Chapter 5

Cyclic Codes

Outlines

- Description of Cyclic Codes
- Generator and Parity-Check Matrices of Cyclic Codes
- Encoding of Cyclic Codes
- Syndrome Computation and Error Detection
- Decoding of Cyclic Codes
- Cyclic Hamming Codes
- Shortened Cyclic Codes

Introduction

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

Description of Cyclic Codes

Description of Cyclic Codes

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

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

  - which is called a cyclic shift of v

- If the $\mathbf{v}$ are cyclically shifted i places to the right, we have

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

- Cyclically shifting $\mathbf{v}$ i places to the right is equivalent to cyclically shifting $\mathbf{v}$ $(n - i)$ place to the left

Description of Cyclic Codes

- **Definition 4.1** An (n, k) linear code $\boldsymbol{C}$ is called a *cyclic code* if every cyclic shift of a code vector in $\boldsymbol{C}$ is also a code vector in $\boldsymbol{C}$
- The (7, 4) linear code given in Table 4.1 is a cyclic code
- To develop the algebraic properties of a cyclic code, we treat the components of a code vector $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ as the coefficients of a polynomial as follows:

$$\mathbf{v}(X) = v_0 + v_1 X + v_2 X^2 + \cdots + v_{n-1} X^{n-1}$$

  - If $v_{n-1} \neq 0$, the degree of $\mathbf{v}(X)$ is $n - 1$
  - If $v_{n-1} = 0$, the degree of $\mathbf{v}(X)$ is less than $n - 1$
  - The correspondence between the vector $\mathbf{v}$ and the polynomial $\mathbf{v}(X)$ is one-to-one

Description of Cyclic Codes

TABLE 5.1 A (7, 4) CYCLIC CODE GENERATED BY $g(X) = 1 + X + X^3$

Messages	Code Vectors	Code polynomials
(0 0 0 0)	0 0 0 0 0 0 0	$0 = 0 \cdot g(X)$
(1 0 0 0)	1 1 0 1 0 0 0	$1 + X + X^3 = 1 \cdot g(X)$
(0 1 0 0)	0 1 1 0 1 0 0	$X + X^2 + X^4 = X \cdot g(X)$
(1 1 0 0)	1 0 1 1 1 0 0	$1 + X^2 + X^3 + X^4 = (1 + X) \cdot g(X)$
(0 0 1 0)	0 0 1 1 0 1 0	$X^2 + X^3 + X^5 = X^2 \cdot g(X)$
(1 0 1 0)	1 1 1 0 0 1 0	$1 + X + X^2 + X^5 = (1 + X^2) \cdot g(X)$
(0 1 1 0)	0 1 0 1 1 1 0	$X + X^3 + X^4 + X^5 = (X + X^2) \cdot g(X)$
(1 1 1 0)	1 0 0 0 1 1 0	$1 + X^4 + X^5 = (1 + X + X^2) \cdot g(X)$
(0 0 0 1)	0 0 0 1 1 0 1	$X^3 + X^4 + X^6 = X^3 \cdot g(X)$
(1 0 0 1)	1 1 0 0 1 0 1	$1 + X + X^4 + X^6 = (1 + X^3) \cdot g(X)$
(0 1 0 1)	0 1 1 1 0 0 1	$X + X^2 + X^3 + X^6 = (X + X^3) \cdot g(X)$
(1 1 0 1)	1 0 1 0 0 0 1	$1 + X^2 + X^6 = (1 + X + X^3) \cdot g(X)$
(0 0 1 1)	0 0 1 0 1 1 1	$X^2 + X^4 + X^5 + X^6 = (X^2 + X^3) \cdot g(X)$
(1 0 1 1)	1 1 1 1 1 1 1	$1 + X + X^2 + X^3 + X^4 + X^5 + X^6 = (1 + X^2 + X^3) \cdot g(X)$
(0 1 1 1)	0 1 0 0 0 1 1	$X + X^5 + X^6 = (X + X^2 + X^3) \cdot g(X)$
(1 1 1 1)	1 0 0 1 0 1 1	$1 + X^3 + X^5 + X^6 = (1 + X + X^2 + X^3) \cdot g(X)$

Description of Cyclic Codes

- The code polynomial that corresponds to the code vector $\mathbf{v}^{(i)}$ is

$$\mathbf{v}^{(i)}(X) = v_{n-i} + v_{n-i+1}X + \cdots + v_{n-1}X^{i-1} + v_0X^i + v_1X^{i+1} + \cdots + v_{n-i-1}X^{n-1}$$

- Multiplying $\mathbf{v}(X)$ by X^i, we obtain

$$X^i\mathbf{v}(X) = v_0X^i + v_1X^{i+1} + \cdots + v_{n-i-1}X^{n-1} + \cdots + v_{n-1}X^{n+i-1}$$

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

$$\begin{aligned} X^i\mathbf{v}(X) &= v_{n-i} + v_{n-i+1}X + \cdots + v_{n-1}X^{i-1} + v_0X^i + \cdots + v_{n-i-1}X^{n-1} \\ &\quad + v_{n-i}(X^n+1) + v_{n-i+1}X(X^n+1) + \cdots + v_{n-1}X^{i-1}(X^n+1) \\ &= \mathbf{q}(X)\cdot(X^n + 1) + \mathbf{v}^{(i)}(X) \end{aligned} \qquad (5.1)$$

Description of Cyclic Codes

- **Theorem 5.1** The nonzero code polynomial of minimum degree in a cyclic code $\boldsymbol{C}$ is unique
- Proof
  - Let $\mathbf{g}(X) = g_0 + g_1X + \cdots + g_{r-1}X^{r-1} + X^r$ be a nonzero code polynomial of minimum degree in $\boldsymbol{C}$
  - Suppose that $\mathbf{g}(X)$ is not unique
  - There exists another code polynomial of degree r, say $\mathbf{g}'(X)$,
    $$\mathbf{g}'(X) = g'_0 + g'_1X + \cdots + g'_{r-1}X^{r-1} + X^r$$
  - Since C is linear,

    $\mathbf{g}(X) + \mathbf{g}'(X) = (g_0 + g'_0) + (g_1 + g'_1)X + \cdots + (g_{r-1} + g'_{r-1})X^{r-1}$ is also a code polynomial which has degree less than r
  - If $\mathbf{g}(X) + \mathbf{g}'(X) \neq 0$, $\mathbf{g}(X) + \mathbf{g}'(X)$ is a nonzero code polynomial with degree less than the minimum degree r

Description of Cyclic Codes

- **Theorem 5.2** Let $\mathbf{g}(X) = g_0 + g_1X + \cdots + g_{r-1}X^{r-1} + X^r$ be the 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

$$\begin{aligned}\mathbf{g}(X) &= g_1X + g_2X^2 + \cdots + g_{r-1}X^{r-1} + X^r \\ &= X \cdot (g_1 + g_2X + \cdots + g_{r-1}X^{r-2} + X^{r-1})\end{aligned}$$

  - If we shift $\mathbf{g}(X)$ cyclically n – 1 places to the right (or 1 place to the left), we obtain a nonzero code polynomial, $g_1 + g_2X + \cdots + g_{r-1}X^{r-2} + X^{r-1}$, which has a degree less than r
  - this is a contradiction to the assumption that $\mathbf{g}(X)$ is the nonzero code polynomial with minimum degree

Description of Cyclic Codes

- **Theorem 5.3** Let $\mathbf{g}(X) = g_0 + g_1X + \cdots + g_{r-1}X^{r-1} + X^r$ be the nonzero code polynomial of minimum degree in an (n, k) cyclic code $\boldsymbol{C}$. A binary polynomial of degree $n - 1$ or less is a code polynomial iff it is a multiple of $\mathbf{g}(X)$
- Proof
  - Let $\mathbf{v}(X)$ be a binary polynomial of degree $n - 1$ or less
  - Suppose that $\mathbf{v}(X)$ is a multiple of $\mathbf{g}(X)$

$$\mathbf{v}(X) = (a_0 + a_1X + \cdots + a_{n-r-1}X^{n-r-1})\mathbf{g}(X)$$
$$= a_0\mathbf{g}(X) + a_1\mathbf{g}(X) + \cdots + a_{n-r-1}X^{n-r-1}\mathbf{g}(X)$$

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

Description of Cyclic Codes

- Proof (cont.)
  - Now, let $\mathbf{v}(X)$ be a code polynomial in $\boldsymbol{C}$, dividing $\mathbf{v}(X)$ by $\mathbf{g}(X)$, we obtain

$$\mathbf{v}(X) = \mathbf{a}(X)\mathbf{g}(X) + \mathbf{b}(X)$$

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

$$\mathbf{b}(X) = \mathbf{a}(X)\mathbf{g}(X) + \mathbf{v}(X)$$

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

Contradiction!! => **b**(X)=0

Description of Cyclic Codes

- Recall that $\mathbf{g}(X) = g_0 + g_1X + \cdots + g_{r-1}X^{r-1} + X^r$.
- 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 $\underline{r=n-k}$.

Description of Cyclic Codes

- **Theorem 5.4** In an (n, k) cyclic code, there exists one and only one code polynomial of degree $n - k$,

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

  Every code polynomial is a multiple of $\boldsymbol{g}(X)$ and every binary polynomial of degree $n - 1$ or less that is a multiple of $\boldsymbol{g}(X)$ is a code polynomial

- An (n,k) cyclic code is completely specified by its nonzero code polynomial of minimum degree, $\boldsymbol{g}(X)$.
- The polynomial $\boldsymbol{g}(X)$ is called the *generator polynomial* of the code
- The degree of $\boldsymbol{g}(X)$ is equal to the number of parity-check digits of the code

Description of Cyclic Codes

- **Theorem 5.5** the generator polynomial $\mathbf{g}(X)$ of an (n, k) cyclic code is a factor of $X^n + 1$
- Proof
  - Multiplying $\boldsymbol{g}(X)$ by X^k results in a polynomial $X^k\boldsymbol{g}(X)$ of degree n
  - Dividing $X^k\boldsymbol{g}(X)$ by $X^n + 1$, we obtain

$$X^k\mathbf{g}(X) = (X^n + 1) + \mathbf{g}^{(k)}(X) \qquad (5.5)$$

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

Description of Cyclic Codes

- **Theorem 5.6** If $\mathbf{g}(X)$ is a polynomial of degree $n-k$ and is a factor of X^n+1, then $\boldsymbol{g}(X)$ generates an (n, k) cyclic code
- Proof
  - Consider the k polynomials $\mathbf{g}(X)$, $X\mathbf{g}(X)$, …, $X^{k-1}\mathbf{g}(X)$, which all have degree $n-1$ or less
  - A linear combination of these k polynomials,

$$\begin{aligned}\mathbf{v}(X) &= a_0\mathbf{g}(X) + a_1\mathbf{g}(X) + \cdots + a_{k-1}X^{k-1}\mathbf{g}(X) \\ &= (a_0 + a_1X + \cdots + a_{k-1}X^{k-1})\mathbf{g}(X)\end{aligned}$$

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

Description of Cyclic Codes

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

$$\begin{aligned} X\mathbf{v}(X) &= v_0X + v_1X^2 + \cdots + v_{n-1}X^n \\ &= v_{n-1}(X^n + 1) + (v_{n-1} + v_0X + \cdots + v_{n-2}X^{n-1}) \\ &= v_{n-1}(X^n + 1) + \mathbf{v}^{(1)}(X) \end{aligned}$$

    where $\mathbf{v}^{(1)}(X)$ is a cyclic shift of $\mathbf{v}(X)$
  - Since both $X\mathbf{v}(X)$ and $(X^n + 1)$ are divisible by $\mathbf{g}(X)$, $\mathbf{v}^{(1)}(X)$ must be divisible by $\mathbf{g}(X)$
  - Thus $\mathbf{v}^{(1)}(X)$ is a multiple of $\mathbf{g}(X)$ and is a linear combination of $\mathbf{g}(X)$, $X\mathbf{g}(X)$, …, $X^{k-1}\mathbf{g}(X)$
  - Hence, $\mathbf{v}^{(1)}(X)$ is also a code polynomial => Cyclic Code

Description of Cyclic Codes

- **Example 5.1**
  - The polynomial $X^7 + 1$ 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 $\mathbf{g}(X) = 1 + X + X^3$
  - 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 $\mathbf{g}(X) = 1 + X + X^3$

Description of Cyclic Codes

- **Example 5.1** (cont.)
  - Let **u** = (1 0 1 0) be the message to be encoded
  - The corresponding message polynomial is $\mathbf{u}(X) = 1 + X^2$
  - Multiplying u(X) by g(X) results in the following code polynomial :

$$\mathbf{v}(X) = (1 + X)(1 + X + X^3)$$
$$= 1 + X + X^2 + X^5$$

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

Description of Cyclic Codes

- Suppose that the message to be encoded is

$$\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$$

- The corresponding message polynomial is

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

- Multiplying $\mathbf{u}(X)$ by X^{n-k}, we obtain a polynomial of degree n – 1 or less,

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

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

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

  - where $\mathbf{a}(X)$ & $\mathbf{b}(X)$ are the quotient and the remainder, respectively.

Description of Cyclic Codes

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

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

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

$$\mathbf{b}(X) + X^{n-k}\mathbf{u}(X) = b_0 + b_1X + \cdots + b_{n-k-1}X^{n-k-1} + u_0X^{n-k} + u_1X^{n-k-1} + \cdots + u_{k-1}X^{n-1}$$

which corresponds to the code vector

$$(b_0, b_1, \cdots, b_{n-k-1}, u_0, u_1, \cdots, u_{k-1})$$

- The process above yields an (n, k) cyclic code in systematic form

Description of Cyclic Codes

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

Description of Cyclic Codes

- **Example 5.2**
  - Consider the (7, 4) cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$
  - Let $\mathbf{u}(X) = 1 + X^3$ be the message to be encoded
  - Dividing $X^3\mathbf{u}(X) = X^3 + X^6$ by $\mathbf{g}(X)$

$$
\begin{array}{r}
X^3 + X \quad \text{(quotient)} \\
X^3 + X + 1 \;\overline{)\; X^6 \qquad\qquad + X^3} \\
\underline{X^6 \qquad + X^4 + X^3} \\
X^4 \\
\underline{X^4 \qquad\quad + X^2 + X} \\
X^2 + X \quad \text{(remainder)},
\end{array}
$$

  - we obtain the remainder $\mathbf{b}(X) = X + X^2$
  - The code polynomial is $\mathbf{v}(X) = \mathbf{b}(X) + X^3\mathbf{u}(X) = X + X^2 + X^3 + X^6$
  - The corresponding code vector is $\mathbf{v}$ = (0 1 1 1 0 0 1)

Description of Cyclic Codes

TABLE 5.2 A (7, 4) CYCLIC CODE GENERATED BY $g(X) = 1 + X + X^3$

Message	Code word	
(0 0 0 0)	(0 0 0 0 0 0 0)	$0 = 0 \cdot g(X)$
(1 0 0 0)	(1 1 0 1 0 0 0)	$1 + X + X^3 = g(X)$
(0 1 0 0)	(0 1 1 0 1 0 0)	$X + X^2 + X^4 = Xg(X)$
(1 1 0 0)	(1 0 1 1 1 0 0)	$1 + X^2 + X^3 + X^4 = (1 + X)g(X)$
(0 0 1 0)	(1 1 1 0 0 1 0)	$1 + X + X^2 + X^5 = (1 + X^2)g(X)$
(1 0 1 0)	(0 0 1 1 0 1 0)	$X^2 + X^3 + X^5 = X^2g(X)$
(0 1 1 0)	(1 0 0 0 1 1 0)	$1 + X^4 + X^5 = (1 + X + X^2)g(X)$
(1 1 1 0)	(0 1 0 1 1 1 0)	$X + X^3 + X^4 + X^5 = (X + X^2)g(X)$
(0 0 0 1)	(1 0 1 0 0 0 1)	$1 + X^2 + X^6 = (1 + X + X^3)g(X)$
(1 0 0 1)	(0 1 1 1 0 0 1)	$X + X^2 + X^3 + X^6 = (X + X^3)g(X)$
(0 1 0 1)	(1 1 0 0 1 0 1)	$1 + X + X^4 + X^6 = (1 + X^3)g(X)$
(1 1 0 1)	(0 0 0 1 1 0 1)	$X^3 + X^4 + X^6 = X^3g(X)$
(0 0 1 1)	(0 1 0 0 0 1 1)	$X + X^5 + X^6 = (X + X^2 + X^3)g(X)$
(1 0 1 1)	(1 0 0 1 0 1 1)	$1 + X^3 + X^5 + X^6 = (1 + X + X^2 + X^3)g(X)$
(0 1 1 1)	(0 0 1 0 1 1 1)	$X^2 + X^4 + X^5 + X^6 = (X^2 + X^3)g(X)$
(1 1 1 1)	(1 1 1 1 1 1 1)	$1 + X + X^2 + X^3 + X^4 + X^5 + X^6 = (1 + X^2 + X^5)g(X)$

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 $\mathbf{g}(X) = g_0 + g_1X + \cdots + g_{n-k}X^{n-k}$
- We have shown that the k code polynomials $\mathbf{g}(X)$, $X\mathbf{g}(X)$, …, $X^{n-r-1}\mathbf{g}(X)$ span C (Proof of Theorem 5.6)
- If the k n-tuples corresponding to these k code polynomials are used as the rows of an $k \times n$ matrix, we obtain the following generator matrix for C :

$$\mathbf{G} = \begin{bmatrix} g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & 0 & \cdots & 0 \\ 0 & g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & 0 & \cdots & 0 \\ 0 & 0 & g_0 & g_1 & g_2 & \cdots & g_{n-k} & 0 & \cdots & 0 \\ \vdots & & & & & & & & & \vdots \\ 0 & 0 & \cdots & 0 & g_0 & g_1 & g_2 & \cdots & \cdots & g_{n-k} \end{bmatrix} \quad (5.9)$$

Generator and Parity-Check Matrices of Cyclic Codes

- In general, **G** is not in systematic form
- **G** can be put into systematic form with row operations
- For example, the (7, 4) cyclic code given in Table 4.1 with generator polynomial $\mathbf{g}(X) = 1 + X + X^3$ has the following matrix as a generator matrix :

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

  - **G** is not in systematic form

Generator and Parity-Check Matrices of Cyclic Codes

- If the first row is added to the third row and the sum of the first two rows is added to the fourth row, we obtain the following matrix :

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

  - which is in systematic form
  - This matrix generates the same code as **G**

Generator and Parity-Check Matrices of Cyclic Codes

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

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

- where the $\mathbf{h}(X)$ has the degree k and is of the following form :

$$\mathbf{h}(X) = h_0 + h_1 X + \cdots + h_k X^k$$

  - with $h_0 = h_k = 1$

- We want to show that a parity-check matrix of C may be obtained from $\mathbf{h}(X)$
  - Let $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ be a code vector in C
  - Then $\mathbf{v}(X) = \mathbf{a}(X)\mathbf{g}(X)$

Generator and Parity-Check Matrices of Cyclic Codes

- (cont.)
  - Multiplying $\mathbf{v}(X)$ by $\mathbf{h}(X)$, we obtain

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

$\mathbf{v}(X) = \mathbf{a}(X)\mathbf{g}(X)$
$\leq n-1 \qquad n-k$

  - Since the degree of $\mathbf{a}(X)$ is $k-1$ or less, the powers $X^k, X^{k+1},\ldots, X^{n-1}$ don't appear in $\mathbf{a}(X) + X^n\mathbf{a}(X)$
  - If we expand the product $\mathbf{v}(X)\mathbf{h}(X)$ 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 :

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

Generator and Parity-Check Matrices of Cyclic Codes

- We take the reciprocal of $\mathbf{h}(X)$, which is defined as follows:

$$X^k\mathbf{h}(X^{-1}) = h_k + h_{k-1}X + \cdots + h_0X^k \qquad (5.13)$$

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

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

Generator and Parity-Check Matrices of Cyclic Codes

- It follows from the $n - k$ equalities of (5.12) that any code vector $\mathbf{v}$ in C is orthogonal to every row of $\mathbf{H}$
- $\mathbf{H}$ is a parity-check matrix of the cyclic code C and the row space of $\mathbf{H}$ is the dual code of C
- Since the parity-check matrix $\mathbf{H}$ is obtained from the polynomial $\mathbf{h}(X)$, we call $\mathbf{h}(X)$ 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 $X^k\mathbf{h}(X^{-1})$, where $\mathbf{h}(X) = (X^n + 1)/\mathbf{g}(X)$
- Example 5.3
  - Consider the (7, 4) cyclic code given in Table 4.1 with generator polynomial $\mathbf{g}(X) = 1 + X + X^3$
  - The parity polynomial is
    $$\mathbf{h}(X) = X^7 + 1 / \mathbf{g}(X) = 1 + X + X^2 + X^4$$
  - The reciprocal of $\mathbf{h}(X)$ is
    $$X^4\mathbf{h}(X^{-1}) = X^4(1 + X^{-1} + X^{-2} + X^{-4}) = 1 + X^2 + X^3 + X^4$$

Generator and Parity-Check Matrices of Cyclic Codes

- **Example 5.3** (cont.)
  - The polynomial $X^4\mathbf{h}(X^{-1})$ divides X^7+1, we have
    $$(X^7+1) / X^4\mathbf{h}(X^{-1}) = 1 + X^2 + X^3$$
  - If we construct all the vectors of the (7, 3) code generated by $X^4\mathbf{h}(X^{-1}) = 1 + X^2 + X^3 + X^4$, we will find that it has minimum distance 4
  - It is capable of correcting any single error and simultaneously detecting any combination double errors

Generator and Parity-Check Matrices of Cyclic Codes

- The generator matrix in systematic form can also be formed easily
  - Dividing X^{n-k-i} by the generator polynomial g(X) for i = 0, 1, …, k – 1, we obtain

$$X^{n-k+i} = \mathbf{a}_i(X)\mathbf{g}(X) + \mathbf{b}_i(X) \qquad (5.15)$$

  - where $\mathbf{b}_i(X)$ is the remainder with the following form:

$$\mathbf{b}_i(X) = b_{i0} + b_{i1}X + \cdots + b_{i,n-k-1}X^{n-k-1}$$

  - Since $\mathbf{b}_i(X) + X^{n-k+i}$ for i = 0, 1, …, k – 1 are multiples of $\mathbf{g}(X)$, they are code polynomials

Generator and Parity-Check Matrices of Cyclic Codes

- (cont.)
  - Arranging these k code polynomials as rows of a $k \times n$ matrix, we obtain:

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

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

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

Generator and Parity-Check Matrices of Cyclic Codes

- **Example 5.4**
  - Again, let $\mathbf{g}(X) = 1 + X + X^3$, dividing X^3, X^4, X^5, and X^6 by $g(X)$
  - We have

$$X^3 = \mathbf{g}(X) + (1 + X),$$
$$X^4 = X\mathbf{g}(X) + (X + X^2),$$
$$X^5 = (X^2 + 1)\mathbf{g}(X) + (1 + X + X^2),$$
$$X^6 = (X^3 + X + 1)\mathbf{g}(X) + (1 + X^2)$$

  - Rearranging the equations above, we obtain the following four code polynomials:

$$\mathbf{v}_0(X) = 1 + X + X^3,$$
$$\mathbf{v}_1(X) = X + X^2 + X^4,$$
$$\mathbf{v}_2(X) = 1 + X + X^2 + X^5,$$
$$\mathbf{v}_3(X) = 1 + X^2 + X^6$$

Generator and Parity-Check Matrices of Cyclic Codes

- **Example 5.4** (cont.)
  - Taking these four code polynomials as rows of a 4×7 matrix, we obtain the following generator matrix in systematic form for the (7, 4) cyclic code:

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

  - which is identical to the matrix $\mathbf{G}'$ obtain earlier in this section

Encoding of Cyclic Codes

Encoding of Cyclic Codes

- Encoding of an (n, k) cyclic code in systematic form consists of three steps:
  - Multiply the message polynomial $\mathbf{u}(X)$ by X^{n-k}
  - Divide $X^{n-k}\mathbf{u}(X)$ by $\mathbf{g}(X)$ to obtain the remainder $\mathbf{b}(X)$
  - Form the code word $\mathbf{b}(X) + X^{n-k}\mathbf{u}(X)$
- 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

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

- Such a circuit is shown in Fig. 5.1

Encoding of Cyclic Codes

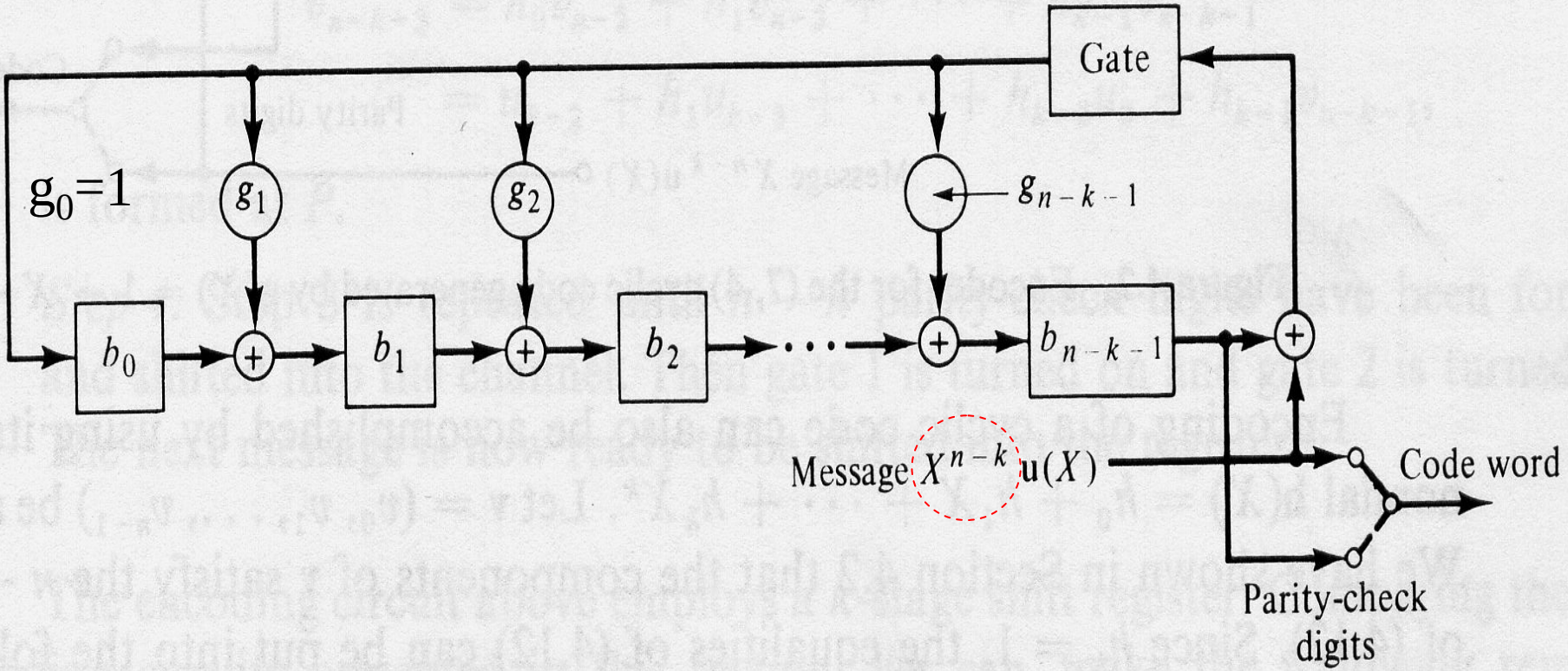

Figure 5.1 Encoding circuit for an (n, k) cyclic code with generator polynomial $\mathbf{g}(X) = 1 + g_1X + g_2X^2 + \cdots + g_{n-k-1}X^{n-k-1} + X^{n-k}$.

Encoding of Cyclic Codes

- 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 $\mathbf{u}(X)$ into the circuit from the front end is equivalent to premultiplying $\mathbf{u}(X)$ 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
  - Brake 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

Encoding of Cyclic Codes

- Example 5.5
  - Consider the (7, 4) cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$
  - The encoding circuit based on $\mathbf{g}(X)$ is shown in Fig. 5.2
  - Suppose that the message $\mathbf{u}$ = (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)

Encoding of Cyclic Codes

- The complete vector is (1 0 0 1 0 1 1) and code polynomial is $1 + X^3 + X^5 + X^6$

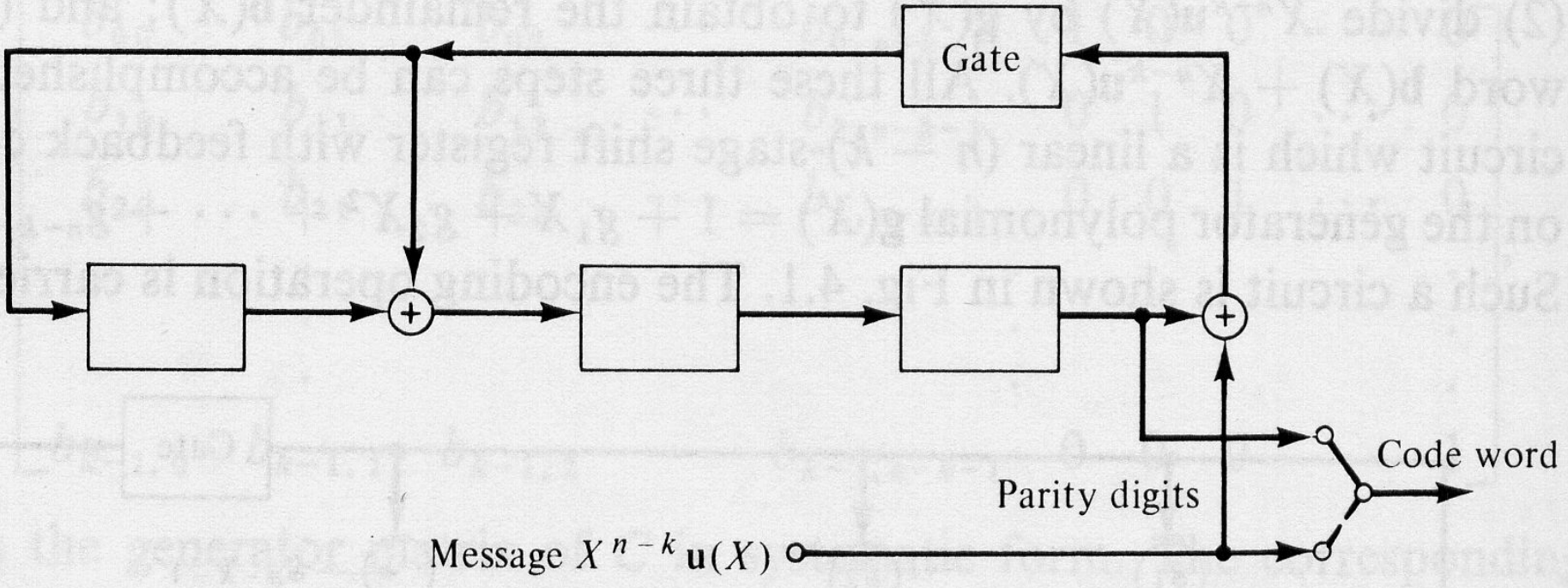

Figure 5.2 Encoder for the (7, 4) cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$.

Encoding of Cyclic Codes

- Encoding of a cyclic code can also be accomplished by using its parity polynomial $\mathbf{h}(X) = h_0 + h_1X + \cdots + h_kX^k$
- Let $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ be a code vector
- Since $h_k = 1$, the equalities of (5.12) can be put into the following form:

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

  - 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, $v_0, v_1, \ldots, v_{n-k-1}$
  - An encoding circuit based on (5.18) is shown in Fig. 5.3

Encoding of Cyclic Codes

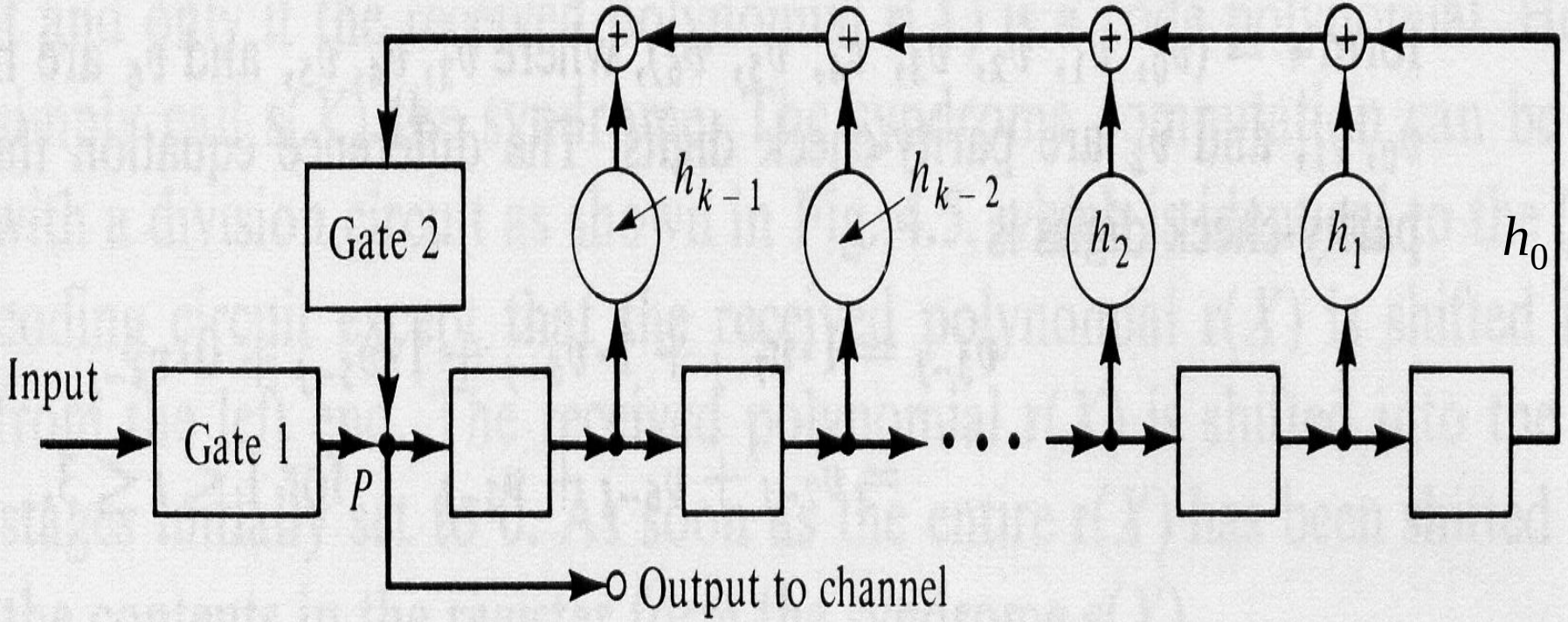

Figure 5.3 Encoding circuit for an (n, k) cyclic code based on the parity polynomial $\mathbf{h}(X) = 1 + h_1 X + \cdots + X^k$.

Encoding of Cyclic Codes

- The feedback connections are based on the coefficients of the parity polynomial $\mathbf{h}(X)$
- The encoding operation can be described in the following steps :
- Step 1
  - initially gate 1 is turned in and gate 2 is turned off
  - The k information digits $\mathbf{u}(X) = u_0 + u_1X + \cdots + u_{k-1}X^{k-1}$ are shifted into the register and the communication channel simultaneously

Encoding of Cyclic Codes

- 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

$$v_{n-k-1} = h_0 v_{n-1} + h_1 v_{n-2} + \cdots + h_{k-1} v_{n-k}$$
$$= u_{k-1} + h_1 u_{k-2} + \cdots + h_{k-1} u_0$$

    is formed and appears at point P
- Step 3
  - The first parity-check digits is shifted into the channel and is also shifted into the register

Encoding of Cyclic Codes

- Step 3 (cont.)
  - The second parity-check digits

$$v_{n-k-2} = h_0 v_{n-2} + h_1 v_{n-3} + \cdots + h_{k-1} v_{n-k-1}$$
$$= u_{k-2} + h_1 u_{k-3} + \cdots + h_{k-2} u_0 + h_{k-1} v_{n-k-1}$$

    is formed at 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

Encoding of Cyclic Codes

- **Example 5.6**
  - The parity polynomial of the (7, 4) cyclic code generated by

$$\mathbf{g}(X) = 1 + X + X^3$$

    is

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

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

$$\begin{aligned} v_{3-j} &= 1 \cdot v_{7-j} + 1 \cdot v_{6-j} + 1 \cdot v_{5-j} + 0 \cdot v_{4-j} \\ &= v_{7-j} + v_{6-j} + v_{5-j} \end{aligned} \qquad \text{for } 1 \le j \le 3$$

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

Encoding of Cyclic Codes

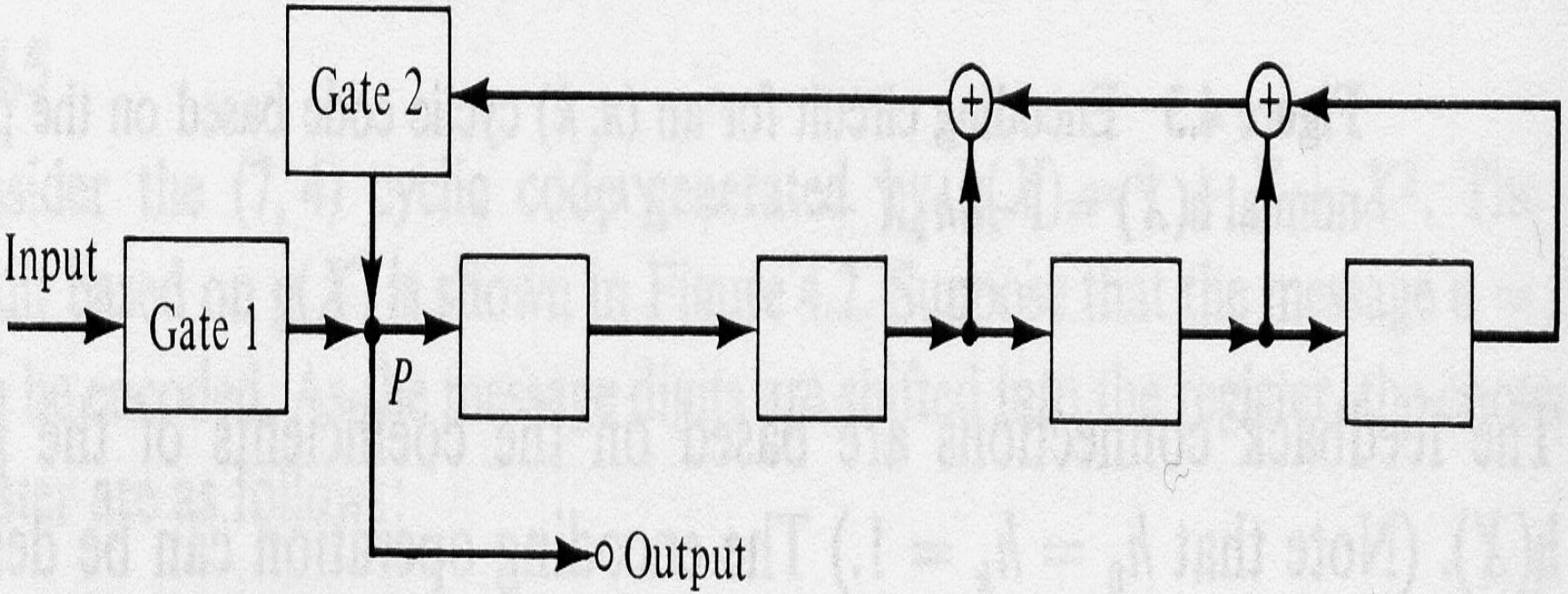

Figure 5.4 Encoding circuit for the (7, 4) cyclic code based on its parity polynomial $\mathbf{h}(X) = 1 + X + X^2 + X^4$.

Encoding of Cyclic Codes

- **Example 5.6** (cont.)
  - The first parity-check digits is
$$v_2 = v_6 + v_5 + v_4 = 1 + 1 + 0 = 0$$
  - The second parity-check digits is
$$v_1 = v_5 + v_4 + v_3 = 1 + 0 + 1 = 0$$
  - The third parity-check digits is
$$v_0 = v_4 + v_3 + v_2 = 0 + 1 + 0 = 1$$
  - Thus, the code vector that corresponds to the message (1 0 1 1) is (1 0 0 1 0 1 1)

Syndrome Computation and Error Detection

Syndrome Computation and Error Detection

- Let $\mathbf{r} = (r_0, r_1, \ldots, r_{n-1})$ 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 $\mathbf{s} = \mathbf{r} \cdot \mathbf{H}^T$, where $\mathbf{H}$ is the parity-check matrix
  - If the syndrome = $\mathbf{0}$, $\mathbf{r}$ is a code vector and decoder accepts $\mathbf{r}$ as the transmitted code vector
  - If the syndrome $\neq \mathbf{0}$, $\mathbf{r}$ is not a code vector and the presence of errors has been detected

Syndrome Computation and Error Detection

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

$$\mathbf{r}(X) = r_0 + r_1 X + \cdots + r_{n-1} X^{n-1}$$

- Dividing $\mathbf{r}(X)$ by the generator polynomial $\mathbf{g}(X)$, we obtain

Degree:$n-k$

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

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

Syndrome Computation and Error Detection

- The received polynomial $\mathbf{r}(X)$ is shifted into the register with all stages initially set to 0
- As soon as the entire $\mathbf{r}(X)$ has been shifted into the register, the contents in the register form the syndrome $\mathbf{s}(X)$

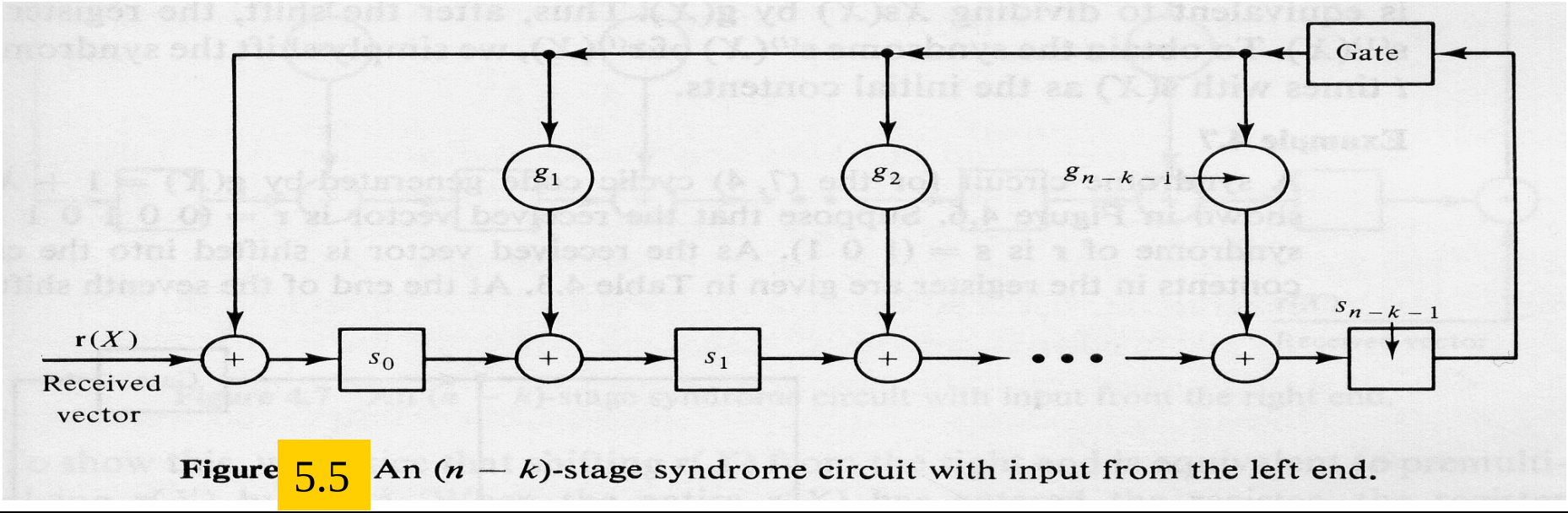

Figure 5.5 An $(n - k)$-stage syndrome circuit with input from the left end.

Syndrome Computation and Error Detection

- **Theorem 5.8** Let $\mathbf{s}(X)$ be the syndrome of a received polynomial $\mathbf{r}(X) = r_0 + r_1X + \cdots + r_{n-1}X^{n-1}$. Then the remainder $\mathbf{s}^{(1)}(X)$ resulting from dividing $X\mathbf{s}(X)$ by the generator polynomial $\mathbf{g}(X)$ is the syndrome of $\mathbf{r}^{(1)}(X)$, which is a cyclic shift of $\mathbf{r}(X)$
- Proof
  - It follows from (5.1) that $\mathbf{r}(X)$ and $\mathbf{r}^{(1)}(X)$ satisfy the following relationship:

$$X\mathbf{r}(X) = r_{n-1}(X^n + 1) + \mathbf{r}^{(1)}(X) \qquad (5.20)$$

  - Rearranging (5.20), we have

$$\mathbf{r}^{(1)}(X) = r_{n-1}(X^n + 1) + X\mathbf{r}(X) \qquad (5.21)$$

Syndrome Computation and Error Detection

- Proof
  - Dividing both sides of (5.21) by g(X) and using the fact that $X^n + 1 = \mathbf{g}(X)\mathbf{h}(X)$, we obtain:

    $$\mathbf{c}(X)\mathbf{g}(X) + \rho(X) = r_{n-1}\mathbf{g}(X)\mathbf{h}(X) + X[\mathbf{a}(X)\mathbf{g}(X) + \mathbf{s}(X)] \quad (5.22)$$

  - where $\rho(X)$ is the remainder resulting from dividing $\mathbf{r}^{(1)}(X)$ by $\mathbf{g}(X)$, $\rho(X)$ is the syndrome of $\mathbf{r}^{(1)}(X)$
  - Rearranging (5.22), we obtain the following relationship between $\rho(X)$ and $X\mathbf{s}(X)$:

    $$X\mathbf{s}(X) = [\mathbf{c}(X) + r_{n-1}\mathbf{h}(X) + X\mathbf{a}(X)]\mathbf{g}(X) + \rho(X) \quad (5.23)$$

  - From (5.23) we see that $\rho(X)$ is also the remainder resulting from dividing $X\mathbf{s}(X)$ by $\mathbf{g}(X)$, therefore, $\rho(X) = \mathbf{s}^{(1)}(X)$

Syndrome Computation and Error Detection

- It follows from **Theorem 5.8** that the remainder $\mathbf{s}^{(i)}(X)$ resulting from dividing $X^i\mathbf{s}(X)$ by the generator polynomial $\mathbf{g}(X)$ is the syndrome of $\mathbf{r}^{(i)}(X)$, which is the ith cyclic shift of $\mathbf{r}(X)$
- The syndrome $\mathbf{s}^{(1)}(X)$ of $\mathbf{r}^{(1)}(X)$ can be obtained by shifting the syndrome register once with $\mathbf{s}(X)$ as the initial contents and with the input gate disabled
- This is due to the fact that shifting the syndrome register once with $\mathbf{s}(X)$ as the initial contents is equivalent to dividing $X\mathbf{s}(X)$ by $\mathbf{g}(X)$
- After the shift, the register contains $\mathbf{s}^{(1)}(X)$

Syndrome Computation and Error Detection

- **Example 5.7**
  - A syndrome circuit for the (7, 4) cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$ is shown in Fig. 5.6
  - Suppose that the received vector is $\mathbf{r}$ = (0 0 1 0 1 1 0)
  - The syndrome of $\mathbf{r}$ is $\mathbf{s}$ = (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 $\mathbf{s}$ = (1 0 1)
  - If the register is shifted once more with the input gate disabled, the new contents will be $\mathbf{s}^{(1)}(X)$ = (1 0 0), which is the syndrome of $\mathbf{r}^{(1)}(X)$ = (0 0 0 1 0 1 1), a cyclic shift of $\mathbf{r}$

Syndrome Computation and Error Detection

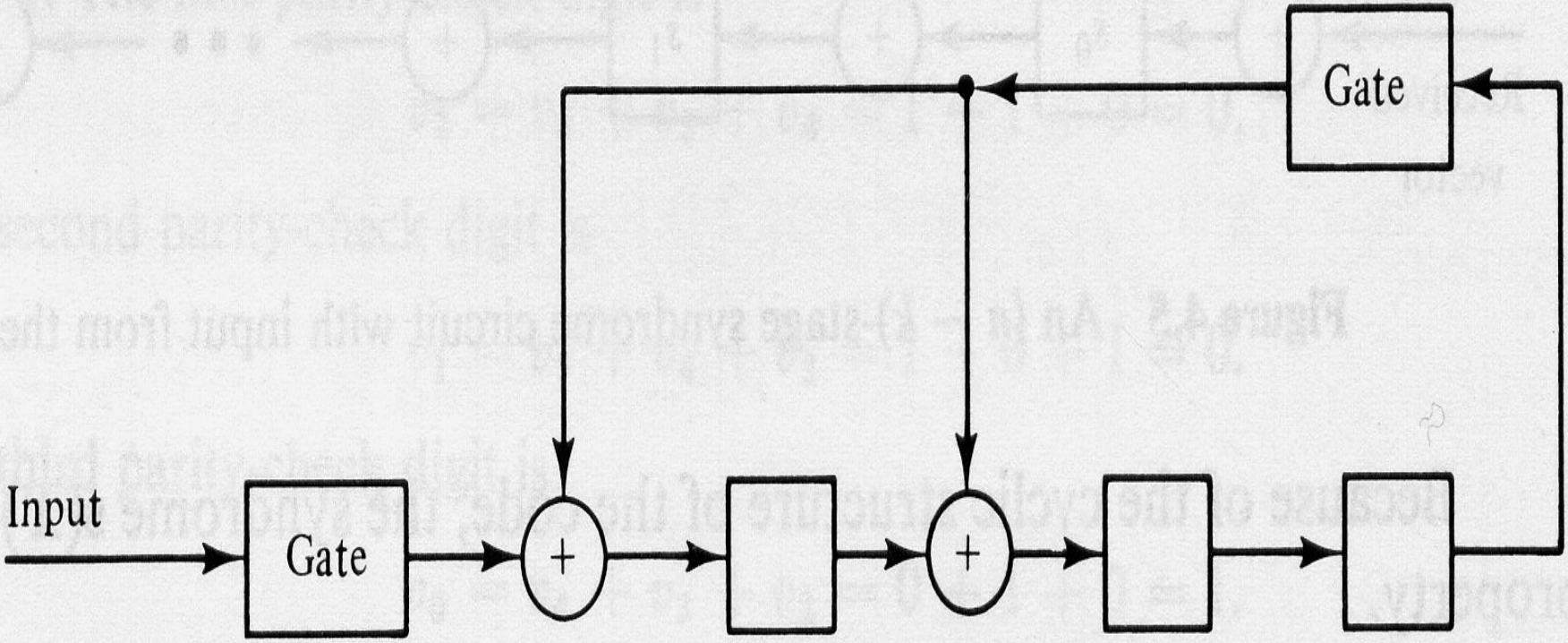

Figure 5.6 Syndrome circuit for the (7, 4) cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$.

Syndrome Computation and Error Detection

TABLE 5.3 CONTENTS OF THE SYNDROME REGISTER SHOWN IN FIGURE 4.6 WITH $\mathbf{r} = (0\ 0\ 1\ 0\ 1\ 1\ 0)$ AS INPUT

Shift	Input	Register contents	
		0 0 0	(initial state)
1	0	0 0 0	
2	1	1 0 0	
3	1	1 1 0	
4	0	0 1 1	
5	1	0 1 1	
6	0	1 1 1	
7	0	1 0 1	(syndrome **s**)
8	—	1 0 0	(syndrome $\mathbf{s}^{(1)}$)
9	—	0 1 0	(syndrome $\mathbf{s}^{(2)}$)

Syndrome Computation and Error Detection

- We may shift the received vector $\mathbf{r}(X)$ into the syndrome register from the right end, as shown in Fig. 5.7
- They form the syndrome $\mathbf{s}^{(n-k)}(X)$ of $\mathbf{r}^{(n-k)}(X)$, which is the $(n-k)$th cyclic shift of $\mathbf{r}(X)$
- Shifting $\mathbf{r}(X)$ from the right end is equivalent to premultiplying $\mathbf{r}(X)$ by X^{n-k}
- When the entire $\mathbf{r}(X)$ has entered the register, the register contains the remainder $\rho(X)$ resulting from dividing $X^{n-k}\mathbf{r}(X)$ by the generator polynomial $\mathbf{g}(X)$
- Thus, we have

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

Syndrome Computation and Error Detection

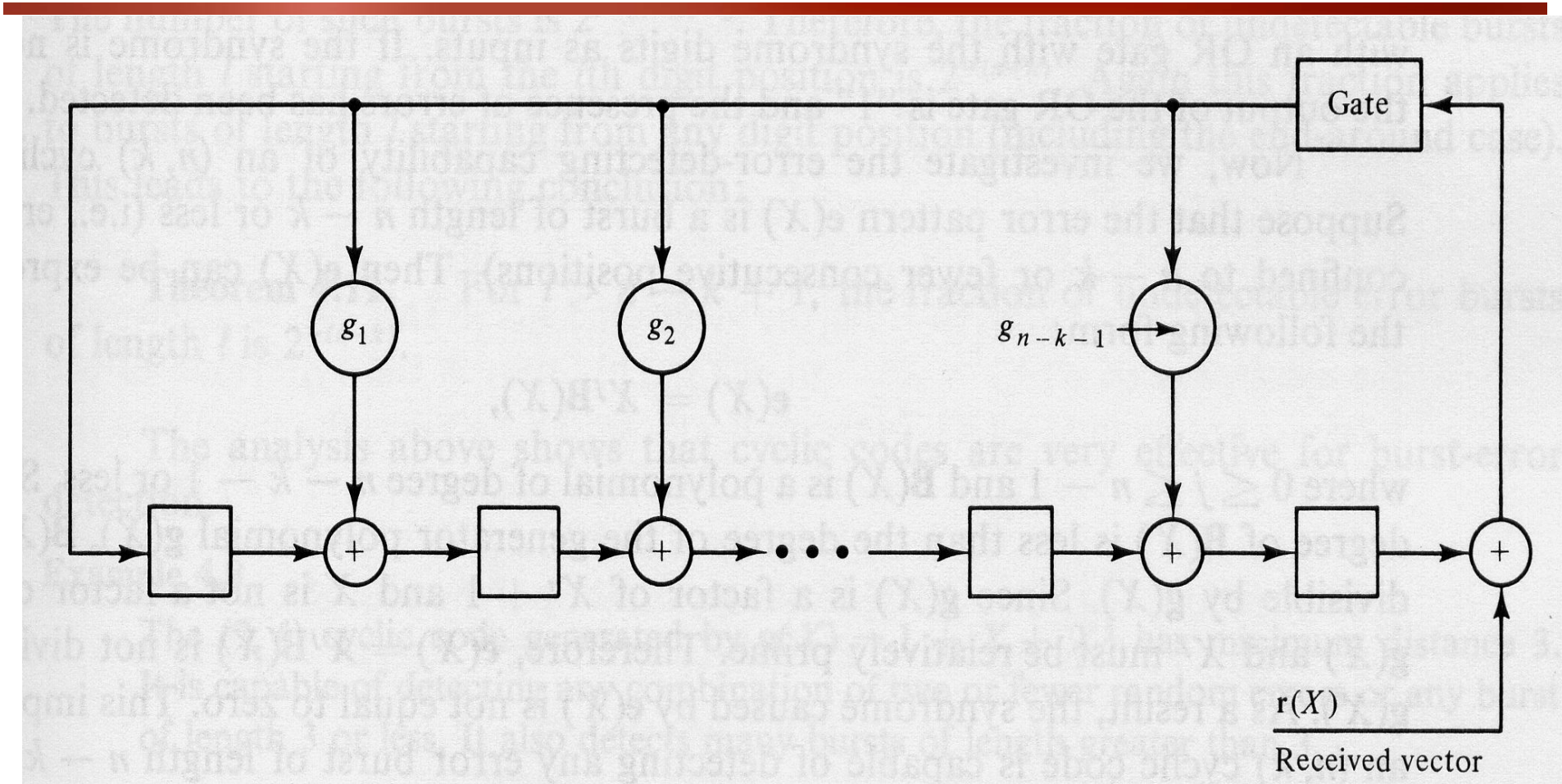

Figure 5.7 An $(n - k)$-stage syndrome circuit with input from the right end.

Syndrome Computation and Error Detection

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

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

- Combining (5.24) & (5.25) and using the fact that

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

- We have $\mathbf{r}^{(n-k)}(X) = [\mathbf{b}(X)\mathbf{h}(X) + \mathbf{a}(X)]\mathbf{g}(X) + \rho(X)$
- Therefore, $\rho(X)$ is indeed the syndrome of $\mathbf{r}^{(n-k)}(X)$

- Let $\mathbf{v}(X)$ be the transmitted code word and let $e(X)=e_0+e_1X+\ldots+e_{n-1}X^{n-1}$ be the error pattern.

Syndrome Computation and Error Detection

- The received polynomial is

$$\mathbf{r}(X)=\mathbf{v}(X)+\mathbf{e}(X)$$

- Since $\mathbf{v}(X)$ is a multiple of the generator polynomial $\mathbf{g}(X)$, we have:

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

$$\mathbf{e}(X)=[\mathbf{a}(X)+\mathbf{b}(X)]\mathbf{g}(X)+\mathbf{s}(X)$$

where $\mathbf{b}(X)\mathbf{g}(X)= \mathbf{v}(X)$.

- This shows that the syndrome is actually equal to the remainder resulting from dividing the error pattern by the generator polynomial.

Syndrome Computation and Error Detection

- Investigating the error-detecting capability of an (n,k) cyclic code
  - Suppose that the error pattern $\mathbf{e}(X)$ is a burst of length n-k or less. Then $\mathbf{e}(X)$ can be expressed in the following form:

$$\mathbf{e}(X)=X^j\mathbf{B}(X)$$

    where $0 \le j \le n$-1 and $\mathbf{B}(X)$ is a polynomial of degree n-k-1 or less.
  - Since the degree of $\mathbf{B}(X)$ is less than the degree of the generator polynomial $\mathbf{g}(X)$, $\mathbf{B}(X)$ is not divisible by $\mathbf{g}(X)$.

Syndrome Computation and Error Detection

- Since $\mathbf{g}(X)$ is a factor of X^n+1 and X is not a factor of $\mathbf{g}(X)$, $\mathbf{g}(X)$ and X^j must be relatively prime.
- $\mathbf{e}(X)=X^j\mathbf{B}(X)$ is not divisible by $\mathbf{g}(X)$.
- As a result, the syndrome caused by $\mathbf{e}(X)$ is not equal to zero.
- This implies that an (n,k) cyclic code is capable of detecting any error burst of length n-k or less.

- For a cyclic code, an error pattern with errors confined to i high-order positions and $l - i$ low-order positions is also regarded as a burst of length l or less
- Such a burst is called *end-around burst*

Syndrome Computation and Error Detection

- **Theorem 5.9** An (n, k) cyclic code is capable of detecting any error burst of length $n - k$ or less, including the end-around bursts
- **Theorem 5.10** The fraction of undetectable bursts of length $n - k + 1$ is $2^{-(n-k-1)}$
- **Theorem 5.11** for $l > n - k + 1$, the fraction of undetectable error bursts of length l is $2^{-(n-k)}$
- **Example 5.8**
  - The (7, 4) cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$ has minimum distance 3
  - It is capable of detecting any combination of two or fewer random errors or any burst of length 3 or less

Decoding of Cyclic Codes

Decoding of Cyclic Codes

- Decoding of cyclic code consists of the same three steps as for decoding linear codes:
  - Syndrome computation
  - Association of the syndrome to an error pattern
  - Error correction
- The syndrome computation for cyclic codes can be accomplished with a division circuit whose complexity is linearly proportional to the number of parity-check digits (i.e., $n - k$)
- A straightforward approach to the design of a decoding circuit is via a combinational logic circuit that implements the table-lookup procedure

Decoding of Cyclic Codes

- 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 $\mathbf{r}(X)=r_0+r_1X+r_2X^2+\ldots+r_{n-1}X^{n-1}$ 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 $\mathbf{s}(X)$ corresponds to a correctable error pattern $\mathbf{e}(X)= e_0+ e_1X+\ldots+e_{n-1}X^{n-1}$ with an

Decoding of Cyclic Codes

error at the highest-order position X^{n-1} (i.e., $e_{n-1}=1$).

- If $\mathbf{s}(X)$ 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 $\mathbf{r}^{(1)}(X)=r_{n-1}+r_0X+r_1X^2+\ldots+r_{n-2}X^{n-1}$ and the new contents in the syndrome register form the syndrome $\mathbf{s}^{(1)}(X)$ of $\mathbf{r}^{(1)}(X)$.
- The same decoding circuit will check whether $\mathbf{s}^{(1)}(X)$ corresponds to an error pattern with an error at location X^{n-1}.
- If the syndrome $\mathbf{s}(X)$ 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 $r_{n-1} \oplus e_{n-1}$.

Decoding of Cyclic Codes

- This correction results in a modified received polynomial $\mathbf{r}_1(X)=r_0+r_1X+r_2X^2+\ldots+(r_{n-1}\oplus e_{n-1})X^{n-1}$.
- The effect of the error digit e_{n-1} on the syndrome is then removed from the syndrome $\mathbf{s}(X)$.
  - This can be achieved by adding the syndrome of $\mathbf{e}'(X)= X^{n-1}$ to $\mathbf{s}(X)$.
  - This sum is the syndrome of the modified received polynomial $\mathbf{r}_1(X)$.
- Cyclically shift $\mathbf{r}_1(X)$ and the syndrome register once simultaneously.
- This shift results in a received polynomial

$$\mathbf{r}_1^{(1)}(X) =(r_{n-1}\oplus e_{n-1}) +r_0X+\ldots+r_{n-2}X^{n-1}.$$

Decoding of Cyclic Codes

- The syndrome $\mathbf{s}_1^{(1)}(X)$ of $\mathbf{r}_1^{(1)}(X)$ is the remainder resulting from dividing $X[\mathbf{s}(X)+X^{n-1}]$ by the generator polynomial $\mathbf{g}(X)$.
- Since the remainders resulting from dividing $X\mathbf{s}(X)$ and X^n by $\mathbf{g}(X)$ are $\mathbf{s}^{(1)}(X)$ and 1, respectively, we have $\mathbf{s}_1^{(1)}(X)= \mathbf{s}^{(1)}(X)+1$.
- Therefore, if 1 is added to the left end of the syndrome register while it is shifted, we obtain $\mathbf{s}_1^{(1)}(X)$.

- A general decoder for an (n, k) cyclic code is shown in Fig. 5.8
- It consists of three major parts
  - A syndrome register
  - An error-pattern detector
  - A buffer register to hold the received vector

Decoding of Cyclic Codes

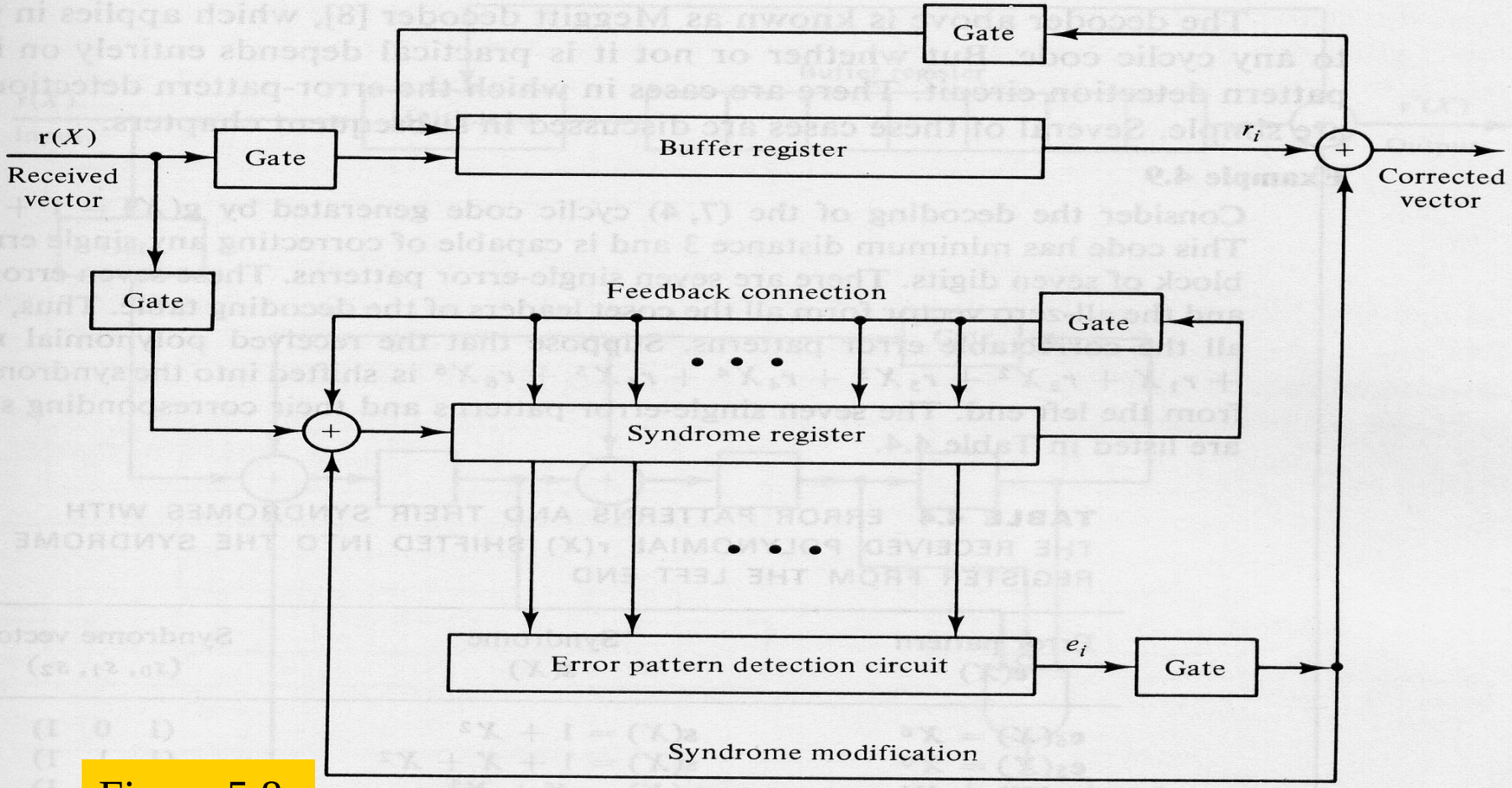

Figure 5.8 General cyclic code decoder with received polynomial $\mathbf{r}(X)$ shifted into the syndrome register from the left end.

Decoding of Cyclic Codes

- 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
- The decoding operation is described as follows:
- Step 1
  - The syndrome is formed by shifting the entire received vector into the syndrome register
  - At the same time 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

Decoding of Cyclic Codes

- Step 2 (cont.)
  - 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 highrst-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 and must be corrected
  - If a “0” appears at the output of the detector, 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

Decoding of Cyclic Codes

- Step 3
  - The first received symbol is read out of the buffer
  - If the first received symbol is detected to be an erroreous 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 erroreous symbol
  - The decoder repeats step 2 and 3

Decoding of Cyclic Codes

- 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 $\mathbf{g}(X) = 1 + X + X^3$
  - 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

Decoding of Cyclic Codes

- **Example 5.9** (cont.)
  - They form all the correctable error patterns
  - Suppose that the received polynomial

$$\mathbf{r}(X) = r_0 + r_1 X + r_2 X^2 + \ldots + r_6 X^6$$

    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 4.4 ERROR PATTERNS AND THEIR SYNDROMES WITH THE RECEIVED POLYNOMIAL r(X) SHIFTED INTO THE SYNDROME REGISTER FROM THE LEFT END

Error pattern $\mathbf{e}(X)$	Syndrome $\mathbf{s}(X)$	Syndrome vector (s_0, s_1, s_2)
$\mathbf{e}_6(X) = X^6$	$\mathbf{s}(X) = 1 + X^2$	(1 0 1)
$\mathbf{e}_5(X) = X^5$	$\mathbf{s}(X) = 1 + X + X^2$	(1 1 1)
$\mathbf{e}_4(X) = X^4$	$\mathbf{s}(X) = X + X^2$	(0 1 1)
$\mathbf{e}_3(X) = X^3$	$\mathbf{s}(X) = 1 + X$	(1 1 0)
$\mathbf{e}_2(X) = X^2$	$\mathbf{s}(X) = X^2$	(0 0 1)
$\mathbf{e}_1(X) = X^1$	$\mathbf{s}(X) = X$	(0 1 0)
$\mathbf{e}_0(X) = X^0$	$\mathbf{s}(X) = 1$	(1 0 0)

Decoding of Cyclic Codes

- The complete decoding circuit is shown in Fig. 5.9

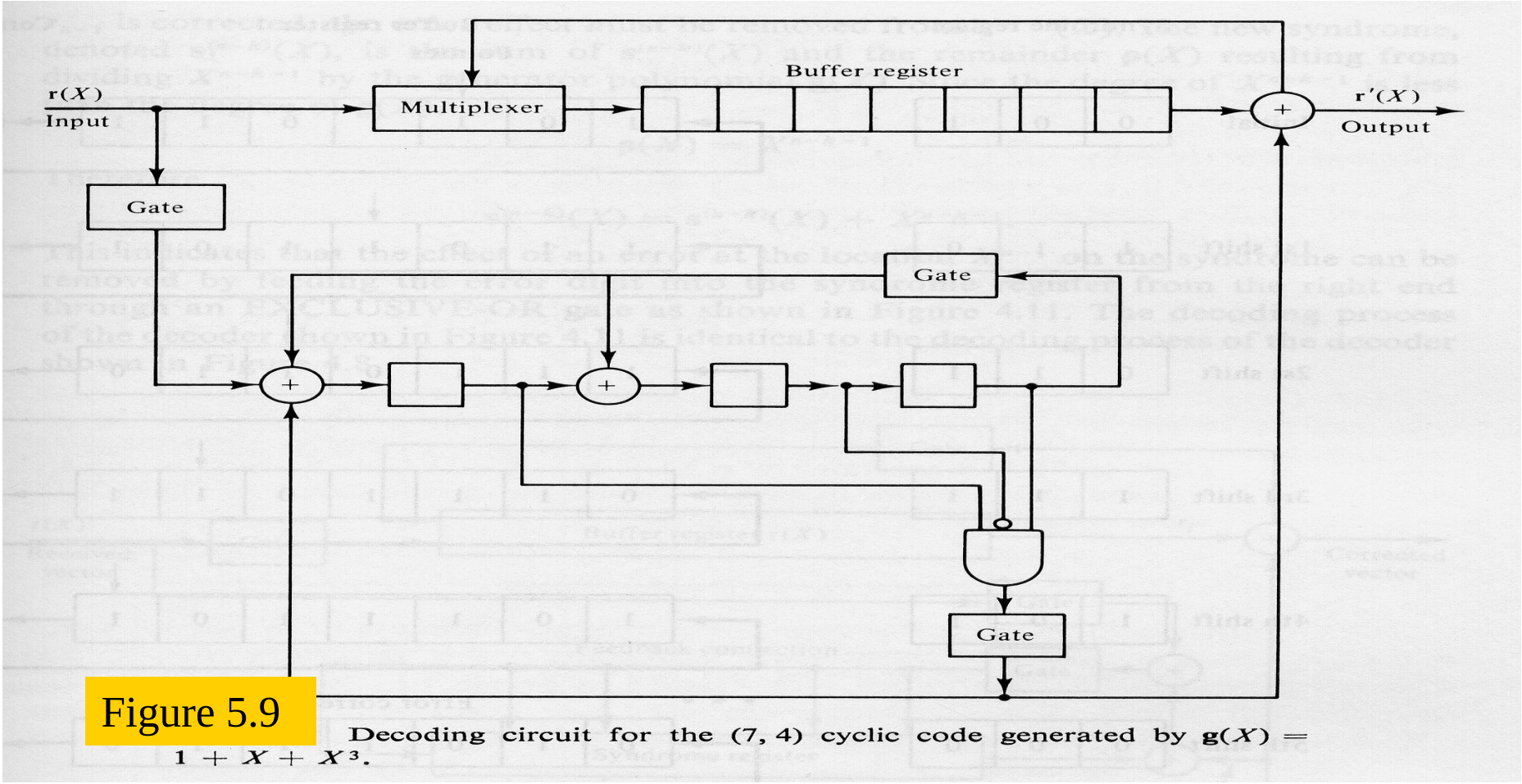

Figure 5.9 Decoding circuit for the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$.

Decoding of Cyclic Codes

- Fig. 5.10 illustrates the decoding process

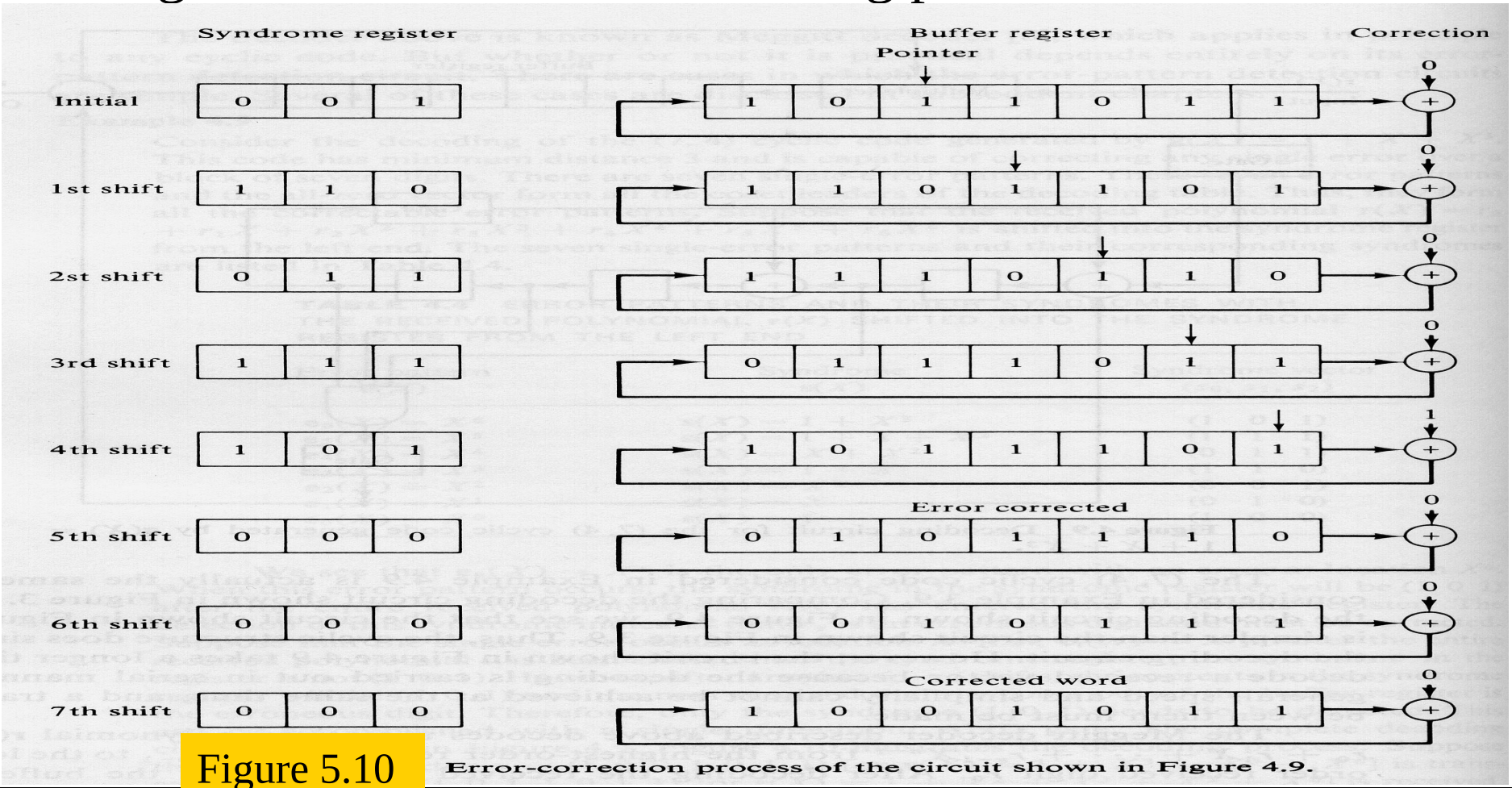

Figure 5.10 Error-correction process of the circuit shown in Figure 4.9.

Decoding of Cyclic Codes

- 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

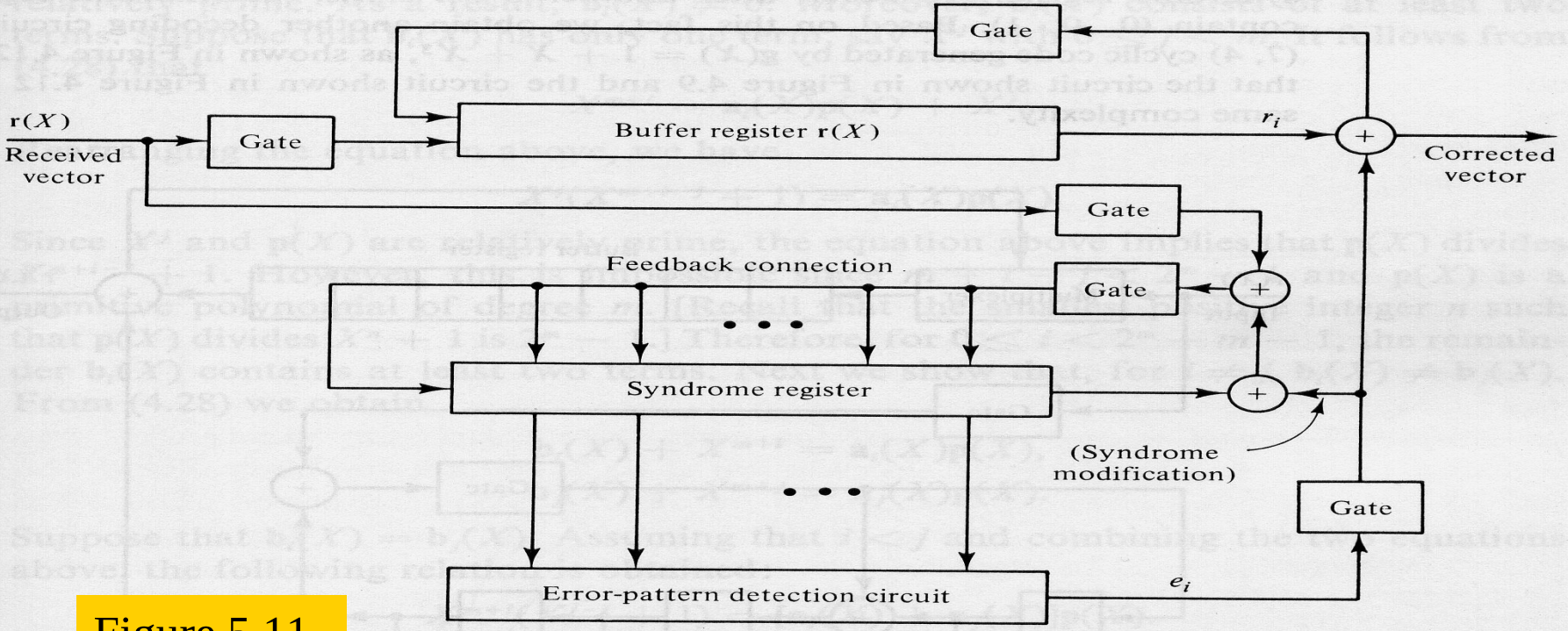

Figure 5.11 General cyclic code decoder with received polynomial $\mathbf{r}(X)$ shifted into the syndrome register from the right end.

Decoding of Cyclic Codes

- **Example 5.10**
  - Again, we consider the decoding of the (7, 4) cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$
  - Suppose that the received polynomial $\mathbf{r}(X)$ 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

Table 5.5

ERROR PATTERNS AND THEIR SYNDROMES WITH THE RECEIVED POLYNOMIAL r(X) SHIFTED INTO THE SYNDROME REGISTER FROM THE RIGHT END

Error pattern $\mathbf{e}(X)$	Syndrome $\mathbf{s}^{(3)}(X)$	Syndrome vector (s_0, s_1, s_2)
$\mathbf{e}(X) = X^6$	$\mathbf{s}^{(3)}(X) = X^2$	(0 0 1)
$\mathbf{e}(X) = X^5$	$\mathbf{s}^{(3)}(X) = X$	(0 1 0)
$\mathbf{e}(X) = X^4$	$\mathbf{s}^{(3)}(X) = 1$	(1 0 0)
$\mathbf{e}(X) = X^3$	$\mathbf{s}^{(3)}(X) = 1 + X^2$	(1 0 1)
$\mathbf{e}(X) = X^2$	$\mathbf{s}^{(3)}(X) = 1 + X + X^2$	(1 1 1)
$\mathbf{e}(X) = X$	$\mathbf{s}^{(3)}(X) = X + X^2$	(0 1 1)
$\mathbf{e}(X) = X^0$	$\mathbf{s}^{(3)}(X) = 1 + X$	(1 1 0)

Decoding of Cyclic Codes

- **Example 5.10** (cont.)
  - We see that only when $\mathbf{e}(X) = X^6$ occurs, the syndrome is (0 0 1) after the entire received polynomial $\mathbf{r}(X)$ has been shifted into the syndrome register
  - If the single error occurs at the location X^i with $i \neq 6$, the syndrome in the register will not be (0 0 1) after the entire received polynomial $\mathbf{r}(X)$ 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 $\mathbf{g}(X) = 1 + X + X^3$, 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

Decoding of Cyclic Codes

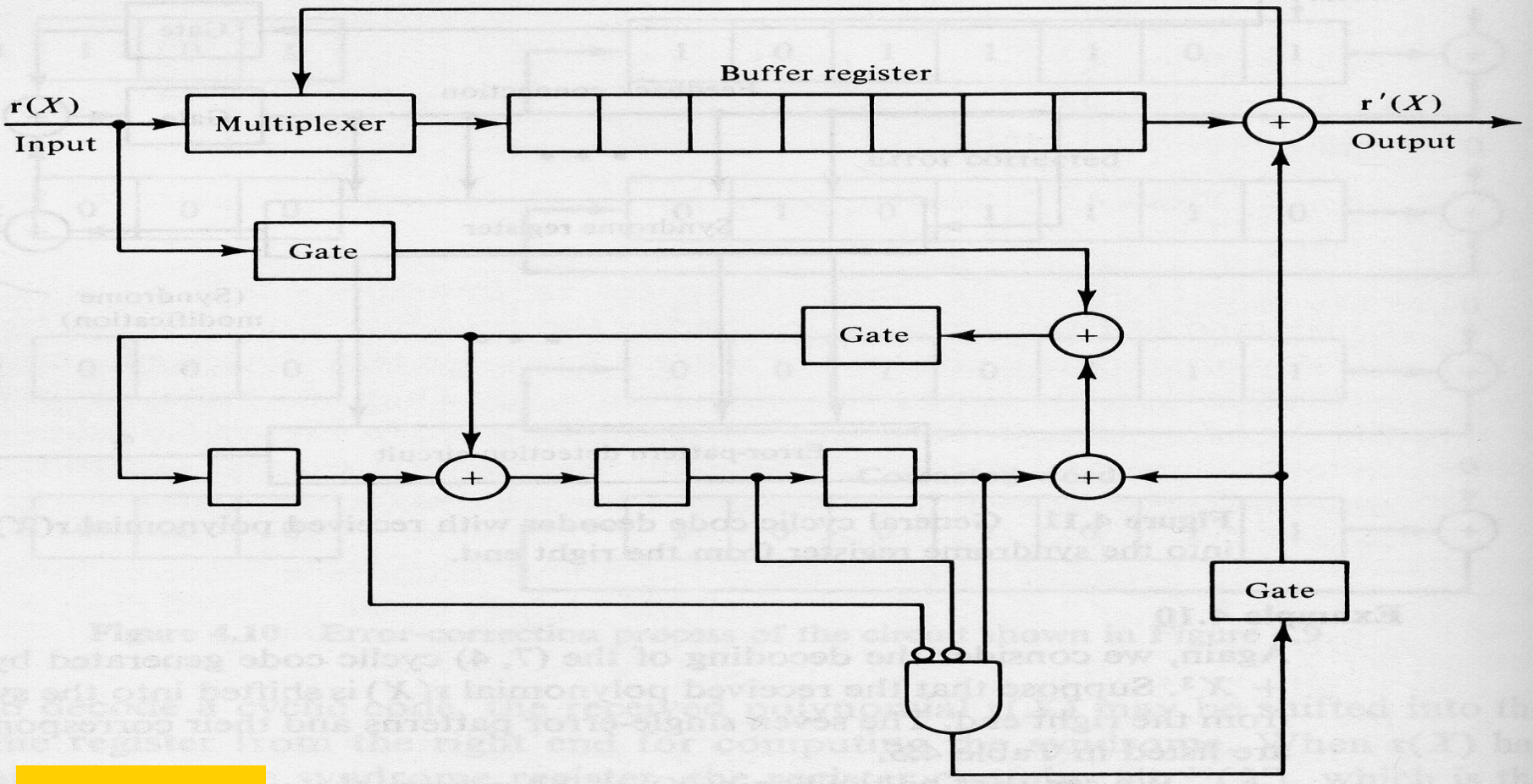

Figure 5.12 Decoding circuit for the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$.

Cyclic Hamming Codes

Cyclic Hamming Codes

- A cyclic Hamming code of length $2^m - 1$ with $m \geq 3$ is generated by a primitive polynomial $\mathbf{p}(X)$ of degree m
- In the following, we show that the cyclic code defined above is indeed a Hamming code.
- For this purpose, we examine its parity-check matrix in systematic form using method presented in Section 5.2.
- Dividing X^{m+i} by the generator polynomial $\mathbf{p}(X)$ for $0 \leq i < 2^m - m - 1$, we obtain
$$X^{m+i} = \mathbf{a}_i(X)\mathbf{p}(X) + \mathbf{b}_i(X) \qquad (5.28)$$
where the remainder $\mathbf{b}_i(X)$ is of the form
$$\mathbf{b}_i(X) = b_{i0} + b_{i1}X + \cdots + b_{i,m-1}X^{m-1}$$
- Since X is not a factor of the primitive polynomial $\mathbf{p}(X)$, X^{m+i}, and $\mathbf{p}(X)$ must be relatively prime
- As a result, $\mathbf{b}_i(X) \neq 0$

Cyclic Hamming Codes

- $\mathbf{b}_i(X)$ consists of at least two terms
- Proof
  - Suppose that $\mathbf{b}_i(X)$ has only one term, say X^j with $0 \le j < m$
  - It follows from (5.28) that
$$X^{m+i} = \mathbf{a}_i(X)\mathbf{p}(X) + X^j$$
  - Rearranging the equation above, we have
$$X^j(X^{m+i-j} + 1) = \mathbf{a}_i(X)\mathbf{p}(X)$$
  - Since X^j and $\mathbf{p}(X)$ are relatively prime, the equation above implies that $\mathbf{p}(X)$ divides $X^{m+i-j} + 1$
  - This is impossible since $m+i-j < 2^m - 1$ and $\mathbf{p}(X)$ is a primitive polynomial of degree m
  - For $0 \le i < 2^m - m - 1$, the remainder $\mathbf{b}_i(X)$ contains at least two terms

$0 \le i < 2^m - m - 1$
$0 \le j < m$
$-m < i-j < 2^m - m - 1$

Cyclic Hamming Codes

- For $i \neq j$, $\mathbf{b}_i(X) \neq \mathbf{b}_j(X)$
- Proof
  - From (5.28), we obtain

$$\mathbf{b}_i(X) + X^{m+i} = \mathbf{a}_i(X)\mathbf{p}(X)$$
$$\mathbf{b}_j(X) + X^{m+j} = \mathbf{a}_j(X)\mathbf{p}(X)$$

  - Suppose that $\mathbf{b}_i(X) = \mathbf{b}_j(X)$
  - Assuming that $i < j$ and combining the two equations above, the following relation is obtained:

$$X^{m+i}(X^{j-i} + 1) = [\mathbf{a}_i(X) + \mathbf{a}_j(X)]\mathbf{p}(X)$$

  - This equation implies that $\mathbf{p}(X)$ divides $X^{j-i} + 1$
  - This is impossible since $i \neq j$ and $j - i < 2^m - 1$
  - Therefore, $\mathbf{b}_i(X) \neq \mathbf{b}_j(X)$

Cyclic Hamming Codes

- Let $\mathbf{H}=[\mathbf{I}_m \ \mathbf{Q}]$ be the parity-check matrix of the cyclic code generated by $\mathbf{p}(X)$ where $\mathbf{Q}$ is an $m\times(2^m-m-1)$ matrix.
- Let $\mathbf{b}_i=[b_{i0},b_{i1},\ldots,b_{i,m-1}]$ be the m-tuple corresponding to $\mathbf{b}_i(X)$.
- It follows from (5.17) that the matrix $\mathbf{Q}$ has the 2^m-m-1 $\mathbf{b}_i$'s with $0 \le i < 2^m - m - 1$ as its columns.
- It follows from the analysis above that no two columns of $\mathbf{Q}$ are alike and each column has at least two 1's (minimum distance is 3).
- Therefore, the matrix $\mathbf{H}$ is indeed a parity-check matrix of a Hamming code and $\mathbf{p}(X)$ generates this code.

Cyclic Hamming Codes

- Decoding of cyclic Hamming code:
  - Suppose that a single error has occurred at the highest-order position, X^{2^m-2}, of the received vector $\mathbf{r}(X)$
  - Suppose that $\mathbf{r}(X)$ is shifted into the syndrome register from the right end
  - After the entire $\mathbf{r}(X)$ has entered the register, the syndrome in the register is equal to the remainder resulting from dividing $X^m X^{2^m-2}$ by the generator polynomial $\mathbf{p}(X)$
  - $\mathbf{s}(X)=X^{m-1}$ since $X^{2^m+m-2}=(X^{2^m-1}+1)\,X^{m-1}+X^{m-1}$ and $\mathbf{p}(X)$ divides $X^{2^m-1}+1$.
  - Therefore, if a single error occurs at the highest-order location of $\mathbf{r}(X)$, the resultant syndrome is (0,0,…,0,1).
  - A complete decoding circuit for a cyclic Hamming code is shown in Fig. 5.13

Cyclic Hamming Codes

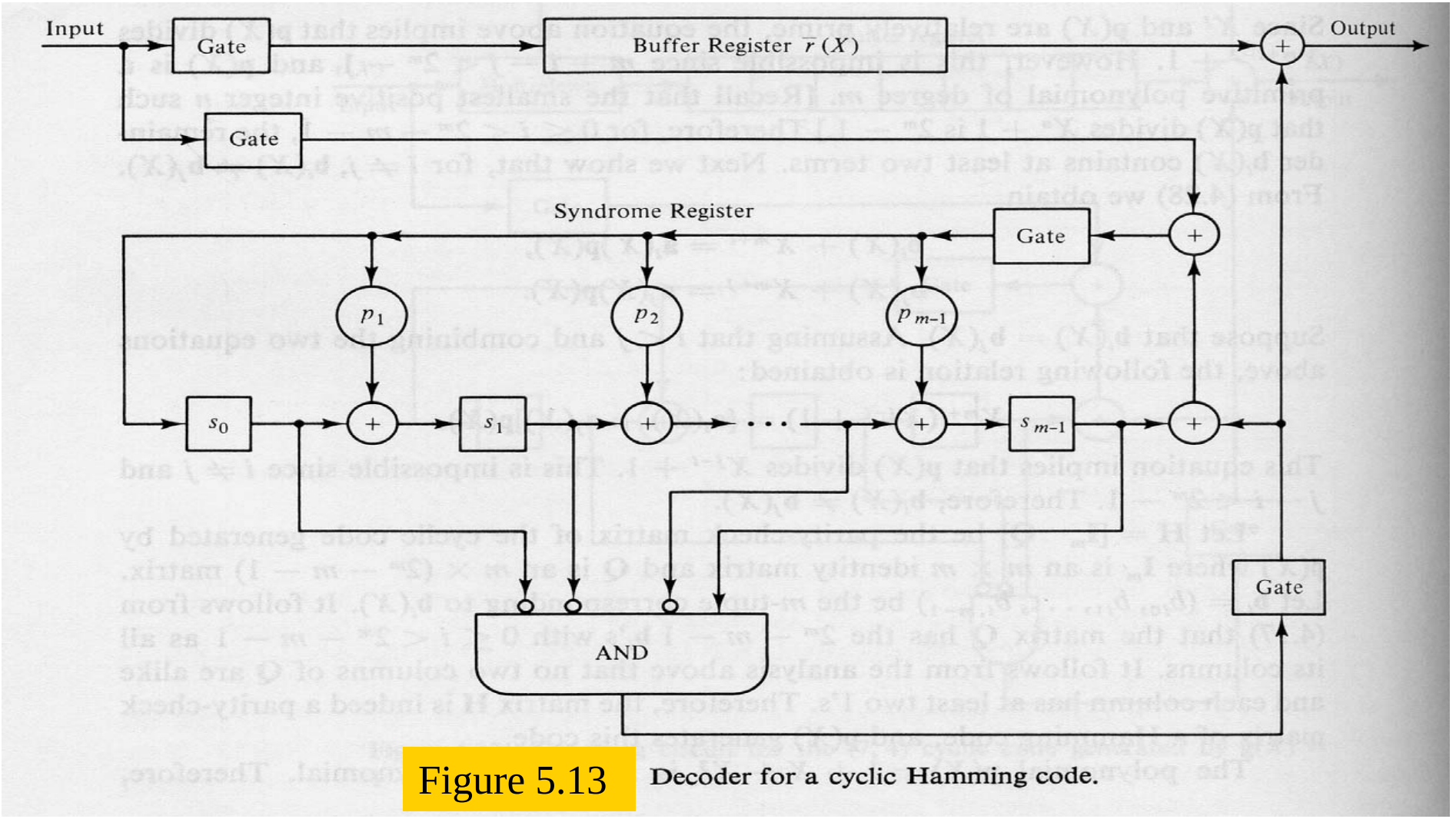

Figure 5.13 Decoder for a cyclic Hamming code.

Cyclic Hamming Codes

- The decoding operation is described in the following :
- Step 1
  - The syndrome is obtained by shifting the entire received vector into the syndrome register
  - At the same time, 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 digits by digits

Cyclic Hamming Codes

- Step 2 (cont.)
  - As soon as the syndrome in the register is (0, 0, 0, …, 0, 1), the next digit 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 is 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

Cyclic Hamming Codes

- The cyclic Hamming code presented above can be modified to correct any single error and simultaneously to detect any combination of double errors
- A single-error-correcting and double-error-detecting cyclic Hamming code of length $2^m - 1$ is generated by

$$\mathbf{g}(X) = (X + 1)\mathbf{p}(X)$$

- Because both $X+1$ and $\mathbf{p}(X)$ divide $X^{2^m-1}+1$ and since they are relative prime, $\mathbf{g}(X)$ must also divide $X^{2^m-1}+1$.
- We denote the single-error-correcting cyclic Hamming code by C_1 and denote the cyclic code generated by $\mathbf{g}(X) = (X + 1)\mathbf{p}(X)$ by C_2

Cyclic Hamming Codes

- Show that the minimum distance of this code is 4
- Proof
  - C_2 consists of the even-weight code vectors of C_1 as all its vectors
  - This is due to the fact that any odd-weight code polynomial in C_1 does not have $X + 1$ as a factor
  - Therefore, an odd-weight code polynomial of C_1 is not divisible by $\mathbf{g}(X) = (X + 1)\mathbf{p}(X)$ and it is not a code polynomial of C_2
  - An even-weight code polynomial of C_1 has $X + 1$ as a factor
  - Therefore, it is divisible by $\mathbf{g}(X) = (X + 1)\mathbf{p}(X)$ and it is also a code polynomial in C_2
  - As a result, the minimum weight of C_2 is at least 4

Cyclic Hamming Codes

- Proof (cont.)
  - Let i, j, and k be three distinct nonnegative integers less than $2^m - 1$ such that $X^i + X^j + X^k$ is not divisible by $\mathbf{p}(X)$
    - For example, we choose i and j, dividing $X^i + X^j$ by $\mathbf{p}(X)$, we obtain
      $$X^i + X^j = \mathbf{a}(X)\mathbf{p}(X) + \mathbf{b}(X)$$
      where $\mathbf{b}(X)$ is the remainder with degree $m - 1$ or less
    - Since $X^i + X^j$ is not divisible by $\mathbf{p}(X)$, $\mathbf{b}(X) \neq 0$
    - We choose an integer k such that, when X^k is divided by $\mathbf{p}(X)$, the remainder is not equal to $\mathbf{b}(X)$
    - Therefore, $X^i + X^j + X^k$ is not divisible by $\mathbf{p}(X)$
  - Dividing this polynomial by $\mathbf{p}(X)$, we have
    $$X^i + X^j + X^k = \mathbf{c}(X)\mathbf{p}(X) + \mathbf{d}(X) \qquad (5.29)$$

Cyclic Hamming Codes

- Proof (cont.)
  - We choose a nonnegative integer l less than $2^m - 1$ such that, when X^l is divided by $\mathbf{p}(X)$, the remainder is $\mathbf{d}(X)$, that is
$$X^l = \mathbf{f}(X)\mathbf{p}(X) + \mathbf{d}(X) \qquad (5.30)$$
  - The integer l cannot be equal to any of the three integer i, j, and k
  - Suppose that $l = i$, from (5.29) & (5.30), we obtain
$$X^j + X^k = [\mathbf{c}(X) + \mathbf{f}(X)]\mathbf{p}(X) = X^k(X^{k-j} + 1)$$
  - This implies that $\mathbf{p}(X)$ divides $X^{k-j} + 1$, which is impossible, since $k - j < 2^m - 1$ and $\mathbf{p}(X)$ is a primitive polynomial
  - Therefore $l \neq i$, similarly, we can show that $l \neq j$ and $l \neq k$

Cyclic Hamming Codes

- Proof (cont.)
  - Using this fact and combining (5.29) & (5.30), we have
$$X^i + X^j + X^k + X^l = [\mathbf{c}(X) + \mathbf{f}(X)]\mathbf{p}(X)$$
  - Since $X + 1$ is a factor of $X^i + X^j + X^k + X^l$ and it is not a factor of $\mathbf{p}(X)$, $\mathbf{c}(X) + \mathbf{f}(X)$ must be divisible by $X + 1$
  - $X^i + X^j + X^k + X^l$ is divisible by $\mathbf{g}(X) = (X + 1)\mathbf{p}(X)$
  - It is a code vector in code generated by $\mathbf{g}(X) = (X + 1)\mathbf{p}(X)$ has minimum weight (or distance) 4
  - It is capable of correcting any single error and simultaneously detecting any combination of double errors
- The decoding circuit for the single-error-correcting Hamming code shown in Fig. 5.13

Cyclic Hamming Codes

- Fig. 5.13 can be modified to decode the single-error-correcting and double-error-detecting Hamming code
- Let $\mathbf{r}(X)$ be the received polynomial
- Dividing $X^m\mathbf{r}(X)$ by $\mathbf{p}(X)$ and $\mathbf{r}(X)$ by $(X + 1)$ respectively, we have

$$X^m\mathbf{r}(X) = \mathbf{a}_1(X)\mathbf{p}(X) + \mathbf{s}_p(X)$$

and

$$\mathbf{r}(X) = \mathbf{a}_2(X)(X + 1) + \sigma$$

where $\mathbf{s}_p(X)$ is of degree m–1 or less and σ is either 0 or 1

- If $\mathbf{s}_p(X) = 0$ and $\sigma = 0$, $\mathbf{r}(X)$ is divisible by $(1 + X)\mathbf{p}(X)$ and is a code polynomial
- Otherwise, $\mathbf{r}(X)$ is not a code polynomial

Cyclic Hamming Codes

- We define the syndrome of $\mathbf{r}(X)$ as

$$\mathbf{s}(X) = X\mathbf{s}_p(X) + \sigma$$

- If a single error occurs, $\mathbf{s}_p(X) \neq 0$ and $\sigma = 1$
- When an error pattern with double errors occurs, we have $\mathbf{s}_p(X) \neq 0$ and $\sigma = 0$
- Based on these facts, we implement a decoder for a single-error-correcting and double-error-detecting cyclic Hamming code as shown in Fig. 5.14

Cyclic Hamming Codes

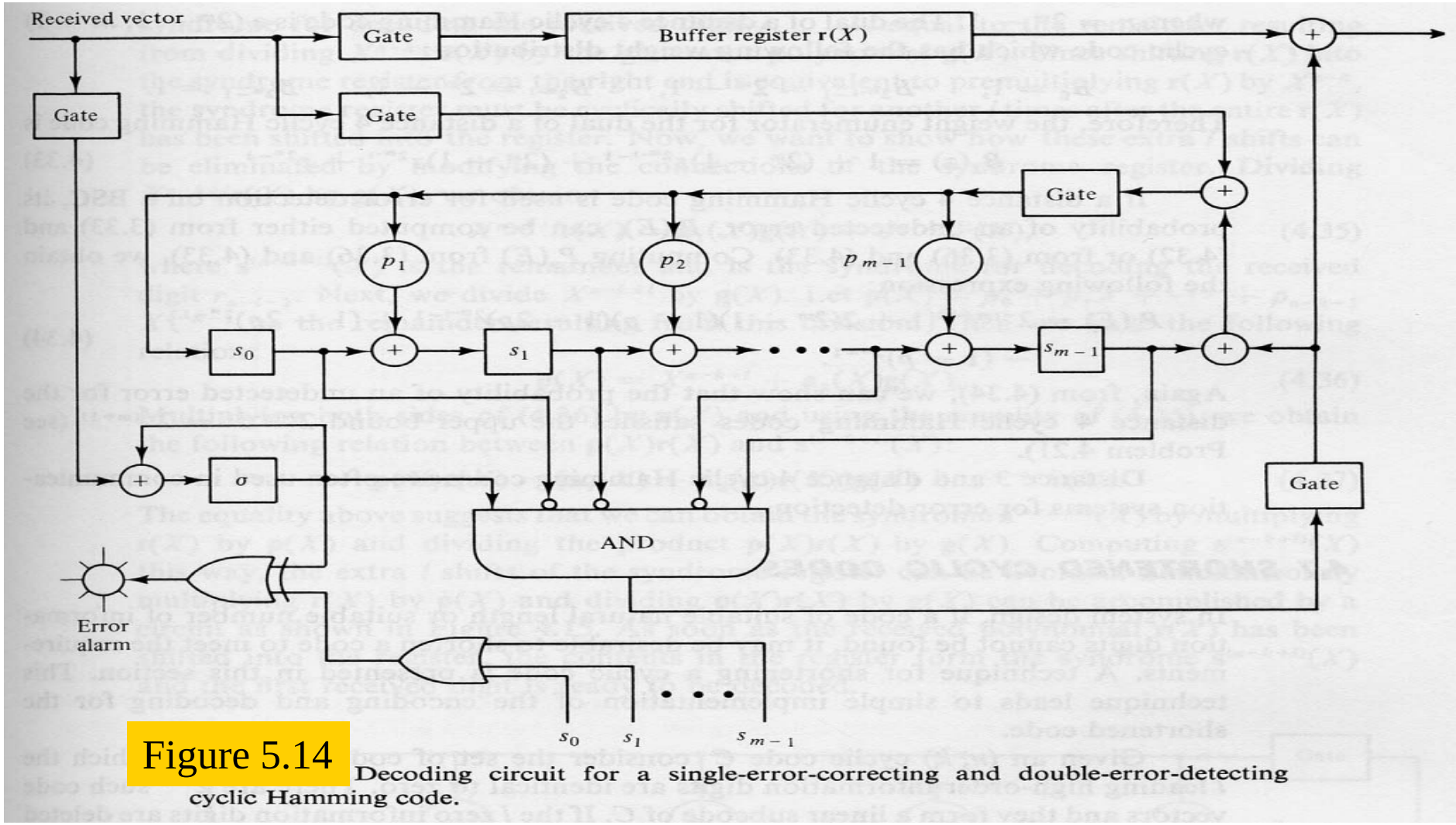

Figure 5.14 Decoding circuit for a single-error-correcting and double-error-detecting cyclic Hamming code.

Cyclic Hamming Codes

- The error-correction and error-detection operations are described as follows:
  - For $\sigma = 0$ and $\mathbf{s}_p(X) = 0$, the decoder assumes that there is no error in the received polynomial
  - For $\sigma = 1$ and $\mathbf{s}_p(X) \neq 0$, the decoder assumes that a single error has occurred and proceeds with the error-correction process as described in the decoding of a single-error-correcting cyclic Hamming code
  - For $\sigma = 0$ and $\mathbf{s}_p(X) \neq 0$, the decoder assumes that double errors have occurred. The error alarm is turned on
  - For $\sigma = 1$ and $\mathbf{s}_p(X) = 0$, the error alarm is also turned on. This happens when an error pattern with odd number (> 1) of errors has occurred an the error pattern is divisible by $\mathbf{p}(X)$

Shortened Cyclic Codes

Shortened Cyclic Codes

- If a code of suitable natural length or suitable number of information digits cannot be found, it may be desirable to shorten a code to meet the requirements
- This technique leads to simple implementation of the encoding and decoding for the shortened code
- Given an (n, k) cyclic code C consider the set of code vectors for which the l leading high-order information digits are identical to zero
- There are 2^{k-l} such code vectors and they form a linear subcode of C

Shortened Cyclic Codes

- If the l zero information digits are deleted from each of these code vectors, we obtain a set of 2^{k-l} vectors of length $n - l$
- these 2^{k-l} shortened vectors form an $(n - l, k - l)$ linear code
- This code is called a *shortened* cyclic code (or *polynomial* code) and it is not cyclic
- The encoding and decoding for a shortened cyclic code can be accomplished by the same circuits as those employed by the original cyclic code

Shortened Cyclic Codes

- In decoding the shortened cyclic code after the entire received vector has been shifted into the syndrome register
- The syndrome register must by cyclically shifted l times to generate the proper syndrome for decoding the first received digit r_{n-l-1}
- Let $\mathbf{r}(X) = r_0 + r_1X + \cdots + r_{n-l-1}X^{n-l-1}$ be the received polynomial
- If the decoding circuit for the original cyclic code is used for decoding the shortened code, the proper syndrome for decoding the received digit r_{n-l-1} is equal to the remainder resulting from dividing $X^{n-k+l}\mathbf{r}(X)$ by $\mathbf{g}(X)$

Shortened Cyclic Codes

- Since shifting $\mathbf{r}(X)$ into the syndrome register must be cyclically shifted for another l times after the entire $\mathbf{r}(X)$ has been shifted into the register
- Now, we want to show how these extra l shifts can be eliminated by modifying the connections of the syndrome register
- Dividing $X^{n-k+l}\mathbf{r}(X)$ by $\mathbf{g}(X)$, we obtain

$$X^{n-k+l}\mathbf{r}(X) = \mathbf{a}_1(X)\mathbf{g}(X) + \mathbf{s}^{(n-k+l)}(X) \qquad (5.39)$$

where $\mathbf{s}^{(n-k+l)}(X)$ is the remainder and is the syndrome for decoding the received digit r_{n-l-1}

Shortened Cyclic Codes

- We divide X^{n-k+l} by $\mathbf{g}(X)$
- Let $\rho(X) = \rho_0 + \rho_1 X + \cdots + \rho_{n-l-1} X^{n-l-1}$ be the remainder resulting from this division, we have

$$\rho(X) = X^{(n-k+l)} + \mathbf{a}_2(X)\mathbf{g}(X) \qquad (5.40)$$

- Multiplying both sides of (5.40) by $\mathbf{r}(X)$ and using the equality of (5.39), we obtain

$$\rho(X)\mathbf{r}(X) = [\mathbf{a}_1(X) + \mathbf{a}_2(X)\mathbf{r}(X)]\mathbf{g}(X) + \mathbf{s}^{(n-k+l)}(X) \qquad (5.41)$$

- The equality above suggests that we can obtain the syndrome $\mathbf{s}^{(n-k+l)}(X)$ by multiplying $\mathbf{r}(X)$ by $\rho(X)$ and dividing the product $\rho(X)\mathbf{r}(X)$ by $\mathbf{g}(X)$
- Computing $\mathbf{s}^{(n-k+l)}(X)$ this way, the extra l shifts of the syndrome register can be avoided

Shortened Cyclic Codes

- Simultaneously multiplying $\mathbf{r}(X)$ by $\rho(X)$ and dividing the product $\rho(X)\mathbf{r}(X)$ by $\mathbf{g}(X)$ can be accomplished by a circuit as shown in Fig. 5.19
- As soon as the received polynomial $\mathbf{r}(X)$ has been shifted into the register, the contents in the register form the syndrome $\mathbf{s}^{(n-k+l)}(X)$ and the first received digit is ready to be decoded

Shortened Cyclic Codes

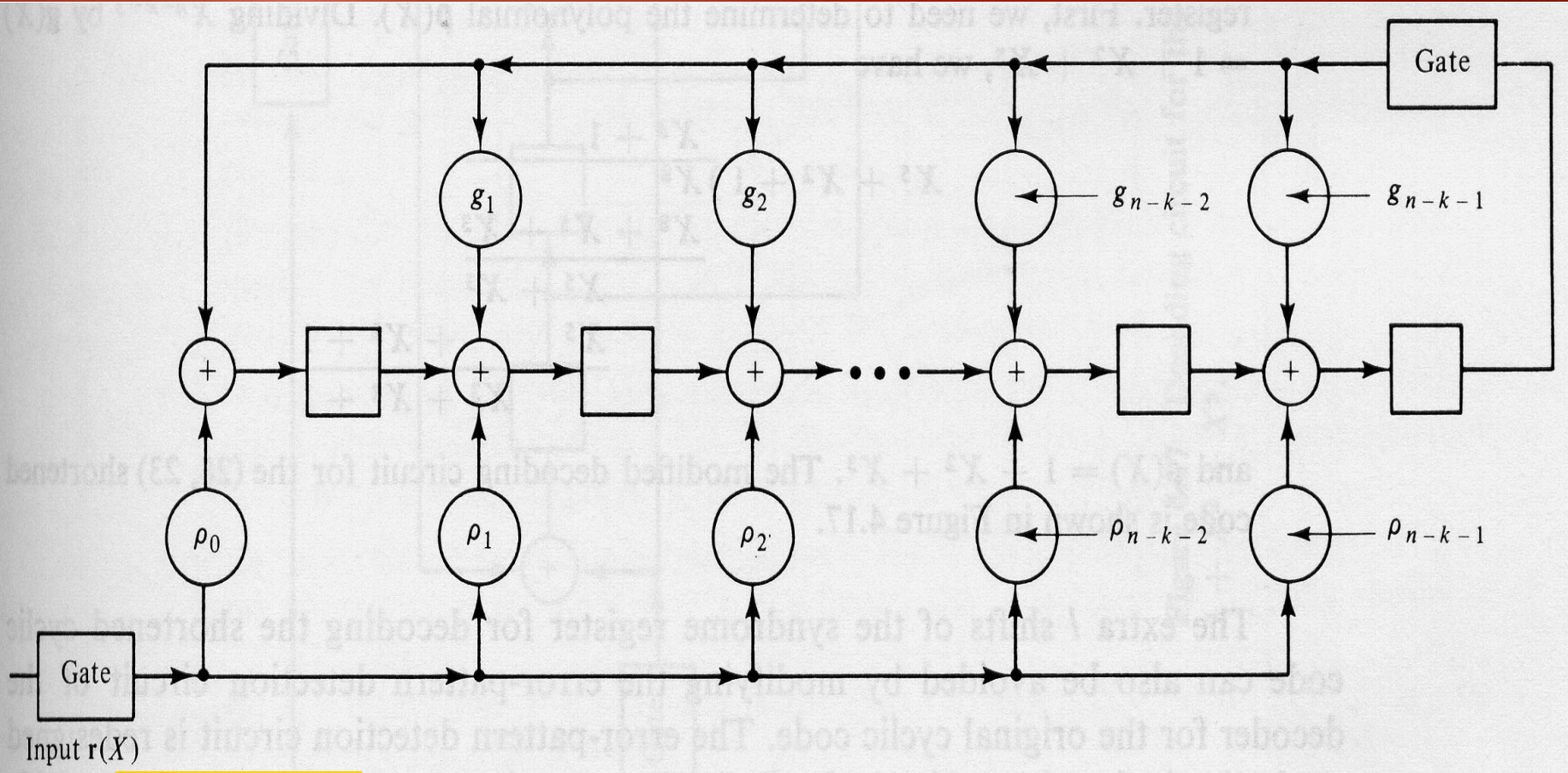

Figure 5.19 Circuit for multiplying $\mathbf{r}(X)$ by $\boldsymbol{\rho}(X) = \boldsymbol{\rho}_0 + \rho_1 X + \cdots + \rho_{n-k-1} X^{n-k-1}$ and dividing $\boldsymbol{\rho}(X)\mathbf{r}(X)$ by $\mathbf{g}(X) = 1 + g_1 X + \cdots + X^{n-k}$.

Shortened Cyclic Codes

- **Example 5.12**
  - For $m = 5$, there exists a (31, 26) cyclic Hamming code generated by $\mathbf{g}(X) = 1 + X^2 + X^5$
  - Suppose that it is shortened by three digits
  - The resultant shortened code is a (28, 23) linear code
  - The decoding circuit for the (31, 26) cyclic code is shown in Fig. 5.20.
  - This circuit can be used to decode the (28, 23) shortened code
  - To eliminate the extra shifts of the syndrome register, we need to modify the connections of the syndrome register

Shortened Cyclic Codes

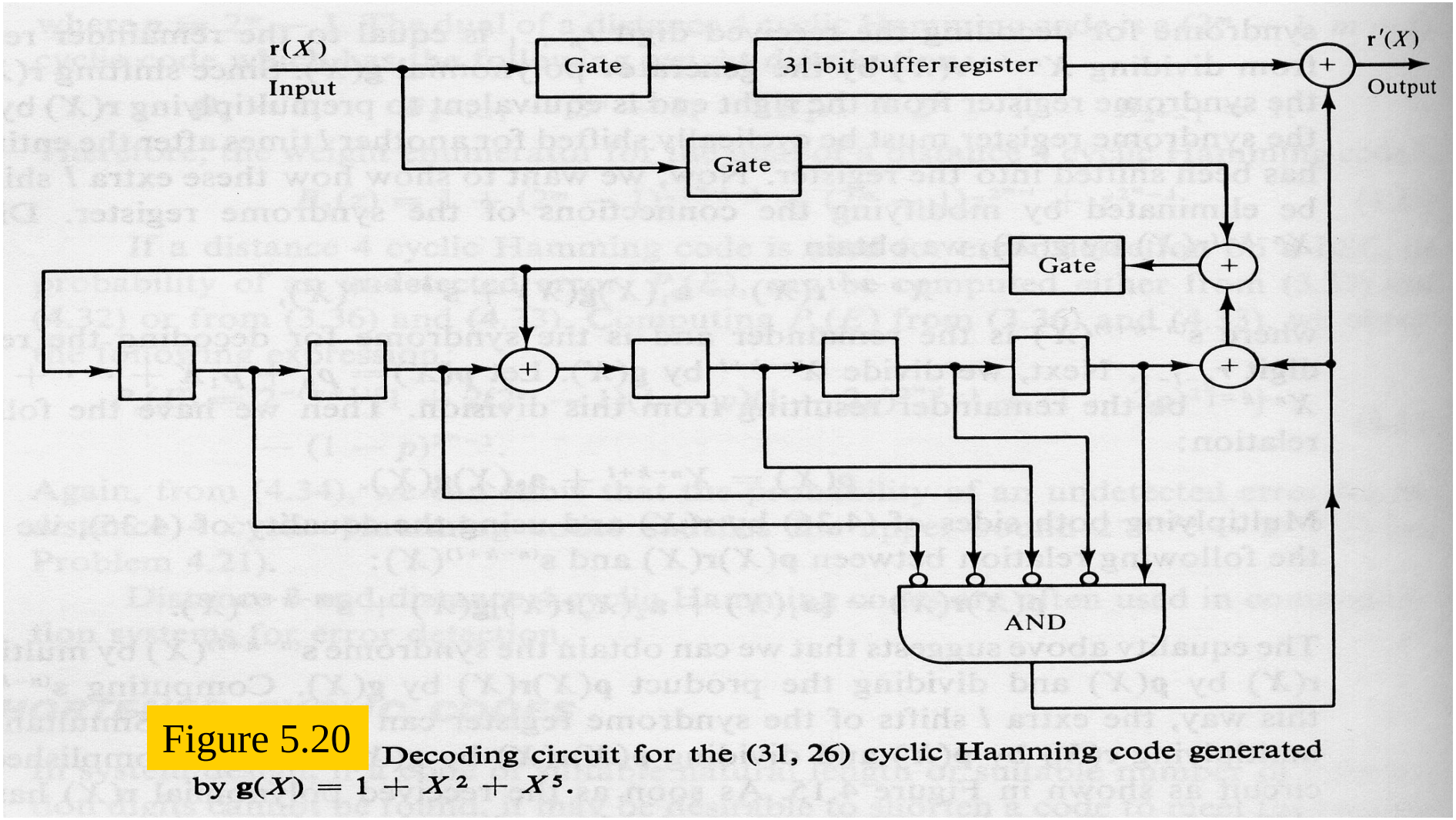

Figure 5.20 Decoding circuit for the (31, 26) cyclic Hamming code generated by $g(X) = 1 + X^2 + X^5$.

Shortened Cyclic Codes

- **Example 5.12** (cont.)
  - We need to determine the polynomial $\rho(X)$
  - Dividing X^{n-k+3} by $\mathbf{g}(X)$ $(= 1 + X^2 + X^5)$, we obtain

$$
\begin{array}{r}
X^3 + 1 \\
X^5 + X^2 + 1 \;\overline{)\; X^8} \\
\underline{X^8 + X^5 + X^3} \\
X^5 + X^3 \\
\underline{X^5 + X^2 + 1} \\
X^3 + X^2 + 1
\end{array}
$$

and $\rho(X) = 1 + X^2 + X^3$

  - The modified decoding circuit for the (28, 23) shortened code is shown in Fig. 5.21

Shortened Cyclic Codes

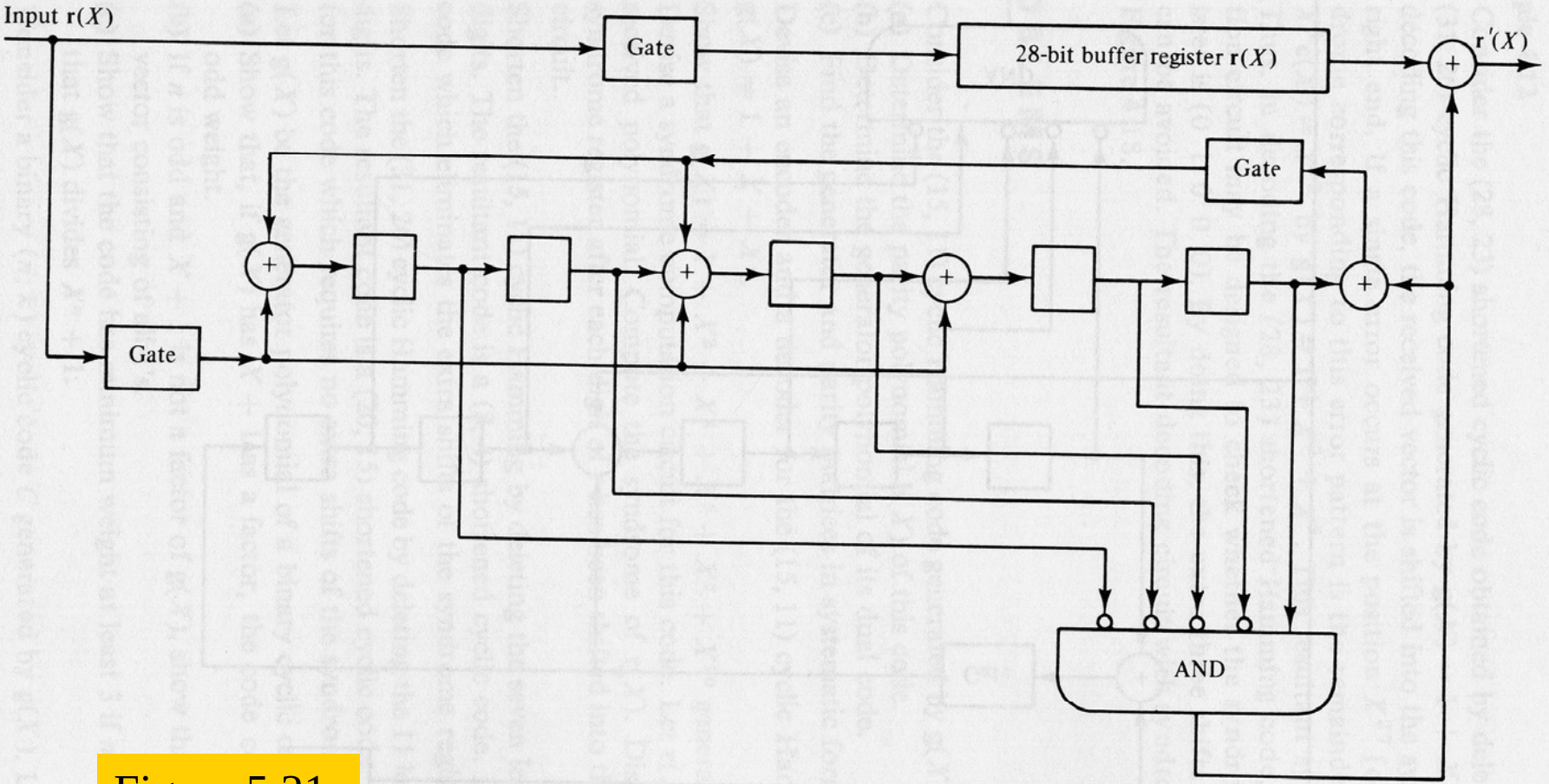

Figure 5.21 Decoding circuit for the (28, 23) shortened cyclic code generated by $g(X) = 1 + X^2 + X^5$.

Shortened Cyclic Codes

- **Example 5.13**
  - Consider the (28, 23) shortened cyclic code obtained by deleting three digits from the (31, 26) cyclic Hamming code generated by $\mathbf{g}(X) = 1 + X^2 + X^5$
  - Suppose that the received vector is shifted into the syndrome register from the right end
  - If a single error occurs at the position X^{27} (or $\mathbf{e}(X) = X^{27}$)
  - The syndrome corresponding to this error pattern is the remainder resulting from dividing $X^5\mathbf{e}(X) = X^{32}$ by $\mathbf{g}(X)$
  - This resultant syndrome is (0 1 0 0 0)
  - The decoding circuit with syndrome resetting is shown in Fig. 5.22

Shortened Cyclic Codes

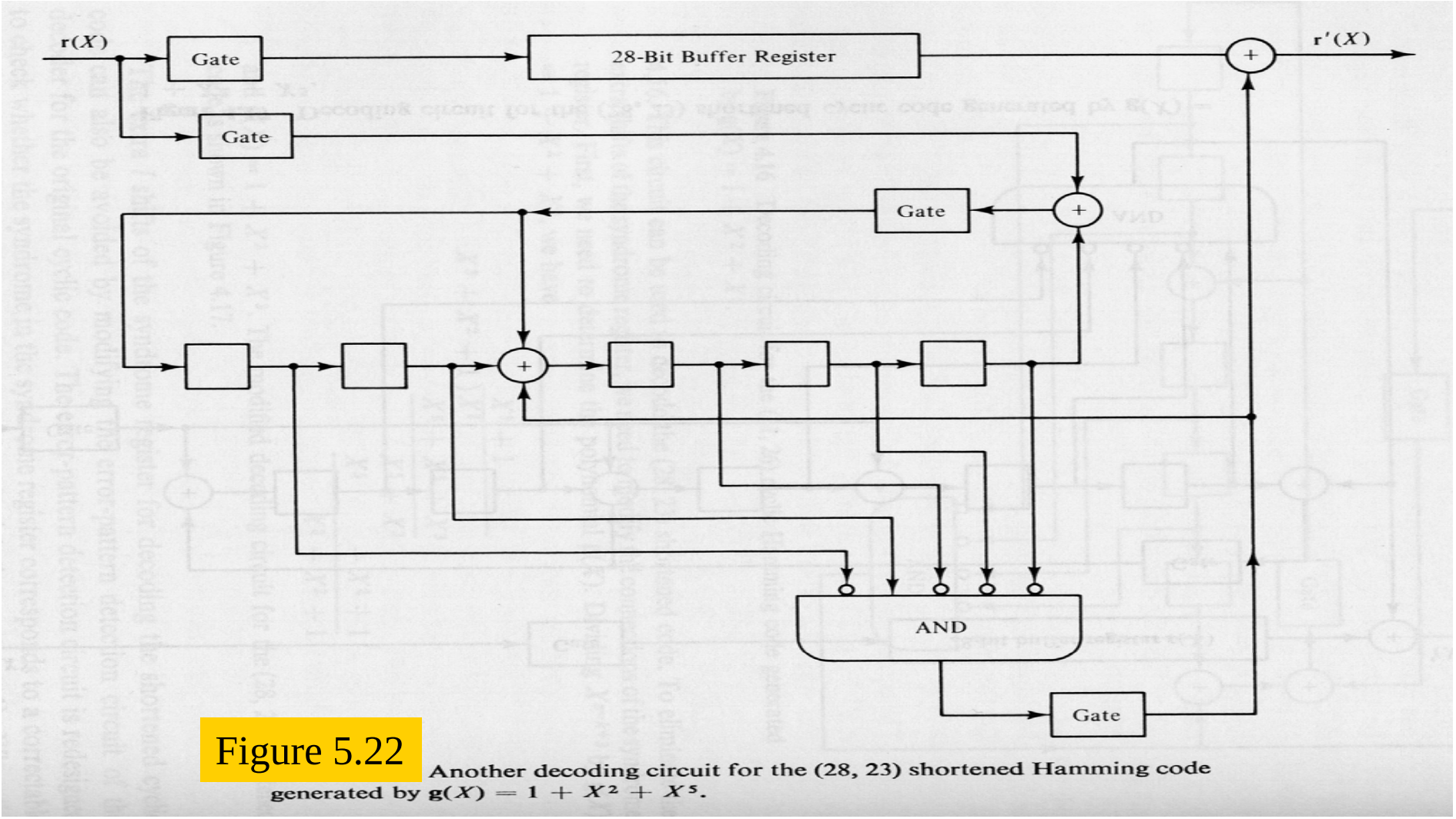

Figure 5.22 Another decoding circuit for the (28, 23) shortened Hamming code generated by $g(X) = 1 + X^2 + X^5$.