A Digital Computer for Scientific Applications*

C. F. WEST†, MEMBER, IRE, AND J. E. DeTURK†

Summary—During the past two years development has been initiated on several large-scale automatic digital computing machines, both in this country and abroad. The present paper is concerned with the over-all organization of one such machine. A logical division of the machine into four major components is described, and the machine performance is interpreted in terms of these component functions. The electronic techniques used to accomplish the storage, transmission, and arithmetic manipulation of numbers, together with certain methods used for control of the computer, are briefly discussed. Although the paper is concerned with the design of a particular machine, it is felt that the design problems and engineering techniques are applicable to most large-scale computing machines.

I. INTRODUCTION

THE TERM "digital computer" applies to a calculating machine in which a number can assume only a discrete value the precision of which is determined by the number of digits used for its representation. A desk calculator is a digital (or discrete-variable) computer, whereas a slide rule is a continuous-variable computer. A large-scale digital computer is a machine not only capable of digital computation, but one which can perform long sequences of computations in accordance with a pre-established program of operation. Machines of this type are also referred to as sequence-controlled calculators. Such a computer is capable of solving complex problems involving thousands or millions of individual arithmetic operations without the intervention of a human operator.

Large-scale computers may be divided roughly into two categories: Scientific machines which are designed to perform large numbers of calculations based upon relatively few input data and yielding relatively few output data, and statistical machines of which the opposite is true. The calculator to be described is intended primarily for scientific applications. Some of the types of problems which the present machine is intended to solve are the following:

(1) The systematic handling of linear arrays.
(2) Solution of the partial differential equations of hydrodynamics.
(3) Fourier analysis and synthesis.
(4) Applications of electromagnetic theory.
(5) Study of shock waves.
(6) The solution of nonlinear differential equations.
(7) The problem of systematic sorting.

The only true computing operations which the machine can perform are the basic arithmetic operations

of addition, subtraction, multiplication, and division. Before the above complex problems can be presented to the machine for solution, they must be reduced to arithmetic processes through application of the methods of numerical analysis. These arithmetic routines must then be expressed in terms of coded commands which the machine is capable of following.

II. MACHINE ORGANIZATION

Fig. 1 is a block diagram showing the principal components of a large-scale digital computer. The arithmetic unit is the only true computing unit in the machine. That is, it is the only one capable of generating new numbers. The internal (or high-speed) memory is a
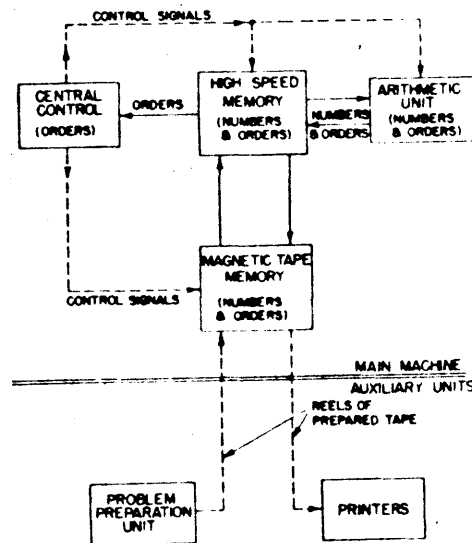

Fig. 1—Block diagram of the digital computer.

storage place for numbers and commands. During computation, numbers which serve as operands are transferred from the internal memory to the arithmetic unit, where the arithmetic operations take place. The result of each arithmetic operation is returned to the internal memory. The central control unit of the machine governs the exchange of numbers between the internal memory and the arithmetic unit. Central control governs this exchange in accordance with orders or commands which are also located in the internal memory. For each arithmetic operation, the central control must select two operands from the internal memory, and must supply these to the arithmetic unit. It must designate to the arithmetic unit which operation (e.g., addition, division) is to be performed, and must transfer the result of the operation to a selected memory position. Central

control then initiates the next operation by selecting from the internal memory the next command.

The magnetic memory units are used to supplement the internal memory. These units store numbers and orders on magnetic tape. The speed of operation of the magnetic units is considerably less than that of the internal memory, but the storage capacity is many times greater. These units also serve as input-output devices for the computer.

The page printers and problem-preparation unit shown in Fig. 1 are auxiliary units which are not directly connected to the main part of the machine, but which communicate with the computer by means of the magnetic memory units. The problem-preparation unit consists of a manually operated keyboard which is used to record initial numbers and commands on magnetic tape. This device makes use of additional magnetic storage containing the commands for frequently used computing routines. Thus, certain complete routines may be introduced into the computer by a single manual operation. The page printers are electrically operated typewriters which respond to signals recorded on magnetic tape. They are used to record the final results of computation.

Because one command is required for each arithmetic operation, it might seem that a prohibitive number of commands would have to be introduced into the machine in order to direct the solution of a relatively simple problem. This is not the case. The iterative methods of numerical analysis involve the repeated performance of computing routines. When a routine is repeated, the commands governing the computation may differ from those of the previous cycle only with respect to some systematic pattern of variation. By storing commands in the internal memory and by the use of suitable schemes for their coding, they may be introduced into the arithmetic unit and modified by addition or subtraction. As an example of the effectiveness of this process, the total number of commands which must be supplied to the machine to obtain all of the roots of a polynomial equation should not exceed fifteen. This number is independent of the degree of the polynomial.

The complexity of the problems which a machine can solve efficiently is limited both by computation speed and memory capacity. For example, partial differential equations in three dimensions and time may require a total storage of 10^6 numbers and may involve 10^{10} arithmetic operations. In the present machine, the internal memory has a capacity of approximately 4000 numbers, and the permanent storage medium associated with each magnetic memory unit has a capacity of 200,000 numbers. The over-all speed of the computer depends primarily upon the time required to perform the basic arithmetic operations and the time required to select a number from the internal memory. In this machine, 900 arithmetic operations together with the associated memory selections are performed each second.

Since the entire function of the machine is to carry out numerical computation in accordance with coded commands, the representations of numbers and commands are basic elements of the machine design. Numbers may be represented in a variety of ways, depending upon both the mathematical and the physical means employed.

Mathematical representations may employ different number bases. Thus an n-digit decimal number (base ten), as conventionally written, is a shorthand expression for the quantity.

$$A_n \cdot 10^n + A_{n-1} \cdot 10^{n-1} + \cdots + A_j \cdot 10^j + \cdots + A_0 \cdot 10^0 \quad (1)$$

where the integer coefficients A_j are the digits of the number. Any A_j in the decimal system may take on a value from 0 through 9. When a number is represented to the base X, it is still written as a sequence of digits A_j, but these now are interpreted as meaning

$$A_n \cdot X^n + A_{n-1} \cdot X^{n-1} + \cdots + A_j \cdot X^j + \cdots + A_0 \cdot X^0 \quad (2)$$

and any digit A_j can now take on only the values 0 through $X-1$.

In addition to the decimal notation, this paper will refer to the binary scale of notation; i.e., $X = 2$ in (2 above. Here the only possible digits are 0 and 1. The binary equivalents of the decimal numbers 0 through 15 are shown in Table I.

TABLE I

DECIMAL BINARY EQUIVALENTS

Decimal	Binary	Decimal	Binary
0	0000	8	1000
1	0001	9	1001
2	0010	10	1010
3	0011	11	1011
4	0100	12	1100
5	0101	13	1101
6	0110	14	1110
7	0111	15	1111

The physical representation of a number to the base X requires a physical representation for each of the possible digits (0 through $X-1$) in each of the n-digit columns. That is, each number is denoted by a particular selection from X^n physical states. For a given number base, several physical representations are possible, depending upon the number of temporal and spatial selections used to designate the number. Fig. 2 shows two ways in which the binary number 101101 may be represented. At (a) the number is being transmitted serially on a single wire and the representation is entirely temporal. At (b) the number is being transmitted in parallel on six wires and the representation is entirely spatial. In either case, the digit 1 is represented by the presence of a pulse, and the digit 0 by the absence of a pulse.

Because the rules of arithmetic are simpler in the binary notation than in any other base, this notation is

used in the present machine. Binary-decimal conversion is required at the input and output of the machine.
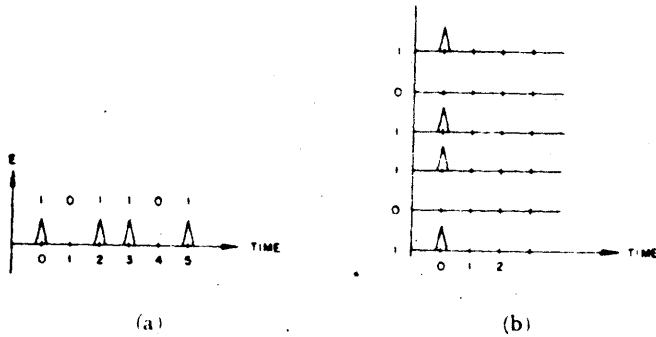

Fig. 2 - Serial and parallel representation of the number 101101. (a) Serial transmission on one wire. (b) Parallel transmission on six wires.

This may be justified in a calculator for scientific problems because of the large amounts of calculation that are done with comparatively few initial data.

When discussing machine operation, it is convenient to speak of a composite pulse group of fixed length as a "word." A word in the present machine contains 45 pulse positions or binary digits which are transmitted between units in a serial manner. Two kinds of words are shown in Fig. 3. These are: (1) a "number word,"
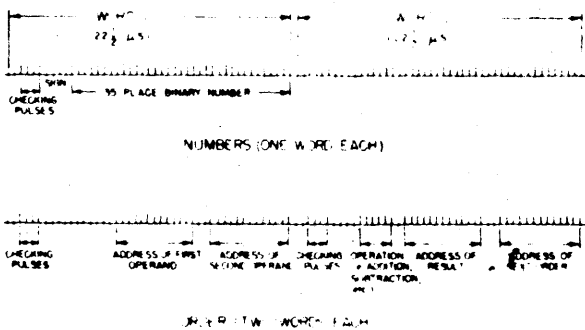

Fig. 3 - Allocation of information in number and order words.

which contains the absolute value of a number, its sign, and auxiliary digits used in checking; and (2) a "command word," which contains coded pulse groups capable of governing the machine operation.

The machine cycle is the basic unit of computing operation, and in this machine is approximately 1 millisecond in duration. During this cycle four distinct events take place. (1) Two operands are selected from the internal memory and are sent to the arithmetic unit. (2) The arithmetic unit performs the desired operation. (3) The result of the operation is sent back to the internal memory. (4) The command which governs the next operation is selected from the memory.

A command is required for each machine cycle and contains all the information necessary for the performance of the cycle, namely, the locations of the operands, the specification of the operation, the location which is to receive the result, and the location of the next com-
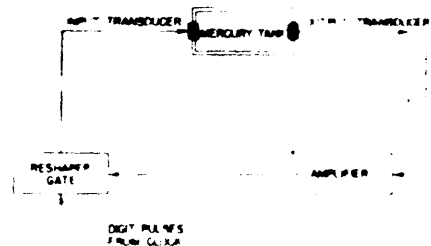
mand. Locations within the memory are termed addresses, and are specified by binary numbers which identify consecutively all of the storage positions of the memory. Operations are also specified by coded pulse groups.

Besides addition, subtraction, etc., several nonarithmetic operations are required during the solution of most problems. The transfer operation serves to transfer a word between different memory locations, or between the internal memory and one of the magnetic memory units. The substitution operation is used to modify a command word by adding or subtracting from one or more of the addresses contained within the command.

The branch operation allows the machine to choose between two computing routines on the basis of the results of past computation. The command governing the branch operation contains the locations of two commands, only one of which is to be chosen to govern the next computing cycle. This choice is determined by the sign of the inequality of the two operands. As an example of the use of the branch operation, consider the solution of a polynomial equation. An approximate root of the equation is calculated by means of computing routine A. The difference between this approximation and the last approximation is obtained. If this difference is greater than some preassigned number, the next command selected is the first command of routine A, and its selection results in the calculation of a nearer approximation. If the difference is less than the pre-established tolerance, the next command selected is the initial command of routine B, which initiates reduction of the degree of the polynomial in preparation for the calculation of the next root.

III. Internal Memory

The internal memory must be capable of storing a large number of words with short access time. The stored information must be easily erasable. In the present machine, the internal memory makes use of the acoustic delay line as the storage mechanism. Fig. 4

Fig. 4 - Delay-line memory unit.

is a block diagram of an acoustic delay line in which mercury is the acoustic medium. Assume that the line contains some pulse configuration at a given instant of time, and that the configuration is propagating down the line at the velocity of sound in mercury. Each pulse received at the output transducer is amplified and

applied to a gate or reshaper which allows a new digit pulse from a continuous pulse source to be fed to the input transducer. Thus, the pulse configuration will circulate around the closed loop indefinitely without progressive degeneration of the pulse shapes or the pulse spacing.

The number of pulse positions which a line contains (i.e., its storage capacity) is proportional to the time delay of the line and the frequency of the continuous pulse source. The former is limited by the attenuation of the mercury and the transducers; the latter, by the bandwidth which can be attained around the circulation path and the dependence of attenuation on frequency.

The access time of the delay line is equal to the delay time. For a given memory capacity, the access time may be decreased by decreasing the length of each line and increasing the total number of lines while the repetition rate is held constant. Reduction obtained in this manner is costly, for the total memory equipment is primarily proportional to the number of circulation paths and is only secondarily influenced by the attenuation per path. The access time can be reduced more effectively by increasing the pulse-repetition rate and shortening the line, while holding constant the number of lines and the pulse capacity of each.

The present machine design makes use of 255 acoustic delay lines, each capable of storing 16 information words and one additional word used for checking. The total memory capacity is, therefore, 4080 words. The delay lines operate at a pulse-repetition rate of 2 Mc, and the digit pulses are amplitude-modulated upon a 30-Mc carrier. The memory access time is about 380 microseconds.

Fig. 5 is a block diagram of a delay line showing the use of additional gates for reading (i.e., taking information from the line), writing (i.e., putting information into the line), and erasing. The erase gate and the write

that the delay time be a constant integral multiple of the period of the continuous pulse source used for reshaping. Because the acoustic velocity in mercury is temperature-dependent, some means of temperature control is required. The temperature coefficient of acoustic velocity is such that a 1 per cent change in delay time results from a temperature change of 30°C. The temperature effect may be expressed more conveniently by the equation

$$\Delta T = \frac{3000\eta}{N} \; °C, \qquad (3)$$

where ΔT is the permissible temperature variation in degrees centigrade, N is the total number of pulse positions in the line, and η is the fraction of a pulse period by which the delay can change and still allow reshaping of the circulating pulses. By using sharp pulses in the reshaping operation, the value of η may be made as high as 0.75 without difficulty. In the present design, each delay line contains 765 pulses, and the corresponding permissible temperature gradient is approximately 3°C.

Temperature gradients in the memory can be controlled, while reasonable equipment accessibility is maintained by subdividing the memory into groups of lines. Each group may consist of several (say, 6 to 10) independent acoustic paths operating within a single container of mercury. The relatively high thermal conductivity of mercury is effective in keeping the gradients between paths small. Each container is then supplied with an independent temperature-control mechanism which maintains the temperature of the pool constant.

Fig. 6 is a photograph of a mercury pool contained in a stainless-steel tank less than 7 inches long, 2¼ inches high, and 2¼ inches wide. Three acoustic paths operate
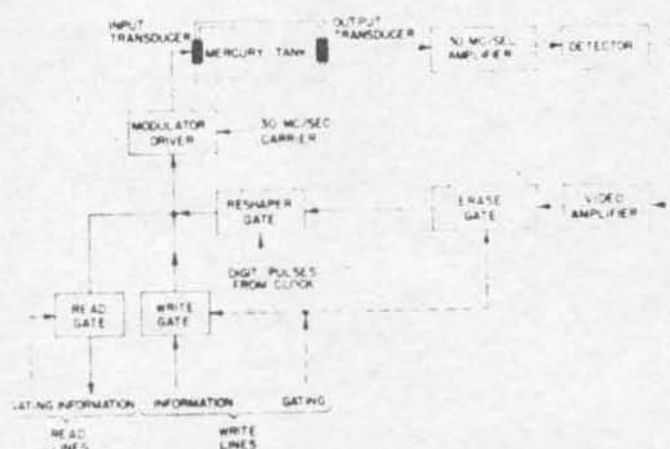

Fig. 5 — Organization of memory-delay-line circuits.

Fig. 6 — Photograph of a mercury pool contained in a stainless-steel tank, with three acoustic paths operating within the pool.

ite are normally connected together, so that a word is erased only when a new word is written into the line.
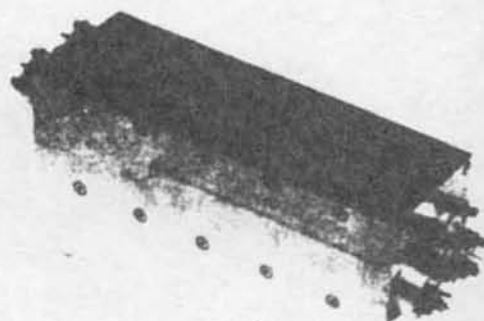
Continuous circulation through a delay line requires

within the pool. Each path is multiple-reflecting and consists of three travels the length of the container. The circulation time of each path is 320 microseconds. This pool has been used in a prototype memory unit

with temperature control used to maintain synchronization between the delay line and a fixed-frequency pulse source.

IV. ARITHMETIC UNIT

The basic arithmetic operation is addition. Other arithmetic operations, though coded as single operations, are compounded of successive additions performed under the local control of the arithmetic unit. Accordingly, the mechanism of addition largely determines the manner in which other operations are carried out. Addition may be performed serially or in parallel. In a serial adder the two operands are added one digit at a time, commencing with the lowest-order digit. In a parallel adder, all of the digits of one operand are simultaneously added to the digits of the other operand, thereby generating all the digits of the sum at once.
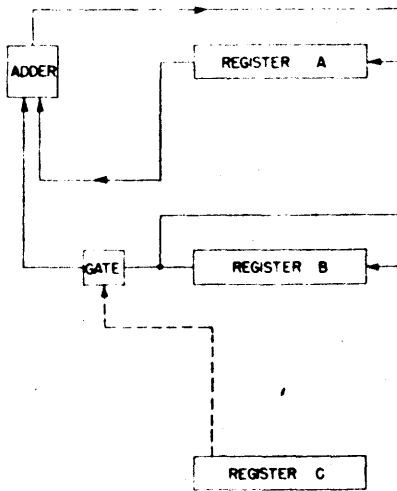

Fig. 7—Serial-adder operation.

Fig. 7 is a block diagram indicating the operation of a serial adder. The two operands are assumed to be in registers A and B which may be delay lines. The successive digits of the operands enter the adder, which generates the digits of the sum and enters these into register A. In multiplication, the multiplicand stands in register B and the multiplier in register C. The partial sums generated during multiplication are stored in register A. The serial adder requires a minimum of equipment. Its chief disadvantage is that the time for addition cannot be less than the circulation time of an operand in its register. The multiplication time is then equal to the product of the addition time and the number of digits used.

Fig. 8 is a block diagram of a parallel adder. The addend stands in register A and the augend in register B. Upon application of a control pulse, all columns of the addend are simultaneously added to the augend and the sum is left standing in the sum-augend register. In principle, the addition is completed in one pulse time, although practical considerations generally require that the process take from 5 to 10 pulse times.

Speed is obtained at the expense of equipment in the parallel adder, for the basic columnar adding circuit must be repeated for each digit column of the addend and augend. A further advantage of the parallel over the serial adder is that the parallel unit requires less complex control circuits to perform operations compounded of repeated additions.
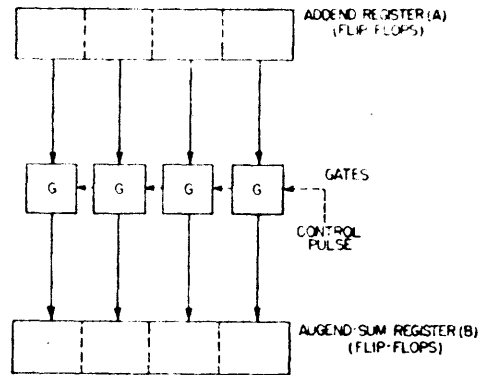

Fig. 8—Parallel-adder operation.

With either type of adder several variations are possible, depending upon the method used to transfer carries from one column to another. For example, a parallel adder may first perform the addition without carries and later add the carries to the result. The addition of carries may generate new carries which must be added in again. Such sequential adding of carries increases the time required for an addition. Simultaneous addition of carries may be performed by adding all carries to the augend in one operation. In this case, the addition speed is limited only by the propagation times of the carry pulses and the add pulses through the necessary circuits.

Before discussing the adder circuits in detail, it is worth while to describe a shift register. This is a simpler device than the parallel adder, but involves similar techniques. The register may be used to convert between serial and parallel number representation, to change the repetition rate of a serially transmitted number, and to act as buffer storage for numbers entering the arithmetic unit.

The register consists of a chain of flip-flops, one for each digit to be stored, as shown in Fig. 9. One plate of each flip-flop is connected to the grid of an adjacent flip-flop through an electrical delay network. Provision is made for applying a reset pulse simultaneously to all flip-flops in the register.

If any configuration of ones and zeros stands in the register and a reset pulse is applied to all flip-flops, those which stand at 1 undergo a change of state. This change produces a pulse at one of the plates which is momentarily stored in the delay network associated with that plate. After the reset pulse has passed, every pulse which entered a delay network emerges and triggers the flip-flop immediately to the right of it. In this manner the configuration of ones and zeros is shifted one column to the right.

In Fig. 9 a three-digit word is shown serially entering

the shift register. A train of reset pulses is applied to the shift register at the same repetition rate as the digits of the entering word. The reset pulses must be advanced, with respect to the applied digit pulses, to avoid interference between the two pulse trains. Once the word
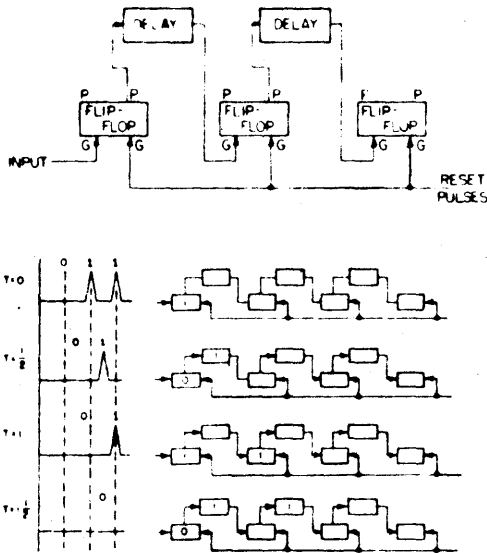

Fig. 9—Shift-register operation.

stands in the register it may be read out in parallel from the plates of the flip-flops, or it may be read out serially at some other repetition rate by applying reset pulses of the desired frequency. Fig. 10 is a circuit diagram of two columns of a shift register which operate reliably at pulse rates up to 2 Mc.
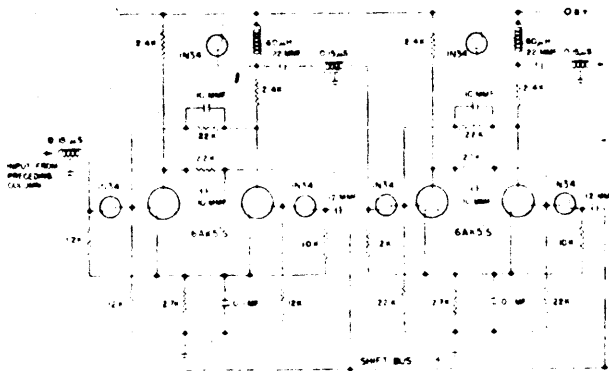

Fig. 10—Shift-register column.

The present machine makes use of a parallel adder with simultaneous carry, as shown in Fig. 11. In this circuit the addition (without carry) of the addend to the augend occurs first, and is followed by the simultaneous addition of all carries. The operands are assumed to stand in the addend (A) and the augend-sum (B) registers. An add control pulse is applied to the gates $G1$. In each column where the addend contains a 1, the control pulse passes through the normally closed gate $G1$ and changes the state of the corresponding column of the augend-sum register. Following the addition without carry, a carry pulse is applied to the nor-

mally closed gates $G2$. Gates $G4$ sense the digits standing in the A and B registers. In each column where a 1 stands in the addend register and a zero in the augend-sum register, gate $G4$ opens gate $G2$. The applied carry pulse passes through $G2$ to an electrical delay network, as well as to gate $G3$, through the phase inverter I. Gate $G3$ is open if a 1 stands in the augend-sum register.
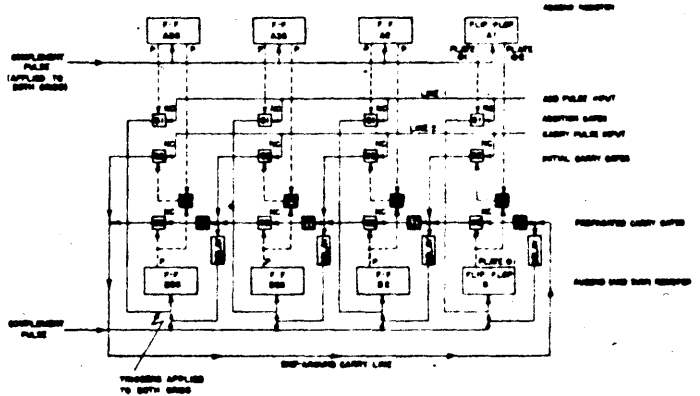

Fig. 11—Parallel adder employing simultaneous carry.

and the carry pulse passes $G3$ to the next higher-order column, etc. Thus, a carry pulse which is initiated in any column may pass several gates $G3$ and be applied to several higher-order columns. Each time a carry pulse passes a gate $G3$, it is applied to the electrical delay network associated with the next column. The propagation of the carry pulses does not immediately cause a change of state in any of the flip-flops, but rather serves to introduce the carry pulses into the appropriate electrical delay networks in accordance with the configurations of the numbers in the A and B registers. After propagation of the carry pulse, all pulses applied to the delay networks emerge and change the states of the associated augend-sum flip-flops. The true sum then stands in the lower register.

Numbers may be placed in the adder registers by the use of shift-register techniques, in which case additional gates and delay circuits are associated with the register flip-flops so that these may function as a shift register during the introduction of a number.

The arithmetic unit of the computer requires numerous control circuits in addition to the basic adder. These are necessary to perform the compound arithmetic operations of multiplication and division, as well as the logical operations described previously.

V. CENTRAL CONTROL

The basic cycle of machine operation requires supervision of the following processes: selection of the operands from the memory, specification of the operation to be performed, disposition of the result, and the selection from memory of the next command. Many possible organizations of the central control unit exist, depending upon the time and manner in which the above steps are carried out. For example: (1) the selection of the two

operands may occur sequentially or simultaneously; (2) the selection of operands may occur while previous operands are being processed in the arithmetic unit, or after the arithmetic process has been completed; (3) the disposition of a result may or may not occur simultaneously with some of the other steps; (4) part of or all of a command may be selected in one step; (5) a command may be selected in accordance with information contained in the previous command, or all commands may be located consecutively in the memory; (6) the control sequence may constitute a fixed cycle in which each step or combination of steps is allocated a fixed amount of time, or a variable cycle in which each step proceeds as soon as the previous step has been completed. There is no known optimum organizational pattern for the control process. Compromises must be made between speed of operation, simplicity of operation, and economy of equipment. By performing a large number of steps in parallel, the speed, cost, and complexity of the machine are all increased.

The control system of the present machine involves fixed-cycle, dual-selection operation. Each command is selected on the basis of information contained within the previous command.

A command consists of two words which are stored in adjacent positions in the internal memory. Referring to Fig. 3, it will be seen that each word of a command contains two address positions and a group of checking pulses. The second command word also contains an operation code. For most operations, address positions 1 and 2 contain the addresses of the two operands, address position 3 contains the address of the result, and address position 4 contains the address of the next command. Two exceptions to the above occur: (1) Certain operations (e.g., transfer) require but one operand. This appears in the first position, and the second position is then vacant. (2) In the branch operation the addresses of two commands appear in the third and fourth positions. The branch operation chooses which of the commands will be used in the next computing cycle.

The fixed-operation cycle is called the machine cycle, and is approximately 1 millisecond in duration. It is composed of three equal parts called major cycles which are equal in duration to the circulation time of a memory delay line, and which are synchronous with the memory circulation. During the first major cycle, the command governing the cycle is selected from the memory, and simultaneously the result of the previous computation is transferred to the memory. The selections made at this time are governed by the third and fourth addresses of the previous command. During the second major cycle, the two operands specified by the new command are selected from the memory, and the operation code is transmitted to the arithmetic unit. During the third major cycle, the arithmetic operation occurs.

From the above discussion it is apparent that the central control unit must contain registers for storing a com-

plete command, and two selection circuits for the simultaneous selection of two memory locations. Since the operands selected from the memory may arrive at central control during any part of the selection cycle, the central control should also have one-word storage registers for holding the operands until they are to be transmitted to the arithmetic unit. A one-word register for the result is also required. The simplified organization of the central control is shown in Fig. 12.
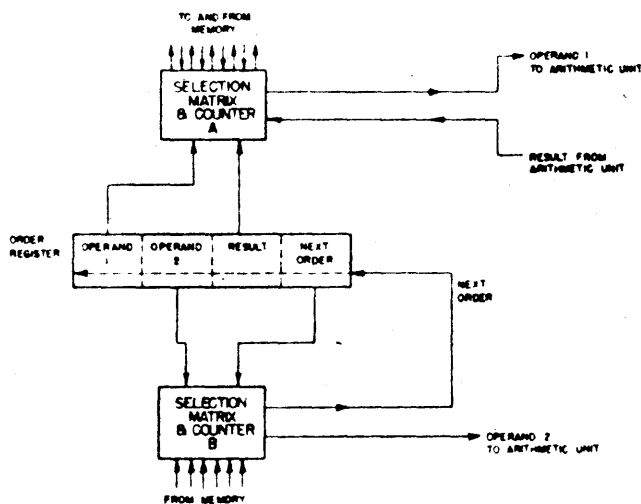

Fig. 12 – Simplified organization of central control.

The operation of central control while making a given selection from the internal memory involves a spatial selection of one of 255 delay lines and a temporal selection of one of 16 word positions within the line. The address of any memory position is specified by a 12-digit binary number, the lower-order four digits of which refer to the temporal selection, while the upper-order. eight digits refer to the spatial selection.

A selection of one of 2^n gating lines in accordance with an n-digit binary number governing the selection can be effected through the use of a diode matrix. Fig. 13 shows an elementary matrix in which a two-digit binary number contained in two flip-flops selects one of four gating tubes. The principle of operation depends
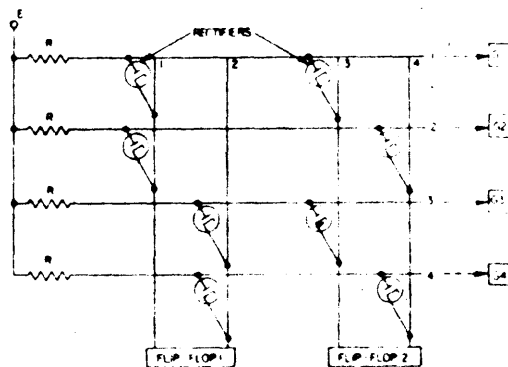

Fig. 13 – Rectifier matrix.

upon the fact that the impedance across a parallel combination of diode gates varies only slightly with the

number of gates closed if at least one gate is closed, but changes abruptly when all gates are opened. The rectifier matrix of Fig. 13 can be extended in both directions to accommodate as many binary digits as desired.

VI. MAGNETIC MEMORY

The magnetic memory units serve to augment the internal memory of the computer, to introduce initial data and commands into the machine, and to record the results of computation. These units make use of magnetic tape as a permanent continuous-storage medium. Each unit has a capacity of approximately 200,000 words, and is capable of operating at a maximum rate of 500 words per second.

Communication between the magnetic-tape units and the other parts of the machine is under the jurisdiction of the central control. Each unit has assigned to it an address which is of the same form as an internal memory address, except that the address in this case includes one of the three operation codes: read, write, and hunt. The read and write codes are always interpreted as applying to the next consecutive word position on the magnetic tape. The hunt code causes the tape unit to hunt for a particular word position, the number of which is supplied by central control.

Because the central control may call upon a magnetic memory unit at highly irregular rates, it is not feasible to propel the tape in response to each individual command. Instead, information is recorded on the tape in blocks of 16 words each and the tape mechanism always operates to read or write an entire block. A mercury-delay-line reservoir consisting of two 16-word delay lines is associated with each tape unit, as shown in Fig. 14. During a sequence of writing operations, the words

soon as one of the delay lines has been emptied by central control, the tape starts and refills the delay line.

The recording medium is $\frac{1}{2}$-inch-wide magnetic tape on which five parallel channels are recorded. Four of these contain information, and the fifth serves as a marker or control channel. The marker channel contains markers which identify the beginning and end of each block of 16 words, as well as binary numbers which consecutively identify the blocks.

The maximum speed of operation of the magnetic memory units depends upon the pulse-repetition rate per channel and the time required for starting and stopping the tape. With operation rates of several hundred words per second, it may be necessary to accelerate the tape at several thousand inches/sec/sec. Fig. 15 shows the arrangement of a tape-drive mechanism which permits rapid acceleration. A magnetic clutch and drive capstan is used to propel a short segment of tape which is in contact with the recording heads. Spring slack absorbers are used to average the motion of the active portion of the tape and to control servomechanisms which drive the take-up reels.
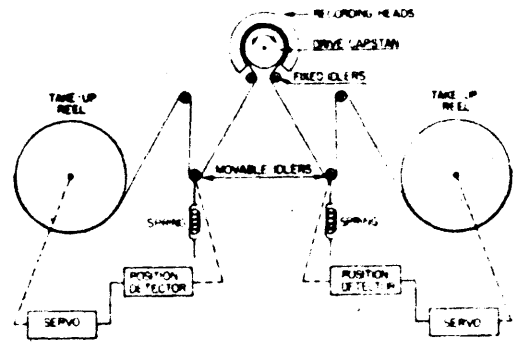
Fig. 15—Input-output drive mechanism.

VII. CHECKING

It is desirable that a computer be self-checking in order to minimize the number of undetected errors. The error-checking mechanism should be highly diagnostic, so that the location of a defective part is indicated whenever an error occurs. By incorporating diagnostic checking equipment in the design, the trouble shooting of equipment failures is greatly facilitated and the per cent of the total time during which the machine is operative is proportionately increased.

Numerous methods for checking digital computers have been suggested. Some of these include the simultaneous operation of duplicate equipments, the repetition of each operation or its inverse on the same equipment, and the use of operational checks which involve the periodic running of a test problem.

Checking by duplication of equipment may be undesirable for several reasons. The total amount of equipment is doubled, thereby doubling the cost as well as the total frequency of failures. Repeating each operation or its inverse reduces computation speed by one-half. It also seems doubtful that checking schemes based on
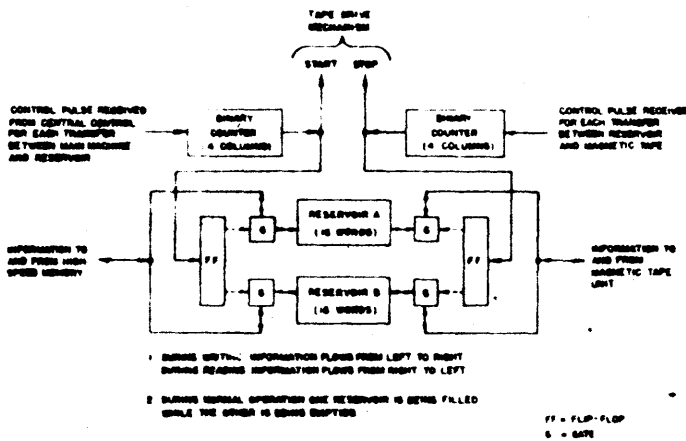

Fig. 14—Organization of magnetic-memory-unit reservoirs.

to be written are accumulated in the consecutive word positions of this reservoir. When 16 words have accumulated in either delay line, the tape mechanism is actuated by local controls, and the contents of the reservoir are transferred to the tape. During a sequence of reading operations, the reservoir is filled from the tape, and as

the repetition of each operation can be made sufficiently diagnostic to indicate the location of a failure. The periodic running of a test problem has many of the disadvantages cited above concerning loss of speed and lack of diagnostic information. The existence of a time lag between an error and its detection makes it difficult to determine the exact point in the computing routine at which failure occurred, and consequently increases the difficulty of resuming operation. It is possible that intermittent errors might escape detection altogether. It may also be difficult or impossible to construct a test problem which completely checks the machine (e.g., tests all storage positions or all control circuits).

A general system of checking developed by Bloch,[1] known as the method of weighted counts, is believed to offer numerous advantages in regard to economy, reliability, speed, and diagnostic ability. A particular adaptation of this theory is used to check memory storage, word transfers, and arithmetic operations in the present machine.

The successive digits of a binary number may be said to be valued or weighted with the value 2^n where n is the number of the column. The conventional value of the number is then the sum of the weights of those columns in which 1's appear. A different number can be obtained by weighting the columns in a different manner. For example, successive columns may be given the weights 1, 2, 4, 1, 2, 4, etc. The new number is the sum of the weights off all columns in which the digit 1 occurs. This new number (or rather, the lowest-order four digits of it) is called the weighted count of the original number. Each number stored in the computer has its weighted count stored with it (note the group of checking pulses in Fig. 3). A number can be checked whenever desired by formulating a new weighted count and checking this for identity with the weighted count carried with the number. Such a check is performed each time a number is transferred from one location to another.

It can be shown that the basic arithmetic processes can be checked by means of the weighted count. Such checks are possible because the sum (difference, product) of the weighted counts of two numbers has a known relation to the weighted count of the sum (difference, product) of the numbers.

The weighted-count system of checking storage and transfers is operative throughout the machine. When initial input data and commands are manually recorded in the problem preparation unit, weighted counts are generated simultaneously with the depression of the recording keys. Thenceforth, each word contains its own weighted count. Each transfer of a word from the time of manual recording up to and including the time when results are printed by a page printer is checked by the weighted-count process.

Additional checking methods are used to verify the selection of memory positions, and the transfer of operation codes. A selection from the memory involves a temporal selection from among the 16 information word positions in each line and a spatial selection from among the 255 delay lines. Each delay line has a seventeenth word position which contains the binary number of the line. Whenever a line is selected, the number of the line is taken from this seventeenth word position, and is compared for identity with the number of the line as it occurs in the command governing the selection. An additional delay line is used to check temporal selections. This line operates in synchronism with the 255 information lines. The 16 information word positions of the extra line contain the binary numbers from zero to fifteen. Each time a word is taken from an information line, a word is simultaneously gated out of the extra delay line, and the number so obtained is compared for identity with the word position as it appears in the command governing the selection. In this way, each selection made from the memory is checked during the selection cycle in which it occurs.

The transmission of operation codes from the central control to the arithmetic unit is checked by incorporating a code generator in the latter unit. For each arithmetic operation the code generator returns an operation code to central control for checking purposes.

Although the checking system is elaborate, in the sense that each function of the computer is checked during each operating cycle, the equipment necessary to provide such checking does not exceed 20 per cent of the total equipment in the computer. The checking equipment has associated with it a set of controls and neon lights which operate to stop the machine in the event of an error and to indicate the location of the fault. The actual numbers and commands which were being processed when the error occurred are displayed to the machine operator. By virtue of these diagnostic aids, a defective part or subassembly can be quickly located and replaced, in many cases without loss of information or the disruption of the computing routine.

BIBLIOGRAPHY

1. R. M. Bloch, R. V. D. Campbell, and M. Ellis, "Logical Design of the Raytheon Computer," Mathematical Tables & Aids to Computation, National Research Council, Washington, D. C., scheduled for publication in vol. 3, no. 24, October, 1948, issue.

2. T. K. Sharpless, "Design of mercury delay lines," *Electronics*, vol. 20, pp. 134–138; November, 1947.

3. L. N. Ridenour, "Radar System Engineering," McGraw-Hill Book Co., New York, N. Y., 1948, pp. 632, 667–671.

4. Computation Laboratory of Harvard University, "Proceedings of a Symposium on Large Scale Digital Calculating Machinery," *Annals* of the Computation Laboratory of Harvard University, Harvard University Press, vol. 16, 1946.

5. H. H. Aiken and G. M. Hopper, "The automatic sequence controlled calculator I, II, III," *Elec. Eng.*, vol. 65, no. 8–11; August–September, 1947.

6. F. L. Alt, "Bell Telephones Laboratories Computer I, II," Mathematical Tables & Aids to Computation, National Research Council, Washington, D. C., vol. 3, no. 21, 22; January and April, 1948.

7. A. Goldstine and H. H. Goldstine, "The Electronic Numerical Integrator (ENIAC)," Math. Tables & Aids to Computation, National Research Council, Washington, D. C., vol. 2. no. 15, July, 1946.