

IDENTIFICATION

PRODUCT CODE DEC-11-UABLB-A-LA
PRODUCT NAME LISTING OF NON-SWITCH REGISTER
 PDP-11 ABSOLUTE LOADER
DATE CREATED JUNE 1975
MAINTAINER 8/11 SMALL SOFTWARE ENGINEERING

COPYRIGHT © 1975

DIGITAL EQUIPMENT CORPORATION

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
55
56
57

```

;
; PDP-11 ABSOLUTE BINARY LOADER == VB07.00
;
;
; DEC-11=UABLB=A=LA
;
; COPYRIGHT 1975
; DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS
;
; THIS SOFTWARE IS FURNISHED UNDER A 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 DIGITAL EQUIPMENT CORPORATION.
;
; 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 THE USE
; OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED
; BY DIGITAL EQUIPMENT CORPORATION
;
; INPUT FORMAT ==
; FRAME =1 001
;        =2 000
;        =3 BYTE COUNT = LOWER ORDER
;        =4 BYTE COUNT = HIGHER ORDER
;        =5 LOAD ADDRESS = LOWER ORDER
;        =6 LOAD ADDRESS = HIGHER ORDER
;
;           , DATA
;           , PLACED
;           , HERE
; CKSM = LAST FRAME CONTAINS THE CHECKSUM
;
; IF THE BYTE COUNT IS EQUAL TO 6, THE LOAD ADDRESS
; SPECIFIED WILL BE CONSIDERED TO BE THE DESIRED JUMP
; ADDRESS, IF THIS ADDRESS IS ODD, THE LOADER WILL HALT,
;
;
; IF THE BYTE COUNT IS > 6, DATA WILL BE LOADED INTO MEMORY,
;
;
; STORAGE REQUIRED = 75 WORDS, REGISTEREDS USED = R1,R2,R3,R4,R5,R6,R7,
;
; PROGRAMMING CONSIDERATIONS AND CAUTIONS = TWO WORDS IMMEDIATELY
; PRECEDING L.DEV ARE USED FOR THE LOADER SP STACK,
;
;
; LOADING PROCEDURES

```

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
107
108
109
110
111
112
113
114

```

; 1. LOAD THE ABSOLUTE LOADER TAPE BY TAPING
; PR<CR> OR TT<CR>
;
; 2. PLACE THE ABSOLUTE TAPE FORMAT IN THE READER
;
; 3. L XXX516<CR> ;ADDRESS OF SOFTWARE SWITCH REGISTER
;
; 4. D YYYYYY<CR> ;DEPOSIT RELOCATION VALUE
;
; 5. L XXX500<CR> ;STARTING ADDRESS OF ABSLDR
;
; 6. S <CR> ;START LOADER
;
; WHERE XXX IS DEPENDENT ON MEMORY SIZE
; YYYYYY IS DEPENDENT ON TYPE OF LOAD
;
;
; L.CKSM = %0
; L.ADR = %1
; L.BC = %2
; L.BYT = %3
; R4 = %4
; L.PTR = %5
; SP = %6
; PC = %7 ;PROGRAM COUNTER
;
;
; .ASECT
; LOAD =17400 ;BOOTSTRAP FORMATED TAPES MAY
; .LOAD+75 ;NOT BE LOADED BELOW THIS ADDRESS
; .BYTE 75
; L.DEV = DEVICE ;DEVICE ADDRESS IN BOOT LOADER
; L.LOAD: HALT
;
;
; ; START OF LOADER
;
; L.LD1: MOV PC,SP ;SET UP STACK
; CMP =(SP),=(SP) ; TO START AT L.LD1=2
; MOV PC,L.PTR ;GET RELOCATED
; ADD #L.READ=.,L.PTR ; START ADDRESS OF READ ROUTINE
; CLR L.ADR ;CLEAR THE ROAD ADDRESS
; L.LD1B: MOV (PC)+,(SP) ;PICK UP THE CONTENT OF
; L.SR: ;THE SOFTWARE SWITCH REGISTER
; ROR #SP ;CHECK RELOCATION FACTOR
; RCS L.LD1C ;JUMP IF SOME RELOCATION NEEDED
; CLR #SP ;USE ADDRESS SPECIFIED ON THE TAPE
; BR L.LD2 ;GO DO LOAD
; L.LD1C: ASL #SP ;CHECK FOR NON=ZERO
; BNE L.LD2 ;JUMP IF LOAD ADDRESS SPECIFIED
; MOV L.ADR,#SP ;OTHERWISE CONTINUE LOADING FROM LAST LOAD
;
;
; ; LOOK FOR THE BEGINNING OF A BLOCK
; L.LD2: CLR L.CKSM ;INITIALIZE CHECKSUM
; JSR PC,L.PTR ;READ A FRAME
; DECR L.BYT ;CHECK FOR +1 (START OF A BLOCK)

```

```

115 017544 001374      BNE L,LD2      ;LOOP UNTIL +1 IS FOUND
116 017546 004715      JSR  PC,0L,PTR  ;READ ANOTHER FRAME
117
118 ; INPUT AND SAVE BYTE COUNT, IF BYTE COUNT IS EQUAL TO 6
119 ; GO TO PROCEED JUMP
120
121 017550 004767 000074      JSR  PC,L,GWRD  ;GET FULL BYTE COUNT
122 017554 010402      MOV  R4,L,BC    ;
123 017556 162702 000004      SUB  #4,L,BC    ;SUBTRACT 4 TO MAKE BYTE COUNT CORRECT
124 017562 022702 000002      CMP  #2,L,BC    ;HAS BYTE COUNT EQUAL TO 6?
125 017566 001441      BEQ  L,JMP      ;JUMP IF NO DATA (E.G. = JUMP BLOCK)
126 017570 004767 000054      JSR  PC,L,GWRD  ;GET LOAD ADDRESS
127 017574 061604      ADD  @SP,R4     ;GENERATE ACTUAL ADDRESS
128 017576 010401      MOV  R4,L,ADR   ;AND PUT IT INTO THE PROPER CELL
129
130 ; READ IN REMAINDER OF DATA
131 ; IF THE LOADER HALTS AT L,BAD, A CHECKSUM ERROR
132 ; HAS OCCURED, R3 WILL CONTAIN THE EXPECTED CHECKSUM,
133 ; AND R0 WILL CONTAIN THE DEVIATION FROM THE EXPECTED
134 ; CHECKSUM,
135
136 017600 004715      L,LD3: JSR  PC,0L,PTR  ;READ A FRAME
137 017602 002004      BGE  L,LD4      ;BRANCH IF MORE DATA REMAINS
138 017604 105700      TSTB L,CKSM    ;IF CHECKSUM IS
139 017606 001753      BEQ  L,LD2      ;CORRECT, THEN CONTINUE
140 017610 000000      L,BAD: HALT    ;CHECKSUM ERROR
141 017612 000751      RR   L,LD2      ;PRESS CONTINUE TO IGNORE CHECKSUM
142 017614 110321      L,LD4: MOVB L,BYT,(L,ADR)+ ;STORE 8 BITS AT A TIME
143 017616 000770      BR   L,LD3      ; THE RE-LOOP
144
145 ; INPUT A FRAME, DECREMENT BYTE COUNT, AND ACCUMULATE CHECKSUM
146
147 017620 016703 000152      L,READ: MOV  L,DEV,L,BYT ;DEVICE ADDRESS TO L,BYT
148 017624 105213      INCR @L,BYT    ;SELECT READER
149 017626 105713      L,R1: TSTB @L,BYT ;DONE ?
150 017630 100376      BPL  L,R1      ;NO
151 017632 116303 000002      MOVB 2(L,BYT),L,BYT ;GET CHARACTER
152 017636 060300      ADD  L,BYT,L,CKSM ;ADD TO CHECKSUM
153 017640 042703 177400      BIC  #177400,L,BYT ;MASK OFF JUNK
154 017644 005302      DEC  L,RC      ;DECREMENT BYTE COUNT BY ONE
155 017646 000207      RTS  PC
156
157 ; ASSEMBLE ONE FULL WORD OF DATA
158
159 017650 012667 000046      L,GWRD: MOV  (SP)+,L,TMP ;SAVE RETURN IN TEMPORY
160 017654 004715      JSR  PC,0L,PTR  ;GET ONE CHARACTER
161 017656 010304      MOV  L,BYT,R4   ;SAVE R3 IN TEMPORARY
162 017660 004715      JSR  PC,0L,PTR  ;GET ANOTHER FRAME
163 017662 000303      SWAB L,BYT     ;PLACE ANOTHER FRAME
164 017664 050304      BIS  L,BYT,R4   ;ASSEMBLE BOTH FRAMES INTO A COMPLETE WORD
165 017666 016707 000030      MOV  L,TMP,PC   ;RETURN
166
167 ; CHECK CORRECTNESS OF JUMP ADDRESS
168 ; HALT IF ADDRESS IS ODD, JUMP TO PROGRAM IF ADDRESS IS EVEN
169
170 017672 004767 177752      L,JMP: JSR  PC,L,GWRD ;GET POSSIBLE TRANSFER ADDRESS
171 017676 004715      JSR  PC,0L,PTR  ;GET CHECKSUM

```

```

172 017700 105700      TSTB L,CKSM    ;IF INCORRECT
173 017702 001342      BNE  L,BAD     ; GO TO CHECKSUM HALT ADDRESS
174 017704 006204      ASR  R4        ;GET LOW ORDER BIT
175 017706 103002      BCC  L,JMP1    ;SKIP IF ADDRESS IS EVEN
176 017710 000000      HALT          ;OTHERWISE HALT
177 017712 000700      BR   L,LD18   ;RETURN TO START OF LOADING LOOP
178 017714 006304      L,JMP1: ASL  R4 ;RESTORE REGISTER
179 017716 061604      ADD  @SP,R4
180 017720 000114      JMP  @R4      ;JUMP TO USER
181 017722 000000      L,TMP: WORD 0 ;TEMPORY TO SAVE STACK SPACE
182
183 ; INITIALIZATION TO RESTORE THE BOOTSTRAP LOADER
184
185 017724 012767 000352 000020      L,INIT: MOV  #352,LOOP+2 ;RESTORE OFFSET IN BOOTSTRAP LOADER
186 017732 012767 000765 000034      MOV  #765,BRNCH ;RESTORE "BR" AT BRNCH
187 017740 000167 177532      JMP  L,LOAD    ;GO HALT AND WAIT FOR "CONT"
188 ; OVERLAY BOOTSTRAP LOADER CODE
189 ; THE FOLLOWING CODE OVERLAY THE BOOTSTRAP LOADER
190 017744 016701 000026      START: MOV  DEVICE,R1
191 017750 012722      LOOP: MOV  (PC)+,R2
192 017752 373 ;,BYTE BRNCH=LOAD-1
193 017753 353 ;,BYTE =<BRNCH-L,INIT+2/2>
194 ;,LOAD+374
195 017774 ;BR START
196 ;,R,+2
197 017776 ;DEVICE: ;0 ADDRESS OF BOOT DEVICE" COMMAND
198 ; STATUS REGISTER IS STORED HERE BY
199 ; THE BOOTSTRAP ROM
200 ;,END

```

SYMBOL TABLE

BRNCH	017774	DEVICE	017776	LOAD	017400
LOOP	017750	L.ADR	000001	L.BAD	017610
L.BC	000002	L.PYT	000003	L.CKSM	000000
L.DEV	017776	L.GWRD	017650	L.INIT	017724
L.JMP	017672	L.JMP1	017714	L.L01	017500
L.LD1B	017514	L.LDIC	017530	L.LD2	017536
L.LD3	017604	L.LD4	017614	L.LOAD	017476
L.PTR	000005	L.READ	017620	L.R1	017626
L.SR	017516	L.TMP	017722	START	017744
.ABS.	017776				
	000000				
	000001				

ERRORS DETECTED: 0

FREE CORE: 9152. WORDS
ABSLB7,OBJ,ABSLB7,LST=ABSLB7,MAC