AAEC/E210

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT
LUCAS HEIGHTS

PDPGENER - AN IBM/360 PROGRAM TO RECONSTRUCT SYMBOLIC SOURCE LISTINGS FROM PDP-9/L OBJECT CODE

by

R.P. BACKSTROM

September 1970

ISBN 0 642 99362 9

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS

PDPGENER – AN IBM/360 PROGRAM TO RECONSTRUCT SYMBOLIC SOURCE LISTINGS FROM PDP-9/L OBJECT CODE

by

R. P. BACKSTROM

ABSTRACT

PDPGENER enables the user to generate original program listings and source decks from machine readable PDP-9/L computer object code in either card or paper tape form. The reconstructed listings contain alphanumeric labelling and complete cross reference tables. PDPGENER may also be used to update and punch new card object decks from card or paper tape input, thus eliminating slow paper tape input to the PDP-9/L. Users of other small computers may also edit object decks by specifying the object card format to PDPGENER.

National Library of Australia card number and ISBN 0 642 99362 9

CONTENTS

1. INTRODUCTION

The reconstruction of source listings and source decks from PDP-9/L object code is the principal aim of PDPGENER, now making possible the relocation of PDP-9/L programs when only the object code is available. The reconstruction is accomplished by a two-pass scan of pseudocore (an area of core set up by PDPGENER to contain an image of the PDP-9/L core). The first pass flags every location referred to from within the region of pseudocore marked for reconstruction. The second numbers these flagged locations sequentially. Whether instructions were originally defined symbolically (e.g. JMP A) or absolutely (e.g. DAC 76) does not matter. This means that cross references are available even for locations referred to absolutely – something that no Assembler would normally do.

Because PDPGENER can read object code from paper tape and also punch card object decks, it is possible to convert paper tape programs into card deck form, thus saving considerable time when loading large programs into the PDP-9/L. For example, diagnostic programs supplied by Digital Equipment Corporation, formerly taking, say, fifteen minutes to load, may now be entered in forty seconds.

PDPGENER also provides a simple means of changing or deleting existing instructions in an assembled program, merging any number of programs and even creating new programs by simulation of the PDP-9/L instructions.

In addition, PDPGENER offers programmers of other small computers all of the editing facilities available to PDP-9/L users. By specifying the column 1 codes for object and transfer cards (see Section 4.11) PDPGENER may be used to merge or change programs or to prepare input for an appropriate simulator.

These various facilities may be specified by the use of the following eleven keywords (see Section 4):

(1) READ

(2) LOAD

(3) PUNCH

(4) GENERATE, SOURCE, LABEL

(5) STORE

(6) CLEAR

(7) DUMP

(8) ENTRY

(9) TRACE

(10) EXECUTE

and (11) FORMAT

2. KEYWORDS

Each keyword must appear on a separate card and must be punched before column 72. Any operands on the card must be separated by a comma, dash, colon, equal sign or brackets, whichever is appropriate. The numeric operands for GENERATE, STORE, CLEAR and TRACE may be continued on subsequent cards by coding an asterisk in column 72 and continuing the operands in any column before column 72. Numbers, however, may not be broken at column 72. Blanks between operands and delimiters are ignored, but numbers containing embedded blanks are treated as errors. All numbers, unless otherwise stated, are treated as octal and may contain any number of digits.

PDPGENER reads all keyword, object and transfer cards in column binary mode and then examines the format of column 1. If it is not that of an object or transfer card, the 160 bytes are translated to form 80 printable characters. The first word is then compared with each of the available keywords and appropriate analysis initiated. If the first word is not a keyword, the card is treated as a comment card. A simple method for ensuring this is to code an asterisk in column 1 of all comment cards.

If PDPGENER detects an error in syntax, the Job Step is terminated by flushing the remaining keyword, object and transfer cards. This is done because a PDPGENER Job Step will normally consist of a series of tasks each dependent upon the successful completion of previous tasks. The remaining cards are flushed in column binary mode until the /* card is reached. It is necessary to ensure that no column binary cards remain to be read because these cause validity checks when read in normal EBCDIC mode by the Operating System (OS).

3. ABBREVIATIONS

In the descriptions of the use of the keywords (see Section 4), the following abbreviations are used:

(1) ℓ_1, ℓ_2 represent one, two, three or four digit octal locations,

(2) i represents an instruction consisting of from one to six octal digits,

(3) a represents an alphabetic label consisting of one, two or three characters,

(4) n represents a decimal number consisting of from one to five digits

and (5) f_1, f_2 represent one, two, three or four digit octal numbers.

4. DESCRIPTION OF USAGE

PDPGENER keyword cards may appear in any order and the operands may be specified in free format. The specific syntax requirements are set out in the following subsections.

4.1 READ ℓ_1 to $\ell_2(n)$

This indicates that an object paper tape is to be read in by the IBM/360. There are two kinds of tape to be considered here. The first, known as Read-In Mode tape, consists of a sequence of instructions to be read into successive locations beginning with the address set on the PDP-9/L console keys. The last instruction on such a tape has channel 7 of the third frame punched, indicating to the hardware that this last instruction is now to be executed. Sometimes, this instruction is a halt instruction, i.e. 740040, but usually it is a jump instruction, e.g. 600200, to start the program at location 200.

The second kind of tape has a loader at the front and this assumes control after having been loaded as a Read-In Mode tape. This loader then reads in blocks of (usually 25) instructions from the tape into various locations until it senses a 'transfer' block on the tape causing a halt or transfer of control to the newly loaded program.

The READ subroutine of PDPGENER recognises the end of the Read-In Mode tape and begins simulation of the instructions in the loader. Sufficient instructions are provided to allow all the instructions in the standard DEC loader to be simulated, and an exit from the READ subroutine occurs if an unrecognisable instruction is met. However, a message is printed, indicating which instruction it could not recognise. Similar action occurs if a halt instruction is reached. An exit from the READ subroutine may also be made after control has been given to the instruction at ℓ_2 for the nth time, provided that these parameters are specified on the READ card.

EXAMPLES: READ 200

This would cause the tape instructions to be read in starting at location 200. Only a halt or unrecognisable instruction could normally terminate the tape read.

READ 7720 TO 1000(9)

This would cause the tape instructions to be read in starting at location 7720. If this tape had a loader at the beginning, control would pass to the loader and the rest of the tape would be read in under simulation of the PDP-9/L instructions. When control is given to the instruction at location 1000 for the 9th time, processing terminates unless a halt instruction or an unrecognisable instruction is reached first.

4.2 LOAD ℓ_1

PDP-9/L object cards may be read by PDPGENER without being preceded by a LOAD card. However, if it is required to re-position the data from an object deck, the LOAD card can be used for this purpose. Notice that the program read in is not relocated, but merely loaded into a different area of pseudocore. If ℓ_1 is omitted, the object cards are read into pseudocore according to the addresses on each card. If ℓ_1 is specified, the contents of the first card are placed at locations ℓ_1, ℓ_1+1, ℓ_1+2 etc. and all subsequent object cards have their contents similarly displaced. For example, if the first card indicated location 200 and the second 250, and LOAD 1000 were specified, the first card's contents would go at locations 1000, 1001, 1002 etc., and the second card's contents at 1050, 1051, 1052 etc. Notice that LOAD is, therefore, not necessarily the same as LOAD 0. No special end card is required after the object deck. If it is a transfer card, PDPGENER notes the transfer address (overwriting previous transfer addresses) and continues processing keyword, object or further transfer cards. The operand on the LOAD card has effect only until the next non-object card.

4.3 PUNCH ℓ_1-ℓ_2,ℓ_3:

This indicates that an object deck is to be punched containing all the instructions and data currently set up in pseudocore locations ℓ_1 to ℓ_2 inclusive, and that a transfer card containing the transfer address ℓ_3 is also to be punched. These two types of operands may be repeated, or continued on successive cards by coding an asterisk in column 72. In this way, input consisting of many small decks may be repunched with interleaved transfer cards but now having complete sequential numbering throughout the resultant deck.

EXAMPLE: PUNCH 100–177,100:,300–4777,6000–6177, *

7000–7777,7000:

To punch out all currently assigned locations and to provide a transfer card with the last specified entry point, code: PUNCH ALL,: This is used, for example, when changes are to be made to an assembled program without special knowledge of the extent or entry point of the PDP-9/L program.

4.4 GENERATE, SOURCE, LABEL $=\alpha$, ℓ_1-ℓ_2

This reconstructs a source listing, including a Cross Reference Table, and also punches a source deck from the assigned locations in the region ℓ_1-ℓ_2. To indicate a disjoint region, several ranges may be specified on this card, separated by commas, and, if necessary, continued on subsequent cards by coding an asterisk in column 72. A single location may be included by coding ℓ_1 among the specified ranges. If no numeric operands appear, 0–7777 is assumed. Any or all of the options following the keyword GENERATE may be omitted, but, if present, must appear in the order given. For variety, a one, two or three letter label may be specified. If, for example, α was EDT, the generated labels would be EDT1, EDT2, EDT3 etc. The default labelling scheme is L1, L2, L3 etc. A maximum of 1023 labels is possible because the ten bits 4–13 of the associated IBM/360 word are used to hold the binary value of the label. If more than this number are required, an error message giving the number required is printed, the GENERATE option is ignored and PDPGENER processing continues. If, during the reconstruction, a memory reference instruction is found to refer to a location outside the specified region, the SOURCE option is cancelled and error messages appear in the source listing and the Cross Reference Table.

EXAMPLES: GENERATE,LABEL=ASM

This would generate a source listing without punching a source deck. The labels would be ASM1, ASM2, ASM3 etc.

GENERATE,SOURCE,0–300,6000,7000–7777

This would generate a composite source listing and source deck of all the assigned locations in the three specified regions. The labels would be L1, L2, L3 etc.

The results of any such reconstruction must, of course, be interpreted very carefully, especially if the program is to be relocated. Constants (such as masks or characters intended for transmission to the line printer buffer) having the form of memory reference instructions may have given rise to spurious references which must be eliminated. Pointers to tables and other parameter lists must be identified and replaced by appropriate symbolic addresses. And finally, the entry point to the program must be determined and punched symbolically on the generated END card.

4.5 STORE i at ℓ_1-ℓ_2

The action of this keyword is to store the instruction i at the specified locations. The numeric operands may be generalised in a manner similar to those on the GENERATE card.

```
EXAMPLE:        STORE   0   AT   1140,1341–3577,        *
                                 70–77,7700–7777
```

This would set the specified locations to zero.

4.6 CLEAR ℓ_1-ℓ_2

The specified locations are reset to the 'unassigned' state, i.e. the IBM/360 word has the sign bit set and bits 1 to 31 cleared. CLEAR on its own on a card implies CLEAR 0–7777. The operands may be continued, if necessary, on subsequent cards by coding an asterisk in column 72.

```
EXAMPLE:        CLEAR   5200–6377,7743–7777,        *
                        0–77,3343–3656
```

4.7 DUMP ℓ_1-ℓ_2

This provides an octal dump of pseudocore between the two specified locations. Each word of pseudocore printed appears as an eleven digit octal number, reflecting the 32 bits of the IBM/360 word plus the 33rd bit (a zero) added at the left. The associated PDP-9/L word appears in the right hand six octal digits. The other bits in the IBM/360 word are used in various ways as internal indicators during the execution of PDPGENER. The assignment of bits 0–13 is as follows:

Bit 0	If set, means that the location is 'unassigned'.
Bit 1	This bit is set if the location is to be labelled in the reconstruction of a source listing. The keyword, TRACE, also sets bit 1.
Bit 2	This bit is set to indicate that the location contains a memory reference instruction in the reconstruction of a source listing.
Bit 3	This bit is set whenever this location is to be included when specified on a GENERATE, STORE, CLEAR or TRACE keyword.
Bits 4–13	These bits contain the binary value of the label when source listings are reconstructed. For example, ASM15 would appear as '0000001111'.

4.8 ENTRY ℓ_1

This card updates the current entry point for a simulator or other program. It has exactly the same effect as a transfer card containing the transfer address ℓ_1.

EXAMPLE: ENTRY 6401

This sets the current entry point to 6401.

4.9 TRACE ℓ_1-ℓ_2

This sets the trace bit on for later tracing of instructions by an appropriate simulator. If a full trace is required, code TRACE ALL; to trace no instructions, code TRACE OFF. The operands may be extended and continued in exactly the same manner as the GENERATE operands.

EXAMPLE: TRACE 0–177,2000–2010,3017

This will provide a printed trace of any instructions encountered within these regions during subsequent simulation.

4.10 EXECUTE name

Subroutine calls may be made to programs on disk using the keyword, EXECUTE. Each subroutine called must exist as a member of the data set referred to on the STEPLIB or JOBLIB card or, alternatively, must be a member of SYS1.LINKLIB. These subroutine calls follow standard IBM linkage conventions. In particular, register 1 points to a three-word parameter list, the first of which contains the address of the pseudocore area. The second is the appropriate core-size and the third contains the IBM/360 address of the entry point to the PDP-9/L program. These three parameters would be needed, for example, by a simulator for the PDP-9/L. One may use PDP9LSIM, a PDP-9/L Simulator program written by the author, to set up constants or data before punching an object deck or for checking any number of PDP-9/L programs in the one Job Step. Return from PDP9LSIM will occur on sensing a halt instruction or an input/output instruction. PDP9LOBJ (Mason 1970) may be called in the same way to set up pseudocore preceded by a parameter list as an object module on disk. This is used for program transmission across the Link from the IBM/360 to the PDP-9/L computer (D.J. Richardson 1970). In fact, the editing of programs for Link transmission has become a major use for PDPGENER.

If, however, return of control to PDPGENER is not required, one may code PARM='LOAD=OBJ' or PARM='LOAD=SIM' on the PDPGENER EXEC card to pass control to PDP9LOBJ or PDP9LSIM respectively at the end of PDPGENER processing. Control returns to OS on exit from either PDP9LOBJ or PDP9LSIM. The only restriction is that the subroutine name must begin with the five characters 'PDP9L' and end with the three specified in the PARM field.

EXAMPLE: EXECUTE EAISIM2

This passes control to a simulator for the EAI640 computer. This, of course, can only be meaningful if the object code set up in pseudocore is EAI640 object code. This can be done quite simply by specifying the column 1 format of EAI640 object and transfer cards as described in the following subsection.

4.11 FORMAT ℓ_1, ℓ_2

The format of object and transfer cards produced by Assemblers written at the Research Establishment of the A.A.E.C. for small computers differs only in the punching of column 1 (see APPENDIX 1 for a full description of object and transfer card layout). Considering row 12 as the high order bit position, the PDP-9/L object and transfer card format of column 1 is 4701 and 4702 (octal) respectively. NOVA decks may be read by PDPGENER by specifying 4711 and 4712 on a FORMAT card. EAI640 decks may similarly be read with FORMAT 4721,4722. This means, of course, that PDP-9/L object and transfer cards are treated as comment cards unless a further FORMAT card re-specifies the PDP-9/L format. If subsequent PUNCH keywords appear, the new format is used in preference to the PDP-9/L format.

EXAMPLE: FORMAT 4711,4712

This enables NOVA object and transfer cards to be recognised and processed by PDPGENER. Details of NOVA object and transfer cards may be found in NOVASM (Sanger 1970).

5. ERROR MESSAGES

These include the following:

(1) Missing Delimiter

(2) Missing Operand

(3) Second Operand Missing

(4) Invalid Continuation Character

(5) Operand Count Exceeds 100

(6) First Card Is Not An Object Card

(7) Job Terminated

(8) XXXX Labels Required (1023 Is Max. Allowable)

(9) Generation Not Attempted

(10) Execution Continuing.

Associated with the keyword PUNCH are the following error messages:

(11) 8 or 9 In Octal Number – Job Terminated

(12) Comma out of Context – Job Terminated

(13) Address outside 4K Range – Job Terminated

(14) Operands out of Order – Job Terminated

(15) Missing Operand – Job Terminated

(16) Colon out of Context – Job Terminated.

Error message 5 may refer to the continued numeric operands of GENERATE, STORE, CLEAR or TRACE. An operand is considered to be either ℓ_1 or the pair ℓ_1-ℓ_2. For example: 0–300,400,500,640–677 counts as four operands.

6. SOME OPERATING CONSIDERATIONS

To read in a paper tape which needs a loader, but which does not have one punched at the beginning, copy the loader from another D.E.C. program and read this in first. Wait a second or two after this tape is read in so that the program can register the change from Read-In Mode to 'instruction' mode. INT REQ on UNIT 007 will appear on the operator's console, and this is the signal to turn off the read switch on the paper tape reader and mount the main part of the tape. If this switch is turned off too soon after the loader is read, the entire tape might be read under Read-In Mode with subsequent persistent messages of Intervention Required on Unit 007 when the main tape has been all read.

If a Read-In Mode tape with a halt instruction at the end is to be converted into object deck form, code READ and GENERATE alone as a first guide. It is usually a simple matter to match the labels with the instructions by a suitable displacement of the instructions on a subsequent PDPGENER run.

7. CONCLUSIONS

The ability to reconstruct source listings with complete Cross Reference Tables should prove a useful debugging aid even when the original listing is available because references to absolutely addressed locations appear as well as all original symbolic references.

The conversion of DEC-supplied object paper tape programs to card decks has made access to utility and diagnostic programs simple and efficient, and has also greatly reduced the possibility of error during program loading.

And finally, the ease with which PDP-9/L and other small computer programs can be changed, merged, extended and relocated using PDPGENER should also prove very valuable.

8. ACKNOWLEDGEMENTS

The author wishes to thank Mr. C.B. Mason for making available PDP9LASM to illustrate this report, and Mr. D.J. Richardson without whose advice and encouragement PDPGENER would not have been written.

9. REFERENCES

D.E.C. (1968). – PDP-9/L User's Handbook.

Mason, C.B. (1970). – A PDP9L Assembler for use on the IBM360 – AAEC report in preparation.

Richardson, D.J. (1970). – Generalised Computer to Computer Communication – Doctoral thesis, University of N.S.W.

Sanger, P.L. (1970). – NOVASM and NOVASIM – An Assembler and Simulator for the NOVA and SUPERNOVA computers written to run on an IBM360 computer – In course of publication.

APPENDIX 1

PDP-9/L OBJECT DECK FORMAT

PDP-9/L object and transfer cards are punched in column binary mode so that the actual bit patterns are directly readable from the cards. Considering row 12 as the high order bit position, each column may be read as a four digit octal number. There are seven different fields on each object card beginning with column 1.

(1) Column 1 contains the octal number 4701.

(2) Column 2 contains the checksum calculated so that the sum of all 80 columns is 7777.

(3) The top six rows of column 3 contain the word count of PDP-9/L words on the card. A maximum of forty one words is punched on each object card by PDP9LASM and PDPGENER.

(4) The bottom six rows of column 3 and all of column 4 give an 18 bit address for the first word of data.

(5) Columns 5–65½ contain PDP-9/L data – 1½ columns (18 bits) per word.

(6) Columns 65½–75 are blank.

(7) Columns 76–80 contain a decimal sequence number in EBCDIC form.

Transfer cards are similar with respect to (2) and (7) above, but contain 4702 in column 1, have no word count or data fields, but have the transfer address punched in the bottom half of column 3 and all of column 4 as an 18 bit address.

NOVA and EAI640 object decks are similar to the above except for variations of the column 1 format, and that the 16-bit data words are expanded to 18 bits by the addition of two high order zeros.

APPENDIX 2

A PDPGENER EXAMPLE

The following is a typical PDP-9/L program and its PDPGENER reconstruction. This program is designed to dump the whole of PDP-9/L core onto the line printer, and may be entered at 6400 by a JMS instruction or by a JMP instruction to 6401 if return of control to the calling program is not required.

Notice the way in which characters meant for the line printer buffer are interpreted by PDPGENER as memory reference instructions.

To obtain this reconstruction, the object deck output from PDP9LASM (Mason 1970) was used as input to PDPGENER.

(continued)

```
//PDP      JOB '        ,0031',R.P,BACKSTROM,MSGLEVEL=1
**JST  70.252  18.02.23 00000 R.17
//STEP1  EXEC  PDP9LASM,PARM=DECK
XXASM    EXEC  PGM=PDP9LASM                                                  00000010
//STEPLIB  DD  DSN=CBM.TESTLIB,VOL=REF=PACK6,DISP=OLD
X/STEPLIB  DD  DSN=AAELIB.LMODA,DISP=OLD                                     00000020
XXSYSLOAD  DD  DSN=SYSOBJ,DISP=OLD    USED BY PDP9LOBJ FOR CSECT OUTPUT      00000030
XXSYSPRINT DD  SYSOUT=A                                                      00000040
XXSYSPUNCH DD  SYSOUT=B,DCB=(MODE=C)                                         00000050
XXSYSUT1   DD  DSN=&&PDP9LASM,SPACE=(CYL,(12,1)),UNIT=SYSDA                  00000060
//ASM.SYSIN DD *
IEF236I ALLOC. FOR PDP      ASM      STEP1
IEF237I STEPLIB  ON 132
IEF237I SYSLOAD  ON 135
IEF237I SYSUT1   ON 130
IEF237I SYSIN    ON 00C
```

```
LOC  OBJ CODE   ERRORS   STMT  SOURCE STATEMENT

                          1    /
                          2    /    THIS PROGRAM PROVIDES AN OCTAL DUMP
                          3    / OF THE WHOLE OF THE PDP-9/L CORE ON
                          4    / THE LINE PRINTER.  IT MAY BE ENTERED
                          5    / BY A JMS TO 6400 OR BY A JMP INSTRUC-
                          6    / TION TO 6401 IF RETURN OF CONTROL TO
                          7    / THE CALLING PROGRAM IS NOT REQUIRED.
                          8    /
06400                     9 *        6400
06400 000000             10 DMP,     0                    / SUBROUTINE ENTRY
06401 705514             11          ISA   10             / DISABLE API
06402 206400             12          LAC   DMP            / SAVE RETURN
06403 046567             13          DAC   DP1S           / ADDRESS
06404 146400             14          DZM   DMP            / CLEAR ENTRY
06405 146570             15          DZM   DP1P           / FIRST DUMP LOCATION
06406 777000             16          LAW   0-1000         / SET LINE COUNT
06407 046571             17          DAC   DP1C           / TO 512 (DECIMAL)
06410 706601             18          LPEF                 / WAIT
06411 741000             19          SKP                  / FOR
06412 606410             20          JMP   .-2            / PRINTER
06413 706502             21          LPCB                 / CLEAR PRINTER BUFFER
06414 106473             22          JMS   DMPW           / WAIT
06415 606421             23          JMP   .+4            / SKIP INITIAL PAGE SKIP
06416 760007             24 DMP1,    LAW   7              / SPACE TO
06417 706626             25          LPLS                 / NEXT PAGE
06420 106473             26          JMS   DMPW           / WAIT
06421 777712             27          LAW   0-66           / 54 SINGLE
06422 046572             28          DAC   DP2C           / SPACES
06423 106477             29 DMP2,    JMS   DMPS           / LOAD SINGLE SPACE
06424 446572             30          ISZ   DP2C           / INCREMENT COUNTER
06425 606423             31          JMP   DMP2           / LOOP 54 TIMES
06426 760001             32          LAW   1              / SPACE AFTER NEXT PRINT
06427 106504             33          JMS   DMPH           / LOAD AND PRINT NEXT 4 WORDS
06430 200420             34          200420               / PDP
06431 557157             35          557157               / -9/
06432 144004             36          144004               / L D
06433 251520             37          251520               / UMP
06434 777706             38          LAW   0-72           / SET LINE
06435 046572             39          DAC   DP2C           / COUNTER TO 58
06436 744000             40 DMP3,    CLL                  / CLEAR LINK
06437 206570             41          LAC   DP1P           / LOAD CURRENT ADDRESS
06440 106522             42          JMS   DMPD           / ASSEMBLE 5 DIGIT NUMBER
06441 106554             43          JMS   DMPN           / LOAD NEXT 8 LOCATIONS
06442 106477             44          JMS   DMPS           / LOAD SINGLE SPACE
06443 106554             45          JMS   DMPN           / LOAD NEXT 8 LOCATIONS
06444 750000             46          CLA                  / PRINT THE
06445 706646             47          LPPS                 / ASSEMBLED LINE
06446 106473             48          JMS   DMPW           / WAIT
06447 446571             49          ISZ   DP1C           / INCREMENT TOTAL LINE COUNTER
06450 741000             50          SKP                  / IS IT END OF PAGE ?
06451 606455             51          JMP   DMP4           / PRINT 'END OF DUMP'
06452 446572             52          ISZ   DP2C           / INCREMENT LINES/PAGE COUNTER
06453 606436             53          JMP   DMP3           / SAME PAGE
06454 606416             54          JMP   DMP1           / NEW PAGE
06455 706626             55 DMP4,    LPLS                 / AC=0, SPACE ONE LINE
06456 106473             56          JMS   DMPW           / WAIT
```

```
LOC  OBJ CODE   ERRORS   STMT  SOURCE STATEMENT

06457 760007             57          LAW   7           / NEW PAGE AFTER
06460 106504             58          JMS   DMPH        / PRINTING
06461 051604             59          051604            / END
06462 401706             60          401706            /  OF
06463 400425             61          400425            /  DU
06464 152040             62          152040            / MP
06465 760000             63          LAW   0           / ENABLE
06466 705504             64          ISA               / API
06467 206567             65          LAC   DP1S        / TEST RETURN ADDRESS
06470 741200             66          SNA               / SKIP IF VALID
06471 606471             67          JMP   .           / WAIT IF NOT
06472 626567             68          JMP I DP1S        / EXIT
06473 000000             69 DMPW,    0                 / RETURN ADDRESS
06474 706501             70          LSDF              / WAIT FOR
06475 606474             71          JMP   .-1         / DONE FLAG
06476 626473             72          JMP I DMPW        / EXIT
06477 000000             73 DMPS,    0                 / RETURN ADDRESS
06500 760040             74          LAW   40          / LOAD SINGLE
06501 706566             75          LPL1              / SPACE
06502 106473             76          JMS   DMPW        / WAIT
06503 626477             77          JMP I DMPS        / EXIT
06504 000000             78 DMPH,    0                 / DATA ADDRESS
06505 046573             79          DAC   DP2S        / SAVE SPACING CODE
06506 777774             80          LAW   0-4         / SET WORD COUNTER
06507 046572             81          DAC   DP2C        / TO FOUR
06510 226504             82 DMPH1,   LAC I DMPH        / LOAD 3 CHARACTERS
06511 706546             83          LPLD              / INTO PRINT BUFFER
06512 106473             84          JMS   DMPW        / WAIT
06513 446504             85          ISZ   DMPH        / UPDATE DATA ADDRESS
06514 446572             86          ISZ   DP2C        / INCREMENT COUNTER
06515 606510             87          JMP   DMPH1       / LOOP 4 TIMES
06516 206573             88          LAC   DP2S        / RESTORE SPACING CODE
06517 706646             89          LPPS              / PRINT AND SPACE
06520 106473             90          JMS   DMPW        / WAIT
06521 626504             91          JMP I DMPH        / EXIT
06522 000000             92 DMPD,    0                 / RETURN ADDRESS
06523 046573             93          DAC   DP2S        / SAVE WORD TO BE PRINTED
06524 740400             94          SNL               / SKIP IF WHOLE WORD REQUIRED
06525 606533             95          JMP   DMPD1       / OTHERWISE, SKIP LOADING SPACE
06526 760040             96          LAW   40          / LOAD SINGLE
06527 706566             97          LPL1              / SPACE
06530 106473             98          JMS   DMPW        / WAIT
06531 777772             99          LAW   0-6         / SET COUNTER
06532 606537            100          JMP   DMPD2       / ENTER ROUTINE
06533 742010            101 DMPD1,   RTL               / LEFT JUSTIFY
06534 740010            102          RAL               / 15 BIT ADDRESS
06535 046573            103          DAC   DP2S        / REPLACE WORD
06536 777773            104          LAW   0-5         / SET COUNTER
06537 046577            105 DMPD2,   DAC   DP4C        / IN PLACE
06540 206573            106 DMPD3,   LAC   DP2S        / LOAD WORD
06541 742010            107          RTL               / ROTATE THREE
06542 740010            108          RAL               / LEFT
06543 046573            109          DAC   DP2S        / REPLACE WORD
06544 740010            110          RAL               / ROTATE 1 LEFT
06545 506576            111          AND   DP1M        / LEAVE AC15-17
06546 346575            112          TAD   DP5C        / ADD 60 OCTAL
```

ASSEMBLY LISTING

```
LOC  OBJ CODE   ERRORS   STMT  SOURCE STATEMENT

06547 706566             113          LPL1              / LOAD 1 CHARACTER
06550 106473             114          JMS   DMPW        / WAIT
06551 446577             115          ISZ   DP4C        / INCREMENT COUNTER
06552 606540             116          JMP   DMPD3       / LOOP 5 OR 6 TIMES
06553 626522             117          JMP I DMPD        / EXIT
06554 000000             118 DMPN,    0                 / RETURN ADDRESS
06555 106477             119          JMS   DMPS        / LOAD A SINGLE SPACE
06556 777770             120          LAW   0-10        / 8 WORD
06557 046574             121          DAC   DP3C        / COUNTER
06560 744002             122 DMPN1,   CLL CML           / SET LINK
06561 226570             123          LAC I DP1P        / LOAD FIRST/NEXT WORD
06562 106522             124          JMS   DMPD        / ASSEMBLE WORD
06563 446570             125          ISZ   DP1P        / UPDATE POINTER
06564 446574             126          ISZ   DP3C        / INCREMENT COUNTER
06565 606560             127          JMP   DMPN1       / LOOP 8 TIMES
06566 626554             128          JMP I DMPN        / EXIT
06567 000000             129 DP1S,    0                 / RETURN ADDRESS SAVE AREA
06570 000000             130 DP1P,    0                 / CURRENT DUMP ADDRESS WORD
06571 777000             131 DP1C,    0-1000            / TOTAL LINE COUNTER
06572 777712             132 DP2C,    0-66              / SPACE COUNTER
06573 000000             133 DP2S,    0                 / SPACING CODE AREA
06574 000000             134 DP3C,    0                 / DIGIT COUNTER
06575 000060             135 DP5C,    60                / ZONE BITS FOR DIGITS
06576 000007             136 DP1M,    7                 / MASK FOR BITS 0-14
06577 777770             137 DP4C,    0-10              / WORDS PER HALF LINE
00000                    138          END
```

```
                                                CROSS-REFERENCE

SYMBOL      VALUE            REFERENCES

DMP         006400            12   14
DMPD        006522            42  117  124
DMPD1       006533            95
DMPD2       006537           100
DMPD3       006540           116
DMPH        006504            33   58   82   85   91
DMPH1       006510            87
DMPN        006554            43   45  128
DMPN1       006560           127
DMPS        006477            29   44   77  119
DMPW        006473            22   26   48   56   72   76   84   90   98  114
DMP1        006416            54
DMP2        006423            31
DMP3        006436            53
DMP4        006455            51
DP1C        006571            17   49
DP1M        006576           111
DP1P        006570            15   41  123  125
DP1S        006567            13   65   68
DP2C        006572            28   30   39   52   81   86
DP2S        006573            79   88   93  103  106  109
DP3C        006574           121  126
DP4C        006577           105  115
DP5C        006575           112

  NO STATEMENTS FLAGGED IN THIS ASSEMBLY
```

```
IEF285I   CBM.TESTLIB                                    KEPT
IEF285I   VOL SER NOS= AAE006.
IEF285I   SYSOBJ                                         KEPT
IEF285I   VOL SER NOS= AAE003.
IEF285I   SYSOUT                                         SYSOUT
IEF285I   VOL SER NOS= OUTPUT.
IEF285I   SYS70252.T173727.RP002.PDP.PDP9LASM            DELETED
IEF285I   VOL SER NOS= AAE008.
**EOS ASM       18.02.50 00027 SECS 0000
//STEP2  EXEC  PDPGENER,PARM='LOAD=SIM'
XXGEN    EXEC  PGM=PDPGENER                                                00000010
XXGENPUNCH DD  SYSOUT=B                                                    00000020
XXPTIN     DD  UNIT=007                                                    00000030
XXSTEPLIB  DD  DSN=AAELIB.LMODA,DISP=OLD                                   00000040
XXSYSLOAD  DD  DSN=SYSOBJ,DISP=OLD                                         00000050
XXSYSPRINT DD  SYSOUT=A                                                    00000060
XXSYSPUNCH DD  SYSOUT=B,DCB=MODE=C                                         00000070
//SYSIN  DD    *
IEF236I ALLOC. FOR PDP      GEN      STEP2
IEF237I PTIN     ON 007
IEF237I STEPLIB  ON 134
IEF237I SYSLOAD  ON 135
IEF237I SYSIN    ON 00C
```

```
                                                PDPGENER
                                                ********

                GENERATE,LABEL=LP
```

```
 LOC    CODE          STMT    SOURCE STATEMENT

00420                    1    *       420
00420                    2    LP1,    DS
00425                    3    *       425
00425                    4    LP2,    DS
01706                    5    *       1706
01706                    6    LP3,    DS
04004                    7    *       4004
04004                    8    LP4,    DS
06400                    9    *       6400
06400 000000            10    LP5,    0
06401 705514            11            705514
06402 206400            12            LAC   LP5
06403 046567            13            DAC   LP24
06404 146400            14            DZM   LP5
06405 146570            15            DZM   LP25
06406 777000            16            LAW   0-1000
06407 046571            17            DAC   LP26
06410 706601            18    LP6,    LPEF
06411 741000            19            SKP
06412 606410            20            JMP   LP6
06413 706502            21            LPCB
06414 106473            22            JMS   LP13
06415 606421            23            JMP   LP8
06416 760007            24    LP7,    LAW   7
06417 706626            25            LPLS
06420 106473            26            JMS   LP13
06421 777712            27    LP8,    LAW   0-66
06422 046572            28            DAC   LP27
06423 106477            29    LP9,    JMS   LP15
06424 446572            30            ISZ   LP27
06425 606423            31            JMP   LP9
06426 760001            32            LAW   1
06427 106504            33            JMS   LP16
06430 200420            34            LAC   LP1
06431 557157            35            SAD   LP36
06432 144004            36            DZM   LP4
06433 251520            37            XOR   LP33
06434 777706            38            LAW   0-72
06435 046572            39            DAC   LP27
06436 744000            40    LP10,   CLL
06437 206570            41            LAC   LP25
06440 106522            42            JMS   LP18
06441 106554            43            JMS   LP22
06442 106477            44            JMS   LP15
06443 106554            45            JMS   LP22
06444 750000            46            CLA
06445 706646            47            LPPS
06446 106473            48            JMS   LP13
06447 446571            49            ISZ   LP26
06450 741000            50            SKP
06451 606455            51            JMP   LP11
06452 446572            52            ISZ   LP27
06453 606436            53            JMP   LP10
06454 606416            54            JMP   LP7
06455 706626            55    LP11,   LPLS
```

```
 LOC   CODE          STMT    SOURCE STATEMENT

06456 106473           56            JMS   LP13
06457 760007           57            LAW   7
06460 106504           58            JMS   LP16
06461 051604           59            DAC   LP34
06462 401706           60            XCT   LP3
06463 400425           61            XCT   LP2
06464 152040           62            DZM   LP35
06465 760000           63            LAW   0
06466 705504           64            ISA
06467 206567           65            LAC   LP24
06470 741200           66            SNA
06471 606471           67    LP12,   JMP   LP12
06472 626567           68            JMP I LP24
06473 000000           69    LP13,   0
06474 706501           70    LP14,   LSDF
06475 606474           71            JMP   LP14
06476 626473           72            JMP I LP13
06477 000000           73    LP15,   0
06500 760040           74            LAW   40
06501 706566           75            LPL1
06502 106473           76            JMS   LP13
06503 626477           77            JMP I LP15
06504 000000           78    LP16,   0
06505 046573           79            DAC   LP28
06506 777774           80            LAW   0-4
06507 046572           81            DAC   LP27
06510 226504           82    LP17,   LAC I LP16
06511 706546           83            LPLD
06512 106473           84            JMS   LP13
06513 446504           85            ISZ   LP16
06514 446572           86            ISZ   LP27
06515 606510           87            JMP   LP17
06516 206573           88            LAC   LP28
06517 706646           89            LPPS
06520 106473           90            JMS   LP13
06521 626504           91            JMP I LP16
06522 000000           92    LP18,   0
06523 046573           93            DAC   LP28
06524 740400           94            SNL
06525 606533           95            JMP   LP19
06526 760040           96            LAW   40
06527 706566           97            LPL1
06530 106473           98            JMS   LP13
06531 777772           99            LAW   0-6
06532 606537          100            JMP   LP20
06533 742010          101    LP19,   RTL
06534 740010          102            RAL
06535 046573          103            DAC   LP28
06536 777773          104            LAW   0-5
06537 046577          105    LP20,   DAC   LP32
06540 206573          106    LP21,   LAC   LP28
06541 742010          107            RTL
06542 740010          108            RAL
06543 046573          109            DAC   LP28
06544 740010          110            RAL
```

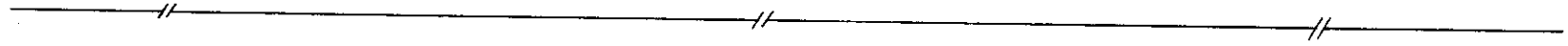
```
 LOC   CODE          STMT    SOURCE STATEMENT

06545 506576          111            AND   LP31
06546 346575          112            TAD   LP30
06547 706566          113            LPL1
06550 106473          114            JMS   LP13
06551 446577          115            ISZ   LP32
06552 606540          116            JMP   LP21
06553 626522          117            JMP I LP18
06554 000000          118    LP22,   0
06555 106477          119            JMS   LP15
06556 777770          120            LAW   0-10
06557 046574          121            DAC   LP29
06560 744002          122    LP23,   STL
06561 226570          123            LAC I LP25
06562 106522          124            JMS   LP18
06563 446570          125            ISZ   LP25
06564 446574          126            ISZ   LP29
06565 606560          127            JMP   LP23
06566 626554          128            JMP I LP22
06567 000000          129    LP24,   0
06570 000000          130    LP25,   0
06571 777000          131    LP26,   LAW   0-1000
06572 777712          132    LP27,   LAW   0-66
06573 000000          133    LP28,   0
06574 000000          134    LP29,   0
06575 000060          135    LP30,   000060
06576 000007          136    LP31,   000007
06577 777770          137    LP32,   LAW   0-10
11520                 138    *       11520
11520                 139    LP33,   DS
11604                 140    *       11604
11604                 141    LP34,   DS
12040                 142    *       12040
12040                 143    LP35,   DS
17157                 144    *       17157
17157                 145    LP36,   DS
00000                 146            END
```

CROSS REFERENCE TABLE

LABEL	ADDR	DEFN	REFERENCES								
LP1	00420	0002	0034								
LP2	00425	0004	0061								
LP3	01706	0006	0060								
LP4	04004	0008	0036								
LP5	06400	0010	0012	0014							
LP6	06410	0018	0020								
LP7	06416	0024	0054								
LP8	06421	0027	0023								
LP9	06423	0029	0031								
LP10	06436	0040	0053								
LP11	06455	0055	0051								
LP12	06471	0067	0067								
LP13	06473	0069	0022	0026	0048	0056	0072	0076	0084	0090	0098	0114
LP14	06474	0070	0071								
LP15	06477	0073	0029	0044	0077	0119					
LP16	06504	0078	0033	0058	0082	0085	0091				
LP17	06510	0082	0087								
LP18	06522	0092	0042	0117	0124						
LP19	06533	0101	0095								
LP20	06537	0105	0100								
LP21	06540	0106	0116								
LP22	06554	0118	0043	0045	0128						
LP23	06560	0122	0127								
LP24	06567	0129	0013	0065	0068						
LP25	06570	0130	0015	0041	0123	0125					
LP26	06571	0131	0017	0049							
LP27	06572	0132	0028	0030	0039	0052	0081	0086			
LP28	06573	0133	0079	0088	0093	0103	0106	0109			
LP29	06574	0134	0121	0126							
LP30	06575	0135	0112								
LP31	06576	0136	0111								
LP32	06577	0137	0105	0115							
LP33	11520	0139	0037								
LP34	11604	0141	0059								
LP35	12040	0143	0062								
LP36	17157	0145	0035								

NO STATEMENTS FLAGGED



POPGENER
****

DUMP

PSEUDOCORE

```
00000  22000000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 22000000000
     SAME AS ABOVE
00420  32001000000 22000000000 22000000000 22000000000   22000000000 32002000000 22000000000 22000000000
00430  22000000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 22000000000
     SAME AS ABOVE
01700  22000000000 22000000000 22000000000 22000000000   22000000000 22000000000 32003000000 22000000000
01710  22000000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 22000000000
     SAME AS ABOVE
04000  22000000000 22000000000 22000000000 22000000000   32004000000 22000000000 22000000000 22000000000
04010  22000000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 22000000000
     SAME AS ABOVE
06400  12005000000 02000705514 06000206400 06000046567   06000146400 06000146570 02000777000 06000046571
06410  12006706601 02000741000 06000606410 02000706502   06000106473 06000606421 12007760007 02000706626
06420  06000106473 12010777712 06000046572 16011106477   06000446572 06000606423 02000760001 06000106504
06430  06000200420 06000557157 06000144004 06000251520   02000777706 06000046572 12012744000 06000206570
06440  06000106522 06000106554 06000106477 06000106554   02000750000 02000706646 06000106473 06000446571
06450  02000741000 06000606455 06000446572 06000606436   06000606416 12013706626 06000106473 02000760007
06460  06000106504 06000051604 06000401706 06000400425   06000152040 02000760000 02000705504 06000206567
06470  02000741200 16014606471 06000626567 12015000000   12016706501 06000606474 06000626473 12017000000
06500  02000760040 02000706566 06000106473 06000626477   12020000000 06000046573 02000777774 06000046572
06510  16021226504 02000706546 06000106473 06000446504   06000446572 06000606510 06000206573 02000706546
06520  06000106473 06000626504 12022000000 06000046573   02000740400 06000606533 02000760040 02000706566
06530  06000106473 02000777772 06000606537 12023742010   02000740010 06000046573 02000777773 16024046577
06540  16025206573 02000742010 02000740010 06000046573   02000740010 06000506576 06000346575 02000706566
06550  06000106473 06000446577 06000606540 06000626522   12026000000 06000106477 02000777770 06000046574
06560  12027744002 06000226570 06000106522 06000446570   06000446574 06000606560 06000626554 12030000000
06570  12031000000 12032777000 12033777712 12034000000   12035000000 12036000060 12037000007 12040777770
06600  22000000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 22000000000
     SAME AS ABOVE
11520  32041000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 22000000000
11530  22000000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 22000000000
     SAME AS ABOVE
11600  22000000000 22000000000 22000000000 22000000000   32042000000 22000000000 22000000000 22000000000
11610  22000000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 22000000000
     SAME AS ABOVE
12040  32043000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 22000000000
12050  22000000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 22000000000
     SAME AS ABOVE
17150  22000000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 32044000000
17160  22000000000 22000000000 22000000000 22000000000   22000000000 22000000000 22000000000 22000000000
     SAME AS ABOVE

     END OF DUMP
```

```
                                                        PDPGENER
                                                        ********
          *
          *       THE FOLLOWING SIMULATION TRACES THE SETTING UP
          *    OF THE HEADING 'PDP-9/L DUMP' ON THE LINE PRINTER.
          *    THE FIRST ADDRESS, 0, AND THE FIRST WORD OF CORE,
          *    001747, IS ALSO SET UP FOR PRINTING.
          *       TO ALLOW A TRACE OF THIS PROGRAM, I/O INSTRUCTIONS
          *    ARE TEMPORARILY REPLACED BY NOP OR SKP INSTRUCTIONS AND
          *    A HALT INSTRUCTION IS INSERTED TO GUARANTEE EXIT
          *    FROM THE SIMULATOR.
          *
                    STORE  001747 AT 0
                    STORE  740000 AT 6401,6410,6413,6501,6511,6517,6527,6547
                    STORE  741000 AT 6474
                    STORE  740040 AT 6563
                    TRACE  6401-6422,6426-6441,6504-6563
                    ENTRY  6401
CONTROL IS PASSED TO PDP9LSIM
```

PDP-9/L SIMULATOR

TIME IN MICROSECS.	PROGRAM COUNTER	OBJECT CODE	INSTRUCTION MNEMONIC	EFFECTIVE ADDRESS	CONTENTS BEFORE	CONTENTS AFTER	LINK, ACCUMULATOR BEFORE	LINK, ACCUMULATOR AFTER	SKIP OBEYED
							0 000000	0 000000	
1.5	06401	740000	NOP				0 000000	0 000000	
4.5	06402	206400	LAC	06400	000000	000000	0 000000	0 000000	
7.5	06403	046567	DAC	06567	000000	000000	0 000000	0 000000	
10.5	06404	146400	DZM	06400	000000	000000	0 000000	0 000000	
13.5	06405	146570	DZM	06570	000000	000000	0 000000	0 777000	
15.0	06406	777000	LAW	0-1000			0 777000	0 777000	
18.0	06407	046571	DAC	06571	777000	777000	0 777000	0 777000	
19.5	06410	740000	NOP				0 777000	0 777000	SKIP
21.0	06411	741000	SKP				0 777000	0 777000	
22.5	06413	740000	NOP				0 777000	0 777000	
25.5	06414	106473	JMS	06473	000000	006415	0 777000	0 777000	
31.5	06415	606421	JMP	06421	777712	777712	0 777000	0 777000	
33.0	06421	777712	LAW	0-66			0 777000	0 777712	
36.0	06422	046572	DAC	06572	777712	777712	0 777712	0 777712	
1170.0	06426	760001	LAW	1			0 760040	0 760001	
1173.0	06427	106504	JMS	06504	000000	006430	0 760001	0 760001	
1176.0	06505	046573	DAC	06573	000000	760001	0 760001	0 760001	
1177.5	06506	777774	LAW	0-4			0 760001	0 777774	
1180.5	06507	046572	DAC	06572	000000	777774	0 777774	0 777774	
1185.0	06510	226504	LAC I	06430	200420	200420	0 777774	0 200420	
1186.5	06511	740000	NOP				0 200420	0 200420	
1189.5	06512	106473	JMS	06473	006503	006513	0 200420	0 200420	
1197.0	06513	446504	ISZ	06504	006430	006431	0 200420	0 200420	
1200.0	06514	446572	ISZ	06572	777774	777775	0 200420	0 200420	
1201.5	06515	606510	JMP	06510	226504	226504	0 200420	0 200420	
1206.0	06510	226504	LAC I	06431	557157	557157	0 200420	0 557157	
1207.5	06511	740000	NOP				0 557157	0 557157	
1210.5	06512	106473	JMS	06473	006513	006513	0 557157	0 557157	
1218.0	06513	446504	ISZ	06504	006431	006432	0 557157	0 557157	
1221.0	06514	446572	ISZ	06572	777775	777776	0 557157	0 557157	
1222.5	06515	606510	JMP	06510	226504	226504	0 557157	0 557157	
1227.0	06510	226504	LAC I	06432	144004	144004	0 557157	0 144004	
1228.5	06511	740000	NOP				0 144004	0 144004	
1231.5	06512	106473	JMS	06473	006513	006513	0 144004	0 144004	
1239.0	06513	446504	ISZ	06504	006432	006433	0 144004	0 144004	
1242.0	06514	446572	ISZ	06572	777776	777777	0 144004	0 144004	
1243.5	06515	606510	JMP	06510	226504	226504	0 144004	0 144004	
1248.0	06510	226504	LAC I	06433	251520	251520	0 144004	0 251520	
1249.5	06511	740000	NOP				0 251520	0 251520	
1252.5	06512	106473	JMS	06473	006513	006513	0 251520	0 251520	
1260.0	06513	446504	ISZ	06504	006433	006434	0 251520	0 251520	
1263.0	06514	446572	ISZ	06572	777777	000000	0 251520	0 251520	SKIP
1266.0	06516	206573	LAC	06573	760001	760001	0 251520	0 760001	
1267.5	06517	740000	NOP				0 760001	0 760001	
1270.5	06520	106473	JMS	06473	006513	006521	0 760001	0 760001	
1278.0	06521	626504	JMP I	06434	777706	777706	0 760001	0 760001	
1279.5	06434	777706	LAW	0-72			0 760001	0 777706	
1282.5	06435	046572	DAC	06572	000000	777706	0 777706	0 777706	
1284.0	06436	744000	CLL				0 777706	0 777706	
1287.0	06437	206570	LAC	06570	000000	000000	0 777706	0 000000	
1290.0	06440	106522	JMS	06522	000000	006441	0 000000	0 000000	
1293.0	06523	046573	DAC	06573	760001	000000	0 000000	0 000000	
1294.5	06524	740400	SNL				0 000000	0 000000	
1296.0	06525	606533	JMP	06533	742010	742010	0 000000	0 000000	

PDP-9/L SIMULATOR

TIME IN MICROSECS.	PROGRAM COUNTER	OBJECT CODE	INSTRUCTION MNEMONIC	EFFECTIVE ADDRESS	CONTENTS BEFORE	CONTENTS AFTER	LINK, ACCUMULATOR BEFORE	LINK, ACCUMULATOR AFTER	SKIP OBEYED
1297.5	06533	742010	RTL				0 000000	0 000000	
1299.0	06534	740010	RAL				0 000000	0 000000	
1302.0	06535	046573	DAC	06573	000000	000000	0 000000	0 000000	
1303.5	06536	777773	LAW	0-5			0 000000	0 777773	
1306.5	06537	046577	DAC	06577	777770	777773	0 777773	0 777773	
1309.5	06540	206573	LAC	06573	000000	000000	0 777773	0 000000	
1311.0	06541	742010	RTL				0 000000	0 000000	
1312.5	06542	740010	RAL				0 000000	0 000000	
1315.5	06543	046573	DAC	06573	000000	000000	0 000000	0 000000	
1317.0	06544	740010	RAL				0 000000	0 000000	
1320.0	06545	506576	AND	06576	000007	000007	0 000000	0 000000	
1323.0	06546	346575	TAD	06575	000060	000060	0 000000	0 000060	
1324.5	06547	740000	NOP				0 000060	0 000060	
1327.5	06550	106473	JMS	06473	006521	006551	0 000060	0 000060	
1335.0	06551	446577	ISZ	06577	777773	777774	0 000060	0 000060	
1336.5	06552	606540	JMP	06540	206573	206573	0 000060	0 000060	
1339.5	06540	206573	LAC	06573	000000	000000	0 000060	0 000000	
1341.0	06541	742010	RTL				0 000000	0 000000	
1342.5	06542	740010	RAL				0 000000	0 000000	
1345.5	06543	046573	DAC	06573	000000	000000	0 000000	0 000000	
1347.0	06544	740010	RAL				0 000000	0 000000	
1350.0	06545	506576	AND	06576	000007	000007	0 000000	0 000000	
1353.0	06546	346575	TAD	06575	000060	000060	0 000000	0 000060	
1354.5	06547	740000	NOP				0 000060	0 000060	
1357.5	06550	106473	JMS	06473	006551	006551	0 000060	0 000060	
1365.0	06551	446577	ISZ	06577	777774	777775	0 000060	0 000060	
1366.5	06552	606540	JMP	06540	206573	206573	0 000060	0 000060	
1369.5	06540	206573	LAC	06573	000000	000000	0 000060	0 000000	
1371.0	06541	742010	RTL				0 000000	0 000000	
1372.5	06542	740010	RAL				0 000000	0 000000	
1375.5	06543	046573	DAC	06573	000000	000000	0 000000	0 000000	
1377.0	06544	740010	RAL				0 000000	0 000000	
1380.0	06545	506576	AND	06576	000007	000007	0 000000	0 000000	
1383.0	06546	346575	TAD	06575	000060	000060	0 000000	0 000060	
1384.5	06547	740000	NOP				0 000060	0 000060	
1387.5	06550	106473	JMS	06473	006551	006551	0 000060	0 000060	
1395.0	06551	446577	ISZ	06577	777775	777776	0 000060	0 000060	
1396.5	06552	606540	JMP	06540	206573	206573	0 000060	0 000060	
1399.5	06540	206573	LAC	06573	000000	000000	0 000060	0 000000	
1401.0	06541	742010	RTL				0 000000	0 000000	
1402.5	06542	740010	RAL				0 000000	0 000000	
1405.5	06543	046573	DAC	06573	000000	000000	0 000000	0 000000	
1407.0	06544	740010	RAL				0 000000	0 000000	
1410.0	06545	506576	AND	06576	000007	000007	0 000000	0 000000	
1413.0	06546	346575	TAD	06575	000060	000060	0 000000	0 000060	
1414.5	06547	740000	NOP				0 000060	0 000060	
1417.5	06550	106473	JMS	06473	006551	006551	0 000060	0 000060	
1425.0	06551	446577	ISZ	06577	777776	777777	0 000060	0 000060	
1426.5	06552	606540	JMP	06540	206573	206573	0 000060	0 000060	
1429.5	06540	206573	LAC	06573	000000	000000	0 000060	0 000000	
1431.0	06541	742010	RTL				0 000000	0 000000	
1432.5	06542	740010	RAL				0 000000	0 000000	
1435.5	06543	046573	DAC	06573	000000	000000	0 000000	0 000000	
1437.0	06544	740010	RAL				0 000000	0 000000	

PDP-9/L SIMULATOR

TIME IN MICROSECS.	PROGRAM COUNTER	OBJECT CODE	INSTRUCTION MNEMONIC	EFFECTIVE ADDRESS	CONTENTS BEFORE	CONTENTS AFTER	LINK, ACCUMULATOR BEFORE	LINK, ACCUMULATOR AFTER	SKIP OBEYED
1440.0	06545	506576	AND	06576	000007	000007	0 000000	0 000000	
1443.0	06546	346575	TAD	06575	000060	000060	0 000000	0 000060	
1444.5	06547	740000	NOP				0 000060	0 000060	
1447.5	06550	106473	JMS	06473	006551	006551	0 000060	0 000060	
1455.0	06551	446577	ISZ	06577	777777	000000	0 000060	0 000060	SKIP
1458.0	06553	626522	JMP I	06441	106554	106554	0 000060	0 000060	
1461.0	06441	106554	JMS	06554	000000	006442	0 000060	0 000060	
1464.0	06555	106477	JMS	06477	006424	006556	0 000060	0 000060	
1479.0	06556	777770	LAW 0-10				0 760040	0 777770	
1482.0	06557	046574	DAC	06574	000000	777770	0 777770	0 777770	
1483.5	06560	744002	CLL CML				0 777770	1 777770	
1488.0	06561	226570	LAC I	00000	001747	001747	1 777770	1 001747	
1491.0	06562	106522	JMS	06522	006441	406563	1 001747	1 001747	
1494.0	06523	046573	DAC	06573	000000	001747	1 001747	1 001747	
1495.5	06524	740400	SNL				1 001747	1 001747	SKIP
1497.0	06526	760040	LAW 40				1 001747	1 760040	
1498.5	06527	740000	NOP				1 760040	1 760040	
1501.5	06530	106473	JMS	06473	006503	406531	1 760040	1 760040	
1507.5	06531	777772	LAW 0-6				1 760040	1 777772	
1509.0	06532	606537	JMP	06537	046577	046577	1 777772	1 777772	
1512.0	06537	046577	DAC	06577	000000	777772	1 777772	1 777772	
1515.0	06540	206573	LAC	06573	001747	001747	1 777772	1 001747	
1516.5	06541	742010	RTL				1 001747	0 007636	
1518.0	06542	740010	RAL				0 007636	0 017474	
1521.0	06543	046573	DAC	06573	001747	017474	0 017474	0 017474	
1522.5	06544	740010	RAL				0 017474	0 037170	
1525.5	06545	506576	AND	06576	000007	000007	0 037170	0 000000	
1528.5	06546	346575	TAD	06575	000060	000060	0 000000	0 000060	
1530.0	06547	740000	NOP				0 000060	0 000060	
1533.0	06550	106473	JMS	06473	406531	006551	0 000060	0 000060	
1540.5	06551	446577	ISZ	06577	777772	777773	0 000060	0 000060	
1542.0	06552	606540	JMP	06540	206573	206573	0 000060	0 000060	
1545.0	06540	206573	LAC	06573	017474	017474	0 000060	0 017474	
1546.5	06541	742010	RTL				0 017474	0 076360	
1548.0	06542	740010	RAL				0 076360	0 174740	
1551.0	06543	046573	DAC	06573	017474	174740	0 174740	0 174740	
1552.5	06544	740010	RAL				0 174740	0 371700	
1555.5	06545	506576	AND	06576	000007	000007	0 371700	0 000000	
1558.5	06546	346575	TAD	06575	000060	000060	0 000000	0 000060	
1560.0	06547	740000	NOP				0 000060	0 000060	
1563.0	06550	106473	JMS	06473	006551	006551	0 000060	0 000060	
1570.5	06551	446577	ISZ	06577	777773	777774	0 000060	0 000060	
1572.0	06552	606540	JMP	06540	206573	206573	0 000060	0 000060	
1575.0	06540	206573	LAC	06573	174740	174740	0 000060	0 174740	
1576.5	06541	742010	RTL				0 174740	0 763600	
1578.0	06542	740010	RAL				0 763600	1 747400	
1581.0	06543	046573	DAC	06573	174740	747400	1 747400	1 747400	
1582.5	06544	740010	RAL				1 747400	1 717001	
1585.5	06545	506576	AND	06576	000007	000007	1 717001	1 000001	
1588.5	06546	346575	TAD	06575	000060	000060	1 000001	1 000061	
1590.0	06547	740000	NOP				1 000061	1 000061	
1593.0	06550	106473	JMS	06473	006551	406551	1 000061	1 000061	
1600.5	06551	446577	ISZ	06577	777774	777775	1 000061	1 000061	
1602.0	06552	606540	JMP	06540	206573	206573	1 000061	1 000061	

PDP-9/L SIMULATOR

TIME IN MICROSECS.	PROGRAM COUNTER	OBJECT CODE	INSTRUCTION MNEMONIC	EFFECTIVE ADDRESS	CONTENTS BEFORE	CONTENTS AFTER	LINK, ACCUMULATOR BEFORE	LINK, ACCUMULATOR AFTER	SKIP OBEYED
1605.0	06540	206573	LAC	06573	747400	747400	1 000061	1 747400	
1606.5	06541	742010	RTL				1 747400	1 636003	
1608.0	06542	740010	RAL				1 636003	1 474007	
1611.0	06543	046573	DAC	06573	747400	474007	1 474007	1 474007	
1612.5	06544	740010	RAL				1 474007	1 170017	
1615.5	06545	506576	AND	06576	000007	000007	1 170017	1 000007	
1618.5	06546	346575	TAD	06575	000060	000060	1 000007	1 000067	
1620.0	06547	740000	NOP				1 000067	1 000067	
1623.0	06550	106473	JMS	06473	406551	406551	1 000067	1 000067	
1630.5	06551	446577	ISZ	06577	777775	777776	1 000067	1 000067	
1632.0	06552	606540	JMP	06540	206573	206573	1 000067	1 000067	
1635.0	06540	206573	LAC	06573	474007	474007	1 000067	1 474007	
1636.5	06541	742010	RTL				1 474007	0 360037	
1638.0	06542	740010	RAL				0 360037	0 740076	
1641.0	06543	046573	DAC	06573	474007	740076	0 740076	0 740076	
1642.5	06544	740010	RAL				0 740076	1 700174	
1645.5	06545	506576	AND	06576	000007	000007	1 700174	1 000004	
1648.5	06546	346575	TAD	06575	000060	000060	1 000004	1 000064	
1650.0	06547	740000	NOP				1 000064	1 000064	
1653.0	06550	106473	JMS	06473	406551	406551	1 000064	1 000064	
1660.5	06551	446577	ISZ	06577	777776	777777	1 000064	1 000064	
1662.0	06552	606540	JMP	06540	206573	206573	1 000064	1 000064	
1665.0	06540	206573	LAC	06573	740076	740076	1 000064	1 740076	
1666.5	06541	742010	RTL				1 740076	1 600373	
1668.0	06542	740010	RAL				1 600373	1 400767	
1671.0	06543	046573	DAC	06573	740076	400767	1 400767	1 400767	
1672.5	06544	740010	RAL				1 400767	1 001757	
1675.5	06545	506576	AND	06576	000007	000007	1 001757	1 000007	
1678.5	06546	346575	TAD	06575	000060	000060	1 000007	1 000067	
1680.0	06547	740000	NOP				1 000067	1 000067	
1683.0	06550	106473	JMS	06473	406551	406551	1 000067	1 000067	
1690.5	06551	446577	ISZ	06577	777777	000000	1 000067	1 000067	SKIP
1693.5	06553	626522	JMP I	06563	740040	740040	1 000067	1 000067	
1695.0	06563	740040	HLT				1 000067	1 000067	

CONTROL HAS REACHED A HALT INSTRUCTION

END OF SIMULATION

```
IEF285I   AAELIB.LMODA                                    KEPT
IEF285I   VOL SER NOS= AAE004.
IEF285I   SYSOBJ                                          KEPT
IEF285I   VOL SER NOS= AAE003.
IEF285I   SYSOUT                                          SYSOUT
IEF285I   VOL SER NOS= OUTPUT.
**EOS GEN       18.03.24 00033 SECS 0000
//
**EOJ PDP       18.03.25 0.02 HOURS
```