BCH Codes -II

Qi Zhang

Aarhus University School of Engieering

27/02, 2014

Calculate the syndrome vector

- Parity check matrix:

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

- Syndrome vector can be expressed by

$$\begin{aligned} \mathbf{S} &= (s_1, s_2, \dots, s_{2t}) = \mathbf{r} \circ \mathbf{H}^T \\ &= (r_0, r_1, \dots, r_{n-1}) \circ \begin{bmatrix} 1 & 1 & \dots & 1 \\ \alpha & \alpha^2 & \dots & \alpha^{2t} \\ \alpha^2 & (\alpha^2)^2 & \dots & (\alpha^{2t})^2 \\ \vdots & \vdots & \dots & \vdots \\ \alpha^{n-1} & (\alpha^2)^{n-1} & \dots & (\alpha^{2t})^{n-1} \end{bmatrix} \end{aligned}$$

therefore,

$$s_i = r_0 + r_1 \cdot \alpha^i + r_2 \cdot (\alpha^i)^2 + \dots + r_{n-1} \cdot (\alpha^i)^{n-1} = r(\alpha^i)$$

with $1 \le i \le 2t$.

Calculate the syndrome vector

- **Summary**:
  - To calculate the ith component of the syndrome vector, we can replace the variable X with the root α^i in the received polynomial $r(X)$.
  - Syndrome vector consists of elements of the GF(2^m).

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

Calculate the syndrome vector

- **Example 4.3**: The Binary BCH code $C_{BCH}(15, 7)$ can correct 2 or less errors. The generator polynomial has roots in GF(2^4) which is generated by primitive polynomial $p_i(X) = 1 + X + X^4$. If the received vector $\mathbf{r} = (100000001000000)$, calculate the syndrome vector.
- **Solution**: Since $\mathbf{r} = (100000001000000)$, $r(X) = 1 + X^8$, then substitute $\alpha^i, 1 \leq i \leq 2t = 4$, and look up Table B.4

$$\begin{aligned}
s_1 &= r(\alpha) = 1 + \alpha^8 = \alpha^2 \\
s_2 &= r(\alpha^2) = 1 + \alpha = \alpha^4 \\
s_3 &= r(\alpha^3) = 1 + \alpha^9 = 1 + \alpha + \alpha^3 = \alpha^7 \\
s_4 &= r(\alpha^4) = 1 + \alpha^2 = \alpha^8
\end{aligned}$$

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

Error location and error evaluation polynomials

- Let's assume that the error vector contains τ non-zero elements, representing an error pattern of τ errors placed at positions X^{j_1}, X^{j_2}, ..., X^{j_τ}, where $0 \leq j_1 < j_2 < \ldots < j_\tau \leq n-1$.

$$e(X) = e_{j_1} X^{j_1} + e_{j_2} X^{j_2} + \ldots + e_{j_\tau} X^{j_\tau}$$

- The error-location number is defined as $\beta_l = \alpha^{j_l}$, where $l = 1, 2, 3, \ldots, \tau$.
- The syndrome vector element has this relation: $s_i = r(\alpha^i) = c(\alpha^i) + e(\alpha^i) = e(\alpha^i)$.
- Thus, a system of $2t$ equations can be formed as follows:

$$\begin{aligned} s_1 &= e(\alpha) = e_{j_1}\beta_1 + e_{j_2}\beta_2 + \ldots + e_{j_\tau}\beta_\tau \\ s_2 &= e(\alpha^2) = e_{j_1}\beta_1^2 + e_{j_2}\beta_2^2 + \ldots + e_{j_\tau}\beta_\tau^2 \\ &\vdots \\ s_{2t} &= e(\alpha^{2t}) = e_{j_1}\beta_1^{2t} + e_{j_2}\beta_2^{2t} + \ldots + e_{j_\tau}\beta_\tau^{2t} \end{aligned}$$

- Variables β_1, β_2, ..., β_τ are unknown.
- An algorithm that solves this set of equations is a decoding algorithm for BCH code.

Error location and error evaluation polynomials

- To decode a BCH code, we need to find the error location and the error values in non-binary case. Generically, it defines two important polynomials:
  - Error-location polynomial is defined as

$$\sigma(X) = (X - \alpha^{-j_1})(X - \alpha^{-j_2}) \ldots (X - \alpha^{-j_\tau}) = \prod_{l=1}^{\tau} (X - \alpha^{-j_l})$$

    where, assuming there are τ errors, errors are at location j_1, j_2, ..., j_τ.
  - Error-evaluation polynomial is defined as

$$W(X) = \sum_{l=1}^{\tau} e_{j_l} \prod_{\substack{i=1 \\ i \neq l}}^{\tau} (X - \alpha^{-j_i})$$

    where, e_{j_l} is the error values.
- In binary BCH code case, the error values are always 1.

Error location and error evaluation polynomials

- In general, the error values can be calculated by

$$e_{j_h} = \frac{W(\alpha^{-j_h})}{\sigma'(\alpha^{-j_h})}$$

  - $\sigma'(X)$ is the derivative of the error location polynomial $\sigma(X)$ with respect to X, hence there is

$$\sigma'(X) = \sum_{l=1}^{\tau} \prod_{\substack{i=1 \\ i \neq l}}^{\tau} (X - \alpha^{-j_i})$$

if for a specific $X = \alpha^{-j_h}$,

$$\sigma'(\alpha^{-j_h}) = \prod_{\substack{i=1 \\ i \neq h}}^{\tau} (\alpha^{-j_h} - \alpha^{-j_i})$$

Error location and error evaluation polynomials

- continuing from last slide...
  - For error evaluation polynomial, there is

$$W(\alpha^{-j_h}) = \sum_{l=1}^{\tau} e_{j_l} \prod_{\substack{i=1 \\ i \neq l}}^{\tau} (\alpha^{-j_h} - \alpha^{-j_i}) = e_{j_h} \prod_{\substack{i=1 \\ i \neq h}}^{\tau} (\alpha^{-j_h} - \alpha^{-j_i}) \neq 0$$

    where, the other items in the $\sum$ are all become zero, because of they contain item $(\alpha^{-j_h} - \alpha^{-j_h})$ in the production$\prod$, when $i = h$.
  - Combining $W(\alpha^{-j_h})$ and $\sigma'(\alpha^{-j_h})$, we obtain

$$e_{j_h} = \frac{W(\alpha^{-j_h})}{\sigma'(\alpha^{-j_h})}$$

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

The key equation

- There are special relationship between the polynomials $\sigma(X)$, $W(X)$ and $S(X)$.
- It is represented by *the key equation*.
- **Theorem 4.1**: There exists a polynomial $\mu(X)$ such that the polynomials $\sigma(X)$, $W(X)$ and $S(X)$ fits the key equation:

$$\sigma(X)S(X) = -W(X) + \mu(X)X^{2t}$$

- The key equation offers a decoding method for BCH codes.

AARHUS UNIVERSITET
INGENIØRHØJSKOLEN

The Euclidean Algorithm

- Assuming $A \geq B$, or $deg(A) \geq deg(B)$, the initial conditions are $r_{-1} = A$, $r_0 = B$.
- In the iterative calculation, the ith iteration, the value r_i is obtained as the remainder of the division of r_{i-2} by r_{i-1}, that is

$$r_i = r_{i-2} - q_i r_{i-1}$$

- There exists s_i and t_i, they meet

$$r_i = s_i A + t_i B$$

- s_i and t_i also meet

$$\begin{aligned} s_i &= s_{i-2} - q_i s_{i-1} \\ t_i &= t_{i-2} - q_i t_{i-1} \end{aligned}$$

- Then

$$\begin{aligned} r_{-1} = s_{-1}A + t_{-1}B = A &\quad\Rightarrow\quad s_{-1} = 1,\ t_{-1} = 0 \\ r_0 = s_0 A + t_0 B = B &\quad\Rightarrow\quad s_0 = 0,\ t_0 = 1 \end{aligned}$$

- Iteration stops when $r_i < t_i$ or $deg(r_i) < deg(t_i)$.

The Euclidean Algorithm

- Example: To calculate the highest common factor (HCF) of $A = 112$ and $B = 54$ by Euclidean algorithm.
- **Solution**: Construct an Euclidean algorithm table:

i	$r_i = r_{i-2} - q_i r_{i-1}$	q_i	$s_i = s_{i-2} - q_i s_{i-1}$	$t_i = t_{i-2} - q_i t_{i-1}$
-1	$r_{-1} = A = 112$	-	$s_{-1} = 1$	$t_{-1} = 0$
0	$r_0 = B = 54$	-	$s_0 = 0$	$t_0 = 1$
1	4	2	1	-2
2	2	13	-13	27

- $r_2 < t_2$, iteration stops.
- So the HCF of 112 and 54 is 2.

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

Apply Euclidean Algorithm to the Key Equation

- Let X^{2t} be A and syndrome polynomial $S(X)$ be B, there is

$$\begin{array}{rclcrcl} r_{-1}(X) & = & X^{2t} & \quad & r_0(X) & = & S(X) \\ s_{-1}(X) & = & 1 & & s_0(X) & = & 0 \\ t_{-1}(X) & = & 0 & & t_0(X) & = & 1 \end{array}$$

- Start Euclidean algorithm iteration and stop when $deg(r_i(X)) < deg(t_i(X))$.
- Assuming iteration stops at ith recursion, there is

$$\begin{aligned} r_i(X) &= s_i(X)A + t_i(X)B \\ r_i(X) &= s_i(X)X^{2t} + t_i(X)S(X) \end{aligned}$$

- Comparing with the key equation:

$$-W(X) = -\mu(X)X^{2t} + \sigma(X)S(X)$$

- As $\sigma(X)$ is a monic polynomial, we can multiply λ to make the resulting polynomial $\lambda t_i(X)$ be a monic polynomial.

$$\lambda r_i(X) = \lambda s_i(X)X^{2t} + \lambda t_i(X)S(X) = -W(X) = -\mu(X)X^{2t} + \sigma(X)S(X)$$

Thus

$$W(X) = -\lambda r_i(X) \quad \sigma(X) = \lambda t_i(X)$$

AARHUS UNIVERSITET
INGENIØRHØJSKOLEN

Apply Euclidean Algorithm to the Key Equation

- **Example 4.5**: For binary BCH code $C_{BCH}(15,7)$ with $t = 2$, the received vector $\mathbf{r} = (100000001000000)$. The GF($2^4$) is generated by primitive polynomial $p_i(X) = 1 + X + X^4$. Find the code polynomial using Euclidean algorithm.
- **Solution**:
  - Calculate the components in the syndrome vector:

$$\begin{array}{ccccc} s_1 & = & r(\alpha) & = & \alpha^2 \\ s_2 & = & r(\alpha^2) & = & \alpha^4 \\ s_3 & = & r(\alpha^3) & = & \alpha^7 \\ s_4 & = & r(\alpha^4) & = & \alpha^8 \end{array}$$

    Therefore, the syndrome polynomial is

$$S(X) = s_1 + s_2X + s_3X^2 + s_4X^3 = \alpha^2 + \alpha^4X + \alpha^7X^2 + \alpha^8X^3$$

  - Initialize process in Euclidean algorithm:

$$\begin{array}{cc} r_{-1} = A = X^{2t} = X^4 & r_0 = B = S(X) \\ s_{-1} = 1 & t_{-1} = 0 \\ s_0 = 0 & t_0 = 1 \end{array}$$

Apply Euclidean Algorithm to the Key Equation

Iteration starts...

$$
\begin{aligned}
S(X) &= \alpha^8 X^3 + \alpha^7 X^2 + \alpha^4 X + \alpha^2 \\
r_i(X) &= s_i(X) X^{2t} + t_i(X) S(X)
\end{aligned}
$$

- Iteration of Euclidean algorithm for the key equation:

i	$r_i = r_{i-2} - q_i r_{i-1}$	q_i	$t_i = t_{i-2} - q_i t_{i-1}$
-1	$A = X^{2t} = X^4$	-	0
0	$B = S(X) = \alpha^8 X^3 + \alpha^7 X^2 + \alpha^4 X + \alpha^2$	-	1
1	$\alpha^4 X^2 + \alpha^{13} X + \alpha^8$	$\alpha^7 X + \alpha^6$	$\alpha^7 X + \alpha^6$
2	α^5	$\alpha^4 X + \alpha^8$	$\alpha^{11} X^2 + \alpha^5 X + \alpha^3$

- As $deg(r_2(X)) < deg(t_2(X))$, iteration stops.

AARHUS UNIVERSITET
INGENIØRHØJSKOLEN

Apply Euclidean Algorithm to the Key Equation

Continuing...

- As $t_i(X) = \alpha^{11}X^2 + \alpha^5 X + \alpha^3$, to let $t_i(X)$ be monic polynomial, λ should be equal to α^4. So
  - Error location polynomial

$$
\begin{aligned}
\sigma(X) = \lambda t_i(X) &= \alpha^4(\alpha^{11}X^2 + \alpha^5 X + \alpha^3) \\
&= X^2 + \alpha^9 X + \alpha^7 \\
&= (X+1)(X+\alpha^7)
\end{aligned}
$$

  - Error evaluation polynomial

$$W(X) = -\lambda r_i(X) = \alpha^4\alpha^5 = \alpha^9$$

Apply Euclidean Algorithm to the Key Equation

Continuing...

- The process of finding the error location is the process of finding the roots of the error-location polynomial $\sigma(X)$.
- To find the roots of $\sigma(X)$, we can substitute all the elements of the corresponding GF, $1, \alpha, \alpha^2, \ldots, \alpha^{n-1}$, in $\sigma(X)$, which is referred to the *Chien search*.
- Since in GF(2^m), let $n = 2^m - 1$, then $\alpha^n = 1$, there is $\alpha^{-h} = \alpha^{n-h}$ or $\alpha^h = \alpha^{-(n-h)}$.
- If α^h is a root of the error location polynomial, then $n - h$ is the corresponding error location number.
- Performing Chien search in the error location polynomial $\sigma(X)$,
- Two roots of the error polynomial are found: $\alpha^{-j_1} = 1$ and $\alpha^{-j_2} = \alpha^7$, so

$$\begin{aligned} \alpha^0 &= \alpha^{-j_1} \\ j_1 &= 0 \end{aligned} \qquad \begin{aligned} \alpha^7 &= \alpha^{-j_2} \\ j_2 &= 8 \end{aligned}$$

AARHUS UNIVERSITET INGENIØRHØJSKOLEN

Apply Euclidean Algorithm to the Key Equation

Continuing...

- As $j_1 = 0$ and $j_2 = 8$, the error location is 0 and 8.
- Furthermore, it is a binary code, the error value is always equal to 1, therefore, the error polynomial can be expressed by

$$e(X) = 1 + X^8$$

- As the received vector is $\mathbf{r} = (100000001000000)$, $\mathbf{c} = \mathbf{r} + \mathbf{e}$, the code vector is a all-zero vector.
- In general, the error value is calculated by $e_{j_h} = \frac{W(\alpha^{-j_h})}{\sigma'(\alpha^{-j_h})}$
  - It is easy to obtain the derivative of $\sigma(X)$,

$$\sigma'(X) = (X + \alpha^7) + (X + 1) = 1 + \alpha^7 = \alpha^9$$

  - Error values at location 0 and 8:

$$\begin{aligned} e_{j_1} &= \frac{W(\alpha^{-j_1})}{\sigma'(\alpha^{-j_1})} = \frac{W(\alpha^0)}{\sigma'(\alpha^0)} = \frac{\alpha^9}{\alpha^9} = 1 \\ e_{j_2} &= \frac{W(\alpha^{-j_2})}{\sigma'(\alpha^{-j_2})} = \frac{W(\alpha^7)}{\sigma'(\alpha^7)} = \frac{\alpha^9}{\alpha^9} = 1 \end{aligned}$$

AARHUS UNIVERSITET INGENIØRHØJSKOLEN