IBM

IBM 610

Auto-Point Computer

Manual of Operation

Preliminary Edition

Printed in U. S. A.
Form 23-6335-0

CONTENTS

INTRODUCTION

BASIC ARITHMETIC OPERATIONS

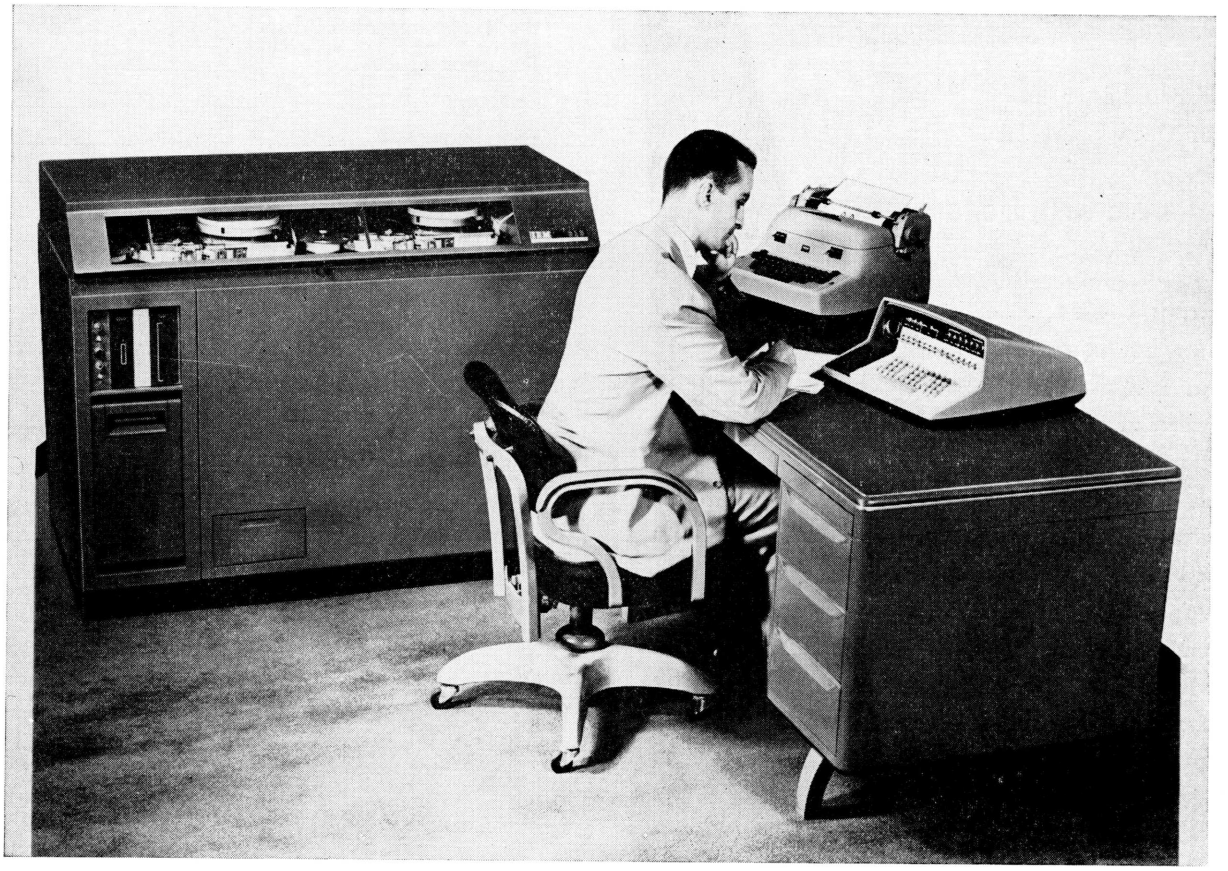
CONTROL DEVICES

OUTPUT DEVICES

OTHER MODES OF OPERATION

SUMMARY OF OPERATION

PROGRAMMING THE 610



IBM 610 Computer

IBM 610 AUTO-POINT COMPUTER

THIS MANUAL is intended as a reference for programmers of the IBM 610. All information about a given topic is assembled in one place with the reference purpose in mind. However, the organization is such that a beginner can gain complete knowledge of the machine through careful study of this manual.

The IBM 610 is a mobile, general-purpose computer specifically developed for a wide range of industrial and engineering organizations, from the small development laboratory where it can function as a fast and automatic primary calculator, to the giant research facility where it will be a useful helper to large-scale computers. It has many of the arithmetic and logical facilities usually associated with modern large-scale automatic computers; yet it is as easy to use as a desk calculator.

The machine consists of three physical units: the keyboard, the typewriter, and the computer itself. The computer unit contains paper-tape punching and reading units and a pluggable control panel, as well as all of the arithmetic and control circuitry.

Numbers up to fifteen digits in length may be entered into the 610 either manually from the keyboard or automatically under computer control from punched paper tape. Data are internally stored in eighty-four registers on the magnetic drum. Each digit is identified by a recognizable pattern of magnetic spots. Any information recorded on the drum will remain there permanently, or until it is erased by recording other information in the same location. The machine may be turned off completely without any danger of loss of data.

Control of the machine may be exercised in three ways:

1. manually from the keyboard,
2. automatically from a punched paper tape program,
3. automatically by control panel wiring.

Transfer between control devices is automatic.

Final or intermediate answers may be obtained on the typewriter, on punched paper tape, or both simultaneously.

SPECIAL FEATURES

SOME outstanding and unusual features have been incorporated into the IBM 610.

Auto-Point

A truly unique feature of the machine is the ability to automatically handle a decimal point. When operating in the auto-point mode, data entered are automatically positioned within storage locations so that the point is centered using one-half of the register for integers and the remainder for fractions. Every computation causes an accurate automatic re-alignment of the point. In comparison to widely used floating-point systems, this feature needs no lengthy programming routines or separate storage locations to remember where the point is. The machine carries an actual decimal-point notation, and within machine capacities, the programmer is completely freed from scaling input numbers and remembering decimal position during computation.

Program Preparation

While a problem is being solved manually, the computer is able to create a program tape *at the same time*. A subsequent problem of the same type will require only the input of new data; control of the machine can then be transferred to the program tape for completion of the calculations and the printing of answers.

Decimal-Octal

It is possible to perform either decimal or octal arithmetic. This means that test programs for large-scale binary computers can be *developed* in decimal form and then *executed* in octal arithmetic for direct comparison to the binary machine results.

While operating in the octal mode, the machine will perform all the functions as in the decimal mode including auto-point octal.

Multiple Operation

The machine provides multiple function instructions for programming simplicity. The Square Root instruction accomplishes in one operation what is normally a many instruction subroutine. Divide-Multiply combines into one operation the division of two factors and multiplication of the quotient by a third.

Checking

The IBM 610 has been designed with reliability as a primary consideration. Conservative circuit design and components chosen for their high reliability are employed throughout. This computer also uses built-in self-checking features, which provide further assurance of accurate results.

ADDRESS SYSTEM

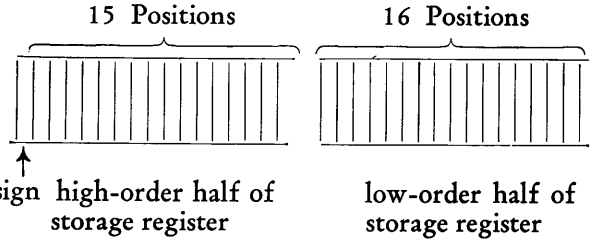
THE machine is concerned with two basic types of communication — problem data and control information. Problem data are entered into the machine and stored as magnetic spots on the surface of the drum. In order for these data to be used they must be stored and retrieved from the same physical location. A system of numerical addresses plus checking and timing circuitry allows the operator to locate data for input, output and computations. The idea is analogous to an automatic parking lot in which cars are mechanically stored in numbered slots and retrieved again from the same slot.

Register

Within the machine, data are stored on the drum in 84 addressable locations called registers. Each of the registers has an address: These addresses are 00, 01 . . . .79, plus special registers A, MP, MC, DIV (to be explained later). This permits retrieval and storage of information during calculation. Each of these locations has a capacity of up to 31 digits plus algebraic sign and radix point.

Numbers up to fifteen digits in length, plus point and sign, may be entered into any storage register. The largest magnitude that can be entered into a register is 9999 99999 99999.9 in auto-point. (Note that these two statements are not equivalent. The number 10000 00000 00000. is a fifteen-digit number, but it may not be entered into the machine because it is greater than the maximum allowable magnitude.)

The register capacity of thirty-one digits allows the machine to position every number with its point in the *middle* of the register for operation in the auto-point mode. A storage register may be visualized as follows:



The first position to the left contains the sign of the number. If this position contains a *0*, the sign of the number is positive; if this position contains a *9*, the sign of the number is negative.

The storage register may be thought of as divided into two parts as indicated by the space in the middle of the register. When the machine is operating in auto-point, the integer portion of a number is in the left half or high-order half of the register. The fractional portion of the number is contained in the right half or the low order of the register. The decimal point is stored in combination with the first digit in the lower half of the register. The number stored in this manner is said to be in *right-hand standard* position. During operation in the auto-point mode, it is permissible to move a number out of the right-hand standard position. However, before any arithmetic operations can be correctly performed in the auto-point mode, the number must be returned to the right-hand standard position.

A number is said to be in left-hand standard position if the first significant (non-zero) digit has been moved to the first position to the right of the sign position, or if the decimal point has been moved to the first position to the right of the sign position.

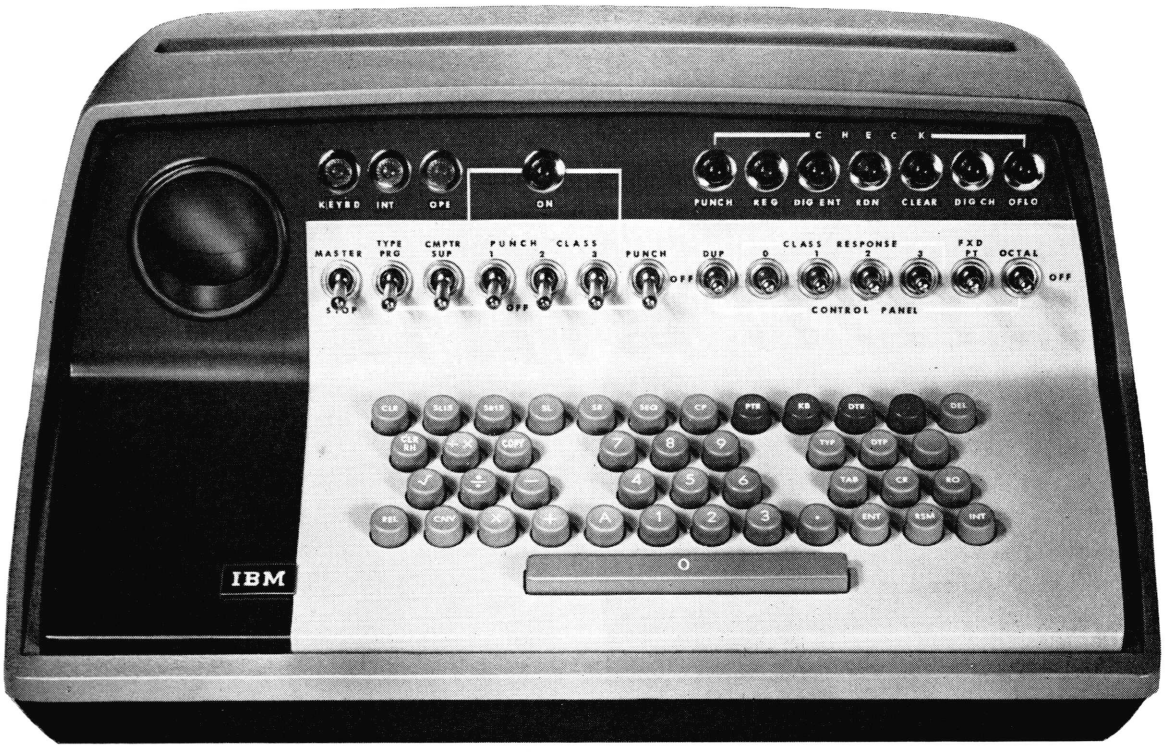


FIGURE 1. KEYBOARD

Word

The basic unit of problem data is called a word, which is 15 digits in length plus algebraic sign and decimal point. The words may enter the machine via paper tape or the keyboard, and once stored on the drum they can be addressed for computations. The 31-digit register size allows automatic double-precision arithmetic. For most purposes, word size need not concern the operator.

Selected Register

There are normally two factors necessary for the execution of any arithmetic instruction. The 610 gives special attention to the address of the first or prime factor by designating it as the *selected register.* All 610 registers are capable of performing arithmetic and logical operations, and therefore are recipients of results. The result for an operation is placed in the *selected register.* This emphasis on the prime factor gives a programming advantage, because a register remains *selected* until a programming change is specified. Obviously, a following instruction could be written supplying only an operation and a second factor, the machine using the selected register contents as the first factor. The conventions used to express this and the few exceptions will be explained in a later section.

When a problem is processed in a one-time calculation, instructions are given to the machine by depressing the keys on the keyboard. Problem solution is therefore accomplished in a manner similar to a desk calculator. (See Figure 1.)

This manual first explains machine performance in terms of manual control and subsequently progresses to automatic control.

Because of the advantage of operation in auto-point mode, all problems will be in the auto-point mode unless otherwise stated.

ENTRY OF DATA

LET US consider the entry of data, remembering that depression of a key activates the machine. We first select one of the registers by keying in its address on the keyboard (00-79). For this operation ENT is depressed, followed by the required digits, decimal point, and finally the algebraic sign. The auto-point

feature inserts zeros in unused digit positions when the number is automatically positioned, requiring the programmer to enter only significant digits.

The depression of the ENT key on the keyboard clears the selected register to zeros and enters the digits next specified into the selected register. The actual data may contain from one to fifteen digits, plus point and sign. Depression of the + or − key gives the numerical data its appropriate sign, causes the auto-point feature to position the data in the register, and completes entry.

If a program is being prepared to process several sets of data under automatic control, provision must be made for the entry of each set of data. To accomplish this, machine calculation is stopped long enough to enter the data manually or to allow instructions for data to be read from the tape.

EXAMPLE: The instruction 07 ENT 62.731+ causes the register 07 to be cleared and the number 62.731+ to be placed in the register. Register 07 remains selected. The process takes place as follows:

1. The 0 key is depressed.
2. The 7 key is depressed.
3. The ENT key is depressed, resulting in the register 07 being cleared to all zeros.

0000000000000000 .0000000000000000

4. The 6 key is depressed giving

0600000000000000 .0000000000000000

5. The 2 key is depressed giving

0620000000000000 .0000000000000000

6. The . key is depressed giving

062.0000000000000 0000000000000000

7. The 7 key is depressed giving

062.7000000000000 0000000000000000

8. The 3 key is depressed giving

062.7300000000000 0000000000000000

9. The 1 key is depressed giving

062.7310000000000 0000000000000000

10. The + key is depressed to give the number a sign causing a 0 to be placed in the sign position or highest-order position of the high-order half of the register and the number is moved to the center of the register giving

0000000000000062 .7310000000000000

If a number + .0076 is entered into the machine, ., 0, 0, 7, 6, and + keys are depressed, and the number appears in the register as:

0000000000000000 .0076000000000000

In the example used with the ENT instruction, the number entered in this manner would be used in a one-time calculation from the keyboard, or as a problem constant. If a program is being prepared to process several sets of data under automatic control, provision must be made for the entry of each set of data. In order to accomplish this, the machine calculation is stopped long enough to enter the data manually from the keyboard or instructions are inserted calling for data from the tape.

At the moment, attention will be directed to one-time, keyboard calculations. Later discussion will be concerned with multiple sets of data.

BASIC ARITHMETIC OPERATIONS

THIS SECTION deals with the basic arithmetic operations in 610 language along with an explanation of their functions.

NOTE: *The IBM 610 automatically obeys all of the usual sign rules of elementary algebra.*

ADDITION

THE FIRST occasion for the depression of this key is giving a sign to a positive number being entered into the 610. Depression of this key terminates the entry of the number and places a 0 in the left-most position of the register (a machine symbol for a + sign), and automatically positions the number in the register for calculation.

During the calculation part of a problem, the depression of this key tells the machine to add the contents of the next addressed register to the register already selected, and to replace the contents of the already selected register by the sum. The contents of the addressed register remain unchanged.

EXAMPLE 1:

Register 01 plus Register 02 (01 selected, 02 addressed)

	Register	
Before	01+000000000000001	.7300000000000000
	02+000000000000256	.0033000000000000
After	01+000000000000257	.7333000000000000
	02+000000000000256	.0033000000000000

Following the addition register 01 is selected.

EXAMPLE 2:

05+17

	Register	
Before	05+000000000000002	.7600000000000000
	17−000000000000003	.0700000000000000
After	05−000000000000000	.3100000000000000
	17−000000000000003	.0700000000000000

Following the addition register 05 is selected.

SUBTRACTION

THE FIRST occasion for the depression of this key is in giving a sign to a negative number being entered. Depression of this key terminates the entry of the number and places a 9 in the left-most position of the register (the machine symbol for a − sign), and automatically positions the number in the register for calculation.

During the calculation part of a problem, the depression of this key tells the machine to subtract the contents of the next addressed register from the selected register, and to replace the contents of the selected register by the difference. The contents of the addressed register remain unchanged.

In general, the operator need not concern himself with the actual representation of the number within the machine. The operator will find it best to think of the number as being stored with a minus sign and proceed as he would if he were writing the problem on a work sheet for hand calculation. When a negative number is read out of the machine to the typewriter, it appears as a true number with a minus sign.

EXAMPLE 1:

Register 06 Minus Register 73

	Register	
Before	06+000000000000005	.7600000000000000
	73+000000000000003	.1500000000000000
After	06+000000000000002	.6100000000000000
	73+000000000000003	.1500000000000000

Following the subtraction register 06 is selected.

EXAMPLE 2:

07 − 14

	Register	
Before	07+000000000000013	.6500000000000000
	14+000000000000017	.7500000000000000
After	07−000000000000004	.1000000000000000
	14+000000000000017	.7500000000000000

07 Selected.

CHANGING THE SIGN

IT IS sometimes necessary, during a calculation, to change the sign of a number from plus to minus or vice-versa. On most computers this necessitates finding a register containing all zeros and subtracting the number from the zero register.

However, in the IBM 610 it is necessary only to give a single command—convert.

The depression of this key tells the machine to replace the contents of the selected register by the negative of the original contents. The register remains selected.

EXAMPLE 1:

```
          06 CNV
        Register
Before  06+000000000000013 .6500000000000000
After   06-000000000000013 .6500000000000000
06 Remains Selected
          17 CNV
```

EXAMPLE 2:

```
Before  17-000000000000003 .0700000000000000
After   17+000000000000003 .0700000000000000
17 Remains Selected
```

CLEARING

Transfer of Data within the Machine

In order to move a number from a storage register to another location in storage we use COPY.

The depression of this key tells the machine to erase the contents of the selected register, to replace it by all zeros with a plus sign, and then add the contents of the next addressed register to the zeros of the selected register. The contents of the addressed register remain unchanged. The selected register remains selected.

One important use of this instruction is for the storage of intermediate results during calculation.

EXAMPLE:

```
          01 CPY 05
        Register
Before  01+000000000000001 .7300000000000000
        05+000000000000002 .6700000000000000
After   01+000000000000002 .6700000000000000
        05+000000000000002 .6700000000000000
01 Selected
```

Setting a Register to Zero

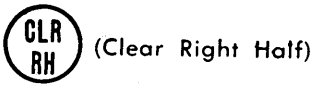
In order to clear a storage location previous to an accumulation we use CLEAR.

The depression of this key tells the machine to erase the contents of the selected register and replace the original contents with all zeros and a plus sign.

EXAMPLE:

```
          22 CLR
        Register
Before  22+000021563185733 .4210000000000000
After   22+000000000000000 .0000000000000000
22 Selected
```

Clear Right Half



The depression of this key causes the last sixteen positions of the selected register to be replaced by zeros. The machine takes no cognizance of decimal position or the contents of the right half of the register. The selected register remains selected. The decimal point is *not* erased.

EXAMPLE:

```
          29 CLR RH
        Register
Before  29+000000000001765 .4180021004000000
After   29+000000000001765 .0000000000000000
29 Selected
```

SHIFTING

IF the operator wishes to shift the contents of a register to the left or to the right, he has four operations available to help him accomplish this. As a result of shifting, the sign position never changes, but numbers shifted beyond register limits are lost.

One-Place Shifts

(Shift Right One Place)

The depression of this key tells the machine to shift the contents of the selected register to the right one place. The point moves with the digits, and the register shifted remains selected. This operation may be given several times in succession. For instance, SR SR SR will cause the contents of the selected register to be moved to the right three places. In using this instruction on auto-point numbers, care must be taken that the number is returned to the right-hand standard (point in the middle) before other operations are carried out on it; otherwise, errors will result.

EXAMPLE:

```
          24 SR
       Register
Before  24+000621053162181 .2560000000000000
After   24+000062105316218 1.256000000000000
24 Selected
```

When a positive number is shifted to the right in a register, the vacated high-order positions of the upper half of the register are automatically filled with zeros.

When a negative number is shifted to the right in a register, the vacated high-order positions of the upper half of the register are automatically filled with nines.

(Shift Left One Place)

The depression of this key tells the machine to shift the contents of the selected register to the left one place. The point moves with the digits, and the register remains selected. This operation may be given several times in succession. Care must be taken that auto-point numbers shifted away from the right-hand standard position with this instruction are returned before the number is used again in an arithmetic operation.

EXAMPLE:

```
          25 SL
       Register
Before  25+000000067142187 .2180000000000000
After   25+00000067142187.2 1800000000000000
25 Selected
```

When a number is shifted to the left in a register, the vacated low-order positions of the lower half of the register are automatically filled with zeros.

Multiple-Place Shifts

There are two additional shifting operations. The operations they cause will be according to whether the 610 is in the auto-point or the fixed point mode:

(Shift Left 15, Auto-Point)

The depression of this key tells the machine to shift the contents of the selected register to the left until one of the following conditions is satisfied:

1. The most significant (non-zero) digit is next to the sign position.
2. The decimal point is next to the sign position.
3. A maximum shift of 15 places has been made.

A shift of exactly fifteen places is *not* implied by this instruction when the 610 is in the auto-point mode.

EXAMPLE 1:

```
AUTO-POINT
          26 SL 15
       Register
Before  26+000010620151843  .2146000000000000
After   26+10620151843.2146  0000000000000000
26 Selected
```

EXAMPLE 2:

```
AUTO-POINT
          30 SL 15
       Register
Before  30+000000000000000 .0076511821750000
After   30+.007651182175000  0000000000000000
30 Selected
```

REMINDER: *In the auto-point mode, if a number is positioned in a register in such a way that no zeros appear to the left of the high-order significant digit, OR if a number is positioned with the decimal point in the high-order position of the register, the number is said to be in the left-hand standard position.*

The operation of the machine as a result of the depression of the SL 15 and SR 15 keys is different in the case of the fixed-point mode.

SL15 (Shift Left 15, Fixed Point)

The depression of this key tells the machine to shift the contents of the selected register to the left fifteen places.

EXAMPLE 1:

```
FIXED-POINT
          26 SL 15
        Register
  Before  26+000000000000000 07.62051831210000
  After   26+07.6205183121000 0000000000000000
  26 Selected
```

EXAMPLE 2:

```
FIXED-POINT
          32 SL 15
        Register
  Before  32+0017700051.02015 7600000000000000
  After   32+760000000000000 0000000000000000
  32 Selected
```

NOTE: If during calculation of a problem, transfer from fixed-point to auto-point mode is made, and no decimal point appears in a register to be shifted, the machine shifts exactly 15 places.

SR15 (Shift Right 15, Auto-Point)

The depression of this key tells the machine to shift the contents of the selected register to the right until the decimal point is in the left-most digit position of the lower half of the register (right-hand standard position). If the decimal point is already in that position or to the right of that position, the operation has no effect on the contents of the register. This instruction does *not* imply a shift of exactly fifteen places in the auto-point mode.

EXAMPLE 1:

```
AUTO-POINT
          28 SR 15
        Register
  Before  28+000010.775251847 2146000000000000
  After   28+000000000000010 .7752518472146000
  28 Selected
```

EXAMPLE 2:

```
AUTO-POINT
          29 SR 15
        Register
  Before  29+000050607088123 .7717000000000000
  After   29+000050607088123 .7717000000000000
  29 Selected
```

REMINDER: *In the auto-point mode, when a number is positioned so that the decimal point is in the high-order position of the lower half of the register, the number is said to be in right-hand standard position.*

SR15 (Shift Right 15, Fixed-Point)

The depression of this key tells the machine to shift the contents of the selected register to the right fifteen places.

EXAMPLE:

```
FIXED-POINT
          30 SR 15
        Register
  Before  30+000010620151772 4186000000000000
  After   30+000000000000000 0000106201517724
  30 Selected
```

Rounding

Because of the large capacity of the 610 registers (31 digits) and the auto-point mode of operation, it will seldom be necessary to round a number. Should it become desirable, however, it can be done as follows:

1. Enter a constant with the 5 in the correct position (one position to the right of that desired correct in the results) into a vacant register.
2. Add the register containing the 5 to the register to be rounded.
3. Shift the number to the left so that the portion of the number to be eliminated is in the lower half of the register and the portion of the number to be retained is in the upper half of the register.
4. Clear the right half of the register.
5. Shift Right (SR 15) to return the number to the right-hand standard position for further computations.

EXAMPLE:

Suppose register 28 contains

+000000076512136 .7718154000000000

and we wish to round to 3 decimal places.

We proceed as follows:

1. Select an unused register, e. g., 37, and give the following instruction:

37 ENT .0005+

The operator's programming of the problem will tell the machine whether the number he wishes to round is positive or negative. If the number is positive, the constant will be +.0005; if negative, −.0005. The rounding 5 may be entered from the program tape, data tape, or manually from the keyboard.

2. The next program step is:

28 + 37

The result in register 28 is:

+000000076512136 .7723154000000000

3. We now give the next steps:

SL SL SL

And the result in register 28 is:

+000076512136.772 3154000000000000

4. The next instruction step is:

CLR RH

+000076512136.772 0000000000000000

5. If we wish to use this number in further processing, we give the instruction:

SR 15

and the result in register 28 is:

+000000076512136 .7720000000000000

It is entirely possible that a program may develop and call for retention of a larger number. If the number in register 31 to be rounded is:

+017631542187721 .7727121000000000

If we wish to round and retain four decimal places, we would add .00005+. The result of adding in register 31 would be:

+017631542187721 .7727621000000000

However, we cannot perform the usual shifts to the left because the instructions

SL SL SL SL

would give in register 31:

+31542187721.7727 6210000000000000

and we have lost the high-order digits of our answer.

Instead of the left shifts and CLR RH, we must find an alternate procedure. The alternate procedure calls for a series of right shifts. In this case the instructions are:

SR SR SR SR SR SR SR SR SR SR SR SR

Register 31 will now contain:

+000000000000017 6315421877 21.7727

To replace the decimal point in the regular auto-point position, we must give the instructions:

SL 15

with the result

+17631542187721.7 7270000000000000

Then

SR 15

with the following final result:

+017631542187721 .7727000000000000

SPECIAL REGISTERS

REGISTER selection can be changed only as the result of intentionally selecting a new register at the completion of an instruction, or after the operations of multiplication, division, or square root when the attention of the machine is focused on the special A (Answer) register containing the answer or result of the calculation just performed.

The 80 registers (00 . . . . 79) used with the operations explained above required no additional register during the calculation itself, because the operations were actually performed within the selected register. When $\times$, $\div$, $\div\times$ or $\sqrt{\ }$ are performed, extra intermediate storage is required; and to supply this need, the machine has 4 special registers A (Answer), MP (Multiplier), DIV (Divisor) and MC (Multiplicand).

The special registers are 31 digits in length plus algebraic sign and decimal point.

Though normally used for these functions, these registers are also addressable bringing to a total 84 addressable registers in this machine.

These registers are addressed as follows:

MP	0.
DIV	1.
A	2.
MC	3.

Note that the decimal point replaces the units digit in the address.

When addressing these four special registers, it must be remembered that their contents are destroyed during the process of $\times$, $\div$, $\div\times$ and $\sqrt{\ }$. Therefore, these registers may not be used for storing data when these arithmetic operations are to be performed. However, for all other operations, such as +, −, COPY, etc., the contents of these registers are not destroyed, and will therefore have the properties of the other 80 registers.

The depression of this key allows the operator to address the A (Answer) register.

Upon completion of the operation of $\times$, $\div$, $\div\times$, and $\sqrt{\ }$ the result appears in the A register and the A register is selected. For this reason the A register is addressed more often than the other special registers.

Hence the special A key is provided to eliminate the need for the use of the (2.) address.

MULTIPLICATION

THE depression of this key tells the machine to multiply the contents of the selected register by the contents of the next addressed register. The product appears in the A register. In the process, the contents of the selected register are placed in the MP register, and the contents of the next addressed register are placed in the MC register. After the movement of the contents of both registers is completed, the multiplication takes place. The contents of both the original registers remain unchanged. At the end of the operation, the A register is selected.

EXAMPLE 1:

Register 08 × Register 18

	Register		
Before	08	+000000000000007	.0000000000000000
	18	+000000000000012	.0000000000000000
	A	xxxxxxxxxxxxxxx	.xxxxxxxxxxxxxxxx
After	08	+000000000000007	.0000000000000000
	18	+000000000000012	.0000000000000000
	A	+000000000000084	.0000000000000000

A Register Selected

EXAMPLE 2:

Register 08 × Register A

	Register		
Before	08	+000000000000007	.0000000000000000
	A	+000000000000003	.0700000000000000
After	08	+000000000000007	.0000000000000000
	A	+000000000000021	.4900000000000000

A Register Selected

In the auto-point mode, the machine will perform a multiplication, provided the product is *99999999999999.9* or smaller. Exceeding this limit causes the machine to stop, and the keyboard OVERFLOW light comes on.

DIVISION

THE depression of this key tells the machine to divide the contents of the selected register by the contents of the next addressed register. The quotient will appear in the A register. In the process the contents of the selected register are placed in the MP register, and the contents of the next addressed register are placed in the DIV register. After both registers are filled, division takes place. The contents of both original registers remain unchanged. The A register stands selected.

EXAMPLE 1:

Register 01 ÷ Register 08

	Register		
Before	01	+000000000000105	.0000000000000000
	08	+000000000000007	.0000000000000000
	A	xxxxxxxxxxxxxxx	.xxxxxxxxxxxxxxxx
After	01	+000000000000105	.0000000000000000
	08	+000000000000007	.0000000000000000
	A	+000000000000015	.0000000000000000

A Register Selected

EXAMPLE 2:

19 ÷ 18

	Register		
Before	19	−000000000000132	.0000000000000000
	18	+000000000000012	.0000000000000000
	A	xxxxxxxxxxxxxxx	.xxxxxxxxxxxxxxxx
After	19	−000000000000132	.0000000000000000
	18	+000000000000012	.0000000000000000
	A	−000000000000011	.0000000000000000

A Register Selected

In the auto-point mode, the machine will perform a division, provided the quotient is *99999999999999.9* or smaller. Exceeding this limit causes the machine to stop, and the keyboard OVERFLOW light comes on.

When an operator is performing a division of a small number by a relatively large number, the result will be an even smaller number. It is entirely normal that the quotient may be so small that it will be beyond the range of the machine. In the event this happens, the contents of the A register will be all zeros.

The operator should be careful not to interpret this as a failure by the machine to perform the operation. Complete knowledge of the problem and the range of accuracy desired must constantly be kept in mind.

NOTE: In auto-point, if division by zero is attempted, the machine will stop and the keyboard overflow light will come on.

SIMULTANEOUS DIVIDE AND MULTIPLY

(Divide Multiply)

THE depression of this key tells the machine to divide the contents of the selected register by the contents of the next addressed register, and to multiply the quotient by the second addressed register. The result will appear in the A register. All of the numbers are moved to special registers before the operation begins; therefore, the contents of the three registers addressed remain unchanged. In the process of division, the dividend (any of the eighty registers) register has its contents transferred to the MP register. The register containing the divisor has its contents transferred to the DIV register. The contents of the two addressed registers are thus undisturbed as division proceeds. The A register is cleared just prior to the placement of the quotient in the A register. The A register stands selected at the end of the operation.

EXAMPLE 1:

10 ÷× 08 18

	Register	
Before	10+000000000000105	.0000000000000000
	08+000000000000007	.0000000000000000
	18+000000000000012	.0000000000000000
	A xxxxxxxxxxxxxxx	.xxxxxxxxxxxxxxx
After	10+000000000000105	.0000000000000000
	08+000000000000007	.0000000000000000
	18+000000000000012	.0000000000000000
	A+000000000000180	.0000000000000000

EXAMPLE 2:

19÷× A 17

	Register	
Before	19 – 000000000000132	.0000000000000000
	A+000000000000012	.0000000000000000
	17+000000000000003	.0700000000000000
After	19 – 000000000000132	.0000000000000000
	A – 000000000000033	.7700000000000000
	17+000000000000003	.0700000000000000

In the auto-point mode, the machine will perform a divide-multiply, provided the result is 9999999999999.9 or smaller. Exceeding this limit causes the machine to stop, and the keyboard OVERFLOW light comes on.

SQUARE ROOT

(Square Root)

THE depression of this key tells the machine to take the square root of the contents of the selected register. The result will appear in the A register. The number in the selected register remains unchanged. The machine will place the point correctly in the answer in A when the machine is operating in the auto-point mode, and the A register will be selected.

EXAMPLE 1:

20 $\sqrt{}$

	Register	
Before	20+000000000000025	.0000000000000000
	A xxxxxxxxxxxxxxx	.xxxxxxxxxxxxxxx
After	20+000000000000025	.0000000000000000
	A+000000000000005	.0000000000000000

EXAMPLE 2:

21 $\sqrt{}$

	Register	
Before	21+000000000000003	.0000000000000000
	A xxxxxxxxxxxxxxx	.xxxxxxxxxxxxxxx
After	21+000000000000003	.0000000000000000
	A+000000000000001	.7320508075688700

If the operator directs the machine to take the square root of a negative number, the machine will treat the number as if it were a positive number and will give a positive answer.

Programming Example

As an example of the mechanics of using the 610 instructions described up to now, suppose we consider a simple problem in which we want to find the values of x that satisfy the equation $2x^2 - 11x + 12 = 0$. We recall that this equation may be written

Instr. No.	Cl.	First Reg.	Oper.	Second Reg.	Reg. Sel. After	REMARKS	
		01	ENT		01	2.+	Entry Steps
		02	ENT		02	11.−	
		03	ENT		03	12.+	
		20	ENT		20	4.+(constant)	

FIGURE 2. PROGRAM STEPS FOR ENTERING DATA

in the general form $ax^2 + bx + c = 0$. Practically all algebra textbooks contain the development of a formula to solve this equation:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

This is the formula we shall use to solve this simple example. To begin, we must enter the data into the machine. Suppose we plan to place the coefficient of x^2 ($a = 2$) in the magnetic-drum storage location 01. This would be equivalent to writing an $a = 2$ at the top of a work sheet when we are doing a hand solution. In order to accomplish this, we depress the '0' and '1' keys on the keyboard to direct the attention of the 610 to register 01. We next depress the ENT key, clearing the 01 register and preparing it to receive data. We are now ready to enter the desired value into register 01. We now depress the 2, . and + keys, and the value of the coefficient of x^2 is entered into the storage register. The machine was actually ready to receive up to fifteen digits, decimal point, and sign, but any lesser number of digits may be inserted. The depression of the sign key terminates the entry (Figure 2).

In summary, our key depressions were:

01 ENT 2. +

In the same manner we enter $b = -11$ in storage register 02:

02 ENT 11. −

And, finally, we enter $c = 12$ into storage register 03:

03 ENT 12. +

We now have all of our problem data in the machine, and we are ready to begin our calculations.

Let us develop the discriminant ($b^2 - 4ac$) first. We will develop 4ac, then b^2 and take the difference. As we begin to develop 4ac, we discover that we should like to have a constant 4 to use as a multiplier. We select a register, say 20, and enter a four into the machine:

20 ENT 4. +

Now we develop the product 4a (Figure 3):

20 × 01

		20	X	01	A	Develop product 4a in A register	Arithmetic Development
			X	03	A	Develop product 4ac in A register	
		11	COPY	A	11	Store 4ac in register 11	
		02	X	02	A	Develop b^2 in A register	
			−	11	A	Develop b^2-4ac in A register	
			√		A	Develop $\sqrt{b^2-4ac}$ in A register	
		02	CNV		02	Change b to −b in register 02	
		09	COPY	02	09	Store −b in register 09	

FIGURE 3. ARITHMETIC DEVELOPMENT

The product is in the A register, and the A register is selected. We multiply by c in register 03, recalling that the product is in A, with A selected:

$\times$ 03

We can combine these two steps on one line

20 $\times$ 01 $\times$ 03

and have the answer (4ac) in A, with A selected.

We wish to store 4ac for later use in a storage register, say 11:

11 COPY A

We next wish to develop b^2. We recall that b stands in storage register 02. Hence:

02 $\times$ 02

gives b^2 in the A register, with A selected.

Because we want $b^2 - 4ac$, we subtract 4ac (in 11) from register A:

$-$ 11

The difference stands in the A register, with A selected.

We now wish to take the square root of the difference. We notice that if we take the square root of the difference by simply depressing the $\sqrt{\ }$ key, we would destroy the difference ($b^2 - 4ac$) and replace it by $\sqrt{b^2 - 4ac}$. *Care must be exercised not to destroy any intermediate results that may be needed later.* In this case, we do not need to retain the difference, $b^2 - 4ac$; so we depress the $\sqrt{\ }$ key and get $\sqrt{b^2 - 4ac}$ in the A register, with the A register selected.

Because we have completed usage of the constant (+b) for this problem, it would be convenient to change it to $-b$. This is accomplished by:

02 CNV

At this point, we notice that both values of x are developed in almost the same manner, the exception being the sign of the radical. Let us now store $-b$ in one additional register for use in developing the numerators of the two fractions. We do this by selecting a register (09) and putting $-b$ in it:

09 COPY 02

09 now stands selected, and $\sqrt{b^2 - 4ac}$ is in A:

The key depressions:

+A

gives $-b + \sqrt{b^2 - 4ac}$ in 09.

To develop $-b - \sqrt{b^2 - 4ac}$ in 02, we instruct the machine:

02 $-$ A

We are now ready to divide the two developed numerators by 2a in order to obtain the two values of x. However, we note that we have only the value of an a in location 01. In order to obtain 2a, we might insert a constant 2 into an arbitrary storage register by an ENT instruction and multiply. However, it is easier to merely say:

01 + 01

This would give 2a in 01; the value of the a would be destroyed in the process.

Now we are ready to divide numerator by denominator to obtain our values of x. For the first value:

02 $\div$ 01

The quotient stands in A, with A selected. Let us store the first value of x momentarily:

15 COPY A

Now we are ready to obtain the second value of x:

09 $\div$ 01

The quotient stands in A, with A selected (Figure 4).

			+	A	09	Develop $-b + \sqrt{b^2 - 4ac}$ in register 09	
		02	−	A	02	Develop $-b - \sqrt{b^2 - 4ac}$ in register 02	
		01	+	01	01	Develop 2a in register 01	
		02	÷	01	A	Develop $\frac{-b - \sqrt{b^2 - 4ac}}{2a}$ in A register	Answer Development
		15	COPY	A	15	Store $\frac{-b - \sqrt{b^2 - 4ac}}{2a}$ in register 15	
		09	÷	01	A	Develop $\frac{-b + \sqrt{b^2 - 4ac}}{2a}$ in A register	

FIGURE 4. ANSWER DEVELOPMENT

The entire program is shown in Figure 5.

IBM 610 PROGRAMMING SHEET

Storage Usage

00	01 a	02 b	03 c	04	05	06	07	08	09
10	11	12	13	14	15	16	17	18	19
20 4	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39

Instr. No.	Cl.	First Reg.	Oper.	Second Reg.	Reg. Sel. After	REMARKS
		01	ENT			2.+
		02	ENT			11.−
		03	ENT			12.+
		20	ENT			4.+
		20	X	01	A	4a IN A
			X	03	A	4ac IN A
		11	COPY	A	11	4ac INTO 11
		02	X	02	A	b^2 IN A
			−	11	A	b^2-4ac IN A
			$\sqrt{\ }$		A	$\sqrt{b^2-4ac}$ IN A
		02	CNV		02	$-b$ IN 02
		09	COPY	02	09	$-b$ INTO 09
			+	A	09	$-b+\sqrt{b^2-4ac}$ IN 09
		02	−	A	02	$-b-\sqrt{b^2-4ac}$ IN 02
		01	+	01	01	2a IN 01
		02	÷	01	A	$\frac{-b-\sqrt{b^2-4ac}}{2a}$ IN A
		15	COPY	A	15	" INTO 15
		09	÷	01	A	$\frac{-b+\sqrt{b^2-4ac}}{2a}$ IN A

FIGURE 5. PROGRAMMING FOR SOLUTION OF QUADRATIC EQUATION

CONTROL DEVICES

CONTROL of the computing system is exercised by the following four devices:

Manual keyboard
Program tape
Data tape
Control panel

They may serve as input devices for numerical information and may also transmit instructions causing the computer to execute any of the functions of which it is capable. (The data tape reader is normally used for the entry of numerical data, but on occasion it may be used in functional control.) See block diagram in Figure 6.

No more than one control unit may be in operation at one time; however, the unit in control can shift automatically from itself to any other control unit as the needs of the problem in process dictate.

MANUAL KEYBOARD

THE keyboard is always in control of the machine until a command is issued via the keyboard to cause automatic control by one of the other units. Thereafter, automatic transfer from one control to another may be made.

Many of the functions of the manual keyboard have already been explained. The remainder of keyboard functions will be covered later in the manual in context with their application.

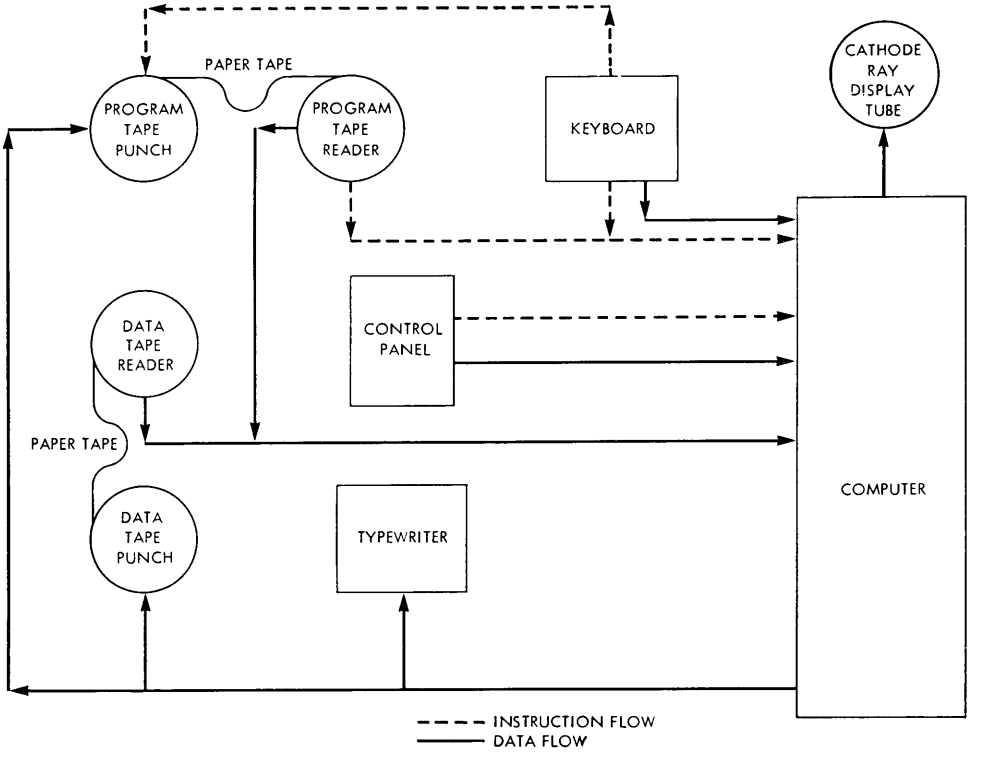


FIGURE 6. INTERNAL FLOW SCHEMATIC

PROGRAM TAPE

THERE are many occasions when it is desirable to process several sets of data for the same problem in some automatic manner. One method of doing this is by using the Program Tape Reader.

While the operator is working a problem for the first time, he may set switches on the keyboard that cause the machine to punch codes into a paper tape equivalent to the keys being depressed. Later he may, by changing switch settings and depressing appropriate keys, turn over control of the machine to this program tape. The only activity required of the operator when the program tape is being used will be the entering of new sets of data into the machine during the processing of each problem.

In order to get the program into the tape during the processing of the first set of data, the operator simply sets the PUNCH switch on the keyboard to the ON position. A full discussion of the switches and lights on the keyboard will be found in a later section.

The operator now proceeds in a manner similar to the one-time calculation. He must always be cognizant of the source of the data. The only difference will be that the program tape punch is active. At the end of the processing of the first calculation, the program is entirely punched into the tape and the tape is accumulated (up to fifty feet) in the machine.

Now the operator is ready to allow the program tape to take over control of the machine. He first turns off the PUNCH switch and turns on the DUP switch. The DUP switch causes the code symbols read by the program tape reader to be duplicated back into program tape that is passing by the program tape punch. The program tape thus continuously recreates itself sequentially so that it may be used again and again. The program tape is placed in control of the machine by depressing the PTR key.

(Program Tape Reader)

The depression of this key tells the machine to turn control over to the instructions punched into the program tape, which is standing ready to be read at the Program Tape Reader. The first character encountered by the reader will be the first step in the program to be executed.

(Keyboard)

When the 610 is under the control of the program tape unit, it may be necessary to give an instruction to return control to the keyboard.

This instruction causes control of the machine to be transferred from the unit in control to the keyboard. One use of this instruction is to return control from the program tape reader to the keyboard for the entry of data. An instruction sequence 05 ENT KB will cause the machine to transfer control to the keyboard for data entry. After the number has been entered, depressing PTR will return control to the program tape reader.

Instruction Classes

In many problems, it may be necessary to choose one of several possible methods of solution after some point in the calculation, depending on conditions generated up to that point.

For example, in the calculation of the roots of a quadratic equation (previously used example), it is possible that the discriminant ($b^2 - 4ac$) may be negative. If the discriminant is negative, the roots of the quadratic equation will be complex numbers and it will be necessary to separate the real and imaginary parts for printing. The possibility of having complex roots was not considered in our previous example.

In solving for the roots of a series of such equations, it is usually inconvenient to separate by visual inspection the equations having real roots, from the equations having imaginary roots, and then solve them with separate programs. It would be extremely desirable to have a program that would solve for either condition, dependent upon the sign of the discriminant. Because all symbols on the program tape must be read in sequence, a problem is posed having both the series of steps for finding the roots of an equation with a positive discriminant and the series of steps for finding the roots of an equation with a negative discriminant on the same tape. It would be very convenient to have the machine make a logical decision and use only one series of steps while ignoring the other. The IBM 610 is able to make such a logical decision.

In order to provide for a logical decision while producing a program tape, the operator may punch each instruction in any one of four classes (0, 1, 2, 3) by turning on the appropriate PUNCH CLASS switch. The operator will punch the program for two alternate processes in consecutive order in, say, classes 1 and 2. The main program is usually punched in class 0.

In the event that the first of the two alternatives is chosen, the 610 will be set to respond to class 0 and 1 instructions.

The class 1 instructions would logically be read first; and if the machine is set to respond to that class, the data is processed by that portion of the program. The program tape reader next encounters the instructions punched in class 2. Because the machine has been set to respond to only classes 0 and 1, the machine reads (and duplicates) but does not respond to class 2 instructions and will not re-act to any instructions until it again reaches class 0 or 1 instructions.

If the second of the two alternatives is chosen, the 610 will be set to respond to class 0 and 2 instructions. The class 1 instructions of the first alternative will be encountered, reproduced (if the DUP switch is on), and ignored by the machine. When the machine has read, reproduced, and ignored the class 1 instructions, it will encounter class 2 instructions. It will read them and process the data. The machine will continue to process data according to class 0 and 2 instructions until otherwise instructed. In either case, the IBM 610 has made the decision as to which of the two sets of instructions to use.

Regardless of class response switch settings, if the DUP switch is on during control by the program tape, all instructions will be reproduced. In effect, unused class coded instructions become NO OPERATION steps that are used but never executed.

Program Tape Response Class

The program tape punch and reader are eight-channel devices. Of these eight channels, five are used for code symbols associated with operation keys, one channel is used for even-order redundancy check, as described in the *Tape Code Section*. The remaining two channels may be used to record the *class marks* previously described, and the computer can be caused

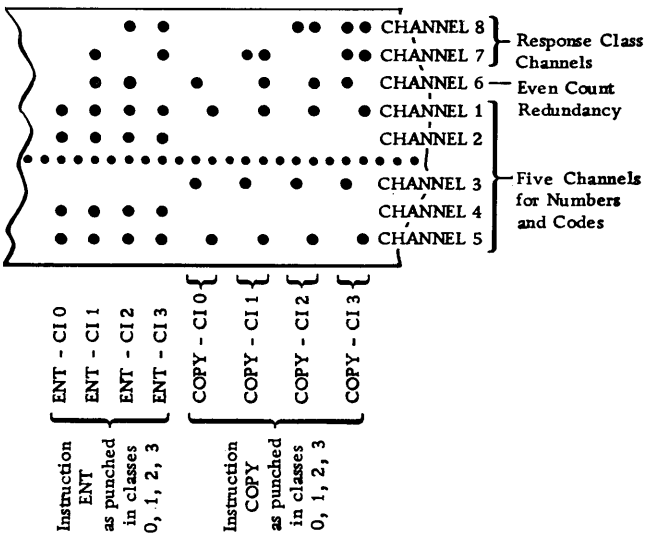


FIGURE 7. TAPE RESPONSE CLASSES

to obey or ignore any given instruction code symbol depending upon its class mark and the positioning of the response class switches (Figure 7).

In the group of switches on the keyboard controlling the program tape operation is a set of three switches labeled PUNCH CLASS. Normally, all these switches are off; and, if the program punch switch is on, the program tape punch records code symbols in the tape, leaving the channels 7 and 8 unpunched. The program is said to be recorded in class 0. If the switch labeled *1* is turned on, the same channels are punched for a code symbol, but in addition, the seventh channel is punched. If switch *2* is on, the same channels are punched for a code symbol, but in addition, the eighth channel is punched. If switch *3* is on, the same code channels are punched, but in addition, both channels seven and eight are punched.

When the control of the machine is turned over to the program tape reader, the operator may set the response class switches to classes 0, 1, 2 or 3. If the 0 switch is on, the machine responds to all instructions with no punches in channels 7 and 8. If the 0 switch is off, the machine will not respond to these instructions. In a similar manner, the switches may be set to respond to classes 1, 2 or 3 or any combination.

The class response may also be turned on automatically from the control panel as explained in the control-panel discussion.

During the planning of a problem for the 610, the operator must select the class responses to be used.

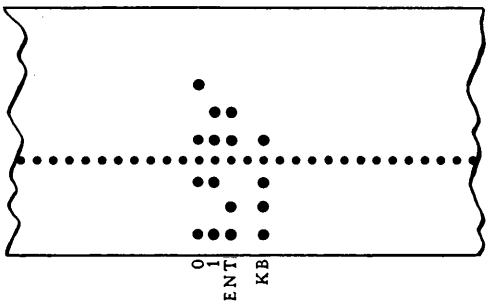
Program Tape Example (Figures 8, 9, 10)

IN ORDER to explain the method of operating the program tape, we shall give an illustrative example. Suppose that the operator wishes to calculate the value of the quantities $N_1 N_2/(N_1 + N_2)$ for each of several pairs of values. The programming for each pair would obviously be the same. In such a situation, the rapid setup facility provided by the program tape unit and the higher speed of the machine under program tape control will expedite substantially the performance of the calculations. Briefly, the operator can instruct the program tape unit to copy the steps performed for a manual first calculation. Thereafter, the program tape assumes control to give instructions repeated from cycle to cycle, pausing and returning control to the keyboard to allow the operator to key-in appropriate data and then returning control of the machine to the program tape to complete the calculation.

In order to set up the machine for operation in this manner, the operator proceeds as follows: First, he turns on the PUNCH switch. With this switch on, a code symbol will be punched in the program tape corresponding to each key depressed on the manual keyboard. After passing through the program punch, the tape accumulates in a slack loop between the program tape punch and the program tape reader. A slack loop of tape up to 50 feet in length can be accumulated. When the tape accumulated in the slack loop is ready to be read, any blank tape will be passed by the reader. When the first character is read, the program tape reader will cause the machine to act upon the program instructions.

After the operator has set the PUNCH switch, he next gives the instruction:

01 ENT KB

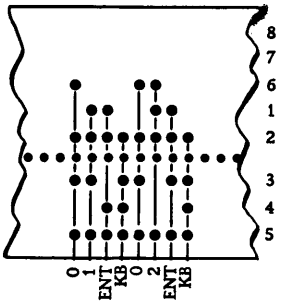


(ALL IN CLASS 0)

FIGURE 8. TAPE REPRESENTATION OF 01 ENT KB

That is, he selects and clears register 01 in preparation for the entry of numerical data. Observe that when the PUNCH switch is on, it is *not* desirable to have the data following ENT KB punched into the program tape and, hence, *actual data punching is always automatically suppressed following this combination of instructions.* Later, when the program tape is in control, the KB instruction will return control to the keyboard so that numerical data may be entered. The operator now enters the first numerical value N_1 which concludes with its sign. The next line of coding is:

02 ENT KB



(ALL IN CLASS 0)

FIGURE 9. TAPE REPRESENTATION OF TWO ENTRY STEPS

After the first value of N_2 has been entered, the operator proceeds to key the first calculation as he would any program:

```
03 COPY 01
     +  02
01 ÷×  03 02
   CR
   RO
```

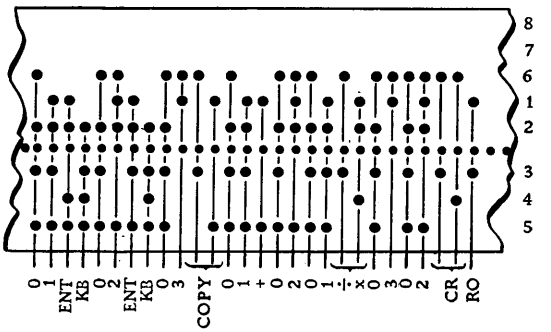


FIGURE 10. TAPE REPRESENTATION OF ENTIRE PROGRAM

Note that the carriage return (CR) instruction causes the return of the typewriter carriage before read-out (RO) so that the results can be tabulated in a vertical column. (These two operations will be explained with the output commands.)

Once the program of calculation has been completed and stored on tape, the operator turns off the PUNCH switch and turns on the DUP switch. With the DUP switch on, the information read by the program reader will be treated as instructions and will simultaneously be repunched by the program punch into blank tape immediately following the first program. Thus, after a given program has been executed, it has been repunched and moved into the slack loop available at the input of the program tape reader for use in subsequent operations.

The computer is now ready to repeat the routine as many times as may be desired. To begin processing the second set of data, the operator depresses the PTR key, transferring control to the program tape reader. The machine proceeds with the calculation until it is time for a numerical entry, at which point it returns control automatically to the keyboard (as indicated by the KB light coming on).

The operator keys in the appropriate number and returns control to the program tape reader by again depressing the PTR key. This process is repeated until the data to be processed has been exhausted with the program tape being automatically repunched each cycle and the answers being typed out in a vertical column for all pairs of N_1 and N_2 entered.

Of special interest is the fact that while the programmer was setting up the automatic program on the keyboard, the first set of data was actually processed. Thus on the 610 the programming itself does useful work.

DATA TAPE

THERE may be occasions when it becomes desirable to record final or partial results in punched paper-tape form, either permanently or temporarily for later use during the computation. Of the two paper-tape punches on the 610; one of these, called the Data Tape Punch, is normally used for these purposes. The operator may direct the machine to punch information into this data tape. This tape can accumulate in the machine up to fifty feet and can be automatically read back into the 610 at a later time if desired. Results of the processing of several sets of data may be stored in the data tape and printed out at one time.

Reading-In from Data Tape

If the operator has punched data into the data tape for automatic entry, he will want to write a program that secures data from the data tape as required by the program.

DTR (Data Tape Read)

When this command is given, the data tape reader is turned on and the next number punched in the tape is read and transferred to the selected storage register on the magnetic drum. At the end of each such number on the data tape, a special two-character symbol (called the *word-end* symbol) is punched. When the word-end symbol is read by the tape reader, an electrical impulse is sent to the control panel. This impulse must be used to transfer program control to some other control device by means of control-panel wiring (i.e., to the control panel itself, to the keyboard, or to the program tape reader); otherwise, the machine will continue to read data from the tape until it runs out of tape or until the master stop switch is turned off. *When the data tape reader is used, it is absolutely necessary that the word-end on the control panel be wired to activate a control device.* A full discussion of control-panel wiring is given in a later section.

The data tape will be treated at greater length in the output section of the manual.

Deleting Characters on Tape

If the operator punches a digit or digits or an operation symbol into the tape incorrectly, he uses the delete instruction to make that section of the tape non-effective.

When this key is depressed, it causes the tape to be punched in all channels on that segment under the punch head. The operator will normally backspace the tape manually until the incorrect section is under the punch head and then depress the delete key.

If the operator is typing the program as he punches it, slashes will be printed through the incorrect symbols when DEL is depressed. If an operation has a two-position tape symbol, both positions must be deleted.

Incorrect Key Depression

Occasionally during the calculation of a problem, the operator may depress the incorrect digit for a register selection or an incorrect operation key. In order to negate such a register selection or operation, the operator depresses the following key:

The depression of this key tells the machine to release the incorrect operation or register selection. The A register is selected by this key depression. The operator must now depress the correct digits to select the correct register, if that was the error. If the operator has depressed the incorrect operation key, he must re-select the correct register (even though it had been correctly selected) and then depress the correct operation key.

EXAMPLES:

1. If the operator depresses the operation key $\sqrt{\ }$ by mistake when he meant to divide in the sequence 08 $\div$ 09, then he must depress REL, then 08 $\div$ 09.

2. If the operator selects the register 15 when he meant to select 14, he depresses REL, then 14 and proceeds as though nothing had happened.

Interrupting Automatic Operation

Two other keys on the keyboard allow the operator to interrupt and restart a program at any time. The automatic programming will stop so that the operator may investigate the contents of various registers, etc., before resuming calculation. Careful operating procedure is required to insure that the program starts again at the right place.

The depression of this key while the 610 is under automatic control tells the machine to stop and to stand ready to accept instructions from the keyboard. When this key is depressed, the INT light on the keyboard comes on. If the automatic program is in the midst of an operation, the OPERATION light will also come on. The OPERATION light *must* be turned off before anything else is done from the keyboard. To do this, the INT key is held down and the RSM key is repeatedly depressed until the OPERATION light goes off. The operator should then carefully record the register standing selected, as indicated on the calculator light panel, before attempting to examine the contents of storage registers or perform any other operations from the keyboard. While the machine is in the interrupted state, the operation light will be on when the machine is instructed to perform an operation, thus warning the operator not to resume until he completes the operation or releases the machine.

NOTE: The INT key is the only manual transfer to the KB while the PTR, DTR, or CP is in control.

Resuming Automatic Operation

RSM (Resume)

When the operator has completed the desired operations on the keyboard, he reselects the register that stood selected at the time the program was interrupted. He then depresses the RSM key.

The depression of this key tells the machine to begin at the point where it was interrupted by the depression of the INT key.

If the operator wishes to examine the program by going through it step-by-step, he holds down the INT key and depresses the RSM key for each step. Because some operation codes have a double symbol in tape code, two depressions are necessary for these steps. The operator should be careful not to depress any of the keys capable of transferring control. If a control transfer is depressed, there is no easy procedure for getting back to the correct portion of the program.

NOTE: To assure correct restart after making an erroneous key depression, the operator must carefully analyze what has taken place. If the error caused register contents to be destroyed, they must be replaced before resumption of programming.

EXAMPLE: Depressing (01) CLR RH in place of (01) + would destroy data in register 01, impossible to restore, except by restoring the original contents of 01.

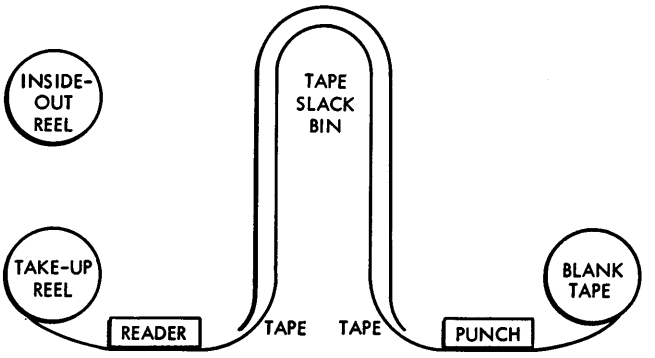


FIGURE 11. USUAL TAPE ARRANGEMENT

If the program is quite lengthy, the program tape can be placed in the inside-out reel rather than duplicating the program each time. The tape is mounted on the inside-out reel, threaded through the read unit, and rewound on the punched-tape take-up reel. When the lengthy program has been used to process a set of data once, it can be removed from the take-up reel, placed on the inside-out reel, threaded, and the program is ready to run again (Figure 12).

If the data punched into the data tape is to be used repeatedly, it can be duplicated by means of control-panel wiring or it can be passed from the inside-out reel to the take-up reel in the same manner as the program tape.

PHYSICAL ARRANGEMENT OF TAPES

THE internal arrangements of the program tape and data tape units are the same. The program tape unit is on the left side of the machine; the data tape unit is on the right. The internal arrangement is indicated in Figure 11.

Blank tape is loaded onto the blank tape reel and threaded through the punch unit. For ease of operation, a piece of tape from one to two feet long should protrude from the punch unit at the beginning of program punching. When the program is ready to be read, the end of the tape is threaded through the read unit and affixed to the punched-tape take-up reel. The slack tape containing the program is stored in the tape slack bin.

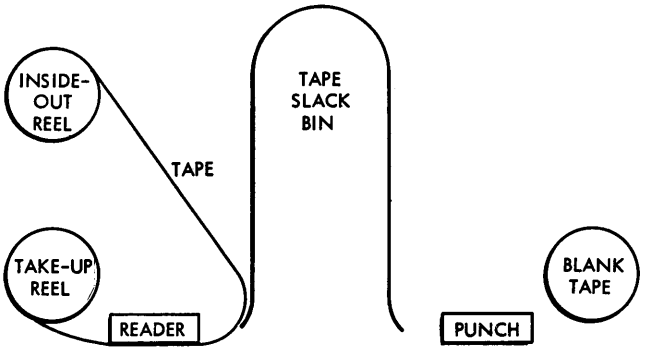


FIGURE 12. TAPE ARRANGEMENT USING INSIDE-OUT REEL

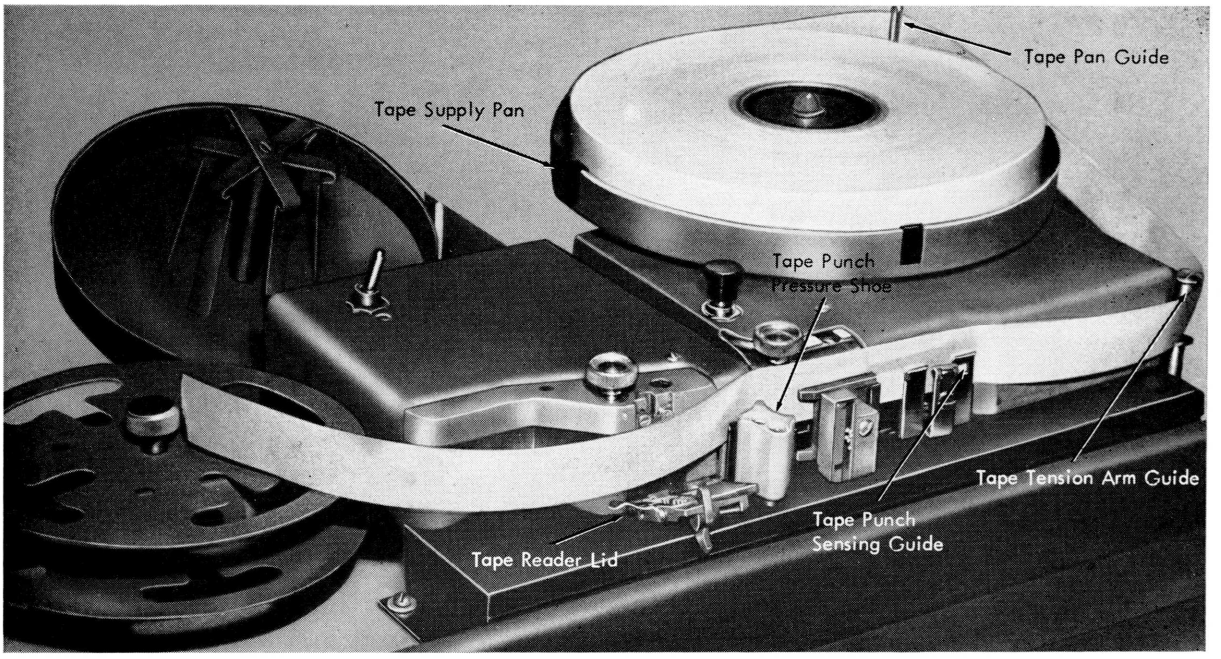


FIGURE 13. TAPE LOADING

Tape Loading

Eight-Channel Tape. The first step in the procedure is to load the tape supply pan (Figure 13). Next the tape is threaded through the tape pan guide and around the tape tension arm guide.

At this point, open the tape punch pressure shoe and the tape reader lid. Then, slip the tape between the tape punch housing and the tape punch sensing guide.

Now, position the tape as shown in Figure 14 so that the tape fits snugly into the tape reader pin guide plate and the tape punch die. Be sure that the tape is threaded through the tape punch sensing guide exactly as shown in this picture. Proceed to insert the tape end into the take-up reel.

Next, referring to Figure 15, close the tape punch pressure shoe and the tape reader lid and replace the take-up reel cover. Finally, before operating the machine, advance the tape slightly by means of the tape punch manual advance knob or, if the machine is in the *Operate* state, by means of the tape punch advance button.

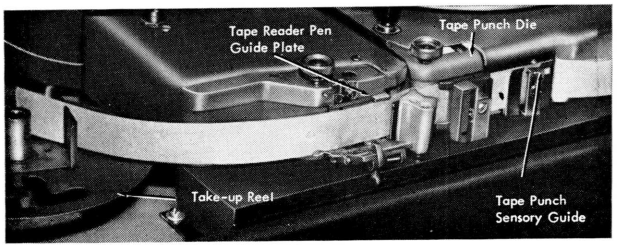


FIGURE 14

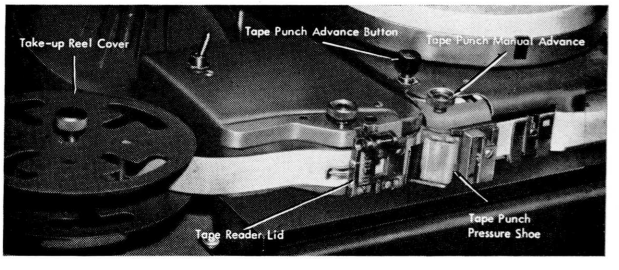


FIGURE 15

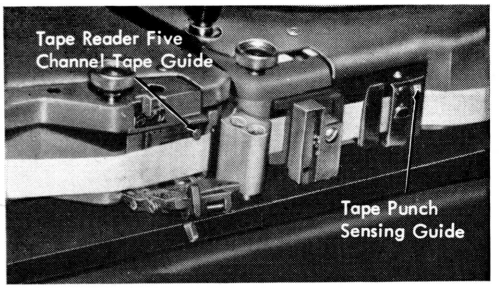


FIGURE 16

Five-Channel Tape. Figure 16 illustrates the threading of five-channel tape, which is identical to the threading of eight-channel tape except for the following differences.

First, the tape must be positioned completely to the bottom of the tape punch sensing guide so that it does not obstruct the small rectangular hole in the guide. A five-channel width plunger rides into the rectangular hole and senses that a five-channel tape is on the machine. The tape must be positioned in the tape reader so that it lies beneath the tape reader five-channel tape guide. The remainder of the procedure is the same. When completely threaded, the tape appears as in Figure 17.

INSIDE-OUT REEL

Eight-Channel Tape. First, place the tape on the inside out reel (Figure 18). Next, pull the inner end of the tape free, looping the tape around the tape reader. Slide the tape down into position, being careful that it enters the tape chute (not shown) through

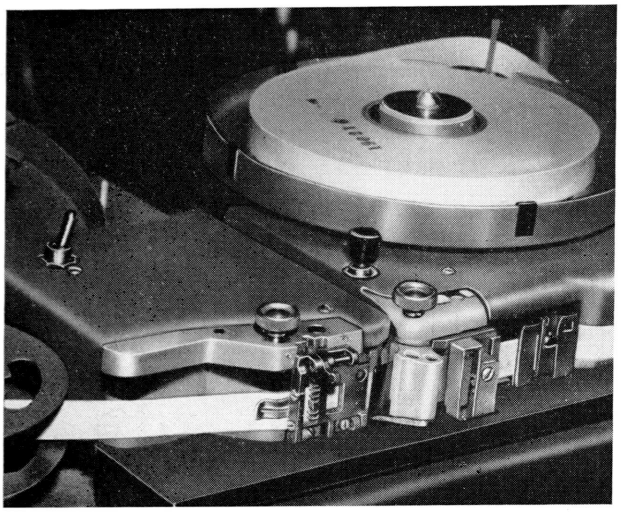


FIGURE 17

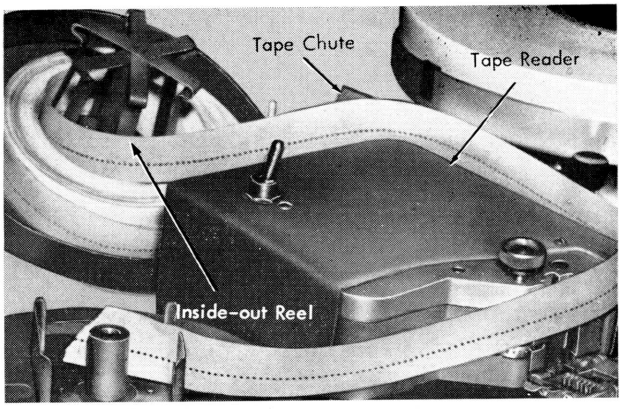


FIGURE 18

the slot provided at the rear of the tape reader cover, located below the arrow marked tape chute. The remainder of the threading procedure is the same as previously described for eight-channel tape. When completely threaded, the tape appears as in Figure 19.

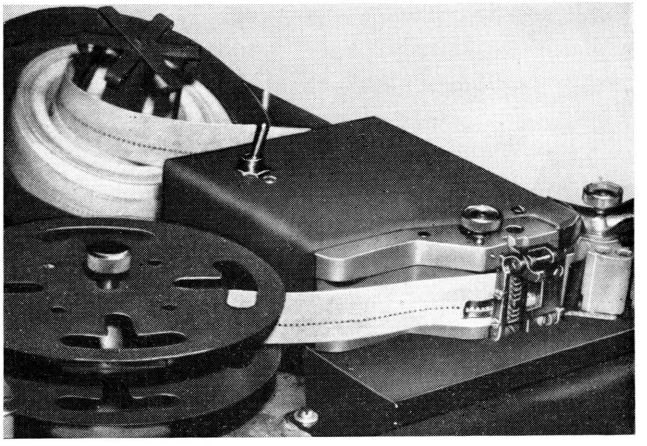


FIGURE 19

Five-Channel Tape. When five-channel tape is used, it must be positioned in the tape reader pin guide plate as described in the five-channel tape loading procedure. All other operations are the same as for eight-channel tape inside-out reel loading covered above.

NOTES: When tape is to be used through the tape reader alone, either blank or punched tape must be inserted in the tape punch.

If tape is to be used independently in the tape punch and tape reader, the free end of the tape, loaded in the tape punch, must be threaded into the tape chute between the tape punch and tape reader.

Tape Code

The tape may be thought of as ruled off into a rectangular grid, eight channels wide and as long as the tape. A single column of eight channels, some of which may be punched and some not, constitutes a character in tape code. More precisely, the bottom five channels constitute a character, the sixth channel contains an even count redundancy check, and the two top channels are used for class marks. In the dictionary of tape codes, there are 31 single-column characters, and 9 double-column characters, called doublets. The first column of a doublet is always the tape character for asterisk (*) or star and is called the modifier. The dictionary for translating from 610 language (i.e., keyboard symbol) to tape code is contained in the accompanying table.

The two channels on paper tape available for class marks are used to classify instructions on the program tape. The combinations of punches in these two channels give four classes of instructions, as has been previously described.

The sixth channel on the tape is called the redundancy channel. This channel is useful in verifying that every 610 character is correctly read. As each row is punched, the machine counts the punches in channels 1 through 5 and the two class channels. If the total number of punches is odd (1, 3, or 5), the machine automatically adds a sixth punch in the redundancy channel to make the total number of punches even. If the total count of punches in channels 1 through 5 and the two class channels is even, channel 6 is left unpunched.

On the tape layout (Figure 20) we have indicated the correspondence between the keyboard type program symbols and the tape hole code.

CONTROL PANEL

INSTRUCTIONS can be directed to the computing portion of the machine from the keyboard, program tape, and the control panel. We have described the method of creating a program and of causing this program to be punched into a program tape for the processing of similar sets of data immediately or at some later time. We have mentioned that the control panel is also capable of giving instructions to the machine. This is a particularly convenient control device for handling sub-routine sequences. Located on the control panel are selectors and balance test hubs capable of making choices between two courses of action. The operator need not turn control of the 610 over to the control panel to gain access to these devices.

Figure 21 is a picture of the pluggable control panel. The board is pierced by a large number of holes (indicated on the figure by small circles). Behind each of these holes, when the board is inserted in the machine, is a metal contact finger. Two of these contact fingers can be electrically connected by inserting the ends of a special wire into two of the holes on the board. This is called connecting two *hubs* on the control panel. If these hubs are interconnected properly, the computer will execute operations as required.

Most of the hubs fall into two classes: source hubs and function hubs. Source hubs emit impulses. Function hubs accept impulses and are connected to control terminals that activate the computing part of the machine. When a control terminal receives an impulse via a wire in the control panel, it causes the machine to perform its particular function. A function hub is similar to an instruction key on the keyboard ready

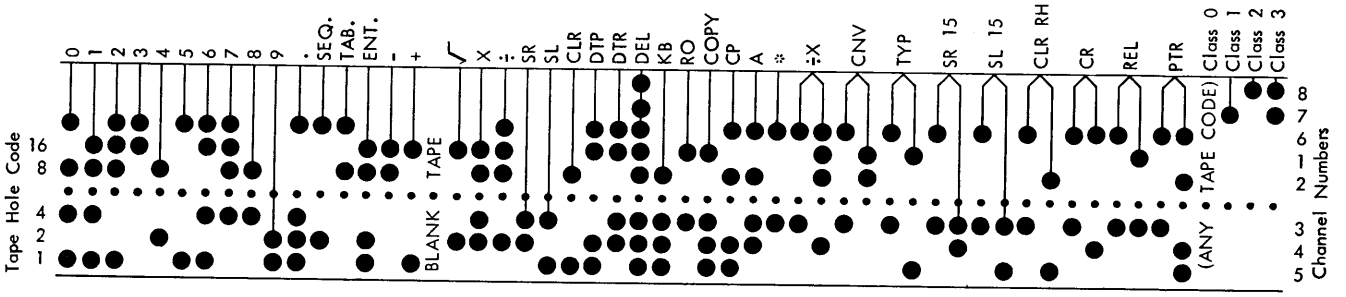


FIGURE 20. TAPE LAYOUT

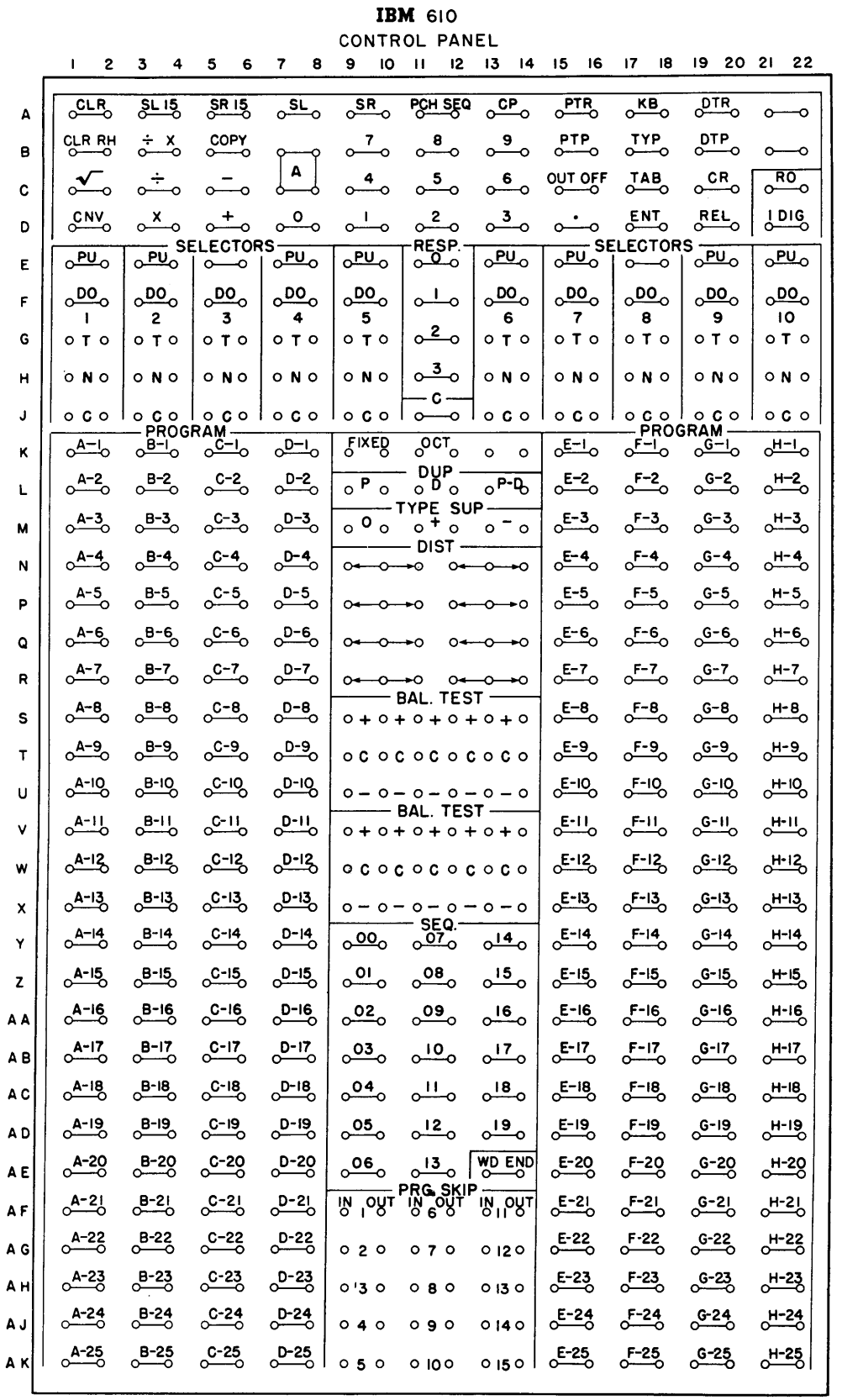


FIGURE 21. IBM 610 CONTROL PANEL

to be depressed. The key depression is equivalent to sending an impulse to the control terminal. In the case of a program tape, the machine *reads* the punched holes in the tape and causes an impulse to be sent to a control terminal.

There is a third class of hubs whose purpose is to affect the routing of electrical impulses from source to function hubs; logical operations depend upon the use of this class of hubs. Selectors, balance test, program-skip and response hubs fall into this class.

A set of source hubs (the program step hubs) is arranged on either side of the lower portion of the control panel (Figure 21). There are 200 pairs of these hubs numbered in order from A-1 through H-25. The two members of a pair are connected by a horizontal line. When an instruction is given to transfer control of the 610 to the control panel, these pairs begin to emit electrical impulses, starting with the step indicated by the light indicator panel on the machine. The program step hubs emit, one pair at a time, in sequence at the rate of 18 impulses per second. (The hub that emits after H-25 is the hub A-1.) Thus, when control-panel wires connect a succesession of program step hubs to a succession of function hubs, the machine automatically performs the corresponding functions or operations, one at a time, in order. This is similar to the succession of key depressions the operator makes when he is directing the activity of the machine from the keyboard.

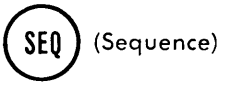
Once the control panel is wired, it is equivalent to a punched program tape. When the 610 is under control of the control panel, the program step hubs cause the machine to carry out the wired program by directing a sequence of timed electrical impulses to the set of control terminals within the machine. The order in which the control terminals are to be impulsed is selected by the operator's wiring on the control panel.

Transferring Control to the Control Panel

If a program involves several classes of instructions or transfers to other control devices, the operator normally does not attempt to compute as the program is being punched for the first time. However, in the execution of one-time programs, there may be subroutines wired on the control panel, which it is necessary to use in solving the problem. There are two commands that may be used in transferring control to the control panel.

Depression of this key causes the machine to transfer control of the 610 to the control panel. The control panel will begin executing instructions *at the next program step* in *sequence* as indicated by the light indicator panel. This means that if the last program step executed on the control panel was C-18, the lights will indicate C-19 and the machine will begin by executing steps C-19, C-20, etc. The operator may observe the next step to be executed by referring to the calculator light panel. (For a full discussion of the calculator light panel, see the section discussing the panel.)

Sequence Operation



The depression of this key followed by the depression of two digit keys (giving a number in the range 00, 01, 02 ---, 19) will cause the indicated SEQ hub on the control panel to emit an impulse that will enable the 610 to utilize the machine functions available on the control panel. A selector, balance test, etc., may be activated with the available impulse.

If the operator wishes to go to a program step on the control panel that is *not* the next program step, he uses a sequence (SEQ) operation. The primary uses of these SEQ commands are in decision-making.

Suppose, for example, that the operator wishes his program to make a decision based upon whether the contents of a register are positive or negative. He may give a sequence command, say SEQ 10, which will cause the SEQ 10 hub on the control panel to emit an impulse. This impulse may be wired through the balance test relays to make the decision, as will be explained.

Suppose the operator wishes to go to a sub-routine wired on the control panel beginning at PROGRAM STEP hub E-20. He may give a sequence command, say SEQ 15. The SEQ 15 hub on the control panel emits an impulse, which is wired through a PROGRAM SKIP hub to a PROGRAM STEP hub E-19; E-20 will be the first program step executed under control of the control panel.

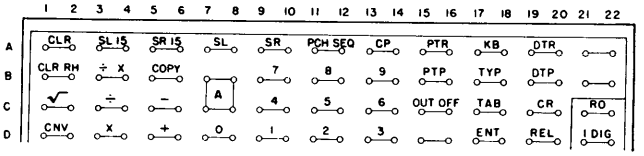


FIGURE 22. FUNCTION HUBS

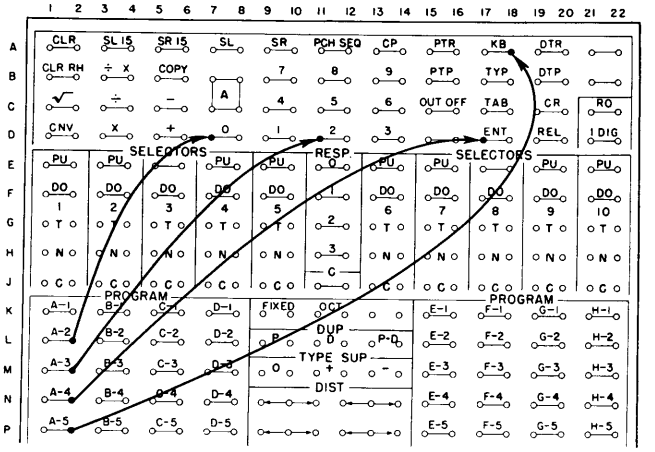


FIGURE 23. WIRING OF 02 ENT KB

Function Hubs

At the top of the panel is a set of function hubs corresponding, with a few exceptions, to the keys on the manual keyboard and arranged in approximately the same way. There are two hubs for each function, and these hubs are connected by a solid line (Figure 22).

02 ENT KB

This instruction on the program tape causes the machine to select the 02 register, clear it, and transfer control to the manual keyboard so that the operator can insert numerical data. After terminating the entry with an algebraic sign, the operator then sends control back to the program tape by depressing the PTR key. To make the machine perform these operations under control-panel control, the operator does the following wiring:

From PROGRAM STEP A-2 to FUNCTION hub 0

From PROGRAM STEP A-3 to FUNCTION hub 2

From PROGRAM STEP A-4 to FUNCTION hub ENT

From PROGRAM STEP A-5 to FUNCTION hub KB
(Shown in Figure 23)

As the PROGRAM STEP hubs A-2 and A-3 emit impulses in succession, the machine selects the 02 register. On step A-4, the machine is prepared for the entry of data. On step A-5 the machine transfers control to the manual keyboard and stops. After the operator enters the appropriate numerical information, he depresses the key marked CP, transferring control back to the control panel. The control panel takes up the program from the step where it left off, emitting from steps A-6, A-7, etc.

NOTE: When the keyboard or program tape executes an operation, the corresponding function hub on the control panel emits an impulse.

Program Skip Hubs

Unless instructed to the contrary, the program step hubs are energized in sequence, starting over automatically with step A-1 after H-25. However, a mechanism is provided whereby the operator may cause the 610 to skip from any program step to any other program step on the control panel and to take up from there the routine of energizing program step hubs in sequence. This is accomplished by means of the fifteen pairs of program skip hubs located at the bottom center of the control panel. They are arranged in associated pairs of IN and OUT hubs. If the operator wishes to cause a jump from PROGRAM STEP A-25 back to STEP A-8, he connects the STEP A-25 to the IN terminal of one of the program skip pairs, say PROGRAM SKIP 1, and the OUT terminal of PROGRAM SKIP 1 to PROGRAM STEP A-8. The OUT terminal is connected to the program step immediately preceding the first step the operator wishes to be executed. In this case, the operator wanted PROGRAM STEP A-9 to be the next step executed. The Program Step hub connected to a PROGRAM SKIP OUT hub cannot be used for any other purpose.

The wiring is shown in Figure 24.

This series of steps suggests one of the important uses of the control panel—an *iterative loop*. That is, a series of steps may be repeated a number of times. Usually, however, it is necessary to repeat the steps only a specific number of times, or until a certain mathematical value is derived, at which time the computer exits the loop. Methods for getting out of the loop will be explained as applications are developed in the following pages.

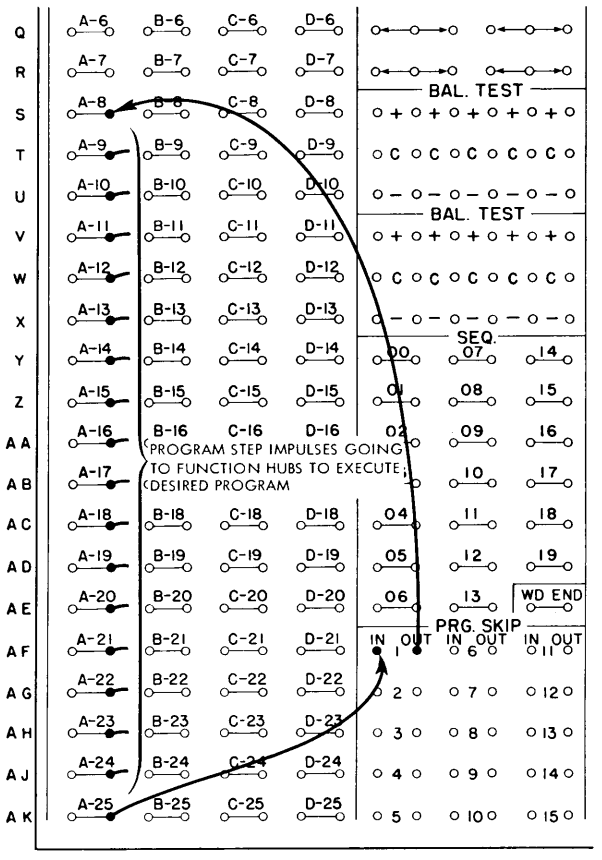


FIGURE 24. PROGRAM SKIP WIRING

Type Suppression Hubs

(Figure 23, Upper Center Portion)

There are three pairs of TYP SUP hubs: 0, +, and −. When the (+) hubs are bridged (wired together), the typewriter will not type the plus sign. Similarly, if the (−) hubs are bridged, the minus sign will be suppressed. If the (0) hubs are bridged, all zeros to the left of the most significant digit will be suppressed.

Sequence Hubs

The sequence (SEQ) hubs on the control panel (Figure 21) can be made to serve as source hubs. For example, if SEQ 17 is read from the program tape, the pair of hubs SEQ 17 will emit an impulse which may be used in the same manner as a program step pulse. One common use is to cause the program unit in the control panel to skip to a particular section of the control panel for execution of a given subroutine, such as computing Sin X, Cos X, etc. At the end of the subroutine, control may be transferred to the program tape reader (Figure 25). It is not necessary to turn control of the machine over to the control panel to use the SEQ hubs (and certain other hubs on

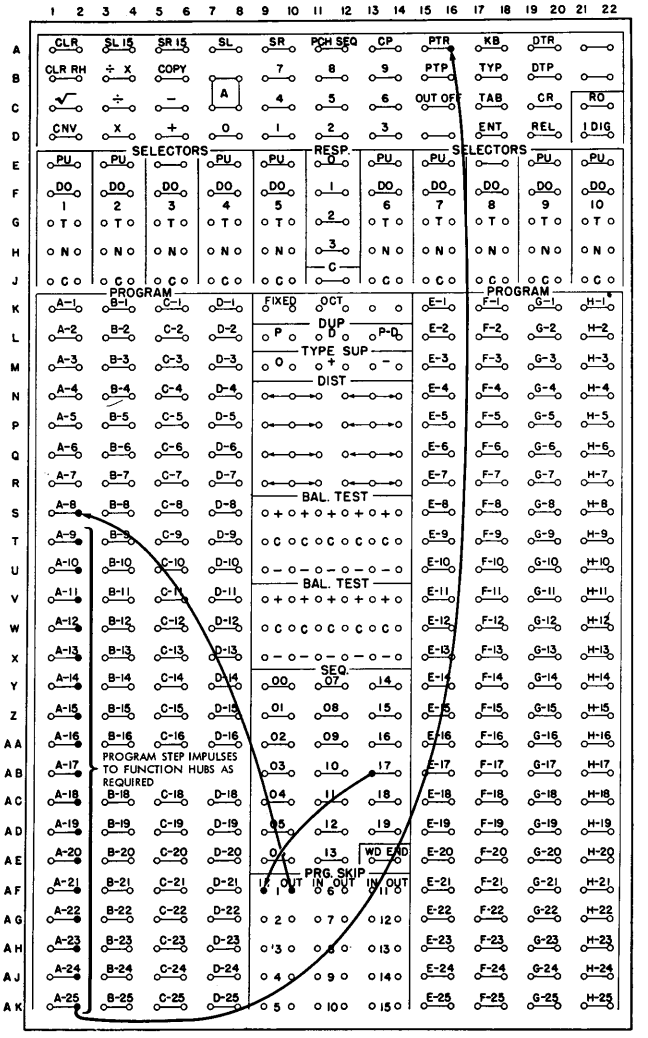


FIGURE 25. USE OF SEQUENCE HUBS

the control panel) as will be shown in the following discussion. Control may be considered to be vested in the Control Panel only when the program step hubs are caused to emit pulses. If, for example, SEQ 09 is energized by action of the Program Tape, and if it is used to select Class Response 2 (Figure 21), control is not relinquished to the panel.

Selectors (Figure 26)

Near the top of the control panel are ten sets of hubs labeled *Selectors*. These, like the program skip

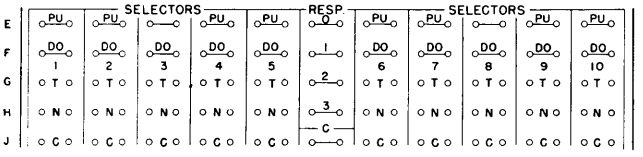


FIGURE 26. SELECTORS

hubs, fall into that class of hubs whose use is to affect the routing of impulses from source hubs to function hubs. The selectors provided in this machine are controllable double-throw switches with their control connections brought out to PU and DO (pick-up and drop-out) hubs. An impulse delivered to the PU terminal of a given selector throws the corresponding switch from the normal position to the transferred position. The switch remains at the transferred position until another impulse is delivered to the DO (drop-out) terminal. When the PU terminal has been impulsed, a connection is established between the C (common) hub and its associated T (transferred) hub. If the switch is at drop-out as a result of impulsing the DO hubs, a connection is established between each C hub and its associated N (normal) hub. *Changes in selector position become effective on the program step following the step on which the* PU *or* DO *hubs are impulsed.*

These two situations are illustrated in Figure 27.

Selectors may be used to modify connections made between hubs on the control panel during the course of a program as the needs of a particular problem arise.

As an example of the use of a selector, we show the wiring for typing out the magnitude of the contents of storage register 14 (i. e., without sign). If

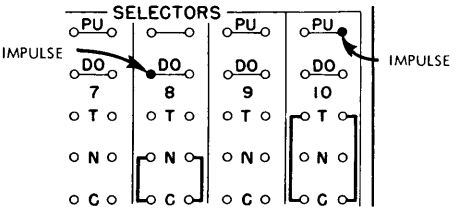


FIGURE 27. SELECTORS IN ALTERNATIVE POSITIONS

the machine receives the following instructions from the keyboard:

14 SEQ 00
CR
RO

with the control panel wired as in Figure 28 then we accomplish the desired result (printing the number in 14 without sign).

Here is an explanation of the wiring and the program:

14	Selects storage register 14.
SEQ 00	Impulses the SEQ 00 hubs on the control panel, which in turn impulses the selector PU hubs. The selector in the transferred position bridges the + and − type suppression hubs so that all numbers will be typed out without sign until the DO hubs are impulsed.

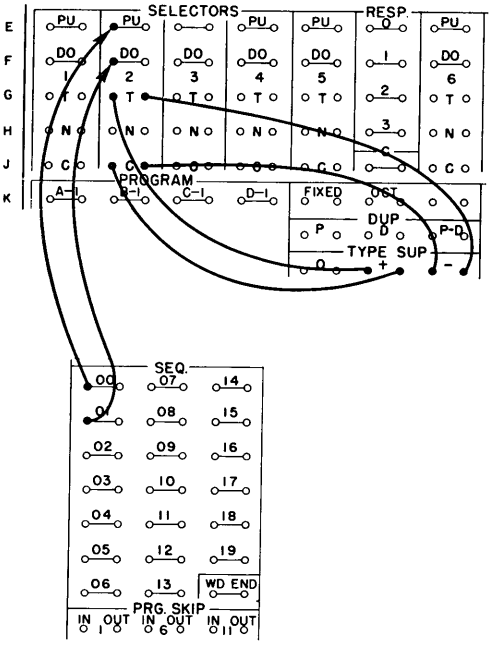


FIGURE 28. SIGN SUPPRESSION VIA SEQUENCE HUBS

CR Turns the typewriter on and returns the carriage.

RO Causes the contents of register 14 to be typed out on the typewriter without sign due to type suppression. When it is desired to begin reading out the contents of registers with their signs, read out (RO) must be preceded by a SEQ 01 to drop out the selector.

A selector may be used without actually turning control of the 610 over to the control panel. Suppose when a SEQ 09 command is given, we wish to turn on the CLASS RESPONSE 2; and when SEQ 10 is given, we wish to turn off CLASS RESPONSE 2. The CLASS RESPONSE 2 switch on the keyboard, must be set in the control-panel position. The wiring is shown in Figure 29.

If the instruction SEQ 09 is given on the program tape, the selector is picked up and all subsequent class 2 instructions will be effective. If the instruction SEQ 10 is given on the program tape, the selector is dropped out and all the subsequent class 2 instructions will be ignored.

The next instruction to the machine will come from the program tape and the program step hubs in *no* way affect the instruction pattern.

Balance Test Hubs

At the center of the control panel is a set of twelve balance test selectors or switches. When any one of the registers of the machine is selected, electrical contact is established between the C hubs and the + or − hubs, depending upon whether the number stored in the register is positive or negative. This is accomplished by a twelve-pole double-throw switch (relay) whose status is controlled by a sign-sensing circuit. The relay will switch automatically from + to − and vice versa, according to the sign in the selected register.

Zero is considered by the machine to be a positive number.

In the event the selected register is positive, the connections between C and + are shown in Figure 30.

Note that all positions are connected simultaneously.

If the number in the selected register is negative, the connections between C and − are shown in Figure 31.

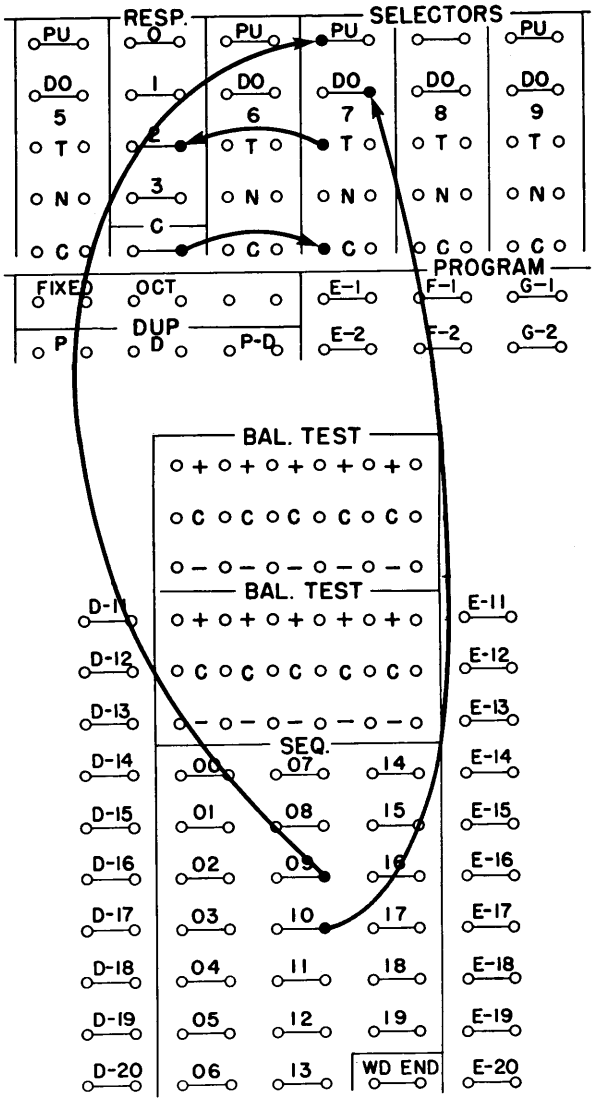


FIGURE 29. BALANCE TEST HUBS

Furthermore, if the sign of the number in the selected register changes in the course of an operation, the state of the balance test relay changes correspondingly to follow this change of sign.

The IBM 610 may be programmed to choose automatically one of several alternative procedures according to the result of an earlier part of the calculation. This is what we mean when we speak of the logical facilities of the 610.

The balance test relay is the logical element most frequently used in conditional programming of the computer. It may be shown that any required logical decision may be reduced to a decision based upon the sign of a numerical quantity standing in one of the registers.

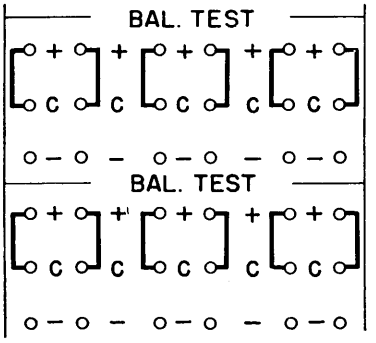


FIGURE 30. BALANCE TEST RELAYS WHEN SELECTED REGISTER IS POSITIVE

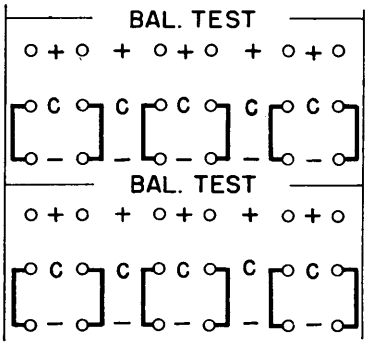


FIGURE 31. BALANCE TEST RELAYS WHEN SELECTED REGISTER IS NEGATIVE

Below is a simple example of the use of the balance test concept. Suppose it is desired to perform the operation *take the magnitude* of a number located in any storage register, for example 01. By use of the balance test relay, the operator is able to build this new operation into the machine. If the number in the register is positive, the operator wishes to leave it undisturbed. If the number is negative, he wishes to change its sign (*convert* it). The wiring for this operation is shown in Figure 32.

We have presumed that we were in the middle of a program on the control panel. On program steps G-6 and G-7 (arbitrarily selected) in the program, the operator selects register 01. The balance test relay immediately positions itself according to the sign of the contents of storage register 01. By control-panel wiring, the operator now sends an impulse from G-8 to the c terminal of a set of balance-test hubs. If the content of 01 is negative, then there is an electrical connection between the c and – terminals on the balance-test relay, and the impulse from G-8 passes through this connection to the CNV hub to initiate the convert operation as shown. The control panel then goes on to program step G-9. If the content of storage register 01 is positive, there is a connection between the c and the + terminals of the balance test relay. Because the + terminal is not electrically connected to a function hub, the impulse from G-8 accomplishes nothing, the program continues on to Steps G-9, G-10, etc., as may be required to perform other functions.

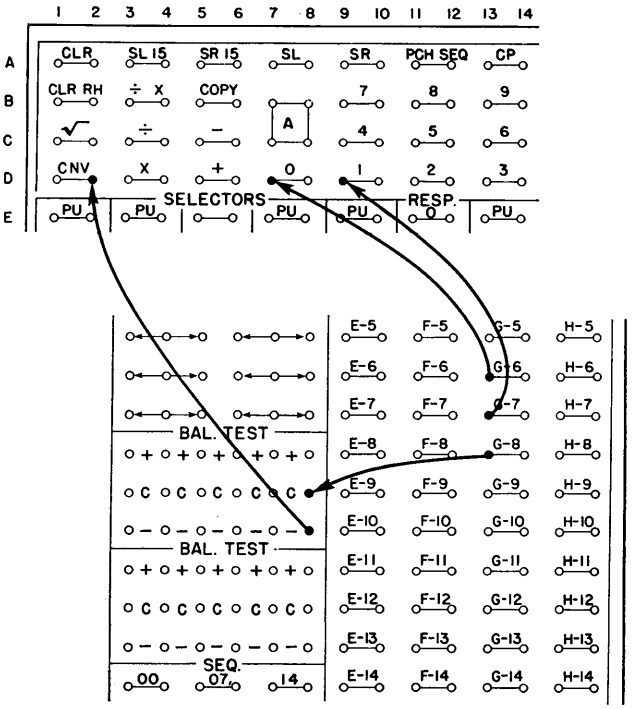


FIGURE 32. WIRING FOR OBTAINING ONLY MAGNITUDE OF A NUMBER

If, at some later step in the program, say H-11, we wish to convert another negative number, we need not wire from H-11 to C of a balance test hub and out of the – terminal to convert (CNV). Note that the only two hubs available to convert (CNV) were used early in the programming. Instead we now use *chain wiring* remembering that its use will not alter the sequence of program steps. In the example just stated we need only wire from H-11 to the second hub of G-8, and the remainder of the wiring from G-8 is effective. The wiring is shown in Figure 33.

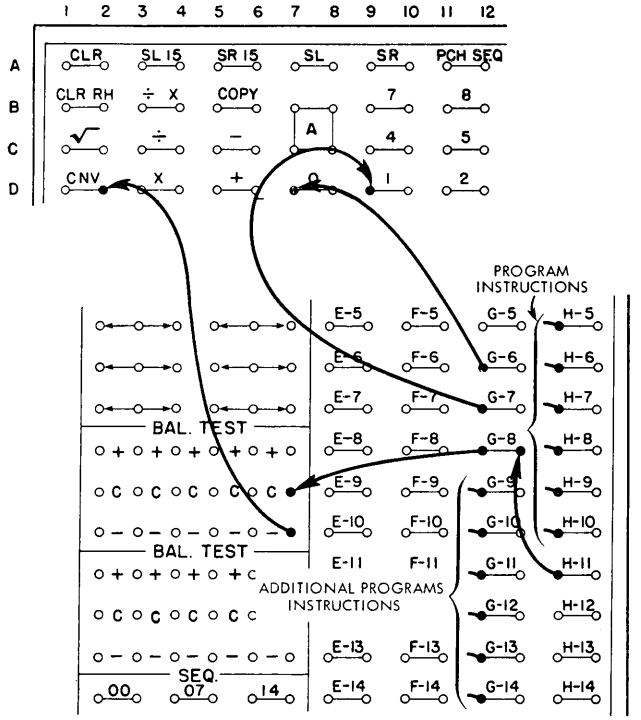


FIGURE 33. CHAIN WIRING FOR CONVERSION OF A NUMBER

The concept of chain wiring is very important to the process of panel wiring in that it provides greater capacity and flexibility.

Distributors

Just above the center of the panel are eight sets of hubs labeled DIST. Impulses can pass through a distributor from the center hub out to the outer hubs, but not in the opposite direction. Other names for distributors are *diodes* or *filters*. Suppose, for example, that on a certain step (A-7) we wish to energize the SR 15 function hub and, simultaneously, to pick up selector 3. Then, on a later step, say A-20, we wish to impulse the SR 15 function only.

The wiring is shown in Figure 34.

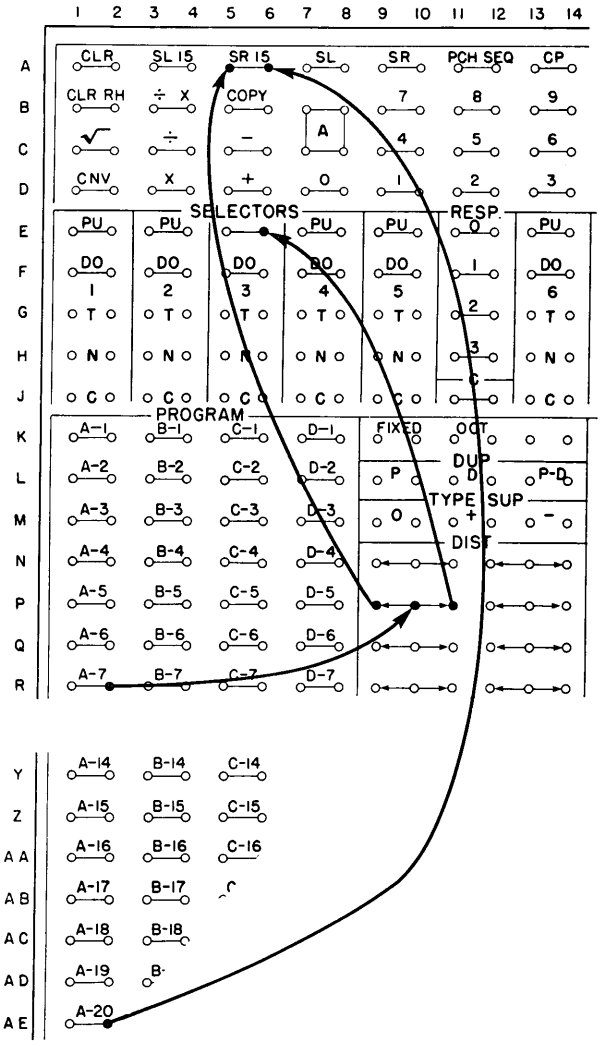


FIGURE 34. USE OF A DISTRIBUTOR

Because the distributor transmits impulses in *only one direction*, there is no danger of the impulse from program step A-20 travelling back through the SR 15 hubs and the distributor to the PU hubs of the selector to pick up the selector again if it has been dropped out between steps A-7 and A-20. Thus, distributors can be used to impulse two hubs from one source hub, or the same function hub from two source hubs, without danger of feedback.

An impulse sent to a distributor *must not be* directed to two function hubs simultaneously. Impulsing two function hubs is like depressing two key-

board keys at the same time and *must not be done.*

An impulse from a source hub used to perform two tasks (e. g., impulse a function hub and pick up a selector) should not be wired directly to both of the hubs. Instead, a single wire should connect the source hub to the center hub of a distributor and the two tasks will be performed by wiring from the two end hubs of the distributor.

Response Class Control Hubs

Between the two groups of selectors is a set of hubs labeled RESP. These hubs make it possible to control the response of the program tape reading unit to the instructions punched in a particular section of the program tape. If one of the C (common) hubs in this group is connected to one of the class hubs (say class 1) the program tape reader will respond to instructions in the tape identified with class Mark 1. In order that this may be a variable condition, the connection is usually made through a selector.

For example, suppose we wish the computer to respond to a certain section of the program tape if, and only if, the contents of register 14 are negative. This situation would arise in the case, mentioned earlier, the solution of the quadratic equation. If the contents of register 14 were the discriminant ($b^2 - 4ac$), the section of the program tape in question would contain the instructions for dealing with the case of complex roots. Suppose further that when the tape was punched, all the instructions in this section were punched in class 2. The control panel would be wired as shown in Figure 35.

On program steps C-6 and C-7, storage register 14 is selected. If the contents of 14 are negative, an electrical connection is established between C and − terminals of the balance test relay; if positive, between C and + terminals. On program step C-8, the common terminal of the balance test relay is impulsed. If the contents of 14 are negative, this impulse travels through the balance test relay to the PU terminal of selector 1. When the PU terminal is impulsed, the C and T terminals of this selector are connected. If the contents of register 14 are positive, the impulse travels through the balance test relay to the DO terminal of selector 1. When the DO terminal is impulsed, the C and N terminals of this selector are connected, thus assuring that C and T are not

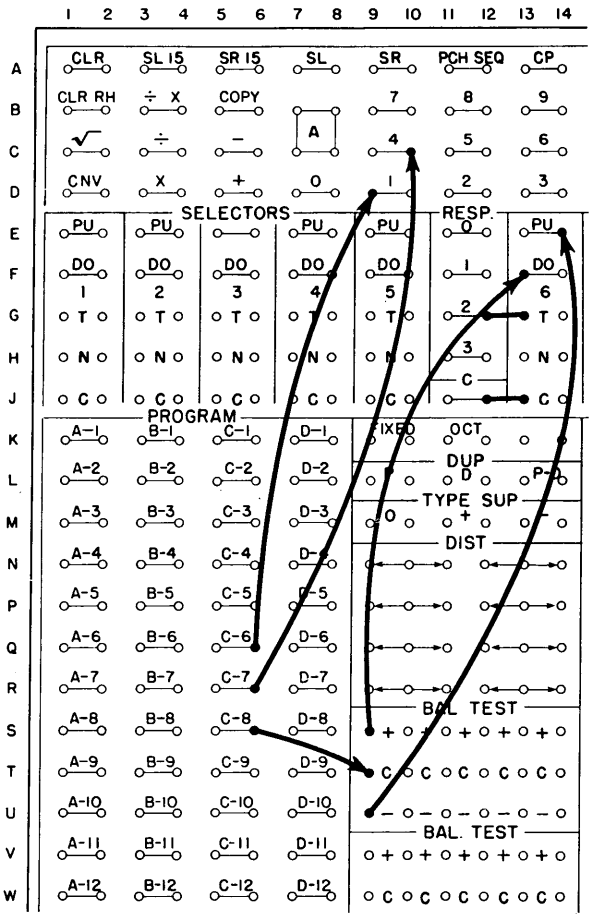


FIGURE 35. RESPONSE CLASS CONTROL

connected. If the register contents are negative and the C and T terminals are connected, this connection can then be used as a portion of the path connecting the common and the class 2 hubs in the group of class response hubs. Under these conditions, the computer responds to class 2 instructions on the program tape until the selector is dropped out.

For proper control panel action, it is necessary that the response switches on the keyboard be set correctly.

Assume that the main program has been punched in class 0, and that the portion of the program we wish to use as the result of a number being negative is punched in class 2. On the keyboard the *0* switch is set ON, the *1* switch is set OFF, the *2* switch is set to control panel, and the *3* switch is set to OFF.

Control panel response class control will be effective only if response class switches are set in the control-panel position.

Figure 36 shows the wiring of a fairly complex decision problem, the selection of any one of four response classes, depending upon the sign of the contents of register 14 and register 23. The table, below, lists the possible combinations of + or − using two registers, and the response class to be chosen.

Register 14	Register 23	Computer Response Class
positive	positive	0
positive	negative	1
negative	positive	2
negative	negative	3

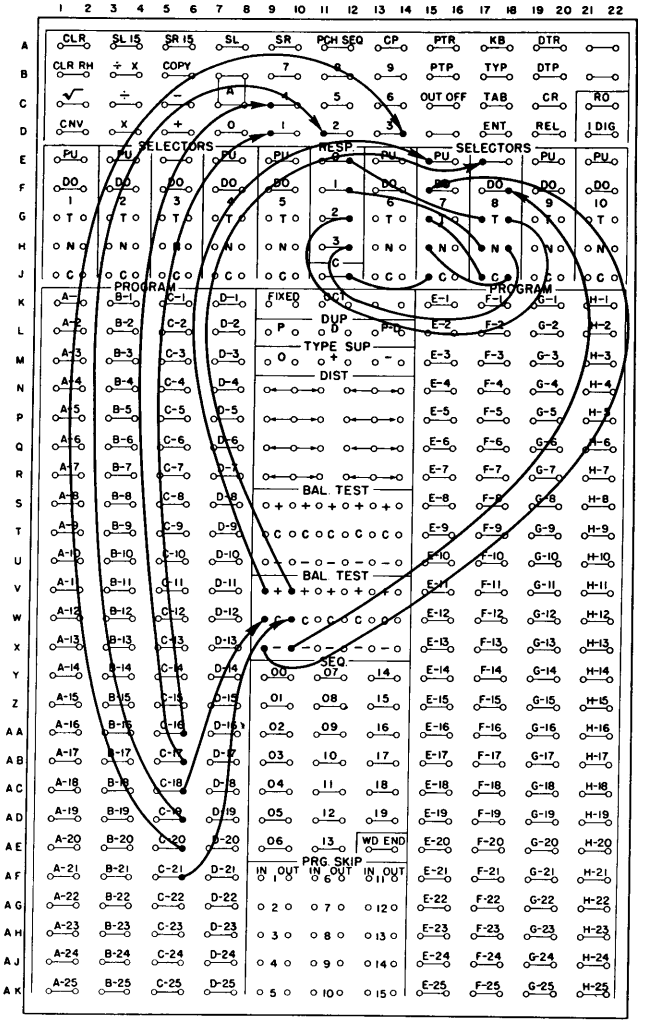


FIGURE 36. MULTIPLE RESPONSE CLASS CONTROL

Miscellaneous Hubs

The remaining hubs on the control panel are the following:

FIGURE 37. WORD-END HUBS (AE 13, 14)

Word End (Figure 37)

(Located at the lower right-hand corner of the group of sequence hubs on the control panel.) Whenever numerical information is recorded on the data tapes by the computer, the end of the number is automatically indicated by the punching of a special word-end code on the tape (tape section). When a number is read by the data tape reader, the reading of the word-end code causes the WORD-END hub to emit an impulse. This impulse notifies the computer that the entry from the tape is completed; the impulse can then be wired to a function hub to cause the desired program source to take control. If the control is to go to the program tape reader, the WORD-END hub is wired to the PTR hub. If the control is to go to the keyboard, the WORD-END hub is wired to the KB function hub. If control is to go to the control panel, the WORD-END hub is wired to the CP hub and the computer takes the next program step as indicated on the calculator light panel. However, if the programmer wishes to begin with a specific program step, he wires to the IN of a PROG SKIP and wires from the OUT of that PROG SKIP to the program step hub just preceding the first program step he wishes to be executed in a sub-routine wired on the control panel.

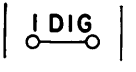


FIGURE 38. SINGLE DIGIT READ-OUT (D, 21, 22)

Single-Digit Read-Out (Figure 38)

At the lower right-hand corner of the group of function hubs is a connected pair labeled 1 DIG. When impulsed, these cause the IBM 610 to read out one digit at a time, thus allowing for variable word-size output. If the operator wishes to read out more than

one digit, the operator leaves unwired the following function hubs according to the number of digits he wishes to read out. If the operator desires to read out five digits, the first source hub (program-step) impulses 1 DIG for the first digit, four program step hubs are left unwired and then the next program step hub is wired to the OUT OFF hub to terminate the one digit read out. A SL 15 instruction should precede single-digit read-out in the auto-point mode (as is explained in the section on output).

The commands for typing the first three digits of register 14, for example, would be:

14 CR (1 DIG) (blank) (blank) (OUT OFF)

The wiring for this procedure is shown in Figure 39.

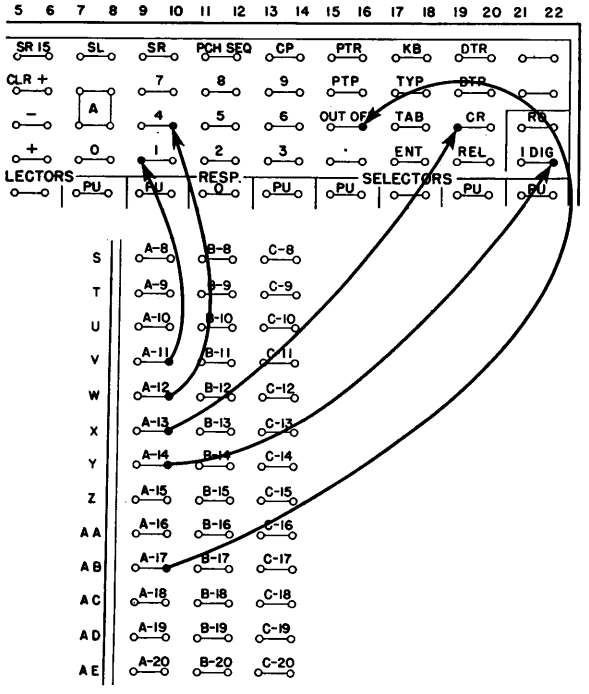


FIGURE 39. WIRING FOR SINGLE DIGIT READ-OUT

FIGURE 40. PROGRAM TAPE PUNCH (B, 15, 16)

Program Punch (Figure 40)

The pair of hubs labeled PTP in the group of function hubs, when impulsed, enables the program punch to record a number being entered or read out, thereby temporarily transforming the program punch into an auxiliary data punch. During this interval, the characteristics of the program punch are identical with those of the data tape punch. The controls exercised over the program punch by the PUNCH and DUP switches on the keyboard and the PTP hub are independent.

FIGURE 41. PUNCH SEQUENCE (A 11, 12)

Punch Sequence (in group of function hubs) (Figure 41)

When these hubs are impulsed, it becomes possible to punch sequence symbols (e. g., SEQ 06) on the data tape under control of the control panel. Such symbols may be used to mark the end of a group of data on the tape. When the PCH SEQ hub is impulsed after the data tape punch is turned on, the tape code corresponding to SEQ is punched into the tape. The two digits following SEQ are entered by impulsing control-panel digit hubs. Because punching into the data tape is normally terminated by punching the sign of the number, and we are not terminating a number at this time, we impulse the OUT-OFF hubs to turn off the data tape punch.

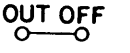


FIGURE 42. OUT-OFF HUBS (C 15, 16)

Out-Off (Output-Off Hubs) (Figure 42)

These hubs (in the function hub group), when impulsed, turn off any or all output devices that may be on at the moment. They can be used when the usual methods of turning off an output device are not

applicable. For instance, in the case of one-digit read-out it may be desirable to terminate the read-out after four or five digits (less than fifteen). Normally, the sign of the number terminates the read-out. In another case, in punching the group end designation, SEQ 06 as just described, the instruction would be:

DTP (PCH SEQ) 06 (OUT OFF)

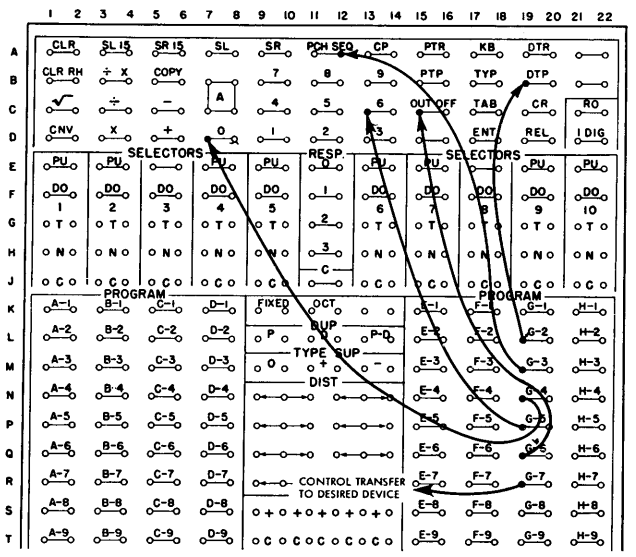


FIGURE 43. USE OF OUT-OFF HUBS

Consider a sample problem using the PCH SEQ and OUT-OFF hubs (Figure 43): We first punch the data in the data tape. Later, during processing, when we read the last piece of data, we wish to tell the machine we have processed all of the data. To do this there are several methods we may employ. The method we shall use here is to place a SEQ command in the data tape. When this SEQ command is sensed, the corresponding sequence hub on the control panel emits an impulse that can be used to initiate the steps of summary and print out.

If our sample problem was merely to sum all of the numbers in the data tape, our program could have been an iterative loop. We will use A as the accumulating register and enter each new number into register 01. The basic instructions are:

01 ENT DTR
A + 01

repeated as many times as necessary. Recall that the word-end symbol read from the data tape causes an impulse to be emitted at the WD END hub. This hub can be used to restart the series of instructions. The instructions are wired on the control panel (Figure 44).

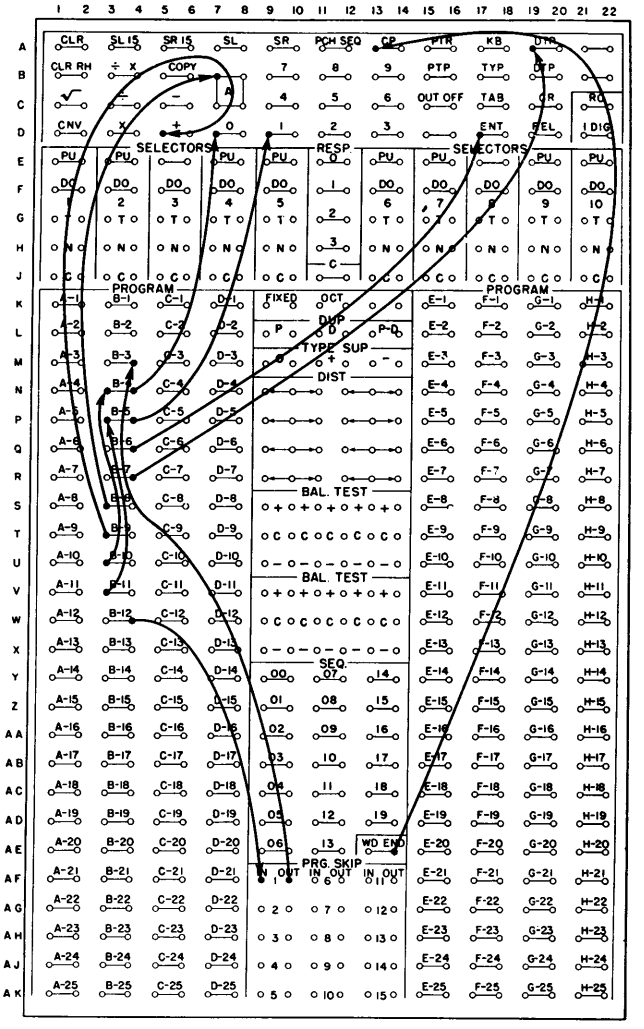


FIGURE 44. PROGRAM FOR SUMMING NUMBERS FROM DATA TAPE

The computer may be instructed to stop repeating the iterative process, above, by placing a SEQ command on the data tape. When the DTR hub is impulsed by step B-7, the SEQ 01 command will be read on the data tape and the SEQ 01 hub will emit a pulse which causes the program to skip as indicated in Figure 45.

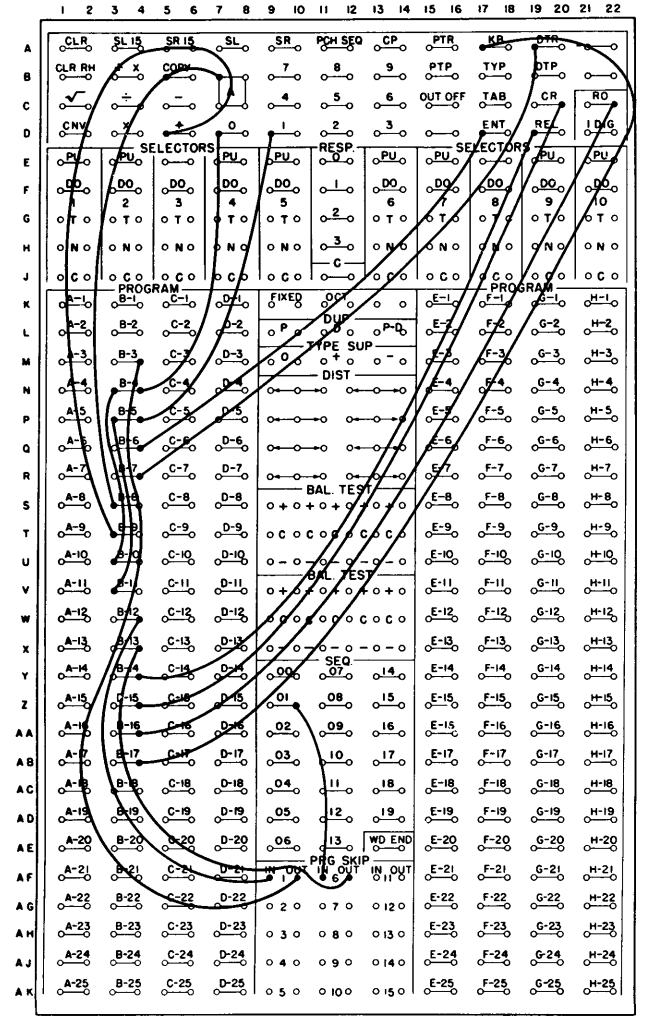


FIGURE 45. PROGRAM FOR SUMMING NUMBERS FROM DATA TAPE WITH "BREAK-OUT"

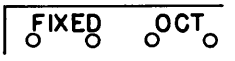


FIGURE 46. FIXED HUBS (K 9, 10) AND OCTAL HUBS (K 11, 12)

Fixed Point (Figure 46)

The pair of hubs labeled *fixed,* when one is connected to the other, places the machine in fixed-point mode, provided the switch FIXED POINT on the keyboard is in the control-panel position.

Octal (Figure 46)

The pair of hubs labeled *octal,* when one is connected to the other, makes the machine perform all operations in octal arithmetic, provided the OCTAL switch is in control-panel position.

If both the FIXED and OCT hubs are bridged, the computer operates in Fixed Point-Octal.

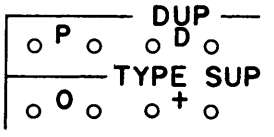


FIGURE 47. DUPLICATE HUBS (L 9, 10, 11, 12)

The REL hub must be impulsed to release the 01 register, because the preceding sequence of instruction (i.e., 01 ENT DTR) has selected 01, cleared it, and prepared it to receive data. Because no data is actually entered, an artificial indication must be given to tell the machine that data entry has been terminated. The REL hub here serves precisely the same purpose that the sign on the end of a word of data normally serves. Once the register 01 is released on step B-13, and A register is selected, the contents (the desired sum) of the A register are read out.

Dup (Figure 47)

Under the set of DUP hubs is a pair of hubs labeled *P.* When these two hubs are connected, information read at the program tape reader will be duplicated at the program tape punch, provided the DUP switch on the keyboard is in control-panel position.

When the two hubs under DUP labeled *D* are connected, material read at the data tape reader will be duplicated at the data tape punch, provided the DUP switch on the keyboard is in control-panel position.

OUTPUT DEVICES

THE OUTPUT DEVICES associated with the 610 are the

IBM Electric Typewriter, Punched tapes, and the Cathode ray tube (Figure 48).

The output devices may be operated singly or in any combination whenever numbers are being read out from the computer, or whenever numbers are being read into the computer from any of the input devices. Also, the cathode ray tube provides immediate visual reference to the contents of whatever register is *selected* at the moment. Moreover, the machine can be set up in such a way that the program tape punch serves as an additional data tape punch. The output devices can provide permanent records of input data, intermediate results, and final results of computation.

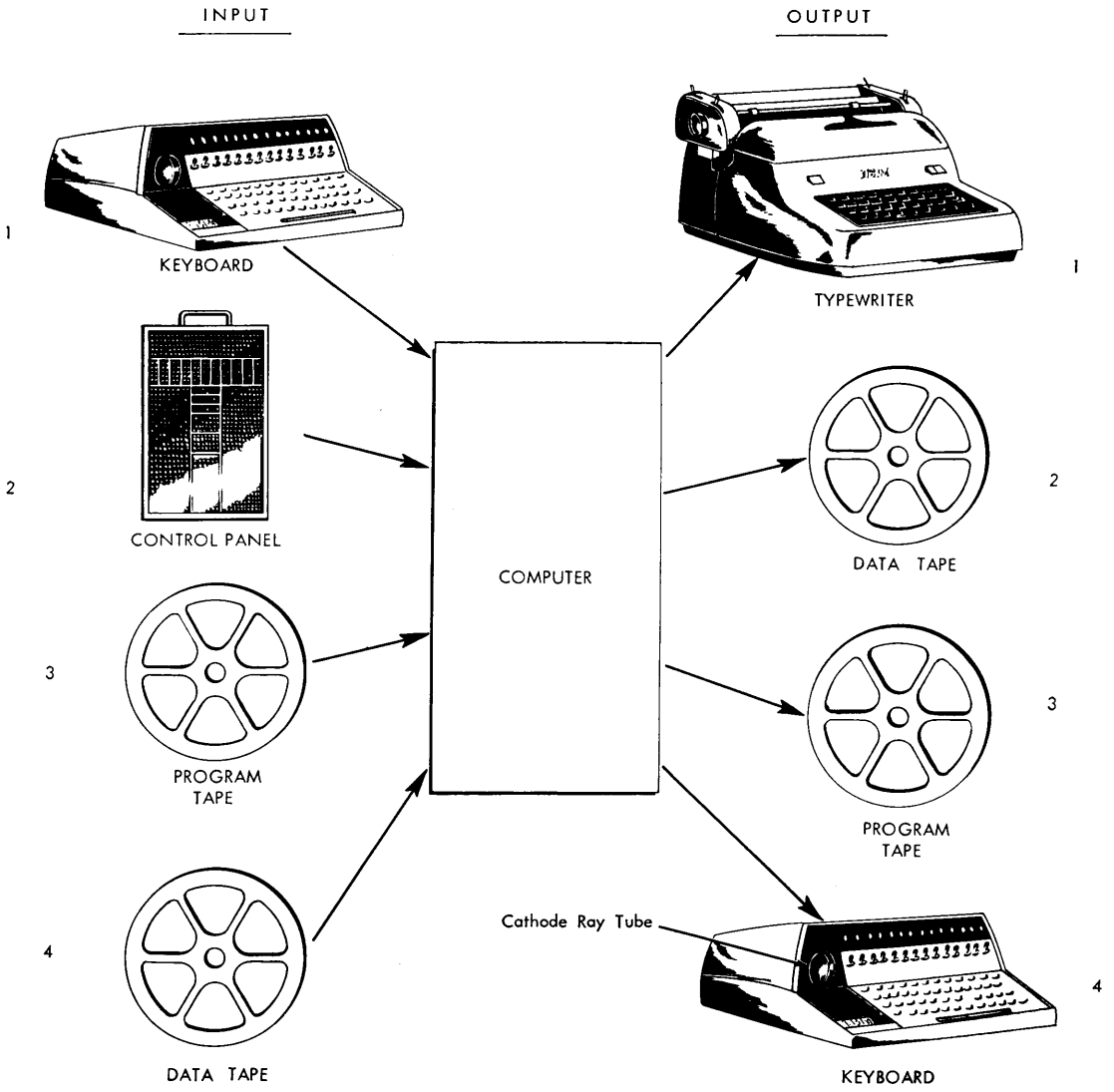


FIGURE 48. INPUT AND OUTPUT OF IBM 610

The data tape punch may also be used in conjunction with the data tape reader to supplement the internal (magnetic drum) storage of the computer by punching intermediate results on paper tape. This tape may be read by the data tape reader when needed, making the numbers stored accessible to the computer. *Both the program tape unit and the data tape unit must be loaded with tape.*

AUTOMATIC TYPEWRITER

WE HAVE discussed how to put data into the 610 and how to perform basic arithmetic operations on it. We now wish to extract the results of our calculations from the machine. In one-time calculations, the results are usually typewritten. To turn on the typewriter, we have the following instructions:

(Carriage Return)

The depression of this key tells the machine to turn on the typewriter, return the carriage, and space up in readiness to type the contents of the selected register. The left-hand margin must be manually set at the beginning of the operation to the position desired by the operator. The line-spacing device on the typewriter may be set to space 1, 2, or 3 lines as desired.

Having turned the typewriter on, the operator is now ready to read out data. This is done by means of read-out instruction.

(Tab)

The depression of this key causes the typewriter to be turned on and to move to the next tabular stop in readiness to type the contents of the selected register. Tabular stops are set at desired positions by the operator at the beginning of a calculation on the 610.

(Typewriter)

The depression of this key causes the typewriter to be turned on in readiness to type the contents of the selected register. If a number key is depressed while the key is on, that number will be typed. This instruction is used when it is not desired to move to the next tabular stop or carriage return.

In all three instructions above, Register Selection cannot be changed when the typewriter is in the ready state.

(Read Out)

The depression of this key causes the machine to read out the contents of the selected register to the selected output devices with the appropriate sign. It is well to recall here that even though a negative number is stored as the ten's complement in the machine, it is read out as a true number with a minus sign. *The RO operation reads out only the upper half of the register.* Therefore, to effect read-out, it is necessary to have the desired portion of the number in the upper half of the register. Thus, when operating in auto-point, SL instructions or a SL 15 instruction is given, to place the number in the upper half of the register, upon completion of read-out, the machine will *automatically* execute a SR 15 instruction to restore the contents of the selected register to the right-hand standard position. NOTE: This automatic re-alignment applies only while in the auto-point mode. At the conclusion of the read-out, the typewriter is turned off, and a second read-out, even though immediately following, must be preceded by an instruction to turn on the typewriter.

We shall use, as an example of read-out, the solution of the quadratic equation described earlier in the manual in the *Programming Example Section.* You may recall that we stopped our program with $x = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$ in the A register.

To read out the answers, we proceed as follows:

CR

turns on the typewriter and returns the carriage.

RO

reads out the value of x in the A register, standing selected.

15 TAB

selects register 15 and moves the carriage to the tabulator stop for the column used for the second value of x.

RO

causes the second value of x to be read out (Figure 49).

			CR		A	Turn on typewriter and return carriage	READ OUT OF ANSWERS
			RO		A	Read out $\frac{-b+\sqrt{b^2-4ac}}{2a}$ to typewriter	
		15	TAB			Select register 15, turn on typewriter and tabulate	
			RO			Read out $\frac{-b-\sqrt{b^2-4ac}}{2a}$ to typewriter	

FIGURE 49. PRINT OUT OF ANSWERS

The entire program is shown in Figure 50.

IBM 610 PROGRAMMING SHEET

Storage Usage

00	01 a	02 b	03 c	04	05	06	07	08	09
10	11	12	13	14	15	16	17	18	19
20 4	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39

Instr. No.	Cl.	First Reg.	Oper.	Second Reg.	Reg. Sel. After	REMARKS
		01	ENT			2.+
		02	ENT			11.-
		03	ENT			12.+
		20	ENT			4.+
		20	X	01	A	4a IN A
			X	03	A	4ac IN A
		11	COPY	A	11	4ac INTO 11
		02	X	02	A	b^2 IN A
			-	11	A	b^2-4ac IN A
			$\sqrt{}$		A	$\sqrt{b^2-4ac}$ IN A
		02	CNV		02	-b IN 02
		09	COPY	02	09	-b INTO 09
			+	A	09	$-b+\sqrt{b^2-4ac}$ IN 09
		02	-	A	02	$-b-\sqrt{b^2-4ac}$ IN 02
		01	+	01	01	2a IN 01
		02	÷	01	A	$\frac{-b-\sqrt{b^2-4ac}}{2a}$ IN A
		15	COPY	A	15	" INTO 15
		09	÷	01	A	$\frac{-b+\sqrt{b^2-4ac}}{2a}$ IN A
			CR		A	
			RO		A	Read out $\frac{-b+\sqrt{b^2-4ac}}{2a}$ to typewriter
		15	TAB		15	Select 15 and turn on and position typewriter
			RO			Read out $\frac{-b-\sqrt{b^2-4ac}}{2a}$ to typewriter

FIGURE 50. PROGRAMMING FOR SOLUTION OF QUADRATIC EQUATION

30	31		32	33	34	35	36	37	38	39
Instr. No.	Cl.	First Reg.	Oper.	Second Reg.	Reg. Sel. After	REMARKS				
		01	ENT		01	Enter N_1 into Register 01 and type it out				
			CR		01	on the typewriter.				
			KB		01					
		02	ENT		02	Enter N_2 into Register 02 and type it				
			TAB		02	out on the typewriter				
			KB		02					
		03	COPY	01	03	Place N_1 in register 03				
			+	02	03	Add N_2 to N_1 in register 03 (N_1+N_2)				
		01	÷ X	03 02	A	Multiply N_1 by N_2 and divide by N_1+N_2, result in A				
			TAB		A	Turn on typewriter and position				
			RO			Read out $\frac{N_1 N_2}{N_1+N_2}$ to typewriter				

FIGURE 51. PROGRAM PROVIDING FOR ENTRY OF SEVERAL SETS OF DATA

The typewriter (and the data tape punch) can record information during entry as well as read-out. Therefore, the operator can, if he wishes, prepare a more complete record of the calculation, with the input and results typed in columns on the same line, thus:

N_1	N_2	$(N_1N_2)/(N_1 + N_2)$
---	---	-----
---	---	-----
---	---	-----

The program for the typing of input and calculated results is shown in Figure 51.

PUNCHED TAPES

DTP (Data Tape Punch)

THE DEPRESSION of this key turns on the data tape punch and prepares it to receive information from the machine. Normally, this operation will be the second in a sequence of three operations, just as with the typewriter. The first will cause the appropriate register to be selected; the second will be DTP, which will turn on the data tape punch; and the third will be RO causing the information to be punched into the data tape. A special word-end symbol is automatically punched at the end of the number. The word-end symbol is used to terminate the reading of digits on read-back.

It is possible to read out to the typewriter and to the data punch simultaneously. To accomplish this, a register is selected, both output devices turned on, and a read-out (RO) command is given. In operational language we have:

15 TYP DTP RO
or 15 TAB DTP RO
or 15 CR DTP RO

During the processing of a problem in which portions of the data are entered and then subjected to lengthy calculation and processing, it may be far more convenient to enter all of the sets of data onto data tape and allow the program to proceed without operator intervention. Such an arrangement will allow the operator to key-in all sets of data at one time, set the program in motion, and check the machine periodically while he is doing other work.

In addition, the data tape may be used as intermediate storage for partial results to be used later in the calculation. Such usage requires careful planning of the arrangement of data as it goes on the tape so that it will be available in the order in which it is needed.

Instead of keying-in the numerical data as needed, it may be recorded in advance on the data tape. The machine can be programmed in such a way that control will go to the tape reader rather than to the keyboard when additional data are required, and will go automatically back to the program reader when the numerical entry is completed. As an example, let us call the following pairs of numbers our data:

N_{11}, N_{21}
N_{12}, N_{22}
N_{13}, N_{23}, etc.

These will be recorded on the data tape as shown on the next page.

This tape might have been prepared by giving the following commands from the manual keyboard:

A ENT DTP (Digits of N_{11} and sign)
ENT DTP (Digits of N_{21} and sign)

and so on. Any available register may be used in place of the A register. Until the machine is told otherwise, the register used for entry remains selected.

If the numbers N_{11}, N_{21}, N_{12}, --- are themselves the results of a previous calculation, they could have been read out to the data tape at the time they were developed.

There are three circumstances under which data can be punched on data tape: first, during entry; second, during read-out; third, when the data punch is duplicating symbols read by the data tape reader (under control-panel direction).

By control-panel wiring, the reading of the word-end symbol by the data reader can be caused to transfer control automatically to any other control agency, such as the keyboard or program tape.

To carry out the simple calculation introduced in the program tape discussion — namely, evaluating $(N_{11} N_{21})/(N_{11} + N_{21})$, given N_{11} and N_{21}, the operator proceeds as follows:

1. Wire the control panel so that reading of the word-end symbol by the data tape reader transfers control to the keyboard (details of control-panel wiring in the control-panel section).
2. Turn on PUNCH switch.
3. Key the following instructions to read-in data from data tape already prepared:

01 ENT CR DTR
02 ENT TAB DTR

As a result of these two lines of instructions, the operator has entered N_{11} and N_{21} from the data tape into registers 01 and 02 and typed that data. Now key:

03 COPY 01
+ 02
01 ÷× 03 02
TAB RO

4. Change the control-panel wiring in such a way that reading of the word-end symbol will transfer control to the program tape reader (details of control-panel wiring in control-panel section).
5. Since the program is now punched on the program tape and all the data are ready on the tape, the calculation can proceed completely automatically. It is initiated by turning off the PUNCH switch, turning on the DUP switch, and depressing the PTR key.

In addition to digits, signs and word-end symbols, special symbols can be punched into the data tape to mark the end of a group of data. The sequence symbols (e.g., SEQ 11) are used for this purpose. These group-end symbols can then be used, for example, to conclude one automatic program and initiate another automatic program.

← DIRECTION OF TAPE MOVEMENT

.....	N_{11}	Sign of N_{11}	Word End Symbol	N_{21}	Sign of N_{21}	Word End Symbol	N_{12}	Sign of N_{12}	Word End Symbol	N_{22}	Sign of N_{22}	Word End Symbol	N_{13}	.....

Data Tape Correction

Any symbol punched on the data tape may be deleted by manually positioning the tape, turning on the data tape punch by depressing the DTP key, and depressing the DEL key on the keyboard. Depressing the REL key will turn off the punch. When the data punch is duplicating symbols read by the data reader, it will not duplicate the deleted characters that now have all 6 channels punched.

Tape Word-End and Group-End Symbols

Eight-Channel

At the end of each word placed on the data tape, the machine automatically inserts a word-end symbol. This tape symbol is actually a combination of two symbols regularly used. They are the symbols for SEQ and TAB.

To end a group of data, the SEQUENCE instruction is used with an associated sequence number. It is not possible to punch the SEQUENCE symbol into the data tape directly and the control panel must be used to accomplish this. The program to be wired into the control panel is:

```
   DTP   (turn on Data Punch)
PCH SEQ
     0
     1
OUT OFF   (turn off Data Punch)
```

A full description of the control-panel wiring is given in the control-panel section. The eight-channel tape is shown in Figure 52.

Five-Channel

At the end of each word on five-channel data tape, the machine automatically inserts a word-end symbol. The word-end symbol is the same as in eight-channel tape except that an additional ENT command occurs as part of the five-channel word-end symbol. In five-channel operation, three non-numerical symbols are needed for format control. If the machine is punching five-channel tape, when the data tape punch is turned on by the DTP command, an ENT symbol is automatically punched into the data tape. Thus, the numbers on five-channel data tape are separated by the tape symbols for SEQ, TAB and ENT (three non-numerical symbols). The ENT symbol is not recognized by the 610 when five-channel tape is read.

It is recommended that any program processing data from five-channel tape include a checking routine to insure accuracy of the data being processed.

In order to end a group of data on five-channel tape, a slightly more complex group end symbol is necessary. The control panel must be used to get this group end symbol into the data tape. The program to be wired into the control panel is:

```
   DTP (turn on Data Punch)
PCH SEQ
    ENT
    ENT
     0
     1
PCH SEQ
    TAB
OUT OFF
```

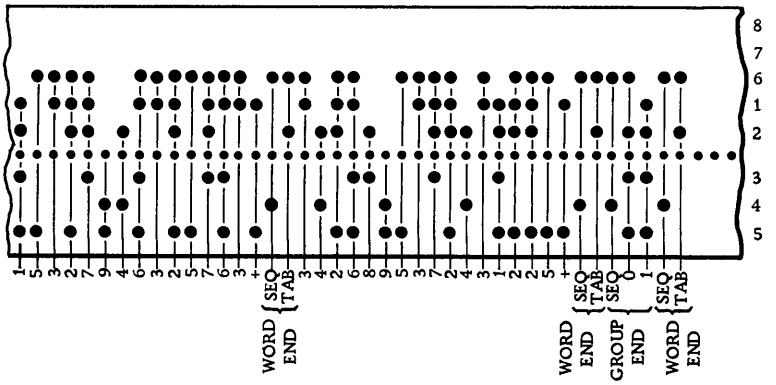


FIGURE 52. WORD AND GROUP-END SYMBOLS IN EIGHT-CHANNEL TAPE

This is clarified in the Control Panel Section. The five-channel tape is shown in Figure 53.

CAUTION: *For efficient and accurate operation of tape units, tape of desired width must be in both the read and the punch mechanisms of both the program and the data tape units.*

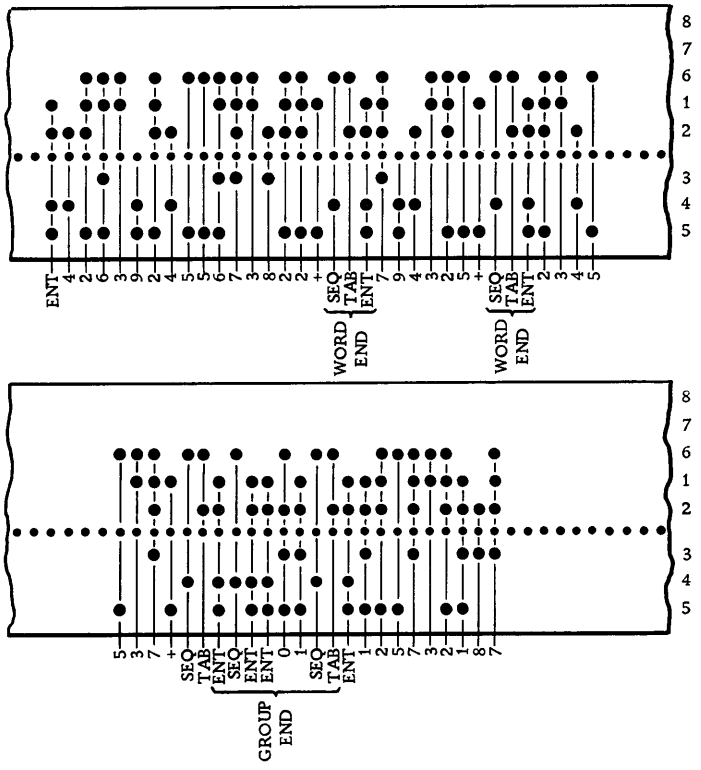


FIGURE 53. WORD AND GROUP-END SYMBOLS IN FIVE-CHANNEL TAPE

Tape Codes Related to Teletype Operation

The tape code for digits used in the IBM 610 is the same as the tape code commonly used in teletype transmissions. In the machine, special meanings have been assigned to the rest of the codes. The tape symbol equivalences for some of these codes are as follows:

1. Tab = Line feed in the standard teletype code
2. Sequence = Carriage return in the standard teletype code
3. Enter = Figure shift in standard teletype code.

There are some tape codes that do not correspond among various teletype services. The minus (−) sign code is a minus sign on all teletype machines except the weather machines. The plus (+) sign code is a plus on weather machines but quotation marks on all other machines. Blank tape is a decimal point on commercial machines and a minus (−) sign on weather machines.

As a result of the interrelation between Sequence and line feed, Tab and carriage return, and Enter and figure shift, the word-end symbol when the machine is punching eight-channel tape at the data punch is the combination Sequence, Tab. When the five channel tape is being punched, the word-end symbol becomes Sequence, Tab, Enter. In teletype equivalent code the word-end symbol is carriage return, line feed and figure shift. This latter combination is exactly what is needed between numbers being transmitted via teletype.

Tape Usage Suggestions

If possible, a keyboard (KB) instruction should be placed at the beginning of the program in an unused class. Then, if the operator wishes to locate the beginning of the program for any reason, he need only set the class response switch of the class of the KB instruction to the ON position, turn all other class response switches off, and depress the PTR key. The machine will then ignore all instructions (except to duplicate them back into the program tape if the DUP switch is on) until it reaches the KB instruction at the beginning of the program, when it will stop with control at the keyboard. This procedure will often be useful in correcting program errors that can only be studied or repaired by starting the calculation over.

To study a tape program that is failing for some reason, it may be useful to type the program from the tape. To do this, the operator turns on the DUP and TYP PROG switches, and depresses the PTR key. The program will then type out. The operator is responsible for advancing the carriage during this process by stopping the program type-out as it nears the end of a typed line. This is done by depressing the INT key, depressing the carriage return on the typewriter, and then depressing the RSM key to resume program type-out. The program may now be typed out and examined for errors or compared to the original written program.

Suppose the error is an incorrect register number, 21 instead of 32.

The operator proceeds as follows:

1. The DUP and TYP PROG switches are turned on, and the program is typed out. As the operator nears the point in the program to be altered, he holds down the INT key and steps the program along by depression of the RSM key.

2. The incorrect register number is typed out and the operator then backspaces the typewriter and the tape at the punch station.

3. The operator depresses the DELETE key twice to delete the incorrect address. The holes in those tape columns will all be punched out and the characters previously typed out will have slash (/) marks through them.

4. The operator sets the TYP PROG and DUP switches to the OFF position and turns on the PUNCH switch.

5. The operator next depresses the correct address keys, then turns the punch switch off and the DUP switch on.

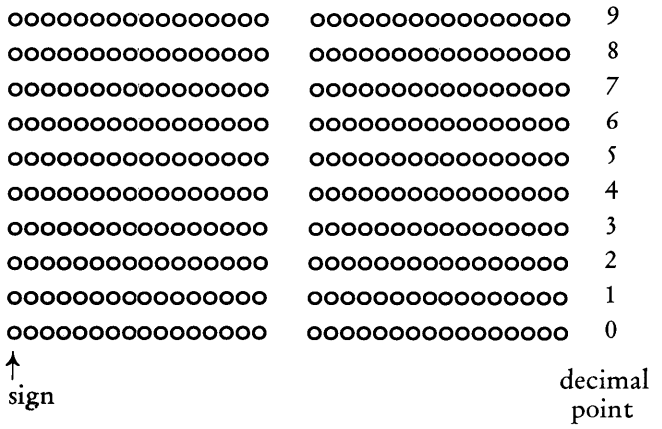
6. The operator sets the class response switches to the OFF position except for the one class of the KB instruction at the beginning of the program.

7. Finally the PTR key is depressed and the program is duplicated correctly but not typed.

Although it is possible to execute a problem containing control transfers while preparing a program tape, the operator may find it simpler to turn the COMPUTE SUPPRESS switch ON. If this is not done, the operator must remember that depressing PTR actually transfers control to the program tape which probably does not contain a program at this point. A SEQ instruction to the control panel, DTR or CP instruction can also cause undesirable results. If an operator uses DTR with data on the data tape, he might temporarily wire the WORD END hub to KB. The operator knowing what transfers his program contains, must decide whether to execute a problem while preparing the program tape or to type out his program for checking purposes and do his first compute run after the tape is prepared.

CATHODE RAY TUBE

ON THE keyboard of the IBM 610, there is a two-inch cathode ray tube for displaying the contents of the register which stands *selected* at any time during the operation of the machine. The face of the tube looks like this:



There are two rectangular arrays, each sixteen points wide by ten points high. Reading from top to bottom, the rows correspond to the digits from 9 through 0. The first column on the left will have a 0 spot on the face of the tube if the number being displayed is positive, and a 9 spot if the number being displayed on the face of the tube is negative.

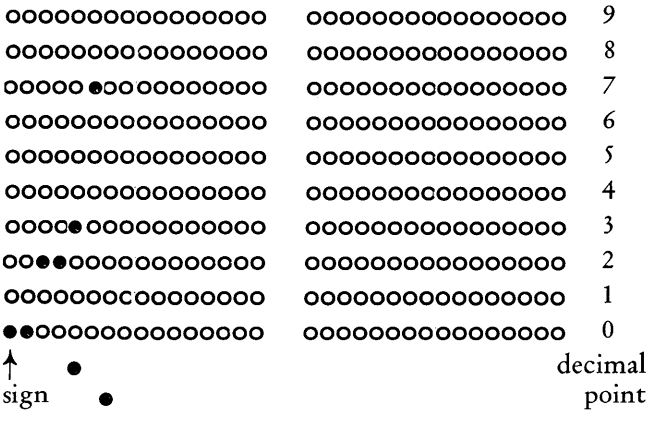


FIGURE 54. FACE OF CATHODE RAY DISPLAY TUBE

The decimal point will appear as a bright spot below the digit column immediately following the point. During entry an additional spot (called the tag) will appear at the lower edge of the tube face below the digit column that is ready to receive a digit. Occasionally, the operator will see a spot near the top edge of the tube; this spot is used for internal control purposes and need not concern the operator.

Figure 54 shows a partially entered number as it would appear on the cathode ray tube display. The number so far entered in the figure is 022.37. A bright spot below the digit 3 in the fifth position from the left indicates that the decimal point immediately precedes the 3. A bright spot near the bottom of the display in the seventh position from the left indicates that that position of the register is ready to receive a digit. The sign of the number will be entered into the machine after all of the digits up to fifteen have been entered and will appear at the far left of the display.

Negative numbers are displayed in tens complement form; e.g., −0.892 is displayed as

sign
↓
9999999999999999.1080000000000000

When the machine is operating in the auto-point mode, the number entered is automatically positioned as seen on the cathode ray tube so that all digits to the left of the decimal point will appear in the left half of the tube and all of the digits to the right of the point will appear in the right half of the tube. Thus, the point will lie below the first column of the right-hand array.

The cathode ray tube is an indication of the existence of data in the selected register and is not recommended for the reading of the data in the selected register. If a person wishes to examine in detail the contents of the selected register, he need only read out to the typewriter.

OTHER MODES OF OPERATION

OCTAL-DECIMAL

THE operation specifies a choice between octal or decimal mode via the control panel or keyboard switch. If the octal mode is selected, data are introduced in octal form and thereafter all calculation takes place in the octal mode with no special demands upon the operator. Programs can be initially run and tested in decimal and subsequently in octal giving the operator results to be used in conjunction with programming and debugging binary computers.

FIXED-POINT

FOR general work, the operator will use the auto-point mode almost exclusively, because it eliminates the necessity for the scaling work associated with the fixed-point mode. The flexibility of a choice is provided, however, for those occasions when an operator feels fixed-point operation is advantageous.

When any machine is operating in the fixed-point mode, the operator must take care to position the numbers correctly in the registers to insure that the machine is really performing the operations he intends it to do. At all times the operator must take into account the magnitudes of the input numbers and the magnitudes of all intermediate numbers to insure against overflow and to insure correct alignment of decimal points in all operations.

The operator may or may not choose to include decimal points in the numbers he enters in the various registers. Whether he does or does not physically enter decimal points in the numbers, he must be aware of decimal placement at all times.

To point out the common errors encountered in the fixed-point mode, let us consider some examples.

If one wishes to add six to fifty-four, using paper and pencil, one is careful to write

$$\begin{array}{r} 54 \\ 6 \\ \hline 60 \end{array} \quad \text{not} \quad \begin{array}{r} 54 \\ 6 \\ \hline 114 \end{array}$$

Correct positioning is essential to correct solution.

An operator must also be careful to avoid overflow. For example, if he is adding a series of pairs of two digit numbers, it would not be wise to position them at the very left-hand or high-order positions of the registers, because

$$\begin{array}{r} +21 \\ +32 \\ \hline +53 \end{array} \text{ would work, but } \begin{array}{r} +65 \\ +84 \\ \hline \text{----} \end{array} \text{ would lead to overflow.}$$

A flashing overflow light on the keyboard indicates this condition. It might, therefore, be wise to position all of the two-digit numbers in the register in this manner:

$$+065$$

In addition or subtraction, when decimal points are entered with the numbers, the points must, of course, be lined up. For example, to subtract 2.2 from 24.2

$$\begin{array}{rr} & +024.2 \\ (-) & +002.2 \\ \hline & +022.0 \end{array} \quad \text{not} \quad \begin{array}{rr} & +024.2 \\ (-) & +02.20 \\ \hline & +00.22 \end{array}$$

If the decimal points are not aligned, the point of the second register dominates. If we are adding the contents of register 15 to register 10 and we have:

$$\begin{array}{rrr} & \text{Register 10} & +04.610 \\ & \text{Register 15} \quad (+) & +021.40 \\ \hline \text{then the result in register 10 would be} & & +067.50 \end{array}$$

To insert decimal points in numbers already standing in registers, we take advantage of the *dominant-point* idea. If we have register 16 containing +0125761 and we wish to insert a decimal point between the 5 and 7, we simply place +000.000 in a storage register (say 30) and then instruct the machine as follows:

$$16 + 30$$

The result is that register 16 contains +0125.761. In the same manner, we may remove decimal points from numbers standing in a register. If we leave +04.71 standing in register 10 and we know register 19 has been cleared and contains +0000, we instruct the machine

10 + 19

The result will appear in register 10 as +0471 with no decimal point.

To clarify the behavior of the decimal point in multiplication, division, division-multiplication and square root, we introduce the term *decimal index*. The decimal index of a number stored in a register is the *number of places* to the left of the real or visualized position of the decimal point in that register. The sign position is not counted. For example, the number +27.81 has a decimal index of two, and the number +0045.72 has a decimal index of four. To generalize, the decimal index for a number x in a register will be denoted by D_X.

Even though both factors in multiplication and division may contain decimal points; though all three factors in division-multiplication may have decimal points; and though the single number having its square root taken may have a decimal point, *no decimal point will appear in the A register at the end of any of these operations.*

The operator may position a decimal point in the A register at the conclusion of the operation by using the technique of inserting decimals just discussed. The operator may wait to insert the decimal point until a more desirable time, but he must mentally keep track of the correct position of the point.

If we multiply the contents of registers X and Z, the rule governing the location of the point in the product (product in the A register) is:

$$D_A = D_X + D_Z$$

Thus, for example:

X		Z		A
+02.0	times	−03.01	equals	−0006020

After correct positioning of the decimal point, A contains −0006.020.

Another example with visualized points is:

X		Z		A
−090	times	− 060	equals	+005400

In these illustrations, it is well to remind the reader that each register contains 31 digits or zeros and in each case the right-hand zeros were omitted in order to highlight the important happenings in the high-order positions. Overflow when taking the product of two numbers is not possible.

In the process of division on the IBM 610, the dividend and divisor must fulfill the condition generally called the *magnitude rule.*

The absolute value of the contents of the divisor register must be greater than the absolute value of the contents of the dividend register without regard to the placement (real or visualized) of the point.

For example:

+3763.00 can be divided by
+750.100

because they appear in the registers as:

37630 00000 00000 00000 00000 000000
75010 00000 00000 00000 00000 000000

and, therefore, fulfill the above condition.

On the other hand,

+3763.00 cannot be divided by
+16270.0

because they appear in the registers as:

37630 00000 00000 00000 00000 000000
16270 00000 00000 00000 00000 000000

and the *magnitude* condition is not fulfilled. Overflow will result from this division and will be indicated.

In order to make the last division possible, we shift the dividend one place to the right and have

03763 00000 00000 00000 00000 000000

Now the magnitude rule is fulfilled and division may proceed.

If we divide the contents of register X by the contents of register Y, the rule for decimal indices governing the location of the point in the quotient (quotient in register A) is:

$$D_A = D_X - D_Y$$

For example:

X		Y		A
+27.0	divided by	+5.0	equals	+54

When the decimal point is correctly positioned, A contains +5.4.

If we perform a divide-multiply operation; namely, if we divide the contents of register X by the con-

tents of register Y and, in the same operation, multiply the quotient by the contents of register Z – the operation is $\frac{X}{Y}Z$ the rule governing the location of the decimal point in the result (result in A register) is:

$$D_A = D_X - D_Y + D_Z$$

For example:

X Y Z A

+16.0 divided by −4.0 times +3.0 equals −120

When the decimal point is correctly positioned, A contains −12.0.

To avoid excessive calculation time and overflow checks, follow the magnitude rule applied to X/Y.

If we take the square root of the contents of register X, the location of the decimal point in the answer (answer in the A register) is given by the following rule:

$$D_A = \tfrac{1}{2} D_X$$

Because decimal indices must always be whole numbers, this rule implies that D_X must be an even number.

For example:

X A

The square root of +25.0 is +5.0

To take the square root of +160.0 is not permissible, and the machine acts as though the number is 16.00. Instead, it should be positioned as: +0160.0, and the correct answer will be obtained.

If we try to take the square root of a negative number, the machine ignores the sign and behaves as though the number were positive.

It should also be noted that the square root of +16.00, +1600.0, and +160000.0 appear as the same number in the A register. Only operator knowledge of the problem enables him to position the decimal point in the correct place in the A register.

NOTE: All of the above discussion applies to the control of the octal point when the machine is operating in the octal mode.

Operators should be cautioned that intermediate results may be sufficiently large to cause overflow even though the input data is numerically small. It is well, therefore, to pay careful attention during all stages of a problem.

Operators should also note that the first set of data is often chosen so that the order of magnitude of the calculations will be a good indicator of their correctness. Subsequent sets of data may not behave as well at intermediate points in the calculation. Care must be exercised in positioning numbers so that overflow, division by zero, and like occurrences do not lead to the belief that the machine is not operating correctly.

All of these considerations for fixed-point calculations are taken care of by the machine when operating in auto-point.

SUMMARY OF OPERATION

THE FOLLOWING chart is a summary of all the operations covered in detail throughout this manual.

KEYBOARD SYMBOL	TYPE PROGRAM SYMBOL	OPERATIONS, SIGNS AND DIGITS
0	0	Zero
1	1	One
2	2	Two
3	3	Three
4	4	Four
5	5	Five
6	6	Six
7	7	Seven
8	8	Eight
9	9	Nine
.	.	Point
SEQ	S	Sequence
TAB	T	Tabulate
ENT	E	Enter
−	−	Minus
+	+	Plus
$\sqrt{}$	$\sqrt{}$	Square Root
×	×	Multiply
÷	÷	Divide
SR	R	Shift Right
SL	L	Shift Left
CLR	C	Clear
DTP	P	Data Punch
DTR	D	Data Reader
DEL	/	Delete
KB	K	Keyboard
RO	W	Read Out
COPY	Y	Copy
CP	B	Control Panel
A	A	Answer Register
÷×	*÷	Divide-Multiply
CNV	* −	Convert
TYP	* Y	Typewriter
SR 15	* R	Shift Right 15
SL 15	* L	Shift Left 15
CLR RH	* C	Clear Right Half
CR	* T	Carriage Return
REL	* W	Release
PTR	* B	Program Reader

* *Indicates a doublet (2 character) tape code.*

MANUAL KEYBOARD

FIGURE 55 is a sketch of the IBM 660 keyboard. The keyboard has the following features:

43 keys	11 lights
14 switches	1 cathode ray display tube

In this section, we shall review the functions of the keys, switches, and lights.

KEYS

Operation Keys

Briefly summarized, the functions of these keys are:

ENT prepares the machine to enter data into a register from an input device.

+ (Plus) prepares the machine to add to the contents of the selected register the contents of the next addressed register. It is also used to terminate the entry of a positive number.

− *(Minus)* prepares the machine to subtract from the contents of the selected register the contents of the next addressed register. It is used to terminate the entry of a negative number.

× *(Multiply)* prepares the machine to multiply the contents of the selected register by the contents of the next addressed register and place the product in the A register.

÷ *(Divide)* prepares the machine to divide the contents of the selected register by the contents of the next addressed register and place the quotient in the A register.

÷× *(Divide Times)* prepares the machine to divide the contents of the selected register by the contents of the next addressed register and multiply the quotient by the third addressed register. The answer is placed in the A register.

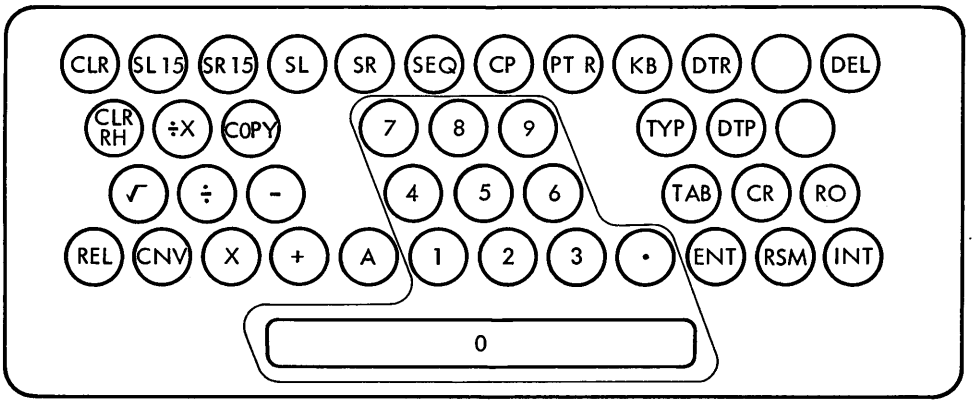


FIGURE 55. KEYBOARD OF IBM 610

$\sqrt{\ }$ *(Square Root)* takes the square root of the contents of the selected register and places the answer in the A register.

CNV (Convert) changes the sign of the number in the selected register.

CLR (Clear) changes the contents of the selected register to all zeros.

COPY (Copy) clears the contents of the selected register and replaces them by the contents of the next addressed register.

CLR RH (Clear Right Half) changes the contents of the right (lower) half of the selected register to zeros.

SL (Shift Left) causes the contents of the selected register to be shifted to the left one place.

SR (Shift Right) causes the contents of the selected register to be shifted one place to the right.

SL 15 (Shift Left 15). This operation has two different effects depending upon whether the machine is in the auto-point mode or the fixed-point mode. In the auto-point mode, the contents of the selected register is shifted to the left:

a. Until a non-zero number is in the high-order position of the upper half of the selected register,

(or)

b. Until the position containing the decimal point is shifted next to the sign position of the selected register (left-hand standard),

(or)

c. Until fifteen shifts to the left have taken place in the selected register.

In the fixed-point mode, the machine always shifts the contents of the selected register fifteen positions to the left.

SR 15 (Shift Right 15). This operation has two effects depending upon whether the machine is in the auto-point mode or the fixed-point mode. In the auto-point mode, the contents of the selected register is shifted to the right:

a. Until the decimal point is in the high-order position of the lower half of the selected register (right-hand standard),

(or)

b. No effect if the decimal point is in the lower half of the selected register,

(or)

c. Until fifteen shifts to the right have taken place in the selected register.

In the fixed-point mode, the machine always shifts the contents of the selected register fifteen places to the right.

Release Key

The REL command drops out any operation key previously depressed, releases the selected register and makes *A* the selected register, all output devices that are ON are turned OFF. In the event that the operator has selected an incorrect register (partially or completely), the REL key will release it and select the A register. All operations will be released by depressing this key.

Number Keys

The number key section of the keyboard is distinguished from the other keys by distinctively colored keys. The keys are 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, (.). These keys have three functions:

1. Register selection
2. Digit entry
3. Sequence selection (covered in the discussion of the SEQ instruction).

A Key

The *A* key has only one function, that of register addressing.

Control Keys

The following keys transfer control of the computer to the corresponding control agencies, if depressed when the keyboard is in control:

KB	(keyboard)
CP	(control panel)
PTR	(program tape reader)
DTR	(data tape reader)

If the program being written contains control transfers from the keyboard, the program should be punched into the program tape with the COMPUTE SUPPRESS switch in the ON position. Once the program has been placed on the program tape and the program tape is in control of the machine, control will be transferred among the control units as indicated by the program.

Typewriter Keys

TYP turns on the typewriter, and the typewriter is ready to type beginning at the position above the ribbon guide.

CR turns on the typewriter and returns the carriage to the left margin.

TAB turns on the typewriter and moves the carriage to the next tabulator stop.

Data Tape Punch Key

DTP turns the data tape punch on.

Read-Out Key

RO causes the read-out of data from the selected register to the selected output devices. Previous to giving the read-out command, the operator must select the register, select the output device or devices, and align the data to be read out in the left half of the register. If any left shift was executed preceding RO, the contents of the register read out will be returned to the right-hand standard position.

Interrupt Key

Depression of the INT key causes interruption of a program that is proceeding automatically under control of the program tape or control panel. When the INT key is depressed, the OPER light may go on, indicating the machine is in the middle of an operation that must be completed before any other key may be depressed. Before touching any other key on the keyboard, the OPER light must be turned off by holding the INT key down and depressing the RSM key step by step until the light goes out.

Resume Key

Depression of the RSM key causes resumption of an automatic program. It is possible to step through an automatic program instruction by instruction, as described in the last paragraph, by holding down the INT key and depressing the RSM key once for each step. If an operation code is a two-character code, the RSM key must be depressed twice for the operation.

Delete Key

Erroneous punching on the data or program tape can be deleted by manually positioning the tape under the punch head, turning on the punch, and depressing the DEL key.

NOTE: *Whenever keys have been depressed so that calculation is taking place within the machine, the keyboard is interlocked and any further key depressions will be ignored until the calculation is completed. If an instruction key is depressed during calculation, it must be depressed again at completion of calculation in order to be effective. An instruction key is effective only when the keyboard light is on.*

SWITCHES

Punch Switch

This is a two-position switch. When on, it causes the program tape to punch a code symbol corresponding to each key depressed on the keyboard. Exception: punching ENT KB in sequence suppresses punching of numbers until the numerical entry is terminated by depressing a sign key (+ or −). It is possible to type out at the same time that data which is being entered. The instructions would be 01 ENT CR KB.

Dup Switch

This is a three-position switch. In the ON position, it causes the program tape punch to duplicate any symbol read by the program tape reader. *All characters read by the reader are duplicated* without regard to whether the computer is responding to that class of operation or not. Duplication may be stopped by placing the switch in the OFF position. Duplication from the program tape reader to the program tape punch may also be accomplished by control panel wiring when the DUP switch is in control-panel position.

Punch Class 1, 2, 3 Switches

These are three 2-position switches. These switches cause a class mark of the appropriate switch to be punched into the program tape when on. If all of the switches are in the OFF position, instructions are punched in class 0. When any one of the switches is on, the PUNCH CLASS SWITCH light is lighted.

Response Class 0, 1, 2, 3 Switches

Four 3-position switches (ON-OFF-CONTROL PANEL). If one of these switches is on, the machine will respond to instructions having the corresponding marks in channels 7 and/or 8 of the program tape. If a switch is in the OFF position, the machine will not respond to a corresponding class mark. If these switches are set to the control-panel position, then control-panel wiring will determine which class the 610 will respond to. It is possible for the machine to respond to any one class, any combination of two classes, any combination of three classes, or all four classes. Hence, the switches may be set ON or wired ON in any combination.

It is strongly recommended that every program has as its first operation a KB operation in a response class not used in the program at any other time. When operating in this way, the operator may find an error in the middle of the program, set all response class switches off except this one, and the program tape will quickly pass through the machine. The program has been duplicated into the program tape for further use. This KB command is particularly helpful in quickly locating the beginning of a program.

At least one of the response class switches in one of the classes used must be on at all times. Usually class 0 is left on.

Compute Suppress Switch

This is a two-position switch, which, when in the ON position, suspends the operations of the computer during program tape punching from the keyboard.

Type Prog Switch

This switch, when on, suppresses computation and causes the typewriter to interpret the program tape by typing out a listing, in code, of the instructions as they are read out or punched. For a key to the code as it is typed out, see the Tape Code Section. Turning on this switch and giving the instruction RSM initiates this process; depressing the INT key terminates the process. During the type-out of the program, the operator must depress the INT key near the end of each line, return the carriage, and depress the RSM key to resume the typing. When the type-out of the program is completed, the operator must turn off the switch before returning control to the keyboard.

Fixed-Point Switch

When this three-position switch is in the ON position, the machine operates in the fixed-point mode. When this switch is in the control-panel position, the 610 may be placed in fixed-point mode by control panel wiring. *When this switch is in the* OFF *position, the machine is in the auto-point mode.*

Octal Switch

This is a three-position switch. When the switch is ON, the machine operates in octal arithmetic. In the control-panel position, the control-panel wiring determines whether the machine is in octal arithmetic or not. When the switch is OFF, *the machine operates in the decimal mode.*

Master Stop Switch

This switch is a spring-acting switch, which, when flipped, restores the computer to the keyboard control in case of faulty programming (e.g., dividing by zero). All operations or instructions are released and the A register is selected. Operation of the master stop switch may cause a digit-check error and should be used only in an emergency.

CHECKING LIGHTS (Flashing Lights)

When a checking light comes on, the RSM key must be used to get back into the program. The program must be started at the beginning.

Punch Check Light

The PUNCH CHECK light goes on when an odd number of punch knives have been driven through the paper tape giving a non-permissible character in a tape.

Reg Sel Light

The REGISTER SELECTION light goes on when an invalid address is given for a register and an operation key is depressed. If the operator does not complete the two-digit address of a register but punches one digit and an operation key, the REG SEL light will go on. If the machine does not select a valid register, the light goes on.

Digit Entry Light

The machine checks the digit code relays; and if there is not a permissible pattern, the light comes on, telling the operator the digit entered is invalid. Any time the machine is in the enter state (following an ENT command), the light comes on when an invalid number is entered.

Redund Light

This light comes on when invalid information is given to the computer. If an invalid character is created within the computer, any attempt to use the character will turn the light on.

Clear Light

If the *clear* operation fails, this light comes on. Clearing is accomplished by subtracting the contents of a register from itself, and zeros should result. The CLR, CPY and ENT operations apply to the selected register. The operations $\times$, $\div$, $\div\times$, and $\sqrt{\ }$ require the clearing of the A register.

Digit Check Light

This light comes on if the signals or digits to or from the drum are not a valid combination or code. The program must be started over if the light comes on.

Overflow Light

This light comes on if a number exceeding allowable magnitude is developed in the selected register. This light comes on when division by zero is attempted or if the numbers exceed the machine capacity *when the machine is in auto-point mode.* The numbers in the original registers will usually be changed in the case of overflow or exceeding capacity, but the contents of the A register are cleared.

KB Light

When this light is on, it indicates that the computer is under control of the keyboard.

Punch Class Switch Light (Flashing)

This light is flashing whenever one of the punch class switches is on. Its primary purpose is to warn the operator that a program is not being punched in class 0.

Operation Light

This light indicates that the machine was in the process of an operation when interrupted in automatic program. To turn this light off, hold down the INT key and repeatedly depress the RSM key until this light goes off. *This light must be turned off before further use of the keyboard, and no other keyboard keys should be touched until it is off.*

Int Light

This light goes on to indicate the interruption of a program.

DISPLAY TUBE

The display tube indicates the contents of the register currently selected. The code for the tube is discussed in *Cathode Ray Tube Section.*

COMPUTER LIGHT PANEL

ON THE computer proper, above the control panel, is a light panel, Figure 56.

At the bottom left of the panel is the ON-OFF switch. When this switch is turned on, power is supplied to the machine. The light adjacent to the switch is lighted whenever the switch is in the ON position.

In the upper left of the panel is a STANDBY-OPERATE switch. When this switch is in the OPERATE position and the light above the switch is on, the machine is ready for use. The light comes on when the electronic tubes in the machine are warmed and the magnetic drum attains its operating speed. When the switch is in STANDBY position and the ON-OFF switch is on, the drum speed is maintained and the electronic tubes remain heated, but the high voltage is off. When the switch is put into OPERATE from STANDBY, the machine is ready to operate immediately.

Under the heading SELECTED REGISTER on the control panel are two columns of illuminated numbers. The row on the left records the tens digit of the selected register. The row on the right records the units digit of the selected register. For instance, if the 2 in the tens row and the 4 in the units row are illuminated, register 24 is selected. If the A register is the selected register, the letter A between the two columns will be illuminated.

When registers MP, DIV and MC are addressed as 0., 1., and 3., the decimal character (above the A) will be illuminated.

The column of lights under the heading SELECTORS indicates those selectors currently in a transferred condition. If 4 and 7 are illuminated, the C and T hubs of selectors 4 and 7 are connected. All other selectors have their C and N hubs connected.

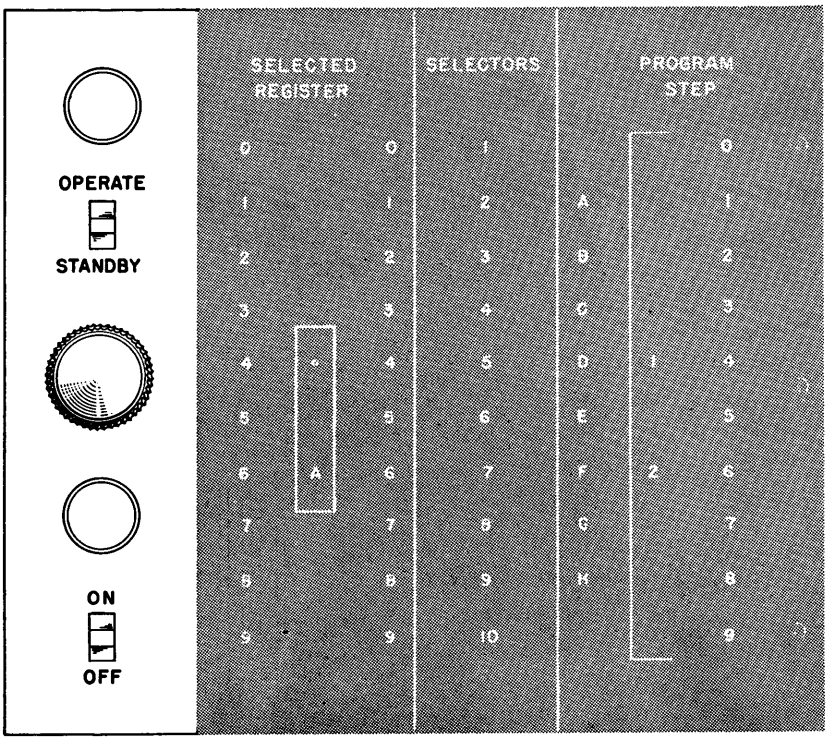


FIGURE 56. COMPUTER LIGHT PANEL

The PROGRAM STEP lights indicate the next program step to be executed. The left-most column indicates the letter prefix of the next program step. The next column consisting of three digits (0, 1, and 2) indicates the tens digit of the next program step. The right-most column of lights indicates the units digit of the next program step. For instance, if A, 1, and 8 are illuminated, the next program step to be executed is A-18. If C is illuminated, zero in the tens column is illuminated, and 3 in the units is illuminated, program step C-03 is the next step to be executed.

CHECKING FEATURES

IN designing the IBM 610 computer, reliability as well as ease of use was a prime consideration.

We have explained those checking features that are brought to the attention of the operator. These checking features plus the inherent conservative design of the computer make for an unusually reliable machine.

In addition to these checking features, listed below are the *self-checking features within the machine:*

1. Each time a character is punched into, or read from, the paper tape, the character is checked to be certain it is represented by a valid code.

2. Whenever numbers are moved within the 610, all digits are checked for valid coding.

3. Arithmetic operations are checked to verify that the sum, product, or quotient does not exceed the maximum allowable quantity.

4. Many arithmetic operations require *clearing* of a register. Each clearing operation is checked to insure that every position of the register is set to zero and that the arithmetic unit is functioning properly.

5. Every time the electronic computer operates, a check is made to insure that the attention of the machine is directed to a permissible register.

PROGRAMMING THE 610

TIMING

THE IBM 610 has been designed with ease of programming as one of the primary considerations. As a result, the programmer is not burdened with timing restrictions that must be considered throughout the planning of a program. Interlocks within the machine make it impossible to violate timing conditions and to cause the machine to give erroneous results.

The basic time element of interest to the programmer is the time required to read one tape character. The read portion of a tape unit reads characters at the rate of 18 per second or .056 second per character, and the punch portion punches at the rate of 18 per second or .056 second per character.

The total time for any program may be easily estimated by counting the number of program steps. A program step is equivalent to a key depression. In the instruction 01 + 02 there are five key depressions or program steps. There is a program step for 0, another for 1, another for +, etc. The total time for this instruction would be 5 × .056 seconds (the time to execute a single step) or .28 seconds to add two numbers in storage.

In some of the arithmetic operations, such as multiply, additional time is required to complete the operation. The additional times are given in the following table.

	AVERAGE TIME IN SECONDS
Read Out	1.06
Read In	1.17
Long Shift	.22
Divide Multiply (auto-point)	1.54
Multiply (auto-point)	1.36
Divide (auto-point)	1.43
Square Root (auto-point)	2.23
Divide Multiply (Fixed Point)	1.15
Multiply (Fixed Point)	1.15
Divide (Fixed Point)	1.15
Square Root (Fixed Point)	1.92

In the problem (in auto-point):

12 × 08
+ 25
CR
RO

Program Steps (11 × .056)	.616 sec.
Multiply	1.365
Read Out	1.06
Total	3.041 sec.

In the first programming example for obtaining the sum of three products:

			PROG. STEPS
01	ENT	P	3
02	ENT	Q	3
03	ENT	R	3
04	ENT	S	3
05	ENT	T	3
06	ENT	U	3
01	× 02		5
07	COPY	A	4
03	× 04		5
07	+	A	4
05	× 06		5
07	+	A	4
	CR		2
	RO		1
			48

48 × .056	2.68 sec.
3 Multiplies	4.095
1 Read Out	1.06
	7.83 sec.

The additional time to enter the digits for the six numbers (P, Q, R, S, T, and U) will depend on the rate of speed at which the data is entered manually.

In the evaluation of the third-degree polynomial, $C_3Y^3 + C_2Y^2 + C_1Y + C_0$:

			PROG. STEPS
01	ENT	C_0	3
02	ENT	C_1	3
03	ENT	C_2	3
04	ENT	C_3	3
05	ENT	Y	3
04	× 05		5
	+ 03		3
	× 05		3
	+ 02		3
	× 05		3
	+ 01		3
	CR		2
	RO		1
			38

37 × .056	2.43 sec.
3 Multiplies	4.10
1 Read Out	1.06
	7.59 sec.

The additional time to enter the digits for the five numbers (C_0, C_1, C_2, C_3, and Y) will depend on the rate of speed at which the data is entered manually.

When performing a numerical calculation of any length, one must always have a plan of calculation. The degree to which this plan is explicit and detailed depends upon the complexity of the project and upon the nature of the mechanical aids to be used (i.e., paper and pencil, desk calculator, 610, or large high-speed calculator). The plan must be formulated before solution of the problem is begun. The plan may be changed once or many times depending upon the problem, and changes are to be expected. Beginning to solve a problem without a plan is similar to driving an automobile from one coast to the other without a road map and with all windows blackened.

For example, in using a desk calculator, you first state the problem, then list the data and constant numbers to be used in the course of the calculation, and then lay out the work sheet. Once these have been done, you can start to perform the numerical operation on the desk calculator. Changes in procedure can be made if intermediate results make changes necessary.

On the other hand, in setting up a problem for a high-speed calculator, the solution must be planned in detail from beginning to end before the machine can be instructed to perform it. To stop the machine and give it new data and instructions at many intermediate points in the calculation would be a complete waste of its resource of speed. Thus, every contingency (numbers in intermediate steps becoming very large or very small, loss of significant figures, division by zero, and so on) must be foreseen and appropriate provision made.

The more the operator takes advantage of the automatic features of the 610, the more completely he must plan his calculation. A thorough familiarity with a problem will find the operator with a plan of calculation formulated in his mind, but not written down explicitly. If the problem is *always* to be run by the same operator, it may seem unnecessary to write an explicit plan of calculation or program. If, however, the author refers the program to another operator to run, the second operator may have difficulty in understanding the logical sequence of steps. It is generally desirable, therefore, for all plans of calculation or programs for calculation to have an explicit listing giving the reason for each step.

In calculating with pencil and paper, a slide rule, or even with a desk calculator, based upon *almost* intuitive thinking, we perform many operations that are so nearly automatic that we are barely conscious that they are a part of the calculation. For example, we round-off numbers, we decide how to arrange our beginning data and answers on a sheet of data paper, we decide how far to carry out a part of the calculation before we are satisfied with the accuracy of the result, or we make a choice of two procedures, depending upon whether an intermediate result is positive or negative. In programming an automatic computer, the operator must be aware of all of these operations and give the machine precise instructions. These instructions must be in terms of elementary operations that the machine *understands.*

PLANNING THE PROGRAM

Preliminary Planning

Suppose we desire to evaluate a simple polynomial $2x + 5$ for all values of x in the range between -2 and $+2$ at intervals of 0.1. This calculation would be performed using an automatic program by starting with the value of x equal to -2.

The machine can be instructed to perform the indicated multiplication and addition for the first value of x, type out the result, add 0.1 to the value of x, multiply and add, type out the result, add 0.1 to the previous value of x, multiply and add, etc. But how does the machine know that it is *not* to go right ahead and get x equal to 2.1, multiply and add, type result, get a new x equal to 2.2, multiply and add, type result, etc.? The answer is that the operator must write a group of instructions into the program instructing the machine to test the value of x at each stage to see whether or not it is greater than 2. That is, the operator gives the 610 instructions that say, "form the difference $x - 2$." If it is negative, calculate, type, and get a new value of x, form a new difference and test it, etc. If the difference is positive, stop calculating. The machine tests the difference and makes the decision referred to in the instruction through the balance test hubs on the control panel. The functions of the balance test hubs is explained in the section on the *Control Panel.*

A similar example arises in evaluating the exponential e^x. Suppose we wish to calculate e^x for a value of x between 0 and 1 and that we wish to leave the result accurate to four decimal places. We could use the series

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots\ldots$$

To cause the machine to stop evaluating terms of the series after the desired accuracy had been reached, one would say something like, "After calculating each term, $\frac{x^n}{n}$, test to see whether it is greater or less than .00005. If it is greater, add or subtract the term as appropriate; if it is less, stop calculating."

These examples are not intended to show in detail how to program these problems. They are merely meant to indicate that, before performing a calculation on the 610, a plan must be made, which can roughly be broken down into the following stages:

1. Analyze the problem into a sequence of basic operations that can be expressed in 610 language on the program tape or can be wired on the control panel.

2. Decide what input data and what results are to be typed or otherwise recorded and in what format.

3. Decide what storage locations are to be used to accommodate the various constants and variables that enter the problem.

4. Write out, sentence by sentence, the precise instructions that will get input data into the 610's memory and will cause the 610 to carry out the necessary operations. If the control panel is to be used, this last stage will also include a plan for wiring the control panel.

It should be emphasized that these plans need not, or should not, be perfectly rigid. The operator should allow himself the freedom of watching the development of the calculation as it is performed and of changing his plans in mid-calculation.

Most-Frequent Types of Calculations

Statistical studies of numerical computations show that about 85% of the basic arithmetic operations performed by computers (human, mechanical or electronic) are of two simple types.

Type I: A product of two numbers is added to a number previously obtained.

Briefly

prev. + a b

For example, in evaluating a sum of products (in mathematical notation $\Sigma_i\ a_i\ b_i$), all the operations are of this type.

Type II: A number is added to a number previously obtained, and the sum is multiplied by a third number. Briefly,

(prev. + a)b

For example, a polynomial may be evaluated in such a way that all operations are of this type. In mathematical notation, a polynomial of the third degree may be written as

$$[(ax + b)\ x + c]\ x + d$$

The IBM 610 performs both of these operations with equal ability.

Example of First Type of Calculation

Suppose we consider a small problem of Type I. Given six input numbers, P, Q, R, S, T, U, the operator wishes to evaluate the sum of the three products PQ + RS + TU. In thinking about the problem, he decides to enter the input data into six storage registers; so, he notes on his planning sheet:

P	01
Q	02
R	03
S	04
T	05
U	06

Further, he decides he will use register 07 as the summation register to accumulate the sum of products as they are formed. At the end of the computation, he will be able to read out the required answer from register 07. From this point on, the amount of detailed note-making the operator does will depend on the complexity of the problem. It is recommended that the beginning programmer write out in full de-

IBM 610 PROGRAMMING SHEET

Storage Usage

00	01	02	03	04	05	06	07	08	09
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39

Instr. No.	CI.	First Reg.	Oper.	Second Reg.	Reg. Sel. After	REMARKS
		01	ENT		01	P
		02	ENT		02	Q
		03	ENT		03	R
		04	ENT		04	S
		05	ENT		05	T
		06	ENT		06	U
		01	X	02	A	PQ Developed
		07	COPY	A	07	Store PQ
		03	X	04	A	RS Developed
		07	+	A	07	PQ & RS Stored
		05	X	06	A	TU Developed
		07	+	A	07	PQ+RS+TU Stored
			CR		A	Turn on typewriter, return carriage
			RO			Read out accumulated sum.

FIGURE 57. PROGRAM FOR ACCUMULATING PRODUCTS

tail the commands to be given to the machine in sentence form as follows (Figure 57):

Data Entry

01 ENT (P)
02 ENT (Q)
03 ENT (R)
04 ENT (S)
05 ENT (T)
06 ENT (U)

Note that the operator concludes the entry of each number with the algebraic sum of the number.

Computation and Output

01 × 02 This sentence causes the machine to evaluate the product of the two numbers in registers 01 and 02 to put the product in the A register, which remains selected (P times Q).

07 COPY A Addresses register 07, erases any previous content, adds to the zeros in 07 the product developed in A, 07 stands selected.

03 × 04 Develops product of contents of registers 03 and 04 and places product in register A. (R times S).

07 + A Adds the product of R and S to the product of PQ. 07 stands selected.

05 × 06 Develops product of the contents of registers 05 and 06 and places the product in register A (T times U).

07 + A Adds product of T and U to the two previous products. 07 stands selected.

CR Turns on the typewriter and returns carriages. 07 stands selected.

RO Reads out the contents of the selected register 07 to the typewriter.

A keyboard instruction (KB) should be inserted at the beginning of a program for purposes of location of the starting point. For repeated use of the program, the PCH and DUP switches are used appropriately.

Example of Second Type of Calculation

Suppose an operator wishes to evaluate a third-degree polynomial

$$C_3y^3 + C_2y^2 + C_1y + C_0$$

This can be expressed as

$$[(C_3y + C_2)y + C_1]y + C_0$$

The operator decides that the constants C_0, C_1, C_2, and C_3 are to be stored in storage registers 01, 02, 03, and 04, respectively. The value of y to be used in the evaluation is stored in register 05. The coding is as follows:

01 ENT C_0
02 ENT C_1
03 ENT C_2
04 ENT C_3
05 ENT y

04 $\times$ *05* 04 is selected and its contents (C_3) multiplied by y. The result is in the A register, A register is selected.

+03 C_2 is added to contents of A register.

×05 The product $(C_3y + C_2)y$ is formed in the A register, A is selected.

+02 C_1 is added to the contents of the A register.

×05 The product $(C_3y + C_2)y + C_1y$ is formed in A register. A register selected.

+01 Finally C_0 is added to the product. A register selected.

CR Typewriter on, carriage returned. A register still selected.

RO The contents of the A register are read out to the typewriter.

For uniformity and convenience, a program sheet has been set up to ease programming. Both programs have been written on this program sheet to illustrate its use (Figures 57 and 58).

Instr. No.	CI.	First Reg.	Oper.	Second Reg.	Reg. Sel. After	REMARKS
		01	ENT			C_0
		02	ENT			C_1
		03	ENT			C_2
		04	ENT			C_3
		05	ENT			Y
		04	X	05	A	C_3Y Developed
			+	03	A	C_3Y+C_2 Developed in A
			X	05	A	$(C_3Y+C_2)Y$
			+	02	A	$(C_3Y+C_2)Y+C_1$
			X	05	A	$[(C_3Y+C_2)Y+C_1]Y$
			+	01	A	$[(C_3Y+C_2)Y+C_1]Y+C_0$
			CR			Turn on typewriter, return carriage
			RO			Read out value of polynomial

FIGURE 58. PROGRAM FOR EVALUATING A POLYNOMIAL

IBM.

Form 23-6335-0 (8-57:5M-44951)