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

```
                        ;
                        ;
                        ; PDP-11 ABSOLUTE BINARY LOADER -- V807.00
                        ;
                        ;
                        ;       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
```



```
                        ;  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 DEPENTENT ON TYPE OF LOAD
                        ;
000000                  L.CKSM =        %0
000001                  L.ADR  =        %1
000002                  L.BC   =        %2
000003                  L.BYT  =        %3
000004                  R4     =        %4
000005                  L.PTR  =        %5
000006                  SP     =        %6
000007                  PC     =        %7                      ;PROGRAM COUNTER
                        ;
                        ;
000000                          .ASECT
017400                  LOAD    =17400                  ;BOOTSTRAP FORMATED TAPES MAY
017475                          .=LOAD+75               ;NOT BE LOADED BELOW THIS ADDRESS
017475  000075                  .BYTE 75
017776                  L.DEV   =       DEVICE          ;DEVICE ADDRESS IN BOOT LOADER
017476  000000          L.LOAD: HALT
                        ;
                        ;
                        ;
                        ; START OF LOADER
                        ;
017500  010706          L.LD1:  MOV     PC,SP           ;SET UP STACK
017502  024646                  CMP     -(SP),-(SP)     ; TO START AT L.LD1+2
017504  010705                  MOV     PC,L.PTR        ;GET RELOCATED
017506  062705  000112          ADD     #L.READ-.,L.PTR ;   START ADDRESS OF READ ROUTINE
017512  005001                  CLR     L.ADR           ;CLEAR THE ROAD ADDRESS
017514  012716          L.LD1B: MOV     (PC)+,(SP)      ;PICK UP THE CONTENT OF
017516  000000          L.SR:   0                       ;THE SOFTWARE SWITCH REGISTER
017520  006016                  ROR     @SP             ;CHECK RELOCATION FACTOR
017522  103002                  BCS     L.LD1C          ;JUMP IF SOME RELOCATION NEEDED
017524  005016                  CLR     @SP             ;USE ADDRESS SPECIFIED ON THE TAPE
017526  000403                  BR      L.LD2           ;GO DO LOAD
017530  006316          L.LD1C: ASL     @SP             ;CHECK FOR NON-ZERO
017532  001001                  BNE     L.LD2           ;JUMP IF LOAD ADDRESS SPECIFIED
017534  010116                  MOV     L.ADR,@SP       ;OTHERWISE CONTINUE LOADING FROM LAST LOAD
                        ;
                        ; LOOK FOR THE BEGINNING OF A BLOCK
017536  005000          L.LD2:  CLR     L.CKSM          ;INITIALIZE CHECKSUM
017540  004715                  JSR     PC,@L.PTR       ;READ A FRAME
017542  105303                  DECB    L.BYT           ;CHECK FOR +1 (START OF A BLOCK)
```

```
017544  001374                  BNE     L.LD2           ;LOOP UNTIL +1 IS FOUND
017546  004715                  JSR     PC,@L.PTR       ;READ ANOTHER FRAME
                                ;
                                ; INPUT AND SAVE BYTE COUNT, IF BYTE COUNT IS EQUAL TO 6
                                ;  GO TO PROCEED JUMP
                                ;
017550  004767  000074          JSR     PC,L.GWRD       ;GET FULL BYTE COUNT
017554  010402                  MOV     R4,L.BC         ;
017556  162702  000004          SUB     #4,L.BC         ;SUBTRACT 4 TO MAKE BYTE COUNT CORRECT
017562  022702  000002          CMP     #2,L.BC         ;WAS BYTE COUNT EQUAL TO 6?
017566  001441                  BEQ     L.JMP           ;JUMP IF NO DATA (E.G. = JUMP BLOCK)
017570  004767  000054          JSR     PC,L.GWRD       ;GET LOAD ADDRESS
017574  061604                  ADD     @SP,R4          ;GENERATE ACTUAL ADDRESS
017576  010401                  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 OCCURED,  R3 WILL CONTAIN THE EXPECTED CHECKSUM,
                                ;  AND R0 WILL CONTAIN THE DEVIATION FROM THE EXPECTED
                                ;  CHECKSUM,
                                ;
017600  004715                  L.LD3:  JSR     PC,@L.PTR       ;READ A FRAME
017602  002004                          BGE     L.LD4           ;BRANCH IF MORE DATA REMAINS
017604  105700                          TSTB    L.CKSM          ;IF CHECKSUM IS
017606  001753                          BEQ     L.LD2           ;CORRECT, THEN CONTINUE
017610  000000                  L.BAD:  HALT                    ;CHECKSUM ERROR
017612  000751                          BR      L.LD2           ;PRESS CONTINUE TO IGNORE CHECKSUM
017614  110321                  L.LD4:  MOVB    L.BYT,(L.ADR)+  ;STORE 8 BITS AT A TIME
017616  000770                          BR      L.LD3           ;    THE RE-LOOP
                                ;
                                ; INPUT A FRAME, DECREMENT BYTE COUNT, AND ACCUMULATE CHECKSUM
                                ;
017620  016703  000152          L.READ: MOV     L.DEV,L.BYT     ;DEVICE ADDRESS TO L.BYT
017624  105213                          INCB    @L.BYT          ;SELECT READER
017626  105713                  L.R1:   TSTB    @L.BYT          ;DONE ?
017630  100376                          BPL     L.R1            ;NO
017632  116303  000002                  MOVB    2(L.BYT),L.BYT  ;GET CHARACTER
017636  060300                          ADD     L.BYT,L.CKSM    ;ADD TO CHECKSUM
017640  042703  177400                  BIC     #177400,L.BYT   ;MASK OFF JUNK
017644  005302                          DEC     L.BC            ;DECREMENT BYTE COUNT BY ONE
017646  000207                          RTS     PC
                                ;
                                ; ASSEMBLE ONE FULL WORD OF DATA
                                ;
017650  012667  000006          L.GWRD: MOV     (SP)+,L.TMP     ;SAVE RETURN IN TEMPORY
017654  004715                          JSR     PC,@L.PTR       ;GET ONE CHARACTER
017656  010304                          MOV     L.BYT,R4        ;SAVE R3 IN TEMPORARY
017660  004715                          JSR     PC,@L.PTR       ;GET ANOTHER FRAME
017662  000303                          SWAB    L.BYT           ;PLACE ANOTHER FRAME
017664  050304                          BIS     L.BYT,R4        ;ASSEMBLE BOTH FRAMES INTO A COMPLETE WORD
017666  016707  000030                  MOV     L.TMP,PC        ;RETURN
                                ;
                                ; CHECK CORRECTNESS OF JUMP ADDRESS
                                ; HALT IF ADDRESS IS ODD, JUMP TO PROGRAM IF ADDRESS IS EVEN
                                ;
017672  004767  177752          L.JMP:  JSR     PC,L.GWRD       ;GET POSSIBLE TRANSFER ADDRESS
017676  004715                          JSR     PC,@L.PTR       ;GET CHECKSUM
```



```
017700  105700                  TSTB    L.CKSM          ;IF INCORRECT
017702  001342                  BNE     L.BAD           ; GO TO CHECKSUM HALT ADDRESS
017704  006204                  ASR     R4              ;GET LOW ORDER BIT
017706  103002                  BCC     L.JMP1          ;SKIP IF ADDRESS IS EVEN
017710  000000                  HALT                    ;OTHERWISE HALT
017712  000700                  BR      L.LD1B          ;RETURN TO START OF LOADING LOOP
017714  006304                  L.JMP1: ASL     R4              ;RESTORE REGISTER
017716  061604                          ADD     @SP,R4
017720  000114                          JMP     @R4             ;JUMP TO USER
017722  000000                  L.TMP:  .WORD   0               ;TEMPORY TO SAVE STACK SPACE
                                ;
                                ;INITIALIZATION TO RESTORE THE BOOTSTRAP LOADER
                                ;
017724  012767  000352  000020  L.INIT: MOV     #352,LOOP+2             ;RESTORE OFFSET IN BOOTSTRTAP LOADER
017732  012767  000765  000034          MOV     #765,BRNCH              ;RESTORE "BR" AT BRNCH
017740  000167  177532                  JMP     L.LOAD                  ;GO HALT AND WAIT FOR "CONT"
                                                                        ;OVERLAY BOOTSTRAP LOADER CODE
                                ;THE FOLLOWING CODE OVERLAY THE BOOTSTRAP LOADER
017744  016701  000026          START:  MOV     DEVICE,R1
017750  012702                  LOOP:   MOV     (PC)+,R2
                                        .BYTE BRNCH-LOAD+1
017753    353                           .BYTE -<BRNCH-L.INIT+2/2>
017754                                  .=LOAD+374
017774                          BRNCH:  ;BR     START
017776                                  .=.+2
017776                          DEVICE: ;0                      ADDRESS OF BOOT DEVICE' COMMAND
                                        ;                       STATUS REGISTER IS STORED HERE BY
                                        ;                       THE BOOTSTRAP ROM
000001'                                 .END
```

```
.MAIN.  MACRO RSTS/E V6.0                       29-MAY-75 16:04  PAGE 1-4
SYMBOL TABLE
BRNCH   017774          DEVICE  017776          LOAD  = 017400
LOOP    017752          L.ADR =%000001          L.BAD   017610
L.BC  =%000002          L.BYT =%000003          L.CKSM=%000000
L.DEV = 017776          L.GWRD  017650          L.INIT  017724
L.JMP   017672          L.JMP1  017714          L.LD1   017500
L.LD1B  017514          L.LD1C  017530          L.LD2   017536
L.LD3   017600          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      000
        000000      001
ERRORS DETECTED:  0

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