

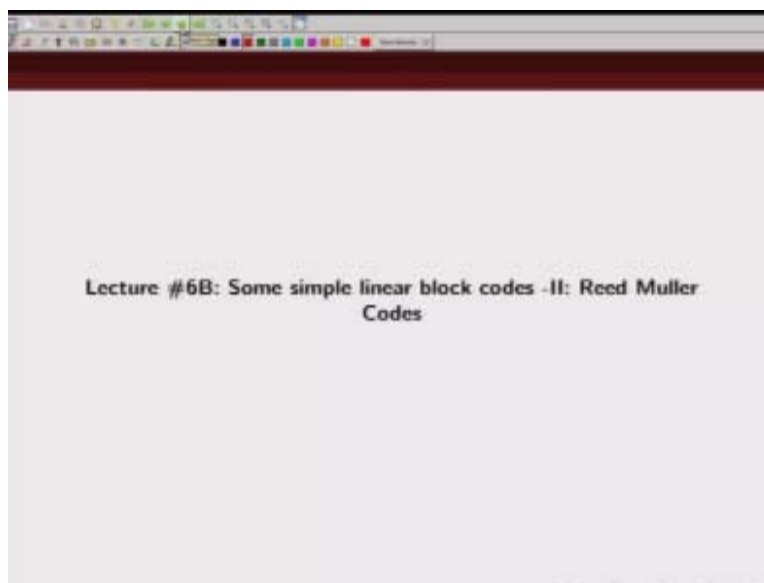
Indian Institute of Technology Kanpur
National Programme on Technology Enhanced Learning (NPTEL)
Course Title
Error Control Coding: An Introduction to Linear Block Codes

Lecture-6B
Some Simple Linear Block Codes-II: Reed Muller Codes

by
Dr. Adrish Banerjee
Department of Electrical Engineering, IIT Kanpur

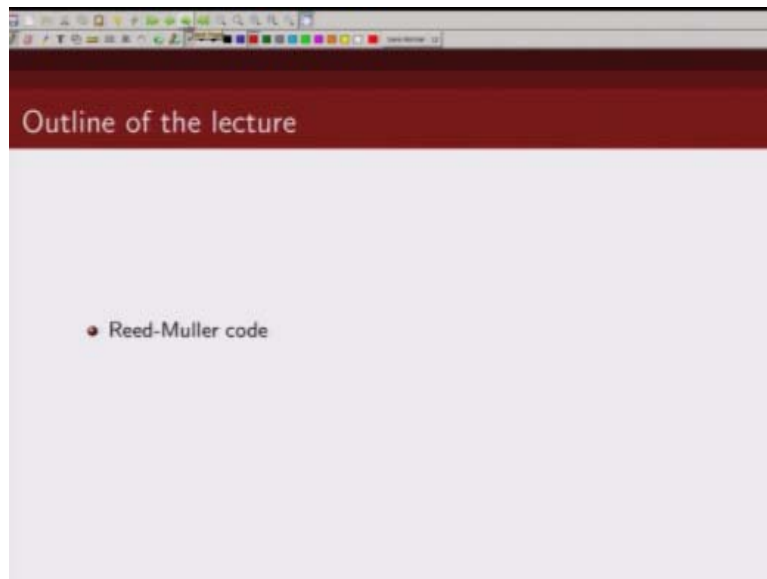
Welcome to the course on error control coding, an introduction to linear block codes.

(Refer Slide Time: 00:24)



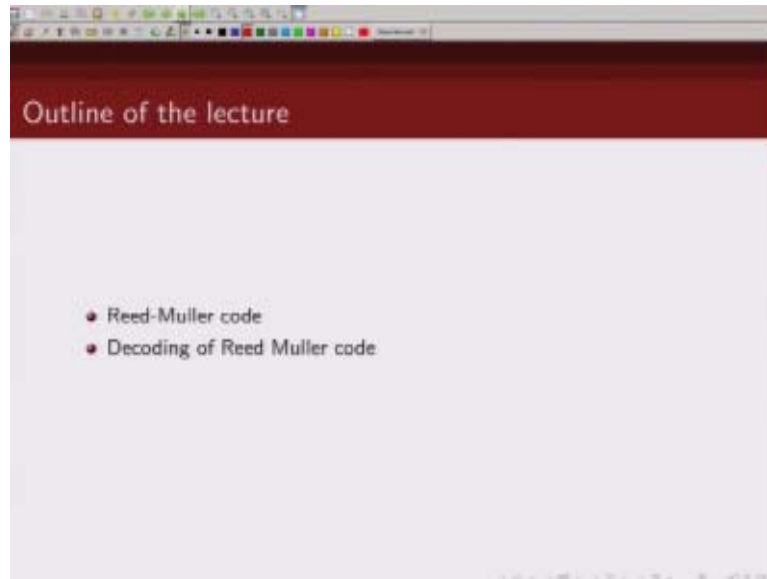
So we will continue our discussions on some simple linear block codes.

(Refer Slide Time: 00:29)



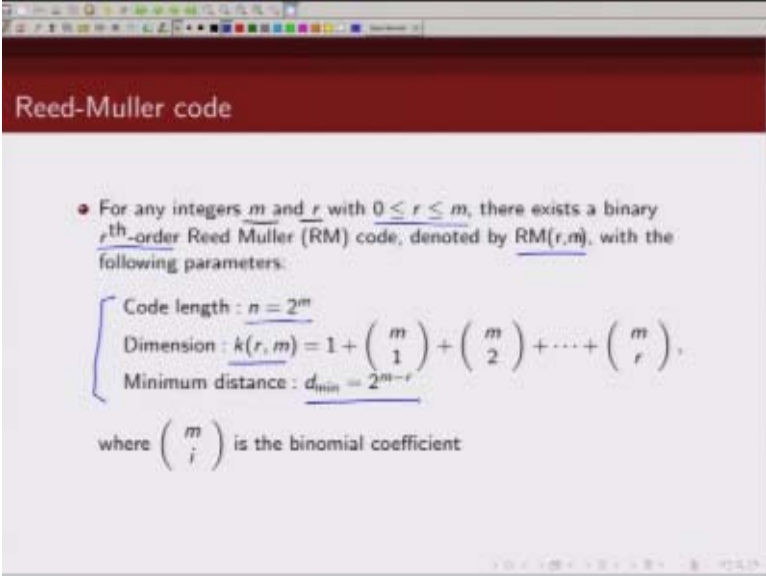
This time we are going to discuss about Reed-Muller codes, we will talk about their construction, we will give an example, we will prove some properties of Reed-Muller code.

(Refer Slide Time: 00:41)



And then we will talk about decoding of Reed-Muller code.

(Refer Slide Time: 00:43)



Reed-Muller code

- For any integers m and r with $0 \leq r \leq m$, 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 : $k(r, m) = 1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r}$,
- Minimum distance : $d_{\min} = 2^{m-r}$

where $\binom{m}{i}$ is the binomial coefficient

So for any integer m and r such that r lies between $-r$ is greater than zero and less than equal to m there exist a binary r^{th} order Reed-Muller code which we denote by these parameter r and m , Reed-Muller code has the following code properties. So the length of the code is 2^m and the dimension key is given by $1 + m \text{ choose } 1 \text{ plus } m \text{ choose } 2 \text{ up to } m \text{ choose } r$. And the minimum distance of the code is given by 2^{m-r} .

(Refer Slide Time: 01:34)

Reed-Muller code

- For any integers m and r with $0 \leq r \leq m$, there exists a binary r^{th} -order Reed Muller (RM) code, denoted by $RM(r,n)$, with the following parameters:
 - Code length : $n = 2^m$
 - Dimension : $k(r, m) = 1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r}$,
 - Minimum distance : $d_{\min} = 2^{m-r}$

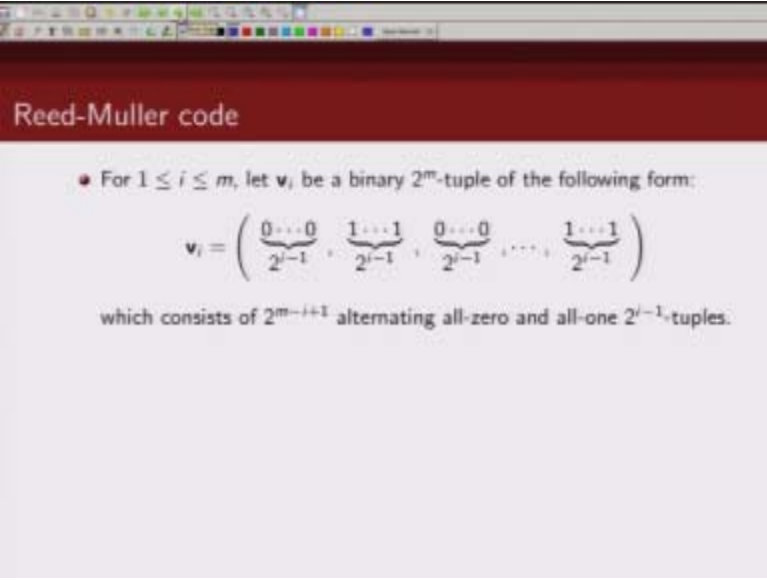
where $\binom{m}{i}$ is the binomial coefficient

- Let $m = 4$, and $r = 2$, then $n = 16$, $k = 11$, and $d_{\min} = 4$

Handwritten calculation for k : $1 + {}^4C_1 + {}^4C_2 = 1 + 4 + \frac{4 \times 3}{2} = 11$

So let us take an example, let us take m to be 4 and r to be 2. So in this case the length of the code word will be 2^4 which is 16, and since the order of this Reed-Muller code is 2, so this k will be $1 + 4C_1 + 4C_2$ so this will be $1 + 4 + 4 \times 3 / 2$ so this will be equal to 11, $1 + 4 + 6$ so k is this thing, and minimum distance is 2^{4-2} which is 4.

(Refer Slide Time: 02:27)



Reed-Muller code

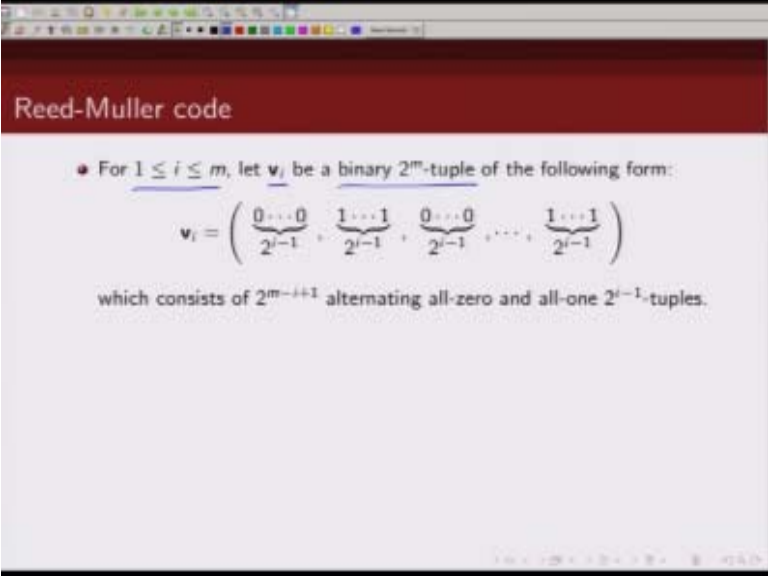
- For $1 \leq i \leq m$, let \mathbf{v}_i be a binary 2^m -tuple of the following form:

$$\mathbf{v}_i = \left(\underbrace{0 \dots 0}_{2^{i-1}}, \underbrace{1 \dots 1}_{2^{i-1}}, \underbrace{0 \dots 0}_{2^{i-1}}, \dots, \underbrace{1 \dots 1}_{2^{i-1}} \right)$$

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

Now how do we construct a Reed-Muller code? So to do that let us define.

(Refer Slide Time: 02:35)



Reed-Muller code

- For $1 \leq i \leq m$, let \mathbf{v}_i be a binary 2^m -tuple of the following form:

$$\mathbf{v}_i = \left(\underbrace{0 \dots 0}_{2^{i-1}}, \underbrace{1 \dots 1}_{2^{i-1}}, \underbrace{0 \dots 0}_{2^{i-1}}, \dots, \underbrace{1 \dots 1}_{2^{i-1}} \right)$$

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

So we are defining an, binary m-tuple let us call it \mathbf{v}_i so for i going from 1 to m we define a binary m-tuple in this particular fashion. So there is alternating runs of zeros and ones.

(Refer Slide Time: 02:59)

Reed-Muller code

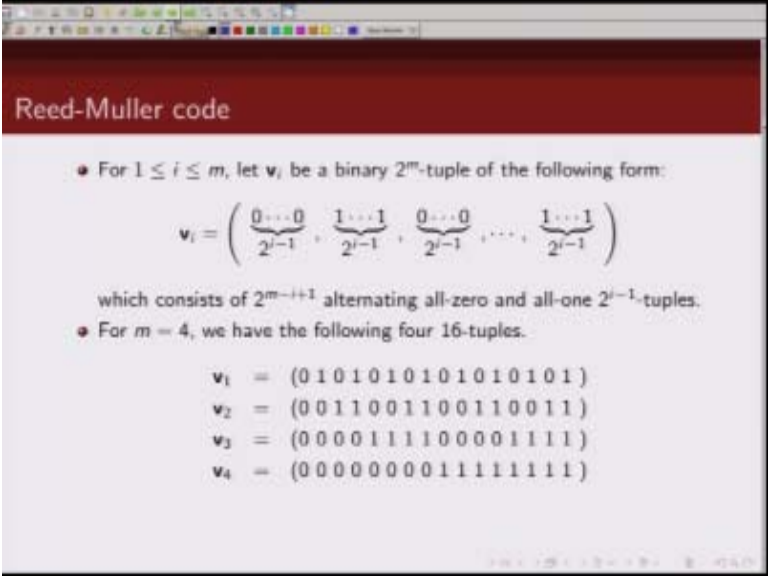
- For $1 \leq i \leq m$, let \underline{v}_i be a binary 2^m -tuple of the following form:

$$\underline{v}_i = \left(\underbrace{0 \dots 0}_{2^{i-1}}, \underbrace{1 \dots 1}_{2^{i-1}}, \underbrace{0 \dots 0}_{2^{i-1}}, \dots, \underbrace{1 \dots 1}_{2^{i-1}} \right)$$

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

So v_i is run of zeros for 2^{i-1} times, then run of ones for 2^{i-1} like that. So this v_i consist of $2m-i+1$ alternating zeros and ones and where each of these runs of zeros and ones are for 2^{i-1} .

(Refer Slide Time: 03:29)



Reed-Muller code

- For $1 \leq i \leq m$, let \mathbf{v}_i be a binary 2^m -tuple of the following form:

$$\mathbf{v}_i = \left(\underbrace{0 \dots 0}_{2^{i-1}}, \underbrace{1 \dots 1}_{2^{i-1}}, \underbrace{0 \dots 0}_{2^{i-1}}, \dots, \underbrace{1 \dots 1}_{2^{i-1}} \right)$$

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.

$$\begin{aligned} \mathbf{v}_1 &= (01010101010101) \\ \mathbf{v}_2 &= (0011001100110011) \\ \mathbf{v}_3 &= (0000111100001111) \\ \mathbf{v}_4 &= (0000000011111111) \end{aligned}$$

So let us take an example, let us consider m to be 4, m to be 4.

(Refer Slide Time: 03:37)

Reed-Muller code

- For $1 \leq i \leq m$, let \mathbf{v}_i be a binary 2^m -tuple of the following form:

$$\mathbf{v}_i = \left(\underbrace{0 \dots 0}_{2^{i-1}}, \underbrace{1 \dots 1}_{2^{i-1}}, \underbrace{0 \dots 0}_{2^{i-1}}, \dots, \underbrace{1 \dots 1}_{2^{i-1}} \right)$$

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.

$\mathbf{v}_1 = (0101010101010101)$	$2^{i-1} = 1$
$\mathbf{v}_2 = (0011001100110011)$	$2^{i-1} = 2$
$\mathbf{v}_3 = (0000111100001111)$	$2^{i-1} = 4$
$\mathbf{v}_4 = (0000000011111111)$	$2^{i-1} = 8$

So then this m-tuples are 2^4 that is 16 okay. So what is \mathbf{v}_1 , now \mathbf{v}_1 should have runs of zeros and ones where this run is 2^{i-1} so when i is 1 this is 1. So that means we should have \mathbf{v}_1 is 0, because that is the run of 1 then followed by a run of 1 for one time, then followed by zero one time, then 1 one time so like that it will continue for this block of 16. Now what is \mathbf{v}_2 , for \mathbf{v}_2 i is 2, so 2^{i-1} would be in this case 2.

So we should have 2 runs of zero followed by run of 1 which is repeated twice, run of zero is repeated twice, 101 this you continue up to block size of 16. What about \mathbf{v}_3 , in this case i is 3. So what will be 2^{i-1} , 2^{i-1} would be 4, so you have runs of zero for four times, followed by runs of 1, four times, then again runs of zero, and runs of 1. What about \mathbf{v}_4 , here i is 4, so 2^{i-1} will be 8, so we have runs of zeros for eight times followed by runs of 1 eight times. So that is how we define our – this binary m-tuple for each of these i going from 1 to m .

(Refer Slide Time: 05:46)

Reed-Muller code

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

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

where " \cdot " denotes the Boolean product of \mathbf{x} and \mathbf{y} .

Next we define a Boolean product. How do we define a Boolean product let us say, we have 2 n -tuples \mathbf{x} and \mathbf{y} . So I am denoting \mathbf{x} by $x_0, x_1, x_2, x_3, x_{n-1}$, similarly denoting \mathbf{y} by y_0, y_1, y_2, y_{n-1} . Now we define these Boolean product as – so this is bitwise and $x_0.y_0, x_1.y_1, x_2.y_2$ up to $x_{n-1}.y_{n-1}$, so this $x_0.y_0$ will be 1, only if both x_0 and y_0 are 1, otherwise it will be 0.

(Refer Slide Time: 06:41)

Reed-Muller code

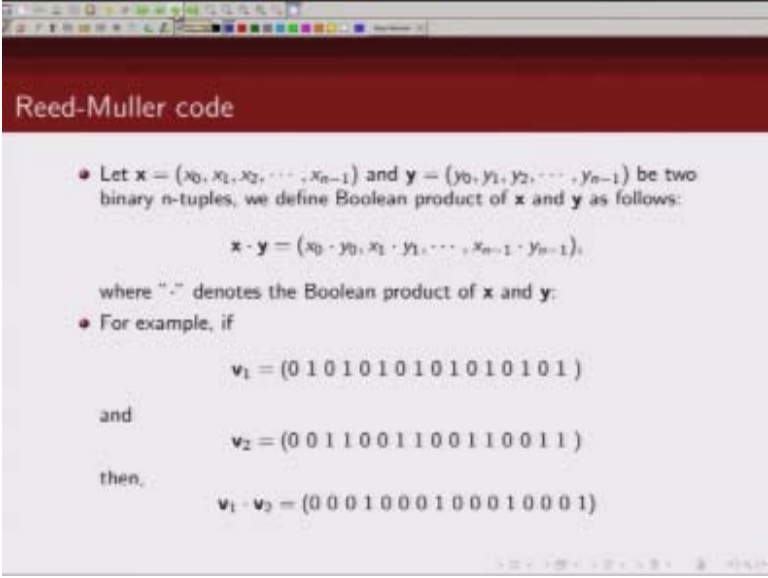
- Let $\mathbf{x} = (x_0, x_1, x_2, \dots, x_{n-1})$ and $\mathbf{y} = (y_0, y_1, y_2, \dots, y_{n-1})$ be two binary n -tuples, we define Boolean product of \mathbf{x} and \mathbf{y} as follows:

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

where " \cdot " denotes the Boolean product of \mathbf{x} and \mathbf{y} .

And same with others so $x_i \cdot y_i$ will be 1 only if both of them are 1.

(Refer Slide Time: 06:49)



Reed-Muller code

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

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

where " \cdot " denotes the Boolean product of \mathbf{x} and \mathbf{y} :

- For example, if

$$\mathbf{v}_1 = (0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1\ 0\ 1)$$

and

$$\mathbf{v}_2 = (0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1\ 0\ 0\ 1\ 1)$$

then,

$$\mathbf{v}_1 \cdot \mathbf{v}_2 = (0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1)$$

So that is how we are defining this Boolean product operation.

(Refer Slide Time: 06:57)

Reed-Muller code

- For $1 \leq i \leq m$, let \mathbf{v}_i be a binary 2^m -tuple of the following form:

$$\mathbf{v}_i = \left(\underbrace{0 \dots 0}_{2^{i-1}}, \underbrace{1 \dots 1}_{2^{i-1}}, \underbrace{0 \dots 0}_{2^{i-1}}, \dots, \underbrace{1 \dots 1}_{2^{i-1}} \right)$$
 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.

$\mathbf{v}_1 =$	(0101010101010101)	$2^{i-1} = 1$
$\mathbf{v}_2 =$	(0011001100110011)	$2^{i-1} = 2$
$\mathbf{v}_3 =$	(0000111100001111)	$2^{i-1} = 4$
$\mathbf{v}_4 =$	(0000000011111111)	$2^{i-1} = 8$

So let us take an example, this is our \mathbf{v}_1 if you recall this was our \mathbf{v}_1 and this is our \mathbf{v}_2 .

(Refer Slide Time: 07:03)

Reed-Muller code

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

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

where " \cdot " denotes the Boolean product of \mathbf{x} and \mathbf{y} :

(Refer Slide Time: 07:05)

Reed-Muller code

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

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

where " \cdot " denotes the Boolean product of \mathbf{x} and \mathbf{y} :

- For example, if

$$\mathbf{v}_1 = (0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1)$$

and

$$\mathbf{v}_2 = (0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1 \ 0 \ 0 \ 1 \ 1)$$

then,

$$\mathbf{v}_1 \cdot \mathbf{v}_2 = (0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 1)$$

So if we define Boolean product between \mathbf{v}_1 and \mathbf{v}_2 we write it at $\mathbf{v}_1 \cdot \mathbf{v}_2$ and $\mathbf{v}_1 \cdot \mathbf{v}_2$ will be 1 only where \mathbf{v}_1 and \mathbf{v}_2 both are 1. So which is like this location number 4 bit, this location, then this location and then this location. So you can see it is only one at the 4th, 8th, 12th and 16th location, all other time is zero. This is zero for all other time okay. So this is how we define the Boolean product.

(Refer Slide Time: 07:53)

Reed-Muller code

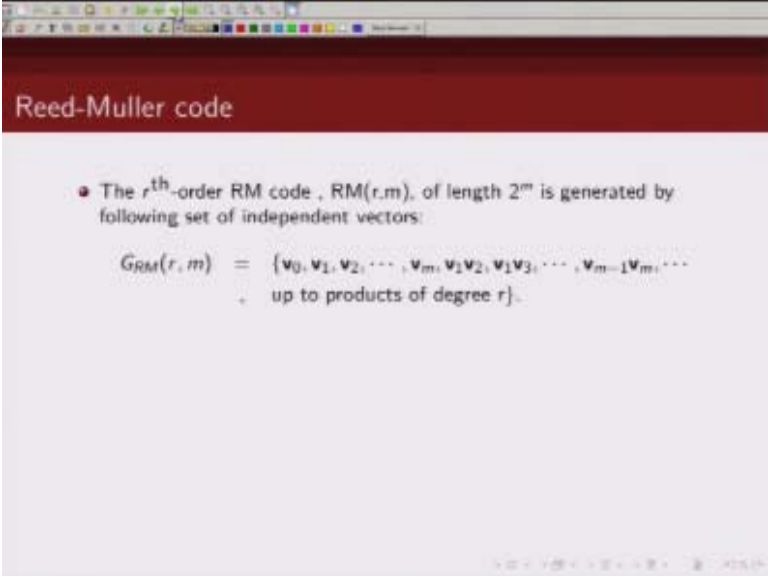
• Let v_0 denote all one 2^m -tuple, $v_0 = (1, 1, \dots, 1)$. For $l \leq i_1 < i_2 < \dots < i_l \leq m$, the product vector

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

is said to have degree l .

We also define an all one tuple so this v_0 is basically all ones of length 2^m . Now for i_1, i_2, i_3, i_l which lies between 1 and m we can define this product vector $v_{i_1}, v_{i_2}, v_{i_3}, v_{i_l}$ where this is basically Boolean product between these v_i 's. And we say this has degree l if there are l v_i 's which are participating in this product.

(Refer Slide Time: 08:37)



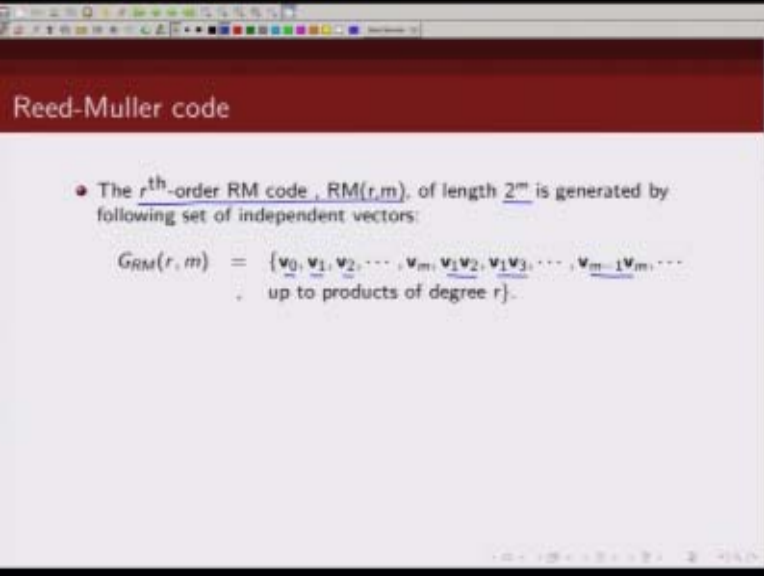
Reed-Muller code

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

$$G_{RM}(r,m) = \{v_0, v_1, v_2, \dots, v_m, v_1v_2, v_1v_3, \dots, v_{m-1}v_m, \dots, \text{up to products of degree } r\}$$

And weight of this product is given by 2^{m-1} . So now that we have defined these tuples v_i 's and the Boolean product between them, we are ready to define the generator matrix for Reed-Muller code.

(Refer Slide Time: 09:03)



Reed-Muller code

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

$$G_{RM}(r, m) = \{ \underline{v_0}, \underline{v_1}, \underline{v_2}, \dots, \underline{v_m}, \underline{v_1 v_2}, \underline{v_1 v_3}, \dots, \underline{v_{m-1} v_m}, \dots \}$$

up to products of degree r .

So an r^{th} order Reed-Muller code which is of length 2^m can be generated by these set of independent vectors where these vectors are v_0, v_1, v_2 then Boolean product of second order which is v_1, v_2, v_1, v_3 these are all second order product, then we will have third order product, fourth order product depending on what the r is. So we generate Reed-Muller code using these 2^m tuples basically of these v_0, v_1, v_2 and their Boolean product.

(Refer Slide Time: 09:46)

Reed-Muller code

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

$$G_{RM}(r, m) = \{v_0, v_1, v_2, \dots, v_m, v_1v_2, v_1v_3, \dots, v_{m-1}v_m, \dots, \text{up to products of degree } r\}.$$

- There are

$$k(r, m) = 1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r},$$

vectors in $G_{RM}(r, m)$

And as you can see that v_0 is all one sequence, so there is one such possible ways, we can get this v_1 this mC_1 ways of choosing v_1 mC_2 ways of – so v_1, v_2, v_3, v_m this is basically m choose 1, then Boolean product of degree 2 can be chosen m chose 2A.

(Refer Slide Time: 10:18)

Reed-Muller code

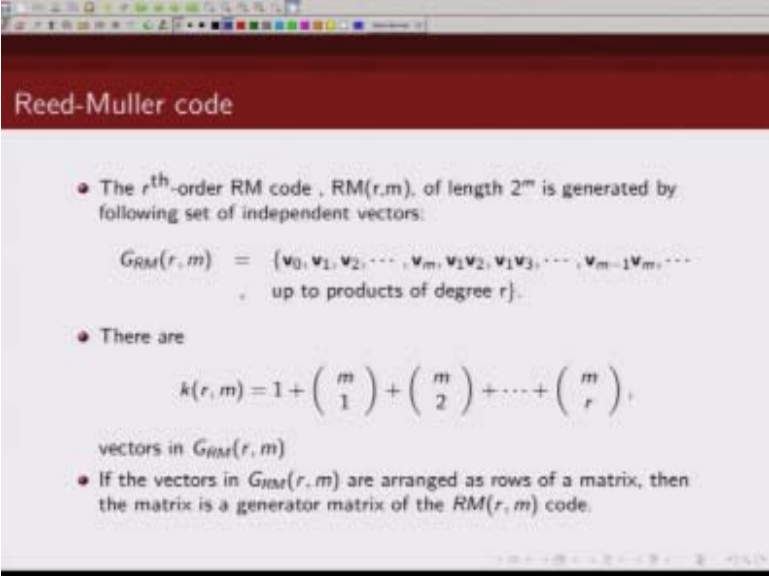
- The r^{th} -order RM code, $\text{RM}(r,m)$, of length 2^m is generated by following set of independent vectors:
$$G_{\text{RM}}(r,m) = \{ \mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m, \mathbf{v}_1\mathbf{v}_2, \mathbf{v}_1\mathbf{v}_3, \dots, \mathbf{v}_{m-1}\mathbf{v}_m, \dots \}$$

up to products of degree r .
- There are
$$k(r,m) = 1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r},$$

vectors in $G_{\text{RM}}(r,m)$

And similarly Boolean product up to order r can be chosen m choose r ways. So that is basically the dimension of the code.

(Refer Slide Time: 10:31)



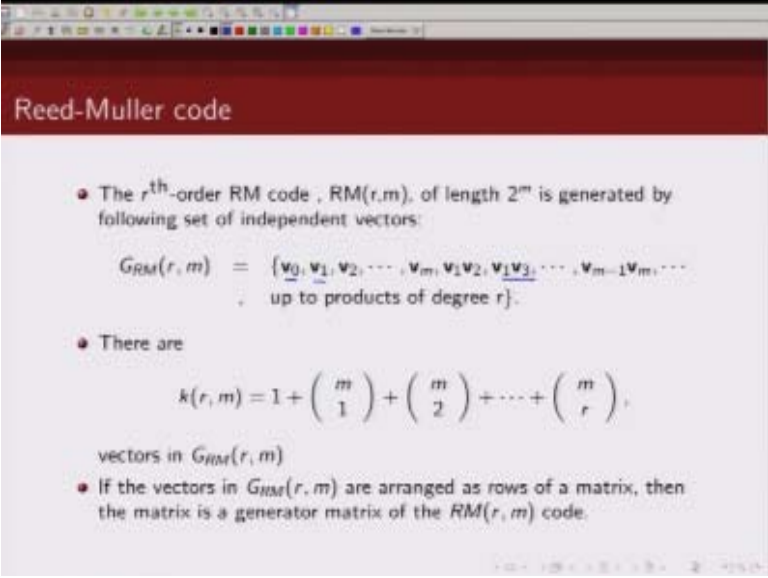
Reed-Muller code

- The r^{th} -order RM code, $RM(r, m)$, of length 2^m is generated by following set of independent vectors:
$$G_{RM}(r, m) = \{v_0, v_1, v_2, \dots, v_m, v_1v_2, v_1v_3, \dots, v_{m-1}v_m, \dots, \text{up to products of degree } r\}.$$
- There are
$$k(r, m) = 1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r},$$

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.

Now if we arrange these vectors v_0, v_1, v_2 and the Boolean product up to order r as rows of a matrix, that will be our generator matrix for Reed-Muller code. And each of these v_0, v_1 and their Boolean product they are basically linearly independent.

(Refer Slide Time: 10:59)



Reed-Muller code

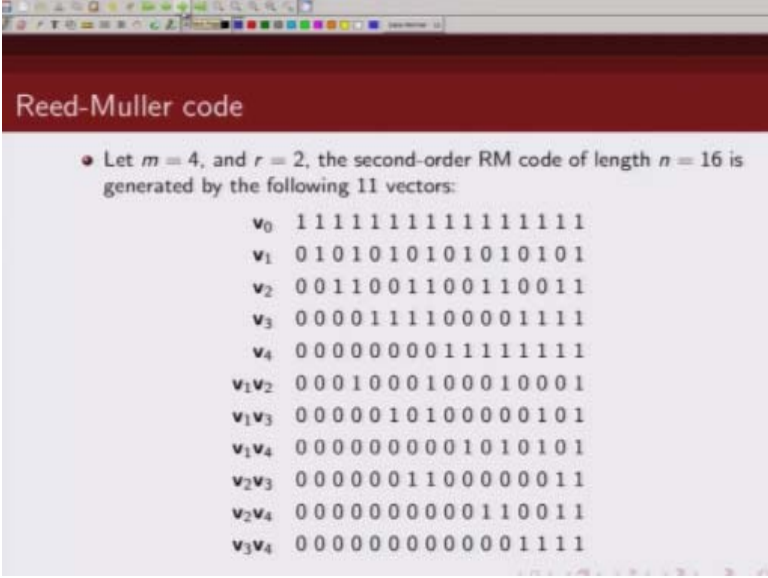
- The r^{th} -order RM code, $RM(r, m)$, of length 2^m is generated by following set of independent vectors:
$$G_{RM}(r, m) = \{ \underline{v}_0, \underline{v}_1, \underline{v}_2, \dots, \underline{v}_m, \underline{v}_1 \underline{v}_2, \underline{v}_1 \underline{v}_3, \dots, \underline{v}_{m-1} \underline{v}_m, \dots \}$$

. up to products of degree r .
- There are
$$k(r, m) = 1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r},$$

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.

So we can generate our Reed-Muller code using these v_0, v_i and their Boolean product as rows of our generator matrix.

(Refer Slide Time: 11:07)



Reed-Muller code

• Let $m = 4$, and $r = 2$, the second-order RM code of length $n = 16$ is generated by the following 11 vectors:

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

So let us illustrate this with an example we take a case where

(Refer Slide Time: 11:12)

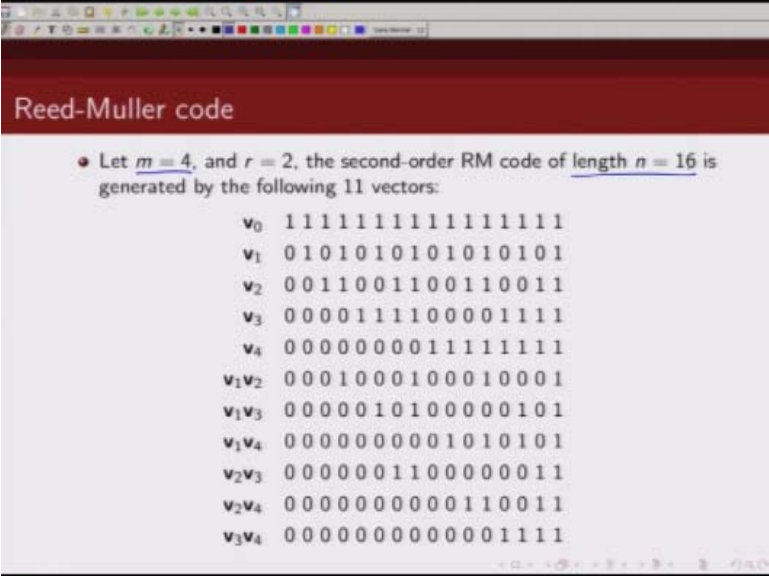
Reed-Muller code

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

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

m is 4 so n is 16 meaning our code word length would be 2^m which is 16 so we are dealing with Reed Muller code of length 16. Now let us consider

(Refer Slide Time: 11:25)



Reed-Muller code

• Let $m = 4$, and $r = 2$, the second-order RM code of length $n = 16$ is generated by the following 11 vectors:

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

A second order Reed Muller code so we will have to now recall what is a degree if you go back

(Refer Slide Time: 11:36)

Reed-Muller code

- Let \mathbf{v}_0 denote all one 2^m -tuple, $\mathbf{v}_0 = (1, 1, \dots, 1)$. For $l \leq i_1 < i_2 < \dots < i_l \leq m$, the product vector $\mathbf{v}_{i_1} \mathbf{v}_{i_2} \dots \mathbf{v}_{i_l}$ is said to have degree l .
- The weight of the product $\mathbf{v}_{i_1} \mathbf{v}_{i_2} \dots \mathbf{v}_{i_l}$ is equal to 2^{m-l} .

This product vector is set to have degree l if there are l such \mathbf{v}_i 's which are participating in this Boolean product

(Refer Slide Time: 11:45)

Reed-Muller code

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

$$G_{RM}(r,m) = \{ \underline{v_0}, \underline{v_1}, \underline{v_2}, \dots, v_m, v_1v_2, v_1v_3, \dots, v_{m-1}v_m, \dots \}$$
, up to products of degree r .
- There are

$$\underline{k(r,m)} = 1 + \binom{m}{\underline{1}} + \binom{m}{\underline{2}} + \dots + \binom{m}{\underline{r}},$$
vectors in $G_{RM}(r,m)$

So we have to write all these as rows are generator matrix up to product of degree r

(Refer Slide Time: 11:57)

Reed-Muller code

• Let $m = 4$, and $r = 2$, the second-order RM code of length $n = 16$ is generated by the following 11 vectors:

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_1
v_2	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1	v_2
v_3	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1	v_3
v_4	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1	v_4
$v_1 v_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1	$v_1 v_2$
$v_1 v_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1	$v_1 v_3$
$v_1 v_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1	$v_1 v_4$
$v_2 v_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1	$v_2 v_3$
$v_2 v_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1	$v_2 v_4$
$v_3 v_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1	$v_3 v_4$

So this your v_0 vector, these are all your $v_1 v_2 v_3 v_4$, this is degree 1, and then these are all possible degree 2 Boolean product vectors, because m is 4 so we will have $v_1 v_2 v_3 v_4$ and r is 2 so we have to consider all possible Boolean products of degree 2 so that would be $v_1 v_2, v_1 v_3, v_1 v_4, v_2 v_3, v_2 v_4, v_3 v_4$ and that's what we have listed here and of course you have your

(Refer Slide Time: 12:48)

Reed-Muller code

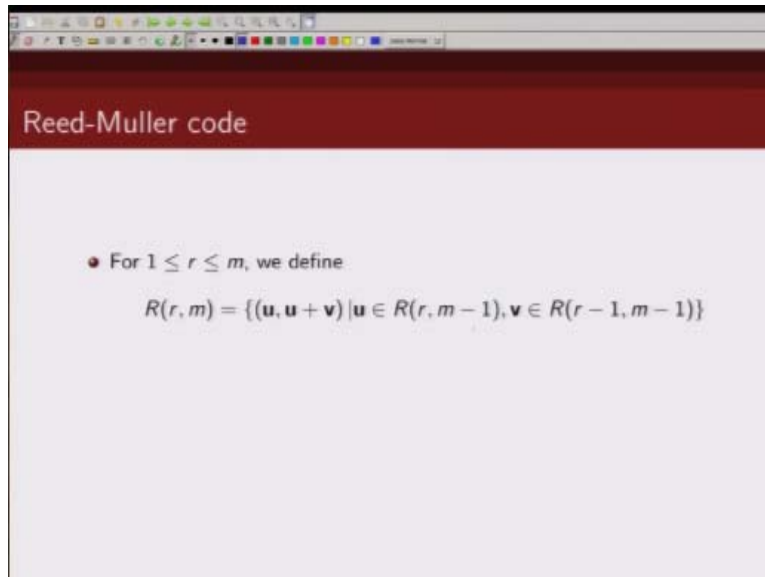
Let $m = 4$, and $r = 2$, the second-order RM code of length $n = 16$ is generated by the following 11 vectors:

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

11 x 16

All one pattern and these so what you are going to do is you are going to arrange these as rows of your generator matrix, so this is your 11 x 16 generator matrix okay and we will use this to generate our set of code words. Now there is another alternative construction of Reed-Muller code

(Refer Slide Time: 13:19)



Reed-Muller code

- For $1 \leq r \leq m$, we define

$$R(r, m) = \{(u, u + v) \mid u \in R(r, m-1), v \in R(r-1, m-1)\}$$

So if you are given Reed-Muller code of length 2^{m-1} then you can use two of them to construct a Reed-Muller code

(Refer Slide Time: 13:36)

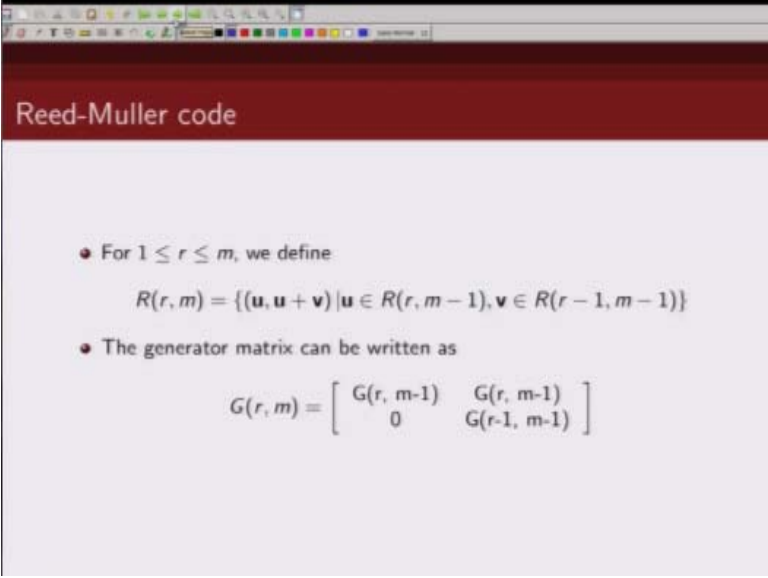
Reed-Muller code

• For $1 \leq r \leq m$, we define

$$R(r, m) = \{(u, u+v) \mid \begin{array}{c} \overset{2^{m-1}}{u \in R(r, m-1)} \quad \overset{2^{m-1}}{v \in R(r-1, m-1)} \\ \left[\begin{array}{cc} u & u+v \\ \hline 2^{m-1} & 2^{m-1} \end{array} \right] \end{array} \}$$

Of length 2^m so how do you do that, so this is done in this particular fashion so if you have two Reed-Muller code so one Reed-Muller code of order R and length 2^{m-1} and you have another Reed-Muller code of order $R-1$ and length 2^{m-1} then these two can be used to construct a Reed-Muller code of order R and length 2^m , and in this particular way so first so you can so if this is this is one code of length 2^{m-1} and some other code of length 2^{m-1} this is your code u which is order R and this is $u+v$ where u is given by this and v is given by this, so in other words you can construct Reed-Muller code recursively from smaller order and smaller length.

(Refer Slide Time: 14:48)



Reed-Muller code

- For $1 \leq r \leq m$, we define
$$R(r, m) = \{(\mathbf{u}, \mathbf{u} + \mathbf{v}) \mid \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$
- The generator matrix can be written as
$$G(r, m) = \begin{bmatrix} G(r, m-1) & G(r, m-1) \\ 0 & G(r-1, m-1) \end{bmatrix}$$

Code, the same thing I can I am writing in terms of generator matrix so as I said

(Refer Slide Time: 14:55)

Reed-Muller code

- For $1 \leq r \leq m$, we define

$$R(r, m) = \{(\mathbf{u}, \mathbf{u} + \mathbf{v}) \mid \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$
- The generator matrix can be written as

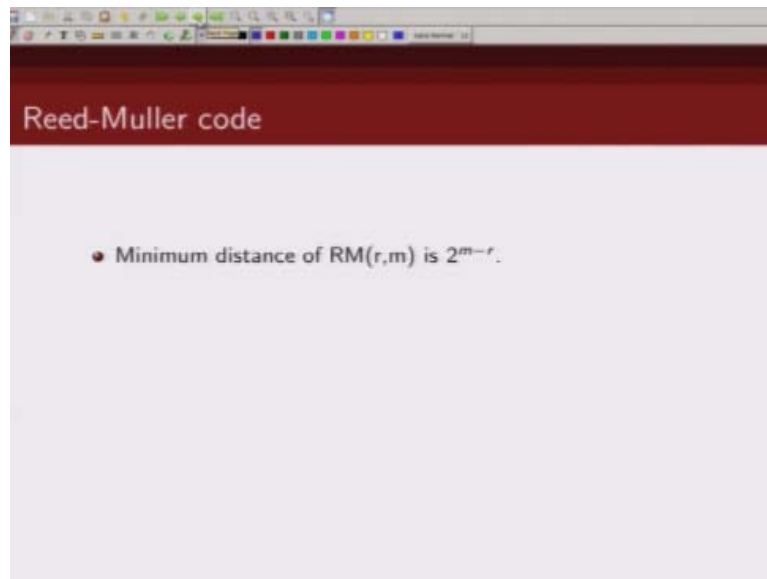
$$G(r, m) = \begin{bmatrix} G(r, m-1) & G(r, m-1) \\ 0 & G(r-1, m-1) \end{bmatrix}$$

2^m

2^{m-1}
 2^{m-1}

This is a Reed-Muller code of length 2^{m-1} , this is another Reed-Muller code of length 2^{m-1} , first is just \mathbf{u} which is this, this code Reed-Muller code order R length to 2^{m-1} and the second is this so this is your \mathbf{u} which is this, and the next one this is your \mathbf{v} which is this. So I can write down so in other words I can construct Reed-Muller code recursively from smaller length Reed-Muller code, this is another way of generating the generator matrix for the Reed-Muller code.

(Refer Slide Time: 15:46)



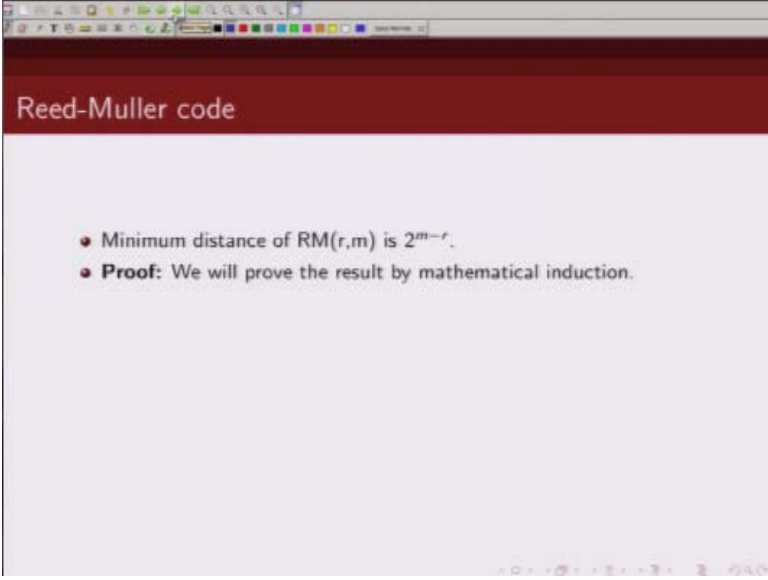
So let us prove some of the properties of Reed-Muller code, the first property that

(Refer Slide Time: 15:53)



We are going to prove is that minimum distance of Reed-Muller code is 2^{m-r}

(Refer Slide Time: 16:03)

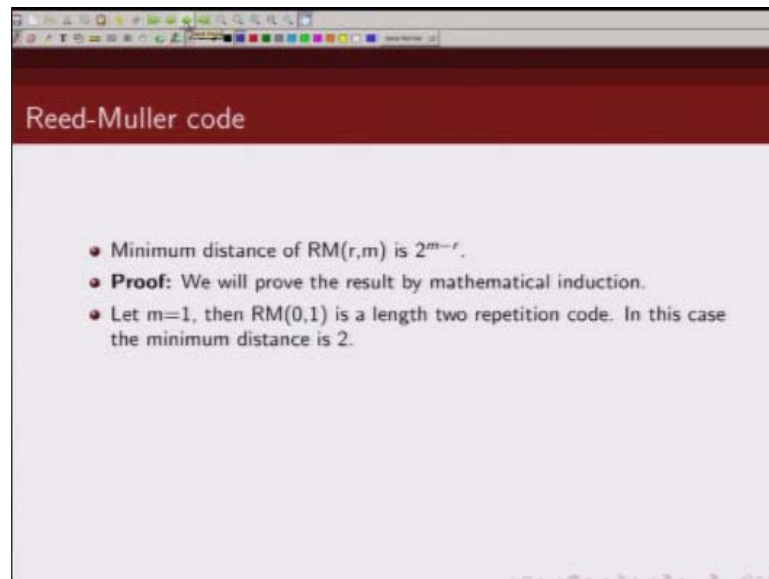


Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.

We are going to prove this result using mathematical induction.

(Refer Slide Time: 16:07)

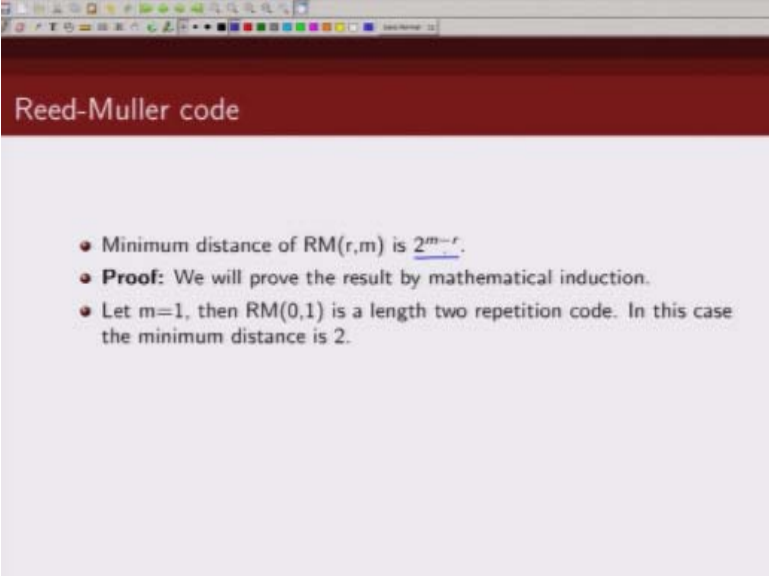


Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2.

So how does this work, so first we assume m to be one and let us check whether this minimum distance holds

(Refer Slide Time: 16:19)

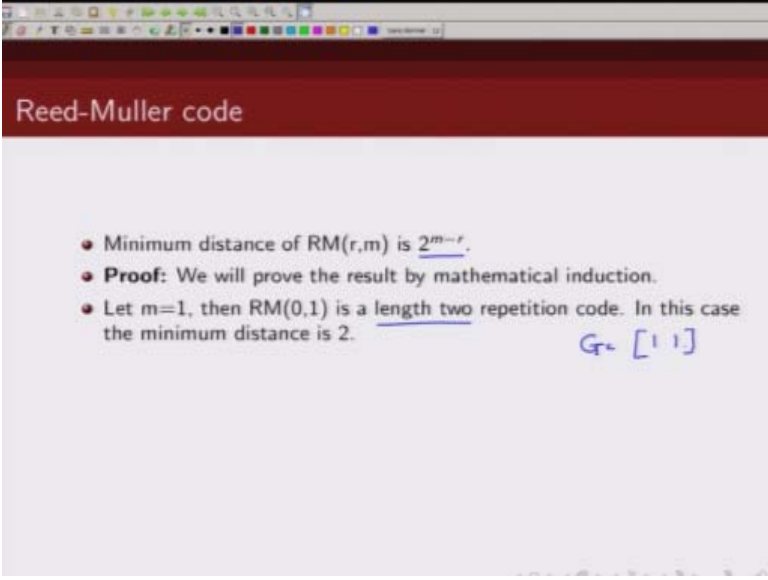


Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2.

Correct for $m = 1$ so for $m=1$ let us consider two scenarios one where r is zero and in second case r is 1. So when m is 1 what is the length of the Reed-Muller code it is 2^m so that is length is two.

(Refer Slide Time: 16:40)



Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2. $G = [1 \ 1]$

Okay and when order is zero so G will consist of only v_0 which is 11 so the Reed-Muller code of order zero and $m=1$ is essentially a length two repetition code and what is the minimum distance of this code it is two. So let us plug that

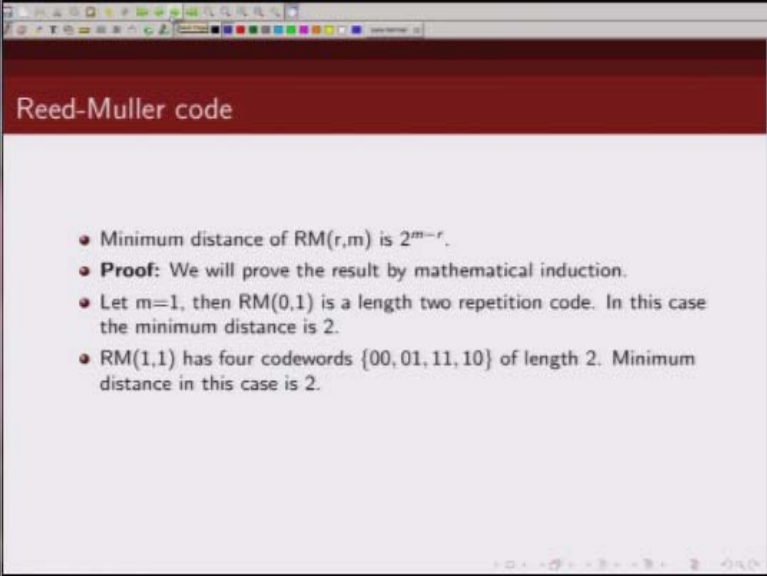
(Refer Slide Time: 17:07)

Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} . $\xrightarrow{m=1, r=0} 2$
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2. $G = [1 \ 1]$

In here and see whether this is correct m in our case is 1 and r is 0 so this gives us minimum distance of 2 and that is precisely what we are getting, so this whole proof for $m=1$ and $r=0$

(Refer Slide Time: 17:25)

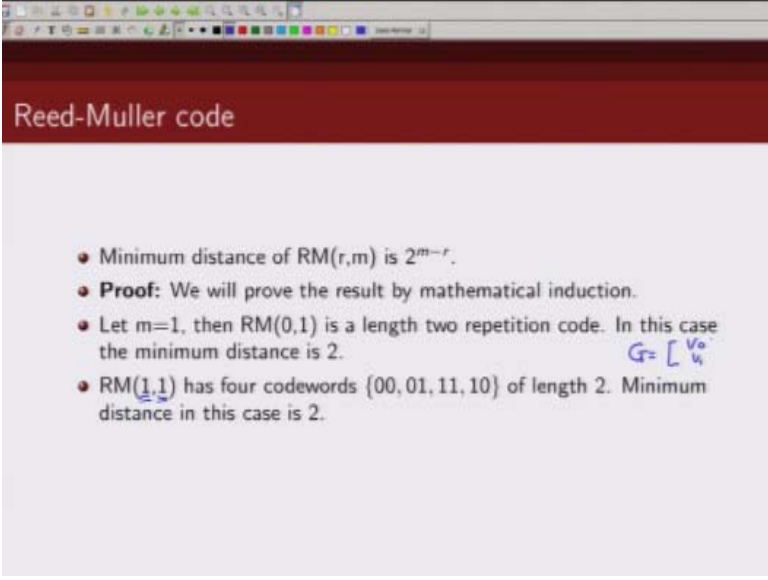


Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, 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.

Now let us say if it holds true also for

(Refer Slide Time: 17:28)

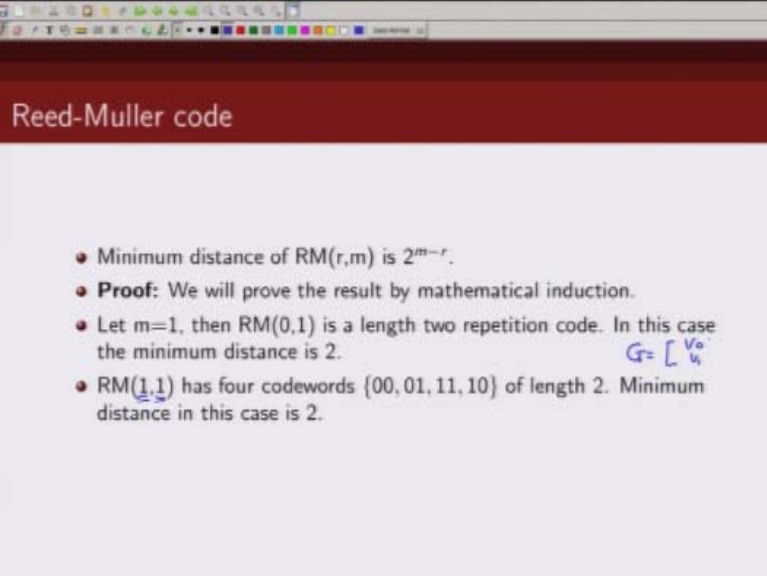


Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2. $G = \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}$
- $RM(1,1)$ has four codewords $\{00, 01, 11, 10\}$ of length 2. Minimum distance in this case is 2.

$M=1$ and $R=1$, now if $M=1$ and $R=1$ so then the length of the code word is again 2 so G will consist of v_0 and v_1 okay, and what is my

(Refer Slide Time: 17:49)



Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2.
- $RM(\underline{1},\underline{1})$ has four codewords $\{00, 01, 11, 10\}$ of length 2. Minimum distance in this case is 2.

$G = \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}$

v_0 and v_1

(Refer Slide Time: 17:50)

Reed-Muller code

- For $1 \leq i \leq m$, let \mathbf{v}_i be a binary 2^m -tuple of the following form:

$$\mathbf{v}_i = \left(\underbrace{0 \dots 0}_{2^{i-1}}, \underbrace{1 \dots 1}_{2^{i-1}}, \underbrace{0 \dots 0}_{2^{i-1}}, \dots, \underbrace{1 \dots 1}_{2^{i-1}} \right)$$
 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.

\mathbf{v}_1	=	(0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1)	$2^{i-1} = 1$
\mathbf{v}_2	=	(0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1)	$2^{i-1} = 2$
\mathbf{v}_3	=	(0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1)	$2^{i-1} = 4$
\mathbf{v}_4	=	(0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1)	$2^{i-1} = 8$

\mathbf{v}_1 is 0101 and \mathbf{v}_0 is 1 so this length 2 so what I will get is

(Refer Slide Time: 18:00)

Reed-Muller code

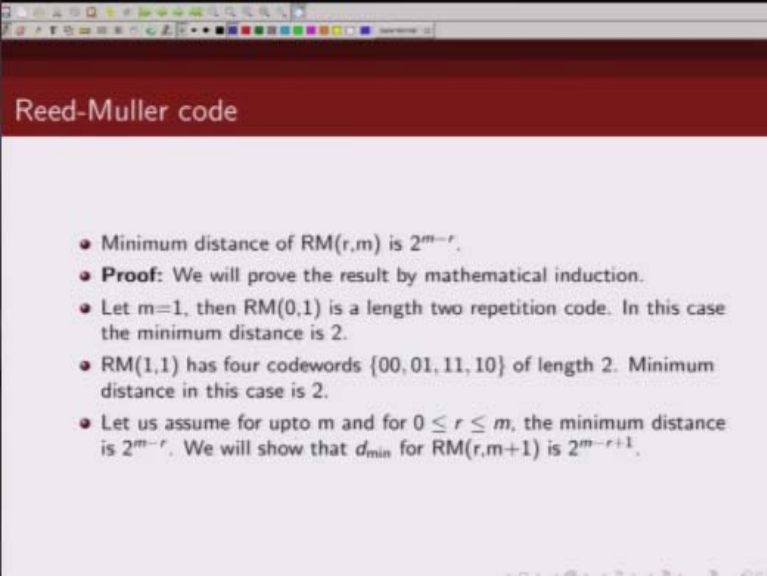
$m=1$

$m=1, r=1$

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2. $G = \begin{bmatrix} 1 & 1 \end{bmatrix}$
- $RM(1,1)$ has four codewords $\{00, 01, 11, 10\}$ of length 2. Minimum distance in this case is 1. $G = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

G is 11 and this is 01 so this will be my generator matrix, now this will generate these following code words of length 2 and what is the minimum distance between these codes that is 1 we can say minimum rate code word is minimum weight of nonzero code word is 1. So minimum distance in this case is 1 okay and let us check, so in this case m is 1 and r is 1 so $2^{1-1} 2^0$ that is 1 and that is what we are getting fine. So then this is true for $m=1$ now, let us assume is true for any $m=m$ and then we will try to prove that it is also true for $m=m+1$ so let us assume that this is true for.

(Refer Slide Time: 19:09)



Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, 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.
- Let us assume for upto m and for $0 \leq r \leq m$, the minimum distance is 2^{m-r} . We will show that d_{\min} for $RM(r,m+1)$ is 2^{m-r+1} .

Up to

(Refer Slide Time: 19:09)

Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, 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.
- Let us assume for upto m and for $0 \leq r \leq m$, the minimum distance is 2^{m-r} . We will show that d_{\min} for $RM(r,m+1)$ is 2^{m-r+1} .

m and for any order where order can be from zero to m let us assume that this is true so minimum distance is given by 2^{m-r} . Now what we are going to show is that this is also true for $m+1$ and what should be the minimum distance for $m+1$ it should be 2^{m+1-r} , so that is this. So next what we are going to show you is that minimum distance of m r^{th} order Reed-Muller code

(Refer Slide Time: 19:50)

Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, 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.
- Let us assume for upto m and for $0 \leq r \leq m$, the minimum distance is 2^{m-r} . We will show that d_{\min} for $RM(r,m+1)$ is 2^{m-r+1} .

RM +1 Reed-Muller code is basically given by this, now to prove this we are going to make use of this construction of Reed-Muller code

(Refer Slide Time: 20:01)

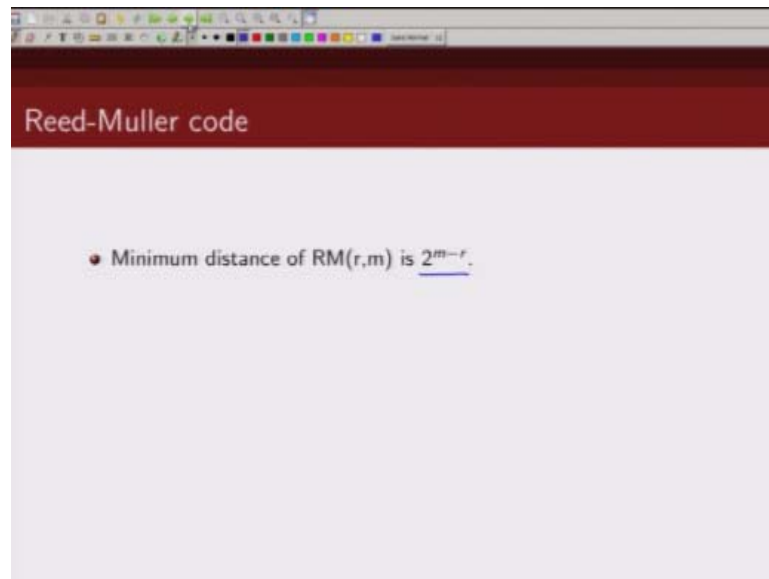
Reed-Muller code

- For $1 \leq r \leq m$, we define
$$R(r, m) = \{(\mathbf{u}, \mathbf{u} + \mathbf{v}) \mid \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$
- The generator matrix can be written as
$$G(r, m) = \begin{bmatrix} G(r, m-1) & G(r, m-1) \\ 0 & G(r-1, m-1) \end{bmatrix}$$

$\underbrace{\hspace{1.5cm}}_{2^{m-1}} \quad \underbrace{\hspace{1.5cm}}_{2^{m-1}}$

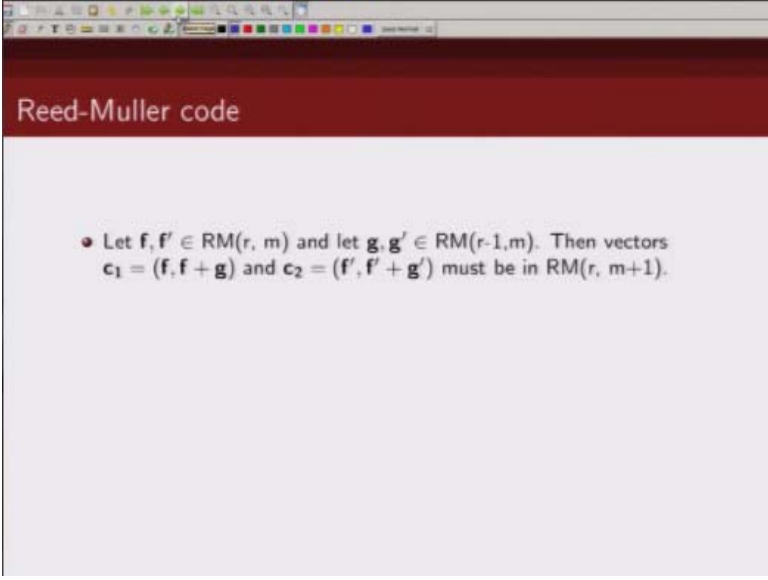
That Reed-Muller code of order r and m can be constructed recursively using this, we are going to make use of this construction to prove our result

(Refer Slide Time: 20:20)



So let us see how we proceed so let us consider

(Refer Slide Time: 20:24)

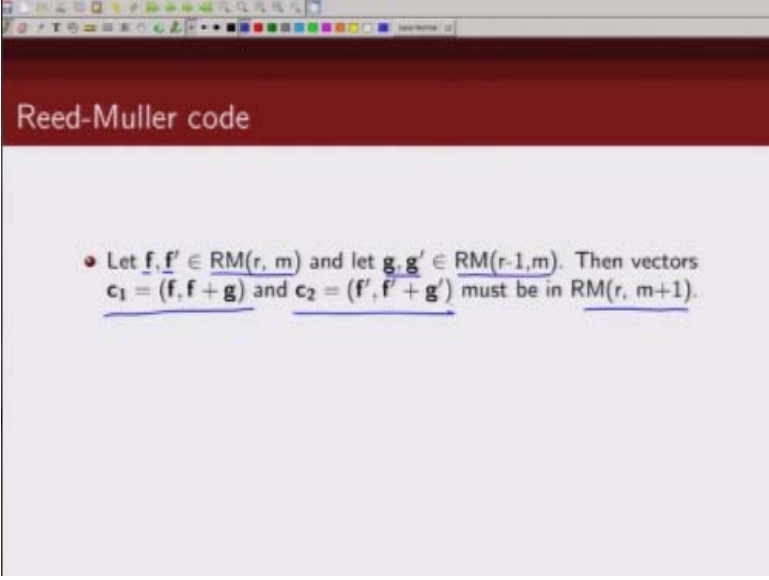


Reed-Muller code

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

Two code word

(Refer Slide Time: 20:25)

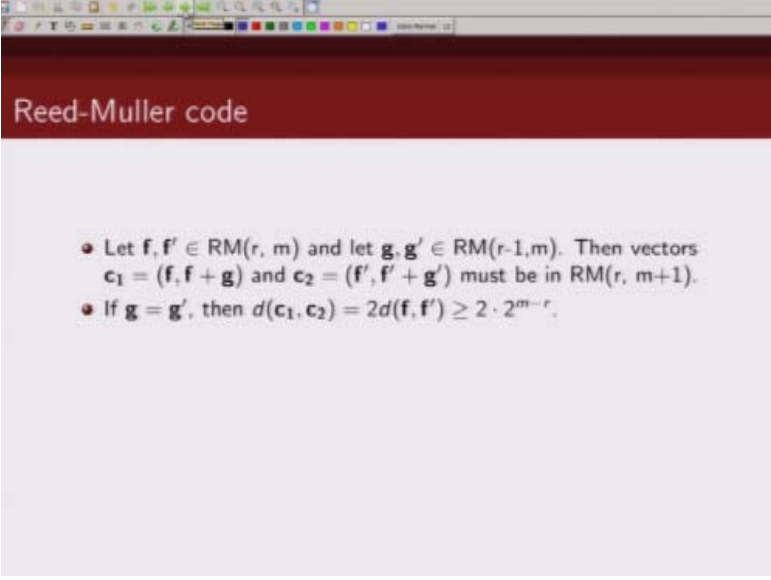


Reed-Muller code

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

f, f' which belongs to Reed-Muller code of order R and length 2^m and let g, g' belongs to Reed-Muller code of order $r-1$ and length 2^m , then we defining two code words then are Reed-Muller code of order R and length 2^{m+1} is of the form we just said u and $u+1$, so these code words and c_1 and c_2 which is of the form f and $f + g, f', f' + g$ they must be code word belonging to this Reed - Muller code and this follows from our recursive construction of Reed-Muller code which we just motioned so c_1 and c_2 must be code words for this Reed-Muller code

(Refer Slide Time: 21:26)

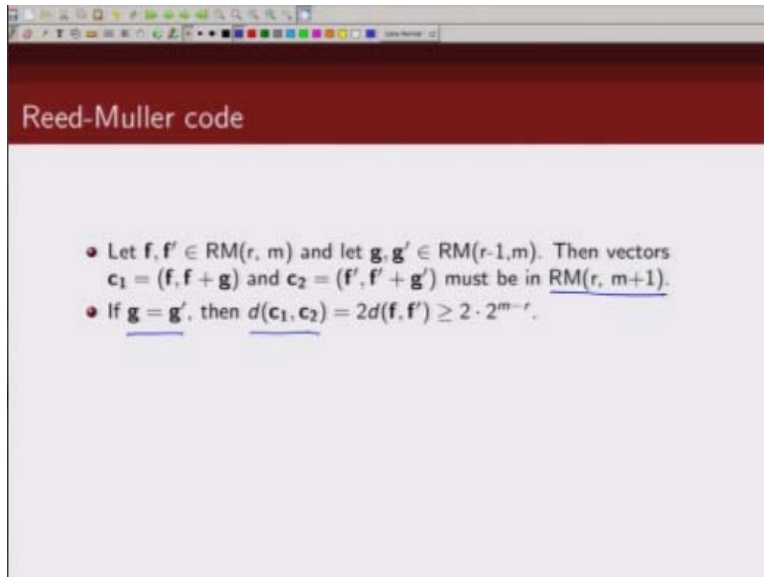


Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.

Now let us try to compute the minimum distance between these codes c_1 and c_2

(Refer Slide Time: 21:35)



Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f+g)$ and $c_2 = (f', f'+g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.

Which are code words Reed-Muller code of order r and length 2^{m+1} , so first case that we will consider is when g is same as g' and second case that we will consider is when G is not same as g' , so when g is same as g' what is the minimum distance between c_1 and c_2 ? Now if g and g' are same then basically your code c_1 is nothing but it is f here.

(Refer Slide Time: 22:06)

Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f+g)$ and $c_2 = (f', f'+g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$ then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r} = 2^{m+1-r}$.

$$c_1 = \left[\frac{f}{2^n} : \frac{f+g}{2^n} \right]$$

$$c_2 = \left[\frac{f'}{2^n} : \frac{f'+g'}{2^n} \right]$$

Of length 2^m and there is another code word f of length 2^m and C_2 is f' of length 2^m and then you have f' of length 2^m , so what is the minimum distance between this code? It is minimum distance between f and f' plus minimum distance between f and f' , so that is what we are writing here. So if g is equal to g' the minimum distance between C_1 and C_2 is 2 times the minimum distance between f and f' .

And what is the minimum distance between f and f' ? f and f' belongs to Reed-Muller code of order r and length 2^m , so their minimum distance should be 2^{m-r} , so then from this we get that minimum distance between C_1 and C_2 which are two code words belonging to Reed-Muller code for order r and length 2^{m+1} this should be greater than equal to 2^{m+1-r} . So for this particular case we have shown.

(Refer Slide Time: 23:30)

Reed-Muller code

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

(Refer Slide Time: 23:31)

Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, 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.
- Let us assume for upto m and for $0 \leq r \leq m$, the minimum distance is 2^{m-r} . We will show that d_{\min} for $RM(r,m+1)$ is 2^{m-r+1} .

That minimum distance is indeed this, now we will also have to show.

(Refer Slide Time: 23:37)

Reed-Muller code

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

(Refer Slide Time: 23:38)

Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f+g)$ and $c_2 = (f', f'+g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$ then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r} = 2^{m+1-r}$.
 $d(c_1, c_2) \geq 2^{m+1-r}$

$$c_1 = \begin{bmatrix} f & f \end{bmatrix}$$

$$c_2 = \begin{bmatrix} f' & f' \end{bmatrix}$$

If g is not same as g' then also we have to show that minimum distance is at least this.

(Refer Slide Time: 23:47)

Reed-Muller code

- Let $\mathbf{f}, \mathbf{f}' \in \text{RM}(r, m)$ and let $\mathbf{g}, \mathbf{g}' \in \text{RM}(r-1, m)$. Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f} + \mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}' + \mathbf{g}')$ must be in $\text{RM}(r, m+1)$.
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2) = 2d(\mathbf{f}, \mathbf{f}') \geq 2 \cdot 2^{m-r}$.
- If $\mathbf{g} \neq \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2) = w(\mathbf{f} - \mathbf{f}') + w(\mathbf{g} - \mathbf{g}' + \mathbf{f} - \mathbf{f}')$.

(Refer Slide Time: 23:50)

Reed-Muller code

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

$$a = x+y \quad b = y \quad a+b = x$$

$$w(x+y) + w(y) \geq w(x) \quad w(x+y) \geq w(x) - w(y)$$

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f+g)$ and $c_2 = (f', f'+g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.

$$c_1 = [f : f+g]$$

$$c_2 = [f' : f'+g']$$

So next we consider the case when g is not same as g' , now if g is not same as g' then weight minimum distance of the code we can say basically number of positions where c_1 and c_2 are differing this can be written as $w(f - f') + w(g - g' + f - f')$ if we are talking about binary codes this will be basically plus this also fine because that is the same thing. So if you have two code words just call it c_1 which is f here and this is $f + g$ and then you have c_2 which is f' , $f' + g'$ then the minimum distance between code is f minus weight of $f - f'$ and weight of this minus this. So that is what we are writing here, that minimum distance between c_1 and c_2 is given by this plus this.

Now we also know that let us say if we have two n – tuples then $w(a) + w(b)$ where a and b are some n – tuples, this is basically $w(a) + w(b)$ is greater than equal to $w(a + b)$, right? Now if I consider 'a' to be $x + y$ and 'b' to be y and let us say $x + y$ they are all binary n – tuples we are talking about, then $a + b$ will be $x + y$ plus y so that is given by x .

So what we will get is $w(x + y) + w(y)$ is greater than equal to $w(x)$, right? Or we can write $w(x + y)$ is greater than equal to $w(x) - w(y)$. Next we are going to make use of this result to simplify

this expression, this you can consider this is my x and this is my y . So I can write $w(x + y)$ to be greater than equal to $w(x) - w(y)$.

(Refer Slide Time: 26:29)

Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.
- Since $w(x + y) \geq w(x) - w(y)$, we have

$$d(c_1, c_2) \geq w(f - f') + w(g - g') - w(f - f') = w(g - g')$$

(Refer Slide Time: 26:34)

Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.
- Since $w(x + y) \geq w(x) - w(y)$, we have $\frac{x}{y} \geq w(x) - w(y)$.

$$\underline{d(c_1, c_2)} \geq \underline{w(f - f')} + \underline{w(g - g')} - \underline{w(f - f')} = \underline{w(g - g')}$$

So when I do that then distance minimum distance between c_1 and c_2 is this term coming here and what did I do? This was $w(x)$ this is x this was y this I can write as this is greater than equal to $w(x) - w(y)$. So this weight of x is this term minus $w(y)$ which is this term, fine? So now this, this cancels out what I get is $w(g - g')$.

(Refer Slide Time: 27:08)

Reed-Muller code

- Let $\mathbf{f}, \mathbf{f}' \in \text{RM}(r, m)$ and let $\mathbf{g}, \mathbf{g}' \in \text{RM}(r-1, m)$. Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f} + \mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}' + \mathbf{g}')$ must be in $\text{RM}(r, m+1)$.
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2) = 2d(\mathbf{f}, \mathbf{f}') \geq 2 \cdot 2^{m-r}$.
- If $\mathbf{g} \neq \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2) = w(\mathbf{f} - \mathbf{f}') + w(\mathbf{g} - \mathbf{g}' + \mathbf{f} - \mathbf{f}')$.
- Since $w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x}) - w(\mathbf{y})$, we have

$$d(\mathbf{c}_1, \mathbf{c}_2) \geq w(\mathbf{f} - \mathbf{f}') + w(\mathbf{g} - \mathbf{g}') - w(\mathbf{f} - \mathbf{f}') = w(\mathbf{g} - \mathbf{g}')$$

- Since $\mathbf{g} - \mathbf{g}' \in \text{RM}(r-1, m)$, so that $w(\mathbf{g} - \mathbf{g}') \geq 2^{m-(r-1)} = 2^{m-r+1}$

(Refer Slide Time: 27:13)

Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.
- Since $w(x + y) \geq w(x) - w(y)$, we have
$$d(c_1, c_2) \geq w(f - f') + w(g - g') - w(f - f') = w(g - g')$$
- Since $g - g' \in \text{RM}(r-1, m)$, so that $w(g - g') \geq 2^{m-(r-1)} = 2^{m-r+1}$

Now what is g ? g belongs to Reed-Muller code of order $r - 1$ and length 2^m .

(Refer Slide Time: 27:21)

Reed-Muller code

$$2^{m-(r-1)} = 2^{m+1-r}$$

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.
- Since $w(x + y) \geq w(x) - w(y)$, we have

$$d(c_1, c_2) \geq w(f - f') + w(g - g') - w(f - f') = w(g - g')$$
- Since $g - g' \in \text{RM}(r-1, m)$, so that $w(g - g') \geq 2^{m-(r-1)} = 2^{m-r+1}$

Then what is the minimum distance of this, so what is the minimum distance between g and g' ?
 This should be 2^{m-r} what is r ? The order here is $r-1$ so this is $r-1$. So this is 2^{m+1-r} . So what we have shown is.

(Refer Slide Time: 27:50)

Reed-Muller code

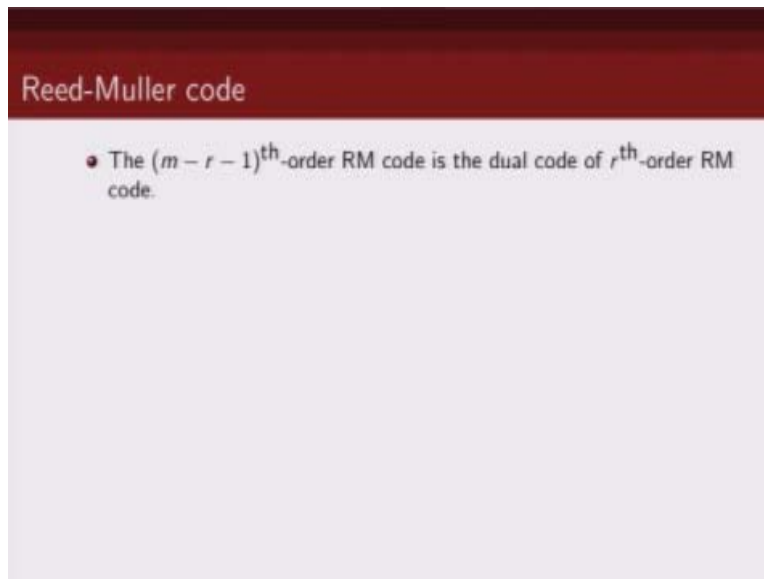
$$2^{m-(r-1)} = 2^{m+1-r}$$

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.
- Since $w(x + y) \geq w(x) - w(y)$, we have

$$d(c_1, c_2) \geq w(f - f') + w(g - g') - w(f - f') = w(g - g')$$
- Since $g - g' \in \text{RM}(r-1, m)$, so that $w(g - g') \geq 2^{m-(r-1)} = \underline{2^{m-r+1}}$

Even when g is not same as g' , our minimum distance is still 2^{m-r+1} . So now we have proved that minimum distance of r^{th} order Reed-Muller code of length 2^{m+1} is basically given by this.

(Refer Slide Time: 28:17)



Reed-Muller code

- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.

So this will conclude the proof using mathematical induction that the minimum distance of Reed – Muller code is 2^{m-r} . The next result which we are going to show you is.

(Refer Slide Time: 28:34)

Reed-Muller code

- The $(m-r-1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.

That $m-r^{\text{th}}$ order Reed–Muller code is the dual code of r^{th} order Reed–Muller code. So let us see this is our original code then the dual code is given by this, now what do we need to show for dual code, if we take a code word from this code and if we take a code word from the dual code they are orthogonal, right? So the dot product should be zero.

(Refer Slide Time: 29:05)

Reed-Muller code

- The $(m-r-1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.

Another point which I should mention here is.

(Refer Slide Time: 29:07)

Reed-Muller code

$$2^{m-(r-1)} = 2^{m+1-r}$$

- Let $\mathbf{f}, \mathbf{f}' \in \text{RM}(r, m)$ and let $\mathbf{g}, \mathbf{g}' \in \text{RM}(r-1, m)$. Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f} + \mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}' + \mathbf{g}')$ must be in $\text{RM}(r, m+1)$.
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2) = 2d(\mathbf{f}, \mathbf{f}') \geq 2 \cdot 2^{m-r}$.
- If $\mathbf{g} \neq \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2) = w(\mathbf{f} - \mathbf{f}') + w(\mathbf{g} - \mathbf{g}' + \mathbf{f} - \mathbf{f}')$.
- Since $w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x}) - w(\mathbf{y})$, we have

$$d(\mathbf{c}_1, \mathbf{c}_2) \geq w(\mathbf{f} - \mathbf{f}') + w(\mathbf{g} - \mathbf{g}') - w(\mathbf{f} - \mathbf{f}') = w(\mathbf{g} - \mathbf{g}')$$

- Since $\mathbf{g} - \mathbf{g}' \in \text{RM}(r-1, m)$, so that $w(\mathbf{g} - \mathbf{g}') \geq 2^{m-(r-1)} = \underline{2^{m-r+1}}$

(Refer Slide Time: 29:08)

Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.
- Since $w(x + y) \geq w(x) - w(y)$, we have $\frac{x}{y} \geq w(x) - w(y)$

$$d(c_1, c_2) \geq w(f - f') + w(g - g') - w(f - f') = w(g - g')$$

(Refer Slide Time: 29:09)

Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f+g)$ and $c_2 = (f', f'+g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$ then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.

$$d(c_1, c_2) \geq 2^{m+1-r}$$

$$c_1 = \begin{bmatrix} f \\ \frac{f}{2^m} \end{bmatrix}, \quad c_2 = \begin{bmatrix} f' \\ \frac{f'}{2^m} \end{bmatrix}$$

(Refer Slide Time: 29:10)

Reed-Muller code

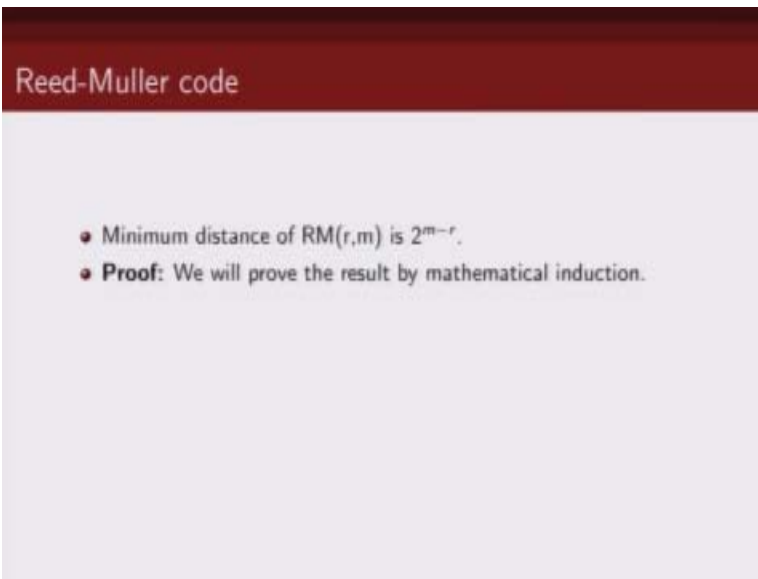
- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2.
- $RM(1,1)$ has four codewords $\{00, 01, 11, 10\}$ of length 2. Minimum distance in this case is 2.
- Let us assume for upto m and for $0 \leq r \leq m$, the minimum distance is 2^{m-r} . We will show that d_{\min} for $RM(r,m+1)$ is 2^{m-r+1} .

(Refer Slide Time: 29:11)

Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} . $\xrightarrow{m=1, r=0} 2$
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2. $G = \begin{bmatrix} 1 & 1 \end{bmatrix}$

(Refer Slide Time: 29:12)

A presentation slide with a dark red header and a light purple body. The header contains the text 'Reed-Muller code'. The body contains two bullet points: '• Minimum distance of RM(r,m) is 2^{m-r} .' and '• **Proof:** We will prove the result by mathematical induction.'

Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.

(Refer Slide Time: 29:13)

Reed-Muller code

- For $1 \leq r \leq m$, we define

$$R(r, m) = \{(\mathbf{u}, \mathbf{u} + \mathbf{v}) \mid \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$
- The generator matrix can be written as

$$G(r, m) = \begin{bmatrix} \underbrace{G(r, m-1)}_{2^{m-1}} & \underbrace{G(r, m-1)}_{2^{m-1}} \\ 0 & G(r-1, m-1) \end{bmatrix}$$

(Refer Slide Time: 29:14)

Reed-Muller code															
<ul style="list-style-type: none"> Let $m = 4$, and $r = 2$, the second-order RM code of length $n = 16$ is generated by the following 11 vectors: 															
v_0	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
v_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
v_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
v_4	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$v_1 v_2$	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0
$v_1 v_3$	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0
$v_1 v_4$	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0
$v_2 v_3$	0	0	0	0	0	1	1	0	0	0	0	0	1	1	0
$v_2 v_4$	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0
$v_3 v_4$	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0

Let us go back to our construction of Reed- Muller code here, please note the way these Boolean products are constructed, in fact we just proved also the minimum distance of the code is even, is 2^{m-r} . So minimum distance of Reed–Muller code is even so Reed – Muller code would not have odd weight code words.

(Refer Slide Time: 29:34)

Reed-Muller code

• Let $m = 4$, and $r = 2$, the second-order RM code of length $n = 16$ is generated by the following 11 vectors:

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

11 x 16

(Refer Slide Time: 29:35)

Reed-Muller code

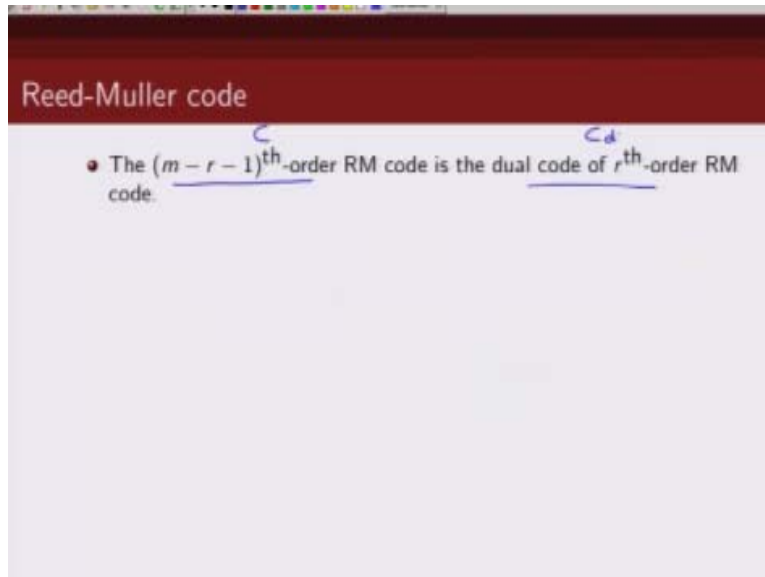
- For $1 \leq r \leq m$, we define

$$R(r, m) = \{(\mathbf{u}, \mathbf{u} + \mathbf{v}) \mid \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$

$\left[\begin{array}{c} \mathbf{u} \\ \hline 2^{m-1} \end{array} \mid \begin{array}{c} \mathbf{u} + \mathbf{v} \\ \hline 2^{m-1} \end{array} \right]$

Diagram illustrating the recursive construction of Reed-Muller codes. The set $R(r, m)$ is defined as the set of pairs $(\mathbf{u}, \mathbf{u} + \mathbf{v})$ where $\mathbf{u} \in R(r, m-1)$ and $\mathbf{v} \in R(r-1, m-1)$. The diagram shows a box containing the pair $(\mathbf{u}, \mathbf{u} + \mathbf{v})$ with dimensions 2^{m-1} and 2^{m-1} indicated above the components. Below the box, the components are shown as $\frac{\mathbf{u}}{2^{m-1}}$ and $\frac{\mathbf{u} + \mathbf{v}}{2^{m-1}}$.

(Refer Slide Time: 29:36)



So now we will show if we take a code word from $(m-r-1)^{\text{th}}$ order Reed-Muller code and if we take another code word from r^{th} order Reed-Muller code then they are orthogonal. That is the first thing we are going to prove.

(Refer Slide Time: 29:54)

Reed-Muller code

- The $(m-r-1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m-r-1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m-r-1$.

So let us consider a code word a , which belongs to $(m-r-1)$ th order Reed–Muller code which is of length 2^m and let us consider another Reed–Muller code ‘ b ’ which is of order r and length 2^m so ‘ a ’ can be viewed as a polynomial of degree $m-r-1$ or less.

(Refer Slide Time: 30:25)

Reed-Muller code

- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m - r - 1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m - r - 1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m - 1$.

And similarly the degree of the polynomial b is less than equal to r .

(Refer Slide Time: 30:31)

Reed-Muller code

- The $(m-r-1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m-r-1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m-r-1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m-1$.
- Therefore $ab \in RM(m-1, m)$ and has even weight. Therefore the dot product $a \cdot b = 0 \pmod{2}$.

So if we consider their product then this will be a polynomial of degree $m-r-1+r$ so that would be of degree less than or equal to $m-1$. So then this product a and b will belong to a Reed-Muller code of order $m-1$ and this is of length 2^m . Now note that Reed-Muller code has only even weight code words.

(Refer Slide Time: 31:14)

Reed-Muller code

- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m - r - 1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m - r - 1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m - 1$.
- Therefore $ab \in RM(m - 1, m)$ and has even weight. Therefore the dot product $a \cdot b = 0 \pmod{2}$.

So when we are considering this dot product $a \cdot b$ since Reed-Muller code has only even weight code word then $a \cdot b$ would be zero.

(Refer Slide Time: 31:24)

Reed-Muller code

- The $(m-r-1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m-r-1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m-r-1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m-1$.
- Therefore $ab \in RM(m-1, m)$ and has even weight. Therefore the dot product $a \cdot b = 0 \pmod{2}$.

So modulo 2 this will be zero. So in other words then what we have shown is if you take a code word 'a' which belongs to $(m-r-1)$ th order Reed–Muller code and if you take another code word which belongs to r th order Reed–Muller code then they are orthogonal to each other.

(Refer Slide Time: 31:52)

Reed-Muller code

- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m - r - 1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m - r - 1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m - 1$.
- Therefore $ab \in RM(m - 1, m)$ and has even weight. Therefore the dot product $a \cdot b = 0 \pmod{2}$.
- Also, $\dim RM(m - r - 1, m) + \dim RM(r, m)$

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

$$= 2^m$$

which implies that $RM(m - r - 1) = RM(r, m)^\perp$.

Next we check the dimension of $(m - r - 1)^{\text{th}}$.

(Refer Slide Time: 31:55)

Reed-Muller code

- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m - r - 1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m - r - 1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m - 1$.
- Therefore $ab \in RM(m - 1, m)$ and has even weight. Therefore the dot product $a \cdot b = 0 \pmod{2}$.
- Also, $\dim RM(m - r - 1, m) + \dim RM(r, m)$

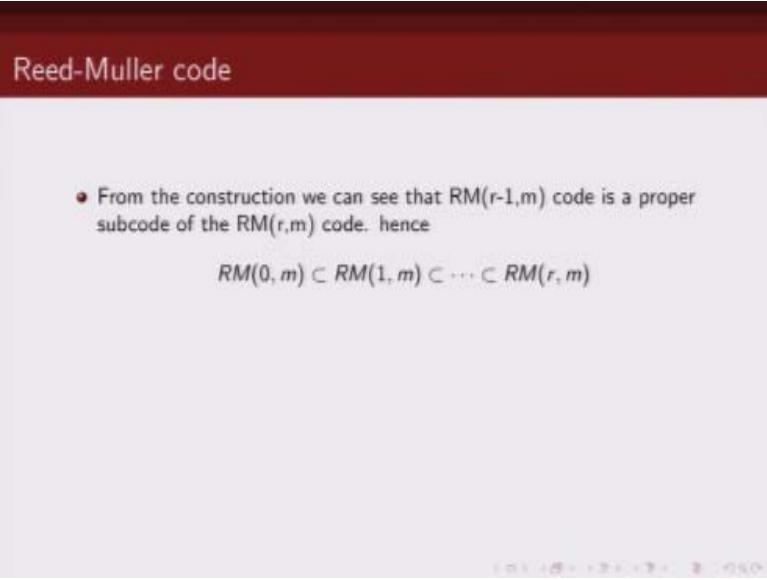
$$= 1 + \binom{m}{1} + \dots + \binom{m}{m - r - 1} + 1 + \binom{m}{1} + \dots + \binom{m}{r}$$

$$= 2^m$$

which implies that $RM(m - r - 1, m) = RM(r, m)^\perp$.

Order Reed-Muller code and r^{th} order Reed-Muller code and we see that some of the dimension is 2^m which is a length of the code word. So this does prove then that $(m-r-1)^{\text{th}}$ order Reed-Muller code this just radon m here, is dual to r^{th} order Reed-Muller code.

(Refer Slide Time: 32:29)



Reed-Muller code

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

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

Now let us say that some of the codes that we have studied are actually a special case of Reed–Muller code. So the first thing which is clear from the construction is.

(Refer Slide Time: 32:46)

Reed-Muller code

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

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

That any $r-1$ order Reed–Muller code is a proper sub code of an r^{th} order Reed–Muller code.

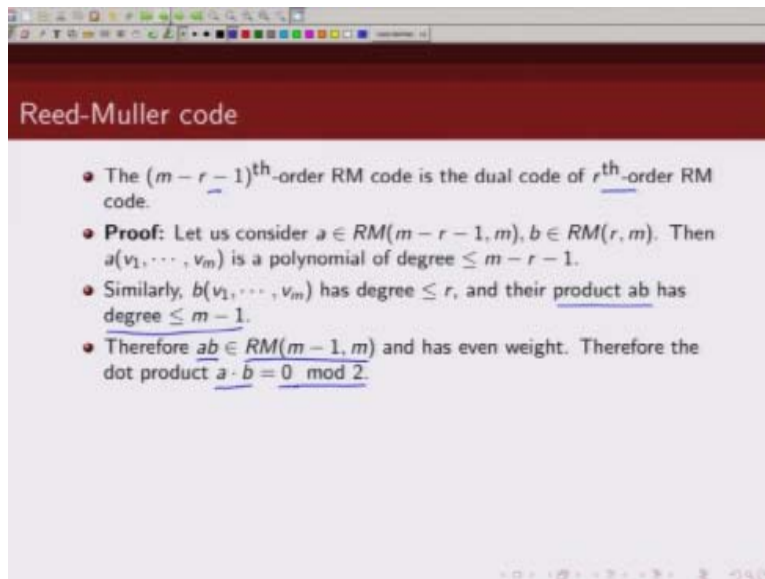
(Refer Slide Time: 32:56)

Reed-Muller code

- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m - r - 1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m - r - 1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m - 1$.
- Therefore $ab \in RM(m - 1, m)$ and has even weight. Therefore the dot product $a \cdot b = 0 \pmod{2}$.

And this is easy to see if you noticed and go back to our code construction, what was our generator matrix? Our generator matrix consists of these tuples.

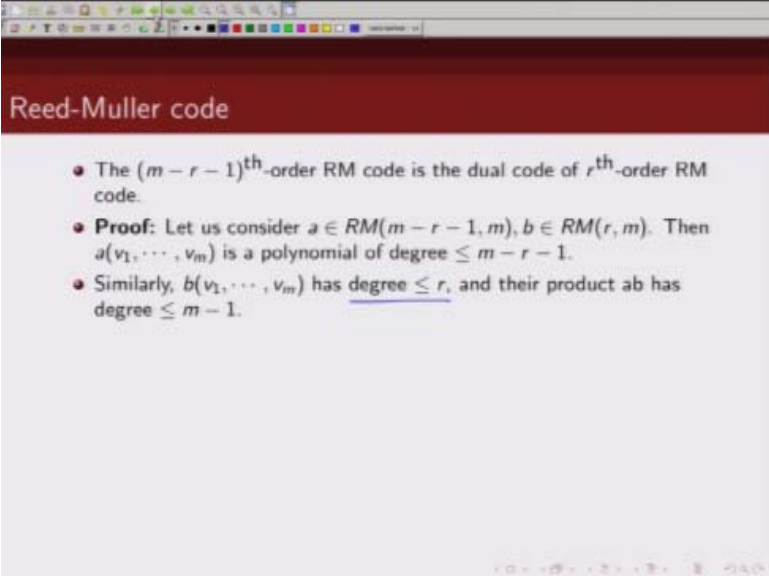
(Refer Slide Time: 32:57)



Reed-Muller code

- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m - r - 1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m - r - 1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m - 1$.
- Therefore $ab \in RM(m - 1, m)$ and has even weight. Therefore the dot product $a \cdot b = 0 \pmod{2}$.

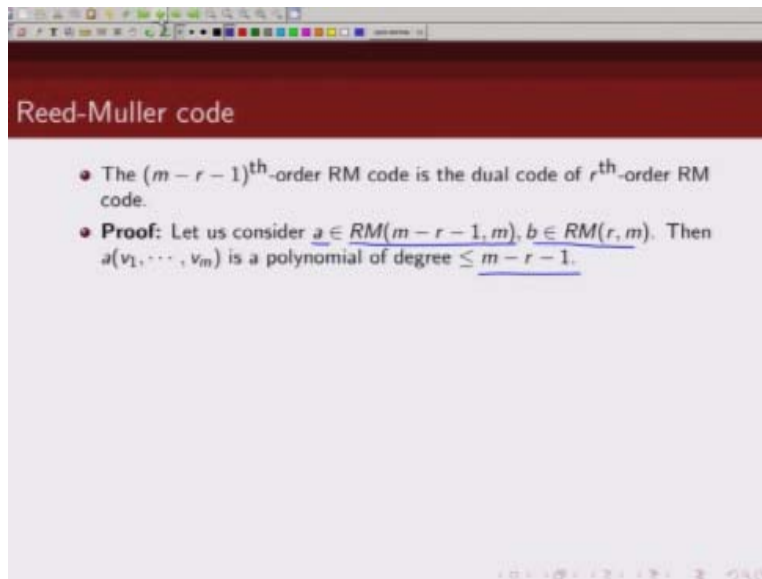
(Refer Slide Time: 32:57)



Reed-Muller code

- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m - r - 1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m - r - 1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m - 1$.

(Refer Slide Time: 32:57)



Reed-Muller code

- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m - r - 1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m - r - 1$.

(Refer Slide Time: 32:58)

Reed-Muller code

- The $(m-r-1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.

(Refer Slide Time: 32:58)

Reed-Muller code

$$2^{m-(r-1)} = 2^{m+1-r}$$

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.
- Since $w(x + y) \geq w(x) - w(y)$, we have

$$d(c_1, c_2) \geq w(f - f') + w(g - g') - w(f - f') = w(g - g')$$
- Since $g - g' \in \text{RM}(r-1, m)$, so that $w(g - g') \geq 2^{m-(r-1)} = \underline{2^{m-r+1}}$

(Refer Slide Time: 32:59)

Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.
- Since $w(x + y) \geq w(x) - w(y)$, we have $\frac{x}{y} \geq w(x) - w(y)$.

$$d(c_1, c_2) \geq w(f - f') + w(g - g') - w(f - f') = w(g - g')$$

(Refer Slide Time: 33:00)

Reed-Muller code

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

$$a = x+y \quad b = y \quad a+b = x$$

$$w(x+y) + w(y) \geq w(x) \quad w(x+y) \geq w(x) - w(y)$$

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f+g)$ and $c_2 = (f', f'+g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.

$$c_1 = [f : f+g]$$

$$c_2 = [f' : f'+g']$$

(Refer Slide Time: 33:00)

Reed-Muller code

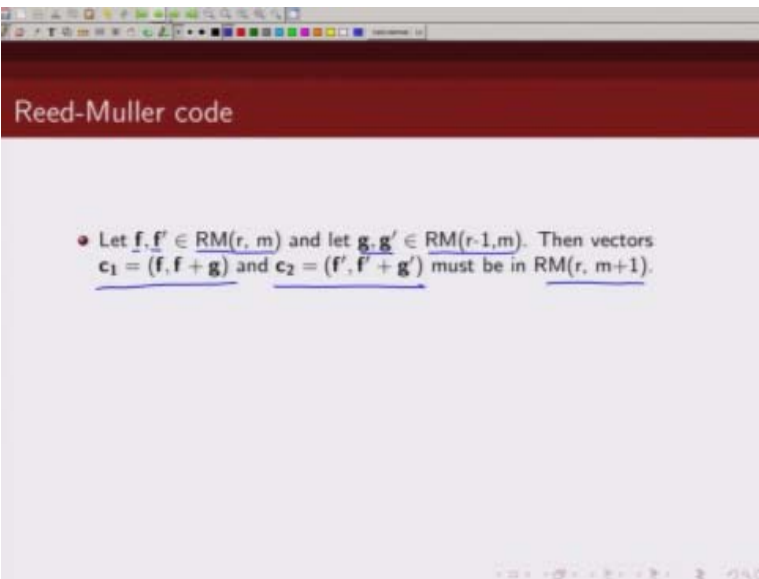
- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$ then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.

$$d(c_1, c_2) \geq 2^{m+1-r}$$

$$c_1 = \left[\frac{f}{2^m} : \frac{f}{2^m} \right]$$

$$c_2 = \left[\frac{f'}{2^m} : \frac{f'}{2^m} \right]$$

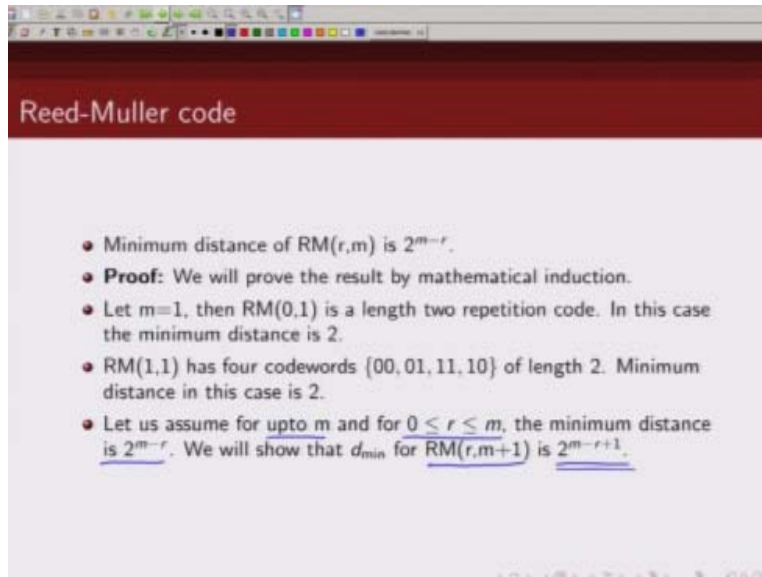
(Refer Slide Time: 33:00)



Reed-Muller code

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

(Refer Slide Time: 33:01)



Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, 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.
- Let us assume for upto m and for $0 \leq r \leq m$, the minimum distance is 2^{m-r} . We will show that d_{\min} for $RM(r,m+1)$ is 2^{m-r+1} .

(Refer Slide Time: 33:02)

Reed-Muller code

$m=1$

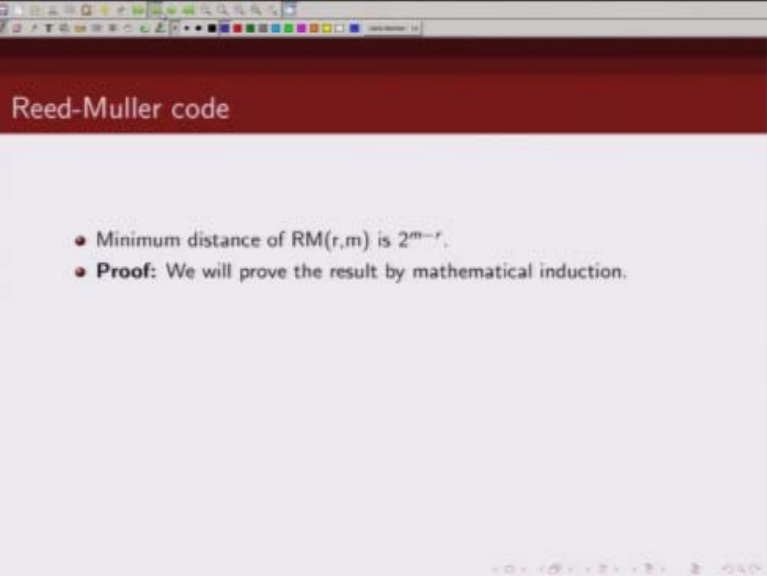
- Minimum distance of $RM(r,m)$ is 2^{m-r} . $m=1, r=1$
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2. $G = \begin{bmatrix} 1 & 1 \end{bmatrix}$
- $RM(1,1)$ has four codewords $\{00, 01, 11, 10\}$ of length 2. Minimum distance in this case is 1. $G = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

(Refer Slide Time: 33:02)

Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} . $m=1, r=0 \rightarrow 2$
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2. $G = \begin{bmatrix} 1 & 1 \end{bmatrix}$

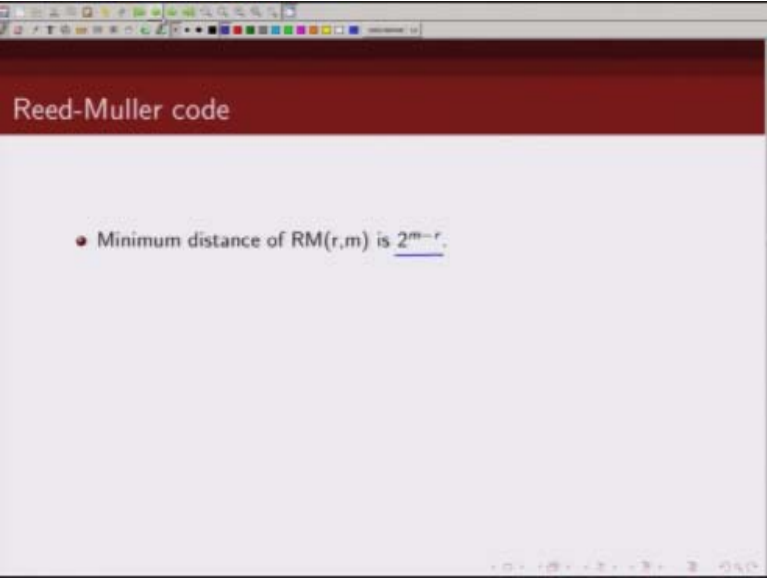
(Refer Slide Time: 33:03)



Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.

(Refer Slide Time: 33:03)



Reed-Muller code

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

(Refer Slide Time: 33:04)

Reed-Muller code

- For $1 \leq r \leq m$, we define

$$R(r, m) = \{(\mathbf{u}, \mathbf{u} + \mathbf{v}) \mid \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$

- The generator matrix can be written as

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

2^m 2^{m-1} 2^{m-1}

(Refer Slide Time: 33:04)

Reed-Muller code

- For $1 \leq r \leq m$, we define

$$R(r, m) = \{(\mathbf{u}, \mathbf{u} + \mathbf{v}) \mid \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$

- The generator matrix can be written as

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

2^m 2^{m-1} 2^{m-1}

(Refer Slide Time: 33:05)

Reed-Muller code

• Let $m = 4$, and $r = 2$, the second-order RM code of length $n = 16$ is generated by the following 11 vectors:

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

11 x 16

(Refer Slide Time: 33:06)

Reed-Muller code

- The r^{th} -order RM code, $RM(r, m)$, of length 2^m is generated by following set of independent vectors:
$$G_{RM}(r, m) = \{ \underline{v_0}, v_1, v_2, \dots, v_m, v_1v_2, v_1v_3, \dots, v_{m-1}v_m, \dots \}$$

up to products of degree r .
- There are
$$\underline{k(r, m)} = 1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r},$$

vectors in $G_{RM}(r, m)$

If you noticed and go back to our code construction what was our generator matrix? Our generator matrix consist of these tuples v_0, v_1, v_2 up to product of degree r . So if you are considering zeroth order Reed-Muller code this will only have v_0 . In the G matrix if you are considering first order Reed-Muller code.

(Refer Slide Time: 33:30)

Reed-Muller code

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

$$G_{RM}(r, m) = \{ \mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m, \mathbf{v}_1\mathbf{v}_2, \mathbf{v}_1\mathbf{v}_3, \dots, \mathbf{v}_{m-1}\mathbf{v}_m, \dots, \text{up to products of degree } r \}.$$

- There are

$$k(r, m) = 1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r},$$

vectors in $G_{RM}(r, m)$

It will have v_0 and it will also have v_1, v_2, v_3, v_m . If you are considering second order Reed-Muller code this will have this and it will have all these second order terms. So you can see that smaller order Reed-Muller code is already embedded in the.

(Refer Slide Time: 33:52)

Reed-Muller code

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

$$G_{RM}(r, m) = \{ \mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_m, \mathbf{v}_1\mathbf{v}_2, \mathbf{v}_1\mathbf{v}_3, \dots, \mathbf{v}_{m-1}\mathbf{v}_m, \dots \}$$

up to products of degree r .

- There are

$$k(r, m) = 1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r},$$

vectors in $G_{RM}(r, m)$

(Refer Slide Time: 33:52)

Reed-Muller code

- The r^{th} -order RM code, $RM(r, m)$, of length 2^m is generated by following set of independent vectors:
$$G_{RM}(r, m) = \{ \underline{v}_0, \underline{v}_1, \underline{v}_2, \dots, \underline{v}_m, \underline{v}_1 \underline{v}_2, \underline{v}_1 \underline{v}_3, \dots, \underline{v}_{m-1} \underline{v}_m, \dots \}$$

, up to products of degree r \}
- There are
$$k(r, m) = 1 + \binom{m}{1} + \binom{m}{2} + \dots + \binom{m}{r},$$

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.

(Refer Slide Time: 33:53)

Reed-Muller code

• Let $m = 4$, and $r = 2$, the second-order RM code of length $n = 16$ is generated by the following 11 vectors:

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

11 x 16

(Refer Slide Time: 33:54)

Reed-Muller code

• For $1 \leq r \leq m$, we define

$$R(r, m) = \left\{ (u, u+v) \mid \begin{array}{c} \xrightarrow{2^m} \quad \xrightarrow{2^{m-1}} \quad \xrightarrow{2^{m-1}} \\ u \in R(r, m-1) \quad v \in R(r-1, m-1) \end{array} \right\}$$

$\left[\begin{array}{c} u \\ \xrightarrow{2^{m-1}} \end{array} \quad \begin{array}{c} u+v \\ \xrightarrow{2^{m-1}} \end{array} \right]$

(Refer Slide Time: 33:54)

Reed-Muller code

- For $1 \leq r \leq m$, we define

$$R(r, m) = \{(\mathbf{u}, \mathbf{u} + \mathbf{v}) \mid \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$

- The generator matrix can be written as

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

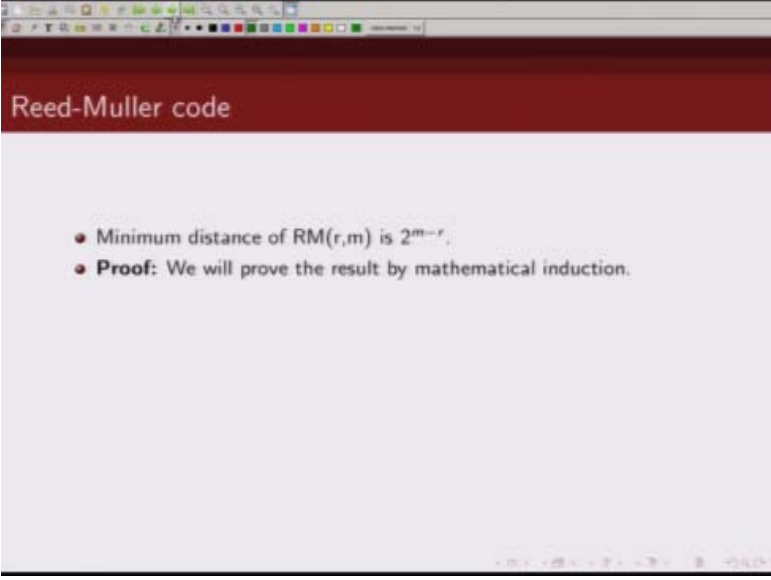
2^m 2^{m-1} 2^{m-1}

(Refer Slide Time: 33:54)

Reed-Muller code

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

(Refer Slide Time: 33:55)



Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.

(Refer Slide Time: 33:56)

Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} . $\rightarrow 2$ $m=1, r=0$
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2. $G = \begin{bmatrix} 1 & 1 \end{bmatrix}$

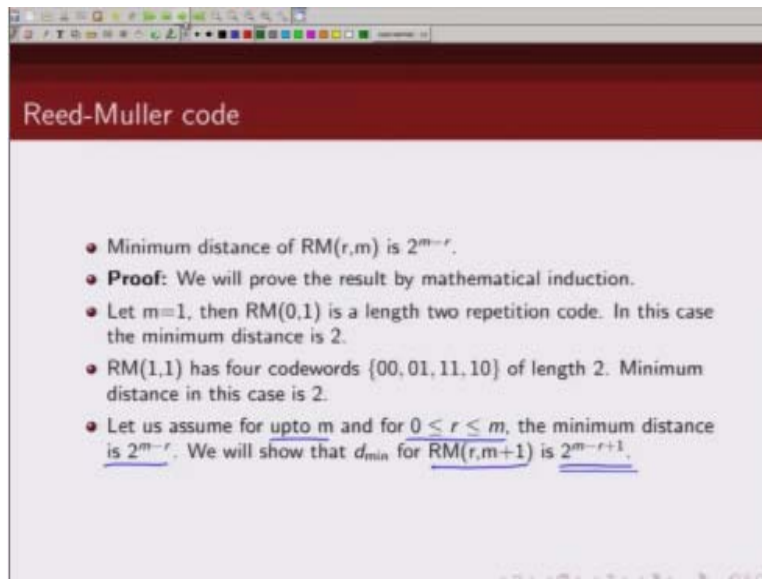
(Refer Slide Time: 33:56)

Reed-Muller code

$m=1$

- Minimum distance of $RM(r,m)$ is 2^{m-r} . $m=1, r=1$
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, then $RM(0,1)$ is a length two repetition code. In this case the minimum distance is 2. $G = \begin{bmatrix} 1 & 1 \end{bmatrix}$
- $RM(1,1)$ has four codewords $\{00, 01, 11, 10\}$ of length 2. Minimum distance in this case is 1. $G = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

(Refer Slide Time: 33:57)



Reed-Muller code

- Minimum distance of $RM(r,m)$ is 2^{m-r} .
- **Proof:** We will prove the result by mathematical induction.
- Let $m=1$, 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.
- Let us assume for upto m and for $0 \leq r \leq m$, the minimum distance is 2^{m-r} . We will show that d_{min} for $RM(r,m+1)$ is 2^{m-r+1} .

(Refer Slide Time: 33:57)

Reed-Muller code

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

(Refer Slide Time: 33:58)

Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$ then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.

$$d(c_1, c_2) \geq 2^{m+1-r}$$

$$c_1 = \left[\frac{f}{2^m} : \frac{f}{2^m} \right]$$

$$c_2 = \left[\frac{f'}{2^m} : \frac{f'}{2^m} \right]$$

(Refer Slide Time: 33:58)

Reed-Muller code

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

$$a = x+y \quad b = y \quad a+b = x$$

$$w(x+y) + w(y) \geq w(x) \quad w(x+y) \geq w(x) - w(y)$$

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f+g)$ and $c_2 = (f', f'+g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f-f') + w(g-g' + f-f')$.

$$c_1 = [f : f+g]$$

$$c_2 = [f' : f'+g']$$

(Refer Slide Time: 33:58)

Reed-Muller code

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.
- Since $w(x + y) \geq w(x) - w(y)$, we have $\frac{x}{y} \geq w(x) - w(y)$

$$d(c_1, c_2) \geq \cancel{w(f - f')} + w(g - g') - \cancel{w(f - f')} = w(g - g')$$

(Refer Slide Time: 33:59)

Reed-Muller code

$$2^{m-(r-1)} = 2^{m+1-r}$$

- Let $f, f' \in \text{RM}(r, m)$ and let $g, g' \in \text{RM}(r-1, m)$. Then vectors $c_1 = (f, f + g)$ and $c_2 = (f', f' + g')$ must be in $\text{RM}(r, m+1)$.
- If $g = g'$, then $d(c_1, c_2) = 2d(f, f') \geq 2 \cdot 2^{m-r}$.
- If $g \neq g'$, then $d(c_1, c_2) = w(f - f') + w(g - g' + f - f')$.
- Since $w(x + y) \geq w(x) - w(y)$, we have

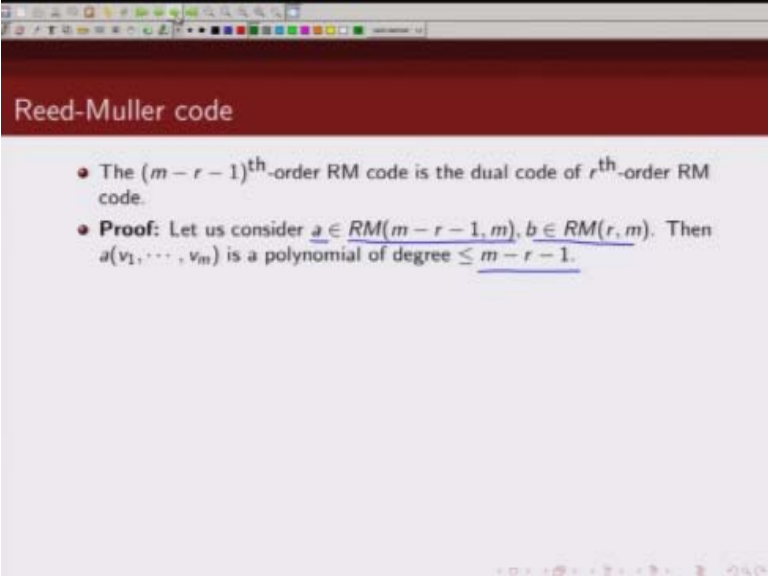
$$d(c_1, c_2) \geq w(f - f') + w(g - g') - w(f - f') = w(g - g')$$
- Since $g - g' \in \text{RM}(r-1, m)$, so that $w(g - g') \geq 2^{m-(r-1)} = \underline{2^{m-r+1}}$

(Refer Slide Time: 34:01)

Reed-Muller code

- The $(m-r-1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.

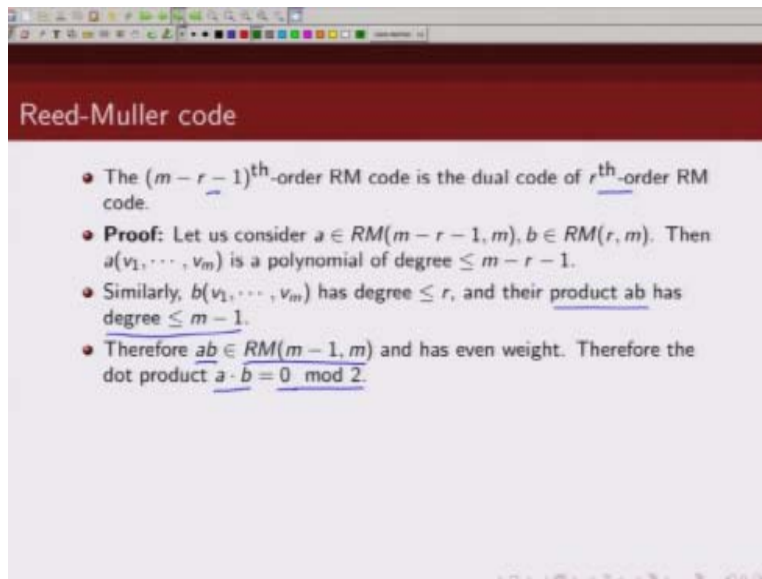
(Refer Slide Time: 34:01)



Reed-Muller code

- The $(m-r-1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m-r-1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m-r-1$.

(Refer Slide Time: 34:02)



Reed-Muller code

- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m - r - 1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m - r - 1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m - 1$.
- Therefore $ab \in RM(m - 1, m)$ and has even weight. Therefore the dot product $a \cdot b = 0 \pmod{2}$.

(Refer Slide Time: 34:02)

Reed-Muller code

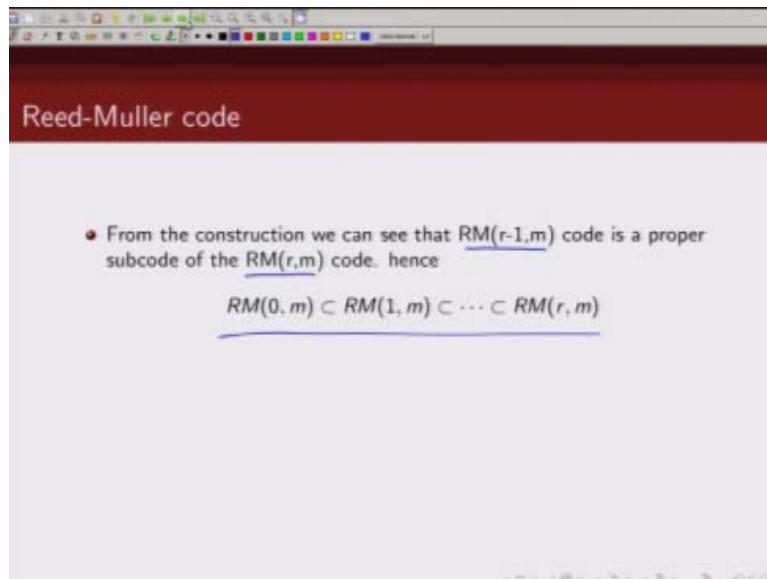
- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m - r - 1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m - r - 1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m - 1$.
- Therefore $ab \in RM(m - 1, m)$ and has even weight. Therefore the dot product $a \cdot b = 0 \pmod 2$.
- Also, $\dim RM(m - r - 1, m) + \dim RM(r, m)$

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

$$= \underline{2^m}$$

which implies that $\underline{RM(m - r - 1, m) = RM(r, m)^\perp}$.

(Refer Slide Time: 34:04)



Reed-Muller code

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

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

Larger order Reed-Muller code, so from the construction you can see that smaller order Reed-Muller code is essentially a proper sub code of a larger order Reed-Muller code. So this, this relation holds and this can be easily seen from the construction of Reed-Muller code.

(Refer Slide Time: 34:16)

Reed-Muller code

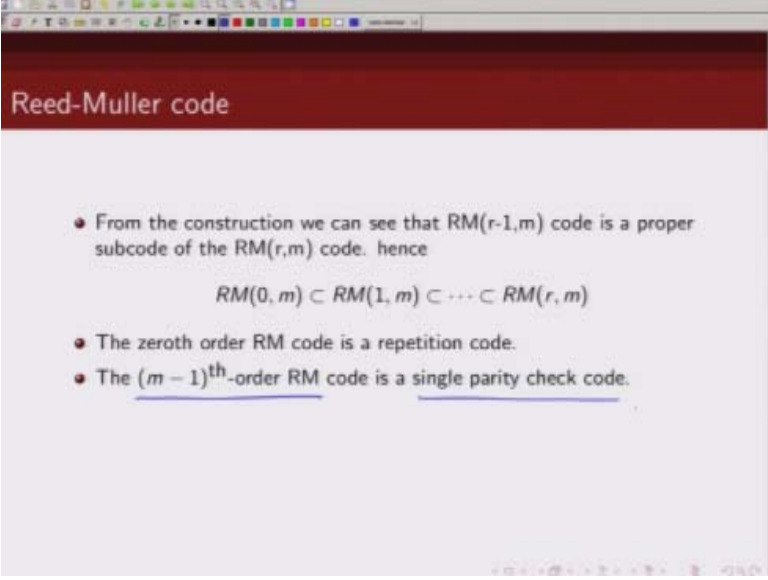
- From the construction we can see that $RM(r-1, m)$ code is a proper subcode of the $RM(r, m)$ code. hence

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

- The zeroth order RM code is a repetition code. $[v_0]$

The zeroth order Reed-Muller code is a repetition code, this we have shown earlier also. Note that for the zeroth order Reed-Muller code your G matrix will only have this v_0 which is all ones. And that is precisely the generator matrix for repetition code.

(Refer Slide Time: 34:40)



Reed-Muller code

- 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 $(m-1)^{\text{th}}$ -order RM code is a single parity check code.

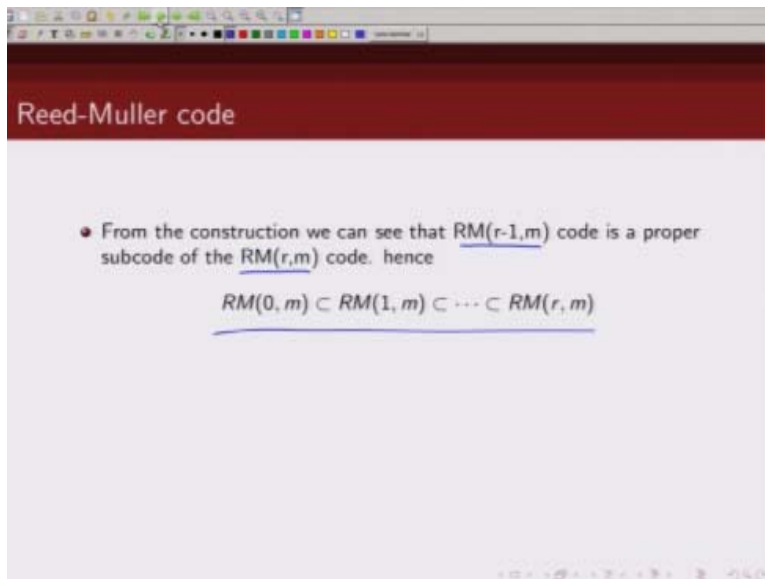
$(m-1)^{\text{th}}$ order repetition code $(m-1)^{\text{th}}$ order Reed-Muller code is actually a single parity check code again this is easy to see.

(Refer Slide Time: 34:52)

Reed-Muller code

- 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. $[v_0]$

(Refer Slide Time: 34:52)



Reed-Muller code

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

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

(Refer Slide Time: 34:53)

Reed-Muller code

- The $(m - r - 1)^{\text{th}}$ -order RM code is the dual code of r^{th} -order RM code.
- **Proof:** Let us consider $a \in RM(m - r - 1, m)$, $b \in RM(r, m)$. Then $a(v_1, \dots, v_m)$ is a polynomial of degree $\leq m - r - 1$.
- Similarly, $b(v_1, \dots, v_m)$ has degree $\leq r$, and their product ab has degree $\leq m - 1$.
- Therefore $ab \in RM(m - 1, m)$ and has even weight. Therefore the dot product $a \cdot b = 0 \pmod{2}$.
- Also, $\dim RM(m - r - 1, m) + \dim RM(r, m)$

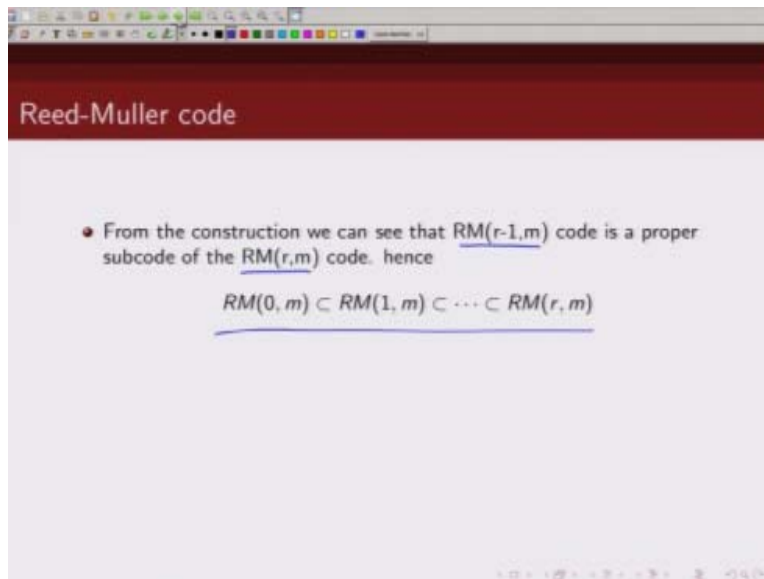
$$= 1 + \binom{m}{1} + \dots + \binom{m}{m - r - 1} + 1 + \binom{m}{1} + \dots + \binom{m}{r}$$

$$= \underline{2^m}$$

which implies that $RM(m - r - 1, m) = RM(r, m)^\perp$.

We can just use the results that we have proved. We know that $(m-r-1)^{\text{th}}$ order Reed-Muller code is dual to the r^{th} order Reed-Muller code. So if r is let us say zero then it is dual to $(m-1)^{\text{th}}$ order Reed-Muller code.

(Refer Slide Time: 35:10)



Reed-Muller code

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

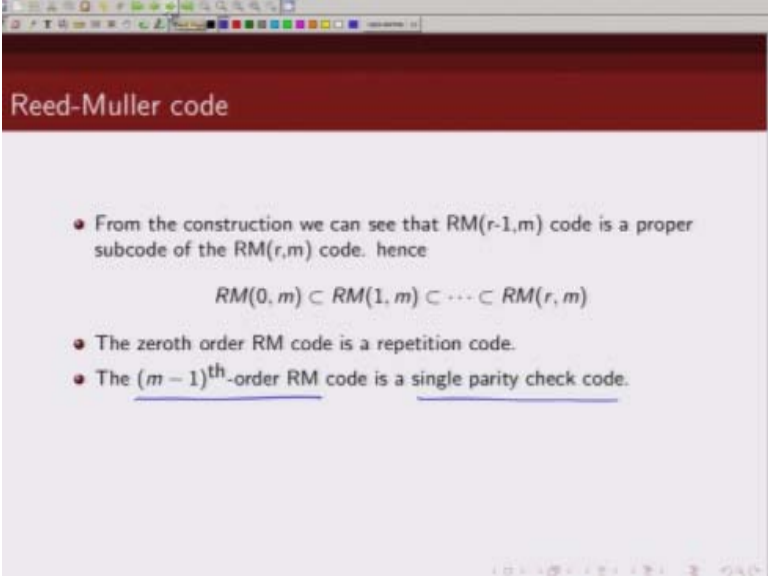
$$\underline{RM(0, m) \subset RM(1, m) \subset \dots \subset RM(r, m)}$$

(Refer Slide Time: 35:11)

Reed-Muller code

- 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. $[v_0]$

(Refer Slide Time: 35:12)

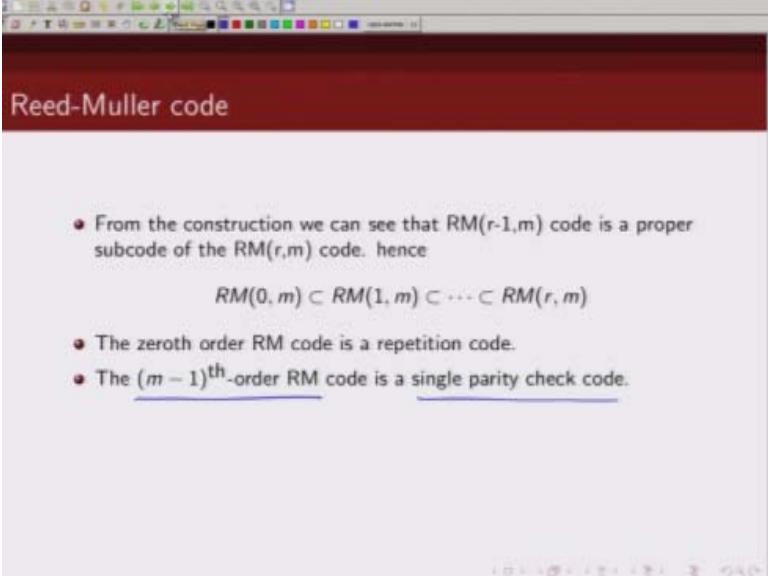


Reed-Muller code

- 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 $(m-1)^{\text{th}}$ -order RM code is a single parity check code.

So zeroth order Reed-Muller code is dual to $(m-1)^{\text{th}}$ order Reed-Muller code. And what is the dual of a repetition code, it is a single parity check code.

(Refer Slide Time: 35:27)



Reed-Muller code

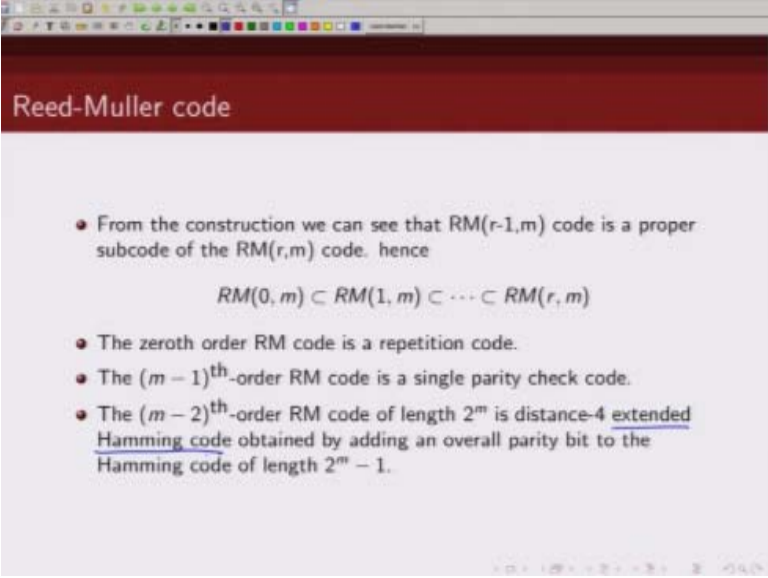
- From the construction we can see that $RM(r-1, m)$ code is a proper subcode of the $RM(r, m)$ code. hence

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

- The zeroth order RM code is a repetition code.
- The $(m-1)^{\text{th}}$ -order RM code is a single parity check code.

So $(m-1)^{\text{th}}$ order Reed-Muller code is nothing but a single parity check code. Similarly, $(m-2)$ order Reed-Muller code is our.

(Refer Slide Time: 35:39)



Reed-Muller code

- 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 $(m-1)^{\text{th}}$ -order RM code is a single parity check code.
- The $(m-2)^{\text{th}}$ -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 $2^m - 1$.

Extended hamming code which we just talked about in the last lecture. So let us discuss how we can decode Reed-Muller code, so we will illustrate the decoding of Reed-Muller code through an example, and we are going to use what we call majority logic decoding. So let us consider the Reed-Muller code.

(Refer Slide Time: 36:05)

Decoding of Reed-Muller code

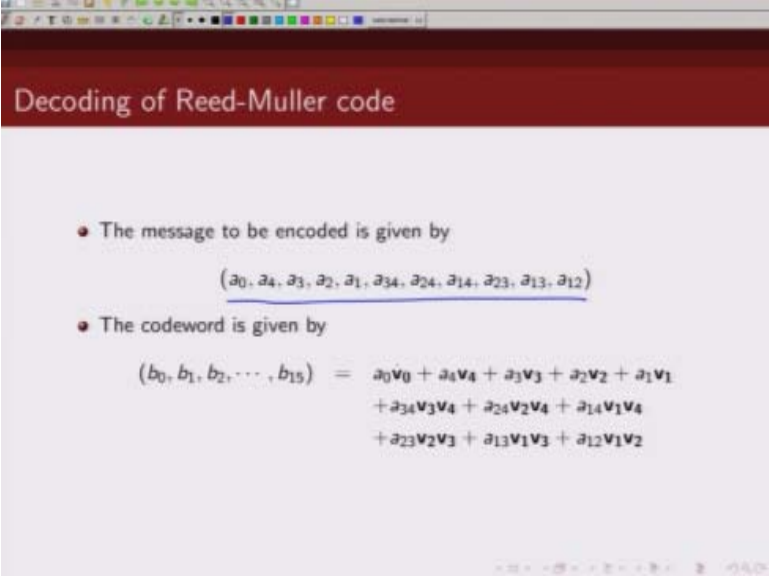
Consider a 2nd order Reed Muller code of length $n = 16$ generated by following 11 vectors $m=4, r=2$

$G =$

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

With parameter $m=4$ and $r=2$. So in other words the generator matrix will then consist of v_0 all first order v_i 's and these Boolean product of order two. We already know how to, how to get this v_1, v_2, v_3, v_m , we just talked about that earlier and we will also know how to compute the Boolean product. So this is essentially our generator matrix G of a 2,4 Reed-Muller code.

(Refer Slide Time: 36:53)



Decoding of Reed-Muller code

- The message to be encoded is given by
 $(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$
- The codeword is given by
$$(b_0, b_1, b_2, \dots, b_{15}) = a_0v_0 + a_4v_4 + a_3v_3 + a_2v_2 + a_1v_1 + a_{34}v_3v_4 + a_{24}v_2v_4 + a_{14}v_1v_4 + a_{23}v_2v_3 + a_{13}v_1v_3 + a_{12}v_1v_2$$

Now the message that we want to encode, let us call it a_0, a_4, a_3 , this is how we are denoting the message tool that we are going to encode and since the rows of our generator matrix are.

(Refer Slide Time: 37:11)

Decoding of Reed-Muller code

• Consider a 2nd order Reed Muller code of length $n = 16$ generated by following 11 vectors $m=4, r=2$

$G =$

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

Given by v_0, v_1, v_2, v_3, v_4 and this, so our code word would be.

(Refer Slide Time: 37:17)

Decoding of Reed-Muller code

- The message to be encoded is given by
 $(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$
- The codeword is given by

$$(b_0, b_1, b_2, \dots, b_{15}) = \underbrace{a_0 v_0 + a_4 v_4 + a_3 v_3 + a_2 v_2 + a_1 v_1}_{\text{Linear combination of rows of the generator matrix}}$$

Linear combination of rows of the generator matrix, so that we are writing denoting by $a_0 v_0 + a_4 v_4, a_3 v_3$ and similarly $a_{34} v_3 v_4, a_{24} v_2 v_4$, so this is how this is linear combination of these 11 rows of this generator matrix.

(Refer Slide Time: 37:39)

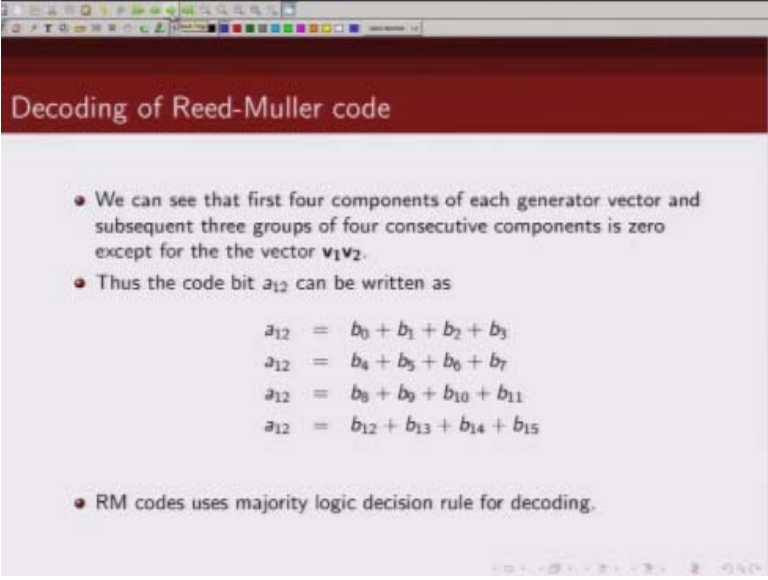
Decoding of Reed-Muller code

- The message to be encoded is given by
 $(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$
- The codeword is given by

$$(b_0, b_1, b_2, \dots, b_{15}) = \begin{array}{l} a_0v_0 + a_4v_4 + a_3v_3 + a_2v_2 + a_1v_1 \\ + a_{34}v_3v_4 + a_{24}v_2v_4 + a_{14}v_1v_4 \\ + a_{23}v_2v_3 + a_{13}v_1v_3 + a_{12}v_1v_2 \end{array}$$

That is how we will generate our code words. So this 16 length code word is basically linear combination of these rows of a generator matrix.

(Refer Slide Time: 37:52)

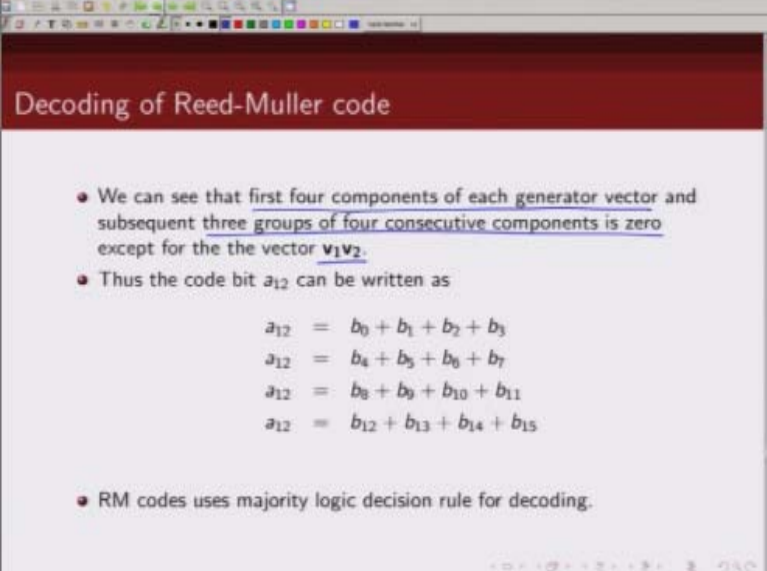


Decoding of Reed-Muller code

- We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the the vector $\mathbf{v}_1\mathbf{v}_2$.
- Thus the code bit a_{12} can be written as
$$\begin{aligned}a_{12} &= b_0 + b_1 + b_2 + b_3 \\a_{12} &= b_4 + b_5 + b_6 + b_7 \\a_{12} &= b_8 + b_9 + b_{10} + b_{11} \\a_{12} &= b_{12} + b_{13} + b_{14} + b_{15}\end{aligned}$$
- RM codes uses majority logic decision rule for decoding.

Now we will spend some time looking at the generator matrix and we will use some observations from the generator matrix to decode our code. So what are these observations so first thing we will see

(Refer Slide Time: 38:07)



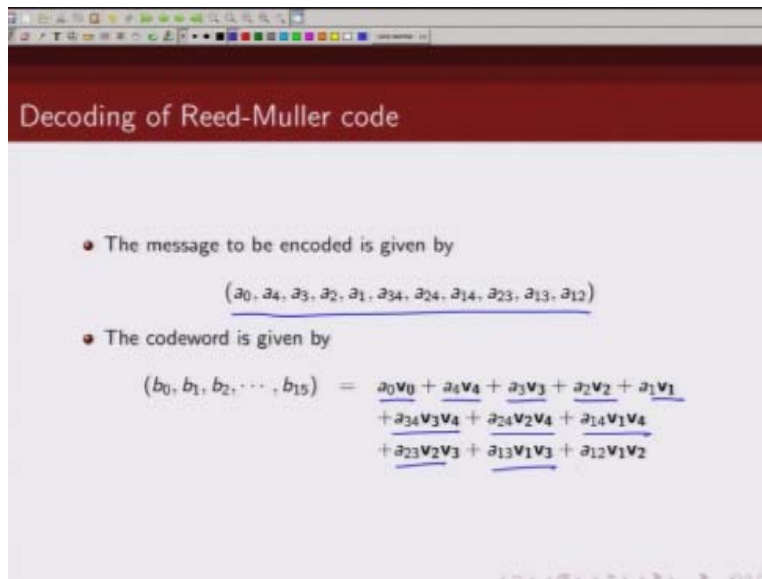
Decoding of Reed-Muller code

- 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_1v_2 .
- Thus the code bit a_{12} can be written as
$$\begin{aligned}a_{12} &= b_0 + b_1 + b_2 + b_3 \\a_{12} &= b_4 + b_5 + b_6 + b_7 \\a_{12} &= b_8 + b_9 + b_{10} + b_{11} \\a_{12} &= b_{12} + b_{13} + b_{14} + b_{15}\end{aligned}$$

- RM codes use majority logic decision rule for decoding.

If we can, if we see the first four components of each generator vector and subsequent groups of three groups of four consecutive components they are zero except for vector v_1v_2 , now what do I mean by that?

(Refer Slide Time: 38:26)



Decoding of Reed-Muller code

- The message to be encoded is given by
 $(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$
- The codeword is given by
$$(b_0, b_1, b_2, \dots, b_{15}) = \underbrace{a_0 v_0} + \underbrace{a_4 v_4} + \underbrace{a_3 v_3} + \underbrace{a_2 v_2} + \underbrace{a_1 v_1} + \underbrace{a_{34} v_3 v_4} + \underbrace{a_{24} v_2 v_4} + \underbrace{a_{14} v_1 v_4} + \underbrace{a_{23} v_2 v_3} + \underbrace{a_{13} v_1 v_3} + \underbrace{a_{12} v_1 v_2}$$

(Refer Slide Time: 38:27)

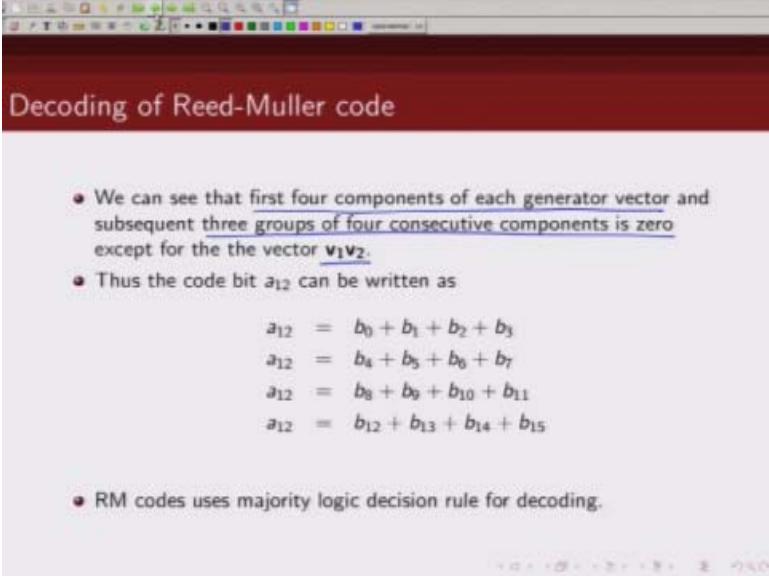
Decoding of Reed-Muller code

• 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
v_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
v_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1
v_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1
v_4	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$v_1 v_2$	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0
$v_1 v_3$	0	0	0	0	1	0	1	0	0	0	0	1	0	1	0
$v_1 v_4$	0	0	0	0	0	0	0	1	0	1	0	1	0	1	0
$v_2 v_3$	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
$v_2 v_4$	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0
$v_3 v_4$	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

So let us look at this group of four this is group of four, this is group of four, so what I am saying is if you look out this group of four and if you add them up. Let us look at this first group of four this will be zero sum will be zero, zero, zero, zero. This is one, this is zero, zero, zero, zero, zero you take any such four, this is zero, zero, zero this one is zero, this one is zero, this is not zero. Again this row, this one is zero, zero, zero, zero, so you take any such groups of four. This one is zero, zero, zero, zero, zero this is not zero and these are all zeros. Similarly this is not zero these are all if you add up these they are all zero, one plus one, one plus, one plus one these are all zero same here one plus one, zero one plus one zero so if you look at these bits four bits at a time you will notice except for this one $v_1 v_2$ all others are zero.

(Refer Slide Time: 39:54)



Decoding of Reed-Muller code

- We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the the vector v_1v_2 .
- Thus the code bit a_{12} can be written as
$$\begin{aligned}a_{12} &= b_0 + b_1 + b_2 + b_3 \\a_{12} &= b_4 + b_5 + b_6 + b_7 \\a_{12} &= b_8 + b_9 + b_{10} + b_{11} \\a_{12} &= b_{12} + b_{13} + b_{14} + b_{15}\end{aligned}$$
- RM codes uses majority logic decision rule for decoding.

Now how can we make use of this fact?

(Refer Slide Time: 39:56)

Decoding of Reed-Muller code

- The message to be encoded is given by
 $(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$
- The codeword is given by

$$(b_0, b_1, b_2, \dots, b_{15}) = \underbrace{a_0 v_0 + a_4 v_4 + a_3 v_3 + a_2 v_2 + a_1 v_1}_{\text{first four elements}} + a_{34} v_3 v_4 + a_{24} v_2 v_4 + a_{14} v_1 v_4 + a_{23} v_2 v_3 + a_{13} v_1 v_3 + \boxed{a_{12} v_1 v_2}$$

$v_1 v_2$ so what we will do is if we add up those first four elements.

(Refer Slide Time: 40:13)

Decoding of Reed-Muller code

- The message to be encoded is given by

$$(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$$
- The codeword is given by

$$(b_0, b_1, b_2, \dots, b_{15}) = \begin{aligned} & a_0 \mathbf{v}_0 + a_4 \mathbf{v}_4 + a_3 \mathbf{v}_3 + a_2 \mathbf{v}_2 + a_1 \mathbf{v}_1 \\ & + a_{34} \mathbf{v}_3 \mathbf{v}_4 + a_{24} \mathbf{v}_2 \mathbf{v}_4 + a_{14} \mathbf{v}_1 \mathbf{v}_4 \\ & + a_{23} \mathbf{v}_2 \mathbf{v}_3 + a_{13} \mathbf{v}_1 \mathbf{v}_3 + a_{12} \mathbf{v}_1 \mathbf{v}_2 \end{aligned}$$

The contribution from all others will be zero except, because $\mathbf{v}_1 \mathbf{v}_2$ is non zero so we will get contribution from what a_{12} is.

(Refer Slide Time: 40:24)

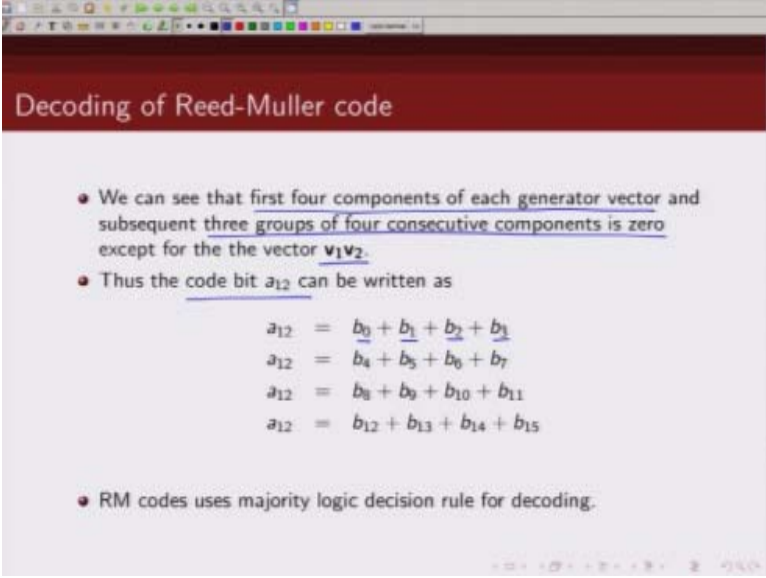
Decoding of Reed-Muller code

- The message to be encoded is given by

$$(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$$
- The codeword is given by

$$(b_0, b_1, b_2, \dots, b_{15}) = \begin{aligned} & \underline{a_0 \mathbf{v}_0} + \underline{a_4 \mathbf{v}_4} + \underline{a_3 \mathbf{v}_3} + \underline{a_2 \mathbf{v}_2} + \underline{a_1 \mathbf{v}_1} \\ & + \underline{a_{34} \mathbf{v}_3 \mathbf{v}_4} + \underline{a_{24} \mathbf{v}_2 \mathbf{v}_4} + \underline{a_{14} \mathbf{v}_1 \mathbf{v}_4} \\ & + \underline{a_{23} \mathbf{v}_2 \mathbf{v}_3} + \underline{a_{13} \mathbf{v}_1 \mathbf{v}_3} + \underline{a_{12} \mathbf{v}_1 \mathbf{v}_2} \end{aligned}$$

(Refer Slide Time: 40:24)



Decoding of Reed-Muller code

- We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the the vector $\mathbf{v}_1\mathbf{v}_2$.
- Thus the code bit a_{12} can be written as
$$\begin{aligned}a_{12} &= b_0 + b_1 + b_2 + b_3 \\a_{12} &= b_4 + b_5 + b_6 + b_7 \\a_{12} &= b_8 + b_9 + b_{10} + b_{11} \\a_{12} &= b_{12} + b_{13} + b_{14} + b_{15}\end{aligned}$$
- RM codes uses majority logic decision rule for decoding.

So in other words these code word bit then can be written as so if I am calling this bit at zeroth location as zero bit at first location as b_1 , second location b_2 , and b_3 then by adding the first four bits I can get information about what a_{12} was.

(Refer Slide Time: 40:52)

Decoding of Reed-Muller code

- The message to be encoded is given by

$$(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$$
- The codeword is given by

$$(b_0, b_1, b_2, \dots, b_{15}) = \begin{aligned} & \underline{a_0 v_0} + \underline{a_4 v_4} + \underline{a_3 v_3} + \underline{a_2 v_2} + \underline{a_1 v_1} \\ & + \underline{a_{34} v_3 v_4} + \underline{a_{24} v_2 v_4} + \underline{a_{14} v_1 v_4} \\ & + \underline{a_{23} v_2 v_3} + \underline{a_{13} v_1 v_3} + \underline{a_{12} v_1 v_2} \end{aligned}$$

And this can continue for next set of bits as well.

(Refer Slide Time: 40:55)

Decoding of Reed-Muller code

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

$$G = \begin{bmatrix} \mathbf{v}_0 & 1111111111111111 \\ \mathbf{v}_1 & 0101010101010101 \\ \mathbf{v}_2 & 0011001100110011 \\ \mathbf{v}_3 & 0000111100001111 \\ \mathbf{v}_4 & 0000000011111111 \\ \mathbf{v}_1\mathbf{v}_2 & 0001000100010001 \\ \mathbf{v}_1\mathbf{v}_3 & 0000010100000101 \\ \mathbf{v}_1\mathbf{v}_4 & 0000000001010101 \\ \mathbf{v}_2\mathbf{v}_3 & 0000001100000011 \\ \mathbf{v}_2\mathbf{v}_4 & 0000000000110011 \\ \mathbf{v}_3\mathbf{v}_4 & 0000000000001111 \end{bmatrix}$$

(Refer Slide Time: 40:57)

Decoding of Reed-Muller code

- The message to be encoded is given by
 $(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$
- The codeword is given by
$$(b_0, b_1, b_2, \dots, b_{15}) = \begin{aligned} & \underline{a_0 \mathbf{v}_0} + \underline{a_4 \mathbf{v}_4} + \underline{a_3 \mathbf{v}_3} + \underline{a_2 \mathbf{v}_2} + \underline{a_1 \mathbf{v}_1} \\ & + \underline{a_{34} \mathbf{v}_3 \mathbf{v}_4} + \underline{a_{24} \mathbf{v}_2 \mathbf{v}_4} + \underline{a_{14} \mathbf{v}_1 \mathbf{v}_4} \\ & + \underline{a_{23} \mathbf{v}_2 \mathbf{v}_3} + \underline{a_{13} \mathbf{v}_1 \mathbf{v}_3} + \underline{a_{12} \mathbf{v}_1 \mathbf{v}_2} \end{aligned}$$

(Refer Slide Time: 40:57)

Decoding of Reed-Muller code

- We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the the vector $\underline{v_1 v_2}$.
- Thus the code bit $\underline{a_{12}}$ can be written as
$$\begin{aligned}a_{12} &= \underline{b_0} + \underline{b_1} + \underline{b_2} + \underline{b_3} \\a_{12} &= \underline{b_4} + \underline{b_5} + \underline{b_6} + \underline{b_7} \\a_{12} &= \underline{b_8} + \underline{b_9} + \underline{b_{10}} + \underline{b_{11}} \\a_{12} &= \underline{b_{12}} + \underline{b_{13}} + \underline{b_{14}} + \underline{b_{15}}\end{aligned}$$

- RM codes uses majority logic decision rule for decoding.

(Refer Slide Time: 40:58)

Decoding of Reed-Muller code

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

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}
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	0	0	0	0
$v_1 v_2$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
$v_1 v_3$	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	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
$v_2 v_4$	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
$v_3 v_4$	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

So this is let say $b_0b_1b_2b_3$ this is $b_4b_5b_6b_7$ this is $b_8b_9b_{10}b_{11}$ this is $b_{12}b_{13}b_{14}b_{15}$. So if I add this $b_0b_1b_2b_3$ or $b_4b_5b_6b_7$, $b_8b_9b_{10}b_{11}$ or $b_{12}b_{13}b_{14}b_{15}$ what I am getting is contributions from all other rows are nullified only I receive the contribution effect of this v_1v_2 .

(Refer Slide Time: 41:45)

Decoding of Reed-Muller code

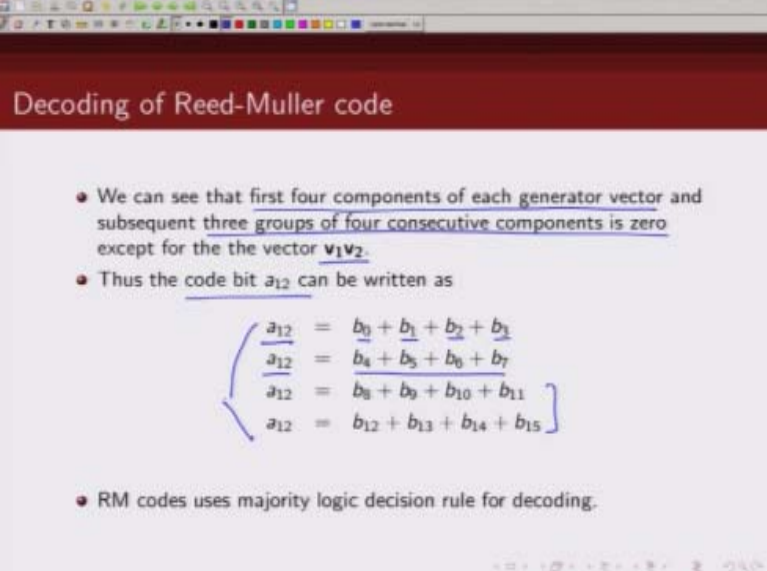
- We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the the vector v_1v_2 .
- Thus the code bit a_{12} can be written as

$$\left. \begin{aligned} a_{12} &= b_0 + b_1 + b_2 + b_3 \\ a_{12} &= b_4 + b_5 + b_6 + b_7 \\ a_{12} &= b_8 + b_9 + b_{10} + b_{11} \\ a_{12} &= b_{12} + b_{13} + b_{14} + b_{15} \end{aligned} \right\}$$

- RM codes uses majority logic decision rule for decoding.

And the bit a_{12} can then be found by adding these four columns together. So I can get the information about a_{12} by looking at these first four columns or first four bits of these code word. Similarly in next four bits of the code word if I add them up I can get another independent information about a_{12} . And same thing I can get from the next set of four coded bits. So what you can see is I am getting four independent views about what a_{12} is. Now the decoder can take a majority logic decode. If there is no error of course all of them will tell me about that a_{12} is the same bit whether zero or one. But if there is, is there is a single error what you will notice is you know in some other bits. Let us say there is an error in some.

(Refer Slide Time: 42:47)

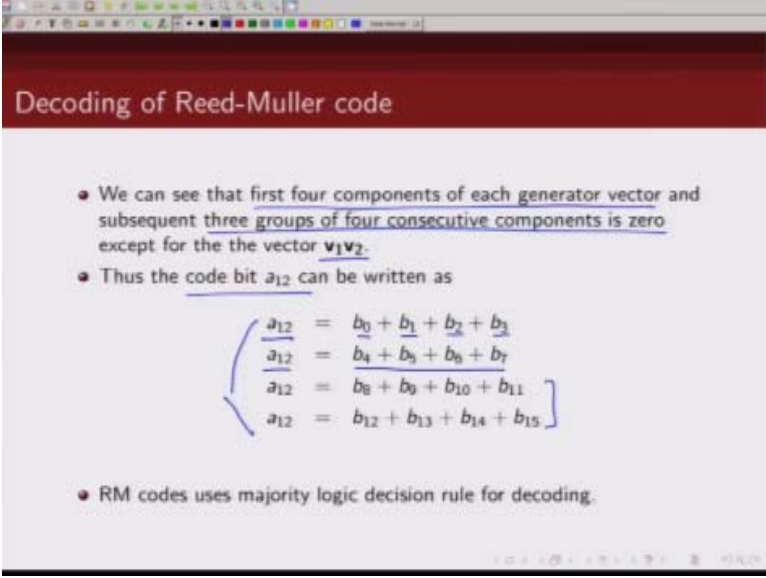


Decoding of Reed-Muller code

- We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the the vector v_1v_2 .
- Thus the code bit a_{12} can be written as
$$\left\{ \begin{array}{l} a_{12} = b_0 + b_1 + b_2 + b_3 \\ a_{12} = b_4 + b_5 + b_6 + b_7 \\ a_{12} = b_8 + b_9 + b_{10} + b_{11} \\ a_{12} = b_{12} + b_{13} + b_{14} + b_{15} \end{array} \right\}$$
- RM codes uses majority logic decision rule for decoding.

Bit location b_1 then a_{12} here would be different from what a_{12} I am getting from other three equations and then I will use majority logic decoding. What is majority logic decoding so I will take the majority decision if, if three of them are saying a_{12} is zero then I will go for zero otherwise I will go for 1 okay.

(Refer Slide Time: 43:10)



Decoding of Reed-Muller code

- We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the the vector $\mathbf{v_1v_2}$.
- Thus the code bit a_{12} can be written as
$$\left. \begin{aligned} a_{12} &= b_0 + b_1 + b_2 + b_3 \\ a_{12} &= b_4 + b_5 + b_6 + b_7 \\ a_{12} &= b_8 + b_9 + b_{10} + b_{11} \\ a_{12} &= b_{12} + b_{13} + b_{14} + b_{15} \end{aligned} \right\}$$
- RM codes uses majority logic decision rule for decoding.

So this is how I can decode bit a_{12} .

(Refer Slide Time: 43:13)

Decoding of Reed-Muller code

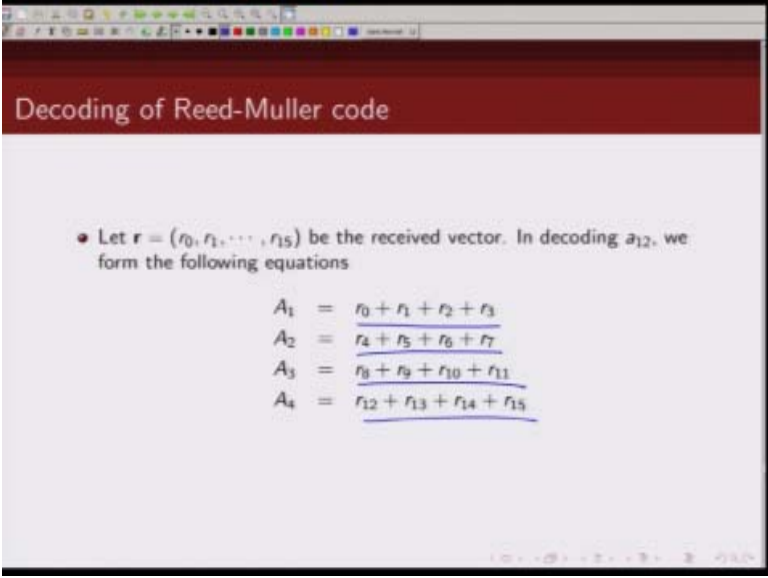
Consider a 2nd order Reed Muller code of length $n = 16$ generated by following 11 vectors

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}
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	1	1	1	1	1	1	1	1	1
$v_1 v_2$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
$v_1 v_3$	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	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
$v_2 v_4$	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
$v_3 v_4$	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Handwritten notes on the slide include blue arrows pointing to specific rows and blue annotations indicating that certain rows (like $v_1 v_3$ and $v_1 v_4$) are equal to zero.

So and this.

(Refer Slide Time: 43:16)



Decoding of Reed-Muller code

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

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

Will be repeated for decoding other bits as well so let us say my receive bit is r_0, r_1, r_2, r_{15} corresponding to the transmitted bit b_0, b_1, b_2, b_{15} then I can decode a_{12} , how, I will just add these first 4 bits, then add the next 4 bits, next 4 bits, next 4 bits, so I am getting 4 independent views about what a_{12} is, and then I will take a majority decision, majority of them are saying 0 I will go for 0 otherwise I will go for 1.

(Refer Slide Time: 43:52)

Decoding of Reed-Muller code

- Similarly we can decode, a_{13} , a_{23} , a_{14} , a_{24} , a_{34} . For example, for a_{13} we have
$$\begin{aligned}A_1 &= r_0 + r_1 + r_4 + r_5 \\A_2 &= r_2 + r_3 + r_6 + r_7 \\A_3 &= r_8 + r_9 + r_{12} + r_{13} \\A_4 &= r_{10} + r_{11} + r_{14} + r_{15}\end{aligned}$$
- For a_{23} we have
$$\begin{aligned}A_1 &= r_0 + r_2 + r_4 + r_6 \\A_2 &= r_1 + r_3 + r_5 + r_7 \\A_3 &= r_8 + r_{10} + r_{12} + r_{14} \\A_4 &= r_9 + r_{11} + r_{13} + r_{15}\end{aligned}$$

Now the same thing exactly same way I can decode other bits.

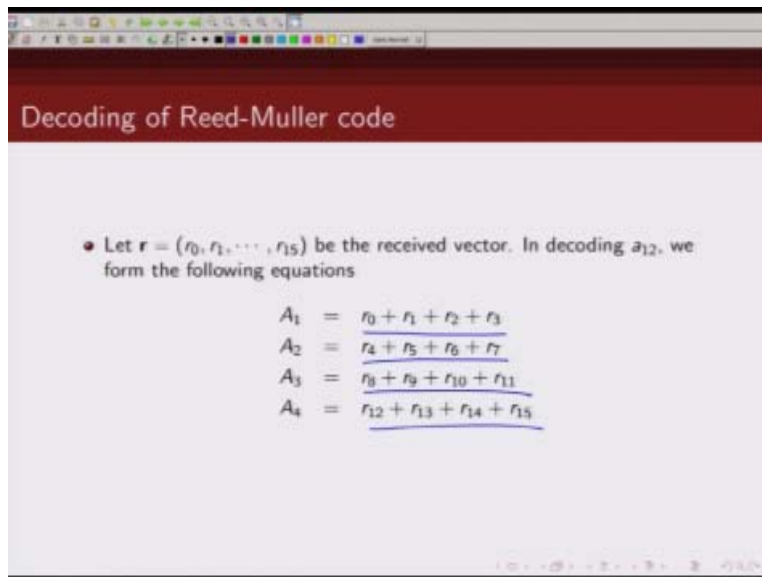
(Refer Slide Time: 43:59)

Decoding of Reed-Muller code

- Similarly we can decode, a_{13} , a_{23} , a_{14} , a_{24} , a_{34} . For example, for a_{13} we have
$$\begin{aligned}A_1 &= r_0 + r_1 + r_4 + r_5 \\A_2 &= r_2 + r_3 + r_6 + r_7 \\A_3 &= r_8 + r_9 + r_{12} + r_{13} \\A_4 &= r_{10} + r_{11} + r_{14} + r_{15}\end{aligned}$$
- For a_{23} we have
$$\begin{aligned}A_1 &= r_0 + r_2 + r_4 + r_6 \\A_2 &= r_1 + r_3 + r_5 + r_7 \\A_3 &= r_8 + r_{10} + r_{12} + r_{14} \\A_4 &= r_9 + r_{11} + r_{13} + r_{15}\end{aligned}$$

So let us look at a_{23} if you look at a_{23} .

(Refer Slide Time: 44:04)



Decoding of Reed-Muller code

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

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

(Refer Slide Time: 44:05)

Decoding of Reed-Muller code

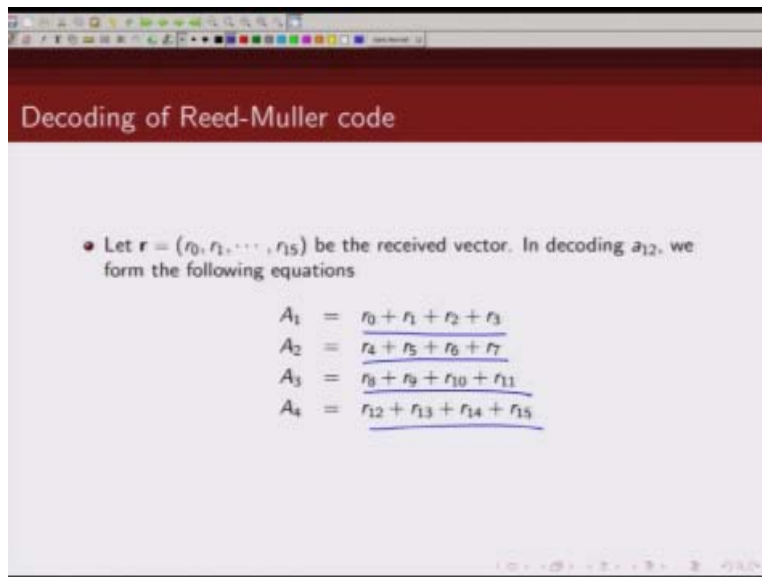
Consider a 2nd order Reed Muller code of length $n = 16$ generated by following 11 vectors

	b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}
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	1	1	1	1	1	1	1	1	1
$v_1 v_2$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
$v_1 v_3$	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	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
$v_2 v_4$	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
$v_3 v_4$	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Handwritten annotations on the slide include a red double-headed arrow on the left side of the vectors, a blue arrow pointing to the $v_1 v_2$ row, and red arrows pointing to the $v_2 v_3$ and $v_2 v_4$ rows. Blue and red vertical lines are drawn through the columns of the matrix to illustrate the evaluation process.

Let us look at this row evaluate with a different pen, let us look this row, this row, this row and this row so if I add bits in this row this will be 0, this will give me 0, this will give me a 1, this will give me 0, this will give me 0, this will give me 0, so you can see all rows will give me 0 except this particular row and same thing I can repeat for

(Refer Slide Time: 44:48)



Decoding of Reed-Muller code

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

$$\begin{aligned} A_1 &= \frac{r_0 + r_1 + r_2 + r_3}{r_4 + r_5 + r_6 + r_7} \\ A_2 &= \frac{r_8 + r_9 + r_{10} + r_{11}}{r_{12} + r_{13} + r_{14} + r_{15}} \end{aligned}$$

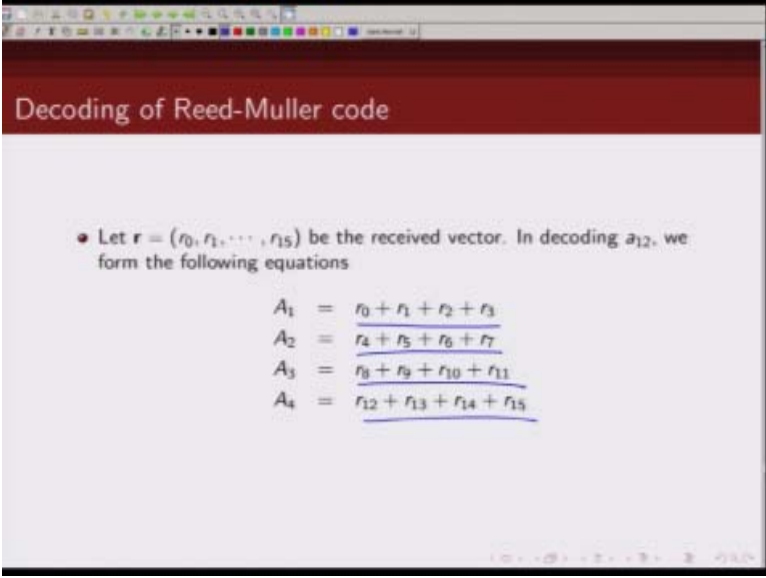
(Refer Slide Time: 44:49)

Decoding of Reed-Muller code

- Similarly we can decode, a_{13} , a_{23} , a_{14} , a_{24} , a_{34} . For example, for a_{13} we have
$$\begin{aligned}A_1 &= r_0 + r_1 + r_4 + r_5 \\A_2 &= r_2 + r_3 + r_6 + r_7 \\A_3 &= r_8 + r_9 + r_{12} + r_{13} \\A_4 &= r_{10} + r_{11} + r_{14} + r_{15}\end{aligned}$$
- For a_{23} we have
$$\begin{aligned}A_1 &= r_0 + r_2 + r_4 + r_6 \\A_2 &= r_1 + r_3 + r_5 + r_7 \\A_3 &= r_8 + r_{10} + r_{12} + r_{14} \\A_4 &= r_9 + r_{11} + r_{13} + r_{15}\end{aligned}$$

If I look at a 2nd row, 4th row 6th row and 8th I will get the same information.

(Refer Slide Time: 45:00)



Decoding of Reed-Muller code

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

$$\begin{aligned} A_1 &= \frac{r_0 + r_1 + r_2 + r_3}{r_4 + r_5 + r_6 + r_7} \\ A_2 &= \frac{r_8 + r_9 + r_{10} + r_{11}}{r_{12} + r_{13} + r_{14} + r_{15}} \end{aligned}$$

So if I look at.

(Refer Slide Time: 45:01)

Decoding of Reed-Muller code

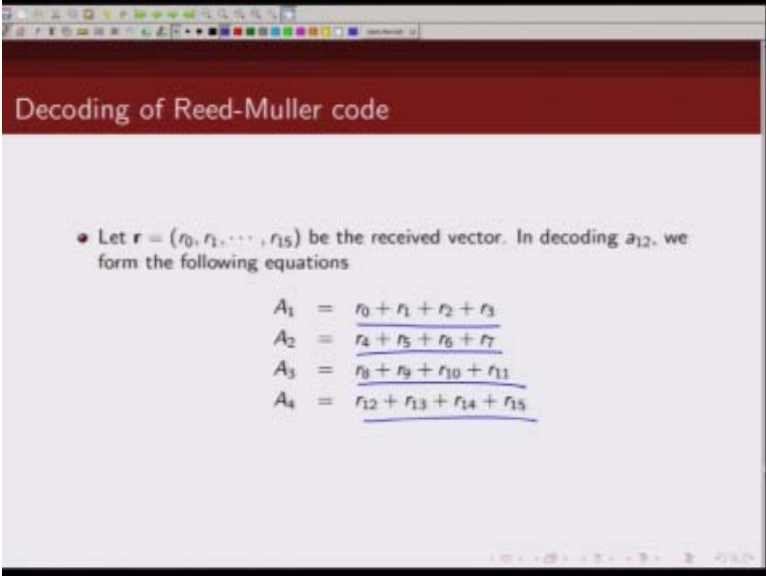
• Consider a 2nd order Reed Muller code of length $n = 16$ generated by following 11 vectors

	$b_0 \ b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6 \ b_7 \ b_8 \ b_9 \ b_{10} \ b_{11} \ b_{12} \ b_{13} \ b_{14} \ b_{15}$
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 1 0 0 0 1 0 0 0 1 0 0 0 1
$v_1 v_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 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
$v_2 v_4$	0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$v_3 v_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

Handwritten annotations on the slide include a red double-headed arrow on the left side of the vectors, a blue arrow pointing to the $v_1 v_2$ row, a green arrow pointing to the $v_2 v_3$ row, and a red arrow pointing to the $v_3 v_4$ row. There are also blue and red arrows pointing to specific columns in the vectors.

Now let us say I look at this row if I look at this row, this row, this row and this row so this will give me 0, this, this, this will give me 0 this will give me 0, now here this is a 1, this is a 0 this a 0 and this is 0, so this will give me 1, and all other rows will give me 0.

(Refer Slide Time: 45:31)



Decoding of Reed-Muller code

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

$$\begin{aligned} A_1 &= \frac{r_0 + r_1 + r_2 + r_3}{4} \\ A_2 &= \frac{r_4 + r_5 + r_6 + r_7}{4} \\ A_3 &= \frac{r_8 + r_9 + r_{10} + r_{11}}{4} \\ A_4 &= \frac{r_{12} + r_{13} + r_{14} + r_{15}}{4} \end{aligned}$$

So if I add up.

(Refer Slide Time: 45:32)

Decoding of Reed-Muller code

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

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

- For a_{23} we have

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

These bits 4 bits at a time in similar fashion I can get independent.

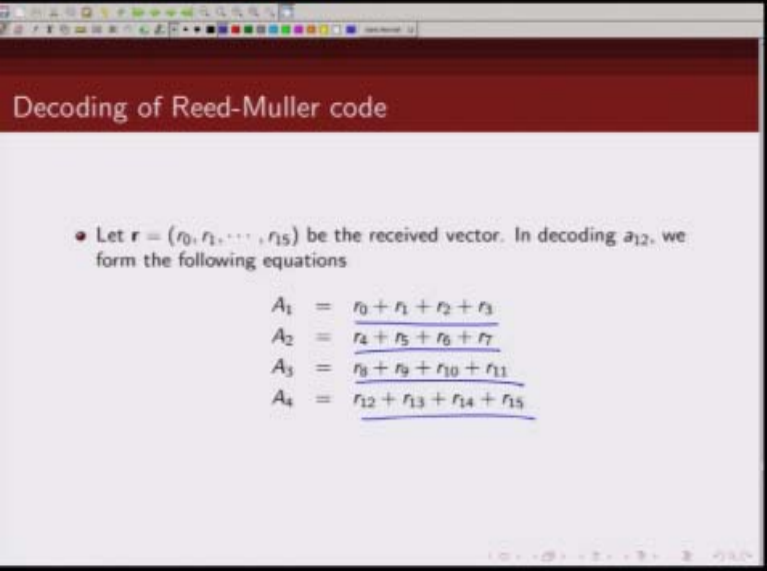
(Refer Slide Time: 45:42)

Decoding of Reed-Muller code

- Similarly we can decode, a_{13} , a_{23} , a_{14} , a_{24} , a_{34} . For example, for a_{13} we have
$$\begin{aligned}A_1 &= r_0 + r_1 + r_4 + r_5 \\A_2 &= r_2 + r_3 + r_6 + r_7 \\A_3 &= r_8 + r_9 + r_{12} + r_{13} \\A_4 &= r_{10} + r_{11} + r_{14} + r_{15}\end{aligned}$$
- For a_{23} we have
$$\begin{aligned}A_1 &= \underline{r_0 + r_2 + r_4 + r_6} \\A_2 &= \underline{r_1 + r_3 + r_5 + r_7} \\A_3 &= \underline{r_8 + r_{10} + r_{12} + r_{14}} \\A_4 &= \underline{r_9 + r_{11} + r_{13} + r_{15}}\end{aligned}$$

Information about a_{23} , so again the point we noted.

(Refer Slide Time: 45:49)



Decoding of Reed-Muller code

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

$$\begin{aligned} A_1 &= \frac{r_0 + r_1 + r_2 + r_3}{r_4 + r_5 + r_6 + r_7} \\ A_2 &= \frac{r_8 + r_9 + r_{10} + r_{11}}{r_{12} + r_{13} + r_{14} + r_{15}} \end{aligned}$$

Here is.

(Refer Slide Time: 45:50)

Decoding of Reed-Muller code

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

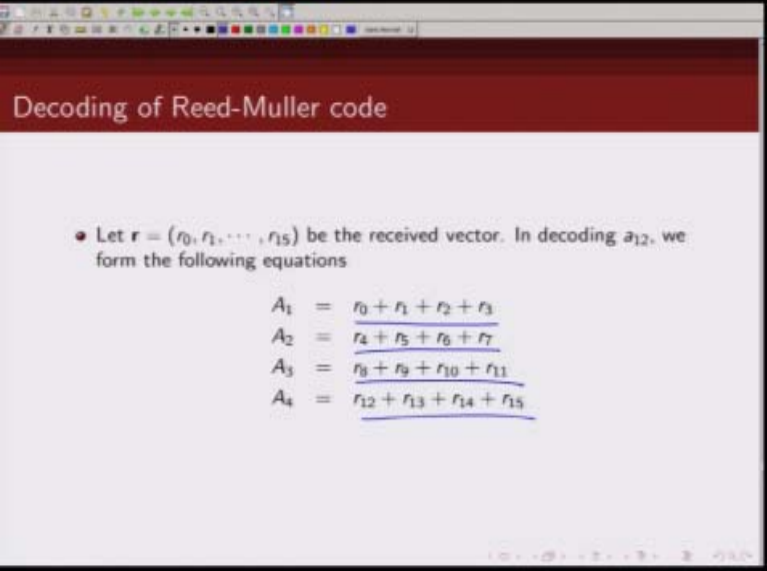
		b_0	b_1	b_2	b_3	b_4	b_5	b_6	b_7	b_8	b_9	b_{10}	b_{11}	b_{12}	b_{13}	b_{14}	b_{15}
v_0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
v_1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0
v_2	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
v_3	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0
v_4	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0
$v_1 v_2$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0
$v_1 v_3$	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	0
$v_1 v_4$	0	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	0	1	1
$v_2 v_4$	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0
$v_3 v_4$	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Handwritten annotations on the slide include:

- A red double-headed arrow on the left side of the vectors, spanning from v_0 to v_4 .
- A green '0' next to the first set of vectors (v_0 to v_4).
- A blue arrow pointing to the vector $v_1 v_2$.
- A red arrow pointing to the vector $v_2 v_3$.
- A green '1' next to the second set of vectors ($v_1 v_2$ to $v_3 v_4$).
- Red arrows pointing to the vectors $v_2 v_4$ and $v_3 v_4$.
- Red arrows pointing to the last four bits (1111) of the vector $v_3 v_4$.

What you need to do is you would look at this and find out.

(Refer Slide Time: 45:57)



Decoding of Reed-Muller code

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

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

Basically like a combination of these receive bits which will give information about one particular transmitted bit and not others and once you do that.

(Refer Slide Time: 46:10)

Decoding of Reed-Muller code

- Similarly we can decode, a_{13} , a_{23} , a_{14} , a_{24} , a_{34} . For example, for a_{13} we have
$$\begin{aligned}A_1 &= r_0 + r_1 + r_4 + r_5 \\A_2 &= r_2 + r_3 + r_6 + r_7 \\A_3 &= r_8 + r_9 + r_{12} + r_{13} \\A_4 &= r_{10} + r_{11} + r_{14} + r_{15}\end{aligned}$$
- For a_{23} we have
$$\begin{aligned}A_1 &= \underline{r_0 + r_2 + r_4 + r_6} \\A_2 &= \underline{r_1 + r_3 + r_5 + r_7} \\A_3 &= \underline{r_8 + r_{10} + r_{12} + r_{14}} \\A_4 &= \underline{r_9 + r_{11} + r_{13} + r_{15}}\end{aligned}$$

(Refer Slide Time: 46:12)

Decoding of Reed-Muller code

- For a_{14}
$$\begin{aligned} A_1 &= r_0 + r_1 + r_8 + r_9 \\ A_2 &= r_2 + r_3 + r_{10} + r_{11} \\ A_3 &= r_4 + r_5 + r_{12} + r_{13} \\ A_4 &= r_6 + r_7 + r_{14} + r_{15} \end{aligned}$$
- For a_{24} we have
$$\begin{aligned} A_1 &= r_0 + r_1 + r_4 + r_5 \\ A_2 &= r_2 + r_3 + r_6 + r_7 \\ A_3 &= r_8 + r_9 + r_{12} + r_{13} \\ A_4 &= r_{10} + r_{11} + r_{14} + r_{15} \end{aligned}$$

You can similarly do for other bits, I just listed here you can verify yourself that if you add these bit location you will get independent formation about a_{14} , similarly for a_{24} .

(Refer Slide Time: 46:28)

Decoding of Reed-Muller code

- For a_{34} we have

$$\begin{aligned} A_1 &= r_0 + r_4 + r_8 + r_{12} \\ A_2 &= r_1 + r_5 + r_9 + r_{13} \\ A_3 &= r_2 + r_6 + r_{10} + r_{14} \\ A_4 &= r_3 + r_7 + r_{11} + r_{15} \end{aligned}$$
- After decoding $a_{12}, a_{13}, a_{23}, a_{14}, a_{24}, a_{34}$, we form a modified received vector as

$$\begin{aligned} \mathbf{r}^{(1)} &= (r_0^{(1)}, r_1^{(1)}, \dots, r_{15}^{(1)}) \\ &= \mathbf{r} - a_{34}\mathbf{v}_3\mathbf{v}_4 - a_{24}\mathbf{v}_2\mathbf{v}_4 - a_{14}\mathbf{v}_1\mathbf{v}_4 - a_{23}\mathbf{v}_2\mathbf{v}_3 - a_{13}\mathbf{v}_1\mathbf{v}_3 - a_{12}\mathbf{v}_1\mathbf{v}_2 \end{aligned}$$

And a_{34} , now once you have decoded a_{12} , a_{23} , or once you have decoded all of these, again remember the way we are decoding is so we are getting 4 independent views about the same bit majority of them are must saying it is 0 we go for that or else the majority of them are saying that they are one, we will go for that.

(Refer Slide Time: 46:53)

Decoding of Reed-Muller code

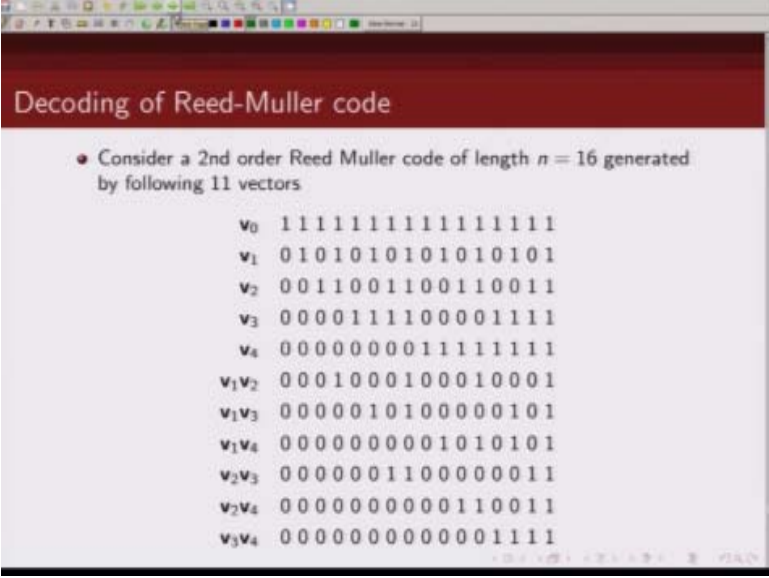
- For a_{34} we have

$$\begin{aligned} A_1 &= r_0 + r_4 + r_8 + r_{12} \\ A_2 &= r_1 + r_5 + r_9 + r_{13} \\ A_3 &= r_2 + r_6 + r_{10} + r_{14} \\ A_4 &= r_3 + r_7 + r_{11} + r_{15} \end{aligned}$$
- After decoding $a_{12}, a_{13}, a_{23}, a_{14}, a_{24}, a_{34}$, we form a modified received vector as

$$\begin{aligned} \underline{r^{(1)}} &= (r_0^{(1)}, r_1^{(1)}, \dots, r_{15}^{(1)}) \\ &= \underline{r} - a_{34}v_3v_4 - a_{24}v_2v_4 - a_{14}v_1v_4 - a_{23}v_2v_3 - a_{13}v_1v_3 - a_{12}v_1v_2 \end{aligned}$$

So once we have decoded these sequences let us just subtract the contribution of these bits from the received signal, so then the new received sequence that we are calling r_1 is the actual rate sequence – the contribution from these Boolean product terms subtracted, now once we do this then what we are left with is.

(Refer Slide Time: 47:26)



Decoding of Reed-Muller code

- 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_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 1 0 0 0 1 0 0 0 1 0 0 0 1
$v_1 v_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 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
$v_2 v_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$v_3 v_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

(Refer Slide Time: 47:27)

Decoding of Reed-Muller code

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

$$(b_0^{(1)}, b_1^{(1)}, \dots, b_{15}^{(1)}) = \underbrace{a_0 \mathbf{v}_0 + a_4 \mathbf{v}_4 + a_3 \mathbf{v}_3 + a_2 \mathbf{v}_2 + a_1 \mathbf{v}_1}_{\text{codeword}}$$
- We can see that sum of every two components of $\mathbf{v}_0, \mathbf{v}_4, \mathbf{v}_3, \mathbf{v}_2$ starting from first is zero, whereas for \mathbf{v}_1 it is 1.
- Therefore we can form eight independent equations for a_1 , given by

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

Essentially we are left with this, so we are now left with decoding a_0 , a_4 , a_3 , a_2 , and a_1 , so first we try to decode the r^{th} order terms then we try to decode $r - 1$ it or it term and finally so here we first decoded the terms related to.

(Refer Slide Time: 47:49)

Decoding of Reed-Muller code

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

$$(b_0^{(1)}, b_1^{(1)}, \dots, b_{15}^{(1)}) = a_0 \mathbf{v}_0 + a_4 \mathbf{v}_4 + a_3 \mathbf{v}_3 + a_2 \mathbf{v}_2 + a_1 \mathbf{v}_1$$
- We can see that sum of every two components of $\mathbf{v}_0, \mathbf{v}_4, \mathbf{v}_3, \mathbf{v}_2$ starting from first is zero, whereas for \mathbf{v}_1 it is 1.
- Therefore we can form eight independent equations for a_1 , given by

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

Second order, now we will try to decode these terms which are related to the first order and we will again follow the same procedure, what we are going to do is we are again going to look at.

(Refer Slide Time: 48:05)

Decoding of Reed-Muller code

• 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
$\rightarrow v_1$	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 1 0 0 0 1 0 0 0 1 0 0 0 1
$v_1 v_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 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
$v_2 v_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$v_3 v_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

This G matrix and we are going to look at the bit so we are now looking at because the contribution of these have been removed so we are now looking at this G matrix, we are only looking at this, assuming we have correctly decoded a_{12} , a_{13} , a_{14} contribution of these have been removed so only we thing we are left with is this, now if you notice if you add up 2 rows like this consider these 2 rows so what you would have noticed for all other except v_1 we will get 0.

(Refer Slide Time: 48:48)

Decoding of Reed-Muller code

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

$$(b_0^{(1)}, b_1^{(1)}, \dots, b_{15}^{(1)}) = \underbrace{a_0 \mathbf{v}_0 + a_4 \mathbf{v}_4 + a_3 \mathbf{v}_3 + a_2 \mathbf{v}_2 + a_1 \mathbf{v}_1}_{\text{sum of components}}$$
- We can see that sum of every two components of $\mathbf{v}_0, \mathbf{v}_4, \mathbf{v}_3, \mathbf{v}_2$ starting from first is zero, whereas for \mathbf{v}_1 it is 1.
- Therefore we can form eight independent equations for a_1 , given by

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

So in other words I can get 8 independent views about what a_1 by just looking at these 2 columns of this matrix so I can I am getting 8 independent equations for a_1 and again I will go for majority logic decoding so whatever majority of them are saying I will decide in favor of that and the same procedure can be repeated to find out.

(Refer Slide Time: 49:14)

Decoding of Reed-Muller code

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

$$(b_0^{(1)}, b_1^{(1)}, \dots, b_{15}^{(1)}) = \underbrace{a_0 \mathbf{v}_0 + a_4 \mathbf{v}_4 + a_3 \mathbf{v}_3 + a_2 \mathbf{v}_2 + a_1 \mathbf{v}_1}_{\text{sum of every two components of } \mathbf{v}_0, \mathbf{v}_4, \mathbf{v}_3, \mathbf{v}_2 \text{ starting from first is zero, whereas for } \mathbf{v}_1 \text{ it is 1.}}$$
- We can see that sum of every two components of $\mathbf{v}_0, \mathbf{v}_4, \mathbf{v}_3, \mathbf{v}_2$ starting from first is zero, whereas for \mathbf{v}_1 it is 1.
- Therefore we can form eight independent equations for a_1 , given by

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

What a_2, a_3, a_4 are again.

(Refer Slide Time: 49:17)

Decoding of Reed-Muller code

- 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

$$\begin{aligned} a_2 &= b_0^{(1)} + b_2^{(1)}, a_1 = b_8^{(1)} + b_{10}^{(1)} \\ a_2 &= b_1^{(1)} + b_3^{(1)}, a_1 = b_9^{(1)} + b_{11}^{(1)} \\ a_2 &= b_4^{(1)} + b_6^{(1)}, a_1 = b_{12}^{(1)} + b_{14}^{(1)} \\ a_2 &= b_5^{(1)} + b_7^{(1)}, a_1 = b_{13}^{(1)} + b_{15}^{(1)} \end{aligned}$$

- We can form eight independent equations for a_3 , given by

$$\begin{aligned} a_3 &= b_0^{(1)} + b_4^{(1)}, a_1 = b_8^{(1)} + b_{12}^{(1)} \\ a_3 &= b_1^{(1)} + b_5^{(1)}, a_1 = b_9^{(1)} + b_{13}^{(1)} \\ a_3 &= b_2^{(1)} + b_6^{(1)}, a_1 = b_{10}^{(1)} + b_{14}^{(1)} \\ a_3 &= b_3^{(1)} + b_7^{(1)}, a_1 = b_{11}^{(1)} + b_{15}^{(1)} \end{aligned}$$

This is just a typo, this should be a_2 here and similarly this is a_3 here.

(Refer Slide Time: 49:31)

Decoding of Reed-Muller code

- We can form eight independent equations for a_4 , given by

$$\begin{aligned} a_4 &= b_0^{(1)} + b_8^{(1)}, a_1 = b_4^{(1)} + b_{12}^{(1)} \\ a_4 &= b_1^{(1)} + b_9^{(1)}, a_1 = b_5^{(1)} + b_{13}^{(1)} \\ a_4 &= b_2^{(1)} + b_{10}^{(1)}, a_1 = b_6^{(1)} + b_{14}^{(1)} \\ a_4 &= b_3^{(1)} + b_{11}^{(1)}, a_1 = b_7^{(1)} + b_{15}^{(1)} \end{aligned}$$
- Equations for decoding a_1 can be written as

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

And this is a_4 here, okay now this is exactly same procedure I followed for a_1 we are using for a_2, a_3, a_4, a_4 .

(Refer Slide Time: 49:47)

Decoding of Reed-Muller code

- We can form eight independent equations for a_4 , given by

$$\begin{aligned} a_4 &= b_0^{(1)} + b_8^{(1)}, a_1 = b_4^{(1)} + b_{12}^{(1)} \\ a_4 &= b_1^{(1)} + b_9^{(1)}, a_1 = b_5^{(1)} + b_{13}^{(1)} \\ a_4 &= b_2^{(1)} + b_{10}^{(1)}, a_1 = b_6^{(1)} + b_{14}^{(1)} \\ a_4 &= b_3^{(1)} + b_{11}^{(1)}, a_1 = b_7^{(1)} + b_{15}^{(1)} \end{aligned}$$
- Equations for decoding a_1 can be written as

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

A4 and then we are getting independent equations, 8 independent equations and we take majority decision in decoding these, now once we have decoded.

(Refer Slide Time: 49:59)

Decoding of Reed-Muller code

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

$$\begin{aligned}\mathbf{r}^{(2)} &= (r_0^{(2)}, r_1^{(2)}, \dots, r_{15}^{(2)}) \\ &= \mathbf{r}^{(1)} - \underline{a_4 \mathbf{v}_4 - a_3 \mathbf{v}_3 - a_2 \mathbf{v}_2 - a_1 \mathbf{v}_1}\end{aligned}$$

- In absence of errors, we have

$$\mathbf{r}^{(2)} = \underline{a_0 \mathbf{v}_0} = (\underline{a_0}, \underline{a_0}, \dots, \underline{a_0})$$

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

a_1, a_2, a_3, a_4 we will then remove the contribution of this from the receive sequence so our receive sequence $\mathbf{r}^{(1)}$ we remove this so what we are now left is the term containing \mathbf{v}_0 so we only left with a_0 , so now we have 16 opinion about a_0 and again we take a majority decision and that is how we decide in favor of a_0 so this in a nut shell.

(Refer Slide Time: 50:31)

Decoding of Reed-Muller code

- After decoding a_1, a_2, a_3, a_4 , we create a modified received vector $\mathbf{r}^{(2)}$
$$\mathbf{r}^{(2)} = (r_0^{(2)}, r_1^{(2)}, \dots, r_{15}^{(2)})$$
$$= \mathbf{r}^{(1)} - a_4 \mathbf{v}_4 - a_3 \mathbf{v}_3 - a_2 \mathbf{v}_2 - a_1 \mathbf{v}_1$$
- In absence of errors, we have
$$\mathbf{r}^{(2)} = a_0 \mathbf{v}_0 = (a_0, a_0, \dots, a_0)$$
- a_0 is decoded to be the value of majority of the bits in $\mathbf{r}^{(2)}$.

As how we are decoding a Reed-Muller code, so first we try to decode the r^{th} terms then $r - 1$ and like that and the key is look at the generator matrix and from there try to find out combinations of bits which will give independent opinion about a particular transmitted bit, so with this I will conclude this discussion on Reed-Muller, code thank you.

Acknowledgement

Ministry of Human Resource & Development

Prof. Satyaki Roy

Co-ordinator, NPTEL IIT Kanpur

NPTEL Team

Sanjay Pal

Ashish Singh

Badal Pradhan

Tapabrata Das

Ram Chandra

Dilip Tripathi

Manoj Shrivastava

Padam Shukla

**Sanjay Mishra
Shubham Gupta
K. K. Mishra
Aradhana Singh
Sweta
Ashutosh Gairola
Dilip Katiyar
Sharwan
Hari Ram
Bhadra Rao
Puneet Kumar Bajpai
Lalty Dutta
Ajay Kanaujia
Shivendra Kumar Tiwari**

an IIT Kanpur Production

©copyright reserved