

A New Approach to Small-Computer Programming and Control

Abstract: A novel approach to computer programming and control, used in the IBM 610 computer, allows the solution of complex problems by an operator whose only previous experience with computing has been the desk calculator. The machine's command structure is designed so that the operator can at all times communicate with the computer by a series of short sentence-type instructions closely resembling the steps of manual arithmetic solution. A type of floating-decimal operation called the "auto-point" mode permits entry of data into storage locations with automatic positioning of the decimal point, without elaborate programming. The decimal point is automatically re-positioned during subsequent computation.

Introduction

There is a growing need for small computers which will solve problems in the area of computation between that of the desk calculator and that of the large electronic computer. Many of the problems in this intermediate range of computation stem from the normal routine of technical activity. In this paper, a new approach to small-computer programming and control will be described which permits the solution of complex problems, but which requires little more operator skill than does a conventional desk calculator. With the IBM 610 computer, which uses this approach, the average scientific worker can profit from the higher output capacity of an automatic calculator and yet be able to communicate with it directly, without the aid of a "middleman" or programming specialist.

Design of the new computing means was not simply a matter of scaling existing computers, but rather the use of proven attributes of such machines together with new design techniques to achieve objectives of the proposed machine. Clearly the design of a computing mechanism should be based on an analysis of the operations it is expected to execute. Studies of the arithmetic involved in practical computation indicate that over four-fifths of computing effort is expended in evaluating rather simple expressions which are of two basic types:

1. Previous result $+ ab$,
2. (Previous result $+ a$) b .

In Type 1, the product of two numbers is added to a previous result. By cyclic application of this operation, the familiar summation of products is developed. Type 2 calls for the addition of a number to the previous result, the

resulting sum to be multiplied by another factor. Operations of this kind are exemplified by the familiar "nesting" procedure for evaluating a polynomial:

$$ax^3 + bx^2 + cx + d = [(ax + b)x + c]x + d.$$

A point of special note is the relative importance of the "previous result" in computation. The probability is rather high that the previous result will appear as an operand in the new expression to be evaluated.

A more efficient command structure than that afforded by conventional single-address computers is used in the new system. Special effort has been made to provide more powerful basic commands and to eliminate the need for commands which simply transfer information with no other useful result.

One of the most difficult problems in planning the solution of a problem by an automatic computer is the placing of numbers in the proper positions in the machine's storage registers so that a maximum of significance may be retained throughout the calculation, yet at the same time avoiding the possibility of developing an intermediate result which will overflow the registers' capacity. Built-in provision for automatic decimal-point control in the computer relieves the operator of this concern. Using the auto-point mode of operation, the operator enters input data containing the decimal-point symbol in its normal position between two adjacent digits into the storage registers. The machine stores the decimal point along with the digital information, uses it to control all intermediate shifting operations necessary for reconciliation of the decimal point, and delivers output results containing the decimal point in its proper position. It is anticipated that the system will be operated

in the auto-point mode a great proportion of the time. However, the operator may, if he wishes, inhibit the functioning of the auto-point circuitry and program the performance of calculations in the conventional fixed-point mode.

Although the machine features an extremely simple coding procedure, it can perform, without supervision, complex computations such as those involved in matrix operations and the solution of differential equations.

A novel option allows the operator to enter data into the system and solve the problem at his own pace. Constant monitoring of progress enables the operator to check all input data and intermediate results, and make corrections if necessary. During the operation a paper tape may be punched, which upon solution of the problem will contain a complete program of the necessary steps. Since corrections are made as the operator creates his program, no debugging run is required and the program tape may be used immediately for other problems of the same type using new input data.

General system description

Units of the new computer are shown below. This is a

complete system in itself, but can work compatibly with other computing installations.

The system comprises three physical units interconnected by flexible cables: (1) a cabinet containing the electronic arithmetic unit together with its magnetic-drum storage unit and electro-mechanical control equipment; (2) a manual control keyboard which provides a cathode-ray-tube display in coded form of the content of any desired machine register; (3) an electric typewriter for automatic output printing. The cabinet is mounted on wheels and may be moved readily. The complete system weighs approximately 750 pounds and draws less than 20 amperes from a single-phase 120-volt source.

As shown in the functional diagram, Fig. 2, the components of the system are divided into three major groups of equipment: the electronic computer proper, the set of input or control devices, and the set of output devices.

• 1. Computer cabinet

The computer cabinet contains three main units: a magnetic-drum storage unit, an arithmetic unit, and an access and control unit governing the reciprocal flow of

Figure 1 The computer opened to show construction.



information between drum storage and the arithmetic unit as well as accepting numerical and control information from and delivering information to the external input and output devices.

The magnetic-drum storage unit provides storage for eighty computing and storage registers and four special-purpose registers. Although each of these registers is capable of storing 32 decimal digits and a symbol representing a decimal point, the machine is regarded normally as a 15-digit machine. The high-order digit is used as an indication of algebraic sign. The operator has access to any of the machine registers, but, in normal computing, addresses only the eighty registers in main storage and one of the special registers, called the *A* register. The magnetic-drum storage registers may be treated by the operator as a set of independent accumulating arithmetic registers, each of which is capable of performing upon its content the following operations which are selected by the operator from the keyboard shown in Fig. 3.

ENT (Entry)

Numerical information in the form of a magnitude followed by plus or minus sign can be entered directly into any register from any of the input devices.

RO (Read out)

Read out the left-hand fifteen digits of the content of any selected register to any of the output devices. (Any desired number of digits may be read out by appropriate control-panel wiring.)

CLR (Clear)

Set register content to zero.

CLR RH (Clear right half)

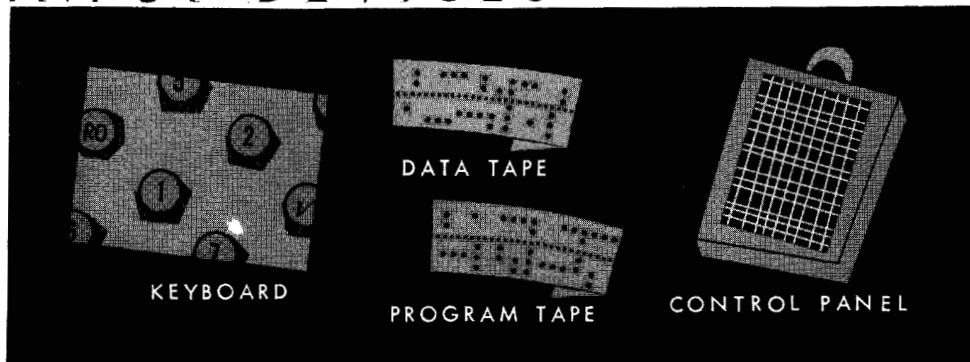
Set the least significant half of the register to zero.

CNV (Convert)

Change the sign of register content.

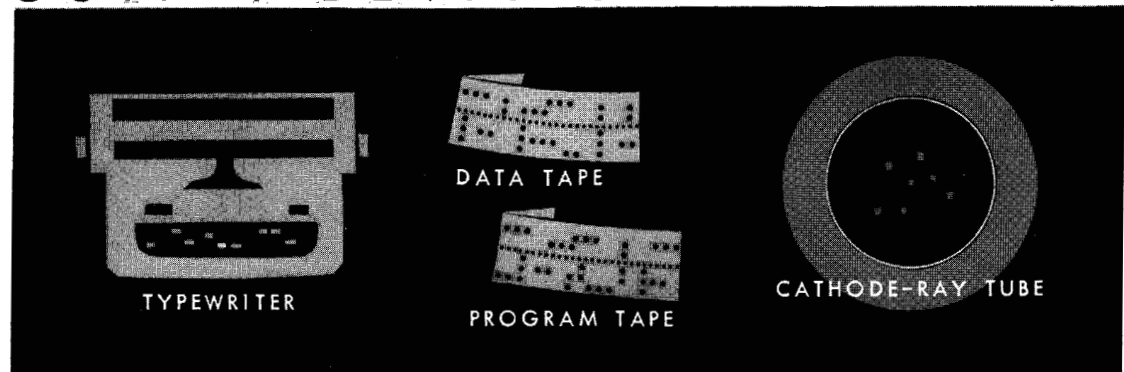
Figure 2 Functional diagram of the IBM 610.

INPUT DEVICES



COMPUTER

OUTPUT DEVICES



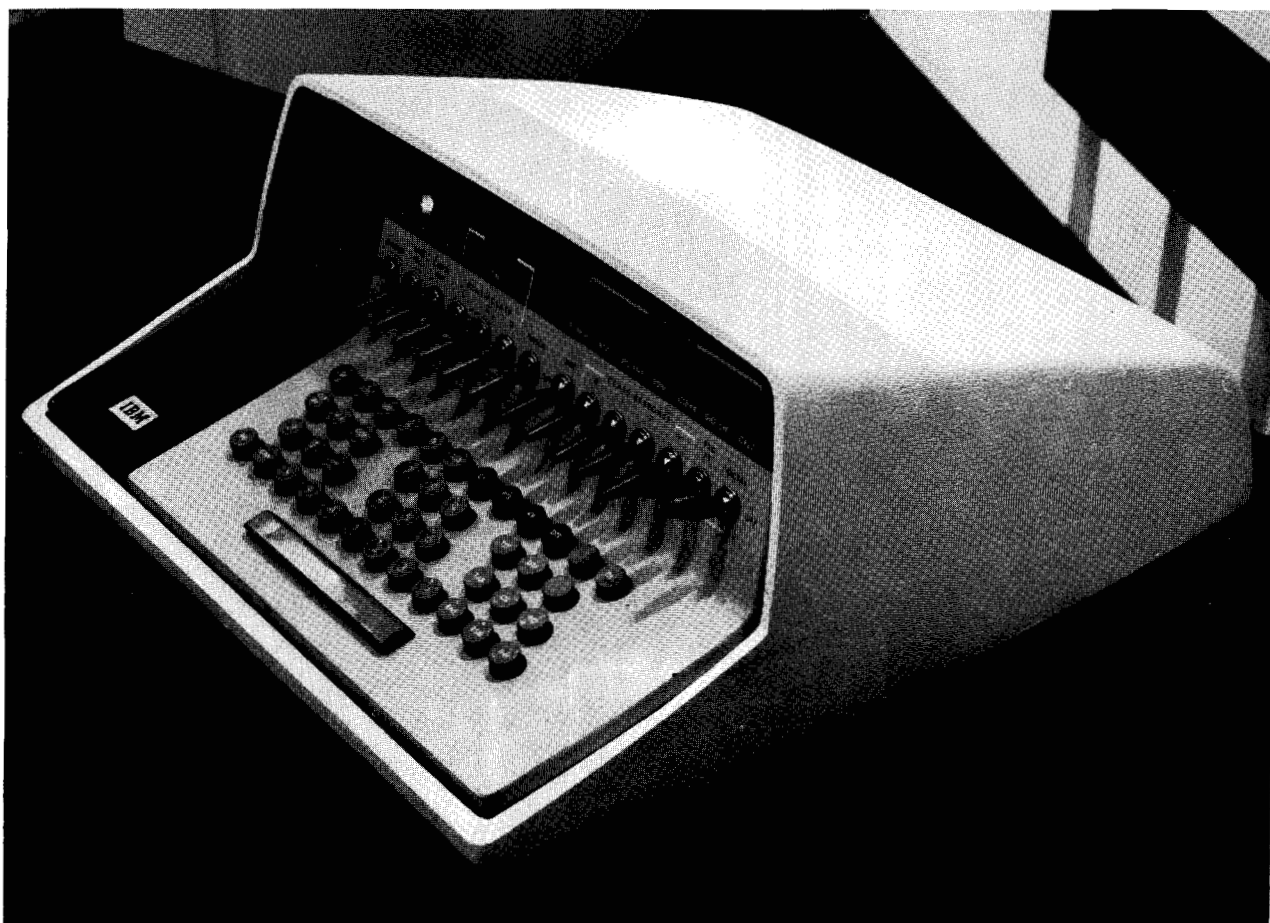


Figure 3 Keyboard for machine control and entry of data.

SL (Shift left)

Shift register content one place to the left.

SR (Shift right)

Shift register content one place to the right.

SL15

Fixed point: Shift register content fifteen places to the left.

Auto point: Shift content as far to the left as possible without losing any significant information (left-hand standard position).

SR15

Fixed point: Shift fifteen places to the right.

Auto point: Shift right until decimal point lies just to the right of the center of the register (right-hand standard position).

Other operations, which involve the content of more than one register, are discussed later in the section on command structure.

• 2. Input (or control) devices

Control of the computing system is exercised by the various input devices shown pictorially in Fig. 2 and listed below:

- a) Manual keyboard;
- b) Pluggable control panel;
- c) Program-tape reader;
- d) Data-tape reader.

These 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 entry of numerical information, but on occasion it also may be used in functional control of the calculator. Similarly, the program-tape reader, though normally used for program insertion, may also be used as a data-handling device.

Eight-channel tape is used for program control, but data tapes may be of either five- or eight-channel width.

The readers will accept standard teletype tape as punched by a standard teletype perforator. No more than one controlling unit may be in operation at any one time; however, the unit in control can transfer control automatically from itself to any other control unit.

3. Output devices

As shown in Fig. 2, output from the computer is available from any of the following means:

- a) IBM electric typewriter;
- b) Data-tape punch;
- c) Program-tape punch;
- d) Visual display on cathode-ray-tube indicator.

The first three 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 to provide permanent records of input data, intermediate results and final results of computation, either for future visual reference or in a form suitable for use as input for this or other automatic computing machines. The tape punches may also be used in conjunction with the tape readers to supplement the internal storage of the electronic computer by punching intermediate results on the paper tape, which may be caused to pass through the associated tape reader when needed, making the numbers so stored accessible to the computer.

The numerical content of any register of the machine can be displayed in coded form on a small cathode-ray-tube indicator.

Command structure

1. Arithmetic operations

Early electro-mechanical computing machines employed storage devices which were also capable of performing the arithmetic operations of addition and subtraction. Direct addition of the content of one register to that of another was an accepted and valued computational technique. With the advent of the electronic calculating machines this system was abandoned, not because it was inconvenient for computation but rather because accumulators were substantially more costly than were simple storage registers. Hence it became common practice to build machines with a single accumulator and a plurality of pure storage registers.

The present machine is so constructed that each of the eighty-four magnetic-drum storage registers is effectively an accumulating register, and the content of any register can be added or subtracted directly to or from the content of any register. The directness and simplicity of this operation is reflected in the simplicity of the machine language required to evoke it. Thus, if the operator wishes the machine to refer to the content of an initial "selected" register 22 and add to it or subtract from it the content of a subsequent "addressed" register 45, he "says" to the machine:

22 + 45 (Refer to register 22; add to its content the content of register 45)

or

22 - 45 (Refer to register 22; subtract from its content the content of register 45).

It is to be noted that these operations change the content of register 22, while leaving the content of register 45 unaffected.

In practical computation this is the most useful form of addition or subtraction command. Only rarely is it necessary to preserve unchanged the content of the first selected register. For these cases, it is a simple matter to evaluate the sum in a third arbitrarily addressed register, using the COPY (Clear and add) command. Thus, to find the sum of register 22 and 45, while preserving their content, choose a third register, say register 23, and instruct the machine:

23 COPY 22 + 45 (Refer to register 23; copy therein the content of register 22; add to this the content of register 45).

The commands required to call for a multiplication are equally direct:

22 × 45 (Refer to register 22; find the product of its content by the content of register 45).

or, of course,

45 × 22.

The product is developed in a special "answer" register called the *A* register, and the contents of 22 and 45 are undisturbed. The *A* register has all the properties of any of the other registers of the machine, including the capability of direct addition or subtraction, but has the additional property that the product is developed therein during multiplication and certain other operations.

Similarly, the operator may instruct the machine as follows:

22 ÷ 45

22 ÷ × 45 36

22 $\sqrt{\quad}$.

The first command obviously causes the machine to divide the number in register 22 by the content of register 45, the answer being developed in the *A* register. The last two commands are two more powerful unit operations of especial utility in technical computation. The second command causes the compound quotient product

$$\frac{C(22)}{C(45)} \times C(36)$$

to be developed in the *A* register; while the third command causes $\sqrt{C(22)}$ to be developed in *A*. The expression $C(22)$ denotes the content of register 22.

A point to be noted especially is that the "word" $\sqrt{\quad}$,

meaning "Extract the square root of the content of the register currently selected," is contained within the machine's "vocabulary," as are the other commands exemplified. The operator need not translate the instruction into an arbitrary numerical code; he has only to write down a mathematical symbol. When he comes to the machine's keyboard he will find a control key labelled with the same symbol which he will depress to give the instruction to the machine.

The foregoing simple examples having been given, a more nearly precise statement of the basic principle of the command structure may be made. In the solution of a problem, the operator first selects one of the machine registers by giving the machine a two-digit numerical address ranging from 00 to 79 to designate one of the machine's eighty registers in main storage, or by depressing the key marked *A* to select the *A* register. Having selected a register, he can then instruct the machine to perform one or more operations involving the content of the selected register. Once a "register selection" is made it remains effective until the operator avails himself of one of the following three options:

1. The operator may make a new register selection. He does this simply by giving the machine a new register address *not preceded by an operational symbol*. Thus to instruct the machine to refer to register 22 and to subtract from its content both the content of registers 45 and 46, the operator may "say"

22 — 45 — 46,

but to tell the machine to subtract the content of register 45 from register 22 and then add the content of register 36 to the content of register 18, he will say

22 — 45

18 + 36

2. The operator may call for one of the operations of the multiply, divide, square-root family, which develop the answer in the *A* register.

3. The operator may give the machine the REL (Release) command.

In either of these last two cases, the machine automatically drops the previous register selection and selects the *A* register. It will be noticed that at the completion of any given operation the result of that operation is contained in the register selected and is available for further use as an operand.

The numerical content of the selected register is displayed automatically on the cathode-ray-tube indicator at the upper left of the keyboard shown in Fig. 3. The numerical address of the selected register is displayed visually at the left of the indicator panel on the front of the main cabinet.

• 2. Input-output operations

To enter a number into a selected register, the operator uses the ENT command, which (1) clears the register and (2) changes the function of the number keys from

register selection to digital entry. He enters the number on the keyboard in form of magnitude *followed by algebraic sign*, upon receipt of which the machine automatically drops out of the entry state so that the number keys again may be used to address registers. Thus,

22 ENT 3.14159+

23 ENT 2.71828+

will enter the numerical equivalents of π and e in registers 22 and 23, respectively.

If the operator wishes to instruct the machine to read the next number from the data tape, he uses one of the control-transfer commands, that is,

22 ENT DTR.

The first two commands select register 22 and prepare it for entry; the last command transfers control to the tape reader which reads the next word, whose termination is indicated by a special end-of-word symbol which was automatically punched on the tape when it was prepared. (If the data tape is prepared by teletype, the operator is instructed to prepare the tape as though he were producing a vertical column of words. The resulting carriage-return, line-feed and figure-shift symbols on the tape are recognized as the end-of-word symbol by the IBM 610.) As will be seen later, the end-of-word symbol can cause control to transfer to any desired control agency.

Control of the output devices is equally simple. The output devices receive their information from a common output bus which is energized whenever either an RO or an ENT command is given. To enable an output device to record the information appearing on the output bus, the operator uses one of a set of output-enabling commands:

TYP — turns on typewriter

CR — turns on typewriter and returns carriage

TAB — turns on typewriter and moves carriage to next tabular stop

DTP — turns on data-tape punch

PTP — turns on program-tape punch (available by control panel wiring).

Thus, to read out the content of register 22 to the typewriter, causing, incidentally, a carriage return before the readout, the operator says

22 CR RO.

He may wish simultaneously to prepare a punched-tape record of the information, in which case he may say

22 CR DTP RO.

The output devices may be enabled simultaneously in any combination.

Notice also that information is supplied to the output bus during an entry, so that the machine may prepare a record of input data. Thus,

22 CRT ENT 3.14159+

causes π to be entered into register 22 and also to be typed in the left-hand column of the output page.

When the machine is operating in the auto-point mode, all numbers are kept in the right-hand standard position. In reading out in the auto-point mode, the operator may wish (1) to read out the fifteen most significant digits of the answer regardless of decimal position, or (2) prepare an output page showing data tabulated in columns with the decimal points lying on a vertical line. To accomplish these alternate objectives the operator says

(1) 22 SL15 CR RO

or

(2) 22 SL SL CR RO.

(1) will read out the most significant digits; (2) will read out two digits to the right of the decimal point. Following the read out, the register content is returned automatically to right-hand standard position.

Example: solution of simultaneous equations

An illustrative example of the way in which the command structure just described makes possible a direct program for the solution of practical problems is afforded by the following routine for the solution of a set of three simultaneous equations. The method employed first transforms the original matrix of coefficients into a triangular matrix and then performs a back solution for the desired variables.*

(01)	(02)	(03)		(13)		(04)
a_{11}	a_{12}	a_{13}		x_1		b_1
(05)	(06)	(07)		(14)	=	(08)
a_{21}	a_{22}	a_{23}		x_2		b_2
(09)	(10)	(11)		(15)		(12)
a_{31}	a_{32}	a_{33}		x_3		b_3

First the operator makes arbitrary storage assignments; those indicated by the bold-face addresses associated with the matrix elements will be taken for this example. Having entered the matrix elements, the solution proceeds:

05 $\div \times$ 01 02

06 $- A$

05 $\div \times$ 01 03

07 $- A$

05 $\div \times$ 01 04

08 $- A$

(End of transformation of second row.)

09 $\div \times$ 01 02

10 $- A$

09 $\div \times$ 01 03

11 $- A$

09 $\div \times$ 01 04

12 $- A$

(End of first transformation of third row.)

10 $\div \times$ 06 07

11 $- A$

10 $\div \times$ 06 08

12 $- A$

(End of second transformation of third row. Completion of triangularization.)

12 $\div$ 11

15 COPY A SL15 CR RO

(x_3 into 15 and type out.)

07 $\times$ 15 CNV $\div$ 08 $\div$ 06

14 COPY A SL15 TAB RO

(x_2 into 14 and type out.)

03 $\times$ 15

13 COPY 04 $- A$

02 $\times$ 14

13 $- A \div$ 01

13 COPY A SL15 TAB RO

(x_1 into 13 and type out.)

This problem, which is not an especially simple one, illustrates the fact that essentially all the engineer needs to know is the mathematics of the solution in order to use the machine to solve practical problems. It is of interest to note that the operations involved in the solution of this problem are almost all of the Type 1 described in the Introduction.

Automatic control devices

• 1. Program-tape unit

If only one solution of a set of equations were desired the manual solution described in the preceding section might be the quickest and easiest method of obtaining the solution. If more than one set of input values are on hand and a solution is desired for each set, the machine's tape program unit is of great assistance in performing repetitions of the calculation. Oversimplifying slightly, it may be said that the operator can "show the machine how" to

perform a calculation by actually carrying out the solution manually as above. During the first solution the machine executes the steps as the control keys are operated and simultaneously records on paper tape a symbol corresponding to each control key depressed. After going through the calculation the first time, the operator has obtained the first set of required answers; thereafter the information stored on the tape can control the operation of the machine for as many subsequent iterations of the procedure as may be desired. The strip of tape also may be preserved and re-used at a later time if the operator so desires.

To use the computer in this manner the operator makes a slight change in the commands given to the machine. To illustrate this, consider the following example.

Suppose it be desired to calculate the distance from the origin to a point whose x and y coordinates are given. The manual solution goes as follows:

```
01 CR ENT (x)
02 TAB ENT (y)

01 × 01
03 COPY A
02 × 02 + 03 √ SL15 TAB RO.
```

This program will cause x , y , and the computed value of $\sqrt{x^2+y^2}$ to be typed on a horizontal line on the output page. If the operator wishes to repeat the solution a number of times he must tell the machine where to look for each new piece of input data. To create such a program, he will first turn on the program punch switch to be found at the center of the line of switches above the operating keys and then key as follows:

```
01 CR ENT KB (x)
02 TAB ENT KB (y)

01 × 01
03 COPY A
02 × 02 + 03 √ SL15 TAB RO.
```

This differs from the preceding only in the insertion of the command KB (Send control to keyboard) with each entry of data. As control is already at the keyboard at the instant the KB key is depressed, no immediately apparent change in the state of the machine takes place; but the KB symbol is punched in the program tape, and on subsequent iterations it will cause the machine to stop at this point in the program to allow the operator to enter new input data. The KB command, if given while the machine is in the entry state, causes suspension of program punching *until the entry state is terminated*. In this way the input number, which is pertinent only to the first execution of the calculation, is prevented from appearing on the program tape. (We note here that an operator may wish to embed a constant in the program

tape, and if so he simply punches such a command sentence as

```
62 ENT 3.1416+
```

in the tape. Since the KB command is *not* given, the numerical value is allowed to be punched on the tape. Thus in each iteration the tape can cause the constant to be entered.)

Having completed execution of the first solution, the operator turns off the PUNCH switch, turns on the DUP switch, and depresses the PTR button, which sends control of the machine to the program-tape reader. The tape just produced is accumulated in a variable-length slack loop between the punch and the reader and is ready to feed into the reader as soon as the initial punching is concluded. As the commands are read to control the next cycle, they are simultaneously duplicated by the program punch (because the DUP switch was turned on) ready to control subsequent iteration.

● 2. Minor-loop iteration

As was mentioned earlier, the second tape unit, which is physically identical with the first, is capable of handling program tapes as well as data tapes. This makes it possible to execute minor iterative loops within a major program loop. Facilities are provided to enable the copying of information being read by the first reader at the second punch. Thus a complete minor cycle punched in the main program and read by the first reader can be copied at the second punch. Control then can be transferred to the second tape unit and the minor iteration carried to termination. Thereafter, control returns to the first tape unit for resumption of the main program.

● 3. Response class

The PUNCH CLASS switches on the keyboard (Fig. 3) cause any one of three special identifying symbols to be recorded on the tape in association with each command while the switch is on. Thus there are said to be four classes of command, zero through three. When the resulting tape is read in control of the machine, a command of a given class is obeyed if and only if the corresponding "class response" is enabled.

Class response can be controlled either from the keyboard or by means of control-panel wiring. It may be used conditionally to allow choices between alternate programs, depending upon criteria established in the program. As a simple example of the use of this feature, which is quite a powerful tool in sophisticated programming, the operator may make it standard practice to conclude all programs such as that in the example above with a Class 1 KB command. This causes the machine to stop and return control to the keyboard at the end of any desired cycle if the CLASS RESPONSE 1 switch is turned on, otherwise the machine will repeat the calculation indefinitely.

● 4. Pluggable control panel

The control panel is a board containing a large number

of holes or "hubs." When the control panel is inserted in the machine it provides the operator with access to internal circuitry. By inserting wires according to a predetermined pattern and with the knowledge of the purpose of each hub, the operator may design a program.

The control panel is particularly valuable for frequently used mathematical routines such as sine or cosine. A routine is prepared once in a rather permanent fashion and is available for use with any program thereafter by transferring control to the inserted panel.

The control panel is shown in Fig. 4. At the top of the panel is a group of function hubs essentially in one-one correspondence with the keys of the keyboard and correspondingly labelled. If a control impulse is brought to one of these hubs by a wire from one of the hubs which are sources of control impulses, the effect upon the computer is the same as though the correspondingly labelled key had been depressed.

There are three sets of source hubs on the panel (Fig. 4):

1. Program-step hubs

Two hundred double hubs arranged to be energized in rotation by a stepping-switch mechanism when control is sent to the control panel. The position is displayed at the right side of the indicator panel. The stepping switch can be instructed to jump to any desired step and take up stepping from that point by wiring a control impulse to one of the PROG SKIP IN hubs and connecting a wire between the corresponding PROG SKIP OUT hub and the desired program step.

2. Sequence hubs

Twenty double hubs which may be energized selectively at any time by manual operation of the keyboard (depressing the SEQ key followed by two digit keys) or by the tape readers reading a SEQ symbol followed by two digits on a tape.

3. Word-end hub

One double hub which emits a control impulse whenever a tape reader reads the end-of-word symbol. This hub can be connected by a wire to the appropriate control-transfer hub to send control to the desired control unit after reading a number from the data tape.

By connecting consecutive program-step hubs directly to the proper function hubs, the operator can wire on the control panel the instruction for execution of a desired routine. Then, if a wire is connected from one of the SEQ hubs, say SEQ 06, to one of the PROG SKIP IN hubs, and if the corresponding OUT hub is wired to the program step preceding the routine, the operator can cause that routine to be executed simply by giving the command SEQ 06 at the keyboard, or it can be called forth by the command SEQ 06 read by a tape reader. Whenever the PROG SKIP mechanism is impulsed, control of the machine is transferred automatically to the control panel. Such a simple command is shown in Fig. 4.

The preceding paragraph presents a conceptually sim-

ple picture of the use of the control panel and the SEQ commands. However, much more powerful techniques are made possible by the incorporation of certain impulse-routing hubs on the panel which make possible conditional programming of a rather sophisticated nature in a direct and straightforward way. The principal logical elements are the balance-test hubs and the selectors.

1. Balance-test hubs

These twelve sets of three hubs are wired to twelve independent transfer points on a relay which is controlled by an electronic circuit which monitors the sign of the content of the register currently selected. If the register content is positive a connection is established between each "C" (or common) hub and the associated "+" hub; whereas if the register content is negative, each "C" hub will be connected to its associated "-" hub and there will be no connection between "C" and "+." Each of the twelve sets is independent; there is no connection at any time between two "C" hubs, for example.

2. Selector hubs

The ten selectors are simply ten independent latching relays which may be set in either of two states by impulses directed to their PU (pick up) and DO (drop out) hubs. Two independent transfer points of each relay are brought to hubs on the control panel. If the selector is picked up, a connection is established between each of its "C" hubs and the associated "T" (transferred) hub. If the selector is dropped out, a connection is established between each "C" hub and the associated "N" (normal) hub. The state of each selector is also indicated visually in the center column of the indicator panel.

These hubs may be interposed between source hubs and function hubs to exercise control over the routing of control impulses and hence exercise control over the course of a computation. As a simple example, suppose one were to connect a wire from hub SEQ 02 to a Bal Test "C" hub, and then from the associated "-" hub connect a wire to the CNV function hub, as shown in Fig. 4. Now imagine what happens when the command SEQ 02 is given. A control impulse is emitted by the hub SEQ 02 and travels along the wire to the Bal Test "C" hub. If the content of the register currently selected is positive, the impulse is routed to the "+" output hub and finds a dead end there; whereas if the register content is negative, the impulse is routed to the "-" output and thence to the CNV function hubs, causing the conversion operation to take place, changing the sign of the register content. Effectively, then, one has wired into the machine the command "Take the magnitude." This example illustrates two things: 1) the use of a Bal Test point to route conditionally a control impulse; 2) a second use of the SEQ channel wherein it is used as a communication link between the keyboard or tape and the logical elements whose terminals are brought to the control panel. Notice that the operation just described does not "transfer control to the control panel."

The selectors are used primarily as memory devices

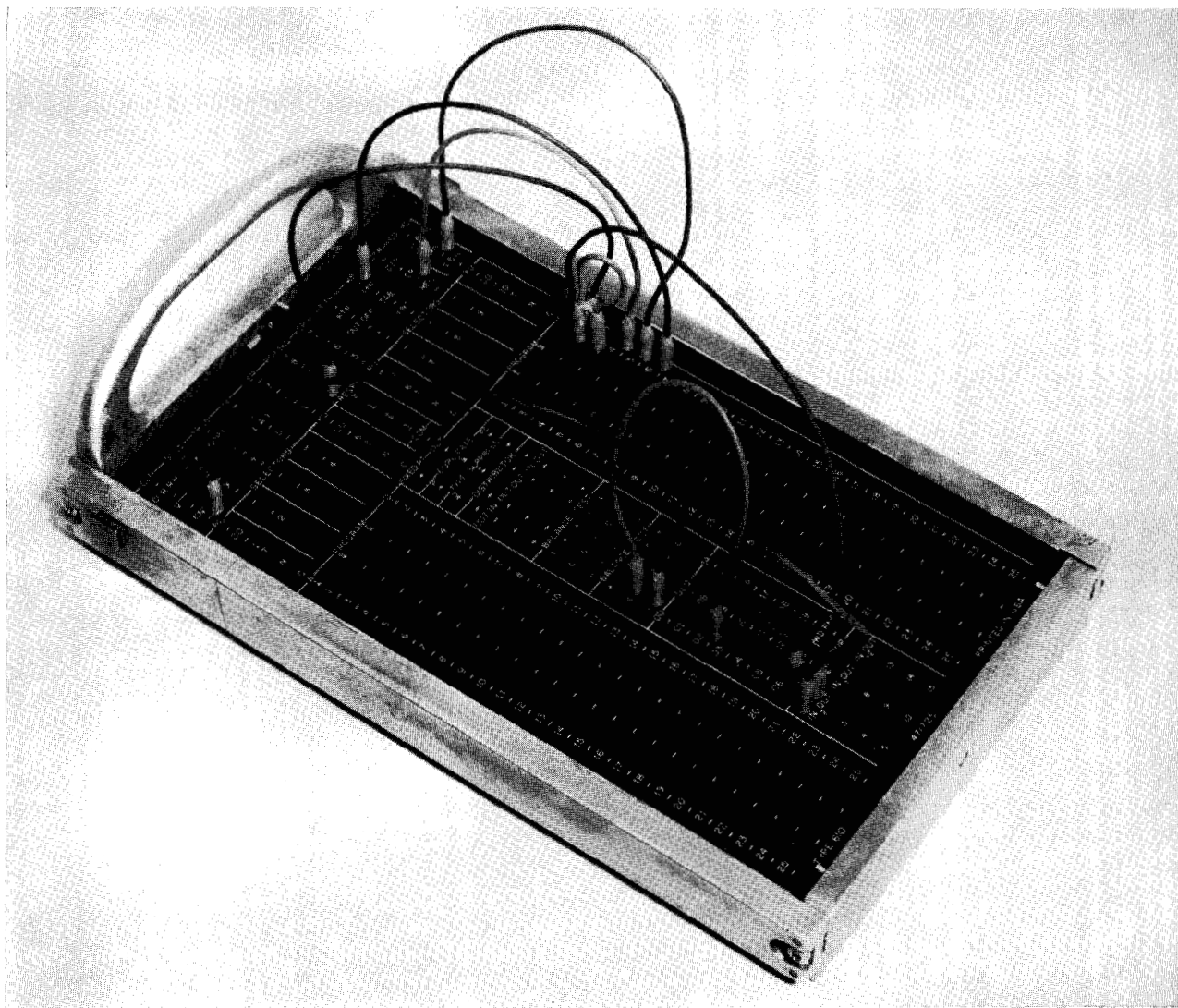


Figure 4 Pluggable control panel.

capable of remembering the fact that a certain control impulse was routed to its PU or DO terminal. As an example, suppose that, in connection with the "magnitude" operation described above, it were desirable to remember for later use in the calculation whether the number in the register were initially positive or negative. As before, connect SEQ 02 hub to Bal Test "C." From "+" wire to Selector 1 PU; from "-" wire to the common connection of a set of "distributor" hubs, from whose two outputs run wires to CNV and Selector 1 DO. As before, SEQ 02 will cause the computer to take the magnitude of the content of the selected register, but in addition it will pick up Selector 1 if the content is initially positive, but will drop it out if the register content is initially negative. Then, at a later stage in the program a control impulse may interrogate Selector 1 to make a decision based upon the original sign of the number in

question. In this example, the use of the "distributors" was mentioned. These are blocking devices which allow control impulses to pass in only one direction, from IN hubs to OUT hubs. Any number of source hubs may be interconnected by wiring, but function hubs may be paralleled only through distributors to prevent the establishment of unwanted "back circuits."

The control panel also bears certain other hubs which are of considerable practical importance. Most significant of the remaining hubs is that set which controls the class response of the tape unit. A connection between the "C" hub of this set and one of the hubs numbered 0, 1, 2, 3 will allow the tape reader to obey commands of the corresponding class. It is easy now to see that if the connection enabling a given class response is established via a contact on one of the selectors, the state of that selector will determine whether or not commands of that class

will be obeyed. Since the selector can be picked up or dropped out conditionally or unconditionally by the tape unit, communicating with the logical devices via the SEQ channel, the tape can bear a program incorporating a number of alternative procedures and choose among them on the basis of results derived during the course of the calculation.

Other hubs on the panel make possible control-panel control of fixed-point or auto-point, octal or decimal computation, zero suppression, sign suppression, and variable-word-length read out.

The control panel may be used as a principal programming agency when it is desired to use both tapes as data-handling devices. For example, a general-purpose simultaneous-equation-solving routine can be wired on a control panel and preserved for use whenever needed. A very important use, however, is as a library of frequently used sub-routines, such as those required for the generation of commonly required functions such as the sine, cosine, or exponential.

Address-tape operation

Machine solution of certain problems, matrix inversion for example, is characterized by extremely simple sets of arithmetic operations to be performed successively upon the contents of each member of a set of storage registers. If the order of the matrix is small, as in the preceding example, the most efficient solution is obtained by detailed programming of the computer in the manner indicated. For higher-order matrices this becomes impractical, and an approach which does not involve a back solution is used to better advantage. Here successive rows of the matrix are read into storage one row at a time and used as operators to transform other rows of the array. During the solution the machine must repetitively scan the elements in the operator row and detect the scanning of the last element in each row.

In solving a high-order matrix-inversion problem, the pluggable control panel is used as the primary control agency and the program tape as an address tape to supply successive element addresses in each row as needed. Therefore, instead of planning a modification routine to generate successive addresses by a process of computation and testing the generated address to determine whether it is the "last address," the operator simply punches the list of row-element addresses into the program tape. Each address is followed by a symbol instructing the machine to send control to the control panel. In addition, the address of the last element of the row is accompanied by a sequence command, e.g. SEQ 01, identifying it as the last element of the row. A given element of the operator row transforms an element of another row of the matrix by means of a simple routine wired on the control panel. At the point in the routine where the address of the operator element is required, the control panel is wired to send control to the program tape which reads the required address and duplicates the punching for use on a successive iteration; control is then returned to the control panel. When the last element

address is read, the SEQ 01 symbol is also read, causing emission of a control impulse from the SEQ 01 hub on the control panel. This impulse can be used to change the state of a selector relay which will allow the routine on the control panel to escape from the iterative loop in which it has been trapped and go to the next phase of the solution. Termination of the entire solution as well as the termination of one complete scan of the matrix elements are also indicated by sequence commands on the tapes.

Checking facilities

Checking facilities are provided within the machine to reduce the probability of an undetected error. The operation of the arithmetic unit is checked automatically each time a clearing operation is performed. Operation of the punched-tape units with eight-channel tape is checked by the use of an even-count redundancy check applied each time a symbol is read from or punched in the tape. This check is suspended when using five-channel tape. An overflow check is activated if the number in any register exceeds the capacity of that register. Additional check circuits monitor the operation of the control relays. If any of the checking devices detect an abnormal operating condition, operation of the machine is suspended and a warning light appears on the keyboard.

Operator errors

Plans for a machine featuring ease of operation must take into account not only machine errors but operator errors, which fall broadly into two classes: (1) mishaps, such as the inadvertent depression of the wrong control key; and (2) planning and programming errors.

Detection of errors of either class is substantially aided by the method of program entry described under Program Tape Control. As the operator keys in the program he can continuously monitor the intermediate results as they are obtained, and the appearance of an implausible result is usually a very helpful indication that an error has been made recently.

If the operator fails to detect the error during the preparation of the program tape, two machine features, single-step operation and program-tape interpretation, may be used advantageously to check the program for errors. Located at the lower right-hand corner of the keyboard (Fig. 3), are two keys labelled INT (interrupt) and RSM (resume). As the names imply, the INT key will cause interruption of the action of an automatic control device, and depression of the RSM key will allow computation to proceed from the point of interruption. If the INT key is held down steadily and then the RSM key is depressed and released, the computer will execute one command and then stop again. In this way, the operator can single-step through either a tape program or a control-panel program to find the point at which the intermediate results differ from expected values.

To determine whether improper operation is due to a planning error or to a mishap, the operator can order the machine to type out a complete interpretation of the

program tape by passing the tape through the reader with the TYPE PRG switch turned on. The resulting type-written list of commands may then be compared with the operator's plan of computation. If these agree, the operator knows that the error lies in program planning, and he must reappraise his analysis of the problem. However, if this inspection reveals a discrepancy, the machine can be of material assistance in producing a corrected tape.

There are two forms of tape correction: (1) deletion of superfluous or erroneous commands; (2) insertion of additional or corrected commands. Placing the erroneous tape in the reader, the operator orders the machine simultaneously to interpret and duplicate. As he nears the vicinity of an error he interrupts and single-steps through the last correct command. At this point, he advances the tape manually by turning the feed knob on the top of the reader so as to pass over the erroneous command(s) without reading. Turning on the PUNCH switch, he punches the correct command(s) on the new tape being produced and then resumes interpretation and duplication. It may be that he fails to stop the tape as it is being duplicated, until one or more erroneous commands have been punched. In this case, he can tell from the typed list how many erroneous commands have been duplicated. He then backspaces the punch manually the required number of spaces to bring the first false command under the die and obliterates it and its successors by overpunching with the DEL (delete) symbol.

Errors caught at the instant of execution may be somewhat easier to correct. Often the error causes no permanent damage to the calculation. It may be that the operator intends to punch

$22 + 26$

and inadvertently keys

$22 + 25$.

If he realizes that he has hit the wrong key and knows which key was depressed erroneously he can "program around" it by undoing his mistake as follows

$(22 + 25) - 25 + 26$.

It may be that the incorrect operation really causes no damage at all and may be left in the program as a superfluous operation so long as the correct operation is inserted immediately following it. If the keying error throws the machine into an undesired state, as, for example, depression of the "+" key instead of the "×" key, the machine may be released from this state by depressing the REL key, which punches an REL symbol on the tape. The operator may then make the appropriate first register selection and proceed as if nothing had happened, again

leaving the erroneous but harmless commands on the tape. If he prefers, of course, he may backspace the punch manually and overpunch with the DEL symbol.

If a mishap causes the irretrievable loss of a number, the operator may be well advised to start over, letting the correctly punched first part of the tape pass through the reader and control a repetition of the initial calculation, meanwhile duplicating the tape. In this way he may get back to the state of affairs as they were just before the catastrophic error occurred with a minimum of trouble.

Other modes of operation

• 1. Octal-decimal

A choice between the octal or decimal mode of operation can be made by means of a keyboard switch (Fig. 3) or control-panel wiring (Fig. 4).

Programs initially run and tested in decimal and subsequently run in octal give the operator results which can be used in conjunction with programming and debugging binary computers.

After selection of the octal mode, data are introduced in octal form (using digit entry keys 0-7) and thereafter all calculation takes place in the octal mode with no special demands upon the operator.

• 2. Fixed point

For general work, the operator will use the auto-point mode almost exclusively, because it eliminates the necessity for scaling associated with the fixed-point mode. For those occasions when an operator feels fixed-point operation is advantageous, the operator may choose this type of operation.

Fixed-point operation is useful in data-reduction runs where a given program is performed repetitively and advantage can be taken of the machine's increased operating speed. Here the scaling of data becomes less of a time factor when amortized over many solution runs.

When any computer 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 actually performing the intended operations. At all times the operator must take into account the magnitudes of the input and all intermediate numbers to insure against overflow and to insure correct alignment of decimal points in all operations.

Acknowledgment

The author wishes to acknowledge the contribution and suggestions of G. R. Stilwell, who reviewed the manuscript in light of equipment changes during the transition from development to production models.

Revised manuscript received November 5, 1957