IBM

TRADE MARK

TECHNICAL NEWSLETTER

No. 3

APPLIED SCIENCE DEPARTMENT

IBM

APPLIED SCIENCE DEPARTMENT

Technical Newsletter No. 3

December, 1951

Technical Newsletter Number 3 contains a set of papers which will prove valuable both in technical computing and commercial computing. The application of automatic computing equipment to matrix arithmetic is important in the fields of chemistry and petroleum and the methods described in Part 1 will be equally helpful in other fields. Part 2 contains a description of a number of interesting and useful ideas which increase the power and applicability of the IBM Type 602-A Calculating Punch and the IBM Type 604 Electronic Calculating Punch. Part 3 contains an excellent paper on the subject of interpolation of the IBM Card-Programmed Electronic Calculator.

The cooperation of the authors of these papers in contributing to the interchange of technical information on computing is greatly appreciated.

CONTENTS

PART 1

Linear Equations and Matrix Operations

PART 2

IBM Type 604 Electronic Calculating Punch and Type 602-A Calculating Punch

PART 3

Application of the IBM Card-Programmed Electronic Calculator

MASS SPECTROMETER CALCULATIONS
on the
IBM 602-A CALCULATING PUNCH

W. H. King, Jr. and William Priestley, Jr.

Esso Laboratories, Standard Oil Development Company

INTRODUCTION

The following instructions describe in detail the control panels for the IBM Type 602-A Calculating Punch which are used to perform a matrix multiplication and subsequent normalization steps. The problem,which has been discussed in detail in another paper[a], is discussed here in brief for reference purposes.

THE PROBLEM

1. To multiply a 20th order matrix by a vector.
2. To calculate from the answers obtained in (1) above the following percentages:
   a. Total normalized mole per cent.
   b. Air free mole per cent.
   c. Fixed gas free mole per cent.
   d. Fixed gas free weight per cent.
   e. Air free specific gravity.
   f. Fixed gas free specific gravity.

The inverted matrix shown in Figure 1 will be multiplied by the peak heights (b_1, b_2, b_3, etc.) to obtain 19 unnormalized mole per cent answers (X_1, X_2, X_3, etc.) and a summation check. Since a summation check is calculated along with the 19 answers one has effectively a 20 × 20 matrix multiplication problem. To obtain any given X value, for example X_1, one multiplies the peak heights b_1, b_2, b_3, etc. by their corresponding matrix values $B_{1,1}$; $B_{1,2}$; $B_{1,3}$; etc. A summation of these products yields X_1. Similarly X_{19} is obtained by multiplying the peak heights by their corresponding matrix values for the 19th row $B_{19,1}$; $B_{19,2}$; $B_{19,3}$; etc. In the twentieth row the B_n values are the sum of the columns, i.e., $B_{n,5}$ is the sum of all B's in column 5 ($B_{1,5} + B_{2,5} + B_{3,5}$ + etc.). The twentieth row is therefore the summation check and the value obtained for ΣX_n should equal the sum of $X_1 + X_2 + X_3 + X_4 + \ldots + X_{19}$ if the matrix has been multiplied correctly. Since ΣX_n is the sum of 19 multiplications and the value from totaling $X_1 + X_2 + X_3$ etc. is the sum of 361 multiplications, a comparison of the two figures is an adequate check on the machine's computation.

[a] See authors' article, "Spectrometic Analysis Employing Punch Card Calculators", in Analytical Chemistry, Vol. 23, No. 10 (October, 1951), pp. 1418-1420.

It was decided to make the matrix multiplication in blocks of 8, as indicated in Figure 1, in order to take advantage of the time saved by making two simultaneous multiplications. This calculation method also utilizes the standard IBM 602-A Calculating Punch at maximum capacity. (Additional capacity is available at increased cost.) Two rows of the matrix are calculated simultaneously, resulting in two answers for each five matrix cards. The matrix has been so arranged that the 17, 18, and 19th answers are fixed gases N_2-CO, CO_2 and air.

The normalization procedure consists of determining the normalization factors and making the division steps in order to provide the required answers.

These factors and steps are symbolized below:

(Total sample mole % normalization factor)

1. $X_1 + X_2 + X_3 + \ldots + X_{19} = \sum_{19}^{1} M\%$

(Normalization factor for the mole % air free)

2. $X_1 + X_2 + X_3 + \ldots + X_{18} = \sum_{18}^{1} M\%$

(Normalization factor for the mole % fixed gas free sample)

3. $X_1 + X_2 + X_3 + \ldots + X_{16} = \sum_{16}^{1} M\%$

(Normalization factor for the weight % of fixed gas free samples and unnormalized specific gravity for this sample)

4. $(X_1)(M.W.F._1) + (X_2)(M.W.F._2) + \ldots + (X_{16})(M.W.F._{16}) = \sum_{16}^{1} Pr$

(Unnormalized specific gravity for the air free sample)

5. $(X_1)(M.W.F._1) + (X_2)(M.W.F._2) + \ldots + (X_{18})(M.W.F._{18}) = \sum_{18}^{1} Pr$

Molecular Weight Factor:

6. $(M.W.F.) = \dfrac{\text{molecular weight of the component}}{\text{molecular weight of air}}$

Total mole per cent answer:

7. $M\%_t = \dfrac{100\,X}{\sum_{19}^{1} M\%} = \dfrac{\text{unnormalized mole \%}}{\text{normalization factor}} \times 100$

Air free mole per cent answer:

8. $M\%_{af} = \dfrac{100X}{\sum_{19}^{1} M\%}$

Fixed gas free mole per cent answer:

9. $M\%_{fgf} = \dfrac{100X}{\sum_{16}^{1} M\%}$

Fixed gas free weight per cent answer:

10. $$\text{Wt. }\%_{fgf} = \frac{(X)\ (M.W.F.)\ (100)}{\sum_{16}^{1} Pr}$$

Air free specific gravity:

11. $$S.G._{af} = \frac{(\sum_{18}^{1} Pr)\ (100)}{\sum_{18}^{1} M\ \%}$$

Fixed gas free specific gravity:

12. $$S.G._{fgf} = \frac{(\sum_{16}^{1} Pr)\ (100)}{\sum_{16}^{1} M\%}$$

For those not familiar with the actual equipment, a brief explanation of the calculating punch will be of interest. The 602-A has three basic operations; reading information from a card, performing calculations, and punching results. Information read from a card may be entered into several storage units designated as 1L, 1R, 2L, 2R, 3L, 3R, 4L, 6L, 6R, 7L, 7R. A number existing in a storage unit will remain there until another number is read into that storage unit. Numbers may be read out of any storage unit many times for calculation purposes without destroying the number. Storage unit 1R is unique since divisors or multipliers must be entered here (from other storage units or from reading of the card) just prior to the division or multiplication. The standard 602-A also has 6 counter units where products, quotients, or sums may be developed and accumulated. There, counters may also be used for storage.

MATRIX CALCULATION CONTROL PANEL #1

(Figures 2 and 3)

The problem at hand requires the storage of 12 numbers, 8 matrix values and 4 multipliers. The planning chart F 182 has been filled out for the first block of 8 multiplications. The 602-A will perform two simultaneous multiplications by a common multiplier resulting in two ten digit products. It is possible to perform more simultaneous multiplications if the number of required digits is less. It is advantageous to do this in order to save time. In this control panel setup, the sum of products of the odd rows will be accumulated in counters (2, 3) where they are developed. Simultaneously, counters (5, 6) will develop and accumulate the products of the even rows. Thus, when the 20th matrix value has been calculated for the first two rows, counters (2, 3) will contain

$$b_1\ (B_{1,1})\ + b_2\ (B_{1,2})\ + b_3\ (B_{1,3})\ +\ldots+ b_{19}(B_{1,19})$$

and counters (5, 6) will contain

$$b_1 (B_{2,1}) + b_2(B_{2,2}) + b_3(B_{2,3}) +\ldots+ b_{19}(B_{2,19}).$$

These two sums will be the first two answers and will be punched on trailer cards following the five detail or matrix cards.

Read Cycle

On the read cycle, the multipliers b_1, b_2, b_3, and b_4 are read from the matrix card into storage units 1R, 2L, 2R, 3L, while the multiplicands $B_{1,1}$, $B_{1,2}$, $B_{1,3}$, etc. are read into storage units 3R, 4L, 4R, 6L, 6R, 7L, 7R and counter (1). The numbers under the matrix symbol in Figure 2 indicate their column position in the card. For example $B_{2,1}$ is a 4 digit number in columns 62, 63, 64, 65 while column 61 is a 0 or 5 to indicate the sign of the 4 digit number. 50064 means - 64, while 00064 means + 64. After the read cycle, the calculator proceeds to the program steps.

Program Step 1 (For Sign Control)

Since the first multiplication will be $(b_1)(B_{1,1})$ and $(b_1)(B_{2,1})$, the multiplier was entered in 1R on the read cycle. The sign control for the multiplication is usually accomplished on the same program step as the multiplication by using a 0 or 9 in a specified digit position of the multiplicand. This would require, for 8 multiplications, the use of 8 pilot selectors (sign controllers). The standard 602-A has 7 pilot selectors and consequently a different system must be employed. By using an extra program step per multiplication, two pilot selectors may be used four times to accomplish sign control.

In this program a 0 or 5 is read out (R. O.) of the first digit position of storage unit 3R, and the board will be wired to permit a 5 to energize pilot selector number 1 to indicate that the sign of $B_{1,1}$ is negative. If a 0 exists in this digit position, pilot selector 1 will not be energized (indicating $B_{1,1}$ is positive) and the succeeding multiplication will proceed positively. A similar arrangement is used for the sign control of $B_{2,1}$ via pilot selector number 2. These two pilot selectors will be used over and over as will become apparent from the succeeding discussion. Pilot selector number 1 (P. S. #1) will always control the sign of all odd row matrix values, and (P. S. #2) will control the sign of all even row matrix values. No sign control is required for the vector or multiplier since in this problem the value is always positive. These pilot selectors must be reset to clear out the previous sign control. This is accomplished by dropping out (resetting) the pilot selectors with the digit selector before a 5 from the storage unit can energize it. In the operation of the machine, digits are read in a time sequence, 9, 8, 7, 6, 5, 4, 3, 2, 1. (No impulses are emitted for a zero.) When the digit selector is impulsed, it immediately emits a series of impulses in this order. The first impulse (9) is wired to drop out the selectors. This occurs before the impulse for (5) from the storage unit can arrive at the pilot selector for sign control.

Program Step 2

Multiplication is accomplished by impulsing the multiplier. In this operation the value in storage unit 3R is read into counter (2, 3) through the sign control pilot selector for as many times as there are digits in the multiplier 1R. For example, if (1R) is 123 and (3R) is 00366, the value 366 will enter counter (2, 3) 123 times. In reality 36600 will enter once, 3660 twice, and 366 three times. The accumulation therefore will be 40626. If 3R were 50666, indicating minus 366, 36600 would be deducted from counter (2, 3) once, 3660 deducted twice, and 366 three times. This would leave 9999959374 in the counter which is the complement of 40626; the 9 in the extreme left digit position serves to indicate it as a complement, not a positive value.[b]

Program Step 3

This step is similar to program step 1 (P_1) except that the multiplier is changed to b_2 by reading b_2 out of storage unit 2L into the multiplier storage unit 1R. The sign control of $B_{1,2}$ and $B_{2,2}$ is accomplished by reading the fifth digit positions of (4L) and (7L) to their respective pilot selectors.

Program Step 4

Multiplication here is again similar to P_2. The counters will now contain the sum of the products of the first two sets of multiplications. Program steps 5, 6, 7 and 8 are self-explanatory since they are a repetition of the early programs which have been explained in detail. At this point, the data furnished on detail or matrix card # 1 have been calculated.

Program Step 9

Since counter (1) was used as a storage unit it must be cleared before another number is entered, hence, counter (1) is reset on this program. The sign control pilot selector is also reset by impulsing the digit selector on this program. The next operation of the machine will be reading a new matrix card, consequently, the read hub is impulsed. Programs # 1 through # 9 will be repeated for each of 5 matrix cards.

After the 5th detail or matrix cards have been computed, counters (2, 3) will contain the odd row answer while counters (5, 6) will contain the even row answer.

Program Step 10

When a trailer or answer card enters the machine with an "X" punch in column 14, all preceeding programs are skipped and the programming will start here.

The function of X14 cards will be to punch the answer of odd rows in counters (2, 3) in this card. Counter (2, 3) is read into punching storage unit 6 and the answer will be punched from this storage unit in the card.

The next step is to read a new card; consequently read is impulsed.

Prcgram Step 11

When a card enters the machine with an "X" punch in column 15 all programs preceeding program 11 will be skipped and programming will start here. The function of

[b] This is in reality a simplification of how the IBM Type 604 Electronic Calculating Punch multiplies. The actual multiplication procedure for the Type 602-A is much more involved.

X15 cards will be to punch the answer for even rows in this card.

Summary of Operation Thus Far

Five detail (or matrix) cards have entered the machine and have been calculated. The results of the calculations have been punched on two trailer cards. The answer to the first row multiplications is in the X 14 trailer card and that of the second row in the X 15 trailer card.

This process is iterated for row 3 and 4 etc. until the entire matrix has been computed. After the matrix has been computed there will be 70 cards to sort. Fifty of these are the matrix or detail cards, while the remaining 20 are the X14, X15 trailer cards containing the 19 unnormalized mole per cent answers and the check row answer. The next control panel will process these 19 answer cards and produce three more cards containing the normalization factors.

The system used here is not limited to a 20 × 20 matrix. Any order matrix may be used by extension of the cards entering the machine prior to the trailer cards and no control panel changes are required.

Figure 3 shows the wiring diagram for the matrix calculation. This drawing is the actual wiring involved in the control panel.

MOLECULAR WEIGHT FACTOR MULTIPLICATION
AND SUMMATION CONTROL PANEL # 2

Figures 4 and 5 show the planning chart for the molecular weight factor multiplication and the summation of certain answers and products.

Answer cards (X14 and X15) were punched on control panel#1 when the matrix was calculated. In these cards the corresponding molecular weight factor must be punched by a gang punch or hand punch prior to this step.

The function of this control panel is:

(a) To multiply the 19 unnormalized mole % answers by the molecular weight factor (M. W. F.)

(b) To determine the normalization factors:

1. ($\sum_{19}^{1}$ M%) the total sample mole % normalization factor which is the sum of the 19 unnormalized mole per cent answers.

2. ($\sum_{18}^{1}$ M%) the air free sample mole % normalization factor which is the sum of 18 unnormalized mole % answers (omitting the air card).

3. ($\sum_{16}^{1}$ Pr) the fixed gas free sample mole % answer which is the sum of 16 unnormalized mole % answers (omitting air, nitrogen-carbon monoxide, and carbon dioxide).

4. ($\sum_{16}^{1}$ Pr) the fixed gas free weight % normalization factor which is the sum of the products of the mole % times the molecular weight factor for 16

cards. This factor when divided by $\sum_{16}^{1}$ M% yields the specific gravity for the fixed gas free sample.

5. ($\sum_{18}^{1}$ Pr) the sum of the mole % times the molecular weight factor for 18 cards. This factor when divided by $\sum_{18}^{1}$ M % yields the specific gravity for the air free sample.

(c) To punch these results in the appropriate cards.

Read Cycle

The molecular weight factor (M. W. F.) in card columns 25 through 29 is read into the multiplier storage unit 1R; the unnormalized mole % answer in columns 17 through 21 is read to both storage unit 2R and counter (1). This will accumulate the sum of the unnormalized mole % answers.

Program Step 1

The unnormalized mole % answer (M %) is multiplied by (M. W. F.) by reading 2R into counters (5, 6), impulsing multiply, and reading out 1R.

Program Step 2

Because the products obtained in program step 1 are to be totaled, this program is used to transfer the product to counters (2, 3). At the same time the product is read into punching storage.

Program Step 3

This program punches the product in this card and sets the machine to read the next card. Programs 1 through 3 are repeated for each of the 19 unnormalized mole % cards.

Program Step 4

When an answer card containing an "X" in column 22 enters the machine, programs 1 and 2 are performed but program 3 is skipped.

At this time 16 cards have been computed and therefore counter (1) contains the sum of the uncorrected mole % answers ($\sum_{16}^{1}$ M %) thus far. Likewise, counters (5, 6) contain the sum of the products ($\sum_{16}^{1}$ Pr).

$\sum_{16}^{1}$ M % is stored in 3L and $\sum_{16}^{1}$ Pr is stored in 3R at this time. These results will be punched later on trailer cards.

Program Step 5

P_5 is identical to P_3 and the product is punched in this card (X 22) and "read" is impulsed to proceed with the next card.

Program Steps 6, 7, 8

These steps are similar to the above except that $\sum_{18}^{1}$ Pr are entered in 4L and 4R while $\sum_{19}^{1}$ M % and $\sum_{19}^{1}$ Pr are left in the counters. At this time 19 answer cards have been processed and the normalization factors have been placed in storage. The next few program steps are used to punch the normalization factors on three trailer cards.

Program Step 9

When a trailer card with an "X" in column 52 enters the machine, programs 1 through 8 are skipped. In this program $\sum_{16}^{1} M\,\%$ is read out of 3L and read into 7R for punching. Similarly, $\sum_{16}^{1} M\% \times$ M. W. F. enters 6R.

Program Step 10

Both punching storage units are impulsed to punch the summations in the X52 card. Read is impulsed to start the next card. The X52 card now contains the normalizing factors ($\sum_{16}^{1} M\,\%$) ($\sum_{16}^{1} Pr$) for the final computation on the next control panel.

Programs 11, 12, 13, and 14

These programs are similar to programs 9 and 10 with the exception that they are initiated by X55 and X60 cards.

Summary of Cards Thus Far

Nineteen answer cards have been processed and three trailer cards have been produced containing the partial and total summations.

Each answer card now contains

A. Unnormalized answer

B. Molecular weight factor

C. A times B

X52 card contains

$\sum_{16}^{1} M\,\%$ and $\sum_{16}^{1} Pr$. These are the normalizing factors for obtaining an analysis without air, nitrogen-carbon monoxide and carbon dioxide.

X55 card contains

$\sum_{18}^{1} M\,\%$ and $\sum_{18}^{1} Pr$. These are the air free normalization factors.

X60 card contains

$\sum_{19}^{1} M\,\%$ and $\sum_{19}^{1}$ Pr which are the total sample normalization factors.

Figure 5 is the actual control panel wiring for the above calculations. Normally the 602-A Calculating Punch has 12 programs, however, 14 were required for this operation. The two extra program steps were obtained on this problem by wiring the program exit hubs of P_1 and P_2 through co-selectors # 5 and # 6. When the co-selectors are not energized, (the normal position) P_1 and P_2 are performed. When a card with "X" punched in column 60 enters the machine the co-selectors are energized by pilot selector # 6. P_1 and P_2 when wired from the transfer side of the co-selectors actually perform P_{13} and P_{14} instead of P_1 and P_2 . In other words, P_1 and P_2 exit hubs are used twice, once to perform P_1 and P_2 and once for P_{13} and P_{14} .

THE NORMALIZATION CONTROL PANEL # 3
(Figures 6 and 7)

Read Cycle

The read cycle has been split into four parts in order to enter the divisors from X52, X55, and X60 cards into storage.

The X52 card picks up pilot selector #1 through the control brushes and reads $\sum_{16}^{1}$ Pr into storage unit 6L and $\sum_{16}^{1}$ M % into storage unit 6R. Similarly, the X55 card reads $\sum_{18}^{1}$ Pr and $\sum_{18}^{1}$ M % into 2L and 2R, while the X60 card reads $\sum_{19}^{1}$ Pr and $\sum_{19}^{1}$ M % into 3L and 3R.

The next card to be read will be an X14 or an X15 answer card. These cards will pick up pilot selector # 4 to control the reading of the unnormalized mole per cent answer (M %) into storage unit 4R and the product M % × M.W.F. into storage unit 4L.

Program Step 1 (Division Set Up)

The divisor $\sum_{16}^{1}$ M % in storage unit 6R is read into the division storage unit 1R and the dividend M % in storage unit 4R is read into the dividend counters (1, 2, 3).

Program Step 2 (Division)

In order to perform division the divisor is subtracted from the dividend in a sequence until the dividend is reduced to zero. The number of times required to do this will be the quotient which will be developed in the counters (5, 6).

1R is read into (1, 2, 3) negatively, the quotient is read into (5, 6) and the divide hub is impulsed.

Program Step 3 (Punch Normalized Answer)

Counters (5, 6) now contain the normalized answer for M % fixed gas free. This value is punched.

Counters (5, 6) are read out to punching storage 7R and punched in the card. At the same time (5, 6) are reset to 5 in order to round off the answer. At this time the card is held in the machine for the next division.

Program Steps 4, 5, 6

These steps are similar to P_1, P_2, P_3 and the normalized M % air free answer is obtained because the divisor this time was $\sum_{18}^{1}$ M %.

Program Steps 7, 8, 9

These steps are again similar to P_1, P_2 and P_3 and the total M % normalized answer is obtained by using $\sum_{19}^{1}$ M % for the divisor.

Program Steps 10, 11, 12

These steps calculate the normalized weight per cent fixed gas free answer by changing the dividend to M % × M.W.F. and the divisor to $\sum_{16}^{1}$ Pr.

At the completion of program 12 the read hub is impulsed through the normal side of pilot selector #6, and the machine reads the next card.

Program Step 13

The next six program steps will be initiated only for answer cards containing an "X" in column 28. The X28 card will pick up pilot selector # 6 and after program step 12 has been completed, program 1 exit hubs will again become active because the read hub impulse from program 12 will be broken by pilot selector # 6. At this time, co-selectors 1, 5 and 8 will be energized and program 1 will function as program 13.

In this program $\sum_{16}^{1}$ Pr in storage unit 6L is read into the dividend counters (1, 2, 3) and $\sum_{16}^{1}$ M % in storage unit 6R is read into the divisor storage unit 1R.

Program Steps 14 and 15

These steps are identical to P_2 and P_3 except that specific gravity fixed gas free is computed and punched.

Program Steps 16, 17, 18

These programs are similar to programs 13, 14, and 15 and the air free specific gravity is calculated and punched.

Summary of Cards Processed Thus Far

1. The matrix calculation produced 19 answer cards.
2. The summation and multiplication calculation punched the M % × M.W.F. in the 19 answer cards and produced three cards containing the six normalization factors.
3. The normalization calculation punched four normalized answers in each of the 19 answer cards and in the 19th answer card and the two specific gravities were also punched.

Figure 7 shows the control panel for the normalization control panel. In general, the wiring is straightforward and may be duplicated from the planning chart by persons familiar with the equipment. Programs 2, 5, 8, 11, 14 and 17 are in all respects identical and have been so wired. Programs 3, 6, 9, 12, 15 and 18 are also identical. On programs 1, 4, 7, 10, 13 and 16, storage unit 1R and counters (1, 2, 3) are read in and have been so wired. One exit hub of program 1 has been wired through co-selector # 5 to have this exit hub read out of 4R on program 1 and read out 6 L on program 13. Similarly, one exit hub of program 4 has been wired normally through co-selector # 5 to read out 4R. On program 16, this exit hub reads out 2 L.

In order to prevent a back circuit with the exit wiring from storage unit 6 L, these wires were run through co-selector # 1. Co-selector # 8 was used to stop the skip impulse from interfering with punching in column 61 when the specific gravities are punched in the X28 card.

19 X 19 INVERTED MATRIX PLUS SUMMATION CHECK

$$
\left[\begin{array}{cccc|cccc|cccc|cccc|ccc}
B_{1,1} & B_{1,2} & B_{1,3} & B_{1,4} & B_{1,5} & B_{1,6} & B_{1,7} & B_{1,8} & B_{1,9} & B_{1,10} & B_{1,11} & B_{1,12} & B_{1,13} & B_{1,14} & B_{1,15} & B_{1,16} & B_{1,17} & B_{1,18} & B_{1,19} \\
B_{2,1} & B_{2,2} & B_{2,3} & B_{2,4} & B_{2,5} & B_{2,6} & B_{2,7} & B_{2,8} & B_{2,9} & B_{2,10} & B_{2,11} & B_{2,12} & B_{2,13} & B_{2,14} & B_{2,15} & B_{2,16} & B_{2,17} & B_{2,18} & B_{2,19} \\
\hline
B_{3,1} & B_{3,2} & B_{3,3} & B_{3,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
B_{4,1} & B_{4,2} & B_{4,3} & B_{4,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\hline
B_{5,1} & B_{5,2} & B_{5,3} & B_{5,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
B_{6,1} & B_{6,2} & B_{6,3} & B_{6,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\hline
B_{7,1} & B_{7,2} & B_{7,3} & B_{7,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
B_{8,1} & B_{8,2} & B_{8,3} & B_{8,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\hline
B_{9,1} & B_{9,2} & B_{9,3} & B_{9,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
B_{10,1} & B_{10,2} & B_{10,3} & B_{10,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\hline
B_{11,1} & B_{11,2} & B_{11,3} & B_{11,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
B_{12,1} & B_{12,2} & B_{12,3} & B_{12,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\hline
B_{13,1} & B_{13,2} & B_{13,3} & B_{13,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
B_{14,1} & B_{14,2} & B_{14,3} & B_{14,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\hline
B_{15,1} & B_{15,2} & B_{15,3} & B_{15,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
B_{16,1} & B_{16,2} & B_{16,3} & B_{16,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\hline
B_{17,1} & B_{17,2} & B_{17,3} & B_{17,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
B_{18,1} & B_{18,2} & B_{18,3} & B_{18,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\hline
B_{19,1} & B_{19,2} & B_{19,3} & B_{19,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet \\
\Sigma B_{n,1} & \Sigma B_{n,2} & \Sigma B_{n,3} & \Sigma B_{n,4} & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet & \bullet
\end{array}\right]
\ \text{TIMES}\
\left[\begin{array}{c}
b_1 \\ b_2 \\ \hline b_3 \\ b_4 \\ \hline b_5 \\ b_6 \\ \hline b_7 \\ b_8 \\ \hline b_9 \\ b_{10} \\ \hline b_{11} \\ b_{12} \\ \hline b_{13} \\ b_{14} \\ \hline b_{15} \\ b_{16} \\ \hline b_{17} \\ b_{18} \\ \hline b_{19} \\ \
\end{array}\right]
=
\left[\begin{array}{c}
X_1 \\ X_2 \\ \hline X_3 \\ X_4 \\ \hline X_5 \\ X_6 \\ \hline X_7 \\ X_8 \\ \hline X_9 \\ X_{10} \\ \hline X_{11} \\ X_{12} \\ \hline X_{13} \\ X_{14} \\ \hline X_{15} \\ X_{16} \\ \hline X_{17} \\ X_{18} \\ \hline X_{19} \\ \Sigma X_n
\end{array}\right]
$$

FIGURE 1

PROGRAM SUPPRESS	PROGRAM STEP	OPERATION	STORAGE UNIT		COUNTER						STORAGE UNITS										
				DIVR.-MULT.	DIVIDEND												PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH				
			1L	1R	1	2	3	4	5	6	2L	2R	3L	3R	4L	4R	6L	6R	7L	7R	
		READ CYCLE		R.I. b_1 21-25	R.I. B_{24} 76-80						R.I. b_2 26-30	R.I. b_3 31-35	R.I. b_4 36-40	R.I. B_{11} 41-45	R.I. B_{12} 46-50	R.I. B_{13} 51-55	R.I. B_{14} 56-60	R.I. B_{21} 61-65	R.I. B_{22} 66-70	R.I. B_{23} 71-75	X8 DETAIL CARD SKIP OUT
	1	SIGN CONTROL												R.O. to P.S.#1				R.O. to P.S.#2			SEE TEXT
	2	b_1 x B_{11} b_1 x B_{21}		R.O. MULT.		R.I.±			R.I.±					R.O. to [2,3]				R.O. to [5,6]			
	3	SIGN CONTROL		R.I.							R.O. to 1R				R.O. to P.S.#1				R.O. to P.S.#2		
	4	b_2 x B_{12} b_2 x B_{22}		R.O. MULT.		R.I.±			R.I.±						R.O. to [2,3]				R.O. to [5,6]		
	5	SIGN CONTROL		R.I.								R.O. to 1R				R.O. to P.S.#1				R.O. to P.S.#2	
	6	b_3 x B_{13} b_3 x B_{23}		R.O. MULT.		R.I.±			R.I.±							R.O. to [2,3]				R.O. to [5,6]	
	7	SIGN CONTROL		R.I.	R.O. to P.S.#2								R.O. to 1R				R.O. to P.S.#1				
	8	b_4 x B_{14} b_4 x B_{24}		R.O. MULT.	R.O. to [5,6]	R.I.±			R I±								R.O. to [2,3]				
	9	IMPULSE READ																			
	10	PUNCH EVEN M% ANSWER				R.O. AND RESET											R.I. FROM [2,3] PUNCH 16-24				X14 TRAILER CARD ONLY
	11	PUNCH ODD M% ANSWER							R.O. AND RESET								R.I. FROM [5,6] PUNCH 16-24				X15 TRAILER CARD ONLY
	12	IMPULSE READ			RESET																

Figure 2. MATRIX CALCULATION

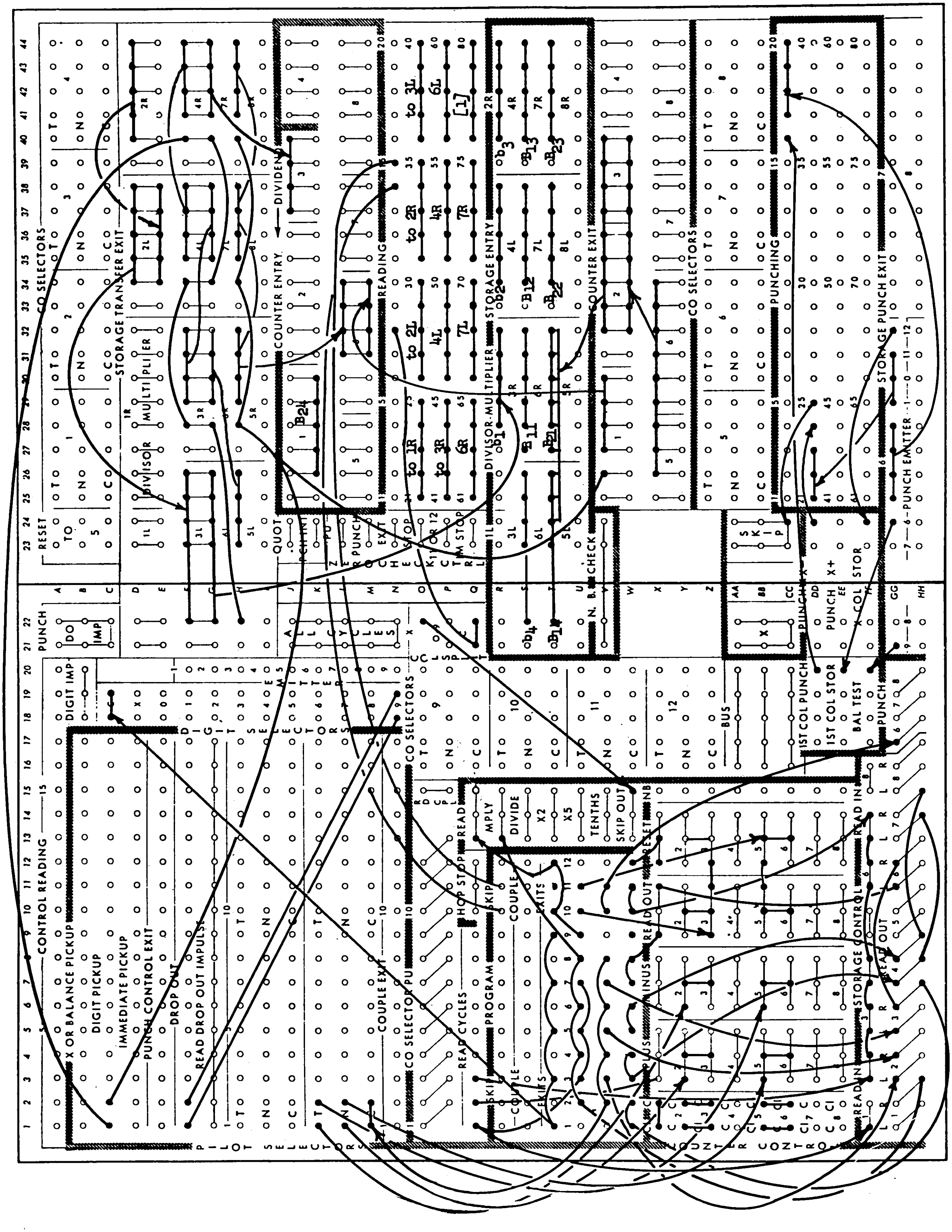


Figure 3. MATRIX CALCULATION

PROGRAM SUPPRESS	PROGRAM STEP	OPERATION	STORAGE UNIT		COUNTER						STORAGE UNITS										
				DIVR.–MULT.	DIVIDEND												PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH				
			1L	1R	1	2	3	4	5	6	2L	2R	3L	3R	4L	4R	6L	6R *	7L	7R *	
		READ CYCLE		R.I. M.W.F. 25-29	R.I. M% 1721							R.I. M% 17-21									
	1	M% x M.W.F.							R.I.+			R.O. to [5,6]									
	2	STORE PRODUCT				R.I.+ FROM [5,6]			R.O. AND RESET									R.I. FROM [5,6]			
	3	PUNCH IMPULSE READ																PUNCH 30-35			
	4	STORE Σ 16			R.O. to 3L	R.O. to 3R							R.I. $\Sigma_{16}^{M\%}$	R.I. $\Sigma_{16}^{Pr.}$							X22 CARD OMITS P_3
	5	PUNCH IMPULSE READ																PUNCH 30-35			
	6	STORE Σ 18			R.O. to 4L	R.O. to 4R									R.I. $\Sigma_{18}^{M\%}$	R.I. $\Sigma_{18}^{Pr.}$					X25 CARD OMITS $P_{3,4,5}$
	7	PUNCH IMPULSE READ																PUNCH 30-35			
	8	OMIT																			
	9												R.O. to 7R	R.O. to 6R				R.I. $\Sigma_{16}^{Pr.}$		R.I. $\Sigma_{16}^{M\%}$	X52 CARD STARTS AT P_9
	10	PUNCH IMPULSE READ																PUNCH 30-35		PUNCH 38-43	
	11														R.O. to 7R	R.O. to 6R		R.I. $\Sigma_{18}^{Pr.}$		R.I. $\Sigma_{18}^{M\%}$	X55 CARD STARTS AT P_{11}
	12	PUNCH IMPULSE READ																PUNCH 30-35		PUNCH 38-43	
	13				R.O. R.S.	R.O.&R.S. to 7R												R.I. $\Sigma_{19}^{Pr.}$		R.I. $\Sigma_{19}^{M\%}$	X 60 STARTS P_{13}
	14	PUNCH IMPULSE READ																PUNCH 30-35		PUNCH 38-43	

Figure 4. MOLECULAR WEIGHT FACTOR MULTIPLICATION AND SUMMATION

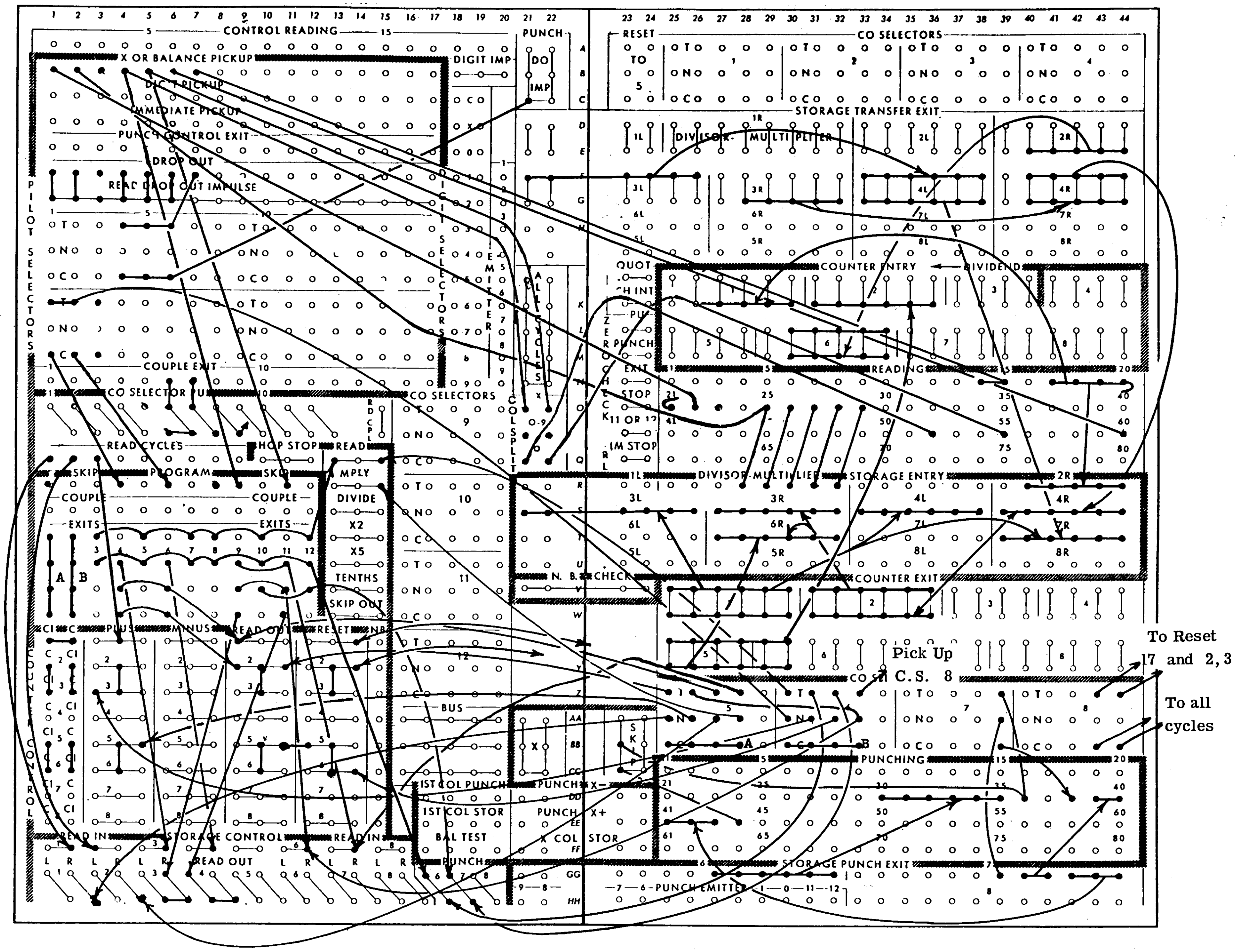


Figure 5. MOLECULAR WEIGHT FACTOR

		OPERATION	STORAGE UNIT		COUNTER						STORAGE UNITS										
				DIVR.-MULT.	DIVIDEND												PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH				
PROGRAM SUPPRESS	PROGRAM STEP		1L	1R	1	2	3	4	5	6	2L	2R	3L	3R	4L	4R	6L	6R	7L	7R	
CONTROL		READ THRU READ CYCLE	P. S.	#1 #2 #3							Σ_{18}Pr.	Σ_{18}M%	Σ_{19}Pr.	Σ_{19}M%			Σ_{16}Pr.	Σ_{16}M%			X52 CARD X55 CARD X60 CARD
CONTROL	1	READ THRU	P. S.	#4 R.I.	R.I.+										M% x MWF	M% R.O. to 1,2,3		R.O. to 1R			X14 or X15
	2	$\frac{M\%}{\Sigma_{16}M\%}$		R.O. DIVIDE	R.I.-				R.I.+												
	3	PUNCH M% F.G. FREE			RESET				R.O. to 7R RESET											R.I. PUNCH 45-48	
	4			R.I.	R.I.+							R.O. to 1R				R.O. to [1,2,3]					
	5	$\frac{M\%}{\Sigma_{18}M\%}$		R.O. DIVIDE	R.I.-				R.I.+												
	6	PUNCH M% AIR FREE			RESET				R.O. to 7R RESET											R.I. PUNCH 49-52	
	7			R.I.	R.I.+									R.O. to 1R		R.O. to [1,2,3]					
	8	$\frac{M\%}{\Sigma_{19}M\%}$		R.O. DIVIDE	R.I.-				R.I.+												
	9	PUNCH M%			RESET				R.O. to 7R RESET											R.I. PUNCH 53-56	
	10			R.I.	R.I.+										R.O. to [1,2,3]		R.O. to 1R			R.I. PUNCH	
	11	$\frac{M\%\ M.W.F.}{\Sigma_{16}Pr.}$		R.O. DIVIDE	R.I.-				R.I.+											R.I. PUNCH 57-60	
	12	PUNCH WT.%							R.O. to 7R RESET											R.I. PUNCH 57-60	
	13			R.I.	R.I.+												R.O. to [1,2,3]	R.O. to 1R			X 28 CARD
	14	$\frac{\Sigma_{16}\ Pr.}{\Sigma_{16}M\%}$		R.O. DIVIDE	R.I.-				R.I.+												
	15	PUNCH S.G. F.G. FREE																		R.I. PUNCH 61-65	
	16			R.I.	R.I.+						R.O. to [1,2,3]	R.O. to 1R									
	17	$\frac{\Sigma_{18}\ Pr.}{\Sigma_{18}M\%}$		R.O. DIVIDE	R.I.-				R.I.+												
	18	PUNCH S.G. AIR FREE							R.O. to 7R RESET											R.I. PUNCH 66-70	

Figure 6. NORMALIZATION

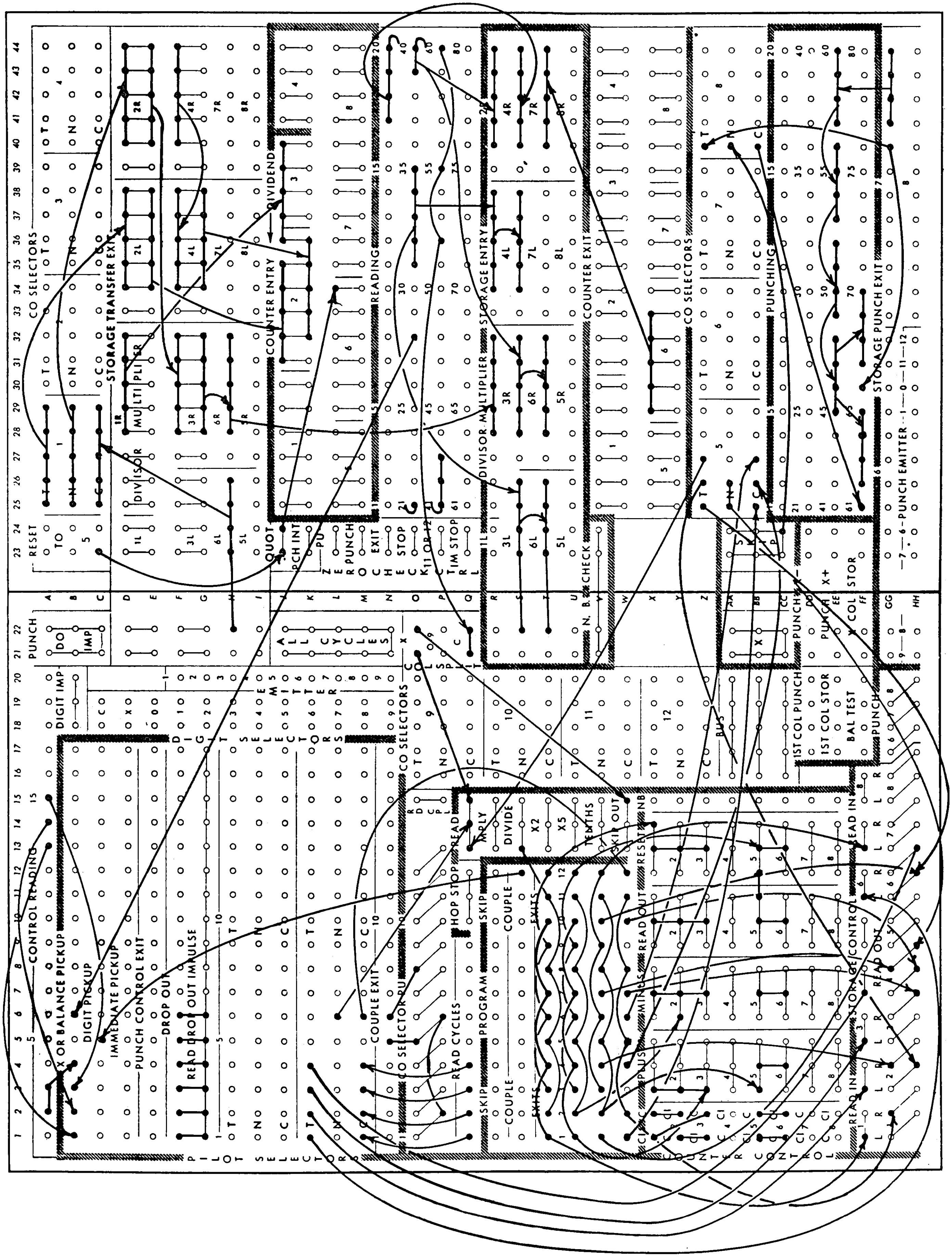


Figure 7. NORMALIZATION

COMPUTATIONS OF INVERSE MATRICES BY MEANS OF IBM MACHINES

Jack Sherman

The Texas Company Research Laboratories

An efficient method for solving the sets of simultaneous linear equations which arise in mass spectrometer analyses is to utilize the inverse matrix of the calibration coefficients. Once the matrix has been obtained, its application to the solution of simultaneous linear equations merely consists of computing sums of binary products. However, the task of obtaining the inverse matrix is considerable inasmuch as the order of the matrix generally ranges from 13-20 in routine operations.

Many methods have been and are being published for obtaining the inverse of a matrix.[a] These generally involve some variation of the method of systematic elimination, such as Doolittle's, and were designed with the idea that the computations would be carried out by means of a desk calculator. It turns out that these methods are not particularly efficient for use with IBM machines.

It is the purpose of this paper to describe an efficient method of obtaining an inverse matrix by means of punched cards which has been found to be relatively easy to carry out and which requires considerably less time than for any of the standard methods.

The computation of the inverse of a matrix of pattern or sensitivity coefficients is begun by means of the previously obtained inverse (as will be seen in the following description of the method, the final inverse does not require that this be the starting point. However, it is convenient to do so.) This starting inverse is transformed into another inverse which results from replacing the first column in the original matrix by the first column corresponding to the new matrix. The equations for doing this are given in an abstract of the paper presented at the Twelfth Summer Meeting of the Institute of Mathematical Statistics at Boulder, Colorado.[b] This may be written as follows:

[a] Cf. H. Hotelling, "Some New Methods in Matrix Calculations", Annals of Mathematical Statistics, 14, pp. 1-34 (1943).

[b] Annals of Mathematical Statistics, Volume 20, page 621, (1949).

Denote the original inverse by [b] whose elements are b_{ij}, the new inverse by [b′] whose elements are b'_{ij}, the original matrix by [a] whose elements are a_{ij}, and the new matrix by [a′] whose elements are a'_{ij}. If [a′] differs from [a] only in the elements of the k th column, then

$$b'_{ij} = b_{ij} - z_i\, b'_{kj}, \qquad i = 1,2,\ldots,k\text{-}1,k{+}1,\ldots N \quad j = 1,2,\ldots N \tag{1}$$

$$b'_{kj} = b_{kj}/z_k, \qquad j = 1,2,\ldots N \tag{2}$$

$$z_i = \sum_{r=1}^{N} b_{ir}\, a'_{rk}, \qquad i = 1,2,\ldots N \tag{3}$$

By letting k = 1,2,...N successively, the new inverse matrix is obtained from the old inverse matrix by successively replacing each column of the given matrix [a] by a column of the new matrix [a′]. The N successive matrices obtained in this process correspond to inverses of the old matrix [a] in which the first column has been replaced by the new calibration coefficients, the old inverse in which the first two columns have been replaced by new calibration coefficients, etc.

Although the foregoing procedure of transforming the old inverse matrix into a new inverse appears to involve a greater absolute number of arithmetical steps than perhaps the Doolittle method, it can be very efficiently carried out on the IBM machines. The following series of steps which are utilized in making one transformation are listed below:

1) N^2 cards are prepared by punching one element of the starting inverse into a card in columns 26-31, the order of the cards being elements of successive rows, proceeding from left to right.

A negative value of an element is indicated by an X in column 31. These cards contain an X in column 52 for control purposes.

For identification purposes the values of i and j for each element and the number of the inverse are punched in columns 1-6, thus 050800 denotes b_{58} of matrix 0, the starting matrix.

2) A second set of N^2 cards is prepared containing the values of the indices i and j in columns 1-4 together with the number 01 in columns 5 and 6 to denote inverse matrix number 1 (the [b′] matrix).

3) The N values of a'_{i1} are reproduced N times into N^2 cards in columns 71-76 of matrix 0.

4) Behind each set of N cards of inverse matrix 0, a trailer card is inserted by means of the collator.

5) The z_i's are computed utilizing equation 3. These z_i's are punched in columns 60-68. An X in column 68 denotes a negative value of z.

It should be noted that if the original elements of the first column of a are used rather than a new set of elements, the values of the z's would be 1, 0, 0, ... If the calibration coefficients a'_{ij} do not differ greatly from the original values a_{ij}, the z values will tend to be close to the 1, 0, 0, . . . , and consequently the order of magnitude of the z values is well controlled. This is the reason for starting with the old inverse.

6) The reciprocal of z_1, is obtained by means of a desk calculator (or reciprocal control panel) and a new card is punched with

$$z'_1 = \frac{1}{z_1} ,$$

the digits being placed in corresponding positions of the z' cards.

7) The z'_1 value is gang-punched into a new set of N cards in columns 61-68, at the same time reproducing b_{1j} values from card columns 26-31 of the first N cards of original matrix into card columns 16-21.

8) The first N cards of matrix number 1 are interspersed with the N cards obtained in the previous step by means of the collator.

9) The values of b'_{1j} are computed in card columns 16-21 according to equation 2 which may be written as

$$b'_{1j} = 0 + z'_1 \; b_{1j} , \tag{4}$$

the computations being arranged so that z'_1 and b_{1j} are obtained from a card prepared as described in step number 7, and the corresponding values of b'_{1j} being punched on the following card, i.e., a card of the number 1 matrix.

10) The values of b'_{1j} are reproduced into card columns 16-21 of the remaining N^2 -N cards of the [b] matrix (matrix 0); at the same time the z_i values are gang-punched into columns 61-68.

11) The corresponding $N^2 - N$ b' cards (matrix number 1) are interspersed with the b cards by means of the collator.

12) The b' values are computed according to the equation 1.

It should be noted that equations 1 and 4 are similar in form, the only difference being that for equation 4 the product term is added whereas in equation 1 it is subtracted. Hence, the same control panel can be used for both computations.

13) The set of two (N^2-N) cards are separated by means of the sorter into the b and b′ cards. For a 13th order matrix, each of the N transformations can be carried out in approximately 25 to 30 minutes, and so the desired inverse can be obtained in approximately seven hours.

The following points should be noted concerning the method of a computation described in the foregoing steps:

A running check not described in the foregoing series of steps is introduced in order to detect any error before the computation is completed. This consists of inserting a column of the a′ matrix into the cards so that when the z's are computed according to step 5, additional summations are carried out, namely, $\sum b'_{ir}\ a'_{rj}$. These summations must yield the values 0, 0, . . . 1, 0, ...

After the foregoing steps are carried out N times to obtain the desired inverse, a complete check is carried out by multiplying the new inverse b′ with the corresponding given matrix a′ to obtain the unit matrix.

Difficulties associated with the position of the decimal point in division are avoided by the fact that the z′ values are obtained separately by means of a desk calculator (or reciprocal control panel).

In addition to the advantage that is gained by making the computation completely routine, the method of inverting a matrix described in the previous steps provides a considerable degree of flexibility; thus, if only a few of the N components are recalibrated, the corresponding new inverse can be obtained with considerably less computation than would be required for the complete calibration.

Although the series of steps described in this paper for computing an inverse matrix requires the use of the calculator, sorter, reproducer, and collator, the sequence is easily learned by the machine operator after one or two examples are carried out.

LINEAR EQUATIONS AND MATRIX INVERSION

Eric V. Hankam

Watson Scientific Computing Laboratory

A. Mathematical Preliminaries

1. Linear Equations:

Consider a square matrix of order n (let n = 4 for simplicity)

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & b_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & b_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & b_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & b_{45} \end{pmatrix} \qquad (1)$$

augmented by a column vector b, representing a set of four simultaneous linear equations.

I. a) Perform the following transformation: Divide each element of the first row by its leading element (a_{11}), thus obtaining:

$$\begin{pmatrix} 1 & \frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} & \frac{a_{14}}{a_{11}} & \frac{b_{15}}{a_{11}} \\ a_{21} & a_{22} & \cdot \; \cdot & \cdot \; \cdot & b_{25} \\ \vdots & & & & \vdots \\ a_{41} & \cdot \; \cdot & \cdot \; \cdot & \cdot \; \cdot & b_{45} \end{pmatrix}. \qquad (2)$$

b) Subract a_{11} times the first row from the respective elements of the succeeding rows (i = 2, 3, 4):

$$\begin{pmatrix} 1 & \frac{a_{12}}{a_{11}} & \cdots & \frac{b_{15}}{a_{11}} \\ 0 & a_{22} - a_{21}\left(\frac{a_{12}}{a_{11}}\right) & \cdots & b_{25} - a_{21}\left(\frac{b_{15}}{a_{11}}\right) \\ 0 & \vdots & & \vdots \\ 0 & a_{42} - a_{41}\left(\frac{a_{12}}{a_{11}}\right) & \cdots & b_{45} - a_{41}\left(\frac{b_{15}}{a_{11}}\right) \end{pmatrix}. \qquad (3)$$

Note: In place of (2) each row may be divided by its leading elements:

$$\begin{pmatrix} 1 & \frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} & \frac{a_{14}}{a_{11}} & \frac{b_{15}}{a_{11}} \\ 1 & \frac{a_{22}}{a_{21}} & \cdots & \cdots & \frac{b_{25}}{a_{21}} \\ 1 & \vdots & & & \vdots \\ 1 & \frac{a_{42}}{a_{41}} & \cdots & \cdots & \frac{b_{45}}{a_{41}} \end{pmatrix}. \qquad (2^*)$$

The first row is then subtracted from all the other rows:

$$\begin{pmatrix} 1 & \frac{a_{12}}{a_{11}} & \cdots & \frac{b_{15}}{a_{11}} \\ 0 & \frac{a_{22}}{a_{21}} - \frac{a_{12}}{a_{11}} & \cdots & \frac{b_{25}}{a_{21}} - \frac{b_{15}}{a_{11}} \\ 0 & \vdots & & \vdots \\ 0 & \frac{a_{42}}{a_{41}} - \frac{a_{12}}{a_{11}} & \cdots & \frac{b_{45}}{a_{41}} - \frac{b_{15}}{a_{11}} \end{pmatrix}. \qquad (3^*)$$

The reduced matrix of (3) or (3*) may be written in the form

$$\begin{pmatrix} a'_{22} & a'_{23} & a'_{24} & b'_{25} \\ a'_{32} & \cdots & \cdots & b'_{35} \\ a'_{42} & \cdots & \cdots & b'_{45} \end{pmatrix}. \qquad (4)$$

II. Applying to (4) the same transformation as in step I., (4) is reduced to a second order matrix:

$$\begin{pmatrix} a''_{33} & a''_{34} & b''_{35} \\ a''_{42} & a''_{44} & b''_{45} \end{pmatrix}. \qquad (5)$$

III. Reducing (5) similarly, the following is obtained:

$$\begin{pmatrix} a'''_{44} & b'''_{45} \end{pmatrix}. \qquad (6)$$

Performing the transformation of step Ia) on (6), we obtain

$$\begin{pmatrix} 1 & \frac{b'''_{45}}{a'''_{44}} = x_4 \end{pmatrix}, \qquad (7)$$

where x_4 = the fourth variable of the set of equations.

This leaves the original matrix in a triangular form:

$$\begin{pmatrix} 1 & \frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} & \frac{a_{14}}{a_{11}} & \frac{b_{15}}{a_{11}} \\ 0 & 1 & \frac{a'_{23}}{a'_{22}} & \frac{a'_{24}}{a'_{22}} & \frac{b'_{25}}{a'_{22}} \\ 0 & 0 & 1 & \frac{a''_{34}}{a''_{33}} & \frac{b''_{35}}{a''_{33}} \\ 0 & 0 & 0 & 1 & \frac{b'''_{45}}{a'''_{44}} = x_4 \end{pmatrix}. \qquad (8)$$

(8) may be written:

$$\begin{pmatrix} 1 & \bar{a}_{12} & \bar{a}_{13} & \bar{a}_{14} & \bar{b}_{15} \\ 0 & 1 & \bar{a}_{23} & \bar{a}_{24} & \bar{b}_{25} \\ 0 & 0 & 1 & \bar{a}_{34} & \bar{b}_{35} \\ 0 & 0 & 0 & 1 & \bar{b}_{45} = x_4 \end{pmatrix}. \qquad (\bar{8})$$

IV. In order to reduce $(\bar{8})$ to the unit matrix, the following transformations are performed on rows 1, 2, 3 of $(\bar{8})$: Multiply an element of the fourth column by the elements of the last row and subtract the products from the respective row yielding (9) (i.e., $\bar{a}_{ij} - \bar{a}_{i4}\,\bar{a}_{4j}$; $i = 1, 2, 3$; $j = 1, 2, \ldots, 5$). This transformation leaves columns 1, 2, 3 unchanged;

$$\begin{pmatrix} 1 & \bar{a}_{12} & \bar{a}_{13} & 0 & \bar{b}_{15} - \bar{a}_{14}\,x_4 \\ 0 & 1 & \bar{a}_{23} & 0 & \bar{b}_{25} - \bar{a}_{24}\,x_4 \\ 0 & 0 & 1 & 0 & \bar{b}_{35} - \bar{a}_{34}\,x_4 = x_3 \end{pmatrix}. \qquad (9)$$

Rewriting (9) with $\bar{\bar{b}}_{15}$ = elements of fifth column, and $\bar{a}_{ij} = \bar{\bar{a}}_{ij}$, we have

$$\begin{pmatrix} 1 & \bar{\bar{a}}_{12} & \bar{\bar{a}}_{13} & 0 & \bar{\bar{b}}_{15} \\ 0 & 1 & \bar{\bar{a}}_{23} & 0 & \bar{\bar{b}}_{25} \\ 0 & 0 & 1 & 0 & \bar{\bar{b}}_{35} = x_3 \end{pmatrix}. \qquad (\bar{9})$$

V. To obtain the next matrix a transformation similar to step IV is applied to $(\bar{9})$:

$$\begin{pmatrix} 1 & \bar{\bar{a}}_{12} & 0 & 0 & \bar{\bar{b}}_{15} - \bar{\bar{a}}_{13}\,x_3 \\ 0 & 0 & 0 & 0 & \bar{\bar{b}}_{25} - \bar{\bar{a}}_{23}\,x_2 = x_2 \end{pmatrix}. \qquad (10)$$

Rewriting (10) as was done above with (9), we have

$$\begin{pmatrix} 1 & \bar{\bar{\bar{a}}}_{12} & 0 & 0 & \bar{\bar{\bar{b}}}_{15} \\ 0 & 0 & 0 & 0 & \bar{\bar{\bar{b}}}_{25} = x_2 \end{pmatrix}. \qquad (\overline{10})$$

VI. Finally, repeating as before, we obtain

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \bar{\bar{\bar{b}}}_{15} - \bar{\bar{\bar{a}}}_{12}\,x_2 = x_1 \end{pmatrix}. \qquad (11)$$

2. **Matrix Inversion:**

Consider (1) with the unit matrix I replacing the column vector b:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} & 1_{15} & 0_{16} & 0_{17} & 0_{18} \\ \cdot & & & \cdot & 0 & 1 & 0 & 0 \\ \cdot & & & \cdot & 0 & 0 & 1 & 0 \\ \cdot & & & \cdot & & & & \\ a_{41} & \cdot\;\cdot\;\cdot\;\cdot & \cdot\;\cdot\;\cdot & a_{44} & 0_{45} & 0 & 0 & 1_{48} \end{pmatrix}. \qquad (1')$$

Proceed as in the case of linear equations step Ia) and b) to obtain:

$$\begin{pmatrix} 1 & \frac{a_{12}}{a_{11}} \cdots \frac{a_{14}}{a_{11}} & \frac{1}{a_{11}} & 0 & 0 & 0 \\ a_{21} & a_{22} \cdots a_{24} & 0 & 1 & 0 & 0 \\ \vdots & \vdots \quad \vdots & & & & \\ a_{41} & a_{42} \cdots a_{44} & 0 & 0 & 0 & 1 \end{pmatrix} \qquad (2')$$

and

$$\begin{pmatrix} 1 & \frac{a_{12}}{a_{11}} \quad \cdots & \frac{1}{a_{11}} & 0 & 0 & 0 \\ 0 & a_{22} - a_{21}\left(\frac{a_{12}}{a_{11}}\right) \cdots & -a_{21}\left(\frac{1}{a_{11}}\right) & 1 & 0 & 0 \\ & \vdots & \vdots & & & \\ 0 & a_{42} - a_{41}\left(\frac{a_{12}}{a_{11}}\right) \cdots & -a_{41}\left(\frac{1}{a_{11}}\right) & 0 & 0 & 1 \end{pmatrix}. \qquad (3')$$

Here again transformations (2*) and (3*) can be used instead of (2) and (3).

Proceeding as in steps II and III a matrix corresponding to $(\bar{8})$ is obtained:

$$\begin{pmatrix} 1 & \bar{a}_{12} & \bar{a}_{13} & \bar{a}_{14} & \bar{b}_{15} & \bar{0}_{16} & \bar{0}_{17} & \bar{0}_{18} \\ 0 & 1 & \bar{a}_{23} & \bar{a}_{24} & \bar{b}_{25} & \bar{b}_{26} & 0 & 0 \\ 0 & 0 & 1 & \bar{a}_{34} & \bar{b}_{35} & \bar{b}_{36} & \bar{b}_{37} & 0 \\ 0 & 0 & 0 & 1 & \bar{b}_{45} & \bar{b}_{46} & \bar{b}_{47} & \bar{b}_{48} \end{pmatrix}. \qquad \overline{(8')}$$

Corresponding to (9) we have

$$\begin{pmatrix} 1 & \bar{a}_{12} & \bar{a}_{13} & 0 & \bar{b}_{15} - \bar{a}_{14}\,\bar{b}_{45} & -\bar{a}_{14}\,\bar{b}_{46} & -\bar{a}_{14}\,\bar{b}_{47} & -\bar{a}_{14}\,\bar{b}_{48} \\ 0 & 1 & \bar{a}_{23} & 0 & \bar{b}_{25} - \bar{a}_{24}\,\bar{b}_{45} & \bar{b}_{26} - \bar{a}_{24}\,\bar{b}_{46} & -\bar{a}_{24}\,\bar{b}_{47} & -\bar{a}_{24}\,\bar{b}_{48} \\ 0 & 0 & 1 & 0 & \bar{b}_{35} - \bar{a}_{34}\,\bar{b}_{45} & \bar{b}_{36} - \bar{a}_{34}\,\bar{b}_{46} & \bar{b}_{37} - \bar{a}_{34}\,\bar{b}_{47} & -\bar{a}_{34}\,\bar{b}_{48} \end{pmatrix} \qquad (9')$$

where again we can write $\bar{\bar{b}}_{ij}$ for the elements of columns five to eight and $\bar{\bar{a}}_{ij}$ for $\bar{a}_{ij}$.

Finally corresponding to (11), the following is obtained:

$$\begin{pmatrix} 1 & 0 & 0 & 0 & \overset{\equiv}{b}_{15}-\overset{\equiv}{a}_{12}\overset{\equiv}{b}_{25} & \overset{\equiv}{b}_{16}-\overset{\equiv}{a}_{12}\overset{\equiv}{b}_{26} & \overset{\equiv}{b}_{17}-\overset{\equiv}{a}_{12}\overset{\equiv}{b}_{27} & \overset{\equiv}{b}_{18}-\overset{\equiv}{a}_{12}\overset{\equiv}{b}_{28} \end{pmatrix}. \quad (11')$$

Rewriting the entire matrix as before, the inverse results:

$$\begin{pmatrix} 1_{11} & 0 & 0 & 0_{14} & \overset{\equiv}{b}_{15} & \cdot\cdot\cdot\cdot & \overset{\equiv}{b}_{18} \\ 0 & 1 & 0 & 0 & \overset{\equiv}{b}_{25} & & \overset{\equiv}{b}_{28} \\ 0 & 0 & 1 & 0 & \overset{=}{b}_{35} & & \overset{=}{b}_{38} \\ 0_{41} & 0 & 0 & 1_{44} & \bar{b}_{45} & \cdot\cdot\cdot\cdot & \bar{b}_{48} \end{pmatrix}. \quad (\overline{11}')$$

Remarks:

α) Check sums: An entire matrix is augmented by an additional column whose elements are the sum of the elements of the respective row: i.e., $\sum_i = \sum_j a_{ij}$, where Σ_i represents the check sum of the i^{th} row. After transformation of the matrix, the sum of the transformed row elements is equal to the transformed element of the sum; i.e., $\sum_j T(a_{ij}) = T\Sigma_i$, due to linearity.

β) Linear equations with the same matrix of coefficients but different constant terms (b_i's) can be treated simultaneously by forming matrix (1) augmented by several column vectors of constant terms. Alternately we can solve for the inverse matrix A^{-1} and then perform matrix-vector multiplications $x = A^{-1}b$ for different sets of column vectors b.

γ) The "back-solution", i.e., the reduction of the triangular matrix (8) to the unit matrix, can be eliminated by the use of the following composite matrix:

$$\begin{pmatrix} A & b & I \\ -I & 0 & 0 \end{pmatrix}.$$

If we apply transformations (2) and (3) [here we cannot use (2*) and (3*)], we obtain the values of the unknowns x in the column of the original 0-vector located under the b-vector, and the inverse matrix A^{-1} will appear in the original 0-matrix located under the I-matrix. (Reference: Verzuh, F. M., The Solution of Simultaneous Linear Equations with the Aid of the 602 Calculating Punch, Mathematical Tables and Other Aids to Computation, Vol. 3, No. 27, July 1949).

The back solution can be avoided without the necessity of having to add the –I and 0 matrices by means of the following scheme: Following the first reduction the first row of the reduced matrix (3), instead of being discarded, is left in that matrix, but is relocated as its last row. (The first row is put below the last row for the purpose of uniformity in machine operations). The leading column is removed, as before.

The reduction is then performed again, this time with the original second row used as the first row. After this second reduction is complete, the original second row becomes its new last row, etc. Thus each row, in turn, is used as the leading row during one reduction, and following that reduction becomes the last row of the reduced matrix. The unknowns x will appear in the columns of the original b vector, and the inverse A^{-1} will appear in the original I-matrix.

B. Machine Procedure

1. Consider the matrix

$$\left(\underbrace{\begin{matrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{matrix}} \quad \underbrace{\begin{matrix} b_{15} & b_{16} \\ b_{25} & b_{26} \\ b_{35} & b_{36} \\ b_{45} & b_{46} \end{matrix}} \quad \underbrace{\begin{matrix} 1_{17} & 0_{18} & 0_{19} & 0_{1,10} \\ 0_{27} & 1_{28} & 0_{29} & 0_{2,10} \\ 0_{37} & 0_{38} & 1_{39} & 0_{3,10} \\ 0_{47} & 0_{48} & 0_{49} & 1_{4,10} \end{matrix}} \quad \underbrace{\begin{matrix} -\Sigma_{1,11} \\ -\Sigma_{2,11} \\ -\Sigma_{3,11} \\ -\Sigma_{4,11} \end{matrix}}\right).$$

Elements in columns 1-4 represent the matrix of the set of four simultaneous equations. Columns 1-4, 5 and 1-4, 6 are the augmented matrices representing the simultaneous equations with two sets of constant values. Both augmented matrices may be solved simultaneously. Elements in columns 7-10 represent the unit matrix which will become the inverse of the original matrix when the same transformations that will transform the original matrix to the unit matrix are applied to elements in columns 7-10. Elements in column eleven represent the negative check sums. The matrix above can be augmented by the negative unit matrix and the zero matrix as previously described (see previous section γ) for the purpose of avoiding the back solution.

As many sets of equations may be solved as desired. Each set is identified by a different equation number. Every element of each matrix to be solved is punched on a separate card.

For matrices with less then ten rows, one card column is sufficient for the row number. Two columns have been set aside here to indicate the procedure for larger matrices.

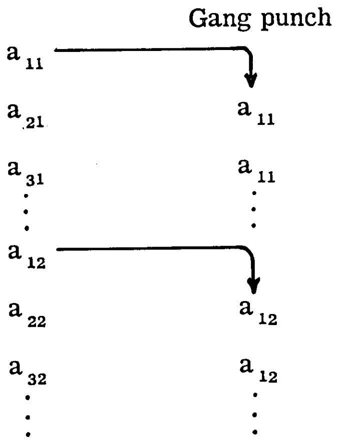
Punch the cards in the following manner (assuming each element to be an eight-digit number):

Col. 1-2 Eq. #	Col. 3-4 Row # i	Col. 5-6 Col. # j	Col. 7-14 Matrix element a_{ij}
01	01	01	$.\text{xxxxxxxx}^{\pm}$
01	01	02	
⋮	⋮	⋮	
01	01	11	
01	02	01	
01	02	02	
⋮	⋮	⋮	

2. The reduction of the matrix involves the repetitive applications of transformations (2) and (3). This can be done on the machines by repeating the following sequence of four main steps:

a. Sort the cards in sequence by columns (j) (i.e., sort on row# i first, then on column#j, finally on equation#).

b. Using row 1 cards as master cards, gang punch its elements into cards of the corresponding column, i.e.,



c. Sort the cards in sequence by rows (i) (i.e., sort on column#j first, then on row #i, finally on equation#).

d. Using column 1 cards as master cards, obtain on the calculating punches 602-A and 604

$$a_{ij} - \frac{a_{i1}\, a_{1j}}{a_{11}}$$

where a_{i1} and a_{11} will appear on master cards.

Remarks:

α) Master cards will have to be X-punched preparatory to step b and d above.[a]

β) After each reduction of the matrix to one of a lower order, the new elements a_{ij} can be reproduced into a new set of cards into the same field where the original a_{ij}'s were punched. Thus none of the control panels used in steps a to d have to be altered. Alternately reading and punching fields can be re-wired on the control panels and new elements can be punched successively to the right of the card thus reducing the number of cards used. After each reduction step the cards corresponding to the first row and column are set aside and the remaining cards represent the matrix of the next lower order. Only the first row of the composite matrix (–I, 0) need be used in the first reduction step; for the second reduction, the second row is added, etc.

γ) Check sums should be verified after each reduction step when the cards are in sequence of rows. Summing the elements in each row on the 405 should give a result close to zero since the negative of the check sum was punched in the original cards.

δ) Step d is first performed on the 602-A and is then verified on the 604 by means of the DPBC device. Note that operation of the type $\frac{AB}{C}$ can be done on one program step on the 602-A. The linear term a_{ij} can be read into the counter group, where

$$\frac{a_{i1}\,a_{1j}}{a_{11}}$$

will be developed at read time; a_{1j} should be used as the dividend since a_{i1} appears only on master cards and cannot be read directly into the dividend counter. The 604 operation should be programmed to divide first and then to multiply so that the calculation is done in exactly the same manner as on the 602-A, thus avoiding any possible discrepancy in the unit position of the result.

[a] An alternate method, avoiding X-punching, is suggested by William D. Bell in his article on " Punched Card Techniques for the Solution of Simultaneous Equations and other Matrix Operations," Proceedings, Scientific Computation Forum, pp. 28-31, New York: IBM, 1948.

MATRIX AND VECTOR ALGEBRA

Eric V. Hankam

Watson Scientific Computing Laboratory

A. Mathematical Preliminaries

Definition: A matrix is a rectangular array of elements:

$$A = (a_{ij}) = \begin{bmatrix} a_{11} & a_{12} & \cdot & \cdot & \cdot & a_{1n} \\ a_{21} & & & & & \cdot \\ \cdot & & & & & \cdot \\ \cdot & & & & & \cdot \\ a_{m1} & \cdot & \cdot & \cdot & \cdot & a_{mn} \end{bmatrix}.$$

In particular, if n = m, we refer to it as a square matrix.

An n-tuple of elements $(a_1 a_2 \ldots a_n)$ is called a vector; thus a 1 x n matrix $[a_1 a_2 \ldots a_n]$ is called a row vector, and an n x 1 matrix $\begin{bmatrix} a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_n \end{bmatrix}$ is called a column vector.

Matrix Algebra

1. Addition of Matrices:

Consider two matrices, $A = (a_{ij})$ and $B = (b_{ij})$. Their sum A + B is defined as that matrix C (c_{ij}) obtained by adding corresponding elements:

$$\begin{bmatrix} a_{11} \cdot & \cdot & \cdot\, a_{1n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ a_{m1} \cdot & \cdot & \cdot\, a_{mn} \end{bmatrix} + \begin{bmatrix} b_{11} \cdot & \cdot & \cdot\, b_{1n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ b_{m1} \cdot & \cdot & \cdot\, b_{mn} \end{bmatrix} = \begin{bmatrix} a_{11} + b_{11} & \cdot & \cdot & \cdot & a_{1n} + b_{1n} \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ \cdot & & & & \cdot \\ a_{m1} + b_{m1} & \cdot & \cdot & \cdot & a_{mn} + b_{mn} \end{bmatrix}$$

$$\text{i.e.,} \quad (a_{ij}) + (b_{ij}) = (a_{ij} + b_{ij}) = (c_{ij}).$$

2. Scalar Multiplication:

Consider a matrix $A = (a_{ij})$ and a constant c. Their scalar product cA is defined as that matrix obtained by multiplying every element a_{ij} of A by c:

$$c \cdot (a_{ij}) = (ca_{ij}).$$

3. Multiplication of Matrices:

Consider two matrices $A = (a_{ij})$ and $B = (b_{ij})$. Their product AB is defined as that matrix $C = (c_{ij})$ obtained in the following manner;

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} .$$

We see from the definition that the product AB is defined only if the number of columns of A equals the number of rows in B. Furthermore, note that $AB \neq BA$. The multiplication of two vectors is frequently called their "inner product". In accordance with the above definition, the inner product is the sum of products of the elements of the two vectors. Thus to obtain the product of two matrices A and B we have to find the inner products of every row vector of A with every column vector of B. The product of a matrix and a column vector arises frequently in the theory of linear equations.

4. Inversion of Matrices:

The inverse A^{-1} of a matrix A is defined as that matrix which satisfies the following relationship:

$$AA^{-1} = A^{-1}A = I,$$

where I is the unit matrix $\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdot & \cdot & 1 \end{bmatrix}.$

An inverse is defined only for non-singular matrices, i.e. matrices with non-zero determinants.

B. Machine Procedure

1. Addition of Matrices:

This can be done on the calculating punches or the accounting machine.

a. Each matrix element a_{ij} is punched on a separate card with identification i and j. Similarly each b_{ij} is punched on a separate card. Cards are then arranged in sequence of i, j in such a way that corresponding a_{ij} and b_{ij} cards follow each other (sorter, collator). The operation $a_{ij} + b_{ij} = c_{ij}$

can then be performed on any of the calculating punches. The result c_{ij} can be punched on the b_{ij} cards, or, alternately, trailer cards can be merged behind each pair of a_{ij} and b_{ij} cards, and c_{ij} can be punched on the corresponding trailer card. If the operation is performed on an accounting machine the sum $a_{ij} + b_{ij}$ can be printed or summary punched on both. The method can readily be extended to the addition of several matrices,

$$A + B + \ldots + H = (a_{ij} + b_{ij} + \ldots + h_{ij}) .$$

At times it may be convenient to punch one pair of elements a_{ij} and the corresponding b_{ij} on the same card prior to their addition.

b. Several elements a_{ij} are punched on one card. If the order of the matrix is not too large, an entire row (or column) of the matrix can be punched on one card. In general, the number of elements to be punched on a single card will depend on the capacity of the machine to be used for the addition as well as the order of the matrix and the size of its elements. For maximum efficiency, the number of elements punched on each card should be such that they can all be summed in one run without any change in the control panel wiring.

2. Scalar Multiplication:
This is an extremely simple operation that can be performed on any calculating punch. The scalar c can be stored from a master card, and used as a group factor on every matrix element a_{ij} . The only consideration is to determine how many such multiplications can be performed per card. This again will depend on the machine, the order of the matrix and the size of the a_{ij}'s. If only one matrix is involved, or if the scalar c is the same for all matrices, c can be digit emitted and does not have to be read into the machine on a master card. Incidentally, it is implicitly assumed throughout this discussion that the various calculations are performed simultaneously on several sets of different matrices. This, of course, will require a further code identifying each card as belonging to a given set.

3. Multiplication of Matrices:

I. a. Each element a_{ij} is punched on a separate card with identification i and j. Similarly each b_{ij} is punched on a separate card. We then proceed to reproduce the a_{ij}'s (in sequence of rows) into the b_{ij}'s (in sequence of columns). Since each row of the A matrix has to be multiplied by each column of the B matrix we need as many duplicates of the B matrix as we have rows in the A matrix. Thus any given row of the A matrix will successively be paired off, i. e. reproduced on every column of the B matrix. Each card will contain a pair of elements a_{ik} and b_{kj} . Trailer cards (one for each c_{ij}) are then inserted behind each group of cards containing elements of one row of A and one column of B. The product elements

$c_{ij} = \Sigma\ a_{ik} b_{kj}$ are obtained on any calculating punch, and are punched into the trailer cards.

b. Several elements b_{ij} are punched on one card. If the order of the matrix is not too large,an entire row of the B matrix can be punched on one card. Then each row of the A matrix has to be reproduced only once into a given set of B cards. This requires as many duplicates of the B matrix as we have rows in A so that any given row of A will appear paired off with a set of B cards containing all the corresponding column elements. Then, upon merging of trailer cards, several $\Sigma a_{ik}\ b_{kj}$ (for different b's) can be obtained simultaneously on the calculating punch. It is desirable to select the number of b_{ij}'s punched per card in such a way that all the sums of products can be obtained in one run without any change in the control panel wiring. A combination of methods I. a. and I. b. is often indicated. This is, in general, the most efficient manner for matrix multiplication. On the other hand punching as many b_{ij}'s on one card as the card will hold, will reduce the number of cards, but will require a rewiring of reading and punching fields of the control panel.

If the order of the matrices is sufficiently small we can not only punch an entire row of the B matrix per card but also reproduce all rows of the A matrix into corresponding cards of the B matrix (in sequence of columns). Then all c_{ij}'s can be obtained on a single trailer card. In the extreme case (e.g.,matrices of order 2) all elements of both matrices can be punched on one card.

Method I. b. essentially suggests to pair off one row of the A matrix with several columns of the B matrix. Conversely, we can reproduce several rows of the A matrix into cards containing corresponding elements of a single column of the B matrix. Which one of the two procedures is adopted depends on the rectangular shape of the matrices. For example, for an A matrix with more columns than rows, i.e., long rows and short columns, and a B matrix with more rows than columns, it will be preferable to pair off one row of A with several columns of B.

Finally, it should be remarked that it is possible to obtain sums of products on an accounting machine by means of "progressive digiting". This method can be useful if the matrix elements are of small size.

II. a. and b. This is essentially the same method as I.a. and I.b. Instead of reproducing elements a_{ij} of A cards into B cards, A cards can be interspersed with B cards on the collator. The factors a_{ij} and b_{ij} are then read from alternate cards into the calculating punch.

In order to avoid duplicating B cards, (i.e., use only one set of A cards and B cards), it may at times be convenient to do several runs on the calculating punch, and to follow each operation on the calculating punch by a collator run; each time the A cards and B cards are interspersed in a different manner until each row of A has been paired off with each column of B.

PART 2

IBM Type 604 Electronic Calculating Punch
and
Type 602-A Calculating Punch

NOTES ON THE IBM TYPE 604 ELECTRONIC CALCULATING PUNCH AND 602-A CALCULATING PUNCH

Eric V. Hankam

Watson Scientific Computing Laboratory

SYMBOLIC NOTATION

P_i	=	ith program step
P_o	=	read cycle
R	=	read ; read cycle
P	=	punch ; punch cycle
RI	=	read into
RO	=	read out
RE	=	reset
RR	=	read out and reset
→	=	read into
⇒	=	yields (implies)
ϵ	=	in (belongs to)
MP	=	multiplier
MC	=	multiplicand
MPL	=	multiply
DIV	=	divide
N	=	normal
T	=	transferred
X-p.u.	=	X-pickup
D-p.u.	=	digit pickup
I-p.u.	=	immediate pickup
Pil-Sel	=	pilot selector

604

F	=	factor storage
G	=	general storage
MQ	=	multiplier-quotient storage
EC	=	electronic counter
Ri	=	read units into
Ro	=	read units out of
Em	=	emitter control
B. T.	=	balance test for step suppression
Sup	=	suppress
Sup +	=	suppress on plus balance
Sup -	=	suppress on minus balance
C. C.	=	card cycle
Pil-Sel	=	pilot selector
Pch-Sel	=	punch selector
Cal-Sel	=	calculator selector

602-A

[ ]	=	counter
( )	=	storage unit
Co-Sel	=	co-selector
B. T.	=	balance test

a) 604

Multiplication:

In multiplication, the MC cannot be stored in either EC or MQ; i.e., a factor that has been developed in EC must be transferred to another storage unit before multiplication by it can be started. It is not possible to square a number by reading it out (as MC) from MQ. Similar statements hold for division. It is possible, however, to read a single-digit MC directly from the emitter control and impulse MPL at the same step. It is not possible to shift during multiplication. $\left\{\begin{matrix}\text{Multiplication}\\ \text{Division}\end{matrix}\right\}$ is performed internally by over and over $\left\{\begin{matrix}\text{addition}\\ \text{subtraction}\end{matrix}\right\}$.

Pilot Selectors and Coupling Exits:

The I-p.u. and coupling exit of a P-Sel are common hubs. (They can be regarded as either outlets or inlets).

Unfinshed Program:

An "unfinished program" should be wired to "Stop 1st Reading" at all times as a precautionary measure. If there has been an unfinished program condition, punching is suppressed from the calculating units, but the gang punching is not affected.

Storage Assignments:

It is not possible to select storage assignments during calculate time on a program step.

By an 8-6 assignment any 3-position storage unit can be associated with any 5-position storage unit. An 8-6 or 6-4 assignment disassociates the storage unit involved from its sign control unit. If used independently (i.e., without an associated storage unit) to RI or RO a number the sign will not travel with it.

Storage Units:

All storage units except EC read true figures with proper sign at all times (internally). In EC a true figure is represented by a complement and conversely. Avoiding overlap, two distinct numbers can be stored in EC provided that their signs are alike. It is possible to read out from one storage unit into several other storage units on the same program step. If a positive number is transferred from one storage unit into another storage unit and EC on the same step, it will enter the storage unit as a complement. This can be used as a method for conversion of numbers to complements. A negative number, however, will read into the storage unit and EC as a negative number. Two distinct numbers with any possible signs can be read into a single storage unit (as positive numbers) at read time (avoiding overlap). Their signs can be "stored" by means of Cal-Sel.

It is possible to read from any general storage unit,MQ,or EC into F at$\left\{\begin{matrix}\text{read}\\ \text{punch}\end{matrix}\right\}$time by wiring all positions of, say, G2→F2 on the double-panel board. G2 RO and F2 RI are wired from C.C. This may be particularly useful on machines without factor storage read-in during calculate time.

Impulsing Storage Units:

To RI from an X-card only, a storage unit can be impulsed directly by the X from the reading brushes. In this case, column splits must be used for digits over-punched with an X. An alternate method that does not require the use of column splits, is to energize the I-p.u. of a Pil-Sel with the X, and to take a C.C. pulse through the T-side of the selector to the storage RI hubs. A C.C. pulse should never be taken through the N-side of a selector (energized by the I-p.u.) because the early part of the C.C. pulse would get through the selector every time before it is picked up (unless a Y-punch is used to energize the Pil-Sel).

Storage RO hubs (2-panel board) accept impulse for the same length of time as C.C. (12.7 to 9.7). They must receive an impulse starting before X-time and extending until past 9-time.

It is sufficient to impulse any storage RI with an X-pulse (11 to 11.3 time) even though the storage RI hubs are inlets from 14.9 to 11.7 time. But with the exception of the units position, the storage unit will not register any digit over-punched with an X unless the X is filtered out by a column split.

Emitter Control:

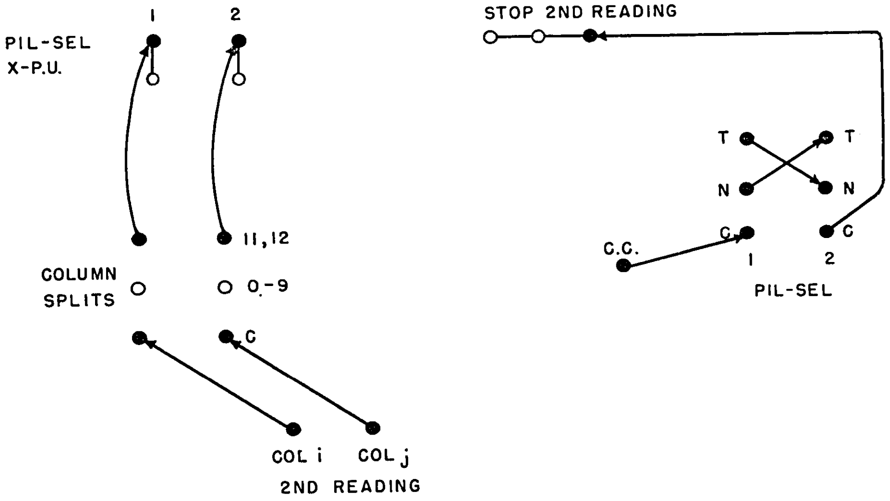
If more than one digit is to be entered into a storage unit by means of the emitter control, the digits must be entered first into EC, since only one digit at a time (on a single program step) can be entered, and a storage unit would reset the i^{th} digit upon entering the $(i+1)^{th}$

Double Punch Blank Column:

To suppress DPBC for cards with an X in column i wire "DPBC" to "Stop 2^{nd} Rdg" through a Pil-Sel controlled by X-p.u. from 2^{nd} reading brushes. The delayed pick-up is necessary because the DPBC hubs emit an impulse on the cycle following the DPBC operation at the second reading station.

Checking:

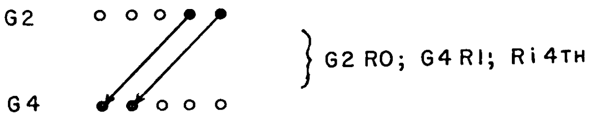
To check that two card columns are both either X-punched or NX-punched, wire:



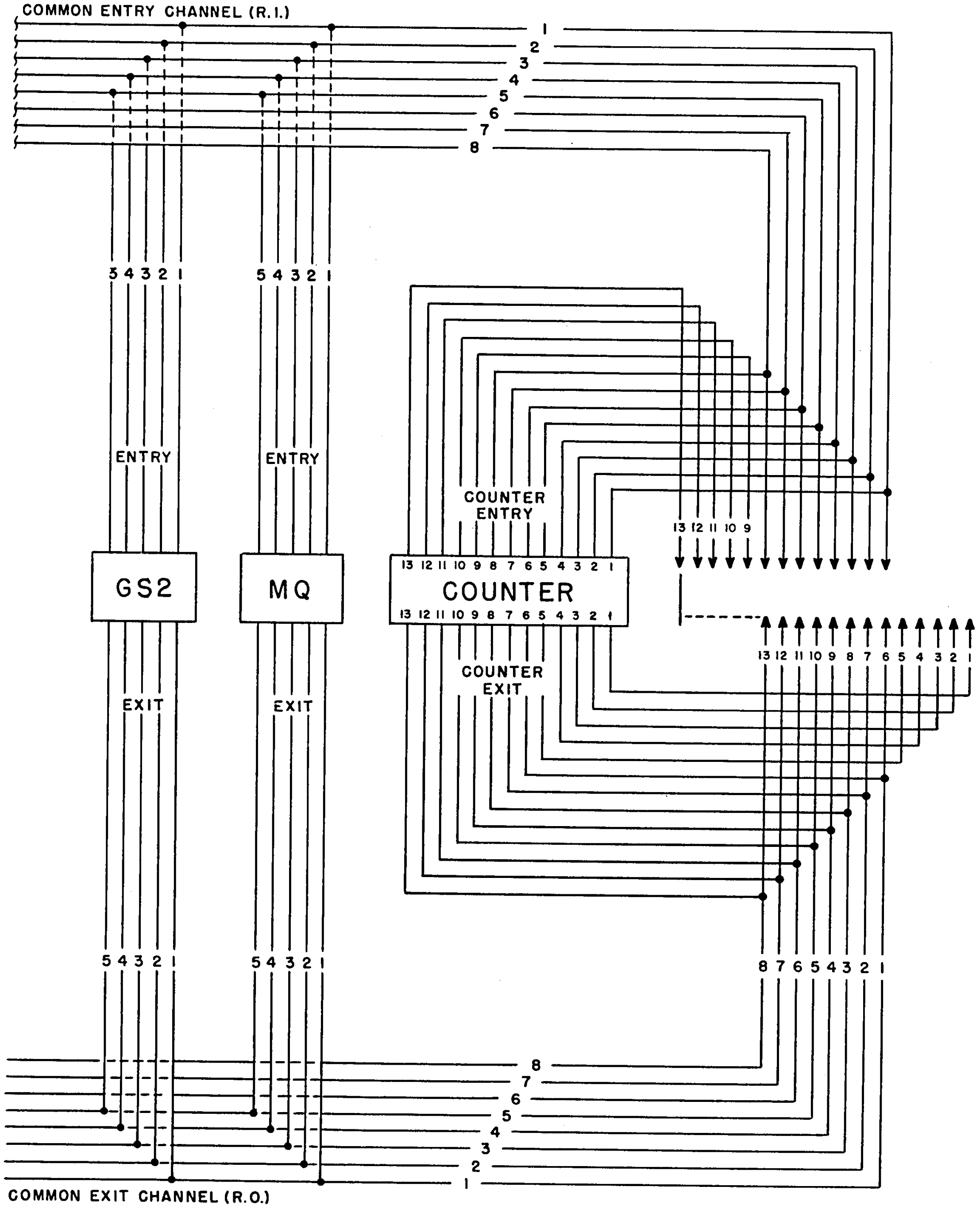
By wiring to "Stop 2nd Reading" the error card will be the last one in the stacker when the machine stops. No error light appears.

Shifting:

It is possible to shift to the "left" directly from one storage unit to another without going through EC; e.g.,



However, it is not feasible to shift into the other direction without going through EC.



Referring to the above figure, in the normal position of the shift unit the first five exit positions of the counter are lost in a transfer out of the counter (i.e., positions 1, 2, . . . , 5.) This position of the shift actually corresponds to Read Units out of 6th. To effect a normal units to units transfer out of the counter it is necessary to shift the units to the farthest position to the left. This will automatically occur when a counter read-out is programmed.

Therefore:

Normal R.O. of counter	=	Ro 6th
Ri 2nd	=	Ro 5th
Ri 3rd	=	Ro 4th
Ri 4th	=	Ro 3rd
Ri 5th	=	Ro 2nd
Ri 6th	=	Ro Normal

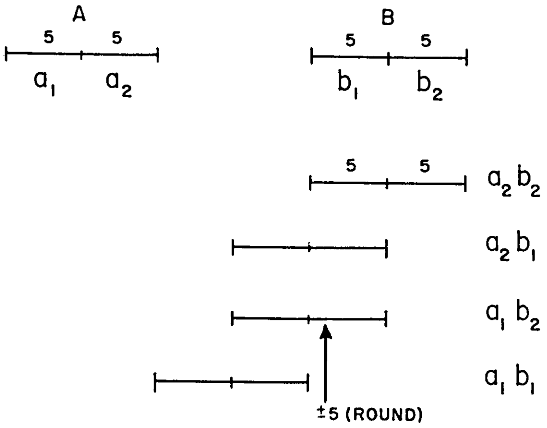
Note: Corresponding hubs are common, i.e., Ri 2nd and Ro 5th are common hubs, etc.

Timing:

Calculate time may last beyond "Y" time; therefore it is seldom safe to use a Y-punch. Since "Y" time precedes card cycle time, it can be used to select the C.C. impulse.

10 × 10 MULTIPLICATION

Examples: 604



R : a_1 → F2

a_2 → MQ

b_1 → F4

b_2 → F1, 3

P_1 : $a_2 b_2$: F1, 3 RO ; MPL + ⇒ $a_2\ b_2 \in$ EC

P_2 : $a_2 b_2$ →G1, 3: EC RR ; Ro 6th ; G1, 3 RI (highest 6 positions only)

P_3 : $a_2 b_2$→ EC : G1, 3 RO ; EC RI+

P_4 : $a_2 b_1$: F4 RO ; MPL + ⇒ $a_2\ b_2 + a_2\ b_1 \in$ EC

P_5 : a_1→ MQ : F2 RO ; MQ RI

P_6 : $a_1\ b_2$: F1,3 RO ; MPL + ⇒ $a_2\ b_2 + a_2\ b_1 + a_1\ b_2 = a'\ b' \in$ EC

P_7 : round : 1/2 Adj ; Ri 5th x xxxxx | xxxxx
carry ±5

P_8 : $a'\ b'$→G1, 3 : EC RR ; Ro 6th ; G1, 3 RI (highest 6 positions only)

P_9 : $a'\ b'$→EC : G1, 3 RO ; EC RI +

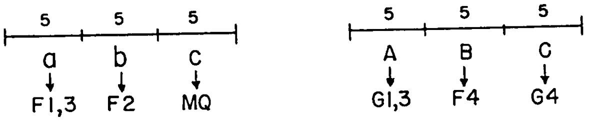
P_{10} : $a_1\ b_1$: F4 RO ; MPL + ⇒ $a_2\ b_2 + a_2\ b_1 + a_1\ b_2 + a_1\ b_1 = AB \in$ EC

P : AB (10 digits) from EC

15 × 15 MULTIPLICATION

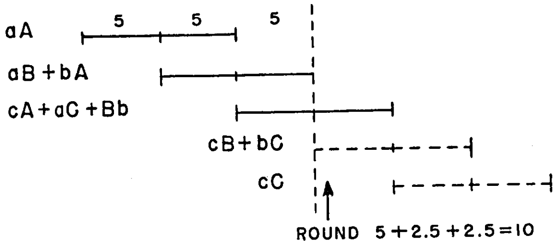
Examples: 604

R:



P_1 :	cA	: G1, 3 RO	; MPL +			⟹ cA ∈ EC
P_2 :	C → MQ	: G4 RO	; MQ RI			
P_3 :	aC	: F1, 3 RO	; MPL +			⟹ cA + aC ∈ EC
P_4 :	B → MQ	: F4 RO	; MQ RI			
P_5 :	Bb	: F2 RO	; MPL +			⟹ cA + aC + Bb = M_1 ∈ EC
P_6 :	round	: ½ Adj	; Ri 5th			
P_7 :		: ½ Adj	; Ri 5th			x xxxxx|xxxxx carry ±1.0
P_8 :	A→G4	: G1, 3 RO	; G4 RI			
P_9 :	M_1→G1, 3	: EC RR	; Ro 6th	; G1, 3 RI		(highest 6 digits only)
P_{10} :	M_1→ EC	: G1, 3 RO.	; EC RI +			
P_{11}:	Ba	: F1, 3 RO	; MPL +			⟹ M_1 + Ba ∈ EC
P_{12}:	A→MQ	: G4 RO	; MQ RI			
P_{13}:	bA	: F2 RO	; MPL +			⟹ M_1 + Ba + bA = M_2 ∈ EC xxxxxxxxxxx
P_{14}:	M_2→G2	: EC RO	; G2 RI			(last 5 positions)
P_{15}:	M_2→ G1, 3	: EC RR	; Ro 6th	; G1, 3 RI		(highest 6 positions)
P_{16}:	M_2→ EC	: G1, 3 RO	; EC RI +			
P_{17}:	aA	: F1, 3 RO	; MPL +			⟹ highest 10 positions of product ∈ EC

P : $\underbrace{\overset{10}{\text{xxxxxxxxxx}}}_{\text{EC}}$ $\underbrace{\overset{5}{\text{xxxxx}}}_{\text{G2}}$

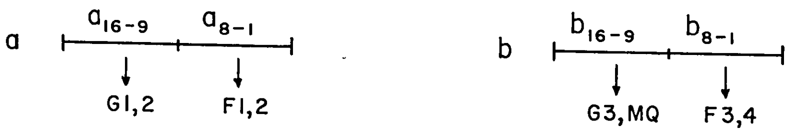


Note: Adjustment of 2.5 for each of the two omitted products cB and bC is necessary.

16 ± 16

Examples: 604

R :



P_1	: $a_{16-9} \rightarrow$ EC	: G1, 2 RO	; EC RI +		
P_2	: $b_{16-9} \rightarrow$ EC	: G3, MQ RO	; EC RI ±		$\Rightarrow$ a ± b_{17-9} ϵ EC
P_3	: a + $b_{16-9} \rightarrow$ G1, 2	: EC RO	; G1, 2 RI		
P_4	: a + $b_{17} \rightarrow$ G3	: EC RO	; Ro 4th	; G3 RI	
P_5	: preserve sign of G3	: EC RO	; Ro 4th	; MQ RI	
P_6	: balance test	: B. T.	; G4 RI		(to clear G4 for suppressed case)
Sup+P_7	: +1 $\rightarrow$ G4 (if < o)	: Em "1"	; G4 RI		$\Rightarrow \left\{ {0 \atop 1} \right\} \epsilon$ G4
P_8	: -1 $\rightarrow$ EC	: Em "1"	; EC RI−		$\Rightarrow$ a + b_{17-9}−1 ϵ EC
P_9	: balance test	: B. T.			
P_{10}	: clear EC	: EC RR			
Sup+P_{11}	: -1 $\rightarrow$ EC	: Em "1"	; EC RI−		$\Rightarrow \left\{ {0 \atop -1} \right\} \epsilon$ EC
P_{12}	: $\left\{ {0 \atop 1} \right\}_{G4} \rightarrow$ EC	: G4 RO	; EC RI +		$\Rightarrow \left\{ {0 \atop -1} \right\} \epsilon$ EC [−1 ϵ E only if a + b_{17-9} = 0]
P_{13}	: balance test	: B. T.			
P_{14}	: clear EC	: EC RR			[note : if < 0, a + b_{17-9} = 0]
P_{15}	: a + $b_{17} \rightarrow$ EC	: G3, MQ RO	; EC RI +	; Ri 4th	
P_{16}	: clear G3	: G3 RI			
P_{17}	: cancel extraneous MQ value	: G3, MQ RO	; EC RI−	; Ri 4th	x 00000000 ϵ EC
P_{18}	: a + $b_{16-9} \rightarrow$ EC	: G1, 2 RO	; EC RI +		$\Rightarrow$ a + b_{17-9} ϵ EC
Sup−P_{19}	: "boost" signed quantity	: $\frac{1}{2}$ Adj			x $\underset{\pm 5}{\text{xxxxxxxx}}$

P_{20} : $a + b_{17-13} \rightarrow$ MQ : EC RO ; Ro 5th ; MQ RI

P_{21} : $a + b_{17-13} \rightarrow$ EC : G3, MQ RO ; EC RI - ; Ri 5th subtract out $\underbrace{00000}$ $\underbrace{xxxx}_{a+b_{12-9}}$

P_{22} : $a + b_{12-9} \rightarrow$ Gl, 2 : EC RR ; Gl, 2 RI

P_{23} : $a + b_{12-9} \rightarrow$ EC : Gl, 2 RO ; EC RI + ; Ri 4th

P_{24} : $a + b_{12-9} \rightarrow$ Gl, 2 : EC RR ; Gl, 2 RI

P_{25} : $a + b_{12-9} \rightarrow$ EC : Gl, 2 RO ; EC RI + ; Ri 6th xxxx 00000000 ϵ EC

P_{26} : $a_{8-1} \rightarrow$ EC : Fl, 2 RO ; EC RI +

P_{27} : $b_{8-1} \rightarrow$ EC : F3, 4 RO ; EC RI ± $\Rightarrow a + b_{13-1}$ x xxxx xxxxxxxx ϵ EC
carry

P_{28} : $a + b_{5-1} \rightarrow$ G4 : EC RO ; G4 RI (for punching $a + b_{5-1}$ from G4 with proper sign)

P_{29} : $a + b_{13-6} \rightarrow$ Gl, 2 : EC RR ; Ro 6th ; Gl, 2 RI

P_{30} : $a + b_{13-6} \rightarrow$ EC : Gl, 2 RO ; EC RI − xxxxx xxx
±5 (from P_{19})

Sup-P_{31} : cancel ±5 : $\frac{1}{2}$ Adj ; Ri 4th ∓5

P_{32} : $\left.\begin{matrix} \\ \\ \end{matrix}\right\}$ $a + b_{13-6} \rightarrow$ EC : Gl, 2 RO ; EC RI +

P_{33} : : Gl, 2 RO ; EC RI + $\Rightarrow a + b_{13-6}$ ϵ EC xxxxxxxx

P_{34} : $a + b_{17-13} \rightarrow$ Gl, 2 : G3, MQ RO ; Gl, 2 RI ; Ri 4th

P_{35} : $a + b_{17-13} \rightarrow$ EC : Gl, 2 RO ; EC RI + ; Ri 5th $\Rightarrow a + b_{17-6}$ ϵ EC $\overbrace{xxxxxxxxxxxx}^{12}$

P : $a \pm b_{17-6}$ from EC ; $a \pm b_{5-1}$ from G4 (use sign of G4)

Remarks:

1) P_5 serves to " store " the sign of $a + b_{17}$ (in G3) in MQ for the transfer of $a + b_{17}$ at P_{15}. The extraneous value in MQ is subtracted out at P_{17}. G3 is cleared at P_{16} and, although "coupled" to the MQ RO, does not interfere with any subsequent MQ RO.

2) P_{19} is required to increase the number in EC sufficiently so that $a + b_{5-1}$ will have the proper sign when it is read into G4 at P_{28}. P_{31} subtracts out the ±5 added at P_{19} .

(continued on next page)

3) $a_{16-9} + b_{16-9} = 0$ is treated as a special case. P_{19} and P_{31} are suppressed whenever $a_{16-9} + b_{16-9} = a + b_{16-9} = 0$.

4) To test whether a number = 0 in EC:
B. T. and if < 0, emit 1 into a storage unit, say G4.

EC	B.T.	G4
> 0	} +	0
0		
< 0	–	+1

Subtract –1 to the number in EC .

B. T. ; clear EC ; and if <0, emit -1 into EC.

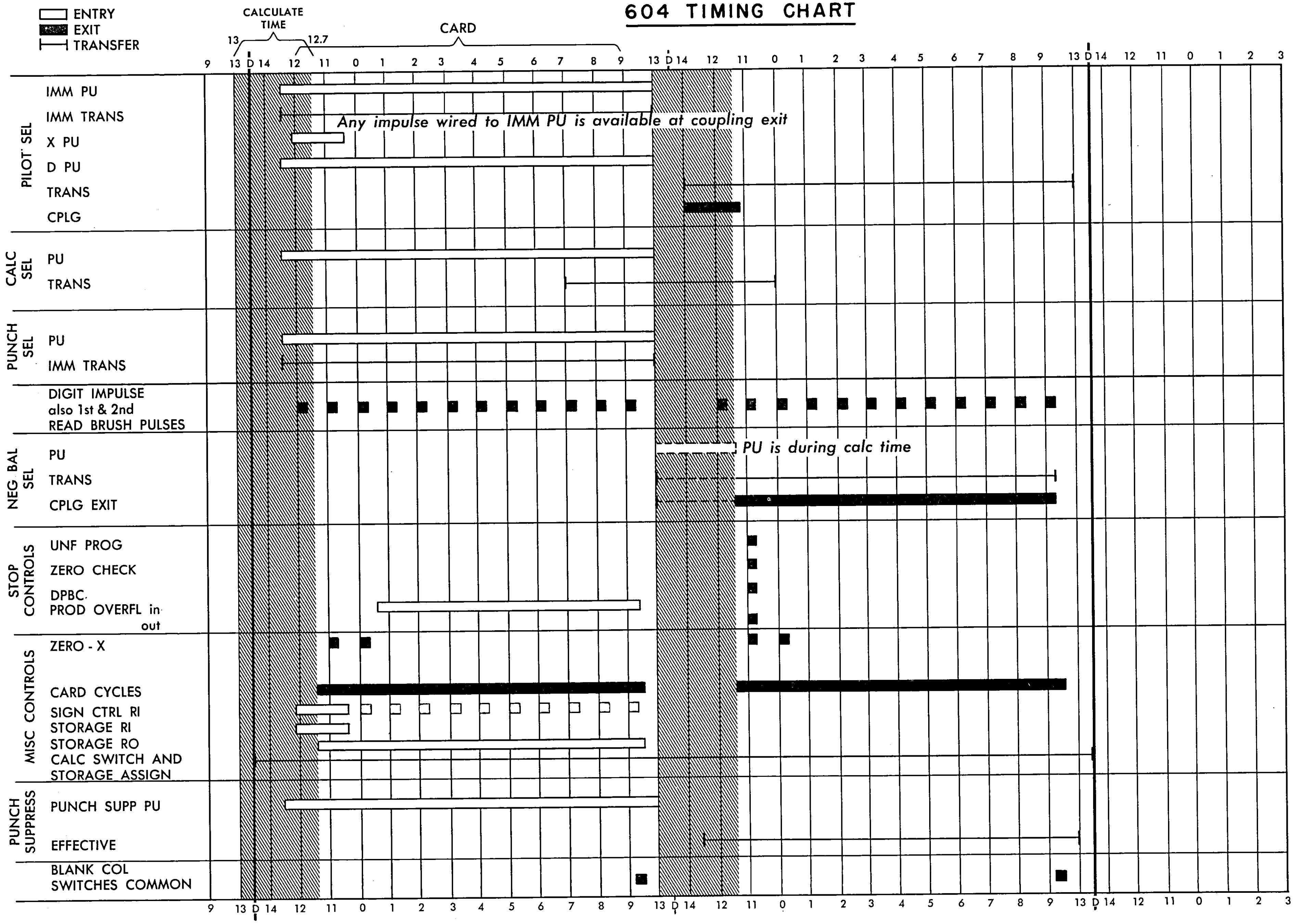
EC	BT	EC
>0	} +	0
0		
<0	–	-1

Read G4 → EC:

		G4		EC		
>0 case	:	0	+	0	=	0
0 case	:	0	–	1	=	-1
<0 case	:	+1	–	1	=	0

B. T. again : Only zero case will give negative balance. (The zero case is characterized by + balance after first and – balance after second B. T.).

604 TIMING CHART



b) 602-A

Multiplication:

It is not possible to perform two multiplications (or two divisions, or one multiplication and one division) on successive program steps. At least one program step must intervene between the multiplications.

When squaring a number it is possible to put the number in (1R) and use it both as a MP and MC;

It is possible to wire a constant MC directly from a digit emitter on the step at which multiplication is performed.

Depending on the digits of the MP, several cycles are required during a multiplication (similarly division). The first multiplication cycle occurs during the program step on which multiplication is impulsed. That program step can be regarded as "repeated" for each of the remaining multiplication cycles.

A Co-Sel (or Pil-Sel; I-p. u.) picked up by a program couple or program exit (of the same step on which MPL is impulsed) will remain transferred throughout the entire multiplication.

If the MC is in a counter, do not reset the counter during the multiplication step. It can be reset on the step following the multiplication.

A counter should not be reset on a program step during which multiplication takes place, even though the counter is not involved in the multiplication.

Couple adjacent counters for the development of a product or quotient.

The digit emitter is active on each multiplication cycle; it is inactive for a zero multiplier.

Therefore a Pil-Sel energized by a digit on a program step on which a multiplication is performed will pick up on the second multiplication cycle, but will not pick up at all if the multiplier is zero. A Pil-Sel needed for the extension of program steps should not be picked up by digit impulses at P_{12} if a multiplication takes place at P_{12} .

Balance Test:

Once a number is punched converted as a result of a B. T., all following results forthe same card will also be converted until a new B. T. takes place.

A B. T. unit can be used more than once by split wiring.

More than one B. T. can be made by two distinct numbers being punched from the same storage unit on the same program step.

Zero (in a counter) is regarded as a positive quantity during a balance test operation on the 602, 602-A and 604.

Punch Interlock:

Wiring P_1 to "punch interlock" will hold up P_1 until the previous card has been completely punched.

Punch Interlock cannot be used to delay a program step until punching (impulsed on a previous program step) on the same card has been completed. It also cannot delay the read cycle of the following card.

Punch Emitter:

Do not split wire two hubs from the (external) punch emitter into a single punching position.

The punch emitter is not timed; it emits as soon as the card moves to the particular punching position.

Counter:

A true figure appears in a counter (internally) as a 9's complement. A counter filled with 9's is treated as zero. (holds for 602 and 602-A).

All counters can be coupled together into a 40-position counter group in which 40-digit numbers can be accumulated.

It is possible to accumulate the product of two numbers in two different counter groups at the same time. In the case mentioned in paragraph 2 under "Multiplication", a^2 was computed, and $+a^2$ was accumulated in one counter while $-a^2$ was developed in another counter.

Card Feed Operation:

There are two cycles to a card feed. A card passes the reading brushes on the first one, the read cycle, andthe card behind it passes the control brushes on the second one, which is normally program 1. Thus,it is not possible for cards to pass the reading brushes on successive cycles. If all program steps are skipped for a card ("Read Cycle" wired to "Read" hub, usually through a selector) the read cycle hubs will become active again on the cycle after that card passes the reading brushes. However,the following card does not pass the reading brushes on this cycle; it is passing the control brushes on the second half of the previous card feed operation.

Transfer of Numbers:

A number must not be transferred from (6), (7) or (8) on the same step when "punching" is impulsed from these units. Transferring a number while it is punched should also be avoided. It may cause the occasional skipping of a column to be punched.

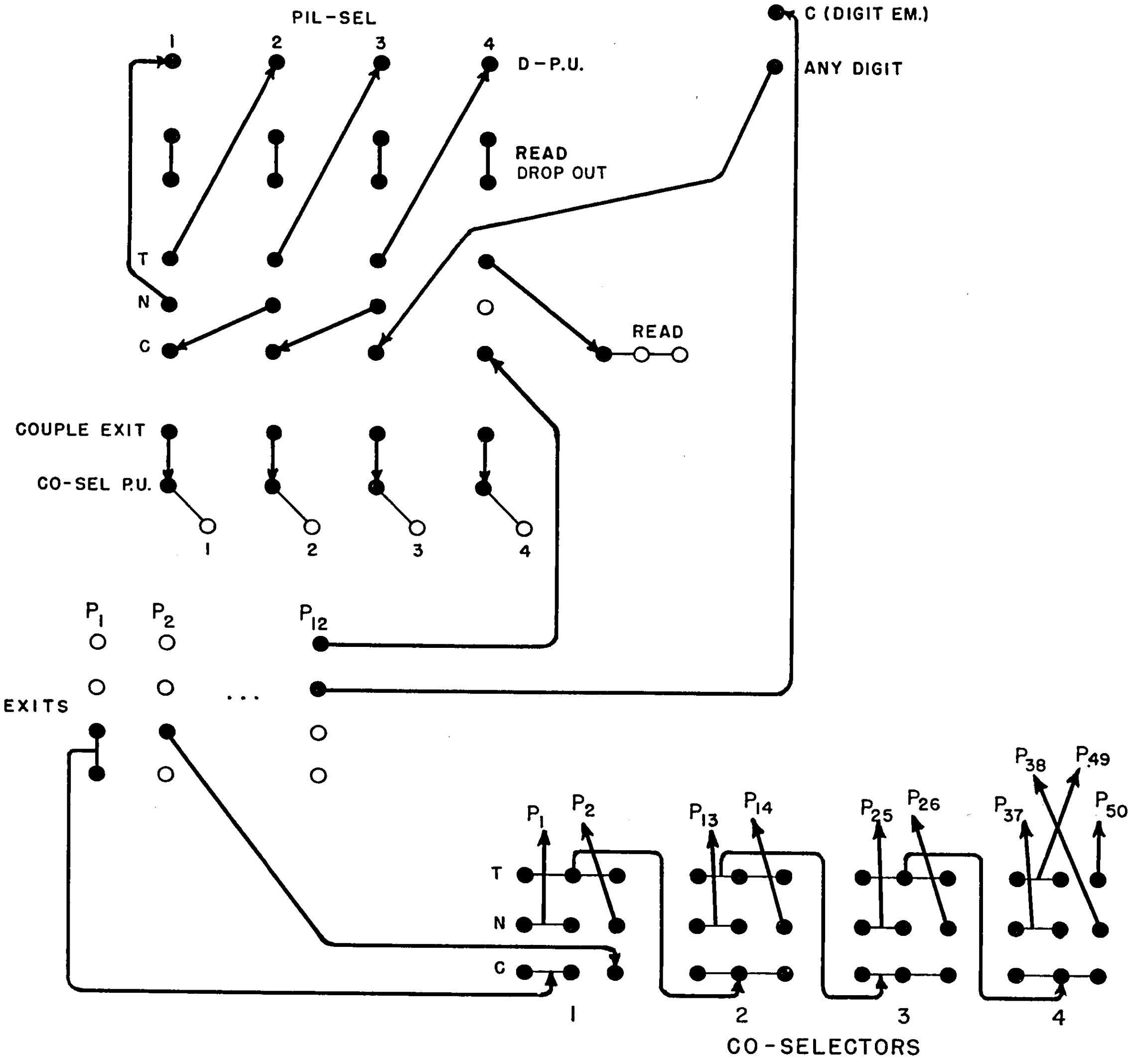
It is not possible to transfer numbers during multiplication (or division).

Co-Selectors:

To pick up the same Co-Sel at several (different) program steps, impulse the Co-Sel pickup hubs directly from the program exits.

Program Steps:

Method to extend program steps beyond 12:

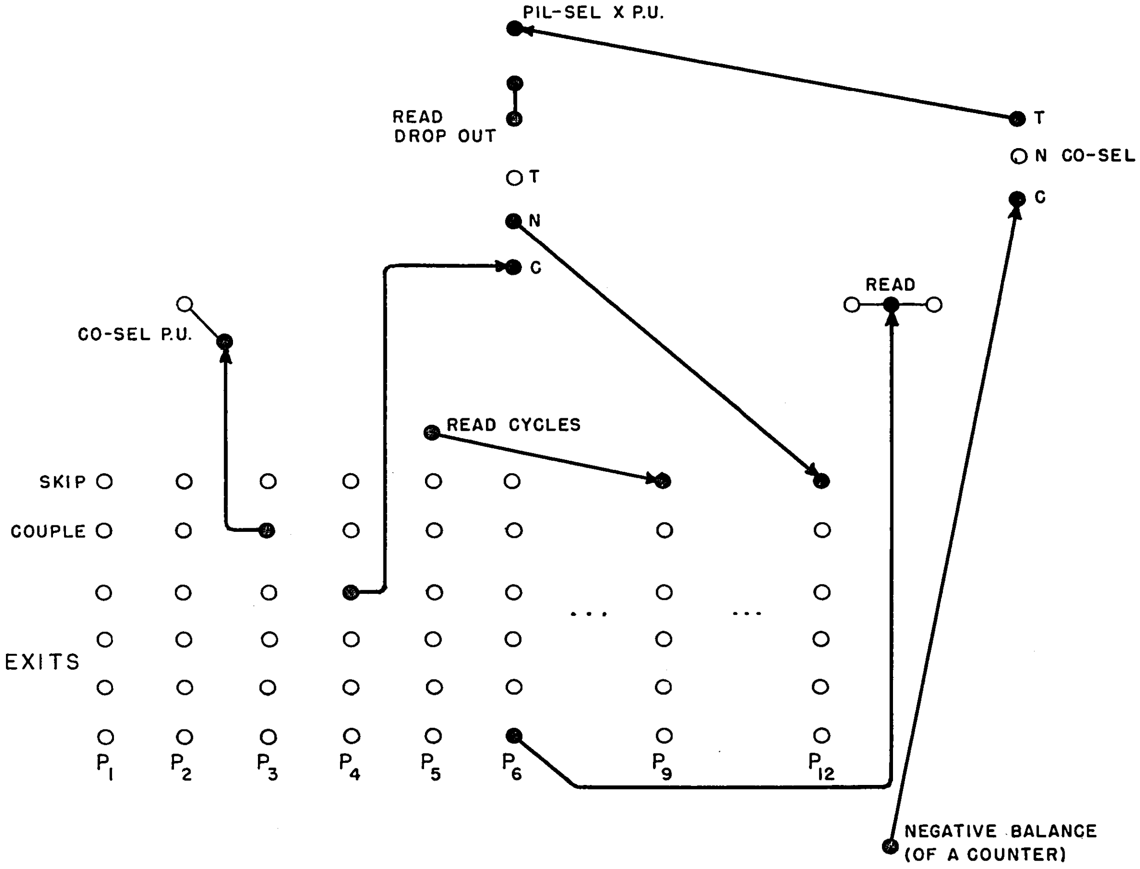


Pil-Sel.1 and Co-Sel 1 will be picked up from P_{13} until Read time.
" 2 " " 2 " " " " " P_{25} " " "
" 3 " " 3 " " " " " P_{37} " " "
" 4 " " 4 " " " " " P_{49} " " "

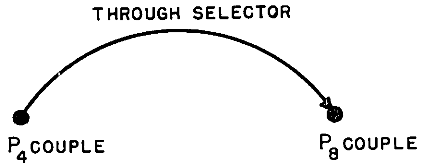
P_{60} will get through Pil-Sel 4 to Read.

Usually several Co-Sel will be coupled to form a sufficiently large group to allow taking through all necessary program exits.

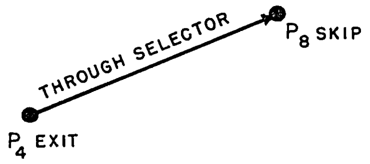
Now suppose we want to start with program steps 9, 10, 11 (following a read cycle); then repeat steps 12, 1, 2, 3, 4; 12, 1, 2, 3, 4; etc., until a negative balance condition arises (during one of these steps, say step 3); and finally end up with steps 5, 6 before going back to Read. A wiring scheme for this is as follows:



To skip program steps in the "middle" (i.e., between 1 and 12), wire:



OR



P_5 , P_6 , P_7 will be skipped.

P_4 and P_8 will occur at the same time.

P_5 , P_6 , P_7 will be skipped.

P_4 and P_8 will not occur at the same time.

Read Drop Out Impulses:

The R.D.O. (Read Drop Out) impulses are common hubs, and may cause back circuits if two Pil-Sel are dropped out in a different manner. For example, if both Pil-Sel are dropped out with a R.D.O. impulse, and if one of the two is also dropped out at the end of a program step, the R.D.O. impulse to the other Pil-Sel should be wired through the transferred side of a Co-Sel picked up at Read time.

Shifting:

Impulsing MPL and TENTH hubs on the same step will cause the product to be shifted one position to the right.

Skipping:

The SKIP OUT hubs accept impulses only during a read cycle. To skip a card on a condition arising during calculation, wire the first position to be punched from the storage exits, and the SKIP pulse through either the N- or T-side of a Pil-Sel (which reflects the condition) whose common point is wired to the punch magnets. If the SKIP pulse is wired through the N-side of the Pil-Sel, it should be taken through a Co-Sel (picked up after the condition arises) to prevent it from getting through before the Pil-Sel is picked up.

The Pil-Sel will usually be dropped out by a "punch drop out" impulse, and "punch interlock" should be impulsed so that the Pil-Sel cannot be activated before punching of the preceding card has been completed.

Also it should be mentioned that the "P_1-Skip" hub is dead.

Division:

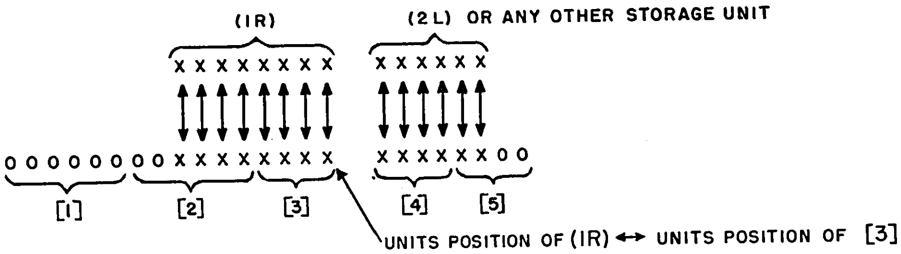
It is possible to perform an operation of the type $\left(\frac{A}{B}\right)C$ on the same step. In addition to the standard division wiring, C is wired into a counter which will develop $\left(\frac{A}{B}\right)C$. MPL must not be impulsed.

A zero divisor will cause the machine to punch zeros for the quotient.

If the difference of significant digits of the dividend and divisor exceeds 7, the machine will punch an incorrect answer after very lengthy calculation.

The highest position of [1] must remain free in division; [1] must always be used as the dividend counter. Adjacent counters can be coupled for placement of a larger dividend; usually [1, 2, 3]. At least one significant digit must appear in (1R), the divisor unit. It is possible to use a divisor with more than 8 digits by placing the low order digits into any storage unit; the dividend can also be extended beyond 15 digits by coupling counters adjacent to [1, 2, 3].

In any division problem there exists a one-to-one correspondence between the units position of (1R) and the units position of [3] in the sense that any significant digit standing in the units position of (1R) must be wired into the units position of [3] during division. Digits located n places to the $\left\{\begin{matrix}\text{left}\\\text{right}\end{matrix}\right\}$ of the units position of (1R) must be wired n places to the $\left\{\begin{matrix}\text{left}\\ \text{right}\end{matrix}\right\}$ of the units position of [3]. For example, consider a 14-digit divisor; it must be read into [1, 2, 3, 4, 5] in the following manner:


(1R)
(2L) OR ANY OTHER STORAGE UNIT
X X X X X X X X
X X X X X X
O O O O O O O O X X X X X X X X
X X X X X X O O
[1]
[2]
[3]
[4]
[5]
UNITS POSITION OF (1R) ↔ UNITS POSITION OF [3]

Thus it can be seen that for an 8-digit divisor located entirely in (1R), as in standard division, [1, 2, 3] suffice as the dividend unit; and for a 4-digit divisor entered into the highest 4 positions of (1R), only [1, 2] need to be coupled as the dividend unit.

The number of quotient digits (≤ 8) developed depends on the placement of the dividend in the dividend counter group, as well as on the original placement of the divisor into (1R).

Rounding the Quotient in Division:[a]

In order to avoid the loss of the quotient position which is used to receive the "reset to 5" and then dropped out, it is possible to round the quotient in the following way and not sacrifice any positions. (This is, in fact, the method of the 602). On the program just preceding the "divide" program, wire the four program impulses to 1) storage 1 read-out, 2) × 5, 3) tenths, 4) dividend counter read-in plus. This will cause half the divisor to go into the dividend counter before dividing starts, and will thereby produce a rounded quotient directly.

Division with a Negative Dividend:[a]

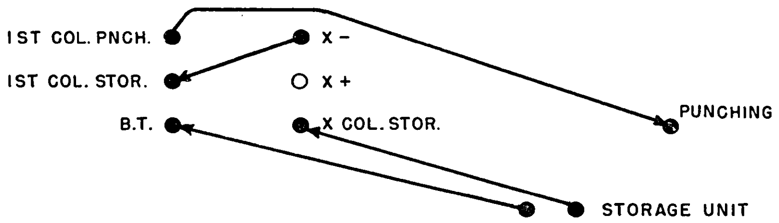
If one uses the same method of rounding recommended in the above paragraph, and if one has satisfied the rules about not having a divisor so small that seven shifts will be unable to test the left end figure of the dividend, then the quotient counter receives the correct negative result and this is converted to a true figure during the punching. Notice that the machine is building the quotient by working the dividend counter <u>up</u> to zero from its original negative number and not <u>down</u> to zero from a positive dividend. Nevertheless,

[a] This process was suggested by Dr. Paul Herget, University of Cincinnati Observatory, Cincinnati, Ohio.

the machine always leaves a positive remainder. For these reasons,one should add 0.5 × divisor just before the division takes place.

Punching:

To punch an X (during a B.T. operation) over the highest position of the result, wire:



If two numbers are to be punched successively (on different program steps) from the same storage unit, an internal interlock is provided to hold the second number until completion of punching of the first. For that reason the units position of (6), (7) or (8) has to be wired to be punched. It is possible to read the first result into, say, (8) at P_i ; impulse it to be punched at P_{i+j} (where j may be 0); read a second result into (8) at P_{i+j+k} (where k ≠ 0); and punch the second result at $P_{i+j+k+1}$ (where l may = 0). Trouble may arise, however, in the case when the same result is to be punched twice (into two different card fields) and it is read into the punch storage unit only once. Since there is no second RI between the two punchings, the internal interlock will not be effective, and either a sufficient number of program steps must intervene to allow for the completion of the first punching, or the same number should be read into the punch storage unit (from its previous storage) again for the second punching.

Punching is accomplished by an internal "punch" emitter sending out an impulse which is selectively taken to the punch magnets depending upon the number standing in the storage position.

Punch storage delays cannot affect the occurrence of a read cycle; i.e.,care should be taken in reading from the card directly into punch storage. Nothing will prevent the storage from clearing and reading in a new value before the punching of the previous card has been completed.

A number may be read from one card and punched into succeeding cards (gang punching operations). If the master card is wired to skip out without punching, the impulse to the punch hub must not come before the master card leaves the punch bed. Because of the storage interlock circuits, if this impulse comes before the master card is actually skipped out, an error stop will result. The use of a "Read cycles" impulse will cause this. Moving the punch time back to the first program will clear up this difficulty.

Split wiring any punch storage position, except the units position, with an X from the punch emitter will cause all columns to the left and right of the split wired position (except the units column) to be overpunched with an X.

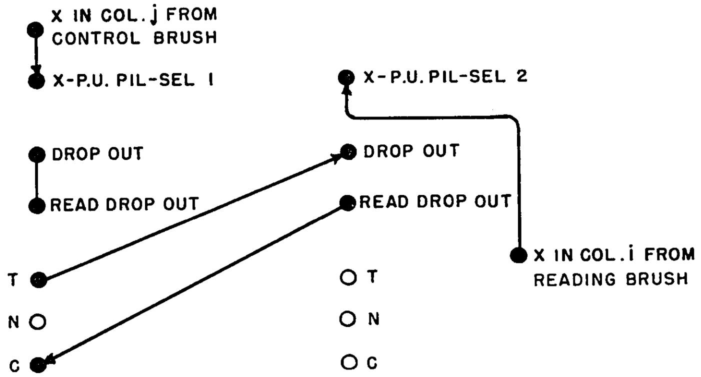
Pilot Selectors:

Even though a Pil-Sel impulsed by an I-p.u. need not be dropped out, it is preferable to do so, because a Pil-Sel can easily remain transferred as a result of a previous control panel (machine run).

Use a "9"-pulse to drop out a Pil-Sel at the end of a program step. Do not use a program exit to drop out a selector.(The drop out impulse should never come between 355° and 50°).

When a Pil-Sel is impulsed by the I-p.u., its couple exit is not active.

To pick up a Pil-Sel with the sign (X in column i) of a group factor appearing on master cards with X in column j, hold the selector throughout the passage of detail cards, and drop out before reading the next master card:



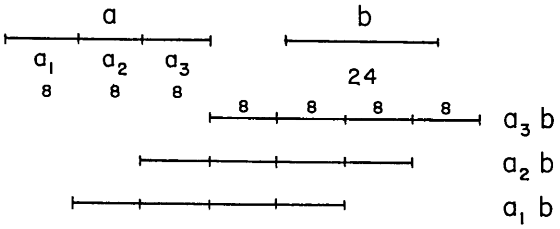
Pil-Sel 1 serves as auxiliary selector.

Pil-Sel 2, if picked up, will remain so from one master card to the next.

24 × 24 MULTIPLICATION

Examples: 602-A

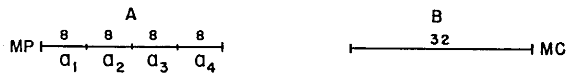
P_0 : a_3 → (1R)
a_2 → (2)
a_1 → (3)
b → (4,5)



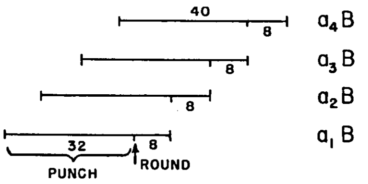
P_1 :	a_3 b	: (4, 5) RO	; [1, 2, . . . , 8] RI +	; MPL
P_2 :	a_2 → (1R)	: (2) RO	; (1R) RI	
P_3 :	a_3 b → (2, 6)	: [1, 2, . . . , 8]RR	; (2, 6) RI	{highest 24 digits only}
P_4 :	a_3 b → [1, 2, . . . , 8]	: (2, 6) RO	; [1, 2, . . . , 8]RI +	{into units position}
P_5 :	a_2b	: (4, 5) RO	; [1, 2, . . . , 8] RI +	; MPL
P_6 :	a_1 → (1R)	: (3) RO	; (1R) RI	
P_7 :	a_1 b	: (4, 5) RO	; [1, 2, . . . , 8] RI +	; MPL
P_8 :	ab → (7, 8)	: [1, 2, . . . , 8]RR	; (7, 8) RI	; (7, 8) P ; READ

Note: b is read from (4, 5) into [1, 2, . . . , 8] into the units position through the N-side of a Pil-Sel at P_1 and P_5 , and into the 9th position through the T-side at P_7 . The Sel is picked up by the P_7 - couple.
[1, 2, . . . , 8] is reset to 5 into the 25th position counting from left, through a Pil-Sel picked up at P_3 .

32 × 32 MULTIPLICATION
Examples: 602-A



P_0 : B → (2,3,4) → lowest 8 positions of (2)
a_4 → (1R)
a_3 → (6) ↗ highest 4 position ↗ highest 4
a_2 → (2L, 5L)
a_1 → (5) ↗ lowest 8



P_1 : a_4B : (2, 3, 4) RO ; [1, 2, . . . , 8] RI + ; MPL ⟹ $a_4B \in [1,2,\dots,8]$ 40 digits
Data card

P_2 : a_3 → (1R) : (6) RO ; (1R) RI

P_3 : a_4 B → (6, 7, 8) : [1, 2, . . . , 8] RR ; (6, 7, 8) RI (highest 32 positions only)

P_4 : a_4 B → [1, 2, . . . , 8] : (6, 7, 8) RO ; [1, 2, . . . , 8] RI + (into units position)

P_5 : a_3 B : (2, 3, 4) RO ; [1, 2, . . . , 8] RI + ; MPL ⟹ $a_4B + a_3B \in [1,2,\dots,8]$ 40 digits

P_6 : a_2 → (1R) : (2L) RO;(5L)RO ; (1R) RI

P_7 : (a_4 + a_3)B→(6,7,8) : [1, 2, . . . , 8] RR ; (6, 7, 8) RI (highest 32 positions only)

P_8 : (a_4 + a_3)B→[1, 2,...,8]: (6, 7, 8) RO ; [1, 2, . . . , 8] RI + (into units position)

P_9 : a_2 B : (2, 3, 4) RO ; [1, 2, . . . , 8] RI + ; MPL ⟹ $a_4B + a_3B + a_2B \in [1,2,\dots,8]$ 40 digits

P_{10} : a_1 → (1R) : (5L) RO ;(5R)RO ; (1R) RI

P_{11} : (a_4 + a_3 + a_2)B → (6, 7, 8) : [1, 2, . . . , 8] RR ; (6, 7, 8) RI (highest 32 positions only)

P_{12} : (a_4 + a_3 + a_2)B→[1, 2,...,8]: (6, 7, 8) RO ; [1, 2, . . . , 8] RI + (into units position)

P_{13} : a_1 B : (2, 3, 4) RO ; [1, 2, . . . , 8] RI + ; MPL ⟹ $(a_4 + a_3 + a_2 + a_1) B = AB \in [1,2,\dots,8]$ 40 digits
READ (Data cards only)

P_1 : AB→(6, 7, 8) : [1, 2, . . . , 8] RR ; (6, 7, 8) RI ; (6, 7, 8) P (highest 32 positions only)
Trailer card READ

Remarks:

a) Due to lack of card space the product is punched into trailer cards inserted alternately behind each data card containing the factors (64 card columns).

b) There can be no carries when combining partial products because the entire MC is always used; e.g., a_4 B + a_3 B = (a_4 + a_3) B constitutes a 16 × 32 = 48 digit multiplication.

(continued on next page)

c) Rounding is accomplished by "resetting to 5" the 8th position (counting from the right) of [1, 2, . . . , 8] through a Co-Sel picked up at P_{11}.

d) The highest 4 positions of (5L) and the highest 4 positions of (2L) must be taken through a Co-Sel picked up at P_6. The lowest two positions of (5L) must be taken through a Co-Sel picked up at P_{10}.

e) (6) RO and (7, 8) RO can be coupled; likewise their RI.

f) (2L) RO must be kept separate from (2R, 3, 4) RO to avoid back circuits; in addition when reading out (6, 7, 8) → [1, 2, . . . , 8] at P_4, P_8 and P_{12}, the highest 4 positions of this 32-digit number (located in (6)) have been taken through a Co-Sel picked up at P_4, P_8, and P_{12}. (For method to pick up the same Co-Sel at more than one program step, see previous section on 602-A Notes, section headed "Co-Selectors".)

g) Since P_1 and P_{13} (data card) are almost identical, only the impulse back to READ has to be wired through a Selector but P_1 for data cards and P_1 for trailer cards are entirely different, and the distinction has to be made through a selector.

$\frac{29}{22}$ = 22 DIVISION

Examples: 602-A

P_0	$: a \rightarrow [1, 2, \ldots, 6]$ $b \rightarrow (1R, 2, 1L)$	$0\overbrace{\text{xx}\ldots\text{x}}^{29}$ $\underbrace{\text{xxxxxxxx}}_{(1R)}\ \underbrace{\text{xxxxxxxxxxxx}}_{(2)}\ \underbrace{\text{xx}}_{(1L)}$ lowest two positions				
P_1	$: \frac{a}{b}$	$: (1R, 2, 1L)$ RO	$; [1, 2, \ldots, 6]$ RI-	$; [7, 8]$ RI +	$;$ DIV $\Rightarrow \left(\frac{a}{b}\right)_{22-15} \in [7, 8]$	first 8 digits of quotient
P_2	$: \left(\frac{a}{b}\right)_{22-15} \rightarrow (8)$	$: [7, 8]$ RR	$; (8)$ RI	$; (8)$ P		(punch first 8 digits of quotient)
	$\text{Rem}_1 \rightarrow (3, 4)$	$: [1, 2, \ldots, 6]$ RR	$; (3, 4)$ RI			(lowest 22 positions of $[1, 2, \ldots, 6]$
P_3	$: \text{Rem}_1 \rightarrow [\dot{1}, 2, \ldots, 6]$	$: (3, 4)$ RO	$; [1, 2, \ldots, 6]$ RI+			(shift 7 places to the left)
P_4	$: \frac{\text{Rem}_1}{b}$	$: (1R, 2, 1L)$RO	$; [1, 2, \ldots, 6]$ RI-	$; [7, 8]$ RI +	$;$ DIV $\Rightarrow \left(\frac{a}{b}\right)_{14-8} \in [7, 8]$	
P_5	$: \left(\frac{a}{b}\right)_{14-8} \rightarrow (8)$	$: [7, 8]$ RR	$; (8)$ RI	$; (8)$ P		(punch second 7 digits of quotient)
	$\text{Rem}_2 \rightarrow (3, 4)$	$: [1, 2, \ldots, 6]$RR	$; (3, 4)$ RI			(lowest 22 positions of $[1, 2, \ldots, 6]$
P_6	$: \text{Rem}_2 \rightarrow [1, 2, \ldots, 6]$	$: (3, 4)$ RO	$; [1, 2, \ldots, 6]$ RI +			(shift 7 places to the left)
P_7	$: \frac{\text{Rem}_2}{b}$	$: (1R, 2, 1L)$RO	$; [1, 2, \ldots, 6]$ RI–	$; [7, 8]$ RI +	$;$ DIV $\Rightarrow \left(\frac{a}{b}\right)_{7-1} \in [7, 8]$	
P_8	$: \left(\frac{a}{b}\right)_{7-1} \rightarrow (8)$	$: [7, 8]$ RR	$; (8)$ RI	$; (8)$ P		(punch third 7 digits of quotient)
	clear Rem_3	$: [1, 2, \ldots, 6]$RR	; READ			

Note: If rounding is to be included, the case of a partial quotient consisting of all 9's must be considered, since in that case a carry should go over into the units position of the preceding partial quotient.

$\frac{29}{22}$ = 29 DIVISION

Examples: 602-A

P_o : a → [1,2,...,6]
b → (1R,2,1L)

$$0\overbrace{xx\ldots x}^{29}$$

$$\underbrace{xxxxxxxx}_{(1R)}\ \underbrace{xxxxxxxxxxxx}_{(2)}\ \underbrace{xx}_{(1L)}\text{ lowest two positions}$$

P_1	: $\frac{a}{b}$	: (1R,2,1L) RO	;[1,2,...,6]RI-	; [7,8] RI +	; DIV $\Rightarrow \left(\frac{a}{b}\right)_{29-22} \in$ [7,8]	first 8 digits of quotient
P_2	: $\left(\frac{a}{b}\right)_{29-22}$ → (8)	: [7,8] RR	; (8) RI	; (8) P		(punch first 8 digits of quotient)
	Rem_1 → (3,4)	:[1,2,...,6]RR	; (3,4) RI			(lowest 22 positions of [1,2,...,6]
P_3	: Rem_1 → [1,2,...,6]	: (3,4) RO	;[1,2,...,6] RI+			(shift 7 places to the left)
P_4	: $\frac{Rem_1}{b}$	: (1R,2,1L)RO	;[1,2,...,6]RI-	; [7,8] RI+	; DIV $\Rightarrow \left(\frac{a}{b}\right)_{21-15} \in$ [7,8]	
P_5	: $\left(\frac{a}{b}\right)_{21-15}$ → (8)	: [7,8] RR	; (8) RI	; (8) P		(punch second 7 digits of quotient)
	Rem_2 →(3,4)	:[1,2,...,6]RR	; (3,4) RI			(lowest 22 positions of [1,2,...,6]
P_6	: Rem_2 →[1,2,...,6]	: (3,4) RO	; [1,2,...,6]RI+			(shift 7 places to the left)
P_7	: $\frac{Rem_2}{b}$	: (1R,2,1L) RO	;[1,2,...,6]RI-	; [7,8] RI+	; DIV $\Rightarrow \left(\frac{a}{b}\right)_{14-8} \in$ [7,8]	
P_8	: $\left(\frac{a}{b}\right)_{14-8}$ → (8)	: [7,8] RR	; (8) RI	; (8) P		(punch third 7 digits of quotient)
	Rem_3→(3,4)	:[1,2,...,6]RR	; (3,4) RI			(lowest 22 positions of [1,2,...,6]
P_9	: Rem_3→[1,2,...,6]	: (3,4) RO	;[1,2,...,6]RI +			(shift 7 places to the left)
P_{10}	: $\frac{Rem_3}{b}$	: (1R,2,1L) RO	;[1,2,...,6] RI-	; [7,8] RI +	; DIV $\Rightarrow \left(\frac{a}{b}\right)_{7-1} \in$ [7,8]	
P_{11}	: $\left(\frac{a}{b}\right)_{7-1}$ → (8)	: [7,8] RR	; (8) RI	; (8) P		(punch fourth 7 digits of quotient)
	clear Rem_4	:[1,2,...,6]RR	; READ			

Note: If rounding is to be included. the case of a partial quotient consisting of all 9's must be considered, since in that case a carry should go over into the units position of the preceding partial quotient.

Remark: In this problem all 80 columns of the card are punched (29+22+29 = 80).

$\Sigma\ x;\ \Sigma\ y;\ \Sigma\ xy;\ \Sigma\ x^2;\ \Sigma\ y^2$

Examples: 602-A

Introductory Description

$\Sigma\ x$ will be developed in [1]

$\Sigma\ y$ " " " " [2]

$\Sigma\ xy$ " " " " [3,4]

$\Sigma\ x^2$ " " " " [5,6]

$\Sigma\ y^2$ " " " " [7,8]

All calculations are to be performed on detail cards.
Each sum is developed progressively in its respective counter group.
The final sums are cleared and punched on a trailer card.
The program step required for reading out the final sums must be skipped for all detail cards. The program steps required for the calculations can be skipped for the trailer card but need not be if the trailer card is blank in the detail card fields containing x and y.

Outline of Program

P_0	: x	[1] and (1R)			
	y	[2] and (2R)			
P_1	: xy	: (2R) RO	; [3,4] RI +	; MPL	$\Longrightarrow \Sigma\ xy\ \epsilon$ [3,4]
	x^2	: (1R) RO	; [5,6] RI +		$\Longrightarrow \Sigma\ x^2\ \epsilon$ [5,6]
P_2	: y→(1R)	: (2R) RO	; (1R) RI		
P_3	: y^2	: (2R) RO	; [7,8] RI +	; MPL	$\Longrightarrow \Sigma\ y^2\ \epsilon$ [7,8]
		READ (detail cards only)			
P_4	: punch	: [1] RR	; [2] RR	; (6) RI	; (6) P
		[3,4] RR	; [5,6] RR	; [7,8] RR	
		(7,8) RI	; (7,8) P	; READ	

8^{th} DIFFERENCES

(11 digits)

Examples: 602-A

P_0	$: \overset{\pm}{f}_i \rightarrow [3,4,5]$	$: [3,4,5]$ RI +			$\Longrightarrow f_i - [f_{i-1} + {}^{1}\Delta^{i-1}_{i-2} + \ldots + {}^{7}\Delta^{i-1}_{i-8}] = {}^{8}\Delta^{i}_{i-8} \in [3,4,5]$
	$\rightarrow [6,8]$	$: [6,8]$ RI +			
P_1	$: {}^{8}\Delta^{i}_{i-8} \rightarrow [1,2]$	$: [3,4,5]$ RR	$; [1,2]$ RI +		$\Longrightarrow {}^{7}\Delta^{i-1}_{i-8} + {}^{8}\Delta^{i}_{i-8} = {}^{7}\Delta^{i}_{i-7} \in [1,2]$
	$\rightarrow (8)$	:	; (8) RI	; (8) P	punch ${}^{8}\Delta^{i}_{i-8}$
P_2	$: {}^{7}\Delta^{i}_{i-7} \rightarrow [3,4,5]$	$: [1,2]$ RO	$; [3,4,5]$ RI-		$\Longrightarrow - {}^{7}\Delta^{i}_{i-7} \in [3,4,5]$
	$\rightarrow (7); (8)$	: (7) RI	; (8) RI	; (8) P	punch ${}^{7}\Delta^{i}_{i-7}$ and store
P_3	$: {}^{6}\Delta^{i-1}_{i-7} \rightarrow [1,2]$	: (6) RO	$; [1,2]$ RI +		$\Longrightarrow {}^{6}\Delta^{i-1}_{i-7} + {}^{7}\Delta^{i}_{i-7} = {}^{6}\Delta^{i}_{i-6} \in [1,2]$
P_4	$: {}^{6}\Delta^{i}_{i-6} \rightarrow [3,4,5]$	$: [1,2]$ RO	$; [3,4,5]$ RI -		$\Longrightarrow - {}^{7}\Delta^{i}_{i-7} - {}^{6}\Delta^{i}_{i-6} \in [3,4,5]$
	$\rightarrow (6);(8)$	: (6) RI	; (8) RI	; (8) P	punch ${}^{6}\Delta^{i}_{i-6}$ and store for next card
P_5	$: {}^{5}\Delta^{i-1}_{i-6} \rightarrow [1,2]$	: (5) RO	$; [1,2]$ RI +		$\Longrightarrow {}^{5}\Delta^{i-1}_{i-6} + {}^{6}\Delta^{i}_{i-6} = {}^{5}\Delta^{i}_{i-5} \in [1,2]$
P_6	$: {}^{5}\Delta^{i}_{i-5} \rightarrow [3,4,5]$	$: [1,2]$ RO	$; [3,4,5]$ RI-		$\Longrightarrow - {}^{7}\Delta^{i}_{i-1} - {}^{6}\Delta^{i}_{i-6} - {}^{5}\Delta^{i}_{i-5} \in [3,4,5]$
	$\rightarrow (5);(8)$	: (5) RI	; (8) RI	; (8) P	punch ${}^{5}\Delta^{i}_{i-5}$ and store for next card
P_7	$: {}^{4}\Delta^{i-1}_{i-5} \rightarrow [1,2]$	: (4) RO	$; [1,2]$ RI +		$\Longrightarrow {}^{4}\Delta^{i-1}_{i-5} + {}^{5}\Delta^{i}_{i-5} = {}^{4}\Delta^{i}_{i-4} \in [1,2]$
P_8	$: {}^{4}\Delta^{i}_{i-4} \rightarrow [3,4,5]$	$: [1,2]$ RO	$; [3,4,5]$ RI -		$\Longrightarrow - {}^{7}\Delta^{i}_{i-7} - \ldots - {}^{4}\Delta^{i}_{i-4} \in [3,4,5]$
	$\rightarrow (4);(8)$	: (4) RI	; (8) RI	; (8) P	punch ${}^{4}\Delta^{i}_{i-4}$ and store for next card
P_9	$: {}^{3}\Delta^{i-1}_{i-4} \rightarrow [1,2]$	: (3) RO	$; [1,2]$ RI +		$\Longrightarrow {}^{3}\Delta^{i-1}_{i-4} + {}^{4}\Delta^{i}_{i-4} = {}^{3}\Delta^{i}_{i-3} \in [1,2]$
P_{10}	$: {}^{3}\Delta^{i}_{i-3} \rightarrow [3,4,5]$	$: [1,2]$ RO	$; [3,4,5]$ RI -		$\Longrightarrow - {}^{7}\Delta^{i}_{i-7} - \cdots - {}^{3}\Delta^{i}_{i-3} \in [3,4,5]$
	$\rightarrow (3) ; (8)$	: (3) RI	; (8) RI	; (8) P	punch ${}^{3}\Delta^{i}_{i-3}$ and store for next card
P_{11}	$: {}^{2}\Delta^{i-1}_{i-3} \rightarrow [1,2]$	: (2) RO	$; [1,2]$ RI +		$\Longrightarrow {}^{2}\Delta^{i-1}_{i-3} + {}^{3}\Delta^{i}_{i-3} = {}^{2}\Delta^{i}_{i-2} \in [1,2]$
P_{12}	$: {}^{2}\Delta^{i}_{i-2} \rightarrow [3,4,5]$	$: [1,2]$ RO	$; [3,4,5]$ RI-		$\Longrightarrow - {}^{7}\Delta^{i}_{i-7} - \ldots - {}^{2}\Delta^{i}_{i-2} \in [3,4,5]$
	$\rightarrow (2); (8)$	: (2) RI	; (8) RI	; (8) P	punch ${}^{2}\Delta^{i}_{i-2}$ and store for next card
P_{13}	$: {}^{1}\Delta^{i-1}_{i-2} \rightarrow [1,2]$	: (1) RO	$; [1,2]$ RI +		$\Longrightarrow {}^{1}\Delta^{i-1}_{i-2} + {}^{2}\Delta^{i}_{i-2} = {}^{1}\Delta^{i}_{i-1} \in [1,2]$
P_{14}	$: {}^{1}\Delta^{i}_{i-1} \rightarrow [3,4,5]$	$: [1,2]$ RR	$; [3,4,5]$ RI -		$\Longrightarrow - {}^{7}\Delta^{i}_{i-7} - \ldots - {}^{1}\Delta^{i}_{i-1} \in [3,4,5]$
	$\rightarrow (1) ; (8)$	: (1) RI	; (8) RI	; (8) P	punch ${}^{1}\Delta^{i}_{i-1}$ and store for next card
P_{15}	$: {}^{7}\Delta^{i}_{i-7} \rightarrow [1,2]$	: (7) RO	$; [1,2]$ RI +		store for next card
	$: f_i \rightarrow [3,4,5]$	$: [6,8]$ RR	$; [3,4,5]$ RI -		$\Longrightarrow - [f_i + {}^{1}\Delta^{i}_{i-1} + {}^{2}\Delta^{i}_{i-2} + \ldots + {}^{7}\Delta^{i}_{i-7}] \in [3,4,5]$
		: READ			

Note: Wire [3,4,5] exits → [1,2] entries through a Co-Sel picked up at P_1.

Remarks:

$-f_{i-1} - {}^{1}\Delta^{i-1}_{i-2} - {}^{2}\Delta^{i-1}_{i-3} - \dots - {}^{7}\Delta^{i-1}_{i-8}$ is stored in [3,4,5] from previous card.

${}^{1}\Delta^{i-1}_{i-2}$ is stored in (1)

${}^{2}\Delta^{i-1}_{i-3}$ " " " (2)

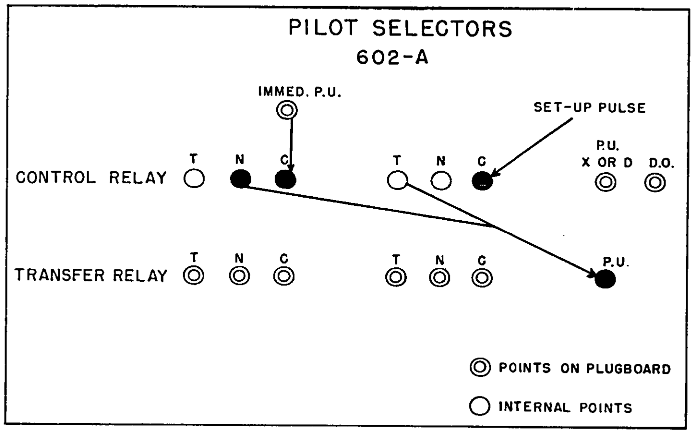
.
.
.

${}^{7}\Delta^{i-1}_{i-8}$ " " " (7)

Functional Notation:

$$
\begin{array}{ccccccccc}
 & & & & & & & {}^{7}\Delta^{i-1}_{i-8} & \\
 & & & & & & {}^{6}\Delta^{i-1}_{i-7} & & {}^{8}\Delta^{i}_{i-8} \\
 & & & & & {}^{5}\Delta^{i-1}_{i-6} & & {}^{7}\Delta^{i}_{i-7} & \\
 & & & & {}^{4}\Delta^{i-1}_{i-5} & & {}^{6}\Delta^{i}_{i-6} & & \\
 & & & {}^{3}\Delta^{i-1}_{i-4} & & {}^{5}\Delta^{i}_{i-5} & & & \\
 & & {}^{2}\Delta^{i-1}_{i-3} & & {}^{4}\Delta^{i}_{i-4} & & & & \\
 & {}^{1}\Delta^{i-1}_{i-2} & & {}^{3}\Delta^{i}_{i-3} & & & & & \\
 f_{i-1} & & {}^{2}\Delta^{i}_{i-2} & & & & & & \\
 & {}^{1}\Delta^{i}_{i-1} & & & & & & & \\
 f_{i} & & & & & & & &
\end{array}
$$

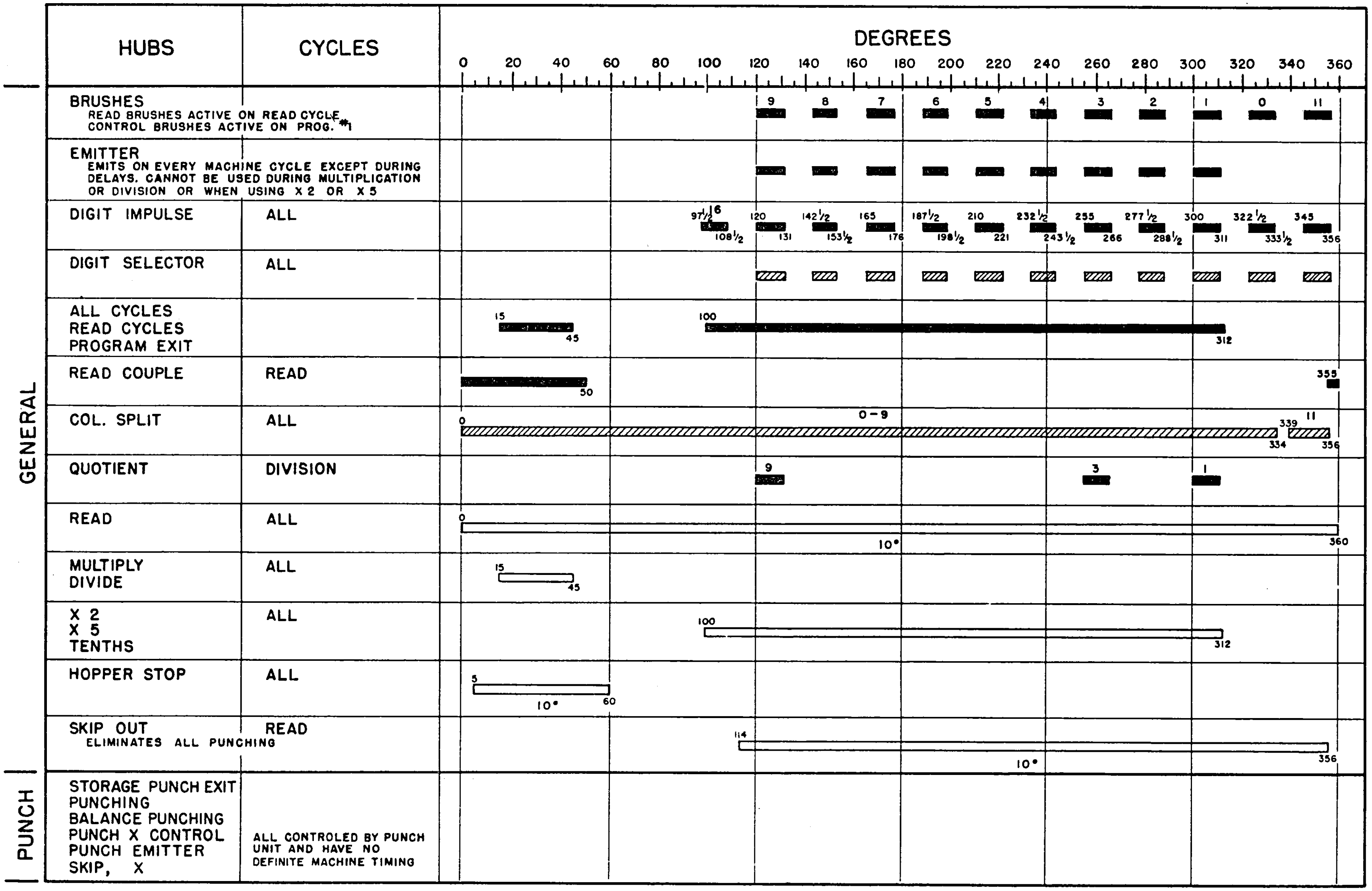
$${}^{8}\Delta^{i}_{i-8} = f_i - [\, f_{i-1} + {}^{1}\Delta^{i-1}_{i-2} + {}^{2}\Delta^{i-1}_{i-3} + {}^{3}\Delta^{i-1}_{i-4} + {}^{4}\Delta^{i-1}_{i-5} + {}^{5}\Delta^{i-1}_{i-6} + {}^{6}\Delta^{i-1}_{i-7} + {}^{7}\Delta^{i-1}_{i-8} \,]$$

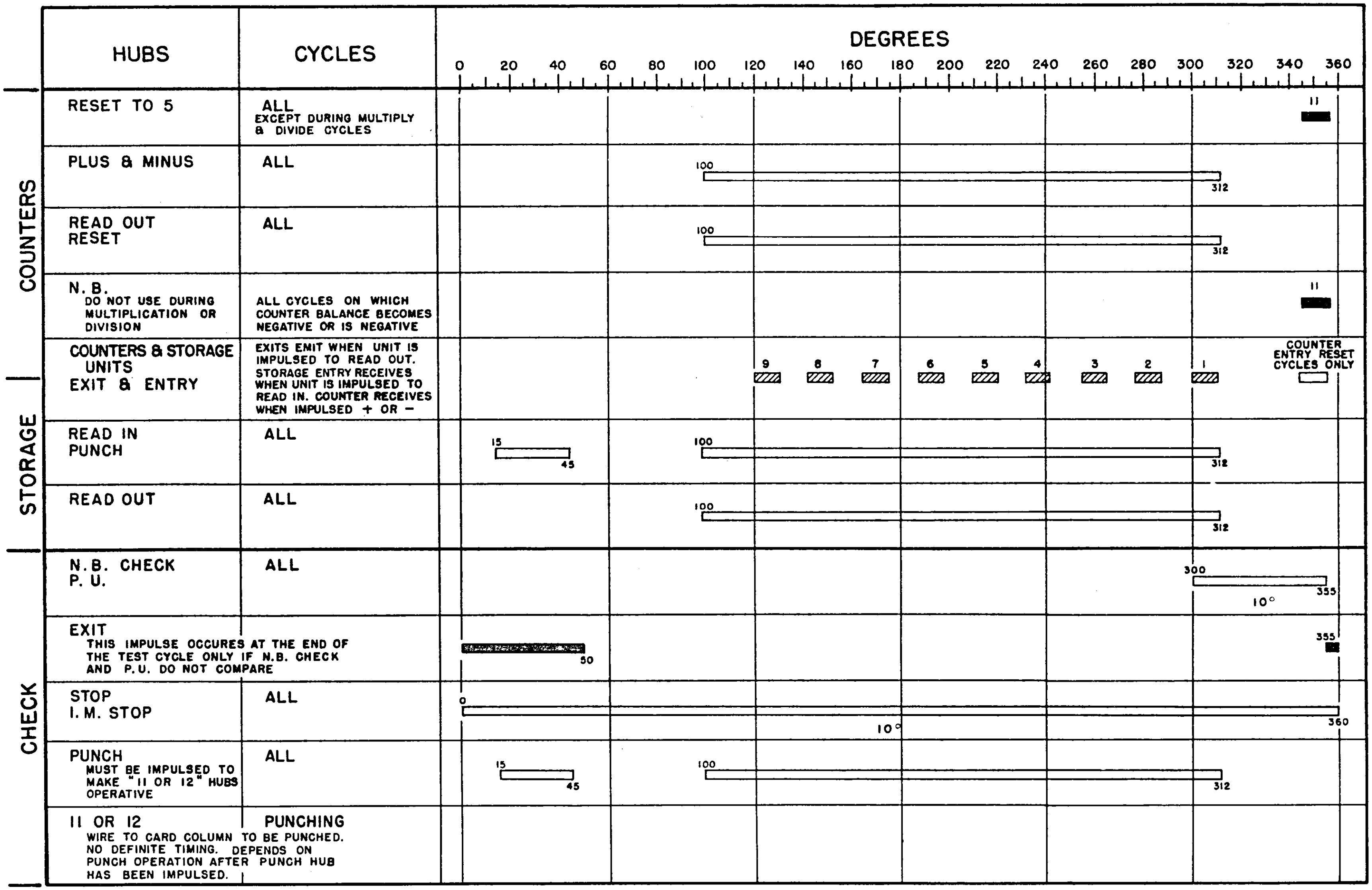


602-A TIMING CHART

PREPARED BY
NORMAN LOEBER
& JOHN LOWE
6-20-49

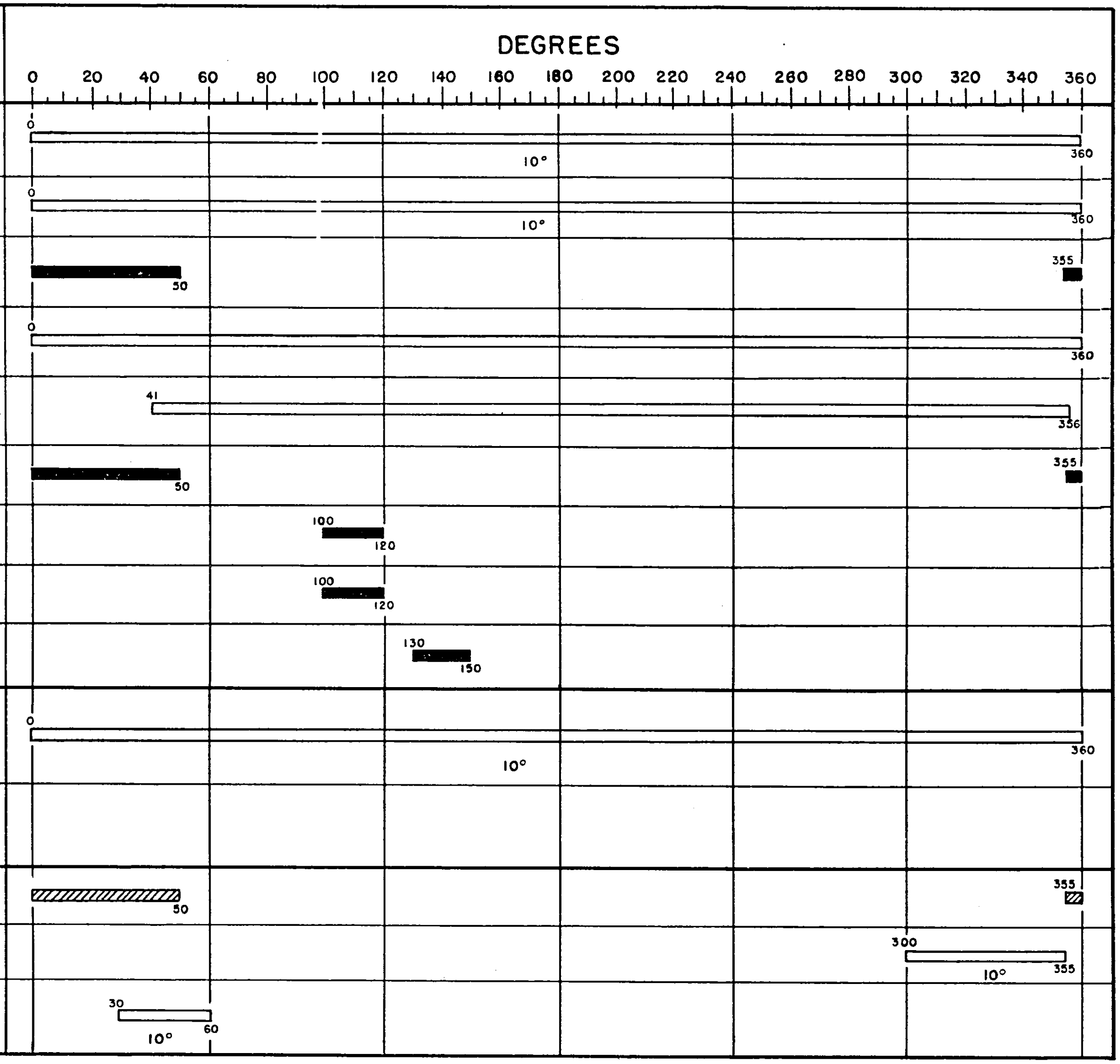
ENTRY □ EXIT ■ ENTRY OR EXIT ▨





HUBS	CYCLES
RESET TO 5	ALL EXCEPT DURING MULTIPLY & DIVIDE CYCLES
PLUS & MINUS	ALL
READ OUT RESET	ALL
N. B. DO NOT USE DURING MULTIPLICATION OR DIVISION	ALL CYCLES ON WHICH COUNTER BALANCE BECOMES NEGATIVE OR IS NEGATIVE
COUNTERS & STORAGE UNITS EXIT & ENTRY	EXITS EMIT WHEN UNIT IS IMPULSED TO READ OUT. STORAGE ENTRY RECEIVES WHEN UNIT IS IMPULSED TO READ IN. COUNTER RECEIVES WHEN IMPULSED + OR −
READ IN PUNCH	ALL
READ OUT	ALL
N. B. CHECK P. U.	ALL
EXIT THIS IMPULSE OCCURES AT THE END OF THE TEST CYCLE ONLY IF N.B. CHECK AND P.U. DO NOT COMPARE	
STOP I. M. STOP	ALL
PUNCH MUST BE IMPULSED TO MAKE "11 OR 12" HUBS OPERATIVE	ALL
11 OR 12 WIRE TO CARD COLUMN TO BE PUNCHED. NO DEFINITE TIMING. DEPENDS ON PUNCH OPERATION AFTER PUNCH HUB HAS BEEN IMPULSED.	PUNCHING

	HUBS	CYCLES
PILOT SELECTORS TO PERMIT EXACT DESCRIPTION, OPERATION OF INDIVIDUAL RELAYS IS SHOWN, INCLUDING SOME INTERNAL CIRCUITS.	CONTROL RELAY TRANSFERS IMMEDIATELY & HOLDS UNTIL DROPPED OUT OPERATED DIRECTLY BY "D" HUB OPERATED BY "X" HUB BETWEEN 339° AND 356°	
	DROP OUT HUB DROPS CONTROL RELAY IMMEDIATELY	
	SET-UP PULSE INTERNALLY EMITTED THRU TRANSFERRED POINTS OF CONTROL RELAY TO OPERATE PICKUP COIL OF TRANSFER RELAY	
	TRANSFER RELAY – PICKUP COIL WHEN IMPULSED, SELECTOR POINTS (HUBS) TRANSFER IMMEDIATELY. "IMMEDIATE" HUB IS INLET TO THIS PICKUP THRU NX POINTS OF CONTROL RELAY	
	TRANSFER RELAY – HOLD COIL INTERNAL HOLD CIRCUIT FOR SELECTOR POINTS (HUBS) IF PICKUP COIL IS UP ANYTIME BETWEEN 41° AND 356° THIS CIRCUIT AUTOMATICALLY HOLDS IT UP UNTIL 356°	
	COUPLE EXIT	CYCLES DURING WHICH SELECTOR IS TRANSFERRED
	READ DROP OUT	READ
	PUNCH DROP OUT OCCURS ON FIRST MACHINE CYCLE WHICH OCCURS AFTER CARD IS COMPLETELY PUNCHED	
	PUNCH CONTROL EXIT OCCURS ON SAME CYCLE AS PUNCH DROP OUT IF SELECTOR IS TRANSFERRED	
CO-SELECTORS	PICKUP PICKUP SHOT TRANSFERS POINTS IMMEDIATELY. POINTS WILL STAY UP AS LONG AS PICKUP IS BEING IMPULSED	ALL
	TRANSFER POINTS IF PICKUP SHOT EXTENDS PAST 41° OR OCCURS BETWEEN 41° AND 356°, POINTS WILL HOLD UNTIL 356° OF THE SAME CYCLE	
PROGRAM	PROGRAM COUPLE EMITS ON CORRESPONDING PROGRAM CYCLE. USED AS ENTRY WHEN COUPLING PROGRAM STEPS	PROGRAM
	PROGRAM SKIP	ALL
	PUNCH INTERLOCK DELAYS & REPEATS PROGRAM STEP EACH CYCLE IMPULSED. HOWEVER, ON REPEATED PROGRAMS ALL MACHINE FUNCTIONS ARE INOPERATIVE	ALL



NOTES ON 602-A TIMING CHART

1) The card feed cycle consists of two machine cycles. On the first of these, a card is read at the reading brushes. On the second, the succeeding card is read at the control brushes. This second cycle coincides with program 1.

2) Each machine cycle may be thought of as broken into two sections: 0° thru 120° and 120° thru 360°. The first section operates control functions and the second section consists of the actual calculating operation. Note that all digit times are in the second section.

3) In some cases, entry times shown are not the entire time that the functions will receive, but rather the times that they must be impulsed for correct operation. Usually, when this condition exists, the relays will pick up but will not hold if impulsed at times other than those shown. Also, during multiplication by 2 or by 5, the timing of several functions is automatically extended past 312° to 334°.

4) The "10°" shown under the times for some entry hubs means that the function must be impulsed for at least 10° somewhere within the time shown to cause proper and safe operation.

5) Timing of C and CI hubs varies widely during different operations. If it is desired to select these hubs, use a selector which is up for 360°.

STOP CONTROLS FOR THE IBM TYPE 604
ELECTRONIC CALCULATING PUNCH

Bruse Moncreiff

The Prudential Insurance Company of America

Stop Controls should be wired so that the error card is the top card in the stacker. If one of the controls is incorrectly wired, the operator may look in vain for an error in the top card. The machine may have been wired so that the stop occurs with the error card still in the machine, to be delivered to the stacker one or two cycles later, or with the error card buried in the stacker under either one or two cards. It is important that the operator be able to find the error card should an error stop occur. More serious than the loss of time is the danger that the operator may decide that since the top card does not contain an error, the machine has stopped by accident.

There are three stop hubs, labeled "lst Reading" , "Punch" , and "2nd Reading". The entry of an impulse into any of these hubs will cause the machine to stop. The difference between them is the number of cycles which are completed after the impulse occurs, before the machine stops. If the impulse is wired to "2nd Reading Stop" , the machine will stop at the end of the cycle during which the impulse occurs. If the impulse is wired to "Punch Stop", the machine will stop at the end of the cycle following the one in which the impulse occurs. In other words there is a one cycle delay in stopping. If the impulse is wired to "lst Reading Stop", there will be a two cycle delay in stopping the machine.

It should be recalled that five machine cycles are required to move the card from the hopper through the machine and into the stacker. The only other facts required to correctly wire the stop controls are contained in the 604 Timing Chart. It can be seen from this chart that the "Unfinished Program" and "Zero Check" hubs both emit X impulses on the cycle following the calculate time in which those conditions are detected. These impulses come when the card which caused either of these conditions is at the punch station (assuming that the factors used in the calculation were read at "lst Reading"). This impulse should be wired to the stop hub which will allow the machine to run until the card in error has been delivered to the stacker. An immediate stop would leave the error card just past the punches; a one cycle delay would leave the card just past the "2nd Reading" brushes; and a two cycle delay would leave the card just delivered to the stacker. The latter is the desired situation. Therefore, under the normal circumstances where all factors are read at "lst Reading", both "Unfinished Program" and "Zero Check" should be wired to "lst Reading Stop". (See chart, line 1, 2.)

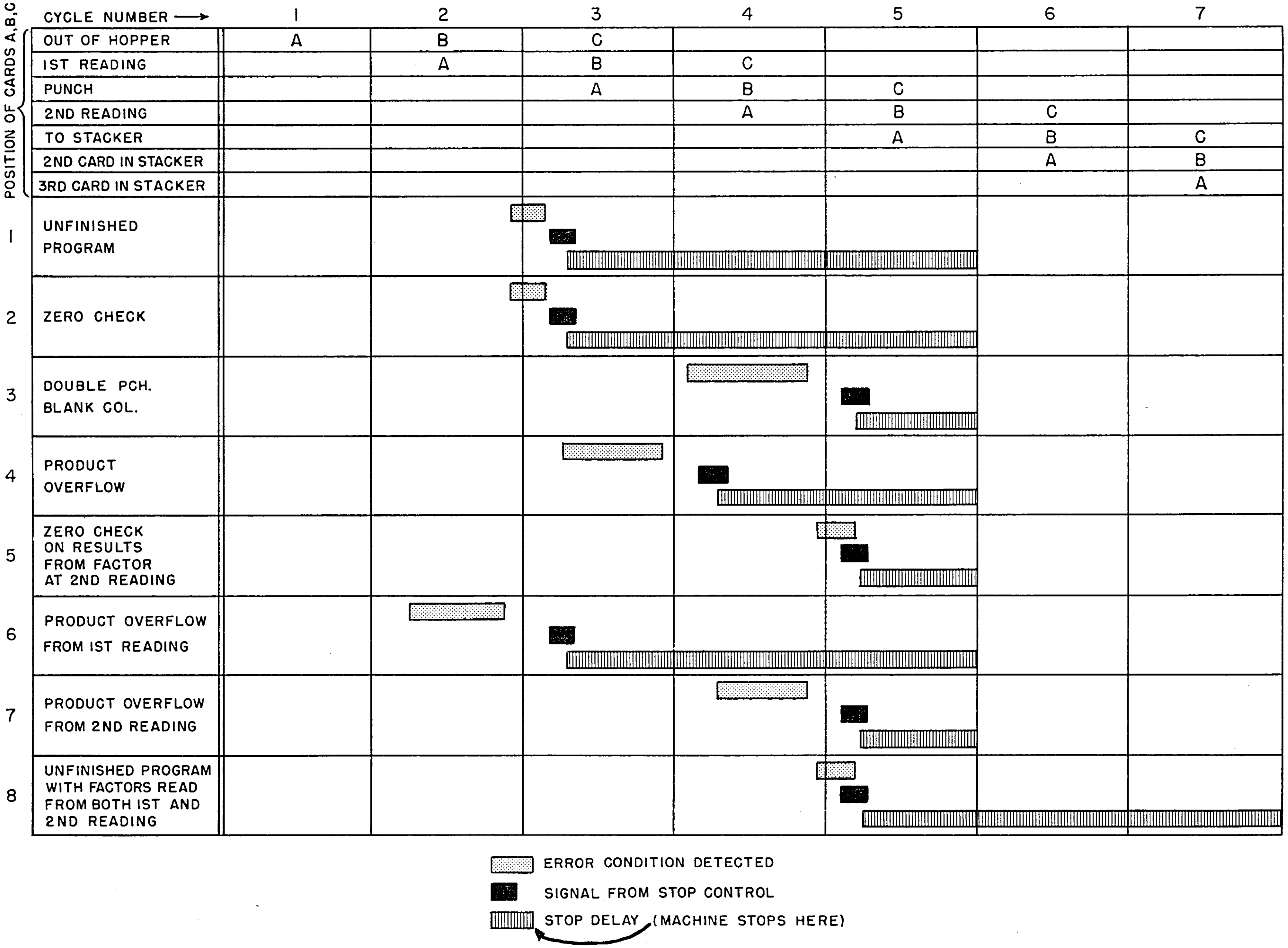
If, however, some of the factors are being read at "2nd Reading" and a recalculation and zero check is wired, then "Zero Check" should be wired to "2nd Reading Stop". (See chart, line 5.) In cases such as this where factors from two cards (one set from "1st Reading" and one set from "2nd Reading") are being operated on during every calculate time (604 program), an "Unfinished Program" signal indicates that both the card being calculated and the one being checked must be re-run one cycle at a time. This is done by alternately depressing the start and stop keys so that there will be a pause between each machine cycle. This gives the electronic unit time to complete the calculation. If "Unfinished Program" has been wired to "1st Reading Stop" the two cards to be re-run are the top card and the third one down. (See chart, line 8.) If, however, "Unfinished Program" was wired to "2nd Reading Stop", the error cards will be the top card and the one to be delivered to the stacker two cycles later. The wiring of "Zero Check" may alter this situation. It is possible that both lights may come on at the same time. The safest course of action is to re-run all the cards in the vicinity and manually check the results.

Double punched or blank column cards read at either brushes into the "Double Punch and Blank Column Entry" hubs are signalled at X time of the following card cycle by an impulse from the "DPBC" hub. Double punches or blank columns detected at the 2nd reading brushes would be signalled in the cycle in which the error card is being delivered to the stacker. In order to stop with the error card uppermost in the stacker, "2nd Reading Stop" should be wired from "DPBC". (See chart, line 3.) If the card is detected at first reading brushes, the "1st Reading Stop" should be wired.

The product overflow device can be used to signal the presence of a digit impulse other than zero from any exit: "1st Reading", "Counter Exit", "General Storage Exit", or "2nd Reading". It is most often used to detect a result being read out to "Punching" which is greater than the card field assigned to it. The condition arises during the cycle in which the error card is being punched, and is signalled by an X impulse from "Product Overflow Out" the following cycle. If this impulse were to be wired to "2nd Reading Stop", the machine would stop with the error card just past the 2nd brushes. The correct wiring would be to "Punch Stop". (See chart, line 4.) This would cause a one cycle delay in the stopping, allowing the error card to be delivered to the stacker. Similar reasoning can be used to determine the correct wiring when product overflow is used for other purposes. (See chart, lines 6, 7.)

It is important that the operator know what the various error lights indicate on each job. Since these controls are pluggable, and their use varies from job to job, the meaning of these error lights should be made a part of the operating instructions. These instructions should include the type of error (double punch, error in calculation, etc.), the location of the error card when the machine stops, the field involved, and the method of calculation. This information will give the operator all the information he needs to detect and correct errors. This dual attack of correct control panel wiring and full information for the operator should prevent errors from slipping through.

604 STOP CONTROLS



CYCLE NUMBER →	1	2	3	4	5	6	7
OUT OF HOPPER	A	B	C				
1ST READING		A	B	C			
PUNCH			A	B	C		
2ND READING				A	B	C	
TO STACKER					A	B	C
2ND CARD IN STACKER						A	B
3RD CARD IN STACKER							A

POSITION OF CARDS A,B,C
1 UNFINISHED PROGRAM
2 ZERO CHECK
3 DOUBLE PCH. BLANK COL.
4 PRODUCT OVERFLOW
5 ZERO CHECK ON RESULTS FROM FACTOR AT 2ND READING
6 PRODUCT OVERFLOW FROM 1ST READING
7 PRODUCT OVERFLOW FROM 2ND READING
8 UNFINISHED PROGRAM WITH FACTORS READ FROM BOTH 1ST AND 2ND READING
ERROR CONDITION DETECTED
SIGNAL FROM STOP CONTROL
STOP DELAY (MACHINE STOPS HERE)

AN EIGHT-DIGIT GENERAL-PURPOSE CONTROL PANEL

William P. Heising

IBM Technical Computing Bureau

Introduction

In the course of our work in the IBM Technical Computing Bureau, we have found it desirable to set up a 604 control panel to evaluate electronically various trigonometric, exponential, and logarithmic series, in addition to the normal arithmetic operations, for a CPC or a 604 equipped with punch delay. For these series, a number, n, related to the highest power of the argument to be used in the truncated series, is read into the 604 with the argument. The series is evaluated by the method of continued products on successive program repeats.

Method

The eight-digit factors, A and B (with seven decimal digits) are read into FS1-2 and FS3-4 respectively. GS3 and GS4 are impulsed to read in on every card cycle; a "1" is emitted into GS4 in the high order position, and the number, n, is read into GS3 from the card for the series evaluations. The result appears in the electronic counter rounded to seven decimal digits (the two lowest positions of the counter are to be ignored). The result automatically is read into the electronic storage for factors A and B in case this result is to be used as a factor for the succeeding card.

The operations outlined in Table I require 47 program steps and six calculate selectors. The first thirteen program steps are left open to permit additional wiring for special operations using the remaining two calculate selectors. These are altered to fit the individual problem. Figure 1 shows the programming involved, and Figure 2 shows the wiring of the calculate selectors. (See pages 80-82)

TABLE I

Operation Code or Calculate Selectors Transferred	Operation	Notes	
4	$A + B$		
3, 4	$-A + B$		
8	$A \times B$		
3, 8	$-A \times B$		
7, 8	$A \div B$		
3, 7, 8	$-A \div B$		
6, 7, 8	$\sqrt{A}$	$B = 1.2$	
5	$\exp(-A)$	$B = 0$ *	n
3, 5	$\exp(+A)$	↓	n
5, 7	$\sin A$	↓	n odd
5, 7	$\cos A$	↓	n even
3, 5, 7	$\sinh A$	↓	n odd
3, 5, 7	$\cosh A$	↓	n even
5, 8	$\log \left\{\frac{1}{1 - A}\right\}$	↓	n
3, 5, 8	$\log \left\{\frac{1}{1 + A}\right\}$	↓	n
5, 7, 8	$\tanh^{-1} A$	↓	n even
3, 5, 7, 8	$\tan^{-1} A$	↓	n even
5, 6, 7, 8	$\sin^{-1} A$	↓	n even
3, 5, 6, 7, 8	$\sinh^{-1} A$	↓	n even

*See later section on "Accuracy of Series"

Notation

The planning diagram of Figure 1 uses the first column of program exits for "read-out" instructions, and the second column is generally reserved for "read-in" instructions. The following notation is used:

A = FS1-2 = $x_{\wedge}xx\ \ xxxxx$
B = FS3-4 = $x_{\wedge}xx\ \ xxxxx$

Result in electronic counter 00x $x_{\wedge}xxxx$ xxx/xx

D_1 = GS3 = n = $xxx_{\wedge}$

D_2 = GS4 = 10000

C = GS1-2 ; All assignments are 8-6

+	=	Ctr. Add	R	=	Ctr. Reset
-	=	Ctr. Subtract	RO	=	Ctr. Read Out
×	=	Multiply	RR	=	Ctr. Read Out and Reset
÷	=	Divide	MQ	=	Multiplier Quotient
Z. T.	=	Zero Test			

Roman Numeral (V) = Suppression Type V

$\underline{4}$ T = Calculate Selector 4 transferred

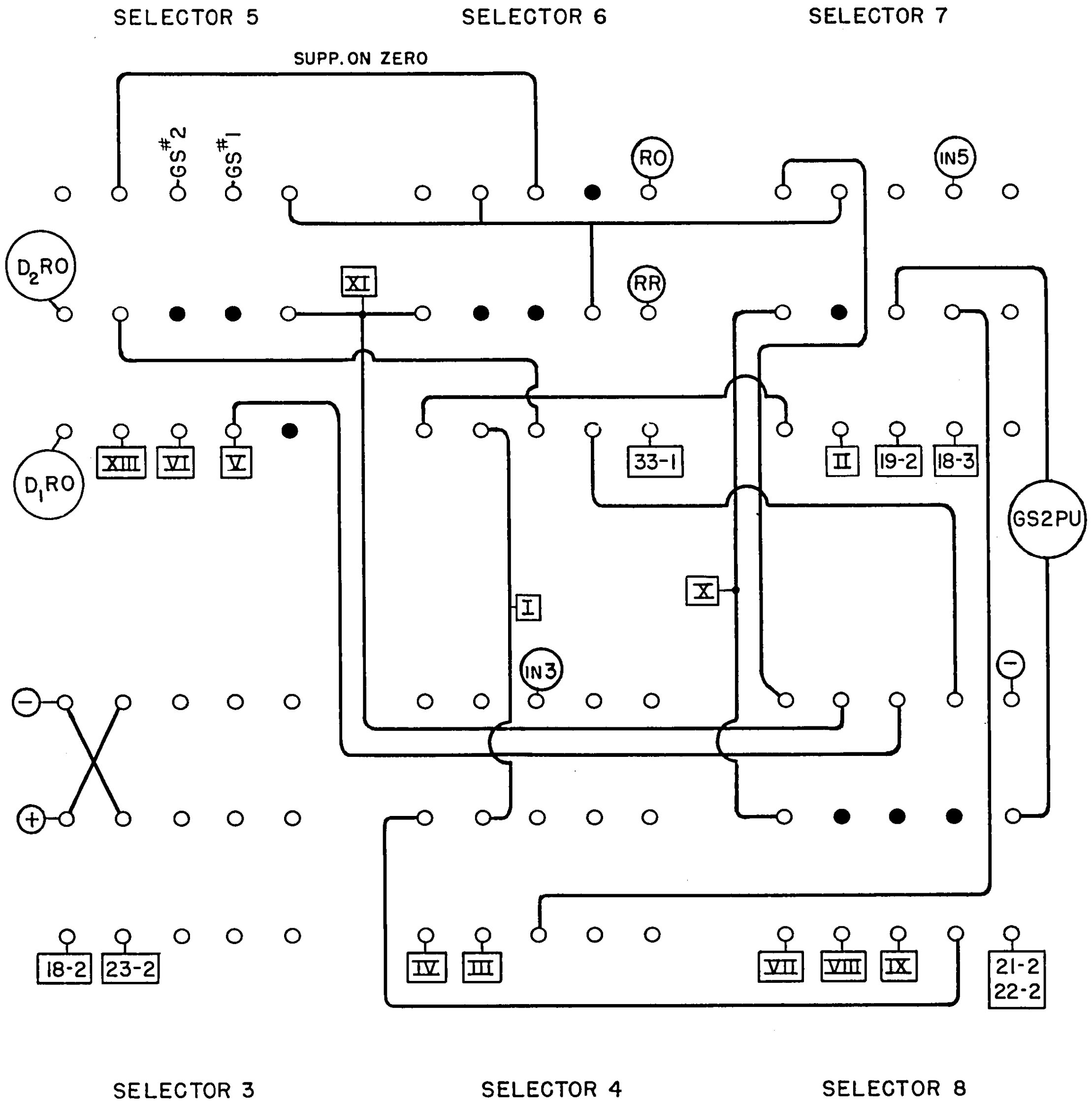
15- 3 = Third program exit of program step 15.

Supp.	P.S.	R.O.	R. I.		*Notes
I	14	B	+		
II	15	B	+		
II	16	RR	D_1	D_2RI	
III	17	B	+	in 3rd	18-2 (- if $\underline{3}$ T)
IV	18	A	+ *	*	18-3 (in 5th if $\underline{7}$ T;
XI	19	GS1 DO	GS2 PU*		in 3rd if $\underline{7}$N and $\underline{4}\underline{T}$)
VI	20	A	+	GS 1 PU	19-2 (unless $\underline{7}$ T)
VI	21	A	{GS 2 PU*}		21-2 and 22-2 (-if $\underline{8}$ T)
VI	22	A			
V	23	A	- *		23-2 (+ if $\underline{3}$ T)
VII	24	RO	MQ		
VII	25	RR	C	out 6th	
VII	26	B	×		
VII	27	C	MQ		
VII	28	RR	C	out 6th	
VII	29	C	+		
VII	30	B	×		
I	31	A	+	in 5th	
VIII	32		1/2 adj.	in 2nd	
VI	33	RR *	B	out 3rd	*33-1 (RO if $\underline{6}$ T)
VI	34	G S 2 PU			

(continued on next page)

Supp.	P.S.	R.O.	R.I.	
IX	35	RR	B	out 3rd
IX	36	D_2	+	in 6th
IX	37	GS 2 DO	GS 1 PU	
X	38	D_1	÷	
X	39	RR	C	
X	40	C	+	in 6th
X	41	MQ	C	
X	42	D_1	÷	
X	43	R		
I	44	C	+	in 6th
I	45	MQ	+	
I	46	B	-	in 3rd
XI	47	D_1	+	
XI	48		Ƶ.T.	in 6th
XI	49	RO	D_1	
VIII	50	Emit 1	+	in 6th
XII	51	R	Ƶ.T.	in 3rd
X	52	C	+	in 6th
X	53	MQ	+	
V	54	D_2	+	in 6th
V	55	GS 2 DO		
VIII	56	B	-	in 3rd
None	57		1/2 adj.	in 2nd
None	58	RO	B	out 3rd
XIII	59	R	P. Rpt.	
Supp. on Non-Zero	60	B	A	

Figure 1. Programming



NOTE: ● MEANS SUPPRESS WITHOUT BALANCE TEST.

Figure 2. Calculate Selector Wiring

Accuracy of Series

A. Exponentials, Circular and Hyperbolic Sines and Cosines

The MacLaurin series of these functions converges for all arguments. In this general-purpose control panel, the argument and the function must be less than ten in magnitude.

Of the two errors, (a) truncation error of the infinite series and (b) accumulated rounding error, the latter is the limiting factor on the accuracy obtainable. The last term retained in the finite series approximation is $Bx^n/n!$. At $n = 40$, the truncation error is negligible for all values of the argument. The accumulated rounding error for arguments of large magnitude, $|A|$, is proportional to $\exp|A|$. This error is $2\text{-}4 \times 10^{-4}$ for an argument of 9.9. The round-off error for angles near 2π radians is 5×10^{-6} in the sine and cosine.

B. Logarithms, Inverse Functions

These series converge for $|A| < 1$ or $|A| \leq 1$ (the latter where the successive terms alternate in sign). Because of the slower convergence, a greater number of terms is required, in general. The last term retained (or first term omitted if $B = 0$) is Bx^n in magnitude (except for arcsin and arcsinh series). For the inverse tangent series, if $B \doteq \frac{1}{2n}$, the function is quite accurately approximated up to and including arguments of unity. Thus for $n=990$, and $B=0.0005057$, the maximum error in the arctangent is 11×10^{-7} (at 44° 58′). For $n=90$, $B=0.0055688$, the maximum error is 1.7×10^{-5}. In both these cases, B was chosen so that $\arctan 1 = \pi/4$ exactly. In general, the rounding errors are negligible, and the truncation error limits the accuracy. The time of electronic calculation is approximately 10 program repeats per second.

PART 3

Application
of the
IBM Card-Programmed Electronic Calculator

INTERPOLATION
ON THE
IBM CARD-PROGRAMMED ELECTRONIC CALCULATOR

Stuart R. Brinkley, Jr. and G. L. Wagner

United States Department of Interior, Bureau of Mines

Stirling's interpolation formula for a function y (x), tabulated at equally spaced values of the argument x, can be written in the form,

$$y = y_0 + a_1 u + a_2 u^2 + a_3 u^3 + a_4 u^4 + \dots , \qquad (1)$$

where $u = (x - x_0)/h, \ h = x_1 - x_0 ,$

$$y = y(x), \ y_0 = y(x_0),$$

$$a_1 = (1/2) \left[\Delta^{(1)}_{-1} + \Delta^{(1)}_{0} \right] - (1/12) \left[\Delta^{(3)}_{-2} + \Delta^{(3)}_{-1} \right],$$

$$a_2 = (1/2)\, \Delta^{(2)}_{-1} - (1/24)\, \Delta^{(4)}_{-2} ,$$

$$a_3 = (1/12) \left[\Delta^{(3)}_{-2} + \Delta^{(3)}_{-1} \right],$$

$$a_4 = (1/24)\, \Delta^{(4)}_{-2} ,$$

and where $\Delta^{(1)}_{-2} = y(x_{-1}) - y(x_{-2}),$

$$\Delta^{(1)}_{-1} = y(x_0) - y(x_{-1}),$$

$$\dots .$$

$$\Delta^{(2)}_{-2} = \Delta^{(1)}_{-1} - \Delta^{(1)}_{-2} ,$$

$$\Delta^{(2)}_{-1} = \Delta^{(1)}_{0} - \Delta^{(1)}_{-1} ,$$

. . . .

$$\Delta_{-2}^{(3)} = \Delta_{-1}^{(2)} - \Delta_{-2}^{(2)} ,$$

$$\Delta_{-1}^{(3)} = \Delta_{o}^{(2)} - \Delta_{-1}^{(2)} ,$$

$$\Delta_{-2}^{(4)} = \Delta_{-1}^{(3)} - \Delta_{-2}^{(3)} .$$

In the present application, it is assumed that fifth differences vanish, and that f(x) is single-valued.

Equation 1 is employed for forward interpolation. The table is arranged in order of increasing values of the argument. Except at the beginning and end of the table, the label subscripts of the table argument are assigned so that

$$x_o \leqslant x < x_1$$

If x is less than the third table argument, then the third table argument is taken to be x_o. If x is greater than or equal to the next to last table argument, then the next to the last table argument is taken to be x_1.

For inverse interpolation, equation 1 is written in the form

$$\begin{aligned} x - x_o &= uh \\ u &= u_o + F(u) \\ u_o &= (y - y_o)/a_1 \\ F(u) &= -\frac{1}{a_1}(a_2 u^2 + a_3 u^3 + a_4 u^4). \end{aligned} \qquad (2)$$

Equation 2 is solved by iteration. The table is arranged in order of increasing values of the function. Except at the beginning and end of the table, the label subscripts of the table argument are assigned so that

$$y(x_o) \leqslant y < y(x_1).$$

If y is less than the third table value, then the third table argument is taken to be x_o. If y is greater than or equal to the next to last table value, then the next to last table argument is taken to be x_1.

The circuits for the solution of equations 1 and 2 assume that the quantity y is a function of the argument x and of a parameter, a.

$$y = y\,(a,\ x).$$

It is assumed that y is a single-valued, monotone increasing function of x.
It is assumed that the argument increment h is a one-digit positive number.
The change in variable

$$x' = a + bx,$$

can be employed to transpose into an acceptable form, cases for which y is a single-valued, monotone decreasing function of the original argument or for which the argument increment h is not a one-digit number. If the function is multiple valued, interpolation except near extrema can be carried out on a series of single-valued sections of the table.

The control circuits are initiated by a control card (first card) identified with an X punch. The control card is followed by the table cards in order. No additional control cards are employed. Six cards must follow the table card with argument x_0. If x_0 is near the end of the table, blank cards in the number necessary to provide six following cards are added to the table. For forward interpolation, the control card contains the value of the interpolation argument x in the same card field as in the table cards. For inverse interpolation, the control card contains the value of the interpolation function y in the same field as in the table cards.

Forward interpolation is performed with Setup Change Switch 2 on. Inverse interpolation is performed with Setup Change Switch 2 off. No other wiring changes are required.

FORWARD INTERPOLATION
Card Program

	Seq	Address	Channel		Counter				
		B C	B	C	2	3	4	5	6
1st Card	1	00	y			(a) x	(a) x		
1st Table Card	1	00 80	y_{-n}		(a) α	(a) $-x_{-n}$			(a) y_{-n}
2nd Table Card	1	00 85	y_{-n+1}	y_{-n}				(b) y_{-n}	
3rd Table Card	1	00 85	y_{-n+2}	$\Delta^{(1)}_{-n}$		(d) $-h$		$\Delta^{(1)}_{-n}$	
4th Table Card	1	00 85	y_{-n+3}	$\Delta^{(1)}_{-n+1}$		(d) $-h$		$\Delta^{(1)}_{-n+1}$	
5th Table Card	1	00 85	y_{-n+4}	$\Delta^{(1)}_{-n+2}$		(d) $-h$		$\Delta^{(1)}_{-n+2}$	
- - - - - - -	- -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -
nth Table Card	1	00 85	y_o	$\Delta^{(1)}_{-2}$		(d) $-h$		$\Delta^{(1)}_{-2}$	
(n + 1)th Table Card	1	00 85	y_1	$\Delta^{(1)}_{-1}$		(d) $-h$		$\Delta^{(1)}_{-1}$	
(n + 2)th Table Card	1	00 85	y_2	$\Delta^{(1)}_{o}$		(d) $-h$		$\Delta^{(1)}_{o}$	
(n + 3)th Table Card		00 85	y_3			(d) $-h$			
(n + 4)th Table Card		83	y_4						
(n + 5)th Table Card	2	00 75	$x-x_o-2h$			$\sim\sim$		(c) $-\Delta^{(1)}_{o}$	
(n + 6)th Table Card		00	y_6	$y-y_o$				$y-y_o$	
(n + 7)th Table Card		Punch			α		x	y	

(a) From 3rd Reading
(b) From Ctr 6 Exit
(c) From GS 4 Exit
(d) From Digit Emitter

Sequence 1 - Differencing Sequence
Sequence 2 - Interpolation Sequence

INVERSE INTERPOLATION

Card Program

		Address		Channel		Counter				
	Seq.	B	C	B	C	2	3	4	5	6
1st Card	1	00		y						(a) y
1st Table Card	1	00	85	y_{-n}		(a) α	(a) x_n			
2nd Table Card	1	00	85	y_{-n+1}	$y_{-n} - y$				$y-y_{-n}$	
3rd Table Card	1	00	85	y_{-n+2}	$\Delta^{(1)}_{-n}$				$-\Delta^{(1)}_{-n}$	
4th Table Card	1	00	85	y_{-n+3}	$\Delta^{(1)}_{-n+1}$				$-\Delta^{(1)}_{-n+1}$	
5th Table Card	1	00	85	y_{-n+4}	$\Delta^{(1)}_{-n+2}$		(b) h		$-\Delta^{(1)}_{-n+2}$	
- - - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -	- - -
nth Table Card	1	00	85	y_0	$\Delta^{(1)}_{-2}$		(b) h		$-\Delta^{(1)}_{-2}$	
(n + 1)th Table Card	1	00	85	y_1	$\Delta^{(1)}_{-1}$		(b) h		$-\Delta^{(1)}_{-1}$	
(n + 2)th Table Card	1	00	85	y_2	$\Delta^{(1)}_{0}$		(b) h		$-\Delta^{(1)}_{0}$	
(n + 3)th Table Card		00	85	y_3			(b) h			
(n + 4)th Table Card		85		y_4						
(n + 5)th Table Card	2	00	73	$y-y_0-\Delta^{(1)}_0$					$\sim\sim\sim$	
(n + 6)th Table Card		00		y_5	$x-x_0$		$x-x_0$			
(n + 7)th Table Card		Punch				α	x			y

(a) From 3rd Reading
(b) From Digit Emitter

Sequence 1 - Differencing Sequence
Sequence 2 - Interpolation Sequence

INTERPOLATION

604 Program

No.	Sup.	Sel.	Factor Storage 1(6-4) 3	Factor Storage 2	Factor Storage 4	Mult. Quot.	Ctr.	General Storage 1(6-4) 3	General Storage 2	General Storage 4	1	2	I	F
1	1			RO	RI						x		x	x
2	2						RO		RI			x	x	x
3	2					RO	RI (-)					x	x	x
4	3	1N		RI			RO, RC					x	x	x
		1T				RI	RO, RC				x		x	x
5	1								RO	RI	x		x	x
6	2					RO	RI (+)					x	x	x
7	2						RI (+)		RO			x	x	x
8	2						RO, RC		RI			x	x	x
9	2	3N					RI (+)			RO		x	x	
		3T	(EMITTER 2h)				RI (+)4					x		x
10	3		RO				RI (+)				x	x	x	x
11	1						RI (-)	RO			x		x	x
12	1		RO					RI			x		x	x
13	1						RO		RI		x		x	x
14	2		RI				RO, RC					x	x	x
15	3	1N					RI (+)			RO		x	x	x
		1T					RI (-)			RO	x		x	x
16	1			RI			RO				x		x	x
17	2						RI (+)			RO		x	x	x
18	3				RO		RI (-)				x	x	x	x
19	2						RO, RC			RI		x	x	x
20	2		(EMITTER 5)			RI4						x	x	x
21	2			RO			MULT(-)					x	x	x
22	2			RI			RO5					x	x	x

(continued on next page)

No.	Sup.	Sel.	Factor Storage			Mult. Quot.	Ctr.	General Storage			2	I	F
			1 (6-4) 3	2	4			1(6-4) 3	2	4			
23	2		(EMITTER 6)			RI5					x	x	x
24	2				RO		MULT(+)				x	x	x
25	2				RI		RO5, RC				x	x	x
26	2						MULT(+)			RO	x	x	x
27	2						RI(-) 5		RO		x	x	x
28	2						RO5, RC	RI			x	x	x
29	2	3N	RO				MULT(+)				x	x	
		3T	(EMITTER h)							RI4	x		x
30	2	4N	RO				MULT(+)				x	x	
		4T	RO				RI (+) 5				x		x
31	2	4N				DIV		RO			x	x	
		4T				DIV				RO	x		x
32	2						RO, RC				x	x	x
33	2	4N	RI			RO					x	x	
34	4						RI(-) 5		RO		x	x	x
35	4			RO			MULT (+)				x	x	x
36	4						RO4, RC			RI	x	x	x
37	4				RO		RI(-) 6				x	x	x
38	4						MULT(+)			RO	x	x	x
39	4						RO5, RC			RI	x	x	x
40	4	5T					RI(-) 6	RO			x		x
41	4						MULT(+)			RO	x	x	x
42	4	5N					RO5, RC			RI	x	x	
		5T					RO5, RC	RI			x		x
43	4	5N 6N					MULT(+)			RO	x	x	

(continued on next page)

No.	Sup.	Sel.	Factor Storage			Mult. Quot.	Ctr.	General Storage			2	I	F
			1(6-4) 3	2	4			1(6-4) 3	2	4			
43	4	5T 6T					MULT(-)	RO			x		x
44	4		(DO GS 2)			RO				RI	x	x	x
45	5					DIV		RO			x	x	
46	4	6N					RO, RC				x	x	
		6T					RO6,RC	RI			x		x
47	5		(PU GS 1)			RO	RI(+)				x	x	
48	5		RO				RI(+)2				x	x	
49	5					RI	RO2				x	x	
50	5						RI(-)2			RO	x	x	
51	4	6N	(EMITTER 2)				RI(-)2				x	x	
		6T	(EMITTER 2)				RI(+)				x		x
52	5		(BALANCE TEST)								x	x	
53	6	(PROGRAM REPEAT, PU GS 2)									x	x	
54	4	7N	(EMITTER 4)				RI(+)2				x	x	
		7T	(EMITTER 1)				RI(+) 2				x		x
55	5		(BALANCE TEST)								x	x	
56	7	(PROGRAM REPEAT, PU GS 2)									x	x	
57	4	7N					RO, RC				x	x	
		7T					RO, RC			RI	x		x
58	8	7N	(EMITTER h)							RI	x	x	
		7T					RI(+)	RO			x		x
59	8	8N					MULT(+)			RO	x	x	
		8T				DIV				RO	x		x
60	8	8N					RO2, RC		RI		x	x	
		8T				RO			RI		x		x

INTERPOLATION

Selector Transfer Times

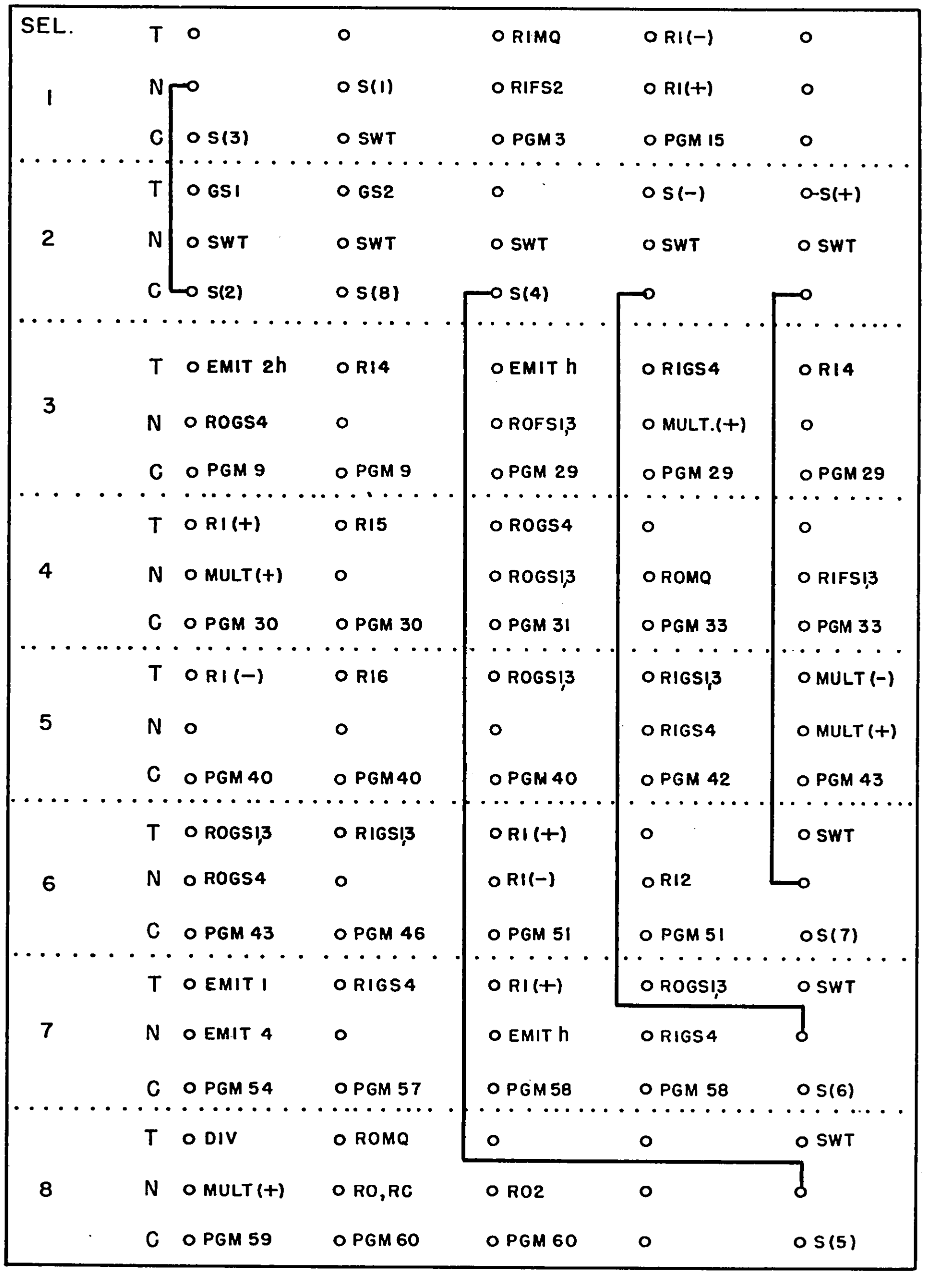
Card	Pilot Selectors											Co Selectors						Calc. Sel.		Field Sel.
	1	2	3	4	5	6	7	8	9	10	11	2	3	4	6	7	8	1	2	
1st Card	T	N	N	N	N	N	N	N	N	N	N	T	N	N	N	N	T	T	N	—
1st Table Card	N	T	N	N	N	N	N	T	N	N	N	N	T	T	N	N	T	T	N	—
2nd Table Card	N	N	T	N	N	N	N	T	N	N	N	N	N	T	N	N	N	T	N	—
3rd Table Card	N	N	N	T	N	N	N	T	N	N	N	N	N	T	N	N	N	T	N	—
4th Table Card	N	N	N	N	T	N	N	T	N	N	N	N	N	T	N	N	N	T	N	—
5th Table Card	N	N	N	N	N	T	N	T	N	N	N	N	N	T	N	N	N	T	N	—
6th Table Card	N	N	N	N	N	N	T	T	N	N	N	N	N	T	N	N	N	T	N	—
....	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
(n + 2)th Table Card	N	N	N	N	N	N	T	T	N	N	N	N	N	T	N	N	N	T	N	—
(n + 3)th Table Card	N	N	N	N	N	N	T	T	T	N	N	N	N	T	N	N	N	N	N	—
(n + 4)th Table Card	N	N	N	N	N	N	N	N	N	T	N	N	N	N	T	N	N	N	N	1
(n + 5)th Table Card	N	N	N	N	N	N	N	N	N	N	T	N	N	N	N	T	N	N	T	—
(n + 6)th Table Card	N	N	N	N	T	N	N	N	N	N	N	N	N	N	N	N	N	N	N	—
(n + 7)th Table Card	N	N	N	N	N	T	N	N	N	N	N	N	N	N	N	N	N	N	N	—
(n + 8)th Table Card	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	N	—

Co Selectors 1 and 5 and Calculate Selectors 3-8 are T all cycles for forward interpolation. (Setup Change Switch 2 on.)

Co Selectors 1 and 5 and Calculate Selectors 3-8 are N all cycles for inverse interpolation. (Setup Change Switch 2 off.)

INTERPOLATION

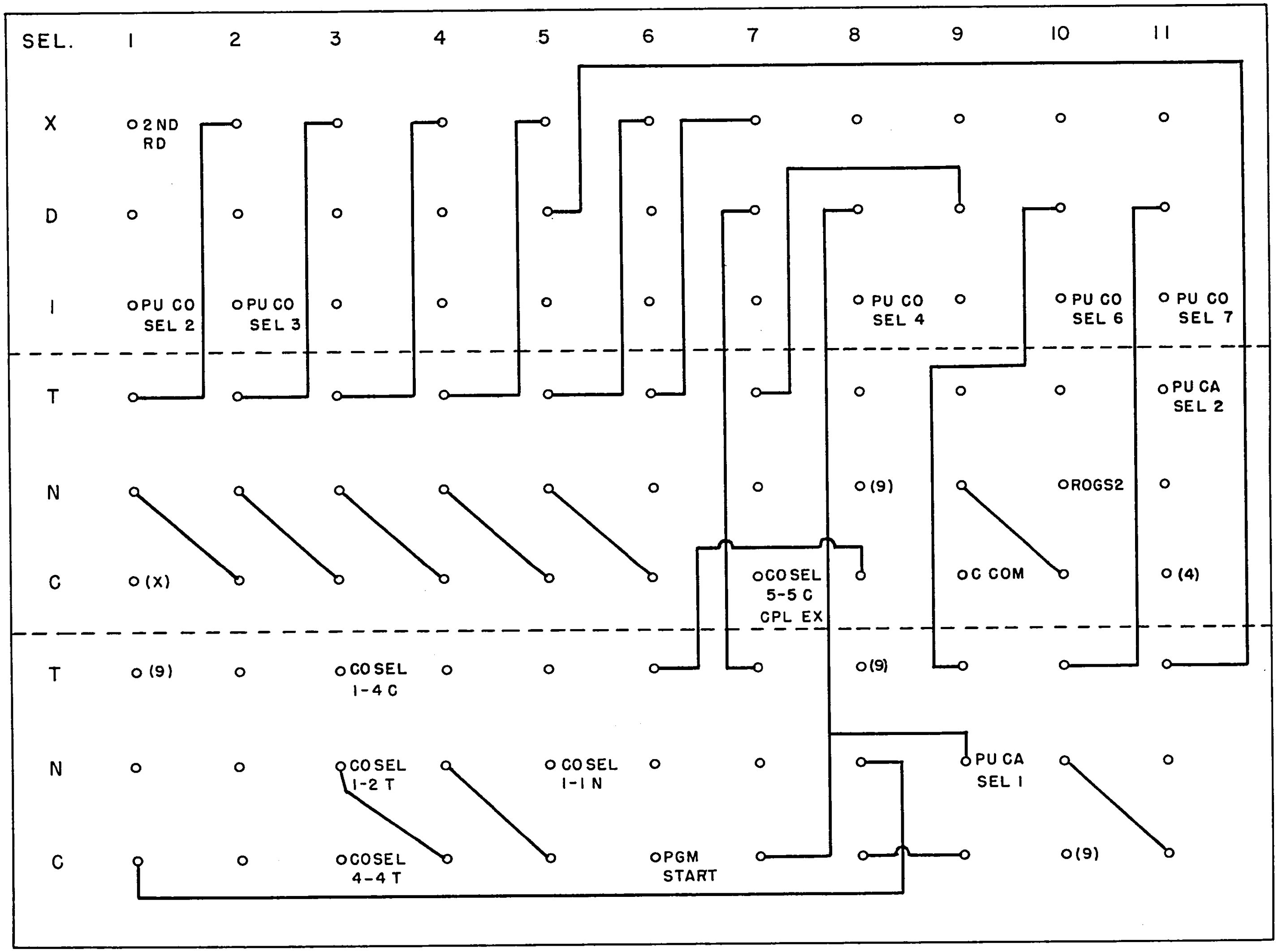
Calculate Selector Circuits



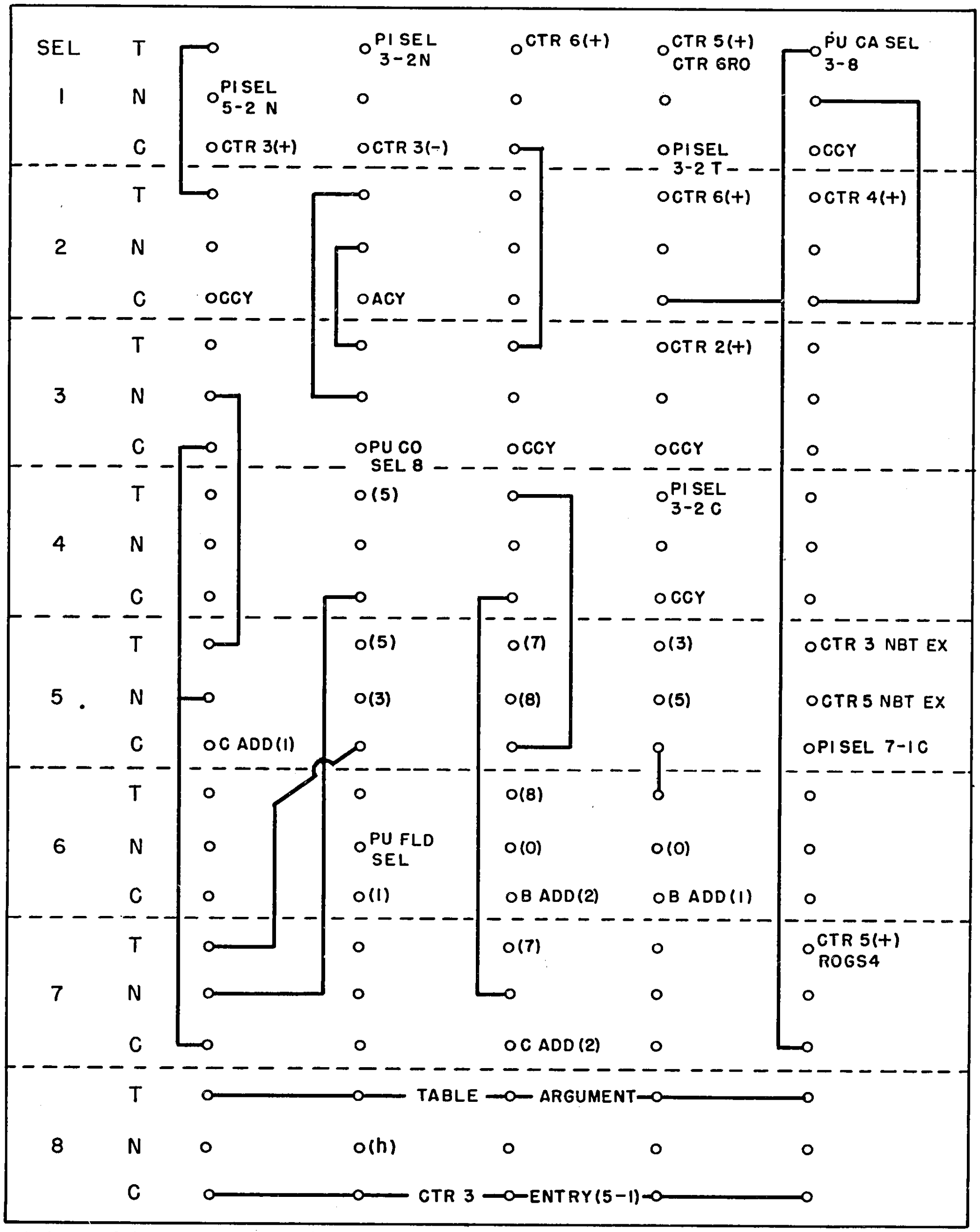
SEL.						
	T	o	o	o RIMQ	o RI(−)	o
1	N	o	o S(1)	o RIFS2	o RI(+)	o
	C	o S(3)	o SWT	o PGM 3	o PGM 15	o
	T	o GS1	o GS2	o	o S(−)	o-S(+)
2	N	o SWT	o SWT	o SWT	o SWT	o SWT
	C	o S(2)	o S(8)	o S(4)	o	o
	T	o EMIT 2h	o RI4	o EMIT h	o RIGS4	o RI4
3	N	o ROGS4	o	o ROFS1,3	o MULT.(+)	o
	C	o PGM 9	o PGM 9	o PGM 29	o PGM 29	o PGM 29
	T	o RI(+)	o RI5	o ROGS4	o	o
4	N	o MULT(+)	o	o ROGS1,3	o ROMQ	o RIFS1,3
	C	o PGM 30	o PGM 30	o PGM 31	o PGM 33	o PGM 33
	T	o RI(−)	o RI6	o ROGS1,3	o RIGS1,3	o MULT(−)
5	N	o	o	o	o RIGS4	o MULT(+)
	C	o PGM 40	o PGM 40	o PGM 40	o PGM 42	o PGM 43
	T	o ROGS1,3	o RIGS1,3	o RI(+)	o	o SWT
6	N	o ROGS4	o	o RI(−)	o RI2	o
	C	o PGM 43	o PGM 46	o PGM 51	o PGM 51	o S(7)
	T	o EMIT 1	o RIGS4	o RI(+)	o ROGS1,3	o SWT
7	N	o EMIT 4	o	o EMIT h	o RIGS4	o
	C	o PGM 54	o PGM 57	o PGM58	o PGM 58	o S(6)
	T	o DIV	o ROMQ	o	o	o SWT
8	N	o MULT(+)	o RO,RC	o RO2	o	o
	C	o PGM 59	o PGM 60	o PGM 60	o	o S(5)

Note: PGM means program step.

INTERPOLATION
Pilot Selector Circuits



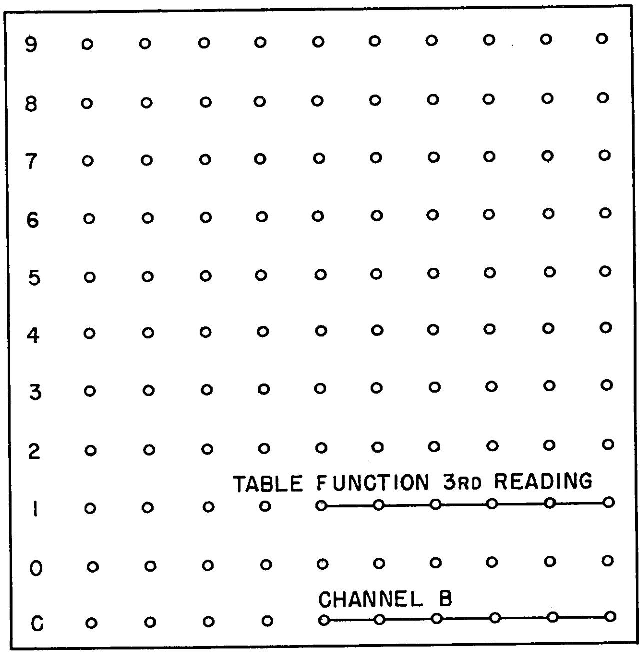
INTERPOLATION
Co Selector Circuits



Note: CCY means card cycle
ACY means add cycle

INTERPOLATION

Field Selector Circuit



INTERPOLATION

Digit Circuits

2nd Reading: Parameter a → Comparing Entry (1)

1st Card X → PU Pi Sel 1

3rd Reading: Argument x → CTR 4 (5-1) Entry

→ Co Sel 8T (5-1)

Parameter a → Comparing Entry (2)

→ CTR 2 Entry

Function y → Field Sel C (6-1)

Field Sel 1 (6-1) → Channel B (6-1)

Channel B (6-1) → FS1, 3 Entry

→ CTR 6 (6-1) Entry

→ CTR 5 (6-1) Entry

Digit Emitter (h) → Co Sel 8N (4)

Co Sel 8C (5-1) → CTR 3 (5-1) Entry

GS2 (5-1) → Channel C (5-1)

GS4 (5-1) → CTR 5 (5-1) Entry

INTERPOLATION
Differencing Sequence

No.	Sup.	Factor Storage			Mult. Quot.	Ctr.	General Storage		
		1(6-4) 3	2	4			1(6-4) 3	2	4
R		y_2	$\Delta^{(2)}_{-1}$	$\Delta^{(2)}_{-2}$	$\Delta^{(3)}_{-3}$	$\Delta^{(3)}_{-2}$	y_1	$\Delta^{(1)}_{o}$	$\Delta^{(1)}_{-1}$
1	1		RO	RI $\Delta^{(2)}_{-1}$					
4	3				RI $\Delta^{(3)}_{-2}$	RO, RC ∿∿∿			
5	1							RO	RI $\Delta^{(1)}_{o}$
10	3	RO				RI(+) y_2			
11	1					RI (-) $\Delta^{(1)}_{1}$	RO		
12	1	RO					RI y_2		
13	1					RO		RI $\Delta^{(1)}_{1}$	
15	3					RI(-) $\Delta^{(2)}_{o}$			RO
16	1		RI $\Delta^{(2)}_{o}$			RO			
18	3			RO		RI(-) $\Delta^{(3)}_{-1}$			
P		y_2	$\Delta^{(2)}_{o}$	$\Delta^{(2)}_{-1}$	$\Delta^{(3)}_{-2}$	$\Delta^{(3)}_{-1}$	y_2	$\Delta^{(1)}_{1}$	$\Delta^{(1)}_{o}$

FORWARD INTERPOLATION

Interpolation Sequence

No.	Sup.	Factor Storage			Mult. Quot.	Ctr.	General Storage		
		1(6-4) 3	2	4			1(6-4) 3	2	4
R		$x-x_o-2h$	$\Delta_o^{(2)}$	$\Delta_{-1}^{(2)}$	$\Delta_{-2}^{(3)}$	$\Delta_{-1}^{(3)}$	y_2	$\Delta_1^{(1)}$	$\Delta_o^{(1)}$
2	2					RO		RI $\Delta_{-1}^{(3)}$	
3	2				RO	RI(-) $24a_4$			
4	3		RI $24\,a_4$			RO,RC 〰			
6	2				RO	RI(+) $\Delta_{-2}^{(3)}$			
7	2					RI(+) $12a_3$		RO	
8	2					RO, RC 〰		RI $12a_3$	
9	2	(EMITTER 2h)				RI(+) 4 2h			
10	3	RO				RI(+) $x-x_o$			
14	2	RI $x-x_o$				RO, RC 〰			
15	3					RI(+) $\Delta_o^{(1)}$			RO
17	2					RI(+) $2\Delta_o^{(1)}$			RO
18	3			RO		RI(-) $2(a_1+a_3)$			
19	2					RO, RC 〰			RI $2(a_1+a_3)$
20	2	(EMITTER 5)			RI4 5				
21	2		RO			MULT(-) $-12a_4$			

(continued on next page)

		Factor Storage			Mult.		General Storage		
No.	Sup.	1(6-4) 3	2	4	Quot.	Ctr.	1(6-4) 3	2	4
22	2		RI $-12a_4$			RO5			
23	2	(EMITTER 6)			RI5 6				
24	2			RO		MULT(+) $12\ a_2$			
25	2			RI $12\ a_2$		RO5, RC			
26	2					MULT(+) $12(a_1+ a_3)$			RO
27	2					RI(-)5 $12\ a_1$		RO	
28	2					RO5, RC	RI $12\ a_1$		
29	2	(EMITTER h)							RI4 h
30	2	RO				RI(+)5 $x-x_o$			
31	2				DIV u				RO
32	2					RO, RC			
34	4					RI(-) 5 $-12\ a_3$		RO	
35	4		RO			MULT(+) $-12(a_3+$ $ua_4)$			
36	4					RO4,RC			RI $-12(a_3+ u\ a_4)$
37	4			RO		RI(-)6 $-12\ a_2$			
38	4					MULT(+) $-12(a_2+a_3u$ $+ a_4^2\ u^2)$			RO

(continued on next page)

		Factor Storage			Mult.		General Storage		
No.	Sup.	1(6-4) 3	2	4	Quot.	Ctr.	1(6-4) 3	2	4
39	4					RO5, RC 〰			RI $-12(a_2 + a_3u + a_4u^2)$
40	4					RI(-) 6 $-12\ a_1$	RO		
41	4					MULT (+) $\frac{-12(y-y_o)}{u}$			RO
42	4					RO5, RC 〰	RI $\frac{-12(y-y_o)}{u}$		
43	4					MULT(-) $12(y-y_o)$	RO		
46	4					RO6, RC 〰	RI $12(y-y_o)$		
51	4	(EMITTER 2)				RI(+) 2			
54	4	(EMITTER 1)				RI(+) 2 12			
57	4					RO, RC 〰			RI 12
58	8					RI (+) $12(y-y_o)$	RO		
59	8				DIV $y-y_o$				RO
60	8				RO			RI $y-y_o$	
P		$x-x_o$	$-12\ a_4$	$12\ a_2$	$y-y_o$		$12(y-y_o)$	$y-y_o$	12

INVERSE INTERPOLATION

Interpolation Sequence)

No.	Sup.	Factor Storage			Mult. Quot.	Ctr.	General Storage		
		1(6-4) 3	2	4			1(6-4) 3	2	4
R		$y-y_o \; -\Delta_o^{(1)}$	$\Delta_o^{(2)}$	$\Delta_{-1}^{(2)}$	$\Delta_{-2}^{(3)}$	$\Delta_{-1}^{(3)}$	y_2	$\Delta_1^{(1)}$	$\Delta_o^{(1)}$
2	2 GS1					RO		RI $\Delta_{-1}^{(3)}$	
3	2 GS1				RO	RI(-) $24\ a_4$			
4	3 GS1		RI $24\ a_4$			RO, RC			
6	2 GS1				RO	RI(+) $\Delta_2^{(3)}$			
7	2 GS1					RI(+) $12\ a_3$		RO	
8	2 GS1					RO, RC		RI $12\ a_3$	
9	2 GS1					RI(+) $-\Delta_o^{(1)}$			RO
10	3 GS1	RO				RI(+) $y-y_o$			
14	2 GS1	RI $y-y_o$				RO, RC			
15	3 GS1					RI(+) $\Delta_o^{(1)}$			RO
17	2 GS1					RI(+) $2\ \Delta_o^{(1)}$			RO
18	3 GS1			RO		RI(-) $2(a_1+ a_3)$			
19	2 GS1					RO, RC			RI $2(a_1+ a_3)$
20	2 GS1		(EMITTER 5)		RI4 5				
21	2 GS1		RO			Mult (–) $-12\ a_4$			

(continued on next page)

No.	Sup.	Factor Storage 1(6-4) 3	 2	 4	Mult. Quot.	Ctr.	General Storage 1(6-4) 3	 2	 4
22	2 GS1		RI -12 a_4			RO5			
23	2 GS1	(EMITTER 6)			RI5 6				
24	2 GS1			RO		MULT(+) 12 a_2			
25	2 GS1			RI 12 a_2		RO5, RC 〰			

26	2
	GS1
						MULT(+)
						12(a_1 + a_3)
									RO

27	2 GS1					RI(-) 5 12 a_1		RO	
28	2 GS1					RO5, RC 〰	RI 12 a_1		
29	2 GS1	RO				MULT(+) 6 (y - y_o)			

30	2
	GS1
		RO				MULT(+)
						12(y - y_o)			

31	2 GS1				DIV u		RO		
32	2 GS1					RO, RC 〰			
33	2 GS1	RI u_o			RO				
34	4					RI(-)5 -12 a_3		RO	
35	4		RO			MULT(+) -12(a_3+a_4u)			
36	4					RO4, RC 〰			RI -12(a_3+a_4u)
37	4			RO		RI(-)6 -12 a_2			

(continued on next page)

		Factor Storage			Mult.		General Storage		
No.	Sup.	1(6-4) 3	2	4	Quot.	Ctr.	1(6-4) 3	2	4
38	4					MULT(+) $12\,a_1F/u^2$			RO
39	4					RO5, RC 〰			RI $12\ a_1\ F/u^2$
41	4					MULT(+) $12\ a_1\ F/u$			RO
42	4					RO5, RC 〰			RI $12\ a_1\ F/u$
43	4					MULT(+) $12\ a_1\ F$			RO
44	4	(DO GS 2)			RO				RI u
45	5				DIV F		RO		
46	4					RO, RC 〰			
47	5	(PU GS 1)			RO	RI (+) F			
48	5	RO				RI (+) 2 u'			
49	5				RI u'	RO2			
50	5					RI(-) 2 $u'-u$			RO
51	4	(EMITTER 2)				RI (-) 2 $u'-u-2(10^{-4})$			
52	5	(BALANCE TEST)							
53	6 NEG BAL	(PROGRAM REPEAT, PU GS 2)							
54	4	(EMITTER 4)				RI (+) 2 $u'-u+2(10^{-4})$			

(continued on next page)

		Factor Storage					General Storage		
No.	Sup.	1(6-4) 3	2	4	Mult. Quot.	Ctr.	1(6-4) 3	2	4
55	5	(BALANCE TEST)							
56	7 POS BAL	(PROGRAM REPEAT, PU GS 2)							
57	4					RO, RC 〰			
58	8 GS2	(EMITTER h)							RI h
59	8 GS2					MULT(+) $x-x_o$			RO
60	8 GS2					RO2, RC 〰		RI $x-x_o$	
P		u_o	$-12\ a_4$	$12 a_2$	u	〰	$12\ a_1$	$x-x_o$	h

INTERNATIONAL BUSINESS MACHINES CORPORATION
590 Madison Avenue, New York 22, N. Y.

Litho in USA