DISTANCE PROPERTIES OF LINEAR BLOCK CODES

In this section, we'll discuss Distance properties of Linear Block codes. The Error correcting capability, and Error detecting capability of a linear block code is dependent on its minimum distance.

⊙ Let $v = (v_0, v_1, \ldots, v_{n-1})$ be a binary n-tuple: The Hamming weight of v, denoted by $d(v)$ is defined as number of non-zero components of v.

Eg: $v = (0\,1\,1\,0\,1\,1\,0)$, $d(v) = 4$.

⊙ Let v and w be two n-tuples. The Hamming distance between v and w, denoted by $d(v, w)$ is defined as the number of places where they differ.

Example 1: $v = 1\,0\,1\,0\,1\,0\,1$
$w = 1\,1\,1\,0\,0\,0\,0$ $\rightarrow$ $d(v, w) = 3$

⊙ Let v, w and x be three binary n-tuples. Then

$$d(v, w) + d(w, x) \geq d(v, x) \quad \text{(Triangle inequality)}$$

Proof: We can write

Hamming distance → $d(v,w) = w(v+w)$, $d(w,x) = w(w+x)$, $d(v,x) = w(v+x)$ ← Hamming weight

For any two code vectors a and b,

$$w(a) + w(b) \geq w(a+b)$$

Let $a = v + w$ and $b = w + x$, we get

$$w(v+w) + w(w+x) \geq w(v + \underbrace{w + w}_{0} + x) = w(v+x)$$

$$\Rightarrow \boxed{d(v,w) + d(w,x) \geq d(v,x)}$$

⊙ The minimum distance, d_{min} of a linear block code C is defined as

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

(Minimum Hamming distance between any two non-distinct codeword.)

Similarly, the minimum weight, w_{min} of C is defined as

$$w_{min} \triangleq \min\{ w(v) : v \in C, v \neq 0 \}$$

(Minimum weight of a non-zero codeword.)

Note:

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

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

(Sum of two valid codewords is also a codeword)

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

$$\boxed{d_{min} = w_{min}}$$

Theorem 1:

Let C be an (n, k) linear code with parity check matrix H. For each codeword of Hamming weight l, there exists l columns of H such that the vector sum of these l columns is equal to the zero vector.

Proof: Let's represent the parity check matrix, $H_{(n-k)\times n}$ whose Rank (H) = n-k, as

$$H = [h_0, h_1, \ldots, h_{n-1}],$$

where h_i represents the i^{th} column of H.

Let $v_{i_1}, v_{i_2}, \ldots, v_{i_l}$ be the l non-zero components of the codeword v, where $0 \leq i_1 \leq i_2 \leq \ldots \leq i_l \leq n-1$, then $v_{i_1} = v_{i_2} = \ldots = v_{i_l} = 1$.

Since v is a codeword, we must have

$$0 = v \cdot H^T$$
$$= v_0 h_0 + v_1 h_1 + \dots + v_{n-1} h_{n-1}$$
$$= v_{i_1} h_{i_1} + v_{i_2} h_{i_2} + \dots + v_{i_l} h_{i_l} + 0 \dots 0$$
$$= h_{i_1} + h_{i_2} + \dots + h_{i_l}$$

Theorem 2:

If there exists l columns of H whose vector sum is zero vector, there exists a codeword of Hamming weight l in C.

Proof:

Suppose $h_{i_1}, h_{i_2}, \dots, h_{i_l}$ are the l columns of H such that

$$h_{i_1} + h_{i_2} + \dots + h_{i_l} = 0.$$

Lets form a binary n-tuple $x = (x_0, x_1, x_2, \dots, x_{n-1})$ whose non-zero components are $x_{i_1}, x_{i_2}, \dots, x_{i_l}$. The Hamming-weight of x is l.

Consider the product

$$x \cdot H^T = x_0 h_0 + x_1 h_1 + \dots + x_{n-1} h_{n-1}$$
$$= x_{i_1} h_{i_1} + x_{i_2} h_{i_2} + \dots + x_{i_l} h_{i_l}$$

$\boxed{x \cdot H^T = 0}$ → which means, x has to be a valid codeword.

Thus, x is a codeword of weight l in C.

Let C be a linear block code with parity check matrix H. If no $d-1$ or fewer columns of H add to 0, the code has minimum weight atleast d.

Let C be a linear block code with parity check matrix H. The minimum weight of C, d_{min} is equal to the fewest number of columns of H (rows of H^T) that add to 0.

Example 2: Let $k=3$ and $n=6$. The table gives a $(6,3)$ linear block code.

Message (u_0, u_1, u_2)	Codewords $(v_0, v_1, v_2, v_3, v_4, v_5)$	
(0 0 0)	(0 0 0 0 0 0)	← All-zero codeword
(1 0 0)	(0 1 1 1 0 0)	→ 3
(0 1 0)	(1 0 1 0 1 0)	→ 3
(1 1 0)	(1 1 0 1 1 0)	→ 4
(0 0 1)	(1 1 0 0 0 1)	→ 3
(1 0 1)	(1 0 1 1 0 1)	→ 4
(0 1 1)	(0 1 1 0 1 1)	→ 4
(1 1 1)	(0 0 0 1 1 1)	→ 3

Minimum weights of these non-zero codewords.

Let A_i be the number of codewords in C with Hamming weight i.

$\Rightarrow A_0 \rightarrow$ No. of codewords with Hamming weight 0 = 1

$A_1 \rightarrow$ No. of codewords with Hamming weight 1 = 0

$A_2 \rightarrow$ No. of codewords with Hamming weight 2 = 0

$A_3 \rightarrow$ No. of codewords with Hamming weight 3 = 4

$A_4 \rightarrow$ No. of codewords with Hamming weight 4 = 3

$A_5 \rightarrow$ No. of codewords with Hamming weight 5 = 0

$A_6 \rightarrow$ No. of codewords with Hamming weight 6 = 0

The set $\{A_0, A_1, \ldots, A_n\}$ is called the weight distribution of C. Note that $A_0 = 1$, and $\sum_{i=0}^{n} A_i = 2^k$.

Example 3: For the $(6,3)$ code in example 2

$A_0 = 1, A_1 = 0, A_2 = 0, A_3 = 4, A_4 = 3, A_5 = 0, A_6 = 0$.

$\sum A_i = 8 = 2^3$

d_{min} in the above example is 3. (∵ Minimum weight of a non-zero codeword is 3).

Lets recall when does Undetected Error happen.
We send a valid codeword. But the receiver receives another valid codeword. So the receiver assumes it as the transmitted codeword, but not actually.

The probability of undetected error on a BSC is given by

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

Example 4.: For the (6, 3) code in Example 2,

$$P_u(E) = \underbrace{4p^3(1-p)^3}_{\text{4 codewords of weight 3 } (i=3)} + \underbrace{3p^4(1-p)^2}_{\text{3 codewords of weight 4 } (i=4)} \simeq 4p^3 \text{ (for small } p)$$

$\Rightarrow A_3 p^3 (1-p)^{6-3}$

$\Rightarrow A_4 p^4 (1-p)^{6-4}$

In general, the probability of undetected error is inversely proportional to number of parity bits. So, more parity bits we have, smaller will be the undetected error probability.

There exists (n, k) linear block codes for which

$$P_u(E) \leq 2^{-(n-k)} \quad , \forall \; p \leq 1/2.$$

on a BSC. From this relation, we can see that we can reduce it exponentially by increasing the no. of parity bits.

The above bound shows that, the undetected error probability can be made to decrease exponentially with the No. of parity check bits $n-k$ in a linear code.

For a codeword with minimum distance d_{min}, no error pattern with weight $d_{min}-1$ or less can change a transmitted codeword into another codeword.

Therefore, all error patterns with $d_{min}-1$ or fewer errors are detectable, and $d_{min}-1$ is called the Random error detecting capability of a block code.

Theorem: A block code C with minimum distance d_{min} is capable of correcting all error patterns of weight t or less, where t is an integer such that $2t+1 \leq d_{min} \leq 2t+2$.

Proof: Assuming codeword v is transmitted and r is the received sequence. Let $w \neq v$ be any other codeword. Then $d(v,w) \leq d(v,r) + d(r,w)$ (triangle inequality).

If the error pattern has weight t', then $d(v,r) = t'$.

Since v and w are codewords,

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

Therefore,

$$d(r,w) \geq d(v,w) - d(v,r) \geq 2t+1-t'$$

If $t' \leq t$, then

$$d(r,w) \geq t+1 > t \quad \text{and} \quad d(v,r) = t' \leq t$$

Hence r is closer to v than any other codeword w, and an ML decoder will decode correctly.

Theorem: For all $l \geq t+1$, there is atleast one error pattern of weight l that may not be correctly decoded by an ML decoder.

Proof: Let v and w be two codewords such that $d(v,w) = d_{min}$. Let e_1 and e_2 be two error patterns such that

(i) $e_1 + e_2 = v + w$

(ii) $w(e_1 + e_2) = w(e_1) + w(e_2)$ (non overlapping 1's)

(iii) $w(e_1) = l \geq t+1$

Then,

$$\begin{aligned} w(e_1) + w(e_2) &= w(e_1 + e_2) \\ &= w(v+w) \\ &= d(v,w) \\ &= d_{min}. \end{aligned}$$

Assuming v is transmitted and $r = v + e_1$ is received. Then

$$\begin{aligned} d(w, r) &= w(w + r) = w(w + v + e_1) \\ &= w(e_1 + e_2 + e_1) \\ &= w(e_2) \\ &= d_{min} - w(e_1) \\ &< 2t + 2 - (t+1) = t + 1 \end{aligned}$$

Therefore, $d(w, r) \leq d(v, r)$ and an ML decoder may decode incorrectly.

Hence, for a block code with minimum distance d_{min}, an ML decoder will correctly decode any error pattern of weight $t \triangleq \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$ or less.

t is called the random error correcting capability of the code.

Theorem: For an (n, k) linear code C with minimum distance d_{min}, all the n-tuples of weight $t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$ or less can be used as Coset Leaders of a standard array of C.

Proof:

Since minimum distance of C is d_{min}, minimum weight of C is also d_{min}.

Let x and y be two n-tuples of weight t or less.

$$w(x+y) \leq w(x) + w(y) \leq 2t < d_{min}$$

Suppose x and y are in the same coset, then $x+y$ must be a non-zero codeword in C.

This is impossible as weight of $x+y < d_{min}$.

Theorem: For an (n, k) linear code C with minimum distance d_{min}, if all the n-tuples of weight $t = \left\lfloor \frac{d_{min} - 1}{2} \right\rfloor$ or less are used as coset leaders of a standard array of C, then there is at least one n-tuple of weight $t+1$ that cannot be used as coset leader.

Proof:

Let v be the minimum weight codeword of C

Let x and y be two n-tuples that satisfies the following conditions:

$x + y = v$

x and y do not have non-zero component in common places.

From definition, x and y must be in the same coset, and

$$w(x) + w(y) = w(v) = d_{min}$$

If we choose $w(y) = t+1$, then

$w(x) = t$ or $t+1$ (since $2t+1 \leq d_{min} \leq 2t+2$)

Therefore, if x is chosen as coset leader, y cannot be coset leader.

SOME LINEAR BLOCK CODES

Examples of linear block codes

1. Repetition code
2. Single parity check code
3. Hamming code
4. Reed-Muller code

Repetition code

A repetition code of length n is a linear $(n,1)$ block code

(i) $k=1$, $n=n$.

It consists of two codewords.

- All zero codeword $0 = (0,0,\ldots,0)$ and
- All One codeword $1 = (1,1,\ldots,1)$

Codeword is obtained by repeating the information bit n times.

Generator Matrix is given by $G = [1\ 1\ \ldots 1]$

Single Parity check code

It is a linear $(k+1, k)$ block code with single parity bit.

(ii) $n = k+1$, $k = k$.

If $u = (u_0, u_1, \ldots, u_{k-1})$, then the parity check bit is given by $p = u_0 + u_1 + \ldots + u_{k-1}$

Each codeword is of the form

$$v = (p; \underbrace{u_0, u_1, \ldots, u_{k-1}}_{\text{Information bits}})$$

p: Parity bit

WKT, $v = uG = (u_0, u_1, \ldots, u_{k-1})\, G$

The Generator matrix for the Single parity check code in systematic form is given by

$$G = \left[\begin{array}{c|ccccc} 1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 1 & 0 & \cdots & 0 \\ 1 & 0 & 0 & 1 & \cdots & 0 \\ \vdots & & & & \vdots & \\ 1 & 0 & 0 & 0 & \cdots & 1 \end{array}\right]$$

Identity Matrix

The parity check matrix for the singe parity check code in systematic form is given by

$$H = [1\ 1 \ldots 1]$$

All codewords of the Single Parity check (SPC) codes are even weight.

Minimum distance of SPC code is 2.

SPC code can detect all error patterns with odd number of error.

Hamming Code.

Hamming codes are Single Error correcting codes.

For any $m \geq 3$, there exist a Hamming code with following parameters.

code length	:	$n = 2^m - 1$
Information bits	:	$k = 2^m - m - 1$
Parity bits	:	$n - k = m$
Error correcting capability	:	$t = 1$ ($\because$ Singe Error correcting code)
Minimum distance	:	$d_{min} = 3$

The parity check matrix

$$H = [I_m : P^T]$$

where the $2^m - m - 1$ columns of P^T consists of all m-tuples of weight 2 or more.

Example: For $m = 3$, the Hamming code is of length

$n = 2^3 - 1 = 7$,

$k = 2^3 - 3 - 1 = 4$,

that has parity check Matrix H,

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

I ← 3-tuples of weight 2 or more

and generator matrix G,

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

P I

$H = [I : P^T]$

$\downarrow$

$G = [P : I]$

We can rearrange the columns of the parity check matrix of Hamming code such that column in position i represents the integer i.

For example, for $m = 3$, the Hamming code is of length $n = 2^3 - 1 = 7$, $k = 2^3 - 3 - 1 = 4$, that has parity check matrix

$$H = \begin{array}{c} \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{matrix} \\ \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} \end{array}$$

Here the column $(x, y, z)^T$ represents the number $x(2^0) + y(2^1) + z(2^2)$.

Let r be the received vector. For decoding, we compute the Syndrome $r H^T$.

If at most one error has occured, the syndrome would be either the zero vector, or a column of H.

When one error has happened, the number represented by the column of the calculated syndrome is the position in codeword which is in error, and since we considered binary code, it can be corrected.

Example: Let 0101010 be a codeword in [7, 4] Hamming code. Suppose we received the vector 0001010. syndrome is given by

$$s = r H^T = [0\ 0\ 0\ 1\ 0\ 1\ 0] \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = (0\ 1\ 0)$$

The number represented by Syndrome is 2, hence the error is in Second bit position. Hence estimated codeword is 0 1 0 1 0 1 0.

↑ Just Flip it.

Reed - Muller code

For any integers m and r with $0 \le r \le m$, there exists a binary r^{th} order Reed Muller (RM) code, denoted by RM (r, m), with the following parameters.

Code length : $n = 2^m$

Dimension : $k(r, m) = 1 + {}^mC_1 + {}^mC_2 + \dots + {}^mC_r$

$= 1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r}$

Minimum distance : $d_{min} = 2^{m-r}$

where, ${}^mC_i$ (or) $\binom{m}{i}$ is the binomial coefficient.

Let $m = 4$, and $r = 2$, then $n = 16$,

$k(2,4) = 1 + {}^4C_1 + {}^4C_2 = 1 + 4 + 6 = 11$

$d_{min} = 2^{4-2} = 2^2 = 4.$

For $1 \le i \le m$, let v_i be a binary 2^m-tuple of the following form.

$$v_i = \left(\underbrace{0 \ldots 0}_{2^{i-1}}, \underbrace{1 \ldots 1}_{2^{i-1}}, \underbrace{0 \ldots 0}_{2^{i-1}}, \ldots, \underbrace{1 \ldots 1}_{2^{i-1}} \right)$$

which consists of 2^{m-i+1} alternating all-zero and all-one 2^{i-1} tuples.

For $m = 4$, we have the following four 16-tuples ($2^4 = 16$).

$v_1 = (0101\ 0101\ 0101\ 0101) \rightarrow i = 1,\ 2^{i-1} = 1$

$v_2 = (0011\ 0011\ 0011\ 0011) \rightarrow i = 2,\ 2^{i-1} = 2$

$v_3 = (0000\ 1111\ 0000\ 1111) \rightarrow i = 3,\ 2^{i-1} = 4$

$v_4 = (0000\ 0000\ 1111\ 1111) \rightarrow i = 4,\ 2^{i-1} = 8$

Let $x = (x_0, x_1, x_2, \ldots, x_{n-1})$ and $y = (y_0, y_1, y_2, \ldots, y_{n-1})$ be two binary n-tuples, we define Boolean product of x and y as follows.

$$x \cdot y = (x_0 \cdot y_0, x_1 \cdot y_1, \ldots, x_{n-1} \cdot y_{n-1}),$$

where "$\cdot$" denotes the Boolean product of x and y.

For example, if

$v_1 = (0101\ 0101\ 0101\ 0101)$

and $v_2 = (0011\ 0011\ 0011\ 0011)$,

then $v_1 \cdot v_2 = (0001\ 0001\ 0001\ 0001)$.

Let v_0 denote all-one 2^m-tuple, $v_0 = (1, 1, \ldots, 1)$.
say for example, for $m = 4$, $v_0 = (1111\ 1111\ 1111\ 1111)$ i.e. 16 1's.
For $1 \le i_1 < i_2 < \ldots < i_l \le m$, the product vector

$$v_{i_1} v_{i_2} \cdots v_{i_l}$$

is said to have degree l.
The weight of the product $v_{i_1} v_{i_2} \cdots v_{i_l}$ is equal to 2^{m-l}.

The r^{th} order RM code, $RM(r, m)$ of length 2^m is generated by following set of independent vectors:

- Generator matrix of Reed Muller code,

$$G_{RM}(r, m) = \{ \underbrace{v_0}_{1}, \underbrace{v_1, v_2, \ldots, v_m}_{{}^mC_1}, \underbrace{v_1 v_2, v_1 v_3, \ldots, v_{m-1} v_m}_{{}^mC_2}, \ldots$$

$\ldots$, upto products of degree r $\}$

There are $k(r, m) = 1 + {}^mC_1 + {}^mC_2 + \cdots + {}^mC_r$ vectors in $G_{RM}(r, m)$. If the vectors in $G_{RM}(r, m)$ are arranged as rows of a matrix, then the matrix is a Generator matrix of the RM (r, m) code.

Example: Let $m = 4$, and $r = 2$ (the second order RM code) of length $n = 16$ is generated by the following 11 vectors. ($n = 2^m = 2^4 = 16$)

v_0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
v_1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
v_2 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
v_3 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
v_4 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$v_1 v_2$ 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$v_1 v_3$ 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$v_1 v_4$ 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
$v_2 v_3$ 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1
$v_2 v_4$ 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1
$v_3 v_4$ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1

$\because r = 2$, we'll have to consider till products of degree 2.

Another way to construct Reed-Muller code.

⊙ For $1 \le r \le m$, we define

$$R(r,m) = \left\{ (u, u+v) \;\middle|\; \begin{array}{l} u \in R(r,m-1), \\ v \in R(r-1,m-1) \end{array} \right\}$$

(length 2^{m-1})

So, using two 2^{m-1} sequences, we can actually construct a Reed-Muller code of length 2^m.

⊙ The generator matrix can be written as

$$G(r,m) = \begin{bmatrix} G(r,m-1) & G(r,m-1) \\ 0 & G(r-1,m-1) \end{bmatrix}$$

⊙ Minimum distance of RM (r,m) is 2^{m-r}.

Proof: We'll prove the result by mathematical induction.

Let $m=1, r=0$, then RM $(0,1)$ is a length two repetition code. In this case, the minimum distance is 2.

RM $(1,1)$ has four codewords $\{00, 01, 11, 10\}$ of length 2. Minimum distance in this case is $2^{1-1} = 2^0 = 1$.

Let us assume, for upto m and for $0 \le r \le m$, the minimum distance is 2^{m-r}. We'll show that d_{min} for RM $(r, m+1)$ is 2^{m-r+1}.

⊙ Let $f, f' \in$ RM (r,m) and let $g, g' \in$ RM $(r-1, m)$. Then vectors $c_1 = (f, f+g)$ and $c_2 = (f', f'+g')$ must be in RM $(r, m+1)$.

⊙ If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \ge 2 \cdot 2^{m-r}$

⊙ If $g \ne g'$, then $d(c_1, c_2) = w(f-f') + w(\underbrace{g-g'}_{x} + \underbrace{f-f'}_{y})$

Since $w(x+y) \ge w(x) - w(y)$, we have

$$\begin{aligned} d(c_1, c_2) &\ge w(f-f') + w(g-g') - w(f-f') \\ &= w(g-g') \end{aligned}$$

Since $g - g' \in RM(r-1, m)$, so that

$$w(g-g') \geq 2^{m-(r-1)} = 2^{m-r+1}$$

⊙ From the construction, we can see that $RM(r-1, m)$ code is a proper subcode of the $RM(r, m)$ code. Hence

$$RM(0,m) \subset RM(1,m) \subset \dots \subset RM(r,m)$$

- The zeroth order RM code is a Repetition code
- The $(m-1)^{th}$ order RM code is a Singly Parity check code.
- The $(m-2)^{th}$ order RM code of length 2^m is distance-4 extended Hamming code obtained by adding an overall parity bit to the Hamming code of length $2^m - 1$.

Decoding of Reed-Muller code

Consider a 2nd order ($r=2$) Reed-Muller code of length $n=16$ ($m=4$) generated by following 11 vectors.

$$G = \begin{bmatrix}
v_0 & 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\
v_4 & 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1 \\
v_3 & 0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1 \\
v_2 & 0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1 \\
v_1 & 0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1 \\
v_3v_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1 \\
v_2v_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1 \\
v_1v_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1 \\
v_2v_3 & 0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1 \\
v_1v_3 & 0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,1\,0\,1 \\
v_1v_2 & 0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1
\end{bmatrix}$$

$2^4 = 16$

$k = 1 + {}^4C_1 + {}^4C_2$
$= 1 + 4 + 6$
$= 11$

11×16

The message to be encoded is given by

$$u = (a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$$

→ 11 bits

The codeword is given by

$$v = uG = (b_0, b_1, b_2, \ldots, b_{15}) = a_0 v_0 + a_4 v_4 + a_3 v_3 + a_2 v_2 + a_1 v_1 + a_{34} v_3 v_4 + a_{24} v_2 v_4 + a_{14} v_1 v_4 + a_{23} v_2 v_3 + a_{13} v_1 v_3 + a_{12} v_1 v_2$$

We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the vector $v_1 v_2$.

Eg.

v_0 : 1111 1111 1111 1111 (0 0 0 1)

$v_3 v_4$: 0000 0000 0000 1111 (0 0 0 1)

$v_1 v_2$: 0001 0001 0001 0001 (1 1 1 1)

Thus the code bit a_{12} can be written as

$$a_{12} = b_0 + b_1 + b_2 + b_3$$

$$a_{12} = b_4 + b_5 + b_6 + b_7$$

$$a_{12} = b_8 + b_9 + b_{10} + b_{11}$$

$$a_{12} = b_{12} + b_{13} + b_{14} + b_{15}$$

RM codes uses majority logic decision rule for decoding.

Consider a 2nd order Reed-Muller code of length $n = 16$ generated by following 11 vectors.

v_0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

v_4 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

v_3 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1

v_2 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1

v_1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

$v_3 v_4$ 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

$v_2 v_4$ 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1

$v_1 v_4$ 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1 0

$v_2 v_3$ 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1

$v_1 v_3$ 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1

$v_1 v_2$ 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1

Let $r = (r_0, r_1, \cdots, r_{15})$ be the received vector. In decoding a_{12}, we form the following equations.

$$A_1 = r_0 + r_1 + r_2 + r_3$$

$$A_2 = r_4 + r_5 + r_6 + r_7$$

$$A_3 = r_8 + r_9 + r_{10} + r_{11}$$

$$A_4 = r_{12} + r_{13} + r_{14} + r_{15}$$

Similarly we can decode $a_{13}, a_{23}, a_{14}, a_{24}, a_{34}$. For example, for a_{13}, we have

$$A_1 = r_0 + r_1 + r_4 + r_5$$

$$A_2 = r_2 + r_3 + r_6 + r_7$$

$$A_3 = r_8 + r_9 + r_{12} + r_{13}$$

$$A_4 = r_{10} + r_{11} + r_{14} + r_{15}$$

Eg. $v_0 =$ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 $\longrightarrow 0$

$v_1 v_3 =$ 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1 $\longrightarrow 1$

$v_1 v_2 =$ 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1 $\longrightarrow 0$

For a_{23} we have

$$A_1 = r_0 + r_2 + r_4 + r_6$$
$$A_2 = r_1 + r_3 + r_5 + r_7$$
$$A_3 = r_8 + r_{10} + r_{12} + r_{14}$$
$$A_4 = r_9 + r_{11} + r_{13} + r_{15}$$

For a_{14} we have

$$A_1 = r_0 + r_1 + r_8 + r_9$$
$$A_2 = r_2 + r_3 + r_{10} + r_{11}$$
$$A_3 = r_4 + r_5 + r_{12} + r_{13}$$
$$A_4 = r_6 + r_7 + r_{14} + r_{15}$$

For a_{24} we have

$$A_1 = r_0 + r_2 + r_8 + r_{10}$$
$$A_2 = r_1 + r_3 + r_9 + r_{11}$$
$$A_3 = r_4 + r_6 + r_{12} + r_{14}$$
$$A_4 = r_5 + r_7 + r_{13} + r_{15}$$

For a_{34} we have

$$A_1 = r_0 + r_4 + r_8 + r_{12}$$
$$A_2 = r_1 + r_5 + r_9 + r_{13}$$
$$A_3 = r_2 + r_6 + r_{10} + r_{14}$$
$$A_4 = r_3 + r_7 + r_{11} + r_{15}$$

After decoding $a_{12}, a_{13}, a_{23}, a_{14}, a_{24}, a_{34}$, we form a modified received vector as

$$r^{(1)} = \left(r_0^{(1)}, r_1^{(1)}, \ldots, r_{15}^{(1)}\right)$$

$$= r - a_{34} v_3 v_4 - a_{24} v_2 v_4 - a_{14} v_1 v_4$$
$$- a_{23} v_2 v_3 - a_{13} v_1 v_3 - a_{12} v_1 v_2$$

Consider a 2nd order Reed-Muller code of length $n = 16$ generated by following 11 vectors.

v_0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

v_4 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

v_3 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1

v_2 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1

v_1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

$v_3 v_4$ 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

$v_2 v_4$ 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1

$v_1 v_4$ 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1

$v_2 v_3$ 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1

$v_1 v_3$ 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1

$v_1 v_2$ 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1

In absence of errors, we can write $r^{(1)}$ as following codeword.

$$\left(b_0^{(1)}, b_1^{(1)}, \ldots, b_{15}^{(1)}\right) = a_0 v_0 + a_4 v_4 + a_3 v_3 + a_2 v_2 + a_1 v_1$$

We can see that sum of every two components of v_0, v_4, v_3, v_2 starting from first is zero, whereas for v_1 it is 1.

Therefore we can form eight independent equations for a_1, given by

$$a_1 = b_0^{(1)} + b_1^{(1)}$$
$$a_1 = b_2^{(1)} + b_3^{(1)}$$
$$a_1 = b_4^{(1)} + b_5^{(1)}$$
$$a_1 = b_6^{(1)} + b_7^{(1)}$$
$$a_1 = b_8^{(1)} + b_9^{(1)}$$
$$a_1 = b_{10}^{(1)} + b_{11}^{(1)}$$
$$a_1 = b_{12}^{(1)} + b_{13}^{(1)}$$
$$a_1 = b_{14}^{(1)} + b_{15}^{(1)}$$

Similarly, independent determination of a_2, a_3 and a_4 can be formed.

We can form eight independent equations for a_2, given by

$$a_2 = b_0^{(1)} + b_2^{(1)} \qquad a_2 = b_8^{(1)} + b_{10}^{(1)}$$
$$a_2 = b_1^{(1)} + b_3^{(1)} \qquad a_2 = b_9^{(1)} + b_{11}^{(1)}$$
$$a_2 = b_4^{(1)} + b_6^{(1)} \qquad a_2 = b_{12}^{(1)} + b_{14}^{(1)}$$
$$a_2 = b_5^{(1)} + b_7^{(1)} \qquad a_2 = b_{13}^{(1)} + b_{15}^{(1)}$$

We can form eight independent equations for a_3, given by

$$a_3 = b_0^{(1)} + b_4^{(1)} \qquad a_3 = b_8^{(1)} + b_{12}^{(1)}$$
$$a_3 = b_1^{(1)} + b_5^{(1)} \qquad a_3 = b_9^{(1)} + b_{13}^{(1)}$$
$$a_3 = b_2^{(1)} + b_6^{(1)} \qquad a_3 = b_{10}^{(1)} + b_{14}^{(1)}$$
$$a_3 = b_3^{(1)} + b_7^{(1)} \qquad a_3 = b_{11}^{(1)} + b_{15}^{(1)}$$

We can form eight independent equations for a_4, given by

$$a_4 = b_0^{(1)} + b_8^{(1)} \qquad a_4 = b_4^{(1)} + b_{12}^{(1)}$$
$$a_4 = b_1^{(1)} + b_9^{(1)} \qquad a_4 = b_5^{(1)} + b_{13}^{(1)}$$
$$a_4 = b_2^{(1)} + b_{10}^{(1)} \qquad a_4 = b_6^{(1)} + b_{14}^{(1)}$$
$$a_4 = b_3^{(1)} + b_{11}^{(1)} \qquad a_4 = b_7^{(1)} + b_{15}^{(1)}$$

Equations for decoding a_1 can be written as

$$A_1^{(1)} = r_0^{(1)} + r_1^{(1)} \qquad A_5^{(1)} = r_8^{(1)} + r_9^{(1)}$$
$$A_2^{(1)} = r_2^{(1)} + r_3^{(1)} \qquad A_6^{(1)} = r_{10}^{(1)} + r_{11}^{(1)}$$
$$A_3^{(1)} = r_4^{(1)} + r_5^{(1)} \qquad A_7^{(1)} = r_{12}^{(1)} + r_{13}^{(1)}$$
$$A_4^{(1)} = r_6^{(1)} + r_7^{(1)} \qquad A_8^{(1)} = r_{14}^{(1)} + r_{15}^{(1)}$$

After decoding a_1, a_2, a_3, a_4, we create a modified received vector $r^{(2)}$.

$$r^{(2)} = \left(r_0^{(2)}, r_1^{(2)}, \ldots, r_{15}^{(2)} \right)$$

$$= r^{(1)} - a_4 v_4 - a_3 v_3 - a_2 v_2 - a_1 v_1$$

In absence of errors, we have

$$r^{(2)} = a_0 v_0 = (a_0, a_0, \ldots, a_0)$$

a_0 is decoded to be the value of majority of the bits in $r^{(2)}$.
