X-520-68-133

PREPRINT

SHIFT REGISTER GENERATORS AND APPLICATIONS TO CODING

JAMES C. MORAKIS

APRIL 1968

X-520-68-133

PREPRINT

SHIFT REGISTER GENERATORS AND APPLICATIONS TO CODING

James C. Morakis

April 1968

GODDARD SPACE FLIGHT CENTER
Greenbelt, Maryland

SHIFT REGISTER GENERATORS

AND APPLICATIONS TO CODING

ABSTRACT

The most important properties of Shift Register Generated sequences are exposed. The application of Shift Registers as multiplication and division circuits leads to the generation of some error correcting and detecting codes.

CONTENTS

SHIFT REGISTER GENERATORS

AND APPLICATIONS TO CODING

INTRODUCTION

Randomlike sequences of 0's and 1's may be generated from any stage of a Shift-Register with the proper type of feedback connections; if the feedback consists only of modulo 2 adders, then the sequence is a linear sequence and the Shift-Register Generator is called a linear Shift-Register Generator. These devices are useful in communication systems and error detection and correction systems.

The contents of a Shift-Register Generator at time t are a function of the number of stages m, the location of the feedback taps, and the contents of the SRG at time $t - 1$. Thus if the m-tuplet in the SRG at time $t - 1$ is considered as the state of the SRG at $t - 1$ then the state at t is the state $(t - 1)$ operated on by some transition matrix.

SRG sequences repeat themselves. The repetition period depends on m, the number of stages, and feedback taps, but cannot exceed $2^m - 1$ bits.* When the period is equal to $2^m - 1$ bits, then the sequence is called a maximal sequence.

MAXIMAL AND NON-MAXIMAL SEQUENCES

When the sequence is maximal, considering the contents of the SRG as an m-dimensional vector, then the $2^m - 1$ successive shifts will cause the contents of the SRG to assume all possible values of a binary m-tuplet; i.e., if the $2^m - 1$ (excluding the all 0's vector) vectors of the m-dimensional space are considered as points, one can think of the maximal SRG as moving from one point to another, thus covering once all $2^m - 1$ points in one period.

In the case of a non-maximal sequence the $2^m - 1$ points are divided into 2 or more sets and the SRG will cover all the points of the set from which a point is used as the initial condition of the SRG. Using the above it can be shown that the number of zeros in a maximal sequence is one less than the number of ones; thus there are $2^{m-1} - 1$ zeros and 2^{m-1} ones.†

*This will be shown later.

† This is the balance property of 0's and 1's of a maximal sequence.

As an example, consider the two 4-stage SRG's in Figures 1 and 2, where a comparison of paths is made between a maximal and a non-maximal SRG's. The maximal sequence has a period $2^m - 1 = 2^4 - 1 = 15$ and the non-maximal sequence has periods of 6, 6, and 3. (6 + 6 + 3 = 15)

Euler Function and SRG Equivalence

Consider a maximal m-stage SRG with certain feedback tap connections. There exist different feedback tap connections for the same m that produce other maximal sequences. The contents of the SRG will again assume all possible values of the m-tuplets (except the all zero vector) but the path will be different, as shown in Figures 1-3.

The number of maximal sequences that may be produced by a linear m-stage SRG is

$$\frac{\Phi(2^m - 1)}{m}$$

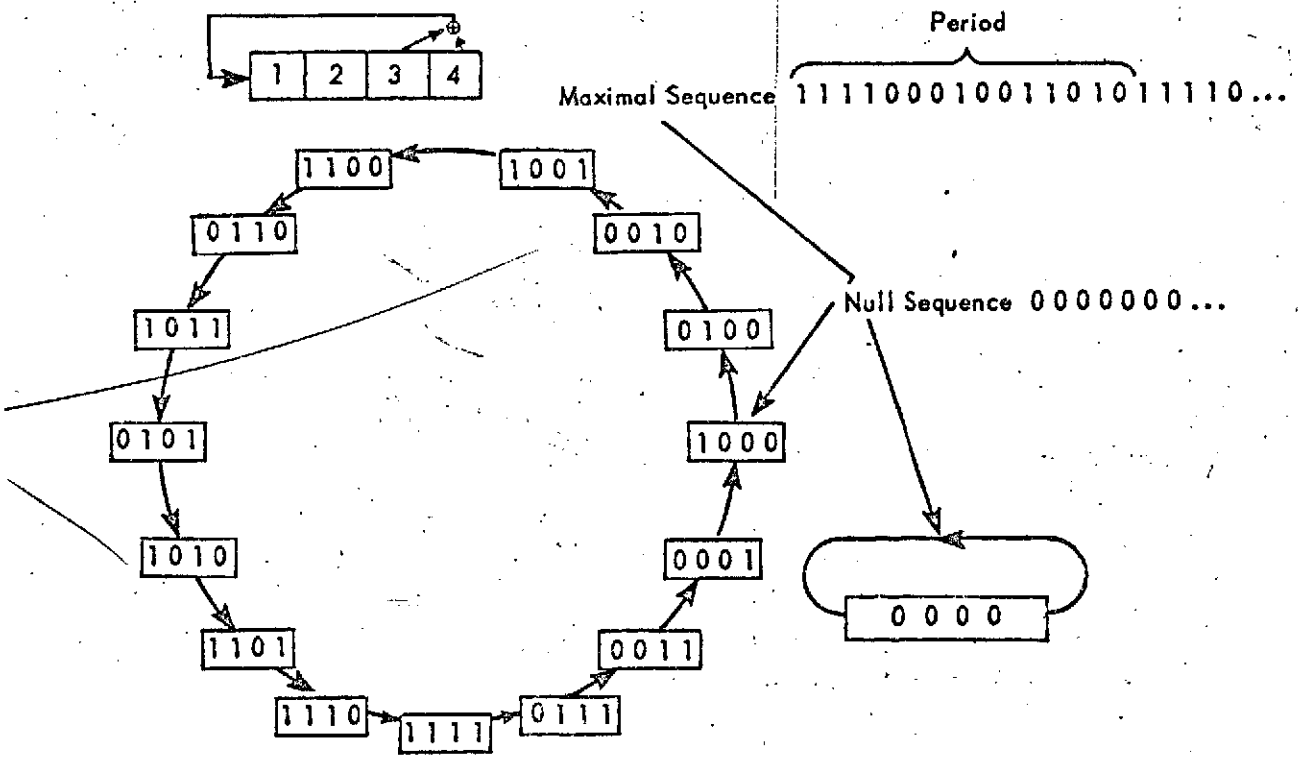


Figure 1. A Maximal SRG

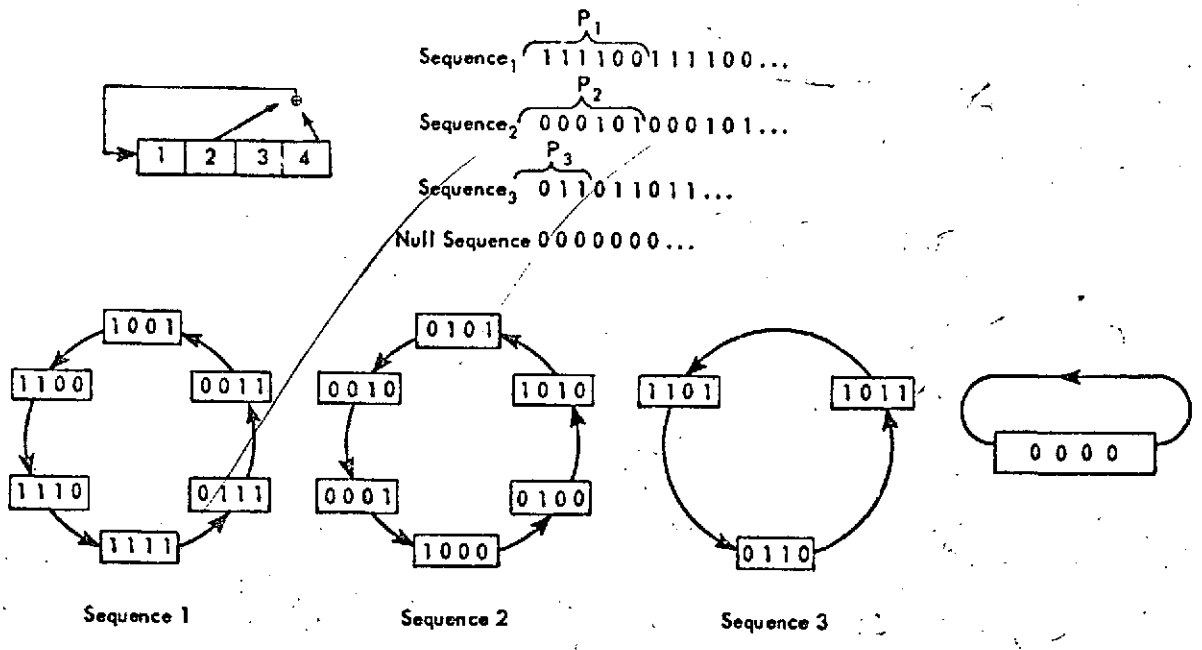


Figure 2. A Non-maximal SRG

where $\Phi(L)$ is the Euler function[1][2] and is equal to the number of integers less than and relatively prime to L.

In equation form,

$$\Phi(L) = \prod_i p_i^{e_i - 1} (p_i - 1), \qquad (1)$$

where p_i are the divisors of L, and e_i is the power to which the corresponding divisors must be raised for $L = \prod_i p_i^{e_i}$ with p being prime.

For example, for $L = 6$, $p_1 = 2$, $p_2 = 3$, $e_1 = 1$, $e_2 = 1$

$$\Phi(6) = 2^0 (2 - 1)\, 3^0 (3 - 1) = 2$$

for $L = 12$, $p_1 = 2$, $p_2 = 3$, $e_1 = 2$, $e_2 = 1$

$$\Phi(12) = 2^1 (2 - 1)\, 3^0 (3 - 1) = 4$$

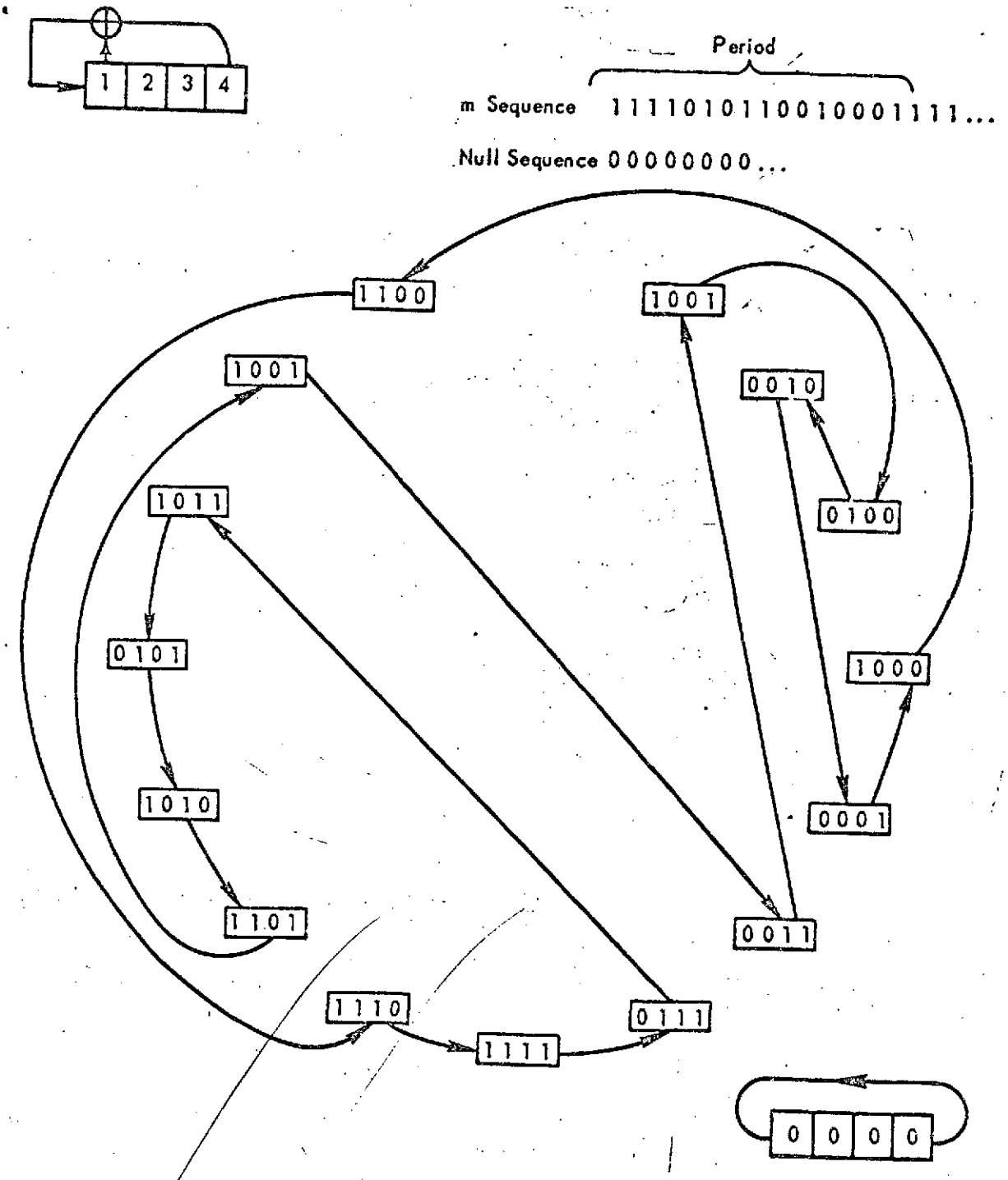


Figure 3. Another maximal sequence m = 4. (The boxes with the individual vector have been arranged identically to the ones in Figure 1 for comparison of the difference in paths.)

similarly for p a prime number

$$\Phi(p) = p^0 (p - 1) = (p - 1)$$

Using the former method the number of integers less than 6 and prime to 6 is 2, (i.e., 1 and 5).

For L = 12 we have 1, 5, 7, 11; 4 in all. For L = 31 we have 1, 2, 3, 4, ... 29, 30; 30 in all. For a 4 stage register there are

$$\frac{\Phi(2^4 - 1)}{4} = \frac{\Phi(15)}{4} = \frac{3^0 5^0 (3 - 1)(5 - 1)}{4} = 2 \text{ maximal sequences;}$$

these are given in Figures 1 and 3.

Up to this point the type of SRG's used are the ones where the feedbacks are taken from a number of stages and the m^{th} stage, added mod 2, and fed back to the first stage. This type of SRG is called a Simple Shift Register Generator (SSRG) in contrast to a Modular Shift Register Generator (MSRG) where the feedback is taken from the m^{th} stage and fed to a number of stages and the first stage through mod 2 adders.

Each SSRG has an equivalent MSRG in the sense that the sequences produced by each stage of any of the two SRG's are identical, except for a time delay. An SSRG and its equivalent MSRG's are shown in Figure 4.

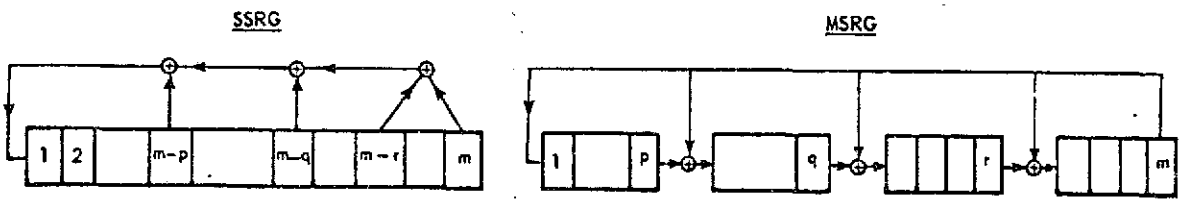


Figure 4.

In the p-nary case the maximal sequence is of length $p^m - 1$, and it contains all possible p-nary m-tuplets.

A polynomial of interest is $f_L(x)$*; this polynomial (2) is of degree $\Phi(L)$ and it contains all primitive L^{th} roots of unity; if $L = \prod_i p_i^{e_i}$ with p_i prime

$$f_L(x) = \frac{(x^L - 1) \cdot \prod_{i,j} (x^{L/p_i p_j} - 1) \cdot \prod_{i,j,k,\ell} (x^{L/p_i p_j p_k p_l} - 1) \cdots}{\prod_i (x^{L/p_i} - 1) \cdot \prod_{i,j,k} (x^{L/p_i p_j p_k} - 1) \cdots} \quad \cdots (2)$$

If $f_L(x)$ can be factored into

$$f_L(x) = \prod_{i=1}^{\Phi(L)/m} P_i(x) \mod p$$

then $P_i(x)$ are primitive polynomials and any $P_i(x)$ will generate a maximal sequence.

Equations Determining the Contents of an SRG and the Matrix that Characterizes a Given SRG

If the contents of an m-stage SRG be represented by an m-dimensional row vector, then the contents of the shift register one shift later are given by the product:

$$C(t+1)^T = A \cdot C(t)^T \quad (3)$$

where A is a transition matrix that characterizes the SRG in question.

Figure 5 demonstrates A and the associated generalized SRG. To construct A one may treat $C(t_i)$ as a state vector; the element a_{ij} of A is unity if the contents of stage j of the SRG go to state i on the next shift. In the p-nary case a_{ij} takes the value of the multiplier connected from stage j to stage i.

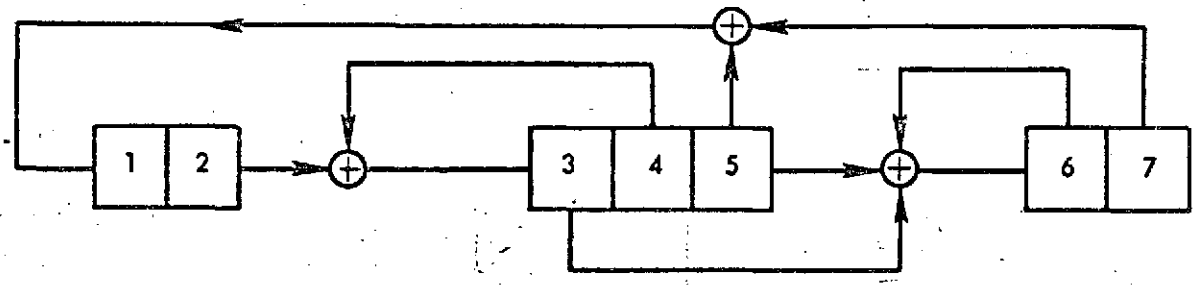
$a_{j+1,j} = 1$ for all $j = 1, 2, \ldots m$ (i.e., $a_{21} = a_{32} = a_{43} = \ldots a_{m,m-1} = 1$). This shift register and associated matrix are shown in Figure 5a.

For the shift register of Figure 5b in addition to the above $a_{34} = a_{63} = a_{66} = a_{15} = a_{17} = 1$.

*$f_L(x)$ is referred to as the cyclotomic polynomial.

$$A = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix}$$

Figure 5a.



$$A = \begin{vmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{vmatrix}$$

Figure 5b.

It was previously stated that Equation (3) gives the contents of the SRG after one shift; the contents of the SRG after two or more shifts are given by successive applications of Equation (3); thus:*

$$C(t+2)^T = A \cdot C(t+1)^T = A \cdot A \cdot C(t)^T = A^2 \cdot C(t)^T$$

Continuing in the same manner

$$C(t+\ell)^T = A^\ell \cdot C(t)^T \tag{4}$$

A in the above equations may now be considered as a shift operator whose exponent ℓ represents the number of shifts.

The matrix Equation (4) represents m equations, (one for each stage of the SRG).

If the content of stage i at time j is represented by $U_i(j)$ then at time $j+1$, (one shift later), the content of stage i is

$$U_i(j+1)^T = A^1 U_i(j)^T \tag{5}$$

where A is now considered as a shift operator and obeys the characteristic equation (to be derived later) of the SRG.

The equation

$$U_i(j+k) = U_{i-k}(j)$$

holds if there are no mod 2 adders between stages $i-k$ and i in the SRG; the above equation says that the content of stage i at $t = j+k$ is identical to whatever the content of stage $i-k$ was, k shifts before.

Equation (5) may be rewritten as

$$A^k U_i(j) = U_{i-k}(j)$$

*T stands for the transpose.

meaning that the contents of stage $i - k$ at time j will be identical to the contents of stage i, k shifts after time j.

In general, instead of speaking of the content of stage i at a specific time j, we may refer to the sequence generated by stage i starting at time t, as $U(t)$, and substitute $U_i(j)$ by $U_i(t)$ in Equation (5)

$$A^k U_i(t) = U_i(t + k) = U_{i-k}(t)$$

or

$$A^k U_i(t) = U_{i-k}(t)$$

In words the above equation says that if there are no mod 2 adders between stages $i - k$ and i the sequence produced by stage $i - k$ is identical to the sequence produced by stage i, k shifts later, or sequence U_i lags sequence U_{i-k} by k bits.

Simple Shift Register Generator (SSRG) and
Modular Shift Register Generator (MSRG) Equivalence

A generalized MSRG is of the form of Figure 6a with $\oplus$ standing for modulo p addition, $\square$ standing for a unit delay (or a shift register stage) and (h_i) for multiplication by h_i; the arrow indicates direction and in the case where $p = 2$ h_i is 0 or 1. The matrices A_{SSRG} and A_{MSRG} for the circuits of Figure 6 are:

$$A_{MSRG} = \begin{bmatrix} 0\,0\,0\,\ldots\,0\,0\,0\,h_0 \\ 1\,0\,0\,\ldots\,0\,0\,0\,h_1 \\ 0\,1\,0\,\ldots\,0\,0\,h_2 \\ 0\,0\,1\,\ldots\,0\,0\,h_3 \\ \vdots \\ h_{m-2} \\ 0\,0\,0\,\ldots\,0\,0\,0\,1\,h_{m-1} \end{bmatrix} ; \quad A_{SSRG} = \begin{bmatrix} 0\,1\,0\,0\,0\,0\,0\,\ldots\,0 & 0 \\ 0\,0\,1\,0\,0\,\ldots\,0 & 0 \\ 0\,0\,0\,1\,0\,\ldots\,0 & 0 \\ \vdots \\ 0\,0\,0\,0\,\ldots\,0\,0\,0 & 1 \\ h_0 h_1 h_2 h_3 \ldots\ldots h_{m-2} h_{m-1} \end{bmatrix}$$

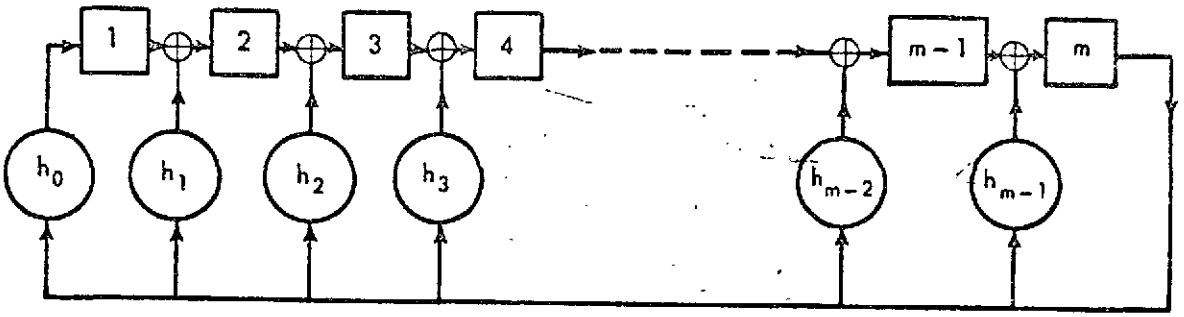


Figure 6a. An MSRG

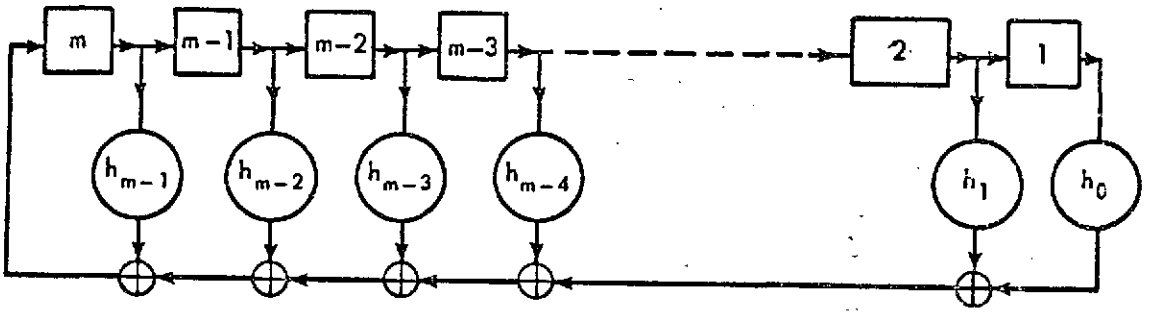


Figure 6b. An SSRG

by inspection $A_{MRG} = A_{SRG}^T$; it will be shown later that these two Shift Registers have the same characteristic equation and therefore they are equivalent.

Characteristic Equation of a Shift Register Generator

The characteristic equation $\phi(\lambda)$ is found by solving[3] the equation

$$\phi(\lambda) = \det | A - \lambda I_m | \tag{6}$$

where I_m is an identity matrix of order m.

Since* $\det | A^T - \lambda I_m | = \det | A - \lambda I_m |$ the SSRG of Figure 6a and the MSRG of Figure 6b have the same characteristic function.

*See Appendix A.

This characteristic function is evaluated* to

$$\phi(\lambda) = \sum_{i=1}^{m} (-1)^{m+i} h_{i-1} (-\lambda)^{i-1} + (-1)^m \lambda^m \quad \text{with } h_0 \neq 0 \tag{7}$$

$$= \sum_{i=0}^{m-1} (-1)^{m-1} h_i \lambda^i + (-1)^m \lambda^m = (-1)^{m-1} \left[\sum_{i=0}^{m-1} h_i \lambda^i - \lambda^m \right]$$

$$(-1)^m \phi(\lambda) = -\left[h_0 + h_1 \lambda + h_2 \lambda^2 + \ldots + h_{m-1} \lambda^{m-1} \right] + \lambda^m \tag{8}$$

a monic polynomial of degree m not divisible by λ.

Applying the Caley-Hamilton Theorem† we have

$$\phi(A) = -\left[I + h_1 A + h_2 A^2 + \ldots + h_{m-1} A^{m-1} \right] + A^m = 0 \tag{9}$$

If the operation is mod 2 addition the minus sign can be replaced by plus.

Example. Finding the Characteristic Equation of a 4-Feedback Tap SRG

Consider the 4-feedback tap MSRG of Figure 6. This is a Modular Shift Register with all h's except h_0, h_p, h_q, h_r equal to zero; application of Equation (9) results in the characteristic equation

$$A^m + A^r + A^q + A^p + I = 0$$

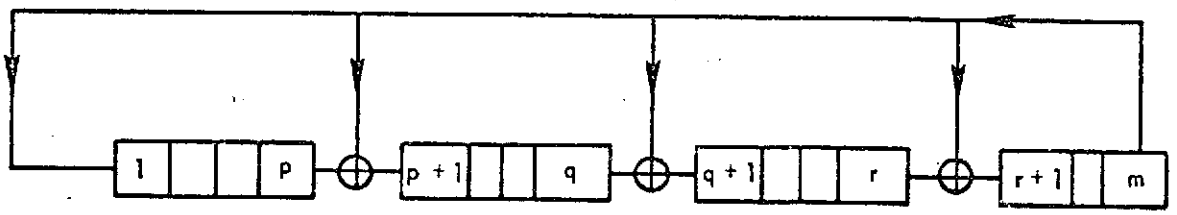


Figure 6.

*See Appendix B.

†$\phi(A) = 0$ i.e. a matrix satisfies its own characteristic equation.

Cycles of a Feedback Shift Register Sequence

Let us assume that we start with the m-tuplet C_1 in the shift register; at the next shift the new contents are C_2

$$C_2^T = AC_1^T$$

and at the i^{th} shift

$$C_{i+1}^T = A^i C_1^T \tag{10}$$

We shall prove that there exists a shift ℓ such that

$$C_{\ell+1} = A^\ell C_1 = C_1$$

which implies that

$$A^\ell = I \quad \text{for } C_1 \neq 0 \tag{11}$$

Suppose that there is no such ℓ so that

$$A^\ell C_1 = C_1$$

then $\ell = \infty$ and this means that no m-tuplet will repeat itself for an infinitely long sequence which in turn implies that the number of m-dimensional vectors is infinite; this is a contradiction because the number of m-dimensional vectors of a finite field p is $p^m - 1$ (excluding the all zero vector; consequently

$$\ell \leq p^m - 1$$

the maximum value of ℓ (designated by $L = p^m - 1$) is obtained only when the roots of $\phi(x)$ are primitive* i.e., the smallest ℓ for which $x^\ell = 1$ is $\ell = p^m - 1$.

*A primitive polynomial in x is one which cannot be factored (irreducible) and which will divide $x^\ell - 1$ for no ℓ less than $p^m - 1$.[4][11] Also see Appendix C.

The number of irreducible (but non-primitive) polynomials of degree 4 is $\phi(\ell_1)/m$

$$5 = p_1^{e_1} = 5^1$$

$$\mu_1 = \frac{\Phi(\ell_1)}{m} = \frac{\Phi(5)}{4} = \frac{5^0(5-1)}{4} = 1$$

Example:

Find the number and lengths of sequences for $m = 6$, $p = 2$; $2^m - 1 = 63$; the factors are $63 = 3^2.7$. Try 3; 3 divides $2^2 - 1$ and $\ell_1 \neq 3$. Try 7; 7 divides $2^3 - 1$; therefore $\ell_1 \neq 7$. Try $3 \times 7 = 21$, it is relatively pure with $2^5 - 1 = 31$, therefore $\ell_1 = 21$; there are 3 subsequences - (63/21)

$$\frac{\Phi(21)}{6} = \frac{3^0(2)\ 7^0(6)}{6} = 2$$

Therefore there are 2 irreducible but not primitive polynomials of degree 6.

When $\phi(x)$ is factorable with non-repeating irreducible polynomials as factors (this implies that $\phi_i(x)$ are relatively prime) we may express $\phi(x)$ as

$$\phi(x) = \phi_1(x)\,\phi_2(x)\,\phi_3(x) \ldots \phi_r(x)$$

The number of sequences, μ_i, and corresponding lengths, ℓ_i, for each of the polynomials, ϕ_i, can be expressed as $[1, \mu_i(\ell_i)]$ and it signifies that $\phi_i(x)$ has one sequence of length 1 (the all zero sequence) and μ_i subsequences of equal length ℓ_i. The reason for the name subsequences is because the space spanned by the ℓ_i vectors of one of the μ_i sequences is disjoint from the space spanned by any other of the μ_i sequences, such that the sum of all $\ell_i \times \mu_i$ vectors due to all μ_i sequences will span the whole spaces of m_i-tuplets (except for the all zero vector).

Thus

$$\ell_i \mu_i = p^{m_i} - 1$$

where m_i is the degree of $\phi_i(x)$; the expression $[1, \mu_i(\ell_i)]$ is called the cycle structure, thus if

$$\phi(x) = \phi_1(x)\,\phi_2(x)$$

and $\phi_1(x)$, $\phi_2(x)$ are irreducible with cycle structures $[1, \mu_1(\ell_1)]$ and $[1, \mu_2(\ell_2)]$ respectively, then the cycle structure of $\phi(x)$ is

$$\left[1 + \mu_1(\ell_1)\right]\ \left[1 + \mu_2(\ell_2)\right] = 1 + \mu_1(\ell_1) + \mu_2(\ell_2) + \mu(\ell) \qquad (12)$$

where*

$$\mu = \mu_1\mu_2\ \gcd(\ell_1, \ell_2)$$

and

$$\ell = \ell\,\mathrm{cm}(\ell_1, \ell_2)$$

then we have 1 sequence of length 1 (0' vector) μ_1 sequences of length ℓ_1, μ_2 sequences of length ℓ_2 and μ sequences of length ℓ.

The sum of the lengths of all sequences is

$$\ell_s = 1 + \mu_1\ell_1 + \mu_2\ell_2 + \mu_1\mu_2\ \gcd(\ell_1, \ell_2)\ \ell\,\mathrm{cm}(\ell_1, \ell_2)$$

but

$$\gcd(\ell_1, \ell_2)\,\ell\,\mathrm{cm}(\ell_1, \ell_2) = \ell_1\ell_2$$

Thus

$$\ell_s = 1 + \mu_1\ell_1 + \mu_2\ell_2 + \mu_1\mu_2\ell_1\ell_2 = (1 + \mu_1\ell_1)\ (1 + \mu_2\ell_2)$$

$$= 2^{m_1}\ 2^{m_2} = 2^{m_1+m_2}$$

*gcd ⟶ greatest common dividor
ℓcm ⟶ lowest common multiple

which is the length of the sequences of an $m_1 + m_2$ - stage shift register.

The above sequences can be generated by the circuit of Figure 7 with the proper initial conditions (contents).

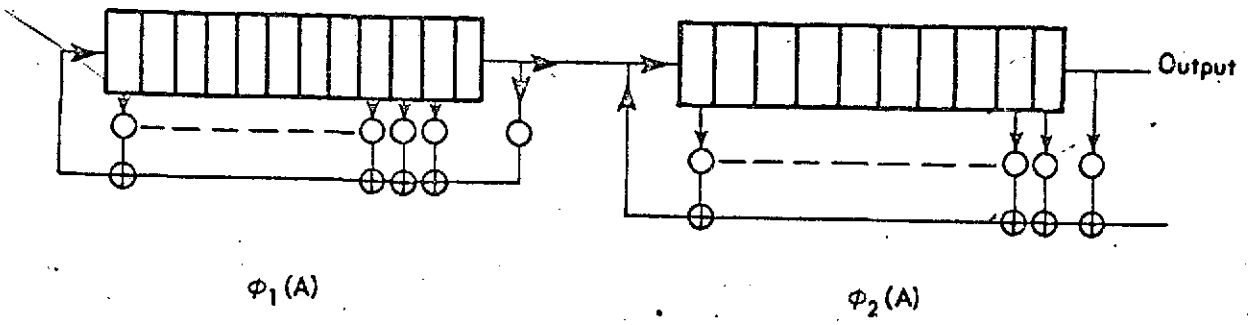


Figure 7. Implementation of a Sequential Circuit Whose $\phi(x)$ is the Product of $\phi_1(x)$ and $\phi_2(x)$

For example, let $\phi_1(\lambda) = \lambda^4 + \lambda^3 + \lambda^2 + \lambda + 1$ with $\ell_1 = 5$ and $\mu_1 = 3$ and $\phi_2(\lambda) = \lambda^3 + \lambda + 1$ with $\ell_2 = 7$ and $\mu_2 = 1$. The cycle structures are $[1, 3(5)]$ and $[1, 1(7)]$, the resultance cycle structure is

$$[1, 3(5),\ 1(7),\ 3 \cdot 1(35)] \ = \ [1, 1(7),\ 3(5),\ 3(35)]$$

thus, besides the all zero sequence there is one sequence in the length 7, three with length 5 and three with length 35 so that

$$1 + 7 + 3 \times 5 + 3 \times 35 \ = \ 128 \ = \ 2^{4+3}$$

When there are repeated factors the results are too complex to cite here.

The Shift and Add Property of Maximal Sequences

The interpretation of A as a shift operator becomes obvious because every time A operates on a sequence the sequence shift by one position, hence the exponent of A represents the number of shifts applied to the sequence or the number of positions by which the contents of the Shift Register are advanced. Since A is an advance operator A^{-1} is the delay operator and it will be denoted by D where $D = A^{-1}$.

Shift and Add Property

If a maximal sequence is first advanced by an amount τ_1 digits and then added to itself mod p, the resultant sequence will be the same sequence with a unique advance τ_2 from the original sequence, which is a function of τ_1 and $\phi(x)$

$$U_i(t) + U_i(t + \tau_1) = U_i(t + \tau_2) \tag{13}$$

or

$$U_i(t) + U_i(t) A^{\tau_1} = U_i(t) A^{\tau_2} \tag{14}$$

dividing the last equation by $U_i(t)$

$$I + A^{\tau_1} = A^{\tau_2} \tag{15}$$

if τ_1 is known, τ_2 can always be found although for most of the cases the procedure may prove laborious and long.

If a given sequence is advanced more than once and if all the advanced sequences are added to the original one using mod p addition, the resultant sequence will be identical to the original one and will have a unique advance from the original sequence, i.e.,

$$U_i(t + \tau_1) + U_2(t + \tau_2) + U_i(t + \tau_3) + \ldots U_i(t + \tau) \tag{16}$$

or

$$U_i(t) \sum_{\ell=0}^{p^m - 1} a_\ell A^{\tau \ell} = U_i(t) A^{\tau} \quad \text{where } a_\ell \text{ is } 0, 1, \ldots p - 1 \tag{17}$$

The above equation may be easily proven if 2 terms of the summation are added at a time and replaced by a single term using Equation (14).

If Equation (17) is divided by $U_i(t)$ we have:

$$\sum_{\ell=0}^{p^m-1} a_\ell \, A^{\tau\ell} = A^\tau \qquad (18)$$

Equation (18) expresses a linear combination of all possible $p^m - 1$ powers of A in terms of a single power, τ, of A if the arithmetic on the exponents τ is mod $p^m - 1$. (For proofs see Appendix C.)

Example 1:

Consider the MSRG $A^3 + A + I = 0$ in Figure 8a with an equivalent (in this case identical) SSRG in Figure 8b. From 8b it can be seen that stage 1 lags stage 2 by 2 shifts, that stage 1 lags stage 3 by 1 shift and stage 3 lags stage 2 by 1 shift.

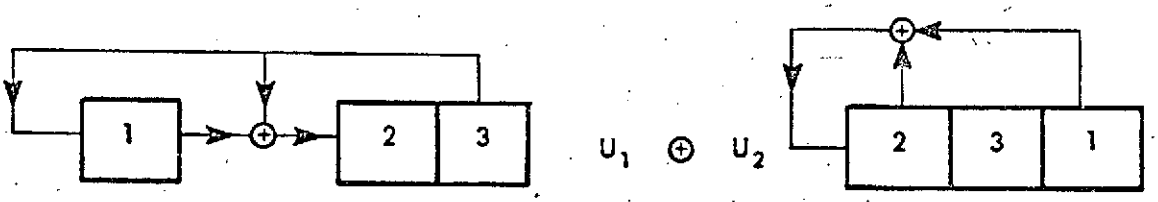


Figure 8b.

1st Bit	1	1	1	0
2nd "	1	0	1	1✓
3rd "	1	0	0	1
4th "	0	1	0	1
5th "	0	0	1	0
6th "	1	1✓	0	0
7th "	0	1	1✓	1
8th "	1✓	1	1	0
9th "	1	0	1	1
10th "	1	0	0	1
11th "	0	1	0	1

Figure 8a.

Now consider the mod 2 addition of stages 1 and 2 from Figure 8b i = 1, i-k = i - 2, k = 2;

$$U_1 + U_2 = U_1 (I + A^2) = A^{t_{12}} U_1 \qquad (19)$$

Taking the characteristic equation $A^3 = A + I$ and squaring it $A^6 = A^2 + I$ rewriting (19)

$$A^{t_{12}} = A^2 + I$$

$t_{12} = 6$ or stage 1 lags the mod 2 sum of 1 and 2 by 6 bits. The delay of stage 2 from 1 is -2 or +5. $\therefore U_2$ lags $U_1 + U_2$ by $t_1 + (-2) = 6 - 2 = 4$ or $t_1 + 5 = 6 + 5 = 4$ mod 7. The delay of U_3 from U_1 is -1 bits or +6. $\therefore U_3$ lags $U_1 + U_2$ by $t_1 + (-1) = 5$ bits or $t_1 + 6 = 6 + 6 = 5$ mod 7. The above results may be verified from the table of Figure 8.

Example 2:

Consider the shift register of Figure 9.

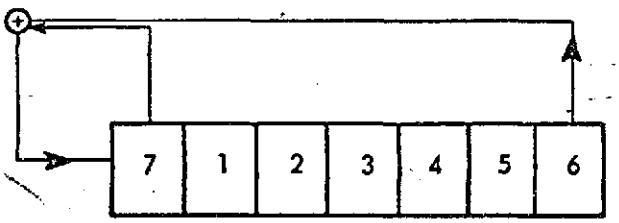


Figure 9.

The characteristic equation is $A^7 + A^6 + I = 0$

$$U_7 = U_2 A^2$$

$$U_5 = U_2 A^{-3}$$

so

$$U_7 + U_5 + U_2 = U_2 [I + A^2 + A^{-3}] = A^t U_2$$

Let us solve for t.

If t refers to U_2

$$A^5 + A^3 + I = A^{t+3}$$

now

$$A^3 + I = A^{t+3} + A^5$$

$$A^6 + I = A^7 = A^{2(t+3)} + A^{10}$$

so

$$A^{10} + A^7 = A^{2(t+3)}$$

$$A^7 (A^3 + I) = A^{2(t+3)}$$

$$A^3 + I = A^{2(t+3)-7}$$

Now let us find $A^3 + I = A^{t_1}$

$$A^6 + I = A^{2t_1} = A^7$$

so

$$2t_1 = 7 \text{ or } 7 \bmod 127 = 134$$

$$t_1 = 67$$

$$A^{67} = A^3 + I = A^{2(t+3)-7}$$

$$A^{74} = A^{2(t+3)}$$

$$t + 3 = 37$$

$$t = 34$$

so

U_2 lags $U_7 \oplus U_2 \oplus U_5$ by 34 bits.

Auto-Correlation of a Maximal Sequence

Recalling the definition of auto-correlation of a sequence

$$R_{xx}(\tau) = \frac{\sum_{t=1}^{L} x(t)x(t+\tau)}{\sum_{0}^{L} x^2(t)} = \frac{1}{L}\sum_{t=1}^{L} x(t)x(t+\tau) \qquad (20)$$

where τ is the integral phase difference between the two sequences, and L is the length of the sequence.

The terms of the above summation will be 1's if the corresponding digits of the two sequences match, and -1's if they do not match. Consequently the terms of the above summation may be considered as another sequence of -1's and 1's whose average must be taken.*

Rewriting the above in equation form

*This definition corresponds to the distance being the Hamming distance for the non-binary case.

$$R_{xx}(\tau) = \frac{1}{L}(\text{number of matching digits or 1's} - \text{number of differing digits or -1's}) \quad (21)$$

To solve for $R_{xx}(\tau)$ one must find the difference between the matching and differing digits. One way to do this is to transform the operation of multiplication to mod 2 addition.

Examination of Figures 10a and 10b, the tables representing multiplication and mod 2 addition respectively, shows that multiplication may be replaced by mod 2 addition if the 1's are replaced by 0's and the -1's are replaced by 1's.

x/y	-1	1
-1	1	-1
1	-1	1

x · y

or

x	y	x · y
+1	+1	1
+1	-1	-1
-1	+1	-1
-1	-1	1

Figure 10a. x · y

x/y	1	0
1	0	1
0	1	0

x ⊕ y

or

x	y	x ⊕ y
0	0	0
0	1	1
1	0	1
1	1	0

Figure 10b. x ⊕ y

Then Equation (20) where $x(t)$ and $x(t + \tau)$ is either 1 or -1 may be replaced by

$$R_{xx}(\tau) = \frac{1}{L}\sum_{t=1}^{L}[1 - 2y(t)]$$

where $x(t)$ and $x(t + \tau)$ is either 0 or 1, and $y(t) = x(t) \oplus x(t + \tau)$.

The difference between matching and differing digits or the difference between the number of 0's and 1's under mod 2 addition may now be found easily by making use of the shift and add property of maximal sequences, since the above operation is equivalent to taking a maximal sequence of 0's and 1's, shifting it by τ bits, ($\tau \neq 0 \bmod 2^n - 1$), and adding it to the original sequence.

We know from the shift and add property that the resulting sequence (whose elements are the terms of the summation for $R_{xx}(\tau)$ will be the same maximal sequence shifted in time by some integer delay τ_2. Using the balance property of maximal sequences, it is readily seen that there will be 2^{n-1} 1's and $2^{n-1} - 1$ 0's in the resultant sequence. Consequently

$$R_{xx}(\tau) = \frac{1}{L}(\text{number of 0's} - \text{number of 1's})$$

$$= \frac{1}{L}(2^{n-1} - 1 - 2^{n-1}) = -\frac{1}{L}$$

where $L = 2^n - 1$.

For $\tau = 0$ obviously

$$R_{xx}(\tau) = \frac{1}{L}\sum_{0}^{L} x^2(t) = 1$$

The Shift Register as a Linear Filter

Consider the Shift Register of Figure 11.

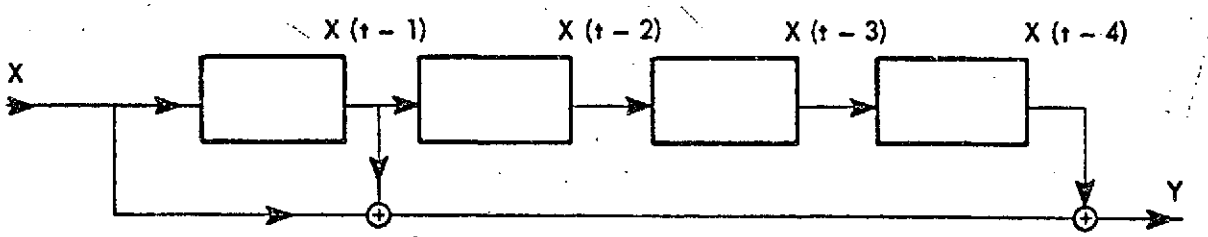


Figure 11.

If the input sequence X is u(t) = ... 0001000000 .....
then the output sequence Y is Tu(t) = ... 0001100100 .....
u(t) will be referred to as the impulse function and Tu(t) as the impulse response thus

$$T_m u(t) = u(t) + u(t-1) + u(t-4) \tag{22}$$

or in general

$$T_m u(t+i) = u(t+i) + u(t+i-1) + u(t+i-4); \quad i = 1, 2,$$

$$= u(t+i) + Du(t+i) + D^4 u(t+i)$$

$$= (I + D + D^4)\, u(t+i) \tag{23}$$

Since this is a linear circuit, if X is a linear combination of k impulse functions as given by Equation (24)

$$X = \sum_{i=1}^{k} a_i u(t+i) \tag{24}$$

then Y will be the sum of the responses due to each of these unit step functions.

Thus in the general case $a_i = 0, 1, 2, \ldots p-1$

$$Y = T_m X = T_m \sum_{i=1}^{k} a_i u(t+i) = (I + D + D^4) \sum_{i=1}^{k} a_i u(t+i) \tag{25}$$

The transfer function of the filter is defined as

$$T_m = \frac{Y}{X} = I + D + D^4 = I + A^{-1} + A^{-4} = A^{-4}(A^4 + A^3 + I) \tag{26}$$

If the input and output of Figure 11 are interchanged as shown in Figure 12 then

$$X = (D + D^4) X + Y$$

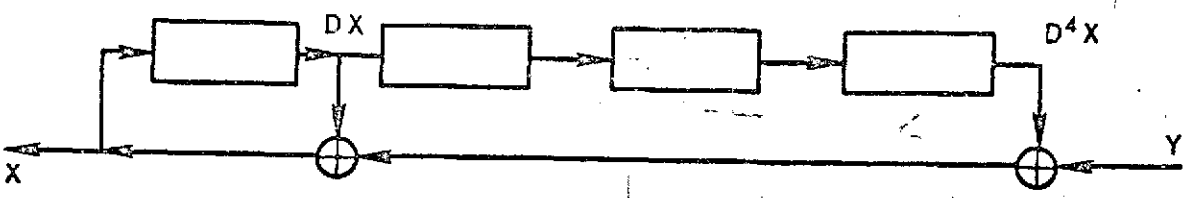


Figure 12.

or

$$(I - D - D^4)\, X = Y$$

and the transfer function becomes

$$T_D = \frac{X}{Y} = \frac{I}{I - D - D^4} \tag{27}$$

in the binary case Equation (27) becomes

$$T_D = \frac{I}{I + D + D^4} \tag{28}$$

By consulting Figure 6b it will be noticed that the characteristic equation of the SRG of Figure 11 is

$$\phi(A) = I + A^3 + A^4 = A^{+4}(A^{-4} + A^{-1} + I) = A^4(I + D + D^4)$$

$$= A^4 T_m$$

or in general for binary

$$T_m = A^{-n}\, \phi(A)$$

Multiplication and Division with Shift Registers

Referring to Figure 11 let

$$X = [a_k, a_{k-1}, a_{k-2}, a_{k-3}, a_{k-4} \ldots] = a_k + Da_{k-1} + D^2 a_{k-2} + D^3 a_{k-3} \ldots$$

where D^i is a delay operator representing a delay equal to i; let y_i be the response due to a_i with the zero vector as the contents of the SRG at time 0^- then

$$y_1 = a_k \quad a_k \quad 0 \quad 0 \quad a_k \quad 0 \quad 0 \quad 0 \quad 0 \quad \ldots.$$

$$Dy_2 = \quad a_{k-1} \quad a_{k-1} \quad 0 \quad 0 \quad a_{k-1} \quad 0 \quad 0 \quad \ldots.$$

$$D^2 y_3 = \quad a_{k-2} \quad a_{k-2} \quad 0 \quad 0 \quad a_{k-2} \quad 0 \quad 0 \quad \ldots.$$

$$D^3 y_4 = \quad a_{k-3} \quad a_{k-3} \quad 0 \quad 0 \quad a_{k-3} \quad 0 \quad 0 \quad \ldots.$$

$$D^4 y_5 = \quad a_{k-4} \quad a_{k-4} \quad 0 \quad 0 \quad a_{k-4} \quad 0 \quad 0 \quad \ldots.$$

$$D^5 y_6 = \quad a_{k-5} \quad a_{k-5} \ldots\ldots\ldots$$

Since this is a linear network, i.e.,

$$\text{if } Y(a_k) = y_1 \longrightarrow Y(a_k + Da_{k-1} + D^2 a_{k-2} + \ldots) = y_i + Dy_2 + D^2 y_3 + \ldots$$

thus

$$Y\left(\sum_{i=0} D^{k-i} a_{k-i}\right) = \sum_{i} D^{k-i} y_i \qquad (29)$$

$$= a_k + D(a_k + a_{k-1}) + D^2(a_{k-1} + a_{k-2}) + D^3(a_{k-2} + a_{k-3})$$

$$+ D^4(a_k + a_{k-3} + a_{k-4}) + D^5(a_{k-1} + a_{k-4} + a_{k-5}) + \cdots$$

$$+ D^{4+i}(a_{k-i} + a_{k-i-3} + a_{k-i-4}) + \cdots$$

$$= (I + D + D^4)\, a_k + (I + D + D^4)\, D a_{k-1} + (I + D + D^4)\, D^2 a_{k-2} + \cdots$$

$$= (I + D + D^4)\,(a_k + D a_{k-1} + D^2 a_{k-2} + \cdots)$$

thus the Y sequence is equal to the product of the X sequence and the factor $I + D + D^4$.

In general for the circuit of Figure 13

$$Y = (h_n + D h_n + D^2 h_{n-2} + D^3 h_{n-3} + \cdots + D^{n-1} h_1 + D^n h_0)\,(a_k + D a_{k-1} + D^2 a_{k-2} + \cdots)$$

$$= h(D) X(D)$$

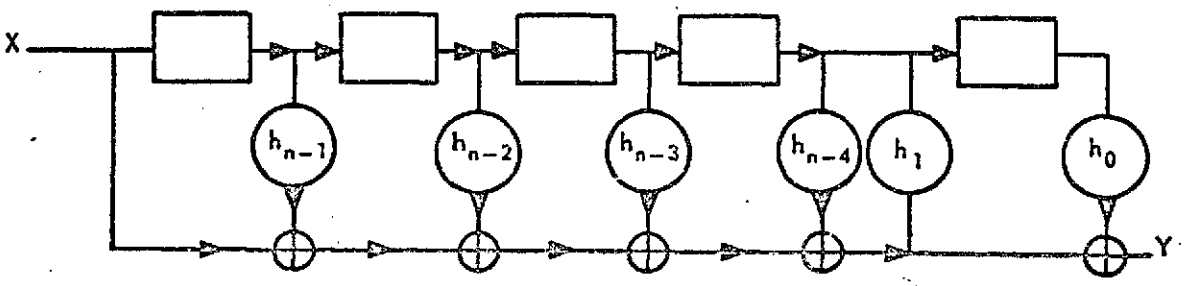


Figure 13. Multiplication by $h_n + D h_{n-1} + D^2 h_{n-2} + \cdots D^i h_{n-i} + \cdots + D^n h_0$

replacing D by A^{-1}

$$h(D) = I + A^{-1}h_{n-1} + A^{-2}h_{n-2} + \ldots + A^{-n-1}h_1 + A^{-n}h_0$$

$$= A^{-n}\left[h_0 + Ah_1 + A^2h_2 + \ldots + A^{n-1}h_{n-1} + A^n\right] = A^{-n}\left[2A^n - \phi(A)\right] = T$$

and

$$X(D) = \left[A^{-k} \quad A^k a_k + A^{k-1}a_{k-1} + \ldots\right]$$

then

$$Y = D^{k+n}\left[A^k a_k + A^{k-1}a_{k-1} + \ldots\right]\left[2A^n - \phi(A)\right]$$

Thus after k + n shifts* the sequence $Y' = TX - a_0$ will be out of the SRG output and the contents of the shift register will [000 .... 00 a_0].

Thus the sequence described by the product of T and X will be out of the register after n + k + 1 shifts.

Notice that the function of the network of Figure 12 becomes obvious by applying the results of circuit theory

$$Y = TX$$

and this clearly indicates a multiplier.

By similar procedure† we could arrive at a division network by reversing the inputs and outputs, i.e., the transfer function of Figure 13 is[11]

$$T_D = \frac{A^n}{\phi(A)}$$

*n 0's added to the end of the input sequence of k + 1 as

†See Appendix F.

and

$$Y = \frac{A^n X(D)}{\phi(A)} \qquad (30)$$

In general the a modular LMSC can be characterized by the equation

$$P(D)\, X = Q(D)\, Y$$

for example in Figure 14

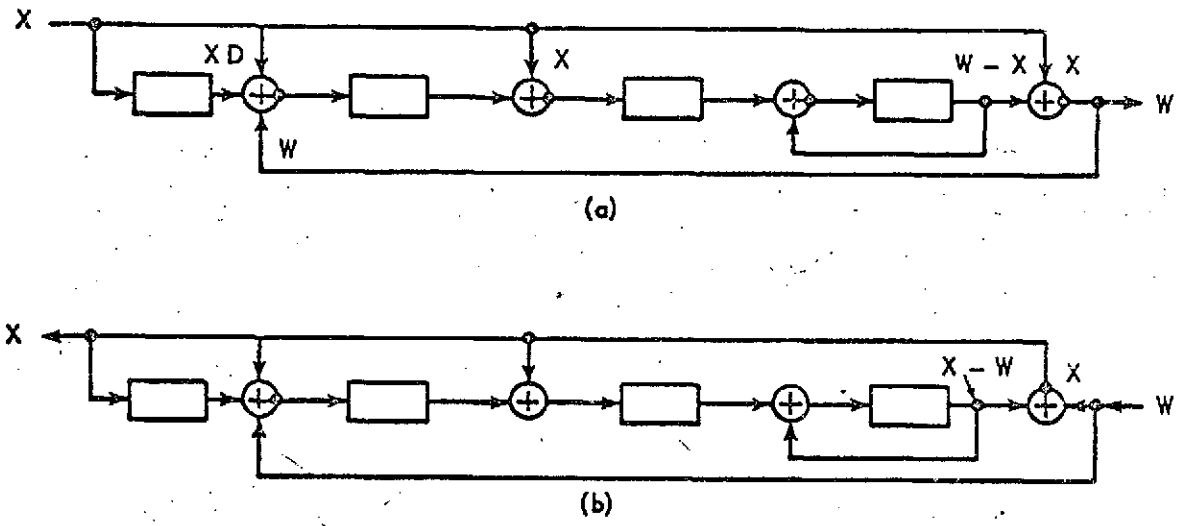


Figure 14. A LMSC and its Inverse

For Figure 14a

$$W = X + D^2X + D^3X + D^4X + D^3W + D(W - X)$$

or

$$(I - D - D^3)\, W = (I - D + D^2 + D^3 + D^4)\, X$$

and

$$T_m = (I - D + D^2 + D^3 + D^4)/I - D^3 - D^4$$

For Figure 14b

$$X = WD^3 + (X - W)\,D + X(D^2 + D^3 + D^4) + W$$

or

$$X(I - D - D^2 - D^3 - D^4) = W(I - D + D^3)$$

$$T_D = \frac{W}{X} = \frac{I - D + D^3}{I - D - D^2 - D^3 - D^4}$$

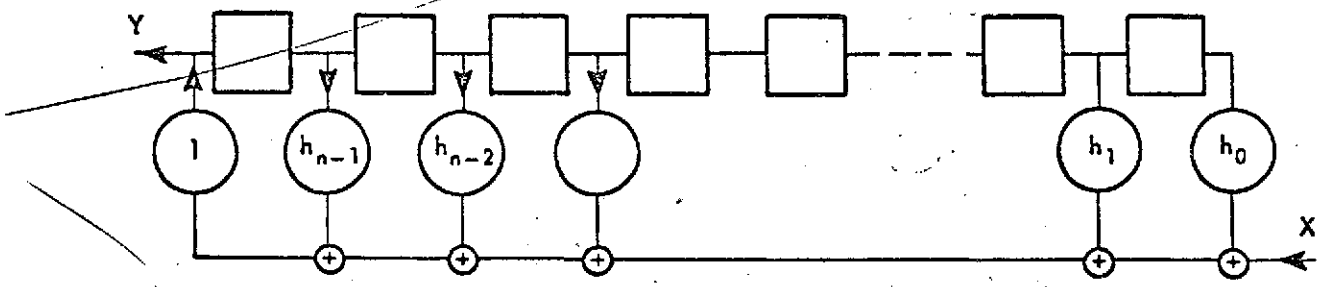


Figure 15. A Division Circuit

$$T_D = \frac{Y}{X} = \frac{I}{I - h_{n-1}D - h_{n-2}D^2 - h_{n-3}D^3 - \ldots - h_{n-i}D^i - \ldots - h_0 D^n} \qquad (31)$$

Appendix F contains more general multiplication and division circuits.

Error Correction with SRG's

Let a sequence X of $2^m - 1$ digits be fed into SRG. The output sequence $W = TX$ where T is the transfer function of the filter; if the noise sequence in the channel is N the received sequence will be

$$Y = W + N$$

if Y is used as the input of the inverse filter, T^{-1}, then

$$X_N = T^{-1} Y = T^{-1} (W + N) = X + T^{-1} N$$

To better understand the generation of a code let us take an example with the SRG of Figure 16 and p = 2.

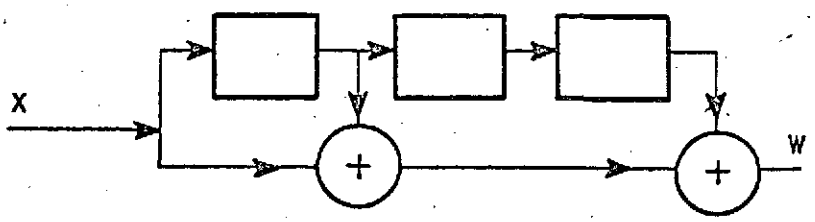


Figure 16.

The sequence W is 7 bits long ($2^3 - 1 = 7$). If the number of information bits in X is 4 then there are $2^4 = 16$ codewords whose first four bits are the information bits and the last 3 are zeros.

Then

$$x_i = (a_1\ a_2\ a_3\ a_4\ 0\ 0\ 0) \qquad a_i = 0 \text{ or } 1$$

then

$$w_i = T\, x_i = (D^3 + D + I)\,(a_1 + Da_2 + D^2 a_3 + D^3 a_4)$$

the x's and the corresponding w's are shown in Table 1.

For the construction of Table 1, the impulse function 1000000 = x_8 was first considered and the response w_8 was found from W = TX; next the delayed impulse functions x_4, x_2 and x_1 were found by delaying w_8 by the corresponding amount; once those basic responses are known then for any i, if x_i is expressed as a linear combination of $x_1 x_2 x_4$ and x_8

$$x_i = b_1 x_1 + b_2 x_2 + b_4 x_4 + b_8 x_8$$

Table 1

The Input and Output of the Encoder

$x_8 = 1\,0\,0\,0\,0\,0\,0$	$w_8 = 1\,1\,0\,1\,0\,0\,0 \leftarrow c(4)$	
$x_4 = 0\,1\,0\,0\,0\,0\,0$	$0\,1\,1\,0\,1\,0\,0 \leftarrow c(5)$	
$x_2 = 0\,0\,1\,0\,0\,0\,0$	$0\,0\,1\,1\,0\,1\,0 \leftarrow c(6)$	
$x_1 = 0\,0\,0\,1\,0\,0\,0$	$0\,0\,0\,1\,1\,0\,1 \leftarrow c(0)$	
$x_0 = 2x_8 = 0\,0\,0\,0\,0\,0\,0$	$w_0 + 2w_8 = 0\,0\,0\,0\,0\,0\,0$	—
$x_3 = x_1 + x_2 = 0\,0\,1\,1\,0\,0\,0$	$w_3 = w_1 + w_2 = 0\,0\,1\,0\,1\,1\,1 \leftarrow r(4)$	
$x_5 = x_1 + x_4 = 0\,1\,0\,1\,0\,0\,0$	$w_5 = w_1 + w_4 = 0\,1\,1\,1\,0\,0\,1 \leftarrow r(1)$	
$x_6 = x_2 + x_4 = 0\,1\,1\,0\,0\,0\,0$	$w_6 = w_2 + w_4 = 0\,1\,0\,1\,1\,1\,0 \leftarrow r(3)$	
$x_7 = x_1 + x_2 + x_4 = 0\,1\,1\,1\,0\,0\,0$	$w_7 = w_1 + w_2 + w_4 = 0\,1\,0\,0\,0\,1\,1 \leftarrow c(2)$	
$x_9 = 1\,0\,0\,1\,0\,0\,0$	$w_9 = 1\,1\,0\,0\,1\,0\,1 \leftarrow r(6)$	
$x_{10} = 1\,0\,1\,0\,0\,0\,0$	$w_{10} = 1\,1\,1\,0\,0\,1\,0 \leftarrow r(0)$	
$x_{11} = 1\,0\,1\,1\,0\,0\,0$	$w_{11} = 1\,1\,1\,1\,1\,1\,1$	
$x_{12} = 1\,1\,0\,0\,0\,0\,0$	$w_{12} = 1\,0\,1\,1\,1\,0\,0 \leftarrow r(2)$	
$x_{13} = 1\,1\,0\,1\,0\,0\,0$	$w_{13} = 1\,0\,1\,0\,0\,0\,1 \leftarrow c(3)$	
$x_{14} = 1\,1\,1\,0\,0\,0\,0$	$w_{14} = 1\,0\,0\,0\,1\,1\,0 \leftarrow c(1)$	
$x_{15} = 1\,1\,1\,1\,0\,0\,0$	$w_{15} = 1\,0\,0\,1\,0\,1\,1 \leftarrow r(5)$	

the corresponding w_i is

$$w_i = b_1 w_1 + b_2 w_2 + b_4 w_4 + b_8 w_8$$

Consider now the sequence generated by $\phi(D) = (I + D + D^3) = 0$ in the binary case; this sequence is $s(0) = 1110100$; the reverse of $s(0)$, $r(0) = 0010111$; the complement of $r(0)$ is $c(0) = 1101000$.

It turns out that the code W is the space consisting of all shifts of the sequence, r and c and the two vectors 0000000 and 1111111.

The set $\{w_3, w_5, w_6, w_9, w_{10}, w_{12}, w_{15}\}$ which contains elements that are formed by all possible shifts of the sequence r is sometimes referred to as a simplex code. The addition of the additive identity (all zero vector) turns the simplex code into a group code R. Our code as generated by the encoder of Figure 16 consists of the elements of R and their complements.*

The received sequence Y is used as the input of the SRG of Figure 17

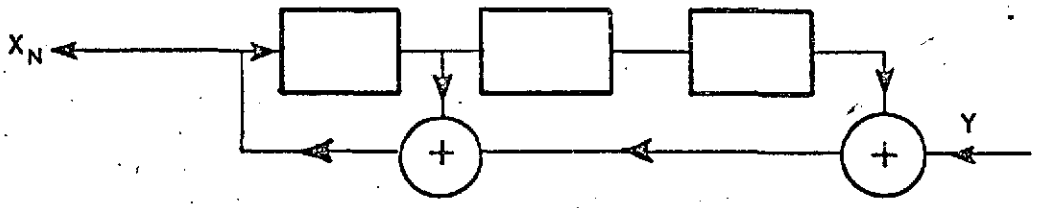


Figure 17.

If the noise is zero then $X_N = X$ (the three last digits of X_N are zero).

Suppose that one noise digit is a 1 (single error) then the noise vector can be considered as an impulse function and the response of the division SRG of Figure 17 to this impulse function is the sequence s, i.e., 1110100, then

$$X_N = X + 1110100$$

an altogether different vector from X. Since the number of codewords for this example is 16, k = 4 and m = 3, the last three bits of any X_i are to be zero; thus a non-zero vector in the last three bits of X_N indicates errors; by delaying the impulse discussed above the single error N's and the corresponding 3 last bits in X_N are shown in Table 2. Then the last three digits of X_N could be considered syndromes or the phase of the maximal sequence to be added to X_N to obtain X. Notice that the distance here is 3 or 4 so that all single errors are correctable.

In general for single error correction in a sequence of length n with m–0's at the end and k = n - m information bits we must have each of the S's different for each of the n impulses. For this to be possible, each set of m consecutive bits in the sequence corresponding to the impulse response must be different.

*The complement of $(a_1, a_2, a_3, a_4, \ldots)$ is defined as $(p-1-a_1, p-1-a_2, p-1-a_3, p-1-a_4, \ldots)$ with p as the modulus.

Table 2

N	$T^{-1}N$	
	Error In Information Bits	S
N_1 - 1 0 0 0 0 0 0	1 1 1 0	1 0 0 ⟵ S_1
N_2 - 0 1 0 0 0 0 0	0 1 1 1	0 1 0 ⟵ S_2
N_3 - 0 0 1 0 0 0 0	0 0 1 1	1 0 1 ⟵ S_3
N_4 - 0 0 0 1 0 0 0	0 0 0 1	1 1 0 ⟵ S_4
N_5 - 0 0 0 0 1 0 0	0 0 0 0	1 1 1 ⟵ S_5
N_6 - 0 0 0 0 0 1 0	0 0 0 0	0 1 1 ⟵ S_6
N_7 - 0 0 0 0 0 0 1	0 0 0 0	0 0 1 ⟵ S_7
N_0 - 0 0 0 0 0 0 0	0 0 0 0	0 0 0 ⟵ N_0

Since we have n non-zero impulses and $2^m - 1$ S's then for this to be possible n must be equal to a less than $2^m - 1$ for all impulses to be correctable and each of the m-bit subsequences of the impulse response of the decoder should be different. If $n = 2^m - 1$ then this is possible only if the impulse response of the decoder is a maximal sequence. To determine the optimality of this code the Hamming Bound* equation will be used letting M be the number of codewords

$$M \leq 2^n \Big/ \sum_{i=0}^{e} \binom{n}{i}$$

For single error correction e = 1 and

$$M \leq 2^n/(n+1)$$

*For single error correction the Hamming Bound gives the same result as the Varsharmov-Gilbert-Sacks condition.

When the equality sign holds the code is perfect*; for $n = 7$

$$M = \frac{2^7}{2^3} = 2^4 = 16$$

this implies that the maximum number of codewords that the code could have is 16. Consequently, since our code has 16 codewords, it is optimum.

Let us now review the necessary conditions for the existence of a single error correcting p-nary code generated from shifts of a feedback shift register sequence. If each single error is characterized by its amplitude ($p - 1$ values) and position (n positions) the number of distinct correctors must be at least equal to $n(p - 1)$.

Since each signal X is an n-tuplet with the first k symbols used for information and the last m symbols being zeros (check symbols) the number of distinct (non-zero) corrector combinations is $p^m - 1$. Thus:

$$n(p - 1) \leq p^m - 1 \tag{36}$$

or

$$n \leq \frac{p^m - 1}{p - 1}$$

which is the familiar Hamming bound (or Varsharmov-Gilbert-Sacks condition) for the length of the code. Since $n = m + k$, the above equation can be rewritten as

$$k \leq \frac{p^m - 1}{p - 1} - m$$

and it is an upperbound on the number of information bits (or number of words $M = p^k$).

*See (13)

Again in the case of equality the code is optimum. Thus for an optimum code

$$n(p-1) = p^m - 1; \quad n = \frac{p^m - 1}{p - 1}$$

This implies that we need $p^m - 1$ distinct m-tuplets in the sequence S, which is the impulse response of the decoder

$$S = T^{-1}(10000 \ldots . 000)$$

The only way to have this number of m-tuplets is to insure that S is a maximal sequence because only the maximal sequence has a length (period) equal to $p^m - 1$.

Then the code will be n symbols long

$$n = \frac{p^m - 1}{p - 1}$$

Since the length of the sequence is $p^m - 1 > n$ one needs only a segment of that sequence for each codeword; the only question is which segment. To illustrate the above let us consider an example.

Let $p = 3$ and $m = 2$; the maximal length of the sequence is $L = p^m - 1 = 8$.

Thus to find a polynomial which is primitive and period equal to 8 we must first find $f_8(\lambda)$ because $f_8(\lambda)$ contains all primitive roots* with period 8.†

The order of the polynomial is four ($\Phi(8) = 2^2 = 4$), and there are 2

$$\left(\frac{\Phi(8)}{m} = \frac{4}{2} = 2\right)$$

maximal sequences.

*See Appendix C.

†Since $8 = 2^3$ $p_1 = 2$ $e_1 = 3$

The length of the code is

$$n = \frac{p^m - 1}{p - 1} = \frac{8}{2} = 4$$

and $k = n - m = 2$; thus the number of the codewords is $p^k = 3^2 = 9$.

By using Equation (2) on page 6

$$f_8(x) = \frac{x^8 - 1}{x^4 - 1} = x^4 + 1 \tag{32}$$

this polynomial can be expressed as the product of two irreducible* polynomials

$$x^4 + 1 = (x^2 + x + 2)(x^2 + 2x + 2)$$

$$= \phi_1(x)\,\phi_2(x) \tag{33}$$

To find the matrices corresponding to $\phi_1(x)$ and $\phi_2(x)$ use the matrix in the companion form

$$\phi(x) = \det \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & & & & & & \vdots \\ 0 & 0 & \cdots & \cdots & \cdots & 0 & 1 \\ h_0 & h_1 & \cdots & \cdots & \cdots & h_{n-2} & h_{n-1} \end{bmatrix}$$

*See Appendix C.

thus

$$\phi_1(x) = x^2 + x + 2 = \det\left\{\begin{bmatrix} 0 & 1 \\ h_0 & h_1 \end{bmatrix} - xI\right\}$$

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} \text{ for } \phi_1(x) \tag{34}$$

and

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \text{ for } \phi_2(x) \tag{35}$$

Now we must see if the polynomial $\phi_2(x)$ is primitive by raising its roots to integer powers and observing the power L for which

$$(\text{root})^L = 1 \tag{37}$$

if the smallest non-zero L that satisfies the above equation is equal to $p^m - 1$ then the root is primitive and so is the polynomial. Since by the Caley-Hamilton theorem the matrix A satisfies the characteristic equation $\phi_2(x)$ then we could carry the above test with the matrix A instead of the root.

$$\phi_2(A) = A^2 + 2A + 2I = 0 \xrightarrow{\text{implies}} A^2 = A + I$$

by substituting all A^2 by A + I we can obtain Table 3 for the powers of A.

Table 3

$A^0 = I$			I
$A^1 = A$			A
$A^2 = A + I$			$A + I$
$A^3 = A(A + I)$	$= A^2 + A$	$=$	$2A + I$
$A^4 = A(2A + I)$	$= 2A^2 + A$	$=$	$2I$
$A^5 = A(2I)$		$=$	$2A$
$A^6 = 2A^2$		$=$	$2A + 2I$
$A^7 = 2A^2 + 2A$		$=$	$A + 2I$
$A^8 = A^2 + 2A$	$= 3A + I$	$=$	I

It should be noticed that the same results could have been obtained by matrix multiplication, i.e.

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

$$A^2 = AA = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} = A + I$$

$$A^3 = AA^2 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix} = 2A + I$$

$$A^4 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 \\ 2 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = 2\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = 2I$$

$$A^5 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix} = \begin{bmatrix} 0 & 2 \\ 2 & 2 \end{bmatrix} = 2A$$

$$A^6 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 2 \\ 2 & 2 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix} = 2A + 2I$$

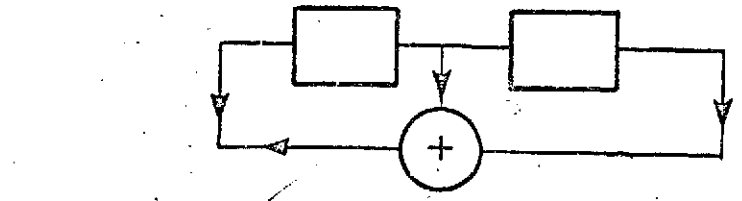
$$A^7 = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 2 & 2 \\ 2 & 1 \end{bmatrix} = \begin{bmatrix} 2 & 1 \\ 1 & 0 \end{bmatrix} = A + 2I$$

$$A^8 = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

The shift register that obeys $\phi_2(\lambda)$ in the form of Equation (10) ($\phi_2(\lambda) = \lambda^2 - h_1\lambda - h_0 = \lambda^2 - \lambda - 1$ which is equivalent to $\lambda^2 + 2\lambda + 2$) is shown in Figure 18; notice that the contents of the shift register at shift i represent the coefficients of the polynomial representing A^i in Table 4; notice also that

Table 4

Number	X	W	Shift of s
0	0 0 0 0	0 0 0 0	0
1	0 1 0 0	0 1 2 2	1
2	0 2 0 0	0 2 1 1	5
3	1 0 0 0	1 2 2 0	2
4	1 1 0 0	1 0 1 2	8
5	1 2 0 0	1 1 0 1	7
6	2 0 0 0	2 1 1 0	6
7	2 1 0 0	2 2 0 2	3
8	2 2 0 0	2 0 2 1	4



Shift	Contents	
0	0	1
1	1	0
2	1	1
3	2	1
4	0	2
5	2	0
6	2	2
7	1	2
8	0	1

Figure 18. The PRG with + $\phi_2(\lambda) = A^2 - A - I = (A^2 + 2A + 2I) = 0$ and the Contents at Time i i = 0, . . . 8

the sequence s = 1120, 2210 is the maximal impulse response and it consists of two subsequences

$$s = S_1, 2S_1$$

where S_1 = 1120.

The code can be generated by using the multiplication circuit which multiplies by the polynomial $\phi_2(\lambda) = A^2 + 2A + 2I$. This can be accomplished with the circuit of Figure 19.

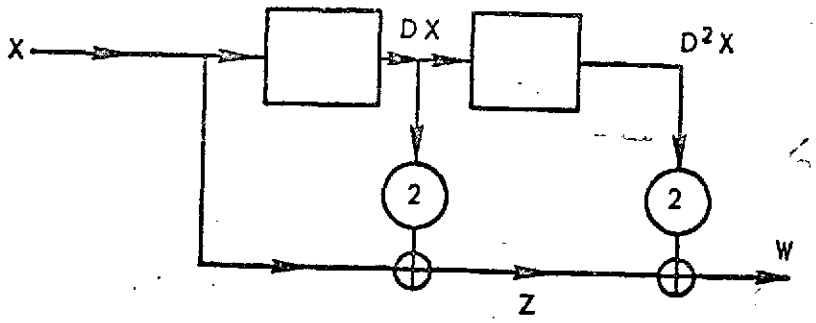


Figure 19. The Encoder

so that

$$W = \phi_2(A)\, X(A) = (A^2 + 2A + 2I)\, X(A)$$

$$= A^{2+n}(I + 2D + 2D^2)\, X(D)$$

The decoder results from the division circuit which is constructed using the following reasoning. Assuming that there are no errors the received signal $Y = W$ multiplied by some function F must result in the transmitted signal X

$$FW = X \tag{38}$$

but since

$$W = TX \tag{39}$$

Equation (38) becomes

$$FTX = X \tag{40}$$

Equation (40) implies that

$$F = \frac{I}{T} \tag{41}$$

Since F is the output X_N divided by the input Y Equation (41) can be rewritten as

$$\frac{X_N}{Y} = \frac{I}{T} \text{ or } X_N = \frac{Y}{T} = \frac{Y}{\phi_2(A)} = \frac{Y}{A^2 + 2A + 2I}$$

Application of Equation (8) and Figure 6 with $m = 2$ gives $h_0 = -2$ $h_1 = -2$ and the decoder becomes the circuit of Figure 20

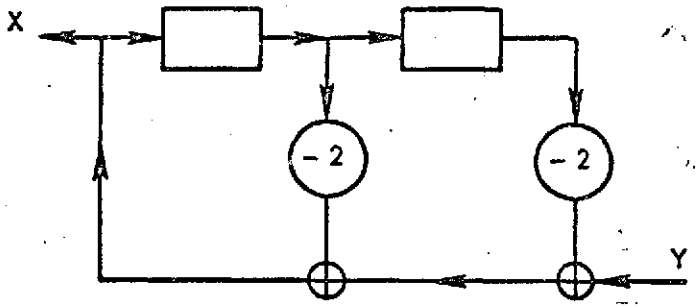


Figure 20.

Since $-2 = 1 \bmod 3$ the decoder of Figure 20 is equivalent to the decoder of Figure 21.

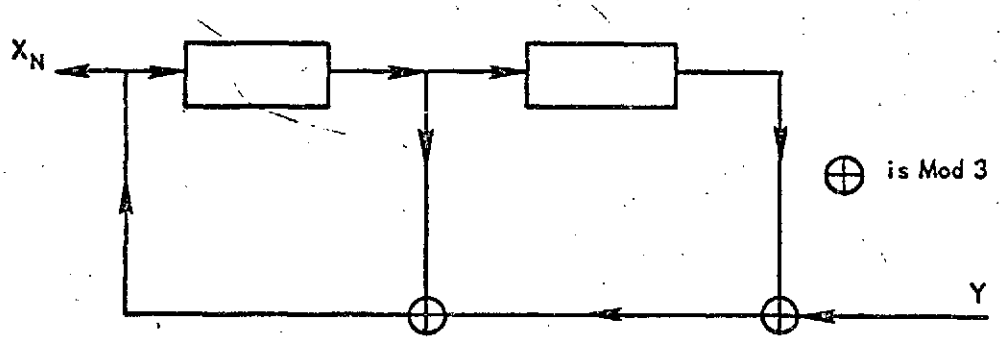
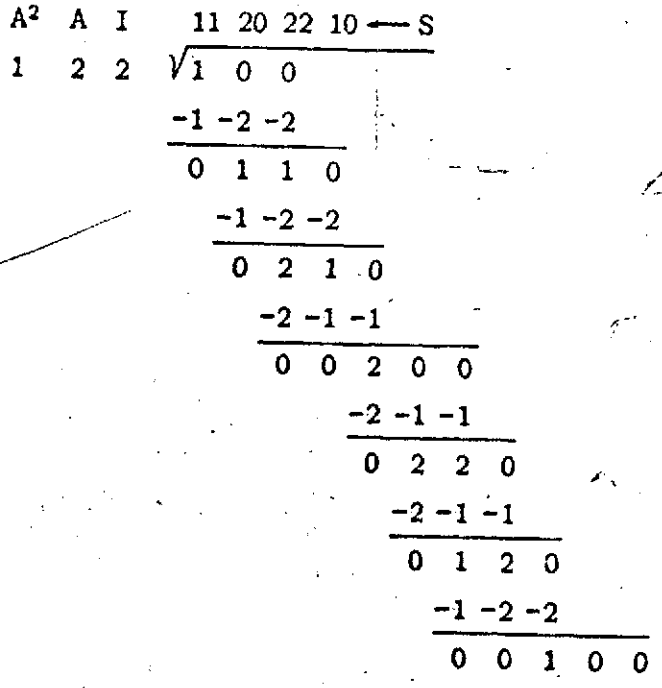


Figure 21. The Decoder Corresponding to the Encoder of Figure 11

The impulse response of the decoder is the sequence s of page 45; this sequence could aiso be obtained by long division as follows.



The code W as generated by the circuit of Figure 19 is shown in Table 4 and is seen to be segments (subsequences) of the reverse sequence r of page 38.

The response of the decoder circuit of Figure 21 to all single error patterns is shown in Table 5. It should be noted that all eight $\left((p-1)\binom{n}{1}\right)$ single error patterns are correctable because the corresponding correctors (the ternary representation of the last two digits in the response excluding the 0000 vector) are distinct.

To reconstruct the information symbols of X no change is required when the corrector S is 11, 01, 22, or 02 (and of course 00).

When the error becomes obvious the beginning of the sequence must be added mod 3 to X_N to correct it; one way to produce the beginning of the sequence if one knows the end is to use the connection for the reverse sequence which is produced by the shift register of Figure 22 (see Appendix E).

Table 5

N	Response = $T^{-1}N$	Error In Information Symbols	S	Number Represented by S
1 0 0 0	1 1 2 0	1 1	2 0	6
0 1 0 0	0 1 1 2	0 1	1 2	5
0 0 1 0	0 0 1 1	0 0	1 1	4
0 0 0 1	0 0 0 1	0 0	0 1	1
2 0 0 0	2 2 1 0	2 2	1 0	3
0 2 0 0	0 2 2 1	0 2	2 1	7
0 0 2 0	0 0 2 2	0 0	2 2	8
0 0 0 2	0 0 0 2	0 0	0 2	2

The Hamming bound for this code gives the maximum k.

$$k_{max} = n_{max} - m \quad \text{for } e = 1$$

$$n_{max} = p^{m-1} + p^{m-2} + \cdots + 1$$

$$= 3 + 1 = 4$$

thus since in our case n is also equal to 4 this code is optimum.

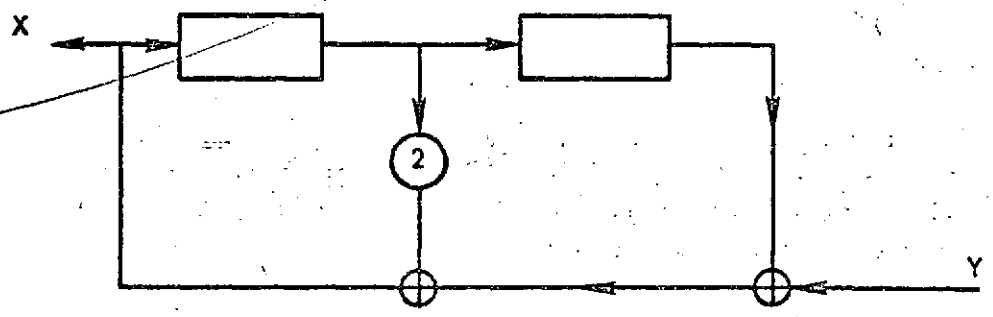


Figure 22.

SUMMARY

Thus far we have used the m-stage Shift Register for single error correction by encoding k digits into $k + m = n$ such that

$$n \leq \frac{p^m - 1}{p - 1} \tag{42}$$

with the equality resulting in an optimum code; it should be kept in mind that for this class of single error correcting codes the check bits are equal to the number of stages of the shift register and n is determined from Equation (42). If n is much smaller than the value indicated by the equality of Equation (42) then the code may be able to correct more than single errors. For example if $p = 2$ $m = 4$ then all single errors will be correctable if

$$n \leq p^m - 1 = 15$$

if $n = 15$ then $k = 11$ and the size of the code, if a SRG is possible, would be 2^{11}. Suppose that we don't need such a large size for the code but we still want n to be the length of the maximal sequence ($n = 15$) then the code if selected properly will be able to correct more than all the single errors if $k = 4$ then $M = 16$ and we can use as codewords the 15 shifts of the maximal sequence and the all zero vector. The Hamming distance for this code is 8 and the code will correct all single double and triple errors and detect all quadruple errors. These codes will be discussed later if we want k to be five $M = 32$, and the code words will be the fifteen shifts of the maximal sequence, the fifteen shifts of the reverse sequence the zero vector and the 1 vector. Since the distance is the weight of each code word, and the minimum weight of the reverse codes is 7, the code will correct all single, double and triple errors, and it will detect some of the quadruple errors (some of the distances have the value 8).*

Pseudorandom Codes

In the last section the codes were in general segments of the maximal sequence; however the codeword was all L shifts of the maximal sequence. In this section we shall consider codes whose words are the $p^k - 1$ shifts of a maximal sequence and the all zero vector.

*This is the code described in (17).

For this code $n = p^k - 1$, the number of information digits are k, and they could be taken as the 1st k digits of the sequence. This code can be generated from a k-stage SRG of Figure 23 loading the stages of the generator with the k information digits and then letting it shift for $p^k - 1$ shifts.

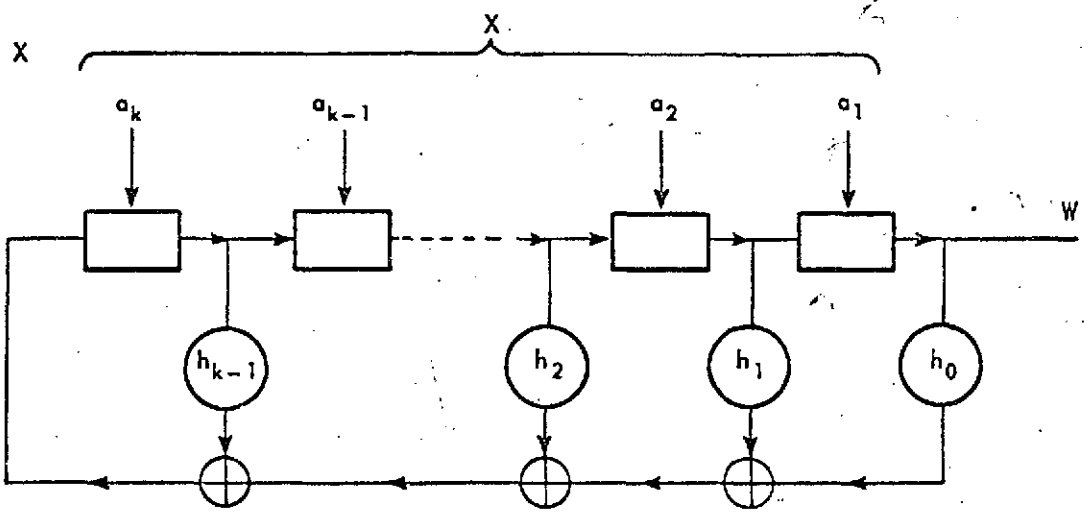


Figure 23. Pseudorandom Encoder

This code can be decoded by choosing the best match* between the received sequence and all shifts of a locally generated sequence using an identically connected SRG, or by a division circuit or by majority decision circuits. The optimality of this code will be proved by finding the minimum distance and comparing it to the theoretical upperbound (see Appendix E) for systematic codes.

Since all non-zero codewords of this code are shifts of a maximal sequence the distance vector is also the same maximal sequence shifted by some shift.

Since a maximal sequence of length $p^k - 1$ contains all possible $p^k - 1$ non-zero k-tuplets and since each digit has the same frequency except for the zero digit that appears one less than any other element, each element must appear p^{k-1} times in the sequence and the zero element must appear $p^{k-1} - 1$ times.

*This process is digital correlation.

Applying the notions of Appendix E, the Hamming distance of each distance vector (which is also the minimum distance d_H) is

$$d_H = p^{k-1} \sum_{i=1}^{p} d_W(a_i) = p^{k-1}(p-1)$$

but for any systematic code it was proven in Appendix E that d_H cannot be larger than

$$\frac{n\, p^{k-1}(p-1)}{p^k - 1}$$

letting $n = p^k - 1$ we find that d_H cannot be larger than $p^{k-1}(p-1)$.

Consequently this code is optimum with respect to the Hamming distance.

Using a similar argument for the Lee distance for this code

$$d_L = p^{k-1} \sum_{i=1}^{p} d_w(a_i) = p^{k-1} \frac{(p+1)(p-1)}{4} = \frac{p^{k-1}(p^2-1)}{4}$$

but from Appendix E for any systematic code

$$d_L \leq \frac{n\, p^{k-1}(p^2-1)}{4(p^k-1)}$$

again letting $n = p^k - 1$

$$d_L \leq \frac{1}{4} p^{k-1}(p^2-1)$$

Again we see that the pseudorandom code is also optimum with respect to Lee distance.

This code will correct all single double triple up to $e = \frac{d-1}{2}$ errors.

It should be pointed out that the correlation and majority decision decoding are equivalent but use different mathematical notions.

In the case of correlation we correlate by counting the matches and mismatches of the sequence and a locally generated sequence. The length of the maximal sequence is L and in the binary case we count A (= number of matches) and D (= number of mismatches).

If $A > D$ we decide that the sequence is correct. Since each sequence disagrees with all others in $\frac{L+1}{2}$ bits it will take a change in $\frac{L+1}{4}$ or more bits $\left(\frac{L+1}{4} \text{ errors}\right)$ to have

$$A > D$$

for the wrong sequence.

In the case of majority decision each bit (say a_0) is given by $\frac{L+1}{2}$ algebraic equations.

If more than half of these equations contain erroneous bits, it is possible that more than half of the equations will give the wrong solution for a_0. Thus it takes at least $\frac{L+1}{4}$ errors to decode incorrectly.

Further Application of Shift Register Generators

The error correcting encoding and decoding application described in this document are simply some of the immediate applications of pseudorandom sequences and multiplication and division circuits to coding.

Some of the important applications of these circuits are their use for encoding and decoding a class of cyclic codes called Bose-Chauduri-Hocquenghem codes,* (BCH), and the Recurrent or Convolutional Codes. Due to their special importance, the BCH and recurrent codes will be discussed in a separate paper.

*The Hamming codes are a special case of the B-C-H codes.

REFERENCES

1. Birdsall, T. G., and M. P. Ristenbatt — Introduction to Linear Shift-Register Generated Sequences, EDG Technical Report No. 90, University of Michigan Research Institute (1958).

2. Dickson, L. E. — "Linear Groups with an exposition to Galois Field Theory", Dover

3. Hohn, F. E. — "Elementary Matrix Algebra", McMillan Company, New York 1964.

4. Huffman, D. A. — "The Synthesis of Linear Sequential Coding Networks" Proc. of the Third London Symposium on Information Theory, September 13, 1955.

5. Huffman, D. A. — "A Linear Circuit Viewpoint on Error-Correcting Codes", IRE Trans., IT-2, 20-28 (1956).

6. Abramson, N. — "Error Correcting Codes from Linear Sequential Networks," TR No. 2002-1, Stanford Electronics Labs, June 1960.

7. Stern, T. E., and B. Friedland — "Application of Modular Sequential Circuits to Single-Error-Correcting P-Nary Codes," IRE Trans., IT-5, 114-123 (1959).

8. Stern, T. E., and B. Friedland — "On the Periodicity of States of Linear Modular Sequential Circuits", IRE Trans. on Circuit Theory, CT-6, pp. 136-137.

9. Friedland, B. — "Linear Modular Sequential Circuits, CT-6, pp. 61-68.

10. Elspas, B. — "The Theory of Autonomous Linear Sequential Networks", IRE Trans., CT-6, 45-60 (1959).

11. Peterson, W. W. — "Error Correcting Codes", MIT-Wiley 1961.

12. Colomb, S. et al. — Digital Communications with Space Applications", Prentice Hall, N.J. 1964.

13. Morakis, J. C. — "On the Structure of algebraic Codes with Maximum Likelihood Decoding", NASA GSFC TMX 520-68-50, February 1958.

14. Church, R. — Tables of Irreducible Polynomicals for the first four prime moduli. Annals of Mathematics, Vol. 36, No. 1, January 1935.

15. Campopiano, C. N. — Construction of Relatively Maximal Codes of Specified Minimum Distance from Linear Recurring Sequences of Maximal Period. IT-6, pp. 523-528, December 1960.

16. Lee, C. Y. — "Some Properties of Nonbinary Error-Correcting Codes", IRE Trans., IT-4, 77-82 (1958).

17. Green, J. H., Jr. and R. L. San Soucie — "An Error-Correcting Encoder and Decoder of High Efficiency", Proc. IRE, 46, 1741-1744 (1958).

APPENDIX A

The Characteristic Equation of the Transpose of a Matrix

It shall be proven here that the matrix A and its transpose A^T have the same characteristic equation

let $$B = |A - \lambda I|$$

where λ is a constant and I the identity matrix

$$B^T = |A - \lambda I|^T = A^T - \lambda I^T = A^T - \lambda I$$

$$\det |A^T - \lambda I| = \det B^T = \det B = \det |A - \lambda I|$$

APPENDIX B

The Determinant of the Companion Matrix

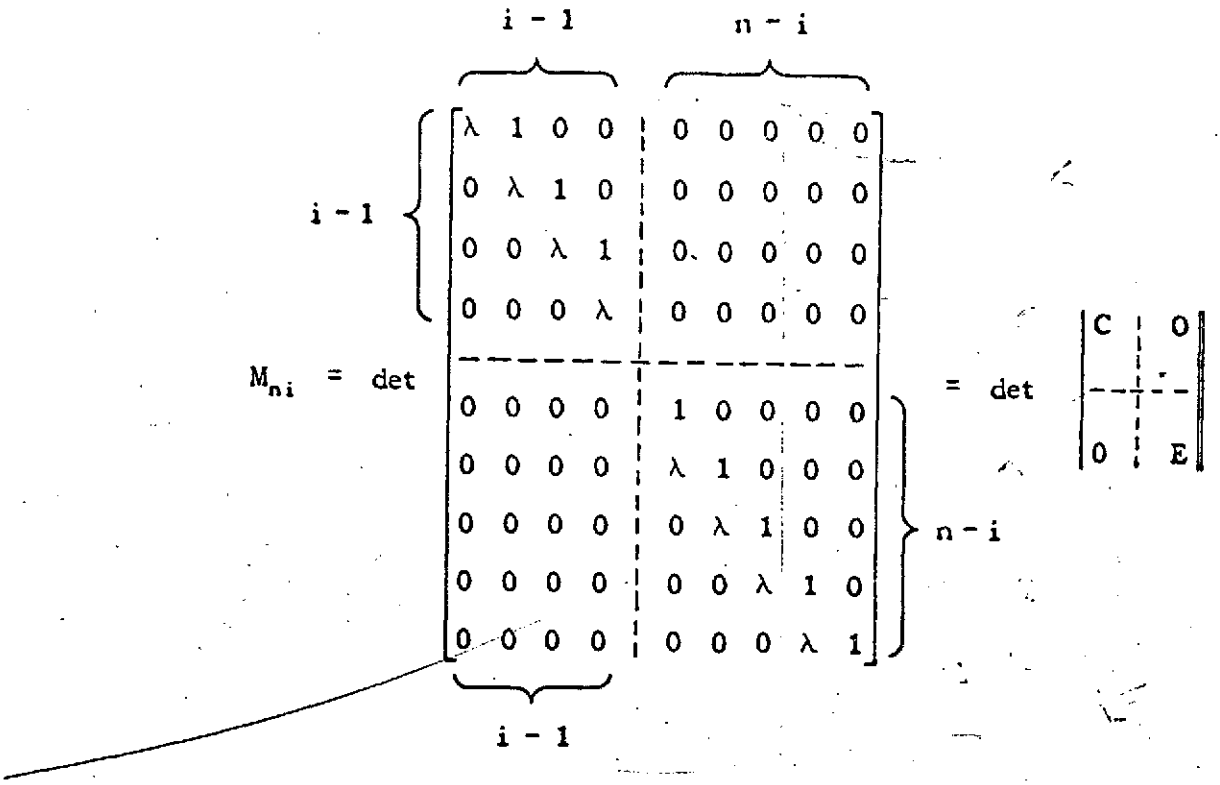
Consider the matrix

$$B = \begin{vmatrix} \lambda & 1 & 0 & 0 & 0 & 0 & \cdots & & & 0 \\ 0 & \lambda & 1 & 0 & 0 & 0 & \cdots & & & 0 \\ 0 & 0 & \lambda & 1 & 0 & 0 & \cdots & & & 0 \\ \vdots & & & & & & & & & \\ 0 & 0 & 0 & 0 & 0 & 0 & \lambda & 1 & \cdots & 0 \\ \vdots & & & & & & & & & \\ 0 & 0 & 0 & 0 & \cdots & & & 0 & \lambda & 1 \\ 0 & 0 & 0 & 0 & \cdots & & & 0 & 0 & \lambda \\ h_0 & h_1 & h_2 & h_3 & & & & h_{n-2} & h_{n-1} & +\lambda \end{vmatrix}$$

to find the determinant of the above matrix expand the determinant in terms of minors of the elements of the last row

$$\det B = \sum_{i=1}^{n-1} (-1)^{n+i} h_{i-1} M_{ni} + (h_{n-1} + \lambda) M_{nn}$$

for any i the minor is the determinant of the matrix B with the i^{th} column and last row deleted is shown below



$$= \det C \det E$$

but C is an upper triangular matrix with λ in the diagonal elements.

$$\det C = (\lambda)^{i-1}$$

and E is a lower triangular matrix with unity diagonal elements thus

$$M_{ni} = \det C \det E = \lambda^{i-1}(1) = \lambda^{i-1}$$

so

$$\det B = \sum_{i=1}^{n-1} (-1)^{n+i} h_{i-1} \lambda^{i-1} + (h_{n-1} + \lambda) \lambda^{n-1}$$

$$= \sum_{i=1}^{n} (-1)^{n+i} h_{i-1} \lambda^{i-1} + \lambda^n$$

APPENDIX C

Some Important Concepts of Linear Algebra and some Results of Galois Field Theory

This Appendix is not intended to be a rigorous exposition of Linear Algebra but rather a collection of definitions and results that are applicable to the material of this paper.

1. Definition: If the difference of two integers a and b is divisible by a third integer p then a and b are said to be congruent modulo p or in equation form

$$a \equiv b \bmod p \tag{C-1}$$

In general if $0 \leq r < p$ then for any integer t

$$r + tp \equiv r \bmod p \tag{C-2}$$

all numbers $c + tp$ for $t = \pm 1, \pm 2, \pm 3, \ldots$ form a class of residues modulo p (C_r).

2. Addition, subtraction, and multiplication of the C_r's present no problem but division is equivalent to finding a solution for x in

$$r = sx + tp$$

letting $r = 1$, $s = 3$, and $p = 6$, there is no solution for x; to insure the existence of a solution of x for all s then p must be a prime.

3. Fermat's Theorem: If an integer a is not dividible by a prime number p then $a^{p-1} \equiv 1 \pmod p$.

Corollary = if a and p are defined as above there exists a unique solution for $ax \equiv b \pmod p$ and it is found by multiplying the left side by 1 and the right by a^{p-1}

$$1 \cdot ax \equiv a^{p-1} b \pmod p$$

dividing by a

$$x \equiv a^{p-2} b \pmod p.$$

4. Groups: A group is an algebraic system with one operation and its inverse. The two operations that we are familiar with are addition and multiplications, with inverses subtraction and division resp; however, the operations associated with an algebraic system may be other than the above and they will be in general denoted by $\odot$. Thus if the Group A is a set of elements a_1, a_2, ... then $a_i \odot a_j = a_k$ implies that element a_k results from the operation $\odot$ on the elements a_i and a_j. The order of the group is the number of element in the group.

For a set of elements a_1, a_2 .... to be a group under the operation $\odot$, the following four axioms must be satisfied.

i. Closure

For $a_i \in A$, $a_j \in A$ then $(a_i \odot a_j) \in A$

ii. Associative Law:

$$(a_i \odot a_j) \odot a_k = a_i \odot (a_j \odot a_k)$$

iii. Identity element I

The group contains a unique identity element, I, such that

$$a_i \odot I = I \odot a_i = a_i$$

iv. Inverses

There exists a unique inverse a_i^{-1} for each element a_i in the group such that

$$a_i \odot a_i^{-1} = a_i^{-1} \odot a_i = I$$

Note: The fact that we were able to commute the identity and the inverse does not imply that all other elements in the group can in general be commuted unless the group is commutative.

5. Definition: An Abelian or commutative group is defined as a set of elements that obey the four axioms that define a Group plus the commutative law. If the operation is addition the group is additive, $I = 0$ and $a_i^{-1} = -a_i$. If the operation is multiplication the group is multiplicative, $I = 1$ and $a_i^{-1} = 1/a_i$. Obviously, since division by zero is undefined axiom 4 holds for all non-zero elements in the group and in axiom 3 the identity must be non-zero; for example all real numbers form an additive group, and all real numbers except zero form a multiplicative group. The group of matrices is not Abelian because matrices are not in general commutative.

6. Rings: A ring R is a set of elements with two operations addition* (+), and multiplication* (.), that satisfy

a. All addition axioms for an Abelian Group

b. The closure and Associative Laws for Multiplication

c. The Distributive Law: if $a, b, c \in R$ then

$$a \cdot (b + c) = a \cdot b + a \cdot c$$

If the elements of a Ring satisfy the commutative law under multiplication, then the Ring becomes a Commutative Ring.

7. Fields: A field is a set of elements that satisfy all axioms of a Commutative Ring and furthermore it has

a. A multiplicative identity

b. A multiplicative inverse for every non-zero element

Thus a Field is

a. An Abelian Group under Addition

b. An Abelian Group under Multiplication

b. The elements obey the distributive law

For example the elements $0, 1, 2, \ldots p - 1$ (where p is prime) form a Field.

*These two operations are not necessarily identical to the ones that we are familiar with.

Thus a set of s distinct elements forms a field of order s if the elements can be combined by addition, subtraction, multiplication and division the divisor not being the element zero (necessarily in the set), these operations being subject to the laws of elementary algebra, and if the resulting sum, difference, product or quotient is uniquely determined as an element of the set.

The requirements set forth by the division are equivalent to the existence of a solution of $r = sx \bmod p$ discussed in 2, and necessitates that p be a prime.

The complete system of classes of residues modulo p forms a field if, and only if p is a prime number.

9. Definition: A finite field as defined above is a Galois field.

Euler's Generalization of Fermat's Theorem.

If a is prime to $m > 0$.

$$a^{\Phi(m)} \equiv 1 \pmod{m}$$

where $\Phi(m)$ is Euler's function and it is equal to the number of integers less than m and relatively prime to m.

SUMMARY

Postulates of Real Numbers R (a, b, c are in R)	Semi Group	Group (Additive)	Group (Multiplicative)	Abelian Group (Additive)	Abelian Group (Multiplicative)	Ring	Commutative Ring	Field
I. Addition postulates:								
(i) a + b in R (Closure)	x	x		x		x	x	x
(ii) a + b = b + a (Commutative)				x		x	x	x
(iii) a + (b + c) = (a + b) + c (Associative)	x	x		x		x	x	x
(iv) There exists "0" such that 0 + a = a (Identity)		x		x		x	x	x
(v) There exists b such that a + b = 0 (Inverse)		x		x		x	x	x
II. Multiplication postulates:								
(i) a.b in R (Closure)			x		x	x	x	x
(ii) a.b = b.a (Commutative)					x		x	
(iii) a.(b.c) - (a.b).c (Associative)			x		x	x	x	x
(iv) There exists "1" such that 1.a-a (Identity)			x		x			x
(v) There exists b such that a.b = 1 (For a ≠ 0) (Inverse)			x		x			x
III. Distributative law: a.(b + c) = a.b + a.c						x	x	x

The aforementioned notions of groups, rings and fields can be extended to polynomials with a few logical modifications on definitions.

a. For example two numbers are relatively prime if they have no common divisors (except for unity). Two polynomials are relatively prime if they have no common factors.

b. A number P is prime if it cannot be factored to a form

$$p = p_1^{e_1} p_2^{e_2} \ldots .$$

where p_i is prime larger then unity and $e_i > 0$.

Similarly a polynomial is prime (or irreducible) if it cannot be factored into a product of prime polynomials.

c. A set of polynomials can form a group just like a set of any other elements; under addition one adds the coefficients of similar powers.

d. Given a number p and an irreducible polynomial P(x) a congruence can be established mod [p, P(x)] between two polynomials F(x) and f(x) by the equation

$$F(x) \equiv f(x) \mod [p, P(x)]$$

which implies that

$$F(x) \equiv f(x) + p \cdot g(x) + P(x) \cdot Q(x)$$

if P(x) is of degree n then the degree of f(x) is less than n and its coefficients are less than p.

(The above is equivalent to dividing F(x) by G(x) to obtain f(x) and then reducing the coefficients of f(x) mod p).

All polynomials of degree n - 1 form a field of p^n elements (the number of distinct polynomials whose coefficients can assume one of p values, and there are n terms i.e. powers of 0, 1, 2, ... n - 1).

Then the ordered n-tuplet consisting of the coefficients of the polynomial completely characterizes the polynomial and there are p^n of these n-tuplets that form the field $GF(p^n)$.

Powers of Elements

We see then that we can treat polynomials as any other elements, so let us cite some of the definitions that apply to elements of a group in general.

a. An element can be raised to a power, the element may be a simple one such as a number; thus if we consider the group consisting of the numbers 0, 1, 2, 3, 4 (mod 5) then

$$2^0 = 1,\ 2^1 = 2,\ 2^2 = 4,\ 2^3 = 8 = 3,\ 2^4 = 16 = 1$$

On the other hand we could consider a root α of the equation

$$x^2 + x + 2 = 0$$

then

$$\alpha^2 + \alpha + 2 = 0 \quad \text{or} \quad \alpha^2 = -\alpha - 2 = 2\alpha + 1 \quad (\text{mod } 3)$$

$$\alpha^0 = 1,\ \alpha^1 = \alpha,\ \alpha^2 = -\alpha - 2 = 2\alpha + 1,\ \alpha^3 = 2\alpha^2 + \alpha = 4\alpha + 2\alpha = 2\alpha + 2,\ \ldots.$$

Still another aspect is an element x^e which is congruent to an element $f(x)$ of $GF(p^n)$ mod $[p, P(x)]$; thus if $p = 3$ and $P(x) = x^2 + x + 2$ such that $x^2 = 2x + 1$ mod $(3, x^2 + x + 2)$

$$x^0 = 1,\ x^1 = x,\ x^2 = 2x + 1,\ x^3 = 2x^2 + x = 4x + 1 + x = 2x + 2,\ \ldots.$$

It should be noticed that in the last two examples there exists a one-to-one correspondence between the powers of an element mod $[p, P(x)]$ and the powers of the roots of the polynomial $P(x)$.

Period of an Element

If a is an element, then the smallest positive integer e ($e \neq 0$) such that $a^e = 1 \bmod p$ is called the period of the element.

Example: let $a = 4 \bmod 5$

$$a^1 = 4 \quad a^2 = 16 = 1 \quad \text{then} \quad e = 2$$

similarly the period of x mod $(3, x^2 + 1)$ $(x^2 = 2)$ is

$$x^1 = x \quad x^2 = 2 \quad x^3 = 2x \quad x^4 = 2x^2 = 4 = 1 \quad e = 4$$

We also say that a belongs to the exponent e mod p or P(x).

Primitive Elements

If the cycle of an element a (mod p) is equal to the total number of non-zero elements in the group then the element is primitive.

Corollary: Since $a^i = b (i < e)$ is another element in the group and since $a^i = c \in G$ $b \neq c$ then the powers of this primitive element a generate all the non-zero elements of the group.

Example: Element 2 of the Group (0, 1, 2, 3, 4) is primitive because its powers generate all non-zero elements of the group

$$2^0 = 1 \quad 2^1 = 2 \quad 2^2 = 4 \quad 2^3 = 3 \quad 2^4 = 16 = 1$$

Element 4 is non-primitive (we could tell because e = 2 less than 4 the total non-zero elements)

$$4^0 = 1, \; 4^1 = 4, \; 4^2 = 16 = 1, \; 4^3 = 64 = 4, \; \text{etc.}$$

then it does not generate elements 2 and 3.

Similarly the powers of the element x generate all polynomials f(x) of degree n - 1 of GF(p^n) modulo 3, $x^2 + x + 2$. Table 5 shows the coefficients a, b, c of $ax^2 + bx + c$ Mod $(3, x^2 + x + 2)$ and the coefficients d, e, f of $d\alpha^2 + e\alpha + f$ of the powers of the root α of the polynomial $x^2 + x + 2$.

Since all $3^2 - 1 = 8$ of the non-zero elements of GF(p^n) are spanned (or generated) by the powers of x the polynomial is primitive and its period is $e = p^n - 1$.

Table C-1

The Powers of x Mod 3, $(x^2 + x + 2)$
and the Powers of the Root α of $x^2 + x + 2 = 0$

	b	c		e	f	Number Represented
x^0	0	1	α^0	0	1	1
x^1	1	0	α^1	1	0	3
x^2	2	1	α^2	2	1	7
x^3	2	2	α^3	2	2	8
x^4	0	2	α^4	0	2	2
x^5	2	0	α^5	2	0	6
x^6	1	2	α^6	1	2	5
x^7	1	1	α^7	1	1	4
$x^8 = 1$	0	1	$\alpha^8 = 1$	0	1	

A primitive polynomial is also defined as an irreducible polynomial whose roots are primitive.

Since usually the problem is to find the polynomial or polynomials belonging to the period* e the procedure of finding all irreducible polynomials with period e will be outlined.

First some preliminaries — since

$$x^e = 1 \mod [p, P(x)]$$

then

$$x^e - 1 = 0 \mod [p, P_i(x)]$$

where $P_i(x)$ are all irreducible polynomials belonging to e. Thus, since $x^e - 1 \equiv 0$ implies

$$x^e - 1 \equiv f(x)\, P_i(x) + 0 \qquad \text{for each } i$$

then each $P_i(x)$ must be a factor of $x^e - 1$; the expression of the product of the irreducible factors of $x^e - 1$ is given by the cyclotomic polynomial

$$f_e(x) = \prod_i^N P_i(x)$$

$$f_e(x) = \frac{(x^e - 1) \prod_{i,j} \left(x^{\frac{e}{p_i p_j}} - 1\right) \prod_{i,j,k,\ell} \left(x^{\frac{e}{p_i p_j p_k p_\ell}} - 1\right) \cdots\cdots}{\prod_i \left(x^{\frac{e}{p_i}} - 1\right) \prod_{i,j,k} \left(x^{\frac{e}{p_i p_j p_k}} - 1\right) \cdots\cdots}$$

where

$$e = \prod_i p_i^{e_i} \qquad p_i \text{ prime}$$

*This is equivalent to finding polynomials that have period e.

As a matter of fact $f_e(x)$ has all the primitive roots of period e. If $f_e(x)$ has N such factors and if the degree of each factor is n then the degree of $f_e(x)$ is nN.

But it can be proven* that the degree of $f_e(x)$ is $\Phi(e)$ thus:

$$Nn = \Phi(e)$$

and

$$N = \frac{\Phi(e)}{n}$$

if $e = p^n - 1$ then $f_e(x)$ contains all irreducible polynomials if with primitive roots.

Determining Whether an Irreducible Polynomial Is Primitive

The irreducible factors of $x^e - 1$ ($e = p^m - 1$) are identical to the irreducible polynomials of all degrees n such that n divides m; then

$$x^e - 1 = P_1(x)\, P_2(x) \ldots . P_N(x)$$

if the number of polynomials of degree n is I(n), then

$$\sum_{n/m} nI(n) = p^m$$

On the Periods of Non-Maximal Sequences

If e divides $p^n - 1$ and e does not divide $p^i - 1$ for any $i < n$ then there are $\frac{\phi(e)}{n}$ irreducible polynomials of degree n with period e.

Obviously if $p^n - 1$ is prime† then e must of necessity be $p^n - 1$ (maximal period).

* [1] p. 20 or [11] p. 138

† Mersenne prime

APPENDIX D

The Reverse Sequence

Consider the shift register of Figure D-1.

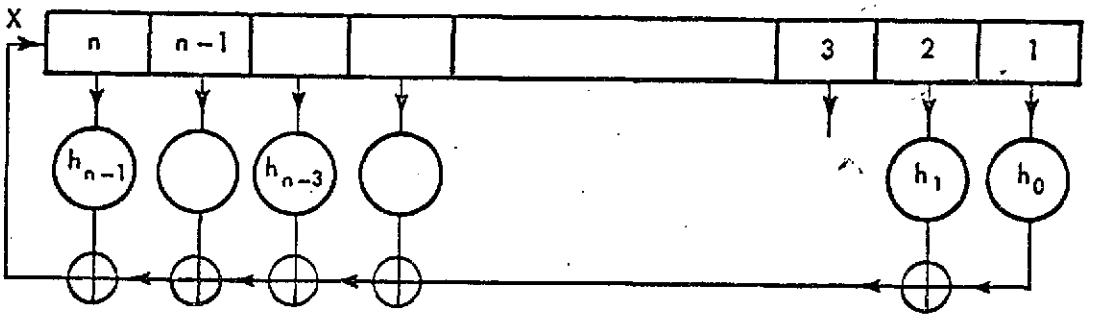


Figure D-1

$$X = (h_{n-1}D + h_{n-2}D^2 + \ldots + h_1D^{n-1} + h_0D^n)\,X \qquad \text{(D-1)}$$

where

$$\phi(D) = (I - h_{n-1}D - h_{n-2}D^2 - \ldots - h_1D^{n-1} - h_0D^n) \qquad \text{(D-2)}$$

A sequence r reverse to X is obtained by letting the new sequence r be the same function of the digits ahead instead of the past digits. This can be accomplished by changing D^i to D^{-i} then the reverse sequence r is

$$r = (h_{n-1}D^{-1} + h_{n-2}D^{-2} + \ldots h_1D^{1-n} + h_0D^{-n})\,r \qquad \text{(D-3)}$$

multiplying both sides by D^n.

$$D^n\,r = (h_{n-1}D^{n-1} + h_{n-2}D^{n-2} + \ldots h_1D + h_0)\,r \qquad \text{(D-4)}$$

rearranging

$$h_0 r = (D^n - h_{n-1} D^{n-1} - h_{n-2} D^{n-2} \cdots - h_1 D)\, r$$

$$= (- h_1 D - h_2 D^2 - \ldots - h_{n-1} D^{n-1} + D^n)$$

$$r = \left[- \frac{h_1}{h_0} D - \frac{h_2}{h_0} D^2 - \ldots - \frac{h_{n-1}}{h_0} D^{n-1} + \frac{1}{h_0} D^n \right] \qquad \text{(D-5)}$$

Comparing Equations (D-1) and (D-5) we can see that to obtain r we must change h_0 to $\frac{1}{h_0}$ and

$$h_i \text{ to } - \frac{h_{n-i}}{h_0}$$

or h_i to the negative mod p of $\frac{h_{n-i}}{h_0}$. Thus the SRG for r is the one in Figure D-2.

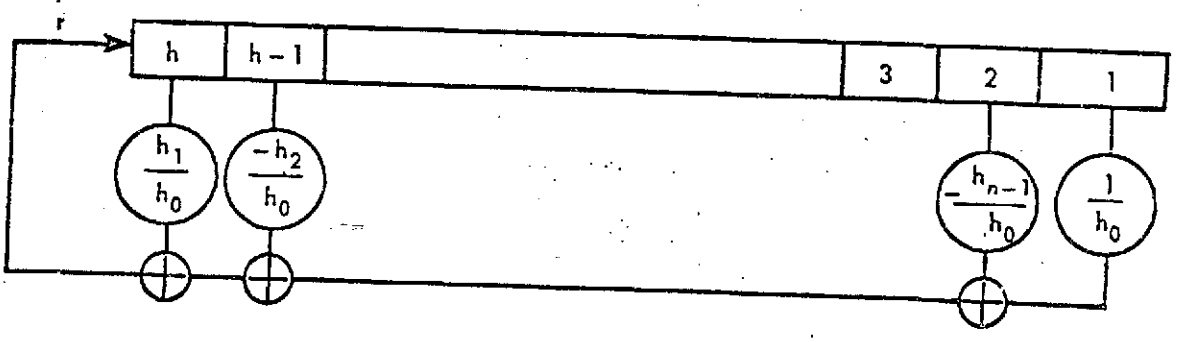
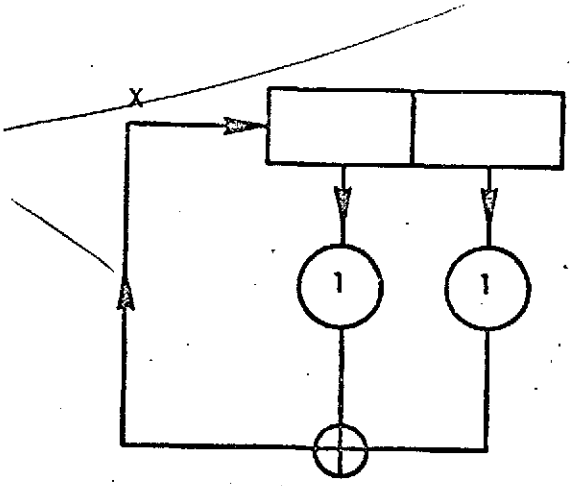


Figure D-2

Example consider the SRG of Figure D-3 $p = 3$.

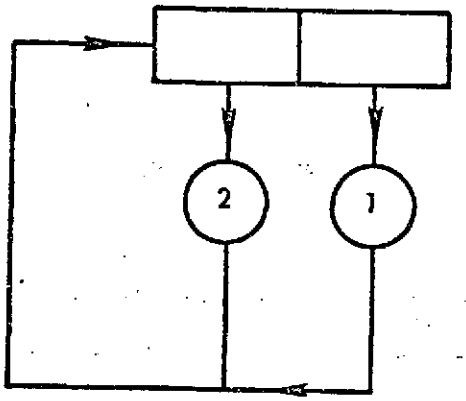


1	0
1	1
2	1
0	2
2	0
2	2
1	2
0	1
1	0

Figure D-3

In this register the sequence is $x = 1\ 1\ 2\ 0\ 2\ 2\ 1\ 0$ and $h_0 = h_1 = 1$.

Changing h_0 to $\frac{1}{h_0} = 1$ and h_1 to $-\frac{h_1}{h_0} = -1 = 2$ we obtain the circuit of Figure D-4.



1	0
2	1
2	2
0	2
2	0
1	2
1	1
0	1
1	0

Figure D-4

The sequence of Figure D-4 is $1\ 2\ 2\ 0\ 2\ 1\ 1\ 0$ which is the reverse of x.

Note: Division $\frac{a}{b}$ mod p is accomplished by taking an integer t, $1 \leq t \leq p - 1$, such that $\frac{pt + a}{b}$ is integer.

APPENDIX E

Optimality of Codes (Hamming and Lee Distance)

There are more than one ways of defining optimum codes; one way is to keep n the number of symbols and M the number of words fixed and find a code such that no other (n, M)* code will result in a smaller probability of error (or larger e).

All e error correcting codes that do not correct any e + 1-tuple errors are called perfect (close-packed) and the ones that do not correct any of the e + 2-tuple errors are called quasiperfect. Obviously both perfect and quasi-perfect codes are optimum.

Since $e = \frac{d-1}{2}$ where d is the minimum distance between any two code-words in the code, maximizing d is equivalent to maximizing e and consequently the performance of a code can be evaluated by means of the parameter d.

In the next section we shall find the maximum-minimum distance of a group code, however, we must first define what we mean by distance.

The cyclic (or Lee) [16] distance is the function

$$d = \sum_{i=1}^{n} d_i$$

where

$$d_i = \min \{a_i - b_i, b_i - a_i\}$$

and w_1, w_2 two codewords are defined as

$$w_1 = (a_1 a_2 a_3 \ldots a_n) .$$

$$w_2 = (b_1 b_2 b_3 \ldots b_n)$$

*$M = p^k$ usually.

On the other hand the Hamming distance is defined as the number of places in which two codewords disagree

$$d = \sum_{i=1}^{n} d_i$$

where

$$d_i = \begin{cases} 1 & \text{if } a_i \neq b_i \\ 0 & \text{if } a_i = b_i \end{cases}$$

In the case of p = 2 or 3 the two above mentioned notions of distance become the same.

Example: Consider the two words in GF (5)

$$(p = 5)$$

$$w_1 = 4\ 3\ 2\ 1\ 2\ 0$$

$$w_2 = 1\ 2\ 4\ 1\ 3\ 4$$

$$w_2 - w_1 = -3\ -1\ 2\ 0\ 1\ 4 = 2\ 4\ 2\ 0\ 1\ 4$$

$$w_1 - w_2 = 3\ 1\ -2\ 0\ -1\ -4 = 3\ 1\ 3\ 0\ 4\ 1$$

the Lee distance vector is 2 1 2 0 1 1 and

$$d_L = 2 + 1 + 2 + 1 + 1 = 7$$

the Hamming distance vector is 1 1 1 0 1 1 and the Hamming distance is

$$d_H = 5$$

Definition: A systematic code is a code W that consists of codewords $w_i \in W$ such that the first k digits are the information digits; the last m digits are the check digits and they are linear combinations of the information digits

$$a_i = \sum_{j=1}^{k} c_j a_j \qquad i = k+1, \ldots, k+m$$

Theorem: A systematic code is also a parity check code.

Theorem: Every parity check code is also a group code and vice-versa.

Since for any code

$$e = \frac{d-1}{2}$$

where d is the minimum distance between any two distinct codewords, then for fixed n and M(k) the code with the maximum d (d = min. distance) is an optimum code.

On the Maximum d of a Systematic Code

Consider an array made up of all p^k codewords of an (n, k) systematic p-nary code. This arrangement generates an $p^k \times n$ array (matrix) called the C matrix.

Theorem: Each element of GF(p) appears exactly p^{k-1} times in each column of C.

Proof: Each column of C consists of p^k elements; to prove that each element will appear p^{k-1} times is equivalent to proving that the number of times that each element appears is a constant K and consequently $Kq = q^m$ and $K = q^{m-1}$.

Clearly each of the information digits appear an equal amount of times because each digit is used with all p^{k-1} combinations of the remaining k - 1 information digits. Now let us consider a check digit in the i^{th} position (k i)

$$a_i = \sum_{j=1}^{k} c_i a_j$$

Since a_i is a linear combination of the information digits and since each information digit appears p^{k-1} times then a_i will also take each of the p values an equal number of times.

Example: Let $a_{k+1} = a_1 + a_5 \mod 3$, Table 6 shows the values of a_{k+1} corresponding to all a_1, a_5 combinations

0	0	0
0	1	1
0	2	2
1	0	1
1	1	2
1	2	0
2	0	2
2	1	0
2	2	1

Now using the closure axiom of a group code, the sum (or difference) of any two words is also another codeword and in the case of difference (sum of w_i and the inverse of w_j) the distance (Lee vector of all two word combinations) is a word in the code.

Thus each element of the distance vectors will appear np^{k-1} times in C.

In the case of Hamming distance each non-zero element contributes a d_w equal to 1 so the total d_w is

$$np^{k-1} \sum_{i=1}^{p} d_w(a_i) = np^{k-1}(0 + 1 + 1 + 1 + 1 \ldots . 1) = np^{k-1}(p - 1)$$

In the case of Lee distance

$$d_w = \min \{a_i, -a_i\}$$

thus d_w takes on the values shown in Table 7

Table E-1

The Lee Distance Functions

a_i	0	1	2	3	...	$\frac{p-1}{2}$	$\frac{p+1}{2}$	$\frac{p+3}{2}$	...	$p-1$	0
$-a_i$	0	$p-1$	$p-2$	$p-3$	...	$\frac{p+1}{2}$	$\frac{p-1}{2}$	$\frac{p-3}{2}$	...	1	0
d_w	0	1	2	3	...	$\frac{p-1}{2}$	$\frac{p-1}{2}$	$\frac{p-3}{2}$	...	1	0

and the total d_w is

$$d_w = np^{k-1}\, 2\left[1 + 2 + 3 + \ldots \frac{p-1}{2}\right] \qquad p \text{ is odd}$$

$$= np^{k-1}\left(\frac{p+1}{2}\right)\left(\frac{p-1}{2}\right) = \frac{n}{4}\, p^{k-1}\,[p^2 - 1] \qquad p \text{ odd}$$

Since there is a total of $p^k - 1$ words the maximum d cannot be larger than the average distance; thus

$$d_H \leq \frac{np^{k-1}(p-1)}{p^k - 1} \tag{E-1}$$

and

$$d_L \leq \frac{np^{k-1}(p^2-1)}{4(p^k - 1)} \tag{E-2}$$

APPENDIX F

Multiplication and Division Using Modular and Simple Shift Registers

Let t be the number of stages of the Modular Shift Register of Figure F-1a with X representing the input sequence and Y the output sequence; if D signifies a delay operator and D^i a delay of i units, then

$$Y = X + h_{t-1} D X + h_{t-2} D^2 X + \ldots + h_1 D^{t-1} X + h_0 D^t X \qquad (F-1)$$

$$= \left[I + h_{t-1} D + h_{t-2} D^2 + \ldots h_1 D^{t-1} + h_0 D^t\right] X$$

if D^i is replaced by A^{-i} where A is an advance operator the above equation becomes

$$Y = A^{-t} \left[h_0 + h_1 A + h_2 A^2 + \ldots h_{t-1} A^{t-1} + A^t\right] X \qquad (F-2)$$

Since the characteristic equation is

$$\phi(A) = -\sum_{i=0}^{t-1} h_i A^i + A^t \qquad (F-3)$$

the Equation (F-2) can be rewritten as

$$Y = D^t \left[-\phi(A) + 2A^t\right] X \qquad (F-4)$$

the multiplication of X by $A^t + \sum_i h_i A^i$ is obvious; the D^t factor indicates that one will need t extra shifts to get the product sequences out of the shift register.

Referring to Figure F-1b with X and Y as the input and output sequences respectively,

$$Y = X + h_{t-1} D X + h_{t-2} D^2 X \ldots + h_0 D^t X$$

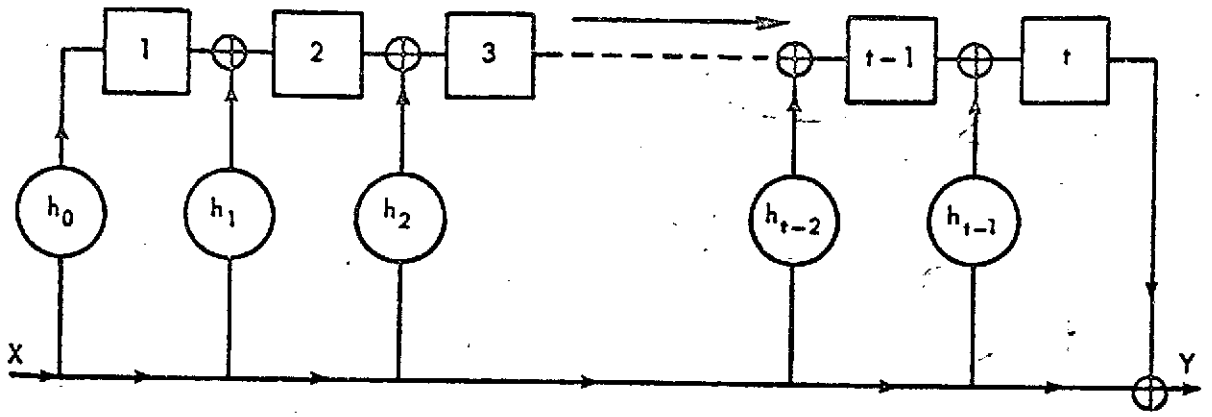


Figure F-1a. MSRG Multiplier

and obviously this equation is identical to Equation (F-1) and will yield the same results.

Referring to Figure F-2a the division by this circuit becomes apparent by solving for Y in terms of X

$$Y = D^t(X + h_0Y) + D^{t-1}h_1Y + h_2D^{t-2}Y + \ldots + h_{t-1}D^1Y \qquad \text{(F-5)}$$

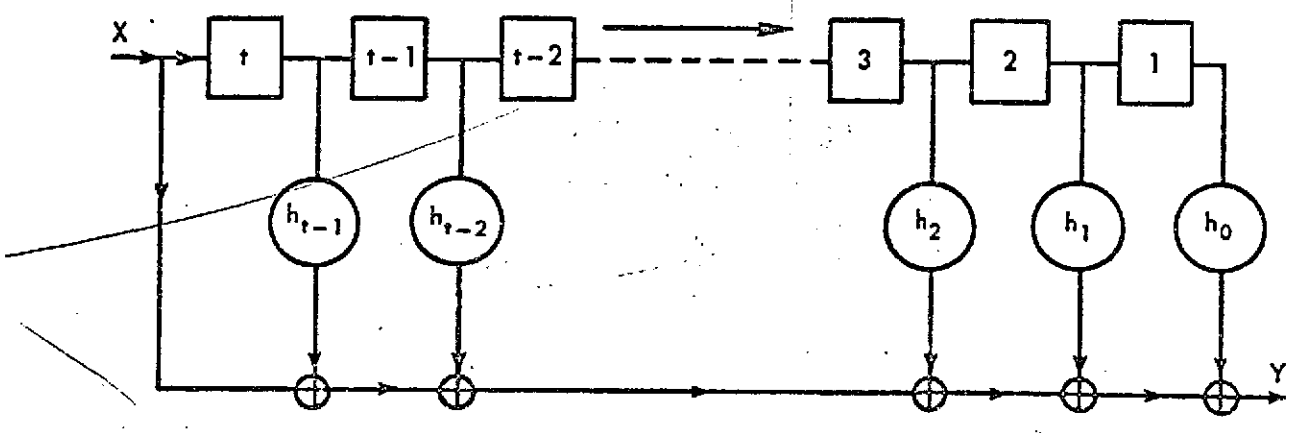


Figure F-1b. SSRG Multiplier

or

$$Y = \frac{D^t X}{I - \sum_{i=0}^{t-1} h_i D^{t-i}} = \frac{X}{A^t - \sum_{i=0}^{t-1} h_i A^i} = \frac{X}{\phi(A)} \qquad \text{(F-6)}$$

Similarly referring to Figure F-2b we can develop Equations (F-7) and (F-8)

$$\left.\begin{aligned} Z &= X + h_{t-1} DZ + h_{t-2} D^2 Z + \ldots h_1 D^{t-1} Z + h_0 D^t Z \\ &= \frac{X}{I - \sum_{i=0}^{t-1} h_i D^{t-i}} = \frac{A^t X}{A^n - \sum_{i=0}^{t-1} h_i A^i} = \frac{A^t X}{\phi(A)} \end{aligned}\right\} \qquad \text{(F-7)}$$

but

$$Y = D^t Z \qquad \text{(F-8)}$$

thus

$$Y = D^t Z = \frac{D^t A^t X}{\phi(A)} = \frac{X}{\phi(A)}$$

Another dividing circuit with an advance equal to t is shown in Figure F-3.

The equivalence of Figure F-3 to Figure F-2b is apparent because X remains in the same place but Y is moved ahead by t stages which accounts to the advance of amount t.

Thus for the circuit of Figure F-3

$$Y = X + DYh_{t-1} + h_{t-2} D^2 Y + \ldots h_0 D^t Y$$

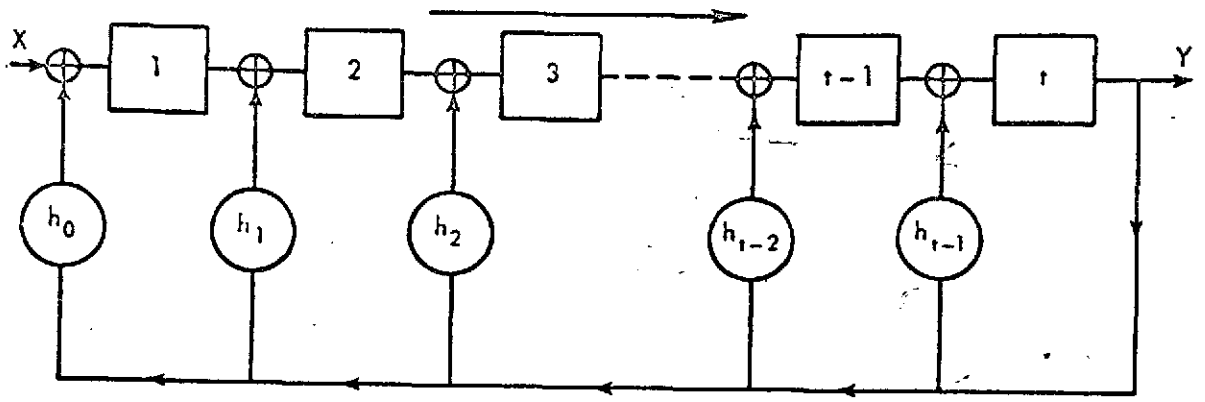


Figure F-2a. MSRG Divider

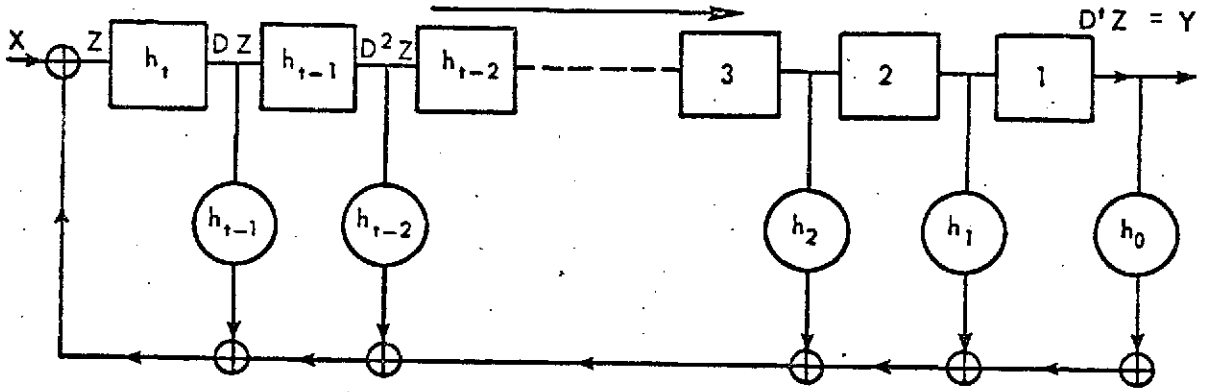


Figure F-2b. SSRG Divider

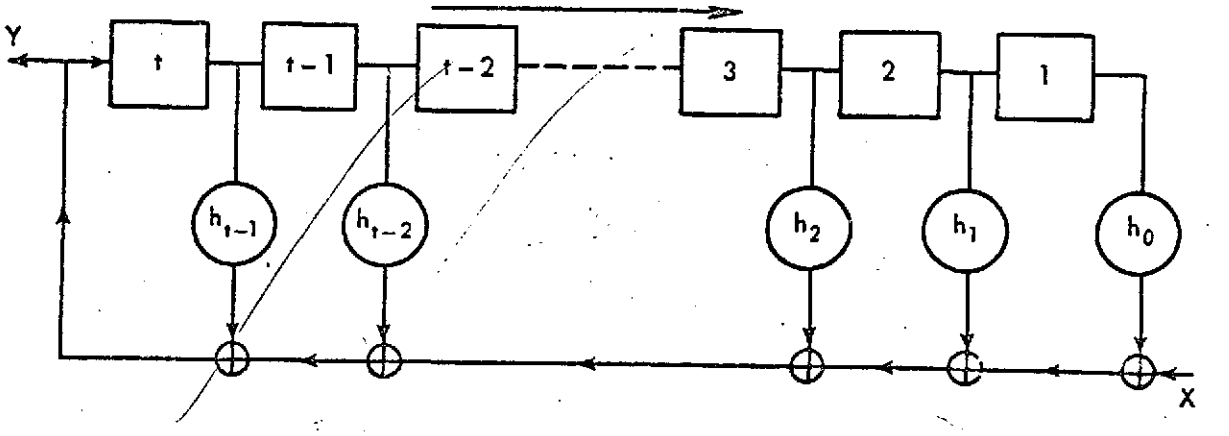


Figure F-3. SSRG

$$Y = \frac{X}{I - \sum_{i=0}^{t-1} h_i D^{t-i}} = \frac{A^t X}{A^t - \sum_{i=0}^{t-1} h_i A^i} = \frac{A^t X}{\phi(A)}$$

In general the circuit of Figure F-4 gives the following results.

$$Y = [Z] + X = \left[(h_0 D^t + h_1 D^{t-1} + \ldots h_{t-1} D)\, Y + (-g_0 D^t - g_1 D^{t-1} - \ldots g_{t-1} D)\, X\right] + X$$

$$(-h_0 D^t - h_1 D^{t-1} - \ldots h_{t-1} D + I)\, Y = (-g_0 D^t - g_1 D^{t-1} - \ldots - g_{t-1} D + I)\, X$$

$$Y = \frac{-\sum_{i=0}^{t-1} g_i D^{t-i} + I}{-\sum_{i=0}^{t-1} h_i D^{t-i} + I} X = \frac{-\sum_{i=0}^{t-1} g_i A^i + A^t}{-\sum_{i=0}^{t-1} h_i A^i + A^t} X = \frac{\phi_g(A) X}{\phi_h(A)}$$

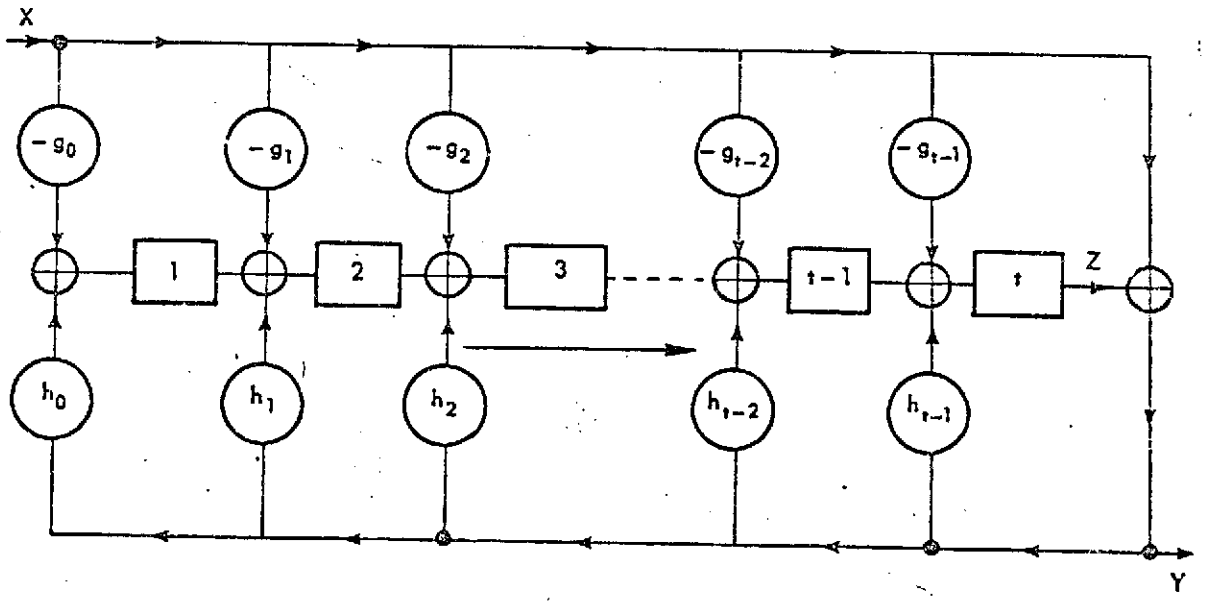


Figure F-4