ELEC 6131: Error Detecting and Correcting Codes
Lecture 8: Reed-Solomon (RS) Codes

RS Codes are a sub-class of non-binary BCH Codes. In a non-binary code, codewords consist of symbols which are each $m \geq 2$ bits long.
In general, non-binary codes can be defined over any Galois Field GF(q) where q is either a prime or a power of a prime. However, for obvious reasons, people are most interested in codes defined over $GF(2^m)$.
For Reed-Solomon Codes take some integer m. Then each symbol is m bits long. This means that symbols belong to $\{0,1, \ldots, 2^m\}$.
An (N, K) RS code consists of N symbols each of which is m bits long and has K information symbols and N-K parity symbols.
For an RS code over $GF(2^m)$ we have $N = 2^m - 1$.
K can be any value less than N.
An (N, K) RS code has the minimum distance $d_{min} = N - K + 1$.
It can correct $t = \left[\frac{d_{min}-1}{2}\right] = \left[\frac{N-K}{2}\right]$
The reason I used N and K instead of n and k was to differentiate between an (n, k) binary code that has codewords that are n bits long and have k information bits and non-binary codes with N and K symbols.
I hope we have so far have got used to the idea of symbols other than a single bit. So, from this point on, I will use n and k.
(n, k) RS code over $GF(2^m)$ has codeword of length n symbols, i.e., $n * m$ bits out of which $k * m$ are information (or systematic) bits.
For example a (255,239) RS Code over $GF(2^8)$ has codewords each 255 bytes and each codeword has 239 bytes of information on (n-k) = 16 bytes of parity. Such a code can correct up to $\frac{16}{2} = 8$ bytes of errors.
Note that here when we correct one symbol, we may have corrected $1,2,..,m$ bits. If we have a burst of errors, that is a lot of errors near. One another, RS Codes can be very useful. An RS Code which can correct t error symbols can correct $(t-1)m$ bits long bursts.
The generating polynomial of t error correcting RS Code is:

$$g(x) = (x + \alpha)(x + \alpha^2) \ldots (x + \alpha^{2t})$$
$$= g_0 + g_1 x + g_2 x^2 + \cdots + g_{2t-1} x^{2t-1} + x^{2t}$$

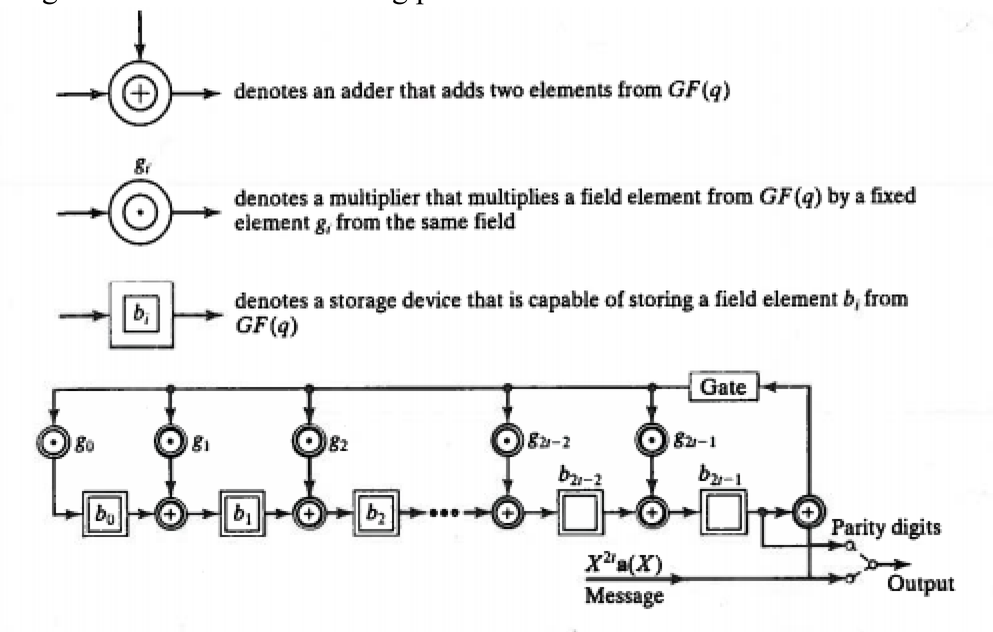
With $g_i \in GF(2^m)$ for $0 \leq i \leq 2t$.
$\alpha, \alpha^2, \ldots \alpha^{2t}$ are roots of X^n+1. G(x) divides X^n+1. So, g(x) generates a $2^m - ry$ cyclic code of length n with 2t parity symbols.

Encoding of RS Codes:

We can simply multiply the information polynomial u(x) by g(x). However, this may not result in a systematic code to make the code systematic, we multiply u(x) by $X^{n-k}$to get X^{n-k}u(x) which we divide by g(x) to get:

$$X^{n-k} u(x) = q(x)g(x) + b(x)$$

$q(x)g(x)$ is a code polynomial. Also we have:
V(x)= $q(x)g(x) = x^{n-k}u(x) + b(x)$
This means that we have u(x) as part of v(x), i.e., the code is systematic and b(x) is the parities polynomial.
The following circuit shows the encoding procedure:



Encoding circuit for a q-ary RS code with generator polynomial $\mathbf{g}(X) = g_0 + g_1X + g_2X^2 + \cdots + g_{2t-1}X^{2t-1} + X^{2t}$.

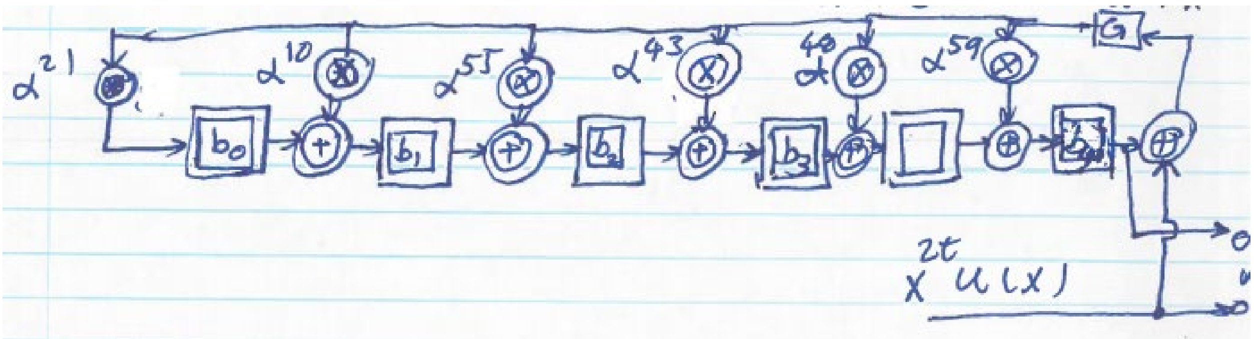
1) First we close the gate and feed the information symbols into the division circuit. At the same time these information symbols are put on the line (to be transmitted): switch in lower position.
2) After feeding all k symbols, we open the gate (disconnect the feedback) and put switch in the up position, transmitting 2t parity symbols.

Example:

Find the generating polynomial of triple error correcting code over $GF(2^6)$.

$$g(x) = (x + \alpha)(x + \alpha^2)(x + \alpha^3)(x + \alpha^4)(x + \alpha^5)(x + \alpha^6)$$

$$= \alpha^{21} + \alpha^{10}x + \alpha^{55}x^2 + \alpha^{43}x^3 + \alpha^{48}x^4 + \alpha^{59}x^5 + x^6$$



The Parity-Check matrix of an RS code is given as:

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

Decoding of RS Codes:

1) Find syndrome.
2) Find error-location polynomial.
3) Find error-value evaluator.
4) Find the error locations and error values and correct.

Assume that the codeword $\underline{v} = (v_0, v_1, \dots, v_{n-1})$ is transmitted or equivalently

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

Assume that r(x) is received:

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

r(x)=v(x)+e(x) where e(x) is the error polynomial

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

Assume we have errors at locations

$$j_1, j_2, \cdots, j_\gamma$$

Denote the values of error by $e_{j_1}, e_{j_2}, \dots, e_{j_\gamma}$

Then:

$$e_i = \begin{cases} 0 & i \neq j_1, \dots, j_\gamma \\ e_{j_e} \; if\ i = j_e \epsilon \{j_1, \dots, j_e\} \end{cases}$$

So, we can write:

$$e(x) = e_{j_1} x^{j_1} + e_{j_2} x^{j_2} + \cdots + e_{j_\gamma} x^{j_\gamma}$$

So what we need to do is to find $j_1, \dots, j_\gamma$ as well as $e_{j_1}, \dots, e_{j_\gamma}$.

That is why we have 2γ unknowns.

Remember that

$$V\left(\alpha^i\right) = 0 \qquad i = 1,2,\dots,2t$$

$$r(\alpha^i) = v(\alpha^i) + e(\alpha^i) = s_i$$

So,

$$S_i = r(\alpha^i) = e(\alpha^i)$$

That is we substitute $\alpha^i, \quad i = 1,2,\dots,2t$ in r(x) to get 2t syndromes. These provide 2t equations with j_i's and e_{j_i} ‘s as their components. In order to be able to solve for the 2γ unknowns, we need to have 2γ equations, i.e., $2t = 2\gamma \rightarrow t = \gamma$. That is a proof that RS Code can correct t errors.

Now let’s expand $S_i = e(\alpha^i) \quad 's$:

$s_1 = e_{j_1}\alpha^{j_1} + e_{j_2}\alpha^{j_2} + \dots + e_{j_\gamma}\alpha^{j_\gamma}$

$s_2 = e_{j_1}\alpha^{2j_1} + e_{j_2}\alpha^{2j_2} + \dots + e_{j_\gamma}\alpha^{2j_\gamma}$

$\vdots$

$s_{2t} = e_{j_1}\alpha^{2tj_1} + e_{j_2}\alpha^{2tj_2} + \dots + e_{j_\gamma}\alpha^{2tj_\gamma}$

Let $B_i \triangleq \alpha^{j_i}$ and $S_i \triangleq e_{j_i}$ For $1 \leq i \leq \gamma$

Then:

$s_1 = s_1\beta_1 + s_2\beta_2 + \dots + s_\gamma\beta_\gamma$

$S_2 = s_1{\beta_1}^2 + s_2{\beta_2}^2 + \dots + s_\gamma\beta_\gamma^2$

$\vdots$

$s_{2t} = s_1\beta_1^{2t} + s_2\beta_2^{2t} + \dots + s_\gamma\beta_\gamma^{2t}$

Define the error location polynomial:

$$\sigma(x) = (1 + \beta_1 x)(1 + \beta_2 x) \dots (1 + \beta_\gamma x)$$

$$=\sigma_0 + \sigma_1 x + \sigma_2 x^2 + \dots + \sigma_\gamma x^\gamma$$

We can see that

$\sigma_0 = 1$

$\sigma_1 = \beta_1 + \beta_2 + \dots + \beta_\gamma = s_1$

$\sigma_2 = \beta_1\beta_2 + \cdots + \beta_{\gamma-1}\beta_\gamma = \sigma_1 s_1 + s_2$

$\vdots$

Overall, we get the following equations named Newton equalities:

$s_{\gamma+1} + \sigma_1 s_\gamma + \sigma_2 s_{\gamma-1} + \cdots + \sigma_\gamma s_1 = 0$

$s_{\gamma+2} + \sigma_1 s_{\gamma+1} + \sigma_2 s_\gamma + \cdots + \sigma_\gamma s_2 = 0$

$\vdots$

$s_{2t} + \sigma_1 s_{2t-1} + \sigma_2 s_{2t-2} + \cdots + \sigma_\gamma s_{2t-\gamma} = 0$

The same as BCH Codes, we start from σ(x)=1 in stage 0, say we call it $\sigma^{(0)}(x)$ and try to increase the number of terms so that all equations are satisfied.

Assume that at stage μ we have

$$\sigma^{(\mu)}(x) = \sigma_0^{(\mu)} + \sigma_1^{(\mu)}x + \cdots + \sigma_{L_\mu}^{(\mu)}x^{l_\mu}$$

This means that we have coefficients $\sigma_0^{(\mu)}, \sigma_1^{(\mu)}, \dots, \sigma_{L_\mu}^{(\mu)}$ of a polynomial that satisfy the first μ Newton equalities. We try to apply coefficients to μ+1-st equality, i.e., form

$$S_{\mu+1} + \sigma_1^{(\mu)}S_\mu + \cdots + \sigma_{L_\mu}^{(\mu)}S_{\mu+1-l\mu}$$

If this gives us a zero it means that $\sigma_0^{(\mu)}, \sigma_1^{(\mu)}, \dots + \sigma_{L_\mu}^{(\mu)}$ satisfy μ+1-st equality.

Otherwise we have to modify the polynomial so form:

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

If the discrepancy $d_\mu = 0$ then

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

And continue.

Otherwise:

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

Where ρ is the stage closest to μ such that $d_\rho \neq 0$

Continue this iteration until we get to stage 2t then

$$\sigma(x) = \sigma^{(2t)}(x)$$

Start by filling out the first two rows:

Berlekamp's iterative procedure for finding the error-location polynomial of a q-ary BCH code.

μ	$\sigma^{(\mu)}(X)$	d_μ	l_μ	$\mu - l_\mu$
-1	1	1	0	-1
0	1	S_1	0	0
1	$1 - S_1 X$			
2				
3				
$\vdots$				
$2t$				

Example:

Consider triple-error correcting code over $GF(2^4)$. Let $r(x) = \alpha^7 x^3 + \alpha^3 x^6 + \alpha^4 x^{12}$

Then

$$g(x) = (x+\alpha)(x+\alpha^2)(x+\alpha^3)(x+\alpha^4)(x+\alpha^5)(x+\alpha^6)$$

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

$s_1 = r(\alpha) = \alpha^{10} + \alpha^9 + \alpha = \alpha^{12}$

$s_2 = r(\alpha^2) = \alpha^{13} + 1 + \alpha^{13} = 1$

$s_3 = r(\alpha^3) = \alpha + \alpha^6 + \alpha^{10} = \alpha^{14}$

$s_4 = r(\alpha^4) = \alpha^4 + \alpha^{12} + \alpha^7 = \alpha^{10}$

$s_5 = r(\alpha^5) = \alpha^7 + \alpha^3 + \alpha^4 = 0$

$s_6 = r(\alpha^6) = \alpha^{10} + \alpha^9 + \alpha = \alpha^{12}$

TABLE 7.2: Steps for finding the error-location polynomial of the (15,9) RS code over $GF(2^4)$.

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	$1+\alpha^3 X$	1	1	1(take $\rho = 0$)
3	$1+\alpha^3 X+\alpha^3 X^2$	α^7	2	1(take $\rho = 0$)
4	$1+\alpha^4 X+\alpha^{12} X^2$	α^{10}	2	2(take $\rho = 2$)
5	$1+\alpha^7 X+\alpha^4 X^2+\alpha^6 X^3$	0	3	2(take $\rho = 3$)
6	$1+\alpha^7 X+\alpha^4 X^2+\alpha^6 X^3$	—	—	—

Step 2. To find the error-location polynomial $\sigma(X)$, we fill out Table 7.1 and obtain Table 7.2. Thus, $\sigma(X) = 1+\alpha^7 X+\alpha^4 X^2+\alpha^6 X^3$.

Step 3. By substituting $1, \alpha, \alpha^2, \cdots, \alpha^{14}$ into $\sigma(X)$, we find that α^3, α^9, and α^{12} are roots of $\sigma(X)$. The reciprocals of these roots are α^{12}, α^6, and α^3, which are the error-location numbers of the error pattern $\mathbf{e}(X)$. Thus, errors occur at positions X^3, X^6, and X^{12}.

A more straightforward algorithm where the correction term is evolved as the iterations go ahead is given in Vicker's text.

The algorithm is as follows:

1) Compute syndromes $S_1, \dots, S_{2t}$.
2) Initialize the algorithm by letting μ=0, $\sigma^{(0)}(x) = 1,\ \ l = 0$ and T(x)=x.
3) Set μ=μ+1 compute discrepancy d_μ ,

$$d_\mu = S_\mu + \sum_{i=1}^{l} \sigma_i^{(\mu-1)} S_{\mu-i}$$

4) If $d_\mu = 0$ then go to 8.
5) Modify the polynomial as:

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

6) If $2l \geq \mu$ then go to step 8.
7) Set $l = \mu - l$ and $T(x) = d_\mu^{-1}\sigma^{(\mu-1)}(x)$.
8) Set $T(x) = x.T(x)$.
9) If $\mu < 2t$ go to step 3.
10) Determine $\sigma(x) = \sigma^{(2t)}(x)$. If the roots are distinct and in the right field, then determine the error values, correct the errors and STOP.
11) Declare a decoding failure and STOP.

Next slide shows the problem above done again.

Example: Consider (7,3) RS Code over GF(8) with $r(x) = \alpha^2x^6 + \alpha^2x^4 + x^3 + \alpha^5x^2$.

Although we have done the generation of g(x) and encoding, let's start from ground zero for doing some exercise in Galois field arithmetic. Let's start with p(x)= x^3+x+1. Take α to be a primitive element of this field, i.e., a root of $s_1 = \alpha^{12}, s_2 = 1, s_3 = \alpha^{14}, s_5 = 0\,, s_6 = \alpha^{12}$

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

μ	s_μ	$\sigma^{(\mu)}(x)$	$d^{(\mu)}$	L_μ	T(x)
0	-	1	-	0	x
1	α^{12}	$1 + \alpha^{12}x$	α^{12}	1	α^3x
*2	1	$1 + \alpha^3x$	α^7	1	$\alpha x^8 + \alpha^5x^2$
**3	α^{14}	$1 + \alpha^{13}x + \alpha^5x^2$	1	2	$x + \alpha^3x^2$
4	α^{10}	$1 + \alpha^4x + \alpha^{12}x^2$	α^{11}	2	$\alpha^4x + \alpha^2x^2 + \alpha^9x^3$
5	0	$1 + \alpha^9x + \alpha^4x^3$	α^{10}	3	$\alpha^5x + \alpha^9x^2 + \alpha^3x^3$
6	α^{12}	$1 + \alpha^7x + \alpha^4x^2 + \alpha^6x^3$	α^{10}	3	-

$$\sigma(x) = 1 + \alpha^7 x + \alpha^4 x^2 + \alpha^6 x^3$$

- $d_1 = s_2 + \sigma_1 s_1 = 1 + \alpha^{12}.\alpha^{12} = \alpha^9 + 1 = \alpha^7$
- $d_2 = s_3 + \sigma_1 s_2 = 1 + \alpha^{14} + \alpha^3.1 = 1$

ρ(x). That is $\alpha^3 + \alpha + 1 = 0$ or $\alpha^3 = \alpha + 1$.

The field elements are:

0	0	0	0	0
1	1	0	0	1
α^1	0	1	0	α
$\alpha^2 = \alpha.\alpha$	0	0	1	α^2
$\alpha^3 = \alpha^2.\alpha$	1	1	0	α+1
α^4	0	1	1	$\alpha^2 + \alpha$
α^5	1	1	1	$\alpha^2 + \alpha + 1$
α^6	1	0	1	$\alpha^2 + 1$
α^7	1	0	0	1

Note:

- $\alpha^3 = \alpha^2.\alpha = \alpha + 1$
- $\alpha^4 = \alpha.\alpha^3 = \alpha(\alpha + 1) = \alpha^2 + \alpha$
- $\alpha^5 = \alpha^4.\alpha = (\alpha^2 + \alpha)\alpha = \alpha^3 + \alpha^2 = \alpha^2 + \alpha + 1$
- $\alpha^6 = \alpha\,(\alpha^2 + \alpha + 1) = \alpha^2 + 1$
- $\alpha^7 = \alpha\,(\alpha^2 + 1) = \alpha^3 + \alpha = \alpha + 1 + \alpha = 1$

Now, g(X) is:

$$g(x) = (x + \alpha)(x + \alpha^2)(x + \alpha^3)(x + \alpha^4)$$

$$= [x^2 + (\alpha + \alpha^2)x + \alpha^3][x^2 + (\alpha^3 + \alpha^4)x + \alpha^7]$$

$$= [x^2 + \alpha^4 x + \alpha^3][x^2 + \alpha^6 x + 1]$$

$$= x^4 + \alpha^3 x^3 + x^2 + \alpha x + \alpha^3$$

Computing Syndromes:

$$S_i = r(\alpha^i). \quad i = 1,2,3,4$$

In this case, since the number of parities are less than the number of information symbols, it is reasonable to use $r(\alpha^i) = S_i$. However, for high rate codes where $n - k \ll k$, it is better to divide r(x) by g(x) to get

$$r(x) = g(x)\,q(x) + b(x)$$

Where b(x) is a polynomial of degree less than or equal n-k.

$S_i = r(\alpha^i) = g(\alpha^i)q(\alpha^i) + b(\alpha^i) \qquad i = 1,2,\ldots,2t.$

Since $g(\alpha^i) = 0 \qquad i = 1,\ldots,2t$

$S_i = b(\alpha^i)$

Dividing $r(x) = \alpha^2x^6 + \alpha^2x^4 + x^3 + \alpha^5x^2$ by g(x)

$r(x) = (\alpha^2x^2 + \alpha^5x)g(x) + \alpha x^4 + \alpha^6x^3 + \alpha^6x^2 + \alpha x.$

So:

$s_1 = b(\alpha) = \alpha^5 + \alpha^9 + \alpha^8 + \alpha^2 = \alpha^6$

$s_2 = b(\alpha^2) = \alpha^9 + \alpha^{12} + \alpha^{10} + \alpha^3 = \alpha^3$

$s_3 = b(\alpha^3) = \alpha^{13} + \alpha^{15} + \alpha^{12} + \alpha^4 = \alpha^4$

$s_4 = b(\alpha^4) = \alpha^{17} + \alpha^{18} + \alpha^{14} + \alpha^5 = \alpha^3$

Now we use the algorithm:

μ	S_μ	$\sigma^{(\mu)}(x)$	d_μ	L	$T(x)$
0	-	1	-	0	x
1	α^6	$1 + \alpha^6x$	α^6	1	αx^*
2	α^3	$1 + \alpha^4x$	α^2	1	$\alpha x^{2^{**}}$
3	α^4	$1 + \alpha^4x + \alpha^6x^2$	α^5	2	$\alpha^2x + \alpha^6x^2$
4	α^3	$1 + \alpha^2x + \alpha x^2$	α^6	-	-

- Note:

For μ=1 L=0 $\rightarrow 2L < \mu \rightarrow L = \mu - L = 1$

And $T(x) = \frac{\sigma^{(0)}(x)}{d_1} = \frac{x}{\alpha^6} = \alpha x.$

- Note:

For μ=2

$$d_\mu = s_\mu \sum_{i=1}^{L} \sigma_i^{(\mu-1)}\, s_{\mu-i} \rightarrow \mu_2 = s_2 + \sigma_1^{(1)} S_1$$

Or

$$\mu_2 = \alpha^3 + \alpha^6.\alpha^6 = \alpha^3 + \alpha^5 = \alpha^2$$

$$2L = 2 \geq \mu = 2 \rightarrow T(x) = xT(x) \rightarrow T(x) = \alpha x^2$$

So:

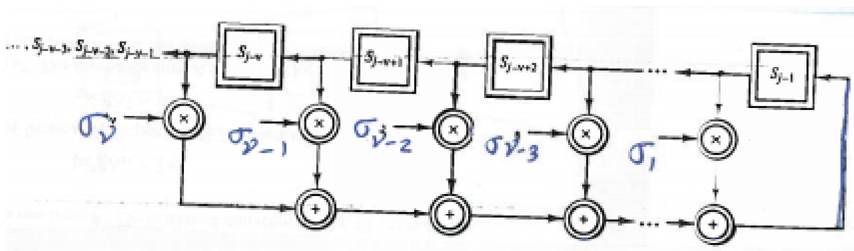
$$\sigma(x) = \alpha x^2 + \alpha^2 x + 1.$$

The above algorithm is based on message's Linear Feedback Shift Register (LESR) synthesis technique.

Note that for γ errors, we have the following Newton equalities.

$$s_j = \sigma_1 s_{j-1} + \sigma_2 s_{j-2} + \cdots + \sigma_\gamma s_{j-\gamma}$$

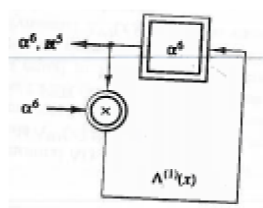
This relationship can be represented as LFSR circuit looking like:



The problem of finding error-locator polynomial is then to find an LFSR of minimal length such that the first 2t elements in the output sequence are $s_1, s_2, \ldots, s_{2t}$.

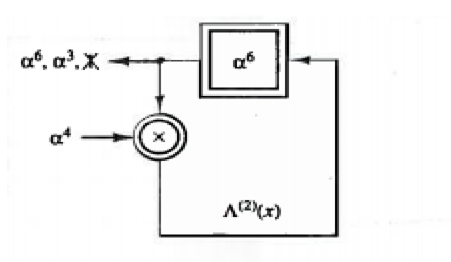
The coefficients of the filter are then the coefficient of σ(x).

For the above (7,3) RS code, we start with



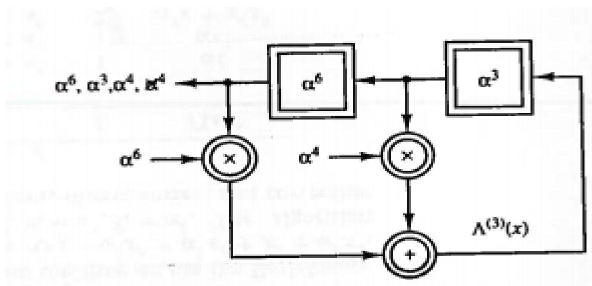
This works for the $s_1 = \alpha^6$ as it outputs the content of the register ,i.e., α^6. But after the application of the seconds clock, the output will be $\alpha^6 . \alpha^6 = \alpha^{12} = \alpha^5$ which is not $s_{2=}\alpha^3$.

To correct the situation, we change the filter tap to α^4 which is $\frac{\alpha^6}{\alpha^2}$ and therefore, the output after the clocking will be $\frac{\alpha^5}{\alpha^2} = \alpha^3 = s_2$.



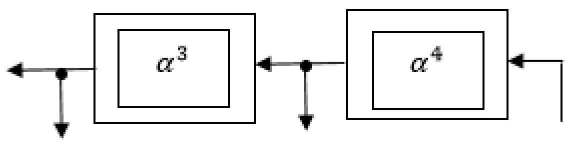
After the next clock the output will be $\alpha^3.\alpha^4 = 1$ which is not equal to $s_3 = \alpha^4$.

To correct this we need to add α^5 so that, we get $1 + \alpha^5 = \alpha^4 = s_3$. We keep the above and add a stage with α^6 in the register and α^6 as the tap.



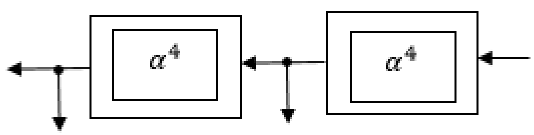
This circuit outputs α^6 first and then calculates $\alpha^6.\alpha^6 + \alpha^3.\alpha^4 = \alpha^5 + 1 = \alpha^4$

Content of the rightmost SR is moved to left and $\alpha^4$is loaded into it.



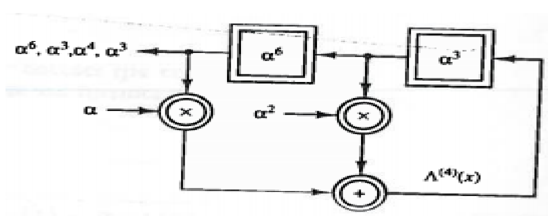
So, the next output is $\alpha^3 = s_2$.

Next $\alpha^3.\alpha^6 + \alpha^4.\alpha^4 = \alpha^2 + \alpha = \alpha^4$ is placed in right register and α^4 is moved left



Now α^4 is output which is s_3. But the next output is $\alpha^4 \neq s_4 = \alpha^3$

To avoid this, we modify the taps of the LFSR to:



It is easy to see that this circuit outputs $\alpha^6, \alpha^3, \alpha^4, \alpha^3, i.e\ s_1, s_2, s_3, s_4$

Finding the <u>Error Values</u>:

Now, we have found error-locator polynomial σ(X). We can solve it to find the error locations $\beta_i = \alpha^{j_i}$ i=1,2, …,γ.

Now we need to find $S_i = e_{j_i}$, i.e., error values at the error locations and correct them, That is the equations are:

$S_1 = e_{j_1}\alpha^{j_1} + e_{j_2}\alpha^{j_2} + \cdots + e_{j_\gamma}\alpha^{j_\gamma}$

$\vdots$

$S_{2t} = e_{j_1}\alpha^{2tj_1} + e_{j_2}\alpha^{2tj_2} + \cdots + e_{j_\gamma}\alpha^{2tj_\gamma}$

With $\alpha^{j_i}{}'s$ and S_i's known. Or equivalently:

$S_1 = s_1\beta_1 + s_2\beta_2 + s_\gamma\beta_\gamma$

$S_2 = s_1\beta_1^2 + s_2\beta_2^2 + \cdots + s_\gamma\beta_\gamma^2$

$\vdots$

$S_{2t} = s_1\beta_1^{2t} + \ s_2\beta_2^{2t} + \cdots + s_\gamma\beta_\gamma^{2t}$

Let's define the syndrome polynomial:

$$S(x) = S_1 + S_2X + \cdots + S_{2t}X^{2t} + S_{2t+1}X^{2t} + \cdots$$

$$= \sum_{j=1}^{\infty} S_j X^{j-1}$$

Note that this has an infinite number of terms whose first 2t terms are known:

$$S_j = \sum_{l=1}^{\gamma} S_l\,\beta_l^j \qquad j = 1,2,\dots,2t$$

Substituting this (but now for all terms), we get:

$$S(x) = \sum_{j=1}^{\infty} x^{j-1} \sum_{l=1}^{\gamma} S_l\,\beta_l^j$$

$$= \sum_{l=1}^{\gamma} S_l\,\beta_l \sum_{j=1}^{\infty} (\beta_l x)^{j-1}$$

But

$$\sum_{j=1}^{\infty} (\beta_l x)^{j-1} = \frac{1}{1 + \beta_l x}$$

So:

$$S(x) = \sum_{l=1}^{\gamma} \frac{S_l \beta_l}{1 + \beta_l x}$$

$\sigma(x) = \prod_{i=1}^{\gamma}(1 + \beta_i x)$So:

$$S(x)\sigma(x) = \sum_{l=1}^{\gamma} S_L\, \beta_L \prod_{i=1.\ i\neq\rho}^{\gamma} (1 + \beta_i x) \triangleq Z_0(x)$$

Also,

$$\sigma(x)S(x) = \left[1 + \sigma_1 x + \cdots + \sigma_\gamma x^\gamma\right][S_1 + S_2 x + S_3 x^2 + \cdots]$$
$$= S_1(S_2 + \sigma_1 S_1)x + (S_3 + \sigma_1 S_2 + \sigma_2 S_1)x^2 + \cdots$$
$$\ldots + \left(\sigma_{2t} + \sigma_1 S_{2t-1} + \cdots + \sigma_\gamma S_{2t-\gamma}\right)x^{2t-1} + \cdots$$

So:

$$Z_0(x) = S_1 + (S_2 + \sigma_1 S_1)x + (S_3 + \sigma_1 S_2 + \sigma_2 S_1)x^2 + \cdots + (S_\gamma + \sigma_1 S_{\gamma-1} + \cdots + \sigma_{\gamma-1} S_1)x^{\gamma-1}$$

Let's substitute β_k^{-1} in $Z_0(x)$:

$$Z_0(\beta_k^{-1}) = \sum_{l=1}^{\gamma} S_l \beta_l \prod_{i=1, i\neq l}^{\gamma} (1 + \beta_i \beta_k^{-1})$$
$$= S_k \beta_k \prod_{i=1, i\neq k}^{\gamma} (1 + \beta_i \beta_k^{-1})$$

Taking derivative of σ(x)

$$\sigma'(x) = \frac{d}{dx}\prod_{i=1}^{\gamma}(1 + \beta_i x) = \sum \beta_l \prod_{i=1, i\neq l}^{\gamma} (1 + \beta_i x)$$

Then

$$\sigma^1(\beta_k^{-1}) = \beta_k \prod_{i=1, i\neq k}^{\gamma} (1 + \beta_i \beta_k^{-1})$$

So,

$$S_k = \frac{Z_0(\beta_k^{-1})}{\sigma^1(\beta_k^{-1})}$$

Let's $[\sigma(x)S(x)]_{2t}$ represent the first 2t terms of σ(x)S(x). Then

$$\sigma(x)S(x) - [\sigma(x)S(x)]_{2t}$$

Is divisible byX^{2t}.

That is:

$$\sigma(x)S(x) \equiv [\sigma(x)S(x)]_{2t} mod\ X^{2t}$$

But,

$$[\sigma(x)S(x)]_{2t} = Z_0(x)$$

And we have:

$$\sigma(x)S(x) \equiv Z_0(x) mod x^{2t}$$

This is called the <u>key equation</u> that has to be solved in decoding of RS codes.

Example: Consider the (7,4) code in the previous example:

We had $S_1 = \alpha^6, S_2 = \alpha^3, S_3 = \alpha^4\ snd S_4 = \alpha^3$,

So:

$S(x) = \alpha^6 + \alpha^3 x + \alpha^4 x^2 + \alpha^3 x^3$

Also, we found:

$$\sigma(x) = 1 + \alpha^2 x + \alpha x^2 \rightarrow \sigma'(x) = \alpha^2 + 2\alpha x = \alpha^2$$

So:

$$Z_0(x) = \sigma(x)S(x) mod x^4$$
$$= (1 + \alpha^2 x + \alpha x^2)(\alpha^6 + \alpha^3 x + \alpha^4 x^2 + \alpha^3 x^3)$$
$$= \alpha^6 + x$$

We can find the error locations by solving σ(x)=0 to get $\beta_1 = \alpha^3$ and $\beta_2 = \alpha^5$

So,

$$e_3 = S_1 = \frac{Z_0(\alpha^{-3})}{\sigma'(\alpha^{-3})} = \frac{\alpha^6 + (\alpha^{-3})}{\alpha^2} = \alpha^4 + \alpha^2 = \alpha$$

And

$$e_5 = S_2 = \frac{z_0(\alpha^{-5})}{\sigma'(\alpha^{-5})} = \frac{\alpha^6 + \alpha^{-5}}{\alpha^2} = \alpha^4 + 1 = \alpha^5$$

So,

$$e(X) = \alpha X^3 + \alpha^5 X^5$$