SYMBOLIC SYNTHESIS OF DIGITAL COMPUTERS

By

Irving S. Reed
Massachusetts Institute of Technology

In an ideal sense a binary digital computer or what might be called more generally a Boolean* machine is an automatic operational filing system. It is a machine which accepts information automatically in the form of words constructed from an alphabet of only two symbols, say 0 and I, the so-called binary coded words. For example a binary number is such a word. This information is stored or recorded in sets of elementary boxes or files, each containing one of the symbols 0 or I. This information is either transformed or used to change other files or itself as a function of the past contents of all files within the system. If the contents of all files within the system are constrained to change only at discrete points of time, say the points n (n = 1,2,3, ...), then the machine may be termed a synchronous Boolean machine. The discussion in this paper will be restricted to the synchronous Boolean machine.

It is evident from the above discussion that the content of each elementary box or file within the system is a two valued function, say A(t), of the real parameter time, t. Since the content of each file is constrained to change only at the discrete points of time $n\gamma$, then a suitable definition for A(t) is the following: Either A(t) = 0 or A(t) = I for $n\gamma \leq t < (n+1)\gamma$ where (n = 0 1,2, ...). In this definition the right continuity of A(t) has been assumed for the sake of definiteness. Let the function A(t) be called a file function.

Table I defines a unary and three binary operations of file functions, F(t) and G(t).

F(t)	G(t)	F'(t)	F(t)+G(t)	F(t)G(t)	F(t)⊕G(t)
0	0	I	0	0	0
0	I	I	I	0	I
I	0	0	I	0	I
I	I	0	I	I	0

TABLE I

From the definition of a file function and Table I the following theorem is now evident.

<u>Theorem 1</u>. The class of all file functions B is a Boolean algebra with respect to the operations ' , . , and +. B is an additive abelian group with respect to the operation ⊕ where the zero element is the file function, 0, which is 0 for all t, $t \geq 0$. More generally B is an algebraic or Stone* ring with respect to the operations . and ⊕ where the unit element of the ring is the file function, I, which is I for all $t \geq 0$.

By Table I, or otherwise, it is not difficult to establish the following useful identities, relating the operations ' , . , + and ⊕ between file functions F and G:

$$(1) \qquad F + G = F \oplus G \oplus FG,$$
$$F \oplus G = F'G + FG' \quad \text{and}$$
$$F' = F \oplus I$$

Now to every file function F(t) of the synchronous Boolean machine there corresponds the complementary file function F'(t). Moreover, F(t) uniquely determines F'(t) and conversely. Thus it may be assumed that the complementary function of any file in a synchronous Boolean machine is always available within the machine for reference.

---

* In honor of the great British mathematician, George Boole, who originally discovered the algebra needed to describe such machines.

* Marshall Stone was the first to establish that every Boolean algebra is a ring. For reference see [1].

In fact, it may be assumed that there are 2 P files containing the functions $F_1(t)$, $F_2(t)$ ..., $F_P(t)$; $F_1'(t)$, $F_2'(t)$, ... $F_P'(t)$ where the first P files completely determine the contents of the remaining P files. From this fact and the description of the synchronous Boolean machine the following set of time difference equations characterize the behaviour of the synchronous Boolean machine as a function of time:

$$(2) \qquad A_j(t+\gamma) = f_j(A_1(t), A_2(t), \ldots A_N(t); W_1(t), W_2(t), \ldots, W_M(t)), \ (j = 1, \ldots N),$$

where the initial condition is $A_1(0)$, $A_2(0)$, ... $A_N(0)$, the functions $W_1(t)$, $W_2(t)$, ..., $W_M(t)$ are the given input file functions or forcing functions, the functions $A_1(t)$, $A_2(t)$, ..., $A_N(t)$ are the dependent file functions of the system and $P = N + M$. The system of time difference Boolean equations(2) will be called the synchronous Boolean system. Within the particular class of Boolean functions of a real variable t, the Boolean algebra B of the file functions, there are 2^N possible solutions of (2), depending uniquely on the 2^N initial conditions (see [2] and [3].) If the forcing functions of (2) had been chosen from certain other subclasses of the class of all Boolean functions of a real parameter t it is possible to show that (2) has solutions in classes other than B. However, the present discussion will be restricted to the class B defined herein.

For brevity in notation let a function $F(t)$ be denoted sometimes by F and in particular denote $G(t+\gamma)$ by $G[\gamma]$. This notation should not lead to any confusion since the subsequent arguments of this paper will be true for all $t \geq 0$. If $f_j = f_j(A_1, \ldots, A_{Nj} W_1, \ldots W_M)$ for $(j = 1,2, \ldots N)$ of (2) is expanded in the disjunctive normal form it is evident that f_j may be expressed in the form

$$(3) \qquad f_j = g_j A_j' + {}_0g_j' A_j \quad \text{for } (j = 1,2, \ldots N)$$

where

$$g_j = f_j(A_1, \ldots A_{j-1}, 0, A_{j+1}, \ldots A_{Nj} W_1 \ldots W_M) \quad \text{and}$$

$${}_0g_j' = f_j(A_1, \ldots, A_{j-1}, I, A_{j+1}, \ldots A_{Nj} W_1 \ldots W_M).$$

Equation (3) with (1) obtains

$$(4) \qquad \begin{aligned} f_j &= g_j(I \oplus A_j) \oplus ({}_0g_j \oplus I) A_j \\ &= A_j \oplus [\ (g_j + {}_0g_j) A_j \oplus g_j] \\ &= A_j \oplus [\ g_j A_j' + {}_0g_j A_j] \end{aligned}$$

Equation (4) and the abelian group property of the operation $\oplus$ transforms (2) into

$$(5) \qquad \begin{aligned} \triangle A_j &= A_j[\gamma] \oplus A_j = (g_j \oplus {}_0g_j) A_j \oplus g_j \\ &= g_j A_j' + {}_0g_j A_j = a_j \quad \text{for } (j = 1,2, \ldots N) \end{aligned}$$

where $\triangle$ is called the change operator, defined by $\triangle F(t) = F(t+\gamma) \oplus F(t)$ for all $t \geq 0$. The system of equations, given by (5) are the change equations of the synchronous Boolean system (2). The word change is derived from the fact that for any file function $F(t)$, $\triangle F(t) = I$ if, and only if, $F(t) = 0$ and $F(t+\gamma) = I$ or $F(t) = I$ and $F(t+\gamma) = 0$. That is, $\triangle F(t) = I$ only when the symbol in the file with function $F(t)$ changes. Although system (5) is merely an algebraic transformation of system (2), it will be seen presently that a_j is precisely the Boolean function of file functions needed as the input function for a clocked one input electronic flip-flop, when considered as a possible dual file to store the functions A_j and A_j'.

When an Eccles-Jordan multivibrator has been designed to change only when triggered by a voltage pulse or spike of time duration ϵ, such that $\epsilon \ll \gamma$, this bistable device is called a flip-flop. The output of the flip-flop, which consists ordinarily of two triode vacuum tubes (valves) and associated circuits, is a voltage from each plate or anode of the tubes. There are essentially only two voltages which may appear on one of the anodes, except during the time of change. If the voltage of one anode is high, the voltage on the other is low and conversely. Let a high voltage on an anode represent the symbol I and the low voltage 0 and suppose the time for switching to be instantaneous in such a manner that the output voltage function is right continuous. Moreover, suppose the voltage pulse function E(t), triggering the flip-flop is such that the flip-flop will change state at every point of time $n\tau$ $(n = 1,2,3,4, \ldots)$ and at only these points. Then the output of one anode is a file function F(t) and the voltage on the other anode is $F'(t)$, the complementary file function.

Now the file function F(t), constructed above, is very specialized in that it is one of the two possible file functions which changes at every allowable point of time $n\tau$ $(n = 1,2,3, \ldots)$. To allow the flip-flop to hold any file function suppose a(t) is a Boolean function of file functions such that a(t) is a file function of the same physical nature as the output of the flip-flop, i.e. two voltage levels. Let a(t) gate with the trigger or clock function E(t) in such a manner that the condition a(t) = 0 will inhibit the next pulse of E(t) from triggering the flip-flop and the condition a(t) = I will allow the next pulse of E(t) to trigger or change the output of the flip-flop. Then the output function of the flip-flop which will now be denoted by A(t), results clearly from Table 2.

a(t)	A(t)	$A(t+\tau)$
0	0	0
0	I	I
I	0	I
I	I	0

TABLE 2.

From Tables 2 and 1, one obtains

$$(6) \qquad A(t+\tau) = a'(t)\,A(t) + a(t)\,A'(t) = A(t) \oplus a(t)$$

as the time difference equation for the ouput file function A(t) of a clock gated flip-flop with file function input a(t). By the group property of the binary operation $\oplus$ equation (6) may be solved for a(t), obtaining,

$$(7) \qquad a(t) = A(t+\tau) \oplus A(t) = \Delta A(t) \ .$$

Equations (6) and (7) with initial condition A(0) are called the equations of the clocked one input flip-flop. It is interesting to note that (7) is a true linear operator equation in A for clearly the operator Δ is an automorphism of the ring B and any dependence of a(t) on A(t) is necessarily linear.

If equations (5) are compared with the change equation (7) of the clocked flip-flop, one observes the fact that if the changes a_j have the right physical nature to act as inputs to flip-flops, that N flip-flops will act as files for $A_j(t)$ and at the same time analyze the synchronous Boolean system (7) for all $t > 0$. It is well known that Boolean function of functions with two voltage levels can be physically realized as a function with two voltage levels. For example, diode gating networks will realize such networks. The following theorem is now evident:

<u>Theorem 2</u>: Every synchronous Boolean system is representable by an electronic synchronous Boolean machine which consists only of clocked one input flip-flops and devices for producing sums and products of the outputs of the flip-flops and conversely.

Once a desired synchronous Boolean machine or automatic filing system has been conceived symbolically as a synchronous Boolean system in the form of (5), Theorem 2 assures one that a physical realization of the desired machine is possible. Therefore, in the initial synthesis of a digital computer it is desirable to concentrate one's attention on the abstract model of the digital computer. Except for certain electronic problems of synchronization, the synchronous

Boolean system is sufficient to represent the abstract model of the desired binary digital computer or two symbol automatic operational filing system. Thus, in this paper the symbolic synthesis of a digital computer may be regarded as equivalent to the synthesis of a synchronous Boolean system.

If one input clocked flip-flops are to be used to analyze a known or desired synchronous Boolean system, it is necessary by (5) and (7) that one obtain the change equations (5) of the system. The desired changes ΔA_j for $(j = 1, 2, \ldots, N)$ of the file function A_j are functions of $A_1, A_2, \ldots A_N ; W_1, \ldots W_M$. Thus, as it is well known, each ΔA_j may be obtained from either the disjunctive or conjunctive normal expansions or its equivalent valuation (truth) tables. A known or desired solution of this derived system of equations is entirely dependent on the initial condition imposed on the system.

In order to illustrate this procedure of synthesis, consider the desired operation of an unusual type of three stage counter, given in Table 3.

The section of the table devoted to the changes, $\Delta A_1, \Delta A_2, \Delta A_3$ is obtained directly from the first two sections of the table. For instance, I is placed in the third row of the column devoted to ΔA_1 since in that row $A_1 = I$ and $A_1[\Upsilon] = 0$. By using the valuation (truth) table approach and the rules of Boolean algebra this counter has the following set of change equations:

$$\Delta A_3 = (A_3'A_2'A_1)' = A_3 + A_2 + A_1'$$

$$\begin{aligned}\Delta A_2 &= A_3'A_2'A_1' + A_3'A_2A_1 + A_3'A_2A_1' + A_3A_2'A_1 \\ &= A_3'A_2'A_1' + A_3'A_2 + A_3A_2'A_1 \\ &= A_3'(A_1' + A_2) + A_3A_2'A_1\end{aligned}$$

$$\begin{aligned}\Delta A_1 &= (A_3'A_2'A_1' + A_3A_2'A_1' + A_3'A_2'A_1')' \\ &= (A_2'A_1' + A_3'A_2'A_1)' \\ &= (A_2' [A_1' + A_3'])' = A_2 + A_1A_3\end{aligned}$$

These equations are the input equations to three one input clocked flip-flops, designated to hold the functions A_1, A_2 and A_3. It may be noted that this counter illustrates a classification of all possible types of states of a counter. By Table 1, if the initial condition of the system is any one of the states or configurations of period 4Υ, the remaining three states follow in succession; then back to the initial state, forming a loop of period 4Υ. From the table this counter has as well a loop of period 2Υ, a transient state and a bound state.

If a more complex computing system is desired, where at the outset very little is known about the end product, one must first hypothesize sets of files and possible operations between them. Then hypothetical combinatiorial relationships may be established in such a manner that the desired in-

A_3	A_2	A_1	$A_3[\Upsilon]$	$A_2[\Upsilon]$	$A_1[\Upsilon]$	ΔA_3	ΔA_2	ΔA_1	Classification of States
0	0	0	I	I	0	I	I	0	Periodic States of period 4
I	I	0	0	I	I	I	0	I	
0	I	I	I	0	0	I	I	I	
I	0	0	0	0	0	I	0	0	
0	I	0	I	0	I	I	I	I	Periodic State of period 2
I	0	I	0	I	0	I	I	I	
I	I	I	0	I	0	I	0	I	Transient State
0	0	I	0	0	I	0	0	0	Bound State

TABLE 3.

put and output relationships of the system are to some degree satisfied. This procedure is aided conceptually by an understanding of the synchronous Boolean system and space-time charts of the various hypothesized files. If the first hypothetical system does not satisfy the desired notions for the system either the notions or the hypothesis or both must be modified and then the above procedure must be reiterated. If one assumes that the above iterative planning procedure arrives at a suitable compromised system, satisfying the desires of the planner, and if the planning has leaned heavily on the symbolization of the contents of the files of the system, the planner will find usually that writing and reducing a set of change equations such as (5) will be accomplished already in many cases and almost mechanically in other cases. The minimization techniques of the Harvard group are sometimes of value in the algebraic reduction of the equations. However, care must be taken in not using these techniques too early in the game, for as it often happens, there may be large common parts to two or more change equations of the system which, in many cases, become lost when an individual function minimization technique is applied before noticing these common parts. To convert the change equations into electronic realizations is a fairly well discussed problem (see [4], [5] and [6]).

In conclusion, it should be mentioned that the use of two input clocked flip-flops as files is similar to the one input clocked flip-flop and, in fact, they are algebraically as well as electronically very intimately related. There are many topics concerning the subject of this paper which have been left out for the sake of brevity. More complete discussions are deferred to later papers.

The author wishes to express his deepest gratitude to his former teacher, Prof. E.T. Bell, who first taught Boolean algebra to the author, and encouraged these particular directions to the subject. Much acknowledgement goes to F.F. Steele who through long personal contact with the author did much by his keen physical insight to bring about the ideas contained in this paper. Finally, this paper probably would not have been written without the consecutive support of Northrop Aircraft, Inc., Computer Research Corp., and lastly the Lincoln Laboratory of the Massachusetts Institute of Technology.

Bibliography

1. Garrett Birkhoff, Lattice Theory, American Math. Soc. Coll. Publications. (1948)

2. I.S. Reed, Some Mathematical Remarks on the Boolean Machine, Technical Report No. 2, Project Lincoln, Mass. Inst. of Technology, (19 Dec. 1951)

3. I.S. Reed, Boolean Functions of a Real Variable and its Applications to a Model of the Digital Computer and Discrete Probability, a forthcoming Project Lincoln Report, M.I.T.

4. R.E. Sprague, Techniques in the Design of Digital Computers, Association of Computing Machinery (March 1951)

5. R.C. Jeffrey, I.S. Reed, The Use of Boolean Algebra in Logical Design, Engineering Note E-458-1, Digital Computer Laboratory, M.I.T. (28 April 1952)

6. R.C. Jeffrey, I.S. Reed, Design of a Digital Computer by Boolean Algebra, Engineering Note E-462, Digital Computer Laboratory, M.I.T. (20 May 1952)