Cyclic Codes

Contents

1. Definition
2. Generator Polynomial
3. Encoding
4. Parity Polynomial
5. Existence of Cyclic Codes
6. Irreducible Polynomial
7. Cyclic Hamming Codes
8. Syndrome Computation and Error Detection
9. Burst Error Detection with Cyclic Codes
10. Decoding of Cyclic Codes
11. Decoding of Cyclic Hamming codes
12. Shortened Cyclic Codes

Contents

13. **Important Cyclic Codes**
14. **Good Error Detection Cyclic Codes**
15. **The CCITT X. 25 Code**
16. **The IEEE Standard 802.3 Code**
17. **Error Correction Performance and MATLAB Example**

1. Definition

- An (n, k) linear code C is called a cyclic code if any cyclic shift of a codeword is another codeword. That is, if
$$\bar{v} = (v_0, v_1, v_2, ..., v_{n-1})$$
is a codeword in C, then
$$\bar{v}^{(1)} = (v_{n-1}, v_0, v_1, v_2, ..., v_{n-2})$$
obtained by **shifting** $\bar{v}$ **cyclically** one place to the right is another codeword.
- Cyclic structure makes the encoding and syndrome computation very easy.
- Cyclic codes have considerable algebraic and geometric structure. As a result, it is possible to devise various simple and efficient methods for decoding them.

2. Generator Polynomial

- Every codeword $\bar{v} = (v_0, v_1, v_2, ..., v_{n-1})$ in an (*n*, *k*) cyclic code *C* can be uniquely represented by a polynomial of degree *n* -1 or less with binary coefficients as follows:

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

- $v(X)$ is called a **code polynomial**.
- The correspondence between $\bar{v}$ and $v(X)$ is one-to-one.
- Every nonzero code polynomial $v(X)$ in *C* must have degree at least *n* - *k* but not greater than *n*-1.
- There exists **one and only one nonzero code polynomial of degree *n* - *k*** of the following form:

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

- $g(X)$ is the nonzero code polynomial of the **lowest degree**.
- Every code polynomial $v(X)$ is divisible by $g(X)$, i.e. , a multiple of $g(X)$.
- Furthermore, every polynomial of degree *n*-1 or less with binary coefficients that is divisible by $g(X)$ (or a multiple of $g(X)$) is a code polynomial.
- Hence the (*n*, *k*) cyclic code *C* is completely specified by the code polynomial $g(X)$.
- This code polynomial $g(X)$ is called the **generator polynomial** of the code.

Example 4.1: Table 4.1 gives a (7, 4) cyclic code with generator polynomial $g(X) = 1 + X + X^3$

Table 4.1

Message	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	$1 + X^3 + X^5 + X^6 = (1 + X + X^2 + X^3) \cdot g(X)$

3. Encoding

- Consider an (*n*, *k*) cyclic code *C* with generator polynomial $g(X)$
- Suppose $\bar{c} = (c_0, c_1, ..., c_{k-1})$ is the message to be encoded.
- Represent with a polynomial of degree *k*-1 or less,

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

- Multiplying $\bar{c}(X)$ by $X^{n\text{-}k}$, we obtain

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

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

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

where $b(X) = b_0 + b_1 X + ... + b_{n-k-1} X^{n-k-1}$ is the remainder.

- Then $b(X)+X^{n-k}c(X)=a(X)g(X)$ is a multiple of $g(X)$ and $c(X)$ has degree *n*-1. Hence it is the code polynomial for the message .
- Note that

$$v(X)=b(X)+X^{n-k}c(X)=$$

$$\underbrace{b_0+b_1X+\cdots+b_{n-k-1}X^{n-k-1}}_{\textit{parity check bits}}$$

$$+\underbrace{c_0X^{n-k}+c_1X^{n-k+1}+\cdots+c_{k-1}X^{n-1}}_{\textit{message bits}}$$

$$=a(X)g(X)$$

- The code polynomial is in systematic form where $b(X)$ is the parity-check part.
- The encoding can be implemented by using a division circuit which is a shift register with feed-back connections based on the generator polynomial $g(X)$ as shown in Figure 4.1.

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

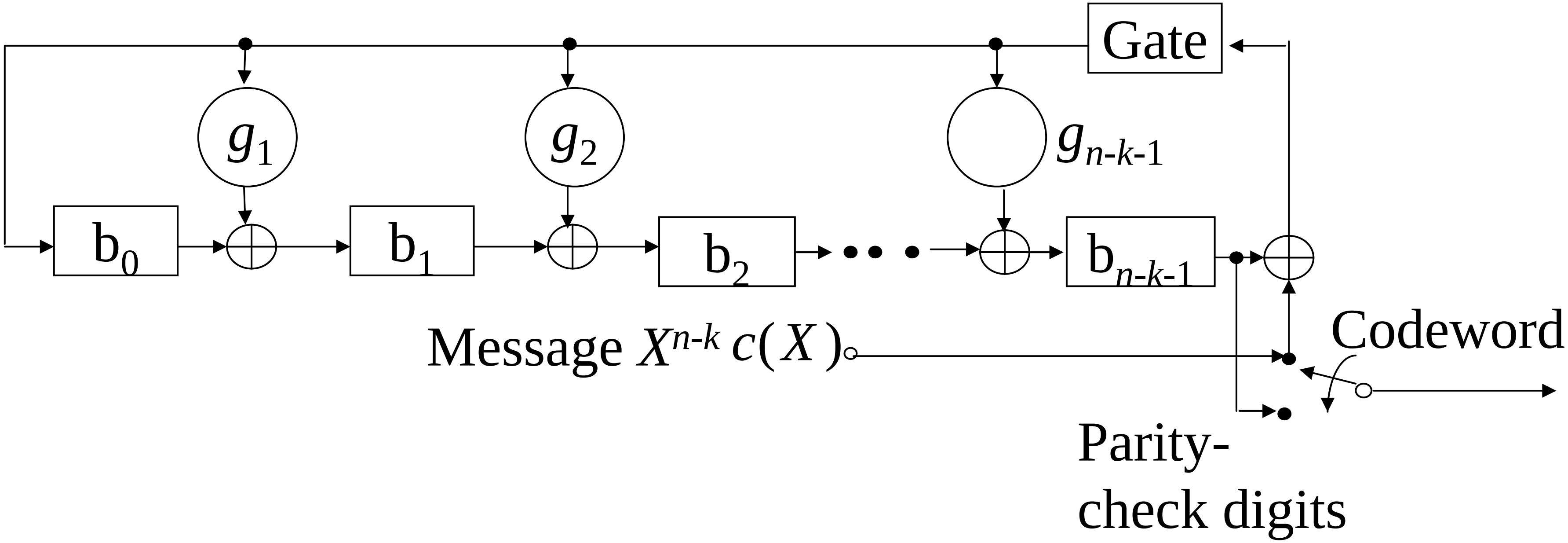

Gate
g_1
g_2
$g_{n\text{-}k\text{-}1}$
b_0
b_1
b_2
$b_{n\text{-}k\text{-}1}$
Message $X^{n\text{-}k}\, c(X)$
Codeword
Parity-check digits

Figure 4.1 An encoding circuit for an (*n*, *k*) cyclic code

Example 4.2: Figure 4.2 shows the encoding circuit of the (7, 4) cyclic code give by Table 4.1 generated by

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

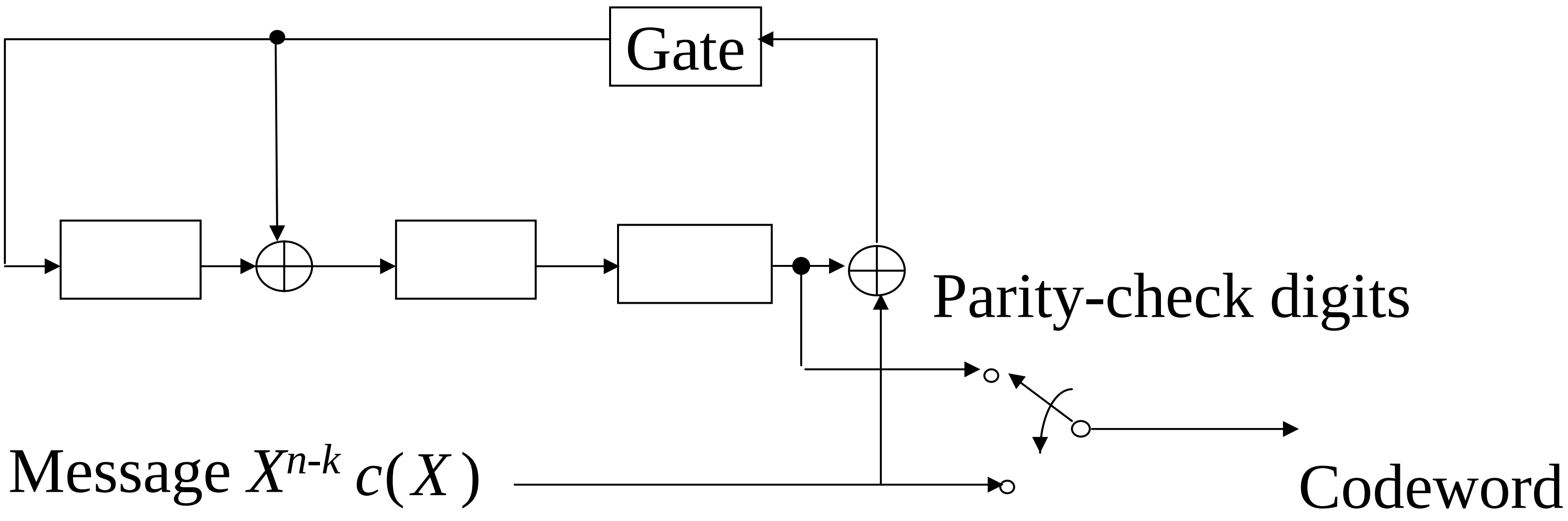


Figure 4.2 Encoder for the (7,4) cyclic code generated by

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

Table 4.1 Given $c(X) = 1 + X^3$, then the output code polynomial is $v(X) = X^6 + X^3 + X^2 + X$

timing	Register contents	Input bits
0 (Gate on)	0,0,0 (initial)	
1 (Gate on)	0,0,0	1 (X^6)
2 (Gate on)	1,1,0	0 (X^5)
3 (Gate on)	0,1,1	0 (X^4)
4 (Gate on)	1,1,1	1 (X^3)
5 (Gate on ->off)	0,1,1	(to read parity check bits)

4. Parity Polynomial

- The generator polynomial $g(X)$ of an (*n*, *k*) cyclic code divides the polynomial $X^n + 1$, i.e.,

$$X^n + 1 = g(X)h(X)$$

- The polynomial $h(X)$ is called the **parity polynomial** and has the following from:

$$h(X) = 1 + h_1 X + h_2 X^2 + \ldots + h_{k-1} X^{k-1} + X^k$$

- Encoding can be done based on $h(X)$.
- An encoding circuit based on $h(X)$ is shown in Figure 4.3.

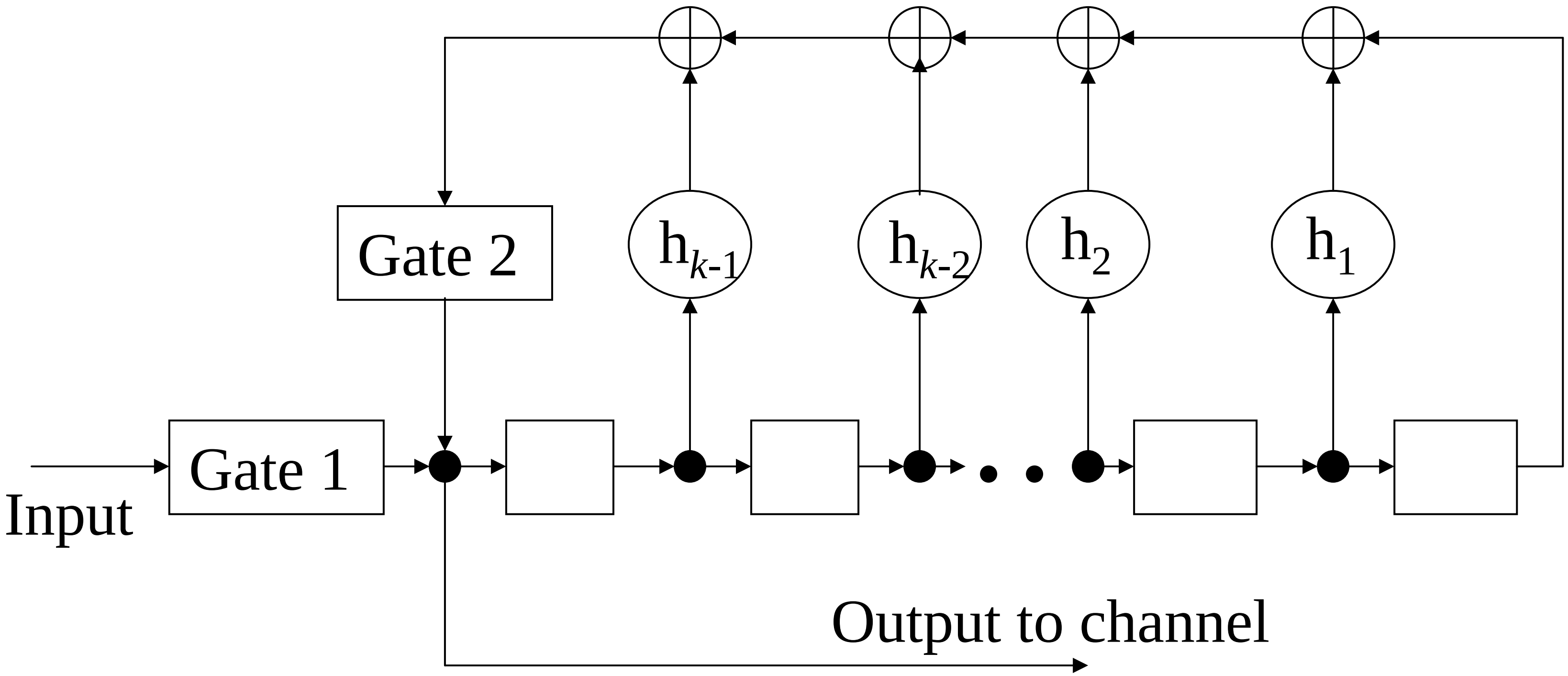


Figure 4.3 Encoding circuit for an (n, k) cyclic code based on the parity polynomial $h(X) = 1 + h_1 X + \ldots + X^k$

Example 4.3: Consider the (7, 4) cyclic code with generator polynomial $g(X) = 1 + X + X^3$. The parity polynomial of this code is

$$\begin{aligned} h(X) &= (X^7 + 1)/(X^3 + X + 1) \\ &= 1 + X + X^2 + X^4 \end{aligned}$$

The encoding circuit based on $h(X)$ is shown in Figure 4.4.

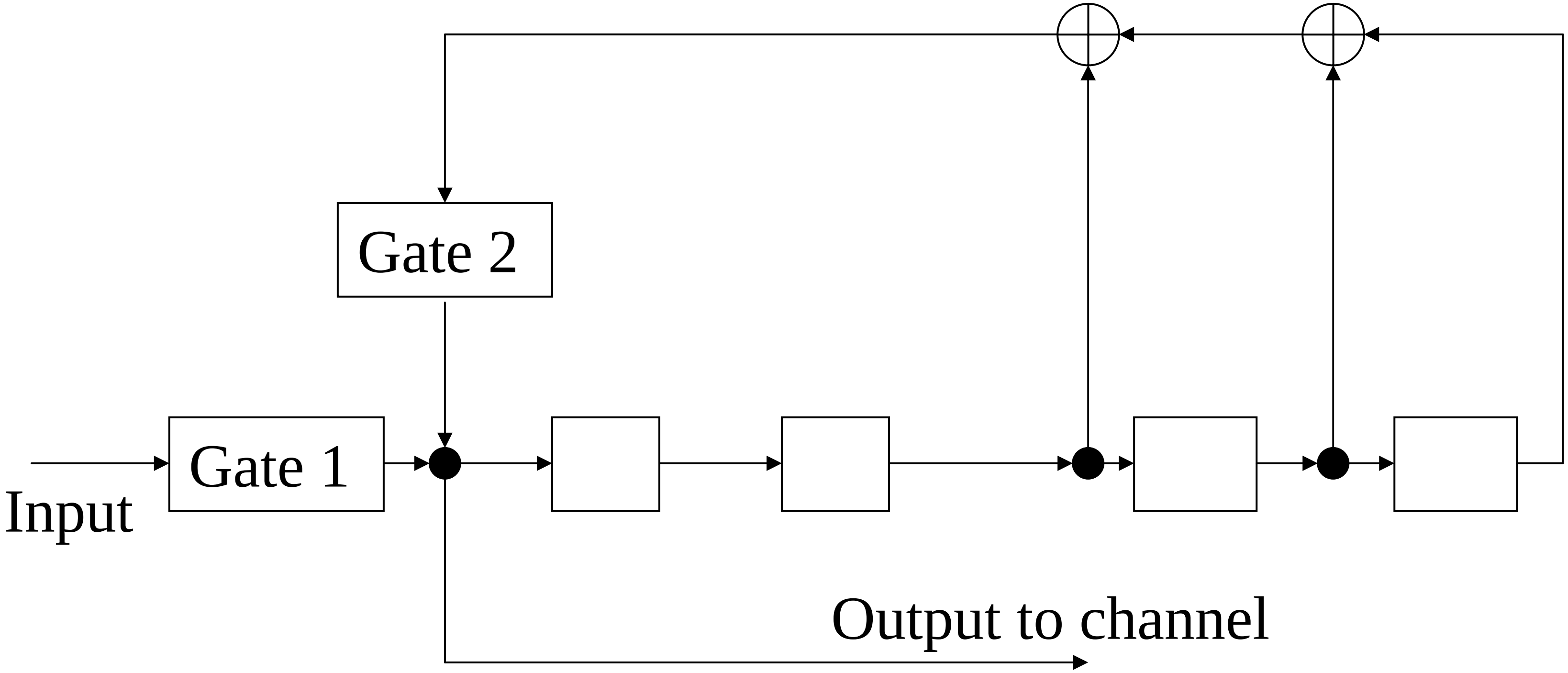


Figure 4.4 Encoding circuit for the (7, 4) cyclic code based on its parity polynomial $h(X) = 1 + X + X^2 + X^4$

Table 4.2 Given $c(X)=1+X^{3}$, then the output code polynomial is $v(X)=X^{6}+X^{3}+X^{2}+X$

Timing	Input bits		Register contents	Output bits	
0, G1on G2off			0,0,0,0(initial)		
1, G1on G2off	1	(X^{6})	0,0,0,0	1	(X^{6})
2, G1on G2off	0	(X^{5})	1,0,0,0	0	(X^{5})
3, G1on G2off	0	(X^{4})	0,1,0,0	0	(X^{4})
4, G1on G2off	1	(X^{3})	0,0,1,0	1	(X^{3})
5, G1off G2on			1,0,0,1	1	(X^{2})
6, G1off G2on			1,1,0,0	1	(X^{1})
7, G1off G2on			1,1,1,0	0	(X^{0})

5. Existence of Cyclic Codes

- For any n and k, is there an (n, k) cyclic code ?
- If $g(X)$ is a polynomial of degree n-k and a factor of X^n + 1, then $g(X)$ generates an (n, k) cyclic.
- As a matter of fact, any factor $g(X)$ of $X^n + 1$ with degree n - k generates an (n, k) cyclic code.
- For large n, $X^n + 1$ may have many factors of degree n - k. Some generate good codes and some generate bad codes.

Example 4.4: The polynomial $X^7 + 1$ can be factored into the following product of irreducible polynomials:

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

(1) $g_1(X)$ = 1 + X + X^3 generates the (7, 4) cyclic code given by Table 4 .1.

(2) $g_2(X)$ = 1 + X^2 +X^3 generates the (7, 4) cyclic code .

(3) $g_3(X)$= (1 + X) (1 + X + X^3) = 1 + X^2 + X^3 + X^4 generates the (7, 3) cyclic code.

6. Irreducible Polynomial

- A binary polynomial of degree *m* is said to be irreducible if it is not divisible by any binary polynomial of degree less than *m* and greater then zero.
- $1 + X + X^2$, $1 + X + X^3$, $1 + X + X^4$, $1 + X^2 + X^5$ are irreducible polynomials.
- For any positive integer $m \geq 1$, there exists at least one irreducible polynomial of degree *m*.
- An irreducible polynomial $p(X)$ of degree *m* is said to be **primitive** if the smallest positive *n* for which $p(X)$ divides $X^n + 1$ is $n = 2^m - 1$.
- For any positive integer *m*, there exists a primitive polynomial of degree *m*.

7. Cyclic Hamming Codes

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

  $n = 2^m - 1$, $k = 2^m - m - 1$, $m = n - k$, $d_{\min} = 3$, $t = 1$
- The (7, 4) cyclic code in Table 4.1 is a cyclic Hamming code generated by the primitive polynomial $p(X) = 1 + X + X^3$
- The primitive polynomial $p(X) = 1 + X + X^4$ generates a (15, 11) cyclic Hamming code.

Distance-4 Cyclic Hamming Codes

- It is generated by $g(X) = (X+1)p(X)$.
- It is subcode of the distance-3 cyclic code generated by $p(X)$
- 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.

8. Syndrome Computation and Error Detection

- Syndrome computation for cyclic codes is easy.
- Let $v(X)$ and $r(X)$ be the transmitted code polynomial and received polynomial respectively.
- Dividing $r(X)$ by the generator polynomial $g(X)$, we have

$$r(X) = a(X)g(X) + s(X)$$

where

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

is the remainder.

- Since $v(X)$ is a code polynomial, then $v(X) = c(X) \cdot g(X)$
- Consequently,

$$e(X) = [a(X) + c(X)]g(X) + s(X)$$

- We see that the syndrome is actually the remainder resulting from dividing the error polynomial $e(X)$.

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

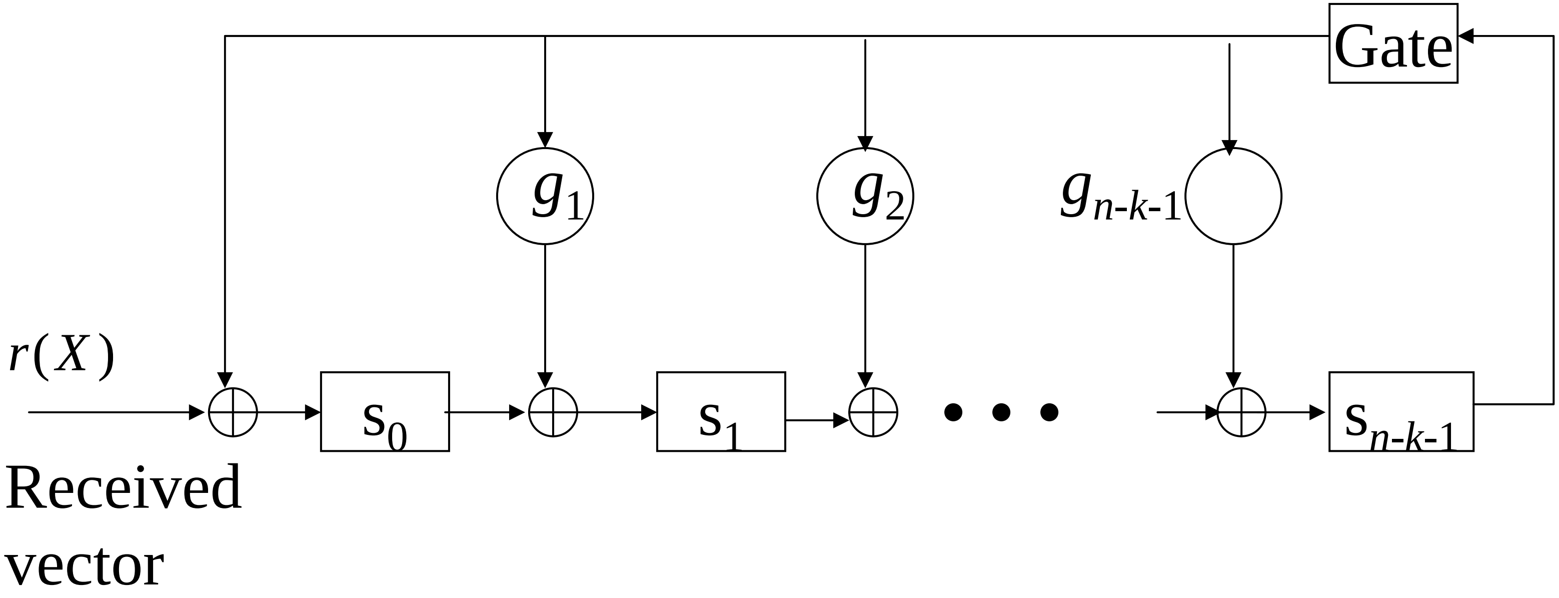


Figure 4.5 An (*n*-*k*) stage syndrome circuit

Example 4.5: A syndrome circuit for the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$. Suppose that the received vector $\bar{r} = (0010110)$. The syndrome of $\bar{r}$ is $\bar{s} = (101)$

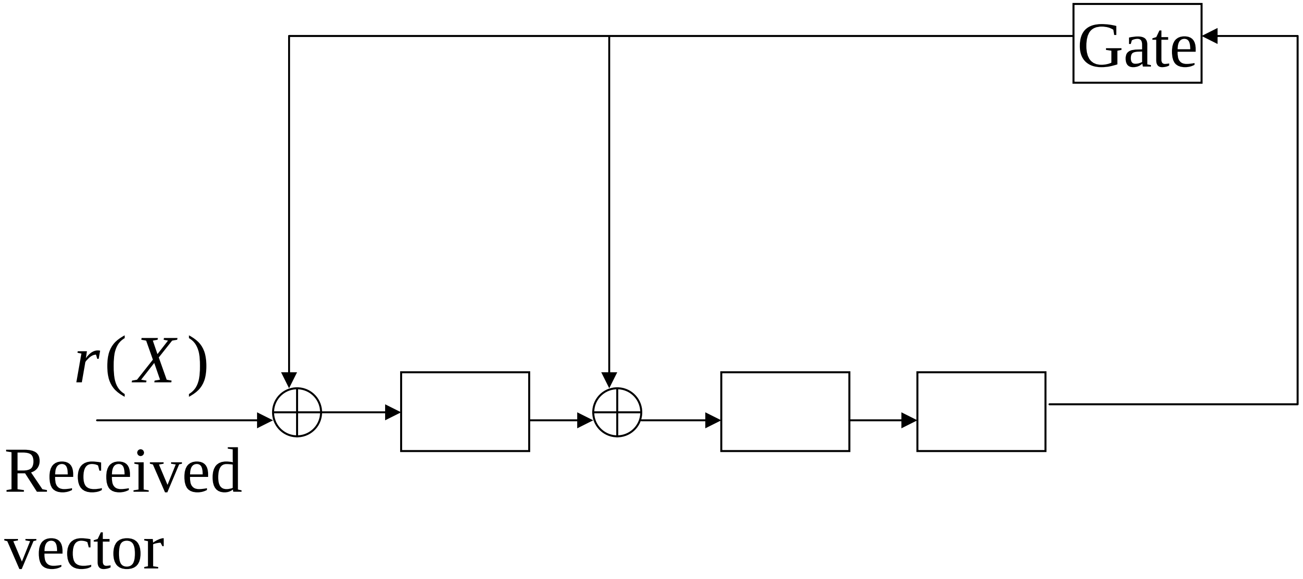


Figure 4.5.1 An syndrome circuit for the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$

Table 4.3 As $\bar{r} = (0010110)$, the contents of the syndrome register

timing	input	Register contents
0	0	0,0,0 (initial)
1	1	0,0,0
2	1	1,0,0
3	0	1,1,0
4	1	0,1,1
5	0	0,1,1
6	0	1,1,1
7	-	1,0,1(syndrome)

HW #5

1.Consider the (15, 11) cyclic Hamming code generated by

$$g(X) = 1 + X + X^4$$

(a) Determine the parity polynomial $h(X)$.

(b) Given $c(X) = 1 + X^2$, then what is the output code sequence ?

2. Devise an encoder and a syndrome circuit for Problem 1.

9. Burst Error Detection with Cyclic Codes

- In certain channels, errors occur in clusters.
- A cluster of errors is called **an error burst**.
- An error burst is said to have length *l* if all the errors are confined to *l* consecutive positions.
- For example, $\bar{e} = (00001101000)$ is an error burst of length 4.
- Using polynomial representation, an error burst of length *l* has the following form:

$$e(X) = X^{i}(1 + e_{i+1}X + \ldots + e_{i+l-2}X^{l-2} + X^{l-1})$$

where X^{i} and X^{i+l-1} are the beginning and ending of the burst.

- For $l \leq n - k$, we see that **no error burst is divisible** by the generator polynomial $g(X)$. Hence its syndrome is nonzero.
- An (*n*, *k*) cyclic code is capable of detecting any burst of length *n* – *k* or less (including the end-around burst).
- In fact, a large percentage of error bursts of length *n* – *k* + +1 or longer can be detected.
- For burst length *l* = *n* – *k* + 1, the fraction of **undetectable** error bursts is

$$2^{-(n-k-1)}$$

- For burst length *l* > *n* – *k* + 1, the fraction of **undetectable** error bursts is

$$2^{-(n-k)}$$

- Cyclic codes are very effective in detecting error bursts.

10. Decoding of Cyclic Codes

- Consists of the same 3 steps as for decoding linear codes syndrome computation, association of the syndrome to a correctable error pattern, and error correction.
- The cyclic structure allows us to decode a received vector

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

in serial manner, one bit at a time from the high order to the end.

- Each received bit is decoded with the same circuitry.
- A general cyclic code decoded is shown in Figure 4.6.

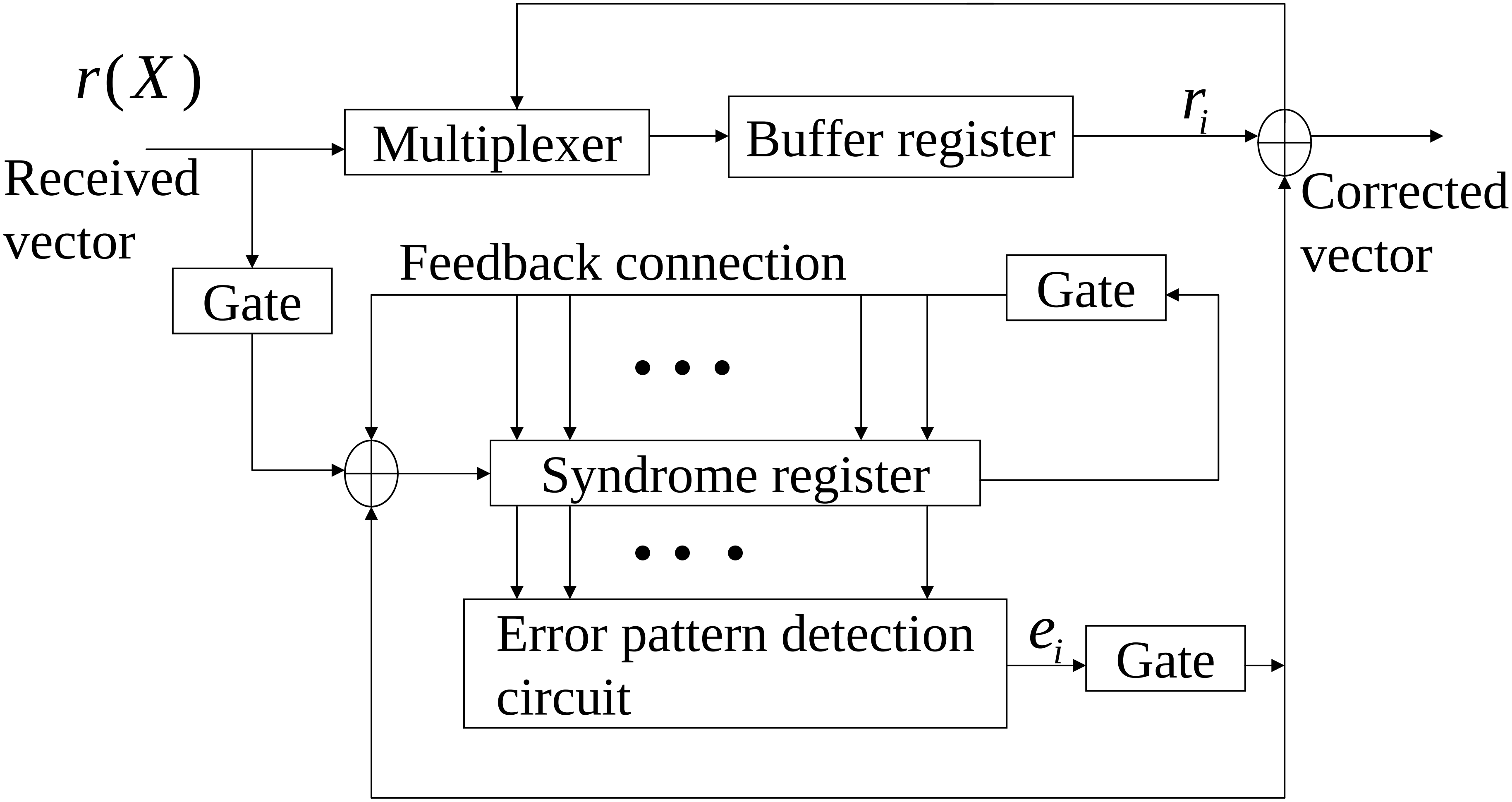


Figure 4.6 General cyclic code decoder

Decoding Process

- Shift the received polynomial $r(X)$ into a buffer and the syndrome registers simultaneously.
- Check whether the syndromes $s(X)$ corresponds to a correctable error pattern

$$e(X) = e_0 + e_1 X + ... + e_{n-1} X^{n-1}$$

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

- Correct $r_{n\text{-}1}$ if $e_{n\text{-}1} = 1$.
- Cyclically shift the buffer and syndrome registers once simultaneously. Now the buffer register contains

$$r^{(1)}(X) = (r_{n-1} + e_{n-1}) + r_0 X + ... + r_{n-2} X^{n-1}$$

and the syndrome register contains the syndrome

$$s^{(1)}(X) \;\; of \;\; r^{(1)}(X)$$

- Check whether $s^{(1)}(X)$ corresponds to a correctable error pattern $e^{(1)}(X)$ with an error at the highest-order position X^{n-1}.
- Correct r_{n-2} if it is erroneous.
- Repeat the same process until *n* shifts.
- If the error pattern is correctable, the buffer register contains the transmitted codeword and the syndrome register contains zeros.
- If the syndrome register does not contain all zero at the end of decoding process, an uncorrectable error pattern has been detected.

11. Decoding of Hamming codes

- Consider the (7, 4) Hamming code generated by

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

- 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

$$e(X) = X^6$$

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

$$X^6 = \underbrace{(X^3 + X + 1)}_{Quotient} \underbrace{(X^3 + X + 1)}_{g(X)} + \underbrace{(X^2 + 1)}_{\mathrm{Re}\,mainder}$$

- If the error pattern is error-free, the buffer register contains the transmitted codeword and the syndrome register contains an all-zero vector.
- If the syndrome register contains a non-zero vector at the decoding process, an uncorrectable/correctable error pattern has been detected.
- Hence the syndrome of $e(X) = X^2$ is

$$s(X) = X^2 \quad \text{or} \quad \bar{s} = (001)$$

- In the decoding process, we check the syndrome in the syndrome register. If the syndrome is (001), the second order bit in the buffer register is erroneous and must be corrected.
- The entire decoding circuit is shown in Figure 4.7.

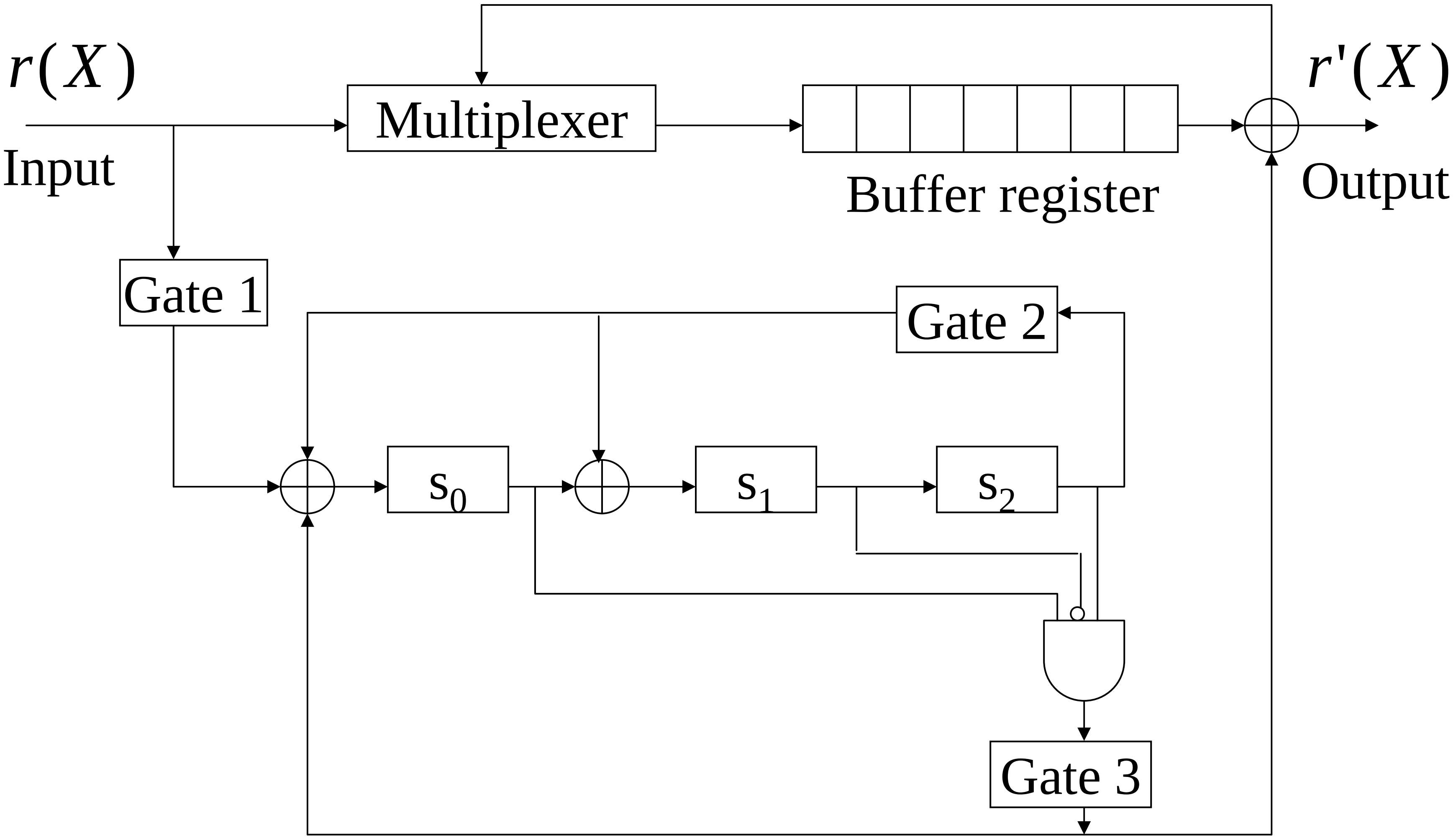


Figure 4.7 Decoding circuit for the (7, 4) cyclic code generated by $g(X) = 1 + X + X^3$

Table 4.4 The error pattern shifted into the syndrome register

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

Example 4.6: The complete decoding circuit is shown in Figure 4.7. Figures 4.8 - 4.9 illustrate the decoding process. Suppose that the transmitted vector is

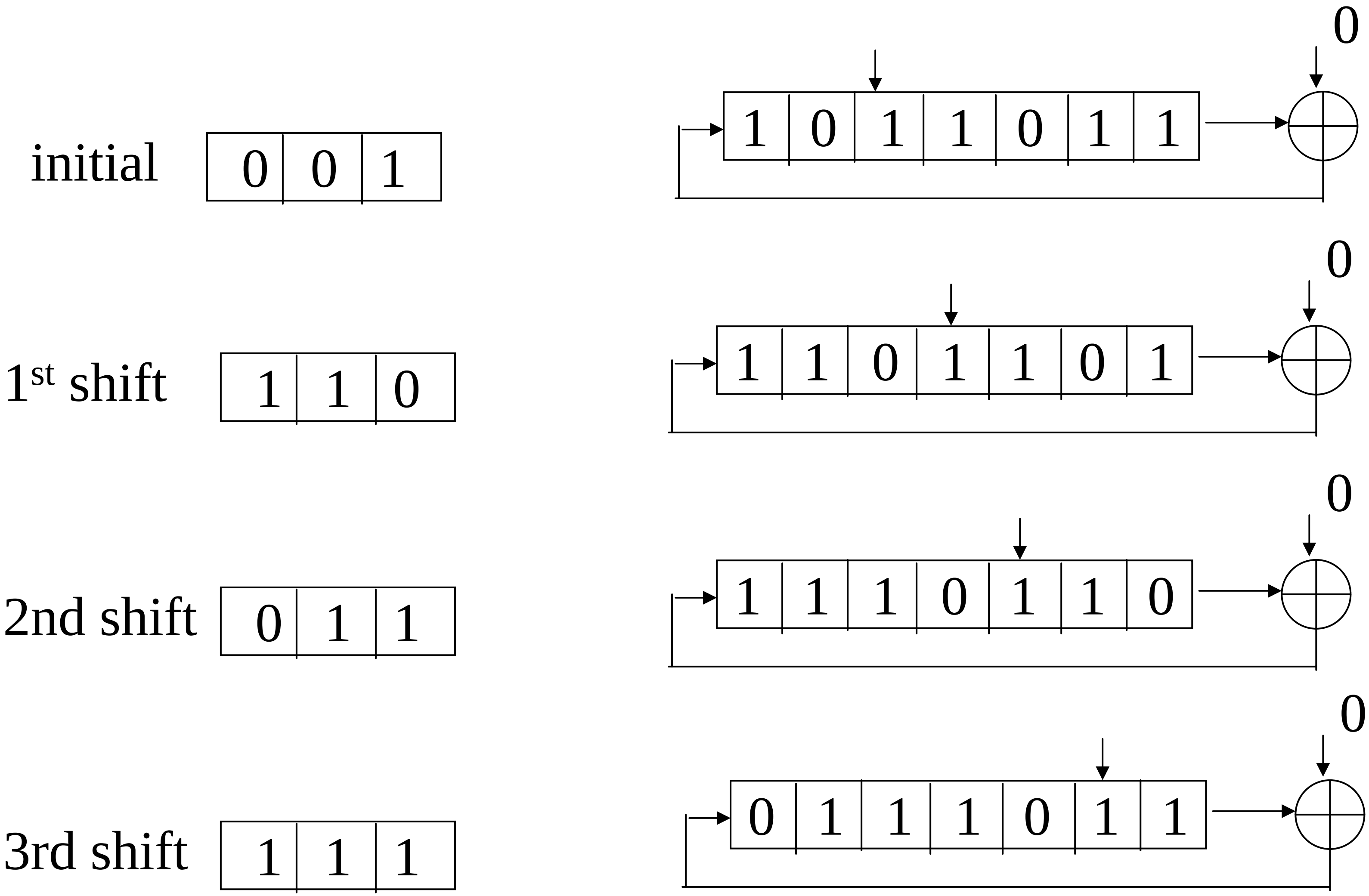
$$\bar{v} = (1001011)$$

$$v(X) = 1 + X^3 + X^5 + X^6$$

The received sequence is

$$\bar{r} = (1011011)$$

A single error occurs at location X^2, when the entire received polynomial has been shifted into the syndrome registers(Gates 1 and 2 are on, as Gate 3 is off during the initial process), the syndrome content is (001). Then, the Gate 1 is off and Gates 2 and 3 are on in the following decoding process. The decoding process is shown in the following figure.



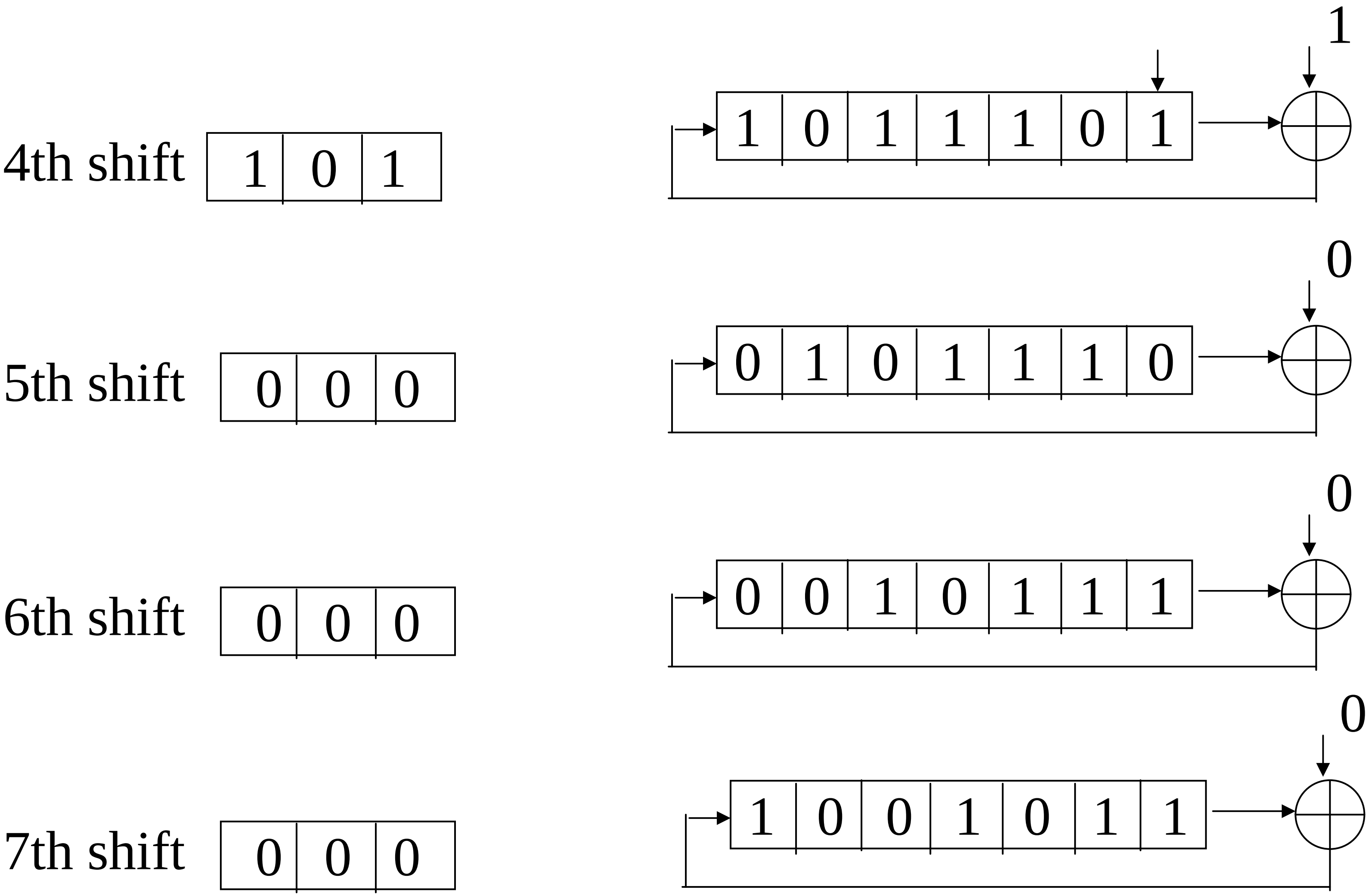
initial 0 0 1 | 1 0 1 1 0 1 1 | 0

1st shift 1 1 0 | 1 1 0 1 1 0 1 | 0

2nd shift 0 1 1 | 1 1 1 0 1 1 0 | 0

3rd shift 1 1 1 | 0 1 1 1 0 1 1 | 0

Figure 4.8 Error correction process



4th shift 1 0 1 — 1 0 1 1 1 0 1 — 1

5th shift 0 0 0 — 0 1 0 1 1 1 0 — 0

6th shift 0 0 0 — 0 0 1 0 1 1 1 — 0

7th shift 0 0 0 — 1 0 0 1 0 1 1 — 0

Figure 4.9 Error correction process (continuous)

12. 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 $g(X)$.
- 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 degree *n* - *l* -1 which are multiples of $g(X)$.
- Let $c(X) = c_0 + c_1 X + ... + c_{k-l-1} X^{k-l-1}$ be the message to be encoded.

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

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

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

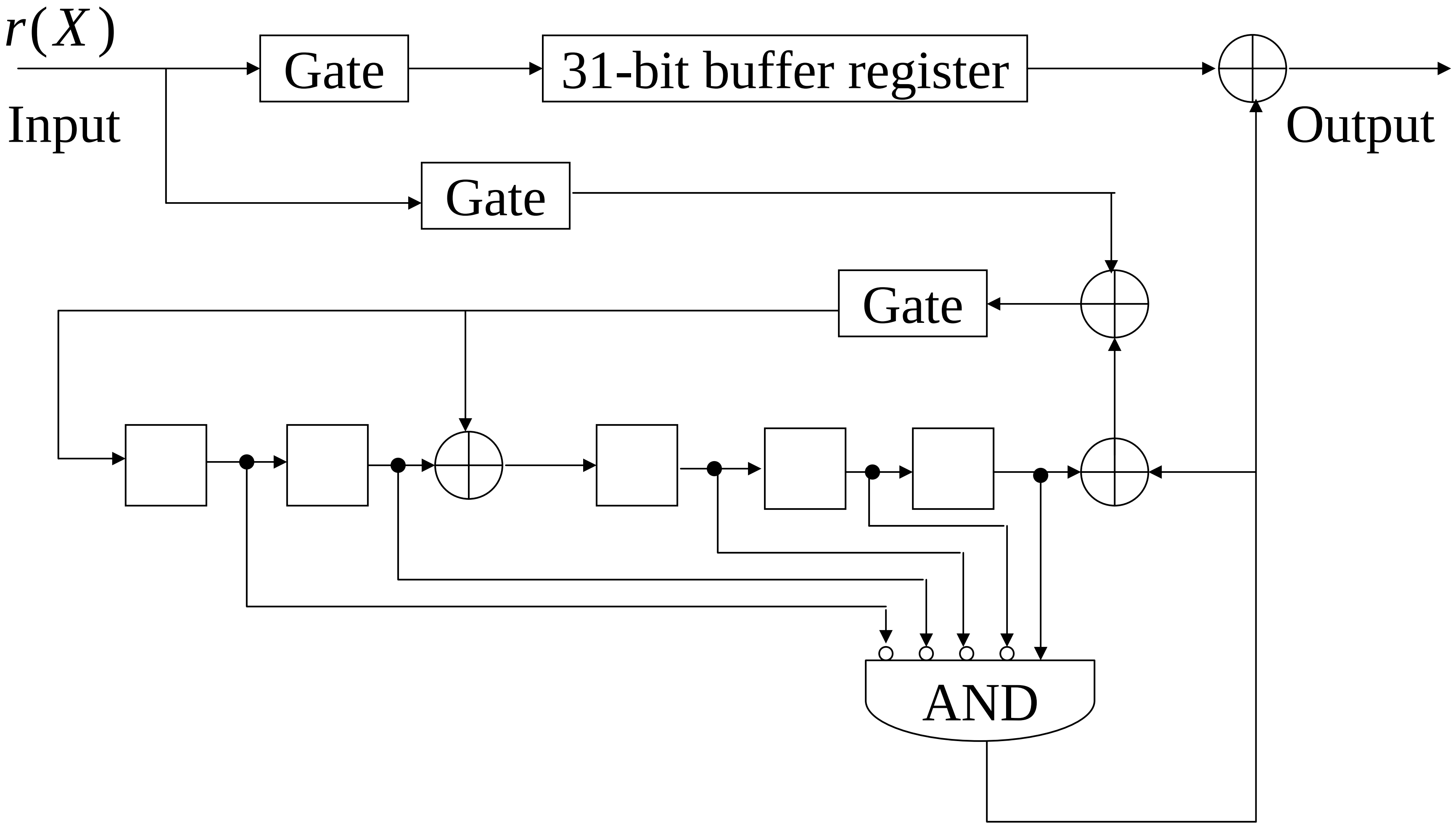


Figure 4.10 Decoding circuit for the (31, 26) cyclic Hamming code generated by $g(X) = 1 + X^2 + X^5$

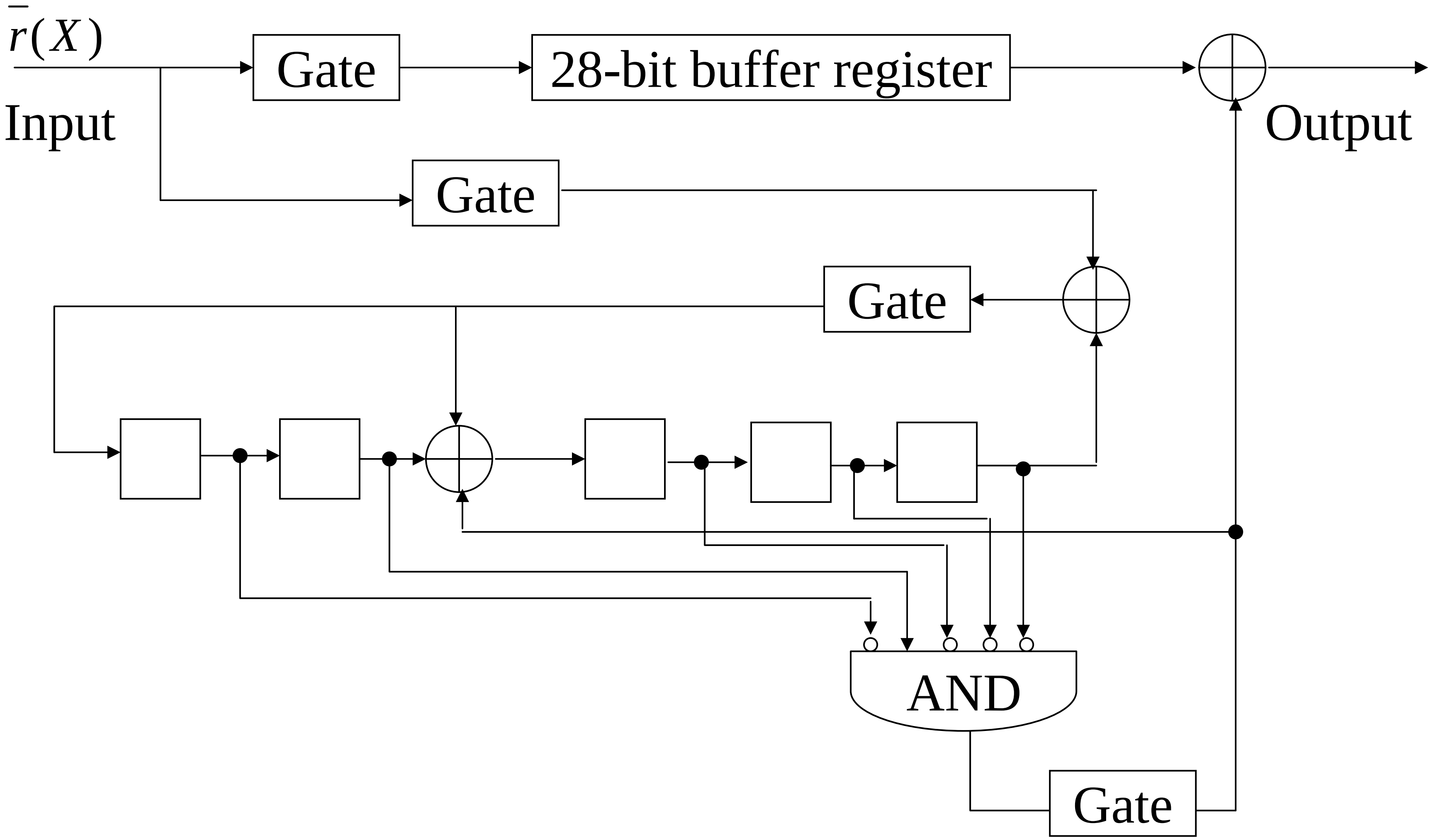


Figure 4.11 Decoding circuit for the (28, 23) shortened cyclic Hamming code generated by $g(X) = 1 + X^2 + X^5$

13. Important Cyclic Codes

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

$$g_1(X) = 1 + X + X^2 + X^4 + X^5 + X^6 + X^{10} + X^{11}$$

or

$$g_2(X) = 1 + X + X^5 + X^6 + X^7 + X^9 + X^{11}$$

- 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.

14. 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:

$$P_{u\,d}(E) \leq 2^{-(n-k)}$$

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

(1) Hamming codes.

(2) Golay (23, 12) code.

(3) Distance 5 – 8 primitive BCH codes.

(4) Reed – Solomon codes in nonbinary case and

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

where q is the size of code alphabet.

15. 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

$$g_1(X) = (1 + X)(X^{15} + X^{14} + X^{13} + X^{12} + X^4 + X^3 + X^2 + X + 1)$$

$$= X^{16} + X^{12} + X^5 + 1$$

or

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

- The natural length of the code is $n = 2^{15} - 1 = 32{,}767$. It is usually shortened to a fewer hundred to a few thousand bits long.

16. The IEEE Standard 802.3 Code

- A Hamming code with 32 parity bits generated by

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

- Used in the Ethernet .

HW #6

1. In Example 4.6, with the received sequence $\bar{r} = (1001010)$ please illustrate the decoding steps.

2. Devise a decoding circuit for (7, 3) Hamming code generated by $g(X) = (X+1)(X^3 + X + 1)$. The decoding circuits corrects all the single error patterns and all the double-adjacent-error patterns.

17. Error Correction Performance and MATLAB Example

- In following figure, the comparison of error correction performance of a shorten cyclic code is shown.
- The shortened cyclic code with length 26, dimension 16, and minimum Hamming distance 5 is illustrated in MATLAB for error correction. Each Chinese character is constituted by 2 bytes (8-bit). This shortened cyclic code is shorten from the (31, 21, 5) cyclic code.
- For details, please download the file “Ctext-crc.zip” in “老胡小舖”

烽火連三月
家書抵萬金
白頭騷更短
渾欲不勝簪

國破山河在
城春草木深
感時花濺淚
恨別鳥驚心

烽佐連三月
家書岳萬金
白頭騷更短
渾欲不븐簪

國?氏沿在城春
草木深感時花
濺淚-恨別鳥驚
心

烽火連三月
家書抵萬金
白頭騷更短
渾欲不勝簪

國破山河在
城春草木深
感時花濺淚
恨別鳥驚心

Figure 4.12: the original Chinese poem (left), degraded by AWGN (middle), recovered with (26, 16, 3) cyclic encoding/decoding (right)

HW #6-1

1. In the previous MATLAB program, we encode each Chinese character with a cyclic encoding.
2. Now, for some reason, we would like to encode this file “杜甫詩.txt” with cyclic encoding by the line-by-line way. And this code is with 2-error correction. Please modify this program and adjust the SNR such that there are no errors in the decoded file.
3. What kind of the shortened cyclic code is used ?