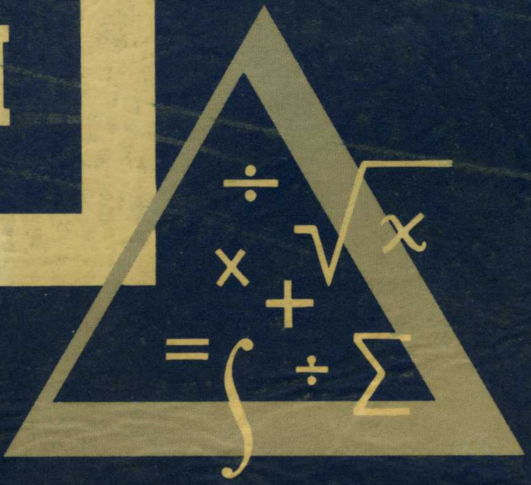
IBM

TRADEMARK



Technical Newsletter No. 7

APPLIED SCIENCE DIVISION

IBM

APPLIED SCIENCE DIVISION

Technical Newsletter No. 7

August 1954

We wish to extend our thanks to the authors of these papers for their cooperation in contributing to this interchange of technical information. Although the majority of the descriptions in this Newsletter have been in use and tested by their authors, it is not possible to provide for every contingency. Since the Applied Science Division is not able to test these procedures it will be appreciated, both by us and the authors, if any errors or discrepancies are reported to us so that we may publish corrections.

CONTENTS

SOLUTION OF SIMULTANEOUS LINEAR EQUATIONS AND COMPUTATION OF INVERSE MATRICES USING THE IBM CARD PROGRAMMED CALCULATOR

Kenneth C. Rich
U. S. Naval Ordnance Test Station
China Lake, California

Introduction

The greatly increased use of matrix methods for solving certain problems encountered in chemistry, aerodynamics, statistics and data analysis makes the development of machine methods for the solution of these problems highly desirable. The introduction of more complex computing machinery, such as the Card Programmed Calculator (CPC), which has certain definite characteristics and advantages over other IBM equipment, makes possible changes in procedure not previously incorporated in IBM methods.

P. S. Dwyer reviews approximately 14 methods for the solution of simultaneous equations and inverse matrices by direct methods.[1] The method of single division as described by Dwyer appears to have better application to CPC operations than any of the others. Indirect or iterative methods at the present time may be performed on the CPC but do not appear to be as conveniently handled by it as by a direct method. In this paper a new direct method is presented which is an elimination method adapted from those suggested by Dwyer[1] and Frank Verzuh,[2] with modifications to fit the CPC.

Form of the Matrix Equations

A set of simultaneous equations may be expressed as

$$\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \dots + a_{1n}x_n &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \dots + a_{2n}x_n &= b_2 \\
\dots \\
\dots \\
a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \dots + a_{nn}x_n &= b_n
\end{aligned}$$

or expressed in matrix notation as

$$AX = B$$

[1] Dwyer, P. S. "The Solution of Simultaneous Equations", PSYCHOMETRIKA, Vol. 6, No. 2 (April 1941), pp. 101-29.

[2] Verzuh, Frank. "The Solution of Simultaneous Equations With the Aid of the 602A Calculating Punch," MATH TABLES, July 1949, pp. 453-62.

By direct elimination methods we may solve for x_n, and by back substitution solve for $x_{n-1}, \ldots, x_2, x_1$. Various existing IBM procedures use this scheme of elimination. However, if we introduce the composite matrix[2,3]

$$\left[\begin{array}{c:c} A & B \\ \hdashline -I & O \end{array}\right]$$

the elements of which are:

$$\begin{array}{llllll}
a_{11} & a_{12} & a_{13} & \ldots & a_{1n} & b_1 \\
a_{21} & a_{22} & a_{23} & \ldots & a_{2n} & b_2 \\
\cdot & & & & & \\
\cdot & & & & & \\
\cdot & & & & & \\
a_{n1} & a_{n2} & a_{n3} & \ldots & a_{nn} & b_n \\
-1 & 0 & 0 & \ldots & 0 & 0 \\
0 & -1 & 0 & \ldots & 0 & 0 \\
\cdot & & & & & \\
\cdot & & & & & \\
\cdot & & & & & \\
0 & 0 & 0 & \ldots & -1 & 0
\end{array}$$

then by the elimination process we obtain a matrix of the form

$$\left[\begin{array}{llllll}
1 & h_{12} & h_{13} & \ldots & h_{1n} & h_1 \\
\cdot & & & & & \\
\cdot & & & & & \\
\cdot & & & & & \\
0 & 0 & 0 & \ldots & 1 & h_n \\
0 & 0 & 0 & \ldots & 0 & x_1 \\
0 & 0 & 0 & \ldots & 0 & x_2 \\
\cdot & & & & & \\
\cdot & & & & & \\
\cdot & & & & & \\
0 & 0 & 0 & \ldots & 0 & x_n
\end{array}\right]$$

without the necessity of making a back substitution. The advantage of this method lies in the similarity of the arithmetical operation performed for all the computations.

In a similar fashion we may compute the inverse by using in place of the (n + 1)th column a unit matrix of n rows and n columns. In matrix notation this is expressed as

[3] B may be an n-columned matrix. For the examples given, however, B is a one-columned matrix.

$$\left[\begin{array}{c:c} A & I \\ \hdashline -I & O \end{array}\right]$$

By direct elimination, the matrix

$$\left[\begin{array}{c:c} H & G \\ \hdashline O & C \end{array}\right]$$

is now generated as follows:

$$\begin{array}{ccccc:cccc}
1 & h_{12} & h_{13} & \ldots & h_{1n} & g_{1,n+1} & 0 & \ldots & 0 \\
0 & 1 & h_{23} & \ldots & h_{2n} & g_{2,n+1} & g_{2,n+2} & \ldots & 0 \\
0 & 0 & 1 & \ldots & h_{3n} & g_{3,n+1} & g_{3,n+2} & \ldots & 0 \\
\cdot & & & & & & & & \\
\cdot & & & & & & & & \\
\cdot & & & & & & & & \\
0 & 0 & 0 & \ldots & 1 & g_{n,n+1} & g_{n,n+2} & \ldots & g_{n,n+n} \\
\hdashline
0 & 0 & 0 & \ldots & 0 & c_{1,n+1} & c_{1,n+2} & \ldots & c_{1,n+n} \\
0 & 0 & 0 & \ldots & 0 & c_{2,n+1} & c_{2,n+2} & \ldots & c_{2,n+n} \\
\cdot & & & & & & & & \\
\cdot & & & & & & & & \\
\cdot & & & & & & & & \\
0 & 0 & 0 & \ldots & 0 & c_{n,n+1} & c_{n,n+2} & \ldots & c_{n,n+n}
\end{array}$$

where [C] is the inverse of [A]. This form appears to provide the quickest and most convenient method for CPC operation that we have found, proceeding as it does in a straightforward manner with identical computation procedures.

Method of Reduction

For the general case the reduction to the triangular matrix may be illustrated step by step by the following equations:

$$\begin{array}{llllllllll}
\text{Row 1} & a_{11} & a_{12} & a_{13} & a_{14} & a_{15} & \ldots & a_{1n} & b_{1} & \\
 & 1 & \dfrac{a_{12}}{a_{11}} & \dfrac{a_{13}}{a_{11}} & \dfrac{a_{14}}{a_{11}} & \dfrac{a_{15}}{a_{11}} & \ldots & \dfrac{a_{1n}}{a_{11}} & \dfrac{b_{1}}{a_{11}} & 1A
\end{array}$$

Row 2

$$\begin{array}{ccccccccl}
a_{21} & a_{22} & a_{23} & a_{24} & a_{25} & \ldots & a_{2n} & b_2 & \\[2ex]
0 & a^2_{22} & a^2_{23} & a^2_{24} & a^2_{25} & \ldots & a^2_{2n} & b^2_2 & a^2_{2j} = a_{2j} - \dfrac{a_{1j}}{a_{11}} a_{21} \\[2ex]
0 & 1 & \dfrac{a^2_{23}}{a^2_{22}} & \dfrac{a^2_{24}}{a^2_{22}} & \dfrac{a^2_{25}}{a^2_{22}} & \ldots & \dfrac{a^2_{2n}}{a^2_{22}} & \dfrac{b^2_2}{a^2_{22}} & \text{2A}
\end{array}$$

Row 3

$$\begin{array}{ccccccccl}
a_{31} & a_{32} & a_{33} & a_{34} & a_{35} & \ldots & a_{3n} & b_3 & \\[2ex]
0 & a^2_{32} & a^2_{33} & a^2_{34} & a^2_{35} & \ldots & a^2_{3n} & b^2_3 & a^2_{3j} = a_{3j} - \dfrac{a_{1j}}{a_{11}} a_{31} \\[2ex]
0 & 0 & a^3_{33} & a^3_{34} & a^3_{35} & \ldots & a^3_{3n} & b^3_3 & a^3_{3j} = a^2_{3j} - \dfrac{a^2_{2j}}{a^2_{22}} \; a_{32} \\[2ex]
0 & 0 & 1 & \dfrac{a^3_{34}}{a^3_{33}} & \dfrac{a^3_{35}}{a^3_{33}} & \ldots & \dfrac{a^3_{3n}}{a^3_{33}} & \dfrac{b^3_3}{a^3_{33}} & \text{3A}
\end{array}$$

Row 4

$$\begin{array}{ccccccccl}
a_{41} & a_{42} & a_{43} & a_{44} & a_{45} & \ldots & a_{4n} & b_4 & \\[2ex]
0 & a^2_{42} & a^2_{43} & a^2_{44} & a^2_{45} & \ldots & a^2_{4n} & b^2_4 & a^2_{4j} = a_{4j} - \dfrac{a_{1j}}{a_{11}} a_{41} \\[2ex]
0 & 0 & a^3_{43} & a^3_{44} & a^3_{45} & \ldots & a^3_{4n} & b^3_4 & a^3_{4j} = a^2_{4j} - \dfrac{a^2_{2j}}{a^2_{22}} a^2_{42} \\[2ex]
0 & 0 & 0 & a^4_{44} & a^4_{45} & \ldots & a^4_4 & b^4_4 & a^4_{4j} = a^3_{4j} - \dfrac{a^3_{3j}}{a^3_{33}} a^3_{43} \\[2ex]
0 & 0 & 0 & 1 & \dfrac{a^4_{45}}{a^4_{44}} & \ldots & \dfrac{a^4_{4n}}{a^4_{44}} & \dfrac{b^4_4}{a^4_{44}} & \text{4A}
\end{array}$$

$$\begin{array}{lccccccccl}
\text{Row n} & a_{n1} & a_{n2} & a_{n3} & a_{n4} & a_{n5} & \ldots & a_{nn} & b_n & \\
 & 0 & a^2_{n2} & a^2_{n3} & a^2_{n4} & a^2_{n5} & \ldots & a^2_{nn} & b^2_n & a^2_{nj} = a_{nj} - \frac{a_{1j}}{a_{11}} a_{n1} \\
 & 0 & 0 & a^3_{n3} & a^3_{n4} & a^3_{n5} & \ldots & a^3_{nn} & b^3_n & a^3_{nj} = a_{nj} - \frac{a^2_{2j}}{a^2_{22}} a^2_{n2} \\
 & 0 & 0 & 0 & a^4_{n4} & a^4_{n5} & \ldots & a^4_{nn} & b^4_n & a^4_{nj} = a^3_{nj} - \frac{a^3_{3j}}{a^3_{33}} a^3_{n3} \\
 & \cdot & \cdot & & & & & \cdot & \cdot & \\
 & \cdot & \cdot & & & & & \cdot & \cdot & \\
 & \cdot & \cdot & & & & & \cdot & \cdot & \\
 & 0 & 0 & 0 & 0 & 0 & \ldots & 1 & \frac{b^n_n}{a^n_{nn}} & \text{nA}
\end{array}$$

The advantage of this procedure is that rows 1A, 2A, 3A, 4A, . . . nA may be stored (up to the machine storage capacity), making it unnecessary to proceed with one elimination at a time, but practicable to operate as far as possible on one row before going on to the next. Card handling is thus reduced to a minimum, and elements of a particular row are handled only once. Since the rows of the matrix are read one at a time into the tabulator counters and since the tabulator counters accumulate, the only routine operations required by the CPC are those of multiplication, division, and transfer. Taking advantage of the form of matrix equations given above, the stored elements (1A, 2A, 3A, . . ., nA) may be read back into the CPC, and by the identical routine operations the unknowns may be computed without the back substitution.

By the above procedures, all row elements are read into the tabulator counters. When $n > 6$, operations are performed on the first portion of the row, after which the second portion of the row is read into the tabulator counters and the operation repeated. This is continued until all the elements of the row have been read into the counters. All the following rows are similarly handled. As the storage counters become loaded, intermediate results are punched out (in order) to be remerged into the following instruction deck. As an example of this procedure, the elements of a 10-by-10 inverse matrix are read into the counters first in a group of four elements, then in a group of six elements. In this way intermediate results are punched out twice (20 cards each) before the inverse is computed. The rows of a 20-by-20 inverse are split into a group of two elements, followed by three groups of six elements each. Nine distinct punching-out operations are necessary (80 cards each). The computing scheme will be illustrated in the following two examples.

Five Simultaneous Equations

The elements of rows 1-5 are read into the tabulator counters on instruction cards 1, 7, 17, 30, and 45, respectively. Card Programmed Calculator operation codes will control the computation for each instruction card. All multiplications are read in negatively to quantities in the tabulator counters. (The instruction card numbers appear in the left-hand column.)

1	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	b_1	
2-6		$\frac{a_{12}}{a_{11}} = K_2^2$	$\frac{a_{13}}{a_{11}} = K_3^2$	$\frac{a_{14}}{a_{11}} = K_4^2$	$\frac{a_{15}}{a_{11}} = K_5^2$	$\frac{b_1}{a_{11}} = K_6^2$	$K_2^2 . . . K_6^2$ to storage
7	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	b_2	
8-12		a_{22}^2	a_{23}^2	a_{24}^2	a_{25}^2	b_2^2	$a_{22}^2 = a_{22} - K_2^2 a_{21}$,etc.
13-16			$\frac{a_{23}^2}{a_{22}^2} = L_3^3$	$\frac{a_{24}^2}{a_{22}^2} = L_4^3$	$\frac{a_{25}^2}{a_{22}^2} = L_5^3$	$\frac{b_2^2}{a_{22}^2} = L_6^3$	L_3^3, . . L_6^3 to storage
17	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	b_3	
18-22		a_{32}^2	a_{33}^2	a_{34}^2	a_{35}^2	b_3^2	$a_{32}^2 = a_{32} - K_2^2 a_{31}$,etc.
23-26			a_{33}^3	a_{34}^3	a_{35}^3	b_3^3	$a_{33}^3 = a_{33}^2 - L_3^3 a_{32}^2$,etc.
27-29				$\frac{a_{34}^3}{a_{33}^3} = M_4^4$	$\frac{a_{35}^3}{a_{33}^3} = M_5^4$	$\frac{b_3^3}{a_{33}^3} = M_6^4$	$M_4^4 . . . M_6^4$ to storage
30	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}	b_4	
31-35		a_{42}^2	a_{43}^2	a_{44}^2	a_{45}^2	b_4^2	$a_{42}^2 = a_{42} - K_2^2 a_{41}$, etc.
36-39			a_{43}^3	a_{44}^3	a_{45}^3	b_4^3	$a_{43}^3 = a_{43}^2 - L_3^3 a_{42}^2$, etc.
40-42				a_{44}^4	a_{45}^4	b_4^4	$a_{44}^4 = a_{44}^3 - M_4^4 a_{43}^3$, etc.
43-44					$\frac{a_{45}^4}{a_{44}^4} = N_5^5$	$\frac{b_4^4}{a_{44}^4} = N_6^5$	N_5^5, N_6^5 to storage

45	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}	b_5	
46-50		a_{52}^2	a_{53}^2	a_{54}^2	a_{55}^2	b_5^2	$a_{52}^2 = a_{52} - K_2^2 a_{51}$, etc.
51-54			a_{53}^3	a_{54}^3	a_{55}^3	b_5^3	$a_{53}^3 = a_{53}^2 - L_3^3 a_{52}^2$, etc.
55-57				a_{54}^4	a_{55}^4	b_5^4	$a_{54}^4 = a_{54}^3 - M_4^4 a_{53}^3$, etc.
58-59					a_{55}^5	b_5^5	$a_{55}^5 = a_{55}^4 - N_5^5 a_{54}^4$, etc.
60						$\frac{b_5^5}{a_{55}^5} = O_6^6$	O_6^6 to storage

Note that down to this point we have triangularized the original matrix and have stored the elements of the resulting matrix. Here also we could begin the back substitution, since O_6^6 is recognized as a solution. We prefer, however, to continue with the same procedure and operations, in effect reducing the triangular matrix to the diagonal form.

61-65	K_2^2	K_3^2	K_4^2	K_5^2	K_6^2			from storage to CTR
66-69		K_3^3	K_4^3	K_5^3	K_6^3			$K_3^3 = K_3^2 - K_2^2 L_3^3$, etc.
70-72			K_4^4	K_5^4	K_6^4			$K_4^4 = K_4^3 - K_3^3 M_4^4$, etc.
73-74				K_5^5	K_6^5			$K_5^5 = K_5^4 - K_4^4 N_5^5$, etc.
75-77					K_6^6	=	X_1	$K_6^6 = K_6^5 - K_5^5 O_6^6$
78-81		L_3^3	L_4^3	L_5^3	L_6^3			from storage to CTR
82-84			L_4^4	L_5^4	L_6^4			$L_4^4 = L_4^3 - L_3^3 M_4^4$, etc.
85-86				L_5^5	L_6^5			$L_5^5 = L_5^4 - L_4^4 N_5^5$, etc.
87-89					L_6^6	=	X_2	$L_6^6 = L_6^5 - L_5^5 O_6^6$
90-92			M_4^4	M_5^4	M_6^4			from storage to CTR
93-94				M_5^5	M_6^5			$M_5^5 = M_5^4 - M_4^4 N_5^5$, etc.
95-97					M_6^6	=	X_3	$M_6^6 = M_6^5 - M_5^5 O_6^6$
98-99				N_5^5	N_6^5			from storage to CTR
100-103					N_6^6	=	X_4	$N_6^6 = N_6^5 - N_5^5 O_6^6$
104-105					O_6^6	=	X_5	

The entire procedure, from the time the first-row elements are read into the machine, takes approximately 45 seconds.

Inversion of Fifth-Order Matrix

The corresponding inverse, being, in a way, five sets of simultaneous equations, may be computed similarly, but at a considerable savings in time over five sets of five simultaneous equations. Instead of 102 instruction cards as needed by a set of simultaneous equations, the inverse may be computed with 160 cards, in a time of approximately $1\frac{1}{4}$ minutes, by using the procedure indicated below in which twenty-five storage positions are required. (The instruction card numbers appear in the left-hand column.)

1	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	1				
2-6		K^2_2	K^2_3	K^2_4	K^2_5	K^2_6				
7	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}					
8-12		a^2_{22}	a^2_{23}	a^2_{24}	a^2_{25}	a^2_{26}	1			
13-17			L^3_3	L^3_4	L^3_5	L^3_6	L^3_7			
18	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}					
19-23		a^2_{32}	a^2_{33}	a^2_{34}	a^2_{35}	a^2_{36}				
24-28			a^3_{33}	a^3_{34}	a^3_{35}	a^3_{36}	a^3_{37}	1		
29-33				M^4_4	M^4_5	M^4_6	M^4_7	M^4_8		
34	a_{41}	a_{42}	a_{43}	a_{44}	a_{45}					
35-39		a^2_{42}	a^2_{43}	a^2_{44}	a^2_{45}	a^2_{46}				
40-44			a^3_{43}	a^3_{44}	a^3_{45}	a^3_{46}	a^3_{47}			
45-49				a^4_{44}	a^4_{45}	a^4_{46}	a^4_{47}	a^4_{48}	1	
50-54					N^5_5	N^5_6	N^5_7	N^5_8	N^5_9	
55	a_{51}	a_{52}	a_{53}	a_{54}	a_{55}					
56-60		a^2_{52}	a^2_{53}	a^2_{54}	a^2_{55}	a^2_{56}				
61-65			a^3_{53}	a^3_{54}	a^3_{55}	a^3_{56}	a^3_{57}			
66-70				a^4_{54}	a^4_{55}	a^4_{56}	a^4_{57}	a^4_{58}		
71-75					a^5_{55}	a^5_{56}	a^5_{57}	a^5_{58}	a^5_{59}	1
76-80						O^6_6	O^6_7	O^6_8	O^6_9	O^6_{10}

To this point we have again triangularized the matrix, this time in the form of a diagonal band through the composite matrix. The inverse might be found by employing back substitution. Again, however, it is preferable to continue the routine operations as before and in this way to find the inverse.

81-85	K_2^2	K_3^2	K_4^2	K_5^2	K_6^2					
86-90		K_3^3	K_4^3	K_5^3	K_6^3	K_7^3				
91-95			K_4^4	K_5^4	K_6^4	K_7^4	K_8^4			
96-100				K_5^5	K_6^5	K_7^5	K_8^5	K_9^5		
101-106					K_6^6	K_7^6	K_8^6	K_9^6	K_{10}^6	(1)
107-111		L_3^3	L_4^3	L_5^3	L_6^3	L_7^3				
112-116			L_4^4	L_5^4	L_6^4	L_7^4	L_8^4			
117-121				L_5^5	L_6^5	L_7^5	L_8^5	L_9^5		
122-127					L_6^6	L_7^6	L_8^6	L_9^6	L_{10}^6	(2)
128-132			M_4^4	M_5^4	M_6^4	M_7^4	M_8^4			
133-137				M_5^5	M_6^5	M_7^5	M_8^5	M_9^5		
138-142					M_6^6	M_7^6	M_8^6	M_9^6	M_{10}^6	(3)
143-148				N_5^5	N_6^5	N_7^5	N_8^5	N_9^5		
149-154					N_6^6	N_7^6	N_8^6	N_9^6	N_{10}^6	(4)
155-160					O_6^6	O_7^6	O_8^6	O_9^6	O_{10}^6	(5)

The elements of the inverse, (1) to (5), are tabulated and summary-punched. Occasionally the diagonal element of a nonsingular matrix becomes very small, or zero. This condition is usually corrected by exchanging rows of the original matrix or by exchanging the rows after an intermediate summary-punching operation. Care should be taken not to change rows already being used in back substitution. These operations will have the effect of exchanging corresponding columns of the inverse matrix. If, by exchanging rows, the diagonal element remains zero for all combinations, the matrix is singular and no solution is possible. The presence of linear combinations of rows and columns is noted by a zero diagonal element. By inspection of the tabulation, the position of the corresponding rows and columns of the original matrix may usually be found.

The examples given are indicative of the method with no restrictions as to the matrix being symmetrical or non-symmetrical. Expansion of the procedure for matrices of higher order has been done with great saving in time over previous IBM methods. For example, by the method described in this paper,[4] a 10-by-10 inverse matrix may be computed in 11 minutes and a 20-by-20 inverse matrix may be computed in 90 minutes. Another IBM procedure using the CPC[5] requires only sixteen 10-digit storage units and, in addition, carries a check column. This procedure computes the 20-by-20 inverse matrix in two hours by use of Crout's method. This method has previously been judged the quickest IBM method available. The procedure described in this paper, however, has an advantage in that no additional equipment is required, such as reproducer, collator, or sorter. Hand merging of summary-punched cards behind marker cards has been found satisfactory in most cases and will not delay the computation time. Larger matrices require automatic merging of the cards. The use of forty-eight 10-digit storage units facilitates the reduction, but extra storage is not a necessary requirement. Changes in procedure make possible the same type of reduction using sixteen 10-digit storage units.

The value of the determinant of the coefficients, if desired, is determined quite simply by obtaining the product of the leading or diagonal terms of each transformed set. These are expressed as

$$\Delta = A_{11} \times A^{2}_{22} \times A^{3}_{33} \times A^{4}_{44} \;.\;.\;.\; A^{n}_{nn}$$

Checks and Improvement of Accuracy

The question of a check on the computations naturally arises. A check row or a check column may be carried along at all stages of computations. The sum of the new row elements or the sum of the column elements is carried as new columns or new rows, respectively. The operations performed on these are identical to the ones performed on the rows or columns of the matrix.

[4] A modification in the method of diagonalizing the triangular matrix has been suggested by B. G. Oldfield, one of the reviewers of this report. By making greater use of the Type 604 Electronic Calculating Punch for the accumulation of products, B - AC, the following saving in time may be made for the inverse:

$$\text{Number of cards saved} = (n - 1)^2$$

where n is the order of the matrix. This saving in time is appreciable with large matrices. The cards saved by using the same procedure for a set of simultaneous equations will be

$$\text{Number of cards saved} = \sum_{k=1}^{n} (n - k) + (2n - 2)$$

where n is the number of equations in the set.

[5] Chancellor, J., and others. "The Solution of Simultaneous Linear Equations Using the IBM Card Programmed Electronic Calculator." Proceedings, Industrial Computation Seminar (September 1950), International Business Machines Corporation, New York.

A check of the accuracy of final approximate results (X_0) may be made by computing the residuals[6] from the equation

$$\delta = b - AX_0$$

A second approximation, $X_0 + X_1$, may then be obtained by finding X_1 from

$$AX_1 = \delta$$

Similarly, a check of the inverse as well as a more accurate inverse may be obtained by using the first approximation, X_0, to obtain $X_0 + X_1$ from the equation

$$X_1 = X_0 (I - AX_0)$$

The latter two methods of checking and improving the accuracy require a large amount of computation which may be performed on the CPC but which is not conveniently handled. Therefore, checks of the computations are made by using the sum of rows or the sum of columns. The sum of the column elements carried as a new row will be in a better form for CPC computation. The addition of the check row increases the computation time very little.

The method of solving simultaneous linear equations and the inversion of matrices described in this paper was set up specifically for use on the CPC. This procedure is believed to be the simplest and most convenient devised for IBM equipment.

ACKNOWLEDGMENT

The author wishes to acknowledge the helpful cooperation of H. E. Tillitt in the use of machines and equipment which aided in the development described in this report; B. G. Oldfield for his comments and suggestions; and D. M. Villars for valuable suggestions in relation to the statistical analyses as a result of which the method described in this report was developed.

[6] Fox, L., and others. "Notes on the Solution of Algebraic Linear Simultaneous Equations," QUART J MECH APPL MATH, Vol. 1 (June 1948), pp. 149-73.

NOTES ON THE CPC MODEL II AND COMPONENT MACHINES

Eric V. Hankam

Watson Scientific Computation Laboratory
International Business Machines Corporation

418 Pilot and Co-selectors

1. I-p.u. of Pilot and Co-selectors

   a) If p.u. between 225° and 252°, selector will transfer, only for duration of p.u. pulse.

   b) If p.u. after 252°, selector will transfer immediately and hold until 225° of the next card, program or idle cycle. Any 418 cycle starts at 330° and therefore, if a selector is p.u. after 330°, it will hold until 225° of the same cycle.

2. D-p.u. of Pilot Selectors 1-60

   Selectors hold through any intervening idle cycles in all cases below where selectors transfer until 225° (also applies to selectors 61-76).

   a) If p.u. between 225° and 235°, selector does not transfer at all.

   b) If p.u. between 235° and 252°, selector transfers immediately and holds until 252°.

   c) If p.u. between 252° and 290°, selector transfers immediately and holds until 225° (acts like I-p.u.)

   d) If p.u. between 290° and 350°, selector does not transfer at all.

   e) If p.u. after 350° and before 225°, selector transfers at 235° and holds until 225° of the next card or program cycle.

   Note: Combining c) and e) above we see that if a p.u. pulse extends from before 290° to after 350° the selector will transfer immediately and hold until 225° of the next card or program cycle.

3. D-p.u. of Pilot Selectors 61-76

   a) If p.u. between 225° and 235°, selector does not transfer at all.

   b) If p.u. between 235° and 290°, selector transfers immediately and holds until 225° (acts like I-p.u.)

   c) If p.u. between 290° and 358°, selector does not transfer at all.

   d) If p.u. after 358° and before 225°, selector transfers at 235° and holds until 225° of the next card cycle.

Note: By means of an externally pluggable "Hold" circuit for selectors 1-60, these selectors in case 2e above, can be dropped out like selectors 61-76 at 225° of the next card cycle, i.e. they will hold through any intervening program cycles.

In the above description of selector timing, as well as in the notes to follow, an attempt has been made to supplement the timing charts contained in the Principles of Operation Manuals on the CPC and the 402 by including the exact timing of all such control panel hubs not described in these charts.

4. 12-11 Pick Up Hubs

The 12-11 p.u. hub does not accept on program cycles, but the 12-9 p.u. hub will accept impulses.

5. Hold Circuits

a) X-H hubs do not accept on program cycles.

b) D-H hubs accept any impulse on card and program cycles.

c) Statement b) implies that the Hold circuits of Pilot selectors 1-60 can be activated during programming in which case selectors will transfer for the remaining program cycles and the following card cycle (like selectors 61-76).

d) The X-H and D-H hubs can be impulsed on any card cycle preceding program cycles. The Hold circuit will then be effective for any selector picked up during subsequent program cycles.

Field Selector

The Field Selector p.u. hub accepts only digit impulses; if impulsed by a "double" digit it will accept the first one, i.e. the larger digit. The selector transfers on the following card or program cycle. If Field Selector Hold is impulsed it will transfer on the following card cycle but, unlike Pilot Selector, it will not transfer throughout any intervening program cycles.

Digit Selectors, Coding Selectors and Column Splits

A coding selector can be transformed into an emitter on card and program cycles by wiring the DI impulse to the common hub of a coding selector. Coding selectors are operative on card and program cycles while Digit Selectors are operative on card cycles only. The DI impulse itself is active on card, program and idle cycles. Column splits are operative on card cycles only; at all other times the common hub is internally connected to the 0-9 hub.

418 Counters

1. Internal Connection of Counter Entries and Exits

a) If a counter is impulsed to RI+, its entry and exit are internally connected.

b) If a counter is impulsed to RI-, its entry and exit are internally connected.

c) If a counter is not impulsed to RI+ or RI-, its entry and exit are not internally connected.

d) If a counter is impulsed both to RI+ and RI-, its entry and exit are not internally connected.

2. Counter Read Out (RO) and Counter Read Out and Reset (RR)

a) Impulsing counter TOTAL will cause counter RR.

b) Impulsing counter TOTAL and RI- will cause counter RR.

c) Impulsing counter TOTAL and RI+ will cause counter RO.

d) Impulsing counter TOTAL and RI+ and RI- will cause counter RO.

Combining 1. with 2., we see that in cases 2a and 2d counter entries and exits are not internally connected while in cases 2b and 2c they are connected. Thus it is possible to interchange entries and exits in any problem.

Whenever "Counter Exit Suppression" is impulsed, counter exits become inactive regardless of whether they are used as exits or entries.

Statements 1. and 2. hold for card and program cycles.

3. General Information

a) If a counter is simultaneously impulsed to RI+ and RI-, digits will not read into counter entries.

b) If an early All cycles pulse is wired to a counter entry position, the counter may not read in digits on the following cycle.

c) Zero test: If the result of an accumulation in a counter is zero, all counter positions will contain 9's. Following a counter reset operation all counter positions will contain zeros. An X-pulse can be wired through an internal path between the CI and C hubs of a counter only when all counter positions are 9's. A RI- impulse to a counter following a reset operation will change zeros to 9's.

d) Both counter exits and top counter exits emit an impulse for zero.

e) Counter exits will read out the digit in each counter wheel and continue emitting all subsequent digits until zero is reached, e.g. a "3" will read out as 3, 2, 1, 0.

f) Two or more counters should not be reset on the same cycle whenever their counter exits are split wired.

g) If Counter Exit Suppression is impulsed by any digit (or 10-pulse) it will suppress the read out of all subsequent digits; e.g. if a 6-pulse is used, digits 5, 4, 3, ... will be suppressed.

418 Program Unit

1. Special Program

Whenever "Program Start" is impulsed, the feeding of cards is suspended and the machine goes through a series of program cycles until "Program Stop" is impulsed. The Channel entry, Minor, Intermediate, Major and Program 4 hubs form a 7 position, 4-way "progressive" selector operative during special programming in the following manner: The seven Channel entry hubs are internally connected to the corresponding Minor hubs on the first program step, to the Intermediate hubs on the second program step, to the Major hubs on the third program step and to the Program 4 hubs on the fourth program step. This internal connection is then repeated for the next four steps, etc. until "Program Stop" is impulsed. The internal connection lasts from 225° to 202° so that practically every impulse except 12-pulses can be wired through.

The "program start" hubs (MI, IN or MA) should not be impulsed during Early All cycles time. The "program stop" hubs (MI, IN, MA) will accept from 320° on, but should not be impulsed later than 8-time. Programming can be started by impulsing any of the three (MI, IN or MA) program start hubs. Programming can then be stopped by impulsing a program stop hub of the corresponding or "higher" order.

2. Program Couple

Split wiring two or more adjacent "program couple" hubs will make corresponding program steps operative simultaneously.

Split wiring two non-adjacent "program couple" hubs will make the two corresponding program steps operative simultaneously, and will skip the program steps in between.

(Note the similarity with the 602-A program unit)

If program steps are repeated by means of the special program unit, the four program couple hubs cannot be splitwired arbitrarily. They can be "coupled" in the following combinations:

1, 2
2, 3
1, 2, 3
1, 3, 4
1, 2, 3, 4
none coupled

The remaining combinations fail:

3, 4
1, 3
1, 4
2, 4
2, 3, 4
1, 2, 4

Run-in and Run-out Operations

1. Last Card Auto Total Switch

a) This switch, if on, causes a Major program start on the run-in and run-out of cards. Furthermore, it suspends the emission of impulses from the comparing exits for any purpose, but it will not suspend programming unless program start is impulsed from comparing exits.

b) Whenever the switch is on, and the special program feature is used, programming must be stopped by impulsing the "Major Program Stop" because the machine would otherwise never stop programming on the run-in when a major program step is started by the switch.

2. LC Hub

a) The LC hub emits a pulse (162° to 217°) on the run-in, i.e. when the first card is at the 2nd Reading Station, and on the run-out, i.e. when the last card is at the 3rd Reading station.

b) If the Last Card Auto Total switch is on, the LC hub can be used as inlet from 320° to 290° on any card cycle. Its function is to cause a Major Program start.

3. Final Total

A counter will not convert on the run-out when its TOTAL hub is wired from Final Total. However, it can be made to convert if the TOTAL hub is wired from an All cycle instead of the Final Total through a selector picked up on the last card.

Final Total conversion will be added as standard feature on forthcoming 418's and can be installed on earlier models.

4. Printing

Printing and spacing is automatically suspended on the run-in, i.e. on all cycles prior to the card cycle of the first card at the 3rd Reading brushes.

Hammerlocks

The Hammerlock I-p.u. can be used on card and program cycles, but the D-p.u., if impulsed on a program cycle, will operate on the first card cycle after programming.

Stop Controls

There are two hubs on the 418 panel which, if impulsed, will cause the machine to stop. The "Test" hub accepts any impulse, while the "Stop" hub accepts only during Early All cycles time.

In CPC operations the "Stop" hub will stop the machine after the 605 has completed its calculation. If the "Test" hub is impulsed not later than 0-time the 605 will not start programming on that same 418 cycle, and cannot be single cycled. If the "Test" hub is impulsed at X-time the 605 will not start programming but can be single cycled. (This applies to later model machines where 605 calculate time has been advanced 5°, i.e. where calculate time extends from 195° to 280° instead of 190° to 275°.) In both cases the 605 counter will not reset if wired to do so on that 418 cycle, i.e. the counter will still contain data from the preceding calculation. If the "Test" hub is impulsed at 12-time the 605 will complete its calculation before it stops.

Summary Punching

The summary punch PU hub accepts any pulse except in the range from 330° to 352°. A summary punch (idle) cycle will occur at 330° following the impulse to the PU hub. Therefore, if the PU hub is impulsed by an All cycles, summary punching will take place right after Early All cycles time; on the other hand if a 10- or 9- pulse is used to activate the PU hub, summary punching will not take place until the end of that cycle at 330°.*

10 Pulse Timing

On some 418's the timing of pulses occurring at 10-time is different from that indicated on the CPC timing chart. The 10-common emits from 349° to 3° while the 10-hub, the * symbol and the 10 segment of the credit symbol exit emit from 348° to 2°.

Channel C Shift

The 15 position Channel C Shift Entry and the 10-position Channel C form a "shift" selector that can be used on the 418 independent of the CPC; if desired. For all operations, it is controlled by the "Shift" (pick-up) hub which will accept 1, 2, 3, 4 and 5 pulses, as well as multiple pulses 1, 5 or 2, 4 or 1, 2, 3 (each adding up to 6). Let us denote the 15 position of Channel C Shift entry by 15, 14, ... 1 from left to right. Normally, positions 15 to 6 are internally connected to the Channel C hubs. Channel C is internally connected to positions 14 to 5 if the "Shift" is picked up by a 1, to positions 13 to 4 if the "Shift" is picked up by a 2, etc. up to a shift of six positions. This internal connection lasts from 235° (following the "Shift" pick up) to 202° of the following card or program cycle.

Associated with the "Shift" hub on the 418 control panel and also under its control, is a 1-position 6-way Field selector (called Channel shift) on the 605 control panel. Normally its internal connection is from C to O; if the "Shift" is picked up by a 1, 2, ... 5 the selector will transfer to the correspondingly labeled levels during 605 calculate time. Similar to a Calculator Selector, this field selector can be used for any selective operation such as selection of program pick-up or suppression.

Channel A (or B) Card Read In

A zero wired to the left position of the Channel A (or B) control hubs will establish an internal connection between Channel A (or B) and Channel A (or B) Card Read In.

* NOTE: Digit and X pulses should not be wired to the summary punch PU hub if summary punching is to take place on two successive 418 cycles; 12 pulses can be used.

The adjacent sign hub (not to be confused with the "Channel Minus" A (or B)) provides an eleventh position of this "selector". The transfer time is from 235° following the pick up by the zero pulse to 202° of the following card or program cycle unless "Card Feed Channel Hold" is impulsed in which case it will transfer on the first card cycle following programming.

Counter Channel Control

The right (units) position of the Channel A, B and C Control hubs determines which of the 8 rows of the Counter Channel Control hubs become active. Multiple digit pulses cannot be used to activate more than one row at a time. Only the largest digit will be recognized.

605 Program Pick up

a) Each of the three program exits of a single program step emit isolated pulses. However, the same source pulse is operative for corresponding exits of series A, B and C, i.e. corresponding exits of series A, B and C emit the same impulse. Therefore, backcircuits will usually result if the same block of 3 program steps is picked up simultaneously for more than one series.

b) It is not possible to pick up the same block of 3 steps for more than one series even though mutually common steps are suppressed so that no step in that block is active on more than one series at once. The reason for this is that the suppression pulse makes the internal source pulse which supplies the program exit pulses to series A, B and C inoperative.

Program Suppression

a) Do not split wire two different suppression pulses into the same program suppression entry hub; always use selectors or filters.

b) Suppression pulses should be wired only to program suppression entry hubs. For example, a suppress on zero pulse should not be wired to a group suppression pick up. (To accomplish this, wire a program exit, which is suppressed on non zero, to the group suppression pick up.)

605 Zero Test and Negative Balance (N. B.) Test

1. N. B. Test

A N. B. test does not change the number tested in the electronic counter (EC). Such a test can be made on any program step except during multiplication or division operations. In particular, it is possible to impulse EC RI, N. B. test and EC RE on the same step.

2. Zero Test

a) A Zero test will, in general, change the number tested in the electronic counter. No other operation should be performed in the counter together with Zero test. The counter can be reset on the same step, however, since a counter reset operation does not occur until the beginning of the next program step. The shift unit can be used, and numbers to the right of the position tested will be ignored. If the number tested is positive, an internal source will enter a - 1 into the position tested. If the number tested is negative a - 1 will first be entered into the units position without carry, and then a + 1 will be entered into the position tested with carry. Thus, because of the loss of carry, negative numbers ending with a 9 in the units position will be altered differently from other negative numbers following a zero test.

All zeros are treated as positive numbers with the exception of a zero remainder following a division with a negative dividend. (If desired, it is possible to change such a - 0 remainder to + 0 in the counter simply by impulsing EC RI+ .)

b)* As seen above, an internal source emits a 1 into the position tested with the opposite sign of the number in the counter. If the tube (located at panel 1-5 U) that emits the 1 were removed, any digit besides a 1 could be emitted from the emitter on the control panel into the counter on a zero test. Also, if a number were transferred from a storage unit to the counter and zero test is impulsed, it would go into the counter with the opposite sign of the counter (regardless of its own sign). If the tube (located at panel 3-5H) that causes the units position of a negative number to be decreased by 1 (it is not an actual 1 pulse) were removed, any number except -1 could still be zero tested; but a - 1 in the units position would be recognized erroneously as a zero.

In the 1/2 Adjust circuit a tube emits a 5 into the counter with the sign of the number standing in the counter. If this tube (located at panel 1-7T) were removed, any number, regardless of its sign, could be read from a storage unit into the counter under the sign of the counter by impulsing 1/2 Adjust. This would greatly facilitate manipulations with absolute values. For a normal 1/2 Adjust operation it would be necessary to wire a 5 from the emitter.

c) It is not possible to Zero test on a program step which is suppressed on zero as a result of a previous zero test.

605 Selectors

1. Calculator Selectors

a) On CPC operations, calculator selectors can be picked up (on the 418 control panel) from 350° to 310°. The selector transfers immediately until 310°. It is essential, however, that the pick up pulse be effective before 605 calculate time (190° - 275°). Thus, a 418 X-pulse will be too late to transfer the selector before the 605 starts programming. Incidentally,

* The tube removals mentioned in this section can be handled by special arrangement with the IBM Corporation.

such a condition cannot be detected by single cycling the machine since in that case the selector will transfer before the first program step is single cycled on the 605.

b) A Program source pulse may be wired through the transferred points of a Program Repeat Selector to the coupling exit of a Calculator selector. In this case the coupling exit serves as entry to pick up the selector (it is not common with the pick up hub on the 418 control panel). The calculator selector will then pick up at the same time as the Repeat Selector and remains transferred until the end of the 605 calculate time unlike the Repeat Selector which may be dropped out at the end of any program sweep.

c) A coupling exit of a calculator-selector p. u. on the 418 control panel can be wired to the coupling exit of another calculator selector which will then transfer together with the first selector.

d) The pick up hub of a calculator selector on the 527 control panel will not accept a pulse before X-time, i. e. a Y-pulse or a 527 Pilot Selector coupling exit pulse cannot be used directly to pick up the selector. (This differs from the 521-604 machines).

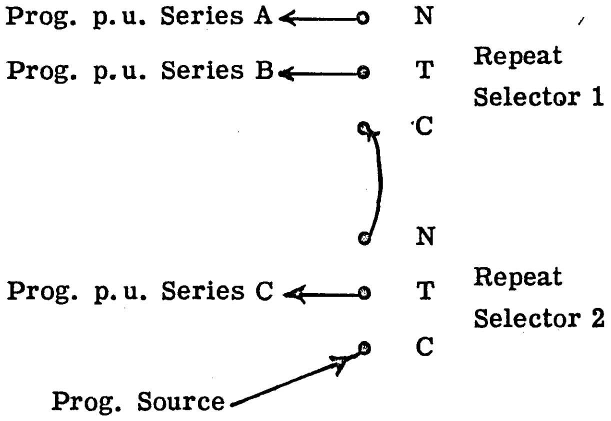
2. Program Repeat Selectors and Repeat Delay

a) A Repeat selector, once picked up, will ordinarily drop out at the end of the earliest program sweep when Repeat Delay is operative (or the end of 605 Calculate time, at the latest). However, if the selector is picked up again it will remain transferred (for at least another program sweep) regardless of whether Repeat Delay is operative or not.

b) If Repeat Delay is both picked up and dropped out on the same program sweep, the later instruction will be effective. If it is picked up and dropped out on the same program step, the outcome is indeterminate.

c) Storage assignment can be selected through a Repeat Selector. Storage assignment exits and Program source pulses are internally common, they emit pulses throughout 605 calculate time, but are broken at the end of each program sweep whenever Program Delay is operative. Calculator coupling exit impulses, on the other hand, are continuous pulses throughout calculate time and should not be wired through a Repeat Selector. (Being continuous pulses, they would arc selector points.) They can, however, be wired directly to program pick up or storage assignment entry hubs.

d) Method to obtain 180 distinct program steps:



Program Repeat, Repeat Sel. 1 p.u., and Repeat Delay p.u. are impulsed by series A program exits. (Any of the 60 steps can be used and all three instructions can be wired from the same step if desired.)

Program Repeat, and Repeat sel. 2 p.u. are impulsed by series B program exits.

Thus, the machine will go through three sweeps A, B and C in that order. If program repeat is impulsed again on series C, the machine will repeat the 180 steps until any of the Program Repeat steps is suppressed. It is possible to obtain more than 180 distinct steps by picking up program repeat selectors (to which Calculator selectors can be "coupled" as described in the section under Calculator selectors) at the end of sweep C, and by wiring program exits of series A through these selectors.

In another variation of the problem, if Repeat Delay d.o. were wired on series C, Repeat selector 2 would remain transferred, and the machine would now start with series A, B, C and then repeat C until the Program Repeat instruction on that series is suppressed.

605 Channel and Shift Unit

a) Whenever two distinct digits are transmitted along the same channel position on the same step (e.g. if two storage units are simultaneously impulsed to read out into the counter), the result will be the absolute value of their difference without carries, and a minus sign if either of the two original numbers were minus.

b) It is possible to emit a single digit from the Emitter into the 2nd, 3rd, ...6th position simultaneously by impulsing the corresponding shift unit on the same program step.

CPC Sign Control Timing

There exist two equivalent representations for negative numbers in CPC calculations. Negative numbers read out without conversion from 418 counters into the 941 will appear in complement form recognizable by a 9 in the eleventh position of a 941 storage unit. Negative numbers read out from the 605 into the 941 will enter the 941 as "true negatives", i.e., as a true figure with a 5 in the eleventh position of a 941 storage unit.

1. 418 Counter N. B. Test Exit

The N. B. test exit emits an impulse (258° - 270°) on card and program cycles, whenever the counter contains a negative number and the NB - AC switch is wired on. (If the switch is not wired, the NB hub emits only on program cycles.) A NB test exit of a counter wired to the Negative Balance Control of the same or another counter will cause a conversion (idle) cycle in the latter counter whenever its TOTAL hub is impulsed. Except for printing purposes, it is almost always desirable to eliminate 418 conversion cycles during CPC operations. For example, it will save time to read from a 418 counter without conversion to the 605. A complement read out of a 418 counter can be converted to a true figure with a sign in the process of reading it into the 605, i.e., a 605 storage unit will register the complement of each digit read into it whenever the sign hub of that storage unit is impulsed between 235° and 275°. The N. B. test exit pulse falls in that region.

2. Top Counter Sign Hub

This hub emits an impulse identical to the N. B. test exit at 258° - 270° on card and program cycles (regardless of whether the NB - AC switch is on or off).

3. 418 Credit Symbol Exit

The Credit Symbol exit of a counter emits an impulse on a card or program cycle following a conversion cycle of that counter. The NB - AC switch must be on for this hub to emit on card cycles. It also emits an impulse when a number is read into the counter negatively on a card cycle. The impulse consists of two parts: 282° to 296° and a 10 pulse.

4. Channel A (or B) - (control panel location BG-BI, 11-12, 23-24)

a) If a complement is transmitted along Channel A (or B) from the 941, the " - " hub emits from 258° - 270° and a 9 pulse.

b) If a true negative is transmitted along Channel A (or B) from the 941, the " - " hub emits from 282° - 297° and a 10 pulse.

c) If the Channel Minus A (or B) hub (control panel location AO: 55-58) has been impulsed by an X- or Y- pulse, the " - " hub will subsequently emit from 282° - 297° and a 10 pulse.

d) If the " - " hub is impulsed by a NB test exit (or Top Counter sign) it will emit a 9 pulse, i.e., in this case the " - " hub is both an inlet and outlet.

It should be noted that the main function of the 258° pulse (if wired to a 605 storage sign hub) is to transform complements to true figures upon entry into 605 storage units. The 282° pulse can be used to activate 418 counter sign reversal. The 10 and 9 pulses are useful in printing operations.

5. Channel C -

If a number is transmitted to a 941 storage unit via Channel C, and

a) the " - " hub is impulsed by an X- or Y- pulse on the previous cycle, or by a 10- pulse, a "5" will register in the eleventh position of the 941 storage unit, identifying a true negative.

b) the " - " hub is impulsed in the region from 235° - 270°, a "9" will register in the eleventh position of the 941 storage unit, identifying a complement.

The " - " hub will not accept a 282° to 297° pulse.

Digit pulses 0 to 9 must not be wired to the " - " hub.

6. FS, GS, and MQ Sign Entry

Accepts any pulse from 255° on to store a minus sign. The sign trigger itself does not turn on to register the sign until 182°, providing the corresponding storage unit is also impulsed to read in. The sign hub should always be impulsed before 605 calculate time, i.e., with an 0- pulse at the latest.* The sign hub has a dual function. As previously explained, if impulsed between 235° and 275°, it controls the subsequent read in of digits so that the storage entry will register the complement of the digits.

7. GS and EC Sign Exit

Emits at 305° - 320° and a 10 pulse whenever a negative number is read out of the corresponding units.

8. Counter Sign Reversal

Accepts any pulse from 270° to 327°.

9. Channel 9, A (or B)

Any impulse from 235° to 270° wired into Channel A (or B) -, or developed in Channel A (or B) - hub (e.g., when transmitting a complement from the 941) will make Channel 9 hubs emit 9 pulses.

Zero Test

The zero test hub on the 418 Control Panel will emit a 10 pulse following a 605 calculation if during that calculation any Zero Test indicated a non-zero condition.

Transfer Operations

a) Whenever the 418 Counter Reconversion feature is used, a counter should not be read out on two successive card or program cycles, since the reconversion cycle would coincide with the conversion cycle of the second read out. The second read out would then be in complement form.

b) A number should never be read out from the 941 on the cycle after it has been entered into the 941. At least one 418 cycle should intervene between entry and

*If the sign hub is impulsed earlier than 330° and an idle cycle occurs, the sign will not hold through that idle cycle. Thus, to preserve the sign, the sign hub should be impulsed after 330°.

read out. In "standard" CPC operation this will mean that two instruction cards must pass through the machine between read in and read out since instructions that control the reading into the 941 are frequently wired from the 3rd reading station while read out instructions are wired from the 2nd reading station.*

c) The 941 can be used as auxiliary storage for the 418 independent of the 605. The ST - INT switch does not have to be wired on in order to read into the 941. It is merely a safeguard since the machine will not run if the switch is wired and the 941 power switch has not been turned on.

d) It is possible to read one number via Channel C into more than one storage unit of the same 941 bank. This is accomplished by impulsing the units position of Channel C Control with any combination of the digits 1, 2, ... , 8 which identify the corresponding storage units of a 941 bank.

e) If the 605 is used by itself, numbers read from 527 brushes cannot be entered into a 605 storage unit unless the storage unit is impulsed to read in. On the other hand, in CPC operations, the entry of numbers into the 605 is wired on the 418 control panel and numbers can enter 605 storage units without a read in impulse. However, unless the storage unit is impulsed to read in, it will not clear, and numbers will enter cumulatively without carries.

f) If two 605 storage units, say G1 and G3, are splitwired to RO on the 605 control panel, and only one of them, say G3, is impulsed to RO on the following 418 cycle (wired on the 418 control panel), G1 would also receive an RO impulse. This would cause an error if G1 were wired to RI on the 418 control panel.

g) If a negative number is read out of the electronic counter (during a 418 cycle) and EC RR is wired from a 418 All cycles (or Card cycles) instead of an Electronic All cycles (or Electronic Card cycles), the number read out will be less by 1 in all positions.

h) Do not split wire Channel A (or B) common or 10-common to any source pulse. This may cause the erroneous entry of a number into the 941 through an internal back circuit.

527 Features During CPC Operations

a) The following hubs on the 418 control panel are common with correspondingly labeled hubs on the 527 control panel:

Calculate selectors pick-up hubs
FS, GS, and MQ entries
FS, GS, and MQ sign entries (not common on idle cycles)
FS, GS, and MQ RI are common at 10 time only
GS and EC exits
GS and EC sign exits (not common on idle cycles)

* Also, a storage unit in the 941 should not be impulsed to read in on two successive 418 cycles.

GS RO and EC RO and RR
The twelve SP Control Entry hubs on the 418 control panel are common with the 12-11 hubs of the 12 column splits on the 527 control panel.

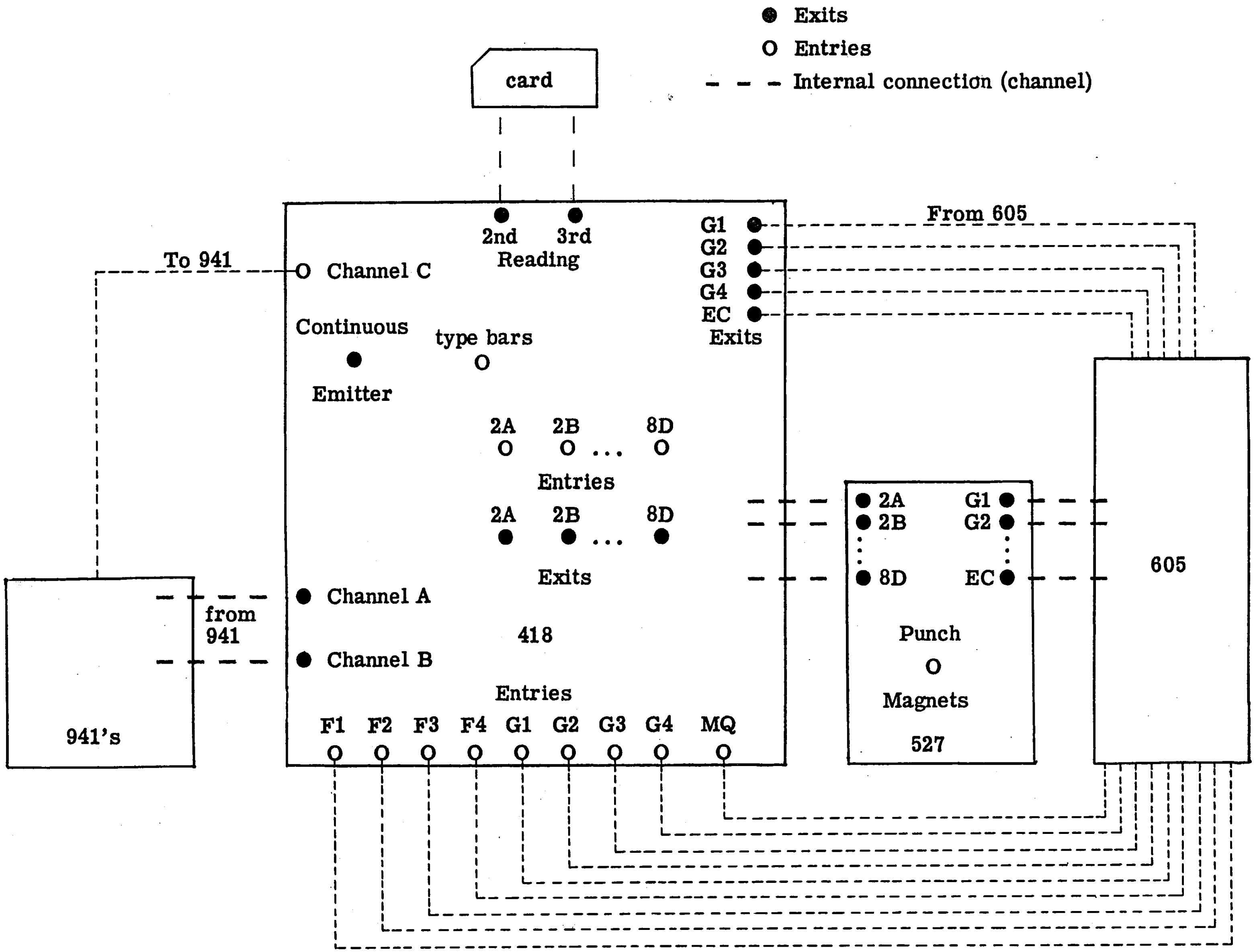
Any of these hubs (providing they are not used for their regular function) can be used as "connectors" to transmit an impulse between the 418 and 527. For example, a 527 Pilot or Punch Selector can be picked up under the control of the 418 by wiring a hot 527 pulse to the 418, through a 418 latch selector, back to the 527 and the Pilot or Punch selector pick up hub. If the hot 527 pulse is a Y-pulse, and if it is wired to the immediate pick up hub of the pilot or punch selector, summary punching of digits and X-pulses can be selected on the same cycle.

b) Calculator selectors cannot be picked up with 527 pulses. Negative Balance Selectors and Zero Check on the 527 control panel cannot be used in CPC operations. The DPBC and Overflow features are operative in CPC operations.

CPC Channels

The block diagram illustrates the principal internal channels on the CPC. Except for summary punching, all pluggable exits and entries appear on the 418 control panel. Any exit can be wired to any entry to complete a path for transmitting numbers between the various units. For example, it can be seen from the diagram that a number from the 941 via Channel A can be wired to any of the 605 storage entries, 418 counters, type bars or even via channel C to another 941 storage unit; but it cannot be wired to the 527 punch magnets directly. Another possibility, evident from the diagram, is the connection of a 605 exit to another 605 entry. Computations on the CPC frequently require the saving of a result obtained in the 605 for the following 605 calculate time. This can easily be accomplished on the 605 control panel, but the idea suggested above might save badly needed 605 program steps.

The author wishes to thank Mr. E. Lupano, IBM Customer Engineer, for his assistance in the preparation of this article.

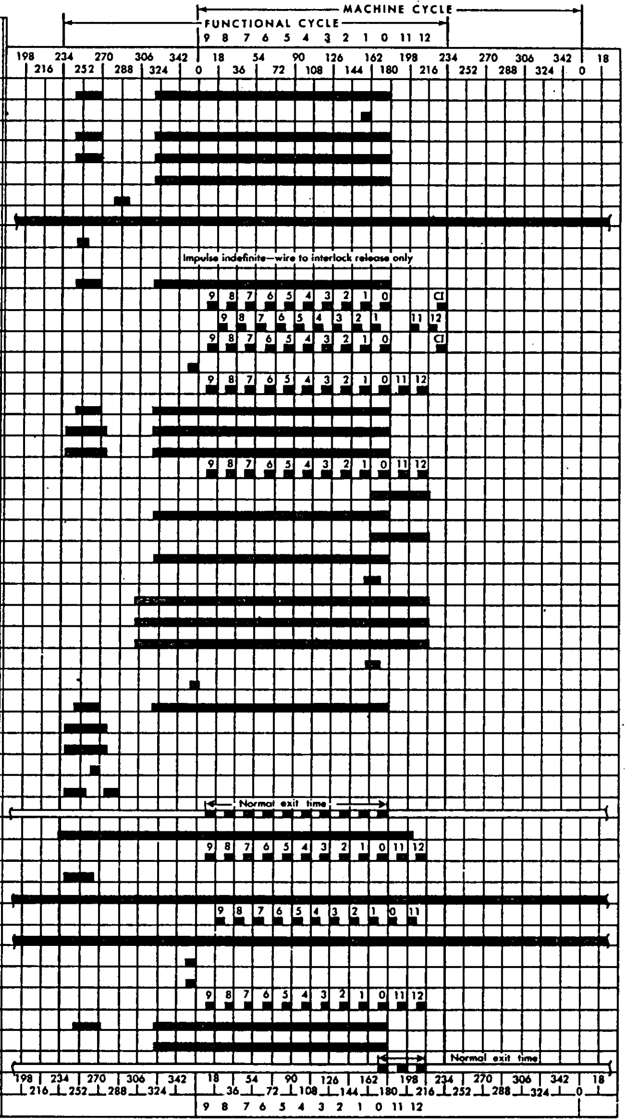


Block Diagram of Internal Channels on the CPC

ACCOUNTING MACHINES, TYPES 402 – 403

TIMING CHART

CONTROL PANEL HUB EXIT	LOCATION	MACHINE CYCLE	NOTES
All Cycles	HH,41-50	CPFO	
Card Count	L,44	C	
Card Cycles	J-N,49	C	
Card Cycles (except HC and MLP)	O-Q,49; J-Q,50	C,3	
Carriage—Indicate Control	K,47-48; L,48	O	
Interlock Suppression (Sw)	K,45	A	5
Inverted Form (Sw)	J,47	A	
Overflow	L,46	A	6
Tape Exit	M,BB,48		
Channel Entry (Spec. Prog. Off)	GG,44-50	CPFO	7
CI Carry Exit	DD,51-66	A	8
Comparing Exit	H,25-44; I,26-44	A	
Counter Exit	S-V,1-40	A	9
Credit Symbol Exit	FF,51-66	A	10
Digit Impulses	K,44	A	
Final Total	BB,41-44	F	
First Card MI,IN,MA	M,41-43	C	
First Card MB	M,44	C-H	
First Reading	A-B,1-40	C,1	*
Last Card Program Start	I,25	C	11
MLP Control—Couple P	E,48	C,2-3-4	*12
Couple Z	E,47	C,1	*
Couple Z	E,47	C,2-3	*12
Couple Z1	F,47	C,1	*
Couple P1,Z2	F,48; G,47	C,2	*
Couple P2,Z3	G,48; H,47	C,3	*
Couple P3	H,48	C,4	*
First Read Delay—Out	E-H,50	C	*13
Single Card (Sw)	I,47	C	*14
Single Card, S-M	I,49-50	C,3	*15
Negative Balance All Cycles (Sw)	BB,49	CPFO	
Negative Balance All Cycles (Sw)	BB,50	P	
Negative Balance Test Exit	AA,51-66	P	16
Pilot Selectors —— Coupling Exit	G,51-66	A	17
Print Selection Exit	J,1-24	A	*
Program Couple 1,2,3,4,All	CC-GG,42	P	
Second Reading	M-N,1-40	C,2	
Setup Change	B,43-44; C,44	A	
Special Program (Sw)	FF,43	A	
Split Column Control	T-V,41-43; T-U,44	A	
Summary Punch (Sw)	AA,49	A	18
Symbol * 1,2,3,4,All	CC-GG,41	P	
Ten (10) Impulse	Y,41-44	C	
Third Reading	O-P,DD-EE,1-40	C,3	
Total Program MI,IN,MA;Spec	CC-EE,44-50	P	
Trsfr & SP X Ctrl Plus & Minus	W-X,51-66	P	
Zone Selection Exit	D,1-24	A	*



EXIT	■
NORMAL ENTRY	▨
ENTRY	□

* TYPE 403 ONLY

NOTES

5 Except overflow cycles

6 Caused by sensing "12" hole from tape. Ineffective when sensed during a skip

7 Active only when special program device is installed

8 Except when total hub is impulsed

9 Except on run-in cycles; also, on first and third line cycles of MLP-3 cards for counters 1-24

10 On non-net balance machines, exit for card cycles only

11 First card in and last card out

12 May be used as entries for selected cycle impulses to pick up MLP selector during non-MLP operations

13 MLP control first read delay IN wired

14 Single card group when programming between first and second reading stations

15 Not active for MLP and heading cards

16 Also card cycles when NB-AC switch is ON

17 Active as an exit for every cycle that the selector is transferred

18 360° (line) impulse. May be selected under control of setup change switch. Other line impulses should not be selected

19 When C carry entry and total hubs are impulsed on the same cycle, carry does not go beyond the units position

20 Effective for overflow except for simultaneous skip to 1 or head control pickup. Action delayed for program cycles and for space suppression when impulsed from coupling exits

21 Except during summary punch cycles

22 Not to be used as entries

23 Effective for entry as an MLP-3 card passes second reading station

24 When respective counter total hub is impulsed

25 First reading station on MLP cards

26 Not receptive on MLP cards

27 MLP cards; also when couple P is impulsed on non-MLP cycles

28 MLP cards; also when couple Z is impulsed on non-MLP cycles

29 Counters 2A through 4B (1-24) not receptive when MLP-3 card is at fourth station

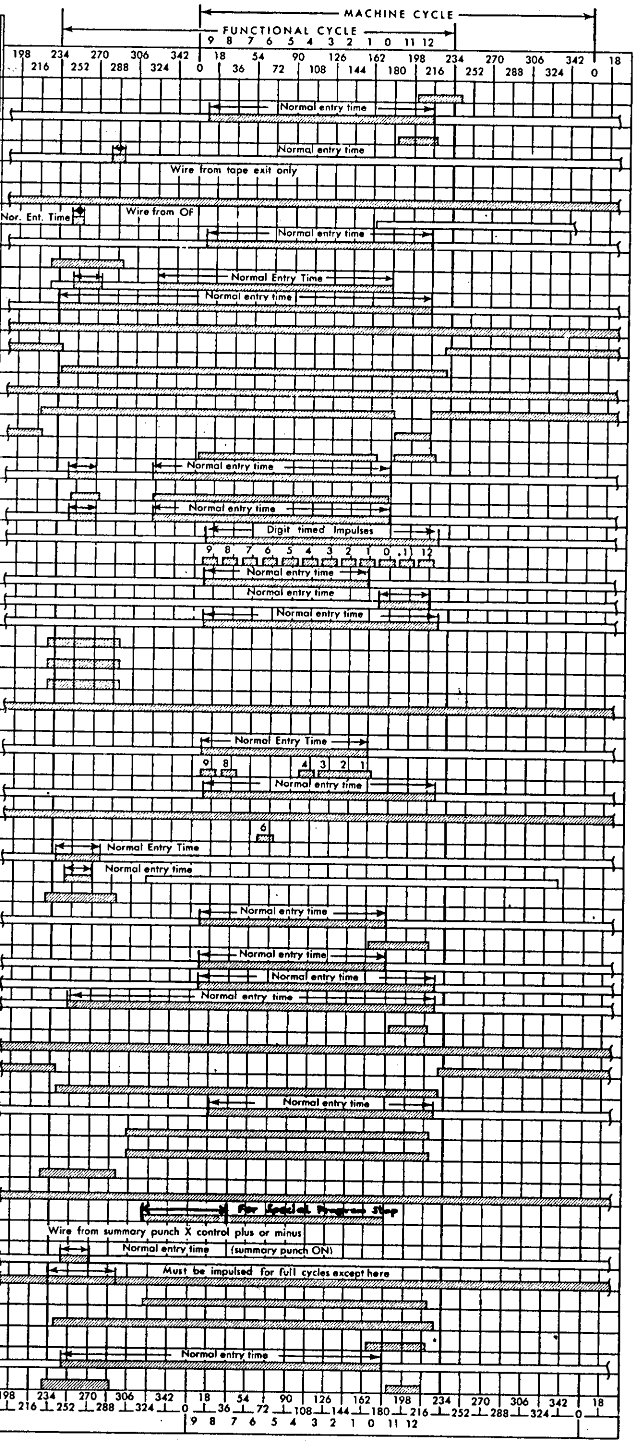
MACHINE CYCLES CODE

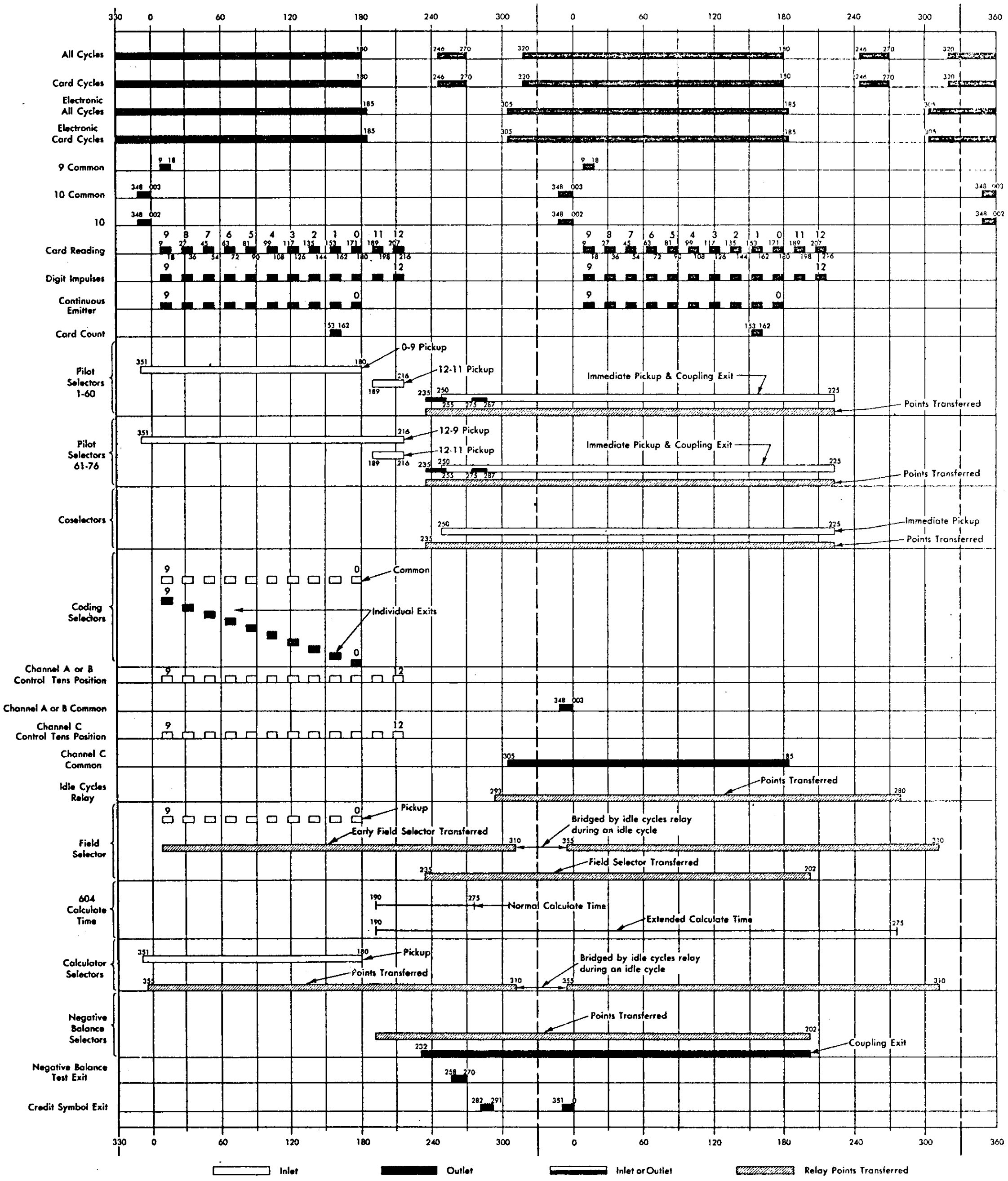
A	All cycles
A-M	All cycles except third and fourth stations MLP cards
A-Z	All cycles except second, third and fourth stations MLP cards
C	Card cycles
C-H	Card cycles except heading cycles
F	Final total cycle
O	Overflow indicate cycle
P	Program cycles
1	First reading station
2	Second reading station
3	Third reading station
4	Fourth station

ACCOUNTING MACHINES, TYPES 402 - 403

TIMING CHART

CONTROL PANEL HUB ENTRY	LOCATION	MACHINE CYCLE	NOTES
C Carry Entry	EE,51-66	A	19
Carriage Head D	I-J,46	A	
Head X	I-J,45	C	
Interlock Suppression (Sw)	K,46	A	5
Interlock Release	M-BB,47		
Inverted Form (Sw)	J,48	A	
Nȯn-Indicate – Indicate	L,45; L,47		20
Skip to D	M-BB,45	A	
Skip to I	M-BB,46	A	
Channel Entry (Spec. Prog. On)	GG,44-50	P	
Co-Selector I Pickup	A-B,58-66; C-D,58-61	A	
C	F,25-44; Y,1-40	A	
N	E,25-44; X,1-40	A	
T	D,25-44; W,1-40	A	
Column Split C	P,41-44	C	
0-9	O,41-44	C	
X-R	N,41-44	C	
Comparing Entry	G,J,25-44	A	
Counter Control Plus & Minus	S-V,51-66	A	21
Counter Entry	Z-CC,1-40	A	
Counter Exit Suppression	Y-Z,51-66	A	
Digit Selectors C	A-D,57	C	
9-12	A-D,45-56	C	
First Line Print Entry	F,1-24	A-M	*
First Line Zone Entry	E,1-24	A-Z	*
Hammerlock D	A,41-42	A	
I	A,43-44	A	
List	Z,41-42	A	
Machine Stop	S,41-43	A	
MLP Control Couple P, Couple Z	E 47-48	A	*
Couple P1, P2, P3, Z1, Z2, Z3	E-H 47-48		*22
First Read Delay—IN	E-H,49	A	*
Pickup	E,45	C	*
Program Start Delay	F-H,46	A	*
Single Card (Sw)	I,48	A	*
Successive Feed	E,46	C	*23
Negative Balance All Cycles (Sw)	BB,50	A	
Negative Balance Control	BB-CC,51-66	P	* 24
Non-Print	Z,43-44	A	
Normal Alphamerical Print Entry	Q,1-43	A	*25
Normal Zone Entry	L,1-43	C	*26
Numerical Print Entry	R,1-44; S,44	A	
Pilot Selectors D Pickup	F,51-66	A	
I Pickup	G,51-66	A	
X Pickup	E,51-66	C	
C	J-M,51-66	A	
N	I-L,51-66	A	
T	H-K,51-66	A	
Program Start Immediate	F-H,45	A	
Second Line Print Entry	H,1-24	C,3	*
Second Line Zone Entry	G,1-24	C,2	*
Space Control S-1-2-3	AA,41-44	A	
Special Program (Sw)	GG,43	A	
Stop MI-IN-MA	CC-EE,43	CPF	
Summary Punch Control Entry	R-W,49-50	A	
Summary Punch Control Entry PU	X,49	A	
Summary Punch (S w)	AA,50	A	
Third Line Zone and Print Entry	I,1-24	C,3	*
Transfer Alphamerical Print Entry	K,1-43	C,2,3,4	* 27
Transfer Zone Entry	C,1-43	C,1,2,3	* 28
Total	GG-HH,51-66	A	*29
Zone Suppress	B,41-42	A	





CPC TIMING CHART

A CARD-NUMBERING SYSTEM FOR CPC CALCULATION

William C. McGee
General Electric Company
Richland, Washington

Have you ever stood at the CPC wondering which deck of cards to feed in next? Have you ever thought to yourself, "Now, did I or didn't I change those loading cards?". Have you ever dropped a CPC instruction deck and had to start again from scratch? If you have, you would probably welcome some scheme for avoiding these situations which have a tendency to repeat themselves with annoying frequency.

With just such situations in mind, we have developed a card-numbering system for CPC calculation which has successfully eliminated almost all of the card-handling problems at the CPC.

Aside from identifying the cards and keeping them in order, you may ask why a card-numbering system is necessary at all. After all, if your calculation requires 1000 cards, why not number them consecutively from 1 to 1000? In this manner they will be positively identified and should you drop them, very easy to sort back into order.

The answer to this question is found in the concept of the loop. A loop is simply a part of a calculation which is repeated a certain number of times. Each time it is repeated, new data is supplied to give new results, unless redundancy is desired, e.g., in checking. On stored-program calculators, executing loops (i.e., getting into them, performing them, and getting out of them) is automatically taken care of by the instructions loaded in the memory. On the CPC, however, loops must be executed by means of card handling alone. This means that if a certain part of the calculation is to be repeated, say 5 times, the deck of cards performing that part of the calculation must be fed through the machine 5 times, each time with different data supplied.

Loops may vary in length. Loops which are very long (say 1000-2000 cards) are probably most efficiently executed by hand, i.e., by running the whole loop through as a unit, each time manually changing the data cards. This is not efficient for shorter loops, especially if they are to be repeated a large number of times, because of the excessive card handling it requires. In the past, shorter loops were executed by reproducing the cards in the loops as many times as the loops were to be repeated, and swapping in the data cards, either manually or on the collator. This was the right approach, but suffered from the fact that the new cards created by reproducing were not uniquely identified. If the deck were dropped, or part of it returned to the wrong place in a tray, the jig, for all practical purposes, was up.

The main purpose of this card-numbering system, then, is to identify uniquely all cards made for the purpose of executing loops in CPC calculation. It is interesting to note that if the problem involves no loops, or the loops are to be executed manually, then no card-numbering problem is involved. The cards may simply be numbered sequentially. If this situation obtains, the system to be described below reduces, as it should, to one of simple sequential numbering.

The Flow Chart

The first step in preparing a deck of instruction cards for CPC calculation under this system is to draw up a flow chart of the computation. A flow chart is simply a summary of the sequence of operations in a given computation. It serves the same purpose as a formal outline of computational procedure, but has the advantage of being concise and graphic. It is valuable for two reasons: (1) it locates each operation in the calculation, and (2) it defines each loop to be used in the calculation. Both of these features will be indispensable when we come to numbering the cards.

The flow chart is constructed from three distinct elements which we have called (1) Operations, (2) Decisions, and (3) Modifications.

Operations: An Operation is defined as anything that a card can do. Whether it is an arithmetic operation, obtaining a function, making a branch test (if such is available), modifying a variable by adding increments, punching, printing, or restoring the paper, it is called an operation if it can be accomplished by means of codes punched in a card. It may appear that this definition is somewhat broader than that usually given for a flow chart operation, in that it includes the branch test (a decision element) and a special type of modification (i.e., by adding increments). It turns out that unless we consider these special card operations as Operation elements on the flow chart, confusion will arise over their interpretation.

Decisions and Modifications: In contrast to the Operations, it might be said that Decisions and Modifications are anything that a card cannot do. Decisions must generally be made at the end of the loop; the decision will be either to go back through the loop again, or to proceed with the rest of the calculation. This decision will always be of the unconditional type, i.e., not depending on any calculated result. (Recall that conditional decisions on the CPC can be made only with the branch test device which we have agreed to consider an Operation.) Modifications are usually made after a decision for the purpose of making the next traverse of a loop unique. Again, this type of modification is the "card swapping" type, and should not be confused with the "incremental" type described under Operations above.

It is important to note that the decisions and modifications as defined here are implied in the manner in which the instruction deck is assembled. The programmer does all the deciding and modifying necessary to the calculation when he puts the deck together. The codes punched in the card have nothing to do with Decisions and Modifications.

With the elements of the flow chart defined, we are now in position to draw a flow chart for any given problem. Let us take for an example a problem in numerical integration. Suppose we wish to prepare a table of $Y(a, b_i)$ versus b_i where Y is defined as follows:

$$Y(a, b_i) = \int_0^a F(x, b_i)\, dx, \qquad i = 1, 2, 3, 4, 5.$$

Using Simpson's rule, we may write approximately

$$Y(a, b_i) = \frac{a}{30} \{ F(x_o, b_i) + 4[F(x_1, b_i) + F(x_3, b_i) + \dots + F(x_9, b_i)] + 2[F(x_2, b_i) + F(x_4, b_i) + \dots + F(x_8, b_i)] + F(x_{10}, b_i) \}$$

where we have divided the range of integration 0 to a into ten intervals. The flow chart for this computation is shown on Chart 1. To emphasize the distinction between functions performed by cards (the Operations) and by card-handling (Decisions and Modifications), we have placed all the Operations in a vertical column, and all the Decisions and Modifications to the side of this column.

Programming

The next step is to number the Operations consecutively, starting from the top. In our example, there are nine Operations. The numbered Operations are called blocks, and the numbers assigned to them are called block numbers. Each block represents a single independent part of the calculation, and may thus be programmed as a single independent unit. The programming for a block may require any number of cards, even just a single card. Each card in a block is given a sequence number to establish its position within the block. Blocks consisting of one card only are given the sequence number zero.

At this point each card programmed has a block number and a sequence number. On our CPC systems we have reserved one column (column 1) for the block number and three columns (columns 4-6) for the sequence number. This implies that we will never encounter problems with more than nine blocks, or more than 999 cards in a block. The allotment of columns for these numbers is, of course, entirely arbitrary. We could just as well have reserved two columns for the block number and two columns for the sequence number, giving us a maximum of 99 blocks, with 99 cards in each block. In fact, it is quite possible to leave this matter flexible, to be determined by the needs of each problem. The subsequent procedure will, of course, have to be modified slightly for each problem.

A third number is required now to identify the cards in the programming process. This number is called the set number and is used to distinguish cards which perform the same operation (e.g., loading or spread entering) on different sets of data. The set number is really synonymous with the "modification number" and its need is clearly signalled by the presence of Modification elements on the flow chart. The modification is understood to apply to a whole block, rather than a single card, even though many blocks requiring modification may consist of only one card. The blocks requiring modification, and therefore requiring set numbers, are those immediately following a Modification on the flow chart. Referring to our example, it is seen that Block 1, which is a load card, follows the modification element in the outermost loop and therefore requires set numbers. Since b_1 will be modified five times we assign five set numbers to Block 1. It is convenient to refer to blocks requiring modification by their Block-set number, and in our example we must prepare cards for Block-sets 101, 102, 103, 104, and 105. In our CPC system, two columns (columns 2-3) are reserved for set numbers, allowing for a possible 99 modifications of a given block. This again, is arbitrary. A majority of the blocks will not require modification, and these are given the set number zero.

Note that there is another modification in our example which we appear to have ignored, namely, the stepping of x by the increment $(a/10)$. Had we decided to step x by loading a different value of x each time, we certainly would have to set-number Block 4. But we have chosen to do this as a card operation, and purposefully omitted the modification element so that no additional set-numbering is called for.

The cards in the instruction deck programming now bear three numbers, arranged (arbitrarily) as follows:

BLOCK NO.	SET NO.	SEQUENCE NO.
1	2 3	4 5 6

A sample of the programming is shown on Chart 2. Note that the entries on the programming sheet are not necessarily in the order that the cards they represent will pass through the machine in solving the problem. Rather, they are in Block-set order. This makes it possible to group all of the information that will be used in the problem, which is extremely useful in checking back to the original data, and somewhat more aesthetically appealing than carrying this data around on the back of a card or in one's head.

Checking the Program

It is usually desirable to test the programming after it has been keypunched. Since the deck has not been expanded, this process will require careful attention from the person doing the testing. However, since the test is ordinarily not too long, the concentration span of the tester is not too protracted, and cards fed in the wrong order will not incur a serious loss. The lack of a procedure for testing the program is probably the weakest point in this system.

Expanding the Instruction Deck

When the program has been checked out the instruction deck is ready to be expanded for the production run. At this point it is necessary to introduce a fourth and final card number which we call the group number. Group numbers are required to sort the expanded deck into the order in which it will pass through the machine. They are determined with the aid of the flow chart as follows.

The first group begins at α (symbol denoting start of computation), and ends when the first modification is encountered (either a card modification or a card-handling modification). The second group begins where the path from this modification intercepts the operation column and ends when a modification is again encountered. In this manner it is possible to number all groups, up to and including the last group, which ends at β (symbol denoting the end of computation).

For example, in our problem the first group begins at α, and ends at the first modification, which happens at Block-set 400 after the first time around the inner loop. Thus Group 1 consists of Block-sets 100, 200, 300, 400, 500, and 600. Group 2 begins with Block-set 400, and ends after the second time around the inner loop. Thus Group 2 consists of Block-sets 400, 500, and 700. This process continues up to Group 10, which consists of Block-sets 400, 500, 800, and 900. This information is summarized on the right-hand side of the CPC Deck Assembly Request (Chart 3).

Strictly speaking, our group numbering is not complete, since we have not reached β. Had we done this, we would have 50 groups instead of 10 because of the 5 changes in b_1. This does not appear practical, inasmuch as it is a simple matter to change the load card at the CPC after the completion of the basic 10-group pass. In practice we have found it sufficient to number the groups up to the point where the first modification on the outermost loop is encountered. This modification may be thought of as the "major" change in the calculation. In our programming of the instruction deck we have assigned the set-number zero to Block 1, as at this point it is just a dummy. Later, at the CPC, this dummy will be replaced in turn by Block-sets 101, 102, 103, 104, and 105, which are shown programmed in the data programming.

The group numbering which has just been described is the heart of the numbering system. It is the group number which makes the cards for executing loops unique. It therefore appears that the numbering up to the group numbering, in particular the block numbering, is somewhat arbitrary. One limitation on block-numbering is obvious; namely, load blocks must bear a separate block-number, since in general they will also have set-numbers. Aside from this, it is perfectly legitimate to assign a single block number to a sequence of operations provided only that that sequence remain unbroken throughout the calculation. This is true even if the operations in the sequence are quite unrelated from the computational standpoint. Notice in our example that we could have abbreviated the block-numbering by assigning a single block number to Blocks 4 and 5, since these two always occur together. The same is true of Blocks 2 and 3, and of Blocks 8 and 9. Whether to make this sort of condensation will depend on the particular problem. Often it is desirable to include more blocks than necessary to hold the sequence numbering to a minimum. Then if a programming error is discovered, the task of renumbering will not be so prodigious.

When the group numbering is completed, the number of copies of each block-set required is entered on the left hand side of the CPC Deck Assembly Request. The operator then makes the required number by 80-80 reproducing and groups the copies according to the group numbering scheme. As each group is formed, the group number is gang punched into the cards forming that group (into columns 9-10 in our systems). When the gang punching is complete all the cards are sorted on sequence number, set number, block number, and group number. The deck is then ready for the CPC.

At the CPC

The properly assembled CPC deck is capable of standing on its own two legs, so to speak. It can (and should) be sequenced every time it is run. It can also be sequenced by the tabulator while it is being run, if the equipment is available. It can be dropped and sorted back into order. We suggest that everyone making a deck in this manner drop it at least once to assure himself that this works.

The important thing to note is that in any problem there need never be more than two types of cards at the CPC. One is the assembled instruction deck and the other may be load cards needed for making the "major change" in the calculation. By making these two decks on different type cards, e.g., corner cut or colored, the problem of CPC operation is vastly simplified.

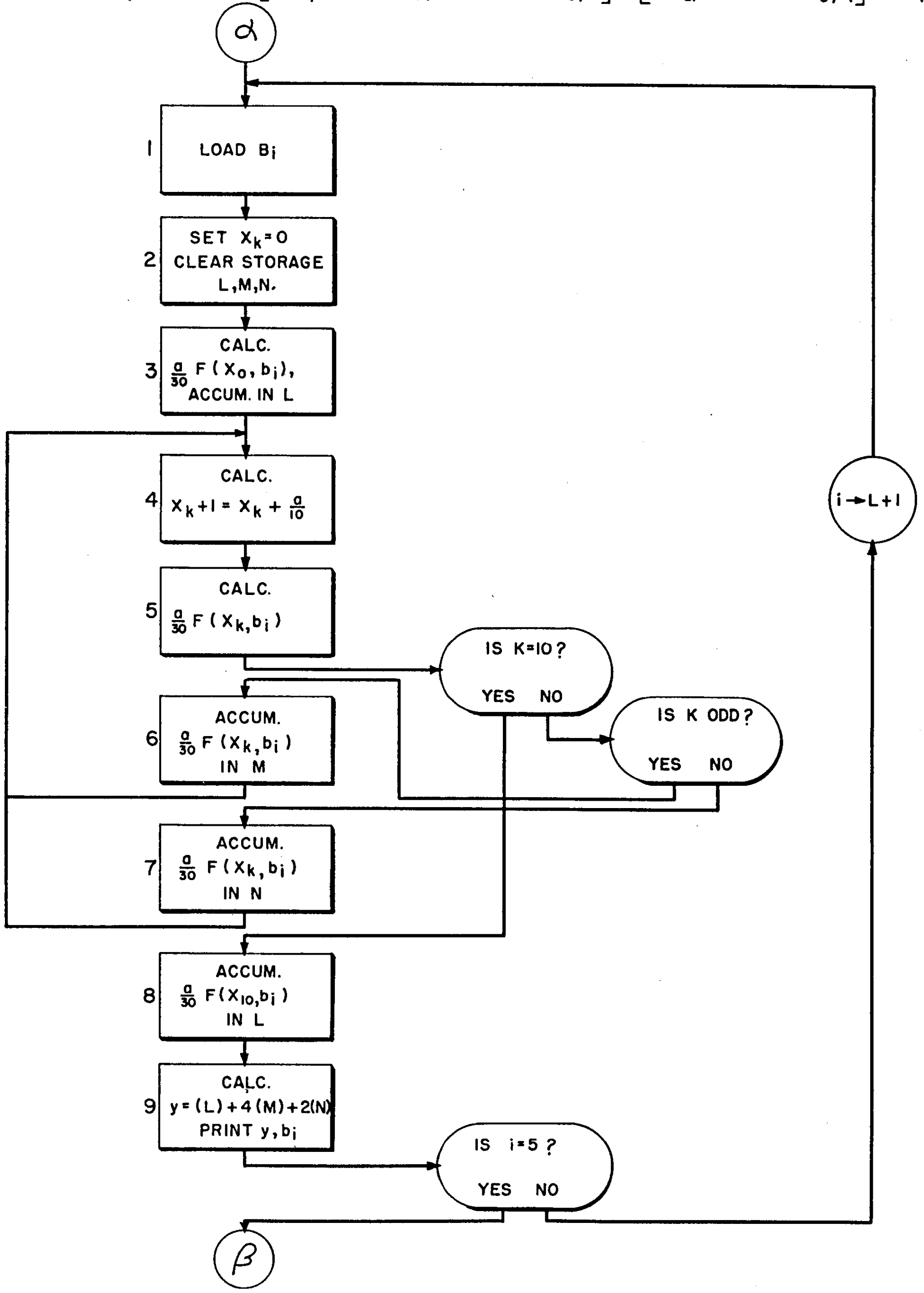
There are, of course, many variations of this system. For example, the form of the system may be dictated by the number of card-numbering columns available on the instruction card. Another variation concerns further expansion of the instruction deck. If the expanded deck is too small to run conveniently, it may be reproduced 80-80 without further numbering. Of course the cards in the deck are no longer uniquely identified, and any sorting operation must be followed by a collating operation to remove the duplicates.

Finally, there are problems which have not been worked out fully. One has already been mentioned in connection with testing the program. The other concerns numbering summary cards which are to be used in further calculation. As time passes, it is hoped that these and other problems will be solved.

FLOW CHART FOR

CHART I

$Y(a,b_i) = \frac{a}{30}\{F(X_0,b_i) + 4[F(X_1,b_i) + F(X_3,b_i) + \ldots + F(X_9,b_i)] + 2[F(X_2,b_i) + \ldots + F(X_8,b_i)] + F(X_{10},b_i)\}$



α
1 LOAD B_i
2 SET $X_k = 0$ CLEAR STORAGE L,M,N.
3 CALC. $\frac{a}{30} F(X_0, b_i)$, ACCUM. IN L
4 CALC. $X_{k+1} = X_k + \frac{a}{10}$
5 CALC. $\frac{a}{30} F(X_k, b_i)$
6 ACCUM. $\frac{a}{30} F(X_k, b_i)$ IN M
7 ACCUM. $\frac{a}{30} F(X_k, b_i)$ IN N
8 ACCUM. $\frac{a}{30} F(X_{10}, b_i)$ IN L
9 CALC. $y = (L) + 4(M) + 2(N)$ PRINT y, b_i
IS K=10? YES NO
IS K ODD? YES NO
IS i=5? YES NO
$i \rightarrow L+1$
β

CPC PROGRAMMING CHART
SEMAC II

PROBLEM NO.

TITLE CHART 2

WRITTEN BY

APPLICATION ILLUSTRATION OF NUMBERING SYSTEM Page 1

CHECKED BY

	OPERATION NOTES	CARD NUMBER	SPEC. INST.	COUNTER 1	COUNTER 2 PRIMARY INSTRUCTIONS						COUNTER 3 PRIMARY CARD ENTRY	COUNTER 4 PRIMARY CARD ENTRY	COUNTER 5 SECONDARY INSTRUCTIONS					
					SI	OP	A	B	C	SH	CHANNEL A	CHANNEL B	SI	OP	A	B	C	SH
1	Dummy	100000																
2	Load b_1	101000				11	00	00	18		7743000049							
3	Load b_2	102000				11	00	00	18		9522000049							
4	Load b_3	103000				11	00	00	18		1034000050							
5	Load b_4	104000				11	00	00	18		1248000050							
6	Load b_5	105000				11	00	00	18		1574000050							
7																		
8	$X_K = 0$	200001							14									
9	Clear L	200002							15									
10	Clear M	200003							16									
11	Clear N	200004							17									
12																		
13	↑	300001																
14		300002																
15	Calc. $\frac{a}{30}F(X_o, b_i)$	300003																
16		300004																
17		300005																
18	↓	300006																
19	Acc. $\frac{a}{30}F(X_o, b_i)$ in L	300007				11	15		15									
20																		
21	$X_{K+1} = X_K + a/10$	400000				11	00	14	14		← a/10 →							
22																		
23																		
24																		
25																		

CPC PROGRAMMING CHART
SEMAC II

PROBLEM NO.

TITLE CHART 2

APPLICATION ILLUSTRATION OF NUMBERING SYSTEM Page 2

WRITTEN BY

CHECKED BY

	OPERATION NOTES	CARD NUMBER	SPEC. INST.	COUNTER 1	COUNTER 2						COUNTER 3										COUNTER 4	COUNTER 5					
					PRIMARY INSTRUCTIONS						PRIMARY CARD ENTRY											SECONDARY INSTRUCTIONS					
					SI	OP	A	B	C	SH	CHANNEL A										CHANNEL B	SI	OP	A	B	C	SH
		1 2 3 4 5 6	7 8 9 10	11 12 13 14 15 16 17 18 19 20	21	22 23	24 25	26 27	28 29	30	31	32	33	34	35	36	37	38	39	40	41 42 43 44 45 46 47 48 49 50	51	52 53	54 55	56 57	58 59	60
1	↑	500001																									
2	│	500002																									
3	Calc. $\frac{a}{30}F(X_K, b_i)$	500003																									
4	│	500004																									
5	│	500005																									
6	↓	500006																									
7																											
8	Acc. $\frac{a}{30}F(X_K, b_i)$ in M	600000				11	16		16																		
9																											
10	Acc. $\frac{a}{30}F(X_K, b_i)$ in N	700000				11	17		17																		
11																											
12	Acc. $\frac{a}{30}F(a, b_i)$ in L	800000				11	15		15																		
13																											
14	4(M)	900001				13	00	16	16		4								5	0							
15	2(N)	900002				13	00	17	00		2								5	0							
16	2(N) + L	900003				11		15	00																		
17	2(N) + L + 4(M) = Y	900004				11		16	00																		
18	Print b_i, $y(a, b_i)$	900005	Y			11	18	00	00																		
19																											
20																											
21																											
22																											
23																											
24																											
25																											

CHART 3

CPC DECK ASSEMBLY REQUEST
REFER TO PROCEDURE 275

REQUESTED BY______________________________JOB NO.________________DATE_______________

BLOCK-SET NO.	NO. OF COPIES
101-105	1 each
200	1
300	1
400	10
500	10
600	5
700	4
800	1
900	1

GROUP NO.	COMPOSED OF BLOCK-SET NOS:
01	100,200,300,400,500,600
02	400,500, 700
03	400,500,600
04	400,500 700'
05	400,500,600
06	400,500, 700
07	400,500,600
08	400,500, 700
09	400,500,600
10	400,500, 800,900

A METHOD FOR DETERMINING THE ROOTS OF A SECULAR DETERMINANT OF ORDER m, BY MEANS OF THE IBM MODEL 602A CALCULATING PUNCH

Andrew Acrivos and Neal R. Amundson
University of Minnesota, Minneapolis 14, Minnesota

I. Mathematical Apparatus

In many applications of quantum mechanics and in a number of other problems it becomes highly desirable to be able to evaluate the roots of the secular equation $D(\lambda) = |a_{ij} - b_{ij}\lambda| = 0$, $i, j = 1, 2 \ldots m$.

To evaluate the m roots of the above secular equation, we will make use of the procedure shown in (1) and (4). This method consists of first reducing the secular determinant to a polynomial of the mth degree in λ; the roots of this polynomial are then evaluated by means of any of the standard methods, i.e., Newton's formula, Graeffe's formula, etc.

Now, to transform the secular determinant into a polynomial in λ one follows these steps:

a. Let $\lambda - \lambda_0 = 0, 1, 2 \ldots m$, thus obtaining m+1 different determinants; λ_0 is any convenient number.

b. Evaluate these m+1 determinants.

c. Then, by means of a simple procedure to be shown later, one uses these results to obtain the desired polynomial.

Step b is by far the most troublesome, since the evaluation of an mth order determinant is by no means an easy task, if m is larger than 3 or 4. It is here that the Model 602A will be found useful. But first let us see how one is to go about evaluating a determinant.

The simplest procedure consists in using the so-called "matrix inversion method", shown in detail in (2). We will give here a brief outline of this method. Consider for example the determinant which is obtained by letting λ equal zero.

$$\Delta = \begin{matrix} a_{11} & a_{12} & a_{13} \ldots & a_{1m} \\ a_{21} & a_{22} & a_{23} \ldots & a_{2m} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & a_{m3} \ldots & a_{mm} \end{matrix}$$

Step 1: Divide each element of the first row by ($-a_{11}$); then write down the matrix

$$\begin{array}{cccc}
-a_{12}/a_{11} & -a_{13}/a_{11} & \cdots & -a_{1m}/a_{11} \\
1 & 0 & & 0 \\
0 & 1 & & 0 \\
\cdot & \cdot & & \cdot \\
\cdot & \cdot & & \cdot \\
\cdot & \cdot & & 1
\end{array}$$

This matrix has m rows, but m-1 columns

Step 2: In the original determinant cross out completely the first row, thus obtaining the matrix

$$\begin{array}{ccccc}
a_{21} & a_{22} & a_{23} & \cdots & a_{2m} \\
a_{31} & a_{32} & a_{33} & \cdots & a_{3m} \\
\cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & \cdot & & \cdot \\
\cdot & \cdot & \cdot & & \cdot \\
a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mm}
\end{array}$$

This matrix has now (m-1) rows, and m columns.

Step 3: The two matrices obtained from the previous two steps are conformable, i.e., one matrix has as many rows as the other has columns; therefore the matrices can be multiplied. So Step 3 consists in premultiplying the matrix obtained in Step 1 by that formed in Step 2. The result is the following:

$$\begin{array}{cccc}
-\dfrac{a_{12}}{a_{11}} a_{21} + a_{22} & -\dfrac{a_{13}}{a_{11}} a_{21} + a_{23} & \cdots & -\dfrac{a_{1m}}{a_{11}} a_{21} + a_{2m} \\
-\dfrac{a_{12}}{a_{11}} a_{31} + a_{32} & -\dfrac{a_{13}}{a_{11}} a_{31} + a_{33} & \cdots & -\dfrac{a_{1m}}{a_{11}} a_{31} + a_{3m} \\
\cdot & \cdot & & \cdot \\
\cdot & \cdot & & \cdot \\
\cdot & \cdot & & \cdot \\
-\dfrac{a_{12}}{a_{11}} a_{m1} + a_{m2} & -\dfrac{a_{13}}{a_{11}} a_{m1} + a_{m3} & \cdots & -\dfrac{a_{1m}}{a_{11}} a_{m1} + a_{mm}
\end{array}$$

It is not difficult to see that the above is again a determinant, but of the (m-1)th order.

Step 4: Steps 1, 2 and 3 are now repeated over and over again until the original determinant is reduced down to one term, i.e., a determinant of the first order.

Step 5: The value of the original determinant is equal to the product of the left hand upper corner elements of the determinants obtained in the preceding steps, from order m down to order 1. Those are the same elements which are used, in each case, to divide the elements of the first row of each determinant, thus forming the matrices similar to the one obtained in Step 1.

Let us now consider more carefully the nature of the arithmetic operations which are carried out in each one of the above steps. Step 1 clearly involves a simple division, while Step 3 requires an operation of the form $\Sigma\, a_n \cdot b_n = P$, where the summation includes only two terms. Step 4 is obviously a repetition of the operations carried out in Steps 1 and 3, while Step 5 consists of a series of multiplications. So one sees quite readily that the evaluation of a determinant of the m^{th} order can be done by means of only two arithmetic operations which are repeated a number of times. If we are then to use the Model 602A to evaluate said determinants we must be able to construct a wiring diagram which will enable the machine to perform said two operations, viz., a division and an additive multiplication.

Before proceeding with the wiring diagrams, however, let us say a few words concerning the degree of accuracy to which each of the two operations mentioned must be carried. One must bear in mind that due to the large number of operations involved, small inaccuracies introduced during the evaluation of the determinants might possibly be the cause of a rather serious error in the final result. We actually found by experience that, for determinants up to the 5th order, carrying one's calculations to the 8th decimal place gave results that were at least within the desired degree of accuracy; i.e., the computed roots were accurate to at least the 4th decimal place. Now it will be seen that the division, as set up on the Model 602A, carries the quotient up to the 8th decimal place only, and it is hoped that this will be sufficient for the accurate evaluation of a determinant of any order m.

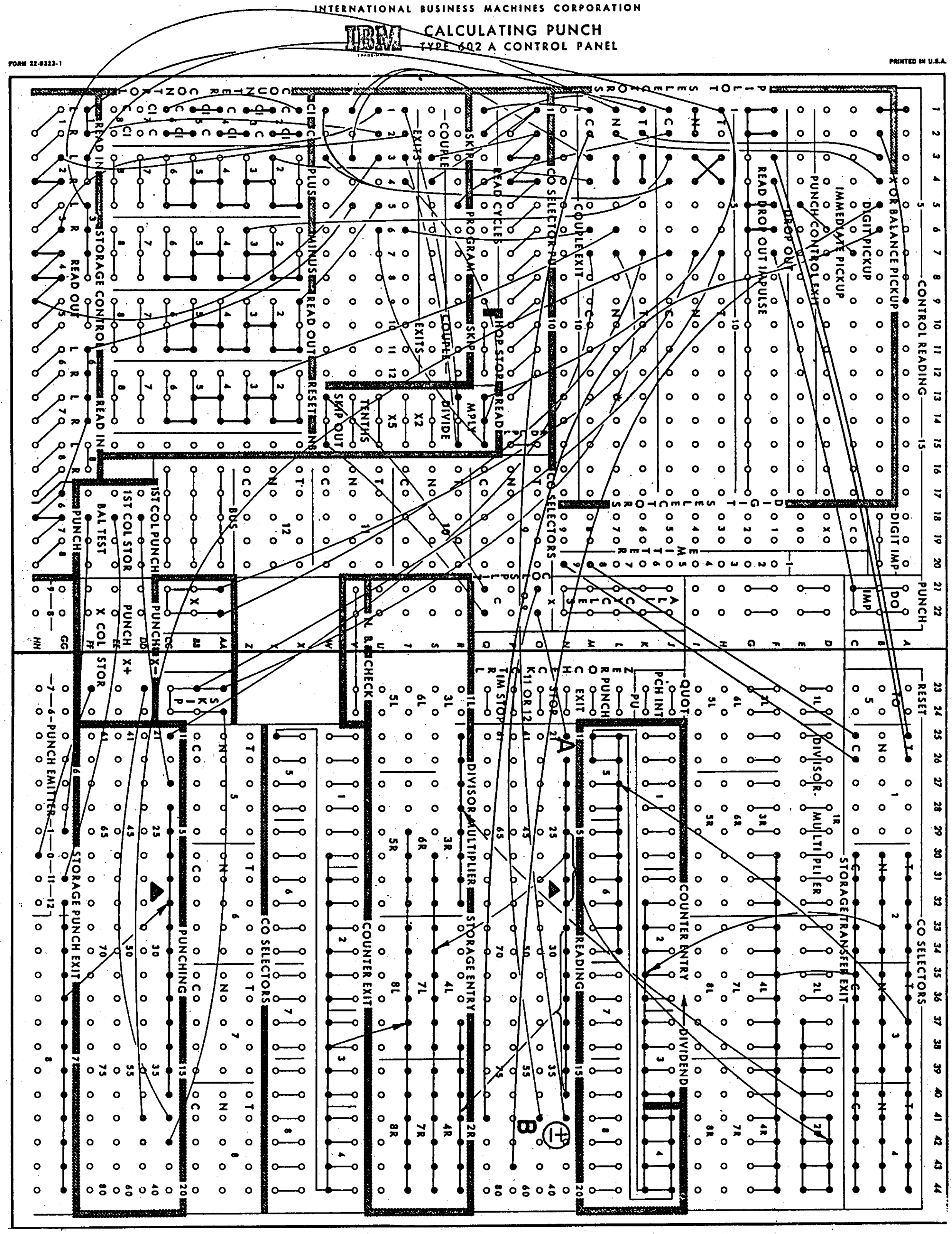
II. The operation $\sum_{i=1}^{n} a_i\, b_i = P$

We will try next to explain the wiring of diagram I which was set up to perform the above operation. Three types of cards are used throughout:

1. The a card which carries an X in the first column.
2. The b card which has an X in the 37th column.
3. The P card which is blank, except for special cases mentioned below.

The a card can have a 12-digit number punched in columns 5-16. The decimal point is between columns 7 and 8. The b card can have 15 digits punched in columns 2-16 ; the product P can have 14 digits punched in columns 3-16 and the decimal point is again between columns 7 and 8. If the numbers on a, b or P are negative, then an X must appear in column 17.

The procedure used is quite easy to follow, inasmuch as it is standard. Since a has more than 8 digits, the operation has to be performed in 2 steps, as is shown in the diagram.



Panel Board I. Additive multiplication.

Now, since the answer card P has to be used in another operation, it must be punched as either an a or a b card. If the P card carries an X in column 59, then the answer card will be an a card, otherwise it will be a b card. Provision is also made to allow the machine to produce a certain number of copies of a given result. All copies, except for the last card, to be punched after each operation must carry an X in column 77, to prevent the counters from resetting.

The calculations are carried out to the 9th decimal place.

It should be noted that the sum $\sum_{i=1}^{n} a_i b_i = P$ can contain an arbitrary number of terms. For more details, c.f. (3), pp. 17-19 and 52-53, where the simple multiplication is discussed quite extensively.

III. The Division (-b/a)

The IBM Model 602A has been constructed to perform a simple division of the above form where the divisor a has only 8 digits. We are now going to propose a scheme by which the machine could be made to divide a number b, by a number a having as many as 12 digits.

Let $a = a' + c$ where a' has the first 8 digits of a while c has the last 4. Then we can write that

$$\frac{b}{a' + c} = \frac{b}{a'}\left(\frac{1}{1 + c/a'}\right) = \frac{b}{a'}\ \left(1 - \frac{c}{a'} + (\frac{c}{a'} + \left(\frac{c}{a'}\right)^2 + \ldots\right)$$

If now $(c/a) < 10^{-4}$ and if b/a' is not much larger than 1, then to the desired degree of accuracy, i.e. to the 8th decimal place,

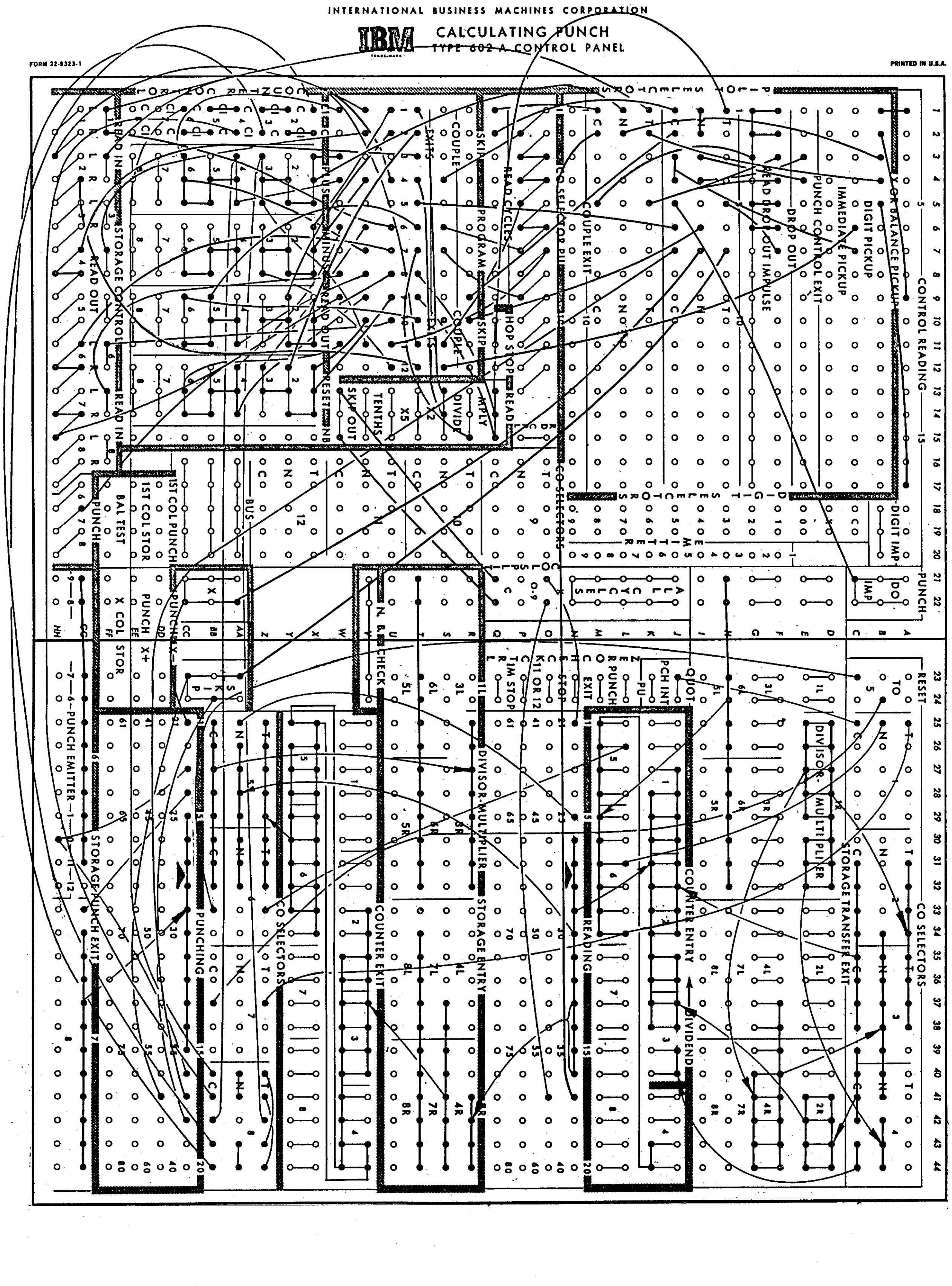
$$\frac{b}{a} = \frac{b}{a'}\left(\frac{a' - c}{a'}\right)$$

In order, then, to evaluate the quotient b/a, if a has more than 8 digits, we would have to perform 4 successive operations, viz., a simple division b/a', a subtraction of two numbers $a' - c$, another simple division $(a' - c)/a'$, and finally a multiplication. As would be expected, the wiring diagram for such an operation is quite complex, and it is for this reason that we are going to analyze it quite thoroughly in the following pages.

Planning Charts and Control Wiring for the Division - b/a

(Diagram II)

a) a' is read into Storage Entry 1R while c is read into Storage Entry 2R If a is negative, i.e. if an X is punched in column 17, then pilot selector 3 is picked up. The a card is next skipped out and the machine is ready to read the b card.



Panel Board II. Division $-\frac{b}{a}$.

b) b is read into counters 1, 2 and 3 which are connected. If b is negative then pilot selector 4 is energized.

Program Steps:

Programs 1, 2, 3 and 4 perform the division b/a' carried out to the ninth decimal place. The operation is standard; c.f. (3), pp. 54-55.

Program 5 is used to skip programs 6, 7, 8 and 9.

Program 10. The quotient b/a' is read into storage entries 6R and 6L.

Programs 11 and 12. c is subtracted from a' while at the same time pilot selector 6 is picked up. Counters 4-6 are reset to 5.

Programs 1', 2', 3' and 4' perform another division, viz. $(a' - c)/a'$. The quotient is obviously less than unity and will contain 8 digits. The last digit has been reset indirectly during Program 11.

Program 6. $(a' - c)/a'$ is read into 1R while the counters are reset.

Program 7. b/a' is multiplied by $(a' - c)/a'$. At the same time all the counters become connected by means of co-selector 8.

Program 8. The final answer is read into storage unit 7 and the machine is ready to read the next card.

c) The answer card is blank except maybe, as will be seen later, for an X punched in column 50.

Program Steps:

All program steps up to 9 are skipped.

Program 9. The answer is punched out of storage punch exit 7; the counters are reset. Since the quotient is equal to $(- b/a)$, the answer card must have an X punched in column 17 unless b and a have different signs. It is easily seen that if either pilot selector 3 or 4 is energized, then pilot selector 7 will be energized before punching and no X will be punched in column 17 of the answer card. One can, of course, reproduce any number of "copies" of the answer. It is very important, however, to note that the last answer card after each operation must have an X punched in column 50, otherwise pilot selectors 3, 4 and 7 will not drop out.

In this particular case, an X is punched during the punching in the first column of the answer card; therefore the answer card becomes an a card. Of course, one could easily change the diagram so that the answer card would become a b card.

Important Note: Let us for a moment consider one of the four operations which had to be carried out during the division $(-b/a)$, i.e. $(a' - c')/a'$. The quotient is a fraction which comprises 8 digits. Those 8 digits are then used to multiply b/a' and thus produce the final answer. Suppose now, however, that $c' \rightarrow 0$; in that case obviously $(a' - c')/a' \rightarrow 1$, and the 8 digits mentioned before turn out to be all equal to zero. If then one uses the wiring diagram unmodified, one obtains an answer equal to zero instead of b/a. It is desired to show here that the diagram can be altered very easily to perform this simple division.

So if c = 0, the wiring diagram should be changed as follows:

1. Disconnect the "Read" from program 8 and connect it to program 5.

2. Read out of 6 and into Storage Unit 6 at program 5, instead of program 10.

3. During program 9, read out of 6 and into Punch 6 instead of reading out of 7 and into Punch 7.

4. Connect column 16 in the Punching section to the first hub of Storage Punch Exit 6, and the other hubs of Storage Punch Exit 6 to Punching Columns 5-15. Connect column 4 to the 0 punch emitter.

The machine then will give the answer of (-b/a) to the ninth decimal place.

Limitations: The quotient (- b/a) will be correct to the 8th decimal figure if:

1. b/a' is not much larger than one.
2. If $(c/a') < 10^{-4}$.

Also a can have at most 12 significant digits (9 decimal places) and at least 9 significant digits. The wiring diagram must then be altered if it is to serve other more specific cases.

IV. How to Evaluate the Roots of an m Order Secular Determinant by Means of the IBM Model 602A

Step 1: Let $\lambda - \lambda_0 = 0, 1, 2, \ldots m$* and set up the m+1 determinants thus obtained. The term a_{11} must be an a card, while all the other elements of each determinant must be b cards.

Step 2: Obtain, by using Panel Board II, the m+1 matrices analogous to the one shown in Step 1 of Part I, performing the divisions continuously (a certain number of copies of each answer must also be produced by the machine).

Step 3: Use is now made of Board I. Follow the procedure explained in Step 3 of Section I. We thus obtain m+1 determinants of the (m-1)th order. It must be noted that the term $\left[-\frac{a_{12}}{a_{11}} a_{21} + a_{22}\right]$ (left hand corner) must be punched as an a card while all the other elements of the determinant must be b cards. In all cases, the additive multiplication contains two terms only, i.e., it is of the form e f + g = p; this can also be written as e f + 1 g = p, which, of course, is the operation Board I is supposed to perform.

In the same way, i.e., by repeating Steps 2 and 3 over and over again, one can

* For $(\lambda - \lambda_0) = h, 2h, \ldots mh$ where $h \neq 1$, c.f. (4).

evaluate the m+1 determinants, all at the same time; this is explained in Section I.

Step 4: With the values of those m+1 determinants at hand, one can easily reduce the secular determinant to a polynomial of the mth degree on λ.' (Consult (1) and (4).) This polynomial is of the form

$$f(\lambda) = \lambda^m + a\lambda^{m-1} + b\lambda^{m-2} + \dots + y\lambda + w$$

and

$$f'(\lambda) = m\lambda^{m-1} + a'\lambda^{m-2} + \dots + y$$

where f' (λ) is the derivative of f (λ) with respect to λ. The Panel Boards I and II can now be used to compute the roots of f (λ) by means of Newton's method, if all the roots are real. The following procedure is recommended:

Step 1: Use Board I and evaluate f (λ) and f' (λ) for an assumed value of λ.

Step 2: By means of Board II, compute the quotient $-f(\lambda)/f'(\lambda)$.

Step 3: A better approximation for one of the roots will be given by

$$\lambda' = \lambda + \left(-\frac{f(\lambda)}{f'(\lambda)} \right).$$

Repeat Step 1, 2 and 3 until all the roots are determined to the degree of accuracy desired.

V. Conclusions

It must be borne in mind that the method, as outlined above, is quite simple in that at no time during all the various operations one is required to read any of the intermediate answers obtained. Only the final answer, i.e., the λ's, should be read out of the cards. This is a distinct advantage, as it eliminates human errors and saves time. Moreover, aside from the punching machine, the only other equipment used is the Model 602A Calculating Punch, which performs all the operations required. Finally, the evaluation of the determinants can be checked by a simple procedure mentioned in (1) and (4).

* * * * * * * * * *

BIBLIOGRAPHY

1. Margenau and Murphy, "The Mathematics of Physics and Chemistry", pp. 482-487. D. van Nostrand Company

2. Frazer, Duncan and Collar, "Elementary Matrices", pp. 106-108. The MacMillan Co.

3. "IBM Electronic Punched Cards Accounting Machines. Principles of Operation", Calculating Punch, Type 602-A, International Business Machines Corporation, New York, New York.

4. Hicks, B. L., J. Chem. Physics 8, 569 (1940).

* * * * * * * * * * * * * * *

ACKNOWLEDGEMENT

Many thanks are due to Dr. D. H. Young, Professor of Applied Mechanics, and to Mr. H. A. Miller, Instructor in Electrical Engineering, for their interest and assistance in this project. This work was supported by the Office of Naval Research under contract Nonr-386(00)-NR-044004.

NOTES ON THE 602-A AND 604

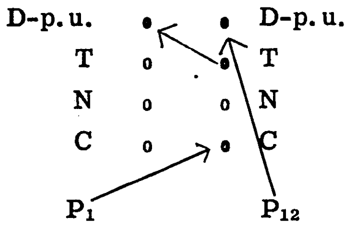
(Supplement to Technical Newsletter No. 3)

Eric V. Hankam

Watson Scientific Computing Laboratory
International Business Machines Corporation

1. The Balance (punch) conversion unit can only be activated by a 0 or 9. Any other digit wired into the B. T. hub will punch in the high order position in place of the actual first digit.

2. To p.u. a selector at the end of P_{12} (when more than 12 steps are required) and if MPL or DIV takes place on P_{12} , wire:



(Supplement to Technical Newsletter No. 3, p. 56)

3. a) Can multiply on two successive steps by X2 or X5; also ordinary multiplication can be followed by X2 or X5.

   b) Cannot MPL or DIV on Read cycle.

   c) Impulsing MPL and TENTH will not shift product one place to the right.

4. Emitter (but not Digit Selector converted into emitter) can be used as a constant multiplicand.

(Supplement to Technical Newsletter No. 3, p. 54)

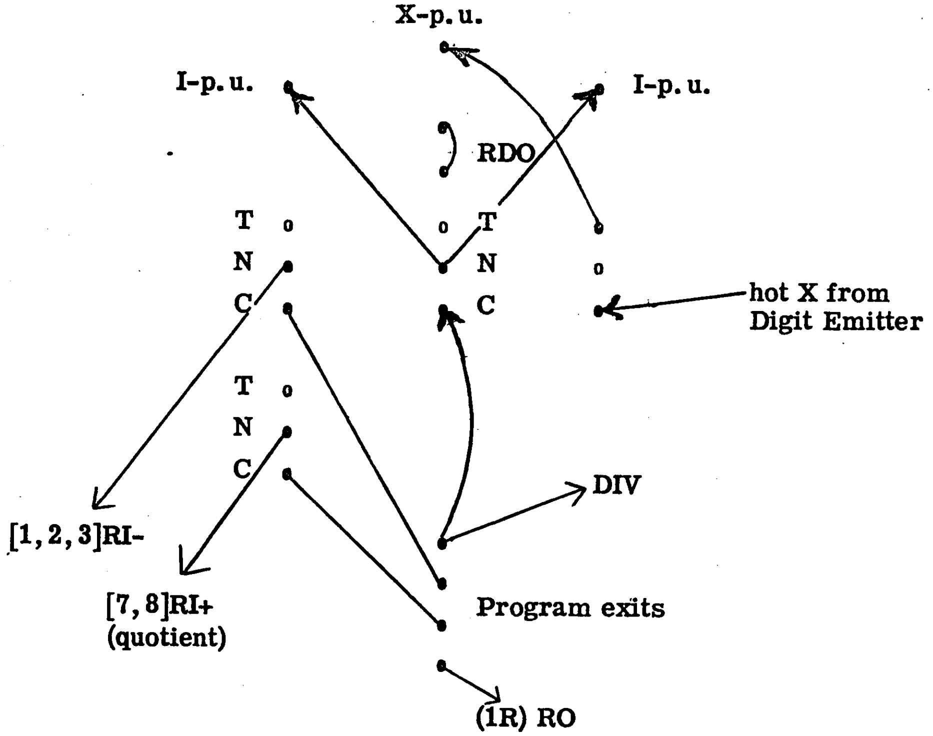
5. Any wires from the Emitter (not Digit selector converted into emitter) to a counter entry should be isolated during MPL or DIV steps (even if that counter is not involved in the MPL or DIV operation, and even if that counter is not impulsed to RI during the MPL or DIV step).

6. Whenever two (or more) counter exit positions (of the same counter or counter group) are split wired together, that counter should not be reset on the same step on which it is RO, and no other counter should be RI on the same step. An internal emitter makes common all like digits (on a given program step), and therefore also unlike digits if any two of them are split wired.

7. It is possible to RO one counter of a group of counters that are coupled together, and, on the same step RI another counter of the same group. In particular, the number in one counter of the coupled group can be transferred to another counter of the coupled group.

8. Whenever a counter is simultaneously impulsed to RI+ and RI-, it will not read in at all.

9. Division with a complement dividend can be done as described in Technical Newsletter No. 3, p. 59, but if the division requires 7 shifts, the first division cycle must be wired differently as outlined in the diagram below.* With this modification, any division problem with a complement dividend can be handled regardless of the number of shifts required.



* This method was suggested by Mr. S. Kelly, IBM Customer Engineer.

16 ± 16 ADDITION ON THE 604

(Supplement and correction to Technical Newsletter No. 3, p. 50)

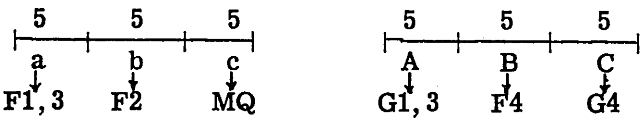
a) On machines equipped with Zero Test, $a + b_{17-9}$ can be zero tested directly on P_6, and several of the subsequent program steps can be eliminated. The $\frac{1}{2}$ Adjust operations on P_{19} and P_{31} will be suppressed if $a + b_{17-9} = 0$.

b) The $\frac{1}{2}$ Adjust operations on P_{19} and P_{31} should also be suppressed if $a + b_{12-9} \neq 0$. For that purpose, program steps should be inserted to zero test $a + b_{12-9}$. The number in EC has to be "boosted" by ±5 to preserve its sign only if $a + b_{12-9}$ is zero, but this method would fail in case $a + b_{12-9} = 9995$.

15 × 15 MULTIPLICATION ON THE 604

Eric V. Hankam

Watson Scientific Computing Laboratory
International Business Machines Corporation



G2 free

P1	: cB		: F4 RO ; MPL+	⟹	$cB \in EC$
P2	: c →	G2	: MQ RO ; G2 RI		
P3	: b →	MQ	: F2 RO ; MQ RI		
P4	: bC		: G4 RO ; MPL+	⟹	$cB + bC = M_4 \in EC$
P5	: c →	MQ	: G2 RO ; MQ RI		
P6	: A →	G2	: G1, 3 RO ; G2 RI		
P7	: M_4 →	G1, 3	: EC RR; G1, 3 RI; Ro 6th		(highest 6 places)
P8	: M_4 →	EC	: G1, 3 RO ; EC RI+		
P9	: cA		: G2 RO ; MPL+	⟹	$M_4 + cA \in EC$
P10	: C →	MQ	: G4 RO ; MQ RI		
P11	: aC		: F1, 3 RO ; MPL+	⟹	$M_4 + cA + aC \in EC$
P12	: B →	MQ	: F4 RO ; MQ RI		
P13	: bB		: F2 RO ; MPL+	⟹	$M_4 + cA + aC + bB = M_3 \in EC$
P14	: round		: 1/2 Adj ; Ri 5th		
P15	: M_3 →	G1, 3	: EC RR ; G1, 3 RI ; Ro 6th		(highest 6 places)
P16	: M_3 →	EC	: G1, 3 RO ; EC RI+		
P17	: aB		: F1, 3 RO ; MPL+	⟹	$M_3 + aB \in EC$
P18	: A →	MQ	: G2 RO ; MQ RI		
P19	: bA		: F2 RO ; MPL+	⟹	$M_3 + aB + bA = M_2 \in EC$
P20	: M_2 →	G2	: EC RO ; G2 RI		(lowest 5 places)
P21	: M_2 →	G1, 3	: EC RR ; G1, 3 RI ; Ro 6th		(highest 6 places)
P22	: M_2 →	EC	: G1, 3 RO ; EC RI +		
P23	: aA		: F1, 3 RO ; MPL+	⟹	$M_2 + aA = M_1 \in EC$
Punch	: EC &	G2			

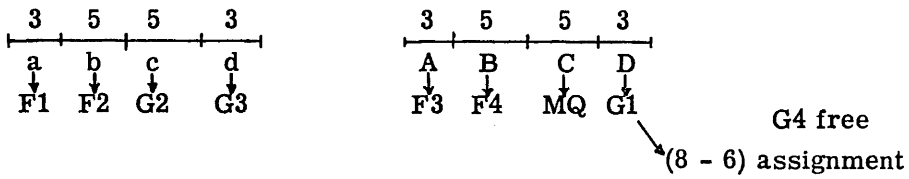
$$\underbrace{\underset{\underset{10}{\downarrow}}{EC} \quad \& \quad \underset{\underset{5}{\downarrow}}{G2}}_{\text{15 positions}}$$

NOTE: This method differs from a previous article (Technical Newsletter No. 3) in that it includes the partial products cB and bC , thus reducing the round-off error. The previous method has the advantage of requiring fewer electronic cycles.

16 × 16 MULTIPLICATION ON 604

Eric V. Hankam

Watson Scientific Computing Laboratory
International Business Machines Corporation



P1	:	C→G4	:	MQ RO ; G4 RI	
P2	:	C→EC	:	MQ RO ; EC RI+	
P3	:	C→MQ	:	EC RR ; Ro 3rd; MQ RI	$\Longrightarrow$ C $\in$ MQ ooxxx dropping lowest 2
P4	:	c C	:	G2 RO ; MPL+	$\Longrightarrow$ c C $\in$ EC xxx|xxxxx
P5	:	c C→MQ	:	EC RR ; Ro 6th; MQ RI	$\Longrightarrow$ highest 3 pos. of c C in MQ
P6	:	D→EC	:	G1RO ; EC RI+	xxx ooooo (due to 8 - 6 assignment)
P7	:	c C→G1	:	MQ RO ; G1 RI; Ri 6th	
P8	:	D→MQ	:	EC RR ; Ro 6th; MQ RI	$\Longrightarrow$ D $\in$ MQ ooxxx
P9	:	c C→EC	:	G1 RO ; EC RI+	xxx ooooo
P10	:	a→G1	:	F1 RO ; G1 RI; Ri 6th	xxx ooooo
P11	:	(a + B) D	:	G1 RO ; F2 RO; MPL+	$\Longrightarrow$ c C + (a + b) D $\in$ EC xxx←cC xxxxxxxxxxx
P12	:	A→G1	:	F3 RO ; G1 RI; Ri 6th	xxx ooooo
P13	:	d→MQ	:	G3 RO ; MQ RI	
P14	:	(A + B) d	:	G1RO ; F4 RO; MPL+	cC + (a + b) D + (A + B) d = M_1 $\in$ EC $\overset{12th}{\underbrace{xxxxxxxx}_{8}}$|xxxx
P15	:	M_1 → MQ	:	EC RO ; MQ RI; Ro 5th	ooo xxxxx
P16	:	→ G1	:	EC RR ; G1 RI; Ro 5th	xxx ooooo
P17	:	M_1 → EC	:	MQ RO ; EC RI+; Ri 2nd	ooxxxxxo
P18	:		:	G1 RO ; EC RI+; Ri 2nd	xxx-----o
P19	:	c→MQ	:	G2 RO ; MQ RI	
P20	:	B c	:	F4 RO ; MPL+	$\Longrightarrow$ M_1 + Bc $\in$ EC carry oxxxxxxxxo←M_1 xxxxxxxxxxx
P21	:	C → MQ	:	G4 RO; MQ RI	
P22	:	b C	:	F2 RO ; MPL+	$\Longrightarrow$ M_1 + Bc + bC = M_2 $\in$ EC $xxxxxx\|\underset{\uparrow 5}{x}xxxx$
P23	:	round	:	1/2 Adj. ; Ri 5th	

P24	: $M_2 \nearrow$	G4	: EC RO ; Ro 6th ; G4 RI		xxxxx
P25	: $\searrow$	G1	: EC RR ; Ro 6th ; G1 RI		ooxoooo
P26	: $M_2 \longrightarrow$	EC	: G1 RO ; G4 RO ; EC RI+		xxxxxx
P27	: Ca		: F1 RO ; MPL+	$\Longrightarrow M_2 + Ca \in$ EC	carry xxxxxx ← M_2 xxxxxxxxx
P28	: c $\longrightarrow$	MQ	: G2 RO ; MQ RI		
P29	: cA		: F3 RO ; MPL+	$\Longrightarrow M_2 + Ca + cA \in$ EC	xxxxxxxxx
P30	: B $\longrightarrow$	MQ	: F4 RO ; MQ RI		
P31	: bB		: F2 RO ; MPL+	$\Longrightarrow M_2 + Ca + cA + bB = M_3 \in$ EC	xxxxxxxxx xxxxxxxxxxx
P32	: $M_3 \longrightarrow$	G4	: EC RO ; G4 RI	(lowest 5 digits) to be punched	
P33	: $M_3 \nearrow$	G2	: EC RO ; Ro 6th ; G2 RI	oooxxxx	highest 6 digits
P34	: $\searrow$	G1	: EC RR ; Ro 6th ; G1 RI	ooxoooo	
P35	: $M_3 \longrightarrow$	EC	: G1 RO ; G2 RO ; EC RI+	(highest 6 digits)	xxxxxx
P36	: aB		: F1 RO ; MPL+	$\Longrightarrow M_3 + aB = M_4 \in$ EC	xxxxxxxx
P37	: A $\longrightarrow$	MQ	: F3 RO ; MQ RI		
P38	: a $\longrightarrow$	G1	: F1 RO ; Ri 6th ; G1 RI		
P39	: (a + b) A		: G1 RO ; F2 RO ; MPL+	$\Longrightarrow M_4 + (a + b)A \in$ EC	xxxxxxxx ← M_4 $\underbrace{\text{xxxxxxxxxxx}}_{\text{11 pos.}}$

Punch : EC & G4

$$\underbrace{\overset{\text{EC}}{\downarrow}\ 11 \qquad \overset{\text{G4}}{\downarrow}\ 5}_{\text{16 positions}}$$

NOTE: Read in and calculate absolute values; determine final sign through selectors.

DIFFERENCING ON THE CPC - MODEL II

Stanley Poley
Department of Civil Engineering
Columbia University
New York, New York

A description is given herein of a set of control panels for the CPC - Model II[1] by means of which high order differences may be easily and rapidly computed. Principal features of this set-up are:

1) The order of the highest difference desired is set by plugging one wire.
2) No time is wasted in computing differences of order higher than called for in (1).
3) No blank cards or instruction cards are required; only function cards are run through the 418.

The control panels described in Part I will accommodate differences up to tenth order for ten digit numbers. In Part II, several auxiliary features are given which increase the capacity to twentieth order differences of thirteen digit numbers.

These control panels were wired and tested at the Watson Scientific Computing Laboratory at Columbia University.

Part I

The flow of data is shown in Figure 1. Note that one 418 card cycle and n 418 program cycles are required to obtain an n^{th} order difference. Thus the speed of of operation is $\frac{150}{n+1}$ cards per minute.

[1] Equipped with a special program unit.

			941 Storage Units									
Cycle	Card	6A 6B	11	12	13		18	21	F2, 4	G2, 4	EC	List
Stored →		y_{i-1}	∇^1_{i-1}	∇^2_{i-1}	∇^3_{i-1}		∇^8_{i-1}	∇^9_{i-1}				
card cycle	y_i →	RR →							→ y_{i-1}	→ y_i	RR	→ y_i
605									RO →	RO →	y_i $-y_{i-1}$ ∇^1_i	
P1		y_i ←	RO →						→ ∇^1_{i-1}	RO	RO →	∇^1_i
605									RO →	∇^1_i ←	RO $-\nabla^1_{i-1}$ ∇^2_i	
P2			∇^1_i ←	RO →					→ ∇^2_{i-1}	RO	RO →	∇^2_i
605									RO →	∇^2_i ←	RO $-\nabla^2_{i-1}$ ∇^3_i	
P3				∇^2_i ←	RO →				→ ∇^3_{i-1}	RO	RO →	∇^3_i
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
P9							∇^8_i ←	RO →	∇^9_{i-1}	RO	RO →	∇^9_i
605									RO →	∇^9_i ←	RO $-\nabla^9_{i-1}$ ∇^{10}_i	
P10								∇^9_i ←		RO	RO →	∇^{10}_i

Figure 1: Flow of Data

Wiring for the 418 is shown in Figure 2, while Figures 3 and 4 give, respectively, 605 program pick-up wiring and 605 programming.

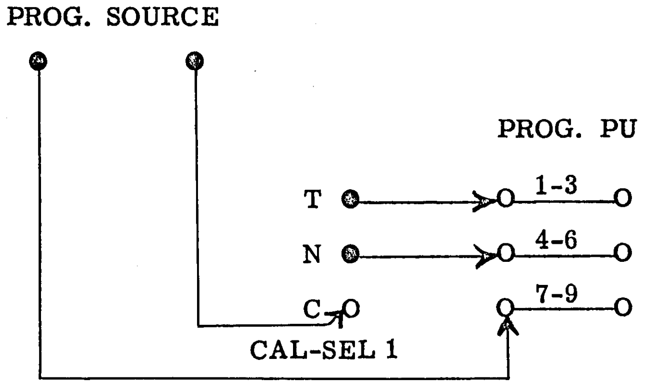


Figure 3: 605 Program Pick-up Wiring

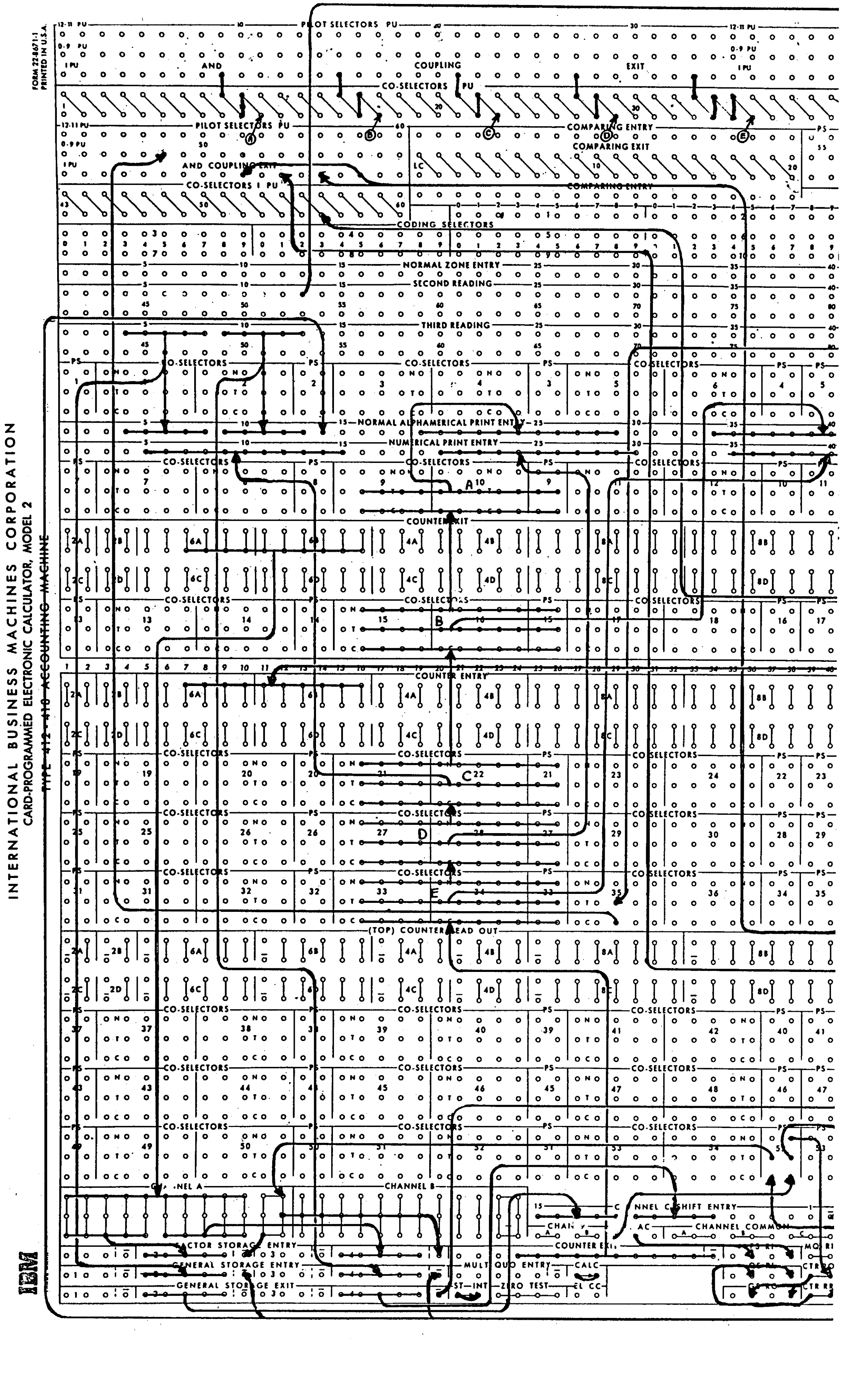
Prog. Step	Operation
1	G4 RO, EC RI+
2	G2 RO, EC RI+, Ri 6th
4	EC RO, G4 RI
5	EC RO, G2 RI, Ro 6th
7	F4 RO, EC RI-
8	F2 RO, EC RI- , Ri 6th

Figure 4: 605 Programming

All differences are stored in the 941 and are read in or out by emitting channel codes through successively transferred levels of a field selector, level 0 corresponding to P1, level 1 to P2, ... , level 9 to P10. However, the function y_{i-1} in Figure 1 is stored in 418 counters 6A, 6B in order to avoid 941 RI-RO timing difficulties which would be encountered when taking first differences.

As indicated in Figure 2, one wire must be plugged to level (n - 1) of the right-hand column of the field selector to obtain n^{th} order differences. This wire impulses MI-STOP and transfers Co-Sel 55.[2] Since the field selector is picked-up through the normal side of Co-Sel 55, it will not be picked-up during Pn and will thus be normal on the following card cycle. Channel A control is wired through the normal side of Co-Sel 55 to prevent a 941 RO during card cycles which would conflict with the RR of counters 6A, 6B.

[2] Notation in this paper is that employed by Eric V. Hankam in his paper, "Notes on the IBM Type 604 Electronic Calculating Punch and Type 602A Calculating Punch", IBM Technical Newsletter No. 3.


FORM 22-8671-1
PRINTED IN U.S.A.
INTERNATIONAL BUSINESS MACHINES CORPORATION
CARD-PROGRAMMED ELECTRONIC CALCULATOR, MODEL 2
~~TYPE 412-418 ACCOUNTING MACHINE~~
IBM
PILOT SELECTORS PU
AND
COUPLING
EXIT
CO-SELECTORS PU
COMPARING ENTRY
COMPARING EXIT
AND COUPLING EXIT
CO-SELECTORS I PU
LC
CODING SELECTORS
NORMAL ZONE ENTRY
SECOND READING
THIRD READING
CO-SELECTORS
NORMAL ALPHAMERICAL PRINT ENTRY
NUMERICAL PRINT ENTRY
COUNTER EXIT
COUNTER ENTRY
(TOP) COUNTER READ OUT
CHANNEL A
CHANNEL B
CHANNEL SHIFT ENTRY
CHANNEL COMMON
FACTOR STORAGE ENTRY
GENERAL STORAGE ENTRY
GENERAL STORAGE EXIT
COUNTER EXIT
MULTIPLIER QUOTIENT ENTRY
CALC
ZERO TEST
CTR RI
A
B
C
D
E

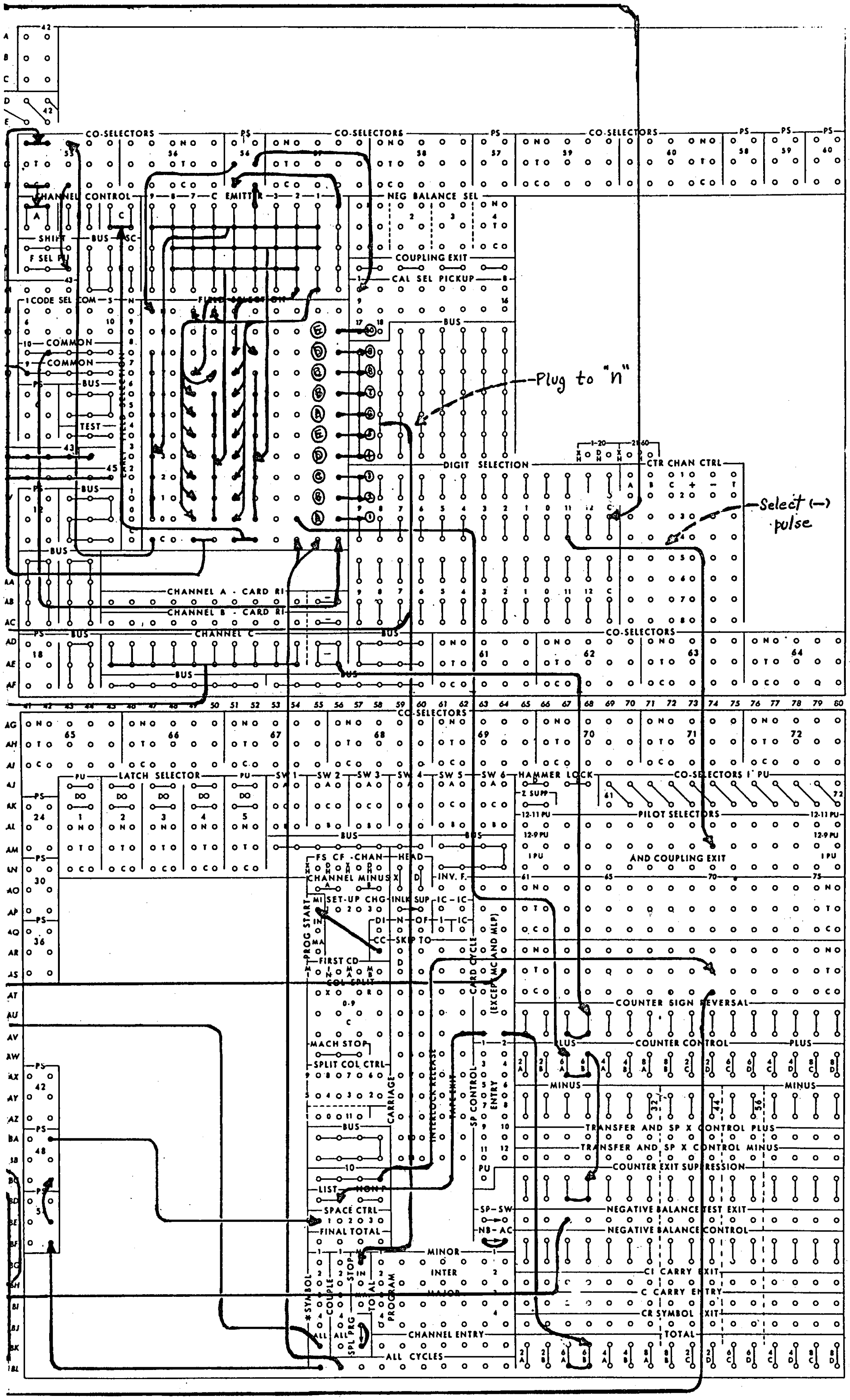


Figure 2: Wiring for up to 10th Order Differences of 10 Digit Numbers

The card column containing the minus punch for y is wired from 2nd reading to the common of a digit selector. Pilot selector 70 should be picked-up from the digit selector hub corresponding to the minus punch.

The differences list from EC through five gating selectors whose pick-up hubs are denoted by A, B, C, D, E in Figure 2. These pick-up hubs should be wired from correspondingly labeled hubs of the field selector. When gating selector E is transferred, Pil-Sel 48 will transfer on the following cycle, suppressing space suppress for that cycle. Thus the differences list in two rows, five across each row. The function y lists to the left of the first row.

Part II

A) Extension for Higher Order Differences:

By wiring a second field selector in series with the first, as shown in Figure 5 (ordinary selectors may be wired to operate like a field selector if only one field selector is available), up to twentieth order differences (of ten digit numbers) may be obtained in a single run.

B) Thirteen Digit Numbers:

By wiring 418 counters to operate in conjunction with the 941 storage units, we may accommodate up to thirteen digit numbers if it is assumed that differences of order greater than eleven will not exceed ten digits. The additional wiring required is shown in Figure 6 (for up to twentieth order differences). Moreover, F1 and G1 should be given (8-6) assignments and the 605 controls of F1, F2 and G1, G2 coupled, i.e., in Figure 4, F2RO becomes F1,2RO, etc. Counter wiring associated with the second field selector may be omitted if only tenth order differences or less are required.

C) Designation of "n" by External Switches:

In some installations, it may be desirable to select n , the order of the highest difference, by using external switches rather than by plugging wires on the 418 control panel. Figure 7 shows a christmas tree circuit wherein n is set in binary form using AB switches (set-up change switches may be substituted for AB switches, if necessary). Setting of the switches for n = 19, for example, is shown in Figure 8. Note that the AC switch position corresponds to 1 while BC corresponds to 0 .

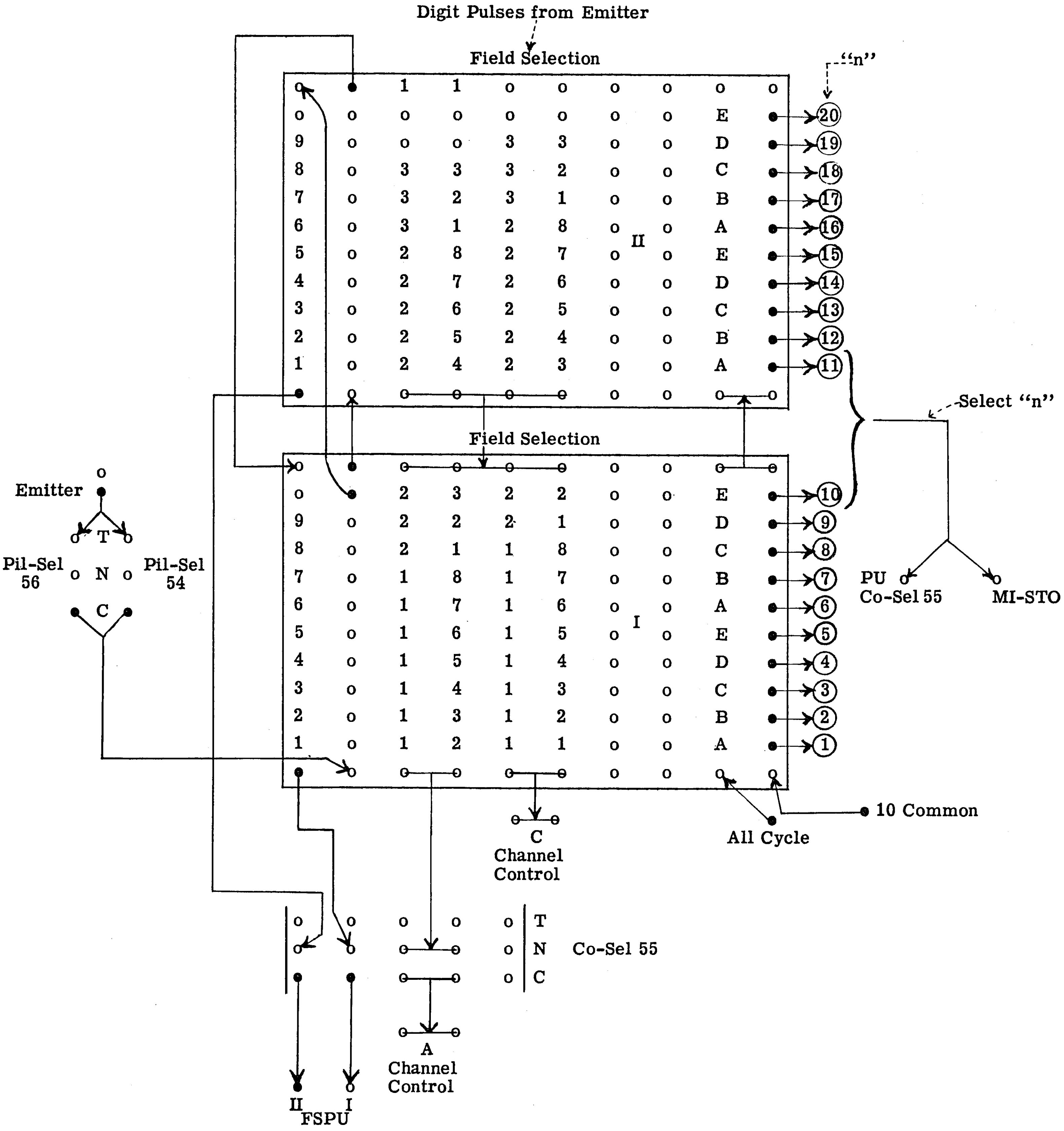
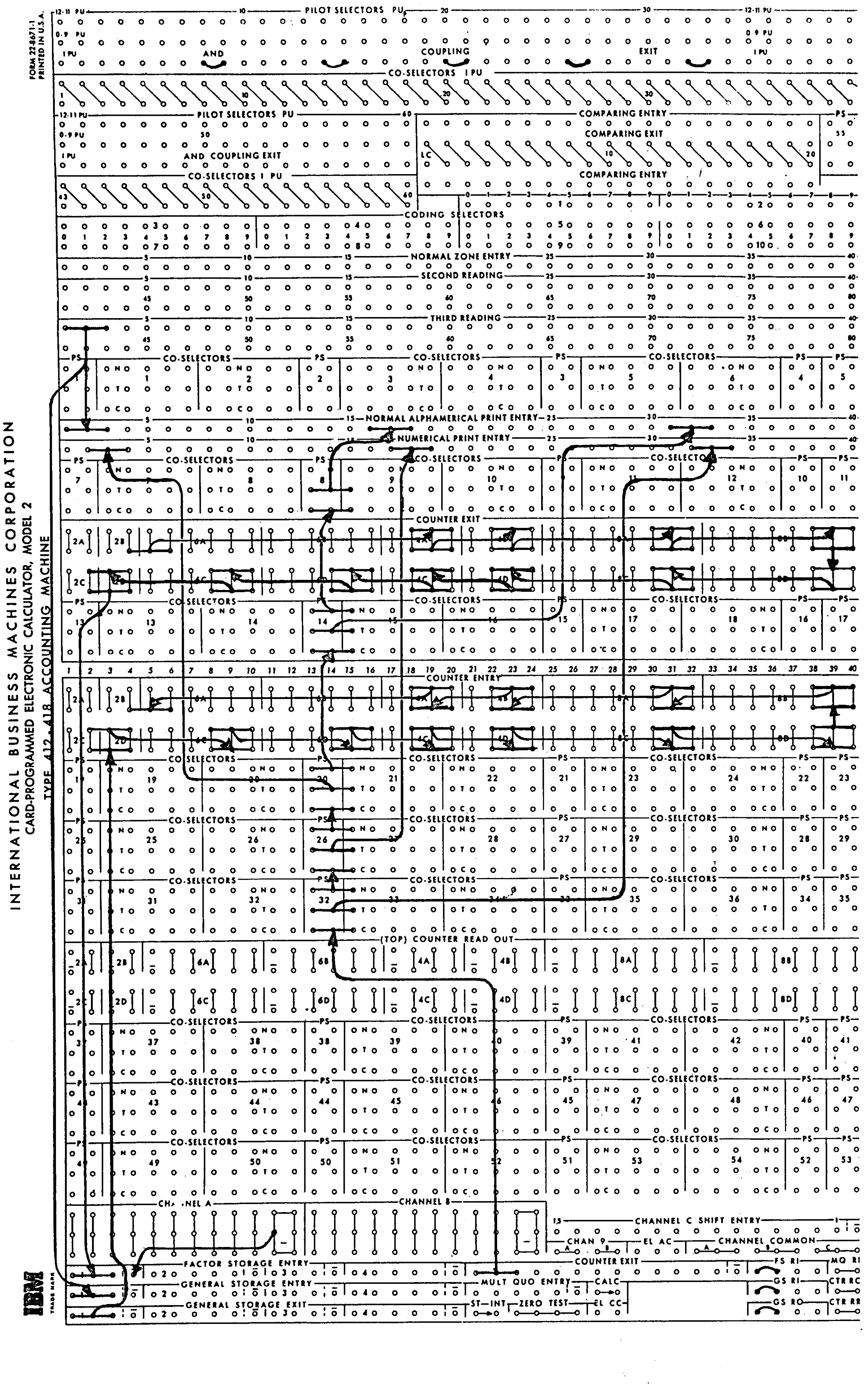


Figure 5: Field Selector Wiring for up to 20th Order Differences


INTERNATIONAL BUSINESS MACHINES CORPORATION
CARD-PROGRAMMED ELECTRONIC CALCULATOR, MODEL 2
TYPE 412-418 ACCOUNTING MACHINE
FORM 22-8671-1
PRINTED IN U.S.A.
IBM
TRADE MARK
PILOT SELECTORS PU
AND
COUPLING
EXIT
CO-SELECTORS I PU
PILOT SELECTORS PU
AND COUPLING EXIT
COMPARING ENTRY
COMPARING EXIT
CODING SELECTORS
NORMAL ZONE ENTRY
SECOND READING
THIRD READING
CO-SELECTORS
NORMAL ALPHAMERICAL PRINT ENTRY
NUMERICAL PRINT ENTRY
COUNTER EXIT
COUNTER ENTRY
(TOP) COUNTER READ OUT
CHANNEL A
CHANNEL B
CHANNEL C SHIFT ENTRY
CHAN 9
EL AC
CHANNEL COMMON
FACTOR STORAGE ENTRY
COUNTER EXIT
GENERAL STORAGE ENTRY
MULT QUO ENTRY
CALC
GENERAL STORAGE EXIT
ST-INT
ZERO TEST
EL CC
FS RI
MQ RI
GS RI
CTR RC
GS RO
CTR RR

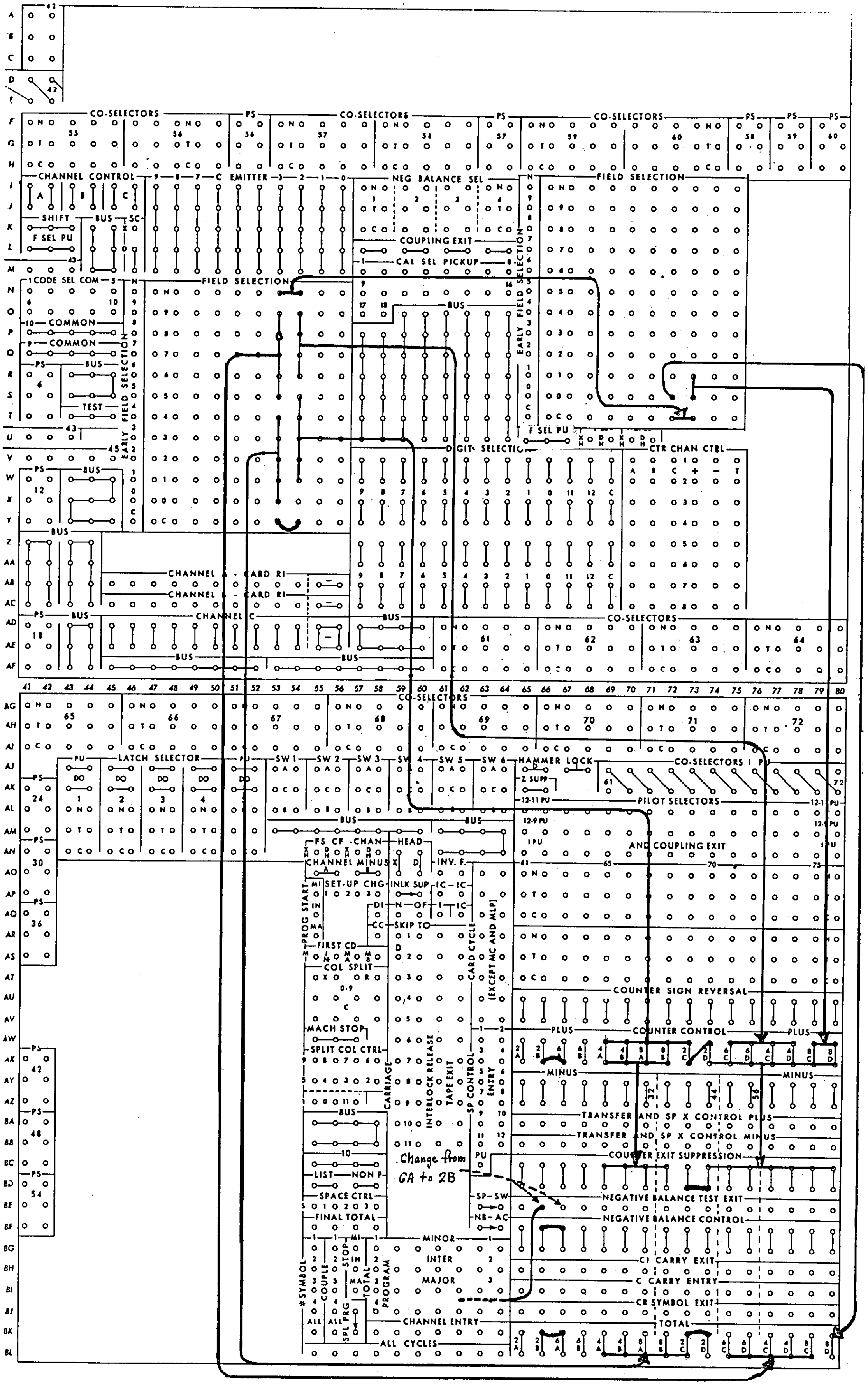


Figure 6. Additional Wiring for 13 Digit Numbers

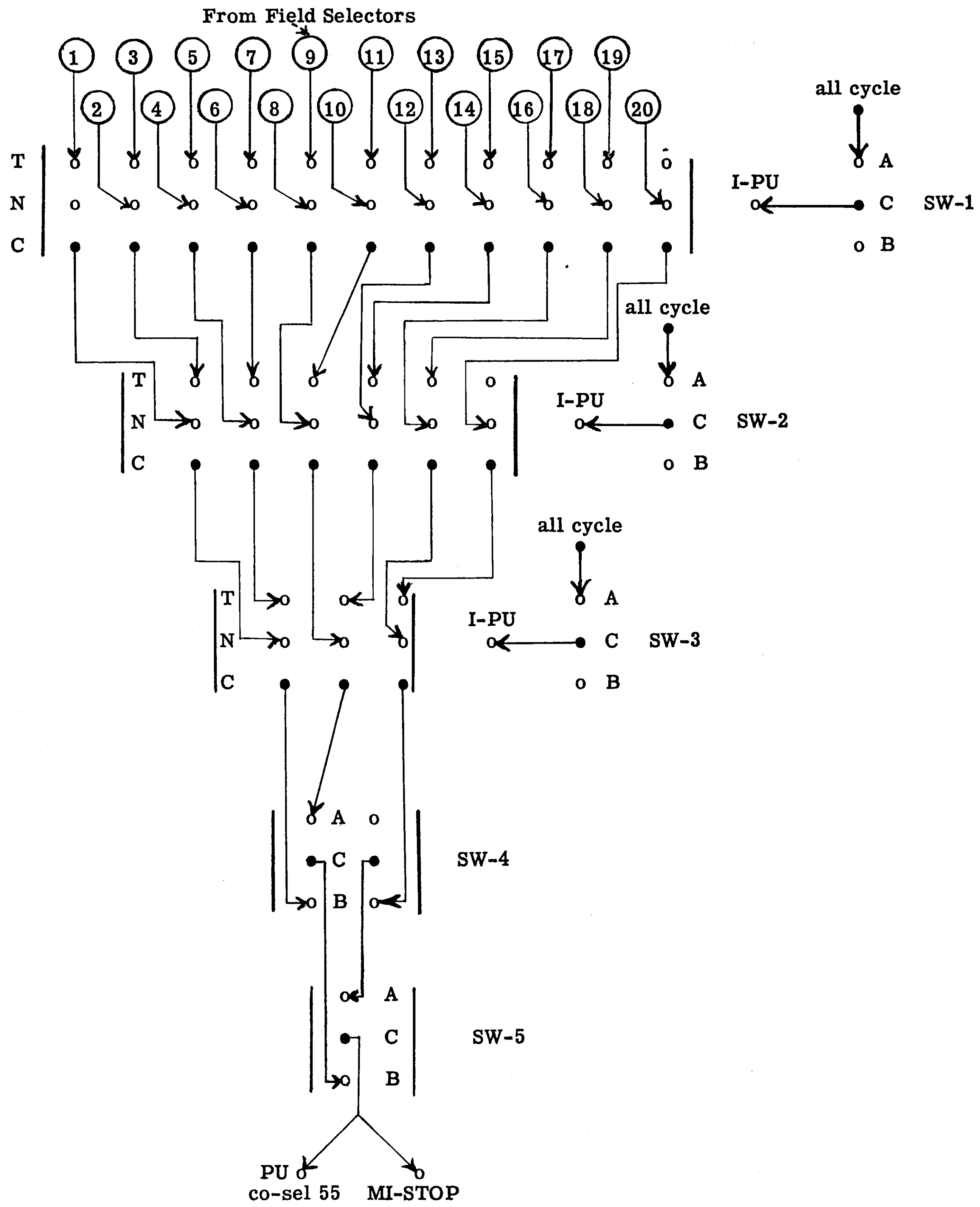


Figure 7: Wiring for External "n" Designation

	A	B
5	x	
4		x
3		x
2	x	
1	x	

Figure 8: Switches Set for 19th Order Differences

CALCULATION OF HIGH ORDER DIFFERENCES ON THE 607 ELECTRONIC CALCULATOR

Eric V. Hankam
Watson Scientific Computing Laboratory
International Business Machines Corporation

On a high speed calculating machine equipped with Program Repeat, such as the 607, where the skipping of program steps does not involve any appreciable loss of time, certain types of problems characterized by a repetitive pattern can be set up by repeating a "loop" of relatively few program steps. This set of program steps can be considered as a "subroutine" which can be incorporated on a control panel on which many other operations are wired. The evaluation of high order differences has been chosen to illustrate this principle.

The program below describes the computation of 8th order differences of 8-digit numbers. This can easily be extended to a) higher order differences, and b) 13-digit numbers. If Program Repeat Selectors were available, a selector network could be set up to count the number of program repeats, and terminate the calculation after the required number of repetitions. Such a selector network would further decrease the number of program steps in the loop. In the example following, steps 13 - 15 perform the counting by successive subtractions of 1's from 7 (one less than the highest order difference to be computed). Group Suppressions might provide an alternate method to keep track of successive program repeats.

607 Program for 8th Differences*

Read: $f_i \longrightarrow$ F1,2
$7 \longrightarrow$ F3

P1 : F1,2 RO ; EC RI+
P2 : EC RO ; G17,18 RI

P3 : G1,2 RO ; EC RI- ; SKIP OFF (develops Δ_i^k , k = 1,2,... 8)
P4 : G3,4 RO ; G1,2 RI

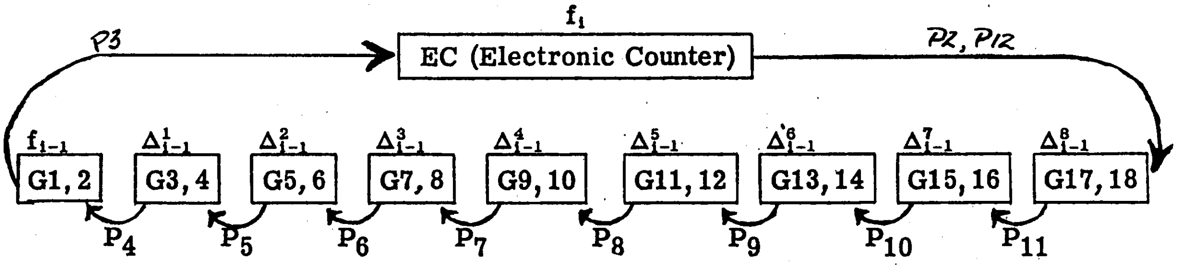
*Description is for i^{th} card. Notation used is similar to that employed in Technical Newsletter No. 3.

```
P5  :    G5,6 RO    ;  G3,4 RI
P6  :    G7,8 RO    ;  G5,6 RI
P7  :    G9,10 RO   ;  G7,8 RI
P8  :    G11,12 RO  ;  G9,10 RI
P9  :    G13,14 RO  ;  G11,12 RI
P10 :    G15,16 RO  ;  G13,14 RI
P11 :    G17,18 RO  ;  G15,16 RI
P12 :    EC  RR     ;  G17,18 RI

P13 :    F3 RO      ;  EC RI+
P14 :    Em 1       ;  EC RI- ; B. T.
P15 :    EC RR      ;  F3 RI
P16 :    G17,18 RO  ;  EC RI+ ; Prog. Repeat ; Supp -
P17 :    SKIP ON
```

Punch: At the end of the calculation the differences will appear in and can be punched from the following units: Δ_i^1 in G3,4; Δ_i^2 in G5,6;... Δ_i^8 in G17,18. The differences remain in these storage units for the (i + 1)st card.

Schematic Diagram of Program Loop



Note that the (i - 1)st 8th difference is first replaced by the i^{th} function on P_2, then the i^{th} first difference is obtained on P_3, and then each difference is successively shifted one storage unit to the left, etc.

Remarks on the Program

1. a) f_i need not be read from the card; it can be the result of a preliminary computation, and must be entered into EC before the differences can be computed. The 7 can also be emitted into F3 on any program step prior to the computation of the differences.

   b) If f_i is read from the card it could be entered directly into G17,18 instead of F1,2; in that case Δ^8 may be transferred to EC to be punched.

2. The result of the B. T. on P_{14} can be used to control program steps which are not part of the differencing loop, e.g. SKIP ON on P_{17} can be suppressed on - if other calculations are to be programmed after the differences have been obtained.

RAPID DIFFERENCING ON IBM ACCOUNTING MACHINES

E. Brown
Physics Dept., Cornell University
Ithaca, New York

Introduction

A standard operation on IBM accounting machines is the calculation of differences of a function. The methods currently in use require program cycles on the accounting machine which may add considerably to the calculation time. The method discussed here is one in which program cycles are not required.

Method

The basis of the present method is in the use of a group of three counters to obtain the first differences of the function read into them. These counters are instructed in accordance with the following scheme:

Machine Cycle #	Counter A_1	Counter A_2	Counter A_3
3n	RI+	RI-	Total
3n+1	Total	RI+	RI-
3n+2	RI-	Total	RI+

It can readily be seen that, if successive values of a function are read in on each cycle, the differences will be successively developed in each one of the three counters. The difference $\Delta_n = f_{n+1} - f_n$ will be read out of one of the counters on the same cycle that f_{n+2} is read into the other two. If the output of this group of counters is read into a similar group of counters, then the output of this latter group is the second difference of the function. This procedure can, in theory, be used to obtain any order difference. The time required is independent of the order. The restriction on the order of the difference is imposed by the number of counters in the machine.

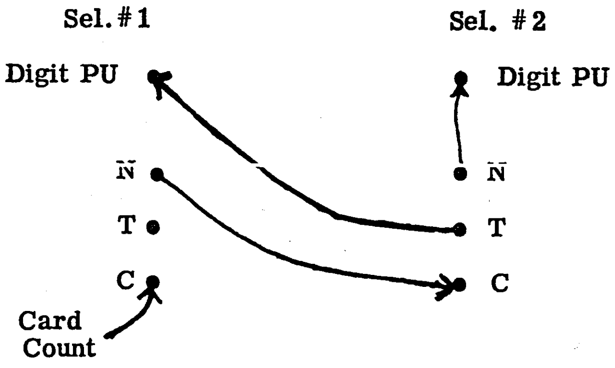
The panel described in this paper obtains third order differences of seven digit numbers and makes use of nine eight-position counters. (A combination of a coupled 6 and 2 position counter will be referred to as an 8 position counter). These nine counters will be referred to by the following notation:

A_1, A_2, A_3, $\longrightarrow$ counters necessary for first differences

B_1, B_2, B_3, $\longrightarrow$ counters necessary for second differences

C_1, C_2, C_3, $\longrightarrow$ counters necessary for third differences

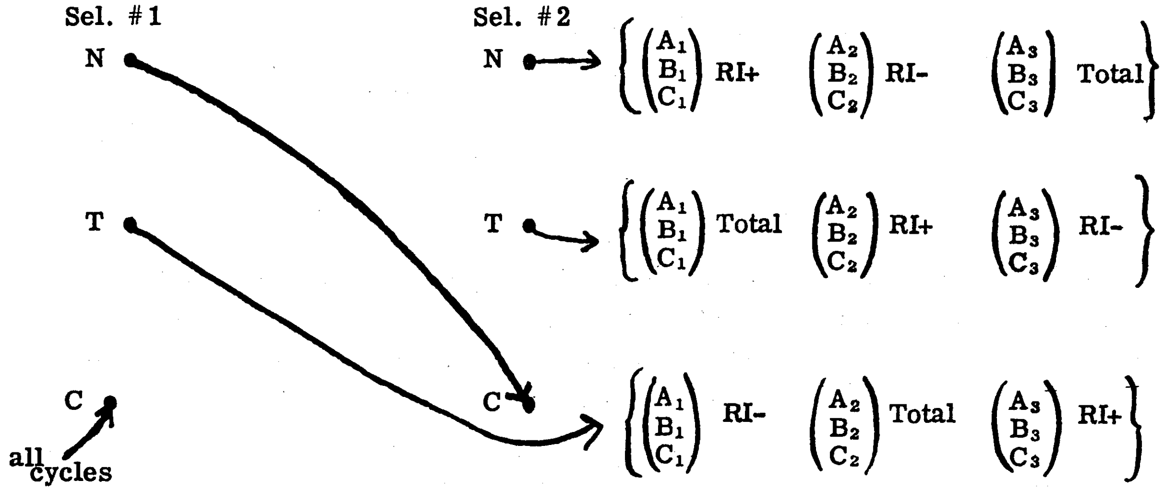
The counters are controlled by means of two pilot selectors. The selector wiring is illustrated below:



Assuming both these selectors to be normal to start with we obtain the following pattern:

Machine Cycle #	Selector #1	Selector #2
3n	N	N
3n+1	N	T
3n+2	T	N

The counter controls are impulsed through the other position of the pilot selectors as shown below:



For simplicity in wiring, the exits of the counters in each group are connected. Because of the fact that a hot pulse reaching a counter exit will prevent it from resetting properly when it is impulsed to total it is necessary to impulse counter exit suppression for each counter when it is reading in (plus or minus).

The method illustrated here yields true figures for the differences. For this purpose the NB exits of the counters are wired to the corresponding NB control entries. Thus, when a counter which contains a negative number in 9 complements form is totalled a conversion cycle takes place which transforms this number to a true figure. These conversion cycles add time to the calculation. (It may be possible to remedy this defect by using the usual procedure for subtracting on non-net balance machines. This procedure however requires twice as many counters as are required here.) However, if one is willing to print complements, one may obtain the maximum speed of one card cycle per set of differences. The wires which are represented by dashed lines may be removed for this purpose. For greater flexibility it is preferable to wire them through co-selectors which may be picked up from a set-up change hub.

The purpose of the field selector and associated selectors is to prevent printing of meaningless differences at the beginning of a run. For example the first difference column has no entry until f_3 is listed. The printing takes place as follows:

Function	First Difference	Second Difference	Third Difference
f_1			
f_2			
f_3	$f_2-f_1=\Delta^1_1$		
f_4	$f_3-f_2=\Delta^1_2$		
f_5	$f_4-f_3=\Delta^1_3$	$\Delta^1_2-\Delta^1_1=\Delta^2_1$	
f_6	$f_5-f_4=\Delta^1_4$	$\Delta^1_3-\Delta^1_2=\Delta^2_2$	
f_7	$f_6-f_5=\Delta^1_5$	$\Delta^1_4-\Delta^1_3=\Delta^2_3$	$\Delta^2_2-\Delta^2_1=\Delta^3_1$
f_8	$f_7-f_6=\Delta^1_6$	$\Delta^1_5-\Delta^1_4=\Delta^2_4$	$\Delta^2_3-\Delta^2_2=\Delta^3_2$
. . .	. . .	. . .	. . .

The card deck should be followed by three blank cards in order to obtain the last significant third difference. This will result in the printing of a meaningless second difference for the last card and a meaningless first difference for the preceding card. No provision has been made to suppress these on the present panel. If one desires to suppress these differences from printing it may be achieved by using x-punches in the trailer cards to pick up the co-selectors through which these print.

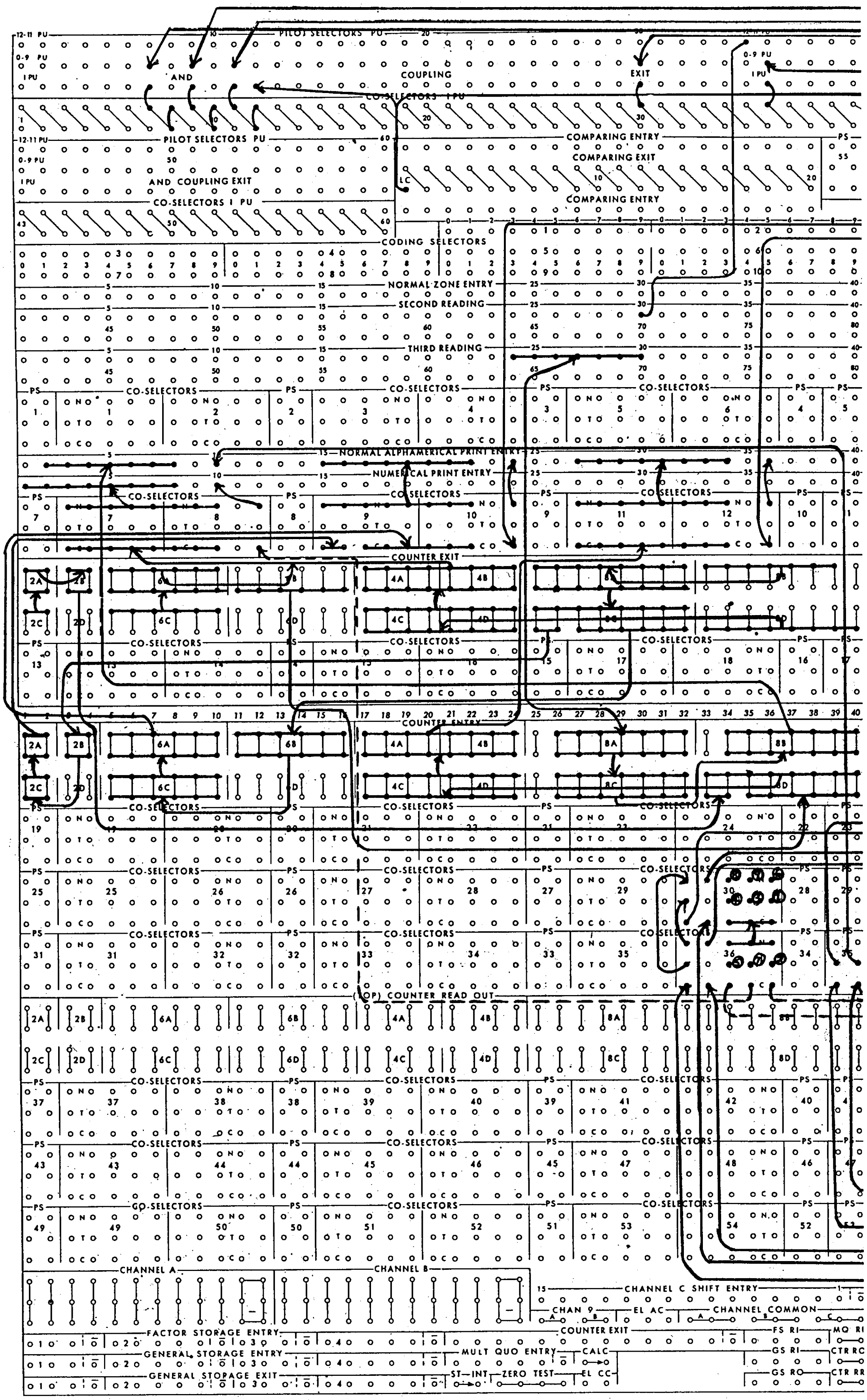
The panel presented here was wired and tested at Watson Scientific Computing Laboratory, International Business Machines Corporation.

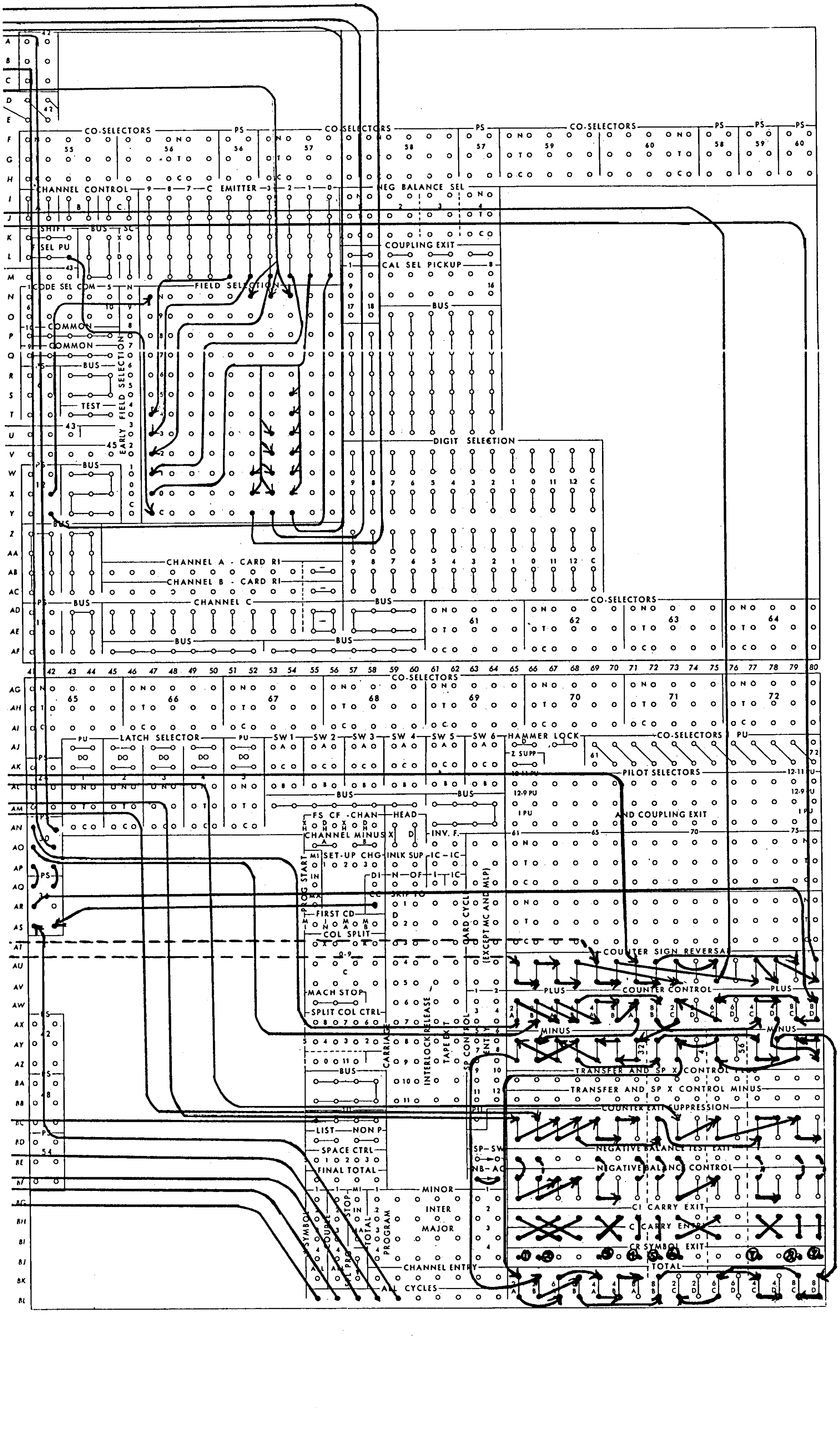
Notes

1) It is possible for the field selector to be transferred at the beginning of a run. This is the case if a deck has just been run since the LC pulse initiates the transfer of the field-selector. For this reason the power switch should be turned off and on prior to running a deck.

2) It is not necessary to clear any counters before starting.

3) The function cards should not be preceded by a blank card.


12-11 PU
PILOT SELECTORS PU
0-9 PU
I PU
AND
COUPLING
EXIT
CO-SELECTORS I PU
PILOT SELECTORS PU
AND COUPLING EXIT
CO-SELECTORS I PU
COMPARING ENTRY
COMPARING EXIT
COMPARING ENTRY
LC
CODING SELECTORS
NORMAL ZONE ENTRY
SECOND READING
THIRD READING
CO-SELECTORS
PS
NORMAL ALPHAMERICAL PRINT ENTRY
NUMERICAL PRINT ENTRY
COUNTER EXIT
COUNTER ENTRY
(OP) COUNTER READ OUT
2A
2B
2C
2D
6A
6B
6C
6D
4A
4B
4C
4D
8A
8B
8C
8D
CHANNEL A
CHANNEL B
CHANNEL C SHIFT ENTRY
CHAN 9
EL AC
CHANNEL COMMON
FACTOR STORAGE ENTRY
COUNTER EXIT
FS RI
MQ RI
GENERAL STORAGE ENTRY
MULT QUO ENTRY
CALC
GS RI
CTR RC
GENERAL STORAGE EXIT
ST
INT
ZERO TEST
EL CC
GS RO
CTR RR

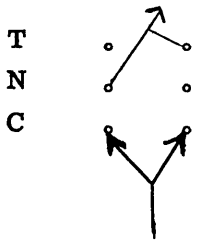


THIRD DIFFERENCES OF SEVEN DIGIT NUMBERS

The numbered hubs on co-selectors 30 and 36 are wired to the correspondingly numbered CR Symbol Exit hubs

ERRATA

In Technical Newsletter No. 5 in the article "Boolean Notation for Selector Networks" by Eric V. Hankam, there is an error in the diagram (d) at the top of page 27. The correct diagram should be:



$\overline{A}^{i}$ v A^{j}

Technical Newsletter No. 5 - A Model II Four-Address Floating-Decimal Coding System - Dura W. Sweeney.

Mr. Richard C. Lesser of Cornell University has discovered the following three inaccuracies which appear in the calculate portion of the CPC board described in this article.

1. The computation of e^{B} where

   $$B = -n \ \ell n \ 10$$

   where n = 1, 2, 3, 4, 10, will cause the machine to compute as if

   $$B = n \ \ell n \ 10$$

   This occurs since division by ℓn 10 now gives a minus zero remainder, and on a later program this sign is lost as the remainder is added into the electronic counter. Since n is later half-added into the counter under control of the sign of the remainder, this causes the answer to appear as 10^{n} rather than 10^{-n}.

2. When the number 9.9999995 is added to zero (or, in fact, if 9.9999995 is the result of any operation), the machine, by virtue of double round-off, gives 10.000000 as the answer.

3. Although the manual (LA 1475) states that multiplication can be performed if numbers are not in true floating decimal form, a restriction regarding significant figures should be noted.

   Multiplying 00741000 x 00741000

   ```
           0.0741 000      54     741.
           0.0741 000      54     741.
      0000549081000000         549081.
   ```

   the machine rounds-off after the 8, and the 1 is dropped, giving, after shifting a result of

   5.4908000 55

Mr. Sweeney has made the following remarks and modifications in connection with these.

1. I have modified the 605 control panel as follows to eliminate the difficulty of obtaining

$$e^{-n \ell n\, 10}$$

   1. Panel C step 49 change "half-add" to "ctr + (63)".
   2. Panel C step 51 change emit 2; divide; ---; #44 to RO; GS3; ---; #15.
   3. Panel C step 52 change emit 1; ctr -; BT; reset; ---; #44 to emit 2; divide; ---; #34.
   4. Panel C step 53 change RO; GS3; ---; #15 to emit 1; ctr -; BT; reset; ---; #34.
   5. Add to explanations of selected operations: (63), CS 11, half-add (S), ctr + other.
   6. Change CS11 from

T	o		●	P		●	half-add	●	P
N	●	#44	●	SUP	to	●	ctr +	●	#15
C	●	SUP	●	#26		●	(63)	●	#26

2. The difficulty of rounding off $.99999999 \geqslant x \geqslant .99999995$ to 1.0000000 is a compromise. Only if a result is in this range is it so rounded. If this rounding were eliminated, all other operations which "overflow" would be incorrectly rounded. Since these few cases are an extremely small percentage of the possible results, it does not seem worthwhile to change the control panels.

3. The third case mentioned is inherent in the design of the panel. The case of inaccuracies in multiplying non-shifted numbers can also occur in division or addition. For example:

   00000234 50 + 01791342 50 = 13062832 46

   actually gives: 13063000 46

   00000234 50 + 17913421 45 = 41313421 45

   actually gives: 41300000 45

It should also be noted that in the operation (A + B)/ X, the result of the addition is not shifted before the division takes place. Specifically, for A = 12345876, B = 12345731, X = 7, the result of the above operation is 207 rather than 20714286.

Technical Newsletter No. 6. 13 Digit Floating Decimal Model II CPC - Dura W. Sweeney.

Dr. F. M. Verzuh of M.I.T. brought the following inaccuracy in this description to our attention.

In using this panel to perform the following division:

$$Q = \frac{A}{\bar{B}} = \frac{4.472135955000}{8.944271910000}$$

An incorrect quotient of 0.4999899999954 is obtained instead of the correct value of 0.5.

Mr. Sweeney has suggested the following changes in the Panel A wiring to eliminate this error.

PANEL A

Prog.	Exit 3
17	In 5
18	In 5
19	In 4
.	.
.	.
.	.
27	In 1
28	Out 2
.	.
.	.
.	.
30	In 5

In the same article the 605 Planning Chart under Panel B, page 39, indicates that Repeat Selectors #3, 4, and 10 are to be picked up by exit 1 of Program Step 59. Experience has proved that multiple wiring of Repeat Selectors has been extremely troublesome. For this reason the following changes to Panel B are suggested.

PANEL B

Prog.	Exit 1	Exit 3
47	Prog. Rpt., Rpt. Del PU	RS #10 PU
.	.	.
.	.	.
.	.	.
59	RS #3, 4	GS1 & 2 RI

Technical Newsletter No. 5, p. 10 line 10 read 1570796350 for 1570596350.

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

Litho in USA