

IDENTIFICATION

PRODUCT CODE DEC-11-UABLA-A-LA
PRODUCT NAME LISTING OF PDP-11 ABSOLUTE LOADER
DATE CREATED JUNE 1975
MAINTAINER 8/11 SOFTWARE ENGINEERING

COPYRIGHT © 1971, 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 -- VA07.00
DEC-11-UABLA=A-LA

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
-2 002
-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,

OPERATING INSTRUCTIONS:

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. USING THE BOOT LOADER, LOAD THE ABS LOADER TAPE, THE DEVICE ADDRESS USED WILL BE LOCATED IN THE LAST WORD OF THE BOOT LOADER (USUALLY 17776). IF THIS DEVICE ADDRESS IS ZERO THE ABSOLUTE LOADER WILL NOT RUN.
2. PLACE THE ABSOLUTE LOADER LOAD ADDRESS IN THE SWITCH REGISTER AND PRESS LOAD ADDRESS.
3. PLACE THE RELOCATION FACTOR, IF ANY, IN THE SWITCH REGISTER.
4. PRESS START.
5. IF THE LOADER HALTS WHEN DONE IT IS ONLY NECESSARY TO DO STEP 4 AND PRESS CONTINUE TO READ IN ADDITIONAL TAPES.

L.CKSM = X0
L.ADR = X1
L.BC = X2
L.BYT = X3
R4 = X4
L.PTR = X5
SP = X6
PC = X7 ;PROGRAM COUNTER
L.SR = 177570 ;SWITCH REGISTER

LOAD .ASECT
=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
CLP L.ADR ;CLEAR THE ROAD ADDRESS
L.LD1B: MOV #L.SR,*SP ;PICK UP THE CONTENTS OF SR AND SAVE
ROR *SP ;CHECK RELOCATION FACTOR
BCS 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
DECB L.BYT ;CHECK FOR #1 (START OF A BLOCK)
BNE L.LD2 ;LOOP UNTIL #1 IS FOUND
JSR PC,*L.PTR ;READ ANOTHER FRAME

INPUT AND SAVE BYTE COUNT, IF BYTE COUNT IS EQUAL TO 6
GO TO PROCEED JUMP

.MAIN, MACRO RSTS/E V6.0

115 017550 004767 000074
 116 017554 010402
 117 017556 162702 000004
 118 017562 022702 000002
 119 017566 001441
 120 017570 004767 000054
 121 017574 061604
 122 017576 010401
 123
 124
 125
 126
 127
 128
 129
 130 017600 004715
 131 017602 002004
 132 017604 105700
 133 017606 001753
 134 017610 000000
 135 017612 000751
 136 017614 110321
 137 017616 000770
 138
 139
 140
 141 017620 016703 000152
 142 017624 105213
 143 017626 105713
 144 017630 100376
 145 017632 116303 000002
 146 017636 060300
 147 017640 042703 177400
 148 017644 005302
 149 017646 000207
 150
 151
 152
 153 017650 012667 000046
 154 017654 004715
 155 017656 010304
 156 017660 004715
 157 017662 000303
 158 017664 050304
 159 017666 016707 000030
 160
 161
 162
 163
 164 017672 004767 177752
 165 017676 004715
 166 017700 105700
 167 017702 001342
 168 017704 006204
 169 017706 103002
 170 017710 000000
 171 017712 000700

29-MAY-75 15:57 PAGE 1-2

```

JSR PC,L.GWRD ;GET FULL BYTE COUNT
MOV R4,L.BC ;
SUR #4,L.BC ;SUBTRACT 4 TO MAKE BYTE COUNT CORRECT
CMP #2,L.BC ;WAS BYTE COUNT EQUAL TO 6?
BEQ L.JMP ;JUMP IF NO DATA (E.G. = JUMP BLOCK)
JSR PC,L.GWRD ;GET LOAD ADDRESS
ADD #SP,R4 ;GENERATE ACTUAL ADDRESS
MOV R4,L.ADR ;AND PUT IT INTO THE PROPER CELL
;
; READ IN REMAINDER OF DATA
; IF THE LOADER HALTS AT L.BAD, A CHECKSUM ERROR
; HAS OCCURRED, R3 WILL CONTAIN THE EXPECTED CHECKSUM,
; AND R0 WILL CONTAIN THE DEVIATION FROM THE EXPECTED
; CHECKSUM,
;
L.LD3: JSR PC,#L.PTR ;READ A FRAME
RGE L.LD4 ;BRANCH IF MORE DATA REMAINS
TSTB L.CKSM ;IF CHECKSUM IS
REQ L.LD2 ;CORRECT, THEN CONTINUE
L.BAD: HALT ;CHECKSUM ERROR
BR L.LD2 ;PRESS CONTINUE TO IGNORE CHECKSUM
L.LD4: MOV# L.BY,(L.ADR)+ ;STORE 8 BITS AT A TIME
BR L.LD3 ; THE RE-LOOP
;
; INPUT A FRAME, DECREMENT BYTE COUNT, AND ACCUMULATE CHECKSUM
;
L.READ: MOV L.DEV,L.BYT ;DEVICE ADDRESS TO L.BYT
INCB #L,BYT ;SELECT READER
L.R1: TSTB #L,BYT ;DONE ?
BPL L.R1 ;NO
MOV# 2(L.BYT),L.BYT ;GET CHARACTER
ADD L.BYT,L.CKSM ;ADD TO CHECKSUM
BIC #177400,L.BYT ;MASK OFF JUNK
DEC L.BC ;DECREMENT BYTE COUNT BY ONE
RTS PC
;
; ASSEMBLE ONE FULL WORD OF DATA
;
L.GWRD: MOV (SP)+,L.TMP ;SAVE RETURN IN TEMPORARY
JSR PC,#L.PTR ;GET ONE CHARACTER
MOV L.BYT,R4 ;SAVE R3 IN TEMPORARY
JSR PC,#L.PTR ;GET ANOTHER FRAME
S#AB L.BYT ;PLACE ANOTHER FRAME
RTS L.BYT,R4 ;ASSEMBLE BOTH FRAMES INTO A COMPLETE WORD
MOV L.TMP,PC ;RETURN
;
; CHECK CORRECTNESS OF JUMP ADDRESS
; HALT IF ADDRESS IS ODD, JUMP TO PROGRAM IF ADDRESS IS EVEN
;
L.JMP: JSR PC,L.GWRD ;GET POSSIBLE TRANSFER ADDRESS
JSR PC,#L.PTR ;GET CHECKSUM
TSTB L.CKSM ;IF INCORRECT
RNE L.BAD ;GO TO CHECKSUM HALT ADDRESS
ASR R4 ;GET LOW ORDER BIT
BCC L.JMP1 ;SKIP IF ADDRESS IS EVEN
HALT ;OTHERWISE HALT
BR L.LD1B ;RETURN TO START OF LOADING LOOP

```

.MAIN, MACRO RSTS/E V6.0

172 017714 006304
 173 017716 061604
 174 017720 000114
 175 017722 000000
 176
 177
 178
 179 017724 012767 000352 000020
 180 017732 012767 000765 000034
 181 017740 000167 177532
 182
 183
 184 017744 016701 000026
 185 017750 012702
 186 017752 373
 187 017753 353
 188 017774
 189 017774 017776
 190
 191 017776
 192
 193
 194 000001'

29-MAY-75 15:57 PAGE 1-3

```

L.JMP1: ASL R4 ;RESTORE REGISTER
ADD #SP,R4
JMP #R4 ;JUMP TO USER
L.TMP: .WORD 0 ;TEMPORARY TO SAVE STACK SPACE
;
;INITIALIZATION TO RESTORE THE BOOTSTRAP LOADER
;
L.INIT: MOV #352,LOOP+2 ;RESTORE OFFSET IN BOOTSTRAP LOADER
MOV #765,BRNCH ;RESTORE "BR" AT BRNCH
JMP L.LOAD ;GO HALT AND WAIT FOR "CONT"
;OVERLAY BOOTSTRAP LOADER CODE
;THE FOLLOWING CODE OVERLAY THE BOOTSTRAP LOADER
START: MOV DEVICE,R1
LOOP: MOV (PC)+,R2
;BYTE BRNCH=LOAD-1
;BYTE =4BRNCH=L.INIT+2/2>
;LOAD+374
BRNCH: JBR START
;R,+2
DEVICE: J0 ;ADDRESS OF BOOT DEVICE' COMMAND
; ;STATUS REGISTER IS STORED HERE BY
; ;THE BOOTSTRAP ROM
.END

```

SYMBOL TABLE

BRNCH	017774	DEVICE	017776
LOOP	017750	L.ADR	%000001
L.BC	%000002	L.BYT	%000003
L.DEV	017776	L.GWRD	017650
L.JMP	017672	L.JMP1	017714
L.LD1B	017514	L.LD1C	017530
L.LD3	017600	L.LD4	017614
L.PTR	%000005	L.READ	017620
L.SR	017750	L.TMP	017722
.ABS.	017776		
	000000		000
	000000		001

LOAD	=	017400
L.BAD		017610
L.CKSM	%	000000
L.INIT		017724
L.LD1		017500
L.LD2		017536
L.LOAD		017476
L.R1		017626
START		017744

ERRORS DETECTED: 0

FREE CORE: 9152, WORDS
ABSLA7.OBJ,ABSLA7.LST=ABSLA7.MAC