

Wireless Information Transmission System Lab.

Chapter 6

BCH Codes



National Sun Yat-sen University

Outline

- Binary Primitive BCH Codes
- Decoding of the BCH Codes
- Implementation of Galois Field Arithmetic
- Implementation of Error Correction
- Nonbinary BCH Codes and Reed-Solomon Codes

Preface

- The Bose, Chaudhuri, and Hocquenghem (BCH) codes form a large class of powerful random error-correcting cyclic codes.
 - This class of codes is a remarkable generalization of the Hamming codes for multiple-error correction.
 - Binary BCH codes were discovered by Hocquenghem in 1959 and independently by Bose and Chaudhuri in 1960.
 - Generalization of the binary BCH codes to codes in p^m symbols (where p is a prime) was obtained by Gorenstein and Zierler.
 - Among the nonbinary BCH codes, the most important subclass is the class of Reed-Solomon (RS) codes.
 - Among all the decoding algorithms for BCH codes, Berlekamp's iterative algorithm, and Chien's search algorithm are the most efficient ones.
-

Wireless Information Transmission System Lab.

Description of the Codes



National Sun Yat-sen University

Binary Primitive BCH Codes

- ✿ \forall integer $m \geq 3$, $t < 2^{m-1}$, \exists a binary BCH codes with
 - $n = 2^m - 1 \rightarrow$ Block length
 - $n - k \leq mt \rightarrow$ Number of parity-check digits
 - $d_{\min} \geq 2t + 1 \rightarrow$ Minimum distance
- ✿ Clearly, this code is capable of correcting any combination of t or fewer errors in a block of $n = 2^m - 1$ digits. We call this code a t -error-correcting BCH code. The generator polynomial of this code is specified in terms of its roots from the Galois field $GF(2^m)$
- ✿ Let α be a primitive element of $GF(2^m)$. The generator poly. $g(x)$ of the t -error-correcting BCH code of length $2^m - 1$ is the lowest-degree poly. over $GF(2)$ which has

$$\alpha \cdot \alpha^2 \cdot \alpha^3 \dots \alpha^{2t}$$
 as its roots.

Binary Primitive BCH Codes

- It follows from [Theorem 2.11](#) that $\mathbf{g}(x)$ has $\alpha, \alpha^2, \dots, \alpha^{2t}$ and their conjugates as all its roots. Let $\phi_i(x)$ be the [minimal poly.](#) of α^i . Then $\mathbf{g}(x)$ must be the *least common multiple* (LCM) of $\phi_1(x), \phi_2(x), \dots, \phi_{2t}(x)$, that is,

$$g(x) = \text{LCM} \{ \phi_1(x), \phi_2(x), \dots, \phi_{2t}(x) \}$$

- If i is an even integer, it can be expressed as a product of the following form :

$$i = i'2^l,$$

where i' is an odd number and $l \geq 1$. Then $\alpha^i = (\alpha^{i'})^{2^l}$ is a conjugate of $\alpha^{i'}$ and therefore α^i and $\alpha^{i'}$ have the same minimal poly., that is,

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

Binary Primitive BCH Codes

- Hence, even power of α has the same minimal poly. as some preceding odd power of α .

$$g(x) = \text{LCM} \left\{ \phi_1(x), \phi_3(x), \dots, \phi_{2t-1}(x) \right\}$$

$$\begin{aligned} & \bullet \quad \deg [\phi_i(x)] \leq m \\ & \therefore \deg [g(x)] \leq mt \quad \therefore n - k \leq mt \end{aligned}$$

- The BCH codes defined above are usually called primitive (or narrow-sense) BCH codes.

Binary Primitive BCH Codes

- ✿ The single-error-correcting BCH codes of length $2^m - 1$ is generated by $g(x) = \phi_1(x)$ since $t = 1$.
 $\because \alpha$ is a primitive element of $GF(2^m)$
 $\therefore \phi_1(x)$ is a primitive poly. of degree m
 $(\alpha^{2^0}, \alpha^{2^1}, \alpha^{2^2} \dots, \alpha^{2^m} = 1)$
 \therefore the single-error-correcting BCH codes of length $2^m - 1$ is a Hamming code
- ✿ Ex 6.1
 α is a primitive element of $GF(2^4)$ given by Table 2.8 such that $1 + \alpha + \alpha^4 = 0$. The minimal polynomials of $\alpha, \alpha^3, \alpha^5$ are

Binary Primitive BCH Codes

$$\therefore \phi_1(x) = 1 + x + x^4$$

$$\phi_3(x) = 1 + x + x^2 + x^3 + x^4$$

$$\phi_5(x) = 1 + x + x^5$$

- ✿ The double-error-correcting BCH code of length $n=2^4-1=15$ is generated by $g(x) = \text{LCM}\{\phi_1(x), \phi_3(x)\}$
Since $\phi_1(x)$ and $\phi_3(x)$ are two distinct irreducible polynomials,

$$\begin{aligned} g(x) &= (1 + x + x^4)(1 + x + x^2 + x^3 + x^4) \\ &= 1 + x^4 + x^6 + x^7 + x^8 \end{aligned}$$

$\therefore (15, 7)$ cyclic code with $d_{\min} \geq 5$

$$\because W(g(x)) = 5 \quad \therefore d_{\min} = 5$$

Binary Primitive BCH Codes

- The triple-error-correcting BCH code of length 15 is generated by

$$\begin{aligned}
 g(x) &= \text{LCM } \{\phi_1(x), \phi_3(x), \phi_5(x)\} \\
 &= (1 + x + x^4)(1 + x + x^2 + x^3 + x^4)(1 + x + x^2) \\
 &= 1 + x + x^2 + x^4 + x^5 + x^8 + x^{10}
 \end{aligned}$$

It's a (15,5) cyclic code with $d_{\min} \geq 7$. Since $W(g(x)) = 7, d_{\min} = 7$.

- Let $v(x) = v_0 + v_1x + \dots + v_{n-1}x^{n-1}$ be a poly. with $v_i \in GF(2)$
 If $v(x)$ has roots $\alpha, \alpha^2, \dots, \alpha^{2t}$, then $v(x)$ is divisible by
 $\phi_1(x), \phi_2(x), \dots, \phi_{2t}(x)$.
- $v(x)$ is a code poly. because $v(x) | g(x) = \text{LCM}\{\phi_1(x), \phi_2(x), \dots, \phi_{2t}(x)\}$

Binary Primitive BCH Codes

- We have a new definition for t -error-correcting BCH code:

A binary n -tuple $\mathbf{v} = (v_0, v_1, v_2, \dots, v_{n-1})$ is a code word if and only if the poly. $v(x) = v_0 + v_1x + \dots + v_{n-1}x^{n-1}$ has $\alpha, \alpha^2, \dots, \alpha^{2t}$ as roots i.e. $v(\alpha^i) = v_0 + v_1\alpha^i + v_2\alpha^{2i} + \dots + v_{n-1}\alpha^{(n-1)i} = 0$

$$(v_0, v_1, v_2, \dots, v_{n-1}) \cdot \begin{bmatrix} 1 \\ \alpha^i \\ \vdots \\ \alpha^{(n-1)i} \end{bmatrix} = 0 \text{ for } 1 \leq i \leq 2t.$$

Binary Primitive BCH Codes

✳ Let

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

If \mathbf{v} is a code word in the t -error-correcting BCH code, then

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

The code is the null space of the matrix \mathbf{H} and \mathbf{H} is the parity-check matrix of the code.

Binary Primitive BCH Codes

- α^j is a conjugate of α^i , then $v(\alpha^j) = 0$ iff $v(\alpha^i) = 0$ Thm. 2.11
 j -th row of \mathbf{H} can be omitted. As a result \mathbf{H} can be reduced to the following form :

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

- EX 6.2
 double-error-correcting BCH code of length $n = 2^4 - 1 = 15$, $(15,7)$ code. Let α be a primitive element in $GF(2^4)$

Binary Primitive BCH Codes

The parity-check matrix is

$$H = \begin{bmatrix} 1 & \alpha & \alpha^2 & \dots & \alpha^{14} \\ 1 & \alpha^3 & \alpha^6 & \dots & \alpha^{42} \end{bmatrix}$$

Using $\alpha^{15} = 1$, and representing each entry of H by its 4-tuple,

$$H = \left[\begin{array}{ccccccccc} 100010011010111 \\ 010011010111100 \\ 001001101011110 \\ 000100110101111 \\ \hline 100011000110001 \\ 000110001100011 \\ 001010010100101 \\ 011110111101111 \end{array} \right] \quad (\text{see p.149})$$

Binary Primitive BCH Codes

★ FACT:

The t -error-correcting BCH code indeed has $d_{\min} \geq 2t + 1$

(pf): Suppose $\exists \underline{v} \neq \underline{0}$ such that $W(v) = \delta \leq 2t$

Let $v_{j1}, v_{j2}, \dots, v_{j\delta}$ be the nonzero components of \underline{v} (i.e. all ones)

$$\Rightarrow \underline{0} = \underline{v}H^T = (v_{j1} v_{j2} \dots v_{j\delta}) \begin{bmatrix} \alpha^{j1} (\alpha^2)^{j1} \cdots (\alpha^{2t})^{j1} \\ \alpha^{j2} (\alpha^2)^{j2} \cdots (\alpha^{2t})^{j2} \\ \vdots \quad \vdots \\ \alpha^{j\delta} (\alpha^2)^{j\delta} \cdots (\alpha^{2t})^{j\delta} \end{bmatrix}$$

$$\Rightarrow (1, 1 \dots, 1) \begin{bmatrix} \alpha^{j1} (\alpha^{j1})^2 \cdots (\alpha^{j1})^{2t} \\ \alpha^{j2} (\alpha^{j2})^2 \cdots (\alpha^{j2})^{2t} \\ \vdots \quad \vdots \quad \vdots \\ \alpha^{j\delta} (\alpha^{j\delta})^2 \cdots (\alpha^{j\delta})^{2t} \end{bmatrix} = \underline{0} \dots \otimes$$

Binary Primitive BCH Codes

$$(1, 1, \dots, 1) \begin{bmatrix} \alpha^{j_1} (\alpha^{j_1})^2 \cdots (\alpha^{j_1})^\delta \\ \alpha^{j_2} (\alpha^{j_2})^2 \cdots (\alpha^{j_2})^\delta \\ \vdots \quad \vdots \quad \vdots \\ \alpha^{j_\delta} (\alpha^{j_\delta})^2 \cdots (\alpha^{j_\delta})^\delta \end{bmatrix} = 0$$

$\underbrace{\hspace{10em}}$

A , a $\delta \times \delta$ square matrix $(\delta \leq 2t)$

$$\Rightarrow |A| = 0$$

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

Binary Primitive BCH Codes

- The determinant in the equality above is a Vandermonde determinant which is nonzero. The product on the left-hand side of $\otimes \otimes$ can not be zero. This is a contradiction and hence our assumption that there exists a nonzero code vector v of $W(v) = \delta \leq 2t \therefore d_{\min} \geq 2t + 1$ is invalid.
- $2t+1$ is the designed distance of the t -error-correcting BCH code. The true minimum distance of a BCH code may or may not be equal to its designed distance.
- Binary BCH code with length $n \neq 2^m - 1$ can be constructed in the same manner as for the case $n = 2^m - 1$.
Let β be an element of order n in $GF(2^m)$, $n | 2^m - 1$ and $g(x)$ be the binary polynomial of minimum degree that has $\beta, \beta^2, \dots, \beta^{2^t}$ as roots.

Binary Primitive BCH Codes

- Let $\psi_1(x), \psi_2(x), \dots, \psi_{2t}(x)$ be the minimal poly. of $\beta, \beta^2, \dots, \beta^{2t}$ respectively then

$$g(x) = \text{LCM}\{\psi_1(x), \psi_2(x), \dots, \psi_{2t}(x)\}$$

- $\because \beta^n = 1, \therefore \beta, \beta^2, \dots, \beta^{2t}$ are roots of $x^n + 1$

$$\Rightarrow g(x) \mid (x^n + 1)$$

- We see that $g(x)$ is a factor of $X^n + 1$.
 - The cyclic code generated by $g(x)$ is a t -error-correcting BCH code of length n .
 - The number of parity-check digits $\leq mt$
 - $d_{\min} \geq 2t + 1$.
- If β is not a primitive element of $\text{GF}(2^m)$, the code is called a nonprimitive BCH code.

Binary Primitive BCH Codes

- General definition of binary BCH codes.

$\beta \in GF(2^m)$, ℓ_0 be any nonnegative integer and consider $\beta^{l_0}, \beta^{l_0+1}, \dots, \beta^{l_0+d_0-2}$. For $0 \leq i \leq d_0 - 1$, let $\psi_i(x), n_i$ be the minimal poly. and order of β^{l_0+i} , respectively.

$$g(x) = \text{LCM}\{\psi_0(x), \psi_1(x), \dots, \psi_{d_0-2}(x)\}$$

and the length of the code is

$$n = \text{LCM} \left\{ n_0, n_1, \dots, n_{d_0-2} \right\}$$

Note that: $d_{\min} \geq d_0$

parity-check digits $\leq m(d_0 - 1)$

is capable of correcting $[(d_0 - 1)/2]$ or fewer errors

Binary Primitive BCH Codes

- If we let $l_0 = 1$, $d_0 = 2t+1$ and β be a primitive element of $\text{GF}(2^n)$, the code becomes a t -error-correcting primitive BCH code of length $2^m - 1$.
- If we let $l_0 = 1$, $d_0 = 2t+1$ and β be not a primitive element of $\text{GF}(2^n)$, the code is a nonprimitive t -error-correcting BCH code of length n , which is the order of β .
- For a BCH code with designed distance d_0 , we require $g(x)$ has $d_0 - 1$ consecutive powers of a field element β as roots. This guarantees that the code has $d_{\min} \geq d_0$. This lower bound on the minimum distance is called the BCH bound.
- In the rest of this chapter, we consider only the primitive BCH codes.

Wireless Information Transmission System Lab.

Decoding of the BCH Codes



National Sun Yat-sen University

Decoding of the BCH Codes

- Suppose that a code word $\mathbf{v}(x) = v_0 + v_1x + v_2x^2 + \dots + v_{n-1}x^{n-1}$ is transmitted and the transmission errors result :

$$\mathbf{r}(x) = r_0 + r_1x + r_2x^2 + \dots + r_{n-1}x^{n-1}$$

Let $\mathbf{e}(x)$ be the error pattern. Then

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

- For decoding, remember

$$H = \begin{bmatrix} 1 & \alpha & . & . & . & \alpha^{n-1} \\ 1 & (\alpha^2) & . & . & . & (\alpha^2)^{n-1} \\ . & . & & & & . \\ . & . & & & & . \\ . & . & & & & . \\ 1 & (\alpha^{2t}) & . & . & . & (\alpha^{2t})^{n-1} \end{bmatrix}$$

Decoding of the BCH Codes

- The syndrome is $2t$ -tuple,

$$\mathbf{S} = (S_1, S_2, \dots, S_{2t}) = \mathbf{r} \cdot \mathbf{H}^T$$

Let $s_i = r(\alpha^i) = r_0 + r_1\alpha^i + \dots + r_{n-1}(\alpha^i)^{n-1} = 0$ for $1 \leq i \leq 2t$

s_i can be evaluated by $b_i(x) = R_{\phi_i(x)}[r(x)]$

$\phi_i(x)$ is the minimal poly. of α^i

$$\because r(x) = a_i(x)\phi_i(x) + b_i(x)$$

$$\therefore s_i = r(\alpha^i) = b_i(\alpha^i)$$

Decoding of the BCH Codes

EX6.4

Consider the double-error-correcting (15, 7) BCH code given in (from Ex6.1). If $\underline{r} = (100000001000000) \Rightarrow r(x) = 1 + x^8$

$$\because \phi_1(x) = \phi_2(x) = \phi_4(x) = 1 + x + x^4 \quad \therefore s_1 = b_1(\alpha) = \alpha^2, s_2 = b_1(\alpha^2) = \alpha^4$$

$$\phi_3(x) = 1 + x + x^2 + x^3 + x^4 \quad s_3 = b_3(\alpha^3) = 1 + \alpha^9$$

$$b_1(x) = R_{\phi_1(x)}[r(x)] = x^2 \quad = 1 + \alpha + \alpha^3 = \alpha^7$$

$$b_3(x) = R_{\phi_3(x)}[r(x)] = 1 + x^3 \quad s_4 = b_1(\alpha^4) = \alpha^8$$

$$\therefore \underline{s} = (\alpha^2, \alpha^4, \alpha^7, \alpha^8)$$

Decoding of the BCH Codes

- ★ $\because v(\alpha^i) = 0$ for $1 \leq i \leq 2t \rightarrow s_i = r(\alpha^i) = e(\alpha^i)$

Suppose $e(x) = x^{j_1} + x^{j_2} + \dots + x^{j_v} \quad 0 \leq j_1 < j_2 < \dots < j_v < n$

$$\Rightarrow s_1 = \alpha^{j_1} + \alpha^{j_2} + \dots + \alpha^{j_v}$$

$$s_2 = (\alpha^{j_1})^2 + (\alpha^{j_2})^2 + \dots + (\alpha^{j_v})^2$$

⋮

$$s_{2t} = (\alpha^{j_1})^{2t} + \dots + (\alpha^{j_v})^{2t}$$

Where $\alpha^{j_1}, \alpha^{j_2}, \dots, \alpha^{j_v}$ are unknown.

Any method for solving these equations is a decoding algorithm for the BCH codes.

Decoding of the BCH Codes

- ✿ Once $\alpha^{j_1}, \alpha^{j_2}, \dots, \alpha^{j_v}$ have been found, the powers j_1, j_2, \dots, j_v tell us the error locations in $e(x)$
- ✿ If the number of errors in $e(x)$ is t or less, the solution that yields an error pattern with the smallest number of errors is the right solution.
- ✿ For convenience, let $\beta_\ell = \alpha^{j_\ell}$, $1 \leq \ell \leq v$ be the error location numbers.

$$\left. \begin{array}{l} s_1 = \beta_1 + \beta_2 + \dots + \beta_v \\ s_2 = \beta_1^2 + \beta_2^2 + \dots + \beta_v^2 \\ \vdots \\ s_{2t} = \beta_1^{2t} + \beta_2^{2t} + \dots + \beta_v^{2t} \end{array} \right\} \begin{array}{l} \text{Power-sum} \\ \text{Symmetric function} \end{array}$$

Decoding of the BCH Codes

- Define

$$\begin{aligned}\sigma(x) &= (1 + \beta_1 x)(1 + \beta_2 x) \dots (1 + \beta_v x) \\ &\downarrow \\ &= \sigma_0 + \sigma_1 x + \sigma_2 x^2 + \dots + \sigma_v x^v \\ &\text{: error-location poly.}\end{aligned}$$

- The roots of $\sigma(x)$ are $\beta_1^{-1}, \beta_2^{-1}, \dots, \beta_v^{-1}$, which are the inverses of the error location numbers.

$$\sigma_0 = 1$$

$$\sigma_1 = \beta_1 + \beta_2 + \dots + \beta_v$$

$$\sigma_2 = \beta_1\beta_2 + \beta_2\beta_3 + \dots + \beta_{v-1}\beta_v$$

$$\vdots$$

$$\sigma_v = \beta_1\beta_2 \dots \beta_v$$

Decoding of the BCH Codes

These coefficients are known as elementary symmetric functions.

σ_i 's are related to s_j 's by Newton's identities

$$s_1 + \sigma_1 = 0$$

$$s_2 + \sigma_1 s_1 + 2\sigma_2 = 0$$

$$s_3 + \sigma_1 s_2 + \sigma_2 s_1 + 3\sigma_3 = 0$$

⋮

$$s_v + \sigma_1 s_{v-1} + \dots + \sigma_{v-1} s_1 + v\sigma_v = 0$$

$$s_{v+1} + \sigma_1 s_v + \dots + \sigma_{v-1} s_2 + \sigma_v s_1 = 0$$

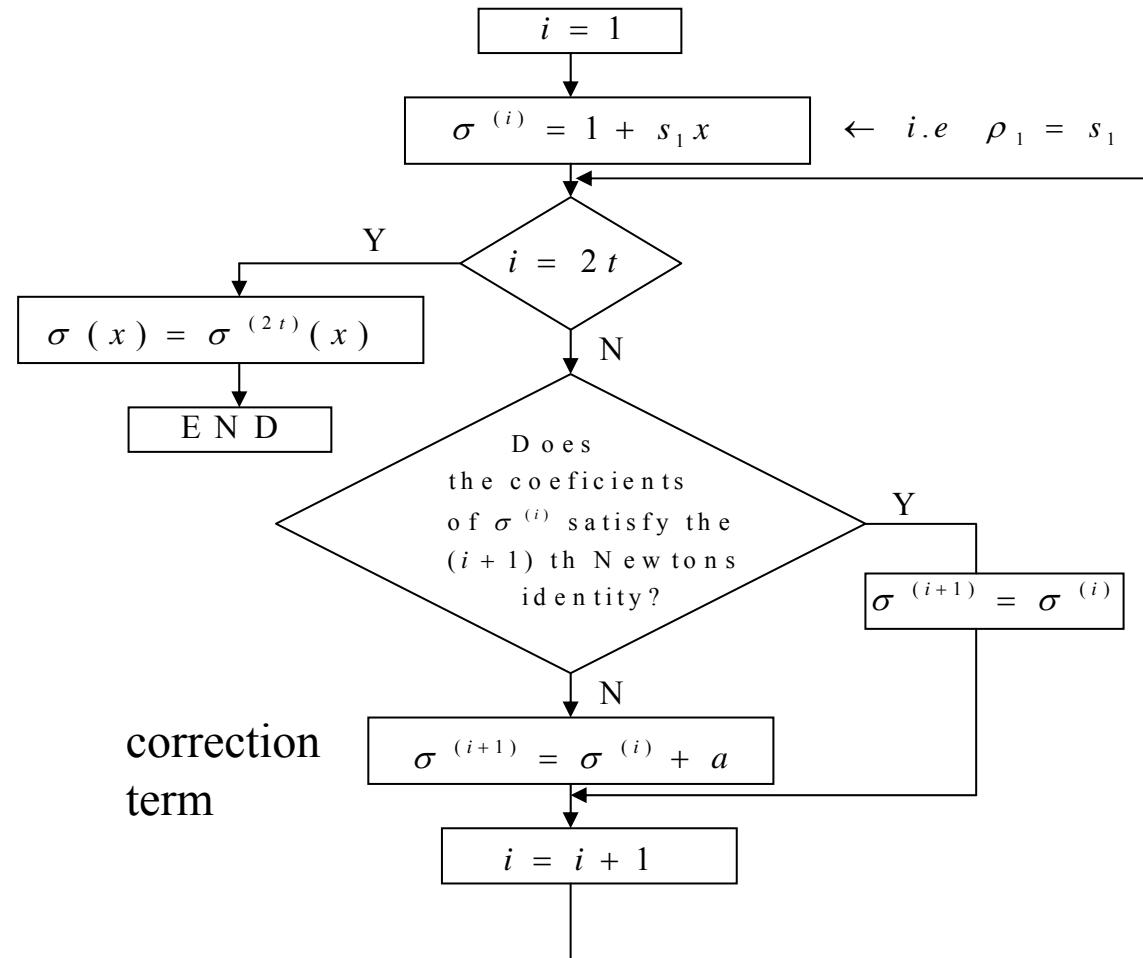
Note that, for binary case, $\because 1+1=2=0$.
we have

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

- The equations may have many solutions.
- We want to find the solution that yields a $\sigma(X)$ of minimal degree. This $\sigma(X)$ would produce an error pattern with minimum number of errors.
- Decoding Procedure
 - step 1. Compute $\underline{s} = (s_1, s_2, \dots, s_{2t})$ from $r(x)$
 - step 2. Determine $\sigma(x)$ from \underline{s}
 - step 3. Determine the error-location number
 $\beta_1, \beta_2, \dots, \beta_v$ by finding the roots of $\sigma(x)$
and correct the errors in $r(x)$

Decoding of the BCH Codes

- Iterative Algorithm for finding $\sigma(x)$ (Berlekamps iterative algorithm)



Decoding of the BCH Codes

- How to add a correction term to $\sigma^{(i)}$?

Let $\sigma^{(\mu)} = 1 + \sigma_1^{(\mu)}x + \sigma_2^{(\mu)}x^2 + \dots + \sigma_{\ell_\mu}^{(\mu)}x^{\ell_\mu}$

be the minimum-degree poly. determined at the μ th step.

To determine $\sigma^{(\mu+1)}(x) \Rightarrow$ compute μ th discrepancy

$$d_\mu = s_{\mu+1} + \sigma_1^{(\mu)}s_\mu + \sigma_2^{(\mu)}s_{\mu-1} + \dots + \sigma_{\ell_\mu}^{(\mu)}s_{\mu+1-\ell_\mu}$$

If $d_\mu = 0 \Rightarrow \sigma^{(\mu+1)}(x) = \sigma^{(\mu)}(x)$

If $d_\mu \neq 0$, go back to the steps prior to the μ th step and determine $\sigma^{(\rho)}(x)$ s.t. ρ th discrepancy $d_\rho \neq 0$, and $\rho - \ell_\rho$ has the largest value. ($\ell_\rho = \deg [\sigma^{(\rho)}(x)]$). Then

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

Decoding of the BCH Codes

is the minimum-degree poly. whose coefficients satisfy the first $\mu+1$ Newton's identities

- ★ To carry out the iteration of finding $\sigma(X)$, we begin with the following table:

μ	$\sigma^{(\mu)}(x)$	d_μ	ℓ_μ	$\mu - \ell_\mu$
-1	1	1	0	-1
0	1	S_1	0	0
1				
2				
:				
$2t$				

Decoding of the BCH Codes

- l_μ is the degree of $\sigma^{(\mu)}(X)$.
- If $d_\mu = 0$, then $\sigma_\mu^{(\mu+1)}(X) = \sigma^{(\mu)}(X)$ and $l_{\mu+1} = l_\mu$.
- If $d_\mu \neq 0$, find another row ρ prior to the μ th row such that $d_\rho \neq 0$ and the number $\rho - l_\rho$ in the last column of the table has the largest value. Then $\sigma^{(\mu+1)}(X)$ is given by

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

- and $l_{\mu+1} = \max(l_\mu, l_\rho + \mu - \rho)$.
- In either case, $d_{\mu+1} = s_{\mu+2} + \sigma_1^{(\mu+1)} s_{\mu+1} + \dots + \sigma_{\ell_{\mu+1}}^{(\mu+1)} s_{\mu+2-\ell_{\mu+1}}$
 - The polynomial $\sigma^{(2t)}(X)$ in the last row should be the required $\sigma(X)$.

Decoding of the BCH Codes

Ex 6.5

Consider (15,5) triple-error-correcting BCH codes given in ex 6.1

$$p(x) = 1 + x + x^4$$

$$\underline{v} = 0 \rightarrow \oplus \xrightarrow{\downarrow \underline{e}} r = x^3 + x^5 + x^{12}$$

$$\phi_1(x) = \phi_2(x) = \phi_4(x) = 1 + x + x^4$$

$$\phi_3(x) = \phi_6(x) = 1 + x + x^2 + x^3 + x^4$$

$$\phi_5(x) = 1 + x + x^2$$

Decoding of the BCH Codes

$$b_1(x) = R_{\phi_1(x)}[r(x)] = 1$$

$$b_3(x) = R_{\phi_3(x)}[r(x)] = 1 + x^2 + x^3$$

$$b_5(x) = R_{\phi_5(x)}[r(x)] = x^2$$

$$s_1 = s_2 = s_4 = 1$$

$$s_3 = 1 + \alpha^6 + \alpha^9 = \alpha^{10}$$

$$s_6 = 1 + \alpha^{12} + \alpha^{18} = \alpha^5$$

$$s_5 = \alpha^{10}$$

$$\therefore \underline{s} = (1, 1, \alpha^{10}, 1, \alpha^{10}, \alpha^5)$$

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

Decoding of the BCH Codes

- $S_1 = 1$
 - $d_0 = 1 \neq 0$
 - $\rho = -1$
 - $\sigma^{(1)}(X) = \sigma^{(0)}(X) + d_0 d_{-1}^{-1} X^{(0+1)} \sigma^{(-1)}(X) = 1 + 1 \cdot 1 \cdot X \cdot 1 = 1 + X$
 - $l_1 = \max(l_0, l_{-1} + \mu - \rho) = \max(0, 0 - 0 + 1) = 1$
 - $\mu - l_\mu = 1 - l_1 = 1 - 1 = 0$
 - $d_1 = S_2 + \sigma_1^{(1)} S_1 = 1 + 1 \cdot 1 = 0$
 - $\sigma^{(2)}(X) = \sigma^{(1)}(X) = 1 + X$
 - $l_2 = l_1 = 1$
 - $\mu - l_\mu = 2 - l_2 = 2 - 1 = 1$
 - $d_2 = S_3 + \sigma_1^{(2)} S_2 + \sigma_2^{(2)} S_1 = \alpha^{10} + 1 \cdot 1 + 0 \cdot 1 = (1 + \alpha + \alpha^2) + 1 = \alpha^5$
-

Decoding of the BCH Codes

- ✿ $d_2 = \alpha^5 \neq 0$
- ✿ $\rho = 0$
- ✿ $\sigma^{(3)}(X) = \sigma^{(2)}(X) + d_2 d_0^{-1} X^{(2-0)}$ $\sigma^{(0)}(X) = 1 + X + \alpha^5 \cdot 1 \cdot X^2 \cdot 1 = 1 + X + \alpha^5 X^2$
- ✿ $l_3 = \max(l_2, l_0 + \mu - \rho) = \max(1, 0 + 2 - 0) = 2$
- ✿ $\mu - l_\mu = 3 - l_3 = 3 - 2 = 1$
- ✿ $d_3 = S_4 + \sigma_1^{(3)}S_3 + \sigma_2^{(3)}S_2 + \sigma_3^{(3)}S_1 = 1 + 1 \cdot \alpha^{10} + \alpha^5 \cdot 1 + 0 \cdot 1$
 $= 1 + (1 + \alpha + \alpha^2) + (\alpha + \alpha^2) = 0$

Decoding of the BCH Codes

μ	$\sigma^{(\mu)}(x)$	d_μ	ℓ_μ	$\mu - \ell_\mu$
-1	1	1	0	-1
0	1	1	0	0
1	$1 + x$	0	1	0
2	$1 + x$	α^5	1	1
3	$1 + x + \alpha^5 x^2$	0	2	1
4	$1 + x + \alpha^5 x^2$	α^{10}	2	2
5	$1 + x + \alpha^5 x^3$	0	3	2
6	$1 + x + \alpha^5 x^3$	-	-	-

$$\begin{aligned}\therefore \sigma(x) &= \sigma^{(6)}(x) = 1 + x + \alpha^5 x^3 \\ &= (1 + \alpha^{-3} x)(1 + \alpha^{-10} x)(1 + \alpha^{-12} x)\end{aligned}$$

$$\therefore x = \alpha^3, \alpha^{10}, \alpha^{12}$$

Decoding of the BCH Codes

error location numbers

$$\beta_1 = \alpha^{-3} = \alpha^{12}, \beta_2 = \alpha^{-10} = \alpha^5, \beta_3 = \alpha^{-12} = \alpha^3$$

$$\therefore e(x) = x^3 + x^5 + x^{12}$$

$$\therefore r(x) = r(x) + e(x) = 0$$

- If the number of errors in the received polynomial $\mathbf{r}(X)$ is less than the designed error-correcting capability t of the code, it is not necessary to carry out the $2t$ steps of iteration to find the error-location polynomial $\sigma(X)$.
- It has been shown that if d_{μ} and the discrepancies at the next $t-l_{\mu}-1$ steps are all zeros (i.e. successive $t-l_{\mu}$ zeros), $\sigma^{(\mu)}(X)$ is the error-location polynomial.
- If $v(v \leq t)$ errors occur, only $v+t$ steps of iteration are needed.
- The iterative algorithm described above not only applies to binary BCH codes but also nonbinary BCH codes.

Decoding of the BCH Codes

Simplified Algorithm for finding $\sigma(x)$

- For a binary BCH code, it is only required to fill out a table with t empty rows. Such a table is presented below.

μ	$\sigma^{(\mu)}(x)$	d_μ	I_μ	$2\mu - I_\mu$
-1/2	1	1	0	-1
0	1	S_1	0	0
1				
2				
.				
.				
t				

Decoding of the BCH Codes

1. If $d_\mu = 0$, then $\sigma^{(\mu+1)}(x) = \sigma^{(\mu)}(x)$
2. If $d_\mu \neq 0$, find another row ρ preceding the μ th row,
s.t. $2\rho - \ell_\rho$ is as large as possible and $d_\rho \neq 0$

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

note that $d_{\mu+1} = s_{2\mu+3} + \sigma_1^{(\mu+1)} s_{2\mu+2} + \sigma_2^{(\mu+1)} s_{2\mu+1}$
 $+ \dots + \sigma_{\ell_{\mu+1}}^{(\mu+1)} s_{2\mu+3-\ell_{\mu+1}}$

$$\ell_{\mu+1} = \deg [\sigma^{(\mu+1)}(x)]$$

Decoding of the BCH Codes

- ✿ The polynomial $\sigma^{(t)}(X)$ in the last row should be the required $\sigma(X)$.
- ✿ If it has degree greater than t , there were more than t errors, and generally it is not possible to locate them.
- ✿ The computation required in this simplified algorithm is one-half of the computation required in the general algorithm.
- ✿ The simplified algorithm applies only to binary BCH codes.
- ✿ If the number of errors in the received polynomial $r(X)$ is less than the designed error-correcting capability t of the code, it is not necessary to carry out the t steps of iteration to find the error-location polynomial $\sigma(X)$ for a t -error-correcting binary BCH code.

★ Remarks:

- If $v \leq t$ errors occur, only $\lceil (t+v)/2 \rceil$ steps needed.
- If, for some μ , d_μ and the discrepancies at the next $(t-l_\mu-1)/2$ steps are zero, then $\sigma(X)$ is the error-location poly. . .

★ EX6.6

The simplified table for finding $\sigma(x)$ for the code in ex6.5 is given below. Thus, $\sigma(x) = \sigma^{(3)}(x) = 1 + x + \alpha^5 x^3$.

μ	$\sigma^{(\mu)}(x)$	d_μ	I_μ	$2\mu - I_\mu$
-1/2	1	1	0	-1
0	1	$S_1=1$	0	0
1	$1+S_1x=1+x$	$S_3+S_2S_1=\alpha^5$	1	1(take $\rho = -1/2$)
2	$1+x+\alpha^5 x^2$	α^{10}	2	2(take $\rho = 0$)
1	$1+x+\alpha^5 x^3$	----	3	3(take $\rho = 1$)

Decoding of the BCH Codes

Finding the Error-Location Numbers and Error Correction.

- Consider ex6.6. The error-location poly. has been found to be

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

By substituting $1, \alpha, \alpha^2, \dots, \alpha^{14}$ into it, we find that $\alpha^3, \alpha^{10}, \alpha^{12}$ are the roots of $\sigma(x)$. Therefore, the error location numbers are

$$\alpha^{12}, \alpha^5, \alpha^3 \Rightarrow e(x) = x^3 + x^5 + x^{12}$$

- Chien's procedure: The received vector

$$r(x) = r_0 + r_1 x + r_2 x^2 + \dots + r_{n-1} x^{n-1}$$

is decoded on a bit-by-bit basis. The high-order bits are decoded first. To decode r_{n-1} , the decoder test whether α^{n-1} is an error-location number; this is equivalent to test whether its inverse α is a root of $\sigma(x)$. If α is a root, then

$$1 + \sigma_1 \alpha + \sigma_2 \alpha^2 + \dots + \sigma_v \alpha^v = 0$$

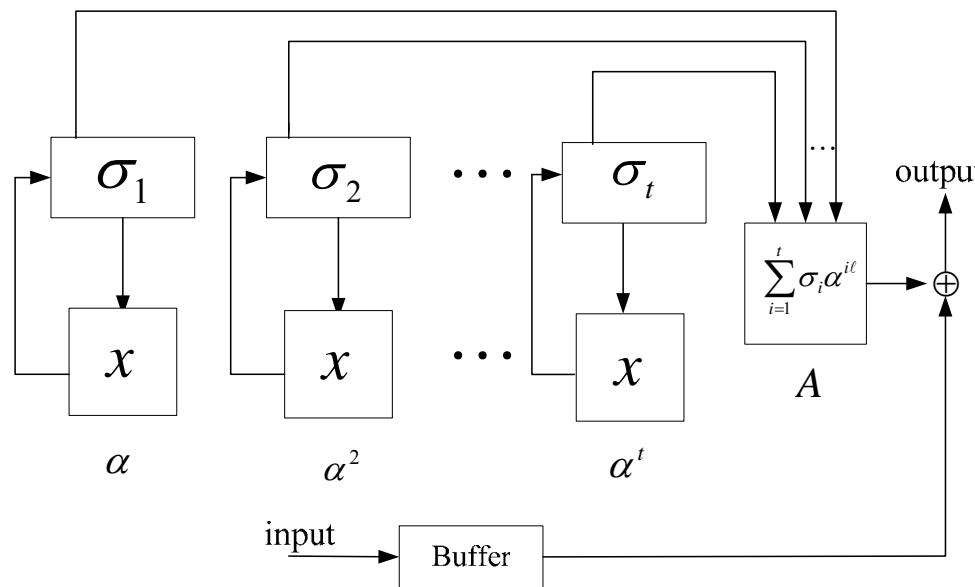
Decoding of the BCH Codes

- To decode r_{n-l} , the decoder forms $\sigma_1\alpha^l, \sigma_2\alpha^{2l}, \dots, \sigma_v\alpha^{vl}$ and tests the sum

$$1 + \sigma_1\alpha^l + \sigma_2\alpha^{2l} + \dots + \sigma_v\alpha^{vl}$$

If the sum is zero, then α^{n-l} is an error-location number and r_{n-l} is an erroneous digit; otherwise, r_{n-l} is a correct digit.

Cyclic error location search unit



Decoding of the BCH Codes

- The $t \sigma$ -registers are initially stored with $\sigma_1, \sigma_2, \dots, \sigma_t$ calculated in step 2 of the decoding ($\sigma_{v+1} = \sigma_{v+2} = \dots = \sigma_v = 0$ for $v < t$). Immediately before r_{n-1} is read out of the buffer, the t multipliers are pulsed once. The multiplications are performed and $\sigma_1\alpha, \sigma_2\alpha^2, \dots, \sigma_v\alpha^v$ are stored in the σ -registers. The output of the logic circuit A is 1 if and only if the sum $1 + \sigma_1\alpha + \sigma_2\alpha^2 + \dots + \sigma_v\alpha^v = 0$; otherwise, the output of A is 0. The digit r_{n-1} is read out of the buffer and corrected by the output of A. Having decoded r_{n-1} , the t multipliers are pulsed again. Now $\sigma_1\alpha^2, \sigma_2\alpha^4, \dots, \sigma_v\alpha^{2v}$ are stored in the σ -registers. The sum

$$1 + \sigma_1\alpha^2 + \sigma_2\alpha^4 + \dots + \sigma_v\alpha^{2v}$$

is tested for 0. The digit r_{n-2} is read out of the buffer and corrected in the same manner as r_{n-1} is corrected.

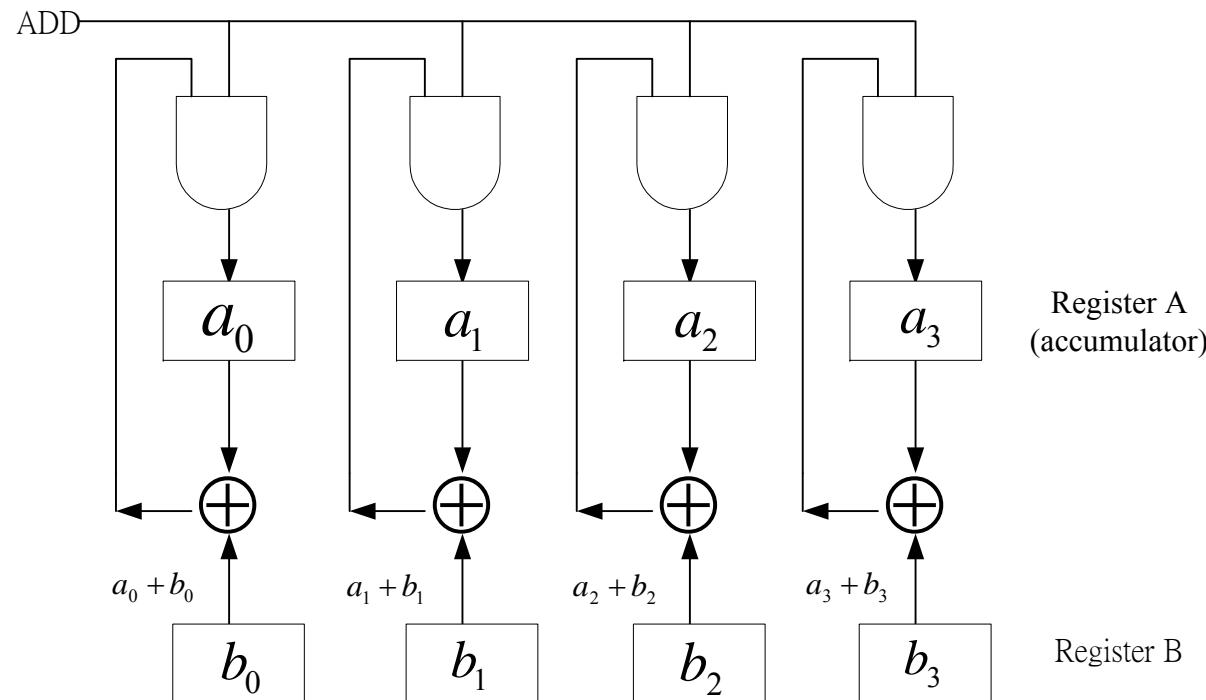
Wireless Information Transmission System Lab.

Implementation of Galois Field Arithmetic



National Sun Yat-sen University

- ✿ Galois field adder ($\text{GF}(2^4)$) given by Table 2.8)

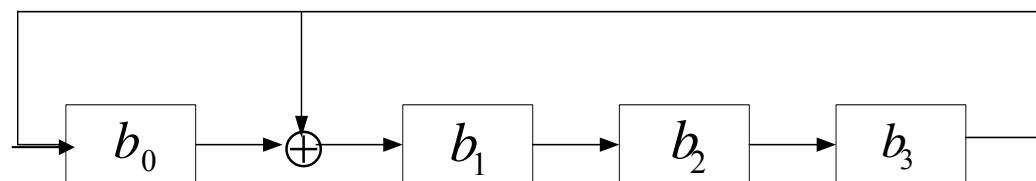


By adding their vector representations with modulo-2 addition.

- For multiplication, we first consider multiplying a field element by a fixed element from the same field. Suppose that we want to multiply a field element β in the $GF(2^4)$ by the primitive element α whose minimal polynomial is $\phi(x) = 1 + x + x^4$.

$$\beta = b_0 + b_1\alpha + b_2\alpha^2 + b_3\alpha^3$$

$$\alpha\beta = b_3 + (b_0 + b_3)\alpha + b_1\alpha^2 + b_2\alpha^3 \because \alpha^4 = 1 + \alpha$$



- Ex1:
let $\beta = \alpha^7 = 1 + \alpha + \alpha^3$. The vector representation of β is (1 1 0 1). Load it into the register. After the register is pulsed, the new content in it will be (1 0 1 0).

the above circuit can be used to generate all the nonzero elements of $GF(2^4)$ by load (1000) first into the register

- Ex2: as another example, multiply β by α^3 :

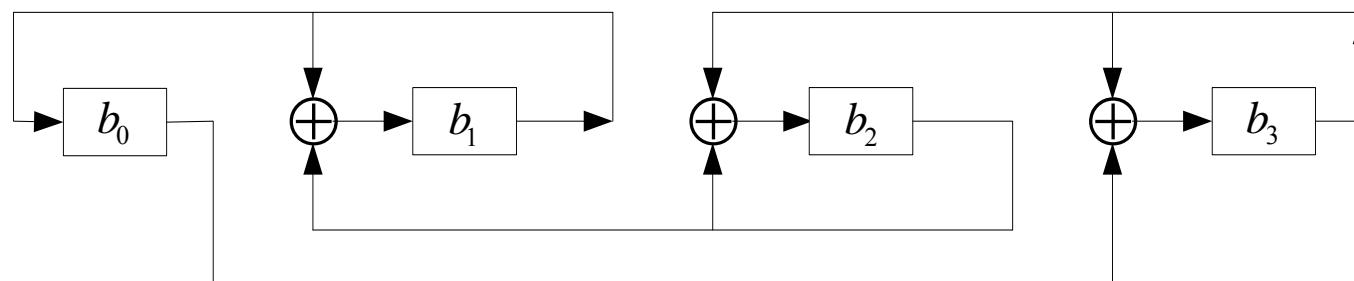
$$\beta = b_0 + b_1\alpha + b_2\alpha^2 + b_3\alpha^3$$

$$\alpha^3\beta = b_0\alpha^3 + b_1\alpha^4 + b_2\alpha^5 + b_3\alpha^5$$

$$= b_0\alpha^3 + b_1(1+\alpha) + b_2(\alpha + \alpha^2)$$

$$+ b_3(\alpha^2 + \alpha^3)$$

$$= b_1 + (b_1 + b_2)\alpha + (b_2 + b_3)\alpha^2 + (b_0 + b_3)\alpha^3$$



- ✿ Ex3:
multiplying two arbitrary elements

$$\beta = b_0 + b_1\alpha + b_2\alpha^2 + b_3\alpha^3$$

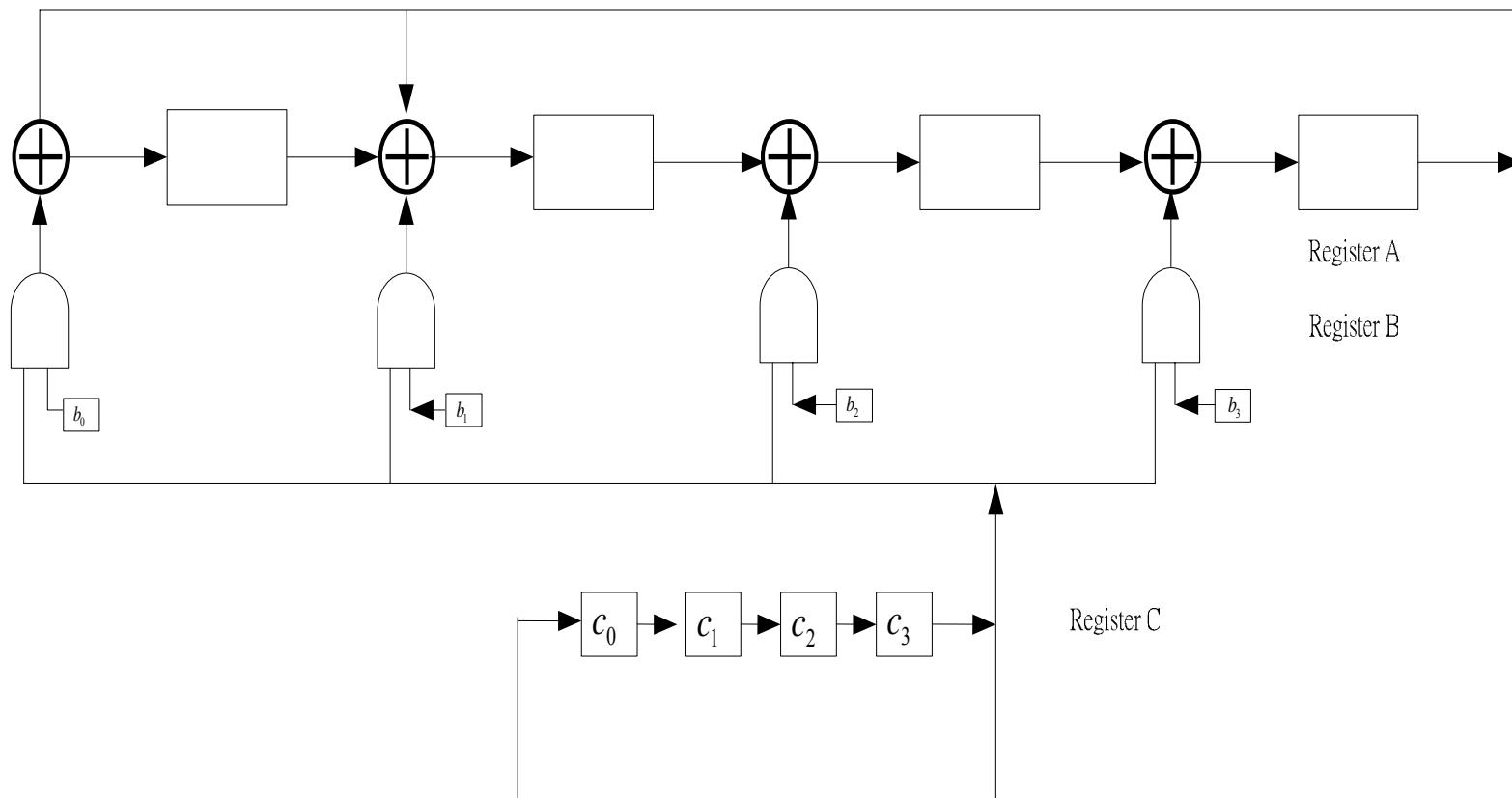
$$r = c_0 + c_1\alpha + c_2\alpha^2 + c_3\alpha^3$$

$$\begin{aligned}\beta r &= c_0\beta + c_1\beta\alpha + c_2\beta\alpha^2 + c_3\beta\alpha^3 \\ &= (((c_3\beta)\alpha + c_2\beta)\alpha + c_1\beta)\alpha + c_0\beta\end{aligned}$$

Steps:

1. multiply $c_3\beta$ by α and add the product to $c_2\beta$
 2. multiply $(c_3\beta)\alpha + c_2\beta$ by α and add the product to $c_1\beta$
 3. multiply $((c_3\beta)\alpha + c_2\beta)\alpha + c_1\beta$ by α and add the product to $c_0\beta$
-

- The circuit in ex2 can be modified to carry out the computation given by ex3



Step 1: Register A are empty

$$B \leftarrow (b_0 \ b_1 \ b_2 \ b_3) \quad C \leftarrow (c_0 \ c_1 \ c_2 \ c_3)$$

Step 2: Registers A. C. are shifted four times.

At the end of first shift, $A \leftarrow ((c_3b_0, c_3b_1, c_3b_2, c_3b_3)$, the vector representation of $c_3\beta$

At the end of second shift, $A \leftarrow ((c_3\beta)\alpha + c_2\beta)$

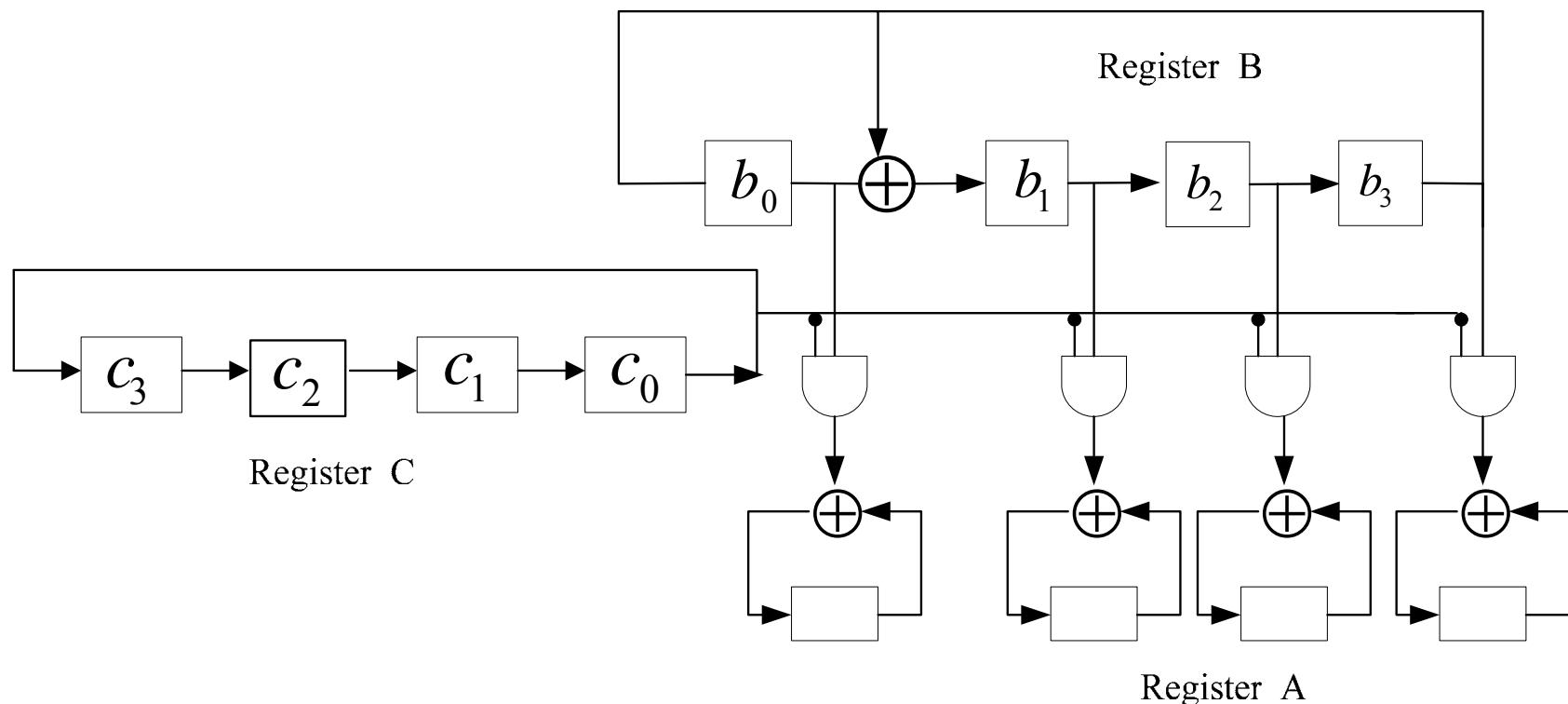
At the end of third shift, $A \leftarrow (((c_3\beta)\alpha + c_2\beta)\alpha + c_1\beta)$

At the end of fourth shift, $A \leftarrow \beta\gamma$

- If we express

$$\beta\gamma = (((c_0\beta) + c_1\beta\alpha) + c_2\beta\alpha^2) + c_3\beta\alpha^3$$

then we obtain a different multiplication circuit.



- ✿ Multiplication can be implemented by combinational logic circuit with $2m$ inputs and m outputs.
 - ✿ advantage → speed
 - ✿ disadvantage → $m > 7$, prohibitively complex and costly.

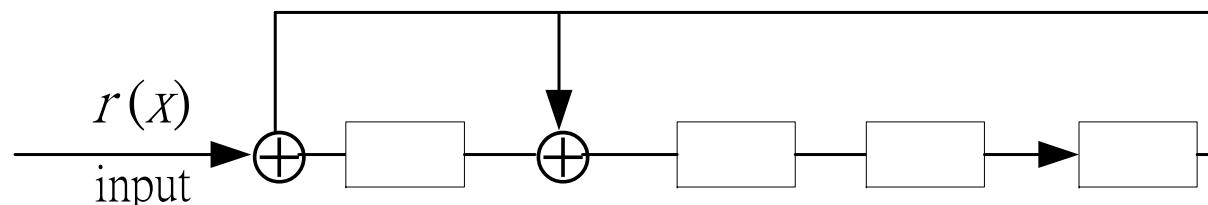
Programmed in a general-purpose computer will require roughly $5m$ instruction executions.

- ✿ How to compute syndrome $s_i = r(\alpha^i)$?
- It can be done with a circuit for multiplying a field element by α^i . Suppose that we wan to compute

$$\begin{aligned}
 r(\alpha) &= r_0 + r_1\alpha + r_2\alpha^2 + \dots + r_{14}\alpha^{14} \\
 &= (\dots(((r_{14})\alpha + r_{13})\alpha + r_{12})\alpha + \dots)\alpha + r_0
 \end{aligned}$$

where α is a primitive element in $\text{GF}(2^4)$

- The computation of $r(\alpha)$ can be accomplished by adding an input to the circuit

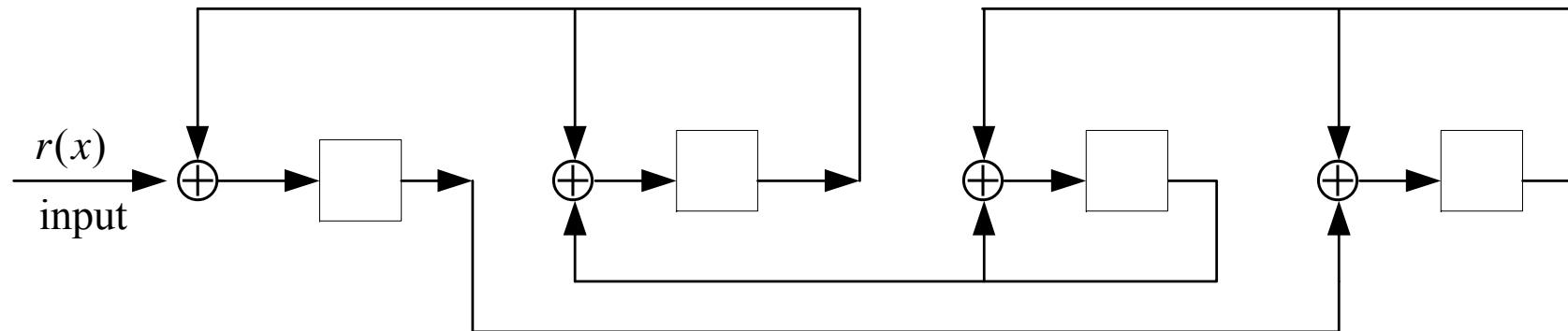


At the end of first shift $(r_{14}, 0, 0, 0)$

At the end of 2nd shift $r_{14}\alpha + r_{13}$

At the end of 3rd shift $(r_{14}\alpha + r_{13})\alpha + r_{12}$

- We can compute $r(\alpha^3)$ by adding an input to the circuit for multiplying by α^3 of figure in Ex2.



- There is another way of computing $r(\alpha^i)$. Let $\phi_i(x)$ be the minimal poly. of α^i . Let $b(x)$ be the remainder resulting from dividing $r(x)$ by $\phi_i(x)$. Then

$$r(\alpha^i) = b(\alpha^i)$$

Thus, computing $\mathbf{r}(\alpha^i)$ is equivalent to computing $\mathbf{b}(\alpha^i)$. Again we consider computation over $\text{GF}(2^4)$. Suppose that we want to compute $\mathbf{r}(\alpha^3)$. The minimal poly. of α^3 is

$$\phi_3(x) = 1 + x + x^2 + x^3 + x^4$$

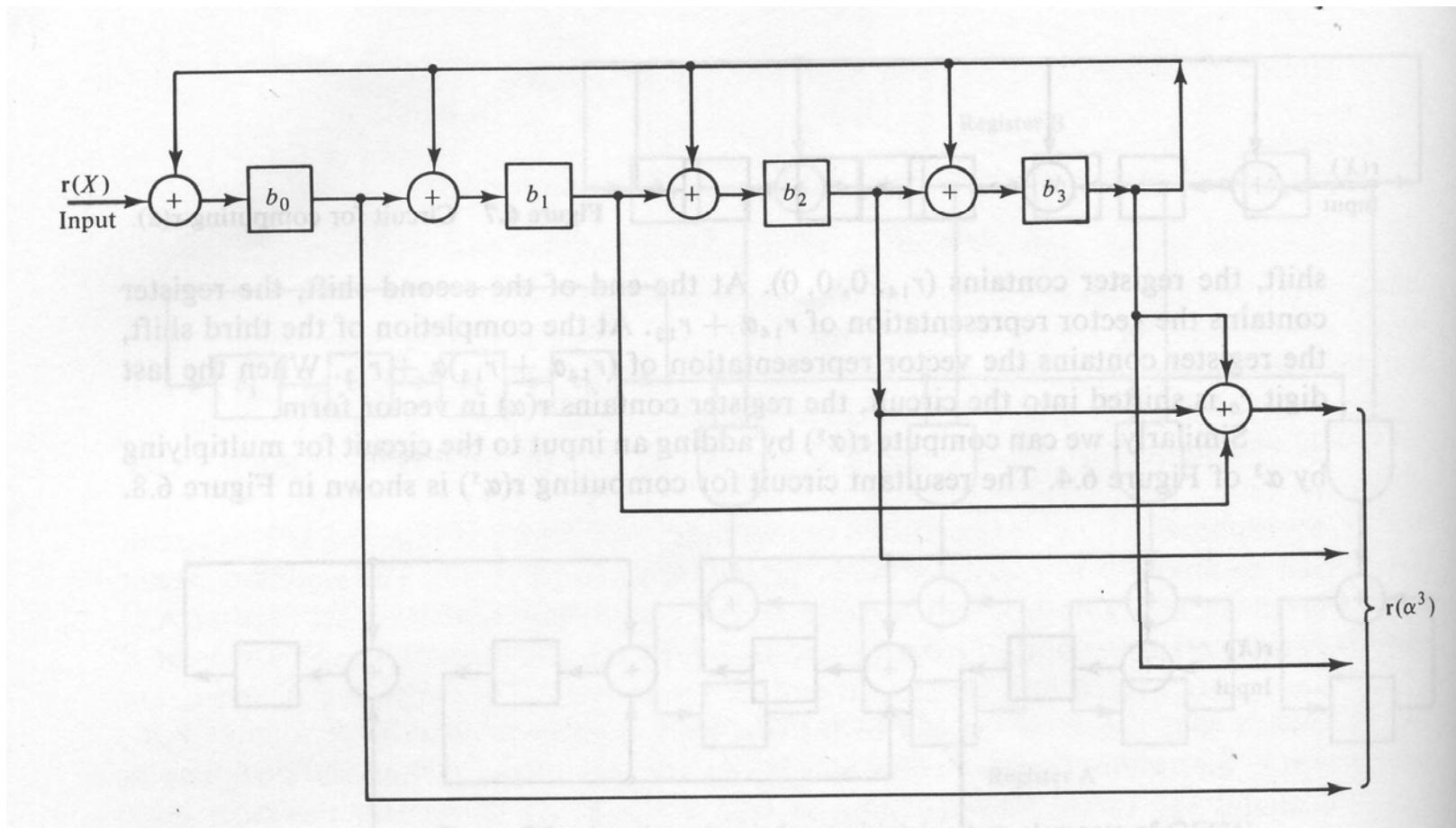
The remainder

$$\mathbf{b}(x) = b_0 + b_1x + b_2x^2 + b_3x^3$$

Then

$$\begin{aligned}
 & \mathbf{b}(\alpha^3) \\
 &= b_0 + b_1\alpha^3 + b_2\alpha^6 + b_3\alpha^9 \\
 &= b_0 + b_1\alpha^3 + b_2(\alpha + \alpha^3) + b_3(\alpha + \alpha^3) \\
 &= b_0 + b_3\alpha + b_2\alpha^2 + (b_1 + b_2 + b_3)\alpha^3
 \end{aligned}$$

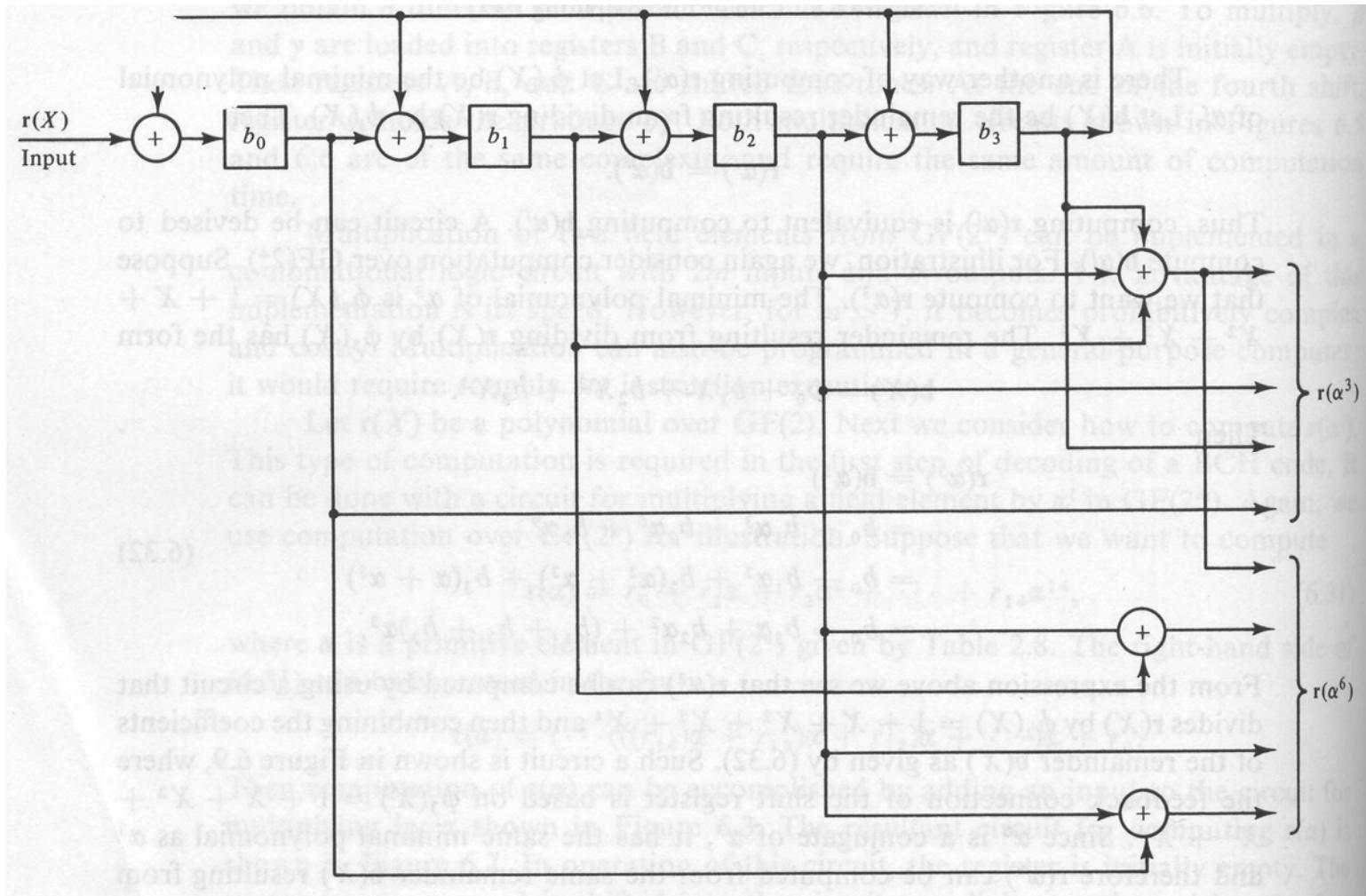
Such a circuit is shown below



Since α^6 is a conjugate of α^3 , it has the same poly. as α^3

$$\begin{aligned} b(\alpha^6) &= b_0 + b_1\alpha^6 + b_2\alpha^{12} + b_3\alpha^{18} \\ &= b_0 + b_1(\alpha^2 + \alpha^3) + b_2(1 + \alpha + \alpha^2 + \alpha^3) + b_3\alpha^3 \\ &= (b_0 + b_2) + b_2\alpha + (b_1 + b_2)\alpha^2 + (b_1 + b_2 + b_3)\alpha^3 \end{aligned}$$

Its circuit is shown in the next page



Wireless Information Transmission System Lab.

Implementation of Error Correction



National Sun Yat-sen University

■ Syndrome Computation

The first step in decoding t -error-correction BCH code is to compute the $2t$ syndrome component S_1, S_2, \dots, S_{2t} .

$$\begin{aligned}
 \text{Software: } S_i &= r(\alpha^i) = r_{n-1}(\alpha^i)^{n-1} + r_{n-2}(\alpha^i)^{n-2} \\
 &\quad + \dots + r_1\alpha^i + r_0 \\
 &= (\dots((r_{n-1}\alpha^i + r_{n-2})\alpha^i + r_{n-3})\alpha^i \\
 &\quad + \dots + r_1)\alpha^i + r_0
 \end{aligned}$$

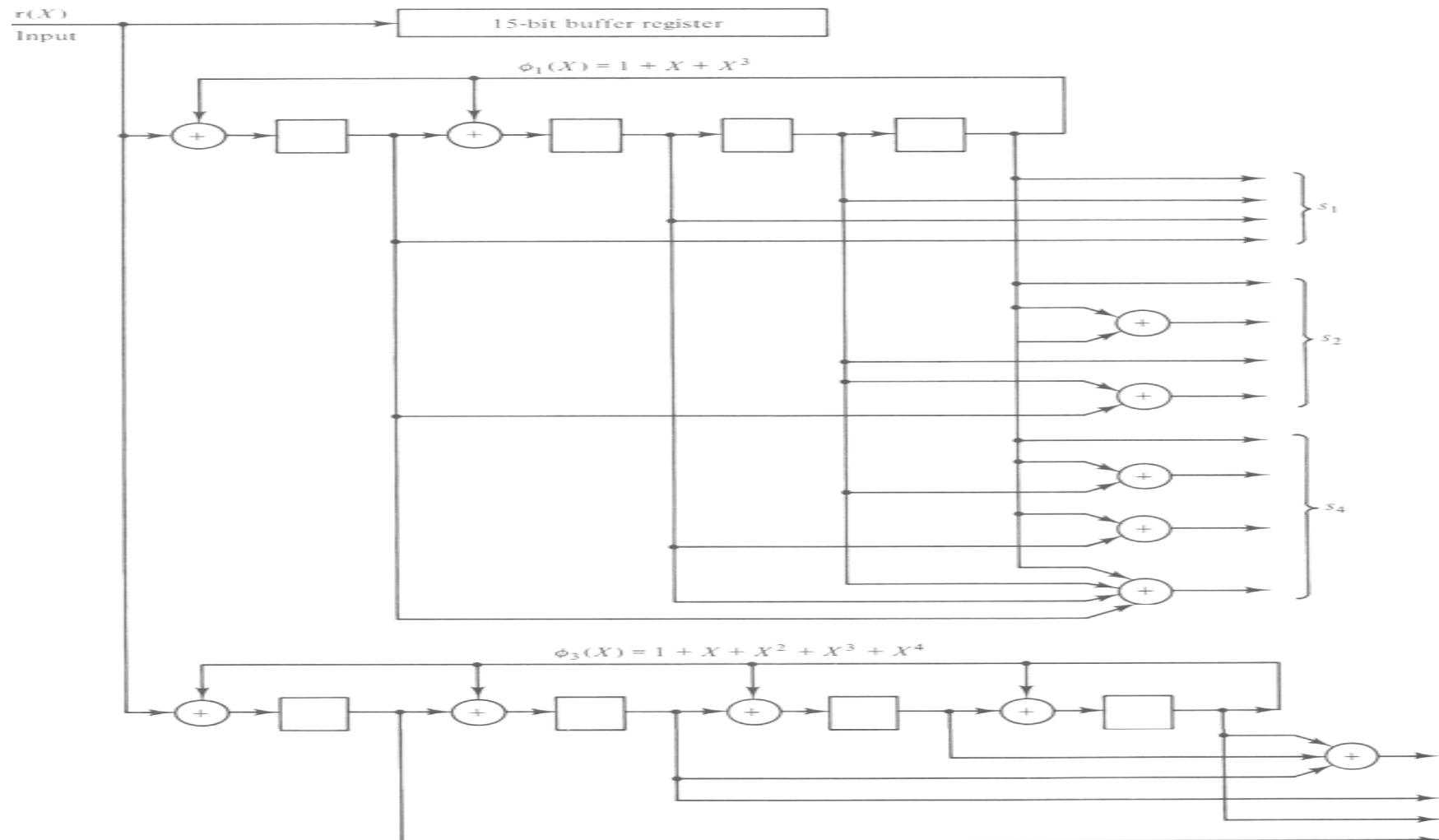
$(n-1)$ addition and $(n-1)$ multiplication

For binary $S_{2i} = S_i^2$: $2t$ syndrome components can be computed with $(n-1)t$ additions and $nt = (n-1)t + t$ multiplications.

- For hardware implementation, a variety of types of circuits can be adopted, e.g. Figure 6.7 and 6.8, or Figure 6.10.

- Alternatively, as shown in Figure 6.11 (the following page), at most t feedback shift registers, each consisting of at most m stages, are needed to form the $2t$ syndrome components because the generator polynomial is a product of at most t minimal polynomials.

- Syndrome computation circuit for the double-error-correcting (15,7) BCH code



- As soon as the entire $r(x)$ has entered the decoder, the $2t$ syndrome components are formed.
- It takes n clock cycles to complete the computation.

- Finding the Error-Location Polynomial $\sigma(X)$
Software computation requires somewhat less than t additions and t multiplications to compute each $\sigma^{(\mu)}(X)$ and each d_μ , and since there are t of each, the total is roughly $2t^2$ additions and $2t^2$ multiplications.

hardware: requires the same total, and the speed would depend on how much is done in parallel.

- Computation of Error-location Numbers and Error Correction

Worst case: requires substituting n field elements into an error-location poly. $\sigma(x)$ of degree t to determine roots.

Software : requiring nt multiplications and nt additions.

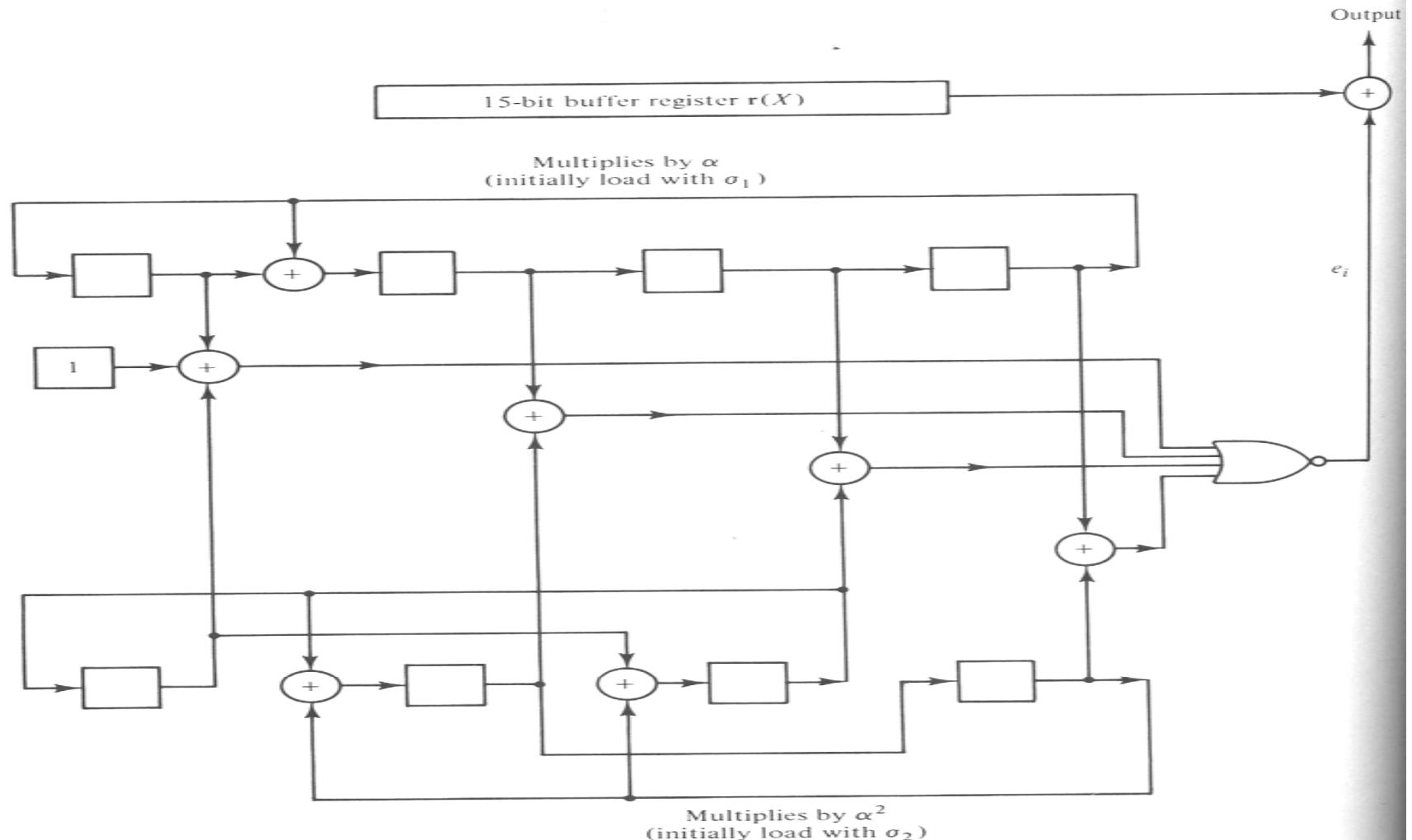
Hardware : using Chien's searching circuit. See Figure 6.1.

It requires t multipliers for multiplying by $\alpha, \alpha^2, \dots, \alpha^t$ respectively.

At the end of the ℓ th shift, If

$$\text{sum} = 1 + \sigma_1 \alpha^l + \sigma_2 \alpha^{2l} + \dots + \sigma_t \alpha^{tl}$$

$\text{sum}=0$, α^{n-l} is an error-location number. The sum can be formed by using t m -input modulo-2 adders. A m -input OR gate is used to test whether the sum is zero. It takes n clock cycles to complete this step. If we only to correct the message digits, only k clock cycles are needed. A Chien's searching circuit for (15, 7, 2) BCH code is shown next slide.



Wireless Information Transmission System Lab.

Nonbinary BCH Codes and Reed-Solomon Codes



National Sun Yat-sen University

Nonbinary BCH Codes and Reed-Solomon Codes

- In addition to the binary codes, there are nonbinary codes.
- If p is a prime number and q is any power of p ($q=p^m$), there are codes with symbols from the Galois field $\text{GF}(q)$. These codes are called *q-ary codes*.
- An (n, k) linear code with symbols from $\text{GF}(q)$ is a k -dimensional subspace of the vector space of all n -tuples over $\text{GF}(q)$.
- A q -ary (n, k) cyclic code is generated by a polynomial of degree $(n-k)$ with coefficients from $\text{GF}(q)$, which is a factor of X^n-1 .
- Encoding and decoding of q -ary codes are similar to that of binary codes.
- For any choice of positive integers s and t , there exists a q -ary BCH code of length $n=q^s-1$, which can correct any combinational of t or fewer errors and requires no more than $2st$ parity-check digits.

Nonbinary BCH Codes and Reed-Solomon Codes

- ✿ Let α be a primitive element in the Galois field $\text{GF}(q^s)$.
- ✿ The generator polynomial $g(X)$ of a t -error-correcting q -ary BCH is the polynomial of lowest degree with coefficients from $\text{GF}(q)$ for which $\alpha, \alpha^2, \dots, \alpha^{2t}$ are roots.
- ✿ Let $\phi_i(X)$ be the minimal polynomial of α^i . Then

$$g(X) = \text{LCM}\{\phi_1(X), \phi_2(X), \dots, \phi_{2t}(X)\}$$

- ✿ The degree of each minimal polynomial is s or less.
- ✿ Therefore, the degree of $g(X)$ is at most $2st$, and hence the number of parity-check digits of the code generated by $g(X)$ is no more than $2st$.
- ✿ For $q=2$, we obtain the binary BCH codes.

Nonbinary BCH Codes and Reed-Solomon Codes

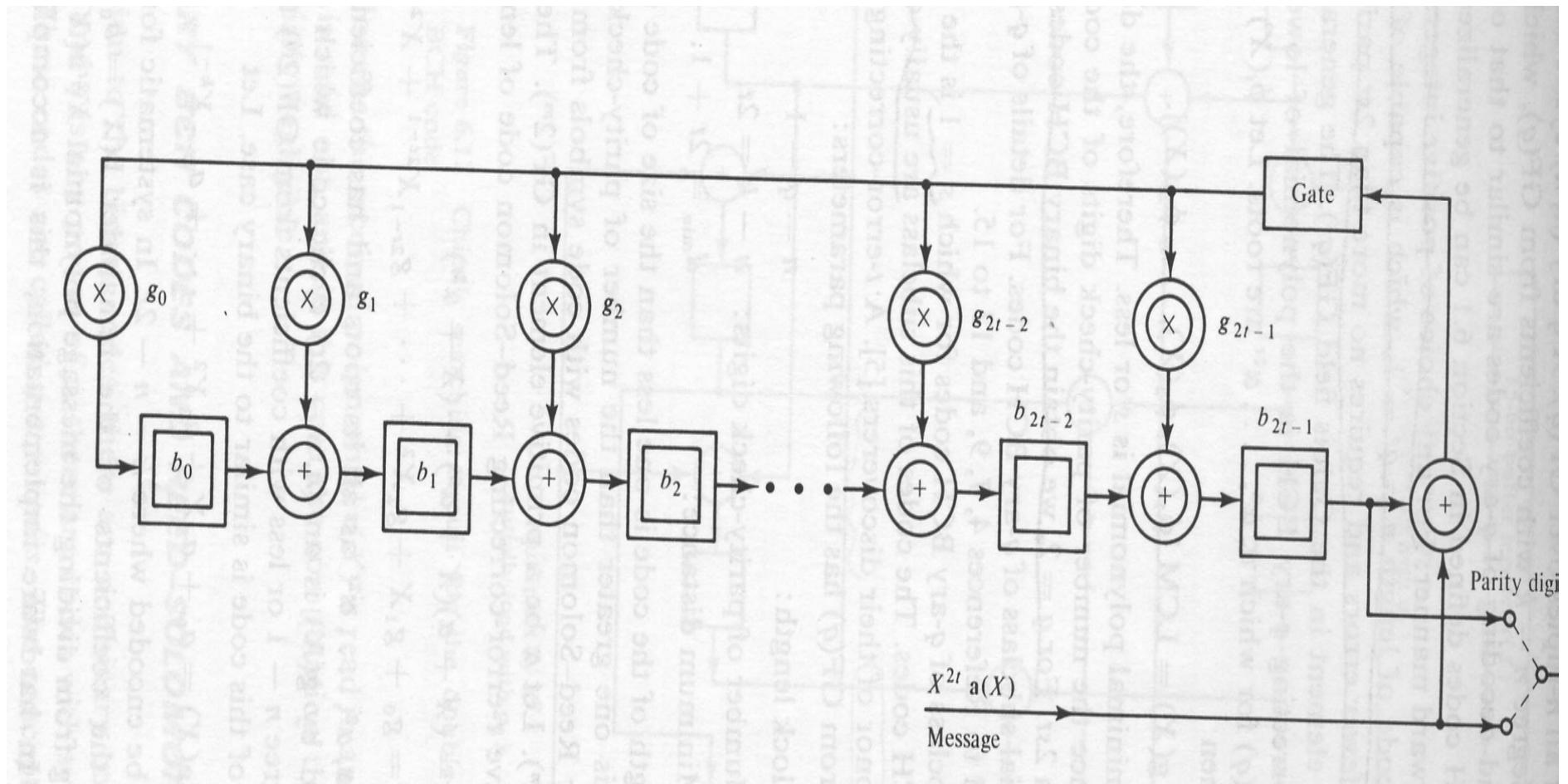
- t -error-correcting Reed-Solomon code with symbols from $\text{GF}(q)$:
 - $s=1$, Block length: $n=q-1$, # parity-check digits: $n-k=2t$,
Minimum distance: $d_{\min}=2t+1$.
- Reed-Solomon codes with code symbols from $\text{GF}(2^m)$ (i.e., $q=2^m$)
 - Let α be a primitive element in $\text{GF}(2^m)$
 - The generator polynomial of code length 2^m-1 is

$$g(x) = (x + \alpha)(x + \alpha^2) \dots (x + \alpha^{2t}) = g_0 + g_1 x + \dots + g_{2t-1} x^{2t-1} + x^{2t}$$
 - $(n, n-2t)$ cyclic code and coefficients of $g(X)$ is from $\text{GF}(2^m)$
- Let Message: $\mathbf{a}(X) = a_0 + a_1 X + a_2 X^2 + \dots + a_{k-1} X^{k-1}$
- In systematic form, the $2t$ parity-check digits are coefficients of

$$\mathbf{b}(X) = R_{g(X)} [X^{2t} \mathbf{a}(X)] = b_0 + b_1 X + \dots + b_{2t-1} X^{2t-1} \quad (\text{Sec 4.3})$$

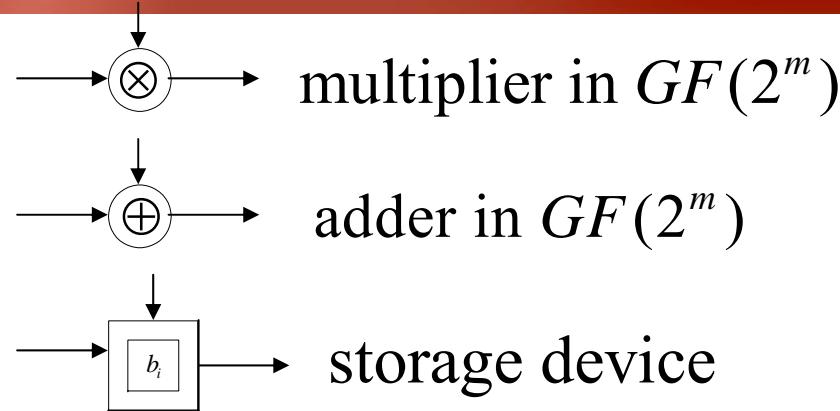
Nonbinary BCH Codes and Reed-Solomon Codes

- Hardware implementation



Nonbinary BCH Codes and Reed-Solomon Codes

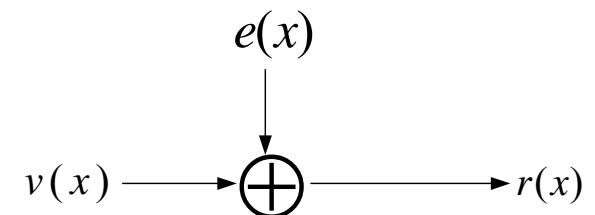
Where



- Decoding of RS code:

$$\mathbf{v}(x) = v_0 + v_1 x + \dots + v_{n-1} x^{n-1}$$

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



$$e(x) = r(x) - v(x) = e_0 + e_1 x + \dots + e_{n-1} x^{n-1}$$

$$e_i = r_i - v_i, \quad \in GF(2^m)$$

Nonbinary BCH Codes and Reed-Solomon Codes

- Suppose the error pattern $\mathbf{e}(x)$ contains v errors (nonzero components), then

$$\mathbf{e}(x) = e_{j_1} x^{j_1} + e_{j_2} x^{j_2} + \dots + e_{j_v} x^{j_v}$$

To determine $\mathbf{e}(x)$, we need error-locations $x^{j_l}'s$ and the error values $e_{j_l}'s$ (i.e. v pairs $(x^{j_l}, e_{j_l})'s$)

- In decoding a RS code, the same three steps used for decoding a binary BCH code are required ; in addition, a fourth step involving calculation of error values is required.

Nonbinary BCH Codes and Reed-Solomon Codes

Let $\beta_\ell = \alpha^{j_\ell} \quad \ell = 1, 2, \dots, v$

$$s_1 = r(\alpha) = e_{j1}\beta_1 + e_{j2}\beta_2 + \dots + e_{jv}\beta_v$$

$$s_2 = r(\alpha^2) = e_{j1}\beta_1^2 + \dots + e_{jv}\beta_v^2$$

$$\begin{matrix} \vdots & \vdots \end{matrix}$$

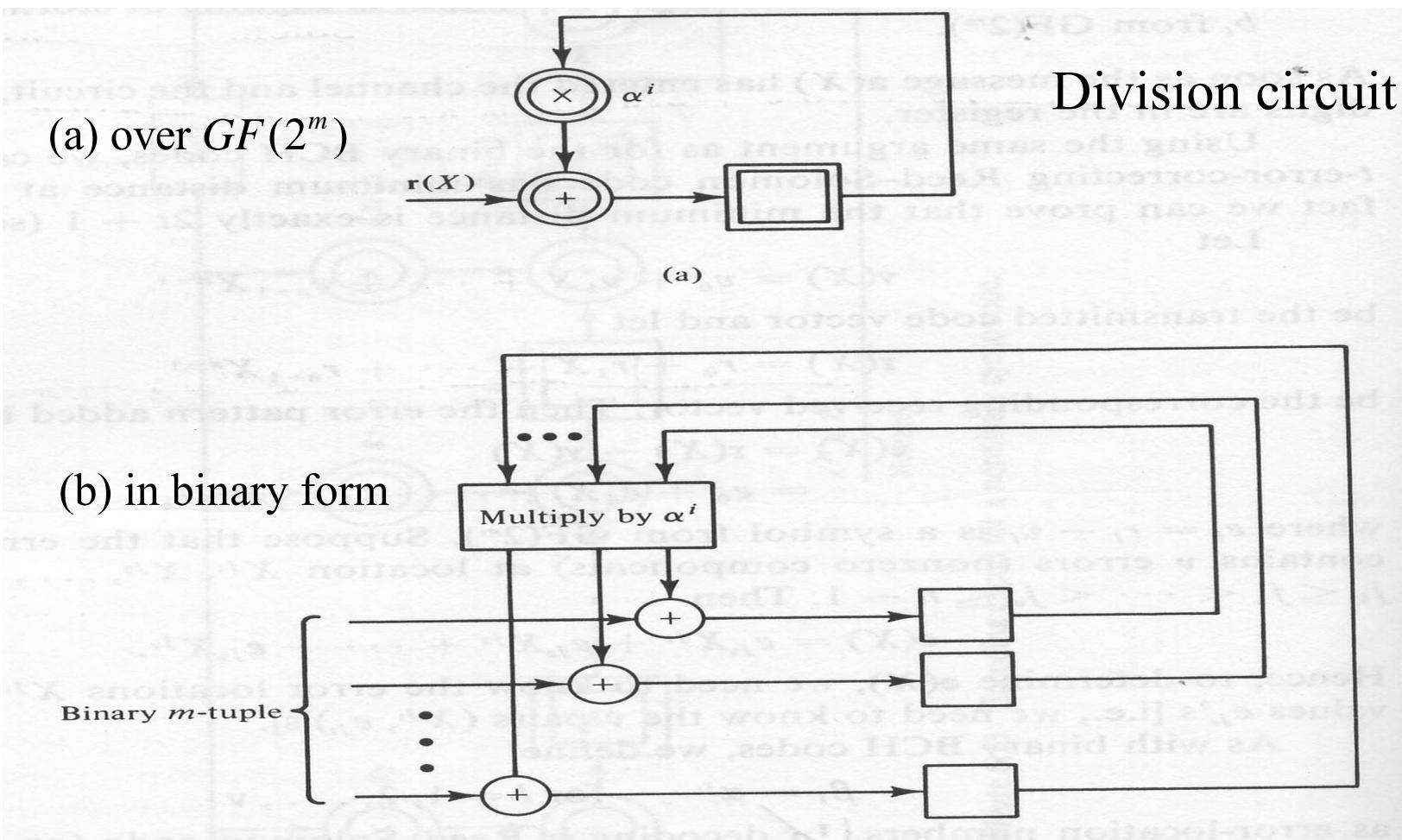
$$s_{2t} = r(\alpha^{2t}) = e_{j1}\beta_1^{2t} + \dots + e_{jv}\beta_v^{2t}$$

note: $S_i = R_{(x+\alpha^i)}[r(x)] = b_i$

(i.e. $r(x) = e_i(x)(x + \alpha^i) + b_i$)

Nonbinary BCH Codes and Reed-Solomon Codes

- ◆ Syndrome computation circuit for R-S codes



Nonbinary BCH Codes and Reed-Solomon Codes

- To find $\sigma(x)$: the error-location poly. by Berlekamp's iterative algorithm

$$\begin{aligned}\sigma(x) &= (1 + \beta_1 x)(1 + \beta_2 x) \dots (1 + \beta_v x) \\ &= 1 + \sigma_1 x + \dots + \sigma_v x^v\end{aligned}$$

Once $\sigma(x)$ is found, we can determine the error values.

Let

$$z(x) = 1 + (s_1 + \sigma_1)x + (s_2 + \sigma_1 s_1 + \sigma_2)x^2 + (s_v + \sigma_1 s_{v-1} + \dots + \sigma_v)x^v$$

- It can be shown that the error value at location $\beta_l = \alpha^{j_l}$ is given by:

$$e_{j_l} = \frac{Z(\beta_l^{-1})}{\prod_{\substack{i=1 \\ i \neq l}}^v (1 + \beta_i \beta_l^{-1})}$$

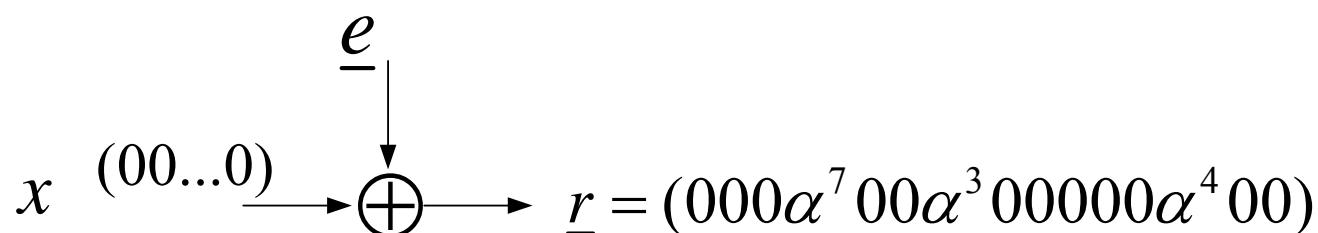
Nonbinary BCH Codes and Reed-Solomon Codes

- Ex: Consider a triple-error-correcting Reed-Solomon code with symbols from $\text{GF}(2^4)$

- The generator polynomial of this code is:

$$g(x) = (x + \alpha)(x + \alpha^2)\dots(x + \alpha^6) \\ = \alpha^6 + \alpha^9 x + \alpha^6 x^2 + \alpha^4 x^3 + \alpha^{14} x^4 + \alpha^{10} x^5 + x^6$$

- (15, 9, 3) RS code, $n = 15$. $n - k = 6$, $t = 3$



$$\therefore r(x) = \alpha^7 x^3 + \alpha^3 x^6 + \alpha^4 x^{12}$$

Nonbinary BCH Codes and Reed-Solomon Codes

- Step 1. Using table 2.8 below to compute syndrome components:

TABLE 2.8 THREE REPRESENTATIONS FOR THE ELEMENTS OF GF(2⁴) GENERATED BY $p(X) = 1 + X + X^4$

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

Nonbinary BCH Codes and Reed-Solomon Codes

$$S_1 = r(\alpha) = \alpha^{10} + \alpha^9 + \alpha = \alpha^{12}$$

$$S_2 = r(\alpha^2) = \alpha^{13} + 1 + \alpha^{13} = 1$$

$$S_3 = r(\alpha^3) = \alpha + \alpha^6 + \alpha^{10} = \alpha^{14}$$

$$S_4 = r(\alpha^4) = \alpha^4 + \alpha^{12} + \alpha^7 = \alpha^{10}$$

$$S_5 = r(\alpha^5) = \alpha^7 + \alpha^3 + \alpha^4 = 0$$

$$S_6 = r(\alpha^6) = \alpha^{10} + \alpha^9 + \alpha = \alpha^{12}$$

- Step 2. Find $\sigma(x) = 1 + \alpha^7x + \alpha^4x^2 + \alpha^6x^3$ by filling out table in next slide.

Nonbinary BCH Codes and Reed-Solomon Codes

μ	$\sigma^{(\mu)}(X)$	d_μ	l_μ	$\mu - l_\mu$
-1	1	1	0	-1
0	1	α^{12}	0	0
1	$1 + \alpha^{12}X$	α^7	1	0 (take $\rho = -1$)
2	$1 + \alpha^3X$	1	1	1 (take $\rho = 0$)
3	$1 + \alpha^3X + \alpha^3X^2$	α^7	2	1 (take $\rho = 0$)
4	$1 + \alpha^4X + \alpha^{12}X^2$	α^{10}	2	2 (take $\rho = 2$)
5	$1 + \alpha^7X + \alpha^4X^2 + \alpha^6X^3$	0	3	2 (take $\rho = 3$)
6	$1 + \alpha^7X + \alpha^4X^2 + \alpha^6X^3$	—	—	—

Nonbinary BCH Codes and Reed-Solomon Codes

- Step 3. By substituting $1, \alpha, \alpha^2, \dots, \alpha^{14}$ into $\sigma(x)$

$\Rightarrow \alpha^3, \alpha^9, \alpha^{12}$ are roots of $\sigma(x)$

$\therefore \alpha^{-3} = \alpha^{12}, \alpha^{-9} = \alpha^6, \alpha^{-12} = \alpha^3$ are the error-location numbers of $e(x)$ $\therefore e(x) = e_3x^3 + e_6x^6 + e_{12}x^{12}$

- Step 4. From 6.34, find $z(x) = 1 + \alpha^2x + x^2 + \alpha^6x^3$

$$\therefore e_3 = \frac{z(\alpha^{-3})}{(1 + \alpha^6\alpha^{-3})(1 + \alpha^{12}\alpha^{-3})} = \alpha^7$$

$$e_6 = \frac{z(\alpha^{-6})}{(1 + \alpha^3\alpha^{-6})(1 + \alpha^{12}\alpha^{-6})} = \alpha^3$$

Nonbinary BCH Codes and Reed-Solomon Codes

$$e_{12} = \frac{z(\alpha^{-12})}{(1 + \alpha^3\alpha^{-12})(1 + \alpha^6\alpha^{-12})} = \alpha^4$$

$$\therefore e(x) = \alpha^7x^3 + \alpha^3x^6 + \alpha^4x^{12}$$

$$\therefore v(x) = r(x) - e(x) = \underline{0}$$

- If β is not a primitive element of $\text{GF}(2^m)$, then the 2^m -ary code generated by

$$g(x) = (x + \beta)(x + \beta^2)\dots(x + \beta^{2t})$$
 is a nonprimitive t -error-correcting RS code.
- The length n of this code is simply the order of β .
- Decoding of a nonprimitive Reed-Solomon code is identical to the decoding of a primitive Reed-Solomon code.

Nonbinary BCH Codes and Reed-Solomon Codes

- ✿ Reed-Solomon codes are very effective for correcting multiple bursts of errors.
- ✿ Two information symbols can be added to a RS code of length n without reducing its minimum distance.
- ✿ The extended RS code has length $n+2$ and the same number of parity-check symbols as the original code.
- ✿ For a t -error-correcting RS code, the parity-check matrix may take the form

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

- The parity-check matrix of the extended RS code is

$$H_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \\ \cdot & \cdot \\ \cdot & \cdot \\ \cdot & \cdot \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \end{bmatrix} H$$