Equational Logic

FRANK MARKHAM BROWN

Abstract—A combinational circuit realizing the switching function \( f(x) \) may be regarded as a solution verifier for the Boolean equation

\[ f(x) = 1. \]  

The output of the circuit is 1, that is, if and only if the input-vector \( x \) is a solution for (\( \ast \)). We use the term "equational logic" to denote an approach to circuit synthesis based on (\( \ast \)) rather than on the function \( f(x) \). The central problem of equational logic is to find a system of equations \( g_i(x) = h_i(x) \) \((i = 1, 2, \ldots, k)\), of the simplest possible form, that has the same solutions as (\( \ast \)). Given such a \( k \)-equation system, \( f(x) \) may be realized as the output of a \( k \)-wide digital comparator whose inputs are the \( 2k \) \( g \)'s and \( h \)'s constituting the system. The problem of finding a simple system of equations equivalent to a given equation was investigated more than a century ago by William Stanley Jevons, who called it "the inverse problem of logic."

It was thought by Jevons and other 19th century logicians that the inverse problem is "always tentative," i.e., that it does not admit of algorithmic solution. It is shown in this paper, however, that the inverse problem may be solved as a covering problem by use of the "table of consequences" of Poretsky. As presently formulated, this approach is limited in practical utility by the large size of the tables involved. It appears that a practical solution technique requires a reformulation of the inverse problem. To this end, a new formulation is presented; this formulation is based on the concept of a bilinear representation for \( f(x) \).

Index Terms—Boolean equations, Boolean matrices, combinational logic, digital comparators.

I. INTRODUCTION

The existing body of combinational switching theory may be called functional in the sense that the synthesis of a switching function \( f(x) \) is carried out by techniques that focus explicitly on the function itself.

The objective of this paper is to describe an alternative approach, which we term \( \text{equational} \). The process of realizing \( f(x) \), from the equation point of view, is one of producing a circuit which acts as a solution verifier for the Boolean equation

\[ f(x) = 1. \]  

The output of an equational circuit has the value 1, that is, if and only if the input-vector \( x \) is a solution for (1). The equational approach therefore shifts attention from the function \( f(x) \) to the equation \( f(x) = 1 \).

Given a function \( f(x) \) to be realized, the central problem of equational synthesis is to find a system

\[ g_i(x) = h_i(x) \]  

of Boolean equations, of the simplest possible form, that is equivalent to (i.e., has the same solutions as) the single equation (1). Once such a system has been found, \( f(x) \) may be realized by the structure shown in Fig. 1. The unit labeled "\( =\)" is a \( k \)-wide digital comparator, directly implemented using standard medium-scale components. The unit labeled "function generator," which produces the \( g \)'s and \( h \)'s of (2), is a multiple-output circuit designed by any of the usual functional techniques.

The equational organization has two main advantages. First, as will be shown by example, equational circuits can be significantly less costly (in gate inputs or IC package count) than optimized two-level circuits. Second, much of the information processing in an equational circuit is performed by a digital comparator, a standard medium-scale integrated circuit. The resulting stereotyped circuit organization is desirable for assembly and fault testing.

The problem of constructing a simplified system of equations equivalent to a single equation of the form \( f(x) = 1 \) was posed by Jevons in 1871 [6], and was studied subsequently by several other 19th century logicians. Jevons called this the "inverse problem of logic," and incorporated it into a comprehensive system which he called "equational logic." We appropriate the latter term to denote both the equational approach to circuit synthesis and the corresponding species of hardware organization.

We seek in this paper to outline the nature and potential of equational logic and to develop some results which appear to be basic for further work. A subsequent paper will present specialized techniques for circuit synthesis.

The work of Jevons and others on the inverse problem is summarized in Section II. Sections III and IV present alternative approaches to the solution of the inverse problem. In Section III, it is shown that the inverse problem can be formulated as a covering problem, by use of the "table of consequences" of Poretsky. The size of the Poretsky tables makes the covering approach difficult to use as it is presently formulated; however, we believe...
that this approach illuminates the inverse problem in a valuable way and will serve as a guide in the development of more practical covering algorithms.

In Section IV it is shown that the equivalence of the equation \( f(x) = 1 \) to a system of equations is intimately tied to the existence of a representation for \( f(x) \) which we call "bilinear."

II. THE INVERSE PROBLEM OF LOGIC

Jevons' Inverse Problem

It is a classical result of the Boole–Schröder algebra of logic [1], [2], [4], [10], [17] that the system (2) of Boolean equations is equivalent to the single equation (1) if and only if \( f \) is related to the \( g \)'s and \( h \)'s as follows:

\[
f = \prod_{i=1}^{k} (\bar{g}_i h_i + g_i h_i).
\]

(3)

It is therefore a straightforward process to reduce a system of equations to a single equivalent equation.

The problem of equational synthesis, however, is the much more difficult task of generating a simplified system of the form (2) which is equivalent to a given switching equation (1). Precisely this problem was posed a century ago by Jevons [6], who called it the "inverse problem of logic." Jevons challenged the readers of his Principles of Science [7] to find a simple system corresponding to the equation \( f(A, B, C, D, E, F) = 1 \), where \( f \) is given by the formula

\[
f = ABDF + ABD\bar{E} + B\bar{C}DF + B\bar{C}D\bar{E} + \bar{B}\bar{C}DE + ABCD\bar{E}F.
\]

(4)

Later, in his Studies in Deductive Logic [8], Jevons listed his own solution,

\[
AC = BC
\]

\[
BDE = BDEF
\]

\[
B = C + D
\]

\[
D = DEF,
\]

(5)

but noted that one of his readers had submitted the following improved solution:

\[
A = A + C
\]

\[
B = C + D
\]

\[
D = \bar{E} + F.
\]

(6)

An equational realization of (6) is shown in Fig. 2. This realization requires 12 gate inputs (counting each input to the comparator as one gate input); an optimized two-level realization, on the other hand, requires 33 gate inputs (in counting gate inputs, we shall consistently assume "double-rail" inputs, i.e., that the complements of all switching variables are available).

Venn's Method

In 1894, Venn [18] presented a method for attacking the inverse problem; a form of this procedure was used implicitly by Jevons. We shall discuss a generalization of Venn's method, beginning by expressing (1) in the equivalent form

\[
\bar{f}(x) = 0.
\]

(7)

If \( \bar{f} \) is expressed as a logical sum of subfunctions, i.e.,

\[
\bar{f} = \gamma_1 + \gamma_2 + \cdots + \gamma_m,
\]

(8)

then (7), and therefore (1), is equivalent to the system

\[
\gamma_1 = 0
\]

\[
\gamma_2 = 0
\]

\[\vdots\]

\[
\gamma_m = 0.
\]

(9)

The system (9) may then be transformed, equation by equation, into a system of the form (2) in any convenient way that conforms to the rule

\[
\gamma_i = g_i \oplus h_i.
\]

(10)

The critical step in the foregoing process is to select \( \gamma \)-functions in (8) that correspond to simple \( g_i, h_i \)-pairs conforming to (10).

The decomposition of \( \bar{f} \) is assisted by use of the equivalence

\[
u(v \oplus w) = 0 \iff w = uvw
\]

(11)

which was given by Jevons. The \( u, v, \) and \( w \) in (11) may of course be complex logical functions. Setting \( u = 1 \) in the equivalence (11) produces a better known equivalence that is also useful:

\[
v \oplus w = 0 \iff v = w.
\]

(12)

Let us consider, for example, the function

\[
f(A, B, C, D, E) = \bar{A}\bar{B}C\bar{D}E + \bar{A}\bar{B}CDE + \bar{A}BC\bar{D}E + ABCD\bar{E}
\]

\[
+ ABCDE + ABC\bar{D}E
\]

\[
+ A\bar{B}\bar{C}DE + A\bar{B}\bar{C}D\bar{E}
\]

(13)

which is plotted on the Karnaugh map of Fig. 3. After contemplation of the distribution of 0's on the map, we develop \( \bar{f} \) into the following logical sum:
\[ \hat{f} = B\bar{C} + (A \oplus D \oplus E) + \bar{A}C(\bar{B} \oplus D) + A(B \oplus C) + \bar{C}(\bar{A} \oplus \bar{D}E). \]  
\hspace{1cm} \text{(14)}

The equation \( f = 1 \) is therefore equivalent to the following 5-equation system:

\[ \begin{align*}
B\bar{C} &= 0 \\
A \oplus D \oplus E &= 0 \\
\bar{A}C(\bar{B} \oplus D) &= 0 \\
A(B \oplus C) &= 0 \\
\bar{C}(\bar{A} \oplus \bar{D}E) &= 0. \\
\end{align*} \]
\hspace{1cm} \text{(15)}

Applying the equivalences (11) and (12), we transform (15) into the system

\[ \begin{align*}
B\bar{C} &= 0 \\
A &= D \oplus E \\
\bar{A}BC &= \bar{A}CD \\
AB &= AC \\
\bar{A}C &= \bar{C}D\bar{E}. \\
\end{align*} \]
\hspace{1cm} \text{(16)}

Another development of \( \hat{f} \),

\[ \hat{f} = BDE + (A \oplus D \oplus E) + (C \oplus (B + DE)), \]  
\hspace{1cm} \text{(17)}
leads to the system

\[ \begin{align*}
BDE &= 0 \\
A &= D \oplus E \\
C &= B + DE. \\
\end{align*} \]
\hspace{1cm} \text{(18)}

A two-level realization of \( (13) \) requires 42 gate inputs. The systems (16) and (18) require 29 and 15 inputs, respectively. The system (18) is particularly economical in IC package count; the realization shown in Fig. 4 requires only two 14-pin packages (one triple 3-input NAND and one quad open-collector EXCLUSIVE NOR).

There is a wide range of choice in attempting to decompose \( \hat{f} \) into the form (8) in the most advantageous way. Unfortunately, “clever” decompositions such as (17) are discovered only after considerable experiment. Jevons stated that “the inverse problem is always tentative” and Venn noted that “the requisite investigation appears rather to be of the kind with which tact and judgment, aided by graphical methods, will best grapple.”

**Poretsky's Law of Forms**

An important specialization of the inverse problem is that of finding a simplified single equation equivalent to (1). Among the equations equivalent to

\[ AB + \bar{A}CE + \bar{A}DE + B\bar{C}E + BDE \]
\[ + AC\bar{D}E + BCD\bar{E} = 1, \]  
\hspace{1cm} \text{(19)}
for example, is the relatively simple equation

\[ AB + C\bar{D} = B\bar{E} + AE. \]
\hspace{1cm} \text{(20)}

To our knowledge, no efficient procedure has been devised to find such simplified single-equation equivalents. On the other hand, the problem of finding all equations equivalent to a given equation was solved by Poretsky in 1898 [15]. Poretsky's result may be stated as follows.

**Poretsky's Law of Forms:** In a finite Boolean algebra \( B \), all equations equivalent to a given equation \( f = 1 \) are obtained by substituting successively for \( g \) in \( fg + \bar{f}\bar{g} \) every distinct member of \( B \).

Thus, for the Boolean algebra of switching functions of \( n \) variables, there are \( 2^n \) pairs \((g,h)\) for which \( g = h \) is equivalent to \( f = 1 \).

Let us say that \( f = 1 \) is *proper* provided that \( f \) is neither the 1-function nor the 0-function (i.e., \( f = 1 \) is neither identically true nor identically false). Given a proper equation, the \( 2^n \) equivalent equations produced by Poretsky’s Law of Forms may be arranged in symmetrical pairs. We do not consider equations to be distinct which differ only by exchange of left and right sides; hence,
there are \(2^{n-1}\) distinct equations that are equivalent, in the algebra of \(n\)-variable switching functions, to a given proper equation. For example, the equation \(A\overline{B} = 1\), for which \(n = 2\), has 8 distinct equational equivalents, namely,

\[
\begin{align*}
0 &= \overline{A} + B \\
1 &= A\overline{B} \\
A &= \overline{A} + \overline{B} \\
\overline{A} &= AB \\
B &= \overline{A}\overline{B} \\
\overline{B} &= A + B \\
\overline{A}\overline{B} + AB &= \overline{A}B \\
AB + A\overline{B} &= A + \overline{B}.
\end{align*}
\]

The foregoing list shows that the equations occur in complementary pairs. If complementary equations are not considered to be distinct, then the number of distinct equations equivalent to \(f = 1\) is \(2^{n-2}\).

Other 19th Century Work

Further studies on Jevons' inverse problem were carried out by J. N. Keynes, who devoted the last chapter of his *Formal Logic* [11] to an analysis of the problem and to the development of three approaches to its solution. Jevons' problem was also discussed by W. E. Johnson [9].

The logicians' interest in the inverse problem seems to have lapsed by the beginning of the 20th century and it is not mentioned in present-day texts on Boolean algebra or symbolic logic. We believe, however, that a new search for solutions to this problem is justified by its relevance to circuit design and by the wealth of concepts in logic and optimization which have become available in this century.

III. CONSEQUENCE TABLES

Designing an equational circuit realizing a function \(f\) consists in solving the inverse problem for \(f = 1\). It is clear that the methods for solving this problem that have been bequeathed to us by the 19th century logicians require more "tact and judgment" than is appropriate for circuit design.

The objective of the present section is to outline a technique, based on the "table of consequences" of Poretsky, for producing least-cost solutions for the inverse problem. This technique shows that Jevons was pessimistic in stating that "the inverse problem is always tentative"; however, the technique in its present form entails working with large tables to solve problems of practical interest.

A Covering Formulation

A number of synthesis problems in switching theory have been formulated as covering, or "worker-job," problems [5]. Consider, for example, the problem of finding a least-cost sum-of-products formula for a switching function \(f\); in this case the "workers" are the prime implicants of \(f\) and the "jobs" are the minterms of \(f\).

A covering technique for the solution of the inverse problem may be based on the observation that the system (2) is equivalent to the (1) if and only if the equation

\[ \bar{f} = (g_1 \oplus h_i) + (g_2 \oplus h_2) + \cdots + (g_k \oplus h_k) \]  

(21) is satisfied. This observation is also the basis for the method of Venn discussed in Section II. An equation \(g_i = h_i\) can therefore be part of a system equivalent to \(f = 1\) if and only if \(g_i \oplus h_i\) is included in \(\bar{f}\). Such an equation, which must be satisfied if \(f = 1\) is satisfied, was termed by Poretsky [15] a "consequence" of \(f = 1\). Accordingly, we shall call an equation \(g = h\) a consequence of \(f = 1\) provided the condition \(g \oplus h \leq \bar{f}\) is satisfied. A consequence \(g = h\) of \(f = 1\) will be said to cover a minterm \(m_i\) of \(f\) provided the condition \(m_i \leq g \oplus h\) is satisfied.

Based on the equational realization of Fig. 1, we assign a cost to a consequence \(g_i = h_i\) according to the prescription

\[ \text{cost} (g_i = h_i) = \text{cost} (g_i) + \text{cost} (h_i) + 2. \]  

(22)

The terms \(\text{cost} (g_i)\) and \(\text{cost} (h_i)\) are numbers of gate inputs of two-level realizations of \(g_i\) and \(h_i\); the final term in (22) accounts for the comparator inputs. We define the cost of a system of the form (2) to be the sum of the costs of its constituent equations.

A covering procedure for solving the inverse problem can be organized as follows: given a switching function \(f\).

**Step 1:** Find all possible consequences \(g_i = h_i\) of \(f = 1\).

**Step 2:** Assign a cost, as defined by (22), to each consequence found in Step 1.

**Step 3:** Construct a covering table having one row for each consequence \(g_i = h_i\) and one column for each min-
term $m_i$ of $\tilde{f}$. Place a cross in cell $i,j$ of the table in case $g_i = h_i$ covers $m_i$; otherwise, leave cell $i,j$ blank. This table has the same logical significance as Quine’s table of prime implicants [16].

**Step 4:** Using the standard techniques [5] for solving covering tables, find a least-cost subset \{ $g_i = h_i$, $g_s = h_s$, \ldots, $g_t = h_t$ \} of the consequences of $f = 1$ such that every minterm $m_i$ of $\tilde{f}$ is covered by at least one consequence in the subset. This collection of equations, when asserted simultaneously, constitutes an optimal solution for the inverse problem.

**Poretsky’s Table of Consequences**

Step 1 of the foregoing procedure is greatly facilitated by the “table of consequences” of Poretsky. Before discussing the table, we prove a theorem which is implicit in Poretsky’s approach.

**Theorem 1:** The equation $g = h$ is a consequence of the equation $f = 1$ if and only if $g$ and $h$ can be expressed as $g = p + r$ and $h = q + r$, where $p \leq \tilde{f}$, $q \leq \tilde{f}$, and $r \leq f$.

**Proof:** Suppose that $g = h$ is a consequence of $f = 1$. Then $g \oplus h \leq \tilde{f}$, i.e., there is some $w$ such that $g \oplus h = w\tilde{f}$. Invoking Boole’s expansion theorem, there exist $u$ and $v$ such that $g = u\tilde{f} + vf$. Hence, $(u\tilde{f} + vf) \oplus h = w\tilde{f}$, which is readily solved for $h$ as follows, $h = (u \oplus w)\tilde{f} + vf$. Thus $p = p\tilde{f}$, $q = q\tilde{f}$, and $r = vf$, whence $g = p\tilde{f} + rf$ and $h = q\tilde{f} + rf$. It is a matter of substitution and computation to verify that $g \oplus h = (p \oplus q)\tilde{f}$; thus, $g \oplus h \leq \tilde{f}$ and we conclude that $g = h$ is a consequence of $f = 1$. Q.E.D.

**Corollary:** Let $g = p + r$ and $h = q + r$, where $p \leq \tilde{f}$, $q \leq \tilde{f}$, and $r \leq f$. Then $g \oplus h = p \oplus q$. Suppose, on the other hand, that $g = p + r$ and $h = q + r$, where $p \leq \tilde{f}$, $q \leq \tilde{f}$, and $r \leq f$. Then $p = p\tilde{f}$, $q = q\tilde{f}$, and $r = rf$, whence $g = p\tilde{f} + rf$ and $h = q\tilde{f} + rf$. It is a matter of substitution and computation to verify that $g \oplus h = (p \oplus q)\tilde{f}$; thus, $g \oplus h \leq \tilde{f}$ and we conclude that $g = h$ is a consequence of $f = 1$. Q.E.D.

Let $f$ be an $n$-variable switching function expressible as the sum of $m$ minterms. Let $A = \{a_1, a_2, \ldots, a_s\}$ be the set of subfunctions of $f$ and let $B = \{\beta_1, \beta_2, \ldots, \beta_t\}$ be the set of subfunctions of $\tilde{f}$, where $s = 2^m - m$. That is, let $A = \{a \mid a \leq \tilde{f}\}$ and $B = \{\beta \mid \beta \leq \tilde{f}\}$. Define the table of consequences for $f = 1$ as a $2^n \times 2^n$ array disposed as follows. Let row $i$ correspond to the $f$-subfunction $a_i$; and let column $j$ correspond to the $\tilde{f}$-subfunction $\beta_j$. The entry $\phi_{ij}$ in row $i$ and column $j$ is defined by the rule

\[
\phi_{ij} = a_i + \beta_j. \tag{23}
\]

It is clear from Theorem 1 that the set of consequences of $f = 1$ is the set of equations generated by assigning all possible values to $i$, $j$, and $k$ in

\[
\phi_{ij} = \phi_{ik}. \tag{24}
\]

Thus, each consequence of $f = 1$ (and nothing else) is found by equating two entries in the same row of the table of consequences. Letting $j = k$ in (24) produces a tautology of the form $g = q$, which is of no interest in the present context. Accordingly, we shall say that a **proper consequence** of $f = 1$ is an equation of the form (24) for which $j \neq k$.

Consider, for example, the function $f = \tilde{A}B + A\tilde{B}$, for which $n = 2$, $m = 2$, and $s = 2$. The corresponding table of consequences is shown as Table I.

**Table of Costs**

At the time that the table of consequences is constructed, it is convenient to construct a corresponding table of costs. The entry $c_{ij}$ in row $i$ and column $j$ of the table of costs is defined by

\[
c_{ij} = \text{cost}(\phi_{ij}) + 1. \tag{25}\]

Note in (25) that the gate-input cost of the function $\phi_{ij}$ is augmented by 1 to account for the associated comparator input. The table of costs corresponding to Table I is shown as Table II.

Comparing (25) with (22), we see that the cost of any consequence is read from the table of costs as follows:

\[
\text{cost}(\phi_{ij} = \phi_{ik}) = c_{ij} + c_{ik}. \tag{26}\]

**Covering Table**

Each consequence $\phi_{ij} = \phi_{ik}$ covers the minterms in a subfunction $\phi_{ij} \oplus \phi_{ik}$ of $\tilde{f}$. Applying the Corollary to Theorem 1, the subfunction covered by such a consequence is $\beta_j \oplus \beta_k$. Thus, all consequences determined by a given pair of columns of the table of consequences cover the same minterms of $\tilde{f}$. It is only necessary, therefore, to select one least-cost consequence from each pair of columns to find candidates for a least-cost cover; the table of costs facilitates this selection. Continuing with the example begun earlier, let us extract a set of candidates from Table I with the help of Table II. The relevant information is tabulated below.

<table>
<thead>
<tr>
<th>Column</th>
<th>Pair</th>
<th>Candidate</th>
<th>Minterms Covered</th>
<th>Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0, $\tilde{A}B$</td>
<td>$\tilde{A}B$</td>
<td>$\tilde{A}B$</td>
<td>$\tilde{A}B$</td>
<td>4</td>
</tr>
<tr>
<td>0, $AB$</td>
<td>$AB$</td>
<td>$AB$</td>
<td>$AB$</td>
<td>4</td>
</tr>
<tr>
<td>0, $\tilde{A}B + AB$</td>
<td>$\tilde{A}B + AB$</td>
<td>$\tilde{A}B + AB$</td>
<td>$\tilde{A}B + AB$</td>
<td>6</td>
</tr>
<tr>
<td>$\tilde{A}B, AB$</td>
<td>$\tilde{A}B, AB$</td>
<td>$\tilde{A}B, AB$</td>
<td>$\tilde{A}B, AB$</td>
<td>2</td>
</tr>
<tr>
<td>$\tilde{A}B, \tilde{A}B + AB$</td>
<td>$\tilde{A}B + AB$</td>
<td>$\tilde{A}B + AB$</td>
<td>$\tilde{A}B + AB$</td>
<td>4</td>
</tr>
<tr>
<td>$AB, \tilde{A}B + AB$</td>
<td>$\tilde{A}B + AB$</td>
<td>$\tilde{A}B + AB$</td>
<td>$\tilde{A}B + AB$</td>
<td>4</td>
</tr>
</tbody>
</table>

The selection of a least-cost cover is carried out by use of a **covering table** as shown in Table III.

The table is solved as an ordinary covering problem with costs. The result in the present example is obvious, since the least-cost consequence, namely, $\tilde{A} = B$, covers all of the minterms of $\tilde{f}$. Thus, an optimal solution to the
inverse problem for $\bar{A}B + A\bar{B} = 1$ is the single equation $\bar{A} = B$.

IV. BILINEAR REPRESENTATIONS

It is a commonplace that the formulation of a problem, i.e., the way the problem is stated, strongly influences the process and nature of the solution of the problem. In Section III, the inverse problem for $f = 1$ was formulated as one of finding a least-cost formula for $f$ having the form

$$\sum_{i=1}^{k} (g_i \oplus h_i).$$

That formulation (which is the basis for Venn's method) converts the inverse problem to a particular covering problem. The practical difficulties in solving that problem impel us to search for an alternative formulation.

We show in this section that all solutions to the inverse problem derive from representations for $f(x)$ which we call "bilinear." This observation leads to a new formulation of the inverse problem. We do not offer a method of solution based on this formulation; however, we believe that the concept of a bilinear representation for $f(x)$ is basic for the development of practical solution techniques. Bilinear representations are conveniently discussed in terms of Boolean matrices, whose relevant properties we now summarize.

Orthogonal Boolean Matrices

A Boolean matrix is a matrix whose elements belong to a Boolean algebra, typically the algebra of Boolean functions of some argument-vector $x$. We summarize in this section some specialized properties of such matrices that are useful for our present purposes. The elementary operations and properties of Boolean matrices are used in this paper without formal development; for such development, see Luce [14] or Brown [3].

We will say that a vector $v = (v_1, \cdots, v_p)^T$ is orthogonal provided the condition

$$v_r \cdot v_s = 0 \quad \text{if} \quad r \neq s \quad (27)$$

is satisfied; we will say that $v$ is normal provided the condition

$$\sum_{i=1}^{p} v_i = 1 \quad (28)$$

is satisfied. If $v$ is both orthogonal and normal, we shall call it orthonormal. The central importance of orthonormal vectors was demonstrated by Löwenheim [12], [13], who called them "disjunctive systems."

We will say that an $m \times n$ Boolean matrix is orthonormal (orthogonal, normal) provided each of its columns is orthonormal (orthogonal, normal).

Code Matrices and Minterm Vectors

We define a code matrix as a Boolean matrix of 0's and 1's whose columns are distinct. Thus,

$$C = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 \end{bmatrix} \quad (29)$$

is a code matrix. An $m$-row code matrix cannot have more than $2^m$ columns. We note also that there are $(2^m)! / (2^m - n)!$ code matrices having $m$ rows and $n$ columns; thus, there are 6720 $3 \times 5$ code matrices.

Given a $p$-element Boolean vector $v$, we define a $2^p$-element minterm-vector $m(v)$ as follows:

$$m(v) = \begin{bmatrix} m_0(v) \\ m_1(v) \\ \vdots \\ m_{2^p-1}(v) \end{bmatrix} = \begin{bmatrix} \bar{v}_1 & \cdots & \bar{v}_{p-1} & \bar{v}_p \\ \bar{v}_1 & \cdots & \bar{v}_{p-1} & v_p \\ \vdots & \cdots & \vdots \\ \vdots & \cdots & \vdots \\ \bar{v}_1 & \cdots & \bar{v}_{p-1} & v_p \end{bmatrix} \quad (30)$$

\[\text{TABLE I}\]

| Table of Consequences for $\bar{A}B + A\bar{B} = 1$ |
|-----------------|-----------------|-----------------|-----------------|
| $0$             | $\bar{A}B$      | $AB$            | $\bar{A}B + AB$|
| $\bar{A}B$      | $\bar{A}B$      | $A$             | $\bar{A}B + AB$|
| $\bar{A}B + AB$ | $\bar{A}B + AB$ | $A + B$         | $A + B$         |

\[\text{TABLE II}\]

| Table of Costs for $\bar{A}B + A\bar{B} = 1$ |
|-----------------|-----------------|-----------------|-----------------|
| $0$             | $\bar{A}B$      | $AB$            | $\bar{A}B + AB$|
| $\bar{A}B$      | $3$             | $1$             | $1$             |
| $\bar{A}B + AB$ | $7$             | $3$             | $1$             |
The elements \( m_0(v), m_1(v), \ldots, m_{2^p-1}(v) \) are called the minterms of \( v \). It is clear that the minterm-vector \( m(v) \) is orthonormal, no matter what the character of \( v \).

The transformation from \( v \) to \( m(v) \) is prescribed by (30). The inverse transformation is specified by the matrix product

\[
v = K_p m(v)
\]

where \( K_p \) is the \( p \times 2^p \) standard code matrix, defined by the following recursion:

\[
K_1 = [0 \ 1]
\]

\[
K_{n+1} = \begin{bmatrix} \cdots & \cdot & \cdots & \cdot \\ K_n & K_n & \cdots & K_n \\ \end{bmatrix}
\]

\[
K_n = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ \end{bmatrix}
\]

Thus,

\[
K_2 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \\ \end{bmatrix}
\]

\[
K_3 = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ \end{bmatrix}
\]

etc.

**Theorem 2:** Let \( g \) and \( h \) be \( k \)-element Boolean vectors. Then the following are equivalent:

1) \( g = h \);
2) \( m(g) = m(h) \);
3) \( \prod_i \delta(h_i + g_h_i) = 1 \); and
4) \( m^T(g)m(h) = 1 \).

**Proof:** The equivalence between 1) and 2) follows from the unique correspondence of a vector \( v \) and its associated minterm-vector \( m(v) \). The transformation from \( v \) to \( m(v) \) is given by (30); the inverse transformation is given by (31). The equivalence between 1) and 3) is the basis for the classical theory of Boolean equations, as noted earlier. The final equivalence we shall consider is that between 2) and 4). By the Müller–Löwenheim Verification Theorem [12], it suffices to consider only 0,1 values for the elements of \( g \) and \( h \) to verify any equivalence or identity concerning these vectors. For all such assignments of value to the elements of \( g \) and \( h \), the vectors \( m(g) \) and \( m(h) \) each have exactly one 1-element, their remaining elements being 0's. Thus \( m(g) = m(h) \) if and only if there is some value of \( i \) such that \( m_i(g) \cdot m_i(h) = 1 \), whence \( m(g) = m(h) \) if and only if \( \sum_i m_i(g) m_i(h) = m^T(g)m(h) = 1 \). We have established the following equivalences:

\[
1) \iff 2) , \quad 1) \iff 3) , \quad \text{and} \quad 2) \iff 4) .
\]

Hence, by the transitivity of the relation of equivalence, each of 1), 2), 3), and 4) is equivalent to any other.

Q.E.D.

A useful byproduct of Theorem 2 is the identity

\[
\prod_i (\delta(h_i + g_h_i)) = m^T(g)m(h)
\]

which holds for arbitrary vectors \( g \) and \( h \) having the same number of elements.

We shall have occasion to employ matrix equations of the form \( v = Wm(x) \). An important property of such equations is given in the following theorem.

**Theorem 3:** Let a vector \( v \) be related to a vector \( x \) as follows:

\[
v = Wm(x).
\]

Then \( v \) is orthonormal if and only if \( W \) is orthonormal.

**Proof:** Let \( v \) be orthonormal. Then \( v_i v_j = 0 \) if \( r \neq s \), whence

\[
(\sum_j w_j m_j(x))(\sum_k w_k m_k(x)) = \sum_j w_j w_k m_j(x) = 0,
\]

where we have invoked the orthogonality of \( m(x) \) to reduce a double summation to a single summation. Thus, \( w_i v_i m_j(x) = 0 \) for all \( j \) if \( r \neq s \). For any value of the index \( j \), we may select \( x \) so that \( m_j(x) = 1 \); thus \( w_j v_j m_j(x) = 0 \) for all \( j \) if \( r \neq s \), from which we conclude that \( W \) is orthogonal. To show that \( W \) is normal, we invoke the assumption that \( v \) is normal, i.e., \( \sum_i v_i = 1 \). Hence,

\[
\sum_j w_j m_j(x) = 1.
\]

For any value of \( t \), we may select \( x \) so that

\[
m_j(x) = \begin{cases} 1 & \text{if } j = t \\ 0 & \text{if } j \neq t \end{cases}
\]
Thus \( \sum_i w_{ij} = 1 \) for all \( t \), whence \( W \) is normal. \( W \) has already been shown to be orthogonal; hence, \( W \) is orthonormal. To prove the converse, let us assume that \( W \) is orthonormal. It has already been shown that \( vR = \sum_j w_{rj}v_j \). If \( r \neq s \), then \( w_{rj}v_j = 0 \) for all \( j \); hence, \( vRv = 0 \) if \( r \neq s \), i.e., \( v \) is orthogonal. To show that \( v \) is normal, we note that

\[
\sum_i v_i = \sum_i \sum_j w_{ij} m_j(x) = \sum_j m_j(x) \sum_i w_{ij}.
\]

We have assumed, however, that \( W \) is orthonormal; hence, \( \sum_i w_{ij} = 1 \) for all \( j \). Thus, \( \sum_i v_i = \sum_j m_j(x) = 1 \), whence \( v \) is normal. We conclude that \( v \) is orthonormal. Q.E.D.

Bilinear Representations

A bilinear representation for a Boolean function \( f(x) \) is an expression of the form

\[
f = \sum_{i=1}^{k} \alpha_i \beta_i
\]

where \( \alpha = (\alpha_1, \cdots, \alpha_k)^T \) and \( \beta = (\beta_1, \cdots, \beta_k)^T \) are orthonormal vectors. The summation in (34) may be written as the matrix product \( \alpha^T \beta \). We will call \( \alpha \) and \( \beta \) the constituents of the bilinear representation.

Let us consider, for example, the function

\[
f(x_1, x_2, x_3, x_4) = x_1x_2 + x_3x_2x_4 + x_1x_2x_3x_4 + x_1x_2x_3x_4.
\]

This function has a bilinear representation whose constituents are exhibited below:

\[
\alpha = \begin{bmatrix}
x_1x_2 + x_1x_3 \\
x_1x_2 + x_3x_2 \\
0
\end{bmatrix},
\]

\[
\beta = \begin{bmatrix}
x_3x_4 \\
(x_1 + x_4)x_3 \\
(x_1 + x_3)x_4
\end{bmatrix}.
\]

A trivial bilinear representation for \( f \) has the constituents \( \alpha = (\hat{f} f)^T \) and \( \beta = (0, 1)^T \). We shall show in the following sections that the problem of constructing simple systems of equations equivalent to \( f = 1 \) is closely related to the problem of constructing simple bilinear representations for \( f \).

Orthonormal Systems and Their Contractions

Given two \( k \)-element orthonormal vectors \( \alpha(x) \) and \( \beta(x) \), we shall call the system of equations

\[
\alpha(x) = \beta(x)
\]

an orthonormal system. Given an \( m \times k \) code-matrix \( C \), we derive from the \( k \)-equation orthonormal system (37) an \( m \)-equation system

\[
C\alpha(x) = C\beta(x).
\]

The system (38) will be called a contraction of (37); we use this term because \( m \) may (and generally will) be chosen to be less than \( k \). Let us consider, for example, the following 5-equation orthonormal system:

\[
\begin{align*}
x_1x_2 &= x_1x_2 \\
x_1x_3 &= x_3x_2 \\
x_2x_4 &= x_2x_3 \\
x_1x_4 &= x_1x_3x_2 \\
0 &= x_1x_2x_3.
\end{align*}
\]

Choosing the code-matrix (29), we derive the 3-equation contraction

\[
\begin{align*}
x_1x_4 &= x_1x_3 \\
x_1 + x_3 &= x_1x_2 + x_3x_2 + x_2x_3 \\
x_1x_4 + x_2x_4 &= x_1x_2 + x_2x_3.
\end{align*}
\]

Theorem 4: If \( \alpha \) and \( \beta \) are \( k \)-element orthonormal vectors and \( C \) is an \( m \times k \) code-matrix, then the following are equivalent:

1) \( \alpha^T \beta = 1 \);
2) \( \alpha = \beta \); and
3) \( C\alpha = C\beta \).

Proof: We first prove the equivalence of 1) and 2). If \( \alpha^T \beta = 1 \), then (multiplying through by \( \alpha \) and invoking the orthogonality of \( \alpha \)) it follows that \( \alpha_i\beta_r = \alpha_r \) for arbitrary \( r \). Likewise, from the orthogonality of \( \beta \), we deduce that \( \alpha_i\beta_r = \beta_r \). Thus, \( \alpha = \beta \), for all \( r \), i.e., \( \alpha = \beta \). Conversely, if \( \alpha = \beta \), then \( \alpha^T \beta = \alpha^T \alpha = 1 \), from the normality of \( \alpha \). Hence 1) and 2) are equivalent. We now prove the equivalence of 2) and 3). If \( \alpha = \beta \), it follows immediately that \( C\alpha = C\beta \). If, on the other hand, \( C\alpha = C\beta \), then

\[
\sum_{i=1}^{k} \alpha_i c_i = \sum_{i=1}^{k} \beta_i c_i,
\]

where \( c_1, c_2, \cdots, c_k \) are the columns of \( C \). We deduce, after multiplying both sides by \( \alpha_i, \) and taking note of the orthogonality of \( \alpha \) and \( \beta \), that \( \alpha_i \beta_r c_r = \alpha_i \beta_r c_r \) for all \( r, s \). If \( r \neq s \), then \( c_i \) and \( c_j \) must differ in at least one row (this is, in fact, the defining condition for \( C \) to be a code matrix). Thus, \( \alpha_i \beta_r = 0 \) if \( r \neq s \). We have assumed that \( \alpha \) is normal; hence,
Multiplying through by \( \beta_s \) and recalling that \( \alpha_s \beta_s = 0 \) if \( r \neq s \), we deduce that \( \alpha_s \beta_s = \beta_s \). A similar argument, beginning with

\[
\sum_{r=1}^{k} \alpha_r = 1.
\]

and multiplying through by \( \alpha_s \), leads to the condition \( \alpha_s \beta_s = \alpha_s \). Thus, \( \alpha_s = \beta_s \) for all \( s \), i.e., \( \alpha = \beta \), establishing the equivalence of 2) and 3). We conclude, from the transitivity of the relation of equivalence, that each of 1), 2), and 3) is equivalent to any of the others. Q.E.D.

Theorem 4 states that an orthonormal system 2) is equivalent to any of its contractions 3); further, 2) is equivalent to an equation of the form \( f = 1 \), where \( f \) has a bilinear representation whose constituents are the orthonormal vectors defining 2). We are assured, for example, that (40) is equivalent to (39), and that both are equivalent to the equation

\[
x_1 x_2 x_3 + x_1 x_2 x_4 + x_1 x_3 x_4 + x_2 x_3 x_4 = 1.
\]

As a further example, let us consider again the function (35). Using the constituents specified by (36) and selecting the code matrix

\[
C = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix},
\]

we transform the 3-equation system \( \alpha = \beta \) into the following 2-equation system:

\[
x_1 x_2 + x_1 x_4 = x_3 + x_1 x_4,
1 = x_4 + x_1 x_4.
\]

A reviewer has observed that a simpler equivalent system can be formed from (43) by omitting the term \( x_1 x_4 \). The simpler system may be developed from the function (35) by use of the constituents

\[
\alpha = \begin{bmatrix} x_1 x_2 + x_1 x_4 \\ x_1 x_2 + x_3 x_4 \\ 0 \\ 0 \end{bmatrix}
\]

\[
\beta = \begin{bmatrix} x_2 x_3 \\ (x_1 + x_1) x_3 \\ x_3 x_4 \\ x_1 x_2 \end{bmatrix}
\]

together with the following code matrix:

\[
C = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}.
\]

Bilinear Representations and the Inverse Problem

The concept of a bilinear representation provides a means of reformulating the inverse problem. We show in the following theorem that the equivalence of the equation \( f(x) = 1 \) to a system of equations is intimately tied to the existence of a bilinear representation for \( f(x) \).

**Theorem 5:** The equation \( f = 1 \) is equivalent to the \( m \)-equation system \( g = h \) if and only if there exist orthonormal vectors \( \alpha, \beta \) and a code-matrix \( C \) such that \( f = \alpha^T \beta, \ g = C \alpha, \ \)and \( h = C \beta \).

**Proof:** Let \( f = 1 \) be equivalent to the system \( g = h \), where \( g = (g_1, \ldots, g_m)^T \) and \( h = (h_1, \ldots, h_m)^T \). Then, from Theorem 2, \( f = m^T(g)m(h) \). Thus, \( f = \alpha^T \beta, \ g = C \alpha, \) and \( h = C \beta \), where \( \alpha = m(g), \ \beta = m(h), \) and \( C = K_m \).

To prove the converse, suppose that there exist orthonormal vectors \( \alpha, \beta \) and a code-matrix \( C \) such that \( f = \alpha^T \beta, \ g = C \alpha, \) and \( h = C \beta \). Now, by Theorem 4, \( f = 1 \) is equivalent to \( C \alpha = C \beta \); hence, \( f = 1 \) is equivalent to \( g = h \). Q.E.D.

Theorem 5 shows that the vaguely formulated task of constructing a system of equations equivalent to the equation \( f(x) = 1 \) may be split into two more specific tasks, namely (1) find a bilinear representation \( f = \alpha^T \beta \) and (2) choose a code-matrix \( C \).

We have not found a general solution algorithm based on the bilinear formulation of the inverse problem. We believe, however, that this formulation provides a useful foundation for further work toward practical methods of solution. A subsequent paper will discuss matrix and map techniques, based on the bilinear formulation, which lead to solutions of specialized form.

VI. CONCLUSIONS

The term “equational logic” was used by Jevons to denote the representation of logical data by systems of Boolean equations. We have appropriated the word “equational” to refer to 1) combinational switching networks that implement systems of Boolean equations, and 2) methods of designing such networks. The central task of equational synthesis is to solve Jevons’ inverse problem. Optimal solutions can be found by means of a covering technique which employs Poretsky’s table of consequences; this technique entails the use of exorbitantly large arrays, however, for all but the simplest problems. The concept of a bilinear representation for a switching function leads to a new formulation of the inverse problem; this formulation, in our opinion, provides a promising foundation for further work toward practical techniques of equational synthesis.
ACKNOWLEDGMENT

The author wishes to thank one of the referees for his extremely thorough and helpful criticism.

REFERENCES


Frank Markham Brown was born in 1930 in San Diego, Calif. He received the B.S. degree from the U.S. Naval Academy, Annapolis, Md., in 1953, the M.S. degree from the Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, in 1958, and the Ph.D. degree from the Ohio State University, Columbus, in 1967.

As an Air Force Officer, he served in R and D positions and was assigned for ten years to the Faculty of the Air Force Institute of Technology. He is presently Associate Professor of Electrical Engineering at the University of Kentucky, Lexington.