Poperties of extended Galois Field $GF(2^m)$

In ordinary algebra, it is very likely that an equation with real coefficient does not have real roots. For example equation x^2+x+1 has to have two roots, but neither of them are in $\mathbb{R}$. The roots of x^2+x+1 are $-\frac{1}{2} \pm j\frac{\sqrt{3}}{2}$. That is, they are from the Complex field $\mathbb{C}$.

The same way, polynomial with coefficients from $GF(2)$, may or may not have roots $\in \{0,1\}$.

For example, it is easy to see that X^4+X^3+1 over $GF(2)$ is irreducible. So it does not have roots in $GF(2)$. But, it is of degree four, so it has to have four roots. These roots are in $GF(2^4)$. For a small field like $GF(2^4)$ it is easy to try all 16 elements (in fact 14, since we know that 0 and 1 are not answers) to find four that solves the equation.

Doing this, i.e., substituting elements of $GF(2^4)$ into

the equation X^4+X^3+1 we find out that α^7, α^{11}, α^{13} and α^{14} are its roots. For example,

$$(\alpha^7)^4+(\alpha^7)^3+1=\alpha^{28}+\alpha^{21}+1=\alpha^{13}+\alpha^6+1=$$

$$=(1+\alpha^2+\alpha^3)+(\alpha^2+\alpha^3)+1=0$$

Similarly we can check α^{11}, α^{13} and α^{14}.

So,

$$X^4+X^2+1=(X+\alpha^7)(X+\alpha^{14})(X+\alpha^{13})(X+\alpha^{11})$$

The following theorem helps us to find other roots of a polynomial after finding one.

Theorem 11 Let $\beta \in GF(2^m)$ be a root of $f(X)$. Then β^{2^i}, $i \geq 0$ is also a root of $f(X)$.

We have seen that $[f(X)]^2 = f(X^2)$

So:

$$[f(\beta)]^2 = f(\beta^2)$$

Since $f(\beta)=0$, $f(\beta^2)=0$

also

$$[f(\beta^2)]^2 = f(\beta^{2^2})$$

So $f(\beta^{2^2}) = f(\beta^4)=0$

and so on. So $f(\beta^{2^i})=0$, $i \geq 0$

These elements β^{2^i} of $GF(2^m)$ are called Conjugates of β.

In the previous example after finding $\beta = \alpha^7$ as a root of $X^4 + X^3 + 1$, we can see that

$\beta^2 = \alpha^{14}$ is a root as well

$\beta^2 = \beta^4 = \alpha^{28} = \alpha^{13}$ is also a root.

and also

$\beta^3 = \beta^8 = \alpha^{56} = \alpha^{11}$.

Theorem 12: The $2^m - 1$ non-zero elements of $GF(2^m)$ form all the roots of $X^{2^m-1} + 1$.

Proof:

In theorem 8, we saw that if β is an element of $GF(q)$ then $\beta^{q-1} = 1$

So for $\beta \in GF(2^m)$ we have

$$\beta^{2^m-1} = 1 \Rightarrow \beta^{2^m-1} + 1 = 0$$

This means that β is a root of $X^{2^m-1} + 1$.

Therefore, every non-zero elements of $GF(2^m)$ is a root of $X^{2^m-1} + 1$ and since this polynomial has $2^m - 1$ roots, the $2^m - 1$ non-zero elements of $GF(2^m)$ form all the roots of $X^{2^m-1} + 1$.

Corollary 12.1

The elements of $GF(2^m)$ form all the roots of $X^{2^m} + X$.

Proof: This polynomial factors as $X[X^{2^m-1} + 1]$

It has a root of zero and all non-zero elements of $GF(2^m)$ as its roots.

While an element β^Y over $GF(2^m)$ is always a root of $X^{2^m-1} + 1$, it may also be a root of a polynomial over $GF(2)$ with degree less than $2^m - 1$. Take $m = 4$, i.e., $GF(2^4)$.

$$X^{2^m-1} + 1 = X^{15} + 1.$$

We can write

$$X^{15} + 1 = (X^4 + X^3 + 1)(X^{11} + X^{10} + X^9 + X^8 + X^6 + X^4 + X^3 + 1)$$

We saw that $\beta = \alpha^7$ is a root of $X^4 + X^3 + 1$

Definition: For any $\beta \in GF(2^m)$ the polynomial $\Phi(X)$ with lowest degree that has β as its root is called the minimal polynomial of β.

Theorem 13: The minimal polynomial $\Phi(X)$ of a field element β is irreducible.

Proof: Suppose $\Phi(X)$ is not irreducible and can be written as $\Phi(X) = \Phi_1(X)\,\Phi_2(X)$.

Since $\Phi(\beta) = \Phi_1(\beta)\,\Phi_2(\beta) = 0$

then either $\Phi_1(\beta) = 0$ or $\Phi_2(\beta) = 0$.

This contradicts the definition that $\Phi(X)$ is the smallest degree polynomial with β as a root.

Theorem 14: If a polynomial $f(X)$ over GF(2) has β as a root, then $\Phi(X)$ divides $f(X)$.

Proof: Suppose $f(X)$ is not divisible by $\Phi(X)$.

Then

$$f(X) = \Phi(X) \cdot a(X) + r(X)$$

with $r(X)$ having degree less than $\Phi(X)$.

But,

$$f(\beta) = \Phi(\beta) \cdot a(\beta) + r(\beta)$$

$f(\beta) = 0$ and $\Phi(\beta) = 0 \Rightarrow r(\beta) = 0$

$\Rightarrow$ Contradiction.

Following properties are simple to prove:

Theorem 15: The minimal polynomial $\Phi(X)$ of $\beta \in GF(2^m)$ divides $X^{2^m} + X$.

Theorem 16: If $f(X)$ is an irreducible polynomial and $f(\beta) = 0$ then $f(X) = \Phi(X)$.

In a previous example, we saw that $\alpha^7, \alpha^{11}, \alpha^{13}$ and α^{14} are roots of $f(X) = X^4 + X^3 + 1$. That is

$$X^4 + X^3 + 1 = (X+\alpha^7)(X+\alpha^{11})(X+\alpha^{13})(X+\alpha^{14}).$$

Note that if we take $\beta = \alpha^7$, we have $\beta^2 = \alpha^{14}$, $\beta^4 = \alpha^{28} = \alpha^{13}$ and $\beta^8 = \alpha^{11}$, $\beta^{16} = \beta = \alpha^7$

That is:

$$X^4 + X^3 + 1 = (X+\beta)(X+\beta^2)(X+\beta^4)(X+\beta^8)$$

Following theorem relates to this observation.

Theorem 17: For $\beta \in GF(2^m)$ if e is the smalles number such that $\beta^{2^e} = \beta$. Then

$$f(X) = \prod_{i=0}^{e-1} (X + \beta^{2^i})$$

is an irreducible polynomial over $GF(2)$.

Proof: First we show that $f(X)$ is a polynomial over $GF(2)$.

$$[f(X)]^2 = \left[\prod_{i=0}^{e-1}(X+\beta^{2^i})\right]^2 = \prod_{i=0}^{e-1}(X+\beta^{2^i})^2$$

But

$$(X+\beta^{2^i})^2 = X^2 + \beta^{2^i}X + \beta^{2^i}X + \beta^{2^{i+1}}$$
$$= X^2 + (\beta^{2^i}+\beta^{2^i})X + \beta^{2^{i+1}}$$
$$= X^2 + \beta^{2^{i+1}}$$

So:

$$[f(X)]^2 = \prod_{i=0}^{e-1}(X^2+\beta^{2^{i+1}}) = \prod_{i=1}^{e}(X^2+\beta^{2^i})$$
$$= \prod_{i=1}^{e-1}(X^2+\beta^{2^i})(X^2+\beta^{2^e})$$
$$= \prod_{i=1}^{e-1}(X^2+\beta^{2^i})(X^2+\beta)$$
$$= \prod_{i=0}^{e-1}(X^2+\beta^{2^i}) = f(X^2)$$

Let $f(X) = f_0 + f_1X + \cdots + f_eX^e$

Then $f(X^2) = f_0 + f_1X^2 + \cdots + f_eX^{2e}$

and

$$[f(X)]^2 = (f_0 + f_1X + \cdots + f_eX^e)^2$$
$$= \sum_{i=0}^{e} f_i^2X^{2i} + (1+1)\sum_{i=0}^{e}\sum_{j=0}^{e}{}^{i\neq j} f_if_jX^{i+j} = \sum f_i^2X^{2i}$$

So $f(X^2) = [f(X)]^2 \Rightarrow f_i^2 = f_i$ all i

This means that $f_i = 0$ or $f_i = 1$ all i.

Therefore $f(X)$ is a polynomial over GF(2).

Now we show that if we assume $f(X)$ is not irreducible we arrive at a contradiction.

Let $f(X)$ not be irreducible and can be written as $f(X) = a(X)\,b(X)$.

Since $f(\beta) = 0$ either $a(\beta) = 0$ or $b(\beta) = 0$.

If $a(\beta) = 0$ then $a(X)$ has β, as well as $\beta^2, \ldots \beta^{2^{e-1}}$ as its roots. So it has degree e and $a(X) = f(X)$. Similarly for $b(X)$. So $f(X)$ must be irreducible.

Def.: $\beta^2, \ldots, \beta^{2^{e-1}}$ are called Conjugates of β

Theorem 15: Let $\Phi(X)$ be the minimal polynomial of $\beta \in GF(2^m)$. Let e be the smallest non-negative integer such that $\beta^{2^e} = \beta$ then:

$$\Phi(X) = \prod_{i=0}^{e-1} (X + \beta^{2^i})$$

Example: Consider Galois Field $GF(2^4)$ and let $\beta = \alpha^3$. The conjugates of α^3 are

$\beta^2 = \alpha^3$, $\beta^{2^2} = \beta^4 = \alpha^{12}$, $\beta^{2^3} = \alpha^{24} = \alpha^9$

So $\Phi(X)$ for $\beta = \alpha^3$ is

$$\Phi(X) = (X+\alpha^3)(X+\alpha^6)(X+\alpha^{12})(X+\alpha^9)$$
$$= X^4 + X^3 + X^2 + X + 1$$

Consider $GF(2^4)$ generated by $p(X) = X^4 + X + 1$

Following is a list of minimal polynomials

Conjugate Roots	$\Phi(X)$
0	X
1	$X+1$
$\alpha, \alpha^2, \alpha^4, \alpha^8$	X^4+X+1
$\alpha^3, \alpha^6, \alpha^9, \alpha^{12}$	$X^4+X^3+X^2+X+1$
α^5, α^{10}	X^2+X+1
$\alpha^7, \alpha^{11}, \alpha^{13}, \alpha^{14}$	X^4+X^3+1

Theorem 19: Let $\Phi(X)$ be the minimal polynomial of $\beta \in GF(2^m)$ and the degree of $\Phi(X)$ is e. Then e is the smallest integer such that $\beta^{2^e} = \beta$. Also $e \leq m$.

Theorem 20: If β is a primitive element of $GF(2^m)$, then $\beta^2, \ldots \beta^{2^i}, \ldots$ (its conjugates) are also primitive elements of $GF(2^m)$.

Theorem 21: All conjugates of $\beta \in GF(2^m)$ have the same order.

Vector Spaces

Let V be a set of elements on which an operation called addition $+$, is defined. Let F be a field. A multiplication, $\cdot$ operation between elements of V and F is defined. The set V is called a vector space over F if the following conditions hold:

i) V is a commutative group under addition.

ii) For any element $a \in F$ and any $\underline{v} \in V$:

$$a \cdot \underline{v} \in V.$$

iii) Distributive law:

$\forall\, a, b \in F$ and $\forall\, \underline{u}, \underline{v} \in V$

$$a \cdot (\underline{u} + \underline{v}) = a \cdot \underline{u} + a \cdot \underline{v}$$

and

$$(a+b)\underline{v} = a \cdot \underline{v} + b \cdot \underline{v}$$

iv) Associative law:

$$(a \cdot b) \cdot \underline{v} = a \cdot (b \cdot \underline{v})$$

v) Let 1 be the unit element of F. Then

$\forall \underline{v} \in V \Rightarrow 1 \cdot \underline{v} = \underline{v}$.

The elements of V are called Vectors.

The elements of the field F are called Scalars.

The addition between elements of V is called Vector addition.

The multiplication between elements of F and V is called Scalar multiplication

Properties of the vector field:

Property I $\forall \underline{v} \in V \Rightarrow 0 \cdot \underline{v} = 0$ where 0 is the zero element of F.

Property II: $\forall c \in F \Rightarrow c \cdot \underline{0} = \underline{0}$ where $\underline{0}$ is the zero element of V.

Property III: $\forall c \in F$ and $\forall \underline{v} \in V$, we have

$(-c) \cdot \underline{v} = c \cdot (-\underline{v}) = -(c \cdot \underline{v})$.

Definition: A subset of a vector space V say S is called a **subspace** if it is also a vector space.

Theorem 22: Let $S \subset V$. Where V is a vector space over F. The S is a subspace of V if:

i) $\forall \underline{u}, \underline{v} \in S$, $\underline{u} + \underline{v} \in S$.

ii) $\forall a \in F$ and $\underline{u} \in S \Rightarrow a \cdot \underline{u} \in S$

---

n-tuples of GF(2) elements as a vector space.

Take $\underline{v} = (v_0, v_1, \cdots v_{n-1})$ where $v_i \in GF(2)$

Define:

$$\underline{v} + \underline{u} = (v_0 + u_0, v_1 + u_1, \cdots v_{n-1} + u_{n-1})$$

where addition is modulo-2.

Also for $a \in GF(2)$ define

$$a \cdot \underline{v} = (a \cdot v_0, a \cdot v_1, \cdots, a \cdot v_{n-1})$$

where multiplication is modulo-2.

Let $\underline{v}_1, \underline{v}_2, \cdots \underline{v}_k$ be k vectors $\in V$ and $a_1, a_2, \cdots a_k \in F$. Then

$$a_1\underline{v}_1 + a_2\underline{v}_2 + \cdots a_k\underline{v}_k$$

is called a linear combination of $\underline{v}_1, \underline{v}_2, \cdots \underline{v}_k$.

It is clear that sum of two linear combinations of $\underline{v}_1, \cdots, \underline{v}_k$ is a linear combinations of $\underline{v}_1, \cdots \underline{v}_k$.

Also $c.(a_1\underline{v}_1 + \cdots a_k\underline{v}_k)$ is a linear combination of $\underline{v}_1, \cdots \underline{v}_k$.

So,

Theorem 23: The set of all linear combination of $\underline{v}_1, \cdots \underline{v}_k \in V$ is a subspace of V.

Definition: $\underline{v}_1, \cdots \underline{v}_k \in V$ are linearly dependent if, there are k scalars $a_1, \cdots a_k \in F$ such that:

$$a_1\underline{v}_1 + a_2\underline{v}_2 + \cdots a_k\underline{v}_k = \underline{0}$$

A set of vectors $\underline{v}_1, \cdots \underline{v}_k \in V$ are linearly independent if they are not linearly dependent.

Consider:

$$\underline{e}_0 = (1, 0, \cdots 0)$$

$$\underline{e}_1 = (0, 1, 0 \cdots 0)$$

$$\vdots$$

$$\underline{e}_{n-1} = (0, 0, \cdots 1)$$

These n-tuples Span the vector space V of all 2^n n-tuples.

Each n-tuple $(a_0, a_1, \cdots a_{n-1})$ is written as

$$(a_0, a_1, \cdots a_{n-1}) = a_0 \underline{e}_0 + a_1 \underline{e}_1 + \cdots a_{n-1} \underline{e}_{n-1}$$

We call $\underline{u} \cdot \underline{v} = u_0 v_0 + u_1 v_1 + \cdots u_{n-1} v_{n-1}$ the inner product of $\underline{u}$ and $\underline{v}$.

If $\underline{u} \cdot \underline{v} = 0$ we say that $\underline{u}$ an $\underline{v}$ are orthogonal.

Let S be a subspace of V. Let the subset S_d of V be the set of all vector $\underline{u}$ of S and any vector $\underline{v} \in S_d$ we have $\underline{u} \cdot \underline{v} = 0$. S_d is called the null-space of S.

Theorem 24: Let S be a k-dimensional subspace of V_n (set of n-tuples over $GF(2)$). The dimension of S_d, the null-space of S is $n-k$.

Linear Block Codes

In a digital communication system the sequence of bits to be transmitted are arranged as blocks of k bits. So there are 2^k possible k-tuples to be transmitted. In a block code, the encoder assigns n bits to each k-tuple where $n > k$.

For a block code to be useful we require that all of 2^k, n-tuples (called codewords) be distinct. That is there should be a 1-to-1 correspondence between the input $\underline{u}$ and the output $\underline{v}$ of the encoder.

Unless the codewords are structured according to a certain structure, the encoding (and obviously decoding) will be prohibitively complex. That is why we are interested in linear block codes.

A code is linear if a linear combination of any two of its codewords is a codeword, or equivalently:

Definition: A block code of length n and 2^k codewords is an (n, k) linear code if and only if

its 2^k codewords form a k-dimensional subspace of the vector space of n-tuples over GF(2

A linear (n, k) code C is a k-dimensional subspace of all the binary n-tuples (V_n). So, we can find k linearly independent members of C, say $\underline{g}_0, \underline{g}_1, \cdots \underline{g}_{k-1}$ such that any $\underline{v} \in V$ can be written as:

$$\underline{v} = u_0 \underline{g}_0 + u_1 \underline{g}_1 + \cdots + u_{k-1} \underline{g}_{k-1}$$

Arranging these k linearly independent in a matrix:

$$G = \begin{bmatrix} \underline{g}_0 \\ \underline{g}_1 \\ \vdots \\ \underline{g}_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & \cdots & g_{0,n-1} \\ g_{10} & g_{11} & \cdots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \cdots & g_{k-1,n-1} \end{bmatrix}$$

where G is a $k \times n$, binary matrix.

Let $\underline{u} = (u_0, u_1, \cdots u_{k-1})$ be the message to be sent. Then, the codeword can be given as:

$$\underline{v} = \underline{u} \cdot G = (u_0, u_1, \cdots, u_{k-1}) \begin{bmatrix} \underline{g}_0 \\ \underline{g}_1 \\ \vdots \\ \underline{g}_{k-1} \end{bmatrix}$$

$$= u_0 \underline{g}_0 + u_1 \underline{g}_1 + \cdots + u_{k-1} \underline{g}_{k-1}.$$

That is, rows of G, span or generate C.

That is why G is called the generator matrix.

Example (Hamming Code)

Consider (7,4) code we saw before:

$$G = \begin{bmatrix} \underline{g}_0 \\ \underline{g}_1 \\ \underline{g}_2 \\ \underline{g}_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Let's message be $\underline{u} = (1101)$. Then

$$\underline{v} = 1 \cdot \underline{g}_0 + 1 \cdot \underline{g}_1 + 0 \cdot \underline{g}_2 + 1 \cdot \underline{g}_3$$

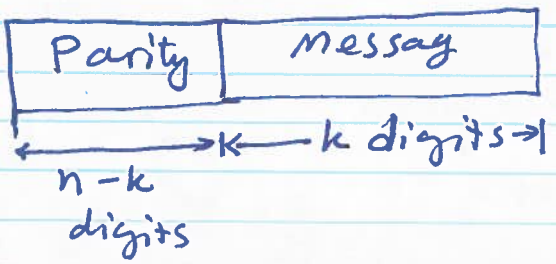
$$= (1101000) + (0110100) + (1010001)$$

$$= (0001101)$$

Example: (7,4) linear block code

Message	Codeword
0000	0000000
1000	1101000
0100	0110100
1100	1011100
0010	1110010
1010	0011010
0110	1000110
1110	0101110
0001	1010001
1001	0111001
0101	1100101
1101	0001101
0011	0100011
1011	1001011
0111	0010111
1111	1111111

Definition: A block code is called **Systematic** if its message bits are consecutive and so are its parity bits.



The generator of a systematic code consists of a $k \times k$ identity matrix (to repeat the message bits) and a $k \times (n-k)$ parity matrix to generate parity bits

$$G = \begin{bmatrix} \underline{g}_0 \\ \underline{g}_1 \\ \vdots \\ \underline{g}_{k-1} \end{bmatrix} = \left[\begin{array}{ccc:ccc} p_{00} & p_{01} & \cdots & p_{0,n-k-1} & 1 \; 0 \cdots & 0 \\ p_{10} & p_{11} & \cdots & p_{1,n-k-1} & 0 \; 1 \cdots & 0 \\ \vdots & & & \vdots & & \\ p_{k-1,0} & p_{k-1,1} & \cdots & p_{k-1,n-k-1} & 0 \; 0 \cdots & 1 \end{array}\right]$$

So $\underline{G} = [P : I_k]$

For an input $\underline{u} = (u_0, u_1, \ldots, u_{k-1})$, the output of the encoder is:

$$\underline{v} = (v_0, \ldots v_{n-1}) = (u_0, u_1, \ldots u_{k-1})G$$

So $v_i = u_0 p_{0i} + u_1 p_{1i} + \cdots + u_{k-1} p_{k-1,i}$

for $0 \le i < n-k$

and $v_{n-k+i} = u_i$ for $0 \le i < k$

Going back to our (7,4) example

$$\underline{v} = (u_0, u_1, u_2, u_3) \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

So:

$$v_0 = u_0 + u_2 + u_3$$

$$v_1 = u_0 + u_1 + u_2$$

$$v_2 = u_1 + u_2 + u_3$$

$$v_3 = u_0$$

$$v_4 = u_1$$

$$v_5 = u_2$$

$$v_6 = u_3$$

Parity check matrix

Let G be the generating polynomial of a code C. Form an $(n-k) \times n$ matrix H whose rows are orthogonal to all rows of G. For a systematic code $G = [P \,|\, I_k]$ and $H = [I_{n-k} \,|\, P^T]$ where P^T is the transpose of P. That is:

$$H = [I_{n-k} \vdots P^T] = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 & p_{00} & \cdots & p_{k-1,0} \\ 0 & 1 & 0 & \cdots & 0 & p_{01} & p_{11} & p_{k-1,1} \\ \vdots & & & & & & \vdots & \\ 0 & 0 & \cdots & & 1 & p_{0,n-k-1} & \cdots & p_{k-1,n-k-1} \end{bmatrix}$$

Then we have

$$G \cdot H^T = \underline{0}$$

Therefore:

For any $\underline{v} \in C \Rightarrow \underline{v} = u \cdot G \cdot H^T = \underline{0}$

For the (7,4) Hamming code:

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

Note that a parity check matrix can generate an $(n, n-k)$ code. Each codeword of this code, C_d is orthogonal to each codewor of C.

C_d is called the dual code of C.

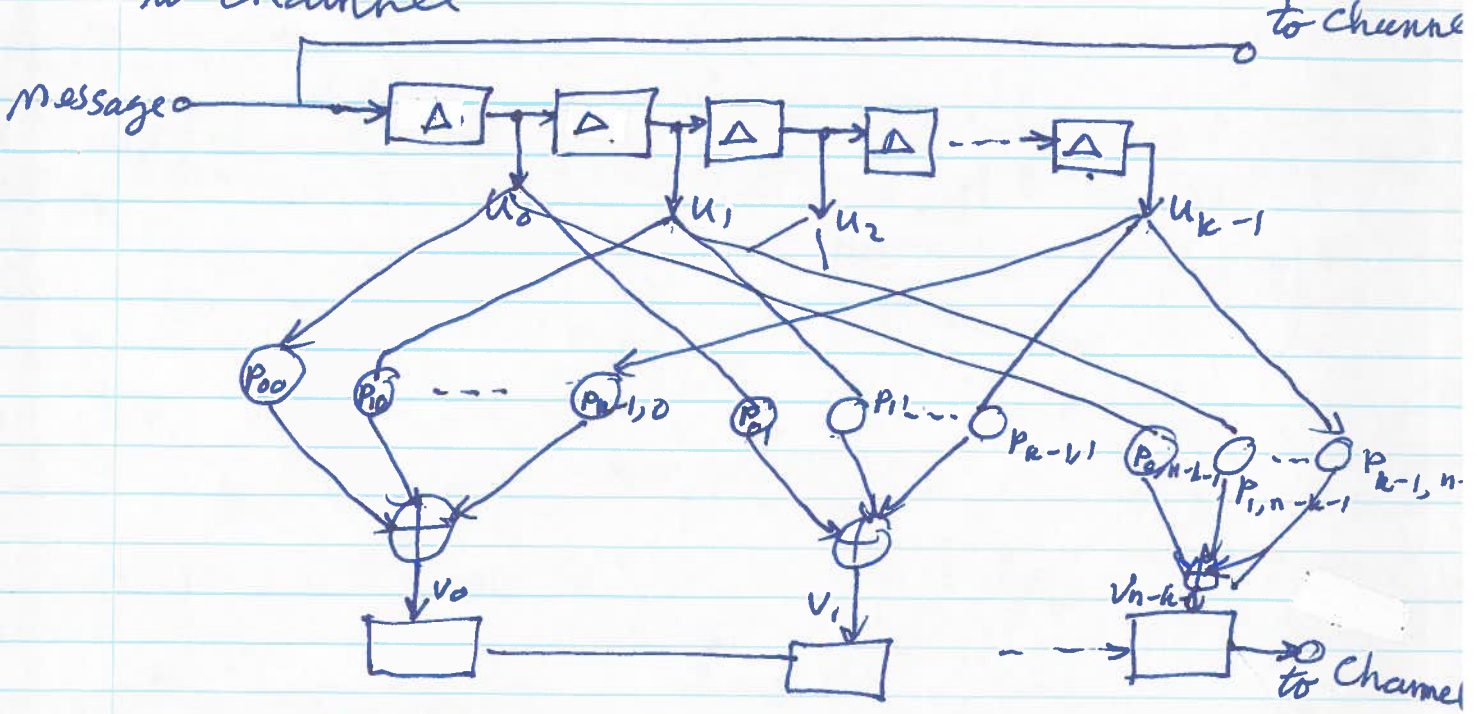
To encode a linear block code, we use XOR gates to form parities.

Following figure shows how a systematic linear block code is encoded:

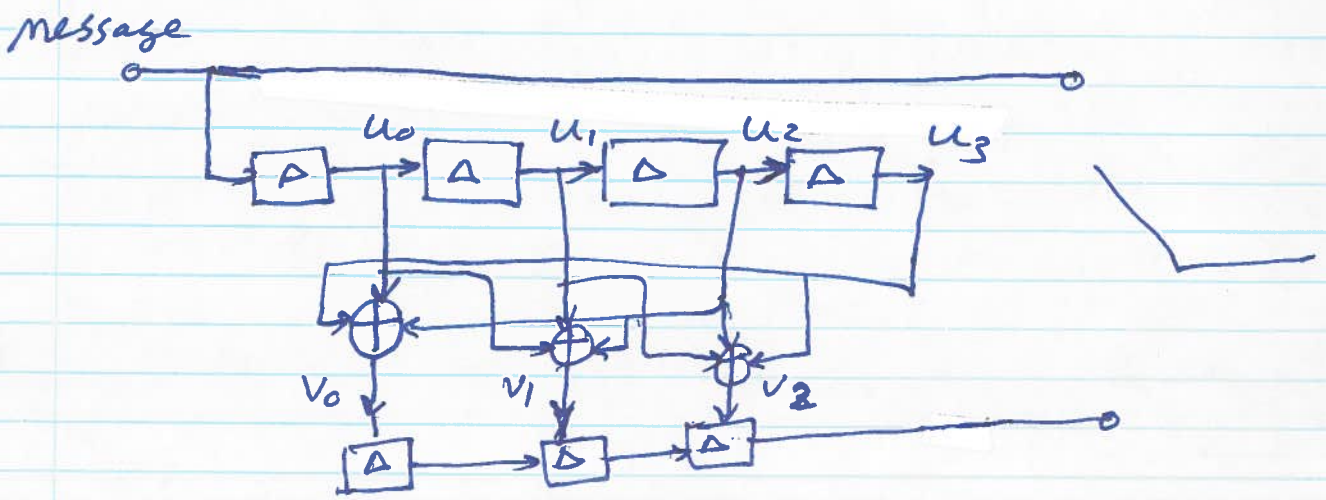
Bits of the message are fed to a shift register and also go to the channel. When they are in the shift register, they are linearly combined according to

$$V_i = P_{0i} u_0 + u_1 P_{1i} + \cdots + u_{k-1} P_{k-1,i}$$

and placed in an output register and fed to channel



For the (7, 4) Code



Syndrome

Assume that the message $\underline{u}$ is encoded as $\underline{v} = \underline{u} \cdot G$. If there is no error, at the receiver we have $\underline{r} = \underline{v}$ and no need for error detection and error correction.

But if there is an error, we get

$$\underline{r} = \underline{v} + \underline{e},$$

where $\underline{e} = (e_0, e_1, \cdots, e_n)$ is an error vector.

If we multiply $\underline{r}$ by H^T, we get

$$\underline{r} \cdot H^T = (\underline{v} + \underline{e}) \cdot H^T = \underline{v} \cdot H^T + \underline{e} \cdot H^T = \underline{e} \cdot H^T$$

It is important to note that the result does not depend on the messag, but on the error pattern $\underline{e}$. We call the vector $\underline{S} = \underline{r} \cdot H^T$ the Syndrome. Since $\underline{r}$ is an n-vector and H^T is $n \times (n-k)$, there are $(n-k)$ bits in vector $\underline{S}$. So, $\underline{S}$ can point to 2^{n-k} patterns (One correct transmission $0,0,\cdots,0$ and $2^{n-k}-1$ error patterns).

Example: Consider the (7, 4) code

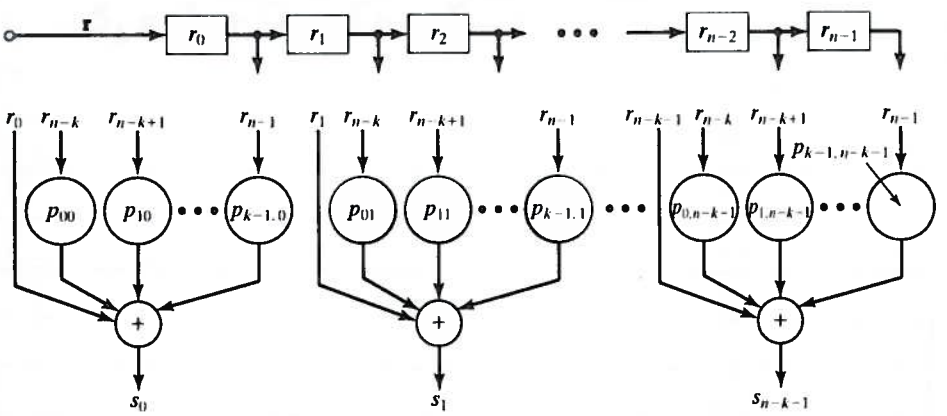
Let $\underline{r} = (r_0, r_1, r_2, r_3, r_4, r_5, r_6)$ be the received vector (output of demodulator). Then the syndrome

$$\underline{S} = (S_0, S_1, S_2) = (r_0, r_1, r_2, r_3, r_4, r_5, r_6) \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

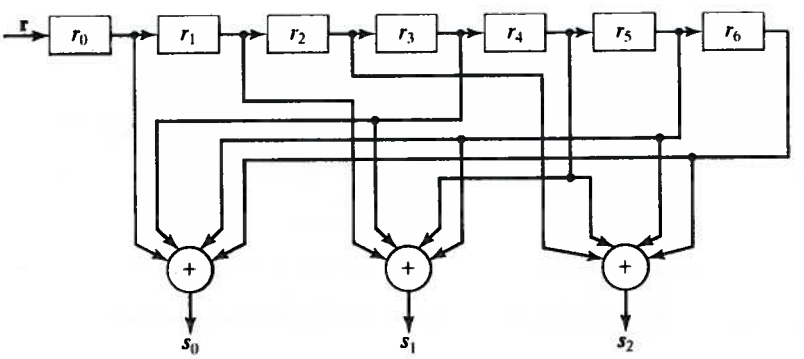
or:

$$S_0 = r_0 + r_3 + r_5 + r_6$$

$$S_1 = r_1 + r_3 + r_4 + r_5$$



Syndrome circuit for a linear systematic (n, k) code.



Syndrome circuit for the (7, 4) code

We saw that:

$$\underline{S} = \underline{r} \cdot H^T = \underline{e} \cdot H^T$$

So: we can write s_i's as

$$s_i = r_i + r_{n-k} p_{0i} + r_{n-k+1} p_{1i} + \cdots r_{n-1} p_{k-1,i}$$

$$i = 0, 1, \ldots, n-k-1$$

since $\underline{r} = \underline{v} + \underline{e}$,

we have:

$$s_i = (v_i + e_i) + (v_{n-k} + e_{n-k}) p_{0i} + \cdots (v_{n-1} + e_{n-1}) p_{k-1,i}$$

But

$$v_i + v_{n-k} p_{0i} + \cdots + v_{n-1} p_{k-1,i} = 0$$

and

$$s_i = e_i + e_{n-k} p_{0i} + e_{n-k+1} p_{1i} + \cdots e_{n-1} p_{k-1,i}$$

$$i = 0, 1, \ldots, n-k-1$$

This shows that $n-k$ syndromes provide us with $n-k$ equations about error pattern. There are 2^n error patterns, but we have 2^{n-k} equations. So, we cannot catch all errors.

In fact there are 2^k error patterns for each syndrome. To put it another way, the code C is a subgroup of the set of n-tuples. The set of

n-tuples is partitioned into 2^{n-k} cosets of C. All the n-tuples in one coset result in the same syndrome. So,

the syndrome only points us to a coset of C not to a single error pattern. Out of 2^k patterns (n-tuples in the coset), we decide (based on the property of the channel) which error has occured.

Example: Take again the (7,4) code.

Assume that we receive $\underline{r} = (1001001)$. Then,

$$\underline{s} = \underline{r} \cdot H^T = (1,1,1)$$

This means that

$$1 = e_0 + e_3 + e_5 + e_6$$
$$1 = e_1 + e_3 + e_4 + e_5$$
$$1 = e_2 + e_4 + e_5 + e_6$$

Any of the following $2^4 = 16$ patterns satisfy these equations:

(0000010),	(1010011),
(1101010),	(0111011),
(0110110),	(1100111),
(1011110),	(0001111),
(1110000),	(0100001),
(0011000),	(1001001),
(1000100),	(0010101),
(0101100).	(1111101).

To decide which error to choose depends on our

expectation about the channel behaviours. For example, in a BSC channel, we know that the probability of a single error is more than multiple errors. So, we decide:

$$\underline{e} = (0000010)$$

as the error and, therefore, the codeword transmitted must have been:

$$\underline{v} = \underline{r} + \underline{e} = (1001001) + (0000010) = (1001011)$$

Minimum Distance of a Code

Hamming distance $d(\underline{v}, \underline{w})$ between two vectors $\underline{v}$ and $\underline{w}$ is the number of places they are different. In binary case, the distance $d(\underline{v}, \underline{w})$ is the weight (the number of places a vector is non-zero) of $\underline{v} + \underline{w}$ or

$$d(\underline{v}, \underline{w}) = w(\underline{v}, \underline{w})$$

The minimum distance of a code C is the minimum value of $d(\underline{v}, \underline{w})$ for all non-identical $\underline{v}$ and $\underline{w} \in C$

$$d_{min} = \{\min\{d(\underline{v}, \underline{w}) : \underline{v}, \underline{w} \in C, \underline{v} \neq \underline{w}\}$$

Since for any $\underline{v}$ and $\underline{w} \in C$, $\underline{v}+\underline{w} \in C$ then the minimum distance of a linear block code is equal to minimum weight of its non-zero codewords:

$$d_{min} = \min\{w(\underline{v}+\underline{w}) : \underline{v}, \underline{w} \in C, \underline{v} \neq \underline{w}\}$$

$$= \min\{w(\underline{x}) : x \in C, \underline{x} \neq \underline{0}\}$$

$$= w_{min}.$$

SO;

Theorem 1: The minimum distance of a linear block code is equal to the minimum weight of its non-zero codewords.

Theorem 2: Let C be an (n,k) linear block code with parity check matrix H.

- For any codeword $\underline{v} \in C$ of weight l there are l columns of H such that their vector sum is $\underline{0}$.
- If there are l columns of H whose vector sum is $\underline{0}$, then there is a codeword $\underline{v} \in C$ with weight l.

Proof: Let $\underline{v} = (v_0, v_1, \cdots v_{n-1})$ have ℓ non-zero elements at places $i_1, i_2, \cdots i_\ell$. Then

$$\underline{v} \cdot H^T = \underline{0} \Rightarrow v_0 \underline{h}_0 + v_1 \underline{h}_1 + \cdots + v_{n-1} \underline{h}_{n-1} = \underline{0}$$

$$\Rightarrow v_{i_1} \underline{h}_{i_1} + v_{i_2} \underline{h}_{i_2} + \cdots + v_{i_\ell} \underline{h}_{i_\ell} = \underline{0}$$

$$\Rightarrow \underline{h}_{i_1} + \underline{h}_{i_2} + \underline{h}_{i_3} + \cdots + \underline{h}_{i_\ell} = \underline{0}$$

So, part 1 is proved.

Now assume that:

$$\underline{h}_{i_1} + \underline{h}_{i_2} + \cdots + \underline{h}_{i_\ell} = 0$$

Take $\underline{x} = (x_0, x_1, \cdots x_{n-1})$ such that

$$\begin{cases} x_j = 1 & \text{at } j = i_1, i_2, \cdots i_\ell \\ x_j = 0 & \text{otherwise.} \end{cases}$$

Then $\underline{x} \cdot H^T = x_0 \underline{h}_0 + x_1 \underline{h}_1 + \cdots + x_{n-1} \underline{h}_{n-1}$

$$= x_{i_1} \underline{h}_{i_1} + x_{i_2} \underline{h}_{i_2} + \cdots + x_{i_\ell} \underline{h}_{i_\ell}$$

$$= \underline{h}_{i_1} + \underline{h}_{i_2} + \cdots + \underline{h}_{i_\ell} = \underline{0} \quad \text{So } \underline{x} \in C$$

Corollary 2.1: Let C be a linear block code with parity check matrix H. If no $d-1$ or less columns of H add to $\underline{0}$, then minimum weight of H is at least d.

Corollary 2.2: The minimum distance of a linear

block code C is the smallest number of columns of H adding to $\underline{0}$.

Error-Detection and Error-Correction Capability of a linear block code.

If the minimum distance of a code is d_{min}, it can detect any error pattern with $d_{min}-1$ or less errors.

Definition: Assume that $A_0, A_1, A_2, \ldots A_n$ are the number of codewords with weight $0, 1, \ldots, n$ in a code C. $A_0, A_1, \ldots, A_n$ are called weight distribution of the code.

For example, for (7,4) Hamming code,

$A_0 = A_7 = 1$, $A_3 = 7$, $A_4 = 7$ and $A_i = 0$ otherwise.

If we send a codeword $\underline{v}$ and we receive $\underline{r} = \underline{v} + \underline{e}$. We can detect errors unless $\underline{e} \in C$.

So, $$P_u(E) = \sum_{i=1}^{n} A_i (1-p)^{n-i} p^i$$

where $P_u(E)$ is the probability of undetected error and p is the probability of error of modulation.

demodulation.

For the (7,4) code, we have

$$P_u(E) = 7p^3(1-p)^4 + 7p^4(1-p)^3 + p^7$$

So, if $p = 10^{-2}$, we get $P_u(E) = 7 \times 10^{-6}$.

That is if one million bits are transmitted on the average 7 errors go through undetected.

Error Correction Capability:

A code C with minimum distance d_{min} can correct $t = \lfloor \frac{d_{min}-1}{2} \rfloor$ and less errors.

$\lfloor i \rfloor$ denote the floor, i.e., the largest number less than i.

$t = \lfloor \frac{d_{min}-1}{2} \rfloor$ means that $d_{min} = 2t+1$ or $d_{min} = 2t+2$ or $2t+1 \leq d_{min} < 2t+2$

$d(\underline{V}, \underline{r}) + d(\underline{w}, \underline{r}) \geq d(\underline{V}, \underline{w})$ triangle inequality

But $d(\underline{V}, \underline{w}) \geq d_{min} \geq 2t+1$

Let $d(\underline{V}, \underline{r}) = t'$, then

$d(\underline{w}, \underline{r}) \geq 2t+1-t'$

If $t' \leq t$ then $d(\underline{w}, \underline{r}) \geq t$.

This means if the distance between the received vector

and the transmitted code is less that or equal to t, the received vector is closer to this codeword, say $\underline{v}$, than any other codeword $\underline{w}$.

A code C with minimum distance d_{min} can correct $t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ errors. It may correct some of the error patterns of weight higher than t, but it cannot correct all of those with $t+1$ errors.

Probability of error is upper bounded as

$$P(E) \leq \sum_{i=t+1}^{n} \binom{n}{i} p^i (1-p)^{n-i}$$

Erasures: Sometimes instead of deciding 0 or 1 at the output of the demodulator, we decide 0 and 1 for those received values far away zero and e or erasure for those close to zero.

A linear block code with d_{min} can correct v errors and e erasures such that:

$$\boxed{d_{min} \geq 2v + e + 1}$$

Standard Arrays

We said that a code of length n and dimension k, i.e., and (n, k) code partitions the set V_n of n-tuples into 2^{n-k} cosets of the code C. If we write elements of C in a row and then from $2^n - 2^k$ remaining n-tuples take a vecte $\underline{e}_2$, add $\underline{e}_2$ to each element of C and write in the second row, then take an unused element of the n-tuples say $\underline{e}_3$, add it to each codeword and write in the second row and continue this until, we have used all n-tuples, we get a standard array.

$$
\begin{array}{llll}
\underline{v}_1 = 0 & \underline{v}_2 & \cdots & \underline{v}_i \cdots \underline{v}_{2^k} \\
\underline{e}_2 & \underline{e}_2 + \underline{v}_2 & \cdots & \underline{e}_2 + \underline{v}_i \cdots \underline{e}_2 + \underline{v}_{2^k} \\
\underline{e}_3 & \underline{e}_3 + \underline{v}_2 & \cdots & \underline{e}_3 + \underline{v}_i \cdots \underline{e}_3 + \underline{v}_{2^k} \\
\vdots & & & \\
\underline{e}_\ell & \underline{e}_\ell + \underline{v}_2 & \cdots & \underline{e}_\ell + \underline{v}_i \cdots \underline{e}_\ell + \underline{v}_{2^k} \\
\vdots & \vdots & & \vdots \\
\underline{e}_{2^{n-k}} & \underline{e}_{2^{n-k}} + \underline{v}_2 & \cdots & \underline{e}_{2^{n-k}} + \underline{v}_i \cdots \underline{e}_{2^{n-k}} + \underline{v}_{2^k}
\end{array}
$$

Theorem 3: No two n-tuples in the same row of identical. Every n-tuple is in one and only one row.

Proof: Since C is a subgroup of V_n and each row is a coset of C.

Since a code C with minimum distance d_{min} can correct upto $t = \left\lfloor \frac{d_{min}-1}{2} \right\rfloor$ errors, we can use as the first coset leaders (e_i's) the patterns with t and less 1's. This covers for:

$$\binom{n}{0} + \binom{n}{1} + \cdots \binom{n}{t} = \sum_{i=0}^{t} \binom{n}{i}$$

coset leaders, but this sum may not be equal to 2^{n-k} so, we may add some error patterns with two or more errors.

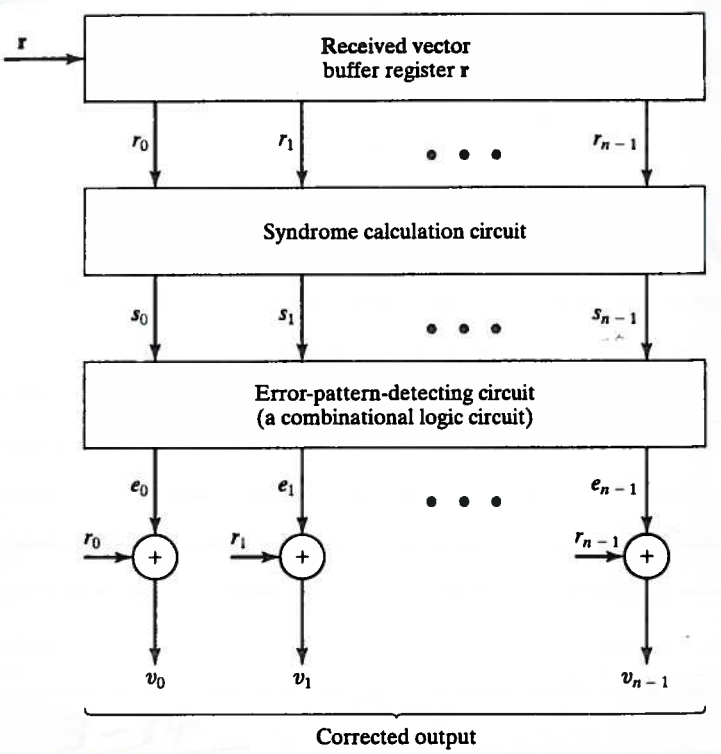
Definition if $\sum_{i=0}^{t} \binom{n}{i} = 2^{n-k}$, we say that the (n, k) code is perfect.

(7,4) code is perfect since it has $d_{min} = 3$ and, therefore $t = 1$ and

$$\sum_{i=0}^{t} \binom{n}{i} = \binom{7}{0} + \binom{7}{1} = 1 + 7 = 8 = 2^3 = 2^{n-k}$$

Note that since the elements on each row of the standard array are the 2^k codewords each added to a unique n-tuple (the coset leader), the syndromes of all members of a coset is the same.

So finding the syndrome, we find out in what row of the standard array the received vector and, hopefully the transmitted codeword is. We can the output the coset leader. For small codes a lookup table is feasible. But for longer codes, we need to calculate the error based on the syndrome.



General decoder for a linear block code.

Truth table for the error digits of the correctable error patterns of the (7, 4) linear code

Syndromes			Correctable error patterns (coset leaders)						
s_0	s_1	s_2	e_0	e_1	e_2	e_3	e_4	e_5	e_6
0	0	0	0	0	0	0	0	0	0
1	0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	0	0	0
0	0	1	0	0	1	0	0	0	0
1	1	0	0	0	0	1	0	0	0
0	1	1	0	0	0	0	1	0	0
1	1	1	0	0	0	0	0	1	0
1	0	1	0	0	0	0	0	0	1

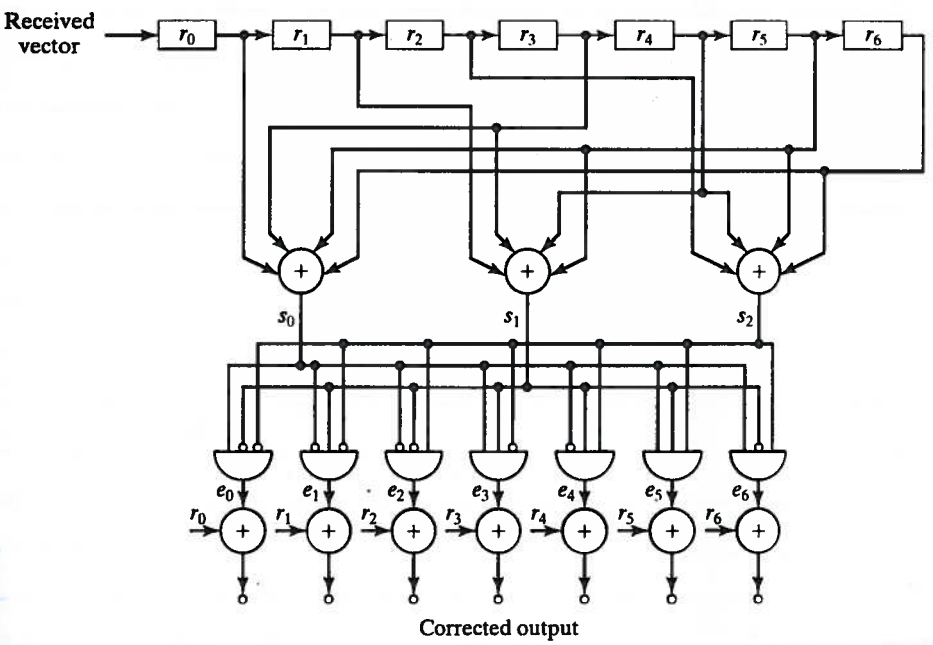


FIGURE 3.9: Decoding circuit for the (7, 4) code