Lecture 4

Cyclic Codes

Cyclic codes form an important subclass of linear block codes. These codes are attractive for two reasons: first, encoding and syndrome computation can be implemented easily by using simple shift-registers

with feedback connections; and second, because they have considerable inherent algebraic structure, it is possible to devise various practical methods for decoding them. Cyclic codes have been widely used in communication and storage systems for error control. They are particular efficient for error detection. Many classes of cyclic codes have been constructed over years. Most well known classes of cyclic codes are BCH codes, Reed-Solomon (RS) codes and finite geometry low-density parity-check (LDPC) codes.

4.1 Definition

- Let $\mathbf{v} = (v_0, v_1, v_2, \ldots, v_{n-1})$ be an n-tuple over GF(2). If we shift every component of $\mathbf{v}$ cyclically one place to right, we obtain the following n-tuple:

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

  which is called the **right cyclic-shift** (or simply cyclic-shift) of $\mathbf{v}$.

- **Definition 4.1**: An (n,k) linear block code $\mathcal{C}$ is said to be cyclic if the cyclic-shift of a code word in $\mathcal{C}$ is another code word in $\mathcal{C}$.

- To analyze the structural properties of a cyclic code, a code word $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ is represented by a polynomial over GF(2) of degree $n-1$ or less with the components of $\mathbf{v}$ as coefficients as follows:

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

- This polynomial is called a **code polynomial**. In polynomial form, an (n,k) cyclic code $\mathcal{C}$ consists of 2^k code polynomials. The code polynomial corresponding to the all-zero code word is the zero polynomial. All the other $2^k - 1$ code polynomials corresponding to the $2^k - 1$ nonzero code words in $\mathcal{C}$ are nonzero polynomials.

4.2 Generation of Cyclic Codes

- A number of important structural properties of cyclic codes are presented in the following without proofs. References [1] contain good and extensive coverage of structure and construction of cyclic codes.
- In an (n,k) cyclic code $\mathcal{C}$, every nonzero code polynomial has degree *at least* $n-k$ but not greater than $n-1$. There exists one and only one code polynomial $\mathbf{g}(X)$ of degree $n-k$ of the following form:

$$\mathbf{g}(X) = 1 + g_1 X + g_2 X^2 + \cdots + g_{n-k-1} X^{n-k-1} + X^{n-k}. \quad (4.1)$$

- Therefore, $\mathbf{g}(X)$ is a nonzero code polynomial of **minimum degree** and is **unique**.
- Every code polynomial $\mathbf{v}(X)$ in $\mathcal{C}$ is divisible by $\mathbf{g}(X)$, i.e., a **multiple** of $\mathbf{g}(X)$. Moreover, every polynomial over GF(2) of degree $n-1$ or less that is divisible by $\mathbf{g}(X)$ is a code polynomial in $\mathcal{C}$.
- Therefore, an (n,k) cyclic code $\mathcal{C}$ is completely specified by the unique polynomial $\mathbf{g}(X)$ of degree $n-k$ given by (4.1). This unique nonzero code polynomial $\mathbf{g}(X)$ of minimum degree in $\mathcal{C}$ is called the **generator polynomial** of the (n,k) cyclic code $\mathcal{C}$. The degree of $\mathbf{g}(X)$ is simply the number of parity-check bits of the code.
- Since each code polynomial $\mathbf{v}(X)$ is $\mathcal{C}$ is a multiple of $\mathbf{g}(X)$

(including the zero code polynomial), it can be expressed as the following product:

$$\mathbf{v}(X) = \mathbf{u}(X)\mathbf{g}(X), \tag{4.2}$$

where $\mathbf{u}(X) = u_0 + u_1 X + \cdots + u_{k-1} X^{k-1}$ is a polynomial over GF(2) of degree $k-1$ or less. If $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ is the message to be encoded, then $\mathbf{u}(X)$ is the polynomial representation of $\mathbf{u}$ (called a **message polynomial**) and $\mathbf{v}(X)$ is the corresponding code polynomial of the message polynomial $\mathbf{u}(X)$.

- With this encoding, the corresponding $k \times n$ generator matrix of

the (n,k) cyclic code $\mathcal{C}$ is given as follows:

$$\mathbf{G} = \begin{bmatrix} 1 & g_1 & g_2 & \cdots & g_{n-k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & g_1 & \cdots & g_{n-k-2} & g_{n-k-1} & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & g_0 & g_1 & g_2 & \cdots & g_{n-k-1} & 1 \end{bmatrix} \tag{4.3}$$

- Note that $\mathbf{G}$ is simply obtained by using the n-tuple representation of the generator polynomial $\mathbf{g}(X)$ as the first row and its $k-1$ right cyclic-shifts as the other $k-1$ rows. $\mathbf{G}$ is not in systematic form but can be made in systematic form by elementary row operations without column permutation.
- A very important property of the generator polynomial $\mathbf{g}(X)$ of

an (n,k) cyclic code $\mathcal{C}$ over GF(2) is that $\mathbf{g}(X)$ divides $X^n + 1$. Consequently, $X^n + 1$ can be expressed as the following product:

$$X^n + 1 = \mathbf{g}(X)\mathbf{f}(X) \tag{4.4}$$

where $\mathbf{f}(X) = 1 + f_1 X + \cdots + f_{k-1} X^{k-1} + X^k$ is a polynomial of degree k over GF(2).

- Let

$$\begin{aligned} \mathbf{h}(X) &= X^k \mathbf{f}(X^{-1}) \\ &= 1 + h_1 X + \cdots + h_{k-1} X^{k-1} + X^k \end{aligned} \tag{4.5}$$

be the **reciprocal polynomial** of $\mathbf{f}(X)$.

- It is easy to prove that $\mathbf{h}(X)$ also divides $X^n + 1$.

- Form the following $(n-k) \times n$ matrix over GF(2) with the n-tuple representation $\mathbf{h} = (1, h_1, \ldots, h_{k-1}, 1, 0, \ldots, 0)$ of $\mathbf{h}(X)$ as the first row and its $n-k-1$ cyclic-shifts as the other $n-k-1$ rows:

$$\mathbf{G} = \begin{bmatrix} 1 & h_1 & h_2 & \cdots & h_{k-1} & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & h_1 & \cdots & h_{k-2} & h_{k-1} & 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & h_1 & h_2 & h_3 & \cdots & h_{k-1} & 1 \end{bmatrix}. \tag{4.6}$$

- Then $\mathbf{H}$ is a parity-check matrix of the (n,k) cyclic code $\mathcal{C}$ corresponding to $\mathbf{G}$ given by (4.3).
- The polynomial $\mathbf{h}(X)$ given by (3.29) is called the

parity-check polynomial of $\mathcal{C}$. In fact, $\mathbf{h}(X)$ is the generator polynomial of the dual code, an $(n,n-k)$ cyclic code, of the (n,k) cyclic code $\mathcal{C}$.

4.3 Systematic Encoding

- Systematic encoding of an (n,k) cyclic code with generator polynomial $\mathbf{g}(X)$ can be accomplished easily.
- Suppose $\mathbf{u} = (u_0, u_1, \ldots, u_{k-1})$ is the message to be encoded.
- Multiplying the message polynomial $\mathbf{u}(X) = u_0 + u_1 X + \cdots + u_{k-1} X^{k-1}$ by X^{n-k}, we obtain

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

which is a polynomial of degree $n-1$ or less.

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

$$X^{n-k}\mathbf{u}(X) = \mathbf{a}(X)\mathbf{g}(X) + \mathbf{b}(X) \tag{4.7}$$

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

- Since degree of $\mathbf{g}(X)$ is $n-k$, the degree of the remainder $\mathbf{b}(X)$ must be $n-k-1$ or less. Then $\mathbf{b}(X)$ must be of the following from:

$$\mathbf{b}(X) = b_0 + b_1 X + \cdots + b_{n-k-1}X^{n-k-1}. \tag{4.8}$$

- Rearranging the expression of (4.7) as follows:

$$\mathbf{b}(X) + X^{n-k}\mathbf{u}(X) = \mathbf{a}(X)\mathbf{g}(X). \tag{4.9}$$

- Expression (4.9) shows that $\mathbf{b}(X) + X^{n-k}\mathbf{u}(X)$ is divisible $\mathbf{g}(X)$.
- Since the degree of $\mathbf{b}(X) + X^{n-k}\mathbf{u}(X)$ is $n-1$ or less, $\mathbf{b}(X) + X^{n-k}\mathbf{u}(X)$ is hence a code polynomial of the (n,k) cyclic code $\mathcal{C}$ with $\mathbf{g}(X)$ as its generator polynomial. The n-tuple representation of the code polynomial $\mathbf{b}(X) + X^{n-k}\mathbf{u}(X)$ is

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

  which is in systematic form, where the $n-k$ parity-check bits, $b_0, b_1, \ldots, b_{n-k-1}$ are simply the coefficients of the remainder $\mathbf{b}(X)$ given by (4.8).
- For $0 \leq i < k$, let $\mathbf{u}_i(X) = X^i$ be the message polynomial with

a single nonzero information bit at the ith position of the message $\mathbf{u}_i$ to be encoded.

- Dividing $X^{n-k}\mathbf{u}_i(X) = X^{n-k+i}$ by $\mathbf{g}(X)$, we obtain

$$X^{n-k+i} = \mathbf{a}_i(X)\mathbf{g}(X) + \mathbf{b}_i(X) \tag{4.10}$$

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

$$\mathbf{b}_i = b_{i,0} + b_{i,1}X + \cdots + b_{i,n-k-1}X^{n-k-1}. \tag{4.11}$$

- Since $\mathbf{b}_i + X^{n-k+i}$ is divisible by $\mathbf{g}(X)$, it is a code polynomial in $\mathcal{C}$.
- Arranging the n-tuple representations of the $n-k$ code

polynomials as rows of a $k \times n$ matrix over GF(2), we obtain

$$\mathbf{G}_{c,sys} = \begin{bmatrix} b_{0,0} & b_{0,1} & \cdots & b_{0,n-k-1} & 1 & 0 & 0 & \cdots & 0 \\ b_{1,0} & b_{1,1} & \cdots & b_{1,n-k-1} & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ b_{k-1,0} & b_{k-1,1} & \cdots & b_{k-1,n-k-1} & 0 & 0 & 0 & \cdots & 1 \end{bmatrix}. \tag{4.12}$$

which is the generator matrix of the (n,k) cyclic code $\mathcal{C}$ in systematic form.

- The corresponding parity-check matrix of $\mathcal{C}$ in systematic form

is

$$\mathbf{H}_{c,sys} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & b_{0,0} & b_{1,0} & \cdots & b_{k-1,0} \\ 0 & 1 & 0 & \cdots & 0 & b_{0,1} & b_{1,1} & \cdots & b_{k-1,1} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & b_{0,n-k-1} & b_{1,n-k-1} & \cdots & b_{k-1,n-k-1} \end{bmatrix} \tag{4.13}$$

- From (4.10) and (4.11), we see that encoding of an (n,k) cyclic code $\mathcal{C}$ can be achieved with a division circuit which divides the message polynomial $X^{n-k}\mathbf{u}(X)$ by its generator polynomial $\mathbf{g}(X)$ and takes the remainder as the parity part of the code word.
- This division circuit can be implemented with an $(n-k)$-stage

shift-register with feedback connections based on the coefficients of the generator polynomial as shown in Figure 4.1. The pre-multiplication of the message polynomial $\mathbf{u}(X)$ by X^{n-k} is accomplished by shifting the message polynomial from the right-end of the encoding feedback shift-register as shown in Figure 4.1.

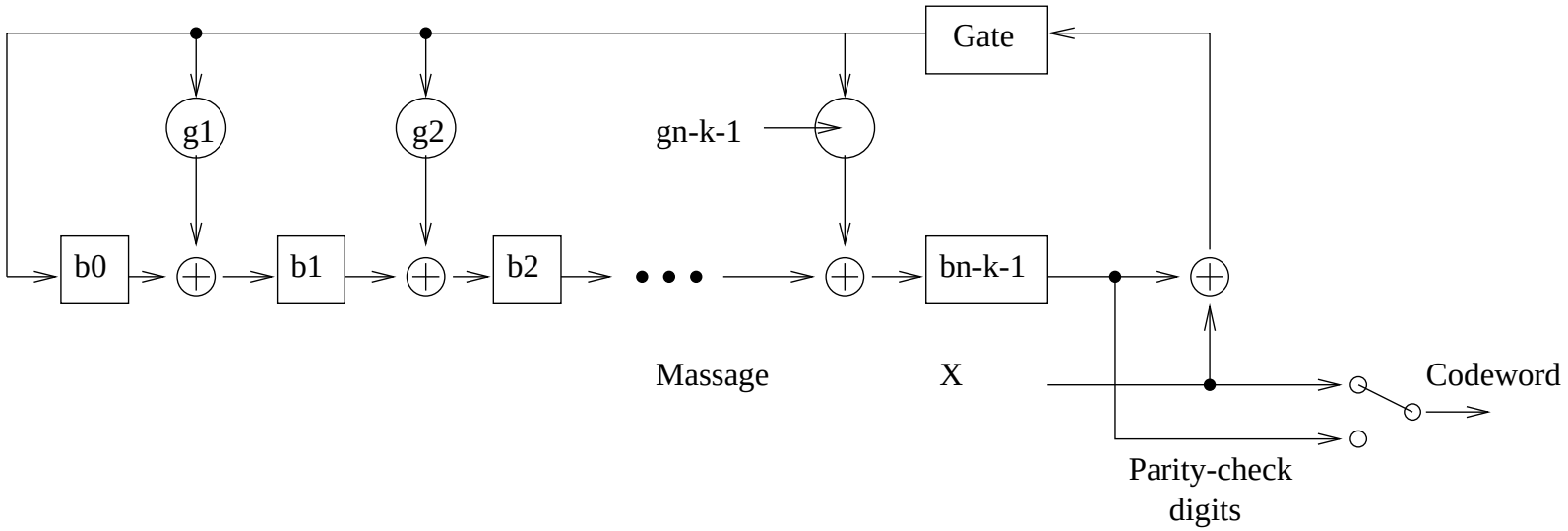

Figure 4.1: An encoding circuit for an (n,k) cyclic code with generator polynomial $\mathbf{g}(X) = 1 + g_1 X + \cdots + g_{n-k-1} X^{n-k-1} + X^{n-k}$.

- **Example 4.1**: Consider the $(7, 4)$ cyclic code with generator polynomial $\mathbf{g}(X) = 1 + X + X^3$ which divides $X^7 + 1$. Using (4.7) to (4.9), we can find its code words in systematic form as given in Table 4.1.

Table 4.1: The code words of the (7,4) cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$.

Messages	Code words	Code polynomials
(0000)	(000**0000**)	$0 = 0 \cdot \mathbf{g}(X)$
(1000)	(110**1000**)	$1 + X + X^3 = \mathbf{g}(X)$
(0100)	(011**0100**)	$X + X^2 + X^4 = X\mathbf{g}(X)$
(1100)	(101**1100**)	$1 + X^2 + X^3 + X^4 = (1 + X)\mathbf{g}(X)$
(0010)	(111**0010**)	$1 + X + X^2 + X^5 = (1 + X^2)\mathbf{g}(X)$
(1010))	(001**1010**)	$X^2 + X^3 + X^5 = X^2\mathbf{g}(X)$
(0110)	(100**0110**)	$1 + X^4 + X^5 = (1 + X + X^2)\mathbf{g}(X)$
(1110)	(010**1110**)	$X + X^3 + X^4 + X^5 = (X + X^2)\mathbf{g}(X)$
(0001)	(101**0001**)	$1 + X^2 + X^6 = (1 + X + X^3)\mathbf{g}(X)$
(1001)	(011**1001**)	$X + X^2 + X^3 + X^6 = (X + X^3)\mathbf{g}(X)$
(0101)	(110**0101**)	$1 + X + X^4 + X^6 = (1 + X^3)\mathbf{g}(X)$
(1101)	(000**1101**)	$X^3 + X^4 + X^6 = x^3\mathbf{g}(x)$
(0011)	(010**0011**)	$X + X^5 + X^6 = (X + X^2 + X^3)\mathbf{g}(X)$
(1011)	(100**1011**)	$1 + X^3 + X^5 + X^6 = (1 + X + X^2 + X^3)\mathbf{g}(X)$
(0111)	(001**0111**)	$X^2 + X^4 + X^5 + X^6 = (X^2 + X^3)\mathbf{g}(X)$
(1111)	(111**1111**)	$1 + X + X^2 + X^3 + X^4 + X^5 + X^6 = (1 + X^2 + X^3)\mathbf{g}(X)$

- The encoding circuit for the $(7, 4)$ cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$ is shown in Figure 4.2

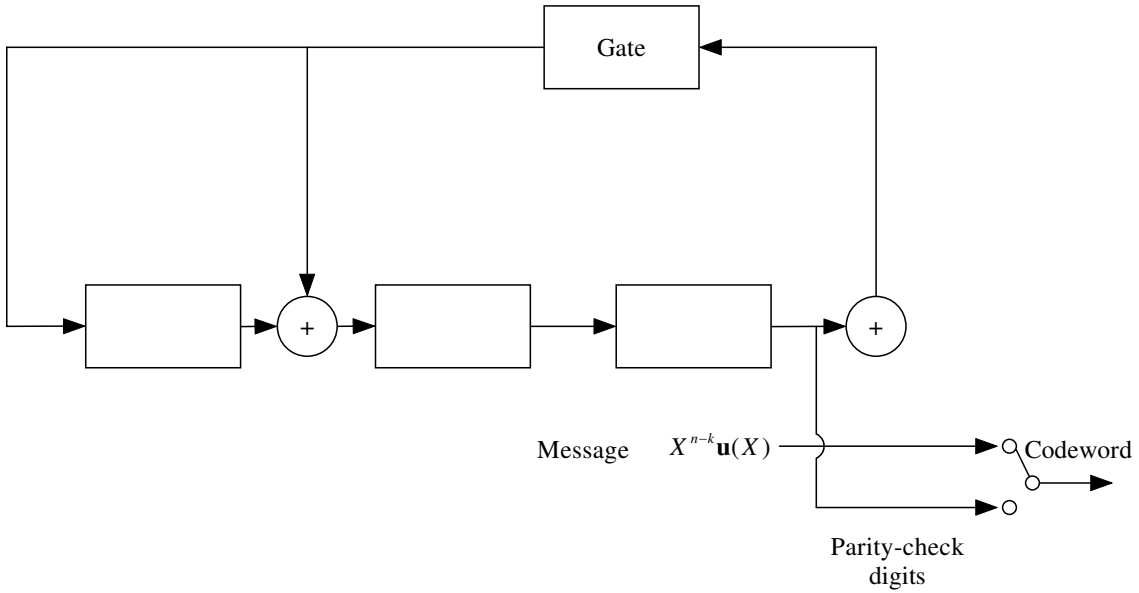

Figure 4.2: An encoding circuit for the (7, 4) cyclic code given in Example 4.1.

4.4 Decoding of Cyclic Codes

- Suppose an (n,k) cyclic code $\mathcal{C}$ with a generator polynomial $\mathbf{g}(X)$ is used for error control over a noisy channel.
- Let $\mathbf{r}(X) = r_0 + r_1 X + \cdots + r_{n-1} X^{n-1}$ be the received polynomial. Then, $\mathbf{r}(X)$ is the sum of a transmitted polynomial $\mathbf{v}(X)$ and an error polynomial $\mathbf{e}(X)$, i.e.,

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

.

- The first step in decoding $\mathbf{r}(X)$ is to compute its syndrome. The syndrome of $\mathbf{r}(X)$, denoted $\mathbf{s}(X)$, is given by the remainder obtained from dividing $\mathbf{r}(X)$ by the generator polynomial $\mathbf{g}(X)$ of code $\mathcal{C}$.
- If $\mathbf{s}(X) = 0$, then $\mathbf{r}(X)$ is a code polynomial and is accepted by the receiver as the transmitted code polynomial.
- If $\mathbf{s}(X) \neq 0$, then $\mathbf{r}(X)$ is not a code polynomial and the presence of transmission errors is detected.
- Computation of the syndrome $\mathbf{s}(X)$ of $\mathbf{r}(X)$ again can be accomplished with a division circuit as shown in Figure 4.3.

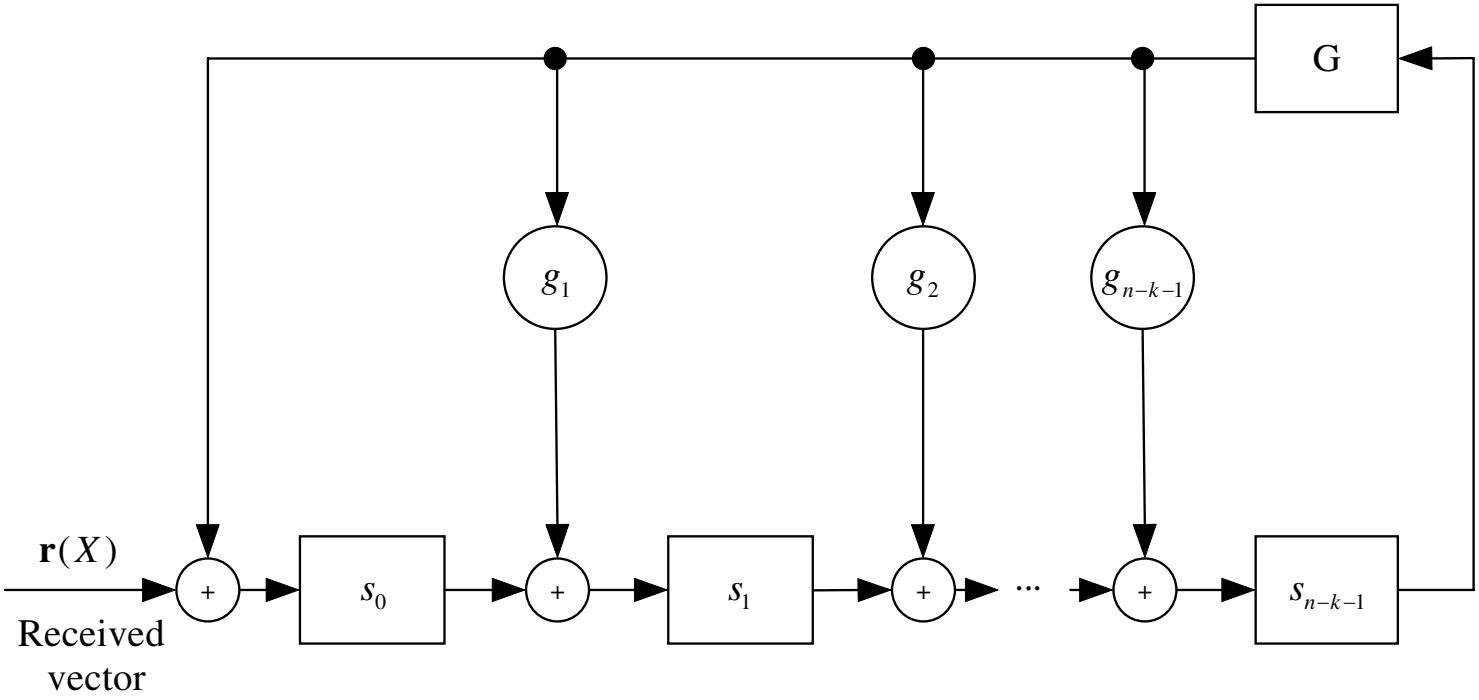

Figure 4.3: A syndrome computation circuit for an (n,k) cyclic code with generator polynomial $\mathrm{g}(X) = 1 + g_1 X + \cdots + g_{n-k-1} X^{n-k-1} + X^{n-k}$.

- **E**xample 4.2: Consider the (7,4) cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$ given in Example 4.1. A syndrome computation circuit is shown in Figure 4.4.

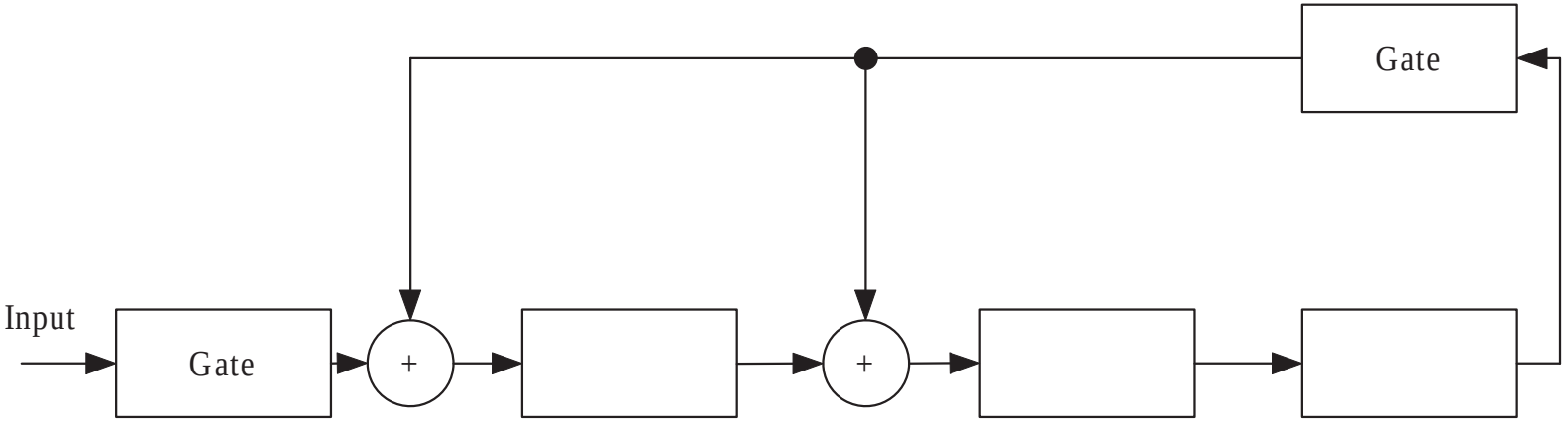

Figure 4.4: A syndrome computation circuit for the (7, 4) cyclic code given in Example 4.1.

- Similar to the decoding of a linear block code described in Lecture 3, decoding of a cyclic code is to associate the

computed syndrome $\mathbf{s}(X)$ of a received polynomial $\mathbf{r}(X)$ to a specific correctable error pattern $\mathbf{e}(X)$. Then the estimated transmitted code polynomial $\mathbf{v}(X)$ is obtained by removing the estimated error pattern $\mathbf{e}(X)$ from the received polynomial $\mathbf{r}(X)$, i.e., $\mathbf{v}(X) = \mathbf{r}(X) + \mathbf{e}(X)$. For bound distance decoding, the decoding circuit, architecture and complexity, very much depends on the decoding algorithm devised for a specific class of codes. For more on decoding of cyclic codes, readers are referred to references [1].

4.5 Shortened Cyclic Codes

- Consider a systematic (n,k) cyclic code $\mathcal{C}$.
- Let l be a nonnegative integer less than k. Consider the set of code polynomials whose l leading high-order information digits,$v_{n-l}, \ldots, v_{n-2}, v_{n-1}$, are zeros. There are 2^{k-l} such code polynomials.
- If the l zero information digits are deleted from each of these code polynomials, we obtain a set of 2^{k-l} polynomials over GF(2) with degree $n-l-1$ or less. These 2^{k-l} shortened polynomials form an $(n-l,k-l)$ linear block code. This code is called a **shortened cyclic code** (or **polynomial code**) and it is not cyclic.

- A shortened cyclic code has at least the same error-correction capability as the code from which it is shortened. The encoding and syndrome computation for a shortened cyclic code can be accomplished by the same circuits as employed by the original cyclic code. This is so because the deleted l leading zero information digits do not affect the parity-check and syndrome computation. The decoding circuit for the original cyclic code can be used for decoding the shortened code simply by prefixing each received vector with l zeros. This prefixing can be eliminated by modifying the decoding circuit.

4.6 Existence of Cyclic Codes

- So far, we have only discussed the structure, encoding and decoding of cyclic codes, nothing about the existence of cyclic codes.
- Recall that the generator polynomial $\mathrm{g}(X)$ of an (n,k) cyclic code divides $X^n + 1$.
- With this fact, it can be proved that any factor $\mathrm{g}(X)$ over GF(2) of $X^n + 1$ with degree $n - k$ can be used as a generator polynomial of an (n,k) cyclic code.
- Two large classes of well known and widely used cyclic codes will be discussed in the next three lectures.

References

1. S. Lin and D.J. Costello, Jr., *Error Control Coding, 2nd edition*. Upper Saddle River, N.J: Prentice Hall, 2004.

2. E. R. Berlekamp, *Algebraic Coding Theory*, McGraw-Hill, New York, 1968; Rev. ed., Aegean Park Press, Laguna hills, N.Y., 1984.

3. W. W. Peterson and E. J. Weldon, Jr., *Error-Correcting Codes, 2nd edition*, MIT Press, Cambridge, 1972.

Lecture 5 Cyclic Codes

5.1 Structure of Cyclic Codes

- Let

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

be an n-tuple over GF(2). If we cyclically shift every component of $\mathbf{v}$ one place to the right, we obtain another n-tuple over GF(2),

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

which is called a cyclic shift of $\mathbf{v}$.

- The i-th cyclic shift of $\mathbf{v}$ is

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

- Note that $\mathbf{v}^{(n)} = \mathbf{v}$.

- **Definition 5.1**: An (n,k) linear code is called a cyclic code if every cyclic shift of a codeword is also a codeword.

- Most of the codes being constructed are cyclic codes. The cyclic structure further simplifies the encoding and decoding implementation of a linear code.

- Cyclic codes are normally studied in terms of polynomials over GF(2) (or GF(q)).

- Note that there is a one-to-one correspondence between an n-tuple and a polynomial of degree $n-1$ or less,

$$\begin{aligned} \mathbf{v} &= (v_0, v_1, v_2, \cdots, v_{n-1}) \\ &\Longleftrightarrow \mathbf{v}(X) = v_0 + v_1 X + v_2 X^2 + \cdots + v_{n-1} X^{n-1}. \end{aligned}$$

- The polynomial corresponding to the i-th cyclic shift $\mathbf{v}^{(i)}$ of $\mathbf{v}$ is

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

- The relation between $\mathbf{v}^{(i)}(X)$ and $\mathbf{v}(X)$: $\mathbf{v}^{(i)}(X)$ is the remainder resulting from dividing $X^i \mathbf{v}(X)$ by $X^n + 1$.

$$X^i \mathbf{v}(X) = \mathbf{q}(X)(X^n + 1) + \mathbf{v}^{(i)}(X).$$

Proof:

$$\begin{aligned} X^i \mathbf{v}(X) &= v_0 X^i + v_1 X^{i+1} + \cdots + v_{n-i-1} X^{n-1} \\ &+ v_{n-i} X^n + \cdots + v_{n-1} X^{n-1+i} \\ &= v_{n-i} + v_{n-i+1} X + \cdots + v_{n-1} X^{i-1} + v_0 X^i \\ &+ v_1 X^{i+1} + \cdots + v_{n-i-1} X^{n-1} + v_{n-i}(X^n + 1) \\ &+ v_{n-i+1} X(X^n + 1) + \cdots + v_{n-1} X^{i-1}(X^n + 1) \\ &= [v_{n-i} + v_{n-i+1} X + \cdots + v_{n-1} X^{i-1}](X^n + 1) + \mathbf{v}^{(i)}(X). \end{aligned}$$

- Consider an (n,k) cyclic code C where each codeword is represented by a code polynomial of degree $n-1$ or less
- 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.
- **Theorem 5.1**: The code polynomial $\mathbf{g}(X)$ is unique.

  Proof: Suppose $\mathbf{g}(X)$ is not unique. Let $\mathbf{g}'(X)$ be another minimum-degree nonzero code polynomial. Then

  $$\mathbf{g}'(X) = g'_0 + g'_1X + \cdots + g'_{r-1}X^{r-1} + X^r.$$

  Since the code 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. If $\mathbf{g}'(X) \neq \mathbf{g}(X)$, then $\mathbf{g}(X) + \mathbf{g}'(X)$ is a nonzero code polynomial with degree less than the minimum degree r. This is impossible. Hence, $\mathbf{g}'(X) = \mathbf{g}(X)$.
- **Theorem 5.2**: The constant g_0 of $\mathbf{g}(X)$ is equal to "1".

  Proof: Suppose $g_0 = 0$. Then

  $$\mathbf{g}(X) = g_1X + g_2X^2 + \cdots + X^r.$$

Cyclically shifting $\mathbf{g}(X)$ $n-1$ times, we obtain

$$\mathbf{g}^{(n-1)}(X) = g_1 + g_2 X + \cdots + X^{r-1},$$

which has a degree less than r. This is not possible. Hence $g_0 = 1$.

- Therefore, $\mathbf{g}(X)$ must have the following form:

$$\mathbf{g}(X) = 1 + g_1 X + \cdots + g_{r-1} X^{r-1} + X^r.$$

- **Theorem 5.3**: Every code polynomial $\mathbf{v}(X)$ is a multiple of $\mathbf{g}(X)$.

  Proof: Dividing $\mathbf{v}(X)$ by $\mathbf{g}(X)$, we have

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

where

$$\begin{aligned} \mathbf{a}(X) &= a_0 + a_1 X + \cdots + a_{n-r-1} X^{n-r-1}, \\ \mathbf{b}(X) &= b_0 + b_1 X + \cdots + b_{r-1} X^{r-1}. \end{aligned}$$

Note that

$$\begin{aligned}\mathbf{a}(X)\mathbf{g}(X) &= a_0\mathbf{g}(X) + a_1 X\mathbf{g}(X) + \cdots + a_{n-r-1}X^{n-r-1}\mathbf{g}(X) \\ &= a_0\mathbf{g}(X) + a_1\mathbf{g}^{(1)}(X) + \cdots + a_{n-r-1}\mathbf{g}^{(n-r-1)}(X)\end{aligned}$$

is a code polynomial. Consequently,

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

is also a code polynomial. If $\mathbf{b}(X) \neq 0$, then $\mathbf{b}(X)$ is a nonzero code polynomial with degree less than r. This is not possible. Hence $\mathbf{b}(X) = 0$ and $\mathbf{v}(X)$ is divisible by $\mathbf{g}(X)$.

- **Theorem 5.4**: A polynomial of degree $n-1$ or less is a code polynomial if it is divisible by $\mathbf{g}(X)$.

  Proof: Let $\mathbf{v}(X)$ be a polynomial of degree $n-1$ or less,

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

  Suppose $\mathbf{v}(X) = \mathbf{a}(X)\mathbf{g}(X)$. Then

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

  and

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

  Hence $\mathbf{v}(X)$ is a code polynomial.

- **Theorem 5.5**: A polynomial $\mathbf{v}(X)$ of degree $n-1$ or less is a code polynomial if and only if it is divisible by $\mathbf{g}(X)$.

- A polynomial $\mathbf{v}(X)$ of degree $n-1$ or less which is a multiple of $\mathbf{g}(X)$ is of the form:

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

  where

  $$\mathbf{a}(X) = a_0 + a_1 X + \cdots + a_{n-r-1} X^{n-r-1}.$$

- There are 2^{n-r} such polynomials and they form all the code polynomials. Hence $2^{n-r} = 2^k$ and $r = n-k$.

- Finally,

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

which is called the generator polynomial of code C.

- **Summary**: An (n,k) cyclic code is completely specified by its generator polynomial of degree $n-k$,

$$\mathbf{g}(X) = 1 + g_1 X + g_2 X^2 + \cdots + X^{n-k}.$$

A polynomial $\mathbf{v}(X)$ of degree $n-1$ or less is a code polynomial if and only if it is divisible by $\mathbf{g}(X)$.

5.2 Existence of Cyclic Codes

- Consider an (n,k) cyclic code with generator polynomial $\mathbf{g}(X)$. Dividing $X^k\mathbf{g}(X)$ by X^n+1, we have

$$X^k\mathbf{g}(X) = (X^n+1) + \mathbf{g}^{(k)}(X). \tag{5.1}$$

Since $\mathbf{g}^{(k)}(X)$ is a code polynomial,

$$\mathbf{g}^{(k)}(X) = \mathbf{a}(X)\mathbf{g}(X). \tag{5.2}$$

Combining (5.1) and (5.2), we have

$$(X^k + \mathbf{a}(X)) \cdot \mathbf{g}(X) = X^n + 1.$$

Hence $\mathbf{g}(X) | (X^n + 1)$.

- **Theorem 5.6**: The generator polynomial $\mathbf{g}(X)$ of an (n,k) cyclic code divides $X^n + 1$.
- Let $X^n + 1 = \mathbf{g}(X) \cdot \mathbf{h}(X)$. The polynomial $\mathbf{h}(X)$ has degree k and is called the parity-check polynomial.
- Next we want to show that every factor $\mathbf{g}(X)$ of $X^n + 1$ with degree $n - k$ generates an (n,k) cyclic codes
- Consider the k polynomials, $\mathbf{g}(X), X\mathbf{g}(X), \cdots, X^{k-1}\mathbf{g}(X)$. Each has degree $n - 1$ or less. They correspond to k linearly independent n-tuples.
- Consider a linear combination of these k polynomials,

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

- Note that $\mathbf{v}(X)$ is a polynomial of degree $n - 1$ or less. There are 2^k such polynomials, and they form an (n,k) linear code, C.
- Next we show that this code is cyclic.

- Consider

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

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

$$\begin{aligned} \mathbf{v}^{(1)}(X) &= X\mathbf{v}(X) + v_{n-1}\mathbf{g}(X)\mathbf{h}(X) \\ &= X\mathbf{a}(X)\mathbf{g}(X) + v_{n-1}\mathbf{g}(X)\mathbf{h}(X) \\ &= [X\mathbf{a}(X) + v_{n-1}\mathbf{h}(X)]\mathbf{g}(X) \\ &= \mathbf{a}'(X)\mathbf{g}(X). \end{aligned}$$

- $\mathbf{v}^{(1)}(X)$ is a linear combination of $\mathbf{g}(X), X\mathbf{g}(X), X^2\mathbf{g}(X), \cdots, X^{k-1}\mathbf{g}(X)$. Therefore $\mathbf{v}^{(1)}(X)$ is also code polynomial in C. Consequently, C is cyclic

- **Theorem 5.7**: Every factor $\mathbf{g}(X)$ of $X^n + 1$ with degree $n - k$ generates an (n, k) cyclic code.

- Given $\mathbf{g}(X)$, we simply form all the polynomials of degree $n - 1$ or less which are multiples of $\mathbf{g}(X)$. They form an (n, k) cyclic code.

- **Example 5.1**: Let $n = 7$. The polynomial $X^7 + 1$ can be factored as follows:

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

$X^3 + X + 1$ or $X^3 + X^2 + 1$ generates an $(7,4)$ cyclic code. $(X + 1)(X^3 + X + 1) = X^4 + X^3 + X^2 + 1$ generates an $(7,3)$ cyclic code.

5.3 Encoding of Cyclic Codes

- Consider an (n,k) cyclic code with generator polynomial,

$$\mathbf{g}(X) = g_0 + g_1 X + \cdots + X^{n-k}.$$

- Let $\mathbf{c} = (c_0, c_1, \cdots, c_{k-1})$ be the message to be encoded. Represent $\mathbf{c}$ with a polynomial of degree $k-1$ or less,

$$\mathbf{c}(X) = c_0 + c_1 X + \cdots + c_{k-1} X^{k-1}.$$

Dividing $X^{n-k}\mathbf{c}(X)$ by $\mathbf{g}(X)$, we obtain

$$X^{n-k}\mathbf{c}(X) = \mathbf{a}(X)\mathbf{g}(X) + \mathbf{b}(X),$$

where

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

- Then

$$\mathbf{b}(x) + X^{n-k}\mathbf{c}(X) = \mathbf{a}(X)\mathbf{g}(X)$$

is a code polynomial corresponding to the message $\mathbf{c}(X)$. It is in systematic form, $\mathbf{b}(X)$ is the parity part and $X^{n-k}\mathbf{c}(X)$ is the message part.

$$(b_0, b_1, \cdots, b_{n-k-1}, c_0, c_1, \cdots, c_{k-1}).$$

- The encoding can be implemented with a division circuit which is a shift register with feedback connections based on the generator polynomial $\mathbf{g}(X)$ as shown in Figure 5.1.

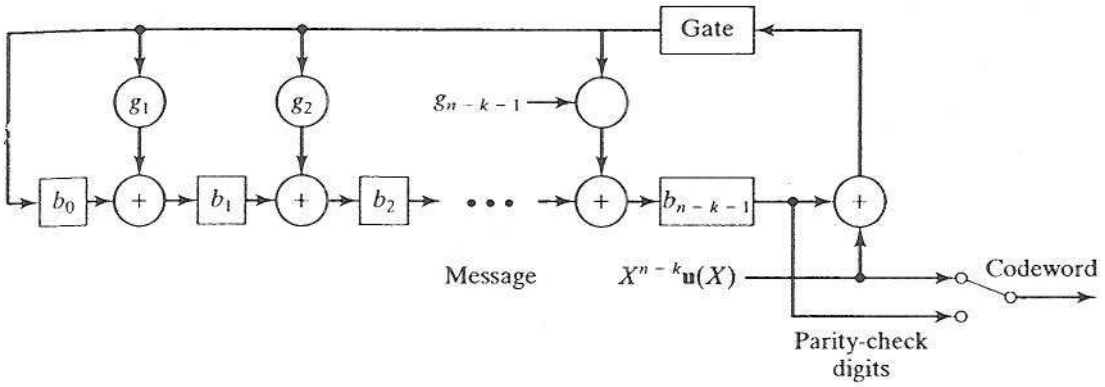

FIGURE 5.1: Encoding circuit for an (n, k) cyclic code with generator polynomial $\mathbf{g}(X) = 1 + g_1X^2 + \cdots + g_{n-k-1}X^{n-k-1} + X^{n-k}$.

- **Example 5.2**: Figure 5.2 shows the encoding circuit of the $(7,4)$ cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$.

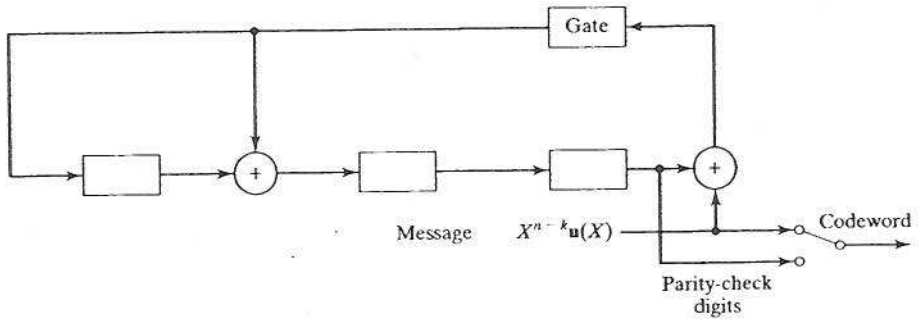

FIGURE 5.2: Encoder for the $(7, 4)$ cyclic code generated by $\mathbf{g}(X) = 1 + X + X^3$.

5.4 Syndrome Computation

- Consider an (n,k) cyclic code with generator polynomial $\mathbf{g}(X)$. Let

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

be the received polynomial. Then

$$\mathbf{r}(X) = \mathbf{v}(X) + \mathbf{e}(X) \tag{5.3}$$

where $\mathbf{v}(X)$ and $\mathbf{e}(X)$ are the transmitted code polynomial and error polynomial, respectively.

- Of course, $\mathbf{v}(X)$ and $\mathbf{e}(X)$ are unknown to the receiver.

- The syndrome of $\mathbf{r}(X)$ is the remainder $\mathbf{s}(X)$ resulting from dividing $\mathbf{r}(X)$ by $\mathbf{g}(X)$,

$$\mathbf{r}(X) = \mathbf{a}(X)\mathbf{g}(X) + \mathbf{s}(X), \tag{5.4}$$

where

$$\mathbf{s}(X) = s_0 + s_1 X + \cdots + s_{n-k-1} X^{n-k-1}.$$

- Note that $\mathbf{s}(X) = 0$ if and only if $\mathbf{r}(X)$ is a code polynomial.

- Syndrome computation can be achieved by a division circuit as shown in Figure 5.3.

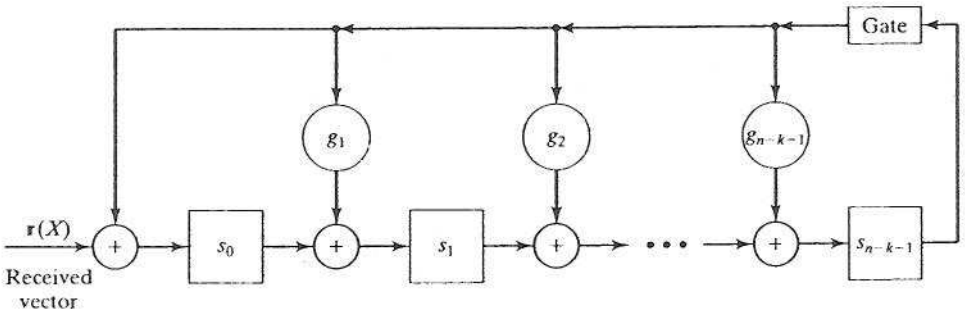

Figure 5.3

- Since $\mathbf{v}(X) = \mathbf{c}(X)\mathbf{g}(X)$, it follows from (5.3) and (5.4) that

$$\mathbf{e}(X) = (\mathbf{a}(X) + \mathbf{c}(X))\mathbf{g}(X) + \mathbf{s}(X) \tag{5.5}$$

  This gives a relationship between the syndrome $\mathbf{s}(X)$ and the error polynomial $\mathbf{e}(X)$.

- Equation (5.5) has 2^k solutions. The real error polynomial is simply one of them. Any method which uses the syndrome $\mathbf{s}(X)$ to estimate the error polynomial $\mathbf{e}(X)$ is a decoding method.

Algebraic Properties of Syndrome

- Let $\mathbf{r}^{(1)}(X)$ be the first cyclic shift of $\mathbf{r}(X)$. Then $\mathbf{r}^{(1)}(X)$ is the remainder resulting from dividing $X\mathbf{r}(X)$ by $X^n + 1$,

$$\begin{aligned} X\mathbf{r}(X) &= r_{n-1}(X^n + 1) + \mathbf{r}^{(1)}(X) \\ &= r_{n-1}\mathbf{g}(X)\mathbf{h}(X) + \mathbf{r}^{(1)}(X) \end{aligned} \tag{5.6}$$

- Let $\mathbf{s}(X)$ and $\mathbf{s}^{(1)}(X)$ be the syndromes of $\mathbf{r}(X)$ and $\mathbf{r}^{(1)}(X)$ respectively. Then, from (5.6), we have

$$\begin{aligned} X\{\mathbf{a}(X)\mathbf{g}(X) + \mathbf{s}(X)\} &= r_{n-1}\mathbf{g}(X)\mathbf{h}(X) + \mathbf{a}_1(X)\mathbf{g}(X) + \mathbf{s}^{(1)}(X), \\ X\mathbf{s}(X) &= \{r_{n-1}\mathbf{h}(X) + X\mathbf{a}(X) + \mathbf{a}_1(X)\}\mathbf{g}(X) \\ &+ \mathbf{s}^{(1)}(X). \end{aligned} \tag{5.7}$$

- **Theorem 5.8**: $\mathbf{s}^{(1)}(X)$ is the remainder resulting from dividing $X\mathbf{s}(X)$ by $\mathbf{g}(X)$.

- From this theorem, it is clear that the syndrome $\mathbf{s}^{(2)}(X)$ of the second cyclic shift $\mathbf{r}^{(2)}(X)$ of $\mathbf{r}(X)$ is the remainder resulting from dividing $X\mathbf{s}^{(1)}(X)$ by $\mathbf{g}(X)$, i.e.,

$$X\mathbf{s}^{(1)}(X) = A_2\mathbf{g}(X) + \mathbf{s}^{(2)}(X). \tag{5.8}$$

- Let $\mathbf{s}^{(i)}(X)$ be the syndrome of the i-th cyclic shift $\mathbf{r}^{(i)}(X)$ of $\mathbf{r}(X)$. Then $\mathbf{s}^{(i)}(X)$ is equal to the remainder resulting from dividing $X\mathbf{s}^{(i-1)}(X)$ by $\mathbf{g}(X)$, i.e.,

$$Xs^{(i-1)}(X) = A_i\mathbf{g}(X) + \mathbf{s}^{(i)}(X). \tag{5.9}$$

- Hence the syndrome sequence

$$\mathbf{s}(X), \mathbf{s}^{(1)}(X), \cdots, \mathbf{s}^{(n-1)}(X)$$

can be formed iteratively.

- These syndromes will be used for decoding.

- To see how to generate $s^{(1)}(X)$ from $\mathbf{s}(X)$, we express $\mathbf{s}^{(1)}(X)$ in terms of the coefficients of $\mathbf{s}(X)$ and $\mathbf{g}(X)$.

- Dividing $X\mathbf{s}(X)$ by $\mathbf{g}(X)$, the remainder is

$$\begin{aligned} \mathbf{s}^{(1)}(X) &= s_{n-k-1}g_0 + (s_0 + s_{n-k-1}g_1)X \\ &+ (s_1 + s_{n-k-1}g_2)X^2 + \cdots \\ &+ (s_{n-k-2} + s_{n-k-1}g_{n-k-1})X^{n-k-1}. \end{aligned} \tag{5.10}$$

- From (5.10), we see that $\mathbf{s}^{(1)}(X)$ is obtained by shifting (clocking) the syndrome circuit once.
- Consequently, the syndrome sequence $\mathbf{s}^{(1)}(X), \mathbf{s}^{(2)}(X), \cdots, \mathbf{s}^{(n-1)}(X)$ is obtained by clocking the syndrome circuit $n-1$ times.

5.5 General Decoding of Cyclic Codes

- Decoding of a cyclic code consists of these basic steps similar to those for decoding a linear block code:

  (1) Compute the syndrome $\mathbf{s}(X)$ of $\mathbf{r}(X)$;

  (2) Associate the syndrome $\mathbf{s}(X)$ to a correctable error pattern $\mathbf{e}(X)$.

  (3) Perform error correction, $\mathbf{r}(X) + \mathbf{e}(X)$.

- The cyclic structure allows us to decode the received polynomial $\mathbf{r}(X) = r_0 + r_1 X + r_2 X^2 + \cdots + r_{n-1} X^{n-1}$ one digit at a time with the same circuit.

- The decoding proceeds in the order: $r_{n-1}, r_{n-2}, \cdots, r_2, r_1, r_0$.

Basic Concept

(1) Decode r_{n-1} based on the syndrome $\mathbf{s}(X)$ of $\mathbf{r}(X) = r_0 + r_1 X + \cdots + r_{n-1}X^{n-1}$.

(2) Cyclic shift $\mathbf{r}(X)$ to generate

$$\mathbf{r}^{(1)}(X) = r_{n-1} + r_0 X + r_1 X^2 + \cdots + r_{n-2}X^{n-1}.$$

Decode r_{n-2} based on the syndrome $\mathbf{s}^{(1)}(X)$ of $\mathbf{r}^{(1)}(X)$.

(3) Generate

$$\mathbf{r}^{(2)}(X) = r_{n-2} + r_{n-1} X + r_0 X^2 + \cdots + r_{n-3}X^{n-1}.$$

Decode r_{n-3} based on the syndrome $\mathbf{s}^{(2)}(X)$ of $\mathbf{r}^{(2)}(X)$.

(4) Repeat the process until r_0 is decoded.

- To decode r_{n-1}, the decoder checks whether $\mathbf{s}(X)$ corresponds to a correctable error pattern,

$$\mathbf{e}(X) = e_0 + e_1 X + \cdots + e_{n-1}X^{n-1},$$

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

- If $e_{n-1} = 0$, no correction for r_{n-1} is needed. We cyclically shift $\mathbf{r}(X)$ to generate $\mathbf{r}^{(1)}(X)$. At the same, we shift the syndrome register once to form $\mathbf{s}^{(1)}(X)$. The same circuit checks whether $\mathbf{s}^{(1)}(X)$ corresponds to a correctable error pattern with an error at the position X^{n-1} of $\mathbf{r}^{(1)}(X)$

- If $e_{n-1} = 1$, r_{n-1} is an erroneous digit. Correction is done by adding e_{n-1} to r_{n-1}, i.e. $v_{n-1} = r_{n-1} \oplus e_{n-1}$. This correction results in a modified received polynomial,

$$\begin{aligned} \mathbf{r}_1(X) &= r_0 + r_1 X + \cdots + r_{n-2} X^{n-2} + (e_{n-1} + r_{n-1}) X^{n-1}. \\ &= \mathbf{r}(X) + e_{n-1} X^{n-1} \end{aligned}$$

- After correcting r_{n-1}, the effect of e_{n-1} on the syndrome $\mathbf{s}(X)$ can be removed. From the expression of $\mathbf{r}_1(X)$, we see that the syndrome of $\mathbf{r}_1(X)$, we see that the syndrome of $\mathbf{r}_1(X)$ is the sum of $\mathbf{s}(X)$ and the syndrome of $e_{n-1} X^{n-1}$.

- To decode r_{n-2}, we cyclically shift $\mathbf{r}_1(X)$ to obtain $\mathbf{r}_1^{(1)}(X)$,

$$\begin{aligned}\mathbf{r}_1^{(1)}(X) &= (e_{n-1}+r_{n-1})+r_0X+\cdots+r_{n-2}X^{n-2}\\ &= e_{n-1}+\mathbf{r}^{(1)}(X).\end{aligned}$$

- Clearly the syndrome of $\mathbf{r}^{(1)}(X)$ is

$$\mathbf{s}_1^{(1)}(X) = e_{n-1}+\mathbf{s}^{(1)}(X).$$

- Decode r_{n-2} based on $\mathbf{s}_1^{(1)}(X)$. The process is the same as decoding of r_{n-1}.

- Here we perform decoding with syndrome modification, i.e., the error effect is removed from the syndrome after each correction.

- If the error pattern is correctable, the syndrome register should contain zeros at the end of decoding.

- A general cyclic code decoder is shown in Figure 5.4.

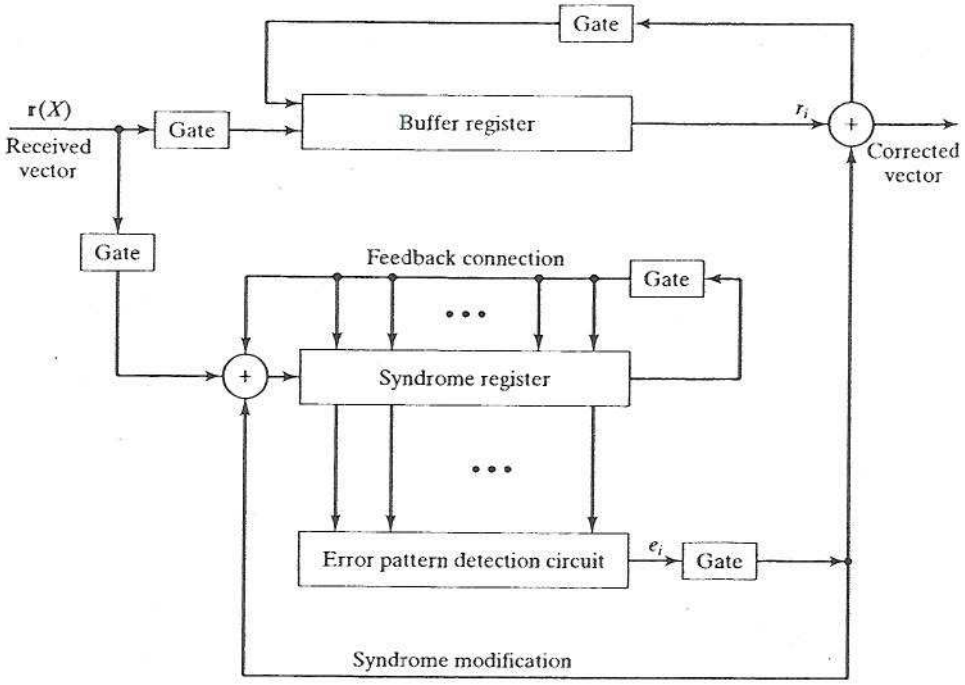
Gate
r(X)
Received vector
Gate
Buffer register
r_i
+
Corrected vector
Gate
Feedback connection
Gate
+
Syndrome register
Error pattern detection circuit
e_i
Gate
Syndrome modification

Figure 5.4

5.6 Cyclic Hamming Codes

- A cyclic Hamming code is generated by a primitive polynomial.
- The cyclic Hamming code generated by a primitive polynomial $\mathbf{p}(X)$ of degree m has the following parameters:

$$\begin{aligned} n &= 2^m - 1 \\ k &= 2^m - m - 1 \\ n-k &= m \\ d_{min} &= 3 \\ t &= 1. \end{aligned}$$

- The primitive polynomial $\mathbf{p}(X) = 1 + X + X^4$ generates a $(15, 11)$ cyclic Hamming code.

Decoding of Hamming Codes

- Consider the $(7,4)$ Hamming code generated by $\mathbf{g}(X) = 1 + X + X^3$.
- The code is capable of correcting any single error over a span of 7 bits.
- The error pattern with an error at the highest order bit position is

$$\mathbf{e}(X) = X^6$$

- The syndrome corresponding to this error pattern is the remainder resulting from dividing X^6 by the generator polynomial.

$$X^6 = \underbrace{(X^3 + X + 1)}_{Quotient}(X^3 + X + 1) + \underbrace{(X^2 + 1)}_{Remainder}$$

- Hence the syndrome of $\mathbf{e}(X) = X^6$ is

$$s = 1 + X^2$$

or $\mathbf{s}(X) = (101)$.

- In the decoding process, we check the syndrome in the syndrome register. If the syndrome is (101), the highest order bit in the buffer register is erroneous and must be corrected.
- The entire decoding circuit is shown in Figure 5.5.

Distance-4 Cyclic Hamming Codes

- It is generated by $\mathbf{g}(X) = (X+1)\mathbf{p}(X)$.
- It is a subcode of the distance-3 cyclic code generated by $\mathbf{p}(X)$.
- It consists of only the even weight codewords.
- It is capable of correcting any single error and detecting any double errors.
- It is widely used for error control.

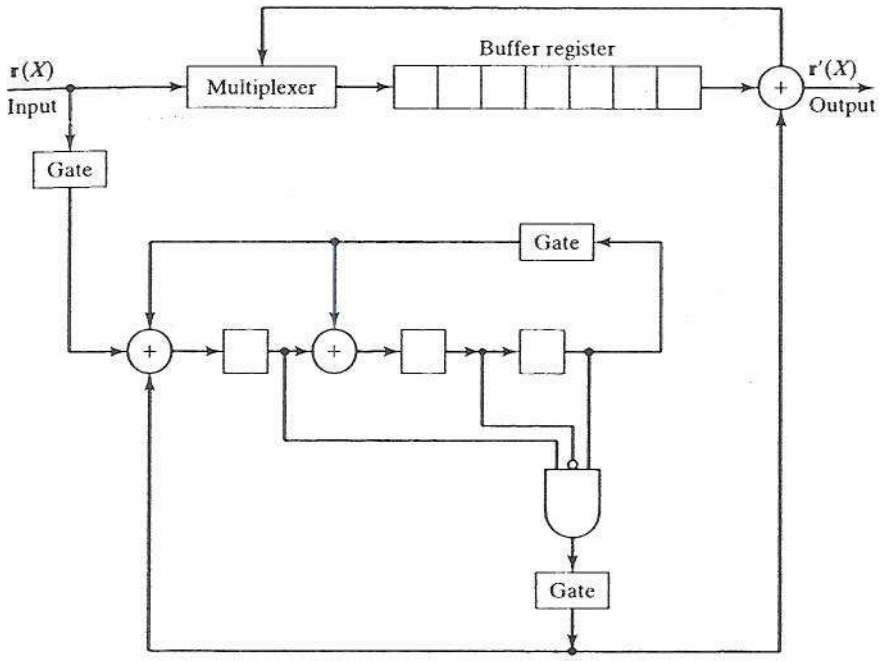

FIGURE 5.5 Decoding circuit for the (7, 4) cyclic code generated by $\mathbf{g}(X) = 1+X+X^3$.

5.7 Shortened Cyclic Codes

- In system design, often we have to shorten a code to meet the system requirements.
- Consider an (n,k) cyclic code with generator polynomial $\mathbf{g}(X)$.
- We can shorten the message and code length by l bits to obtain an $(n-l,k-l)$ shortened cyclic code. The code consists of all the code polynomials of degree $n-l-1$ which are multiples of $\mathbf{g}(X)$.
- Let $\mathbf{c}(X) = c_0 + c_1X + \cdots + c_{k-l-1}X^{k-l-1}$ be the message to be encoded.
- Dividing $X^{n-k}\mathbf{c}(X)$ by $\mathbf{g}(X)$, we have

$$X^{n-k}\mathbf{c}(X) = \mathbf{a}(X)\mathbf{g}(X) + \mathbf{b}(X).$$

- Then $\mathbf{b}(X) + X^{n-k}\mathbf{c}(X)$ is the code polynomial for $\mathbf{c}(X)$.
- Since $\mathbf{g}(X)$ may not divide $X^{n-1}+1$, the shortened cyclic code may not be cyclic.
- However, encoding and decoding of a shortened cyclic code is basically the same as that for the original cyclic code. We simply view that the l leading message bits are zeros.
- A shortened cyclic code has at least the same error correcting capability as the original code.

5.8 Important Cyclic Codes

- **Hamming** codes.
- **BCH** (Bose-Chaudhuri-Hocquenghen) codes– A large class of powerful multiple random error correcting codes, rich in algebraic structure, algebraic decoding algorithms available.
- **Goley** $(23, 12)$ code – a perfect triple-error-correcting code, widely used and generated by

$$\mathbf{g}_1(X) = 1 + X + X^2 + X^4 + X^5 + X^6 + X^{10} + X^{11}$$

or

$$\mathbf{g}_2(X) = 1 + X + X^5 + X^6 + X^7 + X^9 + X^{11}.$$

- **Finite geometry** codes – construction based on finite projective or Euclidean geometries, less efficient than BCH codes but much easier to decode.
- **Reed-Solomon** codes – nonbinary, correcting symbol errors or burst errors, most widely used for error control in data communications and data storage.
- **Fire** codes – burst-error correcting codes, easy to implement, widely used in magnetic disks for error control.
- Computer generated codes - mainly for correcting bursts of errors.

5.9 Good Error Detection Cyclic Codes

- An (n,k) linear block code is said to be good for error detection if its probability of an undetected error $P_{ud}(E)$ is upper bounded as follows:

$$P_{ud}(E) \leq 2^{-(n-k)}.$$

- Cyclic codes which have been proved to be good for error detection are:

  (1) Hamming codes.

  (2) Goley $(23,12)$ code.

  (3) Distance 5 - 8 primitive BCH codes.

  (4) Reed-Solomon codes in nonbinary case and

$$P_{ud}(E) \leq q^{-(n-k)},$$

  where q is the size of code alphabet.

5.10 The CCITT X.25 Code

- It is a distance-4 cyclic Hamming code with 16 parity-check bits for error detection for packet - switched data networks.
- It is generated by the polynomial

$$\begin{aligned} g_1(X) &= (1+X)(X^{15}+X^{14}+X^{13}+X^{12}+X^4+X^3+X^2+X+1) \\ &= X^{16}+X^{12}+X^5+1, \end{aligned}$$

or

$$g_2(X) = (X+1)(X^{15}+X^{14}+1) = X^{16}+X^{14}+X+1.$$

- The natural length of the code is $n-2^{16}-1 = 32,767$. It is usually shortened to a fewer hundred to a few thousand bits long.

5.11 The IEEE Standard 802.3 Code

- A Hamming code with 32 parity bits generated by

$$\begin{aligned} g_1(X) &= X^{32}+X^{26}+X^{23}+X^{22}+X^{16} \\ &+ X^{12}+X^{11}+X^{10}+X^8+X^7 \\ &+ X^5+X^4+X^2+X+1. \end{aligned}$$

- Used in the Ethernet.