

results in output pulses of different amplitudes, it is possible, by mixing a constant-amplitude negative pulse in the video amplifier, to obtain as a final result a positive pulse for a positive element and a negative pulse for a negative element of the target.

A developmental model of the Memory Tube intended for computer use is shown in Fig. 3. The general shape of the tube, the relative position of the guns and screen-and-target structure can be seen. Since the computer Memory Tube is still in the development stage, it is expected that many improvements in the design of both the tube and the associated circuits will be made before a completely satisfactory performance is achieved. It is hoped that the above brief account of the present state of development will stimulate workers in this field to explore further the potentialities of the memory device which has been made available to them.

ANDREW V. HAEFF

Naval Research Laboratory
Washington, D. C.

¹ A. V. HAEFF, "A memory tube," *Electronics*, Sept. 1947, p. 80-83.

The Logical Design of the Raytheon Computer

The design to be described in what follows is that of a high-speed digital computer for general scientific work. It is intended to meet general specifications issued by the National Bureau of Standards, for whom most of the work was done with the support of the Office of Naval Research.

The final design incorporates mercury delay lines as an internal memory device and magnetic tape as an external storage medium. These features of the design, together with the NBS specifications, dictated most of the machine parameters.

The basic logical units of the computer are: (1) internal memory, (2) external memory, (3) arithmetic unit, (4) central control, which constitute what might be called the high speed part of the machine, and two auxiliary units: (5) problem preparation unit, (6) printer.

The Memory and the Representation of Numbers and Orders.—The computer handles several different kinds of numbers. In standard or "normal" operation, numbers are stored to a precision of 35 binary digits and with a fixed binary point. The normal precision of the basic arithmetic operation is also 35 digits. Numbers of 70 binary digits—i.e. numbers of double-precision—can be stored and manipulated as pairs of standard numbers. Floating-point numbers can also be handled as pairs of standard numbers, one of these being a number between $\frac{1}{4}$ and $\frac{1}{2}$, and the other being an exponent on 2. Since decimal-to-binary and binary-to-decimal conversions are performed in the arithmetic unit, instead of in auxiliary equipment, decimal numbers must also be stored in the internal memory. These are stored in the binary-coded decimal notation to a normal precision of 8 decimal digits. Decimal numbers of double-precision can be handled as two normal numbers.

Not only numbers but coded instructions (which will henceforth be called orders) as well are stored in the internal memory. An order contains enough information to specify a complete arithmetic operation, including

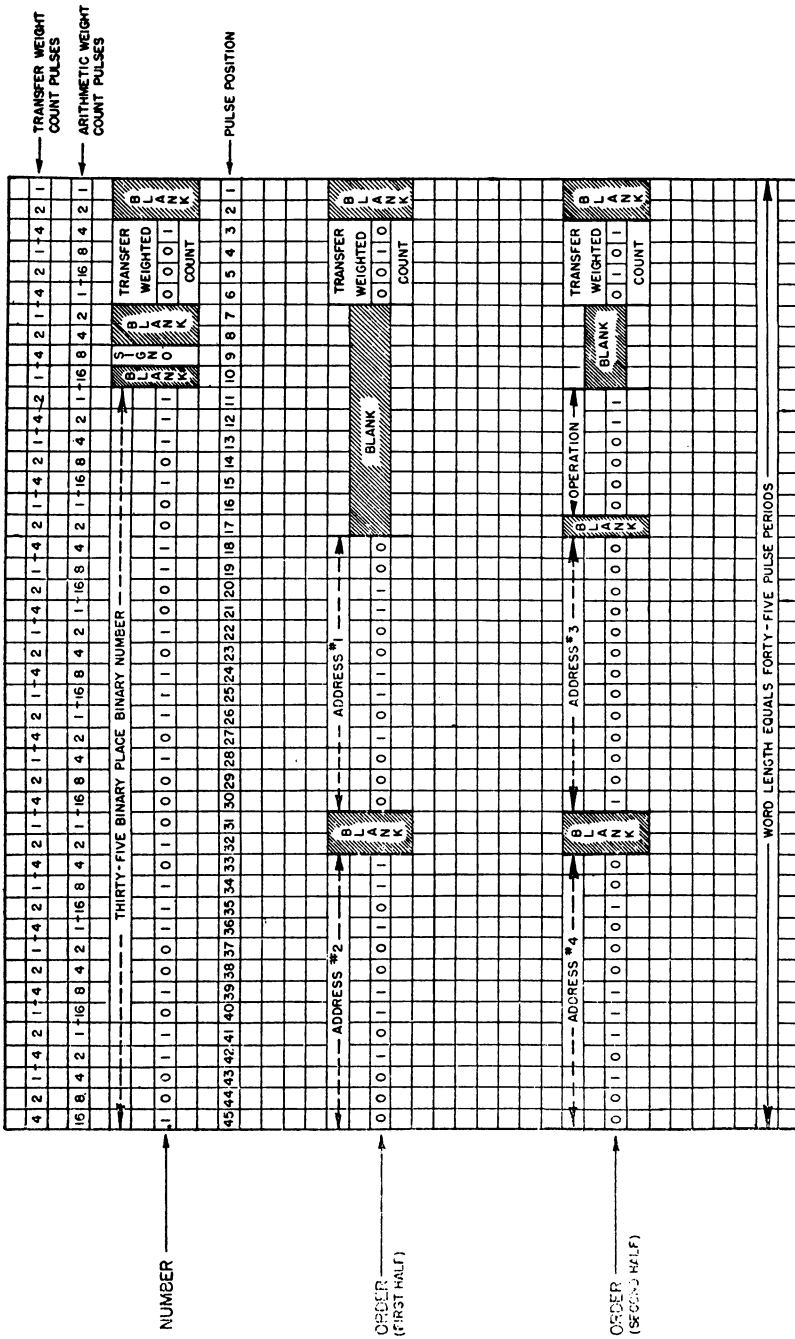


FIG. 4. ALLOCATION OF INFORMATION IN NUMBER- AND ORDER-WORDS.

all the required selections in the memory. Most orders consist of four address codes and one operation code. For these orders, two of the addresses specify the location of the operands in the memory, the third specifies where the result is to be sent, and the fourth where the next order is to be found. In the case of a few operations, the corresponding order does not require the full complement of four addresses. Addresses and operation codes are binary numbers of 13 and 6 binary digits respectively. Because of the presence of a fourth address, there is no preassigned sequence in which orders are normally selected from the memory.

The internal memory stores words, which may represent either numbers or orders. A number-word contains one standard number of 35 binary digits and an algebraic sign, together with a check number to be described subsequently. An order is stored by means of two order-words, one containing two addresses and a check number, and the other two addresses, an operation code, and a check number. The complete length of any word is 45 pulse positions. These are allocated as shown in Fig. 4. The total capacity of the memory can be changed in increments of 1024 words without any changes being made in other units of the calculator, except in the amount of selection equipment included in the central control unit.

The machine will be built to have a memory capacity of 1024 words with design provisions for the ready extension of this capacity to 4096 words. Since the machine design has been arranged to conform with this extended capacity, the ensuing discussion assumes a 4096-word memory.

Three kinds of operations are performed in the internal memory: the writing of new words in the memory, the reading from the memory of information previously written, and the erasure of information so that the same space in the memory can be used to store different words at different times. The logical organization of the machine is such that erasure always occurs automatically just before a word is transferred to the memory. In the four-address order described above (except in the case of a few special operations), addresses nos. 1, 2 and 4 control transfers of words from the memory, and address no. 3 controls a transfer to the memory preceded by an erasure. Words are serially circulated in the memory, the successive digits of a word being all available at the same place, but at successive times. All transfers of words into and out of the memory are therefore done serially; the pulses which represent a word travel in succession along a single "channel."

Each word contains 45 pulse positions, and the pulse repetition rate is 4 megacycles per second. Hence, the duration of one word is 11.25 microseconds; since there are 32 words stored in each line, a complete memory cycle has a duration of 360 microseconds. The time required to select a word in the memory, and to transfer it to the arithmetic unit varies from 2 to 33 word durations.

The external memory is used to supplement the internal memory. Four magnetic-tape units are provided in the design, each capable of storing as many as 10^5 words. Thus, nearly a half million different words can be automatically processed by the machine without any human intervention being required. The different words stored in the external memory have vastly different accessibility times. Any tape can be scanned at the rate of about 500 words per second; hence, in the most unfavorable case it may take as

long as 200 seconds to read a given word out of the external memory. The most efficient use of the external memory occurs when the recorded data can be used in the sequence in which they appear on the tape. If the data must be used in an order different from that in which they are recorded, a mass rearrangement prior to reading might be performed. If this is impossible, the effective computational speed of the machine would conceivably be reduced by a factor as great as 10^4 .

Not only does the external memory supplement the internal memory, but it also is used to introduce initial data—both numbers and orders—into the calculator. These initial data are recorded manually on magnetic tape by means of the problem preparation unit. When the tape so prepared has been manually placed on a magnetic-tape unit, the recorded data are automatically available to the machine.

The external memory also serves as a link between the machine and the output printers. Results obtained by the computer are recorded by means of one of the four magnetic-tape units. The tape containing the data is then manually placed on one of the two automatic printing units.

Since the decimal-binary and binary-decimal conversions are performed in the arithmetic unit, original input and final output numbers handled by the external memory are in the binary-coded decimal notation; all other numbers and all orders are in the binary notation.

Words are recorded on the tape in groups of 32, each group having associated with it a tag for identification purposes. Reading from the tape is also done in blocks of 32. Words are transferred directly between the external memory and the arithmetic unit only and not between the external memory and the internal memory.

Associated with each tape unit are two 32-word delay lines, or reservoirs, which are used as buffers between the main electronic part of the machine and the tape. In recording, the words to be recorded are all sent to one reservoir until it is filled; at this point they are sent to the second reservoir until that in turn is filled, after which they are once again transmitted to the first. As soon as a reservoir is filled, it is emptied automatically onto the tape. Thus, computation and recording can proceed simultaneously. As the machine computes, the results are sent to one reservoir; at the same time, the contents of the other reservoir (computed previously by the machine) are being recorded on the tape. If the computer generates words faster than they can be recorded, the computation is halted from time to time to allow the recording to keep pace; when the opposite condition prevails, the recording is halted periodically. In reading from the tape, the use of the reservoirs is essentially the same as that described above for recording, except that the flow of words is in the opposite direction—from tape to reservoir to arithmetic unit.

Facilities are provided for random hunting on each tape. The process of hunting consists in looking for a block with a given tag, and then in reading that block, or under certain circumstances that block and the next, into the reservoirs. Words are recorded on the tape in four parallel channels.

All storage of words in both the internal and external memories, and all transfers of words are checked by means of a weighted count, i.e., a weighted sum of the digits of the word (*including the algebraic sign* if the word is a number-word). This weighted count is stored with the number or half-order

as part of the word and is called the transfer weighted count. The weights chosen for the sum are the numbers 1, 2, 4, 1, 2, 4, 1, 2, 4, etc., which are assigned to the successive digital positions from right to left. This weighted sum is computed modulo 16 and is then modified by the addition of 1; thus, a number and its weighted count cannot both be zero simultaneously. With this modification, a null word is not a valid word, and the complete failure of a gate or other device controlling the entire transfer channel will be detected. Since the sum is computed modulo 16, only four digital positions instead of seven are required to represent the weighted count.

Arithmetic Unit.—The arithmetic unit is the only unit in the machine capable of generating new numbers or new orders. All of the other machine units have a passive role in the actual computation process, being used merely to introduce operands into the arithmetic unit and to call for the proper arithmetic processes.

Unlike the internal and external memories, the arithmetic unit uses a parallel representation of words, the various digits of the word being stored in separate devices (flip-flops). This permits a considerably higher computing speed than could be obtained in a serial unit. Transfers of words between the memory and the arithmetic unit are accompanied by serial-to-parallel or parallel-to-serial conversions. Arithmetic operations can be performed on either numbers or orders. Built-in operations on numbers include, first, the basic arithmetic operations of addition, subtraction, multiplication, and division; second, transfers of numbers from one memory position to another with at most a change in the algebraic sign of the number; third, number shifts; and fourth, extractions of any preassigned set of digits from a number. Built-in operations on orders (strictly speaking on order-words, i.e., halves of orders) include addition, subtraction, and transfer.

The general way in which these operations are performed can be described in terms of the three registers, or number-storing devices, of which the arithmetic unit is composed. Denote these by the letters A, B, and C. Of the three registers, only B has provision for addition in the form of an accumulator. The number standing in A can be added to that in B, and the sum will appear in B. Numbers stored in registers A, B, and C will be denoted by X_A , X_B , and X_C .

In the process of addition involving standard numbers or order-words, the augend is first read from the memory to B, and the addend from the memory to A. Then the addition $X_A + X_B$ is performed, after which the sum can be read out of B. Negative numbers are stored in the memory as absolute values, with the algebraic sign appended, instead of their being stored as complements. Arithmetic subtraction, however, can only be performed by taking the complement of the subtrahend and adding it to the minuend. Hence, under certain circumstances the augend or the addend must be complemented before the addition takes place, and the sum may have to be complemented before it is sent to the memory. The complement that is used is the complement of $2 - 2^{-35}$; it is the original number subtracted from a number all of whose digits are unity. Thus, the complement is found merely by replacing all 0's by 1's and 1's by 0's. The use of this type of complement requires carry from the extreme left-hand column to the extreme right-hand column under some circumstances. The process of subtraction is similar to that of addition as described above.

In the multiplication of standard numbers, the multiplicand is first read to A and the multiplier to C. Each step of the multiplication then consists of the following: if the extreme right-hand digit of X_C is unity, X_A is added to X_B ; if that digit is zero, null is added to X_B ; then X_B (the partial product) is shifted one column to the right, as is also X_C . The digital columns of X_B which are shifted off the right end of the B register are read into the C register to take the place of the digits of X_C as they are shifted to the right. Thus, at the end of the multiplication, the complete 70-digit product stands in registers B and C. Either 35-digit section of this product, or both sections, can be utilized for subsequent calculation.

In the division of standard numbers, the dividend is transferred to register B, the divisor to register A, and the quotient is formed in register C. The complement of the number in A is first taken. Each step of the division then proceeds as follows: The complement of the divisor is added to X_B . If the sum is positive, unity is added into the right-hand digital position of register C. If the sum is negative, null is added into C, and the complement of the number in A is taken (restoring X_A to its original form). The divisor is then added into B. The complement of the number in A is recomputed. Finally, the remainder in B is shifted one column to the left, as is also the quotient in C. At the end of the division process, both the quotient and the remainder are available.

The operations of addition and subtraction give exact results, and hence require no round-off. If the low-order part of a product is discarded, however, the resulting half-product will generally be inexact. If no compensation were made for the discarded portion, products would be subjected to a mean bias of about one-half unit in the last place retained, i.e., a bias of $\frac{1}{2} \cdot 2^{-35}$. The root-mean-square error would be about $(1/\sqrt{3}) \cdot 2^{-35}$. The occurrence of such a large bias is a particularly serious matter, since repetitive multiplication would lead to excessive resultant values of bias and r.m.s. error. The bias may be reduced by a factor of about 2^{-35} , and the r.m.s. error by a factor of $\frac{1}{2}$ if $\frac{1}{2} \cdot 2^{-35}$ is added to the complete 70-digit product before the right-hand 35 digits are discarded. The use of the above round-off process is optional, multiplication codes being provided for both a rounded and an unrounded product.

The operation of division may also produce an inexact result. If, after 35 digits of the quotient are obtained, the division process is terminated, there is usually a remainder. This is positive (considering only the absolute values of the divisor and the dividend). Merely neglecting the remainder would result in a bias and an r.m.s. error of $\frac{1}{2} \cdot 2^{-35}$ and $1/\sqrt{3} \cdot 2^{-35}$ respectively. It is not convenient to add $\frac{1}{2} \cdot 2^{-35}$ to the quotient to reduce the magnitude of these quantities, since register C, which stores the quotient, is not provided with facilities for addition. Hence, the alternative scheme of merely replacing the right-hand quotient digit by 1 is used. This greatly reduces the bias, but does not change the r.m.s. error.

So far, only arithmetic operations on standard numbers have been described, but double-precision and floating-point operations can be performed also. These are not built-in operations, but are compounded of standard operations which are in some cases slightly modified.

Consider the addition of two double-precision numbers. Each double-precision number is stored as two number-words, it being understood that

the number in one of the words is multiplied by 2^{-35} . In adding $X_1 + X_2 \cdot 2^{-35}$ and $Y_1 + Y_2 \cdot 2^{-35}$, two ordinary additions are performed. First X_2 and Y_2 are added, and the sum is left in the arithmetic unit, having been transferred from register B to C. There may, however, be a carry digit in the units column of $X_2 + Y_2$; this is transferred to the lowest order column of B. Then X_1 and Y_1 are added into register B. It is now necessary only to complete certain carries in order to obtain the double-precision sum. The carries in question occur whenever it is necessary to carry from the extreme left-hand digit of the 70-digit sum to the extreme right-hand digit. (As previously mentioned, such a carry is sometimes required because of the scheme of complementing used.) In order to complete these carries it is necessary first to interchange X_B and X_C , perform certain of the carries, and then interchange them again, performing the remainder. Double-precision subtraction is performed similarly.

In the multiplication of two double-precision numbers, $X_1 + X_2 \cdot 2^{-35}$ and $Y_1 + Y_2 \cdot 2^{-35}$, the products X_1Y_1 , X_1Y_2 , and X_2Y_1 are computed, the first being kept as a double-precision number. These products are then added together by means of double-precision addition.

In double-precision division, the reciprocal of the divisor is first obtained to 35 places by means of a standard division. It is then obtained to 70 places by means of an iteration formula. Note that double-precision operations must be used here. Finally, the double-precision reciprocal is multiplied by the double-precision dividend to obtain the result.

The floating-point operations comprise a second type of nonstandard arithmetic operation. These processes deal with numbers X of the form $X_0 \cdot 2^P$ where $\frac{1}{4} \leq X_0 < \frac{1}{2}$ and P is an integer. X is stored in two words, one containing X_0 , and the other $P \cdot k$, where $k = 2^{-35}$. As illustrations of floating-point operations of the proposed calculator, addition and multiplication will be discussed.

To add the two numbers $X = X_0 \cdot 2^P$ and $Y = Y_0 \cdot 2^Q$ to obtain $Z = Z_0 \cdot 2^R$, where $\frac{1}{4} \leq Z_0 < \frac{1}{2}$, the machine must perform these processes: (1) Compare Pk and Qk , and compute $-|Pk - Qk|$. This is done by means of a normal branch. (2) If $Pk \geq Qk$, (a) use a controlled shift operation to obtain $Y_0' = Y_0 \cdot 2^{-|P-Q|}$. (b) Then add X_0 and Y_0' by means of a special addition, viz., addition (floating). At the end of this addition the quantities Z_0 and Sk have been computed, where $X_0 + Y_0' = Z_0 \cdot 2^S$ and $\frac{1}{4} \leq Z_0 < \frac{1}{2}$. (c) Compute $Rk = Pk + Sk$ by ordinary addition. (3) If $Pk < Qk$, (a) use a controlled-shift operation to obtain $X_0' = X_0 \cdot 2^{-|P-Q|}$. (b) Then add X_0' and Y_0 by means of addition (floating); at the end of this addition the sum stands in the form $Z_0 \cdot 2^S$, where $\frac{1}{4} \leq Z_0 < \frac{1}{2}$. (c) Compute $Rk = Qk + Sk$ by ordinary addition.

The floating-point multiplication of X and Y to obtain Z is performed as follows: (1) Use the multiplication (floating) operation to compute Z_0 and Sk , where $Z_0 \cdot 2^S = X_0 \cdot Y_0$ and $\frac{1}{4} \leq Z_0 < \frac{1}{2}$. (2) Compute $Tk = Pk + Qk$ by ordinary addition. (3) Compute $Rk = Sk + Tk$ by ordinary addition.

Several operations other than the four basic operations of arithmetic can be performed on numbers or orders. Transfers of numbers or orders involve the arithmetic unit only in an incidental manner. In the case of numbers, the algebraic sign can be changed during the transfer. Thus, if X is the number extracted from the memory, and Z is the number returned to the

memory, any one of four relationships can hold between X and Z : $Z = X$, $Z = |X|$, $Z = -|X|$, $Z = YX/|Y|$. In the last case, Y is a control number which is used to govern the sign of Z . (As an application of the fourth type of transfer consider the computation of $\sin X$ for $-A < X < A$ using only a table of $\sin X$ for $0 \leq X < A$. Compute $\sin |X|$, and then $\sin X = X \sin |X|/|X|$).

Three types of shift operations are provided: (1) Shift factor (normal)—The nonzero number X is given in standard form, viz., $2^{-35} \leq |X| < 1$. X_0 and $P \cdot 2^{-35}$ are obtained as two standard numbers such that $\frac{1}{4} \leq X_0 < \frac{1}{2}$, and P is an integer. (This is useful in computing logarithms, and in converting numbers from standard to floating-point form.) (2) Shift factor (square root).—The non-zero number X is given in standard form. X_0 and $P \cdot 2^{-35}$ are obtained as two standard numbers such that $\frac{1}{4} \leq X_0 < 1$, and P is an even integer. (This is useful in computing square roots.) (3) Controlled shift.—Two standard numbers X_0 and $P \cdot 2^{-35}$ are given. $X = X_0 \cdot 2^P$ is obtained as a standard number. This operation is the inverse of (1). (It is involved in computing square roots and in converting numbers from the floating-point to the standard form.)

The extraction of digits from a number X is accomplished by utilizing a preassigned control number Y containing a 1 in every digital position in which the extraction is to be performed. The result of extraction is then a number Z containing a 1 only in each column in which both X and Y contain a 1.

Operations on order-words include addition, subtraction, and transfer, all of which are performed on order-words exactly as on numbers; in fact, the arithmetic unit does not distinguish between order-words and number-words. Other arithmetic operations on order-words are not excluded, but would probably serve no useful purpose. It may not be quite obvious why one wishes to perform additions or subtractions involving orders. This type of operation is useful under two conditions. First, assume that it is not possible to specify one of the addresses in a particular order in advance, because the address depends upon a partial result obtained previously in the calculation. It is necessary, then, to be able to insert the address by addition as soon as it is known. Second, it is often advisable to use one order repeatedly for different operations which may require the insertion of different addresses. Using one order many times instead of many different orders conserves space in the internal memory which may be needed for other purposes. The coded instructions for a matrix multiplication, for example, embody but a single multiplication order. The addresses of the multiplication order are modified in accordance with the memory locations which contain the elements of the matrices. Using this scheme, the entire multiplication of a matrix of n rows and m columns by a matrix of m rows and p columns, requires only 16 orders, regardless of the size of the matrices.

The branch is one of the most useful of all operations. It permits the machine to select alternative computing routines. Normally the fourth address of each order specifies the location in the memory in which the next order is to be found. The branch (normal), however, selects for the address of the next order either the third or the fourth address, depending on the sign of a difference $X - Y$. If $X - Y \geq 0$, the third address is chosen, otherwise the fourth. An example of the utility of the branch is manifest in

the floating-point operations described above, where there are many alternative computing routines needed. A second kind of branch is available in equality sensing. Here $X = Y$ selects the third address, and $X \neq Y$, the fourth. In either type of branch operation, the quantity $-|X - Y|$ is available and can be read out subsequently if needed.

Arithmetic operations are checked by means of the arithmetic weighted count. This is a check number, associated with each operand and each result, and defined as the weighted sum of the digits of the number. The assigned weights are 1, 2, 4, 8, 16, 1, 2, 4, 8, 16, etc. Note that these weights differ from those assigned to the transfer weighted count described earlier. The algebraic sign of a number is not included in this weighted count since signs are checked by another method.

These counts are used to check the basic operations¹ $X + Y = Z$, $XY = Z$, and $X/Y = Z + R/Y$ by means of the identities:

$$\begin{aligned} (|Z_c - (X_c + Y_c)| + 31)_c &\equiv 31 && \text{(addition)} \\ (|(X_c Y_c)_c - Z_c| + 31)_c &\equiv 31 && \text{(multiplication)} \\ (|(Y_c Z_c)_c + R_c - X_c| + 31)_c &\equiv 31 && \text{(division)}. \end{aligned}$$

In the above identities, the subscript c indicates the arithmetic weighted count of the quantity to which it is affixed. In the last identity, R is the remainder.

To illustrate how the above relations are used by the machine, consider the addition check as an example. The sum $Z = X + Y$ is computed as described previously. Next X_c , Y_c , and Z_c are computed by means of an auxiliary arithmetic unit, the weighted count is taken, and the presence of 31 is determined by sensing.

Central Control.—The central control makes use of orders extracted from the memory to govern the machine. It contains facilities for making selections in the internal memory in accordance with addresses; for initiating either reading from the memory or writing into it, depending upon the position of the address in the order; and for initiating the proper arithmetic process as dictated by the operation code.

The operation of the machine is partly synchronous and partly asynchronous. The internal memory is completely synchronous, all pulses being timed from a central clock. The operations in the arithmetic unit are largely asynchronous. A complete machine cycle—viz., the time required to execute one complete order—must be a multiple of the memory word-cycle time of $45 \cdot \frac{1}{4} = 11.25$ microseconds. Memory selections and arithmetic operations are allocated only as much time as they actually require, to within one word-time; unnecessary memory selections are avoided. Thus, if the result of one arithmetic process is to be used solely as an operand in the next, it is not returned to the memory at all. Because of this so-called variable-cycle operation, certain arrangements of numbers and orders in the memory have distinct advantages for any given problem. Considerations of efficiency and logical planning dictate that, whenever it can be reasonably arranged, the result of an operation should be used as an operand in the succeeding operation; such forethought will often lead to a substantial reduction in computing time.

The nature of the central control can best be understood by reference to a list of the operations which it performs during a complete machine

cycle. These are :

- (1) The first operand is selected under control of the first address.
- (2) The line and word locations of the first operand in the internal memory are checked.
- (3) The second operand is selected under control of the second address.
- (4) Line and word locations of the second operand are checked.
- (5) Transfer count check of the first order-word is made.
- (6) Operational code is selected.
- (7) Check-back from arithmetic unit is made to insure correct operation selection.
- (8) End of arithmetic operation signals selection of memory position for result.
- (9) Line and word locations of the position about to receive the result are checked.
- (10) Selection of the new set of order-words is begun.
- (11) The line and word locations of the new order-words are checked.
- (12) Transfer count check of the second (present) order-word is made.

As explained earlier, all four of the addresses may not necessarily be present in an order; consequently, some of the above operations may be omitted.

The Problem Preparation Unit and the Printer: The problem preparation unit is used for the manual recording of problem input data on a magnetic tape by means of a keyboard. Either decimal numbers or binary numbers can be recorded on the tape. Ordinary decimal numbers set into the keyboard are recorded as binary-coded decimal numbers on the tape. Binary numbers are set into the keyboard in the octal (radix 8) notation. In this latter notation, each digital position corresponds to three binary digits. It can be seen that the conversions octal-binary and binary-octal are very elementary. The octal system is to be preferred to the binary since in the former notation only a third as many digits are required to represent a number.

All of the data which are recorded on the tape can be printed by means of an input page printer associated with the problem preparation unit. Thus a permanent, visually acceptable record is obtained of the information which is being introduced into the machine. The unit contains a device for generating transfer weighted counts since these must also be recorded on the magnetic tape.

Output numerical data recorded on magnetic tape by the machine are sent to either of two output printers. These data are normally in decimal form, though binary numbers could also be printed (actually in the octal notation rather than binary). Standard page printers, operating at speeds in excess of seven characters per second, can conveniently be used.

R. M. BLOCH, R. V. D. CAMPBELL & M. ELLIS

The Raytheon Manufacturing Co.
Waltham, Mass.

¹ Subtraction, which is in reality nothing more than an addition after appropriate complementing, is checked by means of the addition identity; the complementing process is independently checked.