represent the logical structure of the operations to be performed (i.e., the decision points, the sequence of operations following each alternative, the convergence points, and the parallel operations).

2. It directs to each state the timing signals and service-unit status signals required for initiation and termination.

3. It connects each active state to drive one or more service units, with the following possibilities:

a. Several states may drive the same service unit through buffers.

b. A control state may either drive on a service unit directly, or may gate the output of one service unit to trigger another.

c. A control state may drive any number of service units; if a data transfer is required, the state may have to energize all gates required to complete the data paths when the service unit triggered is not restricted to a specific data path.

d. A control state also may do no external work, but act only as a synchronizing or delay state.

The counter shown in the foregoing example will be used in other routines. Some uses may be identical, but the counter may be pieced together with other counters in different arrangements. Some uses may be similar; for example, the work of one state may be inhibited by transferring its output bus to the zero level with a relay contact. In other uses, the majority of the input and output leads may be switched. An initial survey of the number of operations performed in consecutive groupings led to the switching of counters in blocks of four states each. The drum posting program unit handles its 26 routines with 19 counter blocks and 24 flip-flops. The longest routine uses eight counter blocks and 20 flip-flops.

Error-Checking Features

The fact that ERMA is a bank accounting machine and operates on-line means that errors are not merely inconvenient. Because all error cannot be prevented, and because ERMA must have all accounts in perfect balance by the end of the working day, errors must be found and corrected as quickly as possible. Since the best time to correct an error is when the paper document is in the hands of the operator, error checking should not be postponed. Because of the on-line nature of its operation, downtime must be minimized.

The following are some general techniques utilized in error-checking the drum-posting area:

1. Parity-check monitors are located at the output of all registers and drum-read amplifiers and may be switched onto commonly used busses. Each binary-coded decimal digit carries an even-parity redundancy bit.

2. Every addition or subtraction is followed by the opposite operation, using the sum read back from its permanent store.

3. All non-arithmetic postings, such as "temporary storage" write-on, are followed by reading back from the drum and comparing with the original items. If the source is a keyboard, the comparison is made using duplicate contacts on the key stems.

4. For certain lists, the keyboard is used as an output printer. Here, each key setup is verified by reading the key contacts back for comparison with the original source.

Summary

The ERMA computer is neither a stored program computer nor a plugboard computer, but it does contain some features of each. It is like a plugboard machine in that its program steps are wired in, and it is like a stored programmed machine in that each mode is called up by a coded instruction, which, however, is not subject to modification by the program.

It is a special purpose machine in that its detailed logical design directly reflects the external operational requirements. It needs a very large drum store to perform its operations on-line; the volume of data handled is large; many steps are required for processing each item and for accuracy checks; the data on the various files is essentially random, and the time allowed is short. Taken together, these conditions require performing numerous operations simultaneously rather than serially; each operation, however, can be performed by a simple unit operating at the moderate drum clock rate. Establishing service units which are unchanged in the various machine modes and switching the interconnections of the central control circuits to change programs provides a relatively inexpensive way of conducting the various modes of the data handling processes.

The Logical Design of a 1-Microsecond Parallel Adder Using 1-Megacycle Circuitry

A. WEINBERGER J. L. SMITH

Synopsis: The logical design of a parallel adder is developed which is capable of adding two 53-bit numbers in 1 microsecond. The design makes use of basically the same 1-megacycle circuitry which has been used successfully in the National Bureau of Standards' SEAC and DYSEAC computers. An analysis of the functional relationships of the carry digits to the augend and addend digits shows that it is feasible to form many carries simultaneously at the expense of relatively few components. The Boolean expressions for many successive carry digits can be expanded as explicit functions of some one lower-order carry, and of the relevant augend and addend digits. These somewhat complicated expressions are simplified by making substitutions for the common terms and factors they contain. These common terms and factors, called auxiliary carry functions, are implemented separately. All functional forms fit within the wide limits of gating complexity allowed by the type of circuitry to be used.

A. Weinberger and J. L. Smith are with the National Bureau of Standards, Washington, D. C.

THE development at the National Bureau of Standards of the diode capacitor memory,[1,2] which is capable of being read or written into at the rate of one word per microscecond, has made it worth while to build devices capable of processing information at comparable rates. Since the basic micro-operation common to most arithmetic processes is the adding together of two numbers, it seemed reasonable to design an adder having a cycle time no greater than 1 microsecond.

The major timing bind in an adder is in the production of carries, and in this paper the problem is attacked from the standpoint of logical organization. Although

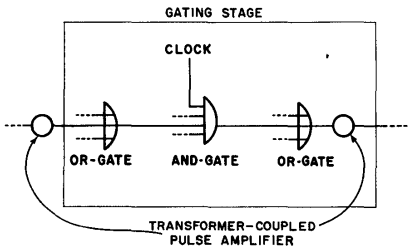

Fig. 1. One stage of SEAC-type circuitry

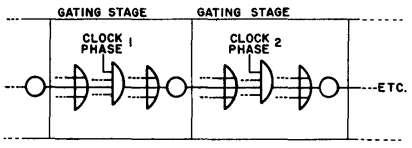

Fig. 2. Gating stages clocked with different clock phases

elsewhere work is being done on this subject using newer and faster basic circuit elements, the analyses to be described show that it is both feasible and economical to achieve 1-microsecond addition times for 53-bit words using the 1-megacycle circuitry which has been successfully utilized in SEAC[3] and DYSEAC.[4,5]

The increased complexity of the logic of this adder necessitated the extensive used of Boolean algebra in arriving at the design itself. Because the procedure used in developing the final design is an interesting example of the practical application of Boolean algebra, the actual logic of the design process is described in considerable detail.

Before discussing the adder, a brief description of the logical capabilities of the SEAC circuitry[6] is in order. As shown in Fig. 1, the basic electronic unit consists essentially of three levels of diode gates in an OR-AND-OR logical array followed by a transformer-coupled pulse amplifier. The rate at which successive pulses pass through such a stage is determined by the clock frequency which is, in this case, 1 megacycle per second. The transit time of a pulse through a stage, however, is much less than 1 microsecond. For this reason, the clock pulses are made available in several phases. The way in which different stages may be controlled by clock pulses of different phases is illustrated in Fig. 2. In SEAC, for example, 1-megacycle clock pulses are available in 3 phases, 1/3 microsecond apart. In DYSEAC, 4-phase clock pulses are used, while in the adder to be described a 5-phase clock is used. Fig. 3 shows graphically these timing relationships for SEAC. Signals resulting from different

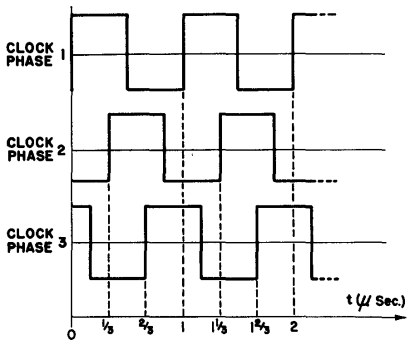

Fig. 3. Time relationships among SEAC clock phases

stages clocked at different times must be synchronized by means of electrical delay lines before they are gated in a common stage as shown in Fig. 4. Both positive and negative signals are available from a stage, the negative signals being used for inhibiting. (See Fig 5.)

The maximum gating complexity used for a stage in the adder to be described is essentially the same as that employed in the packaged building blocks used in constructing DYSEAC, and, therefore, in the OR-AND-OR gating configuration of a stage, up to four AND-gates, and up to six inputs to an AND-gate, are permissible.

Boolean notation of the sort described by Richards[7] will be used hereafter to describe the gating configurations. In Fig. 6 is shown a typical gating stage and the corresponding Boolean expression for the output in terms of the inputs. There are three terms in the expression, each one corresponding to an AND-gate; the first term, $(A + B)\bar{C}\bar{D}EF$, corresponds to the top AND-gate, the second term, $(G + H)I$ corresponds to the middle AND-gate, and the last term, $J(K + L + M)N$, corresponds to the bottom AND-gate. The factors of a term represent the inputs to the corresponding AND-gate. For example, the five factors of the first term $(A + B)$, $\bar{C}$, $\bar{D}$, E, and F, correspond to five inputs to the top AND-gate. Whenever a factor consists of more than one term, it is represented by an OR-gate. For example, the factor $(A + B)$ of the first term corresponds to the 2-input OR-gate of the top AND-gate. A factor could also be a negative or inhibit signal, and in this case it is denoted by a bar on top; e.g., $\bar{C}$ and $\bar{D}$ are two factors of the first term corresponding to the two negative signals which may inhibit the top AND-gate. For the sake of simplicity in the discussion of the Boolean expressions which follow, no distinction is made between delayed and undelayed signals.

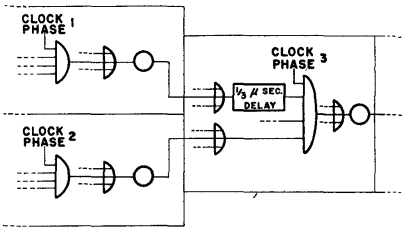

Fig. 4. Synchronizing by means of electrical delay line

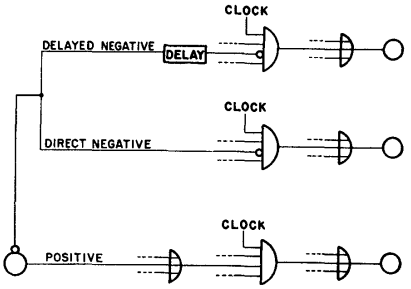

Fig. 5. Use of negative signals for inhibiting

Sequential Carry Propagation

Let

$$A = A_n \times 2^{n-1} + A_{n-1} \times 2^{n-2} + \ldots + A_2 \times 2^1 + A_1 \times 2^0 = \text{augend}$$

$$B = B_n \times 2^{n-1} + B_{n-1} \times 2^{n-2} + \ldots + B_2 \times 2^1 + B_1 \times 2^0 = \text{addend}$$

$$S = S_n \times 2^{n-1} + S_{n-1} \times 2^{n-2} + \ldots + S_2 \times 2^1 + S_1 \times 2^0 = \text{sum}$$

C = carry digit

The well-known rules for forming the sum and carry digits are presented in the form of a function table (Table I).

Table I. Function Table for Binary Addition

Augend.....	A_k	0	0	0	0	1	1	1	1
Addend.....	B_k	0	0	1	1	0	0	1	1
Previous Carry...	C_{k-1}	0	1	0	1	0	1	0	1
Sum.....	S_k	0	1	1	0	1	0	0	1
Carry.....	C_k	0	0	0	1	0	1	1	1

From these, the binary sum and carry can be expressed in Boolean notation as follows:

$$S_k = \bar{A}_k\bar{B}_kC_{k-1} + \bar{A}_kB_k\bar{C}_{k-1} + A_k\bar{B}_k\bar{C}_{k-1} + A_kB_kC_{k-1} \quad (1)$$

$$\begin{aligned} C_k &= \bar{A}_kB_kC_{k-1} + A_k\bar{B}_kC_{k-1} + A_kB_k\bar{C}_{k-1} + A_kB_kC_{k-1} \\ &= A_kB_k + A_kC_{k-1} + B_kC_{k-1} \\ &= (A_k + B_k)(A_k + C_{k-1})(B_k + C_{k-1}) \\ &= A_kB_k + (A_k + B_k)C_{k-1} \end{aligned} \quad (2)$$

Equations 2 show how the carry function, C_k, can be reduced from four terms of three factors each (corresponding to four

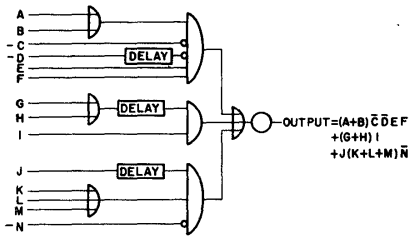

Fig. 6. Typical gating stage and corresponding Boolean expression

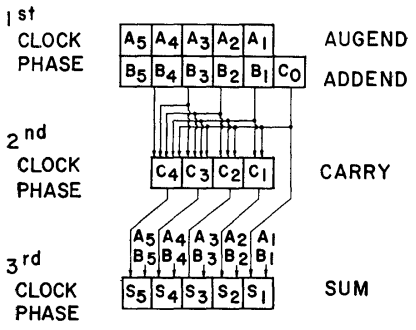

Fig. 7. 5-bit parallel binary adder

AND-gates with three inputs each), as shown in the top line of equations 2, to three alternative forms each involving fewer terms and factors.

Since the expression for S_k in equation 1 can be implemented in one gating stage, any sum digit can be made available during the clock phase immediately following the formation of the preceding carry, C_{k-1}. The speed with which successive sum digits are formed is therefore determined by the speed with which successive carries are generated.

Since each carry is explicitly dependent upon the immediately preceding one in equations 2, successive carries can be generated one clock phase apart, i.e., at the rate determined by the time interval between stages. Using any one of the equations in equations 2, if C_1 is formed during the first clock phase, C_2 can be formed during the second clock phase, C_3 during the third clock phase, etc.

Simultaneous Carry Generation

It will now be shown how C_k can be expanded so as to be independent of the previous carry. For the moment the dependence of a carry upon the appropriate augend and addend digits will be neglected. Then, from equations 2, C_k is a function of C_{k-1}, C_{k-1} is a function of C_{k-2}, etc., so that C_k can be expressed as a function of C_{k-2}. In fact, C_k can be further expanded in this fashion until it is a function of C_{k-3}, then of C_{k-4}, etc. The

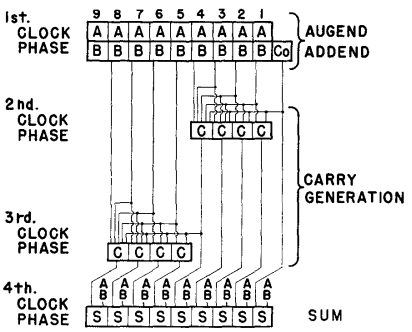

Fig. 8. 9-bit parallel binary adder

limit to this expansion is set by the limit to the gating complexity permitted with the circuitry.

The last of equations 2 is particularly important because of the ease with which it can be expanded so that C_k will be independent of a number of previous carries. Equations 3 show how each additional order of expansion adds only one term to the expression for C_k and only one factor to the largest of the terms.

$$\begin{aligned}C_k &= A_kB_k+(A_k+B_k)C_{k-1}\\ &= A_kB_k+(A_k+B_k)A_{k-1}B_{k-1}+\\ &\quad (A_k+B_k)(A_{k-1}+B_{k-1})\times C_{k-2}\\ &= A_kB_k+(A_k+B_k)A_{k-1}B_{k-1}+\\ &\quad (A_k+B_k)(A_{k-1}+B_{k-1})A_{k-2}B_{k-2}+\\ &\quad (A_k+B_k)(A_{k-1}+B_{k-1})\\ &\quad (A_{k-2}+B_{k-2})C_{k-3}\\ &= \text{etc.} \qquad (3)\end{aligned}$$

Applying the foregoing method for expanding the carry function, how many successive carries can be generated simultaneously is next determined. Beginning with the least significant carry, C_1, four successive carries are shown in equations 4 to be functions of C_0 and to consist of no more terms and factors than can be implemented by the physical gating structures outlined previously.

$$\begin{aligned}C_1 &= A_1B_1+(A_1+B_1)C_0\\ C_2 &= A_2B_2+(A_2+B_2)A_1B_1+\\ &\quad (A_2+B_2)(A_1+B_1)C_0\\ C_3 &= A_3B_3+(A_3+B_3)A_2B_2+\\ &\quad (A_3+B_3)(A_2+B_2)A_1B_1+\\ &\quad (A_3+B_3)(A_2+B_2)(A_1+B_1)C_0\\ C_4 &= A_4B_4+(A_4+B_4)A_3B_3+\\ &\quad (A_4+B_4)(A_3+B_3)A_2B_2+\\ &\quad (A_4+B_4)(A_3+B_3)(A_2+B_2)A_1B_1+\\ &\quad (A_4+B_4)(A_3+B_3)(A_2+B_2)\times\\ &\quad (A_1+B_1)C_0\\ &= A_4B_4+(A_4+B_4)A_3B_3+\\ &\quad (A_4+B_4)(A_3+B_3)A_2B_2+\\ &\quad (A_4+B_4)(A_3+B_3)(A_2+B_2)\times\\ &\quad (A_1+B_1)(A_1+C_0)(B_1+C_0) \qquad (4)\end{aligned}$$

It can be seen that the first expression for C_4 in equations 4 consists of five terms and therefore cannot be formed in one gating stage in this manner. However,

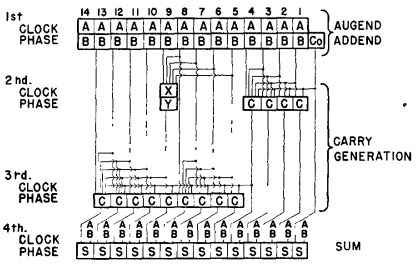

Fig. 9. 14-bit parallel binary adder

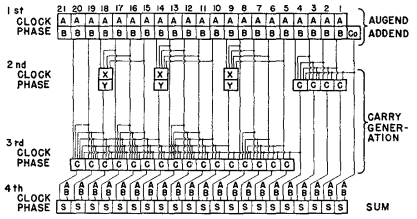

Fig. 10. 21-bit parallel binary adder

the last two terms can be combined into one term by means of equation 5,

$$A_1B_1+(A_1+B_1)C_0=(A_1+B_1)(A_1+C_0)\times (B_1+C_0) \qquad (5)$$

thereby permitting C_4 to be formed in one gating stage as shown in the second expression for C_4 in equations 4.

(C_0 is equivalent to a digit which, together with the least significant augend and addend digits, forms the least significant sum digit, thus:

$$S_1=\bar{A}_1\bar{B}_1C_0+\bar{A}_1B_1\bar{C}_0+A_1\bar{B}_1\bar{C}_0+A_1B_1C_0. \qquad (6)$$

C_0 is used during addition cycles requiring the adding of 1 to the sum, such as the adding of negative numbers in "ones-complement" form.)

Fig. 7 shows in block-diagram form the five least significant digits of a parallel adder utilizing the above principle. Note that the carries C_1 through C_4 are obtained during the clock phase following that of the augend and addend digits. Also, the least significant sum digit, S_1, can actually be obtained during the second clock phase according to equation 6 because it is a function of A_1, B_1, and C_0, all three of which are available during the first clock phase. However, it is desirable in this parallel adder to obtain all of the sum digits at the same time. Consequently, delaying S_1 to occur at the same clock phase as the other sum digits is arranged.

The succeeding four carries, C_5 through C_8, can be formed one clock phase later in a similar fashion using C_4 as the last previous carry, as shown in equations 7. In the case of C_8, the reduction from five to

four terms is made by combining the first two terms instead of the last two as in the expression for C_4 in equations 4, in order to stay within the circuitry limitations for OR-gating delayed signals. (All signals passing through the same OR-gate must originate at the same time.)

$$C_5 = A_5B_5 + (A_5+B_5)C_4$$
$$C_6 = A_6B_6 + (A_6+B_6)A_5B_5 + (A_6+B_6)(A_5+B_5)C_4$$
$$C_7 = A_7B_7 + (A_7+B_7)A_6B_6 + (A_7+B_7)(A_6+B_6)A_5B_5 + (A_7+B_7)(A_6+B_6)(A_5+B_5)C_4$$
$$C_8 = A_8B_8 + (A_8+B_8)A_7B_7 + (A_8+B_8)(A_7+B_7)A_6B_6 + (A_8+B_8)(A_7+B_7)(A_6+B_6)A_5B_5 + (A_8+B_8)(A_7+B_7)(A_6+B_6)\times(A_5+B_5)C_4$$
$$= (A_8+B_8)(A_8+A_7)(A_8+B_7)(B_8+A_7)\times(B_8+B_7) + (A_8+B_8)(A_7+B_7)A_6B_6 + (A_8+B_8)(A_7+B_7)(A_6+B_6)A_5B_5 + (A_8+B_8)(A_7+B_7)(A_6+B_6)\times(A_5+B_5)C_4 \quad (7)$$

Fig. 8 extends the previous block diagram to include a parallel adder accommodating nine binary digits. Again, the formation of the sum digits S_1 through S_5 is delayed to coincide with the rest of the sum digits.

Use of Auxiliary Carry Functions

Of signal importance is the use made of the clock phase available between the input digits and the carries C_5 through C_8. This extra time can be used to form certain auxiliary carry functions which enable us to generate additional carries during the third clock phase simultaneously with C_5 through C_8. More specifically, C_9, C_{10}, etc., can be formed during the third clock phase as functions of C_4 if some of the terms in the expanded relations for C_9, C_{10}, etc., are combined as auxiliary carry functions in separate stages during the intervening clock phase.

For example, the expression for C_9 is expanded in equations 8 to be a function of C_4.

$$\begin{array}{l} C_9 = \boxed{A_9B_9} \\ \quad + \boxed{(A_9+B_9)A_8B_8} \\ \quad + \boxed{(A_9+B_9)(A_8+B_8)A_7B_7} \\ \quad + \boxed{(A_9+B_9)(A_8+B_8)(A_7+B_7)A_6B_6} \\ \quad + \boxed{(A_9+B_9)(A_8+B_8)(A_7+B_7)(A_6+B_6)A_5B_5} \\ \quad + \boxed{(A_9+B_9)(A_8+B_8)(A_7+B_7)(A_6+B_6)(A_5+B_5)}\,C_4 \end{array} \qquad \begin{array}{l} C_9 = \boxed{X_9} \\ \quad + \boxed{Y_9}\,C_4 \end{array} \quad (8)$$

The outlines drawn around the various parts of equations 8 serve merely to correlate the corresponding parts. The five terms enclosed within the triangle and represented by X_9 can be reduced to four terms by combining the first two. The reduced 4-term expression can then be implemented in one gating stage during the second clock phase. Furthermore, the single factor enclosed within the rectangle, called Y_9, can also be implemented during the second clock phase in one gating stage. By means of these two auxiliary carry functions, C_9 can be formed quite easily in one gating stage during the third clock phase according to the second equation in equations 8.

Similarly, C_{10} through C_{13} can be formed during the third clock phase by utilizing these auxiliary carry functions. The most complicated of these expressions, C_{13}, is illustrated in equations 9 where further combinations must be made to bring the number of terms down to four.

$$C_{13} = A_{13}B_{13} + (A_{13}+B_{13})A_{12}B_{12} + (A_{13}+B_{13})(A_{12}+B_{12})A_{11}B_{11} + (A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})\times A_{10}B_{10} + (A_{13}+B_{13})(A_{12}+B_{12})\times(A_{11}+B_{11})(A_{10}+B_{10})X_9 + (A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})\times(A_{10}+B_{10})Y_9C_4$$
$$= (A_{13}+B_{13})(A_{13}+A_{12})(A_{13}+B_{12})\times(B_{13}+A_{12})(B_{13}+B_{12}) + (A_{13}+B_{13})(A_{12}+B_{12})A_{11}B_{11} + (A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})\times A_{10}B_{10} + (A_{13}+B_{13})(A_{12}+B_{12})\times(A_{11}+B_{11})(A_{10}+B_{10})(X_9+Y_9)\times(X_9+C_4) \quad (9)$$

Fig. 9 illustrates a parallel adder handling 14 binary digits using one pair of auxiliary carry functions.

By means of additional auxiliary carry functions it is possible to extend still further the sequence of carries to be formed in one clock phase. For example, as shown in equations 10, C_{14} can also be expressed as a function of C_4 with the aid of auxiliary carry functions X_{14} and Y_{14}.

$$\begin{array}{l} C_{14} = \boxed{A_{14}B_{14}} \\ \quad + \boxed{(A_{14}+B_{14})A_{13}B_{13}} \\ \quad + \boxed{(A_{14}+B_{14})(A_{13}+B_{13})A_{12}B_{12}} \\ \quad + \boxed{(A_{14}+B_{14})(A_{13}+B_{13})(A_{12}+B_{12})A_{11}B_{11}} \\ \quad + \boxed{(A_{14}+B_{14})(A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})A_{10}B_{10}} \\ \quad + \boxed{(A_{14}+B_{14})(A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})(A_{10}+B_{10})}\,X_9 \\ \quad + \boxed{(A_{14}+B_{14})(A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})(A_{10}+B_{10})}\,Y_9C_4 \end{array} \qquad \begin{array}{l} C_{14} = \boxed{X_{14}} \\ \quad + \boxed{Y_{14}}\,X_9 \\ \quad + \boxed{Y_{14}}\,Y_9C_4 \end{array} \quad (10)$$

These again represent the terms within the triangle and rectangles, respectively. C_{15}, C_{16}, and C_{17} can also be implemented in single stages as functions of C_4 using the two pairs of auxiliary carry functions. C_{18}, C_{19}, and C_{20} require still another pair of auxiliary carry functions in order for them to be generated during the third clock phase as functions of C_4.

If the number of auxiliary functions were of no concern, a total of 25 carries could be generated simultaneously as functions of C_4 during the third clock phase. However, by limiting the number of simultaneous carries to 16, only three pairs of auxiliary carry functions are required. Fig. 10 illustrates a parallel adder handling 21 binary digits utilizing this scheme.

Two Levels of Auxiliary Carry Functions

To extend the parallel adder to accommodate 53 binary digits, only one additional clock phase is necessary. During the fourth clock phase the carries C_{21} through C_{52} can all be generated as functions of C_{20}. The entire parallel array of sum digits, S_1 through S_{53}, is then formed during the fifth clock phase.

The ability to generate all of the carries C_{21} through C_{52} during the fourth clock phase stems from the fact that two clock phases are available between these carries and the input digits. This permits the formation of two levels of auxiliary carry functions. The first level consists of sets of X and Y values which are functions of only the appropriate augend and addend digits, as previously. The second level of auxiliary carry functions consists of sets of stages labeled Z and W which are functions of certain first-level functions.

Fig. 11 illustrates in block-diagram form the complete 53-bit adder which makes use of second-level auxiliary carry stages. As in the case of the preceding carries, C_{21} through C_{32} are generated as functions of the appropriate augend and addend digits, some of the first-level auxiliary carry stages, and C_{20}. For example, the most complicated of these, C_{32}, is shown in equations 11 to be reducible to four terms.

$$C_{32} = A_{32}B_{32} + (A_{32}+B_{32})A_{31}B_{31} + (A_{32}+B_{32})(A_{31}+B_{31})A_{30}B_{30} + (A_{32}+B_{32})(A_{31}+B_{31})(A_{30}+B_{30})X_{29} + (A_{32}+B_{32})(A_{31}+B_{31})(A_{30}+B_{30})\times Y_{29}X_{25} + (A_{32}+B_{32})(A_{31}+B_{31})\times(A_{30}+B_{30})Y_{29}Y_{25}C_{20}$$

$$
\begin{aligned}
&=(A_{32}+B_{32})(A_{32}+A_{31})(A_{32}+B_{31})\times\\
&\quad(B_{32}+A_{31})(B_{32}+B_{31})+(A_{32}+B_{32})\times\\
&\quad(A_{31}+B_{31})A_{30}B_{30}+(A_{32}+B_{32})\times\\
&\quad(A_{31}+B_{31})(A_{30}+B_{30})(X_{29}+Y_{29})\times\\
&\quad(X_{29}+X_{25})+(A_{32}+B_{32})(A_{31}+B_{31})\times\\
&\qquad(A_{30}+B_{30})Y_{29}Y_{25}C_{20} \quad (11)
\end{aligned}
$$

The next higher order carry, C_{33}, requires a third pair of auxiliary carry functions, X_{33} and Y_{33}, as shown in equations 12. However, at this point it becames economical to form a pair of second-level auxiliary carry functions, Z_{33}, consisting of terms within the solid-line triangle, and W_{33}, consisting of the single term within the solid-line rectangle. C_{33} can then be easily generated by means of Z_{33} and W_{33} as shown in equations 12. The elements enclosed within the broken-line triangles and rectangles correspond to the first-level auxiliary carry functions, X_{33} and Y_{33}.

The subsequent carries, C_{34}, C_{35}, etc., are similarly generated by means of these and, when necessary, other second-level auxiliary carry functions. For example, for the carries up to C_{37}, one pair of second-level functions is sufficient, as seen from equations 13. C_{38} requires the formation of another pair of first and second-level functions, as shown in equations 14.

Table II. Auxiliary Carry Functions

$$
\begin{aligned}
X_9&=F_9+R_9R_8D_7+R_9R_8R_7D_6+R_9R_8R_7R_6D_5 & Y_9&=R_9R_8R_7R_6R_5\\
X_{14}&=F_{14}+R_{14}R_{13}D_{12}+R_{14}R_{13}R_{12}D_{11}+R_{14}R_{13}R_{12}R_{11}D_{10} & Y_{14}&=R_{14}R_{13}R_{12}R_{11}R_{10}\\
X_{18}&=D_{18}+R_{18}D_{17}+R_{18}R_{17}D_{16}+R_{18}R_{17}R_{16}D_{15} & Y_{18}&=R_{18}R_{17}R_{16}R_{15}\\
X_{25}&=F_{25}+R_{25}R_{24}D_{23}+R_{25}R_{24}R_{23}D_{22}+R_{25}R_{24}R_{23}R_{22}D_{21} & Y_{25}&=R_{25}R_{24}R_{23}R_{22}R_{21}\\
X_{29}&=D_{29}+R_{29}D_{28}+R_{29}R_{28}D_{27}+R_{29}R_{28}R_{27}D_{26} & Y_{29}&=R_{29}R_{28}R_{27}R_{26}\\
X_{33}&=D_{33}+R_{33}D_{32}+R_{33}R_{32}D_{21}+R_{33}R_{32}R_{31}D_{30} & Y_{33}&=R_{33}R_{32}R_{31}R_{30}\\
X_{38}&=F_{38}+R_{38}R_{37}D_{36}+R_{38}R_{37}R_{36}D_{35}+R_{38}R_{37}R_{36}R_{35}D_{34} & Y_{38}&=R_{38}R_{37}R_{36}R_{35}R_{34}\\
X_{43}&=F_{43}+R_{43}R_{42}D_{41}+R_{43}R_{42}R_{41}D_{40}+R_{43}R_{42}R_{41}R_{40}D_{39} & Y_{43}&=R_{43}R_{42}R_{41}R_{40}R_{39}\\
X_{48}&=F_{48}+R_{48}R_{47}D_{46}+R_{48}R_{47}R_{46}D_{45}+R_{48}R_{47}R_{46}R_{45}D_{44} & Y_{48}&=R_{48}R_{47}R_{46}R_{45}R_{44}\\
Z_{33}&=X_{33}+Y_{33}X_{29}+Y_{33}Y_{29}X_{25} & W_{33}&=Y_{33}Y_{29}Y_{25}\\
Z_{38}&=X_{38}+Y_{38}X_{33}+Y_{38}Y_{33}X_{29}+Y_{38}Y_{33}Y_{29}X_{25} & W_{38}&=Y_{38}Y_{33}Y_{29}Y_{25}\\
Z_{43}&=X_{43}+Y_{43}X_{38}+Y_{43}Y_{38}X_{33}+Y_{43}Y_{38}Y_{33}(X_{29}+Y_{29})(X_{29}+X_{25}) & W_{43}&=Y_{43}Y_{38}Y_{33}Y_{29}Y_{25}\\
Z_{48}&=X_{48}+Y_{48}X_{43}+Y_{48}Y_{43}(X_{38}+Y_{38})(X_{38}+X_{33})+ & W_{48}&=Y_{48}Y_{43}Y_{38}Y_{33}Y_{29}Y_{25}\\
&\quad Y_{48}Y_{43}Y_{38}Y_{33}(X_{29}+Y_{29})(X_{29}+X_{25})
\end{aligned}
$$

F_k represents $(A_k+B_k)(A_k+A_{k-1})(A_k+B_{k-1})(B_k+A_{k-1})(B_k+B_{k-1})$
D_k represents A_kB_k
R_k represents (A_k+B_k)

$$
\begin{aligned}
C_{37}&=A_{37}B_{37}+(A_{37}+B_{37})A_{36}B_{36}+\\
&\quad(A_{37}+B_{37})(A_{36}+B_{36})A_{35}B_{35}+\\
&\quad(A_{37}+B_{37})(A_{36}+B_{36})(A_{35}+B_{35})\times\\
&\quad A_{34}B_{34}+(A_{37}+B_{37})(A_{36}+B_{36})\times\\
&\quad(A_{35}+B_{35})(A_{34}+B_{34})Z_{33}+\\
&\quad(A_{37}+B_{37})(A_{36}+B_{36})(A_{35}+B_{35})\times\\
&\quad(A_{34}+B_{34})W_{33}C_{20}\\
&=(A_{37}+B_{37})(A_{37}+A_{36})(A_{37}+B_{36})\times\\
&\quad(B_{37}+A_{36})(B_{37}+B_{36})+(A_{37}+B_{37})\times\\
&\quad(A_{36}+B_{36})A_{35}B_{35}+(A_{37}+B_{37})\times\\
&\quad(A_{36}+B_{36})(A_{35}+B_{35})A_{34}B_{34}+\\
&\quad(A_{37}+B_{37})(A_{36}+B_{36})(A_{35}+B_{35})\times\\
&\quad(A_{34}+B_{34})(Z_{33}+W_{33})(Z_{33}+C_{20}) \quad (13)
\end{aligned}
$$

$$
\begin{aligned}
C_{33}=\ & A_{33}B_{33}\\
+\ & (A_{33}+B_{33})A_{32}B_{32}\\
+\ & (A_{33}+B_{33})(A_{32}+B_{32})A_{31}B_{31}\\
+\ & (A_{33}+B_{33})(A_{32}+B_{32})(A_{31}+B_{31})A_{30}B_{30}\\
+\ & (A_{33}+B_{33})(A_{32}+B_{32})(A_{31}+B_{31})(A_{30}+B_{30})\,X_{29}\\
+\ & (A_{33}+B_{33})(A_{32}+B_{32})(A_{31}+B_{31})(A_{30}+B_{30})\,Y_{29}X_{25}\\
+\ & (A_{33}+B_{33})(A_{32}+B_{32})(A_{31}+B_{31})(A_{30}+B_{30})\,Y_{29}Y_{25}C_{20}
\end{aligned}
\qquad (12)
$$

$$
\begin{aligned}
C_{33}=\ & X_{33}\\
+\ & Y_{33}X_{29}\\
+\ & Y_{33}Y_{29}X_{25}\\
+\ & Y_{33}Y_{29}Y_{25}C_{20}
\end{aligned}
\qquad
\begin{aligned}
C_{33}=\ & Z_{33}\\
+\ & W_{33}C_{20}
\end{aligned}
$$

The last digit position where auxiliary carry functions are introduced is at 48. The carry at this position, C_{48}, is shown in equations 15 to be a simple function of the last pair of second-level auxiliary carry functions.

The number of stages required to implement the adder in this fashion can easily be determined from Fig. 11. Each box in the diagram represents one gating stage. In addition to the four registers of gating stages for the augend digits, addend digits, carry digits, and sum digits, only 26 gating stages, equivalent to one-half of a register, are required to create the auxiliary carry functions.

The expressions for the auxiliary carry functions of this particular adder as well as its final carry functions are shown in Tables II and III, respectively.

The top line of Table IV gives some statistics on the number of components required for the adder represented in Fig. 11. Two other slightly different versions have been worked out in which fewer gates need to be driven by the most heavily loaded tube. As Table IV shows, these

Table III. Carry Functions

$$
\begin{aligned}
C_1&=D_1+R_1C_0\\
C_2&=D_2+R_2D_1+R_2R_1C_0\\
C_3&=D_3+R_3D_2+R_3R_2D_1+R_3R_2R_1C_0\\
C_4&=D_4+R_4D_3+R_4R_3D_2+R_4R_3R_2R_1(A_1+C_0)(B_1+C_0)\\
C_5&=D_5+R_5C_4\\
C_6&=D_6+R_6D_5+R_6R_5C_4\\
C_7&=D_7+R_7D_6+R_7R_6D_5+R_7R_6R_5C_4\\
C_8&=F_8+R_8R_7D_6+R_8R_7R_6D_5+R_8R_7R_6R_5C_4\\
C_9&=X_9+Y_9C_4\\
C_{10}&=D_{10}+R_{10}(X_9+Y_9)(X_9+C_4)\\
C_{11}&=D_{11}+R_{11}D_{10}+R_{11}R_{10}(X_9+Y_9)(X_9+C_4)\\
C_{12}&=D_{12}+R_{12}D_{11}+R_{12}R_{11}D_{10}+R_{12}R_{11}R_{10}(X_9+Y_9)(X_9+C_4)\\
C_{13}&=F_{13}+R_{13}R_{12}D_{11}+R_{13}R_{12}R_{11}D_{10}+R_{13}R_{12}R_{11}R_{10}(X_9+X_9)(X_9+C_4)\\
C_{14}&=X_{14}+Y_{14}X_9+Y_{14}Y_9C_4\\
C_{15}&=D_{15}+R_{15}X_{14}+R_{15}Y_{14}(X_9+Y_9)(X_9+C_4)\\
C_{16}&=D_{16}+R_{16}D_{15}+R_{16}R_{15}X_{14}+R_{16}R_{15}Y_{14}(X_9+Y_9)(X_9+C_4)\\
C_{17}&=F_{17}+R_{17}R_{16}D_{15}+R_{17}R_{16}R_{15}X_{14}+R_{17}R_{16}R_{15}Y_{14}(X_9+Y_9)(X_9+C_4)\\
C_{18}&=X_{18}+Y_{18}X_{14}+Y_{18}Y_{14}X_9+Y_{18}Y_{14}Y_9C_4\\
C_{19}&=D_{19}+R_{19}(X_{18}+Y_{18})(X_{18}+X_{14})+R_{19}Y_{18}Y_{14}(X_9+Y_9)(X_9+C_4)\\
C_{20}&=D_{20}+R_{20}D_{19}+R_{20}R_{19}(X_{18}+Y_{18})(X_{18}+X_{14})+R_{20}R_{19}Y_{18}Y_{14}(X_9+Y_9)(X_9+C_4)\\
C_{21}&=D_{21}+R_{21}C_{20}\\
C_{22}&=D_{22}+R_{22}D_{21}+R_{22}R_{21}C_{20}\\
C_{23}&=D_{23}+R_{23}D_{22}+R_{23}R_{22}D_{21}+R_{23}R_{22}R_{21}C_{20}\\
C_{24}&=F_{24}+R_{24}R_{23}D_{22}+R_{24}R_{23}R_{22}D_{21}+R_{24}R_{23}R_{22}R_{21}C_{20}\\
C_{25}&=X_{25}+Y_{25}C_{20}\\
C_{26}&=D_{26}+R_{26}X_{25}+R_{26}Y_{25}C_{20}\\
C_{27}&=D_{27}+R_{27}D_{26}+R_{27}R_{26}X_{25}+R_{27}R_{26}Y_{25}C_{20}\\
C_{28}&=F_{28}+R_{28}R_{27}D_{26}+R_{28}R_{27}R_{26}X_{25}+R_{28}R_{27}R_{26}Y_{25}C_{20}\\
C_{29}&=X_{29}+Y_{29}X_{25}+Y_{29}Y_{25}C_{20}\\
C_{30}&=D_{30}+R_{30}(X_{29}+Y_{29})(X_{29}+X_{25})+R_{30}Y_{29}Y_{25}C_{20}\\
C_{31}&=D_{31}+R_{31}D_{30}+R_{31}R_{30}(X_{29}+Y_{29})(X_{29}+X_{25})+R_{31}R_{30}Y_{29}Y_{25}C_{20}\\
C_{32}&=F_{32}+R_{32}R_{31}D_{30}+R_{32}R_{31}R_{30}(X_{29}+Y_{29})(X_{29}+X_{25})+R_{32}R_{31}R_{30}Y_{29}Y_{25}C_{20}\\
C_{33}&=Z_{33}+W_{33}C_{20}\\
C_{34}&=D_{34}+R_{34}(Z_{33}+W_{33})(Z_{33}+C_{20})\\
C_{35}&=D_{35}+R_{35}D_{34}+R_{35}R_{34}(Z_{33}+W_{33})(Z_{33}+C_{20})\\
C_{36}&=D_{36}+R_{36}D_{35}+R_{36}R_{35}D_{34}+R_{36}R_{35}R_{34}(Z_{33}+W_{33})(Z_{33}+C_{20})\\
C_{37}&=F_{37}+R_{37}R_{36}D_{35}+R_{37}R_{36}R_{35}D_{34}+R_{37}R_{36}R_{35}R_{34}(Z_{33}+W_{33})(Z_{33}+C_{20})\\
C_{38}&=Z_{38}+W_{38}C_{20}\\
C_{39}&=D_{39}+R_{39}(Z_{38}+W_{38})(Z_{38}+C_{20})\\
C_{40}&=D_{40}+R_{40}D_{39}+R_{40}R_{39}(Z_{38}+W_{38})(Z_{38}+C_{20})\\
C_{41}&=D_{41}+R_{41}D_{40}+R_{41}R_{40}D_{39}+R_{41}R_{40}R_{39}(Z_{38}+W_{38})(Z_{38}+C_{20})\\
C_{42}&=F_{42}+R_{42}R_{41}D_{40}+R_{42}R_{41}R_{40}D_{39}+R_{42}R_{41}R_{40}R_{39}(Z_{38}+W_{38})(Z_{38}+X_{20})\\
C_{43}&=Z_{43}+W_{43}C_{20}\\
C_{44}&=D_{44}+R_{44}(Z_{43}+W_{43})(Z_{43}+C_{20})\\
C_{45}&=D_{45}+R_{45}D_{44}+R_{45}R_{44}(Z_{43}+W_{43})(Z_{43}+C_{20})\\
C_{46}&=D_{46}+R_{46}D_{45}+R_{46}R_{45}D_{44}+R_{46}R_{45}R_{44}(Z_{43}+W_{43})(Z_{43}+C_{20})\\
C_{47}&=F_{47}+R_{47}R_{46}D_{45}+R_{47}R_{46}R_{45}D_{44}+R_{47}R_{46}R_{45}R_{44}(Z_{43}+W_{43})(Z_{43}+C_{20})\\
C_{48}&=Z_{48}+W_{48}C_{20}\\
C_{49}&=D_{49}+R_{49}(Z_{48}+W_{48})(Z_{48}+C_{20})\\
C_{50}&=D_{50}+R_{50}D_{49}+R_{50}R_{49}(Z_{48}+W_{48})(Z_{48}+C_{20})\\
C_{51}&=D_{51}+R_{51}D_{50}+R_{51}R_{50}D_{49}+R_{51}R_{50}R_{49}(Z_{48}+W_{48})(Z_{48}+C_{20})\\
C_{52}&=F_{52}+R_{52}R_{51}D_{50}+R_{52}R_{51}R_{50}D_{49}+R_{52}R_{51}R_{50}R_{49}(Z_{48}+W_{48})(Z_{48}+C_{20})
\end{aligned}
$$

F_k represents $(A_k+B_k)(A_k+A_{k-1})(A_k+B_{k-1})(B_k+A_{k-1})(B_k+B_{k-1})$
D_k represents A_kB_k
R_k represents (A_k+B_k)

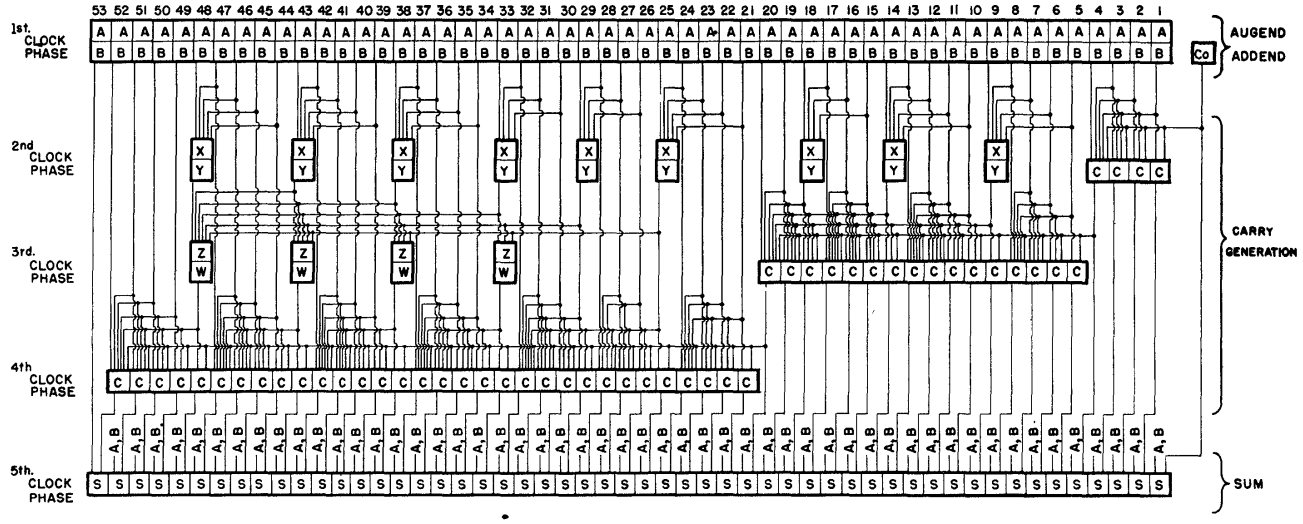

Fig. 11. 53-bit parallel binary adder

Table IV. Component Requirements for a 53-bit Parallel Binary Adder

Max. Load*	No. of Stages	Tubes	Delay Lines (in μsec.)	Germanium Diodes
25	238	238	300	10,000
19	253	253	250	10,000
14	285	285	150	10,000

* Unit of load = one gate-load.

versions also require different proportions of components.

If the adder is to be used for multiplication, division, and other operations requiring the recirculation of the sum digits back into one of the inputs, the clock must be available in five phases in order to complete the addition as well as the recirculation in 1 microsecond.

$$
\begin{aligned}
C_{38} = {} & A_{38}B_{38} \\
& + (A_{38}+B_{38})A_{37}B_{37} \\
& + (A_{38}+B_{38})(A_{37}+B_{37})A_{36}B_{36} \\
& + (A_{38}+B_{38})(A_{37}+B_{37})(A_{36}+B_{36})A_{35}B_{35} \\
& + (A_{38}+B_{38})(A_{37}+B_{37})(A_{36}+B_{36})(A_{35}+B_{35})A_{34}B_{34} \\
& + (A_{38}+B_{38})(A_{37}+B_{37})(A_{36}+B_{36})(A_{35}+B_{35})(A_{34}+B_{34})\,X_{33} \\
& + (A_{38}+B_{38})(A_{37}+B_{37})(A_{36}+B_{36})(A_{35}+B_{35})(A_{34}+B_{34})\,Y_{33}X_{29} \\
& + (A_{38}+B_{38})(A_{37}+B_{37})(A_{36}+B_{36})(A_{35}+B_{35})(A_{34}+B_{34})\,Y_{33}Y_{29}X_{25} \\
& + (A_{38}+B_{38})(A_{37}+B_{37})(A_{36}+B_{36})(A_{35}+B_{35})(A_{34}+B_{34})\,Y_{33}Y_{29}Y_{25}\,C_{20}
\end{aligned}
\tag{14}
$$

$$
\begin{aligned}
C_{38} = {} & X_{38} \\
& + Y_{38}X_{33} \\
& + Y_{38}Y_{33}X_{29} \\
& + Y_{38}Y_{33}Y_{29}X_{25} \\
& + Y_{38}Y_{33}Y_{29}Y_{25}\,C_{20}
\end{aligned}
\qquad
\begin{aligned}
C_{38} = {} & Z_{38} \\
& + W_{38}\,C_{20}
\end{aligned}
$$

$$
\begin{aligned}
C_{48} = {} & X_{48} \\
& + Y_{48}X_{43} \\
& + Y_{48}Y_{43}X_{38} \\
& + Y_{48}Y_{43}Y_{38}X_{33} \\
& + Y_{48}Y_{43}Y_{38}Y_{33}X_{29} \\
& + Y_{48}Y_{43}Y_{38}Y_{33}Y_{29}X_{25} \\
& + Y_{48}Y_{43}Y_{38}Y_{33}Y_{29}Y_{25}\,C_{20}
\end{aligned}
\tag{15}
$$

$$
\begin{aligned}
C_{48} = {} & X_{48} \\
& + Y_{48}X_{43} \\
& + Y_{48}Y_{43}(X_{38}+Y_{38})(X_{38}+X_{33}) \\
& + Y_{48}Y_{43}Y_{38}Y_{33}(X_{29}+Y_{29})(X_{29}+X_{25}) \\
& + Y_{48}Y_{43}Y_{38}Y_{33}Y_{29}Y_{25}\,C_{20}
\end{aligned}
\qquad
\begin{aligned}
C_{48} = {} & Z_{48} \\
& + W_{48}\,C_{20}
\end{aligned}
$$

References

1 An Experimental Rapid Access Memory Using Diodes and Capacitors, **A. W. Holt.** *Proceedings*, Association for Computing Machinery, New York, N. Y., December 1952.

2. High-Speed Memory Developments at the National Bureau of Standards, **R. J. Slutz, A. W. Holt, R. P. Witt, D. C. Friedman.** *Circular 551*, National Bureau of Standards, Washington, D. C., January 1955, pp. 93–108.

3. SEAC, **S. Greenwald, R. C. Haeuter, S. N. Alexander.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 41, October 1953, pp. 1300–13. Also published in *Circular 551*, National Bureau of Standards, Washington, D. C., January 1955, pp. 5–26.

4. System Organization of the DYSEAC, **A. L. Leiner, S. N. Alexander.** *Transactions*, Institute of Radio Engineers Professional Group on Electronic Computers, New York, N. Y., vol. EC-3, no. 1, March 1954.

5. System Design of the SEAC and DYSEAC, **A. L. Leiner, W. A. Notz, J. L. Smith, A. Weinberger.** *Ibid.*, no. 2, June 1954.

6. Dynamic Circuit Techniques Used in SEAC and DYSEAC, **R. D. Elbourn, R. P. Witt.** *Proceedings*, Institute of Radio Engineers, New York, N. Y., vol. 41, October 1953, pp. 1380–87. Also published in *Transactions*, Institute of Radio Engineers Professional Group on Electronic Computers, New York, N. Y., vol. EC-2, no. 1, March 1953, and in *Circular 551*, National Bureau of Standards, Washington, D. C., January 1955, pp. 27–38.

7. Arithmetic Operations in Digital Computers (book), **R. K. Richards.** D. Van Nostrand Company, Inc., New York, N. Y., 1955, pp. 26–50.