Electronics Research Laboratory
University of California
Berkeley, California
Internal Technical Memorandum M-106

THE DESIGN OF CIRCUITS FOR PERFORMING OPERATIONS AND COMPUTING FUNCTIONS OVER FINITE FIELDS

by

Jean-Paul Jacob

The research herein was supported by the Air Force Office of Scientific Research under Grant AF-AFOSR-639-64.

November 30, 1964

ACKNOWLEDGMENT

The author is grateful to Professor A. Gill for having introduced him to this field.

TABLE OF CONTENTS

THE DESIGN OF CIRCUITS FOR PERFORMING OPERATIONS AND COMPUTING FUNCTIONS OVER FINITE FIELDS*

Jean-Paul Jacob†

I. SUMMARY

This paper is intended to partially bridge the gap between the theory of finite fields and some of its applications, such as circuits for coding and decoding, nonlinear modular sequential circuits, etc. The basic idea is the design of a circuit which multiplies two elements of a Galois Field (as per the rules of this field) in one clock pulse. In other words, the circuit contains no delay components.

Special attention is focused on binary Galois Fields. After we discover how to design a circuit which multiplies any two elements of a finite field, we also know how to design a circuit which realizes any polynomial expression of the elements of the field. Any mapping from a finite field into (or onto) itself can be represented by a polynomial expression. Particular cases of importance are permutation mappings and homomorphisms (automorphisms and isomorphisms).

If we are given p Boolean functions of q binary variables, we can realize them by considering an incompletely specified mapping of G. F. (2^n) into (or onto) itself, where n is such that

$$n \geq \max (p, q)$$

* This work was supported (wholly or in part) by the Joint Services Electronics Programs (U. S. Army, U. S. Navy and U. S. Air Force) under Grant No. AF-AFOSR-139-64.

† The author is on educational leave from IBM Nordic Laboratory, Sweden.

The reader is assumed to be familiar with the theory of finite fields (Galois Fields).

II. INTRODUCTION

In algebraic coding schemes, decoding and possibly error-correction (Ref. 1) is usually performed by a circuit which, among other operations, has to multiply two elements of a Galois Field (Ref. 2). In this sense, an algebraic decoder may be though of as an arithmetic unit which sums and multiplies as per the rules of a certain Galois Field. Our aim is to design such a unit which sums or multiplies in one clock pulse, i.e., with no delay.

Let us designate by G. F. $(p)/[c(x)]$ the field of polynomials, with coefficients from G. F. (p), modulo an irreducible polynomial $c(x)$ of degree n (see Ref. 1, Chapters II and VI). This field has p^n elements which can be represented by the set of all polynomials of the form

$$a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$$

where $a_i \epsilon$ G. F. (p), i.e., a_i is an element of a Galois Field of order p. Notice that G. F. $(p)/[c(x)]$ is isomorphic to any finite field of the same order. The particular problem with which we will be concerned in this paper is to design a combinational network, i.e., a network containing only logic gates, which will multiply two such polynomials. This multiplication problem has been solved, so far, only by circuits employing delays (Ref. 1, Chapter VII).

Besides coding applications, one possible utilization of our circuit is in a nonlinear modular sequential network (Ref. 3), where the normal operation of the circuit is based on delays and one would not like to interrupt the main circuit in order to have multiplication done in a secondary circuit also employing delays.

This paper will only be concerned with circuits where $p = 2$, although the same method can be extended for nonbinary Galois Fields.

Over G. F. $(2)/[c(x)]$, addition is very simple; $c(x)$ is not involved in this operation, since the addition is coefficient-wise. For example, over G. F. $(2)/[1+x^2+x^5]$, we have

$$\begin{array}{r}
1 + 1\cdot x + 1\cdot x^2 + + 1\cdot x^4 \\
1\cdot x + 1\cdot x^2 + 1\cdot x^3 \\
\hline
1 + 1\cdot x^3 + 1\cdot x^4
\end{array},$$

i.e., the addition of the polynomials $1 + x + x^2 + x^4$ and $x + x^2 + x^3$ gives $1 + x^3 + x^4$. Addition, therefore, can be performed by circuits which are similar (the carry-over connection omitted) to those for conventional parallel binary addition.

Let us now observe the mechanism of multiplication, by working out an example. We shall now multiply the two polynomials $1 + x + x^2 + x^4$ and $x + x^2 + x^3$ over G. F. $(2)/[1 + x^2 + x^5]$:

<u>Step 1</u>

$$\begin{array}{cccccccc}
 & & 1 + & x + & x^2 + & x^4 & & \\
 & & & x + & x^2 + & x^3 & & \\
\hline
 & & x^3 + & x^4 + & x^5 & & + & x^7 \\
 & x^2 + & x^3 + & x^4 & & x^6 & & \\
x + & x^2 + & x^3 + & & x^5 & & & \\
\hline
x + & & x^3 + & & & x^6 & + & x^7
\end{array}$$

Step 2

$$
\begin{array}{rl}
 & \quad x^2 + x \\ \cline{2-2}
1 + x^2 + x^5 & \big) \, x + x^3 + x^6 + x^7 \\
 & \quad x^2 + x^4 + x^7 \\ \cline{2-2}
 & \quad x + x^2 + x^3 + x^4 + x^6 \\
 & \quad x + x^3 + x^6 \\ \cline{2-2}
 & \quad x^2 + x^4
\end{array}
$$

Therefore,

$$(1 + x + x^2 + x^4)(x + x^2 + x^3) = x^2 + x^4 \quad (\text{over G. F. } (2)/[1 + x^2 + x^5]).$$

Notice that Step 2, i.e., the "reduction" through division by the irreducible polynomial can also be written as:

$$
\begin{aligned}
x + x^3 + x^6 + x^7 &= x + x^3 + x^5 \cdot x + x^5 \cdot x^2 \\
&= x + x^3 + (1 + x^2) \cdot x + (1 + x^2) \cdot x^2 \\
&= x + x^3 + x + x^3 + x^2 + x^4 = x^2 + x^4 .
\end{aligned}
$$

In what follows we shall formalize the mechanism of this type of multiplication and reduction.

III. BASIC PROPOSITION

Let the polynomials $\sum_{i=0}^{n-1} a_i x^i$ and $\sum_{j=0}^{n-1} b_j x^j$ represent two elements in the field G. F. $(2^n)/[c(x)]$, where $c(x) = \sum_{i=0}^{n} c_i x^i$, $c_n = c_0 = 1$, is an irreducible polynomial. Then

$$\left(\sum_{i=0}^{n-1} a_i x^i\right) \cdot \left(\sum_{j=0}^{n-1} b_j x^j\right) = \sum_{k=0}^{n-1} D_k x^k$$

where $D_i = d_i + \underline{d}\,\underline{C}^{-1}\,\underline{B}_i$

with $d_t = \sum_{j+k=t} a_j b_k$; $t = 0, 1, 2, \ldots, 2n-2$

$$\underline{d} = (d_n, d_{n+1}, \ldots, d_{2n-2})$$

$$\underline{C} = \begin{bmatrix} 1 & 0 & 0 \ \ldots & 0 & 0 \\ c_{n-1} & 1 & 0 & \vdots & \vdots \\ c_{n-2} & c_{n-1} & 1 & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ c_2 & c_3 & c_4 & c_{n-1} & 1 \end{bmatrix} \qquad \underline{B}_i = \begin{bmatrix} c_i \\ c_{i-1} \\ \vdots \\ c_1 \\ c_0 \\ \vdots \\ 0 \end{bmatrix}$$

IV. PROOF OF BASIC PROPOSITION

$$\left(\sum_{i=0}^{n-1} a_i x^i\right) \cdot \left(\sum_{j=0}^{n-1} b_j x^j\right) = \sum_{k=0}^{2(n-1)} d_k x^k \text{ (where } d_k = \sum_{i+j=k} a_i b_j)$$

$$= \left(\sum_{k=0}^{n-1} d_k x^k\right) + d_n x^n + \ldots + d_{2(n-1)} x^{2(n-1)} \qquad (1)$$

Now consider the relations derived from the irreducible polynomial:

$$x^n = \sum_{p=0}^{n-1} c_p x^p \tag{2}$$

$$x^{n+1} = x \cdot x^n = \sum_{p=0}^{n-1} c_p x^{p+1} = \sum_{p=0}^{n-2} c_p x^{p+1} + c_{n-1} x^n \tag{3}$$

$$x^{n+2} = x \cdot x^{n+1} = \sum_{p=0}^{n-3} c_p x^{p+2} + c_{n-1} x^{n+1} + c_{n-2} x^n \ . \tag{4}$$

$$\vdots$$

In order to reduce $x^n, x^{n+1}, \ldots, x^{2(n-1)}$ to polynomials with degree less than n, we must substitute (2) into (3), (2) and (3) into (4), etc.

In this way, we obtain

$$x^n = \sum_{p=0}^{n-1} c_p x^p \tag{2'}$$

$$x^{n+1} = \sum_{p=0}^{n-2} c_p x^{p+1} + c_{n-1} \sum_{p=0}^{n-1} c_p x^p \tag{3'}$$

$$x^{n+2} = \sum_{p=0}^{n-3} c_p x^{p+2} + c_{n-2} \sum_{p=0}^{n-1} c_p x^p + c_{n-1} \sum_{p=0}^{n-2} c_p x^{p+1} + c_{n-1} \sum_{p=0}^{n-1} c_p x^p \tag{4'}$$

Expressions (2'), (3'), ... , (2(n-1)'), are now substituted in (1) and, by grouping together coefficients of equal powers of x, (1) becomes

$$\left[\sum_{i=0}^{n-1} a_i x^i\right] \cdot \left[\sum_{j=0}^{n-1} b_j x^j\right] =$$

$$= \Big\{ d_0 + d_n c_0 + d_{n+1} c_{n-1} c_0 + d_{n+2} (c_{n-2} c_0 + c_{n-1}^2 c_0)$$

$$+ d_{n+3} \left[(c_{n-3} c_0 + c_{n-2} c_{n-1} c_0) + c_{n-1} (c_{n-2} c_0 + c_{n-1}^2 c_0) \right] + \ldots \Big\}$$

$$+ x \Big\{ d_1 + d_n c_1 + d_{n+1} (c_0 + c_{n-1} c_1) + d_{n+2} \left[c_{n-2} c_1 + c_{n-1} (c_0 + c_{n-1} c_1) \right]$$

$$+ d_{n+3} \left[c_{n-3} c_1 + c_{n-2} (c_0 + c_{n-1} c_1) + c_{n-1} \left(c_{n-2} c_1 + c_{n-1} (c_0 + c_{n-1} c_1) \right) \right]$$

$$+ \ldots \Big\} + \ldots \tag{5}$$

One now must note that the coefficient, say D_i, of x^i in this expression is a linear combination of the d_i's,

$$D_i = \sum_{j=0}^{2(n-1)} \partial_{i,j} d_j \tag{*}$$

where each $\partial_{i,j}$ can be thought as being an element of a $n x(2n-1)$ matrix:

$$
\left[\partial_{i,j}\right] = \begin{array}{cc} \underbrace{\begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & & & & \\ 0 & 0 & \cdots & & 1 \end{pmatrix}}_{n \text{ columns}} & \underbrace{\begin{pmatrix} \partial_{0,n} & \cdots & \partial_{0,2n-2} \\ \vdots & & \\ & & \\ \partial_{n-1,n} & \cdots & \partial_{n-1,2n-2} \end{pmatrix}}_{(n-1) \text{ columns}} \end{array} \qquad (**)
$$

and, from (5),

$$
\begin{aligned}
\partial_{0,n} &= c_0 = 1 \\
\partial_{0,n+1} &= c_{n-1}c_0 = c_{n-1}\partial_{0,n} \\
\partial_{0,n+2} &= c_{n-2}c_0 + c_{n-1}c_{n-1}c_0 = c_{n-2}\partial_{0,n} + c_{n-1}\partial_{0,n+1} \\
\partial_{0,n+3} &= c_{n-3}\partial_{0,n} + c_{n-2}\partial_{0,n+1} + c_{n-1}\partial_{0,n+2} \\
&\vdots \\
\partial_{0,2(n-1)} &= c_2\partial_{0,n} + c_3\partial_{0,n+1} + \cdots + c_{n-1}\partial_{0,2n-3}
\end{aligned}
$$

This system of equations can be written in matrix notation.

Define $\underline{\partial}_0 = \mathrm{col}(\partial_{0,n}\ \partial_{0,n+1} \cdots \partial_{0,2n-2})$

$$\underline{A} = \begin{pmatrix} 0 & 0 & \cdots & & 0 \\ c_{n-1} & 0 & \cdots & & 0 \\ c_{n-2} & c_{n-1} & \cdots & & 0 \\ \vdots & & & & \vdots \\ c_2 & c_3 & \cdots & c_{n-1} & 0 \end{pmatrix} \; ; \; \underline{B}_0 = \begin{pmatrix} c_0 \\ 0 \\ \vdots \\ \\ 0 \end{pmatrix}_{(n-1)x1}$$

$$\longrightarrow \underline{\partial}_0 = \underline{A}\,\underline{\partial}_0 + \underline{B}_0$$

In the same way we can determine $\underline{\partial}_1$

$$\partial_{1,n} = c_1$$

$$\partial_{1,n+1} = c_{n-1}c_1 + c_0 = c_{n-1}\partial_{1,n} + c_0$$

$$\partial_{1,n+2} = c_{n-2}\partial_{1,n} + c_{n-1}\partial_{1,n+1}$$

$$\vdots$$

In matrix form

$$\longrightarrow \underline{\partial}_1 = \underline{A}\,\underline{\partial}_1 + \underline{B}_1$$

where $\underline{B}_1 = \mathrm{col}(c_1 c_0\ 0 \ldots 0)_{(n-1)x1}$

More generally

$$\longrightarrow \underline{\partial}_i = \underline{A}\,\underline{\partial}_i + \underline{B}_i \ , \quad i = 1,\ 2,\ \ldots\ ,\ n-1 \tag{6}$$

where $\underline{B}_i = \mathrm{col}(c_i c_{i-1} \cdots c_0\, 0 \ldots 0)_{(n-1)x1}$

(note that $\underline{B}_{n-1} = \mathrm{col}(c_{n-1} \cdots c_2 c_1)$).

We now rewrite (6)

$$(\underline{I} - \underline{A})\,\underline{\partial}_i = \underline{B}_i$$

where $\underline{I} - \underline{A} = \underline{A} - \underline{I} = \underline{A} + \underline{I} \triangleq \underline{C}$

(Note that $\det C = 1$)

Therefore $\underline{\partial}_i = \underline{C}^{-1}\underline{B}_i$ (7)

$$\text{where } \underline{C} \triangleq \begin{pmatrix} 1 & 0 & 0 & & \cdots & 0 \\ c_{n-1} & j & 0 & & \cdots & 0 \\ c_{n-2} & c_{n-1} & 1 & & \cdots & 0 \\ \vdots & & & & & \\ c_2 & c_3 & c_4 & \cdots & c_{n-1} & 1 \end{pmatrix} \quad (n-1) \times (n-1)$$

Finally substituting (7) in (*) (also observe (**)) we obtain

$$D_i = d_i + \underline{d}\,\underline{C}^{-1}\,\underline{B}_i \qquad \text{Q. E. D.} \qquad (8)$$

V. COMMENTS ON ABOVE RESULTS

1. <u>First circuit interpretation of the basic proposition.</u>

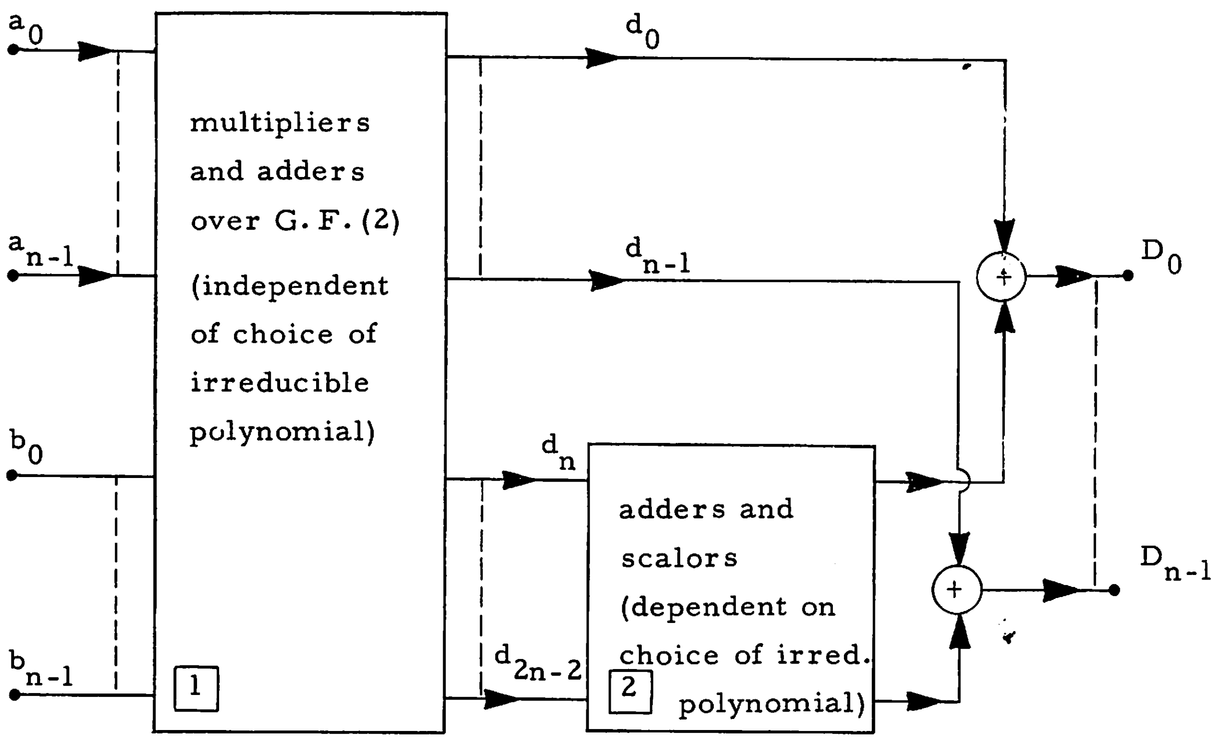
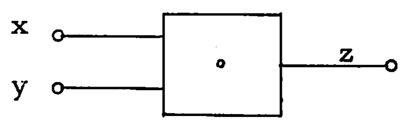


Fig. 1

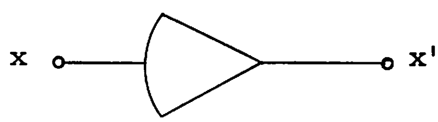
Fig. 1 pictures the block configuration of a circuit which will multiply two polynomials, none of which is known a priori. Note that box 1 contains only multipliers and adders over G.F. (2). A multiplier over G.F. (2) is simply an AND gate and the modulo two adder is an "exclusive OR" circuit (e.g., two AND gates and one OR gate).



y \ x	0	1
0	0	0
1	0	1

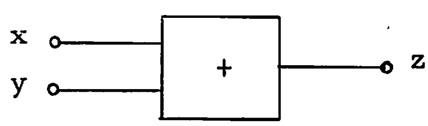
z - table

AND gate



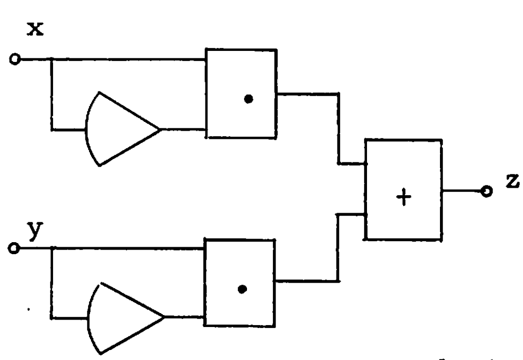
x	x'
0	1
1	0

inverter

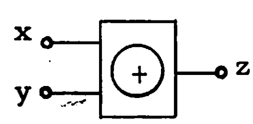


y \ x	0	1
0	0	1
1	1	1

OR gate



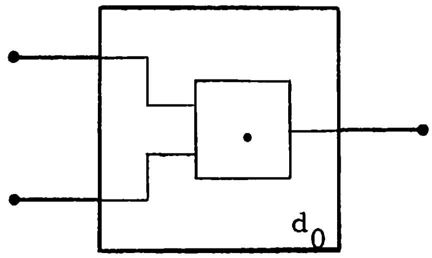
y \ x	0	1
0	0	1
1	1	0



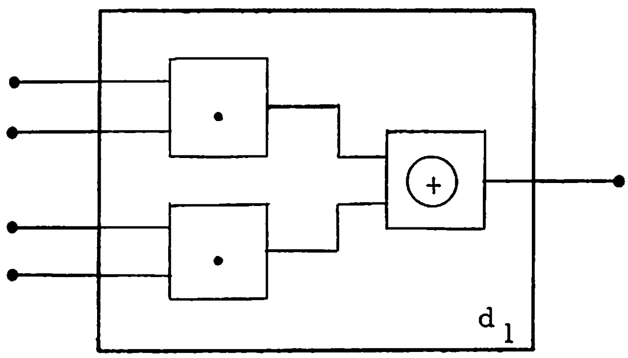
exclusive OR

Box 1 is modular in the sense that it is composed of standard circuits which produce d_0, d_1, d_2, etc.

$d_0 = a_0 b_0$



$d_1 = a_1 b_0 + b_1 a_0$



Given the highest degree n of the input polynomials we just need to place in box 1 the circuits labelled d_0, d_1, $\cdots$, d_n. Each of these circuits is a standard unit which does not depend on the irreducible polynomial or degree of input polynomials. Also for some $i \neq j$ we will have a common circuit which can be labelled d_i or d_j.

Box 2 is a box which only depends on the particular choice of the irreducible polynomial. Once this has been selected we compute

$$\underline{\alpha}_i = \underline{C}^{-1} \underline{B}_i \quad , \quad i = 0, 1, \cdots, n-1$$

Remember that $\alpha_i \in \{0, 1\}$, which means that the multiplication by a scalor is done either by a simple connection or none at all.

Examples:

I-a. In G.F. $[2^4] / (1 + x + x^4)$ (see Fig. 2).

$$c_0 = 1 \qquad c_2 = c_3 = 0$$

$$c_1 = 1 \qquad c_4 = 1$$

$$\underline{C} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \underline{C}^{-1}$$

$$\left.\begin{array}{lllll} \underline{C}^{-1}\ \underline{B}_0 = \underline{B}_0 = \text{col}\ (100) & \Longrightarrow & D_0 & = & d_0 + d_4 \\ \underline{C}^{-1}\ \underline{B}_1 = \underline{B}_1 = \text{col}\ (110) & \Longrightarrow & D_1 & = & d_1 + (d_4 + d_5) \\ \underline{C}^{-1}\ \underline{B}_2 = \underline{B}_2 = \text{col}\ (011) & \Longrightarrow & D_2 & = & d_2 + (d_5 + d_6) \\ \underline{C}^{-1}\ \underline{B}_3 = \underline{B}_3 = \text{col}\ (001) & \Longrightarrow & D_3 & = & d_3 + d_6 \end{array}\right\} \text{(a)}$$

I-b. In G. F. $[2^4]/(1 + x + x^2 + x^3 + x^4)$ (see Fig. 3).

$$c_0 = c_1 = c_2 = c_3 = c_4 = 1$$

$$\underline{C} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{pmatrix} \qquad \underline{C}^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

$$\left.\begin{array}{lllll} \underline{C}^{-1}\ \underline{B}_0 = \text{col}\ (110) & \Longrightarrow & D_0 & = & d_0 + (d_4 + d_5) \\ \underline{C}^{-1}\ \underline{B}_1 = \text{col}\ (101) & \Longrightarrow & D_1 & = & d_1 + (d_4 + d_6) \\ \underline{C}^{-1}\ \underline{B}_2 = \text{col}\ (100) & \Longrightarrow & D_2 & = & d_2 + d_4 \\ \underline{C}^{-1}\ \underline{B}_3 = \text{col}\ (100) & \Longrightarrow & D_3 & = & d_3 + d_4 \end{array}\right\} \text{(b)}$$

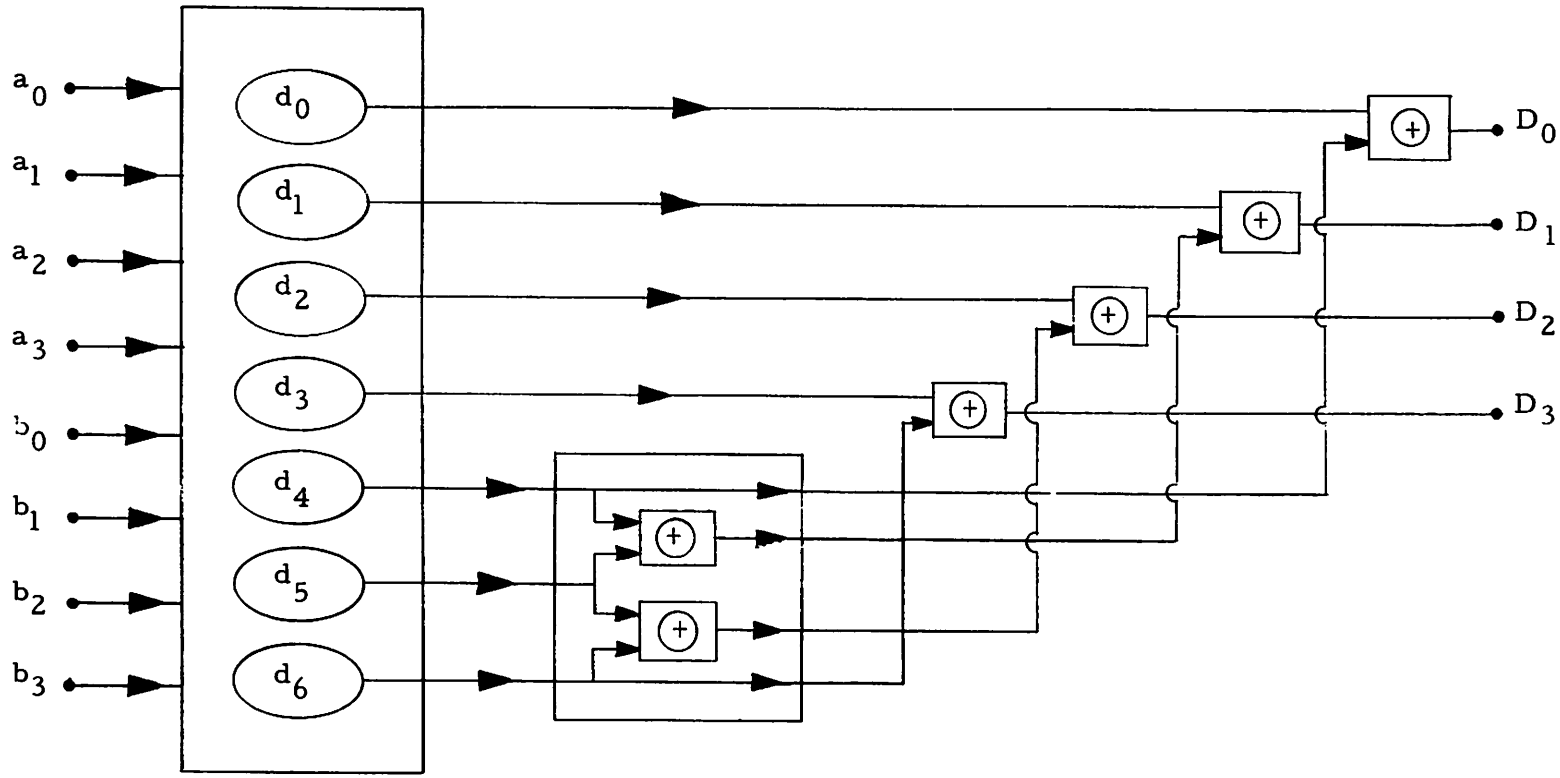


Fig. 2. This circuit requires 16 "AND" gates and 15 "exclusive OR" gates.

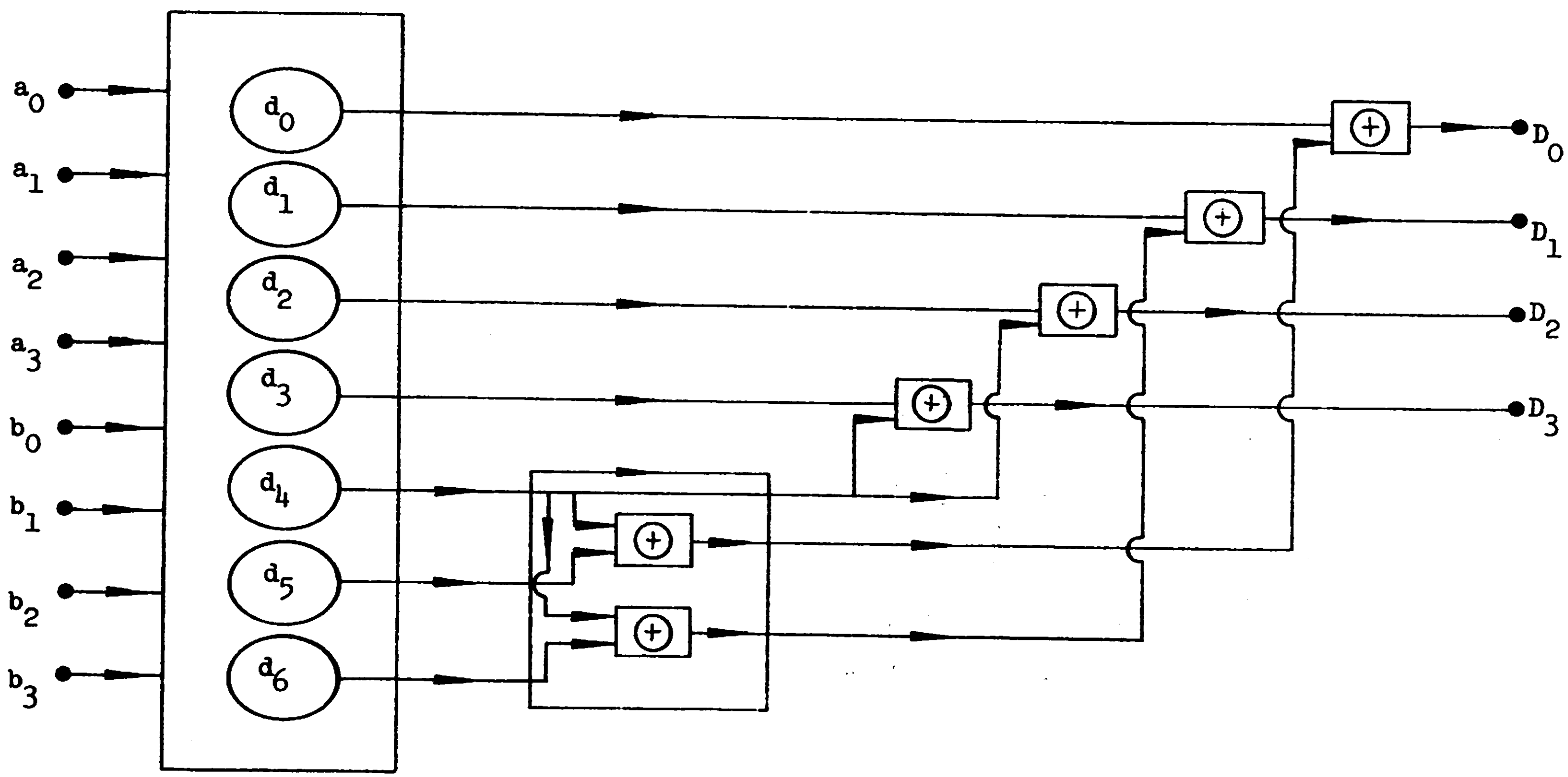


Fig. 3. Multiplication over G. F. $[2^4] / (1 + x + x^2 + x^3 + x^4)$.

It is important to note that Figs. 2 and 3 are only possible realizations of the canonical forms (a) and (b). We call them the standard realizations because they use standard circuits $d_0, d_1, \dots, d_n$. Obviously the canonical forms (a) and (b) may possibly lead to simpler circuit designs. One could think, for instance, of rewriting in (b)

$$
\begin{aligned}
d_4 + d_5 &= a_1 b_3 + a_2 b_2 + a_3 b_1 + a_2 b_3 + a_3 b_2 \\
&= a_1 b_3 + a_2 (b_2 + b_3) + a_3 (b_1 + b_2) \\
d_4 + d_6 &= a_1 b_3 + a_2 b_2 + a_3 b_1 + a_3 b_3 \\
&= a_1 b_3 + a_2 b_2 + a_3 (b_1 + b_3) .
\end{aligned}
$$

We may think that due to common factors on both the above expressions, this manipulation will lead to a more economical realization of the canonical form. This is not the case here or in a few other examples we have worked out. Apparently, in most cases the standard realization has the double advantage of simplicity and economy.

II. Second circuit interpretation of the basic proposition (multiplication by a fixed polynomial).

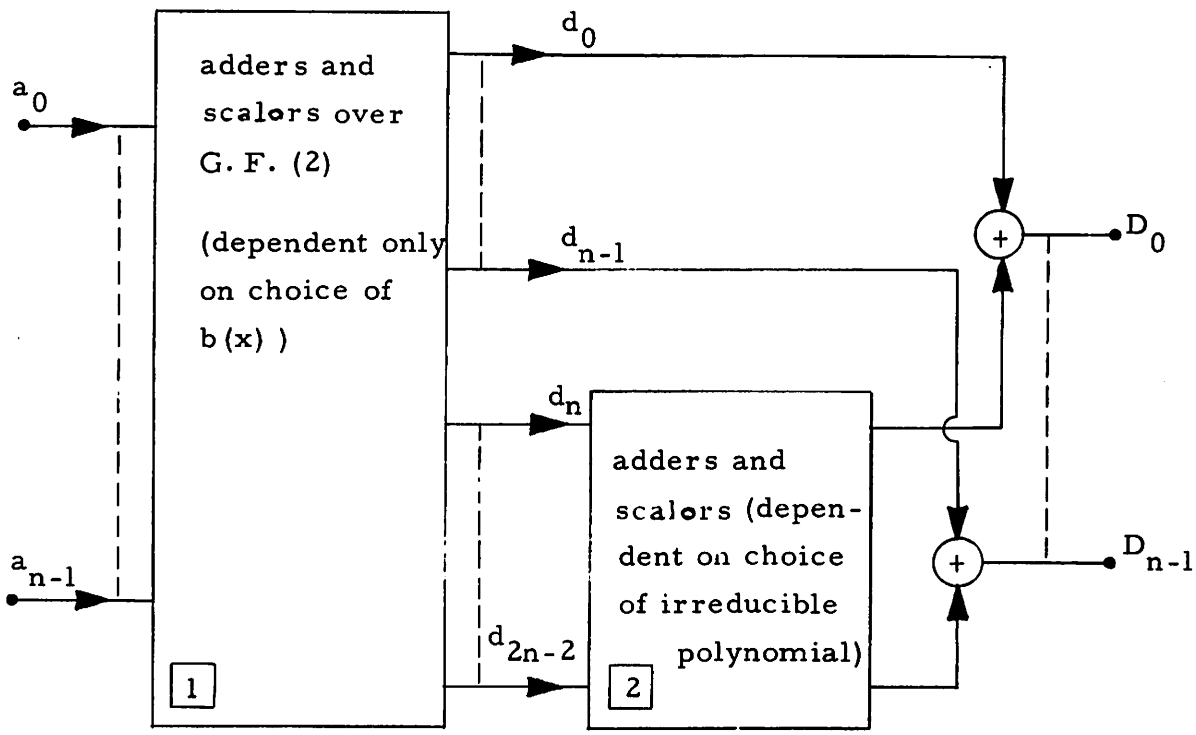


Fig. 4

When $b(x)$ is a fixed polynomial, which is the case when decoding an algebraic code, the internal structure of box 1 is different from case I. Now the expressions

$$d_0 = a_0 b_0$$
$$d_1 = a_0 b_1 + a_1 b_0$$
$$d_2 = a_0 b_2 + a_1 b_1 + a_2 b_0$$
$$\vdots$$

become only linear combinations of the inputs a_i, the coefficients being the known b_i's.

Notice that now boxes 1 and 2 have the same type of structure; they simply perform several binary linear combinations of their inputs.

Box 2 is, for a given irreducible polynomial, exactly the same as we use in case I. For example, suppose we want to multiply by x over G.F. $[2^4] / (1 + x + x^4)$.

$$b_0 = b_2 = b_3 = 0$$
$$b_1 = 1$$

$$d_0 = 0$$
$$d_1 = a_0 b_1 = a_0$$
$$d_2 = a_1 b_1 = a_1$$
$$d_3 = a_2 b_1 = a_2$$
$$d_4 = a_3 b_1 = a_3$$
$$d_5 = 0$$
$$d_6 = 0$$

From example I-a:

$$\left.\begin{aligned} D_0 &= d_0 + d_4 = a_3 \\ D_1 &= d_1 + (d_4+d_5) = a_0 + a_3 \\ D_2 &= d_2 + (d_5+d_6) = a_1 \\ D_3 &= d_3 + d_6 = a_2 \end{aligned}\right\} \quad (c)$$

Canonical form (c) can be realized by a standard circuit. By redrawing it one obtains

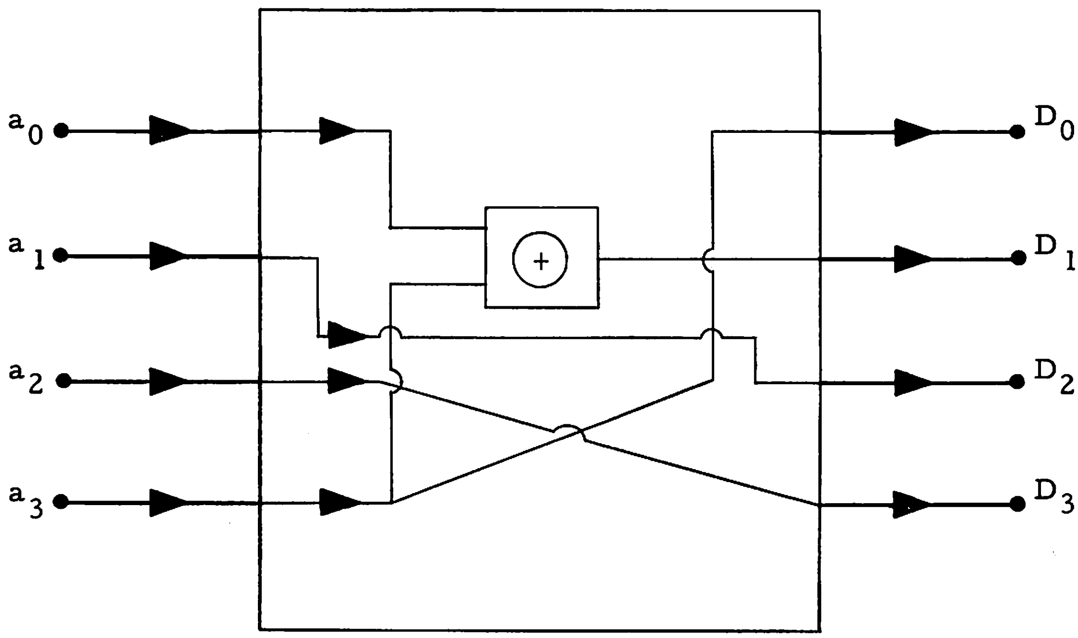


Fig. 5. Multiplication by x over G.F. $[2^4] / (1+x+x^4)$.

Observe that if we reverse the direction of all arrows, hence considering the input at the right hand side and the output at the left hand side, then the circuit of Fig. 5 will perform the division by x over G.F. $[2^4]$ / $(1 + x + x^4)$. This is a consequence of properties of linear binary transformations (Ref. 4).

One can also use canonical form (c) to compute

$$(a_0 + a_1 x + a_2 x^2 + a_3 x^3) \cdot x = D_0 + D_1 x + D_2 x^2 + D_3 x^3$$
$$= a_3 + (a_0 \oplus a_3) x + a_1 x^2 + a_2 x^3 .$$

For instance

$$(1 + x + x^3) \cdot x = 1 + x^2 \text{ over G. F. } [2^4] / (1 + x + x^4) .$$

VI. CONCLUSIONS

In this first part, a method was presented for designing a circuit capable of multiplying two Galois Field elements in one clock pulse. Since any finite field is isomorphic to a Galois Field, our results apply to the multiplication of any two elements of a finite field provided one designs the hardware which realizes the isomorphism.

The algebraic conclusions which we reached also present a way for the analytical multiplication of two polynomials as per the rules of a Galois Field. Friedland and Stern (Ref. 5) have shown that to multiply two polynomials $a(x)$ and $b(x)$ modulo a polynomial $c(x)$, one may define

$$\underline{a} = \begin{bmatrix} a_{n-1} \\ a_{n-2} \\ \vdots \\ a_0 \end{bmatrix}, \quad \underline{b} = \begin{bmatrix} b_{n-1} \\ b_{n-2} \\ \vdots \\ b_0 \end{bmatrix}, \quad \underline{D} = \begin{bmatrix} D_{n-1} \\ D_{n-2} \\ \vdots \\ D_0 \end{bmatrix}, \quad \underline{Q} = \begin{bmatrix} c_{n-1} & 1 & 0 \cdots 0 \\ c_{n-2} & 0 & 1 \cdots 0 \\ \vdots & & \\ c_0 & 0 & 0 \cdots 0 \end{bmatrix}$$

where if $D(x) = a(x)$, then

$$\underline{D} = a(\underline{Q}) \cdot \underline{b} = b(\underline{Q}) \cdot \underline{a} .$$

$\underline{D}$ is therefore given by a matrix polynomial expression and one is required to elevate matrix $\underline{Q}$ to the power m where

$$m = \min \{ \text{degree of } a(x), \text{degree of } b(x) \} .$$

The result of our basic proposition, (8), is an alternative way to obtain $\underline{D}$. In our result, the number of matrix multiplications is a constant, independent of the degree of the polynomials involved. To rewrite Eq. (8), i.e.,

$$D_i = d_i + \underline{d}\,\underline{C}^{-1}\,\underline{B}_i$$

entirely in terms of matrices, one may observe that

$$\begin{bmatrix} d_0 \\ d_1 \\ \vdots \\ d_{n-1} \end{bmatrix} = \begin{bmatrix} a_0 & 0 & \cdots & 0 \\ a_1 & a_0 & & \vdots \\ \vdots & & \ddots & \vdots \\ a_{n-1} & a_{n-2} & \cdots & a_0 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix}$$

and

$$\begin{bmatrix} d_n \\ d_{n+1} \\ \vdots \\ d_{2n-2} \end{bmatrix} = \begin{bmatrix} a_n & a_{n-1} & \cdots & a_1 \\ 0 & a_n & \cdots & a_2 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & a_n & a_{n-1} \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_{n-1} \end{bmatrix} .$$

In the example illustrated by Fig. 5, we commented that the circuit for division by a given polynomial is the same as the multiplying one, provided that input and output arrows are reversed. This does not, however, solve the general problem of division of any element $a(x) \in$ G. F. (2^n) by any $b(x) \in$ G. F. (2^n). This division is realized by multiplying $a(x)$ by the inverse of $b(x)$. The inverse of $b(x)$ in G. F. (2^n) is obtained, for instance, by elevating $b(x)$ to the $(2^n - 2)$ power, this operation requiring $(n-1)$ squaring circuits.[6] Alternatively the combinational network for inverting an element can be obtained directly by rewriting the set of equations (9) in terms of matrices and letting

$$D_o = 1$$

$$D_i = 0, \quad i = 1, 2, \ldots, n-1$$

we obtain

$$\underline{I} = \underline{A}\,\underline{b} + \underline{B}'\,\underline{C}'^{-1}\,\underline{A}^*\,\underline{b} \qquad (10)$$

where the prime indicates transposition and $\underline{B}$ is an $(n-1) \times n$ matrix whose columns are $\underline{B}_i$. $\underline{I}$ is the $n \times 1$ column matrix col. $(1, 0, \ldots, 0)$.

From (10), supposing it has a solution for $\underline{b}$,

$$\underline{b} = (\underline{A} + \underline{B}'\,\underline{C}'^{-1}\underline{A}^*)^{-1}\,\underline{I}$$

This expression determines the combinational network which produces the coefficients b_i of the inverse element of $a(x)$ in G. F. (2^n). Observing the particular structure of $\underline{I}$, we conclude that the right hand side represents only the first column of $(\underline{A} + \underline{B}'\,\underline{C}'^{-1}\underline{A}^*)^{-1}$. One should once more remember that matrices $\underline{B}$ and $\underline{C}$ are entirely determined by the irreducible polynomial $c(x)$.

VII. APPLICATIONS OF THE STANDARD REALIZATION TO NONLINEAR MODULAR SEQUENTIAL CIRCUITS

In the preceding sections, it was shown how we can build a combinational circuit which multiplies in one clock pulse two elements of a Galois Field, denoted G. F. $(2^n)/[c(x)]$ as per the rules of this field.

In the following sections we will show how, by properly combining together standard realizations, one can perform any nonlinear function mapping G. F. (2^n) into or onto itself. An important particular case is the group of automorphisms of a finite field onto itself. Isomorphisms will be discussed and it will also be shown how the synthesis of Boolean functions can be performed using finite field mapping techniques.

Firstly, we will show how our basic standard circuit is simplified when, instead of multiplying two different elements of G. F. (2^n), we multiply an element by itself (squaring). The generation of the n^{th} power of an element of G. F. (2^n) follows immediately.

Secondly, we will show that any nonlinear function mapping a Galois Field into (or onto) itself can be represented by a polynomial in an indeterminate which assumes the values from G. F. (2^n). Standard circuits are then combined to perform a polynomial mapping, i. e., any mapping can be practically realized. Particular cases of isomorphisms and automorphisms are exemplified and an alternate way of multiplying two elements of a Galois Field is suggested.

VIIa. STANDARD REALIZATION FOR SQUARING AN ELEMENT OF G. F. $(2^n)/[c(x)]$

Referring back to our basic proposition (see, for instance, page 4) for the case when $a(x) \equiv b(x) \in$ G. F. $(2^n)/[c(x)]$, the following simplifications are found:

$$d_0 = a_0 b_0 = a_0^2 = a_0$$
$$d_1 = a_1 b_0 + a_0 b_1 = a_1 a_0 + a_0 a_1 = 0$$
$$d_2 = a_0 b_2 + a_1 b_1 + a_2 b_0 = a_0 a_2 + a_1^2 + a_2 a_0 = a_1^2 = a_1$$
$$\vdots$$

In general:

$$d_{2p} = a_p$$

$$d_{2p+1} = 0 \quad p = 0, 1, 2, \ldots, (n-1)$$

These relations reflect as a considerable simplification in the standard realization of our multiplying circuit (Fig. 1) because they entirely eliminate the need of box 1. In other words, no more logic circuitry is necessary to generate $d_0, d_1, d_2, \ldots, d_{2n-2}$. The following examples should help to visualize this.

Example Ia.

In G. F. $(2^4)/(1 + x + x^4)$, the squaring circuit will take the form of Fig. 6 (compare with Fig. 2). If for instance, $a(x) = 1 + x + x^3$, i.e., $a_0 = 1$, $a_1 = 1$, $a_2 = 0$, $a_3 = 1$, we would obtain $D_0 = 1$, $D_1 = 0$, $D_2 = 0$, $D_3 = 1$, and conclude that

$$[a(x)]^2 = 1 + x^3$$

It is interesting to notice that, in Fig. 6, if we reverse the direction of all arrows leaving or arriving at terminals (see Fig. 7) the circuit will now find the square root of the input. The reason is that the coefficients D_i are obtained through an invertible linear transformation from the coefficients a_i, as can be seen from Fig. 6.

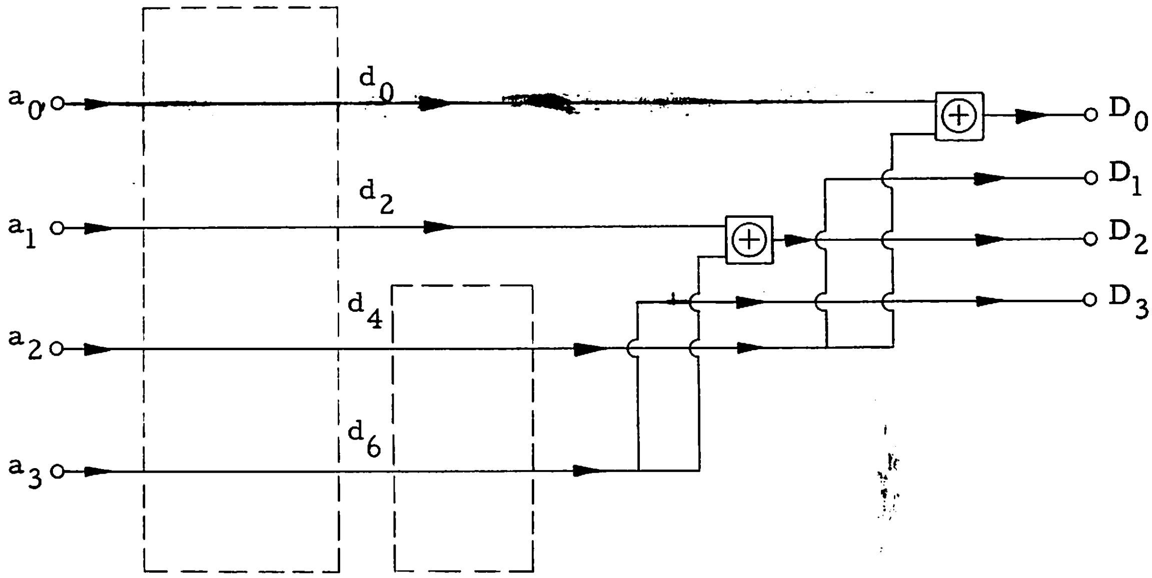


Fig. 6. Squaring circuit over G. F. $(2^4)/(1 + x + x^4)$.

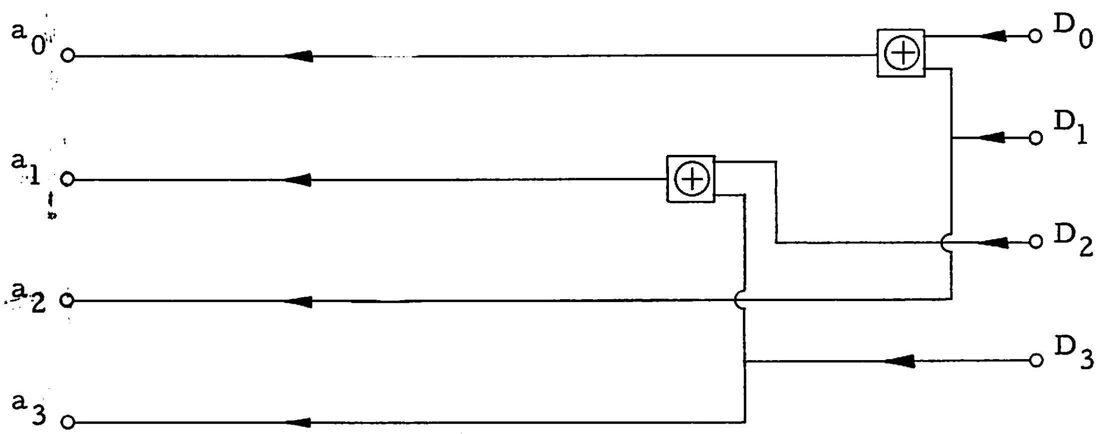


Fig. 7. Square root circuit over G. F. $(2^4)/(1 + x + x^4)$.

$$\begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ D_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

If, for instance, $D(x) = x$, i.e., $D_0 = 0$, $D_1 = 1$, $D_2 = 0$, $D_3 = 0$, we would obtain $a_0 = 1$, $a_1 = 0$, $a_2 = 1$, $a_3 = 0$, and conclude that

$$[D(x)]^{1/2} = 1 + x^2$$

Example Ib.

In G. F. $(2^4)/(1 + x + x^2 + x^3 + x^4)$, the squaring circuit will take the form of Fig. 8 (compare with Fig. 3).

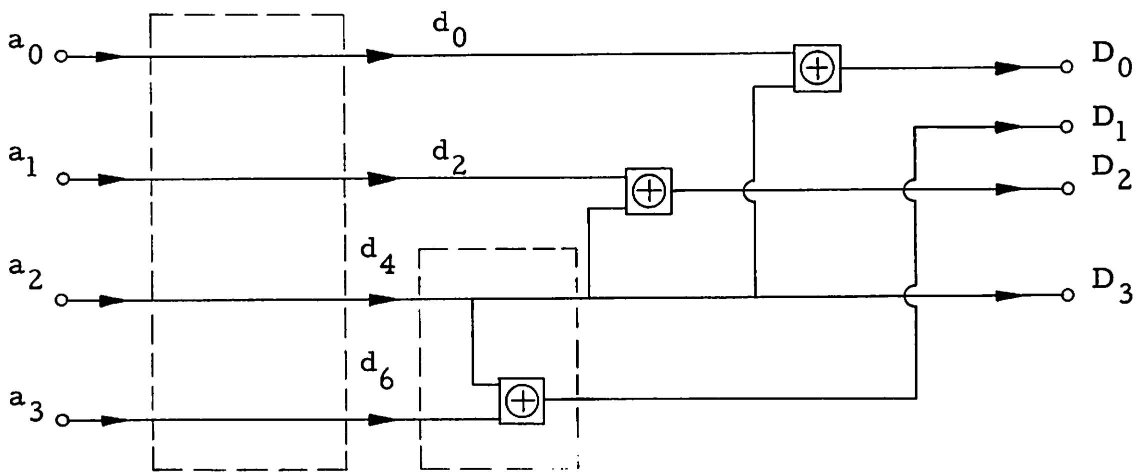


Fig. 8. Squaring circuit over G. F. $(2^4)/(1 + x + x^2 + x^3 + x^4)$.

VIIb. STANDARD REALIZATION FOR HIGHER POWERS OF AN ELEMENT OF G. F. $(2^n)/[c(x)]$

From the results obtained in the preceding paragraphs, the straightforward way of cubing an element $a(x)$ of G. F. $(2^n)/[c(x)]$ is to multiply $a(x)$ by the output of a squaring circuit (Fig. 9).

Once more one should comment that simplifications may appear after one specifies the two circuits appearing in the two main blocks of Fig. 9.

The fourth power of an element can be obtained by cascading two squaring circuits.

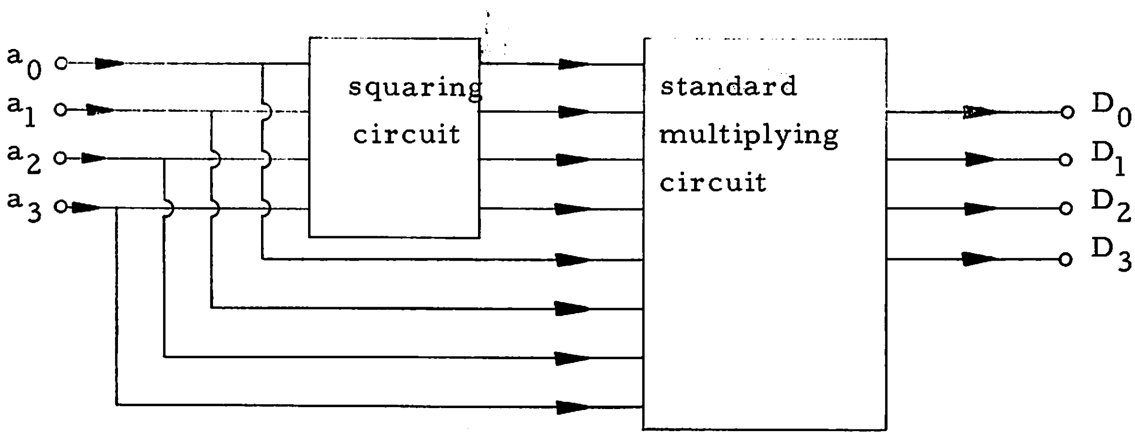


Fig. 9. Standard realization of a cubing circuit.

It is not difficult now to realize how the other powers are obtained and how any polynomial expression with coefficients from G. F. (2^n) can be realized by conveniently connecting standard circuits. We shall see, moreover, that in some important particular cases, the polynomials we want to represent are of a simple form.

VIIc. REMARKS ON MAPPINGS FROM G. F. (2^n) INTO (AND ONTO) ITSELF

We now proceed to define the "indicating function" of an element of G. F. (2^n). Let $f_{\alpha_i}(x)$ be a function whose domain is G. F. (2^n).

We say that $f_{\alpha_i}(x)$ ~~is the~~ indicator function of $\alpha_i \in$ G. F. (2^n) iff

$$f_{\alpha_i}(\alpha_i) = 1 \quad \text{and} \quad f_{\alpha_i}(\alpha_j) = 0, \qquad j \neq i, \quad \alpha_0 = 0$$

The indicator function $f_{\alpha_i}(x)$ of any element $a_i \in$ G. F. (2^n) is a polynomial in x of degree $2^n - 1$ defined by

$$f_{\alpha_i}(x) = \prod_{\substack{j \neq i \\ j=0}}^{2^n-1} (x - \alpha_j) = \frac{x^{2^n} - x}{x - \alpha_i}$$

I) $f_{\alpha_i}(\alpha_j) = 0, \quad j \neq i, \quad j = 0, 1, \ldots, \quad 2^n - 1$

Proof: Immediate consequence of the definition above, since one of the factors under the product sign is zero.

II) $f_{\alpha_i}(\alpha_i) = 1$

Proof: From the definition above,

$$f_{\alpha_i}(\alpha_i) = \prod_{\substack{j \neq 1 \\ j=0}}^{2^n-1} (\alpha_i - \alpha_j) = \prod_{p=1}^{2^n-1} \alpha_p$$

Use now the fact that the product of all nonzero elements of a finite field is unity; indeed, as any $\alpha_i \in$ G. F. (2^n) satisfies the relation

$$\alpha_i^{2^n} - \alpha_i = 0$$

we have

$$x^{2^n} - x = (x - \alpha_0)(x - \alpha_1) \dots (x - \alpha_{2^n-1})$$

or

$$x^{2^n-1} - 1 = (x - \alpha_1) \dots (x - \alpha_{2^n-1})$$

Comparing the coefficients of equal powers of x in this last relation we obtain, for the constant terms:

$$\prod_{p=1}^{2^n-1} \alpha_p = 1$$

and, finally,

$$f_{\alpha_i}(\alpha_i) = \prod_{p=1}^{2^n-1} \alpha_p = 1$$

Hence the indicator function of $\alpha_i \in$ G. F. (2^n) is a polynomial of degree $2^n - 1$ with coefficients from G. F. (2^n).

<u>Example</u>:

Find the indicator functions for the elements of G. F. $(2^2)/(1+x+x^2)$.

Call:

$$\alpha_0 = 0$$
$$\alpha_1 = x$$
$$\alpha_2 = 1 + x$$
$$\alpha_3 = 1$$

I) $f_{\alpha_0}(y) = \dfrac{y^4 - y}{y} = y^3 + 1$

II) $f_{\alpha_2}(y) = y(y - 1)(y - x) = y[y^2 - (1 + x)y + x] = y^3 + \alpha_2 y^2 + \alpha_1 y$

III) $f_{\alpha_1}(y) = y(y-1)(y-1-x) = y[y^2 - xy + (1 + x)] = y^3 + \alpha_1 y^2 + \alpha_2 y$

IV) $f_{\alpha_3}(y) = \dfrac{y^4 - y}{y - 1} = y^3 + y^2 + y$.

One may verify the properties of the indicator functions, such as

$$f_{\alpha_0}(\alpha_2) = \alpha_2^3 - 1 = \alpha_3 - 1 = 0$$

$$f_{\alpha_3}(\alpha_3) = \alpha_3^3 + \alpha_3^2 + \alpha_3 = \alpha_3 + \alpha_3 + \alpha_3 = \alpha_3 = 1 \quad .$$

We next show how one can describe analytically any correspondence from G. F. (2^n) to itself which is given by a table $\varphi(\alpha_i)$. It can easily be verified that

$$\varphi(x) = \sum_{i=0}^{2^n-1} f_{\alpha_i}(x)\, \alpha_{k_i} = \beta_{2^n-1} x^{2^n-1} + \dots + \beta_1 x + \beta_0 \quad ,$$

where $\beta_j \in$ G. F. (2^n), has the property that

$$\varphi(a_j) = \sum_{i=0}^{2^n-1} f_{\alpha_i}(\alpha_j)\, \alpha_{k_i} = f_{\alpha_j}(\alpha_j)\, \alpha_{k_j} = \alpha_{k_j}, \quad j = 0, 1, \dots, 2^n-1 \quad .$$

In the sequel, some particular types of mapping $\varphi(x)$ will be studied.

VIId. PERMUTATION MAPPINGS

If $\varphi(x)$ represents a one-to-one mapping of G. F. (2^n) onto itself it is called a permutation polynomial.

A permutation polynomial $\varphi_p(x)$ has the property that

$$\varphi_p(\alpha_i) \neq \varphi_p(\alpha_j) \quad \text{iff} \quad i \neq j \qquad i = 0, 1, 2, \dots, 2^n-1$$

It can be proved[7] that a given permutation mapping corresponds to one and only one polynomial. There are $2^n!$ permutation mappings (including the identity) over G. F. (2^n), n of which are given by the very simple polynomials

$$\varphi_p(x) = x^{2^i} \qquad i = 0, 1, 2, \ldots, n-1$$

Dickson[7] proves that

$$\varphi(x) = \sum_{i=0}^{n-1} \beta_i x^{2^i}, \quad \beta_i \in \text{G. F. } (2^n),$$

is a permutation polynomial iff zero is the only solution of $\varphi(x) = 0$, in G. F. (2^n).

The circuit interpretation of these permutation polynomials is quite simple. Indeed, the simplicity of a squaring circuit $\varphi(x) = x^2$ was illustrated in Fig. 6. The simplicity of this type of circuits is a consequence of the fact that they represent nonsingular linear transformations.

By cascading two squaring circuits we obtain $f(x) = x^{2^2}$ and the cascading of p identical squaring circuits gives us x^{2^p}. Note that the cascading of two identical circuits sometimes can be represented by a simpler equivalent circuit, mainly when one has in hands a component which can handle modulo 2 addition of more than two binary variables, as described in Ref. 8.

Example:

Consider again Fig. 6, which is a squaring circuit over G. F. $(2^4)/(1 + x + x^4)$. If we designate the elements of this field as:

TABLE 1

$\alpha_0 = 0$

$\alpha_1 = 1$

$\alpha_2 = 1 + x$

$\alpha_3 = x$

$\alpha_4 = 1 + x + x^2$

$\alpha_5 = 1 + x^2$

$\alpha_6 = x + x^2$

$\alpha_7 = x^2$

$\alpha_8 = 1 + x + x^2 + x^3$

$\alpha_9 = 1 + x^2 + x^3$

$\alpha_{10} = 1 + x^3$

$\alpha_{11} = x^3$

$\alpha_{12} = x + x^2 + x^3$

$\alpha_{13} = x + x^3$

$\alpha_{14} = x^2 + x^3$

$\alpha_{15} = 1 + x + x^3$

then the circuit represented in Fig. 6 represents the permutation:

$$\varphi(\alpha_i) = \alpha_{u_i}, \quad \text{where}$$

TABLE 2

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
μ_i	0	1	5	7	6	3	4	2	13	12	9	14	15	11	8	10

We will show below that the mapping represented in Table 2 is an automorphism. It will also become clear that all automorphisms of G. F. (2^4) can be read from Table 2, as well as their corresponding permutation polynomials. Furthermore, if one isomorphism of G. F. (2^4) can be found, all other isomorphisms will be obtained with the help of Table 2.

In Table 1, the choice of a particular i such that α_i represents a certain element of G. F. $(2^4)/(1 + x + x^4)$ was determined haphazardly. There is, however, a more convenient way of naming the elements of a finite field which leads to interesting conclusions. If we call β a primitive root, i.e., a root of the primitive polynomial $1 + x + x^4$ in G. F. (2^4), then any nonzero element of G. F. $(2^4)/(1+x+x^4)$ can be represented as a power of β, say β^i, and any power p of this element β^i in the field is

$$\beta^{pi\,(\text{modulo } 2^4-1)}$$

More generally, in any G. F. (2^n), suppose we define

$$\alpha_i = \beta^i$$

and realize the one-to-one mapping $\Phi(\alpha_i) = \alpha_{u_i}$, where by definition α_{u_i} is the element

$$\alpha_{u_i} = a_0 + a_1 x + \dots + a_{n-1}x^{n-1}$$

where $a_0 a_1 \dots a_{n-1}$ is the binary representation of i. Then, for any $\alpha_i \neq 0$, $\alpha_j \neq 0$,

$$\alpha_i \cdot \alpha_j = \Phi^{-1}\left(\Phi(\alpha_i) \dot{+} \Phi(\alpha_j)\right)$$

where the dot over the addition sign represents the modulo $(2^n - 1)$ addition of two binary numbers.

One should notice that this represents a completely different approach to the problem of multiplying two elements, say α_i and a_j of G. F. (2^n). We first map α_i and α_j into α_{u_i} and α_{u_j} where the coefficients of α_{u_i} and α_{u_j} are the binary representation of i and j and then add modulo $(2^n - 1)$ these coefficients; finally we realize the inverse mapping of the result.

Example:

Let us consider once more G. F. $(2^4)/(1 + x + x^4)$. The procedure described above leads to the following mapping:

$\alpha_0 = 1$	$\longrightarrow 0\ 0\ 0\ 0$	$\longrightarrow \alpha_{15}$
$\alpha_1 = x$	$\longrightarrow 0\ 0\ 0\ 1$	$\longrightarrow \alpha_0$
$\alpha_2 = x^2$	$\longrightarrow 0\ 0\ 1\ 0$	$\longrightarrow \alpha_1$
$\alpha_3 = x^3$	$\longrightarrow 0\ 0\ 1\ 1$	$\longrightarrow \alpha_4$
$\alpha_4 = 1 + x$	$\longrightarrow 0\ 1\ 0\ 0$	$\longrightarrow \alpha_2$
$\alpha_5 = x + x^2$	$\longrightarrow 0\ 1\ 0\ 1$	$\longrightarrow \alpha_8$
$\alpha_6 = x^2 + x^3$	$\longrightarrow 0\ 1\ 1\ 0$	$\longrightarrow \alpha_5$
$\alpha_7 = 1 + x + x^3$	$\longrightarrow 0\ 1\ 1\ 1$	$\longrightarrow \alpha_{10}$
$\alpha_8 = 1 + x^2$	$\longrightarrow 1\ 0\ 0\ 0$	$\longrightarrow \alpha_3$
$\alpha_9 = x + x^3$	$\longrightarrow 1\ 0\ 0\ 1$	$\longrightarrow \alpha_{14}$
$\alpha_{10} = 1 + x + x^2$	$\longrightarrow 1\ 0\ 1\ 0$	$\longrightarrow \alpha_9$
$\alpha_{11} = x + x^2 + x^3$	$\longrightarrow 1\ 0\ 1\ 1$	$\longrightarrow \alpha_7$
$\alpha_{12} = 1 + x + x^2 + x^3$	$\longrightarrow 1\ 1\ 0\ 0$	$\longrightarrow \alpha_6$
$\alpha_{13} = 1 + x^2 + x^3$	$\longrightarrow 1\ 1\ 0\ 1$	$\longrightarrow \alpha_{13}$
$\alpha_{14} = 1 + x^3$	$\longrightarrow 1\ 1\ 1\ 0$	$\longrightarrow \alpha_{11}$
$\alpha_{15} = 0$	$\longrightarrow 1\ 1\ 1\ 1$	$\longrightarrow \alpha_{12}$

Notice that multiplication on the L. H. S. of the above table corresponds to "binary" addition on the R. H. S. The binary addition modulo (2^n - 1) may require some inqenuity from the designer: the starting point is that a classical computer binary adder, with the last "carry over connection" missing, realizes modulo 2^n addition.

VIIe. THE GROUP OF AUTOMORPHISMS. THE "AUTOMORPHISM TRANSFORMER" OR "AUTOMORPHER"

We define an automorphism of a finite field G. F. (2^n) as a 1-1 mapping φ_A onto itself, which is addition preserving and product preserving, i.e.,

$$\varphi_A (\alpha_i + \alpha_j) = \varphi_A (\alpha_i) + \varphi_A (\alpha_j)$$

$$\varphi_A (\alpha_i \cdot \alpha_j) = \varphi_A (\alpha_i) \cdot \varphi_A (\alpha_j)$$

It can be shown[7] that $\varphi(x)$ is an automorphism iff $\varphi(x) = x^{2^i}$, $i = 0, 1, 2, \ldots, n-1$.

The set of all automorphisms of G. F. (2^n) together with the cencatenation operation for mappings forms a group.

The above ideas can better be illustrated by Table 3 below, where all the automorphisms of G. F. $(2^3)/(1 + x + x^3)$ are shown.

TABLE 3: The Automorphisms of G. F. $(2^3)/(1 + x + x^3)$

α_i	$\varphi_{A_1}(x) = x^2$	$\varphi_{A_2}(x) = x^4$	$\varphi_{A_3}(x) = x^{2^3} = x$
α_7 = 0 0 0	0 0 0 = α_7	0 0 0 = α_7	0 0 0 = α_7
α_0 = 1 0 0	1 0 0 = α_0	1 0 0 = α_0	1 0 0 = α_0
α_1 = 0 1 0	0 0 1 = α_2	0 1 1 = α_4	0 1 0 = α_1
α_2 = 0 0 1	0 1 1 = α_4	0 1 0 = α_1	0 0 1 = α_2
α_3 = 1 1 0	1 0 1 = α_6	1 1 1 = α_5	1 1 0 = α_3
α_4 = 0 1 1	0 1 0 = α_1	0 0 1 = α_2	0 1 1 = α_4
α_5 = 1 1 1	1 1 0 = α_3	1 0 1 = α_6	1 1 1 = α_5
α_6 = 1 0 1	1 1 1 = α_5	1 1 0 = α_3	1 0 1 = α_6

From Table 3 one can verify, for instance, that

$$\varphi_{A_1}(\alpha_5 \cdot \alpha_6) = \varphi_{A_1}(\alpha_4) = \alpha_1$$

and

$$\varphi_{A_1}(\alpha_5) \cdot \varphi_{A_1}(\alpha_6) = \alpha_3 \cdot \alpha_5 = \alpha_1$$

Since the automorphisms of a field are given by the successive powers of x^2 and since we can easily realize circuits to square x as per the rules of G. F. $(2^n)/[c(x)]$, a circuit can easily be imagined to produce in one clock pulse all automorphisms of an element of G. F. (2^n), as shown in Fig. 10.

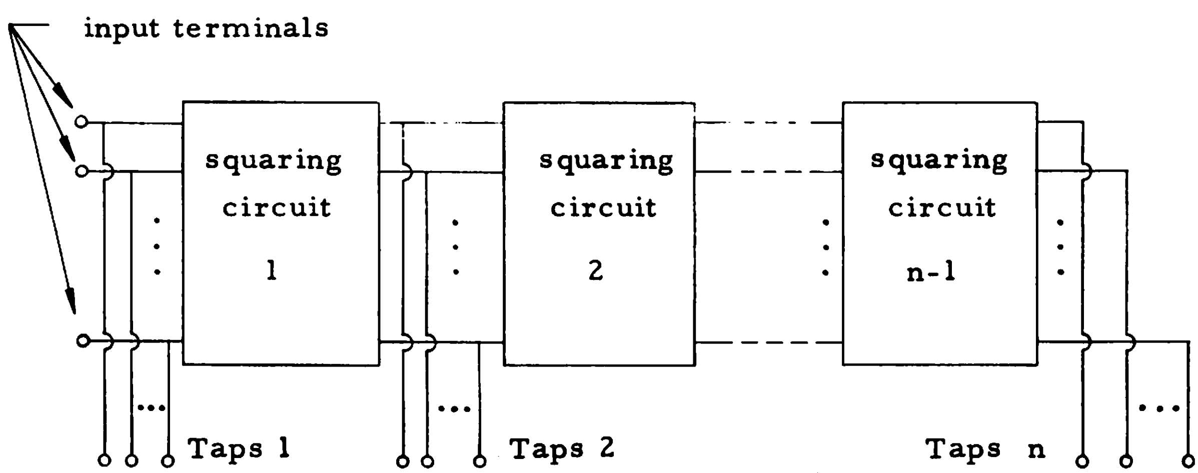


Fig. 10. The Automorpher, an electronic circuit which produces, in one clock pulse, all automorphisms of an element of a finite field G. F. (2^n).

Using the various ideas developed in this paper, we can design the "automorpher" for G. F. $(2^3)/(1 + x + x^3)$.

$$c_0 = 1 \qquad c_1 = 1 \qquad c_2 = 0 \qquad c_3 = 1$$

$$C = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = C^{-1} = I$$

$$\underline{C}^{-1}\,\underline{B}_0 = \underline{B}_0 = \text{col } (10)$$

$$\underline{C}^{-1}\,\underline{B}_1 = \underline{B}_1 = \text{col } (11)$$

$$\underline{C}^{-1}\,\underline{B}_2 = \underline{B}_2 = \text{col } (01)$$

$$D_0 = d_0 + \underline{d}'\,\underline{B}_0 = d_0 + (d_5\, d_4) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = a_0 + (0\ a_2) \begin{pmatrix} 1 \\ 0 \end{pmatrix} = a_0$$

$$D_1 = d_1 + \underline{d}'\,\underline{B}_1 = d_1 + (0\ a_2) \begin{pmatrix} 1 \\ 1 \end{pmatrix} = a_2$$

$$D_2 = d_2 + (0\ a_2) \begin{pmatrix} 0 \\ 1 \end{pmatrix} = a_1 + a_2$$

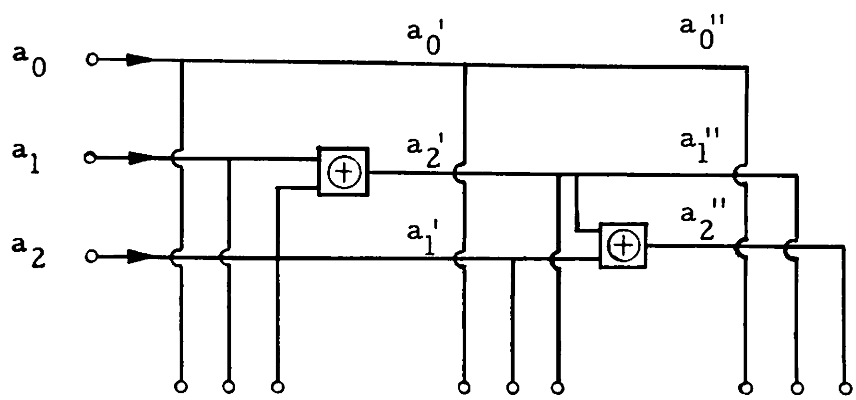


Fig. 11. The automorpher for G. F. $(2^3)/(1 + x + x^3)$.

The automorphisms of a finite field determine an equivalence relation on the set of elements of this field. The automorpher produces at its several taps the equivalent class of any element of the field.

An alternative way of generating all automorphisms of a finite field is pictured in Fig. 12. At each clock pulse a new automorphism, of the initial element in the memory, is produced.

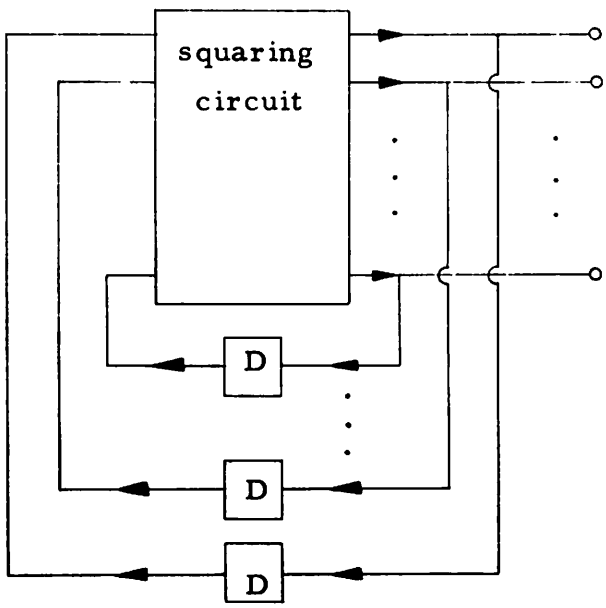


Fig. 12. The generator of automorphisms.

VIII. ISOMORPHISMS BETWEEN TWO FINITE FIELDS. THE "ISOMORPHER" AND "ISOMORPHIC GROUP GENERATOR"

Suppose two fields, F_1 and F_2, of the same cardinality, are given. A 1-1 mapping Ω from F_1 to F_2 is said to be an isomorphism iff:

$$\Omega\ (\alpha_1 \cdot \alpha_2) = \Omega\ (\alpha_1) * \Omega\ (\alpha_2)$$

$$\Omega\ (\alpha_1 + \alpha_2) = \Omega\ (\alpha_1) \dagger \Omega\ (\alpha_2)$$

where

$$F_1 = \left\{ \{\alpha_i\},\ \cdot\ ,\ + \right\}$$

$$F_2 = \left\{ \{\beta_i\},\ *,\ \dagger \right\}$$

Two finite fields of the same cardinality are isomorphic.

If we keep in mind that each field has its own rules of operation, we still can represent each of their elements by a polynomial expression.

We first will show, for two finite fields generated by primitive polynomials of same degree, how to find all isomorphisms. The main problem, here, is to find any <u>one</u> isomorphism because then the others will follow immediately, as we shall show at the end of this section.

Suppose we are given

$$F_1 = \text{G. F. } (p^n)/[c(x)]$$

and

$$F_2 = \text{G. F. } (p^n)/[d(x)]$$

where $c(x)$ and $d(x)$ are primitive irreducible polynomials, as usual;

$$c(x) = c_0 + c_1 x + \ldots + c_n x^n$$

$$d(x) = d_0 + d_1 x + \ldots + d_n x^n$$

and, for at least one i, $c_i \neq d_i$.

Calling α_i the elements of F_1, and β_i the elements of F_2, our problem is to find a mapping $\Omega(\alpha_i) = \beta_{u_i}$, $\forall i$, which is an isomorphism. It is known that the additive and multiplicative unities (zero and one) of F_1, will correspond to the additive and multiplicative unities, respectively, of F_2. We write this as:

$$\Omega(0) = \theta \quad \text{(additive unity mapping)}$$

$$\Omega(1) = i \quad \text{(multiplicative unity mapping)}$$

Now, if we knew one more correspondence $\alpha_p \rightarrow \beta_{u_p}$, we could complete the correspondence table by adding and multiplying elements of each field. To obtain this one more correspondence, we remember that a generator (call it α) of F_1 has to obey the relation

$$c(\alpha) = 0$$

or

$$c_0 + c_1\alpha + \ldots + c_n\alpha^n = 0$$

If we apply Ω to both sides of this equation

$$\Omega[c_0 + c_1 + \ldots + c_n\alpha^n] = \Omega(0) = \theta$$

and use the isomorphic properties described at the beginning of this section, we get

$$c_0 \dagger c_1 * \Omega(\alpha) \dagger \ldots \dagger c_n * [\Omega(\alpha)]^n = \theta \tag{1}$$

Calling $\Omega(\alpha) = \beta^q$, where β is a generator of F_2, our problem is now to solve

$$c_0 \dagger c_1 * \beta^q \dagger \ldots \dagger c_n * \beta^{q^n} = \theta \tag{1'}$$

for q, given that

$$d_0 \dagger d_1 * \beta \dagger \ldots \dagger d_n * \beta^n = \theta \tag{2}$$

Since we are dealing with finite fields and we know that the above problem has at least one solution (actually it has n solutions !) the best way to solve it is to start substituting in 1' all powers of β till we can find one which reduces (1') to (2).

Example: Consider the two finite fields specified by Table 4

TABLE 4

$F_1 = G. F. (2^3)/(1 + x + x^3)$	$F_2 = G. F. (2^3)/(1 + x^2 + x^3)$
$0 = 0\ 0\ 0$	$0 = 0\ 0\ 0$
$1 = \alpha^0 = 1\ 0\ 0$	$1 = \beta^0 = 1\ 0\ 0$
$x = \alpha^1 = 0\ 1\ 0$	$x = \beta^1 = 0\ 1\ 0$
$x^2 = \alpha^2 = 0\ 0\ 1$	$x^2 = \beta^2 = 0\ 0\ 1$
$1 + x = \alpha^3 = 1\ 1\ 0$	$1 + x^2 = \beta^3 = 1\ 0\ 1$
$x + x^2 = \alpha^4 = 0\ 1\ 1$	$1 + x + x^2 = \beta^4 = 1\ 1\ 1$
$1 + x + x^2 = \alpha^5 = 1\ 1\ 1$	$1 + x = \beta^5 = 1\ 1\ 0$
$1 + x^2 = \alpha^6 = 1\ 0\ 1$	$x + x^2 = \beta^6 = 0\ 1\ 1$

We know a priori that

$$0 \longrightarrow 0$$
$$\alpha^0 \longrightarrow \beta^0$$

We now want to find the isomorphic of α, $\Omega(\alpha) = \beta^q$, where

$$1 + \alpha + \alpha^3 = 0$$
$$\Omega(1 + \alpha + \alpha^3) = \Omega(0)$$

or

$$1 + \Omega(\alpha) + [\Omega(\alpha)]^3 = 0 \quad \text{(equation over } F_2 \text{ ! ! !)} \qquad (3)$$

Does β satisfy (3) ? Obviously not.

Does β^2 satisfy (3) ?

$$1 + \beta^2 + (\beta^2)^3 = 1 + \beta^2 + \beta^6 = 1 + \beta^2 + (\beta + \beta^2) = 1 + \beta \neq 0$$

The answer is no.

Does β^3 satisfy (3) ?

$$1 + \beta^3 + (\beta^3)^3 = 1 + \beta^3 + \beta^9 = 1 + \beta^3 + \beta^2 = 0$$

Yes, it does. β^3 is the correspondent of α under an isomorphism. Now we can complete the table of this isomorphism by addition or multiplication of corresponding elements of both fields. We will obtain:

$$
\begin{array}{ccc}
F_1 & \Omega & F_2 \\
& 0\,0\,0 \longrightarrow 0\,0\,0 & \\
\alpha^0 = 1\,0\,0 & \longrightarrow & 1\,0\,0 = \beta^0 \\
\alpha^1 = 0\,1\,0 & \longrightarrow & 1\,1\,0 = \beta^5 \\
\alpha^2 = 0\,0\,0 & \longrightarrow & 1\,0\,1 = \beta^3 \\
\alpha^3 = 1\,1\,0 & \longrightarrow & 0\,1\,0 = \beta \\
\alpha^4 = 0\,1\,1 & \longrightarrow & 0\,1\,1 = \beta^6 \\
\alpha^5 = 1\,1\,1 & \longrightarrow & 1\,1\,1 = \beta^4 \\
\alpha^6 = 1\,0\,1 & \longrightarrow & 0\,0\,1 = \beta^2
\end{array}
$$

In order to obtain another isomorphism between F_1 and F_2 observe that if $\varphi_A(x)$ is an automorphism of F_1 and $\Phi_1(x)$ is an isomorphism from F_1 to F_2, then $\Phi_1\ \varphi_A(x)$ is also an isomorphism from F_1 to F_2. In other words, given an isomorphism from F_1 to F_2, call it $\Phi_1(x)$, then $\Phi_i(x)$ is an isomorphism from F_1 to F_2 iff

$$\Phi_i(x) = \Phi_1\left(f_{A_i}(x)\right)$$

where $\varphi_{A_i}(x)$ is an automorphism of F_1.

We use this result to obtain the general block diagram of the isomorphic group transformer (Fig. 12), which produces all isomorphisms of a field to another. The box labelled isomorpher is the realization of a permutation-type-mapping determined by any particular isomorphism (although the two fields are different, the hardware still represents a permutation polynomial because of the convenient representation of the elements of both fields).

A possible application of the isomorphic group transformer is to match an encoder working in a certain field with a decoder working as per the rules of a different field of same cardinality. This situation may arise when the encoding procedure is straightforward in a certain field but the decoding will be realized by a standard equipment.

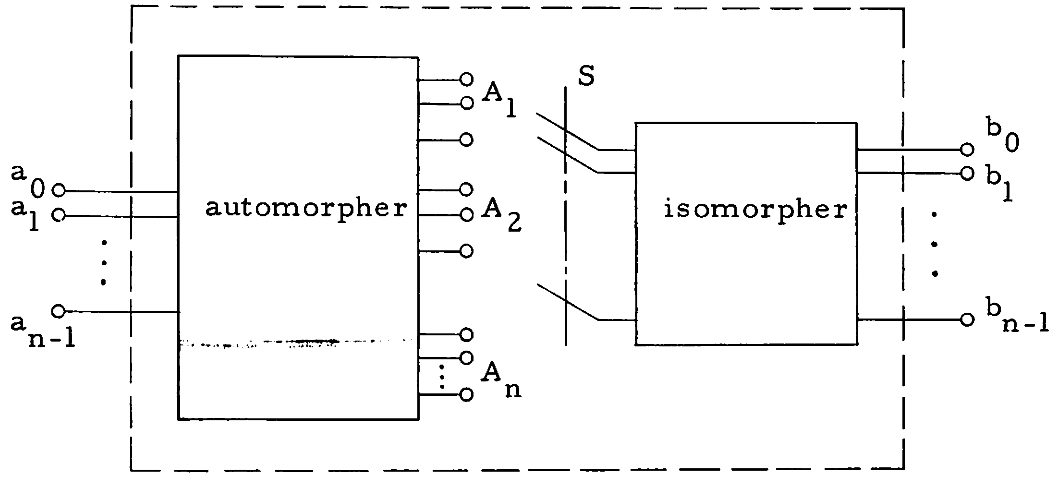


Fig. 13. Isomorhic group transformer. Each position of the switch S corresponds to a different isomorphism.

VIIg. THE SYNTHESIS OF BOOLEAN FUNCTIONS

Suppose one is given p Boolean functions $f_1, f_2, \ldots, f_p$ of q Boolean variables $x_1, x_2, \ldots, x_q$. Call $n = \max(p, q)$.

When $n = p \geq q$, one may consider the n-truple $(f_1, f_2, \ldots, f_p)$ as an element of G. F. (2^n). So is the n-truple $(x, x_2, \ldots, x_q, 0, \ldots, 0)$. For each collection of values of $(x_1, x_2, \ldots, x_q)$, say $(x_1^*, x_2^*, \ldots, x_q^*)$, there is one and only one value of $(f_1^*, f_2^*, \ldots, f_p^*)$. We shall define a mapping of G. F. (2^n) into itself by corresponding all n-truples of the form $(x_1^*, x_2^*, \ldots, x_q^*, x_{q+1}, \ldots, x_n)$ to the n-truple $(f_1^*, f_2^*, \ldots, f_n^*)$, for any values of $x_{q+1}, \ldots, x_n$.

If p is strictly greater than q this is definately a many-to-one correspondence.

When $n = q > p$, we define the mapping to correspond $(x_1^*, x_2^*, \ldots, x_q^*)$ to the n-truple $(f_1^*, f_2^*, \ldots, f_p^*, 0, \ldots, 0)$ or any other n-truple whose first p components are $f_1^*, f_2^*, \ldots, f_p^*$. The mapping is incompletely specified and the last n-p components may be used to simplify the circuitry involved, i.e., to simplify the polynomial describing the mapping.

The problem of synthesizing Boolean functions can, therefore, be thought as a problem of mapping a Galois Field of 2^n elements into itself. We have constructed, in the preceding sections, the whole mechanism which is needed for this. This method may be of help when p and q are big numbers and the Boolean expressions are not simple.

STEPS FOR THE SYNTHESIS OF BOOLEAN FUNCTIONS

Given p Boolean functions $f_1, f_2, \ldots, f_p$, of q binary variables $x_1, x_2, \ldots, x_q$:

1) Write a table listing the values of the p-truples $f_1, f_2, \ldots, f_p$ for each of the 2^q different values of the q-truple $x_1, x_2, \ldots, x_q$.
2) Define a mapping of G. F. (2^n) into, or onto if $p = q$, itself which inbeds Table 1; $n = \max(p, q)$.
3) Find the indicator functions for all elements of a finite field G. F. (2^n). Note that the indicator functions of the elements of a finite field G. F. (2^n) can be precomputed[9] and listed once and for all, like the irreducible polynomials.
4) Determine the polynomial $\varphi(x)$ which represents step 2.
5) Realize this polynomial by standard realization techniques.

REFERENCES

1. W. W. Peterson, Error-Correcting Codes, John Wiley and Sons, Inc., New York; 1961.
2. T. Bartee and D. Schneider, "An electronic decoder for Bose-Chaudhuri-Hoguenghen error-correction codes," IRE Trans., Vol. IT-8, pp. 517-524; Sept. 1962.
3. T. Booth, "Nonlinear sequential networks," IEEE Trans., p. 279; June 1963.
4. D. Huffman, "The synthesis of linear sequential coding networks," Information Theory (C. Cherry, ed.), Academic Press, Inc., New York, 1956, pp. 77-95.
5. B. Friedland and T. Stern, "The linear modular sequential circuit generalized," IRE Trans., Vol. CT-8, p. 79; Mar. 1961.
6. T. Bartee and D. Schneider, "Computation with finite fields," Information and Control, Vol. 6, No. 2, 79-98; June 1963.
7. L. E. Dickson, Linear Groups, Dover Publications, Inc., New York, N. Y.; 1958.
8. K. Menger, "A modulo 2 adder for 3 inputs using a simple tunnel diode," IRE Trans., Vol. EC-10, pp. 530-531, Sept. 1961.
9. A. Gill and J-P. Jacob, The Application of Galois Field Theory To a Class of Recognition Problems, to be published.