# RECOMP II USER'S PROGRAM NO. 1108

| | |
|---|---|
| PROGRAM TITLE: | "APPLE" INPUT ROUTINE |
| PROGRAM CLASSIFICATION: | |
| AUTHOR: | W. Wellman<br>Servomechanisms, Inc.<br>Research Division<br>Goleta, California |
| PURPOSE: | The Apple routine serves to translate hand-written notes into punched tape that will utilize the Recomp II's functions. Simultaneously it delivers a typed version of the program actually entered. Among its more important functions are the conversion from mnemonic to octal codes, and binarization and entry of either floating or fixed point numbers. |
| DATE: | 23 October, 1961 |

Published by

RECOMP User's Library

at

AUTONETICS INDUSTRIAL PRODUCTS

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

# "APPLE" INPUT ROUTINE

The Apple routine serves to translate hand-written notes into punched tape that will utilize the Recomp II's functions. Simultaneously it delivers a typed version of the program actually entered. Among its more important functions are the conversion from mnemonic to octal codes, and binarization and entry of either floating or fixed point numbers.

Upon transferring into the Apple routine, lines such as the following are typed, partly by the computer, partly by the operator, as will be explained.

| | | | | |
|---|---|---|---|---|
| 0000 | Z | CLA | + 00 | 37721 |
| | Z | SUB | + 03 | 21150 |
| 0001 | Z | MPY | + 11 | 21130 |
| | Z | MPY | + 11 | 21170 |
| 0002 | Z | SIN | - 14 | 36651 |

There are five "fields" or columns, named the "location", "type", "mnemonic", "octal", and "address", from left to right. The first is the location of the entry in relative terms, i.e., starting at 0000. The second designates the type of information in the line. The third is the mnemonic symbol for the command. The fourth is the octal machine code for the command. The last is the action address of the command. The whole forms a typed sheet which can be used in tracing programs and debugging.

Upon transferring into Apple, the "0000"is typed in the first column by the routine. A tab follows, and the computer waits for the type designation. "Z", for example, means a "literal" or "protected" command ( i.e., one whose address is not relative ) and hence a command whose address must be entered unchanged. When the type designation has been entered, the computer will tab, and the mnemonic three-letter designation is typed in by the operator. This may be a standard Recomp command or a special one using the trapping mode and having a negative octal designation. ( A table of all such commands is carried internally by Apple.) The machine then tabs and prints the octal designation, complete with sign, and then tabs again, upon which the operator enters the correct five-digit address. The machine then carriage returns, and the cycle repeats. The relative location in the first column does not print the half word bit, but alternates between a printed location, and a blank as shown in the example above, for clarity. Timing is such that the computer does not tab into any field until it is ready to receive data in that field, so there need be no hesitation about making entry. ( The machine will stop after the second half-word address until the operator types a blank, for reasons to be explained later. )

There are four types of commands designated by "space", "Z", "K", and "A", placed in the second field by the operator. As has been explained, Z is used for protected commands. The "space" is exactly like the Z, except that the address typed in the fifth field is assumed to be relative to the start of the program. Apple punches a relocation matrix at the end of the entered subroutine; the use of either Z or space determines whether or not an entry is made in this matrix. "Space" means that the address of this command will change if the routine is relocated; "Z" means that the address stays the same regardless of relocation. The relocation matrix is of the same form as that produced by User's Program #1009 ( Halprin's FM-010 ). (See also RUG 1082)

Constants are entered by typing the designation "K" in the "type" field. The machine will tab to the third field. The operator then goes to the <u>console</u> ( not the typewriter ) and enters the "b" of the number as a sign and two digits followed by the "enter" key. Then enter ( still at console ) the sign and the constant, using the decimal point if needed, followed by the enter key. The routine will then type out the decimal number and the "b" at which it was entered, thus:

```
0005    K    +3.14159      AT +05
0006    K    - .00000001   AT +00
```

In addition, the number is displayed on the console as it is entered. Errors detected by the operator may be corrected any time prior to pushing the enter key; overflows ( $N > 2^b$ ) cause the console to display a row of decimal points, and wait for a new "b" and "N".

Since each number occupies a full word, there is no alternate spacing in the location. The K must not be used on the last half of a command. There is no "no-op" command; if a last half word is not needed, it should be filled with a dummy command.

If the number is to be entered in floating point, the action to be taken is exactly the same, except that the b is entered as "-00". The number is entered as before. The routine will type the number as a normalized decimal (i.e., value lying between 0.1 and 1.0) followed by an exponent of ten, thus:

| | | |
|---|---|---|
| 0005 | K | + .314159 + 1 |
| 0007 | K | + .365 + 3 |
| 0009 | K | - .1 - 7 |

(As with the command type of entry, the computer pauses after the datum has been printed until the operator pushes the blank key, as will be explained later.)

After floating point numbers, an extra carriage return makes for a double space, to emphasize the fact that the location counter has increased by two.

Alphanumeric material is entered by use of the letter "A" in the type field. Apple then types a carriage return. Headings or other alpha-numeric data can then be entered, including all figure shifts, letter shifts, tabs*, etc.. A blank signifies the exit from the alphanumeric mode (and hence cannot be stored alphanumerically). Following this exit, the machine will carriage return and type the next full word location after the alphanumeric fill, and proceed as before. Alpha characters are stored eight to the word, with the last word "filled out" with blanks.

When the full routine is entered, the operator signifies "end" by typing "S" in the "type" column. Apple will then put the relocation matrix immediately following the routine on the tape. This matrix is in the AFIT form, with 38 relocating marks stored per word. The complete tape, except for the matrix, is punched "as you go along" and is ready for use immediately after the matrix is punched. No part of the memory other than that used by Apple itself is required, since the whole tape is never stored.

Most errors made in the entry are inconsistencies that can be found by the machine, such as typing a "type"field entry other than space, Z, K, or A; or using a mnemonic code not entered in Apple's code table, or using a K or A type letter when only the first half of a command word has been filled and the last half is dangling, waiting for fill. Any detectable case such as this results in the machine's printing a carriage return, followed by the last word location before the error. The correct line may then be typed in.

The machine cannot detect consistent errors, such as numerical values entered erroneously by a mis-stroke of the typewriter or the console keys. Errors made when entering numbers on the console can be checked by reading the display, and corrected by using the "clear" key on the console in the usual manner. To prevent the frustration of seeing an error that cannot be corrected, the machine stops after the fifth ( last ) field of the each word and does not carriage-return until the entire word is checked. If the word is incorrect, typing an "X" signifies the fact, and the routine starts the line again. The erroneous entry is marked by the X, so that no confusion is caused by its presence on the paper. A correct entry is signified by the "blank" key, after which the routine carriage returns, and then the location of the next line is set up, proceeding as before.

For reasons of convenience, an error in alphanumeric entry cannot be corrected during use of Apple.

APPLE TABLE (W 23 OCTOBER, 1961)

| | |
|---|---|
| 340 | -00XXX |
| | -10XCR |
| | +43XAR |
| | +11MPY |
| | +13MPR |
| | -02BCD |
| | +00PZE |
| | +16PTW |
| 350 | +76PTC |
| | +14PNW |
| | +74PNC |
| | +77HTR |
| | -07HDG |
| | -03LOG |
| | +51TMI |
| | +53TOV |
| 360 | +52TPL |
| | +12TYW |
| | +72TYC |
| | +50TZE |
| | +57TRA |
| | +54TSB |
| | +55TSC |
| | +56TSD |
| 370 | +02CLS |
| | +00CLA |
| | +66CTV |
| | +64CTL |
| | +67CFV |
| | +65CFL |
| | +07FMP |
| | +34FCS |
| 400 | +30FCA |
| | +45FNM |
| | +05FDV |
| | +06FSB |
| | +44FSQ |
| | +35FST |
| | +04FAD |
| | +71RDY |
| 410 | +73RDZ |
| | +23DVR |
| | +22DIV |
| | +36DIS |
| | +20DSL |
| | +21DSR |
| | -01IND |
| | +25SQR |
| 420 | +60STO |
| | +42STA |
| | +03SUB |
| | -05SIN |
| | +15SAX |
| | +41ALS |
| | -06ATN |
| | +40ARS |
| 430 | +01ADD |
| | -11ASN |
| | -04EXP |
| 433 | +33EXT (BLANKS) |

TABLE IS 'F' FILLED IN ORDER OF INCREASING NUMERICAL VALUE OF THE 15 'MNE' BITS, THE LATTER CONSIDERED TO BE A NUMBER AT A BINAL POINT OF 39.

EACH WORD IN THE TABLE IS OF THE FOLLOWING FORM;

'FIG ± X X LTR M N E'

NEGATIVE (TRAPPING) INSTRUCTIONS:

| | |
|---|---|
| -00 | XXX |
| -01 | IND |
| -02 | BCD |
| -03 | LOG |
| -04 | EXP |
| -05 | SIN |
| -06 | ATN |
| -07 | HDG |
| -10 | XCR |
| -11 | ASN |