

**BM792YA-H,K,L**

**BOOTSTRAP LOADER TEST  
MD-11-DZBMA-E**

**EP-DZBMA-E-DL**

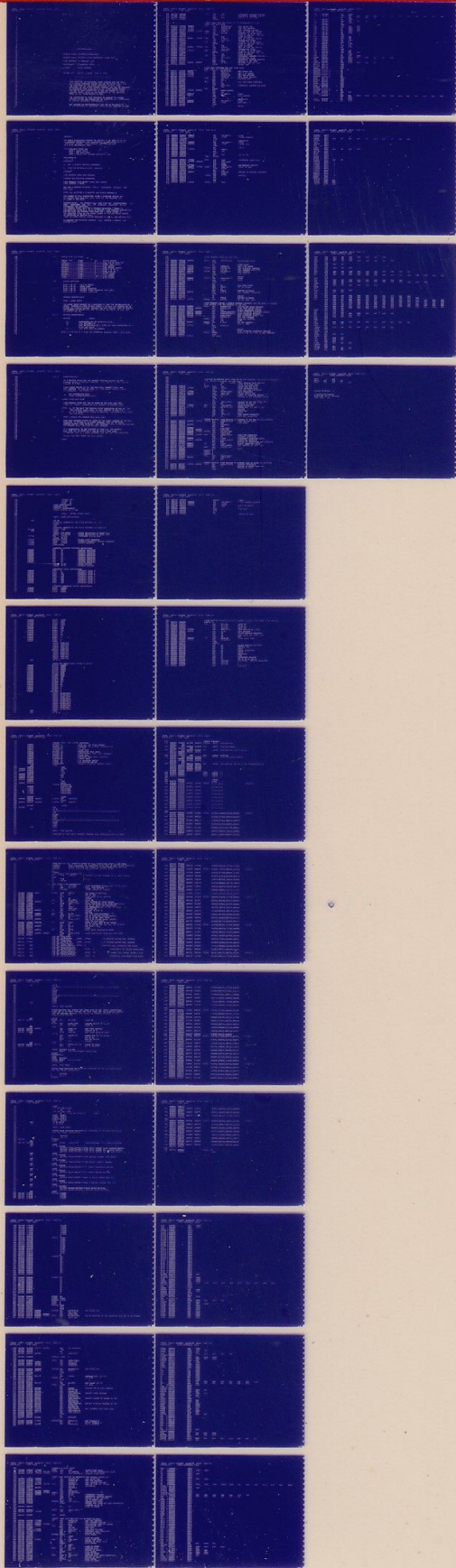
**MAY 1978**

**COPYRIGHT © 72-75**

**digital**

**FICHE 1 OF 1**

**MADE IN USA**





.REPT 0

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DZBMA-E-0

PRODUCT NAME: SM792VA-H,K,SL BOOTSTRAP LOADER TEST

DATE CREATED: 13 FEBRUARY 1973

MAINTAINER: DIAGNOSTIC GROUP

AUTHOR: BRUCE BURGESS

REVISED BY: DAVID L. ADAMS JUNE 3, 1975

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION

1972, 1973, 1975

THIS SOFTWARE IS FURNISHED UNDER LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106

1. ABSTRACT

THE DZBMA DIAGNOSTIC PROGRAM IS WRITTEN TO BE USED AS AN AID TO HARDWARE DEBUGGING AND MAINTENANCE OF THE 8M792YA-M,K,SL 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 8M792YA-M,K,OR L MODULES

2.2 STORAGE

THIS PROGRAM USES CORE 8-ENDAD

3. LOADING AND STARTING PROCEDURE

LOAD PROGRAM INTO MEMORY USING ASS LOADER,  
LOAD ADDRESS = 00200

SET SWR = DESIRED STANDARD POP-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 LA300 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 " PRG0 ". THE OPERATOR TYPES 0,1, OR 2 AS DESIRED. (SEE PROGRAM DESCRIPTION SECTION 5.0) 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 5.3.

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



100  
 109  
 110  
 111  
 112  
 113  
 114  
 115  
 116  
 117  
 118  
 119  
 120  
 121  
 122  
 123  
 124  
 125  
 126  
 127  
 128  
 129  
 130  
 131  
 132  
 133  
 134  
 135  
 136  
 137  
 138  
 139  
 140  
 141  
 142  
 143  
 144  
 145  
 146  
 147  
 148  
 149  
 150  
 151  
 152

3.1 MODULE TYPE AND IMAGE

MODULE TYPE	IMAGE	NO.	OPTION BOOTED
M792	IMAGI	0	(ALL 1'S NO DATA CUT)
M792YA	IMAGA	1	PAPER TAPE LOADER
M792YB	IMAGB	2	DISK LOADER
M792YC	IMAGC	3	CARD READER LOADER
M792YD	IMAGD	4	CARD 1 OF MR 11
M792YE	IMAGE	5	CARD 2 OF MR 11
M792YF	IMAGF	6	DS500 LOADER
M792YH	IMAGH	7	TA11 LOADER
M792YK	IMAGK	10	V120 LOADER
M792YL	IMAGL	11	RA11 LOADER

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 M792YA-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 0.

5.1.1 ROUTINE DESCRIPTIONS

ROUTINE	TESTS
T1	ADDRESSABILITY OF M792YA-M,K,OR L
T2	DATA RELIABILITY
T3	THAT M792YA-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.



154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183  
184  
185  
186  
187  
188  
189  
190

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#0' 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



```

192          .ENABLE ABS
193          .NLIST MC
194          .LIST ME
195          JLOAD ADDRESS=0200
196          JUEPRESS START
197          JRESTART ADDRESS=0210
198          JSTACK POINTER IS AT 500
199
200          .MCALL .STYPE,.SIRAP,.EQUAT
201
(1)          .DBTTL BASIC DEFINITIONS
(1)
(1)          001          .IF NB
(1)          JINITIAL ADDRESS OF THE STACK POINTER *** ***
(1)          STACK=
(1)          .IFF
(1)          JINITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
(1)          001100     STACK= 1100
(1)          000
(1)          .ENDC
(1)          .EQUIV EMT,ERROR          JBASIC DEFINITION OF ERROR CALL
(1)          .EQUIV IOT,SCOPE          JBASIC DEFINITION OF SCOPE CALL
(1)          177776     PS= 177776          JPROCESSOR STATUS WORD
(1)          .EQUIV PS,PSW
(1)          177774     STKLMT= 177774          JSTACK LIMIT REGISTER
(1)          177772     PIR0= 177772          JPROGRAM INTERRUPT REQUEST REGISTER
(1)          177570     SNR= 177570          JSWITCH REGISTER
(1)          177570     DISPLAY=SNR
(1)
(1)          JGENERAL PURPOSE REGISTER DEFINITIONS
(1)          000000     R0= 0          JGENERAL REGISTER
(1)          000001     R1= 1          JGENERAL REGISTER
(1)          000002     R2= 2          JGENERAL REGISTER
(1)          000003     R3= 3          JGENERAL REGISTER
(1)          000004     R4= 4          JGENERAL REGISTER
(1)          000005     R5= 5          JGENERAL REGISTER
(1)          000006     R6= 6          JGENERAL REGISTER
(1)          000007     R7= 7          JGENERAL REGISTER
(1)          .EQUIV R6,SP          JSTACK POINTER
(1)          .EQUIV R7,PC          JPROGRAM COUNTER
(1)
(1)          JPRIORITY LEVEL DEFINITIONS
(1)          000000     PH0= 0          JPRIORITY LEVEL 0
(1)          000040     PH1= 40         JPRIORITY LEVEL 1
(1)          000100     PH2= 100        JPRIORITY LEVEL 2
(1)          000140     PH3= 140        JPRIORITY LEVEL 3
(1)          000200     PH4= 200        JPRIORITY LEVEL 4
(1)          000240     PH5= 240        JPRIORITY LEVEL 5
(1)          000300     PH6= 300        JPRIORITY LEVEL 6
(1)          000340     PH7= 340        JPRIORITY LEVEL 7
(1)
(1)          J"SWITCH REGISTER" SWITCH DEFINITIONS
(1)          100000     SW15= 100000
(1)          040000     SW14= 40000
(1)          020000     SW13= 20000

```



```

(1)      010000      SW12= 10000
(1)      004000      SW11= 4000
(1)      002000      SW10= 2000
(1)      001000      SW09= 1000
(1)      000400      SW08= 400
(1)      000200      SW07= 200
(1)      000100      SW06= 100
(1)      000040      SW05= 40
(1)      000020      SW04= 20
(1)      000010      SW03= 10
(1)      000004      SW02= 4
(1)      000002      SW01= 2
(1)      000001      SW00= 1
(1)      .EQUIV      SW09,SW9
(1)      .EQUIV      SW08,SW8
(1)      .EQUIV      SW07,SW7
(1)      .EQUIV      SW06,SW6
(1)      .EQUIV      SW05,SW5
(1)      .EQUIV      SW04,SW4
(1)      .EQUIV      SW03,SW3
(1)      .EQUIV      SW02,SW2
(1)      .EQUIV      SW01,SW1
(1)      .EQUIV      SW00,SW0
(1)      001      .IF 0 <>
(1)
(1)      .DATA      BIT DEFINITIONS (BIT00 TO BIT15)
(1)      100000      BIT15= 100000
(1)      040000      BIT14= 40000
(1)      020000      BIT13= 20000
(1)      010000      BIT12= 10000
(1)      004000      BIT11= 4000
(1)      002000      BIT10= 2000
(1)      001000      BIT09= 1000
(1)      000400      BIT08= 400
(1)      000200      BIT07= 200
(1)      000100      BIT06= 100
(1)      000040      BIT05= 40
(1)      000020      BIT04= 20
(1)      000010      BIT03= 10
(1)      000004      BIT02= 4
(1)      000002      BIT01= 2
(1)      000001      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 0 <>
  
```







```

(1)                                     ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1) ;*NOTE1: SNULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1) ;*NOTE2: SFILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1) ;*NOTE3: SFILLC 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,STYPE      ;CALL TYPE ROUTINE
(1) ;*   MESADDR      ;FIRST ADDRESS OF MESSAGE
(1)
(1) 001000 105767 000111  STYPE:  TSTB  STPFLG  ;IS THERE A TERMINAL?
(1) 001004 100002          BPL  10      ;BR IF YES
(1) 001006 000000          HALT          ;HALT HERE IF NO TERMINAL
(1) 001010 000407          BR  30      ;LEAVE
(1) 001012 010046 10:     MOV  R0,-(SP) ;SAVE R0
(1) 001014 017600 000002  MOV  02(SP),R0 ;GET ADDRESS OF ASCIZ STRING
(1) 001020 112046 20:     MOVB (R0)+,-(SP) ;PUSH CHARACTER TO BE TYPED ONTO STACK
(1) 001022 001005          BNE  40      ;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 30:     ADD  02,(SP) ;ADJUST RETURN PC
(1) 001034 000002          RTI          ;RETURN
(1) 001036 004767 000026 40:     JSR  PC,78   ;GO TYPE THIS CHARACTER
(1) 001042 126726 000046 50:     CMPB SFILLC,(SP)+ ;IS IT TIME FOR FILLER CHARS.?
(1) 001046 001362          BNE  20      ;IF NO GO GET NEXT CHAR.
(1) 001050 016746 000036          MOV  SNULL,-(SP) ;GET # OF FILLER CHARS. NEEDED
(1) ;*AND THE NULL CHAR.
(1) 001054 105366 000001 60:     DECB 1(SP)   ;DOES A NULL NEED TO BE TYPED?
(1) 001060 002770          BLT  50      ;BR IF NO--GO POP THE NULL OFF OF STACK
(1) 001062 004767 000002          JSR  PC,78   ;GO TYPE A NULL
(1) 001066 000772          BR  60      ;LOOP
(1) 001070 105777 000012 70:     TSTB 0STPS  ;WAIT UNTIL PRINTER IS READY
(1) 001074 100375          BPL  70      ;
(1) 001076 116677 000002 000004  MOVB 2(SP),0STPS ;LOAD CHAR TO BE TYPED INTO DATA REG.
(1) 001104 000207          RTS  PC
(1) ;*IF NDF STPS,STPS:
(1) 001106 177564          ;*IF EQ .-STPS,STPS: .WORD 177564 ;ITY PRINTER STATUS REG. ADDRESS
(1) ;*IF NDF STPB,STPB:
(1) 001110 177566          ;*IF EQ .-STPB,STPB: .WORD 177566 ;ITY PRINTER BUFFER REG. ADDRESS
(1) ;*IF NDF SNULL,SNULL:
(1) 001112 000          ;*IF EQ .-SNULL,SNULL: .BYTE 0 ;CONTAINS NULL CHARACTER FOR FILLS
(1) ;*IF NDF SFILLS,SFILLS:
(1) 001113 002          ;*IF EQ .-SFILLS,SFILLS: .BYTE 2 ;CONTAINS # OF FILLER CHARACTERS
(1) ;*IF NDF SFILLC,SFILLC:
(1) 001114 012          ;*IF EQ .-SFILLC,SFILLC: .BYTE 12 ;INSERT FILL CHARS. AFTER A *LIN
(1) ;*IF NDF STPFLG,STPFLG:
(1) 001115 000          ;*IF EQ .-STPFLG,STPFLG: .BYTE 0 ;"TERMINAL AVAILABLE" FLAG (BIT#
  
```

```

233          001          .IF B
(2)          .IF .....
(2)          .NLIST
(2)          .HEPT
(2)          .LIST
(2)          .NLIST
(2)          .ENDR
(2)          .LIST
(2)          .ENDC
(1)          000
(1)
(1)          .BTTL 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
(1) 001116 010046 .IF B
(1)          STRAP: MOV     RB,-(SP)          ;SAVE RB
(1)          .IF
(1)          STRAP: MOV     2(SP),00PS      ;ASSUME STATUS OF CALLER
(1)          MOV     RB,-(SP)          ;SAVE RB
(1)          .ENDC
(1)          000
(1) 001120 016600 000002 .IF B
(1) 001124 005740
(1) 001126 111000
(1)          001
(1)          .IF NB
(1)          CMP     $TERM,RB          ;CHECK FOR OUT OF BOUNDS
(1)          BGT     .+6              ;BR IF OK
(1)          HALT   ;OUT OF BOUNDS
(1)          BR     .-2              ;HANGUP
(1)          .ENDC
(1)          000
(1) 001130 016000 001136 .IF B
(1) 001134 000200
(1)          .MACRO SETTRAP A,B,MSG
(1)          SSSET  A,B,\<TRAP+STRP>,\STRP,<MSG>
(1)          .NLIST
(1)          STRP=STRP+2
(1)          .LIST
(1)          .ENDM SETTRAP
(1)          .MACRO SSSET A,B,C,D,CUMNT
(1)          .IF EQ STRP
(1)
(1)          .BTTL TRAP TABLE
(1)
(1)          ;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
(1)          ;BY THE "TRAP" INSTRUCTION.
(1)
(1)          ; ROUTINE
(1)          ; -----
(1)          $IRPAD:

```



```

(1) .ENDC
(1) .IF NDF GNS,.NLIST
(1)     A= C
(1) .IF NDF GNS,.LIST
(1)     B          ICALL=A TRAP+D(C)      COMNT
(1) .ENDM  SSSET
(1) .MACRO TRMTRP
(1) STRM=,STRPAD
(1) .ENDM  TRMTRP
(1) .IF DF STYPE
(1)     001
(3)     002
(3)
(3) .BTTL  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 STRPAD:
(3)     001
(3) 001136 001000 .ENDC
(1)     000
(1)     001
(1) .ENDC
(1) .IF DF STYPOC
(1)     SETTRAP TYPOC,STYPOC,"/TYPE OCTAL NUMBER (WITH LEADING ZEROS)/
(1)     SETTRAP TYPOS,STYPOS,"/TYPE OCTAL NUMBER (NO LEADING ZEROS)/
(1)     SETTRAP TYPON,STYPON,"/TYPE OCTAL NUMBER (AS PER LAST CALL)/
(1) .ENDC
(1) .IF DF STYPDS
(1)     SETTRAP TYPDS,STYPDS,"/TYPE DECIMAL NUMBER (WITH SIGN)/
(1) .ENDC
(1) .IF DF STYPBN
(1)     SETTRAP TYPBN,STYPBN,"/TYPE BINARY (ASCII) NUMBER/
(1) .ENDC
(1) .IF DF SRDCHR
(1)     SETTRAP RDCHR,SRDCHR,"/TTY TYPEIN CHARACTER ROUTINE/
(1) .ENDC
(1) .IF DF SRDLIN
(1)     SETTRAP RDLIN,SRDLIN,"/TTY TYPEIN STRING ROUTINE/
(1) .ENDC
(1) .IF DF SRDOCT
(1)     SETTRAP ROOCT,SRDOCT,"/READ AN OCTAL NUMBER FROM TTY/
(1) .ENDC
(1) .IF DF SRDDEC
(1)     SETTRAP RODEC,SRDDEC,"/READ A DECIMAL NUMBER FROM TTY/
(1) .ENDC
(1) .IF DF SSAVEG
(1)     SETTRAP SAVREG,SSAVEG,"/SAVE R0-R5 ROUTINE/
(1)     SETTRAP RESREG,SRRESG,"/RESTORE R0-R5 ROUTINE/
(1) .ENDC
(1)     000
234 001140 173000 RUMADD: 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	IMAG1:	IMAGI				
246	001170	004100		IMAGA				
247	001172	004200		IMAGO				
248	001174	004300		IMAGC				
249	001176	004400		IMAGO				
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	IMAG0:	IMAGA				
281	001272	173000	RUNADD:	173000				
282	001274	000000	TEMP:	0				
283	001276	000000	ICNT:	0				
284	001300	001550	PRSTAB:	PRG0				
285	001302	002324		PRG1				
286	001304	002426		PRG2				
287	001306	012706	STARTZ:	MOV	0STKPTR,X0	1SET STACK PTR		
288	001312	013746		MOV	004,-(X6)			
289	001316	012737		MOV	0XORA,004			
290	001324	012737	000000	177060	MASK:	MOV	00,00177000	100 IS MODIFIED BY THE SELECTION AND SET UP OF IMAGES
291	001332	012637	000004		MOV	(X6)+,004		



292	001336	012737	177777	001354	MOV	0-1,00XORPLG	
293	001344	104400			TYPE		
294	001346	003151			M3		
295	001350	000137	001536		JMP	00RESTART	
296							
297	001354	000000			XORFLG:	,WORD	0
298							
299	001356	022626			XORAI	CMF	(%6)+, (%6)+
300	001360	012637	000004		MOV	(%6)+, 004	
301	001364	005037	001354		CLR	00XORFLG	
302	001370	000137	001536		JMP	00RESTART	
303							
304	001374	012706	000500		PHNTRSI	MOV	00STRPTR, %0
305	001400	005067	000010		CLR	PRGNUM	ISSET STACK PTR
306	001404	104400			TYPE		
307	001406	003077			M6		
308	001410	004567	001112		JSR	5, RECD	IRECEIVE DATA AND PUT
309	001414	000000			PRGNUM:	0	IIT HERE
310	001416	104400			TYPE		
311	001420	003127			M1		
312	001422	004567	001100		JSR	%5, RECD	IGET IMAGE AND PT
313	001426	000000			IMIG:	,WORD	IIT HERE
314	001430	000241			CLC		
315	001432	006137	001426		ROL	00IMIG	IADJUST FOR AN EVEN ADDRESS
316	001436	013700	001426		MOV	00IMIG, %0	
317	001442	062700	001166		ADD	00IMAG1, %0	
318	001446	011037	001270		MOV	(%0), 00IMAG0	ISELECT IMAGE ADDRESS
319	001452	013700	001426		MOV	00IMIG, %0	
320	001456	062700	001214		ADD	0LENGTH, %0	
321	001462	011037	001266		MOV	(%0), 00WONDS	ISELECT NUMBER OF WORDS IN ROM
322	001466	013700	001426		MOV	00IMIG, %0	
323	001472	062700	001140		ADD	0ROMADD, %0	
324	001476	011037	001272		MOV	(%0), 00ROMADD	ISELECT STARTING ADDRESS OF ROM
325	001502	013700	001426		MOV	00IMIG, %0	
326	001506	062700	001242		ADD	0BLKBOX, %0	
327	001512	011067	177610		MOV	(%0), MASK*2	ISSET 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	00STARTZ	
333							
334	001536	016700	177652		RESTART:	MOV	PRGNUM, %0
335	001542	006300			ASL	%0	IShift PROGRAM 0
336	001544	000170	001300		JMP	0PRGTAB(0)	IGO TO PROGRAM
337							

```

339                                     ;PROGRAM B LOGIC TESTS
340 001550 005067 177522                PRG01 CLR ICNT ;CLEAR PASS COUNT
341 001554 012767 001570 001134      PRG01: MOV BT1, LASTPL ;SET RETURN ADDRESS FOR SCOPE
342 001562 016737 177510 177570      MOV ICNT, 00DI0PLAY ;DISPLAY PASS COUNT
343
344                                     ;TEST1 TEST ABILITY TO REFERENCE ROM WITHOUT TIMING OUT
345 001570 012706 000500                T11: MOV 08TKPTR, SP ;SET STACK PTR
346 001574 016700 177472                MOV ROMADD, X0 ;GET ROM ADDRESS
347 001600 016701 177462                MOV WORDS, X1 ;GET ADDRESS COUNTER
348 001604 012767 001650 176172      T1A1: MOV 0ERR01, 4 ;SET UP TIME OUT VECTOR
349 001612 011003                       MOV (0), X3 ;REFERENCE
350 001614 005737 001354                TST 00XORFLG
351 001620 100006                       BPL A18
352 001622 012737 001656 000004      MOV 0ERR2, 004
353 001630 012737 000000 177060      MASK11: MOV 00, 00177000
354 001636 062700 000002              A181: ADD 02, X0 ;INCREMENT POINTEN
355 001642 005301                       DEC X1 ;DECREMENT ADDRESS COUNTER
356 001644 001357                       BNE T1A ;BRANCH IF NOT FINISHED
357 001646 000406                       BR T10 ;GO TO SCOPE LOOP
358 001650 022626                       ERROR11: CMP (6)+, (6)+ ;REPOSITION STACK
359 001652 104000                       ERROR ;ERROR! ROM TIMED OUT WHEN REFERENCED
360                                     ;ADDRESS IS IN R0
361 001654 000753                       BR T1A ;LOOP ON ERROR
362
363 001656 022626                       ERR21: CMP (X6)+, (X6)+
364 001660 000167 177720                JMP T1A
365
366 001664 000004                       T101: SCOPE
367
368                                     ;TEST2 TEST THAT ROM DATA CAN BE READ RELIABLY.
369 001666 016700 177400                T21: MOV ROMADD, X0 ;GET ROM ADDRESS
370 001672 016701 177370                MOV WORDS, X1 ;GET ADDRESS COUNTER
371 001676 012767 000006 176100      MOV 06, 4 ;INITIALIZE TIME OUT VECTOR
372 001704 005067 177364                T4A1: CLR TEMP ;INITIALIZE TEMP
373 001710 011003                       MOV (0), X3 ;GET DATA
374 001712 062067 177356                ADD (0)+, TEMP ;ADD DATA TO TEMP
375 001716 106703 177352                SUB TEMP, X3 ;SUBTRACT DATA FROM DATA
376 001722 001403                       BEQ T20 ;BRANCH IF EQUAL
377 001724 104000                       ERROR21: ERROR ;DATA ERROR
378 001726 005740                       TST -(X0)
379 001730 000765                       BR T2A ;LOOP ON ERROR
380 001732 044067 177336                T4B1: BIC -(0), TEMP ;CLEAR TEMP BITS
381 001736 001403                       BEQ T2C ;BRANCH IF EQUAL TO 0
382 001740 104000                       ERROR ;DATA ERROR
383 001742 005720                       TST (X0)+
384 001744 000772                       BR T2B ;LOOP ON ERROR
385 001746 021010                       T4C1: CMP (0), (0) ;COMPARE DATA
386 001750 001402                       BEQ T2D ;BRANCH IF EQUAL
387 001752 104000                       ERROR ;DATA ERROR
388 001754 000774                       BR T2C ;LOOP ON ERROR
389 001756 122040                       T4D1: CMPB (0)+, -(0) ;COMPARE DATA (BYTE OPERATION)
390 001760 001402                       BEQ T2E ;BRANCH IF EQUAL
391 001762 104000                       ERROR ;DATA ERROR
392 001764 000774                       BR T2D ;LOOP ON ERROR

```





447	002202	010367	000566		MOV	X3,D2BTYP	I TYPE
448	002206	004767	000564		JSR	7,02A	I IMAGE ADDRESS
449	002212	104400			TYPE		
450	002214	003212			M12		I SEPARATOR
451	002216	011367	000552		MOV	(3),D2BTYP	I TYPE
452	002222	004767	000550		JSR	7,02A	I IMAGE DATA
453	002226	104400			TYPE		
454	002230	003124			M8		I CR/LF
455	002232	005737	177570		TST	008HR	
456	002236	100334			BPL	T4C	
457	002240	000000			HALT		
458	002242	000732			BR	T4C	I GO TO T4C
459	002244	000004			T4E1	SCOPE	
460							
461	002246	005267	177024		END1	INC	I INCREMENT PASS COUNT
462	002252	026727	177020	000100		ICNT	
463	002260	001402			CMR	ICNT,#100	
464	002262	000167	177266		BEG	DONE	
465	002266	012737	000207	177566	JMP	PRGR	I GO RESTART PROGRAM
466	002274	105737	177564		DUNE1	0207,00TPB	I RING THE BELL
467	002300	100375			MOV	00TPB	
468	002302	013700	000042		TSTB	,-4	
469	002306	001404			BPL	0042,X0	I RETURN TO DECTAPE MONITOR?
470	002310	004710			MOV	DONE1	
471	002312	000240			BEG	DONE1	
472	002314	000240			LUGIC1	7,(0)	I RETURN!
473	002316	000240			JSR		
474	002320	000167	177224		NOP		
475					NOP		
					DUNE11	JMP	PRGR



```

477                                     /THIS PROGRAM TYPES OUT RUN DATA
478
479 002324 012706 000500  PPG1:  MOV  #STKPTR,X0  /INITIALIZE STACK
480 002330 104400                TYPE
481 002332 003107                M7  /FROM DATA
482 002334 016701 176726  PPG1A: MOV  WORDS,X1  /GET # OF WORDS
483 002340 016700 176726  MOV  ROMADD,X0  /GET STARTING ADDRESS
484 002344 012702 000012  MOV  @12,X2  /GET ADDRESS INDICATOR
485 002350 010067 000420  PPG1B: MOV  X0,D2BTYP  /GET ADDRESS
486 002354 004767 000416  JSR  7,02A  /AND TYPE IT
487 002360 104400                TYPE
488 002362 003124                M8  /CR/LF
489 002364 012067 000404  PPG1C: MOV  (0)+,D2BTYP  /TYPE
490 002370 004767 000402  JSR  7,02A  /DATA
491 002374 104400                TYPE
492 002376 003210                M11
493 002400 005301                DEC  X1  /ALL DATA TYPED
494 002402 001407                BEQ  PRG10  /GO TO FINISH
495 002404 005302                DEC  X2
496 002406 001366                BNE  PRG1C  /RETURN TO PRG1B
497 002410 012702 000012  MOV  @12,X2  /GET ADDRESS INDICATOR
498 002414 104400                TYPE
499 002416 003124                M8  /CR/LF
500 002420 000753                BR   PRG1B  /RETURN TO PRG1B
501 002422 000167 176746  PPG1D: JMP  PRMTRS  /GO GET NEXT TEST
502
503                                     /THIS PROGRAM CYCLES A SINGLE ADDRESS (ADDRESS MUST BE EVEN) TO CHANGE
504                                     /THE ADDRESS TYPE NEW ADDRESS ON THE TTY.
505 002426 012706 000500  PPG2:  MOV  #STKPTR,X0  /INITIALIZE STACK POINTER
506 002432 012737 002516 000004  MOV  #PRG2C,000  /LOAD TRAP ERROR VECTOR
507 002440 005067 175332  CLR  PSW  /CLEAR PROCESSOR STATUS
508 002444 012737 002476 000060  MOV  #PRG2A,00TKVEC  /LOAD KEYBOARD INTERRUPT VECTOR
509 002452 012737 000340 000062  MOV  #340,00TKVEC+2  /LOAD KEYBOARD PRIORITY
510 002460 012737 000100 177560  MOV  #100,00TK0  /SET INTERRUPT ENABLE BIT
511 002466 016700 176600  MOV  ROMADD,X0  /GET ROM ADDRESS
512 002472 005710  PPG21: TST  (0)  /READ ROM ADDRESS
513 002474 000776  BR   =-2  /LOOP
514 002476 004567 000024  PPG2A: JSR  5,RECD  /GO GET ADDRESS &
515 002502 000000  PPG2B: 0  /PUT IT HERE
516 002504 016700 177772  MOV  PRG2B,X0
517 002510 104400                TYPE
518 002512 003124                M8  /CR/LF
519 002514 000002  RTI  /EXIT KEYBOARD INTERRUPT SERVICE
520 002516 104000  PPG2C: ERROR  /ERROR! DID YOU TYPE AN ODD ADDRESS?
521 002520 000001  WAIT
522 002522 000167 177744  JMP  PRG21

```





578	002744	003070		ERRORM		I'PC=
579	002746	011667	000022	MOV	(6),D2BTYP	I'YPE PROGRAM COUNTER
580	002752	004767	000020	JSR	7,02A	
581	002756	005737	177570	TST	003WR	I'HALT ON ERROR?
582	002762	100001		BPL	.+4	I
583	002764	000000		HALT		I'YES HALT
584	002766	104400		TYPE		
585	002770	003124		MO		
586	002772	000002		RTI		I'RETURN TO TEST
587						

589									
590	002774	000000							
591	002776	010246							
592	003000	010146							
593	003002	010046							
594	003004	016700	177764						
595	003010	012701	000006						
596	003014	005002							
597	003016	006100							
598	003020	006102							
599	003022	002702	000260						
600	003026	010267	177742						
601	003032	104400							
602	003034	002774							
603	003036	005002							
604	003040	006100							
605	003042	006102							
606	003044	006100							
607	003046	006102							
608	003050	006100							
609	003052	006102							
610	003054	005301							
611	003056	001361							
612	003060	012600							
613	003062	012601							
614	003064	012602							
615	003066	000207							

```

;THIS ROUTINE CONVERTS AN OCTAL NUMBER TO ASCII AND TYPES IT ON THE TTY.
D2BTYP: 0
O2A:  MOV      X2,-(6)           ;SAVE R2
      MOV      X1,-(6)           ;SAVE R1
      MOV      X0,-(6)           ;SAVE R0
      MOV      D2BTYP,X0        ;GET DATA TO BE TYPED
      MOV      06,X1            ;GET COUNTER
      CLR      X2                ;CLEAR WORKING REGISTER
      ROL      X0                ;MOV FIRST BIT (MSB) INTO
      ROL      X2                ;R2
      ADD      0260,X2          ;FORM ASCII CODE
      MOV      R2,D2BTYP
      TYPE
      D2BTYP
      CLR      X2                ;CLEAR WORKING REGISTER
      ROL      X0                ;ROTATE THE
      ROL      X2                ;NEXT
      ROL      X0                ;OCTAL CHARACTER
      ROL      X2                ;INTO
      ROL      X0                ;REGISTER
      ROL      X2                ;THO
      DEC      X1                ;DECREMENT COUNTER
      BNE      18                ;GO TO 18 IF NOT 0
      MOV      (6)+,X0          ;FINISHED, RESTORE REGISTERS
      MOV      (6)+,X1
      MOV      (6)+,X2
      RTS      7                ;AND EXIT
  
```



Line	Address	Value	Address	Value	Label	Content
617					ASCII MESSAGES	
618	003070	005015	041520	020075	ENRORM:	.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>'MOM DATA'<15><12>
	003114	042040	052101	006501		
	003122	000012				
621	003124	005015	000		M0:	.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	M3:	.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				.03776
629	003776	000000				.WORD
630	004000	177777	177777	177777	IMAGI:	-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	IMAGI:	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

646	004166	001366							
	004170	105222	000142	177560			105222,000142,177560,177550		
	004176	177550							
647	004200	013701	177570	000005	IMAG01		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	IMAG01		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	IMAG01		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			176716,000005,010200,005720		
	004466	005720							
670	004470	012001	005311	005720			012001,005311,005720,012041		
	004476	012041							
671	004500	031011	001776	005720	IMAG01		031011,001776,005720,031041	IMAGE 5	
	004506	031041							
672	004510	001406	000112	173100			001406,000112,173100,000340		



673	004516	000340					
	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	IMAG6:	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	IMAG7:	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	IMAG8:	012701,160000,012702,000006	IMAGE 8
	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	

701	005044	005214					
	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	IMAGL:	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			ENDAD: 0		
715		000001			.END		





IMAGL	005076	254	7050																		
IMAG0	001270	2000	3100	429																	
IMAG1	001166	2450	317																		
IMIG	001426	3130	3150	316	319	322	325														
IOTVEC	000020	2010																			
ITCNT	002714	5590	560	5620	5670																
LASTPC	002716	3410	5630	564	5680																
LENGTH	001214	2560	320																		
LOGIC	002310	210	4700																		
MASK	001324	2900	3270																		
MASK1	001630	3200	3930																		
MASK2	002636	3290	5550																		
M1	003127	311	6220																		
M10	003206	442	6240																		
M11	003210	492	6250																		
M12	003212	450	6260																		
M3	003151	294	6230																		
M6	003077	307	6190																		
M7	003107	401	6200																		
M8	003124	331	446	454	488	499	510	537	585	6210											
O2A	002776	440	444	448	452	486	498	500	5910												
PC	X000007	2010	2320																		
PIRO	177772	2010																			
PIRQVE	000240	2010																			
PRGNUM	001414	3050	3090	334																	
PRGTAB	001300	2040	336																		
PRG0	001550	204	3400	474																	
PRG0R	001554	3410	464																		
PRG1	002324	205	4790																		
PRG1A	002340	4030																			
PRG1B	002350	4050	500																		
PRG1C	002364	4090	496																		
PRG1D	002422	494	5010																		
PRG2	002426	206	5050																		
PRG2A	002476	500	5140																		
PRG2B	002502	5150	516																		
PRG2C	002516	506	5200																		
PRG21	002472	5120	522																		
PRMTRS	001374	227	3040	501																	
PR0	000000	2010																			
PR1	000040	2010																			
PR2	000100	2010																			
PR3	000140	2010																			
PR4	000200	2010																			
PR5	000240	2010																			
PR7	000340	2010																			
PS	177776	2010																			
PSW	177776	2010	5070																		
PWRVEC	000024	2010																			
RECD	002526	300	312	514	5290																
RESTAR	001536	229	295	302	3340																
RESVEC	000010	2010																			
ROMADD	001272	2010	3240	346	369	401	4140	420	483	511											
ROMADD	001140	2340	323																		



RP6	= 000300	2010																		
R0	=X000000	2010	232*	233*																
R1	=X000001	2010																		
R2	=X000002	2010	600																	
R3	=X000003	2010																		
R4	=X000004	2010																		
R5	=X000005	2010	530	539*																
R6	=X000006	2010																		
R7	=X000007	2010																		
SP	=X000006	2010	232*	233*	345*	532*	533*	534	530	540	541*	545	563	564*						
STACK	= 001100	2010																		
STARTZ	001306	2070	332																	
START1	000200	2270																		
START3	000210	2290																		
STKLMY	= 177774	2010																		
STKPTR	= 000500	2200	207	304	345	400	426	479	509											
SWR	= 177570	2010	437	455	549	557	574	561												
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																		
028	002704	550	561	564*	571															
TBITVE	= 000014	2010																		
TEMP	001274	2020	372*	374*	375	300*														
TK0	= 177562	2240	532																	
TK3	= 177560	2230	510*	530																
TKVEC	= 000060	2010	500*	509*																
TP0	= 177566	2220	465*	540*																
TP3	= 177564	2210	466																	
TPVEC	= 000064	2010																		
TRAPVE	= 000034	2010																		
TRTVEC	= 000014	2010																		





COMMEN	2010										
ENDCOM	2010										
ERROR	2010	359	377	382	387	391	405	410	415	436	520
ESCAPE	2010										
MULT	2010										
NEWTST	2010										
POP	2010										
PUSH	2010										
SCOPE	2010	366	396	422	459						
SETTRA	2330										
SETUP	2010										
SKIP	2010										
SLASH	2010										
SPACE	2010										
STARS	2010	232	233								
TRMTRP	2330										
TYPBIN	2010										
TYPDEC	2010										
TYPNUM	2010										
TYPOCS	2010										
TYPOCT	2010										
TYPTXT	2010										
SSESCA	2010										
SSNEWT	2010										
SSSET	2330										
SSSKIP	2010										
.EQUAT	2000	201									
.STRAP	2000	233									
.STYPE	2000	232									

ADD	232	317	320	323	326	354	374	599							
ASL	335	542	543	544											
BEG	376	381	386	390	433	463	469	494	575						
BGT	561														
BIC	380	533	541												
BIS	545														
BIT	437	549	557	574											
BLT	232														
BNE	232	356	395	420	431	438	496	535	550	550	611				
BPL	232	351	456	467	531	552	582								
BR	232	357	361	379	384	388	392	406	411	416	435	458	500	513	546
	571														
CLC	314														
CLR	301	305	340	372	414	507	529	562	596	603					
CMP	299	358	363	385	400	413	418	430	434	462	538	560	569		
CMPB	232	389	534												
DEC	355	394	419	432	493	495	610								
DECB	232														
EMT	201														
HALT	207	232	457	503											
INC	409	461	559												
IOT	201														
JMP	227	229	295	302	332	336	364	464	474	501	522				
JSR	232	300	312	440	444	448	452	470	486	490	514	580			
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	610	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	200	233											
.MACRO	233														
.MCALL	200	201													
.NLIST	193	201	202	233											
.PAGE	338	523	588												



.MAIN. MACY11 27(657) 6-JUN-75 15144 PAGE 7-10  
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: 10K