Reprinted from *Nat. Bur. Stand. Circ. 591*, pp. 3–12 (1958)

A Logic for High-Speed Addition

A. Weinberger and J. L. Smith

1. Introduction

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

The major speed limitation in any adder is in the production of carries, and in this paper the problem is attacked from the standpoint of logical organization. Although work is being done elsewhere 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-µsec addition times for 53-bit words, using the 1-Mc circuitry that has been successfully utilized in SEAC [3] and DYSEAC [4,5].

The increased complexity of the logic of this adder necessitated the extensive use 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] will be presented. As shown in figure 1.1, 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 Mc/sec. The transit time of a pulse through a stage, however, is much less than 1 µsec. For this reason, the clock pulses are made available in several phases. The way in which different stages are controlled by clock pulses of different phases is illustrated in figure 1.2. In SEAC, for example, 1-Mc clock pulses are available in 3 phases, ⅓ µsec apart. In DYSEAC, 4-phase clock pulses are used, whereas for reasons that will be developed later, in the adder to be described a 5-phase clock is used. Figure 1.3 shows graphically these timing relationships for SEAC. Signals emitted from different stages clocked at different times must be synchronized by means of electric delay lines before they are gated in a common stage, as shown in figure 1.4. Both positive and negative signals are available from a stage, the negative signals being used for inhibiting (see fig. 1.5).

The logical gating required in any stage of the adder to be described is essentially of the same complexity as that required in the packaged building blocks used in constructing DYSEAC, and in the OR–AND–OR gating configuration of a stage up to 4 AND-gates and up to 6 inputs in the largest AND-gate are permitted.

Boolean notation of the sort described by Richards [7] will be used hereafter to describe the gating configurations. In figure 1.6 are 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, $(A+B)\overline{C}\overline{D}EF$, corresponds to the top AND-gate; the second term, $(G+H)I$, corresponds to the middle AND-gate; and the last term, $J(K+L+M)\overline{N}$, 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, $(A+B)$, $\overline{C}$, $\overline{D}$, E, and F, correspond to the 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., $\overline{C}$ and $\overline{D}$ 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 that follow, no distinction is made between delayed and undelayed signals.

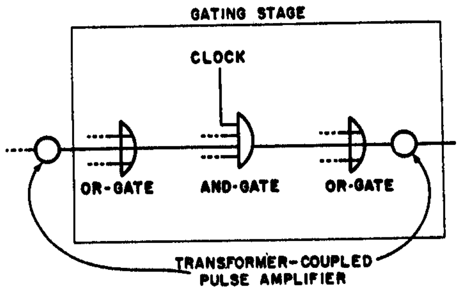

FIGURE 1.1. *One stage of SEAC-type circuitry.*

[1] Figures in brackets indicate the literature references on page 12.

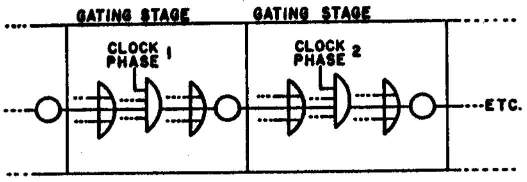

FIGURE 1.2. *Gating stages with different clock phases.*

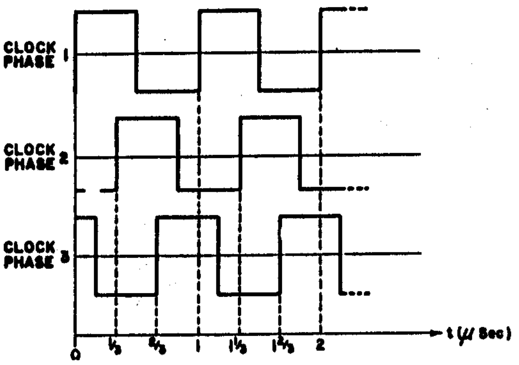

FIGURE 1.3. *Time relationships among SEAC clock phases.*

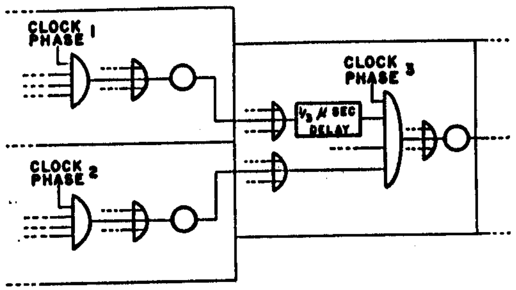

FIGURE 1.4. *Synchronizing by means of electric delay lines.*

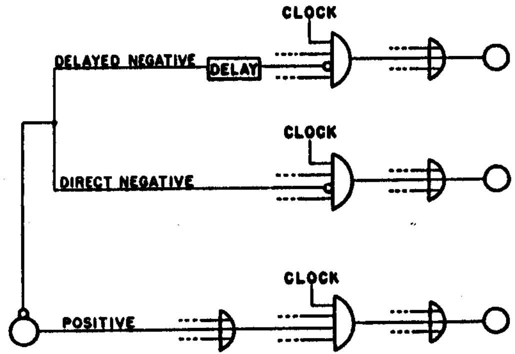

FIGURE 1.5. *Use of negative signals for inhibiting.*

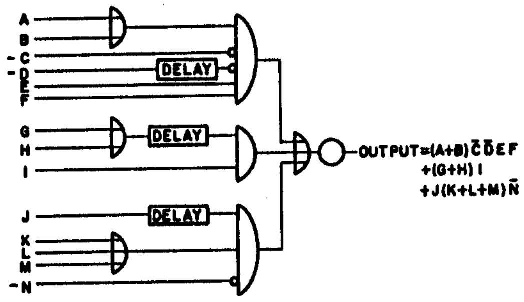

FIGURE 1.6. *Typical gating stage and corresponding Boolean expression.*

2. Sequential Carry Generation

The analysis leading to the design of the parallel adder will now be described in detail.

Let

$$A = \text{augend} = A_k 2^{k-1} + A_{k-1} 2^{k-2} + \dots + A_1 2^0,$$

$$B = \text{addend} = B_k 2^{k-1} + B_{k-1} 2^{k-2} + \dots + B_1 2^0,$$

$$S = \text{sum} \quad = S_k 2^{k-1} + S_{k-1} 2^{k-2} + \quad . + S_1 2^0,$$

C_k=the carry resulting from the addition in the kth digit position.

The well-known rules for binary addition are given in the form of a function table (table 1.1). From these, the binary sum and carry can be expressed in Boolean notation as follows:

$$S_k = \overline{A}_k \overline{B}_k C_{k-1} + \overline{A}_k B_k \overline{C}_{k-1} + A_k \overline{B}_k \overline{C}_{k-1} + A_k B_k C_{k-1}. \tag{1}$$

TABLE 1.1. *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

$$
\begin{aligned}
C_k &= \overline{A}_k B_k C_{k-1} + A_k \overline{B}_k C_{k-1} + A_k B_k \overline{C}_{k-1} + A_k B_k C_{k-1} \\
&= A_k B_k + A_k C_{k-1} + B_k C_{k-1} \\
&= (A_k + B_k)(A_k + C_{k-1})(B_k + C_{k-1}) \qquad (2) \\
&= A_k B_k + (A_k + B_k) C_{k-1}.
\end{aligned}
$$

The carry function, C_k, has been reduced from 4 terms of 3 factors each (corresponding to 4 AND-gates with 3 inputs each), as shown in the top line of eq (2), to 3 alternative forms, each involving fewer terms and factors.

Since the expression for S_k in eq (1) can be implemented in one gating stage, any sum digit can be made available during the clock phase immediately following the formation of its corresponding carry, C_{k-1}. However, if the carries are generated according to eq (2), each carry digit would have to await the formation of the next lower-order carry. As a result, the sum digits could be obtained at the rate of only one per clock phase, for if C_1 is available during the first clock phase, C_2 could be generated during the second clock phase, C_3 during the third clock phase, etc. For numbers having n binary digits, $n-1$ possible carries would have to be provided for, requiring $n-1$ clock phases for their complete determination. If a 4-phase, 1-Mc clock were used, 4 successive sum digits could be obtained during 1 μsec. Such an arrangement, using *sequential* carry generation, would provide an increase in speed of a factor of only four over the addition speed of a completely serial adder.

3. Simultaneous Carry Generation

The limitation on the sequential method of forming the carries stems from the use of eq (2), which specify C_k as an explicit function of C_{k-1}. It can be shown that a carry need not depend explicitly on the preceding one, but can be expressed as a function of only the relevant augend and addend digits and some lower-order carry. A considerable gain in speed may be obtained as a result of this.

Using the functional form given by the last equation in (2), successive carries are shown to be expressible in terms of the same lower-order carry by a method of substitution.

$$
\begin{array}{lll}
C_1 = & A_1B_1 & \\
 & +(A_1+B_1)C_0 & \\
 & & \\
C_2 = & A_2B_2 & = \quad A_2B_2 \\
 & +(A_2+B_2)C_1 & \quad +(A_2+B_2)A_1B_1 \\
 & & \quad +(A_2+B_2)(A_1+B_1)C_0 \\
 & & \\
C_3 = & A_3B_3 & = \quad A_3B_3 \\
 & +(A_3+B_3)C_2 & \quad +(A_3+B_3)A_2B_2 \\
 & & \quad +(A_3+B_3)(A_2+B_2)A_1B_1 \\
 & & \quad +(A_3+B_3)(A_2+B_2)(A_1+B_1)C_0 \qquad (3) \\
 & & \\
C_4 = & A_4B_4 & = \quad A_4B_4 \\
 & +(A_4+B_4)C_3 & \quad +(A_4+B_4)A_3B_3 \\
 & & \quad +(A_4+B_4)(A_3+B_3)A_2B_2 \\
 & & \quad +(A_4+B_4)(A_3+B_3)(A_2+B_2)A_1B_1 \\
 & & \quad +(A_4+B_4)(A_3+B_3)(A_2+B_2)(A_1+B_1)C_0 \\
 & & = \quad A_4B_4 \\
 & & \quad +(A_4+B_4)A_3B_3 \\
 & & \quad +(A_4+B_4)(A_3+B_3)A_2B_2 \\
 & & \quad +(A_4+B_4)(A_3+B_3)(A_2+B_2)(A_1+B_1)(A_1+C_0)(B_1+C_0).
\end{array}
$$

Equations (3) show how as many as 4 successive carries can be expressed as functions of the same carry, with all expressions consisting of no more than 4 terms and with the largest term consisting of no more than 6 factors. These 4 carries can therefore be generated simultaneously by means of only 4 gating stages.

Similarly, the next more significant four carries, C_5 through C_8, can be formed simultaneously during the next clock phase as functions of the appropriate augend and addend digits and C_4. In short, four successive carry digits can be formed simultaneously every clock phase. One gating stage per carry is required.

To summarize, if C_0 is available in the first clock phase, C_1 through C_4 can be generated during the second clock phase, C_5 through C_8 during the third clock phase, etc. Each group of sum digits can be obtained one clock phase after the corresponding group of carries has been formed. Figure 1.7 illustrates in block-diagram form an adder utilizing this principle of simultaneous carry generation.

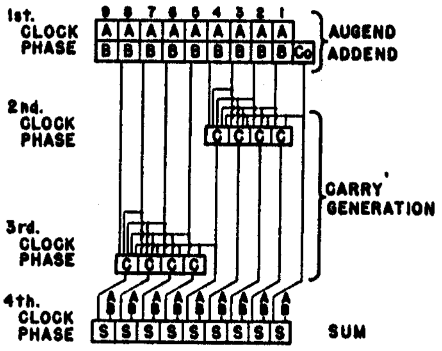

FIGURE 1.7. *Nine-bit parallel binary adder.*

4. Use of Auxiliary Carry Functions

Of signal importance is the use that can be made of the second clock phase to further speed up the addition process. This time can be utilized to form certain *auxiliary carry functions*, which enable additional carries to be generated during the third clock phase simultaneously with the carries 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 involving only the augend and addend digits 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 shown in the first equation in (4) expanded as a function of C_4. Because of limitations on the gating complexity, it is not possible to form C_9 directly even if it were reduced to four terms. Instead, the function is implemented by parts.

$$
\begin{aligned}
C_9 = {} & A_9B_9 \\
& + (A_9+B_9)A_8B_8 \\
& + (A_9+B_9)(A_8+B_8)A_7B_7 \\
& + (A_9+B_9)(A_8+B_8)(A_7+B_7)A_6B_6 \\
& + (A_9+B_9)(A_8+B_8)(A_7+B_7)(A_6+B_6)A_5B_5 \\
& + (A_9+B_9)(A_8+B_8)(A_7+B_7)(A_6+B_6)(A_5+B_5)\,C_4
\end{aligned}
\qquad (4)
$$

$$
\begin{aligned}
C_9 = {} & X_9 \\
& + Y_9\,C_4
\end{aligned}
$$

(The outlines drawn around the various parts of eq (4) serve merely to correlate the corresponding parts of the two equations.) The 5 terms enclosed within the triangle can be reduced to 4 terms by combining the first 2 terms. This reduced 4-term expression can then be implemented in 1 gating stage during the second clock phase, and it is then designated by X_9. The single factor enclosed within the rectangle can also be implemented during the second clock phase in one gating stage. It is designated by Y_9. By means of these 2 auxiliary carry functions, X_9 and Y_9, the actual carry C_9 can be formed quite handily in 1 gating stage during the third clock phase, according to the second equation in (4).

The next 4 carries, C_{10} through C_{13}, can also be formed during the third clock phase by utilizing these same auxiliary carry functions. The most complicated of these expressions, the one for C_{13}, is given in eq (5), where further combinations are made to reduce the number of terms to four.

$$
\begin{aligned}
C_{13} = {} & A_{13}B_{13} \\
& + (A_{13}+B_{13})A_{12}B_{12} \\
& + (A_{13}+B_{13})(A_{12}+B_{12})A_{11}B_{11} \\
& + (A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})A_{10}B_{10} \\
& + (A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})(A_{10}+B_{10})X_9 \\
& + (A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})(A_{10}+B_{10})Y_9C_4
\end{aligned}
\qquad (5)
$$

$$
\begin{aligned}
C_{13} = {} & (A_{13}+B_{13})(A_{13}+A_{12})(A_{13}+B_{12})(B_{13}+A_{12}) \\
& \quad (B_{13}+B_{12}) \\
& + (A_{13}+B_{13})(A_{12}+B_{12})A_{11}B_{11} \\
& + (A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})A_{10}B_{10} \\
& + (A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})(A_{10}+B_{10}) \\
& \quad (X_9+Y_9)(X_9+C_4).
\end{aligned}
$$

Figure 1.8 illustrates a parallel adder that will complete an addition on 14 binary digits in 4 clock phases, using 1 pair of auxiliary carry functions.

By means of additional auxiliary carry functions it is possible to extend still further the sequence of carries that can be formed in the same clock phase. For example, as shown in eq (6), C_{14} can be expressed as a simple function involving C_4 and another pair of auxiliary carry functions, X_{14} and Y_{14}, which are defined implicitly in eq (6).

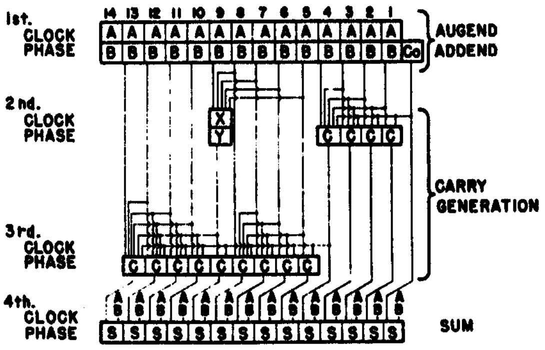

FIGURE 1.8. *Fourteen-bit parallel binary adder.*

$$
\begin{aligned}
C_{14} = {} & A_{14}B_{14} \\
+ {} & (A_{14}+B_{14})A_{13}B_{13} \\
+ {} & (A_{14}+B_{14})(A_{13}+B_{13})A_{12}B_{12} \\
+ {} & (A_{14}+B_{14})(A_{13}+B_{13})(A_{12}+B_{12})A_{11}B_{11} \\
+ {} & (A_{14}+B_{14})(A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})A_{10}B_{10} \\
+ {} & (A_{14}+B_{14})(A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})(A_{10}+B_{10})X_9 \\
+ {} & (A_{14}+B_{14})(A_{13}+B_{13})(A_{12}+B_{12})(A_{11}+B_{11})(A_{10}+B_{10})Y_9C_4
\end{aligned}
\qquad (6)
$$

$$
\begin{aligned}
C_{14} = {} & X_{14} \\
+ {} & Y_{14}X_9 \\
+ {} & Y_{14}Y_9C_4
\end{aligned}
$$

C_{15}, C_{16}, and C_{17} can also be implemented in single stages as functions of C_4 by using the same two pairs of auxiliary carry functions. C_{18}, C_{19}, and C_{20} require still a third pair of auxiliary carry functions in order that they be generated during the same clock phase as functions of C_4.

If it were desired, a total of 25 carries could be generated simultaneously as functions of C_4 during the third clock phase without exceeding the limitations on gating complexity. However, if the number of simultaneous carries is limited to 16, only 3 pairs of auxiliary carry functions are required. Figure 1.9 illustrates a parallel adder that can add numbers of 21 binary digits in 4 clock phases, utilizing this scheme.

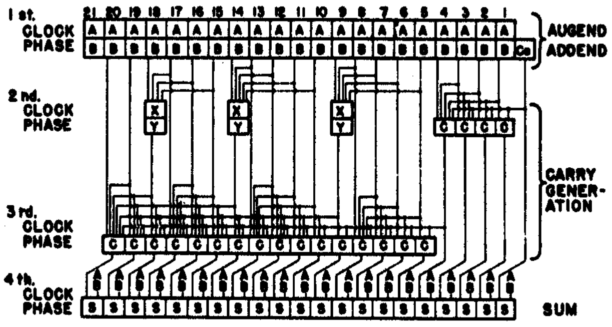

FIGURE 1.9. *Twenty-one-bit parallel binary adder*

5. Two Levels of Auxiliary Carry Functions

To extend the parallel adder to accommodate 53 binary digits, it will be shown that only 1 additional clock phase is necessary, and that 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}, can then be 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 functions. The first level consists of sets of X's and Y's, which are functions of the relevant augend and addend digits only, as was the case previously. The second-level auxiliary carry functions are generated by sets of stages labeled Z and W and are functions of certain first-level functions only.

Figure 1.10 illustrates in block-diagram form the complete 53-bit adder, which makes use of first-level and 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 eq (7) to be reducible to four terms.

$$\begin{aligned}C_{32}=&A_{32}B_{32}\\&+(A_{32}+B_{32})A_{31}B_{31}\\&+(A_{32}+B_{32})(A_{31}+B_{31})A_{30}B_{30}\\&+(A_{32}+B_{32})(A_{31}+B_{31})(A_{30}+B_{30})X_{29}\\&+(A_{32}+B_{32})(A_{31}+B_{31})(A_{30}+B_{30})Y_{29}X_{25}\\&+(A_{32}+B_{32})(A_{31}+B_{31})(A_{30}+B_{30})Y_{29}Y_{25}C_{20}\end{aligned}\tag{7}$$

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

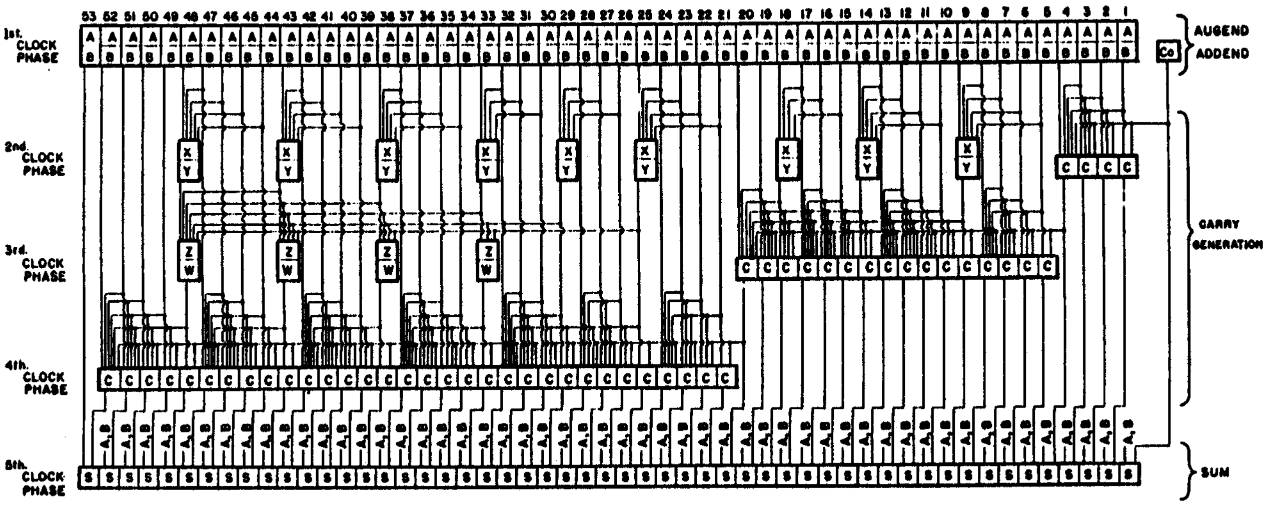

FIGURE 1.10. *Fifty-three-bit parallel binary adder.*

The next higher-order carry, C_{33}, requires a third pair of auxiliary carry functions, X_{33} and Y_{33}, as shown in eq (8). Also, at this point it becomes economical to form a pair of second-level auxiliary carry functions, Z_{33}, consisting of the terms within the solid-line triangle, and W_{33}, consisting of the factors within the solid-line rectangle. C_{33} can then be simply generated by means of Z_{33} and W_{33}, as shown in the last of eq (8).

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}, the same pair of second-level functions, W_{33} and Z_{33}, is sufficient, whereas C_{38} requires the use of another pair of first-level and second-level auxiliary carry functions in a manner exactly analogous to the formation of C_{33}.

$$
\begin{aligned}
C_{33} = {} & A_{33}B_{33} \\
+ {} & (A_{33}+B_{33})A_{32}B_{32} \\
+ {} & (A_{33}+B_{33})(A_{32}+B_{32})A_{31}B_{31} \\
+ {} & (A_{33}+B_{33})(A_{32}+B_{32})(A_{31}+B_{31})A_{30}B_{30} \\
+ {} & (A_{33}+B_{33})(A_{32}+B_{32})(A_{31}+B_{31})(A_{30}+B_{30})X_{29} \\
+ {} & (A_{33}+B_{33})(A_{32}+B_{32})(A_{31}+B_{31})(A_{30}+B_{30})Y_{29}X_{25} \\
+ {} & \boxed{(A_{33}+B_{33})(A_{32}+B_{32})(A_{31}+B_{31})(A_{30}+B_{30})Y_{29}Y_{25}}\,C_{20} \\[2ex]
C_{33} = {} & X_{33} \\
+ {} & Y_{33}X_{29} \\
+ {} & Y_{33}Y_{29}X_{25} \\
+ {} & \boxed{Y_{33}Y_{29}Y_{25}}\,C_{20} \\[2ex]
C_{33} = {} & Z_{33} \\
+ {} & \boxed{W_{33}}\,C_{20}
\end{aligned}
\tag{8}
$$

The last digit position where auxiliary carry functions are introduced is at 48. The carry at this position, C_{48}, is shown in eq (9) to be a simple function of the last pair of second-level auxiliary carry functions.

$$\begin{aligned} C_{48} = {} & X_{48} \\ & + Y_{48}X_{43} \\ & + Y_{48}Y_{43}X_{38} \\ & + Y_{48}Y_{43}Y_{38}X_{33} \\ & + Y_{48}Y_{43}Y_{38}Y_{33}X_{29} \\ & + Y_{48}Y_{43}Y_{38}Y_{33}Y_{29}X_{25} \\ & + \boxed{Y_{48}Y_{43}Y_{38}Y_{33}Y_{29}Y_{25}}\, C_{20} \end{aligned} \tag{9}$$

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

$$\begin{aligned} C_{48} = {} & Z_{48} \\ & + \boxed{W_{48}}\, C_{20} \end{aligned}$$

The collected Boolean expressions for the auxiliary carry functions and for the carry functions themselves for this particular 53-bit adder are given in tables 1.2 and 1.3.

The number of gating stages required to implement this design can be seen by examining figure 1.10. Each square box in the diagram represents one gating stage. Of the 238 stages used in the whole adder register, note that *only 26* are used to create the auxiliary carry functions. The other 212 stages are needed irrespective of how the carry digits are formed, because they comprise 53 sets of 4 stages for the augend, addend, carry, and sum digits.

TABLE 1.2. *Auxiliary carry functions*

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).

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

TABLE 1.3. *Carry functions*

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_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+Y_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}+C_{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}$$

As indicated previously, five clock phases are occupied, starting with the input digits and ending with the sum digits. As the adder is to be used for multiplications and divisions in a repetitive fashion requiring the recirculation of the sum digits back into one of the inputs with appropriate shifts, the clock pulses must occur in five phases to allow an addition cycle to be completed in 1 μsec.

The top line of table 1.4 gives some statistics on the number of components required for the adder represented in figure 1.10. Two other slightly different versions have been worked out in which fewer gates need to be driven by the most heavily loaded tube. As the table shows, these variations also require different proportions of components. Approximately 10,000 germanium diodes are required in each of these versions.

TABLE 1.4. *Number of components required*

Maximum load [1]	Number of stages	Tubes	Delay lines
			μsec
25	238	238	300
19	253	253	250
14	285	285	150

[1] Unit of load=one gate-load.

6. References

[1] A. W. Holt, An experimental rapid access memory using diodes and capacitors, Proc. Assoc. for Computing Machinery Conference, p. 133–142 (December 1952).

[2] R. J. Slutz, A. W. Holt, R. P. Witt, D. C. Friedman, High-speed memory developments at the National Bureau of Standards, NBS Circ. 551, p. 93–108 (January 1955).

[3] S. Greenwald, R. D. Haueter, S. N. Alexander, SEAC, Proc. Inst. Radio Engrs. **41,** p. 1300–1313 (October 1953). See also, NBS Circ. 551, p. 5–26 (January 1955).

[4] A. L. Leiner, S. N. Alexander, System organization of the DYSEAC, Trans. Inst. Radio Engrs.-Prof. Group Electron. Computers, **EC–3,** No. 1 (March 1954).

[5] A. L. Leiner, W. A. Notz, J. L. Smith, A. Weinberger, System design of the SEAC and DYSEAC, Trans. Inst. Radio Engrs.-Prof. Group Electron. Computers, **EC–3,** No. 2 (June 1954).

[6] R. D. Elbourn, R. P. Witt, Dynamic circuit techniques used in SEAC and DYSEAC, Proc. Inst. Radio Engrs. **41,** p. 1380–1387 (October 1953). See also, Trans. Inst. Radio Engrs.-Prof. Group Electron. Computers, **EC–2,** No. 1 (March 1953), and NBS Circ. 551, p. 27–38 (January 1955).

[7] R. K. Richards, Arithmetic operations in digital computers, p. 26–50 (D. Van Nostrand Co., Inc., New York, N. Y., 1955).

ERRATUM

Page 3, line 7 of the left-hand column should read: "Since the basic micro-operation. . ."