ERROR-CORRECTING CODES

W. Wesley Peterson

The M.I.T. Press
Massachusetts Institute of Technology
Cambridge, Massachusetts

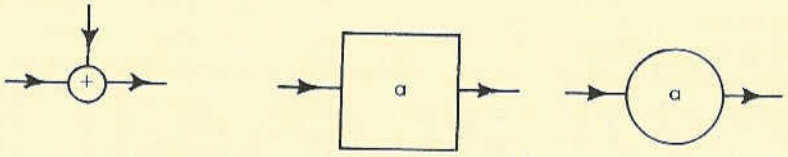
Chapter 7

LINEAR SWITCHING CIRCUITS

The heart of equipment for encoding and error correction or detection with linear codes consists of linear finite-state switching circuits. Some circuits useful for implementing linear codes are described in Sections 7.2, 7.3, and 7.4. Further properties of these circuits are presented in Section 7.5. The theory of the general linear finite-state switching circuit is introduced in Section 7.6, and it is shown that every such circuit is equivalent to a circuit of the type described in Section 7.2.

7.1. Definitions

In linear switching circuits, information is assumed to be some representation of elements of GF(q) . Three types of devices are used. The first is an adder, which has two inputs and one output, the output being the sum of the two inputs. The second is a storage device, which has one input and one output. It can be a delay device, for which the output always is the same as the input was one unit of time earlier. It can also be considered to be a single stage of shift register. In a shift register, there is a shift signal, not shown in the diagrams, which would usually be supplied by timing circuits. When this signal appears, the output of each stage takes the value that the input took immediately before the shift signal appeared. The third type of device is a constant multiplier, which has one input and one output, the output being simply the input multiplied by the constant, which may be any field element. The rule for interconnection of these devices is that any number of inputs may be connected to any output, but that two outputs are never connected together. The representation of these devices in circuit diagrams is shown in Figure 7.1.



a. Adder

b. Storage device which stores a , that is, the output is a

c. Constant multiplier for multiplying by a

Figure 7.1. The building blocks for linear switching circuits

A linear finite-state switching circuit is any circuit consisting of a finite number of adders, memory devices, and constant multipliers connected in any permissible way. Any linear finite-state switching circuit can be constructed out of vacuum tubes, transistors, magnetic cores, or other computer logical circuitry using the techniques of digital computer design. In the binary case, the adder is an "exclusive-or" logical block, and the memory device is either a delay device or a single stage of ordinary binary shift register. The constant multiplier for the constant 1 is simply a connection, and for the constant 0, simply no connection.

Input and output is assumed to be serial; that is, it consists of field elements entering an input line one at a time, one for each unit of time. When an input or output is a polynomial, as if often the case, only the coefficients appear on the input or output line, and they are transmitted high-order coefficients first. The reason is that in division the high-order coefficients of the dividend must be processed first. Thus the polynomial

$$f(X) = f_0 + f_1 X + \ldots + f_n X^n$$

would be entered on an input line or appear on an output line as a succession of n field elements, with f_n coming first, then f_{n-1} one unit of time later, f_{n-2} after another unit of time, and so forth.

7.2. Multiplication and Division of Polynomials

Circuits are given in this section for multiplication or division of any polynomial by a fixed polynomial.

The circuit shown in Figure 7.2 multiplies any input polynomial

$$a(X) = a_0 + a_1 X + \ldots + a_{k-1} X^{k-1} + a_k X^k$$

by the fixed polynomial

$$h(X) = h_0 + h_1 X + \ldots + h_{r-1} X^{r-1} + h_r X^r$$

The storage devices are assumed to contain 0's initially, and the coefficients of a(X) are assumed to enter high order first and to be followed by r 0's.

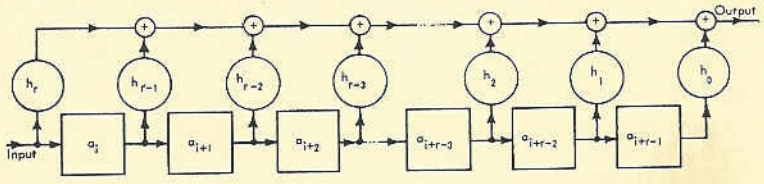


Figure 7.2. A circuit for multiplying polynomials

The product is

$$a(X)h(X) = a_0h_0 + (a_0h_1 + a_1h_0)X + (a_0h_2 + a_1h_1 + a_2h_0)X^2 + \ldots + (a_{k-2}h_r + a_{k-1}h_{r-1} + a_kh_{r-2})X^{k+r-2} + (a_{k-1}h_r + a_kh_{r-1})X^{k+r-1} + a_kh_rX^{k+r}$$

When the first coefficient a_k in $a(X)$ appears at the input, the first coefficient a_kh_r of $a(X)h(X)$ appears at the output. At that point all the storage devices contain 0's. After one unit of time, a_{k-1} appears at the input, a_k is in the first storage devices, and the rest of the storage devices contain 0's. The output can be seen from Figure 7.2 to be $a_{k-1}h_r + a_kh_{r-1}$, which is the correct second coefficient in the product $a(X)h(X)$. Similarly after two units of time a_{k-2} is at the input, and the shift register stages contain a_{k-1}, a_k, 0, ..., 0, 0, 0. The output is $a_{k-2}h_r + a_{k-1}h_{k-1} + a_kh_{r-2}$, which is the correct third coefficient of $a(X)h(X)$. The operation continues in a similar manner. After $r + k - 1$ shifts, the shift register contains 0, 0, 0, ..., 0, a_0, a_1, and the output is $a_0h_1 + a_1h_0$, which is the next-to-last coefficient in $a(X)h(X)$. After $r + k$ shifts, the shift register contains 0, 0, 0, ..., a_0, and the output is a_0h_0, the last coefficient of $a(X)h(X)$, and the product is complete.

Another circuit for multiplication is shown in Figure 7.3.

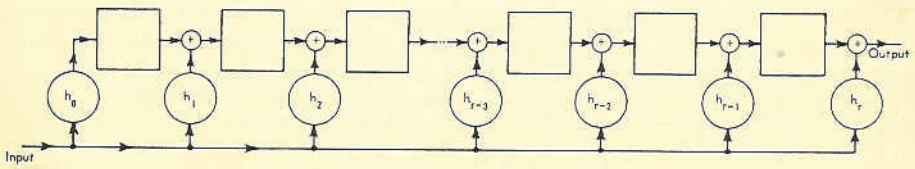


Figure 7.3. Another circuit for multiplying polynomials

The product coefficients are developed in the shift register. As the first symbol enters the input, the output is a_kh_r, and the storage devices contain all 0's. After one shift the storage devices contain a_kh_0, a_kh_1, ..., a_kh_{r-1}, and the input is a_{k-1}. The output is therefore $a_kh_{r-1} + a_{k-1}h_r$, which is the correct second coefficient. After the next shift the storage devices contain $a_{k-1}h_0$, $a_kh_0 + a_{k-1}h_1$, $a_kh_1 + a_{k-1}h_2$, ..., $a_kh_{r-2} + a_{k-1}h_{r-1}$, and the input is a_{k-2}. The output is therefore $a_kh_{r-2} + a_{k-1}h_{r-1} + a_{k-2}h_r$, which is the correct third coefficient. The operation continues in a similar manner.

This circuit can be understood in another manner. The set of r storage devices form a register that stores a polynomial. Initially it is 0. The presence of a_k at the input adds $a_k h(X)$ into the register. Shifting multiplies by X and delivers the first coefficient, whose calculation is complete, to the output. The appearance of a_{k-1} at the input adds $a_{k-1}h(X)$ into the register, and shifting again multiplies by X and delivers the second coefficient to the output, and so forth.

Circuits of the type shown in Figure 7.3 can have more than one input. For example, the circuit shown in Figure 7.4 has two

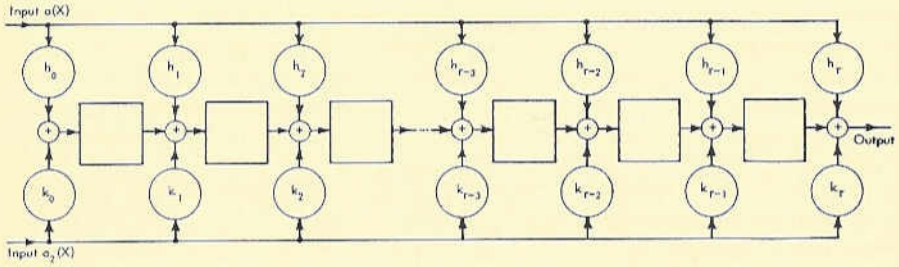


Figure 7.4. A two-input multiplier

inputs, $a_1(X)$ and $a_2(X)$, and the output is

$$b(X) = a_1(X)h(X) + a_2(X)k(X)$$

where

$$h(X) = h_0 + h_1 X + \ldots + h_r X^r$$

$$k(X) = k_0 + k_1 X + \ldots + k_r X^r$$

The circuit is shown as if h(X) and k(X) have the same degree, but in case the degrees are not equal, r can be taken as the larger degree, and the high-order coefficients of one polynomial can be 0.

Example: The circuits shown in Figure 7.5 multiply the input polynomial by $h(X) = 1 + X^3 + X^4 + X^5 + X^6$ over the field of two elements. It is instructive to write out the contents of the storage devices at each step in the process and to compare with the ordinary hand calculation of the product.

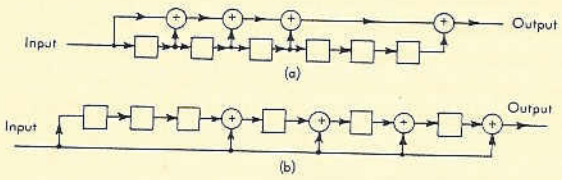


Figure 7.5. Circuits for multiplying by $1 + X^3 + X^4 + X^5 + X^6 = h(X)$

A circuit for dividing $d(X) = d_0 + d_1 X + \ldots + d_n X^n$ by $g(X) = g_0 + g_1 X + \ldots g_r X^r$ is shown in Figure 7.6. The storage devices

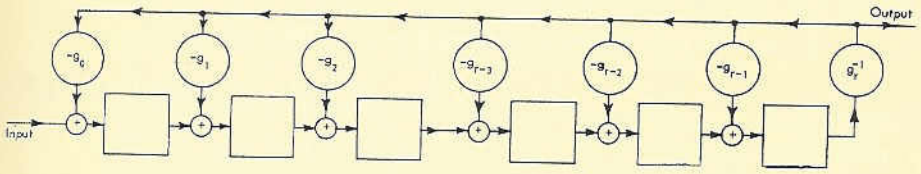


Figure 7.6. A circuit for dividing polynomials

must be set to 0 initially. The output is 0 for the first r shifts until the first input symbol reaches the end of the shift register. Then the first nonzero output appears, and it is $d_n g_r^{-1}$, the first coefficient of the quotient. For each quotient coefficient q_j, the polynomial $q_j h(X)$ must be subtracted from the dividend. The feedback connections accomplish this subtraction. After a total of n shifts, the entire quotient has appeared at the output, and the remainder is in the shift register. The operation of the circuit is best understood through a detailed example.

Example: The circuit shown in Figure 7.7 divides the input polynomial by $g(X) = 1 + X^3 + X^4 + X^5 + X^6$, over the field of

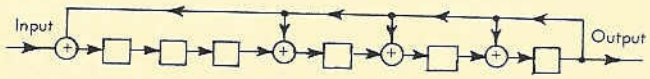


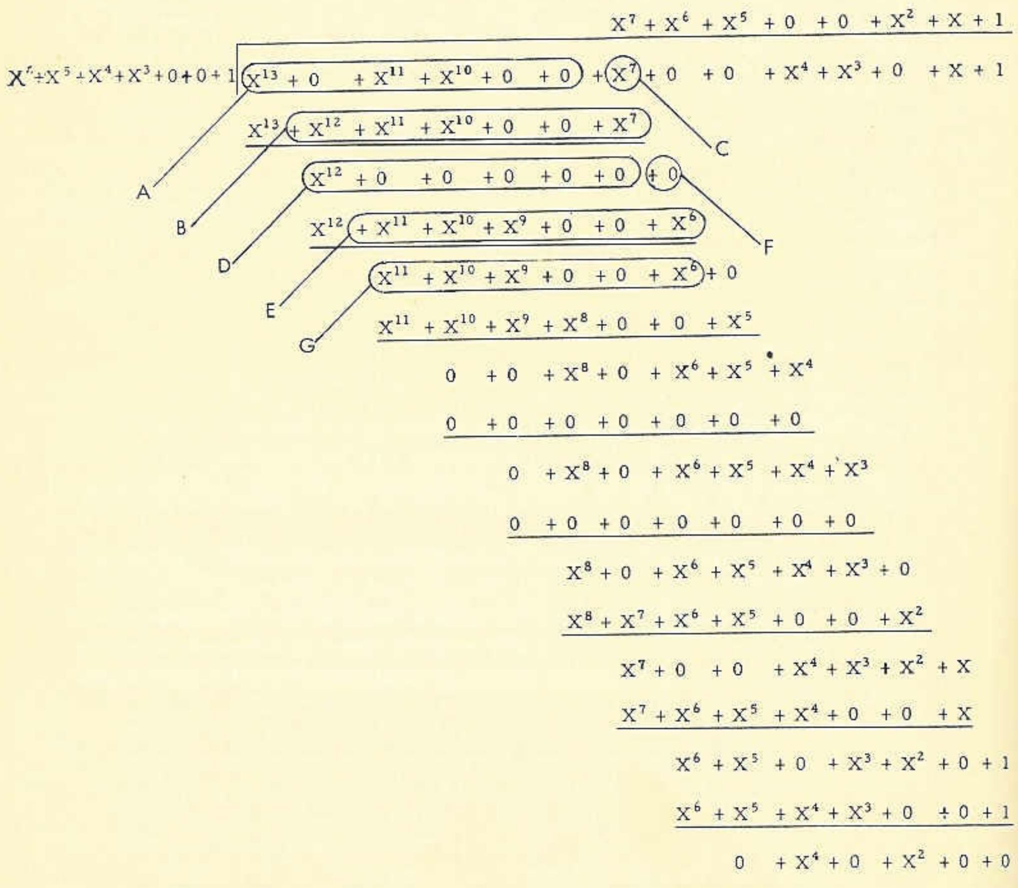
Figure 7.7. A circuit for dividing by $1 + X^3 + X^4 + X^5 + X^6$

two elements. The step-by-step division of $X^{13} + X^{11} + X^{10} + X^7 + X^4 + X^3 + X + 1$ by $1 + X^3 + X^4 + X^5 + X^6$ is compared with the step-by-step operation of this circuit in Table 7.1. Note that in ordinary long division the high-order terms are at the left, while in the shift register the high-order terms are at the right.

The first six shifts have no counterpart in long division. After six shifts the contents of the shift register match the polynomial marked A in Table 7.1a. The leading coefficient is the first quotient symbol, and is also the output after the

Table 7.1 Comparison of Long Division and the Division Circuit

(a) Long Division



seventh shift. The feedback matches the polynomial marked B, and the input corresponds to C, the quantity which is brought down. After the seventh shift the contents of the shift register match the polynomial marked D. The feedback matches E; the term brought down, F, is the same as the input; and after the eighth shift the shift register contents match G. The process continues until after fourteen shifts, one for each coefficient in the dividend, the shift register contains the remainder, and all the quotient coefficients have appeared at the output.

Table 7.1 Comparison of Long Division and the Division Circuit

(b) Step-by-Step Operation of the Division Circuit

j	Shift Register Contents after j Shifts	Output Symbol after jth Shift	Feedback on jth Shift	Input Symbol on jth Shift
0	0 0 0 0 0 0	0	–	–
1	1 0 0 0 0 0	0	0 0 0 0 0 0	1
2	0 1 0 0 0 0	0	0 0 0 0 0 0	0
3	1 0 1 0 0 0	0	0 0 0 0 0 0	1
4	1 1 0 1 0 0	0	0 0 0 0 0 0	1
5	0 1 1 0 1 0	0	0 0 0 0 0 0	0
6	0 0 1 1 0 1	1	0 0 0 0 0 0	0
7	0 0 0 0 0 1	1	1 0 0 1 1 1	1
8	1 0 0 1 1 1	1	1 0 0 1 1 1	0
9	1 1 0 1 0 0	0	1 0 0 1 1 1	0
10	1 1 1 0 1 0	0	0 0 0 0 0 0	1
11	1 1 1 1 0 1	1	0 0 0 0 0 0	1
12	1 1 1 0 0 1	1	1 0 0 1 1 1	0
13	0 1 1 0 1 1	1	1 0 0 1 1 1	1
14	0 0 1 0 1 0	0	1 0 0 1 1 1	1

A single shift-register circuit that multiplies by h(X) and then divides by g(X) can be made by combining the multiplication circuit of Figure 7.3 and the division circuit of Figure 7.6, as shown in Figure 7.8. In this circuit it is assumed that the degree h(X) is no greater than the degree of g(X) (See the example that follows.)

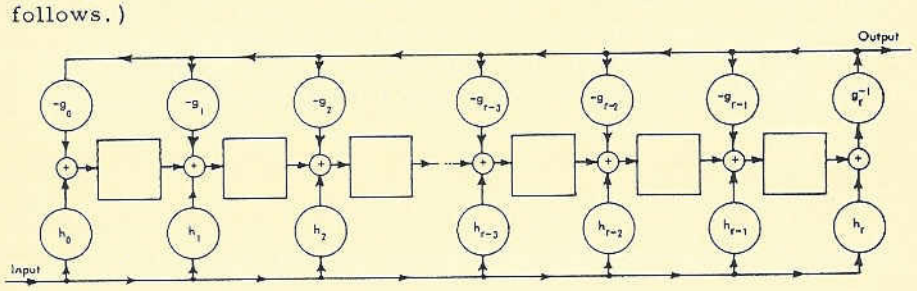


Figure 7.8. A circuit for multiplying by h(X) and dividing by g(X)

Example: A shift-register circuit for multiplying an input polynomial by $1 + X + X^5$ and then dividing by $1 + X^3 + X^4 + X^5 + X^6$ is shown in Figure 7.9. The shift register contains the part of the dividend which is being processed. The input connections add into the shift register the product of $1 + X + X^5$ and the input symbol, instead of simply adding the input symbol as in the division circuit, Figure 7.6.

If the constant factor has higher degree than the divisor, then stages should be added at the low-order end of the shift register, and as many extra shifts with 0 input as added stages are required to complete the division. An example is shown in Figure 7.10, in which the input polynomial is multiplied by $1 + X^5 + X^9 + X^{10}$ and divided $1 + X^3 + X^4 + X^5 + X^6$. In this case, four shifts with 0 input are required after the coefficient of the zero-degree term in the input, to complete the calculation of the quotient and the remainder.

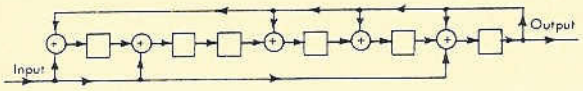


Figure 7.9. A circuit for simultaneously multiplying by $1 + X + X^5$ and dividing by $1 + X^3 + X^4 + X^5 + X^6$

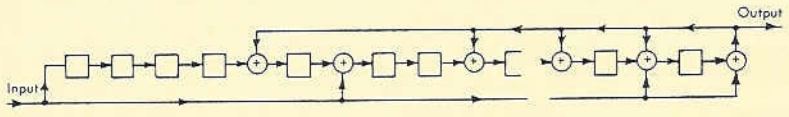


Figure 7.10. A circuit for simultaneously multiplying by $1 + X^5 + X^9 + X^{10}$ and dividing by $1 + X^3 + X^4 + X^5 + X^6$

7.3. Computations in Polynomial Algebras and Galois Fields

The circuits described in the preceding sections can be adapted for use in computations in the algebra of polynomials modulo $g(X)$, a given polynomial.

The shift register of r storage devices in Figure 7.6 stores r field elements that can be considered to be the coefficients of a polynomial

$$b(X) = b_0 + b_1 X + \ldots + b_{r-1} X^{r-1}$$

which has degree $r - 1$ or less. If the register is shifted right once, the contents become

$$b'(X) = b_0 X + b_1 X^2 + \ldots + b_{r-2} X^{r-1} - b_{r-1}(g_r^{-1} g(X) - X^r)$$

The last term is the result of the feedback connections. This can be rearranged to give

$$b'(X) = Xb(X) - b_{r-1} g_r^{-1} g(X) \qquad (7.1)$$

Thus $b'(X)$ is in the same residue class modulo $g(X)$ as $Xb(X)$, and since $b'(X)$ has degree less than r, it must be the unique polynomial in the residue class $Xb(X)$ that has degree less than r.

This can be restated as follows. If S designates the residue class containing X, then $\{b(X)\} = b(S)$ and $\{Xb(X)\} = Sb(S)$. Shifting right once thus corresponds to multiplying by S, and the contents of the shift register always are the coefficients of the unique polynomial in S of degree less than r.

The ideas will be illustrated using the polynomial $g(X) = X^4 + X + 1$ and the field of two elements. This polynomial is primitive, and therefore $\{X\} = \alpha$, which is a root of $X^4 + X + 1$, is a primitive element of $GF(2^4)$. The corresponding shift register is shown in Figure 7.11. If a 1 is placed in the low-order storage device

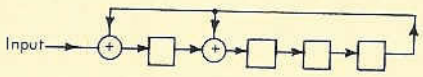


Figure 7.11. A circuit for counting in a Galois field code

and 0's in the others, successive shifts will give representations of successive powers of α, a root of $X^4 + X + 1$, in exactly the same form as they appear in Table 6.1. Note that a 1 shifted out of the high-order position corresponds to α^4 and is in effect replaced by its equal $\alpha + 1$ by the feedback connections.

A variation of this circuit is shown in Figure 7.12. A left shift corresponds to division of α and a one shifted out of the low-order

end, α^{-1} , is replaced by its equivalent $1 + \alpha^3$. Thus this device can count down, or give Galois field elements in reverse order. A multiplier can be mechanized by putting one factor in a device A like that shown in Figure 7.11, the other in a device B like that shown in Figure 7.12. Then both devices are shifted

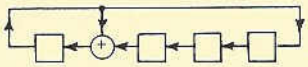
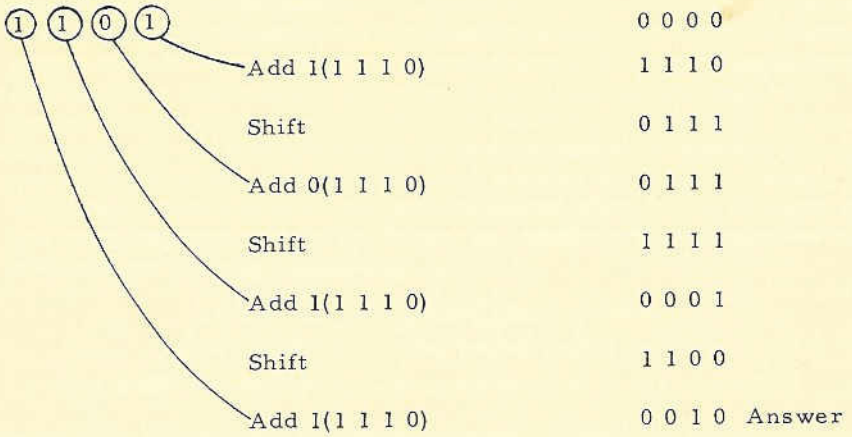


Figure 7.12. A circuit for counting backward in a Galois field code

until the code for 1 appears in device B . The product then appears in A . Division can be done in an analogous manner.

Multiplication can also be accomplished in a manner analogous to the method ordinarily used in a digital computer,* with a shift register of the type shown in Figure 7.11 used as an accumulator. This method applies in general to the algebra of polynomials modulo a polynomial g(X) and in particular to Galois fields. As an example, consider multiplying (1 1 1 0) and (1 1 0 1) as elements of $GF(2^4)$ as represented in Table 6.1. The contents of the "accumulator" are now shown after each operation. Note that vector addition is used.



The value of a polynomial r(X) when the field element α is substituted for X can be found also using the device shown in

*See, for example, Reference 98.

Figure 7.6, by taking $g(X)$ as the minimum function of α. The Galois field representation of

$$r(\alpha) = r_0 + r_1\alpha + \ldots + r_{n-1}\alpha^{n-1}$$

can be calculated by eliminating terms of degree higher than k in α by using the relation $g(\alpha) = 0$. This is exactly what will result if the vector $(r_0, r_1, \ldots, r_{n-1})$ is shifted into the device shown in Figure 7.6.

Example: Let α be a primitive element of $GF(2^4)$ as shown in Table 6.1. The Galois field representation of

$$r(\alpha) = r_0 + r_1\alpha + \ldots + r_{n-1}\alpha^{n-1}$$

can be calculated by shifting $r_{n-1}, \ldots, r_0$ into the circuit shown in Figure 7.11.

To calculate $r(\alpha^j)$ for $j \neq 1$ (or 0) is more complicated if the result must be expressed in terms of a polynomial of lowest degree in α. One method uses a shift register that automatically multiplies by α^j. The example $j = 5$ should make the principles clear. Note that, from Table 6.1,

$$1\alpha^5 = \alpha^5 = \alpha + \alpha^2$$
$$\alpha\alpha^5 = \alpha^6 = \alpha^2 + \alpha^3$$
$$\alpha^2\alpha^5 = \alpha^7 = 1 + \alpha + \alpha^3$$
$$\alpha^3\alpha^5 = \alpha^8 = 1 + \alpha^2$$

so that

$$\begin{aligned}\alpha^5(a_0 + a_1\alpha + a_2\alpha^2 + a_3\alpha^3) &= a_0(\alpha + \alpha^2) + a_1(\alpha^2 + \alpha^3) + a_2(1 + \alpha + \alpha^3) \\ &\quad + a_3(1 + \alpha^2) \\ &= (a_2 + a_3) + (a_0 + a_2)\alpha + (a_0 + a_1 + a_3)\alpha^2 \\ &\quad + (a_1 + a_2)\alpha^3\end{aligned}$$

Thus the new value of a_0 is the old $a_2 + a_3$, the new a_1 is the old $a_0 + a_2$, and so on. A shift register with feedback connection shown in Figure 7.13 will give this result. Then if the received

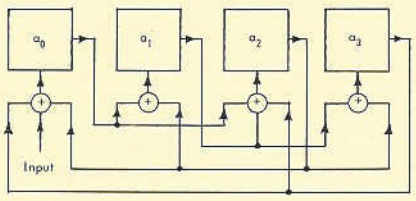


Figure 7.13. A circuit for multiplying by X^5 in $GF(2^4)$

vector $(r_0, r_1, \ldots, r_{14})$ is shifted into this device, after fifteen shifts the result $r(\alpha^5)$ will remain in the register.

7.4. Linear Recurrence Relations and Shift-Register Generators

Consider the recurrence relation, or difference equation,

$$\sum_{j=0}^{k} h_j a_{i+j} = 0 \tag{7.2a}$$

or

$$a_{i+k} = -\sum_{j=0}^{k-1} h_j a_{i+j} \tag{7.2b}$$

where $h_0 \neq 0$ and $h_k = 1$, and each h_j is an element of GF(q). A solution of these equations is a sequence $a_0, a_1, a_2, \ldots$ of elements of GF(q). Given the values of $a_0, a_1, \ldots, a_{k-1}$, Equation 7.2b is a rule for determining a_k. From knowledge $a_1, a_2, \ldots, a_k$ the value of a_{k+1} can be found, and so forth. Since the equations are linear, any linear combination of solutions is a solution, and the solutions form a vector space. The k solutions for which one of the symbols $a_0, a_1, \ldots, a_{k-1}$ is 1 and the rest are 0 span the space, and therefore the space of solutions has dimension no greater than k.

A linear sequential switching circuit that calculates the sum indicated in Equation 7.2b and hence calculates a_i from the previous k values in the sequence is shown in Figure 7.14. The initial values $a_0, a_1, \ldots, a_{k-1}$ are placed in the storage devices. Successive shifts calculate successive symbols, and the output after i shifts is always a_i. This device is called a shift-register generator.

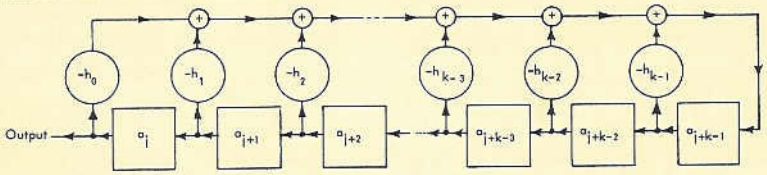


Figure 7.14. A shift-register generator

The solutions of a linear recurrence relation are characterized in the following theorem:

Theorem 7.1. Let $h(X) = \sum_{j=0}^{k} h_j X^j$, $h_0 \neq 0$, $h_k = 1$, and

let n be the smallest positive integer for which $1 - X^n$ is divisible by $h(X)$. Let $g(X) = (1 - X^n)/h(X)$. Then the solutions of the recurrence relation

$$0 = \sum_{j=0}^{k} h_j a_{i+j} \tag{7.2a}$$

are periodic of period n, and the set made up of the first period of each possible solution, considered as polynomials modulo $X^n - 1$,

$$a(X) = a_0 X^{n-1} + a_1 X^{n-2} + \ldots + a_{n-2} X + a_{n-1}$$

is the ideal generated by $g(X)$ in the algebra of polynomials modulo $X^n - 1$. Note that with $a(X)$ defined in this manner, if $a_0, a_1, \ldots$ are generated in order of increasing index, the coefficients of $a(X)$ are generated high-order first in accordance with the convention stated in Section 7.1.

Proof: First it will be shown that, if $\{a(X)\}$ is in the ideal generated by $g(X)$, then the sequence of period n

$$a_0, a_1, \ldots, a_{n-1}, a_0, a_1, \ldots \tag{7.3}$$

is a solution of Equation 7.2a. Consider the product,

$$\{a(X)\}\{h(X)\} = \{c(X)\}$$

where

$$a(X) = a_0 X^{n-1} + a_1 X^{n-2} + \ldots + a_{n-2} X + a_{n-1}$$
$$h(X) = h_0 + h_1 X + \ldots + h_k X^k$$

and

$$c(X) = c_0 + c_1 X + \ldots + c_{n-1} X^{n-1}$$

Comparison with Equation 6.8 shows that if $k \leq \ell \leq n - 1$

$$c_\ell = h_0 a_{n-1-\ell} + h_1 a_{n-\ell} + \ldots + h_k a_{n-1-\ell+k} \tag{7.4}$$

whereas if $0 \leq \ell < k$,

$$c_\ell = h_0 a_{n-1-\ell} + h_1 a_{n-\ell} + \ldots + h_\ell a_{n-1} + h_{\ell+1} a_0 + h_{\ell+2} a_1 + \ldots + h_k a_{k-\ell-1} \tag{7.5}$$

By Theorem 6.12, if $\{a(X)\}$ is in the ideal $(\{g(X)\})$, $\{a(X)\}\{h(X)\} = 0$, and therefore every $c_\ell = 0$.

Now consider the sequence given in Equation 7.3. In it, $a_i = a_{i+n}$ for all $i \geq 0$. This makes the recurrence relation, Equation 7.2a, exactly equivalent to one or the other of Equations 7.4 or 7.5; and thus if $\{a(X)\}$ is in the ideal generated by $g(X)$, the sequence in Equation 7.3 is a solution of the recurrence relation, Equation 7.2.

Since $g(X) = (X^n - 1)/h(X)$ has degree $n - k$, the ideal $(\{g(X)\})$ has dimension k, by Theorem 6.11. This is the same as the dimension of the space of solutions, and therefore by Theorem 2.9 must include all solutions.

Some of the solutions may have period less than n, but there must be some that do not. In particular, the solution obtained from $\{g(X)\}$ has period exactly n. This can be shown as follows. If it has period m less than n, then certainly n is a multiple of m, and each block of n symbols consists of n/m identical blocks of m symbols. For this to be the case, it must be that

$$g(X) = q(X)(1 + X^m + X^{2m} + \ldots + X^{n-m}) = q(X)(X^n - 1)/(X^m - 1)$$

Then

$$(X^n - 1)(X^m - 1) = h(X)g(X)(X^m - 1) = h(X)q(X)(X^n - 1)$$

and

$$(X^m - 1) = h(X)q(X)$$

which contradicts the assumption that n is the smallest integer for which $X^n - 1$ is divisible by $h(X)$.

QED

Example: Over GF(2) determine the period of solutions of the difference equation corresponding to $h(X) = X^4 + X^3 + X + 1 = (X + 1)^2(X^2 + X + 1)$. The factor $X^2 + X + 1$ divides $X^3 + 1$, and so does $X + 1$, but in order for $X^n + 1$ to be divisible by $X^2 + X + 1$ and $(X + 1)^2$, n must be taken to be 6. The period must therefore be 6. Some solutions will have period less than 6, but there must be at least one solution that has period 6. The generator of the ideal from which the solutions are formed is $g(X) = (X^6 - 1)/h(X) = X^2 + X + 1$, and this corresponds to the solution 1 1 1 0 0 0. The other solutions are all vectors in the row space of

$$G = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \qquad (7.6)$$

In particular, the sum of the first two rows, 1 0 0 1 0 0 , actually has period 3, and the sum of the first three rows, 1 0 1 0 1 0 , actually has period 2.

As a further example, the binary shift-register generator corresponding to the polynomial $h(X) = X^8 + X^6 + X^5 + X^3 + 1$ is shown in Figure 7.15. This polynomial is primitive, and therefore divides $X^{255} - 1$ but does not divide $X^n - 1$ for any smaller n . Thus the period of the shift-register output sequence is 255, which is the maximum length possible for an 8-stage shift-register generator. (See Section 8.3 for further details.)

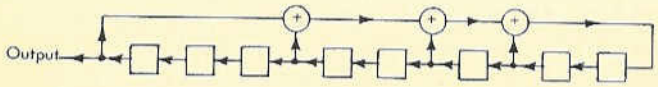


Figure 7.15. A shift-register generator for maximum-length sequences

For the purpose of studying cyclic codes, Theorem 7.1 gives the essential facts in the most convenient form. Other ways of studying linear recurrence relations will be described only briefly.

Consider the recurrence relation corresponding to $h(X)$, as described in Theorem 7.1. Let α be any root of $h(X)$, perhaps in an extension field. Then the sequence

$$1 , \alpha , \alpha^2 , \alpha^3 , \ldots \tag{7.7}$$

obviously satisfies the recurrence relation. Since the recurrence relation is linear, any linear combination of sequences of ascending powers of roots is a solution:

$$a_j = C_1 \alpha_1^{\,i} + C_2 \alpha_2^{\,i} + \ldots + C_k \alpha_k^{\,i} \tag{7.8}$$

where $\alpha_1 , \alpha_2 , \ldots , \alpha_k$ are the k roots of $h(X)$. (This assumes that all roots are distinct.) Since it is known that the space of solutions has k dimensions and since there are k arbitrary constants here, this must be a complete set of solutions. This is, of course, analogous to the classical method for solution of linear differential equations, and there is a close parallel with transform methods, with the roots of $h(X)$ playing the role of roots of the characteristic function or poles of the output function of an ordinary linear system.

Now suppose that $h_1(X)$ is an irreducible factor of degree k_1 of $h(X)$. Let α be a root of $h_1(X)$. Then over $GF(q)$, α and its powers can be written as vectors with k_1 components. Clearly each component in Sequence 7.7 must satisfy the recurrence relation. This corresponds to the fact in the study of linear

differential equations that if a complex-valued function satisfies an equation with real coefficients, both the real and imaginary parts of the complex solution are real solutions.

Example: Let α be a root of $X^3 + X^2 + 1$ over GF(2) . Then the field elements in the extension field $GF(2^3)$ can be represented by column vectors with three components from GF(2) ,

$$(1 , \alpha , \alpha^2 , \alpha^3 , \alpha^4 , \alpha^5 , \alpha^6)$$

$$\begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{7.9}$$

Each row of this matrix satisfies the recurrence relation

$$0 = a_i + a_{i+2} + a_{i+3}$$

corresponding to the polynomial $1 + X + X^3$.

Now consider the polynomial $(1 + X)(1 + X + X^3) = 1 + X^2 + X^3 + X^4$ and the corresponding recurrence relation $0 = a_i + a_{i+2} + a_{i+3} + a_{i+4}$. If β is a root of $1 + X + X^3$, then $1 , \beta , \beta^2 , \beta^3 , \beta^4 , \beta^5 , \beta^6$ satisfies the recurrence relation. Since 1 is a root of $(1 + X)$, so does

$$1 , 1 , 1 , 1 , 1 , 1 , 1$$

With use of the representation of $GF(2^3)$ given by polynomials modulo $X^3 + X + 1$, the successive powers of β give the top three rows of the matrix M , and the last row is the row of powers of 1 . Thus each row is a solution, and the set of all solutions is the row space of M:

$$M = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \tag{7.10}$$

Alternatively, consider the algebra of polynomials modulo $1 + X^2 + X^3 + X^4$, and let S denote the residue class that contains X . Then since $1 + S^2 + S^3 + S^4 = 0$,

$$1 , S , S^2 , S^3 , S^4 , S^5 , S^6$$

satisfies the recurrence relation. Note that S is not a field element. Then each element of the algebra can be represented by a vector

$$1 = \{1\} = (1\ 0\ 0\ 0)$$
$$S = \{X\} = (0\ 1\ 0\ 0)$$
$$S^2 = \{X^2\} = (0\ 0\ 1\ 0)$$
$$S^3 = \{X^3\} = (0\ 0\ 0\ 1)$$
$$S^4 = \{1 + X^2 + X^3\} = (1\ 0\ 1\ 1)$$
$$S^5 = \{1 + X + X^2\} = (1\ 1\ 1\ 0)$$
$$S^6 = \{X + X^2 + X^3\} = (0\ 1\ 1\ 1)$$

and writing $1, S, S^2, S^3, S^4, S^5, S^6$ as column vectors gives a matrix

$$\begin{bmatrix} 1\ 0\ 0\ 0\ 1\ 1\ 0 \\ 0\ 1\ 0\ 0\ 0\ 1\ 1 \\ 0\ 0\ 1\ 0\ 1\ 1\ 1 \\ 0\ 0\ 0\ 1\ 1\ 0\ 1 \end{bmatrix} \qquad (7.11)$$

Now every row of this matrix is a solution of the recurrence relation, and every solution is in the row space of this matrix.

Still another approach is to employ the concept of a "quotient field" of polynomials. The rigorous treatment is given by Zierler,[126] and only a statement of the main theorem and an example will be given here. Zierler's theorem states that if any polynomial $f(X)$ degree $k - 1$ or less is divided formally by long division by $h^*(X) = h_0 X^k + h_1 X^{k-1} + \ldots + h_k$, the coefficients of the resulting nonterminating quotient satisfy the recurrence relation

$$0 = h_0 a_i + h_1 a_{i+1} + \ldots + h_k a_{i+k}$$

Example: Let $h^*(X) = 1 + X + X^3$. Then, by long division

$$1/(1 + X + X^3) = 1 + X + X^2 + X^4 + X^7 + X^8 + X^9 + X^{11} + \ldots$$

and the sequence of coefficients, 1 1 1 0 1 0 0 1 1 1 0 1 0 0 .... agrees with each row in Equation 7.9.

The circuits described in this section and Section 7.2 are analogous to linear filters and feedback systems. They are, in fact, sampled-data systems, the only important difference from conventional systems being that the quantities are elements of a finite field here and are real numbers in conventional systems. The transform methods used for sampled-data systems apply here, and indeed the mathematics involved is that discussed in the preceding paragraphs. These ideas are pursued further in the following section.