A U T O N E T I C S

A DIVISION OF NORTH AMERICAN AVIATION, INC.

INDUSTRIAL PRODUCTS

3584 Wilshire Blvd., Los Angeles 5, Calif.

June 3, 1960

# RECOMP II TECHNICAL BULLETIN NO. 11

TITLE: Logic, wiring, and physical changes to RECOMP II

PURPOSE: To provide detailed information concerning engineering changes to RECOMP II and the effect of those changes on programming and operating procedures for the computer. Descriptions of the revised logic have not been included, since that information is not of general interest.

EFFECTIVE DATE: June 3, 1960

CONTENTS:

## I. INTRODUCTION

A total of six changes to the computer will be instituted:

1. Full alphanumeric output.
2. Absolute value commands.
3. Echo checks for the tape punch and typewriter.
4. X-register and C-register readout on verify error.
5. Alphanumeric input-output indicator light.
6. Miscellaneous logic and wiring changes.

## 1. FULL WORD ALPHANUMERIC OUTPUT

This feature will allow one to eight 5-bit alphanumeric characters to be typed, punched, or punched and typed by a single command from the accumulator. Operation codes 72, 74, and 76 when used with a loop address will accomplish the output. The number of characters to be output will be specified by the least significant octal number of the address, excluding the spare bit, which must be zero.

EXAMPLE 1:

If the address field of one of the above commands is 77740 or 77640, four alphanumeric characters will be output corresponding to the twenty most significant bits of the accumulator, including sign, As each character is output, the 5-bits corresponding to the character are precessed through the sign bit and into the F register. The 5 bits contained in the F register prior to the precession will be recirculated into the least significant 5 bits of the A register, and the remaining 35 bits in the accumularor will be shifted five places to the left. Bits corresponding to the final character output by these commands are not recirculated.

EXAMPLE 2:

If the bit configuration of the accumulator before output by a command +72 77630 is

+11111111111111101010101010101010101010101

the configuration after the execution of the command will be

+01010101010101010101010100000111111111111.

The Preset Stop and Single Command switches will be inoperative at the location which stores the 71, 72, 73, 74 or 76 commands when these commands are used with a loop address (7760-7777).

Descriptions of the Type Alpha, Punch Alpha, and Punch and Type Alpha commands are given in II, below.

2. ABSOLUTE VALUE COMMANDS

When the address of an operand contains a "1" bit in the least significant position, the absolute value of the operand will be used whenever the following operations are executed:

| | | | |
|---|---|---|---|
| 00 | CAM | W.1 | Clear and Add Magnitude |
| 01 | ADM | W.1 | Add Magnitude |
| 02 | CSM | W.1 | Clear and Subtract |
| 03 | SBM | W.1 | Subtract Magnitude |
| 04 | FAM | W.1 | Floating Add Magnitude |
| 05 | FDM | W.1 | Floating Divide by Magnitude |
| 06 | FSM | W.1 | Floating Subtract Magnitude |
| 07 | FMM | W.1 | Floating Multiply by Magnitude |
| 20 | DSM | W.1 | Divide by Magnitude Single Length |
| 21 | SRM | W.1 | Divide by Magnitude Single Length and Round |
| 22 | DVM | W.1 | Divide by Magnitude |

| | | | |
|---|---|---|---|
| 23 | DRM | W.1 | Divide by Magnitude and Round |
| 25 | SQM | W.1 | Square Root of Magnitude |

Descriptions of these commands are given in II, below.

3. ECHO CHECK FOR TAPE PUNCH AND TYPEWRITER OUTPUT

The logical changes involved are to give maximum reliability in punched and/or typed output. When an output error is detected, the output error indicator will be lighted. If a punch or type word command is being executed and an error is detected, the output of the word will be completed, and overflow will occur. If a punch or type character command is being executed and an error is detected, the character in error will be output and overflow will occur. If a TOV (transfer on overflow) command is inserted immediately following a punch or type command, the indicator will be turned off and control will be transferred to the location specified by the TOV command. If a TOV command does not immediately follow a punch or type command, the computer will halt after the word or character has been output.

It is apparent that with this modification and modification No. 1 (Full Word Alphanumeric Output) a definitive memory dump routine can be written which will practically eliminate the possibility of a punched tape being in error.

4. X-REGISTER AND C-REGISTER READOUT FOLLOWING A VERIFY ERROR

Logical changes will permit readout after a verify error of the X and C registers without the necessity of first depressing the error reset button. At present, if a verify error occurs and a readout button is depressed prior to the error reset button, the tape and photoreader are advanced and the next command pair is gated into the C register. This comparison of information in the X and C registers without error reset does not yield pertinent information concerning the word which failed to verify.

5. ALPHANUMERIC INPUT-OUTPUT INDICATOR LIGHT

Removal of present interlock logic will allow an indicator to be used as an alpha-mode indicator. The light will be on during the execution of all alphanumeric input-output commands. A decal will be provided to cover the present etching on the control pannel.

6. MISCELLANEOUS LOGIC AND WIRING CHANGES

Other logical and wiring changes will also be made, but will not affect programming or operation of the computer and will not be described herein.

## II. COMMAND LIST OF ADDITIONAL INSTRUCTIONS

### Clear and Add Magnitude

00 CAM W.1

The C(A) are replaced by the $|C(W)|$. The C(W) remain unchanged. The R register is not affected by this command.

### Add Magnitude

01 ADM W.1

The $|C(W)|$ are algebraically added to the C(A). The sum replaces the original C(A). If the sum is zero, the sign of the original C(A) will remain. Should the capacity of the A register be exceeded during the summation, the overflow indicator will flash and the computer will stop unless the next command is a transfer on overflow (TOV). The C(W) and C(R) remain unchanged.

### Clear and Subtract Magnitude

02 CSM W.1

The C(A) are replaced by the negative of $|C(W)|$. The C(W) are not changed; neither are the C(R).

### Subtract Magnitude

03 SBM W.1

This instruction algebraically subtracts the $|C(W)|$ from the C(A), The difference replacing the original C(A). If the difference is zero, the sign of the original C(A) remains. The C(W) and the C(R) remain unchanged by this instruction. Overflow is possible as a result of this operation. If it occurs, the overflow indicator will flash and the computer will stop unless the next command is a TOV.

### Floating Add Magnitude

04 FAM W.1

The $|C(W)|$ and C(A) are properly and automatically positioned along with adjustment of their respective exponents before addition occurs. After addition, the normalized sum replaces the C(A), and the exponent corresponding to this sum replaces the C(R). The C(W) and C(W + 1) are unaltered.

The details of how this is accomplished are as follows:

1. The |C(W)| and C(W + 1) replace the contents of the two auxiliary registers, B and X, respectively.
2. The C(A) and C(B) are shifted to the right one bit position and the C(R) and C(X) are increased algebraically by a 1 in the least significant bit position.
3. If the magnitude of the C(A) or the C(B) is zero after the shift, the computer proceeds to step 4. If neither the magnitude of the C(A) nor the C(B) is zero, the C(A) or C(B) are shifted to the right one binary position. A 1 is added to the least significant position of the corresponding exponent part in the R or X register until the C(R) and C(X) are made equal or until the C(A) or C(B) are made zero. The C(A) are shifted when the C(R) are less than the C(X); the C(B) are shifted when the C(R) are greater than the C(X).
4. The C(A) and the C(B) are, then, added algebraically and the sum replaces the C(A). Note that overflow cannot occur from this step since the original C(A) and C(B) were shifted one bit position to the right in step 2.
5. The floating point number, with fraction part in the A register and exponent part in the R register, is then normalized by shifting the C(A) (exclusive of the sign bit) to the left and subtracting a 1 from the least significant bit position of the C(R) for each position shifted.

Floating Divide by Magnitude

05 FDM W.1

The floating point number contained in the A and R registers is divided by the absolute value of the floating point number in W and W + 1. This forms a normalized floating point quotient; the fractional part replaces the C(A) and the exponent part replaces the C(R). The C(W) and C(W + 1) are unaltered.

The overflow trigger will be turned on if the fractional part of the divisor contains all zeros in bit positions 1 through 39, or if the divisor is not in normalized form and its fractional part is less than or equal to the fractional part of the dividend after the right shift described in step 2.

The details of how floating point division is accomplished follow:

1. The |C(W)| and C(W + 1) replace the contents of the two auxiliary registers, B and X, respectively.
2. The C(A) are shifted one bit position to the right and a binary 1 is added algebraically to the C(R) in the least significant bit position.
3. The C(X) are subtracted algebraically from the C(R) and the difference replaces the C(X). Exponent overflow may occur at this time but is not indicated in any manner.
4. The C(A) are divided by the C(B) and the quotient thus formed is rounded to the 39 most significant bits. These 39 bits replace C(A) unless the overflow trigger is turned on under the conditions described above. During the last word time of this step the C(X) replace the C(R).

5. The floating point quotient, with the fractional part in the A register and the exponent part in the R register, is then normalized by shifting the C(A) to the left and subtracting algebraically a 1 from the least significant bit position of the C(R) for each position shifted. Finally, the $C(R)_{1,2}$ are replaced with zeros.

Floating Subtract Magnitude

06 FSM W.1

The |C(W)| and C(A) are properly and automatically positioned along with adjustment of their respected exponents before subtraction occurs. After subtraction, the normalized difference replaces the C(A) and the exponent corresponding to this difference replaces the C(R). The C(W) and C(W + 1) are unaltered.

Floating subtraction takes place in a manner similar to the FAD instruction. The details are as follows:

1. The |C(W)| and C(W + 1) replace the contents of the two auxiliary registers, B and X, respectively.
2. Steps 2 to 5 of the FAD instruction are performed, except that a subtraction (instead of addition) of the fractional parts occurs in step 4.

Floating Multiply by Magnitude

07 FNM W.1

The floating point number contained in the A and R registers is multiplied by the absolute value of the floating point number contained in W and W + 1. This forms a normalized floating point product, the fractional part of which replaces the C(A) and the exponent part of which replaces the C(R). The C(W) and C(W + 1) are unaltered.

The details of how floating multiplication is accomplished are as follows:

1. The |C(W)| and C(W + 1) replace the contents of the two auxiliary registers, B and X, respectively.
2. The C(A) are shifted one bit position to the right and a 1 is added algebraically to the C(R) in the least significant bit position.
3. The C(X) are added algebraically to the C(R) and the sum replaces the C(X). Exponent overflow may occur at this time but is not indicated in any manner.
4. The C(A) are multiplied by the C(B) and the product thus formed (rounded to the most significant 38 bits) replaces the C(A) and C(R).
5. The C(X) replaces the C(R).
6. The floating point number, with the fractional part in the A register and the exponent part in the R register, is then normalized by shifting the C(A) to the left and subtracting

algebraically a 1 from the least significant bit position of the C(R) for each position shifted.

Divide By Magnitude Single Length

20 DSM W.1

This instruction operates in a manner similar to the DVM instruction below. In this case only the 39 bits plus sign in the A register will be divided by the $|C(W)|$. Any information in R prior to execution will not be a factor in the division. The quotient and sign replace the initial contents of A, while the contents of R are replaced by the remainder. The sign of the remainder will be the same as the original dividend.

If the magnitude of the dividend is equal to or greater than the magnitude of the divisor, an overflow will occur.

Divide by Magnitude Single Length and Round

21 SRM W.1

Operation will proceed in a manner similar to DRM operation, except that only the 39 bits plus sign of the dividend which is in A will be divided by the $|C(W)|$. The rounded quotient will replace the initial C(A). The contents of R will be replaced by the rounded quotient with the same sign as the original dividend.

The C(W) are unchanged and an overflow will occur if the magnitude of the dividend is equal to or greater than the magnitude of the divisor.

Divide by Magnitude

22 DVM W.1

The C(A) and C(R) are divided by the $|C(W)|$. In this instruction the contents of the A and R registers are used as a continuous 78 bit dividend with the 39 most significant bits in the A register and 39 least significant bits in the R register. The sign associated with the 78 bit dividend is in the A register. The sign of the R register prior to execution of the instruction has no effect on the results.

After execution of the divide instruction, the 39 bit quotient plus its sign supplants the C(A). The remainder replaces the C(R) and the sign of the remainder is the same as that of the dividend. The C(W) are unchanged.

It is possible for an overflow condition to occur as a result of this operation. It occurs if a division is attempted when the magnitude of the dividend is greater than or equal to the magnitude of the divisor which is in W.

Divide by Magnitude and Round

23 DRM W.1

This instruction is executed in a manner similar to the DVM instruction. The 39 most significant bits and the sign of the dividend are placed in A. The 39 least significant bits of the dividend are placed in R.

After execution of the DRM operation, the 78 bit dividend in A and R is replaced by a rounded 39 bit quotient and its sign in A. R contains the same digit configuration as the rounded 39 bit quotient in A; however, the sign of the R register is the same as the original 78 bit dividend.

If the magnitude of the dividend is greater than or equal to the magnitude of the divisor, an overflow will occur.

In rounding, the magnitude of the remainder in R is compared with the magnitude of the divisor. If the C(R) are greater, the quotient is increased by a 1 in position 39. If this causes the A register to overflow, then the overflow trigger will be turned on. This rounding process replaces the remainder in R with the rounded quotient.

Square Root of Magnitude

25 SQM W.1

This instruction extracts the square root of the absolute value of the quantity found at W. The square root will replace the contents of the A register, with a binary scale of one half that of the number in W. The instruction assumes the binary scale of the number in W to be an even number. If it is not an even number the result will be incorrect. That is, the number may be off by a factor of 2 before the extraction; consequently, the result will be off by $\sqrt{2}$.

In the execution of the instruction the C(R) are destroyed and the sign of the A register is made positive. The C(W) remain unchanged.

Type Alpha

72 TYA M.0

If M is less than 7760.0, a single character is typed corresponding to the Baudot code in bit positions 14 through 18 in a left half instruction or 34 through 38 in a right half insrruction. This is identical to the TYPE CHARACTER instruction (TYC).

If M is greater than or equal to 7760 and the least significant bit is a zero, one to eight characters will be typed. The exact number of characters typed will be determined by the least significant octal number of the address specified by the following convention:

7760.0 or 7770.0 = 8 characters
7761.0 or 7771.0 = 1 character
7762.0 or 7772.0 = 2 characters
7763.0 or 7773.0 = 3 characters
7764.0 or 7774.0 = 4 characters
7765.0 or 7775.0 = 5 characters
7766.0 or 7776.0 = 6 characters
7767.0 or 7777.0 = 7 characters

Each character typed will correspond to the most significant 5-bits of the accumulator, including the sign. As each character is typed, these 5 bits will be precessed through the sign position into the F register. The 5 bits contained in the F register prior to the precession will be recirculated into the least significant 5 bits of the A register, and the remaining 35 bits in the accumulator will be shifted five places to the left. Bits corresponding to the final character typed by the command are not recirculated.

If a discrepancy occurs, the output error light on the console will be lighted and the next instruction will be checked. If that instruction is TOV (transfer on overflow) control will be transferred to the location specified by the TOV command; otherwise, computation will cease.

If M is greater than or equal to 7760.0, the least significant bit of the address portion of the instruction myst be a zero. A "1" bit in that position will cause erroneous typeout.

Punch Alpha

74 PNA M.0

If M is less than 7760.0, a single character is punched corresponding to the Baudot code in bit positions 14 through 18 in a left half instruction or 34 through 38 in a right half instruction. This is identical to the PUNCH CHARACTER instruction (PNC).

If M is greater than or equal to 7760 and the least significant bit is a zero, one to eight characters will be punched. The exact number of characters punched will be determined by the least significant octal number of the address specified by the following convention:

7760.0 or 7770.0 = 8 characters
7761.0 or 7771.0 = 1 character
7762.0 or 7772.0 = 2 characters
7763.0 or 7773.0 = 3 characters
7764.0 or 7774.0 = 4 characters
7765.0 or 7775.0 = 5 characters
7766.0 or 7776.0 = 6 characters
7767.0 or 7777.0 = 7 characters

Each character punched will correspond to the most significant 5-bits of the accumulator, including the sign. As each character is punched, these 5 bits will be precessed through the sign position into the F register. The 5 bits contained in the F register prior to the precession will be recirculated into the least significant 5 bits of the A register, and the remaining 35 bits in the accumulator will be shifted five places to the left. Bits corresponding to the final character punched by the command are not recirculated.

If a discrepancy occurs, the output error light on the console will be lighted and the next instruction will be checked. If that instruction is TOV (transfer on overflow) control will be transferred to the location specified by the TOV command; otherwise, computation will cease.

If M is greater than or equal to 7760.0, the least significant bit of the address portion of the instruction must be a zero. A "1" bit in that position will cause erroneous punching to occur.

Punch and Type Alpha

76 PTA M.0

If M is less than 7760.0, a single character is punched and typed corresponding to the Baudot code in bit positions 14 through 18 in a left half instruction or 34 through 38 in a right half instruction. This is identical to the PUNCH AND TYPE CHARACTER instruction (PTC).

If M is greater than or equal to 7760 and the least significant bit is a zero, one to eight characters will be punched and typed. The exact number of characters output will be determined by the least significant octal number of the address specified by the following convention:

7760.0 or 7770.0 = 8 characters
7761.0 or 7771.0 = 1 character
7762.0 or 7772.0 = 2 characters
7763.0 or 7773.0 = 3 characters
7764.0 or 7774.0 = 4 characters
7765.0 or 7775.0 = 5 characters
7766.0 or 7776.0 = 6 characters
7767.0 or 7777.0 = 7 characters

Each character punched and typed will correspond to the most significant 5-bits of the accumulator, including the sign. As each character is output, these 5 bits will be precessed through the sign position into the F register. The 5 bits contained in the F register prior to the precession will be recirculated into the least significant 5 bits of the A register, and the remaining 35 bits in the accumulator will be shifted five places to the left. Bits corresponding to the final character output by the command are not recirculated.

If discrepancy occurs, the output error light on the console will be lighted and the next instruction will be checked. If that instruction is TOV (transfer on overflow) control will be transferred to the location specified by the TOV command; otherwise, computation will cease.

If M is greater than or equal to 7760.0, the least significant bit of the address portion of the instruction must be a zero. A "1" bit in that position will cause erroneous output.

REFERENCES

1. RECOMP II Operating Manual
2. Resume of Logic and Physical Changes proposed by Industrial Products Manufacturing and Mr. T.M. Hertz of Engineering Analysis.

INFORMATION TO: All concerned

WRITTEN BY: H. L. Judd
Applied Mathematics
Autonetics Industrial Products