the symbol to be corrected. The AND gate produces a 1 output only when [1, 0 ··· 0] appears in the syndrome register and corrects the symbol in error.

3. Bose - Chaudhuri - Hocquenghem (BCH) codes

These codes were discovered by Hocquenghem in 1959 and independently by Bose and Chaudhuri in 1960. The BCH codes are the most powerful cyclic block codes known for correcting random errors. However, their performance on random channels can not match convolutional codes with sequential or Viterbi decoding (particularly with soft decision decoding)

Def: Least common multiple (lcm)

The least common multiple of $a(D), b(D) \cdots d(D)$, is the polynomial of lowest degree that is divisible by $a(D), \ldots, d(D)$.

Def: BCH Code

Let α be any nonzero element of $GF(q^m)$ and order of $\alpha = n$. Let m_0 be any integer and d any integer in the range $2 \le d \le n$. The BCH code corresponding to α, m_0, and d is the cyclic code with symbols from $GF(q)$ where the generator polynomial $g(D)$ is the minimum degree monic polynomial over $GF(q)$ having $\alpha^{m_0}, \alpha^{m_0+1}, \ldots, \alpha^{m_0+d-2}$ as roots.

<u>Remark</u>: Since $d \le n$ = order α these $d-1$ roots are distinct.

Let $m_i(D)$ be the minimal polynomial for α^i over $GF(q)$, Then

$$g(D) = \operatorname{lcm}[m_{m_0}(D), m_{m_0+1}(D), \ldots, m_{m_0+d-2}(D)]$$

Even if $d < n$ it may happen that $m_{m_0+d-1}(D)$ is already a factor of $g(D)$ since α^{m_0+d-1} may have the same minimal polynomial as α^i for $m_0 \le i \le m_0+d-2$. A BCH code is not defined for such d since it is exactly the same as the BCH code for a larger d.

EX: $q=2$, $m_0=1$

assume $d=2$ then α^1, $\alpha^{m_0+d-2}=\alpha$ are roots of $g(D)$, but α^2 must also be a root of $g(D)$ so no BCH code is defined with $d=2$ in this example. The code for $d=2$ is the same as the code for $d=3$.

<u>Theorem 17</u> Calculation of N

Let $g(D) = \prod_{i=1}^{l} m_i(D)$ where $\{m_i(D)\}$ are distinct monic irreducible polynomials over $GF(q)$. Let n_i be the multiplicative order of the roots of $m_i(D)$. Then $N = \operatorname{lcm}(n_1, n_2, \ldots, n_l)$

<u>Proof</u>:

For each i, n_i is the smallest integer such that $m_i(D)$ divides $D^{n_i}-1$. Since $g(D)$ must divide D^N-1, α_i is also a root of D^N-1 and therefore

n_i divides N. Therefore $N \geq \operatorname{lcm}(n_1, n_2, \cdots, n_\ell)$. On the other hand $\alpha_i^{\operatorname{lcm}(n_1, \cdots, n_\ell)} = 1$ for all i so that $m_i(D)$ divides $D^{\operatorname{lcm}(n_1, \cdots, n_\ell)} - 1$ $\forall i$ and therefore $g(D)$ divides $D^{\operatorname{lcm}(n_1, \cdots, n_\ell)} - 1$. $\Rightarrow N \leq \operatorname{lcm}(n_1, \cdots, n_\ell)$. Thus $N = \operatorname{lcm}(n_1, \cdots, n_\ell)$.

Q.E.D.

<u>Theorem 18</u> Length of BCH Codes

If $d = 2$, $g(D)$ is the minimal polynomial of α^{m_0} and $N =$ order α^{m_0}

For $d \geq 3$, $N =$ order α

<u>Proof:</u>

Let $n =$ order α. Then $\forall i$, $[\alpha^i]^n = \{\alpha^n\}^i = 1$ so n is a common multiple of $\{n_i\}$ so that n_i divides n. By Theorem 17 $N \leq n$. Since $g(D)$ divides $D^N - 1$, α^i for $m_0 \leq i \leq m_0 + d - 2$ is a root of $D^N - 1$. For $d \geq 3$ $(\alpha^{m_0})^N = 1$ and $(\alpha^{m_0+1})^N = 1 \Rightarrow \alpha^{m_0 N} \alpha^N = 1 \Rightarrow \alpha^N = 1$ since $\alpha^{m_0 N} = 1$. Thus n divides N and $n \leq N$.

QED.

In most cases α is taken as a primitive element of $GF(q^m)$ so that for $d \geq 3$, $N = q^m - 1$. m_0 is usually chosen as 1.

<u>EXAMPLE 1:</u> $m_0 = 1$, $d = 5$, $q = 2$, $m = 4$

Let α be a root of $f_1(D) = D^4 + D + 1$

$GF(2^4)$ Field Elements

	1	α	α^2	α^3
0	0	0	0	0
1	1	0	0	0
α	0	1	0	0
α^2	0	0	1	0
α^3	0	0	0	1
α^4	1	1	0	0
α^5	0	1	1	0
α^6	0	0	1	1
α^7	1	1	0	1
α^8	1	0	1	0
α^9	0	1	0	1
α^{10}	1	1	1	0
α^{11}	0	1	1	1
α^{12}	1	1	1	1
α^{13}	1	0	1	1
α^{14}	1	0	0	1

$m_0 + d - 2 = 1 + 5 - 2 = 4$ so that $g(D)$ must have $\alpha, \alpha^2, \alpha^3, \alpha^4$ as roots

$\alpha, \alpha^2, \alpha^4$ are roots of $m_1(D) = D^4 + D + 1$

For $m_3(D)$ roots are $\alpha^3, \alpha^6, \alpha^{12}, \alpha^{24} = \alpha^9$

$$m_3(D) = c_0 + c_1 D + c_2 D^2 + c_3 D^3 + D^4$$

$$m_3(\alpha^3) = c_0 \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} + c_1 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} + c_2 \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} + c_3 \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = 0$$

$c_0 = 1, \quad c_3 = 1, \quad c_2 = 1, \quad c_1 = 1 + 1 + 1 = 1$

$m_3(D) = 1 + D + D^2 + D^3 + D^4$

and $g(D) = m_1(D)\, m_3(D) = D^8 + D^7 + D^6 + D^4 + 1.$

Therefore $N - K = 8$, $N = 15$, $K = 7$ giving a $(15, 7)$ code.

<u>Comment:</u>

for $q = 2$, $m_i(D) = m_{2i}(D)$. Therefore with $m_0 = 1$, $q = 2$, d odd

$$g(D) = \mathrm{lcm}\,[m_1(D)\, m_3(D) \cdots m_{d-2}(D)]$$

Each $m_i(D)$ has degree $\leq m$ so that $N - K = \deg g(D) \leq m(d-1)/2$. It will be shown that d is a lower bound on d_{min}. Therefore codes of length $2^m - 1$ exist that correct all patterns of t or fewer errors with at most mt check symbols. For $m_0 \neq 1$, $q \neq 2$ the weaker bound $N - K \leq 2mt$ applies.

<u>Specification of H in terms of Roots of $g(D)$</u>

For an arbitrary cyclic code, it has already been shown that a check matrix can be determined from $h(D) = (D^N - 1)/g(D)$. In some cases, particularly for BCH codes, it will be more convenient to specify H in terms of the roots of $g(D)$. This representation

as given in the following theorem.

<u>Theorem 18 A</u>

Let $g(D) = \prod_{i=1}^{\ell} m_i(D)$ where $\{m_i(D)\}$ are distinct, monic, irreducible polynomials. Let α_i be a root of $m_i(D)$. Then a check matrix for the code is:

$$H = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{N-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{N-1} \\ \vdots & & & & \\ 1 & \alpha_\ell & \alpha_\ell^2 & \cdots & \alpha_\ell^{N-1} \end{bmatrix}$$

That is, X is a ~~codeword~~ iff $\vec{X} H^t = \vec{0}$

<u>Proof:</u>

Let $\mathcal{C}$ be the space of codewords. ~~and $\mathcal{N}$ the null space of the rows of H~~ If Let $\vec{X}$ be a vector ~~$\in \mathcal{N}$~~ such ~~so~~ that $\vec{X} H^T = \vec{0}$. ~~and~~ Then $\alpha_1, \cdots, \alpha_\ell$ are roots of $X(D)$. Thus $m_1(D), \cdots, m_\ell(D)$ divide $X(D)$ so that $g(D)$ divides $X(D)$ and $\vec{X} \in \mathcal{C}$. [(Only if) ~~so that $\mathcal{C} \supset \mathcal{N}$.~~ On the other hand any codeword has $\alpha_1, \cdots, \alpha_\ell$ as roots so that all code words $X H^t = 0$ ~~$\in \mathcal{N}$ and $\mathcal{N} \supset \mathcal{C}$~~. Q.E.D.

EXAMPLE 1 (CONT.)

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

$$= \left[\begin{array}{ccccccccccccccc} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ \hline 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \end{array}\right]$$

where the representations for α^k from p. 156 were used.

EXAMPLE 2 Golay (23, 12) Code [See Lin for simple Decodi. algorithm]

To find binary cyclic codes with $N = 23$ it is necessary to find an element $\beta \in GF(2^k)$ with order 23 for some k. In $GF(2^{11})$ there are $2^{11} - 1 = 2047 = 23 \cdot 89$ non zero field elements. Let α be a primitive element $\in GF(2^{11})$ and let $\beta = \alpha^{89}$. Then order $\beta = 2047 / \gcd(2047, 89) = 23$. The roots of the minimal polynomial for β are

$\beta, \beta^2, \beta^4, \beta^8, \beta^{16}, \beta^{32} = \beta^9, \beta^{18}, \beta^{36} = \beta^{13}, \beta^{26} = \beta^3, \beta^6$ and β^{12}.

Thus $\deg m_\beta(D) = 11$. Consider the element $\gamma = \beta^5$ which is not a root of $m_\beta(D)$. The order $\gamma = 23 / \gcd(23, 5) = 23$ so that by similar reasoning $\deg m_\gamma(D) = 11$. Thus the factorization into irreducible polynomials over $GF(2)$ for $D^{23} + 1$ is $D^{23} + 1 = (D+1)\, m_\beta(D)\, m_\gamma(D)$. Through trial and error it can be shown that

$m_\beta(D) = D^{11} + D^9 + D^7 + D^6 + D^5 + D + 1$ and

$m_\gamma(D) = D^{11} + D^{10} + D^6 + D^5 + D^4 + D^2 + 1$. The generator polynomial for the Golay code is $g(D) = m_\beta(D)$. The elements $\beta, \beta^2, \beta^3, \beta^4$ are all roots of $g(D)$ so the code is a BCH code with $m_0 = 1$ and $d = 5$. Thus $d_{min} \geq 5$. By generating all code words or by the analysis in Sec 15.2 of Berlekamp it can be shown that actually $d_{min} = 7$ so that the code can correct all patterns of $t = 3$ or fewer errors. Since $\sum_{i=0}^{3} \binom{23}{i} = 2^{11}$, the code is a perfect code.

Ex. 3 Hamming Codes

Let $f(D)$ be a primitive polynomial of degree $N-K$ over $GF(2)$ with α as a root, $m_0 = 1$, $d = 3$. Then $g(D)$ must have α, α^2 as roots. So

$$g(D) = f(D)$$

Then

$$H = [1 \quad \alpha \quad \cdots \quad \alpha^{N-1}]$$

Suppose $E(D) = D^k$. Then

$$S = \vec{E} H^T = \alpha^k$$

So $k = \log_\alpha S$ is error position.

<u>Theorem 19</u> d_{min} for BCH codes

Let a BCH code over $GF(q)$ have parameters α, m_0, d. Then $d_{min} \geq d$.

<u>Proof:</u>

Let

$$H = \begin{bmatrix} (\alpha^{m_0})^0 & (\alpha^{m_0})^1 & (\alpha^{m_0})^2 & \cdots & (\alpha^{m_0})^{N-1} \\ (\alpha^{m_0+1})^0 & \alpha^{m_0+1} & & \cdots & (\alpha^{m_0+1})^{N-1} \\ \vdots & & & & \\ (\alpha^{m_0+d-2})^0 & \alpha^{m_0+d-2} & & \cdots & (\alpha^{m_0+d-2})^{N-1} \end{bmatrix}$$

It will be shown that any $d-1$ columns of H are linearly independent so that there are no codewords of weight $\leq d-1$. Let $0 \leq i_1 < i_2 < \cdots < i_{d-1} \leq N-1$ be $d-1$ columns corresponding to the roots raised to the i_j power. Let Δ be the determinant over $GF(q^m)$ of these columns, i.e.

$$\Delta = \begin{vmatrix} (\alpha^{m_0})^{i_1} & (\alpha^{m_0})^{i_2} & \cdots & (\alpha^{m_0})^{i_{d-1}} \\ \vdots & \vdots & & \vdots \\ (\alpha^{m_0+d-2})^{i_1} & (\alpha^{m_0+d-2})^{i_2} & \cdots & (\alpha^{m_0+d-2})^{i_{d-1}} \end{vmatrix}$$

$$= \alpha^{m_0(i_1+i_2+\cdots+i_{d-1})} \begin{vmatrix} 1 & 1 & \cdots & 1 \\ \alpha^{i_1} & \alpha^{i_2} & & \alpha^{i_{d-1}} \\ (\alpha^{i_1})^2 & (\alpha^{i_2})^2 & & \vdots \\ \vdots & \vdots & & \\ (\alpha^{i_1})^{d-2} & (\alpha^{i_2})^{d-2} & \cdots & (\alpha^{i_{d-1}})^{d-2} \end{vmatrix}$$

a determinant of this form is known as a Vandermonde determinant.

<u>Lemma</u>:

Let $\bar{\Delta} = \begin{vmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & & x_n \\ x_1^2 & x_2^2 & & x_n^2 \\ \vdots & \vdots & & \vdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{vmatrix}$

Then $\bar{\Delta} = \prod_{j=2}^{n} \left[\prod_{i<j} (x_j - x_i) \right]$

<u>Proof</u>:

If $\bar{\Delta}$ is expanded in a sum of products, each product has one term from each row. Therefore the degree of the factors is $1+2+\cdots+n-1 = (n-1)n/2$. If $x_i = x_j$, $\bar{\Delta} = 0$ because two columns are equal, so that $x_j - x_i$ is a factor of $\bar{\Delta}$ for $j \neq i$. The number of factors of the form $x_j - x_i$ for $j > i \geq 1$ and $j = 2, \cdots, n$ is $1 + 2 + \cdots + (n-1)$. Thus the product of these factors is a constant $\times\ \bar{\Delta}$. Writing out the factors of this product

$$c\bar{\Delta} = (x_2 - x_1)[(x_3 - x_2)(x_3 - x_1)][(x_4 - x_3)(x_4 - x_2)(x_4 - x_1)] \cdots [(x_n - x_{n-1}) \cdots (x_n - x_1)]$$

$$= (x_2 - x_1)[x_3^2 + \cdots][x_4^3 + \cdots] \cdots [x_n^{n-1} + \cdots]$$

Therefore one term in the expansion of the right hand side is $1 \cdot x_2 \cdot x_3^2 \cdot (\cdots) \cdot x_n^{n-1}$. This is just the

product of the diagonal terms of the Vandermonde matrix so that $c = 1$ Q.E.D.

applying the Lemma

$$\Delta = \alpha^{m_0(i_1+i_2+\cdots+i_{d-1})} \prod_{j=2}^{d-1}\left[\prod_{k<j}(\alpha^{i_j} - \alpha^{i_k})\right]$$

Since $0 < i_k < i_j \leq N-1 <$ order α, $\alpha^{i_j} \neq \alpha^{i_k}$ and $\Delta \neq 0$. If $\Delta \neq 0$ no nonzero linear combination of $d-1$ columns of H with coefficients $\in GF(q^m)$ can be $\bar{0}$ and these columns are linearly independent. Therefore $d_{min} \geq d$ QED.

<u>comment:</u>

H has only $d-1$ rows when considered as a matrix over $GF(q^m)$ so that rank of $H \leq d-1$. Thus any set of d columns must be linearly dependent over $GF(q^m)$. However this does not imply that they are linearly dependent over $GF(q)$ so that, in general, all that can be said is that $d \leq d_{min}$.

<u>Special Case</u>: $m=1$, Reed-Solomon Codes

The Reed-Solomon codes have roots and code symbols taken from $GF(q)$. The argument in the previous paragraph shows that there are codewords of weight d over $GF(q)$ and Theorem 19 implies $d_{min} \geq d$. Therefore $d_{min} = d$ for $m=1$. Since $\alpha^{m_0}, \alpha^{m_0+1}, \ldots \alpha^{m_0+d-2}$

-162-

are $\in GF(q)$, $g(D) = (D-\alpha^{m_0})(D-\alpha^{m_0+1})\cdots(D-\alpha^{m_0+d-2})$

The degree of $g(D)$ is $d-1 = N-K$, the number of check symbols. $(d-1 = 2t = N-K)$

<u>Lemma:</u>

For any linear block code with $N-K$ check symbols $d_{min} \leq N-K+1$

<u>Proof:</u>

H has rank $N-K$ so that any set of $N-K+1$ columns must be linearly dependent. This implies $d_{min} \leq N-K+1$ QED.

Therefore the Reed-Solomon codes have the greatest possible minimum distance for a linear block code with $N-K$ check symbols.

<u>EX:</u> $m=1$, $m_0=1$, $q=2^4$, $d=5$

Let α be a root of D^4+D+1, $\Rightarrow N = 2^4-1 = 15$

Then $g(D) = (D-\alpha)(D-\alpha^2)(D-\alpha^3)(D-\alpha^4)$

$$= D^4 + \alpha^{13}D^3 + \alpha^6 D^2 + \alpha^3 D + \alpha^{10}$$

Since $\deg g = 4$, $K = 15-4 = 11$ and this is a $(15, 11)$ code over $GF(2^4)$ that can correct 2 or less errors. A single error occurs when the $GF(q)$ received symbol is not equal to the $GF(q)$ transmitted symbol. The code symbols are taken from the alphabet of 16 $GF(2^4)$ elements. In practice the $GF(2^4)$ elements can be ~~are~~ represented as binary 4-tuples. The codes are then

block codes with $N = 4 \times 15$ and $K = 4 \times 11$.
Notice that 1, 2, 3 or 4 binary bit errors in the 4-tuple representing the $GF(2^4)$ code symbol act as only one $GF(2^4)$ symbol error!

Def: <u>Burst of length b</u>

a burst of length b is a sequence of b consecutive symbols starting and ending with a nonzero field element.

Consider the Reed-Solomon codes over $GF(2^m)$. If the code symbols are transmitted as binary m-tuples, a burst error of $b = (t-1)m+1$ channel symbols can affect at most t $GF(2^m)$ symbols. If $d = 2t+1$ the code can correct channel burst of length $b = (t-1)m+1$.

<u>Error Correction for BCH Codes</u>

Ref: Gallager - Ch 6, Berlekamp, Peterson

For BCH codes $d_{min} \geq d$ so that all patterns of $\left\lfloor \frac{d-1}{2} \right\rfloor$ or fewer errors can be corrected. Except for perfect codes, usually some error patterns of greater weight can be corrected. "No practical algorithm is know to accomplish this!"
Note: With advances in VLSI, syndrome decoding tables are often a good approach.

Let $X(D)$ be the transmitted codeword, $Y(D)$ the received word, and $E(D) = Y(D) - X(D)$ be the channel error sequence. Let the syndrome $\vec{S}$ be defined as $\vec{S} = \vec{Y} H^T$ where

$$H = \begin{bmatrix} 1 & \alpha^{m_0} & (\alpha^{m_0})^2 & \cdots & (\alpha^{m_0})^{N-1} \\ 1 & \alpha^{m_0+1} & & \cdots & (\alpha^{m_0+1})^{N-1} \\ \vdots & & & & \\ 1 & \alpha^{m_0+d-2} & & \cdots & (\alpha^{m_0+d-2})^{N-1} \end{bmatrix}$$

Thus $\vec{s} = [s_0, \ldots, s_{d-2}]$

where $s_i = y(\alpha^{m_0+i})$ for $0 \le i \le d-2$

$= E(\alpha^{m_0+i})$

Suppose that $t \le \lfloor \frac{d-1}{2} \rfloor$ errors have occurred so that $E(D) = e_{n_1} D^{n_1} + e_{n_2} D^{n_2} + \cdots + e_{n_t} D^{n_t}$. Then

$$s_i = \sum_{k=1}^{t} e_{n_k} (\alpha^{m_0+i})^{n_k} \qquad i = 0, \ldots, d-2$$

$$= \sum_{k=1}^{t} e_{n_k} (\alpha^{n_k})^{m_0+i}$$

Let $v_k = e_{n_k}$ be the <u>error value</u> and $u_k = \alpha^{n_k}$ be the <u>error locator</u>. The u_k are all distinct since $0 \le n_1 < n_2 < \cdots < n_t \le N-1 = (\text{order } \alpha) - 1$ for $d \ge 3$.

Then

$$s_i = \sum_{k=1}^{t} v_k u_k^{m_0+i} \qquad \text{for } i = 0, \ldots, d-2 \qquad (1)$$

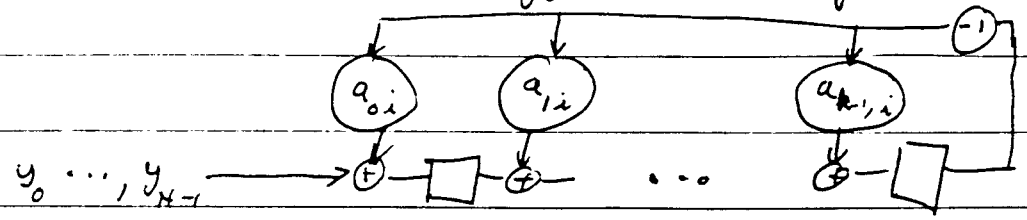
<u>Calculation of $\vec{s}$</u>

Let the minimal polynomial for α^{m_0+i} be $m_{m_0+i}(D)$. Then by the division algorithm

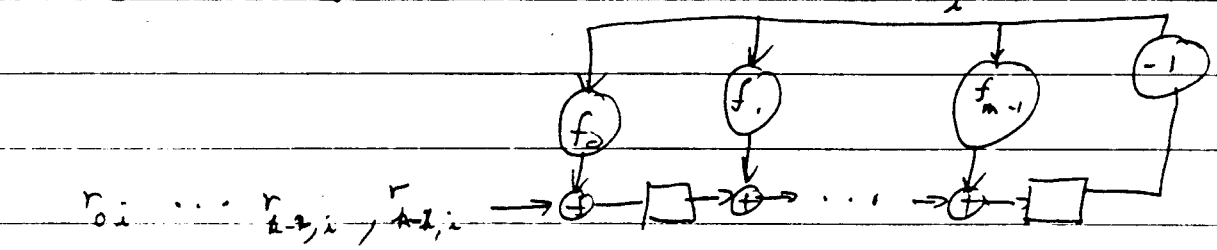
$$y(D) = q_i(D)\, m_{m_0+i}(D) + r_i(D) \quad \text{for } i = 0, \ldots, d-2$$

and $y(\alpha^{m_0+i}) = r_i(\alpha^{m_0+i}) = s_i$

If $m_{m_0+i}(D) = a_{0i} + a_{1i}D + \cdots + a_{ki}D^k$ The circuit below calculates the coefficients of $r_i(D) = r_{0i} + r_{1i}D + \cdots + r_{k-1,i}D^k$



If α is a root of $f(D) = f_0 + \cdots + D^m$ the circuit below calculates $r_i(\alpha^{m_0+i})$.



The coefficient $r_{k-1,i}$ is shifted into the register and then the register is shifted m_0+i-1 times. At this point $r_{k-1,i}\alpha^{m_0+i-1}$ is in the register. On the next shift $r_{k-2,i}$ is added into the register and the contents become $r_{k-2,i} + r_{k-1,i}\alpha^{m_0+i}$. On every m_0+i shifts a new coefficient is shifted into the register so that its final contents are $r_i(\alpha^{m_0+i})$. Since $[r(D)]^q = r(D^q)$, the circuits shown above can be used to calculate several of the s_j's by shifting the lower register the appropriate number of times between entering coefficients. Alternately $y(D^{m_0+i})$ can be shifted into the lower register for $i = 0, \cdots, d-2$. This usually requires more computation.

$r(\alpha^{m_0+i})$ can also be calculated instantaneously from the coefficients $\{r_0, \ldots, r_k\}$. Any element $\in GF(q^m)$ can be represented as a linear combination of $1, \alpha, \ldots, \alpha^{m-1}$. Therefore $\alpha^l = [1, \alpha, \ldots, \alpha^{m-1}] \begin{bmatrix} c_{0,l} \\ c_{1,l} \\ \vdots \\ c_{m-1,l} \end{bmatrix}$

and $$r(\alpha^{m_0+i}) = [1, \alpha, \ldots, \alpha^{m-1}] \underbrace{\begin{bmatrix} 1 & c_{0,m_0+i} & c_{0,2(m_0+i)} & \cdots & c_{0,k(m_0+i)} \\ 0 & c_{1,m_0+i} & c_{1,2(m_0+i)} & & \vdots \\ \vdots & \vdots & \vdots & & \\ 0 & c_{m-1,m_0+i} & c_{m-1,2(m_0+i)} & \cdots & c_{m-1,k(m_0+i)} \end{bmatrix}}_{C} R$$

Therefore in terms of the m-tuple representation of $GF(q^m)$

$$s_i = r(\alpha^{m_0+i}) = C R_{m \times 1}$$

In the binary case the elements of C are just 1's and 0's so that the coefficients are either connected or not connected to the appropriate adder.

The decoding problem given $s_0, \ldots, s_{d-2}$ is to solve Eq 1 for the error locators $\{u_k\}$ and error values $\{v_k\}$. Eq 1 is a set of $d-1$ nonlinear equations in the $2t$ unknowns $\{v_k\}$ and $\{u_k\}$. A brute force solution is to substitute all combinations of u's and v's and choose the most likely solution. This normally involves excessive calculation. The approach followed here is a modification of

a method proposed by Berlekamp. Let

$$S_\infty(D) = \sum_{i=0}^{\infty} S_i D^i$$

where $S_i = \sum_{j=1}^{t} V_j U_j^{m_0+i}$

Then

$$S_\infty(D) = \sum_{i=0}^{\infty} \sum_{j=1}^{t} V_j U_j^{m_0+i} D^i = \sum_{j=1}^{t} V_j U_j^{m_0} \sum_{i=0}^{\infty} U_j^i D^i$$

$$= \sum_{j=1}^{t} V_j U_j^{m_0} \frac{1}{1-U_j D} \qquad (2)$$

Define the error locator polynomial to be

$$\sigma(D) = \prod_{l=1}^{t} (1-U_l D) = \sigma_0 + \sigma_1 D + \cdots + \sigma_t D^t \qquad (3)$$

note: $\sigma_0 = 1$

The roots of $\sigma(D)$ are the inverses of the error locators, i.e., $\{U_l^{-1}\}$. Now

$$\sigma(D) S_\infty(D) = \sum_{j=1}^{t} V_j U_j^{m_0} \prod_{\substack{l=1 \\ l \neq j}}^{t} (1-U_l D) = A(D) \qquad (4)$$

Let $[B(D)]_i^j = \begin{cases} \sum_{l=i}^{j} b_l D^l & j \geq i \\ 0 & j < i \end{cases}$

If $j > \deg B(D) = L$, then $b_{L+1}, \ldots b_j$ will be taken as 0.

Let $S(D) = S_0 + S_1 D + \cdots + S_{d-2} D^{d-2}$

Then $S(D) = [S_\infty(D)]_0^{d-2}$. Using this notation

$$[\sigma(D) S_\infty(D)]_0^{d-2} = [\sigma(D) S(D) + \sum_{i=d-1}^{\infty} S_i D^i \sigma(D)]_0^{d-2}$$

$$= [\sigma(D) S(D)]_0^{d-2} = [A(D)]_0^{d-2}$$

From Eq 4 it can be seen that $\deg A(D) \leq t-1 < d-2$

and $[A(D)]_t^{d-2} = [\sigma(D) S(D)]_t^{d-2} = 0$

This is equivalent to the following set of $d-t-1$ equations:

$$\boxed{\begin{array}{l} \sigma_0 S_t + \sigma_1 S_{t-1} + \cdots + \sigma_t S_0 = 0 \\ \sigma_0 S_{t+1} + \sigma_1 S_t + \cdots + \sigma_t S_1 = 0 \\ \vdots \\ \sigma_0 S_{d-1} + \sigma_1 S_{d-3} + \cdots + \sigma_t S_{d-2-t} = 0 \end{array}} \qquad (5)$$

From Eq 3 it can be seen that $\sigma_0 = 1$ so that there is a set of $d-t-1 \geq t$ linear equations in t unknowns. If a solution exists then the error locator can be found from the roots of $\sigma(D)$.

The decoder does not know the number of channel errors t. The decoder can attempt to solve Eq 5 for different t. For $t \leq \lfloor \frac{d-1}{2} \rfloor$ the following theorem shows that the correct solution is found.

Theorem 20

Let $t \leq \lfloor \frac{d-1}{2} \rfloor$ and $\sigma(D) = \prod_{i=1}^{t} (1 - U_i D)$. Let $\hat{t}$ be the smallest integer for which a polynomial $\hat{\sigma}(D) = 1 + \hat{\sigma}_1 D + \cdots + \hat{\sigma}_{\hat{t}} D^{\hat{t}}$ exists satisfying $[\hat{\sigma}(D) S(D)]_{\hat{t}}^{d-2} = 0$. Then $t = \hat{t}$ and $\sigma(D) = \hat{\sigma}(D)$.

Proof:

$[\hat{\sigma}(D) S(D)]_{\hat{t}}^{d-2} = 0$ is equivalent to

$$\sum_{l=0}^{\hat{t}} \hat{\sigma}_l S_{i-l} = 0 \qquad \hat{t} \leq i \leq d-2$$

or
$$\sum_{\ell=0}^{\hat{t}} \hat{\sigma}_\ell \sum_{j=1}^{t} v_j U_j^{m_0+i-\ell} = 0$$

$$= \sum_{j=1}^{t} v_j U_j^{m_0+i} \sum_{\ell=0}^{\hat{t}} \hat{\sigma}_\ell U_j^{-\ell}$$

$$= \sum_{j=1}^{t} v_j \hat{\sigma}(U_j^{-1}) U_j^{m_0+i} = 0 \qquad \hat{t} \le i \le d-2$$

This is a set of $d-1-\hat{t}$ equations in the t unknowns $v_j \hat{\sigma}(U_j^{-1})$ for $j=1,\dots,t$. By assumption $\hat{t}$ is the smallest integer such that

$$[\hat{\sigma}(D)S(D)]_{\hat{t}}^{d-1} = 0$$

so that $\hat{t} \le t$. If $t \le \lfloor \frac{d-1}{2} \rfloor$, then $2t \le d-1$ or $t \le d-1-t \le d-1-\hat{t}$. Consider only the first t equations in the set above:

$$\underbrace{\begin{bmatrix} U_1^{m_0+\hat{t}} & U_2^{m_0+\hat{t}} & \cdots & U_t^{m_0+\hat{t}} \\ U_1^{m_0+\hat{t}+1} & & \cdots & U_t^{m_0+\hat{t}+1} \\ \vdots & & & \\ U_1^{m_0+\hat{t}+t-1} & & \cdots & U_t^{m_0+\hat{t}+t-1} \end{bmatrix}}_{U} \begin{bmatrix} v_1\hat{\sigma}(U_1^{-1}) \\ v_2\hat{\sigma}(U_2^{-1}) \\ \vdots \\ v_t\hat{\sigma}(U_t^{-1}) \end{bmatrix} = \begin{bmatrix} 0 \\ \vdots \\ 0 \end{bmatrix}$$

By the same argument as in Theorem 19

$$\det U \neq 0$$

and the only solution is $v_j \hat{\sigma}(U_j^{-1}) = 0$ for $j=1,\dots,t$

The error values $\{v_j\}$ are nonzero so that

$$\hat{\sigma}(U_j^{-1}) = 0 \qquad j=1,\dots,t$$

The $\deg \hat{\sigma} \le t$ and $\hat{\sigma}$ has the same roots as σ,

with $\hat{\sigma}_0 = \sigma_0 = 1$ and thus $\sigma(D) = \hat{\sigma}(D)$

Q.E.D

The error location polynomial $\sigma(D)$ can be found by solving Eq's 5 for the smallest possible value of t. This can be done using standard methods for solving sets of linear equations. Berlekamp has discovered an efficient iterative algorithm for finding the solution which will be discussed shortly.

Finding Error Values

From Eq 4 $\quad A(D) = [\sigma(D)\, S(D)]_0^{d-2} = \sum_{j=1}^{t} V_j U_j^{m_0} \prod_{\substack{l=1 \\ l \neq j}}^{t} (1 - U_l D)$

The derivative of $\sigma(D) = \sigma_0 + \sigma_1 D + \cdots + \sigma_t D^t$ is defined as $\sigma'(D) = \sigma_1 + 2\sigma_2 D + \cdots + t\sigma_t D^{t-1}$. also $\sigma(D) = \prod_{j=1}^{t} (1 - U_j D)$ and it can be shown that

$$\sigma'(D) = -\sum_{j=1}^{t} U_j \prod_{\substack{l=1 \\ l \neq j}}^{t} [1 - U_l D]$$

Now $\quad A(U_k^{-1}) = V_k U_k^{m_0} \prod_{\substack{l=1 \\ l \neq k}}^{t} (1 - U_l U_k^{-1})$

and $\quad \sigma'(U_k^{-1}) = -U_k \prod_{\substack{l=1 \\ l \neq k}}^{t} (1 - U_l U_k^{-1})$

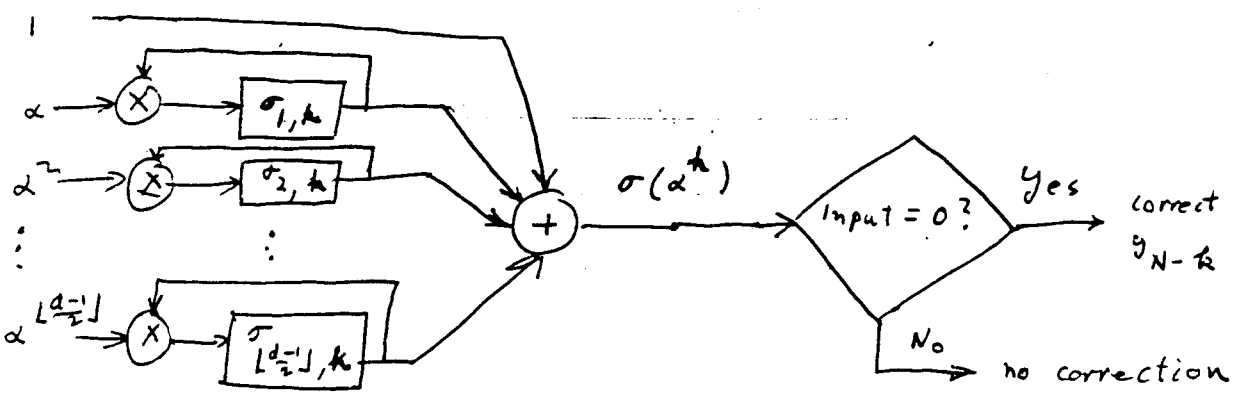
so $\quad V_k = -U_k^{1-m_0} \dfrac{A(U_k^{-1})}{\sigma'(U_k^{-1})} \qquad k = 1, \ldots, t$

or
$$\boxed{e_{n_k} = -\alpha^{n_k(1-n_0)} \frac{A(\alpha^{-n_k})}{\sigma'(\alpha^{-n_k})} \qquad k=1,\dots t} \qquad (16$$

For $n_0=1$ this becomes particularly simple.

Chien Search

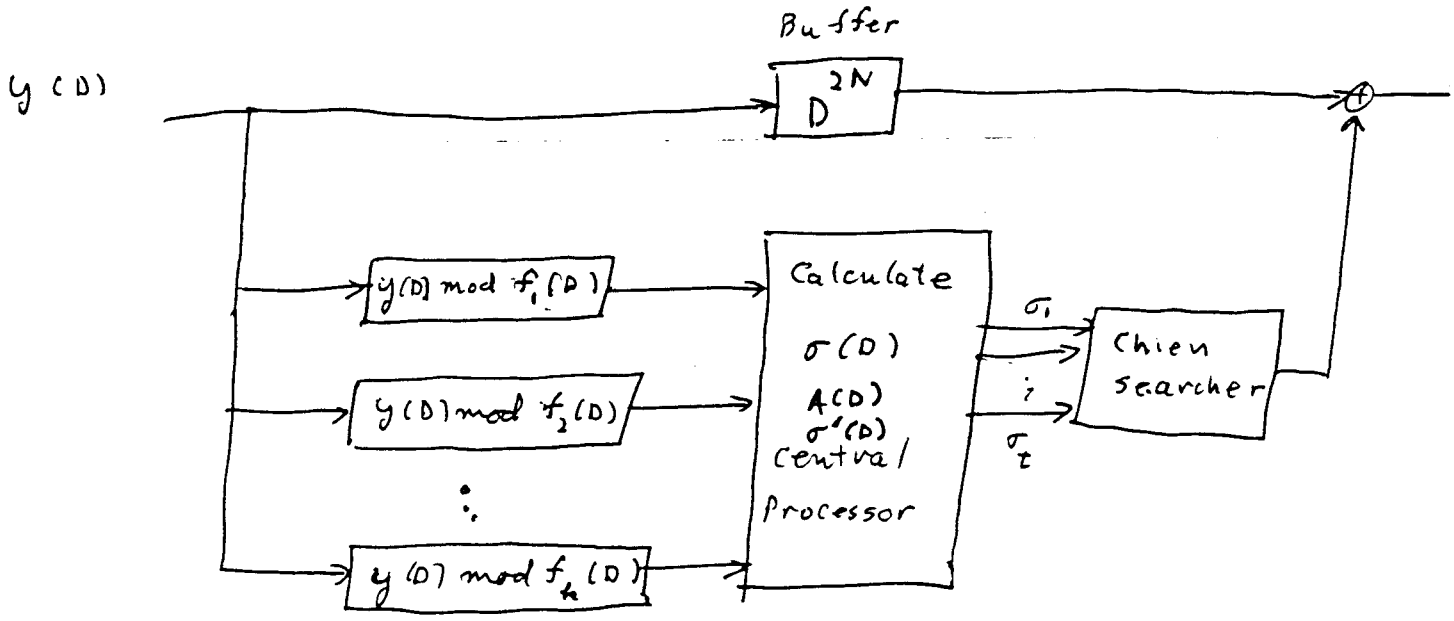
Having found $\sigma(D)$ by solving Eq's 5 for $\sigma_0, \dots, \sigma_t$ it is necessary to factor $\sigma(D)$ to find the error locations. The roots of $\sigma(D)$ can be found by substituting sequentially $\alpha, \alpha^2, \dots, \alpha^N = 1$ for D and checking to see if $\sigma(\alpha^k) = 0$. If $\sigma(\alpha^k) = 0$ then the error locator is $U_k = \alpha^{-k} = \alpha^{N-k}$ so that the code symbol corresponding to D^{N-k} is in error. For the binary case $V_{N-k} = 1$. For higher order alphabets V_{N-k} is given by Eq 6. Thus $\sigma(\alpha)$ checks for an error in y_{N-1}, $\sigma(\alpha^2)$ for an error in y_{N-2}, ..., $\sigma(\alpha^N)$ for an error in y_0. A circuit for performing the tests is shown below.



The registers are initially loaded with $\sigma_1, \dots, \sigma_{\lfloor \frac{d-1}{2} \rfloor}$ and shifted once. The contents then sum to

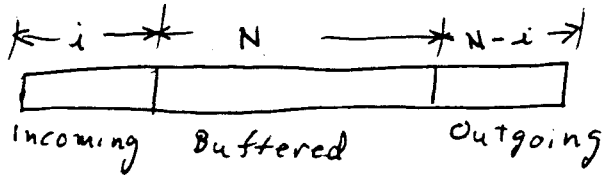
$1 + \sigma_1 \alpha + \sigma_2 \alpha^2 + \cdots + \sigma_t \alpha^t = \sigma(\alpha)$ and on the k th shift the contents sum to $\sigma(\alpha^k)$.

<u>General Decoder Block Diagram</u> (Berlekamp)



$$g(D) = \prod_{i=1}^{k} f_i(D)$$

at a typical time the buffer contains part of 3 blocks.



The Chien searcher is calculating $\sigma(\alpha^i)$ to determine if the next outgoing symbol should be corrected, The central processor is calculating $\sigma(D)$ for the buffered block, and the input registers are calculating $y(D) \bmod f_i(D)$ for the incoming word.