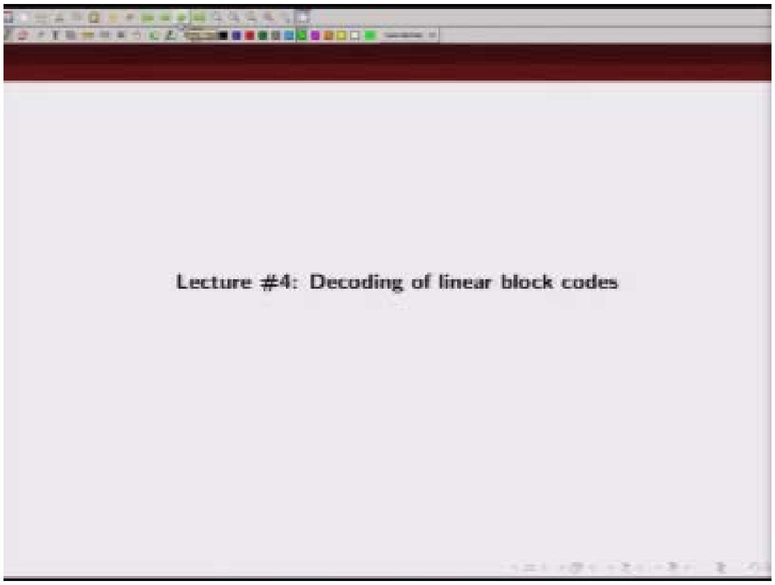
Indian Institute of Technology Kanpur
National Programme on Technology Enhanced Learning (NPTEL)
Course Title
Error Control Coding: An Introduction to Linear Block Codes

Lecture-4
Decoding of Linear Block Codes

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

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

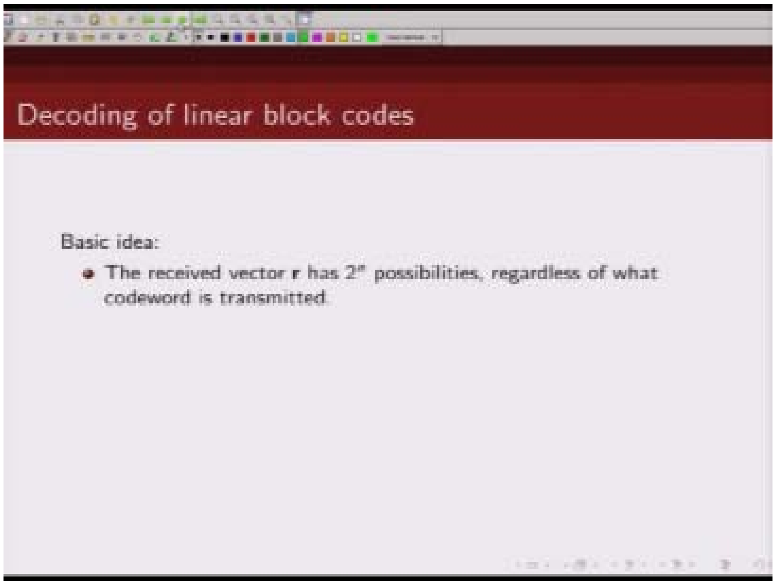
(Refer Slide Time: 00:22)



Lecture #4: Decoding of linear block codes

Today we are going to talk about how to decode and we are going to talk about what is known as syndrome decoding.

(Refer Slide Time: 00:29)



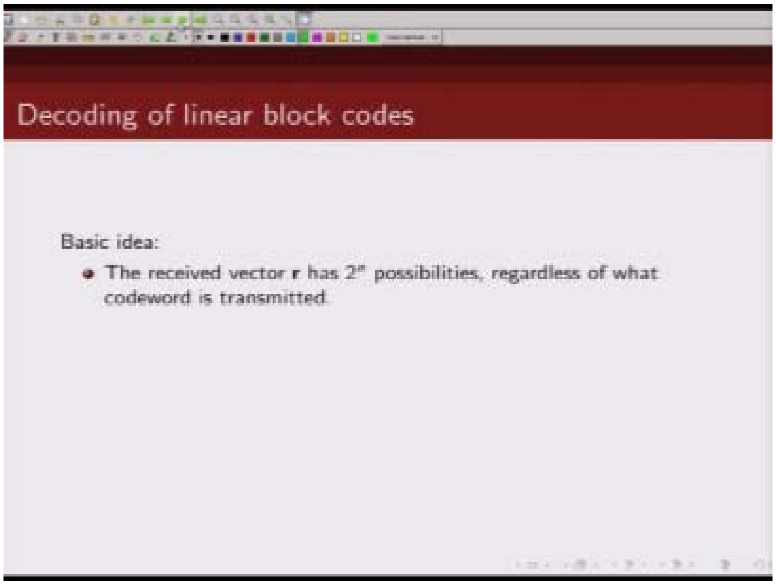
Decoding of linear block codes

Basic idea:

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

So what is a decoding problem? So let us look at scenario where we are transmitting our code word which is n bit tuple over a communication channel. And let us consider a binary symmetric channel, so in a binary symmetric channel bits zeros and ones are transmitted over this communication channel and with probability 1-P they are received correctly and with crossover probability of P the bits zeros and ones can get flipped. So the output channel is also binary.

(Refer Slide Time: 01:05)



Decoding of linear block codes

Basic idea:

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

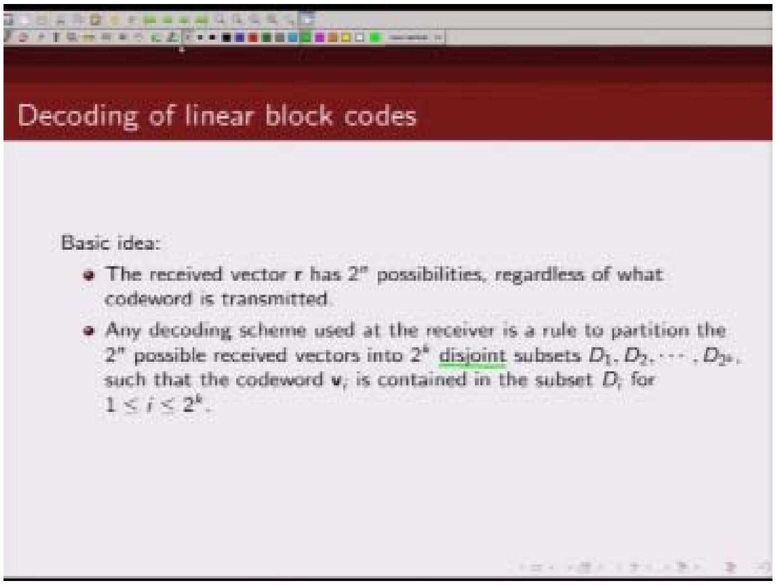
Then we have total 2_k code words right and if we are looking at output received sequence we can have total of 2^n different possibilities, because we – our code bit is – code word is n bit and each location can be zeros or ones. So our decoding problem is we have to partition these 2_n possibilities into sets of 2_kx2_k set. So we have to map these 2_n possibilities into sets, 2_k sets. And corresponding to each set there should be only one unique code word.

In other words when we map our 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 received sequence there is only one code word. 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 code word.

So whenever any of these received sequence you know is mapped to a particular set, then we should be able to say okay if you receive any of these code or receive any of these sequences then the transmitted code word is also this. So you can think of, we have total 2_n possibilities and we want to partition this 2_n possibilities into 2^k different balls in some sense. And each ball has one code word associated with it.

That is 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: 03:15)



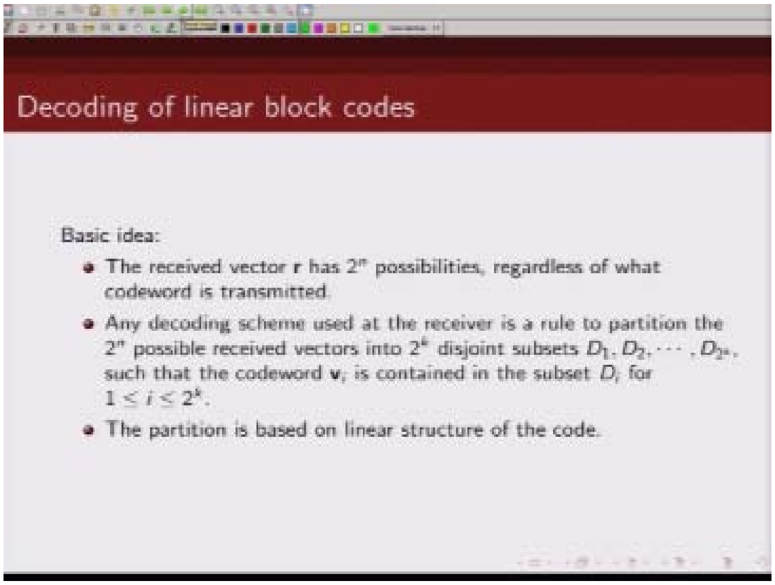
Decoding of linear block codes

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 $1 \leq i \leq 2^k$.

So any decoding scheme is basically to partition these 2_n possible code words received code words into 2_k disjoint subsets, to where disjoint is important because we want to – we do not want basically overlapping decision sets, regions okay.

(Refer Slide Time: 03:45)



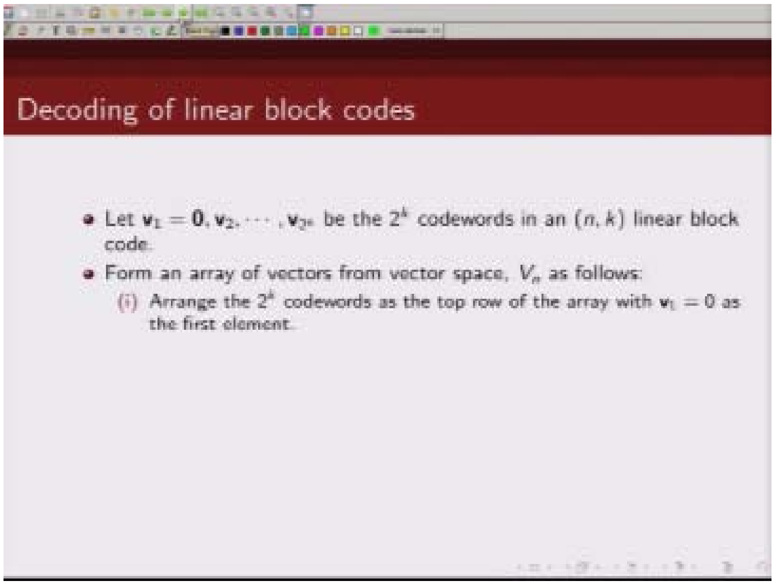
Decoding of linear block codes

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 $1 \leq i \leq 2^k$.
- 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 code and that is what we are going to describe in this lecture.

(Refer Slide Time: 03:57)

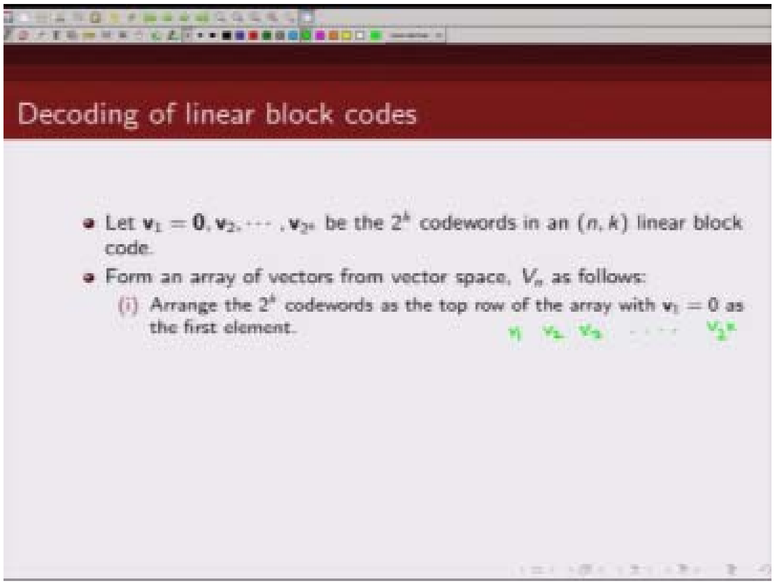


Decoding of linear block codes

- 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.

So let us denote the 2_k code words by v_1, v_2, v_3, $v2_k$, $v2^k$ so let us say these are my 2_k code words in a linear block code (n, k) linear block code where v_1 is all zero code word. Now what do we do is we form an array of vectors from vector space V_n as follows. In the top row of this array we will arrange all of these 2_k code words with all zero code word being the left most entry in the row.

(Refer Slide Time: 04:46)



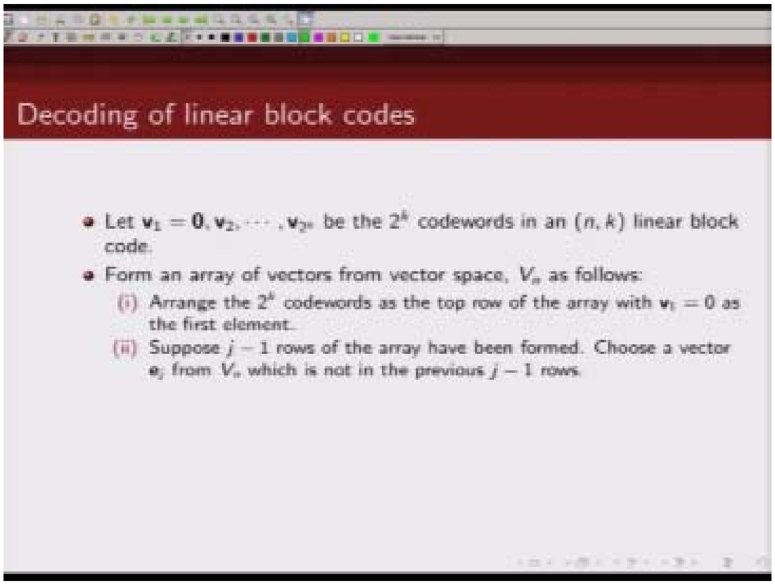
Decoding of linear block codes

- 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.

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

So we are going in the first row of this array, we are going to put all zero code word and then we are going to put the other code words. So this will be the first row of this array.

(Refer Slide Time: 05:03)

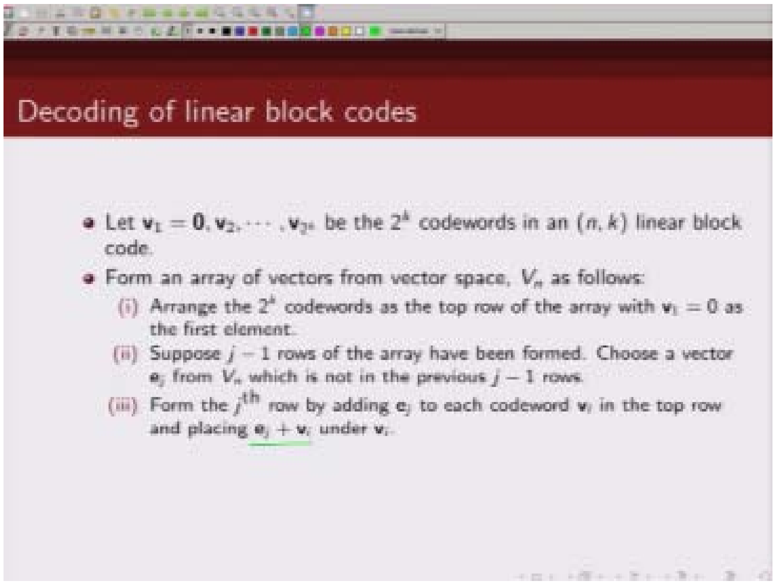


Decoding of linear block codes

- 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.

Now suppose we have formed – now I am going to tell you how we are going to form this array, suppose let us say, we have already formed j-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 previously in any of the previous j-1 rows. We pick that error vector.

(Refer Slide Time: 05:37)

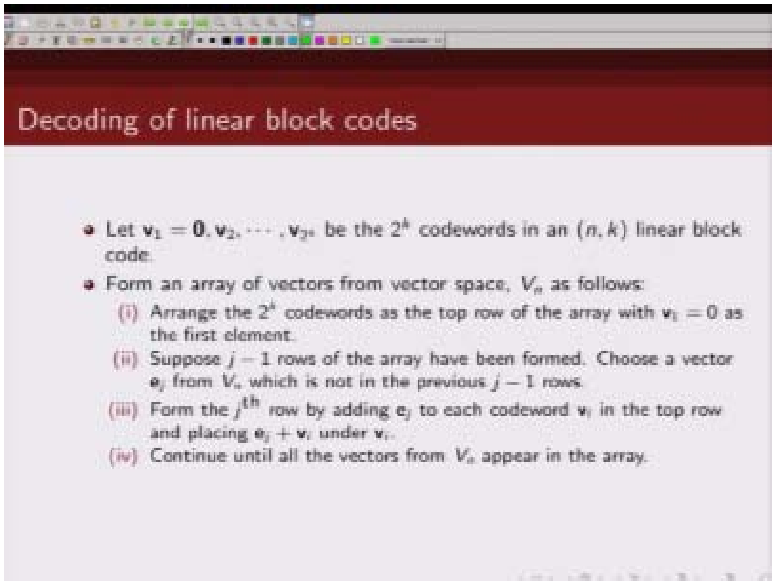


Decoding of linear block codes

- 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 $\mathbf{v}_i$.

Next we form the j^{th} row by adding that error vector to each of the code words on the top row and placing the new vector which is $e_j + v_i$ under the code word v_i.

(Refer Slide Time: 05:58)

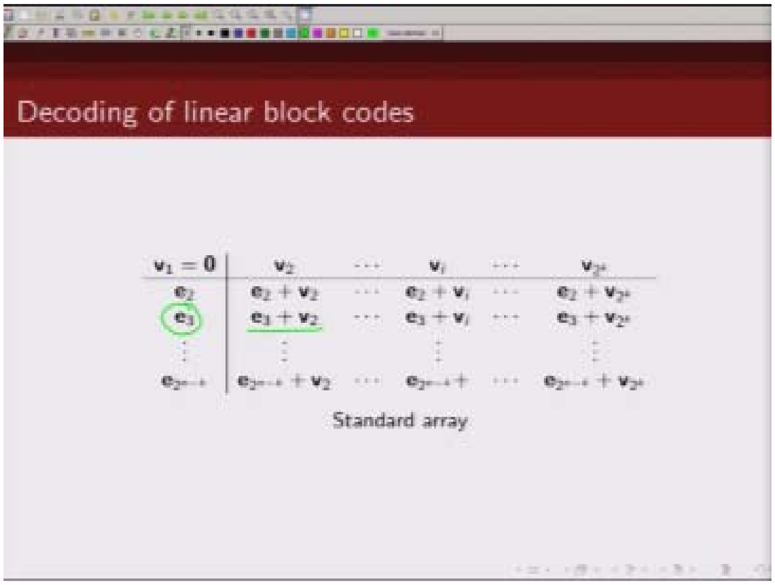


Decoding of linear block codes

- 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.

I will just explain what I mean by this.

(Refer Slide Time: 06:00)



Decoding of linear block codes

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

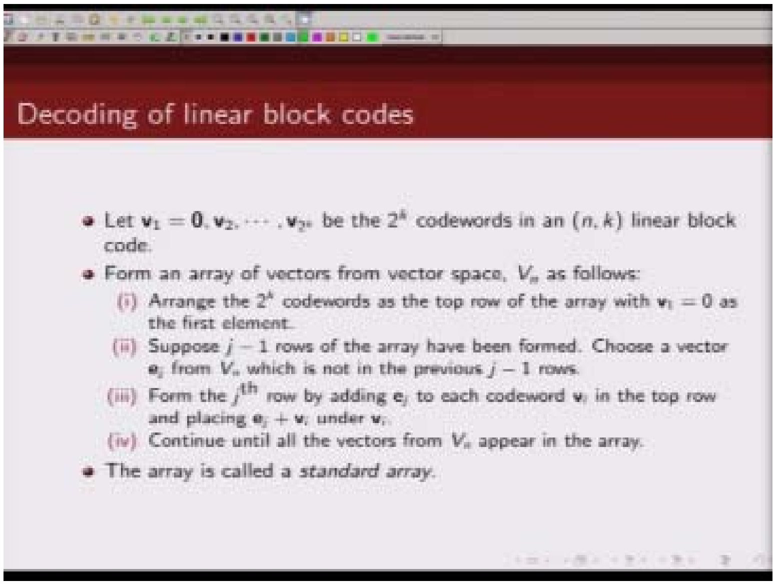
Standard array

So as I said, in the first row we have listed all the code words with all zero code word as my left most entry. Now let us 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 which is – let us say this e_3 which has not appeared before. So e_3 should not be any of these elements which have already been chosen. E_3 should not have appeared in the previous rows of this array.

If I chose 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 code words and I place that element under the same column. Now what do I mean by that, so let us say this was v_2 so I will add e_3 to v_2 and I will add the element e_3+v_2 in the same column as v_2 was. Similarly if I have a code word v_i I will add e_3 to v_i and I will add this element e_3+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: 08:11)

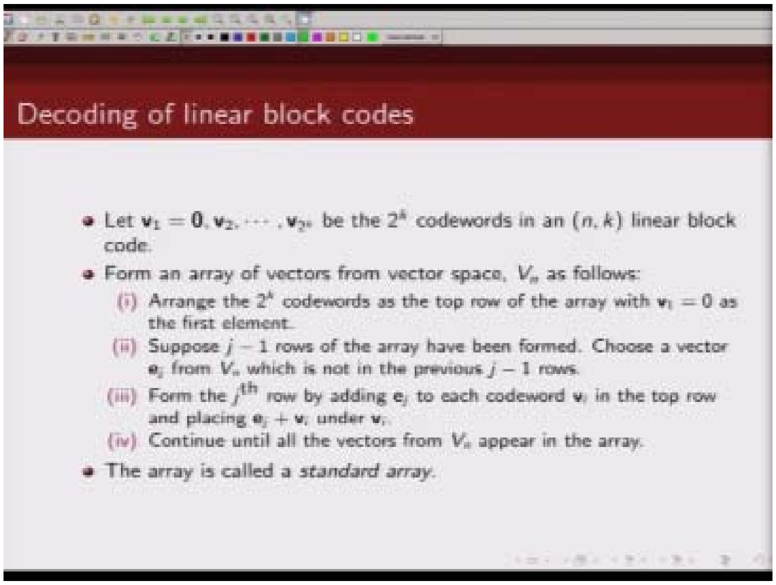


Decoding of linear block codes

- 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 is what I meant, form the j^{th} row by adding e_j to each of these code words in the top row and placing $e_j + v_i$ under the same column as v_i. And we will continue doing this until all n bit vectors have been put in this standard array.

(Refer Slide Time: 08:40)

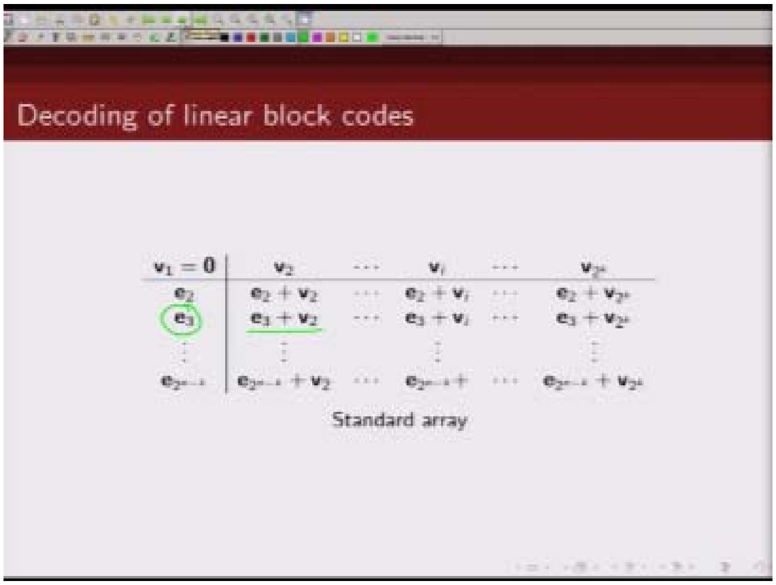


Decoding of linear block codes

- 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.
  - (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*.

And this array formed in this way is known as standard array.

(Refer Slide Time: 08:44)



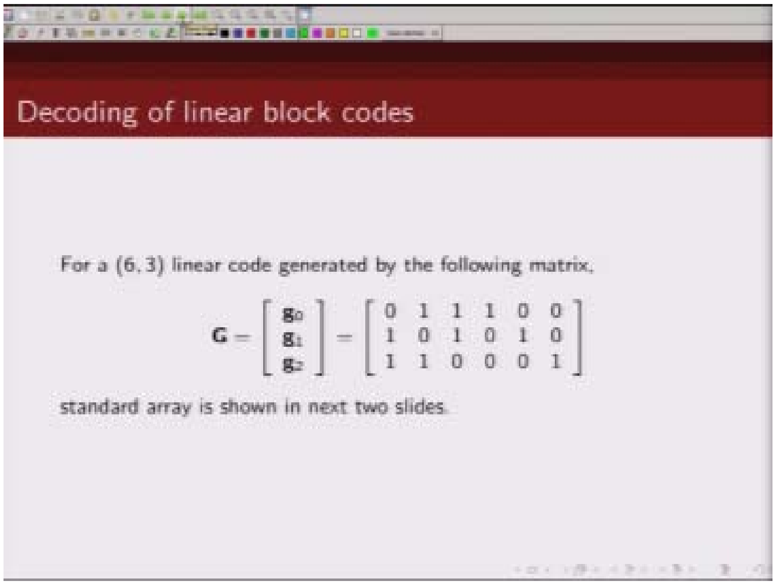
Decoding of linear block codes

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

Standard array

So this is how your standard array will look like. Again I just recap how we are constructing this standard array. The first row of this array is set of code words v$_1$2v2^k and the left most entry here is all zero code word. Next we pick up an element an error vector which has not happened in any of the previous rows. And then we add that error vector to each of the elements of these code words and put the new element under the same column as that code word. And we continue doing that until we have put all the possible vectors in this array.

(Refer Slide Time: 09:45)



Decoding of linear block codes

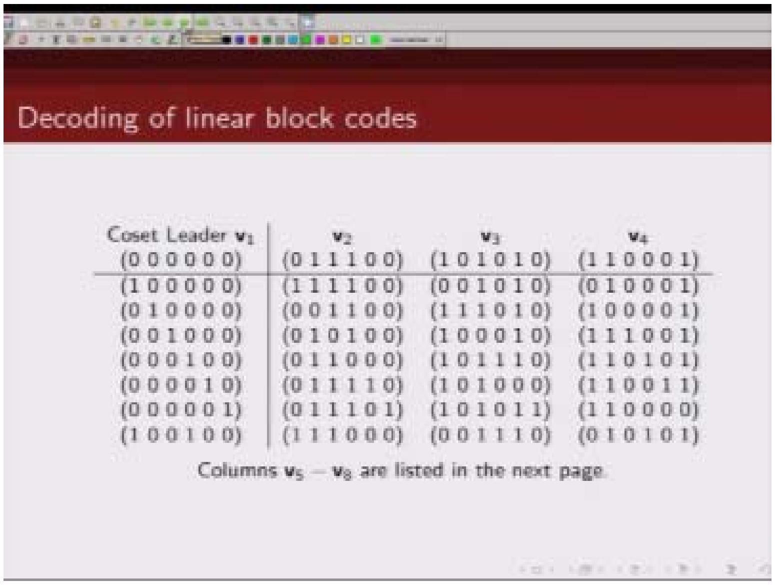
For a (6, 3) linear code generated by the following matrix,

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

standard array is shown in next two slides.

So let us look at a (6, 3) linear block codes whose generator matrix is given here.

(Refer Slide Time: 09:53)



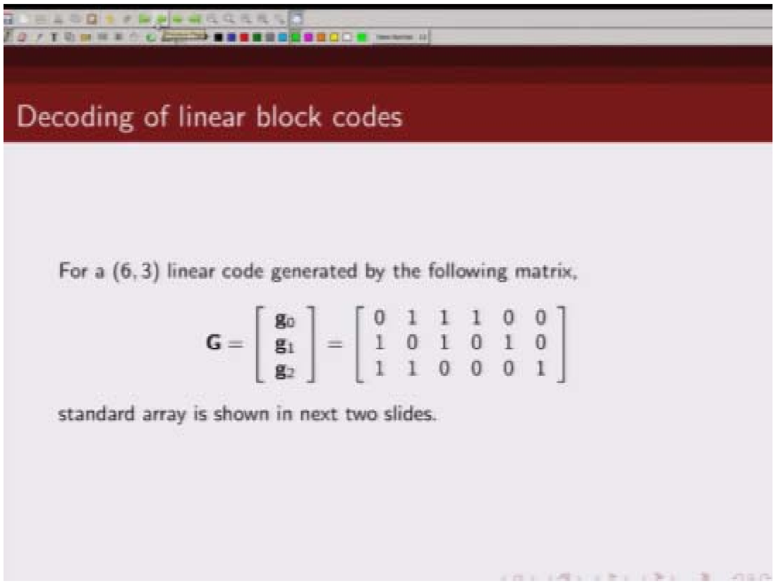
Decoding of linear block codes

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.

(Refer Slide Time: 10:03)



Decoding of linear block codes

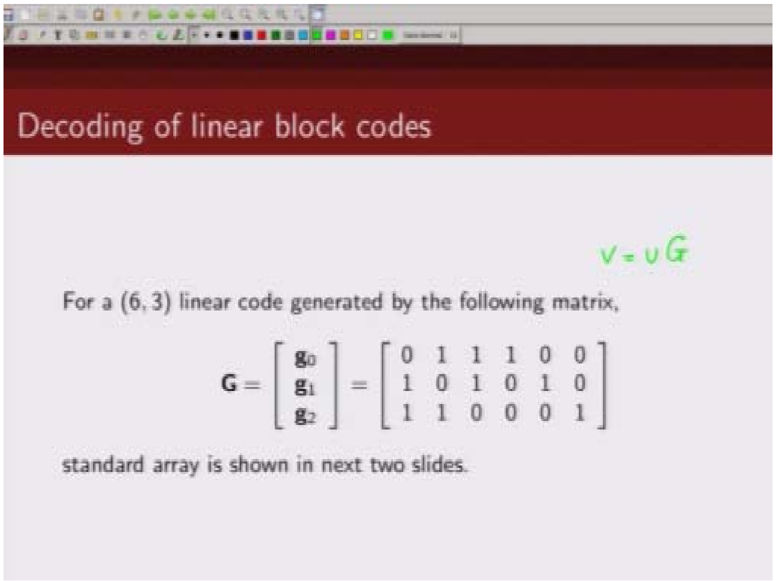
For a (6, 3) linear code generated by the following matrix,

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

standard array is shown in next two slides.

The first step involved is you need to write down all possible code words, now you have already been given the generator matrix for this code.

(Refer Slide Time: 10:17)



Decoding of linear block codes

$v = uG$

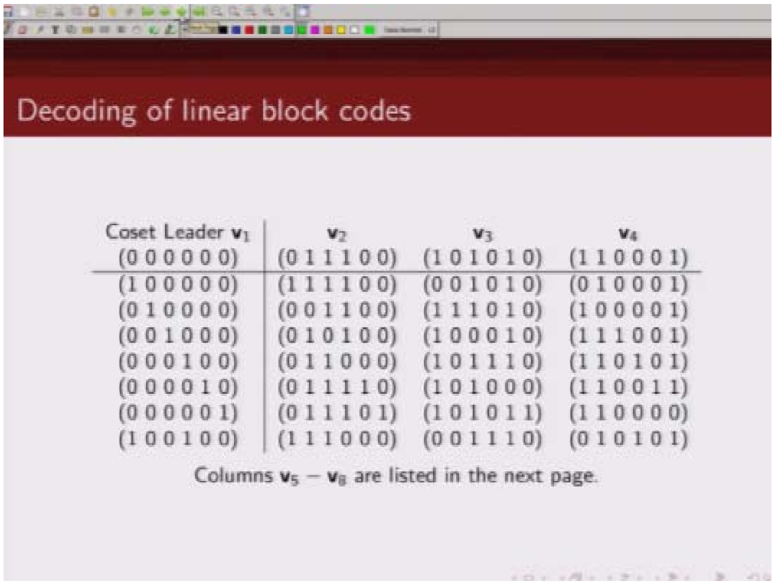
For a $(6,3)$ linear code generated by the following matrix,

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

standard array is shown in next two slides.

So you can find out what are the possible code words, you just have to do v is nothing but u times G, so there are total eight code words and you can find those eight code words because you know that G is given to u and you know what is your u, u is basically goes from 000 to 001, 010, goes to 111 so you can find out what these code words are and I have listed these

(Refer Slide Time: 10:52)



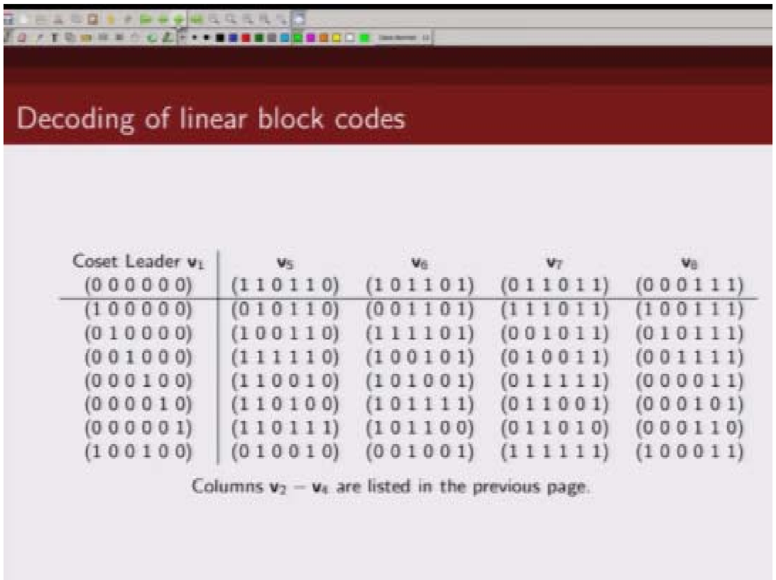
Decoding of linear block codes

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.

Codes words here so you have one all 0 code word and then you have the other code words are 011100, 101010, 11001001, since I could

(Refer Slide Time: 11:07)



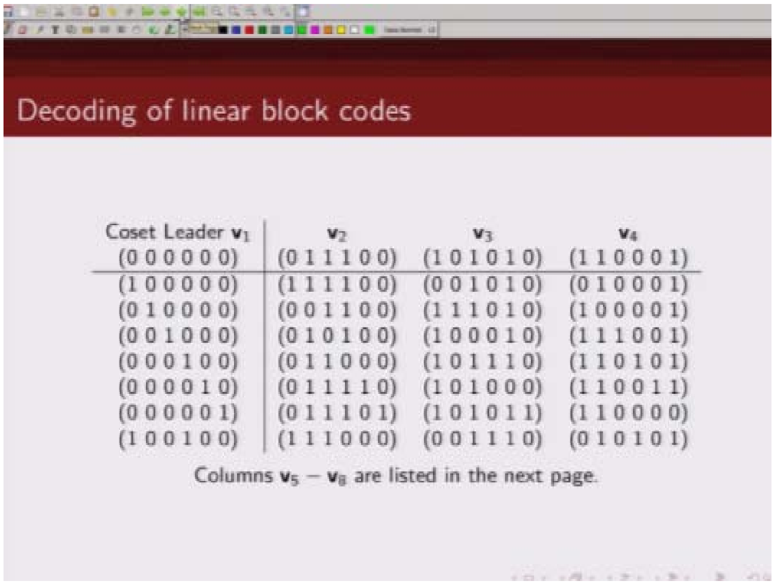
Decoding of linear block codes

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.

Not fit in all the columns in one slide I have continued it here so this is I had up to

(Refer Slide Time: 11:19)



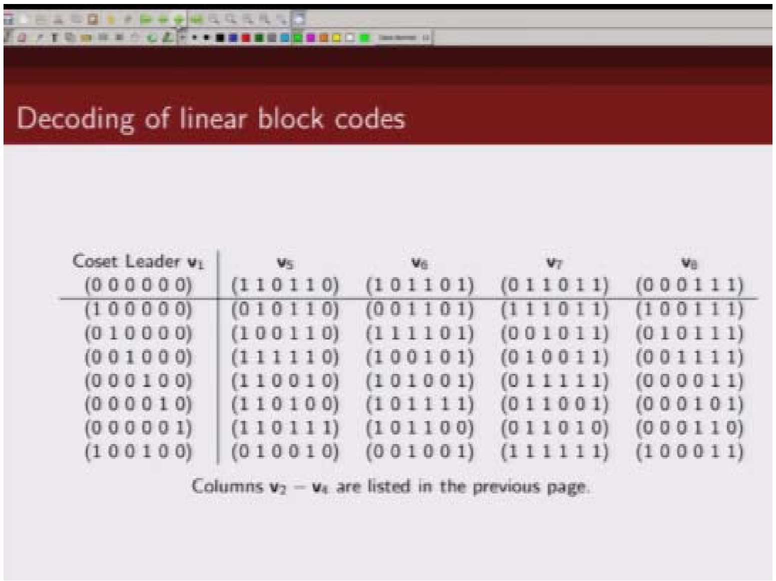
Decoding of linear block codes

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.

V_o I have V_1, V_2, V_3, V_4 and then I had.

(Refer Slide Time: 11:24)



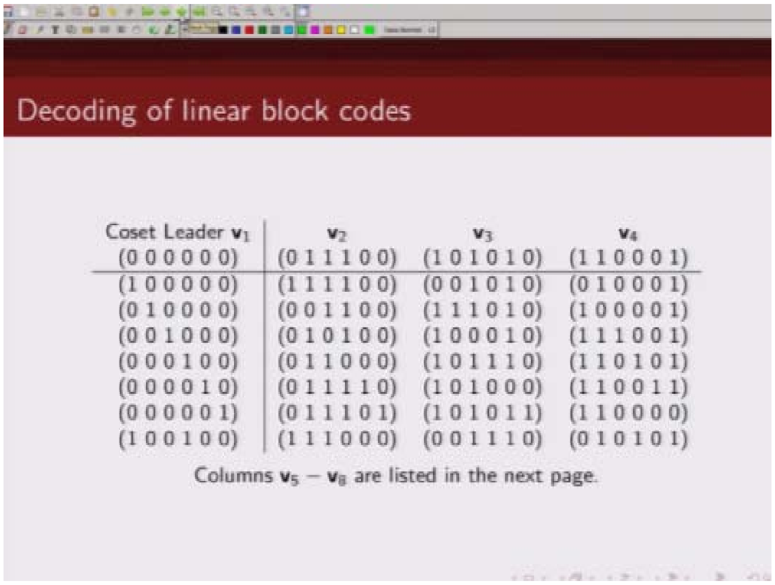
Decoding of linear block codes

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.

V_5, V_6, V_7, V_8, so these are my eight code words for this six three linear block code.

(Refer Slide Time: 11:35)



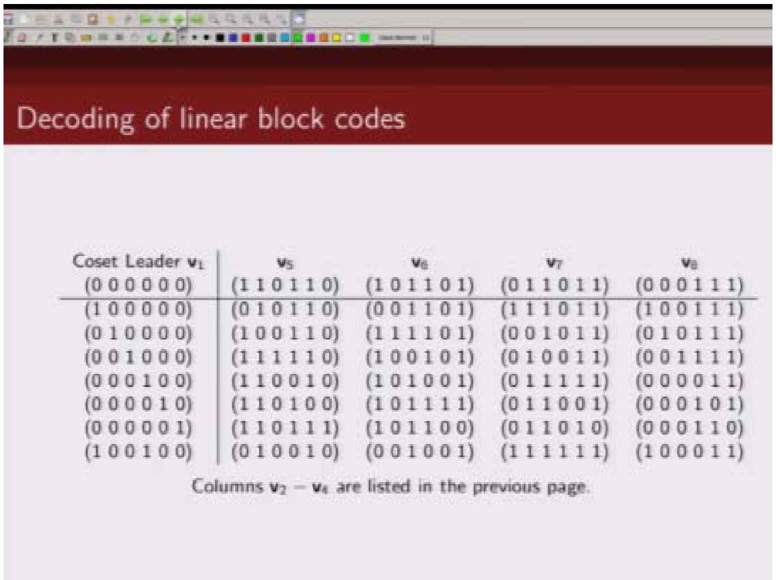
Decoding of linear block codes

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 entries in the next column, as I said I have to pick up a vector which has not appeared in the previous rows. So let us look at what is appeared in the previous row we had all zero sequence here, we had a sequence which has 3 ones, a sequence which has 3 ones, sequence which has I mean

(Refer Slide Time: 11:59)



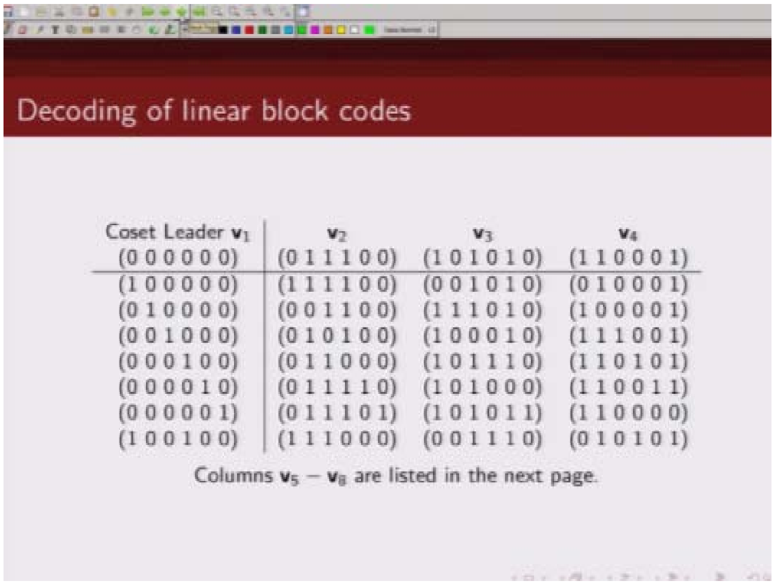
Decoding of linear block codes

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.

3 ones, code word 4 ones, so we see we do not have n tupels which have just weight one hamming weight one

(Refer Slide Time: 21:11)



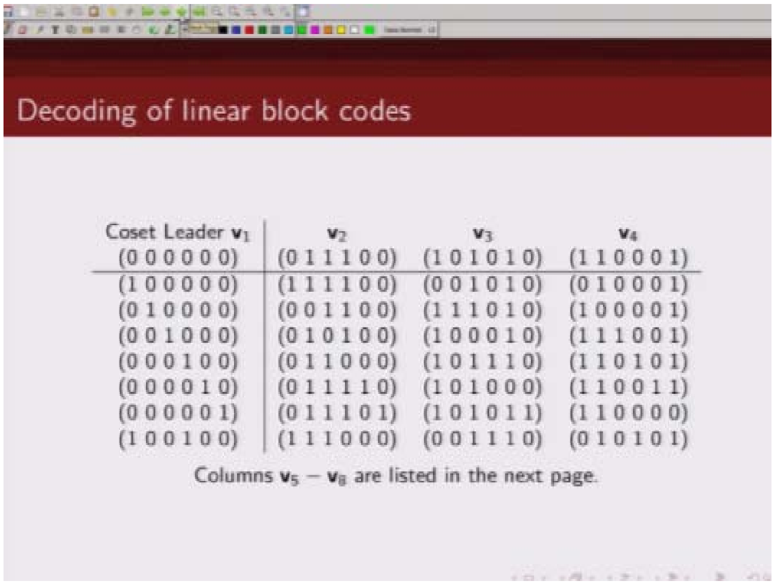
Decoding of linear block codes

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.

They have not appeared so far so this 100000 has not appeared so far in the first row of this array, so I pick this 100000 as my first element now how do I find this element? I am going to add this to this so I am going to add this to this, if I add it what do I get? 1+ 0 is 1, 0+1 is 1, 0+1 is 1,l 0+1 is1,0+0 is 0, 0+0 is 0 so this is what I get, how do I get this entry? Again I add this to this so 1+1 is 0, 0+ 0 is 0, 0+1 is 1, 0+0 is 0, 0+1 is 1, 0+0 is 0. So this is how I populate the entries in this row. Next how do I pick this, I will now have to look at

(Refer Slide Time: 13:24)



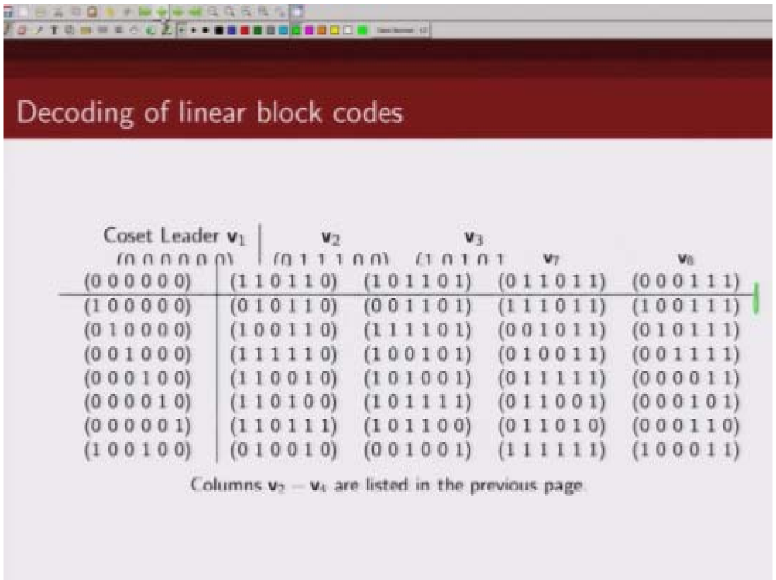
Decoding of linear block codes

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.

The first two rows of this array and see, pick one end tuple which has not happened before and this particular n tuple can see 0 1 and all zeros.

(Refer Slide Time: 13:40)



Decoding of linear block codes

Coset Leader $\mathbf{v}_1$ | $\mathbf{v}_2$ $\mathbf{v}_3$

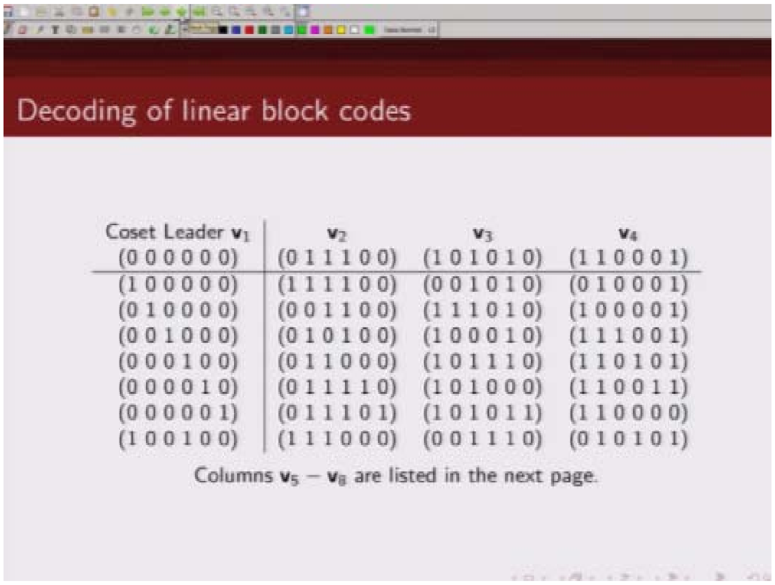
(0 0 0 0 0 0) | (0 1 1 1 0 0) (1 0 1 0 1

			$\mathbf{v}_7$	$\mathbf{v}_8$
(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 standard array, so I can then pick

(Refer Slide Time: 13:40)



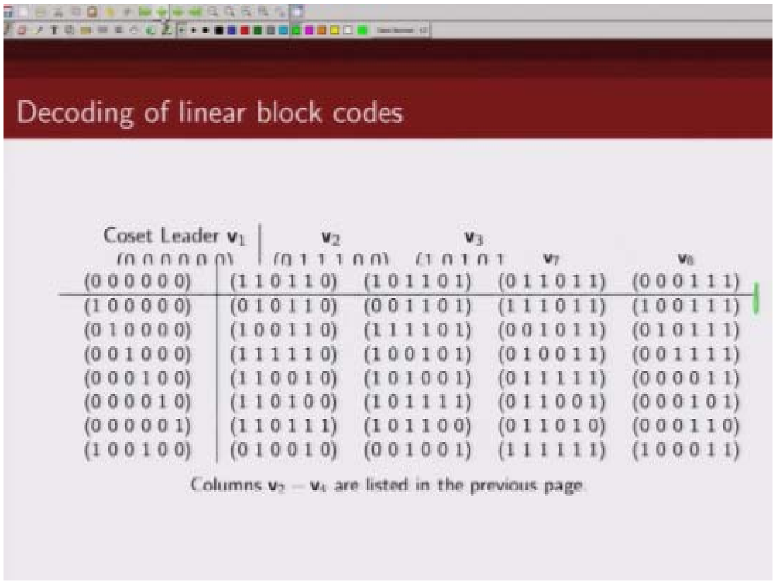
Decoding of linear block codes

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.

This as my n tuple here and then I fill up this whole entry, how? I add this vector to this v_2 put it here, add this vector to this put it here, add this vector to this put it here.

(Refer Slide Time: 14:08)



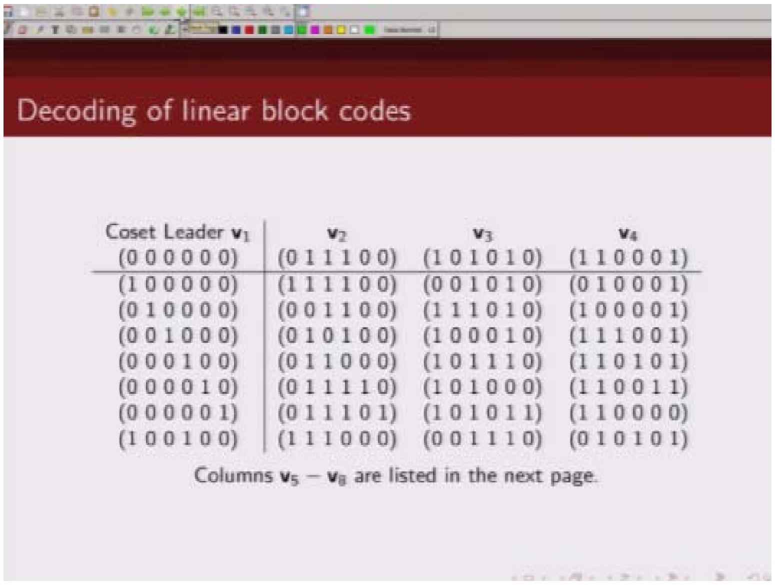
Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1		
(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.

Add this vector to this you can just verify one such entry so 0+1 is 1, 1+0 is 1, 0+1 is 1, 0+1 is 1, 0+0 is 0 and 0+1 is 1, so this is how you populate this entry and we will.

(Refer Slide Time: 14:32)



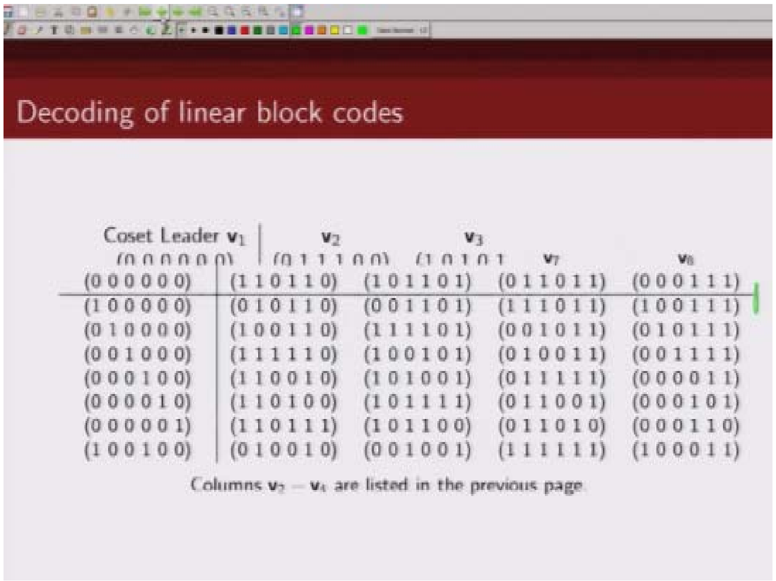
Decoding of linear block codes

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.

Keep on doing it until we have written all those n tuples here, so we have already written this way we have written all the 2 end possibilities, we have written it in this array. Now this array have some interesting properties and we are going to talk about that which we will make use of while decoding our linear block code.

(Refer Slide Time: 15:02)

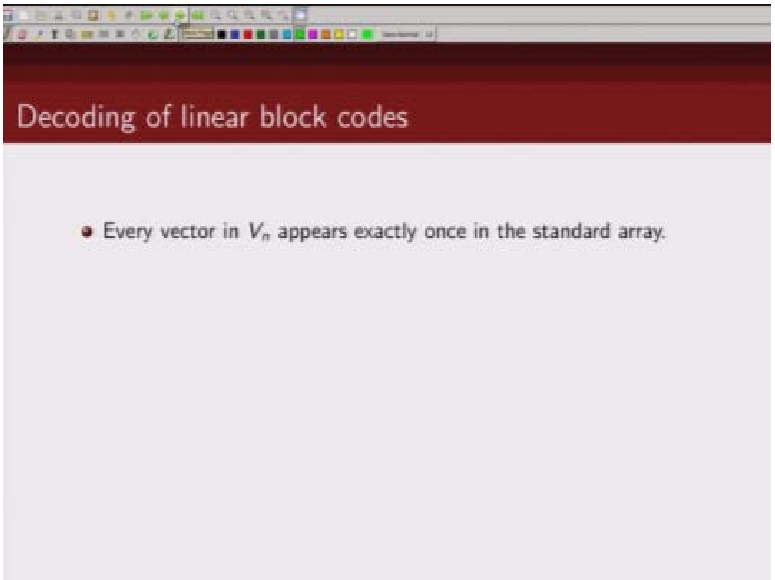


Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1		
(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: 15:03)

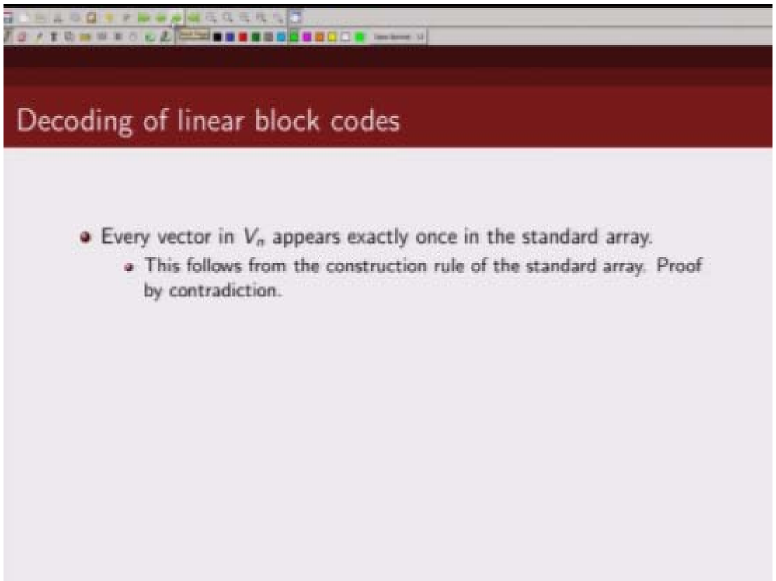


Decoding of linear block codes

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

Every vector in this standard array appears exactly, once this is not very difficult

(Refer Slide Time: 15:11)

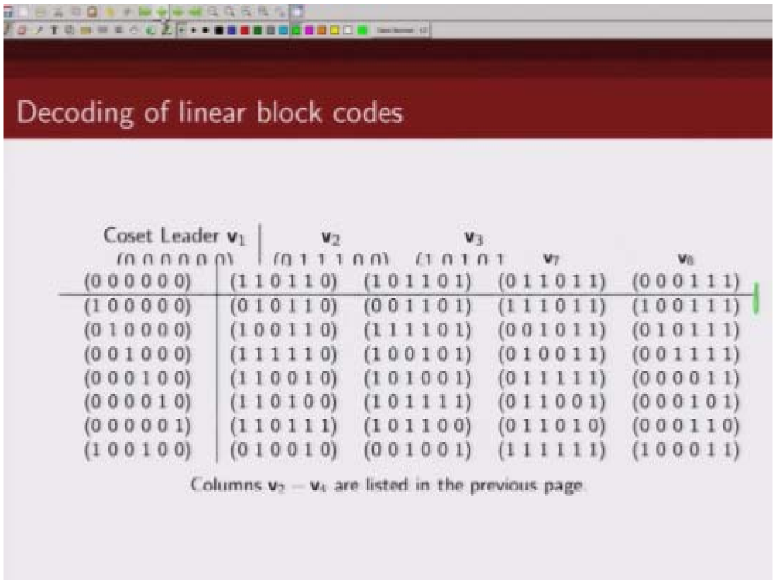


Decoding of linear block codes

- 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.

To prove, this follows from the way we are constructing our standard array okay, and the proof is by contradiction so I will just

(Refer Slide Time: 15:25)

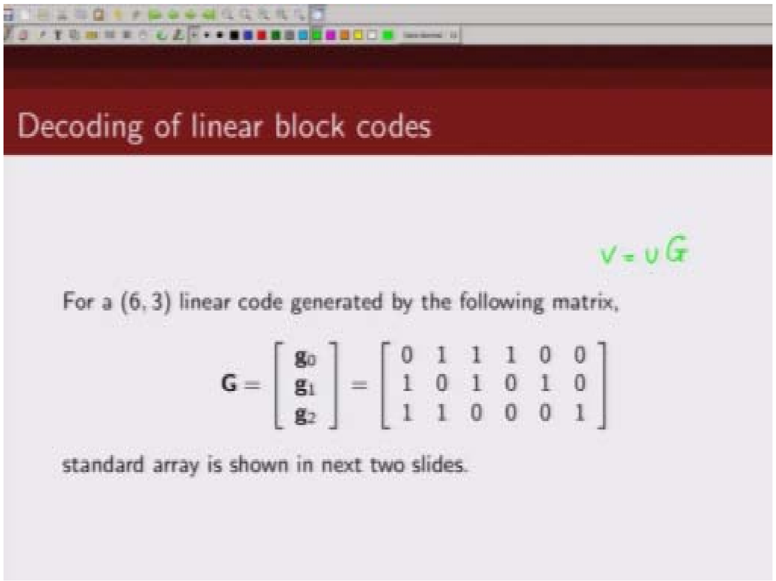


Decoding of linear block codes

Coset Leader $\mathbf{v}_1$ (0 0 0 0 0 0)	$\mathbf{v}_2$ (0 1 1 1 0 0)	$\mathbf{v}_3$ (1 0 1 0 1	$\mathbf{v}_7$	$\mathbf{v}_8$
(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: 15:26)



Decoding of linear block codes

$v = u G$

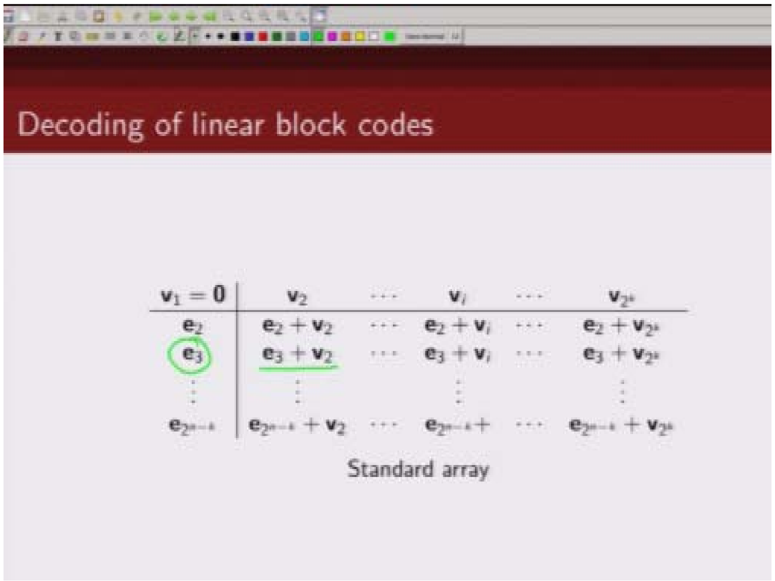
For a (6, 3) linear code generated by the following matrix,

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

standard array is shown in next two slides.

Is a very small proof

(Refer Slide Time: 15:26)



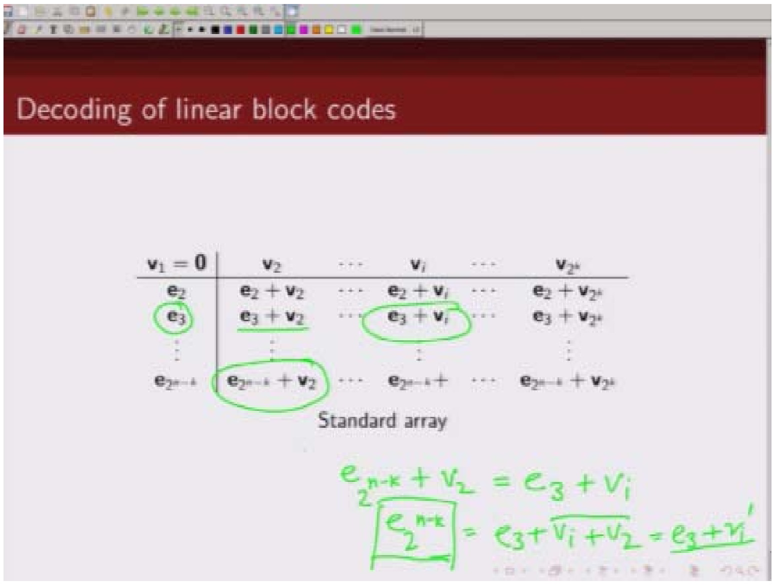
Decoding of linear block codes

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

Standard array

So I can just give you that, let us say so how does the proof by contradiction work? We will assume some thing and then we will show that our assumption is wrong, that is not possible okay so we have to prove that every element.

(Refer Slide Time: 15:43)



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

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

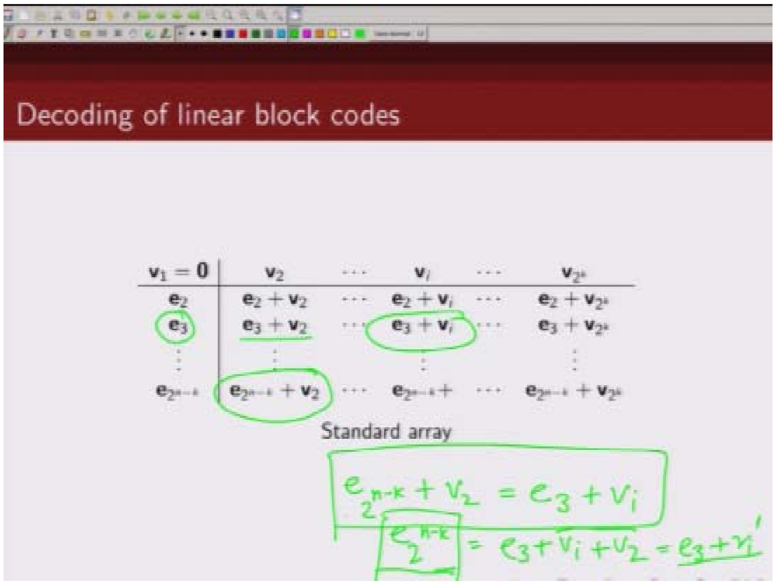
$$\boxed{e_{2^{n-k}}} = e_3 + \overline{v_i + v_2} = \underline{e_3 + v_i'}$$

Here is basically unique so let us say that is not true, let us say two elements let us just call it this element and this element, let us say these two element are same okay, if these two elements are same then we can write this as e $_2$n-k +v_2 to be equal to e_3+ v_i correct. Now we can write this e_2n-k as e_3 + v_i + v_2, why because these are all binary words so basically when we add 1+1 that is basically 0 so we added v_2 to both the sides so v_2 + v_2 will be 0 so we can write this error pattern as e_3 +v_i + v_2, and what is v_i + v_2? v_i + v_2 is another code word, why because that is the property of the linear code word.

Linear block code, so this will be e_3+ some other code word let us call it v_i, v_i´, so this error pattern e_2 n-k is e_3+ v´. Now e_3 + v_i´ 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 code words v_3 and that is what these entries are so e_3 + v_i ´ should have been some entry here, what does that mean, that means we made a mistake in selecting this error pattern.

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 that this error pattern is already there in the row containing e_3, so that means we cannot choose that as our error pattern

(Refer Slide Time: 18:29)



Decoding of linear block codes

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

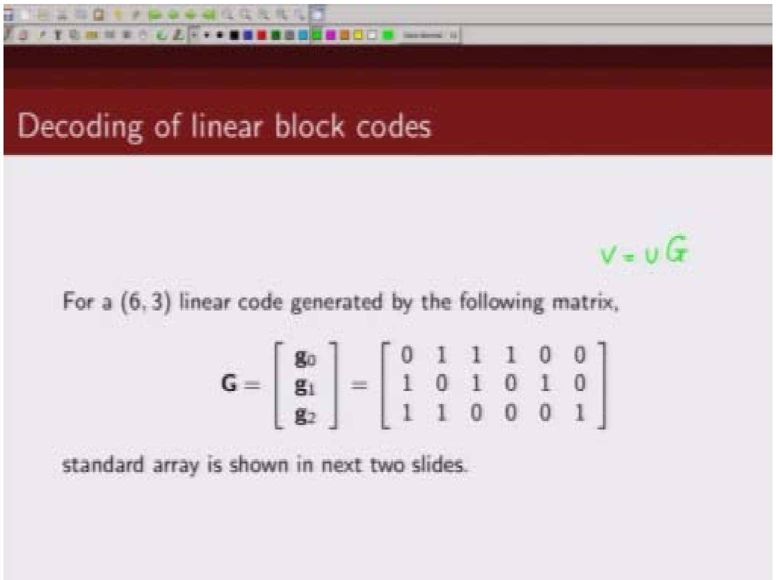
Standard array

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

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

Here because we can only choose an error pattern which has not appeared before hand, so in other words we are contradicting our self, on one hand we are saying we are picking up these error patterns such a way that they have not appeared in the previous rows but if we are saying the 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 a_2n-k as this because this has already appeared, hence this condition that these two elements are same is incorrect okay so all the elements

(Refer Slide Time: 19:19)



Decoding of linear block codes

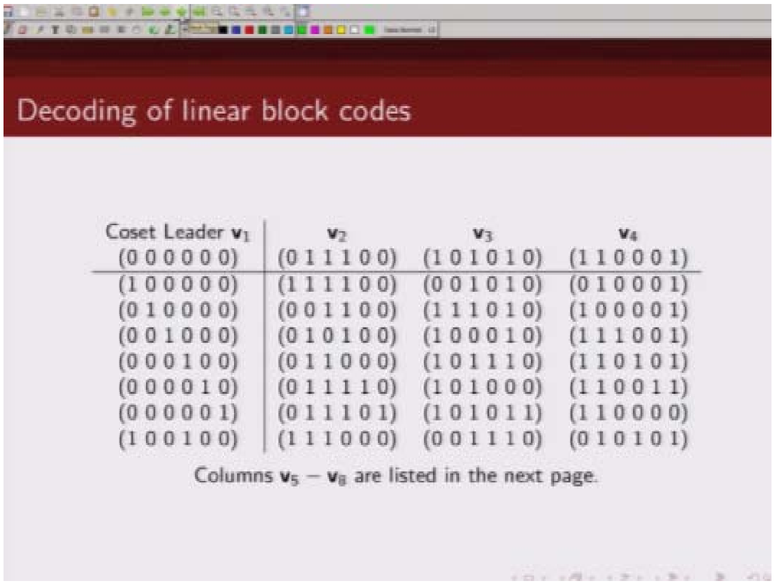
$v = uG$

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.

(Refer Slide Time: 19:20)

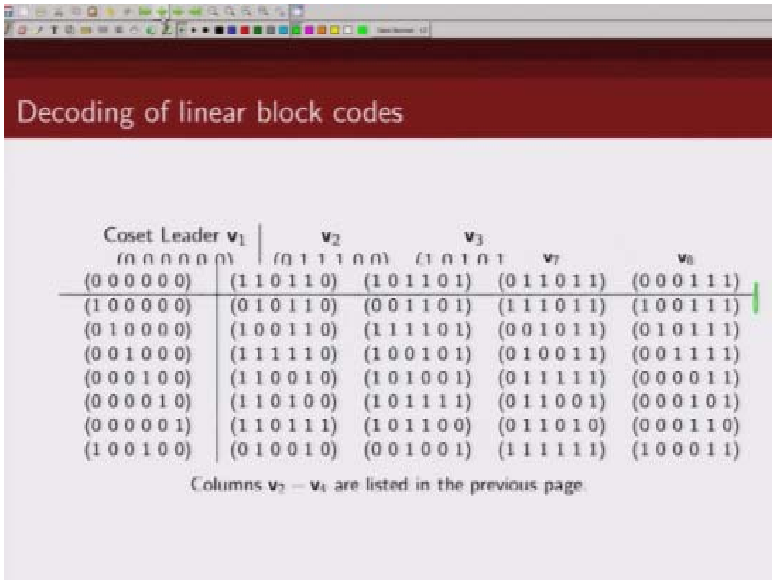


Decoding of linear block codes

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: 19:20)

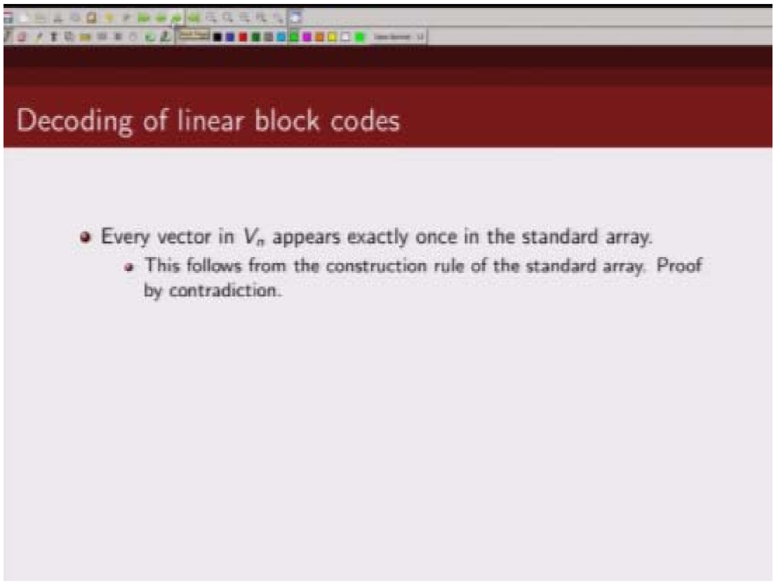


Decoding of linear block codes

Coset Leader $\mathbf{v}_1$	$\mathbf{v}_2$	$\mathbf{v}_3$	$\mathbf{v}_7$	$\mathbf{v}_8$
(0 0 0 0 0 0)	(0 1 1 1 0 0)	(1 0 1 0 1		
(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: 19:22)

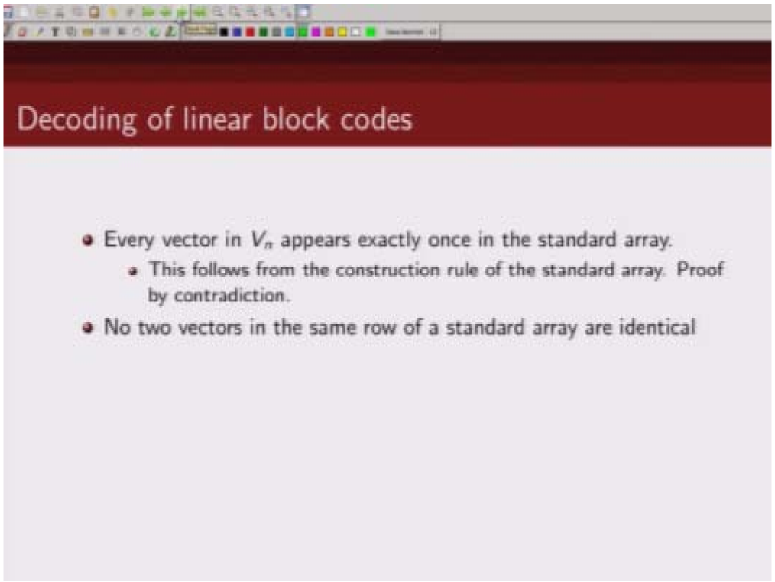


Decoding of linear block codes

- 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.

Of this standard array are distinct and they appear exactly once.

(Refer Slide Time: 19:28)

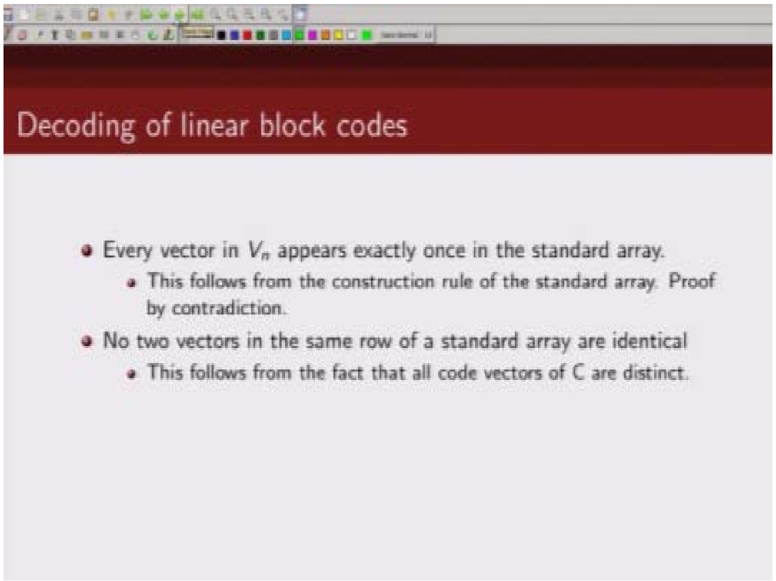


Decoding of linear block codes

- 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

No two vectors in the same row are identical, again this is.

(Refer Slide Time: 19:28)

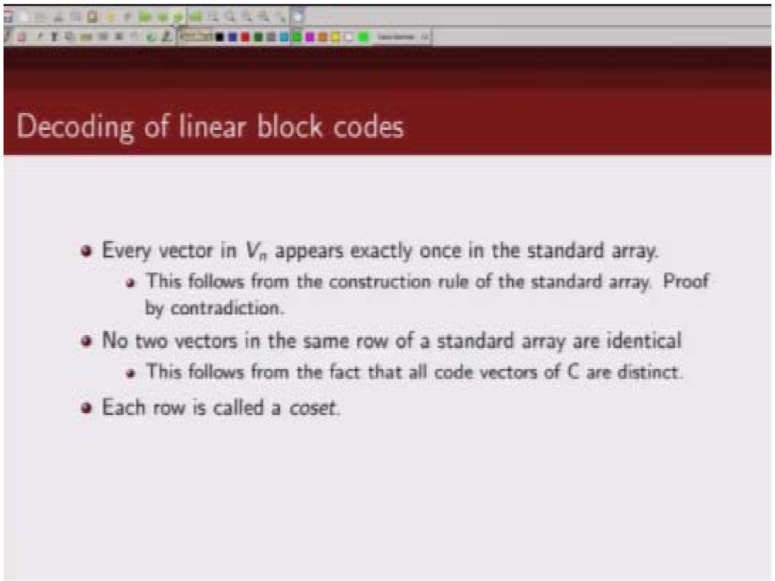


Decoding of linear block codes

- 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.

Very easy to prove because all code words are distinct so elements in one row will all be distinct.

(Refer Slide Time: 19:48)

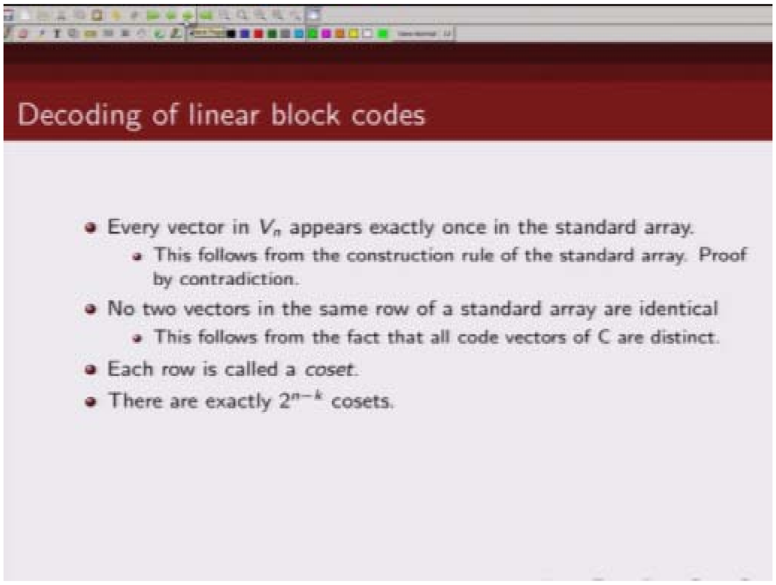


Decoding of linear block codes

- 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 this standard array as coset and the left.

(Refer Slide Time: 19:53)

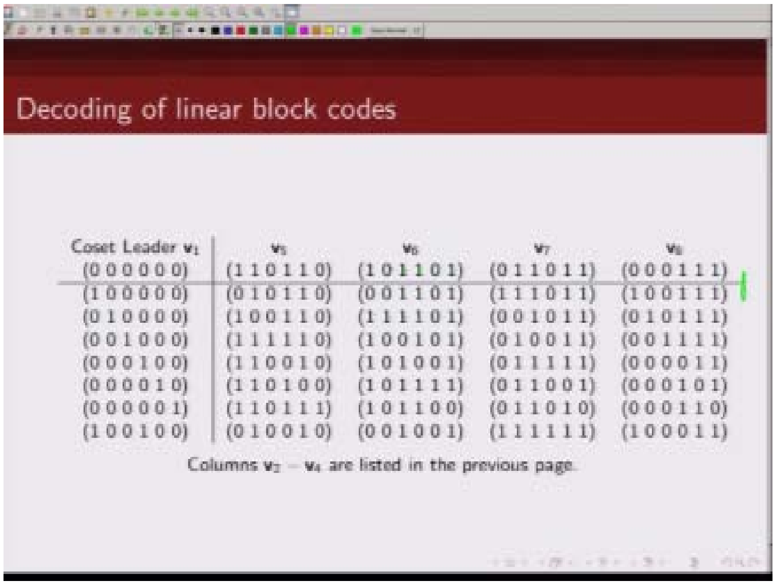


Decoding of linear block codes

- 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.

Most entry of each coset or row is known as coset leader and we have total 2^{n-k} such cosets. Now the question is can we make any other element as our coset leader, so if you just go back here.

(Refer Slide Time: 20:24)



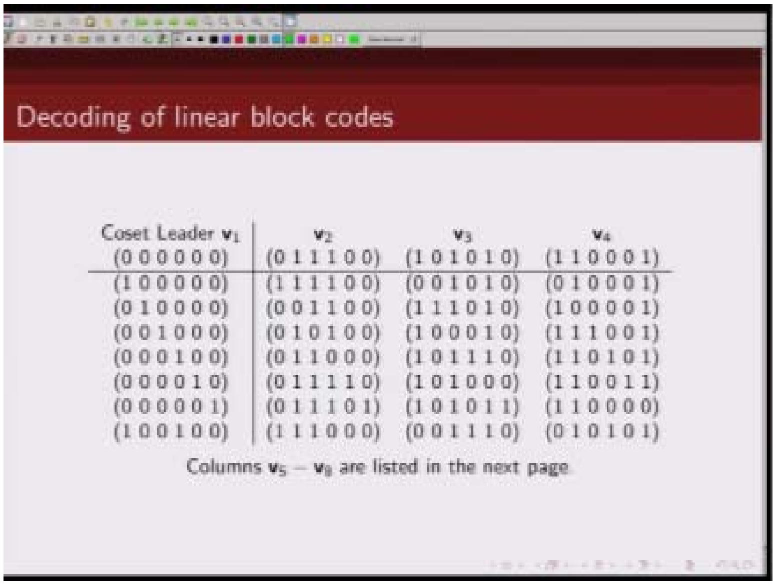
Decoding of linear block codes

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.

Let us say look at this row, coset leader was this right. The left more centre in this standard array, now what happens if we instead of choosing this is as a coset leader if we had chosen let us say this as coset leader, would it have changed our elements of this array no, why?

(Refer Slide Time: 20:50)



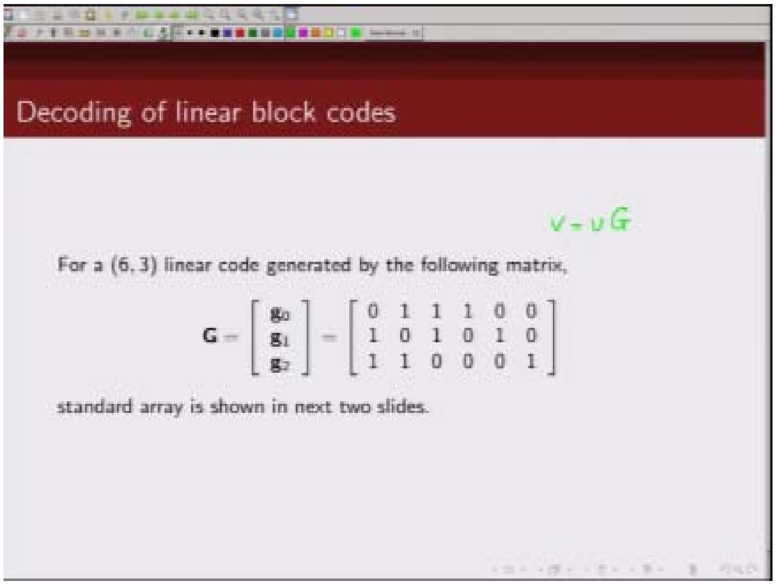
Decoding of linear block codes

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

If you go back.

(Refer Slide Time: 20:51)



Decoding of linear block codes

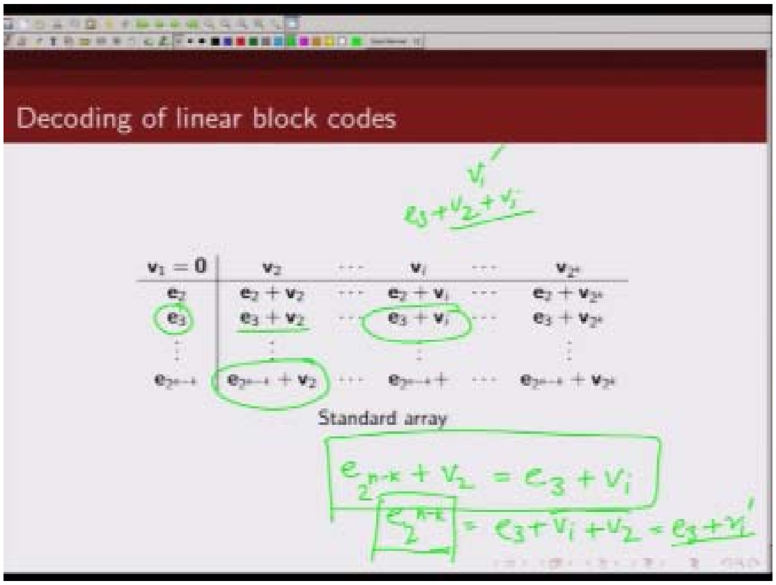
$v = uG$

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.

(Refer Slide Time: 20:52)



Decoding of linear block codes

$$\overset{v_i'}{e_3 + \underline{v_2 + v_i}}$$

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

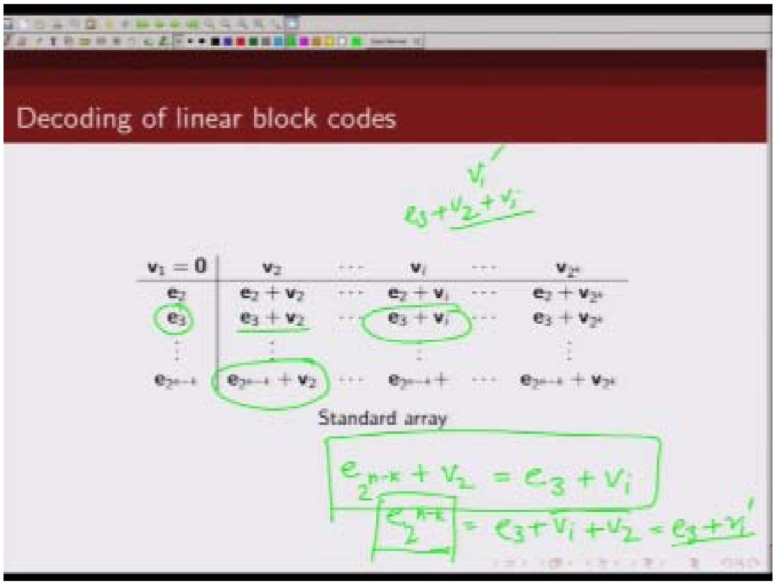
Standard array

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

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

And see the entries of a particular row; here the coset leader was e3, if instead of v3 we would have use e3 + v2 what would have happened, this would have e3 + v2 this would have been e3 + v2+ v2 so this would be e3, this would be e3 + v2 + vi and v2+ vi would be another code word vi', so this would have been some other code word so the elements in each row would have remained the same only they would have just got reordered okay.

(Refer Slide Time: 21:32)



Decoding of linear block codes

v_i'

$e_3+\underline{v_2+v_i}$

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

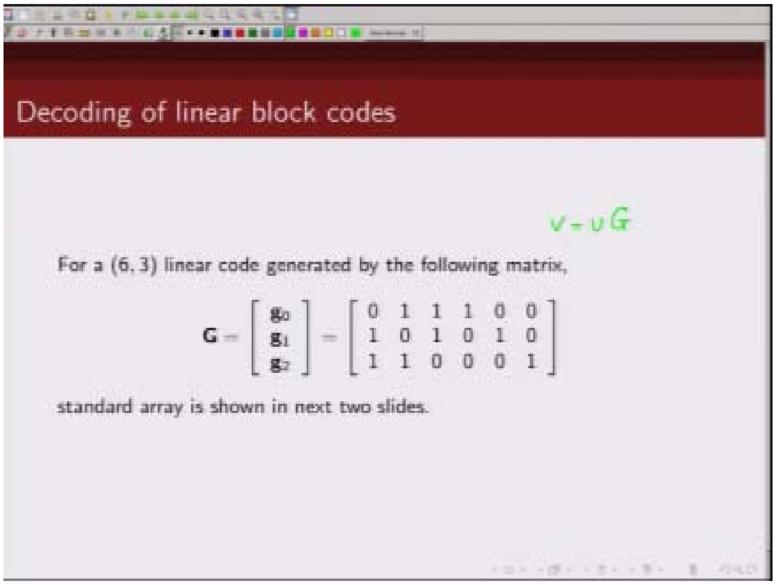
Standard array

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

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

So if we pick any other element in the row as coset leader it does not change the elements in a coset.

(Refer Slide Time: 21:42)



Decoding of linear block codes

$v = uG$

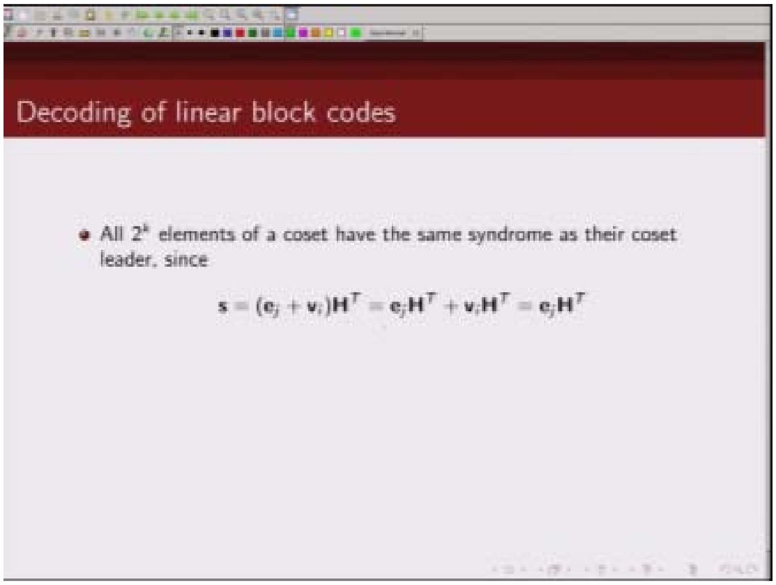
For a (6, 3) linear code generated by the following matrix,

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

standard array is shown in next two slides.

Or in a row.

(Refer Slide Time: 21:44)



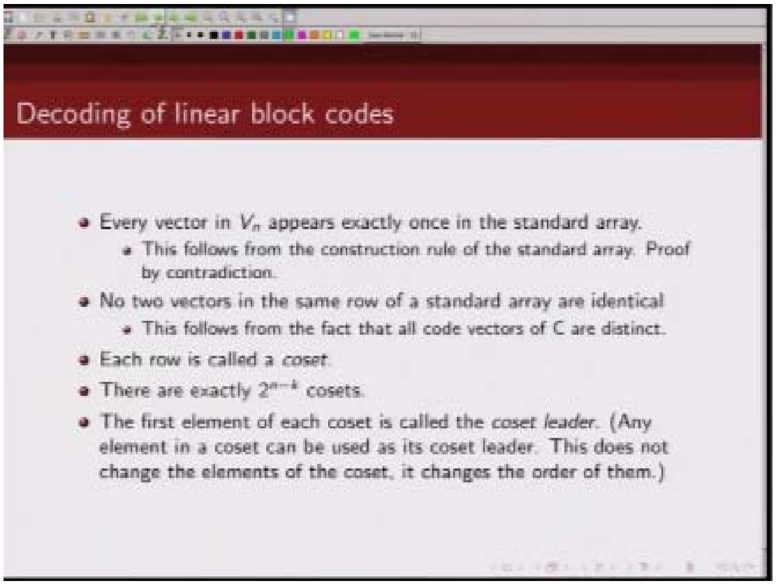
Decoding of linear block codes

- 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 next property is all the 2k elements in a row or in a coset have the same syndrome, this we can show because each element in a coset are of the form like this e_j + v_i H transpose and since v_iH transpose is 0 they will only depend on the error pattern and in each row basically it is the same error pattern that appears in the elements of the row.

(Refer Slide Time: 22:21)

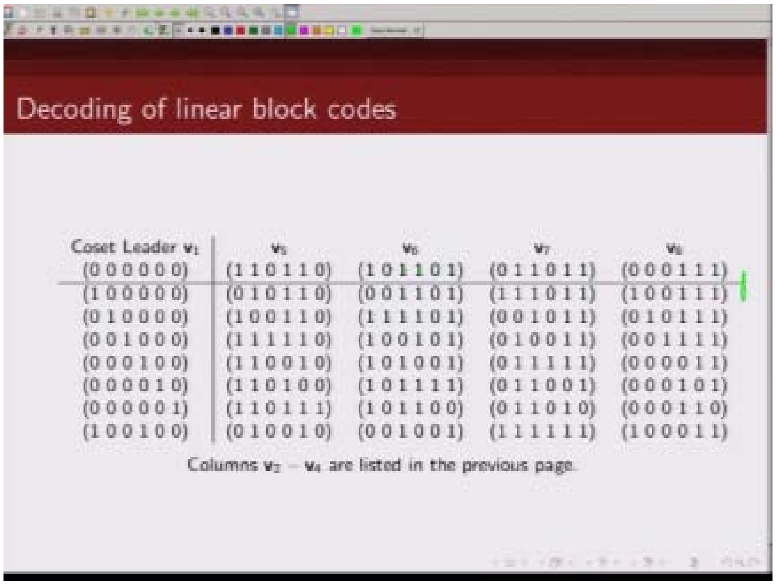


Decoding of linear block codes

- 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.)

We can again go back to our diagram for standard array.

(Refer Slide Time: 22:24)

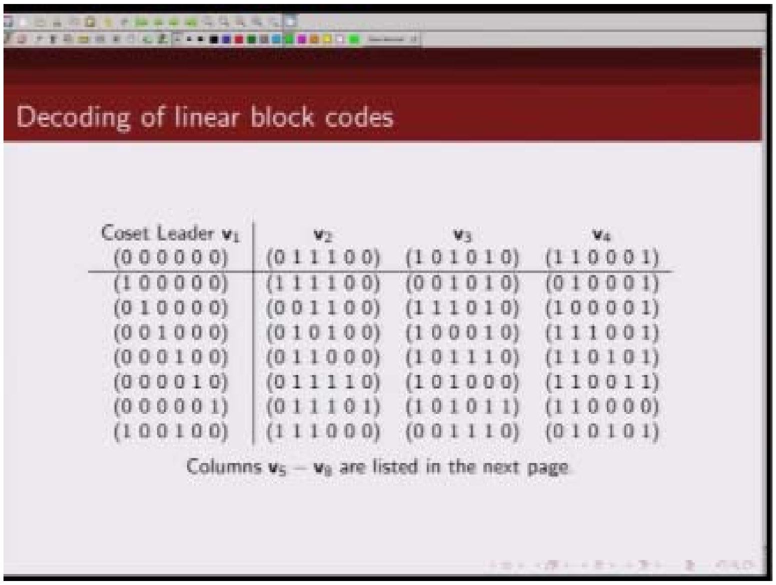


Decoding of linear block codes

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: 22:25)

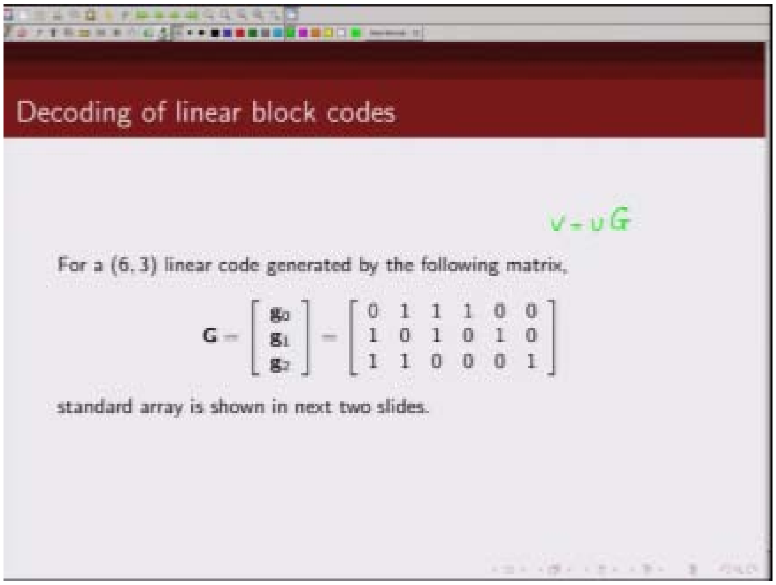


Decoding of linear block codes

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

(Refer Slide Time: 22:25)



Decoding of linear block codes

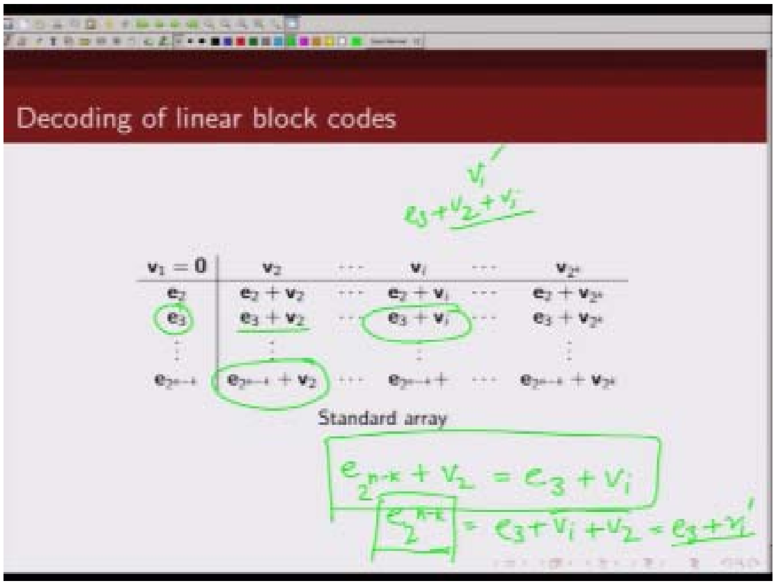
$v = uG$

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.

(Refer Slide Time: 22:26)



Decoding of linear block codes

v_i'

$e_3+\underline{v_2+v_i}$

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

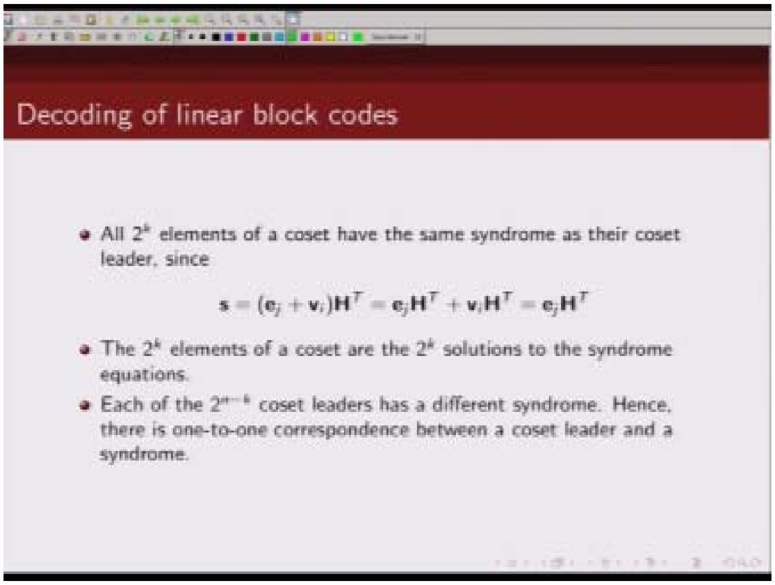
Standard array

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

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

And we can see this, we can see an each row, in this row is e3, in this row all these elements have e2, so they will have the same syndrome.

(Refer Slide Time: 22:39)



Decoding of linear block codes

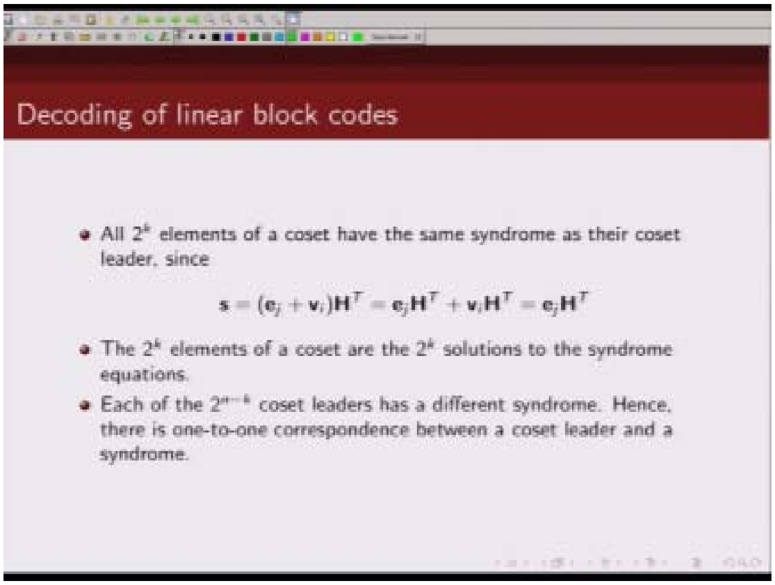
- All 2^k elements of a coset have the same syndrome as their coset leader, since

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

- 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.

And in the previous lecture we talked about that there are n – k syndrome equations and n unknowns and there are total 2k solutions and you can see here.

(Refer Slide Time: 22:54)



Decoding of linear block codes

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

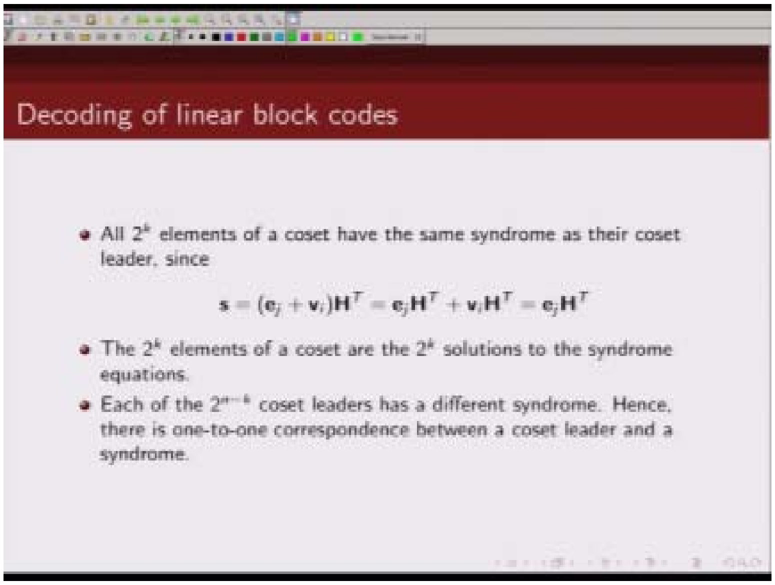
$$\mathbf{s} = (\mathbf{e}_j + \mathbf{v}_i)\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T + \mathbf{v}_i\mathbf{H}^T = \mathbf{e}_j\mathbf{H}^T$$

- 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.

Each row has 2k elements and they are the same syndrome so these 2k elements are exactly the solution of your syndrome equations because they, these 2k elements of a row, of a coset they all have the same syndrome and they corresponds to the 2k solutions of the syndrome equations, so the 2k elements of a coset are actually the solutions of your syndrome equation, and another interesting thing is each of these cosets or each of this 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 a different error pattern, again let us go back.

(Refer Slide Time: 23:52)



Decoding of linear block codes

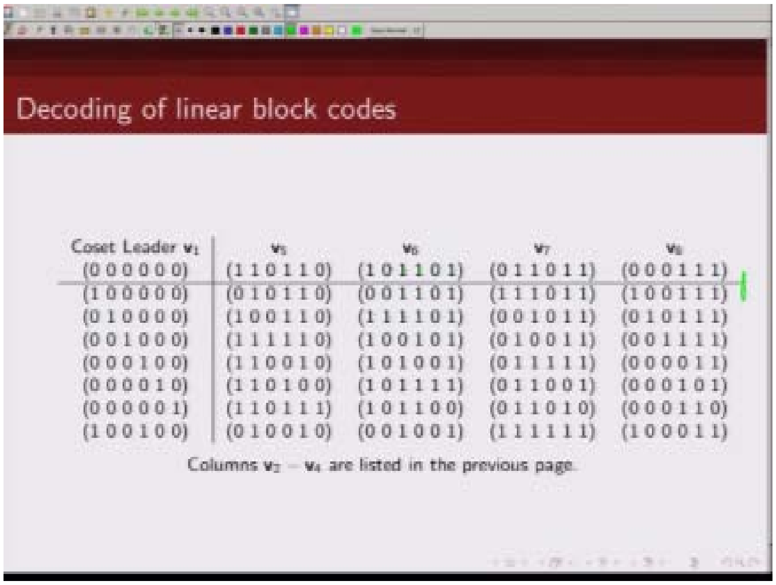
- All 2^k elements of a coset have the same syndrome as their coset leader, since

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

- 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.

To the diagram that we had for the standard array.

(Refer Slide Time: 23:56)

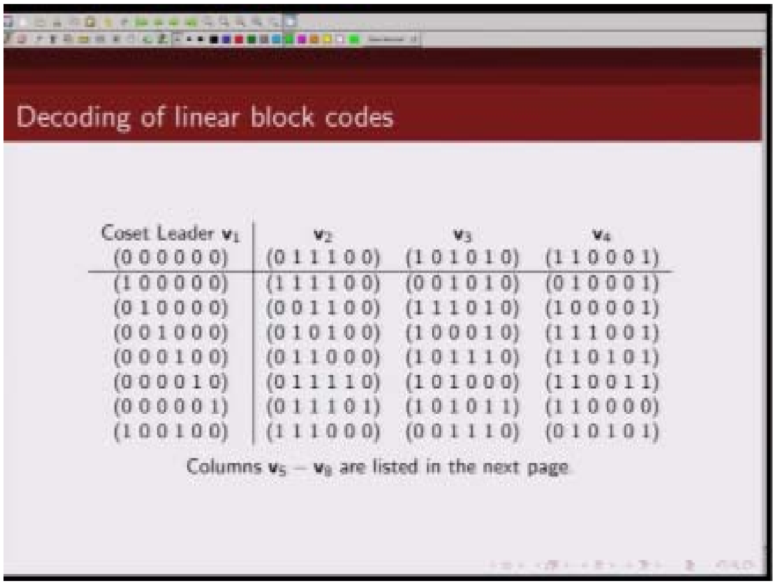


Decoding of linear block codes

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.

(Refer Slide Time: 23:56)

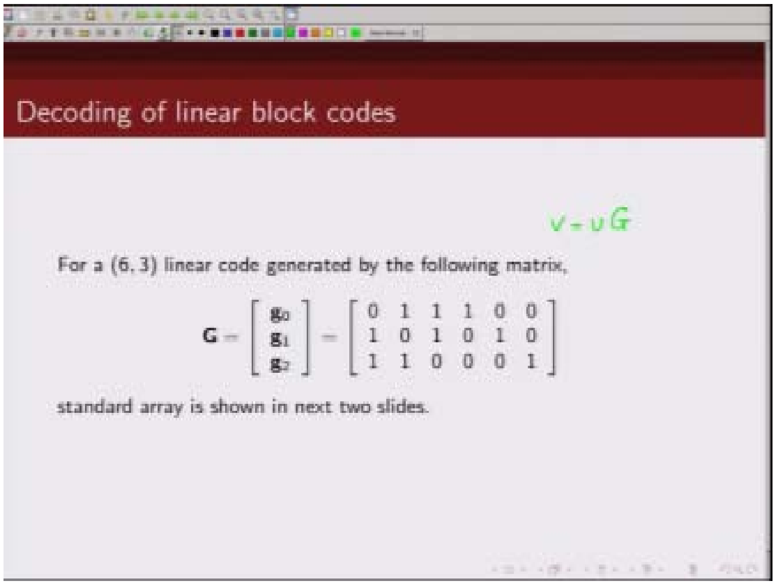


Decoding of linear block codes

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

(Refer Slide Time: 23:57)



Decoding of linear block codes

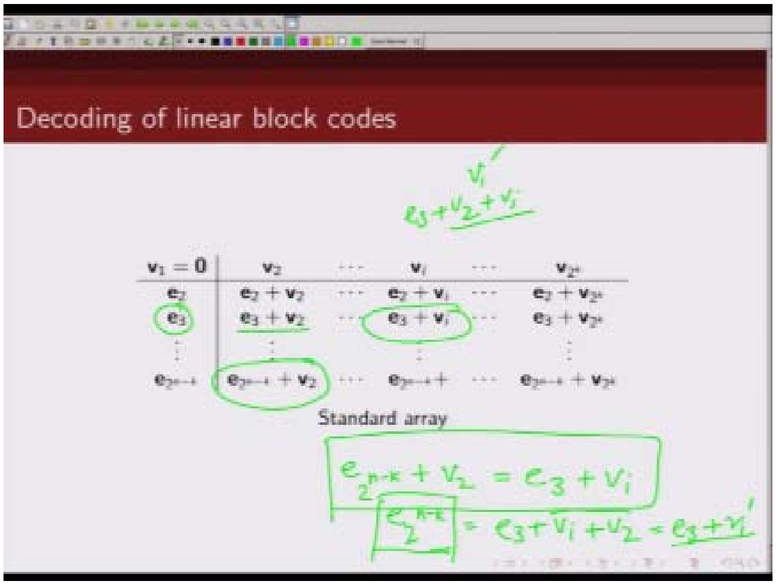
$v = uG$

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.

(Refer Slide Time: 23:57)



Decoding of linear block codes

v_i'

$e_3+\underline{v_2+v_i}$

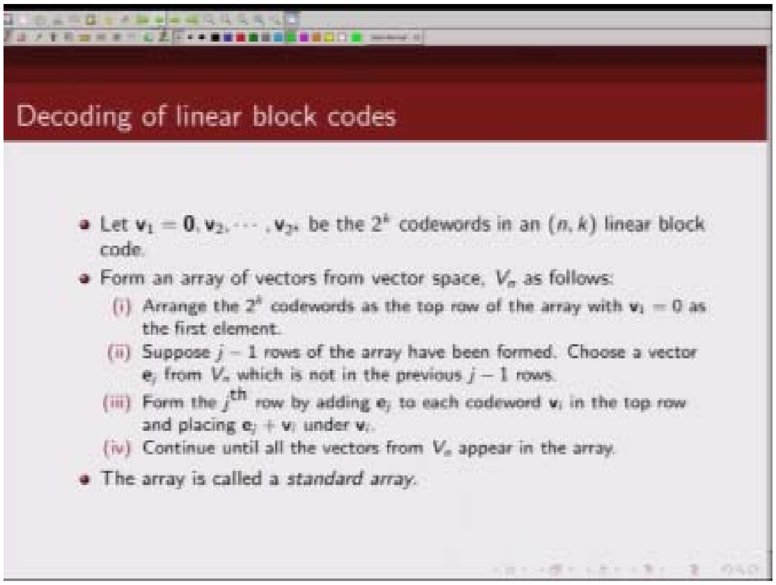
$\mathbf{v}_1 = \mathbf{0}$	$\mathbf{v}_2$	$\cdots$	$\mathbf{v}_i$	$\cdots$	$\mathbf{v}_{2^k}$
$\mathbf{e}_2$	$\mathbf{e}_2 + \mathbf{v}_2$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_i$	$\cdots$	$\mathbf{e}_2 + \mathbf{v}_{2^k}$
$\mathbf{e}_3$	$\underline{\mathbf{e}_3 + \mathbf{v}_2}$	$\cdots$	$\mathbf{e}_3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

Standard array

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

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

(Refer Slide Time: 23:58)

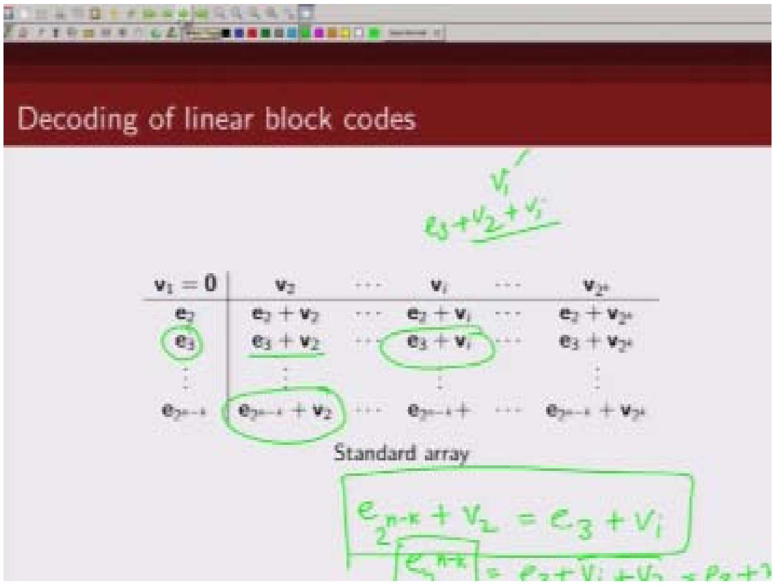


Decoding of linear block codes

- 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*.

This row.

(Refer Slide Time: 23:59)



Decoding of linear block codes

v_i

$e_3 + v_2 + v_i$

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

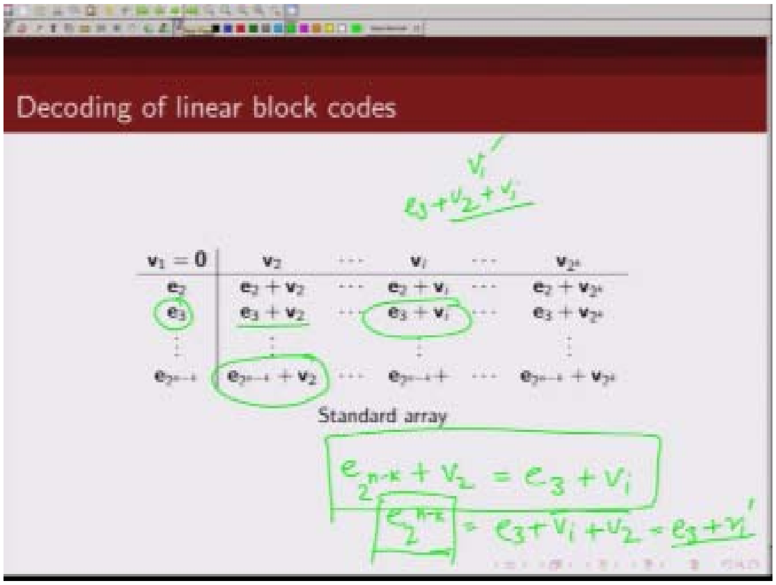
Standard array

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

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

Is related to e2, so if we compute syndrome for any of these receive factors we will get syndrome corresponds to e2, this row corresponds.

(Refer Slide Time: 24:13)



Decoding of linear block codes

v_i'

$\underline{e_3+v_2+v_i}$

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

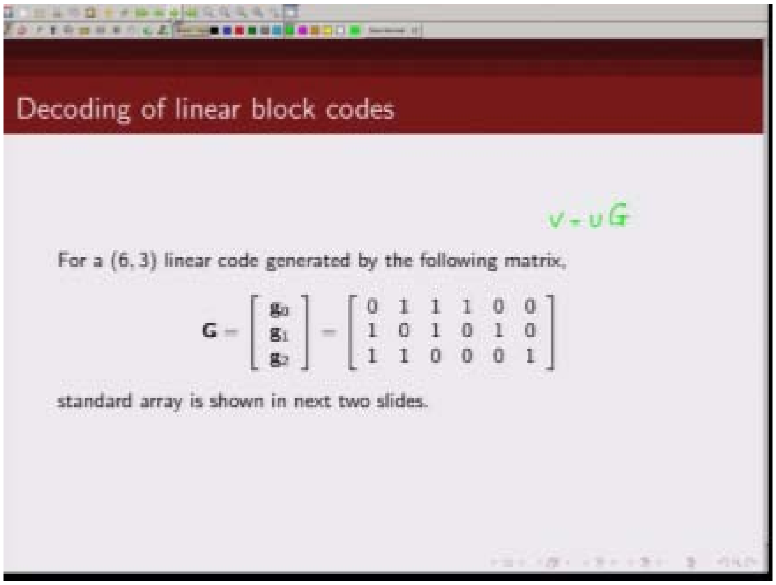
Standard array

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

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

To e3 this e4, this one e is plot to n – k 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 another words you can map one syndrome to one row or you can map one syndrome to one coset.

(Refer Slide Time: 24:44)



Decoding of linear block codes

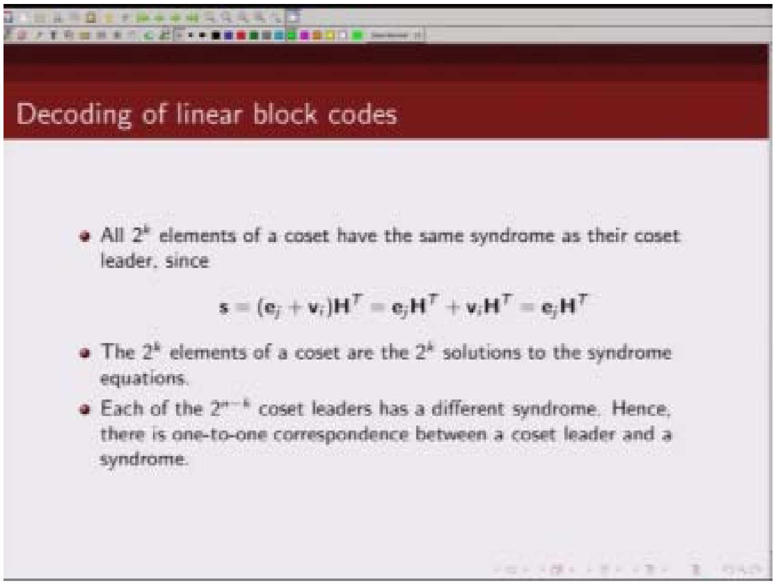
$v = uG$

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.

(Refer Slide Time: 24:45)



Decoding of linear block codes

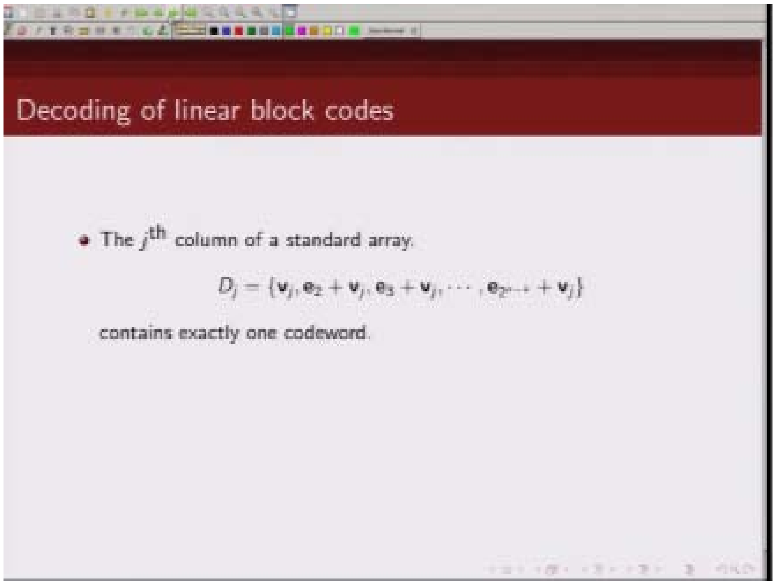
- All 2^k elements of a coset have the same syndrome as their coset leader, since

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

- 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.

So this is interesting that now we have one to one mapping between each cosets or each coset leader to a syndrome.

(Refer Slide Time: 24:59)



Decoding of linear block codes

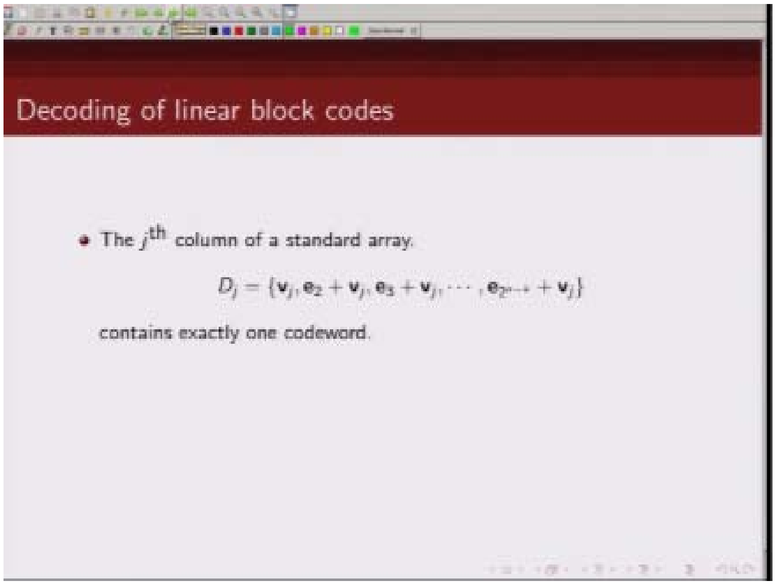
- The j^{th} column of a standard array

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

contains exactly one codeword.

Another thing to note is if you look at columns of these standard array each column of this standard array corresponds to one particular code word. If you recall when you started the lecture we said we want to partition our 2n vectors into 2k different sets and each of this 2k set should corresponds.

(Refer Slide Time: 25:25)



Decoding of linear block codes

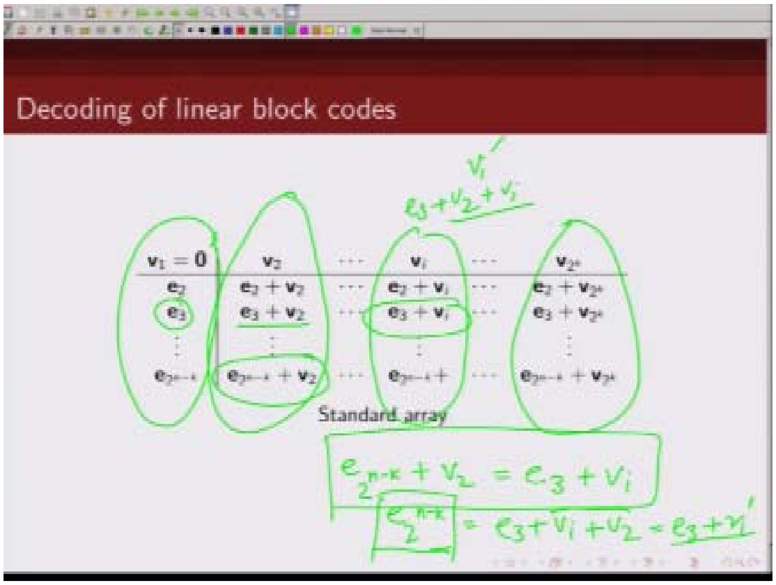
- The j^{th} column of a standard array.

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

contains exactly one codeword.

To only one code word and this is what is happening here as well.

(Refer Slide Time: 25:32)



Decoding of linear block codes

v_i'

$e_3 + v_2 + v_i$

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

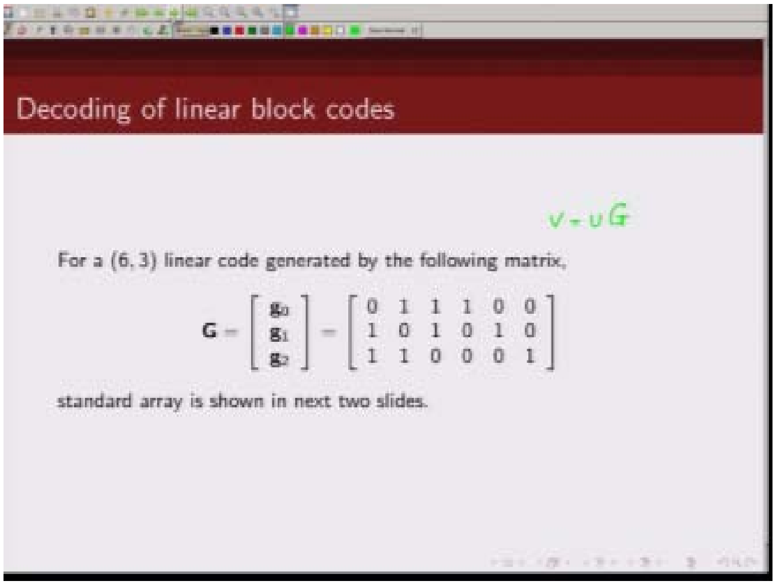
Standard array

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

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

You have this whole thing as total possible to n vectors, now we have already partitioned them into 2k different partitions and these are all distinct partitions, there is no element in here which is common with this element and another thing to be of interest, if you look at each of these partition, this partition has all zero vector, this partition corresponds to v2, this partition corresponds to Vi, this partition corresponds to v2k.

(Refer Slide Time: 26:11)



Decoding of linear block codes

$v = uG$

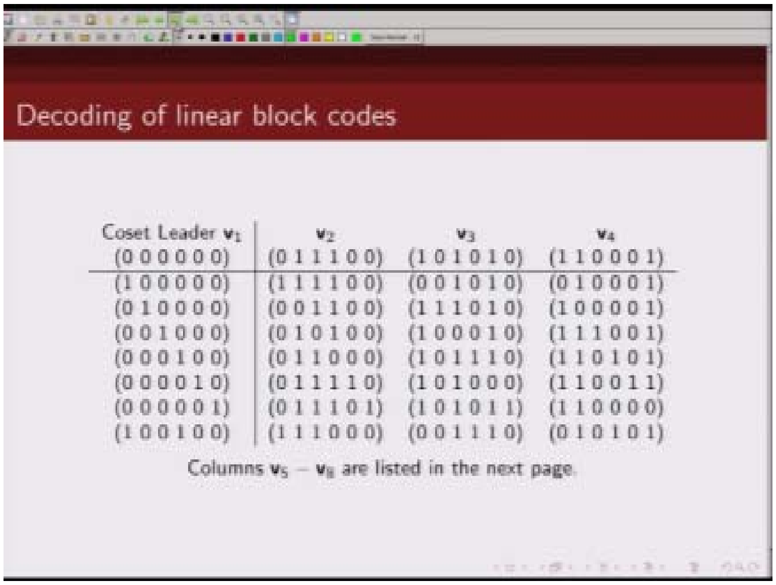
For a (6, 3) linear code generated by the following matrix,

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

standard array is shown in next two slides.

So this is the.

(Refer Slide Time: 26:12)



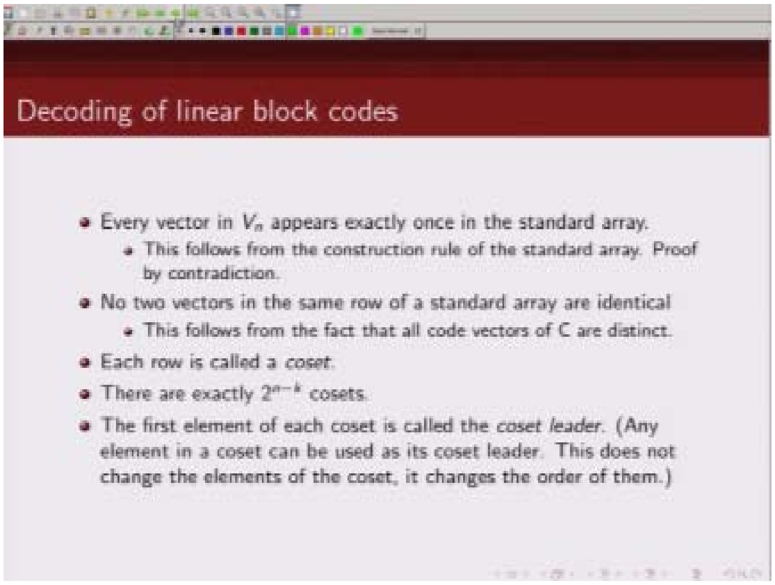
Decoding of linear block codes

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.

Point basically I am trying to make that.

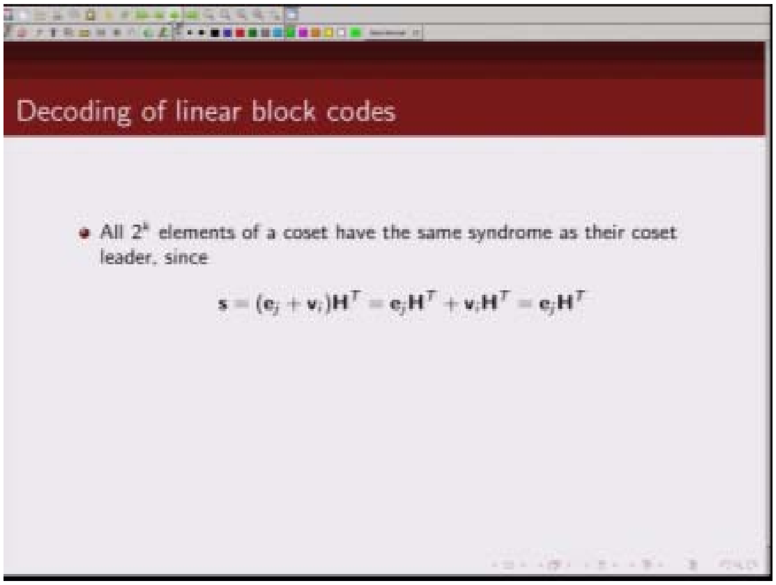
(Refer Slide Time: 26:14)



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.
- 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.)

(Refer Slide Time: 26:16)



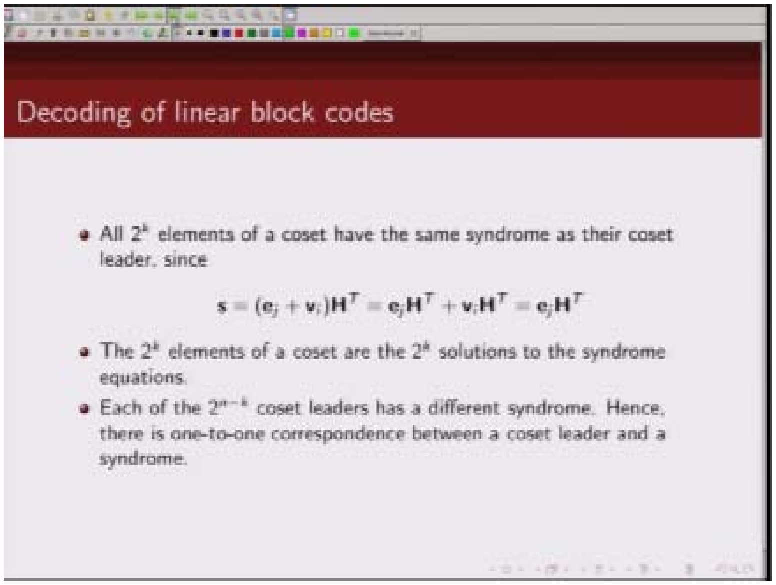
Decoding of linear block codes

- 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 way we have.

(Refer Slide Time: 26:16)



Decoding of linear block codes

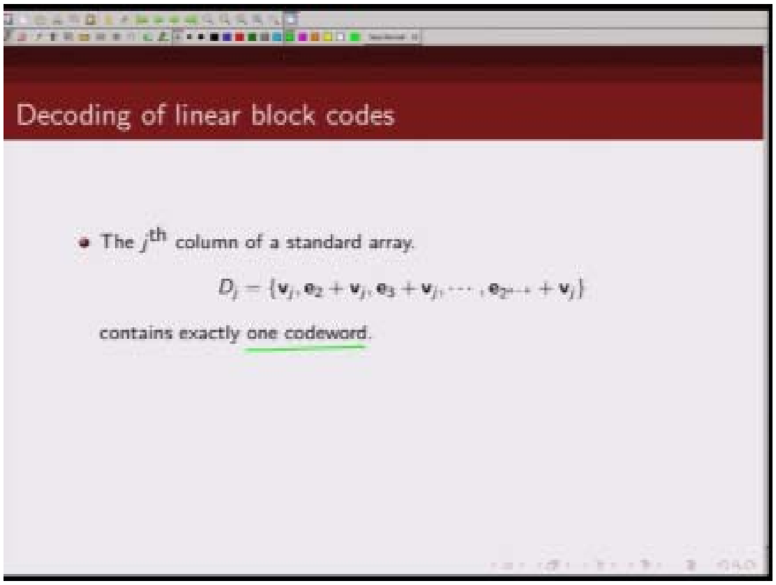
- All 2^k elements of a coset have the same syndrome as their coset leader, since

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

- 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.

Created the standard array.

(Refer Slide Time: 26:19)



Decoding of linear block codes

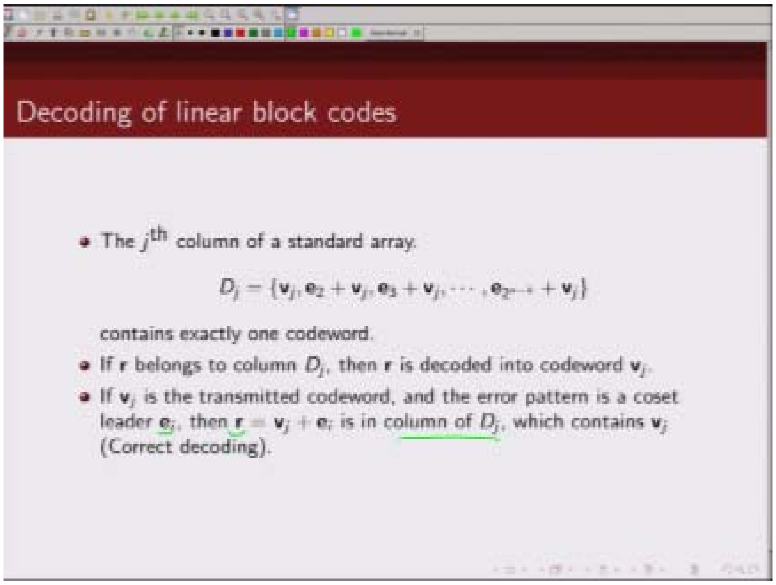
- The j^{th} column of a standard array.

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

contains exactly one codeword.

If you look at the jth column of the standard array it contains exactly one code word, it corresponds to only one particular code word.

(Refer Slide Time: 26:32)



Decoding of linear block codes

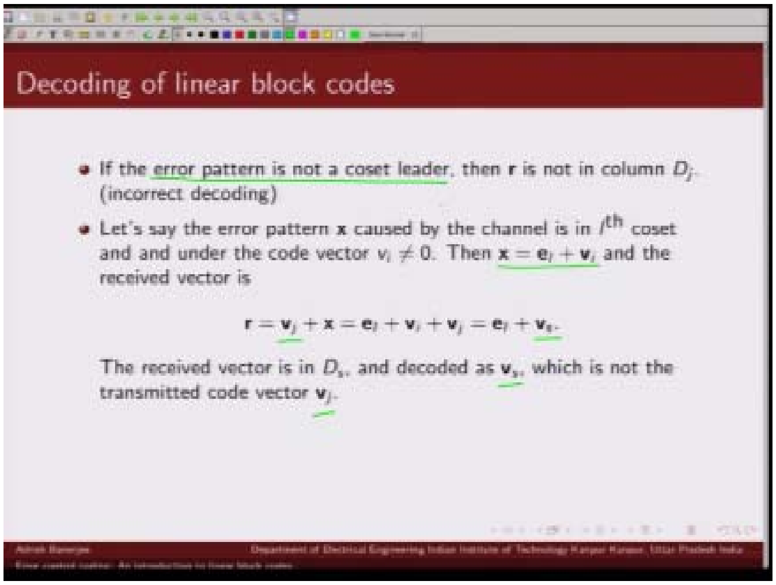
- The j^{th} column of a standard array

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

  contains exactly one codeword.
- If $\mathbf{r}$ 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).

Now if they receive code word belongs to this column D_j then r will be decoded as code word $v_{j,}$ so whenever r belongs to a set this we will decode this r as correspond to code word $v_{j.}$ So if v_j is our transmit code word and the error pattern is e_i and if they receive sequence is in falls in the column D_j then we would decode it as v_j which is correct decoding. We would not make any error, however if the error pattern is not a coset leader then.

(Refer Slide Time: 27: 25)



Decoding of linear block codes

- If the error pattern is not a coset leader, then **r** is not in column D_j. (incorrect decoding)
- Let's say the error pattern **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 $\mathbf{v}_j$.

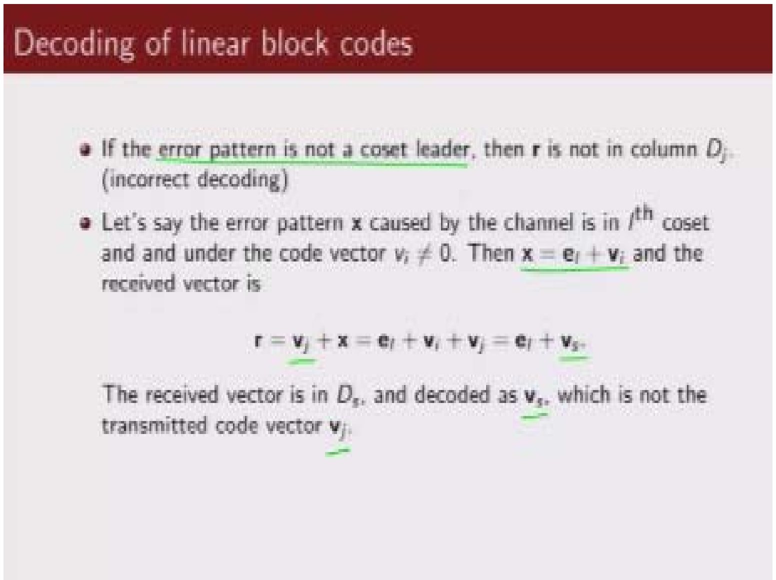
Adrish Banerjee Department of Electrical Engineering Indian Institute of Technology Kanpur Kanpur, Uttar Pradesh India

Error control coding: An introduction to linear block codes

r will not be in column Dj and in that case we will make an error in decoding, so let us look at an error pattern x caused by a channel and it is in the L_{th} coset , so if it is in the L^{th} code coset we are writing this as let us say this error pattern has e_l corresponding to the error pattern $e_{l + vi}$ so in that case if we are transmitting code word vj and we encountered this error pattern what we would receive is v_j + x this would be nothing but el + v_i + $v_{j,}$ now what is v_i + v_j, v_i + v_j sum of two code words is also a valid code word.

Let us call that code word as v_s so now the receive vector is in partition Ds and in this case what are we going to decode it as, we are going to decoded it as v_s which is not same as the transmitted code word vj so that is what I meant, if your error pattern is not a coset leader then r would not be in the same column as D_j if this would have been coset leader this would have been just e_l and then receive sequence would have been just e_l + v_j so this would have still remained in the partition D_j and we would not have made a mistake okay. So if the error pattern is not a coset leader.

(Refer Slide Time: 29:19)



Decoding of linear block codes

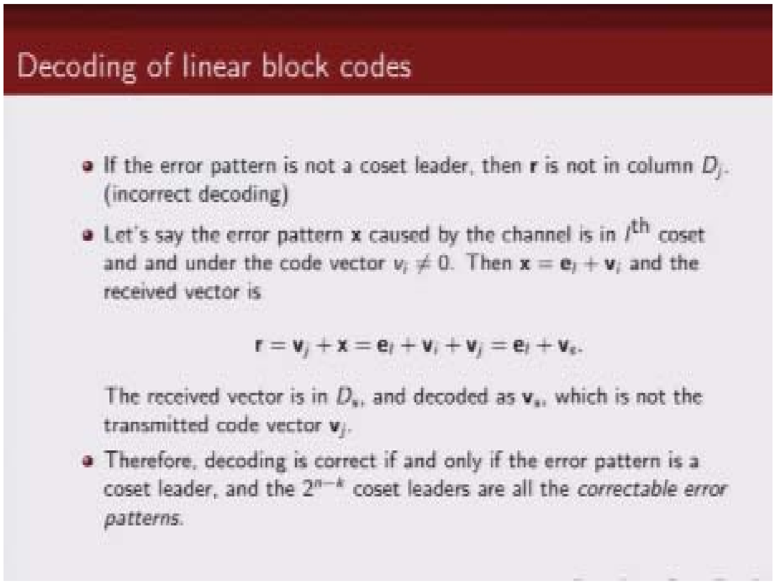
- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and and under the code vector $v_i \neq 0$. Then $\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i$ and the received vector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

Then we are basically going to make an error.

(Refer Slide Time: 29:24)



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

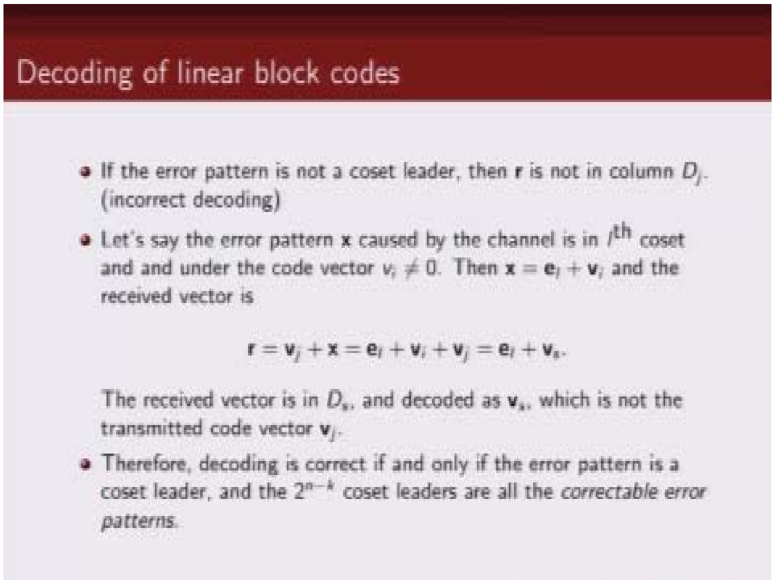
$$\mathbf{r} = \mathbf{v}_j + \mathbf{x} = \mathbf{e}_l + \mathbf{v}_i + \mathbf{v}_j = \mathbf{e}_l + \mathbf{v}_s.$$

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

(Refer Slide Time: 29:39)



Decoding of linear block codes

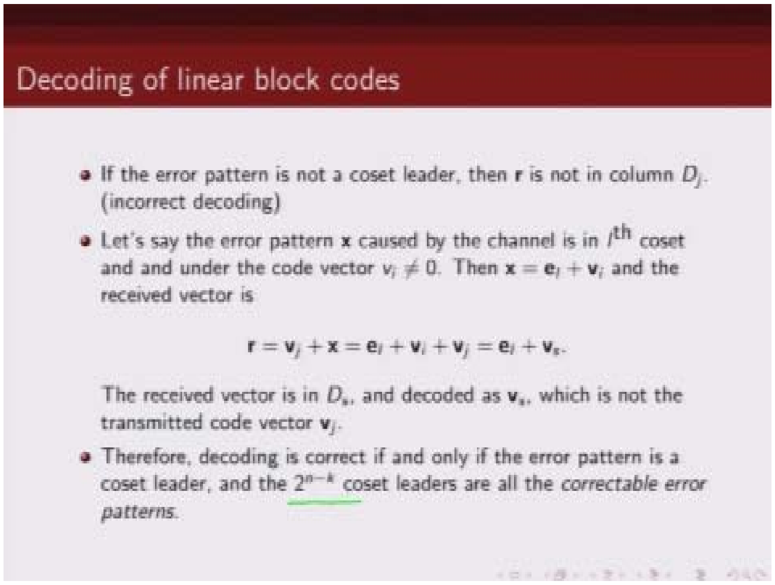
- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and and under the code vector $v_i \neq 0$. Then $\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i$ and the received vector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

(Refer Slide Time: 29:43)



Decoding of linear block codes

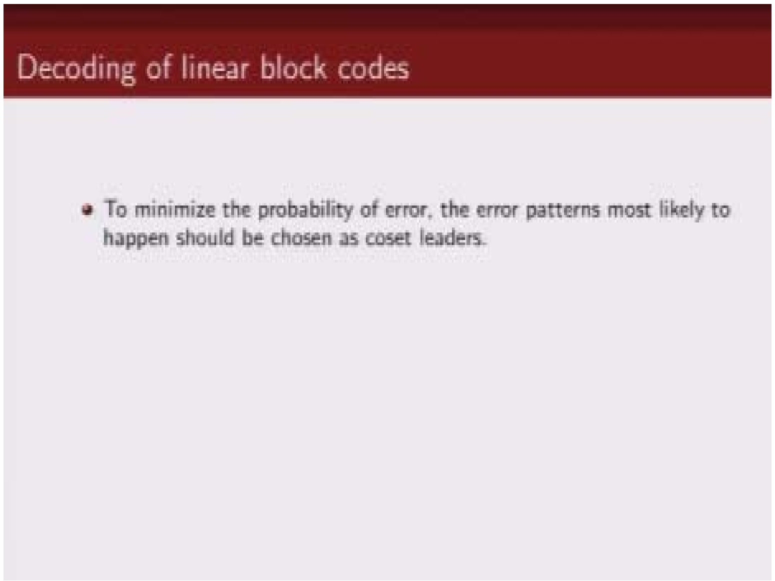
- If the error pattern is not a coset leader, then $\mathbf{r}$ is not in column D_j. (incorrect decoding)
- Let's say the error pattern $\mathbf{x}$ caused by the channel is in l^{th} coset and and under the code vector $v_i \neq 0$. Then $\mathbf{x} = \mathbf{e}_l + \mathbf{v}_i$ and the received vector 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 $\underline{2^{n-k}}$ coset leaders are all the *correctable error patterns*.

We have total 2^{n-k} so this we call as correctable error patterns because whenever our error pattern is a coset leader we are not going to make a mistake in decoding and hence we call these as correctable error pattern.

(Refer Slide Time: 30:04)

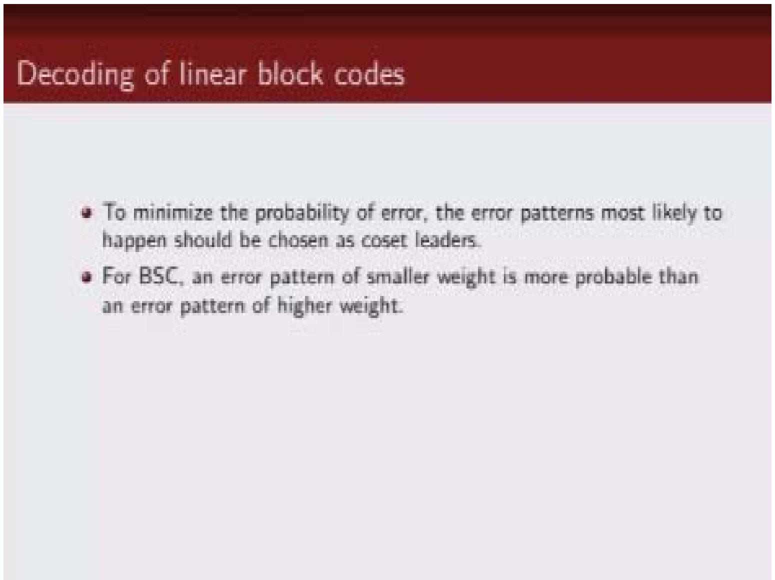


Decoding of linear block codes

- 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 how should we choose our coset leader? So 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 the error patterns which have least hamming weight so we start with first start with error pattern of hamming weight one, if we run out of them start with hamming weight two, three like that because they are more likely error pattern.

(Refer Slide Time: 30:56)

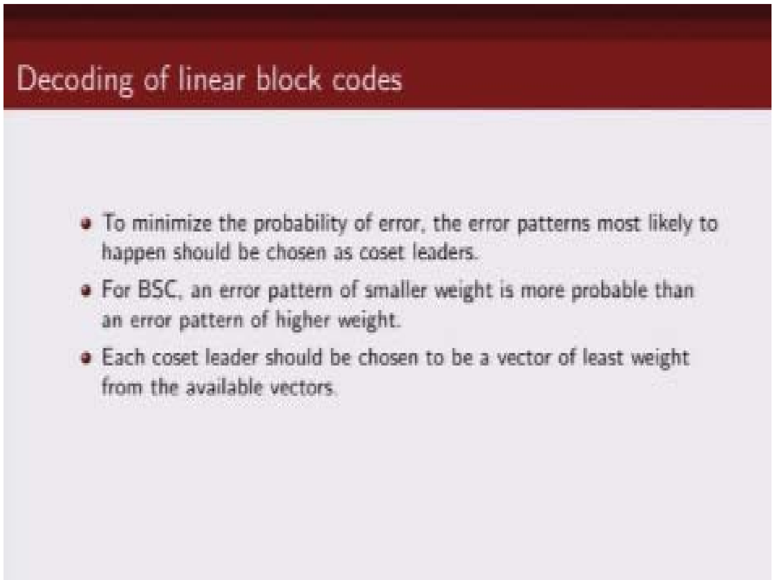


Decoding of linear block codes

- 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 a binary symmetric channel the error pattern with smaller weights are more likely than error pattern.

(Refer Slide Time: 31:05)

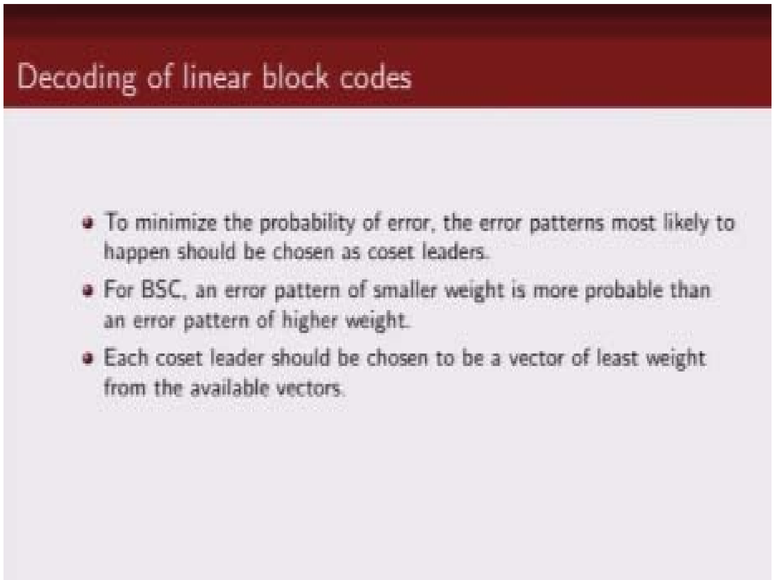


Decoding of linear block codes

- 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.

With larger hamming weight, so among a coset we should choose element which has basically the smallest error pattern as our coset leader.

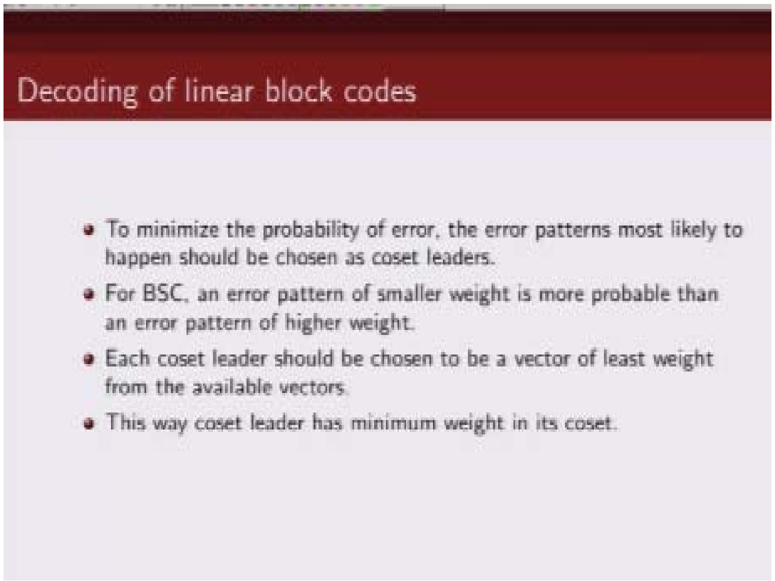
(Refer Slide Time: 31:19)



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.

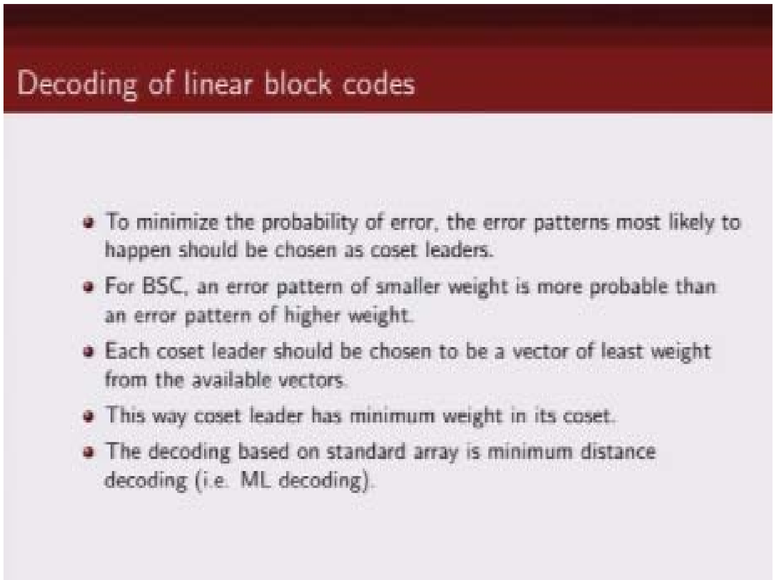
(Refer Slide Time: 31:21)



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.

(Refer Slide Time: 31:23)

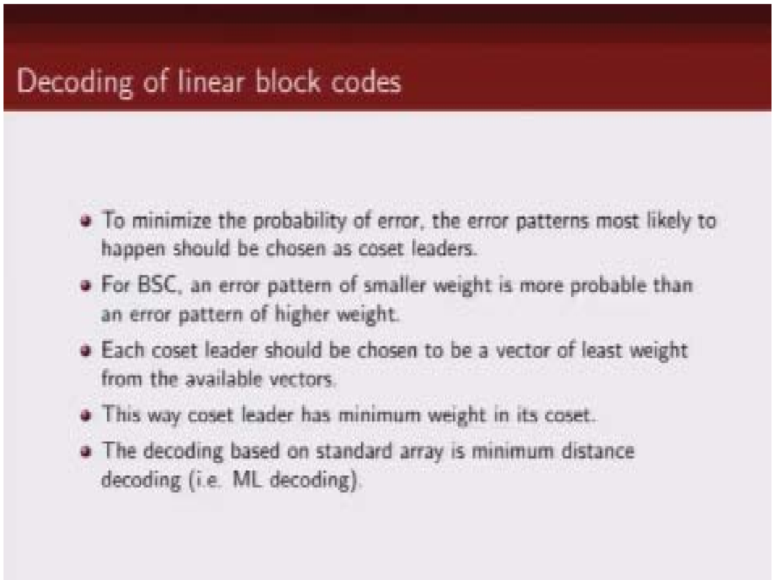


Decoding of linear block codes

- 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 basically also maximum likelihood decoding because we have shown earlier that maximum likelihood rule for binary symmetric channel it basically chooses v in such a way such that the hamming distance between r and v is minimized, so in other words we have to choose an error pattern that has the minimum hamming weight.

(Refer Slide Time: 31:50)

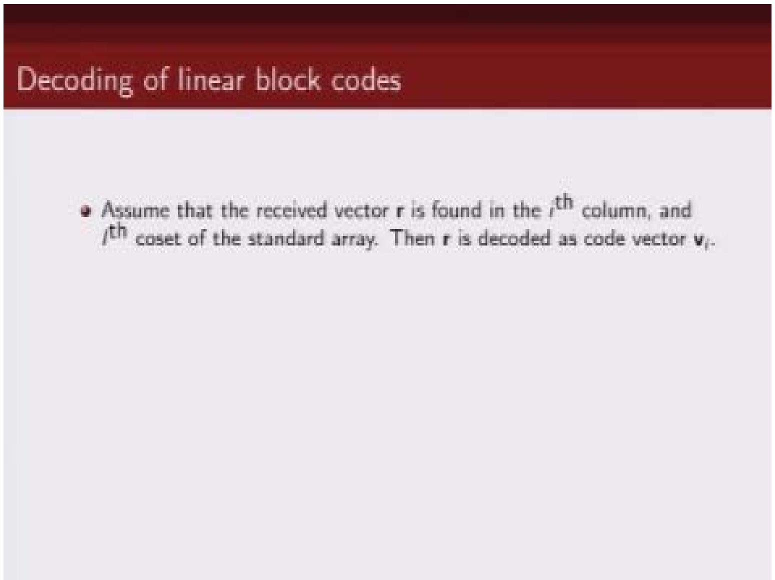


Decoding of linear block codes

- 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).

So that is also the maximum likelihood decoding.

(Refer Slide Time: 31:53)

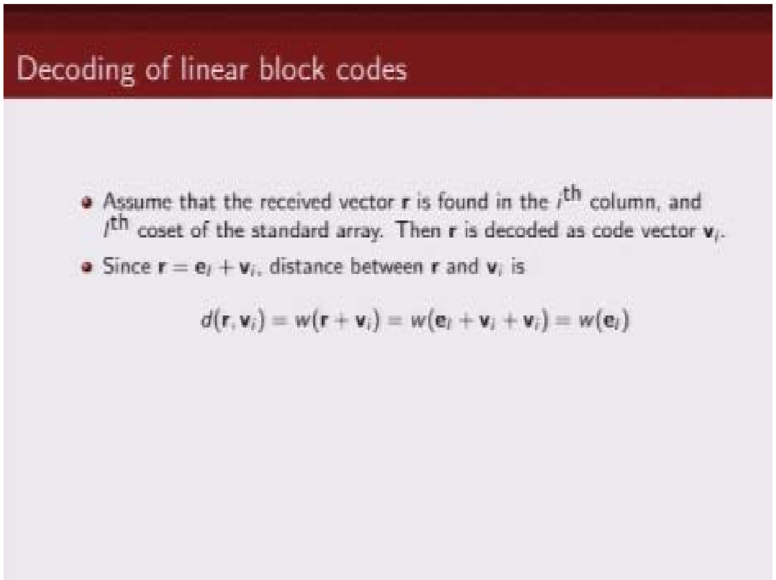


Decoding of linear block codes

- 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 $\mathbf{v}_i$.

So suppose our received vector is found in the i^{th} column and l^{th} coset of the standard array so in that case because our received vector r is in i^{th} column we are going to decode it as $v_{i.}$

(Refer Slide Time: 32:13)



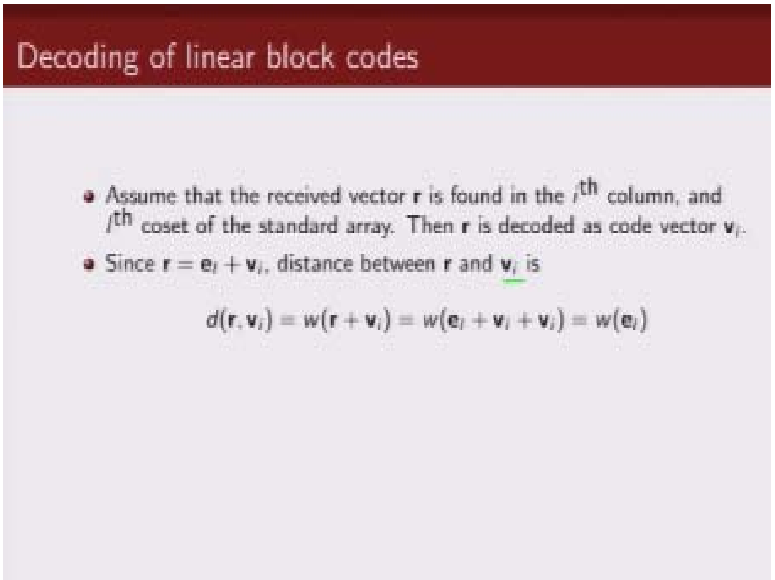
Decoding of linear block codes

- 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 i^{th} column and l^{th} coset so the error pattern is $e^{e}{}_{l}$ so our received sequence is our transmitted code word plus error pattern, now let us try to find out the hamming distance between our receive vector.

(Refer Slide Time: 32:40)



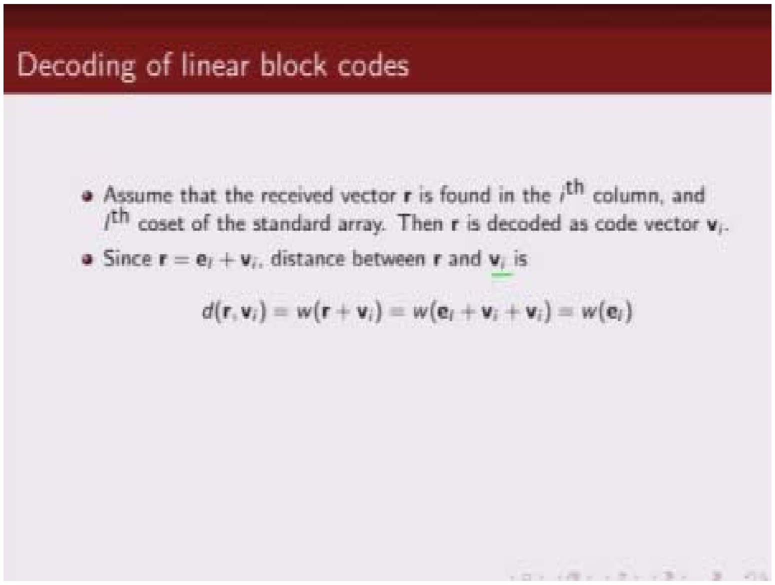
Decoding of linear block codes

- 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

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

And this code vector v_i and hamming distance between receive vector and some other code word, so when we try hamming distance between received code word r and this code vector v_i we can see hamming distance is nothing but number of locations where these two bits are differing, so if we add r and v and then count the number of ones that would give us the hamming distance between r and v.

(Refer Slide Time: 33:09)



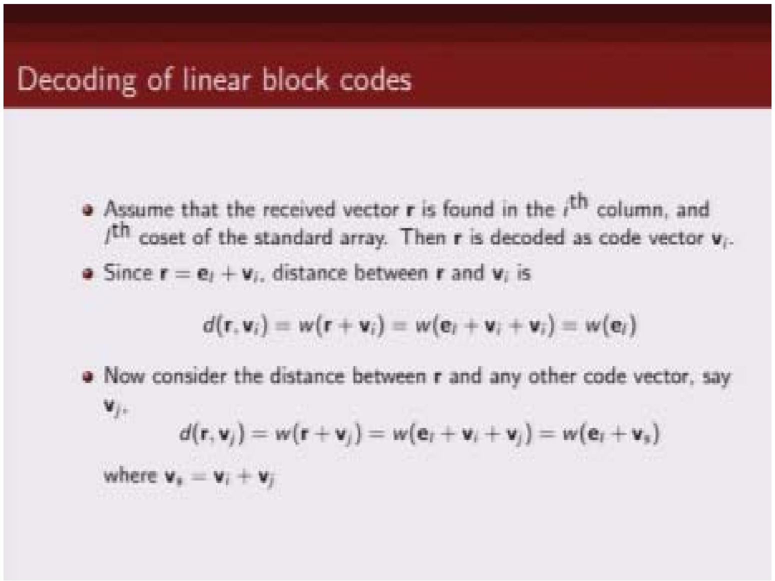
Decoding of linear block codes

- 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

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

So hamming distance in r and v is nothing but hamming weight of the vector r + v_i and what is r? It is $e_l + v_i$ and plus v_i so this was nothing but weight of error vector.

(Refer Slide Time: 33:29)



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

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

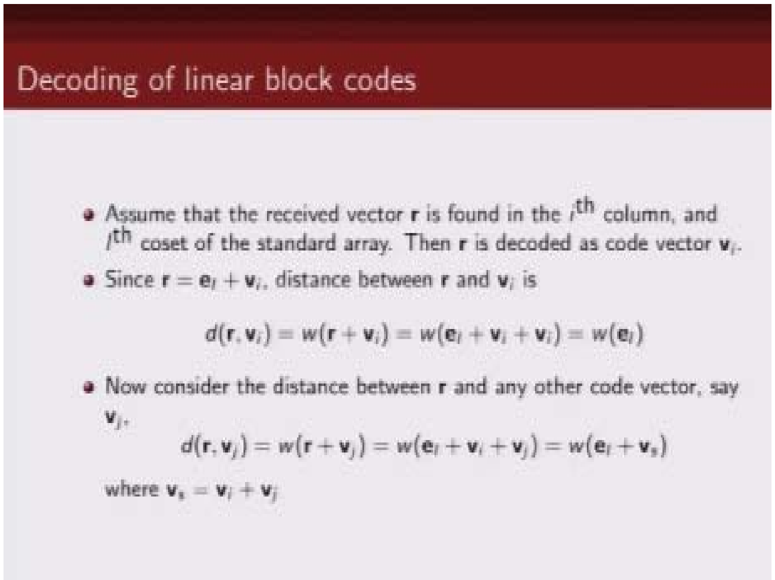
- Now consider the distance between $\mathbf{r}$ and any other code vector, say $\mathbf{v}_j$.

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

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

Next, consider now the hamming distance between r and any other code word let us call it $v_{j,}$ so we are finding hamming distance between r and any other code word v_j so that would be given by weight of this vector r + $v_{j,}$ so r is nothing but $e_l + v_i + v_j$ now $v_i + v_j$ sum of two code words is another code word so this would be let us call that code word v_s .

(Refer Slide Time: 34:06)



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

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

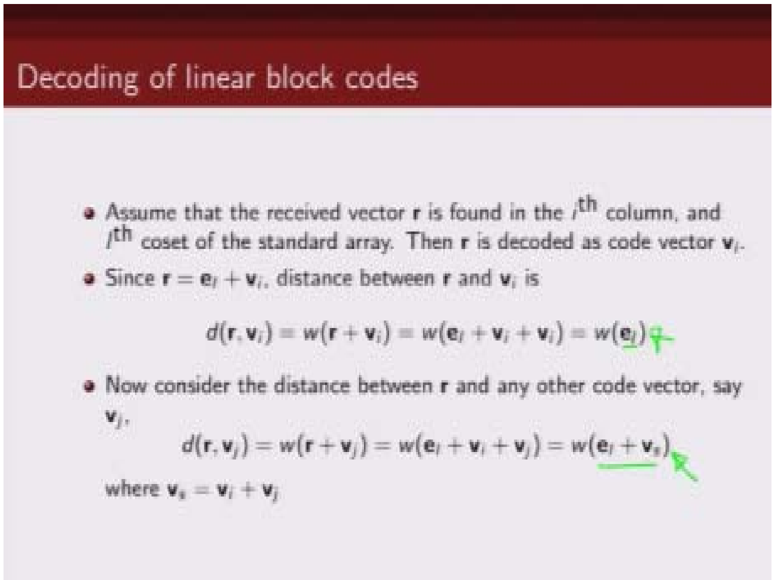
- Now consider the distance between $\mathbf{r}$ and any other code vector, say $\mathbf{v}_j$,

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

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

So this will be weight of e_l + v_s so what have we done so far, we found out that the hamming distance between the receive code word and this code vector v_i is given by this.

(Refer Slide Time: 34:22)



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

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

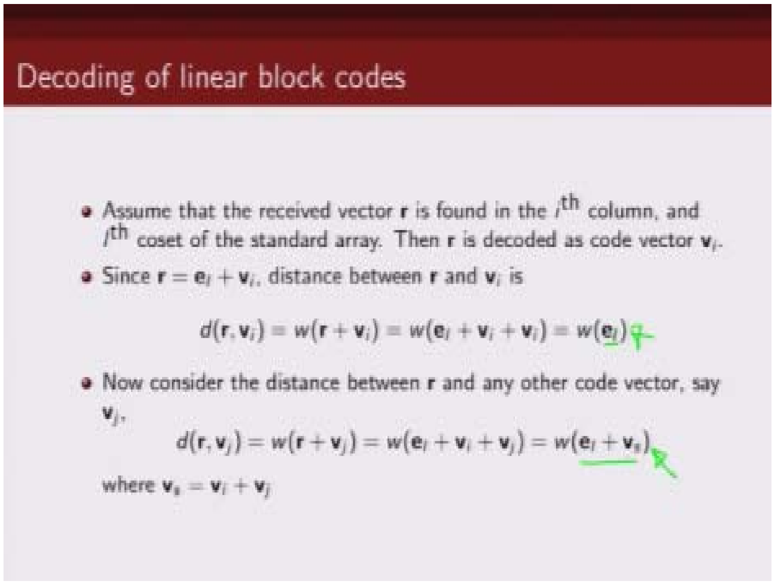
- Now consider the distance between $\mathbf{r}$ and any other code vector, say $\mathbf{v}_j$,

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

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

And hamming distance between receive code word in any other code word which is not v_i is basically given by this, now el and $e_l + v_s$ are going to be the elements in the same coset, right? And if we choose e_l to be our coset leader which has minimum number of ones

(Refer Slide Time: 34:52)



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

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

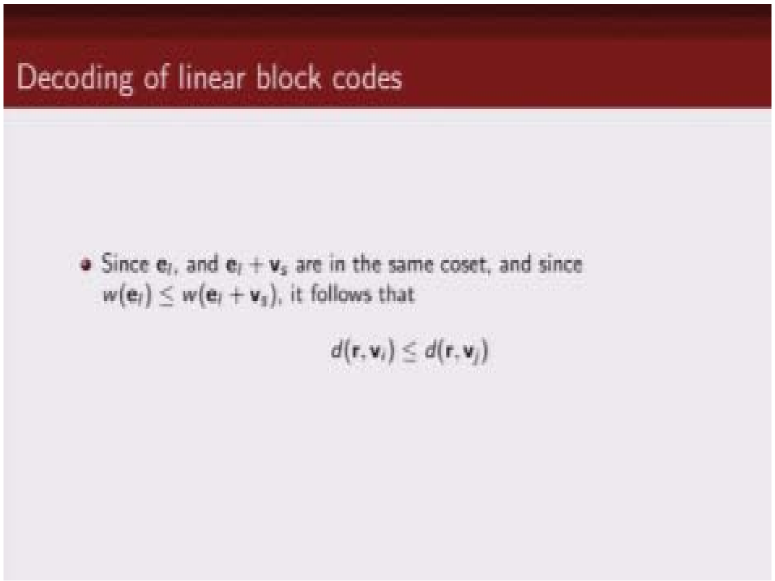
- Now consider the distance between **r** and any other code vector, say $\mathbf{v}_j$,

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

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

Then this would be less than this so we, our minimum distance decoding will decide in favor of v_i and not any other code word v_j.

(Refer Slide Time: 35:05)



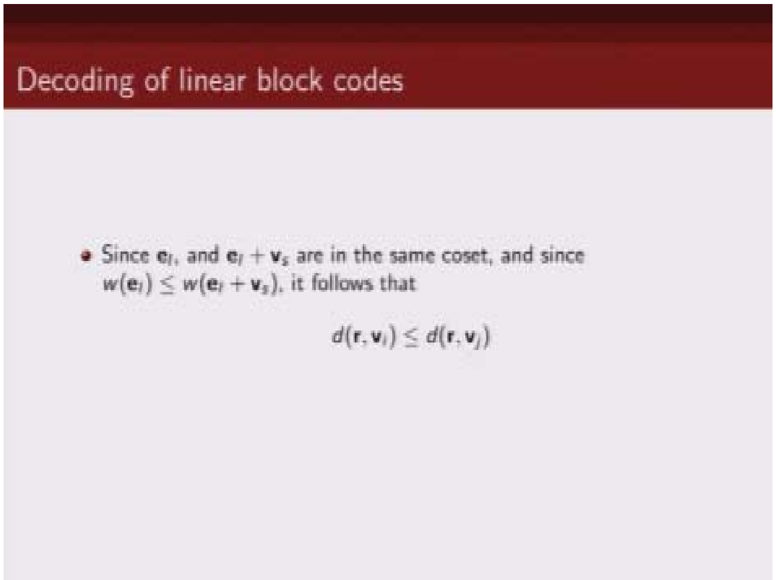
Decoding of linear block codes

- 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 $e_l + v_s$ are in the same coset and our coset leader has minimum hamming weight, weight of this is less than equal to weight of this.

(Refer Slide Time: 35:21)



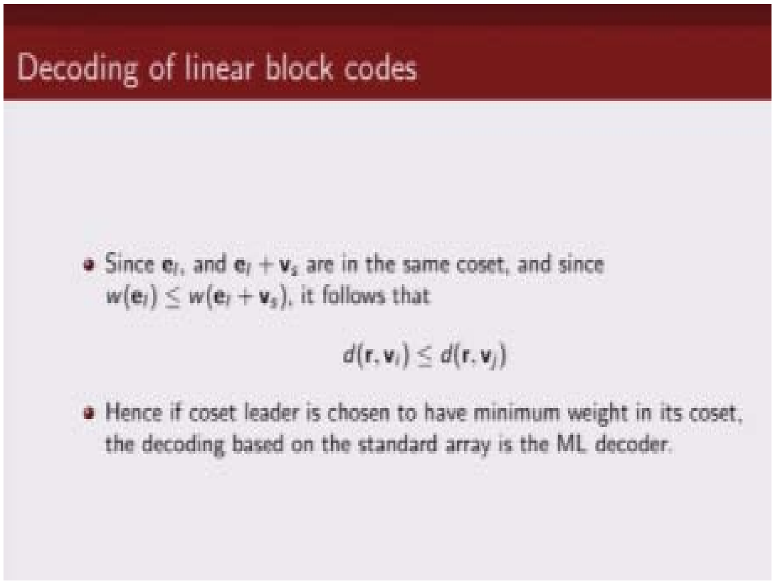
Decoding of linear block codes

- 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

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

Then this will always happen, in other words v_i will be closer to r then any other code word v_j to r so this will be our correct decoding.

(Refer Slide Time: 35:35)



Decoding of linear block codes

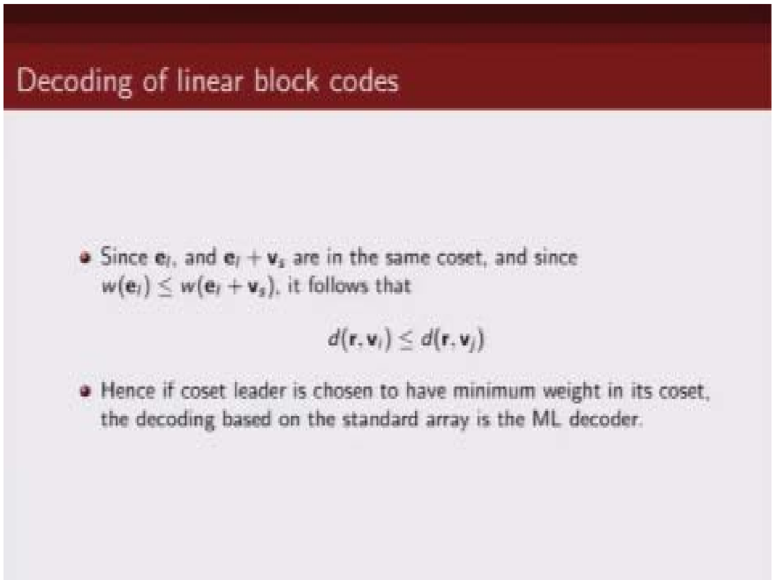
- Since $\mathbf{e}_l$, and $\mathbf{e}_l + \mathbf{v}_s$ are in the same coset, and since $w(\mathbf{e}_l) \leq w(\mathbf{e}_l + \mathbf{v}_s)$, it follows that

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

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

So if we choose 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 a maximum likelihood decoder, because maximum likelihood decoder for binary symmetric channel we have said is the one which minimizes hamming distance between receive code word and the selected code vector v, okay.

(Refer Slide Time: 36:02)



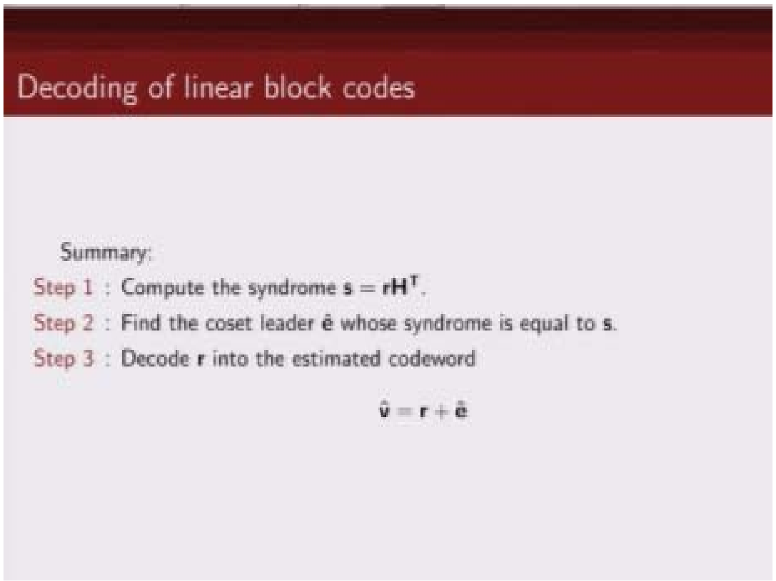
Decoding of linear block codes

- 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.

(Refer Slide Time: 36:07)



Decoding of linear block codes

Summary:

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

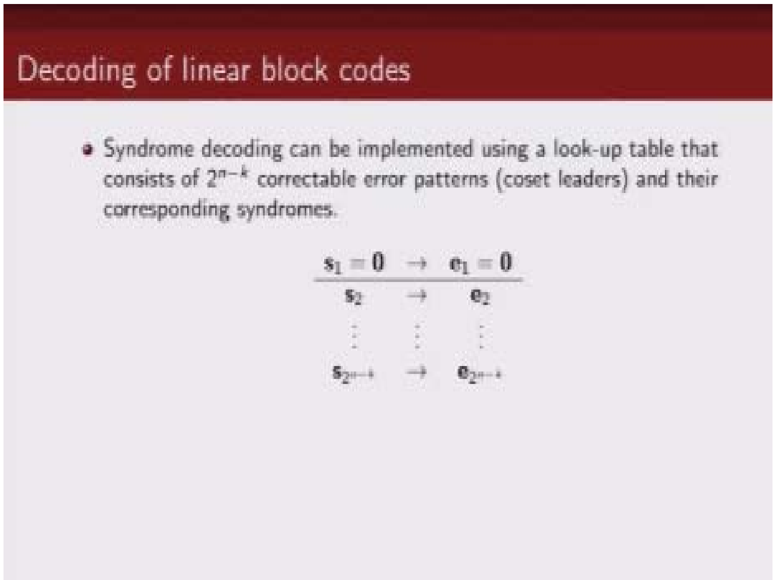
Step 2 : Find the coset leader $\hat{\mathbf{e}}$ whose syndrome is equal to $\mathbf{s}$.

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

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

So now 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 receive sequence and that would be our estimated code word.

(Refer Slide Time: 36:39)



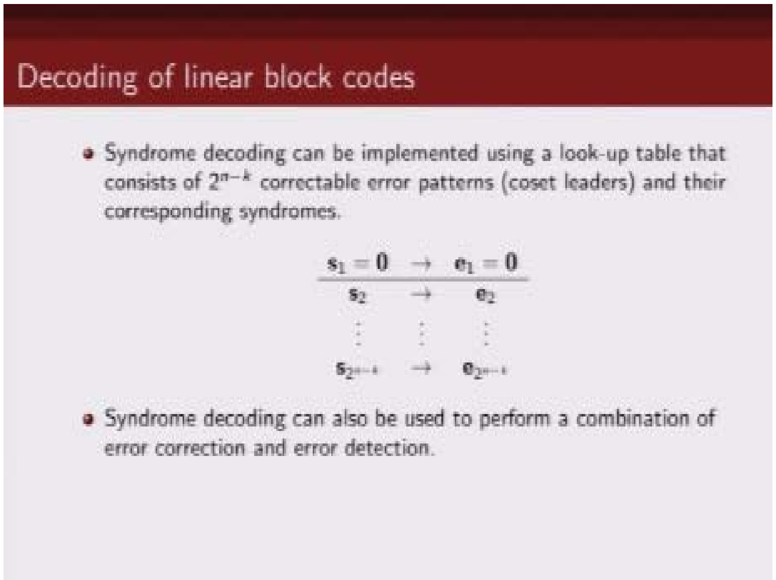
Decoding of linear block codes

- 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.

$\mathbf{s}_1 = \mathbf{0}$	$\rightarrow$	$\mathbf{e}_1 = \mathbf{0}$
$\mathbf{s}_2$	$\rightarrow$	$\mathbf{e}_2$
$\vdots$	$\vdots$	$\vdots$
$\mathbf{s}_{2^{n-k}}$	$\rightarrow$	$\mathbf{e}_{2^{n-k}}$

And this mapping from syndrome to error pattern basically can be implemented as a table look-up, so which syndrome corresponds to which coset leader this can be implemented as a table look-up.

(Refer Slide Time: 36:57)



Decoding of linear block codes

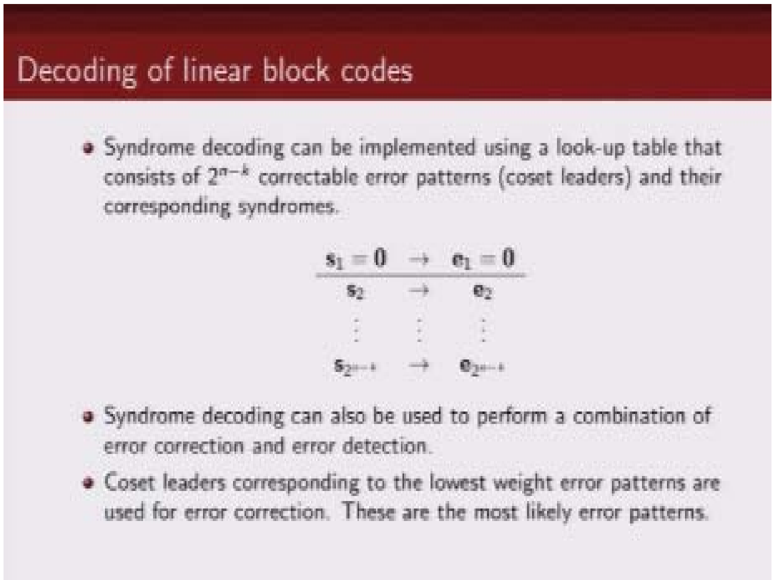
- Syndrome decoding can be implemented using a look-up table that consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes.

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

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

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

(Refer Slide Time: 37:05)



Decoding of linear block codes

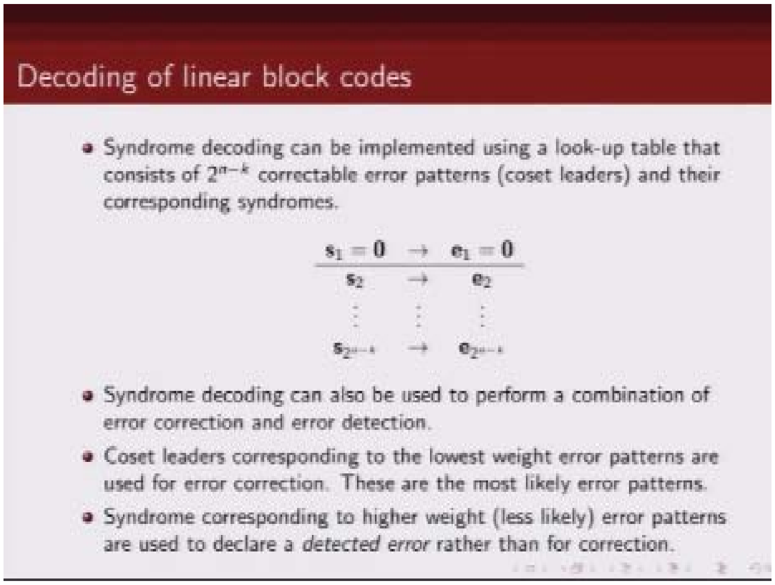
- Syndrome decoding can be implemented using a look-up table that consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes.

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

- 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 correction and these are the most likely error patterns according to the maximum likelihood rule.

(Refer Slide Time: 37:21)



Decoding of linear block codes

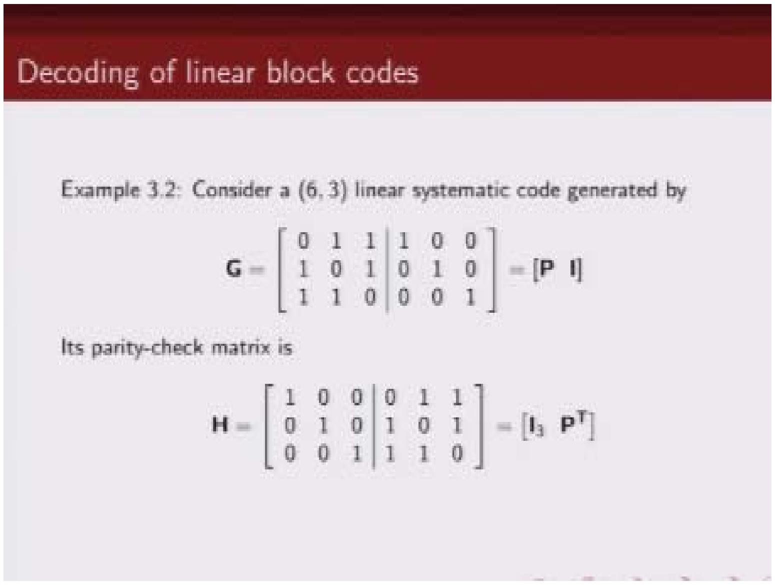
- Syndrome decoding can be implemented using a look-up table that consists of 2^{n-k} correctable error patterns (coset leaders) and their corresponding syndromes.

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

- 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 error correction and error detection and we could, we are going to illustrate this point that we could use a syndrome corresponding to higher weight error pattern for error detection rather than correction.

(Refer Slide Time: 37:41)



Decoding of linear block codes

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

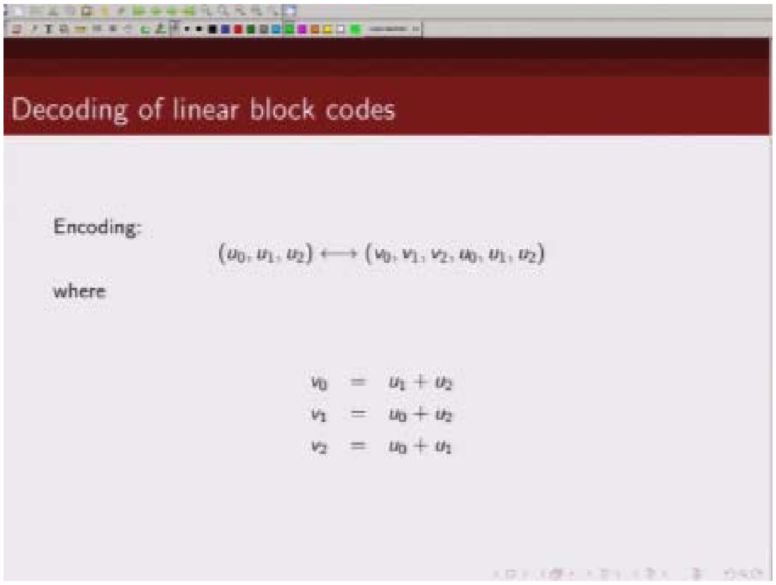
$$\mathbf{G} = \left[\begin{array}{ccc|ccc} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \end{array}\right] = [\mathbf{P} \;\; \mathbf{I}]$$

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 an 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 this.

(Refer Slide Time: 38:06)



Decoding of linear block codes

Encoding:

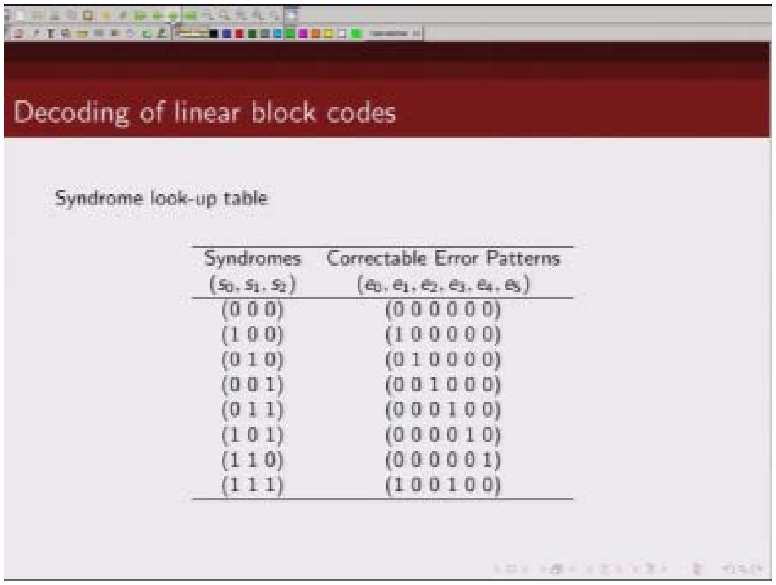
$$(u_0, u_1, u_2) \longleftrightarrow (v_0, v_1, v_2, u_0, u_1, u_2)$$

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: 38:10)



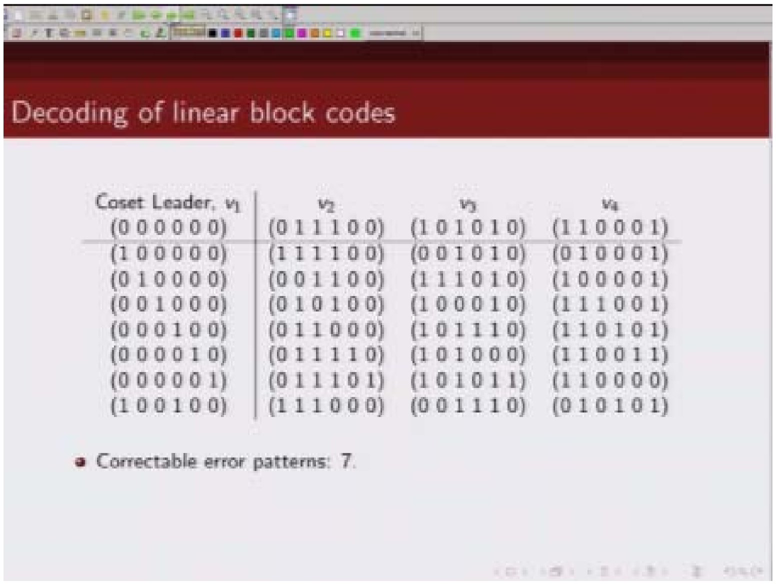
Decoding of linear block codes

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)

This is a mapping of the syndrome to the coset leader.

(Refer Slide Time: 38:17)



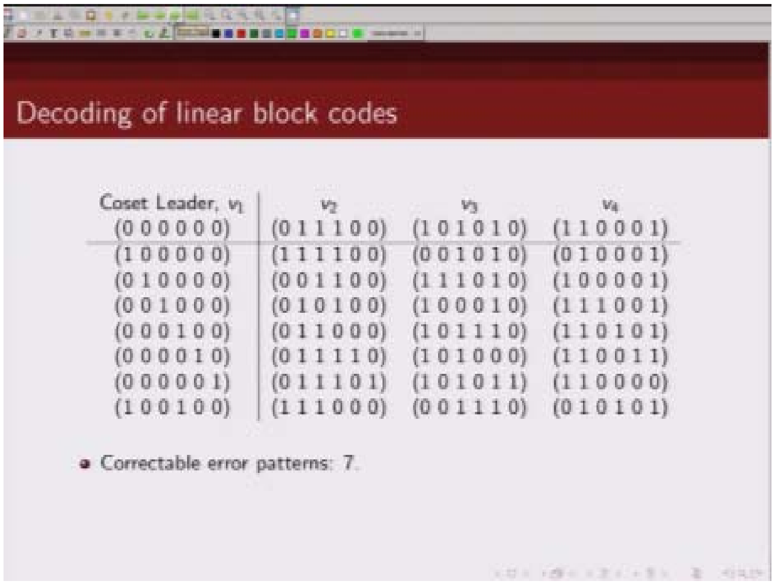
Decoding of linear block codes

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.

So the first step involved in this is creation of standard array. And I believe now you know how to create a standard array. The first step is.

(Refer Slide Time: 38:31)



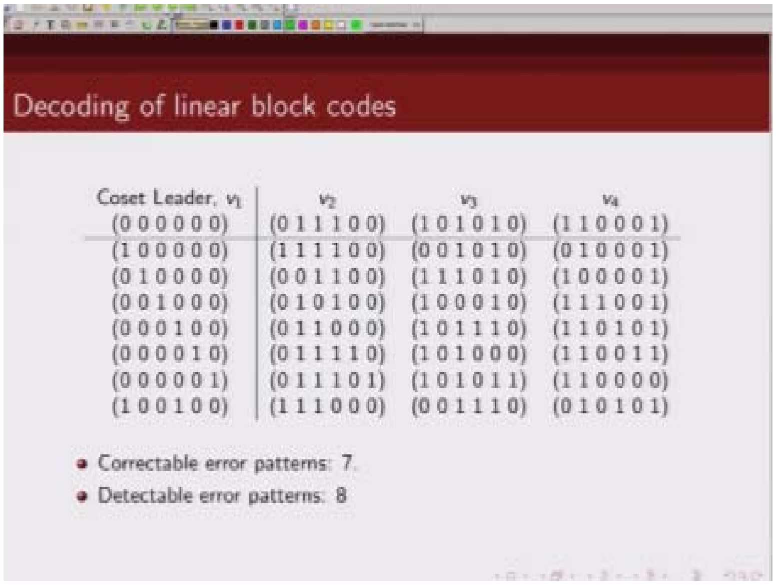
Decoding of linear block codes

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 first row of the standard array will be a set of code words. You are already given the generator matrix so you can generate what are the set of code words. The left most entry in the first row which is the row of code words should be all zero code words and then you can place.

(Refer Slide Time: 38:49)



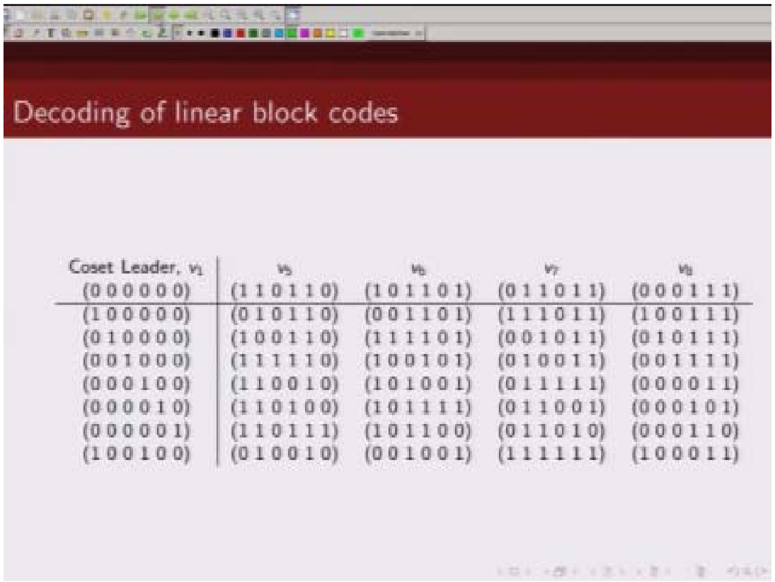
Decoding of linear block codes

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

The other code words in any order. Now I could not 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: 39:01)

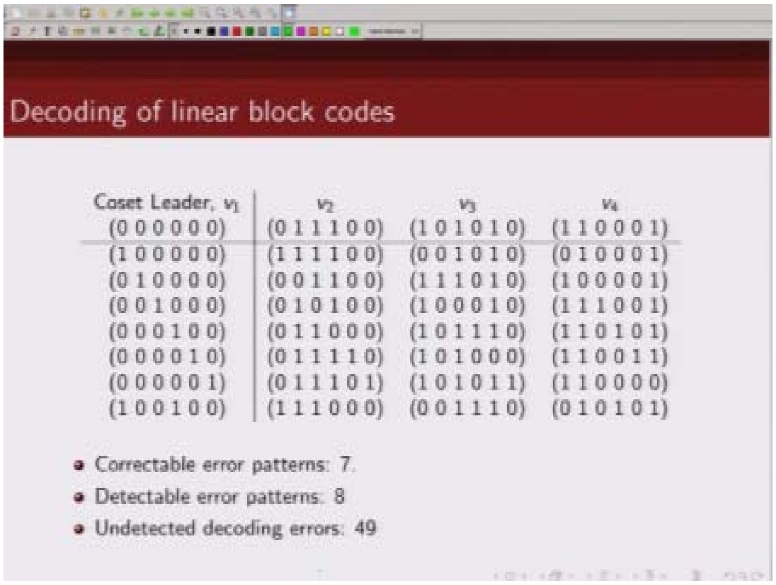


Decoding of linear block codes

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_5 to v_8 in the next slide okay.

(Refer Slide Time: 39:08)



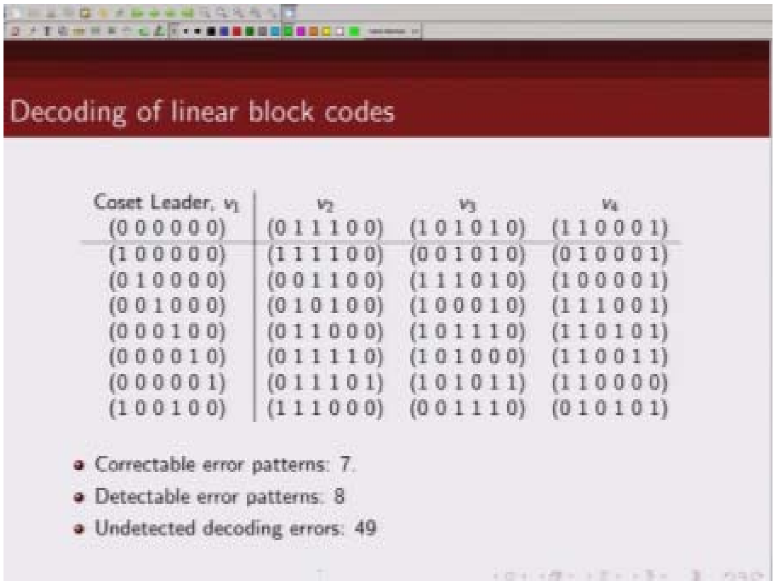
Decoding of linear block codes

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

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

(Refer Slide Time: 39:45)



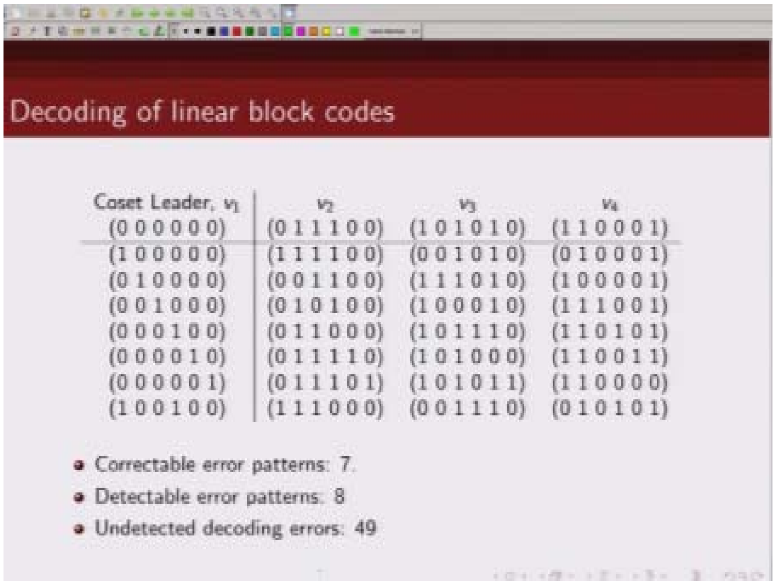
Decoding of linear block codes

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)
(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

Which has the least number of ones, and if you make that error pattern as your coset leader then you can correctly decode those error patterns. So let us look at each of these rows of course.

(Refer Slide Time: 40:01)



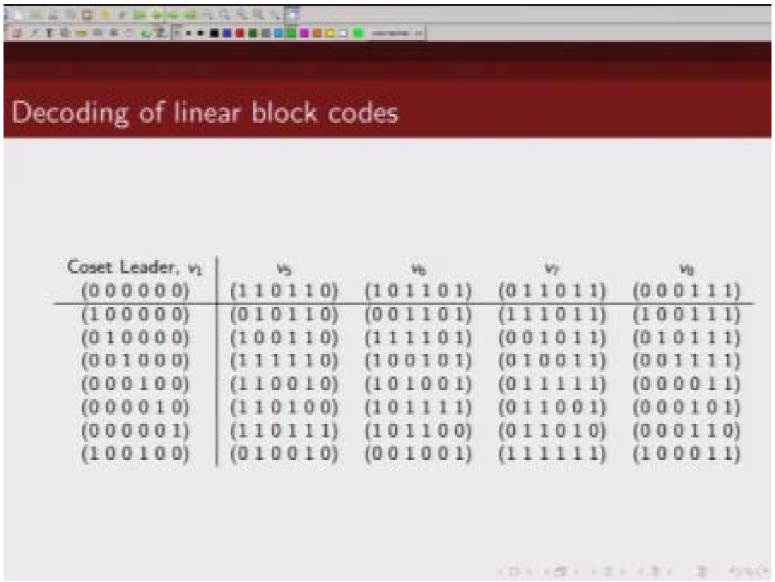
Decoding of linear block codes

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

This corresponds to all correct code words. If the error pattern is this, these are the set of other elements of the coset.

(Refer Slide Time: 40:12)

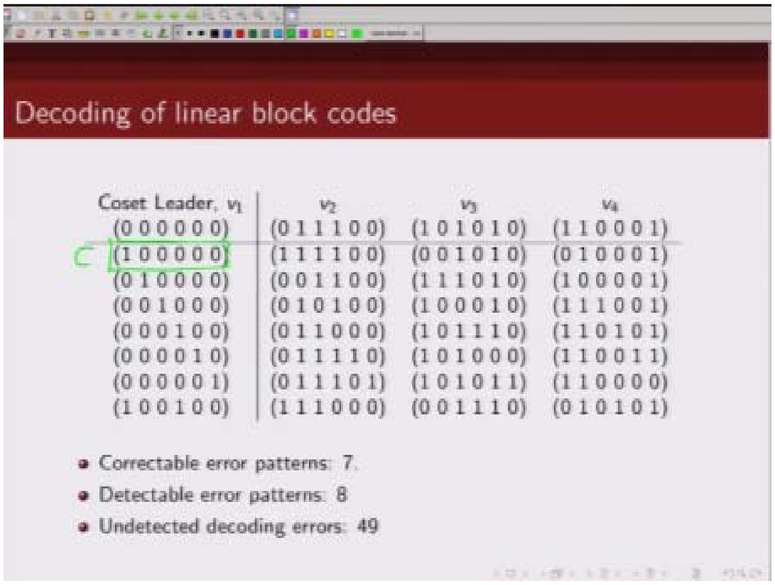


Decoding of linear block codes

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 you can see here, none of these elements have weight less than two, they are all like 3, 3, 5, 4.

(Refer Slide Time: 40:23)



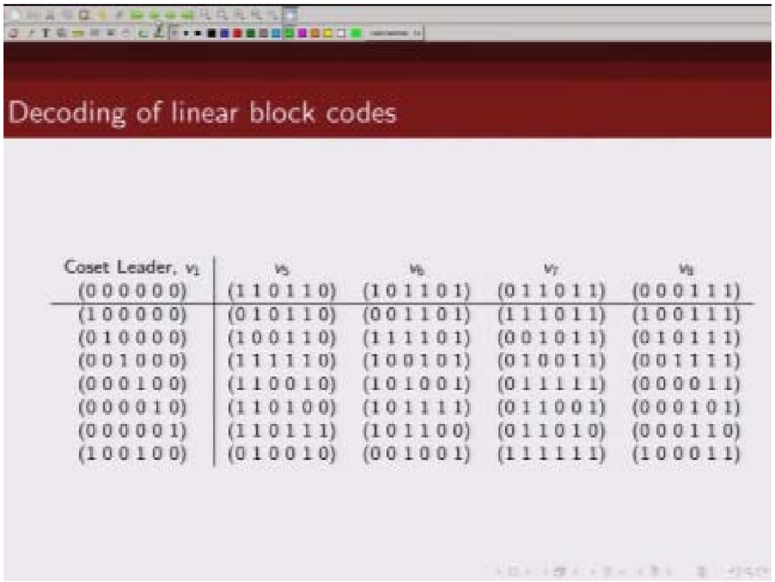
Decoding of linear block codes

	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

And this has weight 4, 2, 2, so this is the minimum weight error pattern and this is hamming weight one. So this error pattern is correctable, if this error happens this can be corrected. What about this? Just look at other elements this has 2 ones, 4 ones, 2 ones.

(Refer Slide Time: 40:54)

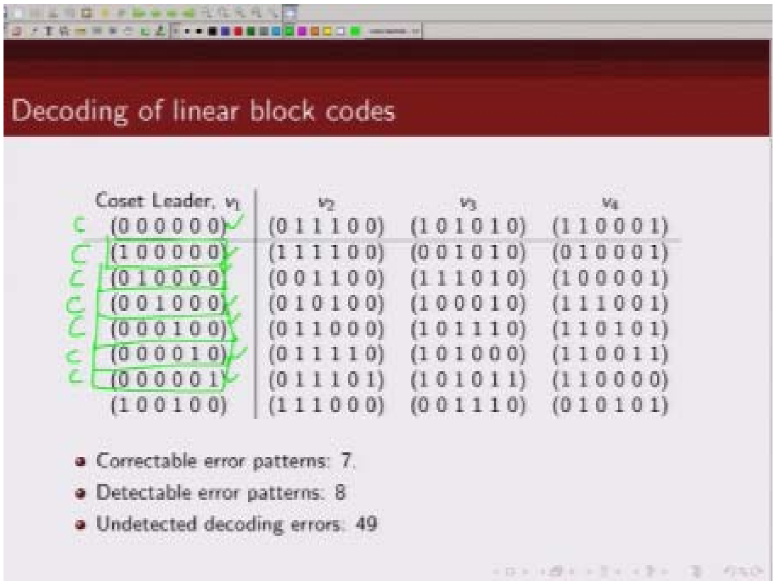


Decoding of linear block codes

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)

3 ones, 5 ones, 3 ones, 4 ones. So clearly this has.

(Refer Slide Time: 41:04)



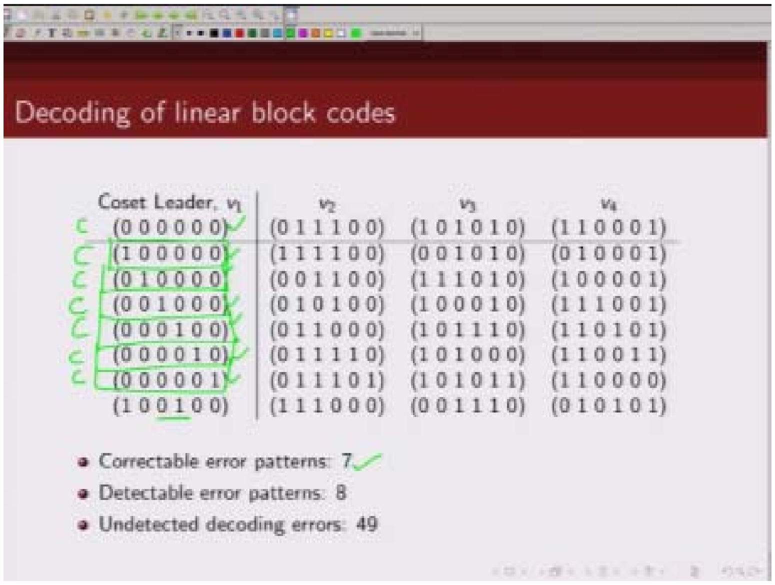
Decoding of linear block codes

	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

Only one, one rest others all have weight two or more. So if this is the error pattern and we make this as coset leader this is also correctable. 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 I talk about correctable patterns essentially I mean if you get this, this, this, this, this, this, of course there is no error case which also I am counting in correctable pattern. So these seven patterns if these are the error patterns then it is correctable. So these are the seven correctable patterns.

(Refer Slide Time: 41:51)



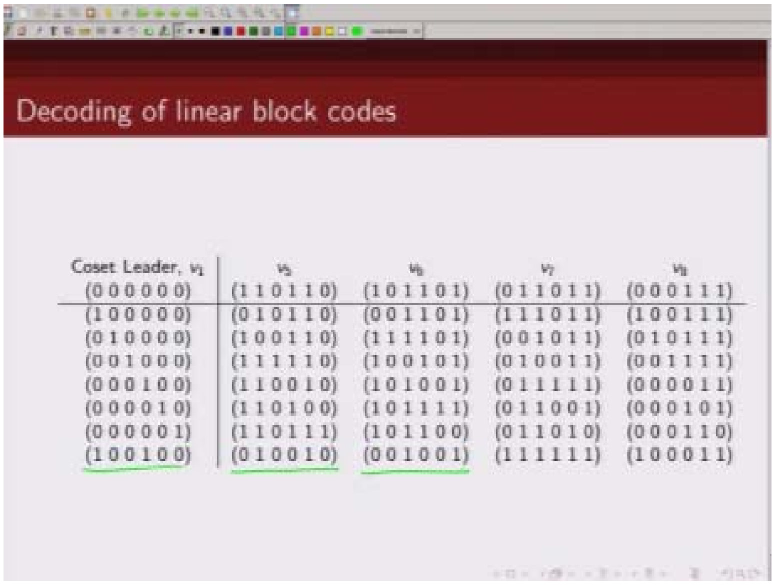
Decoding of linear block codes

	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

Now what happens here, this error pattern has weight two, this has three, this has three, this has three.

(Refer Slide Time: 42:02)

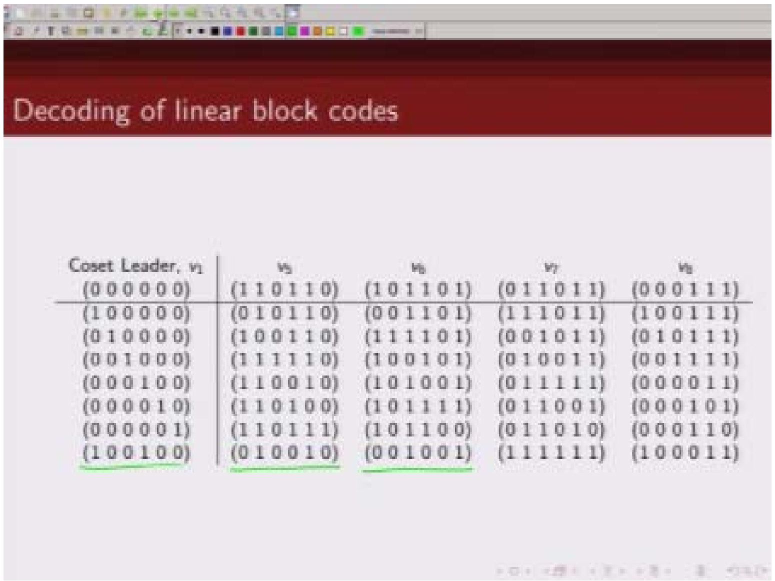


Decoding of linear block codes

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)

Oh this has two. So this has two, this also has two, this also has two, so in this last row my coset leader is no longer the error pattern with lowest hamming weight.

(Refer Slide Time: 42:21)

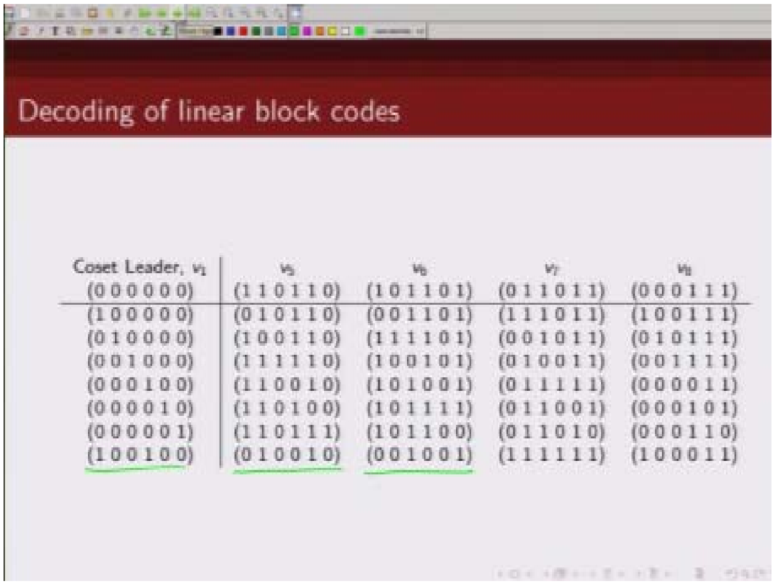


Decoding of linear block codes

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)

There are two other error patterns which also have hamming weight two okay. So in this situation

(Refer Slide Time: 42:30)

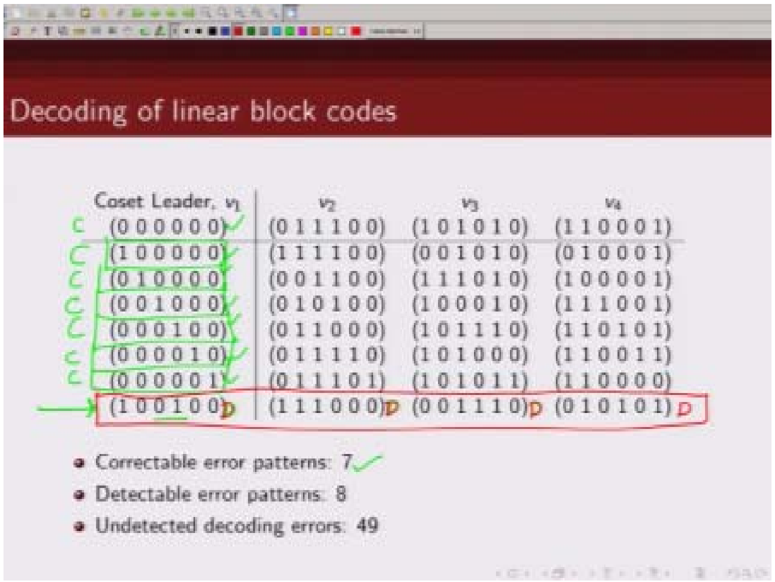


Decoding of linear block codes

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)

When my syndrome is pointing to this coset I would not be able to do error correction, why, because I know this could be a likely error pattern, or this could be a likely error pattern, or this could be a likely error pattern, and they are all equally likely. Because here two bits got flipped, here two bits got flipped, here two bits got flipped. So any of these three patterns are

(Refer Slide Time: 42:57)



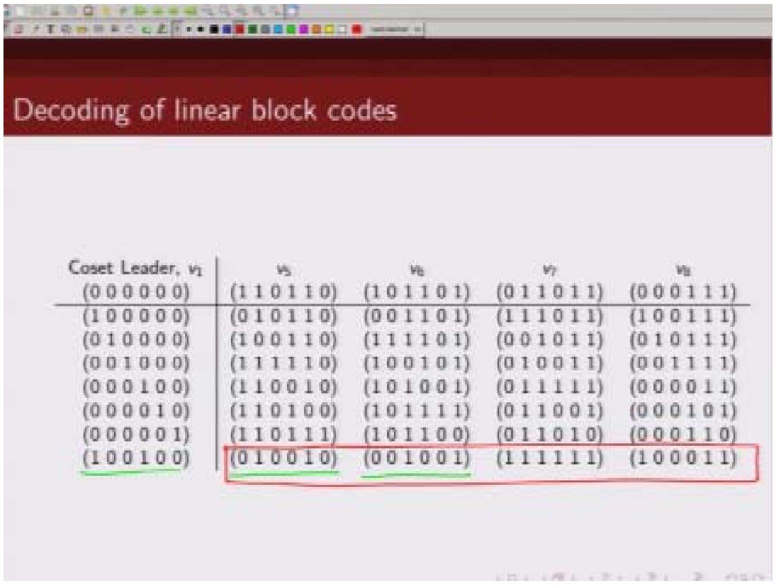
Decoding of linear block codes

	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

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, whenever it points to this row I know there is an error. Because syndrome is non zero, but I do not know what is my error. So that is why I call it as detectable error pattern. So what are those detectable error pattern, this is our detectable error pattern. This is my detectable error pattern, this is my detectable error pattern. Maybe I can use a different this in color so I use red, these are my detectable error pattern. These are my detectable error patterns. These are four of them and then

(Refer Slide Time: 43:50)

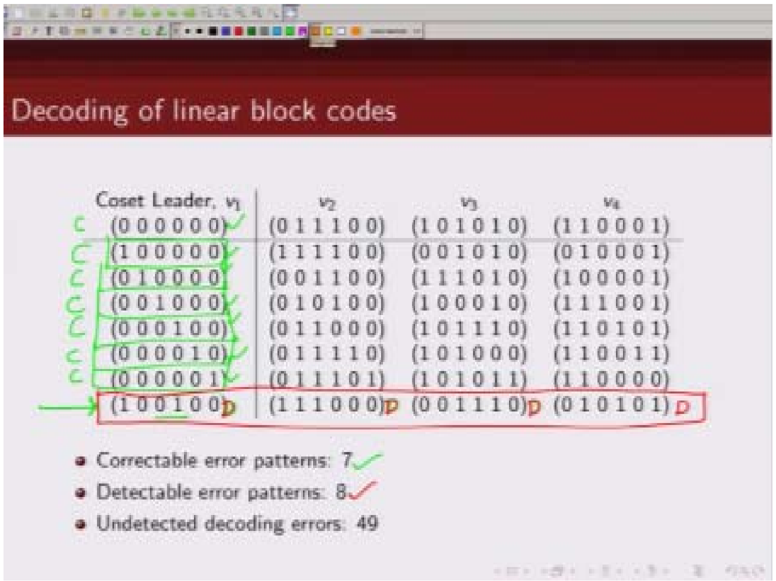


Decoding of linear block codes

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)

Remaining four are these.

(Refer Slide Time: 43:58)



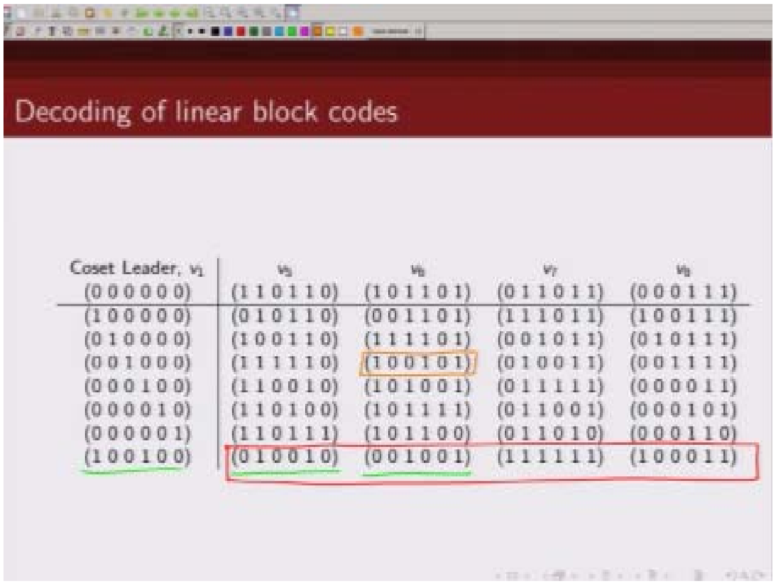
Decoding of linear block codes

	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

So these are my eight detectable patterns, error patterns. Now let us 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?

(Refer Slide Time: 44:18)

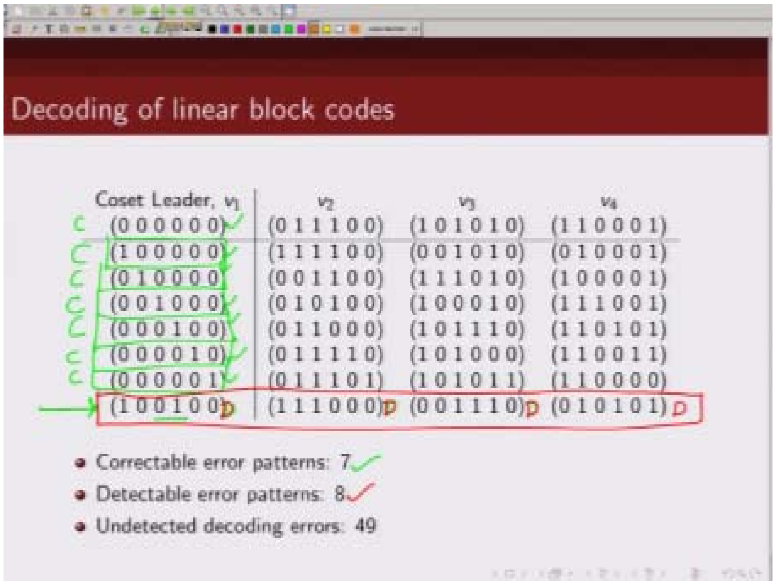


Decoding of linear block codes

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 are red pen let us say if this error pattern happens what is going to happen. If this happens I am not able to.

(Refer Slide Time: 44:29)



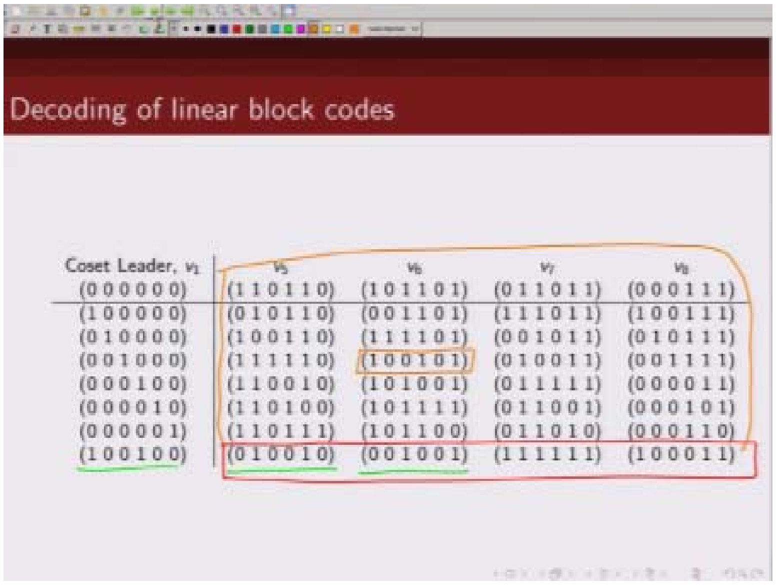
Decoding of linear block codes

	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

Detect these error pattern, why?

(Refer Slide Time: 44:33)

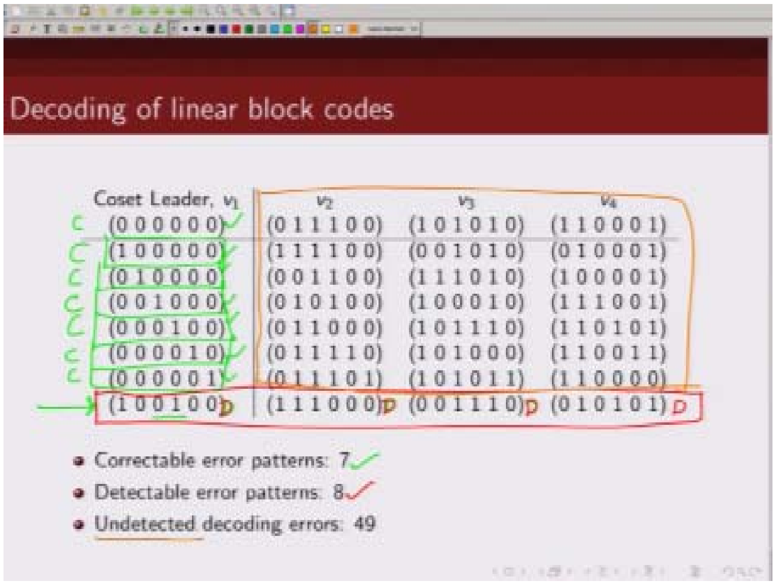


Decoding of linear block codes

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)

Whenever this error pattern happens this is hamming weight three, its coset leader was already hamming weight one. So whenever these error patterns happen, whenever these error patterns happen I am not able to detect.

(Refer Slide Time: 44:50)



Decoding of linear block codes

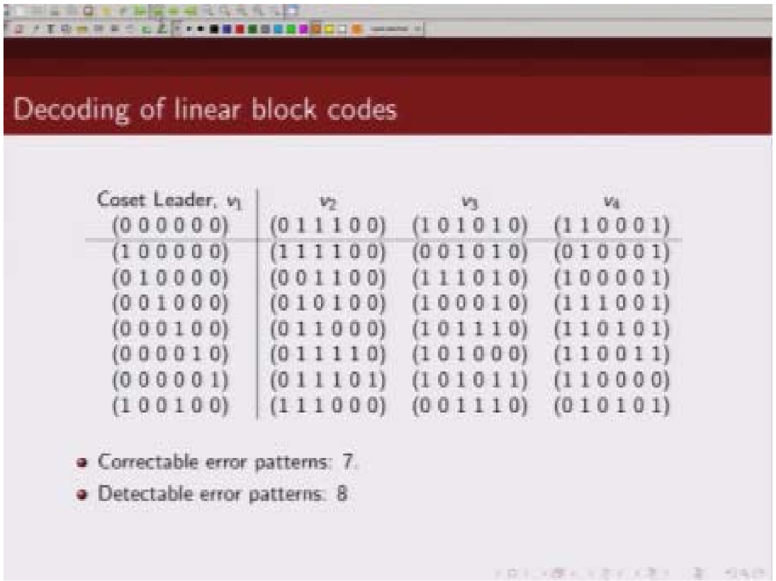
	Coset Leader, v_1	v_2	v_3	v_4
C	(0 0 0 0 0 0) ✓	(0 1 1 1 0 0)	(1 0 1 0 1 0)	(1 1 0 0 0 1)
C	(1 0 0 0 0 0) ✓	(1 1 1 1 0 0)	(0 0 1 0 1 0)	(0 1 0 0 0 1)
C	(0 1 0 0 0 0)	(0 0 1 1 0 0)	(1 1 1 0 1 0)	(1 0 0 0 0 1)
C	(0 0 1 0 0 0) ✓	(0 1 0 1 0 0)	(1 0 0 0 1 0)	(1 1 1 0 0 1)
C	(0 0 0 1 0 0) ✓	(0 1 1 0 0 0)	(1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

Any of these error patterns why? Because whenever any of these error patterns happen any of these error patterns happens I have already taken decision in favor of these coset leaders. Because it is less likely to get this error pattern than this. So these set of 49 error patterns, if any of these error patterns happen then I am going to make a mistake. This will result in undetected error, because whenever any of these error patterns happen I would have assumed 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 this 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 ones. And whenever we get an error our syndrome will be non zero, the syndrome will point out to a particular coset or a row of this standard array, and we will if the coset leader has the minimum number of one’s then we will pick that coset leader as our likely error pattern. And we are going to.

(Refer Slide Time: 46:23)

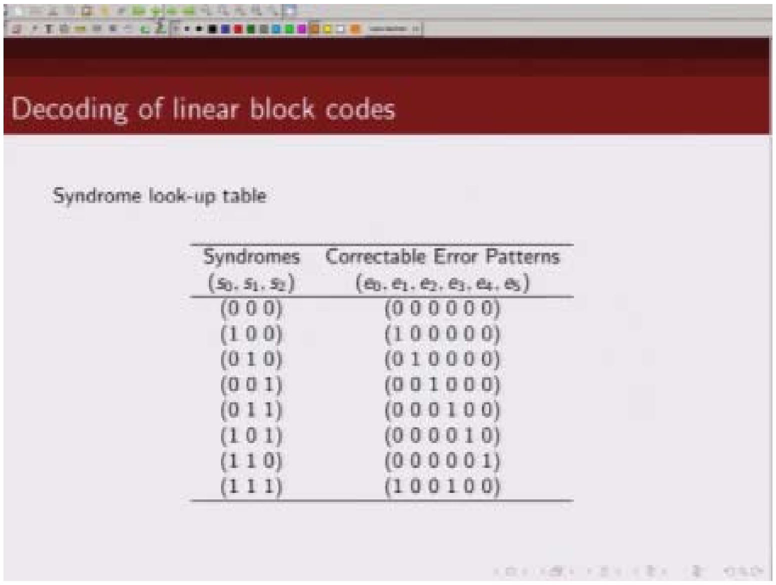


Decoding of linear block codes

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

(Refer Slide Time: 46:24)

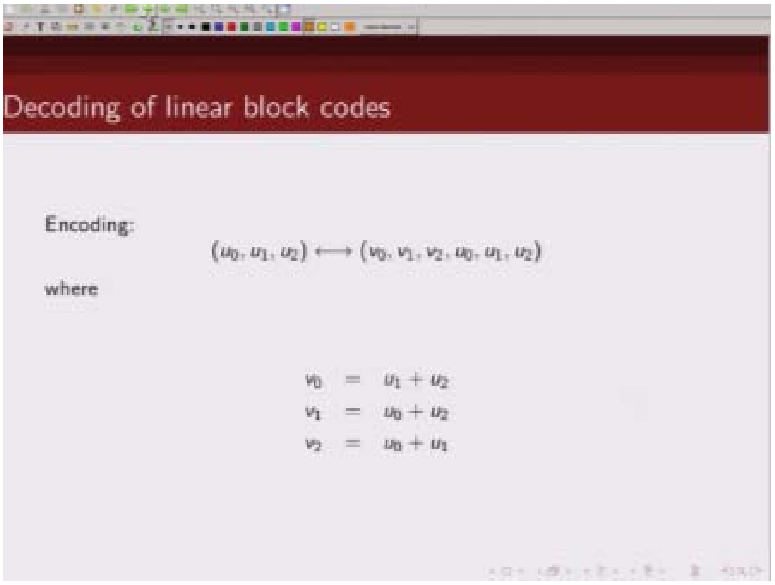


Decoding of linear block codes

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)

(Refer Slide Time: 46:24)



Decoding of linear block codes

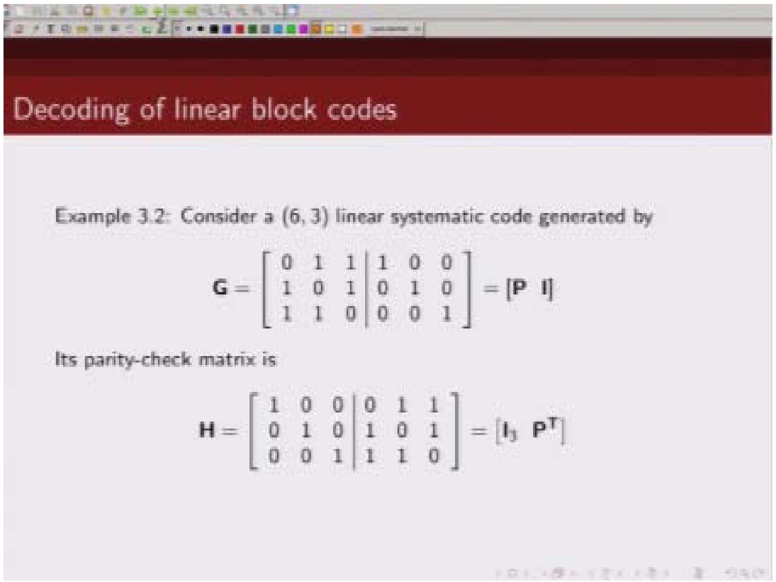
Encoding:

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

where

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

(Refer Slide Time: 46:24)



Decoding of linear block codes

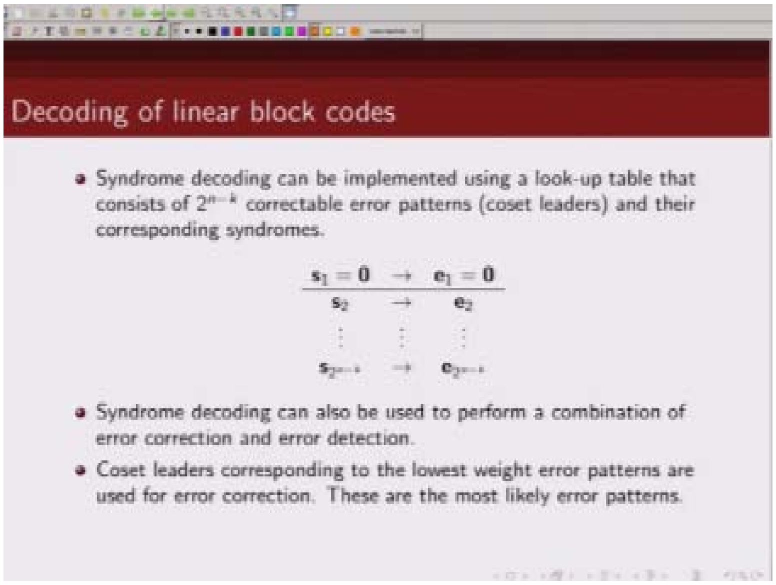
Example 3.2: Consider a (6, 3) linear systematic code generated by

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

Its parity-check matrix is

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

(Refer Slide Time: 46:24)



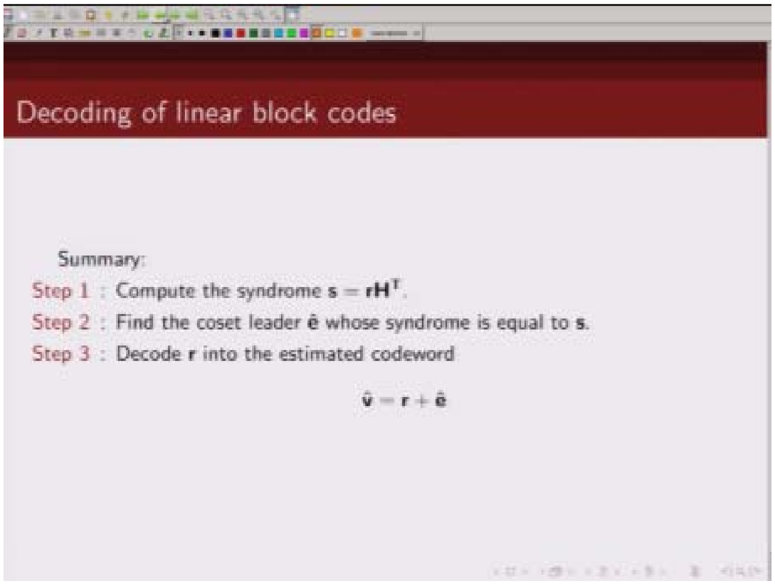
Decoding of linear block codes

- 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.

(Refer Slide Time: 46:25)



Decoding of linear block codes

Summary:

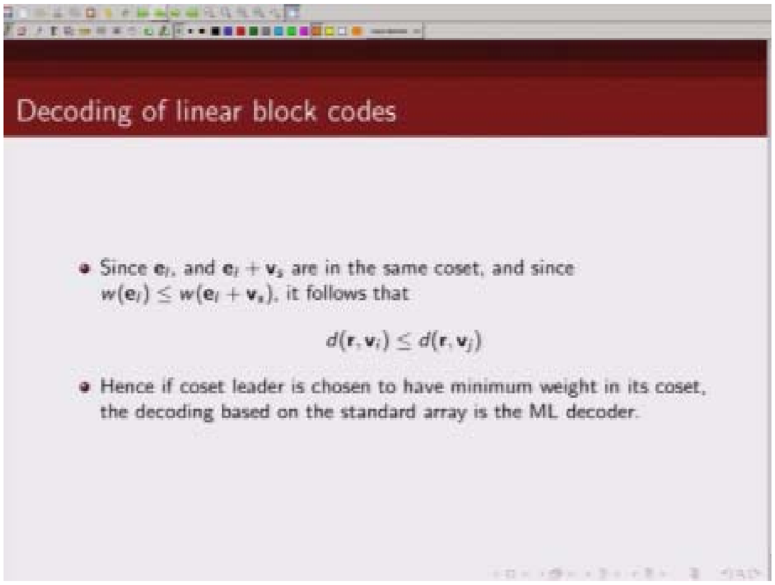
Step 1 : Compute the syndrome $\mathbf{s} = \mathbf{r}\mathbf{H}^T$.

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

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

(Refer Slide Time: 46:25)



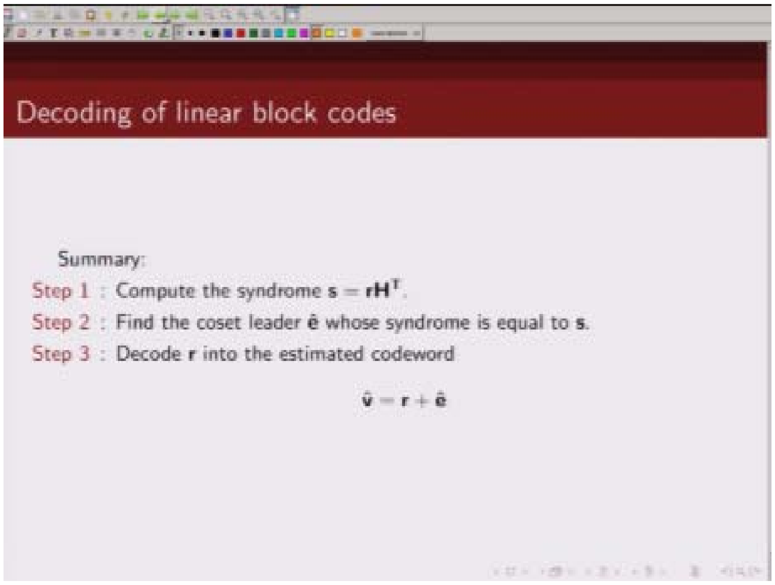
Decoding of linear block codes

- 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.

(Refer Slide Time: 46:26)



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

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

Decode it by picking that likely error pattern, adding it to our receive code word and that would be our estimated code word. So with this I will conclude this decoding of linear block codes. Thank you.

Acknowledgement
Ministry of Human Resource & Development

Prof. Satyaki Roy
Co-ordinator, NPTEL IIT Kanpur

NPTEL Team
Sanjay Pal
Ashish Singh
Badal Pradhan
Tapobrata Das
Ram Chandra
Dilip Tripathi
Manoj Shrivastava
Padam Shukla
Sanjay Mishra
Shubham Rawat
Shikha Gupta
K. K. Mishra
Aradhana Singh

Sweta
Ashutosh Gairola
Dilip Katiyar
Sharwan
Hari Ram
Bhadra Rao
Puneet Kumar Bajpai
Lalty Dutta
Ajay Kanaujia
Shivendra Kumar Tiwari

an IIT Kanpur Production