPTR= 500		
	000004		TYPE= IOT		
	104400		HLT= TRAP		
	104000		SCOPE= EMT		
	000200		.=200		
000200	000167	001132	START1: JMP PRMTRS		
	000210		.=210		
000210	000167	001210	START3: JMP RESTART		
	001300		.=1300		
			;WORDS LOADED BY 'SCOPE'		
001300	000000		SCPBLK: .WORD 0		;CONTAINS PASS COUNT
	001300		ICNT=SCPBLK		
001302	000002		ITCNT: .WORD 2		;CONTAINS SUBTEST ITERATION COUNT
001304	000004		LASTPC: .WORD 4		;CONTAINS LAST SCOPE CALL PC
001306	000000		EPC: .WORD 0		;CONTAINS SCOPE RETURN FOR ERRORS
001310	000000		ERRFLG: .WORD 0		;CONTAINS ERPOP FLAG
001312	000000		TICKS: .WORD 0		;CONTAINS TICK COUNT FOR CLOCKS
001314	177560		INCSR: .WORD 177560		;ADDRESS OF INPUT CSR
001316	177562		INDAT: .WORD 177562		;ADDRESS OF INPUT DEVICE DATA BUFFER REG
001320	000040		WORDS: 32		
001322	004000		IMAGE: 4000		
001324	000000		TEMP: 0		
001326	173000		ROMADD: 173000		;FIRST ADDRESS OF DATA
001330	001442		PRGTAB: PRG0		
001332	002202		PRG1		
001334	002322		PRG2		
001336	012706	000500	PRMTRS: MOV #STKPTR,#6		;SET STACK PTR
001342	005067	000040	CLR PRGNUM		
001346	004767	001750	JSR #7,RESBF		;OVERLAY PRINTOUT AREA
001352	010467	166724	MOV R4,VTS		
001356	012767	000001	MOV #1,VTC		
001364	005737	177570	TST @#SWR		
001370	001415		BEG RESTART		
001372	000004		FIXED: TYPE		
001374	003553		MHED		
001376	000004		TYPE		
001400	003365		M6		

001402	004567	001120	JSP	5,RECD	IRECEIVE DATA AND PUT	
001406	000000		PRGNUM: 0		IIT HEPE	
001410	000004		TYPE			
001412	003414		MB			
001414	026727	177766	000002	CMP	PRGNUM,02	ISEE IF LEGAL PROGRAM NUMBER
001422	003345			BGT	PRMPS	IIF NOT RETYPE QUESTION
001424	005067	000550	PESTART:CLP	PCNT		ICLEAR PASS COUNT
001430	016700	177752	MOV	PRGNUM,00		IGET PROGRAM *
001434	006300		ASL	00		ISHIFT PROGRAM *
001436	000170	001330	JMP	@PFGTAB(0)		IGO TO PROGRAM



```

;PROGRAM 0 LOGIC TESTS
001442 005067 177632 PRG0: CLR ICNT ;CLEAR PASS COUNT
001446 012767 001462 177630 PRG0: MOV #T1,LASTPC ;SET RETURN ADDRESS FOR SCOPE
001454 016737 177620 177570 MOV ICNT,#DISPLAY ;DISPLAY PASS COUNT

;TEST1 TEST ABILITY TO REFERENCE ROM WITHOUT TIMING OUT
001462 012706 000500 T1: MOV #STKPTR,SP ;SET STACK PTR
001466 016700 177634 MOV ROMADD,#0 ;GET ROM ADDRESS
001472 016701 177622 MOV WORDS,#1 ;GET ADDRESS COUNTER
001476 012767 001536 176300 MOV #ERROR1,4 ;SET UP TIME OUT VECTOR
001504 011003 T1A: MOV (0),#3 ;REFERENCE
001506 005720 TST (0)+ ;FROM
001510 064067 177610 ADD #(0),TEMP ;
001514 021010 CMP (0),(0) ;
001516 132020 BITH (0)+,(0)+ ;
001520 164067 177600 SUB #(0),TEMP ;
001524 062700 000002 ADD #2,#0 ;INCREMENT POINTER
001530 005301 DEC #1 ;DECREMENT ADDRESS COUNTER
001532 001364 BNE T1A ;BRANCH IF NOT FINISHED
001534 000403 BR T1B ;GO TO SCOPE LOOP
001536 022626 ERROR1: CMP (6)+,(6)+ ;REPOSITION STACK
001540 104400 HLT ;ERROR! ROM TIMED OUT WHEN REFERENCED
;ADDRESS IS IN R0
;LOOP ON ERROR

001542 000760 BR T1A
001544 104000 T1B: SCOPE

;TEST2 TEST THAT ROM DATA CAN BE READ RELIABLY.
001546 016700 177554 T2: MOV ROMADD,#0 ;GET ROM ADDRESS
001552 016701 177542 MOV WORDS,#1 ;GET ADDRESS COUNTER
001556 012767 000006 176220 MOV #6,4 ;INITIALIZE TIME OUT VECTOR
001564 005067 177534 T2A: CLR TEMP ;INITIALIZE TEMP
001570 011003 MOV (0),#3 ;GET DATA
001572 062067 177526 ADD (0)+,TEMP ;ADD DATA TO TEMP
001576 166703 177522 SUB TEMP,#3 ;SUBTRACT DATA FROM DATA
001602 001402 BEQ T2B ;BRANCH IF EQUAL
001604 104400 ERROR2: HLT ;DATA ERROR
001606 000766 BR T2A ;LOOP ON ERROR
001610 044067 177510 T2B: BIC #(0),TEMP ;CLEAR TEMP BITS
001614 001402 BEQ T2C ;BRANCH IF EQUAL TO 0
001616 104400 HLT ;DATA ERROR
001620 000773 BR T2B ;LOOP ON ERROR
001622 021010 T2C: CMP (0),(0) ;COMPARE DATA
001624 001402 BEQ T2D ;BRANCH IF EQUAL
001626 104400 HLT ;DATA ERROR
001630 000774 BR T2C ;LOOP ON ERROR
001632 122040 T2D: CMPB (0)+,-(0) ;COMPARE DATA (BYTE OPERATION)
001634 001402 BEQ T2E ;BRANCH IF EQUAL
001636 104400 HLT ;DATA ERROR
001640 000774 BR T2D ;LOOP ON ERROR
001642 005720 T2E: TST (0)+ ;INCREMENT ADDRESS POINTER
001644 005301 DEC #1 ;DECREMENT ADDRESS COUNTER
001646 001364 BNE T2A ;RETURN IF NOT DONE
001650 104000 SCOPE

```



```

;TEST3 TEST THAT ROM TIMES OUT IF REFERENCED BY OTHER
;THAN DATA BUS CYCLE
001652 012706 000500 T3:   MOV    %STKPTR,%6    ;SET STACK PTR
001656 016700 177444      MOV    ROMADD,%0     ;GET ROM ADDRESS
001662 016701 177432      MOV    WORDS,%1      ;GET ADDRESS COUNTER
001666 012767 001702 176110 T3AA: MOV    %T3B,4       ;SET UP TIME OUT VECTOR
001674 010010      T3A:   MOV    %0,(0)    ;ATTEMPT TO ALTER DATA
001676 104400      HLT                    ;HERE IF DID NOT TIME OUT
001700 000775      BR     T3A            ;LOOP ON ERROR
001702 012767 001720 176074 T3B:   MOV    %T3D,4       ;SET UP TIME OUT VECTOR
001710 022626      CMP    (6)+,(6)+    ;REPOSITION STACK
001712 005210      T3C:   INC    (0)      ;ATTEMPT TO ALTER DATA
001714 104400      HLT                    ;HERE IF DID NOT TIME OUT
001716 000775      BR     T3C            ;LOOP ON ERROR
001720 012767 001740 176056 T3D:   MOV    %T3F,4       ;SET UP TIME OUT VECTOR
001726 022626      CMP    (6)+,(6)+    ;REPOSITION STACK
001730 005077 177372      T3E:   CLR    %ROMADD    ;ATTEMPT TO ALTER DATA
001734 104400      HLT                    ;HERE IF DID NOT TIME OUT
001736 000774      BR     T3E            ;LOOP ON ERROR
001740 005720      T3F:   TST    (0)+      ;INCREMENT ADDRESS POINTER
001742 022626      CMP    (6)+,(6)+    ;REPOSITION STACK
001744 005301      DEC    %1            ;DECREMENT ADDRESS COUNTER
001746 001347      BNE    T3AA          ;RETURN IF NOT DONE
001750 012737 000006 000004      MOV    %6,%#4       ;RESTORE TIME OUT TRAP
001756 104000      SCOPE                ;SCOPE LOOP

```

```

;THIS TEST COMPARES ROM AND IMAGE DATA
;AND TYPES OUT DIFFERENCES
001760 012706 000500 T4:   MOV    %STKPTR,%6    ;SET STACK PTR
001764 016701 177330      MOV    WORDS,%1      ;GET # OF WORDS
001770 016700 177332      MOV    ROMADD,%0     ;GET ROM ADDRESS
001774 016703 177322      MOV    IMAGE,%3      ;GET IMAGE ADDRESS
002000 021013      T4B:   CMP    (0),(3)     ;COMPARE DATA
002002 001004      BNE    T4D            ;ALL DATA BEEN COMPARED
002004 005301      T4C:   DEC    %1            ;INCREMENT ADDRESS POINTERS
002006 001441      BEQ    T4E
002010 022023      CMP    (0)+,(3)+
002012 000772      BR     T4B
002014 000004      T4D:   TYPE    M8
002016 003414      M8
002020 010067 001110      MOV    %0,D2BTYP    ;TYPE
002024 004767 001106      JSR    7,02A        ;ROM ADDRESS
002030 000004      TYPE    M10
002032 003501      M10
002034 011067 001074      MOV    (0),D2BTYP   ;TYPE
002040 004767 001072      JSR    7,02A        ;ROM DATA
002044 000004      TYPE    M8
002046 003414      M8
002050 032737 040000 177570 18:  BIT    %40000,%#SWR ;CR/LF
002056 001374      BNE    18
002060 010367 001050      MOV    %3,D2BTYP    ;TYPE
002064 004767 001046      JSR    7,02A        ;IMAGE ADDRESS
002070 000004      TYPE    M12
002072 003505      M12
;SEPARATOR

```

002074	011367	001034		MOV	(3),D2BTYP	I TYPE
002100	004767	001032		JSP	7,02A	I IMAGE DATA
002104	000004			TYPE		
002106	003414			MB		I CR/LF
002110	000735			RR	T4C	I GO TO T4C
002112	104000		T4E:	SCCPE		
002114	005267	177160		END:	INC	I INCREMENT PASS COUNT
002120	026727	177154	000100	CMP	ICNT,#100	
002126	001402			BEG	DONE	
002130	000167	177312		JMP	PRGOR	I GO RESTART PROGRAM
002134	000004		DONE:	TYPE		I RING THE BELL
002136	003507			MBELL		
002140	005267	000034		INC	PCNT	
002144	016767	000030	000762	MOV	PCNT,D2BTYP	I GET PASS COUNT
002152	004767	000760		JSR	87,02A	
002156	013700	000042		MOV	8842,80	I RETURN TO DECTAPE MONIIO?
002162	001404			REQ	DONE1	
002164	004710			JSR	7,(0)	I RETURN!
002166	000240			NOP		
002170	000240			NOP		
002172	000240			NOP		
002174	000167	177242	DONE1:	JMP	PRGO	
002200	000000		PCNT:	0		I PASS COUNT



;THIS PROGRAM TYPES OUT ROM DATA

002202 005067 000234  
002206 005267 000230  
002212 001375  
002214 004767 001102  
002220 012706 000500  
002224 000004  
002226 003377  
002230 016701 177064  
002234 016700 177066  
002240 012702 000011  
002244 010067 000664  
002250 004767 000662  
002254 000004  
002256 003414  
002260 012067 000650  
002264 004767 000646  
002270 000004  
002272 003503  
002274 005301  
002276 001407  
002300 005302  
002302 001366  
002304 012702 000011  
002310 000004  
002312 003414  
002314 000753  
002316 000167 177050

```

PRG1:  CLR      PRG2H  ;INITATE A DELAY
        INC      PRG2B  ;AND EXECUTE DELAY
        BNE     .-4
        JSP     %7,RESBF ;RESET BUFFER AREA
        MOV     %STKPTR,%6 ;INITIALIZE STACK
        TYPE    M7
        MOV     WORDS,%1 ;"ROM DATA"
        MOV     ROMADD,%0 ;GET # OF WORDS
PRG1A:  MOV     %11,%2 ;GET STARTING ADDRESS
        MOV     %0,D2BTYP ;GET ADDRESS INDICATOR
PRG1B:  JSP     7,02A ;GET ADDRESS
        TYPE    M6 ;AND TYPE IT
        MOV     (%0)+,D2BTYP ;CR/LF
PRG1C:  JSP     7,02A ;TYPE
        TYPE    M11 ;DATA
        DEC     %1 ;ALL DATA TYPED,
        BEG     PRG1D ;GO TO FINISH
        DEC     %2
        BNE     PRG1C ;RETURN TO PRG1B
        MOV     %11,%2 ;GET ADDRESS INDICATOR
        TYPE    M8 ;CR/LF
        BR      PRG1B ;RETURN TO PRG1B
PRG1D:  JMP     FIXED ;GO GET NEXT TEST
    
```

;THIS PROGRAM CYCLES A SINGLE ADDRESS (ADDRESS MUST BE EVEN) TO CHANGE  
;THE ADDRESS TYPE NEW ADDRESS ON THE TTY.

002322 012706 000500  
002326 012737 002510 000004  
002334 005067 175436  
002340 012737 002412 000320  
002346 012737 000340 000322  
002354 052737 000101 170334  
002362 016700 176740  
002366 000004  
002370 003663  
002372 010067 000536  
002376 004767 000534  
002402 000004  
002404 003414  
002406 005710  
002410 000776  
002412 105767 165716  
002416 100407  
002420 052767 000002 165706  
002426 052767 000101 165700  
002434 000002  
002436 004567 000064  
002442 000000

```

PRG2:  MOV     %STKPTR,%6 ;INITIALIZE STACK POINTER
        MOV     %PRG2C,%4 ;LOAD TRAP ERROR VECTOR
        CLR     PSW ;CLEAR PROCESSOR STATUS
        MOV     %PRG2A,%0FKVEC ;LOAD KEYBOARD INTERRUPT VECTOR
        MOV     %340,%0FKVEC+2 ;LOAD KEYBOARD PRIORITY
        BIS     %101,%0FKS ;SET INTERRUPT ENABLE BIT
        MOV     ROMADD,%0 ;GET ROM ADDRESS
PRG20:  TYPE    MCYC
        MOV     %0,D2BTYP
        JSP     7,02A
        TYPE    M8
        TST     (%0) ;READ ROM ADDRESS
        BR      .-2 ;LOOP
PRG2A:  TSTR    FKS ;DID KB INTR?
        BMI     46 ;YES, GET INFO
        BIS     %2,FKS
        BIS     %101,FKS
        RTI
46:    JSP     %5,RECD ;NO - EXIT
PRG2B:  0 ;GO GET ADDR &
        ;PUT IT HERE
    
```

002444	016700	177772		MOV	PRG2B,80	
002450	000004			TYPE		
002452	003414			MB		ICR/LF
002454	026727	177762	173000	CMP	PRG2B,#173000	
002462	002415			BLT	PRER	
002464	026727	177752	173100	CMP	PRG2B,#173100	
002472	002011			BGE	PRER	
002474	012737	002412	000320	PRERN: MOV	#PRG2A,#FKVEC	
002502	022626			CMP	(6)+,(6)+	IRESET STACK POINTER
002504	000167	177656		JMP	PRG20	
002510	104400			PRG2C: HLT		ERROR! DID YOU TYPE AN ODD ADDRESS?
002512	000167	177726		JMP	PRG2B+2	TRY ADDRESS AGAIN
002516	000004			PRER: TYPE		
002520	003632			WA		
002522	000167	177746		JMP	PRERN	



```

;ROUTINE TO RECEIVE DATA TYPED IN ON THE KEYBOARD, THE DATA IS PLACED IN
;THE ADDRESS FOLLOWING THE JSR CALL:
;
;          JSP      5,RECD  ;CALL RECEIVE DATA ROUTINE
;          0          ;DATA IS PLACED HERE
002526 012767 002572 175564 RECD:  MOV      #RECD1,FKVEC
002534 005015                CLP      (5)
002536 005067 000104                CLR      DONF
002542 052767 000002 165564                BIS      #2,FKS
002550 052767 000101 165556                BIS      #101,FKS
002556 005067 175214                CLR      PSW
002562 005767 000060 RECD1:  TST      DONF
002566 001775                BEQ      RECD1
002570 000205                RTS
002572 105737 170334 RECD1:  TSTB   #FKS                ;TEST KEYBOARD FLAG
002576 100063                BPL      REX                ;AND WAIT FOR CHARACTER
002600 113746 170336                MOVB   #FKB,-(SP)          ;GET CHARACTER
002604 042716 177700                BIC   #177700,(SP)        ;STRIP PARITY BIT
002610 122716 000015                CMPB  #15,(SP)            ;CHECK IF CARRIAGE RETURN
002614 001015                BNE   DONF+2             ;BRANCH IF NOT CARRIAGE RETURN
002616 000004                TYPE
002620 003414                MB
002622 022526                CMP   (R5)+,(SP)+        ;ADJUST R5 AND THE STACK PTR
002624 005267 000016                INC   DONF              ;RETURN TO CALLER
002630 052767 000002 165476                BIS   #2,FKS
002636 052767 000101 165470                BIS   #101,FKS
002644 000002                RTI
002646 000000                DONF:  0
002650 022716 000003                CMP   #3,(SP)
002654 001002                BNE   ,+6
002656 000137 000200                JMP   #200
002662 022716 000077                CMP   #77,(SP)          ;DID OPERATOR TYPE RUBOUT?
002666 001012                BNE   28                ;NO-CONTINUE
002670 112744 000040                MOVB  #40,-(4)          ;YES-REPLACE LAST CHARACTER
002674 000241                CLC
002676 006015                ROR   (5)              ;WITH A SPACE AND
002700 000241                CLC
002702 006015                ROP   (5)              ;RIGHT JUSTIFY CURRENT NUMBER
002704 000241                CLC
002706 006015                ROP   (5)
002710 005726                TST  (6)+
002712 000415                BR   REX

002714 011667 000214 28:  MOV   (SP),D2BTYP        ;ECHO CHARACTER
002720 042767 177400 000206 BIC   #177400,D2BTYP    ;STRIP WORD
002726 000004                TYPE
002730 003134                D2BTYP                ;ECHO CHARACTER
002732 042716 177770 38:  BIC   #177770,(SP)      ;STRIP AWAY ALL BUT 3 LSB
002736 006315                ASL   (5)              ;ROTATE
002740 006315                ASL   (5)              ;PREVIOUS
002742 006315                ASL   (5)              ;DATA
002744 052615                BIS   (SP)+,(5)        ;AND INSERT CHARACTER
002746 052767 000002 165360 REX:  BIS   #2,FKS            ;GET NEXT CHARACTER
002754 052767 000101 165352                BIS   #101,FKS

```

```

002762 000002                                RTI

;SCOPE ROUTINE. THIS ROUTINE IS ENTERED AT THE END OF EACH SUBTEST.
002764 032737 040000 177570 .SCOPE: BIT    $40000,$$SWP    ;TEST SR FOR SCOPE
002772 001030                                BNE    A28      ;YES SCOPE
002774 000411                                BR     A38      ;NOP IF FOR XOR
002776 013746 000004                                MOV    $04,-($6)
003002 012737 003064 000004                                MOV    $XOR,$04
003010 005737 177060                                TST   $0177060
003014 012637 000004                                MOV    ($6)+,$04
003020 032737 004000 177570 A38: BIT    $4000,$$SWP    ;TEST FOR ITERATION
003026 001006                                BNE    A18      ;INHIBIT ITERATION
003030 005267 176246                                INC    ITCNT    ;INCREMENT ITERATION COUNT
003034 026767 176242 000020                                CMP    ITCNT,ICOUNT ;ITERATION COMPLETE
003042 003004                                BGT    A28      ;BRANCH IF ITERATIONS NOT COMPLETE
003044 005067 176232                                A18: CLP    ITCNT ;CLEAR ITERATION COUNT
003050 011667 176230                                MOV    (SP),LASTPC ;GET ADDRESS OF NEXT TEST
003054 016716 176224                                A28: MOV    LASTPC,(SP)
003060 000002                                RTI           ;EXIT
003062 000005                                ICOUNT: S
003064 012626                                XOR:  MOV    ($6)+,($6)+
003066 012637 000004                                MOV    ($6)+,$04
003072 000770                                BR     A28

;ERROR ROUTINE. THIS ROUTINE IS ENTERED WHEN AN ERROR IS DETECTED.
003074 033727 177570 020000 .HLT: BIT    $$SWR,$20000 ;INHIBIT PRINTOUT?
003102 001401                                BEQ    .+4     ;BRANCH IF ERROR PRINT OUT
003104 000002                                RTI           ;RETURN TO TEST
003106 000004                                TYPE
003110 003356                                ERRORM
003112 011667 000016                                MOV    (6),D2BTYP ;*PC=
003116 004767 000014                                JSP    7,02A   ;TYPE PROGRAM COUNTER
003122 005737 177570                                TST   $$SWR   ;HALT ON ERROR?
003126 100001                                BPL    .+4    ;
003130 000000                                HALT
003132 000002                                RTI           ;YES HALT
;RETURN TO TEST

```



```

003134 000000
003136 010246
003140 010146
003142 010046
003144 016700 177764
003150 012701 000006
003154 005002
003156 006100
003160 006102
003162 062702 000260
003166 042702 177700
003172 010267 177736
003176 000004
003200 003134
003202 005002
003204 006100
003206 006102
003210 006100
003212 006102
003214 006100
003216 006102
003220 005301
003222 001357
003224 012600
003226 012601
003230 012602
003232 000207

```

```

;THIS ROUTINE CONVERTS AN OCTAL NUMBER TO ASCII AND TYPES IT ON THE TTY.
D2BTYP: 0
O2A:  MOV  R2,=(6)      ;SAVE R2
      MOV  R1,=(6)      ;SAVE R1
      MOV  R0,=(6)      ;SAVE R0
      MOV  D2BTYP,R0    ;GET DATA TO BE TYPED
      MOV  R6,R1        ;GET COUNTER
      CLP  R2           ;CLEAR WORKING REGISTER
      ROL  R0           ;MOV FIRST BIT (MSB) INTO
      ROL  R2           ;R2
18:   ADD  R2,R0,R2      ;FORM ASCII CODE+STOP CODE (31)
      BIC  R2,R2,R2     ;177700,R2
      MOV  R2,D2BTYP
      TYPE D2BTYP
      CLP  R2           ;CLEAR WORKING REGISTER
      ROL  R0           ;ROTATE THE
      ROL  R2           ;NEXT
      ROL  R0           ;OCTAL CHARACTER
      ROL  R2           ;INTO
      ROL  R0           ;REGISTER
      DEC  R6           ;TWO
      BNE 18           ;DECREMENT COUNTER
      MOV  R0,R0,R0    ;GO TO O2AA IF NOT 0
      MOV  R0,R0,R0    ;FINISHED. RESTORE REGISTERS
      MOV  R0,R0,R0
      RTS  7           ;AND EXIT

```

```

003234 017667 000000 000056
003242 117724 000052
003246 001420
003250 121427 000012
003254 001004
003256 005267 000072
003262 100401
003264 000405
003266 005267 000026
003272 020427 005022
003276 002761
003300 004767 000016
003304 000167 177724

003310 105744
003312 062716 000002
003316 000002
003320 000000
003322 012704 004100
003326 112724 000031
003332 020427 005023
003336 001373

```

```

;THIS ROUTINE SENDS MESSAGES TO
;TUBE 0
VTOUT: MOV  R0,(6),OUTT
VTOUT1: MOVH R0,OUTT,(4)+
      BEQ  VTOUTE
      CMPE (4),R12
      BNE 18
      INC  RETCNT
      BMI 18
      BR  28
18:   INC  OUTT
      CMP  R4,R4,R4    ;R4,R4E
      BLT  VTOUT1
28:   JSP  R7,RESBF
      JMP  VTOUT

VTOUTE: TSTB =(4)
      ADD  R2,R2,(6)
      RTI

OUTT:  0
RESBF: MOV  R4,R4,R4    ;R4,R4E+1
18:   MOVH R31,(4)+
      CMP  R4,R4,R4    ;R4,R4E+1
      BNE 18

;THIS ROUTINE RESETS PRINTOUT
;AREA TO ALL EOS

```

TEST DZMK VT20 BOOTSTRAP LOADER  
DZMK,SPC

MACY11,624 21-JAN-74 10:05 PAGE 2-5

003340 012704 004100  
003344 012767 177762 000002  
003352 000207

MOV #PRUFF,R4  
MOV #-14.,RETCNT  
RIS 87

003354 177762

RETCNT: -14.



```

003356 005015 041520 020075  ;ASCII MESSAGES
003364      000  ERROR: .ASCIZ <15><12>'PC= '
003365      015 173012 051120 M6:  .ASCIZ <15><12><366>'PRG=#'<360>
003372 021507 170075      000 M7:  .ASCIZ <15><12>'ROM DATA'<15><12>
003377      015 051012 046517 M8:  .ASCIZ <15><12>
003404 042040 052101 006501 M9:  .ASCIZ <15><12>'ROM ADDRESS/IMAGE ADDRESS ROM DATA+IMAGE DATA'<15><12>
003412 000012
003414 005015      000
003417      015 051012 046517
003424 040440 042104 042522
003432 051523 044457 040515
003440 042507 040440 042104
003446 042522 051523 051040
003454 046517 042040 052101
003462 025101 046511 043501
003470 020105 040504 040524
003476 005015      000
003501      057      000 M10: .ASCIZ '//'
003503      040      000 M11: .ASCIZ ' '
003505      052      000 M12: .ASCIZ ' '
003507      015 042412 042116 MBELLI .ASCII <15><12>/END PASS /
003514 050040 051501 020123
003522      040
003523      040      040      040 .BYTE 40,40,40,40,40,40,40,40,40,40,40,40
003526      040      040      040
003531      040      040      040
003534      040      040      040
003537      040      040      040 .BYTE 40,40,40,40,40,40,40,40,40,15,12,0
003542      040      040      040
003545      040      040      040
003550      015      012      000
003553      015 005012 046764 MHED: .ASCII <15><12><12><364>/MD-11-DZBMK-B /
003560 026504 030461 042055
003566 041132 045515 041055
003574      040
003575      374 052126 030062 .ASCIZ <374>/VT20 BOOTSTRAP LOADER TEST/<360>
003602 041040 047517 051524
003610 051124 050101 046040
003616 040517 042504 020122
003624 042524 052123 000360
003632 005015 047362 052117 WAI .ASCII <15><12><362>/NOT/<40><374>
003640 176040
003642 020101 047522 020115 .ASCIZ /A ROM ADDRESS/<360><12><15>
003650 042101 051104 051505
003656 170123 006412      000
003663      015 051012 040505 MCYCI .ASCIZ <15><12>/READING DATA FROM ADDRESS: /
003670 044504 043516 042040
003676 052101 020101 051106
003704 046517 040440 042104
003712 042522 051523 020072
003720      000
003776 .#3776
```

003776	000000				
004000	012701	160000	012702		
004006	000006				
004010	012703	173100	005012		
004016	010742				
004020	110706	014304	005714		
004026	100775				
004030	010712	012706	000024		
004036	010441				
004040	040601	010111	011102		
004046	005214				
004050	105714	100376	116412		
004056	000002				
004060	005211	120227	000375		
004066	001366				
004070	105222	000142	177560		
004076	175610				
004100	000000				
	005022				
005022	031	031	031		
005025	031				
	000001				

.WORD  
DATA CUT INTO THE M792-YK  
012701,160000,012702,000006  
012703,173100,005012,010742  
110706,014304,005714,100775  
010712,012706,000024,010441  
040601,010111,011102,005214  
105714,100376,116412,000002  
005211,120227,000375,001366  
105222,000142,177560,175610  
PBUFF: 0  
.M,+720  
PBE: .BYTE 31,31,31,31  
  
.END



AIS	003044	A28	003054	A38	003020	RPTVEC	= 000014
DISPLA	= 177570	DONE	002134	DONE1	002174	DONF	002646
D2RIYF	003134	EMTVEC	= 000030	END	002114	EPC	001306
ERRFIG	001310	ERRORM	003356	ERROR1	001536	ERROR2	001604
ERRVEC	= 000004	FIXED	001372	FKR	= 170336	FKS	= 170334
FKVEC	= 000320	FPEVEC	= 000244	HLT	= 104400	ICNT	= 001300
ICOUNT	003062	IMAGE	001322	INCSR	001314	INDAT	001316
IOTVEC	= 000020	ITCNT	001302	LASTPC	001304	MBELL	003507
MCYC	003663	MHED	003553	MMVEC	= 000250	M10	003501
M11	003503	M12	003505	M6	003365	M7	003377
M8	003414	M9	003417	QUIT	003320	O2A	003136
PARVEC	= 000114	PBE	005022	PBUFF	004100	PC	= 0000007
PCNT	002200	PFVEC	= 000024	PIPG	= 177772	PIRVEC	= 000240
PPER	002516	PRERN	002474	PRGNUM	001406	PRGTAB	001330
PRG0	001442	PRGOR	001446	PRG1	002202	PRG1A	002234
PRG1B	002244	PRG1C	002260	PRG1D	002316	PRG2	002322
PRG2A	002412	PRG2B	002442	PRG2C	002510	PRG20	002366
PRMTRS	001336	PSW	= 177776	RECD	002526	RECDI	002572
RECD1	002562	RESBF	003322	RESTAR	001424	RESVEC	= 000010
RETCNT	003354	REX	002746	ROMADD	001326	R0	= 0000000
R1	= 0000001	R10	= 0000000	R11	= 0000001	R12	= 0000002
R13	= 0000003	R14	= 0000004	R15	= 0000005	R2	= 0000002
R3	= 0000003	R4	= 0000004	R5	= 0000005	SCOPE	= 104000
SCPBLK	001300	SLR	= 177774	SP	= 0000006	START1	000200
START3	000210	STKPTR	= 000500	SWR	= 177570	TBITVE	= 000014
TEMP	001324	TICKS	001312	TKB	= 177562	TKS	= 177560
TKVEC	= 000060	TPB	= 177566	TPS	= 177564	TPVEC	= 000064
TRAPVE	= 000034	TRIVEC	= 000014	TYPE	= 000004	T1	001462
T1A	001504	T1B	001544	T2	001546	T2A	001564
T2B	001610	T2C	001622	T2D	001632	T2E	001642
T3	001652	T3A	001674	T3AA	001666	T3B	001702
T3C	001712	T3D	001720	T3E	001730	T3F	001740
T4	001760	T4B	002000	T4C	002004	T4D	002014
T4E	002112	UBREAK	= 177770	VIC	= 170304	VTOUT	003234
VTOUTE	003310	VTOUT1	003242	VIS	= 170302	WA	003632
WORDS	001320	XOR	003064	.HLT	003074	.SCOPE	002764
.	= 005026						

ERRORS DETECTED: 0