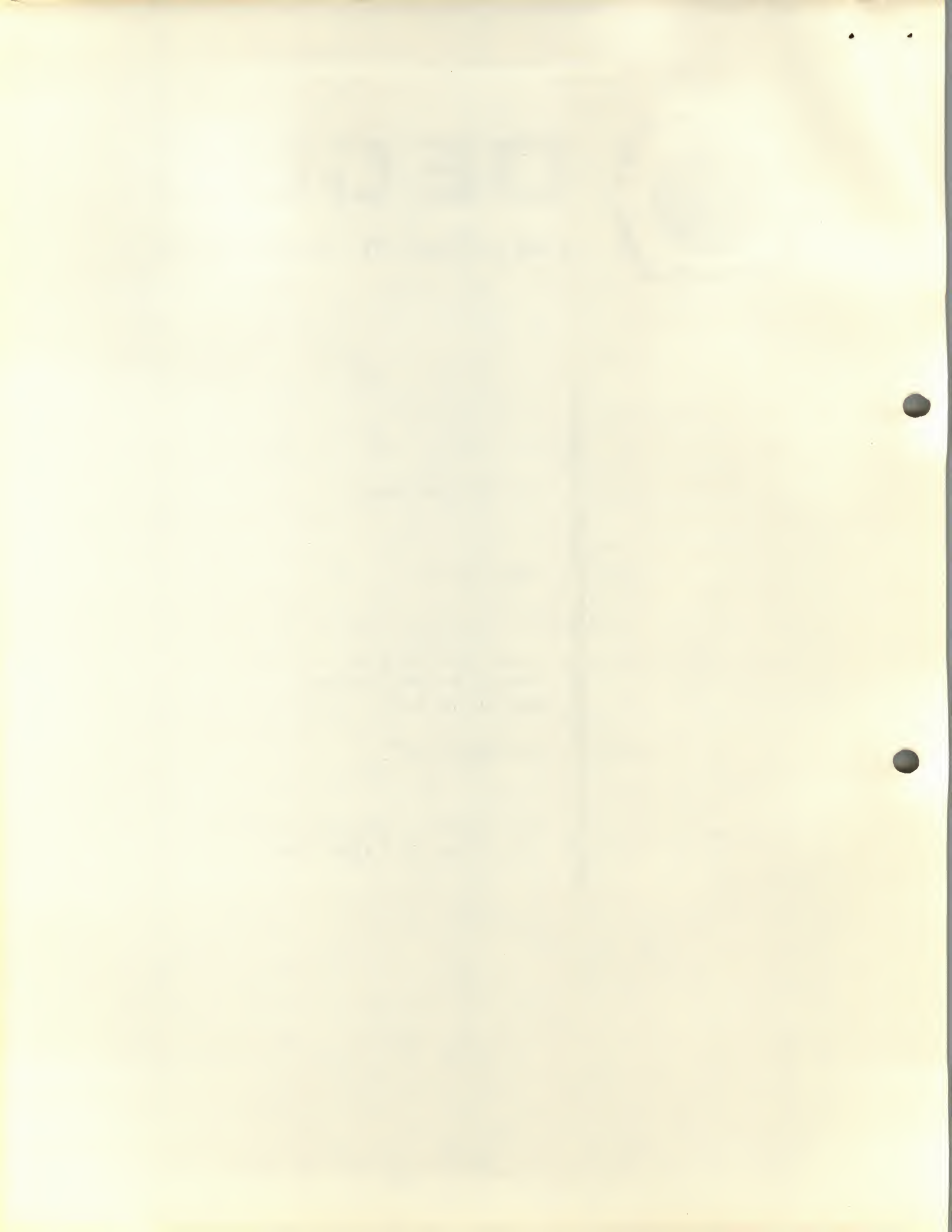




# DECUS

## PROGRAM LIBRARY

|                 |   |
|-----------------|---|
| DECUS NO.       | 8-377   |
| TITLE           | ONE PASS ASSEMBLER  |
| AUTHOR          | Barney Hordos III   |
| COMPANY         | University of California<br>Lawrence Radiation Laboratory<br>Berkeley, California |
| DATE            | September 29, 1970  |
| SOURCE LANGUAGE | NOT SOURCE LANGUAGE COMPATIBLE<br>WITH OTHER PDP-8 ASSEMBLERS                     |



T A B L E   O F   C O N T E N T S

ONE-PASS ASSEMBLER

|    |                              |    |
|----|------------------------------|----|
| 1. | Description . . . . .        | 3  |
| 2. | Pseudo Operations . . . . .  | 6  |
| 3. | Special Characters . . . . . | 9  |
| 4. | Error Indications . . . . .  | 10 |
| 5. | Appendix A . . . . .         | 13 |
| 6. | Appendix B . . . . .         | 17 |
| 7. | Appendix C . . . . .         | 20 |
| 8. | Appendix D . . . . .         | 22 |

[Faint, illegible title text]

[Faint, illegible body text, possibly a list or table]

ONE - PASS ASSEMBLER

ABSTRACT: This program is for on-line use with the PDP-8 Series computers with 4k or greater core. The assembly is made directly into the memory of the computer, and a paper tape in rim or bin format may be punched. The op code field length is limited to three characters with indirect addressing and linking available.

DESCRIPTION: There are 4 major fields in a line.

- (a) THE LOCATION FIELD must start with a letter and can be 3 or more characters long, but only the last three characters are used.
- (b) THE OPERATION FIELD follows and uses only 3 alpha characters. Operate instructions may be combined using a comma. Absolute instructions may be entered directly.
- (c) THE ADDRESS OR NUMERIC FIELD is next and may be absolute or symbolic. If symbolic, the same restrictions are imposed as in the location field. The address may be modified by an octal constant when using symbolic addressing.
- (d) COMMENT FIELD is the last in line. Here comments may be entered by the operator for explanation of procedure.

SYMBOLS: For the purpose of this assembler, a symbol is defined as a string of 3 or fewer characters where the leading character must be alpha. If more than 3 characters are entered, only the last three will be considered. A character is a letter or number. In this assembler, all numeric data is assumed to be octal except the pseudo-operation DEC, which converts decimal numbers to octal. Numbers may be signed. All numbers are right adjusted and four digits long. ie.. 0007 is equal to 7 and 123456 is equal to 3456. Special symbols are not allowed except as noted.

LOCATION: The location field may contain a symbol if desired; but if none is needed, a space terminates the field. A symbol entered in the location field is assigned the current value of the location counter. The symbol can be then referenced from other places in the program. A symbol which appears more than once in the location field is considered a multiply-defined symbol. An error will be detected, and the message, "PREVIOUSLY DEFINED", will be typed and the same line reentered.

OPERATION CODE: The operation code can be the absolute instruction, a legal symbolic 3-letter op code, or a previously defined 3-letter op code. The operation code field is terminated with a space. If a space was the only character typed, it is interpreted as an absolute instruction of value zero. The special symbol "\*", is to be used for indirect addressing.

ADDRESS: The address field is skipped if the op code was an operate instruction, a previously defined op code, a space, or an absolute instruction. However, if the op code is a memory reference instruction, the address field may contain:

- (a) an absolute address,
- (b) a 3-letter symbolic address (which will have to be defined at some time in the location field), or
- (c) the symbol "+n".

The page address is then formed. In numeric address, a space is interpreted as zero. A symbol has the value of the location counter which was defined in the location field. Numbers must be in octal except where the op code is the pseudo-operation DEC, which requires a decimal value. An address field "+n" is interpreted as follows: "." is the symbol which has the present value of the location counter;

"+n" is then algebraically added to the location counter to form the address.

If "." appears, the next character must be a "+". A symbolic address may also have "+n" added algebraically to it. An illegal address which is located on the wrong page will be flagged and the assembler will type "WRONG PAGE" and will reenter the line.

OPERATION CODES: The operation codes are described in the PDP-8 manual or the small computer handbook. Appendix B gives a list of the op codes recognized by this assembler.

## PSEUDO OPERATIONS:

1. OCT: The address field is interpreted as a signed octal number. If "-", the 2's complement is formed.
2. DEC: The address field is interpreted as a signed decimal number. If "-", the 2's complement of the octal number is formed. The decimal number in absolute value must be less than 4095 decimal. (Care must be exercised in signed numbers, since the largest positive 2's complement number is  $+2047_{10}$  and the largest negative,  $-2048_{10}$ .)
3. ALF: Used to introduce alphabetic information into the program. The first 2 characters read into the address field will be converted to a trim code and stored as one packed word. Any character on the TTY may be used. The code used to form the two parts of the word is obtained by subtracting  $240_8$  from the teletype code.
4. DEF: Used to define a new op code. The symbolic name is entered in the location field and the absolute value is entered in the address field. This op code may not be used to change a previously defined op code nor to define a memory reference instruction, as when the defined op code is used, the address field is skipped.

5. BSZ: Defines a block storage having a length specified by the value of the octal number in the address field. The contents of the elements of the block are set to zero.
6. EQU: Used in linking from one page to another. The contents of the location specified by the present value of the location counter will take on the value of the symbol in the address field. Any legal address may be used.
7. END: Determines if all symbolic address have been defined. Two searches are made, one of the Undefined Address Table, and one of the Line Table. If an undefined address is found, the name, location, and "UNDEFINED" is typed. When all addresses have been defined, "DONE" is then typed.
8. ORG: The address field is interpreted as the value to which the location counter is to be set. Only a numeric address can be used.
9. RIM: This pseudo op code punches out information contained in selected blocks of core memory as RIM-coded tape via the ASR 33 perforated tape reader. Up to 9 DEC different blocks of core may be punched on one tape. After the line has been terminated, the TTY will type "SA":

(a) Type the Starting Address of the block to be punched. A space terminates the line.

The TTY will then type "FA":

(b) Type the Final Address of the block to be punched; a space again terminates the line.

The TTY will again type "SA". To terminate the "read in" mode and punch the tape the FA must be zero; either a space or a zero then a space will terminate the "read in" mode. The TTY will then do a CR/LF. Turn on the punch and hit the space bar; the tape then will be punched. A check sum is punched before the trailer.

10. BIN: Punches paper tape in BIN format. The procedure is the same as for RIM.

11. The PDP-8 floating point instructions have been incorporated in the assembler. (See Appendix D.--Floating Point System Summary.) They are:

FLT: This inserts JMS\* 07, an indirect jump into the floating point package. Instructions are interpreted by the package until an exit is encountered.

FNT: Same as FLT, (used with IRL modified floating point package).

EXT: Inserts a 0000. This is the exit from floating point.

ADD: Inserts lxxx. It causes a floating add to the floating accumulator from location xxx.

SUB: Inserts 2xxx. Subtracts xxx from floating AC.

MUY: Inserts 3xxx. Multiply from xxx.

DIV: Inserts 4xxx. Divide from xxx.

GET: Inserts 5xxx. Loads floating AC from xxx without clearing xxx.

PUT: Inserts 6xxx. Stores floating AC in xxx without clearing FAC.

COMMENT FIELD: After termination from the address field, the teletype spaces to the comment field and prints a slash, "/". Any teletype character except special characters may be printed here. A carriage return terminates the line and the Location Counter is incremented by one.

SPECIAL CHARACTERS: Special characters may be used any time; those used in the assembler are:

CARRIAGE RETURN: Used to terminate the line. If this character is used before the line is ended (i.e., not in the comment field), an error "NOT FINISHED" will be typed and the whole line will be reentered.

RUB OUT: Will cause the whole line to be reentered. This symbol may be used any time before the carriage return is used and will erase the whole line.

/ A carriage return and line feed will be executed and teletype will tab over to the comment field.

= Causes the teletype to type the current value of the assembled op code, et al, associated with the current value of the location counter.

\$ Erases the defined list, but the value of the location counter will remain unchanged.

↑ Restarts the program with the location counter reset to 200, clearing all lists.

ALTMODE Causes the computer to jump indirectly to the location specified by location 4026. (an escape route to system loaders, etc.)

ERROR INDICATIONS: Various assembly conditions are illegal;

the appropriate error flags are defined below:

UNDEFINED: Will be typed when an op code not in the legal op code list or not a newly defined op code is encountered in the operation field. No check is made on octal op codes.

ILLEGAL CHAR Will be typed if:

- (a) A number is typed first in the location field.
- (b) A combination of alpha and octal is typed in the operation field.
- (c) A number is typed first, then an alpha character in the address field.
- (d) A letter is typed after a "+".
- (e) An 8 or 9 is typed at any time except on the DEC op code.

PREVIOUS  
DEF.

This will be typed if a symbol has been used in the location field once, and is reused. The "\$" can clear the old list allowing the new name to be used, but all other entries in the old list are then lost.

TOO  
MANY

This is typed while in the location field if no more room is available in the list for a new tag. The only way to name a new tag is to erase the defined list. (Either a "\$" which erases just the defined list, or an " " which erases all lists.) If "TOO MANY" is typed while in the pseudo op code, DEF, (the list for new defined op codes) is full; if in DEC, the number is greater than  $4095_{10}$ , the largest number the computer can handle.

DEFINE  
SOME:

Indicates either the Link Table or the Undefined Table is full, depending upon which one is in use at the time the message was typed.

WRONG  
PAGE!

A reference has been made which is not on the same page or page zero. It could occur in the location field, showing that earlier a reference to the symbol had been made in an address field on another page. It can also occur in the address field, whenever the calculated address is off the current page and not in page zero.

NOT            This means the carriage return had been struck  
FINISHED!     before the teletype was in the comment field.

On any error, the whole line is erased and reentered.

ACKNOWLEDGMENTS: The author wishes to thank Mr. Don M. Evans for the original concept and for his help and criticism, and also Mr. James Guggemos and Mr. Robert Belshe for their valuable assistance.

A P P E N D I X    A

|      |       |          |       |   |
|------|-------|----------|-------|---|
| 0200 | TYP   | DEF      | 4430  | /DEFINE A SUBROUTINE NAMED TYP, /<br>/JMS INDIRECTLY THRU LOCATION 30.                        |
| 0200 | PNT   | DEF      | 4431  | /DEFINE A SUBROUTINE NAMED PNT, /<br>/JMS INDIRECTLY THRU LOCATION 31.                        |
| 0200 | START | CRA      |       | /START OF MAIN PROGRAM, NOTE ONLY THE /<br>/LAST 3 LETTERS OF THE TAG START ARE /<br>/USED.   |
| 0201 |       | TYP      |       | /THIS CALLS THE SUBROUTINE TYP, =4430 /<br>/THE EQUAL SIGN GIVES THE OCTAL /<br>/INSTRUCTION. |
| 0202 |       | EGU      | MSG   | /THIS WILL BE THE ADDRESS OF MSG.   |
| 0203 |       | HLT      |       | /PROGRAM IS OVER .  |
| 0204 |       | ORG      | 500   | /START AT LOCATION 500.   |
| 0500 | TYP   | BSZ      | 1     | /THE SUBROUTINE TYP IS NOW DEFINED.   |
| 0501 |       | TAD*     | TYP   | /TAD INDIRECTLY THRU LOCATION TYP /<br>/TO GET THE ADDRESS OF THE MESSAGE.                    |
| 0502 |       | DCA      | ADR   | /PUT AT LOCATION ADR.   |
| 0503 |       | ISZ      | TYP   | /SET UP RETURN.   |
| 0504 | LOOP  | TAD*     | ADR   | /GET FIRST WORD OF MESSAGE.   |
| 0505 |       | SNA      |       | /SKIP IF AC DOES NOT EQUAL ZERO /<br>/IF ZERO RETURN.   |
| 0506 |       | JMP*     | TYP   | /RETURN.  |
| 0507 |       | DCA      | TMP   | /PUT WORD IN TEMP LOCATION.   |
| 0510 |       | TAD      | TMP   | /GET WORD.  |
| 0511 |       | AND      | M7700 | /MASK OFF LOWER PART OF WORD, SECOND /<br>/LETTER.  |
| 0512 |       | JMS      | R7L   | /JMP TO SUBROUTINE ROTATE 7 LEFT.   |
| 0513 |       | TAD      | K240  | /ASSEMBLE WORD FOR PRINTING.  |
| 0514 |       | PNT      |       | /PRINT IT .   |
| 0515 |       | TAD      | TMP   | /GET WORD AGAIN.  |
| 0516 |       | AND      | M77   | /MASK OFF FIRST LETTER.   |
| 0517 |       | TAD      | K240  | /ASSEMBLE FOR PRINTING.   |
| 0520 |       | PNT      |       | /PRINT IT.  |
| 0521 |       | ISZ      | ADR   | /SET UP LOCATION OF NEXT WORD.  |
| 0522 |       | JMP      | LOOP  | /GET NEXT WORD.   |
| 0523 | ADR   | 0        |       | /SETS THIS LOCATION TO ZERO FOR A /<br>/TEMP LOCATION FOR ADDRESS.                            |
| 0524 | TMP   | BSZ      | 1     | /ALSO SETS THIS LOCATION TO ZERO.   |
| 0525 | M7700 | OCT      | 7700  | /THIS SETS THIS LOCATION TO 7700.   |
| 0526 | M77   | 77       |       | /THIS SETS THIS LOCATION TO 77.   |
| 0527 | K240  | 240      |       | /CONSTANT 240.  |
| 0530 |       | ORG      | 400   | /START AT LOCATION 400.   |
| 0400 | R7L   | BSZ      | 1     | /SUBROUTINE TO ROTATE AC AND L /<br>/7 TIMES LEFT.  |
| 0401 |       | CLL, RAL |       | /COMBINES CLL AND RAL USING COMMA.  |
| 0402 |       | RTL      |       | /ROTATE 2 LEFT.   |
| 0403 |       | RTL      |       | /   |
| 0404 |       | RTL      |       | /   |
| 0405 |       | JMP*     | R7L   | /JMP INDIRECTLY THRU R7L.   |

```

0406      PNT      0          /ZERO THIS LOCATION FOR RETURN.
0407      TLS          /SEND CHARACTER, CLEAR FLAG.
0410      TSF          /SKIP WHEN DONE.
0411      JMP      -1      /JMP BACK 1 LOCATION.
0412      CRA          /CLEAR AC AND L.
0413      JMP*     PNT     /RETURN.
0414      ORG      300     /START AT LOCATION 300.

0300      MSG      ALF      ON      /MESSAGE TO BE PRINTED IN LPL CODE.
0301      ALF      E          /
0302      ALF      PA        /
0303      ALF      SS        /
0304      ALF      A          /
0305      ALF      SS        /
0306      ALF      EM        /
0307      ALF      BL        /
0310      ALF      ER        /
0311      ALF      .          /
0312      /ZERO THIS LOCATION WITH A SPACE /
          /SO PROGRAM WILL KNOW WHEN MESSAGE /
          /IS ENDED.
0313      ORG      30      /START AT 30.

0030      EGU      TYP      /PUTS LOCATION OF TYP HERE.
0031      EGU      PNT      /PUTS LOCATION OF PNT HERE.
0032      END

DONE

0032      BIN          /USING END WE CHECK FOR ANY UNDEFINED /
          /TAGS, BIN WILL GIVE A PAPER TAPE /
          /IN BIN FORMAT FOR THE LOCATIONS /
          /BETWEEN SA: AND FA: .

SA: 200
FA: 203
SA: 500
FA: 527
SA: 400
FA: 413
SA: 300
FA: 312
SA: 30
FA: 31
SA:
FA:
B; $<E<</"SS+?? D9888.0&0!*;.C/.%0!33!33%-",%20+
0032      /PROGRAM IS NOW ALL ASSEMBLED /
          /AND A PAPER TAPE PUNCHED.
0033

```

```

0200          DEF      4500      /THIS WOULD DEFINE A SPACE AS JMS* 100 /
                                     /BECAUSE A SPACE WAS THE ONLY CHARACTER/
                                     /IN THE LOCATION FIELD. /
                                     /THIS IS NOT WANTED SO THE RUBOUT KEY /
0200  TYP      DEF      4430      /IS HIT AND THE LINE RE-ENTERED. +
                                     /NOW LINE IS CORRECT.

0200  PNT      DEF      4431      /ALL NEW OPCODES MUST BE DEFINED /
                                     /BEFORE THEY ARE USED OR THEY CAN NOT /
                                     /BE DEFINED.

0200  START   CRA
0201  START   PREVIOUS DEF.
0201
                                     /START CAN NOT BE USED AS AN OPCODE.
0201  START   DEF      4432      /THE ONLY WAY START CAN BE DEFINED IS /
                                     /TO CLEAR THE DEFINED LIST USING $
                                     /

0201          START
0202  1X ILLEGAL CHAR
0202  X1      BSZ      2          /TO CHECK WHAT AN OPCODE IS HIT =4432
                                     /1X WAS AN ILLEGAL CHARACTER BECAUSE /
                                     /IT STARTED WITH A NUMBER, MUST START /
                                     /WITH A LETTER.

0204          OCT      -1        /=7777 HERE ARE EXAMPLES OF THE USE
0205          DEC      10        /=0012 OF OCT AND DEC.
0206          DEC      -10       /=7766
0207          OCT      1         /=0001 IF DEC IS TOO LARGE:
0210          DEC      4096      4096 TOO MANY!
0210          DEC      -4096     -4096 TOO MANY!
0210          DEC      A         A ILLEGAL CHAR
0210          OCT      A         A ILLEGAL CHAR
0210          OCT      8         8 ILLEGAL CHAR
0210          OCT      1234567  /=4567 ONLY THE LAST 4 NUMBERS ARE USED/
                                     /ONLY NUMBERS CAN BE USED IN THE ADDRESS/
                                     /FIELD WHEN USING DEC, OCT OR BSZ.

0211  A      BSZ      A ILLEGAL CHAR
0211  A      BSZ      1         /
0212          JMP*     A         /=5611 USING A AS AN ADDRESS.
0213          JMP      A         /=5211
0214          JMP      A-10      /=5201 USING A AS AN ADDRESS THEN /
                                     /MODIFYING THE ADDRESS.
0215          JMP      .+10     /=5225 USING . + OR .-
0216          ORG      600      /TO START AT A DIFFERENT ADDRESS.

0600  S      EQU      A         /=0211 THIS CAN BE USED FOR LINKING.
0601          TAD*     S         /=1600 HERE WE CAN GET THE CONTENTS /
                                     /OF A USING LINKING.
0602          /A DOLLAR SIGN CAN BE USED TO CLEAR /
                                     /THE DEFINED LIST BUT KEEP THE OPCODES /
                                     /THAT WERE DEFINED. $
0602  S      TYP
                                     /=4430 TO CLEAR ALL LISTS USE +

0200  S      TYP   UNDEFINED
0200  A      DCA
NOT FINISHED!
0200
                                     /HERE THE RETURN WAS HIT BEFORE THE /
                                     /COMMENTS FIELD WAS REACHED.

```

0201

BIN

/IF A LETTER IS HIT INSTEAD OF A NUMBER/  
/THE TTY WILL TYPE SA: AND START /  
/THE BLOCK AGAIN.

SA: 1  
FA: 2  
SA: 2  
FA: A  
SA: A  
SA: -

0201

RIM

/IF THE RUBOUT KEY IS HIT THE LINE /  
/IS RE-ENTERED. /  
/IF MORE THAN 10 DEC BLOCKS ARE TYPED /  
/AN ERROR IS FLAGGED AND THE LINE /  
/RE-ENTERED.

SA: 0  
FA: 1  
SA: 0  
FA: 2  
SA: 0  
FA: 3  
SA: 0  
FA: 4  
SA: 0  
FA: 5  
SA: 0  
FA: 6  
SA: 0  
FA: 7  
SA: 0  
FA: 10  
SA: 0  
FA: 11  
SA: 0  
FA: 12 TOO MANY!

0201

/THE FOLLOWING IS AN EXAMPLE OF END.

0202 TAD A  
0203 DCA B  
0204 EQU A  
0205 EQU U  
0206 END

A 0202 UNDEFINED  
B 0203 UNDEFINED

A 0204 UNDEFINED  
U 0205 UNDEFINED

0206 A BSZ 1  
0207 B BSZ 1  
0210 U BSZ 1

/FIRST THE UNDEFINED LIST IS TYPED,  
/THEN THE LINK LIST IS TYPED.  
/WHEN ALL LISTS ARE DEFINED, DONE /  
/IS TYPED AND THE LINE IS RE-ENTERED.

0211 END

DONE

0211

## A P P E N D I X B

### Processor Instructions

|      |     |                            |
|------|-----|----------------------------|
| 0000 | AND | Logican AND                |
| 1000 | TAD | Two's complement ADD       |
| 2000 | ISZ | Increment and skip if Zero |
| 3000 | DCA | Deposit and clear AC       |
| 4000 | JMS | Jump to subroutine         |
| 5000 | JMP | Jump to address            |

### Group 1, Operate Instructions

|      |     |                                       |
|------|-----|---------------------------------------|
| 7000 | NOP | No operation                          |
| 7001 | IAC | Increment accumulator                 |
| 7002 | RAT | Rotate accumulator two positions      |
| 7004 | RAL | Rotate accumulator left               |
| 7006 | RTL | Rotate two left                       |
| 7010 | RAR | Rotate accumulator right              |
| 7012 | RTR | Rotate two right                      |
| 7020 | CML | Complement link                       |
| 7040 | CMA | Complement accumulator                |
| 7041 | CIA | Complement and increment accumulator  |
| 7100 | CLL | Clear link                            |
| 7120 | STL | Set link                              |
| 7200 | CLA | Clear accumulator                     |
| 7204 | GLK | Transfer link to AC bit 11 (CLA, RAL) |
| 7240 | STA | Set accumulator to all ONES           |
| 7300 | CRA | Clear accumulator and link            |

### Group 2, Operate Instructions

|      |     |                                       |
|------|-----|---------------------------------------|
| 7402 | HLP | Halt                                  |
| 7404 | OSR | Or Switch register to accumulator     |
| 7410 | SKP | Skip, unconditional                   |
| 7420 | SNL | Skip on non-zero link                 |
| 7430 | SZL | Skip on zero link                     |
| 7440 | SZA | Skip on zero accumulator              |
| 7450 | SNA | Skip on non-zero accumulator          |
| 7500 | SMA | Skip on minus accumulator             |
| 7510 | SPA | Skip on positive accumulator          |
| 7604 | LAS | Load accumulator from switch register |

### Miscellaneous

|      |     |                                   |
|------|-----|-----------------------------------|
| 0000 |     | Blank op code and variable fields |
| 6001 | ION | Interrupt turn on                 |
| 6002 | IOF | Interrupt turn off                |
| 6004 | ADC | Analog to digital convert         |

### Extended Arithmetic Instructions

|      |     |                                    |
|------|-----|------------------------------------|
| 7405 | MUY | Multiply                           |
| 7407 | DVI | Divide                             |
| 7411 | NMI | Normalize                          |
| 7413 | SHL | Shift arithmetic left              |
| 7415 | ASR | Arithmetic shift right             |
| 7417 | LSR | Logical shift right                |
| 7421 | MZL | Load multiplier quotient           |
| 7441 | SCA | Step counter load into accumulator |
| 7501 | MQA | Load accumulator from MQ           |

### Teletype Keyboard-Printer

|      |     |                              |
|------|-----|------------------------------|
| 6031 | KSF | Skip on keyboard flag        |
| 6032 | KCC | Clear keyboard flag          |
| 6034 | KRS | Read keyboard buffer static  |
| 6036 | KRB | Read keyboard buffer dynamic |
| 6041 | TSP | Skip on teleprinter flag     |
| 6042 | TCF | Clear teleprinter flag       |
| 6044 | TPC | Load teleprinter and printer |
| 6046 | TLS | Load teleprinter sequence    |

### Oscilloscope Display Control

|      |     |                                    |
|------|-----|------------------------------------|
| 6051 | DCX | Clear X coordinate buffer          |
| 6053 | DXL | Clear and load X coordinate buffer |
| 6054 | DIX | Intensify                          |
| 6061 | DCY | Clear Y coordinate buffer          |
| 6063 | DYL | Clear and load Y coordinate buffer |
| 6064 | DIY | Intensify                          |
| 6071 | DCL | Clear X, Y, buffer                 |
| 6072 | PLS | Generate a pulse                   |
| 6074 | DRS | Erase storage scope, CLA           |

## Floating Point Package

|      |     |      |     |
|------|-----|------|-----|
| 4447 | ENT | 0001 | SQR |
| 4445 | INP | 0002 | SRT |
| 4446 | OUT | 0003 | SIN |
| 4467 | FIX | 0004 | COS |
| 4470 | FLA | 0005 | ATN |
| 4471 | LIN | 0006 | LOG |
|      |     | 0007 | EXP |
| 1000 | ADD |      |     |
| 2000 | SUB | 0010 | ABS |
| 3000 | MUY | 0012 | CR  |
| 4000 | DIV | 0013 | RED |
| 5000 | GET | 0014 | PRT |
| 6000 | PUT | 0015 | NEG |
| 7000 | NOR | 0016 | ZAC |
| 0000 | EXT | 0017 | NON |

## Pseudo Operations

|     |   |
|-----|---|
| ORG | Set location counter (octal)                  |
| FQU | Define absolute address                       |
| DEF | Op code definition (octal)                    |
| DEC | Signed or unsigned decimal constant           |
| OCT | Octal constant                                |
| ALF | Two TTY characters                            |
| END | End of program, check undefined and link list |
| BSZ | Block storage zero of length N (octal)        |
| RIM | Punch rim tape                                |
| BIN | Punch bin tape                                |

A P P E N D I X C

| <u>Character</u> | <u>8-Bit Code</u> | <u>LRL PDP Code</u> | <u>Character</u> | <u>8-Bit Code</u> | <u>LRL PDP Code</u> |
|------------------|-------------------|---------------------|------------------|-------------------|---------------------|
| BLANK            | 240               | 00                  | A                | 301               | 41                  |
| !                | 241               | 01                  | B                | 302               | 42                  |
| "                | 242               | 02                  | C                | 303               | 43                  |
| #                | 243               | 03                  | D                | 304               | 44                  |
| \$               | 244               | 04                  | E                | 305               | 45                  |
| %                | 245               | 05                  | F                | 306               | 46                  |
| &                | 246               | 06                  | G                | 307               | 47                  |
| '                | 247               | 07                  | H                | 310               | 50                  |
| (                | 250               | 10                  | I                | 311               | 51                  |
| )                | 251               | 11                  | J                | 312               | 52                  |
| *                | 252               | 12                  | K                | 313               | 53                  |
| +                | 253               | 13                  | L                | 314               | 54                  |
| ,                | 254               | 14                  | M                | 315               | 55                  |
| -                | 255               | 15                  | N                | 316               | 56                  |
| .                | 256               | 16                  | O                | 317               | 57                  |
| /                | 257               | 17                  | P                | 320               | 60                  |
| Ø                | 260               | 20                  | Q                | 321               | 61                  |
| 1                | 261               | 21                  | R                | 322               | 62                  |
| 2                | 262               | 22                  | S                | 323               | 63                  |
| 3                | 263               | 23                  | T                | 324               | 64                  |
| 4                | 264               | 24                  | U                | 325               | 65                  |
| 5                | 265               | 25                  | V                | 326               | 66                  |
| 6                | 266               | 26                  | W                | 327               | 67                  |
| 7                | 267               | 27                  | X                | 330               | 70                  |
| 8                | 270               | 30                  | Y                | 331               | 71                  |
| 9                | 271               | 31                  | Z                | 332               | 72                  |

| <u>Character</u> | <u>8-Bit Code</u> | <u>LRL PDP Code</u> | <u>Character</u> | <u>8-Bit Code</u> | <u>LRL PDP Code</u> |
|------------------|-------------------|---------------------|------------------|-------------------|---------------------|
| :                | 272               | 32                  | ☐                | 333               | 73                  |
| ;                | 273               | 33                  | ↘                | 334               | 74                  |
| <                | 274               | 34                  | ☐                | 335               | 75                  |
| =                | 275               | 35                  | ↑                | 336               | 76                  |
| >                | 276               | 36                  | ←                | 337               | 77                  |
| ?                | 277               | 37                  | LINE FEED        | 212               |                     |
| @                | 300               |                     | RETURN           | 215               |                     |

## A P P E N D I X D

### (PDP-8 FLOATING POINT SYSTEM SUMMARY)

#### Entry Points

|      |     |  |
|------|-----|--|
| 4447 | ENT | Enter arithmetic interpreter           |
| 4445 | INP | Floating input                         |
| 4446 | OUT | Floating output                        |
| 4467 | FIX | Convert FAC to integer, leave in AC    |
| 4470 | FLA | Convert (AC) to floating, leave in FAC |
| 4471 | LIN | Type carriage return and line feed     |

#### Interpreter Commands (Memory Reference)

|   |     |     |                            |             |       |
|---|-----|-----|----------------------------|-------------|-------|
| 1 | XXX | ADD | Floating add               | (XXX)+(FAC) | (FAC) |
| 2 | XXX | SUB | Floating subtract          | (FAC)-(XXX) | (FAC) |
| 3 | XXX | MPY | Floating multiply          | (FAC)*(XXX) | (FAC) |
| 4 | XXX | DIV | Floating divide            | (FAC)/(XXX) | (FAC) |
| 5 | XXX | GET | Load floating accumulator  | (XXX)       | (FAC) |
| 6 | XXX | PUT | Store floating accumulator | (FAC)       | (XXX) |
| 7 | XXX | NOR | Normalize FAC              |             |       |

#### Interpreter Commands (Non-memory Reference)

In all cases, the result is left in FAC, original FAC is lost.

|      |     |  |
|------|-----|--|
| 0000 | EXT | Exit from interpretive mode                |
| 0001 | SQR | Square FAC                                 |
| 0002 | SRT | Take square root of FAC                    |
| 0003 | SIN | Take sine of FAC (radians)                 |
| 0004 | COS | Take cosine of FAC (radians)               |
| 0005 | ATN | Take arctangent of FAC (result in radians) |
| 0006 | LOG | Take natural log of FAC                    |
| 0007 | EXP | Raise e to the (FAC) power                 |
| 0010 | ABS | Set FAC positive                           |
| 0012 | CR  | Type carriage return and line feed         |
| 0013 | RED | Floating input                             |
| 0014 | PRT | Floating output                            |
| 0015 | NEG | Complement FAC                             |
| 0016 | ZAC | Zero FAC                                   |
| 0017 | NON | No operation                               |

### Flags and Control Words

|      |   |
|------|---|
| 0054 | Floating accumulator exponent                 |
| 0055 | Floating accumulator high order               |
| 0056 | Floating accumulator low order                |
| 0064 | Contains the last terminator used on input    |
| 0065 | Equals zero if no valid input                 |
| 0066 | Arithmetic error flag                         |
| 0072 | Output field length ( $\emptyset$ = E format) |
| 0073 | Length of fractional field when F format used |

### Memory Locations used by Flopak

|           |                        |
|-----------|------------------------|
| 16, 17    | Index registers        |
| 45-75     | Flags and pointers     |
| 3600-6577 | Floating point package |

