Cyclic Codes[1]

Manjunatha. P

manjup.jnnce@gmail.com

Professor

Dept. of ECE

J.N.N. College of Engineering, Shimoga

October 17, 2014

1 Description of Cyclic Codes

1. Description of Cyclic Codes
2. Generator and Parity-Check Matrices of Cyclic Codes

1. Description of Cyclic Codes
2. Generator and Parity-Check Matrices of Cyclic Codes
3. Encoding of Cyclic Codes

1. Description of Cyclic Codes
2. Generator and Parity-Check Matrices of Cyclic Codes
3. Encoding of Cyclic Codes
4. Syndrome Computation and Error Detection

1. Description of Cyclic Codes
2. Generator and Parity-Check Matrices of Cyclic Codes
3. Encoding of Cyclic Codes
4. Syndrome Computation and Error Detection
5. Decoding of Cyclic Codes

1. Description of Cyclic Codes
2. Generator and Parity-Check Matrices of Cyclic Codes
3. Encoding of Cyclic Codes
4. Syndrome Computation and Error Detection
5. Decoding of Cyclic Codes
6. Cyclic Hamming Codes

1. Description of Cyclic Codes
2. Generator and Parity-Check Matrices of Cyclic Codes
3. Encoding of Cyclic Codes
4. Syndrome Computation and Error Detection
5. Decoding of Cyclic Codes
6. Cyclic Hamming Codes
7. Shortened Cyclic Codes

1. Description of Cyclic Codes
2. Generator and Parity-Check Matrices of Cyclic Codes
3. Encoding of Cyclic Codes
4. Syndrome Computation and Error Detection
5. Decoding of Cyclic Codes
6. Cyclic Hamming Codes
7. Shortened Cyclic Codes
8. The (23, 12) Golay code

1. Description of Cyclic Codes
2. Generator and Parity-Check Matrices of Cyclic Codes
3. Encoding of Cyclic Codes
4. Syndrome Computation and Error Detection
5. Decoding of Cyclic Codes
6. Cyclic Hamming Codes
7. Shortened Cyclic Codes
8. The (23, 12) Golay code

Description of Cyclic Codes

- Cyclic codes form an important subclass of linear codes.

- Cyclic codes form an important subclass of linear codes.
- These codes are attractive for two reasons:

- Cyclic codes form an important subclass of linear codes.
- These codes are attractive for two reasons:
  1. Encoding and syndrome computation can be implemented easily by employing shift registers with feedback connections (or linear sequential circuits).

- Cyclic codes form an important subclass of linear codes.
- These codes are attractive for two reasons:
  1. Encoding and syndrome computation can be implemented easily by employing shift registers with feedback connections (or linear sequential circuits).
  2. They have considerable inherent algebraic structure, it is possible to find various practical methods for decoding them.

- Cyclic codes form an important subclass of linear codes.
- These codes are attractive for two reasons:
  1. Encoding and syndrome computation can be implemented easily by employing shift registers with feedback connections (or linear sequential circuits).
  2. They have considerable inherent algebraic structure, it is possible to find various practical methods for decoding them.
- Cyclic codes were first studied by Eugene Prange in 1957.

- If the n-tuple $v = (v_0, v_1, \ldots v_{n-1})$ are cyclically shifted one place to the right, we obtain another n-tuple

- If the n-tuple $v = (v_0, v_1, \ldots v_{n-1})$ are cyclically shifted one place to the right, we obtain another n-tuple

$$v^{(1)} = (v_{n-1}, v_0, v_1, \ldots v_{n-2})$$

- If the n-tuple $v = (v_0, v_1, \ldots v_{n-1})$ are cyclically shifted one place to the right, we obtain another n-tuple

$$v^{(1)} = (v_{n-1}, v_0, v_1, \ldots v_{n-2})$$

- which is called a cyclic shift of v
- If the components v are cyclically shifted i places to the right

- If the n-tuple $v = (v_0, v_1, \ldots v_{n-1})$ are cyclically shifted one place to the right, we obtain another n-tuple

$$v^{(1)} = (v_{n-1}, v_0, v_1, \ldots v_{n-2})$$

- which is called a cyclic shift of v
- If the components v are cyclically shifted i places to the right

$$v^{(i)} = (v_{n-i}, v_{n-i+1}, \ldots, v_{n-1}, v_0, v_1, \ldots v_{n-i-1})$$

- Cyclically shifting v, i places to the right is equivalent to cyclically shifting v, $(n-i)$ place to the left.

- If the n-tuple $v = (v_0, v_1, \ldots v_{n-1})$ are cyclically shifted one place to the right, we obtain another n-tuple

$$v^{(1)} = (v_{n-1}, v_0, v_1, \ldots v_{n-2})$$

- which is called a cyclic shift of v
- If the components v are cyclically shifted i places to the right

$$v^{(i)} = (v_{n-i}, v_{n-i+1}, \ldots, v_{n-1}, v_0, v_1, \ldots v_{n-i-1})$$

- Cyclically shifting v, i places to the right is equivalent to cyclically shifting v, $(n-i)$ place to the left.
  Definition 4.1
- An (n, k) linear code C is called a cyclic code if every cyclic shift of a code vector in C is also a code vector in C

- If the n-tuple $v = (v_0, v_1, \ldots v_{n-1})$ are cyclically shifted one place to the right, we obtain another n-tuple

$$v^{(1)} = (v_{n-1}, v_0, v_1, \ldots v_{n-2})$$

- which is called a cyclic shift of v
- If the components v are cyclically shifted i places to the right

$$v^{(i)} = (v_{n-i}, v_{n-i+1}, \ldots, v_{n-1}, v_0, v_1, \ldots v_{n-i-1})$$

- Cyclically shifting v, i places to the right is equivalent to cyclically shifting v, $(n-i)$ place to the left.
  Definition 4.1
- An (n, k) linear code C is called a cyclic code if every cyclic shift of a code vector in C is also a code vector in C
- The (7, 4) linear code given in Table 4.1 is a cyclic code.

- Consider the components of a code vector $V = (v_0, v_1, \ldots v_{n-1})$ as the coefficients of a polynomial as:

- Consider the components of a code vector $V = (v_0, v_1, \ldots v_{n-1})$ as the coefficients of a polynomial as:

$$V(X) = v_0 + v_1 X + v_2 X^2 + \ldots + v_{n-1} X^{n-1}$$

- Consider the components of a code vector $V = (v_0, v_1, \ldots v_{n-1})$ as the coefficients of a polynomial as:

$$V(X) = v_0 + v_1 X + v_2 X^2 + \ldots + v_{n-1} X^{n-1}$$

- If $v_{n-1} \neq 0$, the degree of $v(X)$ is $n-1$. If $v_{n-1} = 0$, the degree of $V(X)$ is less than $n-1$

- Consider the components of a code vector $V = (v_0, v_1, \ldots v_{n-1})$ as the coefficients of a polynomial as:

$$V(X) = v_0 + v_1 X + v_2 X^2 + \ldots + v_{n-1} X^{n-1}$$

- If $v_{n-1} \neq 0$, the degree of $v(X)$ is $n-1$. If $v_{n-1} = 0$, the degree of $V(X)$ is less than $n-1$
- The code polynomial that corresponds to the code vector $v^{(i)}$ is

- Consider the components of a code vector $V = (v_0, v_1, \ldots v_{n-1})$ as the coefficients of a polynomial as:

$$V(X) = v_0 + v_1 X + v_2 X^2 + \ldots + v_{n-1} X^{n-1}$$

- If $v_{n-1} \neq 0$, the degree of $v(X)$ is $n-1$. If $v_{n-1} = 0$, the degree of $V(X)$ is less than $n-1$
- The code polynomial that corresponds to the code vector $v^{(i)}$ is

$$\begin{aligned} v^{(i)}(X) &= v_{n-i} X + v_{n-i+1} X + \ldots + v_{n-1} X^{i-1} + \\ & \quad v_0 X^i + v_1 X^{i+1} + \ldots + v_{n-i-1} X^{n-1} \end{aligned}$$

- Multiplying v(X) by X^i, we obtain

$$X^i V(X) = v_0 X^i + v_1 X^{i+1} + \ldots + v_{n-i-1} X^{n-1} + \ldots + v_{n-1} X^{n+i-1}$$

- The equation above can be manipulated into the following form :

$$\begin{aligned} X^i V(X) &= v_{n-i} + v_{n-i+1} X + \ldots + v_{n-1} X^{i-1} + v_0 X^i + \ldots + v_{n-i-1} X^{n-1} \\ & \quad + v_{n-i}(X^n + 1) + v_{n-i+1} X(X^n + 1) + \ldots + v_{n-1} X^{i-1}(X^n + 1) \\ &= q(X).(X^n + 1) + v^{(i)}(X)(5.1) \end{aligned}$$

Table: (7,4) cyclic code generated by $g(x) = 1 + X + X^3$

Messages	Code Vectors	Code Polynomials
(0000)	(0000000)	0=0.g(x)
(1000)	(1101000)	$1 + X + X^3 = 1.g(x)$
(0100)	(0110100)	$X + X^2 + X^4 = X.g(x)$
(1100)	(1011100)	$1 + X^2 + X^3 + X^4 = (1 + X).g(x)$
(0010)	(0011010)	$X^2 + X^3 + X^5 = X^2.g(x)$
(1010)	(1110010)	$1 + X + X^2 + X^5 = (1 + X^2).g(x)$
(0110)	(0101110)	$X + X^3 + X^4 + X^5 = (X + X^2).g(x)$
(1110)	(1000110)	$1 + X^4 + X^5 = (1 + X + X^2).g(x)$
(0001)	(0001101)	$X^3 + X^4 + X^6 = X^3.g(x)$
(1001)	(1100101)	$1 + X + X^4 + X^6 = (1 + X^3).g(x)$
(0101)	(0111001)	$X + X^2 + X^3 + X^6 = (X + X^3).g(x)$
(1101)	(1010001)	$1 + X^2 + X^6 = (1 + X + X^3).g(x)$
(0011)	(0010111)	$X^2 + X^4 + X^5 + X^6 = (X^2 + X^3).g(x)$
(1011)	(1111111)	$1 + X + X^2 + X^3 + X^4 + X^5 + X^6 = (1 + X^2 + X^3).g(x)$
(0111)	(0100011)	$X + X^5 + X^6 = (X + X^2 + X^3).g(x)$
(1111)	(1001011)	$1 + X^3 + X^5 + X^6 = (1 + X + X^2 + X^3).g(x)$

Theorem 5.1

- The nonzero code polynomial of minimum degree in a cyclic code C is unique.
  Proof
- Let $g(X) = g_0 + g_{r-1}X + \ldots + g_{r-1}X^{r-1} + X^r$ be a nonzero code polynomial of minimum degree in C
- Suppose that g(X) is not unique, there exists another code polynomial of degree r, say $g'(X)$

$$g'(X) = g'_0 + g'_1 X + \ldots + g'_{r-1} X^{r-1} + X^r$$

- Since C is linear,

$$g(X) + g'(X) = (g0 + g'0) + (g1 + g'1)X + \ldots + (g_{r-1} + g'_{r-1)}X^{r-1}$$

- is also a code polynomial which has degree less than r
- If $g(X) + g'(X) \neq 0$, then $g(X) + g'(X)$ is a nonzero code polynomial with degree less than the minimum degree r. This is impossible. Hence

$$g(X) + g'(X) = 0$$

- This implies that

$$g(X) = g'(X)$$

- Hence $g(X)$ is unique

Theorem 5.2

- Let $g(X) = g_0 + g_{r-1}X + \ldots + g_{r-1}X^{r-1} + X^r$ be a nonzero code polynomial of minimum degree in (n, k) cyclic code C. Then the constant term g_0 must be equal to 1
  Proof
- Suppose that $g_0 = 0$, then

$$\begin{aligned} g(X) &= g_1X + g_2X^2 + \ldots + g_{r-1}X^{r-1} + Xr \\ &= X.(g_1 + g_2X + \ldots + g_{r-1}X^{r-2} + X^{r-1}) \end{aligned}$$

- If g(X) shifted cyclically n - 1 places to the right (or 1 place to the left), then a nonzero code polynomial is,
  $g_1 + g_2X + \ldots + g_{r-1}X^{r-2} + X^{r-2}$, which has a degree less than r
- This is a contradiction to the assumption that g(X) is the nonzero code polynomial with minimum degree.

Theorem 5.3

- Let $g(X) = g_0 + g_1X + \ldots + g_{r-1}X^{r-1} + X^r$ be the nonzero code polynomial of minimum degree in an (n, k) cyclic code C. A binary polynomial of degree n - 1 or less is a code polynomial iff it is a multiple of g(X)

  Proof
- Let v(X) be a binary polynomial of degree n - 1 or less
- Suppose that v(X) is a multiple of g(X)

$$\begin{aligned} v(X) &= (a_0 + a_1X + \ldots + a_{n-r-1}X^{n-r-1})g(X) \\ &= a_0g(X) + a_1g(X) + \ldots + a_{n-r-1}X^{n-r-1}g(X) \end{aligned}$$

- Since v(X) is a linear combination of the code polynomials,
- $g(X), Xg(X), \ldots, X^{n-r-1}g(X)$, it is a code polynomial in C

Proof (cont.)

- Now, let v(X) be a code polynomial in C, dividing v(X) by g(X), we obtain

$$v(X) = a(X)g(X) + b(X)$$

- where either b(X) is identical to zero or the degree of b(X) is less than the degree of g(X)
- Rearranging the equation above, we have

$$b(X) = a(X)g(X) + v(X)$$

- Since both v(X) and a(X)g(X) are code polynomials, b(X) must be a code polynomial
- If $b(X) \neq 0$, then b(X) is a nonzero code polynomial whose degree is less than the degree of g(X)

- Recall that $g(X) = g_0 + g_1X + \ldots + g_{r-1}X^{r-1} + X^r$.
- The number of binary polynomials of degree n-1 or less that are multiples of g(X) is 2^{n-r}.

$$a(X) = a_0X^{n-r-1} + a_1X^{n-r-2} + \ldots + a_{n-r-1}$$

- It follows from Theorem 5.3 that these polynomials form all the code polynomials of the (n,k) cyclic code C.
- Since there are 2^k code polynomials in C, then 2^{n-r} must be equal to 2^k.
- As a result, we have r=n-k.

Theorem 5.4

- In an (n, k) cyclic code, there exists one and only one code polynomial of degree n- k,

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

- Every code polynomial is a multiple of g(X) and every binary polynomial of degree n - 1 or less that is a multiple of g(X) is a code polynomial
- An (n,k) cyclic code is completely specified by its nonzero code polynomial of minimum degree, g(X).
- The polynomial g(X) is called the generator polynomial of the code
- The degree of g(X) is equal to the

Theorem 5.5

- the generator polynomial g(X) of an (n, k) cyclic code is a factor of $X^n + 1$

  Proof

- Multiplying g(X) by X* results in a polynomial X*g(X) of degree n
- Dividing X*g(X) by $X6n + 1$, we obtain

$$X * g(X) = (X^n + 1) + g^{(k)}(X)(5.5)$$

- where $g^{(k)}(X)$ is the remainder
- It follows from (5.1) that $g^{(k)}(X)$ is the code polynomial obtained by shifting g(X) to the right cyclically k times g(k)(X) is a multiple of g(X), $g^{(k)}(X) = a(X)g(X)$
- From (5.5) we obtain, $X^n + 1 = \{X * + a(X)\}.g(X)$

Theorem 5.6

- If g(X) is a polynomial of degree n k and is a factor of Xn + 1, then g(X) generates an (n, k) cyclic code
  Proof
- Consider the k polynomials $g(X), Xg(X), , X^{k-1}g(X)$, which all have degree n - 1 or less
- A linear combination of these k polynomials,

$$\begin{aligned} v(X) &= a_0 g(X) + a_1 g(X) + \ldots + a_{k-1} X k-1 g(X) \\ &= (a_0 + a_1 X + \ldots + a_{k-1} X^{k-1)} g(X) \end{aligned}$$

- is also a polynomial of degree n 1 or less and is a multiple of g(X)
- There are a total of 2^k such polynomials
- They form an (n, k) linear code.

Proof (cont.)

- Let $v(X) = v_0 + v_1X + \ldots + v_{n-1}X^{n-1}$ be a code polynomial in this code
- Multiplying $v(X)$ by X, we obtain

$$X_v(X) = v_0X + v_1X^2 + \ldots + v_{n-1}X^n = v_{n-1}(X^n + 1) + (v_{n-1} + v_0X +$$

- where $v(1)(X)$ is a cyclic shift of $v(X)$
- Since both $Xv(X)$ and $(Xn + 1)$ are divisible by $g(X), v(1)(X)$ must be divisible by g(X)
- Thus $v(1)(X)$ is a multiple of g(X) and is a linear combination of $g(X), Xg(X), , Xk - 1g(X)$
- Hence, $v(1)(X)$ is also a code polynomial =¿ Cyclic Code

Example 5.1

- The polynomial $X7 + 1$ can be factored as follows :

$$X^7 + 1 = (1 + X)(1 + X + X^3)(1 + X^2 + X^3)$$

- There are two factors of degree 3; each generates a (7, 4) cyclic code
- The (7, 4) cyclic code given by Table 5.1 is generated by $g(X) = 1 + X + X^3$
- This code has minimum distance 3 and it is a single-error correcting code
- Each code polynomial is the product of a message polynomial of degree 3 or less and the generator polynomial $g(X) = 1 + X + X^3$
- Let u = (1 0 1 0) be the message to be encoded
- The corresponding message polynomial is $u(X) = 1 + X^2$
- Multiplying u(X) by g(X) results in the following code polynomial :

$$v(X) = (1 + X)(1 + X + X^3) = 1 + X + X^2 + X^5$$

- or the code vector (1 1 1 0 0 1 0)

- Suppose that the message to be encoded is $u = (u_0, u_1, \ldots, u_{k-1})$
- The corresponding message polynomial is

$$u(X) = u_0 + u_1 X + \ldots + u_{k-1} X^{k-1}$$

- Multiplying u(X) by X^{n-k}, then a polynomial is of degree n 1 or less,

$$X^{n-k} u(X) = u_0 X^{n-k} + u_1 X^{n-k-1} + \ldots + u_{k-1} X^{n-1}$$

- Dividing $X^{n-k} u(X)$ by the g(X),

$$X^{n-k} u(X) = a(X) g(X) + b(X) \qquad (5.6)$$

- where a(X) and b(X) are the quotient and the remainder respectively.
- The degree of g(X)is n-k, the degree of b(x) must be n-k-1 or less

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

- Rearrange (5.6), we obtain the following polynomial of degree n -1 or less:

$$b(X) + X^{n-k}u(X) = a(X)g(X) \quad (5.7)$$

- This polynomial is a multiple of the g(X) and therefore it is a code polynomial of the cyclic code generated by g(X)
- Writing out $b(X) + X^{n-k}u(X)$, we have

$$\begin{aligned} b(X) + X^{n-k}u(X) &= b_0 + b_1X + \ldots + b_{n-k-1}X^{n-k-1} \\ &\quad + u_0X^{n-k} + u_1X^{n-k-1} + \ldots + u_{k-1}X^{n-1} \end{aligned}$$

- which corresponds to the code vector $(b_0, b_1, , b_{n-k-1}, u_0, u_1, \ldots, u_{k-1})$
- The process above yields an (n, k) cyclic code in systematic form

- Encoding in systematic form consists of three steps:
  1. Step 1 Pre-multiply the message u(X) by X^{n-k}

- Encoding in systematic form consists of three steps:
  1. Step 1 Pre-multiply the message u(X) by X^{n-k}
  2. Step 2 Obtain the remainder b(X) from dividing $X^{n-k}u(X)$ by the generator polynomial g(X)

- Encoding in systematic form consists of three steps:
  1. Step 1 Pre-multiply the message u(X) by X^{n-k}
  2. Step 2 Obtain the remainder b(X) from dividing $X^{n-k}u(X)$ by the generator polynomial g(X)
  3. Step 3 Combine b(X) and $X^{n-k}u(X)$ to obtain the code polynomial $b(X) + X^{n-k}u(X)$

- Encoding in systematic form consists of three steps:
  1. Step 1 Pre-multiply the message u(X) by X^{n-k}
  2. Step 2 Obtain the remainder b(X) from dividing $X^{n-k}u(X)$ by the generator polynomial g(X)
  3. Step 3 Combine b(X) and $X^{n-k}u(X)$ to obtain the code polynomial $b(X) + X^{n-k}u(X)$
- Consider the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$

- Encoding in systematic form consists of three steps:
  1. Step 1 Pre-multiply the message u(X) by X^{n-k}
  2. Step 2 Obtain the remainder b(X) from dividing $X^{n-k}u(X)$ by the generator polynomial g(X)
  3. Step 3 Combine b(X) and $X^{n-k}u(X)$ to obtain the code polynomial $b(X) + X^{n-k}u(X)$
- Consider the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$
- Let $u(X) = 1 + X^3$ be the message to be encoded

- Encoding in systematic form consists of three steps:
  1. Step 1 Pre-multiply the message u(X) by X^{n-k}
  2. Step 2 Obtain the remainder b(X) from dividing $X^{n-k}u(X)$ by the generator polynomial g(X)
  3. Step 3 Combine b(X) and $X^{n-k}u(X)$ to obtain the code polynomial $b(X) + X^{n-k}u(X)$
- Consider the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$
- Let $u(X) = 1 + X^3$ be the message to be encoded
- Dividing $X^3u(X) = X^3 + X^6$ by g(X)

- Encoding in systematic form consists of three steps:
  1. Step 1 Pre-multiply the message u(X) by X^{n-k}
  2. Step 2 Obtain the remainder b(X) from dividing $X^{n-k}u(X)$ by the generator polynomial g(X)
  3. Step 3 Combine b(X) and $X^{n-k}u(X)$ to obtain the code polynomial $b(X) + X^{n-k}u(X)$
- Consider the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$
- Let $u(X) = 1 + X^3$ be the message to be encoded
- Dividing $X^3u(X) = X^3 + X^6$ by g(X)
- we obtain the remainder $b(X) = X + X^2$

- Encoding in systematic form consists of three steps:
  1. Step 1 Pre-multiply the message u(X) by X^{n-k}
  2. Step 2 Obtain the remainder b(X) from dividing $X^{n-k}u(X)$ by the generator polynomial g(X)
  3. Step 3 Combine b(X) and $X^{n-k}u(X)$ to obtain the code polynomial $b(X) + X^{n-k}u(X)$
- Consider the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$
- Let $u(X) = 1 + X^3$ be the message to be encoded
- Dividing $X^3u(X) = X^3 + X^6$ by g(X)
- we obtain the remainder $b(X) = X + X^2$
- The code polynomial is $v(X) = b(X) + X^3u(X) = X + X^2 + X^3 + X^6$

- Encoding in systematic form consists of three steps:
  1. Step 1 Pre-multiply the message u(X) by X^{n-k}
  2. Step 2 Obtain the remainder b(X) from dividing $X^{n-k}u(X)$ by the generator polynomial g(X)
  3. Step 3 Combine b(X) and $X^{n-k}u(X)$ to obtain the code polynomial $b(X) + X^{n-k}u(X)$

- Consider the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$
- Let $u(X) = 1 + X^3$ be the message to be encoded
- Dividing $X^3u(X) = X^3 + X^6$ by g(X)
- we obtain the remainder $b(X) = X + X^2$
- The code polynomial is $v(X) = b(X) + X^3u(X) = X + X^2 + X^3 + X^6$
- The corresponding code vector is v = (0 1 1 1 0 0 1)

$$
\begin{array}{r}
x^3 + x \\
x^3 + x + 1 \overline{)\, x^6 \qquad\quad x^3} \\
\qquad x^6 + x^4 + x^3 \\
\hline
\qquad\qquad x^4 \\
\qquad\quad x^4 \;\; + \;\; x^2 + x \\
\hline
\qquad\qquad\quad x^2 + x
\end{array}
$$

Table: (7,4) cyclic code in systematic form generated by $g(x) = 1 + X + X^3$

Messages	Code Vectors	Code Polynomials
(0000)	(0000000)	0=0.g(x)
(1000)	(1101000)	$1 + X + X^3 = 1.g(x)$
(0100)	(0110100)	$X + X^2 + X^4 = X.g(x)$
(1100)	(1011100)	$1 + X^2 + X^3 + X^4 = (1 + X).g(x)$
(0010)	(1110010)	$1 + X + X^2 + X^5 = X^2.g(x)$
(1010)	(0011010)	$X^2 + X^3 + X^5 = (1 + X^2).g(x)$
(0110)	(1000110)	$1 + X^4 + X^5 = (X + X^2).g(x)$
(1110)	(0101110)	$X + X^3 + X^4 = (1 + X + X^2).g(x)$
(0001)	(1010001)	$1 + X^2 + X^6 = X^3.g(x)$
(1001)	(0111001)	$X + X^2 + X^3 + X^6 = (1 + X^3).g(x)$
(0101)	(1100101)	$1 + X + X^4 + X^6 = (X + X^3).g(x)$
(1101)	(0001101)	$X^3 + X^4 + X^6 = (1 + X + X^3).g(x)$
(0011)	(0100011)	$X + X^5 + X^6 = (X^2 + X^3).g(x)$
(1011)	(1001011)	$1 + X + X^3 + X^5 + X^6 = (1 + X^2 + X^3).g(x)$
(0111)	(0010111)	$X^2 + X^4 + X^5 + X^6 = (X + X^2 + X^3).g(x)$
(1111)	(1111111)	$1 + X + X^2 + X^3 + X^4 + X^5 + X^6 = (1 + X + X^2 + X^3).g(x)$

Generator and Parity-Check Matrices of Cyclic Codes

Generator and Parity-Check Matrices of Cyclic Codes

Generator and Parity-Check Matrices of Cyclic Codes

- Consider an (n, k) cyclic code C with generator polynomial $g(X) = g_0 + g_1X + \ldots + g_{n-k}X^{n-k}$
- The k code polynomials $g(X), Xg(X), \ldots, X^{k-1}g(X)$ span C
- If the n-tuples corresponding to these k code polynomials are used as the rows of an k x n matrix, then the generator matrix C is:

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & . & . & . & . & g_{n-k} & 0 & 0 & 0 & . & . & 0 \\ 0 & g_0 & g_1 & g_2 & . & . & . & . & g_{n-k} & 0 & 0 & . & . & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & . & . & . & . & g_{n-k} & 0 & . & . & 0 \\ \vdots & & & & & & & & & & & & & \\ 0 & 0 & . & . & . & 0 & g_0 & g_1 & g_2 & . & . & . & . & g_{n-k} \end{bmatrix}$$

Note: $g_0 = g_{n-k} = 1$

Generator and Parity-Check Matrices of Cyclic Codes

- For example, the (7, 4) cyclic code with generator polynomial $g(X) = 1 + X + X^3$ has the following generator matrix:

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

- G is not in systematic form,(Parity and Identity matrix)
- G can be put into systematic form with row operations. (Add 1st row to the 3rd row, and add the sum of the first two rows to the 4th row)

$$G' = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

- This matrix generates the same code as G

- 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 $n = 2^m - 1$ i.e $X^{2^m-1} + 1$
- Check that if $X^3 + X + 1$ is an irreducible polynomial over GF(2)?

Generator and Parity-Check Matrices of Cyclic Codes

- 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 $n = 2^m - 1$ i.e $X^{2^m-1} + 1$
- Check that if $X^3 + X + 1$ is an irreducible polynomial over GF(2)? Solution:
- m=3 Therefore $X^{2^3-1} + 1 = X^7 + 1$

$$
\begin{array}{r}
x^4 + x^2 + x + 1 \\
x^3 + x + 1 \overline{)\, x^7 \qquad\quad + 1} \\
\underline{x^7 + x^5 + x^4} \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
x^5 + x^4 \quad + 1 \\
\underline{x^5 \;+\; x^3 + x^2} \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
x^4 + x^3 + x^2 + 1 \\
\underline{x^4 + \quad + x^2 + x} \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
x^3 + x + 1 \\
\underline{x^3 + x + 1} \\
\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \\
0
\end{array}
$$

Generator and Parity-Check Matrices of Cyclic Codes

- The generator polynomial g(X) is a factor of $X^n + 1$,

$$X^n + 1 = g(X)h(X) \tag{5.10}$$

- where the h(X) has the degree k and is of the following form:

$$h(X) = h_0 + h_1 X + \ldots + h_k X^k$$

- with $h_0 = h_k = 1$
- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a code vector in C.
- Then $v(X) = a(X)g(X)$. Multiplying v(X) by h(X), we obtain

$$\begin{aligned} v(X)h(X) &= a(X)g(X)h(X) \\ &= a(X)(X^n + 1) \\ &= a(X) + X^n a(X) \end{aligned} \tag{5.11}$$

Example: For message 1011 and $g(x) = 1 + x + x^3$ $a(X) = 1 + x^2 + X^3$ and $X^n a(X) = X^7 + x^8 + X^{10}$

Generator and Parity-Check Matrices of Cyclic Codes

- Since the degree of a(X) is k - 1 or less, the powers $X^k, X^{k+1}, \ldots, X^{n-1}$ do not appear in $a(X) + X^n a(X)$
- If we expand the product v(X)h(X) on the left-hand side of (5.11), the coefficients of $X^k, X^{k+1}, \ldots, X^{n-1}$ must be equal to 0 we obtain the following n-k equalities :

$$\sum_{i=0}^{k} h_i v_{n-i-j} = 0 \quad for \qquad 1 \leq j \leq n-k \qquad (5.12)$$

$$h_0 v_{n-j} + h_1 v_{n-1-j} + h_2 v_{n-2-j} + h_k v_{n-k-j} = 0 \quad for \quad 1 \leq j \leq n-k$$

Generator and Parity-Check Matrices of Cyclic Codes

- The reciprocal of h(X), which is defined as follows:

$$X^k h(X^{-1}) = h_k + h_{k-1}X + \ldots + h_0 X^k \tag{5.13}$$

- $X^k h(X^{-1})$ is a factor of $X^n + 1$
- The polynomial $X^k h(X^{-1})$ generates an (n, n - k) cyclic code with the following (n - k) x n matrix as a generator matrix:

$$H = \begin{bmatrix} h_k & h_{k-1} & h_{k-2} & . & . & . & . & h_0 & 0 & . & . & . & . & 0 \\ 0 & h_k & h_{k-1} & h_{k-2} & . & . & . & . & h_0 & 0 & 0 & . & . & 0 \\ 0 & 0 & h_k & h_{k-1} & h_{k-2} & . & . & . & . & h_0 & 0 & . & . & 0 \\ \vdots & & & & & & & & & & & & & \\ 0 & 0 & . & . & . & 0 & h_k & h_{k-1} & h_{k-2} & . & . & . & . & h_0 \end{bmatrix} \tag{5.14}$$

Generator and Parity-Check Matrices of Cyclic Codes

- It follows from the n-k equalities of (5.12) that any code vector v in C is orthogonal to every row of H
- H is a parity-check matrix of the cyclic code C and the row space of H is the dual code of C
- Since the parity-check matrix H is obtained from the polynomial h(X), we call h(X) the parity polynomial of C
- A cyclic code is also uniquely specified by the parity polynomial

Generator and Parity-Check Matrices of Cyclic Codes

Theorem 5.7

- let C be an (n, k) cyclic code with generator polynomial g(X). The dual code of C is also cyclic and is generated by the polynomial $X^k h(X^{-1})$, where $h(X) = (X^n + 1)/g(X)$
  Example 5.3
- Consider the (7, 4) cyclic code given in Table 4.1 with generator polynomial $g(X) = 1 + X + X^3$
- The parity polynomial is $h(X) = (X^7 + 1)/g(X) = 1 + X + X^2 + X^4$
- The reciprocal of h(X) is
  $X^4 h(X^{-1}) = X^4(1 + X^{-1} + X^{-2} + X^{-4}) = 1 + X^2 + X^3 + X^4$
- The polynomial $X^4 h(X^{-1})$ divides $X^7 + 1$, we have
  $(X^7 + 1)/X^4 h(X - 1) = 1 + X^2 + X^3$

Generator matrix in systematic form:

- The generator matrix in systematic form is as follows.
- Dividing X^{n-k+i} by the generator polynomial g(X) for $i = 0, 1, \ldots, k-1$, we obtain

$$X^{n-k+i} = a_i(X)g(X) + b_i(X) \tag{5.15}$$

- where $b_i(X)$ is the remainder with the following form:

$$b_i(X) = b_{i0} + b_{i1}X + \ldots + b_{i,n-k+1}X^{n-k-1}$$

- Since $b_i(X) + X^{n-k+i}$ for $i = 0, 1, \ldots, k-1$ are multiples of g(X), they are code polynomials

Generator and Parity-Check Matrices of Cyclic Codes

(cont.) Arranging these k code polynomials as rows of a k x n matrix,

$$G = \begin{bmatrix} b_{00} & b_{01} & b_{02} & \cdots & b_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ b_{10} & b_{11} & b_{12} & \cdots & b_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ b_{20} & b_{21} & b_{22} & \cdots & b_{2,n-k-1} & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & & & & & & \\ b_{k-1,0} & b_{k-1,1} & b_{k-1,2} & \cdots & b_{k-1,,n-k-1} & 0 & 0 & 1 & \cdots & 1 \end{bmatrix} \tag{5.16}$$

which is the generator matrix of C in systematic form.
The corresponding parity-check matrix for C is

$$H = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & b_{00} & b_{01} & b_{02} & \cdots & b_{0,n-k-1} \\ 0 & 1 & 0 & \cdots & 0 & b_{10} & b_{11} & b_{12} & \cdots & b_{1,n-k-1} \\ 0 & 0 & 1 & \cdots & 0 & b_{20} & b_{21} & b_{22} & \cdots & b_{2,n-k-1} \\ \vdots & & & & & & & & & \\ 0 & 0 & 1 & \cdots & 1 & b_{k-1,0} & b_{k-1,1} & b_{k-1,2} & \cdots & b_{k-1,,n-k-1} \end{bmatrix} \tag{5.17}$$

Generator and Parity-Check Matrices of Cyclic Codes

Example 5.4

- Let $g(X) = 1 + X + X^3$, dividing X^3, X^4, X^5, and X^6 by g(X)
- We have

$$\begin{aligned} X^3 &= g(X) + (1 + X) \\ X^4 &= Xg(X) + (X + X^2) \\ X^5 &= (X^2 + 1)g(X) + (1 + X + X^2) \\ X^6 &= (X^3 + X + 1)g(X) + (1 + X^2) \end{aligned}$$

- Rearranging the equations above, the following are the four code polynomials:

$$\begin{aligned} v_0(X) &= 1 + X + X^3 \\ v_1(X) &= X + X^2 + X^4 \\ v_2(X) &= 1 + X + X^2 + X^5 \\ v_3(X) &= 1 + X^2 + X^6 \end{aligned}$$

Example 5.4 (cont.)

- Taking these four code polynomials as rows of a 4 x 7 matrix,
- The following is the generator matrix in systematic form for the (7, 4) cyclic code:

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

- which is identical to the matrix G.

Encoding of Cyclic Codes

Encoding operation using generator polynomial:

- Encoding of an (n, k) cyclic code in systematic form consists of three steps:
  1. Multiply the message polynomial u(X) by X^{n-k}
  2. Divide $X^{n-k}u(X)$ by g(X) to obtain the remainder b(X)
  3. Form the code word $b(X) + X^{n-k}u(X)$
- All these three steps can be accomplished with a division circuit which is a linear (n-k)-stage shift register with feedback connections based on the generator polynomial
- $g(X) = 1 + g_1X + g_2X^2 + \ldots + g_{n-k-1}X^{n-k-1} + X^{n-k}$

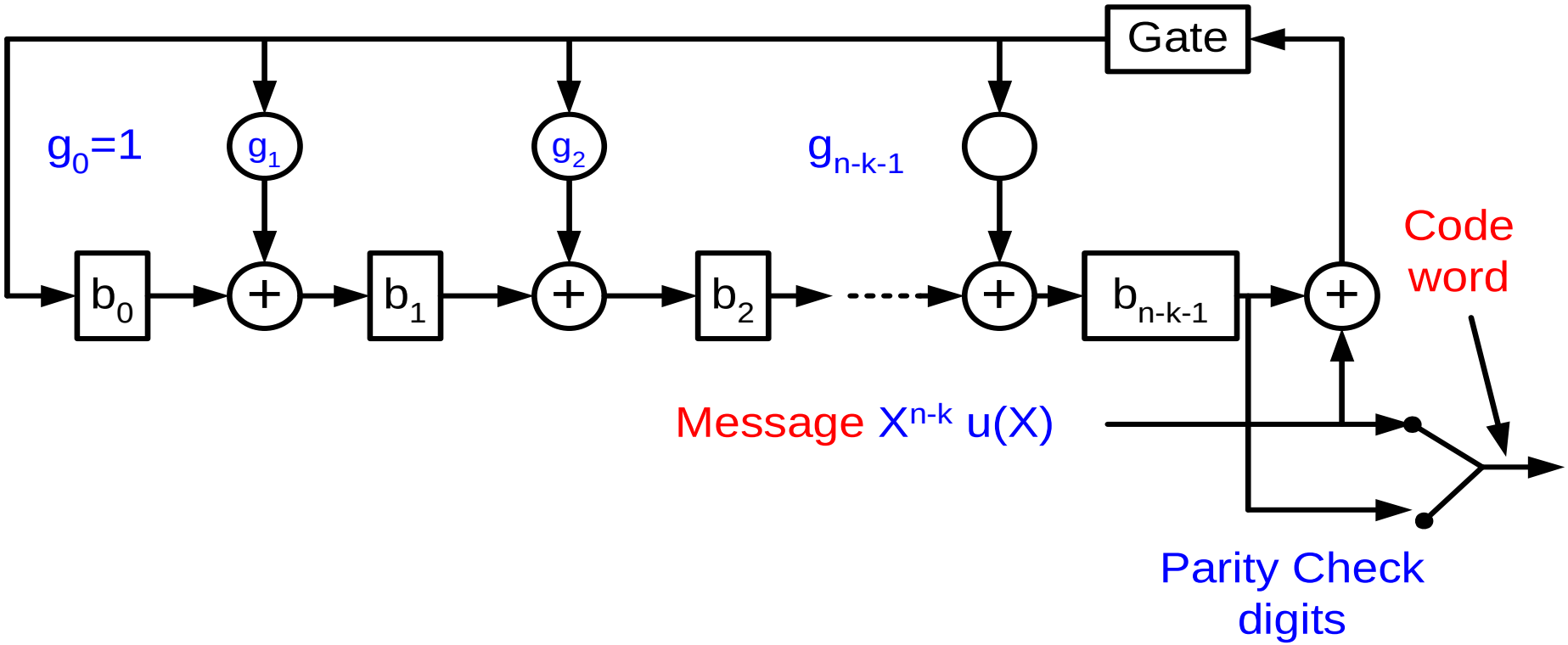

Figure: Encoding circuit for an (n,k) cyclic code

$$g(X) = 1 + g_1X + g_2X^2 + \ldots + g_{n-k-1}X^{n-k-1} + X^{n-k}$$

Encoding operation is as follows:

Step 1

- With the gate turned on, the k information digits $u_0, u_1, \ldots, u_{k-1}$ are shifted into the circuit and simultaneously into the communication channel
- Shifting the message u(X) into the circuit from the front end is equivalent to premultiplying u(X) by X^{n-k}
- As soon as the complete message has entered the circuit, the n - k digits in the register form the remainder and thus they are the parity-check digits.

Step 2

- Break the feedback connection by turning off the gate.

Step 3

- Shift the parity-check digits out and send them into the channel
- These n-k parity-check digits $b_0, b_1, \ldots, b_{n-k-1}$, together with the k information digits, form a complete code vector

Example 5.5

- Consider the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$
- The encoding circuit based on g(X) is shown in Fig. 5.2
- Suppose that the message u = (1 0 1 1) is to be encoded
- As the message digits are shifted into the register, the contents in the register are as follows:

Input	Register contents
	0 0 0 (initial state)
1	1 1 0 (first shift)
1	1 0 1 (second shift)
0	1 0 0 (third shift)
1	1 0 0 (fourth shift)

After four shift, the contents of the register are (1 0 0). The complete codeword is (1 0 0 1 0 1 1)

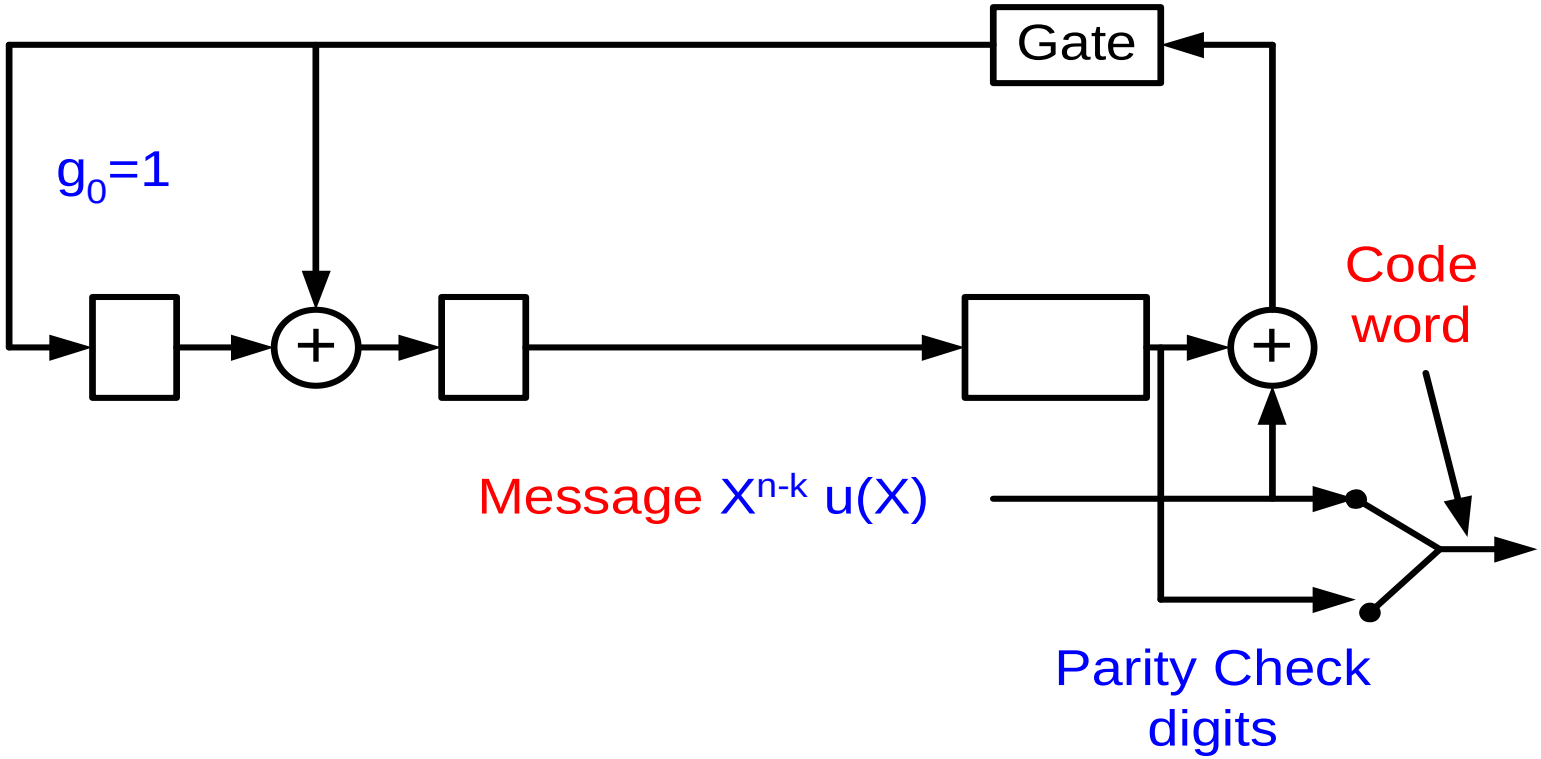

Figure: Encoding circuit for an (n,k) cyclic code

$$g(X) = 1 + g_1X + g_2X^2 + \ldots + g_{n-k-1}X^{n-k-1} + X^{n-k}$$

$$g(X) = 1 + X + X^3$$

Encoding operation using parity polynomial:

- Encoding of a cyclic code can also be accomplished by using its parity polynomial

$$h(X) = h_0 + h_1 X + \ldots + h_k X^k$$

- Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a code vector
- Since $h_k = 1$, the equalities of (5.12) can be put into the following form

$$v_{n-k-j} = \sum_{i=0}^{k-1} h_i v_{n-i-j} \quad \text{for} \quad 1 \leq j \leq n-k \qquad (5.18)$$

- which is known as a difference equation
- Given the k information digits, (5.18) is a rule to determine the n-k parity-check digits, $v_0, v_1, \ldots, v_{n-1}$
- An encoding circuit based on (5.18) is shown in Fig. 5.3

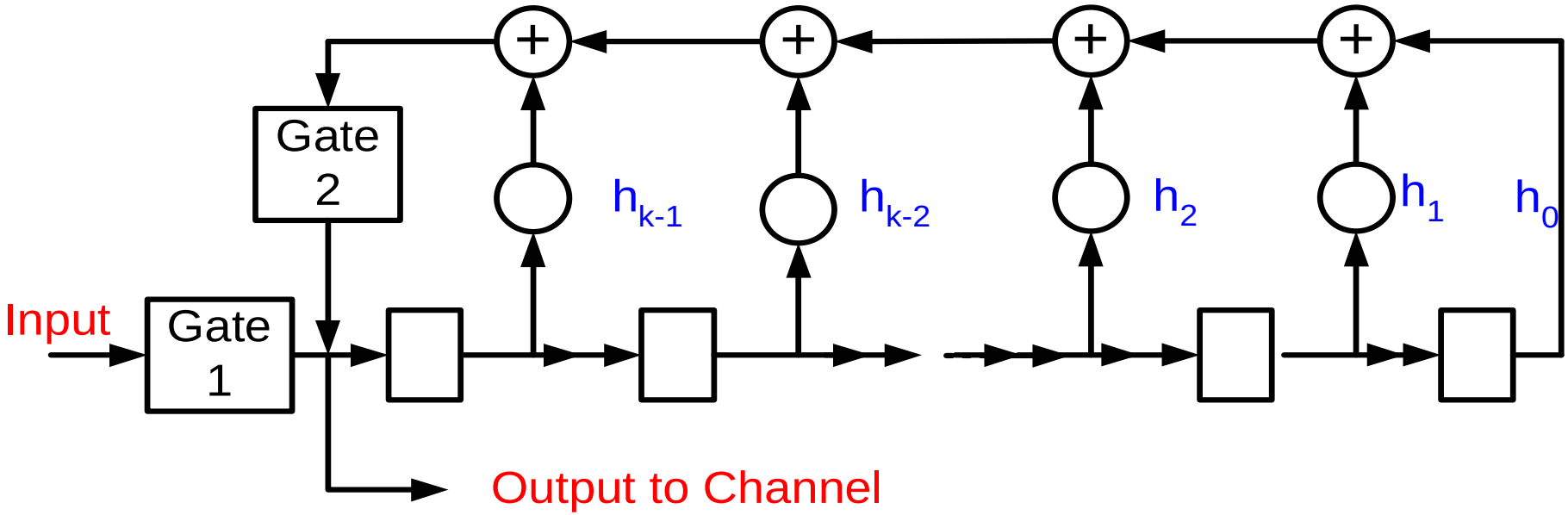

Figure: Encoding circuit for an (n,k) cyclic code

The feedback connections are based on the coefficients of the parity polynomial h(X)

$$h(X) = h_0 + h_1 X + \ldots + h_k X^k$$

The encoding operation can be described in the following steps:

Step 1

- Initially gate 1 is turned on and gate 2 is turned off
- The k information digits $u(X) = u_0 + u_1X + \ldots + u_{k-1}X^{k-1}$ are shifted into the register and the communication channel simultaneously.

  Step 2
- As soon as the k information digits have entered the shift register, gate 1 is turned off and gate 2 is turned on
- The first parity-check digit

$$\begin{aligned} v_{n-k-1} &= h_0 v_{n-1} + h_1 v_{n-2} + \ldots + h_{k-1} v_{n-k} \\ &= u_{k-1} + h_1 u_{k-2} + \ldots + h_{k-1} u_0 \end{aligned}$$

- is formed and appears at point P

Step 3

- The first parity-check digits is shifted
- The second parity-check digits

$$\begin{aligned} v_{n-k-2} &= h_0 v_{n-2} + h_1 v_{n-3} + \ldots + h_{k-1} v_{n-k-1} \\ &= u_{k-2} + h_1 u_{k-3} + \ldots + h_{k-2} u_0 + h_{k-1} v_{n-k-1} \end{aligned}$$

- is formed and appears at point P.

Step 4

- Step 3 is repeated until n-k parity-check digits have been formed and shifted into the channel
- Then gate 1 is turned on and gate 2 is turned off
- The next message is now ready to be shifted into the register

Example 5.6

- The parity polynomial of the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$ is

$$h(X) = (X^7 + 1)/(1 + X + X^3) = 1 + X + X^2 + X^4$$

- The encoding circuit based on h(X) is shown in Fig. 5.4
- The difference equation that determines the parity-check digits is

$$\begin{aligned} v_{3-j} &= 1v_{7-j} + 1v_{6-j} + 1v_{5-j} + 0v_{4-j} \\ &= v_{7-j} + v_{6-j} + v_{5-j} \quad \textit{for} \quad 1 \leq j \leq 3 \end{aligned}$$

- Suppose that the message to be encoded is (1 0 1 1), then $v_3 = 1, v_4 = 0, v_5 = 1, v_6 = 1$

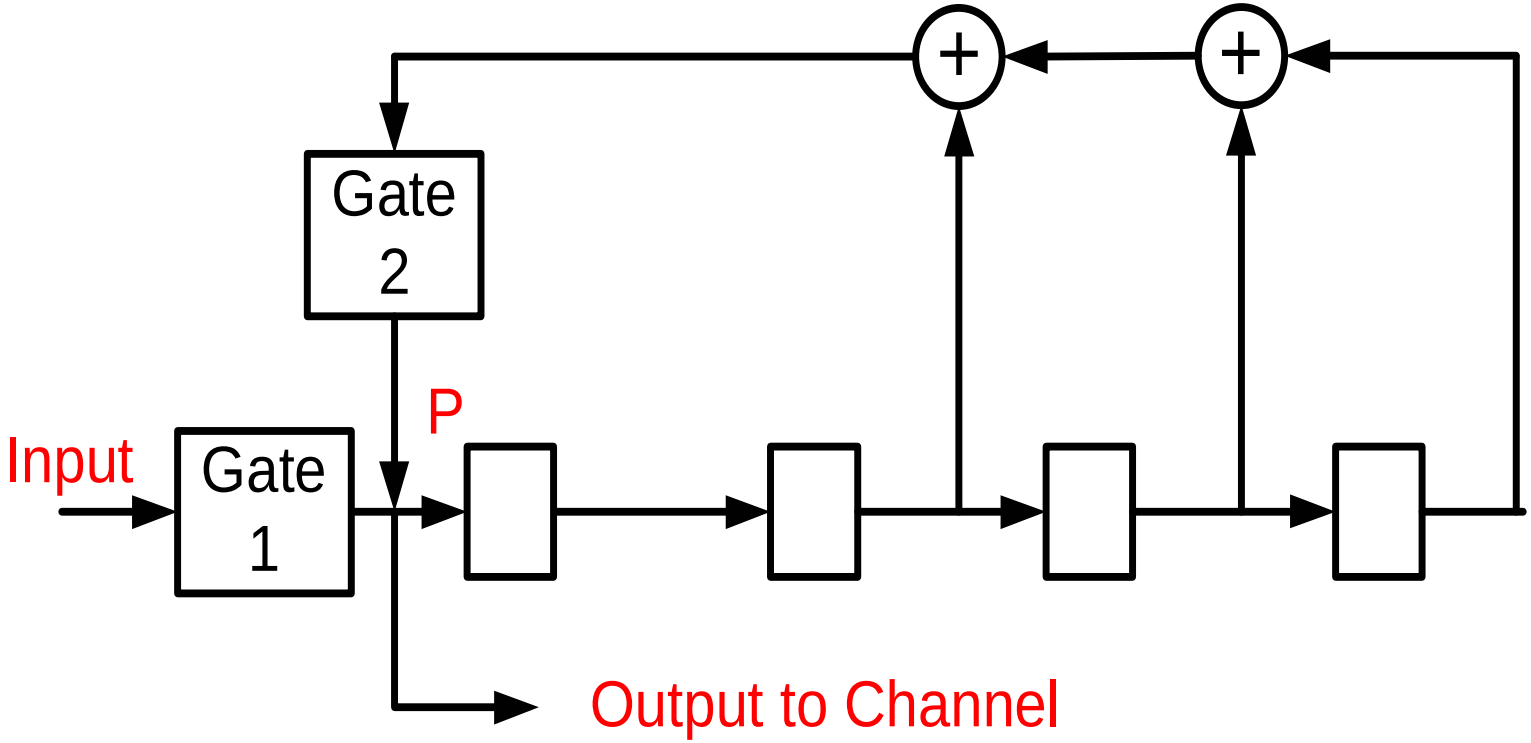

Figure: Encoding circuit for an (n,k) cyclic code

$$h(X) = (X^7 + 1)/(1 + X + X^3) = 1 + X + X^2 + X^4$$

Example 5.6 (cont.)

- The first parity-check digit is $v_2 = v_6 + v_5 + v_4 = 1 + 1 + 0 = 0$
- The second parity-check digit is $v_1 = v_5 + v_4 + v_3 = 1 + 0 + 1 = 0$
- The third parity-check digit is $v_0 = v_4 + v_3 + v_2 = 0 + 1 + 0 = 1$
- The code vector that corresponds to the message (1 0 1 1) is:
- (1 0 0 1 0 1 1)

Input	Register contents
	0 0 0 0 (initial state)
1	1 0 0 0 (first shift)
1	1 1 0 0 (second shift)
0	0 1 1 0 (third shift)
1	1 0 1 1 (fourth shift)
	0 1 0 1 (fifth shift)
	0 0 1 0 (sixth shift)
	1 0 0 0 (seventh shift)

Syndrome Computation and Error Detection

- Let $r = (r_0, r_1, \ldots, r_{n-1})$ be the received vector
- Since the channel noise, the received vector may not be the same as the transmitted code vector
- In the decoding of a linear code, the first step is to compute the syndrome $s = r.H^T$, where H is the parity check matrix
- If the syndrome is zero, r is a code vector and decoder accepts r as the transmitted code vector
- If the syndrome $\neq 0$ r is not a code vector and the presence of errors has been detected

- The received vector r is treated as a polynomial of degree n-1, or less,

$$r(X) = r_0 + r_1X + \ldots + r_{n-1}X^{n-1}$$

- Dividing r(X) by the generator polynomial g(X), we obtain

$$r(X) = a(X)g(X) + s(X) \tag{5.19}$$

- The remainder s(X) is a polynomial of degree n-k-1 or less
- The n - k coefficients of s(X) form the syndrome s
- s(X) is identical to zero if and only if the received polynomial r(X) is a code polynomial.
- The syndrome computation can be accomplished with a division circuit as shown in Fig. 5.5

- The received polynomial r(X) is shifted into the register with all stages initially set to zero.
- As soon as the entire r(X) has been shifted into the register, the contents in the register form the syndrome s(X)

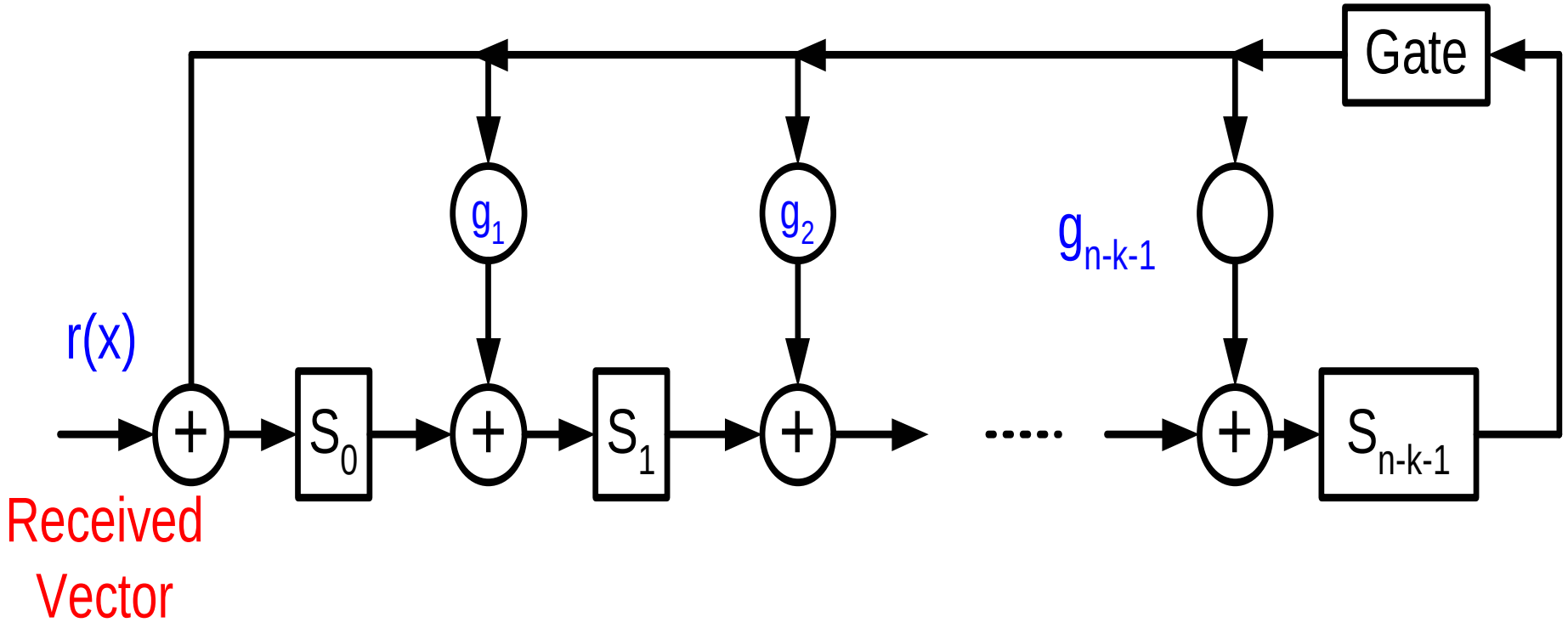

Figure: 5.5: An (n-k) stage syndrome circuit with input from the left end.

Example 5.7

- A syndrome circuit for the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$ is shown in Fig. 5.6
- Suppose that the received vector is r = (0 0 1 0 1 1 0)
- The syndrome of r is, s = (1 0 1)
- As the received vector is shifted into the circuit, the contents in the register are given in Table 5.3
- At the end of the seventh shift, the register contains the syndrome s = (1 0 1)
- If the register is shifted once more with the input gate disabled, the new contents will be $s^{(1)}(X) = (1\ 0\ 0)$, which is the syndrome of $r^{(1)}(X) = (0\ 0\ 0\ 1\ 0\ 1\ 1)$, a cyclic shift of r

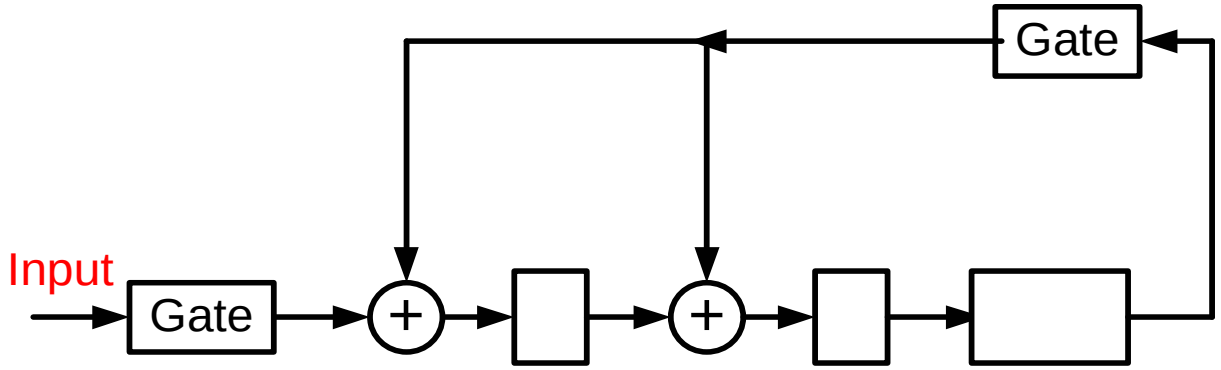

Figure: 5.6: Syndrome circuit for (7, 4) cyclic code generated by $g(x) = 1 + x + x^3$

Table: 5.3: Contents of the syndrome register with r=(0010110)and r=(1011011)

Shift	Input	Register contents		
		0 0 0 (initial state)		0 0 0
1	0	0 0 0 (first shift)	1	1 0 0
2	1	1 0 0 (second shift)	1	1 1 0
3	1	1 1 0 (third shift)	0	0 1 1
4	0	0 1 1 (fourth shift)	1	0 1 1
5	1	0 1 1 (fifth shift)	1	0 1 1
6	0	1 1 1 (sixth shift)	0	1 1 1
7	0	1 0 1 (syndrome s)	1	0 0 1
8	-	1 0 0 (syndrome $s^{(1)}$)		
9	-	0 1 0 (syndrome $s^{(2)}$)		

Syndrome computation by giving the input from the right end.

- Shift the received vector r(X) into the syndrome register from the right end, as shown in Fig. 5.7
- The content of the register do not form the syndrome of r(X) but it form the syndrome $s^{(n-k)}(X)$ of $r^{(n-k)}(X)$, which is the $(n-k)^{th}$ cyclic shift of r(X).
- Shifting r(X) from the right end is equivalent to pre-multiplying r(X) by X^{n-k}
- When the entire r(X) has entered the register, the register contains the remainder $\rho(X)$ resulting from dividing $X^{n-k}r(X)$ by the generator polynomial g(X)
- Thus, we have

$$X^{n-k}r(X) = a(X)g(X) + \rho(X) \qquad (5.24)$$

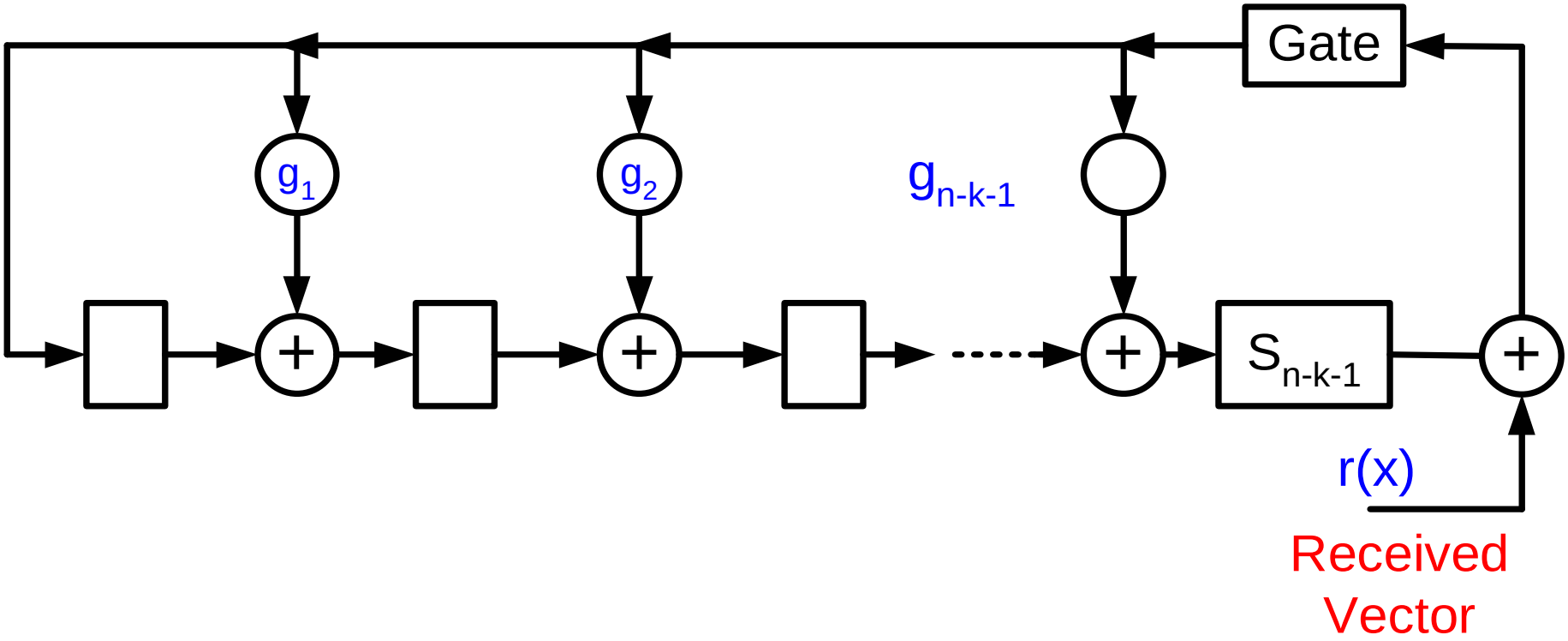

Figure: 5.7: An (n-k) syndrome circuit with input from the right end

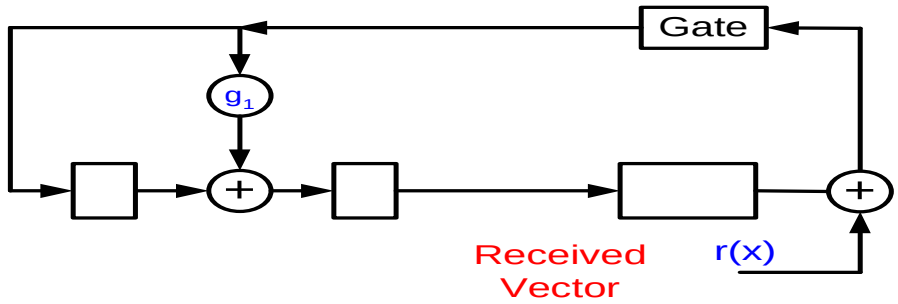

Figure: 5.71: An (n-k) syndrome circuit with input from the right end

Left shift r(x) by (n-k) times
0010110 => 0001011 => 1000101 => 1100010

Table: 5.3: Contents of the syndrome register with r=(1100010)

Shift	Input	Register contents
		0 0 0 (initial state)
1	0	0 0 0 (first shift)
2	1	1 1 0 (second shift)
3	0	0 1 1 (third shift)
4	0	1 1 1 (fourth shift)
5	0	1 0 1 (fifth shift)
6	1	0 1 0 (sixth shift)
7	1	1 0 1 (syndrome s)

- It follows from (5.1) that r(X) and $r^{(n-k)}(X)$ satisfy the following relation:

$$X^{n-k}r(X) = b(X)(X^n + 1) + r^{(n-k)}(X) \qquad (5.25)$$

- Combining (5.24) & (5.25) and using the fact that $X^n + 1 = g(X)h(X)$
- We have $r(n-k)(X) = [b(X)h(X) + a(X)]g(X) + \rho(X)$
- Therefore,$\rho(X)$ is indeed the syndrome of $r^{(n-k)}(X)$
- Let v(X) be the transmitted code word and let $e(X) = e_0 + e_1X \ldots + e_{n-1}X^{n-1}$ be the error pattern.

- The received polynomial is $r(X) = v(X) + e(X)$
- Since v(X) is a multiple of the generator polynomial g(X), we have: $e(X) = [a(X) + b(X)]g(X) + s(X)$
- where $b(X)g(X) = v(X)$.
- This shows that the syndrome is actually equal to the remainder resulting from dividing the error pattern by the generator polynomial.

Decoding of Cyclic Codes

General cyclic(Meggitt) decoder.

- Decoding of cyclic code consists of following three steps:
  1. Syndrome computation
  2. Association of the syndrome to an error pattern
  3. Error correction
- The syndrome computation for cyclic codes can be accomplished with a division circuit.
- A straightforward approach to the design of a decoding circuit is via a combinational logic circuit that implements the table-lookup procedure.
- The limit to this approach is that the complexity of the decoding circuit tends to grow exponentially with the code length and the number of errors that we intend to correct.
- The cyclic structure of a cyclic code allows us to decode a received vector $r(X) = r_0 + r_1X + r_2X^2 + \ldots + r_{n-1}X^{n-1}$ in a serial manner.
- The received digits are decoded one at a time and each digit is decoded with the same circuitry.

- As soon as the syndrome has been computed, the decoding circuit checks whether the syndrome s(X) corresponds to a correctable error pattern $e(X) = e_0 + e_1X + \ldots + e_{n-1}X^{n-1}$ with an error at the highest-order position $X^{n-1}(i.e., e_{n-1} = 1)$.
- If s(X) does not correspond to an error pattern with $e_{n-1} = 1$, the received polynomial and the syndrome register are cyclically shifted once simultaneously.
- By doing so, we obtain $r^{(1)}(X) = r_{n-1} + r_0X + r_1X^2 + \ldots + r_{n-2}X^{n-1}$ and the new contents in the syndrome register form the syndrome $s^{(1)}(X)$ of $r^{(1)}(X)$.
- The same decoding circuit will check whether $s^{(1)}(X)$ corresponds to an error pattern with an error at location X^{n-1}.
- If the syndrome s(X) does correspond to an error pattern with $e_{n-1} = 1$, the first received digit r_{n-1} is an erroneous digit and it must be corrected.
- This correction is carried out by $r_{n-1} \oplus e_{n-1}$.

- This correction results in a modified received polynomial $r_1(X) = r_0 + r_1X + r_2X^2 + \ldots + (_{rn-1} \oplus e_{n-1})X^{n-1}$.
- The effect of the error digit $e^n - 1$ on the syndrome is then removed from the syndrome s(X).
- This can be achieved by adding the syndrome of $e^{'}(X) = X^{n-1}$ to s(X).
- This sum is the syndrome of the modified received polynomial $r_1(X)$.
- Cyclically shift $r_1(X)$ and the syndrome register once simultaneously.
- This shift results in a received $r_1^{(1)}(X) = (r_{n-1} \oplus e_{n-1}) + r_0X + \ldots + r_{n-2}X^{n-1}$.

- The syndrome $s^1_{(1)}(X)$ of $r^1_{(1)}(X)$ is the remainder resulting from dividing $X[s(X) + X^{n-1}]$ by the generator polynomial g(X).
- Since the remainders resulting from dividing $X_s(X)$ and X^n by g(X) are $s^{(1)}(X)$ and 1, respectively, we have $s^{(1)}_1(X) = s^{(1)}(X) + 1$.
- Therefore, if 1 is added to the left end of the syndrome register while it is shifted, we obtain $s^1_{(1)}(X)$.
- A general decoder for an (n, k) cyclic code is shown in Fig. 5.8
- It consists of three major parts:
  1. A syndrome register
  2. An error-pattern detector
  3. A buffer register to hold the received vector
- To remove the effect of an error digit on the syndrome, we simply feed the error digit into the shift register from the left end through an EXCLUSIVE-OR gate

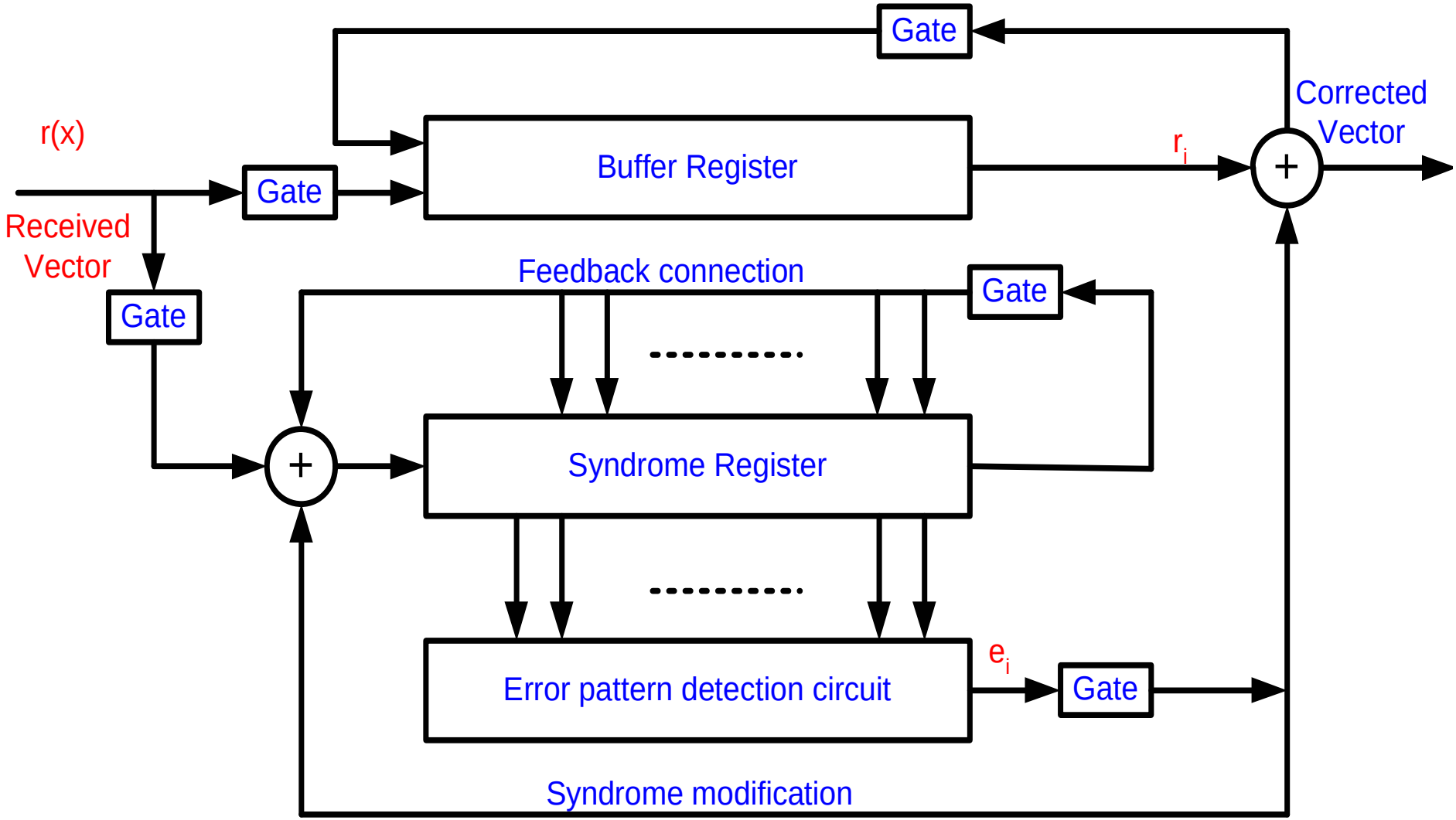

Figure: General cyclic(Meggitt) decoder with received polynomial r(X) is shifted into the syndrome register from the left end.

- The decoding operation is described as follows:

  Step 1
- The syndrome is formed by shifting the received vector into the syndrome register & also the received vector is stored into the buffer register.

  Step 2
- The syndrome is read into the detector and is tested for the corresponding error pattern.
- The detector is a combinational logic circuit which is designed in such a way that its output is 1 iff the syndrome in the syndrome register corresponds to a correctable error pattern with an error at the highest-order position X^{n-1}
- If a "1" appears at the output of the detector, the received symbol in the rightmost stage of the buffer register is assumed to be erroneous, if a "0" appears, the received symbol at the rightmost stage of the buffer register is assumed to be correct and no correction necessary.
- The output of the detector is the estimated error value for the symbol to come out of the buffer

Step 3

- The first received symbol is read out of the buffer
- If the first received symbol is detected to be an erroneous symbol, it is corrected by the output of the detector
- The output of the detector is fed back to the syndrome register to modify the syndrome
- This results in a new syndrome, which corresponds to the altered received vector shifted one place to the right.

Step 4

- The new syndrome formed in step 3 is used to detect whether or not the second received symbol is an erroneous symbol.
- The decoder repeats step 2 and 3

Step 5

- The decoder decodes the received vector symbol by symbol in the manner outlined above until the entire received vector is read out of the buffer register
- The decoder above is known as Meggitt decoder

Example 5.9

- Consider the decoding of the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$
- This code has minimum distance 3 and is capable of correcting any single error over a block of seven digits
- There are seven single-error patterns
- These seven error patterns and the all-zero vector form all the coset leader of the decoding table
- They form all the correctable error patterns
- Suppose that the received polynomial $r(X) = r_0 + r_1X + r_2X^2 + \ldots + r^6X^6$
- is shifted into the syndrome register from the left end
- The seven single-error patterns and their corresponding syndromes are listed in Table 5.4

Table: 5.4: Error patterns & their syndromes with received polynomial r(X) shifted into the circuit from the left end.

Error Pattern	Syndrome	Syndrome vector
$e_6(x) = X^6$	$s(x) = 1 + X^2$	(101)
$e_5(x) = X^5$	$s(x) = 1 + X + X^2$	(111)
$e_4(x) = X^4$	$s(x) = X + X^2$	(011)
$e_3(x) = X^3$	$s(x) = 1 + X$	(110)
$e_2(x) = X^2$	$s(x) = X^2$	(001)
$e_1(x) = X^1$	$s(x) = X$	(010)
$e_0(x) = X^0$	$s(x) = 1$	(100)

Table: 5.5: Error patterns & their syndromes with received polynomial r(X) shifted into the circuit from the right end.

Error Pattern	Syndrome	Syndrome vector
$e_6(x) = X^6$	$s^{(3)}(x) = X^2$	(001)
$e_5(x) = X^5$	$s^{(3)}(x) = X$	(010)
$e_4(x) = X^4$	$s^{(3)}(x) = 1$	(100)
$e_3(x) = X^3$	$s^{(3)}(x) = 1 + X^2$	(101)
$e_2(x) = X^2$	$s^{(3)}(x) = 1 + X + X^2$	(111)
$e_1(x) = X^1$	$s^{(3)}(x) = X + X^2$	(011)
$e_0(x) = X^0$	$s^{(3)}(x) = 1 + X$	(110)

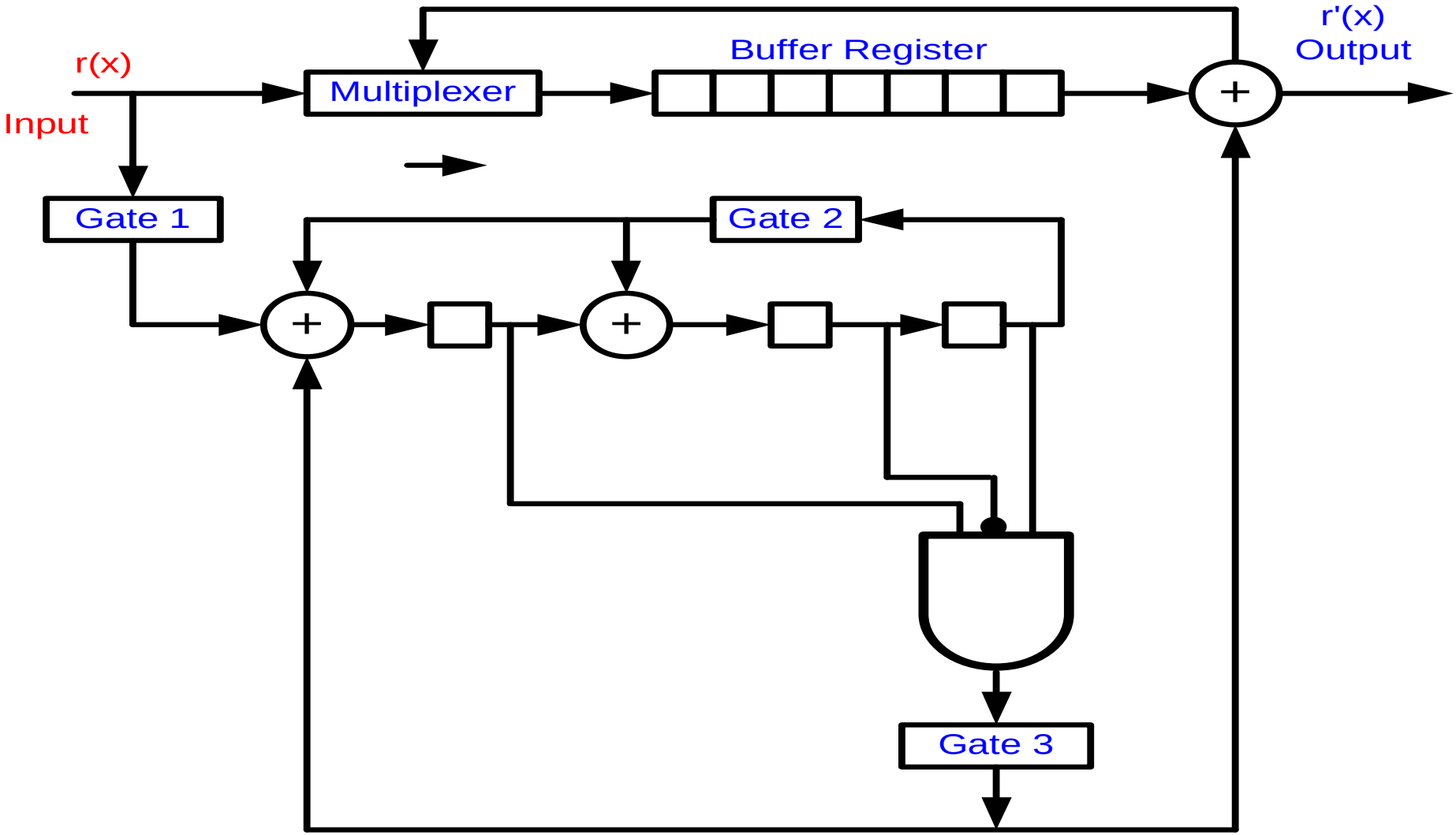

Figure: Decoding circuit for the (7,4) cyclic code generated by $g(x) = 1 + x + x^2$.

Table: 5.41: Contents of the syndrome register with r=(1001010) error at location X^6

Shift	Input	Register contents			Shift
	Input	$s_0 = i_0 \oplus s_2^{(-1)}$	$s_1 = s_0 \oplus s_2^{(-1)}$	$s_2 = s_1^{(-1)}$	
		0	0	0	(initial state)
1	0	0	0	0	(first shift)
2	1	1	0	0	(second shift)
3	0	0	1	0	(third shift)
4	1	1	0	1	(fourth shift)
5	0	1	0	0	(fifth shift)
6	0	0	1	0	(sixth shift)
7	1	1	0	1	(syndrome s)

Table: 5.42: Contents of the syndrome register with r=(1001001) error at location X^5

Shift	Input	Register contents			Shift
	Input	$s_0 = i_0 \oplus s_2^{(-1)}$	$s_1 = s_0 \oplus s_2^{(-1)}$	$s_2 = s_1^{(-1)}$	
		0	0	0	(initial state)
1	1	1	0	0	(first shift)
2	0	0	1	0	(second shift)
3	0	0	0	1	(third shift)
4	1	0	1	0	(fourth shift)
5	0	0	0	1	(fifth shift)
6	0	1	1	0	(sixth shift)
7	1	1	1	1	(syndrome s)

Table: 5.31: Contents of the syndrome register with r=(1011011)error at location X^2

Shift	Input	Register contents			Shift
	Input	$s_0 = i_0 \oplus s_2^{(-1)}$	$s_1 = s_0 \oplus s_2^{(-1)}$	$s_2 = s_1^{(-1)}$	
		0	0	0	(initial state)
1	1	1	0	0	(first shift)
2	1	1	1	0	(second shift)
3	0	0	1	1	(third shift)
4	1	0	1	1	(fourth shift)
5	1	0	1	1	(fifth shift)
6	0	1	1	1	(sixth shift)
7	1	0	0	1	(syndrome s)

Table: 5.32: Contents of the syndrome register with r=(1011011)error at location X^6 after 4 shifts

Shift	Register contents			Shift
	$s_0 = i_0 \oplus s_2^{(-1)}$	$s_1 = s_0 \oplus s_2^{(-1)}$	$s_2 = s_1^{(-1)}$	
	0	0	1	(initial state)
1	1	1	0	(first shift)
2	0	1	1	(second shift)
3	1	1	1	(third shift)
4	1	0	1	(fourth shift)
5	0	0	0	(fifth shift)
6	0	0	0	(sixth shift)
7	0	0	0	(syndrome s)

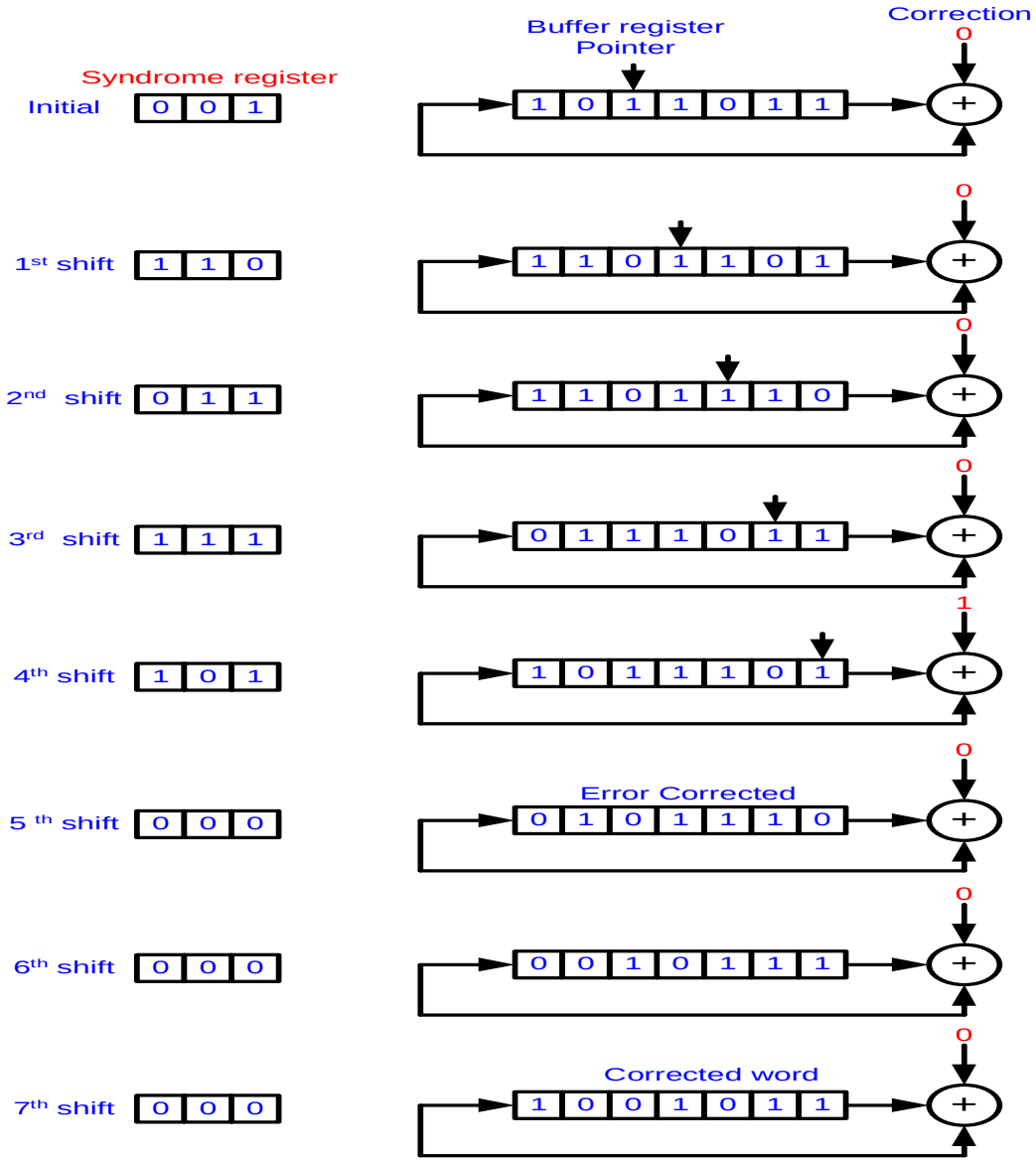

Figure: Error correction process of the circuit shown in Figure 5.9.

The decoding process of the decoder shown in Fig. 5.11 is identical to the decoding process of the decoder shown in Fig. 5.8

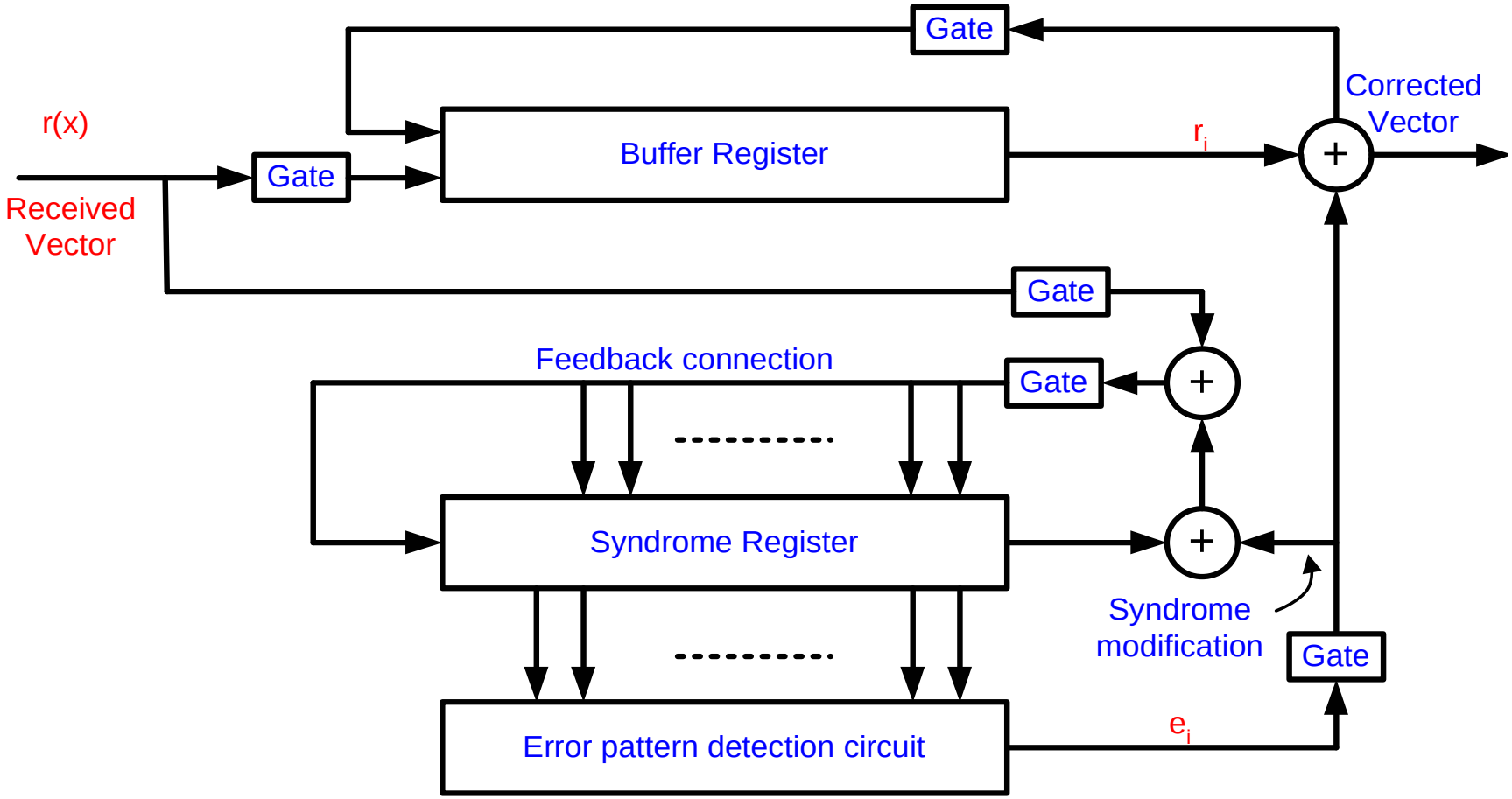

Figure: 5.11 General cyclic decoder with received polynomial r(X) is shifted into the syndrome register from the right end.

Example 5.10

- Suppose that the received polynomial r(X) is shifted into the syndrome register from the right end
- The seven single-error patterns and their corresponding syndromes are listed in Table 5.5
- We see that only when $e(X) = X^6$ occurs, the syndrome is (0 0 1) after the entire received polynomial r(X) has been shifted into the syndrome register
- If the single error occurs at the location Xi with $i \neq 6$, the syndrome in the register will not be (0 0 1) after the entire received polynomial r(X) has been shifted into the syndrome register
- After another 6-i shift, the syndrome register will contain (0 0 1), we obtain another decoding circuit for the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$, as shown in Fig. 5.12
- We see that the circuit shown in Fig. 5.9 and the circuit shown in Fig. 5.12 have the same complexity

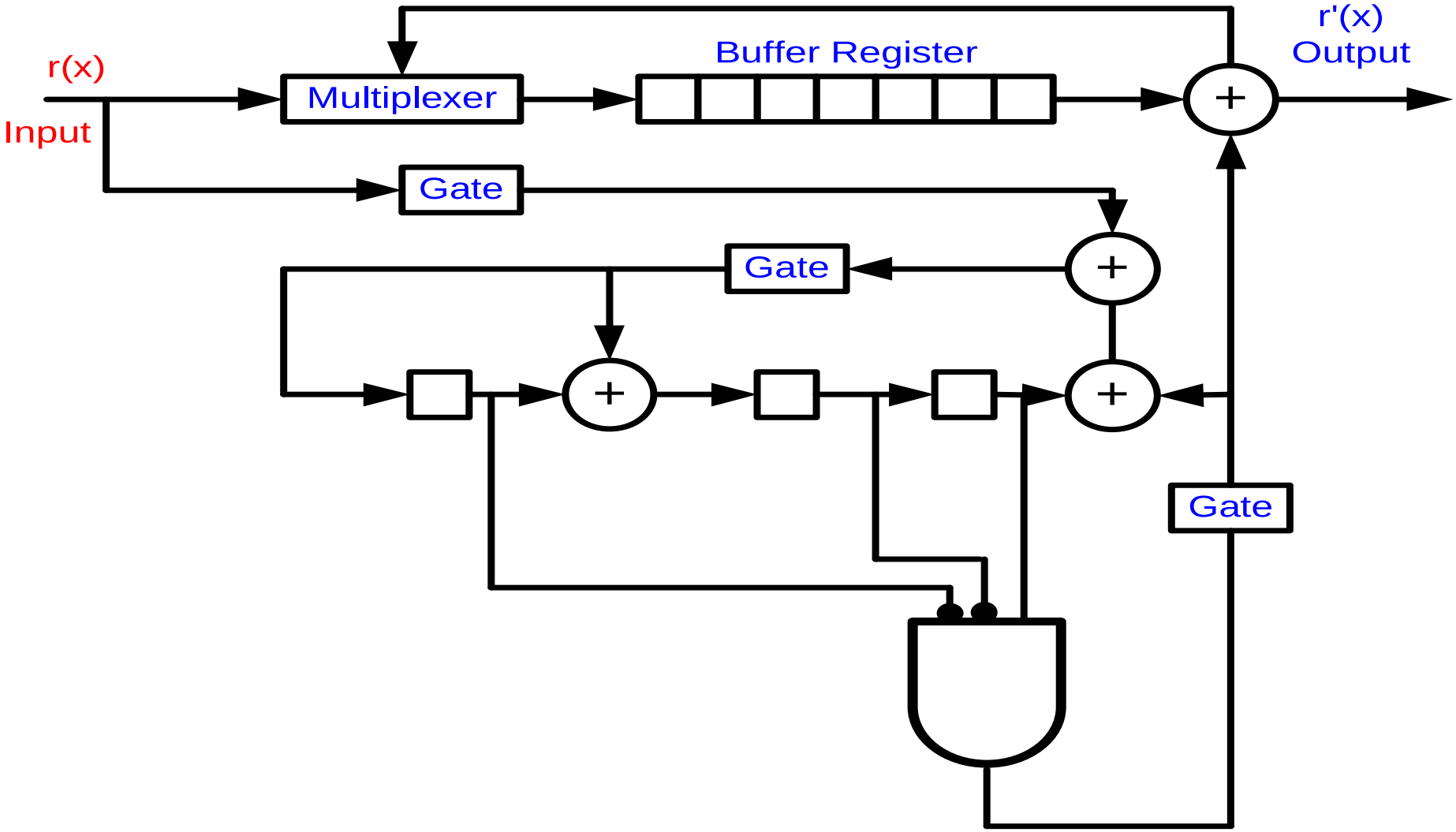

Figure: Decoding circuit for the (7,4) cyclic code generated by $g(x) = 1 + x + x^2$.

Cyclic Hamming Codes

- A Cyclic Hamming code of length $2^m - 1$ with $m \geq 3$ is generated by a primitive polynomial $p(X)$ of degree m.
- Dividing X^{m+i} by the generator polynomial $p(X)$ for $0 \leq i \leq 2^m - m - 1$

$$X^{m+i} = a_i(X)p(X) + b_i(X) \tag{1}$$

- where the remainder $b_i(X)$ is of the form

$$b_i(X) = b_{i0} + b_{i1}X + \ldots + b_{i,m-1}X^{m-1} \tag{2}$$

- Let $H = [I_m \ Q]$ is the parity matrix for the cyclic code generated by $p(X)$, where I_m is an identity matrix and Q is an $m \times (2^m - m - 1)$ matrix
- In the cyclic Hamming code, decoding is done digit by digit. The received digits will be decoded in the same manner with same circuitry.
- Once the received vector r(X) shifted into the syndrome register from the right end, the syndrome in the register is equal to the remainder resulting from dividing $X^m.X^{2^m-2}$ by generator polynomial $p(X)$. The syndrome is of the following form:

$$S(X) = X^{m-1} \tag{3}$$

- Suppose a single error occurs at the highest order position X^{2^m-2} of the received vector r(X) the resultant syndrome is $(0, 0, \ldots, 1)$.

- If a single error occurs at any other location of r(X) the resultant syndrome will be different from $(0, 0, \ldots, 1)$. Based on this result only a single m-input AND gate is needed to detect the syndrome pattern $(0, 0, \ldots, 1)$. The input to this AND gate are $s'_0, s'_1, \ldots s'_{m-2}, s'_{m-1}$ where s_i is a syndrome digit and s'_i denotes its complement.

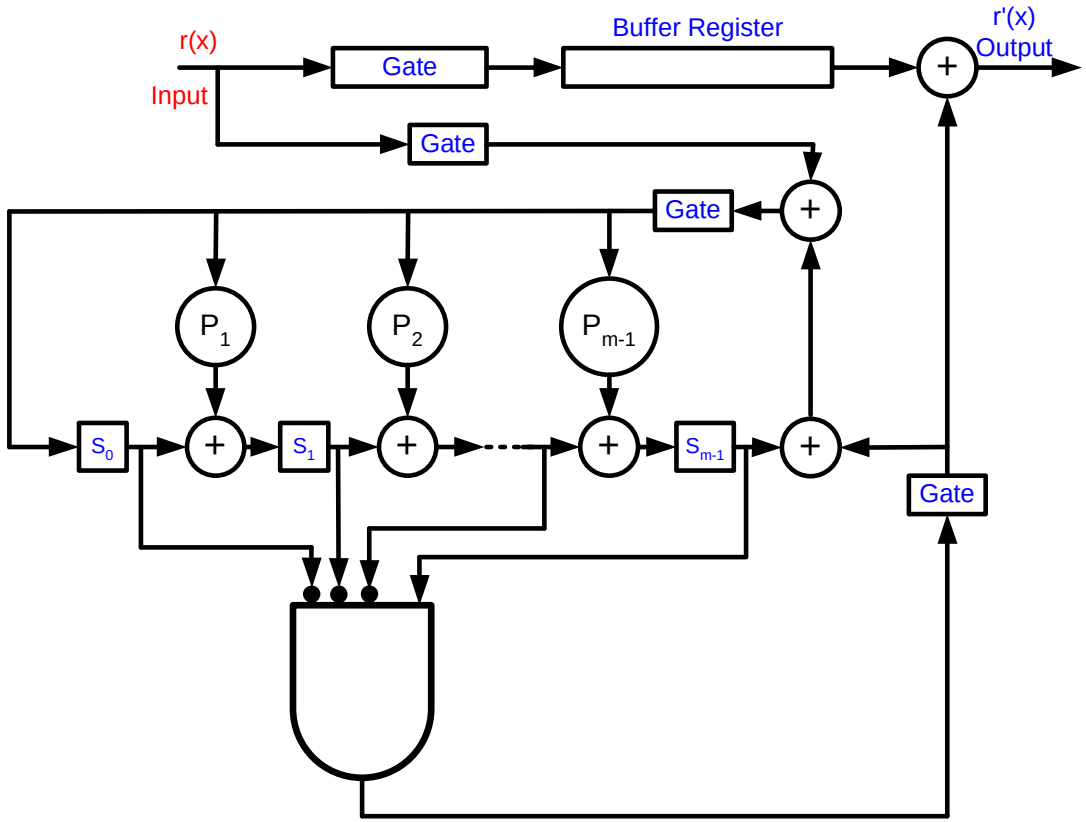

Figure: Decoding for a cyclic Hamming code.

- The decoding operation is described as follows:
  Step 1
- The syndrome is formed by shifting the received vector into the syndrome register & also the received vector is stored into the buffer register. If the syndrome is zero, the decoder assumes that no error has occurred, and no correction is necessary. If the syndrome is not zero, the decoder assumes that a single error has occurred.
  Step 2
- The received word is read out of the buffer register digit by digit. As each digit is read out of the buffer register the syndrome register is (0,0,0,. . ., 0,1) the next digit to come out of the buffer is the erroneous digit, and the output of the m-input AND gate is 1.
  Step 3
- The erroneous digit read out of the buffer register and is corrected by the output of the m-input AND gate. The correction is accomplished by an EXCLUSIVE-OR gate.
  Step 4
- The syndrome register is reset to zero after the entire received vector is read out of the buffer.

References


S. Lin and J. Daniel J. Costello, *Error Control Coding*, 2nd ed. Pearson/Prentice Hall, 2004.