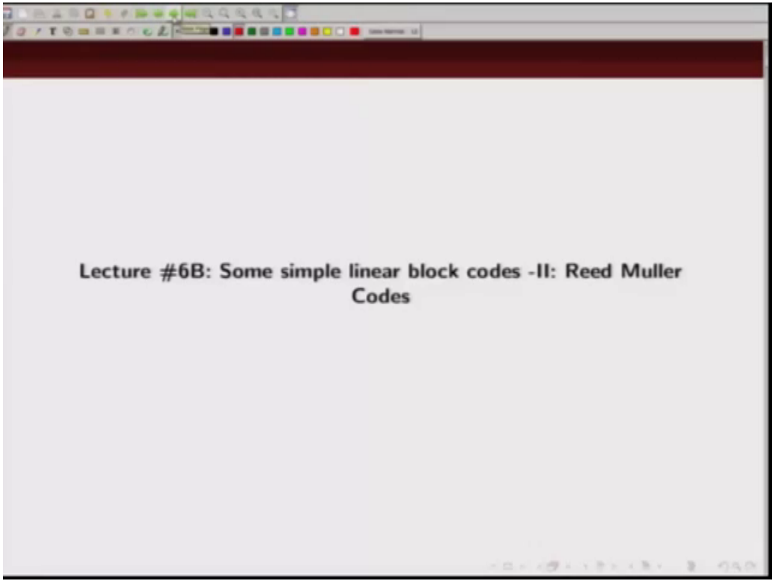
An Introduction to Coding Theory
Professor Adrish Banerji
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Module 03
Lecture Number 12
Some Simple Linear Block Codes-II: Reed Muller Codes

(Refer Slide Time 00:13)

So we will continue our discussions on some simple linear

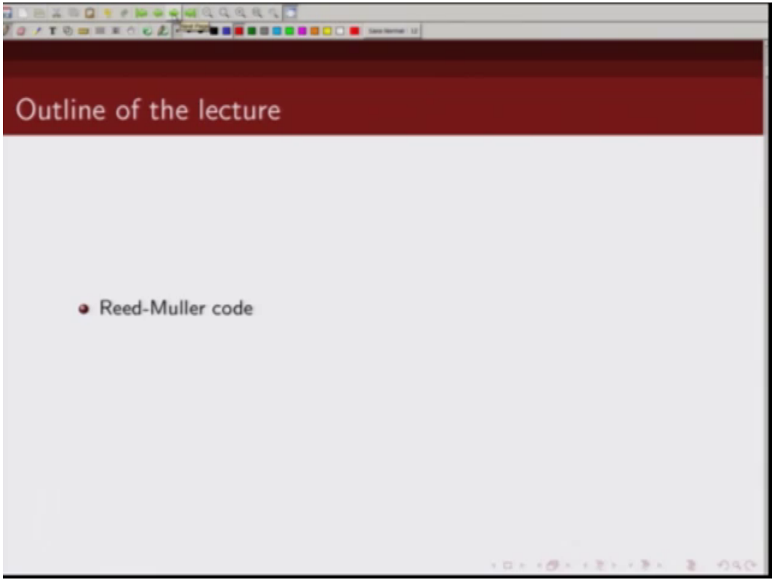
(Refer Slide Time 00:18)



Lecture #6B: Some simple linear block codes -II: Reed Muller Codes

block codes. This time we are going to discuss about Reed-Muller codes.

(Refer Slide Time 00:23)

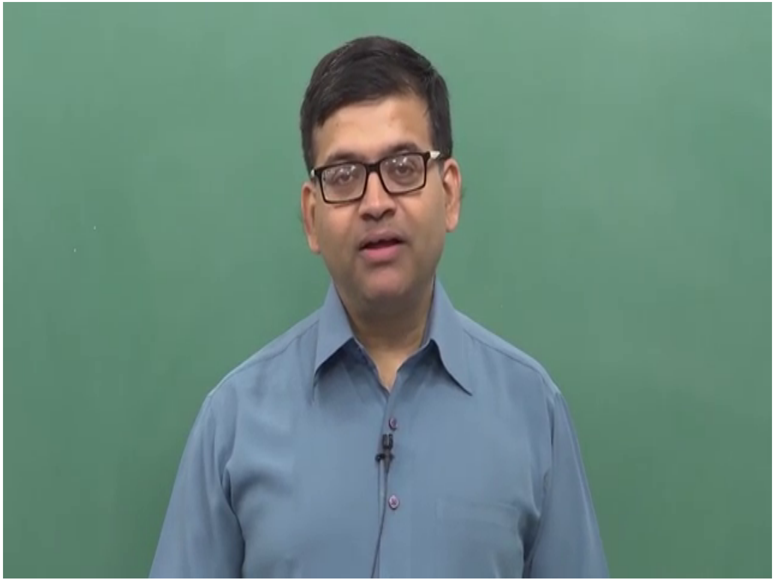


Outline of the lecture

- Reed-Muller code

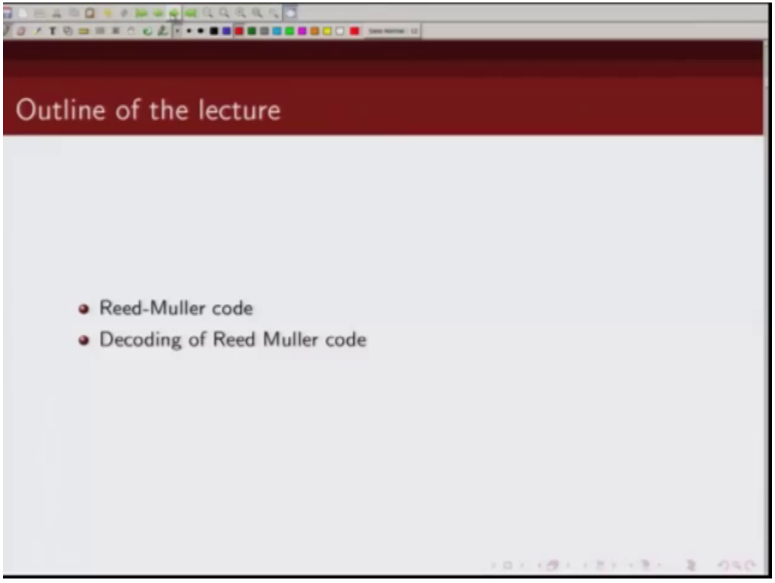
We will talk about their construction. We will give an example.

(Refer Slide Time 00:27)



We will prove some properties of Reed-Muller code and then we will talk about decoding

(Refer Slide Time 00:32)

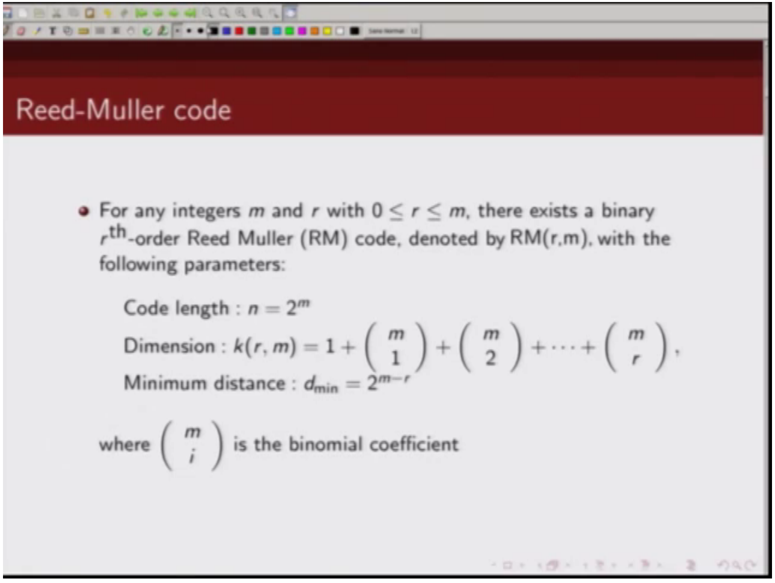


Outline of the lecture

- Reed-Muller code
- Decoding of Reed Muller code

of Reed-Muller code. So for

(Refer Slide Time 00:36)



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 : $k(r,m) = 1 + \begin{pmatrix} m \\ 1 \end{pmatrix} + \begin{pmatrix} m \\ 2 \end{pmatrix} + \cdots + \begin{pmatrix} m \\ r \end{pmatrix}$,

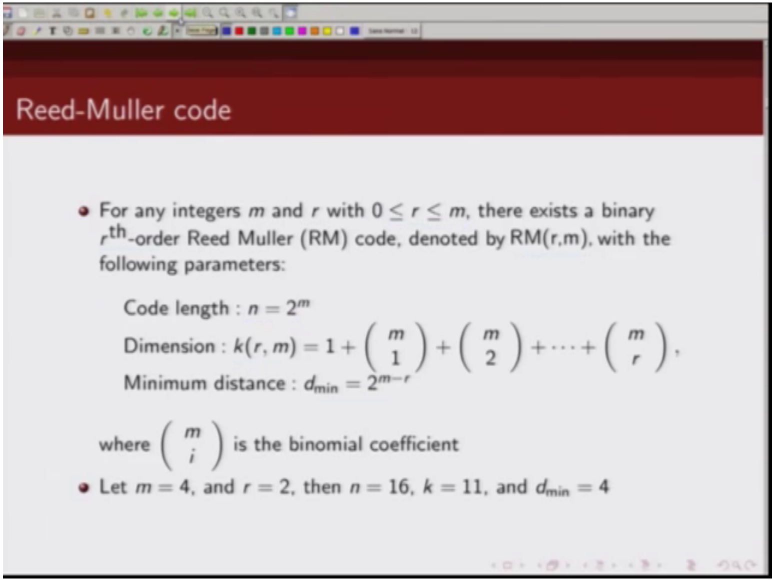
  Minimum distance : $d_{min} = 2^{m-r}$

  where $\begin{pmatrix} m \\ i \end{pmatrix}$ is the binomial coefficient

any integer m and r such that r lies between, r is greater than zero and less than equal to m, there exists a binary rth order Reed-Muller code which we denote by this parameter R and m. Reed-Muller code has following code properties. So the length of the code is 2 raised to power m and this dimension k is given by 1 plus m choose 1 plus m choose 2 up to m choose r and the minimum distance of the code is given by 2 raised to power m minus r.

So let us take an example.

(Refer Slide Time 01:28)



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 : $n = 2^m$

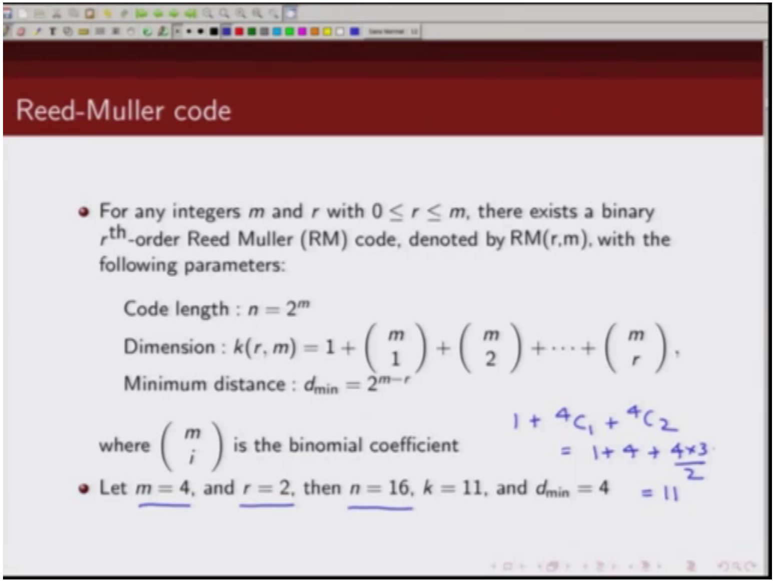
  Dimension : $k(r,m) = 1 + \binom{m}{1} + \binom{m}{2} + \cdots + \binom{m}{r}$,

  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 $d_{min} = 4$

Let us take m to be 4 and r to be 2. So in this case, the length of the codeword would be 2 raised to power 4 which is 16 and since the order of this Reed-Muller code is 2, so this k will be 1 plus 4 C 1 plus 4 C 2. So this will be 1 plus 4 plus 4 times 3 by 2. So this will be equal to 11, 1 plus 4 plus

(Refer Slide Time 02:08)



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

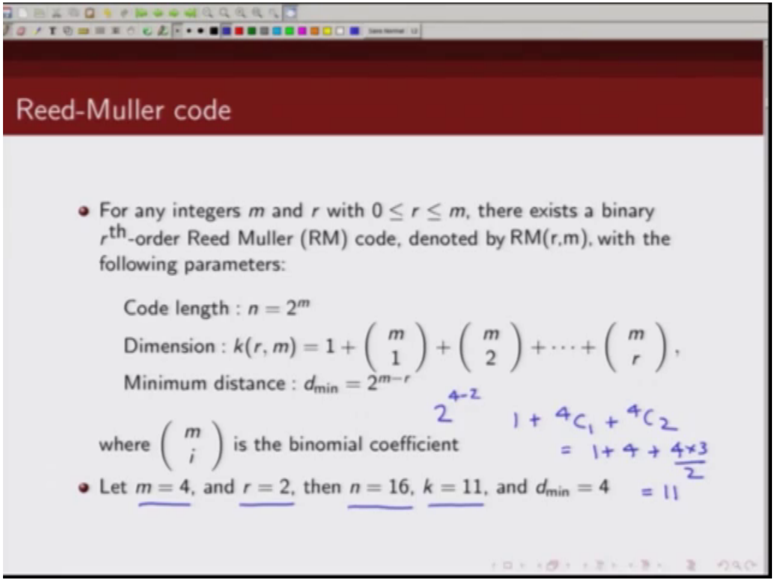
  Minimum distance : $d_{min} = 2^{m-r}$

  where $\binom{m}{i}$ is the binomial coefficient
- Let $\underline{m = 4}$, and $\underline{r = 2}$, then $\underline{n = 16}$, $k = 11$, and $d_{min} = 4$

$$1 + {}^4C_1 + {}^4C_2$$
$$= 1 + 4 + \frac{4 \times 3}{2}$$
$$= 11$$

6. So k is this thing. And minimum distance is 2 raised to power 4 minus 2 which is

(Refer Slide Time 02:18)



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 : $n = 2^m$

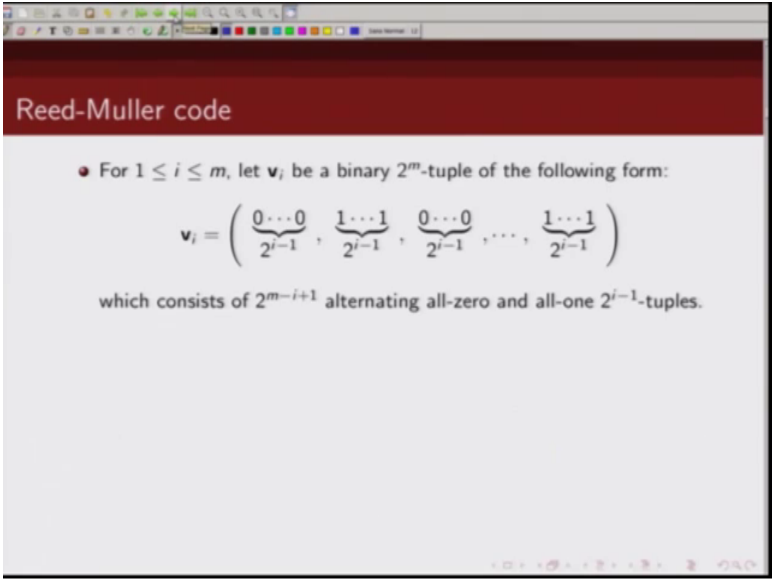
  Dimension : $k(r,m) = 1 + \binom{m}{1} + \binom{m}{2} + \cdots + \binom{m}{r}$,

  Minimum distance : $d_{\min} = 2^{m-r}$ $\quad 2^{4-2}$

  where $\binom{m}{i}$ is the binomial coefficient $\quad 1 + {}^4C_1 + {}^4C_2 = 1 + 4 + \frac{4 \times 3}{2}$
- Let $\underline{m = 4}$, and $\underline{r = 2}$, then $\underline{n = 16}$, $\underline{k = 11}$, and $d_{\min} = 4$ $\quad = 11$

4. Now how

(Refer Slide Time 02:21)



Reed-Muller code

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

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

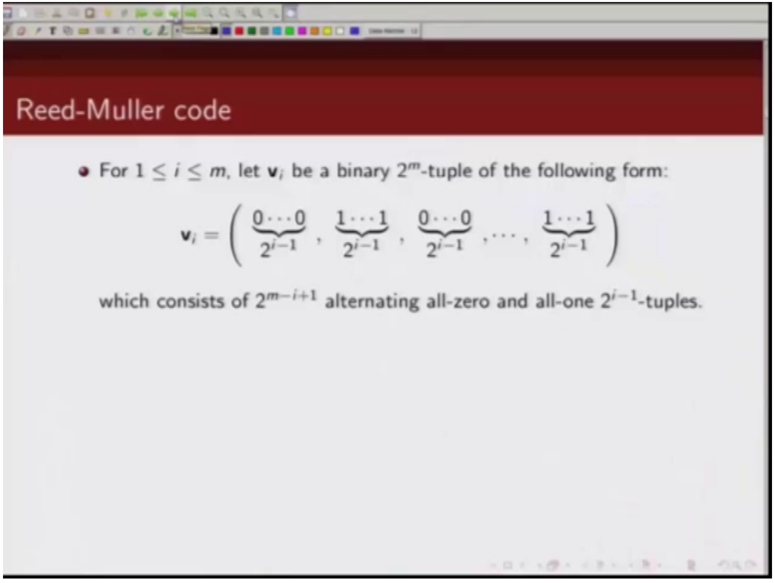
  which consists of 2^{m-i+1} alternating all-zero and all-one 2^{i-1}-tuples.

do we construct

(Refer Slide Time 02:22)

a Reed-Muller code? So to do that, let's define, so we are

(Refer Slide Time 02:28)



Reed-Muller code

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

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

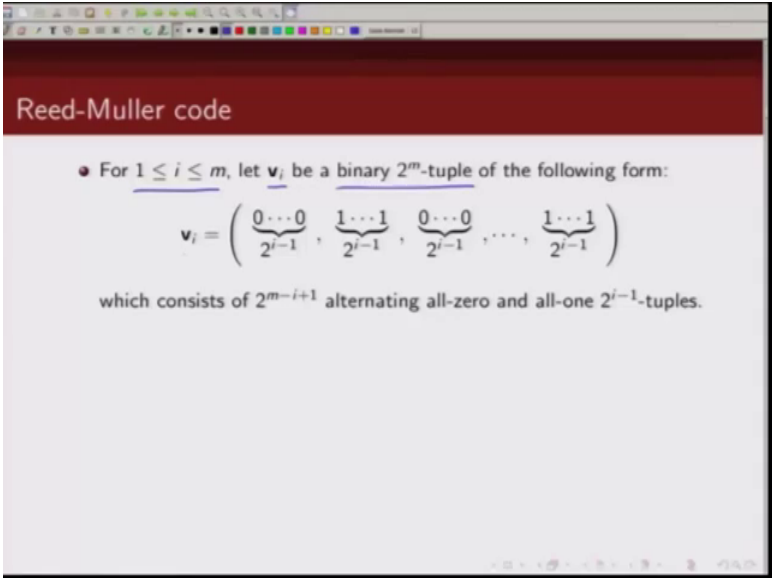
which consists of 2^{m-i+1} alternating all-zero and all-one 2^{i-1}-tuples.

defining a binary m-tuple. Let's call it 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

(Refer Slide Time 02:48)

0's and 1's. So v i

(Refer Slide Time 02:52)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \cdots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

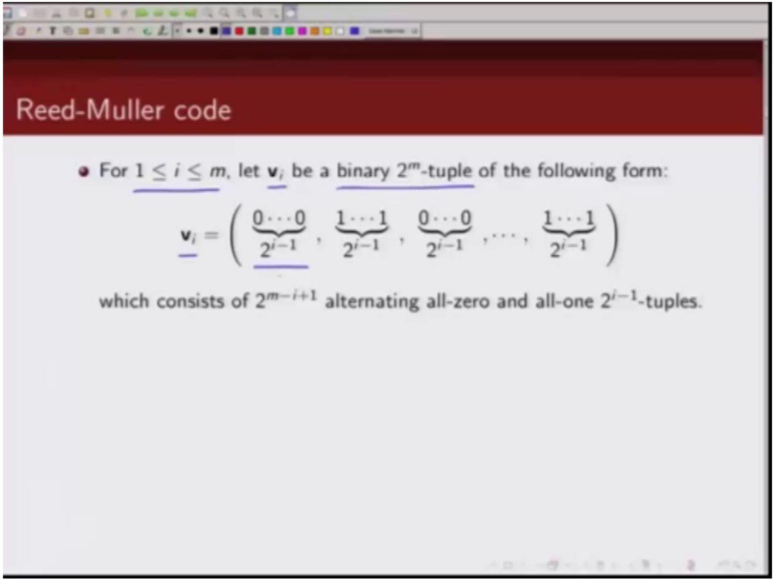
which consists of 2^{m-i+1} alternating all-zero and all-one 2^{i-1}-tuples.

is run of 0's for

(Refer Slide Time 02:57)

2 i minus 1 times then run of 1's for 2 i's minus 1, like that.

(Refer Slide Time 03:02)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

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

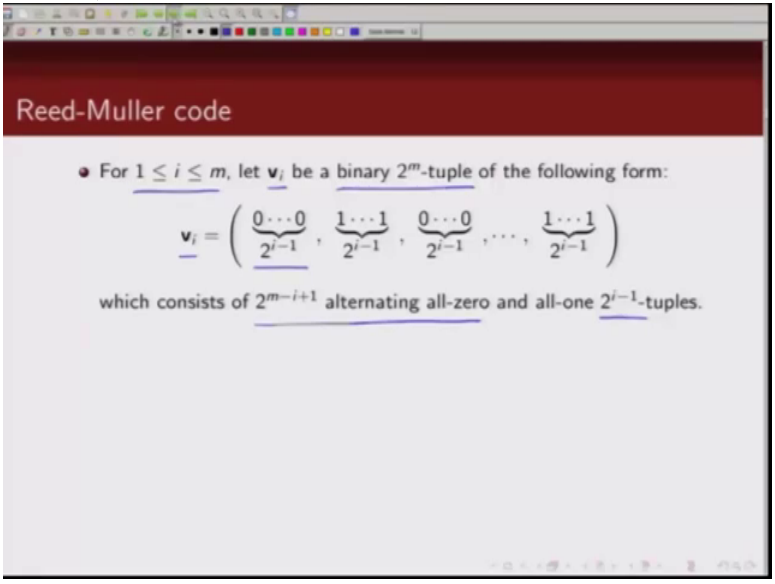
So this v i consists of 2 m minus i plus 1 alternating 0's and 1's and where each of these runs of 0's and 1's are for

(Refer Slide Time 03:18)

2 i minus 1.

Let's take an example.

(Refer Slide Time 03:22)



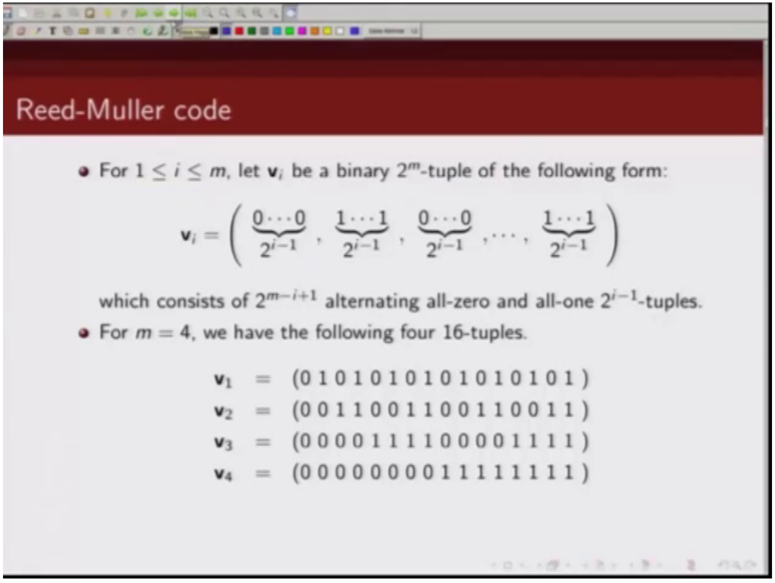
Reed-Muller code

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

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

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

(Refer Slide Time 03:23)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

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.

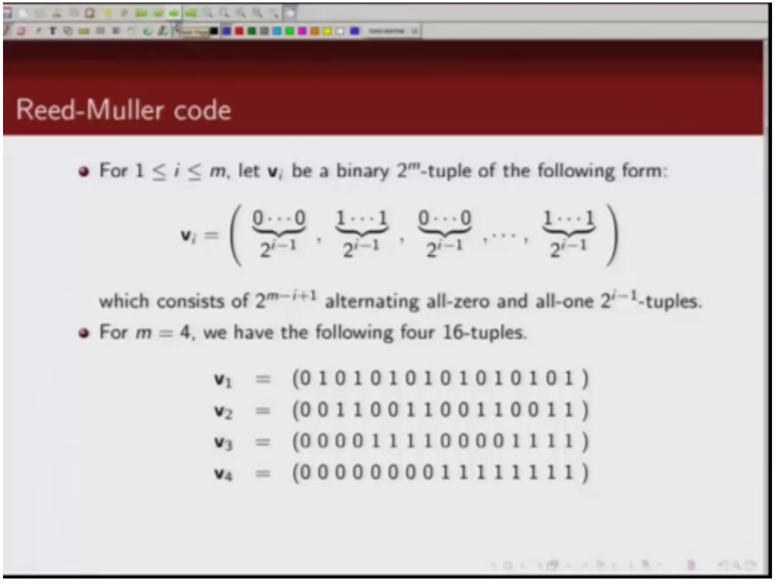
$$\begin{aligned}
\mathbf{v}_1 &= (0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1) \\
\mathbf{v}_2 &= (0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1) \\
\mathbf{v}_3 &= (0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1) \\
\mathbf{v}_4 &= (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1)
\end{aligned}$$

Let's consider m to be 4,

(Refer Slide Time 03:26)

m to be 4. Then this m-tuples are

(Refer Slide Time 03:31)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

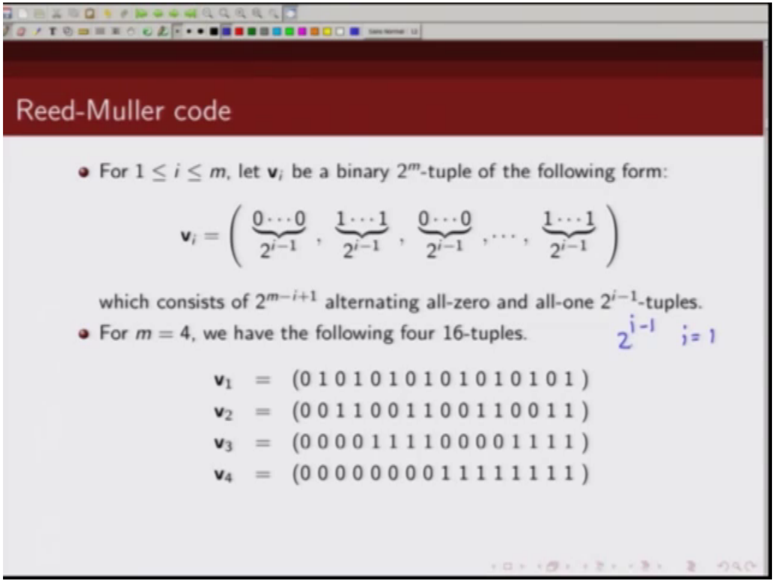
which consists of 2^{m-i+1} alternating all-zero and all-one 2^{i-1}-tuples.

- For $m = 4$, we have the following four 16-tuples.

$$\begin{aligned}
\mathbf{v}_1 &= (0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1) \\
\mathbf{v}_2 &= (0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1) \\
\mathbf{v}_3 &= (0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1) \\
\mathbf{v}_4 &= (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1)
\end{aligned}$$

2 raised to power 4 that is 16, Ok. So what is v 1? Now v 1 should have runs of 0's and 1's where this run is 2 i minus 1. So when i is 1, this is 1.

(Refer Slide Time 03:51)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

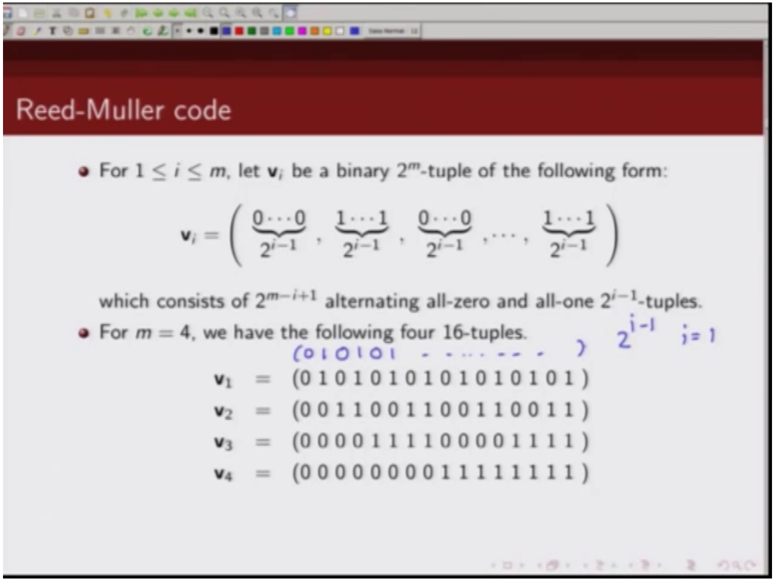
which consists of 2^{m-i+1} alternating all-zero and all-one 2^{i-1}-tuples.

- For $m = 4$, we have the following four 16-tuples. 2^{i-1} $i = 1$

$$\begin{aligned}
\mathbf{v}_1 &= (0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1) \\
\mathbf{v}_2 &= (0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1) \\
\mathbf{v}_3 &= (0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1) \\
\mathbf{v}_4 &= (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1)
\end{aligned}$$

So that means we should have v 1 is zero, because that's a run of 1 then followed by run of 1 one time then followed by 0 one time then 1 one time, so like that it will continue for this block of 16. Now what is

(Refer Slide Time 04:14)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

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.

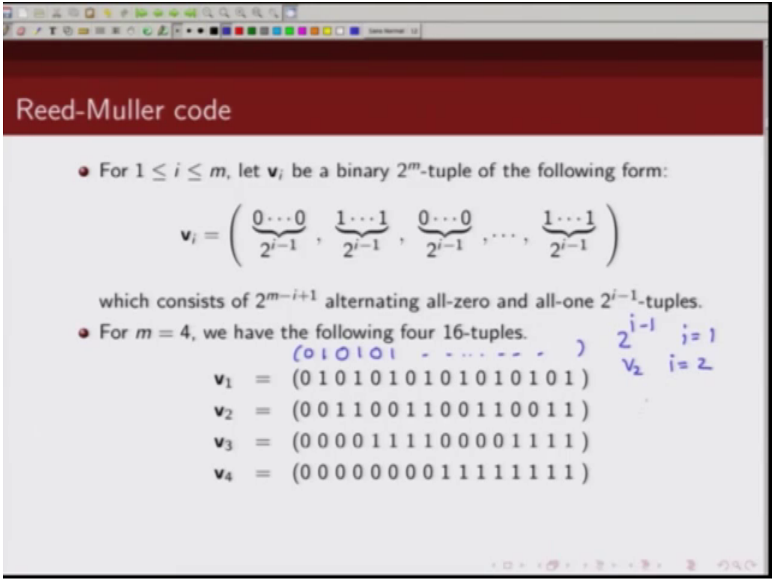
(0 1 0 1 0 1 . . . .)

2^{i-1} $i = 1$

$$\begin{aligned}
\mathbf{v}_1 &= (0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1) \\
\mathbf{v}_2 &= (0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1) \\
\mathbf{v}_3 &= (0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1) \\
\mathbf{v}_4 &= (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1)
\end{aligned}$$

v 2? For v 2, i is 2. So

(Refer Slide Time 04:19)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

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.

(0 1 0 1 0 1 . . . .)

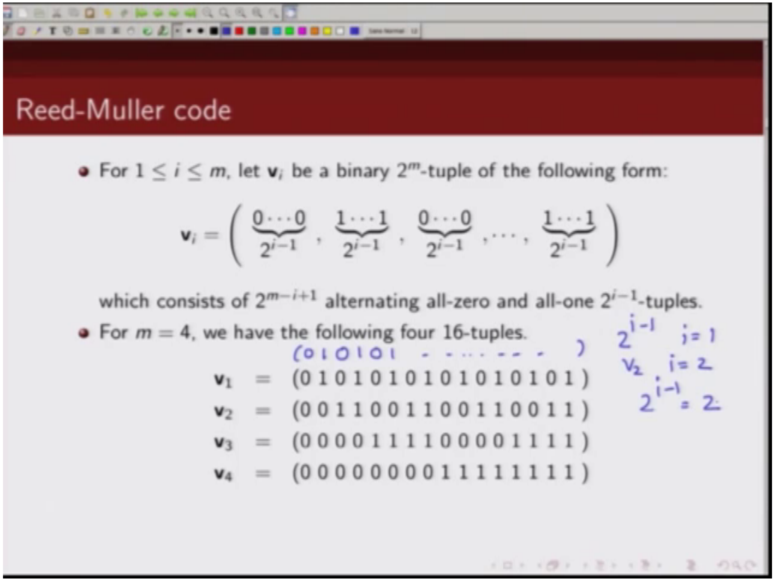
2^{i-1} $i = 1$

V_2 $i = 2$

$$\begin{aligned}
\mathbf{v}_1 &= (0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1) \\
\mathbf{v}_2 &= (0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1) \\
\mathbf{v}_3 &= (0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1) \\
\mathbf{v}_4 &= (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1)
\end{aligned}$$

2 i minus 1 would be, in this case 2. So we should have

(Refer Slide Time 04:25)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

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.

$(0 1 0 1 0 1 \; - - \cdots - - \;)$ $\quad 2^{i-1} \quad i = 1$

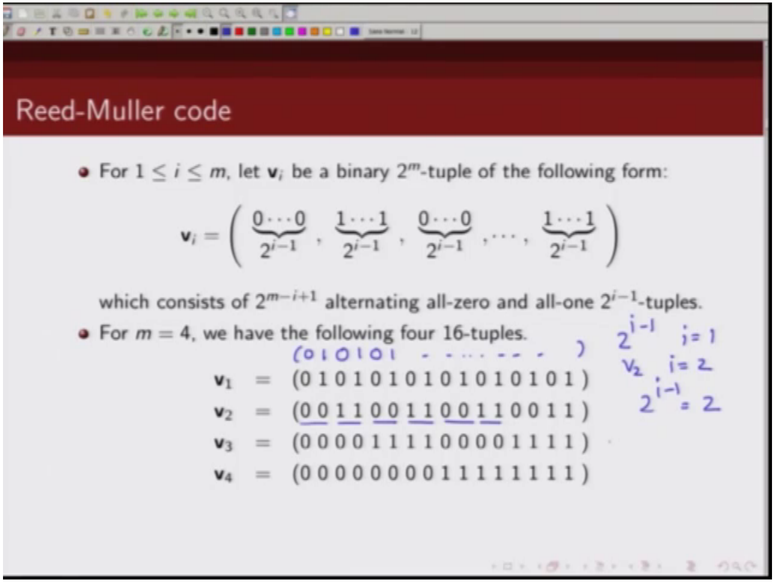
$v_2 \; , \; i = 2$

$2^{i-1} = 2$

$$\begin{aligned} \mathbf{v}_1 &= (0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1) \\ \mathbf{v}_2 &= (0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1) \\ \mathbf{v}_3 &= (0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1) \\ \mathbf{v}_4 &= (0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1) \end{aligned}$$

two runs of 0 followed by run of 1 which is repeated twice, run of 0 repeated twice, 1 0 1 this you continue up to block size of 16. What about

(Refer Slide Time 04:45)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

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.

$(0 1 0 1 0 1 \; - - \cdots - - \;)$ $\quad 2^{i-1} \quad i = 1$

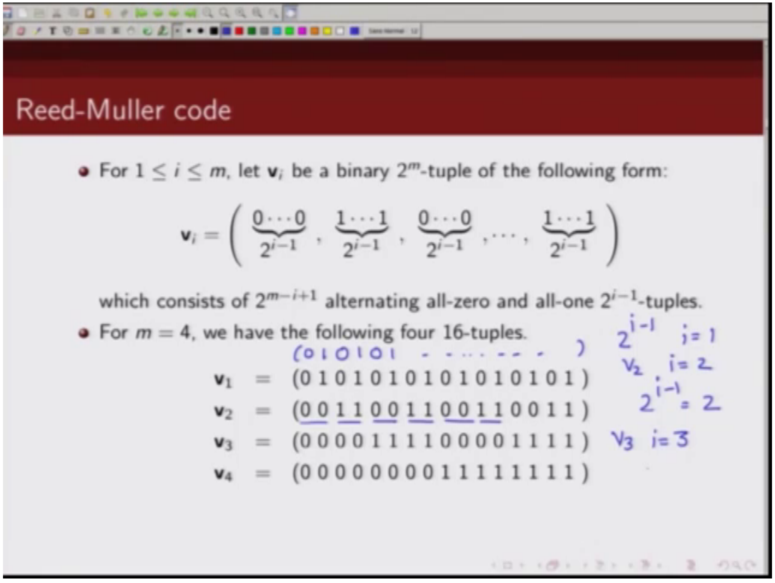
$v_2 \; , \; i = 2$

$2^{i-1} = 2$

$$\begin{aligned} \mathbf{v}_1 &= (0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1) \\ \mathbf{v}_2 &= (\underline{0 0}\,\underline{1 1}\,\underline{0 0}\,\underline{1 1}\,\underline{0 0}\,\underline{1 1} 0 0 1 1) \\ \mathbf{v}_3 &= (0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1) \\ \mathbf{v}_4 &= (0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1) \end{aligned}$$

3? In this case i is 3. So what

(Refer Slide Time 04:51)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}}\right)$$

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.

(0 1 0 1 0 1 . . . .)

$$\begin{aligned}
\mathbf{v}_1 &= (0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1)\\
\mathbf{v}_2 &= (\underline{0\,0}\,\underline{1\,1}\,\underline{0\,0}\,\underline{1\,1}\,\underline{0\,0}\,\underline{1\,1}\,0\,0\,1\,1)\\
\mathbf{v}_3 &= (0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1)\\
\mathbf{v}_4 &= (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1)
\end{aligned}$$

2^{i-1} $i=1$

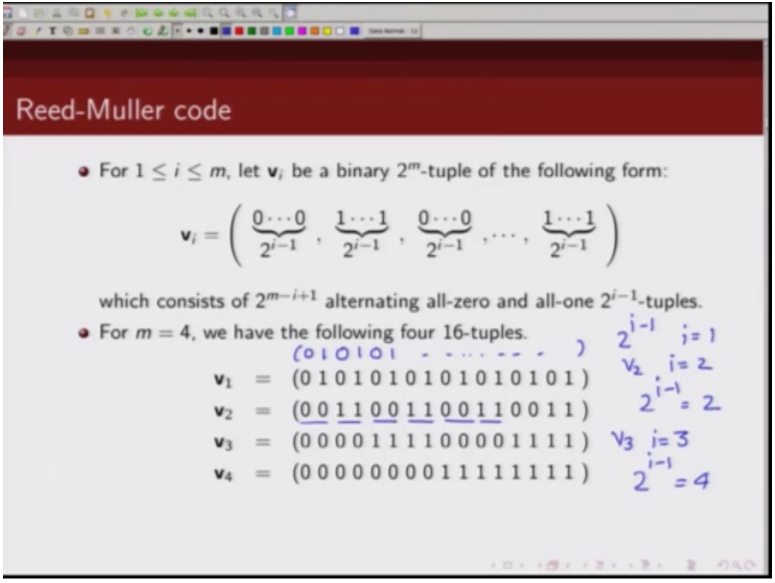
V_2 $i=2$

$2^{i-1}=2$

V_3 $i=3$

will be 2 i minus 1? 2 i minus 1 would be 4. So you have

(Refer Slide Time 04:58)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}}\right)$$

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.

(0 1 0 1 0 1 . . . .)

$$\begin{aligned}
\mathbf{v}_1 &= (0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1)\\
\mathbf{v}_2 &= (\underline{0\,0}\,\underline{1\,1}\,\underline{0\,0}\,\underline{1\,1}\,\underline{0\,0}\,\underline{1\,1}\,0\,0\,1\,1)\\
\mathbf{v}_3 &= (0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1)\\
\mathbf{v}_4 &= (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1)
\end{aligned}$$

2^{i-1} $i=1$

V_2 $i=2$

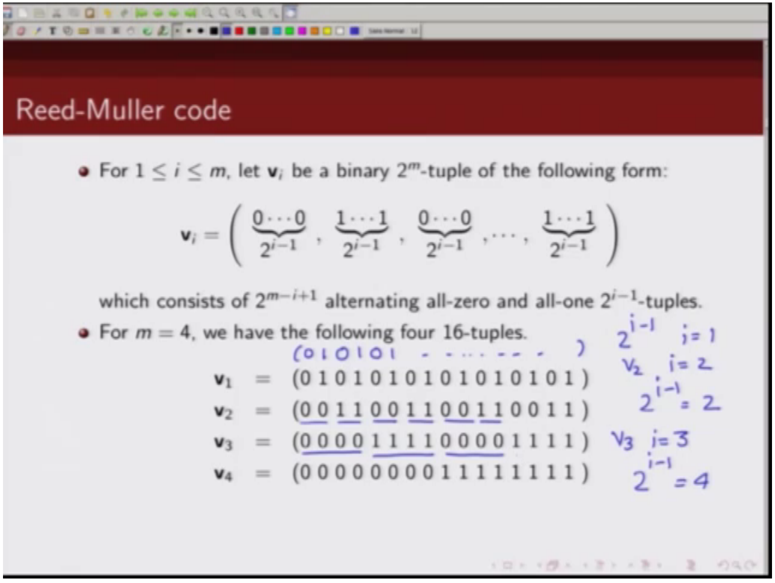
$2^{i-1}=2$

V_3 $i=3$

$2^{i-1}=4$

runs of 0 for 4 times followed by runs of 1 four times then again runs of 0

(Refer Slide Time 05:07)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

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.

(0 1 0 1 0 1 ....)

$$\begin{aligned} \mathbf{v}_1 &= (0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1) \\ \mathbf{v}_2 &= (0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1) \\ \mathbf{v}_3 &= (0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1) \\ \mathbf{v}_4 &= (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1) \end{aligned}$$

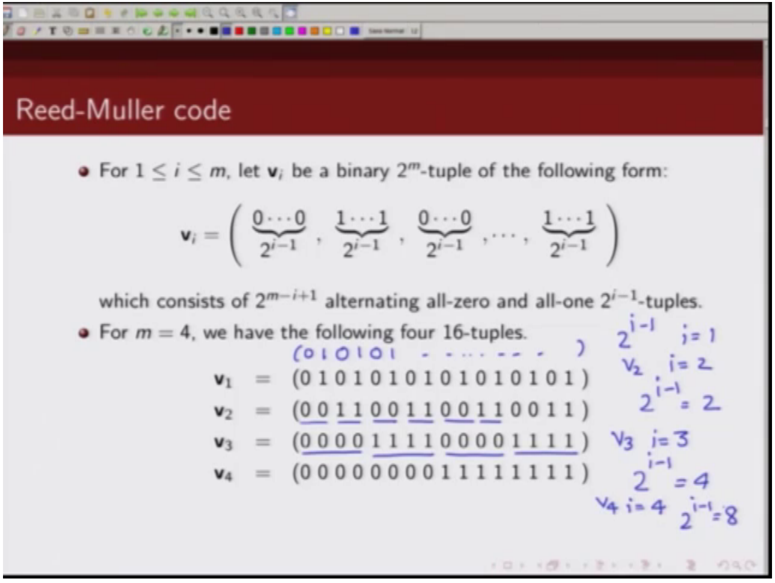
2^{i-1} $i = 1$

V_2 $i = 2$ $2^{i-1} = 2$

V_3 $i = 3$ $2^{i-1} = 4$

and run of 1. What about v 4? Here i is 4. So 2 i minus 1 will be 8. So we have

(Refer Slide Time 05:21)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

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.

(0 1 0 1 0 1 ....)

$$\begin{aligned} \mathbf{v}_1 &= (0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1) \\ \mathbf{v}_2 &= (0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1) \\ \mathbf{v}_3 &= (0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1) \\ \mathbf{v}_4 &= (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1) \end{aligned}$$

2^{i-1} $i = 1$

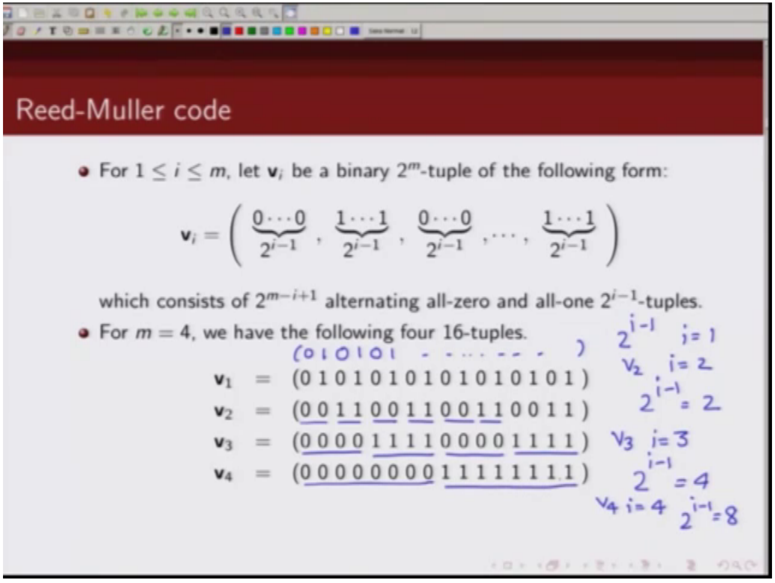
V_2 $i = 2$ $2^{i-1} = 2$

V_3 $i = 3$ $2^{i-1} = 4$

V_4 $i = 4$ $2^{i-1} = 8$

runs of 0's for eight times followed by runs of 1

(Refer Slide Time 05:26)



Reed-Muller code

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

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

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.

$$(0 1 0 1 0 1 \; - \; - \cdots - - \; - \;)$$

$$\begin{aligned}
\mathbf{v}_1 &= (0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1)\\
\mathbf{v}_2 &= (\underline{0 0}\,\underline{1 1}\,\underline{0 0}\,\underline{1 1}\,\underline{0 0}\,\underline{1 1}\,0 0 1 1)\\
\mathbf{v}_3 &= (\underline{0 0 0 0}\,\underline{1 1 1 1}\,\underline{0 0 0 0}\,\underline{1 1 1 1})\\
\mathbf{v}_4 &= (\underline{0 0 0 0 0 0 0 0}\,\underline{1 1 1 1 1 1 1 1})
\end{aligned}$$

2^{i-1} $i = 1$

V_2 $i = 2$

$2^{i-1} = 2$

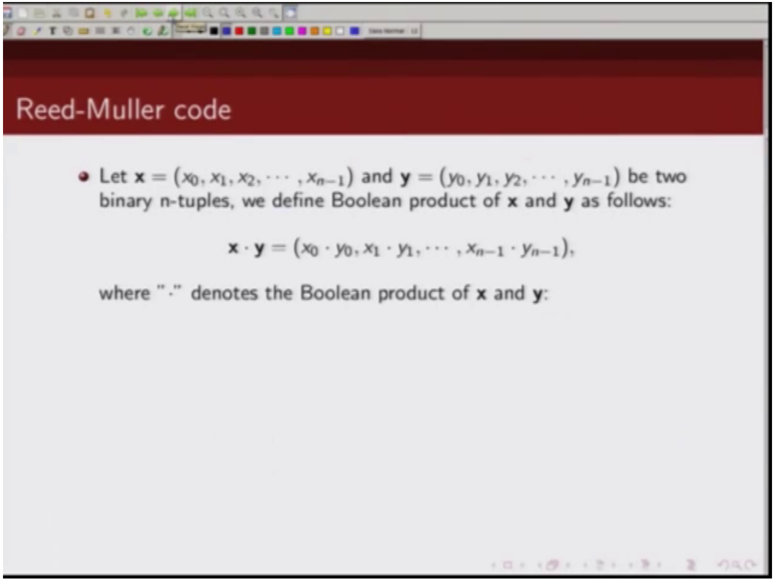
V_3 $i = 3$

$2^{i-1} = 4$

V_4 $i = 4$ $2^{i-1} = 8$

eight times. So that is how we define this binary m-tuple for each of this i going from 1 to m.

(Refer Slide Time 05:39)



Reed-Muller code

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

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

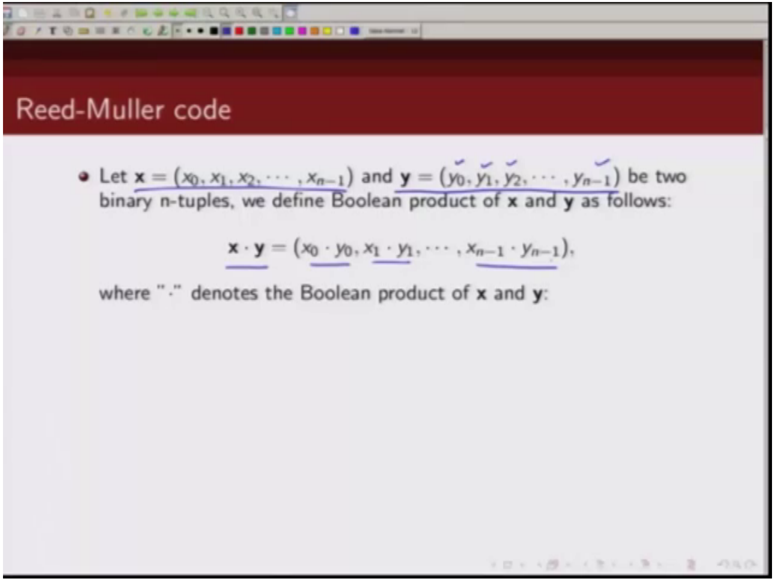
where "$\cdot$" denotes the Boolean product of $\mathbf{x}$ and $\mathbf{y}$:

Next we define a Boolean product. How do we define a Boolean product? Let's say we have 2 m-tuples, x and y. So I am denoting x by x 0, x 1, x 2, x 3, x n minus 1; similarly denoting y by y 0, y 1, y 2, y n minus 1. Now we define these Boolean products as, so this is bitwise And x 0 dot y 0, x 1 dot y 1, x 2 dot y 2 up to x n minus 1 dot y n minus 1. So this x 0 dot y 0 will be 1 only if both x 0 and y 0 are 1.

(Refer Slide Time 06:30)

Otherwise it will be 0 and same with others. So

(Refer Slide Time 06:35)



Reed-Muller code

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

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

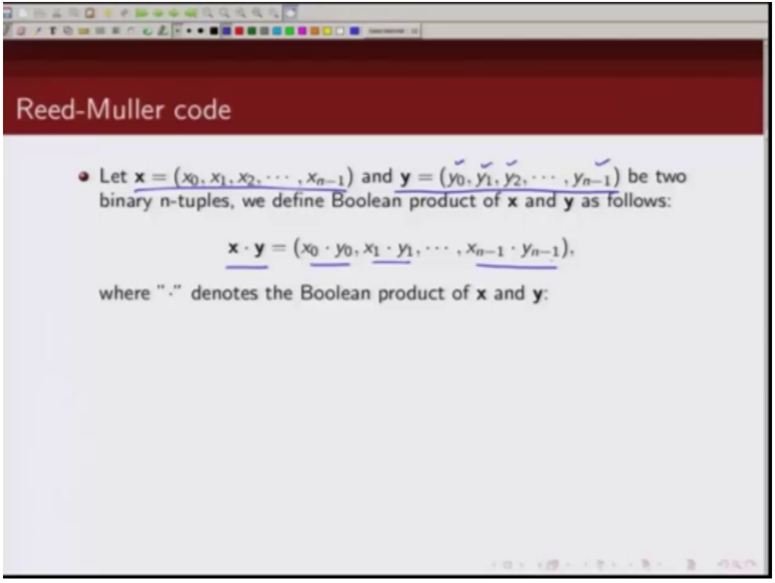
where "$\cdot$" denotes the Boolean product of $\mathbf{x}$ and $\mathbf{y}$:

x i dot y i will be

(Refer Slide Time 06:37)

1 only if

(Refer Slide Time 06:40)



Reed-Muller code

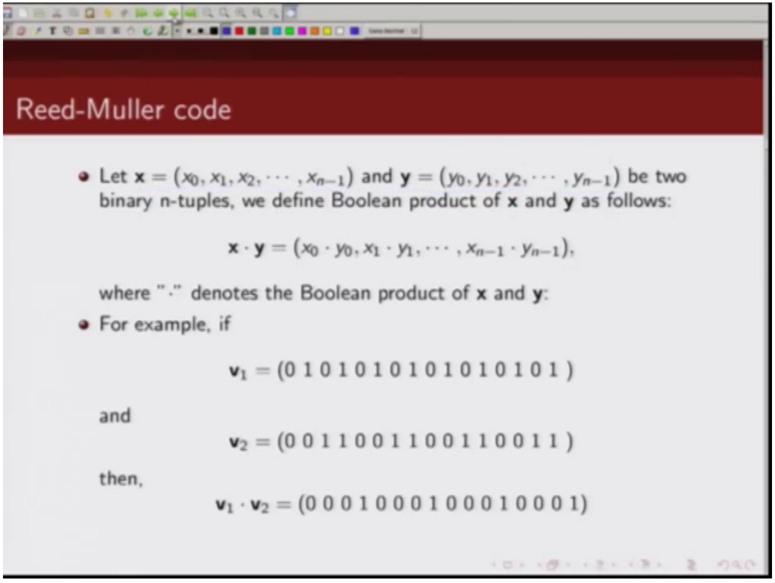
- Let $\mathbf{x} = (x_0, x_1, x_2, \cdots, x_{n-1})$ and $\mathbf{y} = (y_0, y_1, y_2, \cdots, y_{n-1})$ be two binary n-tuples, we define Boolean product of **x** and **y** as follows:

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

where "·" denotes the Boolean product of **x** and **y**;

both of them are 1. So that's how

(Refer Slide Time 06:42)



Reed-Muller code

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

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

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

- For example, if

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

and

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

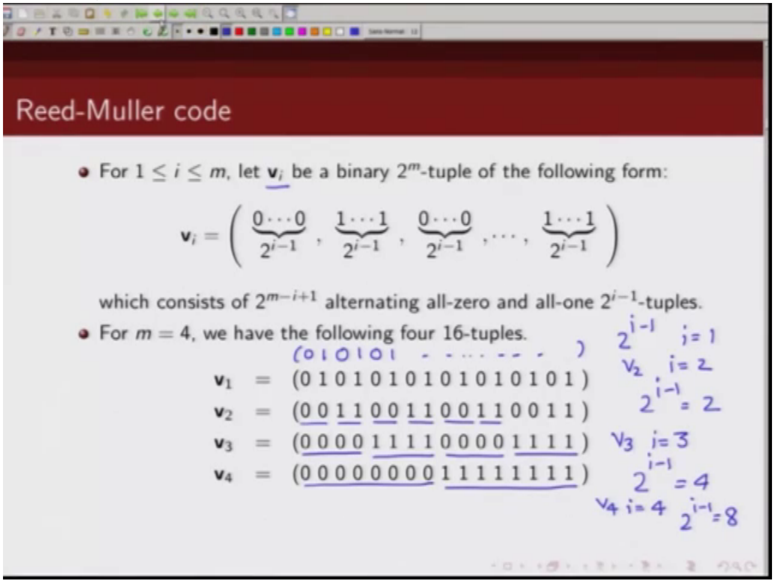
then,

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

we are defining this Boolean product operation.

So let's take an example. This is our v 1, you recall

(Refer Slide Time 06:50)



Reed-Muller code

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

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

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.

$(0\,1\,0\,1\,0\,1\,0\,1 \; - \; - \cdots - - \; - \quad)$

$$\begin{aligned} \mathbf{v}_1 &= (0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1) \\ \mathbf{v}_2 &= (\underline{0\,0}\,\underline{1\,1}\,\underline{0\,0}\,\underline{1\,1}\,\underline{0\,0}\,\underline{1\,1}\,0\,0\,1\,1) \\ \mathbf{v}_3 &= (\underline{0\,0\,0\,0}\,\underline{1\,1\,1\,1}\,\underline{0\,0\,0\,0}\,\underline{1\,1\,1\,1}) \\ \mathbf{v}_4 &= (\underline{0\,0\,0\,0\,0\,0\,0\,0}\,\underline{1\,1\,1\,1\,1\,1\,1\,1}) \end{aligned}$$

2^{i-1} $i = 1$

V_2 , $i = 2$

$2^{i-1} = 2$

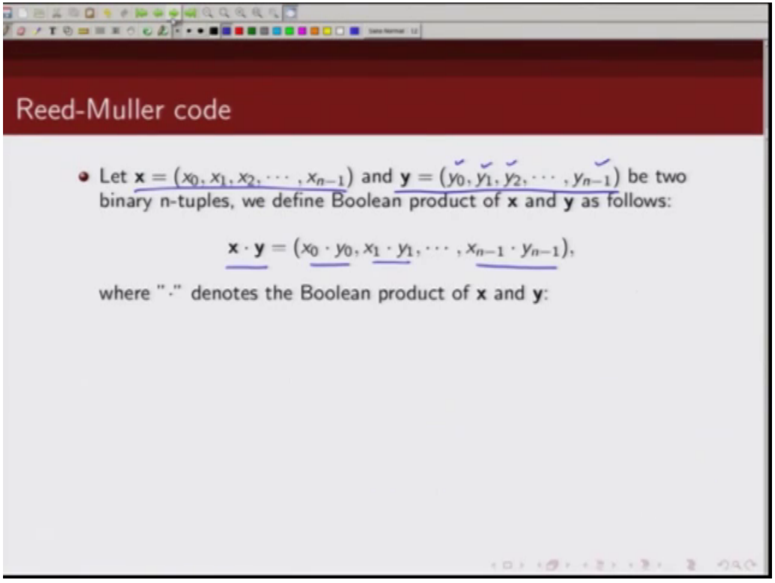
V_3 $i = 3$

$2^{i-1} = 4$

V_4 $i = 4$ $2^{i-1} = 8$

this was our v 1. And this is our v 2. If we define

(Refer Slide Time 06:56)



Reed-Muller code

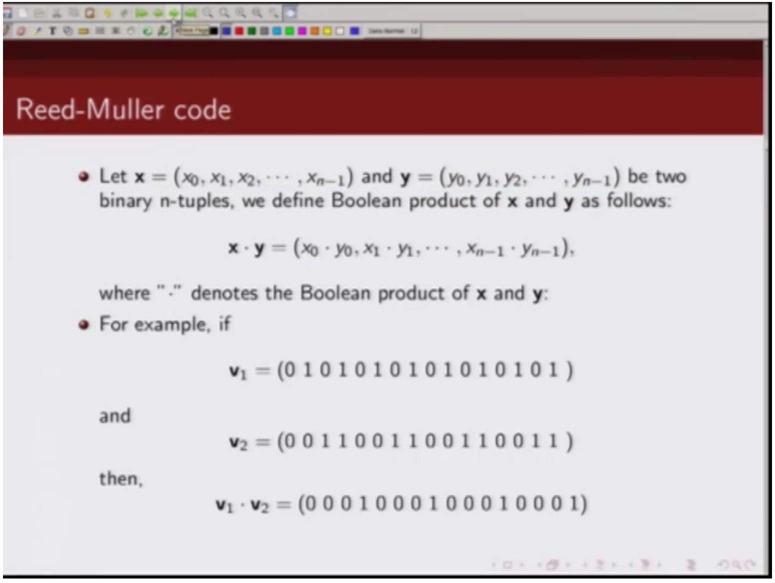
- Let $\mathbf{x} = (x_0, x_1, x_2, \cdots, x_{n-1})$ and $\mathbf{y} = (y_0, y_1, y_2, \cdots, y_{n-1})$ be two binary n-tuples, we define Boolean product of $\mathbf{x}$ and $\mathbf{y}$ as follows:

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

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

Boolean product between v 1 and v 2,

(Refer Slide Time 06:59)



Reed-Muller code

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

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

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

- For example, if

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

and

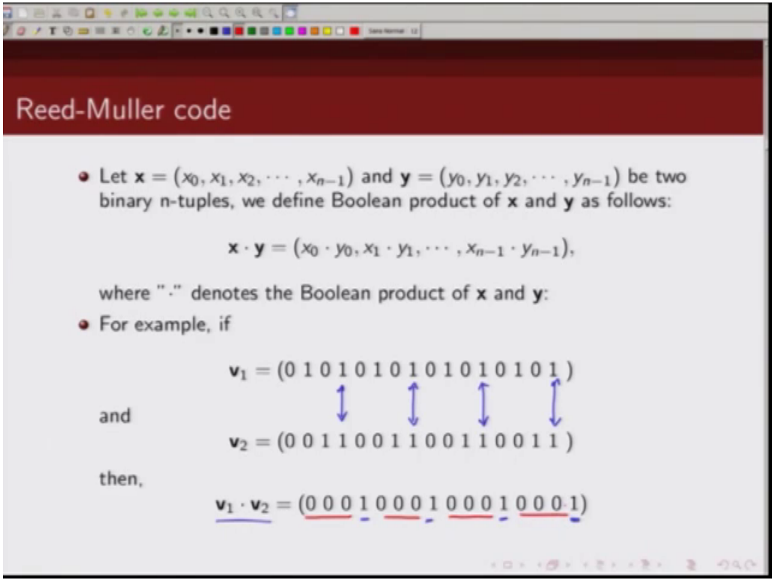
$$\mathbf{v}_2 = (0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1)$$

then,

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

we write it as v it as v 1 dot v 2. And v 1 dot v 2 will be 1 only where v 1 and v 2 both are 1; so which is like this location, number fourth bit, this location then this location and then this location. So you can see it's only one at this fourth, eighth, twelfth and sixteenth location. All other time it’s zero. This is zero for all other times

(Refer Slide Time 07:38)



Reed-Muller code

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

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

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

- For example, if

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

and

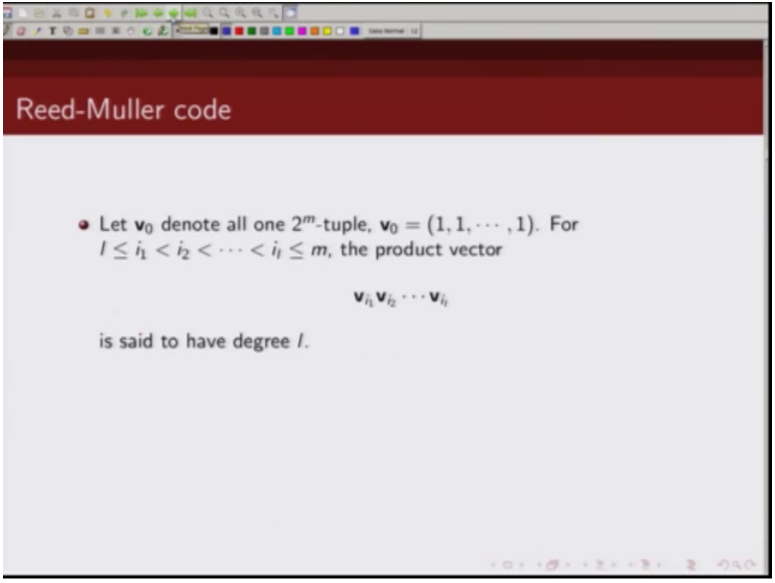
$$\mathbf{v}_2 = (0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1)$$

then,

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

ok; so this is how we define the Boolean product.

(Refer Slide Time 07:47)



Reed-Muller code

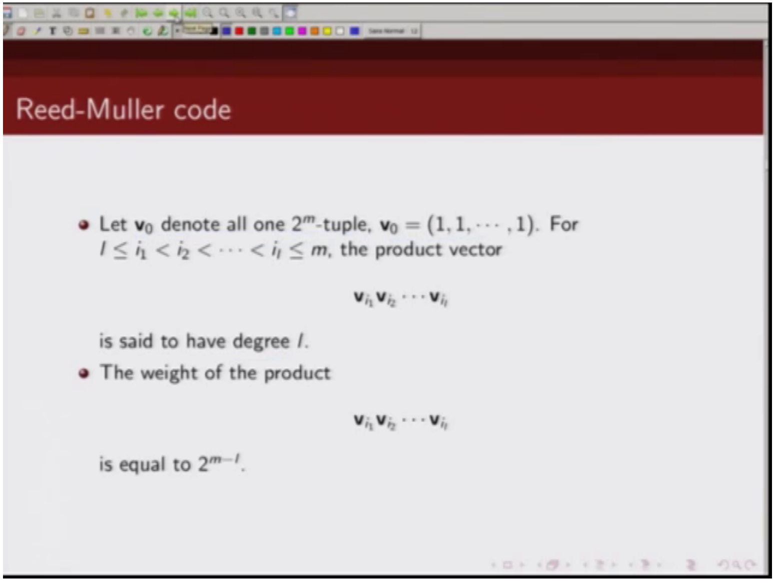
- Let $\mathbf{v}_0$ denote all one 2^m-tuple, $\mathbf{v}_0 = (1, 1, \cdots, 1)$. For $l \leq i_1 < i_2 < \cdots < i_l \leq m$, the product vector

$$\mathbf{v}_{i_1}\mathbf{v}_{i_2} \cdots \mathbf{v}_{i_l}$$

is said to have degree l.

an all 1 tuple, so this v naught is basically all 1s of length 2 raised to power n. 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. And weight of this

(Refer Slide Time 08:31)



Reed-Muller code

- Let $\mathbf{v}_0$ denote all one 2^m-tuple, $\mathbf{v}_0 = (1, 1, \cdots, 1)$. For $l \leq i_1 < i_2 < \cdots < i_l \leq m$, the product vector

$$\mathbf{v}_{i_1}\mathbf{v}_{i_2}\cdots\mathbf{v}_{i_l}$$

is said to have degree l.

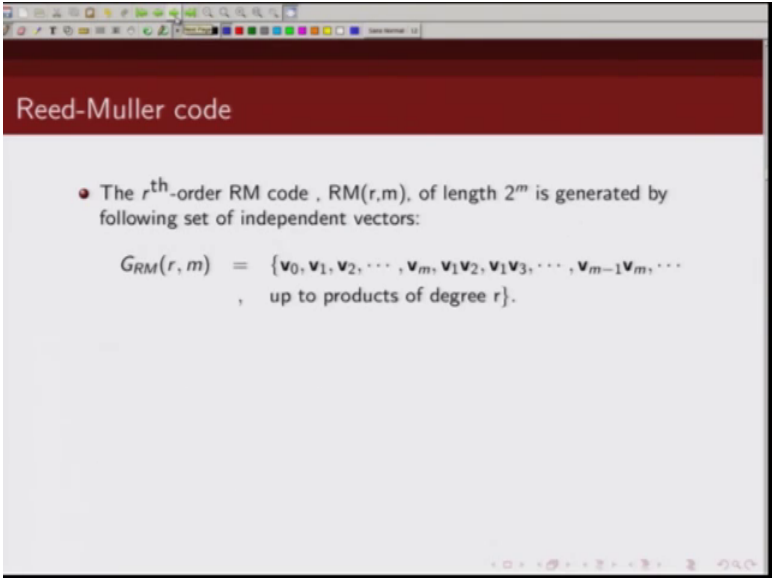
- The weight of the product

$$\mathbf{v}_{i_1}\mathbf{v}_{i_2}\cdots\mathbf{v}_{i_l}$$

is equal to 2^{m-l}.

product is given by two raised to power m minus l.

(Refer Slide Time 08:38)



Reed-Muller code

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

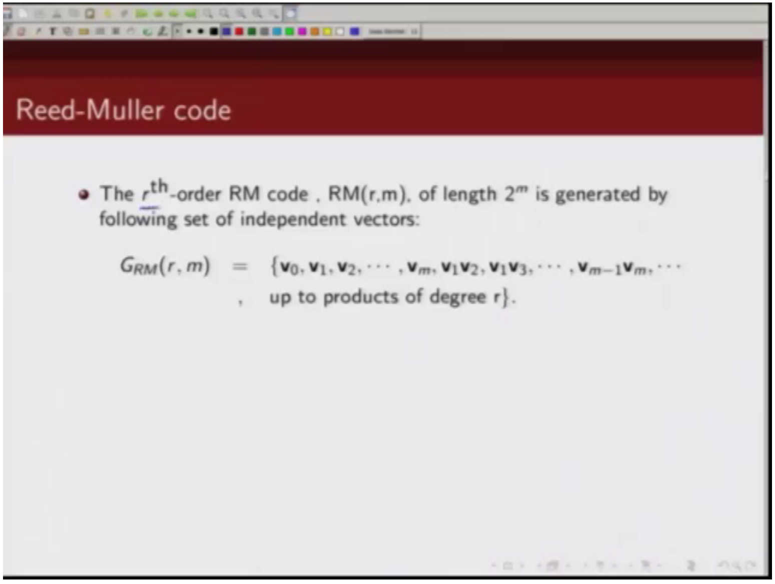
$$\begin{aligned} G_{RM}(r,m) &= \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m, \mathbf{v}_1\mathbf{v}_2, \mathbf{v}_1\mathbf{v}_3, \cdots, \mathbf{v}_{m-1}\mathbf{v}_m, \cdots \\ &, \text{ up to products of degree r}\}. \end{aligned}$$

So now that we have

(Refer Slide Time 08:40)

defined these tuples v i's and the Boolean product between them we are ready to define the generator matrix for Reed-Muller code. So an rth order Reed-Muller

(Refer Slide Time 08:57)



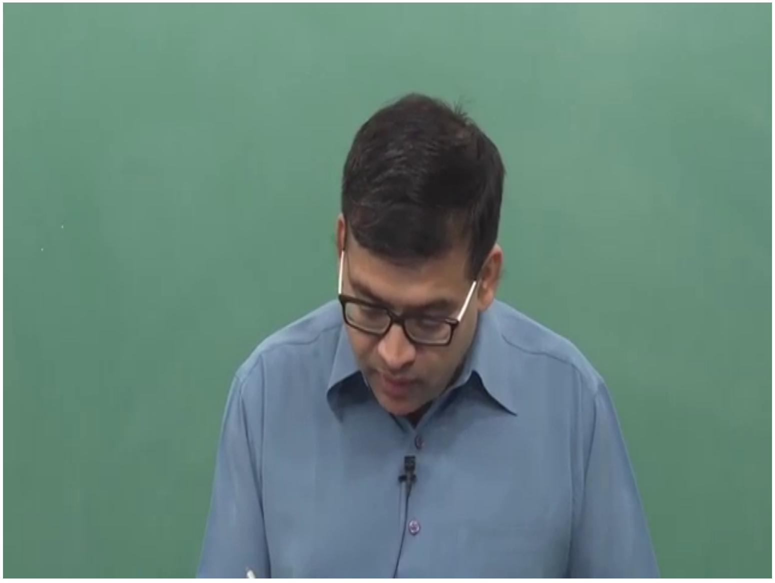
Reed-Muller code

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

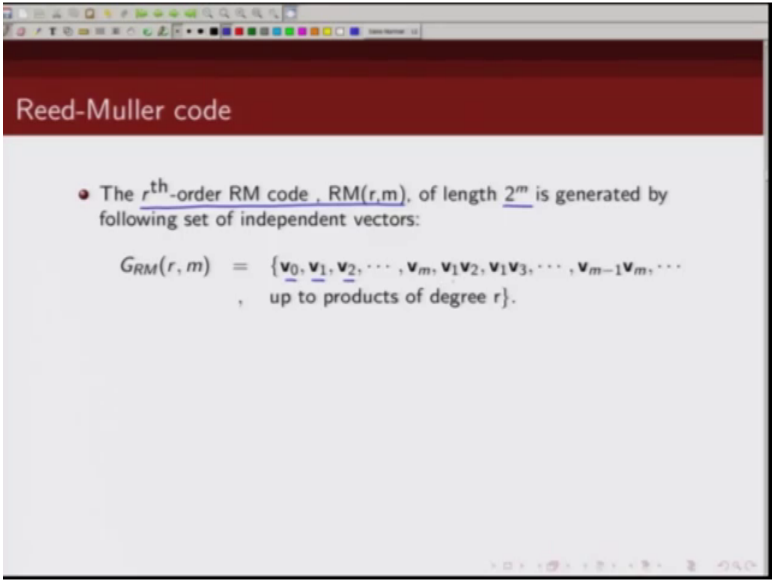
$$G_{RM}(r,m) = \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m, \mathbf{v}_1\mathbf{v}_2, \mathbf{v}_1\mathbf{v}_3, \cdots, \mathbf{v}_{m-1}\mathbf{v}_m, \cdots , \text{ up to products of degree r}\}.$$

code which is of length 2 raised to power m can be generated by the set of independent vectors where these vectors are v 0, v 1, v 2 then Boolean product of

(Refer Slide Time 09:14)



(Refer Slide Time 09:15)



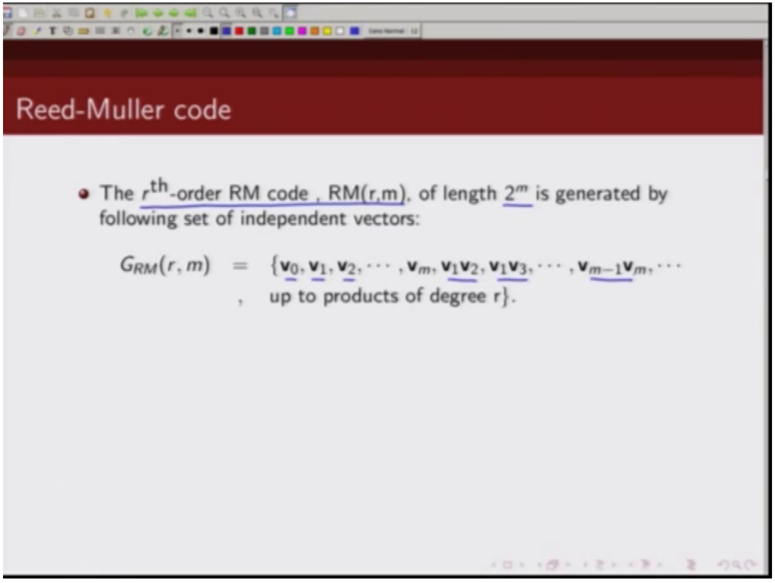
Reed-Muller code

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

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

second order which is v 1 v 2, v 1 v 3 these are all second order products then we will have third order products, fourth order products depending

(Refer Slide Time 09:24)



Reed-Muller code

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

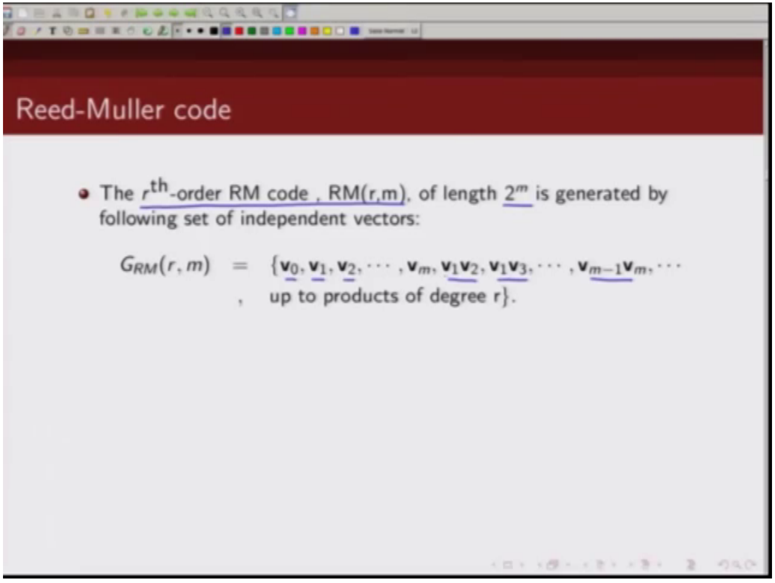
$$G_{RM}(r,m) = \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m, \mathbf{v}_1\mathbf{v}_2, \mathbf{v}_1\mathbf{v}_3, \cdots, \mathbf{v}_{m-1}\mathbf{v}_m, \cdots, \text{up to products of degree r}\}.$$

on what the r is. So we generate Reed-Muller code

(Refer Slide Time 09:29)

using these

(Refer Slide Time 09:31)



Reed-Muller code

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

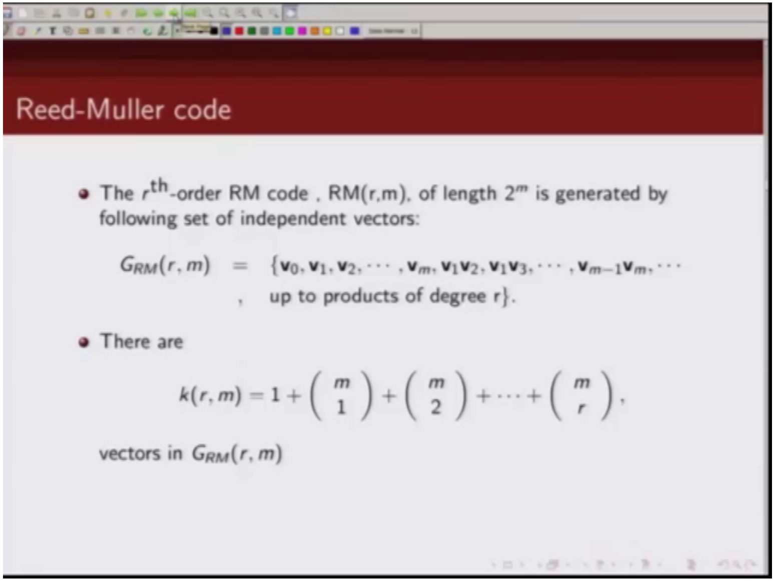
$$G_{RM}(r,m) = \{\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_m, \mathbf{v}_1\mathbf{v}_2, \mathbf{v}_1\mathbf{v}_3, \cdots, \mathbf{v}_{m-1}\mathbf{v}_m, \cdots, \text{ up to products of degree r}\}.$$

2-m tuples basically these v 0, v 1, v 2 and their Boolean product. And as

(Refer Slide Time 09:39)

you can see that v 0

(Refer Slide Time 09:42)



Reed-Muller code

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

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

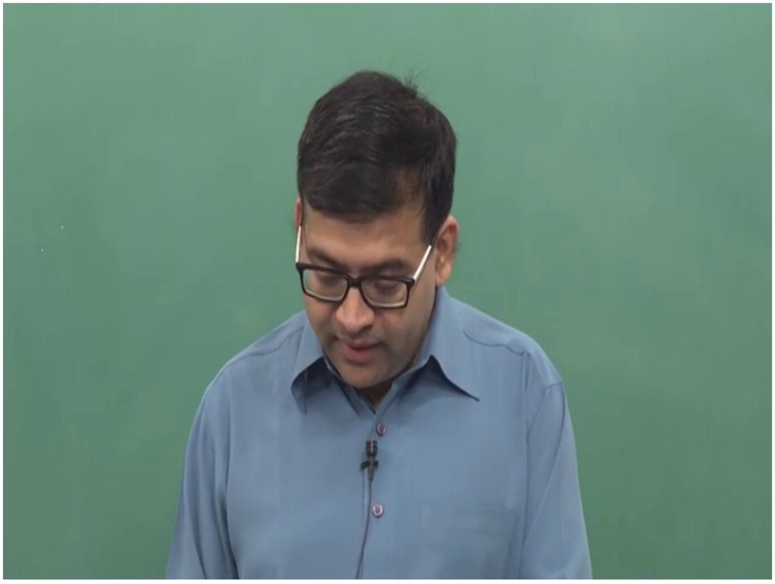
- There are

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

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

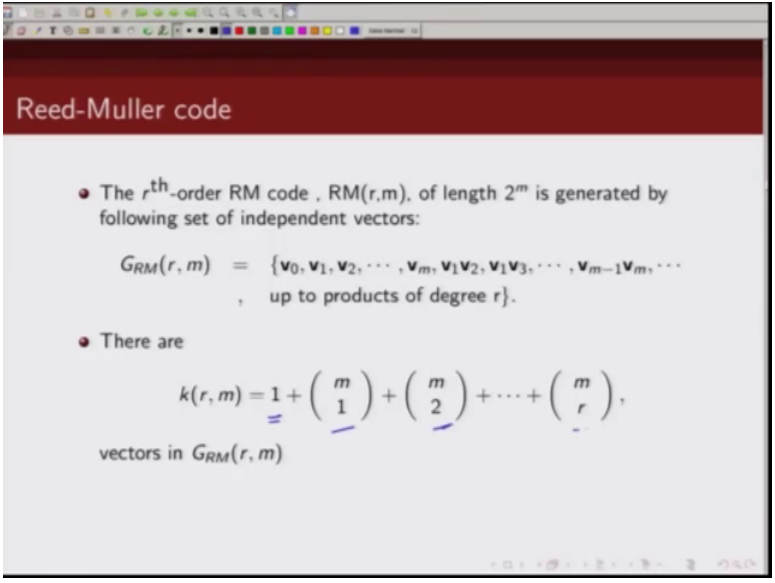
is all 1 sequence, so there is one such possible ways we can get this; v 1, this m C 1 of choosing v 1; m C 2 ways of, so v 1, v 2, v 3, v m this is basically m choose 1, then Boolean product of degree 2 can be chosen m choose 2 way

(Refer Slide Time 10:10)



and similarly

(Refer Slide Time 10:12)



Reed-Muller code

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

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

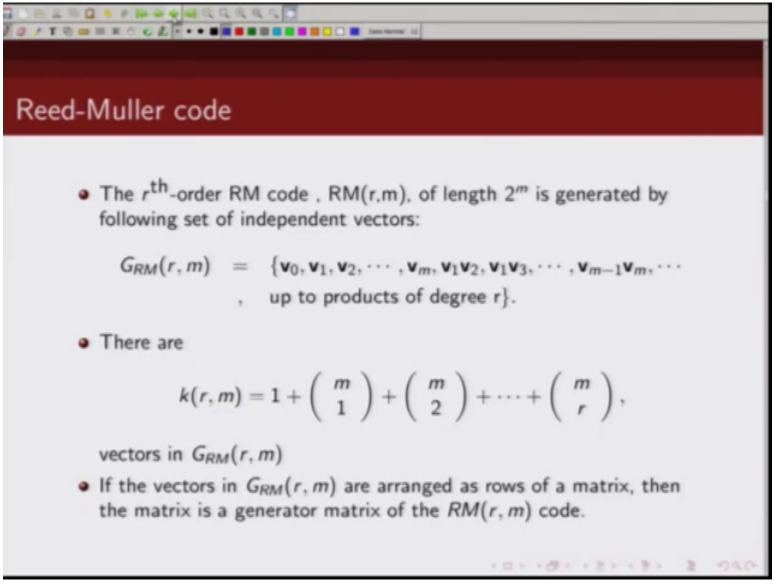
- There are

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

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

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

(Refer Slide Time 10:25)



Reed-Muller code

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

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

- There are

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

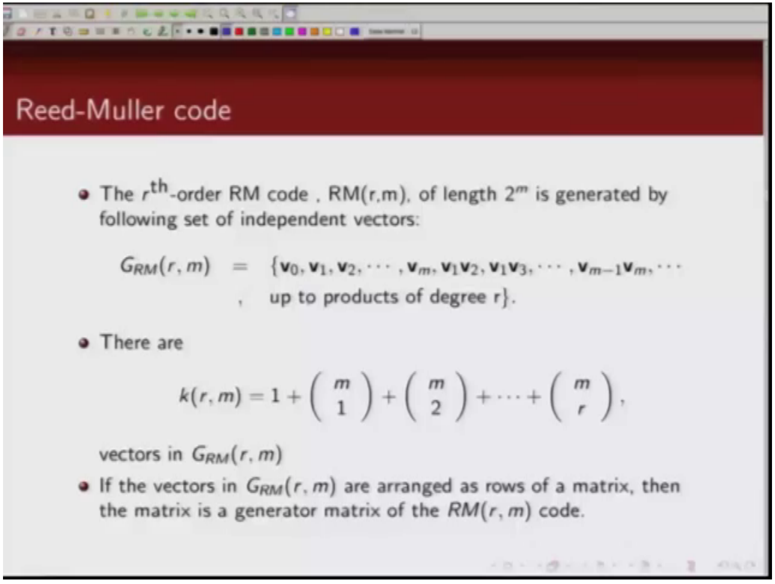
vectors in $G_{RM}(r,m)$
- If the vectors in $G_{RM}(r,m)$ are arranged as rows of a matrix, then the matrix is a generator matrix of the $RM(r,m)$ code.

. Now if we arrange these vectors v 0, v 1, v 2 and their Boolean product up to order r

(Refer Slide Time 10:32)

as rows of a matrix, that will be our generator matrix for Reed-Muller code and each of v 0, v 1 and their

(Refer Slide Time 10:42)



Reed-Muller code

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

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

- There are

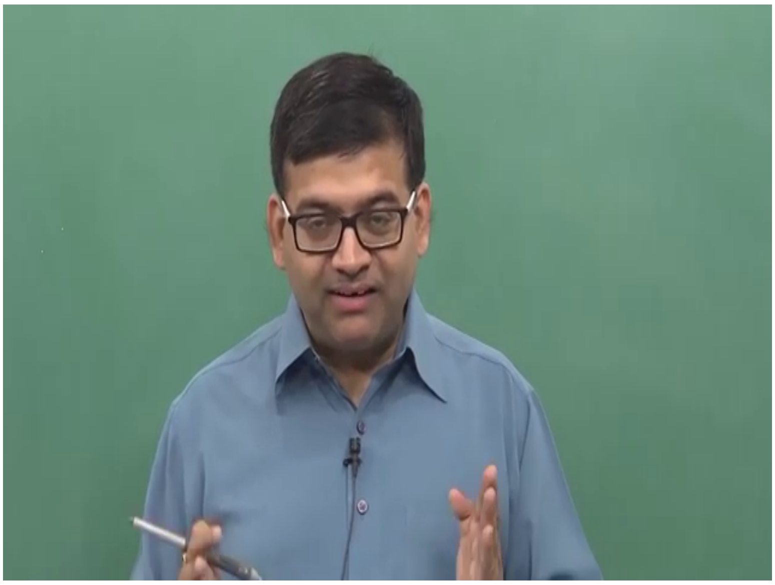
$$k(r,m) = 1 + \binom{m}{1} + \binom{m}{2} + \cdots + \binom{m}{r},$$

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.

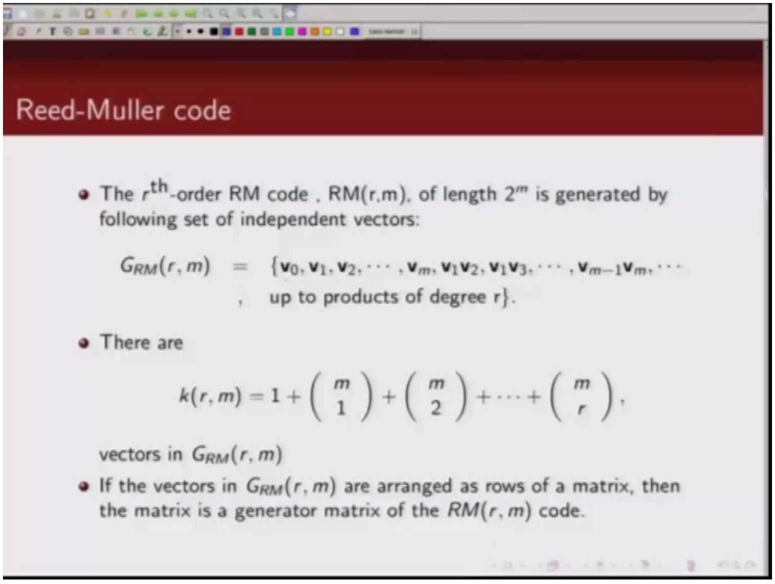
Boolean product they are basically linearly independent, so we can generate our

(Refer Slide Time 10:48)



Reed-Muller code using these

(Refer Slide Time 10:52)



Reed-Muller code

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

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

- There are

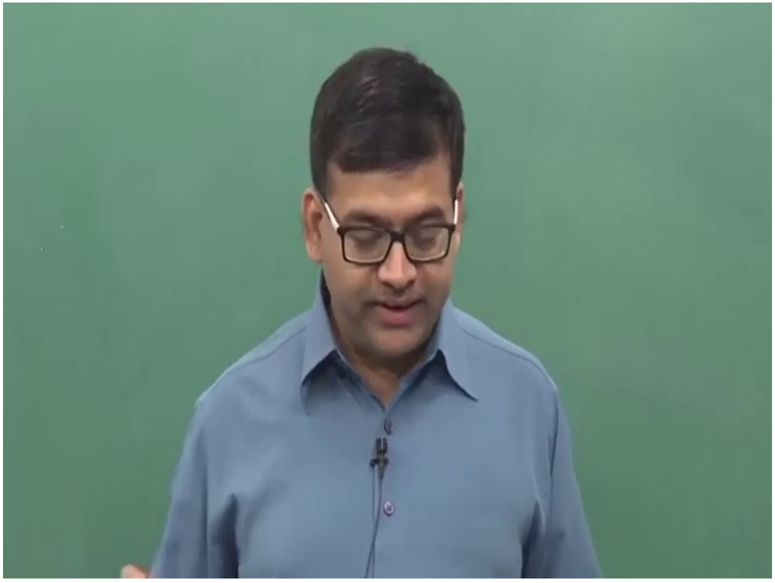
$$k(r,m) = 1 + \binom{m}{1} + \binom{m}{2} + \cdots + \binom{m}{r},$$

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.

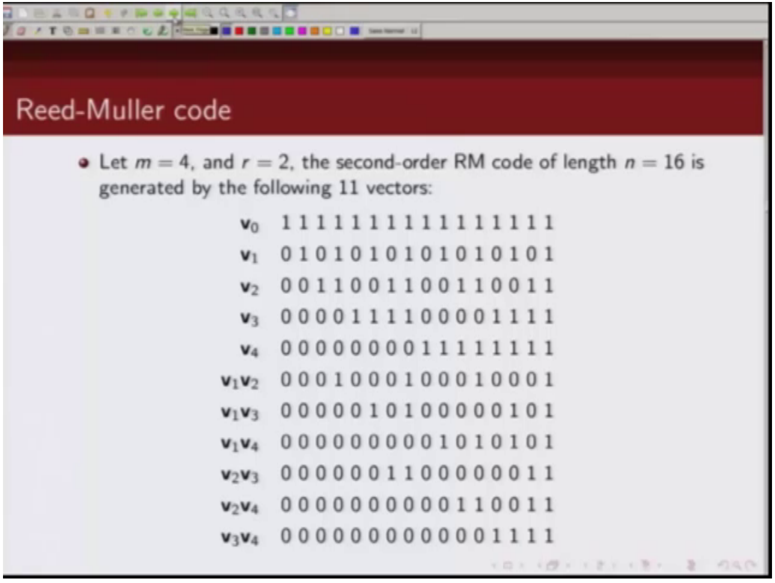
v 0, v i and their Boolean product as rows of a generator matrix.

(Refer Slide Time 10:58)



So let us

(Refer Slide Time 11:00)



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:

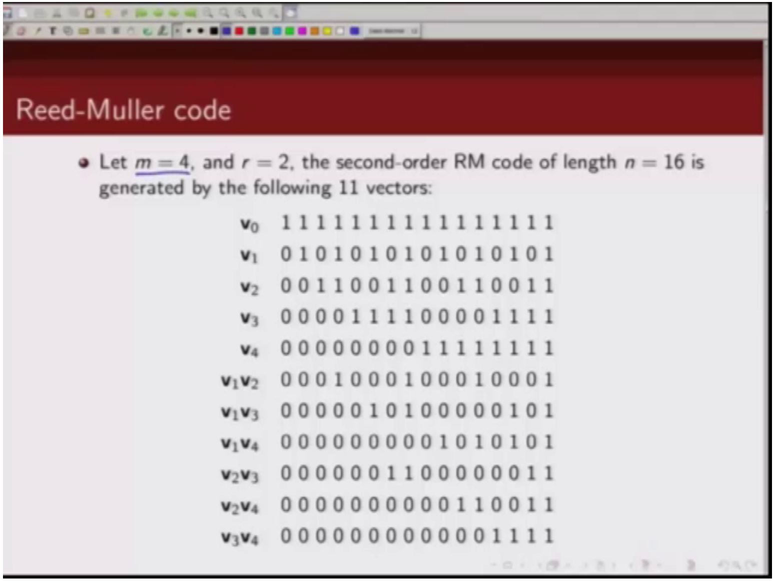
$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

illustrate this with an example.

(Refer Slide Time 11:03)

We take a case where m is 4.

(Refer Slide Time 11:06)



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:

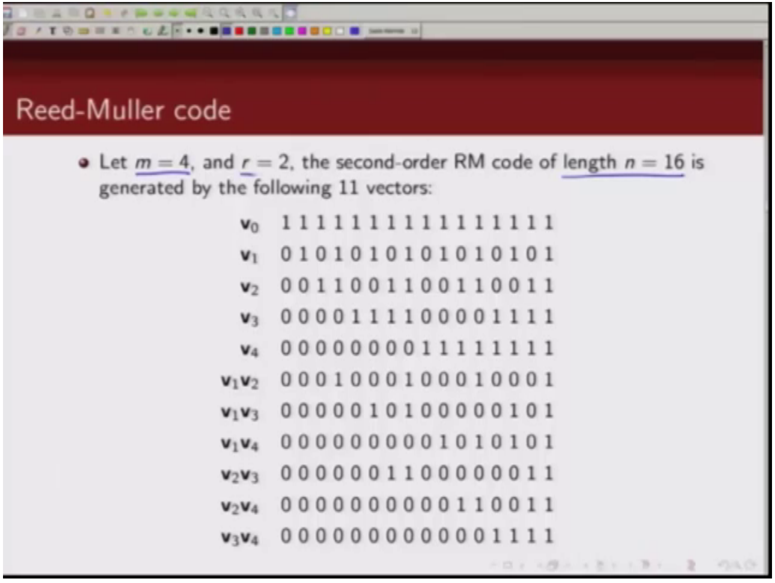
$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

So m is 4 meaning our codeword length would be 2 raised to power m which is 16. So we are dealing with Reed-Muller code of length 16.

(Refer Slide Time 11:17)

Let us consider second order

(Refer Slide Time 11:19)



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:

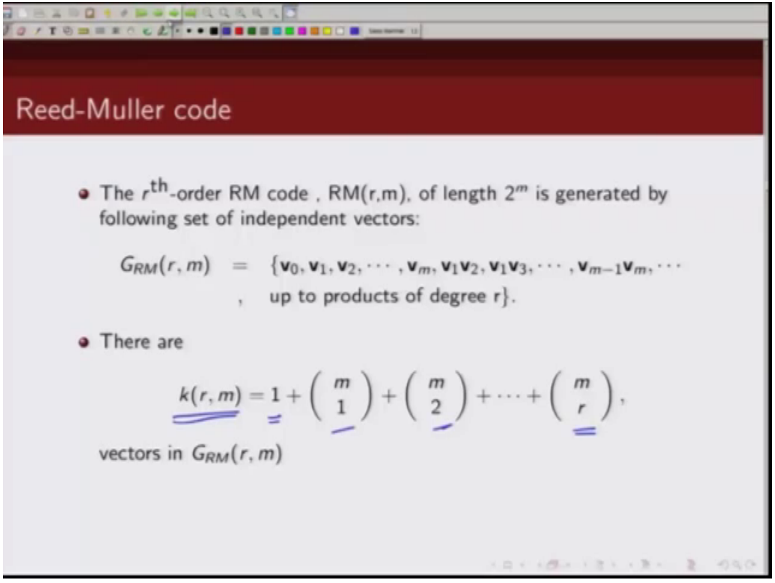
$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

Reed-Muller code. So we will have to, now recall what is a degree. If you go back, this product vector is said to have degree l if there are l such v i's which are participating

(Refer Slide Time 11:36)

in this Boolean product. So we have to

(Refer Slide Time 11:39)



Reed-Muller code

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

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

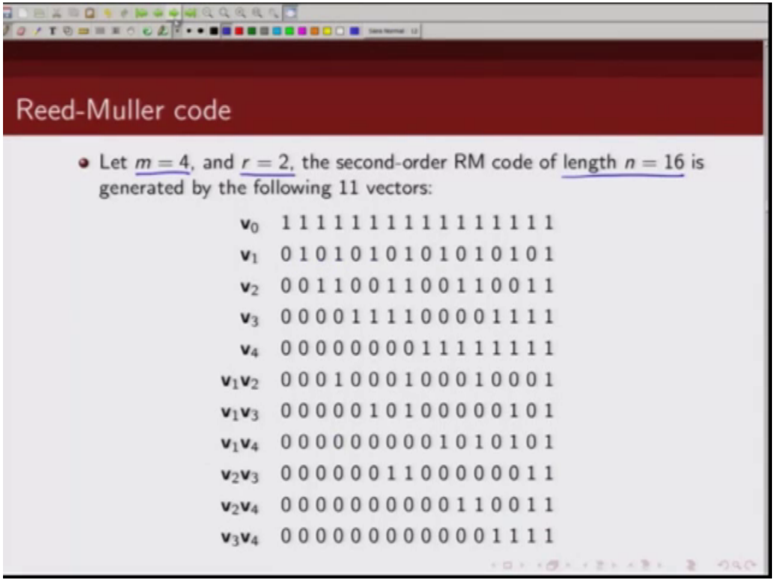
- There are

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

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

write all these as rows of generator matrix up to product of degree r.

(Refer Slide Time 11:50)



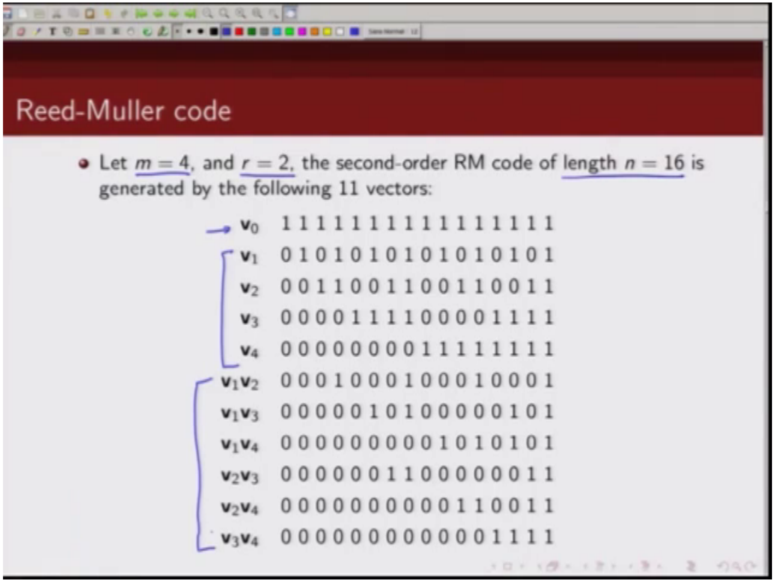
Reed-Muller code

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

$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

So this is 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

(Refer Slide Time 12:07)



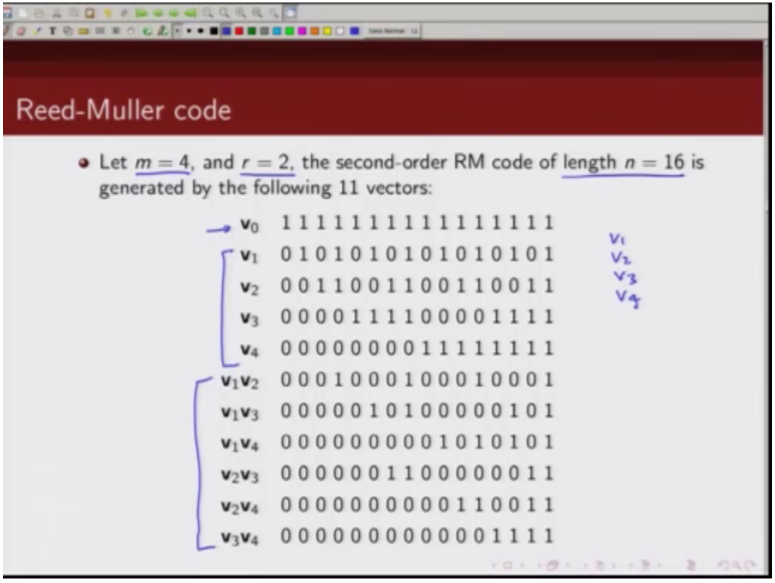
Reed-Muller code

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

$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

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

(Refer Slide Time 12:19)



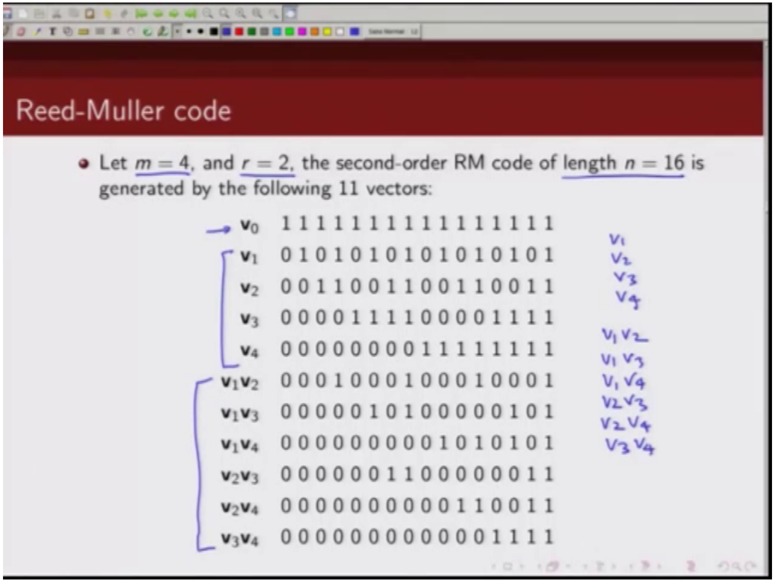
Reed-Muller code

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

$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

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.

(Refer Slide Time 12:37)



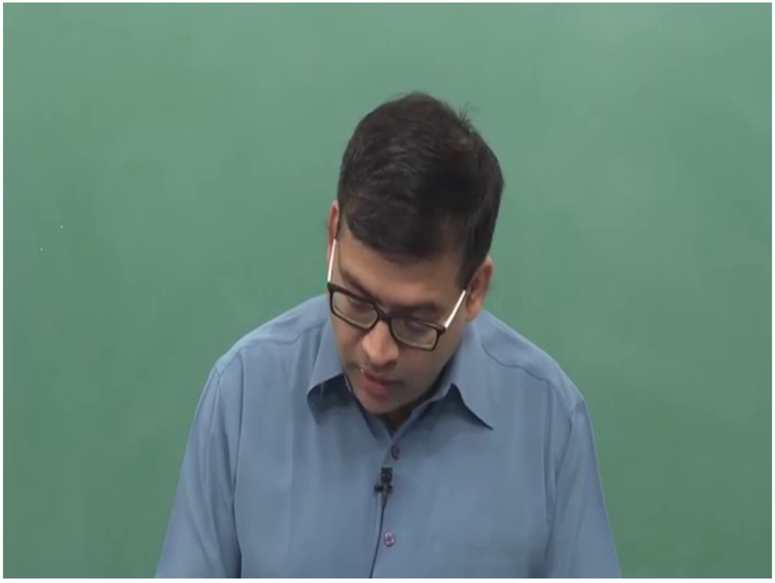
Reed-Muller code

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

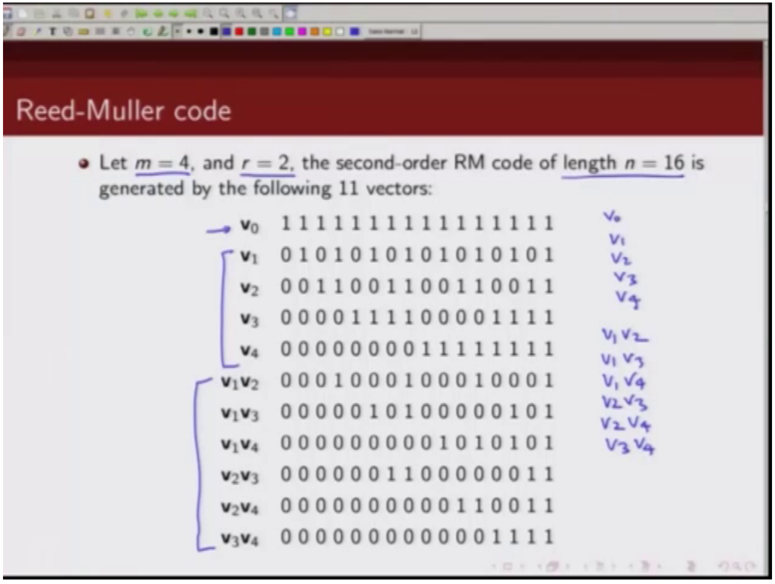
$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

And of course you have your

(Refer Slide Time 12:40)



(Refer Slide Time 12:41)



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:

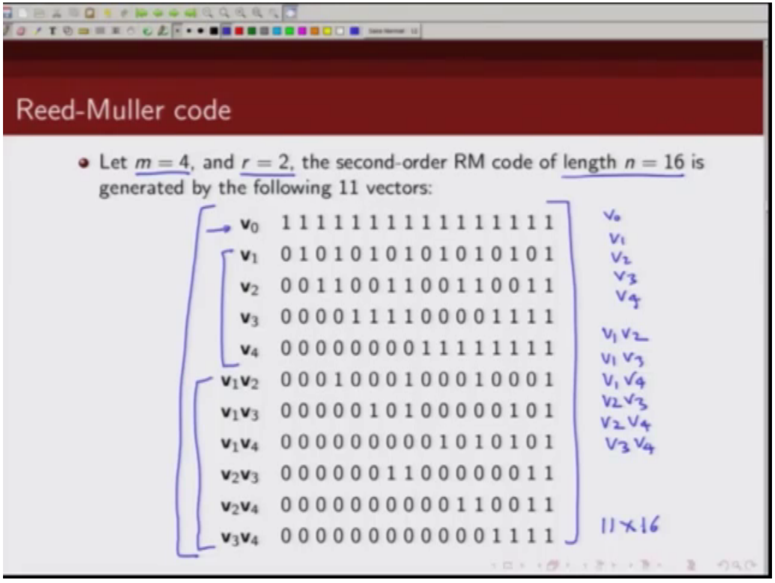
$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

v_0
v_1
v_2
v_3
v_4

$v_1 v_2$
$v_1 v_3$
$v_1 v_4$
$v_2 v_3$
$v_2 v_4$
$v_3 v_4$

all 1 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 cross 16 generator matrix.

(Refer Slide Time 13:00)



Reed-Muller code

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

$$\begin{array}{c|cccccccccccccccc}
\mathbf{v}_0 & 1&1&1&1&1&1&1&1&1&1&1&1&1&1&1&1 \\
\mathbf{v}_1 & 0&1&0&1&0&1&0&1&0&1&0&1&0&1&0&1 \\
\mathbf{v}_2 & 0&0&1&1&0&0&1&1&0&0&1&1&0&0&1&1 \\
\mathbf{v}_3 & 0&0&0&0&1&1&1&1&0&0&0&0&1&1&1&1 \\
\mathbf{v}_4 & 0&0&0&0&0&0&0&0&1&1&1&1&1&1&1&1 \\
\mathbf{v}_1\mathbf{v}_2 & 0&0&0&1&0&0&0&1&0&0&0&1&0&0&0&1 \\
\mathbf{v}_1\mathbf{v}_3 & 0&0&0&0&0&1&0&1&0&0&0&0&0&1&0&1 \\
\mathbf{v}_1\mathbf{v}_4 & 0&0&0&0&0&0&0&0&0&1&0&1&0&1&0&1 \\
\mathbf{v}_2\mathbf{v}_3 & 0&0&0&0&0&0&1&1&0&0&0&0&0&0&1&1 \\
\mathbf{v}_2\mathbf{v}_4 & 0&0&0&0&0&0&0&0&0&0&1&1&0&0&1&1 \\
\mathbf{v}_3\mathbf{v}_4 & 0&0&0&0&0&0&0&0&0&0&0&0&1&1&1&1
\end{array}$$

v_0
v_1
v_2
v_3
v_4
$v_1 v_2$
$v_1 v_3$
$v_1 v_4$
$v_2 v_3$
$v_2 v_4$
$v_3 v_4$

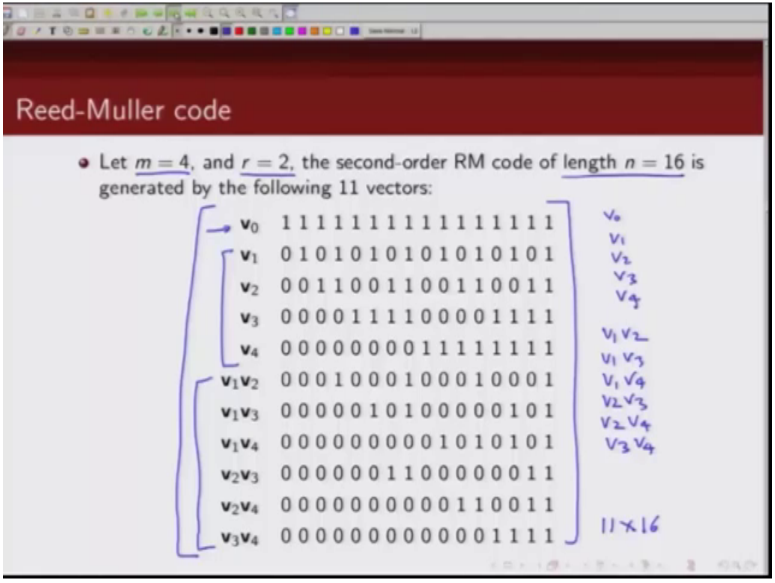
11×16

Ok and we will use this

(Refer Slide Time 13:02)

to generate our set of codewords.

(Refer Slide Time 13:06)



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:

$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

v_0
v_1
v_2
v_3
v_4
$v_1 v_2$
$v_1 v_3$
$v_1 v_4$
$v_2 v_3$
$v_2 v_4$
$v_3 v_4$

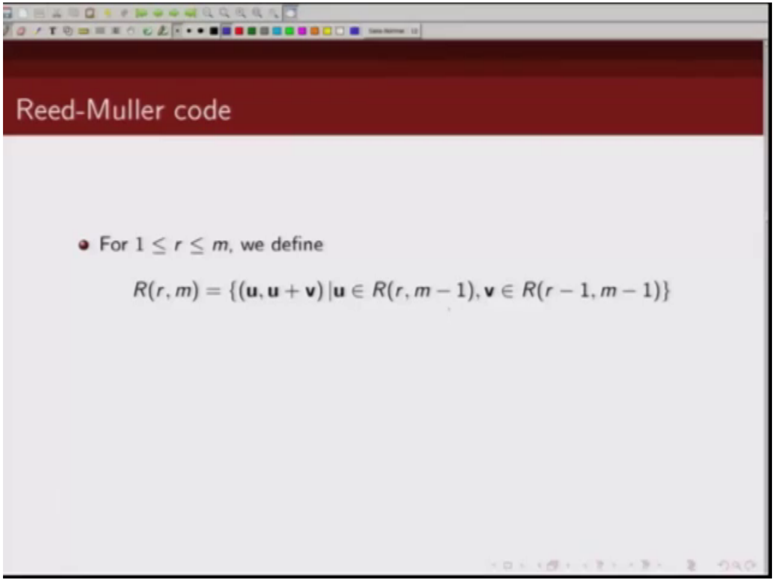
11×16

Now this another

(Refer Slide Time 13:08)

alternative construction of Reed-Muller code, so if you are given

(Refer Slide Time 13:13)



Reed-Muller code

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

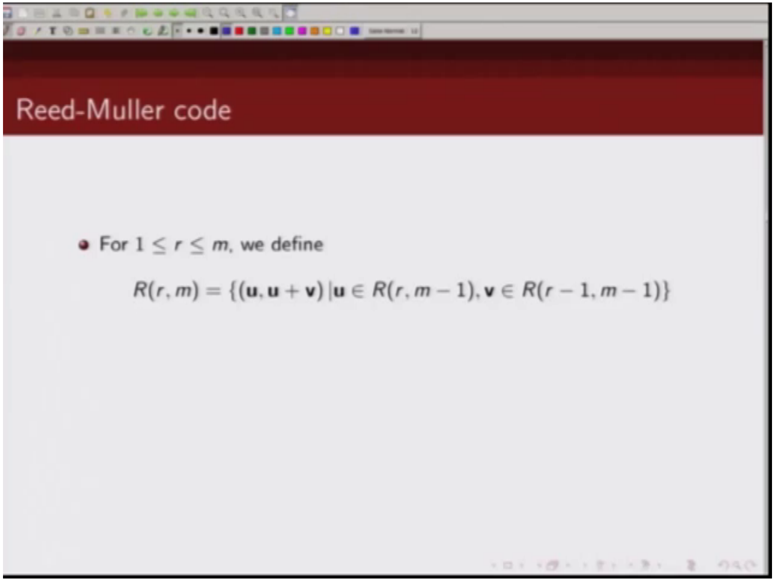
$$R(r,m) = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \,|\, \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$

Reed-Muller code of length 2 raised to power m minus 1, then you can use two of them to

(Refer Slide Time 13:26)

construct a Reed-Muller code of

(Refer Slide Time 13:29)



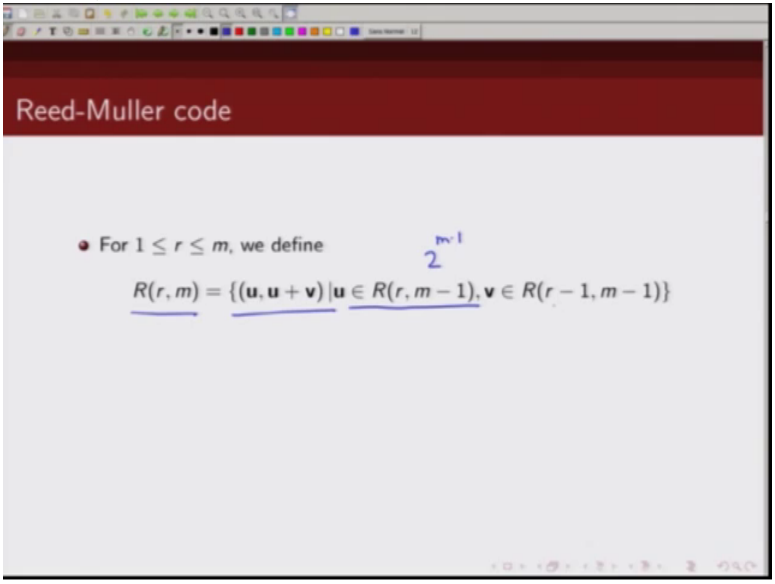
Reed-Muller code

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

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

length 2 raised to power 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 raised to power m minus 1, and you have another

(Refer Slide Time 13:50)



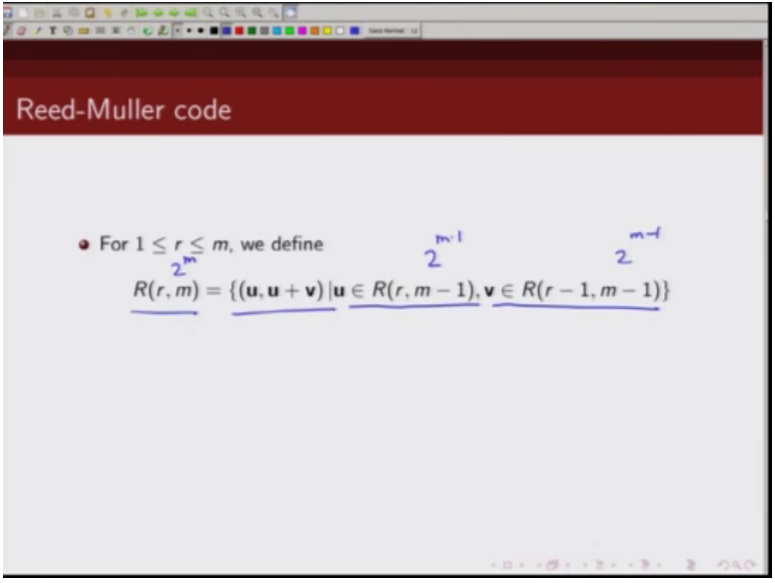
Reed-Muller code

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

$$\underline{R(r,m)} = \{\underline{(\mathbf{u}, \mathbf{u}+\mathbf{v})} \,|\, \mathbf{u} \in \underline{\overset{2^{m-1}}{R(r,m-1)}}, \mathbf{v} \in R(r-1,m-1)\}$$

Reed-Muller code of order r minus 1, and length 2 minus 1, then these two can be used to construct a Reed-Muller code of order r and length 2 raised to power m,

(Refer Slide Time 14:03)



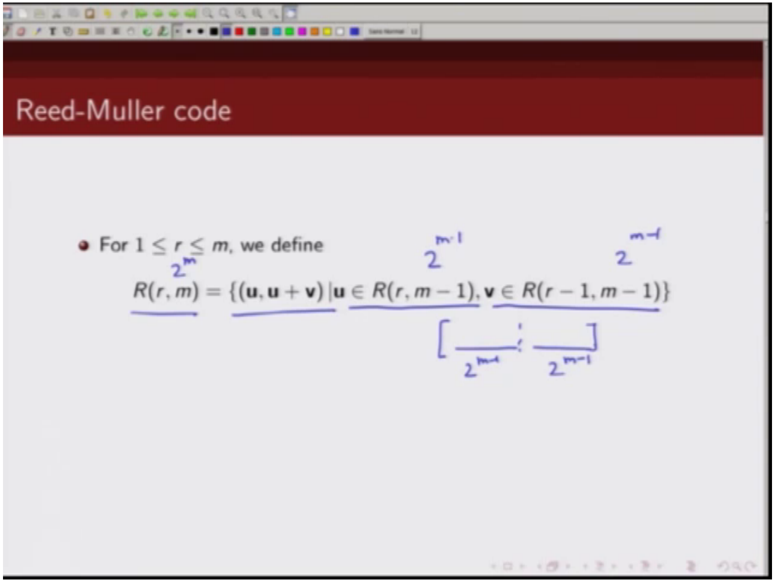
Reed-Muller code

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

$$\underline{\overset{2^m}{R(r,m)}} = \underline{\{(\mathbf{u},\mathbf{u}+\mathbf{v})}\,|\,\mathbf{u} \in \underline{\overset{2^{m-1}}{R(r,m-1)}}, \mathbf{v} \in \underline{\overset{2^{m-1}}{R(r-1,m-1)}}\}$$

and in this particular way. So first, so you can, so if this is, this is one code of length 2 m minus 1 and some another code of length 2 m minus 1, this is

(Refer Slide Time 14:17)



Reed-Muller code

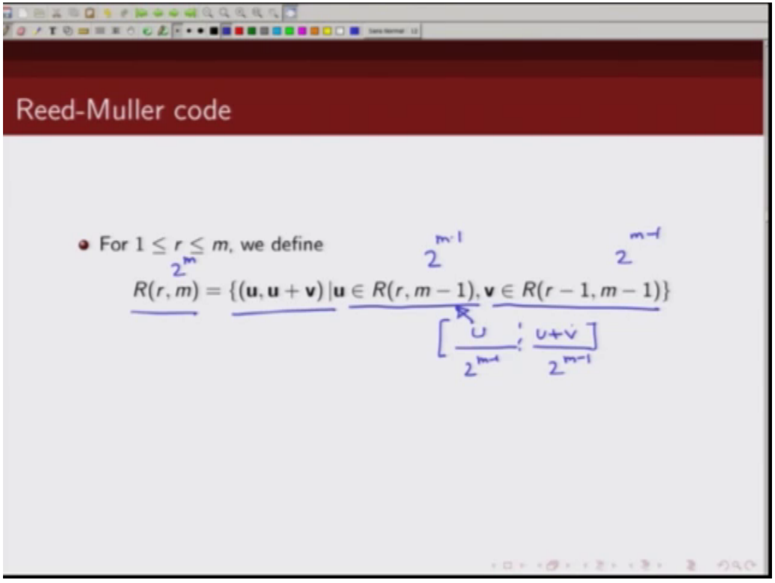
- For $1 \leq r \leq m$, we define

$$\underline{\overset{2^m}{R(r,m)}} = \underline{\{(\mathbf{u},\mathbf{u}+\mathbf{v})}\,|\,\mathbf{u} \in \underline{\overset{2^{m-1}}{R(r,m-1)}}, \mathbf{v} \in \underline{\overset{2^{m-1}}{R(r-1,m-1)}}\}$$

$$\left[\underbrace{\qquad}_{2^{m-1}} : \underbrace{\qquad}_{2^{m-1}}\right]$$

your code u which is order r and this is u plus v where u is

(Refer Slide Time 14:26)



Reed-Muller code

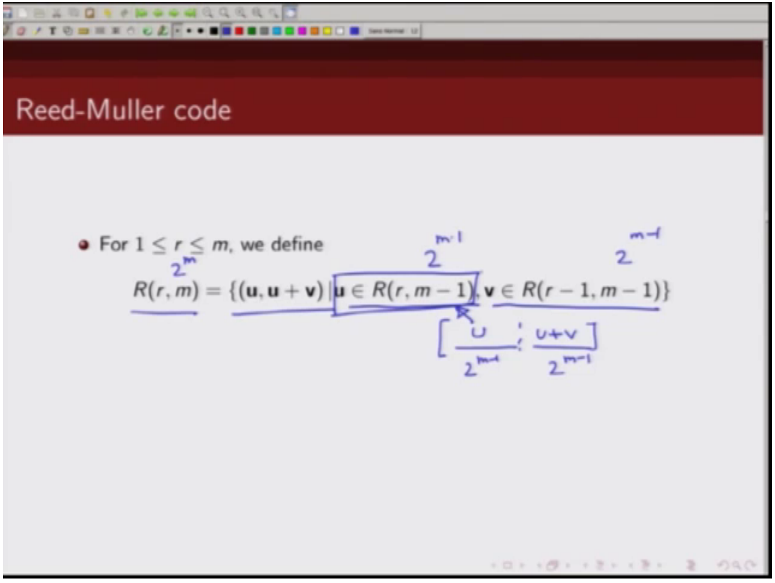
- For $1 \leq r \leq m$, we define

$$\underset{}{\overset{2^m}{R(r,m)}} = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \,|\, \mathbf{u} \in \overset{2^{m-1}}{R(r,m-1)}, \mathbf{v} \in \overset{2^{m-1}}{R(r-1,m-1)}\}$$

$$\left[\underbrace{u}_{2^{m-1}} \,\vdots\, \underbrace{u+v}_{2^{m-1}}\right]$$

given by this and v is

(Refer Slide Time 14:30)



Reed-Muller code

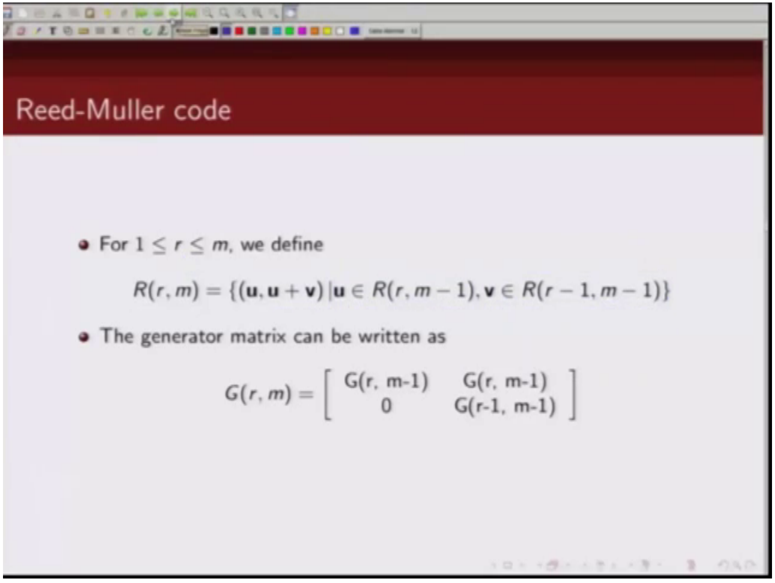
- For $1 \leq r \leq m$, we define

$$\overset{2^m}{R(r,m)} = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \,|\, \boxed{\mathbf{u} \in \overset{2^{m-1}}{R(r,m-1)}}, \mathbf{v} \in \overset{2^{m-1}}{R(r-1,m-1)}\}$$

$$\left[\underbrace{u}_{2^{m-1}} \,\vdots\, \underbrace{u+v}_{2^{m-1}}\right]$$

given by this. So in other words, you can construct Reed-Muller code recursively from smaller order and smaller length code. The same thing

(Refer Slide Time 14:43)



Reed-Muller code

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

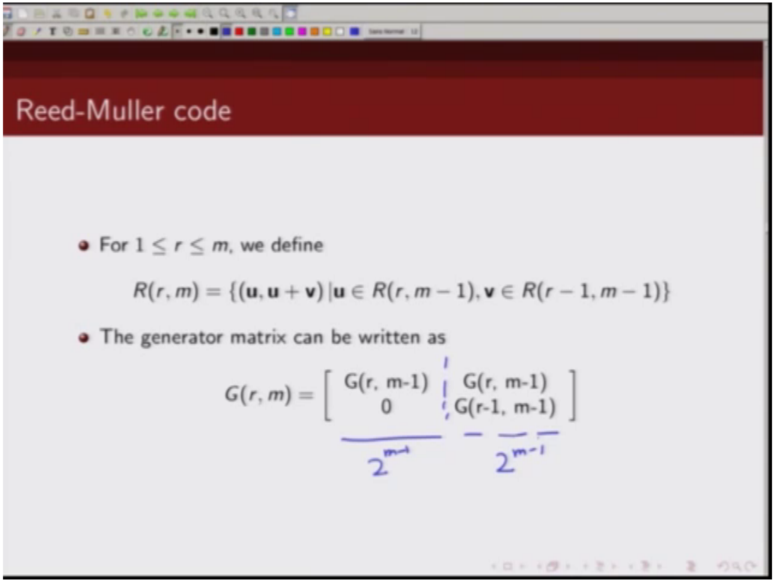
$$R(r, m) = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \,|\, \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$

- The generator matrix can be written as

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

I can, I am writing in terms of generator matrix. So as I said, this is a Reed-Muller code of length 2 m minus 1,

(Refer Slide Time 14:56)



Reed-Muller code

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

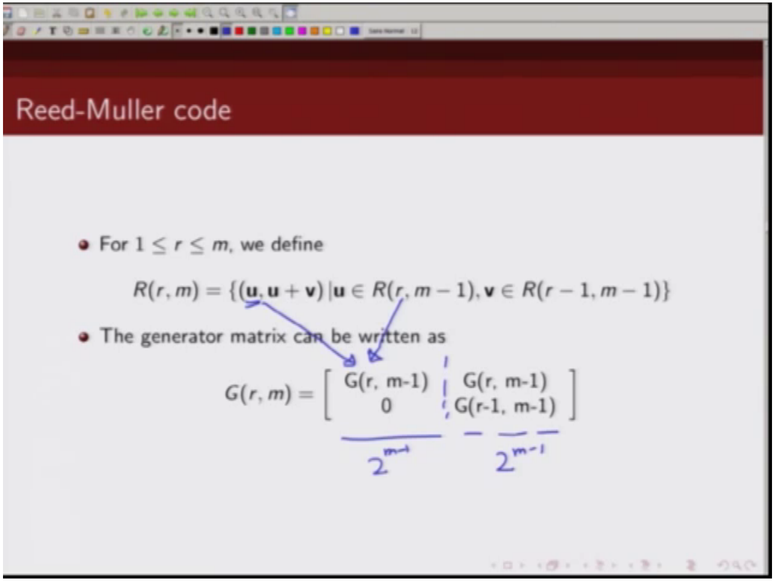
$$R(r, m) = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \,|\, \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$

- The generator matrix can be written as

$$G(r, m) = \left[\begin{array}{c|c} \underbrace{\begin{matrix} G(r, m\text{-}1) \\ 0 \end{matrix}}_{2^{m-1}} & \underbrace{\begin{matrix} G(r, m\text{-}1) \\ G(r\text{-}1, m\text{-}1) \end{matrix}}_{2^{m-1}} \end{array}\right]$$

this is another Reed-Muller code of length 2 m minus 1, first is just u which is this, this code, Reed-Muller code

(Refer Slide Time 15:03)



Reed-Muller code

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

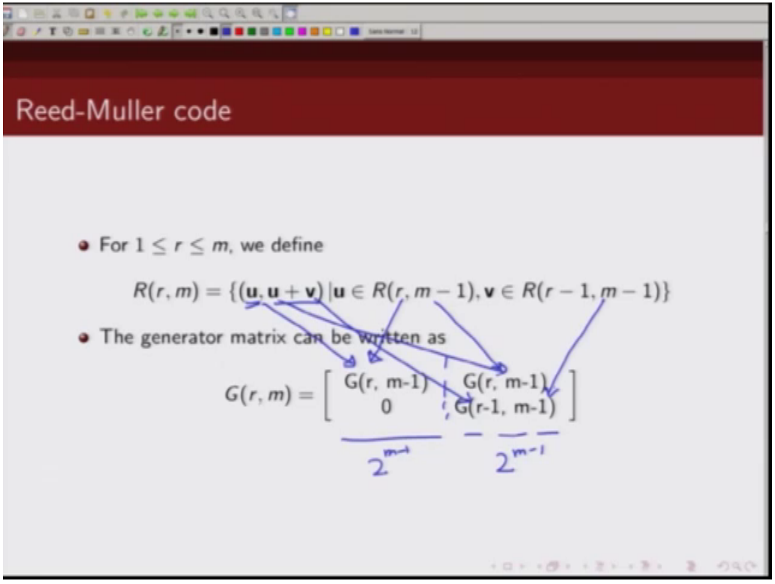
$$R(r,m) = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \,|\, \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$

- The generator matrix can be written as

$$G(r,m) = \begin{array}{c} \left[\begin{array}{c|c} \text{G(r, m-1)} & \text{G(r, m-1)} \\ 0 & \text{G(r-1, m-1)} \end{array}\right] \\ \begin{array}{cc} \underline{\quad 2^{m-1} \quad} & \underline{\quad 2^{m-1} \quad} \end{array} \end{array}$$

order r length 2 raised to power m minus 1. And the second one is this, so this is your u which is this and the next one, this is your v which is this. So I can write down,

(Refer Slide Time 15:24)



Reed-Muller code

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

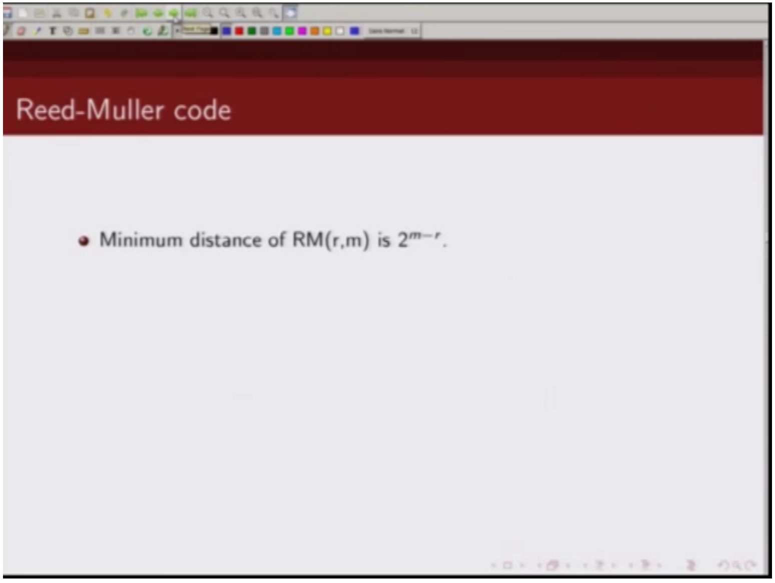
$$R(r,m) = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \,|\, \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$

- The generator matrix can be written as

$$G(r,m) = \begin{array}{c} \left[\begin{array}{c|c} \text{G(r, m-1)} & \text{G(r, m-1)} \\ 0 & \text{G(r-1, m-1)} \end{array}\right] \\ \begin{array}{cc} \underline{\quad 2^{m-1} \quad} & \underline{\quad 2^{m-1} \quad} \end{array} \end{array}$$

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 15:40)

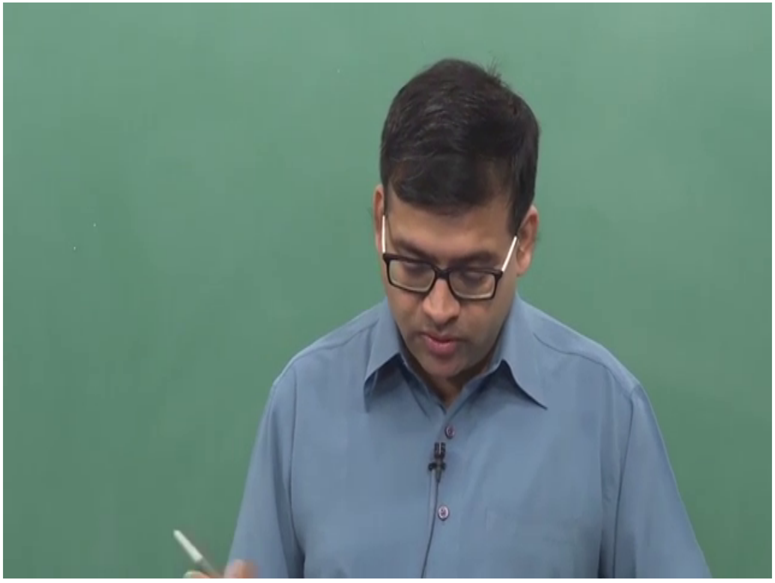


Reed-Muller code

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

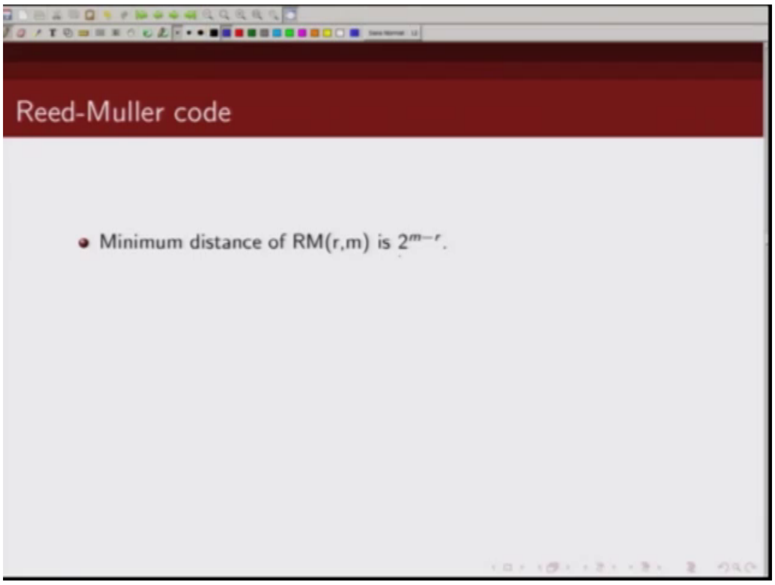
So let us prove some of the properties of Reed-Muller code.

(Refer Slide Time 15:44)



The first property that we are going to

(Refer Slide Time 15:46)



Reed-Muller code

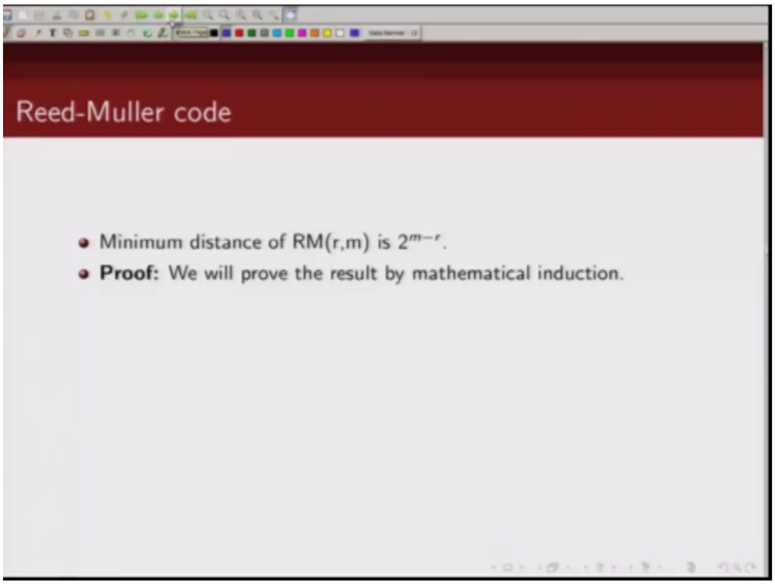
- Minimum distance of RM(r,m) is 2^{m-r}.

prove is that minimum distance of Reed-Muller code is 2 raised to power

(Refer Slide Time 15:52)

m minus r. We are going to prove this result

(Refer Slide Time 15:57)

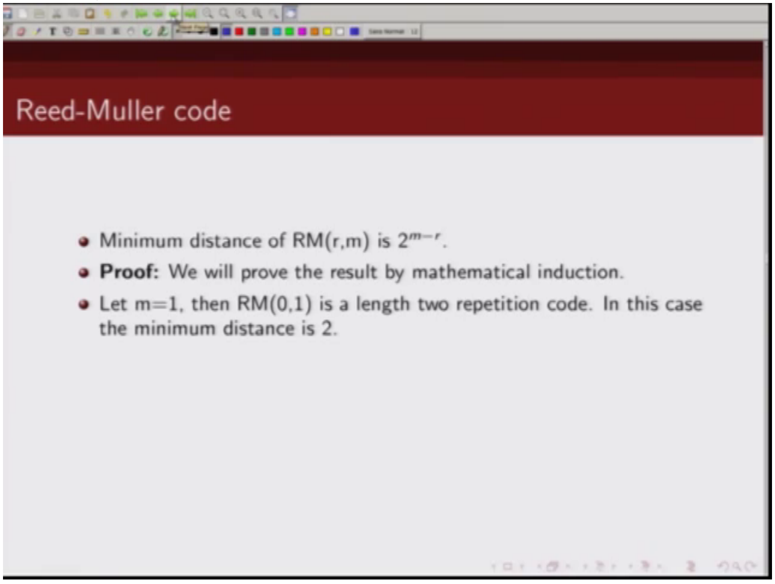


Reed-Muller code

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

using mathematical induction. So how does this work?

(Refer Slide Time 16:02)



Reed-Muller code

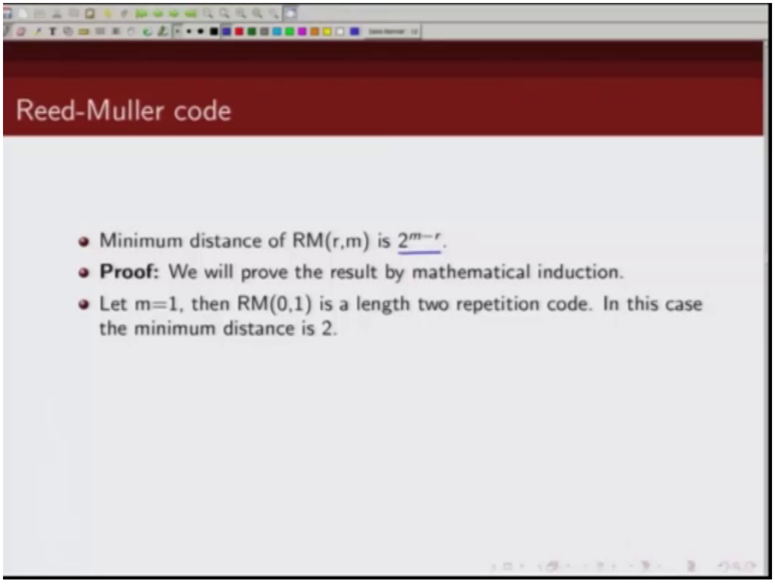
- Minimum distance of RM(r,m) is 2^{m-r}.
- **Proof:** We will prove the result by mathematical induction.
- Let m=1, then RM(0,1) is a length two repetition code. In this case the minimum distance is 2.

So first we assume m to be 1. And let's check whether this minimum distance holds correct for m equal to 1. So for m equal to 1, let us consider 2 scenarios; one where r is zero and in second case r is 1. So when m is one, what is the length of Reed-Muller code?

(Refer Slide Time 16:29)

It is 2 raised to power m. So that's length is 2,

(Refer Slide Time 16:32)



Reed-Muller code

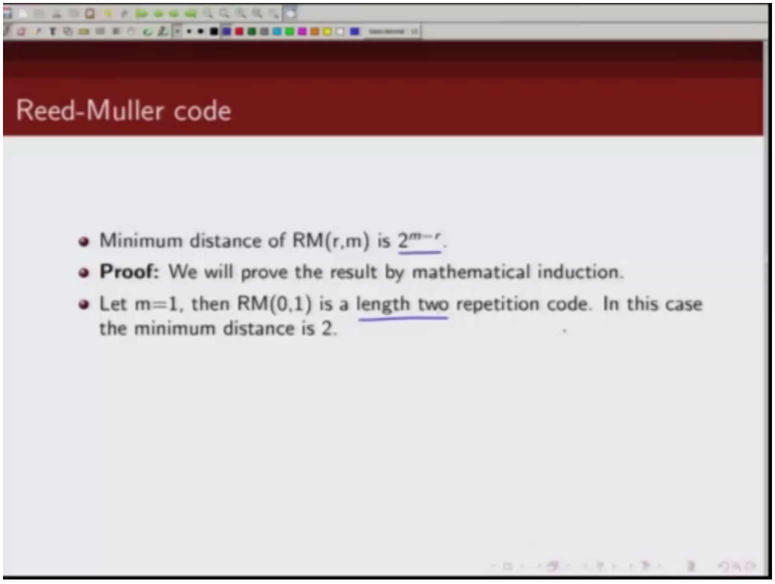
- Minimum distance of RM(r,m) is $\underline{2^{m-r}}$.
- **Proof:** We will prove the result by mathematical induction.
- Let m=1, then RM(0,1) is a length two repetition code. In this case the minimum distance is 2.

Ok and when order is zero

(Refer Slide Time 16:36)

so g will consist of only

(Refer Slide Time 16:38)

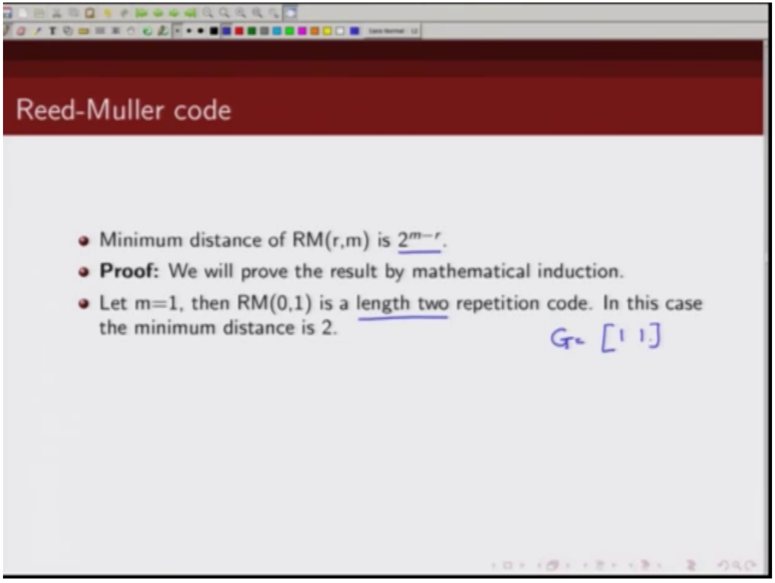


Reed-Muller code

- 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.

only v 0 which is 1 1. So

(Refer Slide Time 16:44)



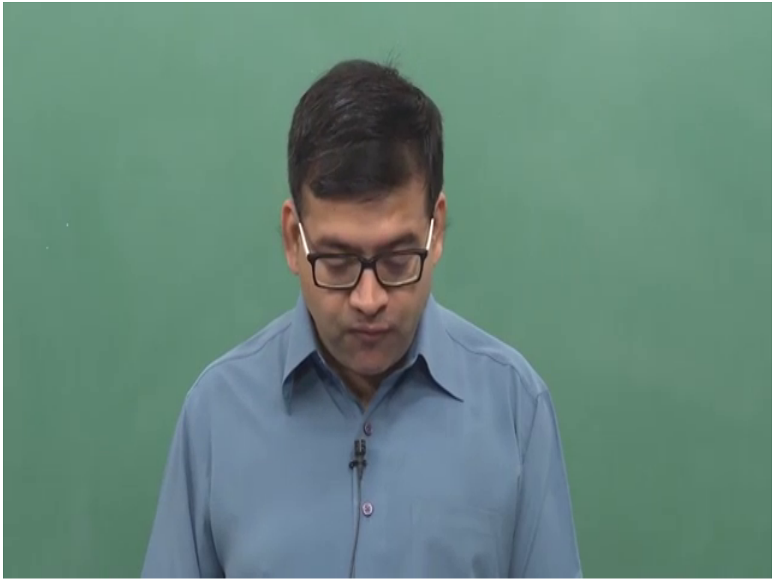
Reed-Muller code

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

$G = [1\ 1]$

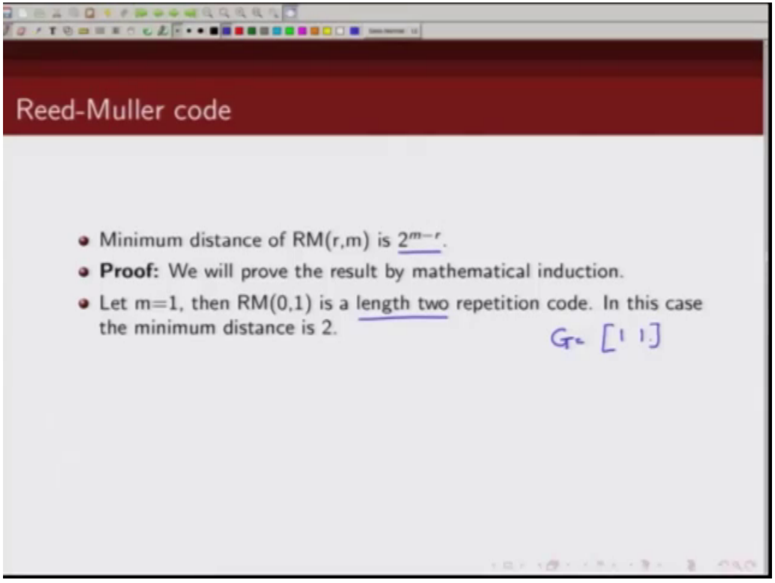
Reed-Muller code of

(Refer Slide Time 16:47)



order 0 and m 1

(Refer Slide Time 16:51)



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.

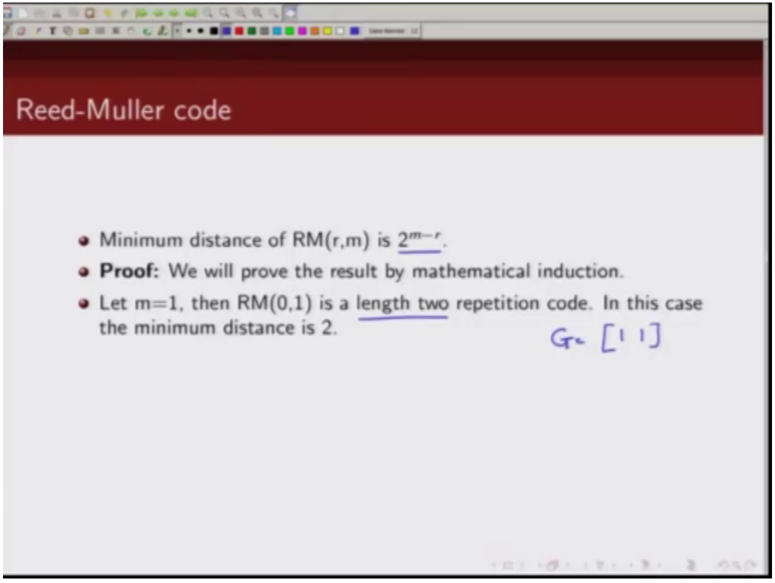
$G = [1\ 1]$

is essentially a length 2 repetition code and what is the minimum distance of this code?

(Refer Slide Time 16:57)

It's 2. So let's plug that in here and

(Refer Slide Time 17:01)



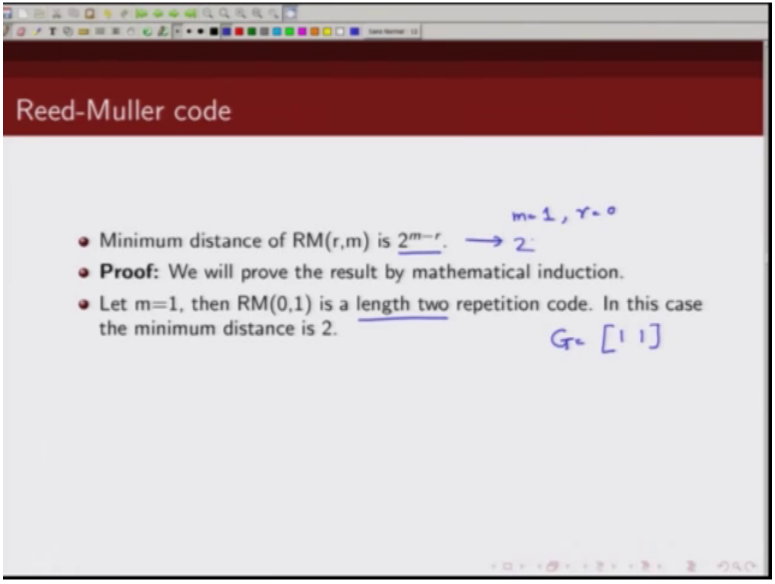
Reed-Muller code

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

$G = [1\ 1]$

see if it's correct. m in our case is 1, and r is zero. So this gives us minimum distance of 2. And that's precisely what we are getting.

(Refer Slide Time 17:12)



Reed-Muller code

- Minimum distance of RM(r,m) is $\underline{2^{m-r}}$. $\longrightarrow 2$ $m=1, r=0$
- **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.

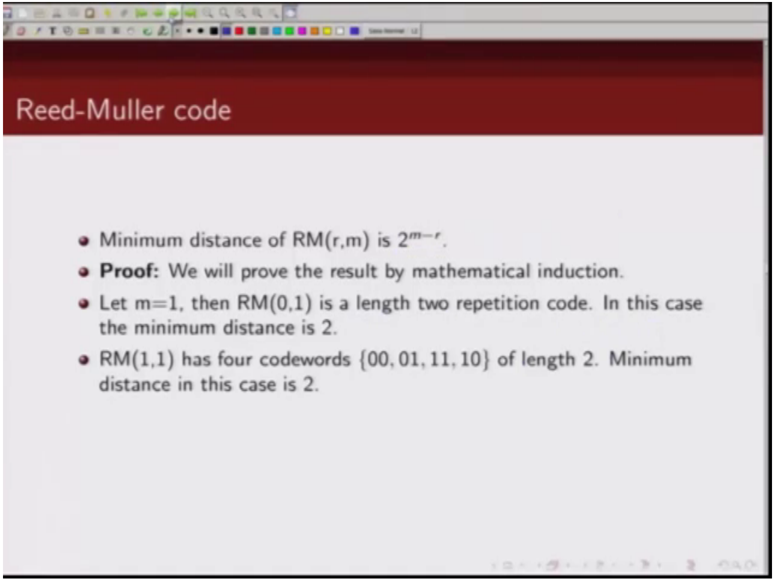
$G = [1\ 1]$

So this holds true for

(Refer Slide Time 17:14)

m equal to 1 and r equal to 0. Now let's see if it holds true also

(Refer Slide Time 17:19)

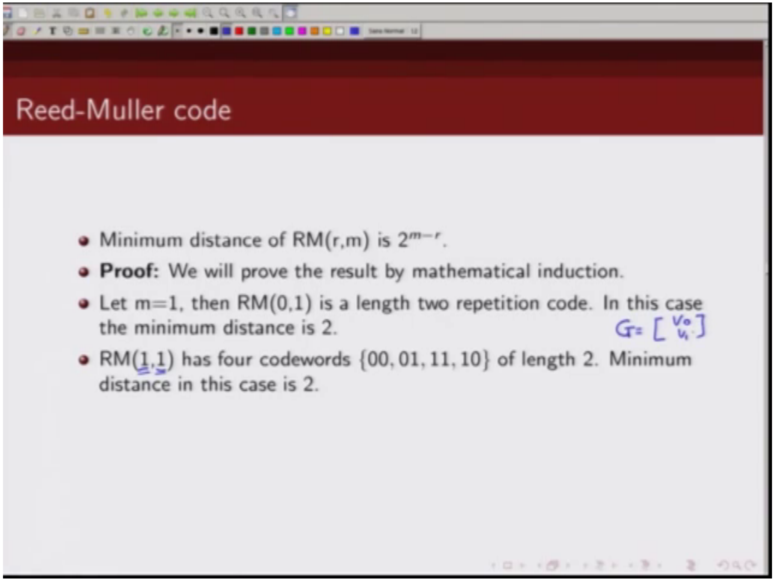


Reed-Muller code

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

for m equal to 1 and r equal to 1. Now m equal to 1 and r equal to 1, so then length of the codeword is again 2. So g will consist of v 0 and v 1,

(Refer Slide Time 17:39)



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.

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

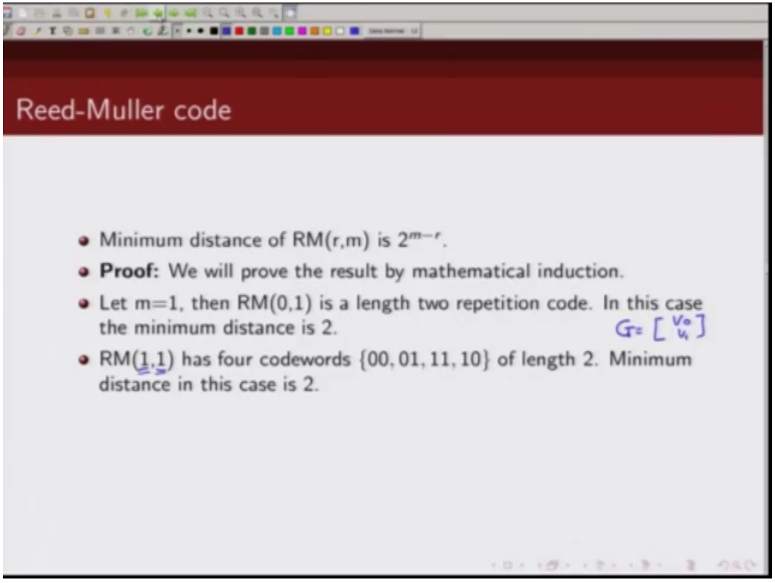
- RM($\underline{1},\underline{1}$) has four codewords $\{00, 01, 11, 10\}$ of length 2. Minimum distance in this case is 2.

Ok

(Refer Slide Time 17:40)

and what is my v 0 and v 1?

(Refer Slide Time 17:43)



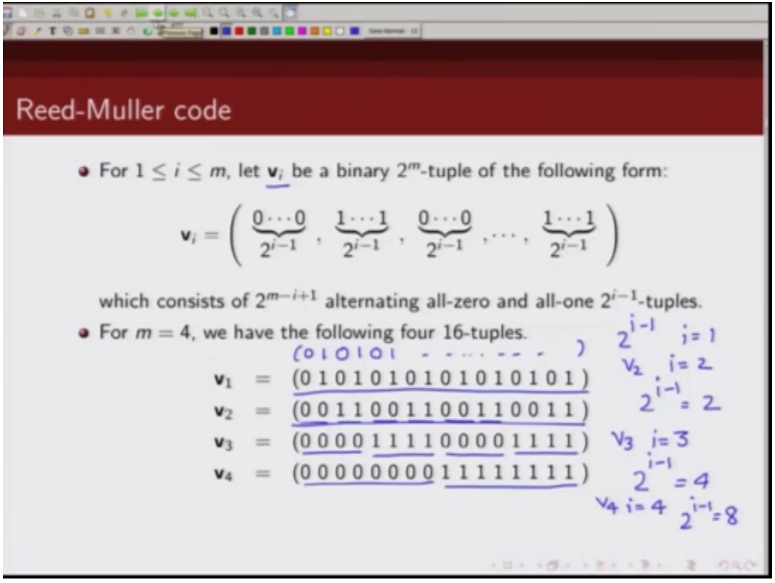
Reed-Muller code

- 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 1 is

(Refer Slide Time 17:44)



Reed-Muller code

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

$$\mathbf{v}_i = \left(\underbrace{0\cdots0}_{2^{i-1}}, \underbrace{1\cdots1}_{2^{i-1}}, \underbrace{0\cdots0}_{2^{i-1}}, \ldots, \underbrace{1\cdots1}_{2^{i-1}} \right)$$

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.

$(0\,1\,0\,1\,0\,1 \; - \; - \cdots - - \; - \;)$

$$\begin{aligned}
\mathbf{v}_1 &= (0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1) \\
\mathbf{v}_2 &= (0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1) \\
\mathbf{v}_3 &= (0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1) \\
\mathbf{v}_4 &= (0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1)
\end{aligned}$$

2^{i-1} $i=1$

v_2 $i=2$

$2^{i-1} = 2$

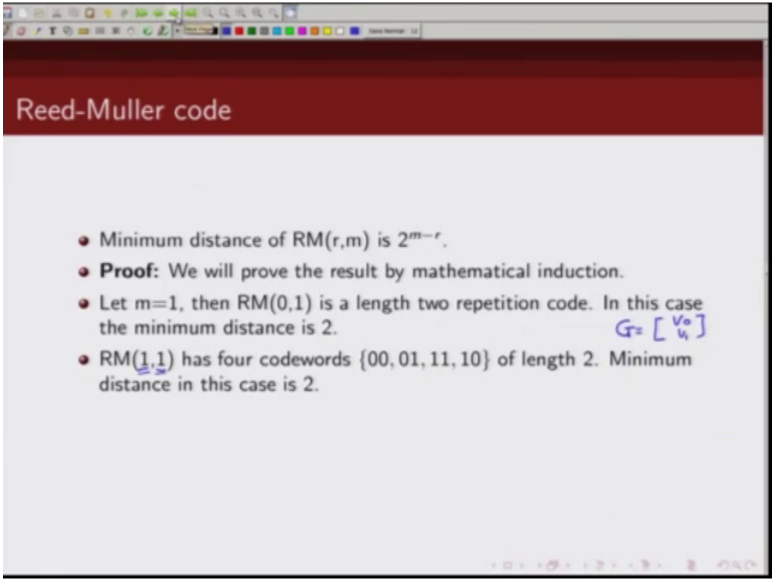
v_3 $i=3$

$2^{i-1} = 4$

v_4 $i=4$ $2^{i-1} = 8$

0 1 0 1 and v 0 is 1, so this is length 2. So what I will get is

(Refer Slide Time 17:53)

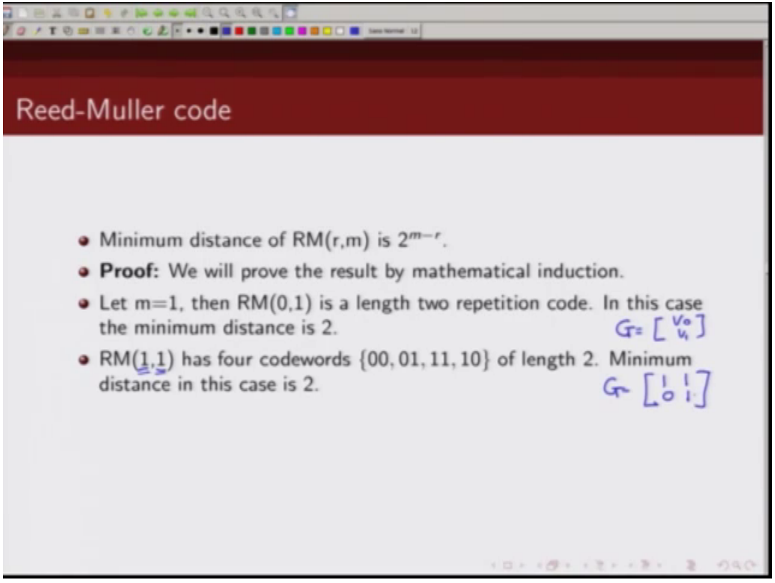


Reed-Muller code

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

g is 1 1 and this is 0 1.

(Refer Slide Time 18:00)

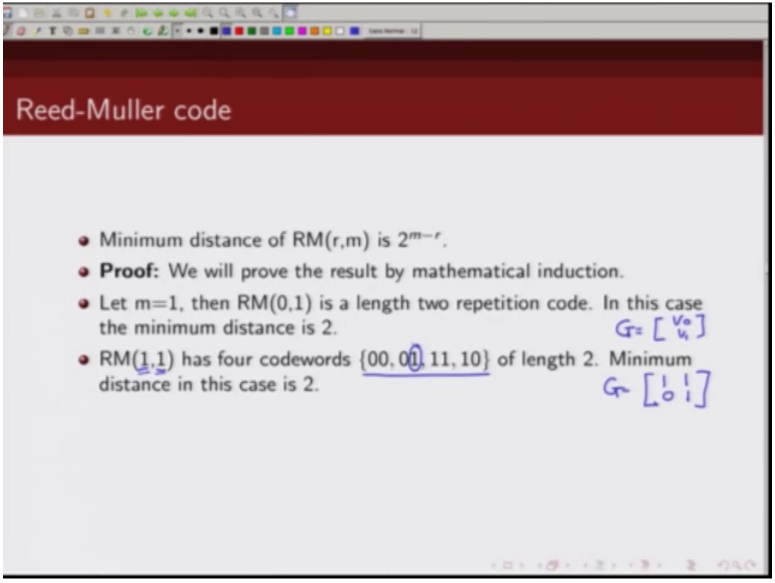


Reed-Muller code

- 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($\underline{1}$,$\underline{1}$) has four codewords {00, 01, 11, 10} of length 2. Minimum distance in this case is 2. $G = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

So this will be my generator matrix. Now this will generate these following codewords of length 2 and what is the minimum distance between these codes? That's 1, we can say minimum weight codeword is minimum weight of non zero codeword is 1;

(Refer Slide Time 18:17)

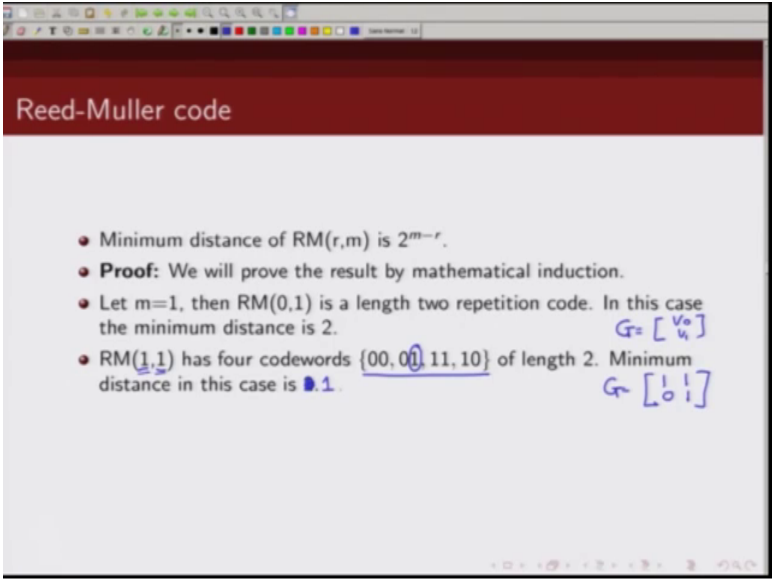


Reed-Muller code

- 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 $\underline{\{00, 01, 11, 10\}}$ of length 2. Minimum distance in this case is 2. $G= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

so minimum distance in this case is 1,

(Refer Slide Time 18:21)

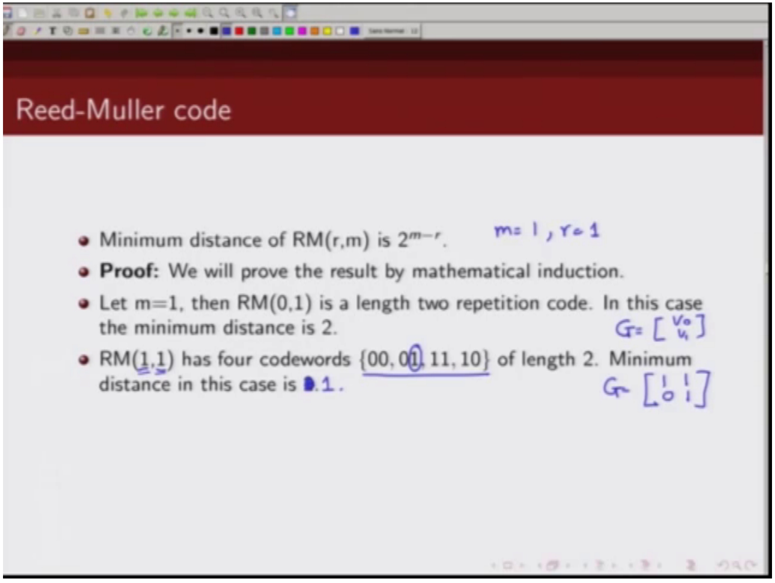


Reed-Muller code

- 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 $\underline{\{00, 01, 11, 10\}}$ of length 2. Minimum distance in this case is 1. $G= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

Ok and let's check. So in this case m is 1 and r is 1. So 2 raised to power 1 minus 1,

(Refer Slide Time 18:33)

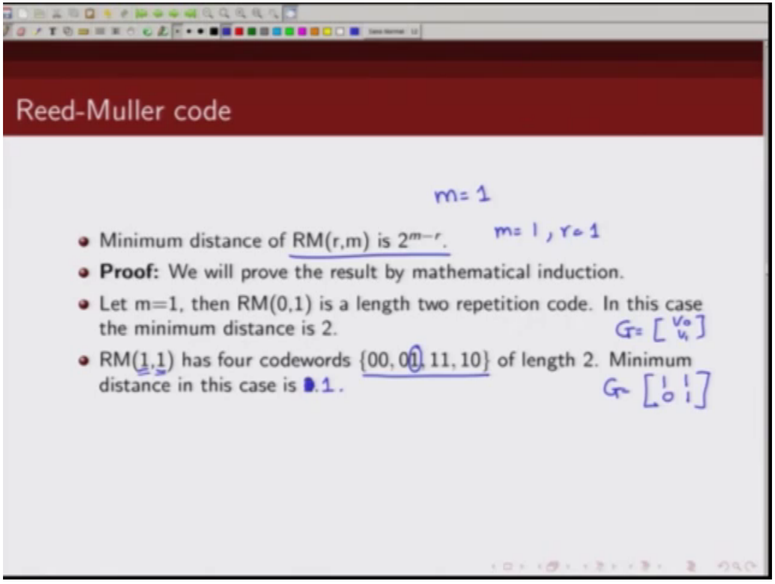


Reed-Muller code

- 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. $G= \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}$
- RM($\underline{1},\underline{1}$) has four codewords $\underline{\{00, 0\boxed{1}, 11, 10\}}$ of length 2. Minimum distance in this case is 1. $G= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$

2 raised to power 0 that's 1. And that's what we are getting, fine? So then this is true for m equal to 1. Now let's

(Refer Slide Time 18:48)



Reed-Muller code

m= 1

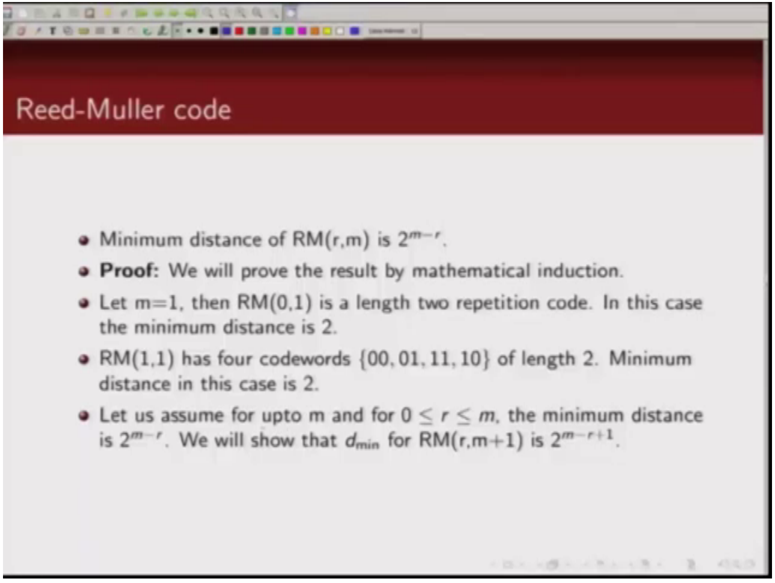
- Minimum distance of $\underline{\text{RM(r,m) is } 2^{m-r}}$. m= 1, r= 1
- **Proof:** We will prove the result by mathematical induction.
- Let m=1, then RM(0,1) is a length two repetition code. In this case the minimum distance is 2. $G= \begin{bmatrix} v_0 \\ v_1 \end{bmatrix}$
- RM($\underline{1},\underline{1}$)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

assume it's true for any m equal to m and then we will try to prove that it is also

(Refer Slide Time 18:55)

true for m equal to m plus 1. So let's assume that it is true for up to m

(Refer Slide Time 19:03)



Reed-Muller code

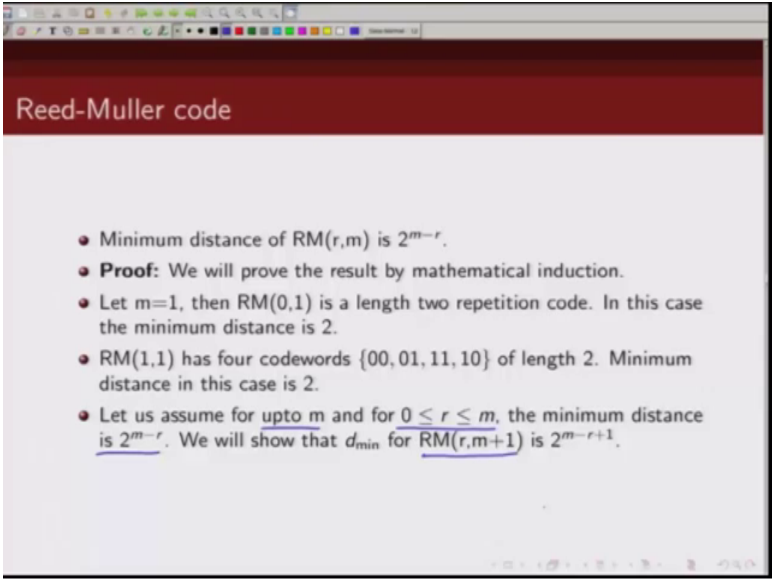
- Minimum distance of RM(r,m) is 2^{m-r}.
- **Proof:** We will prove the result by mathematical induction.
- Let m=1, then RM(0,1) is a length two repetition code. In this case the minimum distance is 2.
- RM(1,1) has four codewords $\{00, 01, 11, 10\}$ of length 2. Minimum distance in this case is 2.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

and for any order where order can be from zero to m, let's assume that this is true. So minimum distance is given by 2 raised to power m minus r. Now what we are going to show is this is also true for m plus 1. And what should be the minimum distance for

(Refer Slide Time 19:25)

m plus 1? It should be 2 raised to power

(Refer Slide Time 19:27)



Reed-Muller code

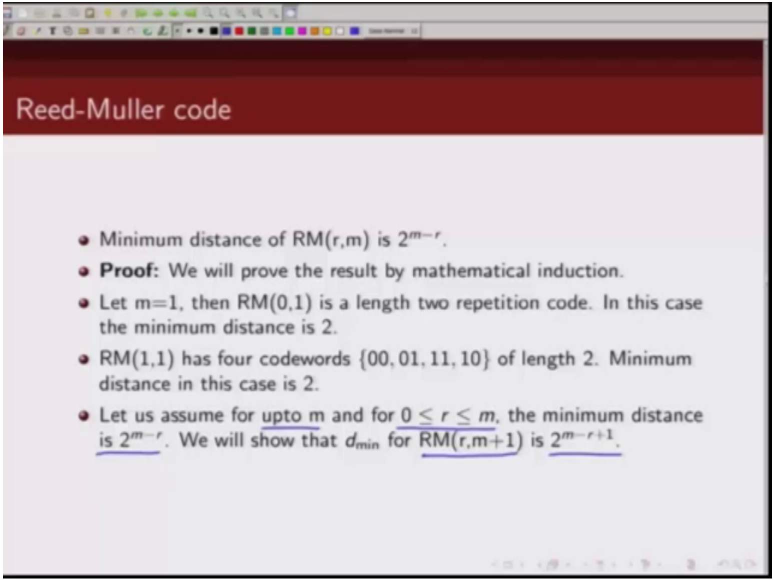
- Minimum distance of RM(r,m) is 2^{m-r}.
- **Proof:** We will prove the result by mathematical induction.
- Let m=1, then RM(0,1) is a length two repetition code. In this case the minimum distance is 2.
- RM(1,1) has four codewords $\{00, 01, 11, 10\}$ of length 2. Minimum distance in this case is 2.
- 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 plus 1 minus r, that's this. So next what we are going to show you is that minimum distance of m rth order

(Refer Slide Time 19:41)

Reed-Muller code r m plus 1 Reed-Muller

(Refer Slide Time 19:44)

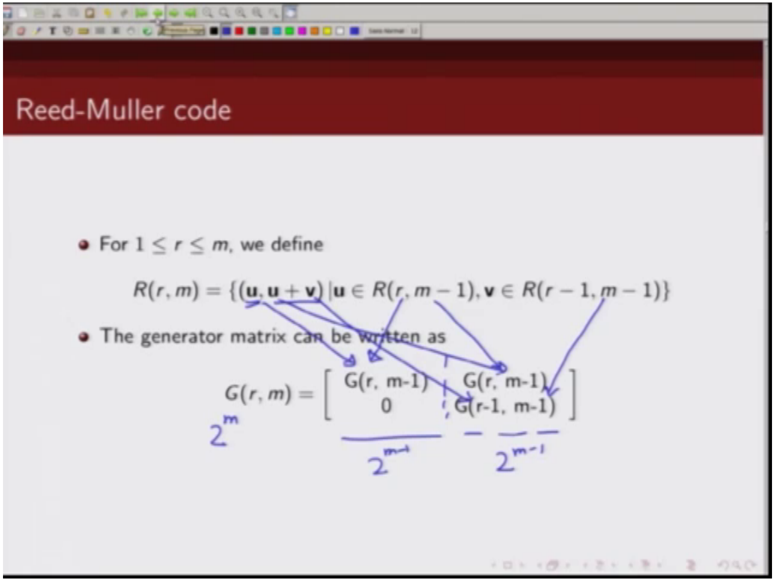


Reed-Muller code

- 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}.

code is basically given by this. Now to prove this, we are going to make use of this construction of Reed-Muller code; that

(Refer Slide Time 19:56)



Reed-Muller code

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

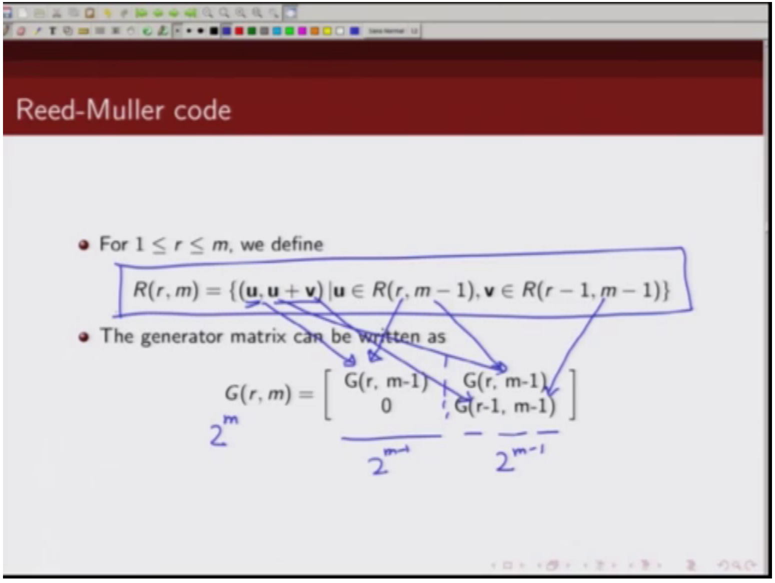
$$R(r,m) = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \,|\, \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}$$

- The generator matrix can be written as

$$\underbrace{G(r,m)}_{2^m} = \left[\begin{array}{c|c} G(r,\ m\text{-}1) & G(r,\ m\text{-}1) \\ 0 & G(r\text{-}1,\ m\text{-}1) \end{array}\right]$$
$$\begin{array}{cc} \underbrace{\qquad\qquad}_{2^{m-1}} & \underbrace{\qquad\qquad}_{2^{m-1}} \end{array}$$

Reed-Muller code of order r and m can be constructed recursively using this. We are going

(Refer Slide Time 20:06)



Reed-Muller code

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

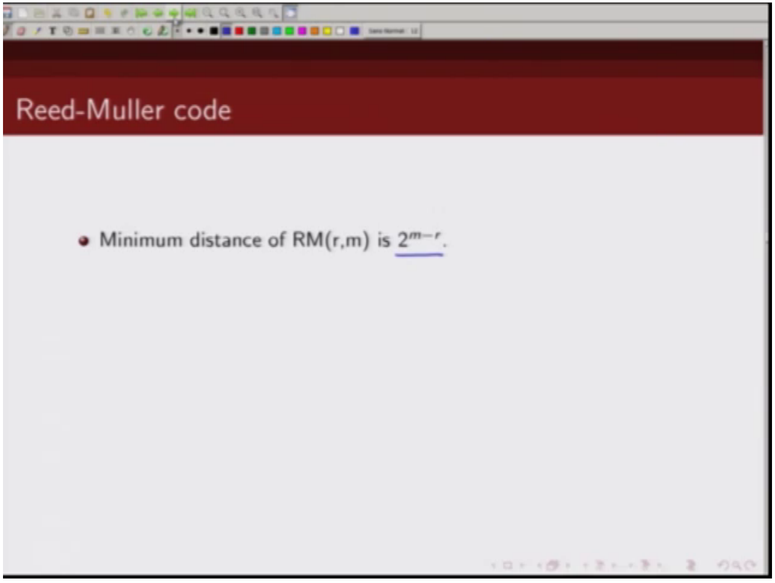
$$\boxed{R(r,m) = \{(\mathbf{u}, \mathbf{u}+\mathbf{v}) \,|\, \mathbf{u} \in R(r, m-1), \mathbf{v} \in R(r-1, m-1)\}}$$

- The generator matrix can be written as

$$\underbrace{G(r,m)}_{2^m} = \left[\begin{array}{c|c} G(r,\ m\text{-}1) & G(r,\ m\text{-}1) \\ 0 & G(r\text{-}1,\ m\text{-}1) \end{array}\right]$$
$$\begin{array}{cc} \underbrace{\qquad\qquad}_{2^{m-1}} & \underbrace{\qquad\qquad}_{2^{m-1}} \end{array}$$

to make use of this construction to prove our result. So let's see

(Refer Slide Time 20:13)



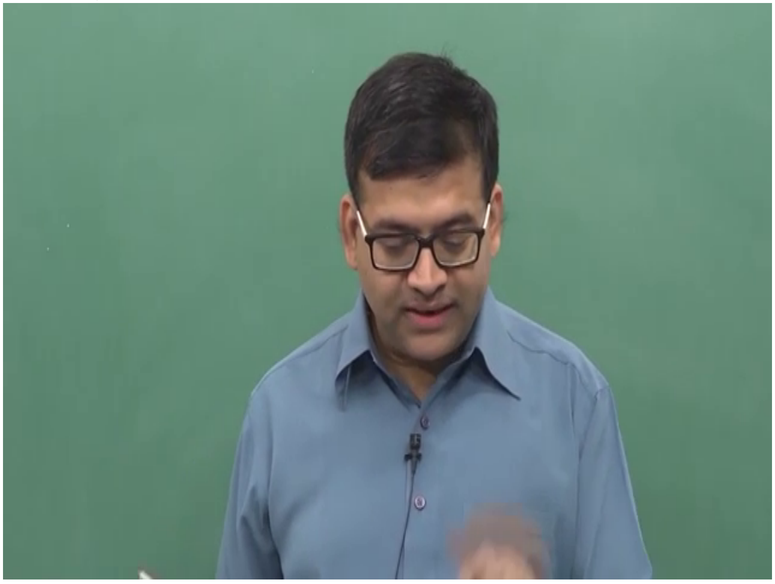
Reed-Muller code

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

how we proceed.

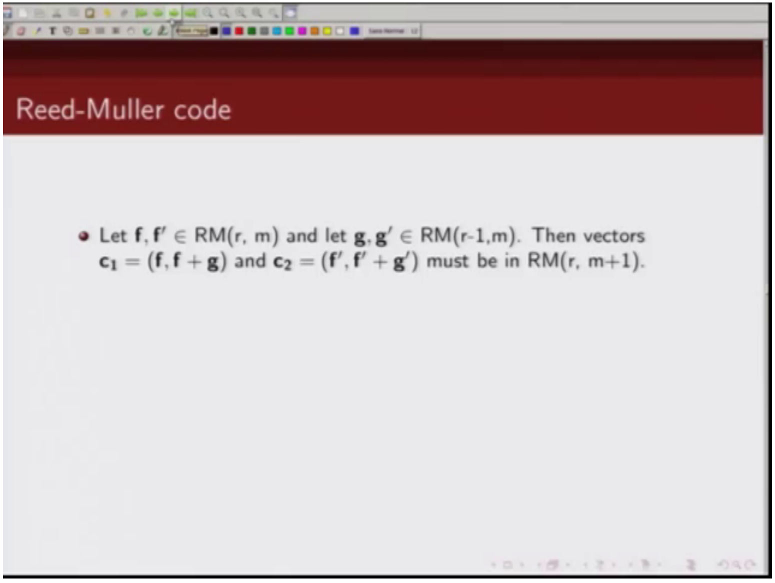
So let's consider

(Refer Slide Time 20:16)



2 codewords

(Refer Slide Time 20:18)

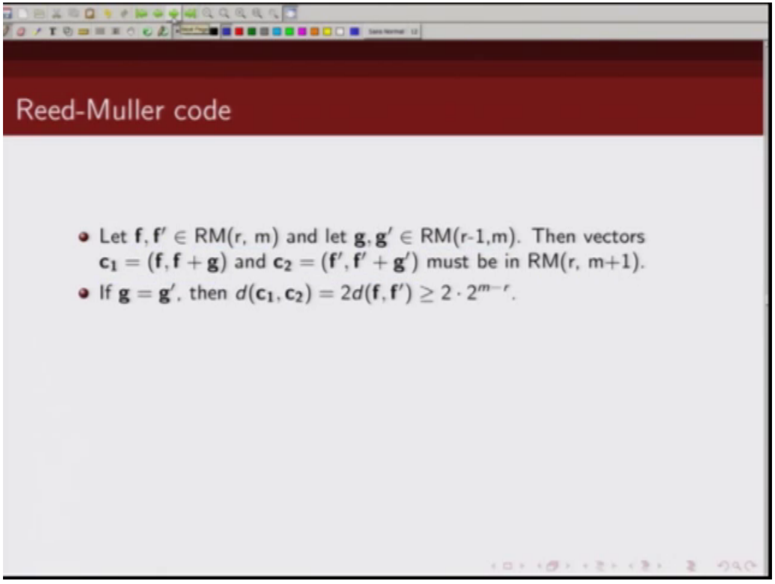


Reed-Muller code

- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ 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 RM(r, m+1).

f, f dash which belongs to Reed-Muller code of order r and length 2 raised to power m. And let g g dash belongs to Reed-Muller code of order r minus 1 and length 2 m. Then we defining two codewords, then Reed-Muller code of order r and length 2 raised to power m plus 1 is of the form, we just said u and u plus 1. So these codeword c 1 and c 2 which is of the form f and f plus g f dash and f dash plus g, they must be codeword belonging to Reed-Muller code. And this follows from our recursive construction of Reed-Muller code which we just mentioned. So c 1 and c 2 must be codewords for this Reed-Muller code.

(Refer Slide Time 21:20)



Reed-Muller code

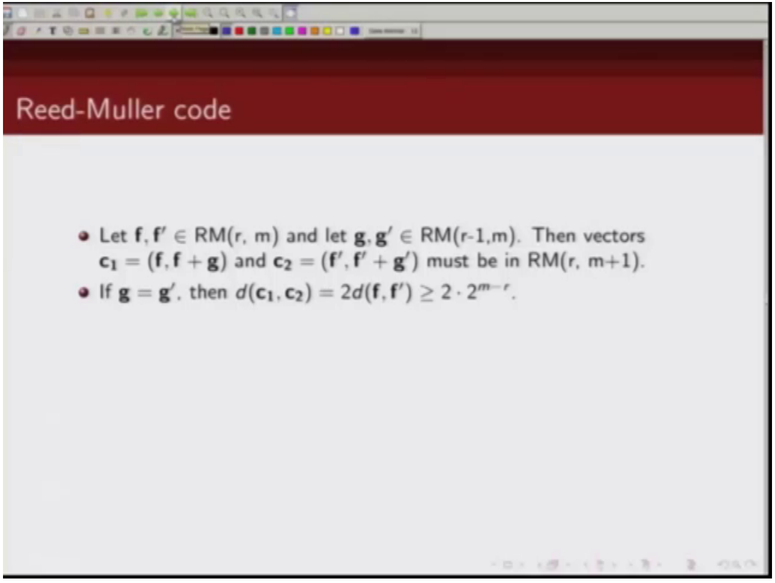
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f} + \mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}' + \mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1,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

Now let us try to compute the minimum distance

(Refer Slide Time 21:24)

between these codes c 1 and c 2 which are codewords

(Refer Slide Time 21:29)

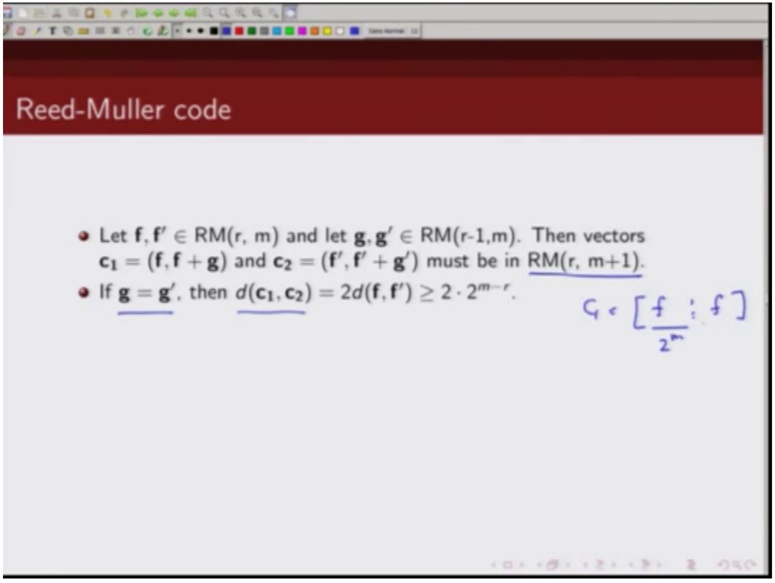


Reed-Muller code

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

Reed-Muller code of order r and length 2 raised to power m plus 1. So first case that we will consider is when g is same as g dash and second case that we will consider is when g is not same as g dash. So when g is same as g dash what is the minimum distance between c 1 and c 2? Now if g and g dash are same then basically your code c 1 is nothing but it is f here of length 2 m and there is another codeword f of

(Refer Slide Time 22:06)



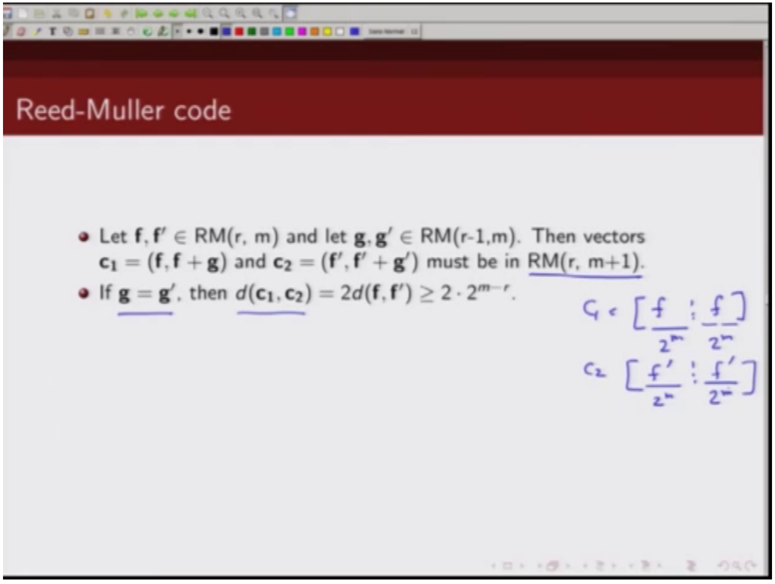
Reed-Muller code

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

$c_1 < [\underbrace{f}_{2^m} : f]$

length 2 m and c 2 is f dash of length 2 m and then you have f dash of length 2 m. So what is the minimum distance

(Refer Slide Time 22:18)



Reed-Muller code

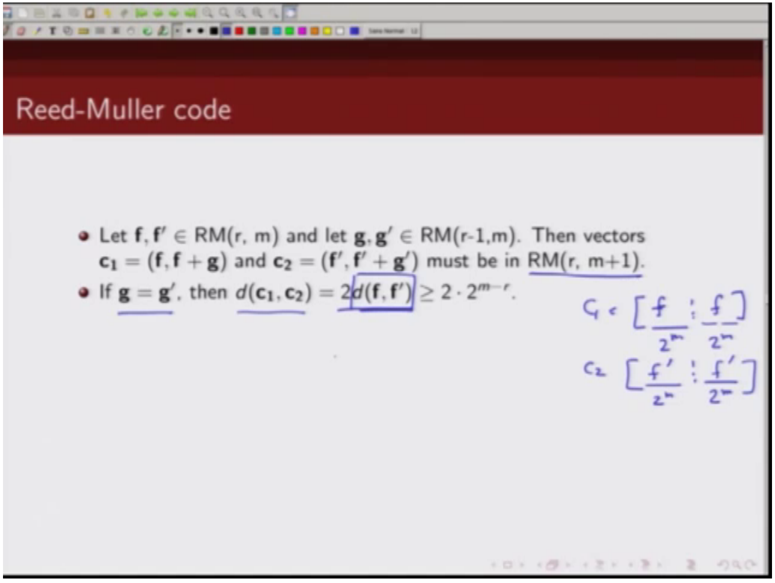
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f}+\mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}'+\mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2)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

$c_1 < [\underbrace{f}_{2^m} : \underbrace{f}_{2^m}]$

$c_2 \; [\underbrace{f'}_{2^m} : \underbrace{f'}_{2^m}]$

between this code? It is, minimum distance between f and f dash plus minimum distance between f and f dash. So that's what we are writing here. So if g is equal to g dash, the minimum distance between c 1 and c 2 is 2 times the minimum distance between f and f dash. And what is the minimum distance between f and f dash?

(Refer Slide Time 22:42)



Reed-Muller code

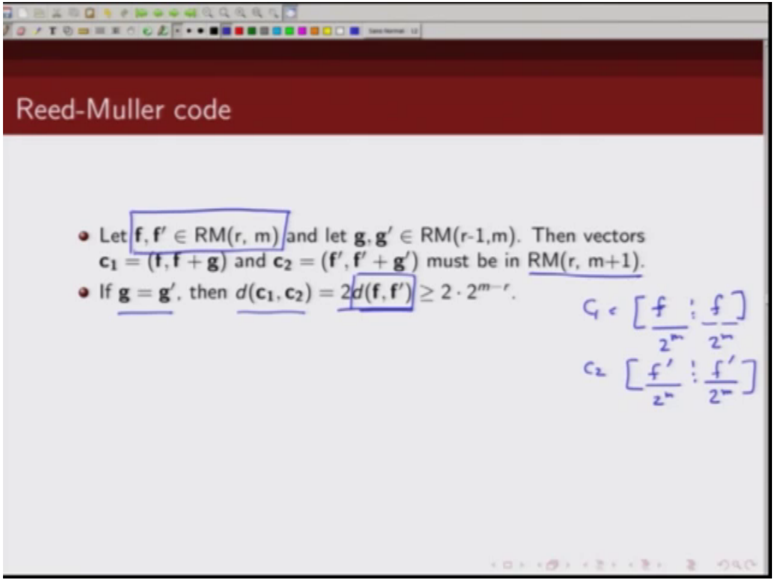
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f} + \mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}' + \mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1,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

$c_1 \left[\underset{2^m}{\underline{f}} \; \vdots \; \underset{2^m}{\underline{f}} \right]$

$c_2 \left[\underset{2^m}{\underline{f'}} \; \vdots \; \underset{2^m}{\underline{f'}} \right]$

f and f dash belongs to Reed-Muller code

(Refer Slide Time 22:46)



Reed-Muller code

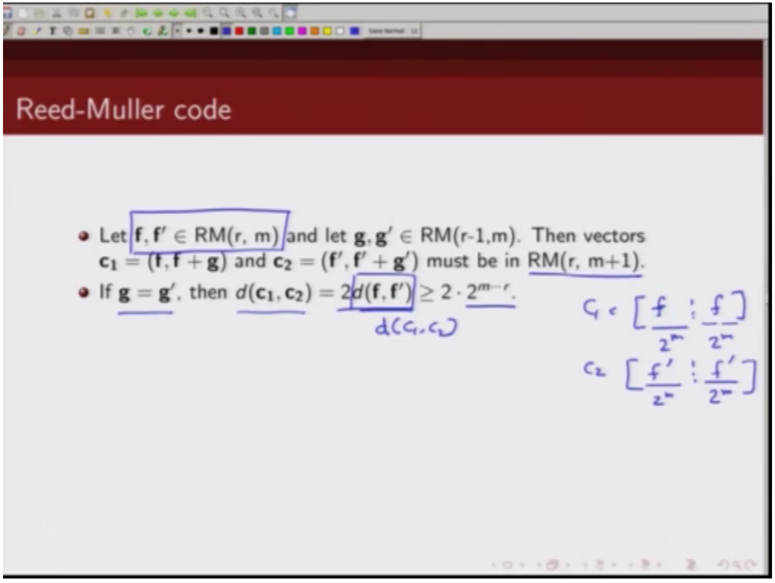
- Let $\boxed{\mathbf{f}, \mathbf{f}' \in \text{RM(r, m)}}$ and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f} + \mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}' + \mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1,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

$c_1 \left[\underset{2^m}{\underline{f}} \; \vdots \; \underset{2^m}{\underline{f}} \right]$

$c_2 \left[\underset{2^m}{\underline{f'}} \; \vdots \; \underset{2^m}{\underline{f'}} \right]$

of order r and length 2 raised to power m. So their minimum distance should be 2 raised to power m minus r. So then from this we get that minimum distance between c 1 and c 2 which are 2 codewords

(Refer Slide Time 23:05)



Reed-Muller code

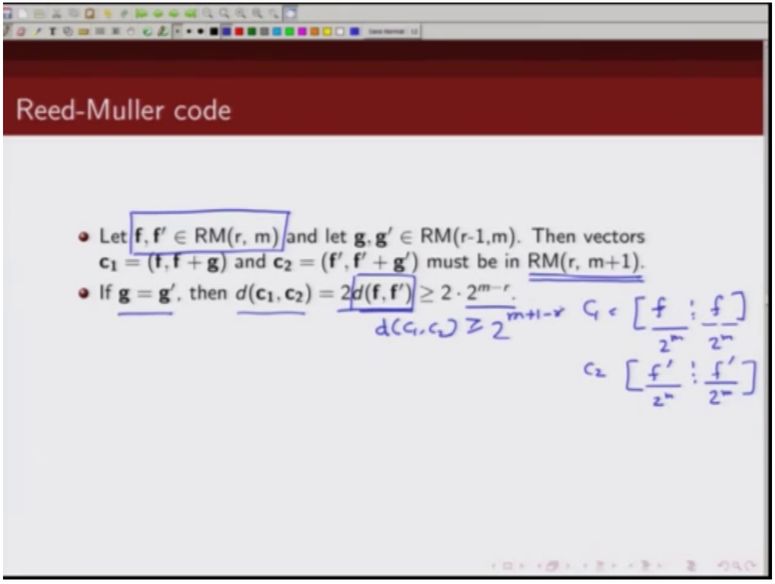
- Let $\mathbf{f}, \mathbf{f}' \in \mathrm{RM}(r, m)$ and let $\mathbf{g}, \mathbf{g}' \in \mathrm{RM}(r\text{-}1,m)$. Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f}+\mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}'+\mathbf{g}')$ must be in $\mathrm{RM}(r, m{+}1)$.
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2) = 2\underbrace{d(\mathbf{f}, \mathbf{f}')}_{d(c_1,c_2)} \geq 2 \cdot 2^{m-r}$.

$c_1 < [\underbrace{f}_{2^m} \vdots \underbrace{f}_{2^m}]$

$c_2 \; [\underbrace{f'}_{2^m} \vdots \underbrace{f'}_{2^m}]$

belonging to Reed-Muller code order r and length 2 raised to power m plus 1, this should be greater than or equal to 2 raised to power m plus 1 minus r.

(Refer Slide Time 23:18)



Reed-Muller code

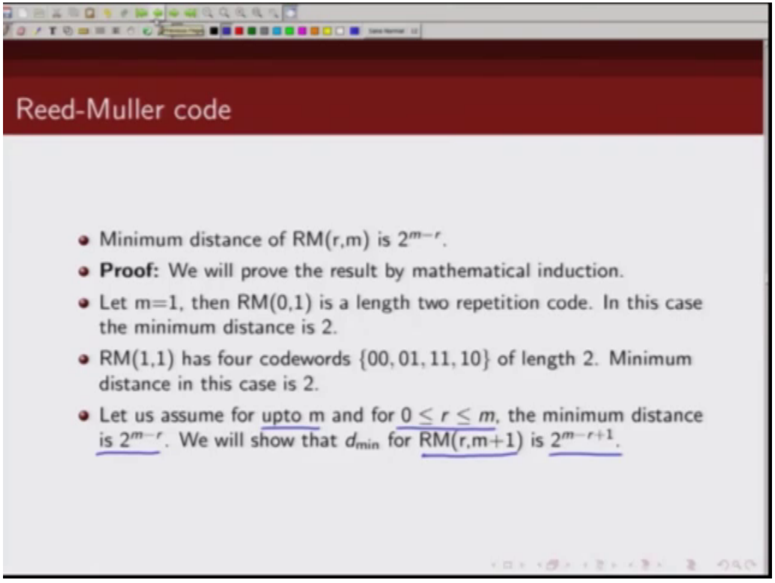
- Let $\mathbf{f}, \mathbf{f}' \in \mathrm{RM}(r, m)$ and let $\mathbf{g}, \mathbf{g}' \in \mathrm{RM}(r\text{-}1,m)$. Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f}+\mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}'+\mathbf{g}')$ must be in $\mathrm{RM}(r, m{+}1)$.
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2) = 2\underbrace{d(\mathbf{f}, \mathbf{f}')}_{d(c_1,c_2) \geq 2^{m+1-r}} \geq 2 \cdot 2^{m-r}$.

$c_1 < [\underbrace{f}_{2^m} \vdots \underbrace{f}_{2^m}]$

$c_2 \; [\underbrace{f'}_{2^m} \vdots \underbrace{f'}_{2^m}]$

So for this particular case, we have shown that minimum distance

(Refer Slide Time 23:25)

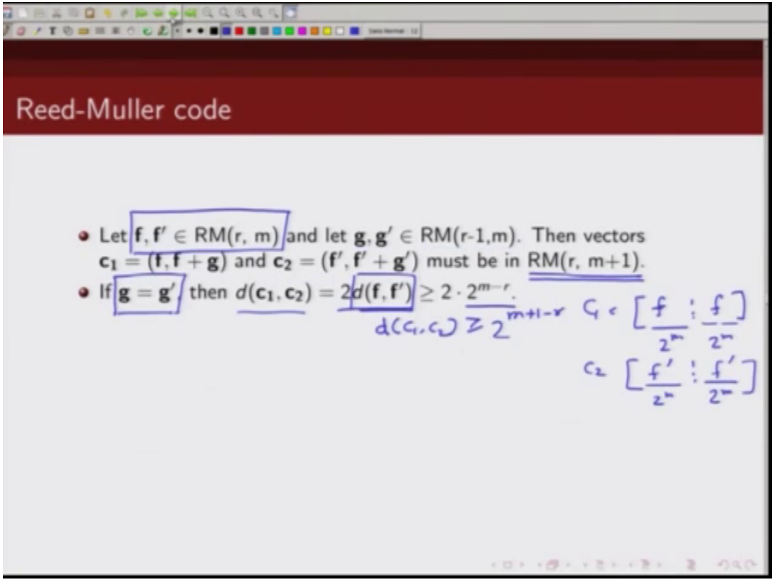


Reed-Muller code

- 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.
- 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}.

is indeed this. Now we will also have to show if g

(Refer Slide Time 23:31)



Reed-Muller code

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

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

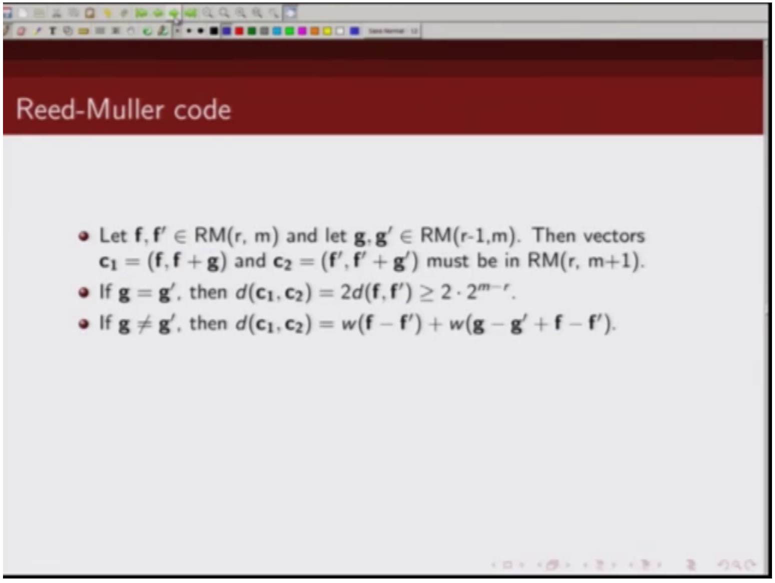
$c_1 \; [\underbrace{f}_{2^m} \,\vdots\, \underbrace{f}_{2^m}]$

$c_2 \; [\underbrace{f'}_{2^m} \,\vdots\, \underbrace{f'}_{2^m}]$

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

So next we

(Refer Slide Time 23:40)

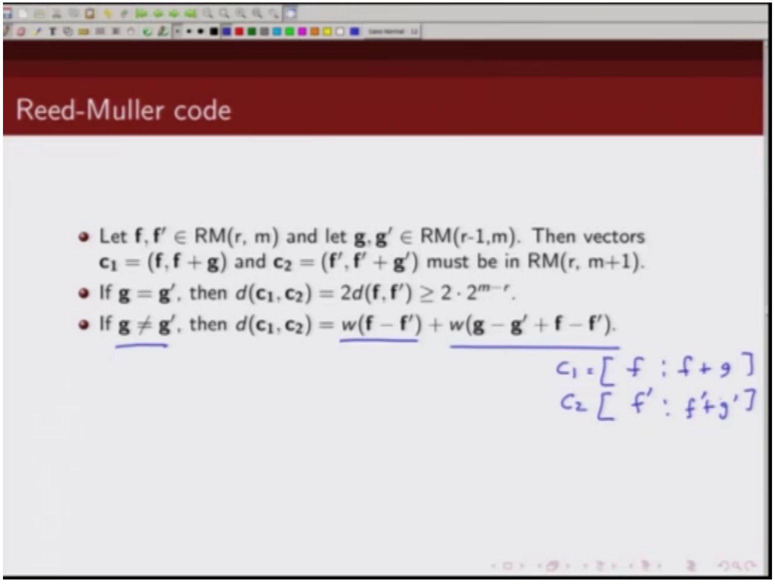


Reed-Muller code

- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ 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 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 $d(\mathbf{c_1}, \mathbf{c_2}) = w(\mathbf{f}-\mathbf{f}') + w(\mathbf{g}-\mathbf{g}'+\mathbf{f}-\mathbf{f}')$.

consider the case when g is not same as g dash. Now if g is not same as g dash, 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 weight of f minus f dash plus weight of g minus g dash plus weight of f minus f dash. If we are talking of binary codes this will be basically plus, you will also find, because that's the same thing. So if you have 2 codewords, let's call it c 1 which is f here and this is f plus g and you have c 2 which is f dash, f dash plus g dash, then minimum distance between code is

(Refer Slide Time 24:32)



Reed-Muller code

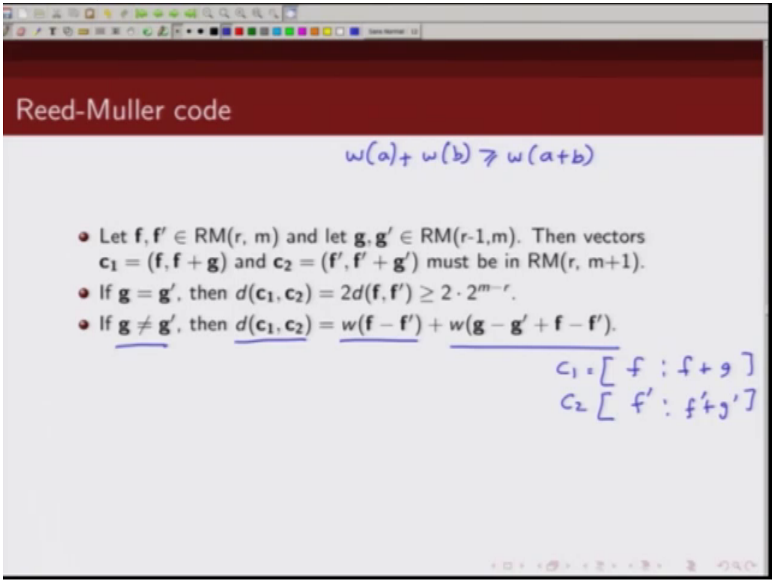
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c_1} = (\mathbf{f}, \mathbf{f}+\mathbf{g})$ and $\mathbf{c_2} = (\mathbf{f}', \mathbf{f}'+\mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c_1}, \mathbf{c_2})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

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

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

f minus, weight of f minus f dash and weight of this minus this. So that's 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's say if you have 2 m-tuples, n-tuples then weight of a plus weight of b where a and b are some n-tuples, this is basically, weight of a plus weight of b is greater than equal to weight of a plus b, right?

(Refer Slide Time 25:13)



Reed-Muller code

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

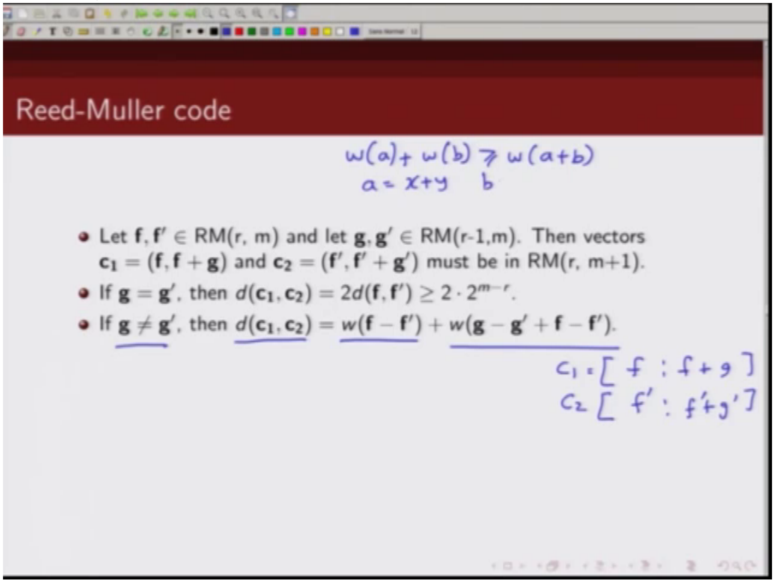
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f}+\mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}'+\mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2)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

$c_1 = [\, f \; : \; f+g \,]$

$c_2 \; [\, f' \; : \; f'+g' \,]$

Now if I consider a to be x plus y and b to be

(Refer Slide Time 25:20)



Reed-Muller code

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

$a = x+y \quad b$

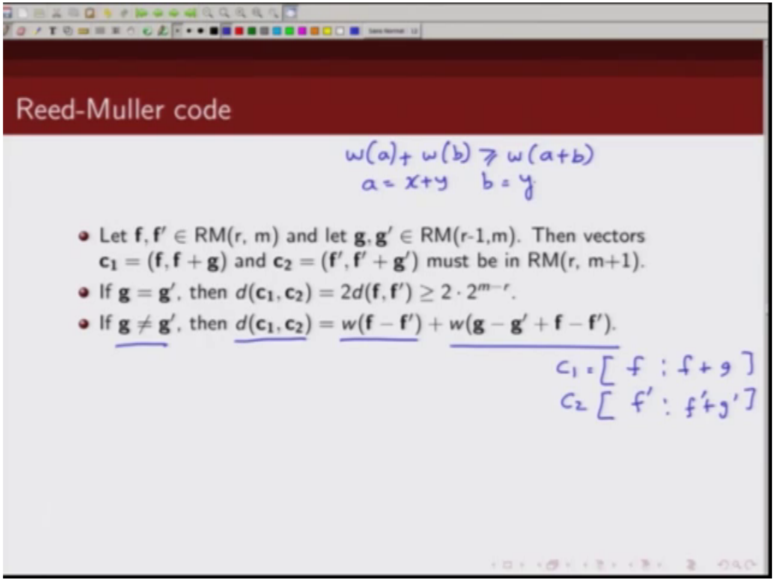
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f}+\mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}'+\mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2)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

$c_1 = [\, f \; : \; f+g \,]$

$c_2 \; [\, f' \; : \; f'+g' \,]$

y and let's say

(Refer Slide Time 25:25)



Reed-Muller code

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

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

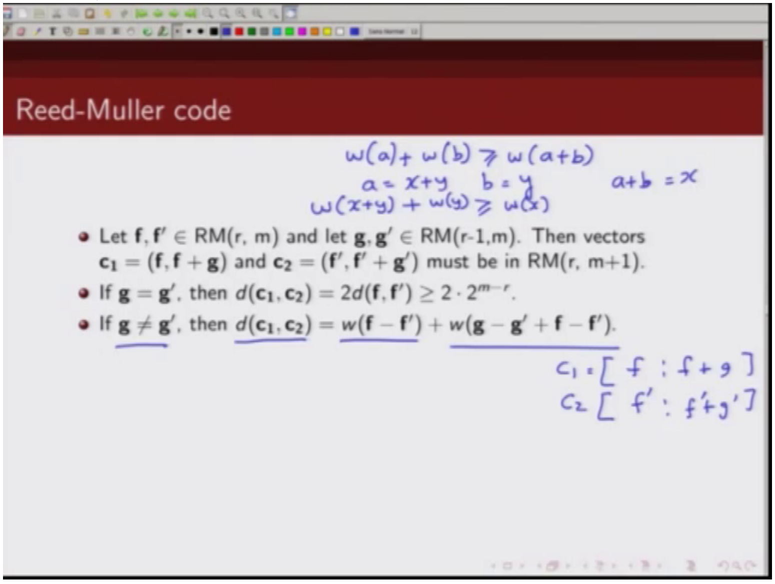
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f}+\mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}'+\mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2)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

$c_1 = [\, f \;:\; f+g \,]$

$c_2 \; [\, f' \;:\; f'+g' \,]$

x plus y they are all binary m-tuples we are talking about, then a plus b will be x plus y, so that's given by x. So what we will get is weight of x plus y plus weight of y is greater than equal to weight of x, right?

(Refer Slide Time 25:50)



Reed-Muller code

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

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

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

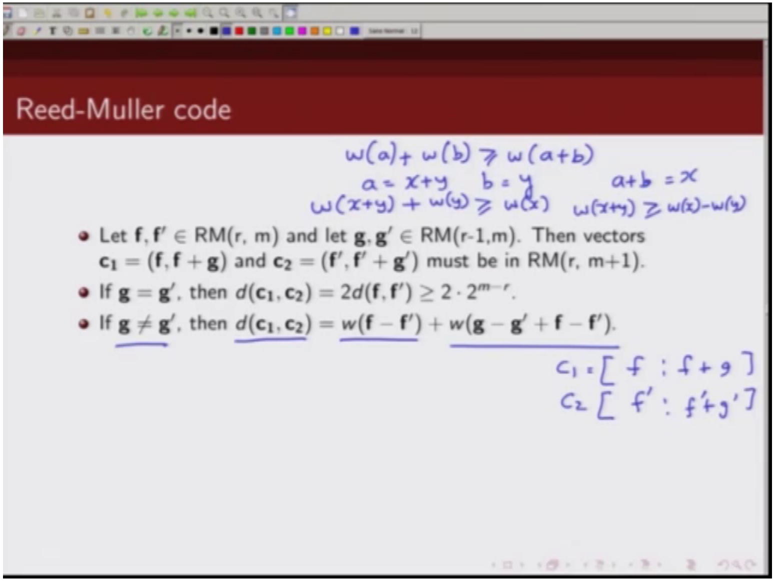
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f}+\mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}'+\mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2)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

$c_1 = [\, f \;:\; f+g \,]$

$c_2 \; [\, f' \;:\; f'+g' \,]$

Or we can write weight of x plus y is greater than equal to weight of x minus weight of y. Next we are going to make use of this result to simplify this expression.

(Refer Slide Time 26:06)



Reed-Muller code

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

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

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

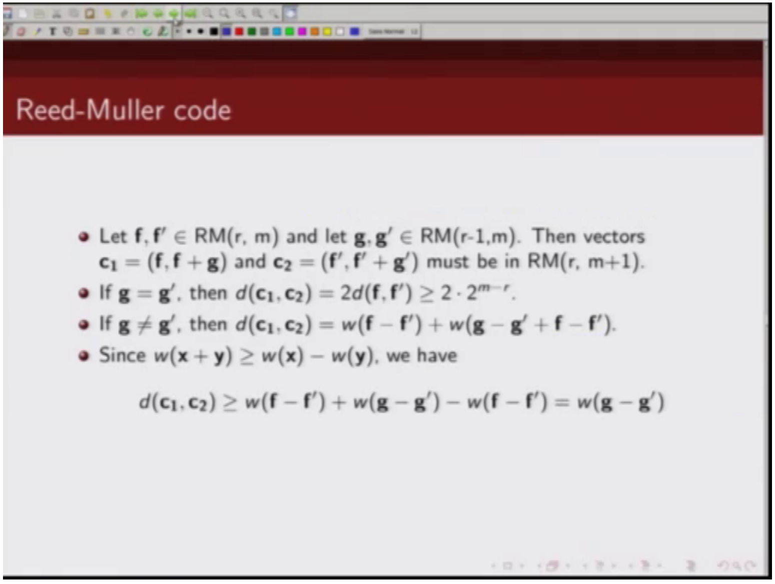
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c_1} = (\mathbf{f}, \mathbf{f}+\mathbf{g})$ and $\mathbf{c_2} = (\mathbf{f}', \mathbf{f}'+\mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c_1}, \mathbf{c_2})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

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

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

This, you can consider , this is my x and this is my y. So I can write weight of x plus y to be greater than equal to weight of x minus weight of y. So when I do that

(Refer Slide Time 26:23)



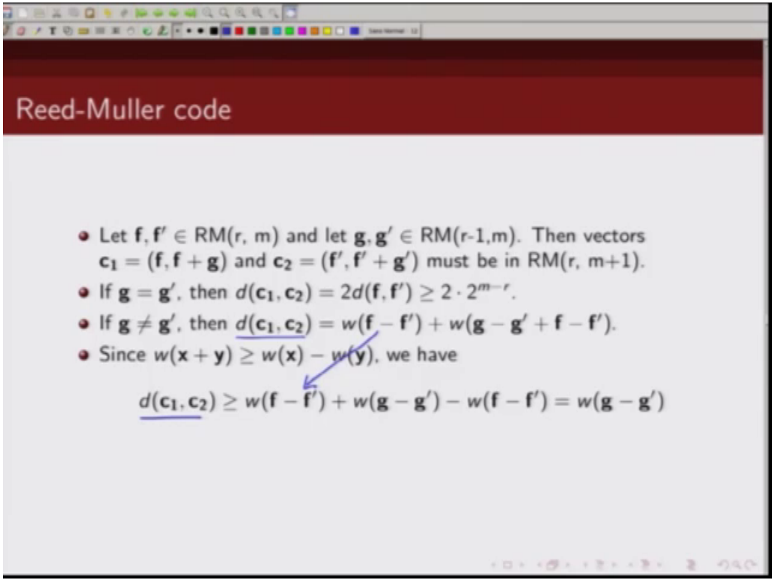
Reed-Muller code

- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ 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 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

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

then distance, minimum distance between c 1 and c 2 is this term coming here and what did I do,

(Refer Slide Time 26:34)



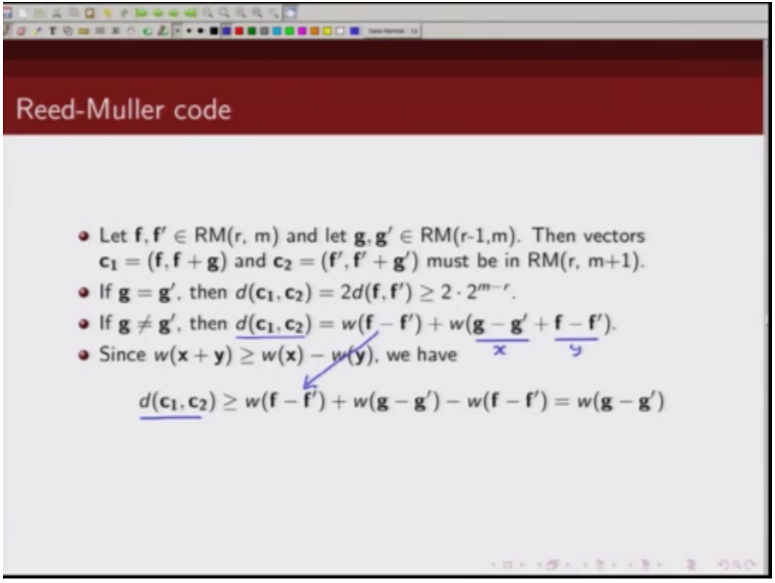
Reed-Muller code

- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ 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 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 $\underline{d(\mathbf{c}_1, \mathbf{c}_2)} = w(\mathbf{f} - \mathbf{f}') + w(\mathbf{g} - \mathbf{g}' + \mathbf{f} - \mathbf{f}')$.
- Since $w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x}) - w(\mathbf{y})$, we have

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

this was weight of x, let's say this was x, this was y. This I can write

(Refer Slide Time 26:39)



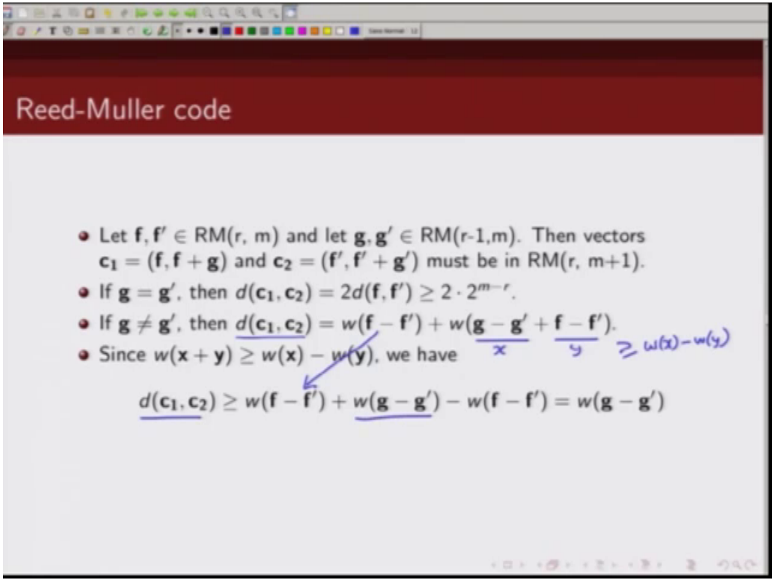
Reed-Muller code

- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ 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 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 $\underline{d(\mathbf{c}_1, \mathbf{c}_2)} = w(\mathbf{f} - \mathbf{f}') + w(\underbrace{\mathbf{g} - \mathbf{g}'}_{x} + \underbrace{\mathbf{f} - \mathbf{f}'}_{y})$.
- Since $w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x}) - w(\mathbf{y})$, we have

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

as, this is then greater than equal to weight of x minus weight of y. So this weight of x is this term

(Refer Slide Time 26:47)



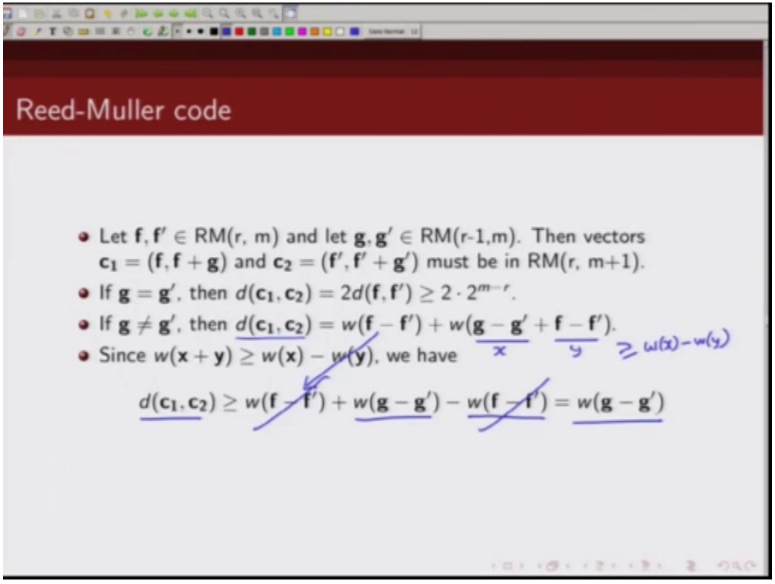
Reed-Muller code

- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ 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 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 $\underline{d(\mathbf{c}_1, \mathbf{c}_2)} = w(\mathbf{f} - \mathbf{f}') + w(\underbrace{\mathbf{g} - \mathbf{g}'}_{x} + \underbrace{\mathbf{f} - \mathbf{f}'}_{y})$. $\geq w(x) - w(y)$
- Since $w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x}) - w(\mathbf{y})$, we have

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

minus weight of y which is this term, fine? Now this this cancels out. What I get is weight of g minus g dash.

(Refer Slide Time 26:59)



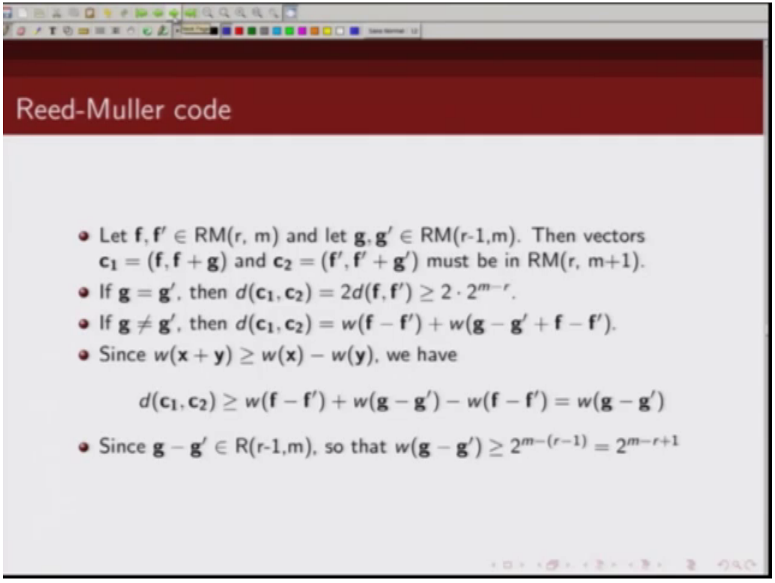
Reed-Muller code

- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ 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 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 $\underline{d(\mathbf{c}_1, \mathbf{c}_2)} = w(\mathbf{f} - \mathbf{f}') + w(\underbrace{\mathbf{g} - \mathbf{g}'}_{x} + \underbrace{\mathbf{f} - \mathbf{f}'}_{y})$. $\geq w(x) - w(y)$
- Since $w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x}) - w(\mathbf{y})$, we have

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

Now what is

(Refer Slide Time 27:02)



Reed-Muller code

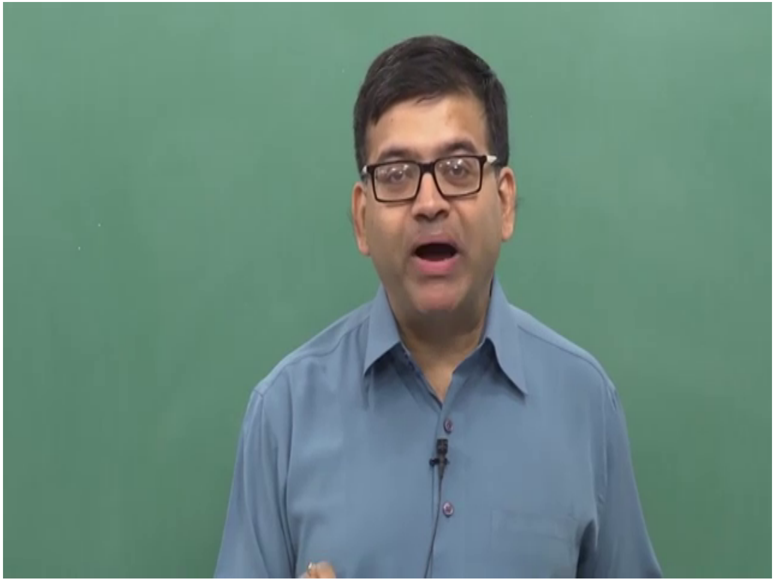
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c_1} = (\mathbf{f}, \mathbf{f} + \mathbf{g})$ and $\mathbf{c_2} = (\mathbf{f}', \mathbf{f}' + \mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c_1},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

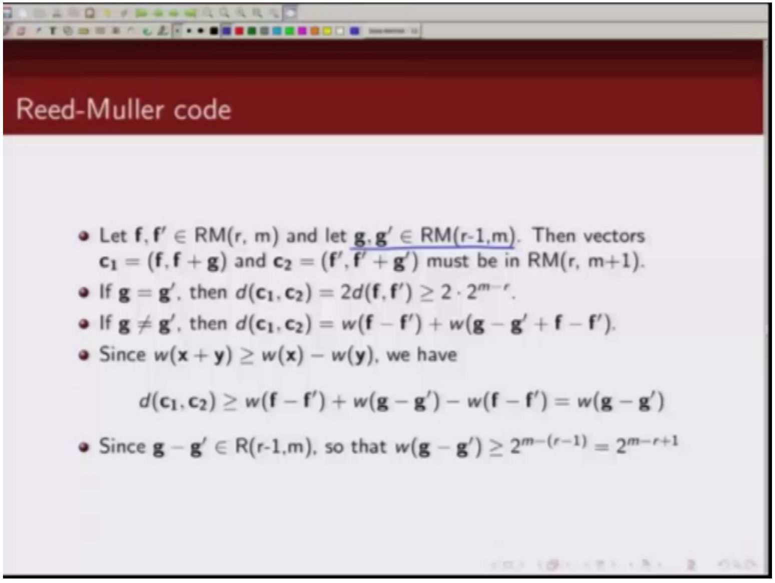
g? g belongs to Reed-Muller code of order r minus 1

(Refer Slide Time 27:10)



and length 2 raised to power m. Then what is the minimum distance

(Refer Slide Time 27:15)



Reed-Muller code

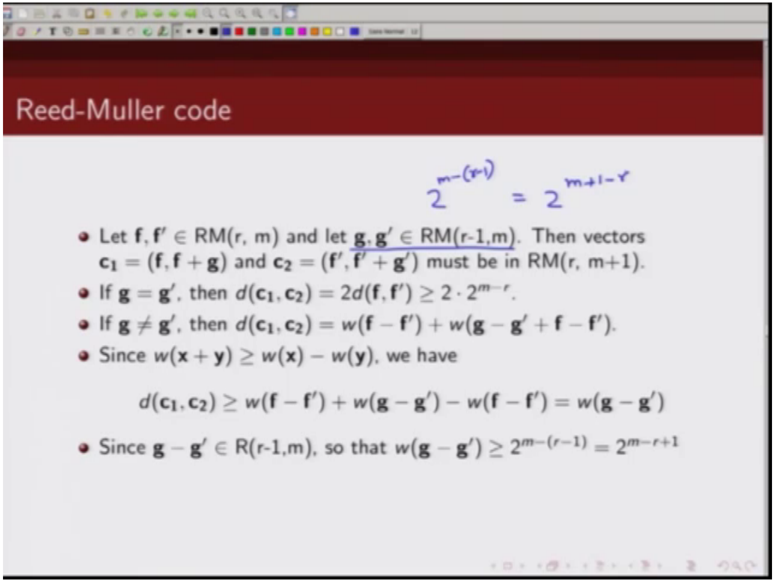
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ RM(r-1,m). Then vectors $\mathbf{c}_1 = (\mathbf{f}, \mathbf{f}+\mathbf{g})$ and $\mathbf{c}_2 = (\mathbf{f}', \mathbf{f}'+\mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then $d(\mathbf{c}_1, \mathbf{c}_2)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

of this, this? So what is the minimum distance between g and g dash? This should be 2 raised to power m minus r.

(Refer Slide Time 27:38)



Reed-Muller code

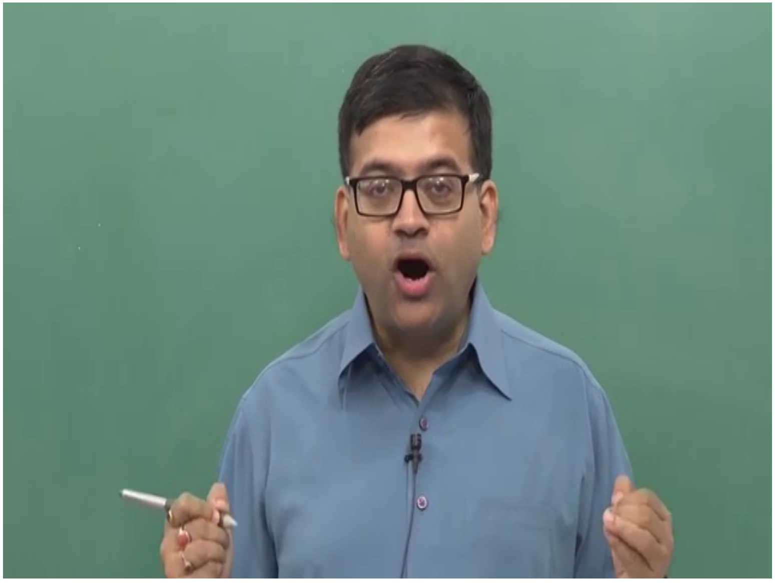
$2^{m-(r-1)} = 2^{m+1-r}$

- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ 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 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

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

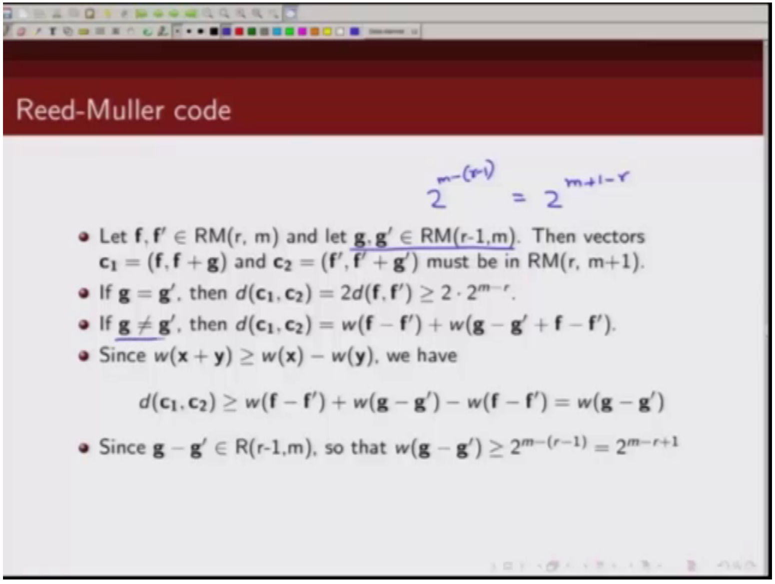
- Since $\mathbf{g}-\mathbf{g}' \in$ R(r-1,m), so that $w(\mathbf{g}-\mathbf{g}') \geq 2^{m-(r-1)} = 2^{m-r+1}$

(Refer Slide Time 27:40)



What is r? The order here is r minus 1. So this is r minus 1. So this is 2 raised to power m plus 1 minus r. So what we have shown is even when

(Refer Slide Time 27:43)



Reed-Muller code

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

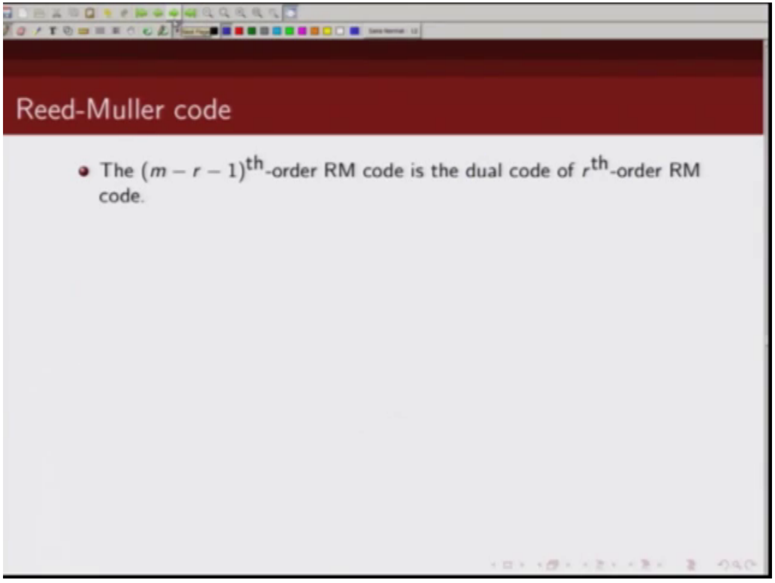
- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\underline{\mathbf{g}, \mathbf{g}' \in \text{RM(r-1,m)}}$. Then vectors $\mathbf{c_1} = (\mathbf{f}, \mathbf{f} + \mathbf{g})$ and $\mathbf{c_2} = (\mathbf{f}', \mathbf{f}' + \mathbf{g}')$ must be in RM(r, m+1).
- If $\mathbf{g} = \mathbf{g}'$, then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

g is not same as g dash, our minimum distance is still 2 raised to power m minus r plus 1. So now we have proved that minimum distance if, minimum distance of rth order Reed-Muller code of length 2 raised to power m plus 1 is basically given by this. So this will conclude the proof

(Refer Slide Time 28:11)

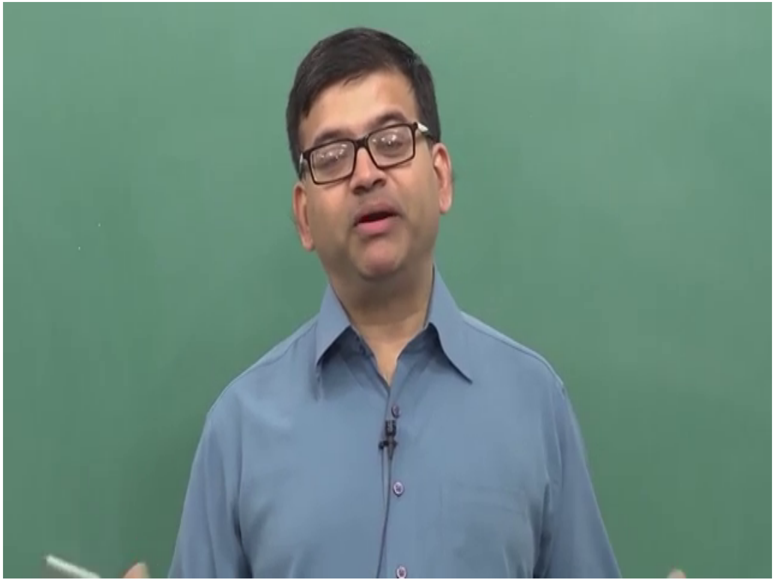


Reed-Muller code

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

using mathematical induction

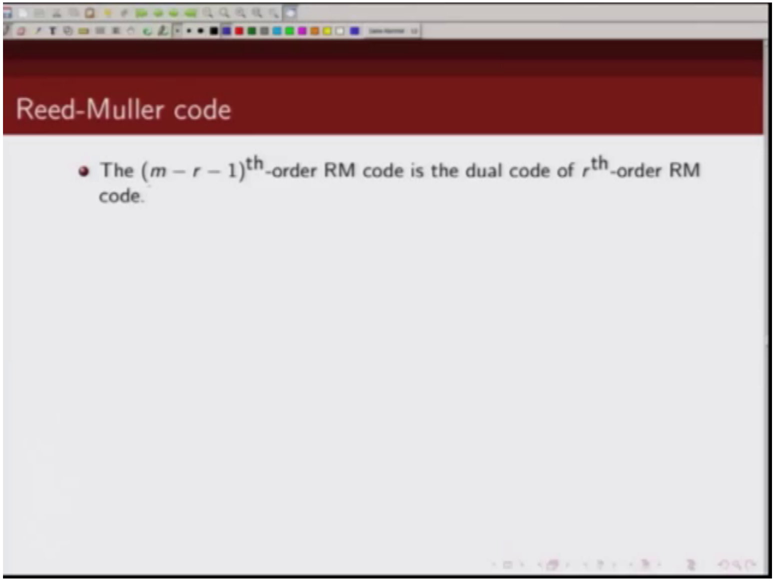
(Refer Slide Time 28:13)



that the minimum distance of Reed-Muller code is 2 raised to power m minus r.

The next result which we are going to show you is that

(Refer Slide Time 28:28)

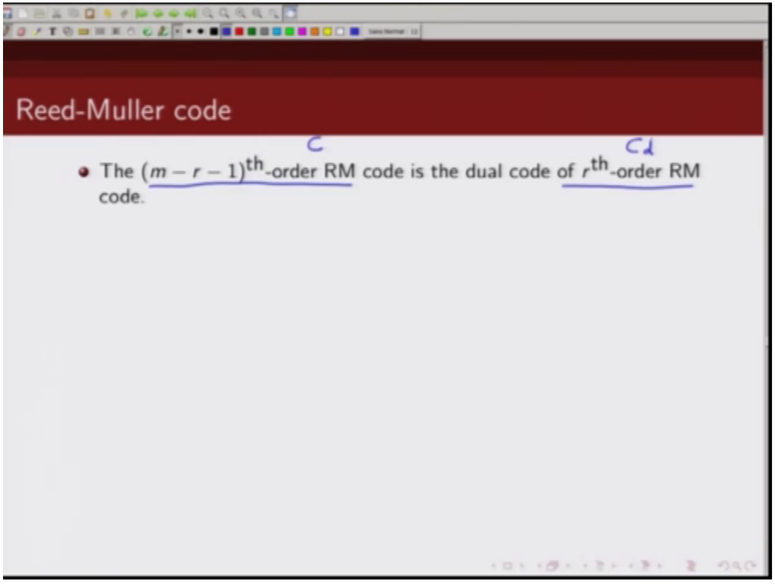


Reed-Muller code

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

m minus rth order Reed-Muller code is the dual code of rth order Reed-Muller code. So let's see. This is our original code and the dual code is given by this. Now what do we

(Refer Slide Time 28:43)



Reed-Muller code

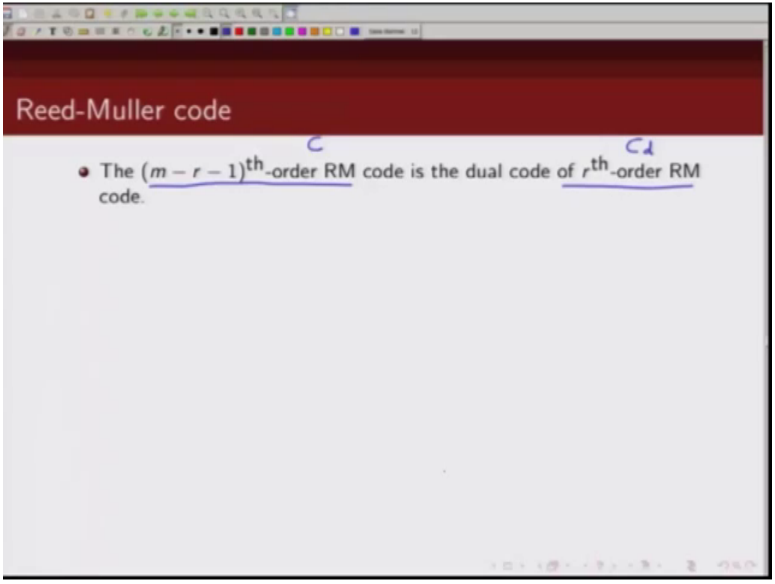
- The $\underbrace{(m-r-1)^{\text{th}}\text{-order RM}}_{}$ code is the dual code of $\underbrace{r^{\text{th}}\text{-order RM}}_{}$ code.

need to show for dual code?

(Refer Slide Time 28:45)

If we take a codeword from this code and if we take a codeword from the dual code they are orthogonal, right? So dot product should be zero. Another

(Refer Slide Time 28:58)

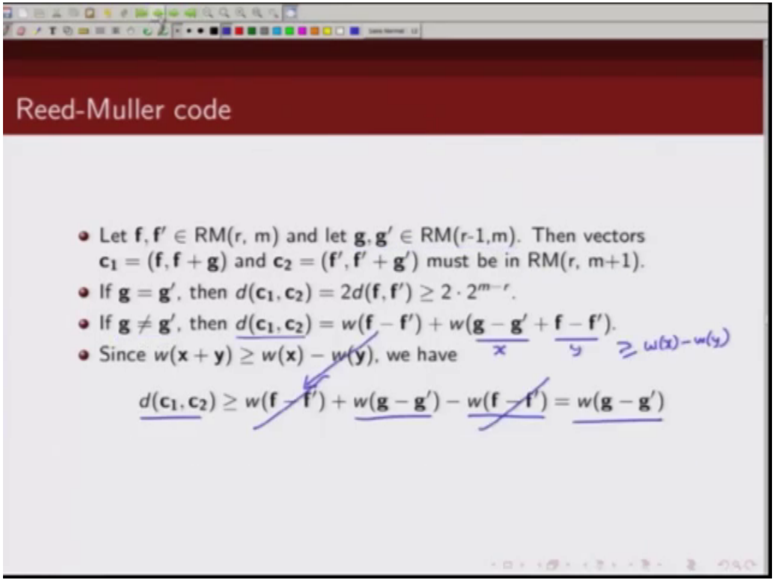


Reed-Muller code

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

point which I should mention here is

(Refer Slide Time 29:01)



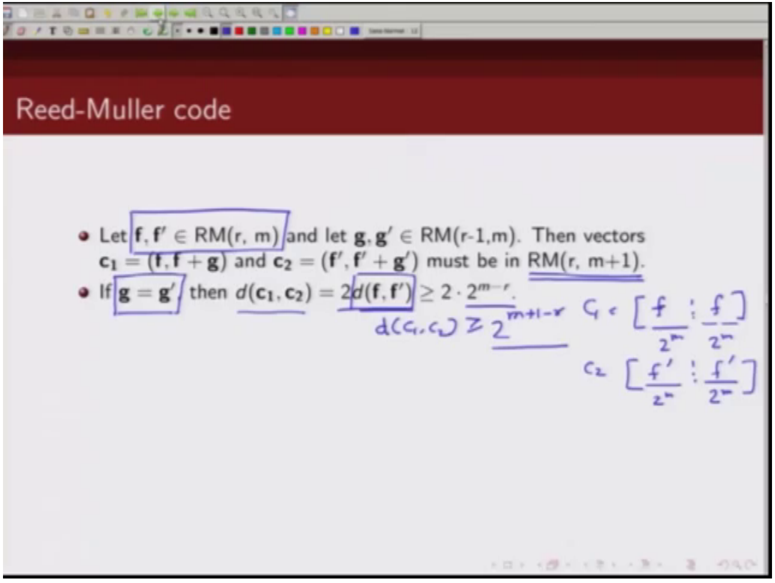
Reed-Muller code

- Let $\mathbf{f}, \mathbf{f}' \in$ RM(r, m) and let $\mathbf{g}, \mathbf{g}' \in$ 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 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 $\underline{d(\mathbf{c}_1, \mathbf{c}_2)} = w(\mathbf{f} - \mathbf{f}') + w(\underbrace{\mathbf{g} - \mathbf{g}'}_{x} + \underbrace{\mathbf{f} - \mathbf{f}'}_{y})$. $\geq w(x) - w(y)$
- Since $w(\mathbf{x} + \mathbf{y}) \geq w(\mathbf{x}) - w(\mathbf{y})$, we have

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

let's go back

(Refer Slide Time 29:02)



Reed-Muller code

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

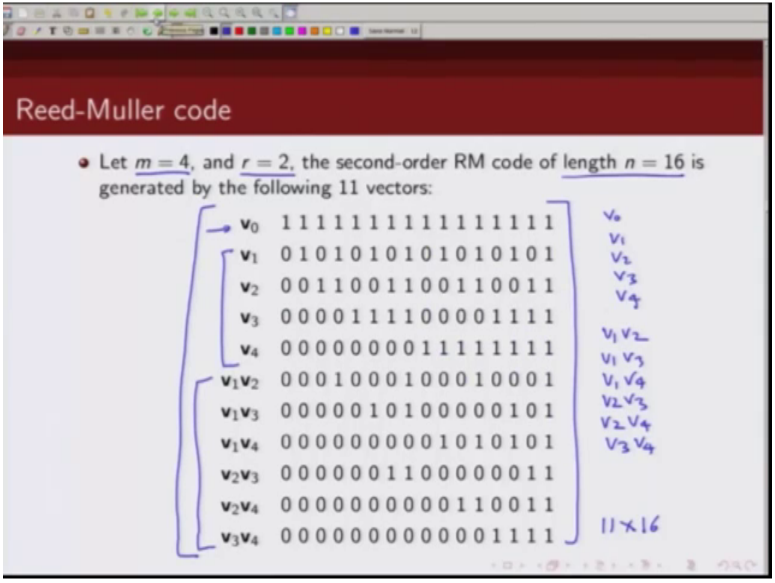
$d(c_1, c_2) \geq \underline{2^{m+1-r}}$

$c_1 \in \left[\underbrace{f}_{2^m} \vdots \underbrace{f}_{2^m}\right]$

$c_2 \left[\underbrace{f'}_{2^m} \vdots \underbrace{f'}_{2^m}\right]$

to our construction of Reed-Muller code here. Please note the way

(Refer Slide Time 29:09)



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:

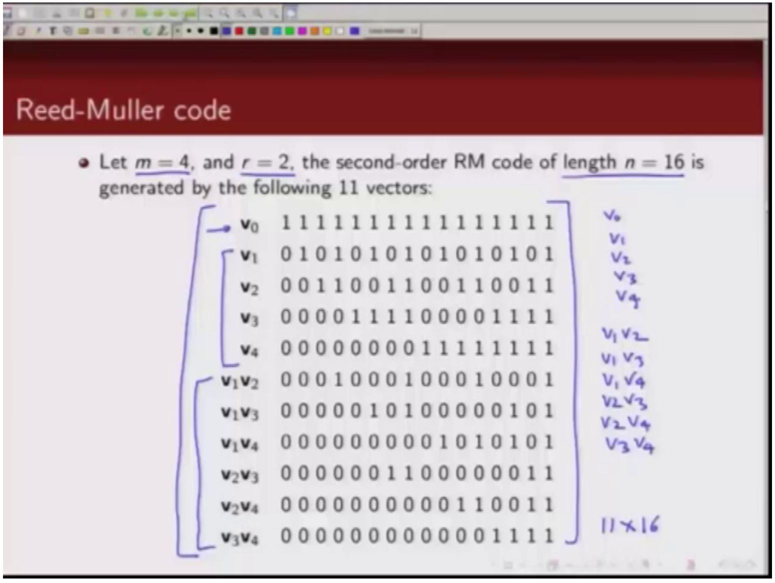
$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

these Boolean products are constructed. In fact we just proved also the minimum distance of the code is even. It is 2 raised to power

(Refer Slide Time 29:20)

m minus r. So minimum distance of Reed-Muller code is even. So Reed-Muller code would not have odd weight codewords.

(Refer Slide Time 29:28)



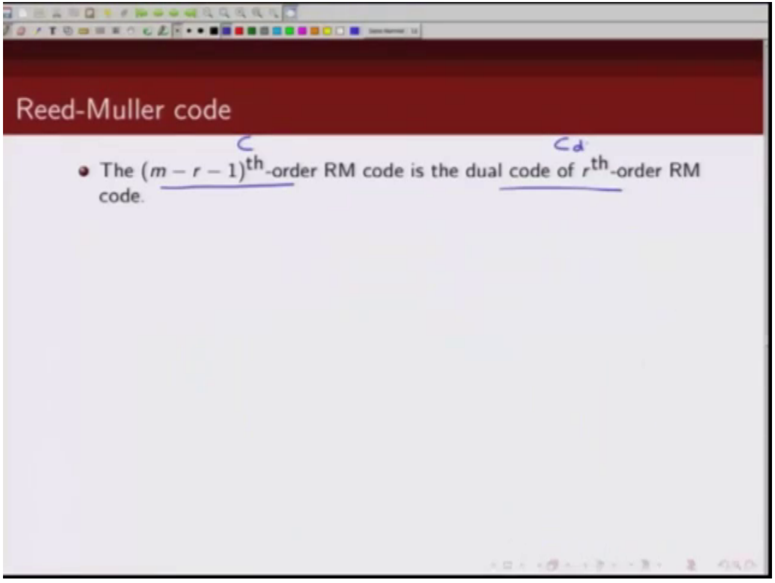
Reed-Muller code

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

$$\left[\begin{array}{l l} \mathbf{v}_0 & 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\ \mathbf{v}_1 & 0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1 \\ \mathbf{v}_2 & 0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1 \\ \mathbf{v}_3 & 0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1 \\ \mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1 \\ \mathbf{v}_1\mathbf{v}_2 & 0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1 \\ \mathbf{v}_1\mathbf{v}_3 & 0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1 \\ \mathbf{v}_1\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1 \\ \mathbf{v}_2\mathbf{v}_3 & 0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1\,1 \\ \mathbf{v}_2\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1 \\ \mathbf{v}_3\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1 \end{array}\right] \begin{array}{l} v_0 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_1v_2 \\ v_1v_3 \\ v_1v_4 \\ v_2v_3 \\ v_2v_4 \\ v_3v_4 \end{array} \quad 11\times 16$$

So now we will

(Refer Slide Time 29:30)

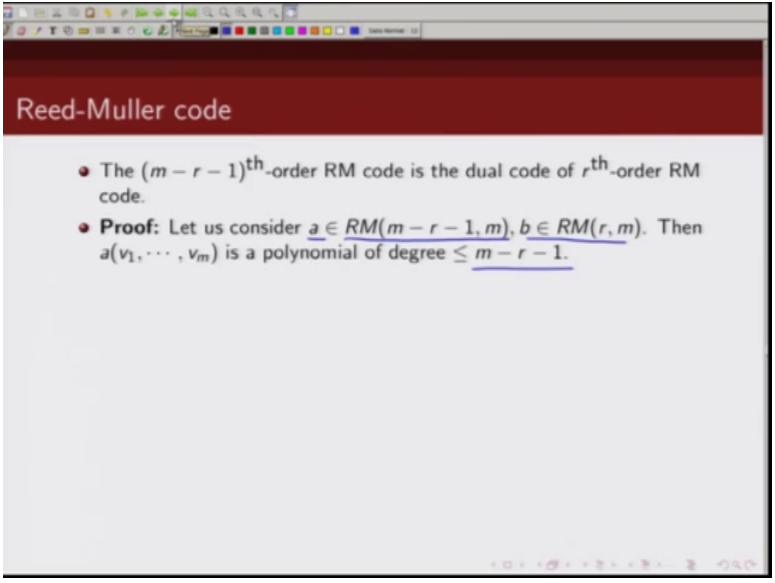


Reed-Muller code

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

show if we take a codeword from m minus r minus 1 nth order Reed-Muller code and if we take another codeword from rth order Reed-Muller code then they are orthogonal. That's the first thing we are going to prove. So let us

(Refer Slide Time 29:48)

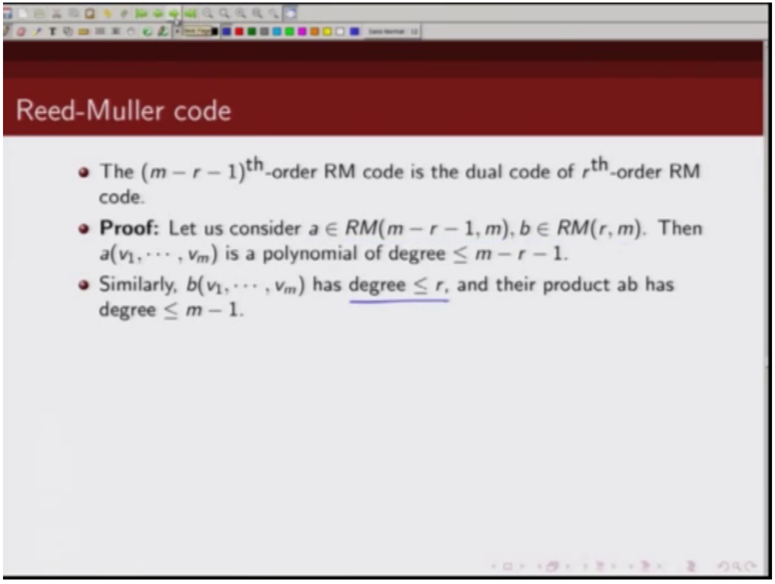


Reed-Muller code

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

consider codeword a which belongs to m minus r minus 1th order Reed-Muller code which is of length 2 raised to power m. And let us consider another Reed-Muller code b which is of order r and length 2 raised to power m. So a can be viewed as a polynomial of degree m minus r minus 1 or less and similarly the degree of the polynomial

(Refer Slide Time 30:19)

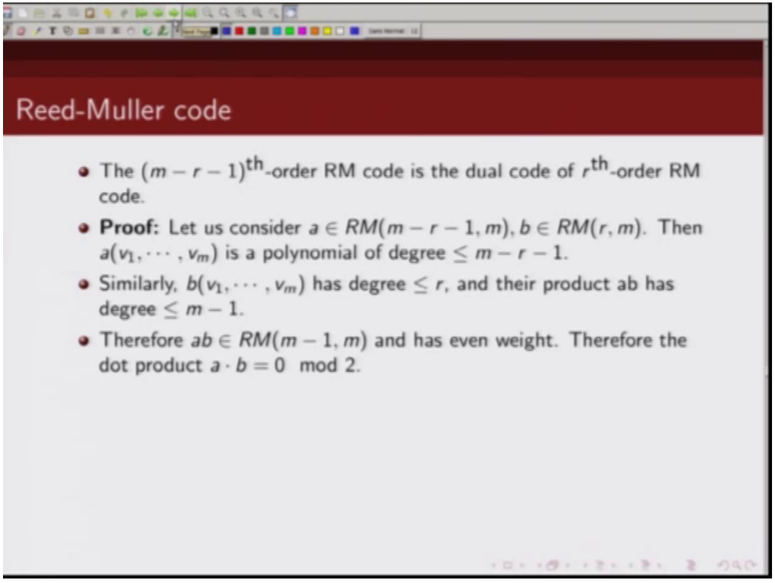


Reed-Muller code

- The $(m-r-1)^{th}$-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 $a(v_1,\cdots,v_m)$ is a polynomial of degree $\leq m-r-1$.
- Similarly, $b(v_1,\cdots,v_m)$ has $\underline{\text{degree} \leq r}$, and their product ab has degree $\leq m-1$.

b is less than equal to r. So if we consider

(Refer Slide Time 30:25)



Reed-Muller code

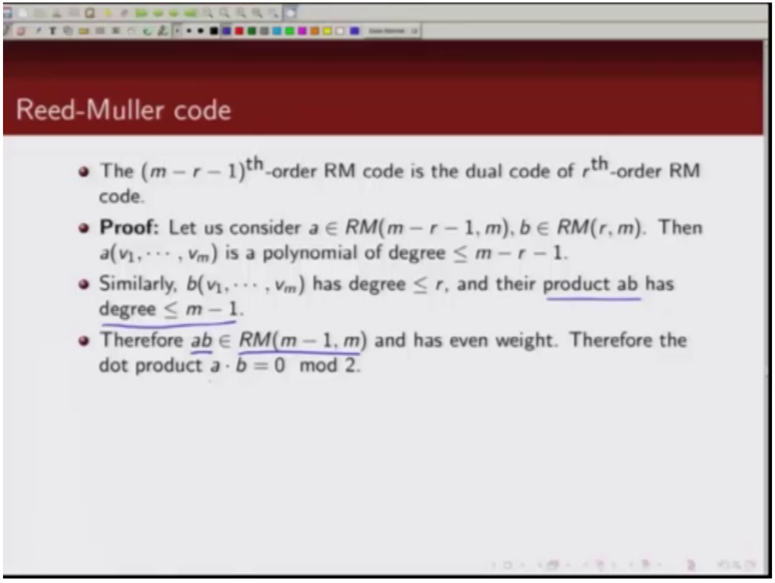
- The $(m-r-1)^{\text{th}}$-order RM code is the dual code of r^{th}-order RM code.
- **Proof:** Let us consider $a \in RM(m-r-1, m), b \in RM(r, m)$. Then $a(v_1, \cdots, v_m)$ is a polynomial of degree $\leq m-r-1$.
- Similarly, $b(v_1, \cdots, v_m)$ has degree $\leq r$, and their product ab has degree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

their product then this will be a polynomial of degree m minus r minus 1 plus r so that would be of degree less than equal to m minus 1. So then this product a and b will belong to a Reed-Muller code of order m minus 1 and this is of length 2 raised to power m. Now note that Reed-Muller code has only even weight codewords.

(Refer Slide Time 31:02)

So when we are considering this dot product a dot b,

(Refer Slide Time 31:08)



Reed-Muller code

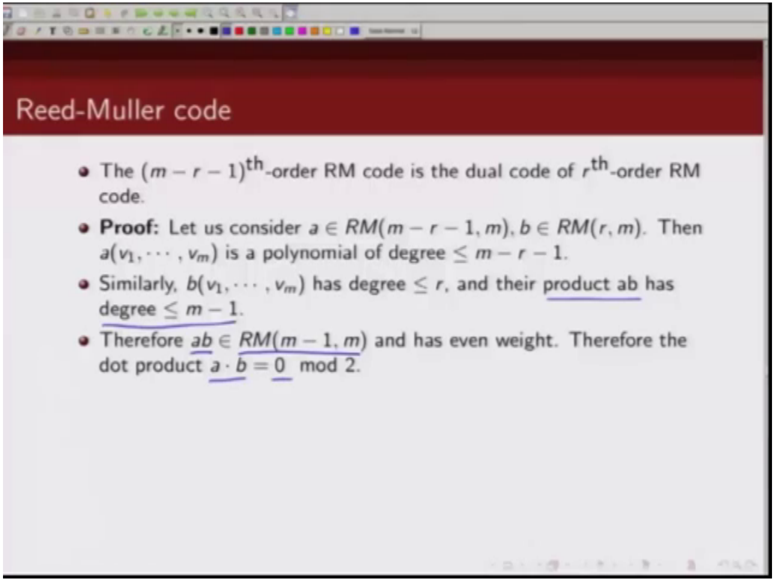
- The $(m-r-1)^{\text{th}}$-order RM code is the dual code of r^{th}-order RM code.
- **Proof:** Let us consider $a \in RM(m-r-1, m), b \in RM(r, m)$. Then $a(v_1, \cdots, v_m)$ is a polynomial of degree $\leq m-r-1$.
- Similarly, $b(v_1, \cdots, v_m)$ has degree $\leq r$, and their product ab has degree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

since Reed-Muller code

(Refer Slide Time 31:10)

has only even weight codeword then a dot b would be zero.

(Refer Slide Time 31:18)



Reed-Muller code

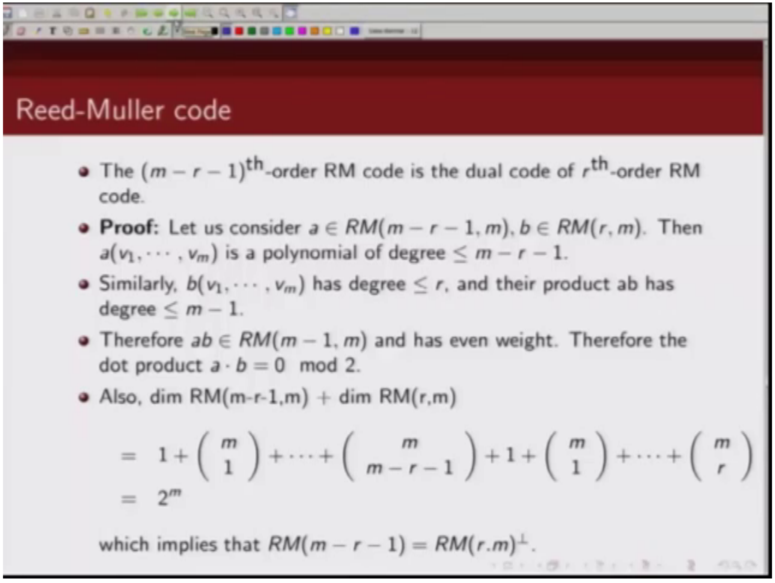
- The $(m-r-1)^{\text{th}}$-order RM code is the dual code of r^{th}-order RM code.
- **Proof:** Let us consider $a \in RM(m-r-1, m)$, $b \in RM(r, m)$. Then $a(v_1, \cdots, v_m)$ is a polynomial of degree $\leq m-r-1$.
- Similarly, $b(v_1, \cdots, v_m)$ has degree $\leq r$, and their product ab has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

So modulo 2 this would be zero. So in other words then what we have shown is if you take a codeword a which belongs to m minus r minus 1th order Reed-Muller code and if you take another codeword which belongs to rth order Reed-Muller code, then they are

(Refer Slide Time 31:42)

orthogonal to each other. Next we check the dimension

(Refer Slide Time 31:46)



Reed-Muller code

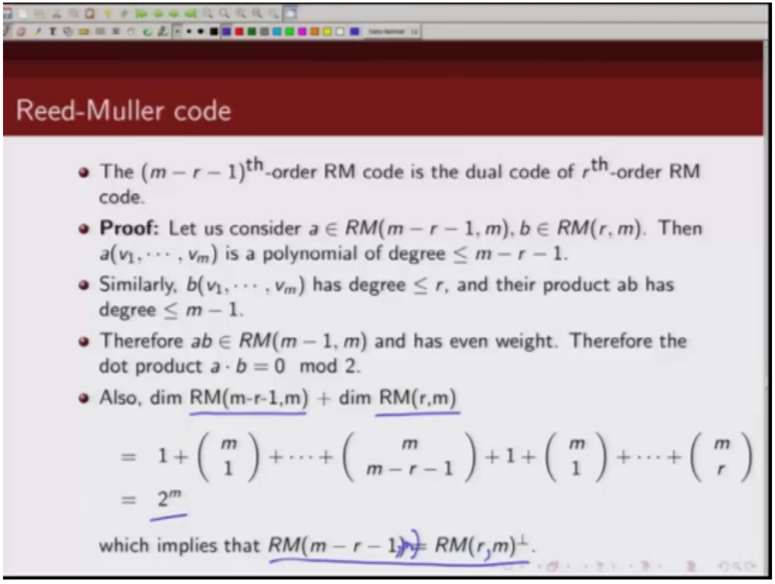
- The $(m-r-1)^{\text{th}}$-order RM code is the dual code of r^{th}-order RM code.
- **Proof:** Let us consider $a \in RM(m-r-1,m), b \in RM(r,m)$. Then $a(v_1,\cdots,v_m)$ is a polynomial of degree $\leq m-r-1$.
- Similarly, $b(v_1,\cdots,v_m)$ has degree $\leq r$, and their product ab has degree $\leq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

$$= 1 + \binom{m}{1} + \cdots + \binom{m}{m-r-1} + 1 + \binom{m}{1} + \cdots + \binom{m}{r}$$
$$= 2^m$$

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

of m minus r minus 1th order Reed-Muller code and rth order Reed-Muller code and we see that sum of their dimension is 2 raised to power m which is the length of the codeword. So this does prove then that m minus r minus 1th order Reed-Muller code, just, just write down m here is

(Refer Slide Time 32:17)



Reed-Muller code

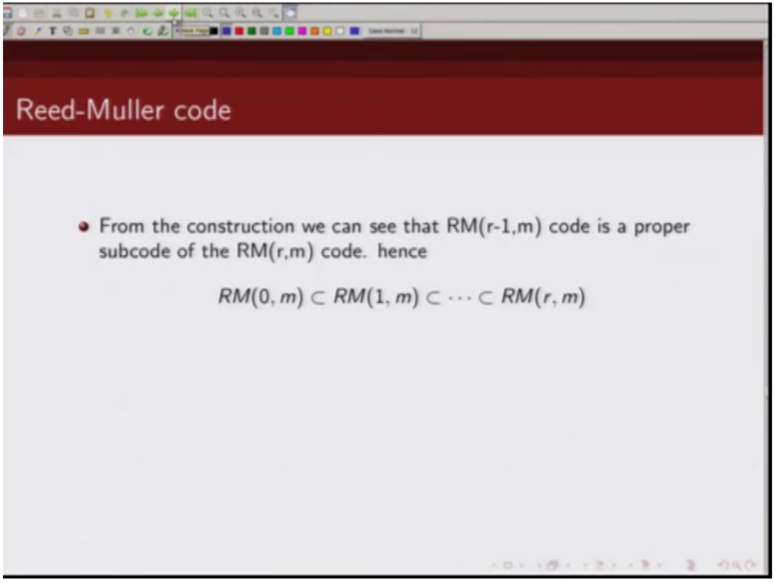
- The $(m-r-1)^{\text{th}}$-order RM code is the dual code of r^{th}-order RM code.
- **Proof:** Let us consider $a \in RM(m-r-1,m), b \in RM(r,m)$. Then $a(v_1,\cdots,v_m)$ is a polynomial of degree $\leq m-r-1$.
- Similarly, $b(v_1,\cdots,v_m)$ has degree $\leq r$, and their product ab has degree $\leq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

$$= 1 + \binom{m}{1} + \cdots + \binom{m}{m-r-1} + 1 + \binom{m}{1} + \cdots + \binom{m}{r}$$
$$= \underline{2^m}$$

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

dual 2 rth order Reed-Muller code. Now

(Refer Slide Time 32:23)



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

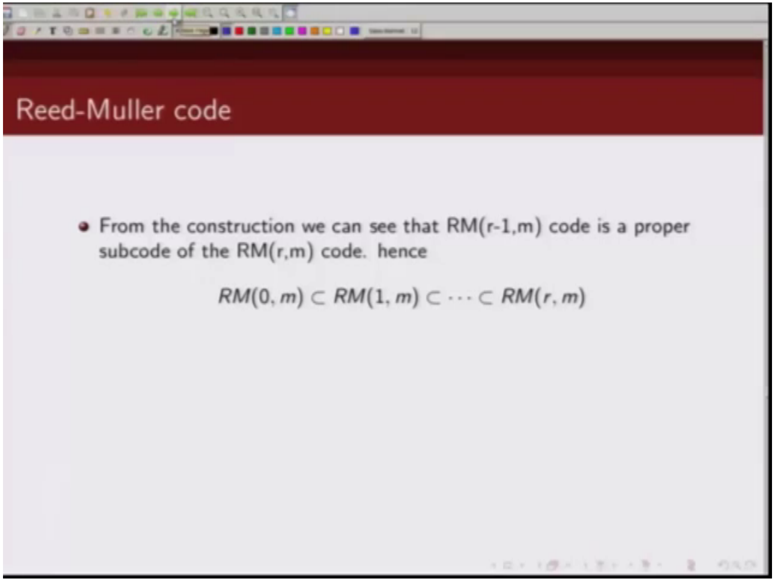
$$RM(0,m) \subset RM(1,m) \subset \cdots \subset RM(r,m)$$

let's see that

(Refer Slide Time 32:26)

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 that any

(Refer Slide Time 32:39)



Reed-Muller code

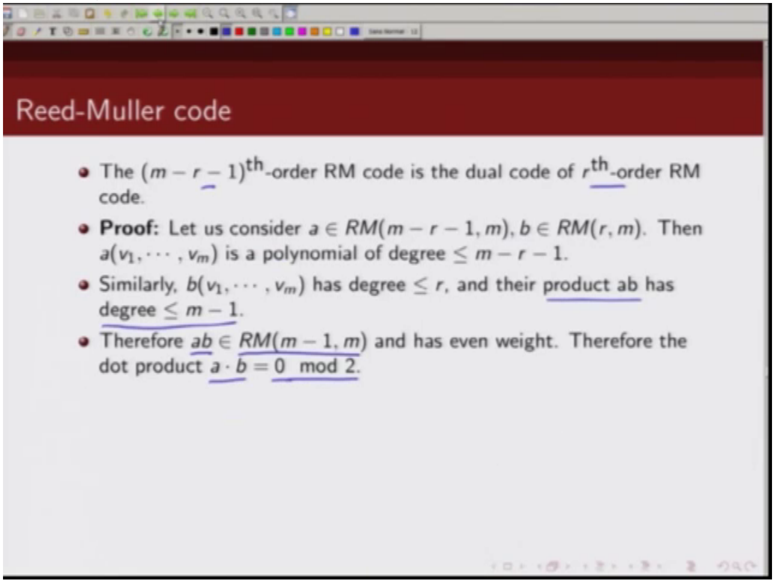
- From the construction we can see that RM(r-1,m) code is a proper subcode of the RM(r,m) code. hence

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

r minus 1 order Reed-Muller code is a proper subcode of an rth order Reed-Muller code.

And this is easy to see if

(Refer Slide Time 32:50)

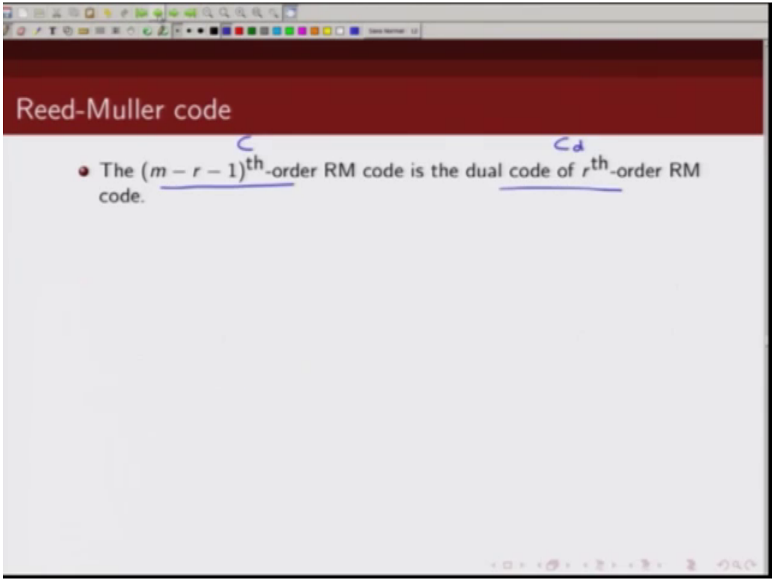


Reed-Muller code

- 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 $a(v_1,\cdots,v_m)$ is a polynomial of degree $\leq m-r-1$.
- Similarly, $b(v_1,\cdots,v_m)$ 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 $a \cdot b = 0 \mod 2$.

you noticed and go back to

(Refer Slide Time 32:52)

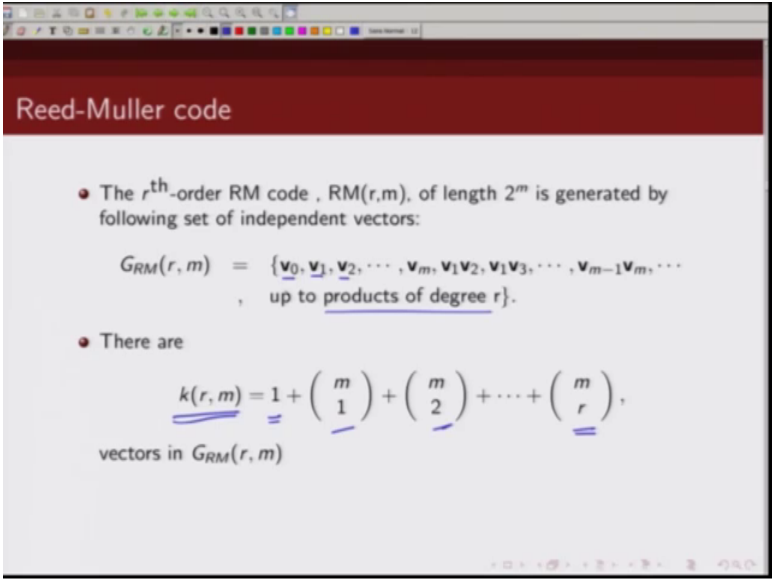


Reed-Muller code

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

our code construction. What was our generator matrix? Our generator matrix consists of

(Refer Slide Time 33:01)



Reed-Muller code

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

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

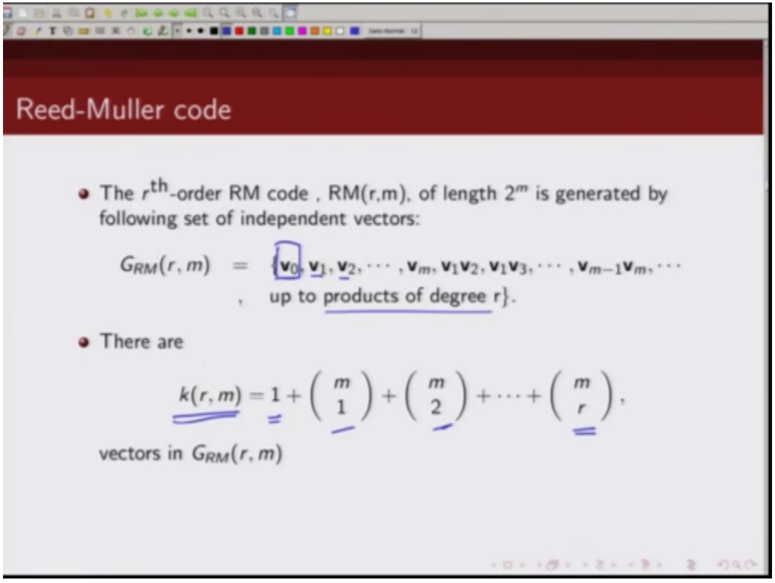
- There are

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

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

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

(Refer Slide Time 33:15)



Reed-Muller code

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

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

- There are

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

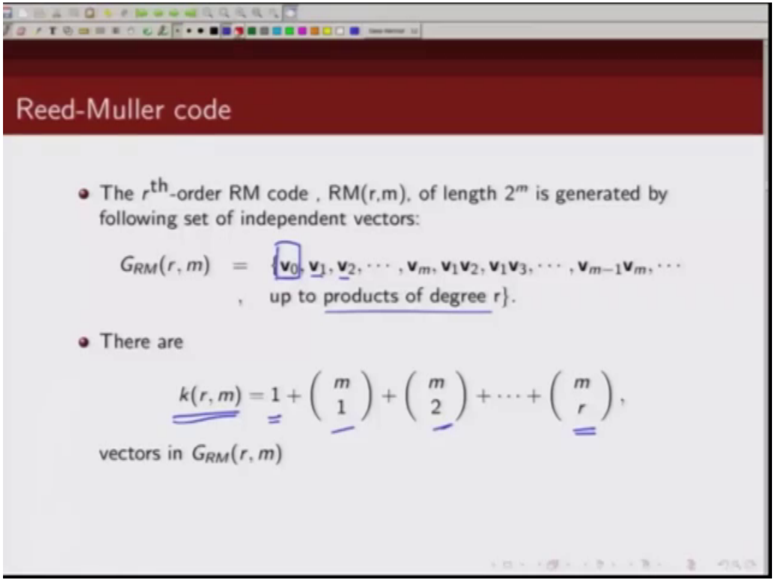
vectors in $G_{RM}(r,m)$

in the G matrix. If

(Refer Slide Time 33:19)

you are considering first order Reed-Muller code, it will have

(Refer Slide Time 33:24)



Reed-Muller code

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

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

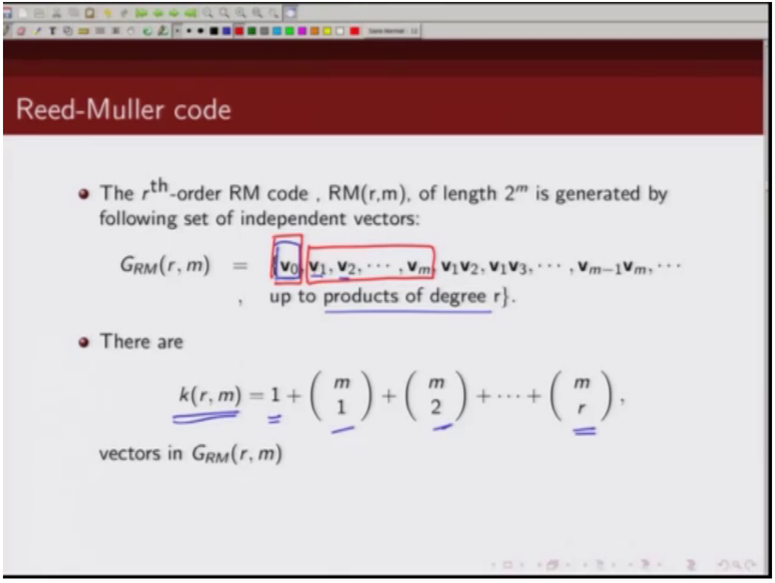
- There are

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

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

v 0 and it will also have v 1, v 2, v 3, v n.

(Refer Slide Time 33:31)



Reed-Muller code

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

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

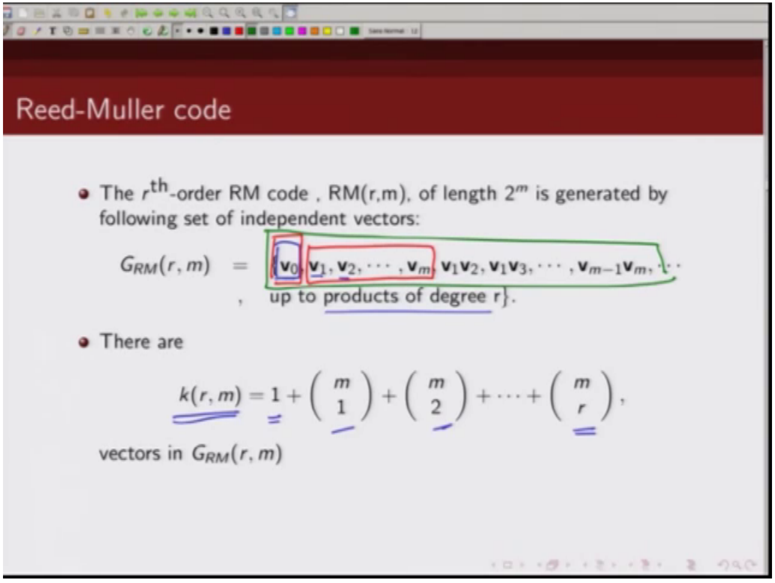
- There are

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

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

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

(Refer Slide Time 33:40)



Reed-Muller code

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

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

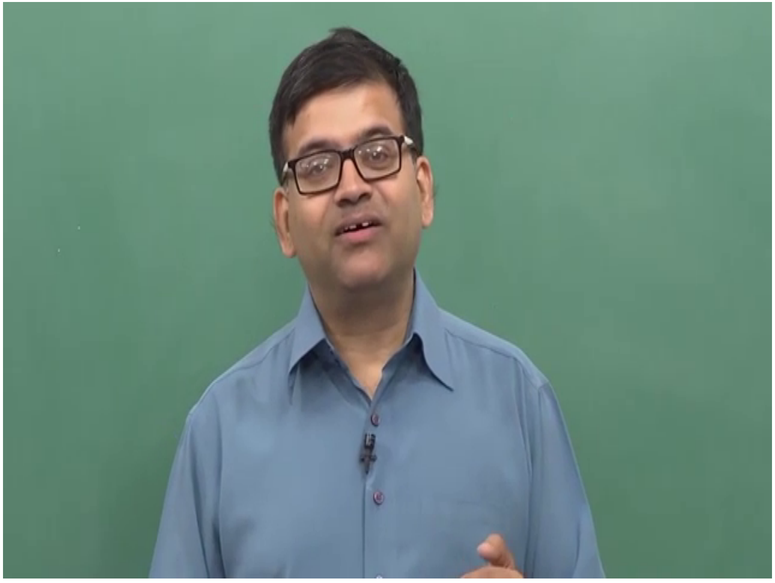
- There are

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

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

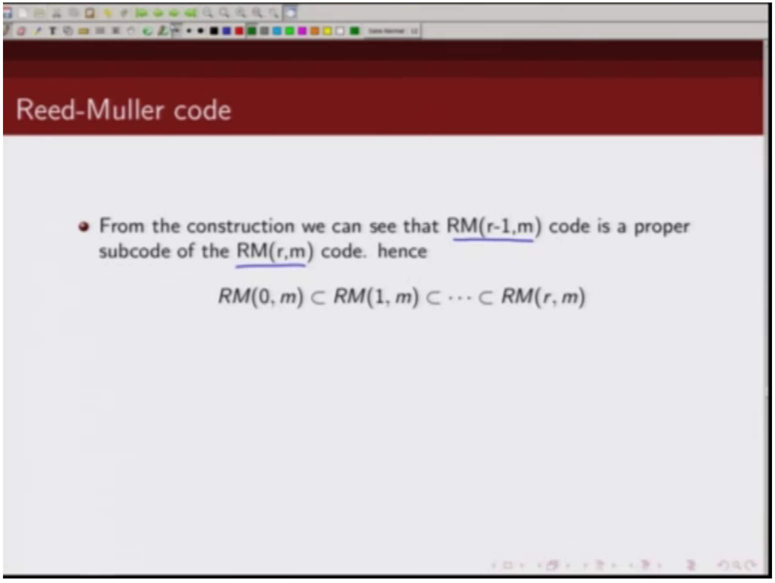
see that

(Refer Slide Time 33:42)



smaller order Reed-Muller code is already embedded in the larger order

(Refer Slide Time 33:59)



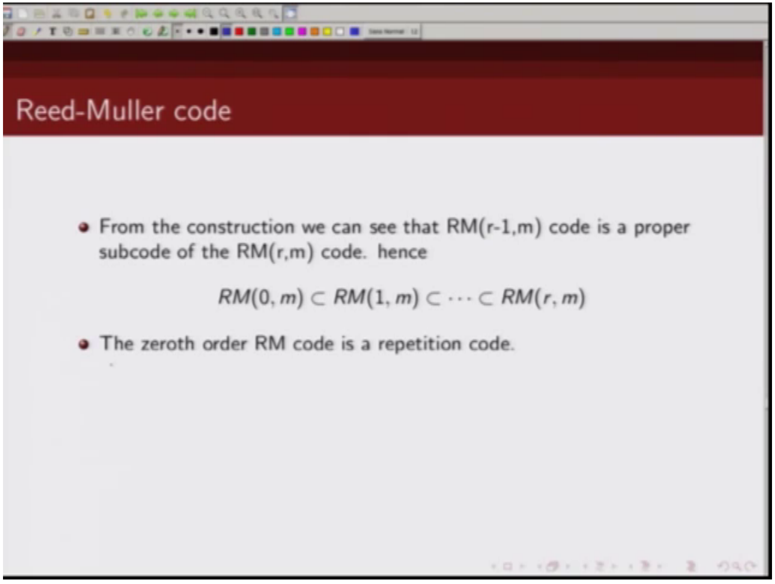
Reed-Muller code

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

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

Reed-Muller code. So you can, from the construction you can see that smaller order Reed-Muller code is essentially a proper subcode 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. The zeroth order

(Refer Slide Time 34:11)



Reed-Muller code

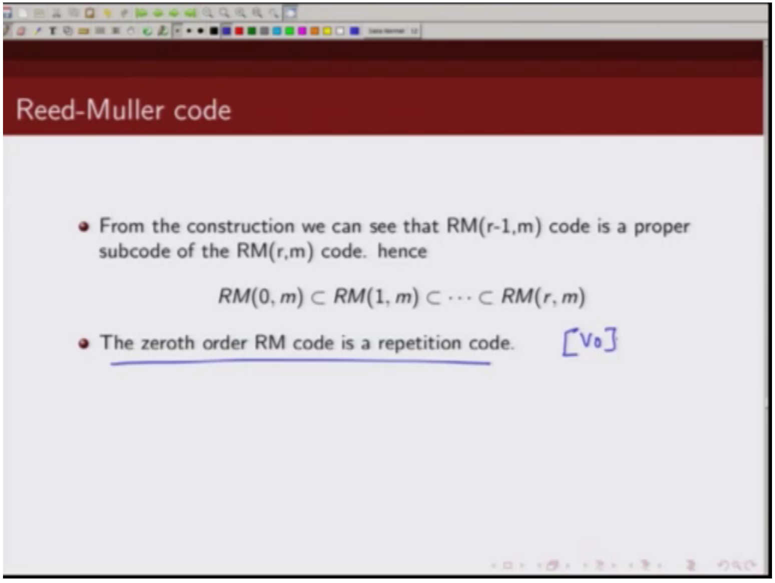
- From the construction we can see that RM(r-1,m) code is a proper subcode of the RM(r,m) code. hence

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

- The zeroth order RM code is a repetition code.

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 1's and that is precisely the

(Refer Slide Time 34:28)



Reed-Muller code

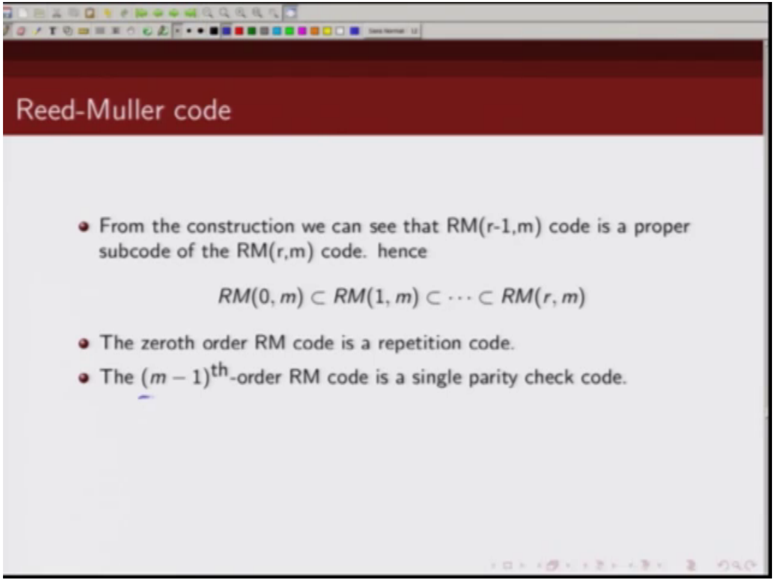
- From the construction we can see that RM(r-1,m) code is a proper subcode of the RM(r,m) code. hence

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

- The zeroth order RM code is a repetition code. $[V_0]$

generator matrix for repetition code. m minus

(Refer Slide Time 34:34)



Reed-Muller code

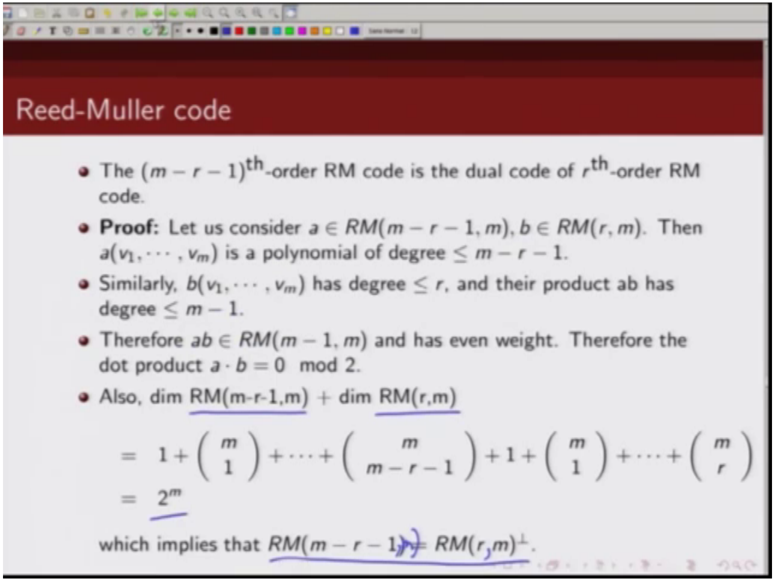
- From the construction we can see that RM(r-1,m) code is a proper subcode of the RM(r,m) code. hence

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

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

1th order repetition code, m minus 1th order Reed-Muller code is actually a single parity check code. Again this is easy to see. We can just use the results that we have proved.

(Refer Slide Time 34:47)



Reed-Muller code

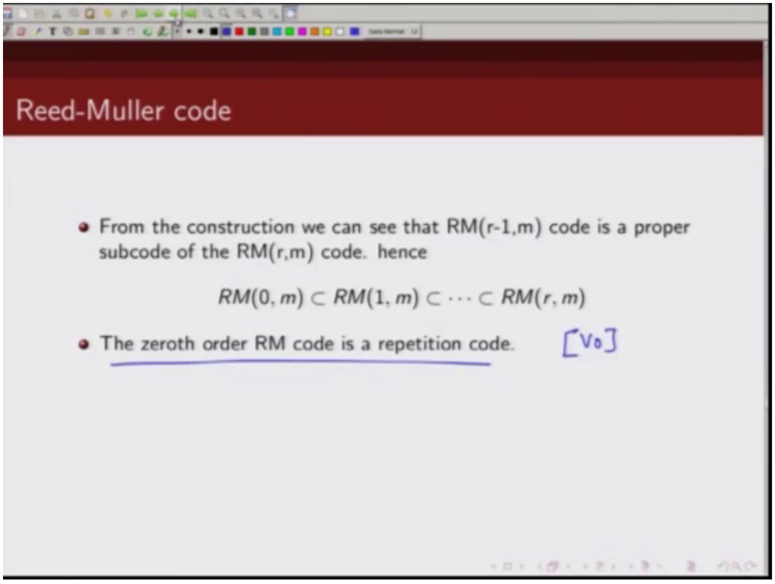
- The $(m-r-1)^{\text{th}}$-order RM code is the dual code of r^{th}-order RM code.
- **Proof:** Let us consider $a \in RM(m-r-1, m)$, $b \in RM(r, m)$. Then $a(v_1, \cdots, v_m)$ is a polynomial of degree $\leq m-r-1$.
- Similarly, $b(v_1, \cdots, v_m)$ has degree $\leq r$, and their product ab has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

$$= 1 + \binom{m}{1} + \cdots + \binom{m}{m-r-1} + 1 + \binom{m}{1} + \cdots + \binom{m}{r}$$
$$= \underline{2^m}$$

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

We know that m minus r minus 1th order Reed-Muller code is dual to the rth order Reed-Muller code. So if r is let's say zero, then it is dual to m minus 1th order Reed-Muller code. So zeroth order

(Refer Slide Time 35:05)



Reed-Muller code

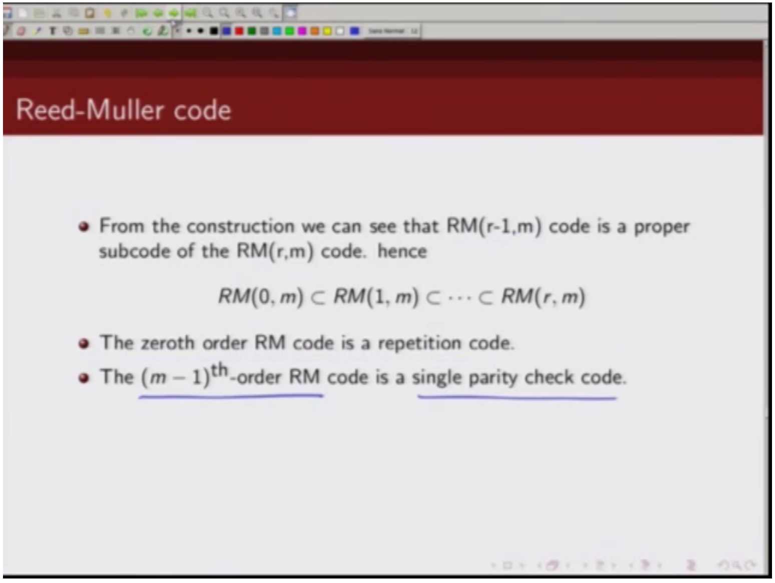
- From the construction we can see that RM(r-1,m) code is a proper subcode of the RM(r,m) code. hence

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

- $\underline{\text{The zeroth order RM code is a repetition code.}}$ $[v_0]$

Reed-Muller

(Refer Slide Time 35:06)



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

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

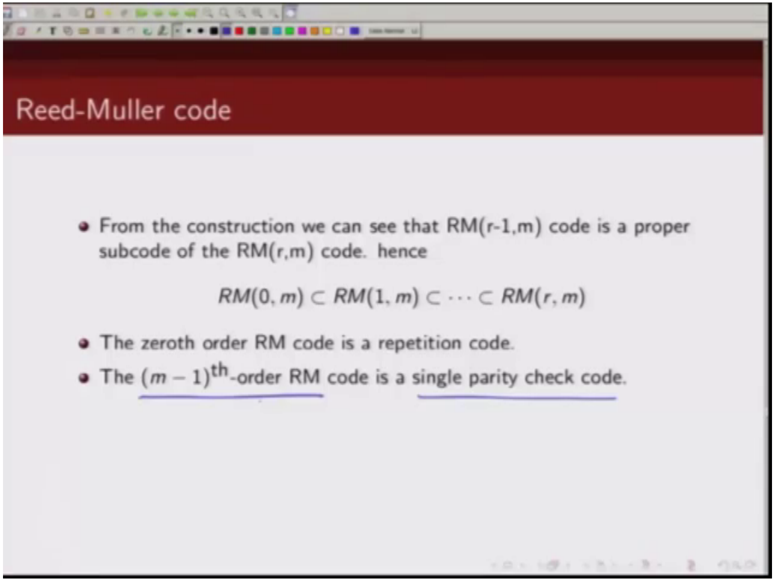
- The zeroth order RM code is a repetition code.
- The $\underline{(m-1)^{th}\text{-order}}$ RM code is a $\underline{\text{single parity check code}}$.

code is dual to m minus 1th order Reed-Muller code. And what is the dual

(Refer Slide Time 35:13)

of a repetition code? It is a single parity check code. So m minus 1th order Reed-Muller code

(Refer Slide Time 35:21)



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

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

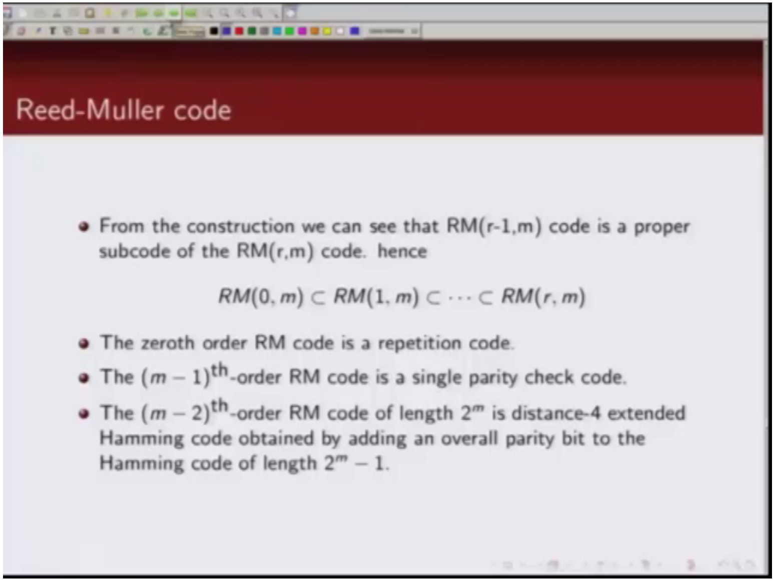
- The zeroth order RM code is a repetition code.
- The $\underline{(m-1)^{\text{th}}\text{-order RM}}$ code is a $\underline{\text{single parity check code}}$.

is nothing but a single parity check

(Refer Slide Time 35:24)

code. Similarly m minus 2 order Reed-Muller code is our extended

(Refer Slide Time 35:33)



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

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

- 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 $2^m - 1$.

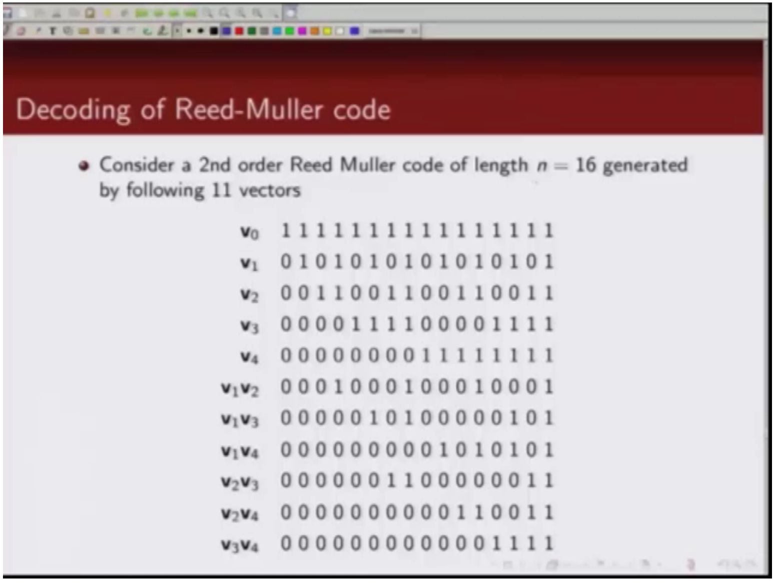
Hamming code which we just talked about in the last lecture. So let's discuss how

(Refer Slide Time 35:42)

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 Reed-Muller code with parameter m

(Refer Slide Time 36:00)



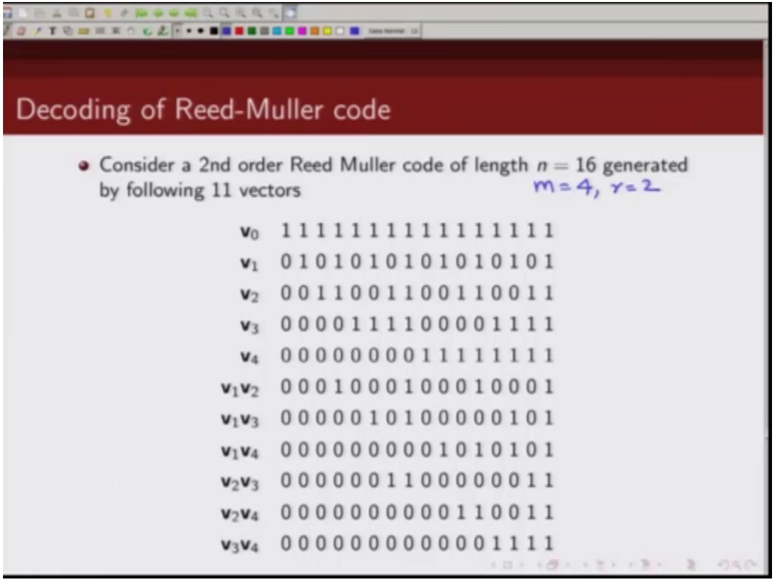
Decoding of Reed-Muller code

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

$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

equal to 4 and r equal to 2. So in other words

(Refer Slide Time 36:06)



Decoding of Reed-Muller code

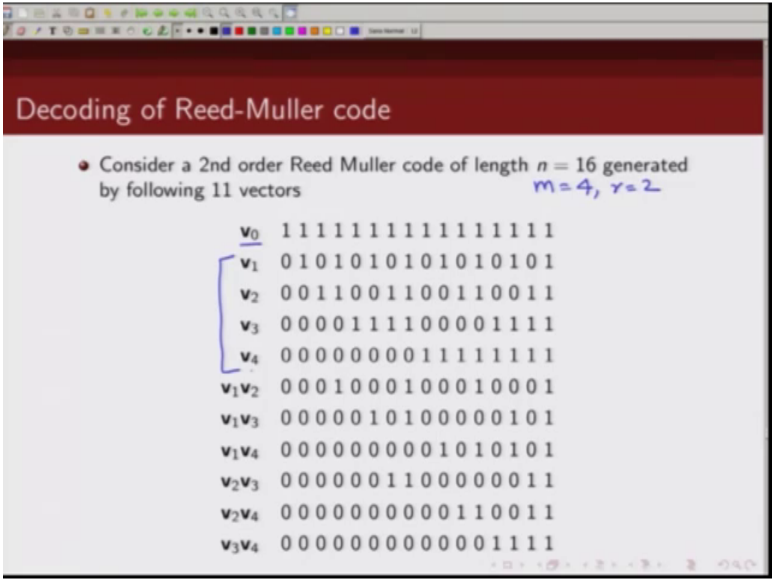
- Consider a 2nd order Reed Muller code of length $n = 16$ generated by following 11 vectors

$m = 4, r = 2$

$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

the generator matrix will then consist of v 0 all first order v i's

(Refer Slide Time 36:15)



Decoding of Reed-Muller code

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

$m=4, r=2$

$\mathbf{v}_0$ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

$\mathbf{v}_1$ 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

$\mathbf{v}_2$ 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1

$\mathbf{v}_3$ 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1

$\mathbf{v}_4$ 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

$\mathbf{v}_1\mathbf{v}_2$ 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1

$\mathbf{v}_1\mathbf{v}_3$ 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1

$\mathbf{v}_1\mathbf{v}_4$ 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1

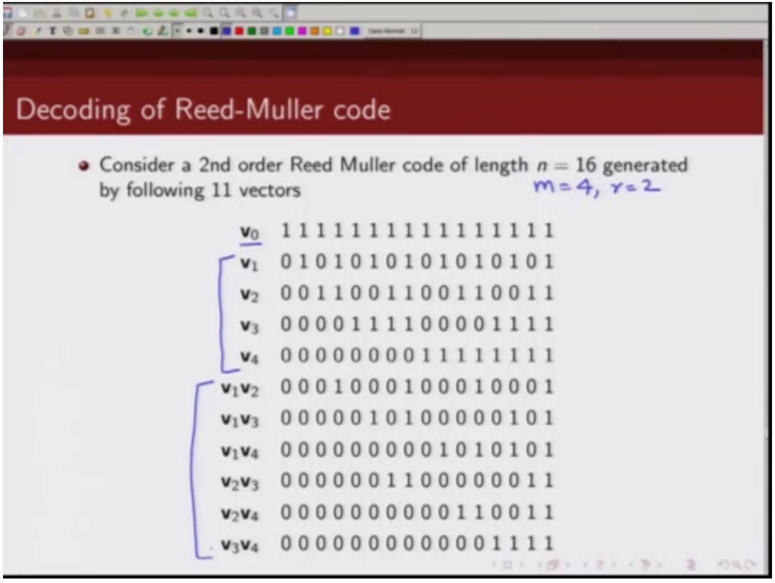
$\mathbf{v}_2\mathbf{v}_3$ 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1

$\mathbf{v}_2\mathbf{v}_4$ 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1

$\mathbf{v}_3\mathbf{v}_4$ 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

and these Boolean

(Refer Slide Time 36:18)



Decoding of Reed-Muller code

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

$m=4, r=2$

$\mathbf{v}_0$ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

$\mathbf{v}_1$ 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1

$\mathbf{v}_2$ 0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1

$\mathbf{v}_3$ 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1

$\mathbf{v}_4$ 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1

$\mathbf{v}_1\mathbf{v}_2$ 0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1

$\mathbf{v}_1\mathbf{v}_3$ 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1

$\mathbf{v}_1\mathbf{v}_4$ 0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1

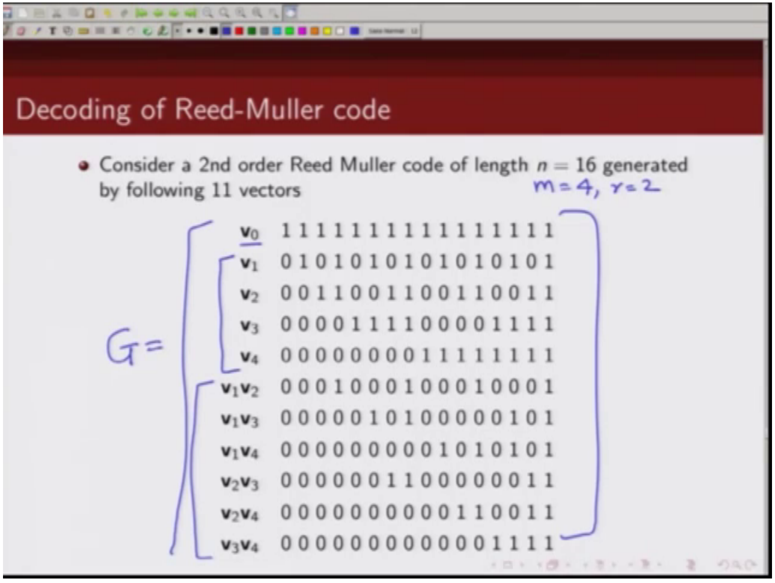
$\mathbf{v}_2\mathbf{v}_3$ 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1

$\mathbf{v}_2\mathbf{v}_4$ 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1

$\mathbf{v}_3\mathbf{v}_4$ 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

product of order 2. 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 also know how to compute the Boolean product. So this is essentially our generator matrix G of a

(Refer Slide Time 36:42)



Decoding of Reed-Muller code

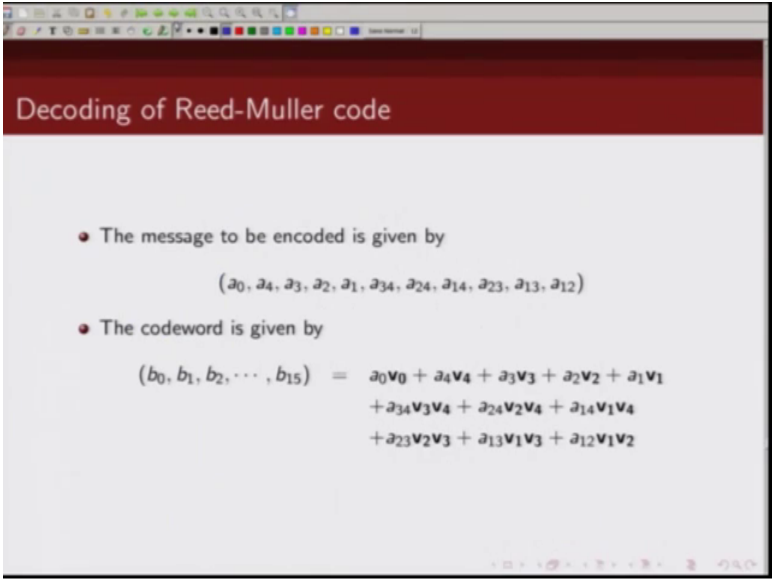
- Consider a 2nd order Reed Muller code of length $n = 16$ generated by following 11 vectors

$m=4, r=2$

$$G = \begin{array}{c|c} \mathbf{v}_0 & 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\ \mathbf{v}_1 & 0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1 \\ \mathbf{v}_2 & 0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1 \\ \mathbf{v}_3 & 0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1 \\ \mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1 \\ \mathbf{v}_1\mathbf{v}_2 & 0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1 \\ \mathbf{v}_1\mathbf{v}_3 & 0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1 \\ \mathbf{v}_1\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1 \\ \mathbf{v}_2\mathbf{v}_3 & 0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1\,1 \\ \mathbf{v}_2\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1 \\ \mathbf{v}_3\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1 \end{array}$$

2 4 Reed-Muller code.

(Refer Slide Time 36:47)



Decoding of Reed-Muller code

- The message to be encoded is given by

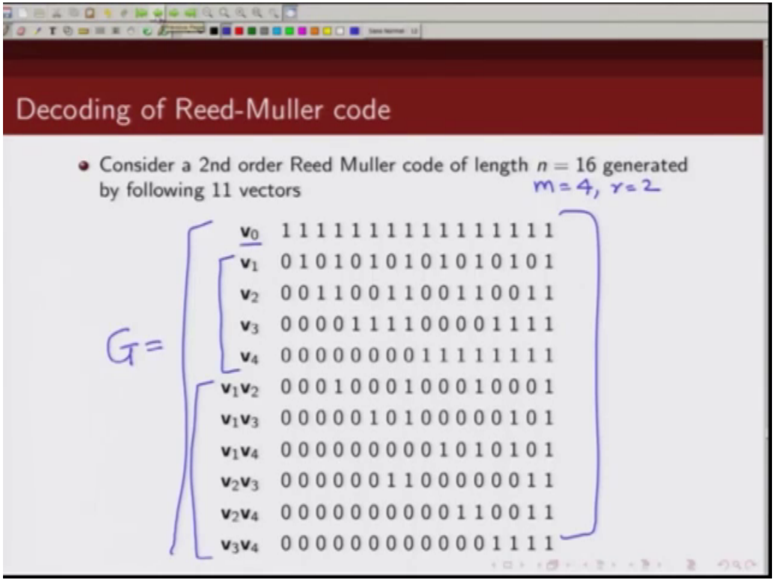
$$(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$$

- The codeword is given by

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

Now the message that we want to encode, let's call it a 0 a 4 a 3, this is how we are denoting the message that we are going to encode and since the rows of our generator matrix are given by

(Refer Slide Time 37:05)



Decoding of Reed-Muller code

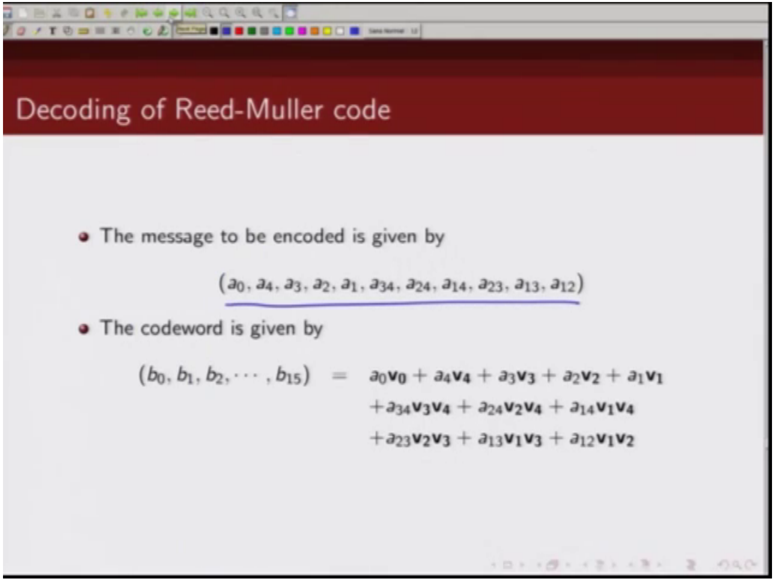
- Consider a 2nd order Reed Muller code of length $n = 16$ generated by following 11 vectors

$m=4, r=2$

$$
G = \left[\begin{array}{ll}
\mathbf{v}_0 & 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\
\mathbf{v}_1 & 0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1 \\
\mathbf{v}_2 & 0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1 \\
\mathbf{v}_3 & 0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1 \\
\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1 \\
\mathbf{v}_1\mathbf{v}_2 & 0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1 \\
\mathbf{v}_1\mathbf{v}_3 & 0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1 \\
\mathbf{v}_1\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1 \\
\mathbf{v}_2\mathbf{v}_3 & 0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1\,1 \\
\mathbf{v}_2\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1 \\
\mathbf{v}_3\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1
\end{array}\right]
$$

v 0, v 1, v 2, v 3, v 4 and this, so our codeword would be linear

(Refer Slide Time 37:11)



Decoding of Reed-Muller code

- The message to be encoded is given by

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

- The codeword is given by

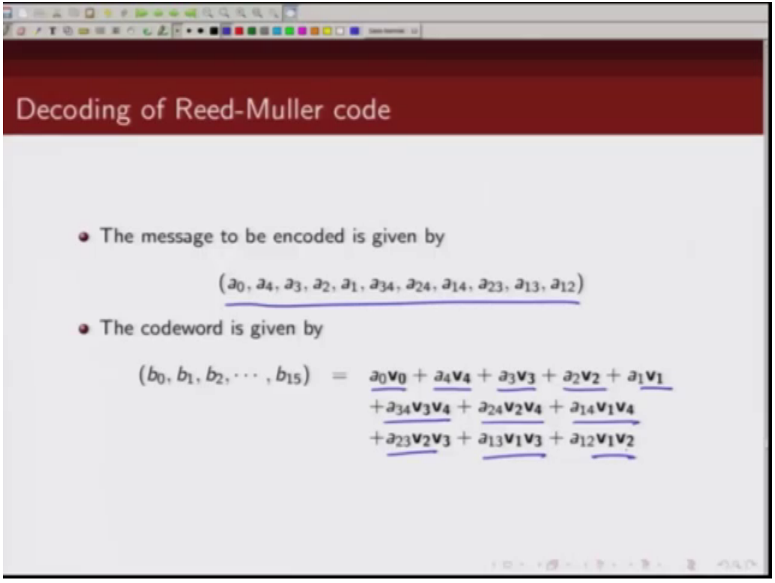
$$
\begin{aligned}
(b_0, b_1, b_2, \cdots, b_{15}) = {} & a_0\mathbf{v_0} + a_4\mathbf{v_4} + a_3\mathbf{v_3} + a_2\mathbf{v_2} + a_1\mathbf{v_1} \\
& + a_{34}\mathbf{v_3v_4} + a_{24}\mathbf{v_2v_4} + a_{14}\mathbf{v_1v_4} \\
& + a_{23}\mathbf{v_2v_3} + a_{13}\mathbf{v_1v_3} + a_{12}\mathbf{v_1v_2}
\end{aligned}
$$

combination of rows of the generator matrix. So that we are writing denoting by a 0 v 0 plus a 4 v 4 a 3 v 3 and similarly a 3 4 v 3 v 4 a 2 4 v 2 v 4. So this is how, this is linear combination of these

(Refer Slide Time 37:30)

eleven rows of this generator matrix. That's how we will generate

(Refer Slide Time 37:33)



Decoding of Reed-Muller code

- The message to be encoded is given by

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

- The codeword is given by

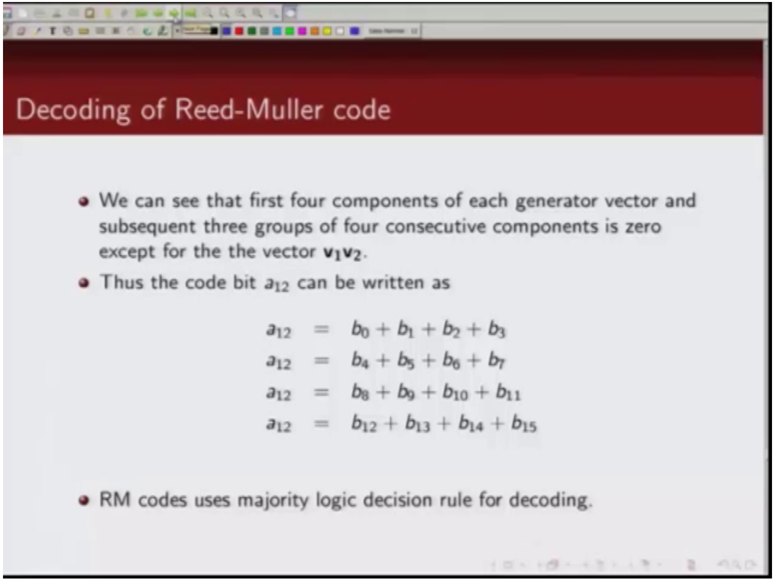
$$\begin{aligned}(b_0, b_1, b_2, \cdots, b_{15}) &= \underline{a_0\mathbf{v_0}} + \underline{a_4\mathbf{v_4}} + \underline{a_3\mathbf{v_3}} + \underline{a_2\mathbf{v_2}} + a_1\underline{\mathbf{v_1}}\\ &+\underline{a_{34}\mathbf{v_3}\mathbf{v_4}} + \underline{a_{24}\mathbf{v_2}\mathbf{v_4}} + \underline{a_{14}\mathbf{v_1}\mathbf{v_4}}\\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

our codewords. So this 16 length codeword is basically linear combination of these rows of this generator

(Refer Slide Time 37:43)

matrix. Now we

(Refer Slide Time 37:46)



Decoding of Reed-Muller code

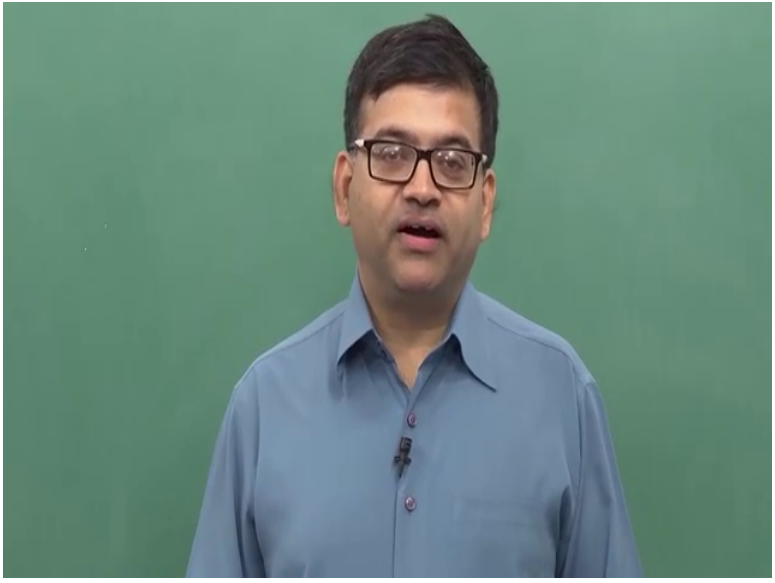
- We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the the vector $\mathbf{v_1 v_2}$.
- Thus the code bit a_{12} can be written as

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

- RM codes uses majority logic decision rule for decoding.

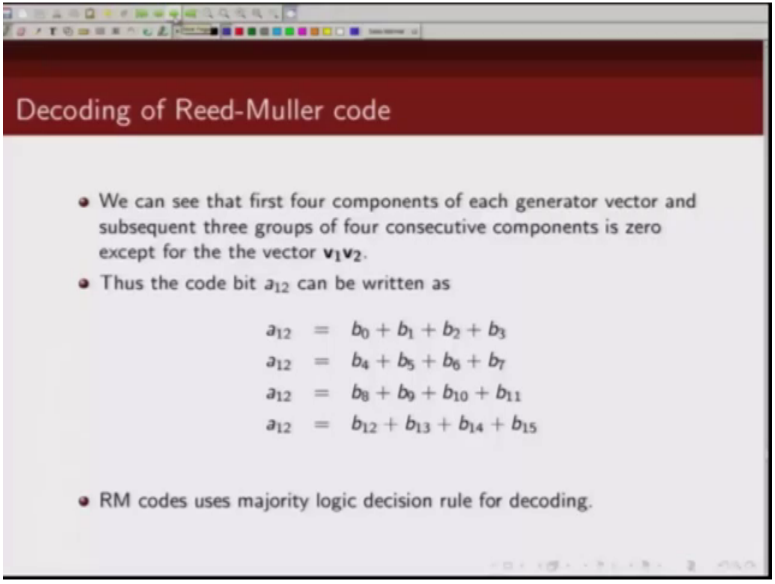
will spend some time looking at the generator matrix and we will use some

(Refer Slide Time 37:52)



observations from the generator matrix to decode our code. So what are these observations? So first thing we will see if we can,

(Refer Slide Time 38:01)



Decoding of Reed-Muller code

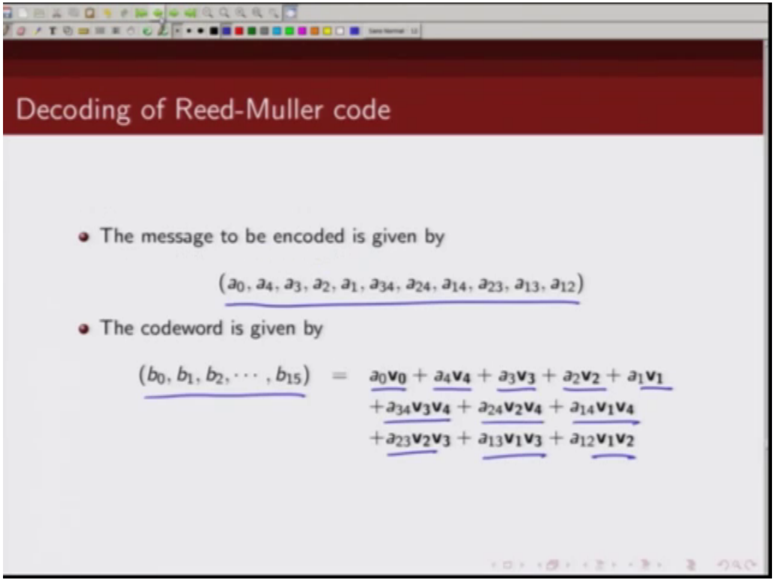
- We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the the vector $\mathbf{v}_1\mathbf{v}_2$.
- Thus the code bit a_{12} can be written as

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

- RM codes uses majority logic decision rule for decoding.

if we see the first four components of each generator vector and subsequent groups of 3 groups of 4 consecutive components they are zero except for vector v 1 v 2. What do I mean by that?

(Refer Slide Time 38:20)



Decoding of Reed-Muller code

- The message to be encoded is given by

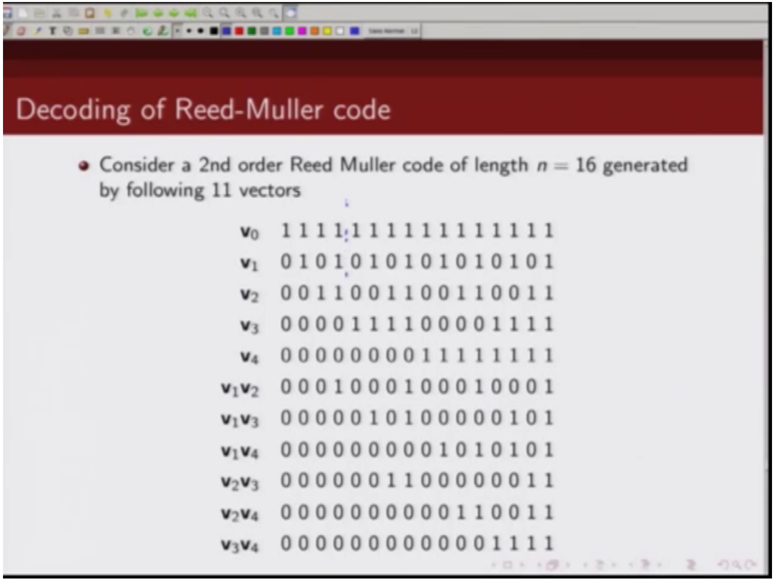
$$\underline{(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})}$$

- The codeword is given by

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

So let's look at

(Refer Slide Time 38:22)



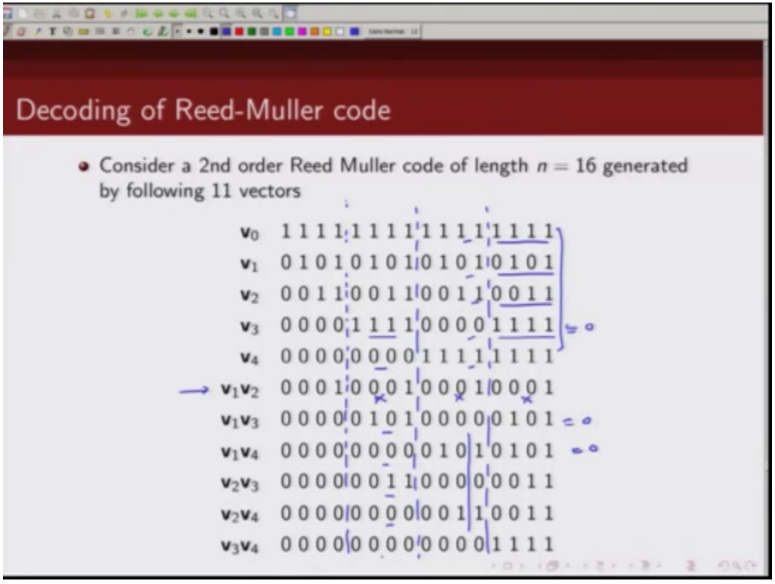
Decoding of Reed-Muller code

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

$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

this, group of four. This is group of four. This is group of four. So what I am saying is if you look at this group of four, and if you add them up. Just look at this first group of four. This will be zero, sum will be zero; zero, zero, zero, zero this is 1. 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 zeroes. Similarly this is not zero. These are all, if you add up these, they are all zeroes, one plus one, one plus one plus one, these are all zeroes. Same here, one plus one zero, one plus one zero. So if you look at

(Refer Slide Time 39:32)



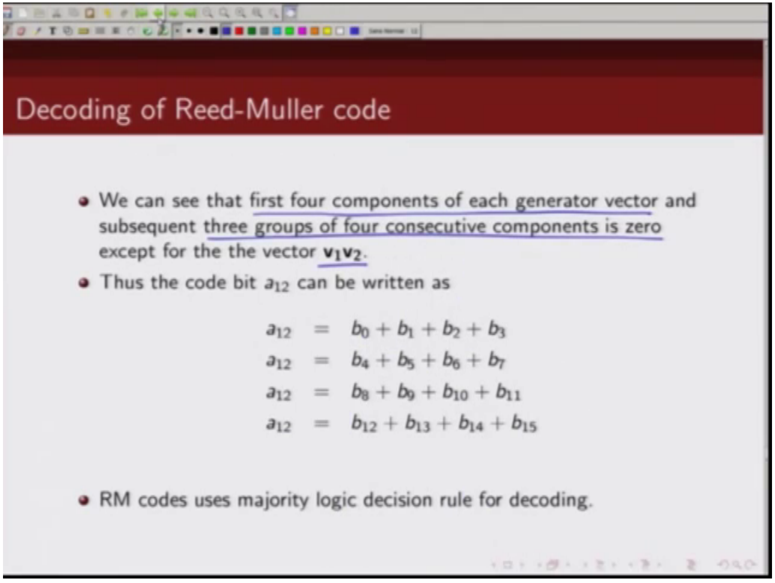
Decoding of Reed-Muller code

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

$$
\begin{array}{ll}
\mathbf{v}_0 & 1\,1\,1\,1\;1\,1\,1\,1\;1\,1\,1\,1\;1\,1\,1\,1 \\
\mathbf{v}_1 & 0\,1\,0\,1\;0\,1\,0\,1\;0\,1\,0\,1\;0\,1\,0\,1 \\
\mathbf{v}_2 & 0\,0\,1\,1\;0\,0\,1\,1\;0\,0\,1\,1\;0\,0\,1\,1 \\
\mathbf{v}_3 & 0\,0\,0\,0\;1\,1\,1\,1\;0\,0\,0\,0\;1\,1\,1\,1 \quad = 0 \\
\mathbf{v}_4 & 0\,0\,0\,0\;0\,0\,0\,0\;1\,1\,1\,1\;1\,1\,1\,1 \\
\rightarrow \mathbf{v}_1\mathbf{v}_2 & 0\,0\,0\,1\;0\,0\,0\,1\;0\,0\,0\,1\;0\,0\,0\,1 \\
\mathbf{v}_1\mathbf{v}_3 & 0\,0\,0\,0\;0\,1\,0\,1\;0\,0\,0\,0\;0\,1\,0\,1 \quad = 0 \\
\mathbf{v}_1\mathbf{v}_4 & 0\,0\,0\,0\;0\,0\,0\,0\;0\,1\,0\,1\;0\,1\,0\,1 \quad = 0 \\
\mathbf{v}_2\mathbf{v}_3 & 0\,0\,0\,0\;0\,0\,1\,1\;0\,0\,0\,0\;0\,0\,1\,1 \\
\mathbf{v}_2\mathbf{v}_4 & 0\,0\,0\,0\;0\,0\,0\,0\;0\,0\,1\,1\;0\,0\,1\,1 \\
\mathbf{v}_3\mathbf{v}_4 & 0\,0\,0\,0\;0\,0\,0\,0\;0\,0\,0\,0\;1\,1\,1\,1
\end{array}
$$

these bits, four bits at a time, you will notice except for this one, v 1 v 2, all others are zero. Now how can we make use of this fact?

(Refer Slide Time 39:49)



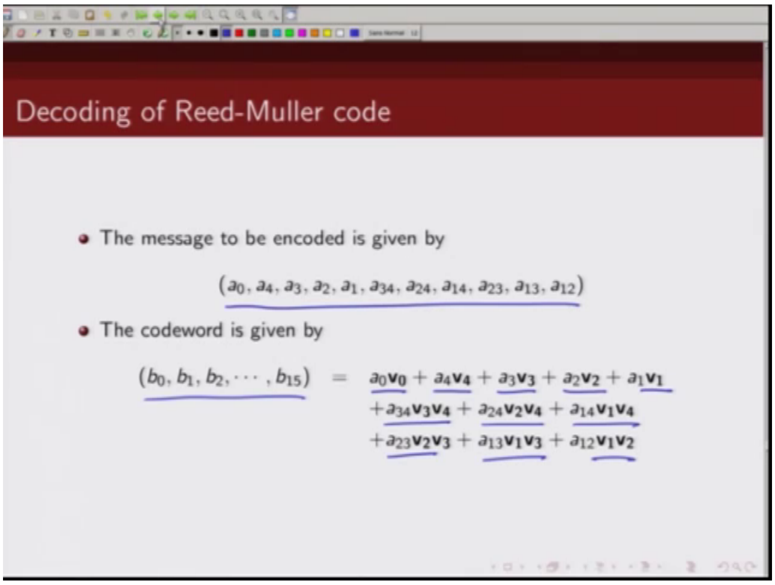
Decoding of Reed-Muller code

- 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.

(Refer Slide Time 39:50)



Decoding of Reed-Muller code

- The message to be encoded is given by

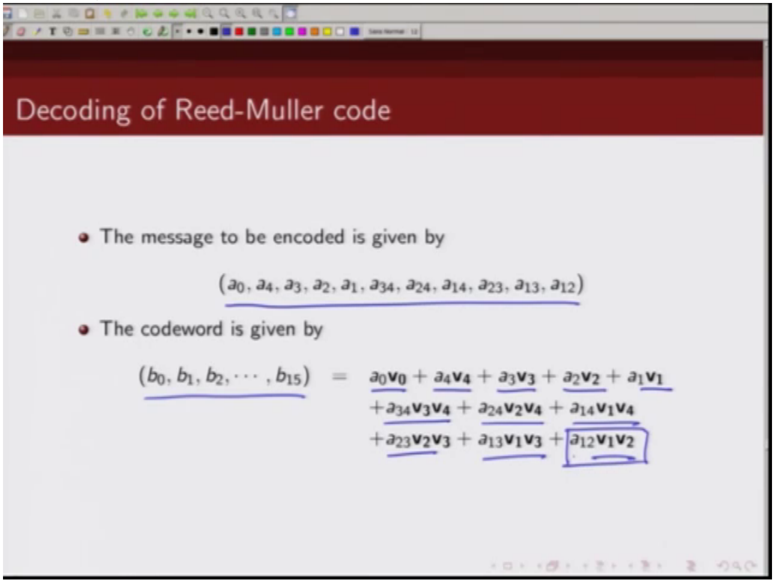
$$(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})$$

- The codeword is given by

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

v 1 v 2, so, so what we will do, if we add up

(Refer Slide Time 39:58)



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

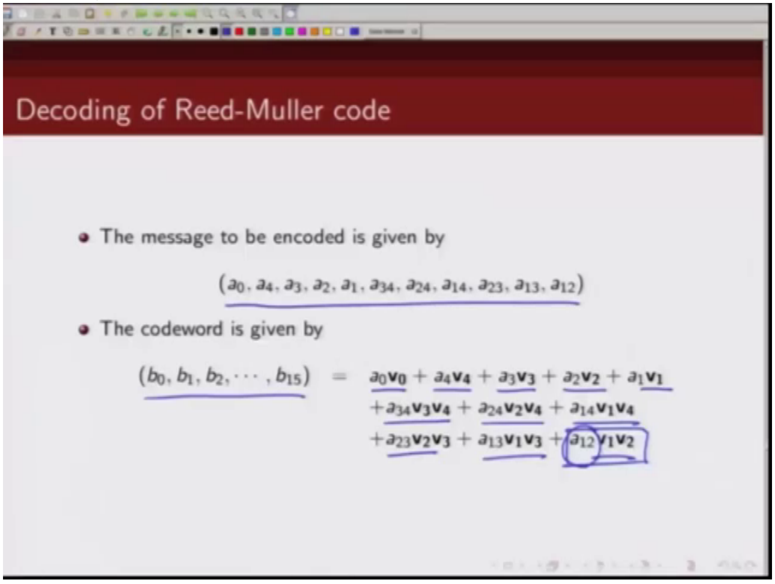
$$\begin{aligned}(b_0, b_1, b_2, \cdots, b_{15}) = \; & a_0\mathbf{v_0} + a_4\mathbf{v_4} + a_3\mathbf{v_3} + a_2\mathbf{v_2} + a_1\mathbf{v_1} \\ & + a_{34}\mathbf{v_3}\mathbf{v_4} + a_{24}\mathbf{v_2}\mathbf{v_4} + a_{14}\mathbf{v_1}\mathbf{v_4} \\ & + a_{23}\mathbf{v_2}\mathbf{v_3} + a_{13}\mathbf{v_1}\mathbf{v_3}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

those first four elements, the contribution from all others will be zero except, because v 1 v 2 is non-zero so we will get contribution

(Refer Slide Time 40:12)

from what a 1 2 is. So in other words,

(Refer Slide Time 40:18)



Decoding of Reed-Muller code

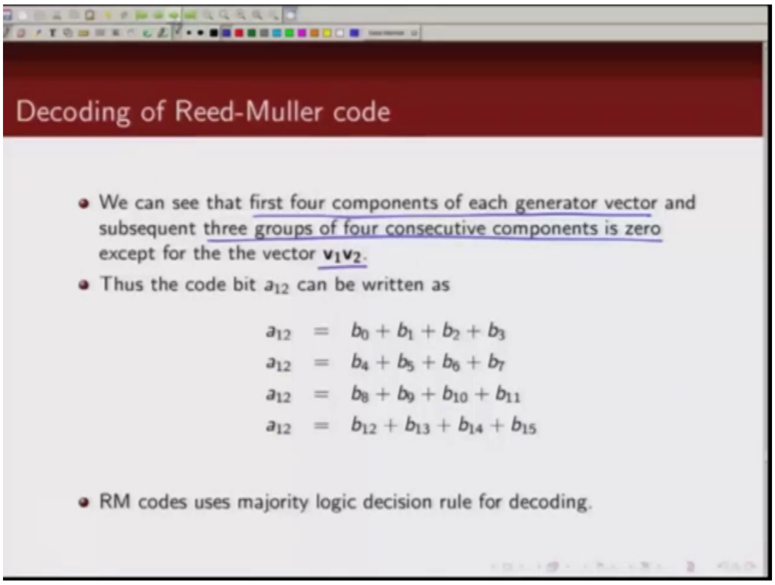
- The message to be encoded is given by

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

- The codeword is given by

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

(Refer Slide Time 40:19)



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 $\mathbf{v_1 v_2}$.
- Thus the code bit a_{12} can be written as

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

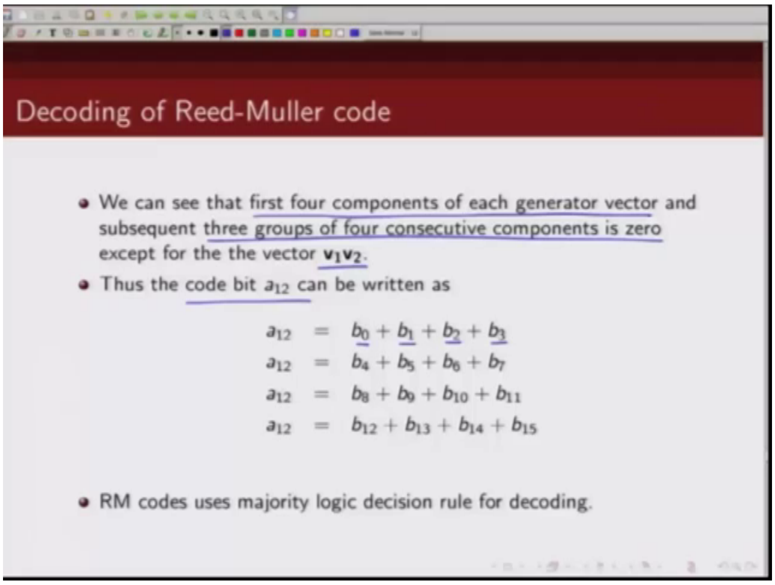
- RM codes uses majority logic decision rule for decoding.

these codeword bit then can be written as, so if I am calling this bit at 0th location as zero, bit at first location as b 1, second location b 2 and b 3 then by adding the first 4 bits I can get information about

(Refer Slide Time 40:39)

what a 1 2 was. And this can continue

(Refer Slide Time 40:46)



Decoding of Reed-Muller code

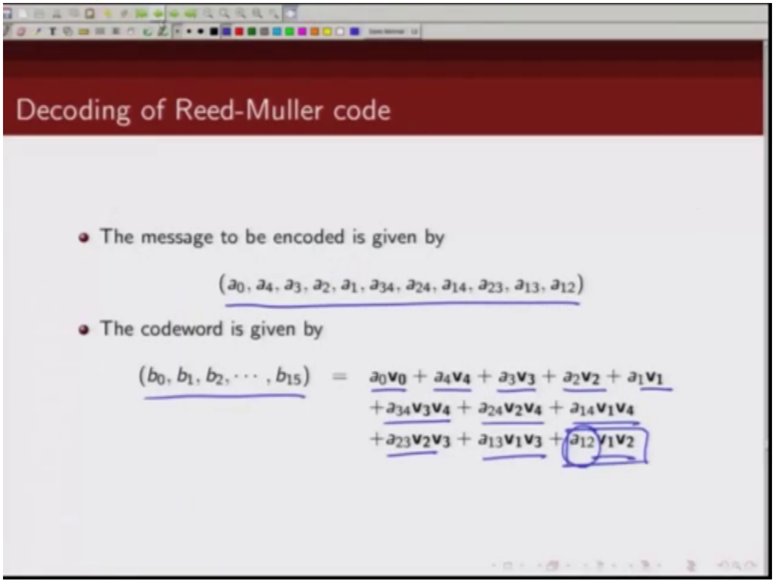
- We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the the vector $\mathbf{v_1v_2}$.
- Thus the code bit a_{12} can be written as

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

- RM codes uses majority logic decision rule for decoding.

for next set of

(Refer Slide Time 40:48)



Decoding of Reed-Muller code

- The message to be encoded is given by

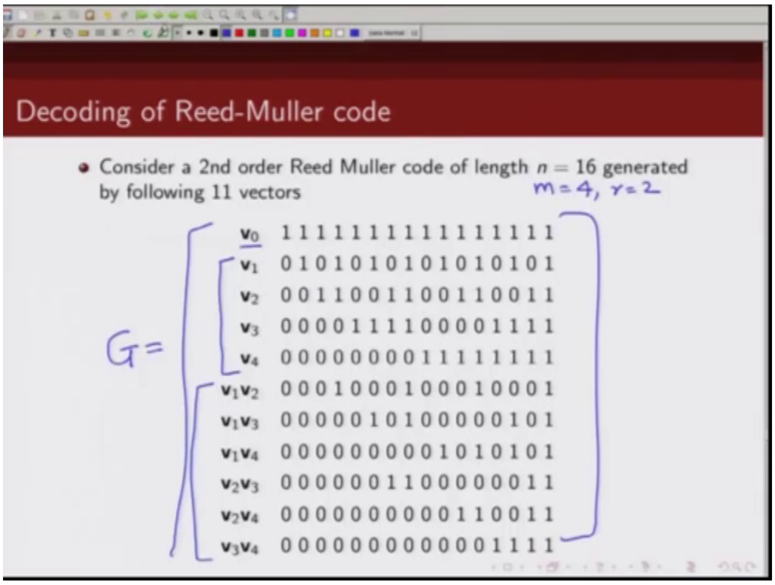
$$\underline{(a_0, a_4, a_3, a_2, a_1, a_{34}, a_{24}, a_{14}, a_{23}, a_{13}, a_{12})}$$

- The codeword is given by

$$\begin{aligned} \underline{(b_0, b_1, b_2, \cdots, b_{15})} &= a_0\mathbf{v_0} + a_4\mathbf{v_4} + a_3\mathbf{v_3} + a_2\mathbf{v_2} + a_1\mathbf{v_1} \\ &+ a_{34}\mathbf{v_3v_4} + a_{24}\mathbf{v_2v_4} + a_{14}\mathbf{v_1v_4} \\ &+ a_{23}\mathbf{v_2v_3} + a_{13}\mathbf{v_1v_3} + \boxed{a_{12}\mathbf{v_1v_2}} \end{aligned}$$

bits as well.

(Refer Slide Time 40:49)



Decoding of Reed-Muller code

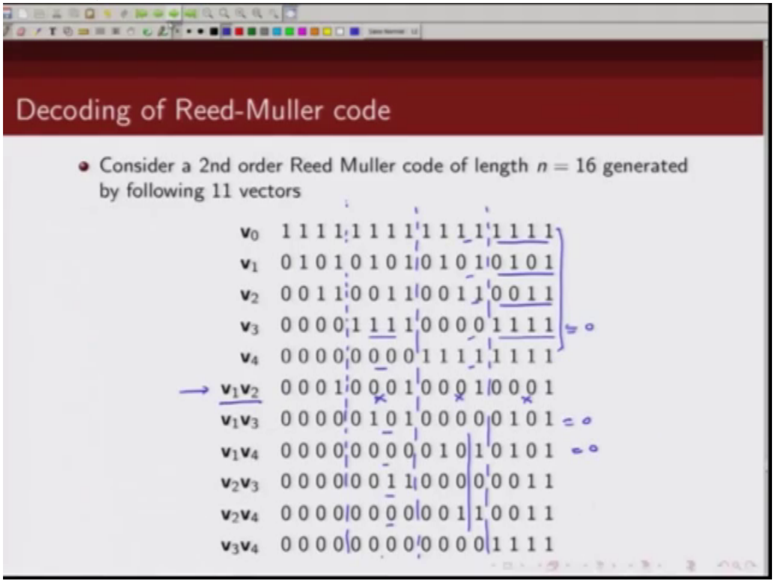
- Consider a 2nd order Reed Muller code of length $n = 16$ generated by following 11 vectors

$m=4,\ r=2$

$$G = \begin{array}{c|c}
\mathbf{v}_0 & 1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1\,1 \\
\mathbf{v}_1 & 0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1\,0\,1 \\
\mathbf{v}_2 & 0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1\,0\,0\,1\,1 \\
\mathbf{v}_3 & 0\,0\,0\,0\,1\,1\,1\,1\,0\,0\,0\,0\,1\,1\,1\,1 \\
\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,1\,1\,1 \\
\mathbf{v}_1\mathbf{v}_2 & 0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1\,0\,0\,0\,1 \\
\mathbf{v}_1\mathbf{v}_3 & 0\,0\,0\,0\,0\,1\,0\,1\,0\,0\,0\,0\,0\,1\,0\,1 \\
\mathbf{v}_1\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,0\,1\,0\,1\,0\,1 \\
\mathbf{v}_2\mathbf{v}_3 & 0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,0\,0\,0\,0\,1\,1 \\
\mathbf{v}_2\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,0\,0\,1\,1 \\
\mathbf{v}_3\mathbf{v}_4 & 0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,1\,1\,1\,1
\end{array}$$

So

(Refer Slide Time 40:52)



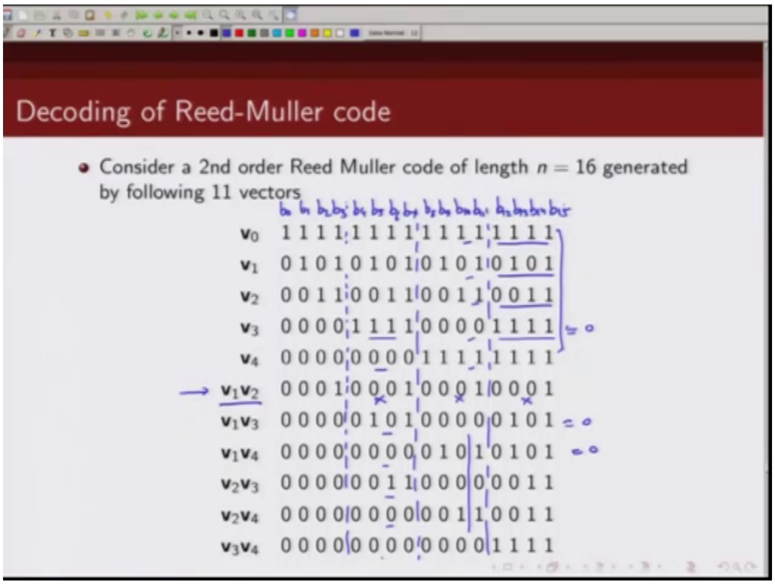
Decoding of Reed-Muller code

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

$$\begin{array}{c|cccc|c}
\mathbf{v}_0 & 1\,1\,1\,1 & 1\,1\,1\,1 & 1\,1\,1\,1 & 1\,1\,1\,1 & \\
\mathbf{v}_1 & 0\,1\,0\,1 & 0\,1\,0\,1 & 0\,1\,0\,1 & 0\,1\,0\,1 & \\
\mathbf{v}_2 & 0\,0\,1\,1 & 0\,0\,1\,1 & 0\,0\,1\,1 & 0\,0\,1\,1 & \\
\mathbf{v}_3 & 0\,0\,0\,0 & 1\,1\,1\,1 & 0\,0\,0\,0 & 1\,1\,1\,1 & = 0 \\
\mathbf{v}_4 & 0\,0\,0\,0 & 0\,0\,0\,0 & 1\,1\,1\,1 & 1\,1\,1\,1 & \\
\rightarrow \mathbf{v}_1\mathbf{v}_2 & 0\,0\,0\,1 & 0\,0\,0\,1 & 0\,0\,0\,1 & 0\,0\,0\,1 & \\
\mathbf{v}_1\mathbf{v}_3 & 0\,0\,0\,0 & 0\,1\,0\,1 & 0\,0\,0\,0 & 0\,1\,0\,1 & = 0 \\
\mathbf{v}_1\mathbf{v}_4 & 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,1\,0\,1 & 0\,1\,0\,1 & = 0 \\
\mathbf{v}_2\mathbf{v}_3 & 0\,0\,0\,0 & 0\,0\,1\,1 & 0\,0\,0\,0 & 0\,0\,1\,1 & \\
\mathbf{v}_2\mathbf{v}_4 & 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,1\,1 & 0\,0\,1\,1 & \\
\mathbf{v}_3\mathbf{v}_4 & 0\,0\,0\,0 & 0\,0\,0\,0 & 0\,0\,0\,0 & 1\,1\,1\,1 &
\end{array}$$

this is let's say b 0, b 1, b 2, b 3, this is b 4, b 5, b 6, b 7. This is b 8, b 9, b 10, b 11; this is b 12, b 13, b 14, b 15. So if I add these b 0,

(Refer Slide Time 41:16)



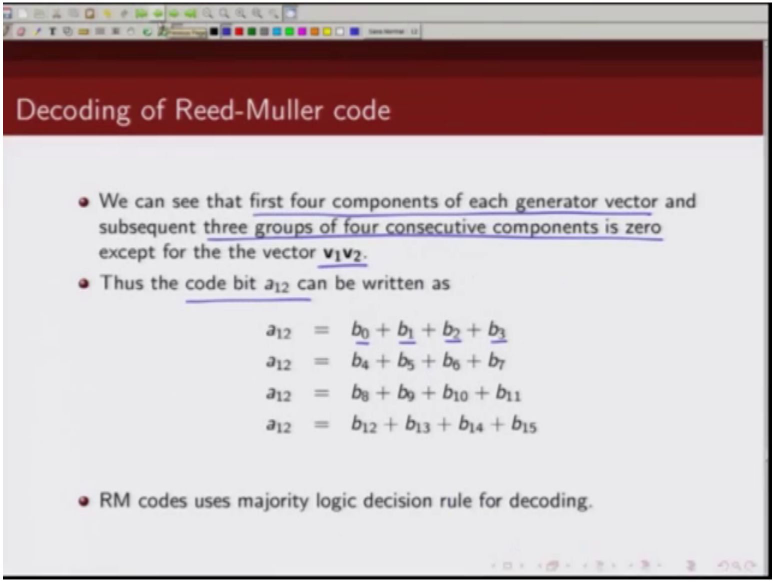
Decoding of Reed-Muller code

- 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}	
$\mathbf{v}_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
$\mathbf{v}_1$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
$\mathbf{v}_2$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
$\mathbf{v}_3$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	= 0
$\mathbf{v}_4$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
→ $\mathbf{v}_1\mathbf{v}_2$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	
$\mathbf{v}_1\mathbf{v}_3$	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	= 0
$\mathbf{v}_1\mathbf{v}_4$	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	= 0
$\mathbf{v}_2\mathbf{v}_3$	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	
$\mathbf{v}_2\mathbf{v}_4$	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	
$\mathbf{v}_3\mathbf{v}_4$	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	

b 1, b 2, b 3, or b 4, b 5, b 6, b 7, b 8, b 9, b 10, b 11, b 12, b 13, b 14, b 15, what I am getting is contributions from all other rows are nullified. Only I will see the contribution, effect of this v 1 v 2 and the bit

(Refer Slide Time 41:39)



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

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

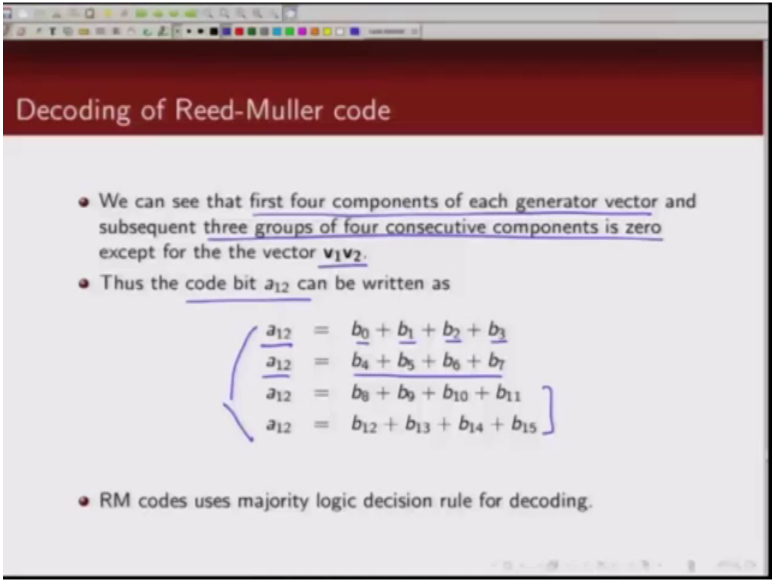
- RM codes uses majority logic decision rule for decoding.

a 1 2 can then be found by adding these four columns together. So I can get information about a 1 2 by looking at these first 4 columns or first 4 bits of these codeword, similarly next four bits of the codeword, add them up, I can get another independent information about a 1 2 and same thing I can get from the next set of 4 coded bits. So what you can see is I am getting 4 independent views about what a 1 2 is. Now the decoder can take the majority logic decoder if there is no error of course all of them will tell me about that a 1 2 is the same bit whether it is zero or 1.

(Refer Slide Time 42:32)

But if there is a single error, what you will notice is you know, in some of the bits, let us say there is an error in some bit location b 1 then

(Refer Slide Time 42:41)



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 $\mathbf{v_1 v_2}$.
- Thus the code bit a_{12} can be written as

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

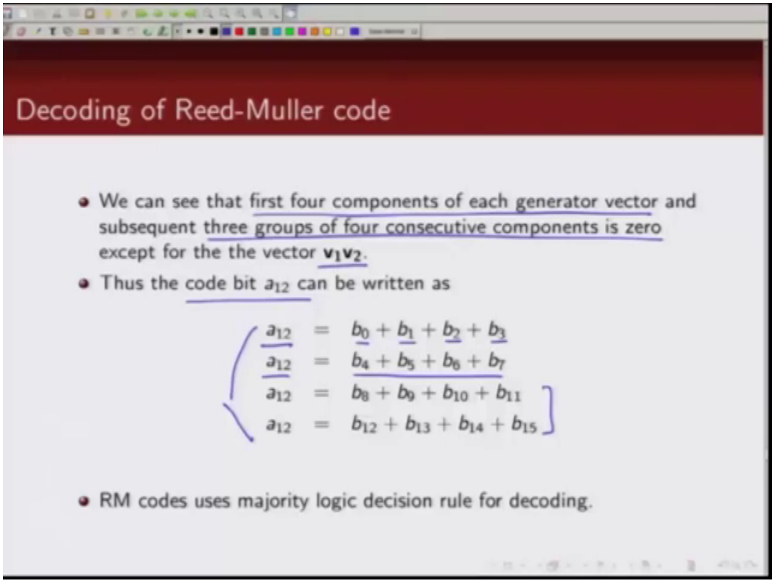
- RM codes uses majority logic decision rule for decoding.

a 1 2 here would be different from what a 1 2 I am getting from other 3 equations. And then I will use majority logic

(Refer Slide Time 42:49)

decoding. What is majority logic decoding? So I will take majority decision, if three of them are saying a 1 2 is zero, then I will go for zero, otherwise I will go for 1, Ok. So this is how I can decode

(Refer Slide Time 43:04)



Decoding of Reed-Muller code

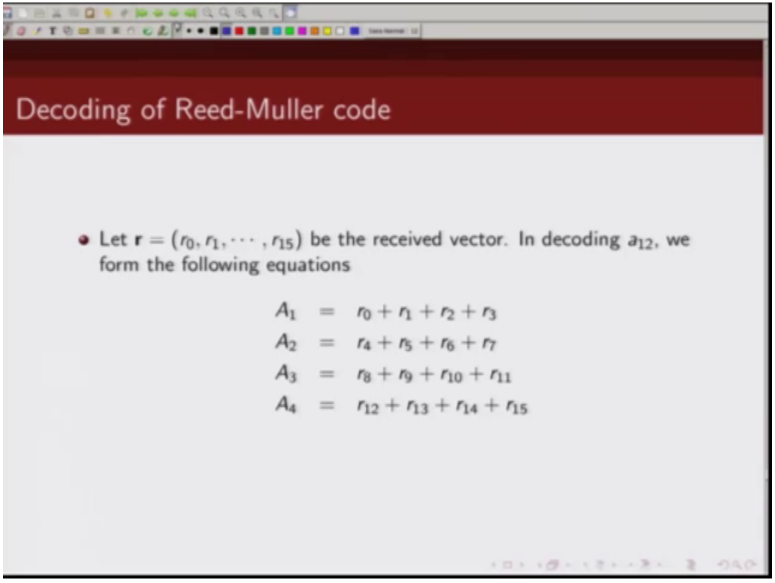
- We can see that first four components of each generator vector and subsequent three groups of four consecutive components is zero except for the the vector $\mathbf{v_1 v_2}$.
- Thus the code bit a_{12} can be written as

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

- RM codes uses majority logic decision rule for decoding.

bit a 1 2. So and this will be repeated for

(Refer Slide Time 43:10)



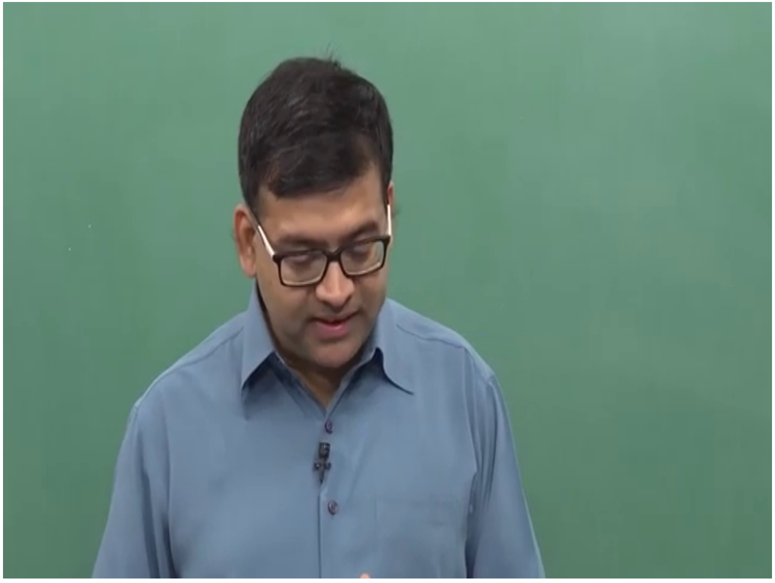
Decoding of Reed-Muller code

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

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

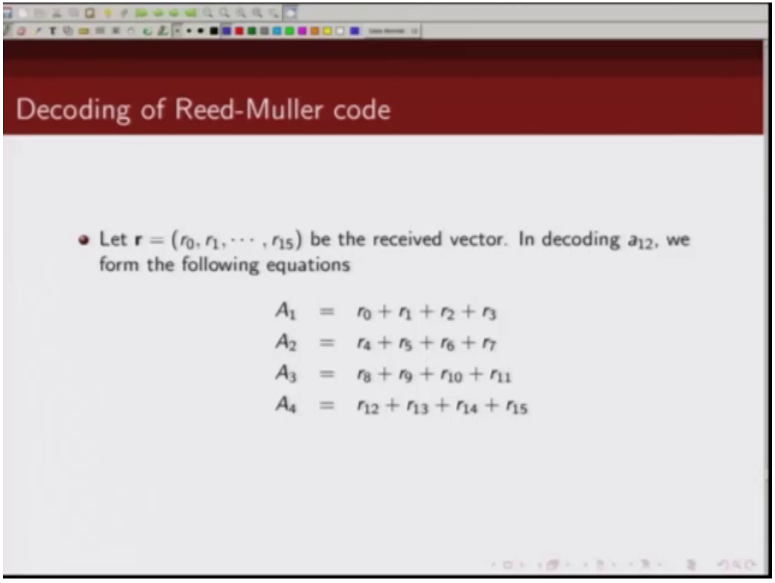
decoding other bits as well.

(Refer Slide Time 43:12)



So let's say my received bit is

(Refer Slide Time 43:15)



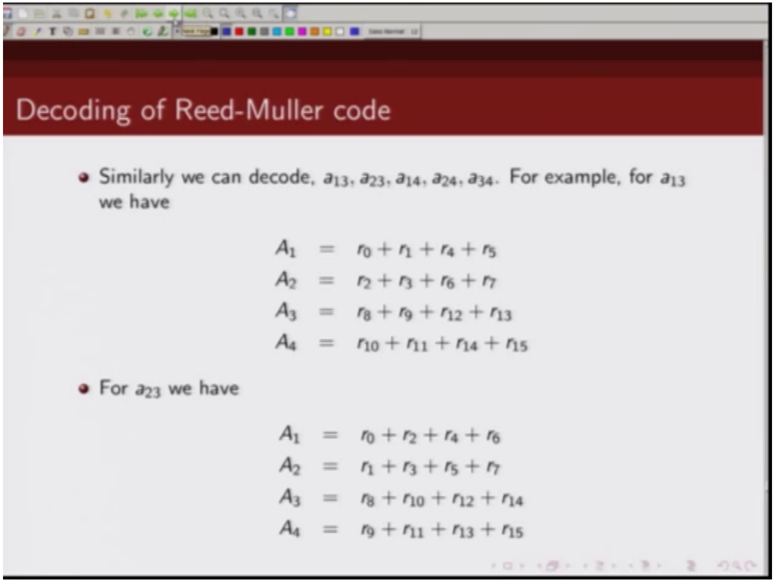
Decoding of Reed-Muller code

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

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

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 1 2. How? I will just add these first 4 bits then I will add the next 4 bits next 4 bits next 4 bits so I am getting 4 independent views about what a 1 2 is and then I will take a majority decision, majority of them are saying zero, I will go for zero. Otherwise I will go for

(Refer Slide Time 43:46)



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

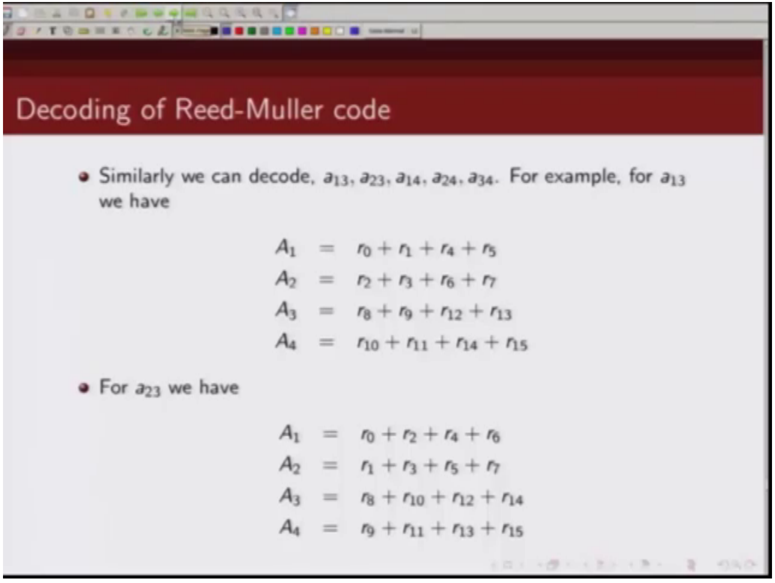
$$\begin{aligned} A_1 &= r_0 + r_2 + r_4 + r_6 \\ A_2 &= r_1 + r_3 + r_5 + r_7 \\ A_3 &= r_8 + r_{10} + r_{12} + r_{14} \\ A_4 &= r_9 + r_{11} + r_{13} + r_{15} \end{aligned}$$

1. Now the same thing, exactly

(Refer Slide Time 43:49)

the same way I can decode other bits. So let's look

(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

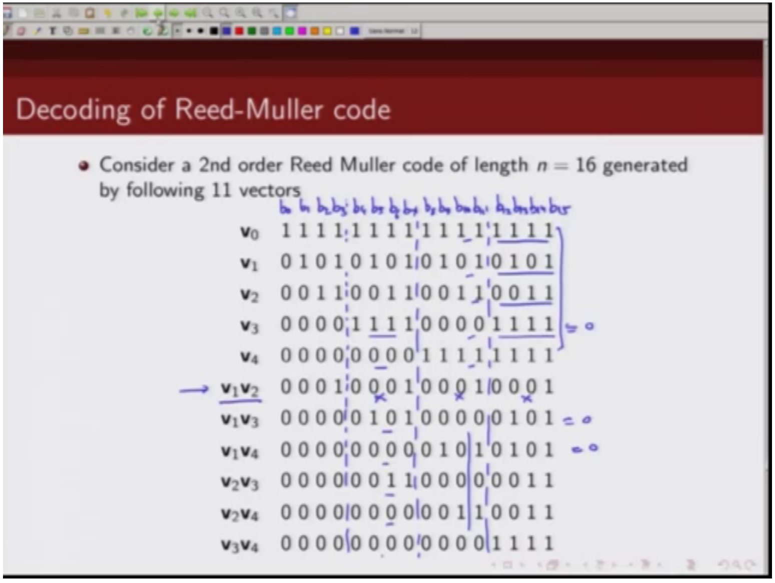
$$\begin{aligned} A_1 &= r_0 + r_1 + r_4 + r_5 \\ A_2 &= r_2 + r_3 + r_6 + r_7 \\ A_3 &= r_8 + r_9 + r_{12} + r_{13} \\ A_4 &= r_{10} + r_{11} + r_{14} + r_{15} \end{aligned}$$

- For a_{23} we have

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

at a 2 3. If we look at a 2 3,

(Refer Slide Time 43:58)



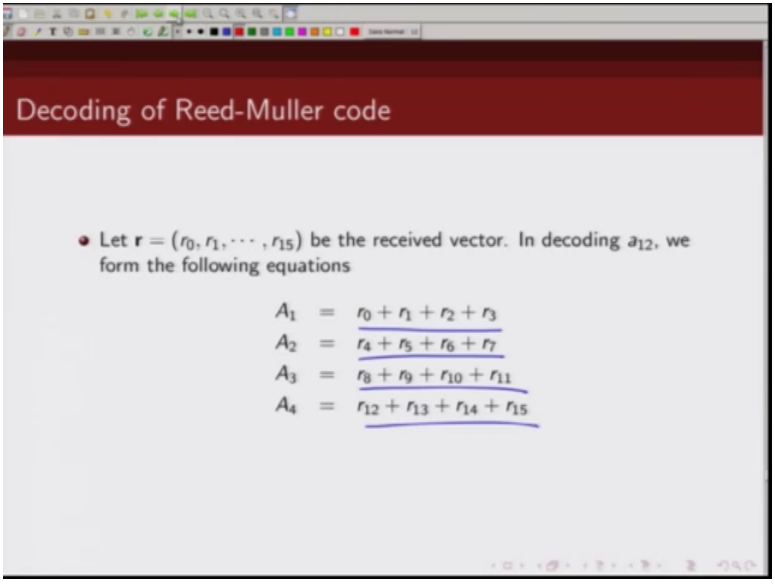
Decoding of Reed-Muller code

- 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}	
$\mathbf{v}_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
$\mathbf{v}_1$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	
$\mathbf{v}_2$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	
$\mathbf{v}_3$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1	= 0
$\mathbf{v}_4$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	
→ $\mathbf{v}_1\mathbf{v}_2$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	
$\mathbf{v}_1\mathbf{v}_3$	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1	= 0
$\mathbf{v}_1\mathbf{v}_4$	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1	= 0
$\mathbf{v}_2\mathbf{v}_3$	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	
$\mathbf{v}_2\mathbf{v}_4$	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1	
$\mathbf{v}_3\mathbf{v}_4$	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	

let's look at this row, I will use a different pen. Let's look at this row, this row, this row, and this row. So if I add bits in this row, this will be zero, this will give me zero, this will give me a 1, this will give me zero, this will give me zero, this will give me zero, so you can see all rows will give me zero except this particular row and same thing I can repeat

(Refer Slide Time 44:42)



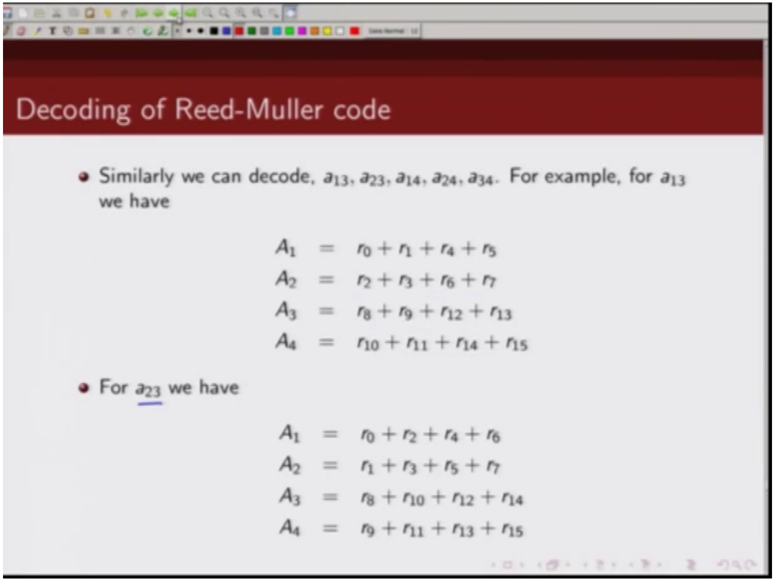
Decoding of Reed-Muller code

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

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

for,

(Refer Slide Time 44:44)



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

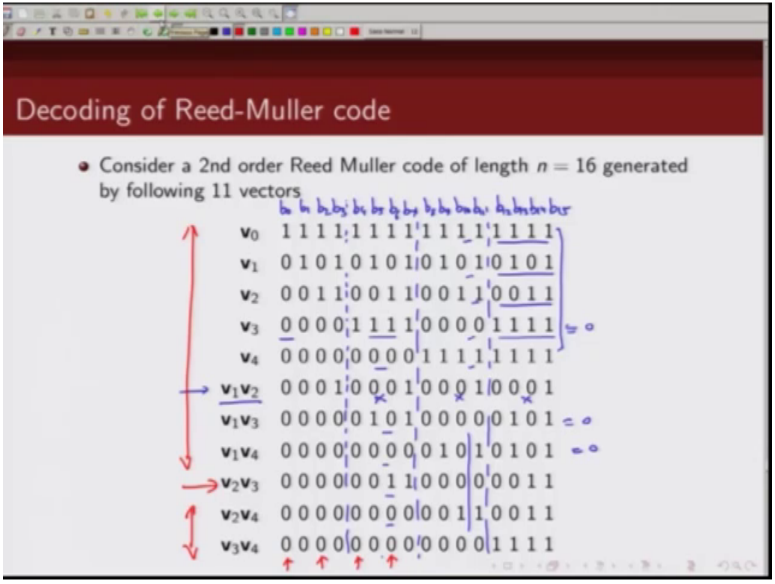
$$\begin{aligned} A_1 &= r_0 + r_1 + r_4 + r_5 \\ A_2 &= r_2 + r_3 + r_6 + r_7 \\ A_3 &= r_8 + r_9 + r_{12} + r_{13} \\ A_4 &= r_{10} + r_{11} + r_{14} + r_{15} \end{aligned}$$

- For $\underline{a_{23}}$ 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 second row, fourth row, sixth row and eighth row I will get the same information. So if I look at, now let's say I

(Refer Slide Time 44:55)



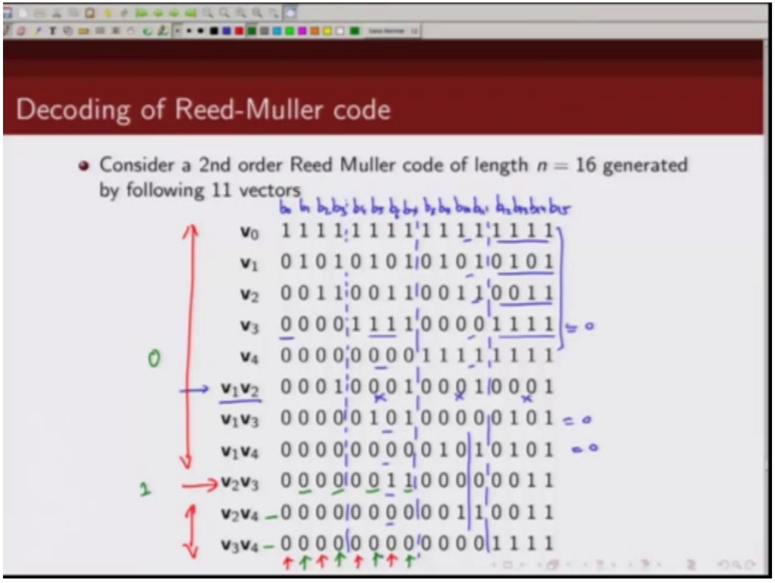
Decoding of Reed-Muller code

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

$$\begin{array}{c|cccccccccccccccc|c}
 & b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & b_{10} & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & \\
\mathbf{v}_0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\
\mathbf{v}_1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & \\
\mathbf{v}_2 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & \\
\mathbf{v}_3 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & = 0 \\
\mathbf{v}_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\
\mathbf{v}_1\mathbf{v}_2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \\
\mathbf{v}_1\mathbf{v}_3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & = 0 \\
\mathbf{v}_1\mathbf{v}_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & = 0 \\
\mathbf{v}_2\mathbf{v}_3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & \\
\mathbf{v}_2\mathbf{v}_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & \\
\mathbf{v}_3\mathbf{v}_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 &
\end{array}$$

look at this row, I look at this row, this row, this row and this row. So this will give me zero, this will give me zero, this will give me zero. Now here this is a 1, this is a zero, this is a zero and this is zero. So this will give me 1. And all other rows will give me zero.

(Refer Slide Time 45:21)



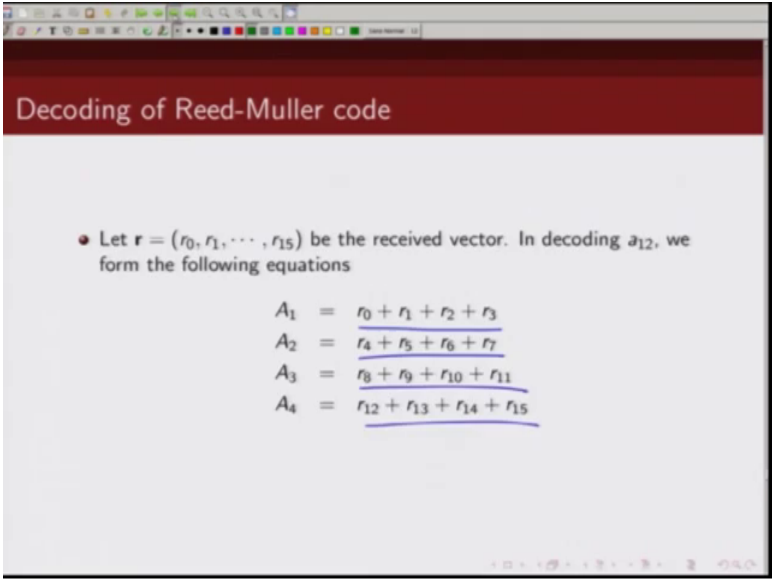
Decoding of Reed-Muller code

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

$$
\begin{array}{ll|cccccccccccccccc|l}
 & & b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 & b_9 & b_{10} & b_{11} & b_{12} & b_{13} & b_{14} & b_{15} & \\
 & \mathbf{v}_0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\
 & \mathbf{v}_1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & \\
 & \mathbf{v}_2 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & \\
 & \mathbf{v}_3 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & = 0 \\
0 & \mathbf{v}_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \\
 & \mathbf{v}_1\mathbf{v}_2 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & \\
 & \mathbf{v}_1\mathbf{v}_3 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & = 0 \\
 & \mathbf{v}_1\mathbf{v}_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & = 0 \\
1 & \mathbf{v}_2\mathbf{v}_3 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & \\
 & \mathbf{v}_2\mathbf{v}_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & \\
 & \mathbf{v}_3\mathbf{v}_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 &
\end{array}
$$

So if I add up

(Refer Slide Time 45:26)

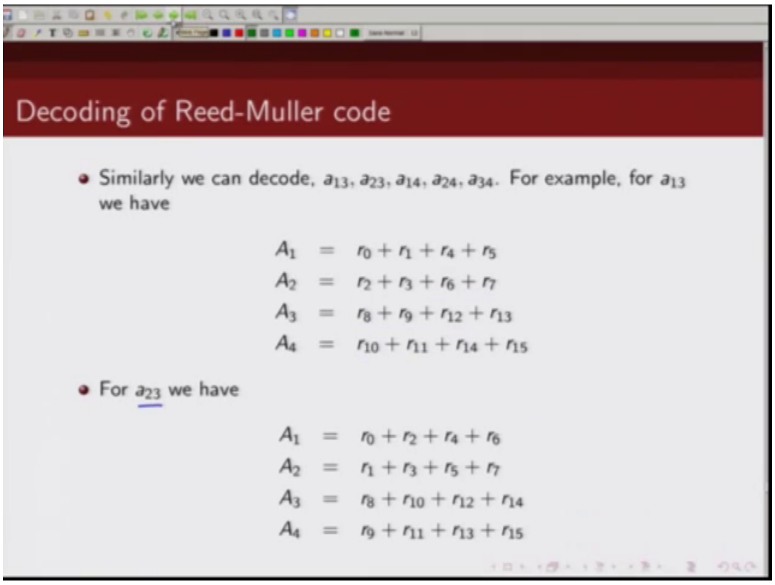


Decoding of Reed-Muller code

- Let $\mathbf{r} = (r_0, r_1, \cdots, r_{15})$ 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 45:27)



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 $\underline{a_{23}}$ we have

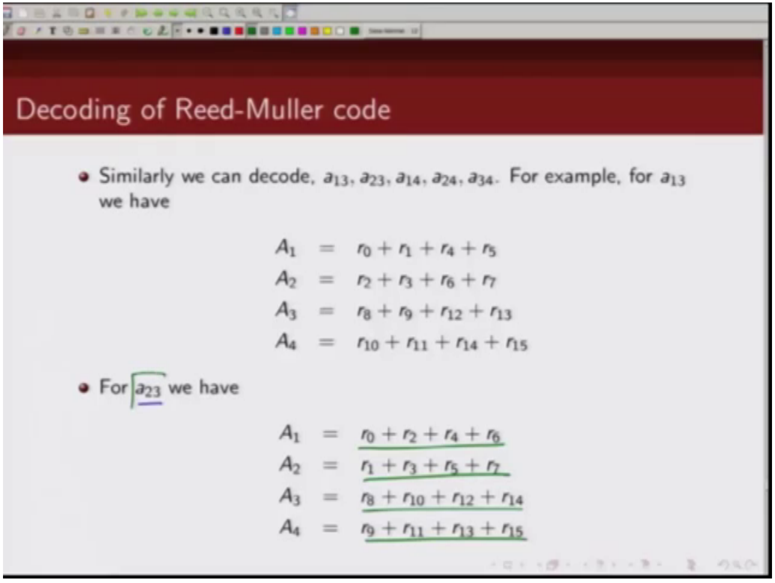
$$\begin{aligned} A_1 &= r_0 + r_2 + r_4 + r_6 \\ A_2 &= r_1 + r_3 + r_5 + r_7 \\ A_3 &= r_8 + r_{10} + r_{12} + r_{14} \\ A_4 &= r_9 + r_{11} + r_{13} + r_{15} \end{aligned}$$

these bits, 4 bits at a time, in similar fashion I can

(Refer Slide Time 45:34)

get independent information

(Refer Slide Time 45:36)



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

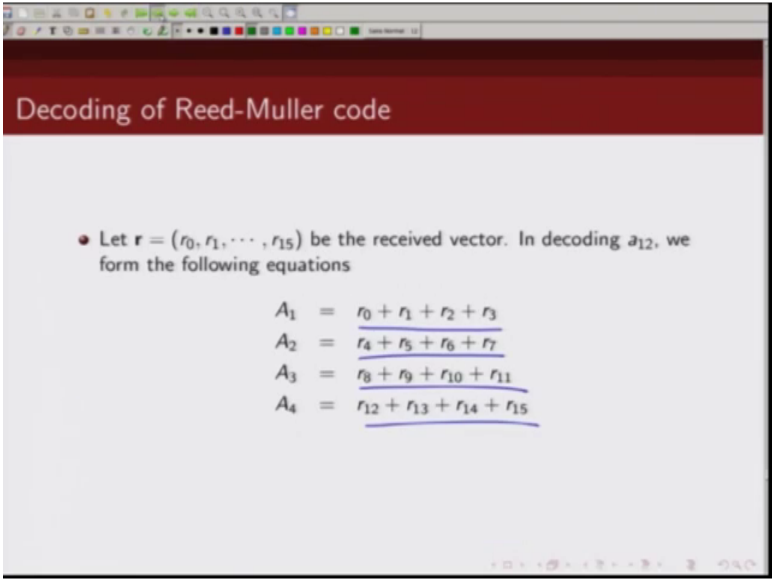
$$\begin{aligned} A_1 &= r_0 + r_1 + r_4 + r_5 \\ A_2 &= r_2 + r_3 + r_6 + r_7 \\ A_3 &= r_8 + r_9 + r_{12} + r_{13} \\ A_4 &= r_{10} + r_{11} + r_{14} + r_{15} \end{aligned}$$

- For $\underline{a_{23}}$ we have

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

about a 2 3. So again the point to be noted here is

(Refer Slide Time 45:44)



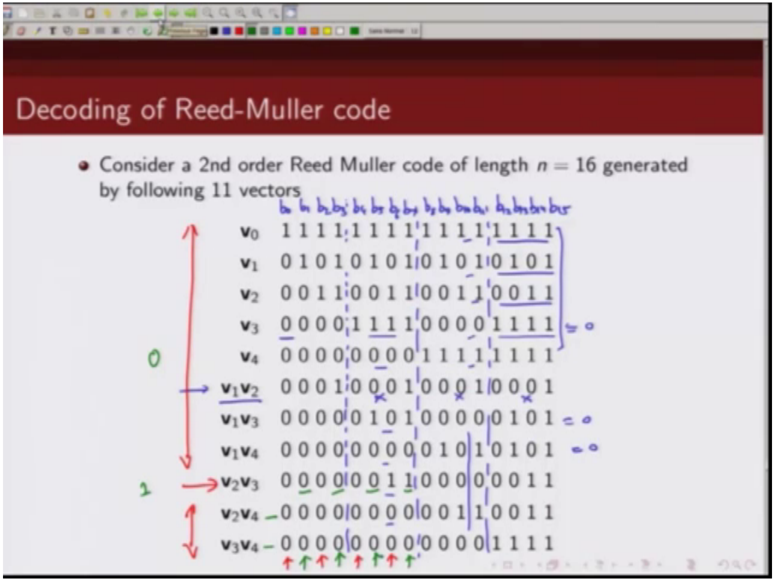
Decoding of Reed-Muller code

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

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

what you need to do

(Refer Slide Time 45:45)



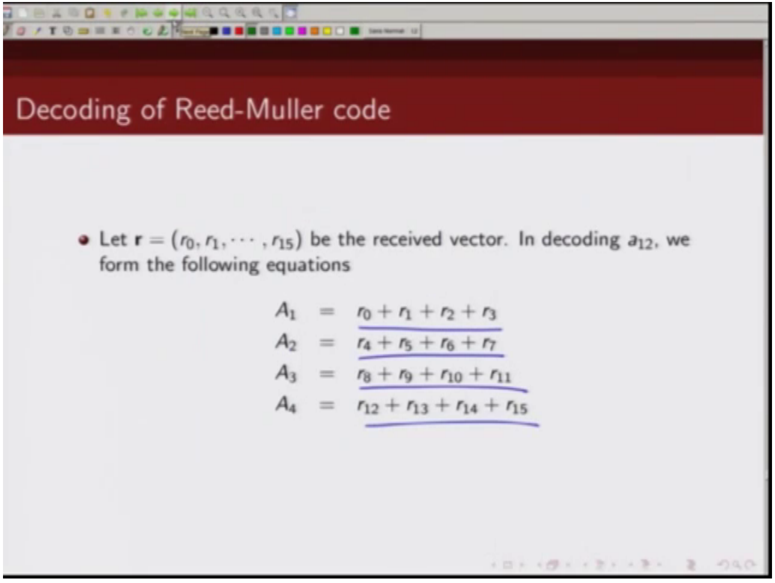
Decoding of Reed-Muller code

- 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}
$\mathbf{v}_0$	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
$\mathbf{v}_1$	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
$\mathbf{v}_2$	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
$\mathbf{v}_3$	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
$\mathbf{v}_4$	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
$\mathbf{v}_1\mathbf{v}_2$	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
$\mathbf{v}_1\mathbf{v}_3$	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1
$\mathbf{v}_1\mathbf{v}_4$	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1
$\mathbf{v}_2\mathbf{v}_3$	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1
$\mathbf{v}_2\mathbf{v}_4$	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1
$\mathbf{v}_3\mathbf{v}_4$	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

if you would look at this and find out this, basically like,

(Refer Slide Time 45:51)



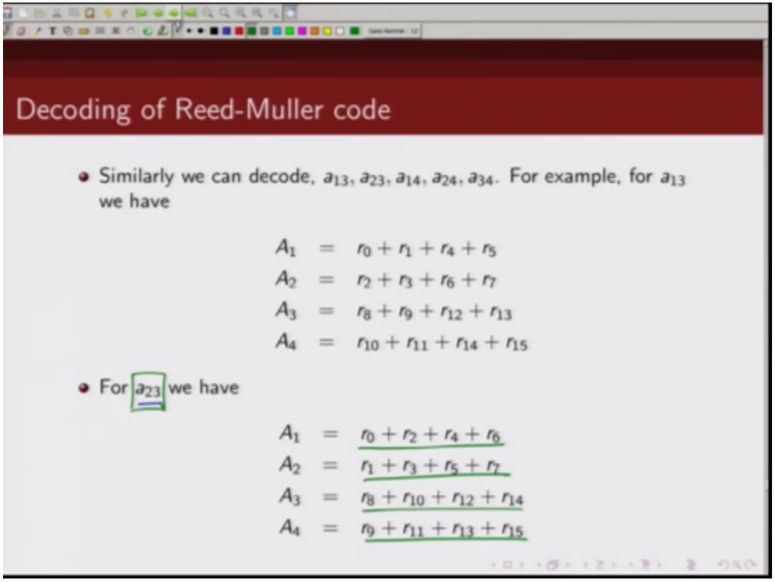
Decoding of Reed-Muller code

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

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

combinations of these received bits which will give information about one particular transmitted bit and not others and once you do that

(Refer Slide Time 46:04)



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

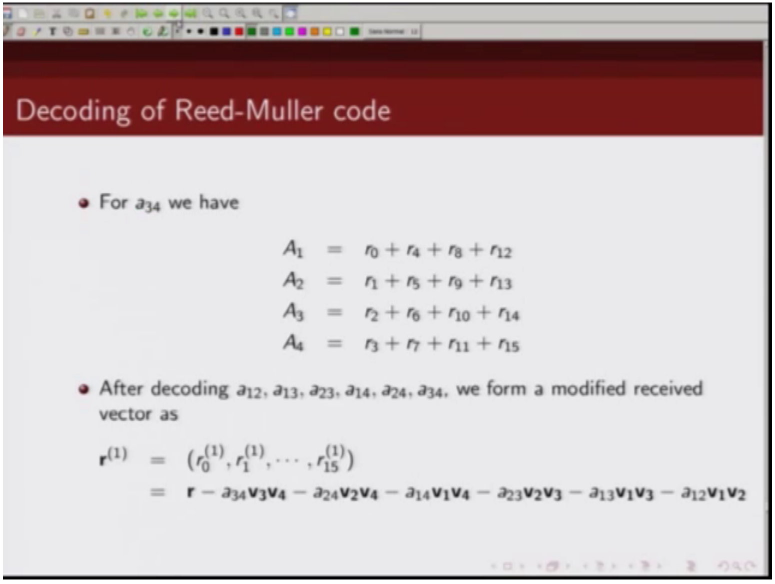
$$\begin{aligned} A_1 &= r_0 + r_1 + r_4 + r_5 \\ A_2 &= r_2 + r_3 + r_6 + r_7 \\ A_3 &= r_8 + r_9 + r_{12} + r_{13} \\ A_4 &= r_{10} + r_{11} + r_{14} + r_{15} \end{aligned}$$

- For $\boxed{\underline{a_{23}}}$ we have

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

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 1 4, similarly for a 2 4 and

(Refer Slide Time 46:22)



Decoding of Reed-Muller code

- For a_{34} we have

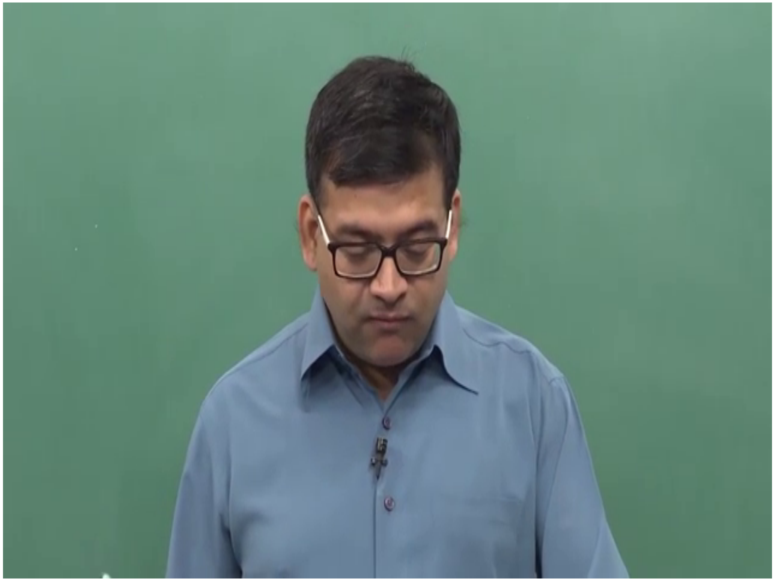
$$\begin{aligned} A_1 &= r_0 + r_4 + r_8 + r_{12} \\ A_2 &= r_1 + r_5 + r_9 + r_{13} \\ A_3 &= r_2 + r_6 + r_{10} + r_{14} \\ A_4 &= r_3 + r_7 + r_{11} + r_{15} \end{aligned}$$

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

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

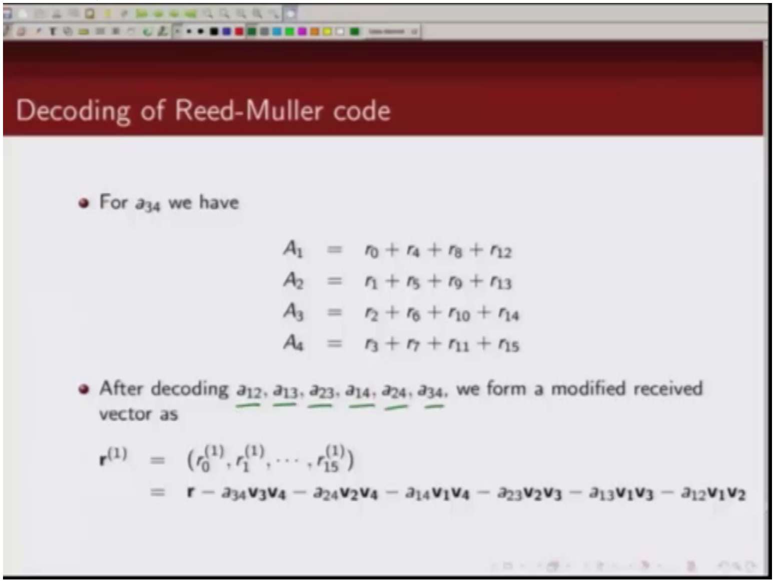
a 3 4. Now once you have decoded a 1 2 a 2 3 or once you have decoded all of these, again remember the way we are decoding is, so we are getting

(Refer Slide Time 46:34)



four independent views about the same bit. Majority of them are saying; it is zero we go for that. Or else if majority of them are saying they are 1, we go for that. So once we have decoded

(Refer Slide Time 46:47)



Decoding of Reed-Muller code

- For a_{34} we have

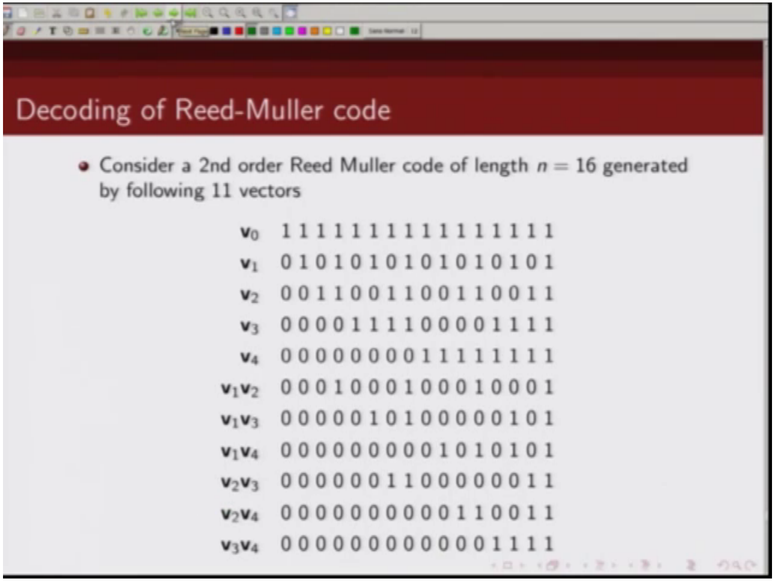
$$\begin{aligned} A_1 &= r_0 + r_4 + r_8 + r_{12} \\ A_2 &= r_1 + r_5 + r_9 + r_{13} \\ A_3 &= r_2 + r_6 + r_{10} + r_{14} \\ A_4 &= r_3 + r_7 + r_{11} + r_{15} \end{aligned}$$

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

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

this, these sequences let's just subtract the contribution of these bits from the received signal. So the new received sequence we are calling r 1 is the actual received sequence minus the contribution from these Boolean product terms subtracted. Now once we do this, then what we are left is

(Refer Slide Time 47:20)



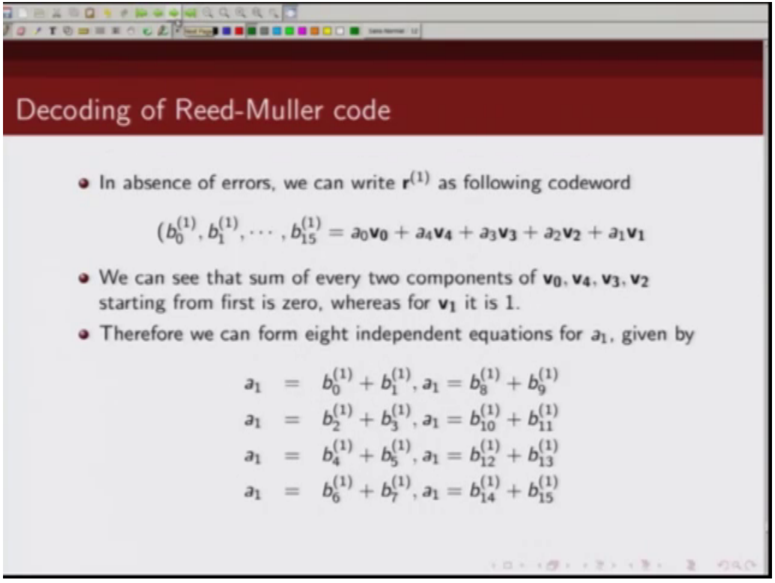
Decoding of Reed-Muller code

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

$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

essentially we are

(Refer Slide Time 47:21)



Decoding of Reed-Muller code

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

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

- 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

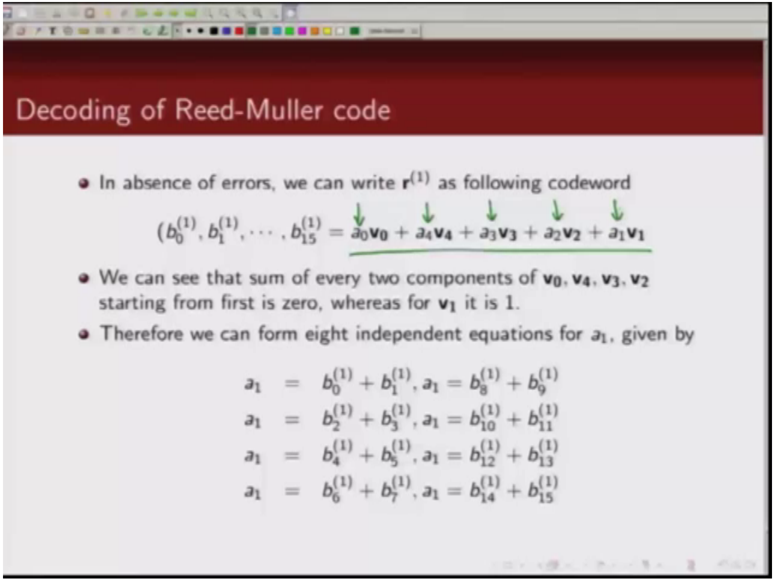
$$\begin{aligned}
a_1 &= b_0^{(1)} + b_1^{(1)}, a_1 = b_8^{(1)} + b_9^{(1)} \\
a_1 &= b_2^{(1)} + b_3^{(1)}, a_1 = b_{10}^{(1)} + b_{11}^{(1)} \\
a_1 &= b_4^{(1)} + b_5^{(1)}, a_1 = b_{12}^{(1)} + b_{13}^{(1)} \\
a_1 &= b_6^{(1)} + b_7^{(1)}, a_1 = b_{14}^{(1)} + b_{15}^{(1)}
\end{aligned}$$

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 rth order terms. Then we try to decode

(Refer Slide Time 47:38)

r minus 1th order term. And finally so here, we first decoded the terms

(Refer Slide Time 47:43)



Decoding of Reed-Muller code

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

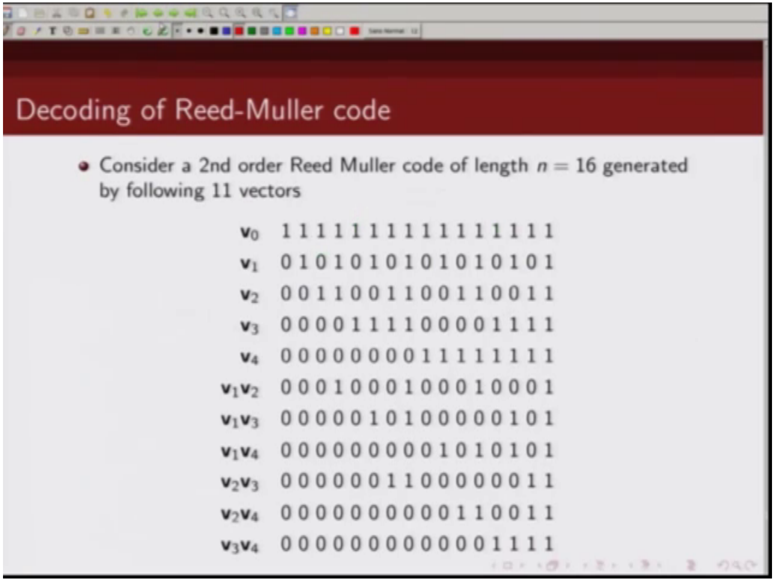
$$(b_0^{(1)}, b_1^{(1)}, \cdots, b_{15}^{(1)}) = \underline{\overset{\downarrow}{a_0}\mathbf{v_0} + \overset{\downarrow}{a_4}\mathbf{v_4} + \overset{\downarrow}{a_3}\mathbf{v_3} + \overset{\downarrow}{a_2}\mathbf{v_2} + \overset{\downarrow}{a_1}\mathbf{v_1}}$$

- 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

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

related to the 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 this G matrix

(Refer Slide Time 47:59)



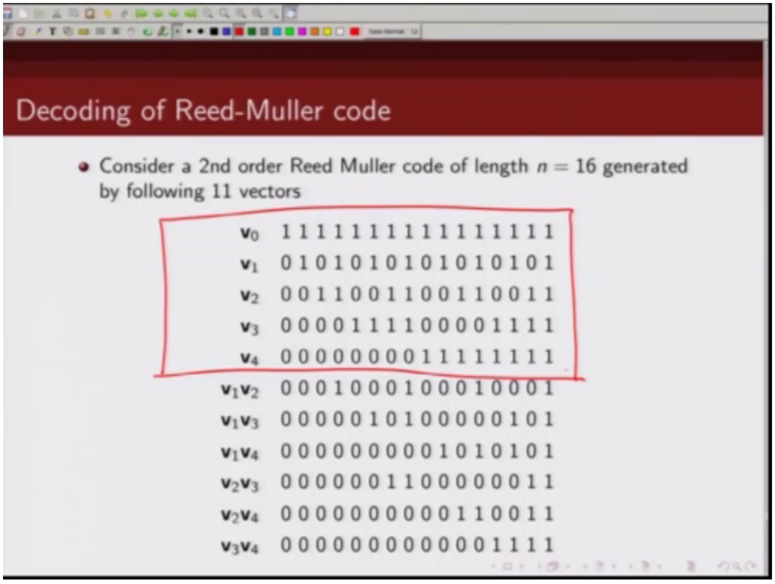
Decoding of Reed-Muller code

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

$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

and we are going to look at the bit. So we are now looking at, because the contribution of these have now been removed. So we are now looking at this G matrix. We are only looking at this.

(Refer Slide Time 48:13)



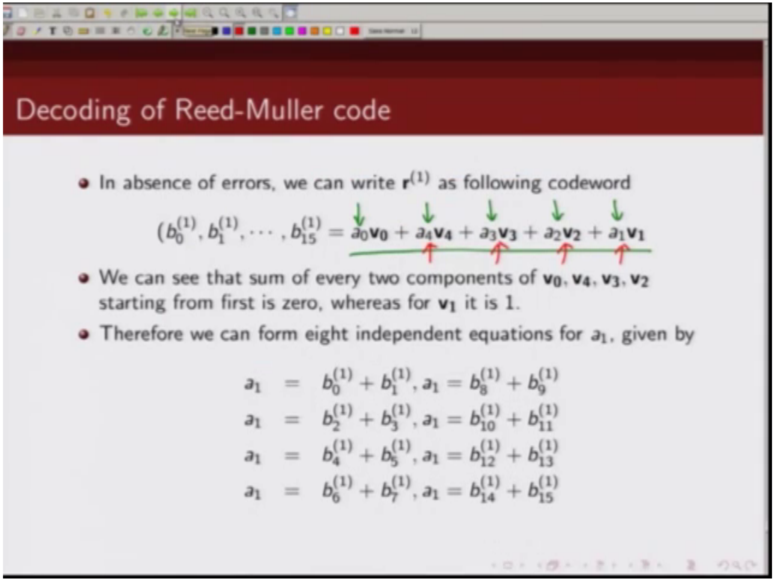
Decoding of Reed-Muller code

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

$\mathbf{v}_0$	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
$\mathbf{v}_1$	0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
$\mathbf{v}_2$	0 0 1 1 0 0 1 1 0 0 1 1 0 0 1 1
$\mathbf{v}_3$	0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 1
$\mathbf{v}_4$	0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1
$\mathbf{v}_1\mathbf{v}_2$	0 0 0 1 0 0 0 1 0 0 0 1 0 0 0 1
$\mathbf{v}_1\mathbf{v}_3$	0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 1
$\mathbf{v}_1\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 1 0 1 0 1 0 1
$\mathbf{v}_2\mathbf{v}_3$	0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1
$\mathbf{v}_2\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 1
$\mathbf{v}_3\mathbf{v}_4$	0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

Assuming we have correctly decoded a 1 2, a 1 3, a 1 4 contributions of this have been removed. So only we think we have been left is this. Now if you notice, if you add up 2 rows like this, consider these 2 rows, so what you would notice for all other except v 1, we will get zero. So in other words,

(Refer Slide Time 48:42)



Decoding of Reed-Muller code

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

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

- 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

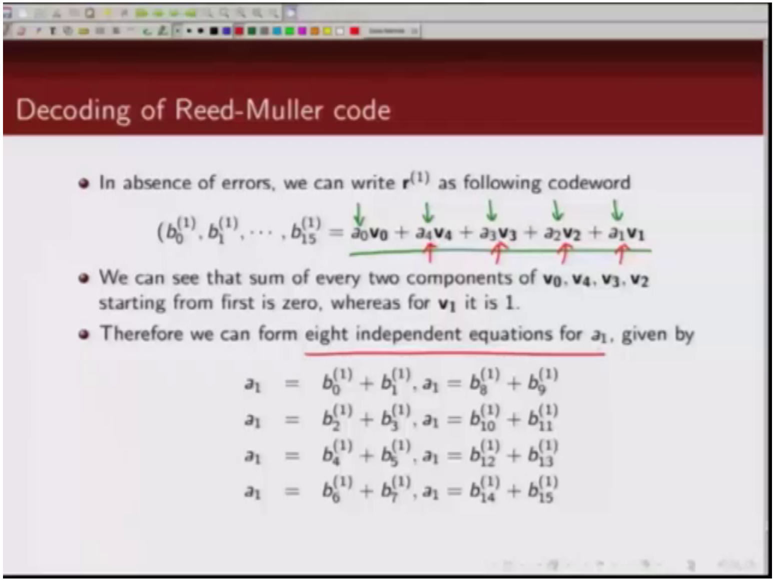
$$\begin{aligned} a_1 &= b_0^{(1)} + b_1^{(1)}, a_1 = b_8^{(1)} + b_9^{(1)} \\ a_1 &= b_2^{(1)} + b_3^{(1)}, a_1 = b_{10}^{(1)} + b_{11}^{(1)} \\ a_1 &= b_4^{(1)} + b_5^{(1)}, a_1 = b_{12}^{(1)} + b_{13}^{(1)} \\ a_1 &= b_6^{(1)} + b_7^{(1)}, a_1 = b_{14}^{(1)} + b_{15}^{(1)} \end{aligned}$$

I can get 8 independent views about what a 1 by just looking at looking 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

(Refer Slide Time 48:59)

whatever majority of them are saying I will, I will decide in favor of that. And the same procedure can be repeated to find out what

(Refer Slide Time 49:08)



Decoding of Reed-Muller code

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

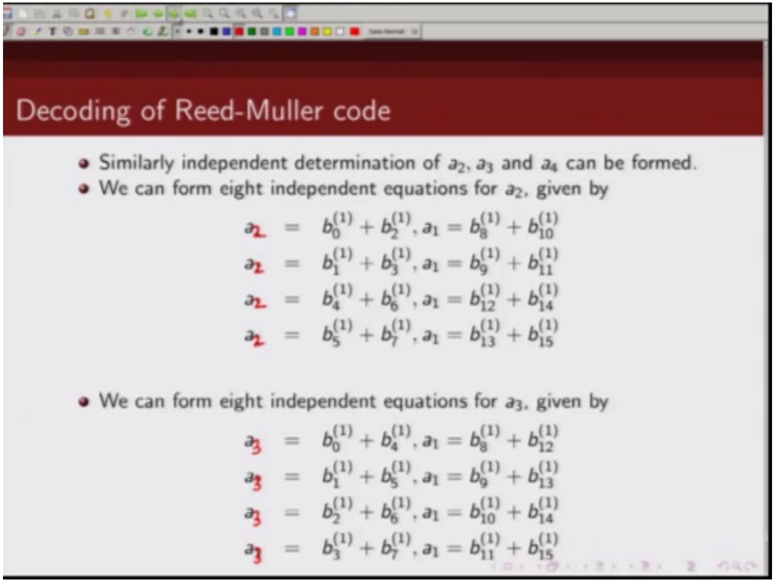
$$(b_0^{(1)}, b_1^{(1)}, \cdots, b_{15}^{(1)}) = \underline{a_0\mathbf{v_0} + a_4\mathbf{v_4} + a_3\mathbf{v_3} + a_2\mathbf{v_2} + a_1\mathbf{v_1}}$$

- 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

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

a 2, a 3, a 4 are. Again this is just the typo, this should be a 2 here and similarly this is a 3 here.

(Refer Slide Time 49:24)



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

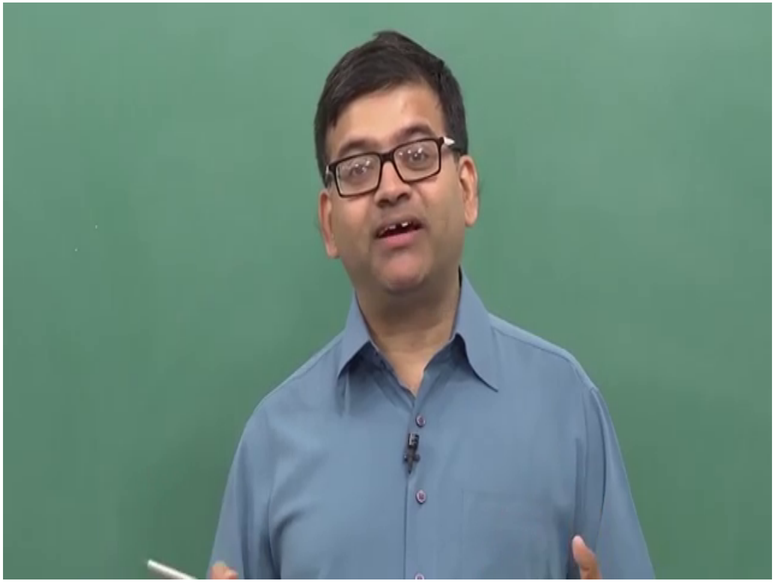
$$\begin{aligned}
a_3 &= b_0^{(1)} + b_4^{(1)}, a_1 = b_8^{(1)} + b_{12}^{(1)} \\
a_3 &= b_1^{(1)} + b_5^{(1)}, a_1 = b_9^{(1)} + b_{13}^{(1)} \\
a_3 &= b_2^{(1)} + b_6^{(1)}, a_1 = b_{10}^{(1)} + b_{14}^{(1)} \\
a_3 &= b_3^{(1)} + b_7^{(1)}, a_1 = b_{11}^{(1)} + b_{15}^{(1)}
\end{aligned}$$

And this is a 4 here, Ok. Now this is exactly same procedure I followed

(Refer Slide Time 49:35)

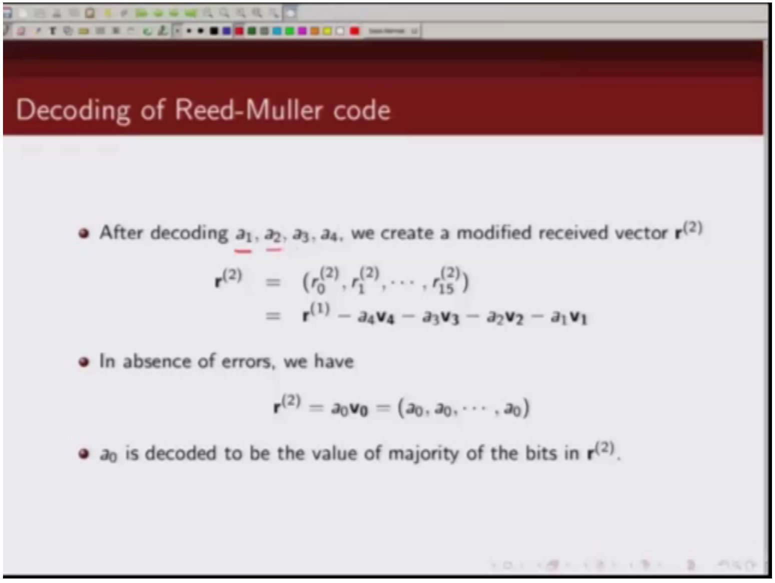
for a 1, we are using for a 2, a 3, a 4. a 4 and then we are getting independent equations, 8 independent equations and we take majority

(Refer Slide Time 49:45)



decisions in decoding these. Now once we have decoded a 1, a 2, a 3,

(Refer Slide Time 49:53)



Decoding of Reed-Muller code

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

$$\begin{aligned} \mathbf{r}^{(2)} &= (r_0^{(2)}, r_1^{(2)}, \cdots, r_{15}^{(2)}) \\ &= \mathbf{r}^{(1)} - a_4\mathbf{v_4} - a_3\mathbf{v_3} - a_2\mathbf{v_2} - a_1\mathbf{v_1} \end{aligned}$$

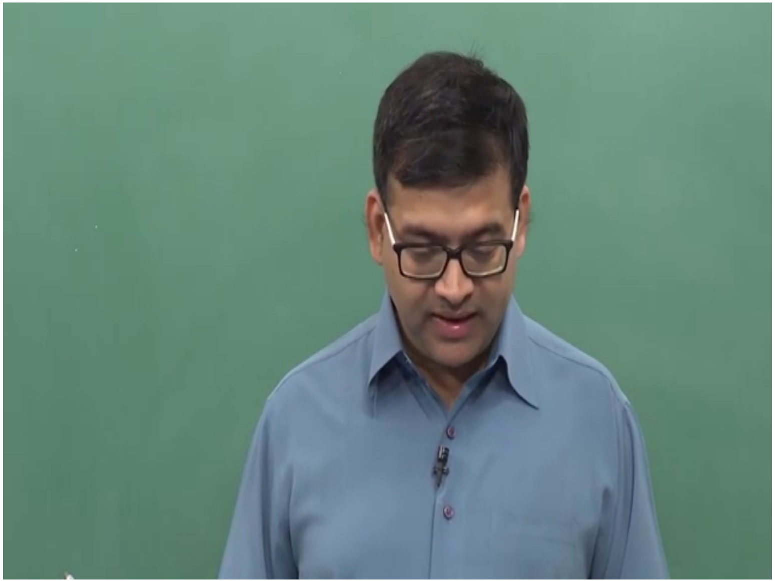
- In absence of errors, we have

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

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

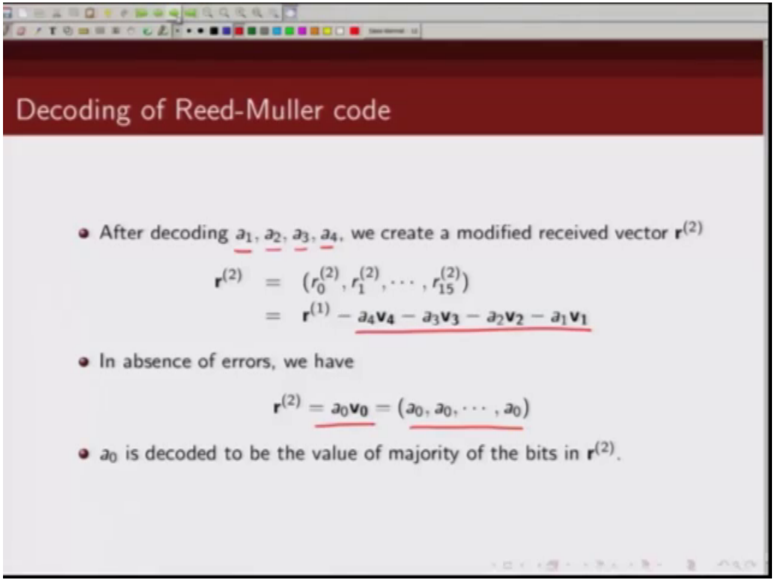
a 4 we will then remove the contribution of this from the received sequence. So our received sequence r 1 we remove this. So what we are now left is the term containing v 0. So we are only left with a 0. So now we have 16 opinion about a 0 and again we take a majority decision and that's how we decide in favor of a 0.

(Refer Slide Time 50:22)



So this in a nutshell

(Refer Slide Time 50:25)



Decoding of Reed-Muller code

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

$$\begin{aligned}\mathbf{r}^{(2)} &= (r_0^{(2)}, r_1^{(2)}, \cdots, r_{15}^{(2)}) \\ &= \mathbf{r}^{(1)} \underline{- a_4\mathbf{v_4} - a_3\mathbf{v_3} - a_2\mathbf{v_2} - a_1\mathbf{v_1}}\end{aligned}$$

- In absence of errors, we have

$$\mathbf{r}^{(2)} = \underline{a_0\mathbf{v_0}} = (\underline{a_0, a_0, \cdots, a_0})$$

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

how we are decoding Reed-Muller code.

So first we tried to decode rth order terms,

(Refer Slide Time 50:33)

then r minus 1 and like that and the key is, look at the generator matrix and from there uh 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.