

RECOMP II USERS' PROGRAM NO. 1033

PROGRAM TITLE: SIGNAL CORPS RECOMP ASSEMBLY PROGRAM - SCRAP II

PROGRAM CLASSIFICATION: Executive and Control

AUTHOR: T. J. Tobias
U. S. Army Signal Engineering Agency
Arlington Hall Station
Arlington, Virginia

This program was modified and adapted for use with a subroutine package - RECOMP II Program No. 50 - P³-2 by L. Raphael and A. W. England, and the text was rewritten by R. S. Lynn of Autonetics Industrial Products.

CHECKED BY: R. S. Lynn and H. D. Goddard

PURPOSE: SCRAP is an assembly program for the RECOMP II computer. It is designed to use mnemonic operation codes and symbolic, absolute, operand, or relative addresses. SCRAP uses all of the RECOMP II commands, as well as pseudo-operation codes and macro-instructions. The SCRAP processor requires two passes to complete the assembly of a program written in the SCRAP language. The primary working media of the SCRAP processor during assembly is paper tape for both input and output data. Printed output is optional during both the first and second passes of assembly.

DATE: January 1960

REISSUED: January 1961

Published by

RECOMP Users' Library

at

AUTONETICS INDUSTRIAL PRODUCTS

A DIVISION OF NORTH AMERICAN AVIATION, INC.
3400 E. 70th St., Long Beach 5, Calif.

T A B L E O F C O N T E N T S

INTRODUCTION

| | | |
|----|--------------------------------------|------------|
| 1. | WRITING INSTRUCTIONS IN SCRAP FORMAT | Page 1 - 4 |
| 2. | OPERATING INSTRUCTIONS | Page 4 - 7 |
| 3. | RESTRICTIONS | Page 7 - 8 |
| 4. | PROGRAMMED HALTS | Page 8 - |

APPENDIX A

RECOMP II COMMANDS

APPENDIX B

| | | |
|----|-------------------------|------------|
| I | PSEUDO OPERATIONS CODES | Page 1 - 2 |
| II | STANDARD MACROS | Page 2 - 4 |

SIGNAL CORPS RECOMP ASSEMBLY PROGRAM

SCRAP II

INTRODUCTION

SCRAP allows the programmer to write in symbolic language, leaving the bookkeeping, etc., associated with machine language programming to be done by the computer. The assembly requires two passes. On the first pass, bookkeeping and location assignment is accomplished. On the second pass, all instructions are punched in the proper machine language, ready to function in conjunction with PPP-2, RECOMP II Program No. 50.

1. WRITING INSTRUCTIONS IN SCRAP FORMAT

Writing instructions in SCRAP format is very similar to coding in symbolic language prior to assembling "by hand," with some added advantages in flexibility. Each instruction contains four fields: LOCATION, COMMAND, TYPE, and ADDRESS.

1.1 LOCATION

The location field may contain a location tag. This tag may be from one to eight alphabetic characters only. The location field may be left blank.

1.2 COMMAND

The command field contains either a standard RECOMP II operation code mnemonic, a pseudo-operation code or a macro-instruction code. The standard RECOMP II commands are listed in Appendix A. Pseudo-operations and macros which are defined in the standard revised SCRAP assembly program are listed in Appendix B.

1.3 TYPE

The type field specifies the type of address in the address field. Six types of addresses are used in SCRAP. A single letter code is used in this field.

The codes and their meanings are:

| | |
|---|---------------------------|
| S | Symbolic |
| N | Numeric address, absolute |
| F | Floating point decimal |
| A | Alphabetic |
| D | Decimal, fixed point |
| C | Command format |

If the type field is left blank, it is interpreted as symbolic.

1.4

ADDRESS

The address field contains one of the six types of addresses.

SYMBOLIC

A symbolic location is one to eight alphabetic characters. SCRAP interprets the single letter C to be the current location. Addresses **before** and **after** a symbolic location may be addressed as in these examples. Note that RECOMP II instructions are a half-word long, so that 2 instructions fill a word.

| | |
|-------------|--------------------------------------------|
| C+1 | meaning one word after current location |
| EXIT-1 | one word before location called EXIT |
| START+00001 | one half-word after START |
| ABLE-00021 | two and a half words before ABLE |
| DELTA+34 | thirty-four words after DELTA |
| C-3 | three words before the current instruction |
| THETA | the location named theta |

NUMERIC ADDRESS, ABSOLUTE

The numeric address may be from one to five Arabic numerals. If five numerals are in this field, they are interpreted as the usual RECOMP II address, with 4 octal digits for channel and sector plus a half-word bit. If there are fewer than five numerals, the address is interpreted as a full word address with a half word bit of zero, and enough leading zeros to right justify the numerals. For example,

| | | |
|-------|-------|--------|
| 7760 | means | 7760.0 |
| 760 | | 0760.0 |
| 13 | | 0013.0 |
| 34711 | | 3471.1 |
| 00001 | | 0000.1 |
| 0 | | 0000.0 |

FLOATING POINT DECIMAL

A floating point decimal number used in the address field may contain a maximum of sixteen characters including the sign and the decimal point. For example:

-100 would be assembled as:

-6200000-0000000
+0000000-0000031

Neither the integer part nor the fractional part may contain more than eleven characters. The sign will be interpreted as plus if no sign is given.

ALPHABETIC

An address field may contain up to eight alphanumeric characters. SCRAP generates the Baudot code for the characters in the address field, inserting leading blanks to complete a total of eight characters, and stores this in a location called an ALFCN. The book-keeping is automatic and the user need not thereafter define that particular alphanumeric word. No carriage return, tab or line feed may be in the address field. Letter shift (L/S) and figure shift (F/S) may be used. For example:

| | | |
|-----|---|------------------------|
| CLA | A | L/S X F/S + L/S Y F/S: |
| TYA | N | 7770 |

These instructions in the final program will cause X+Y: to be typed.

DECIMAL, FIXED POINT

A fixed point decimal number used in the address field may contain a maximum of sixteen characters, counting the sign, decimal point, and location of the binary point. Neither the integral nor the fractional part may contain more than eleven digits. For example:

| | | |
|-------------|-------|------------------------------|
| -1142857+18 | means | -1142,857 at binary point 18 |
| 4+38 | | +4 at binary point 38 |
| -3.335-4 | | -3.335 at binary point (-4). |
| .25+0 | | +.25 at binary point 0. |

The fractional or integral parts may be omitted if they are not needed. If there is no sign, it is taken to be positive. If the location of a binary point is omitted, the number will be converted to a floating point number without an exponent, and assigned only one word of storage. There will be no error halt for this condition.

If the specified binary point would allow the loss of significant high order (left hand) bits, the number will not be converted properly. There will be no error halt for this condition.

COMMAND

A word in command format may be used in the address field. The word is assembled exactly as typed in, if it is the proper format. There is no indication of error in format.

2. OPERATING INSTRUCTIONS

2.1 FIRST PASS PROCEDURES

2.1.1 SENSE SWITCH SETTINGS

Set Sense Switch B on for a printed copy of first pass assembly. Set B off if no printout is desired at this time.

Set Sense Switch C on to type in SCRAP instructions. Set C off to input a SALT-punched tape, or a key-punched tape.

Set Sense Switch D on to permit assignment of absolute addresses for symbolic locations. The assignment occurs after the END instruction if D is on. If D is off, no assignment will be made. After END FIRST PASS has been typed, the assignment may be accomplished by setting D on, and with location counter set at 4513, pressing the start button.

It is possible to go through the steps of the first pass without producing any assembly. This is called key-punching. This is an extra phase which lengthens the total assembly time. To execute this option, set both B and C on. The tape which is produced is similar to a SALT-punched tape, and is processed in the same way. It must be processed by the first pass. It is not necessary to key-punch.

2.1.2 INITIAL PROCEDURE

- (1) Set Sense Switches
- (2) Set typewriter margins at 10 and 90. Set tab stops at 20, 35, and 50. These stops may be set closer or further apart, but space must be allowed for each field to be typed properly.
- (3) Advance at least two folds of blank tape.
- (4) Load SCRAP tape. Wait about six seconds for the memory to be zeroed by a short program at the start of the tape, followed by the reading of the assembly program. The tape may be verified by placing the tape in the photoreader past the memory zero program and pressing the Verify button.
- (5) Press start 1.

2.1.3. TYPING INSTRUCTIONS

- (1) Type location field and tab. This field may be blank.
- (2) Type command field and tab. This field may never be blank except after a GOTOSALT location field entry used during SALT program.
- (3) Type the code for type of address and tab. This field may be blank if the address is symbolic, or if certain pseudo operation codes are used in the command field.
- (4) Type address field and carriage return. The punch will output the instruction. The address field is blank for certain pseudo operations.
- (5) Repeat steps (1) through (4) for each instruction in the program. The last line must be END.
- (6) If an error is typed and detected before the carriage return, press the line feed key to delete the instruction.
- (7) If the output error light goes on because of typing too fast, press in sequence the error reset button, start 1, and line feed. Retype the line as in steps (1) through (4).
- (8) Corrections, insertions, and deletions may be accomplished during a re-assembly of the first pass, according to procedures outlined in §2.3.
- (9) It is possible to patch the program during the first pass by writing instructions which will nullify the errors made.

2.1.4 PUNCHING ASSIGNMENT TABLE

The Assignment Table and other information necessary for assembly is punched by the SAVE routine, which is included in the SCRAP tape. The Assignment Table contains two words for each symbolic location. The first word of the pair is the name of the location in alpha format, and the second word is the memory assignment in command format in the right half of the word.

- (1) After END FIRST PASS has been typed, advance several folds of blank tape and press start 3, which starts the SAVE routine.
- (2) If the first and second passes are to be run at different times, the use of SAVE is required.
- (3) If it is intended to run the first and second passes in sequence, without disturbing memory, the use of SAVE, while not required, is provident. If there is a mis-punch or a mis-read during second pass, the second pass should be re-run from step 2.2 (1), using the SAVE material.

2.2 SECOND PASS PROCEDURES

If the contents of memory have not been changed since END FIRST PASS, proceed from step (3).

- (1) Load and verify SCRAP tape in the manner described in §2.1.2.4. Load and verify SAVE material on the end of the first pass tape.
- (2) Set typewriter margins at 10 and 90. Set tab stops at 20, 35, and 50.
- (3) Place first pass output tape in photoreader.
- (4) Set Sense Switch B on to obtain printed copy of the symbolic and absolute machine language program. If B is off, there will be no printout. Set Sense Switch C off. Sense Switch D is not tested, and has no effect.
- (5) Press start 2. The second pass will punch the final program in machine language.

2.3 CORRECTION PROCEDURES

Changes in the program may be made during the first pass of the assembly of SALT-punched or key-punched tape, or during a re-assembly of the first pass output tape. For a re-assembly of a first pass output tape, load SCRAP, including the memory zero routine, before placing first pass tape in photoreader. To change the program, the user must input symbolic machine language to achieve the desired results. Re-assembly printout will be the same as the second pass symbolic language.

For all changes, the read-out dials are set to 4600. Set Sense Switch B on, C off. After all changes have been made, B may be turned off. Press start 1 to begin re-assembly.

2.3.1 DELETION

To delete a line of coding from the first pass,

- (1) Set preset stop switch to 2ND before the line to be deleted has been printed.
- (2) After a line has been printed, press start button to punch it and keep it in the program.
- (3) If the line which has been printed is to be deleted, return typewriter carriage and turn platen manually, then press start 1. The next line will be printed at once. Repeat steps (2) and (3) for further deletions.
- (4) To resume assembly, set preset stop switch to off and press start button.

2.3.2 INSERTION

To insert a line of coding,

- (1) Set preset stop switch to 1ST before the desired point of insertion.
- (2) If no insertion is to be made when the computer stops, press start button to continue.
- (3) When the computer stops at the point of insertion, set B off, C on, preset stop off and press start button.
- (4) Type insertions in the format specified in § 2.1.3. The print-out of the re-assembly omits the T field, and is in the format of the second pass. Insertions cannot omit the T field.
- (5) After the last insertion to be made, type (tab) PAUSE (tab) (tab) (carriage return).
- (6) Set B on, C off. Press start 1 to continue assembly.

2.3.3 CORRECTION

A correction involves a deletion and insertion.

- (1) Set preset stop switch to 2ND before the line to be corrected has been printed.
- (2) If no correction is to be made when the computer halts, press start button to continue.
- (3) When the computer halts after the line to be corrected is printed, set B off, C on, preset stop switch to off. Press start 1.
- (4) Type the correct line in the format required for first pass typing, including the T field. Repeat step (4) to insert additional statements after the correction.
- (5) Type (tab) PAUSE (tab) (tab) (carriage return).
- (6) Set B on, C off. Press start 1 to continue assembly.

2.3.4 ASSIGNMENT TABLE CORRECTION

If there was a misspelling of a symbolic location in SALT or first pass SCRAP, made by the user, an improper location will be assigned to it in the Assignment Table. Both the proper spelling and the misspelling will appear in the Assignment Table. This may be corrected by manually replacing the assignment for the misspelled symbolic location. Different symbols properly spelled may be made equivalent in the same manner.

The Assignment Table begins at location 0500. Search the table using the visual readout to locate the misspelled word pair. Enter the proper location to replace the improper assignment. This procedure will avoid re-assembly for minor spelling errors.

3. RESTRICTIONS

3.1 In the SCRAP program, there may be a maximum of:

- (1) 512 symbolic names
- (2) 256 constants and no more than 99 of any one type - A, D, F and C.

3.2 The following letter combinations may not be used as the name of a location or address:

- (1) The single letter C, except when it is used to refer to the current address.
- (2) The name of any macro.
- (3) The name of any machine operation code or any pseudo-operation code.

3.3 If CIA A alpha name is inserted as a change, the alphanumeric will not be properly assembled.

4. PROGRAMMED HALTS

| <u>LOCATION</u> | <u>DISPLAY</u> | <u>ERROR CONDITION</u> | <u>CORRECTIVE ACTION</u> |
|-----------------|-----------------|----------------------------------------------|----------------------------------------------------------------------------------------------------------------|
| 4334 | 01 | More than 512 symbols | No immediate action. Segment program. |
| 7777 | 02 | Search error of constant table | Restart at 4345.1 to try again. |
| 7777 | 03 | More than 256 constants | No immediate action. Segment program. |
| 4423.1 | 04 | Location counter greater than 7757.1 | Change ORG or segment program. |
| 4600 | ILLEGAL OP CODE | Code not defined | Delete or correct. |
| 4720 | None | More than 99 constants of one type | No immediate action. A restart at 4724.1 will ignore constant, but will cause a halt in second pass at 6545.1. |
| 6540.1 | None | No absolute assignment for symbolic location | To ignore, press start. |

| <u>LOCATION</u> | <u>DISPLAY</u> | <u>ERROR CONDITION</u> | <u>CORRECTIVE ACTION</u> |
|-----------------|----------------|---------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| 6545.1 | None | No absolute assignment for symbolic address | To ignore, press start. Address of assembled instruction will be 0000. |
| 0000.0 | None | Tape was mispunched or misread | It may be possible to read the tape again by moving the tape back to the last gap and restarting at 5311.0. |

APPENDIX A, SCRAP DESCRIPTION

RECOMP II COMMANDS

Symbols: W = memory location whose address is W, half word bit,
 either 1 or 0

 W.1 = same as W, but with a 1 in the half word bit of address

 W.0 = same as W, but with a 0 in the half word bit of address

 () = contents of

 A = A-register

 R = R-register

 X = X-register

 (W)_a = address portion of memory location W

 (A)_a = address portion of the contents of the A-register

 (A)_s = sign portion of the contents of the A-register

 → means replaces

Operation Symbol

| | |
|---------|----------------------------------------------------------------|
| CLA W.0 | Clear and add. (W) → (A) |
| CAM W.1 | Clear and add magnitude. (W) → (A) |
| ADD W.0 | Add. (W) + (A) → (A) |
| ADM W.1 | Add magnitude. (W) + (A) → (A) |
| CLS W.0 | Clear and subtract. - (W) → (A) |
| CSM W.1 | Clear and subtract magnitude. - (W) → (A) |
| SUB W | Subtract. (A) - (W) → (A) |
| SBM W.1 | Subtract magnitude. (A) - (W) → (A) |
| MPR W | Multiply and round. (W) x (A) → (A) |
| MPY W | Multiply (unrounded). (W) x (A) → (A,R) |
| DSR W | Divide single length and round. (A) / (W) → (A) |
| SRM W.1 | Divide single length and round by magnitude. (A) / (W) → (A) |

| | |
|--------------|------------------------------------------------------------------------------------------------------------|
| DSL W | Divide single length (unrounded). $(A) / (W) \rightarrow (A)$, remainder $\rightarrow (R)$ |
| DSM W.1 | Divide single length (unrounded) by magnitude. $(A) / (W) \rightarrow (A)$, remainder $\rightarrow (R)$ |
| DVR W | Divide (double length) and round. $(A,R) / (W) \rightarrow (A)$ |
| DRM W.1 | Divide (double length) and round by magnitude. $(A,R) / (W) \rightarrow (A)$ |
| DIV W | Divide (double length unrounded). $(A,R) / (W) \rightarrow (A)$, remainder $\rightarrow (R)$ |
| DVM W.1 | Divide (double length unrounded) by magnitude. $(A,R) / (W) \rightarrow (A)$, remainder $\rightarrow (R)$ |
| STO W | Store. $(A) \rightarrow (W)$ |
| **STA W | Store address. $(A)_a \rightarrow (W)_a$ |
| ***ARS N_8 | Accumulator right shift. Shift (A) N_8 places to the right, sign not shifted. |
| ***ALS N_8 | Accumulator left shift. Shift (A) N_8 places to the left, sign not shifted. |
| SQR W | Square root. $\sqrt{(W)} \rightarrow (A)$. Binary point assumed to be even |
| SQM W.1 | Square root magnitude. $\sqrt{ (W) } \rightarrow (A)$. Binary point assumed to be even. |
| FCA W | Floating clear and add. $(W,W+1) \rightarrow (A,R)$ |
| FAD W | Floating add. $(W,W+1) + (A,R) \rightarrow (A,R)$ |
| FAM W.1 | Floating add magnitude. $((W) , W+1) + (A,R) \rightarrow (A,R)$ |
| FCS W | Floating clear and subtract. $-(W,W+1) \rightarrow (A,R)$ |
| FSB W | Floating subtract. $(A,R) - (W,W+1) \rightarrow (A,R)$ |
| FSM W.1 | Floating subtract magnitude. $(A,R) - ((W) , W+1) \rightarrow (A,R)$ |
| FMP W | Floating multiply. $(W,W+1) \times (A,R) \rightarrow (A,R)$ |
| FMM W.1 | Floating multiply magnitude. $((W) , W+1) \times (A,R) \rightarrow (A,R)$ |
| FDV W | Floating divide. $(A,R) / (W,W+1) \rightarrow (A,R)$ |
| FDM W.1 | Floating divide by magnitude. $(A,R) / ((W) , W+1) \rightarrow (A,R)$ |
| FST W | Floating store. $(A,R) \rightarrow (W,W+1)$ |

Operation
Symbol

(Continued)

| | |
|------------|---------------------------------------------------------------------------------------------|
| FSQ W | Floating square root. $\sqrt{(W,W + 1)} \rightarrow (A,R)$ |
| FNM | Floating normalize. (A,R) normalized. Address immaterial. Binary scale must be in R, at 39. |
| *RDY N | Read N characters from typewriter. |
| RYF W | Set location counter to W, and read typewriter in Fill Mode. |
| *RDZ W | Read N characters from tapereader. |
| RZF W | Read tapereader in Fill Mode. |
| DIS W | Display. Display (W) in Fill Mode (Command Format). |
| DSD W.1 | Display decimal, BCD. |
| ****PNC B | Punch character. Punch character corresponding to Baudot code B. |
| *PNA N | Punch N characters in Alpha Mode. |
| ****TYC B | Type character. Type character corresponding to Baudot code B. |
| *TYA N | Type N characters in Alpha. |
| ****PTC B | Punch and type character. Punch and type character corresponding to Baudot code B. |
| *PTA N | Punch and type N characters in Alpha Mode. |
| PNW W | Punch word. Punch (W). |
| PWD W.1 | Punch word. Decimal, BCD. |
| TYW W | Type word. Type (W). |
| TWD W.1 | Type word. Decimal, BCD. |
| PTW W | Punch and type word. Punch and type (W). |
| PTD W.1 | Punch and type word. Decimal, BCD. |
| *1 ≤ N ≤ 8 | Address must be 776N for N = 1-7 and 7760 for N = 8. Half-word bit must be = 0. |
| **STA W.1 | stores right half address of A to right half address of W. |
| STA W.0 | stores left half address of A to left half address of W. |
| | Storing from right to left or vice-versa not permitted. |
| ***N | must be in the sector portion of the address. Channel portion immaterial. |
| ****B | must be in the sector portion of the address. Channel portion immaterial. |

Operation
Symbol (Continued)

| | |
|--------|--------------------------------------------------------------------------------------------|
| #CTL W | Copy to L loop. (W) to (W + 7) → (7760) to (7767). |
| #CFL W | Copy from L loop. (7760) to (7767) → (W) to (W + 7). |
| #CTV W | Copy to V loop. (W) to (W + 7) → (7770) to (7777). |
| #CFV W | Copy from V loop. (7770) to (7777) → (W) to (W + 7). |
| TZE W | Transfer on zero. If (A) = ± 0, transfer to W. |
| TPL W | Transfer on plus. If (A) _S = +, transfer to W. |
| TMI W | Transfer on minus. If (A) _S = -, transfer to W. |
| TOV W | Transfer on overflow. If overflow occurs, transfer to W. |
| TRA W | Transfer (unconditional). Transfer to W. |
| TSB W | Transfer on switch B. If on, transfer to W; if off, go to next instruction. |
| TSC W | Transfer on switch C. If on, transfer to W; if off, go to next instruction. |
| TSD W | Transfer on switch D. If on, transfer to W; if off, go to next instruction. |
| HTR W | Halt and transfer. Stop computer and reset location counter to W. |
| EXT W | Extract. Make zero all bits in A except those in which corresponding bits in W are "ones." |
| XAR | Exchange A and R. Interchange (A) and (R). Address immaterial. |
| SAX W | Store A and exchange A and X. (A) → (W), interchange (A) and (X). |

#The address W should end with 0.0 for simplicity in coding. In using the SCRAP assembly, this is done by an SB pseudo-operation. See RECOMP II Operation Manual for detail on COPY commands.

APPENDIX B, SCRAP DESCRIPTION

I. PSEUDO OPERATION CODES

In the following definitions, L field stands for Location field, C field stands for Command field, T field stands for Type field, and A field stands for Address field.

| <u>CODE</u> | <u>DEFINITION</u> |
|-------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ORG | <p><u>ORIGIN</u> The next instruction after the ORG line is assigned the address specified by the A field of the ORG line. T field must contain N. Example:</p> |

```

                ORG   N   3000
START          FCA   F   1466
  .            .    .    .
  .            .    .    .
  .            .    .    .
    
```

In this example, the command line START will be assigned absolute location 3000.

| | |
|-----|-------------------------------------------------------------------------------------------------------------------------|
| DEF | <p><u>DEFINITION</u> The symbolic location in the L field is assigned the absolute address in the A field. Example:</p> |
|-----|-------------------------------------------------------------------------------------------------------------------------|

```

GOGO   DEF   N   4400
    
```

Symbolic location GOGO is assigned absolute location 4400.0. The T field must be N.

| | |
|----|----------------------------------------------------------------------------------------|
| SL | <p><u>SET LEFT</u> The next instruction is assembled in the left half of the word.</p> |
|----|----------------------------------------------------------------------------------------|

| | |
|----|------------------------------------------------------------------------------------------|
| SR | <p><u>SET RIGHT</u> The next instruction is assembled in the right half of the word.</p> |
|----|------------------------------------------------------------------------------------------|

| | |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SB | <p><u>SET BLOCK</u> The next instruction is assembled in the left half of the next word whose absolute location ends in 0.0. This is usually done to permit a proper copy to the L or V loops.</p> |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| | |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| END | <p><u>END</u> This marks the end of the coding to be assembled. No other information may follow. No instructions are generated. A halt code is punched on the tape.</p> |
|-----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

| | |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| PAUSE | <u>PAUSE</u> This causes the assembly to stop at this point in the first pass. The PAUSE code is used in insertions and corrections. See § 2.3. If a PAUSE code was used in SALT, and if it is desired to resume the assembly, press the start button. |
| ALF or ALPHA | <u>ALPHANUMERIC</u> The information in the A field is assembled in Baudot code and assigned the symbolic location in the L field. T field must be A. |
| DEC or DECIMAL | <u>DECIMAL</u> The decimal information in the A field is assembled as either floating or fixed point, according to the T field, which must be either F or D. The information so assembled is assigned the symbolic location in the L field. |
| COM or COMMAND | <u>COMMAND</u> The command format in the A field is assigned the symbolic location in the L field. The T field must be C. |
| HALT | Same as HTR |
| HLT | Same as HTR |
| DISC | DIS Display in command format |
| DISD | DSD Display in BCD format |
| TYWC | TYW Type word in command format |
| TYWD | TWD Type word in BCD format |
| PNWC | PNW Punch word in command format |
| PNWD | PWD Punch word in BCD format |
| PTWC | PTW Punch and type word in command format |
| PTWD | PTD Punch and type word in BCD format |
| PZE | Absolute operation code of +00 |
| MZE | Absolute operation code of -00 |

II. STANDARD MACROS

Macro-instructions are used much like standard commands and pseudo-operation codes. They usually generate several lines of coding. If the macro has several arguments, they are listed in successive address fields, where the command fields are PZE.

Most of the standard macros listed below generate references to RECOMP II Program No. 50, Program Preparation Package Two. Most of these macros are generated by SALT, but may be used independently.

(A,R) means the contents of the A and R registers. Angle arguments or function values for trigonometric functions are in radian measure, with automatic angle reduction.

| <u>MACRO</u> | <u>DEFINITION</u> |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SIN | SINE (A,R) \rightarrow (A,R) |
| COS | COSINE (A,R) \rightarrow (A,R) |
| TAN | TANGENT (A,R) \rightarrow (A,R) |
| ARCSIN | ARCSINE (A,R) \rightarrow (A,R) |
| ARCCOS | ARCCOSINE (A,R) \rightarrow (A,R) |
| ARCTAN | ARCTANGENT (A,R) \rightarrow (A,R) |
| ANGRED | (A,R) MOD $\pi \rightarrow$ (A,R) |
| LOG | LOG_{10} (A,R) \rightarrow (A,R) |
| LN | LOG_e (A,R) \rightarrow (A,R) |
| LOGTWO | LOG_2 (A,R) \rightarrow (A,R) |
| EXPTEN | $10^{(A,R)}$ \rightarrow (A,R) |
| EXP | $e^{(A,R)}$ \rightarrow (A,R) |
| EXPTWO | $2^{(A,R)}$ \rightarrow (A,R) |
| ABS | $ (A,R) \rightarrow$ (A,R) |
| ROUTINE | Used only thru SALT |
| RETURN | Used only thru SALT |
| READY | Inputs one decimal value from typewriter in any ordinary format. See SALT write-up or description of AN-007.1 for greater detail. Value is left in the (A,R) in normalized floating point format. T and A fields blank. |
| READZ | The photoreader attempts to read in a tape. The next SCRAP instruction must be PZE S name where "name" is the symbolic location at the start of the region to be filled from tape. The input tape must have on it before the first information, a code specifying the mode - N, C, L, or F, and |

after the last information to be entered, eight blank spaces, L22131 in Baudot code, carriage return code, and start code.

- CXP If Sense Switch B is on, the typewriter will type the absolute memory location of the CXP command, followed by the contents of the A and R registers in floating point format. T and A fields are blank.
- PRINT The contents of the A and R registers are printed as floating point values. Most values will be printed in fixed point format. Extremely large values will be in floating point format. T and A fields are blank.
- CRR Letter shift, carriage return is generated. T and A fields are blank.
- TAB Figure shift, tab is generated. T and A fields are blank.
- FLOAT A region of memory starting at the address of the FLOAT command is transformed from mixed number N mode to normalized floating point format. The last entry in the region is specified by the address of a PZE following the FLOAT command.

Each entry in the unfloated region must occupy two consecutive words; the integer part at binary 39 and the fraction at binary 0. No location field entry is proper for the FLOAT instruction. Example:

```

FLOAT    DATA
PZE      DATA+12

```

This will normalize 13 values.