Error Correction Review

- A single overall parity-check equation detects single errors.
- Hamming codes used m equations to correct one error in $2^m - 1$ bits.
- We can use nonbinary equations if we create *symbols* from sequences of bits. E.g., four bits can represent $0, 1, \ldots, 15$.
- Nonbinary check equations can use more advanced arithmetic, such as mod 16.

Nonbinary Single Error Correcting Code

The single check equation

$$c_1 + c_2 + \cdots + c_n = 0$$

allows detection of a single symbol error in a received n-tuple.

Furthermore, the syndrome s defined by

$$s = r_1 + r_2 + \cdots + r_n$$

indicates the *magnitude* of the error. If the error is in location i and the incorrect symbol is $r_i = c_i + e_i$, then

$$s \;=\; r_1 + r_2 + \cdots + r_n \;=\; c_1 + \cdots + (c_i + e_i) + \cdots + c_n \;=\; e_i \,.$$

The syndrome tells exactly *what* should be subtracted from the incorrect symbol in order to obtain a codeword.

What is not known is *where* the error is — which symbol is wrong.

More Equations Needed

A second equation is needed to identify the error location.

The effect on syndrome of an error magnitude should be different for each location.

A reasonable choice for this second equation:

$$1 \cdot c_1 + 2 \cdot c_2 + \cdots + n \cdot c_n = 0\,.$$

Now every valid codeword satisfies two equations:

$$\begin{aligned} 1 \cdot c_1 + 1 \cdot c_2 + \cdots + 1 \cdot c_n &= 0 \\ 1 \cdot c_1 + 2 \cdot c_2 + \cdots + n \cdot c_n &= 0 \end{aligned}$$

We can derive encoding equations to express c_1, c_2 in terms of $c_3, \ldots, c_n$.

Example: let symbols be 4-bit values with addition modulo 16. For $n = 15$,

$$H = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 3 & \cdots & 15 \end{bmatrix}$$

is check matrix for a code that can *almost* correct single symbols errors.

Decoding Procedure

Suppose there is a single error of magnitude $e_i \neq 0$ in location i.

The syndrome $\mathbf{s} = [\, s_0 \; s_1 \,]$ can be expressed in terms of unknowns i and e_i:

$$s_0 = \sum_{j=1}^{n} r_j \;=\; e_i + \sum_{j=1}^{n} c_j \;=\; e_i$$

$$s_1 = \sum_{j=1}^{n} j r_j \;=\; i e_i + \sum_{j=1}^{n} j c_j \;=\; i e_i$$

We can determine e_i and i from the syndrome equations:

$$e_i = s_0$$

$$i \;=\; \frac{i e_i}{e_i} = \frac{s_1}{s_0}$$

Sadly, division is not always defined for modulo 16 arithmetic. E.g., suppose $s_0 = 4$, $s_1 = 8$. Then $s_1 = i s_0 \bmod 16$ has four solutions:

$$2,\; 6,\; 10,\; 14\,.$$

We cannot be certain where the single error is located.

Galois Field Arithmetic

To make error location work, we define a new arithmetic for bit vectors.

- Addition is exclusive-or (binary addition without carry)
- Multiplication is determined multiplying by 2: bit shift with feedback.

Example: multiplication in $\mathrm{GF}(16)$. $2 = 0010$.

$$2 \cdot 0010 = 0100 = 4$$
$$2 \cdot 0100 = 1000 = 8$$
$$2 \cdot 1000 = 0011 = 3$$

Fact: every nonzero 4-bit sequence is a power of 2.

2EC BCH code

Let $\alpha = 2$ and define two equations using α and α^3.

The parity-check matrix is the following:

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \cdots & \alpha^{n-1} \\ 1 & \alpha^3 & \alpha^6 & \cdots & \alpha^{3(n-1)} \end{bmatrix}$$

Codewords defined by H are the polynomials $c(x)$ with zeroes α and α^3.

So every codeword is a multiple of the minimal polynomials of α and α^3.

Therefore the number of check bits satisfies $n - k \leq 2m$.

Syndromes for 2EC BCH code

Consider any error pattern of weight 2:

$$e(x) = x^{i_1} + x^{i_2}\,, \quad 0 \le i_1 < i_2 < n$$

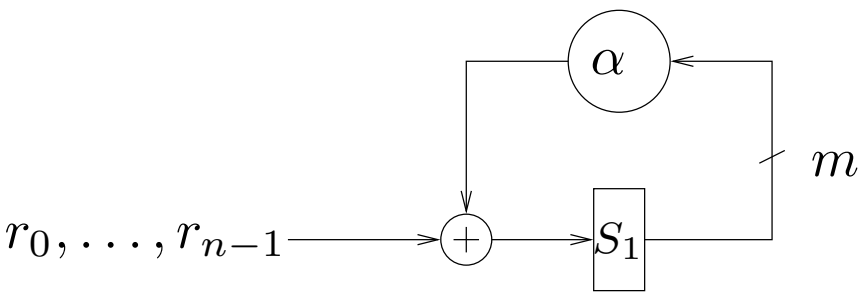
Syndrome of $r(x)$ has two *partial syndrome* components S_1, S_3:

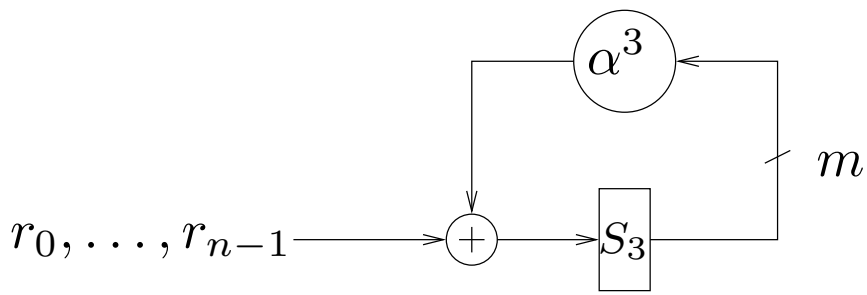
$$S_1 = r(\alpha) = e(\alpha) = \alpha^{i_1} + \alpha^{i_2}$$

$$S_3 = r(\alpha^3) = e(\alpha^3) = \alpha^{3i_1} + \alpha^{3i_2} = (\alpha^{i_1})^3 + (\alpha^{i_2})^3$$

S_1, S_3 are *known* quantities in this system of equations; they are computed at run time.

Partial syndromes can be computed by multiply-accumulate circuits.





Unknowns i_1 and i_2 appear in the exponent — equations are *transcendental*.

2EC BCH decoding: change of variable

Let us simplify the problem by a change of variables:

$$X_1 = \alpha^{i_1}\,, \quad X_2 = \alpha^{i_2}$$

This yields a system of *algebraic* equations of degree 3:

$$S_1 = \alpha^{i_1} + \alpha^{i_2} \;=\; X_1 + X_2$$

$$S_3 = \alpha^{3i_1} + \alpha^{3i_2} \;=\; X_1^3 + X_2^3$$

X_1 and X_2 are called the *error locators*.

Error locations are obtained from X_1, X_2 by taking logarithms to base α:

$$i_1 = \log_\alpha X_1$$

$$i_2 = \log_\alpha X_2$$

The logarithms can be calculated using lookup tables or sequential search.

2EC BCH decoding: degree reduction

We can replace the equation of degree 3:

$$\frac{S_3}{S_1} = \frac{X_1^3 + X_2^3}{X_1 + X_2} = X_1^2 + X_1X_2 + X_2^2$$
$$= (X_1 + X_2)^2 + X_1X_2 \; = \; S_1^2 + X_1X_2$$

Next rewrite the two equations for X_1, X_2 to obtain a system of degree 2:

$$X_1 + X_2 = S_1$$
$$X_1 \cdot X_2 = \frac{S_3}{S_1} + S_1^2 = \frac{S_3 + S_1^3}{S_1}$$

In other words, from the partial syndromes S_1, S_3 (known quantities) we can compute the sum and product of the unknowns X_1, X_2.

We have obtained the coefficients of a quadratic polynomial whose zeroes are X_1, X_2.

2EC BCH decoding: polynomial equation

Given the sum and product of X_1, X_2, we can construct the quadratic polynomial $\Lambda^R(x)$ whose zeroes are X_1 and X_2:

$$\Lambda^R(x) = (x - X_1)(x - X_2) \;=\; x^2 - (X_1 + X_2)x + X_1 X_2$$

$$= x^2 + S_1 x + \frac{S_3 + S_1^3}{S_1} \;=\; x^2 + \Lambda_1 x + \Lambda_2$$

If there are two errors, then $X_1 \neq X_2$. Therefore

$$\Lambda_1 = S_1 = X_1 + X_2 \neq 0$$

and the coefficient

$$\Lambda_2 = \frac{S_3 + S_1^3}{S_1}$$

is well defined and nonzero because $X_1 \neq 0$ and $X_2 \neq 0$.

2EC BCH decoding: other cases

When there are two errors, the error locators are the zeroes of $\Lambda^R(x)$.

- Bit error locations can be found by evaluating $\Lambda^R(\alpha^i)$ for $i = 0, 1, \ldots, n-1$.
- Quadratic formula cannot be used in $GF2^m$. However, there are efficient methods for factoring quadratic polynomials over $GF2^m$.

What if fewer than two errors?

- If one error, the error pattern $e(x) = x^{i_1}$ has syndrome components:

$$S_1 = e(\alpha) = \alpha^{i_1} = X_1$$
$$S_3 = e(\alpha^3) = \alpha^{3i_1} = X_1^3$$

  In this case, $S_3 = S_1^3$ and $\Lambda_2 = 0$. X_1 is zero of $\Lambda^R(x) = x - S_1$.
- When there is no error, $S_1 = S_3 = 0$. $\Lambda^R(x) = 1$ has no zeroes.

2EC BCH code: decoding procedure summarized

1. Compute partial syndromes: $S_1 = r(\alpha)$, $S_3 = r(\alpha^3)$.
2. If $S_1 = 0$ and $S_3 = 0$ then assume no error — most plausible assumption.
3. If $S_1 \neq 0$ and $S_3 = S_1^3$ then assume 1 error.

   Error locator is $X_1 = S_1$. Error location is i_1 satisfying $X_1 = \alpha^{i_1}$.
4. If $S_1 \neq 0$ and $S_3 \neq S_1^3$ then assume 2 errors.

   Error locators are the zeroes in of $\Lambda^R(x)$, where

$$\Lambda^R(x) = x^2 + S_1 x + \frac{S_3 + S_1^3}{S_1}.$$

   If $X_1 = \alpha^{i_1}$, $X_2 = \alpha^{i_2}$ are zeroes of $\Lambda^R(x)$, errors are in locations i_1, i_2.
5. If $\Lambda^R(x)$ does not have 2 zeroes, or if $S_1 = 0$ and $S_3 \neq 0$, then ≥ 3 errors have occurred — detectable but uncorrectable.

Reed-Solomon Codes

Reed-Solomon codes use $2t$ equations to correct t errors.

$$
\begin{aligned}
0 &= c_0 + c_1\alpha + c_2\alpha^2 + \cdots + c_{n-1}\alpha^{n-1} \\
0 &= c_0 + c_1\alpha^2 + c_2\alpha^4 + \cdots + c_{n-1}\alpha^{2(n-1)} \\
&\vdots \\
0 &= c_0 + c_1\alpha^{2t} + c_2\alpha^{4t} + \cdots + c_{n-1}\alpha^{2t(n-1)}
\end{aligned}
$$

The codeword symbols c_i and α are m bits quantities, and $n = 2^m - 1$.

Multiplication and addition in these equations use Galois field arithmetic: every nonzero element has a reciprocal, so decoding equations work.

Reed-Solomon codes are used in CDs, DVDs, and almost all disk drives.