Chapter 2
Finite Field and Linear Block Codes

References:

Lin, S. and Costello Jr. D.J. , Error Control Coding , Pearson Prentice Hall , 2004.

Castineira, J., and Farrel, P.G. , Essential of Error-Control Coding , Wiley, 2006

2.1 Finite Fields

- **A field is a set of elements F for which addition, multipication, subtraction and division performed with its elements result in another elements of the same set.Addition and multiplication operations satisfy the commutative and distributive laws.**

  For example , the system of real number is a field , called real number field .

- **A field with only a finite number of elements is called a finite field. Finite fields are also known as Galois fields after their inventor.**

- **The set { 0, 1 } together with moduo-2 addition and multiplication is called binary field , denoted GF(2) .**

  Binary field is a special case of finite field with two elements..

- **Most of the popular linear block codes , such as Hamming codes , BCH codes and Reed-Solomon codes, are constructed over the finite fields.**

2.1.1 Binary Irreducible Polynomials

- **A polynomial with coefficients from the binary field GF(2) is called a binary polynomial.**
- **A binary polynomial of degree *m* is called p(x) is called irreducible if it is not divisible by any binary polynomial of degree less than *m* and greater than zero .**

  For example , $1 + x + x^2$, $1 + x + x^3$, and $1 + x + x^5$ are irreducible polynomials.
- **For any positive integer $m \geqq 1$, there exists at least one irreducible polynomial of degree *m*.**
- **An irreducible polynomial p(x) of degree *m* is said to be primitive if the smallest positive integer *n* for which p(x) divides $x^n + 1$ is $n = 2^m - 1$.**

- **For any positive integer *m*, there exists a primitive polynomial of degree *m* .**
- **Example**

M	Primitive Polynomial *p(x)*
2	$1 + x + x^2$
3	$1 + x + x^3$
4	$1 + x + x^4$
5	$1 + x^2 + x^5$
6	$1 + x + x^6$
7	$1 + x^3 + x^7$

2.1.2 Construction of GF(2^m)

- For any positive integer $m \geqq 1$, there exists a Galois field of 2^m elements , denoted GF(2^m) . That is an extension field of GF(*2*).
- In general Galois field GF(2^m) can be constructed from binary field as follows.
  - (*1*) Begin with a primitive (irreducible) polynomial p(x) of degree *m* with coefficients from the binary field GF(*2*).
  - (*2*) Let α be the root of $p(x)$, i.e. $p(\alpha) = 0$
  - (*3*) Starting from GF(*2*) = { *0* ,*1*) and α , we define a multiplication operator " . " to introduce a sequence of power of *2* as follows :

$0 \cdot 0 = 0 \; ; \; 0 \cdot 1 = 1 \cdot 0 = 0 \; ;$

$1 \cdot 1 = 1 \, ,$

$0 \cdot \alpha = \alpha \cdot 0 = 0 \; ;$

$1 \cdot \alpha = \alpha \cdot 1 = \alpha \; ;$

$\alpha^2 = \alpha \cdot \alpha$

$\alpha^3 = \alpha \cdot \alpha \cdot \alpha$

.

.

.

$\alpha^j = \alpha \cdot \alpha^{j-1} \; ; \quad \alpha^i \cdot \alpha^j = \alpha^{i+j}$

We now have the following set of elements ,

$\mathbf{F = \{ 0,1 \, , \alpha \, , \alpha^2 \ldots \}}$

which is closed under multiplication “·” .

- Since α is a root of p(x) and p(x) divides $x^{2^m - 1} + 1$, α must also be a root of $x^{2^m - 1} + 1$. Hence $\alpha^{2^m - 1} + 1 = 0$

  This implies that $\alpha^{2^m - 1} = 1$

  As a result, F is finite and consists of following elements

  $$F = \{ 0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^m - 2} \}$$

- Let $\alpha^0 = 1$. Multiplication is carried out as follows.

  For $0 \leq i, j \leq 2^m - 1$,

  $$\alpha^i \bullet \alpha^j = \alpha^{i+j} = \alpha^k$$

  where k is the remainder resulting from dividing $i+j$ by $2^m - 1$.

  Since $\alpha^i \bullet \alpha^{2^m - 1 - i} = \alpha^{2^m - 1 - i}$,

  $\alpha^{2^m - 1 - i}$ is called the multiplicative inverse of α^i and vise versa.

  Also, $\alpha^{2^m - 1 - i} = \alpha^{2^m - 1} \bullet \alpha^{-i} = \alpha^{-i}$

  we can use α^{-i} to denote the multiplicative inverse of α^i. 4

- Since α is a root of p(x) and p(x) divides $x^{2^m - 1} + 1$, α must also be a root of $x^{2^m - 1} + 1$. Hence $\alpha^{2^m - 1} + 1 = 0$

  This implies that $\alpha^{2^m - 1} = 1$

  As a result, F is finite and consists of following elements

  $F = \{ 0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^m - 2} \}$

- Let $\alpha^0 = 1$. Multiplication is carried out as follows.

  For $0 \leq i, j \leq 2^m - 1$,

  $\alpha^i \bullet \alpha^j = \alpha^{i+j} = \alpha^k$

  where k is the remainder resulting from dividing $i+j$ by $2^m - 1$.

  Since $\alpha^i \bullet \alpha^{2^m - 1 - i} = \alpha^{2^m - 1}$,

  $\alpha^{2^m - 1 - i}$ is called the multiplicative inverse of α^i and vise versa.

  Also, $\alpha^{2^m - 1 - i} = \alpha^{2^m - 1} \bullet \alpha^{-i} = \alpha^{-i}$

  we can use α^{-i} to denote the multiplicative inverse of α^i.

- Since α is a root of p(x) and p(x) divides $x^{2^m - 1} + 1$, α must also be a root of $x^{2^m - 1} + 1$. Hence $\alpha^{2^m - 1} + 1 = 0$

  This implies that $\alpha^{2^m - 1} = 1$

  As a result, F is finite and consists of following elements

  $$F = \{ 0, 1, \alpha, \alpha^2, \ldots, \alpha^{2^m - 2} \}$$

- Let $\alpha^0 = 1$. Multiplication is carried out as follows.

  For $0 \leq i, j \leq 2^m - 1$,

  $$\alpha^i \bullet \alpha^j = \alpha^{i+j} = \alpha^k$$

  where k is the remainder resulting from dividing $i+j$ by $2^m - 1$.

  Since $\alpha^i \bullet \alpha^{2^m - 1 - i} = \alpha^{2^m - 1} = 1$,

  $\alpha^{2^m - 1 - i}$ is called the multiplicative inverse of α^i and vise versa.

  Also, $\alpha^{2^m - 1 - i} = \alpha^{2^m - 1} \bullet \alpha^{-i} = \alpha^{-i}$

  we can use α^{-i} to denote the multiplicative inverse of α^i.

- **Next , we define " division " operator as follows :**

  $\alpha^i \div \alpha^j = \alpha^i \cdot \alpha^{-j} = \alpha^{i-j}$

- **The ' addition " operator on F is defined as follows .**

  For $0 \leqq i \leqq 2^m - 2$ **, dividing** x^i **by** $p(x)$ **yields**

  $x^i = a(x)\, p(x) + b(x)$

  where b (x) is the remainder and

  $b(x) = b_0 + b_1 x + b_2 x^2 + \ldots + b_{m-1} x^{m-1}$

  Replacing x **by** α **, we have**

  $\alpha^i = a(\alpha)\, p(\alpha) + b(\alpha) = b_0 + b_1 \alpha + b_2 \alpha^2 + \ldots + b_{m-1} \alpha^{m-1}$

  Therefore , each nonzero element in F can be expressed as a polynomial of α with degree $m-1$ **or less.**

  The "addition" of α^i **and** α^j **is defined as**

  $\alpha^i + \alpha^j = (b_0 + d_0) + (b_1 + d_1)\alpha + \ldots + (b_{m-1} + b_{m-1}) x^{m-1}$

  where $\alpha^j = d_0 + d_1 \alpha + d_2 \alpha^2 + \ldots + d_{m-1} \alpha^{m-1} = \alpha^k$

- **Clearly , $\alpha^j + \alpha^j = 0$**

  Thus , " subtraction " is defined as follows.

  $\alpha^i - \alpha^j = \alpha^i + \alpha^j$

  Hence , subtraction is the same as addition

- **We conclude that F = { $0,1 , \alpha , \alpha^2 \ldots$ } together with the multiplication and addition operators for a field of 2^m elements .**
- **There are three forms to represent the elements in GF (2^m)**

  (1) Power form (easier to perform multiplication)

  F = { $0,1 , \alpha , \alpha^2 \ldots$ }

  (2) Polynomial form

  $\alpha^i = b_0 + b_1 \alpha + b_2 \alpha^2 + \ldots + b_{m-1} \alpha^{m-1}$

  (3) Vector form (easier to perform addition)

  $\alpha^i = (b_0 , b_1 , b_2 , \ldots , b_{m-1})$

Galois field GF (2^4) generated by $p(x) = 1 + x + x^4$

Exp. representation	Polynomial representation	Vector representation
0	0	0 0 0 0
1	1	1 0 0 0
α	α	0 1 0 0
α^2	α^2	0 0 1 0
α^3	α^3	0 0 0 1
α^4	$1 + \alpha$	1 1 0 0
α^5	$\alpha + \alpha^2$	0 1 1 0
α^6	$\alpha^2 + \alpha^3$	0 0 1 1
α^7	$1 + \alpha + \alpha^3$	1 1 0 1
α^8	$1 + \alpha^2$	1 0 1 0
α^9	$\alpha + \alpha^3$	0 1 0 1
α^{10}	$1 + \alpha + \alpha^2$	1 1 1 0
α^{11}	$\alpha + \alpha^2 + \alpha^3$	0 1 1 1
α^{12}	$1 + \alpha + \alpha^2 + \alpha^3$	1 1 1 1
α^{13}	$1 + \alpha^2 + \alpha^3$	1 0 1 1
α^{14}	$1 + \alpha^3$	1 0 0 1

2.2 Minimal Polynomials

- Consider the Galois field GF (2^m) generated by the primitive polynomial

$$p(x) = p_0 + p_1 x + p_2 x^2 + \ldots + p_{m-1} x^{m-1} + x^m$$

The element α, which is a root of $p(x)$, whose powers generate all the non-zero elements of GF (2^m) is called a primitive element of GF (2^m). Usually, there may be more than one primitive elements in a finite field GF (2^m).
For example, α^4 and α^7 are also primitive elements of GF (2^4).

- Let. β be a non-zero element of GF (2^m).
Consider the powers of β : $\beta, \beta^2, \beta^4, \beta^8 \ldots, \beta^{2^e}$.
If e is the smallest nonnegative integer for which $\beta^{2^e} = \beta$,
Then the integer " e " is called the exponent of β.
The minimal polynomial of the element is defined as

$$\phi(x) = (x + \beta)(x + \beta^2,)(x + \beta^4) \ldots (x + \beta^{2^e})$$

We assume thatβ is an element in GF(2^m) and *f*(x) is a polynomial with coefficients from GF(2).

Then for any positive integer λ ≧ 0 , β^{2^λ} **s also a root of f(x) .**

The elements β^{2^λ} **are called conjugates of β.**

Table :Minimal polynomials of the elements in GF(2^4)

Elements	Conjugates	Minimal polynomials
α	$\alpha^2, \alpha^4, \alpha^8$	$m_1(x) = 1 + x + x^4$
α^2	$\alpha^4, \alpha^8, \alpha^{16} = \alpha$	$m_2(x) = 1 + x + x^4$
α^3	$\alpha^6, \alpha^{12}, \alpha^{24} = \alpha^9$	$m_3(x) = 1 + x + x^2 + x^3 + x^4$
α^4	$\alpha^8, \alpha^{16} = \alpha, \alpha^{32} = \alpha^2$	$m_4(x) = 1 + x + x^4$
α^5	α^{10}	$m_5(x) = 1 + x + x^2$
α^6	$\alpha^{12}, \alpha^{24} = \alpha^9, \alpha^{48} = \alpha^3$	$m_6(x) = 1 + x + x^2 + x^3 + x^4$
α^7	$\alpha^{14}, \alpha^{28} = \alpha^{13}, \alpha^{56} = \alpha^{11}$	$m_7(x) = 1 + x^3 + x^4$
α^8	$\alpha^{16} = \alpha, \alpha^{32} = \alpha^2, \alpha^{64} = \alpha^4$	$m_8(x) = 1 + x + x^4$
α^9	$\alpha^{18} = \alpha^3, \alpha^{36} = \alpha^6, \alpha^{72} = \alpha^{12}$	$m_9(x) = 1 + x + x^2 + x^3 + x^4$
α^{10}	$\alpha^{20} = \alpha^5$	$m_{10}(x) = 1 + x + x^2$
α^{11}	$\alpha^{22} = \alpha^7, \alpha^{44} = \alpha^{14}, \alpha^{88} = \alpha^{13}$	$m_{11}(x) = 1 + x^3 + x^4$
α^{12}	$\alpha^{24} = \alpha^9, \alpha^{48} = \alpha^3, \alpha^{96} = \alpha^6$	$m_{12}(x) = 1 + x + x^2 + x^3 + x^4$
α^{13}	$\alpha^{26} = \alpha^{11}, \alpha^{52} = \alpha^7, \alpha^{104} = \alpha^{14}$	$m_{13}(x) = 1 + x^3 + x^4$
α^{14}	$\alpha^{28} = \alpha^{13}, \alpha^{56} = \alpha^{11}, \alpha^{112} = \alpha^7$	$m_{14}(x) = 1 + x^3 + x^4$

2.3 Linear Block Codes

2.3.1 Generation of Linear Block Codes

- Let the message $\mathbf{m} = (m_0, m_1, \ldots, m_{k-1})$ be an arbitrary k-tuple from GF (2) .

  The linear (n, k) code over GF (2) is the set of 2^k codewords in row vector form

  $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$, where $c_j \varepsilon$ GF (2)

  The generator G of the code is a k x n matrix over GF (2) .

  $\mathbf{c} = \mathbf{m} \cdot \mathbf{G}$

  The generator matrix can be expressed as

  $\mathbf{G} = [\, \mathbf{g}_1 \;\; \mathbf{g}_2 \;\; \ldots \;\; \mathbf{g}_k \,]^T$

  The rows of $\mathbf{G}$ are linearly independent since $\mathbf{G}$ is assumed to have rank k .

- For a linear block code, the vector sum of two codewords is a codeword.

- **The generator matrix of an (n ,k) linear systematic code can be expressed as**

  $$\mathbf{G} = [\, \mathbf{I}_k \;\; \mathbf{P} \,]$$

  where $\mathbf{I}_k$ is the k x k identity matrix and $\mathbf{P}$ is a k x (n-k) matrix.

- **An (n , k) linear code C can also be specified by an (n-k) x k matrix H denoted as parity –check matrix .**

  Let $\mathbf{c} = (c_0, c_1, \ldots, c_{n-1})$ be an n-tuple , then c is a codeword

  if and only if $\mathbf{c} \cdot \mathbf{H} = (0, 0, \ldots, 0)_{n-k} = 0$

  The parity-check matrix can be expressed as

  $$\mathbf{H} = [\, \mathbf{P}^T \;\; \mathbf{I}_{n-k} \,]$$

  It is noted that many solutions for H are possible for any given generator matrix G .

2.3.2 Error-Correcting Capability of Linear Block Codes

- **Hamming weight**

  Let c = (c_0, c_1,…, c_{n-1}) be an *n*-tuple codeword.

  The Hamming weight of c denoted by *w*(*c*) , is defined as the number of nonzero components of c.

- **Hamming distance**

  Let u anv are two binary n-tuple words, . The hamming distance between u and v , denoted by d (u,v) , is defined as the number of positions in which they differ from each other.

- **Example :**

  If u= (100110 1) v = (1101110)

  we have *w* (u) = 4 , *w* (v) = 5 ,

  ***d*(u,v) = 3**

- **Minimum distance of a code**

  If we compute the Hamming distances between all possible pairs of code words, the smallest value of the Hamming distance for all pairs of code words is called the minimum distance of the code.

  The minimum distance , designated by d_{min} , is defined as follows :

  $$d_{min} = \min \{ d(u,v) : u \neq v , u, v \in C \}$$

- **For a (n ,k) linear block code**

  $$d_{min} \leq n-k-1$$

  In order for an (n ,k) linear code to correct t errors , we must find a code with a minimum distance that satisfies

  $$d_{min} \geq 2t+1$$

- **Any (n,k) linear code with minimum distance $2t+1$ is called a perfect code.**

Hamming codes and Golay codes are perfect codes.

2.3.3 Hamming Codes and Golay Codes

- **Hamming codes are the first class of binary linear block code discovered by R.W. Hamming in 1950.**

For any positive integer $m \geqq 3$, there exists a Hamming code with the following parameters :

block code length $n = 2^m - 1$

message length $k = 2^m - 1 - m$

minimum Hamming distance $d_{min} = 3$

error-correction capability $t = 1$.

For a (7 ,4) Hamming code

$$G = \begin{bmatrix} 1000\,101 \\ 0100\,111 \\ 1101\,001 \\ 0001\,011 \end{bmatrix} \qquad H = \begin{bmatrix} 1110100 \\ 0111010 \\ 0010110 \end{bmatrix}$$

- **The Golay code , introduced in 1949 by the Swiss mathematician Marcel Golay , is a triple-error-correcting binary linear code (23 ,12) with $d_{min} = 7$.**

  The generator polynomial of the (23,12) Golay code is

  $$g(x) = 1 + x + x^5 + x^6 + x^7 + x^9 + x^{11}$$

- **The (23, 12) Golay code can be extended by adding an overall parity-check bit such that each codeword has even parity. This extension results in a (24,12) code with $d_{min} = 8$. This code is capable of correcting all patterns of three or fewer errors and detecting all error patterns of four errors.**

2.4 Cyclic Codes

- **An (n,k) linear code C is called a cyclic code if any cyclic shift of a codeword is another codeword .**

In polynomial form

$$c(x) = c_0 + c_1 x + c_2 x^2 + \ldots + c_{n-1} x^{n-1}$$

$$c^{(j)}(x) = c_{n-j} + c_{n-j+1} x + c_{n-j+2} x^2 + \ldots + c_{n-j-1} x^{n-1}$$

Cyclic structure makes the encoding and syndrome computation very easy.

2.4.1 Generator Polynomial

- **Every nonzero code polynomial c(x) in C must have degree at least $n-k$ but not greater than $n-1$. There is one and only one nonzero generator polynomial g(x) for a cyclic code.**

- **It can be shown that the generator polynomial g(x) of an (n ,k) cyclic code is always a polynomial factor of the polynomial**

  $x^n - 1$ **, or** $x^n + 1$ **.**

  $g(x) = 1 + g_1 x + g_2 x^2 + \ldots + g_{n-k-1} x^{n-k-1} + x^{n-k}$

  Since $g(x)$ **divides** $x^n - 1$ **, it follows that**

  $x^n - 1 = h(x)\, g(x)$

  where $h(x) = h_0 + h_1 x + h_2 x^2 + \ldots + h_k x^k$

  and $h_0 = h_k = 1$

  h(x) is called the parity polynomial of the (n , k) cyclic code.

- **The message polynomial is expressed as**

  $m(x) = m_0 + m_1 x + m_2 x^2 + \ldots + m_{k-1} x^{k-1}$

  Then , the product m(x)g(x) is the polynomial representing the code word polynomial of degree n-1 or less.

In general, c(x) and $c^{(j)}(x)$ are related by the formula

$$c^{(j)}(x) = x^j c(x) \bmod (x^n - 1)$$

We can see that

$$c^{(j)}(x) = x^j m(x)g(x) \bmod (x^n - 1) = m^j(x)\, g(x)$$

2.4.2 Encoding of Cyclic Codes

- **Consider an (n ,k) cyclic code with generator polynomial g(x)**
  Suppose m = (m_0, m_1,…, m_{k-1}) is the message to be encoded.

$$m(x) = m_0 + m_1 x + m_2 x^2 + \ldots + m_{k-1} x^{k-1}$$

Multiplying m(x) by x^{n-k} and the dividing by g(x) , we obtain

$$x^{n-k}\, m(x) = q(x)\, g(x) + p(x)$$

where $p(x) = p_0 + p_1 x + p_2 x^2 + \ldots + p_{n-k-1} x^{n-k-1}$

is the remainder .

Then $p(x) + x^{n-k} m(x) = q(x)g(x)$ **is a multiple of g(x) and has degree** $n-1$**. Hence it is the code polynomial for the message.**

- **Note that**

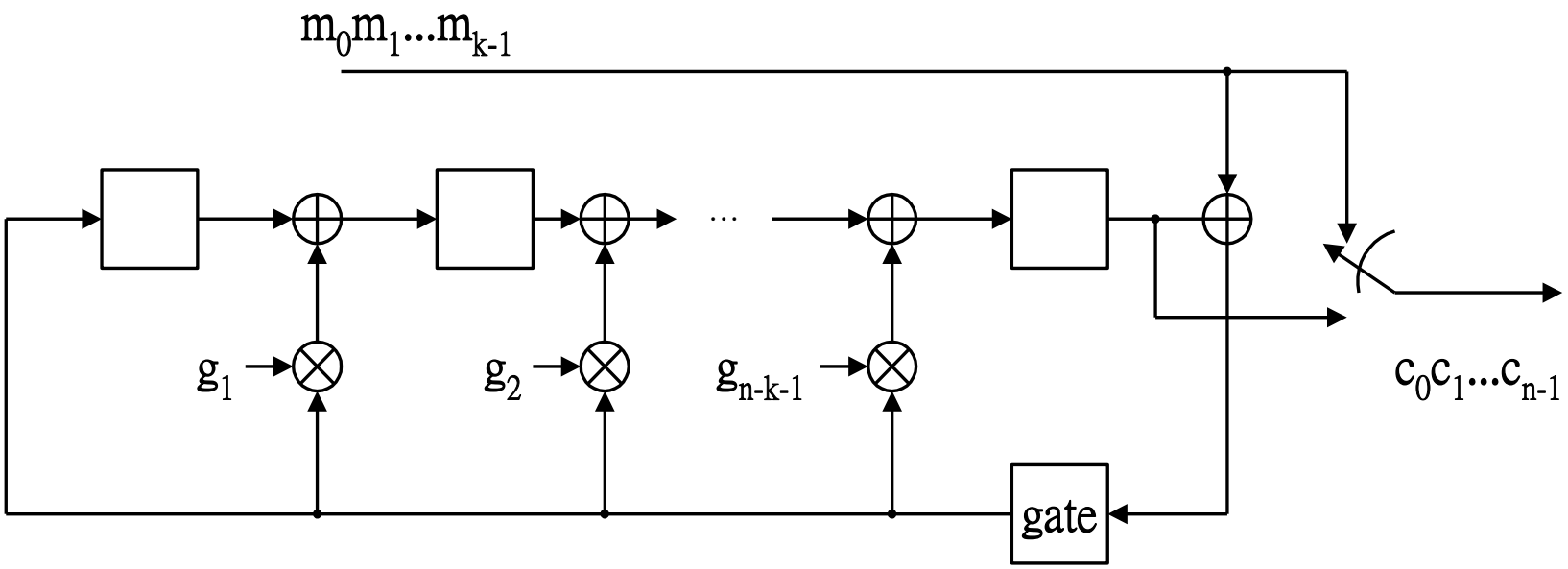
$$p(x) + x^{n-k} m(x)$$

$$= p_0 + p_1 x + p_2 x^2 + \ldots + p_{n-k-1} x^{n-k-1} + m_0 x^{n-k} + m_1 x^{n-k+1} + \ldots + m_{k-1} x^{k-1}$$

The code polynomial is in systematic form where p(x) is the parity-check part .

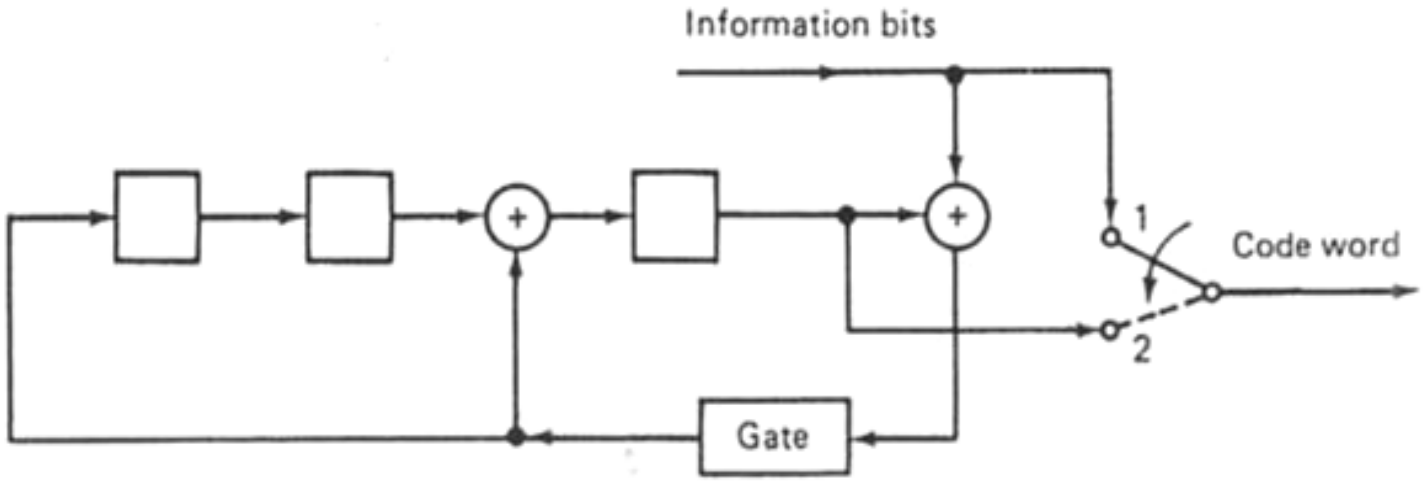
- **The encoding can be implemented by using a division circuit consisting of shift registers and feedback connections based on the generator polynomial g(x) ,as show below Fig.2.1) .**
- **In the figure the right-most symbol is the first symbol to enter the encoder. The gate is turned on until all information digits have been shifted into the circuit.**

Fig.2.1 Encoding circuit based on g(x)



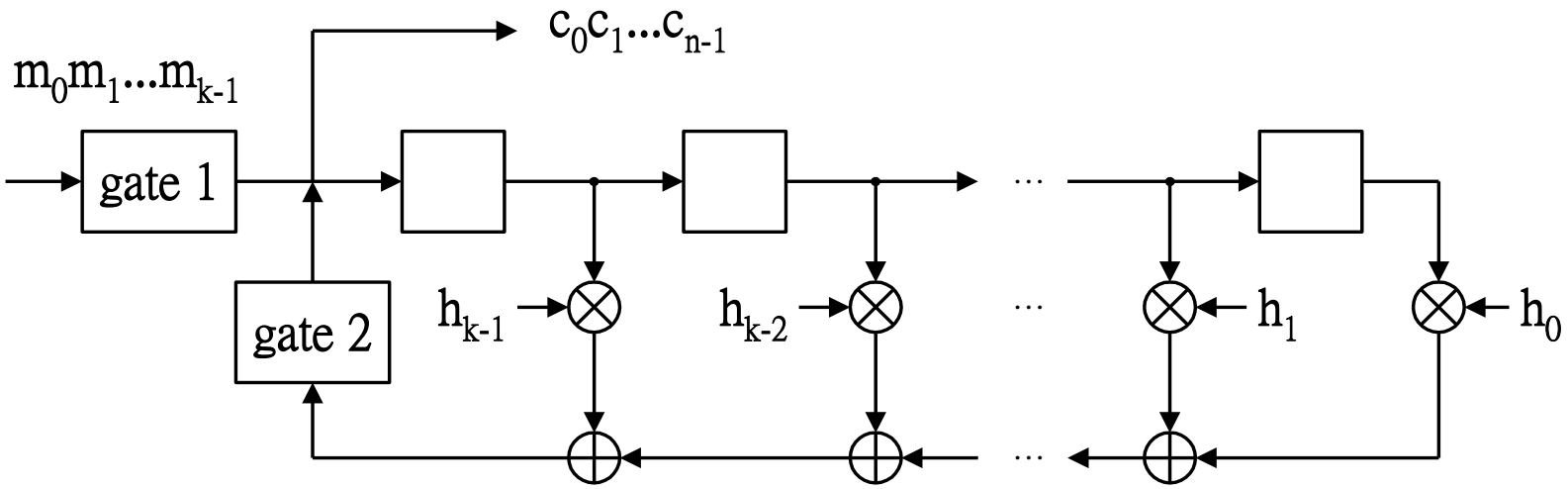
Example : Encoding of cyclic (7,4) Hamming code

$g(x) = 1+x^2+x^3$, message bits m = (1001)



Shift no. i	Gate	After ith shift: Register contents	Output
0	On	0 0 0	1
1	On	1 0 1	0 1
2	On	1 1 1	0 0 1
3	On	1 1 0	1 0 0 1
4	Off	1 1 0	0 1 0 0 1
5	Off	0 1 1	1 0 1 0 0 1
6	Off	0 0 1	1 1 0 1 0 0 1

- **It can be shown that cyclic codes can also be generated by using the parity polynomial $h(x)$, where $h(x) = h_0 + h_1 x + h_2 x^2 + \ldots + h_k x^k$.**
- **The k-stage shifter-register encoder based on h(x) is shown in Fig.2.2 .**



2.5 Syndrome Computation

- **Let c(x)and r(x) be the transmitted code polynomial and received polynomial, respectively.**

  Dividing r(x) by the generator polynomial g (x) , we have

  $$r(x) = q(x)\, g(x) + s(x)$$

  where s(x) is the remainder and

  $$s(x) = s_0 + s_1 x + s_2 x^2 + \ldots + s_{n-k-1} x^{n-k-1}$$

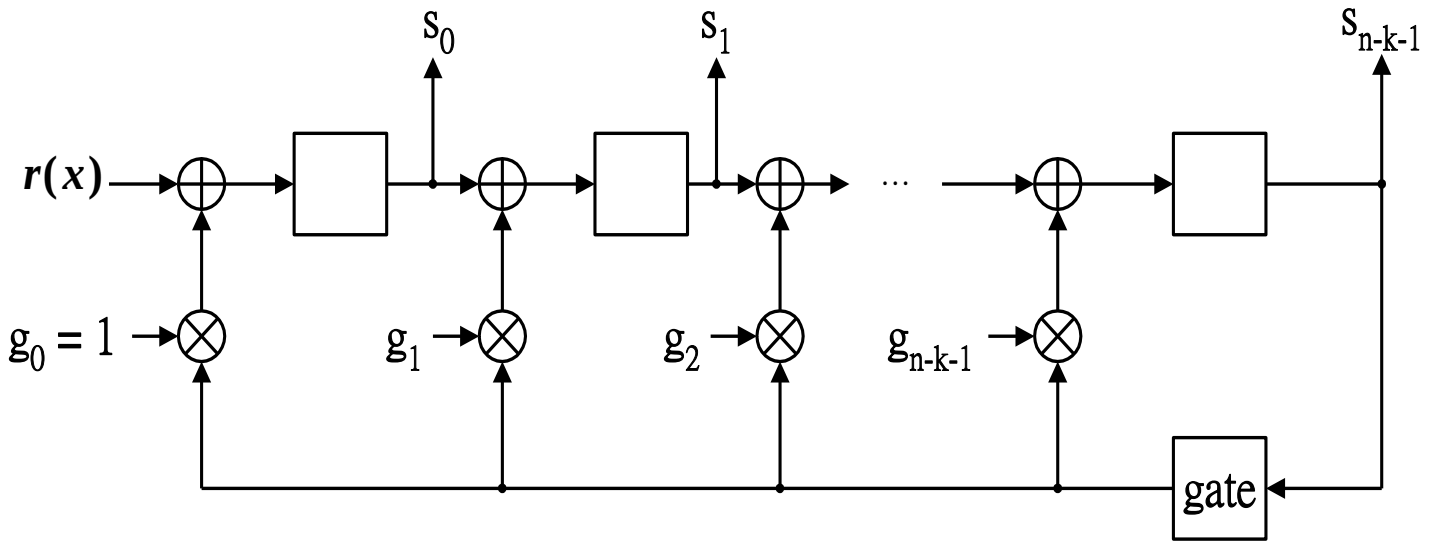
  Then s(x) is the syndrome polynomial of r(x).

  The received polynomial r(x) is a code polynomial if and only if s(x) = 0.

- **Syndrome computation can be done by a division circuit shown in Fig.2.3 .**

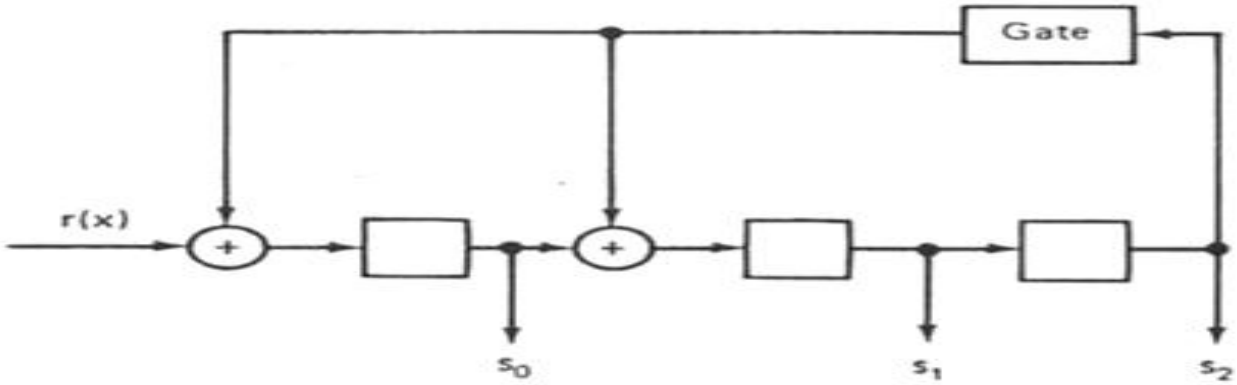
  As soon as the entire r(x) has been shifted into the register , the contents in the register form the s(x) .

Fig. 2.3 Syndrome Computation Circuit



Example : Syndrome circuit for a (7,4) cyclic code with $g(x) = 1 + x + x^3$

Received sequence r = (1001000)



Shift no.	Input	Register contents s_0, s_1, s_2
0	—	0 0 0
1	0	0 0 0
2	0	0 0 0
3	1	0 0 0
4	0	1 0 0
5	0	0 1 0
6	1	0 0 1
7	0	1 0 0
8	0	0 1 0
9	0	0 0 1
10	0	1 0 0
11	0	0 1 0
12	0	0 0 1

- **Since $r(x) = c(x) + e(x)$**

  and also $r(x) = q(x)\,g(x) + s(x)$

  we have $e(x) = r(x) + c(x)$

  $$= q(x)g(x) + s(x) + c(x)$$

  $$= q(x)\,g(x) + s(x) + m(x)\,g(x)$$

  $$= [\,q(x) + m(x)\,]\,g(x) + s(x)$$

  or $s(x) = e(x) \bmod g(x)$

  Hence the syndrome polynomial s(x) is also the remainder that results from dividing *e*(x) by g(x) .

Table 2.1

Galois field $GF(2^5)$ constructed by using the primitive polynomial $p(x) = 1 + x^2 + x^5$

Field element (polynomial notation)	5-tuple representation
0	0 0 0 0 0
1	1 0 0 0 0
α^1	0 1 0 0 0
α^2	0 0 1 0 0
α^3	0 0 0 1 0
α^4	0 0 0 0 1
$\alpha^5 = 1 + \alpha^2$	1 0 1 0 0
$\alpha^6 = \alpha + \alpha^3$	0 1 0 1 0
$\alpha^7 = \alpha^2 + \alpha^4$	0 0 1 0 1
$\alpha^8 = 1 + \alpha^2 + \alpha^3$	1 0 1 1 0
$\alpha^9 = \alpha + \alpha^3 + \alpha^4$	0 1 0 1 1
$\alpha^{10} = 1 + \alpha^4$	1 0 0 0 1
$\alpha^{11} = 1 + \alpha + \alpha^2$	1 1 1 0 0
$\alpha^{12} = \alpha + \alpha^2 + \alpha^3$	0 1 1 1 0
$\alpha^{13} = \alpha^2 + \alpha^3 + \alpha^4$	0 0 1 1 1
$\alpha^{14} = 1 + \alpha^2 + \alpha^3 + \alpha^4$	1 0 1 1 1
$\alpha^{15} = 1 + \alpha + \alpha^2 + \alpha^3 + \alpha^4$	1 1 1 1 1
$\alpha^{16} = 1 + \alpha + \alpha^3 + \alpha^4$	1 1 0 1 1
$\alpha^{17} = 1 + \alpha + \alpha^4$	1 1 0 0 1
$\alpha^{18} = 1 + \alpha$	1 1 0 0 0
$\alpha^{19} = \alpha + \alpha^2$	0 1 1 0 0
$\alpha^{20} = \alpha^2 + \alpha^3$	0 0 1 1 0
$\alpha^{21} = \alpha^3 + \alpha^4$	0 0 0 1 1
$\alpha^{22} = 1 + \alpha^2 + \alpha^4$	1 0 1 0 1
$\alpha^{23} = 1 + \alpha + \alpha^2 + \alpha^3$	1 1 1 1 0
$\alpha^{24} = \alpha + \alpha^2 + \alpha^3 + \alpha^4$	0 1 1 1 1
$\alpha^{25} = 1 + \alpha^3 + \alpha^4$	1 0 0 1 1
$\alpha^{26} = 1 + \alpha + \alpha^2 + \alpha^4$	1 1 1 0 1
$\alpha^{27} = 1 + \alpha + \alpha^3$	1 1 0 1 0
$\alpha^{28} = \alpha + \alpha^2 + \alpha^4$	0 1 1 0 1
$\alpha^{29} = 1 + \alpha^3$	1 0 0 1 0
$\alpha^{30} = \alpha + \alpha^4$	0 1 0 0 1

Table 2.2 Minimal polynomials of the elements in GF(2^6)

Elements	Minimal polynomials
$\alpha, \alpha^2, \alpha^4, \alpha^{16}, \alpha^{32}$	$1 + X + X^6$
$\alpha^3, \alpha^6, \alpha^{12}\alpha^{24}, \alpha^{48}\alpha^{33}$	$1 + X + X^2 + X^4 + X^6$
$\alpha^5, \alpha^{10}, \alpha^{20}, \alpha^{40}, \alpha^{17}, \alpha^{34}$	$1 + X + X^2 + X^5 + X^6$
$\alpha^7, \alpha^{14}, \alpha^{28}, \alpha^{56}, \alpha^{49}, \alpha^{35}$	$1 + X^3 + X^6$
$\alpha^9, \alpha^{18}, \alpha^{36}$	$1 + X^2 + X^3$
$\alpha^{11}, \alpha^{22}, \alpha^{44}, \alpha^{25}, \alpha^{50}, \alpha^{37}$	$1 + X^2 + X^3 + X^5 + X^6$
$\alpha^{13}, \alpha^{26}, \alpha^{52}, \alpha^{41}, \alpha^{19}, \alpha^{38}$	$1 + X + X^3 + X^4 + X^6$
$\alpha^{15}, \alpha^{30}, \alpha^{60}, \alpha^{57}, \alpha^{51}, \alpha^{39}$	$1 + X^2 + X^4 + X^5 + X^6$
α^{21}, α^{42}	$1 + X + X^2$
$\alpha^{23}, \alpha^{46}, \alpha^{29}, \alpha^{58}, \alpha^{53}, \alpha^{43}$	$1 + X + X^4 + X^5 + X^6$
$\alpha^{27}, \alpha^{54}, \alpha^{45}$	$1 + X + X^3$
$\alpha^{31}, \alpha^{62}, \alpha^{61}, \alpha^{59}, \alpha^{55}, \alpha^{47}$	$1 + X^5 + X^6$

Table 2.3

Galois field GF(2^6) constructed by using the primitive polynomial $p(x) = 1 + x + x^6$

0	0									(0 0 0 0 0 0)
1	1									(1 0 0 0 0 0)
α			α							(0 1 0 0 0 0)
α^2					α^2					(0 0 1 0 0 0)
α^3							α^3			(0 0 0 1 0 0)
α^4								α^4		(0 0 0 0 1 0)
α^5									α^5	(0 0 0 0 0 1)
α^6	1	+	α							(1 1 0 0 0 0)
α^7			α	+	α^2					(0 1 1 0 0 0)
α^8					α^2	+	α^3			(0 0 1 1 0 0)
α^9							α^3	$+ \alpha^4$		(0 0 0 1 1 0)
α^{10}								α^4	+ α^5	(0 0 0 0 1 1)
α^{11}	1	+	α						+ α^5	(1 1 0 0 0 1)
α^{12}	1			+	α^2					(1 0 1 0 0 0)
α^{13}			α				α^3			(0 1 0 1 0 0)
α^{14}					α^2			$+ \alpha^4$		(0 0 1 0 1 0)
α^{15}							α^3		+ α^5	(0 0 0 1 0 1)
α^{16}	1	+	α					$+ \alpha^4$		(1 1 0 0 1 0)
α^{17}			α	+	α^2				+ α^5	(0 1 1 0 0 1)
α^{18}	1	+	α	+	α^2	+	α^3			(1 1 1 1 0 0)
α^{19}			α	+	α^2	+	α^3	$+ \alpha^4$		(0 1 1 1 1 0)
α^{20}					α^2	+	α^3	$+ \alpha^4$	+ α^5	(0 0 1 1 1 1)

TABLE 6.2: (*continued*)

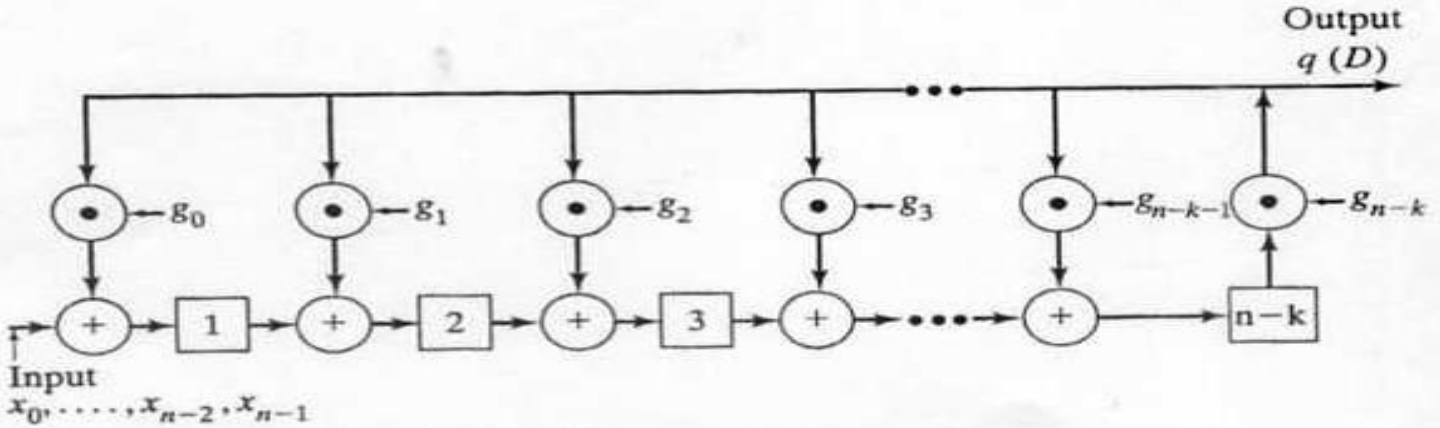
α^{21}	1	+	α			+	α^3	+ α^4	+	α^5			(1 1 0 1 1 1)
α^{22}	1			+	α^2			+ α^4	+	α^5			(1 0 1 0 1 1)
α^{23}	1					+	α^3		+	α^5			(1 0 0 1 0 1)
α^{24}	1							+ α^4					(1 0 0 0 1 0)
α^{25}			α						+	α^5			(0 1 0 0 0 1)
α^{26}	1	+	α	+	α^2								(1 1 1 0 0 0)
α^{27}			α	+	α^2	+	α^3						(0 1 1 1 0 0)
α^{28}					α^2	+	α^3	+ α^4					(0 0 1 1 1 0)
α^{29}							α^3	+ α^4	+	α^5			(0 0 0 1 1 1)
α^{30}	1	+	α										(1 1 0 0 1 1)
α^{31}	1			+	α^2					+	α^5		(1 0 1 0 0 1)
α^{32}	1					+	α^3						(1 0 0 1 0 0)
α^{33}			α						α^4				(0 1 0 0 1 0)
α^{34}					α^2					+	α^5		(0 0 1 0 0 1)
α^{35}	1	+	α			+	α^3						(1 1 0 1 0 0)
α^{36}			α	+	α^2			+	α^4				(0 1 1 0 1 0)
α^{37}					α^2		α^3			+	α^5		(0 0 1 1 0 1)
α^{38}	1	+	α			+	α^3	+	α^4				(1 1 0 1 1 0)
α^{39}			α	+	α^2			+	α^4	+	α^5		(0 1 1 0 1 1)
α^{40}	1	+	α	+	α^2	+	α^3			+	α^5		(1 1 1 1 0 1)
α^{41}	1			+	α^2	+	α^3	+	α^4				(1 0 1 1 1 0)
α^{42}			α			+	α^3	+	α^4	+	α^5		(0 1 0 1 1 1)
α^{43}	1	+	α	+	α^2			+	α^4	+	α^5		(1 1 1 0 1 1)
α^{44}	1			+	α^2	+	α^3			+	α^5		(1 0 1 1 0 1)
α^{45}	1					+	α^3	+	α^4				(1 0 0 1 1 0)
α^{46}			α					+	α^4	+	α^5		(0 1 0 0 1 1)
α^{47}	1	+	α	+	α^2					+	α^5		(1 1 1 0 0 1)
α^{48}	1			+	α^2	+	α^3						(1 0 1 1 0 0)
α^{49}			α			+	α^3	+	α^4				(0 1 0 1 1 0)
α^{50}					α^2			+	α^4		α^5		(0 0 1 0 1 1)
α^{51}	1	+	α			+	α^3			+	α^5		(1 1 0 1 0 1)
α^{52}	1			+	α^2			+	α^4				(1 0 1 0 1 0)
α^{53}			α			+	α^3			+	α^5		(0 1 0 1 0 1)
α^{54}	1	+	α	+	α^2			+	α^4				(1 1 1 0 1 0)
α^{55}			α	+	α^2	+	α^3			+	α^5		(0 1 1 1 0 1)
α^{56}	1	+	α	+	α^2	+	α^3	+	α^4				(1 1 1 1 1 0)
α^{57}			α	+	α^2	+	α^3	+	α^4	+	α^5		(0 1 1 1 1 1)
α^{58}	1	+	α	+	α^2	+	α^3	+	α^4	+	α^5		(1 1 1 1 1 1)
α^{59}	1			+	α^2	+	α^3	+	α^4	+	α^5		(1 0 1 1 1 1)
α^{60}	1					+	α^3	+	α^4	+	α^5		(1 0 0 1 1 1)
α^{61}	1							+	α^4	+	α^5		(1 0 0 0 1 1)
α^{62}	1									+	α^5		(1 0 0 0 0 1)

$\alpha^{63} = 1$

Appen.: Division circuit for dividing X(D) by G(D)

$X(D) = x_0 + x_1 D + x_2 D^2 + \ldots + x_{n-1} D^{n-1}$

$G(D) = g_0 + g_1 D + g_2 D^2 + \ldots + g_{n-k} D^{n-k}$



1) Input high order coefficients first

2) First output is coefficient of D^{n-1} of quotient (always equal to zero but mentioned here to associate outputs with correct power of D in quotient) and is present before first shift register clock pulse

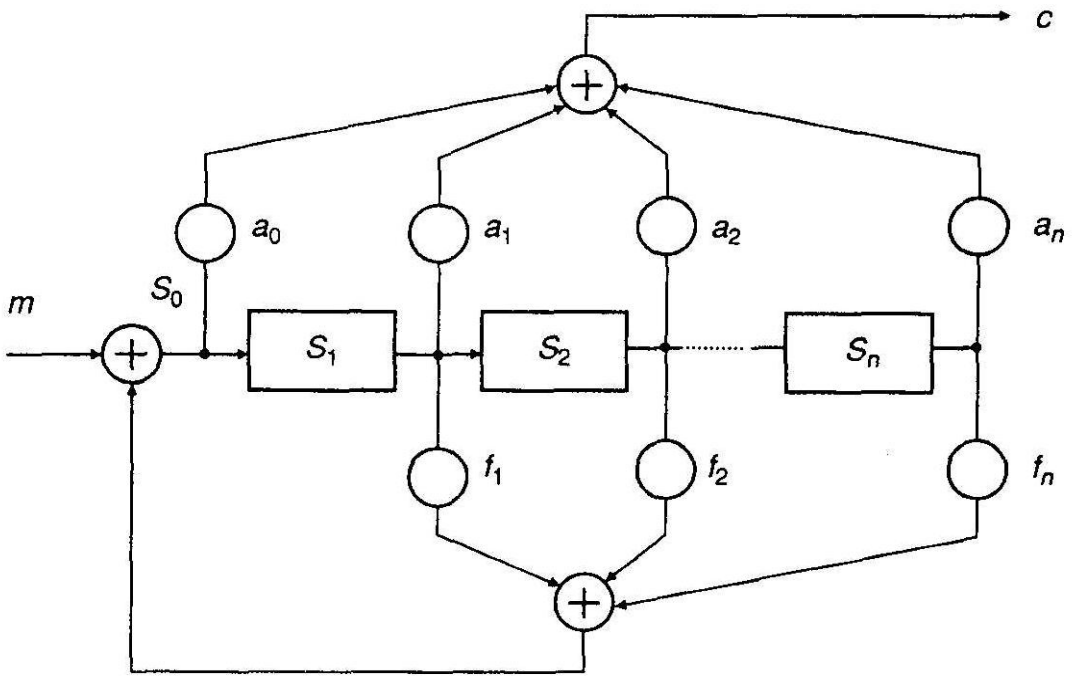
3) First non- zero output occurs after $(n-k)^{th}$ clock pulse and is coefficient of D^{n-k} in quotient

4) Last term of quotient appears at output after $(n-1)^{th}$ clock pulse and is coefficient of D^0 in quotient

5) Shift register contains coefficients of remainder $r(D) = r_0 + r_1 D + \ldots r_{n-k-1} D^{n-k-1}$ from left to right after n^{th} clock pulse

Divider circuit using linear feedback shift register structure

$$G(D) = \frac{C(D)}{M(D)} = \frac{a_0 + a_1 D + a_2 D^2 + \cdots + a_n D^n}{1 + f_1 D + f_2 D^2 + \cdots + f_n D^n} \tag{28}$$



Binary primitive polynomials of degree m

m	minimal polynomial
2	x^2+x+1
3	x^3+x+1 , x^3+x^2+1
4	x^4+x+1 , x^4+x^3+1 , $x^4+x^3+x^2+x+1$
5	x^5+x^2+1 , x^5+x^2+1 , $x^5+x^4+x^3+x^2+1$
6	x^6+x+1 , x^6+x^3+1 , x^6+x^5+1 ,
7	x^7+x^3+1
8	$x^8+x^4+x^3+x^2+1$
9	x^9+x^4+1
10	$x^{10}+x^3+1$