INFORMATION AND CONTROL 6, 79–98 (1963)

Computation with Finite Fields

THOMAS C. BARTEE AND DAVID I. SCHNEIDER

Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, Massachusetts*

A technique for systematically generating representations of finite fields is presented. Relations which must be physically realized in order to implement a parallel arithmetic unit to add, multiply, and divide elements of finite fields of 2^n elements are obtained. Finally, techniques for using a maximal length linear recurring sequence to modulate a radar transmitter and the means of extracting range information from the returned sequence are derived.

INTRODUCTION

Recent studies in the area of communications have drawn heavily upon theorems and techniques from modern abstract algebra. For instance, the properties of error-detecting and error-correcting block codes are generally proven by arguments based on theorems from linear algebra. Further, the techniques for encoding and decoding are generally stated using the language and symbology of modern algebra. As might be expected, the algorithms which have been invented sometimes cannot be easily implemented in a conventional general-purpose digital computer. For instance, a rather simple special-purpose device, the shift register with feedback, can rapidly perform encoding, decoding, or checking operations which cannot be easily performed by the conventional general-purpose computer.

Among the axiomatic systems which have received special attention are mathematical structures known as Galois or finite fields. For instance, the Bose-Chaudhuri-Hocquenghem (Bose and Ray-Chaudhuri, 1960; Hocquenghem, 1959) error-correcting code, the most efficient multiple-error-correcting block code for independent errors now known, owes its conception primarily to results from this area. Further, the decoding procedure for this code, discovered by Peterson (1961) and

* Operated with support from the U.S. Army, Navy and Air Force.

later generalized by Zierler and Gorenstein (1961) is stated in terms of the theory of finite fields. In constructing an electronic decoder for the Bose-Chaudhuri-Hocquenghem code, we found that by designing the arithmetic unit to perform not conventional binary arithmetic, but instead, Galois field operations, the decoding procedure could be implemented by a rather simple special-purpose digital machine (Bartee and Schneider, 1962). A similar requirement for Galois field arithmetic circuits has arisen in some recent applications of linear recurring sequences to space object tracking.

In this paper, we present a technique for (1) systematically generating representations of Galois field elements for a field with a given number of elements, and (2) describing, in a compact, closed form, the relations which must be physically realized in order to implement a parallel arithmetic unit which can add, subtract, multiply, and divide, using Galois field elements. Finally, techniques for using a maximal-length linear recurring sequence to modulate a radar transmitter and the means of extracting range information from the returned sequence are described. This involves determining the number of digits separating two n-tuples occurring in a given sequence, and an algorithm which is fast and readily implemented is presented.

I. REPRESENTATION SYSTEM

A knowledge of the basic properties of Galois fields is assumed. (The required material may be found in either Peterson (1961) or Albert (1956).) As is conventional, we denote the Galois field of q elements as $GF(q)$. Let $f(X) = X^n + f_1X^{n-1} + f_2X^{n-2} + \cdots + f_n$ be a primitive polynomial over $GF(q)$. That is, the coefficients f_i are elements of $GF(q)$, $f(X)$ is irreducible, and each of its roots is a generator of the multiplicative group of $GF(q^n)$.

Using the primitive polynomial $f(X)$, we can construct a maximal-length linear sequence with period q^n-1 as follows. Let s_0, s_1, s_2, $\cdots$, s_{n-1} be any sequence of n-elements from $GF(q)$ such that $s_i \neq 0$ for some $i \leqq n - 1$. Then continue the sequence by letting $s_j = -(f_1s_{j-1} + f_2s_{j-2} + \cdots + f_ns_{j-n})$ for $j = n, n + 1, \cdots$ where the f_i are the coefficients of the primitive polynomial chosen above. A sequence formed in this way will have period $q^n - 1$, and each of the q^n n-tuples which may be formed of elements of $GF(q)$, with the exception of the all zero n-tuple, will occur exactly once in each period of the sequence. (For completeness, short proofs of these two properties may be found in the

appendix. Much of what is now known in this area has been contributed by Zierler (1955, 1959), Huffman (1956a, 1956b), and Elspas (1959).) We shall call a maximal-length linear recurring sequence an M-sequence.

Now, let us designate by α^r the n-tuple of consecutive elements $(s_r, s_{r+1}, \cdots, s_{r+n-1})$ from the linear recurring sequence, where $0 \leqq r \leqq q^n - 2$. Therd will be q^n-1 such n-tuples, and by adding to this set of q^n-1 n-tuples the n-tuple, $00 \cdots 0$, which we designate as 0, we can form a Galois field of q^n elements subject to the following two rules.

1. Addition of n-tuples is as follows

$$(a_1, a_2, \cdots, a_n) + (b_1, b_2, \cdots, b_n) = (a_1 + b_1, a_2 + b_2, \cdots, a_n + b_n).$$

2. Multiplication is defined by the relation $\alpha^r \cdot \alpha^s = \alpha^{r+s(\mathrm{mod}\ q^n-1)}$ and $0 \cdot \alpha^r = \alpha^r \cdot 0 = 0 \cdot 0 = 0$.

Using these two rules, a Galois field may be quite easily generated. Two basic theorems in Galois field theory state that, (1) any given Galois field must have p^n elements, where p is a prime integer, and (2) every field of p elements is isomorphic to the set of integers added and multiplied modulo p. Therefore, in order to construct a field of p^n elements, we start with the integers modulo p and a primitive polynomial, form an M-sequence as described, and the field operations follow directly. Proof that the above rules systematically generate the field of p^n elements may be found in the appendix.

As an example, let us generate a field of nine elements. $GF(3) = \{0, 1, 2\}$ is the field of three elements with addition and multiplication mod 3. $X^2 + X + 2$ is a primitive polynomial over $GF(3)$. Let $s_0 = 1$, $s_1 = 0$, and, using the relation derived from the primitive polynomial $s_i = -(s_{i-1} + 2s_{i-2})$, we form the sequence 1, 0, 1, 2, 2, 0, 2, 1, 1, 0, 1, 2, 2, 0, $\cdots$ which has period $3^2 - 1 = 8$, and every 2-tuple except 00 occurs exactly once in each period.

Using the rule for forming the α^i described above, we find that $\alpha^0 = 10$, $\alpha^1 = 01$, $\alpha^2 = 12$, $\alpha^3 = 22$, $\cdots$, $\alpha^7 = 11$.

As examples of the addition and multiplication rules, let us find the values of the sum $\alpha + \alpha^3$ and product $\alpha^5 \cdot \alpha^4$.

$$\alpha = (0, 1), \alpha^3 = (2, 2).$$

$$\alpha + \alpha^3 = (0, 1) + (2, 2) = (0 + 2, 1 + 2) = (2, 0) = \alpha^4.$$

Hence, $\alpha + \alpha^3 = \alpha^4$.

$$(0, 2) = \alpha^5, (2, 0) = \alpha^4.$$

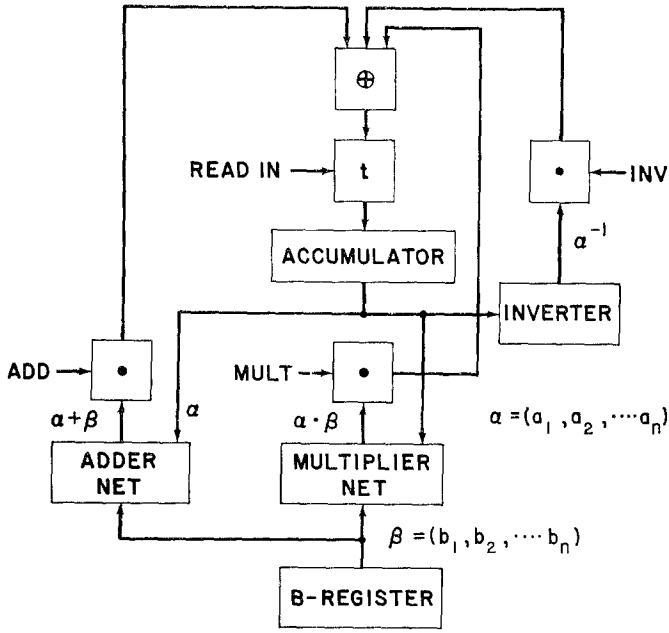

FIG. 1. Galois field arithmetic unit

$$\alpha^5 \cdot \alpha^4 = \alpha^{9(\text{mod } 8)} = \alpha^1 = (0, 1).$$

Therefore, $(0, 2) \cdot (2, 0) = (0, 1)$.

II. THE LOGICAL DESIGN OF GALOIS FIELD ARITHMETIC UNITS

Arithmetic operations over finite fields of 2^n elements are surprisingly well suited to implementation by means of digital circuitry. In fact, for fields with a reasonable number of elements, arithmetic operations can be implemented by switching circuits which will yield a sum, product, quotient, or difference in a single clock period. That is, such operations can be realized by means of combinational switching circuits[1] instead of by sequences of operations. A block diagram of such an arithmetic element is shown in Fig. 1. The operands are stored in the *B*-register and accumulator before the arithmetic operation is to be performed. The combinational networks have as outputs the sum, product, and inverse. The box with a *t* inside represents a transfer gate; when the input line

[1] A combinational switching circuit is a circuit containing only gates (no memory elements).

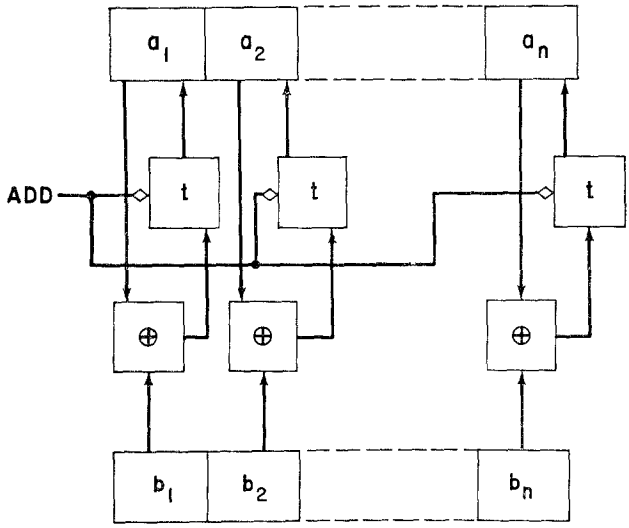

FIG. 2. Galois field adder

connected to the left of the box carries a value of 1, the other input value is gated into the memory cell to which the transfer gate is connected (McCuskey, 1962). It is well known that, given a Boolean algebra expression for a combinational network, the network can be implemented in a straightforward manner. The remaining part of this section will describe a technique for deriving the Boolean algebra expressions for networks which will physically realize Galois arithmetic operations.

In order to construct a combinational network which will add two n-tuples, we need only implement the n Boolean equations $a_1 + b_1$, $a_2 + b_2$, $\cdots$, $a_n + b_n$, for addition is componentwise, mod 2 ($0 + 0 = 1 + 1 = 0$, $0 + 1 = 1 + 0 = 1$). Such a circuit is shown in Fig. 2, where the conventional engineering symbol $\oplus$ is used to designate mod 2 adders.

Since subtraction is the same operation as addition (for $b_i = -b_i$ when b_i is an element of $GF(2)$), the adder will suffice.

A combinational network which will realize Galois field multiplication over $GF(2^n)$ will have $2n$ inputs and n outputs, and, therefore, n Boolean expressions must be derived. We now show how to derive, in a compact closed form, the n required expressions.

Let ω_i be the n-tuple $(0, 0, \cdots, 0, 1^i, 0, \cdots, 0)$. The set of vectors

$\omega_1, \omega_2, \cdots, \omega_n$ then form a basis for $GF(2^n)$, and a given element α^r of $GF(2^n)$ may be represented in the form $a_1\omega_1 + a_2\omega_2 + \cdots + a_n\omega_n$. Since each nonzero n-tuple occurs in a given M-sequence, for some $i_1 : \alpha^{i_1} = \omega_1$; and for some $i_2 : \alpha^{i_2} = \omega_2$; $\cdots$; and for some $i_n : \alpha^{i_n} = \omega_n$. We can therefore also represent a given element of $GF(2^n)$ in the form $a_1\alpha^{i_1} + a_2\alpha^{i_2} + \cdots + a_n\alpha^{i_n}$.[2]

Now, let $(a_1, a_2, \cdots, a_n)$ be the multiplier and $(b_1, b_2, \cdots, b_n)$ the multiplicand. The n components of $(c_1, c_2, \cdots, c_n) = (a_1, a_2, \cdots, a_n) \cdot (b_1, b_2, \cdots, b_n)$ can be found by forming the n expressions

$$c_k = [a_1 \cdots a_n] [M_k] \begin{bmatrix} b_1 \\ \vdots \\ b_n \end{bmatrix} \qquad k = 1, 2, \cdots, n \tag{1}$$

where the n^2 elements of matrix M_k are $[m_{pq}^k]$; $1 \leqq p, q \leqq n$ and m_{pq}^k is the kth coordinate of $\omega_p\omega_q = \alpha^{i_p}\alpha^{i_q}$.

As an example of the above technique, let us derive the expressions for a multiplier for $GF(2^3)$. Consider the primitive polynomial $X^3 + X + 1$ and let us start our M-sequence with $s_0 = 1$, $s_1 = 0$, and $s_2 = 0$. The M-sequence will then be 10010111001011 $\cdots$ and our representation system will be

$$\begin{array}{lll} 0 = 000 & \alpha^2 = 010 & \alpha^4 = 011 \\ 1 = 100 & \alpha^3 = 101 & \alpha^5 = 111 \\ \alpha = 001 & & \alpha^6 = 110 \end{array}$$

A basis will then be $\omega_1 = 1$; $\omega_2 = \alpha^2$; and $\omega_3 = \alpha$, and the expression for the first component c_1 of $(c_1, c_2, c_3) = (a_1, a_2, a_3) \cdot (b_1, b_2, b_3)$ is

$$c_1 = [a_1, a_2, a_3] \begin{bmatrix} & M_1 & \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$= [a_1, a_2, a_3] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

$$= a_1 b_1 + a_3 b_2 + a_2 b_3 .$$

[2] This procedure for deriving the multiplier expressions holds under any choice of basis where $\{\omega_1, \cdots, \omega_n\}$ is the basis and α is a generator of the multiplicative group. That is, the procedure is not limited to representation systems derived from M-sequences.

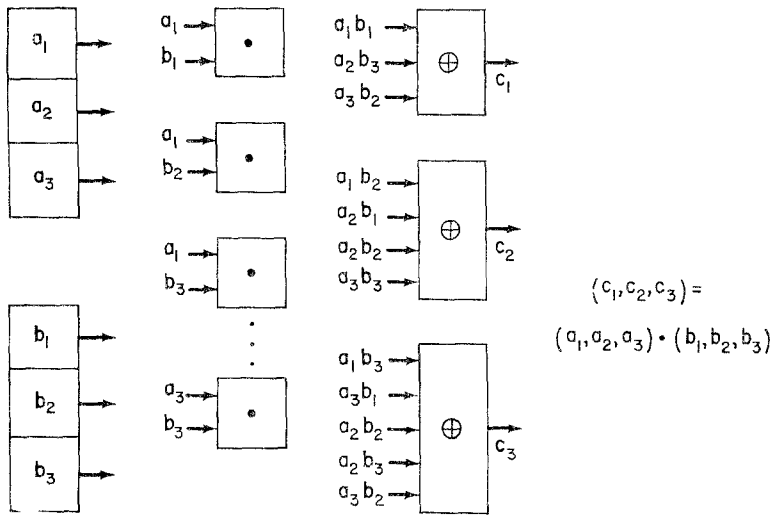

FIG. 3. Galois field multiplier for $GF(2^3)$

Using the same procedure we obtain the relations for c_2 and c_3.

$$c_2 = a_2b_1 + a_1b_2 + a_2b_2 + a_3b_3 ,$$

$$c_3 = a_3b_1 + a_2b_2 + a_3b_2 + a_1b_3 + a_2b_3 .$$

A block diagram for this particular multiplier may be found in Fig. 3. Proof that the relations derived above hold may be found in the appendix.

The arithmetic operation of division can be performed by first forming the multiplicative inverse of the divisor and then multiplying this inverse by the dividend, thus forming the quotient. If time permits—if, for instance, only a few divisions are required—then the multiplicative inverse may be formed by a sequence of multiplications. This procedure will be described first, then a technique will be presented for deriving the Boolean algebra equations for a logical circuit which directly forms the multiplicative inverse of a given element.

When the multiplicative inverse is formed using a programmed routine, $2n - 2$ multiplications are required. A property of Galois fields is that the nonzero elements of $GF(2^n)$ comprise a cyclic, multiplicative

group of order $2^n - 1$, and hence $\beta^{2^n-1} = 1$ for all β in the group.[3] Therefore, $\beta^{2^n-2} = \beta^{-1}$, for $\beta \cdot \beta^{2^n-2} = \beta^{2^n-1} = 1$.

There are several efficient ways to form the $(2^n - 2)$th power of a given element β. An efficient procedure is to form the sequence β, β^2, β^3, β^6, β^7, $\cdots$, β^{2^n-2} using the relation

$$\beta^2 \cdot \beta^4 \cdot \beta^8 \cdots \beta^{2^{n-1}} = \frac{\{[(\beta)^2 \beta]^2 \beta \cdots \beta\}^2}{n - 1 \text{ times}} = \beta^{2^n-2}.$$

Each step of the procedure requires multiplying the current value of the sequence by β and squaring the result. In all, $n - 1$ steps are required.

The multiplicative inverse may also be formed by a combinational network. We shall derive the equations of this network for the case where $\omega_1 = 1$. For the case where $\omega_1 \neq 1$, the equations can be obtained from these by performing a change of basis.

Let $\beta = (b_1, \cdots, b_n)$ be the element to be inverted. That is, we want to find $\beta^{-1} = (b_1^*, \cdots, b_n^*)$ such that $(b_1, \cdots, b_n) \cdot (b_1^*, \cdots, b_n^*) = 1$. Let A be the matrix whose i,jth entry is the jth component of $\omega_i \cdot \beta$. Then b_k^* is the minor of b_{k1}. Proof that these relations hold may be found in the appendix.

As an example, let us derive the equations for the inverting network over $GF(2^3)$ as constructed above. We shall make use of the equations derived for multiplication over $GF(2^3)$.

$$\omega_1 \cdot \beta = (1, 0, 0) \cdot (b_1, b_2, b_3) = (b_1, b_2, b_3)$$

$$\omega_2 \cdot \beta = (0, 1, 0) \cdot (b_1, b_2, b_3) = (b_3, b_1 + b_2, b_2 + b_3)$$

$$\omega_3 \cdot \beta = (0, 0, 1) \cdot (b_1, b_2, b_3) = (b_2, b_3, b_1 + b_2)$$

$$A = \begin{bmatrix} b_1 & b_2 & b_3 \\ b_3 & b_1 + b_2 & b_2 + b_3 \\ b_2 & b_3 & b_1 + b_2 \end{bmatrix}$$

$$b_1^* = (b_1 + b_2)(b_1 + b_2) + b_3(b_2 + b_3) = b_1 + b_2 + b_3 + b_2 b_3$$

$$b_2^* = b_2(b_1 + b_2) + b_3 \cdot b_3 = b_2 + b_3 + b_1 b_2$$

$$b_3^* = b_2(b_2 + b_3) + b_3(b_1 + b_2) = b_2 + b_1 b_3$$

[3] This property also holds for all fields of p^n elements, but we are interested primarily in the binary case for computer application.

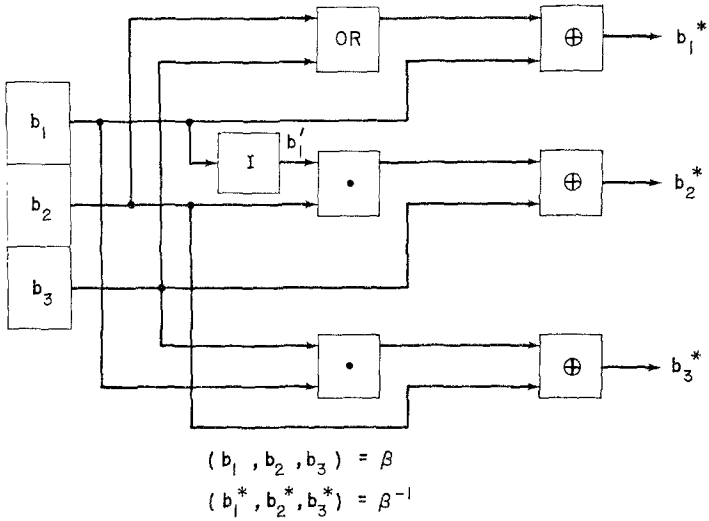

FIG. 4. Galois field inverter

i.e.,

$$b_1^* = b_1 + (b_2 \vee b_3)$$
$$b_2^* = b_3 + (b_1' \cdot b_2)$$
$$b_3^* = b_2 + b_1 b_3 .$$

See Fig. 4.

If it is desired to construct a combinational network which physically realizes division, then expressions for the inverter can be combined with the multiplication expressions yielding the required equations.

III. APPLICATION TO TRACKING RADARS

A conventional tracking radar transmits bursts of electromagnetic energy of short time duration; between these pulses the transmitter waits for a time period longer than that required for a signal to reach the target, be reflected, and then return to the receiver. If the transmitter decreases the time interval between pulses below this point, a condition known as "range ambiguity" results, making it impossible to determine the actual range of the target unless further information is available. If the range of the target to be tracked is large, the transmitter must wait for significant periods between pulses in order to avoid range ambiguity.

Further, in order to determine range precisely, it is desirable that the transmitted pulses be of short duration. As a result, the percentage of time the radar set is actually transmitting electromagnetic energy may be quite small. This results in inefficient usage of the power capabilities of the transmitter; if the transmitter could transmit during a higher percentage of the total time, more joules would be concentrated on the target and reflected, and, from the viewpoint of total energy received, a more efficient tracking system constructed. One therefore searches for a technique for encoding the transmitted signals so they may be decoded without a loss in ranging precision or range ambiguity and with an increase in the total "on time" for the transmitter.

M-sequences present an attractive method for encoding the transmitter for a tracking radar. A sequence with a very long period may be generated by a shift register with relatively few stages. Further, for a binary M-sequence of period 2^n-1, 2^{n-1} of the digits in a period will be ones and $2^{n-1} - 1$ will be zeros, since all nonzero n-tuples occur. Thus if the radar output is amplitude-modulated, the radar will be transmitting one-half of the time. Similar advantages also occur if the radar transmitter is phase- or angle-modulated and, therefore, on continuously.

When an M-sequence is used to encode the output of a radar transmitter, the problem of most efficiently using the returned signals from the target (space vehicle) may be approached in several ways. We assume, for purposes of this paper, that the returned signals are to be converted to signals representing binary digits before the steps required to determine the range are initiated. The problem is then to extract the range of the target from the sequence of binary digits which comprise the output of the radar receiver circuitry (refer to Fig. 5). Since the radar channel, even when a transponder is used, is likely to be less than perfect, we expect that some of the output digits from the receiver will be in error. We would therefore like to use any capabilities that the M-sequence might have to minimize the effects of these errors. Previous results (Zierler, 1955, 1959; Huffman 1956a, 1956b) in this area have dealt with the autocorrelation properties of M-sequences when examined through a complete period. Since the periods necessary to avoid range ambiguity are rather long, the problem of comparing a sequence through an entire period before making a decision introduces a delay which may be longer than can be afforded. That is, in many applications a short acquisition time may be a necessity, and attempting to autocorrelate, in a serial manner, many translates of the outgoing sequence against

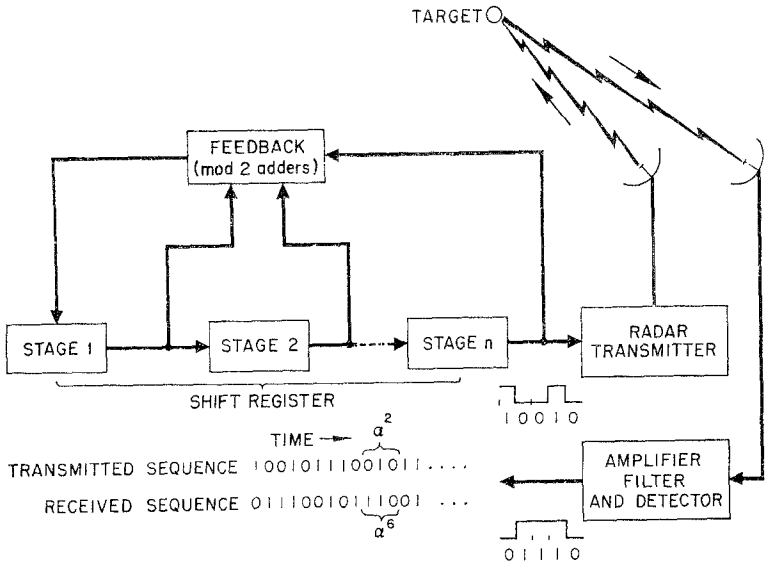

FIG. 5. Ranging using linear sequences

the returned sequence, may introduce a prohibitive delay. We have therefore chosen to consider m-tuples of the returned signal of length much less than $2^n - 1$. Given an m, the set of all m-tuples in an M-sequence plus the 0 m-tuple forms an additive group. The set of all m-tuples, where $m \geqq n$, therefore forms a linear or group code, and results from this area apply (Peterson, 1961). Recent studies (Bartee and Wood, 1963) of the codes generated in this way indicate that they are quite good: data concerning the distance properties of codes with periods of up to $2^{22} - 1$ and for m's ranging from 30 to 100 may be found in Bartee and Wood (1963) which also presents a technique for eliminating errors from a returned sequence.

Since we have placed a premium on shortening the delay time to acquisition, we are left with the problem of determining the range of the target quickly, having eliminated the errors from the incoming sequence. Now, knowing the last n-digits that have been transmitted and the last n-digits received, we have only to determine the number of digits, q, separating the two n-tuples and then to multiply q by a constant to determine the range. That is, given that the transmitter is now transmitting the first digit of the n-tuple α^t and that the receiver has just received the n digits comprising α^r, the range to the target would be

$$\frac{(pE([t - r \pmod{2^n - 1}]}{2}$$

where p is the duration of one pulse (the inverse of the p.r.f.), and E is the velocity of electromagnetic propagation (refer to Fig. 5). The problem is therefore to determine t and r (or $t - r$). Since the transmitter sequence is generated by a shift register, it is a simple process to keep track of t by simply connecting a counter which counts mod $2^n - 1$ and stepping it each time the shift register is shifted.

The problem of determining, in binary integer form, the value of the exponent r of α^r remains for the returned sequence. It would be possible to store an outgoing n-tuple and count the returning digits until the n-tuple was received, but this would involve waiting a time interval equal to that required for the signal to reach the target and return. We now present two ways to shorten the time required to determine the range.

First, notice that if δ is a fixed element of $GF(2^n)$, then multiplication by δ is a linear transformation and can be physically realized by a logical network, corresponding to a matrix, and can be designed using the equations from the preceding section. To derive the proper equations we substitute the binary values for the components of $\delta = (d_1, d_2, \cdots, d_n)$ into the b_i's for the n matrices described by the equations for the multiplier.

For instance, let us consider the previous example where the primitive polynomial was $X^3 + X + 1$ and the M-sequence was 10010111001011 $\cdots$. Let us multiply a given n-tuple $\alpha^i = (a_1, a_2, a_3)$ by $\alpha^3 = (1, 0, 1)$. The appropriate substitution into (b_1, b_2, b_3) yields

$$c_1 = [a_1, a_2, a_3]\begin{bmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}1\\ 0\\ 1\end{bmatrix} = a_1 + a_2$$

$$c_2 = [a_1, a_2, a_3]\begin{bmatrix}0 & 1 & 0\\ 1 & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1\\ 0\\ 1\end{bmatrix} = a_2 + a_3$$

$$c_3 = [a_1, a_2, a_3]\begin{bmatrix}0 & 0 & 1\\ 0 & 1 & 1\\ 1 & 1 & 0\end{bmatrix}\begin{bmatrix}1\\ 0\\ 1\end{bmatrix} = a_1 + a_2 + a_3$$

where $\alpha^i \cdot \alpha^3 = (a_1, a_2, a_3) \cdot (1, 0, 1) = (c_1, c_2, c_3)$.

A block diagram of the logical circuit is shown in Fig. 6, and the corre-

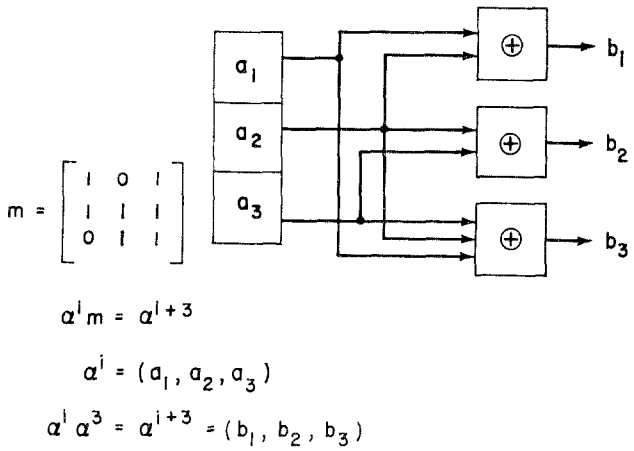

FIG. 6. Multiplier for α^3

sponding matrix is

$$[a_1, a_2, a_3]\begin{bmatrix}1 & 0 & 1\\ 1 & 1 & 1\\ 0 & 1 & 1\end{bmatrix} = [c_1, c_2, c_3]$$

Using this technique, let us assume that the minimum range to the target is known, and this range corresponds to d digits in the sequence. Then the incoming n-tuple α^r is transferred into a register, the n-tuple α^r multiplied by α^d, and the result α^{r+d} compared with the outgoing n-tuple α^t. If a binary counter is used to record the number of digits that α^{r+d} must be shifted until the n-tuple α^t is reached, then the binary number in the counter plus d equals $t - r \pmod{2^n - 1}$, the number of digits separating the n-tuples in the sequence.

A faster procedure for determining r, which requires the construction of two logical networks, is as follows.

Construct a logical network which realizes the function

$$g(\alpha^i) = 0, \qquad 0 \leqq i \leqq 2^{n-1} - 1$$
$$g(\alpha^i) = 1, \qquad 2^{n-1} \leqq i \leqq 2^n - 1.$$

The Boolean algebra expression for this function can be derived in either standard sum-of-products or product-of-sums canonical forms and then minimized. This is a standard problem and procedures for its solution

have been described by Bartee (1960) and McCluskey and Bartee (1962).[4]

Construct a logical network to realize the function $h(\alpha^i) = (\alpha^i)^2 = \alpha^{2i(\text{mod } 2^n-1)}$. Since $(\beta + c\gamma)^2 = \beta^2 + c\gamma^2$ where $\beta, \gamma \in GF(2^n)$, $c \in GF(2)$, h is a linear transformation and hence corresponds to an $n \times n$ matrix with entries in $GF(2)$ (the ith row of this matrix is ω_i^2). This matrix yields directly the equations for the logical network which realizes the function h.

Now, let the n-tuple α^r be given. The problem is to determine r in binary integer notation, i.e., $r = \sum_{i=0}^{n-1} r_i 2^i$. We determine the r_i serially, starting with r_{n-1}, then r_{n-2}, $\cdots$, and finally r_0.

$$g(\alpha^r) = r_{n-1} \quad \text{by definition of } g.$$

$$(\alpha^r)^2 = \alpha^{2r(\text{mod } 2^n-1)}.$$

$$\begin{aligned} 2r(\text{mod } 2^n - 1) &= 2(r_0 + r_1 2 + \cdots + r_{n-2}2^{n-2} + r_{n-1}2^{n-1}) \\ &= r_0 2 + r_1 2^2 + \cdots + r_{n-2}2^{n-1} + r_{n-1}2^n \\ &= r_{n-1} + r_0 2 + r_1 2^2 + \cdots + r_{n-2}2^{n-1} \end{aligned}$$

since $2^n = 1 \ (\text{mod } 2^n - 1)$.

Hence

$$g[(\alpha^r)^2] = r_{n-2}.$$

Continuing we see that $g\{[(\alpha^r)^2]^2\} = r_{n-3}$, etc. Compute r as follows:

Step 1. (a) Put α^r in register.

(b) g(register) $= r_{n-1}$.

Step i, $i = 2, \cdots, n$. (a) Put h(register) into register (that is, square the register).

(b) g(register) $= r_{n-i}$.

Using this algorithm, r is computed in $2n$ operations.

APPENDIX

A. M-Sequence

In order to present the proofs for the three properties of M-sequences given in Section I, we shall generate a sequence of elements from $GF(q)$ and show that this sequence satisfies the properties of M-sequences

[4] For large n the size of this network can be quite large, and the standard design procedure, although straightforward, can involve many calculations.

stated in Section I. Then we shall show that this sequence is actually an M-sequence.

Since q is a prime power, so is q^n, and there exists a field of q^n elements, $GF(q^n)$. $GF(q)$ is a subfield of $GF(q^n)$, and consists of all elements β of $GF(q^n)$ such that $\beta^q = \beta$.

DEFINITION. Let S be the function on $GF(q^n)$ defined by

$$S(\beta) = \sum_{i=0}^{n-1} \beta^{q^i}.$$

LEMMA 1. Let β and γ be two members of $GF(q^n)$ and let c be a member of $GF(q)$. Then:

(1) $S(\beta)$ is a member of $GF(q)$.

(2) $S(\beta + \gamma) = S(\beta) + S(\gamma)$.

(3) $S(c\beta) = cS(\beta)$.

PROOF. (1)

$$\begin{aligned}[S(\beta)]^q &= [\beta + \beta^q + \cdots + \beta^{q^{n-2}} + \beta^{q^{n-1}}]^q \\ &= (\beta)^q + (\beta^q)^q + \cdots + (\beta^{q^{n-2}})^q + (\beta^{q^{n-1}})^q \\ &= \beta^q + \beta^{q^2} + \cdots + \beta^{q^{n-1}} + \beta^{q^n} \\ &= S(\beta) \quad \text{since} \quad \beta^{q^n} = \beta.\end{aligned}$$

Therefore, $[S(\beta)]^q = S(\beta)$ implies that $S(\beta)$ is a member of $GF(q)$.

(2)

$$\begin{aligned}S(\beta + \gamma) &= \sum_{i=0}^{n-1} (\beta + \gamma)^{q^i} = \sum_{i=0}^{n-1} (\beta^{q^i} + \gamma^{q^i}) = \sum_{i=0}^{n-1} \beta^{q^i} + \sum_{i=0}^{n-1} \gamma^{q^i} \\ &= S(\beta) + S(\gamma).\end{aligned}$$

(3)

$$\begin{aligned}S(c\beta) &= \sum_{i=0}^{n-1} (c\beta)^{q^i} = \sum_{i=0}^{n-1} c^{q^i}\beta^{q^i} \\ &= \sum_{i=0}^{n-1} c\beta^{q^i} \quad \text{since} \quad c^q = c. \\ &= c \sum_{i=0}^{n-1} \beta^{q^i} = cS(\beta).\end{aligned}$$

As in Section I, let $f(x) = X^n + f_1X^{n-1} + \cdots + f_n$ be a primitive polynomial over $GF(q)$. Let α be a fixed root of $f(X) = 0$ in $GF(q^n)$.

Then each of the $q^n - 1$ nonzero elements of $GF(q^n)$ equals some power of α. From now on, $f(X)$ and α shall be fixed in our discussion.

Let $m \geqq 0$ be a positive integer and consider the sequence $S(\alpha^m)$, $S(\alpha^{m+1}), \cdots, S(\alpha^{m+i}), \cdots$. We shall show that this sequence satisfies the properties of Section I. First we show that there is a one-to-one correspondence between the n-tuples occurring in the sequence and the powers of α.

THEOREM 1. *There exists a basis* $\omega_1, \cdots, \omega_n$ of $GF(q^n)$ over $GF(q)$ such that

$$\alpha^i = S(\alpha^{m+i})\omega_1 + S(\alpha^{m+i+1})\omega_2 + \cdots + S(\alpha^{m+i+n-1})\omega_n .$$

PROOF:

$$\begin{aligned}
S(\alpha^{m+i}) &= \alpha^{m+i} + (\alpha^{m+i})^q + \cdots + (\alpha^{m+i})^{q^{n-1}} \\
S(\alpha^{m+i+1}) &= \alpha^{m+i+1} + (\alpha^{m+i+1})^q + \cdots + (\alpha^{m+i+1})^{q^{n-1}} \\
&\vdots \\
S(\alpha^{m+i+n-1}) &= \alpha^{m+i+n-1} + (\alpha^{m+i+n-1})^q + \cdots + (\alpha^{m+i+n-1})^{q^{n-1}}.
\end{aligned}$$

$$S(\alpha^{m+i}) = (\alpha^m)(\alpha^i) + (\alpha^m)^q(\alpha^i)^q + \cdots + (\alpha^m)^{q^{n-1}}(\alpha^i)^{q^{n-1}}$$

$$S(\alpha^{m+i+1}) = (\alpha^m)(\alpha)(\alpha^i) + (\alpha^m)^q(\alpha^q)(\alpha^i)^q + \cdots + (\alpha^m)^{q^{n-1}}(\alpha^{q^{n-1}})(\alpha^i)^{q^{n-1}}$$

$$\vdots$$

$$S(\alpha^{m+i+n-1}) = (\alpha^m)(\alpha)^{n-1}(\alpha^i) + (\alpha^m)^q(\alpha^q)^{n-1}(\alpha^i)^q + \cdots + (\alpha^m)^{q^{n-1}}(\alpha^{q^{n-1}})^{n-1}(\alpha^i)^{q^{n-1}}.$$

i. e.,

$$\begin{bmatrix} S(\alpha^{m+i}) \\ S(\alpha^{m+i+1}) \\ \vdots \\ S(\alpha^{m+i+n-1}) \end{bmatrix} = \begin{bmatrix} (\alpha^m)1 & (\alpha^m)^q 1 & \cdots (\alpha^m)^{q^{n-1}} 1 \\ (\alpha^m)\alpha & (\alpha^m)^q(\alpha^q) & \cdots (\alpha^m)^{q^{n-1}}(\alpha^{q^{n-1}}) \\ \vdots & \vdots & \vdots \\ (\alpha^m)(\alpha^{n-1}) & (\alpha^m)^q(\alpha^q)^{n-1} & \cdots (\alpha^m)^{q^{n-1}}(\alpha^{q^{n-1}})^{n-1} \end{bmatrix} \begin{bmatrix} (\alpha^i) \\ (\alpha^i)^q \\ \vdots \\ (\alpha^i)^{q^{n-1}} \end{bmatrix}.$$

Designate by A the above $n \times n$ matrix whose j, kth term is $a_{jk} = (\alpha^m)^{q^{k-1}}(\alpha^{q^{k-1}})^{j-1}$. Factoring out $(\alpha^m)^{q^{k-1}}$ from the kth column for $k = 1, 2, \cdots, n$ we have $|A| = \prod_{k=1}^{n} (\alpha^m)^{q^{k-1}} |A'|$ where

$$[A'] = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ (\alpha) & (\alpha^q) & \cdots & (\alpha^{q^{n-1}}) \\ & & \cdots & \\ (\alpha)^{n-1} & (\alpha^q)^{n-1} & \cdots & (\alpha^{q^{n-1}})^{n-1} \end{bmatrix}.$$

Since α has order $q^n - 1$; $\alpha, \alpha^q, \cdots, \alpha^{q^{n-1}}$ are distinct. Therefore, A' is a Van-der-Monde matrix and has nonzero determinant. Since $\alpha \neq 0$, $\prod_{k=1}^{n} (\alpha^m)^{q^{k-1}} \neq 0$ and therefore $|A| \neq 0$. Hence, A is invertible. Let

$$A^{-1} = \begin{bmatrix} \omega_1 & \omega_2 & \cdots & \omega_n \\ \vdots & \vdots & & \vdots \end{bmatrix}, \omega_1 \text{ in } GF(q^n).$$

Then

$$\begin{bmatrix} (\alpha^i) \\ (\alpha^i)^q \\ \vdots \\ (\alpha^i)^{q^{n-1}} \end{bmatrix} = \begin{bmatrix} \omega_1 & \omega_2 & \cdots & \omega_n \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \end{bmatrix} \begin{bmatrix} S(\alpha^{m+i}) \\ S(\alpha^{m+i+1}) \\ \vdots \\ S(\alpha^{m+i+n-1}) \end{bmatrix}.$$

Therefore,

$$\alpha^i = S(\alpha^{m+i})\omega_1 + S(\alpha^{m+i+1})\omega_2 + \cdots + S(\alpha^{m+i+n-1})\omega_n .$$

The space spanned linearly by $\omega_1, \cdots, \omega_n$ over $GF(q)$ includes all powers of α and also 0. Since this is all of $GF(q^n)$ and since the linear dimension of $GF(q^n)$ over $GF(q)$ is n, $\omega_1, \cdots, \omega_n$ is a basis for $GF(q^n)$ over $GF(q)$.

PROPERTIES OF $\{S(\alpha^{m+i})\}$

Property 1. The sequence $\{S(\alpha^{m+i})\}$ is periodic with periodicity $q^n - 1$.

PROOF: α is a generator of the multiplicative group of $GF(q^n)$ and hence $1, \alpha, \alpha^2, \cdots, \alpha^i, \cdots$ is a sequence with periodicity $q^n - 1$. Since each field element has a unique representation with respect to the basis $\omega_1, \cdots, \omega_n$, Theorem 1 shows that the sequence $\{S(\alpha^{m+i})\}$ also has periodicity $q^n - 1$.

Property 2. Every n-tuple occurring in the sequence is nonzero and every nonzero n-tuple occurs exactly once within each period.

PROOF: Under any basis there is a one-to-one correspondence between the nonzero elements of $GF(q^n)$ and the $q^n - 1$ nonzero n-tuples over $GF(q)$.

Property 3: Under the rules for multiplication and addition defined in Section I, the n-tuples of $\{S(\alpha^{m+i})\}$ form the field $GF(q^n)$.

PROOF: Follows directly from Theorem 1.

Now we show that any M-sequence can be derived as a sequence of the form $\{S(\alpha^{m+i})\}$. In order to form the M-sequence $\{s_i\}$, we must be given a primitive polynomial and a nonzero n-tuple over $GF(q)$. Let $f(X) = X^n + f_1X^{n-1} + \cdots + f_n$ be a fixed primitive polynomial and let $\alpha \in GF(q^n)$ be a zero of $f(X)$. The polynomial $f(X)$ shall be used to generate $\{s_i\}$ and α shall be used to generate $\{S(\alpha^{m+i})\}$. Let $s_0, s_1, \cdots, s_{n-1}$ be the nonzero n-tuple chosen to start the sequence $\{s_i\}$.

Consider the sequence $\{S(\alpha^i)\}$. Since every nonzero n-tuple over $GF(q)$ occurs in this sequence, there is an integer t such that $s_0 = S(\alpha^t)$, $s_1 = S(\alpha^{t+1}), \cdots s_{n-1} = S(\alpha^{t+n-1})$. Let $m = t$ and consider the sequence $\{S(\alpha^{m+i})\}$. The first n terms of the sequence are $s_0, s_1, \cdots, s_{n-1}$. Now we find the rest of the sequence.

Since $f(\alpha) = 0$,

$$\alpha^n + f_1\alpha^{n-1} + \cdots + f_n = 0.$$

Multiplying by α^{m+i},

$$\alpha^{m+i+n} + f_1\alpha^{m+i+n-1} + \cdots + f_n\alpha^{m+i} = 0.$$

Therefore,

$$S(\alpha^{m+i+n} + f_1\alpha^{m+i+n-1} + \cdots + f_n\alpha^{m+i}) = S(0) = 0.$$

By Lemma 1,

$$S(\alpha^{m+i+n}) + f_1S(\alpha^{m+i+n-1}) + \cdots + f_nS(\alpha^{m+i}) = 0$$

i.e.,

$$S(\alpha^{m+i+n}) = -(f_1S(\alpha^{m+i+n-1}) + \cdots + f_nS(\alpha^{m+i})).$$

But we form $\{s_i\}$ by the rule

$$s_{i+n} = -(f_1s_{i+n-1} + \cdots + f_ns_i).$$

Therefore, $S(\alpha^{m+n}) = s_n$, $S(\alpha^{m+n+1}) = s_{n+1}$, etc., i.e.,

$$\{S(\alpha^{m+i})\} = \{s_i\}.$$

Every M-sequence therefore, satisfies the properties mentioned in Section I.

B. MULTIPLICATION

Let F_i be the mapping of $GF(2^n) \times GF(2^n) \to GF(2)$ defined by $F_i[(a_1, \cdots, a_n), (b_1, \cdots, b_n)] = c_i$ where c_i is the ith component of the

product of $(a_1, \cdots, a_n)$ and $(b_1, \cdots, b_n)$. This mapping is dependent upon our choice of basis.

LEMMA. F_i is a bilinear functional, i.e., $F_i(\alpha, \beta_1 + \beta_2) = F_i(\alpha, \beta_1) + F_i(\alpha, \beta_2)$ and $F_i(\alpha_1 + \alpha_2, \beta) = F_i(\alpha_1, \beta) + F_i(\alpha_2, \beta)$.

PROOF: Since $\alpha(\beta_1 + \beta_2) = \alpha\beta_1 + \alpha\beta_2$, the ith component of $\alpha(\beta_1 + \beta_2)$ equals the ith component of $(\alpha\beta_1 + \alpha\beta_2)$. Since addition of n-tuples is componentwise, the ith component of $(\alpha\beta_1 + \alpha\beta_2)$ equals the sum of the ith components of $\alpha\beta_1$ and $\alpha\beta_2$. Hence $F_i(\alpha, \beta_1 + \beta_2) = F_i(\alpha, \beta_1) + F_i(\alpha, \beta_2)$. Similarly $F_i(\alpha_1 + \alpha_2, \beta) = F_i(\alpha_1, \beta) + F_i(\alpha_2, \beta)$. Q.E.D.

Using the fact that $F_i(\alpha, \beta)$ is a bilinear functional, we have that[5]

$$F_i(\alpha,\beta) = [a_1, \cdots, a_n]\begin{bmatrix} F_i(\omega_1, \omega_1) & F_i(\omega_1, \omega_2) & \cdots & F_i(\omega_1, \omega_n) \\ F_i(\omega_2, \omega_1) & F_i(\omega_2, \omega_2) & \cdots & F_i(\omega_2, \omega_n) \\ \vdots & \vdots & & \vdots \\ F_i(\omega_n, \omega_1) & F_i(\omega_n, \omega_2) & \cdots & F_i(\omega_n, \omega_n) \end{bmatrix}\begin{bmatrix} b_1 \\ \cdot \\ \cdot \\ \cdot \\ b_n \end{bmatrix}$$

C. INVERSION

Let $\beta = (b_1, \cdots, b_n)$. Multiplication by β is a linear transformation on $GF(2^n)$ over $GF(2)$ and, hence, can be represented by a matrix A with respect to the basis $\omega_1 = 1, \omega_2, \omega_3, \cdots, \omega_n$. The ith row of this matrix contains the n-components of $\omega_i\beta$. That is, the i, jth component of A will be the jth component of $\omega_i\beta$. The first row of that matrix contains the components of β. The matrix A is invertible, A^{-1} corresponds to multiplication by β^{-1}, and, hence, its first row contains the components of β^{-1}. Since the determinant of A is nonzero, it equals 1, and hence the formula for inverting a matrix yields the result that the 1, kth entry in A^{-1} is the minor of the k, 1st entry of A.

ACKNOWLEDGMENTS

We would like to thank W. W. Peterson for helping simplify the range determining algorithm in Section III.

RECEIVED: October 23, 1962

REFERENCES

ALBERT, A. A., "Fundamental Concepts of Higher Algebra," The University of Chicago Press, Chicago, Ill., (1956).

[5] A discussion of the bilinear functional has been made by Hoffman and Kunze (1961).

BARTEE, T. C., (1960), "Digital Computer Fundamentals," McGraw-Hill, New York.

BARTEE, T. C., AND SCHNEIDER, D. I., (1962), An electronic decoder for Bose-Chaudhuri-Hocquenghem error-correcting codes. *IRE Trans. Inform. Theory* **8,** No. 5, 17–24.

BARTEE, T. C., AND WOOD, P., (1963), Coding for tracking radar ranging. Lincoln Laboratory, MIT, report now in preparation.

BOSE, R. C., AND RAY-CHAUDHURI, C. K., (1960), On a class of error-correcting binary group codes. *Inform. and Control* **3,** 68–79.

ELSPAS, B., (1959), The theory of autonomous linear sequential networks. *IRE Trans. Circuit Theory* **6,** 45–60.

HOCQUENGHEM, A., (1959), "Codes of Correctures d'erreurs," pp. 147–156. Association Francaise de Calcul et de Traitement de L'information, Paris, France.

HOFFMAN, K., AND KUNZE, R., (1961), "Linear Algebra." Prentice-Hall, Englewood Cliffs, New Jersey.

HUFFMAN, D. A. (1956a), The synthesis of linear sequential coding networks. *In* "Information Theory," COLIN CHERRY, ed. Academic Press, New York.

HUFFMAN, D. A., (1956b), A linear circuit viewpoint on error-correcting codes. *IRE Trans. Inform. Theory* **2,** 20–28.

MCCLUSKEY, E. J., AND BARTEE, T. C., (1962), "A Survey of Switching Circuit Theory." McGraw-Hill, New York.

PETERSON, W. W., (1961), "Error Correcting Codes." The MIT Press, Cambridge, Mass.

ZIERLER, N., (1955), Several binary-sequence generators. Lincoln Laboratory, MIT, Tech. Rept. 95.

ZIERLER, N., (1959), Linear recurring sequences. *J. Soc. Ind. Appl. Math.* **7,** No. 1, 31–48.

ZIERLER, N., AND GORENSTEIN, D., (1961), A class of error-correcting codes in p^m symbols. *J. Soc. Ind. Appl. Math.***9,** No. 2, 207–214.