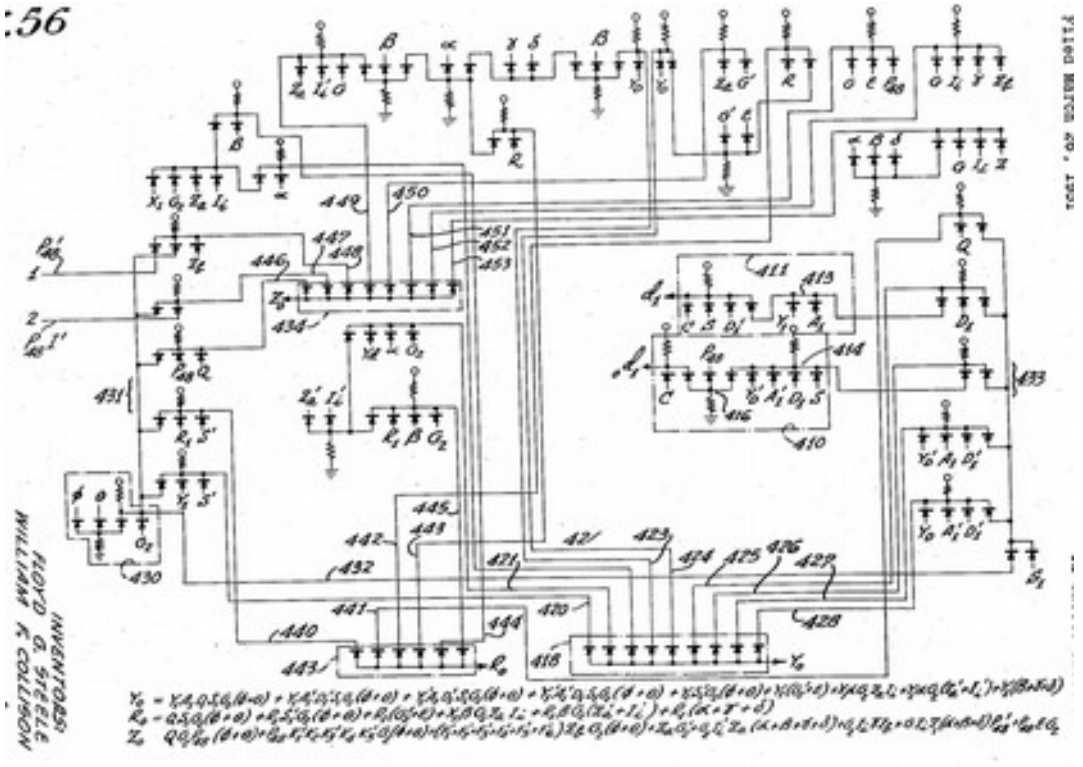
Computers as Boolean machines: Between operational clarity and physical reality.



Maarten Bullynck
(Paris 8, Dépt. de mathématiques & histoire des sciences & UMR IDHE.S)

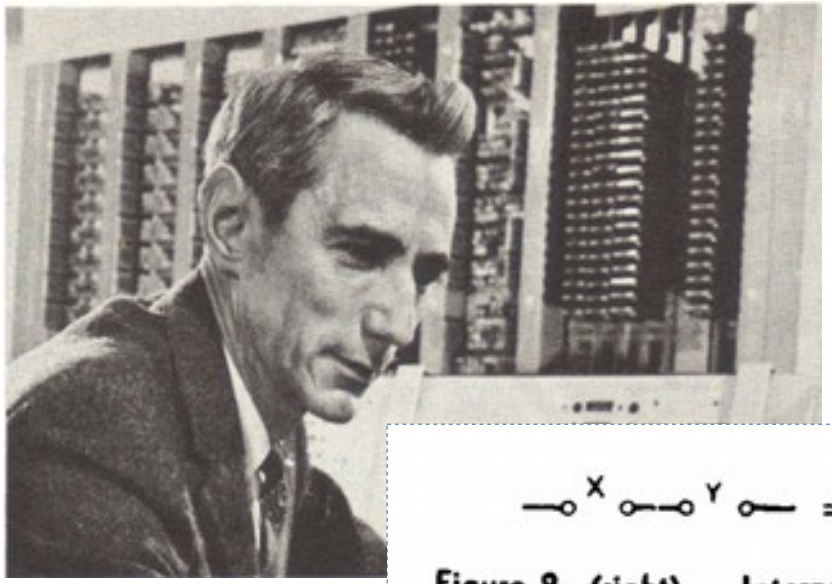
PROGRAMme workshop, June 2019, Lille

Overview

1. *Shannon's Boolean circuit analysis and the engineer's "peculiar sort of frustration"*
2. *East Coast's block diagram approach of designing computers*
3. *West Coast's equation approach to designing computers*
4. *Enter the transistor & Automation of computer design*
5. *Some Conclusions*

Shannon's Boolean circuit analysis and the engineer's "peculiar sort of frustration"

Shannon 1938: A symbolic analysis of relay switching circuits



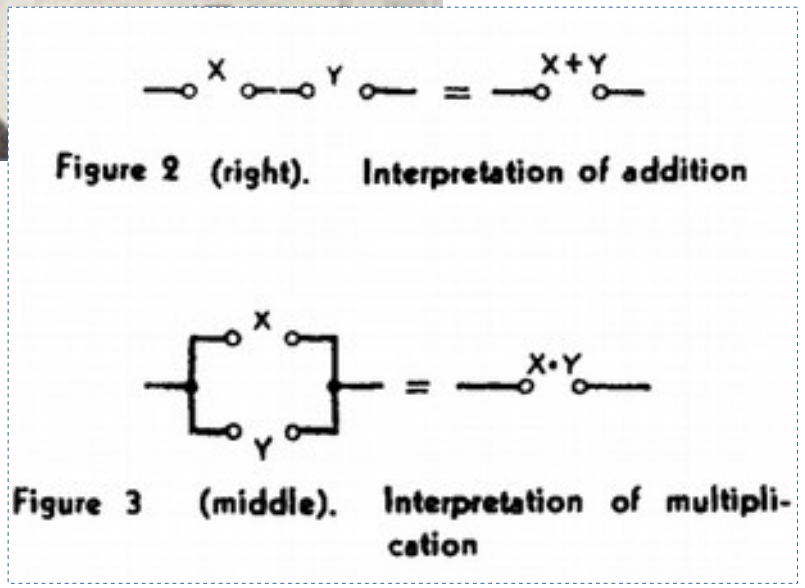


Figure 2 (right). Interpretation of addition

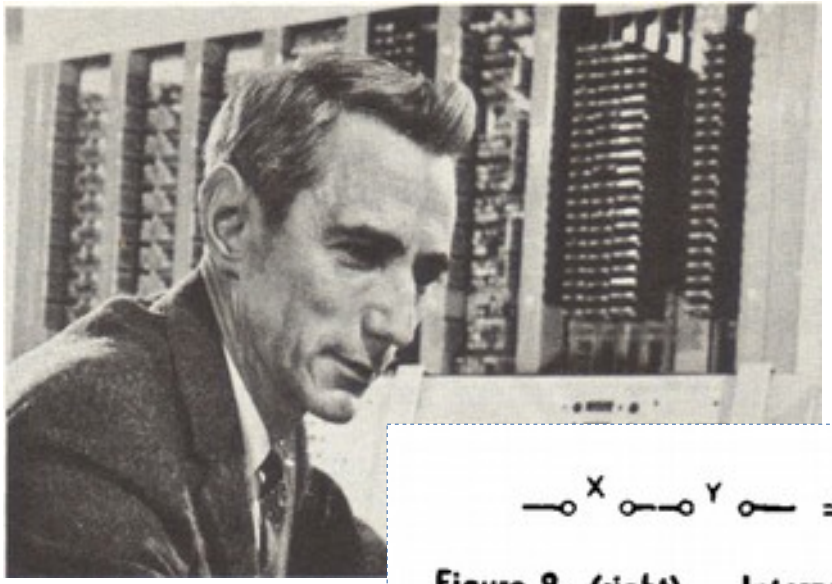
Figure 3 (middle). Interpretation of multiplication

"The paper was a landmark in that it helped to change digital circuit design from an art to a science" (Goldstine 1977)

Or wasn't it?

(cf. Bullynck, "Switching the Engineer's Mind-Set to Boolean: Applying Shannon's Algebra to Control Circuits and Digital Computing (1938–1958)", in Haigh (e.d), Exploring the Early Digital, 2019.)

Limits of Shannon 1938



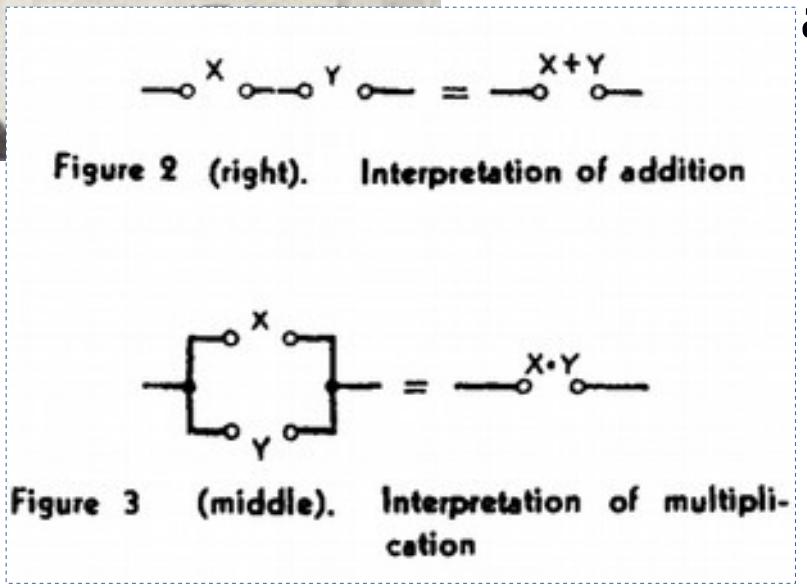


Figure 2 (right). Interpretation of addition

Figure 3 (middle). Interpretation of multiplication

- The paper does “symbolic analysis”, ***not synthesis***
- Mainly useful to ***simplify*** already known circuits
- Many circuits ***not amenable to*** all-or-nothing ***Boolean*** analysis
  - Analogue components
  - Filters
  - Time-dependent circuits such as flip-flops or sequential machines
  - ...

Shannon 1938 and the engineer's "peculiar sort of frustration"

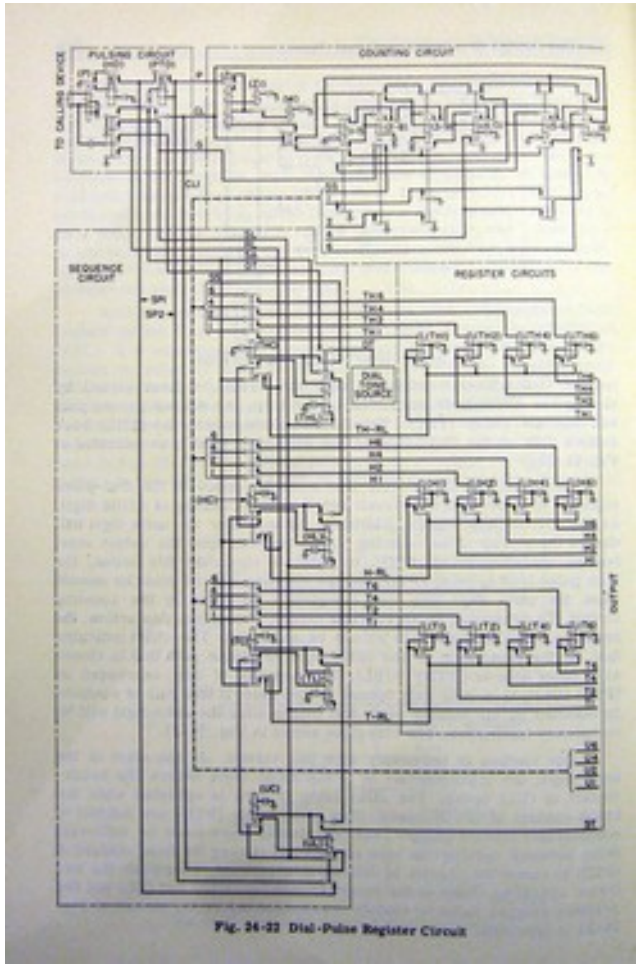


Fig. 24-22 Dial-Pulse Register Circuit

"Switching algebra could be used directly for the simplification of contact networks", but "the situation was different with respect to the synthesis of a network" (Caldwell)

"design tools for the synthesis of the building blocks [are] outside the province of Boolean algebra." (Washburn)

"when problems of any magnitude were at tempted, the method broke down both because of the difficulty of writing word statements and because of the difficulty of converting bulky word statements into algebraic expressions, [when] one attempted to use the method, there arose a peculiar sort of frustration." (Karnaugh)

"the problem of synthesizing really large circuits has not been touched — one wonders whether it is really possible to do this with any method that relies on the use of a truth-table without making use of automatic design aids inasmuch as large truth-tables become unmanageable." (Mealy)

East Coast's block diagram approach of designing computers

The block diagram method

"From a statement of circuit requirements, a functional plan is developed in terms of ***known*** or conceptually evident ***circuit blocks***, representing simple circuits similar to single-function circuits [...] as the design proceeds, the functional blocks are ***coordinated and integrated*** to the point where a ***comprehensive block diagram*** of the proposed circuit exists. [...] The most satisfactory approach to developing a block diagram is to start with a few main subdivisions of the over-all circuit and successively break these down until each block represents a unifunctional circuit. [...] In a surprisingly large number of cases in the planning, familiar functional circuits are found to applicable. When a ***new circuit concept*** is encountered, the designer can usually recognize whether an appropriate circuit can readily be designed. If this is so, the circuit can be ***designated on the diagram and the design deferred until later***. […] the attempt should be made to obtain the simplest and most efficient arrangement among the various blocks. [...] the designer should from the start make a conscious effort to familiarize himsself with different types of basic circuits already in use and to classify them in terms of function. In this way he develops a constantly growing ***'catalogue' of circuit building blocks*** which expedites is planning and design of circuits."" (Keister et al 1951)

The block diagram method

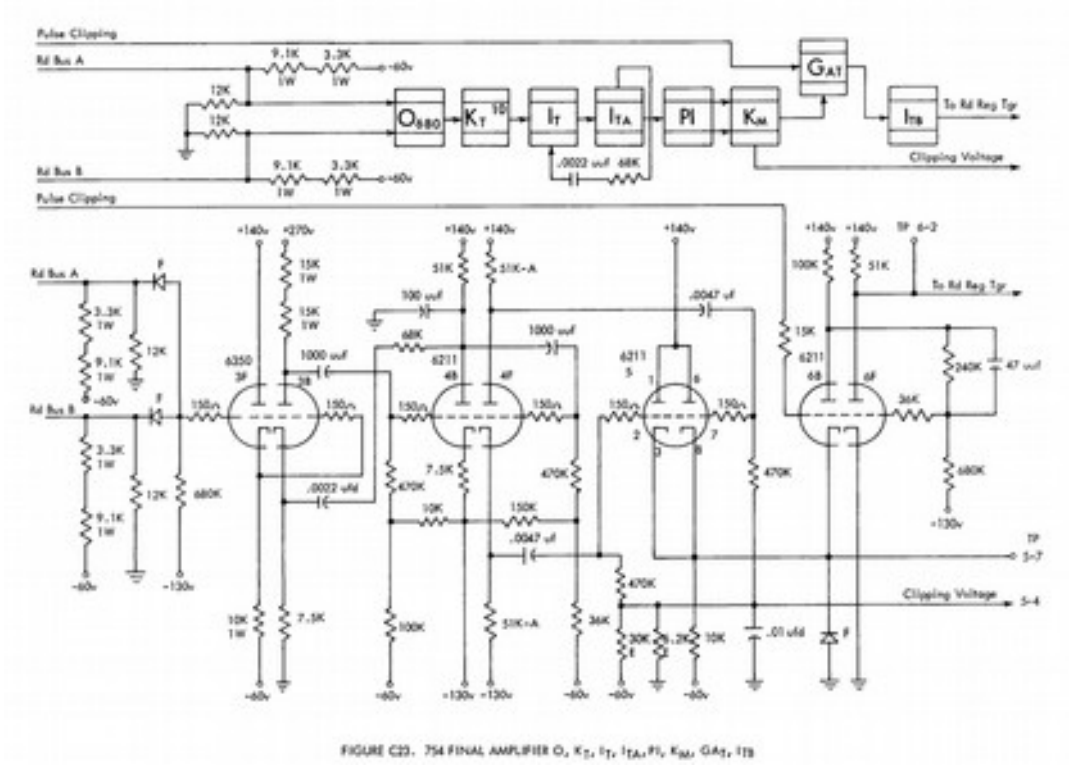
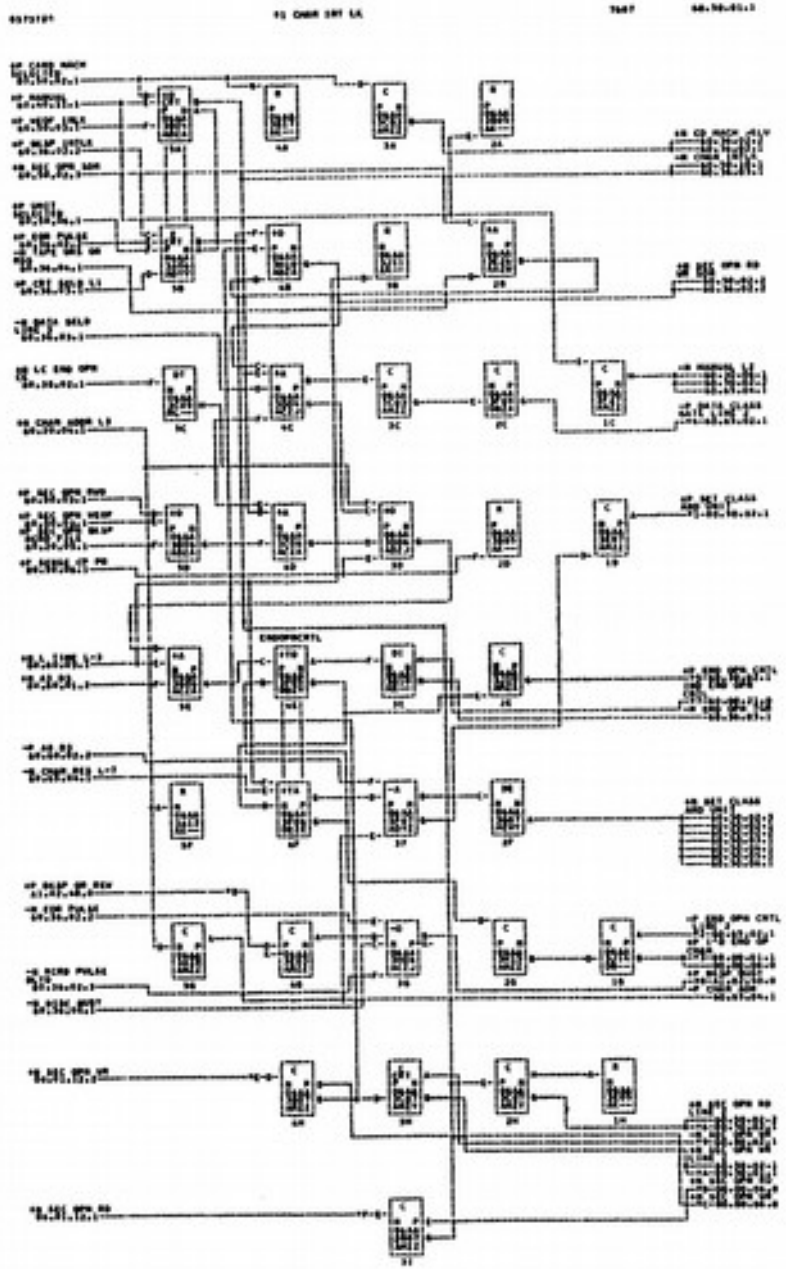
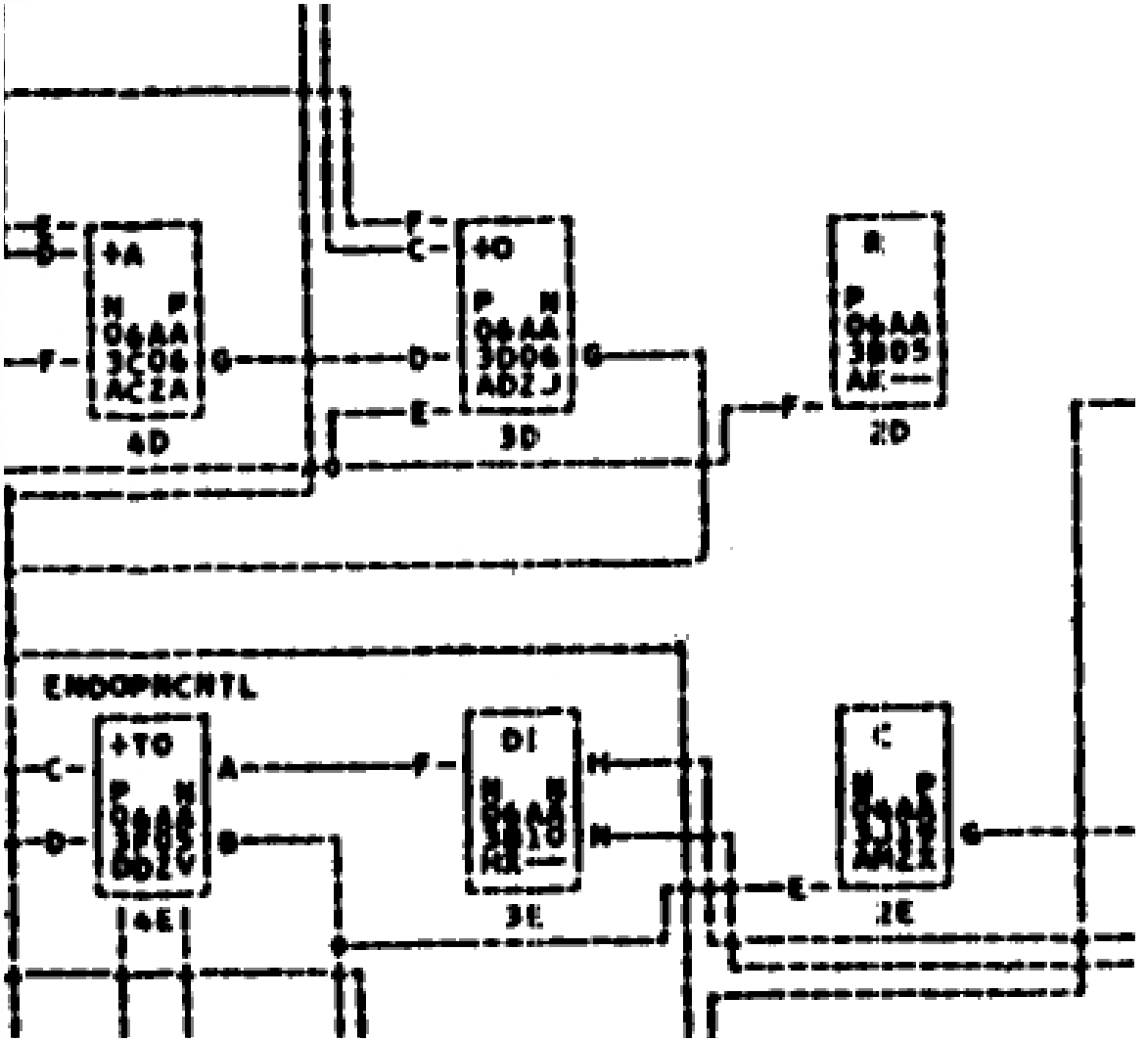


FIGURE C23. 754 FINAL AMPLIFIER O, K_T, I_T, I_{TA}, PI, K_M, GA_T, I_{TB}

- Used at all the ***big companies:*** Bell, IBM, Univac, MIT, …
- Inherent to the approach is the ***"black box"***: "When a new circuit concept is encountered, [...] the circuit can be designated on the diagram and the design deferred until later."
- Usage of ***standard*** blocks in the design
- ***Standardisation*** of block diagrams practices at Bell Labs and IBM



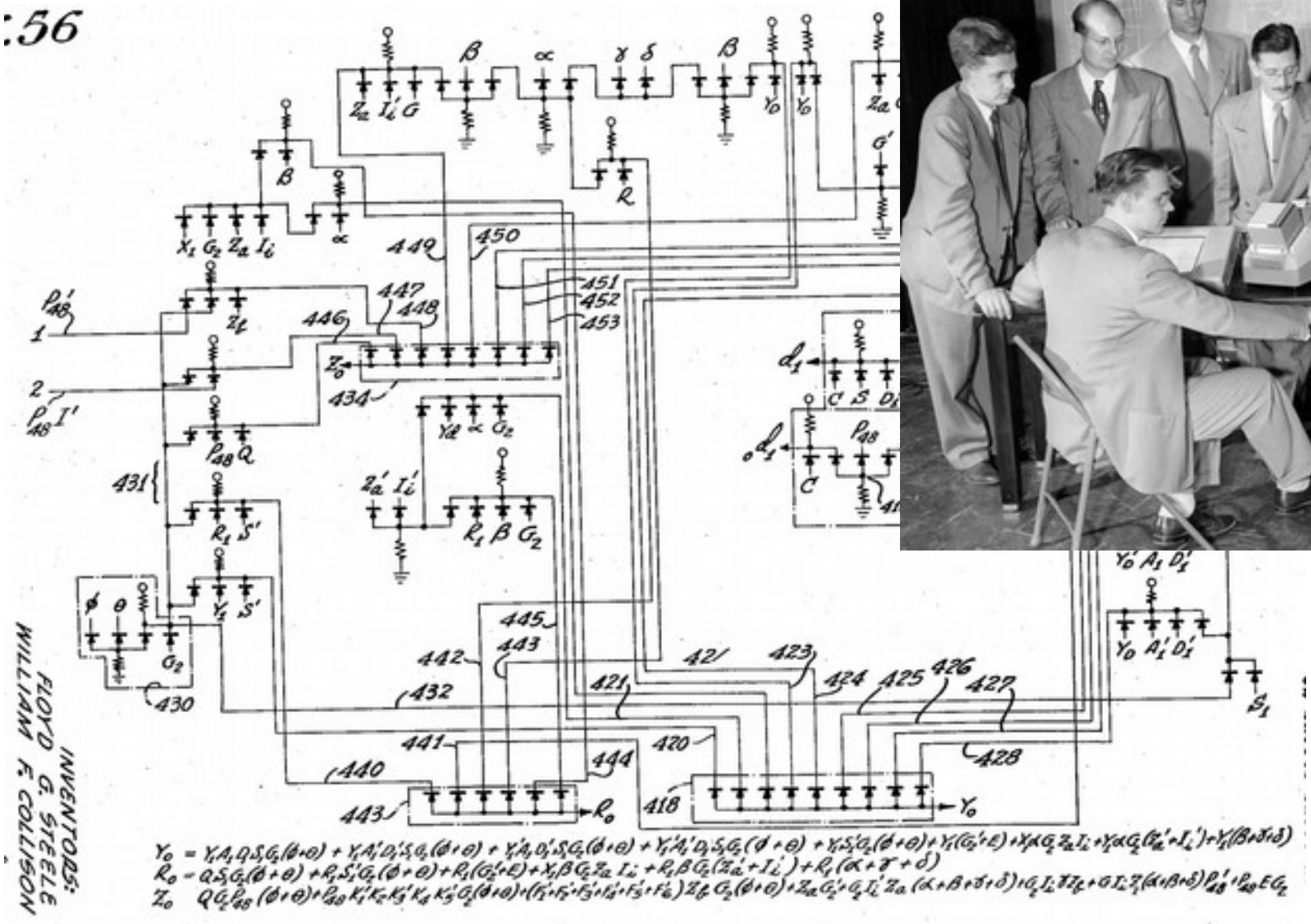

+A
4D
+D
3D
B
2D
ENDOPNCNTL
+TO
4E
D1
3E
C
2E

West Coast's Boolean equations approach of designing computers

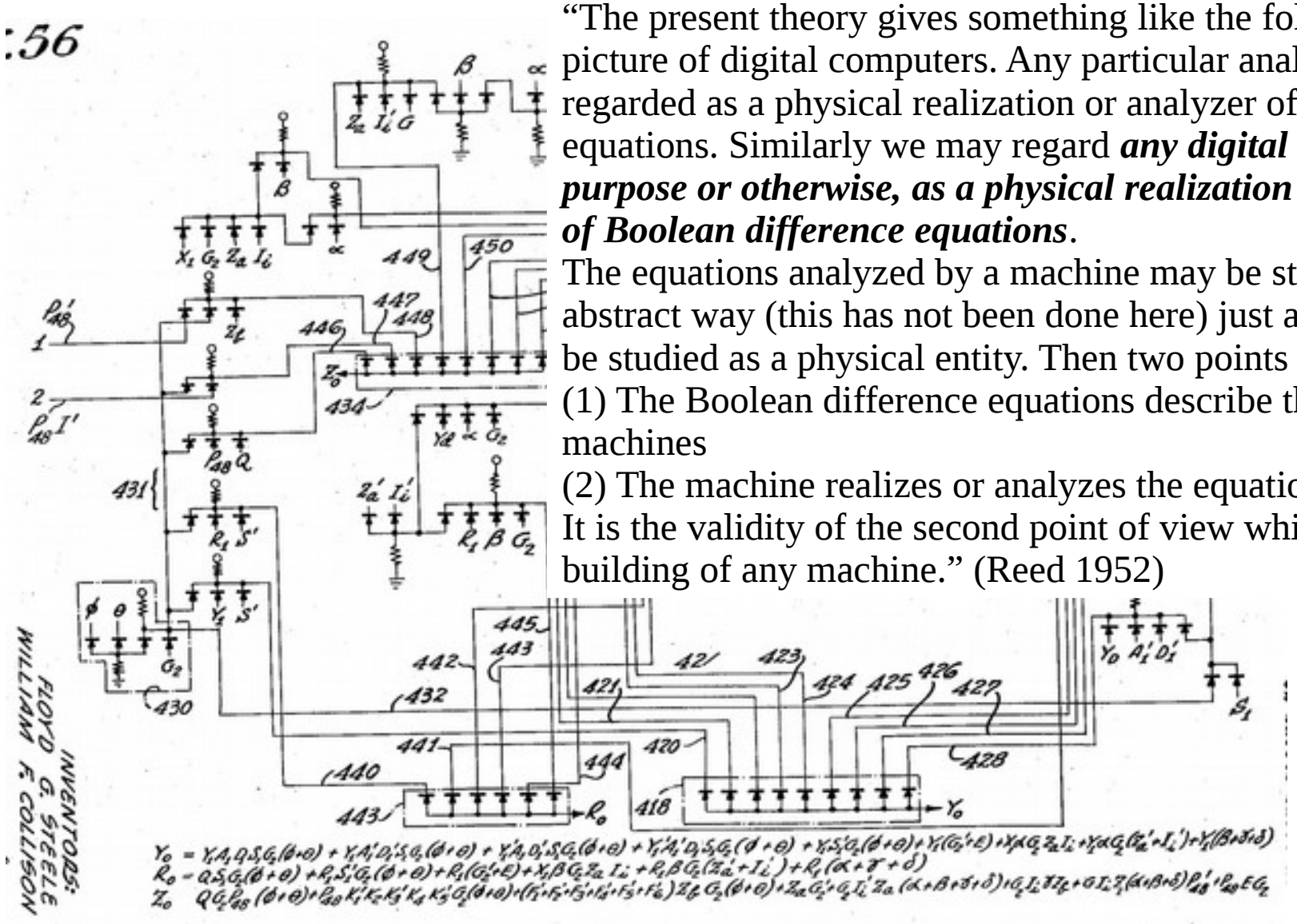
The Boolean equations approach

"Usually computer design is carried out by drawing block diagrams showing the circuit elements and their connections. The construction of a block diagram from a set of statements describing a computer process is often tedious, owing to the difficulty of tracing the path of the signals through the maze of circuit elements involved. An a***lgebraic representation of computer processes*** has been developed in which there is a ***direct correspondence between the algebraic operations and the circuit elements***. This algebra exhibits the various signals involved and shows how they are related to the primary signals. It applies not only to the state of a computer at a definite time but also takes account of the dynamic behavior of the computer, exhibiting the time relationships between the signals in the various parts of the computer. The translation of a verbal description of a computer process into a list of algebraic equations is usually much simpler than the direct construction of a block diagram, for the equations show more clearly the role of the various signals and their relative timing. A block diagram can be constructed from the equations in a matter of minutes." (Nelson 1951)

Floyd Steele's Digital differential analyzer 1947-51



The Boolean machine



"The present theory gives something like the following general picture of digital computers. Any particular analog computer may be regarded as a physical realization or analyzer of a set of differential equations. Similarly we may regard ***any digital computer, general purpose or otherwise, as a physical realization or analyzer of a set of Boolean difference equations***.

The equations analyzed by a machine may be studied in a perfectly abstract way (this has not been done here) just as the machine may be studied as a physical entity. Then two points of view are possible:

(1) The Boolean difference equations describe the working of the machines

(2) The machine realizes or analyzes the equations.

It is the validity of the second point of view which motivates the building of any machine." (Reed 1952)

The Boolean equations approach

"Many computer engineers have the mistaken impression that the primary value in logical equations is to be found in the purely logical reduction of a proposition in the manner previously described. It should be noted, however, that there are two very distinct aspects or phases to logical design. These the author prefers to call strategic design phase and tactical design phase. ***Tactical design*** is concerned primarily with the ***determination of final form*** of an expression once the function and purpose of the expression has been ascertained. Thus, the purely ***logical reduction*** falls into the tactical design phase. Of much greater importance, however, is the ***strategic design*** phase, which ***determines the function of each building block, the configuration of signal flow paths, and the overall pattern for machine structure.*** **It is in the strategic design that the logical equations really prove their value**" (Hesse 1958)

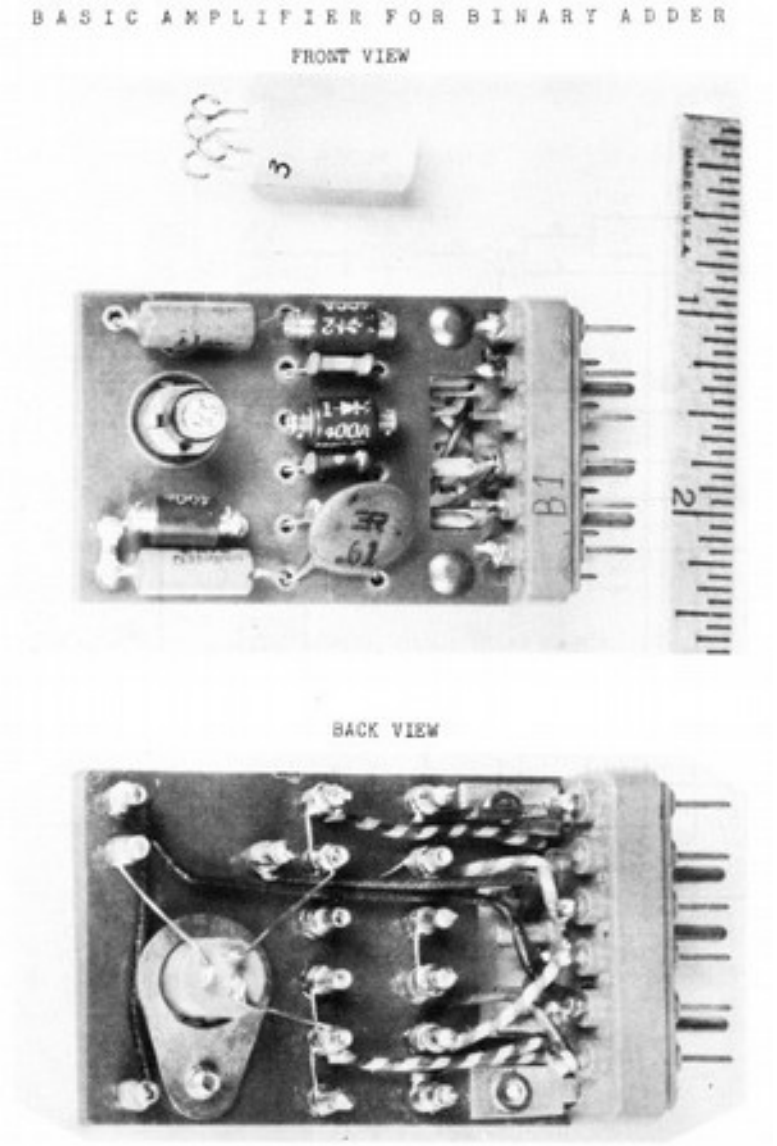
The Boolean equations approach

"The design techniques described here require the extensive use of Boolean algebra, and the complete description of a ***computer designed in this way is given by a set of Boolean equations***. Furthermore, the ***circuit components*** employed here are almost entirely ***synchronous***, that is, the signal wires connecting circuit elements contain meaningful information only at discrete intervals of time, usually called clock-pulse times. A great many very successful computers have been constructed without the use of Boolean algebra, and many computers employ asynchronous circuits which require no clock pulses. Most of the current literature on computers refers to these other methods of design.

However, several years of experience in computer work have convinced me that there are a great many advantages to be gained in carrying out design in the way described in this book. The computer systems now desired are so complicated and the logical properties of circuit components are so intricate that it is often difficult to derive computer circuits without using Boolean algebra; and although much work is being done on the synthesis and analysis of asynchronous computer circuits, such circuits have inherent timing problems which tend to make their interconnection difficult and which thus obscure their intended functions." (Phister 1958)

Enter the transistor & Automation of computer design

The transistor



- 1947 Semiconductor physics
- 1951 Beginning of industrial production and use for digital computing
- 1955-1960 First transistorized computers, both special-purpose and general-purpose
- 1960s beginnings of (large) integrated circuits

- Problems of transistor circuits design:
  “The properties of the two devices [vacuum tubes and transistors] are so radically different … some of the transistor circuits bear little resemblance to vacuum tube circuits designed to perform the same function” (Wallace and Raisbeck, 1951)
  → “equivalent circuits”

The transistor

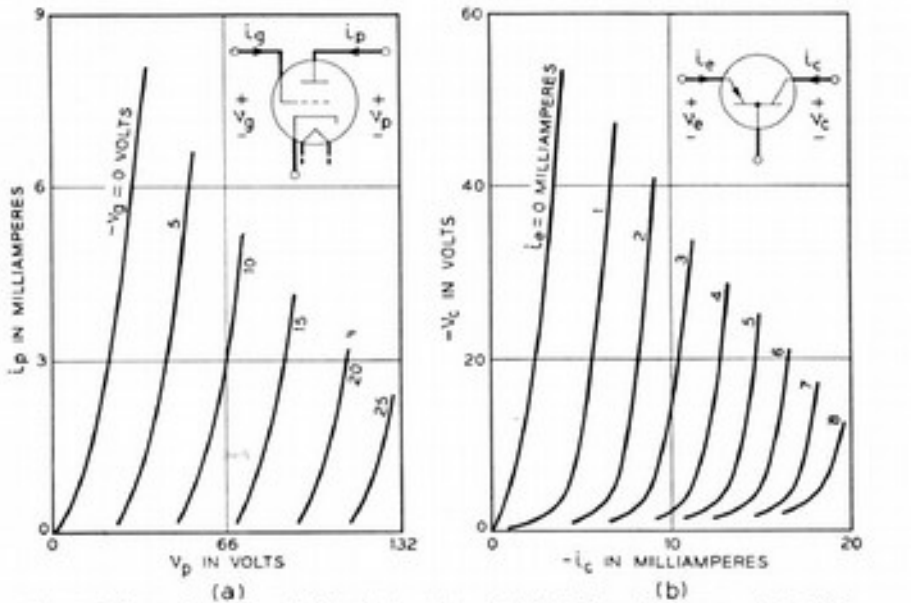
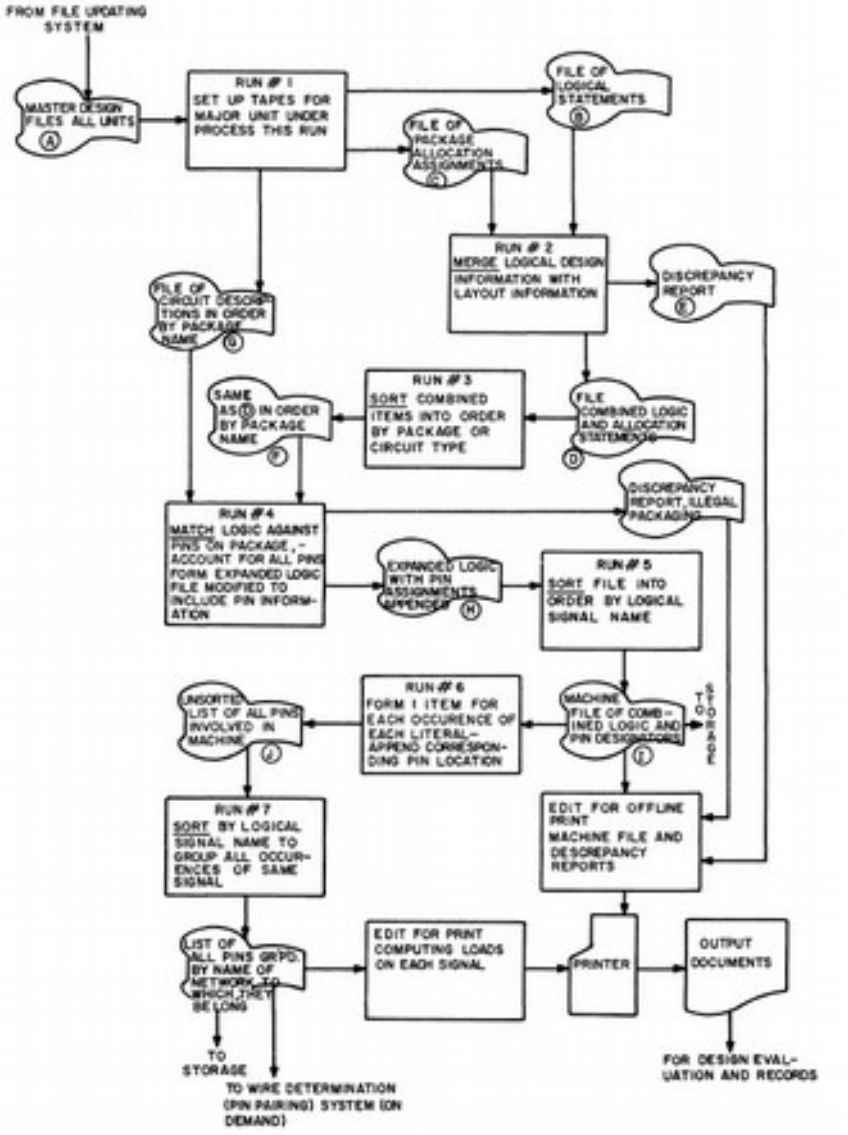


Fig. 1—The static characteristics of a transistor look like those of a vacuum tube triode provided transistor currents are compared with vacuum tube voltages and vice versa.

- “In principle, one needs no knowledge of the physics of the transistor in order ***to treat it circuitwise, any “black box” with the same electrical behavior*** at its terminals would act the same way.[...] we need go only to the literature of electrical engineering to find much practically useful information on properties of circuits which could be built around the unit” (Ryder and Kircher 1949)
- “We shall use the word "***circuit" to refer both to physical circuits and to abstract representations*** of circuit requirements (such as flow charts). The latter of course, may correspond to many physical circuits.
  We will be concerned mainly with a technology in which these assumptions are nearly satisfied — In this technology, one uses AND gates (with or without inhibiting inputs), OR gates, delay lines, and amplifiers. For our purposes, we may ignore the need for amplifiers. ...The ***properties of these circuit blocks are defined by the algebraic expressions***” (Mealy 1955)

Automation of computer design



- Mid 1950s “simulation” of circuits on a general-purpose computers
- Mid 1950s beginnings of automation of synthesizing transistor packages and modules
- After 1960, general development of fully blown systems to automate computer design
  - Either by starting from Boolean equations
  - Or by starting from wiring diagrams

Automation of computer design

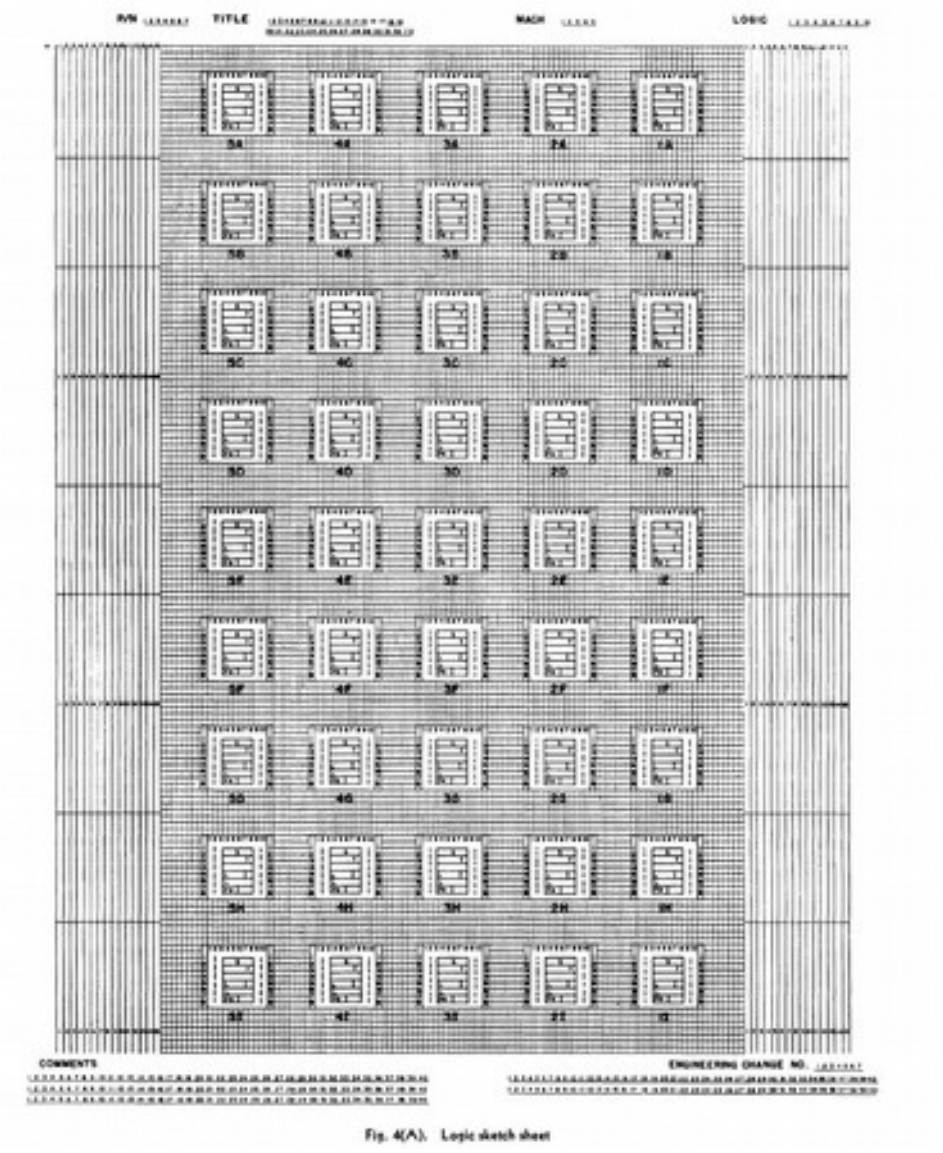


Fig. 4(A). Logic sketch sheet

- “The design philosophy which makes design mechanization a practical reality is the far-reaching standards program adopted ar IBM. The key to this ***standardization*** is the “standard modular system”” (Kloomok et al. 1958)
- “Inasmuch as the first step in the logic simulation process is the preparation of logic equations, it follows that the usefulness of the scheme is dependent on the possibility of representing the computer by logic equations. Most presently successful commercial computers, for example, have much “hodge-podge” in their logical design, and it is difficult, if not impossible, to represent such computers by sets of logic equations. It will be interesting to observe, as time passes, if the need for logic simulation forces a more complete adoption of the logic equation technique by those companies which so far have only made limited use of it” (Richards 1960)

The East Coast West Coast discussions

EAST VS. WEST

DIAGRAMS VS. EQUATIONS

THE COMPUTER'S ANSWER TO A LONG-STANDING COMPUTER ISSUE.

For a decade East Coast and West Coast computer designers have been using different methods of representing computer logic—the Easterners with diagrams, the Westerners with equations.

$$LBSM^{*} = (LXA1)(LXA2^{*})(LFCA^{*}) + (LXA1^{*})(LXA2)(LFCA^{*}) + (LXA1^{*})(LXA2^{*})(LFCA) + (LXA1)(LXA2)(LFCA)$$

$$LFCAJ = (LXA1)(LXA2)$$

$$LFCAK = (LXA1^{*})(LXA2^{*})$$

In the example illustrated here, the diagram and the equation tell us exactly the same thing. Either represents a serial full adder where the sequence of pulses at the output, LBSM, will represent a serial binary number that is the sum of two serial binary input numbers occurring at LXA1 and LXA2. (The asterisks indicate binary complements; for example, whenever LXA1 is energized LXA1* is not, and vice versa. LFCA is a carry flip-flop.)

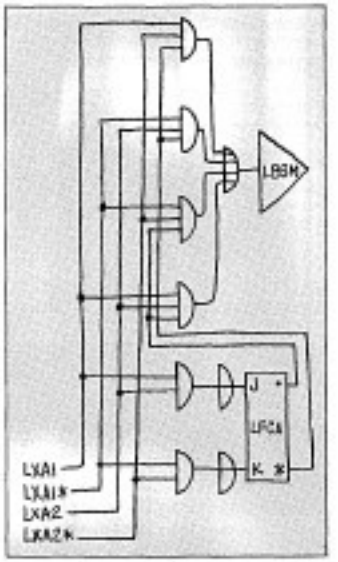
There are persuasive arguments on both sides. Eastern proponents of diagrams point out that the logical interconnections can be seen at a glance and followed through any number of stages by eye. The logical structure of an entire system can be understood from a diagram more directly and intuitively, they maintain, than from a set of equations.

The Western argument for equations goes like this. It's not true that diagrams communicate better to the viewer's intuition, except at first exposure. The human mind is highly adaptive. After working analytically with the equations for a while, the mind begins to operate intuitively in that symbology. Then the intrinsic superiority of equations over diagrams begins to make itself evident. One advantage, say the Westerners, is that equations can represent the same information more compactly and efficiently, as our illustration shows. Another is that equations lend themselves better to computer manipulation of logical design information.

As evidence of the latter advantage Westerners point to a recent achievement of some Litton Systems people: a completely mechanized procedure for translating logical designs into wiring lists, including operational simulation of the design to verify its accuracy. A procedure enormously facilitated by the computerizability of logical equations. It's easy to picture the benefits in cost, delivery schedules, reliability, price. Using only a partial development of this method Litton Systems recently brought a major computer system from concept to operation in less than a year.

Now under consideration at Litton: a machine that will accept as inputs a supply of standard computer components and a set of coded specifications defining the logical functions desired, and will crank out completely fabricated systems.

Maybe you think we've loaded the argument in favor of equations. You're right. But we're ready to listen to arguments on either side. Drop us a card. Or better still, drop in in person. You'll like the



imagination-stretching atmosphere generated by Litton management's appreciation of the rewards of creative controversy. And we have a few excellent opportunities for computer design people. Ask for S. L. Hirsch at Litton Systems, Inc., Data Systems Division, 6700 Eton Ave., Canoga Park, California.

An equal opportunity employer



LITTON SYSTEMS, INC.

A DIVISION OF LITTON INDUSTRIES

DATA HANDLING & DISPLAY SYSTEMS · GUIDANCE & CONTROL SYSTEMS · COMPUTER SYSTEMS · SPACE SCIENCE · MICROELECTRONICS · ADVANCED COMMUNICATIONS TECHNOLOGY

The paedagogic synthesis

From 1958 onwards, many textbooks on circuit design appear featuring Boolean algebra prominently:

> “Shannon’s work presented an opportunity to supplement skill with methods based on science, and thus to increase the productivity of circuit designers. Of greater importance though, was the ***possibility of developing a science*** which would promote deep understanding and nourish creative imagination … ***algebra needs no physical support,*** … this understanding … unifies the subsequent study of switching circuits in which many kinds of physical components are used” (Caldwell 1958)

Some conclusions

- The use of Boolean algebra for the synthesis of a machine relies on the ***abstraction from the physical substratum***
- In practice, this abstraction is made possible by ***standardization***, and its scaling is dependent on the physical properties
- Though Boolean logic has ***“unified” circuit theory*** for theoretical and paedagogic pruposes, its practice is much more limited
- Its initial use for simplifying circuits only got a wider, more general scope thanks to:
  - ***Sequential*** digital machinery
  - Increasing “***digitization***” of components
  - ***Semiconductor*** scalability
  - Need for ***automation*** due to increasing complexity
- Variants of the language of Boolean logic are developed, such as ***register transfer languages*** etc., to accommodate some of the intricacies of interfacing between logical structure and engineering design