WORKING PAPER W-53-20

U. S. NAVAL ORDNANCE PLANT
Indianapolis, Indiana
RESEARCH AND TEST DEPARTMENT

A FIVE ADDRESS, THREE OPERATION CARD PROGRAMMED CALCULATOR

by

Ralph Hafner and Robert L. LaFara

Copy No. 84

Presented by

K. L. Nielsen

at the

IBM CPC Conference, Endicott, N. Y.

17 August 1953

PRELIMINARY DATA

This is an informal report and is transmitted for information only. The data presented are tentative and subject to later revision.

17 August 1953

[illegible]
Navy—DPPO [illegible], Great Lakes, Ill.

U. S. NAVAL ORDNANCE PLANT
Indianapolis 18, Indiana
Research and Test Department

A FIVE ADDRESS, THREE OPERATION CARD PROGRAMMED CALCULATOR

by

Ralph Hafner and Robert L. LaFara

This paper presents a complete description of the wiring and operation of five-address control panels for the Model II CPC.

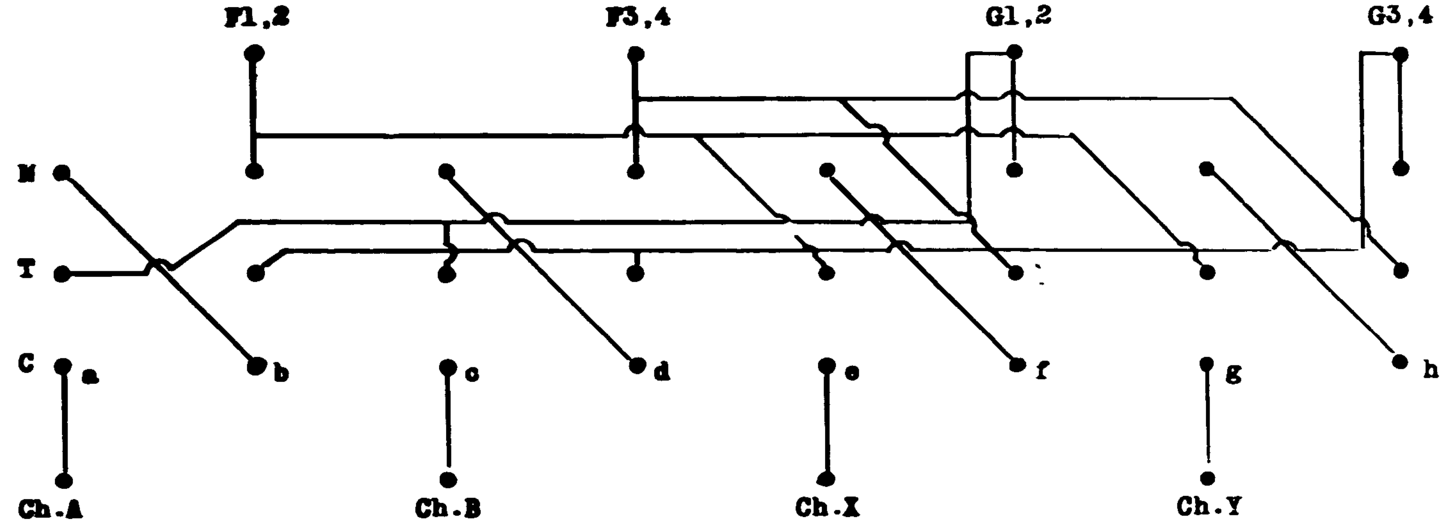
A FIVE ADDRESS, THREE OPERATION CARD PROGRAMMED CALCULATOR

Addition, subtraction, multiplication and division of two eight-digit factors can be completed in much less than the normal calculate time allotted to the 605 calculator during a CPC card cycle. It would be desirable, therefore, to do more than one operation on more than two factors during a single card cycle. Since the CPC is basically a three address machine, this would seem impossible; however, five address control panels may be wired for the Model II CPC which will have the following features:

1. Eight 8-digit counters available on all four input channels. Eight digit factors from cards available on all four input channels.
2. Factors from 941 storage available on any two of the four input channels.
3. Three operations available in two different orders.
4. Table look-up with linear interpolation.
5. Special functions.

1. For convenience, let us call the two additional addresses channel X and channel Y. In order to have the first feature listed, the exits of the eight counter groupings are chain wired through selectors to the entries of channel A and with another chain to channel B; this wiring is done in the normal fashion. The counter exits are also wired into the field selector transfer hubs. The field selector is picked up by the channel X code to gate the proper counter or card columns to the calculator at read-in time. Another chain of selectors, which is controlled by the channel Y code is also used to gate a counter or card columns to the calculator. The field selector is used for channel X counter wiring in order to have co-selectors left for what is called channel interchange.

2. Normally, the channel A hubs are connected to factor storage 1 and 2, channel B hubs to factor storage 3 and 4, the commons of the field selector (which might be called the channel X hubs) to general storage 1 and 2, and the last commons of the channel Y chain to general storage 3 and 4. In order to read numbers from 941 storage to general storage in the calculator (feature 2 listed above) the channel A hubs are wired through selectors so that they may be read into any one of three places in the calculator; F1,2; G1,2; G3,4. If channel A is selected to read into G1,2 then the channel X data instead of reading into G1,2 is read into F1,2; likewise, if data is read from channel A into G3,4, then the channel Y data is read into F1,2. The channel B hubs are selected in a similar fashion. In programming, care must be taken that channel A and B data are not sent to the same place. Figure one shows the wiring of one position for channel interchange. The read-in impulses must also be selected through these selectors in case that it is desired to use a number already in the calculator when interchanging channels.



a and e are picked up by a code to interchange A and X
b and g are picked up by a code to interchange A and Y
c and f are picked up by a code to interchange B and X
d and h are picked up by a code to interchange B and Y

FIGURE 1. Channel Interchange Wiring

3. In the calculator, one program sweep is used for multiple operations. The programming for add, subtract, multiply and divide only takes sixteen program steps and hence may be wired three times on the 605 control panel. The following is the programming of the multiple operation sweep.

Operation	Step	R.O.	R.I.		Notes
	1				1. Ri 3rd on add & sub; Ri 5th on div; Ri 1st on mult.
All	2	F12RO	ECRI+	See Note 1	
Add,Sub	3	F34RO	See Note 2	Ri 3rd	
Mult	4	ECRO	F12RI	Ro 6th	2. ECRI+ on add; ECRI- on sub.
Mult	5	ECRR	MQRI		
Mult	6	F34RO	Mult+		3. Ri 6th on div.
Mult	7	F12RO	MQRI		4. Normally F12RO; G12RO when order of operations is changed.
Mult	8	ECRR	F12RI	Ro 6th	
Mult	9	F34RO	Mult+		
Div	10	F34RO	Div		5. Normally G12RO; G34RO when order of operations is changed.
Div	11	ECRR	F12RI		
Div	12	F12RO	ECRI+	Ri 6th	
Div	13	MQRO	F12RI		6. Normally F12RI; G12RI when order of operations is changed.
Div	14	F34RO	Div	Reset	
Div	15	MQRO	ECRI+		
Div,Mult	16	F12RO	ECRI+	See Note 3	7. Normally F12RI; G34RI when order of operations is changed.
All	17	1/2 adj.		Ri 2nd	
	18				
All	19	ECRR	F12RI	Ro 3rd	
All	20	See Note 4	ECRI+	See Note 1	
Add,Sub	21	See Note 5	See Note 2	Ri 3rd	
Mult	22	ECRO	See Note 6	Ro 6th	
Mult	[illegible]	ECRR	MQRI		
Mult	24	See Note 5	Mult+		
Mult	25	See Note 4	MQRI		
Mult	26	ECRR	See Note 6	Ro 6th	
Mult	27	See Note 5	Mult+		

Operation	Step	R.O.	R.I.		Notes
Div	28	See Note 5	Div		1. Ri 3rd on add & sub; Ri 5th on div; Ri 1st on mult.
Div	29	ECRR	See Note 6		
Div	30	See Note 4	ECRI+	Ri 6th	
Div	31	MQRO	See Note 6		2. ECRI+ on add; ECRI- on sub.
Div	32	See Note 5	Div	Reset	
Div	33	MQRO	ECRI+		3. Ri 6th on div.
Div,Mult	34	See Note 4	ECRI+	See Note 3	4. Normally F12RO; G12RO when order of operations is changed.
All	35	1/2 adj.		Ri 2nd	
	36				
All	37	ECRR	See Note 7	Ro 3rd	5. Normally G12RO; G34RO when order of operations is changed.
All	38	F12RO	ECRI+	See Note 1	
Add,Sub	39	G34RO	See Note 2	Ri 3rd	
Mult	40	ECRO	F12RI	Ro 6th	6. Normally F12RI; G12RI when order of operations is changed.
Mult	41	ECRR	MQRI		
Mult	42	G34RO	Mult+		
Mult	43	F12RO	MQRI		7. Normally F12RI; G34RI when order of operations is changed.
Mult	44	ECRR	F12RI	Ro 6th	
Mult	45	G34RO	Mult+		
Div	46	G34RO	Div		
Div	47	ECRR	F12RI		
Div	48	F12RO	ECRI+	Ri 6th	
Div	49	MQRO	F12RI		
Div	50	G34RO	Div	Reset	
Div	51	MQRO	ECRI+		
Div,Mult	52	F12RO	ECRI+	See Note 3	
All	53	1/2 adj.		Ri 2nd	
	54				
All	55	ECRO	F12RI	Ro 3rd	
All	56	F12RO	G12RI		
All	57	F12RO	F34RI		
All	58	F12RO	G34RI		

To summarize the calculator wiring, steps 2 through 17 perform operation one on the factors in F1,2 and F3,4. Step 19 transfers the answer back to F1,2. Steps 20 through 35 perform operation two on the previous result and what is in G1,2. Step 37 transfers this answer back to F1,2. Steps 38 through 53 perform operation three on the previous result and what is in G3,4. Steps 55 through 58 transfer the final answer back to F1,2; F3,4; G1,2; G3,4. This wiring permits all operations of the form [(A ① B) ② X] ③ Y = C; where ① means operation one and may be add, subtract, multiply, or divide, likewise with ②, ③. There are 64 such possible operations and they are listed below.

A+B+X+Y	A-B+X+Y	AB+X+Y	A÷B+X+Y
A+B+X-Y	A-B+X-Y	AB+X-Y	A÷B+X-Y
(A+B+X)Y	(A-B+X)Y	(AB+X)Y	(A÷B+X)Y
(A+B+X)÷Y	(A-B+X)÷Y	(AB+X)÷Y	(A÷B+X)÷Y
A+B-X+Y	A-B-X+Y	AB-X+Y	A÷B-X+Y
A+B-X-Y	A-B-X-Y	AB-X-Y	A÷B-X-Y
(A+B-X)Y	(A-L-X)Y	(AB-X)Y	(A÷B-X)Y
(A+B-X)÷Y	(A-B-X)÷Y	(AB-X)÷Y	(A÷B-X)÷Y
(A+B)X+Y	(A-B)X+Y	ABX+Y	(A÷B)X+Y
(A+B)X-Y	(A-B)X-Y	ABX-Y	(A÷B)X-Y
[(A+B)X]Y	(A-B)XY	ABXY	(A÷B)XY
[(A+B)X]÷Y	(A-B)X÷Y	ABX÷Y	(A÷B)X÷Y
(A+B)÷X+Y	(A-B)÷X+Y	AB÷X+Y	(A÷B)÷X+Y
(A+B)÷X-Y	(A-B)÷X-Y	AB÷X-Y	(A÷B)÷X-Y
[(A+B)÷X]Y	[A-B)÷X]Y	(AB÷X)Y	[(A÷B)÷X]Y
[(A+B)÷X]÷Y	[(A-B)÷X]÷Y	(AB÷X)÷Y	[(A÷B)÷X]÷Y

It would be very desirable if the parentheses could be rearranged so that operations like AB+XY could be performed (feature 3). This may be done by picking up a calculate selector which changes the programming such that F12RO is replaced by G12RO on steps 20, 25, 30, and 34; G12RO is replaced by G34RO on steps 21, 24, 27, 28, and 32; F12RI is replaced by G12RI on steps 22, 26, 29, and 31; and F12RI is replaced by G34RI on step 37. This changes steps 20 through 35 so they perform operation two on the factors in G1,2 and G3,4. Step 37 transfers this answer into G3,4 so that steps 38 through 53 perform operation 3 on the result of steps 2 through 17 and the result of steps 20 through 35. This permits all operations of the form (A ① B) ③ (X ② Y); there are 64 such operations but 32 of these are duplicates of the previously mentioned ones. Due to scaling difficulties, it may be desirable to use one order in preference to another. For example, A+B+X+Y is equivalent to (A+B)+(X+Y) but if A = 4, B = 3, X = 6 and Y = -5 the wrong answer will result if the first method is used since one of the partial answers would exceed 10. The 32 operations which are not duplicates of the above are listed below.

(A+B)+XY	(A-B)+XY	AB+XY	(A+B)+XY
(A+B)+(X+Y)	(A-B)+(X+Y)	AB+(X+Y)	(A+B)+(X+Y)
(A+B)-XY	(A-B)-XY	AB-XY	(A+B)-XY
(A+B)-(X+Y)	(A-B)-(X+Y)	AB-(X+Y)	(A+B)-(X+Y)
(A+B)(X+Y)	(A-B)(X+Y)	AB(X+Y)	(A+B)(X+Y)
(A+B)(X-Y)	(A-B)(X-Y)	AB(X-Y)	(A+B)(X-Y)
(A+B)+(X+Y)	(A-B)+(X+Y)	AB+(X+Y)	(A+B)+(X+Y)
(A+B)+(X-Y)	(A-B)+(X-Y)	AB+(X-Y)	(A+B)+(X-Y)

4. The fourth feature of this board is table look-up with linear interpolation. If $f(a_n)$ is the table function corresponding to the table argument a_n, then the formula for finding the function $f(x)$ at the argument x, which is not necessarily a table entry, using linear interpolation is:

$$f(x) = f(a_n) + \left(\frac{x - a_n}{a_{n+1} - a_n}\right)[f(a_{n+1}) - f(a_n)] .$$

When this formula is written in the form:

$$f(x) = (x - a_n) Q_n + f(a_n)$$

where

$$Q_n = \frac{f(a_{n+1}) - f(a_n)}{a_{n+1} - a_n}$$

then it is easily seen that the operation is one of the form (A-B)X+Y = C.

This means that the entire operation may be performed on a single card cycle without taking any program cycles. The argument is read onto channel A and into counter 8 prior to running the table deck. The first table card has a special control punch which disconnects the negative balance test exit of counter 8 from its normal function and connects it to the pick-up of a selector by means of a latch selector which is not dropped out until the end of the table. The table must be arranged such that the arguments are always positive and in ascending order but they need not be spaced at equal intervals. The second table card reads the first table argument into counter 8 negatively and each card thereafter reads the difference between two successive arguments into counter 8 negatively. Counter 8 will go negative when the sum of the differences plus the first table argument exceeds the argument read in; then the negative test exit will give the impulse to pick up a selector, this selector emits the instructions for subtract, multiply, and add into the coding selectors for operation one, two, and three respectively. The cards are punched so that a_n is in the card columns for entry onto channel B, Q_n is in the card columns for entry into channel X, and $f(a_n)$ is in the card columns for entry into channel Y; the difference $a_{n+3} - a_{n+2}$ is entered in a special field from which it is read into counter 8 negatively, it is displaced two cards as indicated by the subscripts in order that the timing comes out correctly. The argument x stays on channel A until the interpolation calculation is made. New information, a_n, Q_n, and $f(a_n)$ is read in on every card but is not used until the proper time. After the interpolation, the answer is read on to all channels but is erased from B, X, and Y when new data is read in. The last table card clears counter 8 and returns it to its normal function. At the end of the table, the answer is available on channel A.

5. Since the multiple operation wiring requires only one program sweep, there are two sweeps still available for other programming. One of these is used to compute special functions and is essentially the wiring described by Bill Heising in Technical Newsletter No. 3. The primary difference is that the number of terms necessary for sin A, cos A, exp A, exp (-A), sinh A, and cosh A is predicted by the calculator instead of the programmer. This is done by evaluating the formula $N = 2|1.5\,A| + 11$ where only the integral part of $|1.5\,A|$ is retained. In case N must be even, then $N = 2\,|1.5\,A| + 12$. This is sufficient to assure that $A^N/N! \leq 10^{-7}$. The third program sweep has been kept free for temporary wiring of functions that are not needed frequently; as an example, this sweep has been used to compute the Bessel functions, $J_n(A)$, where $n = 0,1,2,3,4,5,6,7,8,9$.

The channel C wiring is standard as well as most other wiring not specifically mentioned. Since this is a fixed decimal system, the high order positions of the electronic counter are wired into the compare unit to provide an overflow stop in case $C \geq 10$.

As an example of the savings afforded by the multiple operation system, consider the evaluation of the polynomial

$$y = ax^4 + bx^3 + cx^2 + dx + e \ .$$

First re-write in the factored form:

$$y = \{[(ax + b)\ x + c]\ x + d\}\ x + e \ .$$

With a single operation, three address system, this would probably be performed in the following manner:

Card No.	Operation
1	$ax = C_1$
2	$C_1 + b = C_2$
3	$C_2\ x = C_3$
4	$C_3 + c = C_4$
5	$C_4\ x = C_5$
6	$C_5 + d = C_6$
7	$C_6\ x = C_7$
8	$C_7 + e = y$

With the multiple operation system, provided numbers are allocated to storage properly, the computation would be done as follows:

Card No.	Operation
1	$(ax + b)\ x = C_1$
2	$(C_1 + c)x + d = C_2$
3	$C_2\ x + e = y$

The above case is somewhat idealized since frequently intermediate answers must be stored. Furthermore, if all three operations are multiplications and/or divisions, the calculator may not have time to finish the calculation and will hold up the card feed resulting in a loss of time. On typical jobs, the multiple operation system increases the speed by a factor of one and one-half.

Actually, the number of selectors needed to wire the 418 control panel exceeds the number available. In wiring this board, some back circuits were included; however, the programmer could avoid these by remembering certain rules but this is clumsy. By adding ten selenium rectifiers to the machine, the back circuits were eliminated and some pilot selectors were freed for other purposes.

Prepared by

Ralph Hafner

Ralph Hafner
Mathematics Division

Robert L. LaFara

Robert L. LaFara
Mathematics Division

Approved by

K. L. Nielsen

K. L. Nielsen
Head, Mathematics Division

DISTRIBUTION LIST

1. & 2. BuOrd (Re8)

3. W-53-20 (via W.A. Key, A and B)

4. Engineering Department

5. Mathematics Division - 810

6. Physics Division - 820

7.

8.

9.

10. - 85. Participants IBM-CPC Conference