

BM792YH

CASSETTE BOOTSTRAP LOADER
MD-11-DZBMH-A

EP-DZBMH-A-DL
COPYRIGHT 1973
FICHE 1 OF 1

MAY 1978
digital
MADE IN USA

The microfiche strip contains the following content from top to bottom:

- Frame 1: A small diagram or table.
- Frame 2: A small diagram or table.
- Frame 3: A small diagram or table.
- Frame 4: A flowchart with several rectangular boxes connected by arrows.
- Frame 5: A table with multiple columns and rows of data.
- Frame 6: A table with multiple columns and rows of data.
- Frame 7: A table with multiple columns and rows of data.
- Frame 8: A table with multiple columns and rows of data.
- Frame 9: A table with multiple columns and rows of data.
- Frame 10: A table with multiple columns and rows of data.
- Frame 11: A table with multiple columns and rows of data.
- Frame 12: A table with multiple columns and rows of data.
- Frame 13: A table with multiple columns and rows of data.
- Frame 14: A table with multiple columns and rows of data.
- Frame 15: A table with multiple columns and rows of data.



IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZBMM-A-D

PRODUCT NAME: BM792YM CASSETTE BOOTSTRAP LOADER

DATE CREATED: 9 MARCH 1973

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: JIM KAPADIA

COPYRIGHT(C) 1973
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS

1. ABSTRACT

THE DZBHM DIAGNOSTIC PROGRAM IS WRITTEN TO BE USED AS AN AID TO HARDWARE DEBUGGING AND MAINTENANCE OF THE BM792YM CASSETTE 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
- B. BM792-YM 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
 THE NUMBER OF FILL CHARACTERS AFTER A CARRIAGE RETURN MAY BE PATCHED INTO LOCATION 1101. NINE (9) ARE REQUIRED FOR AN L300 AT 300 BAUD.

DEPRESS START, THE PROGRAM WILL THEN TYPE OUT INSTRUCTIONS. ALL USER RESPONSES ARE VIA THE 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 BM792YM LOGIC. PROGRAM 0 LOOPS WITHIN ITSELF UNTIL A NEW PROGRAM NUMBER IS PROVIDED I.E. PRG00 ?

5.1.1 ROUTINE DESCRIPTIONS

ROUTINE	TESTS
T1	ADDRESSABILITY OF BM792YM
T2	DATA RELIABILITY
T3	THAT BM792YM 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 'PRG00' 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	000000	6
4010	012703	10

ETC.

1

.TITLE TEST DEBMM CASSETTE BOOTSTRAP LOADER
.MLIST SC0:MC
.LIST ME
.ADS
!LOAD ADDRESS=0200
!DEPRESS START
!RESTART ADDRESS=0910
!STACK POINTER IS AT 900

.MCALL SCRRGS,SCPVECS
.MCALL STYPE
.MCALL SCMTAG
!DEFINITIONS AND REGISTER ASSIGNMENTS
!GENERAL REGISTER ASSIGNMENTS

000000	R0=X0
000001	R1=X1
000002	R2=X2
000003	R3=X3
000004	R4=X4
000005	R5=X5
000006	SP=X6
000007	PC=X7
000008	R10=X8
000009	R11=X9
000010	R12=X10
000011	R13=X11
000012	R14=X12
000013	R15=X13
000014	R16=X14
000015	R17=X15

!REGISTER ADDRESSES

177770	PSW=	177770
177774	SLR=	177774
177772	PIR=	177772
177770	UBREAK=	177770
177570	SWR=	177570
177570	DISPLAY=	177570
177560	TKB=	177560
177562	TKB=	177562
177564	TPB=	177564
177566	TPB=	177566

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

!VECTOR ADDRESSES

000004	ZRRVEC=4
000010	RESVEC=10
000014	TRITVEC=14
000014	TRIVVEC=14
000014	OPTVEC=14
000020	IOIVVEC=20
000024	PFVEC=24
000030	EMIVVEC=30
000034	TRAPVEC=34
000060	TKVEC= 60
000064	TPVEC=64
000114	PARVEC=114

!ADDRESS OF ERROR VECTOR
!ADDRESS OF RESERVED INST. TRAP VECTOR
!ADDRESS OF 'I' BIT TRAP VECTOR
!ADDRESS OF 'TRACE' TRAP VECTOR
!ADDRESS OF 'BREAKPOINT' TRAP VECTOR
!ADDRESS OF 'IO' TRAP VECTOR
!ADDRESS OF POWER FAIL TRAP VECTOR
!ADDRESS OF 'EM' VECTOR
!ADDRESS OF TRAP VECTOR
!ADDRESS OF TTY KEYBOARD INT. VECTOR
!ADDRESS OF TTY PRINTER INTERRUPT VECTOR
!ADDRESS OF MA/MF PARITY ERROR VECTOR

	ADR240		PIRVEC=240		ADDRESS OF PIRG VECTOR
	ADR244		FPEVEC=244		ADDRESS OF FLOATING POINT INT. VECTOR
	ADR250		MMVEC=250		ADDRESS OF MEM MGMT ERROR TRAP VECTOR
	ADR020		.B20		
ADR020	ADR110		.WORD	:TYPF	
ADR022	ADR240		.WORD	240	
	ADR030		.B30		
ADR030	ADR2376		.SCOPE		
ADR032	ADR340		340		
ADR034	ADR452		.MLT		
ADR036	ADR000		0		
	ADR0500		IEQUATE STATEMENTS		
	ADR004		STKPTR=	500	
	ADR400		TYPE=	!0?	
	ADR000		MLT=	TRAP	
	ADR000		SCOPE=	ENT	
	ADR000		.B200		
ADR200	ADR167	ADR1132	START1: JMP	PRMTRB	
	ADR210		.B210		
ADR210	ADR167	ADR1156	START3: JMP	RESTART	
	ADR100		.B100		
			ROUTINE TO TYPE ASCII MESSAGE: MESSAGE MUST TERMINATE WITH A 0 BYTE:		
			THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.		
			CALL TYPE		
			MSADR		MSADR IS FIRST ADDRESS OF ASCII STRING
			ITAGS USED BY THE TYPE ROUTINE BELOW		
ADR100	000		SNULLI .BYTE	0	CONTAINS NULL CHARACTER
ADR101	002		SFILLI .BYTE	2	CONTAINS # OF FILLER CHARACTERS
ADR102	000		STPFLG1 .BYTE	0	CONTAINS TELEPRINTER AVAILABLE FLAG
					0/3?? = AVAIL/NOT AVAIL
ADR103	000		STRFLG1 .BYTE	0	CONTAINS KEYBOARD AVAILABLE FLAG
ADR104	ADR177564		STPS1 .WORD	ADR177564	ADDRESS OF TELEPRINTER STATUS REGISTER
ADR106	ADR177566		STPS1 .WORD	ADR177566	ADDRESS OF TELEPRINTER DATA BUFFER
ADR110	ADR10040		.TYPE1 MOV	R0,0(SP)	SAVE R0
ADR112	ADR17600	ADR00002	MOV	02(SP),R0	GET MESSAGE ADDRESS
ADR116	ADR62766	ADR00002	ADD	02,2(SP)	ADJUST RETURN PC
ADR124	ADR112046		15) MOVB	(R0),0(SP)	PUSH CHARACTER TO BE TYPED ONTO STACK
ADR126	ADR10003		BNE	25	BRANCH IF NOT THE TERMINATOR
ADR130	ADR05726		TST	(SP),0	POP TERMINATOR CHAR OFF THE STACK
ADR132	ADR012600		MOV	(SP),R0	RESTORE R0
ADR134	ADR00002		RTI		RETURN TO CALLER
ADR136	ADR004767	ADR000026	25) JSR	PC:55	TYPE CHARACTER
ADR142	ADR122726	ADR000012	35) CMPB	0121(SP),0	CHECK IF CHARACTER HAS A LINE FEED
ADR146	ADR001366		BNE	15	BRANCH IF NOT LINE FEED
ADR150	ADR016746	ADR177724	MOV	SNULL,0(SP)	GET # OF FILLERS REQUIRED AND FILLER CHARACTER
ADR154	ADR109366	ADR000001	45) DECB	1(SP)	DECREMENT FILLERS REQ. COUNT

001140	002770		BLT	35		I BRANCH IF NO MORE FILTERS ARE REQUIRED
001142	004767	000002	JSR	PC:35		I TYPE FILTER CHARACTER
001146	000772		BR	48		
001170	109777	177710	SSI	TSTB	05TPB	I WAIT FOR OUTPUT DEVICE
001174	100375		BPL	:-4		
001176	116677	000002 177702	MOVB	2(9P),05TPB		I OUTPUT CHARACTER
001204	000207		RTS	PC		
	001300			.01300		
I WORDS LOADED BY 'SCOPE'						
001300	000000		SCPBLK1	.WORD	0	I CONTAINS PASS COUNT
	001300			ICNT=SCPBLK		
001302	000002		ITCNT1	.WORD	2	I CONTAINS SURTEST ITERATION COUNT
001304	000004		LASTPC1	.WORD	4	I CONTAINS LAST SCOPE CALL PC
001306	000000		EPCI	.WORD	0	I CONTAINS SCOPE RETURN FOR ERRORS
001310	000000		ERRFLG1	.WORD	0	I CONTAINS ERROR FLAG
001312	000000		TICKS1	.WORD	0	I CONTAINS TICK COUNT FOR CLOCKS
001314	177560		INCSR1	.WORD	177560	I ADDRESS OF INPUT CSR
001316	177562		INDATI	.WORD	177562	I ADDRESS OF INPUT DEVICE DATA BUFFER REG
001320	000040		WORDS1	32		
001322	004000		IMAGE1	4000		
001324	000000		TEMP1	0		
001326	173300		ROMADD1	173300		I FIRST ADDRESS OF DATA
001330	001404		PRGTAB1	PRG0		
001332	000120			PRG1		
001334	000222			PRG2		
001336	012700	000500	PRMTR1	MOV	05TKPTR,X0	I SET STACK PTR
001342	009007	000010		CLR	PRNUM	
001346	000737	177570		TST	00000	
001352	001407			BEC	RESTART	
001354	000004			TYPE		
001356	000010			NO		
001360	004567	000732		JSR	9,RECD	I RECEIVE DATA AND PUT
001364	000000		PRNUM1	0		I IF HERE
001366	000004			TYPE		
001370	000042			NO		
001372	016700	177766	RESTART1	MOV	PRNUM,X0	I GET PROGRAM 0
001376	006300			ASL	X0	I SHIFT PROGRAM 0
001400	000170	001330		JMP	OPRGTAB(0)	I GO TO PROGRAM

```

PROGRAM @ LOGIC TESTS
PR001 CLR ICNT ICLEAR PASS COUNT
PR001 MOV BT, LASTPC ISET RETURN ADDRESS FOR SCOPE
PR001 MOV ICNT, 0001 DISPLAY IDISPLAY PASS COUNT

;TEST1 TEST ABILITY TO REFERENCE ROM WITHOUT TIMING OUT
T1I MOV BSYMPTR, SP ISET STACK PTR
MOV ROMADD, XP IGET ROM ADDRESS
MOV WORDS, X1 IGET ADDRESS COUNTER
MOV ERROR1, 4 ISET UP TIME OUT VECTOR
T1A1 MOV (0), X3 IREFERENCE
TST (0) FROM
ADD -(0), TEMP !
CMP (0), (0) !
BITB (0), (0) !
SUB -(0), TEMP !
ADD 02, X0 IINCREMENT POINTER
DEC X1 IDECREMENT ADDRESS COUNTER
BNE T1A IBRANCH IF NOT FINISHED
BR T1B IGO TO SCOPE LOOP
ERROR11 CMP (6), (6) IREPOSITION STACK
HLT IERROR1 ROM TIMED OUT WHEN REFERENCED
IADDRESS IS IN ROM
ILOOP ON ERROR

T1B1 BR T1A
SCOPE

;TEST2 TEST THAT ROM DATA CAN BE READ RELIABLY.
T2I MOV ROMADD, XP IGET ROM ADDRESS
MOV WORDS, X1 IGET ADDRESS COUNTER
MOV 06, 4 IINITIALIZE TIME OUT VECTOR
T2A1 CLR TEMP IINITIALIZE TEMP
MOV (0), X3 IGET DATA
ADD (0), TEMP IADD DATA TO TEMP
SUB TEMP, X3 ISUBTRACT DATA FROM DATA
BEO T2B IBRANCH IF EQUAL
ERROR21 HLT IDATA ERROR
BR T2A ILOOP ON ERROR
T2B1 BIC -(0), TEMP ICLEAR TEMP BITS
BEO T2C IBRANCH IF EQUAL TO 0
HLT IDATA ERROR
BR T2B ILOOP ON ERROR
T2C1 CMP (0), (0) ICOMPARE DATA
BEO T2D IBRANCH IF EQUAL
HLT IDATA ERROR
BR T2B ILOOP ON ERROR
T2D1 CMPB (0), -(0) ICOMPARE DATA (BYTE OPERATION)
BEO T2E IBRANCH IF EQUAL
HLT IDATA ERROR
BR T2B ILOOP ON ERROR
T2E1 TST (0) IINCREMENT ADDRESS POINTER
DEC X1 IDECREMENT ADDRESS COUNTER
BNE T2A IRETURN IF NOT DONE
SCOPE

```


ITEST3 TEST THAT ROM TIMES OUT IF REFERENCED BY OTHER
ITMAN DATA BUS CYCLE

FB1614	R127F6	0005F0		T3i	MOV	05HPTX,X6	ISSET STACK PTR
FB1620	R16700	177502			MOV	ROMADD,XP	IGET ROM ADDRESS
FB1624	R16701	177470			MOV	WORDS,X1	IGET ADDRESS COUNTER
FB1630	R12767	001644	176140	T3AAI	MOV	0730,4	ISSET UP TIME OUT VECTOR
FB1636	R10010			T3AI	MOV	X0,(0)	IATTEMPT TO ALTER DATA
FB1640	I044F0				HLT		HERE IF D1N NOT TIME OUT
FB1642	000775				BR	T3A	ILOOP ON ERROR
FB1644	R12767	001662	176132	T30i	MOV	0730,4	ISSET UP TIME OUT VECTOR
FB1652	022626				CMP	(0), (0)	IPOSITION STACK
FB1654	005210			T3Ci	INC	(0)	IATTEMPT TO ALTER DATA
FB1656	I044F0				HLT		HERE IF D1N NOT TIME OUT
FB1660	000775				BR	T3C	ILOOP ON ERROR
FB1662	R12767	001702	176114	T30i	MOV	073F,4	ISSET UP TIME OUT VECTOR
FB1670	022626				CMP	(0), (0)	IPOSITION STACK
FB1672	005077	177430		T3Ei	CLR	0ROMADD	IATTEMPT TO ALTER DATA
FB1676	I044F0				HLT		HERE IF D1N NOT TIME OUT
FB1700	000774				BR	T3E	ILOOP ON ERROR
FB1702	005720			T3Pi	TST	(0)	INCREMENT ADDRESS POINTER
FB1704	022626				CMP	(0), (0)	IPOSITION STACK
FB1706	005301				DEC	X1	IDECREMENT ADDRESS COUNTER
FB1710	001347				BNE	T3AA	IRETURN IF NOT DONE
FB1712	R12737	000006	000004		MOV	06,004	IRESTORE TIME OUT TRAP
FB1720	I040F0				SCOPE		ISCOPE LOOP

ITMIS TEST COMPARES ROM AND IMAGE DATA
IAND TYPES OUT DIFFERENCES

FB1722	R127F6	000500		T4i	MOV	05HPTX,X6	ISSET STACK PTR
FB1726	R16701	177366			MOV	WORDS,X1	IGET # OF WORDS
FB1732	R16700	177370			MOV	ROMADD,X0	IGET ROM ADDRESS
FB1736	R16703	177360			MOV	IMAGE,X3	IGET IMAGE ADDRESS
FB1742	R21013			T40i	CMP	(0), (3)	ICOMPARE DATA
FB1744	R01004				BNE	T40	
FB1746	005301			T4Ei	DEC	X1	IALL DATA BEEN COMPARED
FB1750	001433				BEQ	T4E	
FB1752	022023				CMP	(0), (3)	INCREMENT ADDRESS POINTERS
FB1754	000772				BR	T40	
FB1756	R10067	000530		T40i	MOV	X0,020TYP	ITYPE
FB1762	004767	000526			JSR	7,02A	IRON ADDRESS
FB1766	R00004				TYPE		
FB1770	007727				H10		ISEPARATOR
FB1772	R11067	000514			MOV	(0), 020TYP	ITYPE
FB1776	004767	000512			JSR	7,02A	IRON DATA
FB1802	R00004				TYPE		
FB1804	002642				H0		ICR/LF
FB1806	R10367	000500			MOV	X3,020TYP	ITYPE
FB1812	004767	000476			JSR	7,02A	IIMAGE ADDRESS
FB1816	R00004				TYPE		
FB1820	002733				H12		ISEPARATOR
FB1822	R11367	000464			MOV	(3), 020TYP	ITYPE
FB1826	004767	000462			JSR	7,02A	IIMAGE DATA
FB1832	R00004				TYPE		
FB1834	002642				H0		ICR/LF

PC7076	PC743			OR	T4C		IGO TO T4C
PC7040	104000			T4E:	SCOPE		
PC7042	005267	177232		END:	INC	ICNT	I INCREMENT PASS COUNT
PC7046	026727	177226	000100		CMF	ICNT,010P	
PC7054	001402				BEO	00NE	
PC7056	000167	177326			JMP	PRGR	IGO RESTART PROGRAM
PC7042	012737	000207	177566	DONE1	MOV	0207,00TPB	IRING THE OFIL
PC7070	109737	177564			TSTB	00TPB	
PC7074	100375				BPL	.-4	
PC7076	013700	000042			MOV	0042,XB	IRETURN TO OFCTAPE MONITOR?
PC7102	001404				BEO	00NE1	
PC7104	004710				JSR	7.(0)	IRETURN:
PC7106	000240				NOP		
PC7110	000240				NOP		
PC7112	000240				NOP		
PC7114	000167	177264		DONE1:	JMP	PRGR	

THIS PROGRAM TYPES OUT ROM DATA

```

002120 012706 000500
002124 000004
002126 002625
002130 016701 177164
002134 016700 177166
002140 012702 000012
002144 010067 000342
002150 004767 000340
002154 000004
002156 002642
002160 012067 000326
002164 004767 000324
002170 000004
002172 002731
002174 009301
002176 001407
002200 009302
002202 001366
002204 012702 000012
002210 000004
002212 002642
002214 000793
002216 000167 177114
    
```

```

PRG11  MOV  @SYMPTR,X6      ;INITIALIZE STACK
        TYPE
        M7
        MOV  WORDS,X1      ;FROM DATA
PRG1A1  MOV  ROMADD,X0      ;GET # OF WORDS
        MOV  @19,X2        ;GET STARTING ADDRESS
PRG1B1  MOV  @0,D2BTYP      ;GET ADDRESS INDICATOR
        JSR  7,02A        ;GET ADDRESS
        TYPE
        M8
        MOV  @0,D2BTYP      ;CR/LF
        JSR  7,02A        ;TYPE
        TYPE
        M11
        DEC  X1
        BEQ  PRG10        ;DATA TYPED
        DEC  X2
        BNE  PRG1C        ;GO TO FINISH
        MOV  @19,X2
        TYPE
        M8
        BR   PRG1B        ;RETURN TO PRG1B
PRG1D1  JMP  @RMTYS        ;GO GET NEXT TEST
    
```

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

```

002222 012706 000500
002226 012737 002312 000004
002234 009067 179536
002240 012737 002272 000060
002246 012737 000340 000062
002254 012737 000100 179560
002262 016700 177040
002266 009710
002270 000776
002272 004567 000020
002276 000000
002300 016700 177772
002304 000004
002306 002642
002310 000002
002312 104400
002314 000777
    
```

```

PRG21  MOV  @SYMPTR,X6      ;INITIALIZE STACK POINTER
        MOV  @PRG2C,004    ;LOAD TRAP ERROR VECTOR
        CLR  PSM           ;CLEAR PROCESSOR STATUS
        MOV  @PRG2A,@TKVEC  ;LOAD KEYBOARD INTERRUPT VECTOR
        MOV  @340,@TKVEC+2  ;LOAD KEYBOARD PRIORITY
        MOV  @100,@TKS      ;SET INTERRUPT ENABLE BIT
        MOV  ROMADD,X0      ;GET ROM ADDRESS
        TST  @0
        BR   :-?          ;LOOP
PRG2A1  JSR  @,REC0        ;GO GET ADDRESS &
PRG2B1  @
        MOV  @PRG2B,X0
        TYPE
        M8
        RTI
PRG2C1  MLE
        BR   :
    
```

```

;CR/LF
;EXIT KEYBOARD INTERRUPT SERVICE
;ERROR! DID YOU TYPE AN ODC ADDRESS?
;STAY HERE UNTIL CORRECT ADDRESS IS TYPED IN
    
```

```

ROUTINE TO RECEIVE DATA TYPED IN ON THE KEYBOARD. THE DATA IS PLACED IN
THE ADDRESS FOLLOWING THE JSR CALL:
|
| JSR 9,RECD |CALL RECEIVE DATA ROUTINE
| RECD: CLR (5) |DATA IS PLACED HERE
15i TSTB 007NS |CLEAR OUT OLD DATA
| BPL -4 |TEST KEYBOARD FLAG
| MOVB 007NR,(9P) |AND WAIT FOR CHARACTER
| BIC 0200,(SP) |GET CHARACTER
| CMFB 019,(SP) |STRIP PARITY BIT
| BNE 25 |CHECK IF CARRIAGE RETURN
| TYPE |BRANCH IF NOT CARRIAGE RETURN
| MB
| CMP (R5),(SP) |ADJUST R5 AND THE STACK PTR
| RYS R5 |RETURN TO CALLER
25i MOV (SP),007PB |ECHO CHARACTER
35i BIC 017770,(SP) |STRIP AWAY ALL BUT 3 LSB
| ASL (5) |ROTATE
| ASL (5) |PREVIOUS
| ASL (5) |DATA
| BIS (SP),(5) |AND INSERT CHARACTER
| BR 15 |GET NEXT CHARACTER

SCOPE ROUTINE. THIS ROUTINE IS ENTERED AT THE END OF EACH SUBTEST.
SCOPE: BIT 040000,005NR |TEST SR FOR SCOPE
| BNE 25 |YES SCOPE
| BIT 04000,005NR |TEST FOR ITERATION
| BNE 15 |INHIBIT ITERATION
| INC ITCNT |INCREMENT ITERATION COUNT
| CMP ITCNT,ICOUNT |ITERATION COMPLETE
| BGT 25 |BRANCH IF ITERATIONS NOT COMPLETE
15i CLR ITCNT |CLEAR ITERATION COUNT
25i MOV (SP),LASTPC |GET ADDRESS OF NEXT TEST
| MOV LASTPC,(SP)
| RTI |EXIT
ICOUNT: 5

ERROR ROUTINE. THIS ROUTINE IS ENTERED WHEN AN ERROR IS DETECTED:
HLTI BIT 005NR,02F000 |INHIBIT PRINTOUT?
| BEO -4 |BRANCH IF ERROR PRINT OUT
| RTI |RETURN TO TEST
| TYPE
| ERRORM |IPCO
| MOV (6),0207VP |TYPE PROGRAM COUNTER
| JSR 9,00A
| TST 005NR |HALT ON ERROR
| BPL -4 |
| HALT |YES HALT
| RTI |RETURN TO TEST

```

```

THIS ROUTINE CONVERTS AN OCTAL NUMBER TO ASCII AND TYPES IT ON THE TTY.
D2BTYP1 0
024:  MOV      X2,P(6)      ISAVE R2
      MOV      X1,P(6)      ISAVE R1
      MOV      X0,O(6)      ISAVE R0
      MOV      D2BTYP,XR    IGET DATA TO BE TYPED
      MOV      06.21        IGET COUNTER
      CLR      X2           ICLEAR WORKING REGISTER
      ROL      X0           INOV FIRST BIT (MSB) INTO
      ROL      X2           IR2
      ADD      0260,XR      IFORM ASCII CODE
      MOV      R2,D2BTYP
      TYPE
      D2BTYP
      CLR      X2           ICLEAR WORKING REGISTER
      ROL      X0           IROTATE THE
      ROL      X2           INEXT
      ROL      X0           IOCTAL CHARACTER
      ROL      X2           IINTO
      ROL      X0           IREGISTER
      ROL      X2           IYMO
      DEC      X1           IDECREMENT COUNTER
      BNE      15          IGO TO 02AA IF NOT 0
      MOV      (6)0,XR     IFINISHED. RESTORE REGISTERS
      MOV      (6)0,X1
      MOV      (6)0,X2
      RTS      7          IAND EXIT

```

002512	000000	
002514	010246	
002516	010146	
002520	010046	
002522	016700	177764
002526	012701	000006
002532	009002	
002534	006100	
002536	006102	
002540	062702	000260
002544	010267	177742
002550	000004	
002552	002512	
002554	009002	
002556	006100	
002560	006102	
002562	006100	
002564	006102	
002566	006100	
002570	006102	
002572	009301	
002574	007361	
002576	012600	
002600	012601	
002602	012602	
002604	000207	

```

ASCII MESSAGES
P02606 009019 041920 020075 ERRORMI .ASCII <19><12>'PC0 '
P02614      000
P02619      019 050012 043522 M6i .ASCII <19><12>'PRG00'
P02622 036443      000
P02625      019 051012 046517 M7i .ASCII <19><12>'ROM DATA'<19><12>
P02632 042040 052101 006501
P02640 000012
P02642 009019      000 M8i .ASCII <19><12>
P02645      019 051012 046517 M9i .ASCII <19><12>'ROM ADDRESS/IMAGE ADDRESS ROM DATA,IMAGE DATA'<19><12>
P02652 040440 042104 042522
002660 051923 044457 040519
002666 042907 040440 042104
002674 042522 051923 051040
002702 046517 042040 052101
P02710 029101 046511 043501
P02716 020105 040504 040524
P02724 009019      000
P02727      057      000 M10i .ASCII ' / '
002731      040      000 M11i .ASCII ' '
P02733      052      000 M12i .ASCII '0'

      003776      .03776
003776 000000      .WORD
      IDATA OUT INTO THE D2BMM
P04000 012700 177500 009010 012700,177500,009010,010701
P04006 010701
P04010 062701 000052 012702 062701,000052,012702,000379
004016 000379
P04020 112103 112110 100413 112103,112110,100413,130310
P04026 130310
P04030 001776 105202 100772 001776,105202,100772,116010
004036 116012
004040 000002 120337 000000 000002,120337,000000,001769
004046 001767
004050 000000 000755 005710 000000,000755,005710,100774
004056 100774
004060 005007 017640 002419 005007,017640,002419,112024
004066 112024
004070 000000 000000 173300 000000,000000,173300,000340
004076 000340

      .END
  
```


JPTVEC = 000714	DISPLA = 17757F	DONE = 002F62	DONE1 = 002114
D2BTYP = 0F2519	ENTVEC = 00003F	END = 002F42	EPC = 001306
ERRFLG = 0F171F	ERRORM = 002606	ERROR1 = 0F1500	ERROR2 = 001546
ERRVEC = 0F0004	FPEVEC = 0F0244	HLT = 1F4400	ICNT = 001300
ICOUNT = 0F245F	IMAGE = 0F1322	INCSR = 001314	INDAT = 001316
IOTVEC = 0F002F	ITCNT = 001302	LASTPC = 001304	MMVEC = 0F0250
M1F = 0F2727	M11 = 002731	M12 = 002733	M6 = 0F2619
M7 = 0F2625	M8 = 0F2642	M9 = 002645	N2A = 002514
PARVEC = 000114	PC = X000007	PFVEC = 0F0F24	PIRO = 177772
PIRVEC = 0F024F	PRGNUM = 001364	PGTAB = 001330	PR60 = 001404
PR60R = 0F1410	PRG1 = 0F2120	PRG1A = 002134	PR610 = 002146
PRG1C = 0F2160	PRG1D = 0F2216	PRG2 = 002222	PRG2A = 002272
PRG20 = 0F227A	PRG2C = 0F2310	PRMTRS = 001336	PSW = 177776
RECD = 002716	RESTAR = 0F1370	RESVEC = 0F0E10	ROMADD = 001320
R0 = X0F000F	R1 = X000001	R10 = X000000	R11 = X000001
R12 = X0F0007	R13 = X000003	R14 = X000004	R15 = X000009
R2 = X0F0007	R3 = X000003	R4 = X000004	R5 = X000009
SCOPE = 10400F	SCPBLK = 001300	SLR = 177774	SP = X000006
START1 = 0F020F	START3 = 000210	STKPTR = 000500	SWR = 177570
YBITVE = 000014	TEMP = 001324	TICKS = 001312	TK0 = 177562
YKS = 177560	TKVEC = 000060	TP0 = 177566	TPS = 177564
YPVEC = 000064	TRAPVE = 000034	TRTVEC = 000E14	TYPE = 000004
T1 = 001424	T1A = 001446	T1B = 001506	T2 = 001510
T2A = 00152A	T20 = 001502	T2C = 001564	T20 = 001574
T2E = 001404	T3 = 001614	T3A = 001636	T3AA = 001630
T30 = 001644	T3C = 001654	T3D = 001662	T3E = 001672
T3F = 001707	T4 = 001722	T40 = 001742	T4C = 001746
T4D = 00175A	T4E = 002040	UBREAK = 177770	WORDS = 001320
SFILL = 0F1101	SNULL = 001100	SYNPLG = 0F1103	STPB = 001106
STPFLG = 0F1107	SYPS = 001104	.HLT = 002452	.SCOPE = 002376
.TYPE = 0F111F	.		

ERRORS DETECTED 0

TEST DZBMH CASSETTE BOOTSTRAP LOADER
DZBMH

MACY11.623 12-MAR-73 16,33 PAGE 12

•DZBMH;DZBMH/SOL•DZBMH
RUN-TIME 0 3 0 SECONDS
CORE USED 4K