Lecture 5

Binary BCH Codes

BCH (Bose-Chauduri-Hoquenghem) codes form a large class of cyclic codes for correcting multiple random errors. This class of codes was first discovered by Hocquenhem in 1959 [1] and independently by Bose and Chaudhuri in 1960 [2]. The first algorithm for decoding binary BCH codes was devised by Peterson in 1960 [3]. Since then, Petersons decoding algorithm was improved and generalized by others. The most efficient algorithm for decoding BCH codes is the Berlekamp-Massey algorithm [4, 5]. A good coverage of BCH codes can be found in [6]. In this lecture, we focus in binary BCH codes. Non-binary BCH codes will be discussed in next lecture.

5.1 Primitive BCH Codes

- For any positive integer $m \geq 3$ and $t < 2^{m-1}$, there exists a binary cyclic BCH code with the following parameters:

  Length: $n = 2^m - 1$

  Number of parity-check bits: $n - k \leq mt$

  Minimum distance: $d_{min} \geq 2t + 1$.

- This code is capable of correcting t or fewer random errors over a span of $2^m - 1$ bit positions and hence called a t-error-correcting BCH code, denoted $C_{bch,t}$.
- The parameters t is called the **designed error-correcting capability** and the parameters $2t + 1$ is called the **designed minimum distance**.
- For example, for $m = 6$ and $t = 3$, there exists a triple-error-correcting BCH code with $n = 2^6 - 1 = 63$, $n - k = 6 \times 3 = 18$, $d_{min} = 2 \times 3 + 1 = 7$.

5.2 Generation of Binary Primitive BCH Codes

- Let α be a primitive element of GF(2^m), an extension field of GF(2).
- The order of α is $2^m - 1$ and its minimal polynomial is a primitive polynomial of degree m over GF(2).
- The generator polynomial $\mathbf{g}(X)$ of a binary primitive BCH code of length $n = 2^m - 1$ is the polynomial over GF(2) of the smallest degree that has the following **consecutive powers** of α,

$$\alpha,\ \alpha^2,\ \ldots,\ \alpha^{2t} \tag{5.1}$$

  as roots.
- It follows from **Theorem 2.2** in Lecture 2 (or **Theorem 2.11** in Lin/Costello) that $\mathbf{g}(X)$ also has all the conjugates of $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots.

- For $1 \leq i \leq 2t$, let ϕ_i be the minimal polynomial of α^i.
- Then, it follows from the definition of $\mathbf{g}(X)$ that

$$\mathbf{g}(X) = LCM\{\phi_1(X), \phi_2(X), \ldots, \phi_{2t}(X)\}. \tag{5.2}$$

- Suppose i is even. Then i can be expressed a product of an odd integer i' and a power of 2, say 2^l, as follows:

$$i = i'2^l, \tag{5.3}$$

- Consider

$$\alpha^i = \alpha^{i'2^l} = (\alpha^{i'})^{2^l}. \tag{5.4}$$

  This says that every even power α^i of α in the sequence of (5.1) is a conjugate of some preceding odd power $\alpha^{i'}$ of α in the sequence of (5.1).
- Therefore, α^i and $\alpha^{i'}$ have the same minimal polynomial, i.e.,

$$\phi_i(X) = \phi_{i'}(X).$$

- Consequently, we can remove all the minimal polynomials with even subscripts from the expression (5.2). This result in the following expression of $\mathbf{g}(X)$:

$$\mathbf{g}(X) = LCM\{\phi_1(X), \phi_3(X), \ldots, \phi_{2t-1}(X)\}. \tag{5.5}$$

- Since every minimal polynomial $\phi_i(X)$ in (5.5) has degree m or less and there are at most t different minimal polynomial in (5.5), the degree of $\mathbf{g}(X)$ is at most mt.

- Since every minimal polynomial $\phi_i(X)$ in (5.5) divides $X^{2^m-1}+1$, $\mathbf{g}(X)$ divides $X^{2^m-1}+1$.

- Since the minimal polynomial $\phi_1(X)$ of α (a primitive element) is a primitive polynomial of degree m, 2^m-1 is the smallest positive integer such that $\phi_1(X)$ divides $X^{2^m-1}+1$.

- Therefore, 2^m-1 is the smallest positive integer for which $\mathbf{g}(X)$ divides $X^{2^m-1}+1$.

- The binary BCH code generated by $\mathbf{g}(X)$ is a cyclic code $C_{bch,t}$ of length $2^m - 1$ with no more than mt parity-check bits, i.e., its dimension k is at least $2^m - mt - 1$. It will be proved that the minimum distance of $C_{bch,t}$ is at least $2t + 1$.
- The field GF(2^m) is called the code construction field.
- For $t = 1$,

$$\mathbf{g}(X) = \phi_1(X). \tag{5.6}$$

Since $\phi_1(X)$ is a primitive polynomial of degree m, $\mathbf{g}(X)$ generate a single-error-correcting BCH code $C_{bch,1}$ with

$$n = 2^m - 1, \qquad n - k = m, \qquad d_{min} = 3.$$

This single-error-correcting BCH code is simply a **Hamming code** in cyclic form.

- **Example 5.1**: Let $m = 4$ and $t = 3$. Let α be a primitive element of GF(2^4) that is constructed based on the primitive polynomial $p(X) = X^4 + X + 1$ (see Table 2.6 of Lecture-2 or Table 2.8 of the text book). Suppose we want to construct a triple-error-correcting binary BCH code of length $n = 2^4 - 1$. Then the generator polynomial $\mathbf{g}(X)$ of this BCH code is the smallest degree polynomial $\mathbf{g}(X)$ over GF(2) that has

$$\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5, \alpha^6,$$

  and their conjugates as roots.

- Using Table 2.6, we find that the minimal polynomials of α, α^3 and α^5 are:

$$\begin{aligned} \phi_1(X) &= 1 + X + X^4, \\ \phi_3(X) &= 1 + X + X^2 + X^3 + X^4, \\ \phi_5(X) &= 1 + X + X^2. \end{aligned}$$

- It follows from (5.5) that

$$\begin{aligned}\mathbf{g}(X) &= LCM\{\phi_1(X), \phi_3(X), \phi_5(X)\} \\ &= \phi_1(X)\phi_3(X)\phi_5(X) \\ &= 1 + X + X^2 + X^4 + X^5 + X^8 + X^{10}.\end{aligned}$$

- The cyclic code generated by $\mathbf{g}(X)$ is a binary $(15, 5)$ BCH code.

Table 2.6: GF(2^4) generated by the primitive polynomial $p(X) = 1 + X + X^4$.

Power representation	Polynomial representation							Vector representation
0	0							(0 0 0 0)
1	1							(1 0 0 0)
α			α					(0 1 0 0)
α^2					α^2			(0 0 1 0)
α^3							α^3	(0 0 0 1)
α^4	1	$+$	α					(1 1 0 0)
α^5			α	$+$	α^2			(0 1 1 0)
α^6					α^2	$+$	α^3	(0 0 1 1)
α^7	1	$+$	α			$+$	α^3	(1 1 0 1)
α^8	1			$+$	α^2			(1 0 1 0)
α^9			α			$+$	α^3	(0 1 0 1)
α^{10}	1	$+$	α	$+$	α^2			(1 1 1 0)
α^{11}			α	$+$	α^2	$+$	α^3	(0 1 1 1)
α^{12}	1	$+$	α	$+$	α^2	$+$	α^3	(1 1 1 1)
α^{13}	1			$+$	α^2	$+$	α^3	(1 0 1 1)
α^{14}	1					$+$	α^3	(1 0 0 1)

5.3 Structural Properties

- Consider a binary t-error-correcting primitive BCH code $C_{bch,t}$ of length $n = 2^m - 1$ with generator polynomial $\mathbf{g}(X)$ that has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots, i.e.,

$$\mathbf{g}(\alpha^i) = 0,$$

for $1 \leq i \leq 2t$.

- Since a code polynomial

$$\mathbf{v}(X) = v_0 + v_1 X + v_2 X^2 + \ldots + v_{n-1} X^{n-1} \tag{5.7}$$

in a cyclic code is a multiple of its generator polynomial $\mathbf{g}(X)$, $\mathbf{v}(X)$ also has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots, i.e.,

$$\mathbf{v}(\alpha^i) = 0, \tag{5.8}$$

for $0 \leq i \leq 2t$.

- Conversely, if a polynomial $\mathbf{v}(X) = v_0 + v_1 X + \cdots + v_{n-1} X^{n-1}$ over GF(2) with degree $n-1$ or less has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots, then $\mathbf{v}(X)$ must be divisible by $\mathbf{g}(X)$ and hence a code polynomial in $C_{bch,t}$.

- Summarizing the above results, we have Theorem 5.1.

- **Theorem 5.1**: Let α be a primitive element of GF(2^m). A polynomial $\mathbf{v}(X)$ of degree $2^m - 2$ over GF(2) is a code polynomial in the binary t-error-correcting primitive BCH code $C_{bch,t}$ of length $2^m - 1$ if and only if it has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots.

- It follows from (5.7) and (5.8) that for $1 \leq i \leq 2t$,

$$\begin{aligned} \mathbf{v}(\alpha^i) &= v_0 + v_1 \alpha^i + v_2 \alpha^{2i} + \cdots + v_{n-1} \alpha^{(n-1)i} \\ &= 0. \end{aligned} \tag{5.9}$$

- The equality of (5.9) can be expressed as the follow matrix product

$$\left[v_0,\ v_1,\ \ldots,\ v_{n-1}\right] \cdot \begin{bmatrix} 1 \\ \alpha^i \\ \alpha^{2i} \\ \vdots \\ \alpha^{(n-1)i} \end{bmatrix} = 0, \tag{5.10}$$

for $0 \leq i \leq 2t$.

- Eq. (5.10) simply says that the inner product of the code word $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ and $(1, \alpha^i, \alpha^{2i}, \ldots, \alpha^{(n-1)i})$ over GF(2^m) is equal to 0.

- Form the following $2t \times n$ matrix over GF(2^m):

$$\mathbf{H} = \begin{bmatrix} 1 & \alpha & \alpha^2 & \ldots & \alpha^{n-1} \\ 1 & \alpha^2 & (\alpha^2)^2 & \ldots & (\alpha^2)^{n-1} \\ 1 & \alpha^3 & (\alpha^3)^2 & \ldots & (\alpha^3)^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \alpha^{2t} & (\alpha^{2t})^2 & \ldots & (\alpha^{2t})^{n-1} \end{bmatrix} \tag{5.11}$$

- It follow from (5.10) that for every code word $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ in the binary t-error-correcting primitive BCH code $C_{bch,t}$, the following condition holds:

$$\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}.$$

where $\mathbf{0} = (0, 0, \ldots, 0)$ is a zero $2t$-tuple.

- On the other hand, if an n-tuple $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ over GF(2) satisfies the condition of (5.12), then it follows form (5.9) and (5.10) that its corresponding polynomial $\mathbf{v}(X) = v_0 + v_1 X + \ldots + v_{n-1} X^{n-1}$ has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots. Consequently, $\mathbf{v}(X)$ is divisible by the generator polynomial $\mathbf{g}(X)$ of the t-error-correcting BCH code $C_{bch,t}$ of length $n = 2^m - 1$ and hence a code polynomial.

- Hence, the t-error-correcting BCH code $C_{bch,t}$ generated by $\mathbf{g}(X)$ given by (5.5) is the null space of $\mathbf{H}$; and $\mathbf{H}$ is a parity-check matrix of the code.

- **Theorem 5.2**: Let $n = 2^m - 1$. An n-tuple $\mathbf{v}$ over GF(2) is code word in the t-error-correcting BCH code generated by $\mathbf{g}(X)$ given by (5.5) if and only if

$$\mathbf{v} \cdot \mathbf{H}^T = \mathbf{0}. \tag{5.12}$$

- $\mathbf{H}$ is a parity-check matrix of $C_{bch,t}$ over GF(2^m). If each entry of $\mathbf{H}$ is represented by an m-tuple over GF(2) in column form, we obtain a binary parity-check matrix $\mathbf{H}_b$ of $C_{bch,t}$.
- For decoding the BCH code $C_{bch,t}$, the parity-check matrix $\mathbf{H}$ over GF(2^m) given by (5.11) is used.

5.4 Minimum Distance

- Now we are ready to prove that the minimum distance of BCH code $C_{bch,t}$ generated by $\mathbf{g}(X)$ of (5.5) is at least $2t+1$. All we need to do is show that no nonzero code word of $C_{bch,t}$ has weight less than $2t+1$.

- Let $\mathbf{v} = (v_0, v_1, \ldots, v_{n-1})$ be a nonzero code word in $C_{bch,t}$ whose nonzero components are $v_{j_1}, v_{j_2}, \ldots, v_{j_\delta}$, i.e.

$$v_{j_1} = v_{j_2} = \cdots = v_{j_\delta} = 1.$$

  Hence, the Hamming weight of $\mathbf{v}$ is δ.

- **Hypothesis**: Suppose that $\delta \leq 2t$.

- If we can prove that the above hypothesis is invalid, then the weight of any nonzero code word in $C_{bch,t}$ is at least $2t+1$. This implies that the minimum distance of the code is at least $2t+1$.

- It follows from (5.11) and (5.12) that

$$
\begin{aligned}
\mathbf{0} &= \mathbf{v}\cdot\mathbf{H}^T \\
&= (v_0, v_1, \ldots, v_{n-1})\begin{bmatrix} 1 & 1 & \ldots & 1 \\ \alpha & \alpha^2 & \ldots & \alpha^{2t} \\ \alpha^2 & (\alpha^2)^2 & \ldots & (\alpha^{2t})^2 \\ \alpha^3 & (\alpha^2)^3 & \ldots & (\alpha^{2t})^3 \\ \vdots & \vdots & & \vdots \\ \alpha^{n-1} & (\alpha^2)^{n-1} & \ldots & (\alpha^{2t})^{n-1} \end{bmatrix} \\
&= (v_{j_1}, v_{j_2}, \ldots, v_{j_\delta})\begin{bmatrix} \alpha^{j_1} & (\alpha^2)^{j_1} & \ldots & (\alpha^{2t})^{j_1} \\ \alpha^{j_2} & (\alpha^2)^{j_2} & \ldots & (\alpha^{2t})^{j_2} \\ \vdots & \vdots & & \vdots \\ \alpha^{j_\delta} & (\alpha^2)^{j_\delta} & \ldots & (\alpha^{2t})^{j_\delta} \end{bmatrix} \\
&= (v_{j_1}, v_{j_2}, \ldots, v_{j_\delta})\begin{bmatrix} \alpha^{j_1} & (\alpha^{j_1})^2 & \ldots & (\alpha^{j_1})^{2t} \\ \alpha^{j_2} & (\alpha^{j_2})^2 & \ldots & (\alpha^{j_2})^{2t} \\ \vdots & \vdots & & \vdots \\ \alpha^{j_\delta} & (\alpha^{j_\delta})^2 & \ldots & (\alpha^{j_\delta})^{2t} \end{bmatrix} \qquad (5.13)
\end{aligned}
$$

- Eq.(5.13) implies that

$$\mathbf{0} = (v_{j_1}, v_{j_2}, \ldots, v_{j_\delta}) \begin{bmatrix} \alpha^{j_1} & (\alpha^{j_1})^2 & \ldots & (\alpha^{j_1})^\delta \\ \alpha^{j_2} & (\alpha^{j_2})^2 & \ldots & (\alpha^{j_2})^\delta \\ \vdots & \vdots & & \vdots \\ \alpha^{j_\delta} & (\alpha^{j_\delta})^2 & \ldots & (\alpha^{j_\delta})^\delta \end{bmatrix} \tag{5.14}$$

- Note that all the components of $(v_{j_1}, v_{j_2}, \ldots, v_{j_\delta})$ are nonzero, in fact equal to 1. For the equality of (5.14) to hold, the determinant of the $\delta \times \delta$ matrix must be equal zero, i.e,

$$\begin{vmatrix} \alpha^{j_1} & (\alpha^{j_1})^2 & \ldots & (\alpha^{j_1})^\delta \\ \alpha^{j_2} & (\alpha^{j_2})^2 & \ldots & (\alpha^{j_2})^\delta \\ \vdots & \vdots & & \vdots \\ \alpha^{j_\delta} & (\alpha^{j_\delta})^2 & \ldots & (\alpha^{j_\delta})^\delta \end{vmatrix} = 0$$

- The above determinant can be simplified as follows:

$$\alpha^{j_1+j_2+\cdots+j_\delta} \begin{vmatrix} 1 & \alpha^{j_1} & \ldots & (\alpha^{j_1})^{\delta-1} \\ 1 & \alpha^{j_2} & \ldots & (\alpha^{j_2})^{\delta-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha^{j_\delta} & \ldots & (\alpha^{j_\delta})^{\delta-1} \end{vmatrix} = 0 \tag{5.15}$$

- The equality of (5.15) implies that the determinant

$$\Delta = \begin{vmatrix} 1 & \alpha^{j_1} & \dots & (\alpha^{j_1})^{\delta-1} \\ 1 & \alpha^{j_2} & \dots & (\alpha^{j_2})^{\delta-1} \\ \vdots & \vdots & & \vdots \\ 1 & \alpha^{j_\delta} & \dots & (\alpha^{j_\delta})^{\delta-1} \end{vmatrix} = 0$$

- Note that Δ is a **Vandermonde deteminant** and

$$\Delta = \prod_{j_i \neq j_k} (\alpha^{j_i} - \alpha^{j_k}) \neq 0. \qquad (5.16)$$

- Hence, $\mathbf{vH}^T \neq \mathbf{0}$. This is a contradiction to **Theorem 5.2** which says that a code word $\mathbf{v}$ in $C_{bch,t}$ satisfies the condition, $\mathbf{vH}^T = \mathbf{0}$. Therefore, our hypothesis that $\delta \leq 2t$ is invalid.

- As a result, we must have

$$d \geq 2t + 1. \tag{5.17}$$

- Summarizing the above results, we conclude that the minimum distance of the BCH code generated by $\mathbf{g}(X)$ of (5.5) has minimum distance d_{min} at least $2t + 1$.

- The number $2t + 1$ is a lower bound on the minimum distance of a t-error-correcting BCH code. This bound is referred to as the BCH bound.

- This bound is actually based on the fact that the generator polynomial $\mathbf{g}(X)$ has $2t$ consecutive powers of a primitive element α in GF(2^m).

5.5 Syndrome Computation and Error Dectection

- Consider a t-error-correcting BCH code $C_{bch,t}$ of length $n = 2^m - 1$ with generator polynomial $\mathbf{g}(X)$ given by (5.5).
- Suppose a code polynomial

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

is transmitted.

- Let

$$\mathbf{r}(X) = r_0 + r_1 X + r_2 X^2 + \cdots + r_{n-1} X^{n-1}$$

be the received polynomial. Then

$$\mathbf{r}(X) = \mathbf{v}(X) + \mathbf{e}(X) \tag{5.18}$$

where

$$\mathbf{e}(X) = e_0 + e_1 X + e_2 X^2 + \cdots + e_{n-1} X^{n-1}$$

is the error pattern caused by the channel noise and/or interferences.

- Error detection is to check whether $\mathbf{r}(X)$ is a code polynomial. To accomplish this, we simply check whether $\mathbf{r}(X)$ has $\alpha, \alpha^2, \ldots, \alpha^{2t}$ as roots.
- Therefore, error detection can be done by computing

$$\begin{aligned} S_i &= \mathbf{r}(\alpha^i) \\ &= r_0 + r_1\alpha^i + r_2\alpha^{2i} + \cdots + r_{n-1}\alpha^{(n-1)i}, \end{aligned} \tag{5.19}$$

for $1 \leq i \leq 2t$.

- The $2t$-tuple over GF(2^m)

$$\mathbf{S} = (S_1, S_2, \ldots, S_{2t}) \tag{5.20}$$

is called the **syndrome** of $\mathbf{r}(X)$ and $S_1, S_2, \ldots, S_{2t}$ are the syndrome components.

- If $\mathbf{S} \neq \mathbf{0} = (0, 0, ..., 0)$, $\mathbf{r}(X)$ is **not** a code polynomial and the **presence of errors** is being detected.
- If $\mathbf{S} = \mathbf{0} = (0, 0, ..., 0)$, then $\mathbf{r}(X)$ is a code polynomial. In this case, we assume that $\mathbf{r}(X)$ is **error-free** and deliver it to the user. In the event that $\mathbf{r}(X)$ contains an **undetectable error pattern**, a decoding error is committed.
- In Lecture 2, we showed that for a polynomial $f(X)$ over GF(2),

$$[f(X)]^2 = f(X^2).$$

- Since the received polynomial $\mathbf{r}(X)$ is a polynomial over GF(2), we have

$$[\mathbf{r}(X)]^2 = \mathbf{r}(X^2).$$

- Substituting X with α^i in $\mathbf{r}(X)$, we have

$$[\mathbf{r}(\alpha^i)]^2 = \mathbf{r}(\alpha^{2i}).$$

- From the above equality, we find that

$$S_{2i} = S_i^2. \tag{5.21}$$

- Equality (5.21) says that to compute the syndrome $\mathbf{S} = (S_1, S_2, \ldots, S_{2t})$, we only need to compute the syndrome components with odd subscripts, i.e., $S_1, S_3, \ldots, S_{2t-1}$.

- **Example 6.2**: Consider the triple-error-correcting $(15, 5)$ BCH code $C_{bch,3}$ given in Example 6.1. The generator polynomial $\mathbf{g}(X)$ has $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5$ and α^6 as roots.

- Suppose a code polynomial $\mathbf{v}(X)$ is transmitted and

$$\mathbf{r}(X) = X^3 + X^5 + X^{12}$$

  is received.

- The syndrome of $\mathbf{r}(X)$ is a 6-tuple $\mathbf{S} = (S_1, S_2, S_3, S_4, S_5, S_6)$. Using the field GF($2^4$) given by Table 2.6 of Lecture 2 (Table 2.8 of Lin/Costello), we compute the syndrome components with odd subscripts,

$$\begin{aligned}
S_1 &= \mathbf{r}(\alpha) = \alpha^3 + \alpha^5 + \alpha^{12} = 1, \\
S_3 &= \mathbf{r}(\alpha^3) = \alpha^9 + \alpha^{15} + \alpha^{36} = \alpha^9 + 1 + \alpha^6 = \alpha^{10}, \\
S_5 &= \mathbf{r}(\alpha^5) = \alpha^{15} + \alpha^{25} + \alpha^{60} = 1 + \alpha^{10} + 1 = \alpha^{10},
\end{aligned}$$

- Using (2.21), we find the syndrome components with even subscripts,

$$S_2 = S_1^2 = 1, S_4 = S_2^2 = 1, S_6 = S_3^2 = \alpha^{20} = \alpha^5.$$

- Hence, the syndrome is $\mathbf{S} = (1, 1, \alpha^{10}, 1, \alpha^{10}, \alpha^5) \neq \mathbf{0}$ and the presence of errors in $\mathbf{r}(X)$ is being detected.

5.6 Syndrome and Error Pattern

- Since $\mathbf{r}(X) = \mathbf{v}(X) + \mathbf{e}(X)$, then

$$\begin{aligned} S_i &= \mathbf{r}(\alpha^i) = \mathbf{v}(\alpha^i) + \mathbf{e}(\alpha^i) \\ &= \mathbf{e}(\alpha^i), \end{aligned} \tag{5.22}$$

  For $1 \leq i \leq 2t$. Equality (5.22) gives a relationship between the syndrome $\mathbf{S} = (S_1, S_2, \ldots, S_{2t})$ and the error pattern $\mathbf{e}(X)$.

- Suppose $\mathbf{e}(X)$ has ν errors at the locations $X^{j_1}, X^{j_2}, \ldots, X^{j_\nu}$. Then

$$\mathbf{e}(X) = X^{j_1} + X^{j_2} + \cdots + X^{j_\nu}, \tag{5.23}$$

  where $0 \leq j_1 < j_2 < \ldots < j_\nu < n$.

- From (5.22) and (5.23), we have the following $2t$ equations that relate the error locations to the computed syndrome components:

$$\begin{aligned} S_1 &= \mathbf{e}(\alpha) = \alpha^{j_1} + \alpha^{j_2} + \cdots + \alpha^{j_\nu} \\ S_2 &= \mathbf{e}(\alpha^2) = (\alpha^{j_1})^2 + (\alpha^{j_2})^2 + \cdots + (\alpha^{j_\nu})^2 \\ &\vdots \\ S_{2t} &= \mathbf{e}(\alpha^{2t}) = (\alpha^{j_1})^{2t} + (\alpha^{j_2})^{2t} + \cdots + (\alpha^{j_\nu})^{2t}. \end{aligned} \tag{5.24}$$

- If we can solve these $2t$ equations, we can determine $\alpha^{j_1}, \alpha^{j_2}, \ldots, \alpha^{j_\nu}$ whose exponents $j_1, j_2, \ldots, j_\nu$ give the locations of errors in the error pattern $\mathbf{e}(X)$.
- Since the elements $\alpha^{j_1}, \alpha^{j_2}, \ldots, \alpha^{j_\nu}$ give the location of errors, they are called **error-location numbers**.

- To simplify the notations of (5.24), we define

$$\beta_l = \alpha^{j_l}, \tag{5.25}$$

With $1 \leq l \leq \nu$.

- Then, the $2t$ equations can be simplified as follows:

$$\begin{aligned} S_1 &= \beta_1 + \beta_2 + \cdots + \beta_\nu \\ S_2 &= \beta_1^2 + \beta_2^2 + \cdots + \beta_\nu^2 \\ &\vdots \\ S_{2t} &= \beta_1^{2t} + \beta_2^{2t} + \cdots + \beta_\nu^{2t} \end{aligned} \tag{5.26}$$

- The equations of (5.26) are known as the **power-sum symmetric functions**. They are nonlinear equations.

5.7 Error-Location Polynomial

- Define

$$\begin{aligned}\sigma(X) &= (1+\beta_1 X)(1+\beta_2 X)\cdots(1+\beta_\nu X) \\ &= \sigma_0 + \sigma_1 X + \sigma_2 X^2 + \cdots + \sigma_\nu X^\nu \end{aligned} \quad (5.27)$$

  Where $\sigma_0 = 1$.

- From (5.27), we see that $\sigma(X)$ has $\beta_1^{-1}, \beta_2^{-1}, \ldots, \beta_\nu^{-1}$ (the reciprocals (or inverses) of the location numbers) as roots.

- This polynomial $\sigma(X)$ is called the **error-location polynomial**.

- If we can determine $\sigma(X)$ from the syndrome $\mathbf{S} = (S_1, S_2, \ldots, S_{2t})$, then the roots of $\sigma(X)$ gives the error-location numbers, and the error-pattern $\mathbf{e}(X)$ can be determined.

- From (5.27), we find the following ν equalities that relate the coefficients of the error-location polynomial $\sigma(X)$ and the ν error-location numbers:

$$\begin{aligned}
\sigma_1 &= \beta_1 + \beta_2 + \cdots + \beta_\nu \\
\sigma_2 &= \beta_1\beta_2 + \beta_1\beta_3 + \cdots + \beta_{\nu-1}\beta_\nu \\
\sigma_3 &= \beta_1\beta_2\beta_3 + \beta_1\beta_2\beta_4 + \cdots + \beta_{\nu-2}\beta_{\nu-1}\beta_\nu \\
&\vdots \\
\sigma_\nu &= \beta_1\beta_2\cdots\beta_\nu
\end{aligned} \tag{5.28}$$

and $\sigma_0 = 1$.

- The above equalities are called the **elementary-symmetric functions**.

- From (5.26) and (5.28), we can derive the following equations that relate the coefficients of the error-location polynomial $\sigma(X)$ and the computed syndrome components:

$$
\begin{aligned}
&S_1 + \sigma_1 = 0 \\
&S_2 + \sigma_1 S_1 + 2\sigma_2 = 0 \\
&S_3 + \sigma_1 S_2 + \sigma_2 S_1 + 3\sigma_3 = 0 \\
&\vdots \\
&S_\nu + \sigma_1 S_{\nu-1} + \sigma_2 S_{\nu-2} + \cdots + \sigma_{\nu-1} S_1 + \nu\sigma_\nu = 0 \\
&S_{\nu+1} + \sigma_1 S_\nu + \sigma_2 S_{\nu-1} + \cdots + \sigma_{\nu-1} S_2 + \sigma_\nu S_1 = 0 \\
&\vdots
\end{aligned}
\tag{5.29}
$$

- Note $1+1=0$. Then

$$i\sigma_i = \begin{cases} \sigma_i, & \text{for odd } i; \\ 0, & \text{for even } i. \end{cases}$$

- The identities of (5.29) are referred to as the **Newton's identities**.

- If we can determine the coefficients $\sigma_1, \sigma_2, \ldots, \sigma_\nu$ of the error-location polynomial from the Newton's identities, then we can determine the error-location numbers, $\beta_1, \beta_2, \ldots, \beta_\nu$, of the error pattern $\mathbf{e}(X)$ by finding the roots of $\sigma(X)$.

5.8 A Procedure for Decoding BCH Codes

- Based on the developments given in Sections 5.6 and 5.7, a procedure for decoding a binary t-error-correcting BCH code can be formulated into the following steps:

(1) Compute the syndrome $\mathbf{S} = (S_1, S_2, \ldots, S_{2t})$ from the received polynomial $\mathbf{r}(X)$.

(2) Determine the error-location polynomial $\sigma(X)$ form the Newton's identity.

(3) Find the roots, $\beta_1^{-1}, \beta_2^{-1}, \ldots, \beta_\nu^{-1}$, of $\sigma(X)$ in GF(2^m). Take the inverses of these roots to obtain the error-location numbers, $\beta_1 = \alpha^{j_1}$, $\beta_2 = \alpha^{j_2}, \ldots, \beta_\nu = \alpha^{j_\nu}$. Then the error pattern is $\mathbf{e}(X) = X^{j_1} + X^{j_2} + \cdots + X^{j_\nu}$.

(4) Perform the error correction by adding $\mathbf{e}(X)$ to $\mathbf{r}(X)$. This gives the decoded code word, $\mathbf{v}(X) = \mathbf{r}(X) + \mathbf{e}(X)$.

- Steps (1), (3) and (4) can be carried out easily, however, Step 2 involves in solving the Newton's identities.
- There are in general more than one error pattern for which the coefficients of its error-location polynomial satisfy the Newton's identities.
- To minimize the probability of a decoding error, we need to find the **most probable** error pattern for error correction.
- For BSC, finding the most probable error pattern is to determine the error-location polynomial of the **minimum degree** whose coefficients satisfy the Newton's identities.

5.9 Berlekamp-Masey Iterative Algorithm for Finding Error-Location Polynomial

- The error-location polynomial $\sigma(X)$ can be computed iteratively with $2t$ steps.
- At the μ-th step, we determine a minimum-degree polynomial

$$\sigma^{(\mu)}(X) = 1 + \sigma_1^{(\mu)} X + \sigma_2^{(\mu)} X^2 + \cdots + \sigma_{l_\mu}^{(\mu)} X_\mu^l, \qquad (5.30)$$

  whose coefficients satisfy the first μ Newton's identities.
- At the $(\mu + 1)$-th step, we find the next minimum-degree polynomial $\sigma^{(\mu+1)}(X)$ whose coefficients satisfy the first $\mu + 1$ Newton's identities based on $\sigma^{(\mu)}(X)$.
- First, we check whether the coefficients of $\sigma^{(\mu)}$ also satisfy the $(\mu + 1)$-th Newton's identity.
- If yes, $\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X)$ is the minimum-degree polynomial whose coefficients satisfy the first $\mu + 1$ Newton's identities.

- If not, a **correction term** is added to $\sigma^{(\mu)}(X)$ to form next solution $\sigma^{(\mu+1)}(X)$ whose coefficients satisfy the first $\mu + 1$ Newton's identities.

- To test whether the coefficients of $\sigma^{(\mu)}$ satisfy the $(\mu + 1)$-th Newton's identity, we compute

$$d_\mu = S_{\mu+1} + \sigma_1^{(\mu)} S_\mu + \sigma_2^{(\mu)} S_{\mu-1} + \cdots + \sigma_{l_u}^{(\mu)} S_{\mu+1-l_\mu}. \tag{5.31}$$

  This quantity is called the μ-th **discrepancy**. The sum of the right-hand side of (5.31) is actually the left-hand side of the $(\mu + 1)$-th Newton's identity.

- If $d_\mu = 0$, then the coefficients of $\sigma^{(\mu)}(X)$ satisfy the $(\mu + 1)$-th Newton's identity. In this case, we set

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X),$$

  i.e., the current solution $\sigma^{(\mu)}(X)$ is also the next solution $\sigma^{(\mu+1)}(X)$.

- If $d_\mu \neq 0$, then $\sigma^{(\mu)}(X)$ needs to be adjusted to obtain a new minimum-degree polynomial $\sigma^{(\mu+1)}(X)$ whose coefficients satisfy the first $\mu + 1$ Newton's identities.

- **Correction**: Go back to the steps prior the μ-th step and determine a step ρ at which the partial solution is $\sigma^{(\rho)}(X)$ such that $d_\rho \neq 0$ and $\rho - l_\rho$ has the largest value, where l_ρ is the degree of $\sigma^{(\rho)}(X)$. Then the solution at the $(\mu + 1)$-th step of the iteration process for finding the error-location polynomial $\sigma(X)$ is

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X) + d_\mu d_\rho^{-1} X^{(\mu-\rho)} \sigma^{(\rho)}(X), \tag{5.32}$$

where

$$d_\mu d_\rho^{-1} X^{(\mu-\rho)} \sigma^{(\rho)}(X) \tag{5.33}$$

is the correction term with degree $\mu - (\rho - l_\mu)$.

- Since ρ is chosen to maximize $\rho - l_\mu$, this choice of ρ is equivalent to minimize the degree $\mu - (\rho - l_\rho)$ of the correction term.

- repeating the above testing and correction process until we reach the $2t$-th step. Then

$$\sigma(X) = \sigma^{(2t)}(X). \tag{5.34}$$

- The above iteration method for finding error-location polynomial $\sigma(X)$ applies to both binary and non-binary BCH codes.

- From the first Newton's identity, we readily see that

$$\sigma^{(1)}(X) = 1 + S_1 X. \tag{5.35}$$

Execution of the Iteration Process

- To carry out the iteration process to find the error-location polynomial $\sigma(X)$, we set up a table as below:

Table 5.1: Berlekamp-Massey iterative procedure for fi nding the error-location polynomial of a BCH code.

Step μ	**Partial solution** $\sigma^{(\mu)}(X)$	**Discrepancy** d_μ	**Degree** l_μ	**Step/degree difference** $\mu - l_\mu$
-1	1	1	0	-1
0	1	S_1	0	0
1	1 + $S_1 X$			
2				
$\vdots$				
$2t$				

- Fill the table.

- Example 5.3: Consider the triple-error-correcting $(15, 5)$ BCH code given in Example 5.1 whose generator polynomial $\mathbf{g}(X)$ has $\alpha, \alpha^2, \alpha^3, \alpha^4, \alpha^5$, and α^6 as roots where α is a primitive element of GF(2^4) (see Table 2.6 of Lecture 2). Suppose the all-zero code word

$$\mathbf{v} = (0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0)$$

is transmitted and

$$\mathbf{r} = (0\ 0\ 0\ 1\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0)$$

is received.

- Then the received polynomial is $\mathbf{r}(X) = X^3 + X^5 + X^{12}$. From the computations given in Example 5.2, we find that the syndrome is

$$\mathbf{S} = (S_1, S_2, S_3, S_4, S_5, S_6),$$

With $S_1 = 1, S_2 = 1, S_3 = \alpha^{10}, S_4 = 1, S_5 = \alpha^{10}$ and $S_6 = \alpha^5$.

- Iterative process results in the following table:

Table 5.2: Steps for finding the error-location polynomial of the $(15, 5)$ BCH given in Example 5.3.

μ	$\sigma^{(\mu)}(X)$	d_μ	l_μ	$\mu - l_\mu$
-1	1	1	0	-1
0	1	1	0	0
1	$1 + X$	0	1	0 $(\rho = -1)$
2	$1 + X$	α^5	1	1
3	$1 + X + \alpha^5 X^2$	0	2	1 $(\rho = 0)$
4	$1 + X + \alpha^5 X^2$	α^{10}	2	2
5	$1 + X + \alpha^5 X^3$	0	3	2 $(\rho = 2)$
6	$1 + X + \alpha^5 X^3$	–	–	–

- The error-location polynomial is

$$\sigma(X) = 1 + X + \alpha^5 X^3.$$

- Substituting the variable X of $\sigma(X)$ with the elements, $\alpha^0, \alpha, \ldots, \alpha^{14}$ of GF(2^4) in turns, we find that

$$\sigma(\alpha^3) = \sigma(\alpha^{10}) = \sigma(\alpha^{12}) = 0.$$

  Hence, α^3, α^{10} and α^{12} are the roots of $\sigma(X)$.

- The inverses of these three roots of $\sigma(X)$ are: $\alpha^{-3} = \alpha^{12}, \alpha^{-10} = \alpha^5$ and $\alpha^{-12} = \alpha^3$, which give the error-location numbers. The power of these three locations numbers are 12, 5 and 3.

- Consequently, the error pattern is

$$\mathbf{e}(X) = X^3 + X^5 + X^{12}$$

  which has three errors within the error correction capability of the code, $t = 3$.

- Removing $\mathbf{e}(X)$ from the received polynomial $\mathbf{r}(X)$, we obtain the decoded code polynomial,

$$\mathbf{v}(X) = \mathbf{r}(X) + \mathbf{e}(X) = 0 \text{ (zero polynomial)},$$

which is identical to the transmitted code polynomial.

- Hence, decoding is correct.

2.10 Simplification of Decoding Binary BCH Codes

- For decoding binary BCH code, we can prove that if the first, third, . . . , $(2t-1)$-th Newton's identities hold, then the second, fourth, . . . , $2t$-th Newton's identities also hold.
- This implies that with the iterative algorithm for finding the error-location polynomial $\sigma(X)$, the solution $\sigma^{(2\mu-1)}(X)$ at the $(2\mu-1)$-th step of iteration is also the solution $\sigma^{(2\mu)}(X)$ at the 2μ-th step of iteration, i.e.,

$$\sigma^{(2\mu)}(X) = \sigma^{(2\mu-1)}(X). \tag{5.36}$$

  for $1 \leq \mu \leq t$.
- This fact is demonstrated in Table 5.2.
- This suggest that the $(2\mu-1)$ and 2μ steps of iteration can be combined into one step. As a result, the forgoing algorithm for finding the error-location polynomial $\sigma(X)$ can be reduced to t steps. This simplification only applies to decoding of binary BCH codes (not to decoding of non-binary codes).

- The simplified algorithm for finding the error-location polynomial in decoding of a binary BCH code can be carried out by filling a table with only t steps as shown in Table 5.3.

Table 5.3: A simplifi ed Berlekamp-Massey iterative procedure for fi nding the error-location polynomial of a binary BCH code.

Step	**Partial solution**	**Discrepancy**	**Degree**	**Step/degree difference**
μ	$\sigma^{(\mu)}(X)$	d_μ	l_μ	$2\mu - l_\mu$
$-\frac{1}{2}$	1	1	0	-1
0	1	S_1	0	0
1				
2				
$\vdots$				
t				

- Suppose we have filled out all the rows up to and including the μ-th row, we fill out the $(\mu+1)$-th row as follows:

(1) If $d_\mu = 0$, then we set

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X).$$

(2) If $d_\mu \neq 0$, we find a row prior to the μ-th row, say the ρ-th row, with partial solution $\sigma^{(\rho)}(X)$ such that $d_\rho \neq 0$ and $2\rho - l_\mu$ is the largest. Then

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X) + d_\mu d_\rho^{-1} X^{2(\mu-\rho)} \sigma^{(\rho)}(X). \tag{5.37}$$

(3) Compute the discrepancy

$$d_{\mu+1} = S_{2\mu+3} + \sigma_1^{(\mu+1)} S_{2\mu+2} + \sigma_2^{(\mu+1)} S_{2\mu+1} + \cdots + \sigma_{l_{\mu+1}}^{(\mu+1)} S_{2\mu+3-l_{\mu+1}}. \tag{5.38}$$

where $l_{\mu+1}$ is degree of $\sigma^{(\mu+1)}(X)$.

- Note that with the simplified algorithm, the computation required to find the error-location polynomial $\sigma(X)$ is **half** of the computation required by the general algorithm for decoding both binary and non-binary BCH codes.
- Example 5.4: Using the simplified algorithm for finding the error-location polynomial, Table 5.2 given in Example 5.3 is reduced to Table 5.4.

Table 5.4: Steps for finding the error-location polynomial of the binary $(15, 3)$ BCH code given in Example 5.1

μ	$\sigma^{(\mu)}(X)$	d_μ	l_μ	$2\mu - l_\mu$
$-\frac{1}{2}$	1	1	0	-1
0	1	$S_1 = 1$	0	0
1	$1 + X$	α^5	1	$1\ (\rho = -\frac{1}{2})$
2	$1 + X + \alpha^5 X^2$	α^{10}	2	$2\ (\rho = 0)$
3	$1 + X + \alpha^5 X^3$	–	–	–

2.11 Finding the Roots of the Error-Location Polynomial

- The roots of the error-location polynomial $\sigma(X)$ can be determined by substituting the variable X with the elements of GF(2^m), $\alpha^0, \alpha, \ldots, \alpha^{2^m-2}$, in turn.
- For $0 \leq i < 2^m - 1$, if $\sigma(\alpha^i) = 0$, then α^i is a root of $\sigma(X)$. In this case, α^{2^m-1-i} is an error-location number and there is an error at the location $2^m - 1 - i$ of $\mathbf{r}(X)$, i.e., $e_{2^m-1-i} = 1$.

Reference

[1] A. Hocquenghem, “Codes corecteurs d’erreurs,” Chiffres, 2: 147-156, 1959.

[2] R. C. Bose and D. K. Ray-Chaudhuri, “On a class of error correcting binary group codes,” Inform. Control, 3: 68-79, March 1960.

[3] W. W. Peterson, “Encoding and error-correction procedures for Bose-Chaudhuri code,” IRE Trans. Inform. Theory, IT-6: 459-470, Sept. 1960.

[4] E. R. Berlekamp, Algebraic Coding Theory, McGraw-Hill, New York, 1968.

[5] J. L. Massey, “Shift-register synthesis and BCH decoding,” IEEE Trans. Inform. Theory, IT-15, pp. 122-127, Jan. 1969.

[6] S. Lin and D. J. Costello, Jr., Error Control Coding, Pearson Prentice-Hall, second edition, Upper Saddle River, NJ, 2004.

- Solution at the μ-th step

$$\sigma^{(\mu)}(X) = 1 + \sigma_1^{(\mu)} X + \sigma_2^{(\mu)} X^2 + \cdots + \sigma_{l_\mu}^{(\mu)} X^{l_\mu}$$

$$d_\mu = S_{\mu+1} + \sigma_1^{(\mu)} S_\mu + \sigma_2^{(\mu)} S_{\mu-1} + \cdots + \sigma_{l_\mu}^{(\mu)} S_{\mu+1-l_\mu}$$

- If $d_\mu = 0$, $\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X)$.

- If $d_\mu \neq 0$, find $\rho < \mu$ such that $d_\rho \neq 0$ and $\rho - l_\rho$ is the largest where l_ρ is the degree of the solution $\sigma^{(\rho)}(X)$ at the ρ-th step. Then

$$\sigma^{(\mu+1)}(X) = \sigma^{(\mu)}(X) + d_\mu d_\rho^{-1} X^{(\mu-\rho)} \sigma^{(\rho)}(X)$$

(1) $\mu=0$, $\sigma^{(0)}(X)=1$ and $d_0=1\neq 0$.

Take $\rho=-1$. $\sigma^{(-1)}(X)=1$.

$$\sigma^{(1)}(X)=\sigma^{(0)}(X)+d_0 d_{-1}^{-1} X^{(0-(-1))}\sigma^{(-1)}(X)$$

$$=1+X.$$

$$d_1 = S_2+\sigma_1^{(1)}S_1 = 1+1=0.$$

(2) $\mu=1$, $\sigma^{(1)}(X)=1+X$ and $d_1=0$.

$$\sigma^{(2)}(X)=\sigma^{(1)}(X)=1+X.$$

$$d_2 = S_3+\sigma_1^{(2)}S_2 = \alpha^{10}+1=\alpha^5\neq 0.$$

(3) $\mu=2$, $\sigma^2(X)=1+X$, $d_2\neq 0$.

Take $\rho=0$. $\sigma^{(0)}(X)=1$.

$$\sigma^{(3)}(X)=\sigma^{(2)}(X)+d_2 d_0^{-1}X^{(2-0)}\sigma^{(0)}(X)$$

$$=1+X+\alpha^5X^2$$

$$d_3=S_4+\sigma_1^{(3)}S_3+\sigma_2^{(3)}S_2=1+\alpha^{10}+\alpha^5=0.$$

(4) $\mu=3$, $\sigma^{(3)}(X)=1+X+\alpha^5X^2$, $d_3=0$.

$$\sigma^{(4)}(X)=\sigma^{(3)}(X)=1+X+\alpha^5X^2.$$

$$d_4 = S_5+\sigma_1^{(3)}S_4+\sigma_2^{(3)}S_3$$

$$=\alpha^{10}+1+\alpha^5\cdot\alpha^{10}=\alpha^{10}.$$

(5) $\mu=4$, $\sigma^{(4)}(X)=1+X+\alpha^5X^2$, $d_\mu=\alpha^{10}$.

Take $\rho=2$. $\sigma^{(2)}(X)=1+X$.

$$\sigma^{(5)}(X)=\sigma^{(4)}(X)+d_4d_2^{-1}X^{(4-2)}\sigma^{(2)}(X)$$

$$=1+X+\alpha^5X^2+\alpha^{10}\alpha^{-5}X^2(1+X)$$

$$=1+X+\alpha^5X^2+\alpha^5(X^2+X^3)$$

$$=1+X+\alpha^5X^3.$$

$$d_5=S_6+\sigma_1^{(5)}S_5+\sigma_2^{(5)}S_4+\sigma_3^{(5)}S_3$$

$$=\alpha^5+\alpha^{10}+\alpha^5\alpha^{10}=1+\alpha^5+\alpha^{10}=0.$$

(6) $\sigma^{(6)}(X)=\sigma^{(5)}(X)=1+X+\alpha^5X^3$.