Galois Switching Functions and Their Applications

B. Benjauthrit and I. S. Reed
University of Southern California

The Boolean difference expansion of Boolean algebra is generalized to finite (Galois) fields. A systematic method is provided for calculating the coefficients of this type of multivariable polynomial expansion. It is applied then to the synthesis of switching functions. Applications include multivalued logics as well as binary-valued logics.

I. Introduction

As the computer technology continues to flourish, logic design grows ever more complex. This, combined with the rapid advance of the integrated circuit technology, indicates the need for alternative logic designs which are simple and systematic in nature. Towards this end, this article generalizes to finite fields an expansion of Boolean algebra (Ref. 1). This expansion over finite fields includes the transform recently provided by Menger in Ref. 2 for single variable functions. This result was previously described in Ref. 3 for finite fields having 2^n elements where n is any positive integer.

Since the circuit design methods introduced in this article utilize polynomial expansions of finite or Galois fields, it is appropriate to call the physical realizations of this new expansion Galois switching functions. Such functions can be implemented in the same manner as logic networks using AND and OR gates. Nonetheless, instead of using only single gates, they often employ complex modules. To keep pace with the technology of large-scale integrated (LSI) circuits, following Menger (Ref. 2), such modules will be referred to as Galois PLUS and Galois TIMES gates.[1]

The present approach of mapping mathematical descriptions onto hardware differs from the Fourier-like transform technique discussed elsewhere (Ref. 4). The relationship of the mathematics to the circuit in the transform method is often lost and difficult to appreciate. Though the new method, presented here, still involves considerable mathematics, its complexity is compensated partly by the transparency of its mapping onto hardware.

[1]Rigorous treatments of these gates may be found in Refs. 3, 5, and 9.

Since the mathematical background necessary for this presentation is ubiquitous (Refs. 6 and 12), it will not be provided here. Several points of attraction for this type of logic design are presented. A few suggestions for future research in this area are also made.

II. The Polynomial Expansions

The polynomial expansion for single-variable Galois switching functions will be given first; then the result is extended to multivariable Galois switching functions. From this, a formal definition of Galois switching functions is also described.

Theorem 1 below, though given by Menger (Ref. 2), is reexpressed without proof here in a form needed to develop the multivariable expansion to be described in Theorem 2.

THEOREM 1: For any function $F{:}GF(p^n) \rightarrow GF(p^n)$, there exists a unique function f: $\{0, 1, \cdots, p^n - 1\} \rightarrow GF(p^n)$ such that

$$F(x) = \sum_{k=0}^{p^n-1} f(k)x^k \tag{1}$$

where the function f is given by

$$\begin{aligned} f(0) &= F(0) \\ f(k) &= \sum_{\gamma \in GF'(p^n)} [F(0) - F(\gamma)]\gamma^{-k}, 0 < k < p^n \end{aligned} \tag{2}$$

so that $F(x)$ has the following "power series" expansion:

$$F(x) = F(0) + \sum_{k=1}^{p^n-1} \left[\sum_{\gamma \in GF'(p^n)} [F(0) - F(\gamma)]\gamma^{-k} \right] x^k \tag{3}$$

The existence of Eq. (1) has long been known (Ref. 6). A more generalized version of it was first proven by Reed and Solomon (Ref. 7). Edwards (Ref. 8) later proved Eq. (1) in its present form when dealing with synthesis of switching networks. Menger provided Eq. (2) in his Theorem 2, a more generalized version of the Boolean difference described by Reed (Ref. 1). From Eq. (2), one can compute the coefficients $f(k)$ of Eq. (1).

With Theorem 1 and the recognition that Eq. (2) is a generalized Boolean difference, a method was found in Ref. 3 for deriving any multivariable polynomial expansion over Galois fields $GF(2^n)$. This method is now extended to the general Galois field $GF(p^n)$.

Let us first symbolize Eq. (2) differently by defining a generalized finite "difference" as follows:

$$f(k) = \underset{x^k}{\Delta} F(0) = \sum_{\gamma \in GF'(p^n)} [F(0) - F(\gamma)]\gamma^{-k} \tag{4}$$

where $0 < k < p^n$. Equation (4) is a sum of weighted differences $[F(0) - F(\gamma)]$ over the field $GF'(p^n)$. For $GF(2)$, it simply reduces to the usual Boolean difference (Ref. 1). This is the reason for calling the above sum a finite difference. Due to the linearity of Eq. (4) in the differences $[F(0) - F(\gamma)]$,

$$\begin{aligned} f(k_1, k_2) &= \underset{x_1^{k_1} x_2^{k_2}}{\Delta^{(2)}} F(0,0) = \underset{x_1^{k_1}}{\Delta} \left[\underset{x_2^{k_2}}{\Delta} F(0,0) \right] \\ &= \sum_{\gamma \in GF'(p^n)} \left[\underset{x_2^{k_2}}{\Delta} F(0,0) - \underset{x_2^{k_2}}{\Delta} F(\gamma_1, 0) \right] \gamma_1^{-k_1} \end{aligned}$$

But

$$\underset{x_2^{k_2}}{\Delta} F(0,0) = \sum_{\gamma_2 \in GF'(p^n)} [F(0,0) - F(0,\gamma_2)]\gamma_2^{-k_2}$$

and

$$\underset{x_2^{k_2}}{\Delta} F(\gamma_1,0) = \sum_{\gamma_2 \in GF'(p^n)} [F(\gamma_1,0) - F(\gamma_1,\gamma_2)]\gamma_2^{-k_2}$$

both also by Eq. (4). Hence, using the linear property of Σ,

$$\begin{aligned} f(k_1,k_2) &= \sum_{\gamma_1 \in GF'(p^n)} \left[\sum_{\gamma_2 \in GF'(p^n)} [F(0,0) - F(0,\gamma_2)]\gamma_2^{-k_2} - \sum_{\gamma_2 \in GF'(p^n)} [F(\gamma_1,0) - F(\gamma_1,\gamma_2)]\gamma_2^{-k_2} \right] \gamma_1^{-k_1} \\ &= \sum_{\gamma_1 \in GF'(p^n)} \sum_{\gamma_2 \in GF'(p^n)} [F(0,0) - F(0,\gamma_2) - F(\gamma_1,0) + F(\gamma_1,\gamma_2)]\gamma_1^{-k_1}\gamma_2^{-k_2} \end{aligned}$$

follows from the property $-u = -1 \cdot u, u \epsilon GF(p^n)$. For simplicity in notation, $f(k_1,k_2)$ will be rewritten as:

$$f(k_1,k_2) = \sum_{GF'(p^n)} \sum [F(0,0) - F(0,\gamma_2) - F(\gamma_1,0) + F(\gamma_1,\gamma_2)]\,\gamma_1^{-k_1}\gamma_2^{-k_2}$$

Proceeding in a similar fashion, the following higher order partial differences can be derived and are given below.

$$\begin{aligned} f(k_1,k_2,k_3) &= \underset{x_1^{k_1}x_2^{k_2}x_3^{k_3}}{\Delta^{(3)}} F(0,0,0) \\ &= \underset{x_1^{k_1}x_2^{k_2}}{\Delta^{(2)}} \left[\underset{x_3^{k_3}}{\Delta} F(0,0,0) \right] = \underset{x_1^{k_1}}{\Delta} \left[\underset{x_2^{k_2}}{\Delta} \left[\underset{x_3^{k_3}}{\Delta} F(0,0,0) \right] \right] \\ &= \sum_{GF'(p^n)} \sum \sum [F(0,0,0) - F(0,0,\gamma_3) - F(0,\gamma_2,0) - F(\gamma_1,0,0) \\ &\quad + F(0,\gamma_2,\gamma_3) + F(\gamma_1,0,\gamma_3) + F(\gamma_1,\gamma_2,0) - F(\gamma_1,\gamma_2,\gamma_3)]\gamma_1^{-k_1}\gamma_2^{-k_2}\gamma_3^{-k_3} \end{aligned}$$

$$\begin{aligned} f(k_{i_1},\cdots,k_{i_p}) &= \underset{x_{i_1}^{k_{i_1}}\cdots x_{i_p}^{k_{i_p}}}{\Delta^{(p)}} F(0,\cdots,0) = \underset{x_{i_1}^{k_{i_1}}\cdots x_{i_{p-1}}^{k_{i_{p-1}}}}{\Delta^{(p-1)}} \left[\underset{x_{i_p}^{k_{i_p}}}{\Delta} F(0,\cdots,0) \right] \\ &= \cdots = \underset{x_{i_1}^{k_{i_1}}}{\Delta} \left[\underset{x_{i_2}^{k_{i_2}}}{\Delta} \left[\cdots \left[\underset{x_{i_p}^{k_{i_p}}}{\Delta} F(0,\cdots,0) \right] \cdots \right] \right] \\ &= \cdots = \sum_{GF'(p^n)}^{i_p} \cdots \sum [F(0,\cdots,0) - F(0,\cdots,0,\gamma_{i_p}) \\ &\quad - \cdots - F(\gamma_{i_1},0,\cdots,0) + F(0,\cdots,\gamma_{i_{p-1}},\gamma_{i_p}) + \cdots + F(\gamma_{i_1},\gamma_{i_2},0,\cdots,0) \\ &\quad - \cdots {}^{*}F(\gamma_{i_1},\cdots,\gamma_{i_p})\,\gamma_{i_1}^{-k_{i_1}}\cdots\gamma_{i_p}^{-k_{i_p}} \end{aligned}$$

where the asterisk represents "−" if p is odd and "+" otherwise.

From the above expansions, an important theorem for multivariable Galois switching functions over the field $GF(p^n)$ can now be inferred. For notational conveniences, let $q = p^n - 1$. Also represent the vector of m 0's by $\underline{0}$.

THEOREM 2: For every function F: $GF(p^n)^m \rightarrow GF(p^n)$, there exists a unique function f: $\{0,1,\cdots,p^n-1\} \rightarrow GF(p^n)$ such that

$$F(x_1,\cdots,x_m) = \sum f(k_1,\cdots,k_m)x_1^{k_1}\cdots x_m^{k_m} \tag{5}$$

where the function f is given by

$$f(\underline{0}) = F(\underline{0})$$

$$f(k_1, 0, \cdots, 0) = \underset{x_1^{k_1}}{\Delta} F(\underline{0}) = \sum_{GF'(p^n)} [F(\underline{0}) - F(\gamma_1, 0, \cdots, 0)]\gamma_1^{-k_1}$$

$$f(k_1, k_2, \cdots, 0) = \underset{x_1^{k_1} x_2^{k_2}}{\Delta^{(2)}} F(\underline{0})$$

$$= \sum_{GF'(p^n)} \sum [F(\underline{0}) - F(0, \gamma_2, 0, \cdots, 0) - F(\gamma_1, 0, 0, \cdots, 0)$$

$$+ F(\gamma_1, \gamma_2, 0, \cdots, 0)]\gamma_1^{-k_1}\gamma_2^{-k_2}$$

$$\vdots$$

$$f(k_1, \cdots, k_m) = \underset{x_1^{k_1} \cdots x_m^{k_m}}{\Delta^{(m)}} F(\underline{0})$$

$$= \sum_{GF'(p^n)}^{m} \cdots \sum [F(\underline{0}) - F(0, \cdots, 0, \gamma_m) - \cdots - F(\gamma_1, 0, \cdots, 0)$$

$$+ F(0, \cdots, 0, \gamma_{m-1}, \gamma_m) + \cdots + F(\gamma_1, \gamma_2, 0, \cdots, 0) - \cdots {}^{*}F(\gamma_1, \cdots, \gamma_m)]\gamma_1^{-k_1} \cdots \gamma_m^{-k_m} \tag{6}$$

and

$$\underset{x_{i_1} \cdots x_{i_p}}{\Delta^{(p)}} F(\underline{0}) = \underset{x_{i_p}}{\Delta}\left[\underset{x_{i_1} \cdots x_{i_{p-1}}}{\Delta^{(p-1)}} F(\underline{0})\right] \text{ for } p = 1, 2, \cdots, m$$

where the asterisk represents "$-$" if m is odd and "$+$" otherwise. The function $F(x_1, \cdots, x_m)$ has the following "power series" expansion:

$$F(x_1, \cdots, x_m) = F(\underline{0}) + \left[\underset{x_1}{\Delta} F(\underline{0})\right]x_1 + \cdots + \left[\underset{x_m}{\Delta} F(\underline{0})\right]x_m$$

$$+ \left[\underset{x_1^2}{\Delta} F(\underline{0})\right]x_1^2 + \cdots + \left[\underset{x_m^2}{\Delta} F(\underline{0})\right]x_m^2 + \cdots + \left[\underset{x_m^q}{\Delta} F(\underline{0})\right]x_m^q$$

$$+ \left[\underset{x_1 x_2}{\Delta^{(2)}} F(\underline{0})\right]x_1 x_2 + \cdots + \left[\underset{x_{m-1}^q x_m^q}{\Delta^{(2)}} F(\underline{0})\right]x_{m-1}^q x_m^q$$

$$+ \left[\underset{x_1 \cdots x_m}{\Delta^{(m)}} F(\underline{0})\right]x_1 \cdots x_m + \cdots + \left[\underset{x_1^q \cdots x_m^q}{\Delta^{(m)}} F(\underline{0})\right]x_1^q \cdots x_m^q \tag{7}$$

Proof: We shall prove this theorem by an induction on m, the number of variables. The case $m = 1$ has already been established in Theorem 1. Assume now that Eq. (7) is true for all integers $k \leq m$ and consider the expansion of $F(x_1, \cdots, x_m, x_{m+1})$ with respect to its first m variables. By the induction hypothesis,

$$F(x_1, \cdots, x_m, x_{m+1}) = F(\underline{0}, x_{m+1})$$

$$+ \left[\underset{x_1}{\Delta} F(\underline{0}, x_{m+1})\right]x_1 + \cdots + \left[\underset{x_m^q}{\Delta} F(\underline{0}, x_{m+1})\right]x_m^q$$

$$+ \cdots + \left[\underset{x_1 x_2}{\Delta^{(2)}} F(\underline{0}, x_{m+1})\right]x_1 x_2 + \cdots + \left[\underset{x_{m-1}^q x_m^q}{\Delta^{(2)}} F(\underline{0}, x_{m+1})\right]x_{m-1}^q x_m^q$$

$$+ \cdots + \left[\underset{x_1 \cdots x_m}{\Delta^{(m)}} F(\underline{0}, x_{m+1})\right]x_1 \cdots x_m + \cdots + \left[\underset{x_1^q \cdots x_m^q}{\Delta^{(m)}} F(\underline{0}, x_{m+1})\right]x_1^q \cdots x_m^q \tag{8}$$

But $\underset{x_{i_1} \cdots x_{i_p}}{\Delta^{(p)}} \cdot F(\underline{0}, x_{m+1})$ is a mapping of $GF(p^n) \rightarrow GF(p^n)$ for $p = 0, 1, \cdots, m$. Hence, Eq. (3) can be used to compute

$$\underset{x_{i_1}^{k_{i_1}} \cdots x_{i_p}^{k_{i_p}}}{\Delta^{(p)}} F(\underline{0}, x_{m+1}) = \underset{x_{i_1}^{k_{i_1}} \cdots x_{i_p}^{k_{i_p}}}{\Delta^{(p)}} F(\underline{0}, 0) + \sum_{k_{m+1}=1}^{q} \left[\underset{x_{m+1}^{k_{m+1}}}{\Delta} \left[\underset{x_{i_1}^{k_{i_1}} \cdots x_{i_p}^{k_{i_p}}}{\Delta^{(p)}} F(\underline{0}, 0) \right] \right] x_{m+1}^{k_{m+1}}$$

$$= \underset{x_{i_1}^{k_{i_1}} \cdots x_{i_p}^{k_{i_p}}}{\Delta^{(p)}} F(\underline{0}, 0) + \sum_{k_{m+1}=1}^{q} \left[\underset{x_{i_1}^{k_{i_1}} \cdots x_{i_p}^{k_{i_p}} x_{m+1}^{k_{m+1}}}{\Delta^{(p+1)}} F(\underline{0}, 0) \right] x_{m+1}^{k_{m+1}} \tag{9}$$

For example,

$$\Delta^{(0)} F(\underline{0}, x_{m+1}) = \Delta^{(0)} F(\underline{0}, 0) + \sum_{k_{m+1}=1}^{q} \left[\underset{x_{m+1}^{k_{m+1}}}{\Delta^{(1)}} F(\underline{0}, 0) \right] x_{m+1}^{k_{m+1}}$$

or

$$F(\underline{0}, x_{m+1}) = F(\underline{0}, 0) + \sum_{k_{m+1}=1}^{q} \left[\underset{x_{m+1}^{k_{m+1}}}{\Delta} F(\underline{0}, 0) \right] x_{m+1}^{k_{m+1}}$$

Similarly,

$$\underset{x_1^{k_1}}{\Delta} F(\underline{0}, x_{m+1}) = \underset{x_1^{k_1}}{\Delta} F(\underline{0}, 0) + \sum_{k_{m+1}=1}^{q} \left[x_1^{k_1} \underset{x_{m+1}^{k_{m+1}}}{\Delta^{(2)}} F(\underline{0}, 0) \right] x_{m+1}^{k_{m+1}}$$

and so on. Substituting Eq. (9) into Eq. (8) yields

$$F(x_1, \cdots, x_m, x_{m+1}) = F(\underline{0}, 0) + \sum_{k_{m+1}=1}^{q} \left[\underset{x_{m+1}^{k_{m+1}}}{\Delta} F(\underline{0}, 0) \right] x_{m+1}^{k_{m+1}}$$

$$+ \left[\underset{x_1}{\Delta} F(\underline{0}, 0) + \sum_{k_{m+1}=1}^{q} \left[x_1 \underset{x_{m+1}^{k_{m+1}}}{\Delta^{(2)}} F(\underline{0}, 0) \right] x_{m+1}^{k_{m+1}} \right] x_1$$

$$+ \cdots + \left[\underset{x_m^q}{\Delta} F(\underline{0}, 0) + \sum_{k_{m+1}=1}^{q} \left[x_m^q \underset{x_{m+1}^{k_{m+1}}}{\Delta^{(2)}} F(\underline{0}, 0) \right] x_{m+1}^{k_{m+1}} \right] x_m^q$$

$$+ \cdots + \left[\underset{x_{m-1}^q x_m^q}{\Delta^{(2)}} F(\underline{0}, 0) + \sum_{k_{m+1}=1}^{q} \left[x_{m-1}^q x_m^q \underset{x_{m+1}^{k_{m+1}}}{\Delta^{(3)}} F(\underline{0}, 0) \right] x_{m+1}^{k_{m+1}} \right] x_{m-1}^q x_m^q$$

$$+ \cdots + \left[\underset{x_1^q \cdots x_m^q}{\Delta^{(m)}} F(\underline{0}, 0) + \sum_{k_{m+1}=1}^{q} \left[x_1^q \cdots x_m^q \underset{x_{m+1}^{k_{m+1}}}{\Delta^{(m+1)}} F(\underline{0}, 0) \right] x_{m+1}^{k_{m+1}} \right] x_1^q \cdots x_m^q$$

and the induction is complete. The uniqueness of f follows from the fact that the total number of possible functions F is $(p^n)^{(p^n)^m}$, which is exactly the total number of possible functions f.

The existence of Eq. (5) has long been known (Refs. 2, 6, 9, 10) but not the expansion given in Eq. (6). The canonical expansion of Theorem 2 generalizes to Galois field $GF(p^n)$ a result previously formed over $GF(2^n)$ (Ref. 3).

Theorem 2, which includes Theorem 1, provides a general means of synthesizing Galois switching functions for multiple-valued logics. For example, logic elements having three-level input and three-level output voltages (Ref. 11) can be used to implement Galois switching functions in the field $GF(3^n)$. Though most studies of logic design in the past have been for the binary-valued logic, the advent of Theorems 1 and 2 may change this.

For continuity of presentation, illustrative examples, using Theorem 2, will be postponed to the next section. From Theorem 2 one can now give a formal definition of Galois switching functions.

Definition 1: A Galois switching function F of m variables $v_1, \cdots, v_m$ over field $GF(p^n)$ is a rule that associates every m-tuple of valued input variables $(v_1, \cdots v_m)$ with an m'-tuple of valued output variables $(z_1, \cdots, z_{m'})$ over field $GF(p^n)$:

$$F\colon GF(p^n)^m \rightarrow GF(p^n)^{m'}$$

Mathematically, a Galois switching function describes the logic performance of a "combinational" switching circuit with m input terminals $v_1 \cdots, v_m$ and m' output terminals $z_1, \cdots, z_{m'}$.

If a parametrization of a function is the function obtained by replacing a subset of its variables with constants, then we can further state a more general definition of Galois functions.

Definition 2: Let $\underline{v} = (v_0, v_1, \cdots, v_{m-1})$ be a sequence of length m and $\underline{f} = (f_0, f_1, \cdots, f_{(p^n)^{m-1}})$ be a sequence of length $(p^n)^m$. A universal Galois function for $\underline{v}$ is a Galois function $U(\underline{v},\underline{f})$ such that each of the $(p^n)^{(p^n)^m}$ functions of $\underline{v}$ appears exactly once among the various parametrizations of U obtained by replacing f by a sequence of Galois field constants. (The active variables of $U(\underline{v},\underline{f})$ are $\underline{v}$, while its selection variables are f.) In other words, there exists a sequence $\underline{\alpha} = (\alpha_0, \alpha_1, \cdots, \alpha_{(p^n)^m-1})$ of field elements for each function F: $[GF(p^n)]^{|\underline{v}|} \rightarrow GF(p^n)$, such that

$$F(\underline{v}) = U(\underline{v}, \underline{\alpha})$$

where $|\underline{v}|$ is the length of v.

On the other hand, a general polynomial is defined as follows:

Definition 3: A general polynomial in m variables of length $\underline{q} = (q_0, \cdots, q_{m-1})$ is the sum of all terms

$$a_{i_0 \cdots i_{m-1}} v_0^{i_0} \cdots v_{m-1}^{i_{m-1}}$$

where $0 \leq i_0 \leq q_0, \cdots, 0 \leq i_{m-1} \leq q_{m-1}$:

$$\sum_{i_{m-1}=0}^{q_{m-1}-1} \cdots \sum_{i_0=0}^{q_0-1} a_{i_0 \cdots i_{m-1}} v_0^{i_0} \cdots v_{m-1}^{i_{m-1}}$$

For example, a general polynomial in two variables of length $2^2 = (2^2, 2^2)$ is of the form:

$$\begin{aligned}\sum_{i_1=0}^{3} \sum_{i_0=0}^{3} a_{i_0 i_1} v_0^{i_0} v_1^{i_1} = {} & a_{00} + a_{01} v_1 + a_{02} v_1^2 + a_{03} v_1^3 \\ & + a_{10} v_0 + a_{11} v_0 v_1 + a_{12} v_0 v_1^2 + a_{13} v_0 v_1^3 \\ & + \cdots + a_{30} v_0^3 + a_{31} v_0^3 v_1 + a_{32} v_0^3 v_1^2 + a_{33} v_0^3 v_1^3\end{aligned}$$

Definition 4: A universal polynomial over field $GF(p^n)$ in m variables, each of which has length p^n, is a general polynomial of length $(p^n, \overset{(m \text{ times})}{\cdots}, p^n)$.

From the above definitions and Theorem 2, Corollary 1 is immediate.

Corollary 1: If the polynomial expansion is implemented in hardware such that the p^m coefficients are left unspecified, and in fact treated as the other m inputs, then such an implementation is "universal" in the sense that it can simulate any given m-argument function. It is necessary only to supply the correct values to the coefficient inputs and these values are obtainable from Eq. (6) of Theorem 2.

III. Applications

We now give an example to show how Theorem 2 can be used to find Galois switching functions for multiple-valued logic. The computational procedures will be shown explicitly. Then, Galois switching functions for realizing conventional binary-valued logic are provided to illustrate the possible varieties of the logic design using Galois field theory.

Example 1: Consider the four-input two-output specification given in Table 2. Integers 0, 1, and 2 represent three signal levels of certain three-valued logic elements. We would like to derive the Galois switching function over field $GF(3^2)$ for this truth table.

The addition and multiplication tables for $GF(3^2)$ are described in Table 1 (a derivation for obtaining these tables is discussed in Ref. 5). The truth values given in Table 2 can be realized as a two-variable input, one-variable output function by the following partition:

$$x_1 = \{u_1, u_2\}; x_2 = \{u_3, u_4\}; F = \{v_1, v_2\}$$

Representing the field elements as

$$0 = 00, 1 = \alpha^0 = 10, \alpha^1 = 01, \alpha^2 = 12, \alpha^3 = 22, \alpha^4 = 20, \alpha^5 = 20, \alpha^6 = 21, \alpha^7 = 11$$

a truth table in terms of 0, 1, and α^i can be derived from Table 2.

The Galois switching function over field $GF(3^2)$ representing the truth values of Table 2 is derivable from Theorem 2. Utilizing the field operations of Table 1, some of the computations are given as follows:

$$f(0,0) = F(0,0)$$

$$f(1,0) = \sum_{GF'(3^2)} [F(0,0) - F(\gamma_1, 0)]\gamma_1^{-1}$$

$$= -\alpha \cdot 1 - \alpha^2 \cdot \alpha^7 - \alpha^3 \cdot \alpha^6 - \alpha^4 \cdot \alpha^5 - \alpha^5 \cdot \alpha^4 - \alpha^6 \cdot \alpha^3 - \alpha^7 \cdot \alpha^2 - 1 \cdot \alpha = \alpha$$

$$f(0,3) = \sum_{GF'(3^2)} [F(0,0) - F(0,\gamma_2)]\gamma_2^{-3}$$

$$= -1 - 1 - 1 - 1 - 1 - 1 - 1 - 1 = 1$$

$$f(1,1) = \sum\sum_{GF'(3^2)} [F(0,0) - F(0,\gamma_2) - F(\gamma_1, 0) + F(\gamma_1, \gamma_2)]\gamma_1^{-1}\gamma_2^{-1} = 1$$

Computations of this sort reveal that $f(5,7) = f(8,8) = 1$, and the remaining 75 functions f are all zero. Hence, the desired Galois switching function is of the form:

$$F(x_1, x_2) = \alpha x_1 + x_2^3 + x_1 x_2 + x_1^5 x_2^7 + x_1^8 x_2^8 \tag{10}$$

The fact that $F(x_1, x_2)$ realizes the truth values of Table 2 can be verified by direct substitution.

The above shows how three-voltage-level logic elements might be used to implement the truth values given in Table 2. If one had Galois PLUS and TIMES gates that were capable of performing the field operations given in Table 1, Eq. (10) could be implemented with only a few gates of these types.

Since most of present day logic elements are binary-valued type, we now provide more carefully an application of Theorem 2 in this regard.

Example 2: Consider the output table of a six-input two-output binary variable function given in Table 3 (Ref. 3). For economy of space, both the binary inputs and outputs of Table 3 are described by their decimal integer representations. The input of an output in row $n = i$ can be determined by substituting i for n in the expression given in the column of that output. For example, the input of the output 3 in row $n = 5$ and column $n + 16$ is 21. Thus, the binary output of the input 21, $u_1u_2u_3u_4u_5u_6 = 010101$, is $F_1F_2 = 11$. Similarly, the binary output of the input 10, $u_1u_2u_3u_4u_5u_6 = 001010$, is $F_1F_2 = 10$. These two outputs are encircled as shown in Table 3.

Employing either the Karnaugh map method or the Quine-McCluskey method, the minimal logical equations of the above output table are found to be as follows:

$$\begin{aligned} F_1(u_1, u_2, u_3, u_4, u_5, u_6) = {} & u_3u_4\bar{u}_5 \dot{+} \bar{u}_3u_5u_6 \dot{+} \bar{u}_1\bar{u}_2\bar{u}_3\bar{u}_4 \dot{+} u_1\bar{u}_2\bar{u}_3u_4 \\ & \dot{+} \bar{u}_1\bar{u}_2\bar{u}_5\bar{u}_6 \dot{+} u_1\bar{u}_2\bar{u}_5u_6 \dot{+} u_1u_2\bar{u}_3u_5 \dot{+} u_1u_2u_3\bar{u}_5 \\ & \dot{+} \bar{u}_1u_2\bar{u}_3u_4u_6 \dot{+} \bar{u}_1\bar{u}_3\bar{u}_4\bar{u}_5\bar{u}_6 \dot{+} \bar{u}_2u_3\bar{u}_4u_5\bar{u}_6 \end{aligned}$$

$$\begin{aligned} F_2(u_1, u_2, u_3, u_4, u_5, u_6) = {} & u_3u_4\bar{u}_6 + \bar{u}_4u_5u_6 \dot{+} \bar{u}_1\bar{u}_2\bar{u}_3\bar{u}_4 \\ & \dot{+} \bar{u}_1u_2u_3\bar{u}_4 \dot{+} u_1u_2\bar{u}_4u_6 \dot{+} u_1u_2u_4\bar{u}_6 \\ & \dot{+} \bar{u}_1\bar{u}_2\bar{u}_5\bar{u}_6 \dot{+} \bar{u}_1u_2u_5\bar{u}_6 \dot{+} u_1\bar{u}_2u_3\bar{u}_4u_5 \\ & \dot{+} \bar{u}_1\bar{u}_3u_4\bar{u}_5u_6 \dot{+} \bar{u}_2\bar{u}_3\bar{u}_4\bar{u}_5\bar{u}_6 \end{aligned}$$

where each $+$ denotes an OR operation.

These two logical equations can be implemented with 22 AND gates and 2 OR gates.

Now, either by Theorem 2 or by direct expansion, the Reed-Muller expansions of F_1 and F_2 are of the forms:

$$\begin{aligned} F_1(u_1, u_2, u_3, u_4, u_5, u_6) = {} & 1 \oplus u_1 \oplus u_1u_4 \oplus u_1u_6 \oplus u_2u_3 \oplus u_2u_4 \oplus u_2u_5 \oplus u_2u_6 \\ & \oplus u_3u_6 \oplus u_4u_5 \oplus u_4u_6 \oplus u_1u_3u_5 \oplus u_1u_3u_6 \oplus u_1u_4u_5 \\ & \oplus u_2u_3u_5 \oplus u_2u_4u_6 \end{aligned}$$

$$\begin{aligned} F_2(u_1, u_2, u_3, u_4, u_5, u_6) = {} & 1 \oplus u_2 \oplus u_1u_3 \oplus u_1u_4 \oplus u_1u_5 \oplus u_1u_6 \oplus u_2u_3 \oplus u_2u_5 \\ & \oplus u_3u_5 \oplus u_3u_6 \oplus u_4u_5 \oplus u_1u_3u_5 \oplus u_1u_4u_6 \oplus u_2u_3u_6 \\ & \oplus u_2u_4u_5 \oplus u_2u_4u_6 \end{aligned}$$

where each $\oplus$ denotes an EXCLUSIVE-OR operation. Note that in $GF(2)$, $+$ is commonly written as $\oplus$. Implementation of these expressions requires 20 AND gates and 2 sixteen-input EXCLUSIVE-OR gates.

Table 3 can be realized with far fewer basic gates over the field $GF(2^2)$ in the following manner. Let the input and output variables be partitioned as:

$$x_1 = \{u_1, u_2\}; x_2 = \{u_3, u_4\}; x_3 = \{u_5, u_6\}$$

and

$$F = \{F_1, F_2\}$$

Using the representation $GF(2^2) = \{0, 1, \alpha, \alpha^2\} = \{00, 11, 01, 10\}$ and Theorem 2, the Galois logic equation for the function F is derived to be:

$$F(x_1, x_2, x_3) = 1 + x_1 + x_1x_2 + x_1x_3 + x_2x_3 + x_1x_2x_3$$

Note that when $p = 2$, we have the identity: $- \equiv +$.

Two realizations of F are given in Fig. 1. If these Galois gates are comparable in cost with Boolean gates, F can be realized economically.

Examples showing more drastic savings of hardware may be found in Ref. 3.

It was shown in Ref. 3 that realizations of these types are also useful from the fault diagnostic point of view. One weak point about such a method is that it may create large delay time; this may be an important factor in certain systems. In spite of this fact, Theorem 2 can be considered as a systematic method for deriving multilevel logic networks, an alternative approach to the conventional methods—the Karnaugh map method and the Quine-McCluskey method.

IV. Points of Attraction

The following are a few points of attraction for using Galois field theory in logic design (Ref. 3).

(1) Galois theory offers a wider choice of basic logic gates, and conceptually it can be adapted to suit each stage of an architectural development (Ref. 10).

(2) In constructing an electronic encoder for the Bose-Chaudhuri-Hocquenghem code, Bartee and Schneider (Ref. 5) found that the decoding procedure could be economically implemented by designing the arithmetic unit based on Galois field operations.

(3) Similar requirements for Galois field arithmetic circuits have arisen in applications of linear recurring sequences to space object tracking.

(4) Networks derived from the concept of Galois field theory are usually composed of multioutput gates that are suitable for integrated circuit fabrication (Ref. 2).

(5) The realization of switching functions by modular algebra has been shown to be less complex than by Post algebra (Ref. 10). When m is prime, the modulo m number system forms a finite field. For this case, its application to multiple-valued logic is promising. Example 1 illustrates this fact.

(6) The circuit derived from Galois field theory possesses many properties that are suitable for fault diagnosis.

(7) Conceivably, Galois logic may play an important role in telephone networks or networks of a similar nature. Specifically, each multiwire cable or bundle may be considered as a Galois variable that can assume many values.

V. Conclusion

A previously given polynomial expansion technique for single-variable functions is recognized to contain a generalized Boolean difference from which a polynomial expansion for multivariable functions is derived. The expansion is then applied to the synthesis of switching functions using multioutput blocks called PLUS and TIMES gates. This application indicates that by employing an appropriate size of the domain and a suitable representation for the elements in that domain, the synthesis of switching functions can be accomplished economically.

The amount of tedious computation required in the above application also suggests the need for good computer programs. Such programs will enable one to select an appropriate size and assignment suitable for each individual synthesis. For a given truth table having m-variable inputs and n-variable outputs, $m \geq n$, one tends to select the size of the field according to m. This would create "don't care" conditions. Whether such a selection is beneficial or not is also a subject of further research.

References

1. Reed, I. S., "A Class of Multiple-Error-Correcting Codes and the Coding Scheme," *IRE Trans. on Inf. Theory*, IT-4, Sept. 1954, pp. 38–49.
2. Menger, K. S., Jr., "A Transform for Logic Networks," *IEEE Trans. on Computers*, C-18, March 1969, pp. 241–251.
3. Benjauthrit, B., *Design and Diagnosis of Galois Logic Networks*, Ph.D. Dissertation, University of Southern California, Los Angeles, California, June 1974.
4. Lechner, R. J., "Harmonic Analysis of Switching Functions," in *Recent Developments in Switching Theory*, A. Mukhopadhyay (ed.), Academic Press, Inc., New York, 1971, pp. 121–225.
5. Bartee, T. C., and Schneider, D. I., "Computation with Finite Fields," *Inf. and Control*, Vol. 6, June 1963, pp. 79–98.
6. Van der Waerden, B. L., *Algebra*, Vol. 1, Federick Unger Publishing, 1966.
7. Reed, I. S., and Solomon, G., *A Decoding Procedure for Polynomial Codes*, Group Report 47.24, Lincoln Lab., MIT, March 1959, pp. 1–6.
8. Edwards, R. W., *Algebraic Synthesis of Switching Networks*, Stanford Electronics Lab. Tech. Report No. 2205-1, April 1963.
9. Ellison, J. T., and Kolman, B., *Galois Logic Design*, Univac Division of Sperry Rand Corp., Final Report No. AD717205, Oct. 1970.
10. Pradhan, D. K., and Patel, A. M., "Muller Like Canonic Form for Multiple Multi-Valued Functions," *Corres. IEEE Trans. on Comp.*, Feb. 1975, pp. 206–210.
11. Epstein, G., et al., "Multiple-Valued Logic Design and Applications," 1971 Symposium on Multi-Valued Logic Design, Buffalo, N. Y.
12. Berlekamp, E. R., *Algebraic Coding Theory*, McGraw-Hill Book Co., 1968.

Table 1. Field operations

Addition defined for field $GF(3^2)$										Multiplication defined for field $GF(3^2)$									
+	0	1	α	α^2	α^3	α^4	α^5	α^6	α^7	•	0	1	α	α^2	α^3	α^4	α^5	α^6	α^7
0	0	1	α	α^2	α^3	α^4	α^5	α^6	α^7	0	0	0	0	0	0	0	0	0	0
1	1	α^4	α^7	α^3	α^5	0	α^2	α	α^6	1	0	1	α	α^2	α^3	α^4	α^5	α^6	α^7
α	α	α^7	α^5	1	α^4	α^6	0	α^3	α^2	α	0	α	α^2	α^3	α^4	α^5	α^6	α^7	1
α^2	α^2	α^3	1	α^6	α	α^5	α^7	0	α^4	α^2	0	α^2	α^3	α^4	α^5	α^6	α^7	1	α
α^3	α^3	α^5	α^4	α	α^7	α^2	α^6	1	0	α^3	0	α^3	α^4	α^5	α^6	α^7	1	α	α^2
α^4	α^4	0	α^6	α^5	α^2	1	α^3	α^7	α	α^4	0	α^4	α^5	α^6	α^7	1	α	α^2	α^3
α^5	α^5	α^2	0	α^7	α^6	α^3	α	α^4	1	α^5	0	α^5	α^6	α^7	1	α	α^2	α^3	α^4
α^6	α^6	α	α^3	0	1	α^7	α^4	α^2	α^5	α^6	0	α^6	α^7	1	α	α^2	α^3	α^4	α^5
α^7	α^7	α^6	α^2	α^4	0	α	1	α^5	α^3	α^7	0	α^7	1	α	α^2	α^3	α^4	α^5	α^6

Table 2. An input-output truth table

u_1	u_2	u_3	u_4	v_1	v_2	u_1	u_2	u_3	u_4	v_1	v_2	u_1	u_2	u_3	u_4	v_1	v_2
0	0	0	0	0	0	1	0	0	0	0	1	2	0	0	0	0	2
0	0	0	1	2	2	1	0	0	1	1	2	2	0	0	1	2	2
0	0	0	2	1	1	1	0	0	2	1	0	2	0	0	2	0	2
0	0	1	0	1	0	1	0	1	0	1	1	2	0	1	0	0	2
0	0	1	1	0	2	1	0	1	1	2	2	2	0	1	1	0	2
0	0	1	2	2	1	1	0	1	2	0	2	2	0	1	2	0	0
0	0	2	0	2	0	1	0	2	0	1	1	2	0	2	0	2	2
0	0	2	1	1	2	1	0	2	1	2	0	2	0	2	1	2	1
0	0	2	2	0	1	1	0	2	2	0	0	2	0	2	2	2	2
0	1	0	0	1	2	1	1	0	0	1	0	2	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	0	1	2	1	0	1	2	0
0	1	0	2	0	1	1	1	0	2	1	2	2	1	0	2	2	2
0	1	1	0	0	2	1	1	1	0	0	0	2	1	1	0	1	0
0	1	1	1	2	2	1	1	1	1	0	0	2	1	1	1	0	0
0	1	1	2	2	1	1	1	1	2	1	0	2	1	1	2	1	2
0	1	2	0	1	2	1	1	2	0	1	0	2	1	2	0	0	2
0	1	2	1	2	0	1	1	2	1	0	0	2	1	2	1	0	0
0	1	2	2	2	2	1	1	2	2	1	0	2	1	2	2	1	2
0	2	0	0	2	1	1	2	0	0	2	2	2	2	0	0	2	0
0	2	0	1	2	1	1	2	0	1	1	1	2	2	0	1	0	0
0	2	0	2	1	1	1	2	0	2	2	0	2	2	0	2	0	0
0	2	1	0	1	1	1	2	1	0	0	0	2	2	1	0	1	0
0	2	1	1	0	2	1	2	1	1	2	1	2	2	1	1	2	1
0	2	1	2	1	1	1	2	1	2	2	0	2	2	1	2	2	2
0	2	2	0	2	1	1	2	2	0	0	1	2	2	2	0	2	0
0	2	2	1	2	1	1	2	2	1	1	1	2	2	2	1	1	1
0	2	2	2	0	0	1	2	2	2	1	0	2	2	2	2	1	2

Table 3. Truth value table for a six-input two-output binary variable function

Input n	$n+0$	$n+8$	$n+16$	$n+24$	$n+32$	$n+40$	$n+48$	$n+56$
$u_1u_2u_3u_4u_5u_6$	F_1F_2							
0	3	3	2	1	1	0	0	2
1	3	0	0	1	2	2	1	3
2	3	(2)	1	1	0	3	2	0
3	3	1	3	1	3	1	3	1
4	3	3	0	3	2	3	1	3
5	1	2	(3)	2	2	2	0	2
6	0	1	1	1	2	1	3	1
7	2	0	2	0	2	0	2	0
	Output							

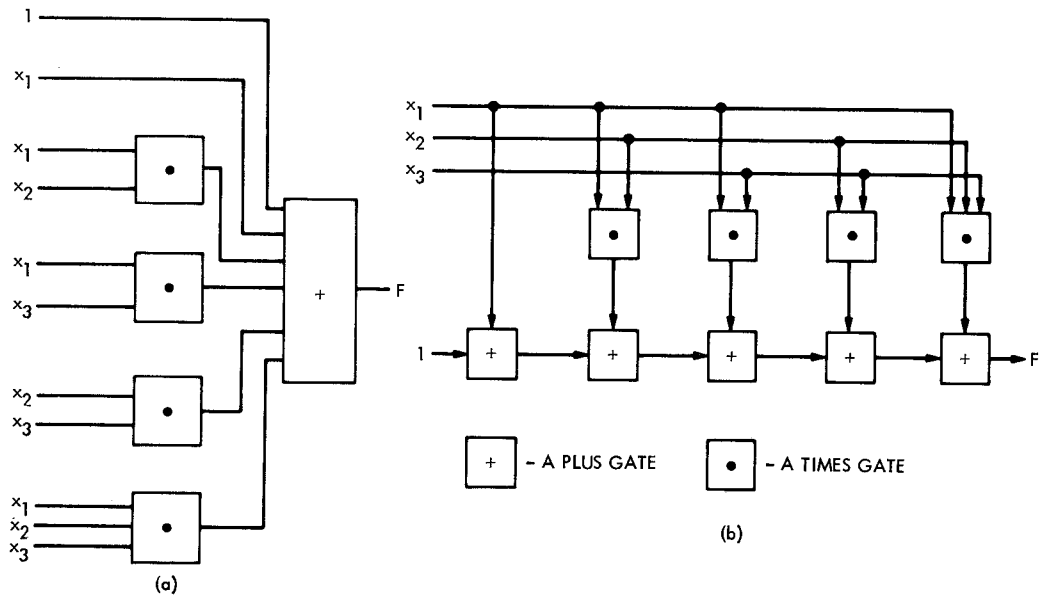

Fig. 1. Two realizations of F