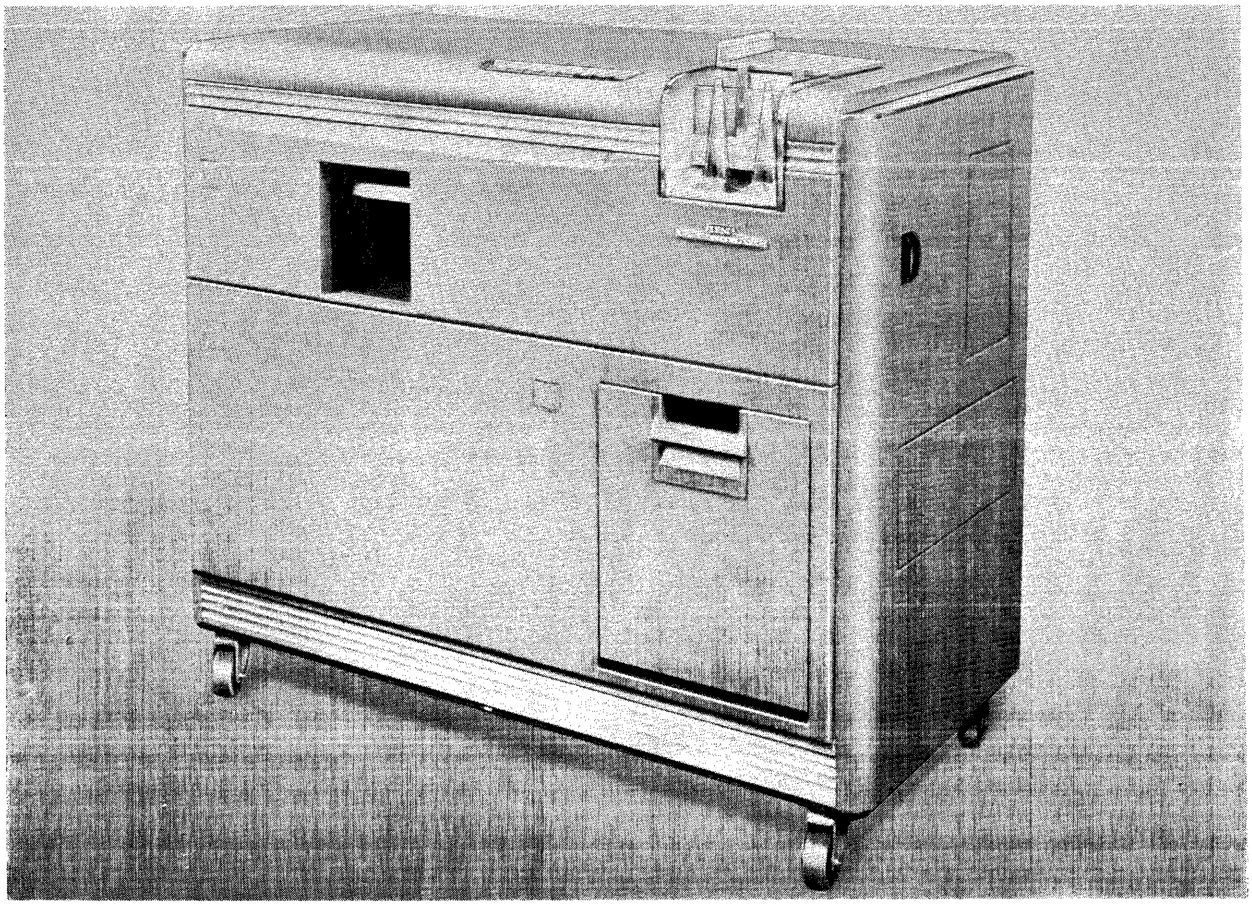
IBM Reference Manual

# 602 Calculating Punch

This edition, Form A22-0506-1, is a minor revision of the preceding edition and does not obsolete Form A22-0506-0. The principal change is the addition of a section on Series 50 operation, and changing the machine number from 602-A to 602.

# Contents



IBM 602 CALCULATING PUNCH

# IBM 602 Calculating Punch

THE IBM 602 Calculating Punch, Models 1 and 2, reads factors punched in IBM cards and performs the mathematical operations of addition, subtraction, multiplication, and division. The results of these calculations can then be punched into the same cards from which the factors were read or into trailer cards. The standard machine can multiply a 22-digit multiplicand by an 8-digit multiplier to obtain a 30-digit product. A 15-digit dividend can be divided by an 8-digit divisor to obtain and punch an 8-digit quotient. It is possible to expand both the multiplier and the quotient beyond the eight digits by additional calculations. Group multiplication can be performed with either the multiplier or multiplicand as the group factor. Many other group operations can be performed such as the accumulation of factors from a group of cards and the punching of summarized or calculated results in the last card of a group.

The 602 can test for plus or minus amounts in the counters as the result of a calculation and recognize zero balances. These decisions may be used to alter the type of calculations that will be performed or even to eliminate certain calculations.

Punched results from any operation may be verified on a separate run of the cards through the machine. Programming can cause an error to stop the machine, or to punch a 12 for all correct cards and an 11 for all error cards.

To describe the operation of the 602 Calculating Punch, this manual is divided into nine sections:

1. Operation Principles
2. Single-Card Operations
3. Selection
4. Multiple-Card Operations
5. Checking Operations
6. Typical Applications
7. Operating Suggestions
8. Series 50, 602 Operations, Models 50 and 51
9. Appendix

The functions of the machine are illustrated with planning charts and wiring diagrams. The control panel hubs are explained as they are first introduced. Ready reference to the function of each control panel hub may be found in the index.

The sections are arranged in order of complexity, and each is explained in terms of knowledge of features already covered in preceding sections. To learn machine operation, therefore, most effective results will be obtained by reading the sections in order.

Section 8 describes the capacity of the IBM 602 Calculating Punch, Models 50 and 51, and should be carefully studied if a problem is to be planned and wired on a Series 50 model.

The typical applications section shows complete wiring for each example and illustrates the combined use of many of the features that have been individually explained on preceding pages.

# Operating Principles

ALL OPERATIONS of the 602 are performed in steps or operating cycles which occur at the rate of 200 per minute or 12,000 per hour. The number of cycles required for a card depends on the size of factors and the type of problem.

The 602 consists of three basic parts: storage units, counters (arithmetic unit), and the program unit.

### Storage Units

The storage units in this machine, of electro-mechanical design, store numerical factors as they are read from the card for temporary storage during calculation, and also serve as output for punching the results. In the standard machine there are six 12-position storage units divided into a right and left section. These two sections in each storage unit can be used to hold separate factors. Factors can be read out of either section of the unit independently, but information to be read must be entered into both sides at the same time. If two factors are not read in at the same time, one would be erased whenever the storage unit was instructed to accept the next. All storage units may be used as working storage during calculation; however, only two of the six storage units may be used as output for punching results.

### Counters

All addition, subtraction, multiplication, or division is done in a counter. There are six independent counters in the 602 composed of electro-mechanical type counters. Three of the counters are 6-position counters; the other three are 4-position counters. These counters may be operated separately, or they may be coupled together to perform as larger-capacity counters. It is possible to combine all the counters together, thus making the maximum-size counter on the standard 602 thirty positions. Two additional counters can be obtained at the option of the customer.

Electro-mechanical counters of the 602 can also be used as storage units because information can enter them directly from the cards and be held during all or part of the calculation. Factors or results can remain in some of the counters while operations are performed in other counters.

### Program Unit

The program unit of this machine provides a series of 12 sequential steps which are used to instruct the machine to calculate a given set of factors. Each program step on the machine is equivalent to one machine cycle except when a multiply or divide command is given. These arithmetic instructions cause as many machine cycles to occur on a single program step as are necessary to develop a product or a quotient.

The program unit is designed to permit the steps to be taken in ascending sequence or to skip steps when it is desirable to eliminate program steps for certain codes in the cards or according to the sign of a counter.

Not all of the steps may be necessary to complete a given problem; consequently, the series of steps may be ended by control panel wiring.

### Operating Principles, Series 50

All operations of the 602 Calculating Punches, Series 50, are performed in steps or operating cycles which occur at the rate of 150 per minute or 9,000 per hour.

A 602 Model 50 or 51 is equipped with counters 1, 2, 3, and 6, storage units 1, 2, 4, and 7, and program steps 1 through 6. Each of these units operates just as it does in the standard 602. A Model 51 incorporates the above units in the machine without the divide feature. A Model 50 incorporates the above units and includes the divide feature.

The wiring principles for the Series 50 Calculating Punches are identical to those illustrated for the standard 602. Planning a problem for a Series 50 602 simply requires programming the problem with the counters, storage units and program steps listed above.

## CARD FEEDING AND PUNCHING

THE FEED schematic diagram of operation (Figure 1) shows the path of the card through the machine. An understanding of the general way in which the machine feeds, reads, calculates, and punches is necessary for analyzing and wiring a problem for the 602.

The feeding of cards, face down, 9 edge first is started by depressing the start key. This action moves the first card into the machine and past the 20 control brushes. At this station control X's or digits may be read. At this point a card lever senses the first card and feeding stops until all commands wired in the panel are executed. No actual results are calculated because no factors have been read in. The steps just described are called a *run-in reset;* their purpose is to clear out all units which are used with this control panel prior to the calculation of the first card. The first card then moves past the reading brushes where factors are read for making the calculation. The second card is at the control brushes one machine cycle later. Therefore, digits can be read from the control brushes and entered into either storage or the counters. (The choice of reading or control brushes for picking up control punching is an important one.)

The cards continue to feed in a semicircular path. If no previous card is being punched, the card passes directly to the punch bed and is then skipped into position for punching the first column of the first result field. In this case the card is in position for punching by the time the machine has reached the second or third calculating cycle. However, punching of one card may continue through the reading and calculation of the succeeding card. When this condition occurs, the second card will be delayed in the punch entry station until the first card that is being punched is ejected. The reading of a third card cannot take place until the first card punched has been ejected and the following card is in the punching rack. (The read cycle is automatically delayed until the punch entry station is empty.)

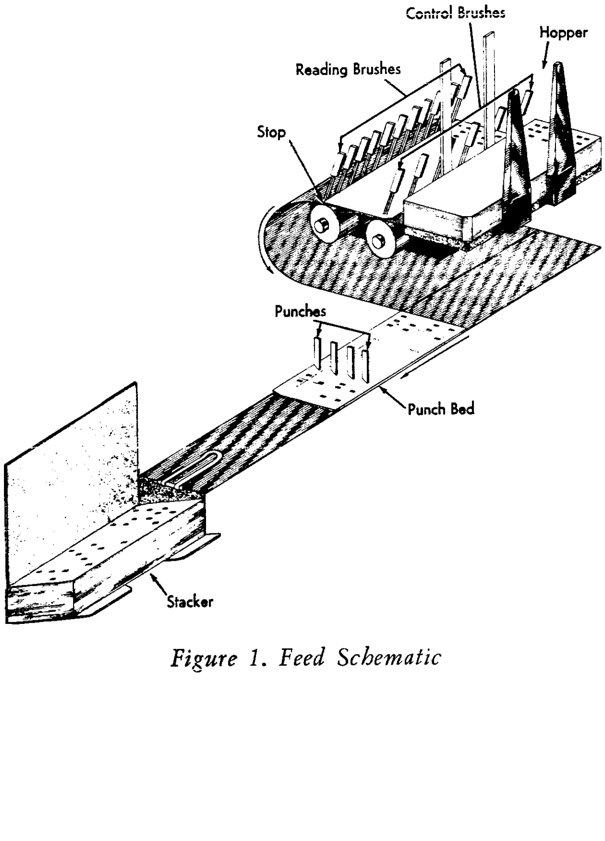


*Figure 1. Feed Schematic*

There can never be more than three cards in the feed unit: one in the punch bed, a second card waiting in the punch magazine, and a third between the two sets of reading brushes.

The instruction to punch may be given on any program step or on the following read cycle, and punching will begin when the card reaches the first column wired to punch. If the first column to be punched is reached before the instruction to punch is received, the card waits at that column until punch is impulsed. The time necessary to complete punching depends upon the number of columns to be punched. Punching proceeds at an approximate rate of four columns per cycle.

For example, if a punch storage unit is wired to punch columns 51-58 on program step 3 and the card is not delayed in the punch magazine, it will probably have reached column 51 by the time program 3 occurs and punch is impulsed. Punching actually begins at the start of program 4, and since eight columns are to be punched, punching is completed by the end of program 5, two cycles later.

If punching is still taking place in the previous card, the card in the punch magazine waits to be fed until this punching is completed. In this case the punch instruction given on program 3 waits until the card reaches column 51 before it is executed.

The variable time relation between simultaneous punching and calculating requires careful consideration, particularly in the control of selectors, because the rate of punching rarely has a definite time relationship to calculate program steps.

The 602 uses the serial method of punching. The card feeds lengthwise, one column at a time, and is positioned under the punching unit. As each column is positioned under the punches, an emitter which moves in step with the card columns sends an impulse through control panel wires to test for the digit stored in a storage unit or relay network for X's or 12's.

The impulse which controls the punch magnets in each column is directed from one storage position to the next by means of the column emitter. A schematic diagram of this principle appears in Figure 2.

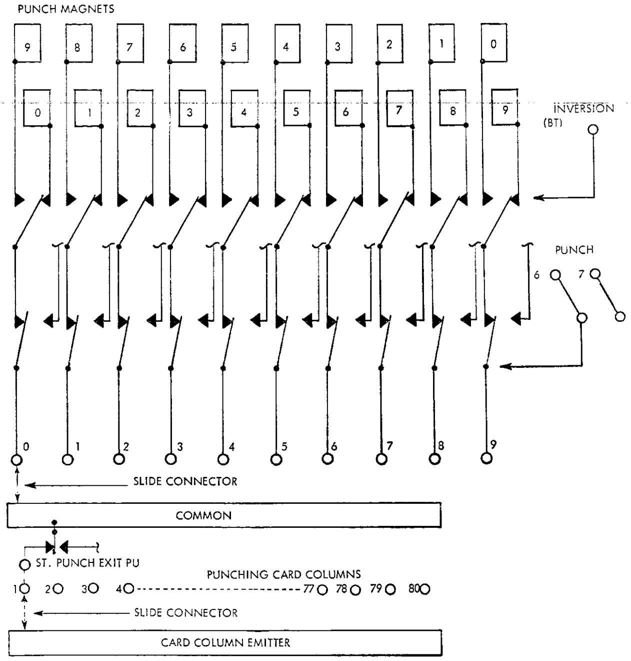


*Figure 2. Punch Schematic*

A wire from the column emitter on the control panel carries an impulse to a punch exit hub of the storage unit position which is to be punched. This impulse is directed to one of the ten digit punch magnets, depending on the digit value held in that storage position. The activation of a punch-unit magnet causes a corresponding hole to be punched in the card. The card automatically moves on to the next column, where the process is repeated. The first column after a punching field should be wired to SKIP. These skip hubs are entries to a magnet which moves the card rack to the next skip-stop insert.

The 602 has a built-in interlock to prevent the punch storage units from reading in a new value while punching is going on. As soon as the storage unit receives an impulse to punch, an interlock relay is picked up and held until the units position of the storage unit has been punched. Therefore, the machine waits until the punching from the storage unit is completed before reading in a new value. Punching from the units position of the storage unit signals completion of punching. Thus, whenever punch is impulsed, the units position of all punch storage units used must be wired to punch.

An understanding of the flow of information between the various machine components and the manner in which these units operate facilitates planning a problem on the 602. A schematic diagram of operations, Figure 3, shows the entry of information from the card (input), the flow of information in making a calculation, and the punching of results (output). The diagram is made from sections of the control panel rearranged to show the flow of a calculation.

Factors are read from the card by wiring from the reading hubs, which are exits for the reading brushes. These factors may be entered into any of the six counter groups or into storage units. Some of the units are assigned a specific function: storage unit 1R is always used for entry of a multiplier or divisor; units 6L, 6R, 7L and 7R must always be used for punching a result; counter 1-2-3 must always be used for entering a dividend.

Counters and storage units also have exits from which information is transferred to storage units or other counters. Entry into and exit from any type of unit cannot be made on the same step. Multiplication and division are performed by a special system of repeated transfers between storage units and counters, as explained under "Multiplication and Division."

Results developed directly in the counter, or previously developed and stored in another storage unit, can be transferred to storage unit 6 or 7 for punching. Results to be punched may be entered into these units at one time and punched from them at another time. These units have special exits provided for wiring to the punching hubs.

Punching hubs may be thought of as inlets for punching results in any of the 80 columns of the cards. Successive results punched while other calculations are being performed must be punched in logical sequence from left to right and the cards must be designed with this sequence in mind.

## OPERATING KEYS, LIGHTS, AND SWITCHES

*Main Line Switch.* The main line switch is located to the right of the feed hopper on the right side of the machine. When the switch is on, power is supplied to the calculating punch. An unlabeled light signals that the machine is ready to operate.

*Start Key.* Depressing this key feeds the cards and starts calculation.

*Stop Key.* Depressing this key stops the machine immediately, even though the machine may be in the middle of a calculation. The machine may be restarted by depressing the start key and the calculation will continue.

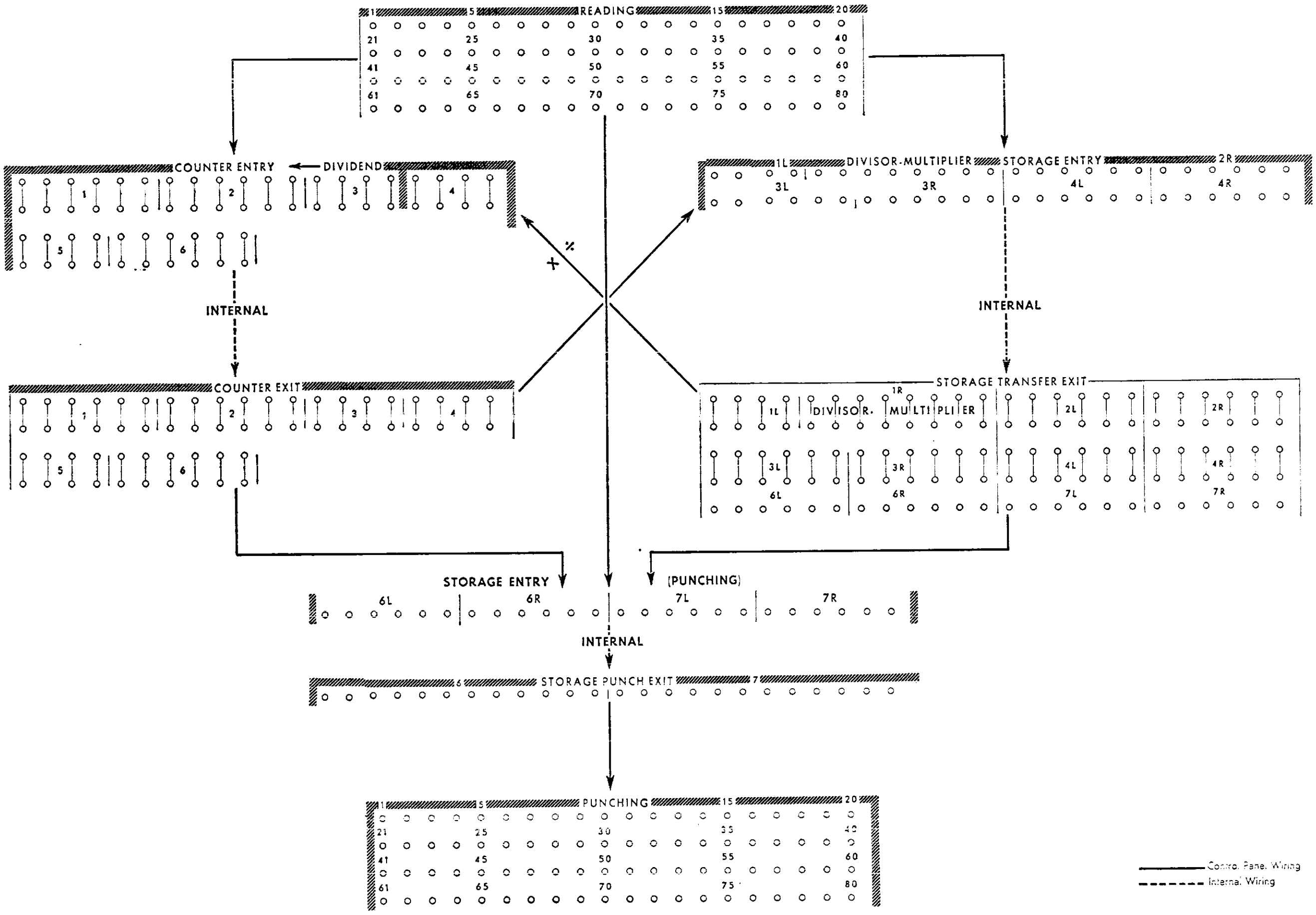


*Figure 3. Machine Flow Schematic*

*Reset Key.* Depressing this key turns off the compare light.

*Unlabeled Light.* This light glows as soon as the main line switch is turned on. It turns off as cards pass through the machine and turns on when the machine stops.

*Compare Light.* This light signals an error detected during a checking operation and must be turned off by depressing the reset key. This light also indicates errors in control panel wiring, card feed failure, or punch jams. See "Operating Suggestions."

## OPERATION

To PREPARE this machine for operation, insert a properly wired control panel in the rack at the lower right of the machine. When the lower front cover is opened, the rack tilts forward and upward for convenient insertion of the panel. Wires must not be permitted to fall between the panel and the rack or to interfere with the operating mechanism at the sides for closing the front cover.

An adjustable skip bar (Figure 4) must be inserted in the machine immediately behind the punch bed located under the upper right cover of the machine. This skip bar contains 80 positions, one for each column of the card, in which a small insert is placed for the first column of each field to be punched after a skip. The small notch under the triangular head of the insert faces toward the back of the bar. The skip bar is most easily removed by pushing it up and slightly to the right with the thumb and the forefinger. The card rack holding the skip bar should be all the way to the left before the skip bar is removed.

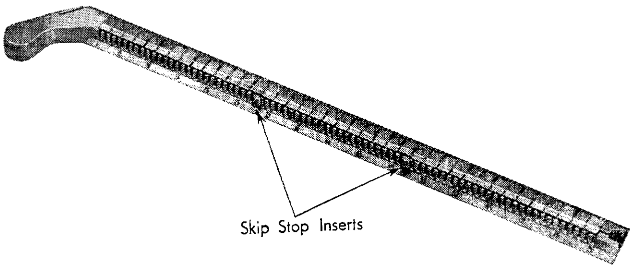


*Figure 4. Adjustable Skip Bar*

Single skip stop inserts cannot be placed in adjacent positions. Use a special double skip stop insert for this purpose.

For operations requiring no punching, a skip bar without inserts must be used.

Normally, the cards run out of the hopper automatically. If the machine stops for any reason, cards in the machine can be run out by opening the control panel cover. The cards are ejected into the stacker without calculation or punching. Before the control panel cover is opened, cards in the hopper should be removed, or they also will run through the machine and be ejected into the stacker.

## PLANNING THE OPERATION

A problem to be calculated on the 602 should first be developed on a planning chart (Figure 5) to determine the most logical arrangement of steps to be taken by the machine. With this form the problem can be developed by entering the actual figures for the test problem and the functional commands used to accomplish the result. Notes on selectors, negative balance testing, and so on, should all be indicated on the planning chart.

A column is provided in the chart for each of the operating features shown in the schematic diagram, except card reading and punching. A line is provided for each of the program steps which the machine can take in performing a calculation. In the example, these lines are filled in for the steps used to perform the simple crossfooting operation, $A + B = T$. The first step is the reading of the card. Factor $A$ is read into counter 6, and factor $B$ is read into storage unit 2R. Following the reading cycle is the first of a series of program steps used in a calculation. On the first of these (program step 1), factor $B$ is read out from 2R to counter 6, where it is added to produce the total $T$. On the final step, the result $T$ is read out of counter 6 to storage unit 6R for punching, and the punching unit is controlled to start punching immediately as indicated by the word "punch" followed by an arrow. During the final program step of the calculation, the machine is controlled to take no further steps for the first card, but to proceed with reading the next card on the next cycle.

The principal operation controls are located in the lower left section of the control panel, and, in Figure 5, this section has been superimposed over the planning chart to facilitate association between the op-

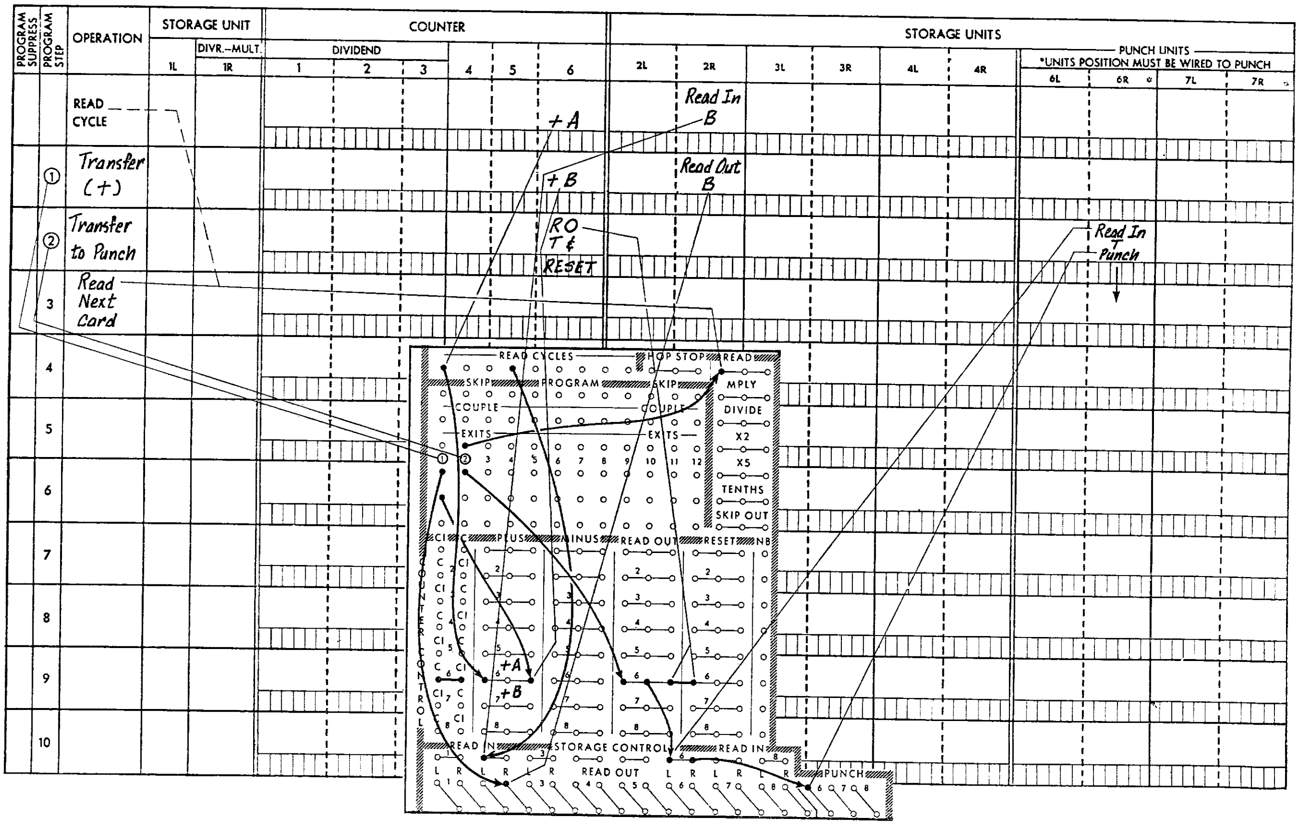


*Figure 5. Planning Chart*

erating features shown on the planning chart and the control hubs on the control panel. Each program step on the planning chart corresponds to the same set of program hubs on the control panel, as shown by the light connecting lines.

The nine read cycles hubs are control impulses which are active during the read cycle shown on the planning chart. Program exits, numbered 1 to 12 on the control panel, emit controlling impulses during each of the program steps shown vertically on the planning chart. Each program step has four exit hubs. They may be used to control counters to add, subtract, multiply, divide, read out, and reset. They may also be used to control read-in and read-out of storage units. The last program step of a given problem must always be used to signal the machine that the problem has been completed and that a new card can be read on the next cycle.

## CONTROL PANEL

The complete control panel is shown in Figure 6. Letters down the side and numbers across the top of the panel diagram facilitate reference to specific hubs.

Some hubs are entries and some are exits. Exit hubs are always connected to entry hubs either directly or by selection. The wiring may be changed to suit each new application. Wiring for reading from the cards, transferring from one unit to another, and punching are all accomplished on the right side of the panel. The left panel is used for controlling counters, storage units, selectors, and the functions required to complete a calculation.

All hubs performing similar functions are grouped together and labeled with their general title.

Hubs that are connected by vertical, diagonal or horizontal lines are alike or common. Their purpose is to eliminate the need for split wires.

Shaded hubs on this diagram are not features of the standard machine.

The control panel is explained in detail in a series of diagrams illustrating the many types of calculations which can be performed by the machine. As various operating features are used for the first time, their control hubs are explained with the wiring of that feature. After the first problem, the planning chart and the left side of the panel are explained together because of their close association. As each line of the chart is explained, the corresponding wiring on the left panel is likewise explained.

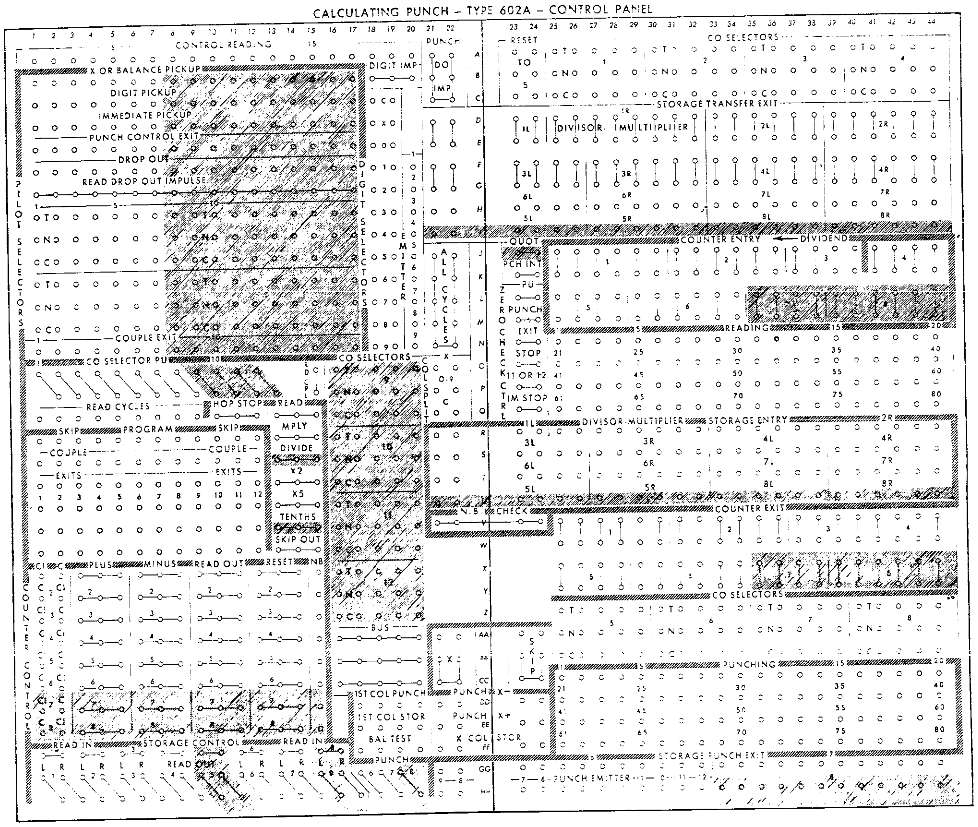


*Figure 6. Control Panel*

# Single-Card Operations

## CROSSFOOTING

$A + B = T$

THE NUMBER of factors that may be crossfooted is limited by the number of factors which may be stored in the machine from the cards. On the 602 this would represent 102 digits, the sum of the 72 storage positions plus the 30 counter positions.

This example shows a simple crossfooting operation in which factor $A$ is added to factor $B$ to give a total $T$, and a date is emitted on all cards.

### Control Panel Hubs

*Reading (N-Q, 25-44).* These hubs are exits from the 80 columns of the card as the card passes the reading brushes.

*Counter Entry (J-M, 24-44).* These hubs are entries to the 30 counter positions (standard) grouped as three 4-position and three 6-position counters numbered 1 through 6. For each position there is a common pair of entry hubs. Each counter must be controlled to add or subtract in order to make the corresponding entries accept impulses.

*Counter Exit (V-Y, 25-44).* These hubs are exits when the counter is controlled to read out to transfer information to other counters or to storage units.

*Storage Entry (R-U, 21-44).* These hubs are entries to the storage units. The storage unit may store either factors or results in the machine. The entry hubs accept information when they are controlled to receive information either from the card, from a counter, or from another storage unit. They accept and hold information like a counter but they do not add. A storage unit is cleared by its entry control; entry of a new factor or a result clears any previous information from the unit before the new information is accepted. The basic machine is equipped with six storage units, numbered 1, 2, 3, 4, 6, and 7. Each storage unit has 12 positions, divided (except unit 1) into left and right groups of six positions each. Storage unit 1R has a special use for multiplication and division in addition to its normal use as a storage unit, and is divided into four left-hand positions and eight right-hand positions. The eight right-hand positions must be used to store the multiplier and the divisor. For entry, all 12 positions of a storage unit operate together; therefore, entry into the left and right sides of the units, if both sides are used, must be made at the same time.

*Storage Entries 6 and 7 (T, 21-44).* These entries are used primarily to hold results for punching. Although these are the only storage units from which information can be punched, they are like all other storage units in that they may also hold information to be transferred for calculation.

*Storage Transfer Exit (D-I, 21-44).* These hubs are exits for the storage units when information is to be read out for transfer from a storage unit to a counter, or from one storage unit to another. At exit time, the left and right sections of storage transfer exits can be controlled to read out as independent units. This independent read-out allows the transfer of six digits or less at a time. A transfer can be made only when the particular half of the storage unit is controlled to read out.

*Storage Punch Exit (GG, 21-44).* These hubs are exits for storage units 6 and 7 when the machine is instructed to punch. They are the only storage units that can be used for punching results in the card. Although the hubs are labeled exits on the control panel, they are exits only in the sense that results stored are punched from these units. Because the units position of the storage unit signals the completion of punching, this position must be wired to punch. Storage units 6 and 7 should not be impulsed to read out and punch simultaneously.

*Punch Emitter (HH, 21-32).* These hubs are entries to the punch magnets at punch time only. They cannot be used to enter factors into storage or counters. An impulse from the column emitter wired from PUNCHING (card columns) to the punch emitter causes the corresponding hole to be punched in the column of the card. If the same digit is to be used in more than one column of the card, the emitter position may be split-wired to any number of punch positions. Care must be taken that two emitter positions are not connected to each other to cause an unwanted double punch.

*X Emitter (AA-CC, 21-22).* Six common exit hubs emit X impulses for punching. They are similar to the 11 impulse in the punch emitter.

*Skip (AA-CC, 23-24).* These hubs are entries to the skip magnet which moves the card rack to the next skip stop insert, or into the stacker when there is no other stop. The column following the last one punched must be wired to SKIP. When column 80 is punched, SKIP must be wired from column 1. Failure to impulse the skip hub stops the card rack in the column following the last punched position.

*Read Cycles (Q, 1-9).* These hubs emit impulses on each card feed cycle and are used to control counters or storage units to accept information from the card.

*Program Exits (R-W, 1-12).* These hubs emit impulses during all calculating cycles except card read. They occur one after another and control the machine as it performs a specific part of a calculation. During a program step, the four exit hubs are active. Each hub is independent. The machine can take 12 successive program steps and then be instructed to repeat, or the succession of steps may be ended at any time by wiring one of the exits of the last required program to the read hub. This stops the programming operation and causes a new card to feed.

*Read (Q, 13-15).* These three common hubs are entry hubs to stop the succession of program steps and cause a new card to feed. The new card feeds immediately following the program step in which the read hub receives an impulse. This hub accepts any impulse on program cycles but only an X impulse on the card read cycle. It may be necessary to wire the X impulse through a column split to prevent the program impulse from feeding back to any other control panel hub.

*Storage Read-in (FF, 1-16).* This pair of common hubs for each storage unit is used to control entry of information. Impulsing read-in resets the unit and controls all 12 positions of the storage unit to receive any digits reaching the entries on a read or program cycle. If read-in is impulsed, all storage entry positions not receiving an impulse are reset to zero.

*Storage Read-out (GG-HH, 1-16).* There are two separate read-out controls, one for each half of the storage unit. Impulsing either the left or right section causes the corresponding exits to emit. A storage unit may be impulsed to read out any number of times and the factor remains intact in storage until an instruction to read in is given. Impulsing storage read-out does *not* make the storage punch exits active. Storage punch exits are active only when storage unit 6 or 7 receives an impulse to punch.

*Storage Punch (GG-HH, 17-19).* These hubs are the independent controls for punching results from storage units 6 and 7. All 12 positions of the unit are controlled simultaneously. Any positions from the storage unit can be punched as long as the units position is also punched. The same impulse which is used for controlling read-in to the punch storage unit can be used to start punching. If punching is to take place at a later time than entry into the unit, different impulses would control read-in and punch. When the punch controls receive an impulse, punching starts on the following cycle and proceeds at the rate of four columns per machine cycle. Punching must be completed before the information can be transferred to another unit for calculation.

*Counter Controls (X-EE, 1-15).* Each of the six counter groups can be controlled to add, subtract, read out, and reset. Multiple hubs are provided for ease in wiring to control two or more counters simultaneously, or for controlling other operations simultaneously with counter operation. A counter cannot perform two opposing functions at the same time. Obviously, it cannot add and subtract at the same time; similarly, it cannot add or subtract and read out at the same time. However, it can both read out and reset at the same time. When a counter is controlled to add or subtract, the counter-entry hubs in the right-hand panel receive impulses wired to them;

when a counter is controlled to read out, the counter exit hubs emit impulses. Counters should not be controlled to reset on a multiply or divide step.

Internally, positive amounts are entered into the counter as complement figures, and negative amounts are entered as true figures. The result is always read out of the counter as the complement of the amount indicated in the counter. Thus, a positive amount is entered into the counter as a complement number and upon read-out is again inverted to a true number; conversely, a negative amount is entered into the counter as a true number and is converted to the complement on the read-out operation.

At the left of the counter controls are the counter coupling hubs. Because all counters on the 602 reset to 9's, the coupling hubs must be wired for correct addition or subtraction whenever a counter group is either used independently or coupled with another group. Failure to wire the coupling hubs causes each positive number entered in the counter to be one position less than the actual value. The CI (carry impulse) hub emits whenever the high-order position of the counter rolls from nine back to zero. The carry impulse must be connected back to the units position of the counter. The carry entry provides an inlet to the units position.

Two or all six counter groups can be coupled together to form a larger counter. When counters are coupled together, the CI of the counter containing the units position is wired to the C of the counter containing the high-order position in order to increase the capacity of the low-order counter by the number of positions in the high-order counter. The CI of the counter containing the high-order position is then wired to the C of the counter containing the units position to complete the carry-back.

Counters may be coupled in any order for addition or subtraction, but adjacent counters must be coupled during a multiplication or division operation. Be careful when coupling three or more counters, since the positions of the CI and C hubs alternate from one counter to the next. For example, the CI of counter 1 is located over the C of counter 2. Thus, the same pattern for coupling counter 1-2-3 cannot be used for coupling counter 4-5-6. Figure 7 shows counters 2 and 3 coupled for a 10-position counter. Counter 5 is wired for independent operation.

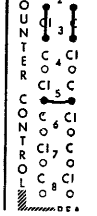


*Figure 7*

## Planning Chart and Control Wiring

To add factor *A*, punched 16784 in columns 40-44, to factor *B*, punched 321 in columns 45-47, requires a read cycle to enter the data to counter and storage and two program steps to complete the calculation and punch the total, 17105, in columns 48-52. Figure 8 shows the planning chart and control panel diagram. This example also includes emitting a date into columns 15-19.

1. To read in the factors *A* and *B* for calculation, *A* is entered into counter entry 6, and *B* is entered into storage entry 2R; each is entered from the proper reading brushes.

2. Each of the units used is told to read in as the card passes the reading brushes by a READ CYCLES impulse wired to read into counter 6 plus and to read into storage 2.

3. Factor *B* is read out from storage transfer exit unit 2R to counter entry 6 where it will add to *A*. This same wiring would apply for *A* minus *B*, except that factor *B* would be read into counter 6 minus, with corresponding changes in counter control wiring.

4. Program 1 instructs storage unit 2R to read out and to add the factor *B* in counter 6. At the end of this program step, counter 6 contains the result *T*.

5. The answer *T* is transferred from the counter to storage unit 6R, from which it is wired to punch in columns 48-52. An insert, represented by a small block triangle on the control panel diagram, is placed in the skip bar for the first column to be punched and a skip is wired to the column following the result *T*.

6. Program 2 exits provide the impulses to instruct the machine to read out the result *T* from the counter into storage unit 6R and tell the machine to punch, to reset the counter, and to signal the end of the program for the card by impulsing READ. This impulse tells the machine to feed the next card.

7. The date "02-19," which is common to all cards, is wired from the punch emitter to punch in columns 15-19. An insert is placed in the skip bar for the first column to be punched and a skip is wired to the column following the date.

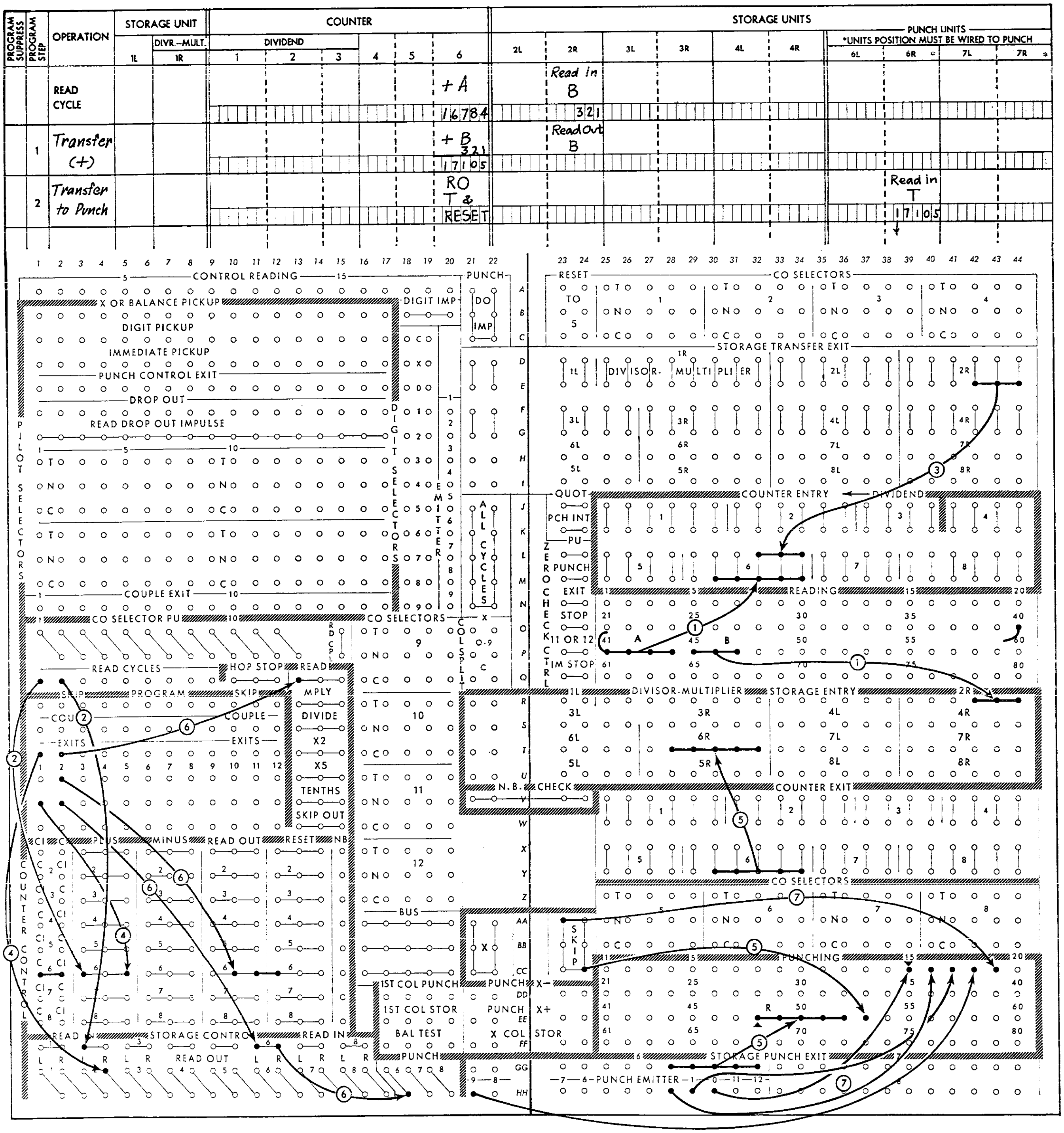


*Figure 8. Crossfooting*
*A + B = T*

## BALANCE CONVERSION

### A + B — C = ± T

Whenever a negative factor is included in a crossfooting operation, it is always possible for the result to be negative. The machine punches negative results in complement form unless balance conversion is wired. This problem demonstrates the method for converting complements to true figures and for punching X identification for either positive or negative balances by using the balance conversion unit.

*Balance Test (DD-FF, 17-24).* The high-order position of a counter exit, when wired to storage punch entry, may be used to control the balance test unit, provided there is one more position in the counter group than there are digits in the result. If the result in the counter is negative, the high-order counter position to the left of the significant values is a nine; if the result is positive, it will be a zero. The nine is used as a signal that the result is a complement and should be converted.

There are four independent balance test units, each consisting of a balance test hub for entry of the 9 or 0, a first column storage hub for entry of the first position of the result, and a first column punch hub for exit of the first column of the result to be punched. If a 9 is recognized at the balance test hub, the conversion unit is energized, and when first column storage and first column punch are properly wired, the unit converts to complement form whatever is punched from that storage unit until another 9 or 0 from the high-order position of a result reaches a balance test hub (from the same or another storage exit) or a new card feeds. The actual conversion of the complement number held in a punch storage unit is accomplished by a relay which reverses the connections between the positions of the storage unit and the punch magnets so that the result is punched in true form.

## X Punch over Units Position

Figure 9 shows the control panel wired for conversion of a complement answer with an X punch over the units position of the field to signify a negative balance. Refer to this figure in order to follow the actual operation of the balance conversion unit.

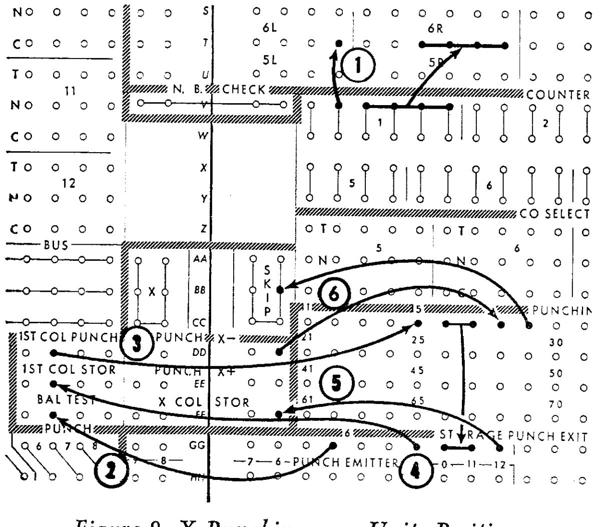


*Figure 9. X Punching over Units Position*

Counter 1 has developed a four-position negative result. Either the fifth or sixth position of the counter may be wired to any free position of storage entry.

1. The corresponding position of the storage punch exit is wired to balance test.

2. This position always contains either a zero if the result is positive or a nine if the result is negative. When the card is positioned at the first column of the field to be punched (column 5), the impulse from the column emitter is carried by control panel wiring from the PUNCHING column to 1ST COL. PUNCH.

3. This impulse picks up a relay network which directs it out to BALANCE TEST. It is carried to the test position of the storage punch exit by control panel wiring. If the storage wheel stands at 9, the impulse is directed to the pickup of the internal punch-balance relay. As soon as the balance test has been completed, the impulse from PUNCHING is switched from BALANCE TEST to 1ST COL. STOR.

4. From here it goes by way of control panel wiring to the storage position containing the highest-order digit of the quantity to be punched. Punching proceeds normally if the punch-balance relay has not been picked up. If it has, the 9's complement of each number in the storage unit is punched. Once this punch-balance relay has been picked up, it holds until another impulse from PUNCHING is entered into another of the first column punch hubs of the balance conversion unit, or until a new card feeds. Therefore, if several fields are to be punched, and one of them is tested for possible conversion, the next value to be punched must also be tested, even though the value will always be a positive number. If the test is not made, the remaining values punched on that same card will be converted if the previous value was converted.

5. Conversion alone is not enough. The result must be identified as negative either by X or D punching. For X identification there are four punch-X units located to the right of the balance test. Each unit has a hub (X column storage) for entry of the result digit from the storage punch exit and two exits: one (punch X+) which punches an X for a positive balance, and another (punch X—) which punches an X for a negative balance. The X may be punched over any column of the result or outside the result field. If the X is to be punched over the result, the punch exit for the column over which it is to be punched is wired to X column storage. Then,

either punch X— or punch X+ is wired to the proper column to punch an X over a negative or positive result, respectively.

6. If the X is to be punched outside the result field, X column storage is wired from SKIP. The punch X— or punch X+ is wired to the column to be punched or skipped.

### X Punch over High-Order Position

It is possible to punch an X over the first column of the field (Figure 10) provided the balance test takes place first. The wiring of the punching impulse directly to the balance conversion unit and then through the punch X+ or punch X— unit to the storage exit insures that the punch balance relay is controlled as a result of the balance test before the circuit is made through the punch X unit.

This means that the first column to be punched must be wired directly from the first column punch.

The punch X unit may be used whenever it is necessary to punch both a digit and an X in the same column even if the X indicates some condition other than a positive or negative result. For example, to punch an X in column 61 as well as a digit from storage 6, the X+ hub is wired from column 61 and X column storage to the storage punch exit.

The timing of the punch X+ or punch X— prohibits punching from an X prior to the punching of the first field. An X from the emitter, properly selected, could be used.

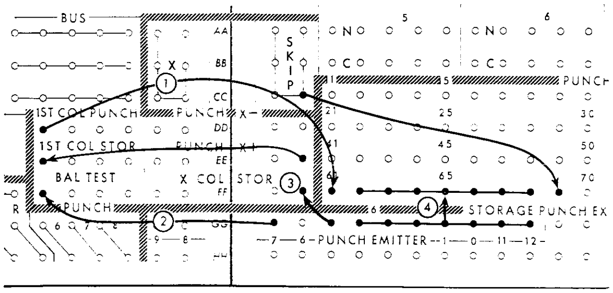


*Figure 10. X Punching over High-Order Position*

### Planning Chart and Control Wiring

Planning and wiring are shown in Figure 11. An explanation of the planning chart is, in effect, an explanation of the program wiring on the left side of the control panel and, because of their close association, they may be described together.

*Read Cycle.* $A$ is added in coupled counters 2 and 3. $B$ and $C$ are entered into storage units 2 and 3. It would also be possible to plan the problem so that all three factors were entered into three sets of storage units. In this case, it would take additional time to complete the problem, since one cycle would be used in moving the first factor from storage to a counter. Although no time would be lost in entering all three factors into three sets of counters, it would use counter capacity unnecessarily and require that they be reset after use.

*Program 1.* $B$ is read out of 2L and 2R and added into counter 2-3.

*Program 2.* $C$ is read out of 3L and 3R and subtracted in counter 2-3.

*Program 3.* The result $\pm T$ is read out of counter 2-3 and read into storage unit 6 where it is held for punching. The counters reset at the same time. The total is ready to punch as soon as it enters storage unit 6; therefore, program 3 also impulses storage unit 6 to punch. Since program step 3 is the last step required for this calculation, a step 3 exit is wired to READ.

### Right-Hand Panel Wiring

1. $A$ is wired to counter 2-3, $B$ to storage unit 2, and $C$ to storage unit 3.

2. $B$ and $C$ are wired to read out of storage exits 2 and 3 into counter 2-3. Split wires are avoided by wiring $C$ through the common hubs of 2L and 2R.

3. The result is wired from counter exits 2-3 to 6L and 6R for punching.

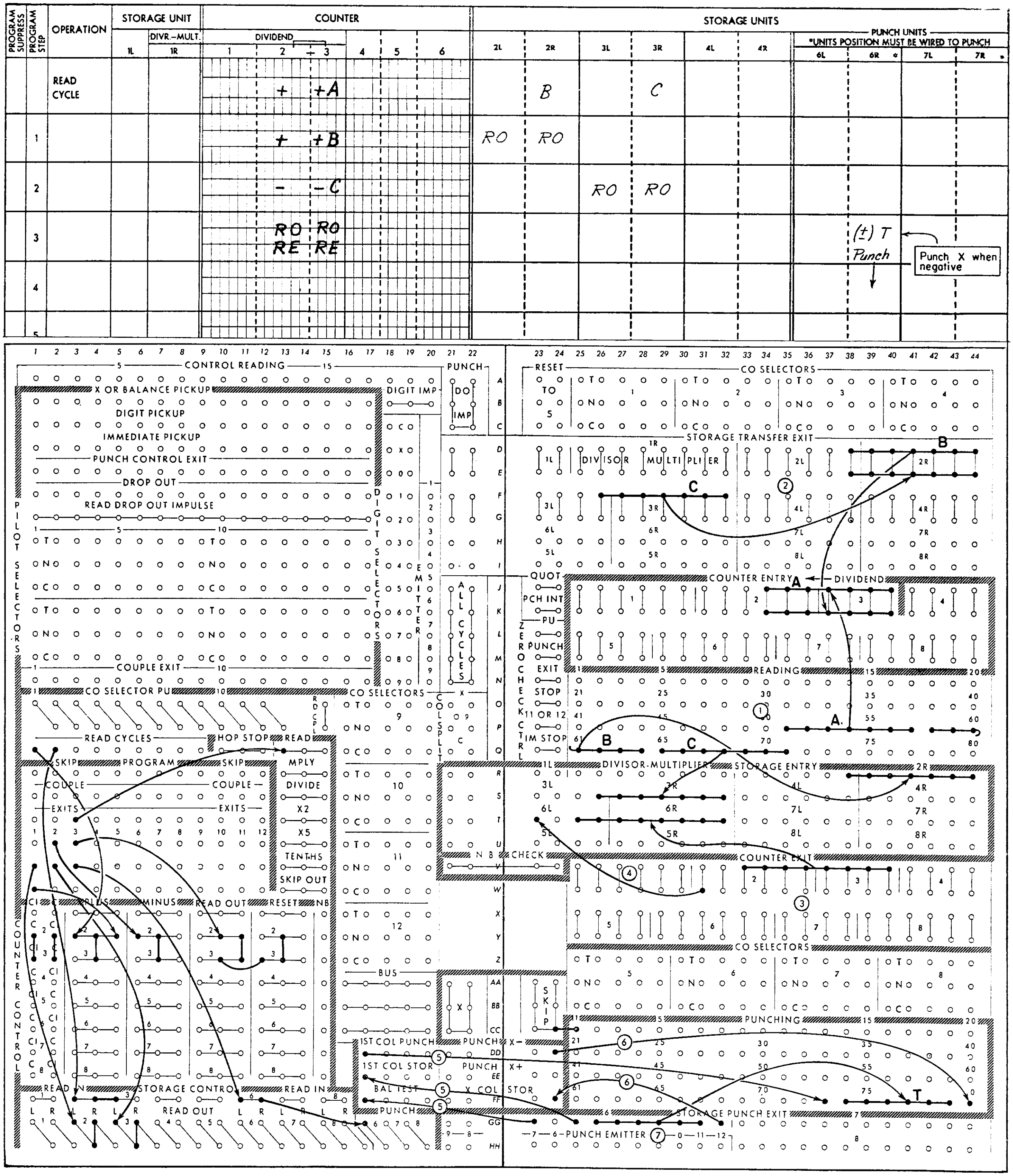
4. The left-hand position of counter 2 is wired to 6L so that it will be available from the storage exit to be used as a signal for balance conversion. It may be wired to any of the remaining hubs in 6L.

5. The storage exit corresponding to the left-hand position of counter 2 (complement 9) may be wired to any one of the four balance test units as a pickup for that unit. The first column of the result is wired to 1ST COL. STOR. and from 1ST COL. PUNCH to punch in column 73. A 9 in counter 2 will convert to positive form the complement result in storage unit 6.

6. An X is to be punched over the units position of the result for all negative amounts. The units position is wired from storage punch exit 6 to X column storage, and punch X— is wired to column 80. Punch X— emits an X for all minus balances,

and combines with the digits wired into X column storage to punch both the units digit of the field as well as the X punch.

7. The remaining part of the result is wired to punch. Whenever column 80 is wired to punch, the skip must be wired to column 1.



| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT DIVR.-MULT. 1R | COUNTER DIVIDEND 1 | COUNTER DIVIDEND 2 | COUNTER DIVIDEND 3 | COUNTER 4 | COUNTER 5 | COUNTER 6 | STORAGE UNITS 2L | 2R | 3L | 3R | 4L | 4R | PUNCH UNITS 6L | 6R | 7L | 7R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | READ CYCLE | | | | + | +A | | | | | B | | C | | | | | | |
| | 1 | | | | | + | +B | | | | RO | RO | | | | | | | | |
| | 2 | | | | | - | -C | | | | | | RO | RO | | | | | | |
| | 3 | | | | | RO<br>RE | RO<br>RE | | | | | | | | | | | (±) T<br>Punch | Punch X when negative | |
| | 4 | | | | | | | | | | | | | | | | | | | |
| | 5 | | | | | | | | | | | | | | | | | | | |

*UNITS POSITION MUST BE WIRED TO PUNCH

*Figure 11. Balance Conversion*
$A + B - C = \pm T$

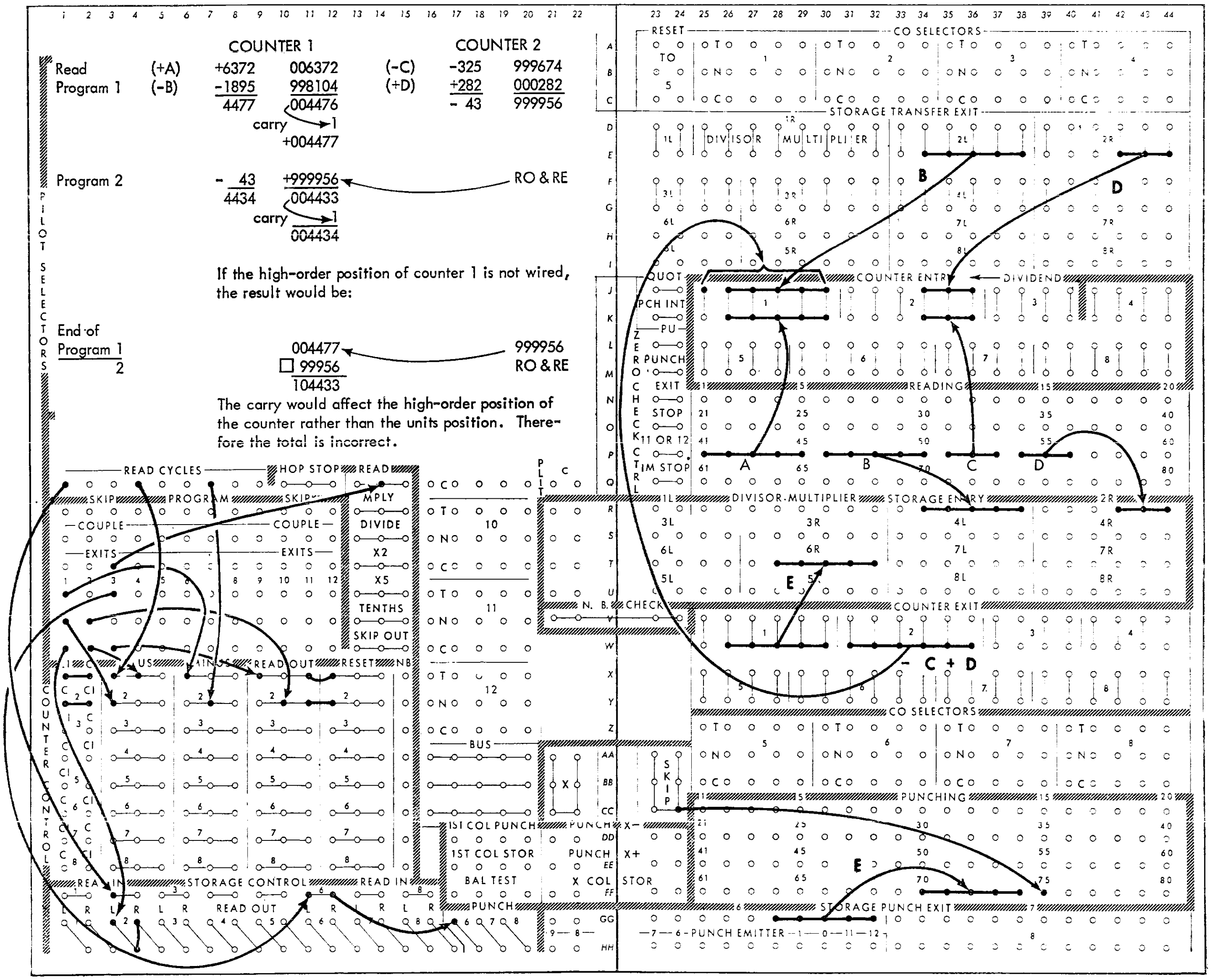
## COUNTER TRANSFER OF NEGATIVE NUMBERS

$$A-B-C+D=E$$

In crossfooting more than three factors, fewer program steps may be used by wiring simultaneous crossfooting in two or more counters, and transferring the results from one counter into another. In the following example, one more program step would be needed to crossfoot all four factors in a single counter.

When there is a possibility of a negative total in the counter to be transferred, the receiving counter must be wired for the full capacity, as shown in the problem and solution in Figure 12. When a 4-position

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT | | COUNTER | | | | | | STORAGE UNITS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | DIVR.-MULT. | DIVIDEND | | | | | | | | | | | | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH | | | |
| | | | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R * | 7L | 7R * |
| | | READ CYCLE | | | +A<br>006372 | -C<br>999674 | | | | | B<br>1895 | D<br>282 | | | | | | | | |
| | 1 | | | | -B<br>004477 | +D<br>999956 | | | | | RO | RO | | | | | | | | |
| | 2 | | | | +(-C+D)<br>004434 | RO<br>RE | | | | | | | | | | | | | | |
| | 3 | | | | RO<br>RE | | | | | | | | | | | | | PUNCH<br>E<br>4434 | | |



COUNTER 1 — COUNTER 2

Read (+A) +6372 006372 (-C) -325 999674

Program 1 (-B) -1895 998104 (+D) +282 000282

4477 004476 - 43 999956

carry 1

+004477

Program 2 - 43 +999956 ← RO & RE

4434 004433

carry 1

004434

If the high-order position of counter 1 is not wired, the result would be:

End of Program 1 / 2

004477 ← 999956 RO & RE

□ 99956

104433

The carry would affect the high-order position of the counter rather than the units position. Therefore the total is incorrect.

*Figure 12. Counter Transfer of Negative Numbers*

counter containing a negative balance is transferred to a 6-position counter, the high-order position of the smaller counter must be split-wired to all remaining positions of the larger counter. If the sign of the transferred total is to be changed, the factors must be subtracted in the receiving counter.

## MULTIPLICATION

$A \times B = P$

THE 602 offers considerable flexibility for multiplication. The punch, equipped with wheel-type counters, refers to mathematical tables built into its electrical circuits. (See "Method of Multiplication.") Factors are read from the card: the multiplicand into storage or a counter and the multiplier into the multiplier storage unit 1R. In this unit, the multiplier factor is retained so that each digit in the factor can be referred to the necessary mathematical table. The component resulting from multiplication by this digit is entered into a group of wheel counters to be accumulated. Then the next digit in the multiplier is applied to the multiplicand by means of built-in circuit tables. Each of the components resulting as a product of those multiplications is added to the accumulated total in the counters.

This process continues until every possible position in the multiplier has been processed through the multiplication table circuits, and the counter stands with the accumulated product of the multiplication. The entire multiplication is completed by a single program step, although the number of cycles in this one program step depends on the number of digits in the multiplier. Therefore, the factor with the smaller number of digits should be used as the multiplier, because the speed of the machine when multiplying is about 1.4 cycles per multiplier digit.

It is possible to develop a 30-digit product because there are 30 counter positions. This may be done by multiplying an 8-digit multiplier by a 22-digit multiplicand, or by expanding the multiplier beyond the 8-digit capacity of the multiplier unit by treating the multiplier in parts.

Figure 13 illustrates the use of a 5-position multiplier with one decimal position and a 5-digit multiplicand with three decimal positions.

### Control Panel Hubs

*Multiply (FR, 13-15).* Three common entry hubs receive impulses from program exits to cause the machine to multiply. On the same program step on which the multiply hub is impulsed, two other commands must be given. One of these is to identify the multiplicand by a read-out instruction to the storage unit or the counter; the other is to instruct the counter which develops the product to add if a plus product is to be developed, or to subtract, if a negative product is to be developed. Failure to read out the multiplicand or to impulse the product counter develops a zero answer. If the multiplicand is stored in a counter rather than in a storage unit, the multiplicand counter can only be controlled to read out on the multiply step. If it is controlled to read out and reset, the multiplicand would reset on the first of several multiplication cycles and, thus, develop an erroneous product.

If the product is too large for a single counter, any number of adjoining counters may be coupled. They must be adjoining because of the shifting of components as the multiplier moves from one digit to another.

No other function should be performed on the same program step with multiplication. This includes crossfooting, punching, and negative balance testing. For example, if crossfooting were attempted on the same program cycle with multiplication, the crossfooting factor would be treated as a multiplicand because of the read-out instruction, and an unwanted product would be developed in the counter which was adding or subtracting.

Multiply hubs should not be wired from two successive program steps; one cycle must intervene to allow time for the repetitive action of multiply circuits to stop.

*Reset to 5 (Half-Adjustment) (A-C, 23-24).* If some of the decimals in a product are to be dropped, the last position retained may be adjusted to the nearest whole number. This is done by adding five to the first dropped position. The machine accomplishes this by resetting the counter position representing the first dropped decimal to five instead of zero. Hubs labeled "reset to 5" are exits which may be wired for this purpose to any of the 30 counter positions. Because all machine commands are also executed on the run-in reset, the counter position which has been wired from reset-to-5 will stand with a five before the first card calculates a result. This means that a sum developed on program 5 may be read out and reset on program 6 for punching, but program 6 of the run-in reset has entered the half-

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT DIVR.-MULT. 1R | COUNTER (DIVIDEND 1, 2, 3; 4, 5, 6) | STORAGE UNITS 2L | 2R | 3L | 3R | 4L | 4R | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH 6L | 6R * | 7L | 7R * |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | READ CYCLE<br>Card 1 |  | A<br>1746.3 | RO<br>RE<br>5 |  | B<br>28.049 |  |  |  |  |  |  |  |  |
|  | 1 | Multiply |  |  | 489819687<br>+ +P<br>489819737 |  | RO |  |  |  |  |  |  |  |  |
|  |  | Read<br>Card 2 |  | A<br>2168.1 | RO<br>RE<br>5 |  | B<br>78.945 |  |  |  |  |  | P<br>PUNCH<br>↓ |  |  |
|  | 1 | Multiply |  |  | 1701606545<br>+ + P<br>1701606595 |  | RO |  |  |  |  |  | 48981.97 |  |  |
|  |  | Read<br>Card 3 |  | A | RO<br>RE<br>5 |  | B |  |  |  |  |  | P<br>PUNCH<br>↓ |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | 170160.65 |  |  |

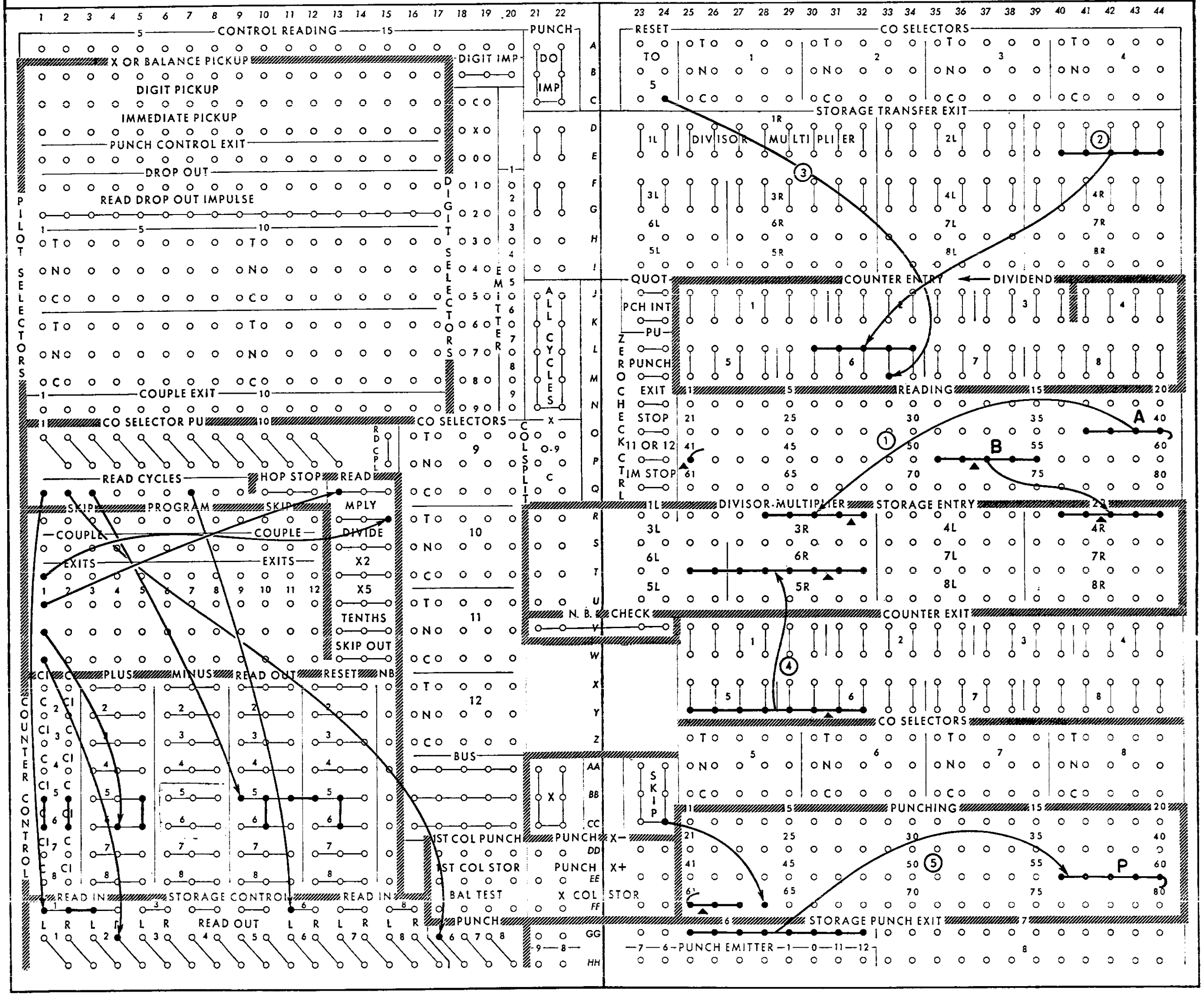


*Figure 13. Multiplication*
$A \times B = P$

adjusted figure for the first card before the sum has been developed. Other digits entered into that position on the actual calculating cycles will add to the five and adjust the position to the left.

The reset-to-5 impulse has the same timing as an X impulse. When a factor is entered into two counters by wiring from one counter entry to another, only one of the products may need to be adjusted. The reset-to-5 impulse must be stopped from getting to the second counter by entering that position of the second counter through a column split device, which filters out the reset-to-5, or X, impulse. See "Operating Suggestions."

*Column Splits (O-Q, 21-22).* There are two independent column split units which have the function of separating 0-9 impulses from X impulses. Each unit has a C (common) hub, a 0-9 hub, and an X hub. If a card column from the reading brushes is entered in C, only digits from 0 through 9 are available at the 0-9 hub, and only X impulses at the X hub.

### Planning Chart and Control Wiring

*Read Cycle (Card 1).* Factor *A* is entered into the multiplier unit 1R and *B* into storage unit 2. Factor *B* could have been entered into any other storage unit or counter not needed for developing the product. Ro, RE and 5 (half-adjust) are explained below.

*Program 1.* The three multiply instructions are wired on this program step. Factor *B*, the multiplicand, is read out of storage 2R, coupled counters 5 and 6 are impulsed to add to develop a positive product, and the machine is told to multiply by impulsing the multiply hub.

*Read Cycle (Cards 2 and 3).* Advantage may be taken of the fact that counters 5 and 6 do not add or subtract on the read cycle. For example, these counters can be reset on the read cycle, thus saving a program. If counters 5 and 6 were impulsed to add or subtract at read time, it would be necessary to take another program to clear these counters, because a counter cannot read in and out at the same time. The reset-to-5 occurs at the time the counter resets, and, since in this case the counter is reset on a read cycle, the reset to 5 also occurs at that time. When the product is developed in the counter group on program 1, it is adjusted to the nearest whole number.

Storage unit 6 is impulsed to read in and punch on the read cycle. The punching of the first card starts on program 1 of card 2, and because multiplication may take several cycles, and four columns may be punched on each cycle, there is no loss of time for punching in this problem.

Punching of the result for card 2 starts as program 1 of card 3 is being executed, and so on.

### Right-Hand Panel Wiring

1. *A* is read into the multiplier unit and *B* into storage unit 2R.
2. The multiplicand *B* is read out of storage 2R into counter 5-6 where the product is developed.
3. Two of the four decimal positions in the product are to be retained. Reset to 5 is therefore wired to the first position dropped, which in this case is the tens position.
4. The product, with two positions dropped, is wired to storage unit 6 for punching.
5. The product is read from storage punch exit 6 to punch. A skip is wired to the column following the last column punched.

## MULTIPLICATION AND CROSSFOOTING WITH DECIMAL ALIGNMENT

$$(A \times B) + C + D = R$$

Multiplication and crossfooting may be combined in one operation. This problem demonstrates the calculation of $A \times B$ added to two crossfooting factors, *C* and *D*, with decimal alignment (Figure 14). Although only two crossfooting factors are demonstrated, additional factors can be crossfooted using the same wiring principles.

### Planning Chart and Control Wiring

*Read.* *A*, *B*, and *C* are entered into storage units 1, 2, and 3, respectively. *D* is added in counters 5 and 6, which are coupled. Since the product of $A \times B$ is not to be punched, time is saved by adding *D* into the same counters in which the product is developed.

*Program 1.* The multiplicand is read out of 2L and 2R into counter 5-6, where a positive product is developed. The multiply hub is impulsed. At the conclusion of the multiplication, the counters contain $A \times B + D$.

*Program 2.* Factor *C* is read out of 3L into counter 5-6 and added to $A \times B + D$.

*Program 3.* Counter 5-6 cannot read out and reset on the read cycle because it is impulsed to add on

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT DIVR.-MULT. 1R | COUNTER DIVIDEND 1 | 2 | 3 | 4 | 5 | 6 | STORAGE UNITS 2L | 2R | 3L | 3R | 4L | 4R | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH 6L | 6R * | 7L | 7R * |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | READ CYCLE |  | A |  |  |  |  | + | +D |  | B | C |  |  |  |  |  |  |  |
|  | 1 | Mult. |  |  |  |  |  |  | + | +P |  | RO |  |  |  |  |  |  |  |  |
|  | 2 |  |  |  |  |  |  |  | + | +C |  |  | RO |  |  |  |  |  |  |  |
|  | 3 |  |  |  |  |  |  |  | RO<br>RE | RO<br>RE<br>S(Corr) |  |  |  |  |  |  |  | R<br>Punch |  |  |
|  | 4 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

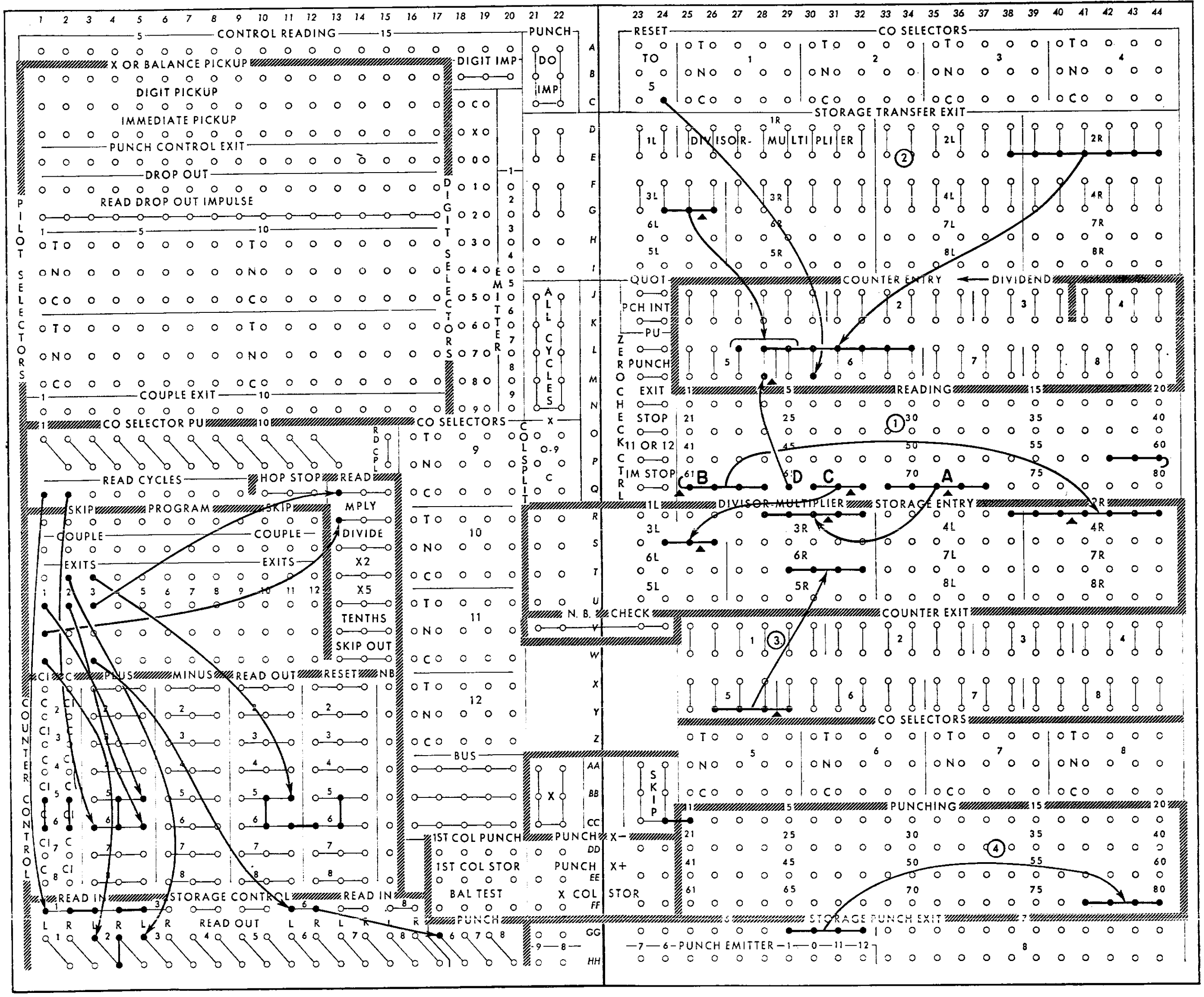


*Figure 14. Multiplication and Crossfooting*
*$(A \times B) + C + D = R$*

that cycle. Counters cannot read in and out at the same time. Therefore, program 3 is used to read out and reset the counter and to transfer the result to 6R for punching. The proper position in counter 6 is reset to 5 to adjust the product. Program 3 is the last program used, and is wired to READ.

### Right-Hand Panel Wiring

1. *A, B, C,* and *D* are wired to their respective storage units or counters. Since *D* is wired directly to the counter that will contain the result, it must be properly positioned. *D* has no decimal places. The product will have six decimals. Therefore, *D*, with no decimals, is placed to the left of the decimal point or seven positions from the right of the counter.

2. The multiplicand in 2L and 2R is read into counter 5-6. Decimals do not have to be considered when wiring a multiplicand from any unit to the counter in which the product is developed. The multiplicand need not be wired to the units position of the counter, but in most cases it is desirable to do so. Offsetting the multiplicand or the multiplier in storage will increase the product by the corresponding number of positions.

Factor *C*, with one decimal, is wired from storage transfer exit 3L to counter 5-6, with its decimal properly positioned.

Of the six decimals in the product, five are to be dropped. Reset to 5 will therefore be wired to the highest order position dropped, or position 5.

3. With five places dropped, the result is read out of counter exits 5-6 into storage entry 6R.

4. The result is punched from storage punch exit 6R. SKIP is wired to column 1.

## DIVISION

## $A \div B = Q$

DIVISION in the 602 Model 1 requires entering the dividend into coupled counters 1, 2, and 3 and the divisor into storage unit 1R. The 602 uses built-in divisor look-up tables to accomplish division. See "Method of Division."

Each digit in the divisor is considered separately, and the dividend is reduced by repeated subtractions until the remainder is less than the divisor. A count of the number of subtractions necessary to make this reduction is made in counters 4, 5, or 6, and becomes the quotient.

Since storage units 1R and counters 1, 2, and 3 must be used in a divide operation, capacity of the 602-A is limited to a maximum 15-position dividend, a maximum 8-position divisor, and the development of an 8-position quotient. Expansion of the quotient and the divisor is possible by special wiring.

If the division involves decimals, care must be taken to position the dividend properly in the dividend counters. This may be best compared with the long division procedure. For example, to divide 45.76 by 3.51 and develop an adjusted quotient of 13.04 requires that three zeros be added to the dividend. Because the last position of the quotient was five or greater, the tens position was increased one and the third decimal position dropped.

```
          13.037 adjusted quotient = 13.04
3.51√45.76 000
     35 1
     10 66
     10 53
        13 00
        10 53
         2 470
         2 457
            13
```

This same procedure must be followed if these numbers are to be divided on the 602. The factor 45.76 would become 4576000 in counter 1-2-3 by shifting the units position of the factor over into the fourth position of the counter.

A rapid determination of the position of the dividend in the counter when dividing with decimals may be made by applying the following rule. The number of decimals in the divisor plus the number of decimals desired in the quotient determines the number of decimal positions in the dividend counter. If the quotient is to be half-adjusted, another decimal position must be added. In this problem (Figure 15) there are two decimals in the divisor and two in the quotient. Since the quotient is to be rounded, the number of places in the dividend is five. Notice that the dividend 45.76 was entered as 4576000 before the long division was started.

Once the divisor and dividend are properly positioned, four simultaneous instructions from a single program step accomplish this division. These are: (1) impulse the divide hub, (2) read out the divisor, (3) impulse the dividend counter to subtract, and (4) add into the counter designated for the quotient.

*Divide (S, 13-15).* Three common entry hubs receive impulses from program exits to cause the machine to divide. Impulsing this hub begins the comparison of the highest-order divisor digit and the highest-order dividend digit to determine a quotient digit. On the same program step on which the divide hub is impulsed, three other commands must be given to complete the division. One of these is a read-out instruction to the divisor unit 1R; another is a minus command to the dividend counter 1-2-3. These two instructions allow the product of each quotient digit times the divisor to reduce the dividend progressively to zero. A fourth instruction on this program step is to add into the counter which develops the quotient.

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT |  | COUNTER |  |  |  |  |  | STORAGE UNITS |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  | DIVR.–MULT. | DIVIDEND |  |  |  |  |  |  |  |  |  |  |  | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH |  |  |  |
|  |  |  | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R * | 7L | 7R * |
|  |  | READ CYCLE |  | B<br>3.51 | + | +<br>45.7 | +A<br>6000 |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 1 | Divide |  | RO | – | – | – |  |  | +Q<br>13.042 |  |  |  |  |  |  |  |  |  |  |
|  | 2 |  |  |  | RE | RE | RE<br>13 |  |  | RO<br>RE<br>5 |  |  |  |  |  |  |  |  |  | Q<br>PUNCH<br>13.04 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

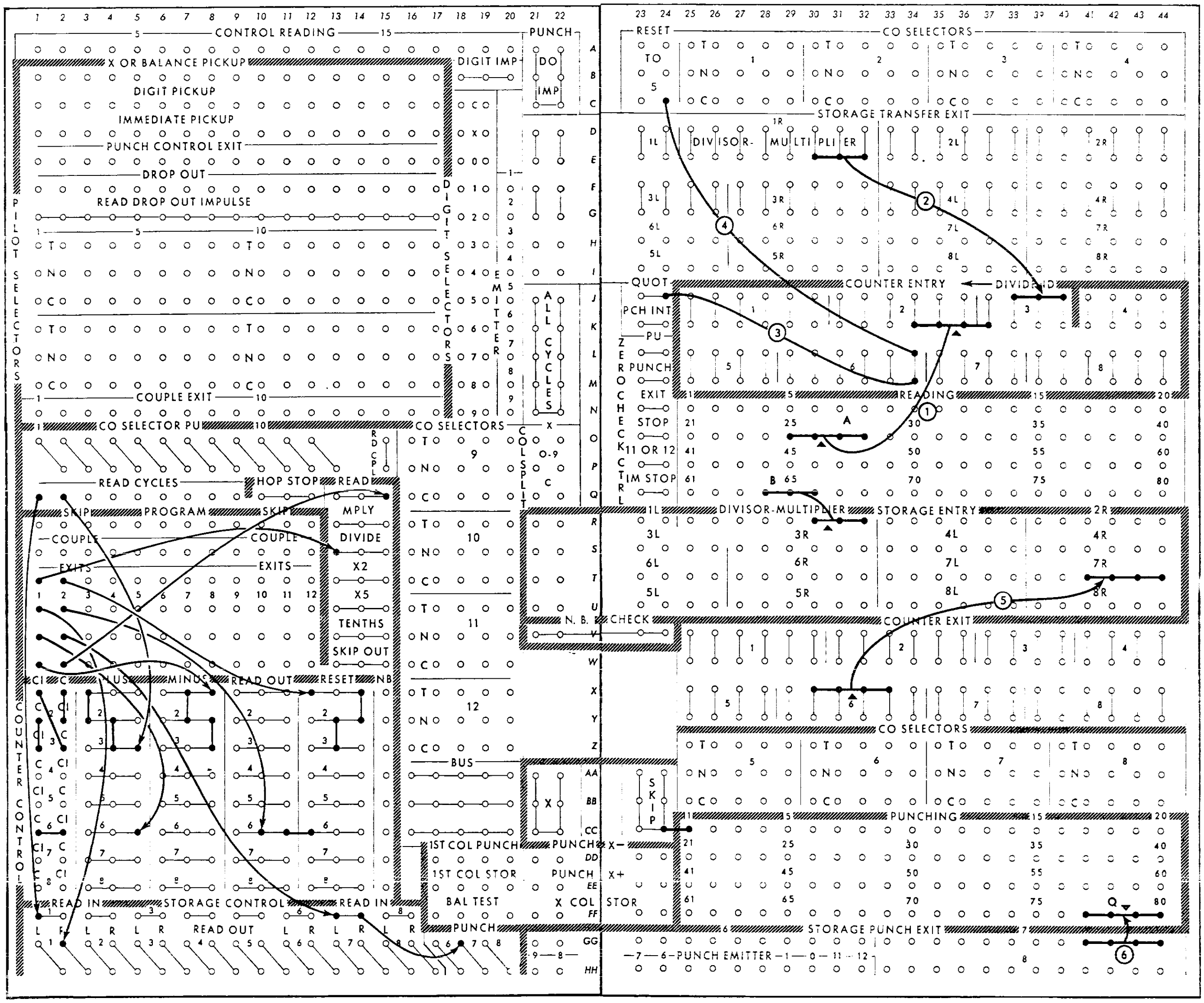


*Figure 15. Division*
$A \div B = Q$

This allows each quotient digit resulting from the internal division comparison to add as it develops from left to right. Failure to wire any one of these steps develops a quotient of zero.

Divide hubs should not be wired from two successive steps because the repetitive use of divide circuits requires an intervening program step.

*Quotient (J, 23-24)*. The quotient is developed one position at a time as the internal test compares the size of the dividend with the size of the divisor. This single quotient position is available from two common quotient hubs and is normally wired to the units position of a counter. Even though only a single position is wired, a multiple-position quotient would be developed because of the internal shift that occurs as the divide cycles are taken.

### Planning Chart and Control Wiring

*Read.* The dividend *A* is added into the dividend counter 1-2-3. The divisor is entered in 1R.

*Program 1.* Division is accomplished by impulsing the divide switch, by reading out the divisor from 1R, by subtracting the dividend, and by adding the quotient into counter 6. When the dividend has been reduced to zero or to a point where the remainder is plus and less than the divisor, the dividing operation is completed.

*Program 2.* The remainder, if any, in the dividend must be cleared (counter 1-2-3 wired to reset). The quotient in counter 6 is read out into storage unit 7 and punched. Counter 6 is also reset on this program, with reset-to-5 wired to the units position of counter 6 for adjusting the quotient on the following card. The last program used is wired to READ.

### Right-Hand Panel Wiring

1. The dividend is positioned in counter 1-2-3 for five decimal places. The divisor is entered in 1R.

2. The divisor is read out of 1R to subtract from the dividend.

3. The quotient exit is wired to the units position of counter 6, where the quotient digit adds as it develops from left to right.

4. The units position of counter 6 is reset to 5 to half-adjust the quotient.

5. The quotient is wired out of counter 6 to storage 7 for punching.

6. The quotient is wired to punch and skip is wired to column 1.

## SIMULTANEOUS DIVISION AND MULTIPLICATION

## $(A \div B) \times C = R$

IT IS possible to obtain the result of $(A \div B) \times C = R$ on the same program step (Figure 16). This operation is sometimes referred to as quotient multiplication. As explained under "Method of Division," dividing is accomplished by a succession of multiplication cycles during which each quotient digit as it is developed is multiplied by the whole divisor, and the result subtracted from the dividend. At the same time that this operation is being performed, a third factor may be multiplied by the quotient and the result obtained without loss of time.

To punch the quotient as well as the product of $Q \times C$, either counter 4 or counter 6 may be used for developing the quotient, and the other two may be coupled together for developing the quotient product.

Whenever the quotient is punched, it may be half-adjusted normally. However, factor *C* is multiplied by the true quotient of $A \div B$, and not by the adjusted quotient. For example, a quotient of .348 may be adjusted and punched as .35, but .348 *(not .35)* would be multiplied by *C* to develop *R*.

### Planning Chart and Control Wiring

*Read.* The divisor *B* is entered into the divisor unit 1R, factor *A* is entered into the dividend counter, and factor *C* is entered into storage unit 2R.

*Program 1.* *B* is read out of 1R, and counter 1-2-3 is wired to subtract. Since the quotient is not to be punched, QUOT is not wired to a counter. The result of $(A \div B) \times C$ is developed by impulsing DIVIDE, reading *C* out of 2R, and adding in counter 5-6.

*Program 2.* Counter 1-2-3 is reset to clear the dividend remainder. *R* is read out of counter 5-6 to storage unit 7 for punching. The counter is reset to 5.

### Right-Hand Panel Wiring

1. Since there are no decimals in the divisor and three decimals are desired in the unadjusted quotient, three decimal positions are allowed in the dividend counter. Therefore, the decimal point is placed between the third and fourth positions of the dividend counter. The dividend *A* with two decimals is wired to the dividend counter to line up with the

decimal point already established. *B* is wired to the multiplier and *C* to storage unit 2R.

2. The divisor exit is wired to the dividend counter without consideration for the decimal point.

3. Factor *C* is wired out of 2R to counter 5-6 where the result of $(A \div B) \times C$ is developed.

4. Since the quotient has three decimals and *C* has none, there will be three decimal positions in the product. The units position of counter 6 is reset to 5, thus adjusting the result.

5. The result is wired out of counter exits 5-6 to storage unit 7. The units position is dropped.

6. The result is wired to punch in columns 74-80, and SKIP is wired to column 1.

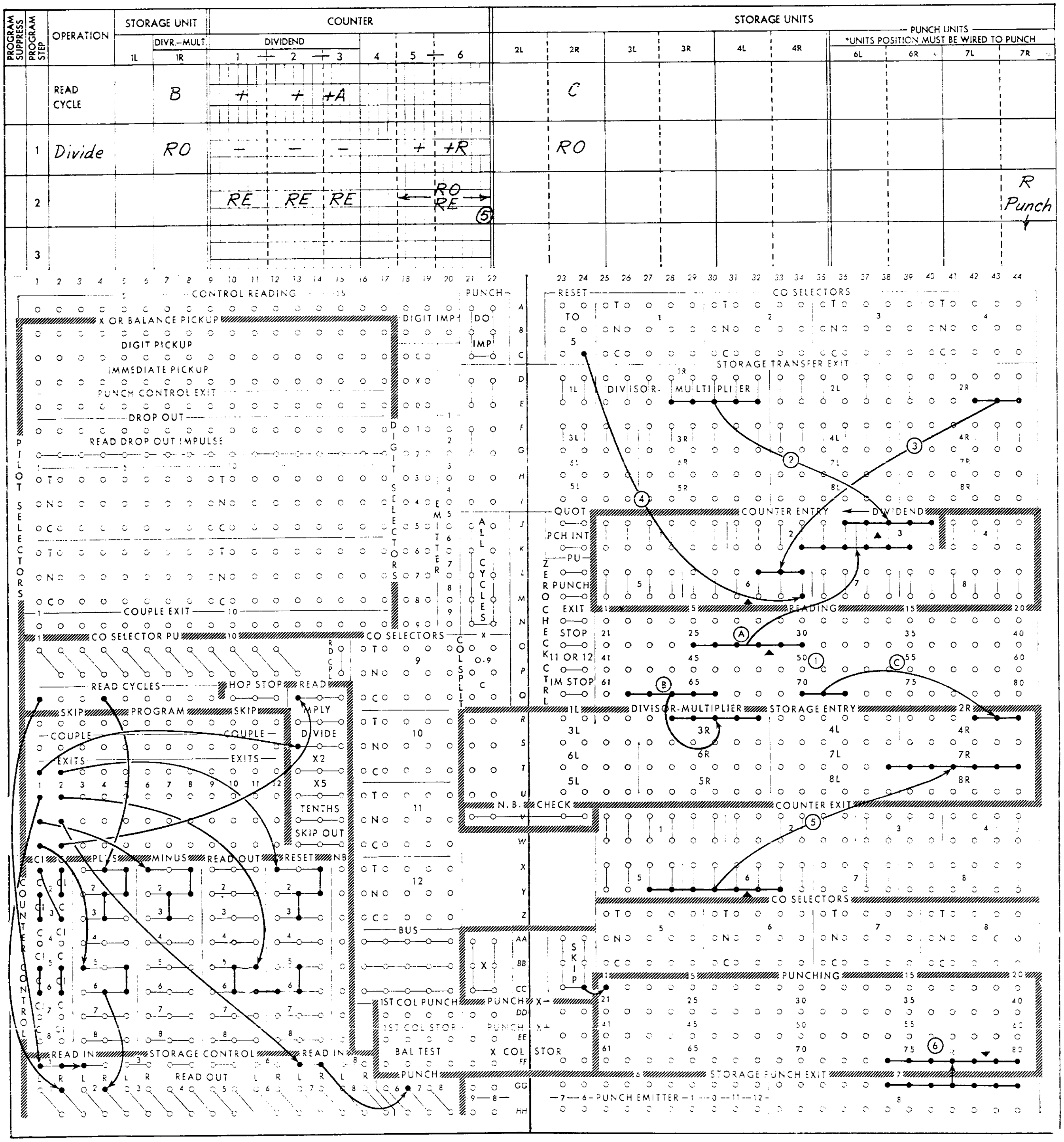


*Figure 16. Simultaneous Division and Multiplication*
$(A \div B) \times C = R$

## SELECTION

The concept of selection adds a great deal of flexibility to 602 operation. By the correct and resourceful use of selectors, the machine can be made to perform many complex operations not otherwise possible. Understanding the principles of selection makes it possible to solve problems with the least amount of time and effort.

Selectors have two main uses: they allow isolation of certain impulses at chosen times and they allow for branching to alternative conditions in a problem. Along with these two main functions, selectors also allow the machine to remember distinguishing punches or problem conditions for a given length of time.

To get an understanding of 602 selectors, three main questions must be answered:

1. How do 602 selectors operate?
2. When can selectors be used?
3. How are they wired to perform their functions?

A selector can be compared to a switch with two separate circuits. Each circuit, called a position of the selector, contains a normal condition (N) or an alternative transferred condition (T). Several different signals from the cards or internal machine circuits can be used to transfer a selector. When a selector has been switched from its normal to its transferred condition, it is said to be "picked up." An impulse entered at the common (C) of a selector in this picked-up condition is available from the transferred (T) hub; with the selector in the normal or dropped-out condition, the impulse is available from the normal (N) hub. The time at which a selector picks up and the length of time it remains picked up depend on the type of selector and the type of pickup impulse used.

The selectors in the 602 can be used during any of the major machine functions of reading, calculating, and punching. The first two in the following list are used to perform alternative selection, where either the normal or transferred hubs are used as entries or exits from the selector. The last two are used for isolation; an impulse entered at the common gets through only when the selector is in a predetermined condition, either normal or transferred (Figure 17).

1. Direct an impulse to one or the other of two alternate entries.
2. Select an impulse from one or the other of two alternate sources.
3. ~~Allow an impulse to get through only at certain~~ times.
4. Prevent an impulse from getting through at certain times.

There are two types of selectors on the 602: pilot selectors and co-selectors. Each can be used independently or in conjunction with one another.

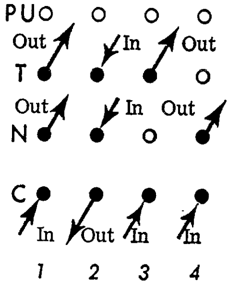


*Figure 17*

### Control Panel Hubs

*Pilot Selectors (B-N, 1-7).* Seven two-position latch type pilot selectors are on the standard machine. The two positions in each selector are vertically arranged. Each position has a C (common), an N (normal), and a T (transferred) hub operated by two relays, the control relay and the transfer relay. Generally, once the transfer relay is impulsed, an additional hold circuit keeps the selector transferred until the end of that cycle.

Latch-type pilot selectors can be picked up and remain transferred for the duration of the particular machine cycle in which they were picked up, or they can be picked up and remain transferred until a separate signal instructs the selector to return to its normal position. Therefore, latch-type pilot selectors can be used to control selection on the same cycle only, or to control selection on the same and/or later operations.

Each selector has three pickup hubs: X, D, and I.

*I (Immediate) Pickup.* This is a direct entry to the transfer relay to pick up the selector (provided the selector has not previously been impulsed at the x or d pu). It accepts any impulse and causes the selector to transfer immediately, as the name implies, but does not latch it; the selector drops out (returns to normal) automatically at the end of the cycle during which it was impulsed.

*D (Digit) Pickup.* This hub is an entry that accepts any impulse to pick up the selector. Once the selector is picked up, the selector is latched until impulsed to drop out.

Digit-type impulses (9-11) latch the selector in a transferred position one cycle later, and the selector remains transferred until it is dropped out by another impulse.

Control impulses (program exits, program couple, all cycles, or read cycles) latch the selector on the same cycle and the selector remains transferred until it is dropped out by another impulse.

*X or Balance Pickup.* This hub is an entry for X-timed or negative balance impulses. The action is the same as if it were impulsed at the D PU, in that the selector transfers on the following cycle and remains transferred until dropped out.

*Drop-out (F, 1-7).* This is an entry to unlatch the selector and may be impulsed from any of the following sources:

1. *Read Drop-out Impulses (G, 1-7)* emit on all read cycles and drop out the selector at the end of the next read cycle after the selector was impulsed to pick up.
2. *Punch DO Impulses (A-C, 21-22)* emit at the conclusion of punching and drop out the selector at the end of the cycle following the completion of punching.
3. *DI (B, 18-20)*, or digit impulses 9-1 from a counter or storage unit, drop out the selector on the end of the cycle during which drop-out is impulsed. Normally, these impulses would be wired through a selector before impulsing drop-out. The digit impulse must come before any possible impulse which is to be used to pick up the control selector again. It may come in some program before the read cycle on which the dropped-out selector is to be picked up again. Read cycles, all cycles, program exit, and couple impulses are not suitable for selector drop-out.

Although drop-out is not necessary when selectors are picked up immediately, it is advisable to drop out all selectors used because they may be in a latched position at the start of the operation.

*Couple Exit (N, 1-7).* This hub emits an impulse when the selector control relay is latched by impulsing the X or D PU. This impulse may be used to pick up other selectors which will then transfer and drop out at the same time as the "master" pilot selector. The couple exit does not emit when the I PU is impulsed because this pickup does not latch the selector control relay.

*Punch Control Exit (E, 1-7).* If a selector is transferred, this hub emits an impulse just before the card is moved into punching position. It is normally used to control another selector during punching.

*Co-Selectors (A-C, 25-44, Z-BB, 25-44).* The standard machine has eight 5-position selectors. Their pickup hubs *(O-P, 1-8)* are diagonally arranged for convenience in jackplugging and are like the I pickup of pilot selectors in that they accept any impulse to transfer the selector immediately. Co-selectors may be used independently or in conjunction with pilot selectors to expand the number of operating positions. If a co-selector is used in conjunction with a pilot selector, it should be picked up by the couple exit of the pilot selector. The co-selector then transfers and drops out exactly like the pilot selector. If a co-selector is picked up from any impulse other than a couple exit, it is transferred immediately and held for the remainder of the cycle.

Every selection problem involves either the control of a machine function (multiplication, read-in, reset, and so on) or the flow of information between the card and the various machine units and/or between one machine unit and another. The control may be in the form of a distinguishing punch in the card if more than one card is involved, or from conditions in the calculation such as a negative condition in a counter.

The following planning charts and control panel diagrams illustrate only the basic principles of selectors and selector networks on the 602 to control machine functions. The elements such as program steps, selectors, and brush positions used in these examples have been arbitrarily chosen and are not intended to limit application of these principles to other units.

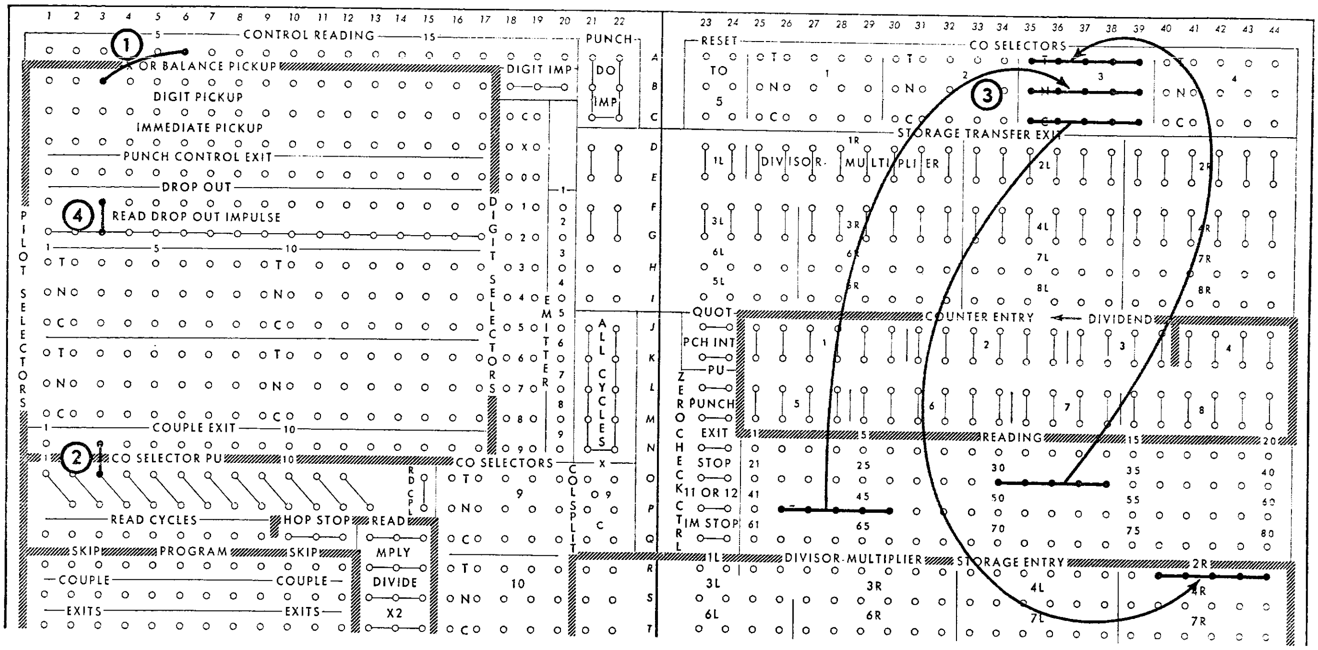


*Figure 18. Field Selection on Read Cycle*

## Field Selection on the Read Cycle

This example (Figure 18) recognizes a distinguishing X punch to control the reading of either of two fields (columns 30-34 from the X21 card, or columns 42-46 of NX cards) into storage 2R.

This type of selection requires that the distinguishing punch be read ahead of time, at the control brushes, in order to transfer the selector for the entire reading of the card at the reading brushes.

*Control Reading (A, 1-20).* There are 20 control reading brushes on the standard machine. They are some times referred to as the upper brushes, and are located just ahead of the 80 card reading brushes. They should be set by a customer engineer to read any 20 of the 80 card columns. Once they are set, the punched positions in the columns which they read are available from the corresponding 20 hubs on the panel. Control brushes normally are used to recognize distinguishing punches X-9 in the card in order to control functions occurring on the read cycle of the control card, or to control the calculation (program 2 on) of the preceding card.

Figure 19 illustrates schematically the pickup and transfer of a pilot selector from the control brushes.

1. Control brush 6 is set to read column 21.

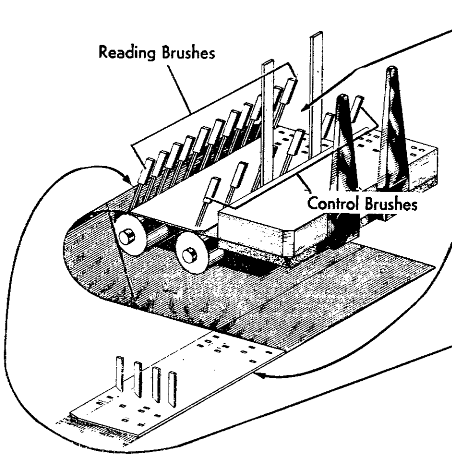
2. The couple exit of pilot selector 3 is wired to the pickup of co-selector 3. The couple exit emits as the pilot selector transfers and is used to expand the number of positions in the selector. Co-selector 3 picks up and drops out like pilot selector 3.

3. Card columns 30-34 of the X21 card are wired to the transferred side of the co-selector; card columns 42-46 of the NX card are wired to the normal side. The common side of the selector is wired to storage entry 2R.

4. The latch-type pilot selector must be dropped out from a read drop-out impulse which will emit on the next read cycle.

## Selection: Machine Function on a Read Cycle

This example (Figure 20) shows a counter controlled by a distinguishing X to add or subtract on the read cycle.



A pilot selector picked up from the control brushes

will transfer for program 2 of the preceding card and remain transferred through the calculation of that card,

and until after the control card has been read by the reading brushes.

*Figure 19. Pilot Selector Picked up from Control Read*

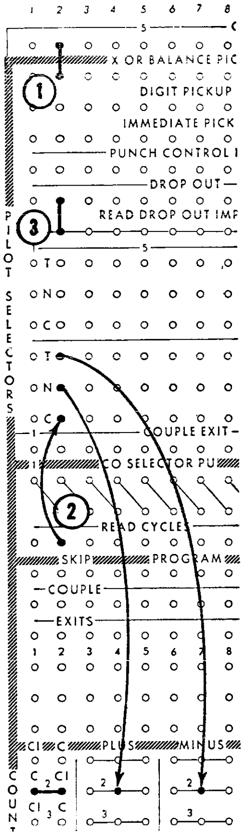


*Figure 20. Selection : Machine Function on Read Cycle*

1. Control brush 2 is set to read the column with the distinguishing punch.

2. A read cycles impulse is controlled through a selector to cause counter 2 to add NX cards and to subtract X cards. Pilot selector 2 is transferred through the entire read cycle of the control-X card; hence, any machine function like storage RI or counter RE, that occurs on the read cycle of a particular type of card, may be selected through a pilot selector picked up from the control brushes.

3. Read drop-out is wired to drop out pilot selector 2.

## Selection: Program Exit from a Distinguishing Punch

A distinguishing punch in a card may be used to select a program exit impulse. This example (Figure 21) illustrates the use of a pilot selector to control program 3 to add in counter 5 only on X cards and to read into storage 4 on program 6 only on NX cards. Since the calculating cycles of a particular type of card are to be controlled, the distinguishing punch

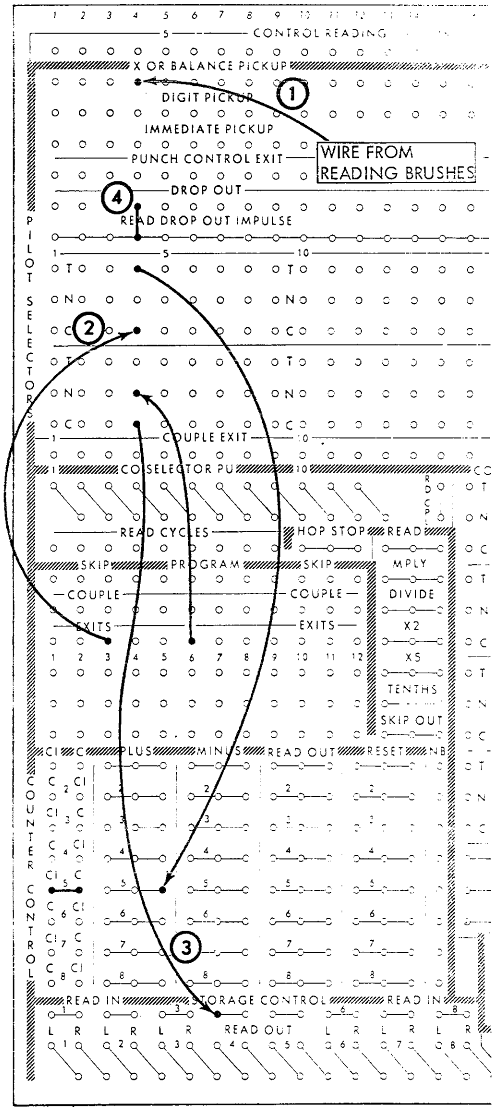
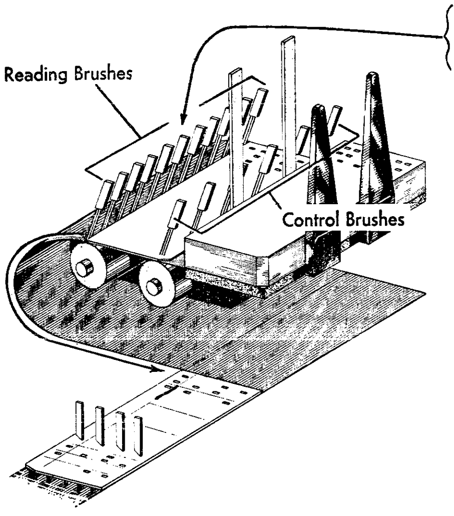


*Figure 21. Selection: Program Exit from Distinguishing Punch*



A pilot selector picked up from the reading brushes

will transfer for program 1 of the same card,

and remain transferred until the following card has been read by the reading brushes.

*Figure 22. Pilot Selector Picked up from Read Brush*

must be recognized from the reading brushes. The schematic in Figure 22 illustrates a pilot selector picked up from the reading brushes. Since digit impulses transfer a pilot selector one cycle later, and it remains transferred until it is dropped out, the selector is controlled during all the calculating cycles of the control card and the reading of the card behind it.

1. Pilot selector 4 is wired from the column containing the distinguishing digit at the reading brushes.

2. Program exit 3 is wired through the transferred side of pilot selector 4 to cause counter 5 to add for X cards.

3. Program exit 6 is wired through the normal side of the selector to storage 4 read-in on NX cards.

4. Read drop-out impulses the selector to drop out at the end of the next read cycle.

## Selection: Expanding Number of Impulses

One of the basic uses of selectors is to allow an impulse to get through only at certain times. Creating additional program exits or read cycles impulses involves isolating an impulse so that it will be available at one particular time only. This example (Figure 23) shows how a co-selector can be wired to provide five additional program exits for program 6, how two rows of program exits may be coupled to serve as one, and how the number of read cycles impulses may be increased.

*Couple (S, 10-12).* Each of the 12 program steps has a corresponding couple exit which emits slightly before the program exit. For this reason they are used to pick up selectors that are to be used in selecting control impulses (left side of control panel) including program exits. The program exits may also be used to pick up selectors, but only for selecting functions that occur after the program impulse starts.

Another use for the program couple is to join two or more program units together to expand the capacity of the program exits from four to as many more as desired. If the couple for program step 9 is wired to the couple of program step 10, the exits for program 9 have been expanded from four to eight and operate simultaneously. In such a case, both program step 9 and program step 10 are recognized by the machine as program 9. Program steps to be coupled for purposes of expansion must be adjacent. If they are not adjacent, such as 2 and 5, the program steps in between will be nullified and the exits for these program steps will not function.

*Read Couple (O-P, 15).* Two common read couple hubs emit impulses that are normally used to pick up selectors on the read cycle. If a RD CPL and a PROGRAM COUPLE both pick up the same co-selector,

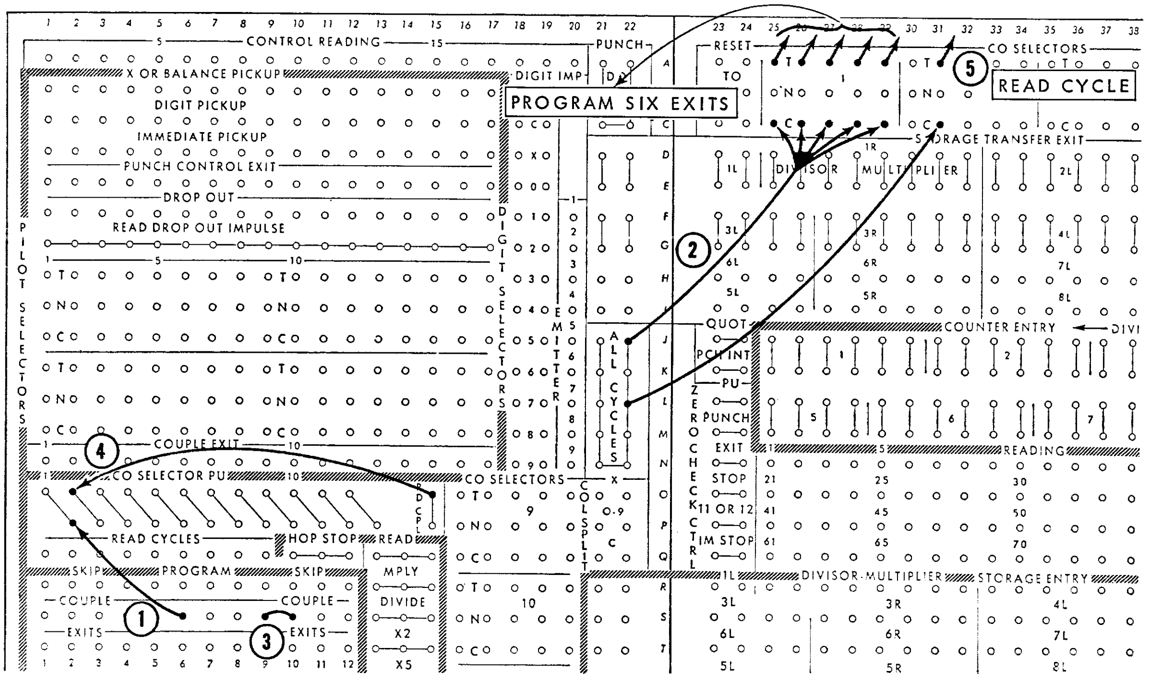


*Figure 23. Selection: Expanding Number of Impulses*

the read cycle and the program cycle occur simultaneously and intervening programs are skipped.

1. The couple exit of program 6 picks up co-selector 1.

2. ALL CYCLES wired into the common side is available out of the transferred side as program 6 exits.

3. The couple exit of program 9 is wired to the couple of program 10. Program exits 9 and 10 are recognized by the machine as program 9. Program 11 would then become 10, program 12 would become program 11, and so on.

4. Read couple picks up co-selector 2.

5. Co-selector 2 is transferred during the entire read cycle. Therefore, an ALL CYCLES entered into the common is available from the transferred side on a read cycle only.

## Selection: Reset to Five

Many problems require the use of a counter on successive steps to develop different answers, each of which may require half adjustment in a different position. This example (Figure 24) shows a planning chart in which counter 6 has been used to develop the two different results *P* and *Q*. The product *P* must be half-adjusted in the third position of counter 6; the quotient must be half-adjusted in the units position.

Only the right-hand panel wiring is shown.

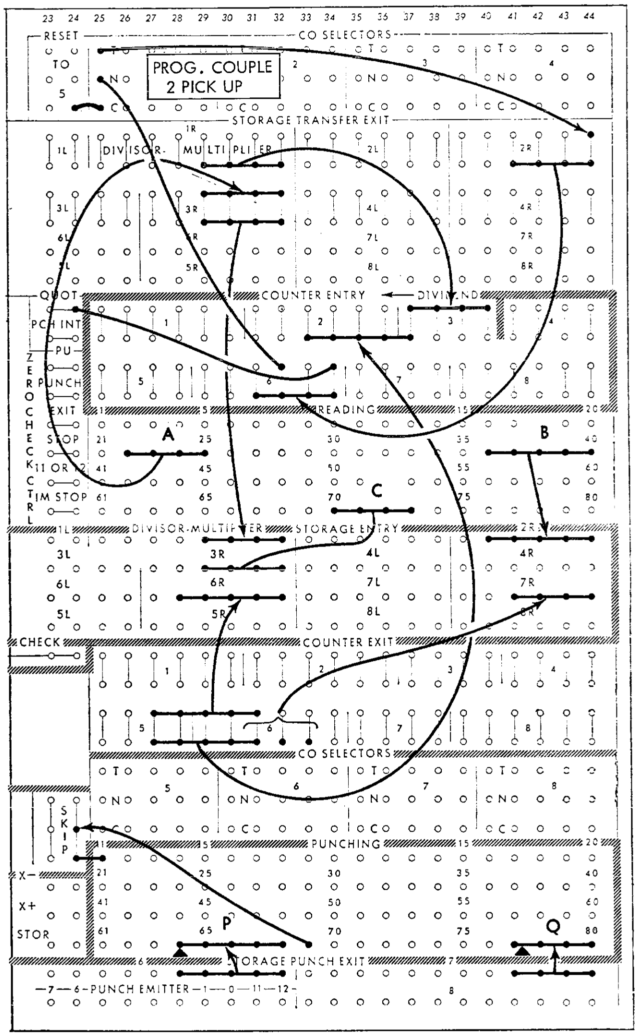
1. Factors *A, B,* and *C* are entered into storage entries 1R, 2R, and 3R, respectively. Factor *A* uses a common pathway through the storage transfer exits of 3R to avoid split wiring on program 3 when factor *C* must be transferred to 1R.

2. The multiplicand is read out of storage unit 2R and entered in counter 6.

3. Factor *C* is transferred from storage transfer exit 3R to the divisor unit 1R, replacing factor *A*.

4. Reset to 5 is wired through the normal side of co-selector 1 to the third position of counter 6 to adjust the product.

5. The adjusted product *P* is transferred from counter 6 to the dividend counter 1-2-3 where it is



| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT | | COUNTER | | | | | | STORAGE UNITS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | DIVR.-MULT. | DIVIDEND | | | | | | | | | | | | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH | | | |
| | | | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R | 7L | 7R |
| | | READ CYCLE | | A<br>10.45 | RE | RE | RE | | RO<br>RE | 5 | | B<br>19575 | | C<br>4200 | | | | | | RI<br>Q<br>487 |
| | 1 | Mult. | | | | | | | + | + P<br>20445875<br>20446375 | | RO | | | | | | | | |
| | 2 | | | RIC<br>42.00 | + | +<br>20446--- | +P | | RO<br>RE | 5 | | | | RO | | | | RI P<br>20446 | | |
| | 3 | Divide | | RO | – | – | – | | + | +Q<br>4868<br>4873 | | | | | | | | | | |

*Figure 24. Selection: Reset to Five*

properly positioned in order to develop a two-decimal adjusted quotient. The number of decimal positions in the dividend counter is determined by adding the two decimals in the divisor plus the two decimals desired in the quotient, plus one more position for half-adjustment.

6. The adjusted product with two positions dropped is wired to storage 6 for punching.

7. The divisor is read out of storage transfer exit 1R into the dividend counter.

8. The quotient hub is wired to the units position of counter 6.

9. The quotient is half-adjusted by wiring a reset-to-5 through the transferred side of co-selector 1 to the units position of counter 6. The co-selector is transferred by a program 2 couple exit to prevent the reset from reaching the units position at any other time.

10. The adjusted quotient is transferred from counter 6 to storage 7R for punching.

11. Storage punch exits 6 and 7 are wired to the proper columns for punching P and Q, respectively.

## Selection: Emitted Factors

Constant factors are often involved in problems. Flexibility in card design as well as problem planning can be achieved by the use of emitted information. Either the emitter or the digit selectors wired as an emitter can be used as a source of manufactured impulses (Figure 25).

*Emitter (F-N, 20).* Digit emitter hubs 1 through 9 automatically emit corresponding digit impulses on

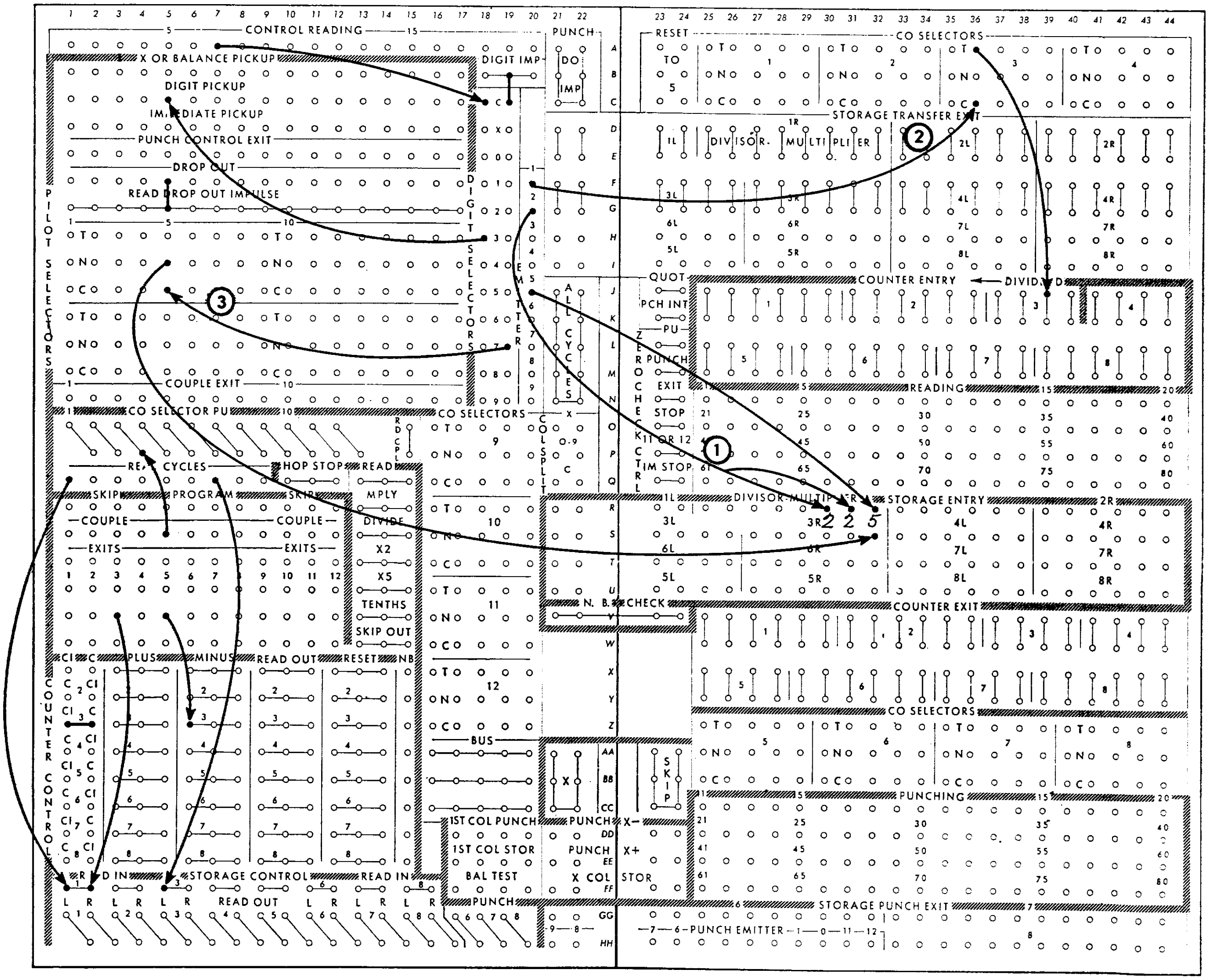


*Figure 25. Selection: Emitted Factors*

every machine cycle. If a problem takes 15 cycles to complete, digit impulses 1 through 9 will emit fifteen times. They may be used at will to enter counters or storage units but, because of their repetitive nature, selection is usually required. This emitter cannot be used to punch emitted data. Only the punch emitter can be wired to PUNCHING.

*Digit Impulse (DI) (B, 18-20).* These hubs emit digit impulses 9 through X on every machine cycle. When a DI is wired to C of a digit selector, the selector becomes an emitter.

*Digit Selectors (C-N, 18-19).* There are two digit selectors, each having 0-9, X, and C hubs. C is the common hub and, when it is wired from reading or control brushes, counters, or storage units, specific punches can be selected. (There is no digit impulse available from zero from counter or storage exits.) When C is wired from DI, the selector becomes an emitter.

1. Emitted factor 225 is entered into 1R on a read cycle and program 3.
2. A minus ten is entered in counter 3 on program 5 through co-selector 3.
3. Storage unit 3R receives a seven on the read cycle of a no-digit 3 in the column read by control brush 7.

### Selection: Skipping Programs

The normal sequence of program steps can be altered as the result of a decision made by the machine, transmitted through a selector. Whenever the sequence of program steps is altered, it is called branching or program skipping (Figure 26).

Branching can be the result of control instructions punched in the card or of a decision made during calculation. If branching is the result of the presence or absence of a control punch, pilot selectors must be used to transfer the circuits to the alternative operation during the entire calculating time. If branching is the result of a positive or negative balance in a counter, pilot selectors or co-selectors picked up immediately can be used to transfer to the alternative operation.

*Program Skip (R, 1-12).* Each program step has a corresponding skip entry hub located directly above the program couple hubs. When a program skip entry

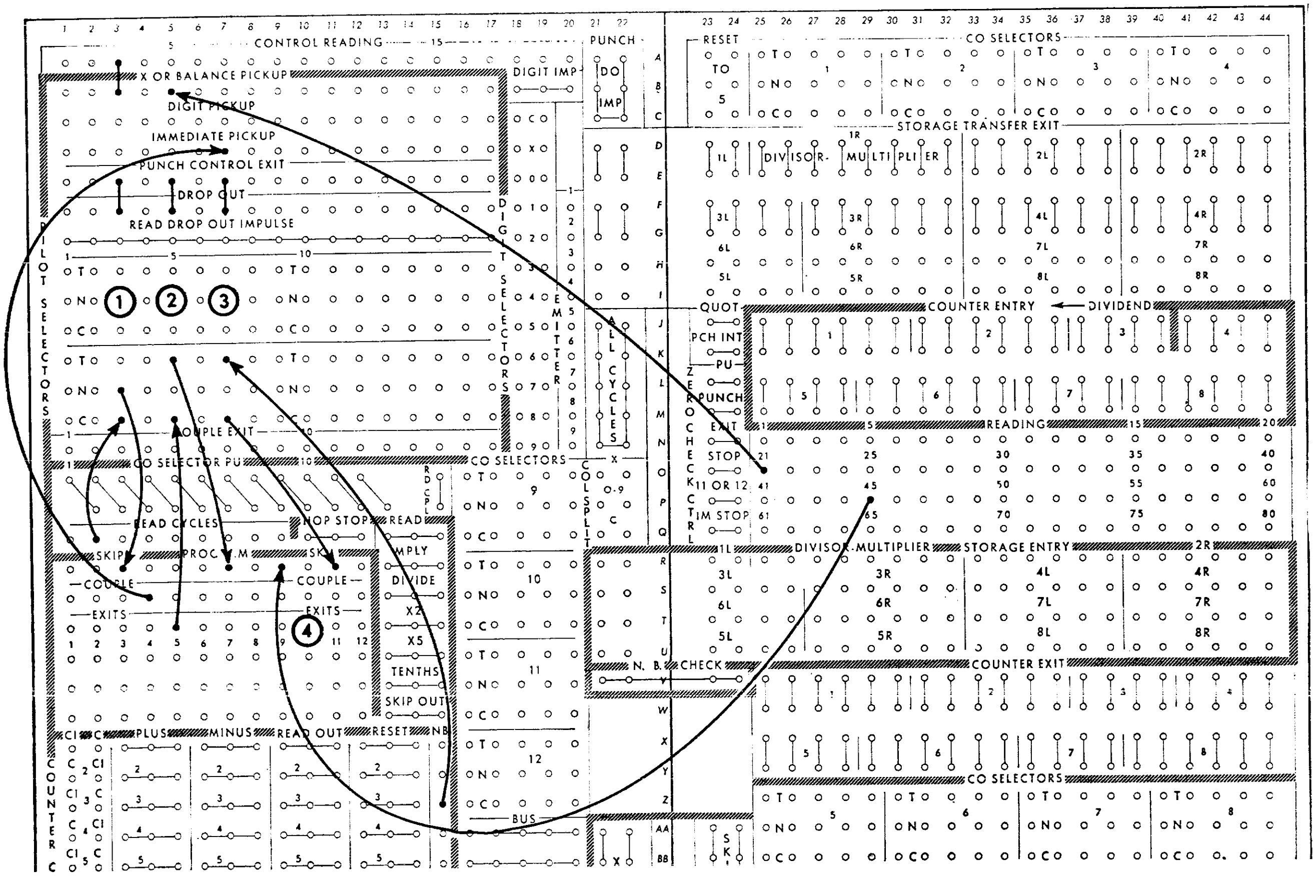


*Figure 26. Selection: Program Skipping*

hub is impulsed, programs preceding it are skipped. When a program is skipped, the cycle is eliminated, and the corresponding program exit hubs are inactive.

The program exit may be used to impulse a skip entry, but the program couple cannot be used.

Depending upon the problem, certain programs may be skipped immediately after the card is read, or from one program to another within a problem.

Figure 26 illustrates the wiring to skip programs under various conditions.

1. Programs 1 and 2 may be skipped on NX cards by wiring a read cycles impulse through the normal side of a selector, picked up from a control brush to program skip 3.

2. Program 6 only may be skipped on X cards by wiring a program 5 exit, through the transferred side of a selector picked up from the reading brush, to program skip entry 7.

3. Programs may be skipped from one program to another by wiring an NB impulse to program skip entry. If counter 3 is negative at the end of program 4, a skip is made to program 11.

4. An X wired directly from a reading brush to program skip entry 9 will skip programs 1 through 8.

## Selection: Punching

Punching in different fields of a card, or selecting information to be punched, may be controlled either from distinguishing punching in the card or by means of some condition developed during the calculation. These two situations present different timing problems in the wiring necessary to control the selectors. It

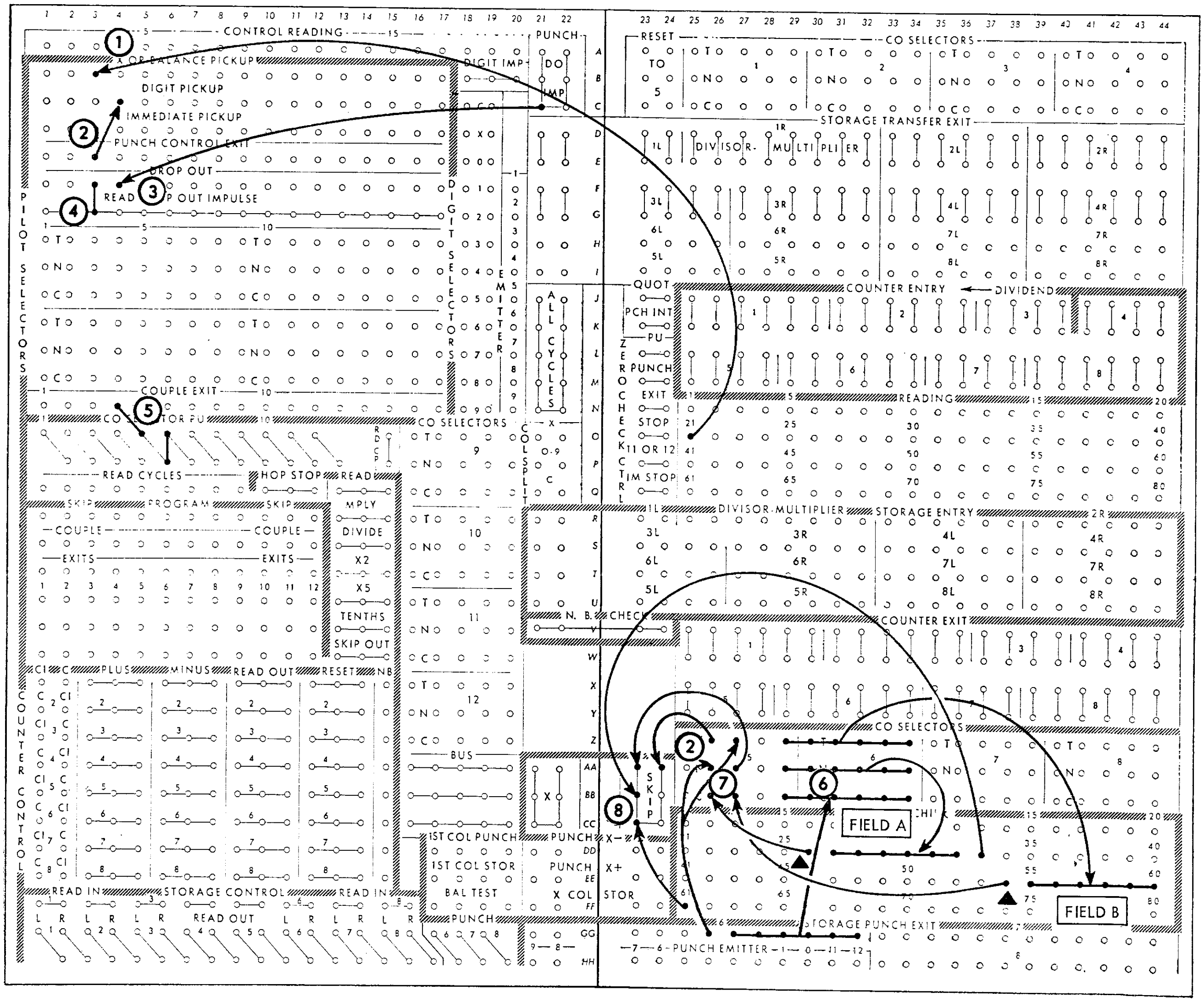


*Figure 27. Selection: Punching from Distinguishing Punch*

should be recalled that punching continues while the following card is being read and calculated. However, programming cannot be completed on one card until the preceding card has been completely punched, since this would involve reading a new result into a punch storage unit before the units position of the previous result had been punched in the preceding card. An automatic interlock prevents this. Because of the overlapping of the punching of one card with the reading and calculation of the next, the switching of punch circuits, if controlled directly from card reading or calculation, might occur during the punch operation. This would produce incorrect results.

Figures 27 and 28 illustrate the proper control of selectors during punching.

### FROM A DISTINGUISHING PUNCH

Figure 27 is an example of punch selection as a result of a distinguishing punch in the card. The result stored in storage punch exit 6 must punch in field A on NX cards and in field B on X cards.

The control punch read from the reading brushes must control a selector all during the punching operation, no matter how long that may take. Two selectors are required to accomplish this. One selector is used to recognize the controlling X punch from the card. A second pilot selector, picked up by the punch control exit of the first pilot selector, will be latched throughout the entire punching of the card if it is dropped out by a punch drop-out impulse. The first pilot selector can be dropped out by a read drop-out.

This wiring insures that the selector switching the punching circuits will be controlled by information read from that same card, and also that the selectors remain transferred for the duration of punching.

1. The controlling X in column 21 picks up pilot selector 3. This pilot selector will be transferred during the entire calculation of the X21 card.
2. The punch control exit of pilot selector 3 emits just before the X21 card is moved into punching position and can be wired to the D pickup of pilot selector 4 to latch this selector.
3. The punch DO impulse drops out pilot selector 4 immediately after punching is completed. This means that pilot selector 4 is transferred during the entire punching of the X2$^1$ card, regardless of how long the punching may take.
4. Pilot selector 3 can be dropped out from a read drop-out.
5. The couple exit of pilot selector 4 picks up co-selectors 5 and 6 to expand the number of available positions.
6. The result from storage punch exit 6 is wired through the co-selectors to punch in field A on NX21 cards and field B in X21 cards.
7. A skip insert is placed in the first position of each field to be punched (columns 26 and 54). Since only one field or the other is punched in any one card, it is necessary to bypass one of the inserts on every card. If a skip impulse is wired to the punching position where the insert is placed, skipping takes precedence. Therefore, the high-order position is selected through the normal side of the co-selector to reach column 26 on NX cards. A skip impulse is wired through the transferred side to reach column 26 on X cards. This prevents punching in columns 26 through 32 on X cards. The same wiring principle is followed for the X field.
8. The columns following each field punched (columns 33 and 61) are wired to skip.

### FROM CALCULATION

Figure 28 shows the selector wiring for punching a card conditioned by the results of a calculation. In this example, an X is punched if a negative balance

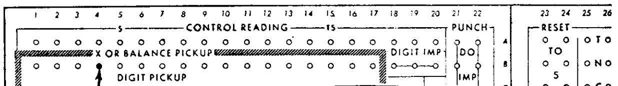


*Figure 28. Selection: From Calculation*

was developed on program 5. The condition may be the presence of specific digits, as well as the condition in a counter. In either case, the condition is discovered on one of the calculation steps. In general, the problem is the same as in Figure 27: the punching of one card continues on into the reading and calculation of the next card. In this example, punch selection is to be controlled from a program step, but to a certain extent care must be taken to tie the punching of a card with its own program time. A single selector should be used. It can be picked up from the specified condition (negative balance or presence of a certain digit) on a particular program step. The selector should be wired for punch drop-out so that it remains transferred during punching of the entire card. In addition, to prevent picking up the selector from one set of calculations before punching in the previous card is completed, punch interlock must be impulsed on the program step on which the condition is recognized.

*Punch Interlock (K, 23-24).* This is an entry hub for a program exit to suspend calculation until punching is completed. For example, when one card is being punched, ordinarily another card is being calculated. If punch selection is controlled by a condition arising during a calculation, such as a negative balance, then the selection for the card being punched could be disturbed by the calculation of the following card. By impulsing punch interlock from the same program on which the test is made to pick up a selector, that program is delayed and its exit does not emit until the one card has been punched and the next card is fed into punching position.

1. The negative balance test exit of counter 6 is wired through pilot selector 2 and picked up on program couple 5 to isolate a negative condition on that program only.

2. The negative test impulse picks up pilot selector 4 at the X or balance pickup to latch the selector.

3. Pilot selector 4 is wired for punch drop-out so that it will be transferred until punching is complete.

4. An X from the emitter is wired through the transferred side of pilot selector 4 to punch in column 42 in all cards for which a negative balance was developed on program 5.

5. Program exit 5 impulses punch interlock, which prevents the occurrence of program 5 and the resulting negative balance test before the card has been punched. If punch interlock were not wired, an X would be punched in one card as a result of a negative balance condition developed from the next card.

### Selection: Negative Balance

One of the most common types of selection is based on the ability of the 602 to recognize a positive or negative result in a counter. Many conditions can be detected. A zero balance can be identified, a negative number can be recognized, or numbers can be checked to determine if they are above or below a determined limit. This example (Figure 29) illustrates the selection of the factor to be added depending upon whether the balance in the counter is either positive or negative: $A - B = \pm T + (C \text{ or } D)$. A zero balance in a counter is recognized as a positive balance.

*Negative Balance (NB) (X-EE, 15).* Each counter has an NB hub which emits an X-timed impulse at the end of the cycle during which the counter turns negative. Testing for plus or minus balances may be done on any program step except those used for multiplication or division.

NB is normally wired to the X or balance pickup of a pilot selector in order to have the selector reflect the balance in the counter. If the selector is transferred, the counter is negative; if it is normal, the counter is positive.

The planning chart should be analyzed to see how often a counter may turn negative in order to be sure that the selector through which the negative balance test is made reflects the condition of the counter only on a specific program. For example, a counter may be negative on program 1, turn positive on program 2, and negative again on program 3. If NB is wired directly to the X or balance pickup of the selector, the selector is picked up on program 1 and will latch and remain transferred for the entire problem. Thus, the pilot selector would not always reflect the true position of the counter on program 3.

NB should never be wired to the I pickup of a pilot selector, or to a co-selector pickup directly. Such wiring would cause the selectors to transfer at the end of a cycle during which the counter went negative and drop out immediately thereafter. The selectors would hold only for the duration of an X impulse.

The following is a description of first the planning chart and then the right-hand panel wiring.

*Read Cycle.* Factors *A, B, C,* and *D* are entered into their respective counter and storage units.

*Program 1.* Factor *B* is read out of storage 2L and subtracted in counter 2. If at the end of this program step the result is negative, the NB test exit emits to pick up pilot selector 2.

*Program 2.* The program exit impulse is selected to read out of either 2R or 3L and to add in counter 2.

*Program 3.* Counter 2 is read out, storage unit 6 is impulsed to read in, the counter is reset, and storage 6 is told to punch.

1. NB test exit will emit if counter 2 is minus and will pick up pilot selector 3.

2. Program exit 2 is wired through the selector to read out storage 2R (*C*) if counter 2 was plus and to read out storage 3L (*D*) if counter 2 was minus.

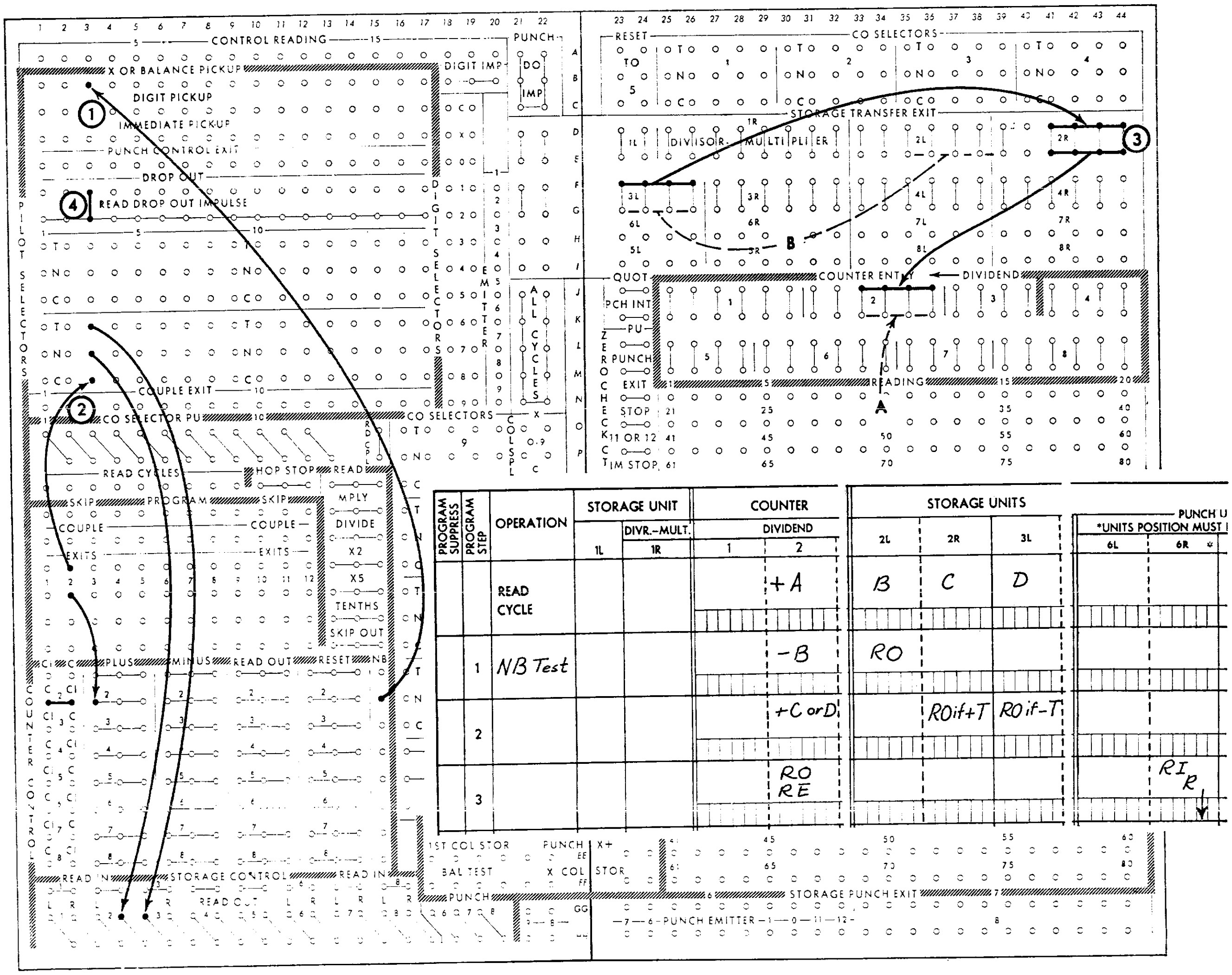
3. Storage transfer exits 3L and 2R are wired to counter entry 2. The storage transfer exits need not be selected since these exits do not emit unless their read-out hub is impulsed. Therefore, only one selector position is required to wire this problem.

4. Pilot selector 3 is dropped by a read drop-out so that the selector will remain transferred from program 2 on and until the next card has been read.

## Selection: Zero Balance

Zero balances are recognized by counters in the 602 as plus balances. Therefore, it would not be possible to distinguish between zero balances and plus balances by means of the negative balance test shot.

Various methods for detecting zero balances have been devised.



| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT |  | COUNTER |  | STORAGE UNITS |  |  | PUNCH U *UNITS POSITION MUST I |  |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  | DIVR.-MULT. | DIVIDEND |  |  |  |  |  |  |
|  |  |  | 1L | 1R | 1 | 2 | 2L | 2R | 3L | 6L | 6R |
|  |  | READ CYCLE |  |  |  | +A | B | C | D |  |  |
|  | 1 | NB Test |  |  |  | -B | RO |  |  |  |  |
|  | 2 |  |  |  |  | +C or D |  | ROif+T | RO if-T |  |  |
|  | 3 |  |  |  |  | RO RE |  |  |  | RI R |  |

*Figure 29. Selection: Negative Balance*

For example, two program steps can be used to test the signs of a counter. If the amount standing in a counter on the first step tests positive, that amount may be all zeros or a true positive amount. If a one is subtracted in the counter on the next step, the counter could only turn minus if the previous amount was a zero balance. The rule, then, may be stated as follows: a plus on the first test followed by a minus on the second test indicates that a zero balance was in the counter prior to the tests. A selector network can establish this sequence of signs.

For example:

| | ZERO BALANCE | POSITIVE BALANCE | NEGATIVE BALANCE |
|---|---|---|---|
| Balance in the counter | 0000 | 0001 | —0000 |
| Program 1 NB test | +Balance | +Balance | —Balance |
| Program 2 NB test | —0001 | 0000 | —0002 |
| | —Balance | +Balance | —Balance |

Figure 30 shows the selector network set up to recognize either a zero balance, a positive balance, or a negative balance.

1. The negative balance test impulse is selected on the program steps on which the test is made to prevent either pilot selector from transferring from a negative condition prior to the test.

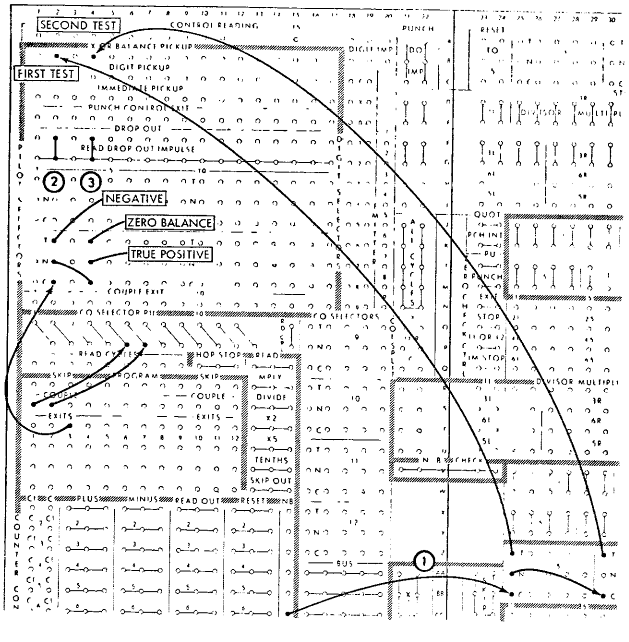


*Figure 30. Selection: Zero Balance*

2. Pilot selector 2 will be normal for zero balances or plus balances, and transferred for minus balances.

3. Pilot selector 4 will be normal for positive balances, and transferred for zero balances.

Any impulse wired through the selectors in the manner shown reflects one of these three conditions. If the original amount was in storage, as well as the counter used for testing the balance, the selected program exit could not be used to read out the proper balance or to direct the machine to do any of the possible commands required.

A second method of recognizing a zero balance uses two internal connections which exist whenever a counter zero balances. There is a common connection between CI and C and a common connection between C and NB. Each of these methods is illustrated under "Typical Applications."

## Selection: Sign Control

In multiplication or division, the sign of the result for two factors is determined by the rule that like signs produce a positive result and unlike signs produce a negative result. The result will be negative only when there is an odd number of minus signs. Many formulas, especially in the engineering field, contain positive or negative factors whose signs must be recognized by the machine before the correct result of multiplication or division can be obtained. The sign of a factor in terms of machine operation is the distinguishing punch (digit or X) which identifies an amount as positive or negative.

On the 602, a selector network can be used to determine the sign of the result from the signs of the factors indicated in the card. The signs may be designated by X or digit punches. This example (Figure 31) shows three different selector networks for sign determination. The test could be from impulses such as a program exit, punch control exit, or all cycles, depending on the problem.

Figure 31A provides a test impulse for both positive and negative results.

Figure 31B provides a test impulse for negative results only.

Figure 31C provides a test impulse for punch selection of negative result.

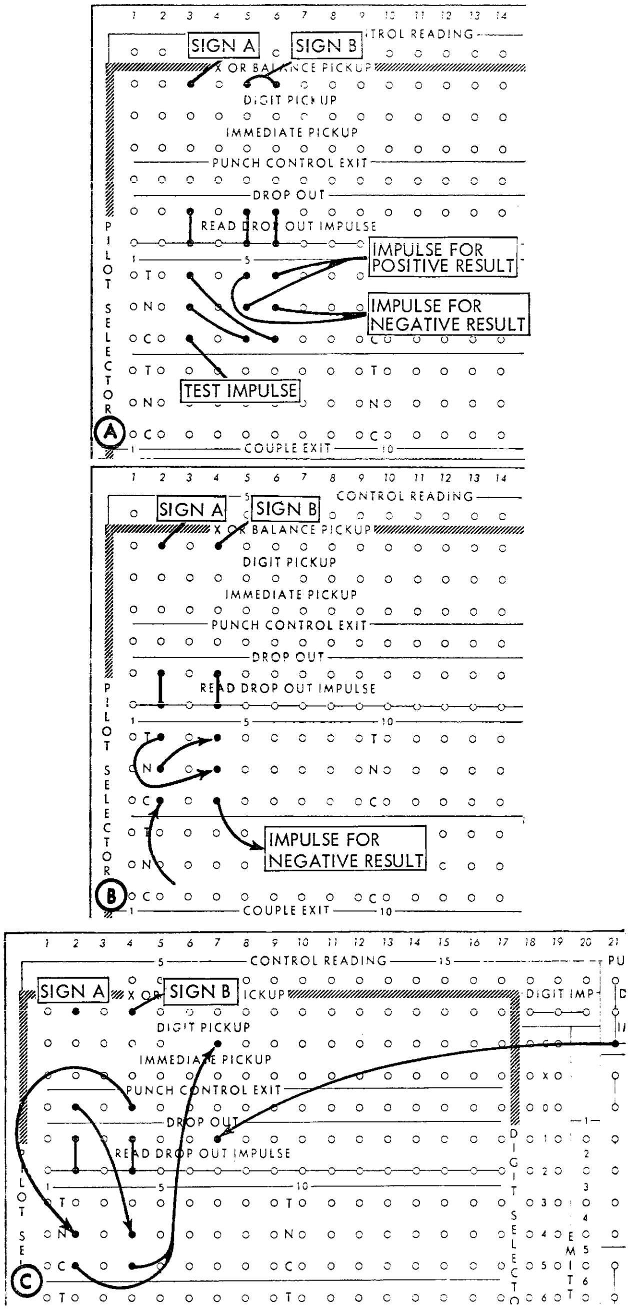


*Figure 31. Selection: Sign Control*

# Multiple-Card Operations

MANY 602 applications involve the processing of multiple cards for the completion of a single problem in an application. This processing of multiple cards can be a group operation or a summarizing operation.

## GROUP MULTIPLICATION OR DIVISION

THIS type of operation is performed wherever the two factors of a multiplication or division calculation are in different cards and one of those factors is a constant and the other is a variable.

In a group payroll, for example, if a rate is based on the work done by the group, it is the multiplier for calculating the earnings for each man in the group. That rate can be punched into a master card and used as a group multiplier. The rate can be entered into a storage unit from the master card. As each earnings card is calculated, the rate is read out of the storage unit to be used as the multiplier in the multiplication.

Each time a new master card is read, the new rate replaces the previous one in the storage unit and is used for multiplying the following cards. If the calculation is a division operation, the group factor (either the divisor or the dividend) is stored and used in the same way.

In a group multiplication or division, either the master cards or the detail cards must be identified by a control punch to distinguish between them.

In addition to the usual machine commands for multiplication and division, multiple-card operations require use of additional instructions. These are:

1. Store the group factor for use with the successive cards of that group. Each time a new group card is read, the new constant replaces the previous one in storage.
2. Prevent punching in the group card.
3. Suspend programming for the group card.

This example illustrates a planning chart and control panel diagram for a group multiplication in which the multiplier is the group factor stored in an X card (Figure 32).

*Skip out (W, 13-15).* When a card being read is to be skipped out without punching, as in the case of an X master card, a skip-out hub may be impulsed from any impulse on the read cycle. If all the program and punch impulses are suspended, the card is ejected immediately into the stacker. If the programs for the problem are not suspended for an X card, the card skips to column 80 and waits there until completion of the program cycles before passing to the stacker. If the punch impulses are not suspended for an X card, either by selection or program skipping, the card skips to column 80, where it remains until the control panel is removed or until the main line switch is turned off. The primary function of skip-out, therefore, is to skip the card to column 80. If read cycles is wired to punch, it must be selected through a pilot selector. If a program exit is wired to punch and the program is skipped, no selection is necessary.

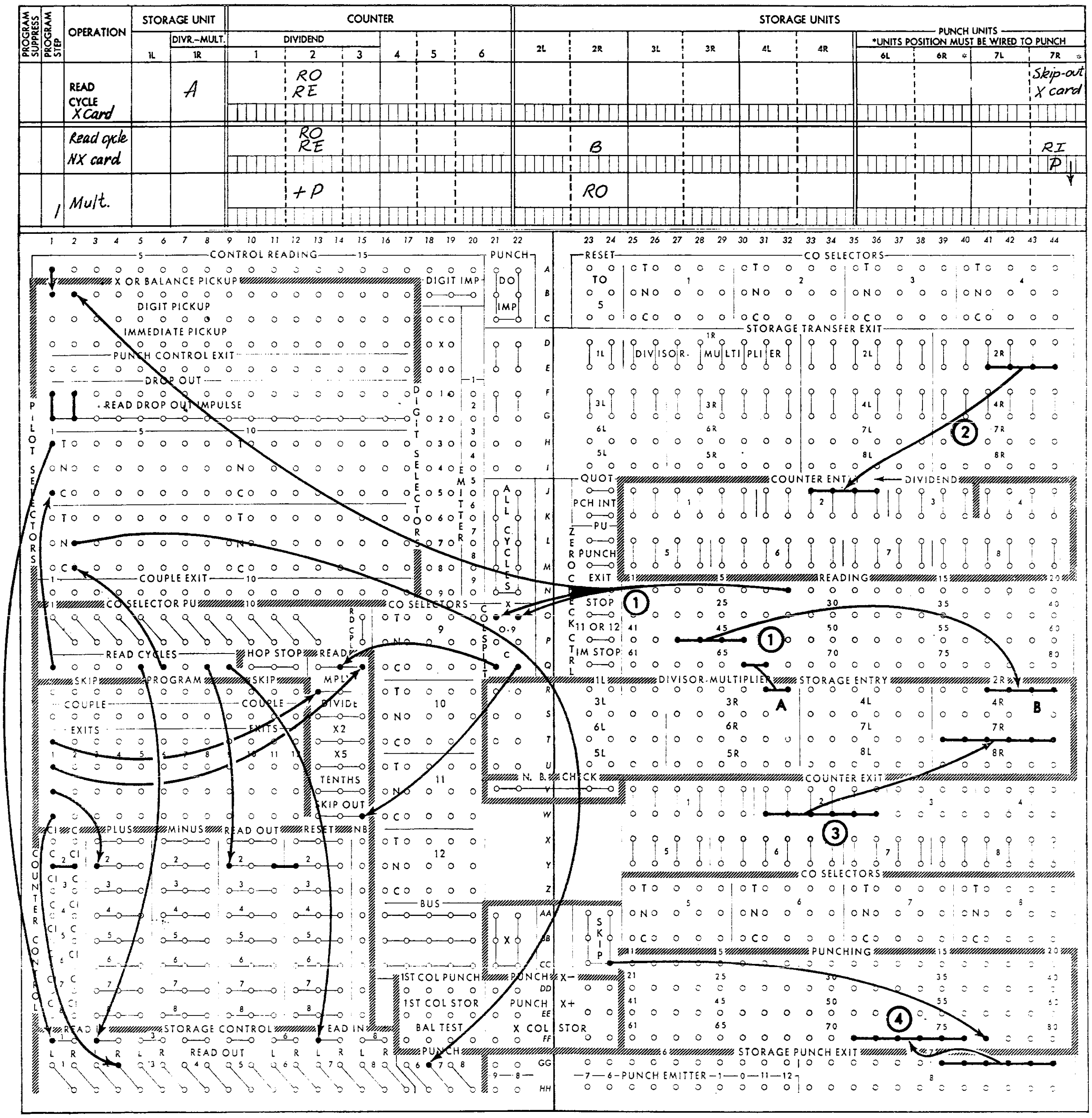
### Planning Chart and Control Wiring

*Read X Card. A,* the group factor, is entered into the multiplier for X cards only. This is done by selecting the read cycles impulse to 1R through pilot selector 1, picked up by an X from the control reading brushes. When a pilot selector is picked up from a card at the control read station and is wired for read drop-out, it remains transferred until the same card passes the 80 reading brushes, or, in other words, through the following read cycle.

It is unnecessary to take program 1 on the X card.

All that need be done is to read the group factor *A* and then eject the card. Programming is eliminated by wiring the master card X from read brushes to READ. The X master card is skipped to column 80 by split-wiring the X from the read brushes to SKIP OUT. The X impulse is tested through two column splits to eliminate interference from digits punched in the same column.

Whether the card is ejected into the stacker or not after reaching column 80 will depend upon how the punch impulse is controlled. Whenever PUNCH receives an impulse, it must punch before the card will



| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT |  | COUNTER |  |  |  |  |  | STORAGE UNITS |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  | DIVR.-MULT. | DIVIDEND |  |  |  |  |  |  |  |  |  |  |  | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH |  |  |  |
|  |  |  | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R | 7L | 7R |
|  |  | READ CYCLE X Card |  | A |  | RO<br>RE |  |  |  |  |  |  |  |  |  |  |  |  |  | Skip-out X card |
|  |  | Read cycle NX card |  |  |  | RO<br>RE |  |  |  |  |  | B |  |  |  |  |  |  |  | RI<br>P |
|  | 1 | Mult. |  |  |  | +P |  |  |  |  |  | RO |  |  |  |  |  |  |  |  |

*Figure 32. Group Multiplication*

be ejected. In this problem, X master cards are to be ejected without punching and NX cards are to punch before being ejected. In order to keep the read cycles impulse from reaching the punch hub on X cards, it is selected through N of pilot selector 2, picked up from the read brushes. When a pilot selector is picked up from a card at the read brushes and is wired for read drop-out, it remains transferred for all the calculations and the punching of the card, as well as through the read cycle of the following card. The punch hub in this problem will be impulsed for NX cards only.

*Read NX Card.* *B* is entered into storage unit 2. This factor is punched in NX cards only, and the read cycles impulse that controls its entry can be wired directly. Even if this field were punched on the X card, it would be cleared upon entry of *B* from the NX card.

*Program 1.* The multiplicand in storage unit 2R is read out to be multiplied by *A*. Counter 2 is impulsed to add to develop the product. Program 1, being the last program used, is wired to READ.

*Read NX Card.* Counter 2 is cleared on every read cycle. The product for each NX card is read into storage entry 7 for punching. The read cycle is controlled so that it never reaches the punch switch on X cards. Storage control read-in 7 is wired directly from read cycles because nothing is read into it from the X card.

### Right-Hand Panel Wiring

1. *A* is wired to the multiplier and *B* to storage entry 2R. The X in column 9 of the master card is wired as previously explained under "Read X Card."

2. The multiplicand is read out of 2R into counter 2 to develop the product.

3. The product is wired to storage entry 7R where it is held for punching.

4. The product is wired from storage exit 7R to punch.

## GROUP MULTIPLICATION WITH DECIMAL ACCUMULATION

IN GROUP multiplication, the half-adjustment, if introduced for each card, would show a discrepancy in the group total of as much as half a cent for each card. By accumulating the decimals for each card and introducing the half-correction once for each group, there will be no discrepancy in the group total. Figure 33 demonstrates this method of decimal accumulation.

| Time | Rate | Individually Adjusted | Actual Calculation | Accumulated Decimals | Decimal Accumulated Adjustment |
|---|---|---|---|---|---|
| (half adjustment) | | | | 0.5 | |
| 2.5 | 1.25 | 3.13 | 3.125 | 1.0 | 3.13 |
| 2.5 | 1.25 | 3.13 | 3.125 | 0.5 | 3.12 |
| 2.5 | 1.25 | 3.13 | 3.125 | 1.0 | 3.13 |
| 0.5 | 1.25 | .63 | 0.625 | .5 | 0.62 |
| Total 8.0 | | 10.02 | 10.000 | .5 | 10.00 |
| Daily 8.0 | 1.25 | 10.00 | 10.000 | | |

*Figure 33. Principle of Decimal Accumulation*

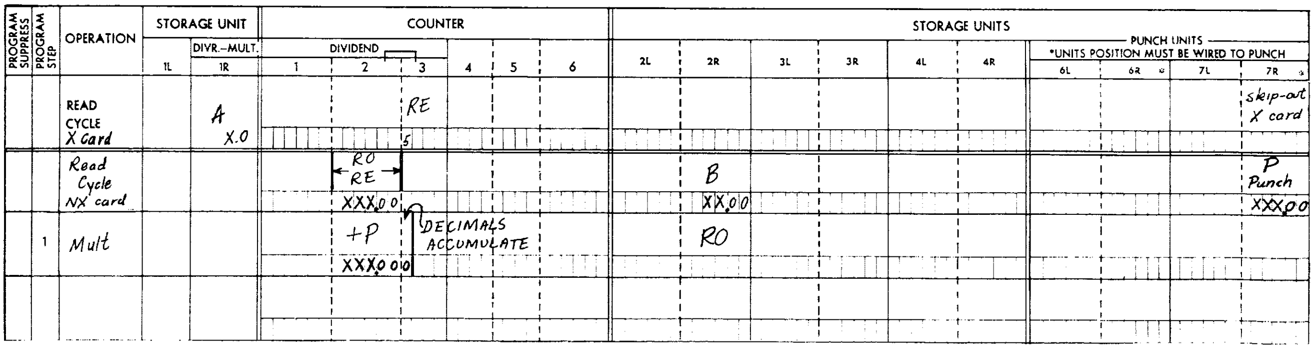


*Figure 34. Group Multiplication*

In the previous illustration of group multiplication, the product was not half-adjusted. Provision can be made for decimal accumulation with a few changes in the planning. Figure 34 shows the same planning chart with decimal accumulation. Only those steps which adjust the total are explained.

*Read Cycle X Card.* For decimal accumulation, counter 3 will be reset on the X card only by wiring a read cycles impulse through the transferred hubs of a pilot selector picked up from the control brush. This will introduce a 5 into the high-order counter position to adjust the group total.

*Program 1.* The multiplicand in storage unit 2R is read out into counter 2-3 to develop the product of three decimal positions, of which two will be punched and one dropped. Since the dropped decimal position must accumulate for all NX cards, it is wired to the left-hand position of counter 2. Counter 2 develops the adjusted product and is cleared for each card.

*Read Cycle NX Card.* The product, less the decimal position in counter 3, is wired to storage entry 7R for punching.

## SUMMARY PUNCHING

$A \times B = P;\ P_1 + P_2 \ldots = \Sigma P$

THE USE of calculating machines is greatly expanded when the factors of each card of a group, or the calculated results for each card of a group, are accumulated and summarized. Summary punching, then, is a second type of 602 multiple-card operation; it is the accumulation of information from each card of a group and recording that information into a separate summary card. In the same operation, a calculation can be made in each detail card as well as in the summary card.

Figure 35 shows the sequence of cards to be processed through the calculator. In this example, each detail card is a NX item card which is extended and punched. The extended amounts are accumulated and punched into the X summary card.

The series of planning charts with explanations show the accumulation of the total item amount in counter 2 and the machine commands for each type of card as it moves through the machine. Careful analysis of the charts illustrates the importance of noting the movement of a card past the various stations so that selectors will be transferred at the correct time to control operation of calculation or punching.

*Hopper Stop (Q. 10-12).* In some applications, such as group multiplication or summary punching, it is desirable to stop the machine when the last card is fed to the control brushes so that additional cards can be placed in the hopper before restarting and thus prevent a break in sequence. When the hopper stop is wired from ALL CYCLES, no further feeding takes place after the last card is sensed at the control brushes unless the start key is depressed.

### Right-Hand Panel Wiring

1. The quantity $A$ is entered into the multiplier unit 1R, and unit price $B$ into storage entry 2R.

2. The multiplicand $B$ is read out of 2R into counter 6 to develop the product. A reset-to-5 is entered in the units position of the counter for half-adjustment.

3. The item amount $P$ is read into storage entry 6R for punching and into counter entry 2 for accumulation.

4. The sum of the item amounts $\Sigma P$ is entered into storage 7 for punching.

5. Item amount $P$ is wired to punch in all NX item cards in columns 44-49 from storage unit 6 and the $\Sigma P$ from storage 7 in columns 52-56. Since a result must punch in a different field on each type of card, the high-order position in storage must be selected in each result. The high-order position of storage 6 must be impulsed to punch; it is wired to the normal side of pilot selector 6, a skip impulse is wired to through the transferred side, and the common to column 44, the first column to be punched.

Since this is punch selection, pilot selector 6 must be controlled to pick up and hold throughout the punching of X cards. This is done by wiring the punch control exit of pilot selector 5 to the D PU of pilot selector 6. Pilot selector 6 is dropped out by a punch drop-out impulse.

6. The high-order position of storage 7 must punch in column 52 of X cards only; it is wired to the transferred side of pilot selector 6, a skip impulse is wired through the normal side, and the common is wired to column 52.

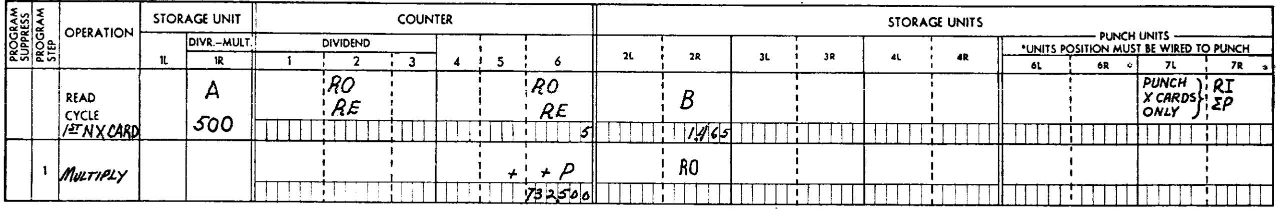
7. ALL CYCLES is wired to hopper stop to prevent a break in sequence as the hopper empties.

## FIRST NX CARD OF A GROUP

| | ITEM CARD 1 | | | | | |
|---|---|---|---|---|---|---|
| | ITEM NO | QTY | UNIT PRICE | | ITEM AMT | |
| | 475 | A<br>500 | X B<br>1.465 | = | P<br>732.50 | |

Read Cycle: Factors A (500) and B (1465) are read into their respective storage units. Counter 6, which will be used to develop the product (732.50), is read out and reset and a 5 is positioned for half-adjustment. Counter 2, which will accumulate the total of item amounts, is read out and reset prior to the addition of the first product of the group.

Program 1. Multiply hub is impulsed, the multiplicand is read out of 2R, and counters 5 and 6 are impulsed to add to develop item amount. Since this is the only program step required to complete the calculation on the detail card, programming is suspended after this step by wiring a program 1 exit to read, and the next card feeds.

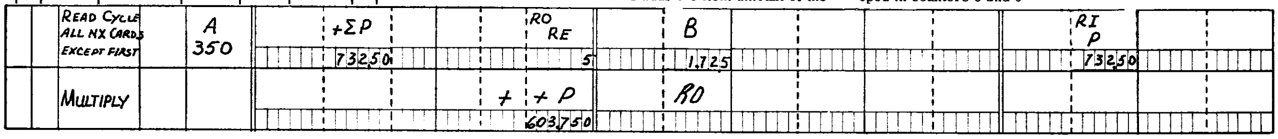


## ALL NX CARDS EXCEPT THE FIRST

| | ITEM CARD 2 | | | | | |
|---|---|---|---|---|---|---|
| | ITEM NO | QTY | UNIT PRICE | | ITEM AMT | |
| | 681 | A<br>350 | X B<br>1.725 | = | P<br>603.75 | |

Read Cycle: The multiplier unit, 1R, is impulsed to accept quantity A and storage entry 2 to accept unit priceB. The preceding card is now in the punch bed. Counters 5 and 6, which contain the product (732.50) of the preceding card are read into storage for punching in columns 44-49 and into counter 2 to develop the sum of the item amounts. Since counter 2 adds the item amount of the NX cards only, the counter is impulsed to add through the normal side of a pilot selector picked up by the distinguishing X summary card at the reading brushes. The reading brushes must be used in order to transfer the selector during the following read cycle.

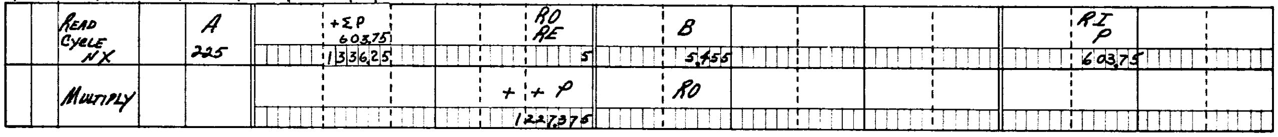
Program 1: A and B are multiplied, and the product is developed in counters 5 and 6



| | ITEM CARD 3 | | | | | |
|---|---|---|---|---|---|---|
| | ITEM NO | QTY | UNIT PRICE | | ITEM AMT | |
| | 749 | A<br>225 | X B<br>5.455 | = | P<br>1227.38 | |

## ALL NX CARDS EXCEPT THE FIRST

(Same steps as for preceding section)

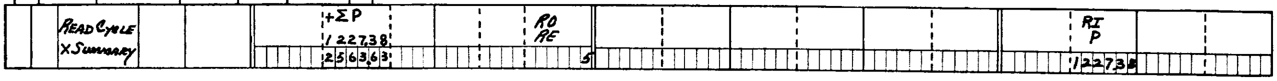


| | TOTAL AMOUNT CARD | | | | |
|---|---|---|---|---|---|
| | ITEM NO | | | | Sum of Item Amts |
| | X Punch | | | | ΣP<br>2563.63 |

## X SUMMARY CARD

Read Cycle: The adjusted product (1227.38) is read out of counters 5 and 6 into storage 7 for punching into the last NX card of the group, now in the punch bed. This new product is also added to the sum of the item amounts in counter 2.

Sensing the X in the summary card at the read brushes signals the end of a group, stops programming because no calculation is required for this card, and picks up a pilot selector to control punching of the accumulated total into the summary card.



## FIRST NX CARD OF THE NEXT GROUP

| | ITEM CARD 1 | | | | | |
|---|---|---|---|---|---|---|
| | ITEM NO | QTY | UNIT PRICE | | ITEM AMT | |
| | | A<br>150 | X B<br>9.550 | = | P<br>1432.50 | |

Read Cycle: The total of the item amounts (2563.63) accumulated in counter 2 is read out into storage 7 for punching in columns 52-57 of the X card, which is now in the punch bed.

The read cycles impulse to control the counter and storage unit is selected through the transferred side of the pilot selector which was picked up when the X card was at the reading station.

Factors A and B are entered into their respective units for multiplying.

Program 1: This program develops the product for the first card and the sequence of steps begins again for the new group.

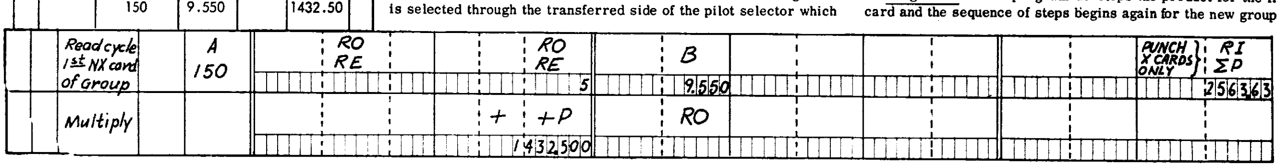


*Figure 35A. Summary Punching Planning Chart*

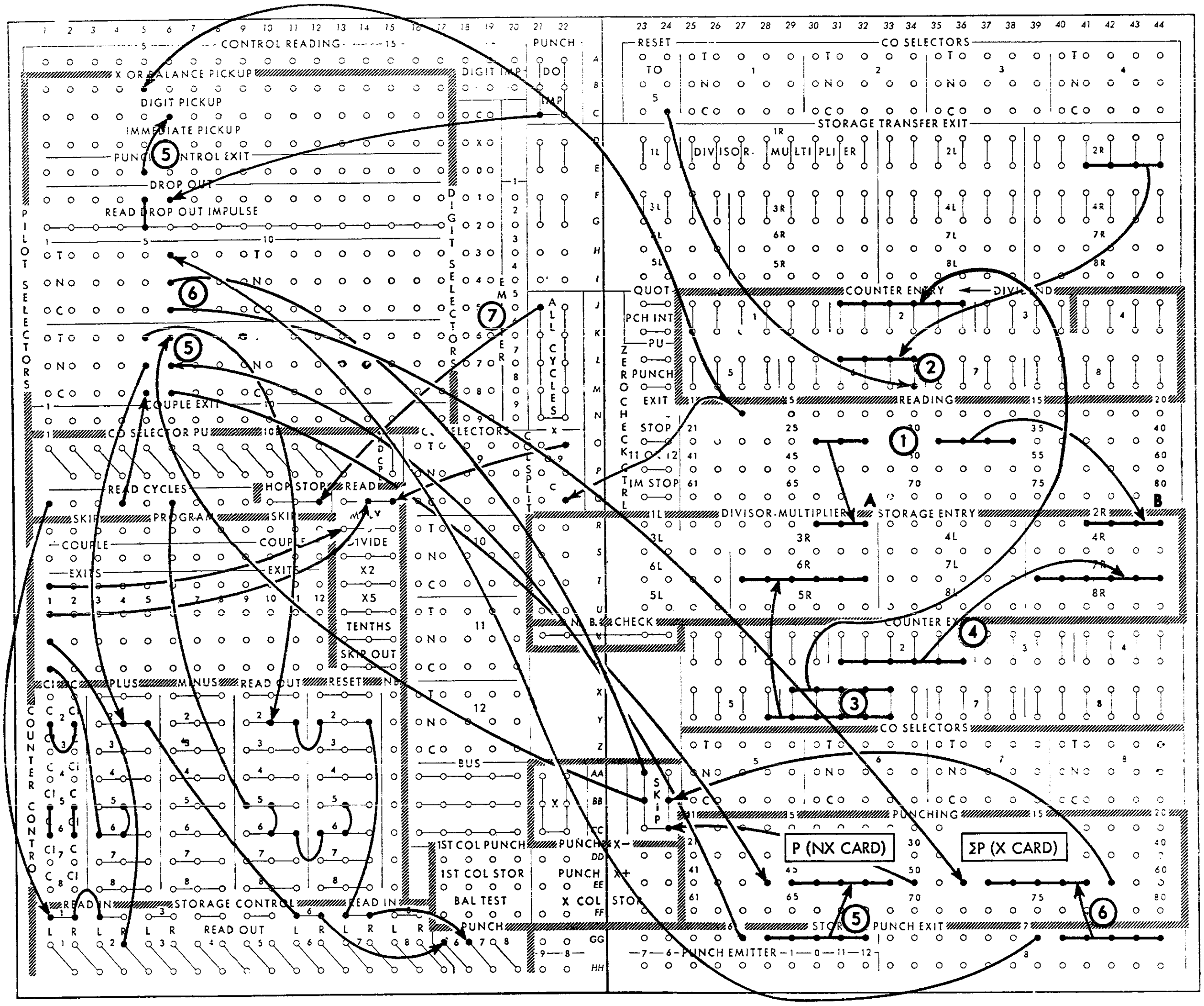


*Figure 35B. Summary Punching*
$A \times B = P;\ P_1 + P_2 = \Sigma P$

## SUMMARY PUNCHING

### $A \times B = P;\ \Sigma P \times C = R;\ P - R = S$

The previous example of summary punching may be expanded to include a calculation on the X summary card. In this problem (Figure 36) each detail card represents an invoice item which is extended and punched. The extended amounts are accumulated and the gross total is multiplied by a discount rate in the X summary card to calculate the discount amount. The discount amount is then subtracted from the gross total to obtain net invoice total. Gross, discount, and net amount are punched in the summary card; individual item amounts are punched in the detail cards. The problem may be stated as follows:

Quantity (*A*) × unit price (*B*) = Item amount (*P*)
Sum of item amts. × disc. rate (*C%*) = disc. amt. (*R*)
Sum of item amts. — disc. amt. (*R*) = net invoice amt. (*S*)

### Planning Chart and Control Wiring

*Read, NX Card.* The multiplier unit 1R is impulsed to accept quantity *A*, and storage entry 2 to accept unit price *B*.

*Program 1, NX Card.* The multiply switch is impulsed. *B* is read out of 2R on NX cards only, by wiring program 1 exit through the normal side of pilot selector 2, picked up by an X from the reading station. Counter 5-6 is impulsed to add to develop the item amount.

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT | | COUNTER | | | | | | STORAGE UNITS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | DIVR.-MULT | DIVIDEND | | | | | | | | | | | | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH | | | |
| | | | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R * | 7L | 7R * |
| | | READ CYCLE *NX Card* | | *Quantity A* | | | | | | | | *Unit Price B* | | | | | | | | |
| | 1 | *Mult.* | | | | | | | *+* | *Item Amt. +P* | | *RO* | | | | | | | | |
| | 2 | | | | *+ Sum of* | *+P P's* | | | *RO RE* | *RO RE 5(Corr)* | | | | | | | | *Item Amt P Punch ↓* | | |
| | | READ CYCLE *X Card* | | *Discount C %* | *RO* | *RO* | | | | | | | | | | | | | | *Sum of P's* |
| | 1 | *Mult* | | | *RO* | *RO* | | | *Dis. +* | *Amt. +R* | | | | | | | | | | |
| | 2 | | | | *− Net* | *−R S* | | | *RO RE* | *RO RE 5(Corr)* | | | | | | | | *Dis. Amt. R* | | *Punch ↓* |
| | 3 | | | | *RO RE* | *RO RE* | | | | | | | | | | | | *Punch ↓* | | *Net Amt. S Punch ↓* |
| | 4 | | | | | | | | | | | | | | | | | | | |

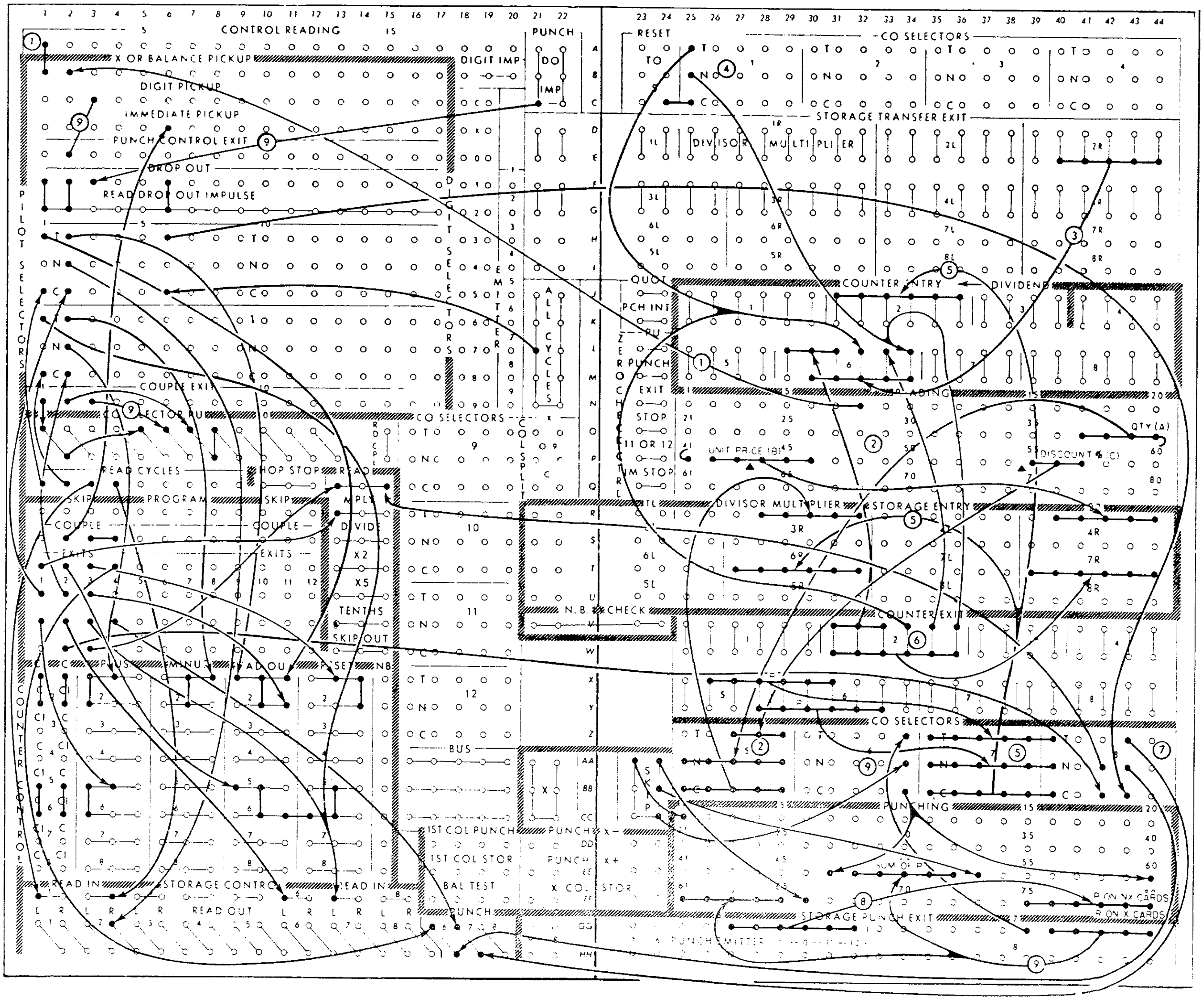


*Figure 36. Summary Punching*

$A \times B = P;\ \Sigma P \times C = R;\ \Sigma P - R = S$

*Program 2, NX Card.* The item amount *P* is read out of counter 5-6 into storage unit 6 for punching. It is also added in counter 1-2 where the sum of the item amounts for NX cards will accumulate. Counter 5-6 is wired to adjust the next result. The program 2 couple exit is wired to the immediate pickup of pilot selector 6 and an all cycles impulse is wired through the transferred side of the selector to expand program 2 exits. Storage unit 6 is impulsed to punch on program 2 on NX cards only, by wiring the selected all cycles impulse through the normal side of co-selector 8, which is controlled from an X at the reading station. READ is impulsed by wiring a program 2 exit through the normal side of co-selector 8.

*Read, X Card.* The multiplier unit is impulsed to read in *C*. The sum of the *P*'s from the preceding NX cards is read out of counter 1-2 and entered into storage unit 7. This is accomplished by wiring read cycles impulses through the upper and lower positions of pilot selector 1, picked up by an X from control reading. Storage unit 7 is impulsed to punch on program 2. If it were impulsed on the read cycle, the punching would be effective for the last NX card.

*Program 1, X Card.* The sum of the *P*'s is multiplied by the discount rate *(C)* to obtain discount amount *(R)*. This is accomplished by impulsing the multiply hub, by reading out the sum of the *P*'s in counter 1-2 through pilot selector 2, and by impulsing counter 5-6 to add.

*Program 2, X Card.* The discount amount *(R)* is read out of counter 5-6 and subtracted from the sum of the *P*'s in counter 1-2 to obtain the net *(S)*. The subtract impulse is wired through the transferred side of pilot selector 2, picked up from the reading station. At the same time, R is read into storage unit 6. Storage unit 6 is impulsed to punch discount amount on program 3, following the punching of net amount. Counter 5-6 is reset, and the reset to 5 is properly wired to adjust the item amount *(P)* on the next card. Storage unit 7 is impulsed to punch on program 2 on X cards only, through the transferred side of co-selector 8.

*Program 3, X Card.* The net *(S)* is read out of counter 1-2 into storage unit 7 for punching. Counter 1-2 is also reset. Storage unit 6 is wired to punch the discount amount. Since this is the last program used, READ is impulsed.

## Right-Hand Panel Wiring

1. Pilot selector 1 is used to select read cycles impulses and must, therefore, be picked up from control reading. Pilot selector 2 is used to select program exits and must, therefore, be picked up from the reading station. Both selectors are dropped out normally by read drop-out impulses.

2. Quantity and discount are wired to the multiplier through co-selector 5, so that quantity will enter the unit on NX cards and discount will enter the unit on the X cards. Co-selector 5 is picked up from the couple exit of pilot selector 1 and will hold up through the read cycle of the X card. Unit price is wired to storage entry 2R.

3. The multiplicand *B* (unit price) is wired out of storage exit 2 R into counter 6 entry where the product is developed.

4. Reset to 5 is wired through the normal side of co-selector 1 to the units position of counter 6 to adjust item amount on NX cards, and through the transferred side to the hundreds position of counter 6 to adjust discount amount on X cards. There are three decimals in the item amount *P*, one of which is dropped, and five decimals in the discount amount *R*, three of which are dropped.

Co-selector 1 is picked up from the couple exit of pilot selector 1 and will hold throughout the calculation of the last NX card of the group.

5. Adjusted item amount *P*, on NX cards, and adjusted discount amount *R*, on X cards, enter both counter 1-2 and storage unit 6R simultaneously. Ordinarily, direct wiring from counter exit 5-6 to counter 1-2 and storage entry 6 would suffice. The need for selection arises from the fact that one decimal is dropped when reading out *P* from counter 5-6, and three decimals when reading out *R*. *P* is selected on NX cards through the normal side of co-selectors 7 and 8, and *R* is selected on X cards through the transferred side of these same selectors. The selectors are picked up from the couple exit of pilot selector 2 controlled from the reading brush X.

6. The sum of the item amounts is wired out of counter 2 exit to storage entry 7R on the read cycle of the X card. The sum of the item amounts is also wired from counter 2 exit to counter 6 entry to develop the discount amount. The net (difference between discount amount and the sum of the item amounts) is also developed in counter 2. It is wired

out of counter 2 exit to storage entry 7R.

7. X cards require three program steps while NX cards require only two. To avoid taking program 3 on NX cards, program 2 exit is wired through the normal side of co-selector 8 to READ. Storage unit 6 must be impulsed to punch on program 2 on NX cards only, and storage unit 7 to punch on program 2 on X cards only. Therefore, a program 2 impulse (an all cycles selected on program 2) is wired through the normal and transferred sides of another position of co-selector 8 to punch 6 and 7, respectively.

8. Item amount *(P)* on NX cards and discount amount *(R)* on X cards are wired out of storage punch exit to punch in columns 75-80.

9. Both the sum of the *P*'s and *S* are wired out of storage punch exit 7, to punch the sum of the *P*'s in columns 47-52 and *S* in columns 60-65 of X cards only. These two fields are punched for X cards and skipped for NX cards by wiring the high-order position of storage punch exit 7 through the transferred side of co-selector 6, a skip impulse through the normal side, and the first column to be punched in each field to the common.

Since this is punch selection, co-selector 6 must be controlled to pick up and hold throughout the punching of X cards. This is done by wiring the couple exit of pilot selector 3 to pick up co-selector 6. Pilot selector 3 is transferred for punch selection.

# Checking Operations

## VERIFICATION OF BASIC MULTIPLICATION $(B \times A) - P - 1$

THE ANALYSIS and wiring principles for basic multiplication have been illustrated and explained in a previous problem. The same problem (Figure 37) now is reconsidered from the standpoint of verification of the punched product.

In multiplication, two fields are read from the card, one as a multiplier and the other as a multiplicand. The product is calculated by the machine and punched in a product field.

In verification of basic multiplication, all three fields *(A, B,* and *P)* must be read on the reading cycle, a second calculation of $B \times A$ must be performed, and the first product must be subtracted from the second. If the result is a positive figure, the punched product could be correct. If the result is negative, an error is indicated. Because the first test does not establish definite proof in the case of a positive result, another test is made by subtracting 1 from the result $(B \times A) - P$. The result of the second test must be minus, because if it is plus, an error is indicated. Consider these examples:

**CORRECT**

| | | |
|---|---|---|
| 2 × 2 = | +4 | **second multiplication** |
| | —4 | **punched product of 2 × 2** |
| | +0 | **first test + (right)** |
| | —1 | |
| | —1 | second test — (right) |

PUNCHED PRODUCT TOO LOW

| | | |
|---|---|---|
| 2 × 2 = | +4 | second multiplication |
| | —2 | punched product of 2 × 2 |
| | +2 | first test + (right) |
| | —1 | |
| | +1 | second test + (wrong) |

PUNCHED PRODUCT TOO HIGH

| | | |
|---|---|---|
| 2 × 2 = | +4 | second multiplication |
| | —5 | punched product of 2 × 2 |
| | —1 | first test — (wrong) |
| | —1 | |
| | —2 | second test — (wrong) |

VERIFICATION PRODUCT TOO LOW

| | | |
|---|---|---|
| 2 × 2 = | +3 | second multiplication |
| | —4 | punched product of 2 × 2 |
| | —1 | first test — (wrong) |
| | —1 | |
| | —2 | second test — (wrong) |

VERIFICATION PRODUCT TOO HIGH

| | | |
|---|---|---|
| 2 × 2 = | +5 | second multiplication |
| | —4 | punched product of 2 × 2 |
| | +1 | first test + (right) |
| | —1 | |
| | +0 | second test + (wrong) |

*The rule may be stated as follows:*
*Plus on the first test, minus on the second test — the answer is right.*
*Plus on the first test, plus on the second test — the answer is wrong.*
*Minus on the first test — the answer is wrong.*

Note that the punched product is always correct if the first test is plus and the second test minus. It is wrong if both tests are minus or if both tests are plus. It is not possible to have a minus test followed by a plus test.

The machine can be controlled to indicate an error card by one of the three methods: first, to punch in any column of the card a 12 for correct cards and an X for error cards; second, to eject the error card into the stacker and stop the machine; or, third, to stop the machine while the error card is still in the card bed. When the machine stops for error cards, the comparing light signals the error.

*Zero Check Control and NB Check (L-Q, 23-24; V, 21-24)*. These are always used together and have the function of checking multiplying, dividing, or crossfooting operations either on the same run or on a second run of the cards through the machine. Most verifying operations are done on a second run because the punched result as well as the calculated result can be checked.

The PU (pickup) hubs of the zero check control unit are normally wired from the program exit that controls the second test. When this unit is so controlled, it will do one of three things according to the way it is wired: punch a signal for correct and incorrect cards, stop after the error card is in the stacker, or stop immediately upon detection of an error.

The unit may be used only once for each card. To verify more than one answer in a card, if the answers are independent, selectors may be used to recognize the plus-minus sequence of signs and to control the punching of a verification signal. The zero check control unit can be used for the last answer verified.

If the card is to be punched, the punch hubs must be impulsed from the program on which the test for error is made, which makes available out of the 11 or 12 hubs an 11 punch for error cards and a 12 punch for correct cards. The 11 or 12 exit is wired directly to the punching column.

If the machine is to be stopped, one of the common exit hubs is wired to STOP. When the exit is wired to STOP, the machine will stop after the error card is punched and stacked, and the error signal will light. Before the machine can be started again, the reset key must be depressed. IM STOP cannot be wired if the machine is instructed to punch the X or 12 code.

The stop hubs may be used to stop the machine even though the zero check control unit is not used. For example, a program exit wired to IM STOP will stop the machine on that particular stop.

The NB check unit *(V, 21-24)* is normally controlled from the counter NB hubs. If both NB check and zero check control pickup are impulsed at the same time, no error is indicated. If zero check pickup and NB check fail to receive an impulse, no error is indicated. If either of them receives an impulse and the other does not, an error is indicated.

Two tests are made for every card. The method of wiring each test depends upon the type of calculation being performed. In the case of multiplication, a minus on the first test indicates an error. A plus on the first test may or may not be wrong, so a second test is made by subtracting a one from the result. If the new result is plus, an error is indicated. If it is negative on the second test and was not negative on the first test, it is correct. In the case of division, the first and second test must have opposite signs, as explained in detail under "Division Checking."

## Planning Chart and Control Wiring

*Read*. *B* is read into the multiplier, *A* into storage unit 2R and the punched product *($P_1$)* into storage unit 3. In verification of multiplication, it is good practice to reverse the multiplier and the multiplicand.

*Program 1*. The multiply switch is impulsed, *A* is read out of 2R and the product is developed in counter 5-6. This counter now contains the product of the second calculation.

*Program 2*. $P_1$ is subtracted from $P_2$. At the end of program 2 counter 5-6 contains the difference between the products of the first and second calculations. If the result is negative an error is indicated.

*Program 3*. Although the result may be plus on the first test, it could still be wrong, as in the case of (2 × 2) —3. A second test is made by subtracting one from the units position of the punched product. If the new result is plus, an error is indicated. Program 3 is wired to READ, to stop further programming. The 11 or 12 is punched in the card on this cycle.

*Read Cycle*. Counter 5-6 is reset.

## Right-Hand Panel Wiring

1. *A* is entered into storage unit 2R, *B* into the multiplier, and *P* into 3L-3R.

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT | | COUNTER | | | | | | STORAGE UNITS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | DIVR.-MULT. | DIVIDEND | | | | | | | | | | | | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH | | | |
| | | | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R | 7L | 7R |
| | | READ CYCLE | | B | | | | | RE | RE(5) | | A | $P_1$ | $P_1$ | | | | | | |
| | 1 | Mult. | | | | | | | + | +(BxA) | | RO | | | | | | | | |
| | 2 | 1st Test | | | | | | | − | −$P_1$ | | | RO | RO | | | | | | |
| | 3 | 2nd Test<br>Emit 1<br>Punch<br>11 or 12 | | | | | | | − | − 1 | | | | | | | | | | |
| | | Read Cycle | | | | | | | RE | RE(5) | | | | | | | | | | |

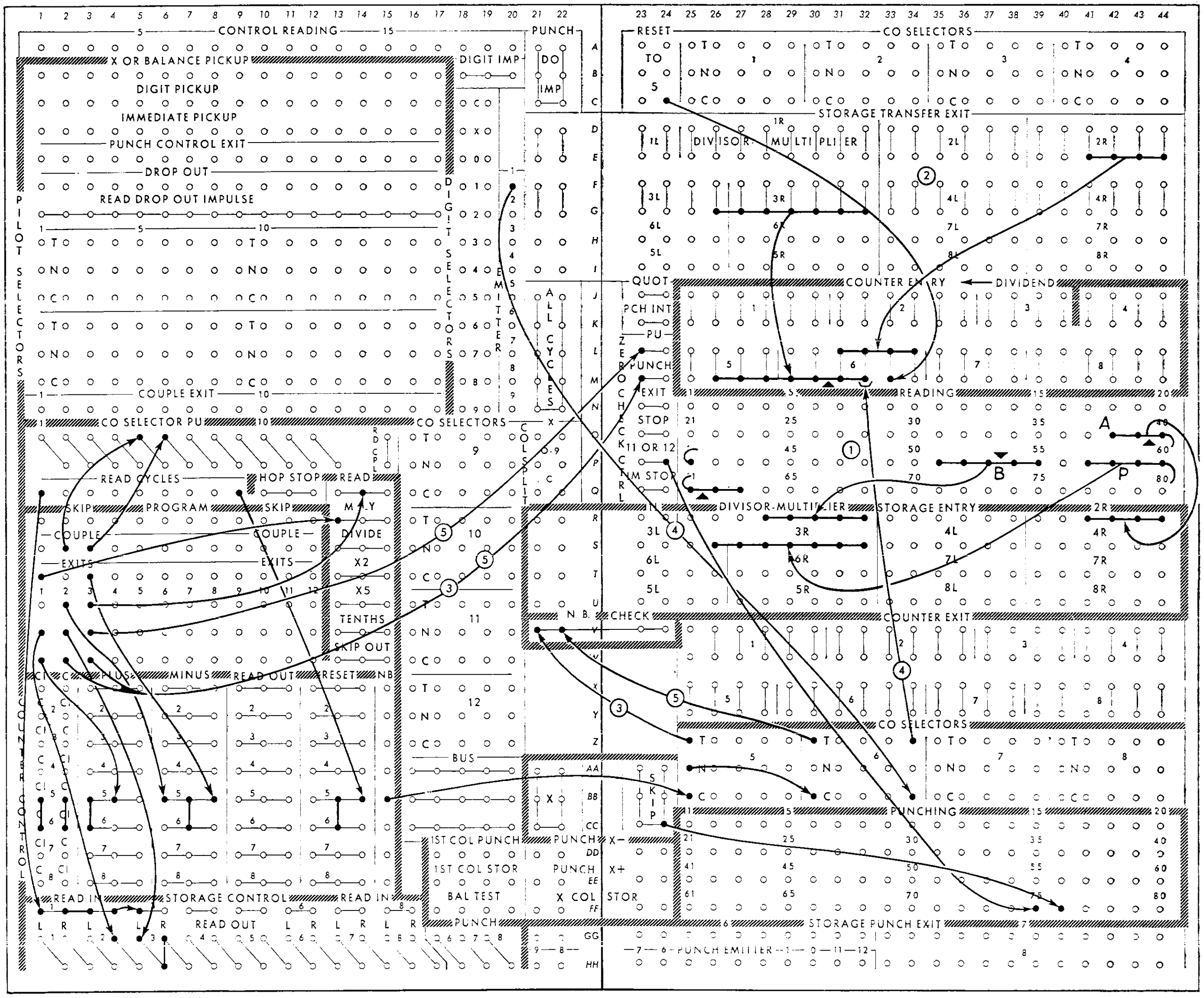


*Figure 37. Verification of Basic Multiplication*
*(B × A) — P — 1*

2. *A* is wired out of 2R and entered into counter 5-6 to calculate the result of *B* × *A* on program 1. *P* is wired out of 3L-3R and entered into counter 5-6 where it is subtracted on program 2. It is offset two positions to line up with the decimal in the product of *B* × *A*. Reset to 5 is wired to the tens position of the counter.

3. The NB of counter 5 (high-order counter in the group) is wired to NB CHECK through the transferred side of co-selector 5, picked up on program 2. If the result of *(B × A)* — *P* is negative on program 2, it is wrong. NB CHECK will, therefore, be energized, but the zero check control unit will not, because program 2 is not wired to its pickup. This indicates an error on the first test. The unequal condition between the two test units, together with the impulse received at PUNCH, sets up a condition in the zero check control that will not be disturbed by the second test, and an X will punch in the card.

If the balance is positive on the first test, NB CHECK would not receive an impulse. Since the zero check control unit is not picked up, further checking is delayed until the second test one cycle later.

4. The digit 1 from the emitter, controlled through co-selector 6, is subtracted on program 3 in the hundreds position of the counter, representing the units position of the punched result as shown below:

| ORIGINAL | | VERIFICATION | | |
|---|---|---|---|---|
| Reset to 5 | 5 | Reset to 5 | | 5 |
| .5 × .135 | = .0675 | .5 × .135 | + | .0675 |
| Punched result | .07 | | | .0725 |
| | | Punched result | — | .07 |
| | | First Test | + | .0025 |
| | | | — | 1 |
| | | Second Test | | 999925 (right) |

If the 1 in the above example had been subtracted from either of the dropped decimal positions, it would have resulted in a positive balance. It must be subtracted from the units position of the punched result.

5. The NB of counter 5 is wired to NB CHECK through the transferred side of co-selector 6, picked up on program 3. If, on program 3, the result of *(B × A)* —*P* —1 is negative, NB CHECK is impulsed and, since the zero check control (PU and PUNCH) is impulsed on the same cycle, the card is treated as correct and is punched with a 12. If the result is positive, NB is not impulsed, but the zero check control is. The 11 or 12 hubs will emit an X impulse.

If the machine is to be stopped for error cards, the wires into PUNCH and out of 11 and 12 would be replaced by a wire from exit to STOP, or from exit to IM STOP.

## VERIFICATION OF BASIC DIVISION
## $-(A \times B) + A \pm (.5 \times B)$

BASIC division is checked by making two tests (Figure 38). The first test is made after the dividend is added to the minus product of the quotient times the divisor. This difference may be plus or minus. The second test consists of subtracting half the divisor if the result is positive, or adding half the divisor if the result is negative. If both tests are negative or if both tests are positive, the result is wrong as shown in the following examples.

| ORIGINAL | CHECK | |
|---|---|---|
| 2 (right) | —( 2 × 3) = —6 | quotient times divisor |
| 3√6 | +6 | dividend |
| | +0 | first test |
| | —(.5 × 3) = —1.5 | half the divisor |
| | —1.5 | second test (right) |

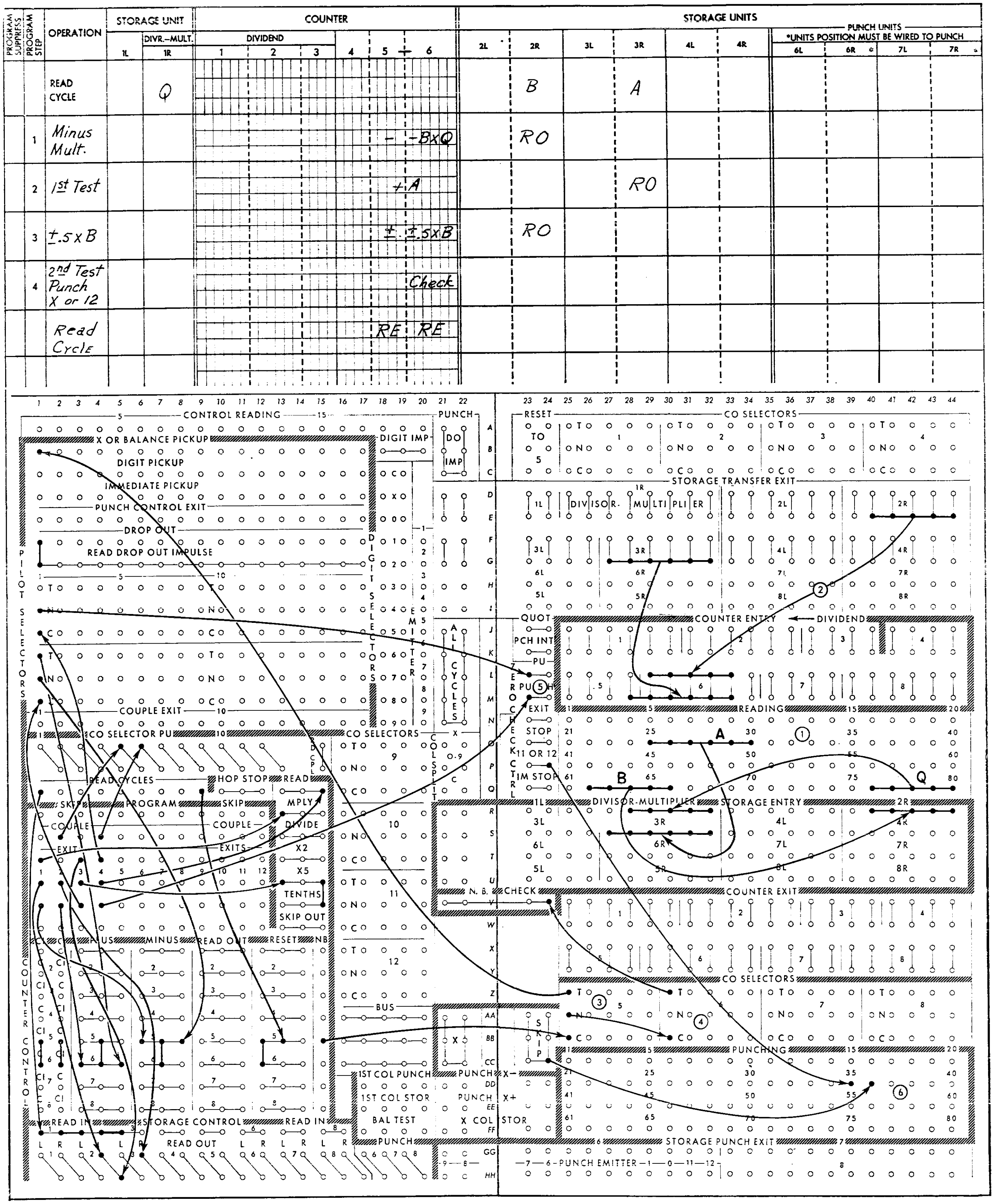
| ORIGINAL | CHECK | |
|---|---|---|
| 2.5 or 3 (right) | —( 3 × 2) = —6 | quotient times divisor |
| 2√5 | +5 | dividend |
| | —1 | first test |
| | +(.5 × 2) = +1 | half the divisor |
| | +0 | second test (right) |

| ORIGINAL | CHECK | |
|---|---|---|
| 3 (wrong) | —( 3 × 3) = —9 | quotient times divisor |
| 3√6 | +6 | dividend |
| | —3 | first test |
| | +(.5 × 3) = +1.5 | half the divisor |
| | —1.5 | second test (wrong) |

| ORIGINAL | CHECK | |
|---|---|---|
| 1 (wrong) | —( 1 × 3) = —3 | quotient times divisor |
| 3√6 | +6 | dividened |
| | +3 | first test |
| | —(.5 × 3) = —1.5 | half the divisor |
| | +1.5 | second test (wrong) |

The rule may be stated as follows:

*Unlike signs on the two tests, the answer is right.*
*Like signs on the two tests, the answer is wrong.*



| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT 1R DIVR.-MULT. | COUNTER DIVIDEND 1 | 2 | 3 | 4 | 5 | 6 | STORAGE UNITS 2L | 2R | 3L | 3R | 4L | 4R | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH 6L | 6R * | 7L | 7R * |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | READ CYCLE | | Q | | | | | | | | B | | A | | | | | | |
| | 1 | Minus Mult. | | | | | | | – | -BxQ | | RO | | | | | | | | |
| | 2 | 1st Test | | | | | | | +A | | | | | RO | | | | | | |
| | 3 | ±.5xB | | | | | | | ± | ±.5xB | | RO | | | | | | | | |
| | 4 | 2nd Test Punch X or 12 | | | | | | | | Check | | | | | | | | | | |
| | | Read Cycle | | | | | | | RE | RE | | | | | | | | | | |
| | | | | | | | | | | | | | | | | | | | | |

*Figure 38. Verification of Basic Division*
*— (Q × B) + A ± (.5 × B)*

*×2; ×5; Tenths (T, 13-15) (U, 13-15).* Multiplication by 2 or by 5 may be done by wiring a program exit to the ×2 hubs or to the ×5 hubs and by reading out the multiplicand in the usual manner. The multiply hub must not be impulsed for this type of multiplication. The tenths hubs are standard only on dividing machines. They can be impulsed from a program exit, only in conjunction with ×5 or ×2 multiplication, to cause the product to shift one position to the right. They are most frequently used with ×5 multiplication as a quick method of obtaining half of some factor, since .5 times a factor is the same as dividing by 2.

## Planning Chart and Control Wiring

*Read.* The quotient *Q* is entered into the multiplier unit, the divisor *B* into storage unit 2R, and the dividend *A* into storage unit 3R.

*Program 1.* B is minus-multiplied by Q, by reading *B* out of 2R, subtracting in counter 5-6, and impulsing the multiply switch. The minus product is developed in counter 5-6.

*Program 2. A* is read out of 3R and added to the minus product of *Q* × *B*. The result may be plus or minus, and a determination of its sign is made on this cycle. Co-selector 5 is picked up from the couple exit of program 2. NB of counter 5 is wired to the common of co-selector 5 and out of the transferred hub to pick up pilot selector 1. If the result in counter 5 is negative at the end of program 2, pilot selector 1 is transferred and holds through the following read cycle.

*Program 3.* At the completion of program 2, the balance in counter 5-6 could be either plus or minus. If the balance is plus, half the divisor should be subtracted from it, and if it is minus, half the divisor should be added to it.

One-half the divisor is obtained by wiring program 3 to the ×5 hubs and also to the tenths hubs, resulting in a multiplication by .5. Program 3 is also wired to the common of pilot selector 1, picked up by NB of counter 5 on program 2. It is also wired from the normal hub of that selector to subtract in counter 5-6 and from the transferred hub to add in counter 5-6. Thus, if the balance standing in counter 5-6 at the end of program 2 is positive, half of the divisor is subtracted from it, and if the balance is negative, half of the divisor is added to it.

The balance at the end of this cycle will be tested on the next program since testing cannot be done on a program used for multiplication or division.

*Program 4.* Co-selector 6 is picked up from the couple exit of program 4. NB of counter 5 (normal of co-selector 5) is wired through the transferred hub of co-selector 6 to NB CHECK. Program 4 is also wired through the normal hub of pilot selector 1 to zero check control pickup. If both the first and second tests show positive results, NB CHECK receives no impulse. The zero check control unit receives an impulse from program 4 on the second test through the normal hub of selector 1. Whenever one of these units is controlled and the other is not, an error condition is signaled and an X (11) will punch in the card. If both the first and second tests show negative results, the zero check control unit does not receive an impulse because program 4 cannot get through pilot selector 1. The NB CHECK receives an impulse from the NB counter 5 through the normal hub of co-selector 5 and the transferred hub of co-selector 6. In this instance, NB CHECK receives an impulse and zero check control does not. Therefore, an error is indicated. These conditions represent a plus-to-plus, or a minus-to-minus, test, either of which shows the card to be wrong.

If the result of the first test is plus, program 4 picks up the zero check unit. If the result of the second test is minus, the NB of counter 5 energizes the NB CHECK. Both units are controlled, indicating a correct result, and a 12 will be emitted from the 11 or 12 hubs. If the result of the first test is minus, the zero check unit is not controlled. If the result of the second test is plus, NB CHECK is not impulsed. In this instance, neither unit is controlled, indicating that the card is correct. These conditions represent a plus-to-minus, or minus-to-plus, test, either of which shows the card to be correct.

Program 4 impulses the punch hubs so that the 11 or 12 may be punched in the card. The machine can be made to stop for all errors by substituting for PUNCH and 11 or 12 a wire from EXIT to STOP or to IM STOP. Program 4 is wired to read to stop further programming.

*Read Cycle.* Counter 5-6 is reset.

## Right-Hand Panel Wiring

1. The dividend *A* is wired to storage unit 3R. The divisor *B* is wired to storage unit 2R, and *Q*, the quotient, is wired to the multiplier.

2. *B* is read out of 2R into counter 6, offset one position, where it is minus-multiplied by *Q*, and plus- or minus-multiplied by .5. If *B* were not offset, the decimal resulting from multiplying the divisor by .5 would be ignored.

3. *A* is wired from storage unit 3 to counter entries 5-6, offset one position. Co-selector 5 is picked up on program 2. NB of counter 5 is wired through the transferred hub of co-selector 5 to pick up pilot selector 1. This selector is used to select program 3 to the plus or minus hubs of counter 5-6, depending upon the balance in the counter after the first test; also to select program 4 to the pickup of the zero check control unit.

4. Co-selector 6 is picked up on program 4. NB of counter 5 is wired from the normal of co-selector 5 through the transferred side of co-selector 6 to energize NB check.

5. Zero check control is picked up only if the first test is positive by wiring program 4 through the normal side of pilot selector 1. The punch hub is wired directly from program 4, since either an 11 or a 12 must always be punched.

6. The 11 or 12 hub is wired to column 35. All correct cards will be punched with a 12 and all error cards with an X.

## Alternate Method

The formula $-(Q \times B) + A - B$ is required for verification of basic division when the punched quotient was not half adjusted during the first run. The planning chart and wiring diagram are essentially the same except that on program 3 the whole divisor *(B)* is always subtracted instead of $\pm .5 \times B$. Error conditions are recognized on each test as follows:

*Minus on the first test; or plus on the first test and plus on the second test.*

When the divisor is zero and, consequently, the quotient is, the rule for testing would not apply, since both tests would indicate a plus result as follows:

| ORIGINAL | CHECK | |
|---|---|---|
| 0 (right) | $-(0 \times 0) = 0$ | |
| $0\sqrt{6}$ | $+6$ | dividend |
| | $+6$ | first test |
| | $-(.5 \times B) = -0$ | half the divisor |
| | $+6$ | second test (wrong) |

It may be seen from the above example that a zero divisor and a correct zero quotient would be signaled as an error. In all such cases, the wiring for any of the formulas given above would, therefore, stop the machine or punch an X. This error condition may be eliminated by subtracting the divisor and the quotient in any available counter and by testing the counter for a negative balance. If the counter is negative, indicating either a significant divisor, a significant quotient, or both, the regular checking circuit is allowed to function through the transferred side of a pilot selector. If the counter is positive (a zero balance is positive), the pilot selector is normal, and the regular checking circuit is eliminated.

A planning chart and the selector wiring are illustrated in Figure 39.

1. NB of counter 2 is wired to the balance pickup of pilot selector 4 through the transferred side of co-selector 5. If counter 2 is negative on program 2, pilot selector 4 will transfer on the next machine cycle and remain transferred for the remainder of the problem.

2. Program 4 is tested through the transferred side of pilot selector 4 and the normal side of pilot selector 1 before reaching the pickup of zero check control. If significant digits are punched in either the quotient or the divisor, program exit 4 will pass through the transferred side of pilot selector 4 and normal division checking will take place. If zeros are punched in the quotient and the divisor, pilot selector 4 will be normal and program 4 exit will be eliminated. Since NB check will not be impulsed (plus on second test), both zero check control and NB check will be normal and the result is proved correct.

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT DIVR.-MULT. 1R | COUNTER DIVIDEND 1 | COUNTER DIVIDEND 2 | COUNTER DIVIDEND 3 | COUNTER 4 | COUNTER 5 | COUNTER 6 | STORAGE UNITS 2L | 2R | 3L | 3R | 4L | 4R | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH 6L | 6R | 7L | 7R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | READ CYCLE | | Q | | −Q | | | RE | RE | | B | | A | | | | | | |
| | 1 | Minus Mult. | | | | | | | | −(Q×B) | | RO | | | | | | | | |
| | 2 | 1st Test ctrs. 2&5 | | | | −B | | | +A | | | RO | | RO | | | | | | |
| | 3 | ± .5×B | | | | | | | ± | (5×B) | | RO | | | | | | | | |
| | 4 | 2nd Test ctr. 5 Punch 11 or 12 | | | | RE | | | CHECK | | | | | | | | | | | |
| | 5 | | | | | | | | | | | | | | | | | | | |

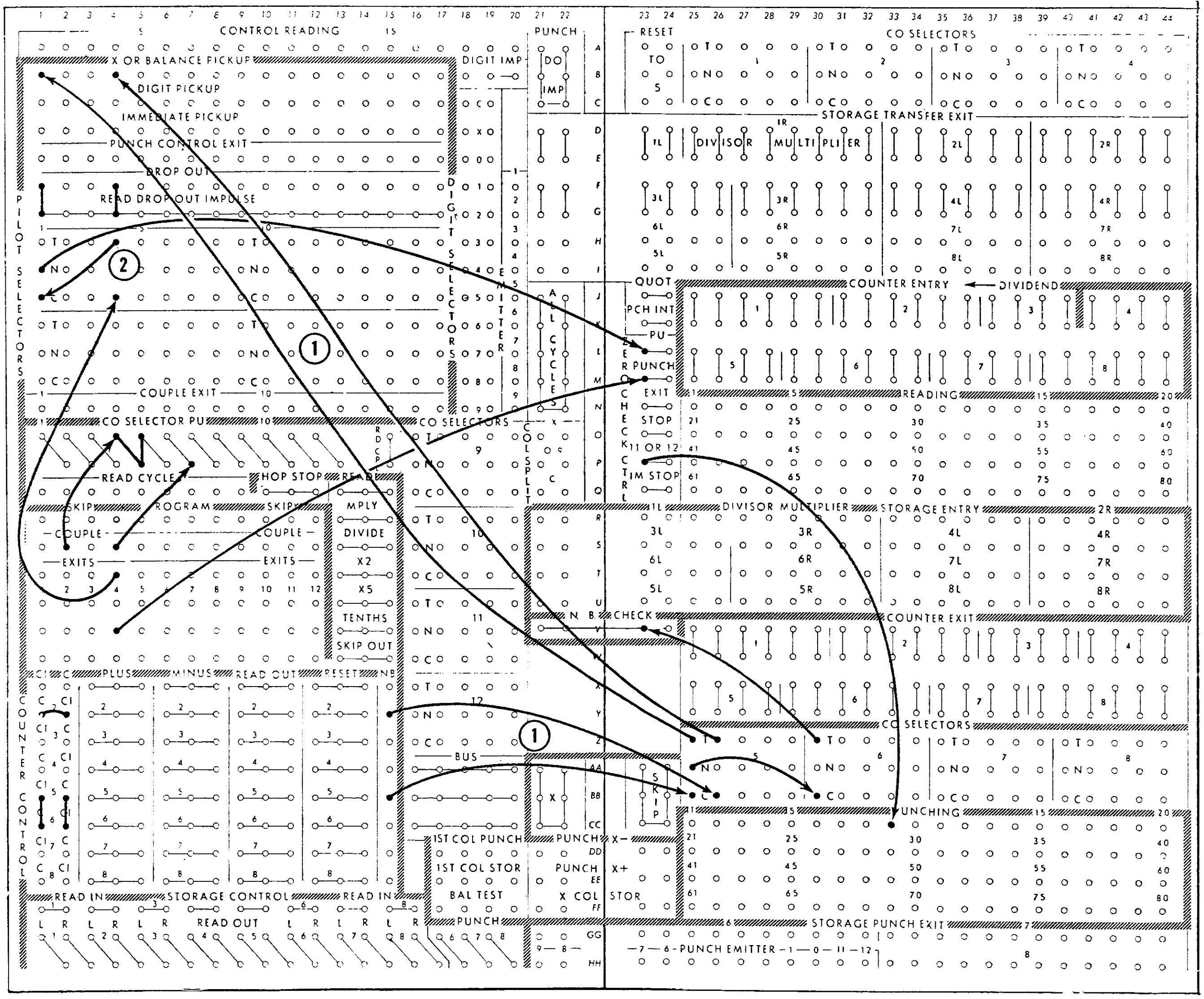


*Figure 39. Verification of Division, Divisor Zero*

# Typical Applications

## SUCCESSIVE CALCULATION, PUNCHING INTERMEDIATE RESULTS

$A \times B = P; P \div C = R$

THE 602 may perform many successive multiplying or dividing operations or combinations of both. This problem (Figure 40) demonstrates two such successive calculations, involving both multiplication and division. The product of the first calculation is punched and used as the dividend for the succeeding operation, the quotient of which is also punched.

In series operations of this type, punching may be intermediate; that is, punching of a result may begin as soon as it is obtained, or any time during the remaining part of the problem. This has the advantage of freeing counters for other uses, thus increasing the calculating capacity of the machine considerably. This type of punching also enables the machine to punch beyond the 24-position capacity of the two punch storage units, for, once they have finished punching one set of results, they are ready to accept another set. Theoretically, all 80 columns of the card could be punched in this manner, if the factors to be calculated were entered from a preceding card.

Punching need not be intermediate if the result to be punched does not exceed 24 columns (punch storage capacity) for then it may take place at the conclusion of the problem. Counter capacity, card design and speed are the major considerations when deciding which of the two methods is to be used. Usually, when a large portion of the card is to be punched, several results are represented, each of which may be obtained successively. Punching results as they occur, in most cases, can be done during the cycles for a succeeding calculation. Whichever method is used, punching should be so controlled that little or no time is lost. With the knowledge that the machine can punch four columns for every cycle taken, and that punching of one card may take place during the calculation of another, a proper determination can be made as to which method is the faster.

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT DIVR.-MULT. 1R | COUNTER DIVIDEND 1 | COUNTER DIVIDEND 2 | COUNTER DIVIDEND 3 | COUNTER 4 | COUNTER 5 | COUNTER 6 | STORAGE UNITS 2L | STORAGE UNITS 2R | STORAGE UNITS 3L | STORAGE UNITS 3R | STORAGE UNITS 4L | STORAGE UNITS 4R | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH 6L | 6R * | 7L | 7R * |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | READ CYCLE |  | A |  |  |  |  |  |  |  | B |  | C |  |  |  |  |  |  |
|  | 1 | Mult. |  |  |  |  |  |  | + | +P |  | RO |  |  |  |  |  |  |  |  |
|  | 2 | Program Exits 2 & 3 |  | C | + | + | +P |  | RO<br>RE | RO<br>RE ⑤ |  |  |  | RO |  |  |  | P<br>Punch |  |  |
|  | 3 | Divide Program Exits 4 & 5 |  | RO | − | − | − |  | + | +R |  |  |  |  |  |  |  |  |  |  |
|  |  | Read Cycle |  |  | RE | RE | RE |  | RO<br>RE | RO<br>RE ⑤ |  |  |  |  |  |  |  |  |  | R<br>Punch |

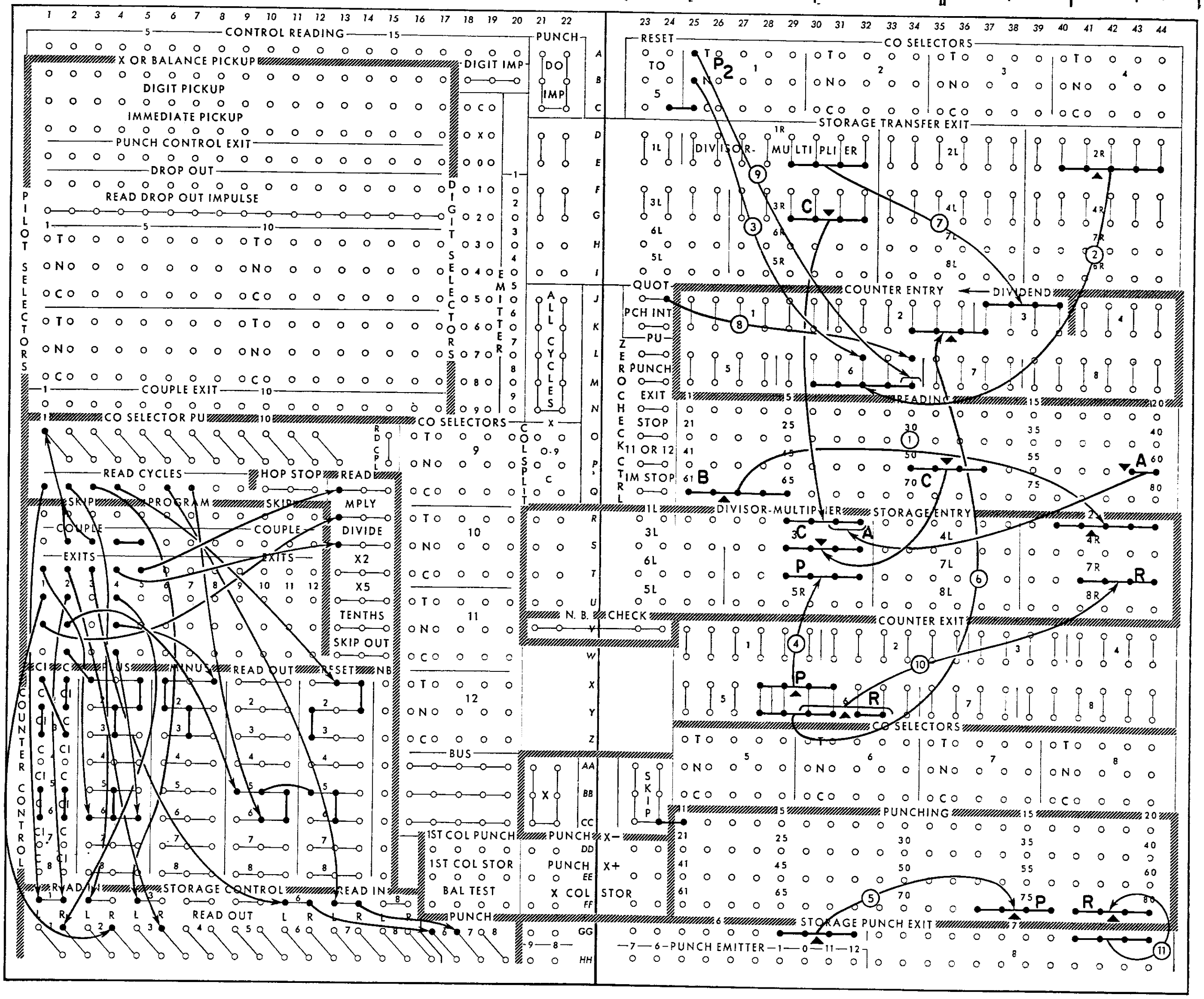


Figure 40. Successive Calculation: Punching Intermediate Results

$A \times B = P; P \div C = R$

### Planning Chart and Control Wiring

*Read.* *A* is read into the multiplier, *B* is stored in storage unit 2, and *C* is stored in storage unit 3.

*Program 1.* *B* is read out of 2R and multiplied by *A*. The product is developed in counter 5-6 which is impulsed to add.

*Program 2 (Exits 2 and 3).* *C* is read out of storage unit 3 and entered into the divisor unit 1R. The product of *A* × *B* is read out of counter 5-6 and entered into storage unit 6R to start punching immediately. At the same time, the product is also added into the dividend counter 1-2-3 in preparation for the division operation. Counters 1, 2 and 3 are coupled to act as one group. Counter 5-6 is reset during this program, with the proper position in that group being reset to 5 for half-adjustment of the quotient. The exits for program 2 are expanded by coupling program 2 with program 3. Co-selector 1 is picked up on this program in order to select the reset-to-5 impulse.

*Program 3 (Exits 4 and 5).* Program 3 is expanded by coupling step 4 with step 5. The divide hub is impulsed, the divisor *C* is read out of 1R, and the product of the divisor times each quotient digit is subtracted in the dividend counters. The quotient is added in counter 5-6. Because program 3 is the last program used in the problem, it is wired to READ.

*Read.* All counters can be cleared on the read cycle. The quotient in counter 5-6 is read out and entered in storage unit 7 for punching. The proper position of counter 6 is wired to reset to 5 for half-adjustment of the product.

Punching in this problem is intermediate. The product of *A* × *B* is punched as soon as it is available. While it is punching, the machine is dividing *P* by *C*. Punching of the quotient takes place during the multiplication of the following card. Note that counter 5-6 is used to develop first the product and later the quotient. This action is made possible by the successive punch feature of the machine, allowing the same counter to be used repeatedly to develop different results at different times.

Both *P* and *R* could have been punched from the same storage punch exit, in this problem, without loss of time for punching. Because division takes place on program 3, several cycles occur, allowing ample time for punching *P* before *R* is to be read into the unit. By the same reasoning, program 1 is made up of several multiply cycles, allowing ample time for *R* to punch before *P* of the next card is read into the unit.

### Right-Hand Panel Wiring

1. *A* is wired to the multiplier, *B* to storage unit 2R, and *C* to storage unit 3R.

2. *B* is read out of 2R into counter group 5-6, where the product develops.

3. Of the five decimals obtained in the product *P*, only two are needed. The high-order position dropped will be reset to 5 when the counters reset. The quotient *R* will also be half-adjusted, but in a different position of the same counter. The reset to 5 is selected through co-selector 1 so that the third position of counter 6 will reset to 5 for the product *P* and the first position of counter 6 will reset to 5 for the quotient *R*. Co-selector 1 is transferred on program 2, so the selector will be normal when the counters reset on the read cycle.

4. The product of *A* × *B*, with three positions dropped, is wired to storage unit 6R for punching.

5. The product is punched as soon as it is available from storage unit 6.

6. Two decimals are desired in the quotient. Two decimals in the divisor, when added to the extra position necessary for half-adjusting the quotient, places the decimal point in the fifth position of the dividend. Factor *P*, with only two decimals, is wired to the dividend counter to line up with the decimal point.

7. The divisor is read out of 1R and entered into the dividend counter without regard to decimal position.

8. The quotient is developed in counter 6, so the QUOT is wired to the units position of that counter.

9. Reset to 5 for the quotient is wired to the units position of counter 6 through the transferred side of co-selector 1, which is transferred when the counters reset on program 2.

10. Quotient *R* in counter 6, with one position dropped, is read into storage unit 7R for punching.

11. The quotient is punched from storage unit 7.

## SIMULTANEOUS MULTIPLICATION, ONE MULTIPLIER AND SEVERAL MULTIPLICANDS

$A \times B = P_1; A \times C = P_2; A \times D = P_3$

SEVERAL multiplicands can be multiplied by one multiplier to develop as many products simultaneously as the counter capacity will allow. In this problem (Figure 41), three 6-position multiplicands are multiplied by a 4-position multiplier to develop three 10-position products.

### Planning Chart and Control Wiring

*Read.* The multiplier $A$ is entered into the multiplier counter and multiplicands $B$, $C$, and $D$ into storage units 2, 3, and 4, respectively.

*Program 1.* Factors $B$, $C$, and $D$ are read out of storage units 2R, 3R, and 4R into the counters in such a way that $P_1$ will develop in counter 1 and part of counter 2, $P_2$ will develop in counters 4, 3, and part of counter 2, and $P_3$ will develop in counter 5-6. Counters 1, 2, 3, and 4, are coupled to act together as a single unit to develop products $P_1$ and $P_2$. Counters 5 and 6 are coupled to develop $P_3$. The multiplier switch is impulsed.

*Program 2.* Counter 1-2-3-4 reads out products $P_1$ and $P_2$ and enters them into storage units 6 and 7 for punching. This counter is also reset at this time. Although $P_3$ in counter 5-6 is ready for punching at this time, 20 of the 24 storage punch positions are needed for $P_1$ and $P_2$. Therefore, $P_3$ will be transferred to storage unit 6 on the next cycle. Since program 2 is the last program used, it is wired to READ.

*Read.* The product of $A \times D$ ($P_3$) is read out of counter 5-6 into storage unit 6 for punching. These counters are also reset at this time. It is on this cycle that $P_1$ starts to punch. Punching is almost continuous in this problem. There is a slight delay in reading $P_3$ into storage unit 6 because of the 10-column field to be punched for $P_1$ which requires more than one cycle. An internal interlock holds the reading of $P_3$ until the units digit of 6R is punched. The punching of $P_2$ from storage unit 7 will start during the read cycle of the following card and continue during the multiplication. The punching of $P_3$ from storage unit 6 will start and be completed during the multiplication of the following card. No loss of time will result from punching $P_2$ and $P_3$.

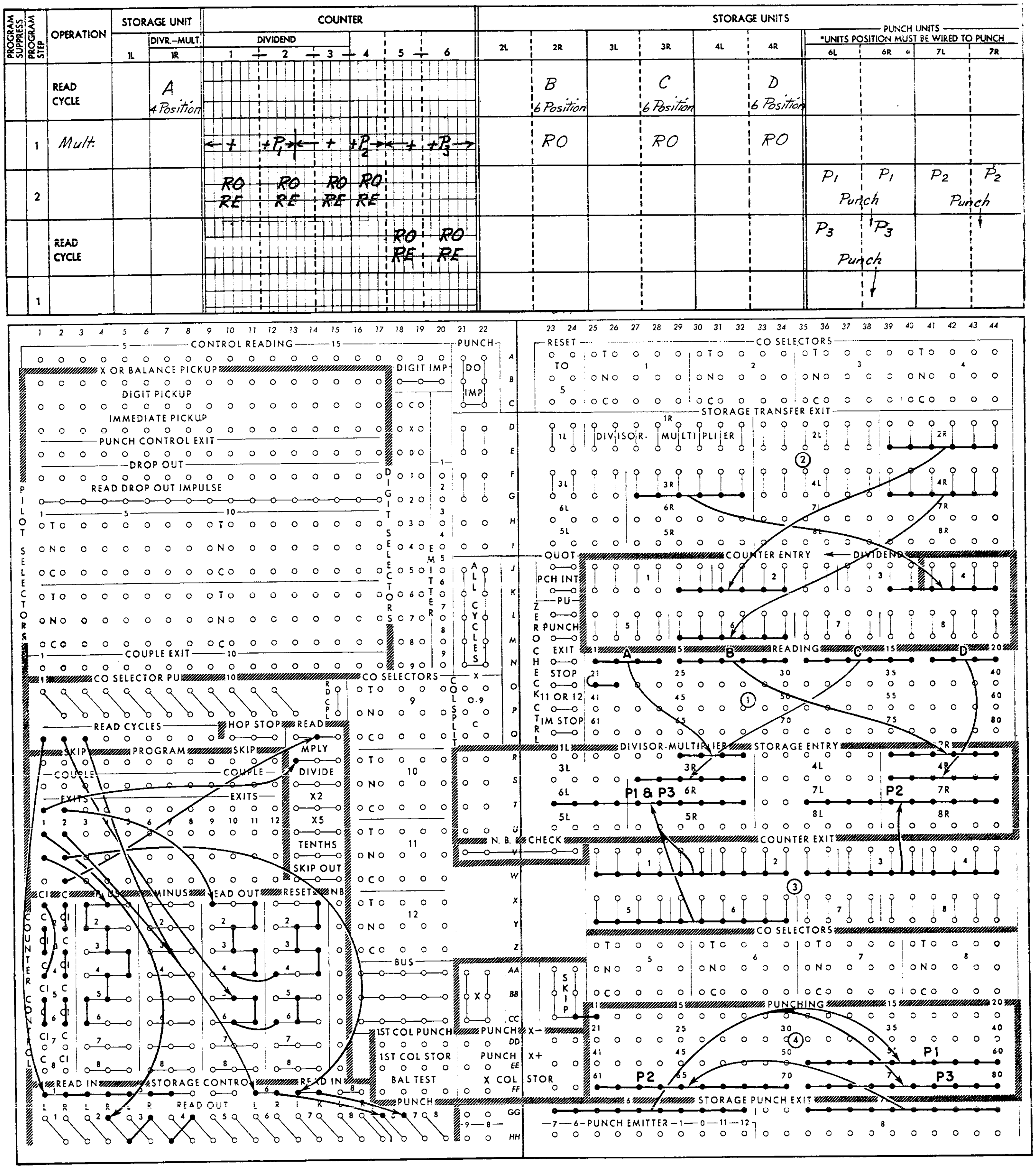
### Right-Hand Panel Wiring

1. The multiplier $A$ is entered into 1R. The multiplicands $B$, $C$, and $D$ are entered into storage units 2, 3, and 4.

2. Storage exits 2R, 3R, and 4R are wired to the counters in such a way that the product of $A \times B$ ($P_1$) will develop in the first ten positions of the counter, the product of $A \times C$ ($P_2$) will develop in the next ten positions, and the product of $A \times D$ ($P_3$) will develop in the last ten positions.

3. The products $P_1$ and $P_3$ are wired to storage unit 6 for punching. The product $P_2$ is wired to storage unit 7.

4. Three 10-position products are punched from storage punch exits 6 and 7.



| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT DIVR.-MULT. 1R | COUNTER DIVIDEND 1 | 2 | 3 | 4 | 5 | 6 | STORAGE UNITS 2L | 2R | 3L | 3R | 4L | 4R | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH 6L | 6R | 7L | 7R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | READ CYCLE |  | A<br>4 Position |  |  |  |  |  |  |  | B<br>6 Position |  | C<br>6 Position |  | D<br>6 Position |  |  |  |  |
|  | 1 | Mult |  |  | ← + | +P1 → | ← + | +P2 → | ← + | +P3 → |  | RO |  | RO |  | RO |  |  |  |  |
|  | 2 |  |  |  | RO<br>RE | RO<br>RE | RO<br>RE | RO<br>RE |  |  |  |  |  |  |  |  | P1 | P1<br>Punch | P2 | P2<br>Punch |
|  |  | READ CYCLE |  |  |  |  |  |  | RO<br>RE | RO<br>RE |  |  |  |  |  |  | P3 | P3<br>Punch |  |  |
|  | 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

*Figure 41. Simultaneous Multiplication: One Multiplier, Several Multiplicands*
$A \times B = P_1;\ A \times C = P_2;\ A \times D = P_3$

## SUCCESSIVE CALCULATION, INDEPENDENT FACTORS AND PRODUCTS

$$A \times B = P_1; C \times D = P_2; E \times F = P_3$$
$$G \times H = P_4$$

In a single run of the cards through the machine the 602 may perform a series of different multiplication operations, each with a different multiplier and multiplicand; a series of different division operations each with a different dividend and divisor; or a series of different multiplication and division operations combined, each with different factors. The number of such calculations depends upon the size of the factors used and the ability of the machine to store them on the read cycle.

This problem (Figure 42) shows how four different sets of products are developed, from four different multipliers and four different multiplicands punched in the same card. Only the planning chart and control wiring are explained.

*Read.* The multiplier $A$ for the first calculation is read into 1R, and the multiplicand $B$ into 2L. The read cycles impulse to 1R is wired through a row of bus hubs since 1R is to be impulsed at other times for other multipliers. The remaining multipliers $C$, $E$, and $G$ are stored in units 2R, 3R, and 4R. The remaining multiplicands $D$, $F$, and $H$ are stored in units 3L, 4L, and counter 2, respectively. These units are also impulsed on the read cycle.

*Program 1.* $B$ is read out of 2L and the product of $A \times B$ is developed in counter 4-5-6. The exit for this program is wired to the multiply hubs through the second row of bus hubs, into which other impulses directed to the multiply hubs will be wired.

*Program 2.* The product, $P_1$, is read out of counter 4-5-6 and entered into storage unit 6 for punching, which will begin on the next cycle. The counter is also reset, with the reset to 5 properly wired to adjust the next product, $C \times D$. The entry and punch of storage unit 6 are impulsed through the third row of bus hubs.

At the same time that $P_1$ is read out to punch, factor $C$ is read out of 2R and entered into 1R, which is impulsed through the first row of bus hubs.

The couple exit of program 2 is wired to the pickup of co-selector 2 to select the reset-to-5 wiring as explained under wiring for the right panel.

*Program 3.* Multiplication of $C \times D$ takes place by reading out $D$ from 3L and developing $P_2$ in counter 4-5-6. Program 3 reaches the multiply hubs through the second row of bus hubs.

*Program 4.* The product $P_2$ is read out of counter 4-5-6 and entered into storage unit 6 for punching, which begins on the next cycle. This counter is also reset, with the reset to 5 properly wired to adjust the next product, $E \times F$. The entry and punch of storage unit 6 are impulsed through the third row of bus hubs. At the same time that $P_2$ is read out to punch, factor $E$ is read out of 3R and entered into 1R which is impulsed through the first row of bus hubs. The couple exit of program 4 is wired to the pickup of co-selector 1, to select reset-to-5 wiring as explained under wiring for the right panel.

*Program 5.* Multiplication of $E \times F$ takes place by reading out $F$ from 4L and developing $P_3$ in counter 4-5-6. Program 5 reaches the multiply hubs through the second row of bus hubs.

*Program 6.* The product $P_3$ is read out of counter 4-5-6 and entered into storage unit 7 for punching, which begins on the next cycle. The counter is also reset on this cycle without half-adjustment. Program 6 is wired directly to storage unit 7 entry and punch. At the same time that $P_3$ is read out to PUNCH, factor $G$ is read out of 4R and entered into 1R, which is impulsed through the first row of bus hubs.

*Program 7.* The multiplication of $G \times H$ takes place by reading out $H$ from counter 2 and developing $P_4$ in counter 4-5-6. Program 7 reaches the multiply hubs through the second row of bus hubs.

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT DIVR.–MULT. 1R | COUNTER DIVIDEND 1 | COUNTER DIVIDEND 2 | COUNTER DIVIDEND 3 | COUNTER 4 | COUNTER 5 | COUNTER 6 | STORAGE UNITS 2L | STORAGE UNITS 2R | STORAGE UNITS 3L | STORAGE UNITS 3R | STORAGE UNITS 4L | STORAGE UNITS 4R | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH 6L | 6R | 7L | 7R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  | READ CYCLE |  | A<br>4 Position |  | +H<br>4 Position |  |  |  |  | B<br>6 Position | C<br>5 Position | D<br>6 Position | E<br>6 Position | F<br>6 Position | G<br>4 Position |  |  |  |  |
|  | 1 | Mult. |  |  |  |  |  | + | + | +$P_1$ | RO |  |  |  |  |  |  |  |  |  |
|  | 2 |  |  | C |  |  |  |  |  | RO<br>RE<br>5(Corr) |  | RO |  |  |  |  |  | $P_1$<br>Punch<br>↓ |  |  |
|  | 3 | Mult. |  |  |  |  |  | + | + | +$P_2$ |  |  | RO |  |  |  |  |  |  |  |
|  | 4 |  |  | E |  |  |  |  |  | RO<br>RE<br>5(Corr) |  |  |  | RO |  |  |  | $P_2$<br>Punch<br>↓ |  |  |
|  | 5 | Mult. |  |  |  |  |  | + | + | +$P_3$ |  |  |  |  | RO |  |  |  |  |  |
|  | 6 |  |  | G |  |  |  |  |  | RO<br>RE |  |  |  |  |  | RO |  |  |  | $P_3$<br>Punch<br>↓ |
|  | 7 | Mult. |  |  |  | RO |  | + | + | +$P_4$ |  |  |  |  |  |  |  |  |  |  |
|  | 8 |  |  |  |  | RE |  |  |  | RO<br>RE<br>5(Corr) |  |  |  |  |  |  |  |  |  | $P_4$<br>Punch |
|  | 9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | ↓ |

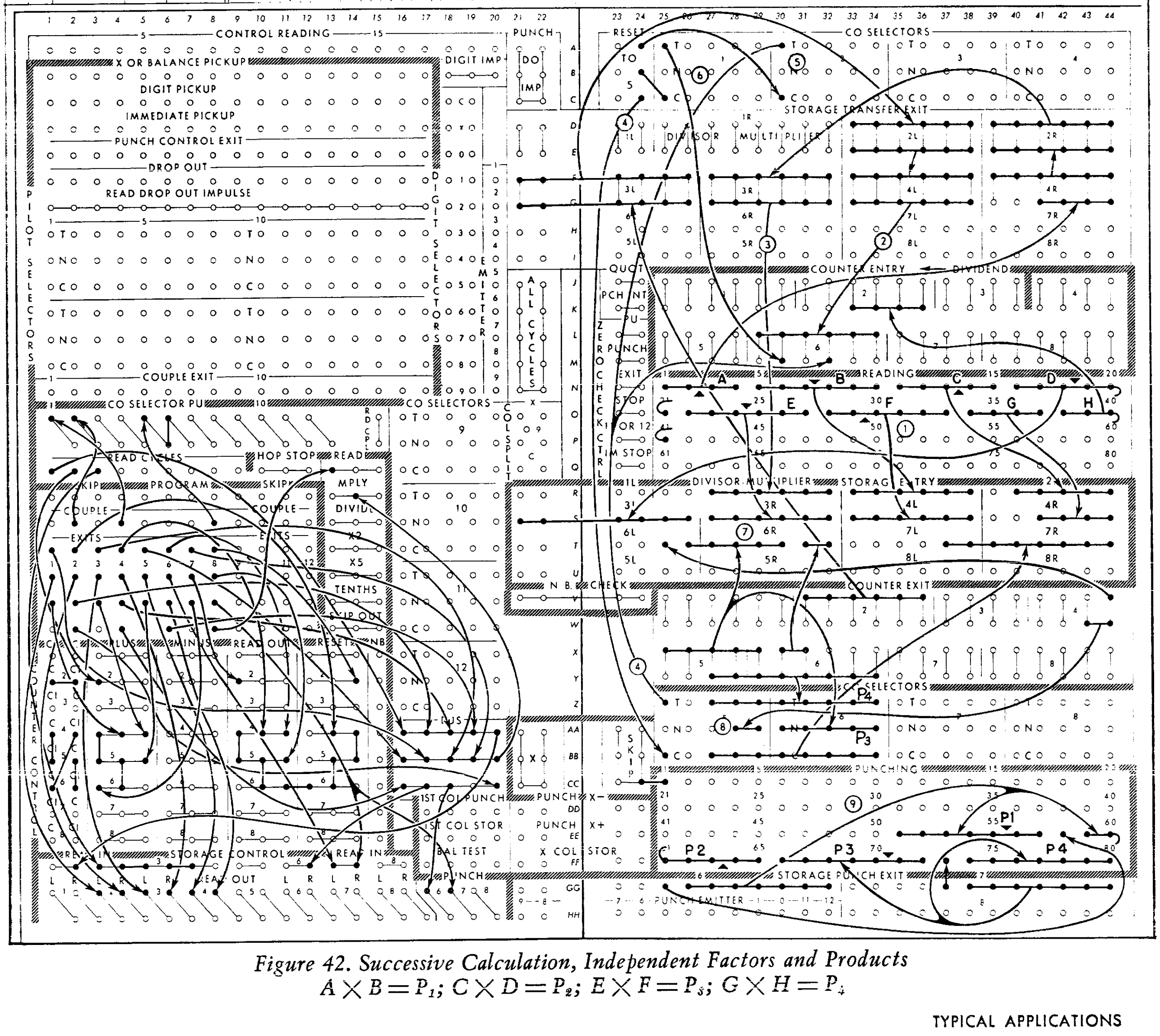


*Figure 42. Successive Calculation, Independent Factors and Products*
$A \times B = P_1$; $C \times D = P_2$; $E \times F = P_3$; $G \times H = P_4$

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT DIVR.-MULT. 1R | COUNTER DIVIDEND 1 2 3 4 5 6 | STORAGE UNITS 2L | 2R | 3L | 3R | 4L | 4R | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH 6L | 6R | 7L | 7R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | READ CYCLE | | 11 | | 11 | 11 11 11 | 123456 | 789123 | | | | | | |
| | 1 | Multiply | | | 01358024680353 | | | RO | RO | | | | | | |
| | 2 | | | 11 11 1111 | | RO | RO | | | | | | | | |
| | 3 | Multiply | | | 13717420876492245653<br>01371742101229492456 53 | | | RO | RO | | | | | | |
| | | Read | | | RO<br>RE | | | | | | | 0137 | 174210 | 122949 | 245653 |

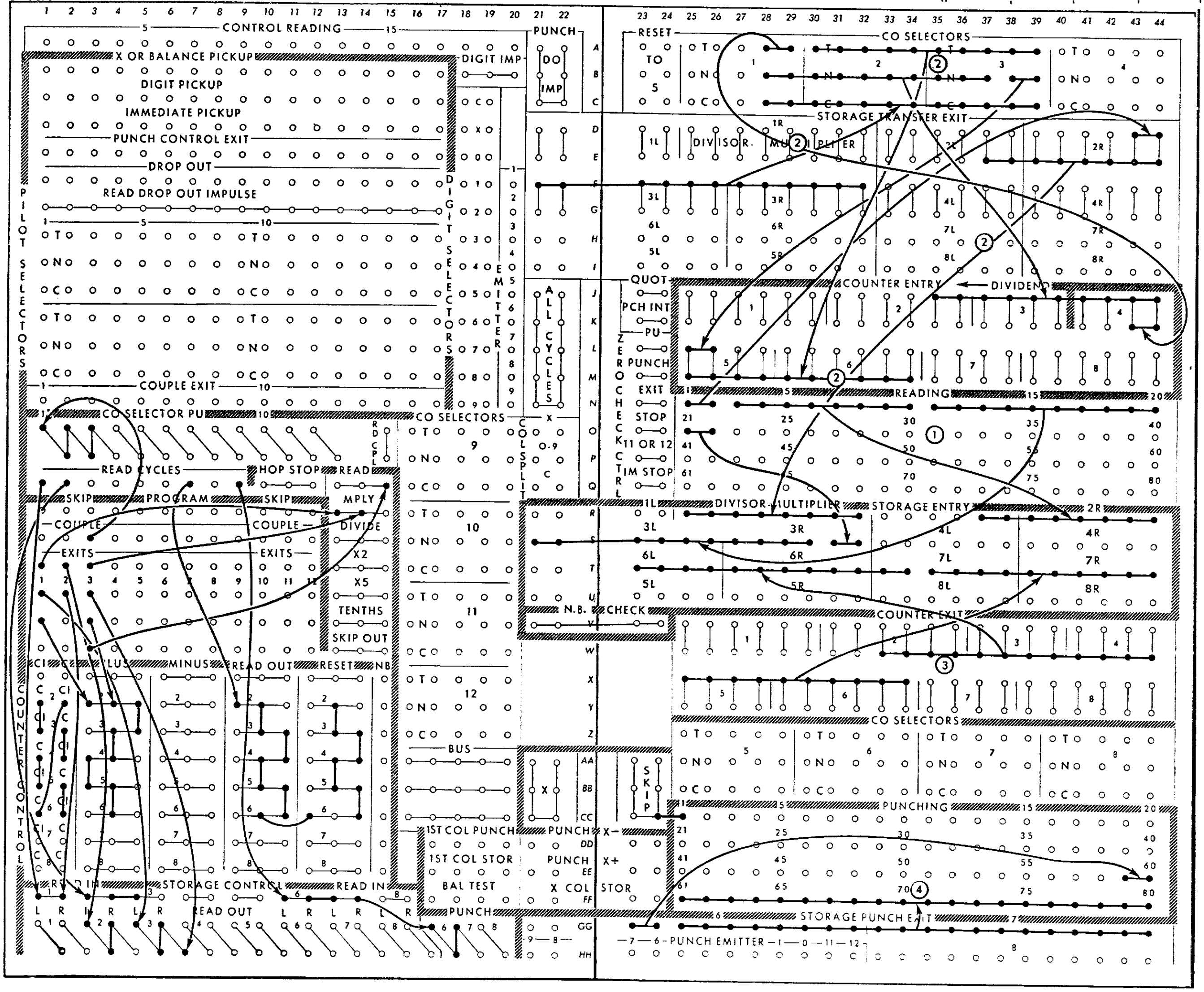


Figure 43. Multiplier Expansion

## MULTIPLIER EXPANSION

The multiplier may be expanded beyond the 8-digit capacity of the multiplier unit 1R by calculating the multiplication in parts (Figure 43). For example, the calculation of:

123,456,789,123 × 1,111,111,111 = 137,174,210,122,949,245,653

may be accomplished as follows:

$$\begin{array}{llr} (1) & 123{,}456{,}789{,}123 \times 11 = & 135{,}802{,}468{,}035{,}3\phantom{00{,}000{,}000} \\ (2) & 123{,}456{,}789{,}123 \times 11{,}111{,}111 = & \underline{1{,}371{,}742{,}087{,}649{,}245{,}653} \\ & & 137{,}174{,}210{,}122{,}949{,}245{,}653 \end{array}$$

Note from the above example that a 10-digit multiplier is treated in two parts: (1) The multiplicand is multiplied by the two high-order digits of the multiplier and the result is shifted to the left the number of positions in the remaining part of the multiplier, or eight positions. (2) The multiplicand is multiplied by the remaining eight positions of the multiplier.

There is no set rule for the way the multiplier must be split, when it exceeds eight positions. The best approach to solving this type of problem is to do it on paper first and then apply the same procedure to the machine wiring.

### Planning Chart and Control Wiring

*Read.* The two high-order digits of a 10-digit multiplier are read into 1R and the remaining eight digits into storage unit 2. The multiplicand is entered into storage unit 3.

*Program 1.* The multiplicand is multiplied by 11 (tenth and ninth positions of the multiplier) and the product is developed with its units position shifted eight places to the left in the products counter. The number of places to be shifted should equal the number of positions in the remaining part of the multiplier.

Counters 2, 3, 4, 5, and 6 are coupled and impulsed to add to develop the product. Storage units 3L and 3R are read out and the multiply hub is impulsed.

*Program 2.* The remaining part of the multiplier is read out of 2L and 2R and entered into 1R.

*Program 3.* The multiplicand is multiplied by the remaining eight positions of the multiplier and the product is developed normally in counter 2-3-4-5-6, adding to the first product already standing in this counter. Counter 2-3-4-5-6 is impulsed to add to develop the product. Storage units 3L and 3R are read out and the multiply hub is impulsed. The couple of program 3 is wired to the pickup of co-selectors 1, 2, and 3, to select the multiplicand as explained under wiring for the right panel. Since this is the last program used, it is wired to read.

*Read.* The product of the multiplicand times the 10-digit multiplier stands in counters 2 through 6. It is read out into storage units 6 and 7 for punching. The counters are reset.

### Right-Hand Panel Wiring

1. The first two positions of the multiplier are wired into 1R through the common hubs of 2R exit to avoid split wires. The remaining eight positions of the multiplier are entered into storage unit 2. The multiplicand is entered into storage unit 3.

2. The 12-position multiplicand is wired to the common of co-selectors 1, 2, and 3. The selectors are picked up on program 3 so that the multiplicand enters the product counters starting with the ninth position for the first multiplication (selectors normal), and with the units position for the second multiplication (selectors transferred).

3. The whole product is wired out of counter exits 2 through 6 to storage entries 6 and 7 for punching.

4. The product is wired out of storage punch exit 6 and 7 to punch.

## QUOTIENT EXPANSION

A maximum of eight quotient digits may be calculated in a single dividing operation. The quotient may be expanded, however, to as many more digits as desired, by treating the problem in parts.

The following example (Figure 44) shows how a 14-digit dividend can be divided by a 2-digit divisor to obtain a 13-digit quotient (adjusted). This problem cannot be calculated in a single dividing operation even though the dividend counter is large enough to accommodate the 14-digit dividend.

The rule stated under "Size of Factors in Division" is that the dividend must never be larger than the number of significant digits in the divisor, plus seven. It follows, therefore, that with a divisor with two significant digits the dividend is limited to nine digits, and, if the divisor has one significant digit, the dividend is limited to eight digits. The dividend in this example, therefore, must be handled in parts with eight digits to the left in the first dividing operation, and six digits to the right in the second operation, as follows:

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT DIVR.–MULT. 1R | COUNTER DIVIDEND 1 | COUNTER DIVIDEND 2 | COUNTER DIVIDEND 3 | COUNTER 4 | COUNTER 5 | COUNTER 6 | STORAGE UNITS 2L | 2R | 3L | 3R | 4L | 4R | PUNCH UNITS 6L | 6R * | 7L | 7R * |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | READ CYCLE | | 12 | | 12345678 | | | | | | 912345 | | | | | | | | |
| | 1 | Divide | | RO | – | –<br>Remainder 06 | – | | +01 | 028806 | | | | | | | | | | |
| | 2 | (Exits 2 & 3) | | | RO<br>RE | RO<br>RE | RO<br>RE | | RO<br>RE | RO<br>RE S | | | Remainder<br>06 | | | | 01 | 028806<br>Punch<br>↓ | | |
| | 3 | (Exit 4) | | | | 06912345 | | | | | | RO | RO | | | | | | | |
| | 4 | (Exit 5)<br>Divide | | RO | – | – | – | | 5760287<br>5760292 | | | | | | | | | | | |
| | 5 | (Exit 6) | | | RE | RE | RE | | RO<br>RE | RO<br>RE | | | | | | | | | | 576029<br>Punch<br>↓ |

*UNITS POSITION MUST BE WIRED TO PUNCH

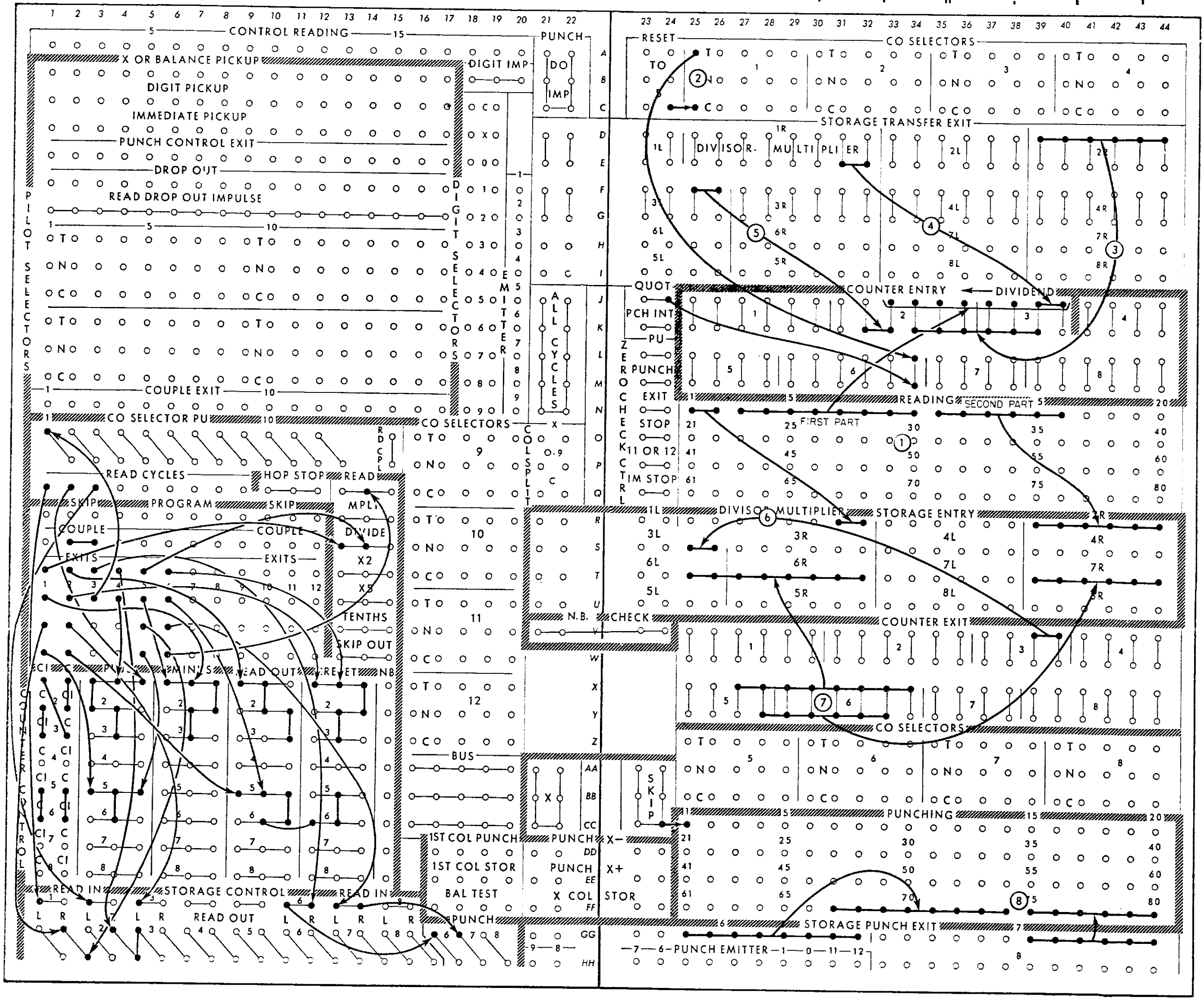
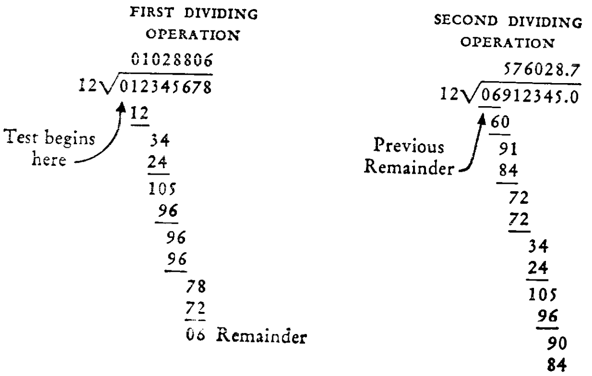


*Figure 44. Quotient Expansion*

12345678912345 ÷ 12 = 1028806576028.7
(Carried out one extra place for ½ correction)



## Planning Chart and Control Wiring

*Read.* The 14-digit dividend is split into two parts. The eight high-order positions are added in the dividend counter and the remaining six positions are entered into storage unit 2. The divisor is entered into 1R.

*Program 1.* The first part of the dividend is divided by 12 and the quotient is developed in counter 5-6. The divide hub is impulsed, the divisor is read out, the dividend counter subtracts, and counter 5-6 adds the quotient.

*Program 2 (Exits 2 and 3).* The remainder standing in the dividend counter must be retained so that it can be placed beside the second part of the dividend before proceeding with the next divide operation. The remainder in the dividend counter is read out and entered into storage unit 3. The dividend counter is also reset. At the same time, the quotient for the first part of the problem is read into storage unit 6 for punching. The RESET TO 5 is wired to the units position of counter 6 to adjust the quotient.

*Program 3 (Exit 4).* The second part of the original dividend is read out of storage unit 2R and added into the dividend counter, offset one position for half-adjustment. The remainder from the previous dividing operation is read out of 3L and placed to the left of the second part of the dividend. The machine is ready to divide the second part of the dividend by 12.

*Program 4 (Exit 5).* The second part of the original dividend and the remainder from the first dividing operation are divided by 12, and the second part of the quotient is developed in counter 5-6. The divide hub is impulsed, the divisor is read out, the dividend counter subtracts, and counter 5-6 adds. This part of the quotient is adjusted.

*Program 5 (Exit 6).* The dividend is reset. The second part of the quotient is read out of counter 5-6, with one position dropped, to storage unit 7 for punching. The first part of the quotient is punched while the second part is being calculated. The last program is always wired to READ.

## Right-Hand Panel Wiring

1. The divisor is wired to 1R. The first part of the dividend is wired to the dividend counter and the second part to storage entry 2R.

2. Co-selector 1 is picked up from program 2. Its function is to select the reset to 5 so that it will enter the units position of the quotient counter to correct the second part of the quotient only.

3. The second part of the dividend is wired out of 2R into the dividend counter, offset one position.

4. The divisor exit is wired to the dividend, as for all dividing operations.

5. The remainder from the first division is wired out of 3L to the left of the second part of the dividend.

6. The remainder from the first division is wired out of counter 3 to 3L entry.

7. The first part of the quotient is wired to storage entry 6, and the second part of the quotient is wired to storage entry 7. Only the exact number of positions left to be punched are wired out of the quotient counter the second time. If more positions are wired, unwanted zeros would separate the first and second parts of the quotient and make the answer wrong.

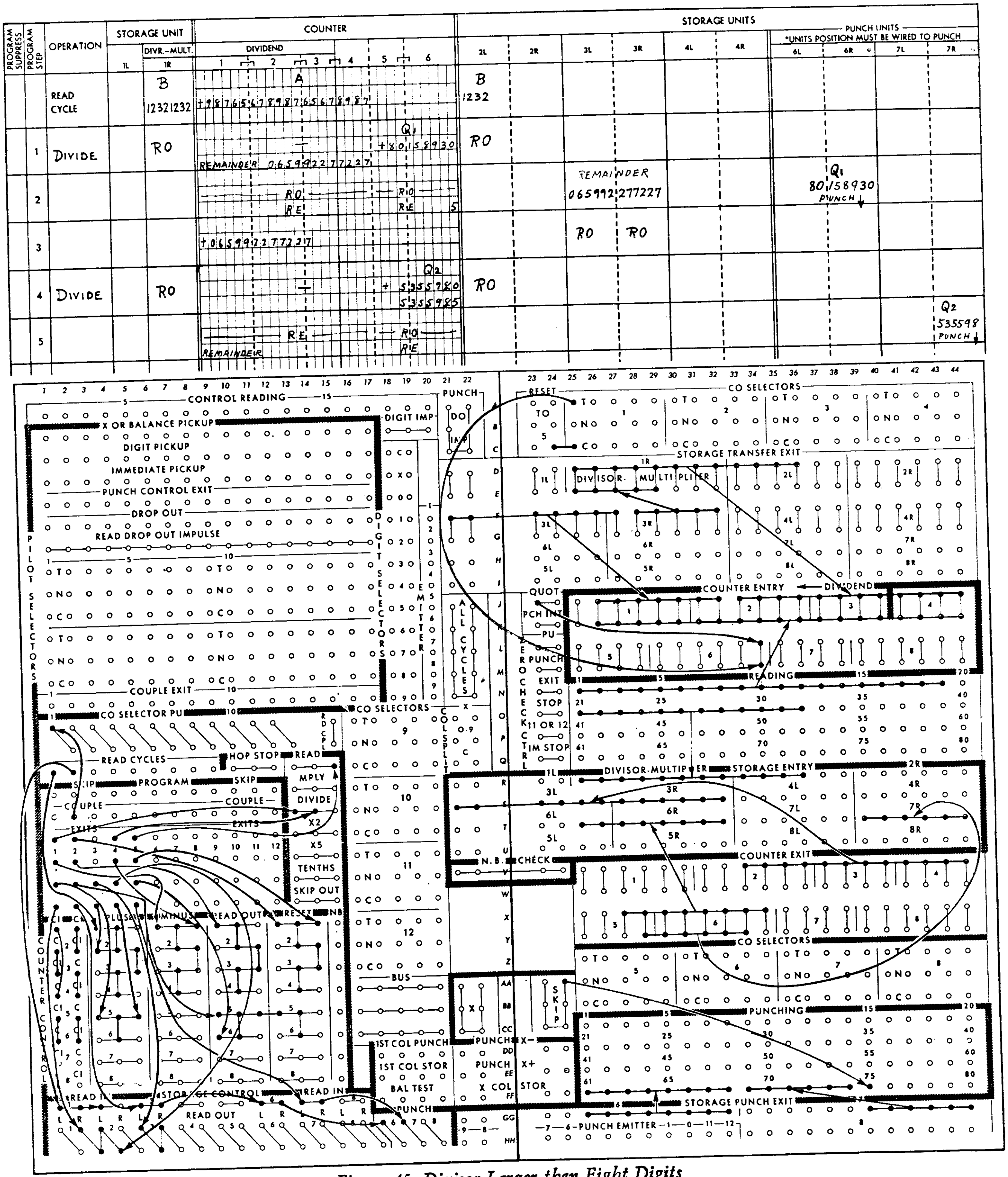
8. The first eight positions of the quotient are punched out of storage punch exit 6, and the last six positions are punched out of storage punch exit 7.

## DIVISOR LARGER THAN EIGHT DIGITS

THIS problem illustrates a method of dividing when the divisor field is greater than the divisor unit 1R. The divisor field contains 12 positions. A dividend field of 19 digits is used and the division is carried out to compute and punch a quotient of 14 digits, by use of quotient expansion. Eight quotient digits are computed on the first division step and seven on the second, half-correcting and dropping the units position.

Within the counter capacity of the machine, different sizes of divisor and dividend fields can be used to compute quotients of various size by properly positioning the factors according to the rules stated in the following paragraphs. Only the control panel wiring is shown in Figure 45.

*Read.* The dividend is entered in counter 1-2-3-4. Whenever a divisor is expanded beyond the normal eight positions, the dividend counter must be expanded the same number of positions, regardless of the size of the dividend field in the card. Therefore,



| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT DIVR.-MULT. 1R | COUNTER DIVIDEND 1 2 3 4, 5 6 | STORAGE UNITS 2L | 2R | 3L | 3R | 4L | 4R | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH 6L | 6R | 7L | 7R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | READ CYCLE | | B<br>12321232 | A<br>+9876567898765678987 | B<br>1232 | | | | | | | | | |
| | 1 | DIVIDE | | RO | Q1<br>+80158930<br>REMAINDER 065992277227 | RO | | | | | | | | | |
| | 2 | | | | RO RO<br>RE RE 5 | | | REMAINDER<br>065992 | 277227 | | | 80 | Q1<br>158930<br>PUNCH | | |
| | 3 | | | | +065992277227 | | | RO | RO | | | | | | |
| | 4 | DIVIDE | | RO | Q2<br>+5355980<br>5355985 | RO | | | | | | | | | |
| | 5 | | | | RE RO<br>REMAINDER RE | | | | | | | | | | Q2<br>535598<br>PUNCH |

*Figure 45. Divisor Larger than Eight Digits*

since there are four extra places in the divisor in this problem, the dividend counter must be expanded four places also. If a divisor field had a maximum of ten places, the dividend counter would have to be expanded two places, making the units position of the dividend counter fall in the third place of counter four.

In this example, the divisor is entered in storage units 1R and 2L. When a divisor field is larger than eight positions, the high-order positions must enter the divisor unit 1R; the extra positions can be wired to any other storage unit (including 1L). To perform the division operation, at least one significant digit must be in the divisor unit 1R. Therefore, a 12-digit divisor field must be punched with at least five significant digits.

*Program 1.* The first part of the quotient is computed and developed in counter 5-6. The divide hub is impulsed, the divisor is read out, the dividend counter subtracts, and the quotient counter adds. The whole 12-position divisor is wired from storage to the dividend counter, with the units position of the divisor wired to the units position of the dividend counter.

*Program 2.* The first part of the quotient, $Q_1$, is entered in storage unit 6 for punching. The remainder (maximum of twelve digits) in the dividend counter is entered in storage unit 3. The dividend and quotient counters are both impulsed to read out and reset, with the reset to 5 wired to the units position of counter 6 to half-adjust the second part of the quotient.

*Program 3.* The remainder is entered in the dividend counter for the second part of the division. It is entered to the left in the dividend counter in order to carry the division out to the maximum number of places and compute a 7-digit quotient.

*Program 4.* The second part of the quotient is computed and developed in counter 5-6.

*Program 5.* The second part of the quotient, $Q_2$, is entered in storage unit 7 for punching. The quotient counter is read out and reset, and the dividend counter is reset.

## PROGRAM EXTENSION

ALTHOUGH the control panel provides for only 12 program exits, once a read cycle is taken by the machine, programming continues until READ is impulsed. In all problems illustrated heretofore, programming is stopped at the end of the problem by impulsing READ on the last program taken. The problems illustrated do not require more than 12 programs and READ is impulsed on the first round of programs.

If READ is not impulsed, programming continues indefinitely and program exits 1 to 12 become program exits 13 to 24 on the second round, 25 to 36 on the third round, and so on. It thus becomes possible to extend programs beyond 12 by controlling the impulse wired to READ.

Program 1 exits are always active on a repeated round and, therefore, cannot be skipped after the first round of programs, but program 1 exits on repeated rounds cannot be initiated unless program 12 exits are active on the preceding round. In other words, in all problems requiring extension of programs, program exit 12 cannot be skipped.

Program exits used on more than one round of programs may need to be selected to determine their identity. For example, program 1 exits on the first round become program 13 exits on the second round. Selection of program 1 exits is necessary, therefore, to distinguish between program 1 on the first round and program 13 on the second round. This selection is necessary only if the functions to be performed on these programs differ. If functions are to be identical, as illustrated in the square root problem, selection is not necessary.

Figure 46A illustrates a method of extending programs up to 23; program exits 1 through 11 are used on both rounds to perform different functions, and program 12 is used only on the first round.

Figure 46B illustrates a method of extending programs up to 35. Program exits 1 through 11 are used on the first round as programs 1-11, on the second round as programs 13-23, and on the third round as programs 25-35; program 12 exits are used only on the first two rounds, as program 12 on the first and as program 24 on the second. As in Figure 43, co-selector positions are necessary to identify programs (1, 2, 3, and so on) on the first round; in addition, co-selector positions (transferred for the third round) are required to distinguish between programs 13 and 25, 14 and 26, 15 and 27, and so on. Five pilot selectors are required to control the co-selectors for the three rounds.

The co-selectors used (9-12) in both figures are not standard; any co-selectors available can be used.

## Control Panel Wiring, Figure 46A

1. Program couple 12 is wired to the D pickup of pilot selector 1. Since the selector is wired for normal drop-out, the selector transfers on program 12 and remains transferred for the remainder of the problem.

2. When pilot selector 1 is transferred, couple exit of pilot selector 1 emits an impulse which is used, in turn, to pick up co-selectors.

3. Co-selector 11 is used to distinguish between programs 1 and 13. Impulses are emitted on both programs from program 1 exits. When the exits are wired to the common hubs of co-selector 11, the normal hubs provide exits for program 1 and the transferred hubs provide exits for program 13.

4. Co-selector 10 is used to distinguish between programs 11 and 23. Impulses are emitted on both programs from program exits 11. When these exits are wired to the common hubs of co-selector 10, the normal hubs provide exits for program 11 and the transferred hubs provide exits for program 23.

5. Program 23 exit wired to READ.

Any selection required for program exits 2 through 10 is accomplished in a manner similar to that shown for program exits 1 and 11.

## Control Panel Wiring, Figure 46B

1. On the first round of programs all selectors are normal, and program 1 exits pass through the normal side of co-selector 12.

2. On program 12 of the first round, program couple 12, wired to the D pickup of pilot selector 1, transfers the selector immediately, and the selector remains transferred for the remainder of the problem.

3. On program 13, program couple 1 passes through the transferred side of pilot selector 1 and transfers pilot selector 2 and co-selector 12 immediately. Both selectors remain transferred for the remainder of the problem. Program 1 exits (program 13) pass through the transferred side of co-selector 12 and the normal side of co-selector 11.

4. On program 24, program couple 12 passes through the transferred side of pilot selector 2 and immediately transfers pilot selector 3, which remains transferred for the remainder of the problem.

5. On program 25, program couple 1 passes through the transferred side of pilot selector 3 and transfers pilot selector 4 and co-selector 11 immediately. Both selectors remain transferred for the remainder of the problem. Program 1 exits (program

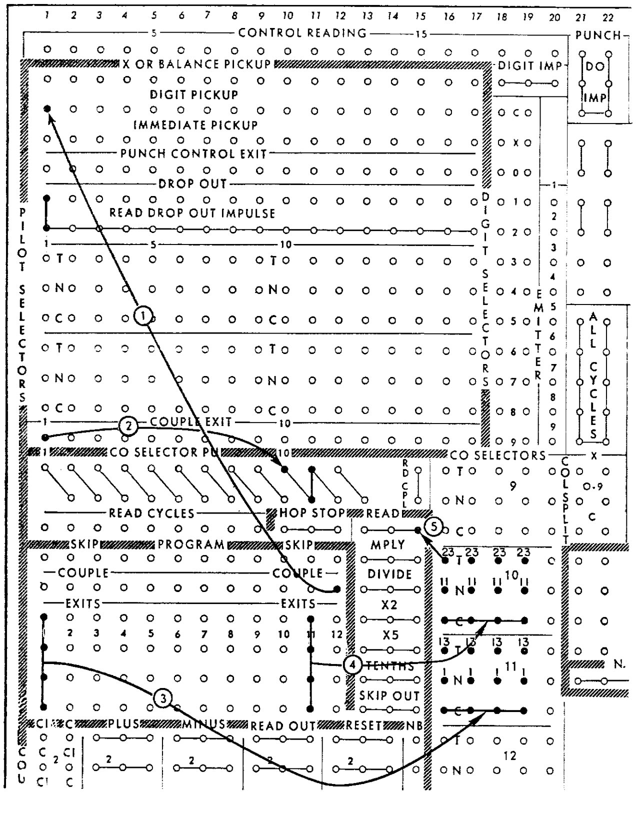


Figure 46A. Program Extension, 23 Programs

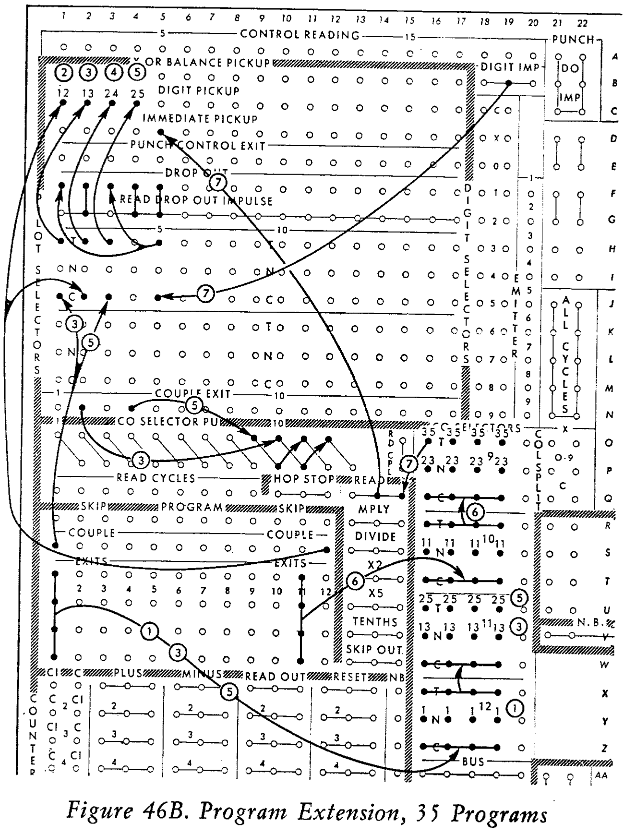


Figure 46B. Program Extension, 35 Programs

25) pass through the transferred sides of co-selectors 12 and 11.

6. As in the wiring for program 1 exits, program 11 exits are identified as program 11, 23, or 35 by wiring through co-selectors 10 and 9.

7. Program 25 is the last program taken. It is wired to READ and also to the immediate pickup of pilot selector 5. The digit impulse passing through the transferred side of this selector drops out pilot selectors 1 and 3. When pilot selectors are used to select program couple 1, normal read drop-out cannot be used. This is because program couple 1 originates slightly before drop-out time and is active while the selector drops. Program couple 1, therefore, would pass partly through the transferred side of the selector and partly through the normal side. This timing interference would cause erroneous results. Thus, it is necessary to drop out selectors 1 and 3 at the end of the last program taken. Program exits cannot be used to drop out pilot selectors because the early portion of the program cycle would burn the selector points.

Wiring for selection of any other program exits is the same as that shown for program exits 1 and 11.

## CROSSFOOTING, TWELFTHS

IN THIS problem (Figure 47) eight fields, each punched with dozens and fractional dozens no higher than 9/12, are crossfooted. The dozens and twelfths are crossfooted separately; the sum of the twelfths is converted to whole dozens and fractional dozens, and the grand total of the whole dozens and the remaining twelfths is punched. Fractional dozens are punched in one column with 10/12 designated by an X punch and 11/12 by a 12 punch.

The same wiring principle can be followed when more fields are to be crossfooted or when twelfths are to be accumulated from a series of detail cards to be punched on an X summary card.

### Planning Chart and Control Wiring

*Read.* The whole dozens in field A are added in counter 5. The partial dozens (*a*) are added in counter 1-2-3, where the partial dozens from all fields are accumulated to become the dividend later in the problem. The partial dozens (*a*) are also entered into counter 4 where they are subtracted. Counter 4 will be used later as a testing counter to determine whether or not any dividend has been accumulated. The whole dozens and the partial dozens of field B are added separately in counter 6. The whole and partial dozens of fields *C, D, E, F, G,* and *H* are entered in storage units 2L, 2R, 3L, 3R, 4L, and 4R, respectively. A 12 is entered into the divisor from the emitter.

*Program 1.* The whole dozens (*C*) and the partial dozens (*c*) are read out of 2L. The whole dozens are added in counter 1-2-3 and subtracted in counter 4. Both the whole and partial dozens (*D, d*) are read out of 2R and added in counter 6.

*Program 2.* The whole and partial dozens (*E, e*) are read out of 3L. The whole dozens are added in counter 5; the partial dozens are added in counter 1-2-3 and subtracted in counter 4. Both the whole and partial dozens (*F, f*) are read out of 3R and added in counter 6.

*Program 3.* The whole and partial dozens (*G, g*) are read out of 4L. The whole dozens are added in counter 5; the partial dozens are added in counter 1-2-3 and subtracted in counter 4. The whole and partial dozens (*H, h*) are read out of 4R and added in counter 6.

*Program 4.* The sum of fields *B, D, F,* and *H* is read out of counter 6. The whole dozens are added in counter 5; the partial dozens for these fields are added in counter 1-2-3 and subtracted in counter 4. Counter 6 is reset. Pilot selector 1 is picked up with the NB (negative balance) of counter 4. If counter 4 is plus at this time, there are no partial dozens and it is unnecessary to divide on program 5. Program 6, which controls reading out the remainder, may therefore be skipped. If counter 4 goes negative on the read cycle, or on program 1, 2, or 3, division is necessary on program 5.

*Program 5.* If counter 4 is negative on program 5, the total of the partial dozens is divided by 12 to convert it to whole dozens and twelfths, by impulsing divide through the transferred side of pilot selector 1. Unit 1R is impulsed to read out only if counter 4 is negative by wiring through the transferred side of pilot selector 1. The one-position quotient develops in counter 5 and adds to the whole dozens already standing in the counter. The dividend is positioned so that only whole numbers (no decimals) are computed when dividing. The remainder standing in counter 1-2-3 is the number of twelfths to be punched in the result field.

If counter 4 is positive, program 6 is skipped by wiring a program 5 exit to the program 7 skip, through the normal side of pilot selector 1.

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT |  | COUNTER |  |  |  |  |  | STORAGE UNITS |  |  |  |  |  |  |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  | DIVR.—MULT. | DIVIDEND |  |  |  |  |  |  |  |  |  |  |  | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH |  |  |  |
|  |  |  | 1L | 1R | 1 ÷ | 2 ÷ | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R * | 7L | 7R * |
|  |  | READ CYCLE |  | 12 |  |  | a | — a | A | B b | C c | D d | E e | F f | G g | H h |  |  |  |  |
|  | 1 |  |  |  |  |  | c | — c | C | D d | RO | RO |  |  |  |  |  |  |  |  |
|  | 2 |  |  |  |  |  | e | — e | E | F f |  |  | RO | RO |  |  |  |  |  |  |
|  | 3 |  |  |  |  |  | g | — g | G | H h |  |  |  |  | RO | RO |  |  |  |  |
|  | 4 | NB Test CTR 4<br>Pickup Pilot Sel. 1 |  |  |  |  | bd<br>fh | bd<br>fh | BD<br>FH | RO<br>RE |  |  |  |  |  |  |  |  |  |  |
|  | 5 | Divide (if CTR. 4 is MINUS) |  | RO (if CTR 4 ⊖) | —<br>REMAINDER—PARTIAL DOZ. | — | — |  | +Q<br>WHOLE DOZENS |  |  |  |  |  |  |  |  |  |  |  |
| ✓ | 6 | Suppress if CTR. 4 ⊕<br>Pickup Pilot Sel. 6 & 7 |  |  | RO | RO | RO |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | 7 | (Exits 7 & 8) |  |  | RO<br>RE | RO<br>RE | RO<br>RE | RE | RO<br>RE |  |  |  |  |  |  |  |  | WHOLE DOZENS PUNCH ↓ |  | PARTIAL DOZENS (1—9) PUNCH ↓ |
|  | 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

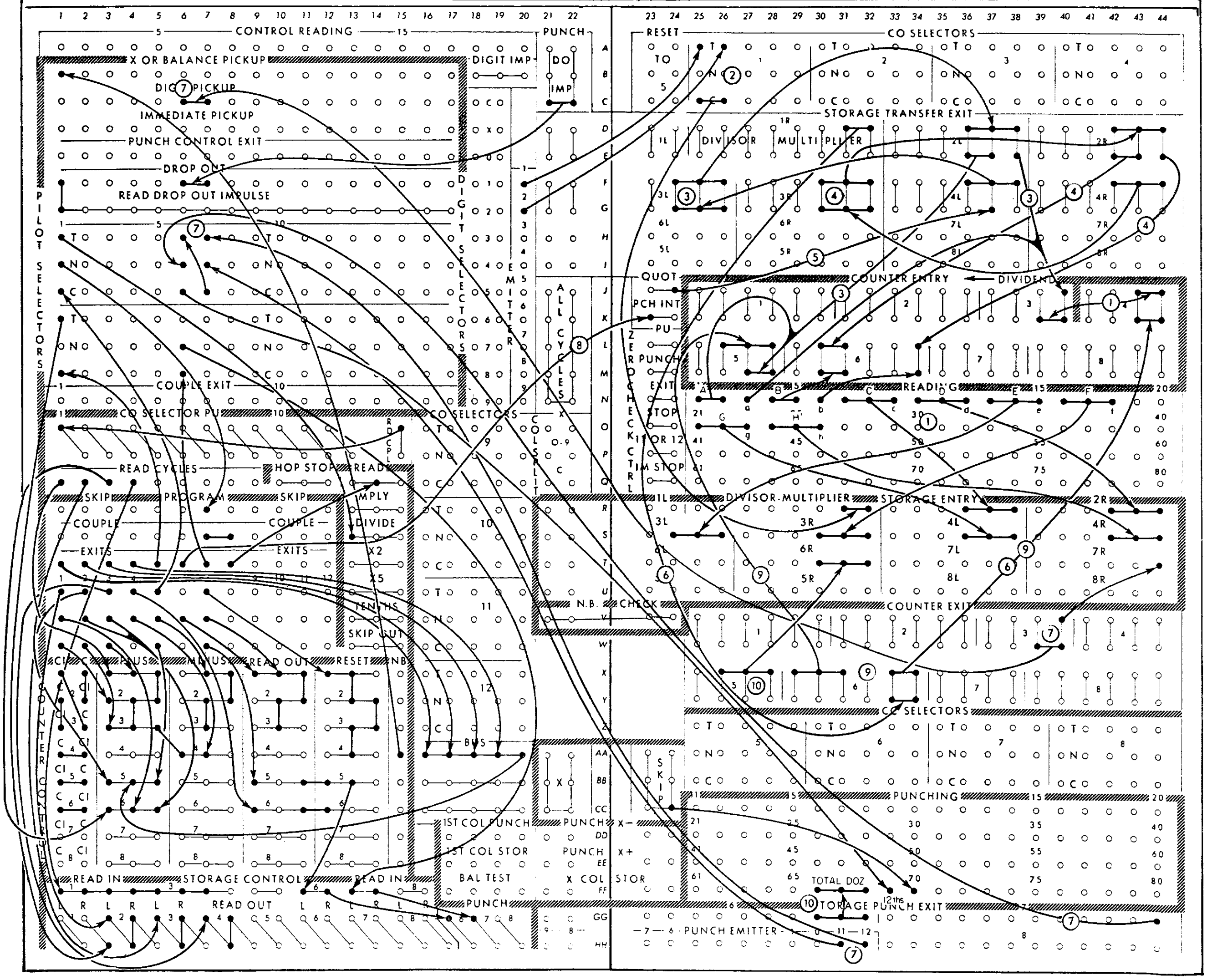


Figure 47. Crossfooting, Twelfths

*Program 6.* If the remainder in counter 1-2-3 is any digit from 1 through 9, it is punched directly in the twelfths column of the total field. However, if the remainder is 10, it must be converted to an X punch, and, if it is 11, it must be converted to a 12 punch. In order to control the punching of the proper designation, pilot selectors 6 and 7 are picked up by the tens and units positions of the remainder in counter 1-2-3. Counter 1-2-3 is impulsed to read out.

*Program 7 (Exits 7 and 8).* The total of the whole dozens is read out of counter 5 to storage unit 6 for punching. Counter 5 is reset. The fractional dozen is read out of the dividend counter into storage unit 7 for punching only if it is 9 or less. Storage unit 7 is controlled to read in for digits 1-9 only, by wiring a program exit through the normal side of pilot selector 6. Counter 1-2-3 is reset. Counter 4 is reset. Since this is the last program used, READ is impulsed.

## Right-Hand Panel Wiring

1. The whole dozens in field *A* are wired to counter 5, while the partial dozens (*a*) are entered in counter 1-2-3 and counter 4. Counters 1-2-3 and 4 are laced together so that any amounts entering one counter will enter the other also. After all the fields are crossfooted, the total whole dozens will be in counter 5, and the total partial dozens in counters 1-2-3 and 4. The partial dozens are entered in the extreme right of counter 1-2-3, since no decimals will be desired when dividing by 12. Field *B* is wired to counter 6, the partial dozens to the right and the whole dozens to the left. The accumulated partial dozens will not exceed two positions, and the accumulated whole dozens will not exceed three positions. Field *C* is entered in 2L, *D* in 2R, *E* in 3L, *F* in 3R, *G* in 4L, and *H* in 4R.

2. A 12 is wired from the digit emitter to the divisor storage unit through the transferred side of co-selector 1, which is picked up by the read couple.

3. The whole dozens (*C*) are transferred from storage 2L to counter 5 to be added to the whole dozens in field *A*, and the partial dozens (*c*) are transferred to counters 1-2-3 and 4 to be accumulated with the partial dozens (*a*). By wiring through the common hubs of the storage transfer exits, factors *E* from 3L and *G* from 4L are transferred to counter 5 and *e* and *g* are transferred to counters 1-2-3 and 4.

4. The whole and partial dozens of field *D* are transferred from 2R to counter 6 and are positioned to add with factors *B* and *b*. By wiring through the common hubs of the storage transfer exits, factors *F-f* from 3R and *H-h* from 4R are transferred to counter 6.

5. The quotient is wired to the units position of counter 5 and through the common hubs of storage transfer exits 4L, 3L, and 2L.

6. The divisor is wired to the dividend counter through the common hubs of counter 6 exit and counter 4 entry.

7. The remainder, standing in the dividend counter after the division, can be any number from 1 to 11. Remainders of 1 to 9 are punched from storage unit 7; a remainder of 10 is designated by an X punch and a remainder of 11 by a 12 punch. To control the punching of the digits 1-9, the X, or the 12, pilot selectors 6 and 7 are picked up by the tens and units positions of counter exits 1-2-3. Counter exits do not emit zero impulses and the selection of fractional dozens is controlled as follows: for digits from 01 to 09, pilot selector 6 does not transfer (zero in the tens position) and digits 1 through 9 are punched through the normal side of selector 6. For 10/12 pilot selector 6 transfers but pilot selector 7 does not. The X from the punch emitter is wired through the normal side of selector 7 and the transferred side of the selector 6. For 11/12 both selectors 6 and 7 pick up. The 12 from the punch emitter is punched through the transferred sides of both selectors. The selectors are dropped out by the punch drop-out impulse.

8. Because an X is punched for 10/12, a 12 for 11/12 and 1-9 for other twelfths, punch selection is required. This is accomplished through pilot selectors 6 and 7 which are picked up from counter exit 1-2-3 whenever there are fractional twelfths remaining in the dividend counter. Since pilot selectors 6 and 7 are picked up during calculation and not from punch control exit, punch interlock is also necessary. Punch interlock is impulsed from program exit 6, the program on which the punching selectors are picked up.

9. The whole dozens from the left of counter 6 are wired to counter 5, and the partial dozens, from the right of counter 6, are wired to counters 1-2-3 and 4. This wiring transfers factors *B*, *D*, *F*, and *H* to factors *A*, *C*, *E*, and *G*.

10. The total whole dozens are wired from counter exit 5 to storage punch entry 6 and from storage punch exit 6 to columns 66-68.

## MULTIPLICATION, TWELFTHS

As DESCRIBED in the previous problem, when twelfths are punched in a single column of the card, 10/12 is identified by an X punch, 11/12 by a 12 punch, and the remaining twelfths are punched with the digits 1 through 9. This problem shows how the X and 12 punches are entered in the machine as 10 and 11, the twelfths are changed to their decimal equivalents, and the extension of unit price times dozens is computed (Figure 48).

### Planning Chart and Control Wiring

*Read.* The partial dozens are entered in storage unit 1R (through pilot selectors 1 and 2) and the whole dozens are entered in 3R. The unit price is entered in 2R. The decimal equivalent of 1/12 (.083333) is entered into storage unit 2L from the digit emitter. Co-selector 1 is picked up at this time from read couple to select "1" impulses from the digit emitter.

*Program 1.* The decimal equivalent of 1/12 is multiplied by the partial dozens by impulsing multiply, reading out of 2L, and adding in counter 5-6. This converts the number of 12ths punched in the card to its decimal equivalent.

*Program 2.* The unit price is read out of 2R and entered into the multiplier. The whole dozens are read out of 3R and added to the decimal equivalent of the partial dozens standing in counter 5-6.

*Program 3.* The dozens are multiplied by the unit price by impulsing multiply, reading out of counter 5-6, and adding in counter 2-3-4. Since this is the last program used, READ is impulsed.

*Read.* The product is read out of counter 2-3-4 to storage unit 7 for punching. Counters 2-3-4 and 5-6 are reset.

### Right-Hand Panel Wiring

1. The unit price is entered into storage unit 2R and the whole dozens are entered into storage unit 3R. The decimal equivalent of 1/12 (.083333) is entered into 2L from the digit emitter directly, since 2L is impulsed to read only on the read cycle.

2. The partial dozens are entered to 1R through pilot selectors 1 and 2 and the common hubs of storage transfer exit 2R to eliminate split wires. To control the reading of the fractional twelfths into 1R, the control brush reading the fractions is wired to the X pickup of pilot selector 1 and the digit pickup of pilot selector 2. The pilot selectors cannot transfer with a 12 punch; therefore, the selectors would be controlled as follows:

Pilot selector 1 picked up for 10/12 only.
Pilot selector 2 picked up for 1/12 through 10/12.
Pilot selectors 1 and 2 remain normal for 11/12.

The upper positions of pilot selectors 1 and 2 are used to control the entry of the digit 1 on the read cycle into the tens positions of 1R for 10/12 or 11/12. For 11/12, a 1 from the digit emitter is wired through the normal side of both selectors; for 10/12, a 1 is wired through the transferred side of pilot selector 1 only.

The lower position of pilot selector 2 is used to control the entry into the units position of 1R which must be wired with the digit punched in the card for twelfths of 1-9, a 1 for 11/12, or nothing for 10/12. To enter the digit from the card, the column is wired through the transferred side of pilot selector 2; for 11/12, a 1 is wired through the normal side of the same selector.

3. The reset to 5 is wired to the sixth position of counter 2-3-4 to adjust the product which is carried to eight places.

4. Co-selector 1 is picked up on the read cycle to control the entry of the digit 1 from the digit emitter into storage unit 1R.

5. The decimal equivalent of 1/12 (.083333) is wired from 2L to counter 5-6 for the conversion of the partial dozen to its decimal equivalent.

6. The whole dozens are transferred from 3R to counter 5-6, offset six positions, to add to the converted partial dozens.

7. The price per dozen is transferred from 2R to the multiplier storage unit.

8. The sum of the whole and partial dozens is wired from counter 5-6 to counter 2-3-4 entries for the multiplication of unit price times quantity. The multiplicand consists of three whole numbers and six decimals.

9. The product is wired from counter 2-3-4 exits, with six decimals dropped, to storage unit 7, and is punched in columns 74-80.

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT |  | COUNTER |  |  |  |  |  | STORAGE UNITS |  |  |  |  |  | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  | DIVR.–MULT. | DIVIDEND |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R | 7L | 7R |
|  |  | READ CYCLE |  | Partial Dozens |  |  |  |  |  |  | .083333 | Unit Price |  | Whole Dozens |  |  |  |  |  |  |
|  | 1 | Mult. |  |  |  |  |  |  | Dec. Equiv. Part. Doz.<br>+ | + | RO |  |  |  |  |  |  |  |  |  |
|  | 2 |  |  | Unit Price |  |  |  |  | Whole Doz.<br>+ | + |  | RO |  | RO |  |  |  |  |  |  |
|  | 3 | Mult. |  |  |  | Product<br>+ | + | + | RO | RO |  |  |  |  |  |  |  |  |  |  |
|  |  | Read |  |  |  | RO RE | RO RE<br>5 | RO RE<br>(corr) | RE | RE<br>(Corr) |  |  |  |  |  |  |  |  | Product Punch<br>↓ |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

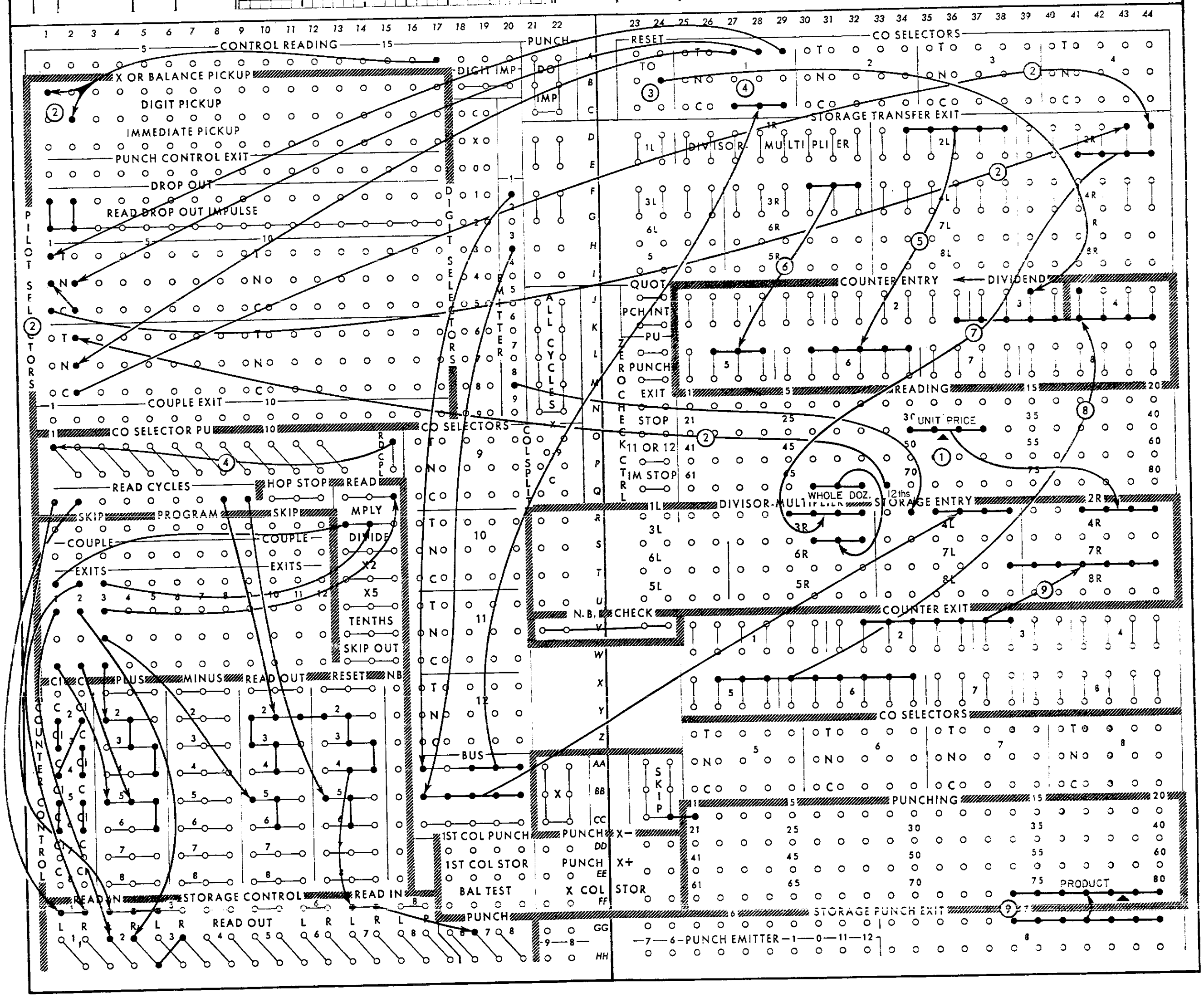


Figure 48. Multiplication, Twelfths

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT | | COUNTER | | | | | | STORAGE UNITS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | DIVR.-MULT. | DIVIDEND | | | | | | | | | | | | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH | | | |
| | | | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R * | 7L | 7R * |
| | | READ CYCLE | A | B | +C | +D | +E | +F | +G | +H | I | J | K | L | M | N | O | P | Q | R |
| | 1 | Program Exits 1 & 2 | RO | RO | +A | +B | +I | +J | +K | +L | RO | RO | RO | RO | | | | | | |
| | 2 | Program Exits 3 & 4 | | | +M | +N | +O | +P | +Q | +R | | | | | RO | RO | RO | RO | RO | RO |
| | 3 | Program Exits 5 | | | +F<br>+J<br>+P | +E<br>+I<br>+O | RO<br>RE | RO<br>RE | RO<br>RE | +G<br>+K<br>+Q | | | | | | | | | | |
| | 4 | Program Exit 6 | | | RO<br>RE | +F+C<br>+J+A<br>+P+M | | | | | | | | | | | | | | |
| | 5 | Program Exit 7 | | | | +H+G<br>+L+K<br>+R+Q | | | | RO<br>RE | | | | | | | | | | |
| | 6 | Program Exit 8 | | | | RO<br>RE | | | | | | | | | | | | | | Total Punch ↓ |
| | 7 | | | | | | | | | | | | | | | | | | | |

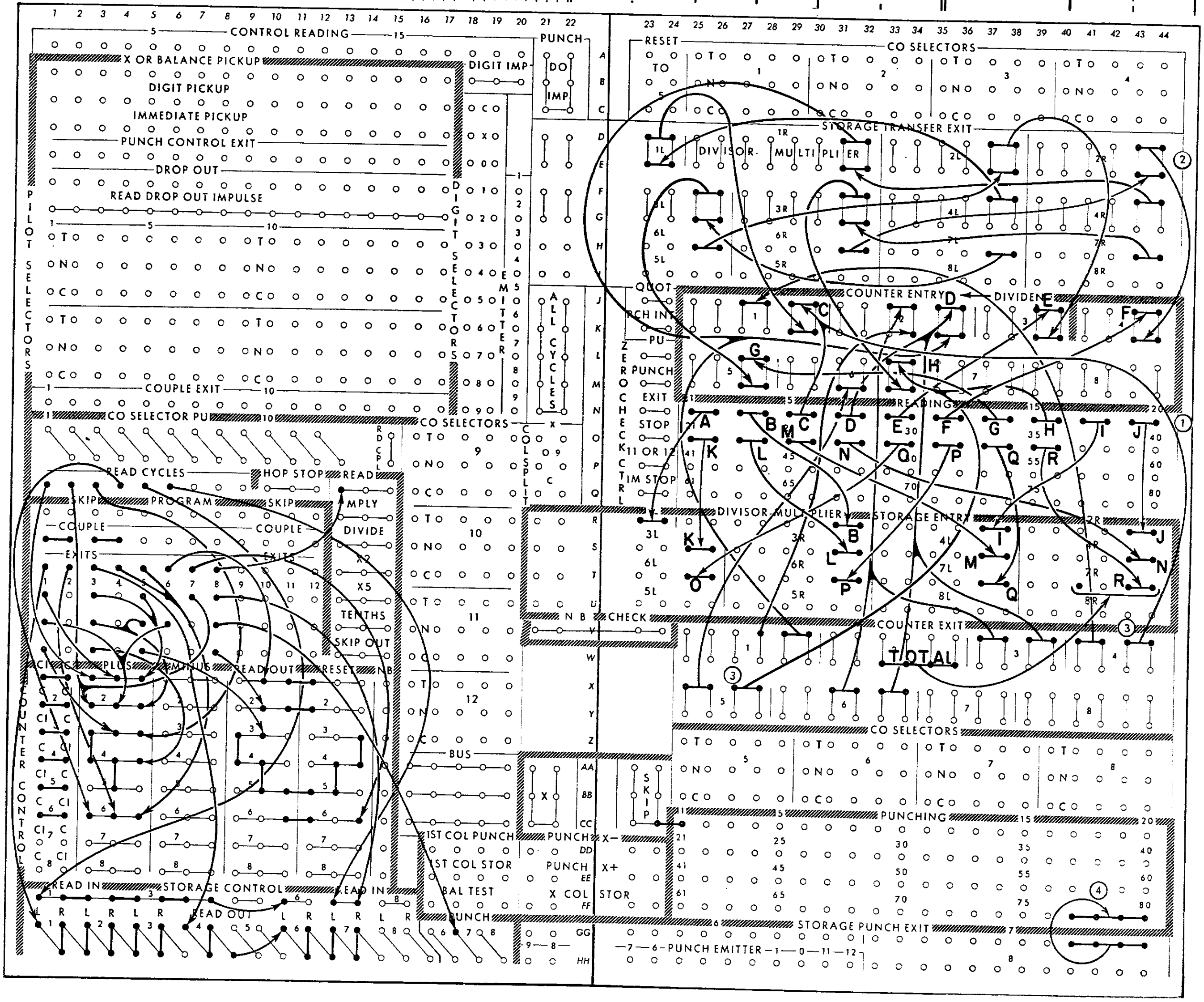


*Figure 49. Eighteen-Field Crossfooting*

## EIGHTEEN-FIELD CROSSFOOTING

BECAUSE there are 18 separate units in the machine, 18 fields can easily be crossfooted, as illustrated in Figure 49. With small fields, like those shown, it would actually be possible to store more than one field in each unit and, thus, crossfoot more than 18 fields; with large fields, however, fewer than 18 fields can be crossfooted. Thus, the number of fields that can be crossfooted depends on the size of the fields as well as the total number of digits that must be stored and accumulated.

### Planning Chart and Control Wiring

*Read.* Factors *A* through *R* enter all 18 units, which are impulsed on the read cycle. Independent read cycles impulses are wired to counters 1, 2, and 6 because these counters are impulsed to read, add, read out, and reset independently.

*Program 1 (Exits 1 and 2).* In order to expand the exit capacity of program 1, it is coupled with program 2. *A, B, I, J, K,* and *L* are read out of storage units 1, 2, and 3 and added into counters 1 through 6.

*Program 2 (Exits 3 and 4).* *M, N, O, P, Q,* and *R* are read out of storage units 4, 6, and 7 and added into counters 1 through 6. Program 2 exits are expanded by coupling program 3 with program 4.

*Program 3 (Exit 5).* The totals in counters 3, 4, and 5 are read out and added in counters 2, 1, and 6, respectively. Counters 3, 4, and 5 are reset.

*Program 4 (Exit 6).* The total in counter 1 is read out and added into counter 2. Counter 1 is reset.

*Program 5 (Exit 7).* The total in counter 6 is read out and added in counter 2. Counter 6 is reset.

*Program 6 (Exit 8).* The grand total in counter 2 is read out into storage unit 7 for punching. Counter 2 is also reset. READ is wired from this program which is the last one used for the problem.

### Right-Hand Panel Wiring

1. Factors *A* through *R* are wired from the reading hubs to counters and storage entry as shown.

2. Read-out for storage units 1L, 1R, 2L, 2R, 3L, 3R is wired to counters 1 through 6, respectively. Read-out for storage units 4L, 4R, 6L, 6R, 7L, and 7R is wired through the common hubs of storage units 1L, 1R, 2L, 2R, 3L, and 3R (to avoid split wires) to the same set of counters.

3. The sum of factors *E, I,* and *O* is read out of counter 3 into counter 2. The sum of factors *F, J* and *P* is read out of counter 4 into counter 1. The sum of factors *G, L,* and *Q* is read out of counter 5 into counter 6. The sum of factors *A, C, F, J, M,* and *P* is read out of counter 1 into counter 2. The sum of factors *G, H, K, L,* and *Q* is read out of counter 6 into counter 2. Counter 2 now contains the sum of the factors, and is wired to read into storage unit 7 for punching.

4. The total is punched from storage punch exit 7.

## SIGN CONTROL, THREE FACTORS

$$\pm A \times \pm B = \pm P \times \pm C = \pm R$$

IN ANY EQUATION involving sign control, an even number of minus signs in the factors will result in a positive answer, and an odd number of minus signs will result in a negative answer. For example,

| | |
|---|---|
| One minus sign | $+A \times -B \times +C = -R$ |
| Two minus signs | $-A \times +B \times -C = +R$ |
| Three minus signs | $-A \times -B \times -C = -R$ |

When there are no minus signs for the factors, the result is always positive.

The signs (X's or D's) are read at the reading station and wired to pick up pilot selectors which are used later in the problem for the selection of the sign of the product. The determination of the signs to be punched is ascertained by the position of the selectors at punching time.

In this problem (Figure 50), two results are calculated and punched: *P*, the result of *A* × *B*, and *R*, the result of *P* × *C*.

### Planning Chart and Control Wiring

*Read.* *A* is read into the multiplier and *B* and *C* into storage units 2 and 3, respectively. The sign of each factor is wired to pick up a pilot selector, as explained under "Right-Hand Panel Wiring."

*Program 1.* The multiply hub is impulsed, *B* is read out of 2R, and counter 5-6 is impulsed to add to develop the product *P* of *A* × *B*.

*Program 2.* *P* is read out of counter 5-6 into storage unit 6 for punching. *C* is read out of 3R into the multiplier storage unit.

*Program 3.* The multiply hub is impulsed, *P* is read out of counter 5-6, and counter 1-2 is impulsed

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT | | COUNTER | | | | | | STORAGE UNITS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | DIVR-MULT | DIVIDEND | | | | | | | | | | | | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH | | | |
| | | | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R | 7L | 7R |
| | | X or D to Control Sign<br>READ CYCLE | | A(±) | | | | | | | | B(±) | | C(±) | | | | | | |
| | 1 | Multiply | | | | | | | + | +P | | RO | | | | | | | | |
| | 2 | | | C(±) | | | | | RO | RO | | | | RO | | | | P(±)<br>Punch | | |
| | 3 | Multiply | | | + | +R | | | RO | RO | | | | | | | | | | |
| | | READ CYCLE | | | RO<br>RE | RO<br>RE<br>5 (Corr) | | | RE | RE<br>5 (Corr) | | | | | | | | | | R(±)<br>Punch |
| | 1 | | | | | | | | | | | | | | | | | | | |
| | 2 | | | | | | | | | | | | | | | | | | | |

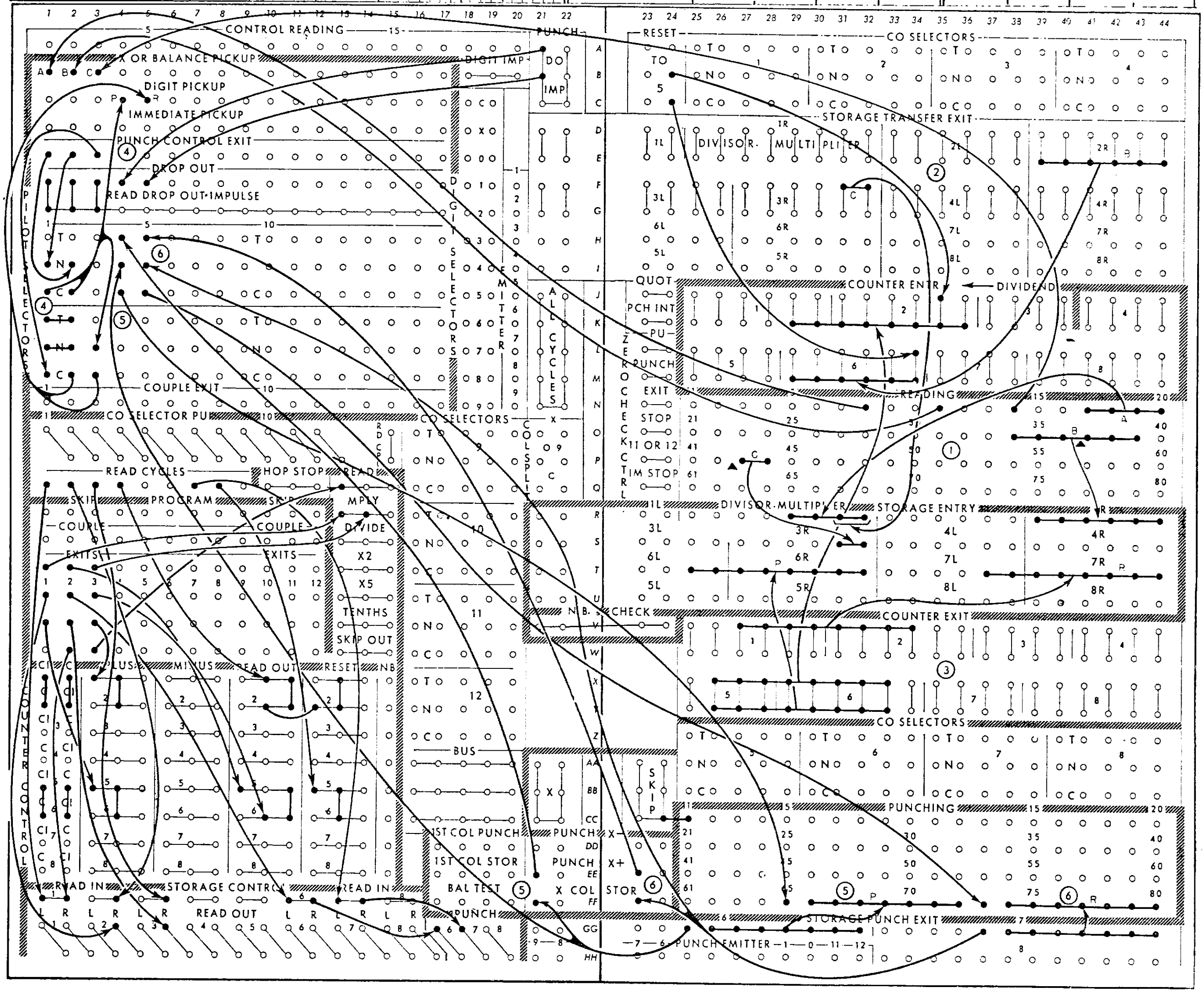


*Figure 50. Sign Control: Three Factors*
$\pm A \times \pm B = \pm P \times \pm C = \pm R$

to add to develop the product of $P \times C$ $(R)$. Since this is the last program used, READ is impulsed.

*Read.* $R$ is read out of counters 1 and 2 into storage unit 7 for punching. Counters 1-2 and 5-6 are reset. All counters can be reset on the read cycle because they are not being impulsed to add at this time.

### Right-Hand Panel Wiring

1. $A$, $B$, and $C$ are wired to storage units 1R, 2R, and 3R, respectively. Columns 8, 11, and 14, which contain the signs of $A$, $B$, and $C$, are wired from reading brushes to the X pickup of pilot selectors 1, 2, and 3. These selectors are dropped out normally by a read drop-out impulse.

2. $B$ is wired from 2R to counter 6, where the product of $A \times B$ is developed. Of the three decimal places in the product $P$, one is dropped, so reset to 5 is wired to the units position of counter 6. $C$ is wired from 3R to the multiplier storage unit 1R. Of the four decimal places in the product $R$, two are dropped, so reset to 5 is wired to the tens position of counter 2.

3. The adjusted product of $A \times B$ $(P)$ is wired from counter 5-6 to storage unit 6 for punching, and to counter 1-2 to be multiplied by $C$. The adjusted product of $P \times C$ $(R)$ is wired from counter 1-2 to storage unit 7 for punching.

4. Pilot selector 4 is used to select the X punching when $P$ is minus, and pilot selector 5 is used to select the X punching when $R$ is minus. Since this is punch selection, these selectors are picked up by the punch control exit, just before punching starts, and are dropped out by punch drop-out impulses after punching is completed.

A punch control exit emits an impulse only if its selector is transferred. The punch control exit impulse of pilot selector 1 ($A$ minus) is wired through the normal side of pilot selector 2 ($B$ plus); the punch control exit impulse of pilot selector 2 ($B$ minus) is wired through the normal side of pilot selector 1 ($A$ plus). If an impulse is available from the common hub of either selector, pilot selector 4 is picked up, indicating that $P$ is minus.

The impulse for minus $P$ is also wired through the normal side of pilot selector 3 ($C$ plus) to pick up pilot selector 5, indicating that $R$ is minus. The punch control exit of pilot selector 3 ($C$ minus) is wired through the lower positions of pilot selectors 1 and 2 in such a way that pilot selector 5 is picked up ($R$ minus) if the signs of $A$ and $B$ are alike.

5. The result $P$ is read out of storage punch exit 6 and punched in columns 65-72 with its sign in column 65. The digit and the X are punched in column 65 through the transferred side of pilot selector 4, which is picked up when $P$ is minus. The X impulse is obtained through X column storage. The digit alone is punched in column 65 through the normal side of pilot selector 4 when $P$ is plus.

6. The result $R$ is read out of storage punch exit 7 and punched in columns 73-80 with its sign in column 73. The digit and the X are punched in column 73 through the transferred side of pilot selector 5, which is picked up when $R$ is minus. The X impulse is obtained through X column storage. The digit alone is punched in column 73 through the normal side of pilot selector 5 when $R$ is plus.

## PAYROLL, HOURS AND PIECES TO GROSS

SOME payroll applications require the calculation of piecework earnings as well as hourly earnings to obtain gross pay. This problem demonstrates the calculation and punching of piecework earnings, regular earnings, premium earnings, and gross pay from a card containing number of pieces, piecework rate, hourly rate, and total number of hours worked (Figure 51). The problem may be stated as follows:

Piecework rate × Pieces = Piecework Earnings
Hourly Rate × Hours = Regular Earnings
Hours × .5 = ½ Overtime Hours
Hourly Rate × ½ Overtime Hours = Premium Earnings
Piecework Earnings + Regular Earnings + Premium Earnings = Gross Pay.

Calculation of net pay from gross pay is demonstrated in another problem.

### Planning Chart and Control Wiring

The preparation of the planning chart for problems of this type is facilitated if a typical example is carried through the complete operation using actual figures. The half-adjustment (reset to 5) is indicated on the read cycle line by the figure 5 encircled and properly positioned in counters 2-3 and 4-5. Actually, this adjustment does not occur on the read cycle. In this problem, the reset to 5 occurs on program 8 for counter 2-3 and program 5 for counter 4-5. For the first card, however, the reset to 5 occurs on the

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT | | COUNTER | | | | | | STORAGE UNITS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | DIVR.–MULT. | DIVIDEND | | | | | | | | | | | | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH | | | |
| | | | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L * | 4R | 6L | 6R * | 7L | 7R * |
| | | READ CYCLE | | *Pieces*<br>*000199* | *Hours*<br>*41.05* | | *5* | | *5* | | *P/w Rate*<br>*.12345* | *Hr. Rate*<br>*1.75* | *Hours*<br>*41.05* | | | | | | | |
| | 1 | *Mult.* | | | | *P/w Earnings*<br>*24.56655*<br>*24.57* | | | | | *RO* | | | | | | | | | |
| | 2 | *NB Test CTR 1 (Exits 2 & 3)* | | *Rate*<br>*1.75* | *-4000*<br>*1.05*<br>*O.T. Hrs* | *RO* | | | | | | *RO* | | | | | | *P/w Earn*<br>*24.57* | | |
| | 3 | *Mult. (Exit 4)* | | | | | | *Reg. Earn*<br>*71.8375*<br>*71.84* | | | | | *RO* | | | | | | | |
| *NB* | 4 | *Mult. X .5 (Exit 5)* | | | *RO* | | | | | *½ O.T. Hrs*<br>*.525* | | | | | | | | | | |
| | 5 | *(Exit 6)* | | | *RE*<br>*5* | *71.84* | | | *RO*<br>*RE*<br>*5* | | | | | | | | | | | *Reg. Earn*<br>*71.84*<br>*Punch* ↓ |
| | 6 | *Mult. (Exit 7)* | | | *Prem. Earn*<br>*.9100*<br>*.91* | | | | | $\boxed{\text{RO}}$ | | | | | | | | | | |
| | 7 | *(Exit 8)* | | | *RO* | *Gross Pay*<br>*.91*<br>*97.32* | | | | | | | | | | | | | | |
| | 8 | *(Exits 9 & 10)* | | | *RO*<br>*RE* | *RO*<br>*RE* | *5* | | | *RE* | | | | | | | | *Punch* ↓ | *Prem. Earn*<br>*.91*<br>*Punch* ↓ | *Gross*<br>*97.32*<br>*Punch* ↓ |
| | 9 | | | | | | | | | | | | | | | | | | | |

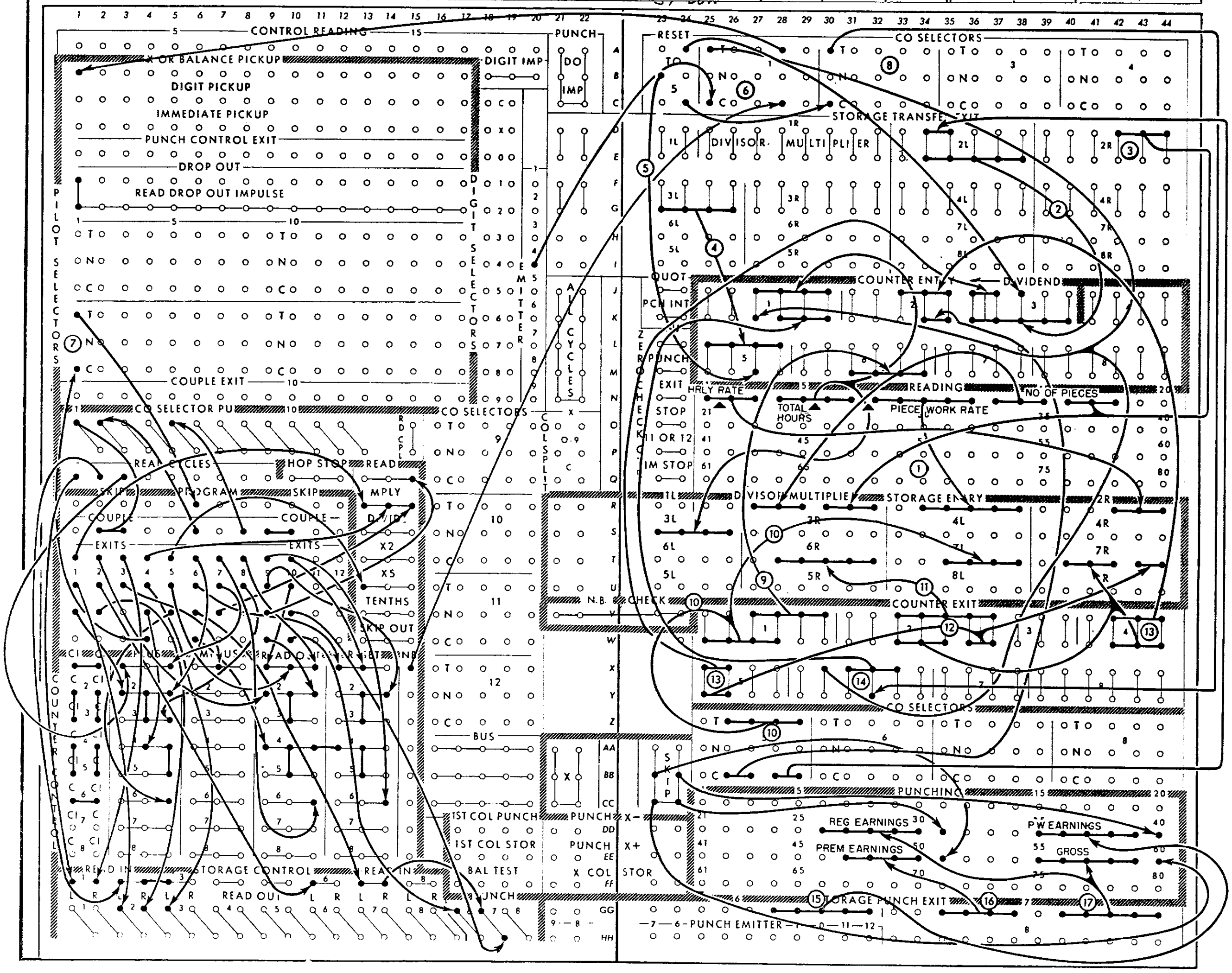


*Figure 51. Payroll, Hours and Pieces to Gross*

dummy programs preceding the read cycle. It is shown here encircled merely to assist in the manual computation of the problem as the program steps are developed.

*Read.* The number of pieces produced is entered into 1R, piecework rate into 2L, hourly rate into 2R, and total hours worked into 3L and counter 1.

*Program 1.* Piecework rate is multiplied by number of pieces, by impulsing the multiply hub, reading out 2L, and adding in counter 2-3 where the piecework earnings are developed.

*Program 2 (Exits 2 and 3).* The piecework earnings amount is read out of counter 2-3 into storage unit 6R where it will be held for punching until the last program. Although it is the first result calculated, it is not the first field to be punched, so it must be stored until after the punching of regular earnings. Counter 2-3 is not reset on this program as it is used to develop gross pay.

Hourly rate is read out of 2R and entered into the multiplier 1R. A 4 from the digit emitter (40.00 hour work week) is subtracted from total hours worked in counter 1. The remainder standing in that counter represents overtime hours. Co-selector 1 is picked up from the couple hubs of this program. The selector is used to select 40.00 hours from the digit emitter and to control the NB of counter 1 so that pilot selector 1 is picked up.

*Program 3 (Exit 4).* Total hours are multiplied by hourly rate to obtain regular earnings by impulsing the multiply hub, reading out of 3L, and adding in counter 4-5. This program is also wired through the controlled side of pilot selector 1 (picked up from the NB of counter 1) to the skip of program 5 so that program 4 is skipped when there are no overtime hours.

*Program 4 (Exit 5).* The overtime hours in counter 1 are multiplied by 5 to develop half the overtime hours in counter 6. This is accomplished by wiring program 4 (exit 5) to × 5 hours. Multiplying by 5 is the same as dividing by 2 when the decimal point in the result is offset one position. In this example, half the overtime hours is not adjusted. Circumstances in a given case would determine whether the adjustment is necessary.

*Program 5 (Exit 6).* Regular earnings amount is read out of counter 4-5 and added to piecework earnings in counter 2-3. It is also entered into storage unit 7 from which it is punched on the following cycle. Counter 4-5 is reset on this program, with the reset to 5 properly wired to adjust regular earnings on the following card. Counter 1 is reset on this program, with the reset to 5 properly wired to adjust premium earnings later in the problem. Co-selector 2 is picked up from the program couple to select a reset to 5 (wiring 8).

*Program 6 (Exit 7).* Half the overtime hours is multiplied by the rate by impulsing the multiply switch, reading out of counter 6 (units position dropped), and adding in counter 1 to develop premium earnings.

*Program 7 (Exit 8).* Premium earnings amount is read out of counter 1 and added into counter 2-3. The total represents gross pay. Co-selector 5 is picked up from the couple hubs of this program (wiring 10).

*Program 8 (Exits 9 and 10).* Counter 6 is reset. Gross pay is read out of counter 2-3 and entered into storage unit 7 for punching. Counter 2-3 is reset with the reset to 5 properly wired to adjust piecework earnings on the following card. Premium earnings amount is read out of counter 1 and entered into storage unit 7 for punching. Counter 1 is reset on this program. Program 8 (exit 10) is wired to the punch of 6 so that piecework earnings, stored in this unit on program 2, may be punched. The last program is always wired to read.

## Right-Hand Panel Wiring

1. Hourly rate is wired to 2R, total hours to 3L and counter 1, piecework rate to 2L, and the number of pieces to 1R.

2. Piecework rate is wired out of 2L into counters 2 and 3 to develop piecework earnings.

3. Hourly rate is read out of 2R into the multiplier 1R.

4. Total hours is wired out of 3L into counter 5 to develop regular earnings.

5. Reset to 5 is wired to the tens position of counter 5 to half-adjust regular earnings.

6. Co-selector 1 is picked up on program 2. The digit 4 from the emitter is wired through the controlled side of the selector to the thousands position of counter 1, as explained under program 2.

The NB of counter 1 is wired through the transferred side of co-selector 1 to the pickup of pilot selector 1.

7. Program 3 (exit 4) is wired through the transferred side of pilot selector 1 to the skip entry of program 5, thus skipping program 4 when counter 1 is negative.

8. Co-selector 2 is picked up on program 5 (exit 6 couple). Reset to 5 is wired through its transferred side to the tens position of counter 1, by way of the common exits of counter 6 to avoid split wires. The reset to 5 is selected in order to adjust only the premium earnings.

9. Overtime hours are wired out of counter 1 exit to counter 6 entry to develop half the overtime hours.

10. Premium earnings amount, with two places dropped, is wired out of counter 1 exit to 7L for punching. It is also wired through the transferred side of co-selector 5 to counter 2-3 entries, shifted three positions to the left to line up with the decimal for the total already standing in those counters. To avoid split wires, the two right-hand positions of co-selector 5 common are wired, through the fourth and fifth common hubs of 2L exit, to counter 2-3. The field is wired through the selector to prevent premium earnings from also reading into 7R, and gross pay into 7L, on program 8.

11. Adjusted piecework earnings amount is wired out of counter exits 2-3 to 6R for punching.

12. Adjusted gross pay is wired out of counter exits 2-3 to 7R for punching.

13. Adjusted regular earnings amount is wired out of counter 4-5 to 7R for punching, and to counter 2-3 to be added to piecework earnings.

14. One-half the overtime hours with one decimal dropped is wired out of counter 6 exit to counter 1 to develop premium earnings.

15. Piecework earnings amount is wired from storage punch exit 6 to punch in columns 35-39.

16. Premium earnings amount is wired out of storage punch exit 7 to punch in columns 47-50.

17. Regular earnings and gross pay are split-wired out of storage punch exit 7 to punch in columns 26-30 and 55-59, respectively. SKIP is wired after each field punched.

## PAYROLL, GROSS TO FIRST NET

PAYROLL applications on the 602 vary considerably, depending upon the industry, location, and tax rates in effect. (The rates used in this problem are not necessarily those currently in effect.) This problem (Figure 52) describes one method of computing:

Withholding tax (18% of taxable earnings).

State Unemployment Insurance Tax (2% of gross earnings up to $3,000).

Federal Insurance Contribution Act (2¼% of gross earnings up to $4,200).

Net Pay (gross earnings minus the three taxes above).

The following factors are prepunched in the card:

Gross Earnings (current period).

Earnings-to-Date (end of previous period).

Tax Allowance.

The tax allowance punched in the card is 18% of the exemption amount. The exemption amount and the tax allowance for each dependent are as follows:

| PAY PERIOD | EXEMPTION AMOUNT | TAX ALLOWANCE |
|---|---|---|
| Weekly | $13.00 | $ 2.34 |
| Bi-weekly | 26.00 | 4.68 |
| Semi-monthly | 28.00 | 5.04 |
| Monthly | 56.00 | 10.08 |

In cases where the tax allowance exceeds the tax computation, the withholding tax is not deducted.

The payroll cards are punched X in 78 when earnings are $3,000 or over, and X in 76 when earnings are $4,200 or over. SUI tax is not calculated for X 78 cards; FICA tax is not calculated for X 76 cards. Provision is also made to punch either or both X's when the limits are exceeded during the current period.

The method illustrated can be adapted to varying tax rates with little change in control panel wiring.

Figures for a typical case (gross earnings $175.85, earnings to date $2,900, and tax allowance $5.04) are used to develop the planning chart and assist with the understanding of the problem.

### Planning Chart and Control Wiring

*Read Cycle.* Since withholding tax is the first calculation to be made, withholding tax rate of .180 is entered in the multiplier storage unit from the digit emitter.

Earnings to date are subtracted in counter 1 and counter 2 from the taxable limits of $3,000 and $4,200, respectively. For the first card to be calculated these limits are entered on the last "dummy" run-in cycle; for all other cards they are entered on the last program of the preceding card.

Counter 1 is used for two purposes:

1. To determine the taxable balance for the year subject to SUI tax — $100 in this example.

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT | | COUNTER | | | | | | STORAGE UNITS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | DIVR.-MULT. | DIVIDEND | | | | | | | | | | | | PUNCH UNITS<br>*UNITS POSITION MUST BE WIRED TO PUNCH | | | |
| | | | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R * | 7L | 7R * |
| | | READ CYCLE | | .180 | EARNINGS TO DATE<br>3000.00<br>1000.00 | EARNINGS TO DATE<br>4200.00<br>1300.00 | | | TAX ALLOWANCE<br>-504 ⑤<br>50.35 | | | GROSS<br>175.85 | | | | | | | | |
| | 1 | Multiply W. TAX | | | | | | | WITHOLDING TAX<br>31.65300<br>26.61 | | | RO | | | | | | | | |
| | 2 | NB TEST Ctr. 5 | | | RO | RO | GROSS<br>175.85 | | | | | RO | SUI TAXABLE BALANCE<br>100.00 | | FICA TAXABLE BALANCE<br>1300.00 | | | | | |
| | 3 | NBTEST ctrs. 1&2<br>Pch X78 if ctr 1-⊖<br>Pch X76 if ctr 2-⊖ | | SUI RATE<br>.020 | -175.85<br>- 75.85 | -175.85<br>1124.15 | WITHOLD TAX<br>149.24 | | RO<br>RE | F ⊕<br>5 | | RO | | | | | | WITH TAX PUNCH ↓<br>26.61 | | |
| X 78 | 4 | Multiply SUI (EXIT 5) | | | | | | | | SUI<br>200000<br>2.00 | | RO if ctr.1 ⊕ | RO if ctr.1 ⊖ | | | | | | | |
| | 5 | EXITS 6 and 7 | | .0225 | RE | RE | SUI<br>- 2.00<br>147.24 | | RO | RE<br>5 | | | | | | | | SUI PUNCH ↓<br>2.00 | | |
| X 76 | 6 | Multiply FICA (EXIT 8) | | | | | | | FICA<br>3.956625<br>3.96 | | | RO if ctr 2 ⊕ | | | RO if ctr. 2 ⊖ | | | | | |
| | 7 | EXITS (9 and 10) | | | 3000.00 | 4200.00 | FICA<br>-3.96<br>143.28 | | RO | RE<br>5 | | | | | | | | FICA PUNCH ↓<br>3.96 | | |
| | 8 | READ CYCLE | | | | | RO | RE | | | | | | | | | | | | NET PUNCH ↓<br>143.28 |

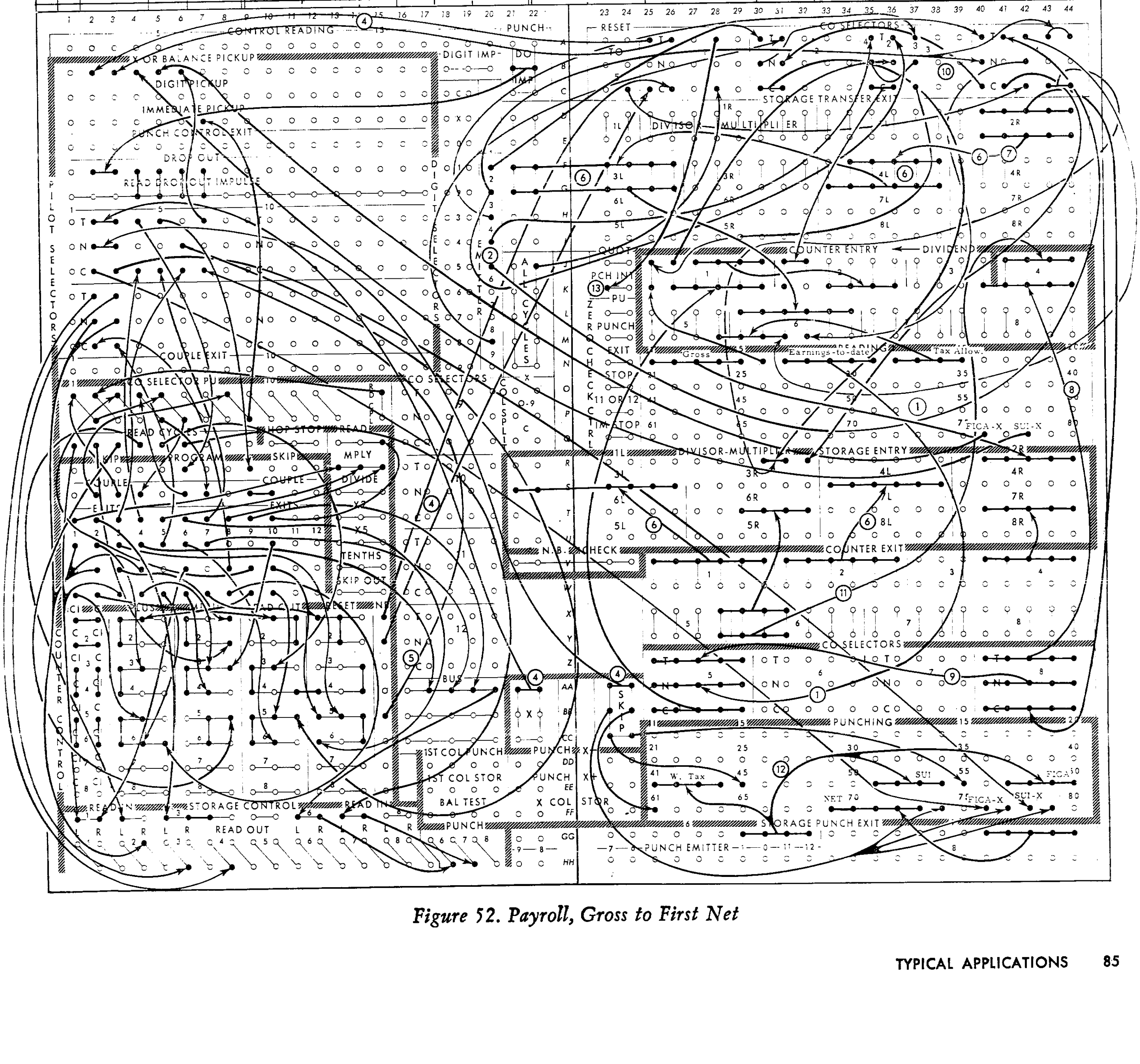


*Figure 52. Payroll, Gross to First Net*

2. As a testing counter to determine whether the current gross earnings ($175.85) is greater than the taxable balance ($100). This condition will be tested on a later program when the gross is deducted from the balance now in the counter. At that time, if the result in this counter is plus, the SUI tax is computed on the gross earnings; if the result is minus, the SUI tax is computed on the taxable balance, and an X is punched in column 78 because the $3,000 limit has been exceeded.

Counter 2 is used for two purposes:

1. To determine the taxable balance for the year subject to FICA tax — $1,300.00 in this example.
2. As a testing counter to determine whether the current gross earnings ($175.85) is greater than the taxable balance ($1,300). This condition will be tested on a later program when the gross is deducted from the balance now in the counter. At that time, if the result in this counter is plus, the FICA tax will be computed on the gross earnings; if the result is minus, the FICA tax will be computed on the taxable balance, and an X will be punched in column 76 because the $4,200 limit has been exceeded.

Counter 5-6 is used to compute the withholding tax and as a testing counter to determine whether the tax allowance is greater than 18% of the gross earnings. It also is used later to compute SUI and FICA taxes. On the read cycle the tax allowance is subtracted in counter 5-6 from the half-adjustment entered in the counter on the last "dummy" run-in cycle for the first card, and on the last program cycle of each preceding card. This reset to 5 is necessary to adjust the withholding tax.

Gross earnings ($175.85) is entered in storage unit 2R and is used as explained in programs 1, 2, 3, 4, and 6.

*Program (Exit 1).* Withholding tax is computed in counter 5-6 by impulsing multiply, reading out 2R, and adding in counter 5-6. A plus result in this counter represents the withholding tax on the taxable earnings. A minus result indicates that the tax allowance is greater than 18% of the gross earnings. On a later program this counter will be impulsed to read out only if the result is plus.

*Program 2 (Exit 2).* Counter 5-6 is tested to determine the above condition. The SUI taxable balance is read out of counter 1 and stored in 3L. The FICA taxable balance is read out of counter 2 and stored in 4L. This transfer from counters to storage units is necessary because the taxable balances may have to serve as multiplicands based on minus conditions detected in counter 1-2. Gross is read out of counter 2R and added in counter 3-4 which is used to develop the net pay.

*Program 3 (Exits 3 and 4).* The SUI rate (.020) is entered in the multiplier storage unit from the digit emitter. Gross is read out of 2R and subtracted in counter 1 and counter 2. Each counter is tested as explained under "Read." If counter 5-6 is plus, the withholding tax is subtracted from gross in counter 3-4 and entered in 6R for punching. If counter 5-6 is minus, the amount is not read out, and zeros will punch in the SUI field. Counter 5-6 is reset to 5 in the third position to half-adjust the SUI calculation to follow.

*Program 4 (Exit 5).* If the $3,000 SUI limit was reached in a previous pay period, the card was punched X in column 78 at that time, and SUI tax is not to be computed. Therefore, program 4 is skipped for all X78 cards. SUI tax is to be computed on NX78 cards only. In this case the tax is developed in counter 5-6 by impulsing multiply, adding in counter 5-6, and reading out of 2R or 3L. The gross in 2R is read out if counter 1 was plus on program 3; the taxable balance in 3L is read out if counter 1 was minus on program 3.

*Program 5 (Exits 6 and 7).* The FICA tax rate (.0225) is entered from the digit emitter in the multiplier storage unit. Counter 1-2 is reset. The SUI tax is read out of counter 5-6, subtracted in counter 3-4, and entered in 6R for punching. If program 4 was skipped because of an X in 78, zeros will punch in the SUI field. Counter 5-6 is reset to 5 in the fourth position to half-adjust the FICA calculation to follow.

*Program 6 (Exit 8).* If the $4,200 FICA limit was reached in a previous pay period, the card was punched X in column 76 at that time, and the FICA tax is not to be computed. Therefore, program 6 is skipped for all X76 cards. FICA tax is to be computed on NX76 cards only. In this case the tax is developed in counter 5-6 by impulsing multiply, adding counter 5-6, and reading out of 2R or 4L. The gross in 2R is read out if counter 2 was plus on program 3; the taxable balance in 4L is read out if counter 2 was minus on program 3.

*Program 7 (Exits 9 and 10).* The FICA tax is read out of counter 5-6, subtracted in counter 3-4, and entered in 6R for punching. If program 6 was skipped because of an X in 76, zeros punch in the FICA field. The $3,000 and $4,200 taxable limits are entered in counters 1 and 2, respectively, from the digit emitter, and counter 5-6 is reset to 5 in preparation for the next card. This completes the problem, and program 7 is wired to READ.

*Read Cycle.* Counter 3-4 is impulsed to read out and reset, and the net amount is transferred to 7R for punching.

## Right-Hand Panel Wiring

1. Gross earnings are wired to storage unit 2R. The earnings-to-date field is wired through the normal side of co-selector 5 to counter 1-2. The purpose of the co-selector is to prevent the net from reaching this counter on the read cycle. Tax allowance is wired to counter 6, offset three positions, to line up with the decimal point.

The X in column 76, which indicates that the earnings to date have previously exceeded the $4,200 FICA limit, is wired to the X pickup of pilot selector 5. This selector is used to skip program 6 whenever the $4,200 FICA limit has previously been exceeded.

The X in column 78, which indicates that the earnings to date have previously exceeded the $3,000 SUI limit, is wired to the X pickup of pilot selector 4. This selector is used to skip program 4 whenever the $3,000 SUI limit has previously been exceeded.

2. The withholding tax, SUI, and FICA rates are entered into the multiplier unit from the emitter. The withholding tax rate enters on the read cycle through the transferred side of co-selector 2, the SUI rate on program 3 through the normal side of co-selectors 2 and 4, and the FICA rate on program 5 through the transferred side of co-selector 4 and the normal side of co-selector 2.

3. Reset to 5 is wired to the third position of counter 6 to adjust the withholding tax.

4. On NX78 cards, if counter 1 is negative at the end of program 3, SUI is computed on the taxable balance for the year. If it is positive, the SUI computation is based on the current gross. On NX76 cards, if counter 2 is negative at the end of program 3, FICA is computed on the taxable balance for the year. If it is positive, the FICA computation is based on the current gross. To determine the negative or positive condition of counter 1-2, the NB hubs of these counters are wired through the transferred side of co-selector 1 to the balance pickups of pilot selectors 3 and 2, respectively. The purpose of the pilot selectors is to punch an X in column 76 when counter 2 is negative and in column 78 when counter 1 is negative. When the counters are positive, skip impulses reach columns 76 and 78. The selectors also serve to select either the gross or the SUI taxable balance on program 4, and the gross or the FICA taxable balance on program 6.

5. If counter 5-6 is negative at the end of program 2, the tax allowance is greater than the tax computation, and no withholding tax is to be deducted. The NB hub of counter 5 is wired through the transferred side of pilot selector 7 to the balance pickup of pilot selector 6. Pilot selector 7 is controlled on program 2. If the withholding tax counter 5-6 is plus, it is controlled to read out for punching on program 3, through the normal side of pilot selector 6.

6. The SUI taxable balance is stored in 3L on program 2. The SUI tax is developed in counter 5-6 by reading out 3L (taxable balance) if counter 1 is minus or 2R (gross) if counter 1 is plus.

The FICA tax is developed in counter 5-6 by reading out 4L (taxable balance) if counter 2 is minus, or 2R (gross) if counter 2 is plus.

7. Storage exit 2R is wired to counter 5-6 entries to allow the withholding tax to be computed.

8. Gross is wired to counter 3-4 through the transferred side of co-selector 8. The co-selector is picked up on program 2. The selector is needed to prevent the gross from entering counter 3-4 on program 3.

9. The gross from 2R enters counter 1-2 through the normal side of co-selector 8 and the transferred side of co-selector 5. Co-selector 5 is picked up on program 3.

10. The SUI tax limit is entered into counter 1, and the FICA tax limit into counter 2 on program 7. Both limits come from the emitter and are controlled through the transferred side of co-selector 3.

11. The withholding tax is wired from counter 5-6 to counter 3-4 where it is subtracted from the gross on program 3.

12. The withholding tax, SUI tax and FICA tax are wired out of counter 5-6, with three positions dropped, to 6R for punching. Net pay is wired out of counter 3-4 to 7R for punching. When the FICA

tax is exceeded, an X is punched in column 76. When the SUI tax is exceeded, an X is punched in column 78.

13. Since the X punching in columns 76 and 78 depends upon a condition arising during calculation (program 3), punch interlock must be impulsed on the same program on which the test is made. An all-cycles impulse reaches punch interlock on program 3, through the transferred side of co-selector 1.

## PAYROLL, HOURS TO NET

THE PREVIOUS problem (gross to first net) illustrates the method of calculating net pay when gross earnings are punched in the card. This problem (Figure 53) goes further in that the gross earnings, as well as net earnings-to-date, must be calculated. Specifically, the employee's earnings summary card is punched with rate, premium hours, total hours, tax allowance, miscellaneous deductions, and previous earnings-to-date. Calculations to be performed and punched are premium earnings, gross earnings, FICA, withholding tax, net pay, and new earnings-to-date. The punching of new earnings-to-date eliminates the necessity of summarizing previous earnings-to-date summary cards with current gross earnings.

The punching of the X when earnings-to-date exceed $4,200 is controlled in the same manner as illustrated in the previous problem. The FICA tax is also computed in the same manner; i.e., 2.25% of gross earnings, if earnings-to-date do not exceed $4,200, or a special computation when the $4,200 is exceeded during the current period. The withholding tax allowance ($7.02) is punched in the card and represents the tax deduction for three people at the weekly rate of $2.34 per person (13.00 × 18%).

### Planning Chart and Control Wiring

*Read.* Read cycles impulses are wired to storage units 1, 2, 3, and 4 and to the minus of counters 2 and 5-6. Rate is entered in the multiplier, premium hours and total hours in storage unit 2, tax allowance and miscellaneous deductions in storage unit 3, and previous earnings-to-date in storage unit 4. Previous earnings-to-date are also subtracted in counters 2 and 5-6. These two counter groups contain the $4,200 FICA limit.

*Program 1.* The multiply hub is impulsed, and regular earnings (rate × total hours) and premium earnings (rate × premium hours) are computed simultaneously. Premium hours are read out of 2L to develop premium earnings in counter 1 which is impulsed to add. Total hours are read out of 2R to develop regular earnings in counter 3-4 which is impulsed to add. Both amounts are adjusted by a reset to 5 previously entered in the tens position of the counters.

*Program 2.* Adjusted premium earnings are read out of counter 1 to storage unit 6 for punching and to counter 3-4 to add to regular earnings. The result then standing in counter 3-4 is gross earnings. Counter 1 is reset on this program.

*Program 3.* Gross earnings amount is read out of counter 3-4 to storage unit 7 for punching, to counter 1, where it is added, and to counter 5-6 where it is subtracted from the remaining FICA taxable amount (difference between $4,200 and previous earnings-to-date). If the balance in counter 5-6 is positive, the full gross amount is taxable. If counter 5-6 is negative, the difference standing in counter 2 is the FICA taxable balance.

To determine the positive or negative condition of counter 5-6 on program 3, the NB of counter 5 is wired through the transferred side of pilot selector 3 to the balance pickup of pilot selector 1. Pilot selector 3 is picked up through its immediate pickup. Thus, pilot selector 1 transfers on program 3, provided counter 5-6 is negative. Pilot selector 3 is dropped out normally, and pilot selector 1 is dropped out from a punch drop-out impulse.

When a pilot selector is transferred through its immediate pickup, the selector holds for the remainder of the cycle and drops out automatically. Although drop-out is not necessary, when selectors are picked up immediately, it is advisable to drop out all selectors used. This is because pilot selectors picked up with X or D do not drop out when the last card is calculated and, consequently, they remain in a latched position for the next operation until drop-out is impulsed.

The purpose of pilot selector 1 is to control the reading from counter 5-6 to counter 2, as explained under program 4, and to select the X if counter 5-6 is negative or to skip if the counter is positive. Since this is punch selection controlled during calculation, punch drop-out and punch interlock are necessary. Punch interlock is impulsed on program 3.

Program 3 exits are expanded by wiring couple 3 with couple 4. Pilot selector 3 is picked up on program 3 couple to control the NB of counter 5.

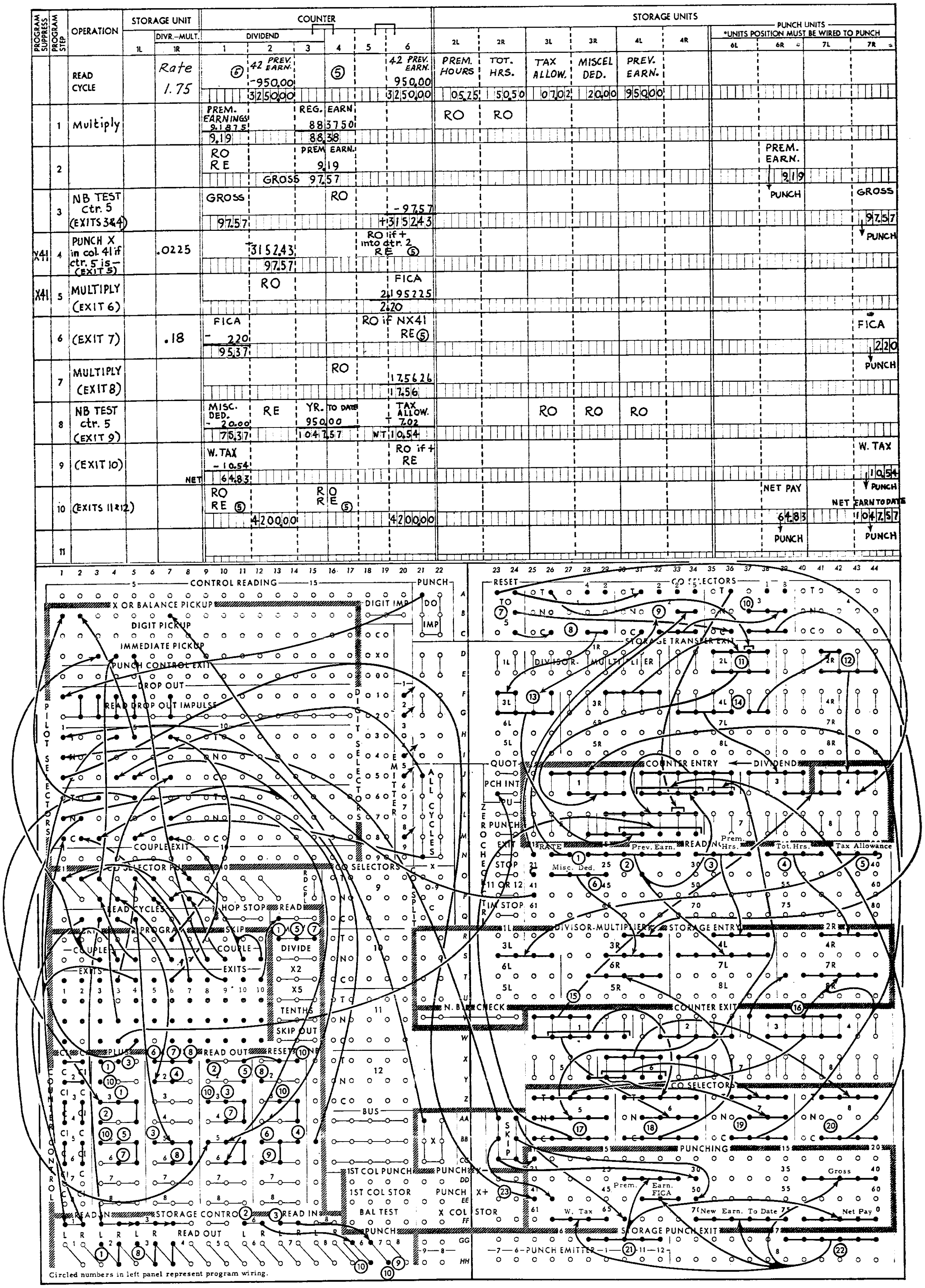


*Figure 53. Payroll, Hours to Net*

*Program 4.* If the total in counter 5-6 is positive, it is read out and subtracted in counter 2. Program exit 4 is wired through the normal side of pilot selector 1 to the read-out of counter 5-6. Counter 2 is wired to subtract, and 2.25% FICA tax rate is read into the multiplier unit. Counter 5-6 is reset.

If counter 5-6 is negative, the $4,200 tax limit has been exceeded, and an X is punched in column 41 through the transferred side of pilot selector 1. A skip is wired through the normal side. The X is used to suppress programs 4 and 5 (FICA calculations) for subsequent weeks. Co-selectors 2 and 8 are picked up from program-4 couple for selection described under "Right-Hand Panel Wiring."

*Program 5.* The FICA taxable amount in counter 2 is multiplied by 2.25% by reading out of counter 2 and impulsing the multiply hub. The product is developed in counter 5-6. This program is suppressed for X41 cards.

*Program 6.* The FICA tax is read out of counter 5-6 for all NX41 cards into counter 1, to be subtracted from the gross, and into storage unit 7 to be punched. Counter 5-6 is reset. Co-selector 3 and pilot selector 4 are picked up from program 6 couple. Pilot selector 4 is used for expanding program 6 exits. Withholding tax rate of 18% is entered into the multiplier unit.

*Program 7.* The gross in counter 3-4 is read out and multiplied by 18%. The product is developed in counter 5-6. One of the exits of program 7 is used to pick up co-selector 5 along with program 3. Although a selector is normally picked up from a program couple, picking up one selector from two program couples would couple the two programs and nullify those in between.

*Program 8.* The following operations take place on this program step:

1. The tax exemption amount is read out of 3L and subtracted from the gross × 18% in counter 5-6. A test is made at this time to determine the positive or negative condition of counter 5. Any plus amount left in the counter represents the withholding tax. If the counter is negative, no withholding tax is required.
2. Previous earnings are read out of 4L and added to the gross in counter 3-4 to obtain year-to-date earnings.
3. Counter 2 is reset.
4. Miscellaneous deductions read out of 3R and subtract from the gross in counter 1.

Pilot selector 5 is picked up from program 8 couple. The selector is used for expanding program exits and for controlling the NB of counter 5. To determine the positive or negative position of counter 5 on program 8, the NB of counter 5 is wired through the transferred side of pilot selector 5 to the balance pickup of pilot selector 2. Pilot selector 5 is picked up through its immediate pickup.

*Program 9.* If counter 5-6 is plus, the withholding tax is read out to storage unit 7 for punching and to counter 1 to be subtracted from the gross. The read-out of counter 5-6 is controlled through the normal side of pilot selector 2. Counter 5-6 is reset. Co-selector 7 is picked up from program 9 couple.

*Program 10.* The following operations take place on this program step:

1. Net pay is read out of counter 1 to storage unit 6 for punching. The counter is reset.
2. The $4,200 FICA limit is added into counters 2 and 5-6 from the digit emitter.
3. The year-to-date earnings in counter 3-4 is read out to storage unit 7 for punching.
4. The last program in a problem is always wired to read.

Co-selectors 1 and 6 are picked up from program 10 couple. The exits of program 10 are expanded by joining couple exits 11 and 12.

### Right-Hand Panel Wiring

1. Rate is wired to the multiplier (1R) on the read cycle through the normal side of co-selectors 2 and 3. Withholding tax percent is wired through the transferred side of co-selector 3 which is picked up on program 6 and FICA tax percent is wired through the transferred side of co-selector 2, which is picked up on program 4.

2. The previous earnings-to-date field is wired to counter entry 2 and through its common hubs to the entry of counter 6; it is also wired to storage entry 4L.

3. Premium hours are wired to storage entry 2L.

4. Total hours are wired to storage entry 2R.

5. Tax allowance is wired to storage entry 3L.

6. Miscellaneous deductions amount is wired to storage entry 3R.

7. Reset to 5 is wired as follows:

   a. To the tens positions of counter 6, through the transferred side of pilot selector 3, picked up on program 6. The purpose of this wiring is to adjust the withholding tax computation.

   b. To the tens position of counter 4 to adjust regular earnings.

   c. To the tens position of counter 1, through the transferred side of co-selector 1 and the common hubs of 2L exit. Co-selector 1 is picked up on program 10. The purpose of this wiring is to adjust premium earnings.

   d. To the thousands position of counter 6, through the transferred side of co-selector 2 and the common hubs of 3L exit. Co-selector 2 is picked up on program 4. The purpose of this wiring is to adjust FICA.

8. An emitted 42 is wired through the transferred side of co-selector 1 to the two high-order positions of counters 2 and 6. The co-selector is picked up on program 10 and allows the FICA taxable limit to enter counters 2 and 6 on the last program of each calculation.

9. The FICA tax percent is wired through the transferred side of co-selector 2 to the multiplier unit, as already explained.

10. The withholding tax percent is wired through the transferred side of co-selector 3 to the multiplier unit, as already explained.

11. Premium hours are wired from the exit of 2L to counter 1 entry so that they may be multiplied by rate. Miscellaneous deductions from 3R exit are wired to counter 1 through the common hubs of 2L to avoid split wires.

12. Total hours are wired out of 2R exit to counter 3-4 so that they may be multiplied by rate. The units and tens positions of 4L are wired through the third and fourth common hubs of 2R to counter entries 3-4 to line up with the decimal point.

13. The tax allowance amount is wired out of 3L exit to the entry of counter 6, offset two positions to line up with the decimal point in that counter. The emitted 42 and half-adjustment reach counter 6 entry through the common hubs of 3L.

14. The previous earnings-to-date field is wired out of 4L to the entry of counter 3-4.

15. The premium earnings field with two positions dropped is wired to counter 3-4 entry to add to regular earnings.

16. The gross from counter 3-4 is wired to counter 1 entry and to storage unit 7 for punching.

17. Gross is read out of counter 3-4 exits to counter 6 entry through the transferred side of co-selector 5 on programs 3 and 7. This taxable FICA amount is wired out of counter 2 through the normal side of co-selector 5 to counter 6 entry to develop the FICA tax. The purpose of co-selector 5 is to prevent interference between counters 1 and 6 on program 8.

18. The premium earnings field with two positions dropped is wired out of counter 1 through the normal side of co-selector 6 to storage unit 6 for punching. Net pay with no positions dropped is wired out of counter 1 through the transferred side of co-selector 6 to storage unit 6 for punching. Co-selector 6 is picked up on program 10 to select premium earnings and net pay.

19. FICA with four positions dropped is wired out of counter 6 through the normal side of co-selector 7 to counter 1 entry and storage unit 7. Withholding tax with two positions dropped is wired out of counter 6 through the transferred side of co-selector 7 to counter 1 entry and storage unit 7. Co-selector 7 is picked up on program 9; it is needed because the number of dropped positions is not the same for both FICA and withholding tax.

20. The four high-order positions of previous earnings-to-date are wired out of counter 6 exit through the transferred side of co-selector 8 to the corresponding positions of counter 2 entry. The units and tens positions are wired directly. Co-selector 8 is picked up on program 4 and is used to prevent interference between counters 1 and 2 on program 8.

21. Premium earnings and net pay are wired out of storage punch exit 6.

22. Gross earnings, FICA, withholding tax and new earnings-to-date are wired from storage punch exit 7.

23. Whenever new earnings-to-date exceed $4,200, an X is punched in column 41 by the X wired to the transferred side of pilot selector 1. When earnings are less than $4,200, column 41 is skipped by a skip impulse wired to the normal side. Common is wired to column 41 punch. Pilot selector 1 is picked up from the NB of counter 5 wired through the transferred side of pilot selector 3 on program 3. Whenever counter 1 is negative ($4,200 limit exceeded), an X will punch in column 41, and, whenever it is positive, that column will be skipped.

Since this is punch selection controlled during calculation, punch interlock is wired from program 3.

## ROUNDING TO NEAREST NICKEL

Half-adjustment is normally used to round a calculated result to the nearest cent, dime, or dollar. In problems such as premium calculations there is often a need to round the calculated result to the nearest nickel (Figure 54).

| CALCULATED RESULTS | PUNCHED RESULTS |
|---|---|
| 25.175 to 25.224 | 25.20 |
| 25.225 to 25.274 | 25.25 |
| 25.275 to 25.324 | 25.30 |

The operation may be done on the 602 by the following method:

1. Reset the mill position of the product counter to 5.
2. Calculate the product.
3. Analyze the cent position of the product with a digit selector.
4. If the cent position is 0, 1, or 2, transfer the dollar and dime positions from the product counter to a punch storage unit (offset one position).
5. If the cent position is 3, 4, 5, 6, or 7, transfer the dollar and dime positions as above, but also read an emitted 5 into the units position of the storage unit.
6. If the cent position is 8 or 9, take a program step, add an emitted 1 to the dime position of the counter, and transfer the dollar and dime positions to storage.

The additional program step is required only when the cent position of the product counter is 8 or 9.

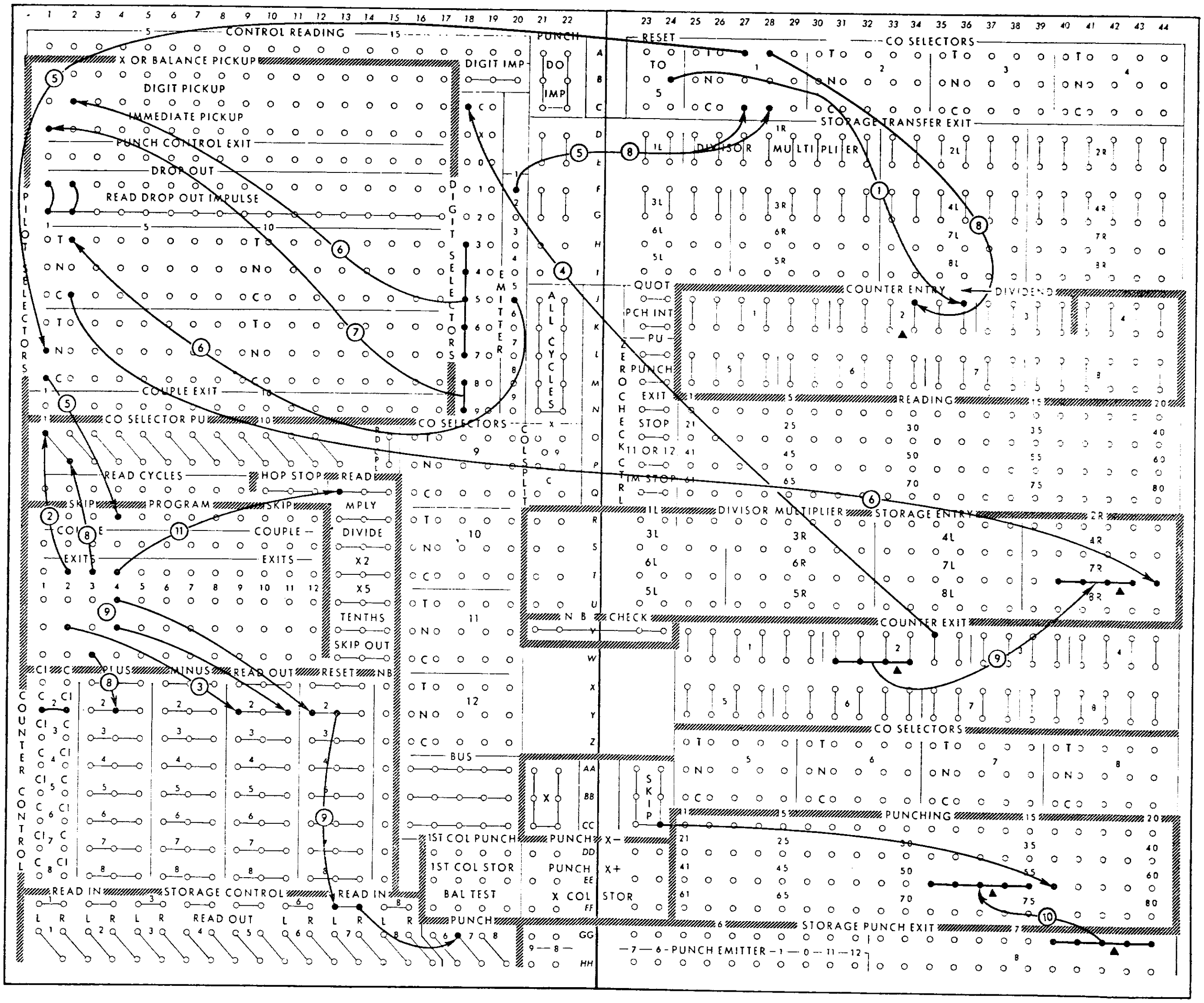


*Figure 54. Rounding to Nearest Nickel*

At all other times, the program step can be skipped.

Only the wiring pertinent to rounding to the nearest nickel is shown in Figure 54.

1. The mill position of the product counter is reset to 5.

2. Co-selector 1 is transferred on program 2.

3. The product is read out on program 2.

4. The cent position of the product is analyzed through a digit selector.

5. If the cent position is not 8 or 9, an emitted 1 impulsed through the transferred side of co-selector 1 and the normal side of pilot selector 1 causes programming to skip from program 2 to program 4.

6. If the cent position is 3, 4, 5, 6, or 7, pilot selector 2 is picked up and an emitted 5 is entered into the units position of the storage unit when the product is stored.

7. If the cent position is 8 or 9, pilot selector 1 is transferred immediately and program 3 is not skipped.

8. On program 3 (taken only if the cent position is 8 or 9), co-selector 1 is again transferred and an emitted 1 is added to the dime position of the product counter through the transferred side.

9. The product counter is wired to read out and reset and storage unit 7 is wired to read in and punch on program 4. The dollar and dime positions of the counter are transferred to the punch storage unit.

10. The rounded result is punched from storage.

11. A new card is fed.

## CALCULATING SQUARE ROOTS

The approximate square root of any number up to twelve digits can be calculated automatically (Figure 55). A mathematical method known as succes-

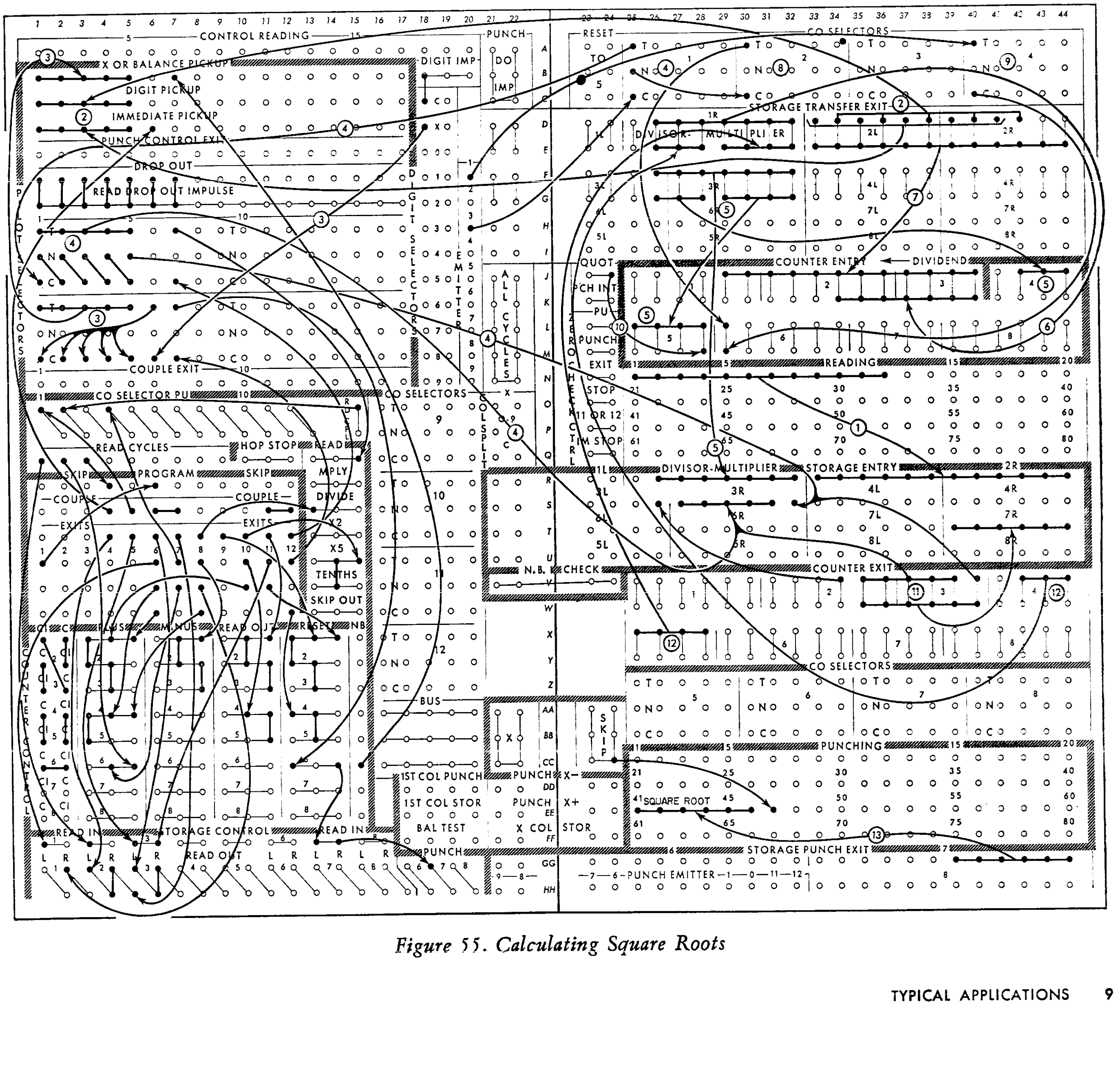


*Figure 55. Calculating Square Roots*

sive approximation is used. The formula for calculating square root is expressed as follows:

First Iteration $\left(\frac{B}{A_1} + A_1\right) \times .5 = A_2$

Second Iteration $\left(\frac{B}{A_2} + A_2\right) \times .5 = A_3$

Third Iteration $\left(\frac{B}{A_3} + A_3\right) \times .5 = A_4$

Four Iteration $\left(\frac{B}{A_4} + A_4\right) \times .5 = A_5$

Fifth Iteration $\left(\frac{B}{A_5} + A_5\right) \times .5 = A_6$

Sixth Iteration $\left(\frac{B}{A_6} + A_6\right) \times .5 = A_7$ (Square Root)

*B* is the base number for which the square root is to be found. $A_1$ is the first approximate square root set up in the machine by the digit emitter. It may be 3, 30, 300, 3,000, 30,000, or 300,000, depending upon the number of significant digits in the base number. For a 1- or 2-digit base number, 3 is used; for a 3- or 4-digit base number, 30 is used; for a 5- or 6-digit base number, 300 is used; and so on. Theoretically, any number could be used as a first approximation. The number 3 is chosen because greater accuracy is obtained when the square root is carried beyond the decimal point.

$A_2$, $A_3$, $A_4$, $A_5$, and $A_6$ are the successive approximations developed by the machine on each iteration, or step, in the formula. $A_7$ is the approximate square root.

Some base numbers do not require six iterations, but a maximum of six is required for any base number from 1 to 12 digits. For example, the square root of 16 would be developed in one iteration as follows:

$$\left(\frac{16}{3} + 3\right) \times .5 = 4 \text{ (square root of 16)}$$

but the square root of 5625 would be developed in three iterations as follows:

$$\left(\frac{5625}{30} + 30\right) \times .5 = 108 \text{ (second approximation)}$$

$$\left(\frac{5625}{108} + 108\right) \times .5 = 80 \text{ (third approximation)}$$

$$\left(\frac{5625}{80} + 80\right) \times .5 = 75 \text{ (square root of 5625)}$$

It may be seen from the formula that each iteration is a repetition of the previous step; the only difference is the introduction of a new approximation as a divisor. Thus, this problem becomes primarily one of extending programs beyond 12; in other words, it makes repeated use of the program unit until the calculation is completed.

The method used in this problem to make use of the program unit six times and to stop further programming is to add 3 in the high-order position of a counter on the read cycle and add 1 on each iteration until the counter reaches 9. Further programming is stopped on the basis of a negative balance.

When repeated use of the program unit is required, program exit 12 must be active before another round can be started, and program exit 1 must be active to start each subsequent round. In other words, program exit 1 may be skipped only after the read cycle, and program exit 12 may be ignored only on the last round.

The above formula for calculating square root is derived from a basic formula for finding any root. This formula is expressed as follows:

$$A_2 = \frac{1}{n}\left[\frac{B}{A_1\ i - 1} + A_1\ (i - 1)\right]$$

*B* is the base number for which the root is to be found.

$A_1$ is the first approximation and may be any number, preferably 3, properly positioned according to the size of the base number.

*i* is the index of the number; that is, the root to be found.

$A_2$ is the second approximation and is substituted for $A_1$ in the formula for further calculation.

## Planning Chart and Control Wiring

*Read.* The base number is entered into storage unit 2 and the digit 3, from the emitter, into the high-order position of counter 6. A one is added to the 3 for each iteration taken. When the total reaches nine, six iterations will have been taken and the NB of counter 6 is used to stop further calculation.

*Program 1 (Exit 4).* Program exits 1, 2, and 3 are skipped after the read cycle by wiring a read cycles impulse to program skip 4. Program exit 4 then becomes active as the first program following the read cycle.

The base number is read out of storage units 2L and 2R and is used to pick up pilot selectors 1 through 5. This wiring, used to determine the size of the base number, is explained under "Right-Hand Panel Wiring." Pilot selector 6 is picked up on this program and is used to control an X from the digit selector. Program exit 4 is not needed for the rest of the problem.

*Program 2 (Exit 5).* The selectors that pick up on program 1 transfer on program 2, and are used to enter 3, 30, 300, 3,000, 30,000, or 300,000 from the digit emitter in storage unit 3. The 3, 30, and so on, are first approximations ($A_1$) and are determined by the size of the base number. Co-selector 1 is controlled on this program to select the digit 3. Program exit 5 is not needed for the rest of the program.

*Program 3 (Exits 6 and 7).* The base number is read out of 2L and 2R and added in the dividend counter. The first approximation ($A_1$) is read out of storage unit 3 and entered into the divisor. It is also added in counters 4 and 5. The machine is now set up to do:

$$\frac{B}{A_1} + A_1$$

Digit 1 is added to the 3 standing in counter 6.

*Program 4 (Exit 8).* The divide hub is impulsed, the dividend counter is wired to subtract, the divisor is read out, and counter 4-5 is impulsed to add. Thus, the base number is divided by $A_1$, (3, 30, 300, and so on) and the quotient, developed in counter 4-5, adds to $A_1$.

*Program 5 (Exit 9).* The dividend counter is reset.

*Program 6 (Exit 10).* The result in counter 4-5 is multiplied by .5. The product $A_2$ (second approximation) is developed in counter 1-2-3.

*Program 7 (Exits 11 and 12).* $A_2$ is read out of counter 1-2-3 into storage unit 3. Counters 1-2-3 and 4-5 are reset, and the whole process is repeated five times.

*Program 8 (Exit 1).* Program exits 2, 3, 4, and 5 are skipped, after the first iteration, by wiring program exit 1, which is always active after a round of programs, to program skip entry 6.

*Program 37 (Exits 11 and 12).* Program exits 1 and 6 through 12 are repeated on each iteration until READ is impulsed. READ is impulsed on program 37, at which time the high-order position of counter 6 has reached 9. Program 37 (exit 12) is wired through the transferred side of pilot selector 7 to READ. Pilot selector 7 is picked up by the NB of counter 6. Counter 6 is reset by wiring a program exit 12 through the transferred side of pilot selector 7. The square root ($A_7$) is read out of counter 1-2-3 into storage unit 7 for punching. Storage unit 7 is impulsed to read in and punch when counter 6 resets.

## Right-Hand Panel Wiring

1. The base number is entered into storage unit 2.

2. Pilot selectors may be picked up from storage transfer exits. Storage transfer exits, however, do not emit impulses for zeros; therefore, the pilot selectors transfer only for significant digits 1 through 9. The same is true of counter exits. This method of distinguishing between significant digits and zeros can be used to determine the size of a number, which is a basic requirement in calculating a square root.

In order to determine the size of the base number, it is read out of storage unit 2 to pick up pilot selectors 1-5. No two positions of a storage transfer exit may be connected together; therefore, the selectors are picked up by wiring as follows: with the exception of the units and tens positions of storage exit 2, all the odd-numbered positions (3, 5, 7, and so on) are wired to pick up pilot selectors 1 through 5 through their digit pickup hubs, and all the even-numbered positions (4, 6, 8, and so on) are wired to pick up pilot selectors 1 through 5 through their immediate pickup hubs. Thus, adjacent positions independently transfer a single selector, which is used as a testing device. For example, if pilot selector 1 transfers it means that a significant digit is punched in either the 11th or 12th position of the base number. In either case, 300,000 is the first approximation.

3. Selectors 1-5 are used to position the 3 on the cycle following their pickup. The digit pickup transfers the selectors on the following cycle, but the immediate pickup does not. Therefore, some method must be used for holding the selectors in their transferred positions when the immediate pickup is impulsed. This is done by wiring X impulses from the digit selector (wired as an emitter) through the transferred side of the lower part of pilot selectors 1 through 5 to their X pickups. The X is first tested through the transferred side of pilot selector 6 on program 1.

4. Digit 3, from the emitter, is wired through the transferred side of co-selector 1, picked up on program 2 (exit 5), to the common of pilot selector 1. If all five pilot selectors are normal, a base number of one or two significant digits is indicated and the 3 is distributed out of the normal side of pilot selec-

tor 5 to the units position of 3R. If the first four pilot selectors are normal and the fifth one is transferred, a 3- or 4-digit base number is indicated and the 3 is distributed from the transferred side of pilot selector 5 to the second position of 3R. By the same reasoning, a 3 is distributed from the transferred side of pilot selector 4 to the third position of 3R for a 5- or 6-digit base number, from the transferred side of pilot selector 3 to the fourth position of 3R for a 7- or 8-digit base number, from the transferred side of pilot selector 2 to the fifth position of 3R for a 9- or 10-digit base number, and from the transferred side of pilot selector 1 to the sixth position for an 11- or 12-digit number.

5. The first approximation $A_1$, secured by the positioning of the 3, and the second, third, fourth, fifth, and sixth approximations, obtained by calculation, are wired from storage exits 3L and 3R to the divisor unit and to counter 4-5.

6. The divisor is wired to the dividend, as it is in all dividing operations.

7. The base number is wired from 2L and 2R exits to the dividend.

8. Co-selector 2 is picked up from a read couple and is used to enter the digit 3 in the high-order position of counter 6 on the read cycle only. The 3 is added in the counter through the transferred side of this selector from the normal side of co-selector 1.

9. Digit 1 from the digit emitter is wired through the transferred side of co-selector 4 to the high-order position of counter 6. The digit 1 will add to the 3 entered in the counter on the read cycle, for the purpose of stopping further calculation when nine is reached. Co-selector 4 is picked up on the first program of each iteration (exit 7).

10. The quotient is developed in counters 4 and 5.

11. The approximations ($A_1$, $A_2$, and so on) are wired out of counter 2-3 into storage unit 3. Counter 2-3 is also wired to 7R from which the square root is punched on program 37.

12. The result of $B/A + A$ is read out of counter 4-5 for each iteration and wired into counter 1-2-3 through the common hubs of the divisor exit. This is the multiplicand wiring for the multiplication by .5 to develop the approximations $A_2$ through $A_7$.

13. The square root is wired from storage punch exit 7 to punch in columns 41-46.

## GROUP MULTIPLIER CHECK

In group multiplication operations, each group multiplier master card precedes the group of detail cards with which it is associated. The master cards are usually filed in front of their respective groups either by hand or in a collating operation. Obviously, if any one of them is misfiled, the following detail cards will be multiplied by the wrong group multiplier.

This example describes a method of stopping the machine immediately when master cards and detail cards are not properly associated. The group indication information (common control field) is read into a storage unit from an X-punched group multiplier master card at the same time that the group multiplier is entered into the multiplier storage unit. The group indication from each detail card is added into a counter. During some convenient program step (usually one during which other transfer operations are performed), the group indication in storage is subtracted from the group indication in the counter. The result in the counter should be zero if the master card and the detail card are properly associated. A zero test is made to stop the machine in case of error.

The planning chart and control panel wiring are illustrated in Figure 56. The zero test is made on program 2 in the diagram shown, but in actual practice any convenient program step not used for multiplication or division may be used.

1. The X in the group-multiplier master card picks up pilot selector 2.

2. The group-multiplier is read into the multiplier storage unit and the control field is read into storage unit 4.

3. The X in the group-multiplier master card causes the master card to skip out without programming and feed the first detail card.

4. The control field in the detail card is added into counter 1.

5. The following operations are performed on program 2:

   a. The control field in storage is subtracted in counter 1.

   b. Co-selector 6 is picked up from program couple.

   c. Digit 9 from the digit selector passes through the transferred side of co-selector 6 and picks up co-selector 5.

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT |  | COUNTER |  |  |  |  |  | STORAGE UNITS |  |  |  |  |  | PUNCH UNITS |  |  |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  |  | DIVR.-MULT. | DIVIDEND |  |  |  |  |  |  |  |  |  |  |  | *UNITS POSITION MUST BE WIRED TO PUNCH |  |  |  |
|  |  |  | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R * | 7L | 7R * |
|  |  | READ CYCLE X MASTER CARD |  | RI GROUP MLPR |  |  |  |  |  |  |  |  |  |  |  | RI MASTER CONTROL FIELD XXXXX |  |  |  |  |
|  | ~~1~~ | READ CYCLE NX DETAIL CD. |  |  | ADD DETAIL CONTROL FIELD XXXXX |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | ~~2~~ 1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | ~~3~~ 2 | SUBT & TEST |  |  | SUBT XXXXX 999999 |  |  |  |  |  |  |  |  |  |  | RO |  |  |  |  |
|  | ~~4~~ 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | ~~5~~ 4 |  |  |  | RESET |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | ~~6~~ | READ CYCLE NX DETAIL CD |  |  | ADD CONTROL FIELD XXXXX |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | ~~7~~ |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

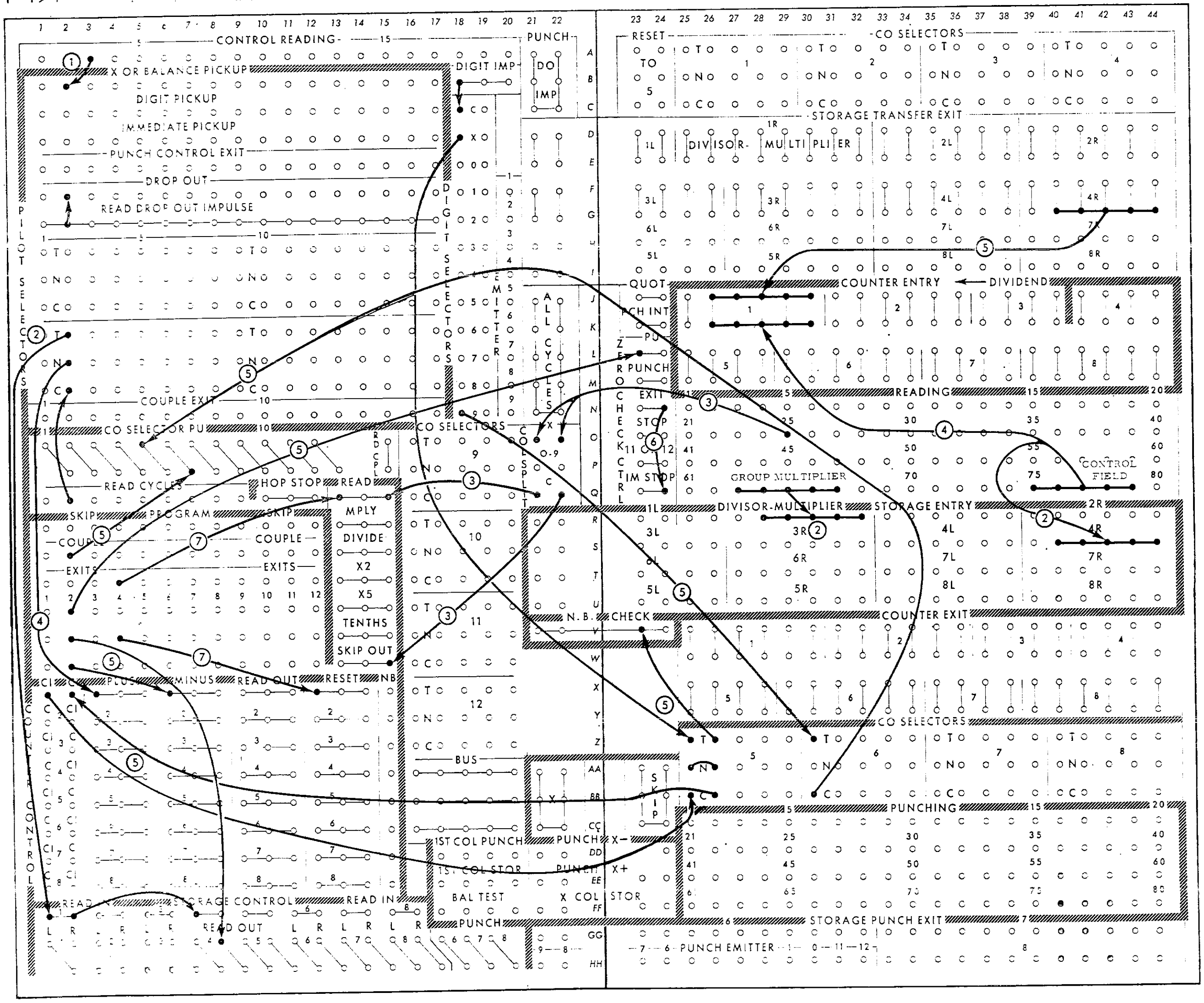


*Figure 56. Group Multiplier Check*

d. The X from the digit selector passes through the transferred side of co-selector 5 through the CI and C hubs of counter 1 only if the result is zero.

e. If the result is zero, the X impulses NB CHECK.

f. Zero check control is picked up. An error is indicated when zero check control is impulsed and NB CHECK is not.

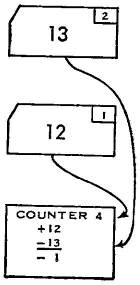
6. Exit to IM STOP stops the machine in case of error.

7. End of programming. Counter 1 is reset in preparation for the next card.

## COMPARING AND SEQUENCE CHECKING

It is possible to recognize control group changes automatically on the 602 without using distinguishing code punches. It is also possible to check the sequence of cards passing through the 602 and stop the machine whenever a "step down" is recognized.

Both of these functions are accomplished by column comparison between successive cards, made possible by reading the columns to be compared at both the upper (control) station and the lower (reading) station into counters whose negative or positive position then determines a change in grouping or an error in sequence (Figure 57).

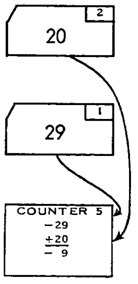


COMPARING

Card at control (upper) station subtracts in counter 4 on program 1.

Card at reading (lower) station adds in counter 4 on read cycle.

If counter 4 is negative on program 1, a change in control is indicated.



SEQUENCE CHECKING

Card at control (upper) station adds in counter 5 on program 1.

Card at reading (lower) station subtracts in counter 5 on read cycle.

If counter 5 is negative on program 1, cards are out of sequence.

As shown in the schematic diagram, a change in control group is recognized by the negative position of counter 4 at the end of program 1. As card 1 passes the read station, the columns to be compared are added in the counter. On program 1, as card 2 passes the control station, the columns to be compared are subtracted in counter 4. If the balance is zero, no change in control is indicated. If the balance is negative, the card passing the control station is the first card of a new group.

A step down in sequence is recognized by the negative position of counter 5 at the end of program 1. As card 1 passes the read station, columns to be sequence-checked are subtracted in counter 5. On program 1, as card 2 passes the control station, the columns to be sequence-checked are added in counter 5. If the balance is plus, the cards are in sequence. If the balance is negative, the cards are out of sequence and the machine may be wired to stop and the comparing light to turn on.

Although the wiring diagram shows the use of only two control columns, the maximum number of columns that may be compared depends upon counter capacity and the number of control brushes that may be set adjacent to each other without affecting card feeding. Brush setting should be balanced as much as possible. For example, if five brushes are set to read columns 10-14, a corresponding number of brushes should be set to ride over the right end of the card even though they are not to be used for reading purposes.

In Figure 57, the use of two counters is shown, counter 4 for comparing and counter 5 for sequence checking. Either operation may be performed without the other, in which case only one counter would be necessary. It should be remembered that when a comparing operation is done by itself, the cards must be in sequence.

1. Columns 64-65 are wired from the reading station to counter entries 4 and 5.

?. Control reading brushes 14 and 15, set to read columns 64 and 65, are wired to entries 4 and 5.

3. Counter 4 adds and counter 5 subtracts on the read cycle.

4. During program 1, the card at the control reading station is subtracted in counter 4 and adds in counter 5.

5. If counter 5 is negative at the end of program 1, a step down in sequence is indicated. The NB of counter 5 is wired to pick up pilot selector 3, and program 2 is wired through the transferred side of pilot selector 3 to immediate stop.

6. If counter 4 is negative at the end of program 1, a control change is indicated. The NB of counter 4 is wired to pick up pilot selector 2, and program 2 is wired through the transferred side of pilot selector 2 to perform some specific function, such as reading out and resetting counter 3.

7. Counters 4 and 5 are reset on program 2 and wired to read.

Because of balance testing operations on program 1, it is not possible to multiply or divide on this step. False error conditions that may be signaled on the run-in or the run-out may either be ignored or eliminated by selection.

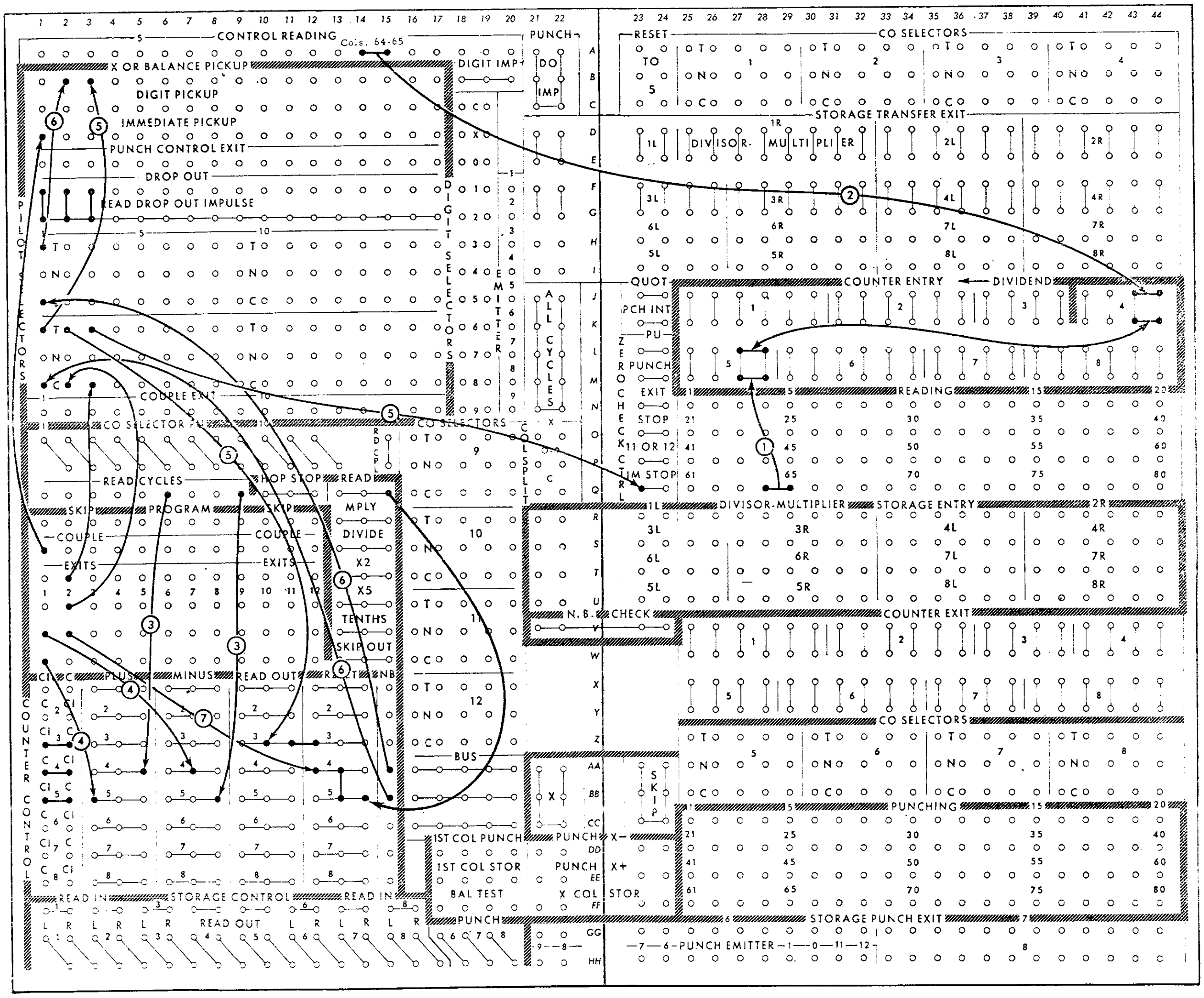


*Figure 57. Compare and Sequence Checking*

# Operating Suggestions

BEFORE a feature or a control panel hub was used in this manual it was described. Because of the wide flexibility in the use of hubs, however, it was not always possible, without complicating the basic instruction, to present some of the corrective measures that must be taken when a feature or a control panel hub is used in an unusual way. Some of these measures are described here, as well as the techniques used to analyze control panel wiring errors.

## SIGNAL LAMP UNIT

IN ORDER to simplify the analysis of control panel wiring errors, the signal-lamp unit of the machine may be used. This unit, located under the top cover of the machine, is used primarily for circuit analysis by customers engineers but is helpful when analyzing control panel errors.

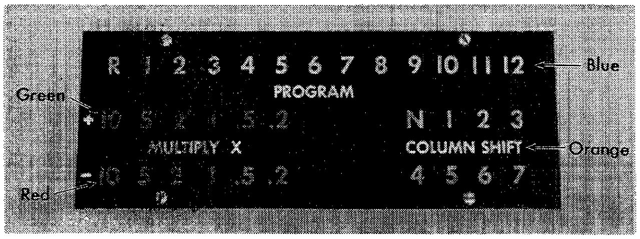


*Figure 58*

The signal lamp unit (Figure 58) is designed to indicate these four conditions:

1. The program step active during each cycle.
2. The shift setup during each cycle.
3. The value of the multiplication, if any, occurring during each cycle.
4. The sign of the counter operation during each multiplying cycle.

The arrangement, color, and meaning of the lamps are indicated in Figure 58. The lights are made operative by an indicator light switch on the lower right end of the machine. This switch should be off during normal machine operation.

A motor switch on the lower right end of the machine may be used to run one cycle only, each time the single cycle switch is operated. On later model machines, the switch has been replaced by a portable start-key cord which serves the same purpose. This switch may be used in conjunction with the signal lamp unit to investigate various conditions at length following each cycle of a calculation or program. The program step light is probably the most valuable indicator, since the light indicates what step is being taken. On a multiply or divide step, the multiply X light, the column shift light, and the sign-of-counter light will go on. Although it is not practical for the person testing to develop the product or quotient by the same arithmetic table used by the machine, he can note that either the column shift lights or the multiple X digits do not change as the machine advances from one calculate cycle to the next. Very often, this type of indication reveals a control panel error which can be readily recognized.

At any given time the digits in storage may be read by referring to the storage units beneath the top cover of the machine. These storage units are arranged in sequence from left to right: storage units 1, 2, 3, and 4 on the lower segment, and storage units 6 and 7 on the top segment. The storage positions are arranged vertically so that the first six represent the digits in the left side of the storage unit and the next six represent the digits in the right side of the storage units.

The counters are not readily visible and, in almost all cases, considerable time can be saved if the value in the counter is temporarily transferred to a storage unit to check the existence of an amount on a particular cycle. The number appears in storage in true form if the value in the counter is plus, or in complement form if the value is minus.

## PROCEDURE FOR SINGLE-CYCLE TESTING

THE FOLLOWING steps should be used for single-cycle testing:

1. Prepare a number of test cards which contain the same values that were entered on the planning chart.

2. Insert the control panel to be tested into the machine. Check to see that the skip bar contains inserts in the proper columns and that the control brush settings have been made if the operation involves the use of a control brush.

3. Turn on the main line switch, the indicator light switch, and the motor switch.

4. Feed the first card into the machine by raising the single cycle switch once. A second operation of the switch moves the card past the control reading brushes.

5. The third operation of the switch initiates the run-in reset cycle. At this time, storage units impulsed to read in from the card restore to zeros, and factors emitted into storage appear in the positions wired. Each calculate and program cycle wired on the control panel requires that the switch be operated. When the required number of cycles have occurred to complete the run-in reset, the card moves past the reading brushes. The read cycle light in the signal lamp unit will be on.

6. Check that the correct factors have entered the proper storage units by comparing the storage unit readings with the planning chart entries.

7. Continue operating the single-cycle switch. As the machine advances from one program step to the next, the accuracy of the wiring can be verified by comparing the factors in storage with the test values on the planning chart. The signal light unit indicates the program step active at a given time, the multiply or divide program, and the column shift occurring.

8. At the first discrepancy between the machine results and the planning results, the card can be removed by pulling forward the control panel door. This releases the card without punching.

9. Correct the control panel wiring, insert the test card, and begin the test again.

10. If testing is complete, automatic machine operation may be resumed by turning the motor switch off, and depressing the start key to initiate card feeding.

## TYPICAL CONTROL PANEL ERRORS

CONSIDERABLE time can be saved during control panel testing if the program planner is aware of some of the more common errors which can occur and the techniques that can be used to locate these errors. Although the errors are obvious when shown individually, they are more difficult to isolate in a complex problem with many wires. Usually, the complexity of the problem creates the errors. As each step is added, it is often necessary to use the same hub over again as factors are transferred between various units on the right side of the control panel and machine commands are added on the left side of the panel. As a result, conflicting instructions may be given to one unit at the same time, or units may be inadvertently connected.

One of the most helpful techniques is to wire and test the problem in parts. For instance, each intermediate result, as it is developed, can be read into an unused storage unit from the counter to verify the accuracy of the program steps up to that point. Using the single-cycle unit facilitates checking of each step. As a result of this step-by-step problem development, it is easier to isolate the step on which an error has been made, since previous steps have been verified.

The following illustrations show the more common types of control panel errors and a correct wiring solution.

Figure 59 shows a back circuit resulting when storage unit 4 receives an unintentional impulse to read in and read out on program 3. The program 3 impulse to counter 2 plus travels through wire A to the read-out of storage 4L, although 4L should be read out on program 2 only. This conflicting set of instructions can be recognized by observing a rocking motion of the storage units.

The error can be corrected by removing wire A and adding an independent program 2 exit to the read-out of 4L.

Figure 60 illustrates the improper wiring of the reset to 5. Because counters 2 and 3 both read in the value from columns 34-36, the reset-to-5 impulse uses the external wiring path between counter entries 2 and 3 and adds to the amount in both counters if each counter is impulsed simultaneously. Reset to 5 (an X-timed impulse) can be made to half-adjust counter 2 only by isolating the units position of counter 2 through a column split.

Figure 61 points out an error created by multiple use of the column 21 impulse. The program 3 exit, wired to the read hub to end programming on ND21 cards, uses the common pathway through the read hub to pick up pilot selector 4 every time program 3 occurs. SKIP OUT does not accept a program exit impulse.

This error can be corrected by selecting the D impulse and the program exit through pilot selector 2.

The control panel error in Figure 62 develops an erroneous answer in counter 2 if storage 4R is read out into counter 2 on the same step on which counter 6 reads into counter 5. The connection between counter exit 6 and counter entry 2 must be selected on the program step on which the transfer is made in order to break the common connection between counter 6 and storage 4R.

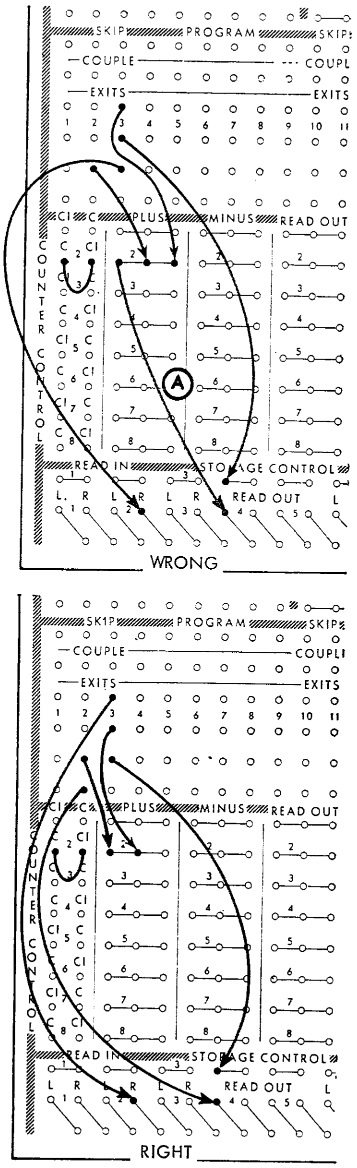


*Figure 59*

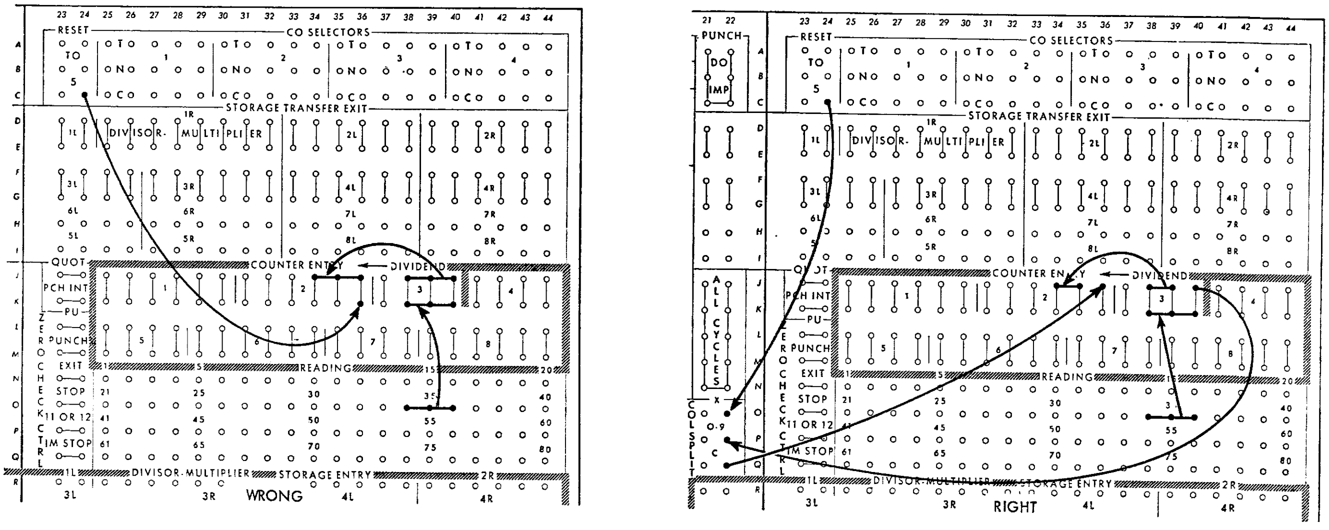


*Figure 60*

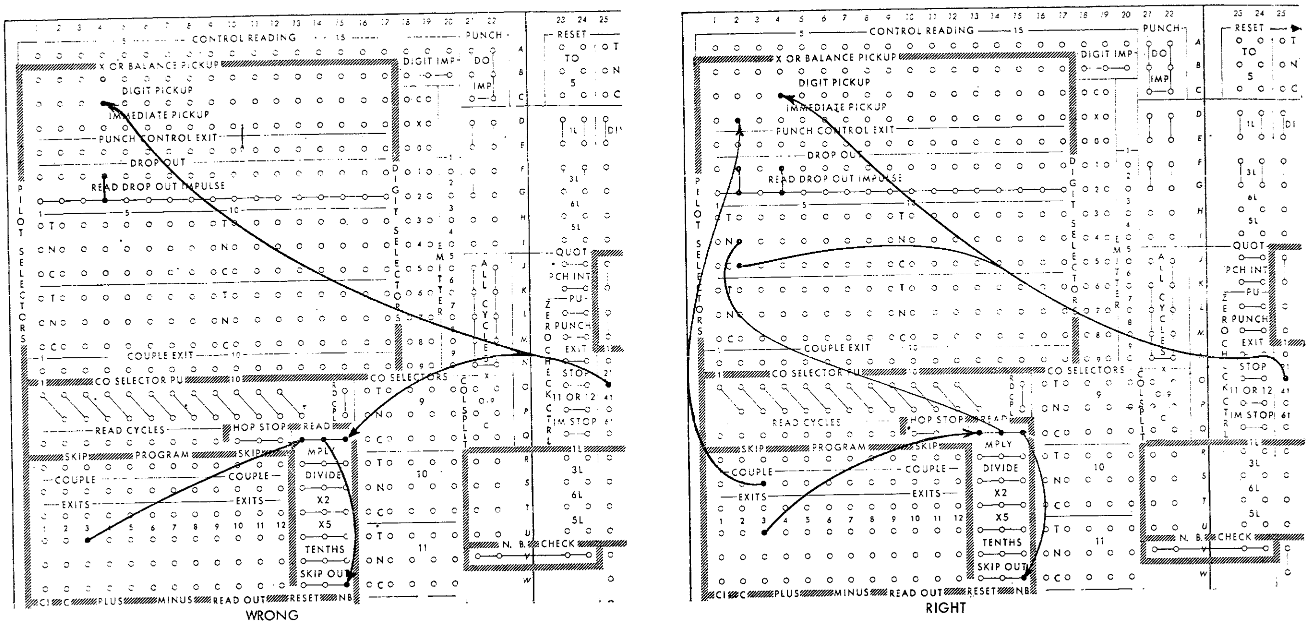


*Figure 61*

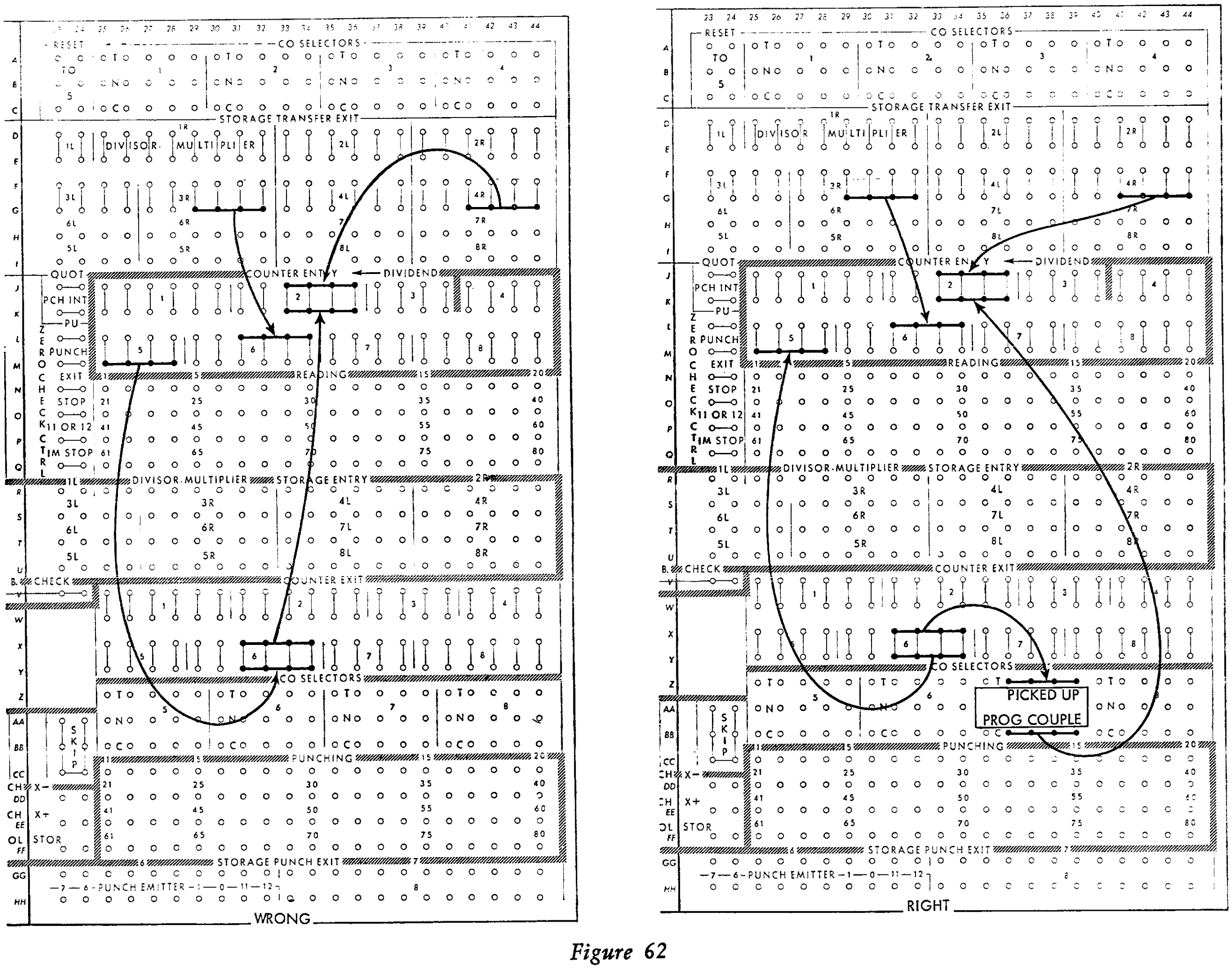


*Figure 62*

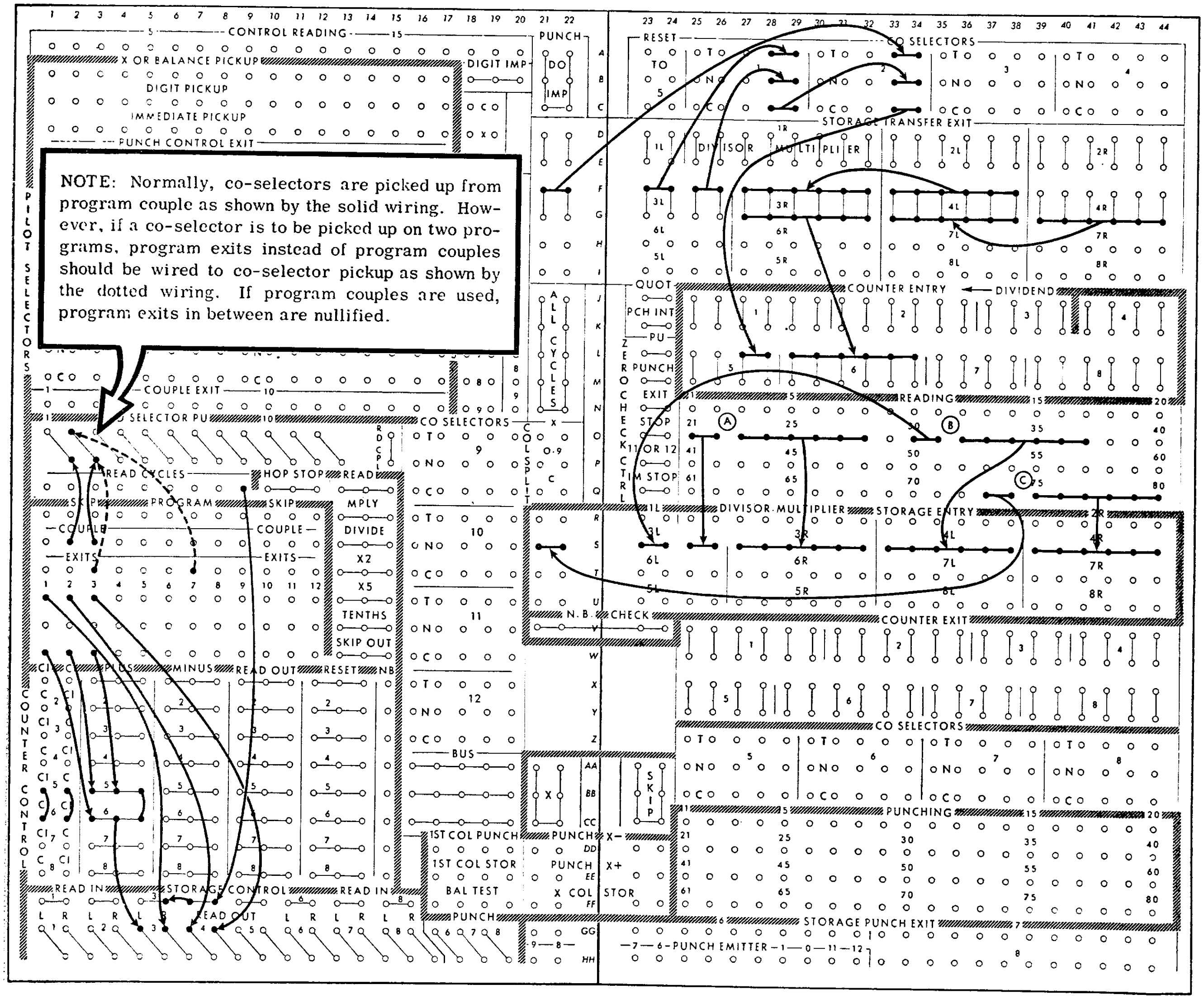
## STORAGE ECONOMY

The twelve-position storage units in the 602 are split up into a left and a right section, each section having its own independent read-out control. Normally, this arrangement permits the storing of one field in the left section and another field in the right section. Each section may then be controlled to read out independently of the other.

Fields larger than any one section of a storage unit are normally stored in the left and right sections of the same storage unit, in which case both sections would be controlled to read out simultaneously. This method, however, often results in wasted storage capacity, and, in some problems, may even necessitate a second run owing to lack of sufficient storage capacity on the first run.

This example (Figure 63) presents a method of economizing on storage capacity by making use of normally wasted storage positions. Three 8-position fields are stored in two available 12-position storage units. The two high-order positions of fields *A*, *B*, and *C* are wired into 3L, and the six low-order positions of each field are wired into 3R, 4L, and 4R, respectively. The fields are then transferred to counter 5-6 in the following order:

Program 1—Field *A* (3L and 3R read out)<br>
Program 2—Field *B* (3L and 4L read out)<br>
Program 3—Field *C* (3L and 4R read out)



NOTE: Normally, co-selectors are picked up from program couple as shown by the solid wiring. However, if a co-selector is to be picked up on two programs, program exits instead of program couples should be wired to co-selector pickup as shown by the dotted wiring. If program couples are used, program exits in between are nullified.

*Figure 63. Storage Economy*

## OPERATION OF THE COMPARE LIGHT

The compare light on the 602 is normally used to indicate an error during a multiplication or division verification. It is turned off by depressing the reset key and then the start key to resume operation.

The light may also turn on as a result of any of the following conditions:

1. Card jam at control brush station. The compare light goes on if a card jams. Programming will advance only one step beyond the step on which an error occurred.

2. Improper counter reset. If a counter fails to reset to 9's, the machine stops.

3. Improper storage punching. If a storage unit is impulsed to punch and the units position has not been punched when the card passes column 80, the machine stops.

4. Punch jam. The compare light goes on if a card fails to feed out of the punch magazine because of a jam.

5. Feed jam. The compare light goes on if a card jams in the feed throat.

6. Feed failure. The compare light turns on if the punch card carrier fails to pull the card from under the die card lever.

7. Impulsing stop. This causes a stop after a punch eject.

8. Impulsing immediate stop. This causes a stop two to three cycles later.

The compare light may be of considerable help in directing the attention of the operator to several types of difficulties. While these troubles occur infrequently, a great deal of time may be saved by a full understanding of what the compare light indicates.

# Series 50 Operations

## CONTROL PANEL

THE COMPLETE control panel is shown in Figure 64. Shaded hubs on this diagram indicate the features that are not available on the Model 50 or 51 machines. Model 50 incorporates the divide features. With division, the following hubs are standard: divide, tenths, and quotient.

## COUNTER COUPLING

THE RULES for coupling counters on a 602 Series 50 are the same as those described under the heading "Counter Controls." The counters may be coupled in any order for addition or subtraction, but adjacent counters must be coupled during a multiplication or division operation. Counter 6 is not considered to be adjacent to counters 3, 2, or 1. Therefore, in either a multiplication or division operation, counter 6 cannot be coupled with any other counter. Counter 3 may be coupled with 2, or counter 2 may be coupled with 1, or all three may be coupled together to develop a product. In a divide operation, counters 1, 2, and 3 are always coupled together to enter the dividend.

## PUNCHING

A MODEL 50 or 51 has one 12-position storage unit from which all digits must be punched. The storage unit is divided into left and right sections of six positions each, but all 12 positions are controlled simultaneously when reading in or punching. Any positions from the storage unit can be punched as long as the units position is also punched. The same impulse which is used for controlling read-in to the punch storage unit can be used to start punching. If punching is to take place at a later time than entry into the unit, different impulses would control read-in and punch. When punch control receives an impulse, punching starts on the following cycle and proceeds at the rate of four columns per machine cycle. As in any other storage unit, information in storage 7 is not lost even though it has been punched and the data stored can be read out for transferring to another unit either before or after punching, but not simultaneously.

All 12 positions of storage unit 7 are cleared when impulsing read-in and any storage entry position not receiving a digit impulse is reset to zero. As a result, storage unit 7 can be used repeatedly throughout a problem for punching results, provided the results can be punched in logical sequence from left to right in the card.

The 602 has a built-in interlock to prevent storage punch units from reading in a new value while punching is going on. As soon as the storage unit receives an impulse to punch, an interlock relay is picked up and held until the units position of the storage unit has been punched. Therefore, the machine waits until the punching from the storage unit 7 is completed. Punching from the units position of the storage unit signals completion of punching.

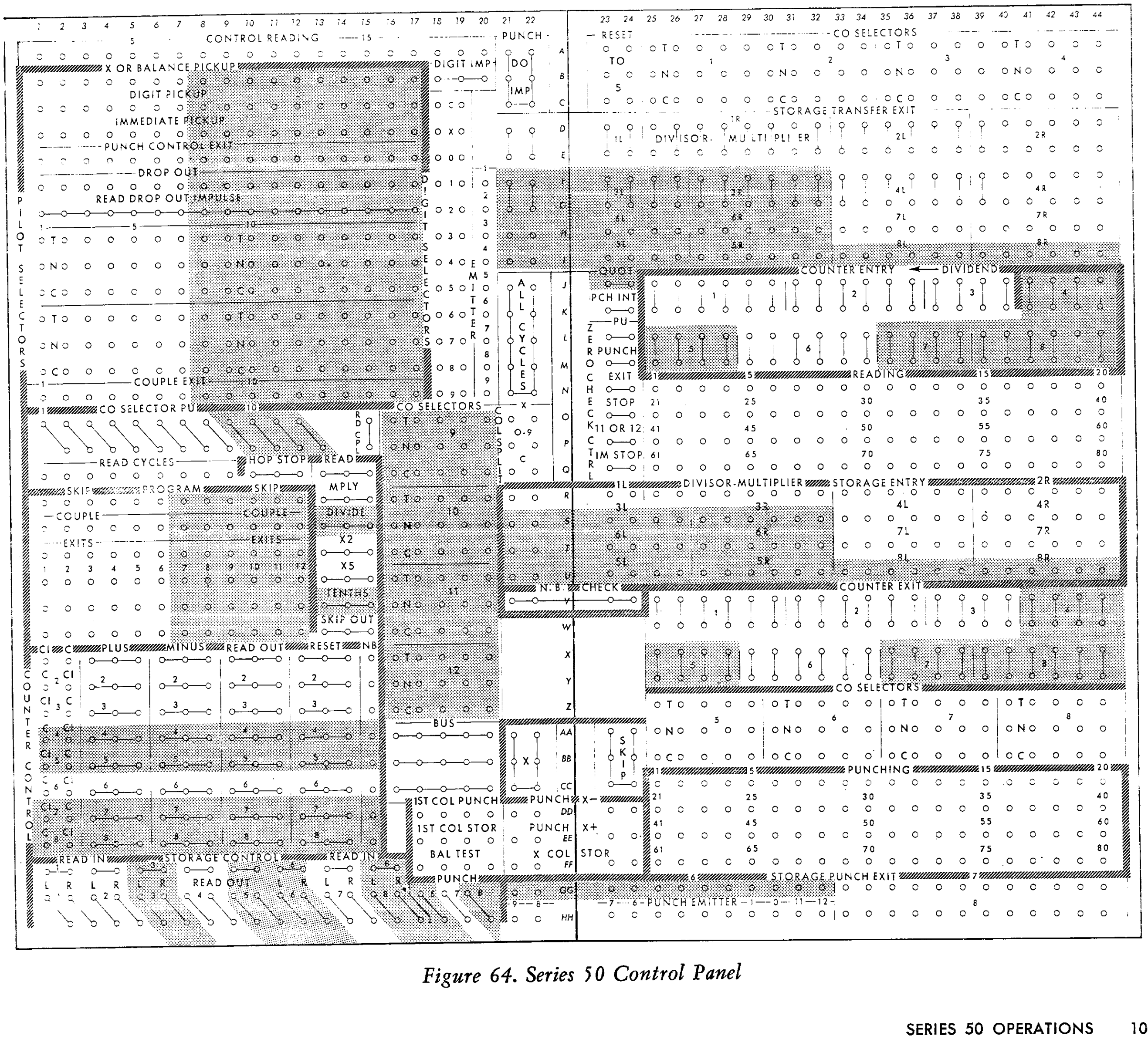


*Figure 64. Series 50 Control Panel*

## PUNCHING MORE THAN 12 COLUMNS

## $A \times B = P_1;\ A \times C = P_2;\ A \times D = P_3$

In this example (Figure 65), several multiplicands of different sizes are multiplied by one multiplier to develop three products simultaneously. Each product punches in the card. Storage unit 7 is impulsed to read in and punch all three products. Since $P_1$ and $P_2$ punch in fields to the left of $P_3$, $P_1$ and $P_2$ must punch first.

### Planning Chart and Control Wiring (Figure 65)

*Read.* The multiplier A is entered in 1R. The multiplicands are entered in storage entries 2L, 2R, and 4L.

*Program 1.* All three multiplicands are read out of storage into three separate counters to develop the three independent products. The counters are impulsed to add, and the multiply hub is impulsed from program exit 1.

*Program 2.* The products in counters 3 and 1 are read out and entered in storage for punching. Storage 7 is impulsed to punch and the counters are reset. It does not matter in what sequence the digits are stored for punching, as long as the units position is wired to punch. In other words, even though column 46 is the first column punched, it need not be the left-hand column in the storage punch exit. However, column 58, the last column punched on that step, must be wired from the units position.

*Read Cycle, Card 2.* The third product is read out of counter 6 and entered into storage entry 7R. The counter is reset and the storage unit impulsed to punch. The third product will not read in until product 2 has been completely punched.

### Right-Hand Panel Wiring

1. The multiplier is entered into 1R. The multiplicands B, C, and D are entered into storage units 2 and 4.

2. Storage exits 2 and 4 are wired to counter entries 1, 3, and 6 in such a way that the product of A × B will develop in counter 3, the product of A × C in counter 1, and the product of A × D in counter 6.

3. The product $P_1$ is wired from counter exit 3 to storage entry 7L. The product $P_2$ is wired from counter exit 1 to storage entry 7R, and the product $P_3$ is wired from counter exit 6 through the common hubs of counter 1 to 7R.

4. $P_1$ is wired to punch in columns 46-49, $P_2$ in columns 53-58, and $P_3$ in columns 62-66. The last five positions of storage 7R are split-wired because they are used to punch both $P_2$ and $P_3$.

## MULTIPLICATION

A Series 50 602 Calculating Punch can multiply an 8-digit multiplicand by an 8-digit multiplier to obtain a 16-digit product on a single program step. The maximum size of the multiplier is determined by the fixed size of storage unit 1R which must be used to store the multiplier. The number of digits in the multiplicand may be increased beyond eight, if the multiplier is decreased proportionately. In any case, the product limit is established by the size of coupled counters 1, 2 and 3. Because counters must be adjacent to be coupled in a multiplication or division problem, counters 1, 2, and 3 provide the 16-position limit for developing a product.

### Multiplier Expansion

The multiplier may be expanded beyond the 8-digit capacity of the multiplier unit 1R by calculating the multiplication in parts (Figure 66). For example, the calculation

$$33333322222222 \times 55555555 = 1851851216049376543210$$

may be accomplished as follows:

$$\begin{array}{lll r}
(1) & 55555555 \times 22222222 & = & 1234567876543210 \\
(2) & 55555555 \times 333333 & = & \underline{185184998148150000000} \\
 & & & 1851851216049376543210
\end{array}$$

Note from the above example that a 14-digit multiplier is divided in two parts: (1) The multiplicand is multiplied by the eight low-order positions and the result is stored. (2) The multiplicand is multiplied by the six remaining positions of the multiplier and the previous product added to the second product.

There is no set rule for the way the multiplier must be split when it exceeds eight positions. The best approach to solving this type of problem is to plan it carefully on a planning chart, then substitute actual values which have been calculated on paper first, and apply the same factors to the machine problem.

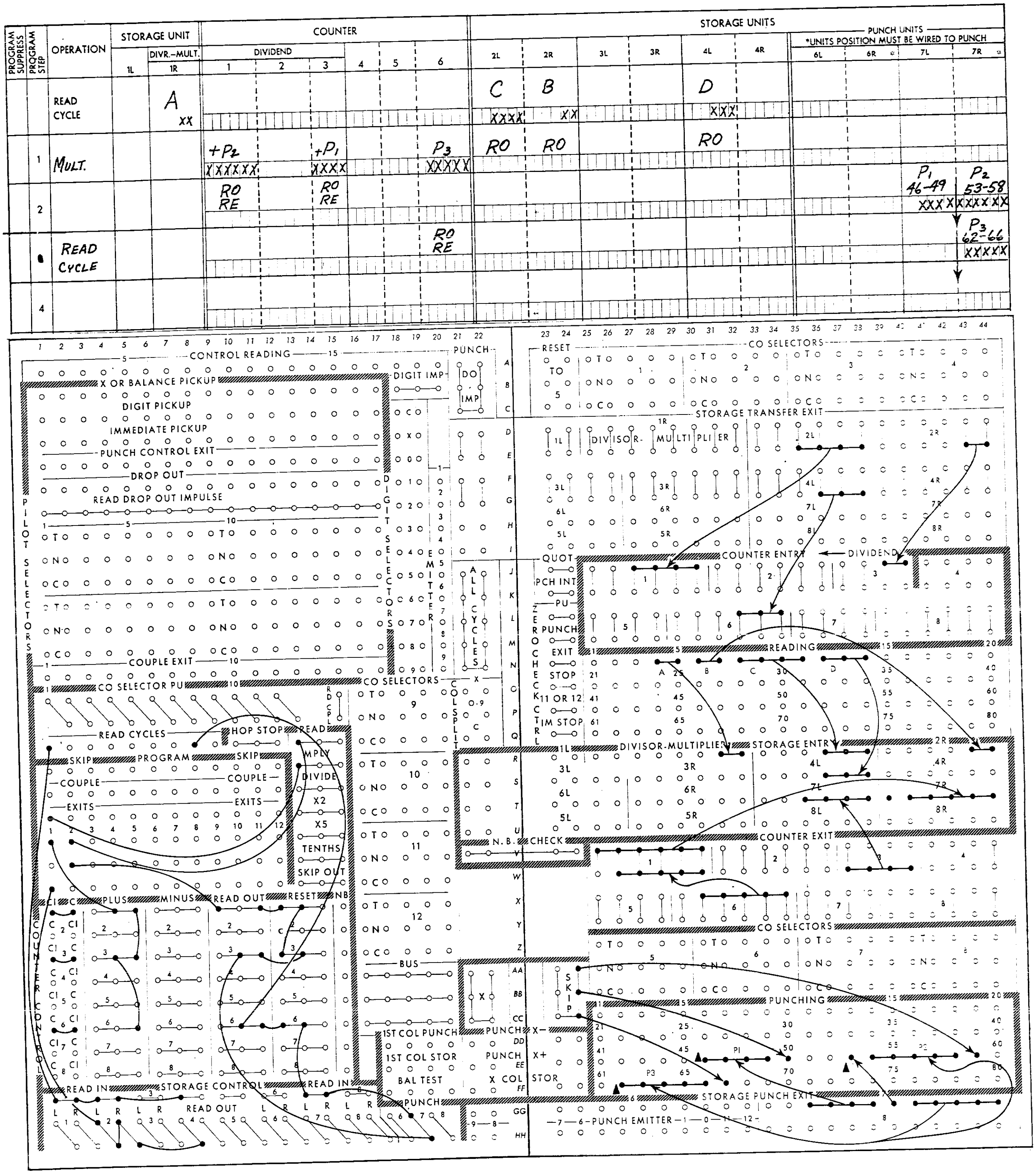


*Figure 65. Punching More than 12 Columns*

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT 1L | STORAGE UNIT DIVR.-MULT. 1R | COUNTER DIVIDEND 1 | COUNTER DIVIDEND 2 | COUNTER DIVIDEND 3 | COUNTER 4 | COUNTER 5 | COUNTER 6 | STORAGE UNITS 2L | STORAGE UNITS 2R | STORAGE UNITS 3L | STORAGE UNITS 3R | STORAGE UNITS 4L | STORAGE UNITS 4R | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH 6L | 6R | 7L | 7R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | READ CYCLE CARD 1 | | A<br>22222222 | | | | | | | B<br>55555555 | | | | | A<br>333333 | | | | |
| | 1 | MULT. | | | +<br>12345678 | +<br>76543 | +<br>210 | | | | RO | RO | | | | | | | | |
| | 2 | | | RI<br>333333 | | | | | | | | | | | | RO | | | | |
| | 3 | | | | RO<br>RE | RO<br>RE | RO<br>RE | | | RI<br>543210 | | | | | RI<br>123456 | RI<br>7876 | | | | |
| | 4 | MULT. | | | +<br>18518499 | +<br>8148 | +<br>15 | | | | RO | RO | | | | | | | | |
| | 5 | | | | +<br>185185 | +<br>1234<br>1216 | +<br>567876<br>049376 | | | | | | | | RO | RO | | | | |
| | 6 | | | | RO | RO | | | | | | | | | | | | | RI<br>185185 | RI<br>121604 |
| ■ | | READ CYCLE CARD 2 | | A | RE | RE | RO<br>RE<br>9376 | | | RO<br>RE<br>543210 | | B | | | | | | | RI<br>9376 | RI<br>543210 |
| | 8 | | | | | | | | | | | | | | | | | | | |

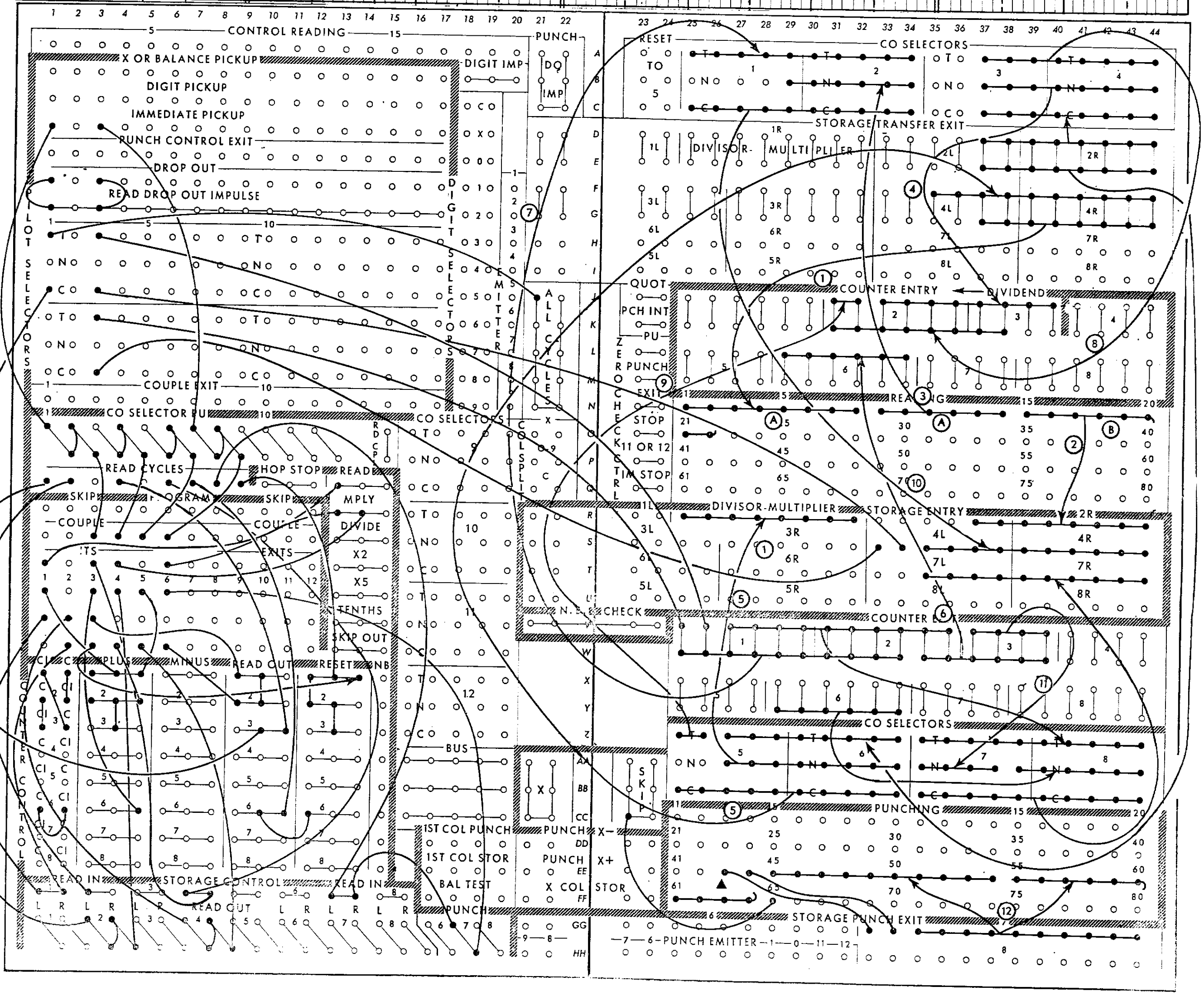


*Figure 66. Multiplier Expansion, Series 50*

## Planning Chart

*Read Card 1.* The eight low-order digits of the multiplier are entered into 1R and the six high-order digits of the multiplier are entered into storage unit 4. The multiplicand is entered into storage unit 2.

*Program 1.* The multiplicand is multiplied by the eight low-order positions and the product of 16 digits is developed in counters 1, 2, and 3 coupled. Counters 1, 2 and 3 are impulsed to add. Storage units 2L and 2R are read out and the multiply hub is impulsed.

*Program 2.* The remaining part of the multiplier is read out of 4R and entered into 1R.

*Program 3.* Eight positions of the final product are now correct. Six of these positions are stored in counter 6 for punching. Because these are the low-order positions of the final product, they cannot be punched until the high-order positions are punched if it is assumed that the entire product is punched in sequence in a card field. Counter 6 could not be coupled with counters 1, 2 and 3 during the multiplication, since adjacent counters only can be coupled for multiplication or division.

The other ten positions of the product are stored in storage unit 4.

*Program 4.* The multiplicand is now mulitplied by the second part of the multiplier. The multiplicand is shifted into the product counter to position the second product for adding with the first product. In effect, the second product must shift eight positions to the left of the first product (see example above). The number of places to be shifted should equal the number of positions in the first multiplier. Because of the limited size of the product counter, a shift of eight places is not possible on the second multiplication. However, six positions of the final product are already stored for punching and need not be added to the second calculation. Therefore, the multiplicand must be shifted two positions in the product counter on the second multiplication.

*Program 5.* The first product is read out of 4L and 4R and added to the second product in counters 2 and 3. The remainder of the final product is not in the counter.

*Program 6.* The twelve high-order positions of the product are read out of counters 1 and 2 and entered into storage 7 for punching.

*Read Cycle Card 2.* While a new multiplier and a new multiplicand are entered into storage from the second card, the remaining ten low-order positions of the product developed during the calculation of card 1 are punched. The 12th, 13th, 14th, and 15th positions of the product are read out of counter 3 and entered into 7L while the positions 17 through 22 are read out of counter 6 and entered into 7R. All counters are reset and storage unit 7 is impulsed to punch.

## Right-Hand Panel Wiring

1. The first eight positions of the multiplier are wired into storage unit 1R through the common hubs of storage exit 4L and 4R and the normal side of co-selectors 5 and 6.

2. The multiplicand is wired directly into storage units 2L and 2R.

3. The six high-order positions of the multiplier are wired through the normal side of co-selectors 1 and 2 to storage entry 4.

4. The multiplicand reads out of storage exit 2L and 2R into counters 1, 2, and 3 through the normal side of co-selectors 3 and 4.

5. The second part of the multiplier reads out of storage exit 4L and 4R through co-selectors 5 and 6 to storage entry 1R.

6. The six low-order positions of the product are read out of counters 2 and 3 and stored in counter 6.

7. The ten high-order positions of the first product are wired through the transferred side of co-selectors 1 and 2 to storage entry 4.

8. The multiplicand is read out of storage exit 2 through the transferred side of co-selectors 3 and 4 to counter entry 1, 2, and 3 offset 2 positions.

9. The remaining portion of the first product is transferred to the counter to add to the second product. Storage exit 4R is wired through the transferred side of co-selectors 5 and 6, and laced through the common hubs of storage exits 2 to the common side of co-selectors 3 and 4 and out of the normal side to counters 1, 2, and 3.

10. Counter exits 1 and 2 are wired through the transferred side of co-selectors 1 and 2 to storage entry 7.

11. Counter exit 3 and counter exit 6 are wired through the normal side of co-selector 7 to storage entry 7 for punching.

12. Storage punch exit 7 is the product field for punching. The two high-order positions are wired directly from the first two positions of 7L. The remaining 20 positions are split-wired.

## DIVISION

A MAXIMUM of six quotient digits may be calculated in a single dividing operation. The quotient may be expanded, however, to as many more digits as desired by treating the problem in parts. It is not possible to divide with a divisor larger than eight digits. To do so would require expanding the dividend counter by the same number of digits, regardless of the size of the dividend field in the card. Since counter 6 cannot be coupled with counters 1, 2, and 3, the problem is not possible when the divisor exceeds eight positions.

### Quotient Expansion

The following example (Figure 67) shows how a 13-digit dividend can be divided by a 2-digit divisor to obtain a 12-digit quotient (unadjusted). This problem cannot be calculated in a single dividing operation even though the dividend counter is large enough to accommodate the 13-digit dividend.

The rule stated under "Size of Factors in Division" is that the dividend must never be larger than the number of significant digits in the divisor, plus seven. It follows, therefore, that, with a divisor with two significant digits, the dividend is limited to nine digits, and, if the divisor has one significant digit, the dividend is limited to eight digits. Furthermore, in this example, the size of counter 6 limits the development of a quotient of more than six digits on one step. In this example, the dividend must be handled in parts with seven digits to the left in the first dividing operation, and six digits to the right in the second operation, as follows:

$$1234567123456 \div 12 = 102880593621$$

```
FIRST DIVIDING OPERATION                SECOND DIVIDING OPERATION
          102880                                  593621
   12 √ 01234567                           12 √ 07123456
         12                                      60
         --                                      --
           34                                    112
           24                                    108
           --                                    ---
           105                                     43
            96                                     36
            --                                     --
             96                                     74
             96                                     72
             --                                     --
              07 Remainder                           25
                                                     24
                                                     --
                                                      16
                                                      12
                                                      --
                                                       4 Remainder
```

If the quotient is to be adjusted, the dividend must be positioned in the dividend counter to provide for the extra quotient position that must be developed. Since the dividend is handled in parts, the second part must be shifted into the proper position of the dividend counter and the second part of the quotient half adjusted in the units position.

### Planning Chart and Control Wiring

*Read.* The 13-digit dividend is split into two parts. The seven high-order positions are added in the dividend counter and the remaining six positions are entered into storage unit 2. The divisor is entered into 1R.

*Program 1.* The first part of the dividend is divided by 12 and the quotient is developed in counter 6. The divide hub is impulsed, the divisor is read out, the dividend counter subtracts, and counter 6 adds the quotient.

*Program 2.* The remainder standing in the dividend counter must be retained so that it can be placed beside the second part of the dividend, before proceeding with the next divide operation. The remainder in the dividend counter is read out and entered into storage unit 4. The dividend counter is also reset. At the same time, the quotient for the first part of the problem is read into storage unit 7 and punched.

*Program 3.* The second part of the original dividend is read out of storage unit 2R and added into the dividend counter.

*Program 4.* The second part of the original dividend and the remainder from the first dividing operation are divided by 12, and the second part of the

| PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT<br>1L | DIVR.-MULT.<br>1R | COUNTER<br>DIVIDEND<br>1 | <br><br>2 | <br><br>3 | <br><br>4 | <br><br>5 | <br><br>6 | STORAGE UNITS<br>2L | <br>2R | <br>3L | <br>3R | <br>4L | <br>4R | PUNCH UNITS<br>*UNITS POSITION MUST BE WIRED TO PUNCH<br>6L | <br><br>6R | <br><br>7L | <br><br>7R |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | READ CYCLE | | RI<br>12 | + | +<br>1234567 | + | | | | | RI<br>123456 | | | | | | | | |
| | 1 | DIVIDE | | RO | −<br>REMAINDER 07 | − | − | | | +<br>102880 | | | | | | | | | | |
| | 2 | | | | RO<br>RE | RO<br>RE | RO<br>RE | | | RO<br>RE | | | | | | RI<br>REMAINDER<br>07 | | | | RI<br>PUNCH<br>102880 |
| | 3 | | | | + | RI+<br>07123456 | + | | | | | RO | | | | RO | | | | |
| | 4 | DIVIDE | | RO | − | − | − | | | +<br>593621 | | | | | | | | | | |
| | 5 | | | | RE | RE | RE | | | RO<br>RE | | | | | | | | | | RI<br>PUNCH<br>593621 |
| | 6 | | | | | | | | | | | | | | | | | | | |

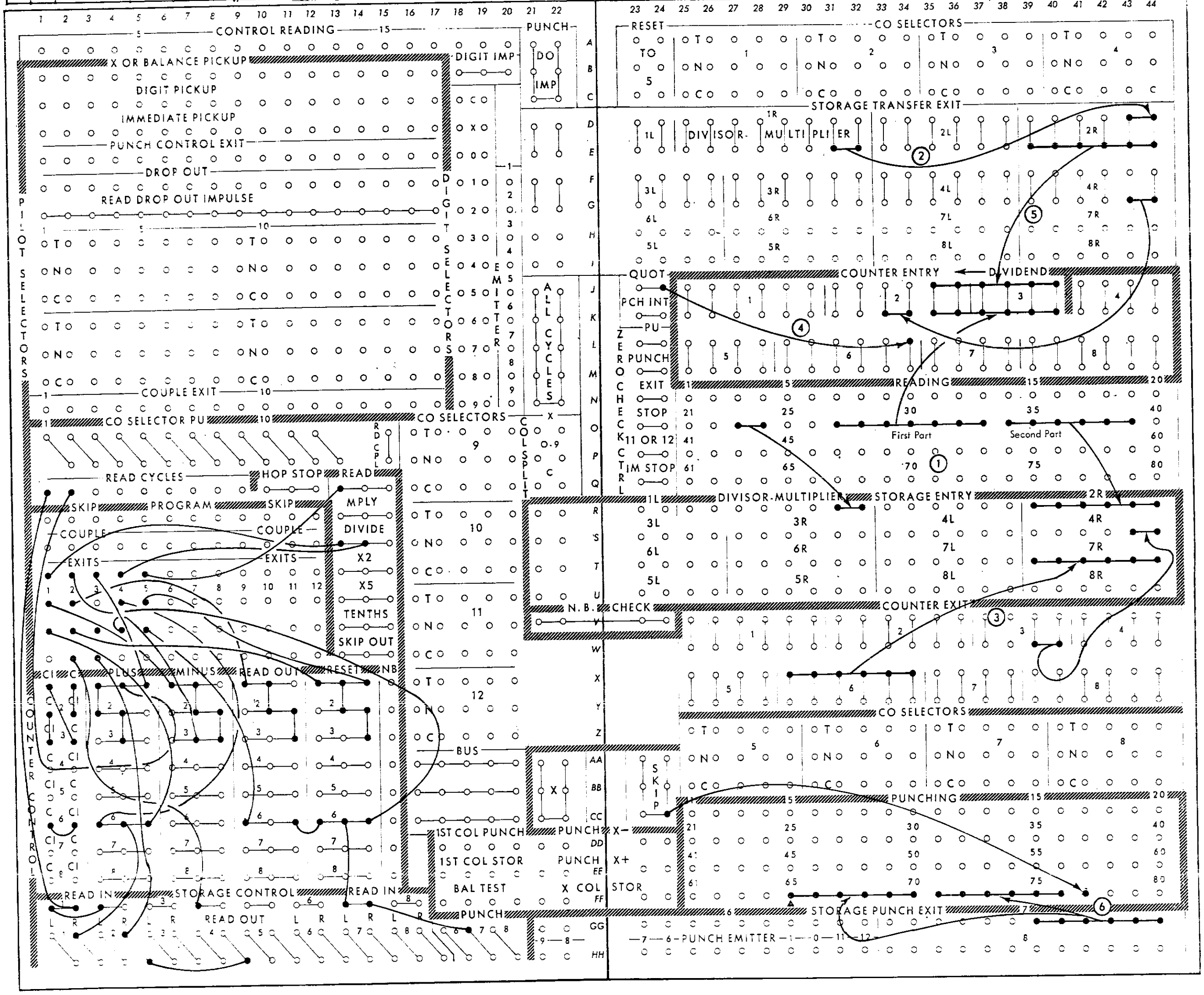


*Figure 67. Quotient Expansion, Series 50*

quotient is developed in counter 6. The divide hub is impulsed, the divisor is read out, the dividend counter subtracts, and counter 6 adds.

*Program* 5. The dividend counter is reset. The second part of the quotient is read out of counter 6 to storage unit 7 and punched. The first part of the quotient is punched while the second part is being calculated. The last program is wired to READ.

### Right-Hand Panel Wiring

1. The divisor is wired to 1R. The first part of the dividend is wired to the dividend counter and the second part to storage entry 2R.

2. The divisor exit is wired to the dividend counter.

3. The remainder from the first division is wired to 4R and the first six positions of the quotient are wired to storage unit 7R for punching.

4. The quotient hub is wired to counter 6 entry.

5. The remainder from the first division is wired from storage exit 4R to the dividend counter and the remainder of the dividend is wired from storage exit 2R to the dividend counter.

6. Storage punch exit 7 is split-wired to the quotient field. Because storage punch exit 7R is used for punching both portions of the quotient, care must be taken to be certain the right-hand position of the storage punch exit is wired to a punching position each time the unit is impulsed to punch.

## PROGRAM EXTENSION

ALTHOUGH a Series 50 602 provides for only six program exits, once a read cycle is taken by the machine, programming continues until READ is impulsed. In all problems which do not require more than six programs, programming is stopped at the end of a problem by impulsing READ on the last program taken.

If READ is not impulsed, programming continues indefinitely and program exits 1 to 6 become program exits 7-12 on the second round, 13 to 18 on the third round, and so on. It then becomes possible to extend programs beyond six by controlling the impulse wired to READ.

Program 1 exits are always active on a repeated round, and, therefore, cannot be skipped after the first round of programs, but program 1 exits on repeated rounds cannot be initiated unless program 6 exits are active on the preceding round. In other words, in all problems requiring extension of programs, program exit 6 cannot be skipped.

Program exits used on more than one round of programs may need to be selected to determine their identity. For example, program 1 exits on the first round become program 7 exits on the second round. Selection of program 1 exits is necessary, therefore, to distinguish between program 1 on the first round and program 7 on the second round. This selection is necessary only if the functions to be performed on these programs differ. If functions are to be identical, selection is not necessary.

Figure 68 illustrates a method of extending programs up to 11; program exits 1 through 5 are used on both rounds to perform different functions, and program 6 is used only on the first round.

Figure 69 illustrates a method of extending programs up to 17. Program exits 1 through 5 are used on the first round as programs 1-5, on the second round as 7-11, on the third round as programs 13-17; program 6 exits are used only on the first two rounds, as program 6 on the first and as program 12 on the second. As in Figure 68, co-selector positions are necessary to identify programs (1, 2, 3 and so on) on the first round; in addition, co-selector positions transferred for the third round are required to distinguish between programs 7 and 13, 8 and 14, 9 and 15, and so on. Five pilot selectors are required to control the co-selectors for three rounds.

### Control Panel Wiring (Figure 68)

1. Program couple 6 is wired to the D pickup of pilot selector 1. Since the selector is wired for normal drop-out, the selector transfers on program 6 and remains transferred for the remainder of the problem.

2. When pilot selector 1 is transferred, the couple exit of pilot selector 1 emits an impulse which is used, in turn, to pick up co-selectors.

3. Co-selector 2 is used to distinguish between programs 1 and 7. Impulses are emitted on both programs from program 1 exits. When the exits are wired to the common hubs of co-selector 2, the nor-

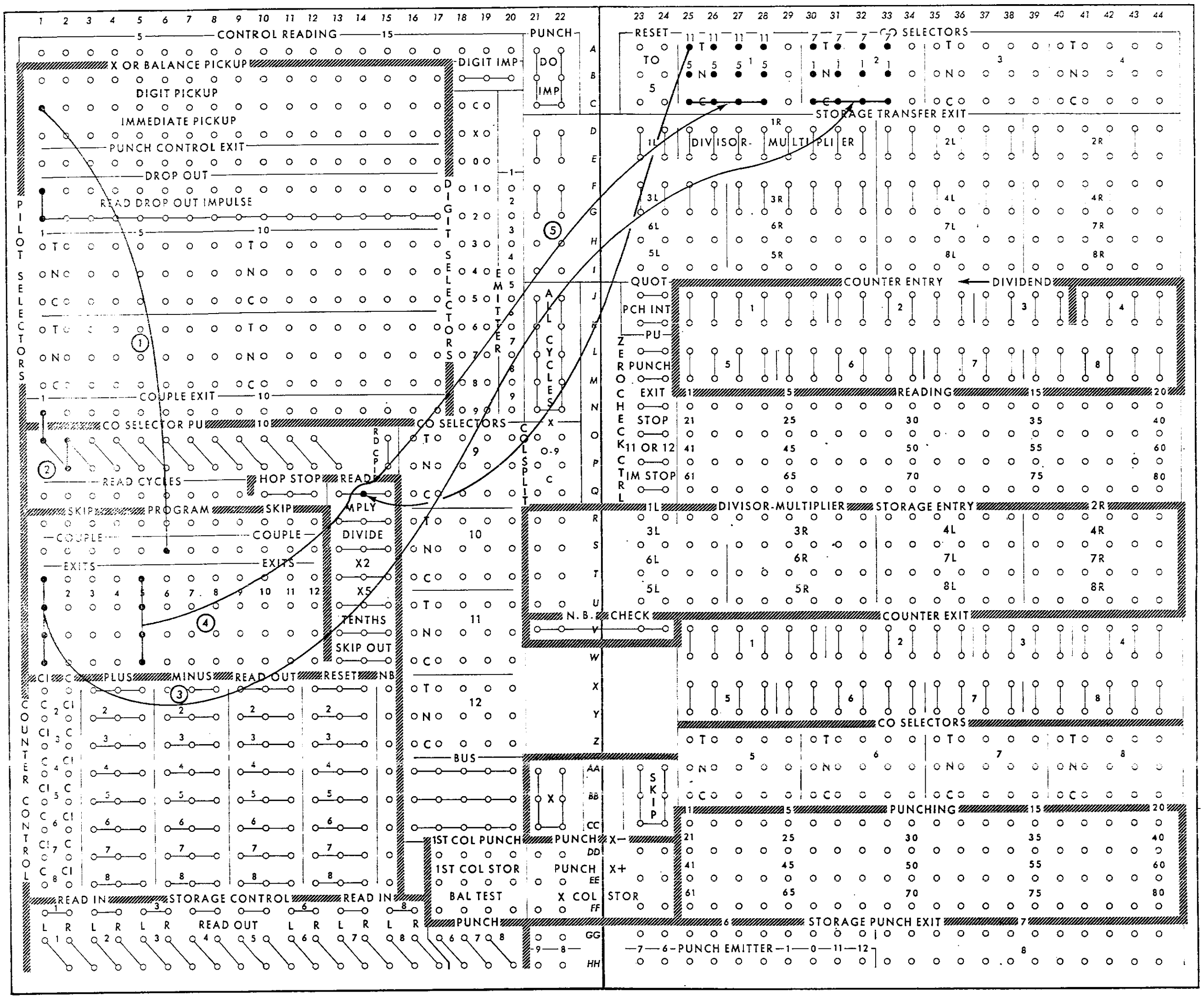


Figure 68. Program Extension, 11 Programs

mal hubs provide exits for program 1 and the transferred hubs provide exits for program 7.

4. Co-selector 1 is used to distinguish between programs 5 and 11. Impulses are emitted on both programs from program exit 5. When these exits are wired to the common hubs of co-selector 1, the normal hubs provide exits for program 5 and the transferred hubs provide exits for program 11.

5. Program 11 is wired to READ. Any selection required for program exits 2 through 4 is accomplished in a manner similar to that shown for program exits 1 and 5.

## Control Panel Wiring (Figure 69)

1. On the first round of programs all selectors are normal, and program 1 exits pass through the normal side of co-selector 4.

2. On program 6 of the first round, program couple 6, wired to the D pickup of pilot selector 1, transfers the selector immediately, and the selector remains transferred for the remainder of the problem.

3. On program 7, program couple 1 passes through the transferred side of pilot selector 1 and transfers pilot selector 2 and co-selector 4 immediately. Both selectors remain transferred for the remainder of the problem. Program 1 exits (program 7) pass through the transferred side of co-selector 4 and the normal side of co-selector 3.

4. On program 12, program couple 6 passes through the transferred side of pilot selector 2 and picks up pilot selector 3 immediately; selector 3 remains transferred for the remainder of the problem.

5. On program 13, program couple 1 passes through the transferred side of pilot selector 3 and transfers pilot selector 4 and co-selector 3 immediately. Both selectors remain transferred for the remainder of the problem. Program 1 exits (program 13) pass through the transferred sides of co-selectors 4 and 3.

6. As in the wiring for program 1 exits, program 5 exits are identified as programs 5, 11 or 17 by wiring through co-selectors 2 and 1.

7. Program 17 is the last program taken. It is wired to READ and also to the immediate pickup of

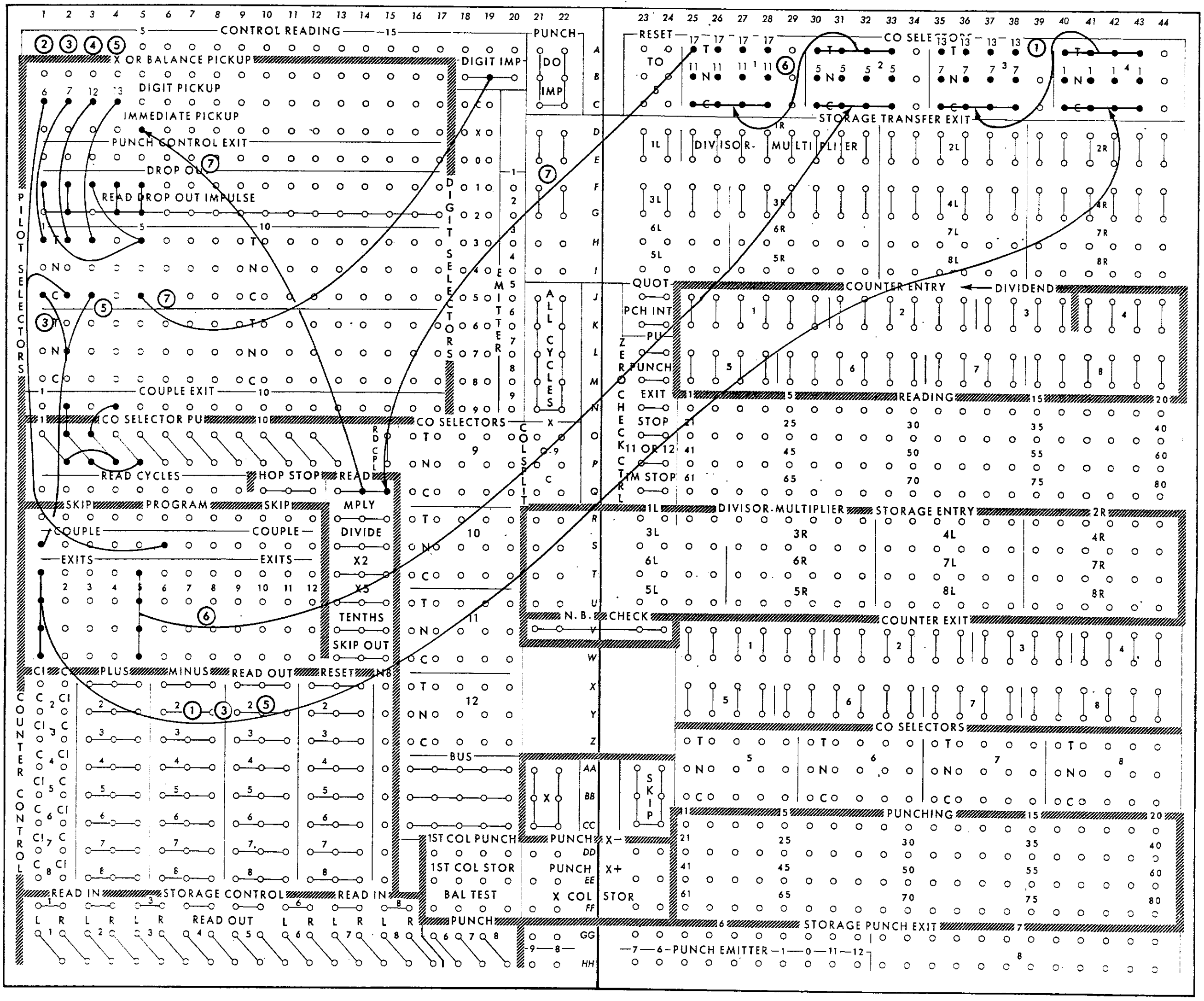


*Figure 69. Program Extension, 17 Programs*

pilot selector 5. The digit impulse passing through the transferred side of this selector drops out pilot selectors 1 and 3. When pilot selectors are used to select program couple 1, normal read drop-out cannot be used because program couple 1 would pass partly through the normal side. This timing interference would cause erroneous results. Thus, it is necessary to drop out selectors 1 and 3 at the end of the last program taken. Program exits cannot be used to drop out pilot selectors because the early portion of the program cycle would burn the selector points.

Wiring for selection of any other program exits is the same as that shown for program exits 1 and 5.

# Appendix

## METHOD OF MULTIPLICATION

When one digit is multiplied by another digit, the result or product is made up of right and left components. In the multiplication 5 × 9 = 45, the 5 is the right component and the 4 is the left component. A study of the multiplication table (Figure 70) will show that the products of 1 times the multiplicands have no left components and no right components greater than 9. The products of 2 times the multiplicands show no left components higher than 1 and no right components higher than 8. The products of 5 times the multiplicands show no left components greater than 4 and no right components greater than 5. No possible combination of right and left components for anything multiplied by 1, 2, or 5 can exceed 9.

This is not true when multiplying by 3, 4, 6, 7, 8, and 9. For example, when multiplying by 3, the highest left component is 2 and the highest right is 9; 9 + 2 = 11.

If the sum of the right and left components of a multiplication by a single digit does not exceed 9, they can be accumulated into a single counter position in a single cycle, part of the cycle to enter the rght components and part of the cycle to enter the left components. For example:

```
 98  multiplicand
 ×2  multiplier
----
 18 }
 16 }one machine cycle
----
196  product
```

Multiplication starts from left to right, and in the example shown, the right component of 2 × 90 (8) and the left component of 2 × 8 (1) are both entered into the same counter position on the same cycle as shown by the figures in the illustration.

All multiplication on the 602 is done by multipliers of 1, 2, 5, 10, or certain combinations of them. Multiplication by ten is merely offsetting the multiplication by 1.

MULTIPLICAND

| MULTIPLIER | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 2 | 2 | 4 | 6 | 8 | 1 0 | 1 2 | 1 4 | 1 6 | 1 8 |
| 3 | 3 | 6 | 9 | 1 2 | 1 5 | 1 8 | 2 1 | 2 4 | 2 7 |
| 4 | 4 | 8 | 1 2 | 1 6 | 2 0 | 2 4 | 2 8 | 3 2 | 3 6 |
| 5 | 5 | 1 0 | 1 5 | 2 0 | 2 5 | 3 0 | 3 5 | 4 0 | 4 5 |
| 6 | 6 | 1 2 | 1 8 | 2 4 | 3 0 | 3 6 | 4 2 | 4 8 | 5 4 |
| 7 | 7 | 1 4 | 2 1 | 2 8 | 3 5 | 4 2 | 4 9 | 5 6 | 6 3 |
| 8 | 8 | 1 6 | 2 4 | 3 2 | 4 0 | 4 8 | 5 6 | 6 4 | 7 2 |
| 9 | 9 | 1 8 | 2 7 | 3 6 | 4 5 | 5 4 | 6 3 | 7 2 | 8 1 |

*Figure 70. Multiplication Table*

Digits other than 1, 2, or 5 require two cycles, as shown in the table below:

| PLUS MULTIPLICATION | | | MINUS MULTIPLICATION | | |
|---|---|---|---|---|---|
| DIGIT | CYCLE 1 | CYCLE 2 | DIGIT | CYCLE 1 | CYCLE 2 |
| +3 | + 5 | −2 | −3 | − 5 | +2 |
| +4* | + 5 | −1 | −4 | − 5 | +1 |
| +6 | + 5 | +1 | −6 | − 5 | −1 |
| +7 | + 5 | +2 | −7 | − 5 | −2 |
| +8 | +10 | −2 | −8 | −10 | +2 |
| +9* | +10 | −1 | −9 | −10 | +1 |

*Units position only

Digits 4 and 9 require two cycles only when they are in the units position. In all other positions a method has been provided for multiplying by them in one cycle. When multiplying by a 4 in any position other than the units position, the machine treats it as a 5, because it takes only one cycle. After this, the signs for all of the remaining multiplier digits to the right are reversed and are treated in their complement form instead of true figures. For example:

| | | | | | |
|---|---|---|---|---|---|
| The multiplier | 3 | 4 | 6 | 2 | 7 |
| will be treated as | +3 | +5 | −3 | −7 | −3 |

When the 4 is reached in the thousands position, it is treated as a 5. Multiplication by 35,000 is 373 times more than 34627, and so the remaining digits are multiplied by —373 to compensate for the overage.

Assuming that the number 34627 has for its multiplicand the digit 1, the result would be developed as follows:

```
1 × 3        1 × +5 =        50000
             1 × —2 =      —2
                           -------
                             30000
1 × 4        1 × +5 =         5
                           -------
                             35000
1 × —3       1 × —5 =       —5
                           -------
                             34500
             1 × +2 =       +2
                           -------
                             34700
1 × —7       1 × —5 =        —5
                           -------
                             34650
             1 × —2 =        —2
                           -------
                             34630
1 × —3       1 × —5 =         —5
                           -------
                             34625
             1 × +2 =          2
                           -------
                             34627
```

The digit 9 follows the same rule as the digit 4. When it is sensed in any position other than the units position, it reads as a 10, because multiplication by 10 can be done in one cycle. All digits to the right have their signs reversed until another 4 or 9 is read, at which time the signal is given for another sign change. Sign reversal and sign restoration may take place several times during the multiplication.

Only single zeros in the multiplier are skipped. For every two adjacent zeros, an idle cycle occurs before the next digit is sensed. Zeros to the right of the lowest-order significant digit are also skipped. Consider the following examples:

| | |
|---|---|
| 3 0 8 0 7 | No time lost for zeros. |
| 3 0 0 8 0 | 1 cycle for two adjacent zeros. |
| 3 8 0 0 0 | No time lost for zeros to the right of lowest order significant digit. |

DIVIDEND HIGH-ORDER DIGIT

DIVISOR HIGH-ORDER DIGIT

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 2 | 2 | 2 | 5 | 5 | 5 | 5 | 5 |
| 2 | .5 | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 5 |
| 3 | .5 | .5 | 1 | 1 | 2 | 2 | 2 | 2 | 2 |
| 4 | .2 | .5 | 1 | 1 | 1 | 1 | 2 | 2 | 2 |
| 5 | .2 | .5 | .5 | 1 | 1 | 1 | 1 | 2 | 2 |
| 6 | .2 | .5 | .5 | .5 | 1 | 1 | 1 | 1 | 1 |
| 7 | .2 | .2 | .5 | .5 | .5 | 1 | 1 | 1 | 1 |
| 8 | .2 | .2 | .5 | .5 | .5 | 1 | 1 | 1 | 1 |
| 9 | .2 | .2 | .5 | .5 | .5 | .5 | 1 | 1 | 1 |

QUOTIENT

IDENTICAL VALUES FOR 7, 8

*Figure 71. Quotient Selection Table*

The number of machine cycles required in multiplication depends upon three things: first, the size of the multiplier; second, whether the digits in the multiplier are 1, 2, or 5 which require one cycle, or 3, 6, 7, or 8 which always require two, or 4 and 9 which might require two; and third, the presence of zeros and their position.

Speed of multiplication can be reasonably determined by using an average of 1.4 cycles per multiplier digit.

## METHOD OF DIVISION

When dividing, the machine makes a comparison between the highest-order divisor digit and the highest-order dividend digit to determine a quotient digit.

$$833059 \div 297 = 2804 \frac{271}{297}$$

In the above example, *2* is the highest-order digit of the divisor. It is compared with 8, the highest-order digit of the dividend. A quotient digit of 2 is selected according to the table. The quotient digit 2 is multiplied by the divisor, and the result is subtracted from the dividend in one cycle.

```
                          2
              (2) 97√(8)3 3 0 5 9
2 × divisor          —5 9 4
                     -----------
                      2 3 9 0 5 9   remaining dividend
```

Since the high-order position of the remaining dividend is not a zero, no shifting takes place, and another comparison is made, this time 2 with 2. According to the table (Figure 71) the quotient digit selected is 1. The 1 is multiplied by the divisor, the product is subtracted from the remainder, and one is added to the quotient digit, increasing it from 2 to 3.

```
                           3
              (2)97√(2)39059   remaining dividend
1 × divisor        —2 97
                   --------
                   —0 57941
```

The remainder is now minus. Because there is a zero in the high-order position, a shift takes place. The 2 is then compared with 5 and, according to the table, a quotient digit of 2 is selected. Because the remainder is negative, the product of 2 times the divisor is added to the dividend, and 2 is subtracted from 30 in the quotient. The new quotient is now 28.

```
                              28
              (2)97√—0(5)7941   remaining dividend
2 × divisor        +   5 94
                   ---------
                      0 1459
```

The remainder again is plus. Another shift takes place because of the zero in the high-order position. Comparing 2 with 1, a quotient digit of .5 is selected. The .5 is the same as the 5, except that it causes an additional shift to the right, resulting in total shift of two places in the quotient. The product of .5 × 297 is subtracted from the remaining dividend.

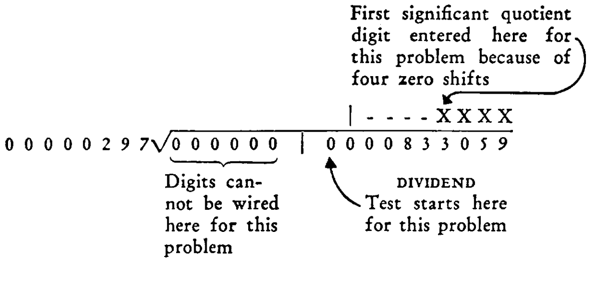
```
                        2 805
                  (2)97√0(1)459   remaining dividend
.5 × divisor           — 1 485
                       — 0 026
```

When the last quotient digit is selected, the dividend remainder may be positive or negative, but division is never completed until the dividend remainder turns positive and is less than the divisor. When the dividend remainder is negative, the divisor is added to it, and a 1 is subtracted from the quotient, reducing it from 2805 to 2804.

```
                       28 0 4
                 (2)97√—00(2)6   remaining dividend
1 × divisor           + 2 9 7
                      + 2 7 1   end of division
```

## Size of Factors in Division

Consideration must be given to the size of the dividend and the divisor because of the maximum 8-digit quotient. The machine is able to shift seven times when placing the quotient; therefore, by adding seven to the number of digits in the divisor, a determination can be made as to where the test of the high-order digits will be made. For example, if the divisor has three significant digits, the test for the positioning of the proper quotient digit starts in the tenth position of the dividend. If the dividend is wired for a 14-column field, the four positions to the left of the tenth place would not be recognized in the test and would show incorrect remainders throughout the problem, thus developing a wrong quotient. The rule, then, may be stated as follows: the dividend must never be larger than the number of significant digits in the divisor plus seven.


QUOTIENT
First significant quotient digit entered here for this problem because of four zero shifts
| - - - - X X X X
0 0 0 0 0 2 9 7√0 0 0 0 0 0 | 0 0 0 0 8 3 3 0 5 9
Digits cannot be wired here for this problem
DIVIDEND
Test starts here for this problem

Although the number of significant digits in this problem is three, other cards may have a divisor of one significant digit, in which case the test would begin in the eighth position (7 + 1) of the dividend. Any digits in positions 9-16 of the dividend would cause the quotient to be developed incorrectly. Not only would those positions fail to be tested, but each remainder would be wrong, thereby developing an incorrect quotient. In the case of an 8-digit divisor, the test would start in the 15th dividend position (8 + 7).

Therefore, both the quotient and the dividend must be consistent in size with the number of digits in the divisor, as shown by the following examples:

$$\frac{99{,}999{,}999}{1} = 99{,}999{,}999$$

$$\frac{99{,}999{,}999{,}999}{1000} = 99{,}999{,}999$$

$$\frac{999{,}999{,}999{,}999{,}999}{10{,}000{,}000} = 99{,}999{,}999$$

In each of the above examples, the dividend is the largest possible and the divisor is the smallest possible to develop an 8-digit quotient.

## METHOD OF ESTIMATING PRODUCTION SPEEDS

THE NUMBER of cycles per card as well as the number of columns to be punched determine the speed of the 602 for any given problem. The maximum speed is 3000 cards an hour; it is obtained for all problems requiring from one to four cycles to calculate and from one to two columns to be punched. The speed then decreases proportionately with the increase in both the number of cycles taken and the number of columns punched. Generally, the approximate number of cards per hour is obtained by dividing cycles per card into 12,000.

Production speed for a given problem can be best estimated by first analyzing the problem on a planning chart (Figure 72). Each step in the problem except multiply and divide is counted as one cycle. Obtain the number of cycles for multiplication and division for a given number of digits by reference to Table 1 (multiplication) and Table 2 (division). Next, add the number of cycles for all steps as shown by the planning chart. In case of fractions, drop the fraction and increase the number of cycles by 1. Then determine the approximate production speed and the maximum number of columns that can be punched to obtain that speed, by reference to Table 3 (approximate speed). The three tables and a planning chart on which the number of cycles have been estimated are illustrated in Figure 66.

Table 1. Multiplication

| NO. OF DIGITS IN MULTIPLIER | AVG. NO. OF CYCLES |
|---|---|
| 1 | 1.4 |
| 2 | 2.8 |
| 3 | 4.2 |
| 4 | 5.6 |
| *5 | 7.0 |
| 6 | 8.4 |
| 7 | 9.8 |
| 8 | 11.2 |

Table 2. Division

| NO. OF DIGITS IN QUOTIENT | AVG. NO. OF CYCLES |
|---|---|
| 1 | 5.4 |
| 2 | 5.4 |
| 3 | 6.3 |
| ‡4 | 7.2 |
| 5 | 8.1 |
| 6 | 9.0 |
| 7 | 9.9 |
| 8 | 10.8 |

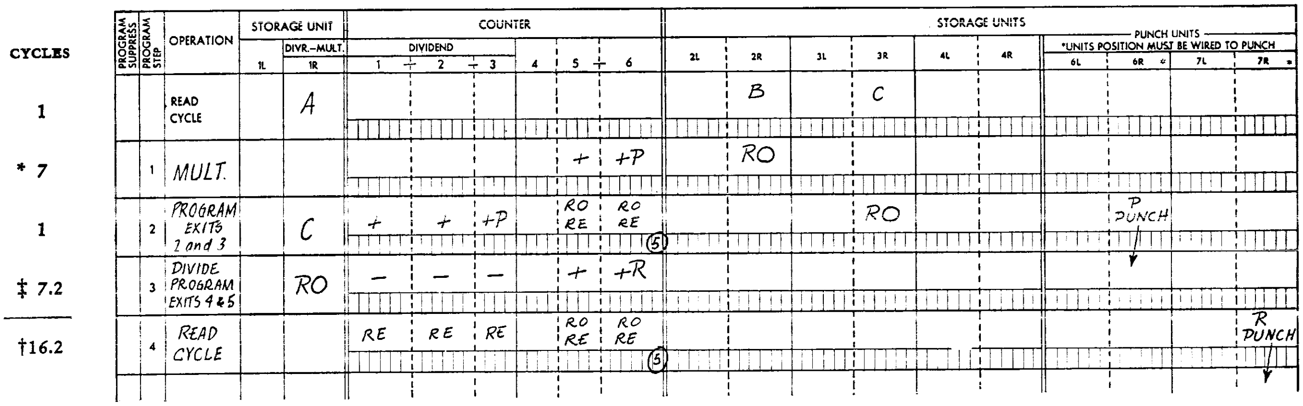
Table 3. Approximate Speed

| CYCLES PER CARD | APPROX. CARDS PER HOUR | MAX. COL. PUNCHED |
|---|---|---|
| 1-4 | 3000 | 2 |
| 5 | 2400 | 7 |
| 6 | 2000 | 11 |
| 7 | 1714 | 15 |
| 8 | 1500 | 19 |
| 9 | 1333 | 24 |
| 10 | 1200 | 28 |
| 11 | 1090 | 32 |
| 12 | 1000 | 36 |
| 13 | 923 | 40 |
| 14 | 857 | 44 |
| 15 | 800 | 48 |
| 16 | 750 | 52 |
| †17 | 705 | 56 |
| 18 | 666 | 60 |
| 19 | 631 | 64 |
| 20 | 600 | 68 |
| 21 | 571 | 72 |
| 22 | 545 | 76 |
| 23 | 521 | 80 |

* 7 cycles (5-digit multiplier — see Table 1)

‡ 7.2 cycles (4-digit multiplier — see Table 2)

†16.2 cycles (705 cards per hour — see Table 3)



| CYCLES | PROGRAM SUPPRESS | PROGRAM STEP | OPERATION | STORAGE UNIT | | COUNTER | | | | | | STORAGE UNITS | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | | DIVR.-MULT. | DIVIDEND | | | | | | | | | | | | PUNCH UNITS *UNITS POSITION MUST BE WIRED TO PUNCH | | | |
| | | | | 1L | 1R | 1 | 2 | 3 | 4 | 5 | 6 | 2L | 2R | 3L | 3R | 4L | 4R | 6L | 6R * | 7L | 7R * |
| 1 | | | READ CYCLE | | A | | | | | | | | B | | C | | | | | | |
| * 7 | | 1 | MULT. | | | | | | | + | +P | | RO | | | | | | | | |
| 1 | | 2 | PROGRAM EXITS 2 and 3 | | C | + | + | +P | | RO<br>RE | RO<br>RE<br>⑤ | | | | RO | | | | P<br>PUNCH | | |
| ‡ 7.2 | | 3 | DIVIDE PROGRAM EXITS 4 & 5 | | RO | − | − | − | | + | +R | | | | | | | | | | |
| †16.2 | | 4 | READ CYCLE | | | RE | RE | RE | | RO<br>RE | RO<br>RE<br>⑤ | | | | | | | | | | R<br>PUNCH |

*Figure 72. Planning Chart*

$A \times B = P; \quad P \div C = R$

## TIMING CHARTS

The purpose of timing charts (Figures 73 and 74) is to increase the general knowledge of machine operation and to assist experienced operators who must resort to unusual wiring methods to accomplish a desired result. A good working knowledge of the machine is necessary before timing charts can be used effectively.

*Cycle.* A cycle is a period of time required for a given series of events, at the completion of which the series is repeated. Since the machine operates at a speed of 200 cycles per minute, one cycle requires three-tenths of a second. Each cycle is divided into 16 equal parts called "points," each point consisting of 22½ degrees. There are 360° from one given point of a cycle to the same point of the next cycle. Since the functional cycle in this machine begins at approximately 356°, the timing charts show a range of 356° to 356°.

Basically, there are only two type of machine cycles: read cycles and program cycles. Various modifications of these cycles shown in the cycle column of the timing charts are explained in the timing chart notes.

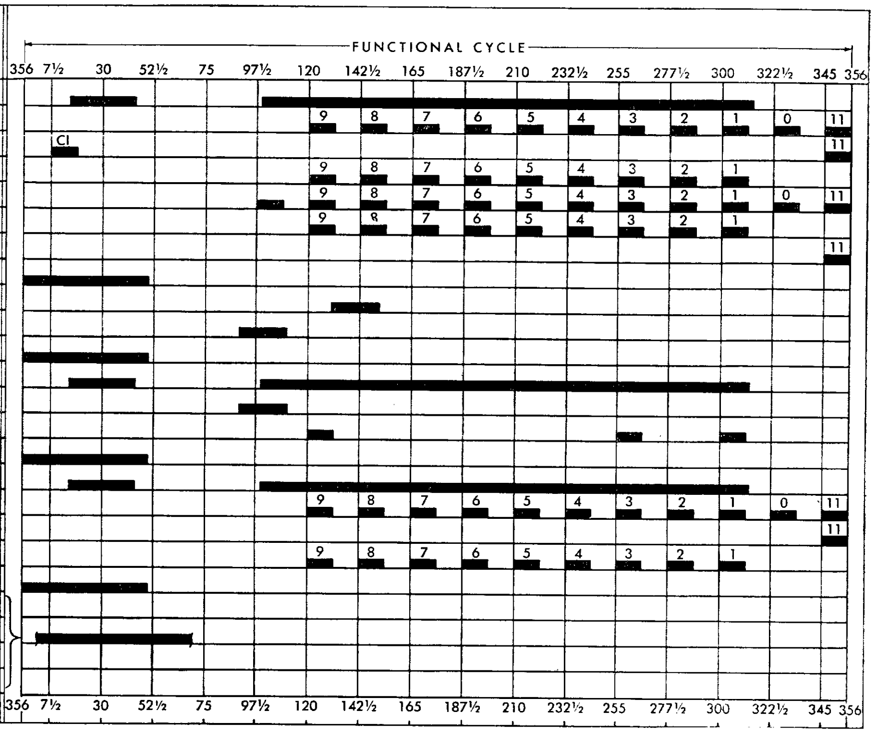
*Exit and Entry Hubs.* Exit hubs (Figure 67) emit impulses, and entry hubs (Figure 68) receive impulses at certain times during the various machine cycles. Thus, in wiring a control panel it is necessary to consider not only the exit and entry time of the two hubs to be connected, but also the machine cycle in which the timing is effective. For example, suppose a program exit hub is to be wired to the skip-out hub. The timing chart shows that program exits emit from 100° to 312° and that the skip-out hubs are receptive at this time; however, program exits emit on program cycles, and skip-out hubs receive only on read cycles. Thus, skip-out cannot be impulsed from a program exit.

Most machine functions are controlled by impulses emitted from read cycles, program exits, or all cycles hubs. The timing of these impulses is from 15° to 45° (setup impulse), and from 100° to 312° (entry control). The setup impulse prepares the machine for the operation that takes place later in the cycle, or causes a non-entry control cycle if the machine is not ready to perform the operation.

The entry control impulse controls read-in, read-out, reset, adding, subtracting, multiplying, and dividing. It generally ends at 312°, but when the machine multiplies by 2 or 5, it ends at 334°.

*Selectors.* The X or balance pickup hub of a pilot selector accepts an X impulse to cause the selector to transfer at 355° and remain transferred until 356° of the cycle during

| CONTROL PANEL HUB EXIT<br>▬ EXIT | LOCATION | CYCLE | NOTES |
|---|---|---|---|
| All Cycles | J-N, 21-22 | A | 1 |
| Control Reading | A, 1-20 | C | |
| Counter Control-CI | X-EE, 1-2 | A | 2 |
| Counter Exit | V-Y, 25-44 | EC | |
| Digit Impulse | B, 18-20 | EC | |
| Emitter | F-N, 20 | EC | |
| Negative Balance (NB) | X-EE, 15 | EC | 3 |
| Pilot Selector — Couple Exit | N, 1-17 | A | 4 |
| Punch Control Exit | E, 1-17 | | 5 |
| Read Drop Out Impulse | G, 1-17 | R | 6 |
| Program Couple | S, 1-12 | P | 7 |
| Program Exit | T-W, 1-12 | P | 1 |
| Punch Drop Out Impulse | A-C, 21-22 | | 8 |
| Quotient | J, 23-24 | D | 9 |
| Read Couple | O-P, 15 | R | |
| Read Cycles | Q, 1-9 | R | 1 |
| Reading | N-Q, 25-44 | R | |
| Reset to Five | A-C, 23-24 | EC, C | |
| Storage Transfer Exit | D-I, 21-44 | EC | |
| Zero Check Control Exit | N, 23-24 | EC | 10 |
| Balance Test | FF, 17-20 | Pch | 11 |
| First Column Storage | EE, 17-20 | Pch | 12 |
| Punching | CC-FF, 25-44 | Pch | 13 |
| X Column Storage | FF, 21-24 | Pch | 14 |



EXIT ▬

NORMAL ENTRY ▨

ENTRY ▭

CYCLE CODE — EXPLANATION

A All cycles (all program and read cycles).

C Control reading cycle, the cycle during which a card passes the control brushes. This cycle occurs during the first program cycle of the preceding card.

D Dividing cycle, a program or read cycle on which the machine is controlled to divide.

EC Entry control cycles, all program or read cycles except those in which the following conditions occur: storage delay cycle (occurs most commonly when a punch storage unit is impulsed to read in before punching from the unit has been completed), feed failure, or zero skip cycles in multiplication or division.

EC-C Entry control cycles except when multiplying or dividing.

P Program cycles, all cycles except read cycles.

PCH Punch control. Punching takes place at any time during program or read cycles. Operation bears no relationship to the machine cycle shown on the chart.

R Reading cycle, the cycle during which the card passes the reading brushes.

NOTES:

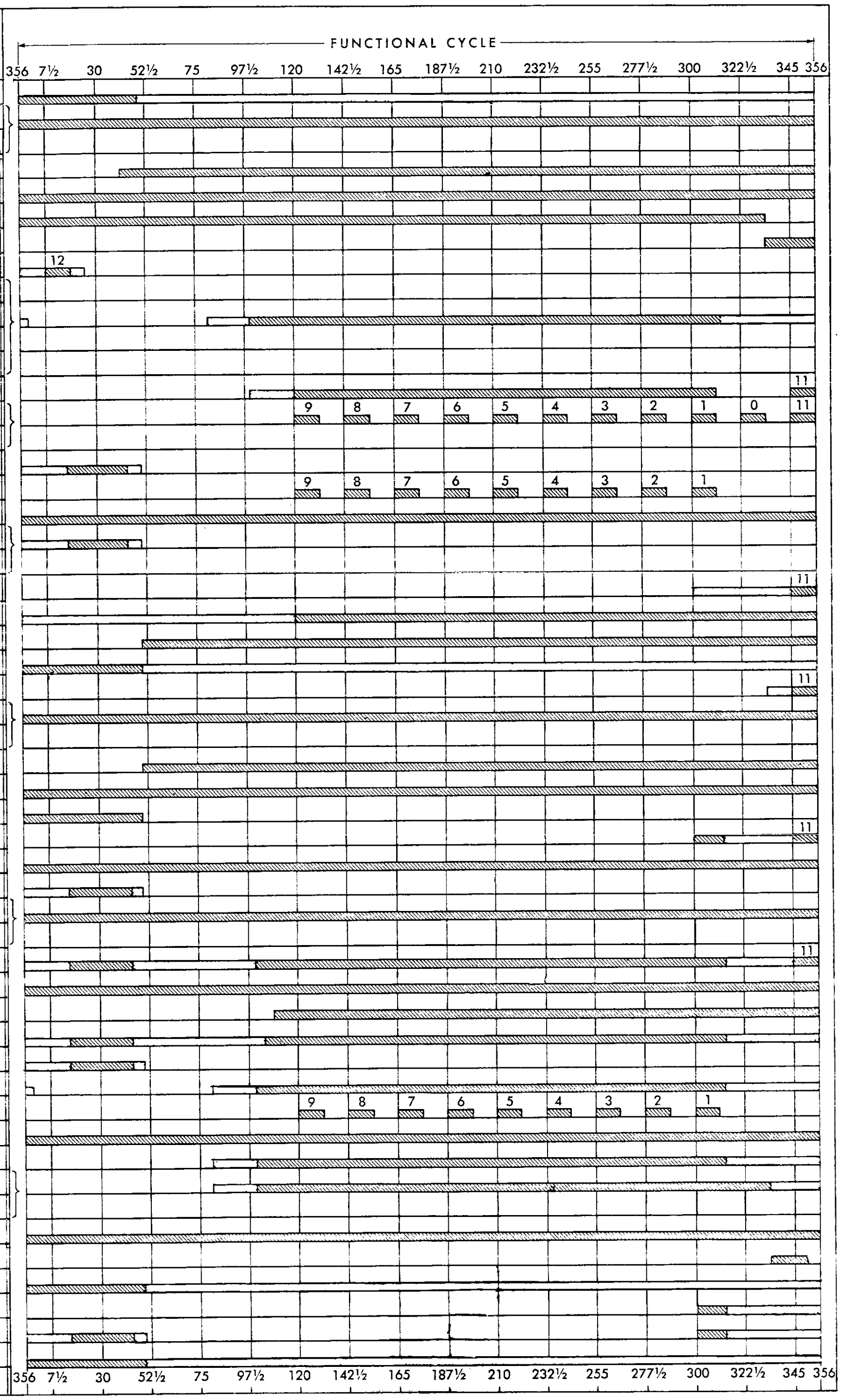
1. Timing normally ends at 312 degrees. When machine multiplies by 2 or 5, timing ends at 334 degrees.
2. The 11 impulse is the balance test impulse which occurs on every cycle during which the counter is negative or will go negative due to the carry impulse (CI) occurring at the beginning of the next cycle. It does not emit when the corresponding counter is controlled to reset.
3. Common with the C hub of corresponding counter at 11 time whenever the counter zero balances. Emits impulse when respective counter is negative or is about to go negative due to the carry impulse (CI) occurring at the beginning of the next cycle. It does not emit when the corresponding counter is controlled to reset.
4. Emits only when a pilot selector transfers from an X or D pickup.
5. Emits as one card is ejected into the stacker and another one is fed into punching position, provided the selector is transferred.
6. Also emits on run-in before the first card is read, and on run-out after the last card is punched.
7. These hubs are normally exits. They may be used as entries from other program couple hubs to expand program exits for a given program step.
8. Emits as one card is ejected into the stacker and another one is fed into punching position. Also emits on run-in and run-out at which time it is common with read drop-out.
9. The effect of these impulses is to add 5, 2, or 1 into the quotient counter. The changed timing of these impulses is the reason why other operations (except quotient multiplication as explained in the manual) cannot normally be performed on the same program with calculations.
10. Emits on the cycle when NB check or zero check control PU is impulsed, but not when both are impulsed.
11. Internally connected to first column punch until the balance test is completed. Therefore, it emits the punching impulse wired to first column punch as soon as storage punch exit becomes active.
12. Internally connected to first column punch after the balance test is completed. Therefore, it emits the punching impulse which is wired to first column punch.
13. For specific columns, punching hubs emit when corresponding columns of the card are in punching position. Punching impulses are interrupted as the card moves from one column to the next. Punching hubs do not emit while skipping or skipping out. Approximately 4 columns are punched in each machine cycle.
14. Emits impulse wired from punching to punch X plus or punch X minus.
15. Refer to write-up on Selectors,
16. Receives CI impulse. Common with NB at 11 time when all positions in the counter stand at 9. Common with CI at 1 to 8-time when all positions in the counter stand at 9. Inactive when corresponding counter is controlled to reset.
17. Only read cycles, program exits, or all cycles should be wired to these hubs.
18. Will accept an 11 impulse during reset to cause a reset to five. When performing simultaneous multiplication of several multiplicands, as many unused counter entry positions must be allowed between multiplicands as there are digits in the multiplier. Entries of two adjacent counter groups cannot be laced when multiplying in the left-hand counter, unless there are as many unused positions between the two entry wirings as there are digits in the multiplier. Failure to observe this rule may result in an erroneous product.
19. On machines with serial numbers under 22050EN, only interrupted impulses, such as digit impulse or card reading impulses should be wired to this hub. On machines with serial numbers 22050EN and over, all cycles, program exits, and read cycles impulses may also be wired to this hub.
20. Accepts impulse when cards run out of the hopper.
21. Active only when selector is normal and D pickup has not been impulsed on the same cycle.
22. Also active during multiply and divide zero skip cycles.
23. Normally entries from punching hubs to punch corresponding digits. Also exits when corresponding digits are punched from storage punch exit or zero check control 11-12. For this reason, punch emitter hubs should never be split-wired with any position from storage punch exit.
24. Accepts impulse for a given card on any program step until that card is placed into punching position. The impulse accepted by punch interlock causes programming to be suspended until the preceding card has finished punching.
25. Accepts punching impulse to cause an X to punch if a balance test was negative. This punching impulse is then internally connected to X column storage.
26. Accepts punching impulse to cause an X to punch if balance test was positive or not made. This punching impulse is then internally connected to X column storage.
27. The setup pulse initiates a storage delay cycle if punch is impulsed on the first program.
28. Read-out hubs for storage units 6, 7 and 8 should not be impulsed to read-out while punching. Since punching proceeds at approximately four columns per cycle, allow at least one machine cycle for each four columns to be punched before reading out of a storage punch unit.
29. The punching impulse is internally directed by storage punch exit to the punch magnet corresponding to the digit in punch storage (or the complement of this digit if a negative balance test was made).
30. Common with 11 of punch emitter.
31. Accepts a punching impulse to punch an 11 or 12.
32. Normally impulsed from zero check control exit.
33. The setup pulse has no effect except to cause a storage delay cycle if the previous zero check is not completed.

*Figure 73. Timing Chart Exits*

which drop-out is impulsed. The selector action is the same when the D pickup is impulsed from 50° to 355°. When the D pickup is impulsed from 355° to 50°, the selector transfers immediately and remains transferred until 356° of the cycle in which the drop-out hub is impulsed.

The immediate pickup of a pilot selector (if X or D has not been impulsed) or a co-selector may be impulsed at any time, and the selector transfers immediately. If the duration of the pickup impulse or any part of it falls between 41° and 356°, the selector remains transferred until 356°. Otherwise, the selector transfers and remains transferred only for the duration of the pickup impulse.

| CONTROL PANEL HUB ENTRY<br>▨ NORMAL<br>□ OTHER | LOCATION | CYCLE | NOTES |
|---|---|---|---|
| Co-Selector Pickup | O-P, 1-13 | A | 15 |
| C (Common) | C-BB, 25-44 | A | 15 |
| N (Normal) | B-AA, 25-44 | A | 15 |
| T (Transferred) | A-Z, 25-44 | A | 15 |
| Column Split C | Q, 21-22 | A | |
| 0-9 | P, 21-22 | A | |
| X | 0, 21-22 | EC, C | |
| Counter Control C | X-EE, 1-2 | A | 16 |
| Minus | X-EE, 6-8 | EC | 17 |
| Plus | X-EE, 3-5 | EC | 17 |
| Read Out | X-EE, 9-11 | EC | 17 |
| Reset | X-EE, 12-14 | EC | 17 |
| Counter Entry | J-M, 25-44 | EC | 18 |
| Digit Selector C | C, 18-19 | A | 19 |
| X-9 | D-N, 18-19 | A | |
| Divide | S, 13-15 | A | |
| Divisor Multiplier Entry | R, 25-32 | EC | |
| First Column Punch | DD, 17-20 | Pch | 11 |
| Hopper Stop | Q, 10-12 | | 20 |
| Multiply | R, 13-15 | A | |
| Negative Balance (NB) Check | V, 21-24 | EC | |
| Pilot Selector — Digit Pickup | C,.1-17 | A | 15 |
| Drop Out | F, 1-17 | A | 15 |
| Immediate Pickup | D, 1-17 | A | 21 |
| X or Bal. Pickup | B, 1-17 | EC, C | 15 |
| C (Common) | J-M, 1-17 | A | 15 |
| N (Normal) | I-L, 1-17 | A | 15 |
| T (Tranf'd, I P.U.) | H-K, 1-17 | A | 15 |
| T (Transf d, Latched) | H-K, 1-17 | A | 15 |
| Program Couple | S, 1-12 | P | 7 |
| Program Skip | R, 1-12 | EC | 22 |
| Punch Emitter 9-12 | HH, 21-32 | Pch | 23 |
| Punch Interlock | K, 23-24 | P | 24 |
| Punch X Minus | DD, 21-24 | Pch | 25 |
| Plus | EE, 21-24 | Pch | 26 |
| Read * | Q, 13-15 | A | |
| Skip | AA-CC, 23-24 | Pch | |
| Skip Out | W, 13-15 | R | |
| Storage Control — Punch (6, 7, 8) | GG-HH, 17-20 | EC | 27 |
| Read In | FF, 1-16 | A | |
| Read Out | GG-HH, 17-20 | EC | 28 |
| Storage Entry | R-U, 21-44 | A | |
| Storage Punch Exit (6, 7, 8) | GG, 21-44; HH, 33-44 | Pch | 29 |
| Tenths (Dividing Machine Only) | V, 13-15 | EC, C | |
| Times Five (X5) | U, 13-15 | EC, C | |
| Times Two (X2) | T, 13-15 | EC, C | |
| X | AA-CC, 21-22 | Pch | 30 |
| Zero Check Control — 11 or 12 | P, 23-24 | Pch | 31 |
| Immediate Stop | Q, 23-24 | A | 32 |
| Pickup | L, 23-24 | EC | |
| Punch | M, 23-24 | EC | 33 |
| Stop | O, 23-24 | A | 32 |



* X only on card read; any impulse on program cycle.

*Figure 74. Timing Chart Entries*

To prevent the burning of selector points, the following rule must be observed: *a selector should not change from normal to transferred (or vice versa) during the time that an impulse is passing through the selector.* Because of the time delay inherent in the transfer of selectors, an allowance of at least 11° should be made between the time a selector starts to transfer and the impulse to be selected. Thus, it would not be correct to select an all cycles impulse through the normal side of a selector picked up by a program exit, since both impulses occur at the same time and on the same cycle. The drop-out of a pilot selector should not be impulsed between 355° and 50°, as this damages the selector; therefore, it would not be correct to drop out a selector from a program exit or a program couple.

Whenever a setup impulse (15° to 45°) is selected, the selector should be picked up from program couple or from an impulse of similar timing. The program couple timing is from 355° to 50°, thus overlapping the timing of the setup impulse.

*Punch Hubs.* Although the 80 punching hubs are treated as entries throughout the manual, the timing chart shows them as exits. Conversely, storage punch exits are treated as exits in the manual while they are shown as entries on the timing chart. Other hubs labelled PCH in the cycle column of both timing charts may reveal a similar inconsistency. The labeling on the timing charts is technically correct because the 80 punching hubs are actually on the exit side of the line and all hubs that may be connected to them are on the entry side of the line.

All hubs labeled PCH bear no definite timing relationship to the machine cycle. Punching impulses are available only from the hub corresponding to the card column which is in punching position. Punching impulses to a storage unit start right after 355° of a program or read cycle during which storage control punch is impulsed, provided the card is in position to punch that particular column. Punching impulses to the punch emitter start as soon as the card is in punching position for that particular column. The punching impulse is interrupted as the card moves from one column to the next. Approximately four columns are punched in each machine cycle.

## CONTROL PANEL SUMMARY

EACH SECTION of the control panel is assigned a number under which the hubs are briefly described.

1. *Reset to 5.* Six independent exit hubs that have the same timing as an X impulse. When wired to a counter entry position, they cause that counter position to reset to 5, instead of to zero, when the counter itself resets. This is the method used for half-adjustment.

2. *Co-selectors 1-4.* Five-position selectors controlled normally from either a pilot selector couple exit or a program exit. When picked up they transfer immediately and hold for the remainder of the cycle.

3. *Storage Transfer Exit.* Exits to transfer information from the storage units to counters or from one storage unit to another. Units 5 and 8 are not standard.

4. *Counter Entry, Dividend.* Entry hubs to all counters. The first three counters must be used for the dividend. Counters 7 and 8 are not standard.

5. *Reading.* Exits for the 80 card reading brushes.

6. *Divisor-Multiplier-Storage Entry.* Entry to all the storage units. Each unit has 12 positions; all but unit 1 are divided equally into two parts of six positions each. Unit 1 is divided into a four-position left section and an eight-position right section. The right section (1R) must be used for the divisor or the multiplier. Information to be punched must be entered into storage units 6 or 7. Units 5 and 8 are not standard.

7. *Counter Exit.* Exits for all counters. Counters 7 and 8 are not standard.

8. *Co-selectors 5-8.* Same as co-selectors 1-4, item 2. Co-selectors 9, 10, 11, and 12 are not standard.

9. *Punching.* Eighty punching positions representing the 80 columns of the card.

10. *Storage Punch Exit.* Exits for punching from storage units 6 and 7. For punching from either 6 or 7, the units position must always be wired. Unit 8 is not standard.

11. *Punch Emitter.* Exits for punching digits 0 through 12. These hubs may be wired directly to punching.

12A. *Bal Test.* Four independent units for converting negative balances to true figures. Balance test hubs are entries from the position in the storage units representing the 9 or 0 signal. First column storage hubs are entries from the position in the storage unit representing the first column (high-order position) to be punched. First column storage punch hubs are exits when the other two rows are impulsed. They are wired to the first column to be punched.

12B. *Punch X+ or X—.* Four independent units to emit X punches for identifying negative or positive balances. An X punch is available out of any one of the four hubs in the top row (punch X—), whenever the balance is negative. An X punch is available out of any one of the four hubs in the middle row (punch X+), whenever the balance is positive. X column storage hubs in the bottom row are entries for digits which will be emitted by X+ or X— hubs together with the X.

13. *X.* Six common exit hubs that emit X impulses for punching.

14. *Skip.* Six common exit hubs that cause skipping from any column to which they are wired.

15. *NB Check.* Four common entry hubs, normally wired from any counter NB and used in conjunction with

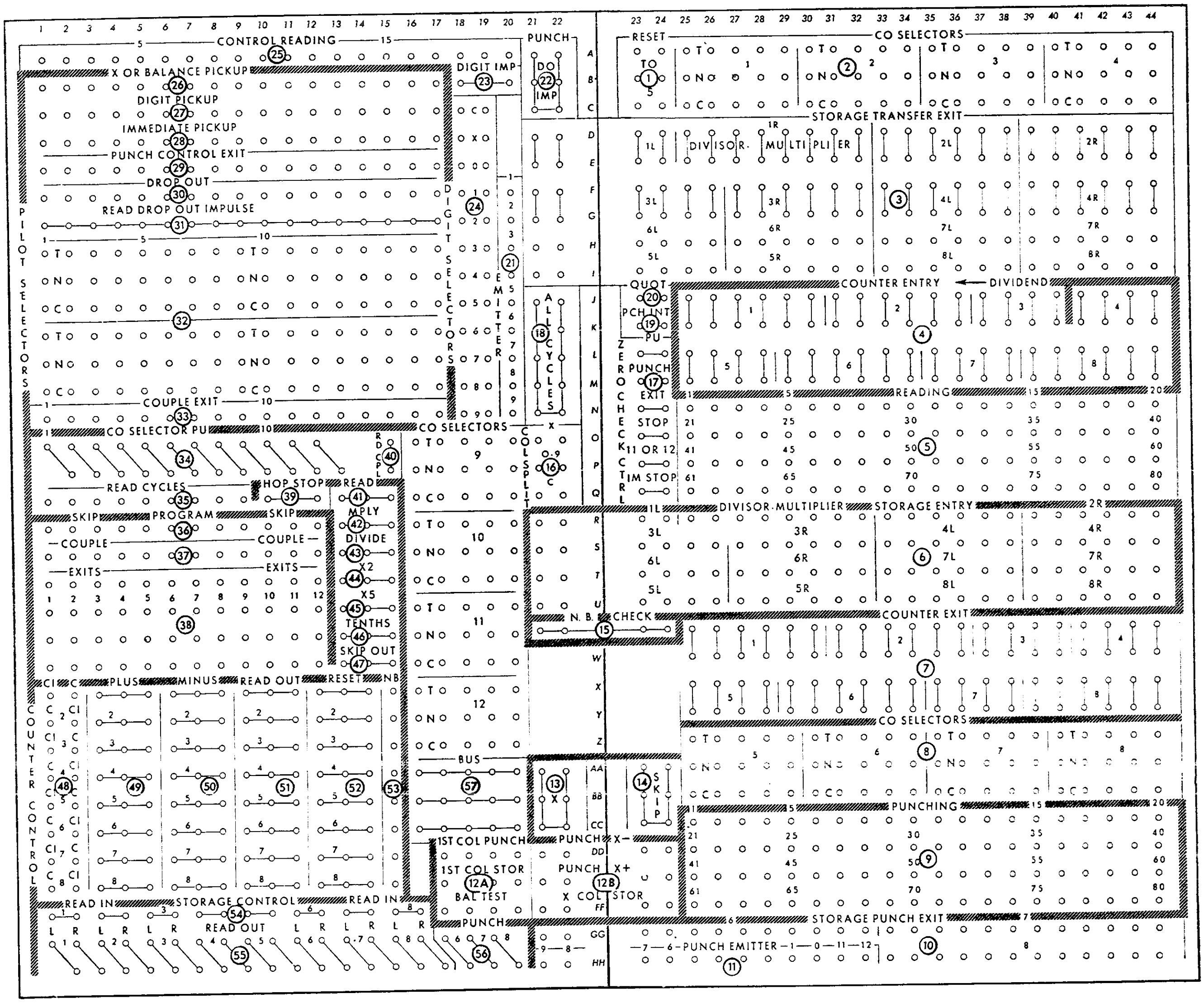


Figure 75. Control Panel Summary

the zero-check control unit for multiplication and division checking.

16. *Column Split.* Two independent units, each consisting of a common hub, a 0-9 hub, and an X hub, to filter either X or digit impulses when both are punched in the same column.

17. *Zero Check Control.* Used in conjunction with NB check, for multiplication and division checking.

*PU.* Two common pickup hubs normally wired from a program exit.

*Punch.* When impulsed, punch sets up an X or a 12 to be punched from the 11 or 12 hubs. An X is punched in error cards and a 12 in correct cards.

*Exit.* Exit hubs normally wired to stop the machine when a card is in error.

*Stop.* Two entry hubs normally wired from EXIT to cause the machine to stop after the error card is ejected into the stacker.

*11 or 12.* Emits an X for incorrect cards and a 12 for correct cards when PUNCH is impulsed.

*Immediate Stop.* Two common entry hubs to cause the machine to stop immediately upon detection of an error if wired from EXIT. Card does not automatically eject into the stacker.

18. *All Cycles.* Ten common exit hubs, of the same duration as program or read cycles, available every machine cycle.

19. *Punch Interlock.* Two common PUNCH INTERLOCK entry hubs, normally wired from the same program on which the punching selectors are picked up. They are used when punch selection is not under the control of control or read brushes to stop the calculation of a card until the card ahead has finished punching.

20. *Quotient.* Quotient exit hubs that emit one quotient digit at a time during a divide operation. They are wired to a counter entry.

*21. Emitter.* Exit hubs labeled 1 through 9 which emit corresponding digits on every machine cycle. They are wired to counters or to storage units.

*22. Punch DO Impulse.* Six common exit hubs that emit impulses, at the conclusion of punching, to the drop-out of pilot selectors.

*23. Digit Impulse.* Three common exit hubs that emit an impulse covering nine through X on every machine cycle. When they are wired to the C of a digit selector, the selector becomes an emitter.

*24. Digit Selectors.* Two digit selectors, each having hubs representing a C, X and all digits from 0 through 9. When C is wired from READING or CONTROL READING, specific punched digits can be selected. When C is wired from DI, specific digits are emitted.

*25. Control Reading.* Twenty control reading hubs representing any 20 card columns. They emit X or digit impulses, and are used to pick up selectors.

*26. X or Balance Pickup.* Pickup hubs for pilot selectors that are wired from CONTROL READING, READING, or any counter NB. A selector impulsed through these hubs transfers one cycle later and latches until impulsed to drop out.

*27. Digit Pickup.* Digit pickup hubs for pilot selectors that may be wired from any control source. When D pickup hubs are impulsed from program couple or program exits, the selectors transfer for the same cycle. When D is impulsed from any other source, the selectors transfer one cycle later. Once transferred, the selectors remain transferred until impulsed to drop out.

*28. Immediate Pickup.* Entry hubs. Impulses (X, digit or NB) introduced into these hubs cause selectors to pick up immediately and hold for the remainder of the cycle.

*29. Punch Control Exit.* Exit hubs (if selector is controlled) that are active just before punch time. They are normally used to pick up another pilot selector for punch selection.

*30. Drop-out.* Entry hubs to drop out pilot selectors from READ DROP-OUT IMPULSE, PUNCH DO IMP, DIGIT IMP, or any digit 9-1 from a counter or storage unit.

*31. Read Drop-out Impulse.* Common hubs that emit impulses at the end of a read cycle. They are normally wired to DROP-OUT ENTRY.

*32. Pilot Selectors.* Seven two-position pilot selectors are standard. Their function depends upon pick up and drop-out explained under sections 26 through 31.

*33. Couple Exit.* Exit hubs of pilot selectors used to pick up co-selectors. These hubs emit impulses during transfer time of pilot selectors if the pilot selector is picked up with X, balance or digit.

*34. Co-selector PU.* Pickups for co-selectors 1-4 (section 2) and 5-8 (section 8).

*35. Read Cycles.* Nine individual exit hubs that emit impulses on the read cycle only.

*36. Program Skip.* Entry hubs for each program step to cause program skipping if wired from program exits, read cycle, NB of a counter, or an X from the reading brushes. An entry into the skip hub of a program step starts programming on that step, skipping all programs preceding it.

*37. Program Couple.* Exits to control co-selectors from program steps, or to couple one program step to another adjacent step for the purpose of expanding program exits from four to any number desired.

*38. Program Exits.* Four independent exits for each program step to control the type of operation being performed. The last program used is always wired to READ (section 41).

*39. Hopper Stop.* Three common entry hubs, that, when impulsed from all cycles, stop the last card between the control brushes and reading brushes.

*40. RD CPL* (Read Couple). Two common exit hubs used to pick up co-selectors during a read cycle.

*41. Read.* Entry hubs to stop further programming. They are wired from a read cycles hub to suspend all programs, or from a program exit to suspend any additional programs.

*42. MPLY (Multiply).* Entry hubs that, when wired from a program exit, cause multiplication. They should not be wired from two successive program steps.

*43. Divide.* Entry hubs, that, when wired from a program exit, cause division. They should not be wired from two successive program steps.

*44. ×2.* Entry that, when impulsed from a program exit, causes multiplication by 2.

*45. ×5.* Entry that, when impulsed from a program exit, causes multiplication by 5.

*46. Tenths.* Entry that, when impulsed from a program exit, shifts the product one position to the right. These hubs are standard only on dividing machines.

*47. Skip Out.* Entry hubs to skip the card to column 80 regardless of where skip bar inserts are placed. These hubs are normally wired from one of the 80 reading brushes to eject master cards.

*48. C-CI.* Carry and carry impulse hubs. CI must be wired to C for all counter operations, because the counters in this machine are of the net balance (complement 9) type. They are also used in counter coupling.

*49. Plus.* Entry for impulses to cause counters to add.

*50. Minus.* Entry for impulses to cause counters to subtract.

*51. Read-out.* Entry for impulses to cause counters to read out.

52. *Reset.* Entry for impulses to cause counters to clear.

53. *NB.* Exit that emits an X impulse at the end of the cycle during which the counter turns negative and continues to emit an impulse for every cycle thereafter until counter is either cleared or turns positive. Testing for plus or minus balances may be done on any program step except those used for multiplication or division.

54. *Storage Control Read-in.* Entry hubs for controlling the read-in of both left and right halves of storage units 1, 2, 3, 4, 6, and 7. Units 5 and 8 are not standard.

55. *Storage Control Read-out.* Entry hubs for controlling read-out from each storage unit independently when transferring.

56. *Punch.* Entry hubs to control punching from storage units 6 and 7. Unit 8 is not standard. Normally wired from read cycle or program exits.

57. *Bus.* Entry for impulses to elminate split wiring.

# Index

## WIRING DIAGRAMS

(1/60: 8M-ME)

International Business Machines Corporation
Data Processing Division, 112 East Post Road, White Plains, N.Y.

Printed in U.S.A. A22-050