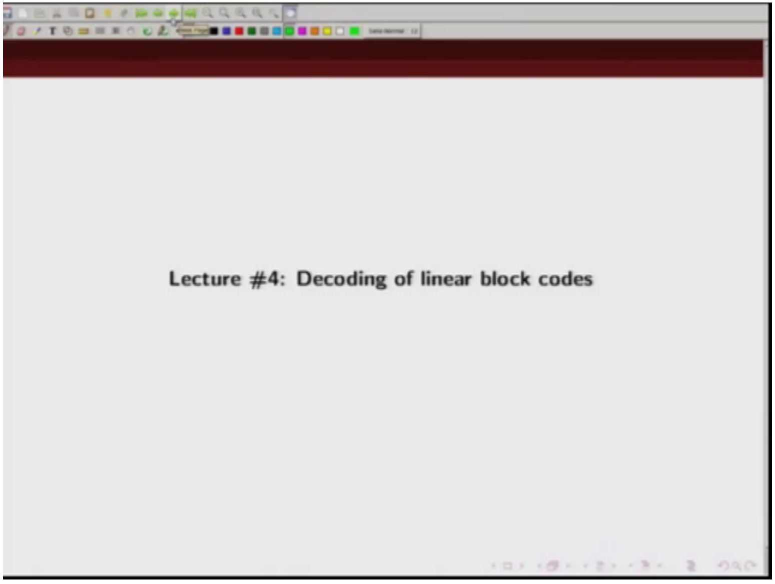
An Introduction to Coding Theory
Professor Adrish Banerji
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Module 02
Lecture Number 07
Decoding of Linear Block Codes

(Refer Slide Time 00:13)

Today we are going to talk about

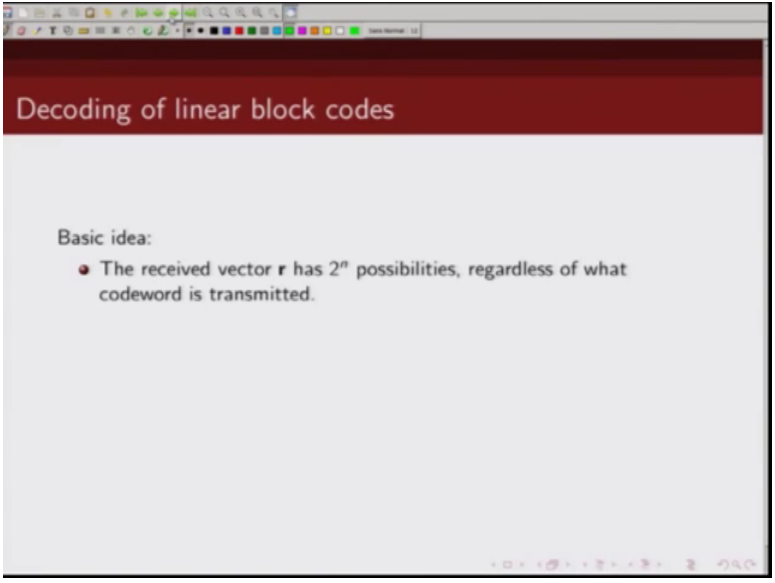
(Refer Slide Time 00:15)



Lecture #4: Decoding of linear block codes

how to decode and we are going to talk about what is known as Syndrome decoding. So what is a decoding problem?

(Refer Slide Time 00:22)



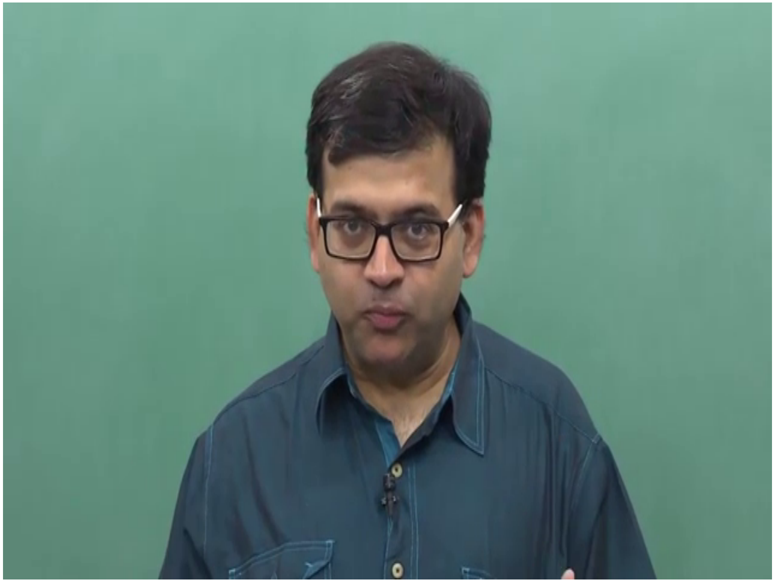
Decoding of linear block codes

Basic idea:

- The received vector **r** has 2^n possibilities, regardless of what codeword is transmitted.

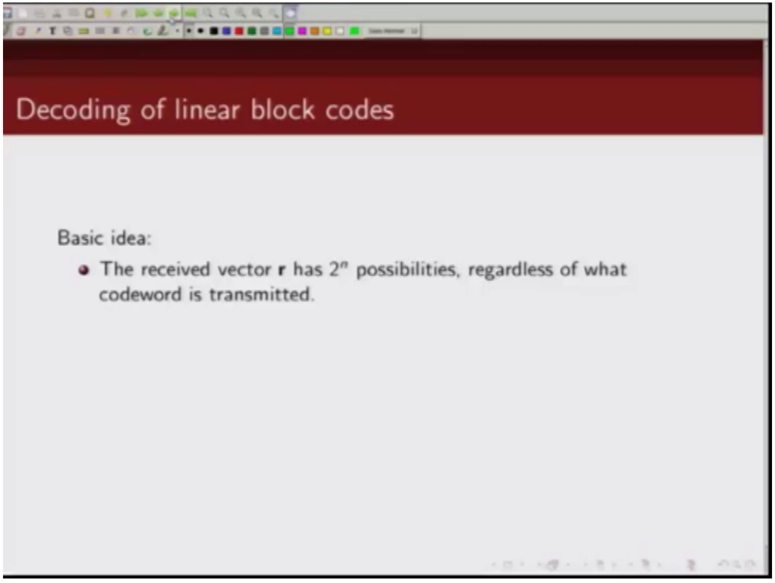
So let us look at scenario where we are transmitting our codeword

(Refer Slide Time 00:28)



which is n-bit tuple over a communication channel. Let us consider a binary symmetric channel. So in a binary symmetric channel, bits 0s and 1s are transmitted over this communication channel and with probability 1 minus p they are received correctly and with crossover probability of p the bits 0s and 1s can get flipped. So the output channel is also binary. Then we have

(Refer Slide Time 00:59)



Decoding of linear block codes

Basic idea:

- The received vector **r** has 2^n possibilities, regardless of what codeword is transmitted.

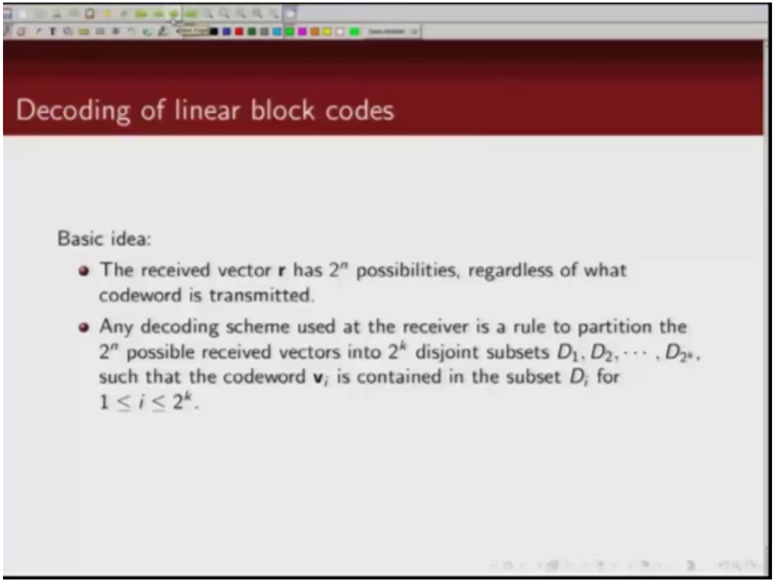
total 2 k codewords, right? And if you are looking at output received sequence, we can have total of 2 to the power n different possibilities.

(Refer Slide Time 01:09)

Because we, our code bit is, codeword is n-bit and each location can be 0s or 1s. So our decoding problem is we have to partition these 2 n possibilities into sets of 2 k, into 2 k sets. So we have to map these 2 n possibilities into set, 2 k sets and corresponding to each set there should be only one unique codeword. In other words, when we map a received sequence to a particular set, we should be able to say if these set of received sequence are obtained or we get these received sequence then corresponding to these sequence there is only one codeword. So the decoding problem is we have to partition these 2 n different possibilities into total 2 k sets such that in each of the set, there is only one valid codeword. So whenever any of these received sequence, you know is mapped to a particular set, then we should be able to say Ok, if you receive any of these code, receive any of these sequences then the transmitted codeword was this. So you can think of, we have total 2 n possibilities and we want to partition these 2 n possibilities into 2 to the power k different balls in some sense and each ball has 1 codeword associated with it. That's basically our decoding problem.

So how do we partition these 2 n different possibilities into 2 k sets is what we are going to talk about.

(Refer Slide Time 03:08)



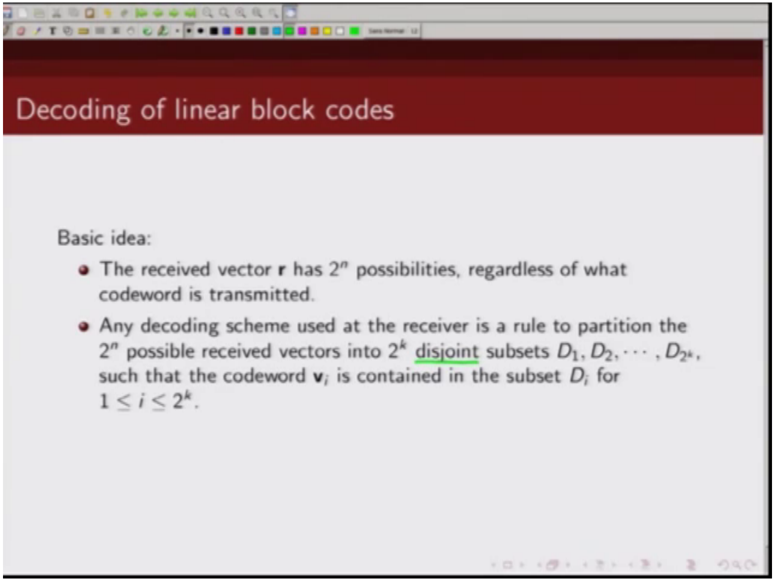
Decoding of linear block codes

Basic idea:

- The received vector **r** has 2^n possibilities, regardless of what codeword is transmitted.
- Any decoding scheme used at the receiver is a rule to partition the 2^n possible received vectors into 2^k disjoint subsets $D_1, D_2, \cdots, D_{2^k}$, such that the codeword $\mathbf{v}_i$ is contained in the subset D_i for $1 \leq i \leq 2^k$.

So any decoding scheme is basically to partition these 2 n possible codewords, received codewords into 2 k disjoint subsets. The word disjoint is important

(Refer Slide Time 03:23)



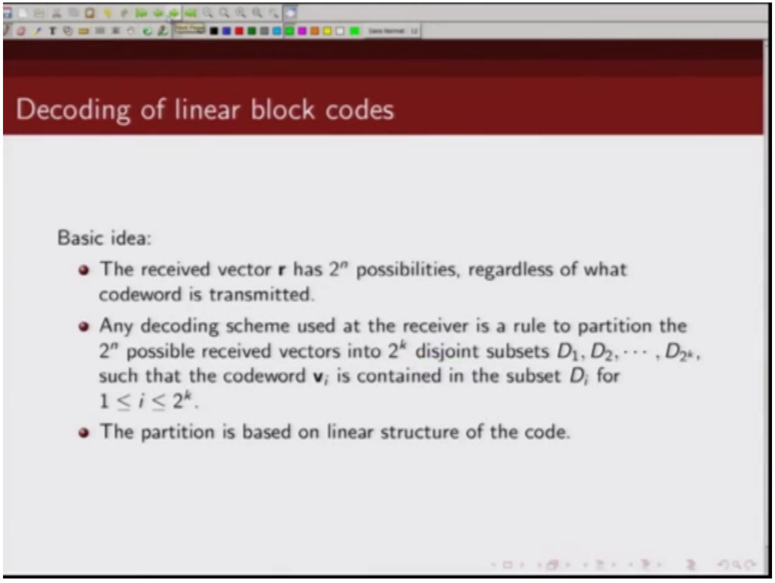
Decoding of linear block codes

Basic idea:

- The received vector $\mathbf{r}$ has 2^n possibilities, regardless of what codeword is transmitted.
- Any decoding scheme used at the receiver is a rule to partition the 2^n possible received vectors into 2^k disjoint subsets $D_1, D_2, \cdots, D_{2^k}$, such that the codeword $\mathbf{v}_i$ is contained in the subset D_i for $1 \leq i \leq 2^k$.

because we want to; we don’t want basically overlapping decision sets, regions Ok.

(Refer Slide Time 03:38)



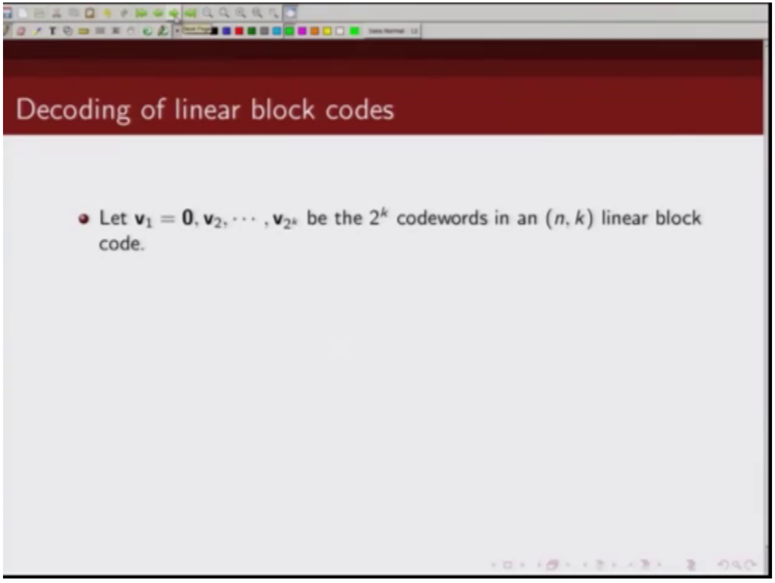
Decoding of linear block codes

Basic idea:

- The received vector $\mathbf{r}$ has 2^n possibilities, regardless of what codeword is transmitted.
- Any decoding scheme used at the receiver is a rule to partition the 2^n possible received vectors into 2^k disjoint subsets $D_1, D_2, \cdots, D_{2^k}$, such that the codeword $\mathbf{v}_i$ is contained in the subset D_i for $1 \leq i \leq 2^k$.
- The partition is based on linear structure of the code.

And how do we do this partition is basically based on the structure of the linear block and that's what we are going to describe in this lecture.

(Refer Slide Time 03:50)

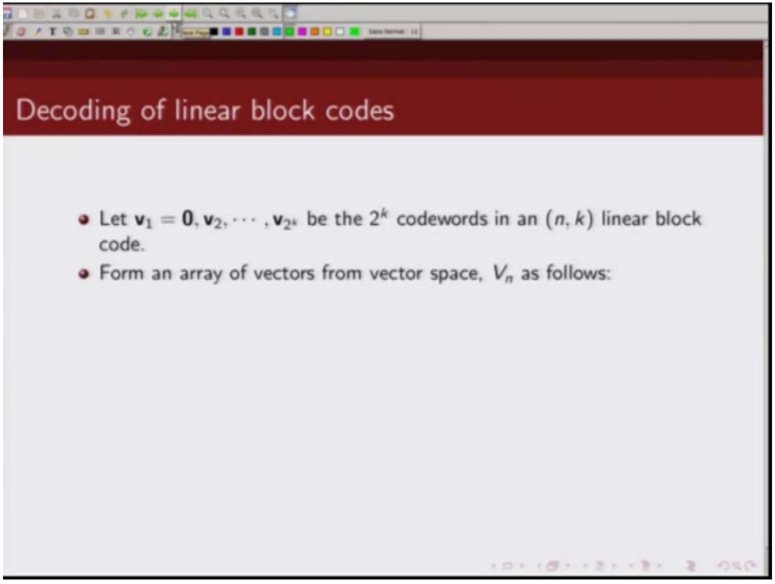


Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.

So let us denote the 2 k codewords by v 1, v 2, v 3, v 2 k, v 2 raised to power k. So let's say these are my 2 k codewords in a linear block code, in n k linear block code where

(Refer Slide Time 04:07)

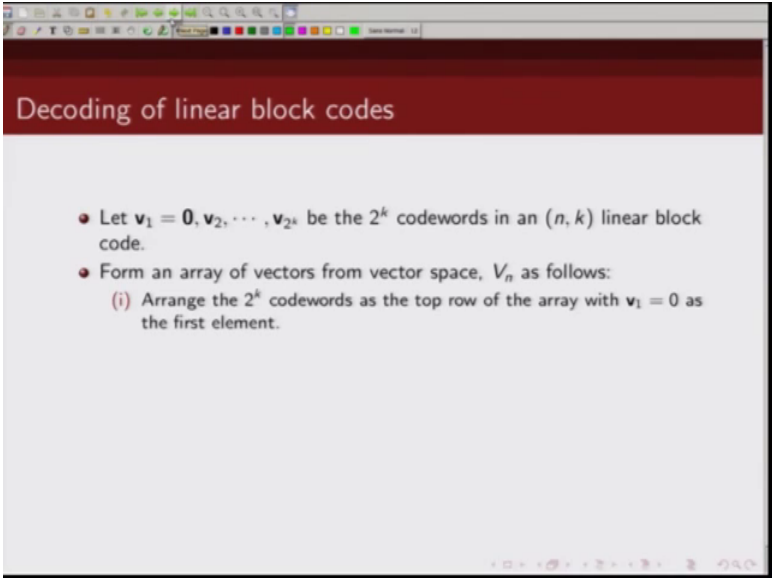


Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:

v 1 is all zero codeword. Now what do we do is we form an array of vectors from vector space v n as follows. In the top row

(Refer Slide Time 04:20)



Decoding of linear block codes

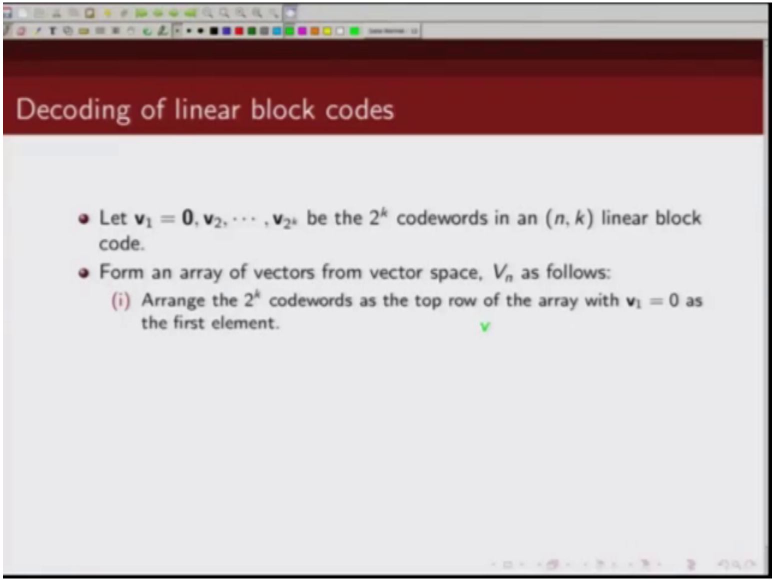
- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
  - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = 0$ as the first element.

of this array, we will arrange all of these 2 k codewords with all zero codeword

(Refer Slide Time 04:29)

being the leftmost entry in the row. So we are going, in the first row of this array, we are going to put

(Refer Slide Time 04:40)



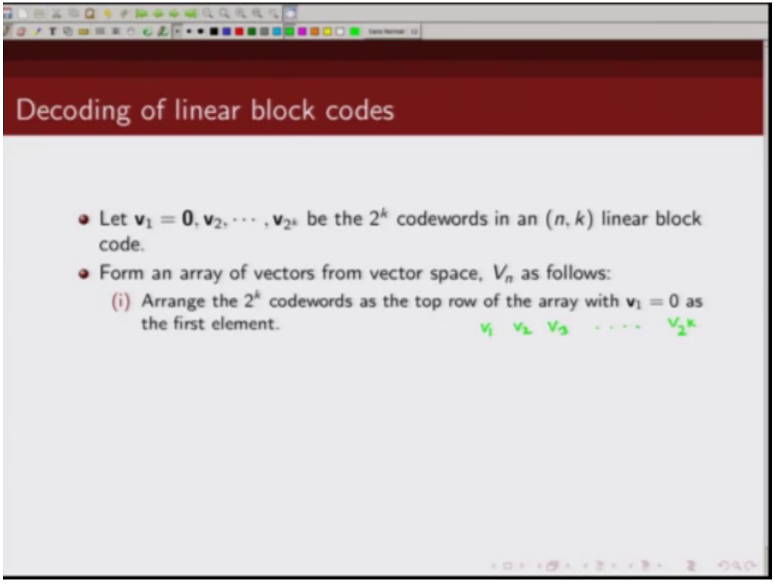
Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
  - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = 0$ as the first element.

v

all zero codeword and then we are going to put other codewords. So this will be the first row of this

(Refer Slide Time 04:53)



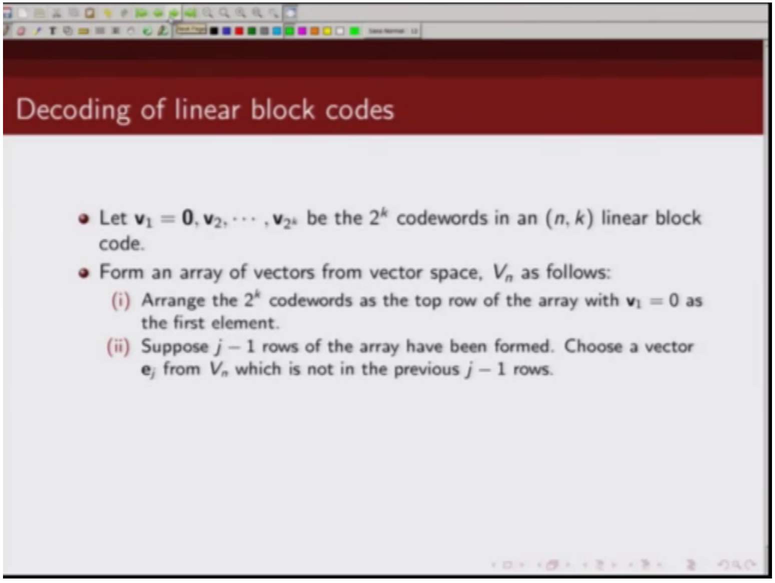
Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
  - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = 0$ as the first element.

$v_1 \quad v_2 \quad v_3 \quad \cdots \quad v_{2^k}$

array. Now suppose we have formed,

(Refer Slide Time 04:57)



Decoding of linear block codes

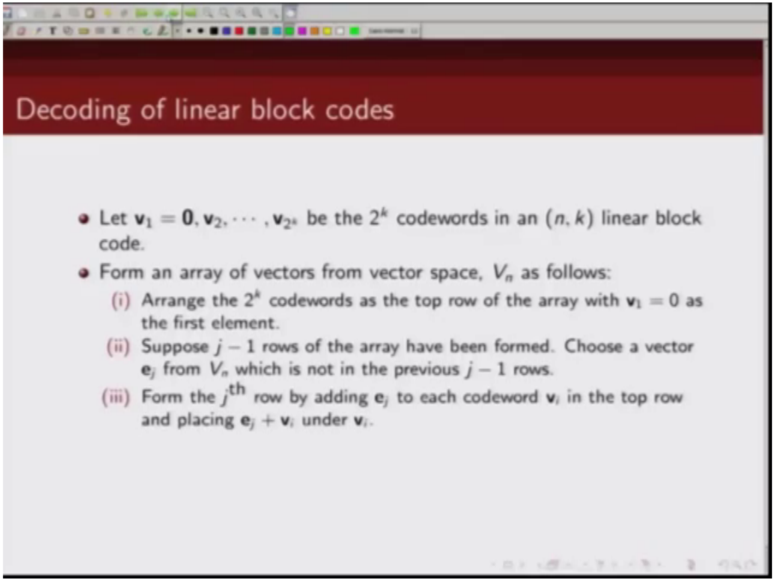
- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
  - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = 0$ as the first element.
  - (ii) Suppose $j - 1$ rows of the array have been formed. Choose a vector $\mathbf{e}_j$ from V_n which is not in the previous $j - 1$ rows.

I am going to tell you how we are going to form this array. Suppose let’s say we have already formed j minus 1 rows of this array then what do we do? We choose a vector e j from vector space v n so we pick up a n-bit error vector which has not been chosen

(Refer Slide Time 05:19)

previously in any of the previous j minus 1 rows. We pick that error vector. Next

(Refer Slide Time 05:30)

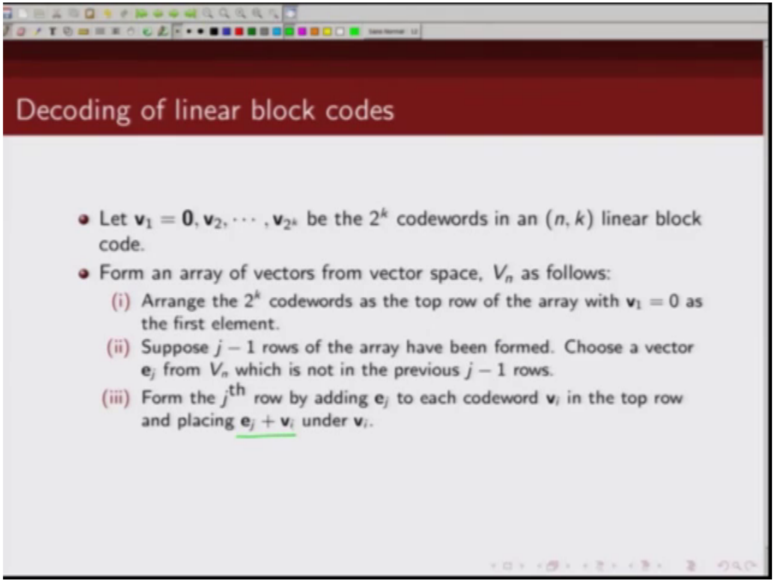


Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
  - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = 0$ as the first element.
  - (ii) Suppose $j-1$ rows of the array have been formed. Choose a vector $\mathbf{e}_j$ from V_n which is not in the previous $j-1$ rows.
  - (iii) Form the j^{th} row by adding $\mathbf{e}_j$ to each codeword $\mathbf{v}_i$ in the top row and placing $\mathbf{e}_j + \mathbf{v}_i$ under $\mathbf{v}_i$.

we form the jth row by adding that error vector to each of the codewords on the top row and placing the new vector

(Refer Slide Time 05:43)

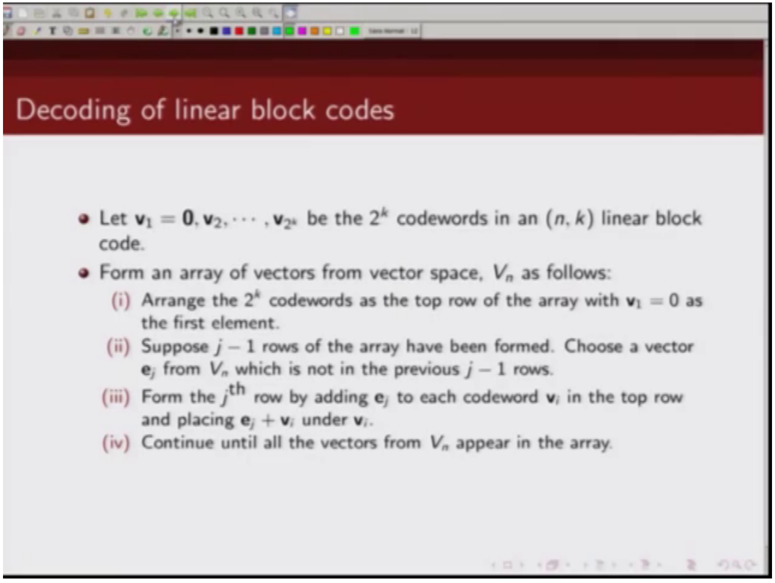


Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
  - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = 0$ as the first element.
  - (ii) Suppose $j-1$ rows of the array have been formed. Choose a vector $\mathbf{e}_j$ from V_n which is not in the previous $j-1$ rows.
  - (iii) Form the j^{th} row by adding $\mathbf{e}_j$ to each codeword $\mathbf{v}_i$ in the top row and placing $\underline{\mathbf{e}_j + \mathbf{v}_i}$ under $\mathbf{v}_i$.

which is e j plus v i under the codeword v i. I will just explain what I mean

(Refer Slide Time 05:52)

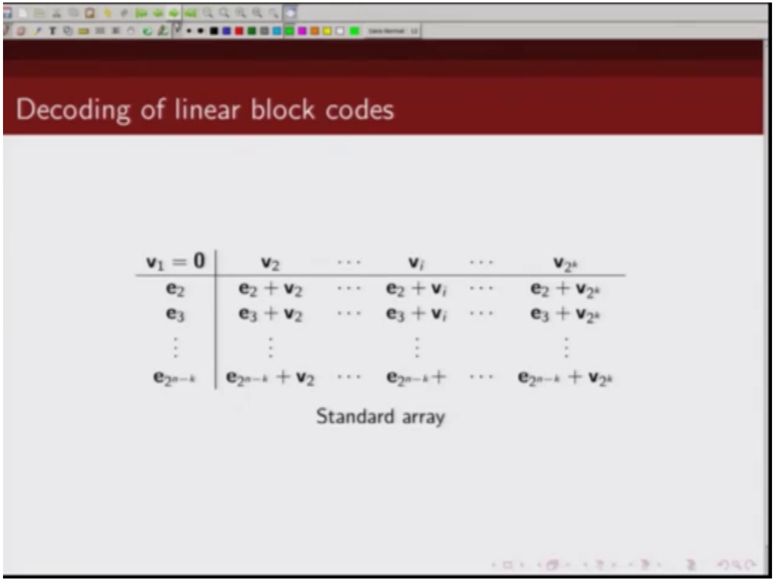


Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
  - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = 0$ as the first element.
  - (ii) Suppose $j - 1$ rows of the array have been formed. Choose a vector $\mathbf{e}_j$ from V_n which is not in the previous $j - 1$ rows.
  - (iii) Form the j^{th} row by adding $\mathbf{e}_j$ to each codeword $\mathbf{v}_i$ in the top row and placing $\mathbf{e}_j + \mathbf{v}_i$ under $\mathbf{v}_i$.
  - (iv) Continue until all the vectors from V_n appear in the array.

by this.

(Refer Slide Time 05:53)



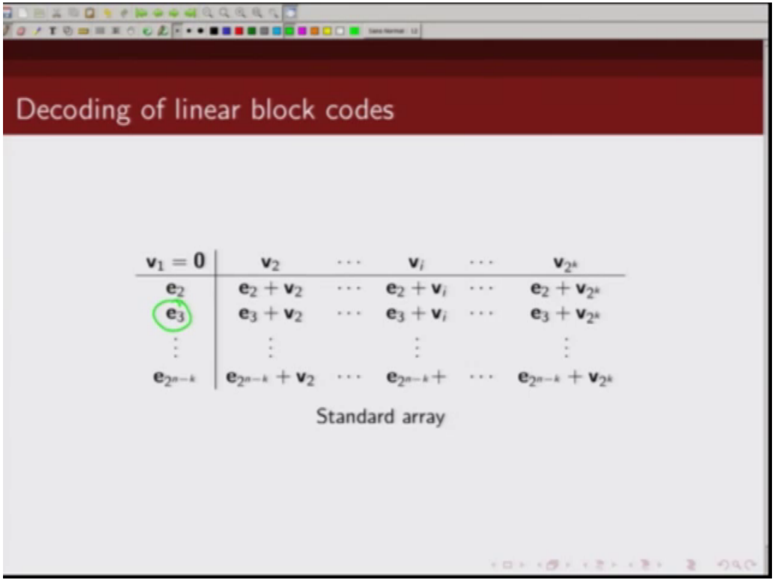
Decoding of linear block codes

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\mathbf{e}_3 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} +$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

Standard array

So as I said, in the first row, we have listed all the codewords with all zero codeword as my leftmost entry. Now let's say I have already formed some rows and I want to form jth row. So how do I find jth row? I will pick up an error vector e j

(Refer Slide Time 06:22)



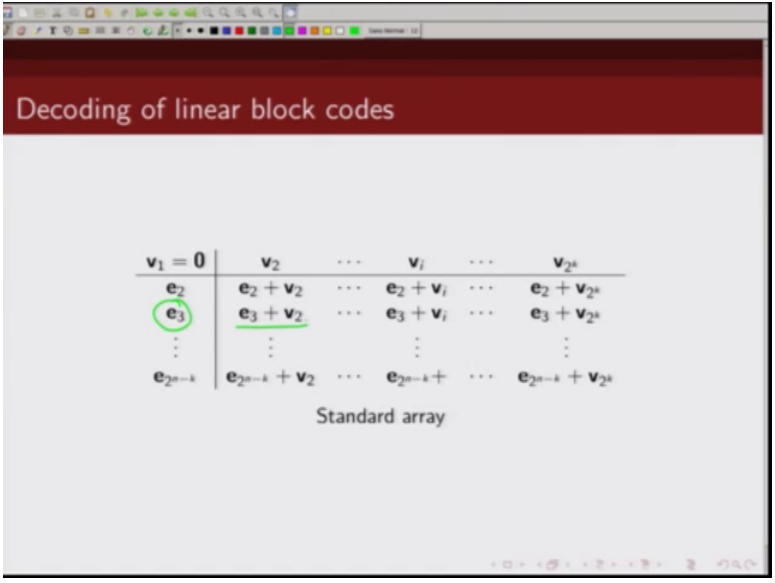
Decoding of linear block codes

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\mathbf{e}_3 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} +$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

Standard array

which is; let us say e 3, which has not appeared before; so e 3 should not be any of these elements which I have already been chosen. e 3 should not have appeared in the previous rows of this array. So I choose such a error vector and then what do I do is I add this error vector to each of these elements in the first row which is nothing but codewords and I place that element under the same column. Now what do I mean by that? Let's say this was v 2. So I will add e 3 to v 2 and I will add the element e 3 plus v 2 in the

(Refer Slide Time 07:11)



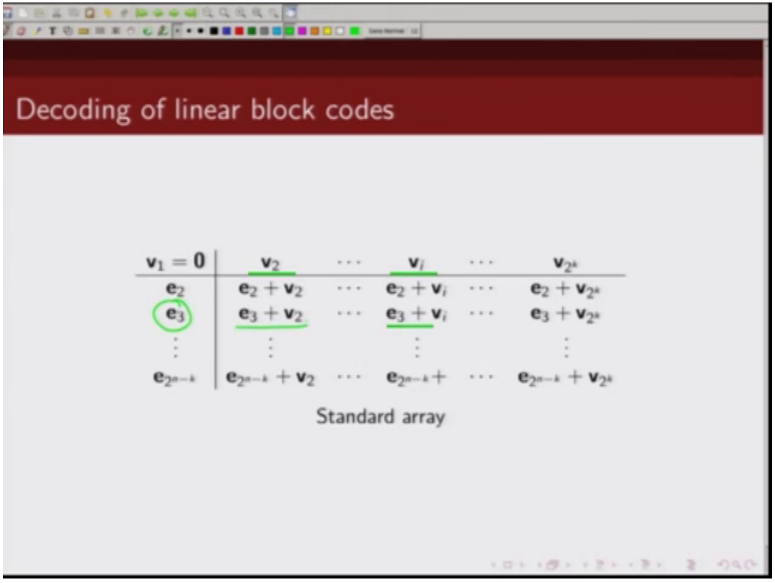
Decoding of linear block codes

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\underline{\mathbf{e}_3 + \mathbf{v}_2}$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} +$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

Standard array

same column as v 2 was. Similarly if I have a codeword v i I will add e 3

(Refer Slide Time 07:21)



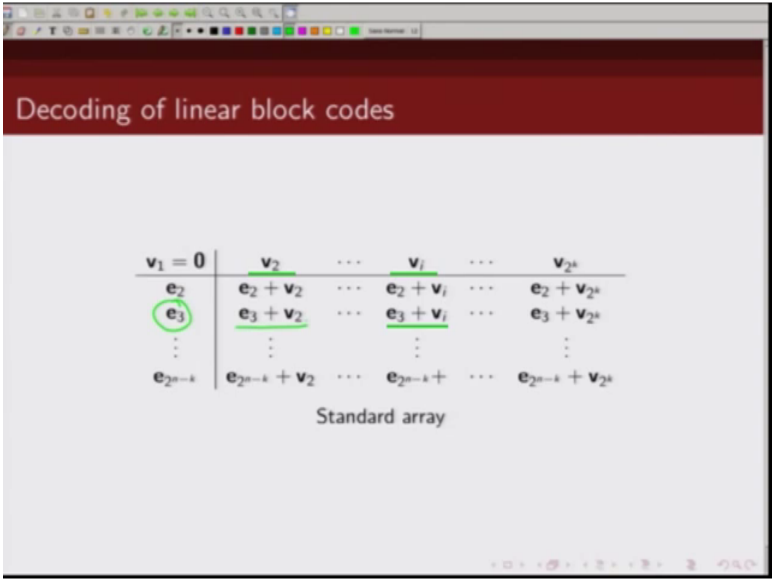
Decoding of linear block codes

$v_1 = 0$	$\underline{v_2}$	$\cdots$	$\underline{v_i}$	$\cdots$	v_{2^k}
e_2	$e_2 + v_2$	$\cdots$	$e_2 + v_i$	$\cdots$	$e_2 + v_{2^k}$
e_3	$\underline{e_3 + v_2}$	$\cdots$	$\underline{e_3 + v_i}$	$\cdots$	$e_3 + v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	$\cdots$	$e_{2^{n-k}} +$	$\cdots$	$e_{2^{n-k}} + v_{2^k}$

Standard array

to v i and I will

(Refer Slide Time 07:23)



Decoding of linear block codes

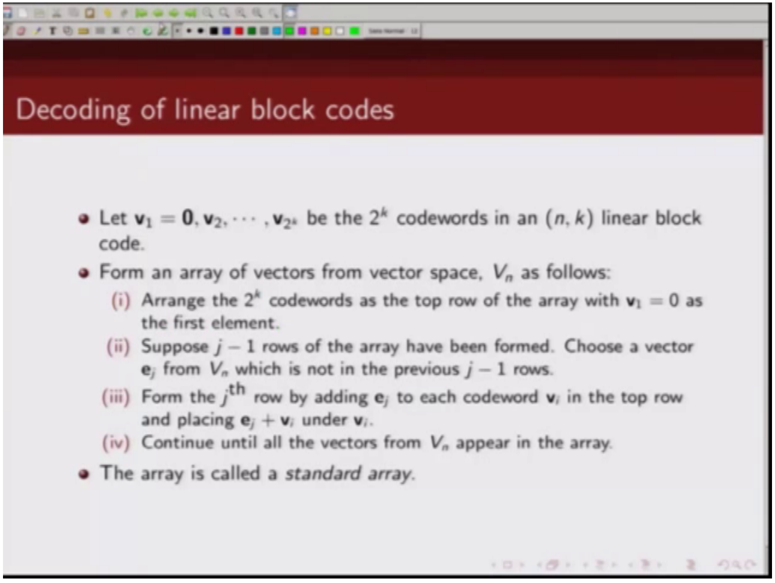
$v_1 = 0$	$\underline{v_2}$	$\cdots$	$\underline{v_i}$	$\cdots$	v_{2^k}
e_2	$e_2 + v_2$	$\cdots$	$e_2 + v_i$	$\cdots$	$e_2 + v_{2^k}$
e_3	$\underline{e_3 + v_2}$	$\cdots$	$\underline{e_3 + v_i}$	$\cdots$	$e_3 + v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	$\cdots$	$e_{2^{n-k}} +$	$\cdots$	$e_{2^{n-k}} + v_{2^k}$

Standard array

add this element e 3 plus v i in the same column as v i. So this is how I am going to build up this row in this array.

The next row, how do I build up? Again I will pick an error pattern which has not happened before. I will pick up that error pattern and then I will add that error pattern to v 2, put that element here. Add that error pattern to v i, put that pattern here. So this is how I will fill up the entries in this array.

(Refer Slide Time 08:05)



Decoding of linear block codes

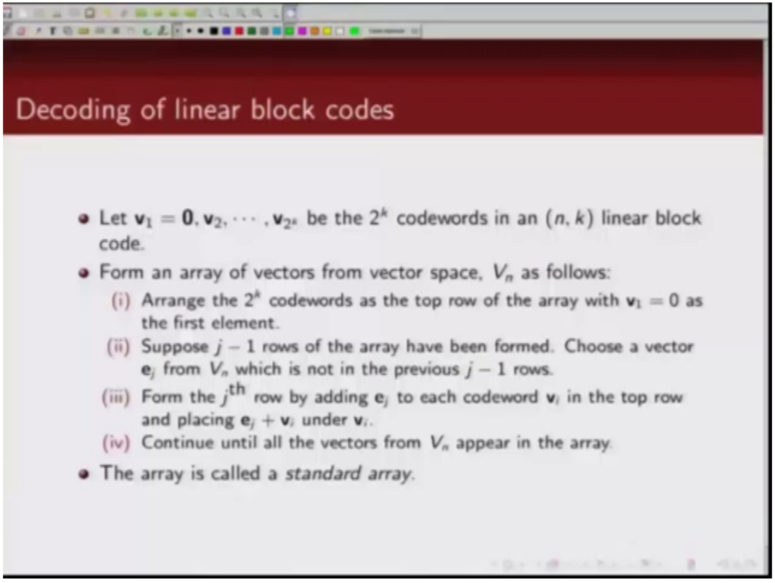
- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
  - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = 0$ as the first element.
  - (ii) Suppose $j - 1$ rows of the array have been formed. Choose a vector $\mathbf{e}_j$ from V_n which is not in the previous $j - 1$ rows.
  - (iii) Form the j^{th} row by adding $\mathbf{e}_j$ to each codeword $\mathbf{v}_i$ in the top row and placing $\mathbf{e}_j + \mathbf{v}_i$ under $\mathbf{v}_i$.
  - (iv) Continue until all the vectors from V_n appear in the array.
- The array is called a *standard array*.

So that's what I meant. Form the jth row by adding e j to each of these codewords in the top row and placing e j plus v i under the same column as v i and we will continue doing this until all end bit vectors have been put

(Refer Slide Time 08:27)

in this standard array. And this array formed in this

(Refer Slide Time 08:33)

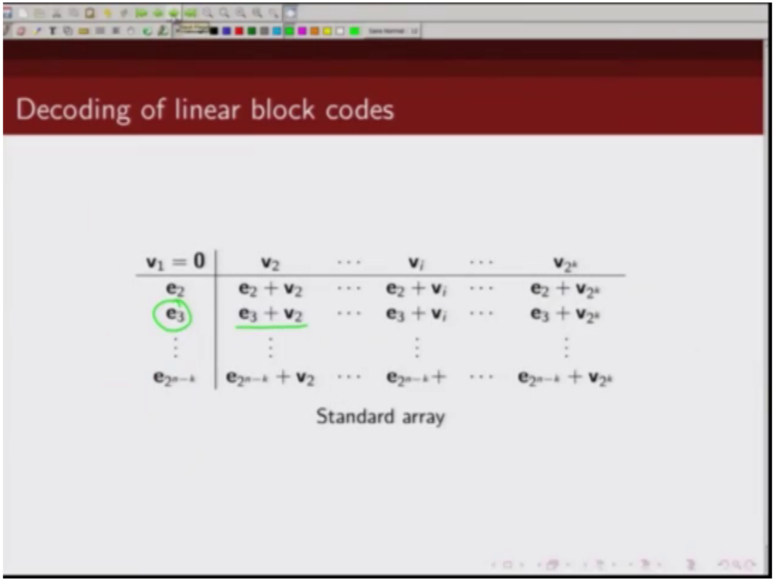


Decoding of linear block codes

- Let $\mathbf{v}_1 = \mathbf{0}, \mathbf{v}_2, \cdots, \mathbf{v}_{2^k}$ be the 2^k codewords in an (n, k) linear block code.
- Form an array of vectors from vector space, V_n as follows:
  - (i) Arrange the 2^k codewords as the top row of the array with $\mathbf{v}_1 = 0$ as the first element.
  - (ii) Suppose $j - 1$ rows of the array have been formed. Choose a vector $\mathbf{e}_j$ from V_n which is not in the previous $j - 1$ rows.
  - (iii) Form the j^{th} row by adding $\mathbf{e}_j$ to each codeword $\mathbf{v}_i$ in the top row and placing $\mathbf{e}_j + \mathbf{v}_i$ under $\mathbf{v}_i$.
  - (iv) Continue until all the vectors from V_n appear in the array.
- The array is called a *standard array*.

way is known as standard array. So this is how your

(Refer Slide Time 08:39)



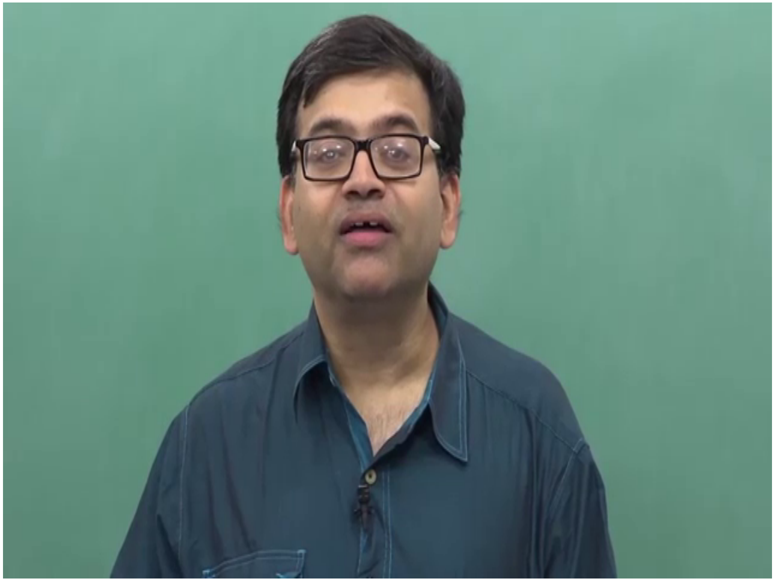
Decoding of linear block codes

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\underline{\mathbf{e}_3 + \mathbf{v}_2}$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} +$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

Standard array

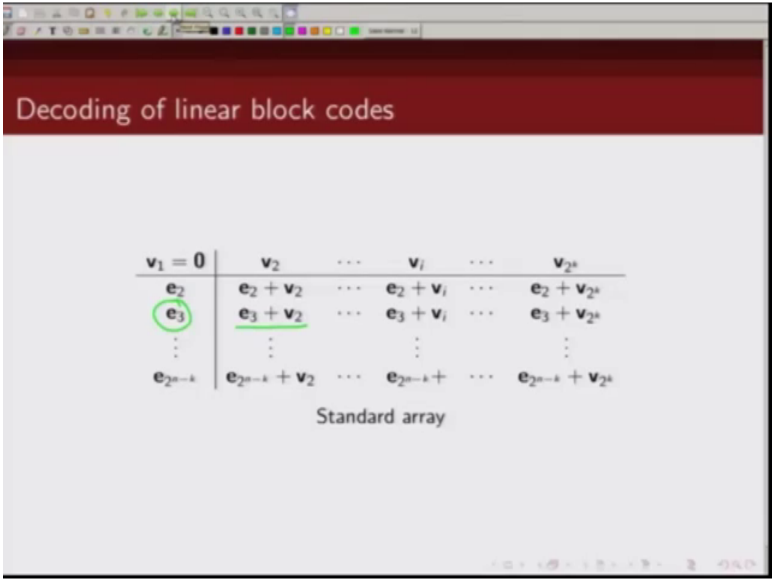
standard array will look like. Again I will just recap how we are constructing the standard array. The first row of this array is set

(Refer Slide Time 08:52)



of codewords, v 1 to v 2 raised to power k.

(Refer Slide Time 08:56)



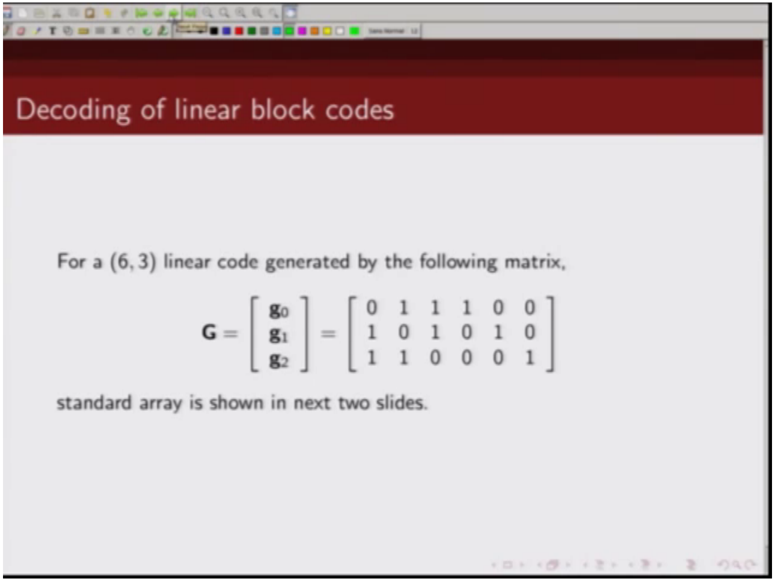
Decoding of linear block codes

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\underline{\mathbf{e}_3 + \mathbf{v}_2}$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} +$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

Standard array

And the leftmost entry here is all zero codeword. Next we pick up an element, an error vector which has not happened in any of the previous rows and then we add the error vector to each of the elements of these uh codewords and put the new element under the same column as that codeword and we continue doing that until we have put all the possible vectors in this array. So let us look at

(Refer Slide Time 09:39)



Decoding of linear block codes

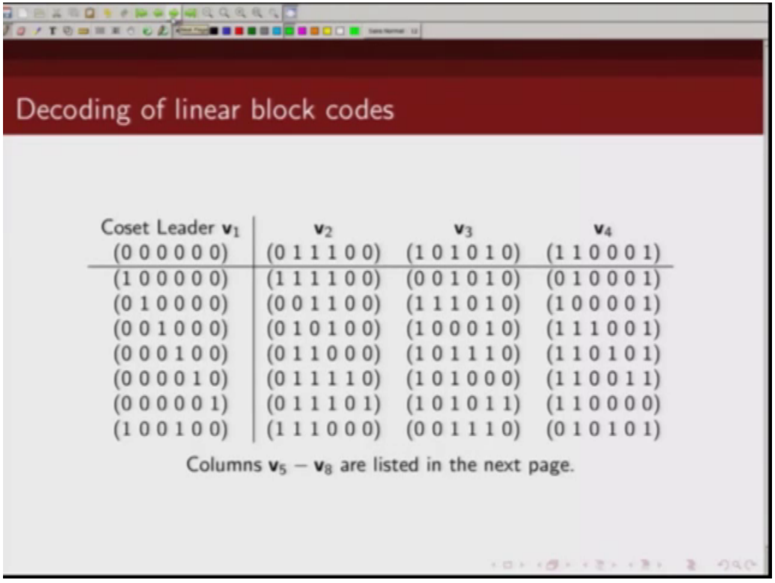
For a (6, 3) linear code generated by the following matrix,

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

standard array is shown in next two slides.

6 3 linear block code whose generator matrix is given here.

(Refer Slide Time 09:46)



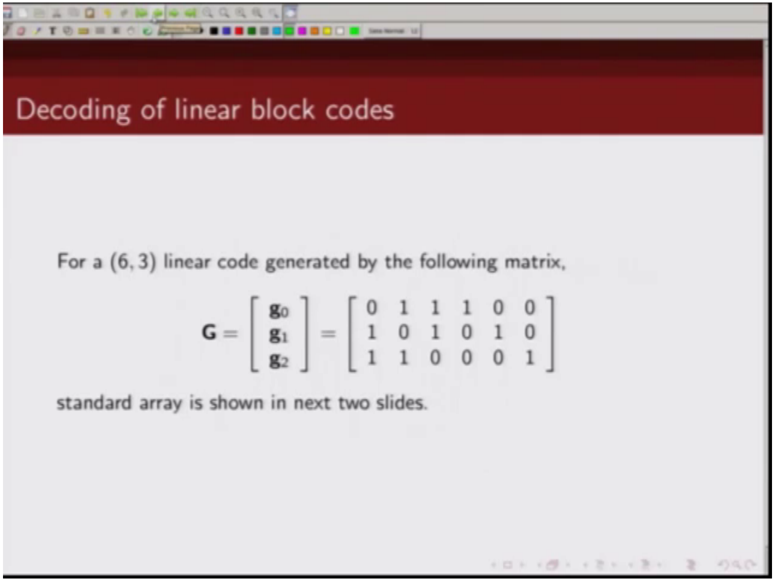
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

Now how do I find its standard array? So as I said the first step involved is

(Refer Slide Time 09:56)



Decoding of linear block codes

For a (6, 3) linear code generated by the following matrix,

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

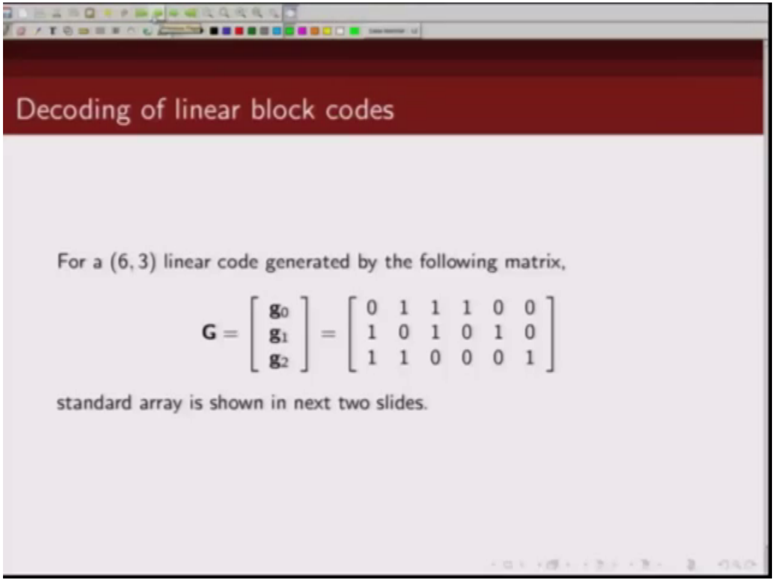
standard array is shown in next two slides.

you need to write down all possible

(Refer Slide Time 10:04)

codewords. Now you are already been given the generator matrix for this

(Refer Slide Time 10:09)



Decoding of linear block codes

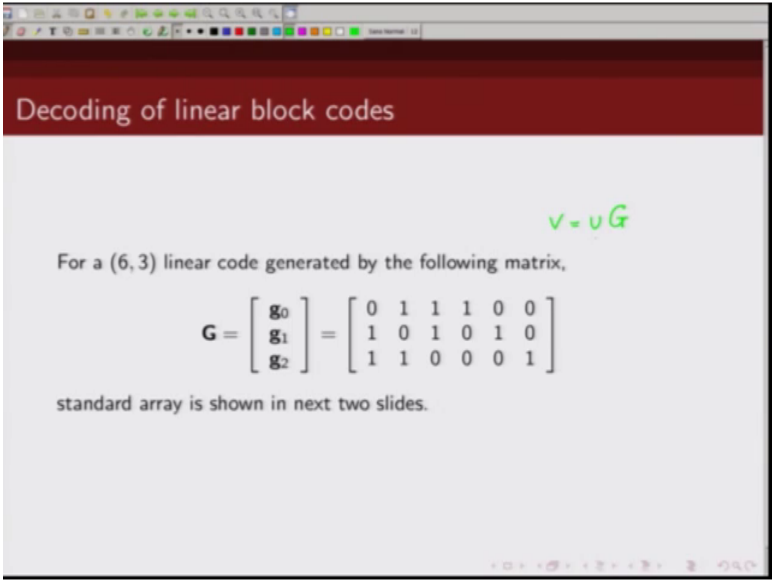
For a (6, 3) linear code generated by the following matrix,

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

standard array is shown in next two slides.

code so you can find out what are the possible codewords. You just have to do v is nothing but u times G.

(Refer Slide Time 10:18)



Decoding of linear block codes

$v = uG$

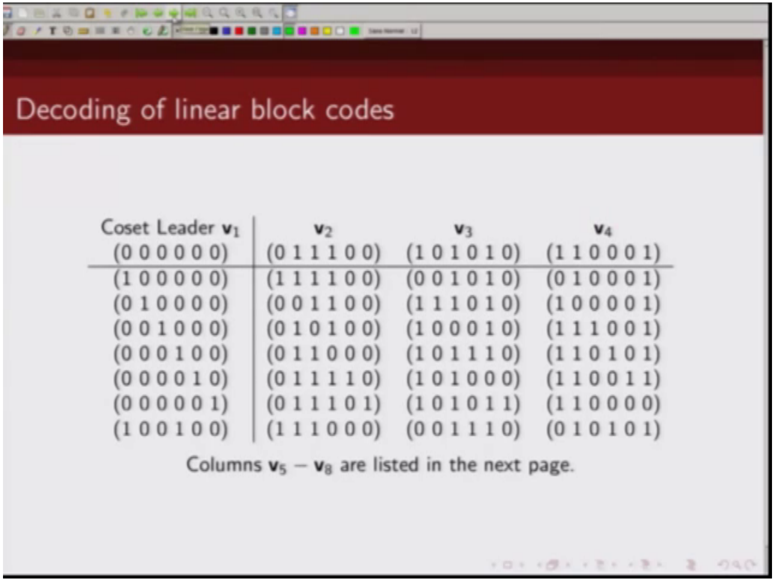
For a (6, 3) linear code generated by the following matrix,

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

standard array is shown in next two slides.

So there are total 8 codewords and you can find those 8 codewords because you know G is given to you and you know what is your u, u basically goes from 0 0 0 to 0 0 1, 0 1 0 it goes to 1 1 1. So you can find out what these codewords are and I have listed these

(Refer Slide Time 10:46)



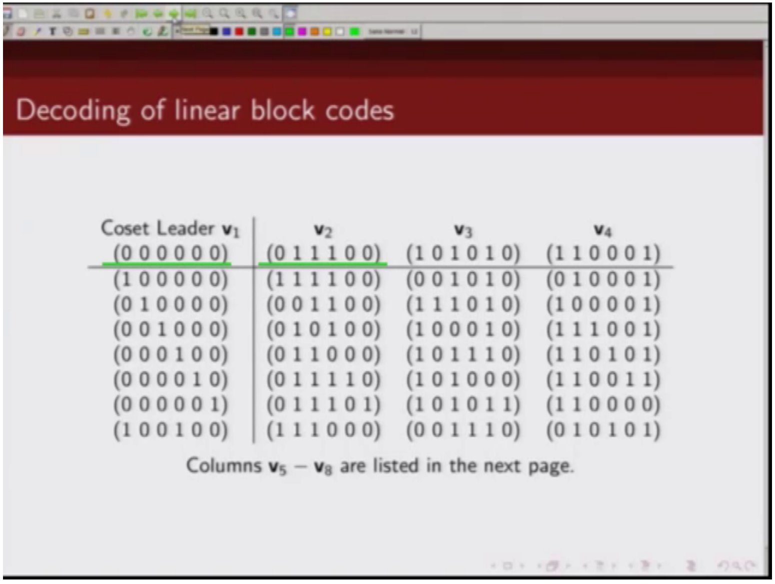
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

codewords here. So you have one all zero codeword and then you have the other codewords are 0 1 1 1 0 0,

(Refer Slide Time 10:56)



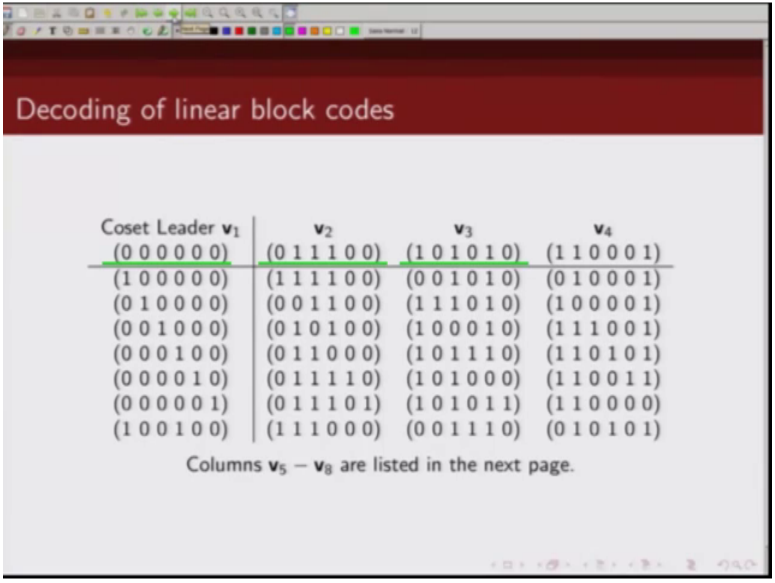
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

1 0 1 0 1 0,

(Refer Slide Time 10:57)



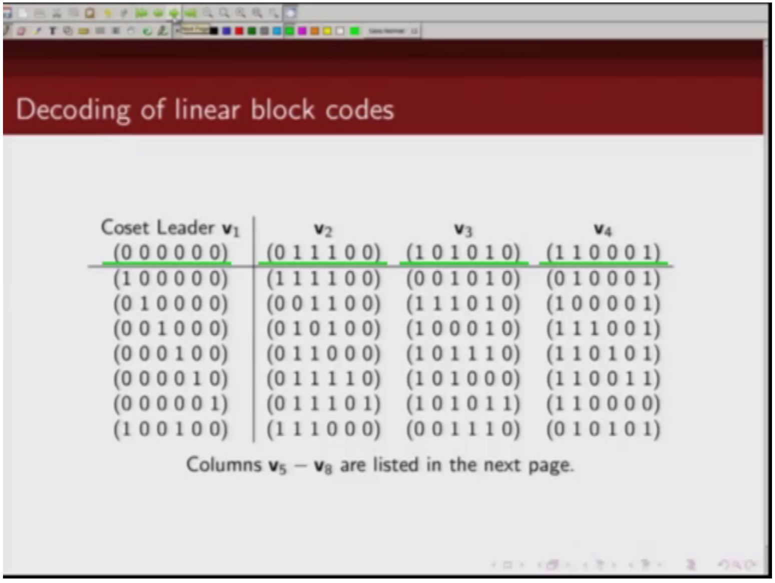
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

1 1 0 0 1,

(Refer Slide Time 11:00)



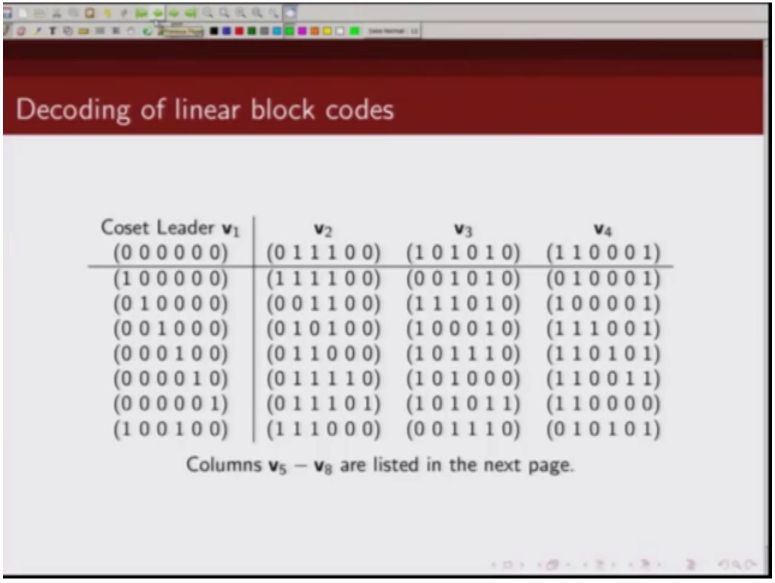
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

0 0 0 1. Since I could not fit in all the columns in one slide, I have continued it here, so this is I had up to v 0,

(Refer Slide Time 11:12)



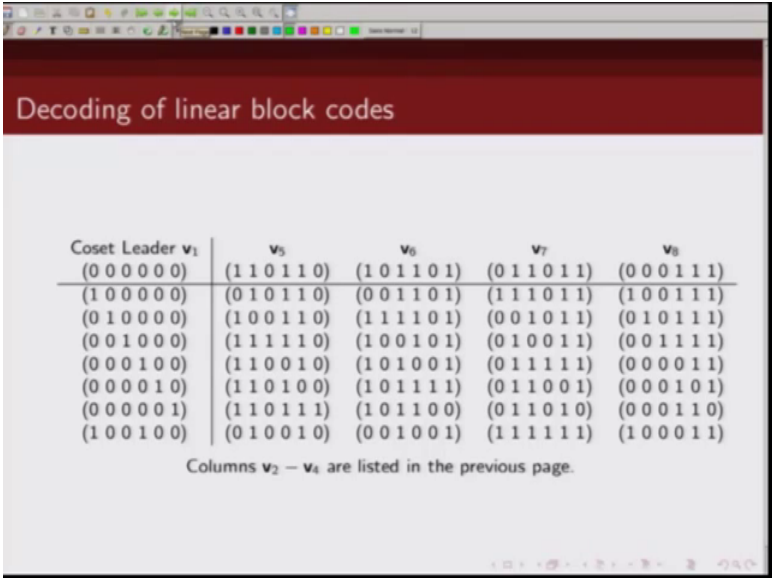
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

I have v 1, v 2, v 3, v 4. And then I had v 5, v 6, v 7, v 8.

(Refer Slide Time 11:18)



Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

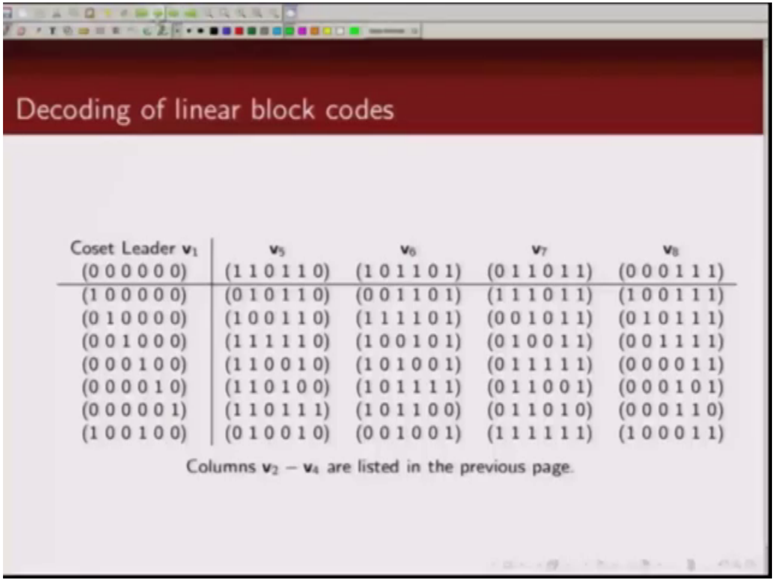
Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

So these are 8 codewords for this 6 3

(Refer Slide Time 11:24)

linear block code. Now

(Refer Slide Time 11:29)



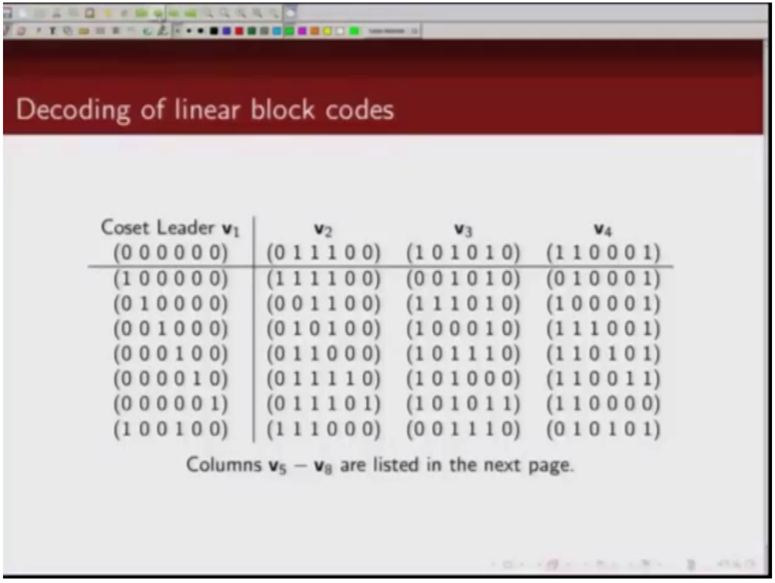
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

how do I find entries

(Refer Slide Time 11:31)



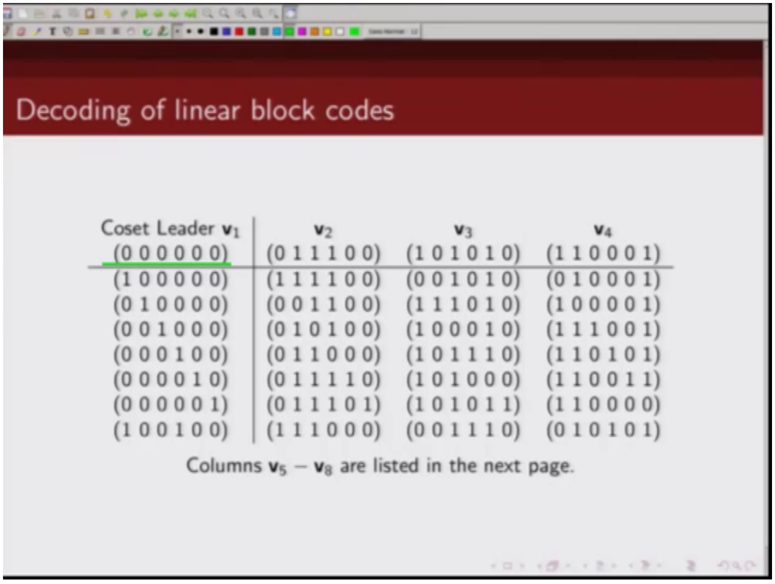
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

in the next column? As I said I have to pick up a vector which has not appeared in the previous rows. So let's look at what has appeared in the previous rows. We had all zero sequence here,

(Refer Slide Time 11:46)



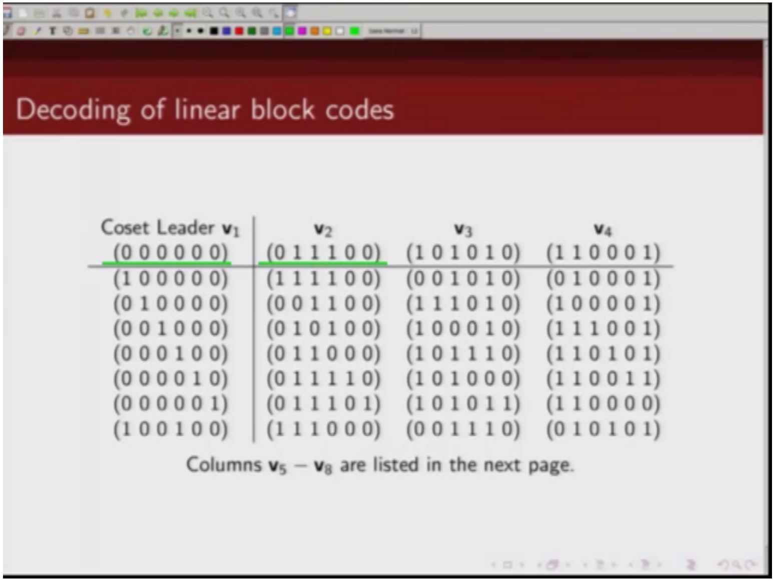
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

we had a sequence which has three 1's,

(Refer Slide Time 11:48)



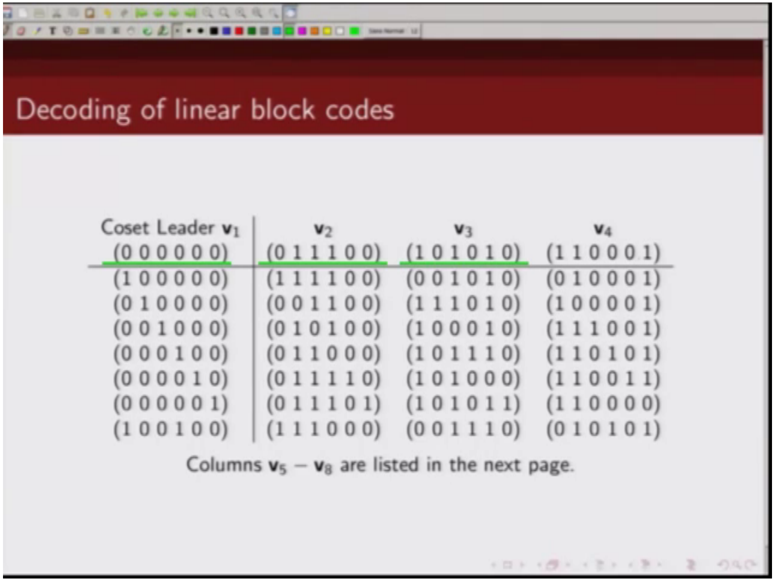
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

a sequence which has three 1's, sequence has

(Refer Slide Time 11:51)



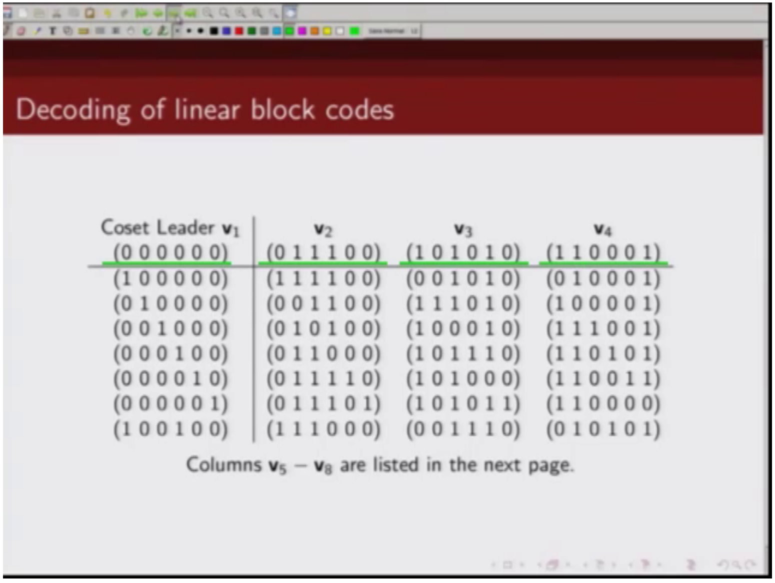
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

I mean

(Refer Slide Time 11:52)



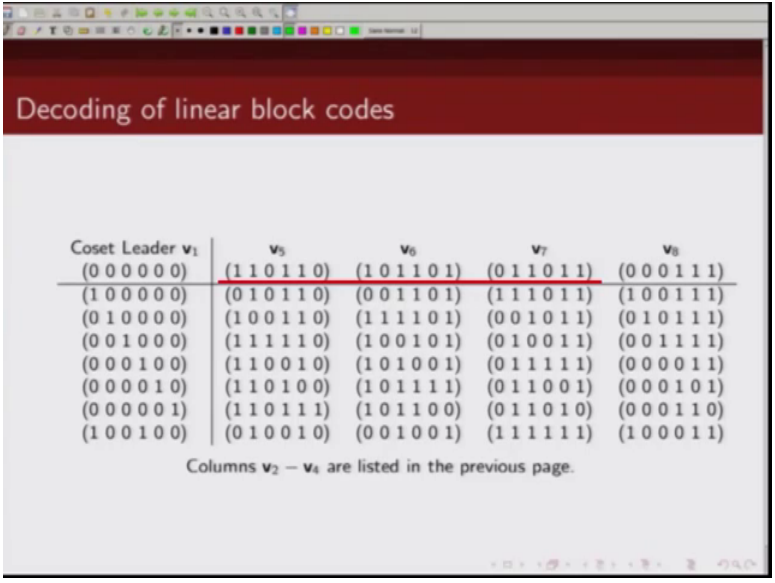
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

three 1's, codeword four 1's so we see

(Refer Slide Time 11:56)



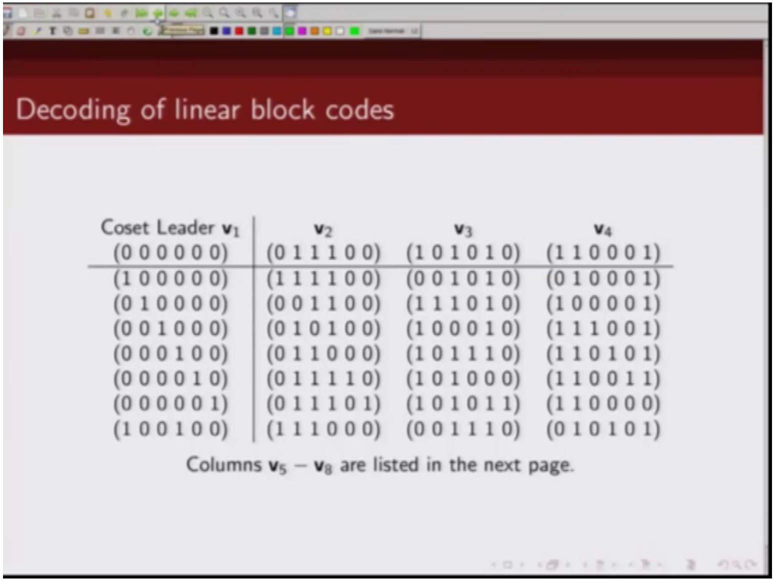
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

we don't have n-tuples which have just weight 1, Hamming weight 1, they have not appeared so far. So this

(Refer Slide Time 12:05)



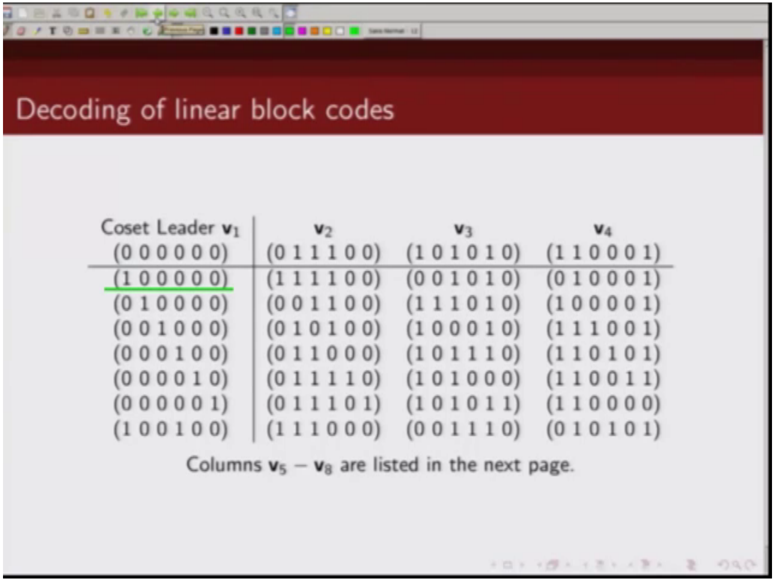
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

1 0 0 0 0 0

(Refer Slide Time 12:07)



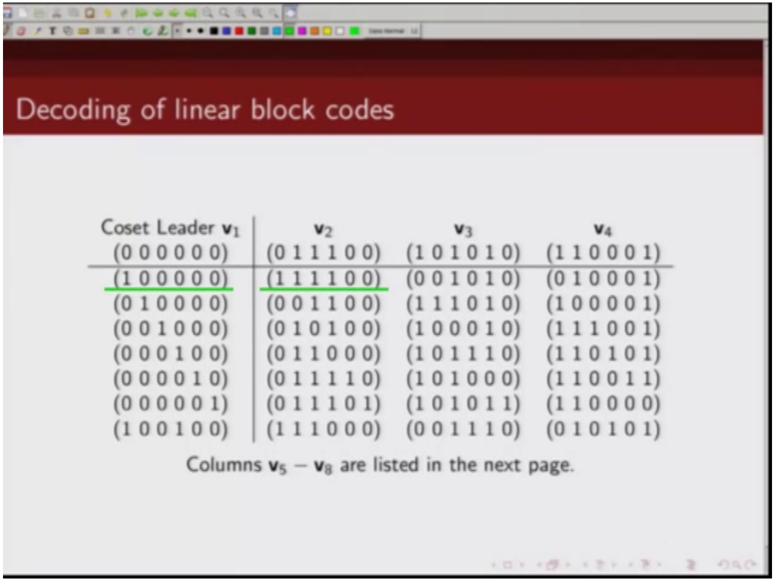
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

has not appeared so far in first row of this array. So I pick this 1 0 0 0 0 0 as my first element. Now how do I find this element? I am going to

(Refer Slide Time 12:22)



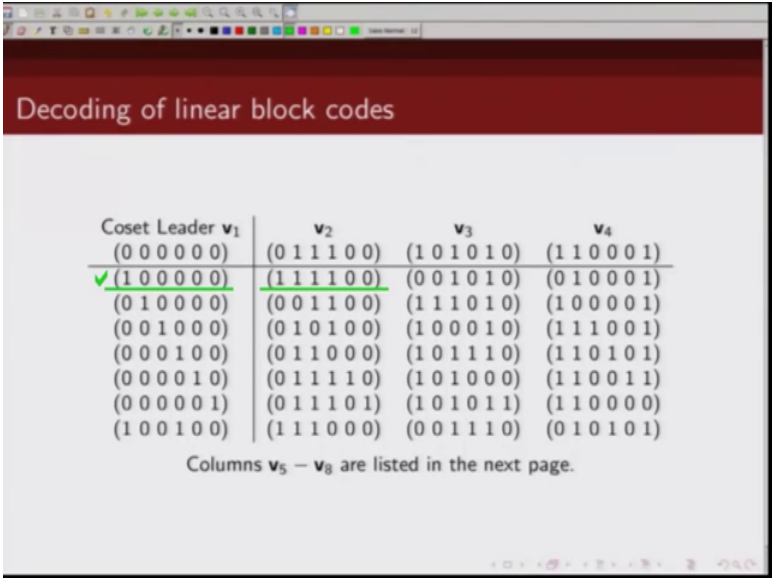
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

add this

(Refer Slide Time 12:24)



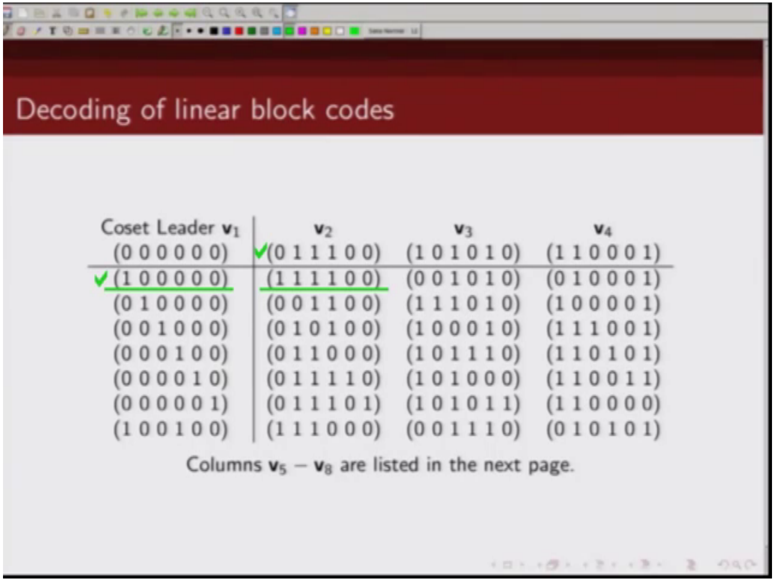
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
✓ (1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

to this.

(Refer Slide Time 12:25)



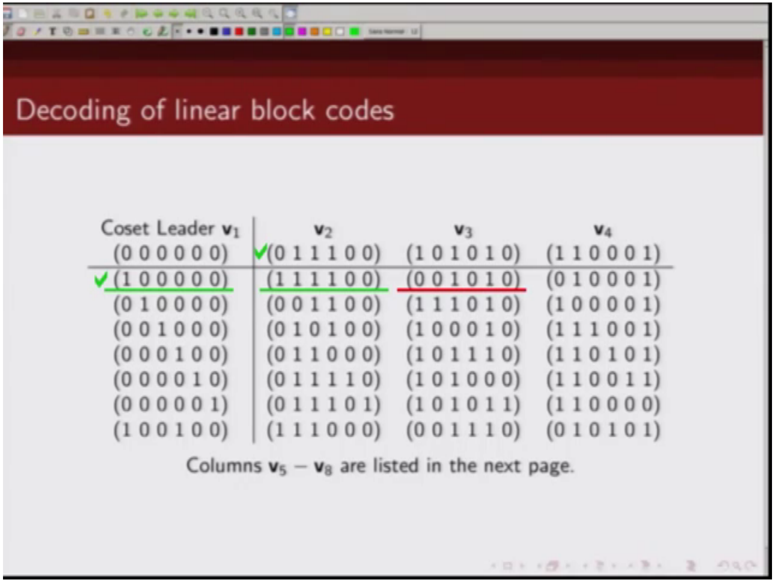
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	✓(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
✓(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

So I am going to add this to this. If I add it, what do I get? 1 plus 0 is 1, 0 plus 1 is 1, 0 plus 1 is 1, 0 plus 1 is 1, 0 plus 0 is 0, 0 plus 0 is 0. So this is what I get. How do I get this entry?

(Refer Slide Time 12:45)



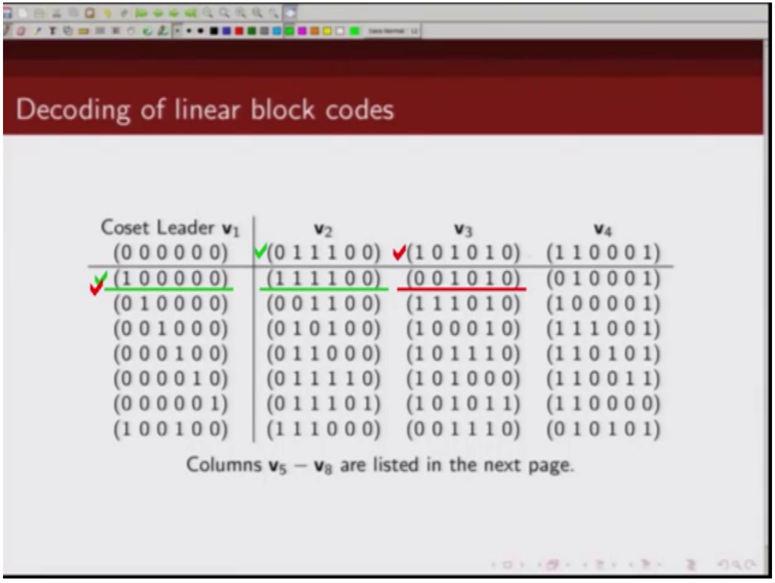
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	✓(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
✓(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

Again, I add this to this.

(Refer Slide Time 12:51)



Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	✓(0 1 1 1 0 0)	✓(1 0 1 0 1 0)	(1 1 0 0 0 1)
✓(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

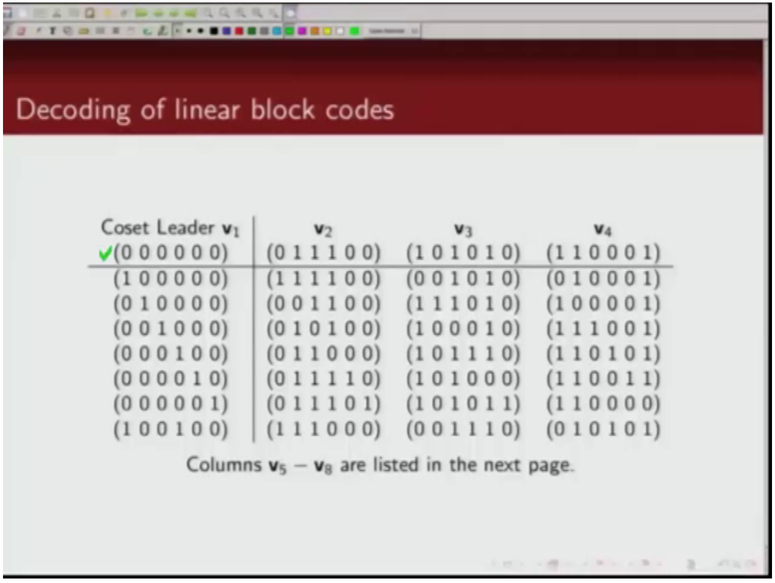
So 1 plus 1 is 0, 0 plus 0 is 0, 0 plus 1 is 1,0 plus 0 is 0, 0 plus 1 is 1, 0 plus 0 is 0. So this is how I populate the entries in this row. Next

(Refer Slide Time 13:11)

how do I pick this?

I will now have to look at

(Refer Slide Time 13:17)



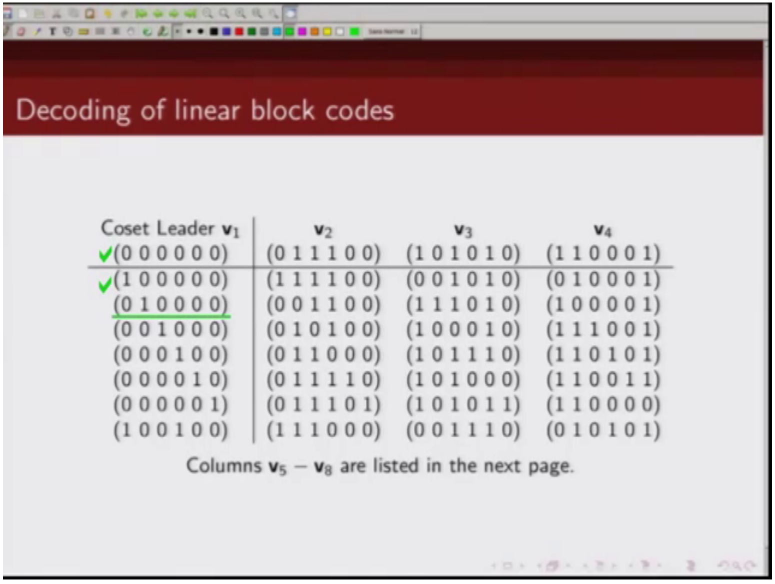
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
✓(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

the first two rows of this array and see, pick one n-tuple which has not happened before and this particular tuple you can see,

(Refer Slide Time 13:30)



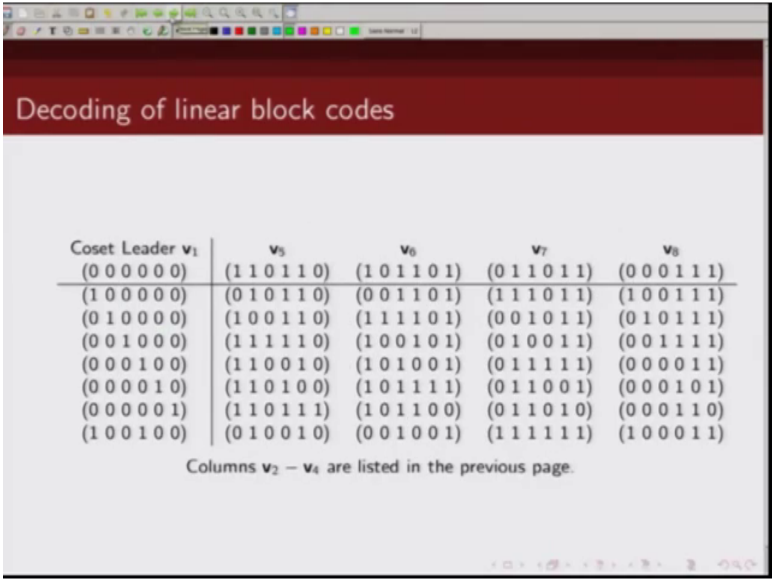
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
✓(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
✓(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

0 1 and all zeroes,

(Refer Slide Time 13:34)



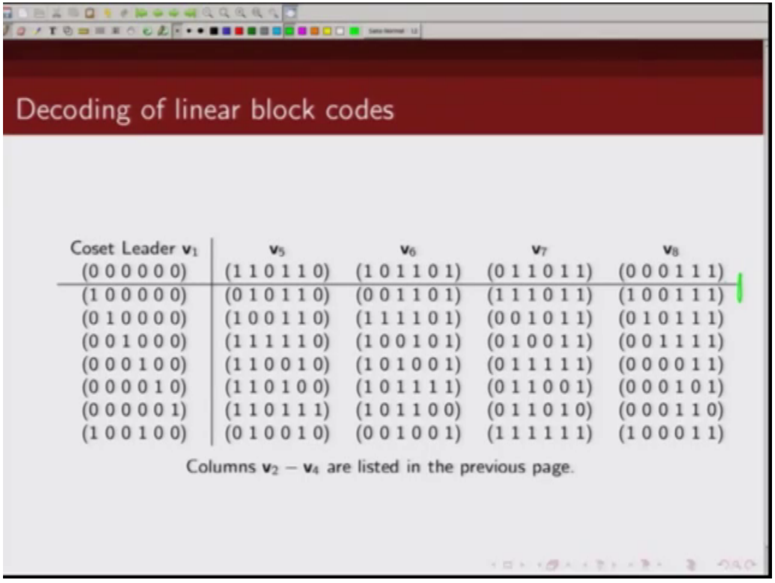
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

it has not appeared so far in the first two rows of the

(Refer Slide Time 13:40)



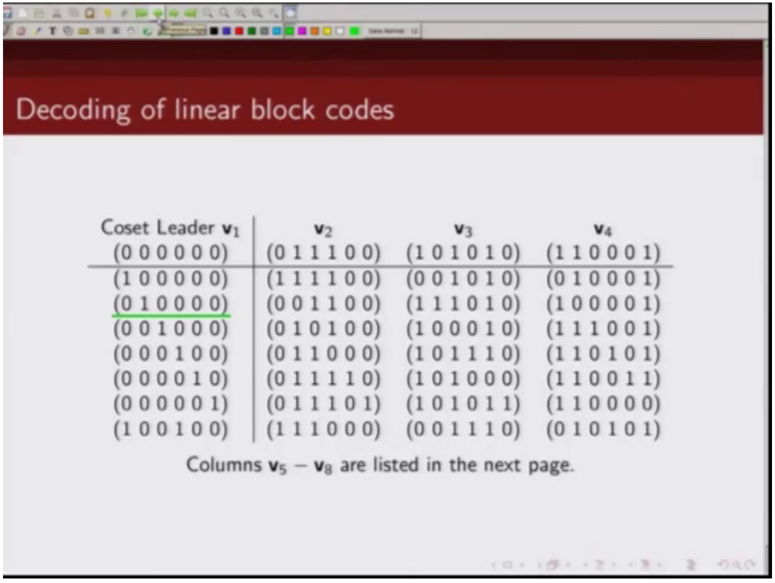
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

standard array. So I can then pick this as

(Refer Slide Time 13:45)



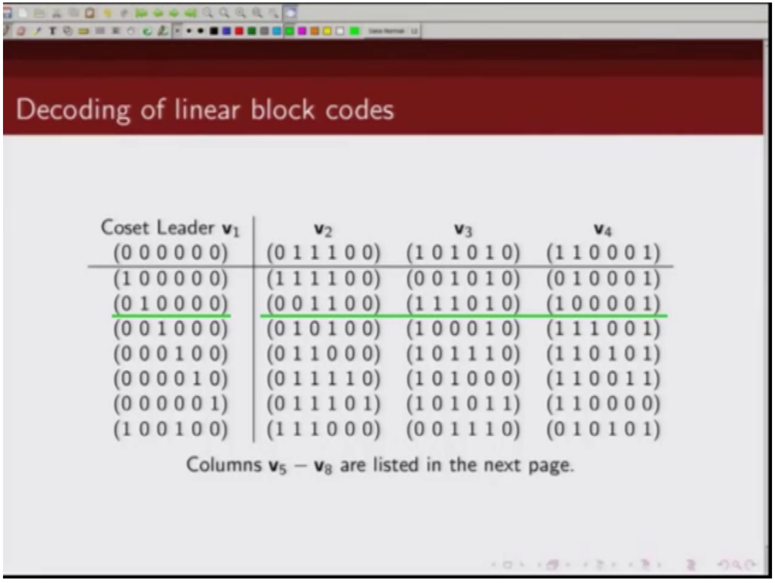
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

my n-tuple here. And then I fill up this whole entry. How?

(Refer Slide Time 13:51)



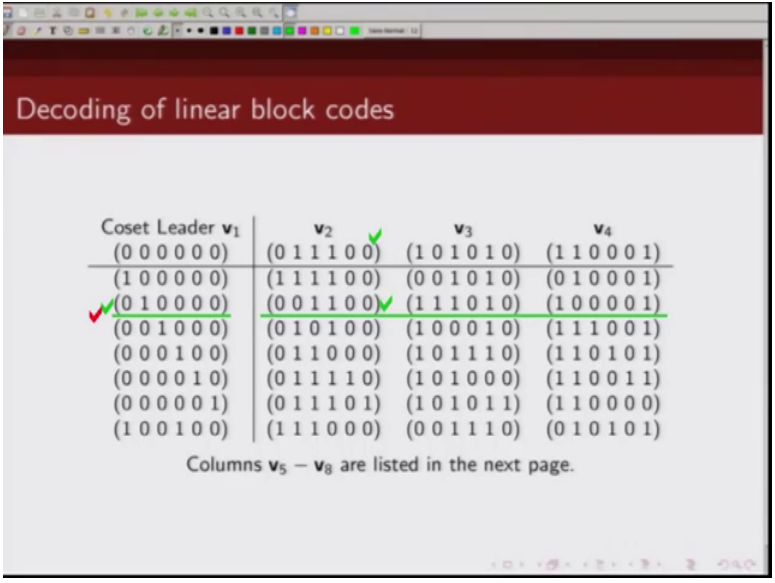
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

I add this vector to this v 2, put it here. Add this vector to this. Put it here. Add this vector to this,

(Refer Slide Time 13:56)



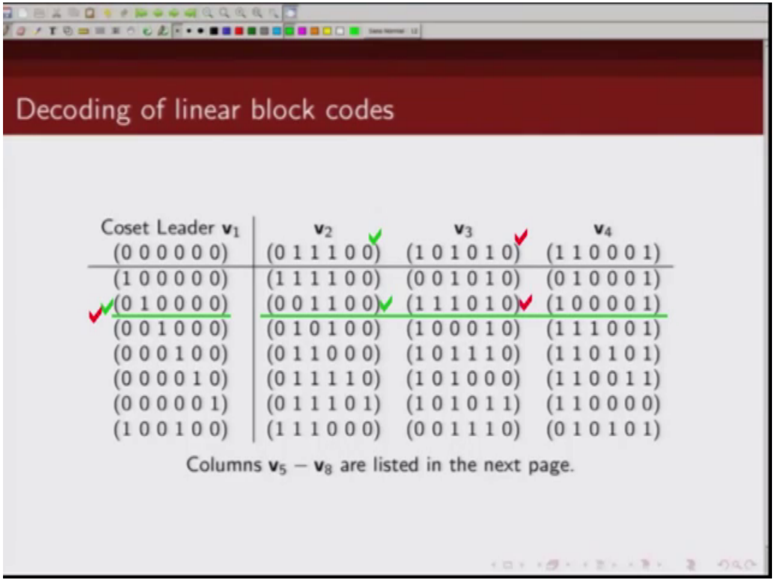
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0) ✓	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
✓✓ (0 1 0 0 0 0)	(0 0 1 1 0 0) ✓	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

put it here. Add this vector to this. Put it here. Add this vector to this.

(Refer Slide Time 13:58)

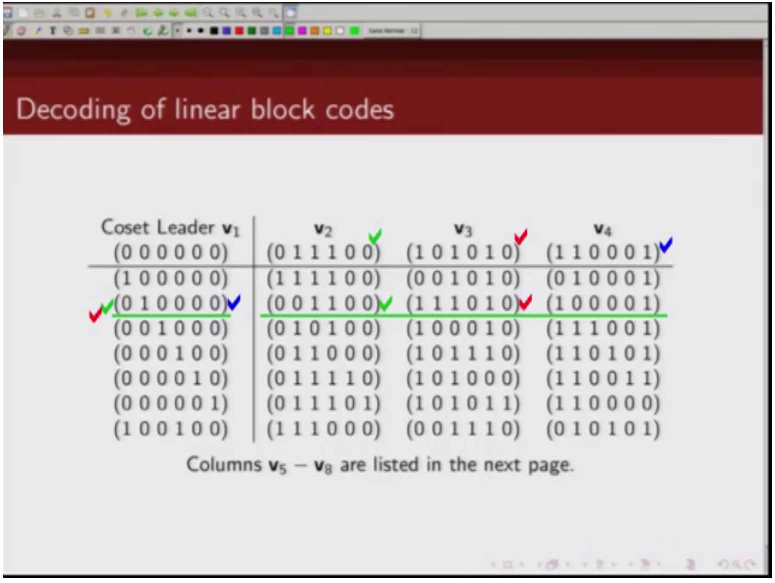


Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0) ✓	(1 0 1 0 1 0) ✓	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
✓✓ (0 1 0 0 0 0)	(0 0 1 1 0 0) ✓	(1 1 1 0 1 0) ✓	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

(Refer Slide Time 13:59)

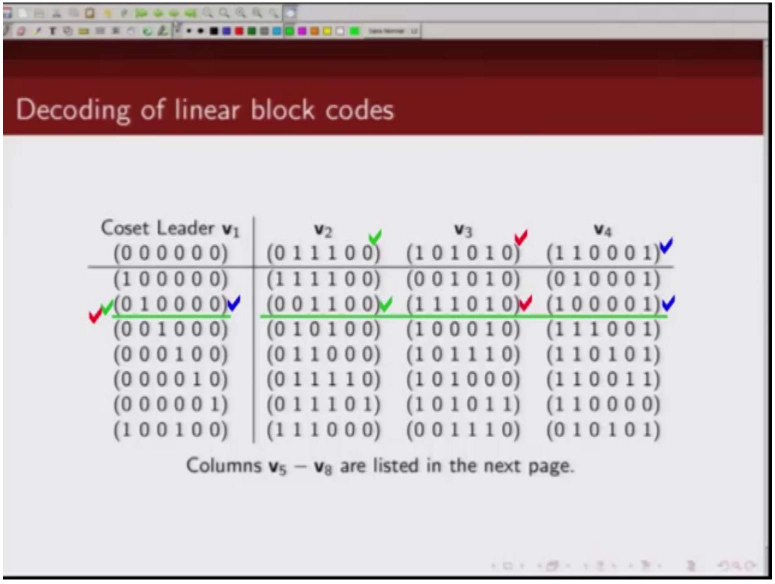


Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$ ✓	$\mathbf{v}_3$ ✓	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)✓
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
✓✓(0 1 0 0 0 0)✓	(0 0 1 1 0 0)✓	(1 1 1 0 1 0)✓	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

(Refer Slide Time 14:01)

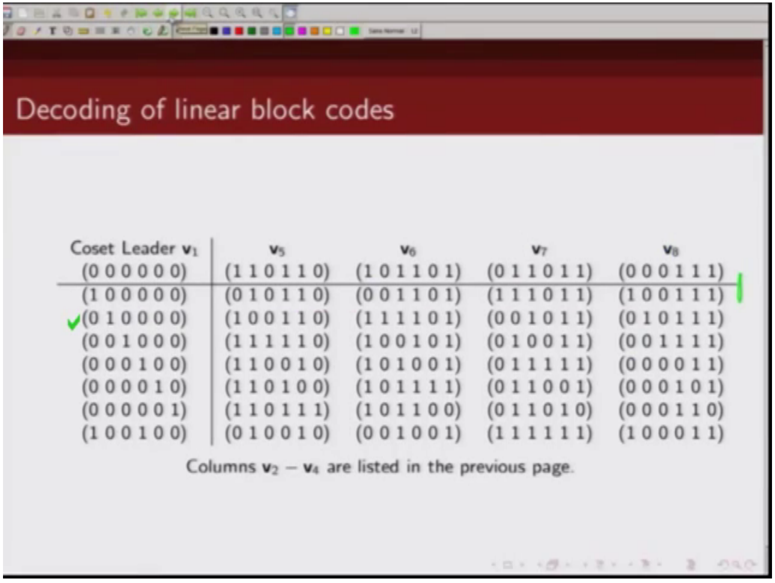


Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$ ✓	$\mathbf{v}_3$ ✓	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)✓
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
✓✓(0 1 0 0 0 0)✓	(0 0 1 1 0 0)✓	(1 1 1 0 1 0)✓	(1 0 0 0 0 1)✓
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

(Refer Slide Time 14:02)

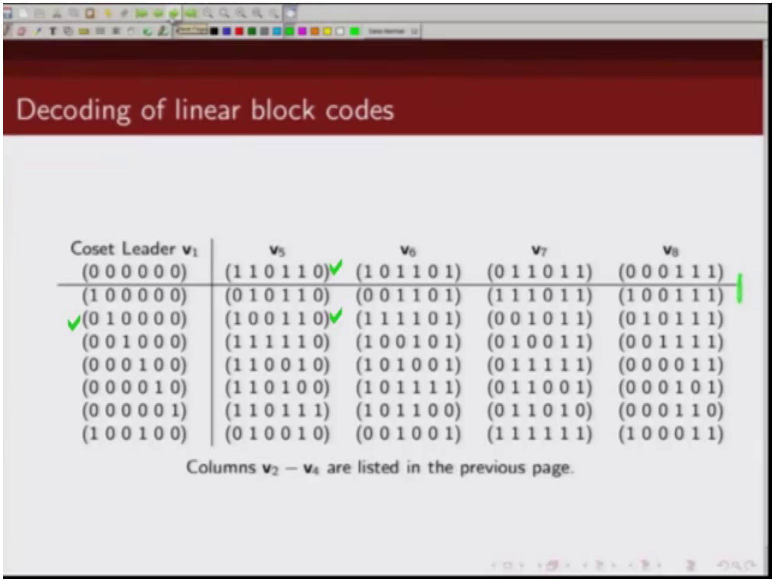


Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
✓(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

(Refer Slide Time 14:04)

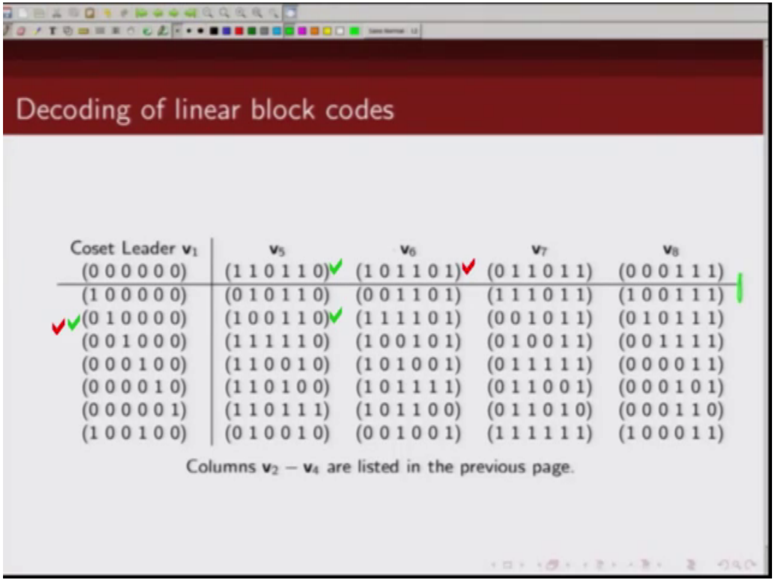


Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)✓	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
✓(0 1 0 0 0 0)	(1 0 0 1 1 0)✓	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

(Refer Slide Time 14:07)



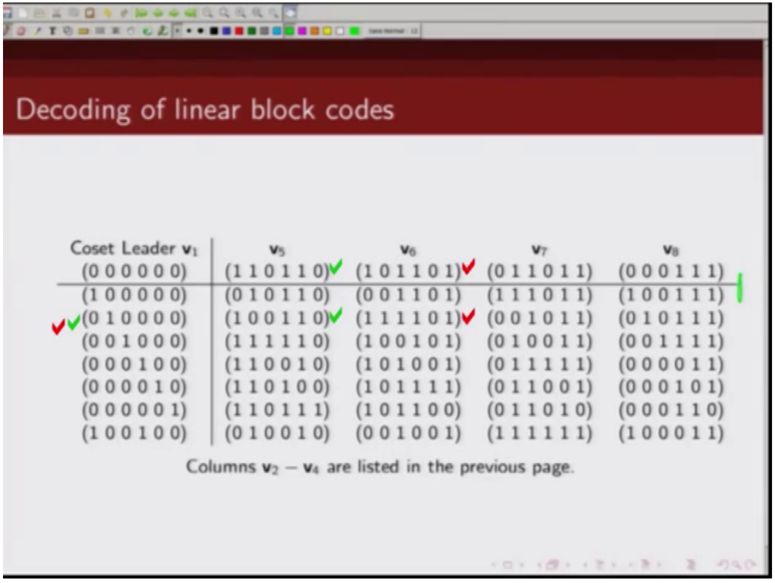
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)✓	(1 0 1 1 0 1)✓	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
✓✓(0 1 0 0 0 0)	(1 0 0 1 1 0)✓	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

You can just verify one such entry. So

(Refer Slide Time 14:08)



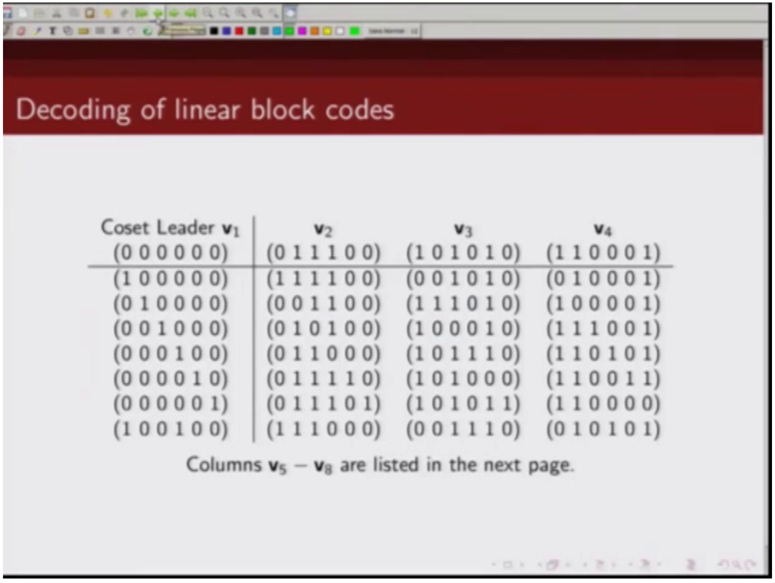
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)✓	(1 0 1 1 0 1)✓	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
✓✓(0 1 0 0 0 0)	(1 0 0 1 1 0)✓	(1 1 1 1 0 1)✓	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

0 plus 1 is 1. 1 plus 0 is 1, 0 plus 1 is 1, 0 plus 1 is 1, 0 plus 0 is 0 and 0 plus 1 is 1. So this is how you populate this entry. And we will keep on doing it

(Refer Slide Time 14:26)



Decoding of linear block codes

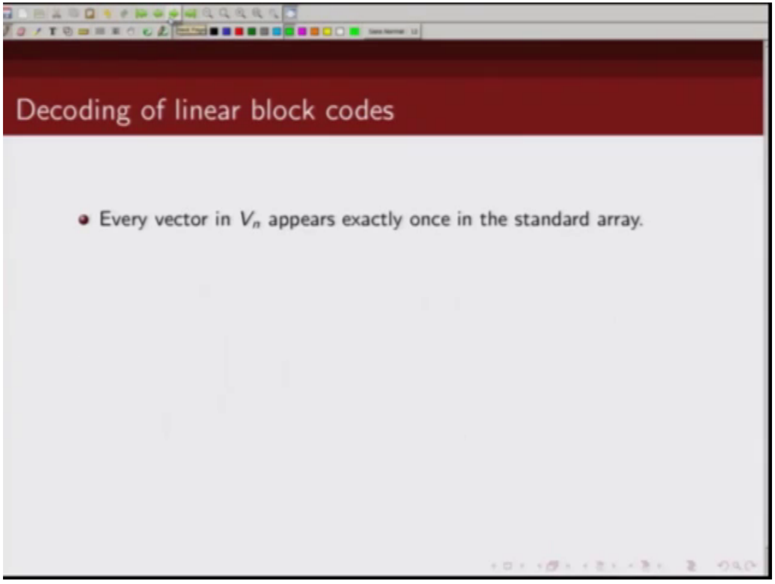
Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_4$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

Columns $\mathbf{v}_5 - \mathbf{v}_8$ are listed in the next page.

until we have written all those n-tuples here; so we have already written, this way we have written all the 2 n possibilities. We have written it in this array.

Now this array has some interesting properties; and we are going to talk about that which we will make use of while decoding our linear block code. Every vector in this

(Refer Slide Time 14:57)

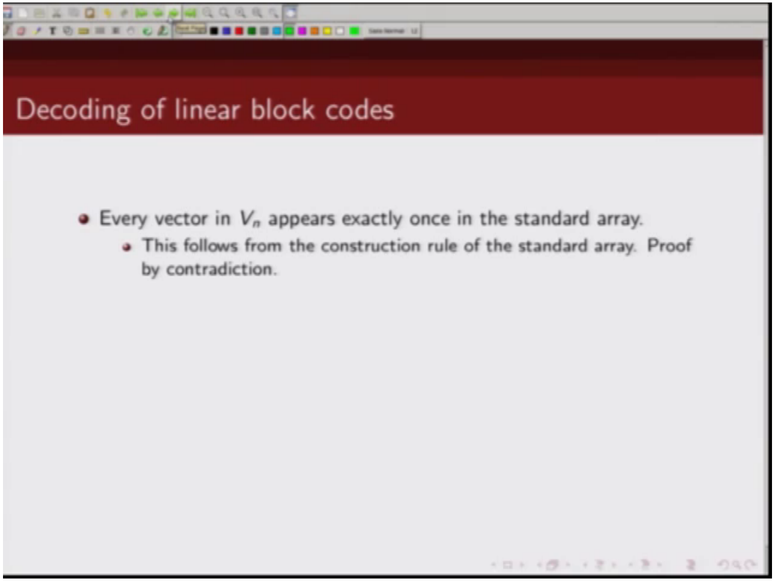


Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.

standard array appears exactly once. This is not very difficult to prove.

(Refer Slide Time 15:06)

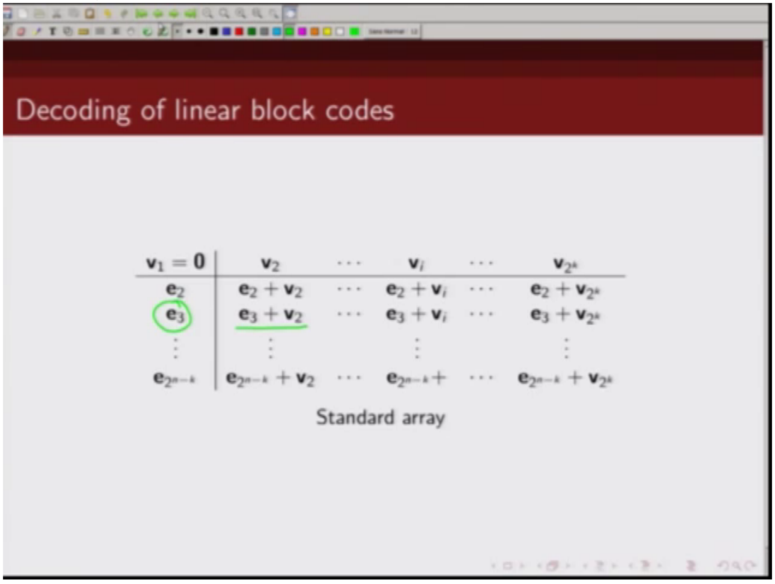


Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
  - This follows from the construction rule of the standard array. Proof by contradiction.

This follows from the way we are constructing our standard array, Ok and the proof is by contradiction. So I will just, it is a very small proof so I can just give you that. Let's say how does the proof by contradiction work?

(Refer Slide Time 15:24)



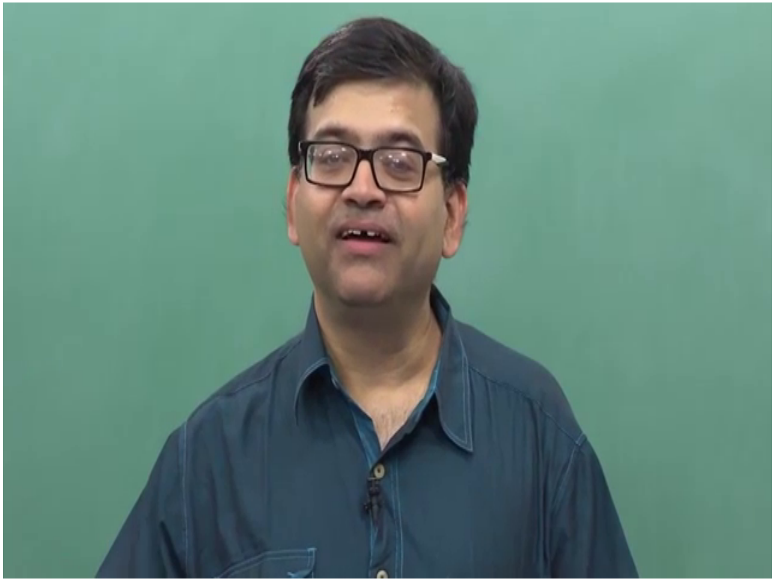
Decoding of linear block codes

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\underline{\mathbf{e}_3 + \mathbf{v}_2}$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} +$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

Standard array

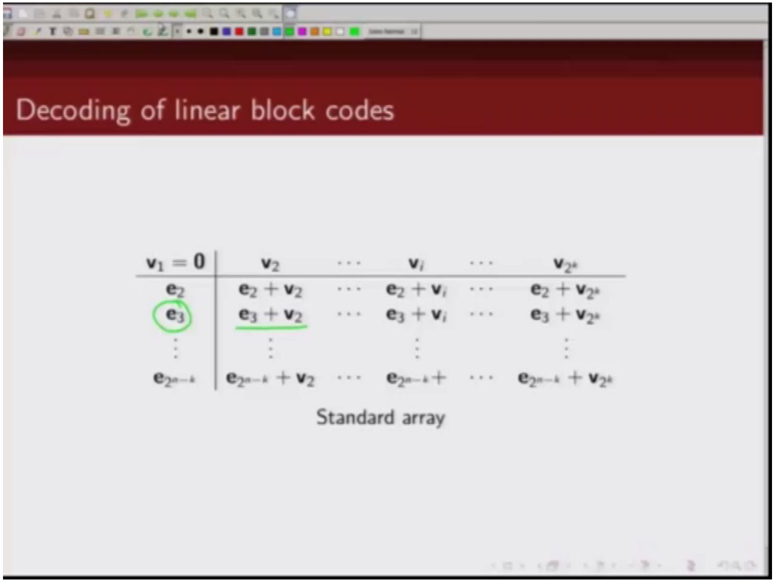
We will assume something and then we will show that

(Refer Slide Time 15:30)



our assumption is wrong. That's not possible, Ok. So we have to prove that every element here is

(Refer Slide Time 15:37)



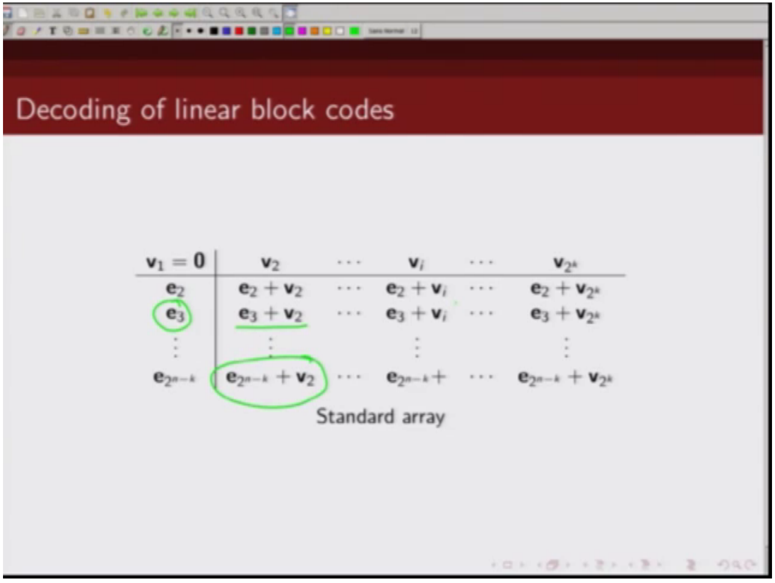
Decoding of linear block codes

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\underline{\mathbf{e}_3 + \mathbf{v}_2}$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} +$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

Standard array

basically unique. So let's say that's not true. Let's say 2 elements, let us just call it this element and this element.

(Refer Slide Time 15:45)



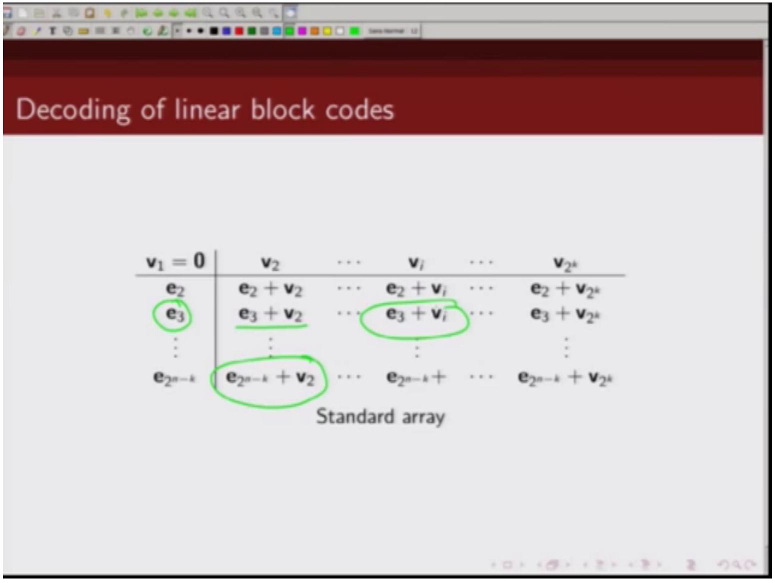
Decoding of linear block codes

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	$e_2 + v_2$	$\cdots$	$e_2 + v_i$	$\cdots$	$e_2 + v_{2^k}$
e_3	$e_3 + v_2$	$\cdots$	$e_3 + v_i$	$\cdots$	$e_3 + v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}} + v_{2^k}$

Standard array

Let's say these

(Refer Slide Time 15:46)



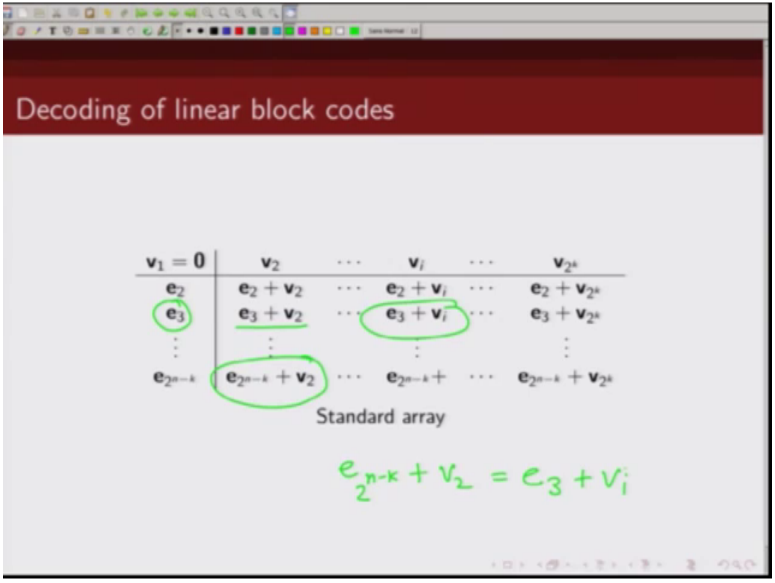
Decoding of linear block codes

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	$e_2 + v_2$	$\cdots$	$e_2 + v_i$	$\cdots$	$e_2 + v_{2^k}$
e_3	$e_3 + v_2$	$\cdots$	$e_3 + v_i$	$\cdots$	$e_3 + v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}} + v_{2^k}$

Standard array

2 elements are same, Ok. If these two elements are same, then we can write this as e 2 raised to power n minus k plus v 2 to be equal to e 3 plus v i, correct.

(Refer Slide Time 16:07)



Decoding of linear block codes

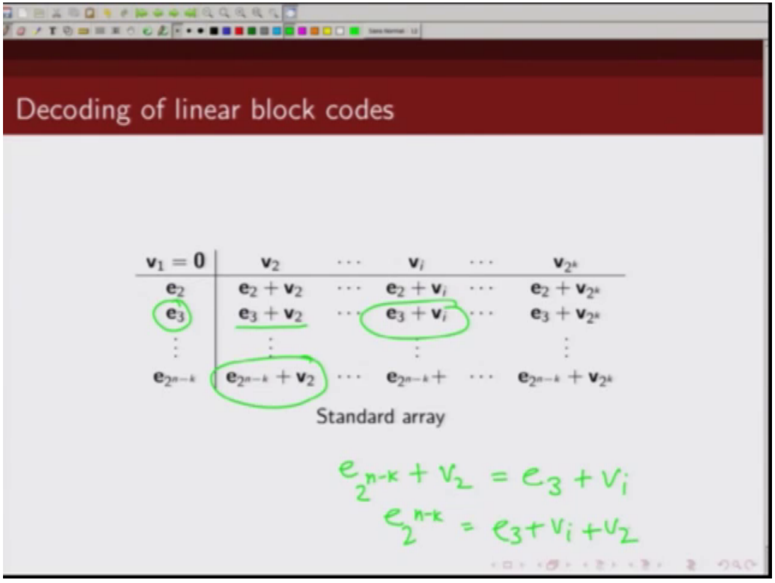
$v_1 = 0$	v_2	⋯	v_i	⋯	v_{2^k}
e_2	$e_2 + v_2$	⋯	$e_2 + v_i$	⋯	$e_2 + v_{2^k}$
e_3	$e_3 + v_2$	⋯	$e_3 + v_i$	⋯	$e_3 + v_{2^k}$
⋮	⋮		⋮		⋮
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	⋯	$e_{2^{n-k}} +$	⋯	$e_{2^{n-k}} + v_{2^k}$

Standard array

$$e_{2^{n-k}} + V_2 = e_3 + V_i$$

Now we can write this e 2 n minus k as e 3 plus v i plus v 2,

(Refer Slide Time 16:18)



Decoding of linear block codes

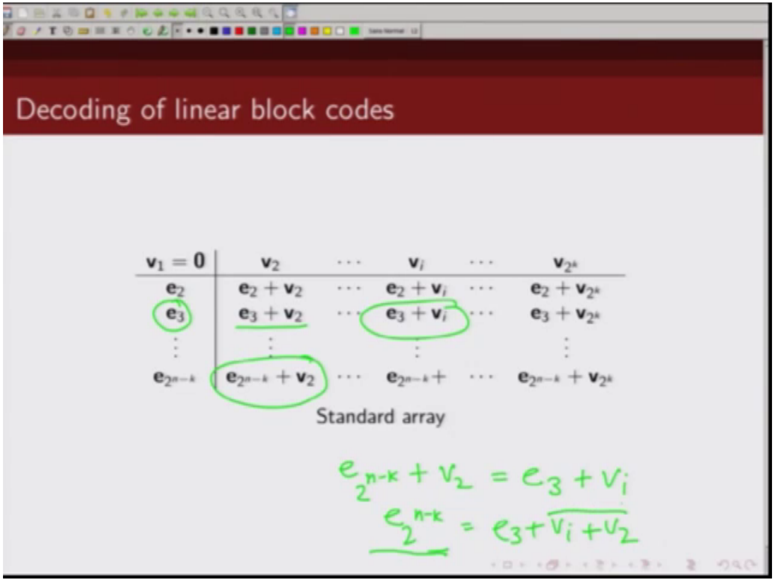
$v_1 = 0$	v_2	⋯	v_i	⋯	v_{2^k}
e_2	$e_2 + v_2$	⋯	$e_2 + v_i$	⋯	$e_2 + v_{2^k}$
e_3	$e_3 + v_2$	⋯	$e_3 + v_i$	⋯	$e_3 + v_{2^k}$
⋮	⋮		⋮		⋮
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	⋯	$e_{2^{n-k}} +$	⋯	$e_{2^{n-k}} + v_{2^k}$

Standard array

$$e_{2^{n-k}} + V_2 = e_3 + V_i$$
$$e_{2^{n-k}} = e_3 + V_i + V_2$$

why? Because these are all binary words so basically when we add 1 plus 1, that's basically 0. So we add v 2 to both the sides, so v 2 plus v 2 will be 0. So we can write this error pattern as e 3 plus v i plus v 2. And what is v i plus v 2? v i plus v 2 is another codeword. Why, because that's the property of the linear

(Refer Slide Time 16:44)



Decoding of linear block codes

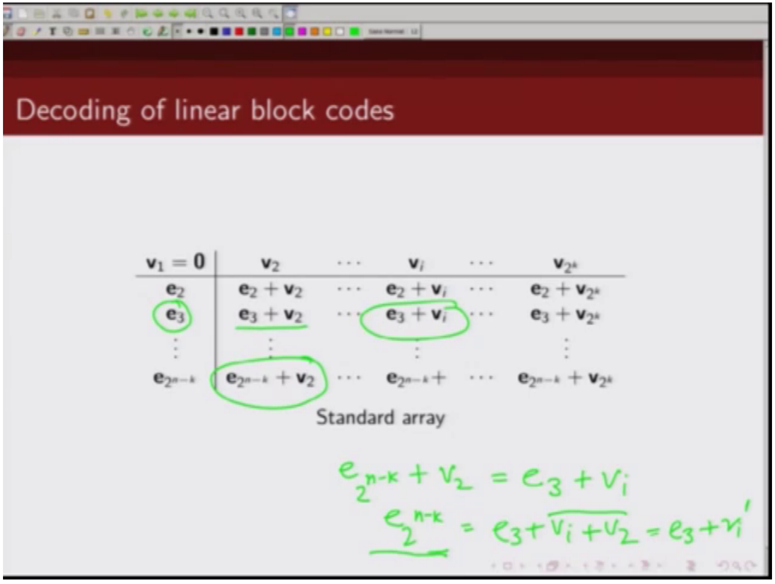
$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	$e_2 + v_2$	$\cdots$	$e_2 + v_i$	$\cdots$	$e_2 + v_{2^k}$
e_3	$e_3 + v_2$	$\cdots$	$e_3 + v_i$	$\cdots$	$e_3 + v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}} + v_{2^k}$

Standard array

$$e_{2^{n-k}} + v_2 = e_3 + v_i$$
$$\underline{e_{2^{n-k}}} = e_3 + \overline{v_i + v_2}$$

codeword, linear block code. So this will be e 3 plus some other codeword, let's call it v i, v i hat.

(Refer Slide Time 16:54)



Decoding of linear block codes

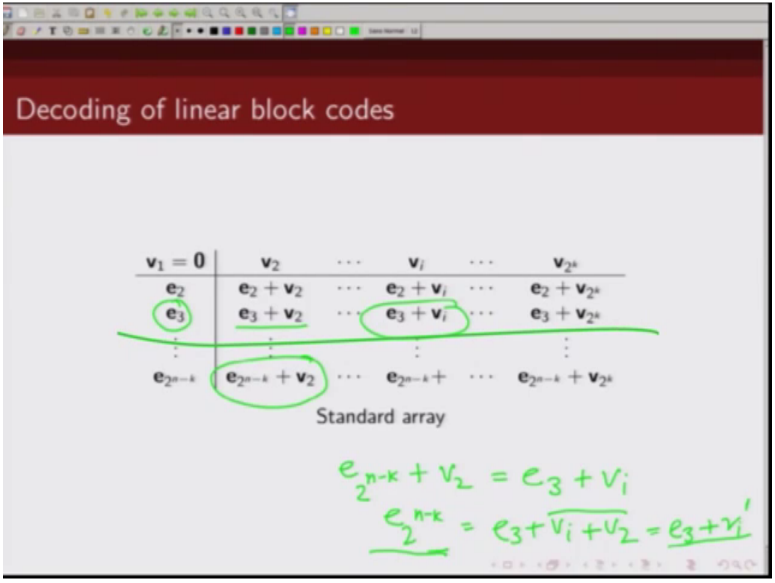
$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	$e_2 + v_2$	$\cdots$	$e_2 + v_i$	$\cdots$	$e_2 + v_{2^k}$
e_3	$e_3 + v_2$	$\cdots$	$e_3 + v_i$	$\cdots$	$e_3 + v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}} + v_{2^k}$

Standard array

$$e_{2^{n-k}} + v_2 = e_3 + v_i$$
$$\underline{e_{2^{n-k}}} = e_3 + \overline{v_i + v_2} = e_3 + v_i'$$

So that error pattern e 2 n minus k is e 3 plus v hat. Now e 3 plus v i hat, this should be in the row containing e 3 because these are, how do we find the entries in the row containing e 3? We add all codewords to e 3 and that's what these entries are. So e 3 plus v i hat should have been some entry here.

(Refer Slide Time 17:36)



Decoding of linear block codes

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	$e_2 + v_2$	$\cdots$	$e_2 + v_i$	$\cdots$	$e_2 + v_{2^k}$
e_3	$e_3 + v_2$	$\cdots$	$e_3 + v_i$	$\cdots$	$e_3 + v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}} + v_{2^k}$

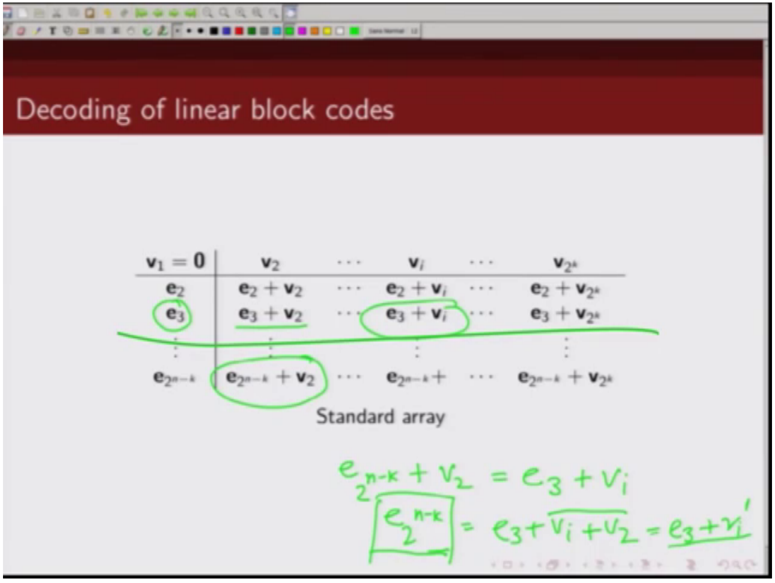
Standard array

$$e_{2^{n-k}} + v_2 = e_3 + v_i$$

$$e_{2^{n-k}} = e_3 + \overline{v_i + v_2} = e_3 + v_i'$$

What does that mean? That means we made a mistake in selecting this error pattern.

(Refer Slide Time 17:47)



Decoding of linear block codes

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	$e_2 + v_2$	$\cdots$	$e_2 + v_i$	$\cdots$	$e_2 + v_{2^k}$
e_3	$e_3 + v_2$	$\cdots$	$e_3 + v_i$	$\cdots$	$e_3 + v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}} + v_{2^k}$

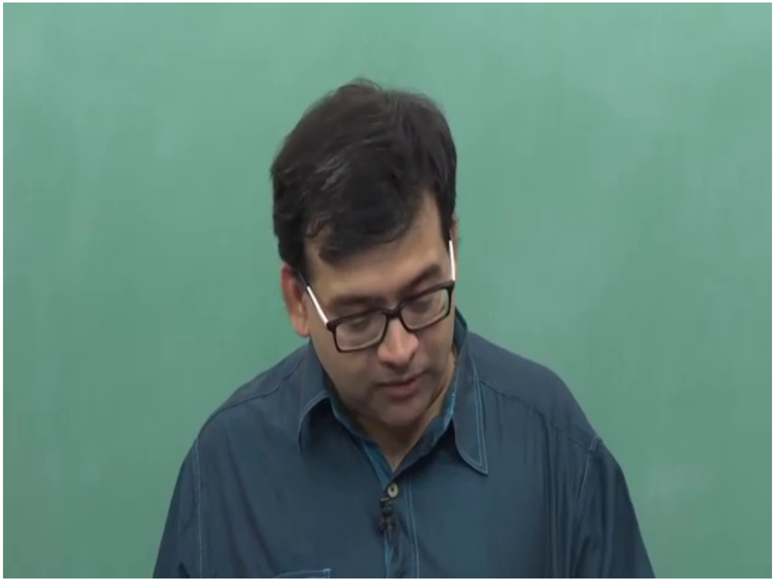
Standard array

$$e_{2^{n-k}} + v_2 = e_3 + v_i$$

$$\boxed{e_{2^{n-k}}} = e_3 + \overline{v_i + v_2} = e_3 + v_i'$$

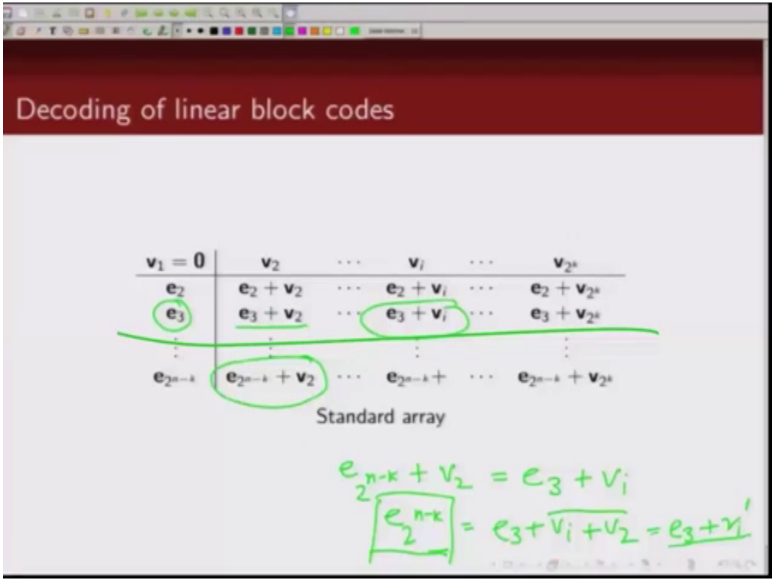
Note what did we say? We are choosing these error patterns in such a way that in the previous rows, this error pattern has not appeared. But here we have shown, if these two elements would have been same, if these two vectors in this standard array would have been same then this is the condition. Which would mean this error pattern is already there in the row containing e 3. So that means

(Refer Slide Time 18:18)



we cannot choose that as our error pattern here because

(Refer Slide Time 18:24)



Decoding of linear block codes

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\mathbf{e}_3 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} +$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

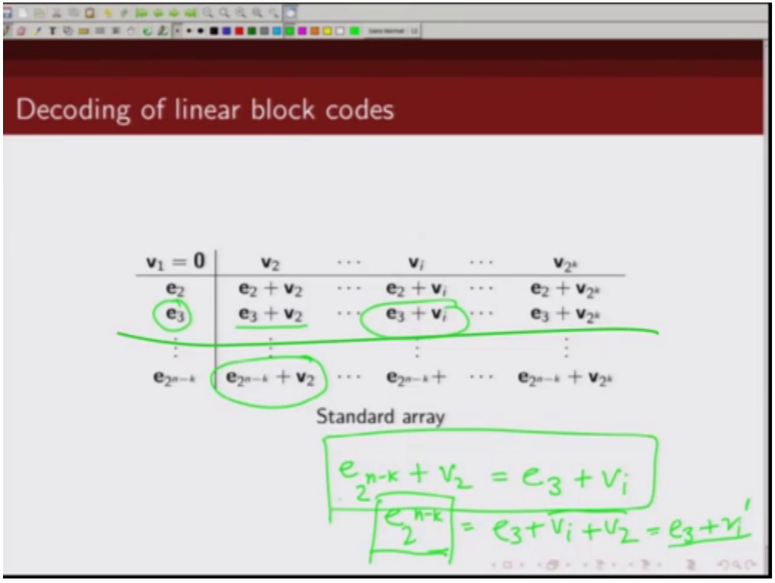
Standard array

$$e_{2^{n-k}} + v_2 = e_3 + v_i$$

$$\boxed{e_{2^{n-k}}} = e_3 + \overline{v_i + v_2} = \underline{e_3 + v_i'}$$

we can only choose an error pattern which has not appeared beforehand. So in other words, we are contradicting ourselves. On one hand we are saying we are picking up these error patterns in such a way that they have not appeared in the previous rows but if we are saying these two elements in this array are same, then this is not possible. So hence by contradiction basically we prove that this is not possible. We could not choose e 2 n minus k as this because this has already appeared. Hence this condition that these 2 elements are same

(Refer Slide Time 19:06)



Decoding of linear block codes

$$
\begin{array}{c|ccccccc}
\mathbf{v}_1 = \mathbf{0} & \mathbf{v}_2 & \cdots & \mathbf{v}_i & \cdots & \mathbf{v}_{2^k} \\
\hline
\mathbf{e}_2 & \mathbf{e}_2 + \mathbf{v}_2 & \cdots & \mathbf{e}_2 + \mathbf{v}_i & \cdots & \mathbf{e}_2 + \mathbf{v}_{2^k} \\
\mathbf{e}_3 & \underline{\mathbf{e}_3 + \mathbf{v}_2} & \cdots & \mathbf{e}_3 + \mathbf{v}_i & \cdots & \mathbf{e}_3 + \mathbf{v}_{2^k} \\
\vdots & \vdots & & \vdots & & \vdots \\
\mathbf{e}_{2^{n-k}} & \mathbf{e}_{2^{n-k}} + \mathbf{v}_2 & \cdots & \mathbf{e}_{2^{n-k}} + & \cdots & \mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}
\end{array}
$$

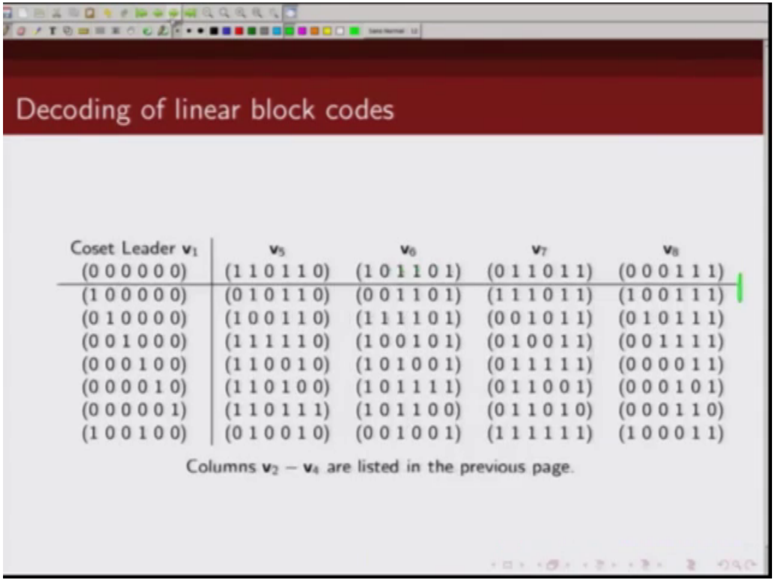
Standard array

$$\boxed{e_{2^{n-k}} + v_2 = e_3 + v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3 + v_i + v_2 = \underline{e_3 + v_i'}$$

is incorrect, Ok. So all the elements of the standard array are

(Refer Slide Time 19:14)



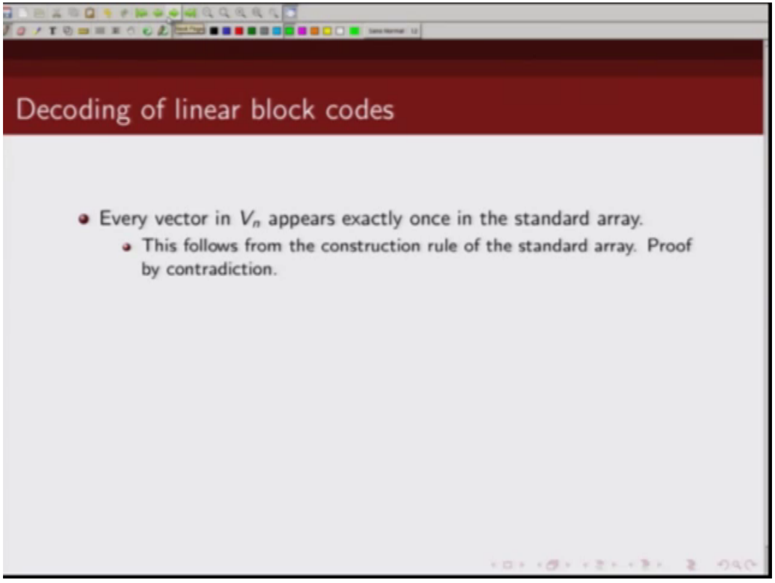
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

distinct and they appear exactly once.

(Refer Slide Time 19:19)

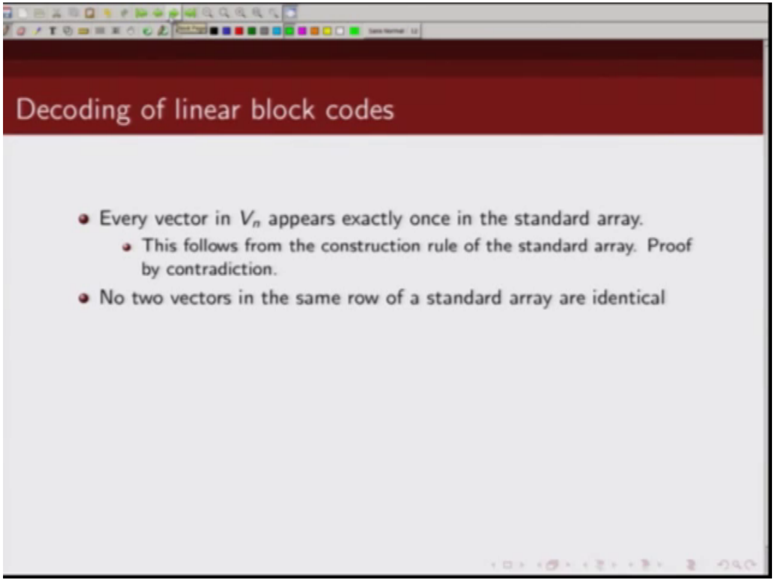


Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
    - This follows from the construction rule of the standard array. Proof by contradiction.

No two

(Refer Slide Time 19:22)

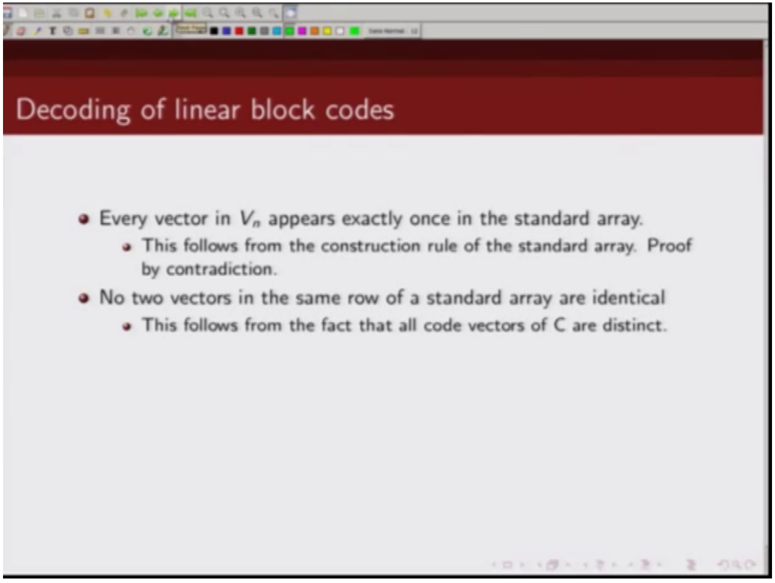


Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
    - This follows from the construction rule of the standard array. Proof by contradiction.
- No two vectors in the same row of a standard array are identical

vectors in the same row are identical. Again this is very easy

(Refer Slide Time 19:29)

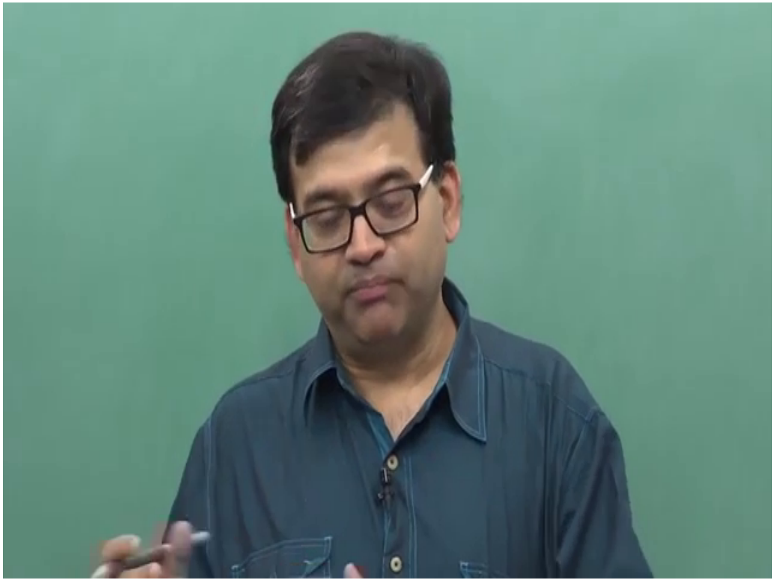


Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
  - This follows from the construction rule of the standard array. Proof by contradiction.
- No two vectors in the same row of a standard array are identical
  - This follows from the fact that all code vectors of C are distinct.

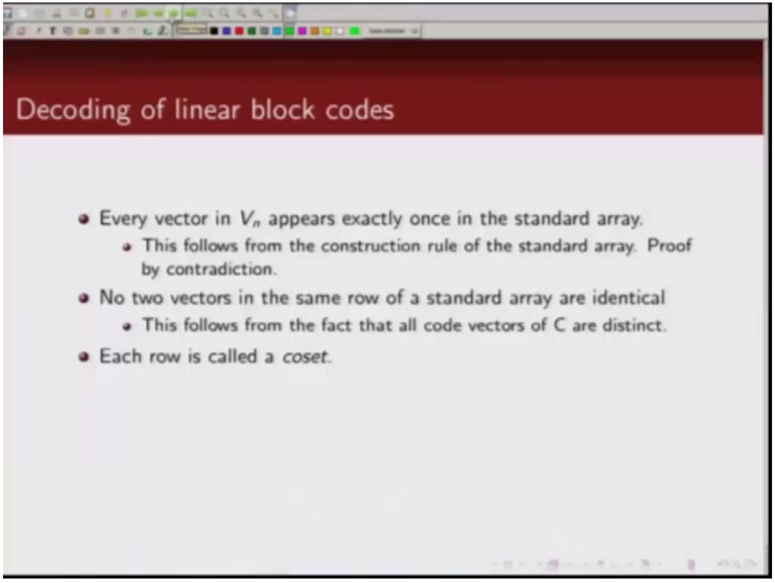
to prove because all codewords are distinct. So

(Refer Slide Time 19:34)



elements in one row will all be distinct.

(Refer Slide Time 19:41)

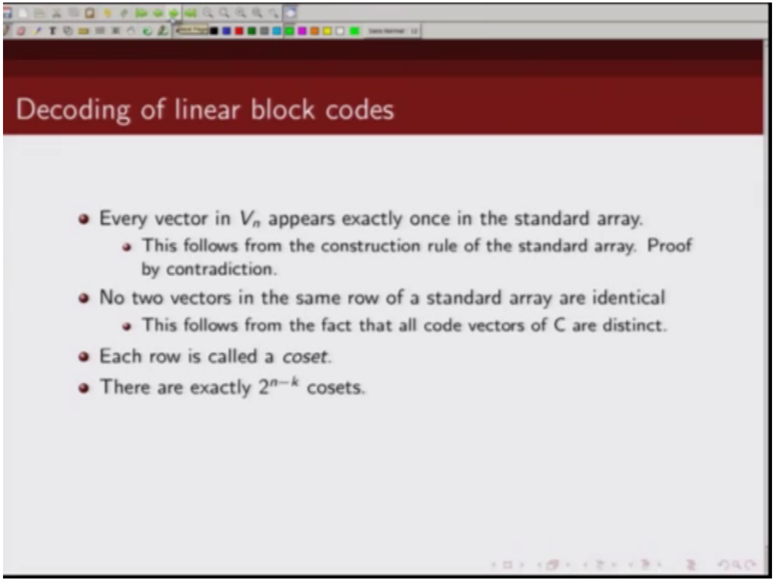


Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
  - This follows from the construction rule of the standard array. Proof by contradiction.
- No two vectors in the same row of a standard array are identical
  - This follows from the fact that all code vectors of C are distinct.
- Each row is called a *coset*.

We call each row of the standard array as coset and the leftmost entry

(Refer Slide Time 19:47)

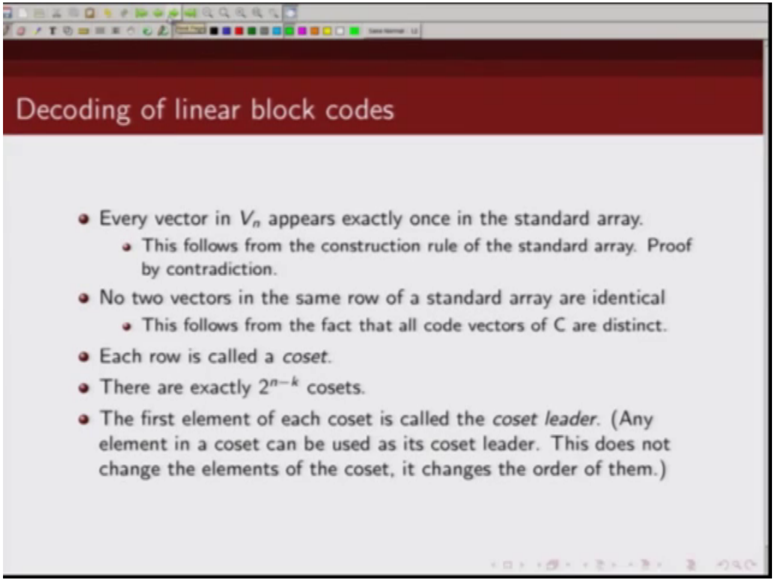


Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
  - This follows from the construction rule of the standard array. Proof by contradiction.
- No two vectors in the same row of a standard array are identical
  - This follows from the fact that all code vectors of C are distinct.
- Each row is called a *coset*.
- There are exactly 2^{n-k} cosets.

of each coset or row is known as coset leader. And we have total 2 raised to power n minus k such cosets.

(Refer Slide Time 19:59)

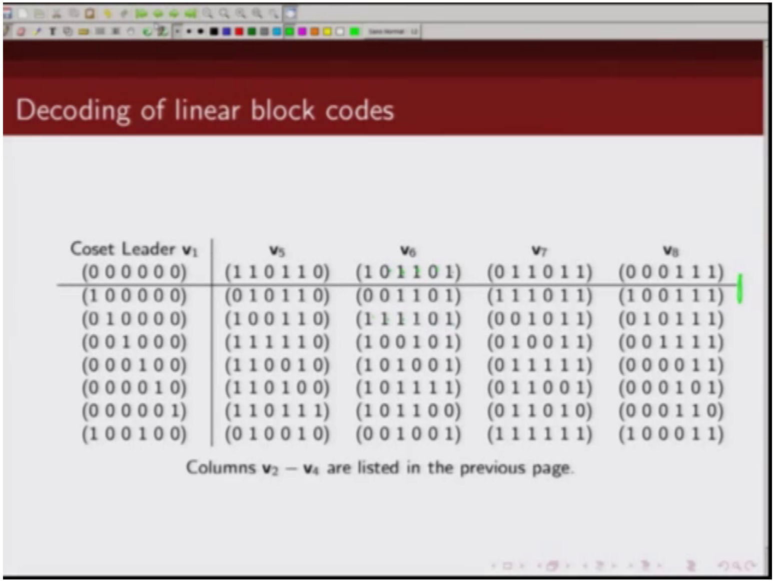


Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
  - This follows from the construction rule of the standard array. Proof by contradiction.
- No two vectors in the same row of a standard array are identical
  - This follows from the fact that all code vectors of C are distinct.
- Each row is called a *coset*.
- There are exactly 2^{n-k} cosets.
- The first element of each coset is called the *coset leader*. (Any element in a coset can be used as its coset leader. This does not change the elements of the coset, it changes the order of them.)

Now question is can we make any other element as our coset leader? So if you just go back here. Let's say

(Refer Slide Time 20:17)



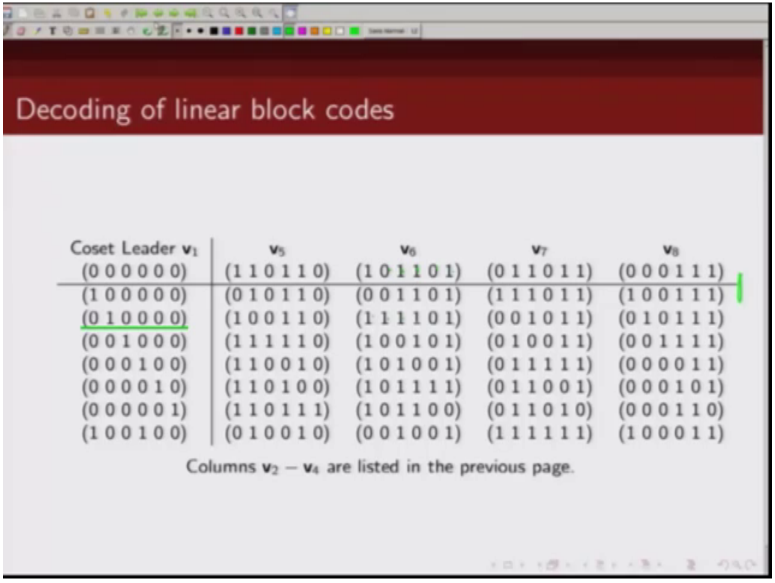
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

look at this row. Coset leader was this, right, the leftmost entry in the standard array. Now what happens if we instead of choosing this as a coset leader,

(Refer Slide Time 20:28)



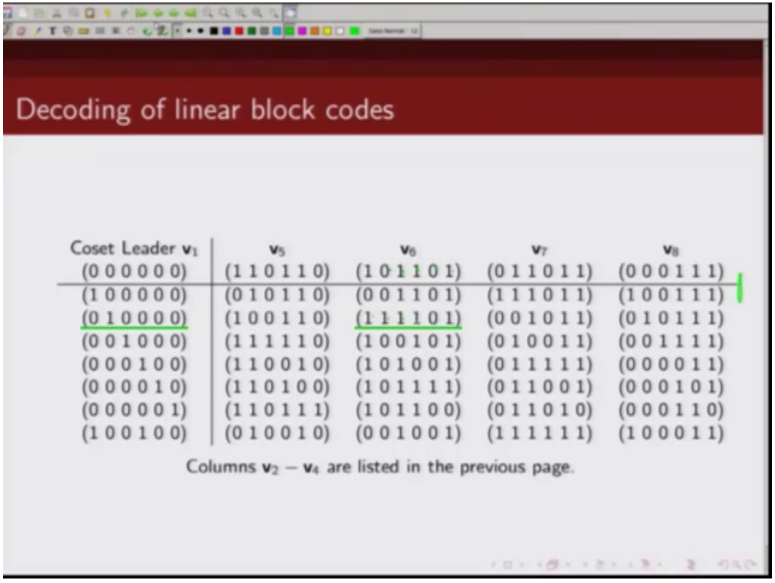
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

if we had chosen say this as coset leader?

(Refer Slide Time 20:32)



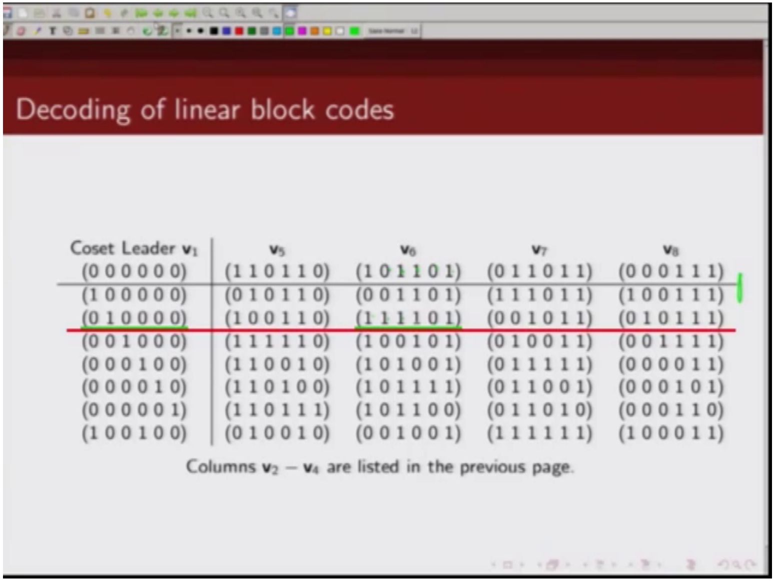
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

Would it had changed our elements of this

(Refer Slide Time 20:39)



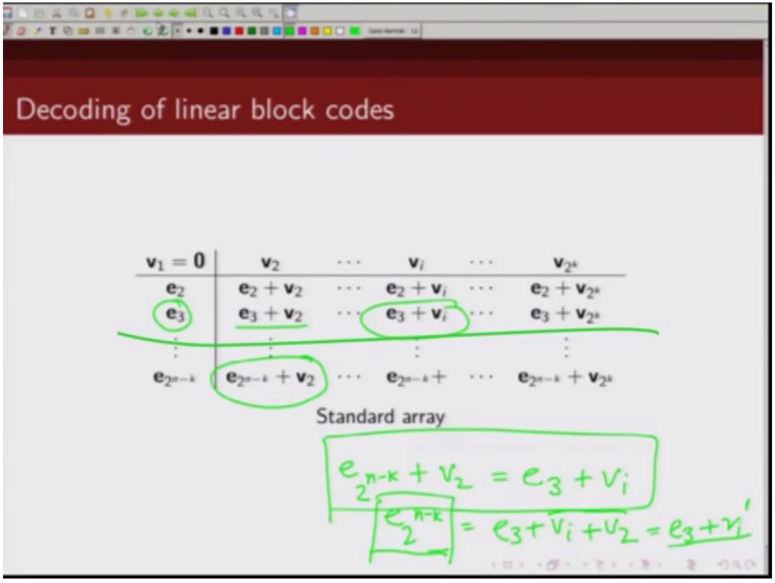
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_5$	$\mathbf{v}_6$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

Columns $\mathbf{v}_2 - \mathbf{v}_4$ are listed in the previous page.

array? No. Why? Because if you go back and see the entries

(Refer Slide Time 20:46)



Decoding of linear block codes

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\mathbf{e}_3 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} +$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

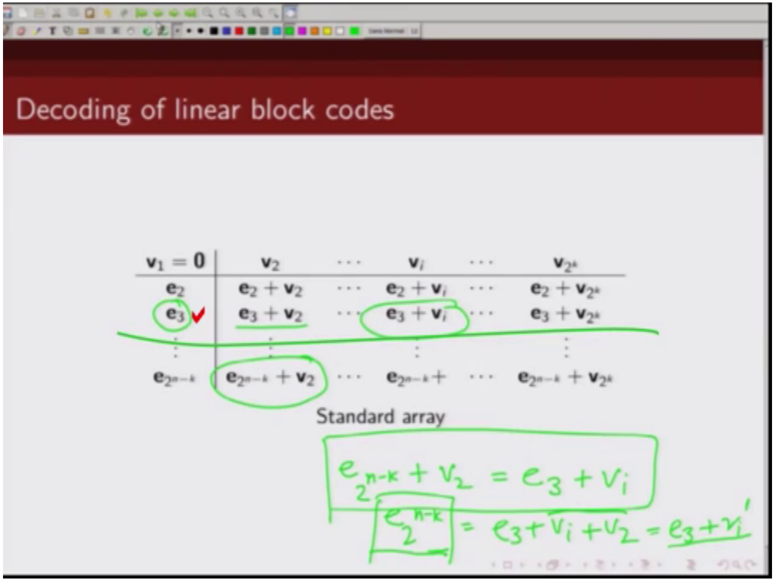
Standard array

$$\boxed{e_{2^{n-k}} + v_2 = e_3 + v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3 + v_i + v_2 = \underline{e_3 + v_i'}$$

of a particular row, here the coset leader of e 3, if instead of e 3 we would have used e 3 plus v 2, what would have happened? This would have been e 3 plus v 2;

(Refer Slide Time 20:58)



Decoding of linear block codes

$v_1 = 0$	v_2	⋯	v_i	⋯	v_{2^k}
e_2	$e_2 + v_2$	⋯	$e_2 + v_i$	⋯	$e_2 + v_{2^k}$
e_3 ✓	$e_3 + v_2$	⋯	$e_3 + v_i$	⋯	$e_3 + v_{2^k}$
⋮	⋮		⋮		⋮
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	⋯	$e_{2^{n-k}} +$	⋯	$e_{2^{n-k}} + v_{2^k}$

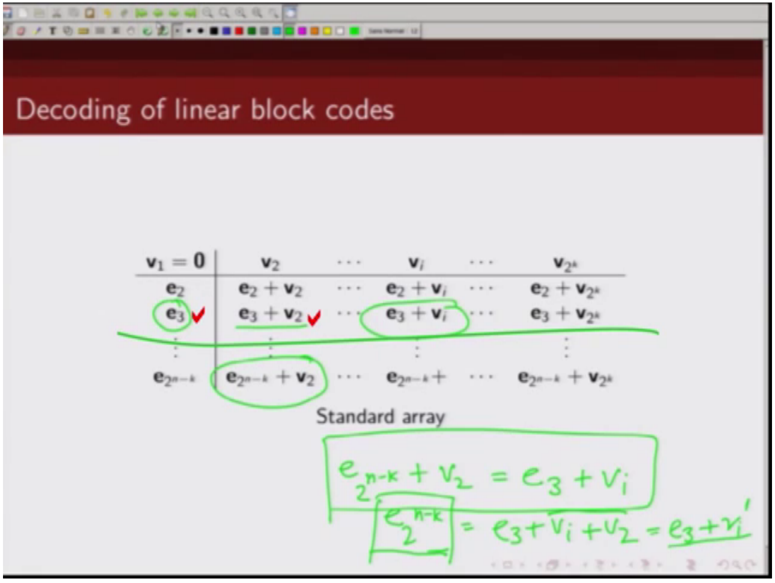
Standard array

$$\boxed{e_{2^{n-k}} + v_2 = e_3 + v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3 + v_i + v_2 = \underline{e_3 + v_i'}$$

this would have e 3 plus v 2 plus v 2.

(Refer Slide Time 21:01)



Decoding of linear block codes

$v_1 = 0$	v_2	⋯	v_i	⋯	v_{2^k}
e_2	$e_2 + v_2$	⋯	$e_2 + v_i$	⋯	$e_2 + v_{2^k}$
e_3 ✓	$e_3 + v_2$ ✓	⋯	$e_3 + v_i$	⋯	$e_3 + v_{2^k}$
⋮	⋮		⋮		⋮
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	⋯	$e_{2^{n-k}} +$	⋯	$e_{2^{n-k}} + v_{2^k}$

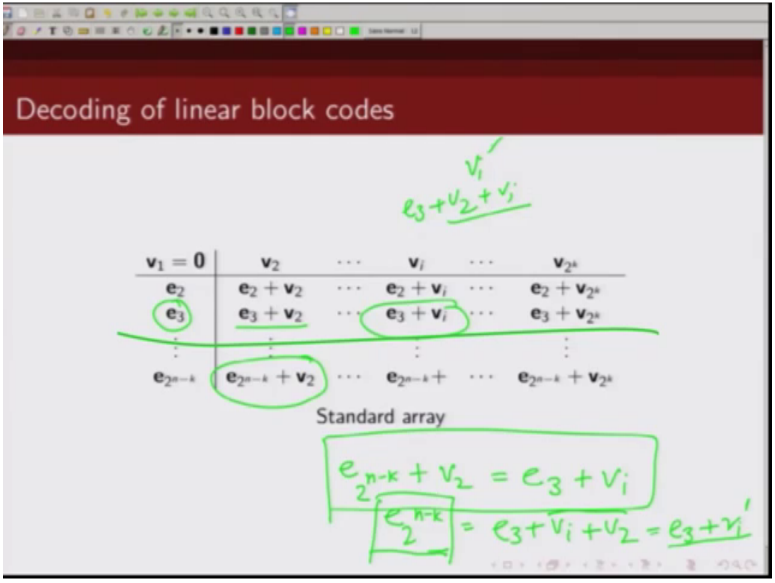
Standard array

$$\boxed{e_{2^{n-k}} + v_2 = e_3 + v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3 + v_i + v_2 = \underline{e_3 + v_i'}$$

So this would be e 3. This would be e 3 plus v 2 plus v i. And v 2 plus v i would be another codeword v i hat. So this would have been some other codeword. So the elements

(Refer Slide Time 21:15)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_2+\mathbf{v}_i$	$\cdots$	$\mathbf{e}_2+\mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\mathbf{e}_3+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_3+\mathbf{v}_i$	$\cdots$	$\mathbf{e}_3+\mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}}+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}}+$	$\cdots$	$\mathbf{e}_{2^{n-k}}+\mathbf{v}_{2^k}$

Standard array

$$\boxed{e_{2^{n-k}} + v_2 = e_3 + v_i}$$

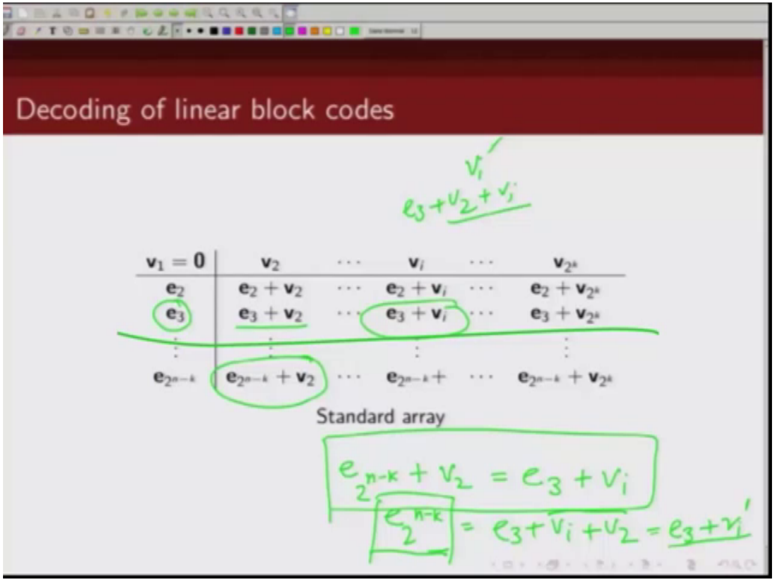
$$\boxed{e_{2^{n-k}}} = e_3 + v_i + v_2 = e_3 + v_i'$$

in each row would have remained the same, only they would have just got reordered,

(Refer Slide Time 21:22)

Ok. So if we pick any other

(Refer Slide Time 21:25)



Decoding of linear block codes

v_i'

$e_3 + v_2 + v_i$

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\mathbf{e}_3 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} +$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

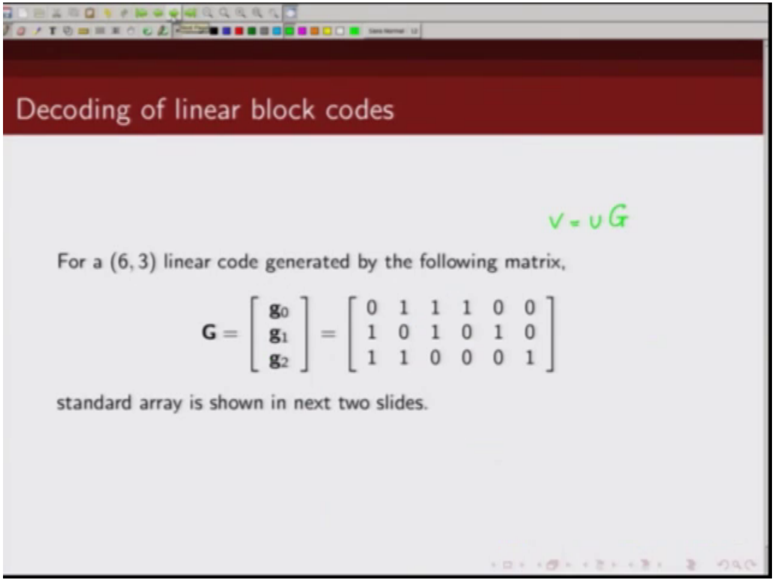
Standard array

$$\boxed{e_{2^{n-k}} + v_2 = e_3 + v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3 + v_i + v_2 = e_3 + v_i'$$

element in the row as coset leader, it does not change the elements in the coset or in a row.

(Refer Slide Time 21:36)



Decoding of linear block codes

$v = uG$

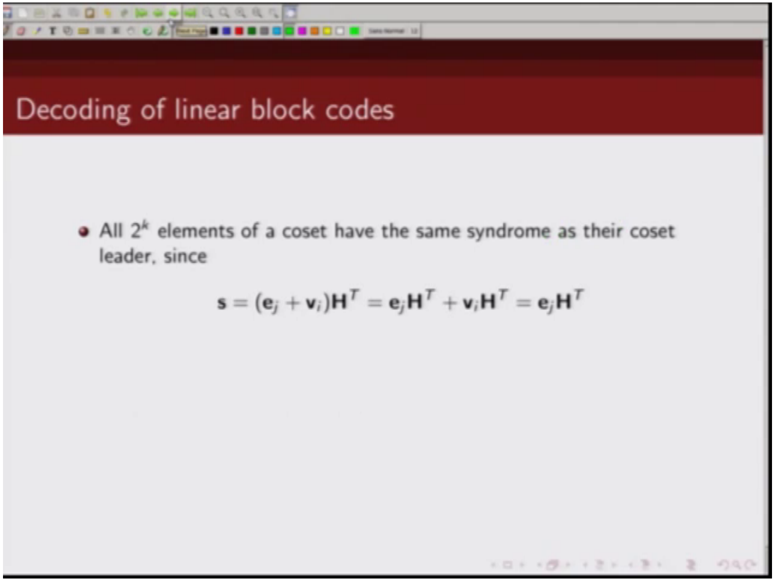
For a $(6, 3)$ linear code generated by the following matrix,

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

standard array is shown in next two slides.

The next

(Refer Slide Time 21:38)



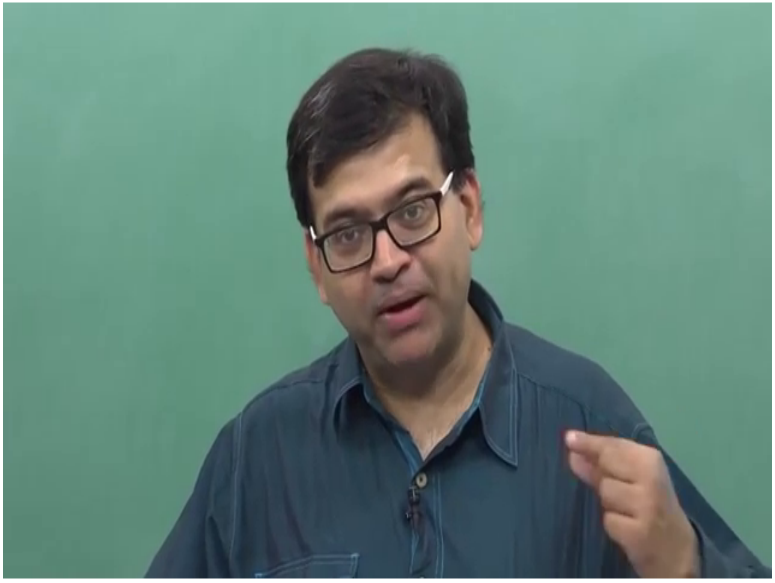
Decoding of linear block codes

- All 2^k elements of a coset have the same syndrome as their coset leader, since

$$\mathbf{s} = (\mathbf{e}_j + \mathbf{v}_i)\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T + \mathbf{v}_i\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T$$

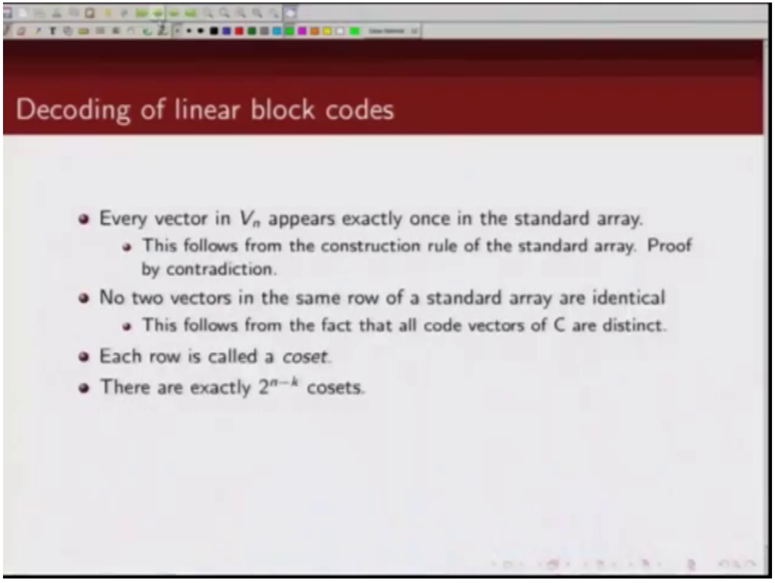
property is all the 2 k elements in a row or in a coset have the same Syndrome. This we can show because each element in the coset are of the form like this, e j plus v i H transpose and since v i H transpose is 0; they will only depend on the error pattern. And in each row, basically it's the same error pattern

(Refer Slide Time 22:08)



that appears in the elements of the row. We can again go back to our diagram

(Refer Slide Time 22:14)

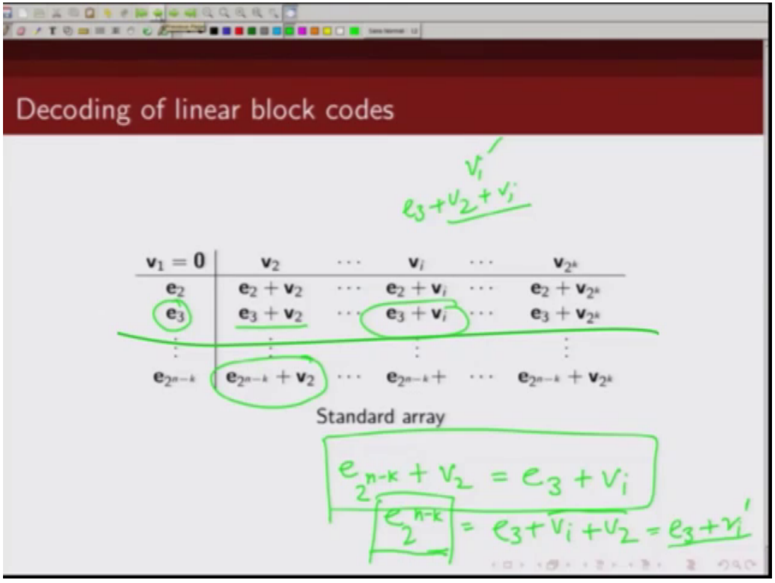


Decoding of linear block codes

- Every vector in V_n appears exactly once in the standard array.
  - This follows from the construction rule of the standard array. Proof by contradiction.
- No two vectors in the same row of a standard array are identical
  - This follows from the fact that all code vectors of C are distinct.
- Each row is called a *coset*.
- There are exactly 2^{n-k} cosets.

for standard array and we can see this. We can see in each row;

(Refer Slide Time 22:21)



Decoding of linear block codes

v_i'

$e_3 + v_2 + v_i$

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\mathbf{e}_3 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_3 + \mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}} +$	$\cdots$	$\mathbf{e}_{2^{n-k}} + \mathbf{v}_{2^k}$

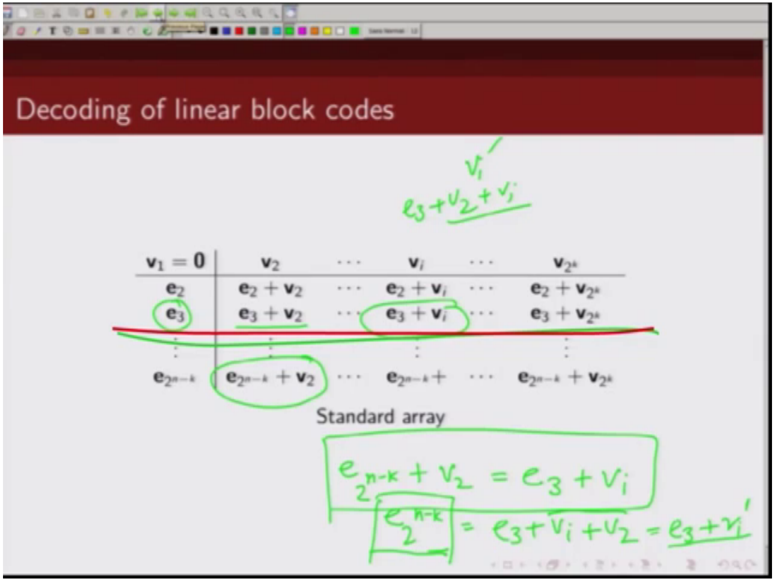
Standard array

$$\boxed{e_{2^{n-k}} + v_2 = e_3 + v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3 + v_i + v_2 = e_3 + v_i'$$

in this row, it's e 3,

(Refer Slide Time 22:23)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	e_2+v_2	$\cdots$	e_2+v_i	$\cdots$	$e_2+v_{2^k}$
e_3	e_3+v_2	$\cdots$	e_3+v_i	$\cdots$	$e_3+v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}}+v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}}+v_{2^k}$

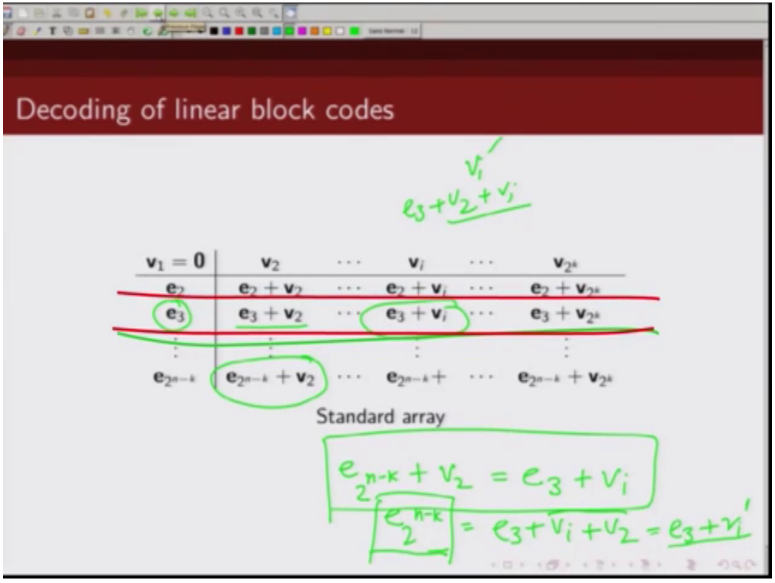
Standard array

$$\boxed{e_{2^{n-k}}+v_2 = e_3+v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3+v_i+v_2 = e_3+v_i'$$

in this row

(Refer Slide Time 22:25)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	e_2+v_2	$\cdots$	e_2+v_i	$\cdots$	$e_2+v_{2^k}$
e_3	e_3+v_2	$\cdots$	e_3+v_i	$\cdots$	$e_3+v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}}+v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}}+v_{2^k}$

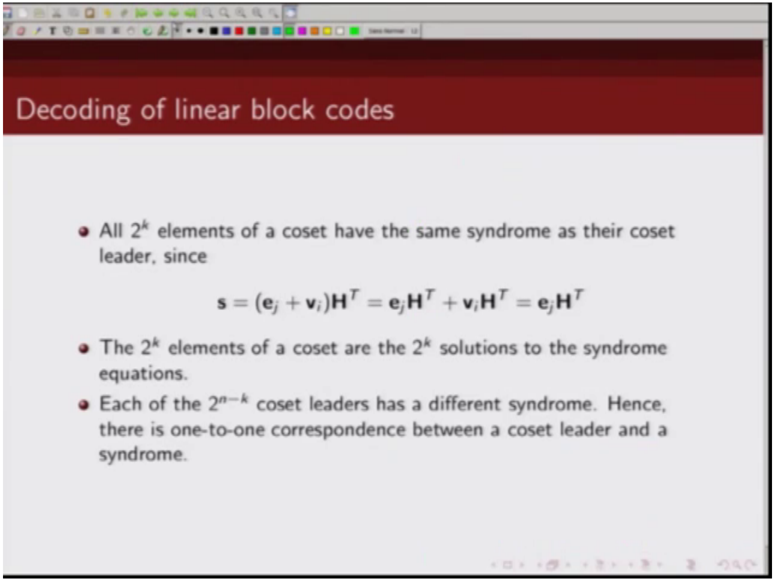
Standard array

$$\boxed{e_{2^{n-k}}+v_2 = e_3+v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3+v_i+v_2 = e_3+v_i'$$

all these elements have e 2. So they will have the same Syndrome and

(Refer Slide Time 22:33)



Decoding of linear block codes

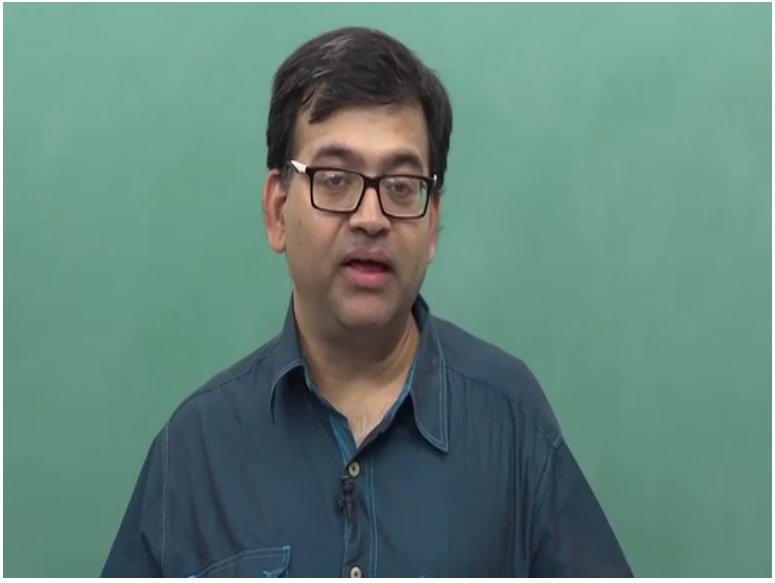
- All 2^k elements of a coset have the same syndrome as their coset leader, since

$$\mathbf{s} = (\mathbf{e}_j + \mathbf{v}_i)\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T + \mathbf{v}_i\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T$$

- The 2^k elements of a coset are the 2^k solutions to the syndrome equations.
- Each of the 2^{n-k} coset leaders has a different syndrome. Hence, there is one-to-one correspondence between a coset leader and a syndrome.

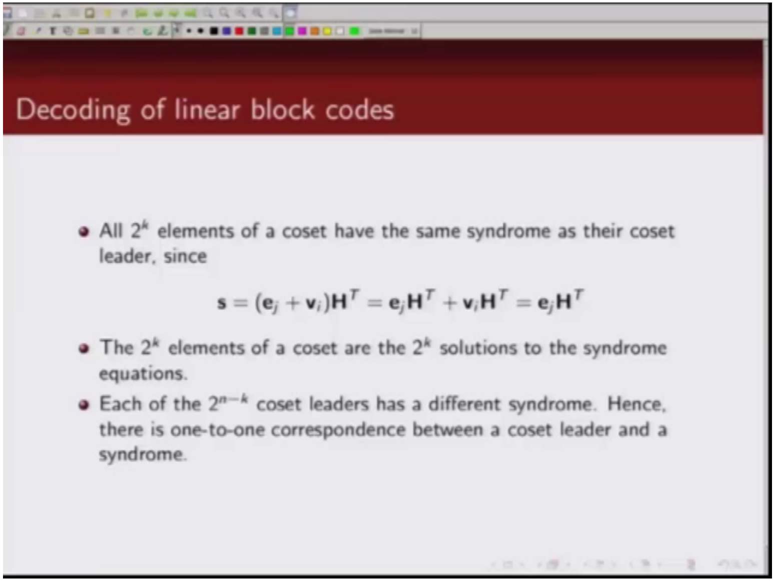
in the previous lecture we talked about that there are n minus k Syndrome equations and

(Refer Slide Time 22:41)



n unknowns and there are total 2 k solutions and you can see here, each row

(Refer Slide Time 22:47)



Decoding of linear block codes

- All 2^k elements of a coset have the same syndrome as their coset leader, since

$$\mathbf{s} = (\mathbf{e}_j + \mathbf{v}_i)\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T + \mathbf{v}_i\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T$$

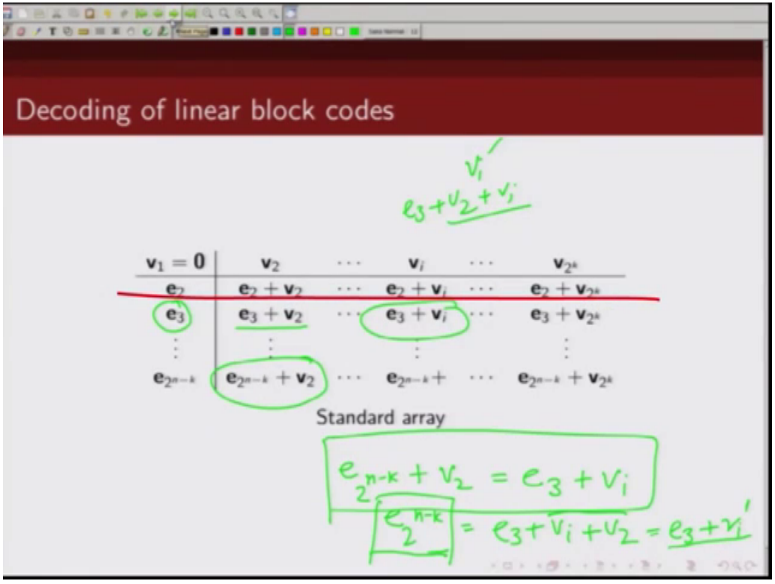
- The 2^k elements of a coset are the 2^k solutions to the syndrome equations.
- Each of the 2^{n-k} coset leaders has a different syndrome. Hence, there is one-to-one correspondence between a coset leader and a syndrome.

has 2 k elements and they have the same Syndrome. So these 2 k elements are exactly the solution of your Syndrome equations because they, these 2 k elements of a row, of a coset, they all have the same Syndrome and they correspond to the 2 k solutions of the Syndrome equations. So the 2 k elements of a coset are actually the solutions of your Syndrome equation. An another interesting thing is each of these cosets or each of these coset leader will have

(Refer Slide Time 23:30)

another interesting thing is each of these cosets or each of these coset leader will have a different Syndrome. Why, because each of these row, if you look at each of these row, each of them corresponds to different error pattern. Again let's go back to the diagram that we had for the standard array. This row split it to e 2.

(Refer Slide Time 23:55)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	e_2+v_2	$\cdots$	e_2+v_i	$\cdots$	$e_2+v_{2^k}$
e_3	e_3+v_2	$\cdots$	e_3+v_i	$\cdots$	$e_3+v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}}+v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}}+v_{2^k}$

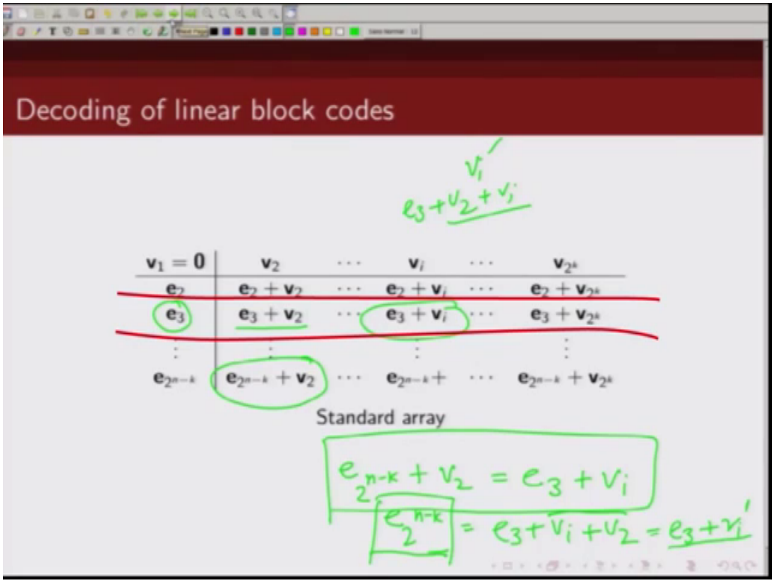
Standard array

$e_{2^{n-k}} + v_2 = e_3 + v_i$

$e_{2^{n-k}} = e_3+v_i+v_2 = e_3+v_i'$

So if we compute Syndrome for any of these received vectors we will get Syndrome corresponds to e 2. This row corresponds to e 3.

(Refer Slide Time 24:06)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	e_2+v_2	$\cdots$	e_2+v_i	$\cdots$	$e_2+v_{2^k}$
e_3	e_3+v_2	$\cdots$	e_3+v_i	$\cdots$	$e_3+v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}}+v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}}+v_{2^k}$

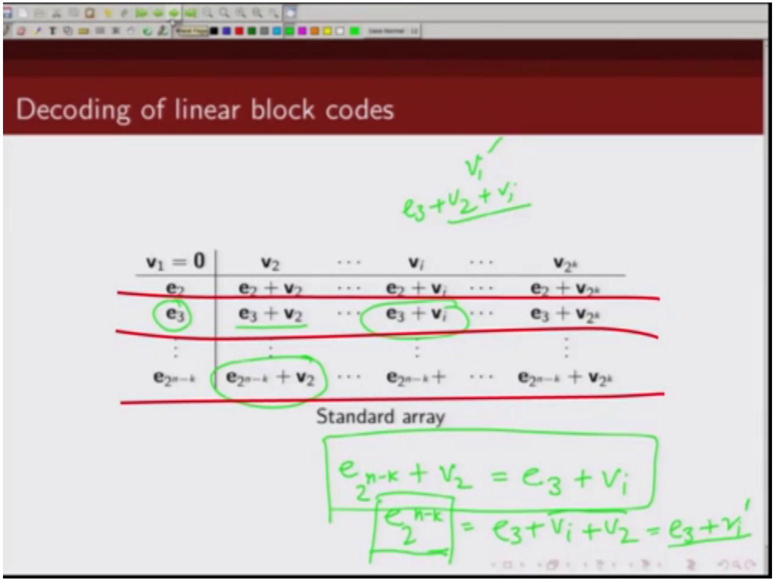
Standard array

$e_{2^{n-k}} + v_2 = e_3 + v_i$

$e_{2^{n-k}} = e_3+v_i+v_2 = e_3+v_i'$

This e 4, e raised to power n minus k.

(Refer Slide Time 24:09)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_2+\mathbf{v}_i$	$\cdots$	$\mathbf{e}_2+\mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\mathbf{e}_3+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_3+\mathbf{v}_i$	$\cdots$	$\mathbf{e}_3+\mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}}+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}}+$	$\cdots$	$\mathbf{e}_{2^{n-k}}+\mathbf{v}_{2^k}$

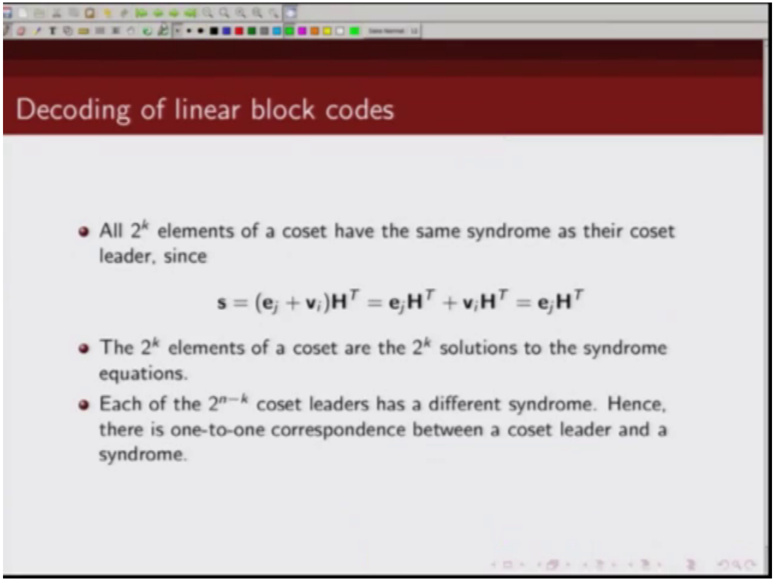
Standard array

$$\boxed{e_{2^{n-k}} + v_2 = e_3 + v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3 + v_i + v_2 = \underline{e_3 + v_i'}$$

So if you look at Syndrome for each of these rows, they all correspond to, each row corresponding to a different Syndrome. But within a row, the Syndrome is same. So in other words, you can map one Syndrome to one row or you can map one Syndrome to one coset. So this is interesting that

(Refer Slide Time 24:39)



Decoding of linear block codes

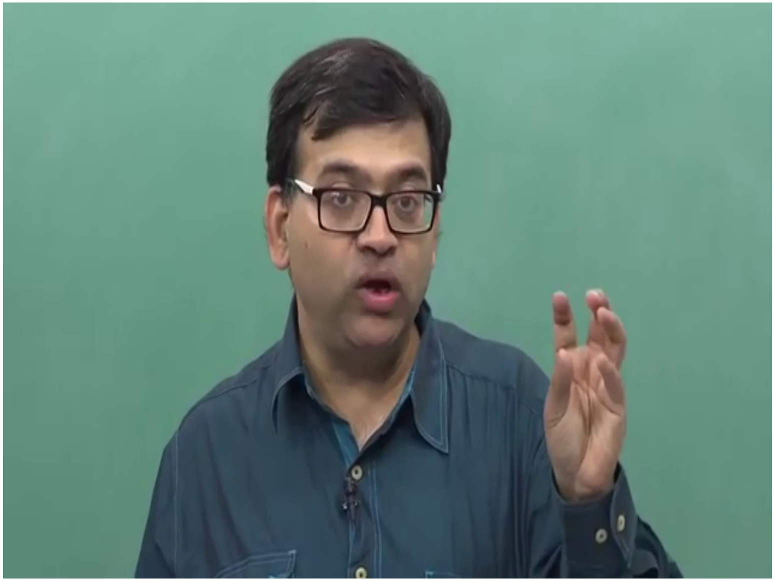
- All 2^k elements of a coset have the same syndrome as their coset leader, since

$$\mathbf{s} = (\mathbf{e}_j + \mathbf{v}_i)\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T + \mathbf{v}_i\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T$$

- The 2^k elements of a coset are the 2^k solutions to the syndrome equations.
- Each of the 2^{n-k} coset leaders has a different syndrome. Hence, there is one-to-one correspondence between a coset leader and a syndrome.

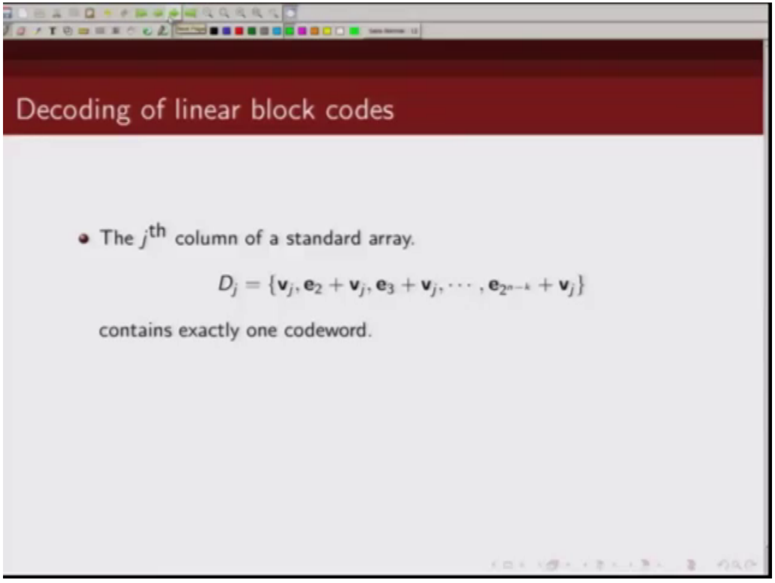
now we have one to one mapping between

(Refer Slide Time 24:44)



each cosets or each coset leader to a Syndrome.

(Refer Slide Time 24:52)



Decoding of linear block codes

- The j^{th} column of a standard array.

$$D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \cdots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}$$

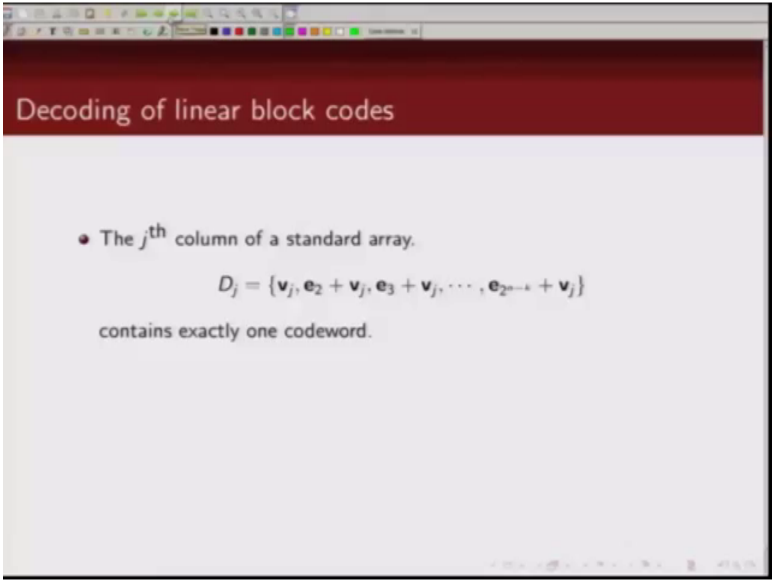
contains exactly one codeword.

Another thing to note is, if we look at columns of this standard array, each column of the standard array corresponds to one particular codeword.

(Refer Slide Time 25:03)

If you recall when we started the lecture we said we want to partition our 2 n vectors into 2 k different sets and each of these 2 k set should correspond to only one

(Refer Slide Time 25:19)



Decoding of linear block codes

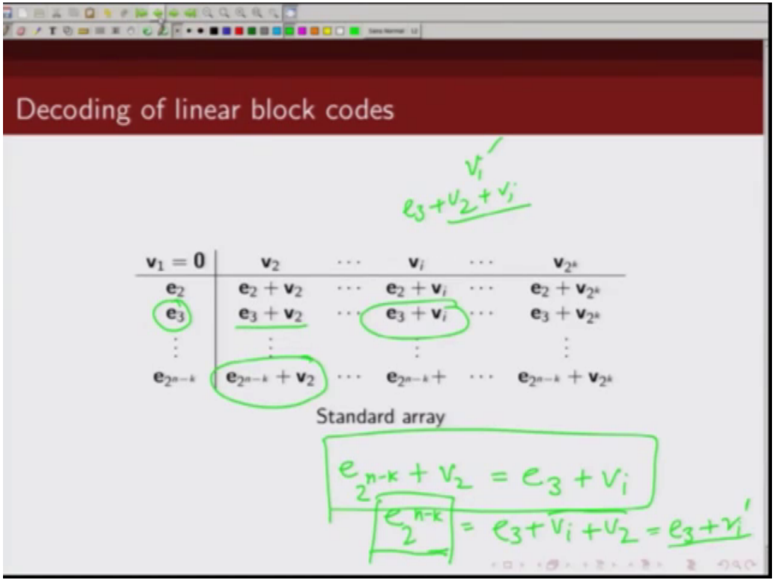
- The j^{th} column of a standard array.

$$D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \cdots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}$$

contains exactly one codeword.

codeword and this is what is happening here as well.

(Refer Slide Time 25:25)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	e_2+v_2	$\cdots$	e_2+v_i	$\cdots$	$e_2+v_{2^k}$
e_3	e_3+v_2	$\cdots$	e_3+v_i	$\cdots$	$e_3+v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}}+v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}}+v_{2^k}$

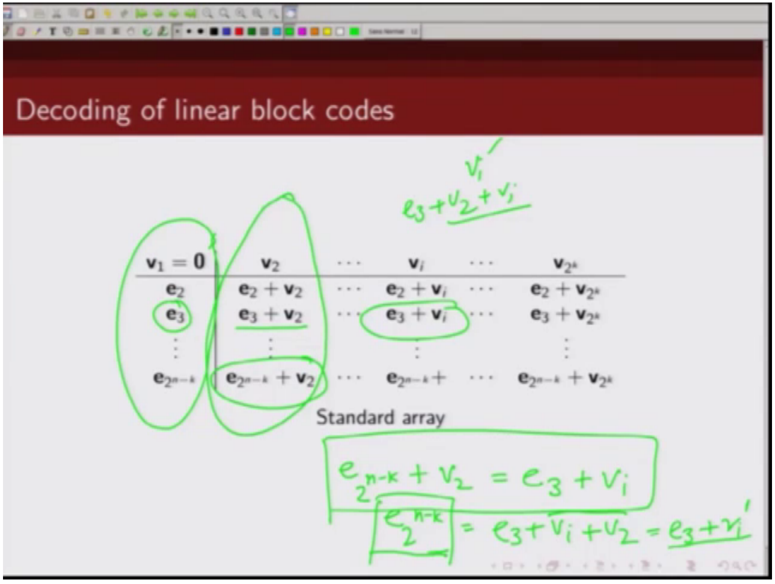
Standard array

$$\boxed{e_{2^{n-k}}+v_2 = e_3+v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3+v_i+v_2 = e_3+v_i'$$

You have this whole thing as total possible 2 n vectors. Now we have already partitioned them into 2 k different partitions and these are all distinct

(Refer Slide Time 25:38)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	e_2+v_2	$\cdots$	e_2+v_i	$\cdots$	$e_2+v_{2^k}$
e_3	e_3+v_2	$\cdots$	e_3+v_i	$\cdots$	$e_3+v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}}+v_2$	$\cdots$	$e_{2^{n-k}}+$	$\cdots$	$e_{2^{n-k}}+v_{2^k}$

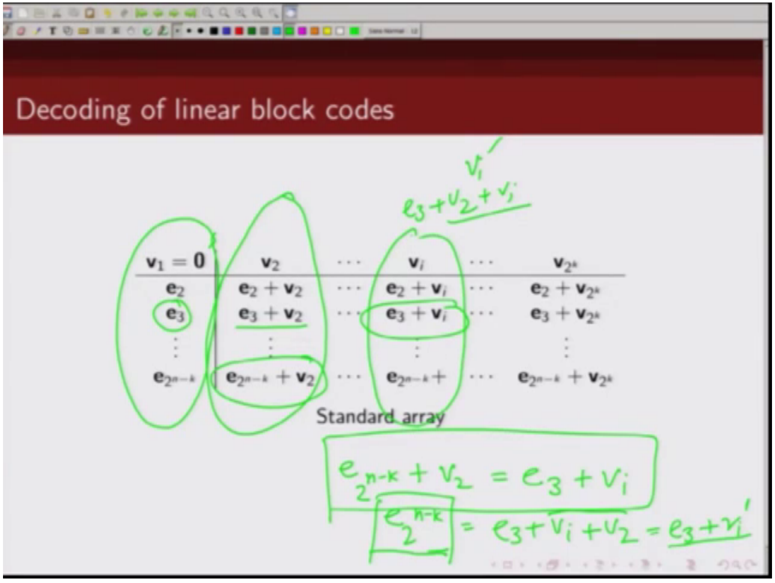
Standard array

$$\boxed{e_{2^{n-k}}+v_2 = e_3+v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3+v_i+v_2 = e_3+v_i'$$

partitions. There is no element in here which is

(Refer Slide Time 25:43)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_2+\mathbf{v}_i$	$\cdots$	$\mathbf{e}_2+\mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\mathbf{e}_3+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_3+\mathbf{v}_i$	$\cdots$	$\mathbf{e}_3+\mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}}+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}}+$	$\cdots$	$\mathbf{e}_{2^{n-k}}+\mathbf{v}_{2^k}$

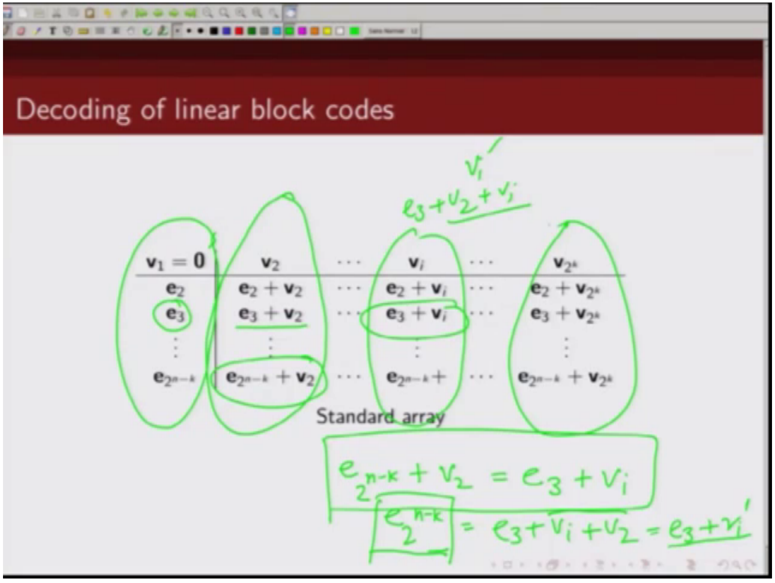
Standard array

$$\boxed{e_{2^{n-k}}+v_2 = e_3+v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3+v_i+v_2 = e_3+v_i'$$

common with this element. And another thing to be of interest,

(Refer Slide Time 25:48)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_2+\mathbf{v}_i$	$\cdots$	$\mathbf{e}_2+\mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\mathbf{e}_3+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_3+\mathbf{v}_i$	$\cdots$	$\mathbf{e}_3+\mathbf{v}_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$\mathbf{e}_{2^{n-k}}$	$\mathbf{e}_{2^{n-k}}+\mathbf{v}_2$	$\cdots$	$\mathbf{e}_{2^{n-k}}+$	$\cdots$	$\mathbf{e}_{2^{n-k}}+\mathbf{v}_{2^k}$

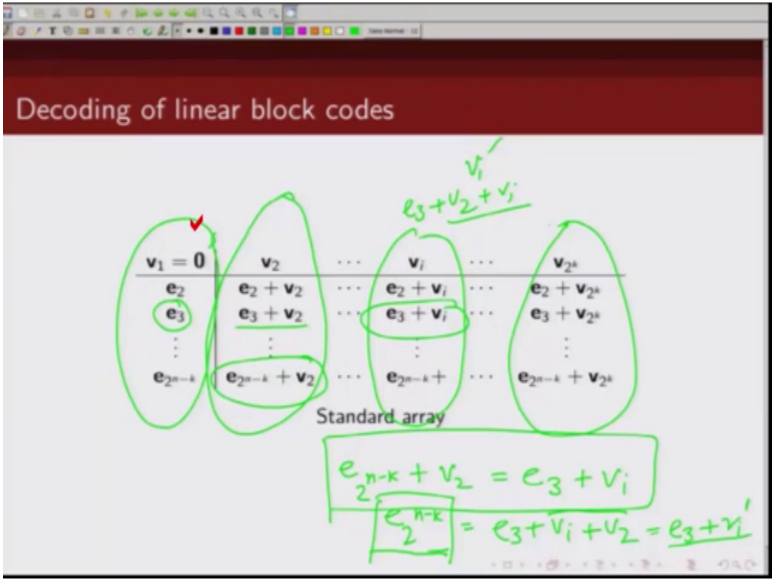
Standard array

$$\boxed{e_{2^{n-k}}+v_2 = e_3+v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3+v_i+v_2 = e_3+v_i'$$

if you look at each of these partitions, this partition has all zero vector.

(Refer Slide Time 25:54)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	$e_2 + v_2$	$\cdots$	$e_2 + v_i$	$\cdots$	$e_2 + v_{2^k}$
e_3	$e_3 + v_2$	$\cdots$	$e_3 + v_i$	$\cdots$	$e_3 + v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	$\cdots$	$e_{2^{n-k}} +$	$\cdots$	$e_{2^{n-k}} + v_{2^k}$

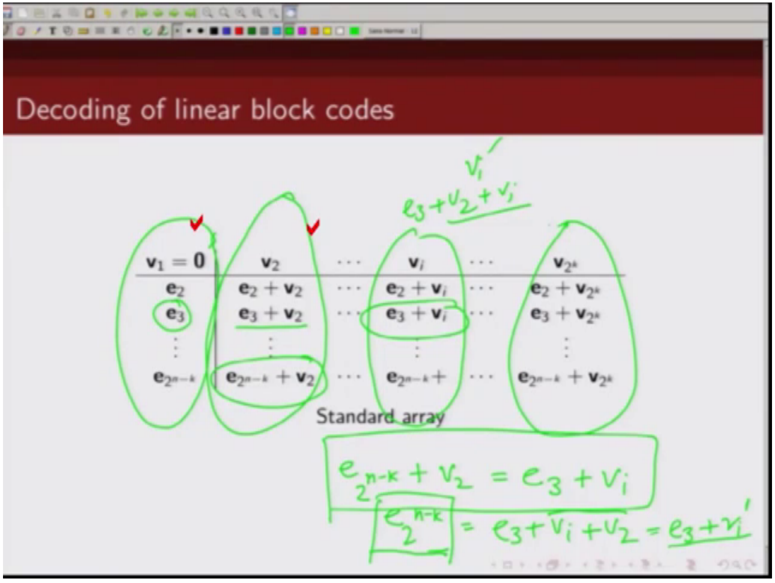
Standard array

$$e_{2^{n-k}} + v_2 = e_3 + v_i$$

$$\boxed{e_{2^{n-k}}} = e_3 + v_i + v_2 = e_3 + v_i'$$

This partition corresponds

(Refer Slide Time 25:55)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$v_1 = 0$	v_2	$\cdots$	v_i	$\cdots$	v_{2^k}
e_2	$e_2 + v_2$	$\cdots$	$e_2 + v_i$	$\cdots$	$e_2 + v_{2^k}$
e_3	$e_3 + v_2$	$\cdots$	$e_3 + v_i$	$\cdots$	$e_3 + v_{2^k}$
$\vdots$	$\vdots$		$\vdots$		$\vdots$
$e_{2^{n-k}}$	$e_{2^{n-k}} + v_2$	$\cdots$	$e_{2^{n-k}} +$	$\cdots$	$e_{2^{n-k}} + v_{2^k}$

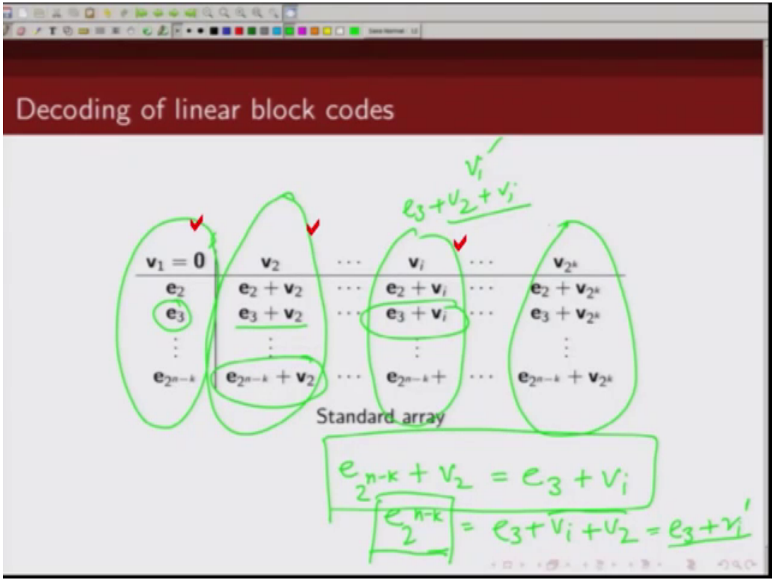
Standard array

$$e_{2^{n-k}} + v_2 = e_3 + v_i$$

$$\boxed{e_{2^{n-k}}} = e_3 + v_i + v_2 = e_3 + v_i'$$

to v 2. This partition corresponds to

(Refer Slide Time 25:58)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$v_1 = 0$	v_2	⋯	v_i	⋯	v_{2^k}
e_2	e_2+v_2	⋯	e_2+v_i	⋯	$e_2+v_{2^k}$
e_3	e_3+v_2	⋯	e_3+v_i	⋯	$e_3+v_{2^k}$
⋮	⋮		⋮		⋮
$e_{2^{n-k}}$	$e_{2^{n-k}}+v_2$	⋯	$e_{2^{n-k}}+$	⋯	$e_{2^{n-k}}+v_{2^k}$

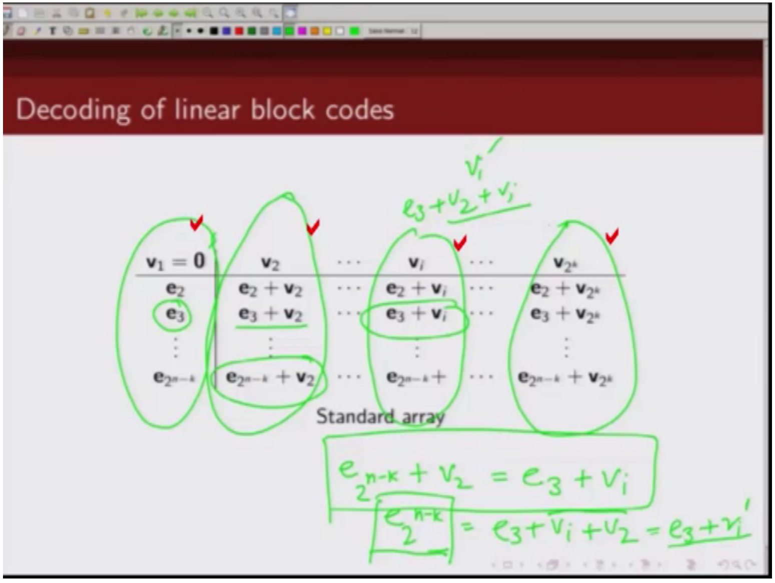
Standard array

$$\boxed{e_{2^{n-k}}+v_2 = e_3+v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3+v_i+v_2 = e_3+v_i'$$

v i. This partition corresponds to v 2 k.

(Refer Slide Time 26:01)



Decoding of linear block codes

v_i'

$e_3+v_2+v_i$

$v_1 = 0$	v_2	⋯	v_i	⋯	v_{2^k}
e_2	e_2+v_2	⋯	e_2+v_i	⋯	$e_2+v_{2^k}$
e_3	e_3+v_2	⋯	e_3+v_i	⋯	$e_3+v_{2^k}$
⋮	⋮		⋮		⋮
$e_{2^{n-k}}$	$e_{2^{n-k}}+v_2$	⋯	$e_{2^{n-k}}+$	⋯	$e_{2^{n-k}}+v_{2^k}$

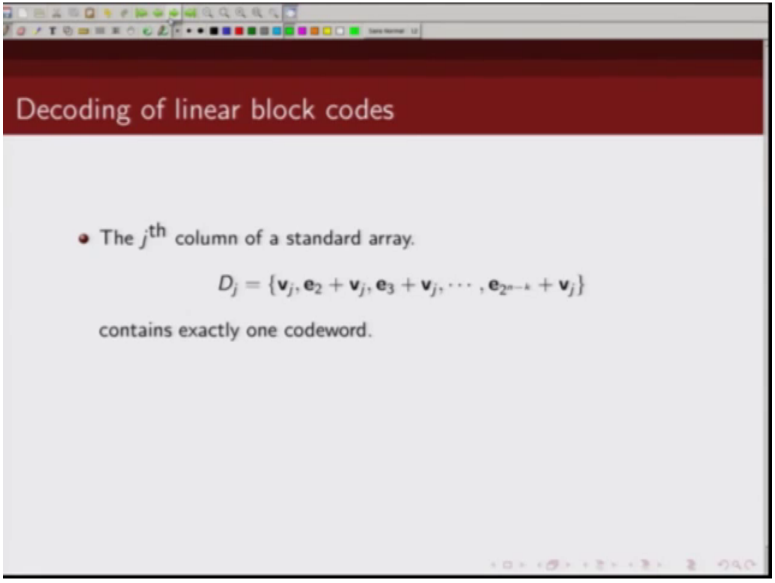
Standard array

$$\boxed{e_{2^{n-k}}+v_2 = e_3+v_i}$$

$$\boxed{e_{2^{n-k}}} = e_3+v_i+v_2 = e_3+v_i'$$

So this is a point basically I am trying to make that the way we have created the standard array, if you look at

(Refer Slide Time 26:12)



Decoding of linear block codes

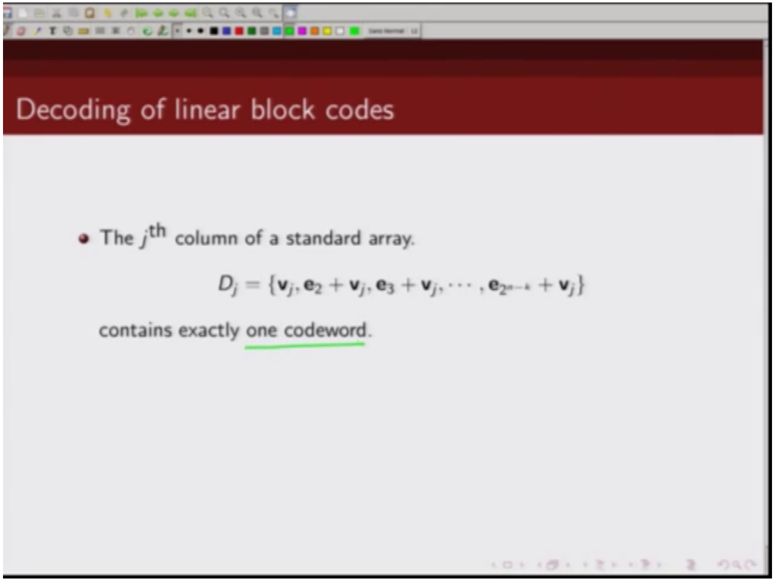
- The j^{th} column of a standard array.

$$D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \cdots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}$$

contains exactly one codeword.

the jth column of the standard array it contains exactly

(Refer Slide Time 26:17)



Decoding of linear block codes

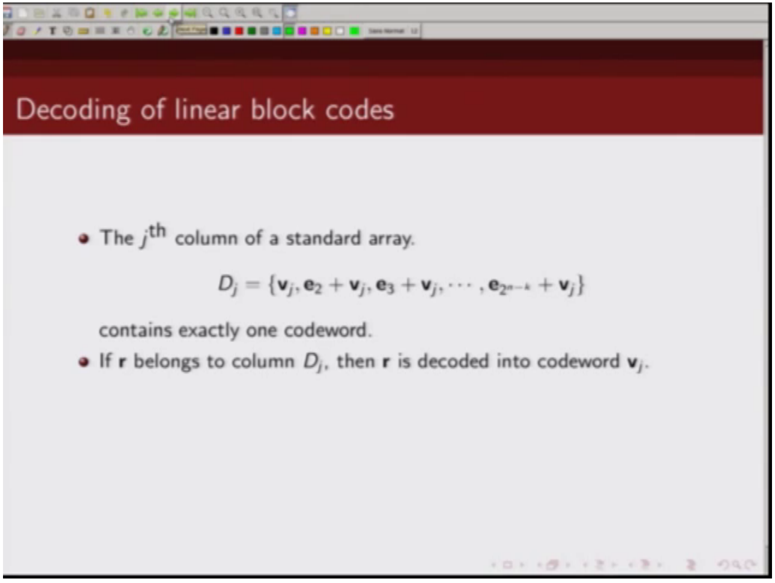
- The j^{th} column of a standard array.

$$D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \cdots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}$$

contains exactly one codeword.

one codeword. It corresponds to only one particular codeword. Now if received

(Refer Slide Time 26:26)



Decoding of linear block codes

- The j^{th} column of a standard array.

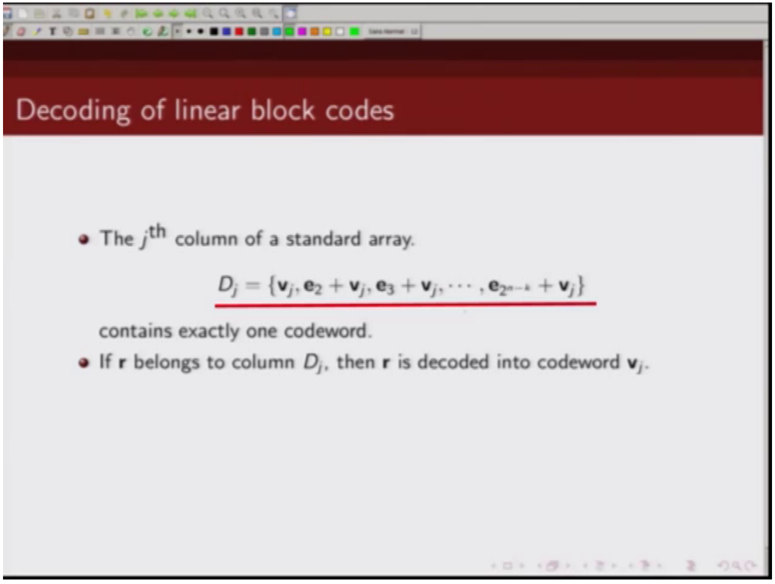
$$D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \cdots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}$$

contains exactly one codeword.

- If **r** belongs to column D_j, then **r** is decoded into codeword $\mathbf{v}_j$.

codeword belongs to column d j then r will be decoded as codeword v j. So whenever r belongs to a set this, we will decode

(Refer Slide Time 26:40)



Decoding of linear block codes

- The j^{th} column of a standard array.

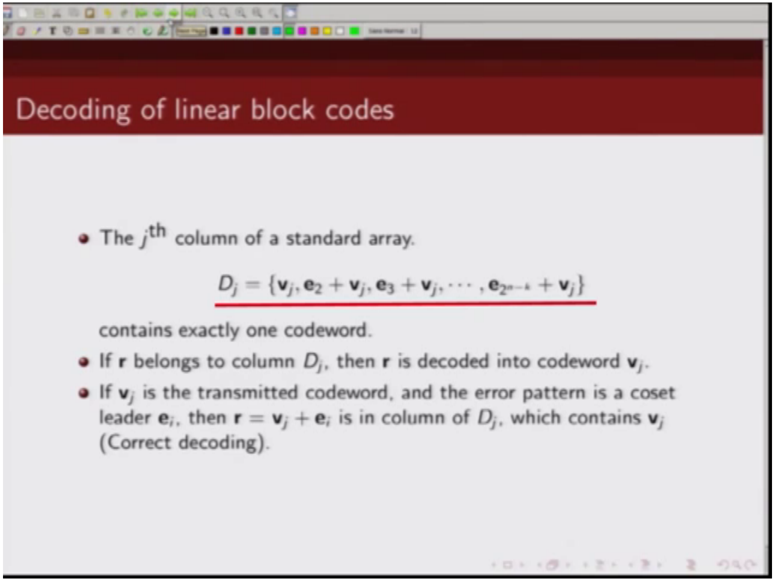
$$\underline{D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \cdots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}}$$

contains exactly one codeword.

- If **r** belongs to column D_j, then **r** is decoded into codeword $\mathbf{v}_j$.

this r as correspond to codeword v j.

(Refer Slide Time 26:47)



Decoding of linear block codes

- The j^{th} column of a standard array.

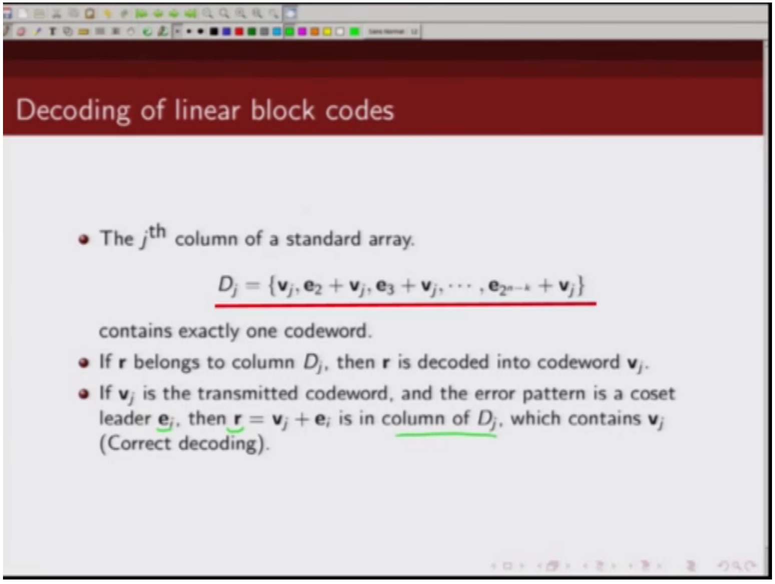
$$\underline{D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \cdots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}}$$

contains exactly one codeword.

- If $\mathbf{r}$ belongs to column D_j, then $\mathbf{r}$ is decoded into codeword $\mathbf{v}_j$.
- If $\mathbf{v}_j$ is the transmitted codeword, and the error pattern is a coset leader $\mathbf{e}_i$, then $\mathbf{r} = \mathbf{v}_j + \mathbf{e}_i$ is in column of D_j, which contains $\mathbf{v}_j$ (Correct decoding).

So if v j is our transmitted codeword, and the error pattern is e i and if the received sequence is in, falls in the column

(Refer Slide Time 26:59)



Decoding of linear block codes

- The j^{th} column of a standard array.

$$\underline{D_j = \{\mathbf{v}_j, \mathbf{e}_2 + \mathbf{v}_j, \mathbf{e}_3 + \mathbf{v}_j, \cdots, \mathbf{e}_{2^{n-k}} + \mathbf{v}_j\}}$$

contains exactly one codeword.

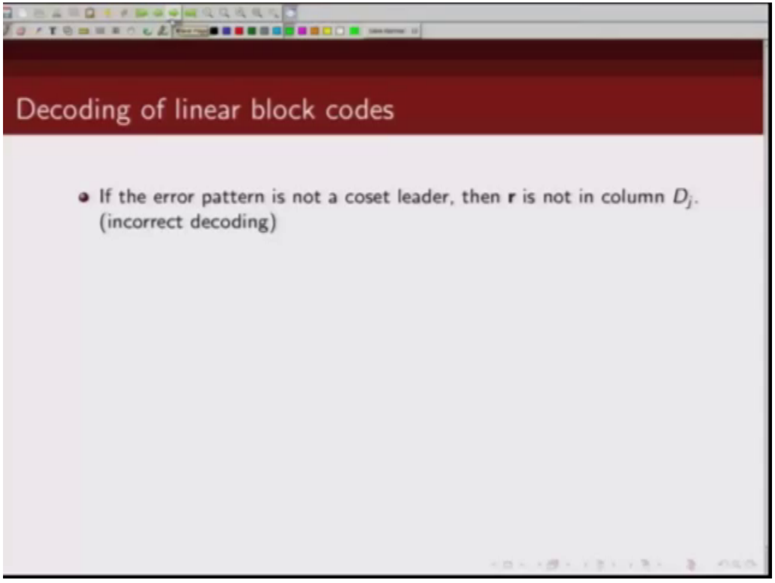
- If $\mathbf{r}$ belongs to column D_j, then $\mathbf{r}$ is decoded into codeword $\mathbf{v}_j$.
- If $\mathbf{v}_j$ is the transmitted codeword, and the error pattern is a coset leader $\underline{\mathbf{e}_i}$, then $\underline{\mathbf{r}} = \mathbf{v}_j + \mathbf{e}_i$ is in $\underline{\text{column of } D_j}$, which contains $\mathbf{v}_j$ (Correct decoding).

v j then we would decode it as v j which is correct decoding. We would not make

(Refer Slide Time 27:07)

any error. However if the error pattern is not a coset leader, then r will not be

(Refer Slide Time 27:19)

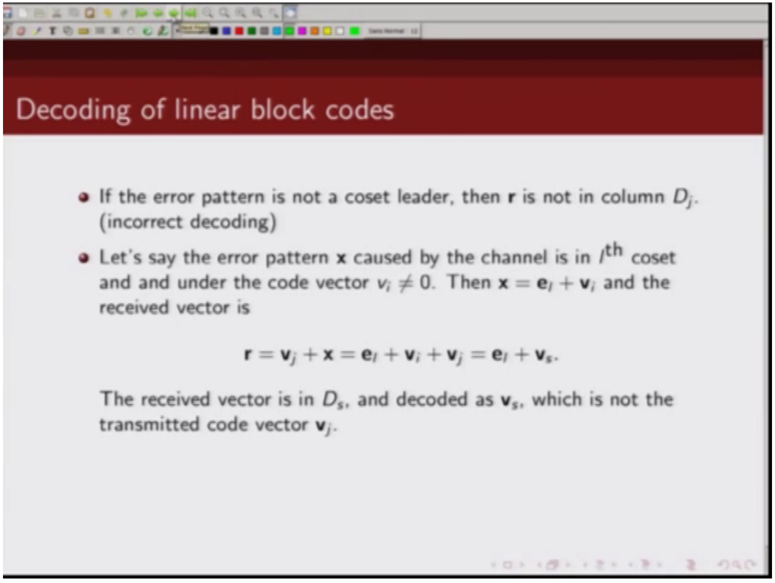


Decoding of linear block codes

- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)

in column v j and in that case we will make an error in decoding.

(Refer Slide Time 27:29)



Decoding of linear block codes

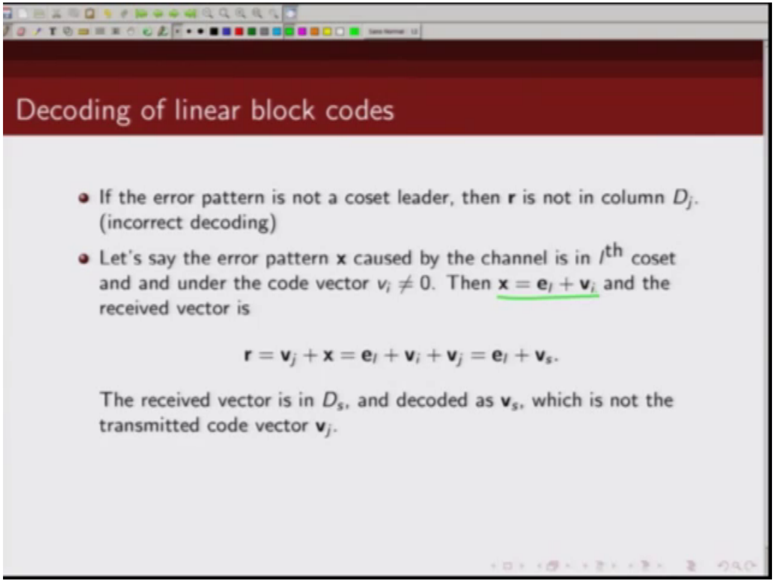
- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and and under the code vector $v_i \neq 0$. Then $\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i$ and the received vector is

$$\mathbf{r} = \mathbf{v}_j + \mathbf{x} = \mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j = \mathbf{e}_l + \mathbf{v}_s.$$

The received vector is in D_s, and decoded as $\mathbf{v}_s$, which is not the transmitted code vector $\mathbf{v}_j$.

So let us look at an error pattern x caused by a channel and it is in the lth coset. So if it is in the lth coset we are writing this as, let us

(Refer Slide Time 27:44)



Decoding of linear block codes

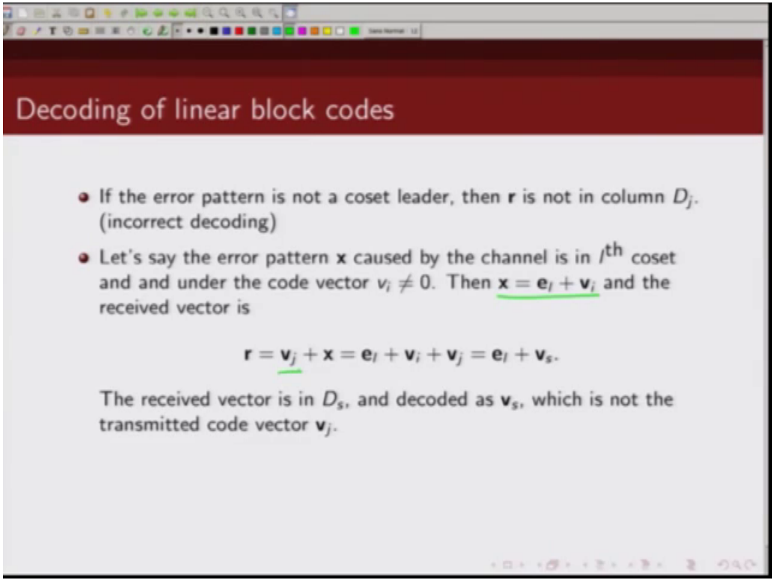
- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and and under the code vector $v_i \neq 0$. Then $\underline{\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i}$ and the received vector is

$$\mathbf{r} = \mathbf{v}_j + \mathbf{x} = \mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j = \mathbf{e}_l + \mathbf{v}_s.$$

The received vector is in D_s, and decoded as $\mathbf{v}_s$, which is not the transmitted code vector $\mathbf{v}_j$.

say error pattern as e l corresponding to the error pattern e l plus v i. So in that case if we are transmitting codeword v j

(Refer Slide Time 27:55)



Decoding of linear block codes

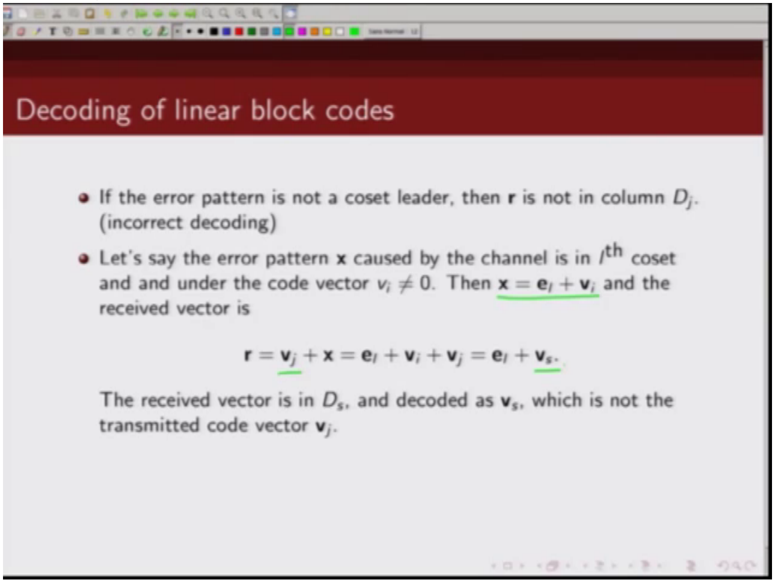
- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and and under the code vector $v_i \neq 0$. Then $\underline{\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i}$ and the received vector is

$$\mathbf{r} = \underline{\mathbf{v}_j} + \mathbf{x} = \mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j = \mathbf{e}_l + \mathbf{v}_s.$$

The received vector is in D_s, and decoded as $\mathbf{v}_s$, which is not the transmitted code vector $\mathbf{v}_j$.

and we encounter this error pattern, what we would receive is v j plus 6, this would be nothing but e l plus v i plus v j. Now what is v i plus v j? v i plus v j sum of two codewords is also a valid codeword. Let's call that codeword as v s.

(Refer Slide Time 28:19)



Decoding of linear block codes

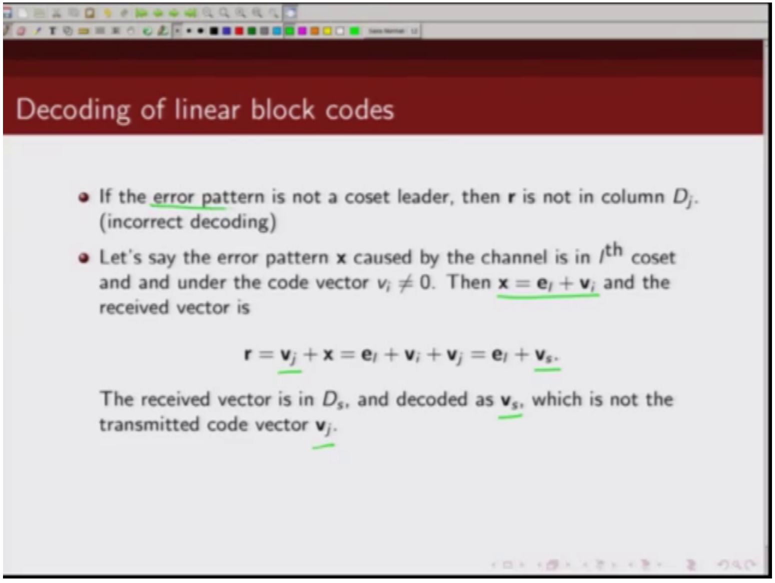
- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and and under the code vector $v_i \neq 0$. Then $\underline{\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i}$ and the received vector is

$$\mathbf{r} = \underline{\mathbf{v}_j} + \mathbf{x} = \mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j = \mathbf{e}_l + \underline{\mathbf{v}_s}.$$

The received vector is in D_s, and decoded as $\mathbf{v}_s$, which is not the transmitted code vector $\mathbf{v}_j$.

So now the received vector is in partition d s. And in this case, what are we going to decode it as? We are going to decode it as v s which is not same as the transmitted codeword v j. So that's what I meant. If your error pattern is not

(Refer Slide Time 28:41)



Decoding of linear block codes

- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and and under the code vector $v_i \neq 0$. Then $\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i$ and the received vector is

$$\mathbf{r} = \mathbf{v}_j + \mathbf{x} = \mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j = \mathbf{e}_l + \mathbf{v}_s.$$

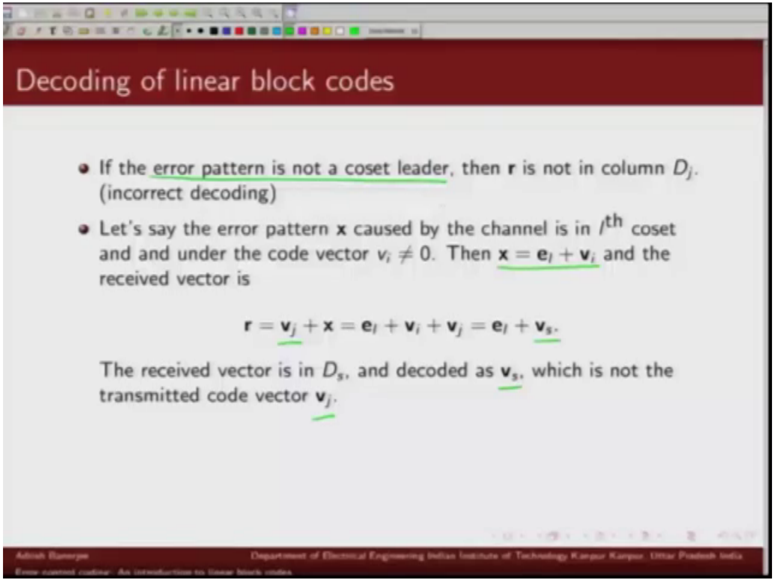
The received vector is in D_s, and decoded as $\mathbf{v}_s$, which is not the transmitted code vector $\mathbf{v}_j$.

a coset leader then r would not be in the same column as v j. If this would have been coset leader this would have been just e l. And then received sequence would have been just e l plus v j. So this would have still remained in the partition d j. And we would not have made a mistake, Ok.

(Refer Slide Time 29:08)

So the error pattern is not a coset leader then we

(Refer Slide Time 29:12)



Decoding of linear block codes

- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and and under the code vector $v_i \neq 0$. Then $\underline{\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i}$ and the received vector is

$$\mathbf{r} = \underline{\mathbf{v}_j} + \mathbf{x} = \mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j = \mathbf{e}_l + \underline{\mathbf{v}_s}.$$

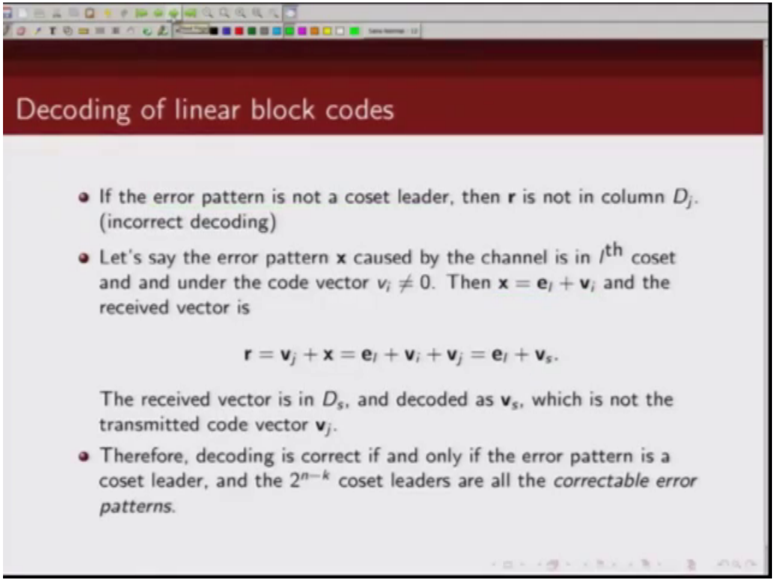
The received vector is in D_s, and decoded as $\underline{\mathbf{v}_s}$, which is not the transmitted code vector $\underline{\mathbf{v}_j}$.

Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India

Error control coding: An introduction to linear block codes

are basically going to make an error.

(Refer Slide Time 29:17)



Decoding of linear block codes

- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and and under the code vector $v_i \neq 0$. Then $\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i$ and the received vector is

$$\mathbf{r} = \mathbf{v}_j + \mathbf{x} = \mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j = \mathbf{e}_l + \mathbf{v}_s.$$

The received vector is in D_s, and decoded as $\mathbf{v}_s$, which is not the transmitted code vector $\mathbf{v}_j$.

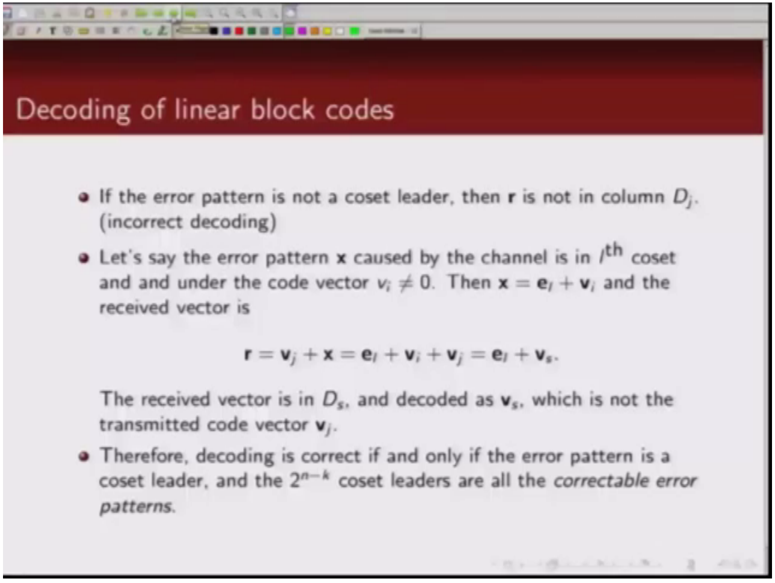
- Therefore, decoding is correct if and only if the error pattern is a coset leader, and the 2^{n-k} coset leaders are all the *correctable error patterns*.

So from this we can conclude that

(Refer Slide Time 29:21)

our decoding is going to be correct if and only if our error pattern is a coset leader.

(Refer Slide Time 29:32)



Decoding of linear block codes

- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and and under the code vector $v_i \neq 0$. Then $\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i$ and the received vector is

$$\mathbf{r} = \mathbf{v}_j + \mathbf{x} = \mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j = \mathbf{e}_l + \mathbf{v}_s.$$

The received vector is in D_s, and decoded as $\mathbf{v}_s$, which is not the transmitted code vector $\mathbf{v}_j$.

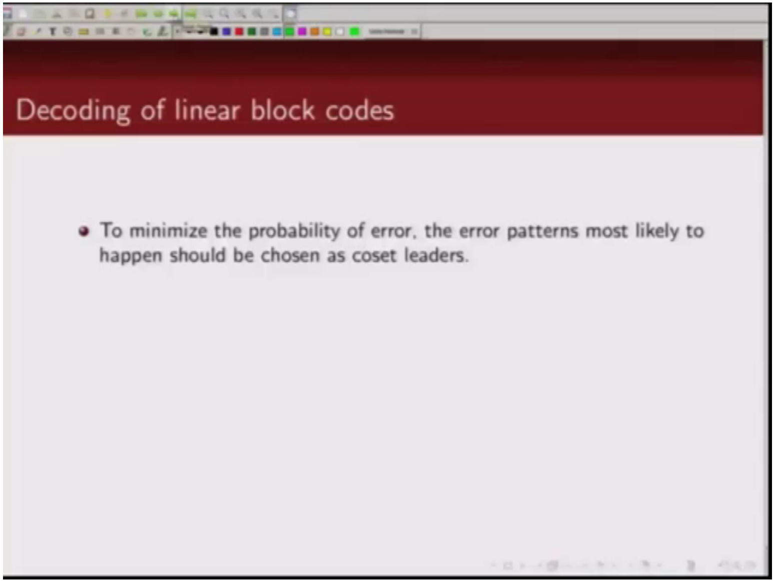
- Therefore, decoding is correct if and only if the error pattern is a coset leader, and the 2^{n-k} coset leaders are all the *correctable error patterns*.

And how many such coset leaders exist? We have total 2 n minus k. So this we call as correctable error patterns. Because whenever

(Refer Slide Time 29:45)

our error pattern is coset leader we are not going to make a mistake in decoding and hence we call these as correctable error pattern.

(Refer Slide Time 29:57)



Decoding of linear block codes

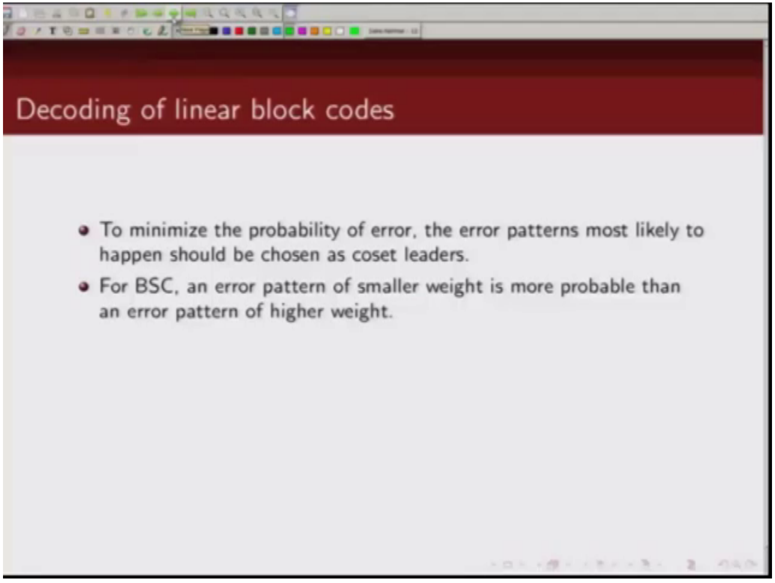
- To minimize the probability of error, the error patterns most likely to happen should be chosen as coset leaders.

Now the next question to think about is

(Refer Slide Time 30:02)

how should we choose our coset leader? Clearly our objective is to minimize probability of error. So the error patterns that are more likely to happen, we should choose them as our error, coset leader and what are those error patterns? These are error patterns which have least Hamming weight. So we start with, first start with error pattern of Hamming weight 1. If we run out of them, start with Hamming weight 2, 3 like that. Because they are more likely error pattern

(Refer Slide Time 30:49)

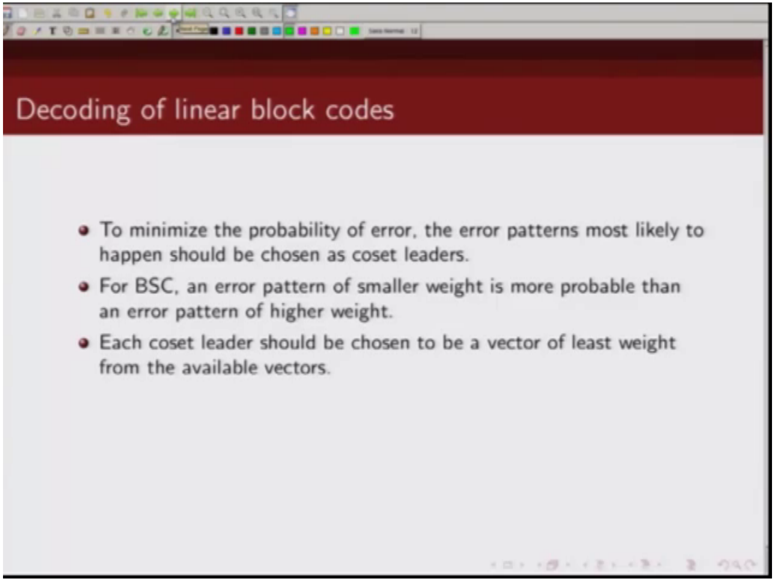


Decoding of linear block codes

- To minimize the probability of error, the error patterns most likely to happen should be chosen as coset leaders.
- For BSC, an error pattern of smaller weight is more probable than an error pattern of higher weight.

and we have shown that for binary symmetric channel, the error pattern with smaller weights are more likely than error pattern with

(Refer Slide Time 30:58)

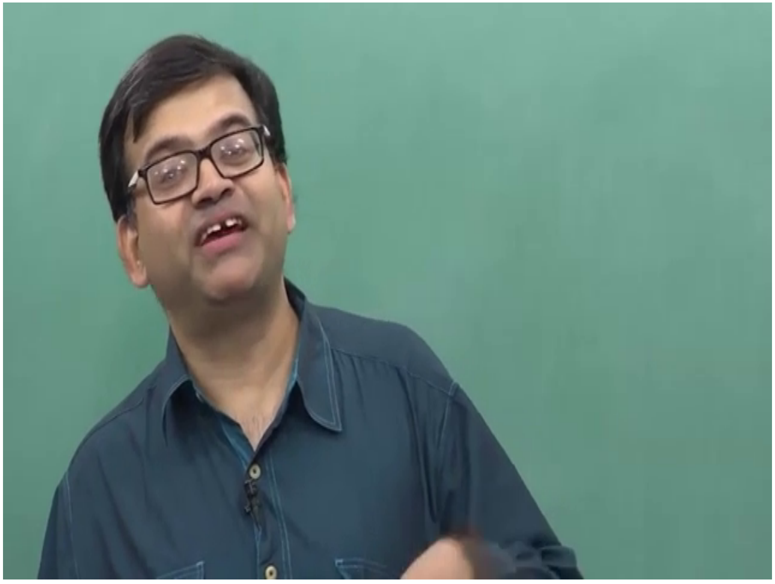


Decoding of linear block codes

- To minimize the probability of error, the error patterns most likely to happen should be chosen as coset leaders.
- For BSC, an error pattern of smaller weight is more probable than an error pattern of higher weight.
- Each coset leader should be chosen to be a vector of least weight from the available vectors.

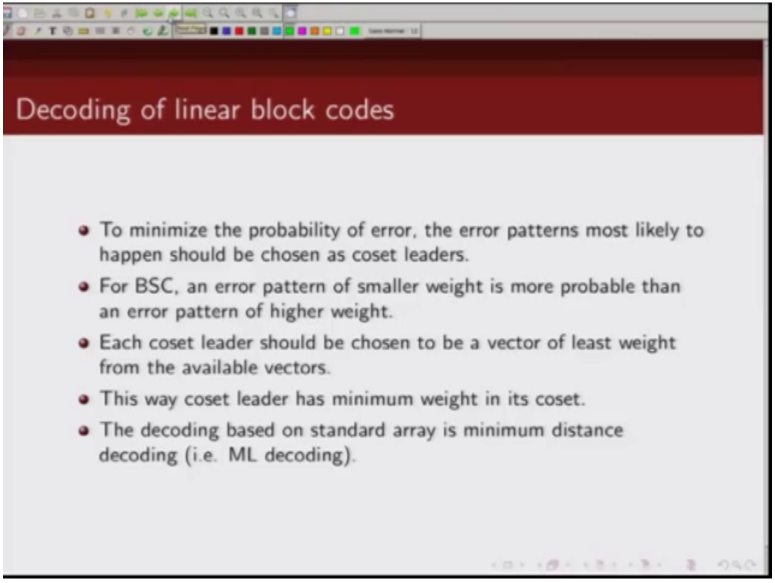
larger Hamming weight. So among the coset we should choose the element which has basically

(Refer Slide Time 31:07)



the smallest error pattern as our coset leader.

(Refer Slide Time 31:16)

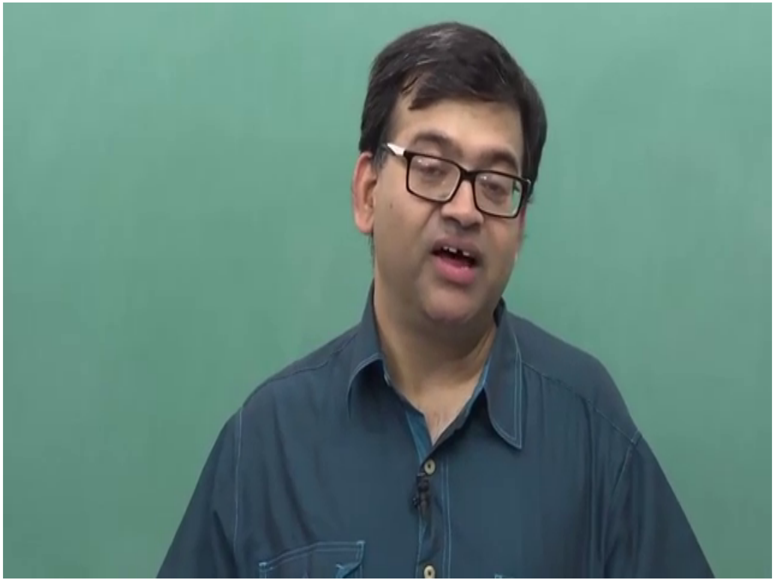


Decoding of linear block codes

- To minimize the probability of error, the error patterns most likely to happen should be chosen as coset leaders.
- For BSC, an error pattern of smaller weight is more probable than an error pattern of higher weight.
- Each coset leader should be chosen to be a vector of least weight from the available vectors.
- This way coset leader has minimum weight in its coset.
- The decoding based on standard array is minimum distance decoding (i.e. ML decoding).

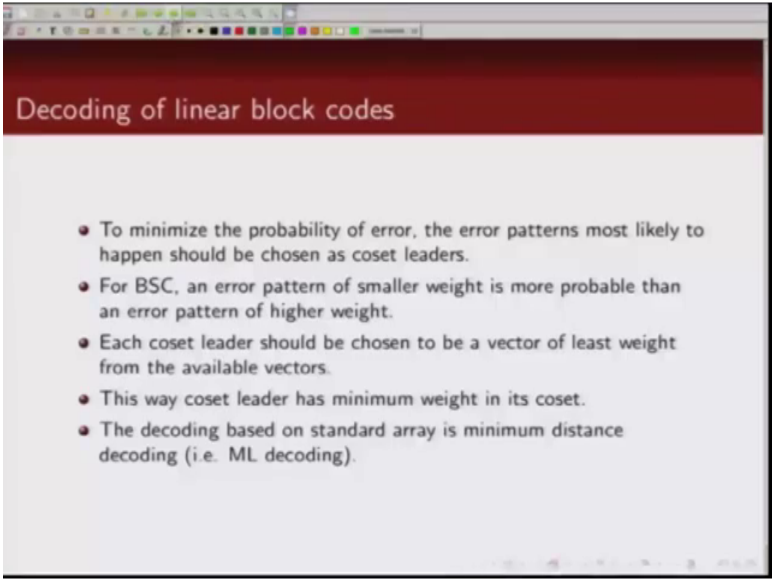
And this decoding is also maximum likelihood decoding because we have shown

(Refer Slide Time 31:22)



earlier that maximum likelihood rule for binary symmetrical channel, it basically chooses v in such a way such that the Hamming difference between r and v is minimized. So in other words, we have to choose an error pattern that has the minimum Hamming weight. So that's also

(Refer Slide Time 31:43)

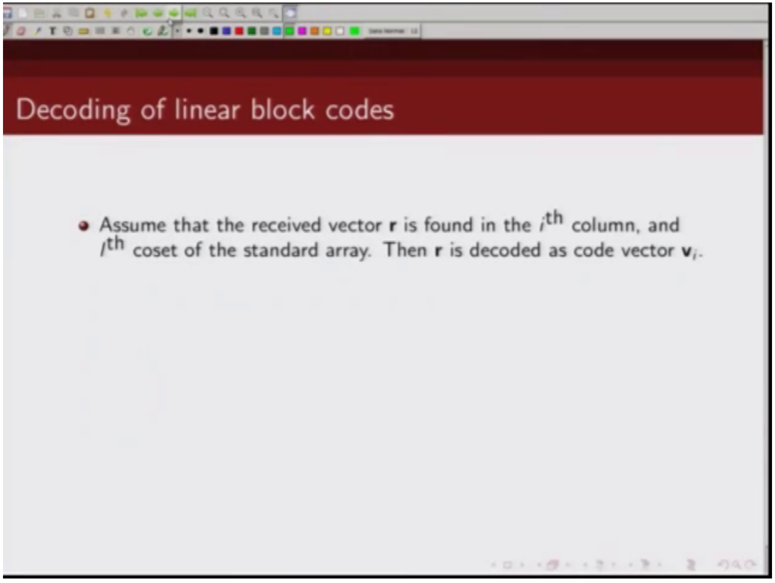


Decoding of linear block codes

- To minimize the probability of error, the error patterns most likely to happen should be chosen as coset leaders.
- For BSC, an error pattern of smaller weight is more probable than an error pattern of higher weight.
- Each coset leader should be chosen to be a vector of least weight from the available vectors.
- This way coset leader has minimum weight in its coset.
- The decoding based on standard array is minimum distance decoding (i.e. ML decoding).

the maximum likelihood decoding.

(Refer Slide Time 31:46)

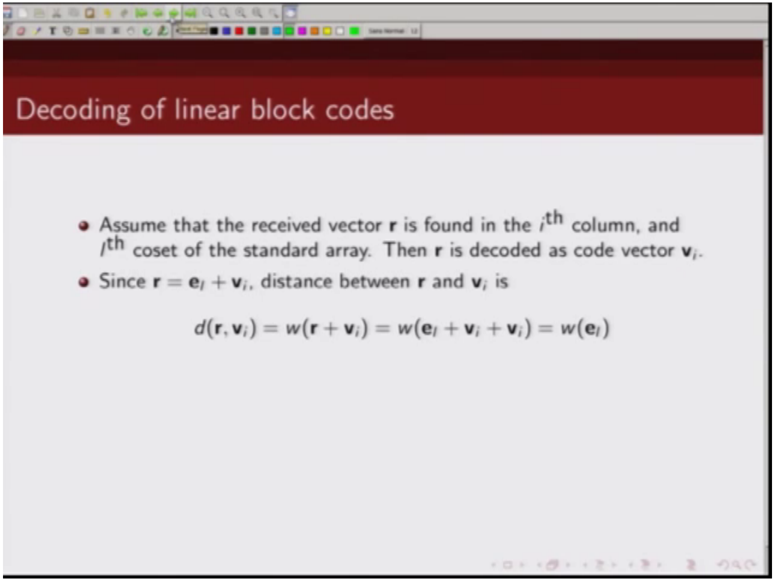


Decoding of linear block codes

- Assume that the received vector **r** is found in the i^{th} column, and l^{th} coset of the standard array. Then **r** is decoded as code vector $\mathbf{v}_i$.

So suppose our received vector is found in the ith column and lth coset of the standard array, so in that case because our received vector r is in the ith column, we are going to decode it as v i.

(Refer Slide Time 32:06)



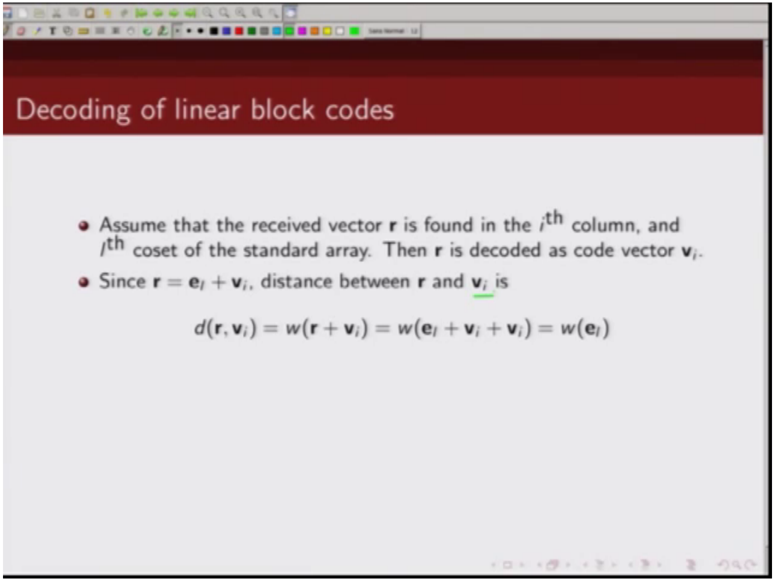
Decoding of linear block codes

- Assume that the received vector $\mathbf{r}$ is found in the i^{th} column, and l^{th} coset of the standard array. Then $\mathbf{r}$ is decoded as code vector $\mathbf{v}_i$.
- Since $\mathbf{r} = \mathbf{e}_l + \mathbf{v}_i$, distance between $\mathbf{r}$ and $\mathbf{v}_i$ is

$$d(\mathbf{r}, \mathbf{v}_i) = w(\mathbf{r} + \mathbf{v}_i) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_i) = w(\mathbf{e}_l)$$

Now our received vector is in ith column and lth coset. So the error pattern is e sub l. So our received sequence is our transmitted codeword plus our error pattern. Now let's try to find out the Hamming distance between our received vector and this code vector v i and Hamming distance between

(Refer Slide Time 32:36)



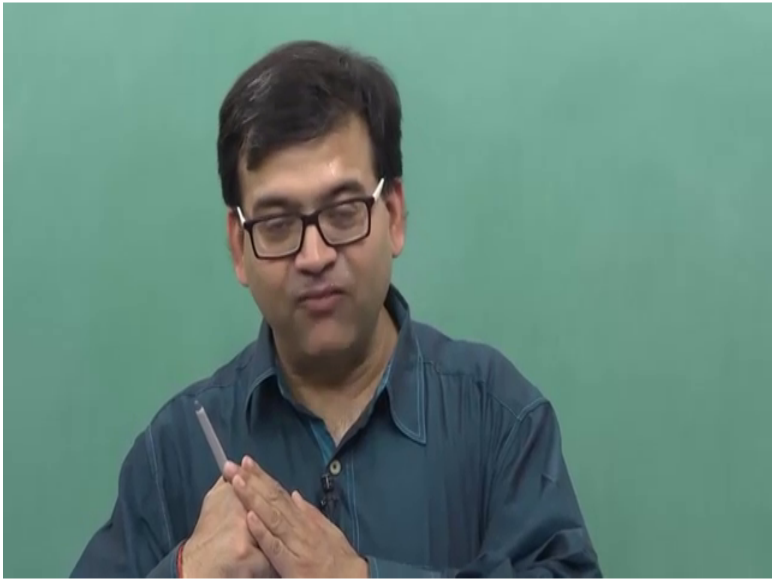
Decoding of linear block codes

- Assume that the received vector $\mathbf{r}$ is found in the i^{th} column, and l^{th} coset of the standard array. Then $\mathbf{r}$ is decoded as code vector $\mathbf{v}_i$.
- Since $\mathbf{r} = \mathbf{e}_l + \mathbf{v}_i$, distance between $\mathbf{r}$ and $\underline{\mathbf{v}_i}$ is

$$d(\mathbf{r}, \mathbf{v}_i) = w(\mathbf{r} + \mathbf{v}_i) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_i) = w(\mathbf{e}_l)$$

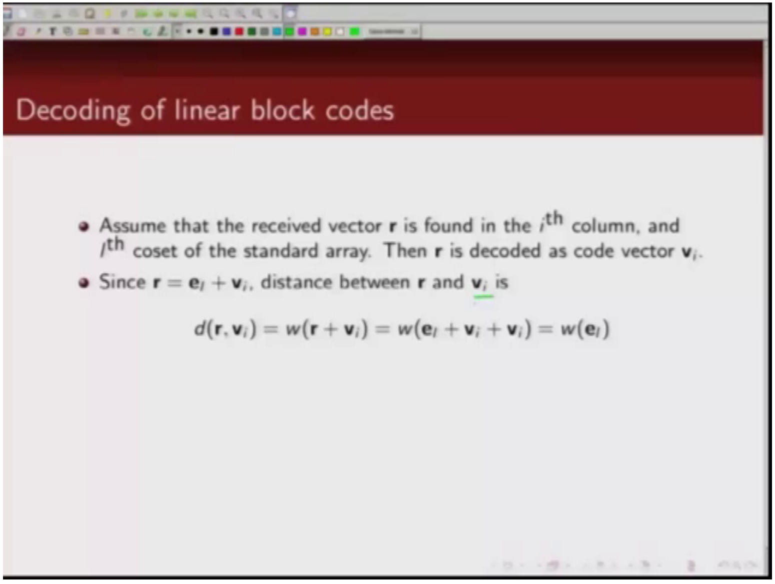
received vector and some other codeword. So when we try Hamming distance between received codeword r and this code vector v i we can see Hamming distance is nothing but number of locations where these 2 bits are differing; so if we add r and v and then

(Refer Slide Time 32:54)



count the number of 1s that would give us the Hamming distance between r and v. So the Hamming distance between r and v

(Refer Slide Time 33:02)



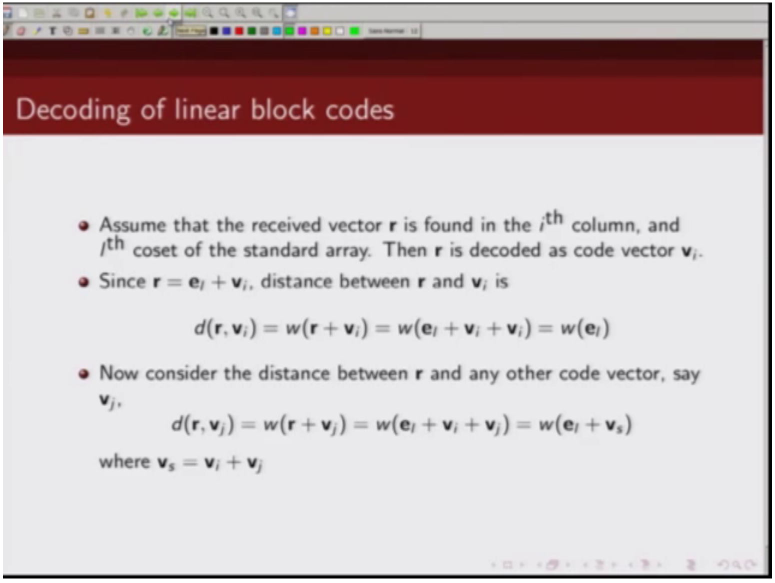
Decoding of linear block codes

- Assume that the received vector $\mathbf{r}$ is found in the i^{th} column, and l^{th} coset of the standard array. Then $\mathbf{r}$ is decoded as code vector $\mathbf{v}_i$.
- Since $\mathbf{r} = \mathbf{e}_l + \mathbf{v}_i$, distance between $\mathbf{r}$ and $\underline{\mathbf{v}_i}$ is

$$d(\mathbf{r}, \mathbf{v}_i) = w(\mathbf{r} + \mathbf{v}_i) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_i) = w(\mathbf{e}_l)$$

is nothing but Hamming weight of the vector r plus v i and what is r? It is e l plus v i, and plus v i so this was nothing but weight of error vector. Next,

(Refer Slide Time 33:22)



Decoding of linear block codes

- Assume that the received vector **r** is found in the i^{th} column, and l^{th} coset of the standard array. Then **r** is decoded as code vector $\mathbf{v}_i$.
- Since $\mathbf{r} = \mathbf{e}_l + \mathbf{v}_i$, distance between **r** and $\mathbf{v}_i$ is

$$d(\mathbf{r}, \mathbf{v}_i) = w(\mathbf{r} + \mathbf{v}_i) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_i) = w(\mathbf{e}_l)$$

- Now consider the distance between **r** and any other code vector, say $\mathbf{v}_j$,

$$d(\mathbf{r}, \mathbf{v}_j) = w(\mathbf{r} + \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_s)$$

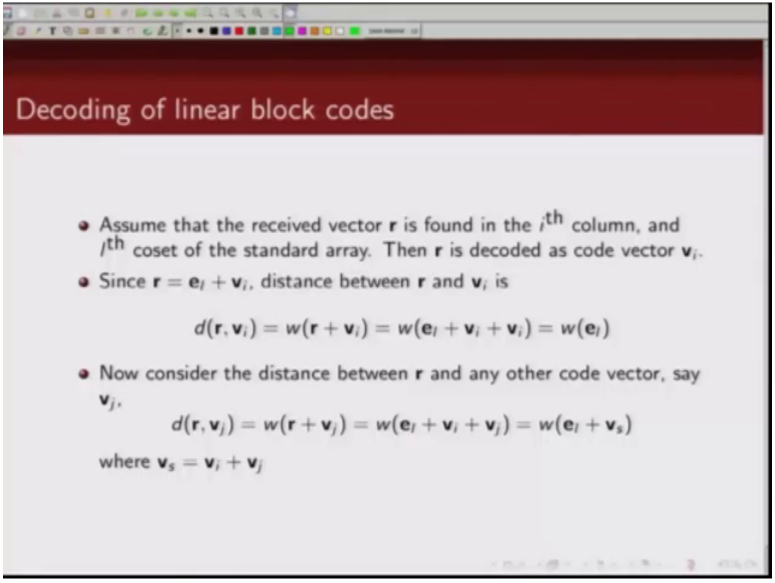
where $\mathbf{v}_s = \mathbf{v}_i + \mathbf{v}_j$

consider now the Hamming distance between r and any other codeword. Let's call it v j. So we are finding Hamming distance between r and any other codeword v j. So that would be given by weight of this vector r plus v j. So r is nothing but e l plus v i plus v j. Now v i plus v j, sum of 2 codewords is another codeword,

(Refer Slide Time 33:53)

so this would be, let's call that codeword v s. So this will be

(Refer Slide Time 33:58)



Decoding of linear block codes

- Assume that the received vector $\mathbf{r}$ is found in the i^{th} column, and l^{th} coset of the standard array. Then $\mathbf{r}$ is decoded as code vector $\mathbf{v}_i$.
- Since $\mathbf{r} = \mathbf{e}_l + \mathbf{v}_i$, distance between $\mathbf{r}$ and $\mathbf{v}_i$ is

$$d(\mathbf{r}, \mathbf{v}_i) = w(\mathbf{r} + \mathbf{v}_i) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_i) = w(\mathbf{e}_l)$$

- Now consider the distance between $\mathbf{r}$ and any other code vector, say $\mathbf{v}_j$,

$$d(\mathbf{r}, \mathbf{v}_j) = w(\mathbf{r} + \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_s)$$

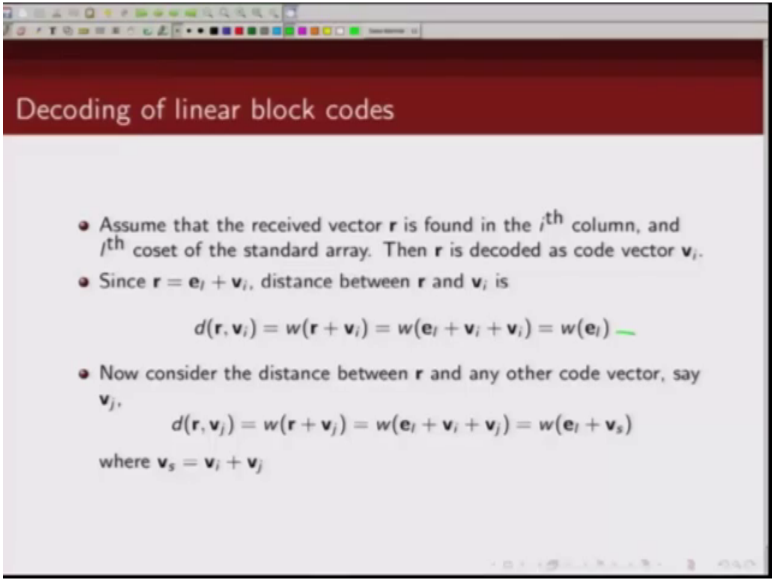
where $\mathbf{v}_s = \mathbf{v}_i + \mathbf{v}_j$

weight of e l plus v s. So what have we done so far? We found that the Hamming distance between the

(Refer Slide Time 34:10)

received codeword and this code vector v i is given by this

(Refer Slide Time 34:14)



Decoding of linear block codes

- Assume that the received vector $\mathbf{r}$ is found in the i^{th} column, and l^{th} coset of the standard array. Then $\mathbf{r}$ is decoded as code vector $\mathbf{v}_i$.
- Since $\mathbf{r} = \mathbf{e}_l + \mathbf{v}_i$, distance between $\mathbf{r}$ and $\mathbf{v}_i$ is

$$d(\mathbf{r}, \mathbf{v}_i) = w(\mathbf{r} + \mathbf{v}_i) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_i) = w(\mathbf{e}_l)$$

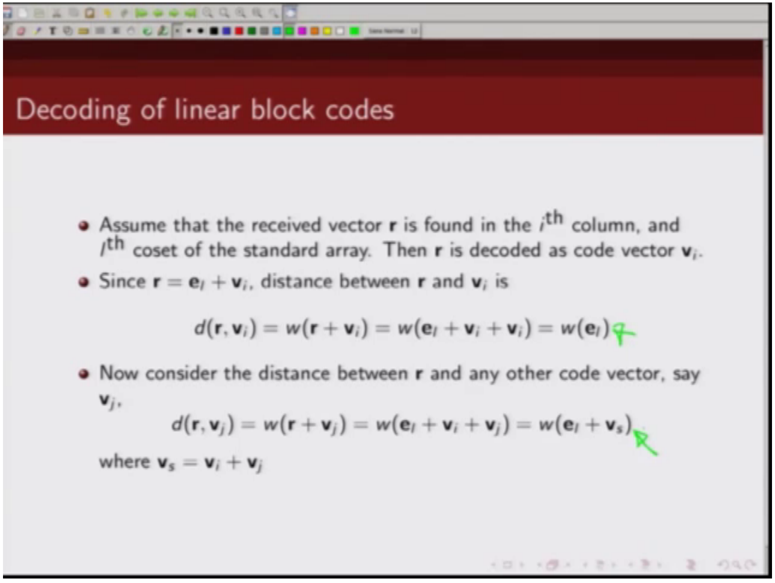
- Now consider the distance between $\mathbf{r}$ and any other code vector, say $\mathbf{v}_j$,

$$d(\mathbf{r}, \mathbf{v}_j) = w(\mathbf{r} + \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_s)$$

where $\mathbf{v}_s = \mathbf{v}_i + \mathbf{v}_j$

and Hamming distance between received codeword and any other codeword

(Refer Slide Time 34:21)



Decoding of linear block codes

- Assume that the received vector $\mathbf{r}$ is found in the i^{th} column, and l^{th} coset of the standard array. Then $\mathbf{r}$ is decoded as code vector $\mathbf{v}_i$.
- Since $\mathbf{r} = \mathbf{e}_l + \mathbf{v}_i$, distance between $\mathbf{r}$ and $\mathbf{v}_i$ is

$$d(\mathbf{r}, \mathbf{v}_i) = w(\mathbf{r} + \mathbf{v}_i) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_i) = w(\mathbf{e}_l)$$

- Now consider the distance between $\mathbf{r}$ and any other code vector, say $\mathbf{v}_j$,

$$d(\mathbf{r}, \mathbf{v}_j) = w(\mathbf{r} + \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_s)$$

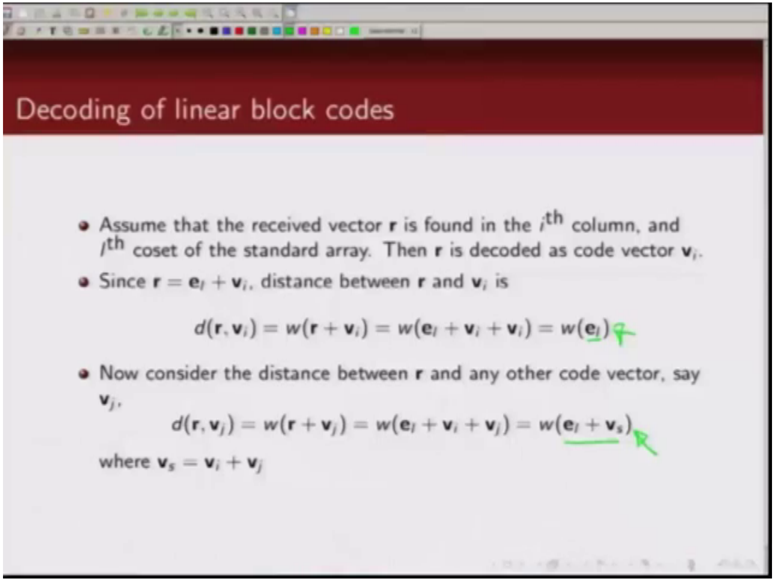
where $\mathbf{v}_s = \mathbf{v}_i + \mathbf{v}_j$

which is not v i is basically given by this. Now e l and e l plus v s are going to be the elements in the same coset,

(Refer Slide Time 34:33)

right and if we choose e to be our coset leader which has minimum number of 1s then

(Refer Slide Time 34:45)



Decoding of linear block codes

- Assume that the received vector $\mathbf{r}$ is found in the i^{th} column, and l^{th} coset of the standard array. Then $\mathbf{r}$ is decoded as code vector $\mathbf{v}_i$.
- Since $\mathbf{r} = \mathbf{e}_l + \mathbf{v}_i$, distance between $\mathbf{r}$ and $\mathbf{v}_i$ is

$$d(\mathbf{r}, \mathbf{v}_i) = w(\mathbf{r} + \mathbf{v}_i) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_i) = w(\underline{\mathbf{e}_l})$$

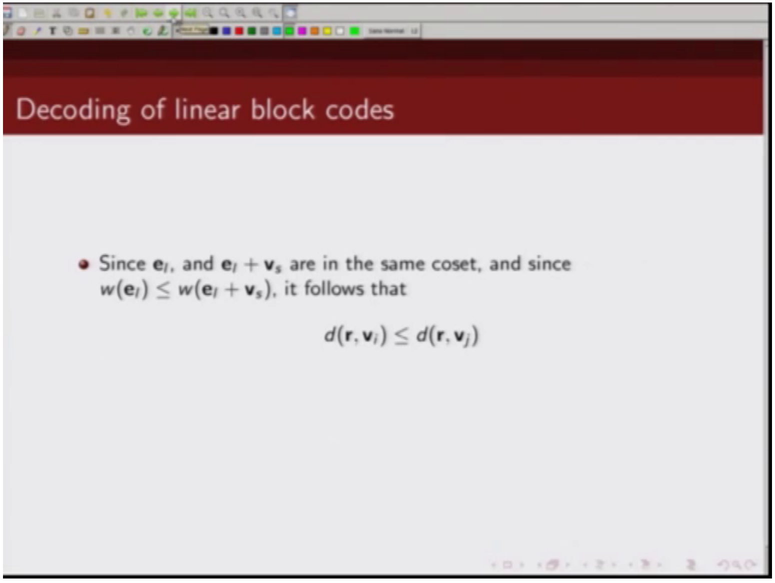
- Now consider the distance between $\mathbf{r}$ and any other code vector, say $\mathbf{v}_j$.

$$d(\mathbf{r}, \mathbf{v}_j) = w(\mathbf{r} + \mathbf{v}_j) = w(\mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j) = w(\underline{\mathbf{e}_l + \mathbf{v}_s})$$

where $\mathbf{v}_s = \mathbf{v}_i + \mathbf{v}_j$

this would be less than this, so we, our minimum distance decoding will decide in favor of v i and not any other codeword v j.

(Refer Slide Time 34:58)



Decoding of linear block codes

- Since $\mathbf{e}_l$, and $\mathbf{e}_l + \mathbf{v}_s$ are in the same coset, and since $w(\mathbf{e}_l) \leq w(\mathbf{e}_l + \mathbf{v}_s)$, it follows that

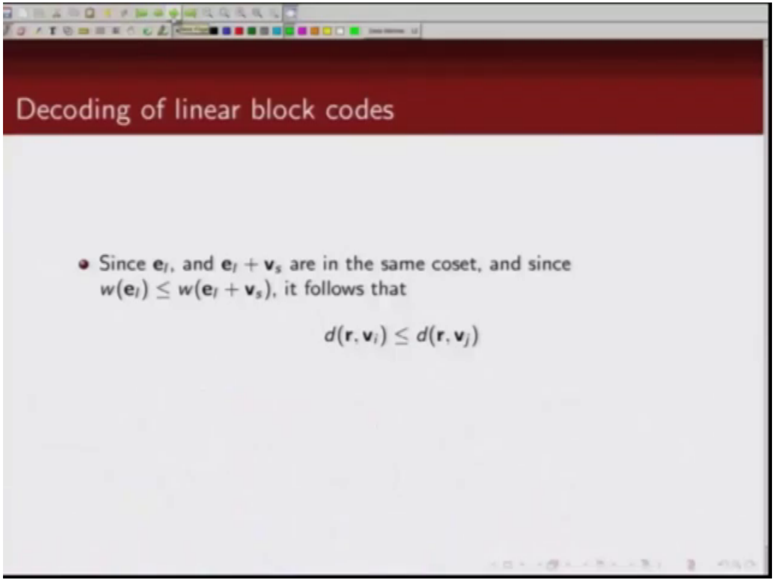
$$d(\mathbf{r}, \mathbf{v}_i) \leq d(\mathbf{r}, \mathbf{v}_j)$$

So as I said since e l and e l plus v s are in the same coset and our coset leader has

(Refer Slide Time 35:06)

minimum Hamming weight, weight of this is less than equal to weight of this then this will

(Refer Slide Time 35:13)



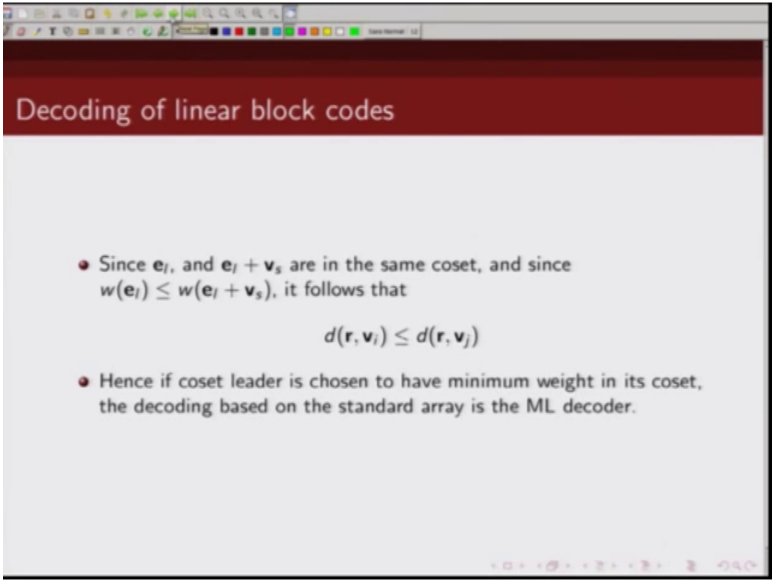
Decoding of linear block codes

- Since $\mathbf{e}_l$, and $\mathbf{e}_l + \mathbf{v}_s$ are in the same coset, and since $w(\mathbf{e}_l) \leq w(\mathbf{e}_l + \mathbf{v}_s)$, it follows that

$$d(\mathbf{r}, \mathbf{v}_i) \leq d(\mathbf{r}, \mathbf{v}_j)$$

always happen. In other words v i will be closer to r than any other codeword v j to r so this will be our correct decoding. So if we choose

(Refer Slide Time 35:28)



Decoding of linear block codes

- Since $\mathbf{e}_l$, and $\mathbf{e}_l + \mathbf{v}_s$ are in the same coset, and since $w(\mathbf{e}_l) \leq w(\mathbf{e}_l + \mathbf{v}_s)$, it follows that

$$d(\mathbf{r}, \mathbf{v}_i) \leq d(\mathbf{r}, \mathbf{v}_j)$$

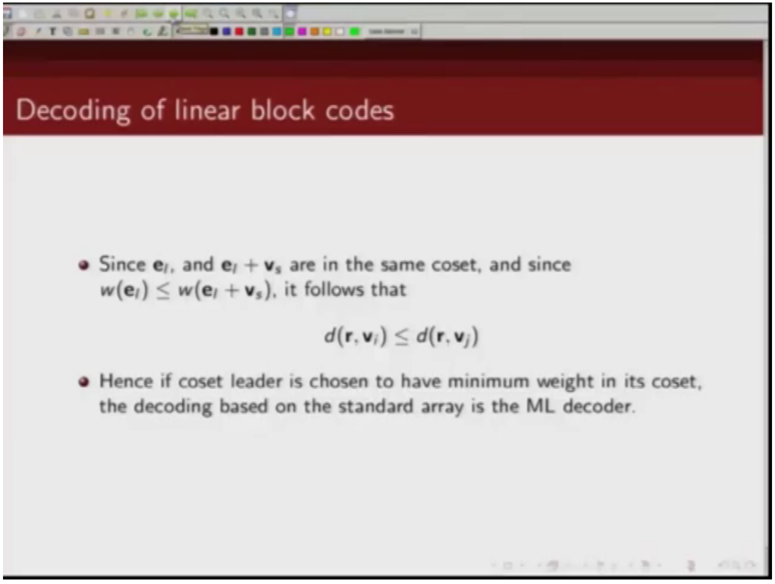
- Hence if coset leader is chosen to have minimum weight in its coset, the decoding based on the standard array is the ML decoder.

our coset leader to be the one which has minimum weight in that coset the decoding basically based on maximum likelihood, decoding will be basically maximum likelihood decoding because maximum likelihood decoder for binary

(Refer Slide Time 35:44)

symmetric channel we have said is the one which minimizes Hamming distance between received codeword and the selected

(Refer Slide Time 35:54)



Decoding of linear block codes

- Since $\mathbf{e}_l$, and $\mathbf{e}_l + \mathbf{v}_s$ are in the same coset, and since $w(\mathbf{e}_l) \leq w(\mathbf{e}_l + \mathbf{v}_s)$, it follows that

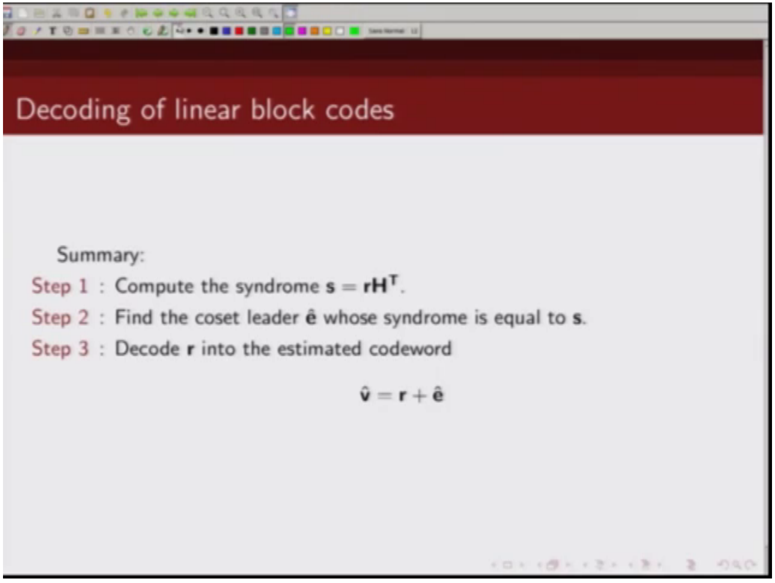
$$d(\mathbf{r}, \mathbf{v}_i) \leq d(\mathbf{r}, \mathbf{v}_j)$$

- Hence if coset leader is chosen to have minimum weight in its coset, the decoding based on the standard array is the ML decoder.

code vector v, Ok.

So now

(Refer Slide Time 36:00)



Decoding of linear block codes

Summary:

Step 1 : Compute the syndrome $\mathbf{s} = \mathbf{r}\mathbf{H}^{\mathbf{T}}$.

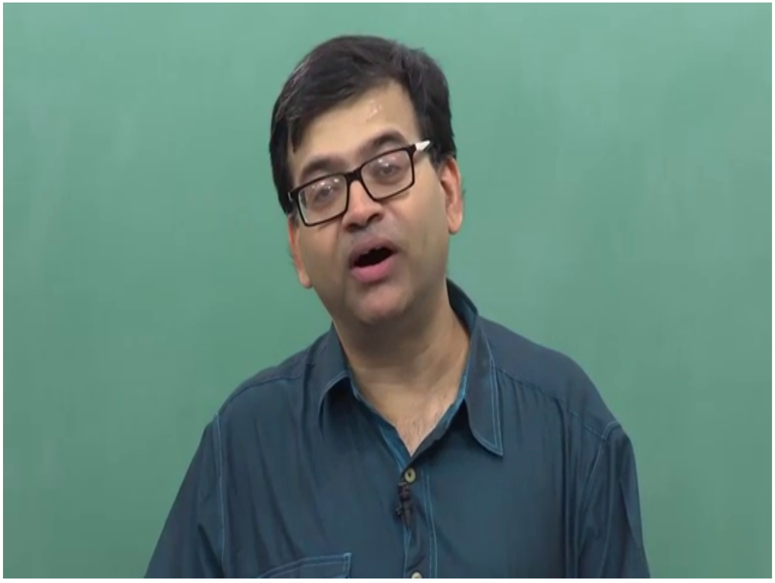
Step 2 : Find the coset leader $\hat{\mathbf{e}}$ whose syndrome is equal to $\mathbf{s}$.

Step 3 : Decode $\mathbf{r}$ into the estimated codeword

$$\hat{\mathbf{v}} = \mathbf{r} + \hat{\mathbf{e}}$$

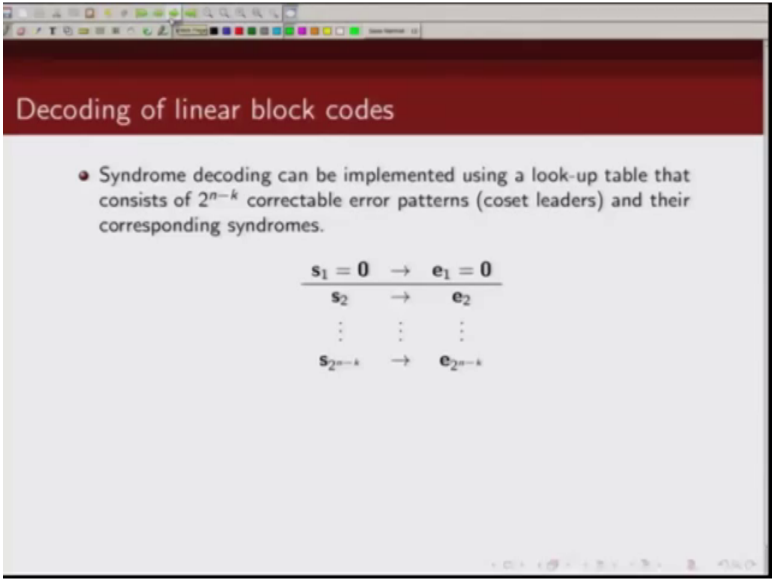
to summarize then, how do we do the decoding? We will first compute the Syndrome, then each of these Syndrome corresponds to one coset leader, so we find out the coset leader corresponding to each Syndrome and once we find the coset leader we add that coset leader which is our likely error pattern to our received sequence and that could be our

(Refer Slide Time 36:27)



estimated codeword.

(Refer Slide Time 36:31)



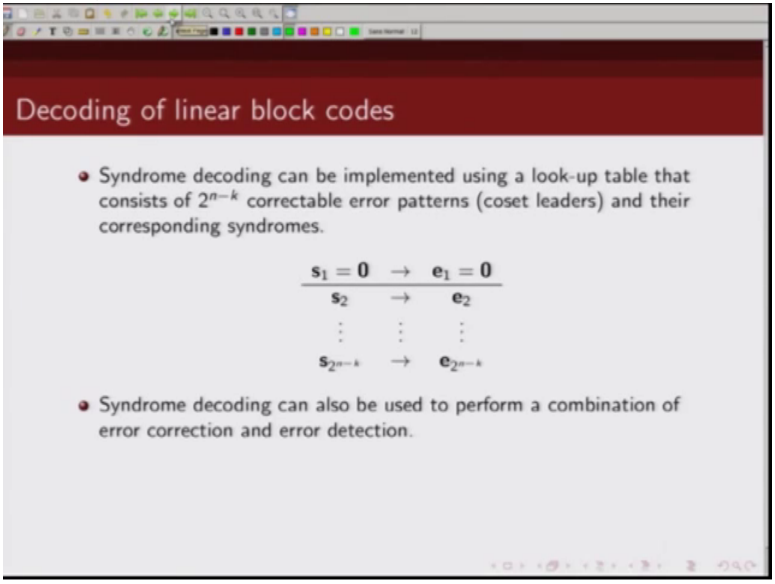
Decoding of linear block codes

- Syndrome decoding can be implemented using a look-up table that consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes.

$$\begin{array}{ccc} \mathbf{s}_1 = \mathbf{0} & \rightarrow & \mathbf{e}_1 = \mathbf{0} \\ \hline \mathbf{s}_2 & \rightarrow & \mathbf{e}_2 \\ \vdots & \vdots & \vdots \\ \mathbf{s}_{2^{n-k}} & \rightarrow & \mathbf{e}_{2^{n-k}} \end{array}$$

And this mapping from Syndrome to error pattern basically can be implemented as a table lookup; so which Syndrome corresponds to which coset leader, this can be uh implemented as a table lookup. Now we

(Refer Slide Time 36:50)



Decoding of linear block codes

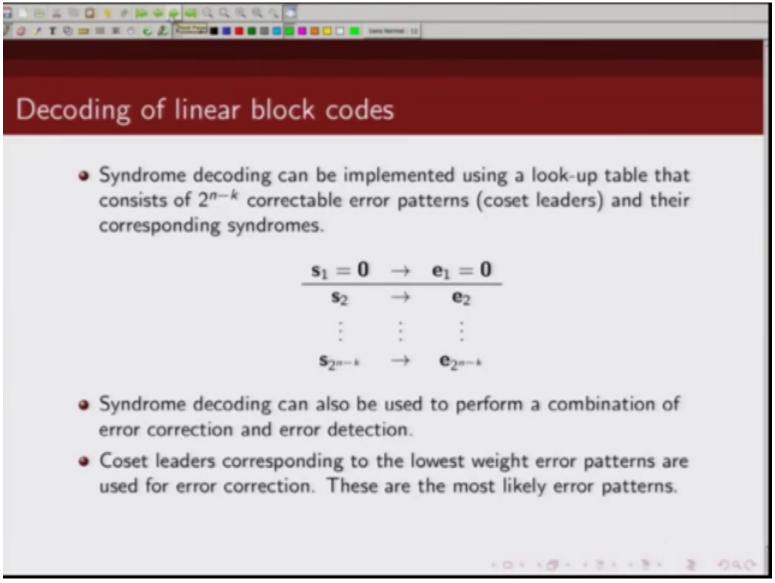
- Syndrome decoding can be implemented using a look-up table that consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes.

$$\begin{array}{ccc} \mathbf{s}_1 = \mathbf{0} & \rightarrow & \mathbf{e}_1 = \mathbf{0} \\ \hline \mathbf{s}_2 & \rightarrow & \mathbf{e}_2 \\ \vdots & \vdots & \vdots \\ \mathbf{s}_{2^{n-k}} & \rightarrow & \mathbf{e}_{2^{n-k}} \end{array}$$

- Syndrome decoding can also be used to perform a combination of error correction and error detection.

can use this Syndrome decoding for both error correction and error detection and we will give an example to illustrate this.

(Refer Slide Time 36:58)



Decoding of linear block codes

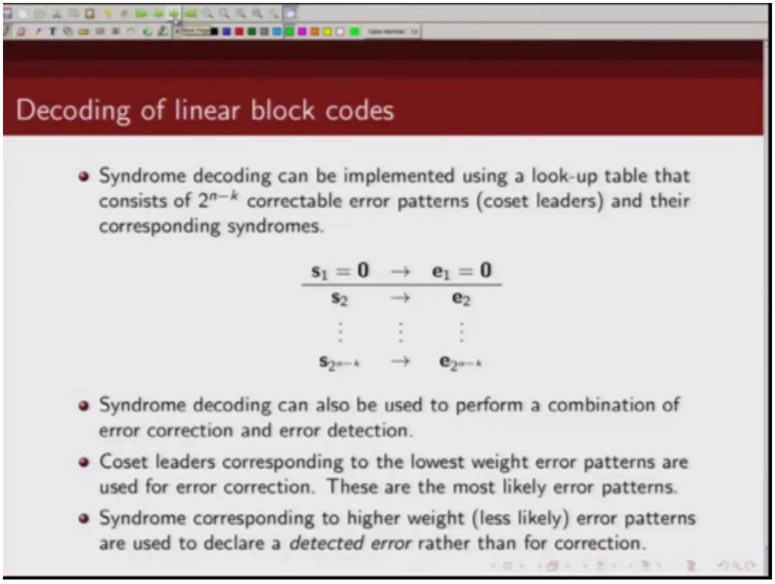
- Syndrome decoding can be implemented using a look-up table that consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes.

$$\begin{array}{ccc} \mathbf{s}_1 = \mathbf{0} & \rightarrow & \mathbf{e}_1 = \mathbf{0} \\ \hline \mathbf{s}_2 & \rightarrow & \mathbf{e}_2 \\ \vdots & \vdots & \vdots \\ \mathbf{s}_{2^{n-k}} & \rightarrow & \mathbf{e}_{2^{n-k}} \end{array}$$

- Syndrome decoding can also be used to perform a combination of error correction and error detection.
- Coset leaders corresponding to the lowest weight error patterns are used for error correction. These are the most likely error patterns.

Again basically as we said the coset leader corresponding to lowest weight error patterns are essentially used for error corrections and these are most likely error patterns according to the maximum likelihood rule.

(Refer Slide Time 37:13)



Decoding of linear block codes

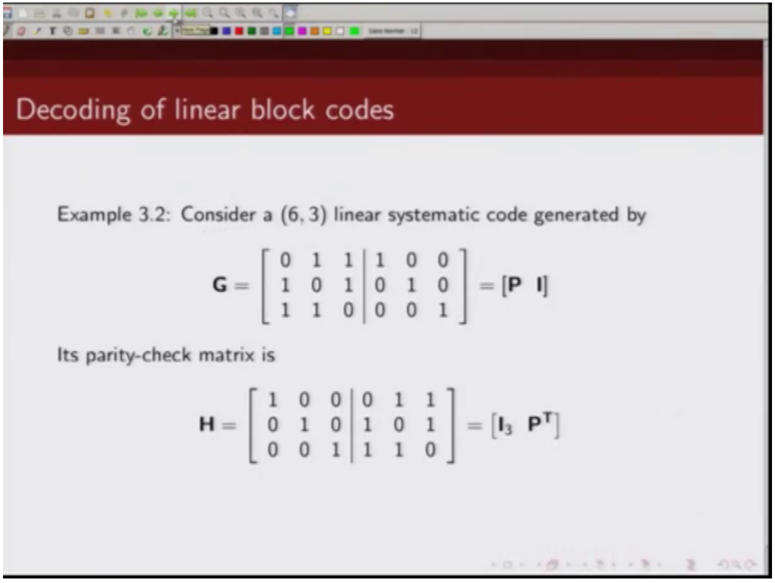
- Syndrome decoding can be implemented using a look-up table that consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes.

$$\begin{array}{ccc} \mathbf{s}_1 = \mathbf{0} & \rightarrow & \mathbf{e}_1 = \mathbf{0} \\ \hline \mathbf{s}_2 & \rightarrow & \mathbf{e}_2 \\ \vdots & \vdots & \vdots \\ \mathbf{s}_{2^{n-k}} & \rightarrow & \mathbf{e}_{2^{n-k}} \end{array}$$

- Syndrome decoding can also be used to perform a combination of error correction and error detection.
- Coset leaders corresponding to the lowest weight error patterns are used for error correction. These are the most likely error patterns.
- Syndrome corresponding to higher weight (less likely) error patterns are used to declare a *detected error* rather than for correction.

Now we could use this standard array for both, a combination of both error correction and error detection and we could, we are going to illustrate this point that we could use Syndrome corresponding to higher weight uh error pattern, error detection rather than correction.

(Refer Slide Time 37:33)



Decoding of linear block codes

Example 3.2: Consider a (6, 3) linear systematic code generated by

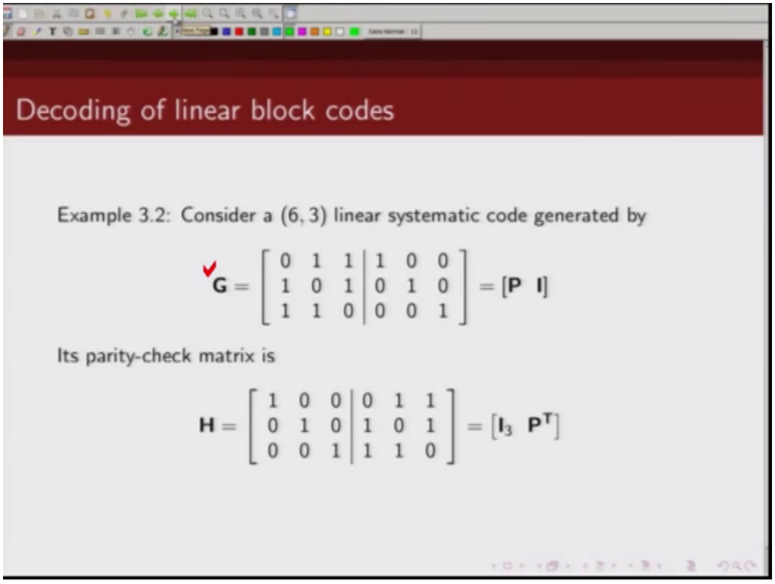
$$\mathbf{G} = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] = [\mathbf{P} \ \ \mathbf{I}]$$

Its parity-check matrix is

$$\mathbf{H} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right] = [\mathbf{I}_3 \ \ \mathbf{P}^T]$$

So let us take example to illustrate how we can use this standard array for error correction and error detection. So this is our 6 3 systematic linear binary code whose generator matrix is given by this and parity check matrix is given by

(Refer Slide Time 37:56)



Decoding of linear block codes

Example 3.2: Consider a (6, 3) linear systematic code generated by

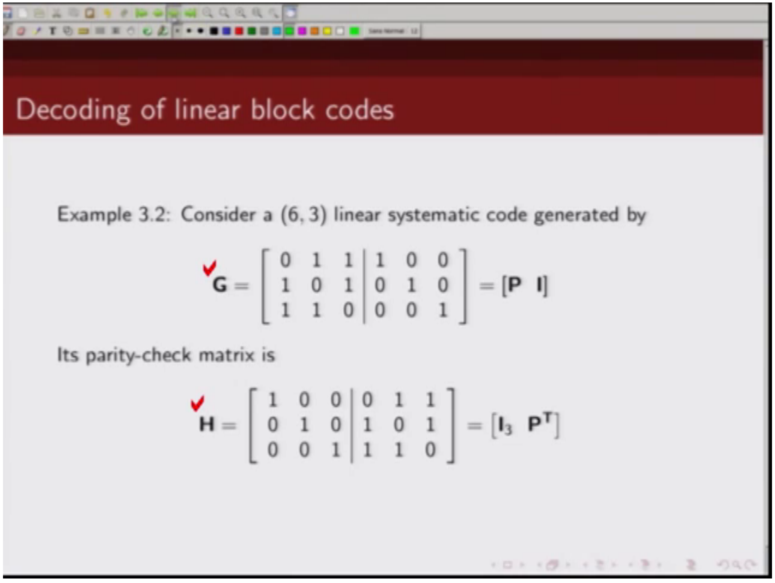
$$✓\mathbf{G} = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] = [\mathbf{P} \ \ \mathbf{I}]$$

Its parity-check matrix is

$$\mathbf{H} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right] = [\mathbf{I}_3 \ \ \mathbf{P}^T]$$

this.

(Refer Slide Time 37:58)



Decoding of linear block codes

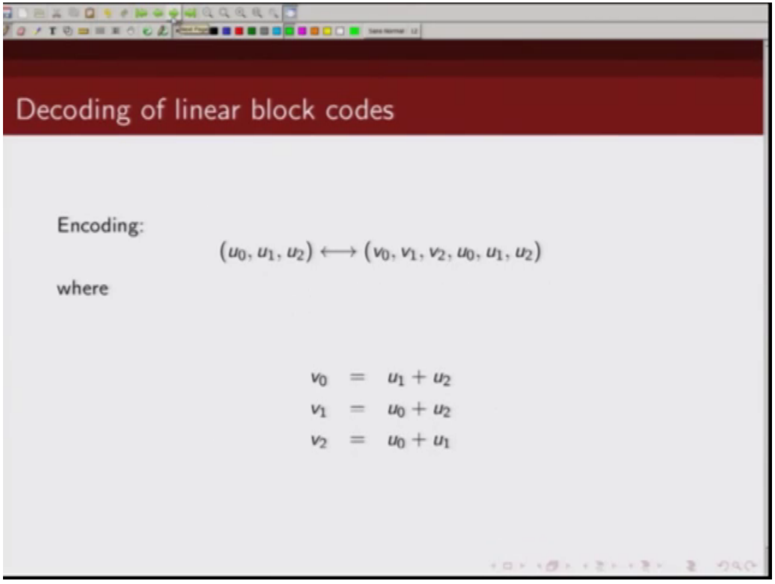
Example 3.2: Consider a (6, 3) linear systematic code generated by

$$\mathbf{G} = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] = [\mathbf{P} \;\; \mathbf{I}]$$

Its parity-check matrix is

$$\mathbf{H} = \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{array}\right] = [\mathbf{I}_3 \;\; \mathbf{P}^T]$$

(Refer Slide Time 37:59)



Decoding of linear block codes

Encoding:

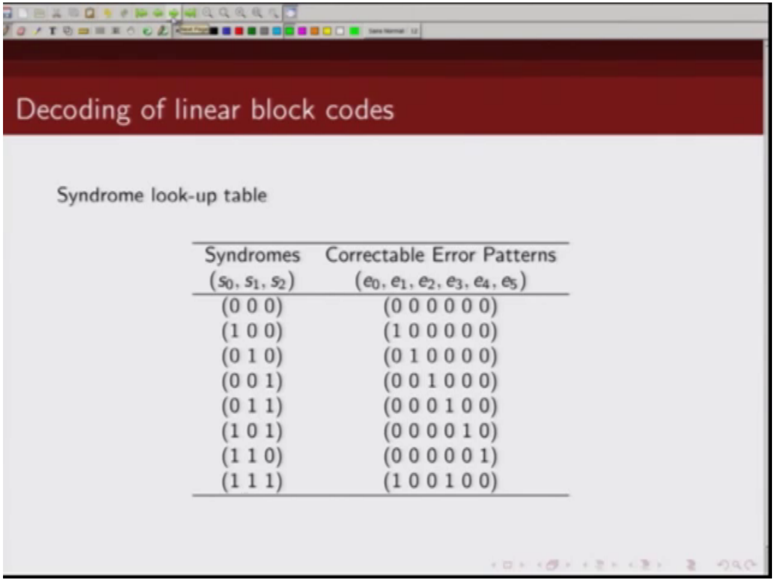
$$(u_0, u_1, u_2) \longleftrightarrow (v_0, v_1, v_2, u_0, u_1, u_2)$$

where

$$\begin{aligned} v_0 &= u_1 + u_2 \\ v_1 &= u_0 + u_2 \\ v_2 &= u_0 + u_1 \end{aligned}$$

These are encoding equations

(Refer Slide Time 38:03)



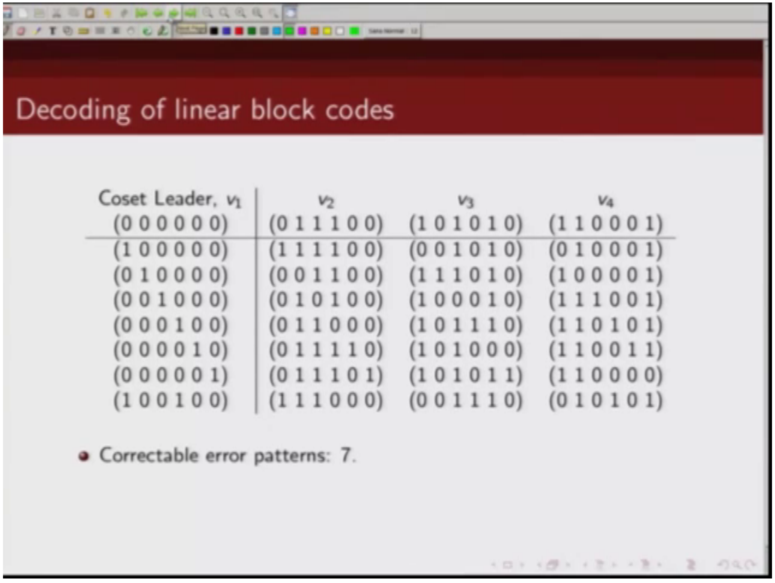
Decoding of linear block codes

Syndrome look-up table

Syndromes (s_0, s_1, s_2)	Correctable Error Patterns $(e_0, e_1, e_2, e_3, e_4, e_5)$
(0 0 0)	(0 0 0 0 0 0)
(1 0 0)	(1 0 0 0 0 0)
(0 1 0)	(0 1 0 0 0 0)
(0 0 1)	(0 0 1 0 0 0)
(0 1 1)	(0 0 0 1 0 0)
(1 0 1)	(0 0 0 0 1 0)
(1 1 0)	(0 0 0 0 0 1)
(1 1 1)	(1 0 0 1 0 0)

and this is the mapping of the Syndrome to the coset leader. So the first step involved

(Refer Slide Time 38:10)



Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

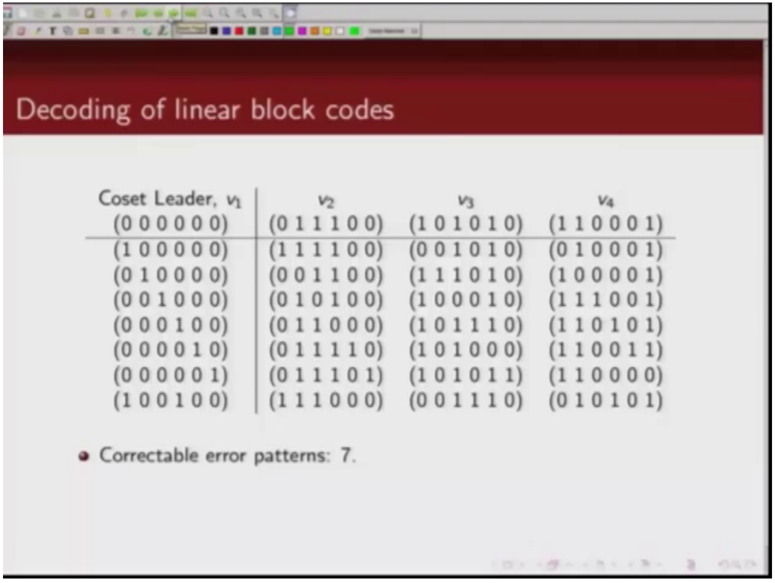
- Correctable error patterns: 7.

in this is creation of standard array. And I believe, now you know

(Refer Slide Time 38:18)

how to create a standard array, the first step is first row of

(Refer Slide Time 38:24)



Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

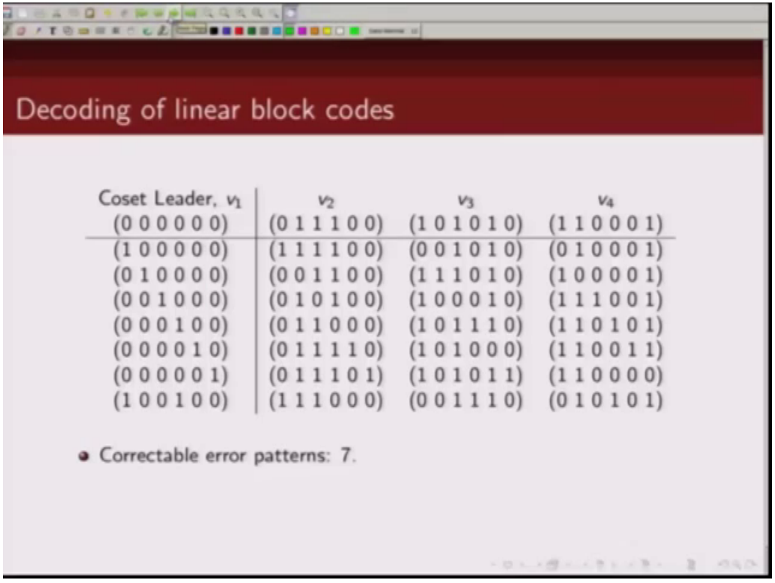
- Correctable error patterns: 7.

the standard array will be set of codewords. You are already given the generator matrix so you can generate what are set of codewords. The leftmost entry in the first row which is the row of codewords should be

(Refer Slide Time 38:39)

all zero codewords and then you can place

(Refer Slide Time 38:42)



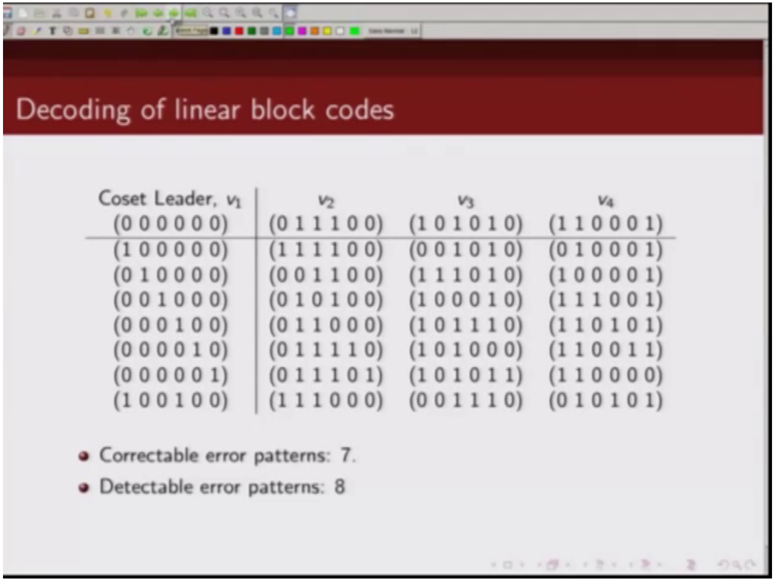
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.

the other codewords in any order. Now I couldn't fit in all the columns in one slide so I have v 1 to v 4 in this slide, and next I have

(Refer Slide Time 38:53)



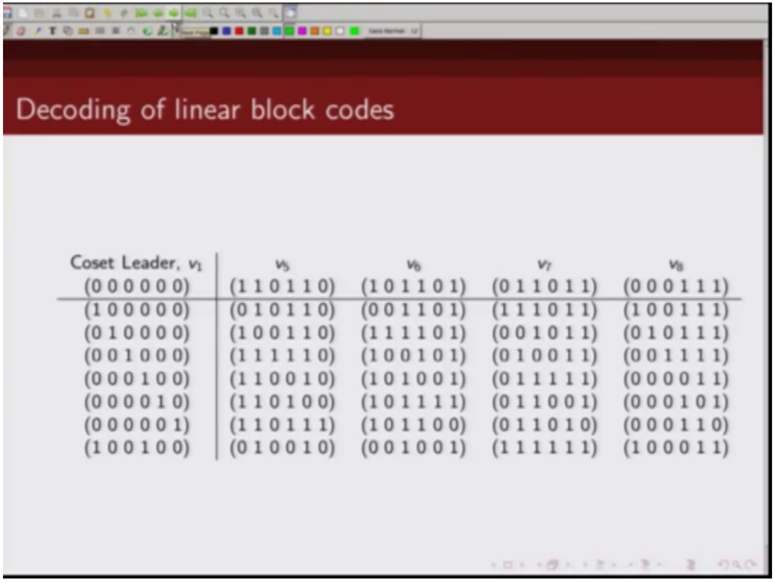
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8

v 5 to

(Refer Slide Time 38:55)



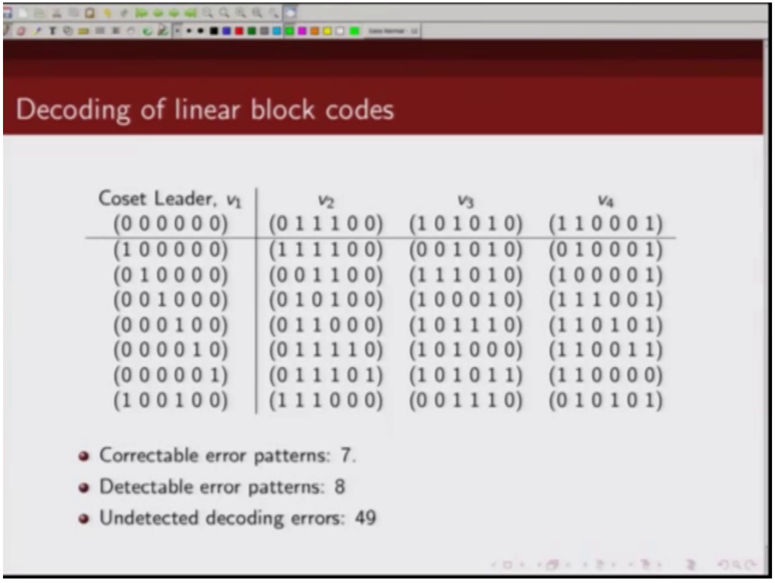
Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

v 8 in the next slide, Ok.

Now

(Refer Slide Time 39:01)



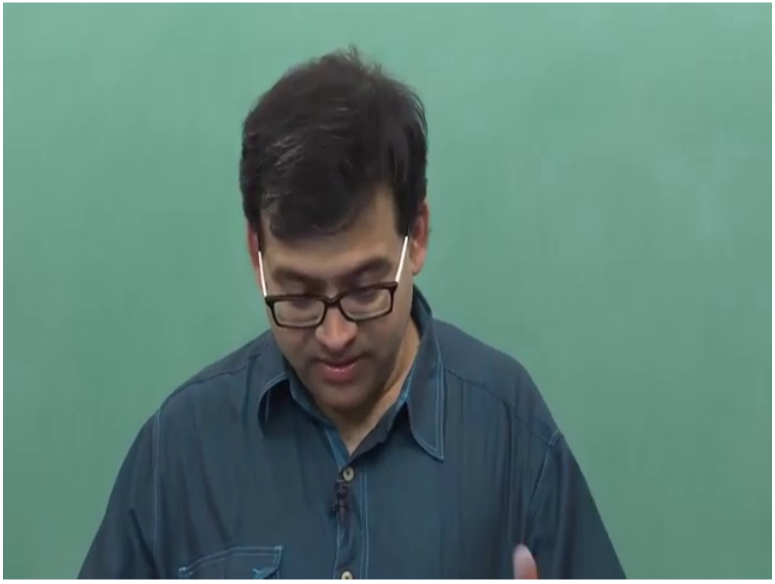
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

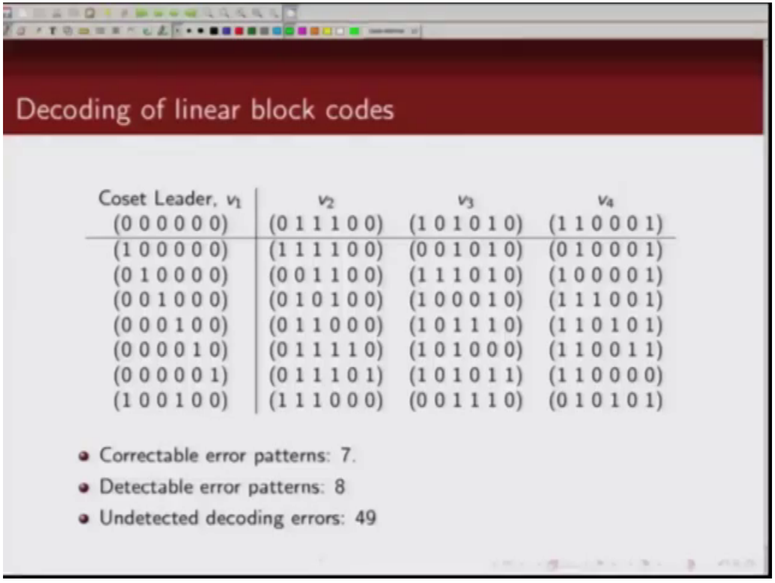
let me explain what I mean by correctable error patterns, detectable error patterns and undetected decoding error. So, out of these

(Refer Slide Time 39:16)



all possible error patterns, which are the error patterns that are correctly decodable? Now if you choose your coset leader to be the one which has the least Hamming weight, which has the least number of 1s

(Refer Slide Time 39:38)



Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

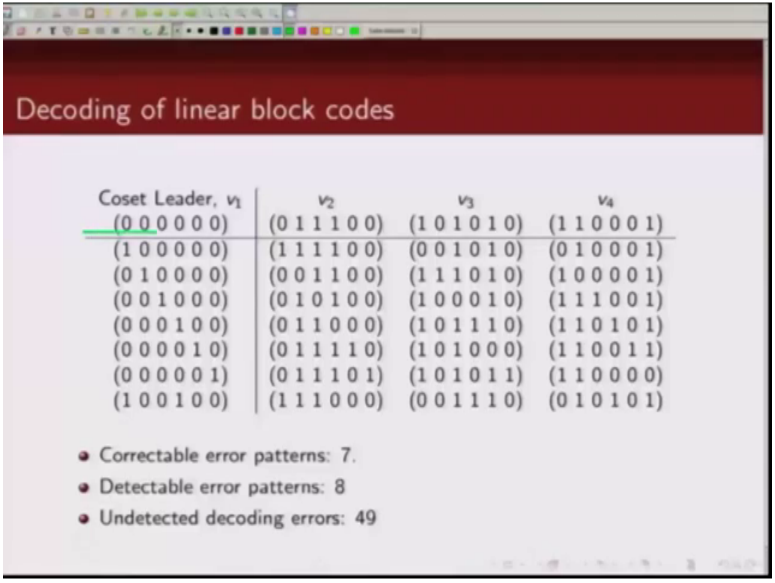
- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

and if you make that error pattern as your coset leader, then you can correctly decode those

(Refer Slide Time 39:47)

those error patterns. So let's look at each of these rows. Of course

(Refer Slide Time 39:53)



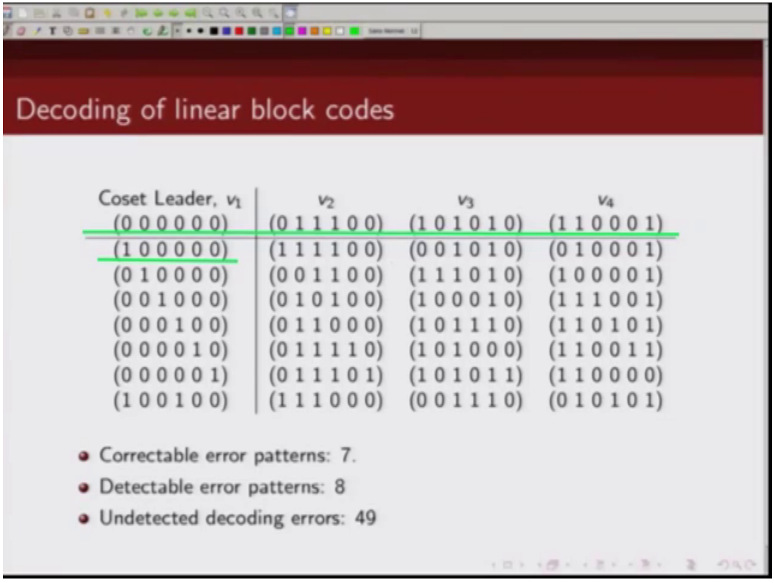
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

this corresponds to all correct codewords. If the error patterns is this,

(Refer Slide Time 39:59)



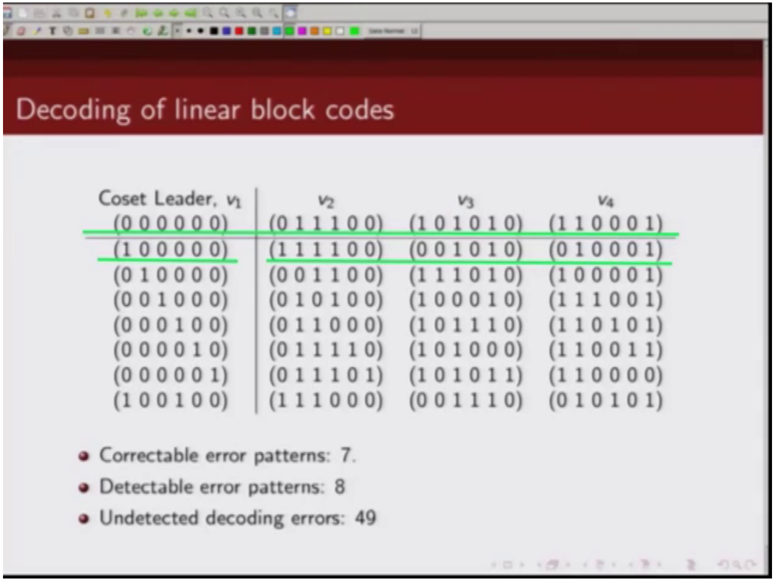
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

these are the set of

(Refer Slide Time 40:01)



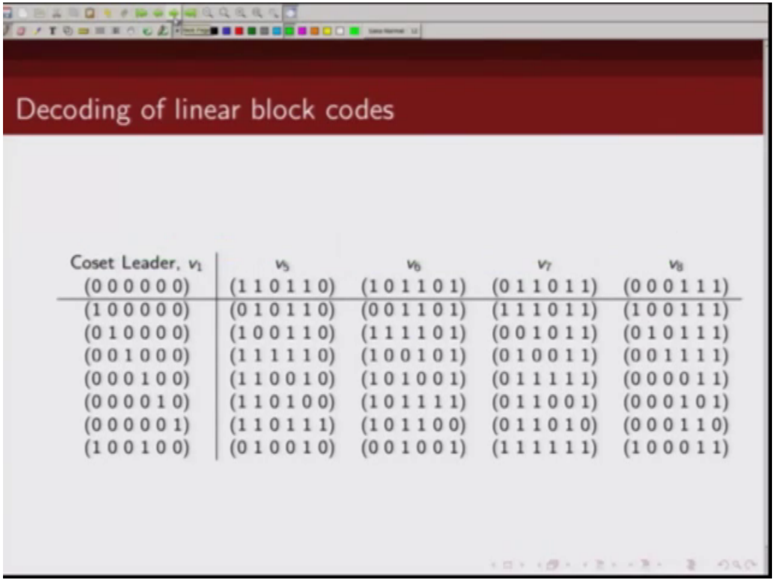
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

other elements of the coset, and you can

(Refer Slide Time 40:04)

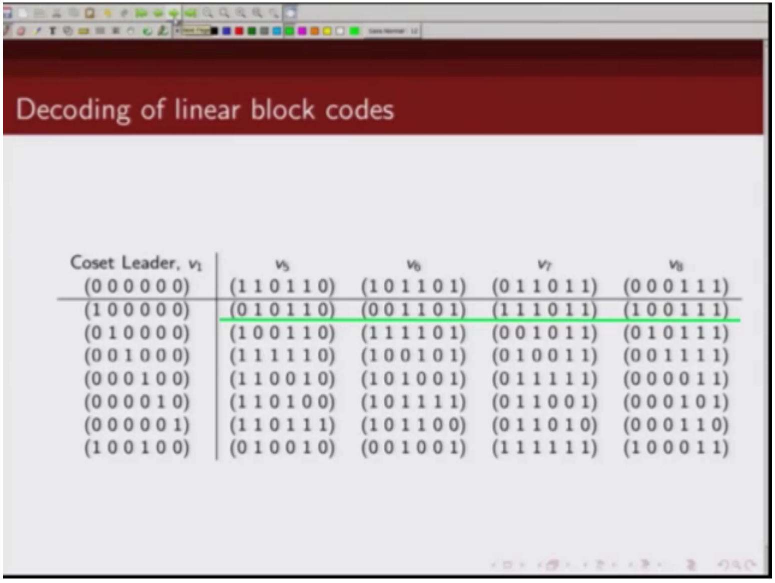


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

see here, none of

(Refer Slide Time 40:06)

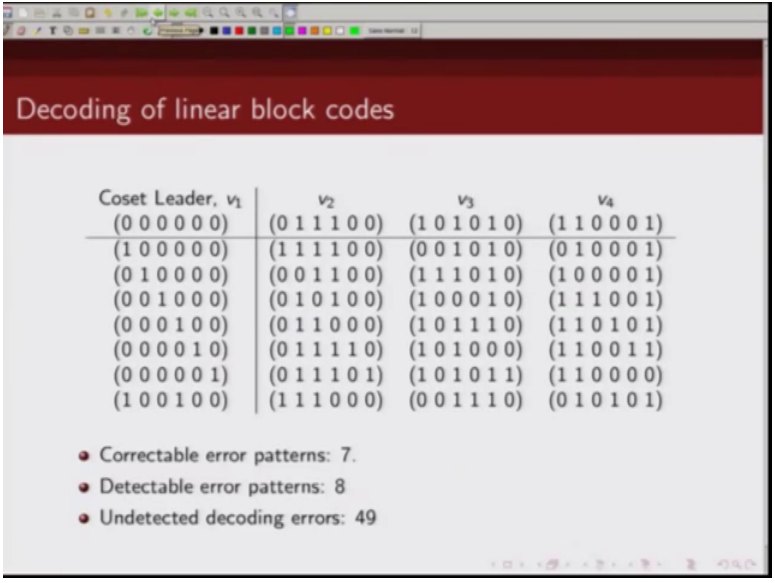


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

these elements have weight less than 2, they are all 3, 3, 5, 4 and this has

(Refer Slide Time 40:15)



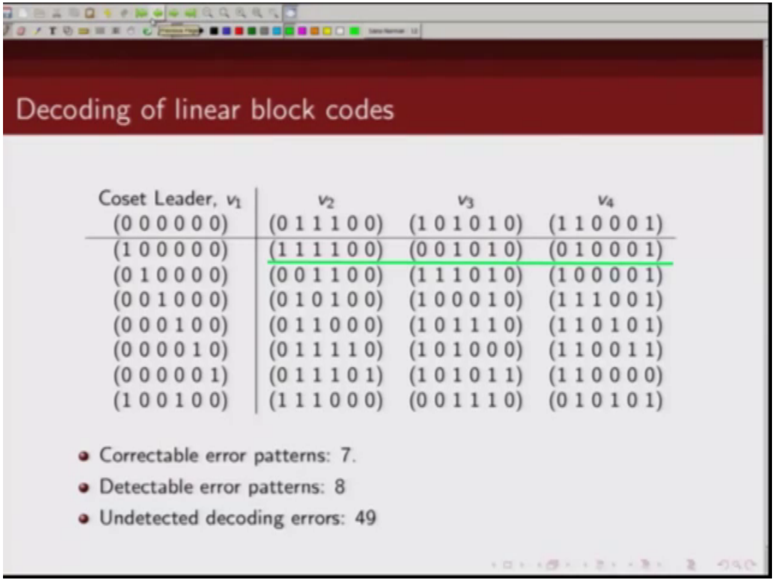
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

weight 4,

(Refer Slide Time 40:17)



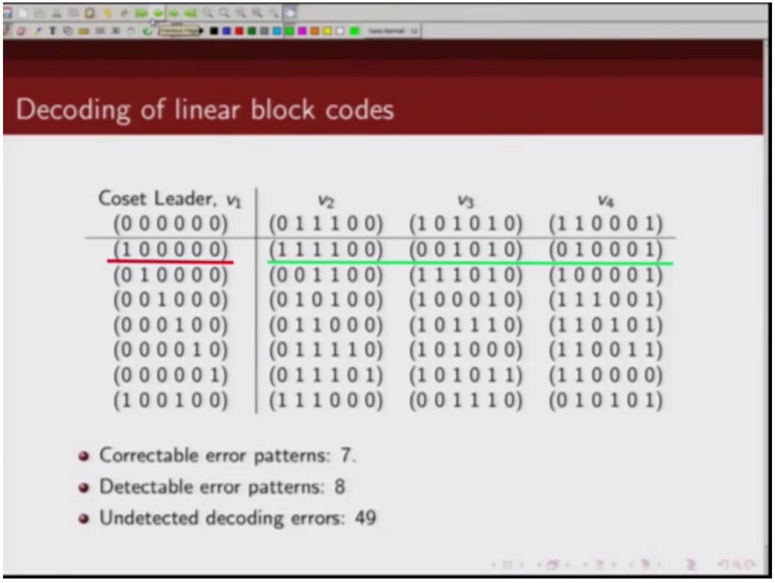
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

2, 2 so this is the

(Refer Slide Time 40:21)



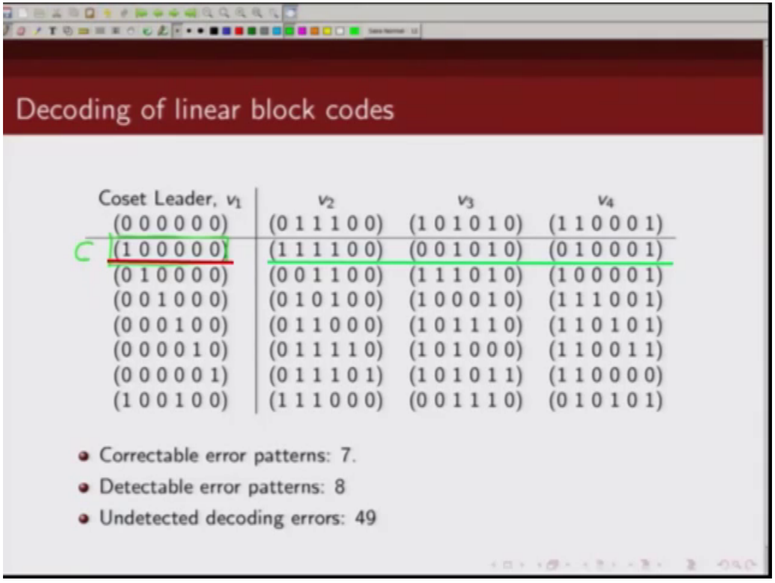
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

the minimum weight error pattern and this is Hamming weight 1. So this error pattern is correctable.

(Refer Slide Time 40:32)



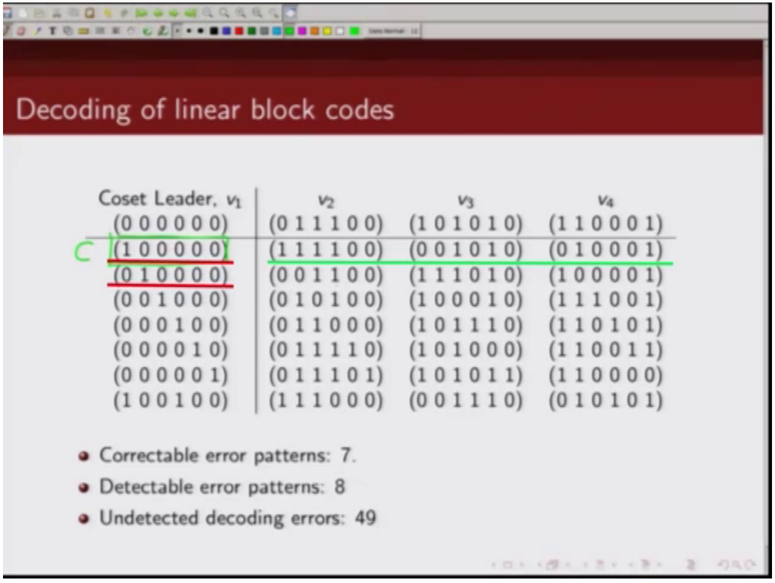
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

And if this error happens this can be corrected. What about this? Just look at other elements.

(Refer Slide Time 40:42)



Decoding of linear block codes

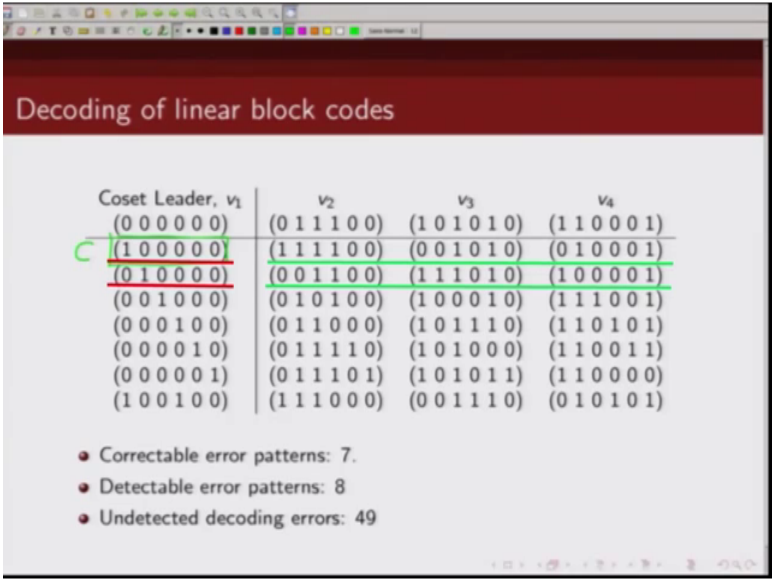
C

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

This has two 1s, four 1s, two 1s,

(Refer Slide Time 40:45)

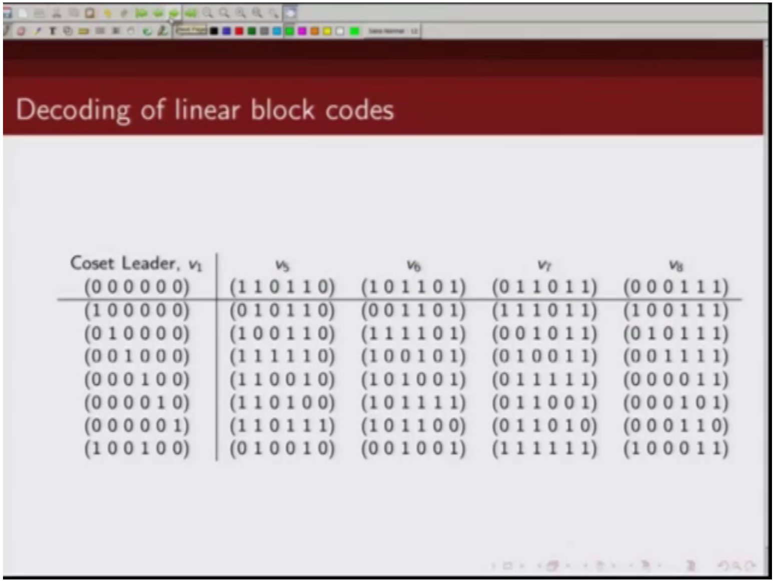


Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C (1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

(Refer Slide Time 40:47)

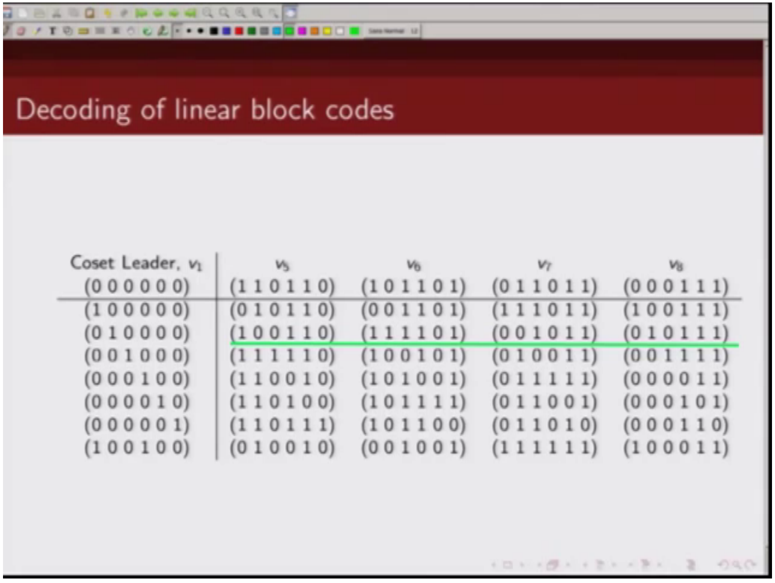


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

three 1s, five 1s, three 1s,

(Refer Slide Time 40:51)

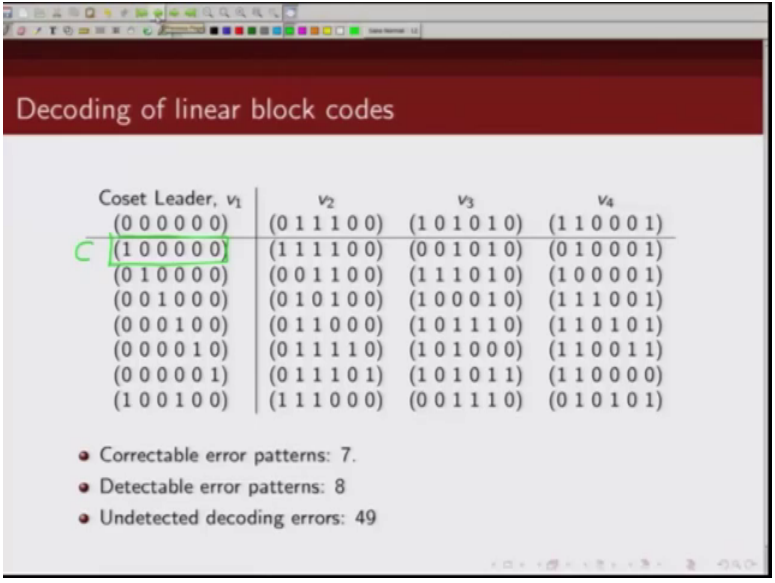


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

four 1s. So clearly this has only one 1, rest

(Refer Slide Time 40:57)



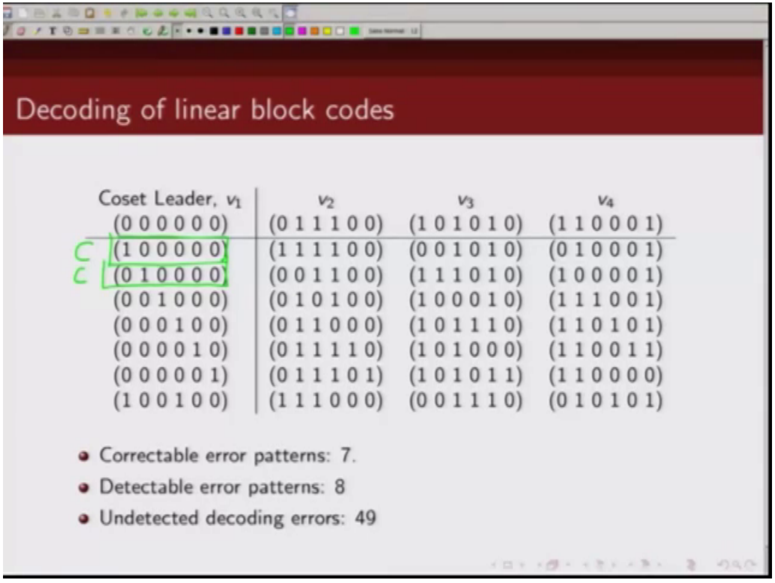
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

others all have weight 2 or more. So this is the error pattern and if we make this as coset leader this is also correctable.

(Refer Slide Time 41:06)



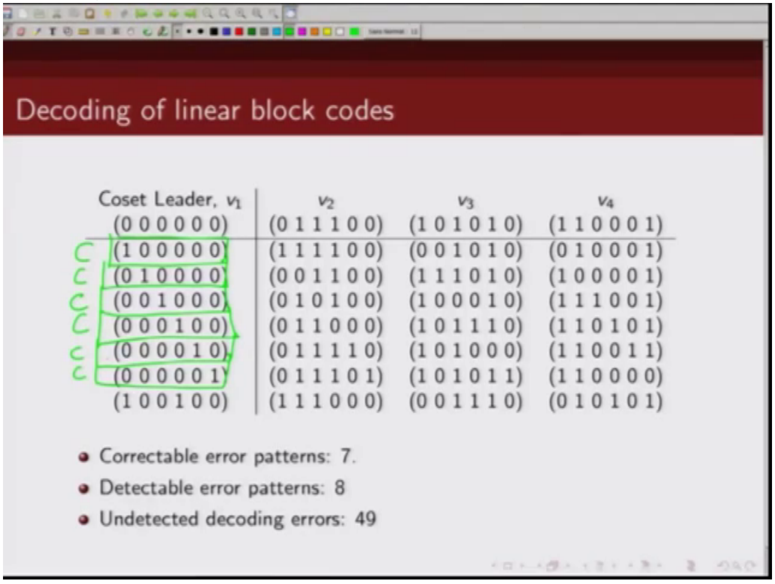
Decoding of linear block codes

	Coset Leader, v_1	v_2	v_3	v_4
	(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
	(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
	(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
	(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
	(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
	(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

Similarly we can see from other rows also. This single error pattern is correctable, this single error pattern is correctable, this is correctable, this is correctable. These are all correctable patterns. So when

(Refer Slide Time 41:21)



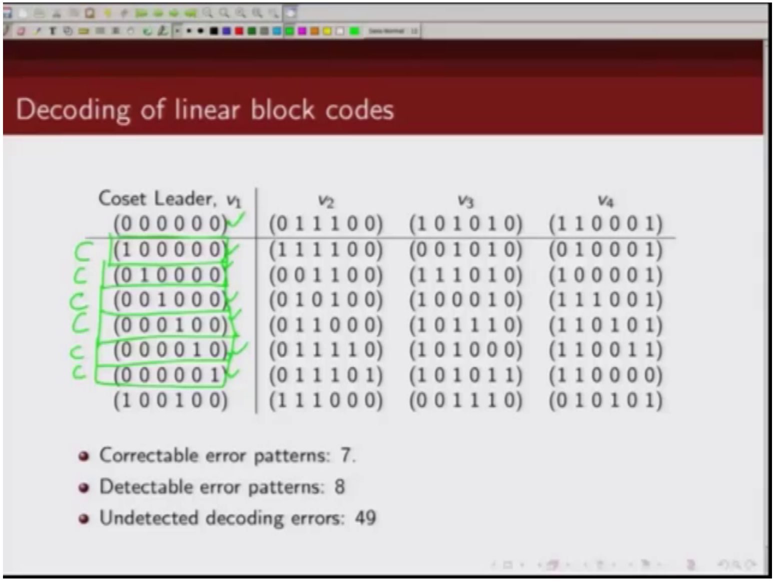
Decoding of linear block codes

	Coset Leader, v_1	v_2	v_3	v_4
	(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C	(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C	(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
	(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

I talk about correctable patterns essentially I mean, if you get this, this, this, this, this, this of course this is no error case

(Refer Slide Time 41:28)



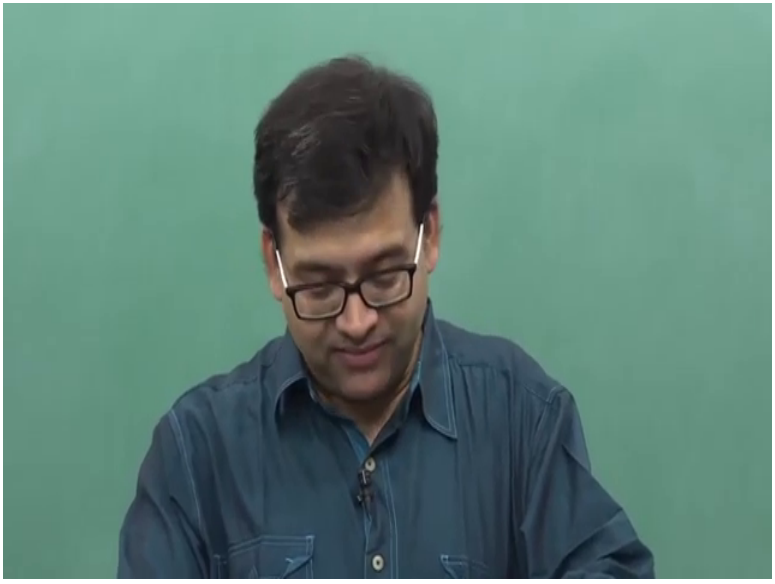
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
(0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C (1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C (0 1 0 0 0 0) ✓	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C (0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C (0 0 0 1 0 0) ✓	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C (0 0 0 0 1 0) ✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C (0 0 0 0 0 1) ✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7.
- Detectable error patterns: 8
- Undetected decoding errors: 49

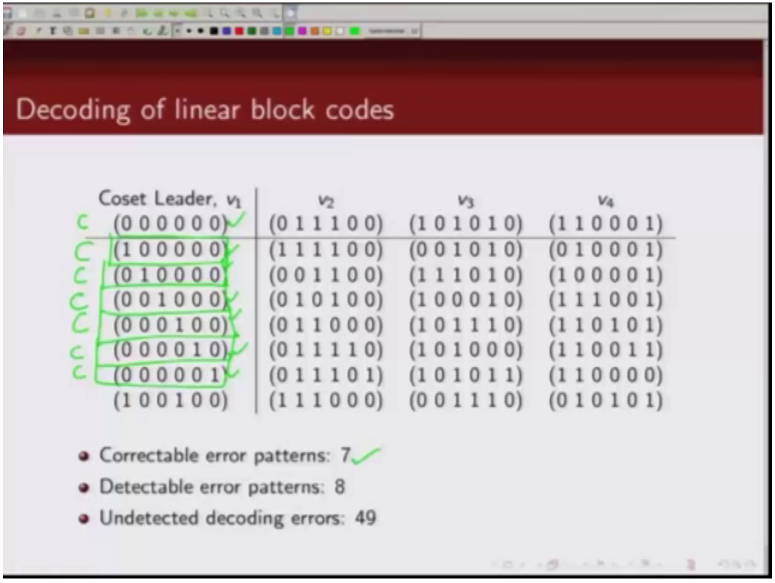
which also I am counting in correctable patterns, so these 7 patterns, if these are the error patterns

(Refer Slide Time 41:36)



then it is correctable. So these are the 7 correctable

(Refer Slide Time 41:40)



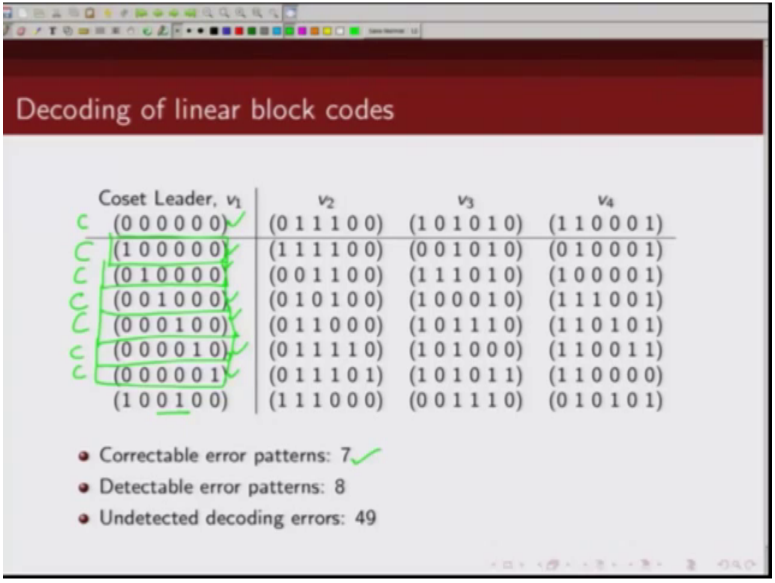
Decoding of linear block codes

	Coset Leader, v_1	v_2	v_3	v_4
C	(0 0 0 0 0 0)✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0)✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0)✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0)✓	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C	(0 0 0 0 1 0)✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C	(0 0 0 0 0 1)✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
	(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7✓
- Detectable error patterns: 8
- Undetected decoding errors: 49

patterns. Now what happens here? This error pattern has weight 2.

(Refer Slide Time 41:48)



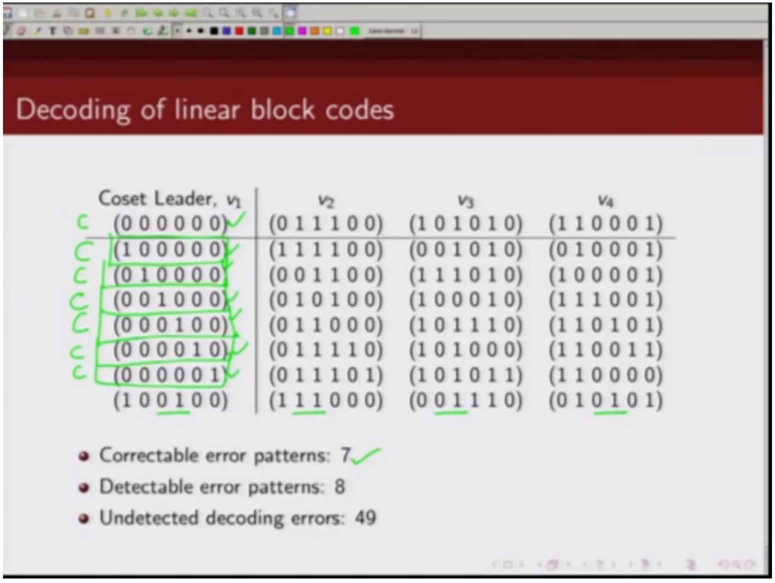
Decoding of linear block codes

	Coset Leader, v_1	v_2	v_3	v_4
C	(0 0 0 0 0 0)✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0)✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0)✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0)✓	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C	(0 0 0 0 1 0)✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C	(0 0 0 0 0 1)✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
	(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7✓
- Detectable error patterns: 8
- Undetected decoding errors: 49

This has 3, this has 3, this has 3,

(Refer Slide Time 41:54)



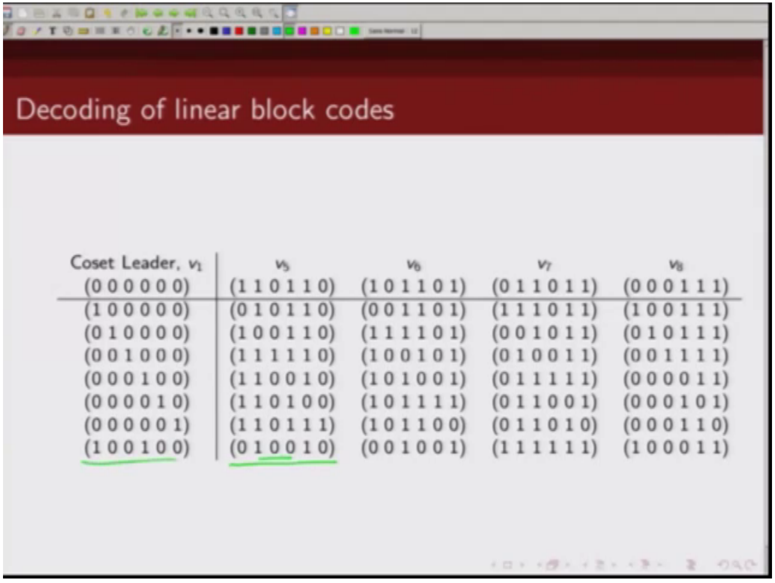
Decoding of linear block codes

	Coset Leader, v_1	v_2	v_3	v_4
C	(0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0) ✓	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C	(0 0 0 0 1 0) ✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C	(0 0 0 0 0 1) ✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
	(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7 ✓
- Detectable error patterns: 8
- Undetected decoding errors: 49

oh this has 2. So this has 2, this also has

(Refer Slide Time 41:59)

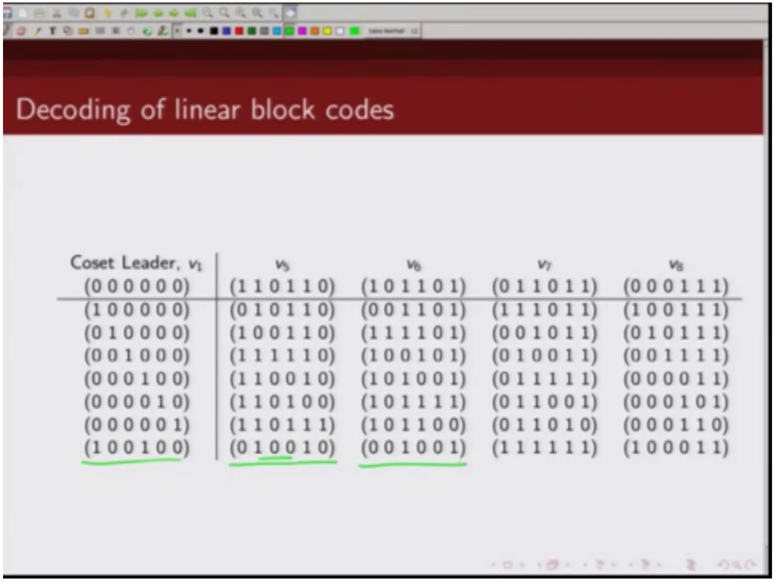


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

2, this also has 2,

(Refer Slide Time 42:03)



Decoding of linear block codes

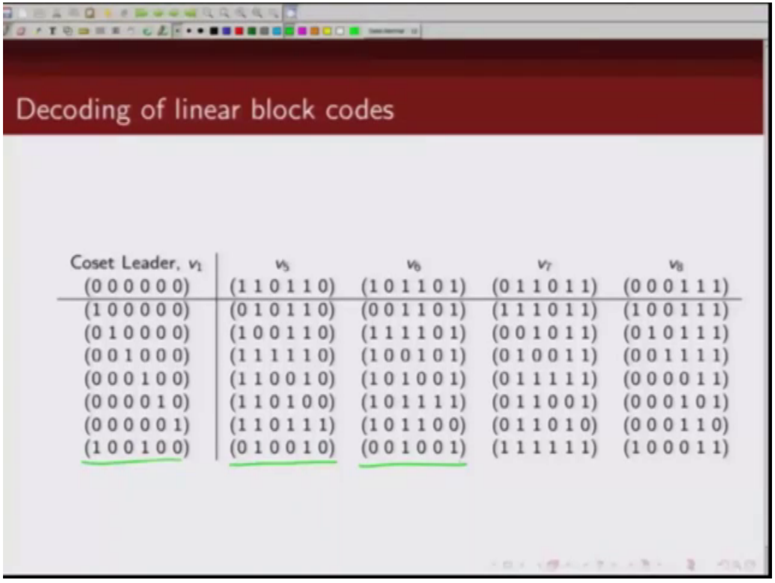
Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

so in this last row my coset leader is no longer

(Refer Slide Time 42:09)

the error pattern with lowest Hamming weight. There are 2 other error patterns

(Refer Slide Time 42:14)

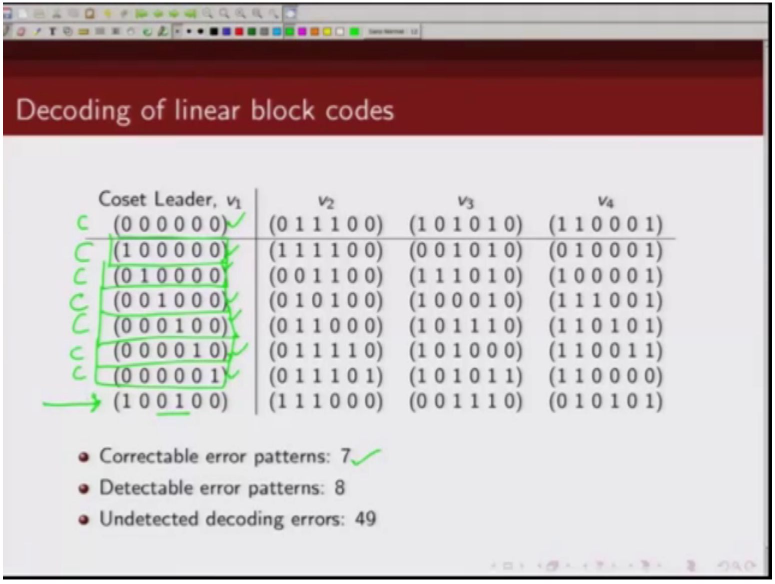


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

which also have Hamming weight 2, Ok. So in this situation when my Syndrome is pointing to this coset,

(Refer Slide Time 42:28)



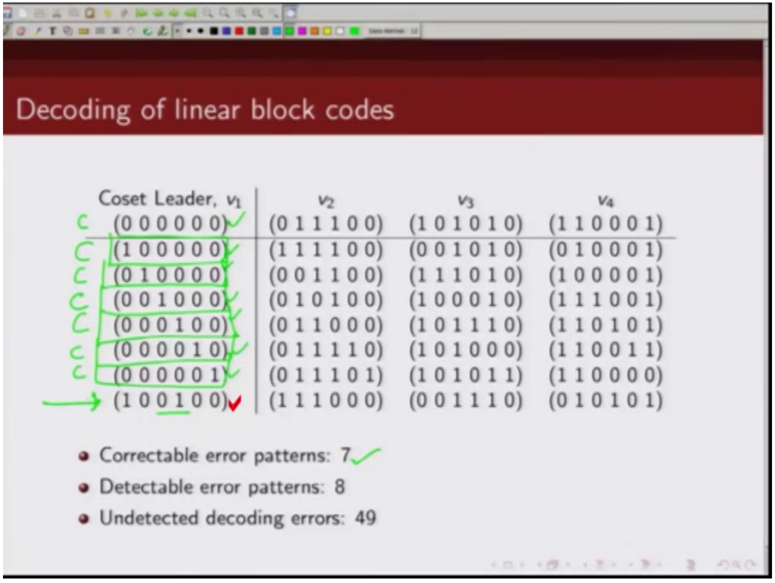
Decoding of linear block codes

	Coset Leader, v_1	v_2	v_3	v_4
C	(0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0) ✓	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0) ✓	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C	(0 0 0 0 1 0) ✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C	(0 0 0 0 0 1) ✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
→	(1 0 0 1 0 0)	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7 ✓
- Detectable error patterns: 8
- Undetected decoding errors: 49

I would not be able to do error correction, why? Because I know this could be a likely error pattern

(Refer Slide Time 42:36)



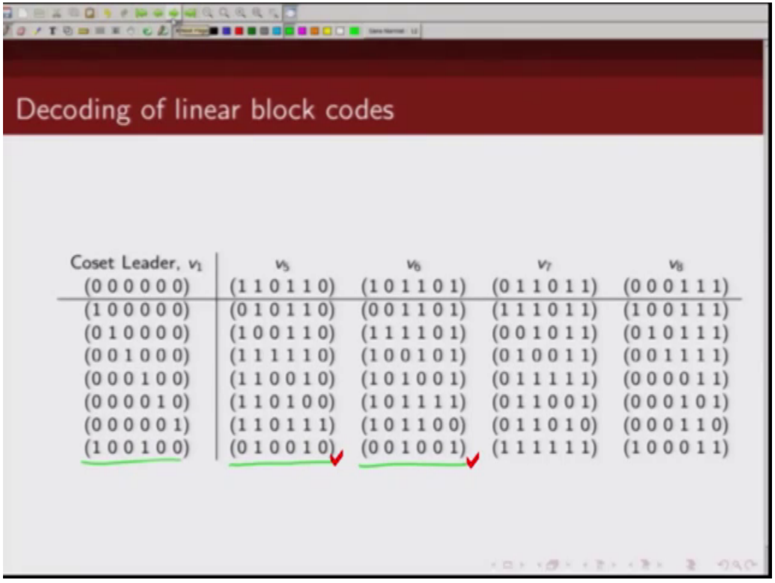
Decoding of linear block codes

	Coset Leader, v_1	v_2	v_3	v_4
C	(0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0) ✓	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0) ✓	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C	(0 0 0 0 1 0) ✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C	(0 0 0 0 0 1) ✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
→	(1 0 0 1 0 0) ✓	(1 1 1 0 0 0)	(0 0 1 1 1 0)	(0 1 0 1 0 1)

- Correctable error patterns: 7 ✓
- Detectable error patterns: 8
- Undetected decoding errors: 49

or this could be a likely error pattern or this could be a likely

(Refer Slide Time 42:40)

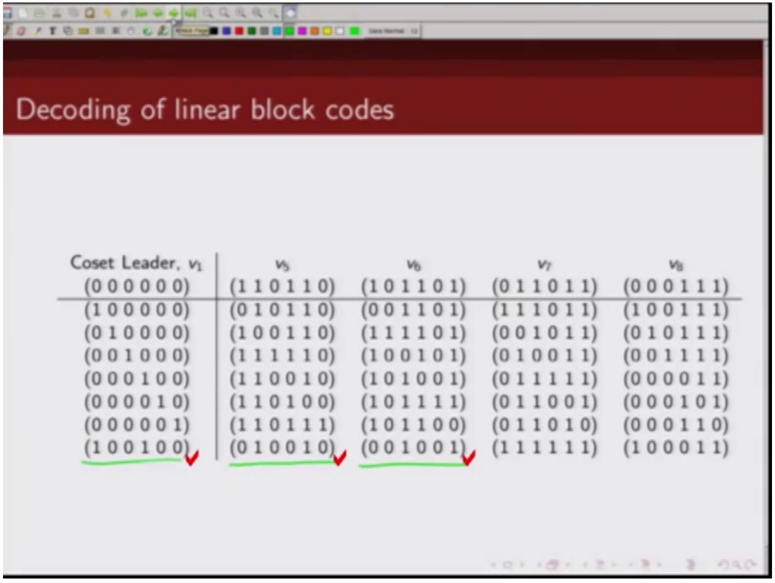


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0) ✓	(0 0 1 0 0 1) ✓	(1 1 1 1 1 1)	(1 0 0 0 1 1)

and they are all equally likely because here 2 bits got flipped. Here 2 bits got flipped. Here 2 bits got flipped. So

(Refer Slide Time 42:47)

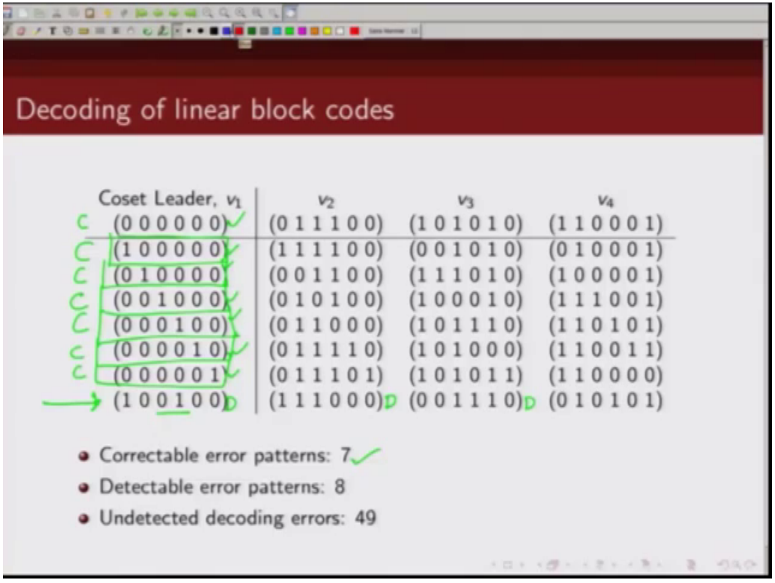


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0) ✓	(0 1 0 0 1 0) ✓	(0 0 1 0 0 1) ✓	(1 1 1 1 1 1)	(1 0 0 0 1 1)

any of these 3 patterns are equally likely in my, in this case. So whenever Syndrome points to this coset I cannot do error correction. However I can detect error. Why? Because whenever it points to this row I know there is an error because Syndrome is non-zero. But I don't know what is my error. So that's why I call it as detectable error pattern. So what are those detectable error pattern? This is my detectable error pattern, this is my detectable error pattern, this is my detectable error pattern, maybe I can use a different this thing, color. So

(Refer Slide Time 43:28)



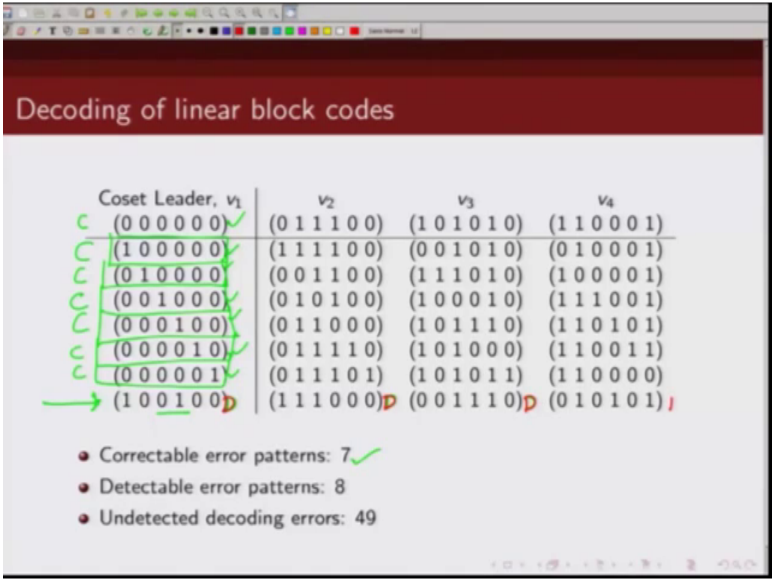
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
C (0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C (1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C (0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C (0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C (0 0 0 1 0 0) ✓	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C (0 0 0 0 1 0) ✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C (0 0 0 0 0 1) ✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
→ (1 0 0 1 0 0) D	(1 1 1 0 0 0) D	(0 0 1 1 1 0) D	(0 1 0 1 0 1)

- Correctable error patterns: 7 ✓
- Detectable error patterns: 8
- Undetected decoding errors: 49

I use red, these are my detectable error pattern. These are my

(Refer Slide Time 43:34)



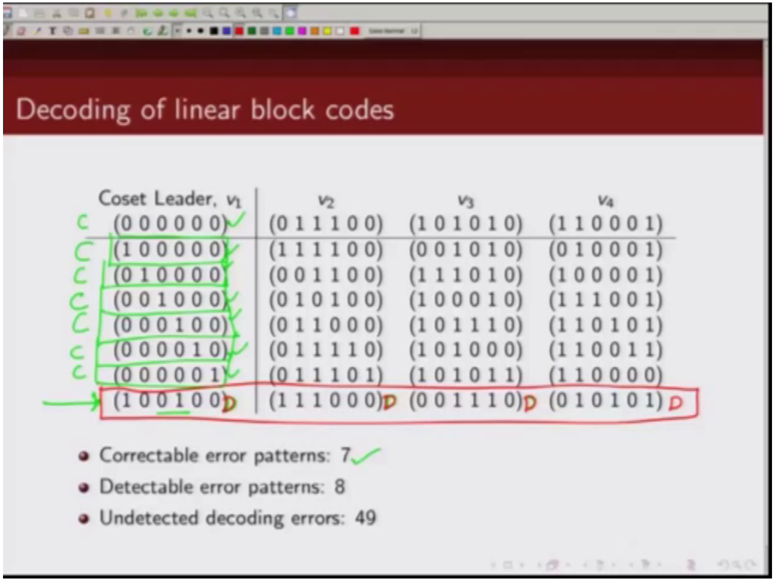
Decoding of linear block codes

	Coset Leader, v_1	v_2	v_3	v_4
C	(0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C	(0 0 0 0 1 0) ✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C	(0 0 0 0 0 1) ✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
→	(1 0 0 1 0 0) D	(1 1 1 0 0 0) D	(0 0 1 1 1 0) D	(0 1 0 1 0 1)

- Correctable error patterns: 7 ✓
- Detectable error patterns: 8
- Undetected decoding errors: 49

detectable error patterns. These are 4

(Refer Slide Time 43:40)



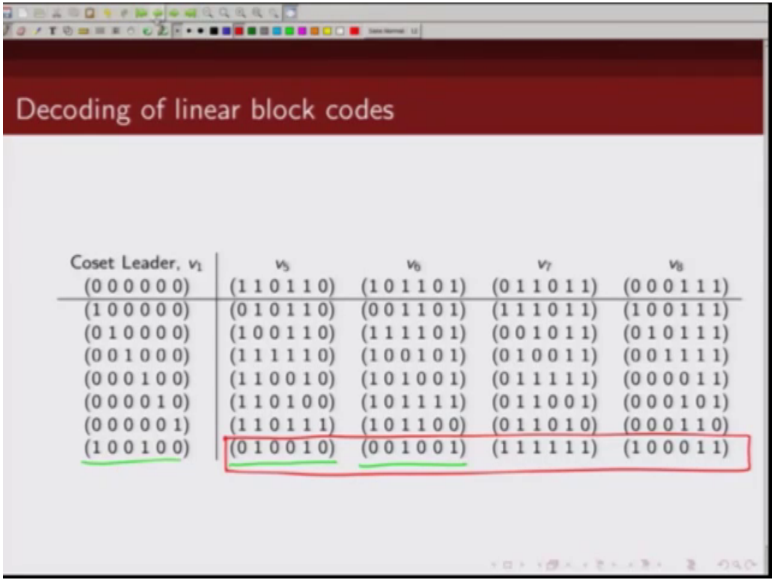
Decoding of linear block codes

	Coset Leader, v_1	v_2	v_3	v_4
C	(0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C	(0 0 0 0 1 0) ✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C	(0 0 0 0 0 1) ✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
→	(1 0 0 1 0 0) D	(1 1 1 0 0 0) D	(0 0 1 1 1 0) D	(0 1 0 1 0 1) D

- Correctable error patterns: 7 ✓
- Detectable error patterns: 8
- Undetected decoding errors: 49

of them, then remaining 4 are these. So these are my

(Refer Slide Time 43:50)

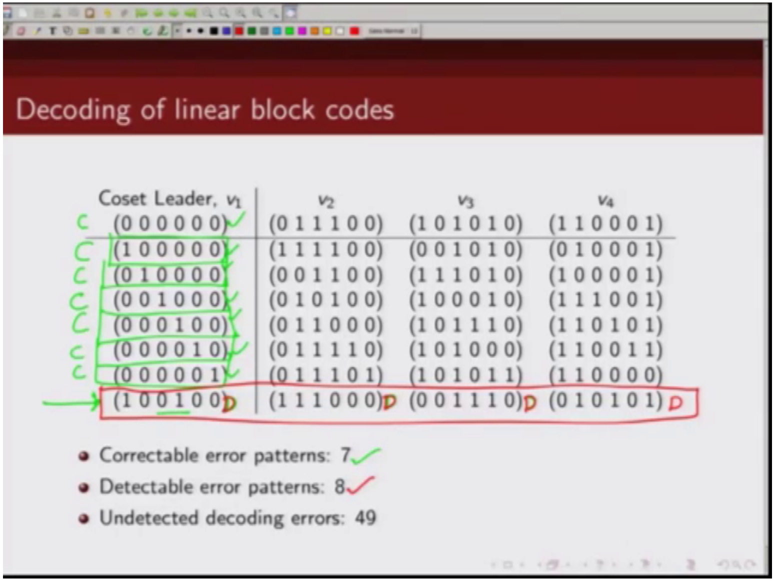


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

8

(Refer Slide Time 43:52)



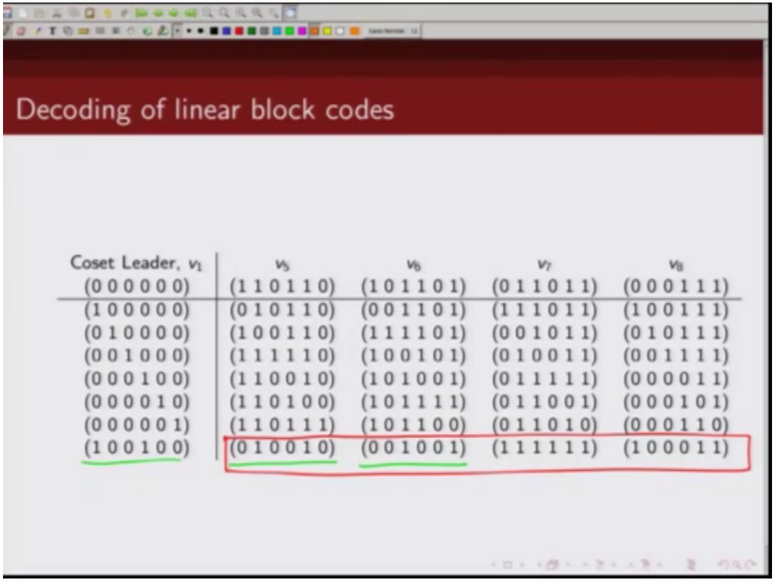
Decoding of linear block codes

	Coset Leader, v_1	v_2	v_3	v_4
C	(0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0) ✓	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C	(0 0 0 0 1 0) ✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C	(0 0 0 0 0 1) ✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
→	(1 0 0 1 0 0) D	(1 1 1 0 0 0) D	(0 0 1 1 1 0) D	(0 1 0 1 0 1) D

- Correctable error patterns: 7 ✓
- Detectable error patterns: 8 ✓
- Undetected decoding errors: 49

error patterns, detectable patterns, error patterns. Now let's use a different color pen. What about these patterns which have been left out? This, this, this like other error patterns? What happens to them? These error patterns, let us say

(Refer Slide Time 44:12)

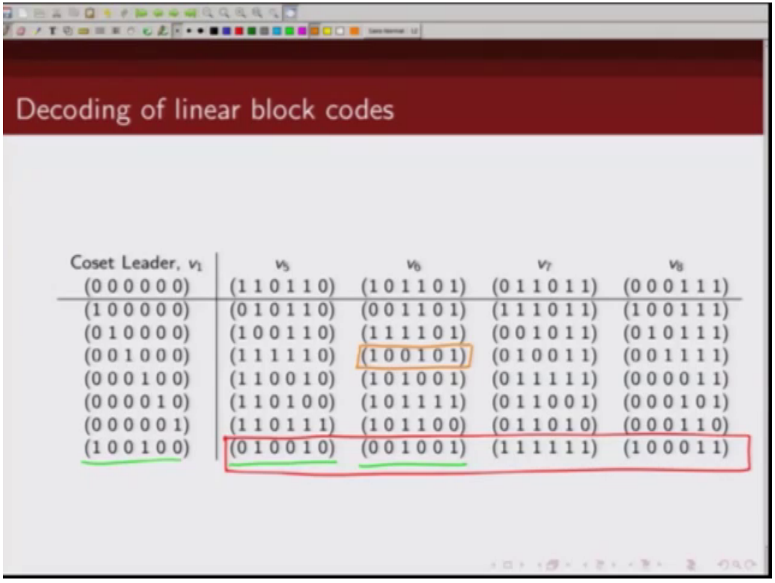


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(000000)	(110110)	(101101)	(011011)	(000111)
(100000)	(010110)	(001101)	(111011)	(100111)
(010000)	(100110)	(111101)	(001011)	(010111)
(001000)	(111110)	(100101)	(010011)	(001111)
(000100)	(110010)	(101001)	(011111)	(000011)
(000010)	(110100)	(101111)	(011001)	(000101)
(000001)	(110111)	(101100)	(011010)	(000110)
(100100)	(010010)	(001001)	(111111)	(100011)

if this error pattern happens what is going to happen?

(Refer Slide Time 44:16)

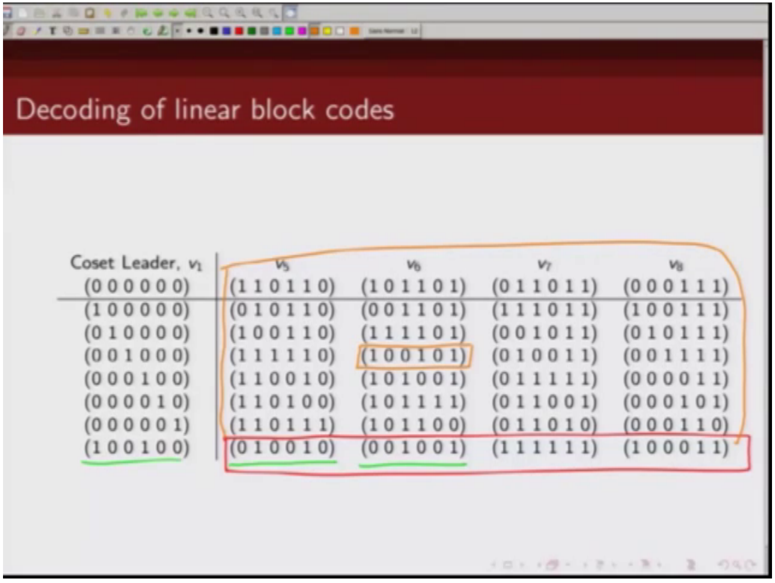


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(000000)	(110110)	(101101)	(011011)	(000111)
(100000)	(010110)	(001101)	(111011)	(100111)
(010000)	(100110)	(111101)	(001011)	(010111)
(001000)	(111110)	(100101)	(010011)	(001111)
(000100)	(110010)	(101001)	(011111)	(000011)
(000010)	(110100)	(101111)	(011001)	(000101)
(000001)	(110111)	(101100)	(011010)	(000110)
(100100)	(010010)	(001001)	(111111)	(100011)

If this happens I am not able to detect these error patterns, why? Whenever this error pattern happens, this is Hamming weight 3; its coset leader was already Hamming weight 1. So whenever these error patterns happen, whenever these error patterns happen, I am not able to detect

(Refer Slide Time 44:40)

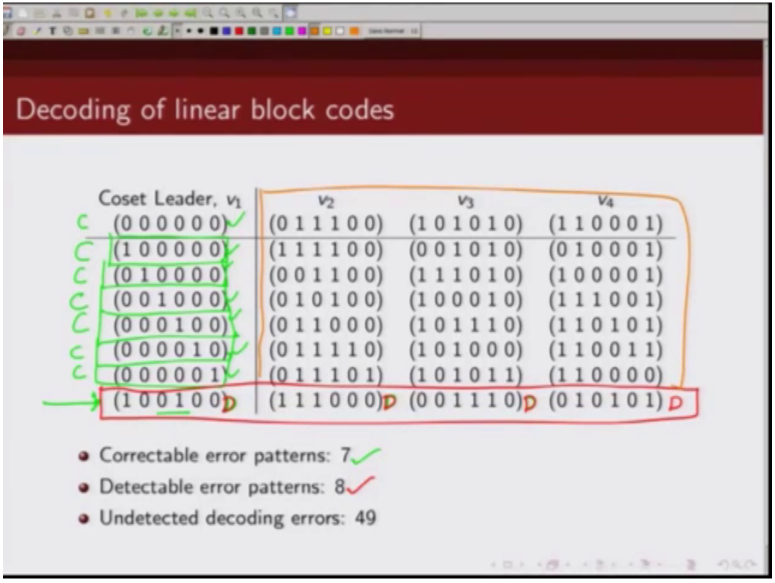


Decoding of linear block codes

Coset Leader, v_1	v_5	v_6	v_7	v_8
(0 0 0 0 0 0)	(1 1 0 1 1 0)	(1 0 1 1 0 1)	(0 1 1 0 1 1)	(0 0 0 1 1 1)
(1 0 0 0 0 0)	(0 1 0 1 1 0)	(0 0 1 1 0 1)	(1 1 1 0 1 1)	(1 0 0 1 1 1)
(0 1 0 0 0 0)	(1 0 0 1 1 0)	(1 1 1 1 0 1)	(0 0 1 0 1 1)	(0 1 0 1 1 1)
(0 0 1 0 0 0)	(1 1 1 1 1 0)	(1 0 0 1 0 1)	(0 1 0 0 1 1)	(0 0 1 1 1 1)
(0 0 0 1 0 0)	(1 1 0 0 1 0)	(1 0 1 0 0 1)	(0 1 1 1 1 1)	(0 0 0 0 1 1)
(0 0 0 0 1 0)	(1 1 0 1 0 0)	(1 0 1 1 1 1)	(0 1 1 0 0 1)	(0 0 0 1 0 1)
(0 0 0 0 0 1)	(1 1 0 1 1 1)	(1 0 1 1 0 0)	(0 1 1 0 1 0)	(0 0 0 1 1 0)
(1 0 0 1 0 0)	(0 1 0 0 1 0)	(0 0 1 0 0 1)	(1 1 1 1 1 1)	(1 0 0 0 1 1)

any of these error patterns, why? Because any of these error

(Refer Slide Time 44:46)



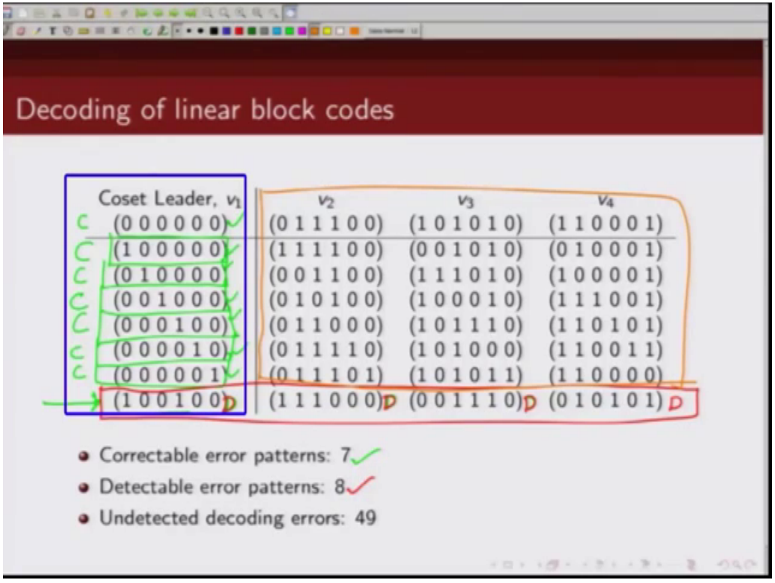
Decoding of linear block codes

	Coset Leader, v_1	v_2	v_3	v_4
C	(0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0) ✓	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C	(0 0 0 0 1 0) ✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C	(0 0 0 0 0 1) ✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
→	(1 0 0 1 0 0) D	(1 1 1 0 0 0) D	(0 0 1 1 1 0) D	(0 1 0 1 0 1) D

- Correctable error patterns: 7 ✓
- Detectable error patterns: 8 ✓
- Undetected decoding errors: 49

patterns happen, any of these error patterns happen, I have already taken decision in favor of these coset leaders

(Refer Slide Time 44:53)



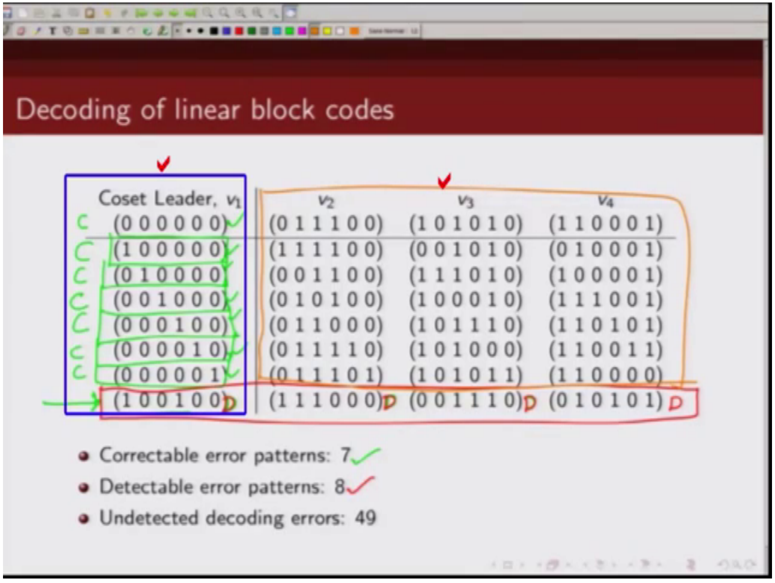
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
C (0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C (1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C (0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C (0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C (0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C (0 0 0 0 1 0) ✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C (0 0 0 0 0 1) ✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0) D	(1 1 1 0 0 0) D	(0 0 1 1 1 0) D	(0 1 0 1 0 1) D

- Correctable error patterns: 7 ✓
- Detectable error patterns: 8 ✓
- Undetected decoding errors: 49

because it is less likely to get this error pattern than this. So

(Refer Slide Time 44:59)



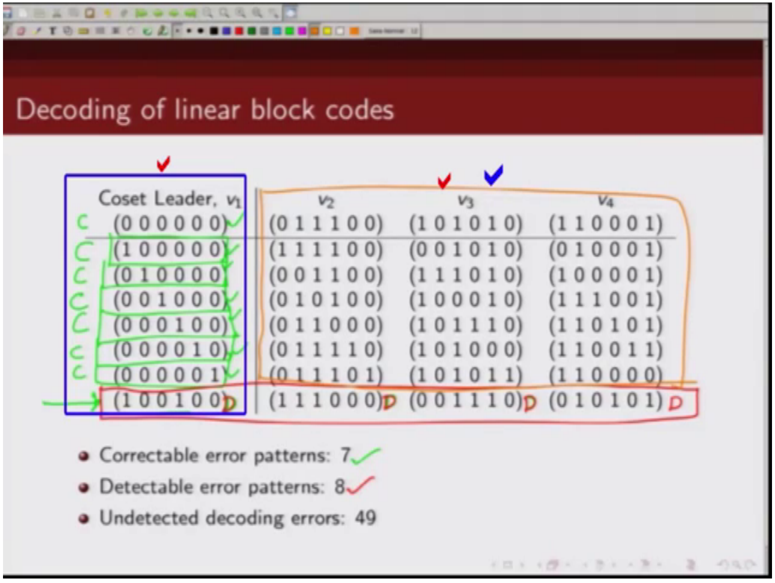
Decoding of linear block codes

Coset Leader, v_1	v_2	v_3	v_4
C (0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C (1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C (0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C (0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C (0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C (0 0 0 0 1 0) ✓	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C (0 0 0 0 0 1) ✓	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
(1 0 0 1 0 0) D	(1 1 1 0 0 0) D	(0 0 1 1 1 0) D	(0 1 0 1 0 1) D

- Correctable error patterns: 7 ✓
- Detectable error patterns: 8 ✓
- Undetected decoding errors: 49

these set of 49 error patterns, if any of these error patterns happen then I am going to make a mistake.

(Refer Slide Time 45:11)



Decoding of linear block codes

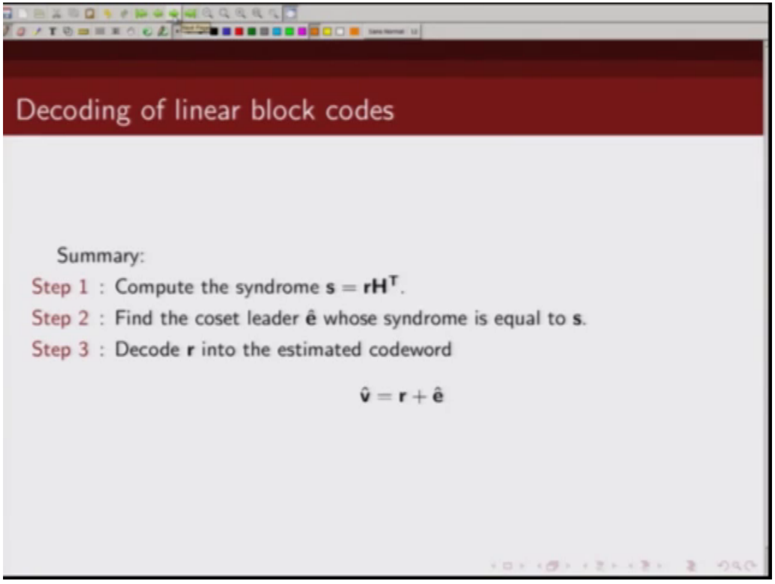
	Coset Leader, v_1	v_2	v_3	v_4
C	(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0)	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0)	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0)	(0 1 1 0 0 0)	(1 0 1 1 1 0)	(1 1 0 1 0 1)
C	(0 0 0 0 1 0)	(0 1 1 1 1 0)	(1 0 1 0 0 0)	(1 1 0 0 1 1)
C	(0 0 0 0 0 1)	(0 1 1 1 0 1)	(1 0 1 0 1 1)	(1 1 0 0 0 0)
	(1 0 0 1 0 0) D	(1 1 1 0 0 0) D	(0 0 1 1 1 0) D	(0 1 0 1 0 1) D

- Correctable error patterns: 7✓
- Detectable error patterns: 8✓
- Undetected decoding errors: 49

This will result in undetected error because whenever any of these error patterns happen, I would assume that the error pattern was this. So this would result in undetected error probability. So through this example essentially I have illustrated how we can use the standard array for error correction and error detection.

So in summary we would like to make our coset leader as one having minimum Hamming weight, minimum number of 1s. And whenever we get an error, our Syndrome will be non-zero, Syndrome will point out to a particular coset or a row of the standard array, and we will, if the coset leader has the minimum number of 1s, then we will pick that coset leader as our likely error pattern and we are going to decode it by picking that likely error pattern, adding

(Refer Slide Time 46:21)



Decoding of linear block codes

Summary:

Step 1 : Compute the syndrome $\mathbf{s} = \mathbf{r}\mathbf{H}^T$.

Step 2 : Find the coset leader $\hat{\mathbf{e}}$ whose syndrome is equal to $\mathbf{s}$.

Step 3 : Decode $\mathbf{r}$ into the estimated codeword

$$\hat{\mathbf{v}} = \mathbf{r} + \hat{\mathbf{e}}$$

it to our received codeword and that would be our estimated codeword. So with this I conclude uh

(Refer Slide Time 46:32)

this decoding of linear block codes. Thank you