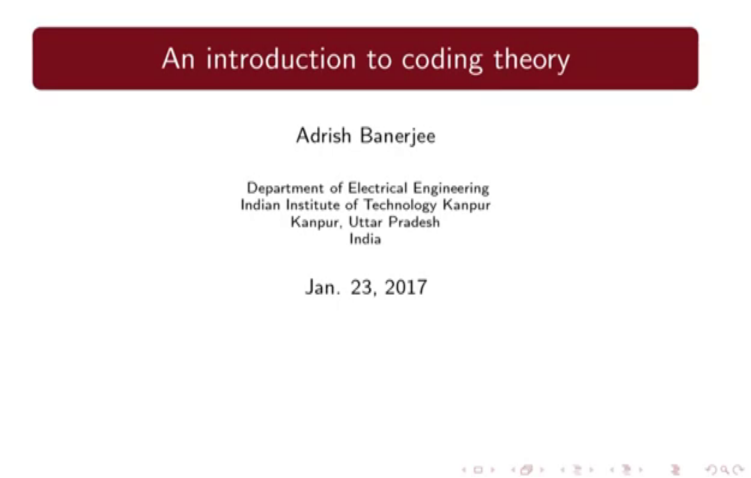
An Introduction to Coding Theory
Professor Adrish Banerji
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Module 01
Lecture Number 04
Generator Matrix and Parity Check Matrix

(Refer Slide Time 00:14)

(Refer Slide Time 00:16)

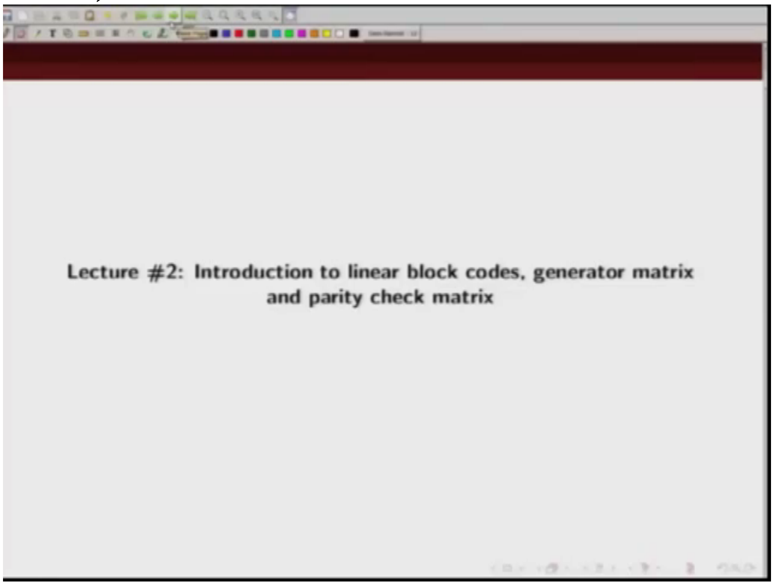


Welcome to the course on Coding Theory. Today in this lecture we are going to describe what we mean by generator matrix

(Refer Slide Time 00:23)

and parity check matrix So we will continue our discussion with introduction to linear block codes.

(Refer Slide Time 00:35)

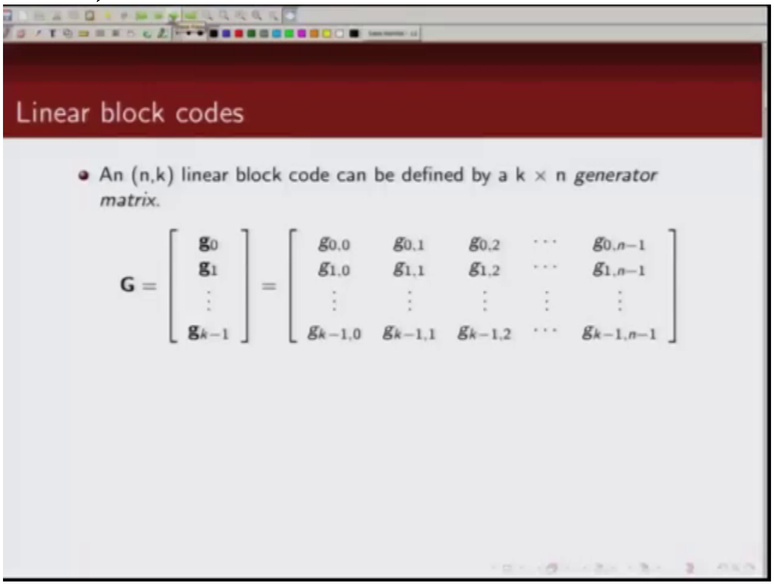


We will first describe what is a generator matrix and what is a parity check matrix and how are they related. So as we described in the last class,

(Refer Slide Time 00:44)

an encoder for a linear block codes, what it does it takes a block of k-bits and maps it to the, to n-bit. Now the matrix,

(Refer Slide Time 00:57)

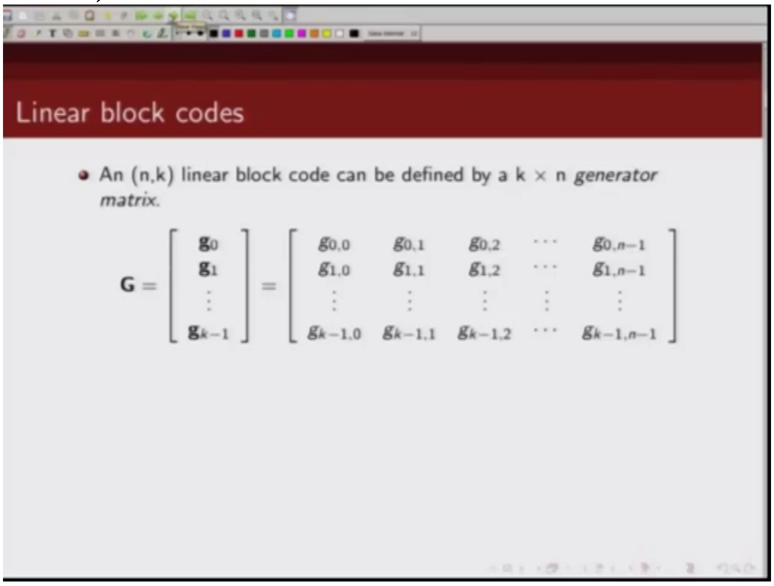


we can use a k cross n matrix to define this mapping from k information bits to n coded bits

(Refer Slide Time 01:09)

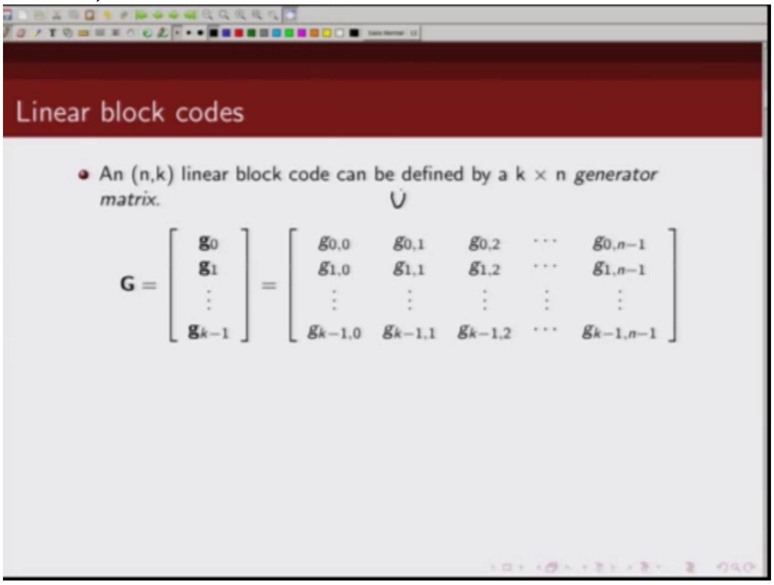
and this matrix is basically our

(Refer Slide Time 01:12)



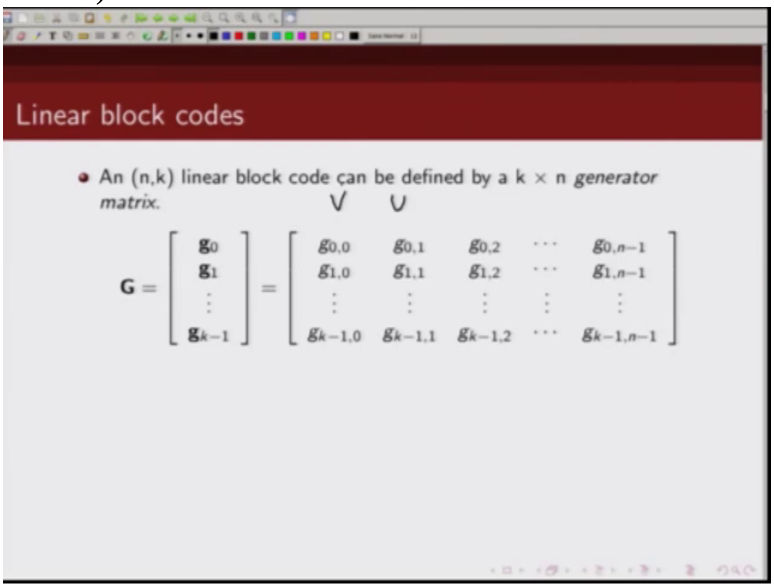
generator matrix for a, so for a n-k linear block code the mapping of k information bits to n-coded bits is defined by this generator matrix G which is of rank k so if we denote information bits by u

(Refer Slide Time 01:33)



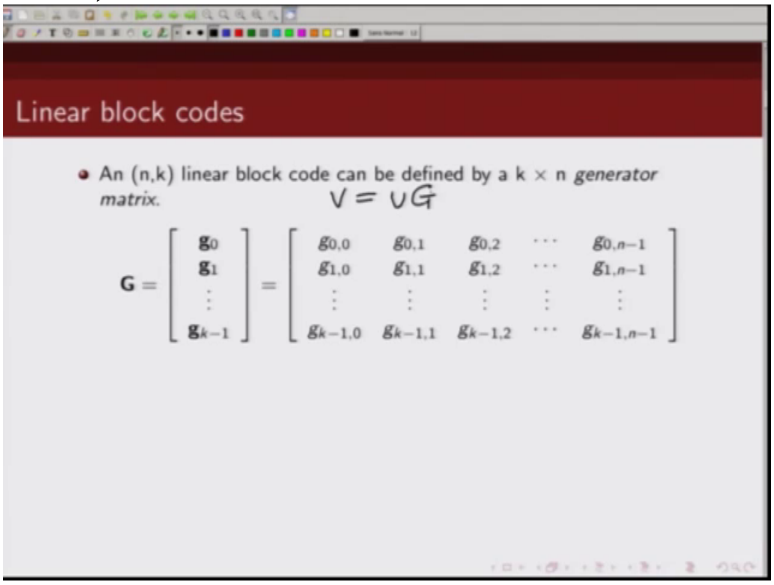
and we denote our coded bits by v,

(Refer Slide Time 01:37)



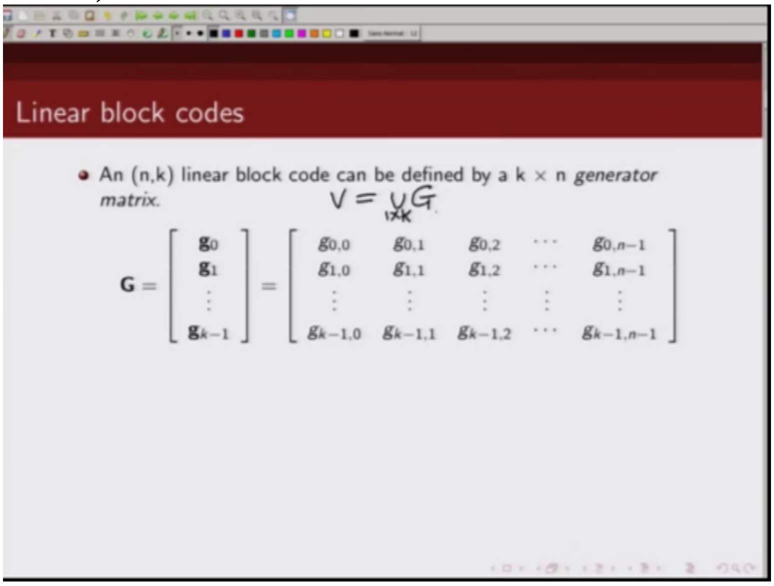
then we can write v as u times g

(Refer Slide Time 01:44)



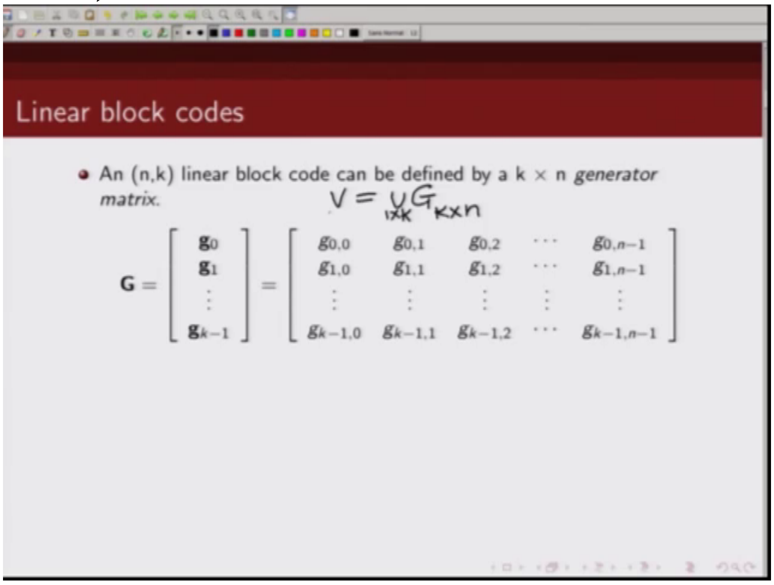
where our u is 1 cross k vector and this is,

(Refer Slide Time 01:50)



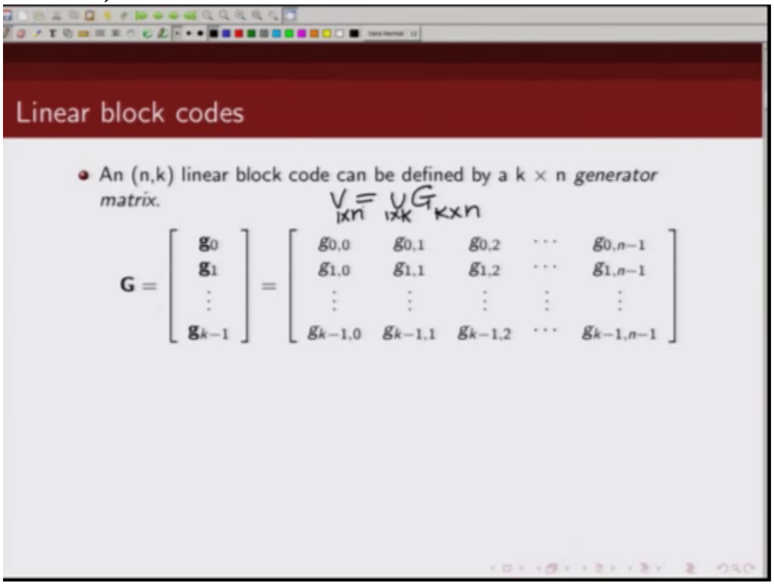
generator matrix is k cross n matrix and our output

(Refer Slide Time 01:55)



coded bit is 1 cross n vector.

(Refer Slide Time 01:58)

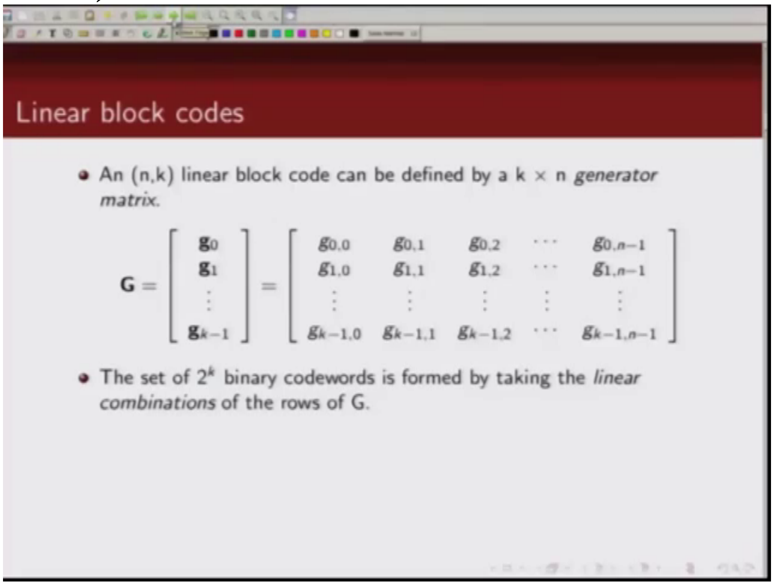


So as a name suggests, basically generator matrix is used to generate our codeword. So we generate our codewords using this generator matrix and this generator matrix gives the mapping between the information bits u to coded bits v. So how do we find codewords? We find codewords by taking linear combinations of rows of this generator matrix. In case of binary codes so then these entries in the generator matrix are either 0 or 1 depending upon which bits are used to generate a particular coded sequence. So we form

(Refer Slide Time 02:50)

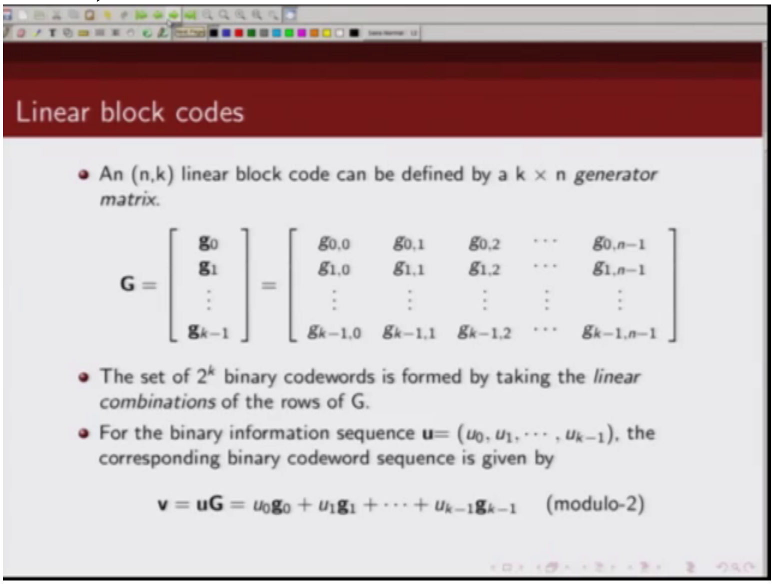
a set of 2 k codewords by taking linear combinations

(Refer Slide Time 02:55)



of rows of these generator matrix

(Refer Slide Time 03:01)

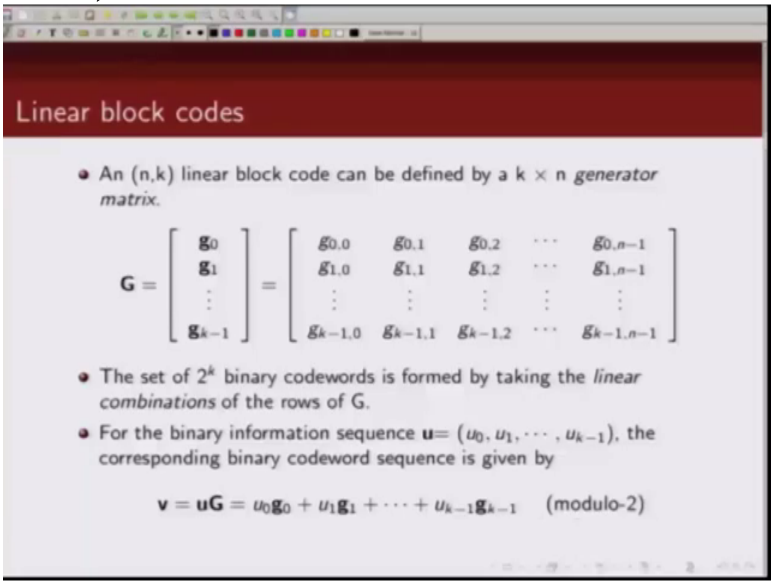


So we can, as I said, we can write our coded sequence as q times p which is basically linear combination of rows of the generator matrix. So these are basically linearly independent k rows and the rank of this generator matrix is k. Since we are without loss of generality, since

(Refer Slide Time 03:28)

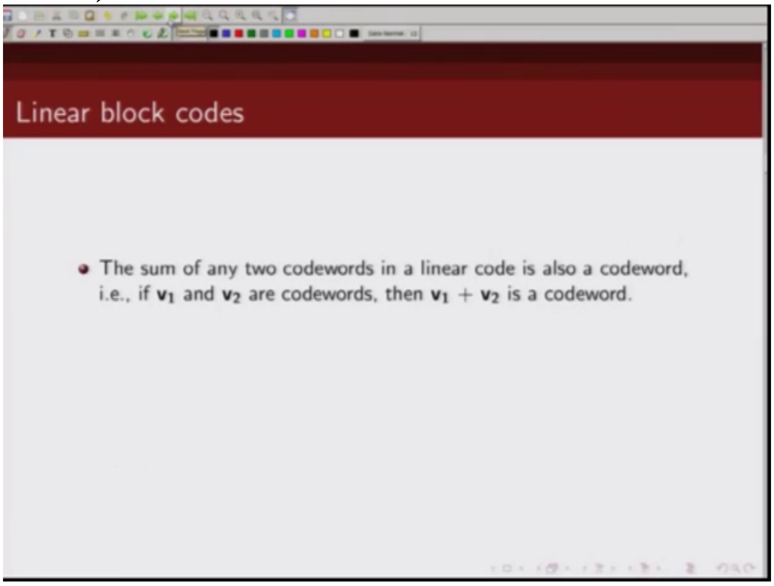
we are talking about binary linear block codes, so we will be doing this addition

(Refer Slide Time 03:35)



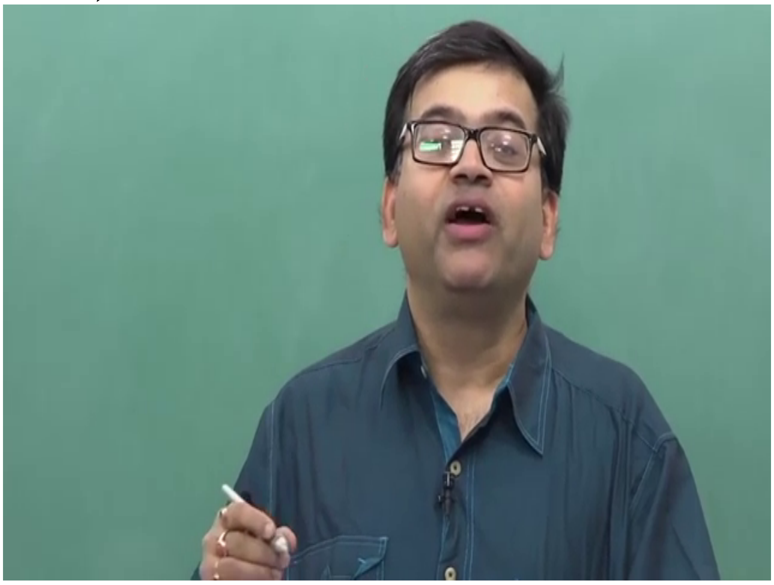
modulus 2 So what are the

(Refer Slide Time 03:38)



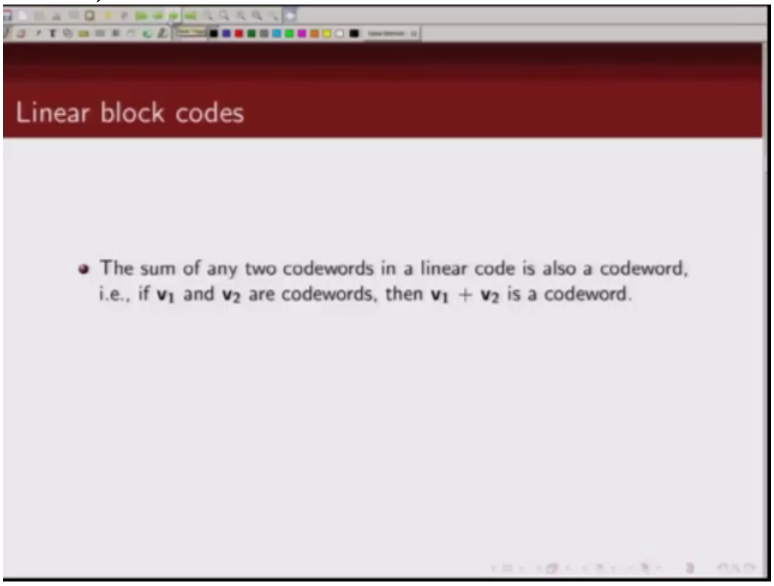
properties of linear block code? Sum of any two codewords in a linear code is also a valid codeword. So if v 1 and

(Refer Slide Time 03:48)

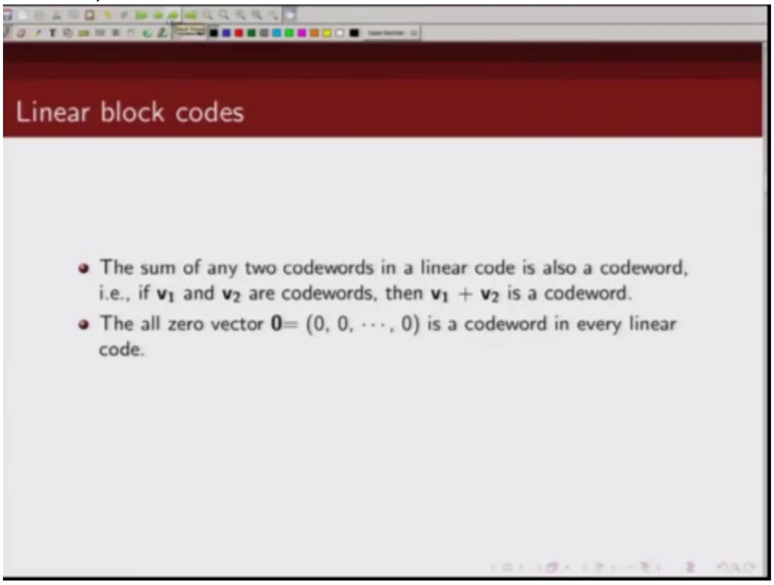


v 2 are valid codewords then v 1 plus v 2 will also be a valid codeword

(Refer Slide Time 03:58)

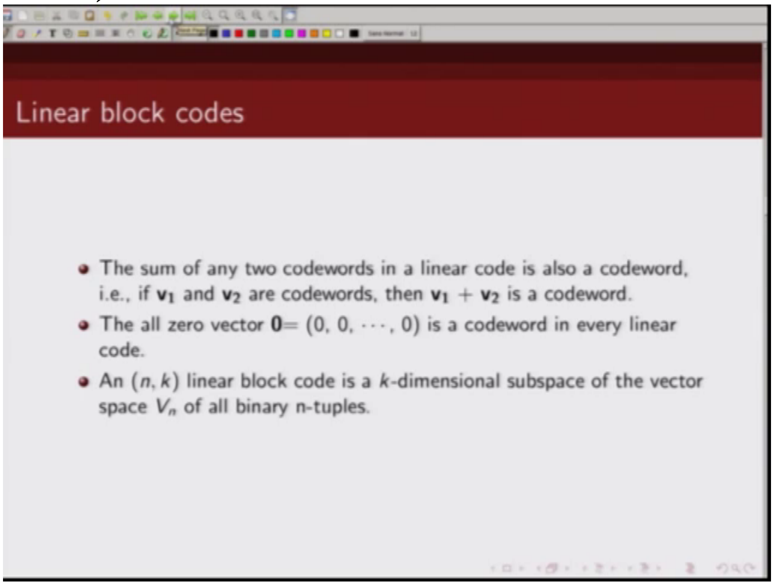


(Refer Slide Time 03:59)



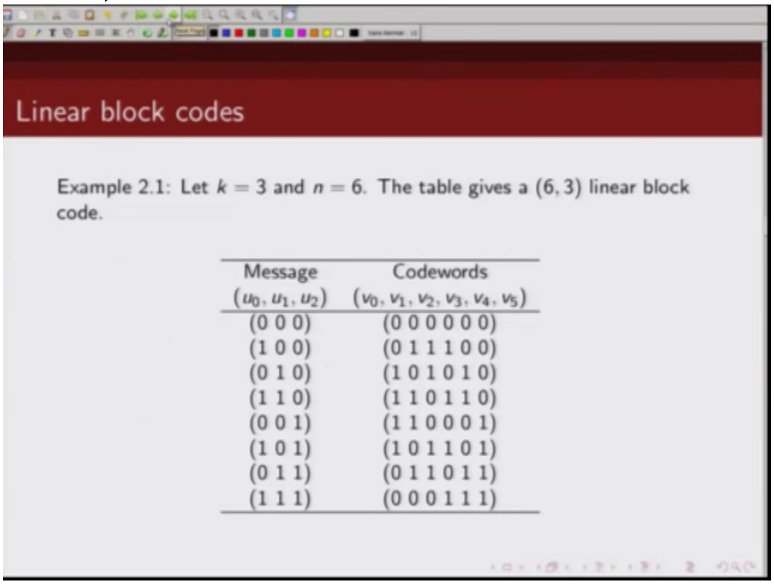
Also, an all zero codeword is a valid codeword in any linear block codes.

(Refer Slide Time 04:08)



So we can define a linear block code, n k linear block code as k dimensional space of vector space v n of all binary n-tuples; so we can define a linear binary block codes as a k dimensional subspace of vector space v n of all binary n tuples.

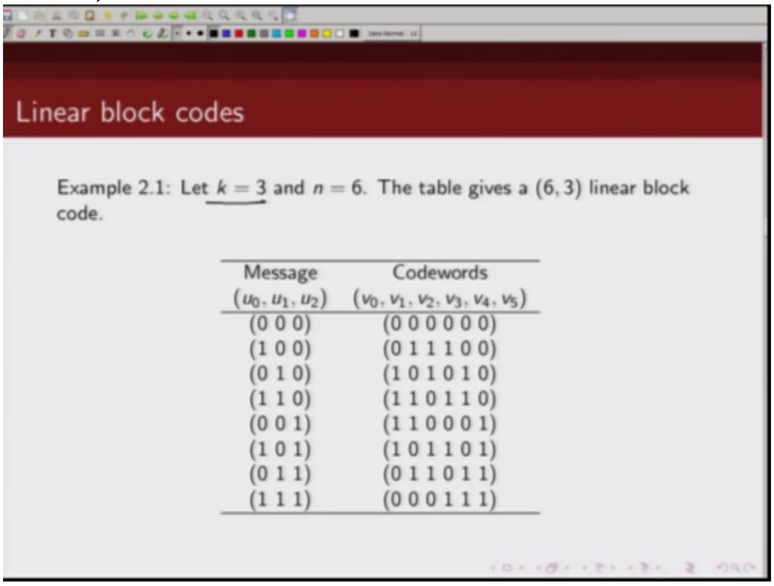
(Refer Slide Time 04:35)



Message (u_0, u_1, u_2)	Codewords $(v_0, v_1, v_2, v_3, v_4, v_5)$
(0 0 0)	(0 0 0 0 0 0)
(1 0 0)	(0 1 1 1 0 0)
(0 1 0)	(1 0 1 0 1 0)
(1 1 0)	(1 1 0 1 1 0)
(0 0 1)	(1 1 0 0 0 1)
(1 0 1)	(1 0 1 1 0 1)
(0 1 1)	(0 1 1 0 1 1)
(1 1 1)	(0 0 0 1 1 1)

Now let us take an example to illustrate what is a generator matrix. So in this example, we have considered 3

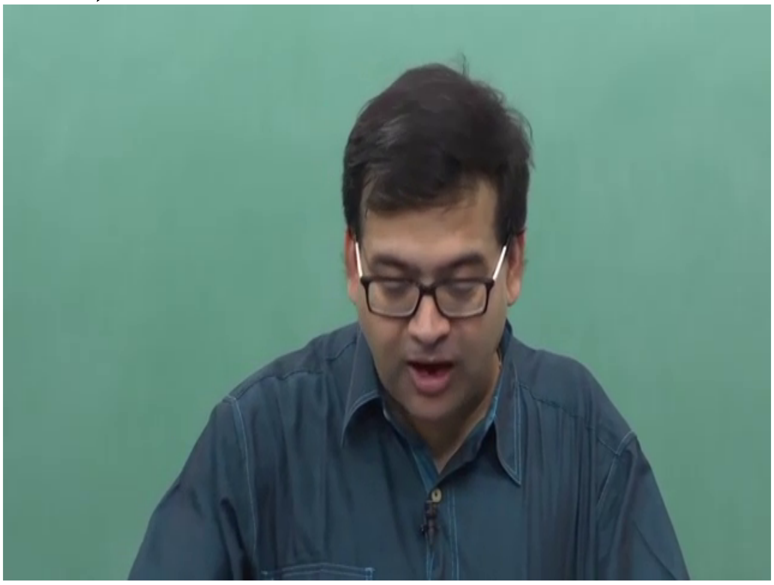
(Refer Slide Time 04:45)



Message (u_0, u_1, u_2)	Codewords $(v_0, v_1, v_2, v_3, v_4, v_5)$
(0 0 0)	(0 0 0 0 0 0)
(1 0 0)	(0 1 1 1 0 0)
(0 1 0)	(1 0 1 0 1 0)
(1 1 0)	(1 1 0 1 1 0)
(0 0 1)	(1 1 0 0 0 1)
(1 0 1)	(1 0 1 1 0 1)
(0 1 1)	(0 1 1 0 1 1)
(1 1 1)	(0 0 0 1 1 1)

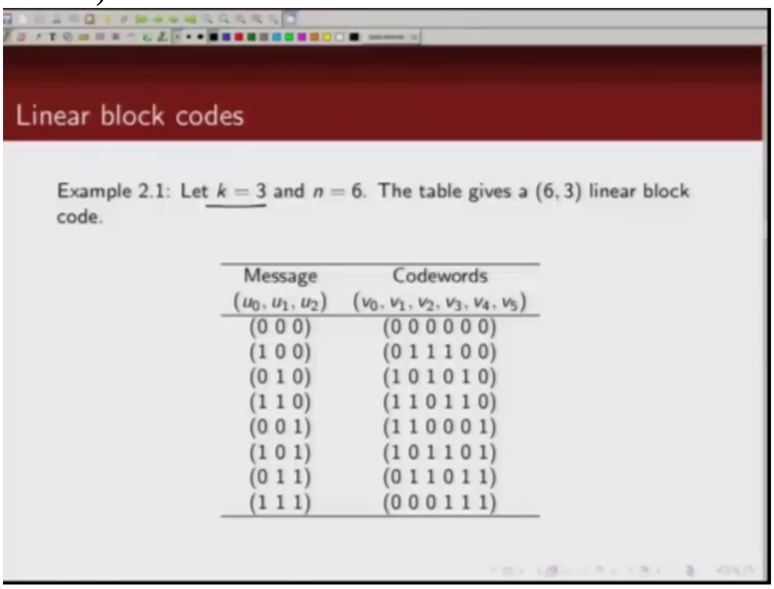
information bits and 6 coded bits. And in this table I have given you the set of 8 information sequences and their corresponding codewords. So how do we find the generator matrix in this case? So we will have to look at each of this code bits and see how are we generating these code bits in terms of

(Refer Slide Time 05:16)



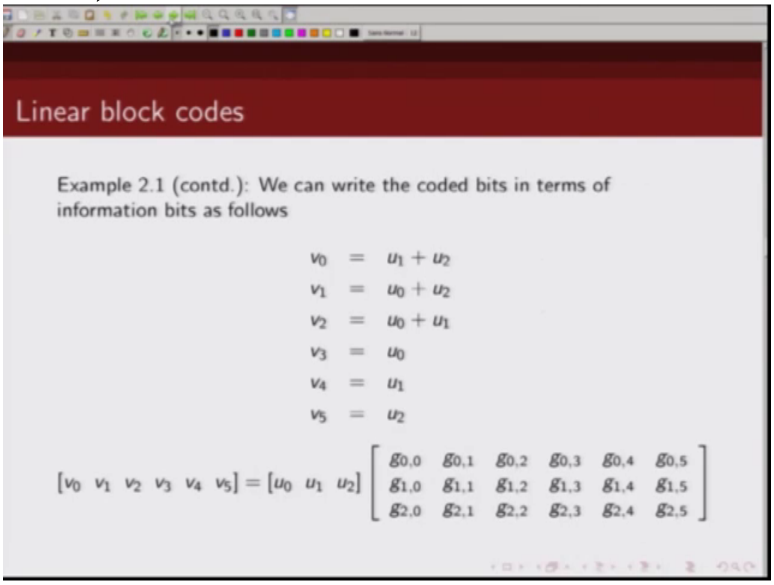
message bits u 0, u 1 and u 2. So first thing we are going to do is

(Refer Slide Time 05:24)



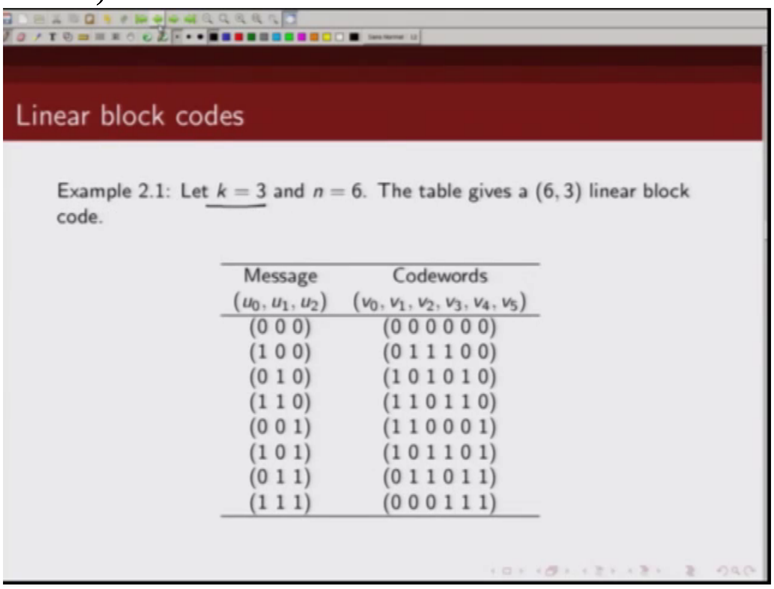
look at each of these code bits, v 0, v 1, v 2, v 3, v 4, v 5 and write them in terms of u 0, u 1, u 2, Ok. So let's look at each of these. So v 0 is u 1 plus u 2, we can see easily v 0 is this column and we can see this is same as u 1 plus u 2. So u 1 plus u 2 in this case is 0, u 1 plus u 2 is 0, 1 plus 0 is 1, 1 plus 0 plus 1 is 1, 1 plus 1 is 0 modular 2 and 1 plus 1 is 0 modular 2. So this v 0 is basically given by

(Refer Slide Time 06:12)



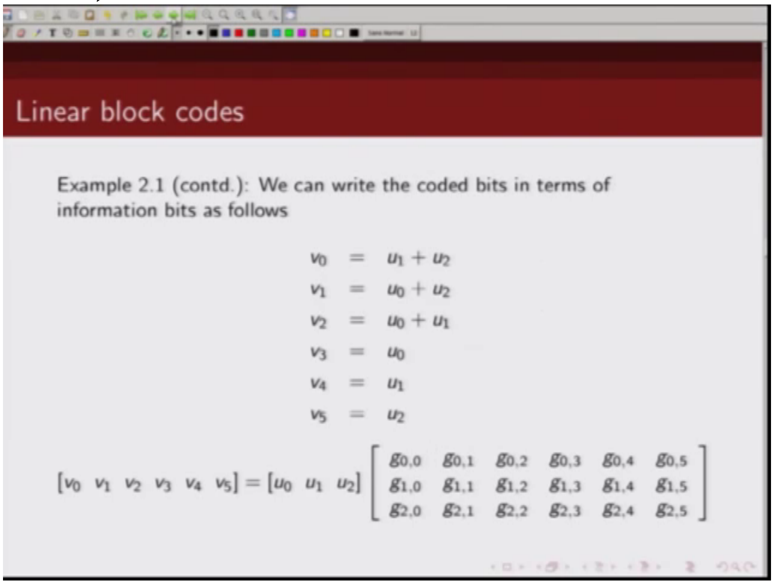
u 1 plus u 2 Similarly we can see v 1 is given by u 0 plus u 2 and v 2 is given by u 0 plus u 1. So let's just

(Refer Slide Time 06:25)



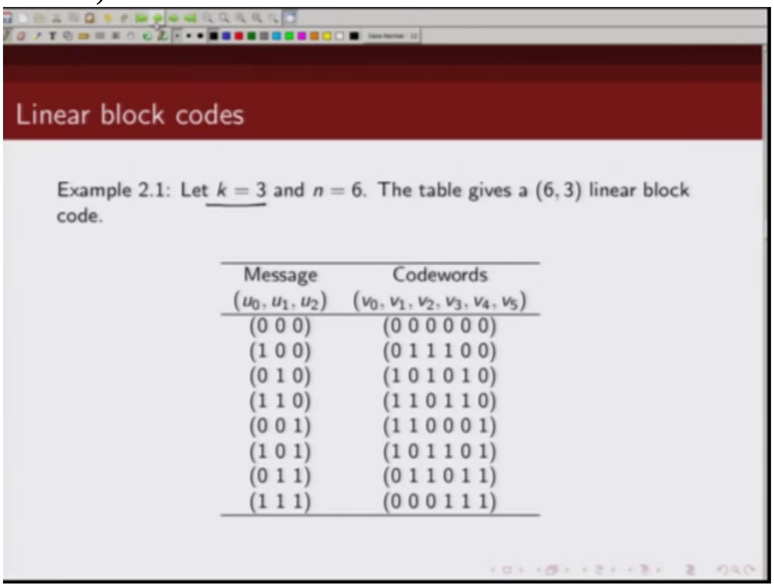
check, let's say v 2. v 2 you can see, is given by, v 2 is

(Refer Slide Time 06:33)



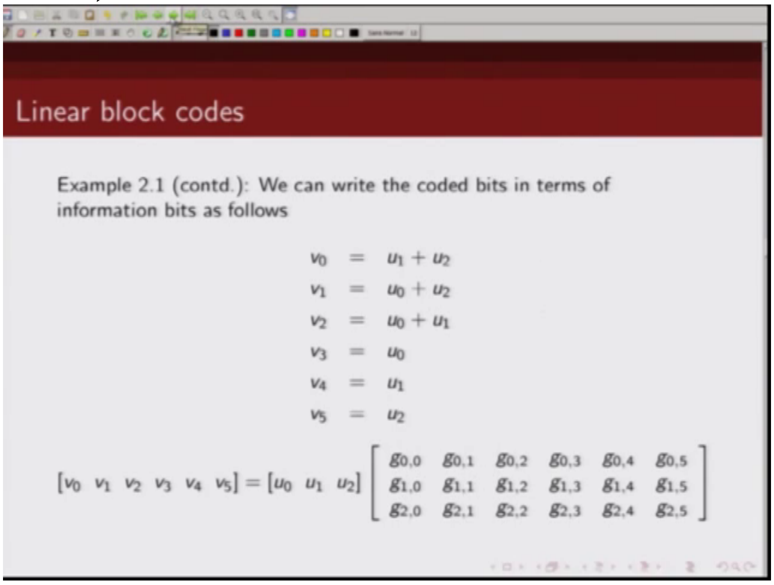
given by u 0 plus u 1 You can check v 2

(Refer Slide Time 06:36)



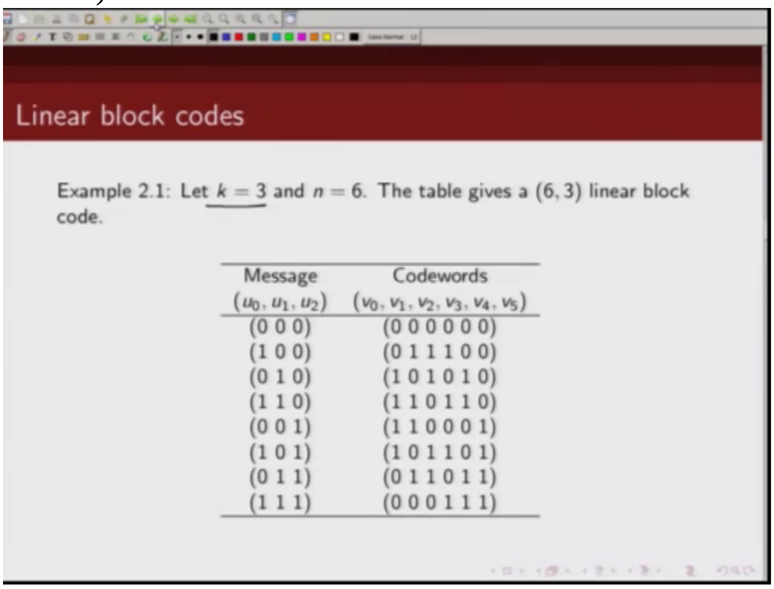
is given by, so u 0 plus u 1, 0 plus 0 is 0, 1 plus 0 is 1, 0 plus 1 is 1, 1 plus 1 is 0, 0 plus 0 is 0, 1 plus 0 is 1, 0 plus 1 is 1, and 1 plus 1 is 0. Similarly

(Refer Slide Time 06:58)



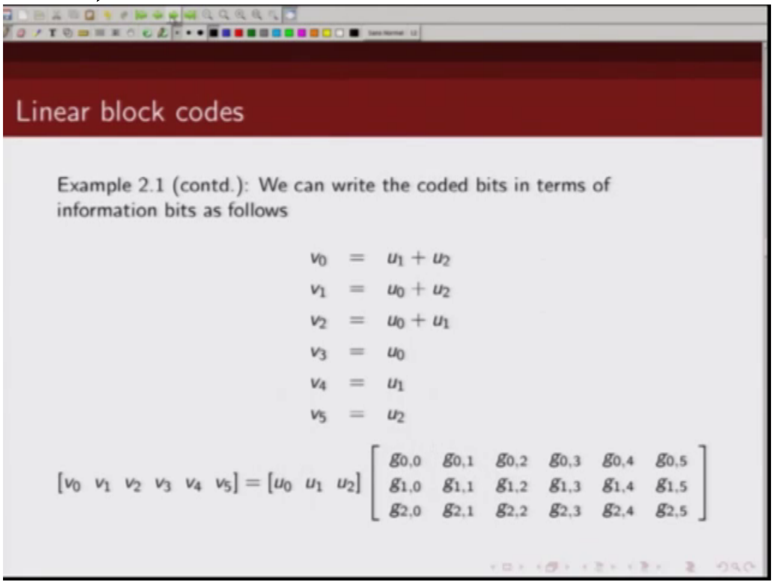
we notice that v 3, v 4, v 5 are nothing but information bits u 0, u 1 and u 2 respectively. So let's go back.

(Refer Slide Time 07:10)



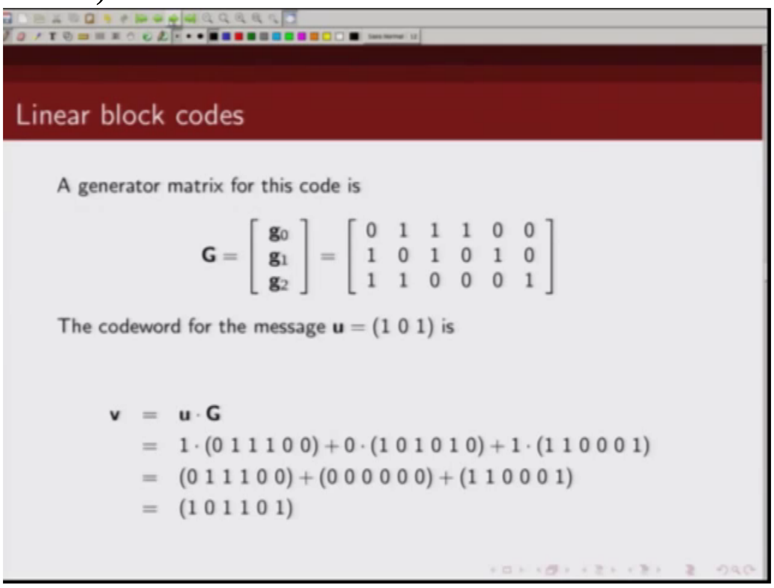
v 3 is this column and we can see this is same as u 0. 0 1 0 1 0 1 0 1, similarly v 4 is equal to u 1 and v 5 is same as u 2. So now

(Refer Slide Time 07:31)



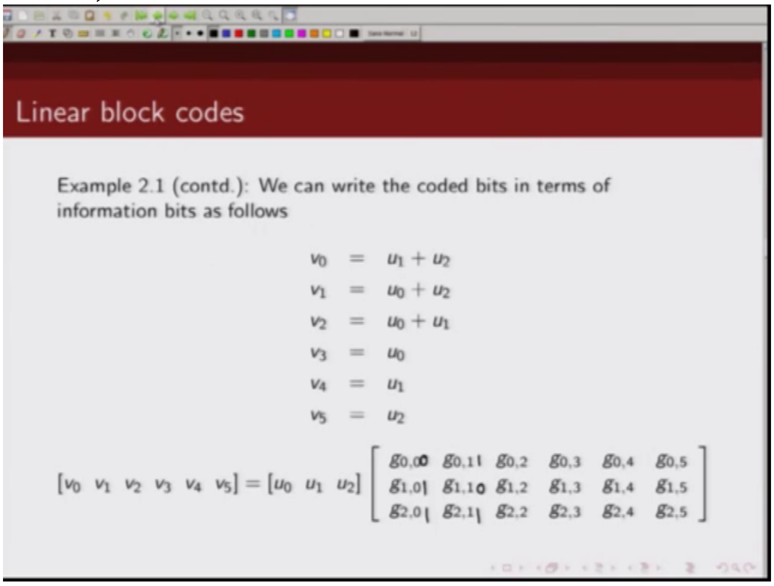
we have written our coded bits in terms of our information bits. This set of 6 equations I can write it in a matrix form. So I can write my coded bits in terms of information bits and this matrix G which is our generator matrix will tell us how are we generating each of these coded bits as a linear combination of these information bits. So if we compare each equation, let's look at v 0. So what is v 0? v 0 is u 0 g 0 0 plus u 1 g 1 0 plus u 2 g 2 0 and what do we see here? v 0 is u 1 plus u 2. So that means g 0 0 is 0 because there is no u 0 term here. g 1 0 is 1 because there is a u 1 term here and g 2 0 is 1 because there is a u 2 term here. So this will be 0 1 1. Similarly look at v 1. v 1 is u 0 g 0 1 plus u 1 g 1 1 plus u 2 g 2 1. And if we compare it with v 1 here we see v 1 is u 0 plus u 2 that means this g 0 1 should be 1, g 1 1 should be 0, and this should be 1. Likewise we build up the other columns of the matrix. So if we do that

(Refer Slide Time 09:27)



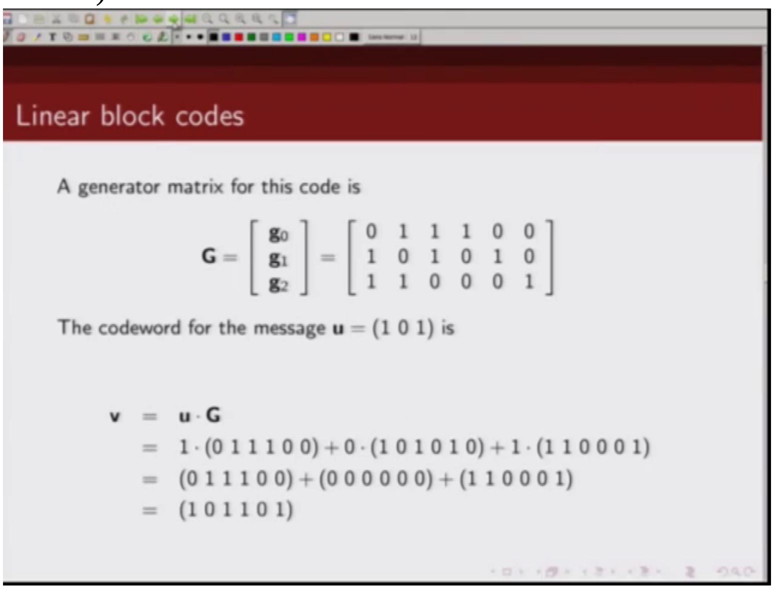
what we get is something like this. We can verify basically. Let's take second last column,

(Refer Slide Time 09:37)



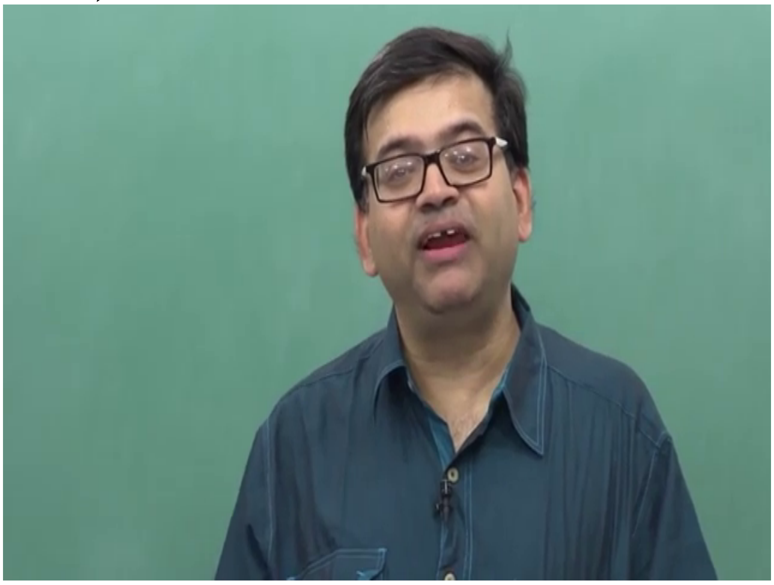
second last column so what is v 4. v 4 is u 0 times g 0 4 plus u 1 times g 1 4 plus u 2 times g 2 4 and what is v 4, v 4 is u 1 so then this should be 1 and this should be 0 and this should be 0 and this is what we have,

(Refer Slide Time 10:01)



0 1 0 So now we can basically find out

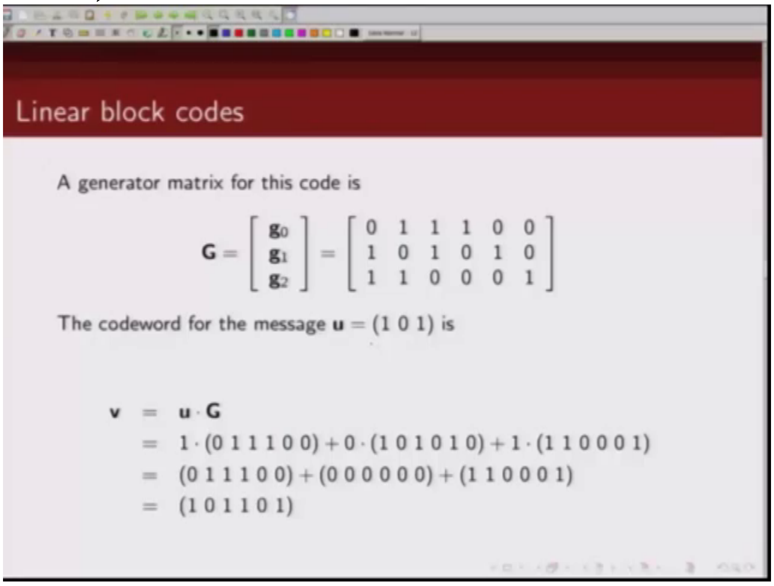
(Refer Slide Time 10:06)



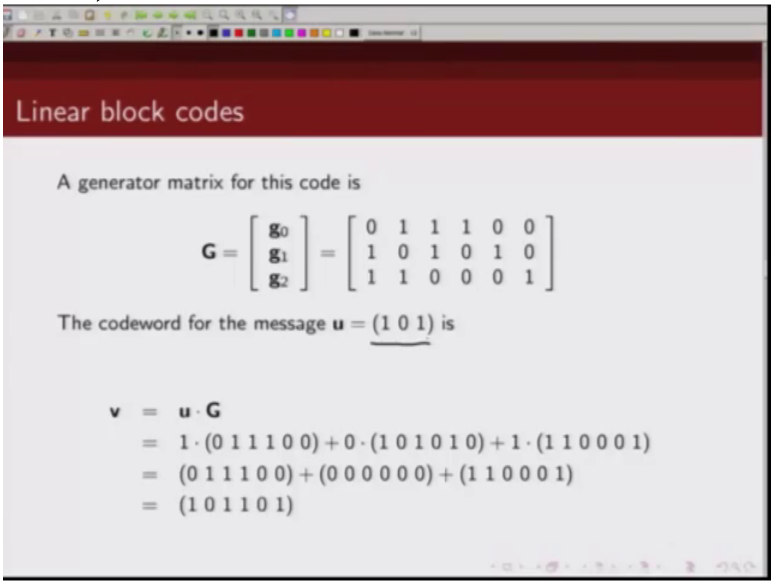
the generator matrix; so linear block code is completely described by its generator matrix And we said we can use the generator matrix to generate our codewords.

For example, if my information sequence is

(Refer Slide Time 10:25)

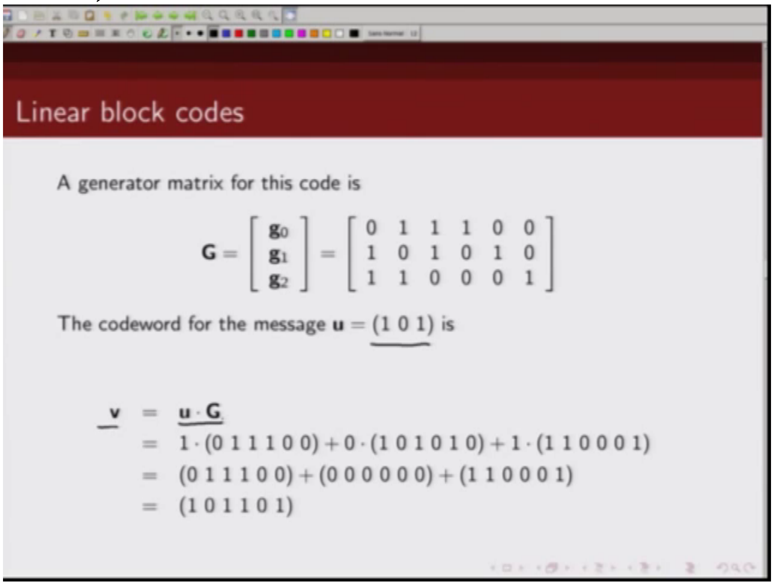


(Refer Slide Time 10:27)



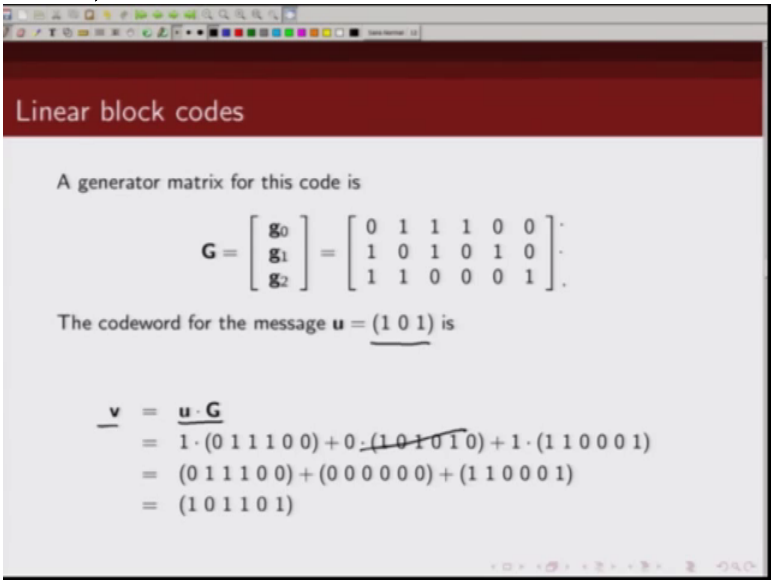
1 0 1, what should be my corresponding coded bits for the information sequence 1 0 1? How do I find that? So as I know, my output codeword is basically u times G

(Refer Slide Time 10:41)



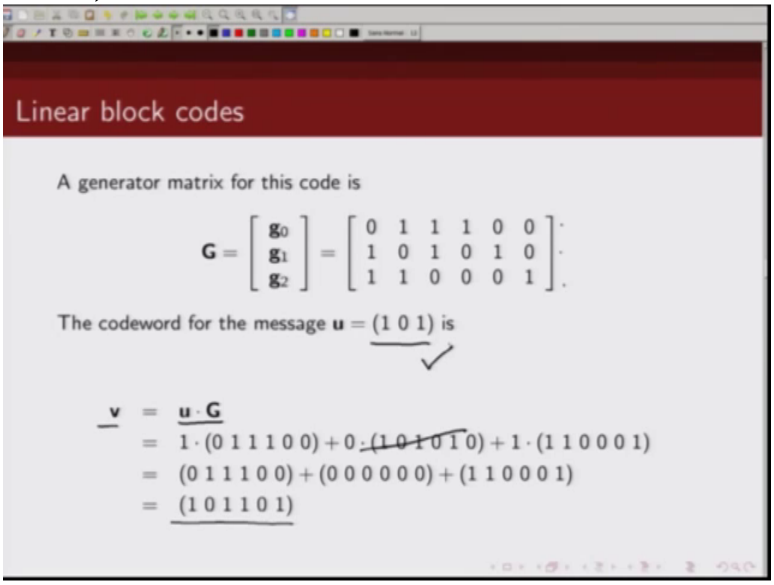
so I will take linear combinations of rows of my generator matrix. What are the rows of my generator matrix? These are the 3 rows of my generator matrix. So my coded bit corresponding to this information sequence would be 1 times G 0 plus 0 times G 1 plus 1 times G 2. So that's what I have written here, 1 times G 0, 0 times G 1 plus 1 times G 2. So this is basically 0. So what I have

(Refer Slide Time 11:19)



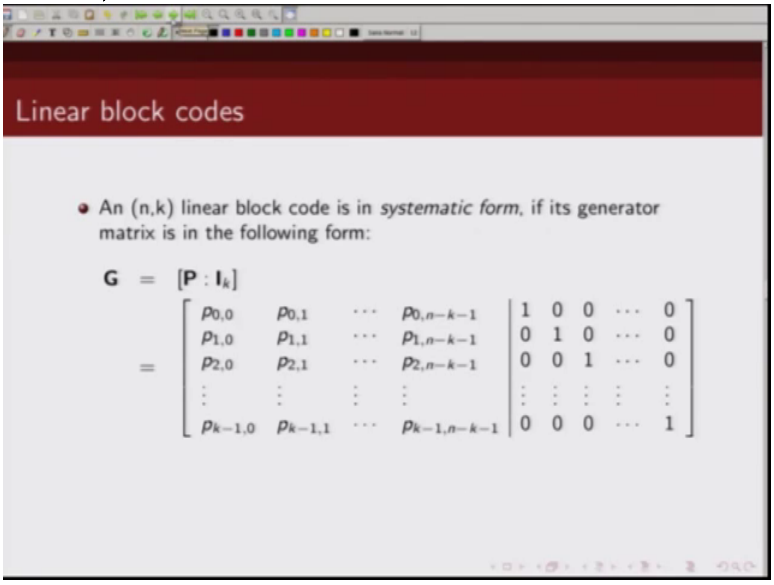
is then, this plus this right? So let's look at 0 plus 1 would be 1, 1 plus 1 would be 0, 1 plus 0 is 1, 1 plus 0 is 1, 0 plus 0 is 0 and 0 plus 1 is 1. So my codeword corresponding to this information

(Refer Slide Time 11:42)



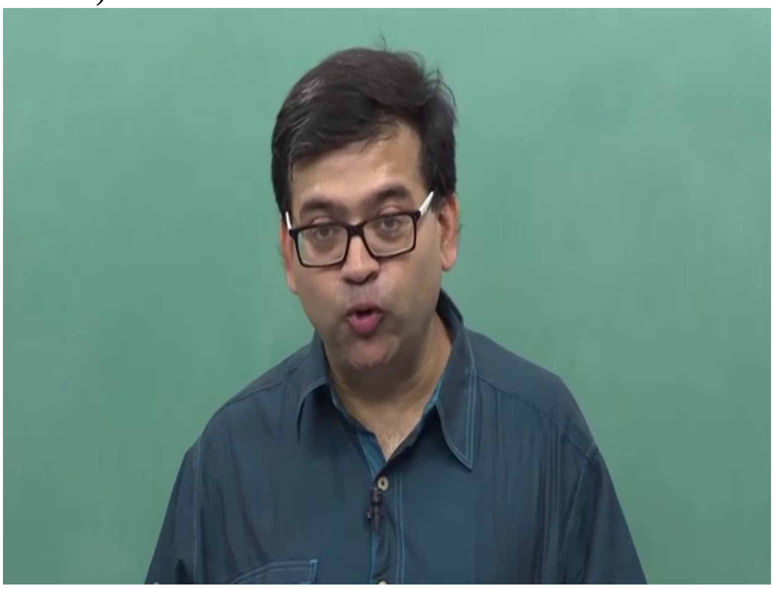
message bits, this information bit is given by 1 0 1 1 0 1, Ok?

(Refer Slide Time 11:52)



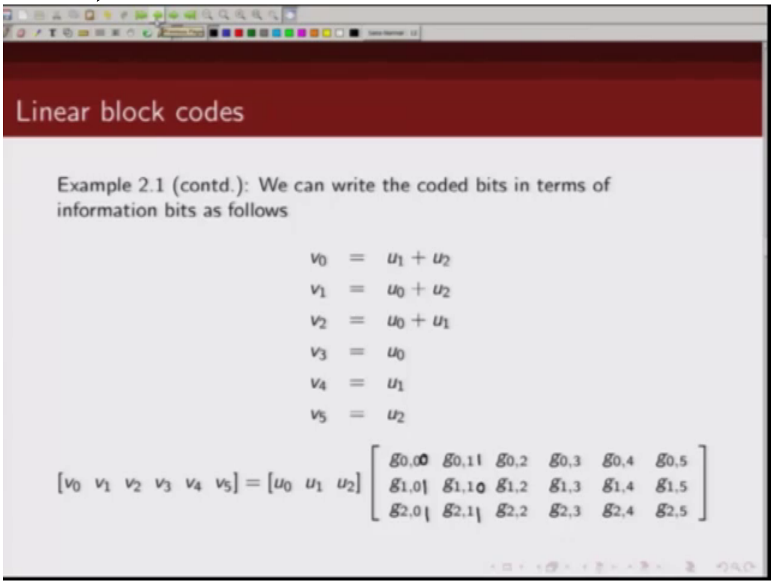
Now what do we mean by a linear code in systematic form? Now if we are able to,

(Refer Slide Time 12:01)



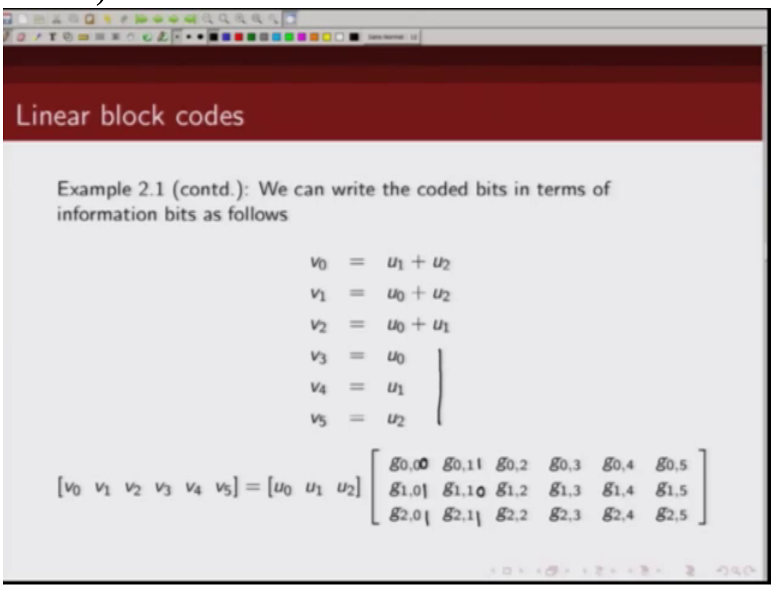
among the coded bits if we are able to separate them out into, if the message bits appear directly in the coded bit sequence then we can separate out the message bits from the parity bits. For example, go back to this example.

(Refer Slide Time 12:22)



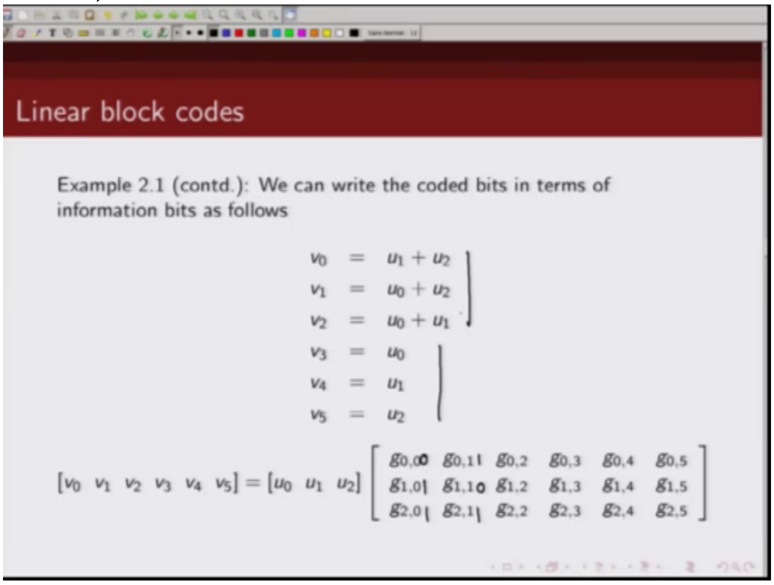
What do we have here? We have 3 of these coded bits

(Refer Slide Time 12:27)



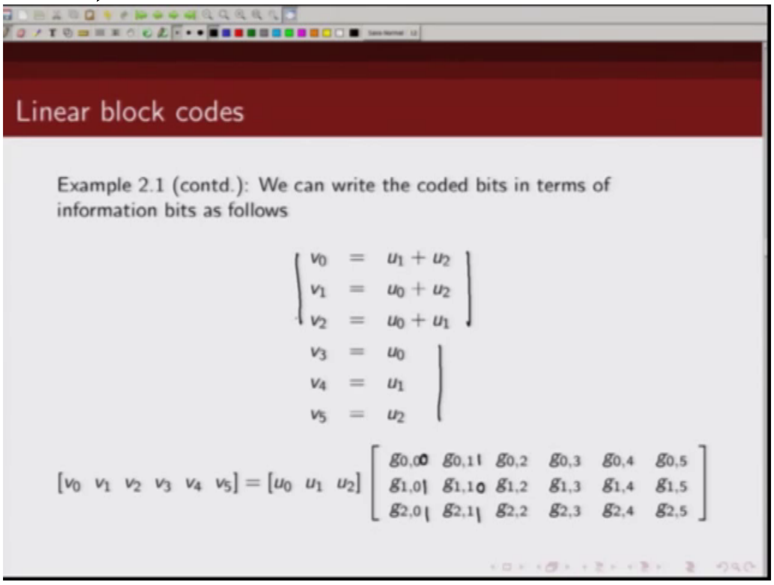
exactly same as information bits, and the other 3 bits,

(Refer Slide Time 12:33)



parity bits are linear combination of this message bits. So from the output codeword we, we can clearly separate out the information sequence which is in this case given by v 3, v 4 and v 5. So in this case, v 1 v 2 v 3

(Refer Slide Time 12:56)

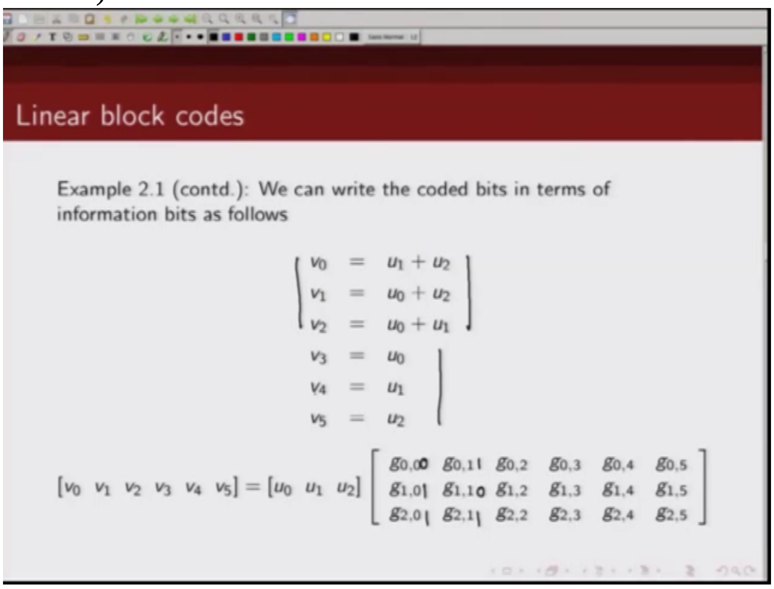


are these n minus k parity bits

(Refer Slide Time 12:59)

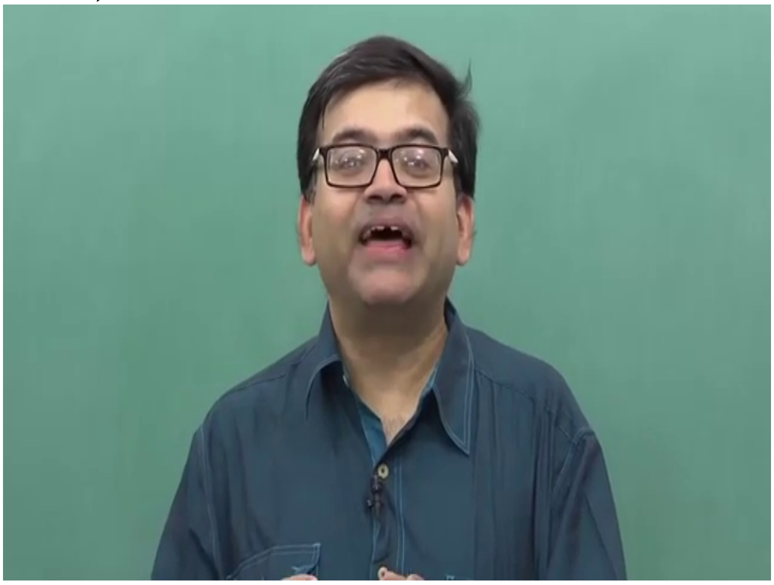
and v 3 v 4 v 5 are my

(Refer Slide Time 13:02)



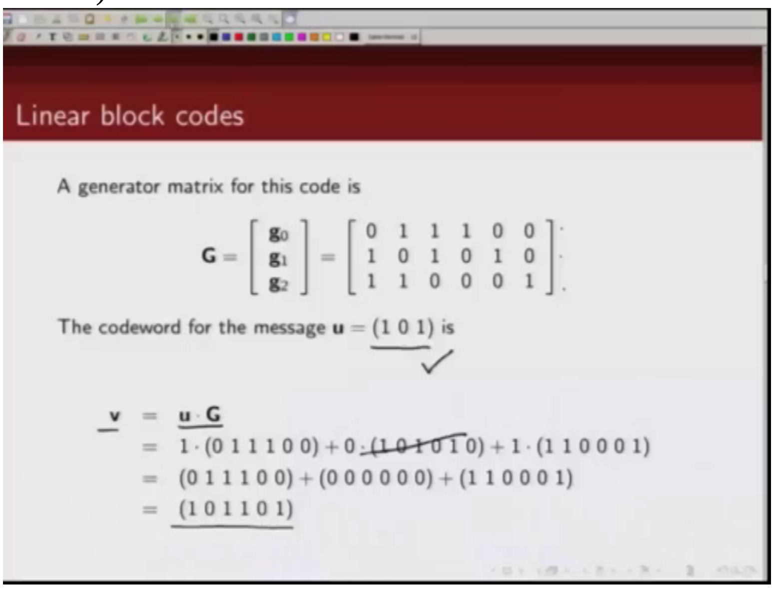
information bits So in this particular example, we can see that

(Refer Slide Time 13:07)

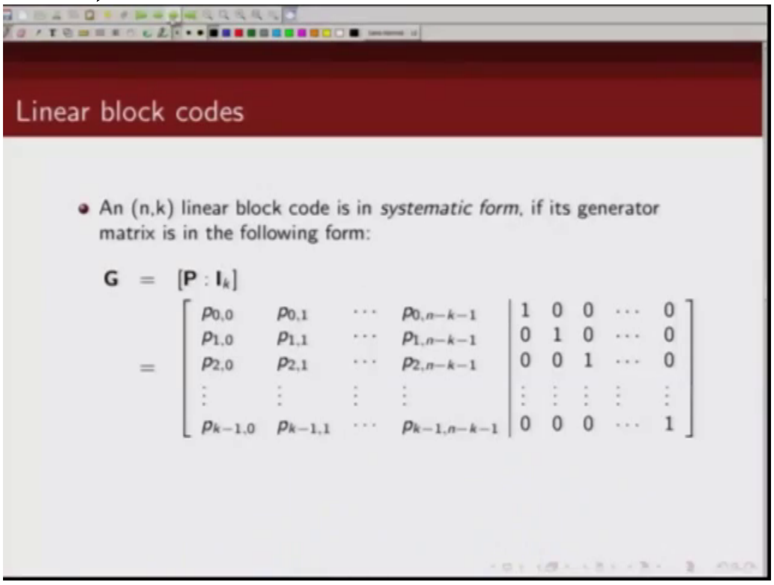


that we are able to separate out information bits directly from the coded bits. So in a systematic,

(Refer Slide Time 13:16)

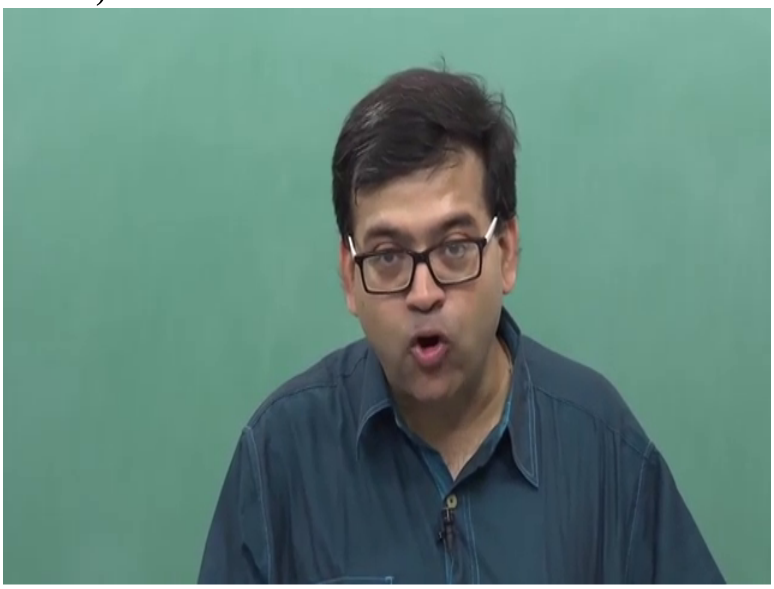


(Refer Slide Time 13:17)



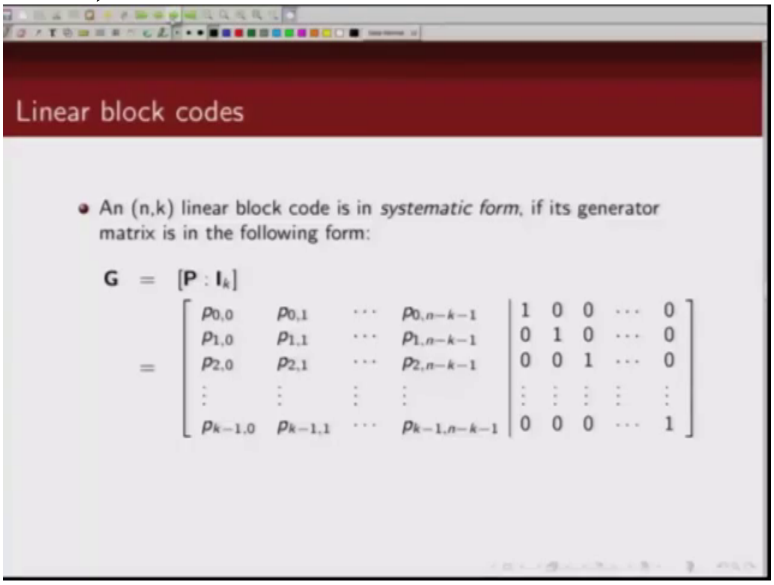
a block code in a systematic form, we are able to separate out

(Refer Slide Time 13:22)



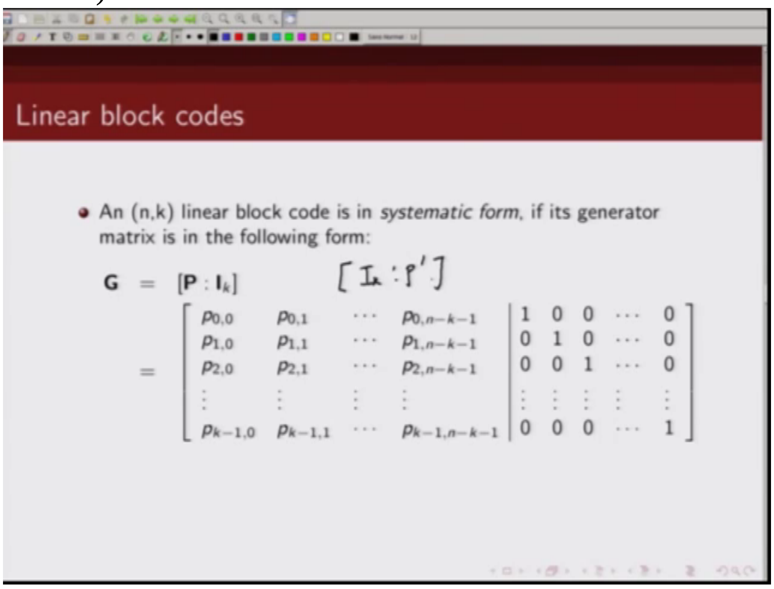
out the information bit part from the coded bits So a

(Refer Slide Time 13:28)



generator matrix for a linear block code in systematic form will be of the form like this or it would be basically i times i k times some some,

(Refer Slide Time 13:47)

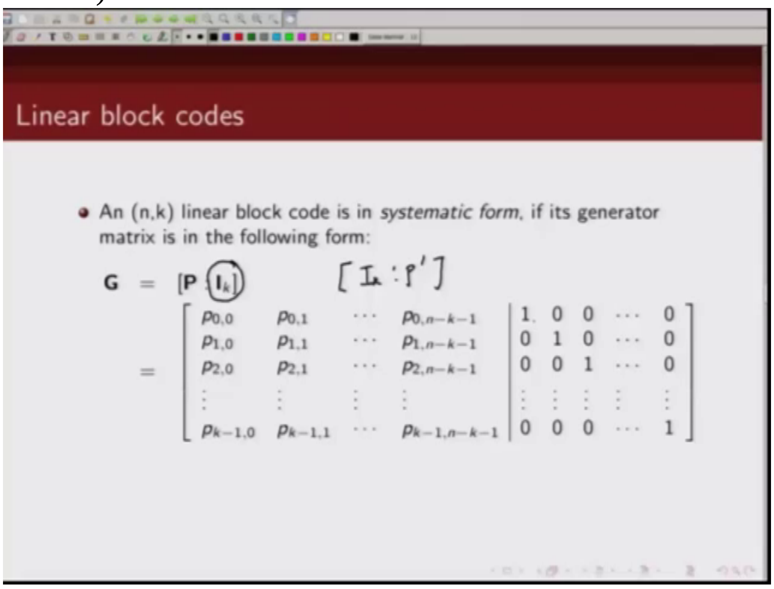


either of this form. Now why do we say that? So only when we have our, part of our generator matrix of the form of identity, then what is going to happen? When we multiply our information sequence

(Refer Slide Time 14:06)

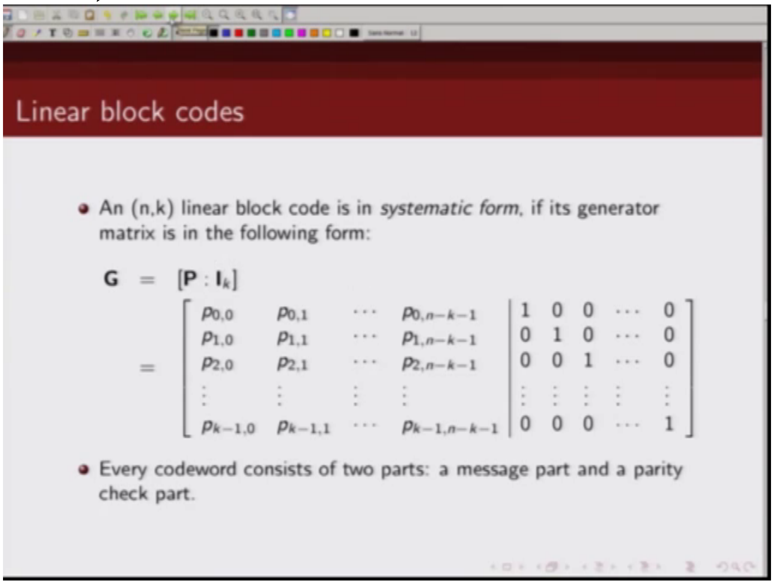
with this sort of generator matrix you will see part of

(Refer Slide Time 14:09)



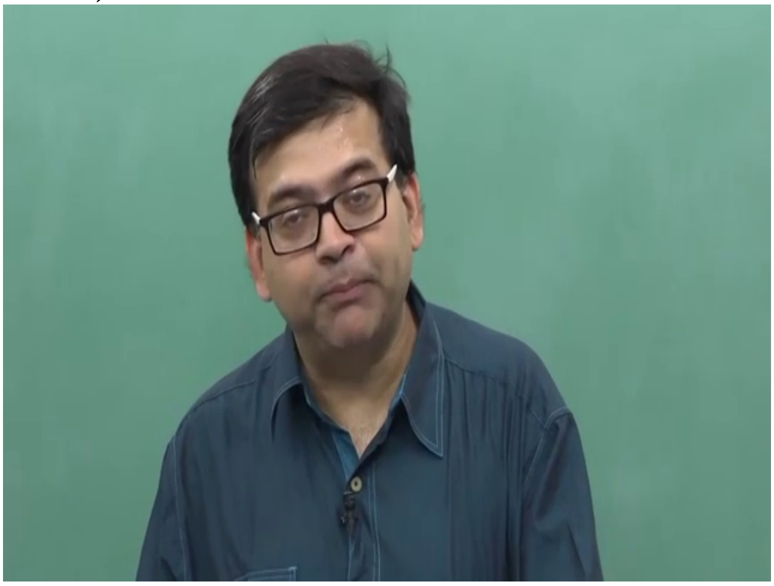
my coded bits will just depend on one particular information bit sequence. So if I have write down the corresponding equations for coded sequence

(Refer Slide Time 14:24)



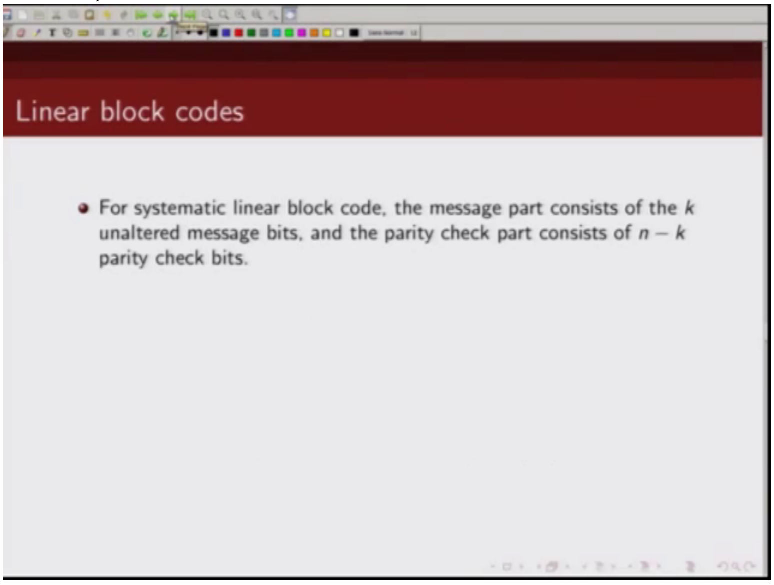
what you will see that some coded bits directly depend on the

(Refer Slide Time 14:31)



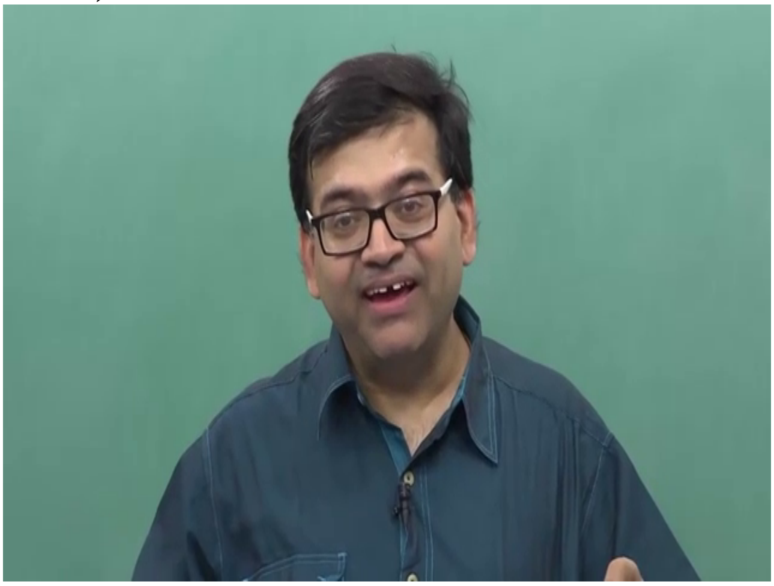
message bits and then rest are, which are parity bits are linear combination of these message bits. So in a systematic form basically we can separate out the message part from the parity bit part. So

(Refer Slide Time 14:49)



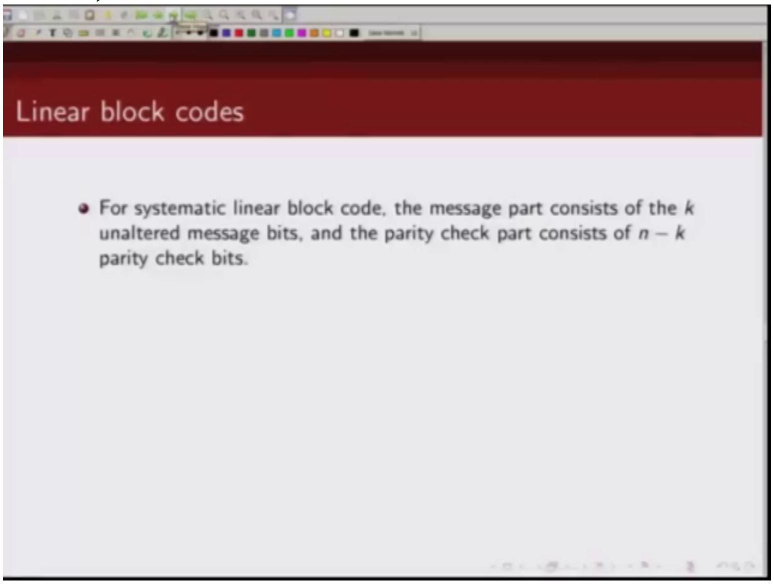
as I said, for a systematic linear block codes, the message part will consist of the k information

(Refer Slide Time 14:59)



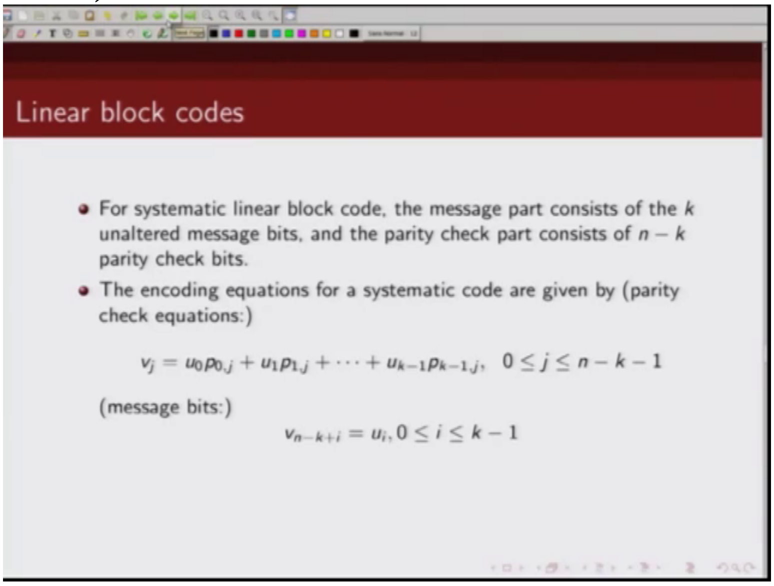
bits and the remaining n minus k bits which are the parity bits,

(Refer Slide Time 15:05)



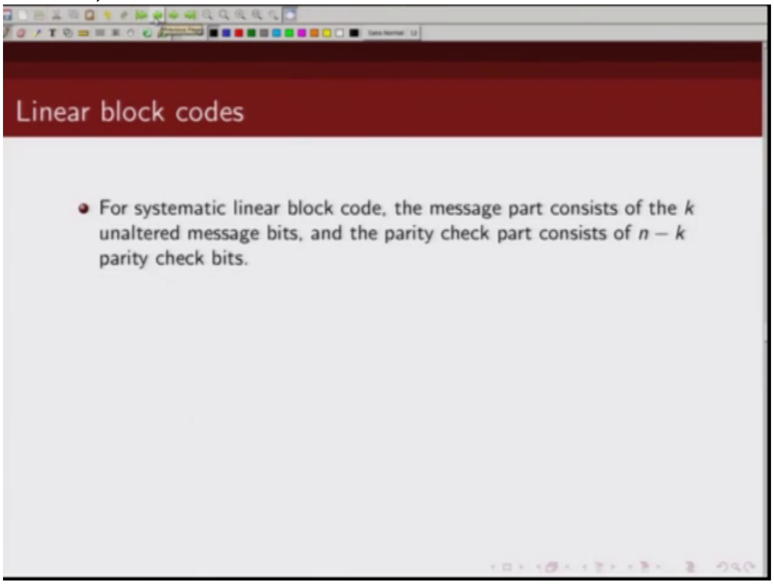
basically they will be linear combination of these message bits. So we

(Refer Slide Time 15:11)



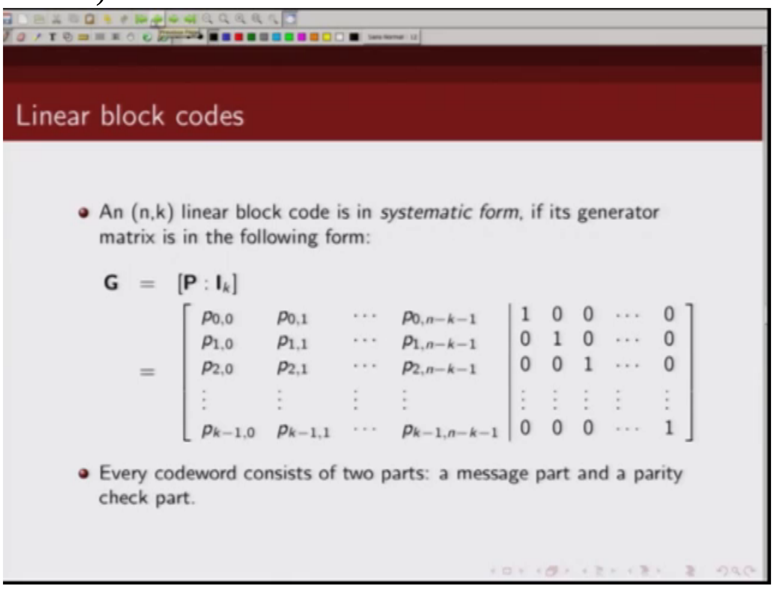
can write down the encoding equations for these matrices, for these, for systematic code, so if you look at

(Refer Slide Time 15:19)



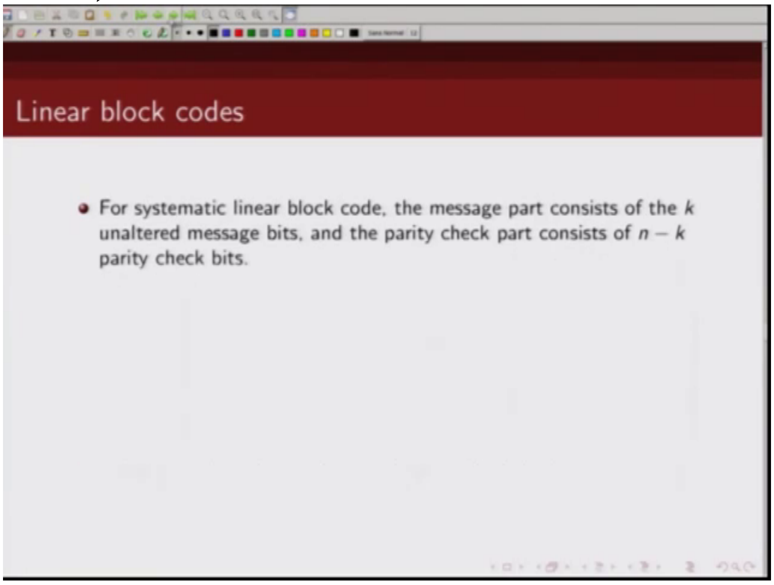
what is our

(Refer Slide Time 15:21)



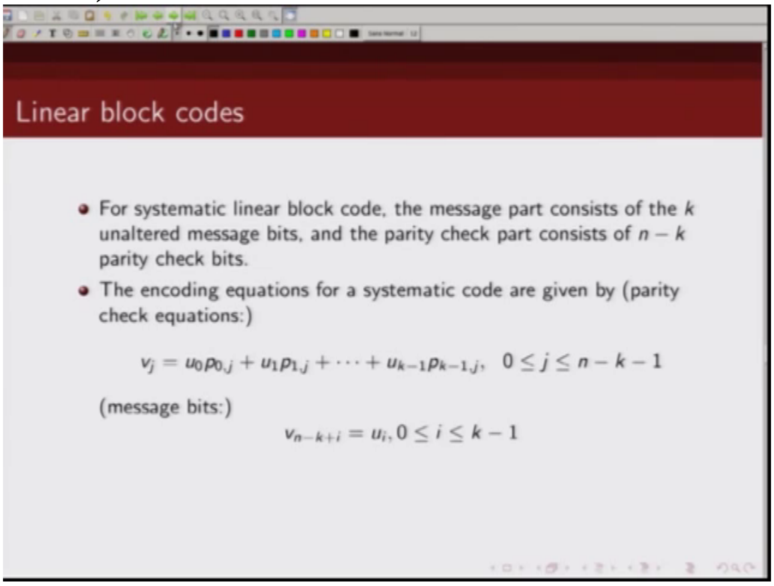
encoding equation? Our v is u times G where G is of form like this, Ok? So if we write u which is basically u 0, u 1 to u k minus 1 times this G matrix, what we will get is

(Refer Slide Time 15:46)



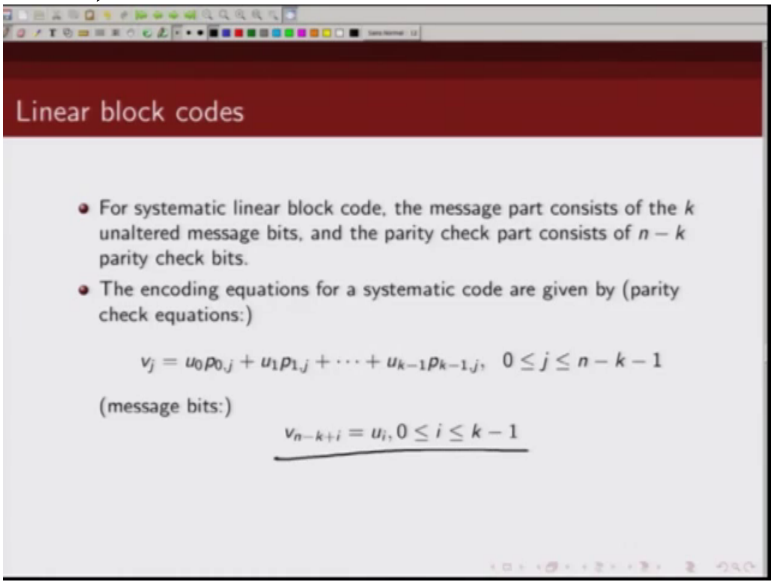
a form

(Refer Slide Time 15:47)



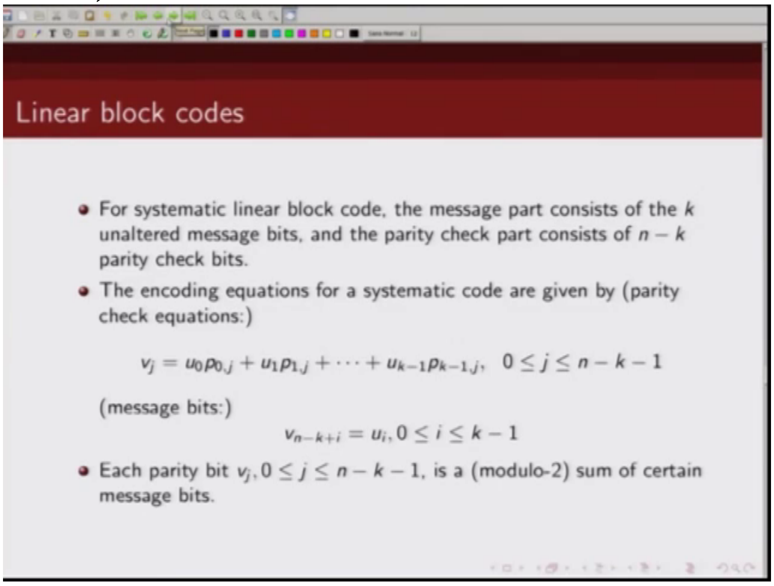
like this. So you will have n minus k parity equations, parity check equations which are given by this expression and then you will have remaining

(Refer Slide Time 15:59)



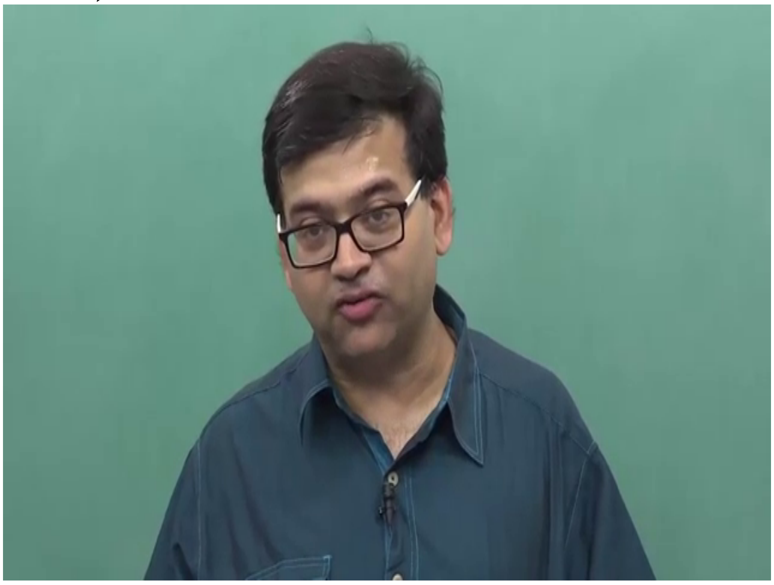
k unaltered message bits. So for a linear block code in a systematic form the encoding equations will be of this form. And as

(Refer Slide Time 16:13)



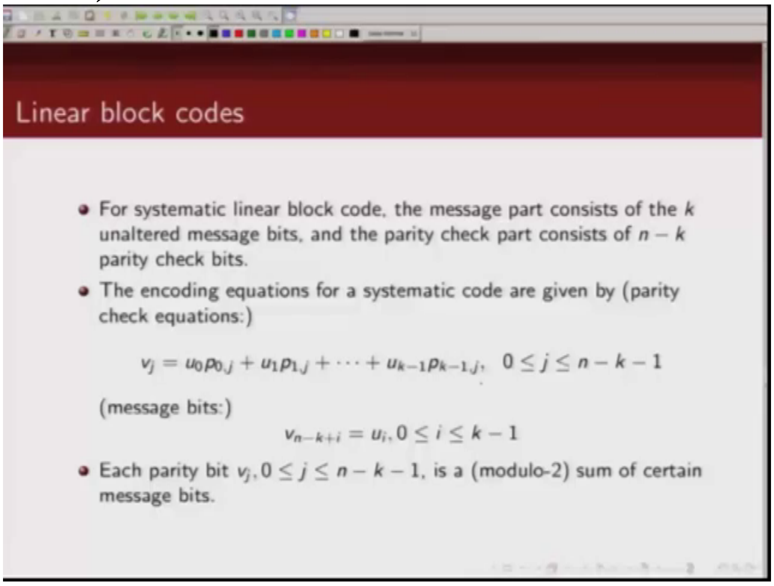
as I said since we are restricting ourselves without any loss of generality

(Refer Slide Time 16:17)



to binary codewords, this addition is basically done

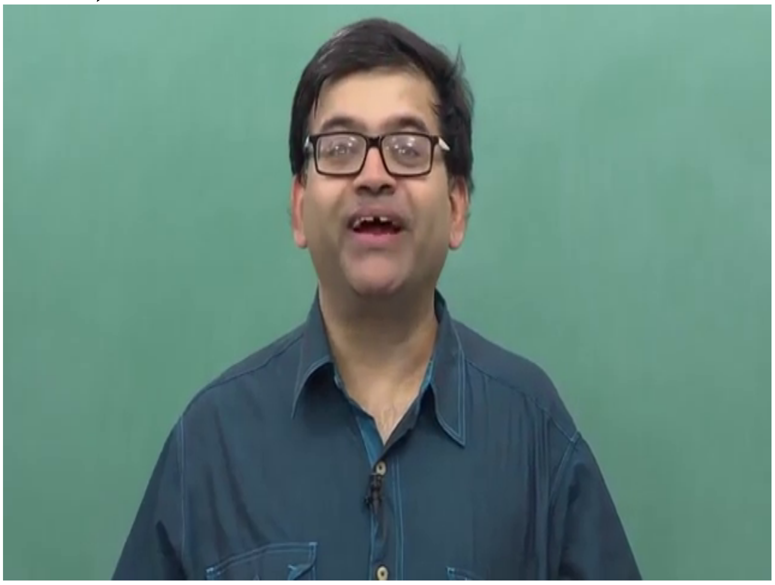
(Refer Slide Time 16:23)



modulo 2

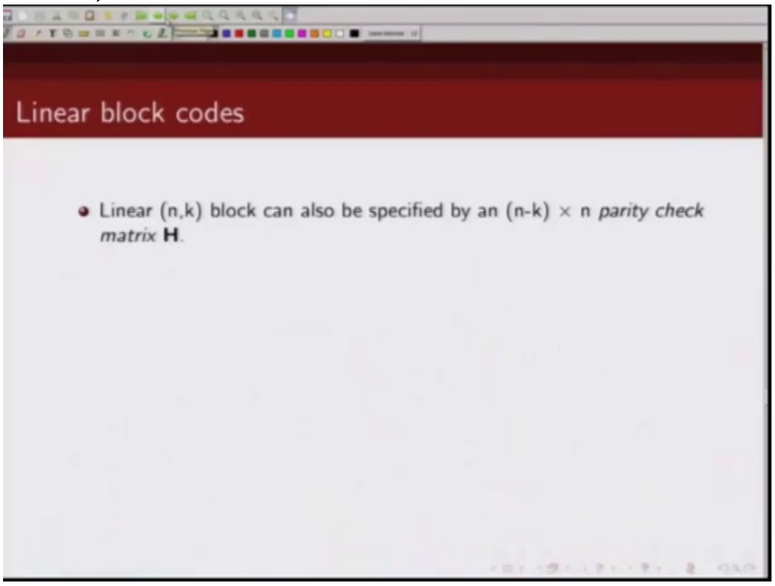
So what we have seen so far is we can describe

(Refer Slide Time 16:28)



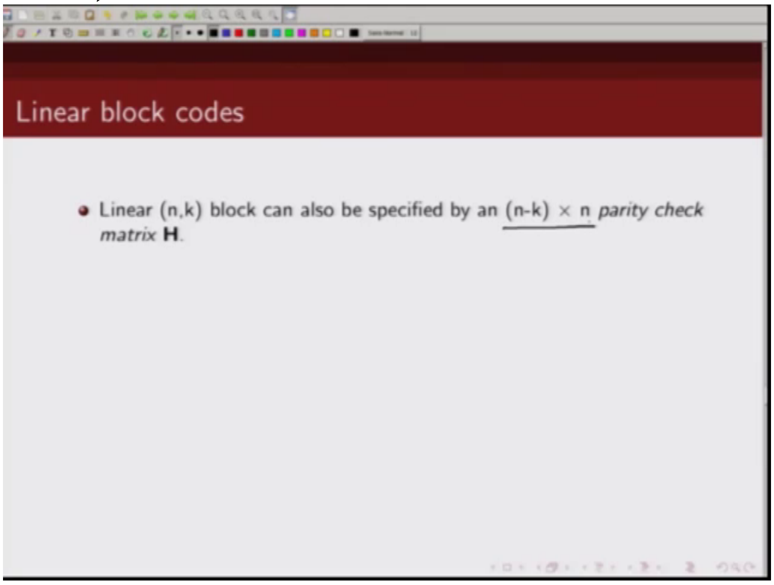
a linear block code by its generator matrix which is a k cross n matrix and we can use this generator matrix to generate our set of codewords. Now there is another matrix which we call parity check matrix which is related to our generator matrix, we will show, which can also be used to completely describe a linear block code. So for a n k linear block code

(Refer Slide Time 17:03)



can be specified by a n minus k cross n

(Refer Slide Time 17:09)

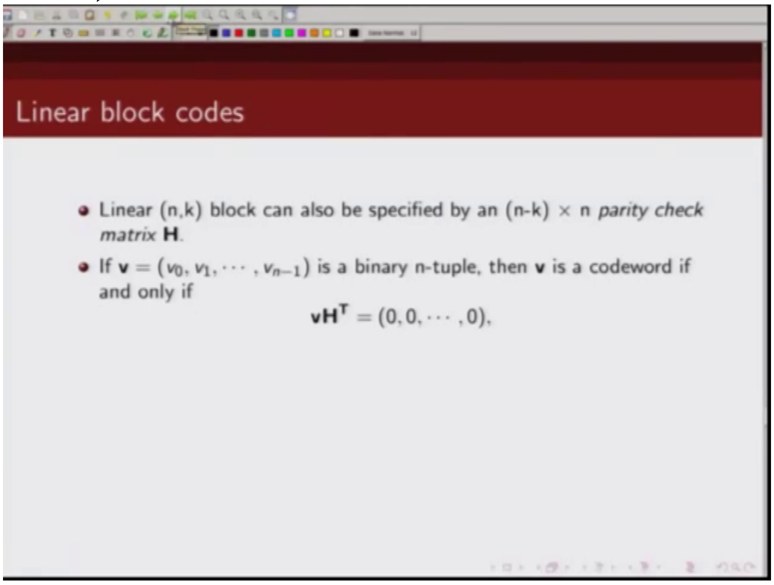


parity check matrix which we denote by G, H, the generator matrix we denote by G and the parity

(Refer Slide Time 17:17)

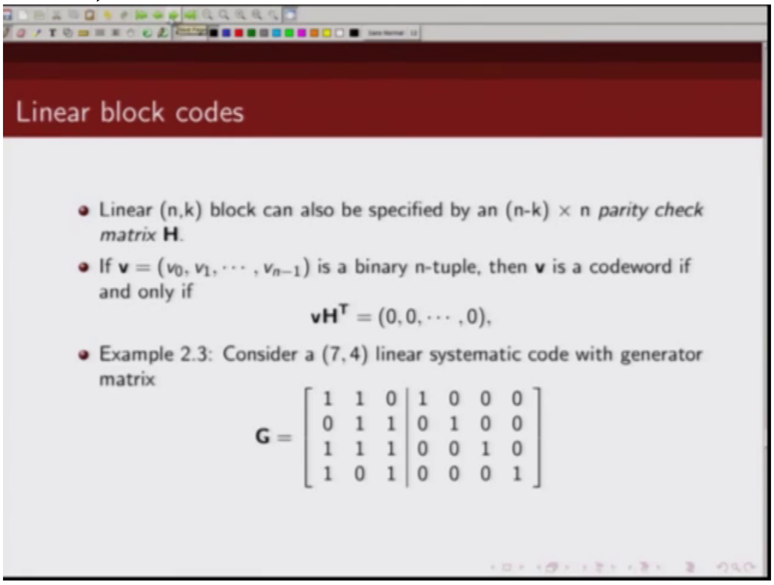
check matrix we denote by H. Now this parity check

(Refer Slide Time 17:22)



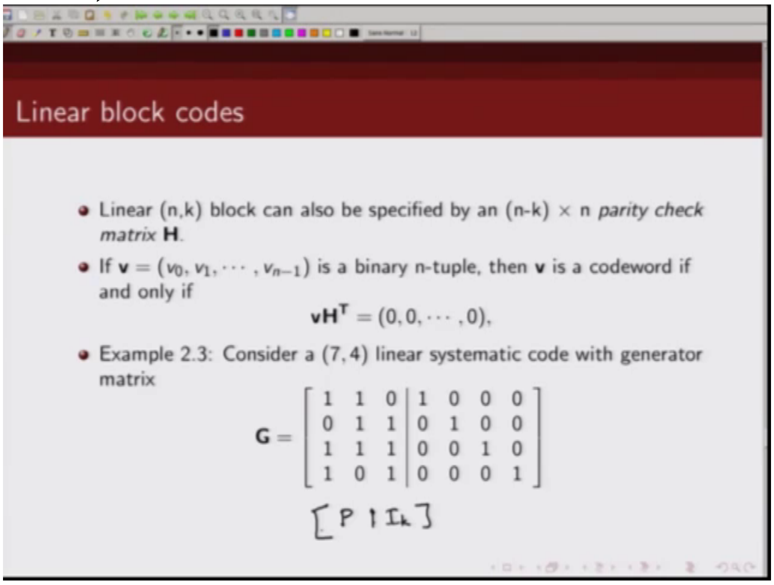
matrix has this property that if v is your valid codeword, if and only if v H transpose is going to be 0. So if v is a valid codeword v H transpose will be 0. So let us see how we can derive our parity check matrix from a

(Refer Slide Time 17:46)



generator matrix and what's a relation of the generation matrix with the parity check matrix. So we will take an example of a 7 4 systematic linear block code whose generator matrix is given by this. So since this is a systematic code we can write it of the form p times this i k. This generator matrix can be written of this form,

(Refer Slide Time 18:18)

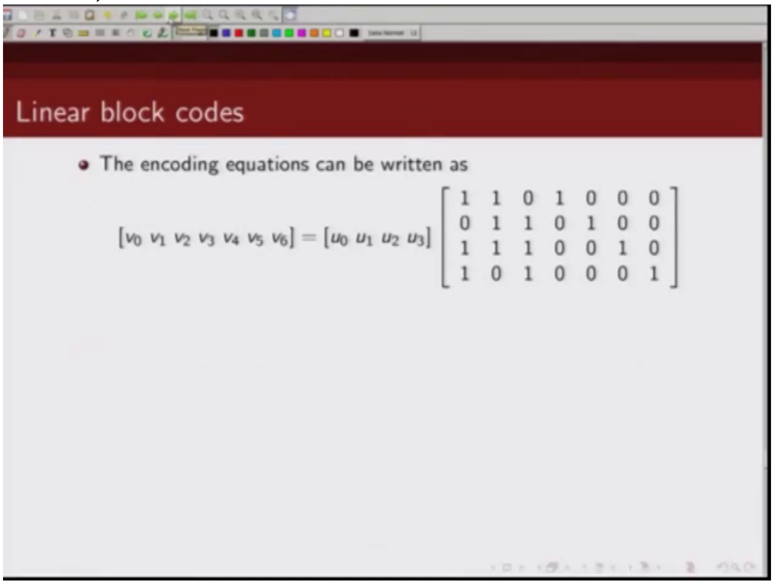


Ok. Now from this generator equation we can write our coded bits

(Refer Slide Time 18:24)

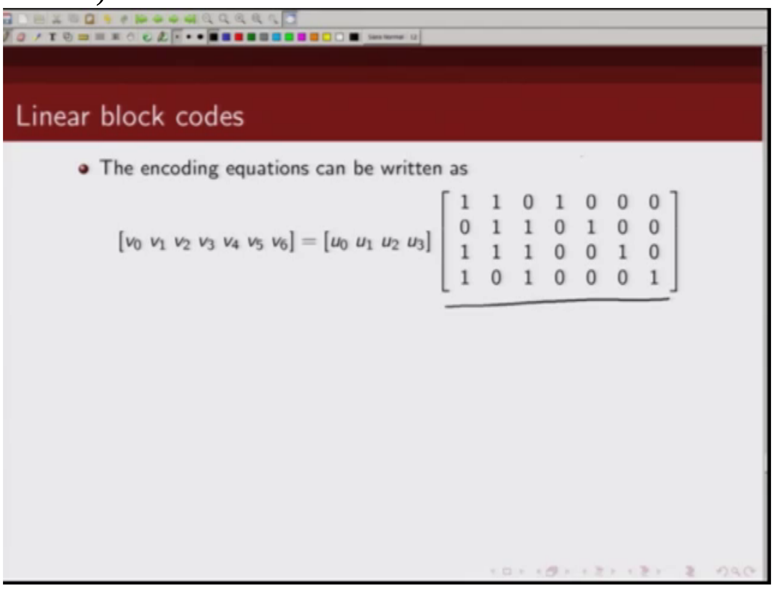
in terms of our message bits So let's do that. So

(Refer Slide Time 18:31)



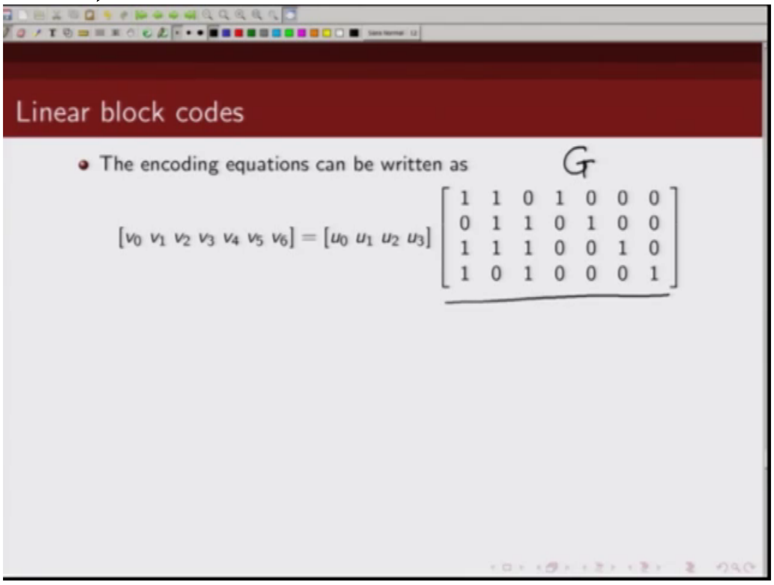
the encoding equation; this was our generator matrix G,

(Refer Slide Time 18:35)



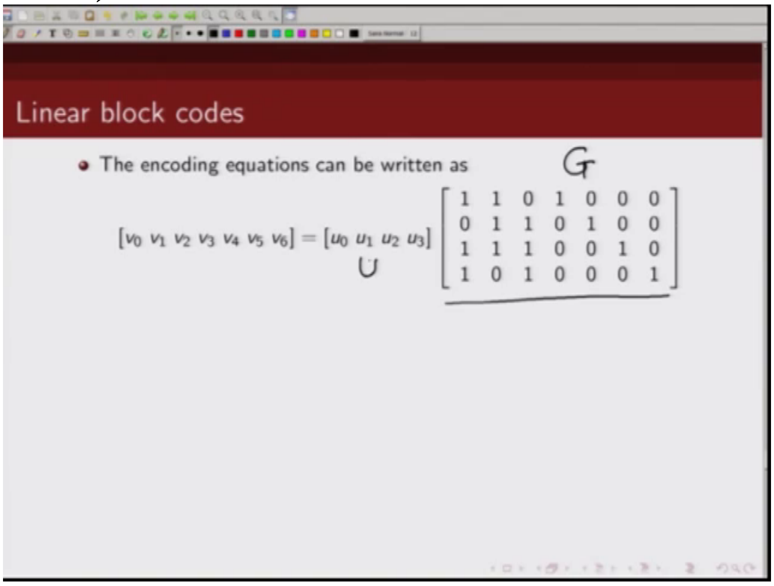
this is

(Refer Slide Time 18:37)



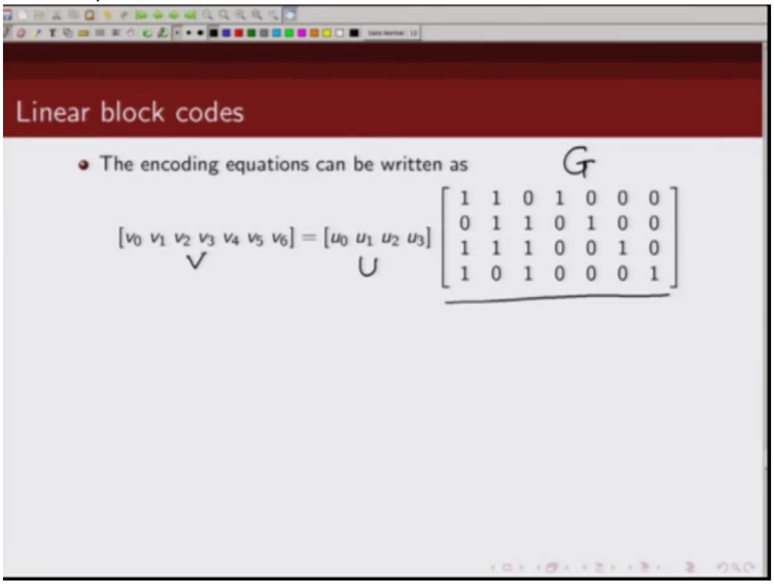
our information bits message

(Refer Slide Time 18:39)



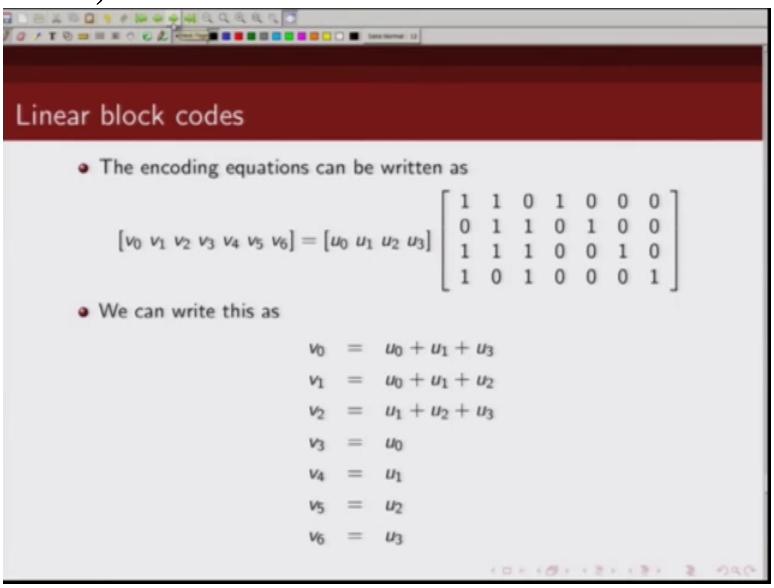
bits u and this is our coded bits v. So we can write v

(Refer Slide Time 18:44)



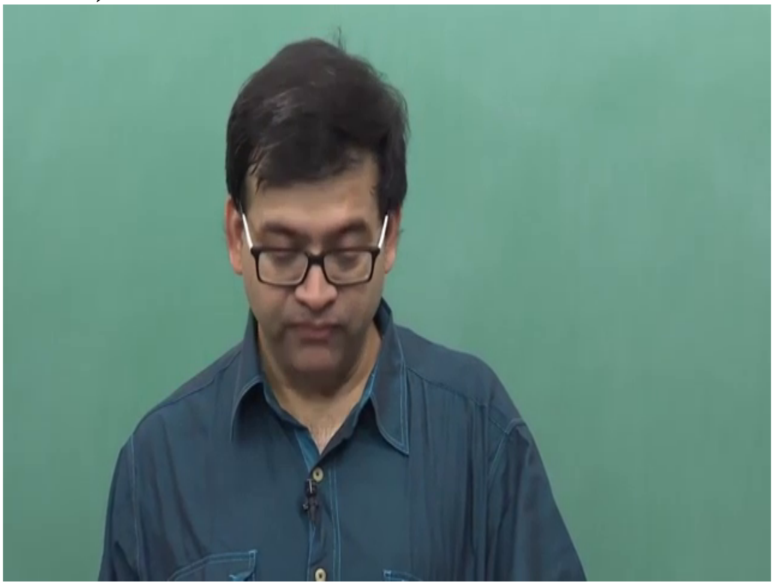
as u times G So it's a 7 4 code. So there are 4 information bits. I denote them by u 0 u 1 u 2 u 3. And there are 7 coded bits. I denote them by v 0 v 1 v 2 to v 6. And this generator matrix I have already given you this. So I can write down the encoding equations. We do that.

(Refer Slide Time 19:10)



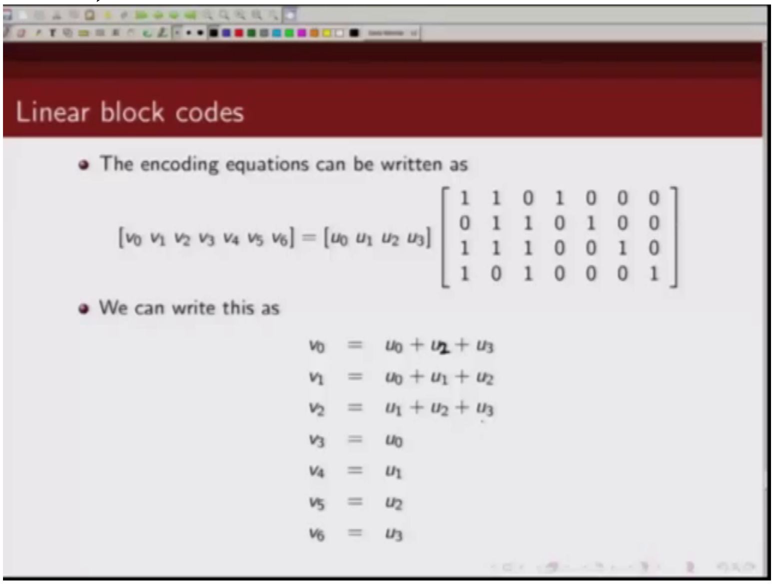
You can see from this what is v 0. It is u 0 plus u 2 plus u 3. There is a typing mistake here. This should have been u 2. So v 0 is u 0 times 1 plus 0 times u 1 plus 1 times u 2 plus 1 times u 3. So this v 0 is given by u 0 plus u 2 plus u 3. Similarly what is v 1? v 1 is given by u 0 plus u 1 plus u 2. v 2 is given by u 1, u 0 into 0, u 1 into 1, u 2 plus u 2 into 1 plus u 3 into 1. So v 2 is given by u 1 plus u 2 plus u 3. So that's what I have here. What is v 3? v 3 is given by u 0 into 1 and rest are all 0s So v 3 is nothing but u 0. Similarly v 4 is u 1, v 5 is u 2 and v 6 is u 3.

(Refer Slide Time 20:37)



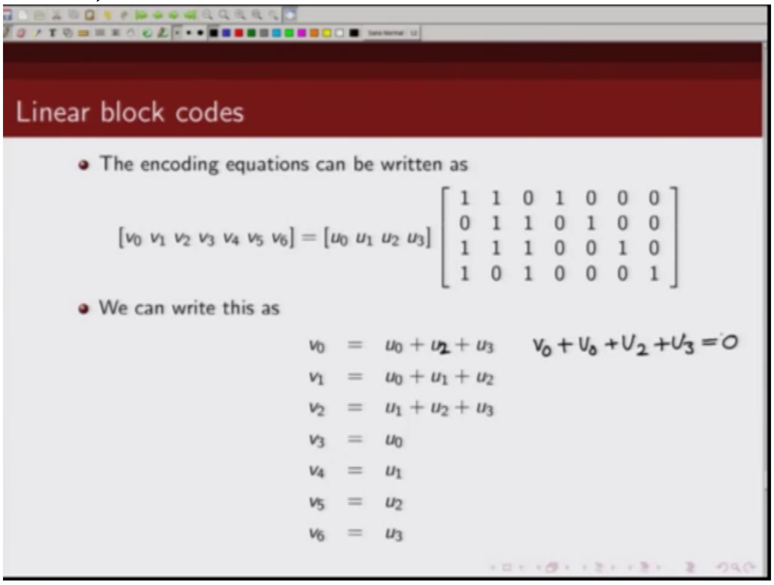
So I now have set of 7 coded bits and this shows the relation between the coded bits and information bits. Now we are, since we are restricting ourselves to binary codes we can

(Refer Slide Time 20:52)



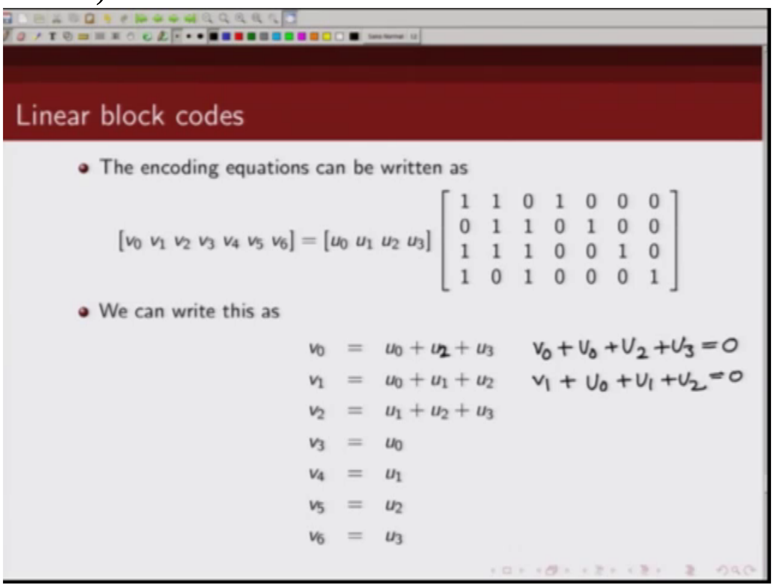
even write this equation like this, v 0 plus u 0 plus u 2 plus u 3 is equal to 0, correct?

(Refer Slide Time 21:09)



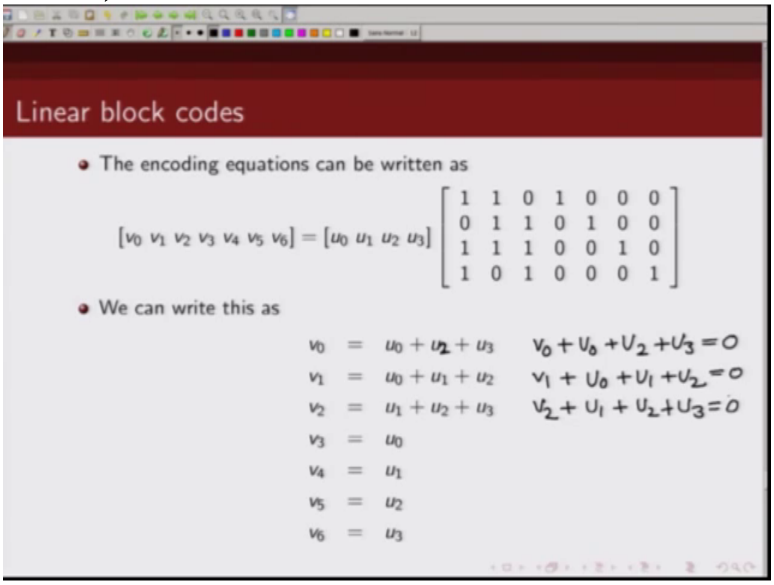
Because this v 0 v 1 is nothing but parity bit which is basically nothing but like 1 or 0. So we add this to this. Modulo 2 sum will be 0. So this similarly we can write as v 1 plus u 0 plus u 1 plus u 2 equal to 0

(Refer Slide Time 21:32)



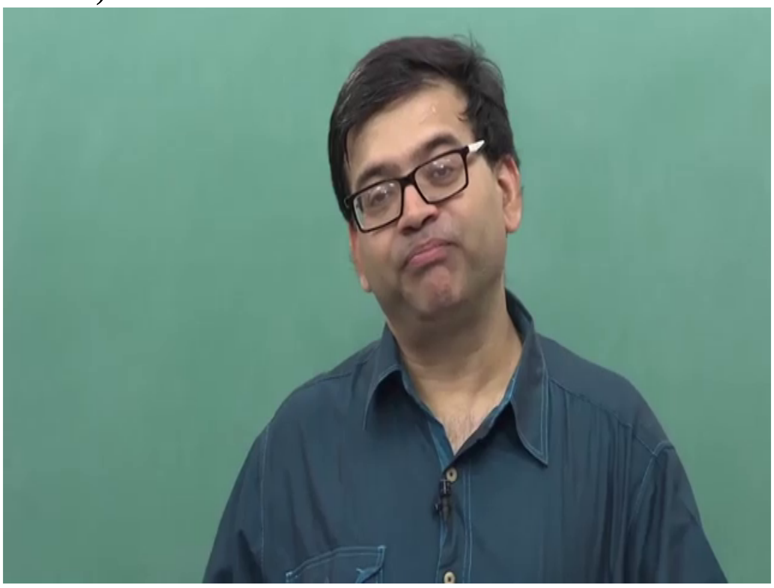
and this can be written as v 2 plus u 1 plus u 2 plus u 3 is equal to 0.

(Refer Slide Time 21:44)



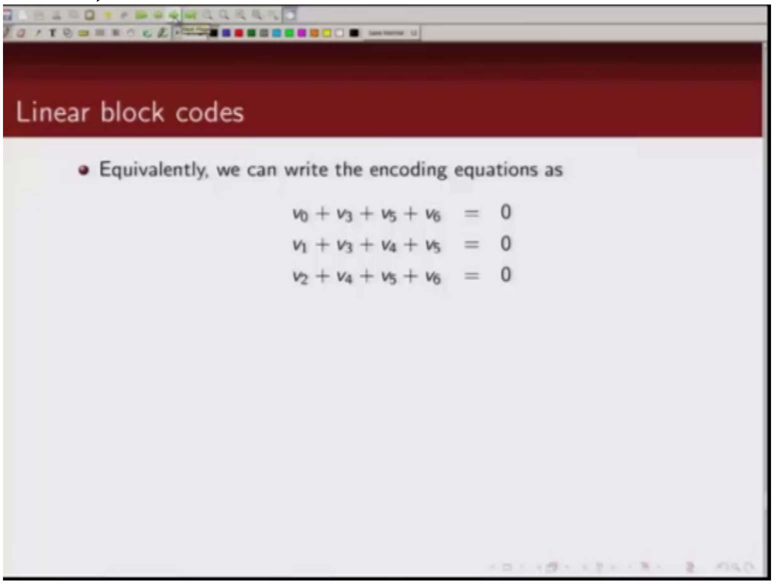
The next, what we would try to do is we would try to write these parity check equations in terms of other coded bits. So we can see here, u 0 is nothing but v 3. So wherever u 0 appears, we can replace it by v 3. Similarly u 1 is equal to v 4. So wherever u 1 appears we can replace it by v 4. u 2 is equal to v 5, so we can replace u 2 by v 5. And u 3 is equal to v 6. We can replace u 3 in terms of v 6. By doing this, what we will get is set of

(Refer Slide Time 22:31)



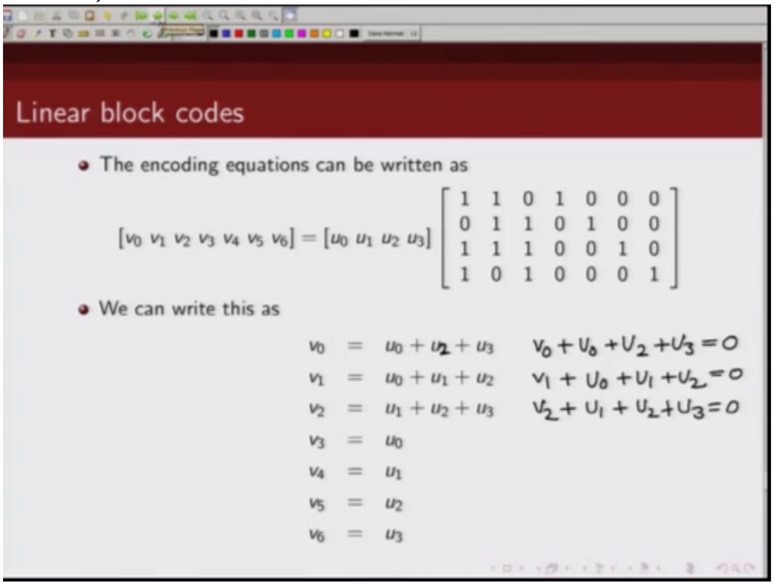
equations which basically are dependent on these coded bits. If we do that, what we get

(Refer Slide Time 22:40)



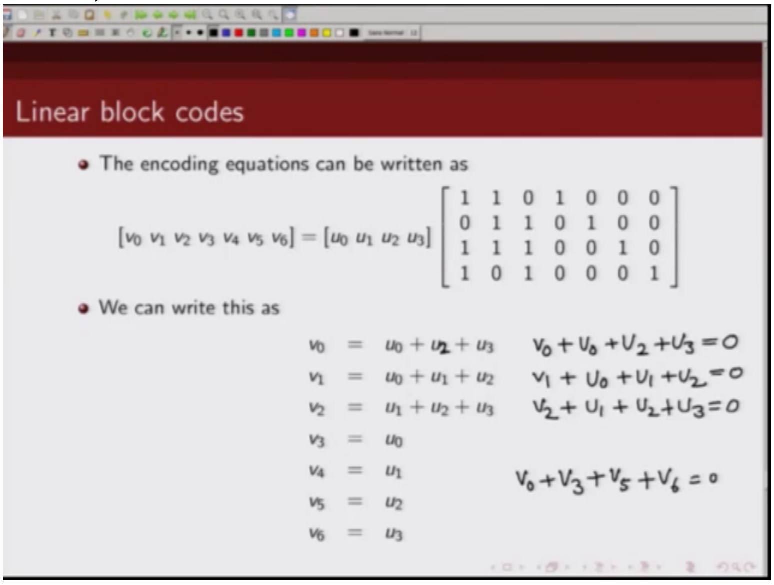
is something like this The first expression basically which was,

(Refer Slide Time 22:47)



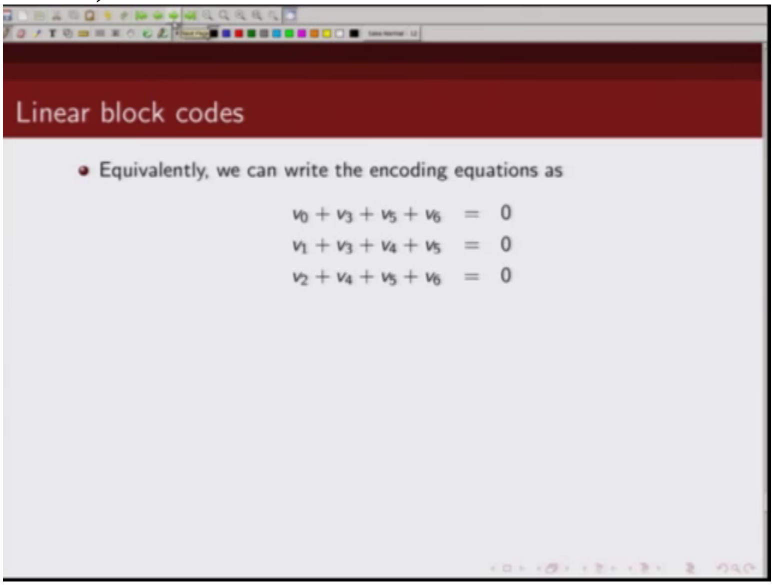
v 0 plus u 0 the u 2 plus u 3 Now this can be re-written as v 0 plus what is u 0, u 0 is v 3, v 3 plus what is u 2, u 2 is v 5, v 5 plus what is u 3, it is v 6. So

(Refer Slide Time 23:11)



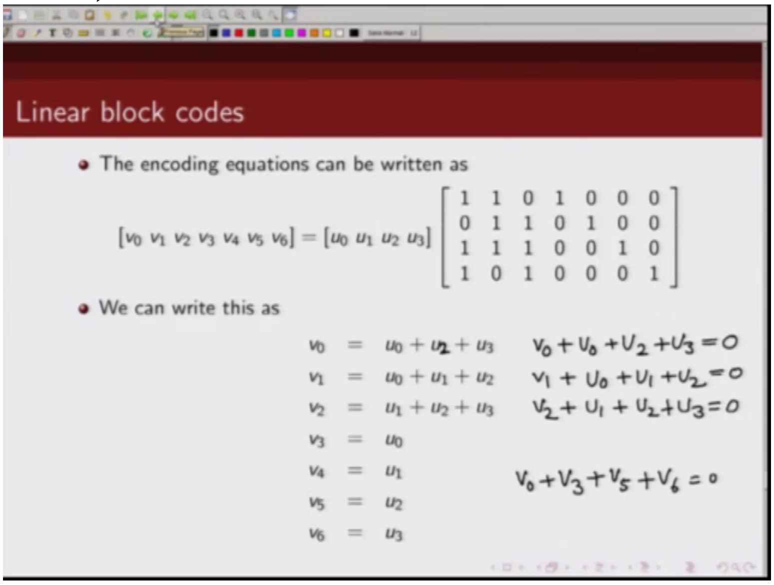
v 0 plus v 3 plus v 5 plus v 6 is 0, and that is what we have here.

(Refer Slide Time 23:18)

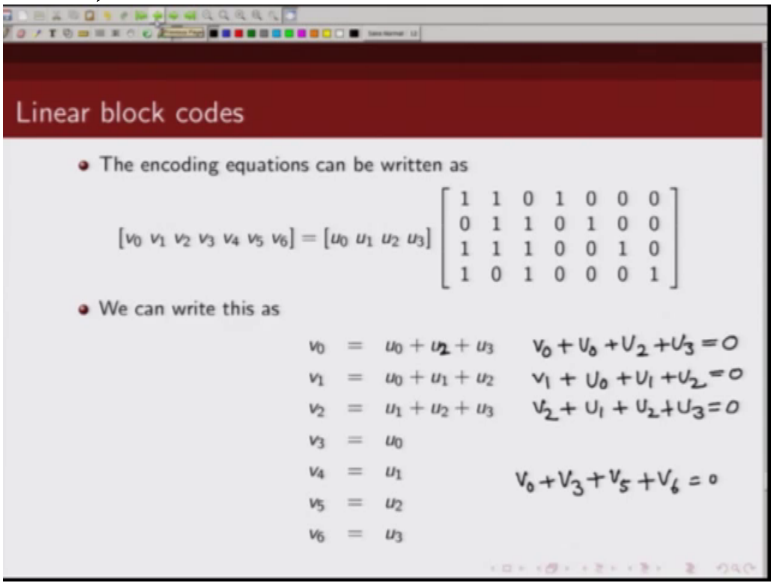


v 0 plus v 3 plus v 5 plus v 6 is equal to 0. Similarly we can write

(Refer Slide Time 23:27)

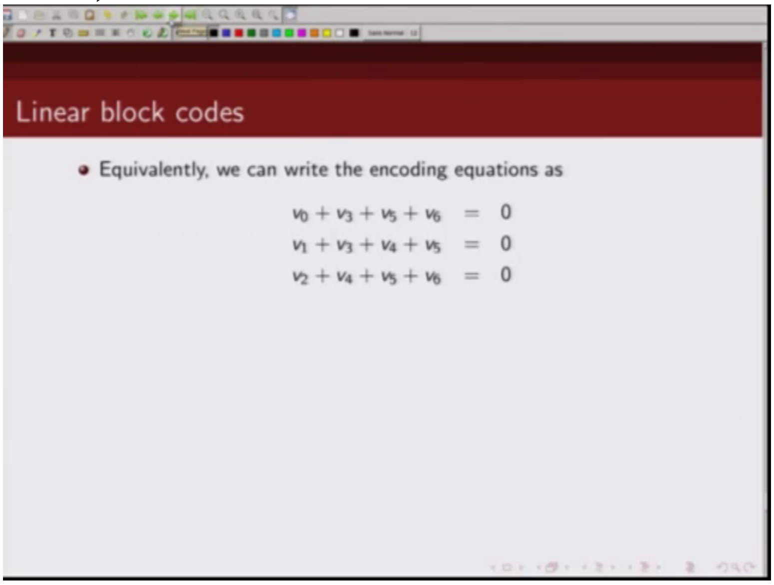


(Refer Slide Time 23:28)



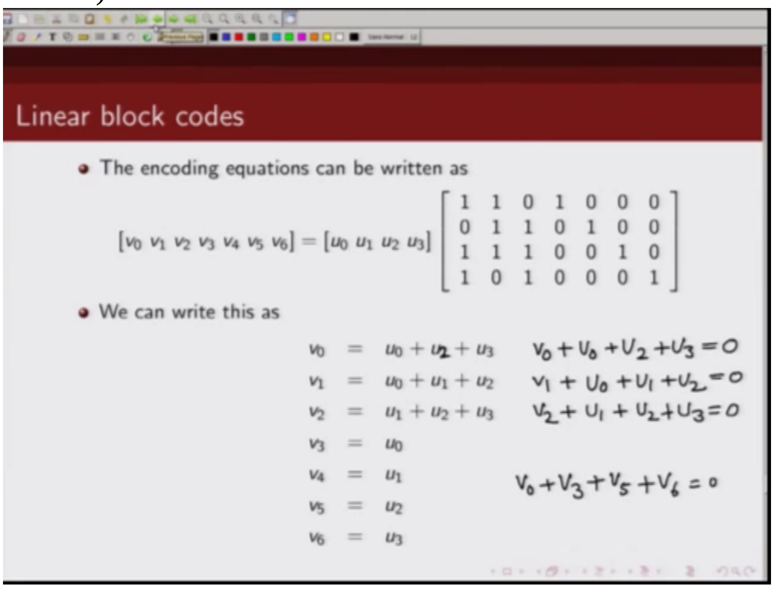
the other equations as well Here also we will replace u 0, u 1, u 2 by v 3, v 4, v 5 and what we will get

(Refer Slide Time 23:39)



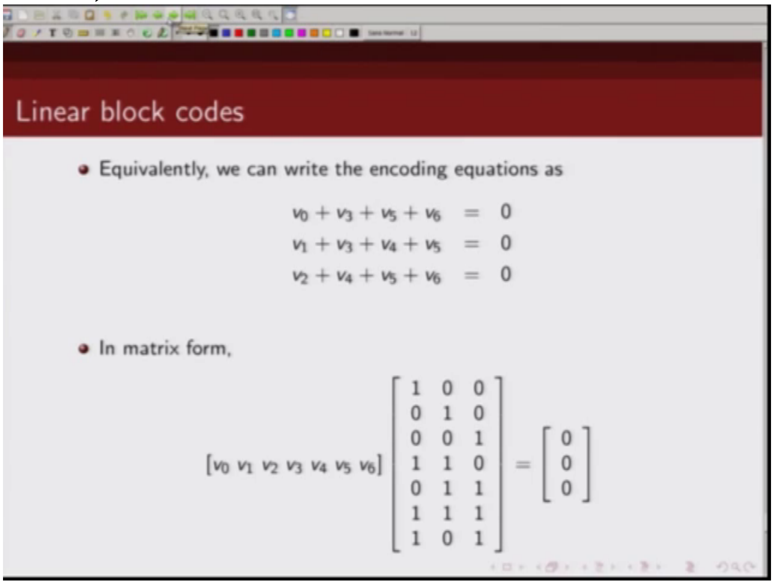
is v 1 plus v 3 plus v 4 plus v 5 is equal to 0 and similarly the last parity check equation can be written as v 2 plus, v 2 plus u 1 is

(Refer Slide Time 23:52)



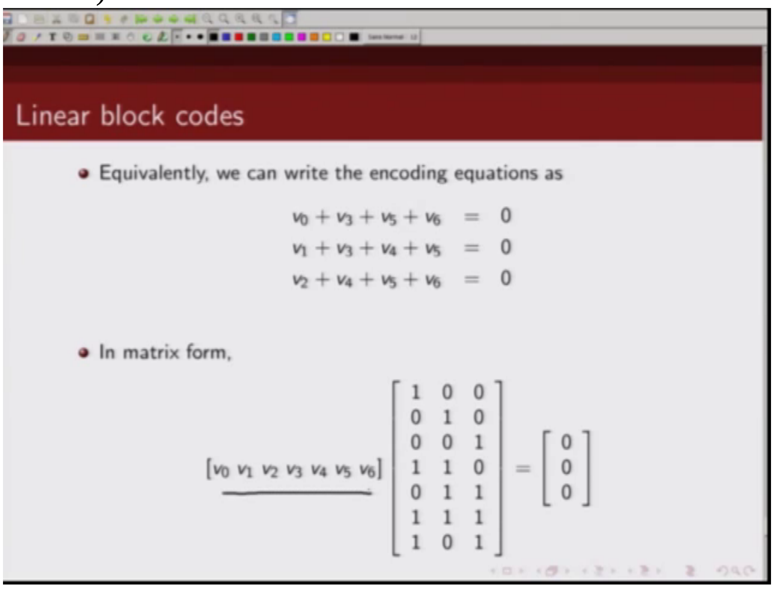
v 4, u 2 is v 5 plus u 3 is v 6. So that's what we have here. u 2, v 2 plus v 4 plus v 5 plus v 6 is equal to 0. So now we have set of encode equations in terms of coded bits. Next,

(Refer Slide Time 24:11)



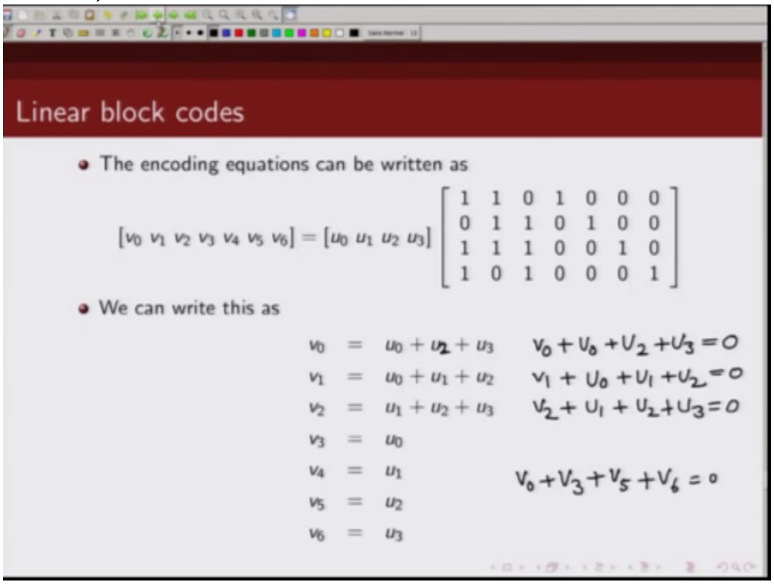
the same thing we can write it in a matrix form. So I have my

(Refer Slide Time 24:18)

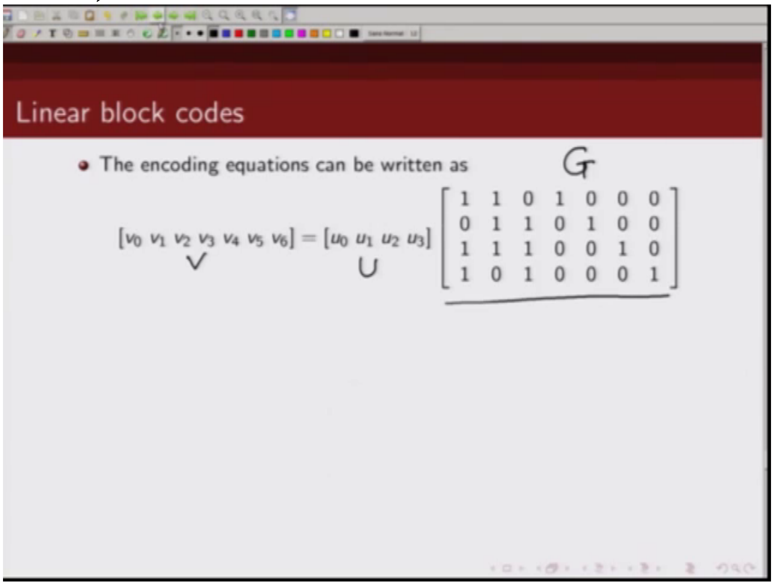


coded bits, v 0 to v 6 I have 3 sets of parity check equations, this, this and this. And the same thing I can write it in a matrix form like this. Now you can see these are equivalent. So look, let's look at first equation. This is v 0 plus v 3 plus v 5 plus v 6 is 0. You can see which are the elements which are so v 0 times 1; this is v 3 times 1 plus v 5 times 1 plus v 6 times 1. So that's what is defined in this equation. Similarly we can see this equation. This v 1 plus v 3 plus v 4 plus v 5 is equal to 0 and this last equation, this is v 2 plus v 4 plus v 5 plus v 6 is 0. And what we did we say about parity check matrix?

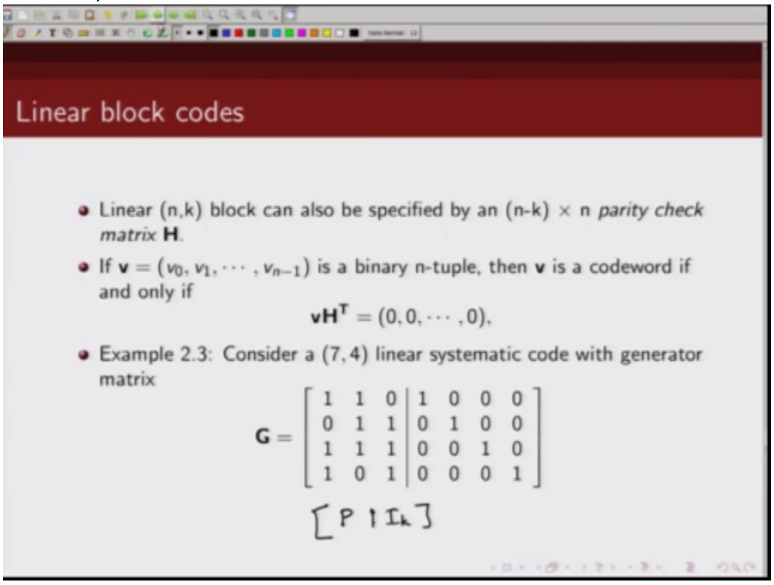
(Refer Slide Time 25:19)



(Refer Slide Time 25:20)

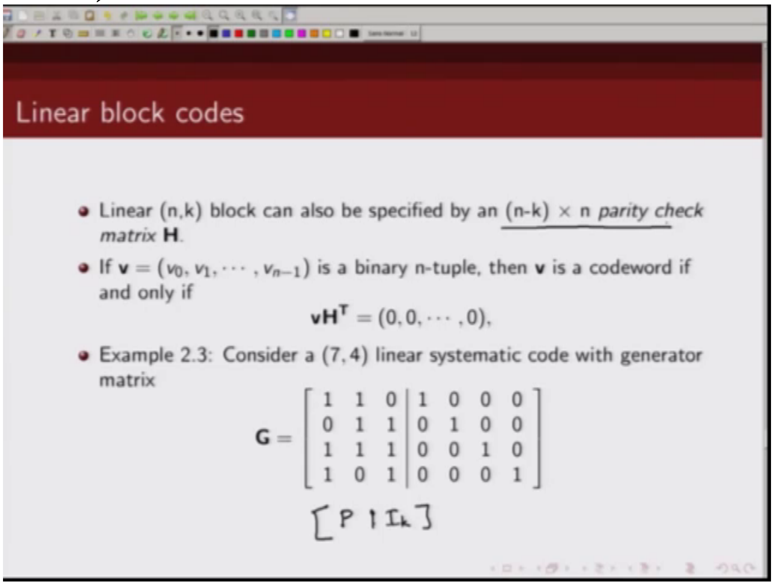


(Refer Slide Time 25:21)



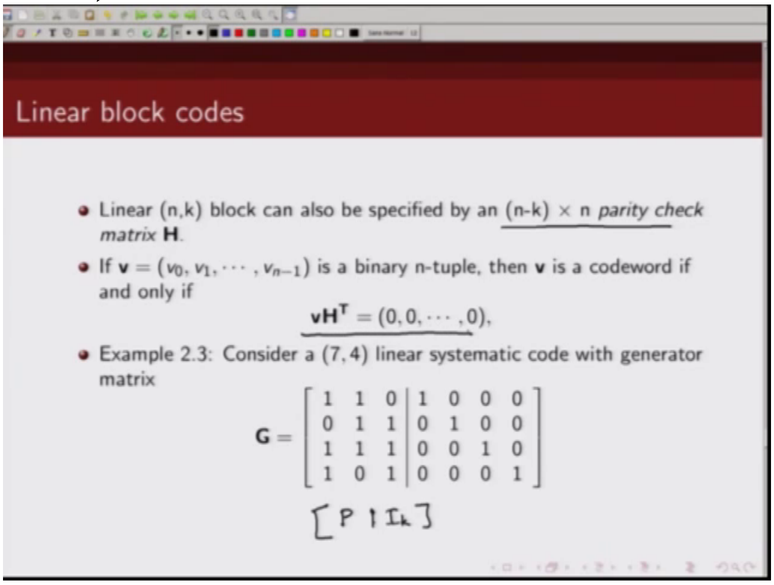
We said that if H is a parity check matrix it is

(Refer Slide Time 25:28)



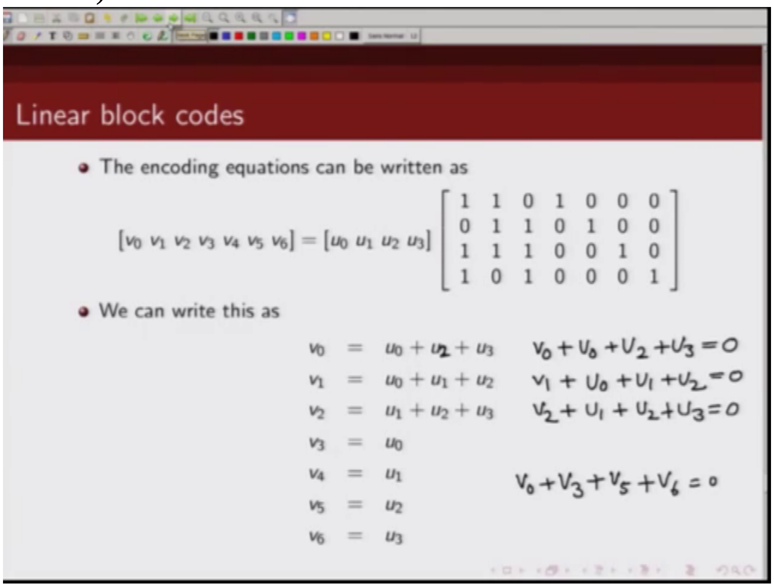
n minus k cross matrix and it has this property that v H transpose is

(Refer Slide Time 25:34)



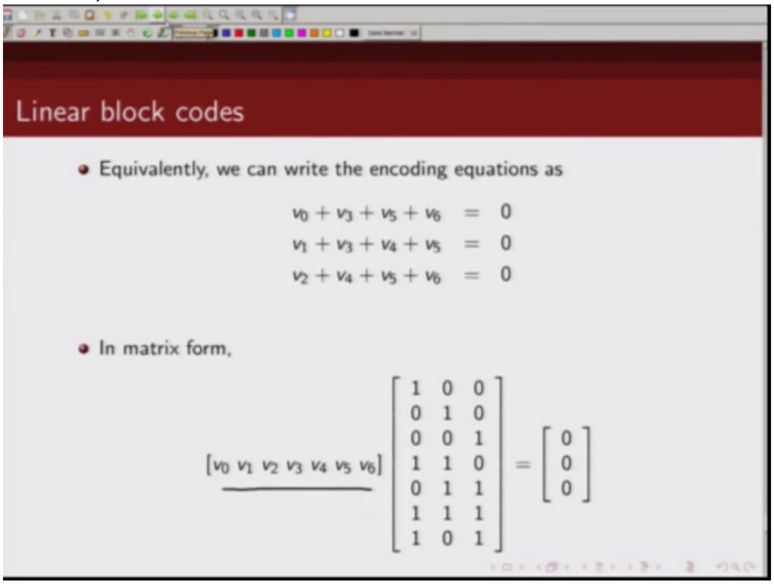
0. So we have,

(Refer Slide Time 25:37)



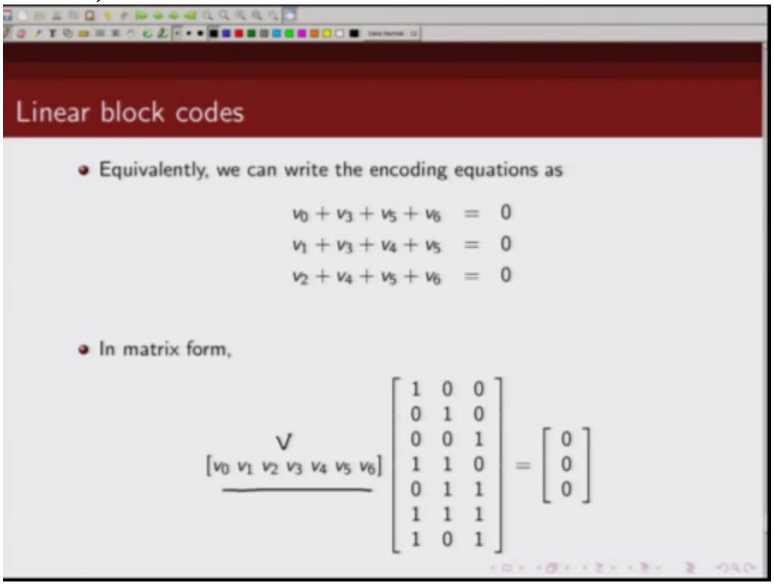
we can write this as,

(Refer Slide Time 25:39)



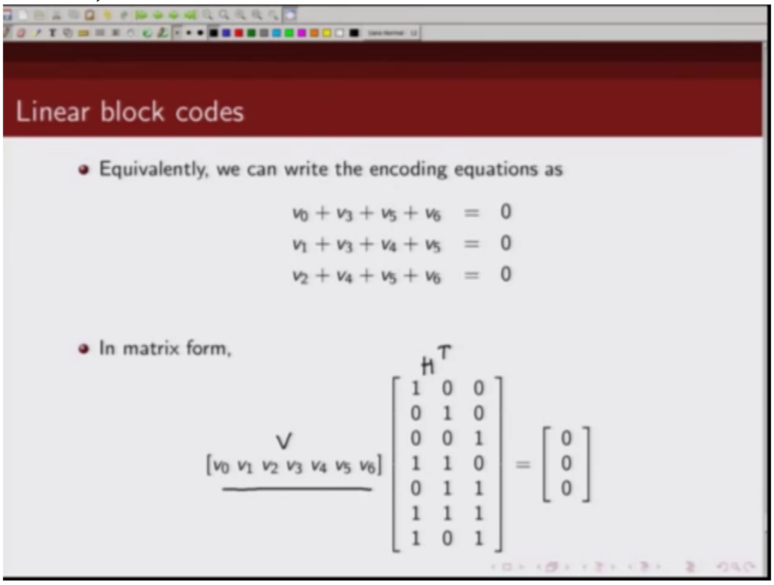
this is my v,

(Refer Slide Time 25:41)



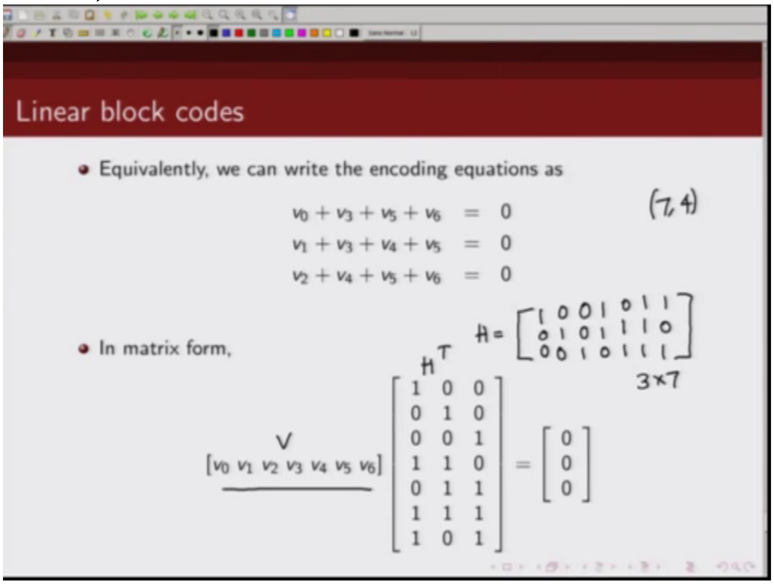
this is my H transpose. v H transpose is

(Refer Slide Time 25:45)



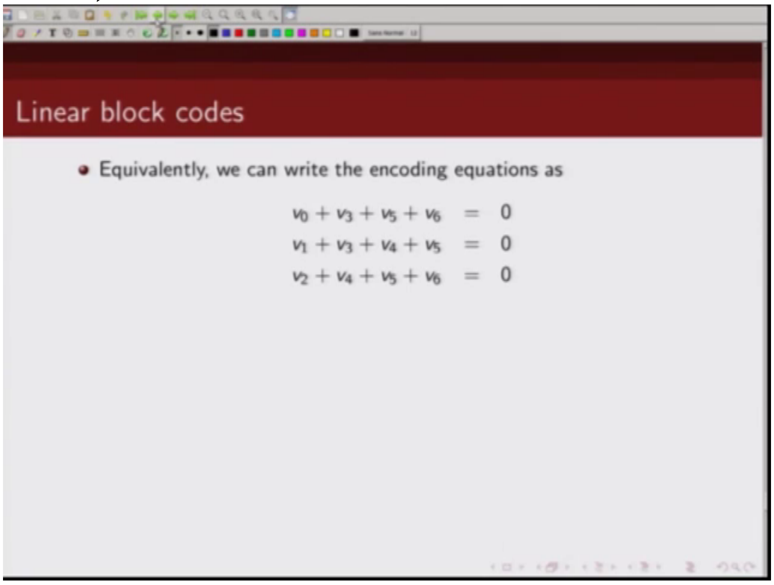
0, so then what is my H matrix? H matrix is a transpose of this matrix, so this will be 1 0 0, 0 1 0, 0 0 1, 1 1 0, 0 1 1, 1 1 1 and 1 0 1. This is my, so for the 7 4 code, 7 4 code this is basically 3 cross 7. As I said, n minus k cross n matrix,

(Refer Slide Time 26:20)

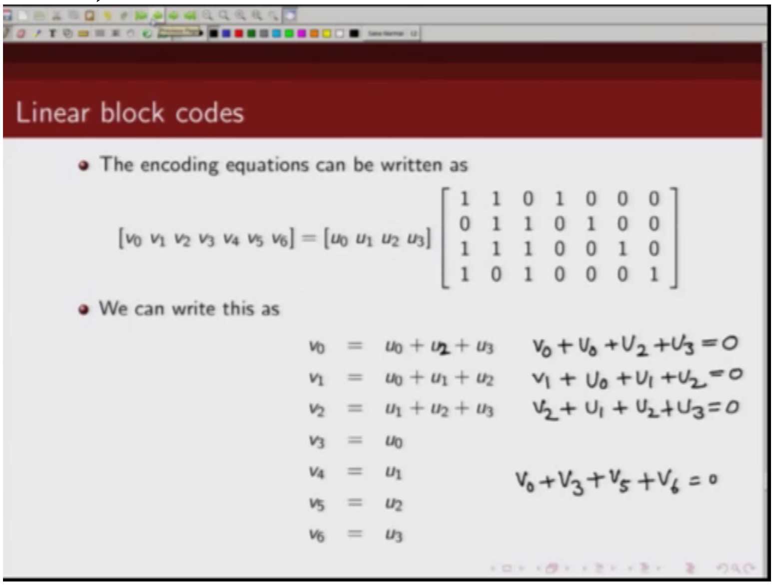


this is my parity check matrix corresponding to this same code which is generated by

(Refer Slide Time 26:28)

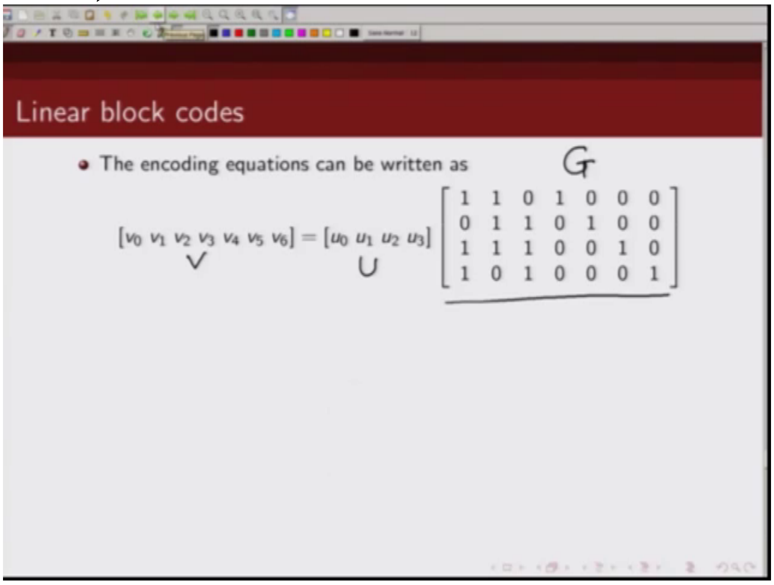


(Refer Slide Time 26:29)



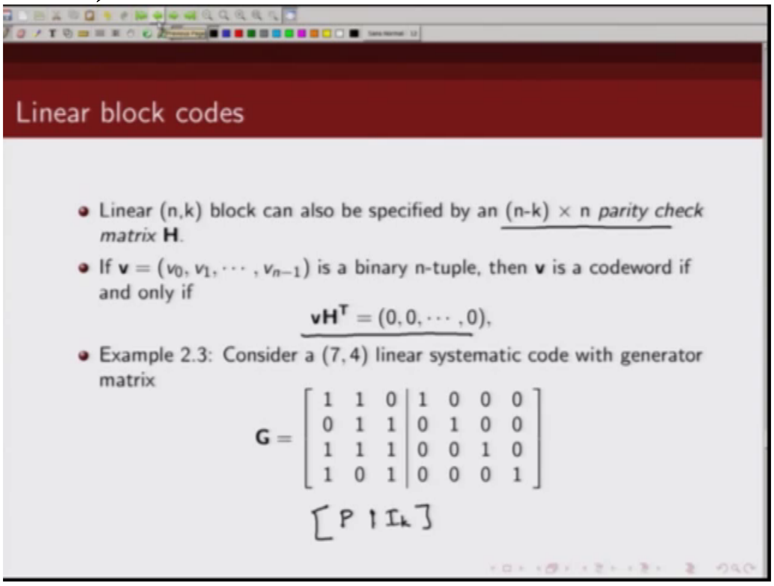
this

(Refer Slide Time 26:32)



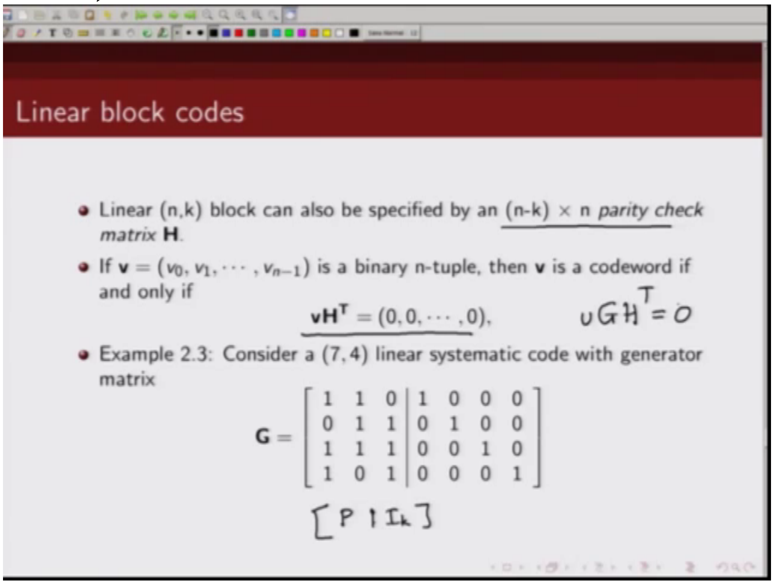
Another interesting

(Refer Slide Time 26:34)



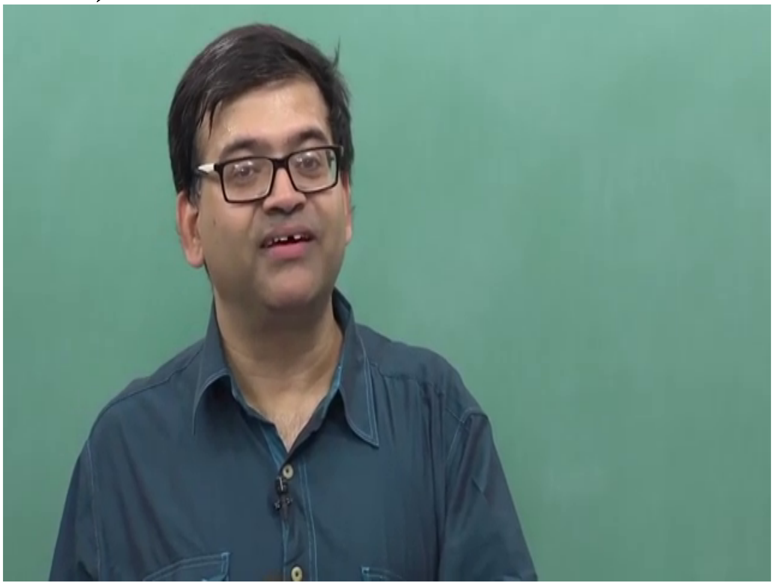
property which you can generally see is, so v H transpose is 0; I can write this u times v H transpose is equal to 0. In other words,

(Refer Slide Time 26:45)



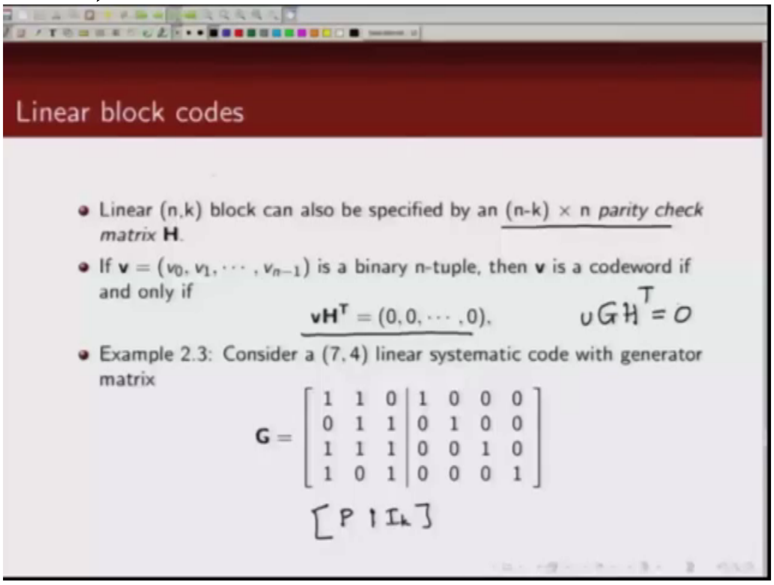
v H transpose is

(Refer Slide Time 26:47)



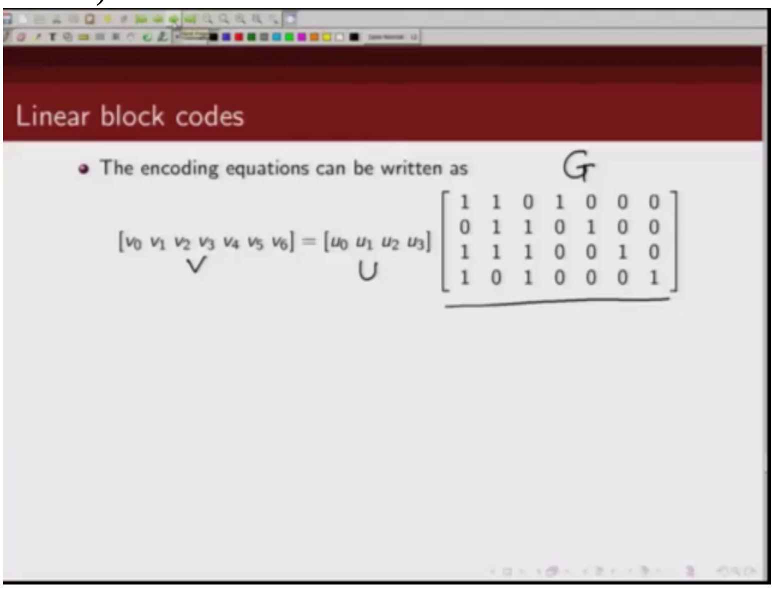
0, so what does that mean? The rows of G matrix and rows of H matrix are orthogonal to each other.

(Refer Slide Time 26:58)



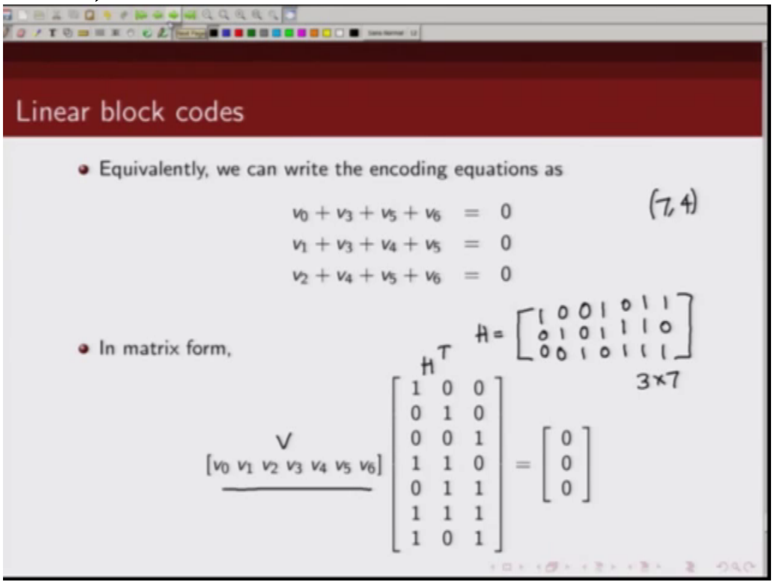
So the H lies in

(Refer Slide Time 27:00)



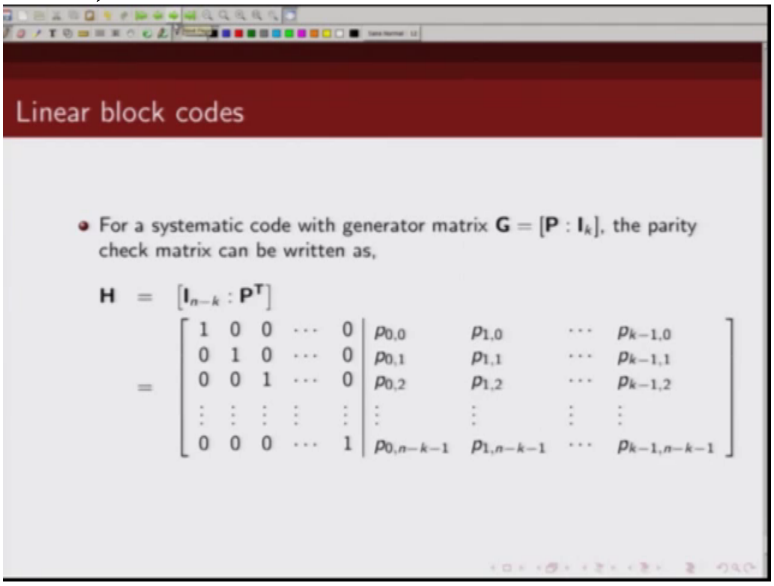
the null space of G, so as we can see from this that generator matrix and parity check matrix are related to each other.

(Refer Slide Time 27:11)



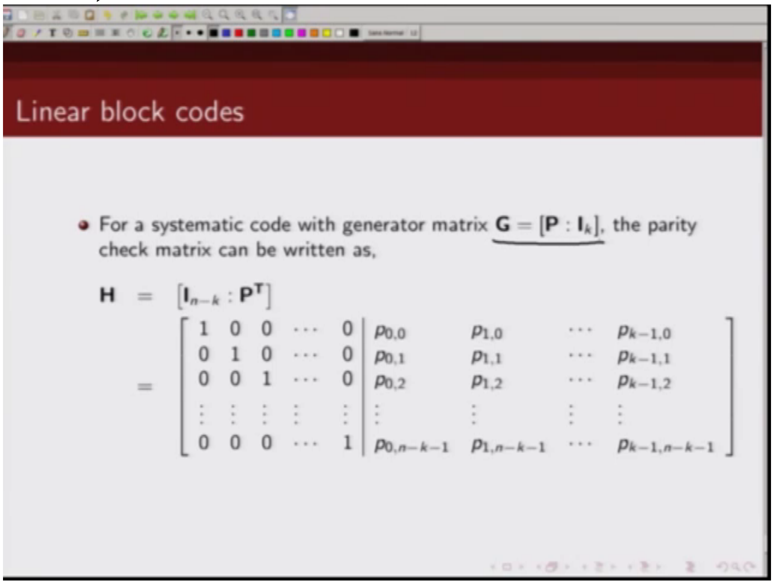
And they have this property that rows of G matrix and H matrix are basically orthogonal to each other. So if you have

(Refer Slide Time 27:23)



a systematic code whose generator matrix can be written

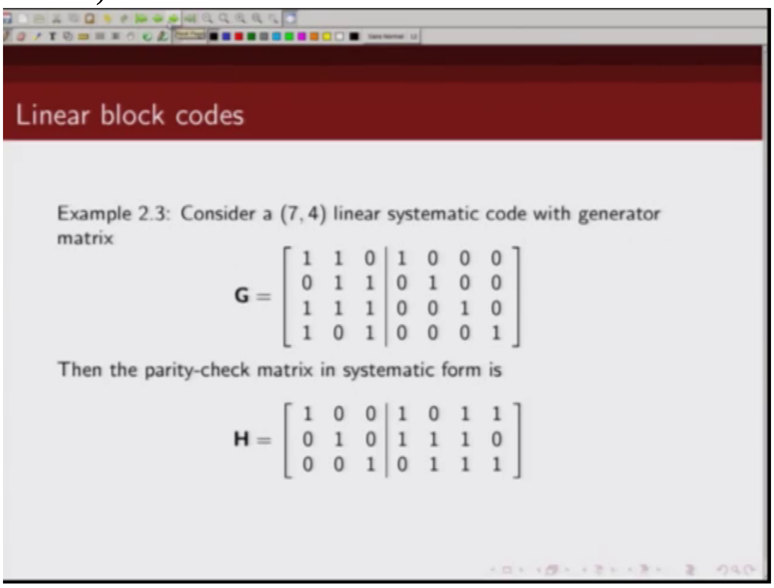
(Refer Slide Time 27:29)



in this form, because H lies in the null space of G, we can write down its corresponding H matrix very easily. And this is basically given by, so if a generator matrix can be written of the form P and identity matrix, we can write its parity check matrix as identity matrix and P transpose.

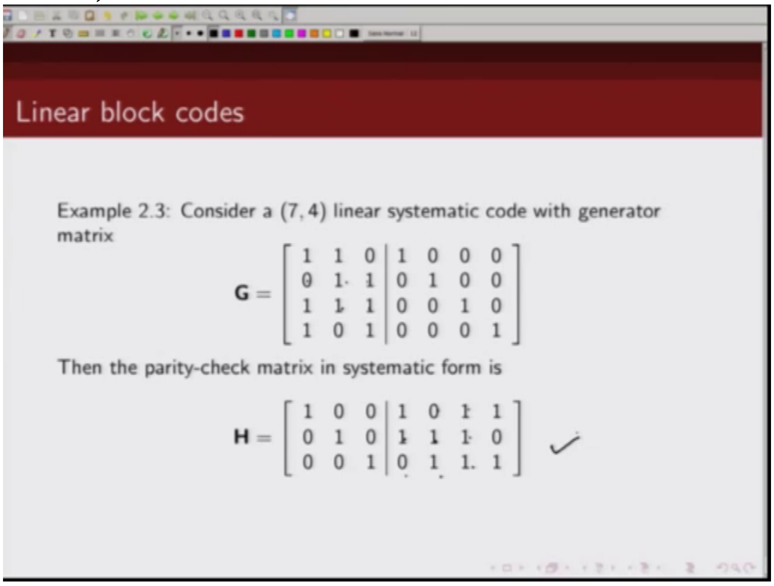
So let's take an example

(Refer Slide Time 27:55)



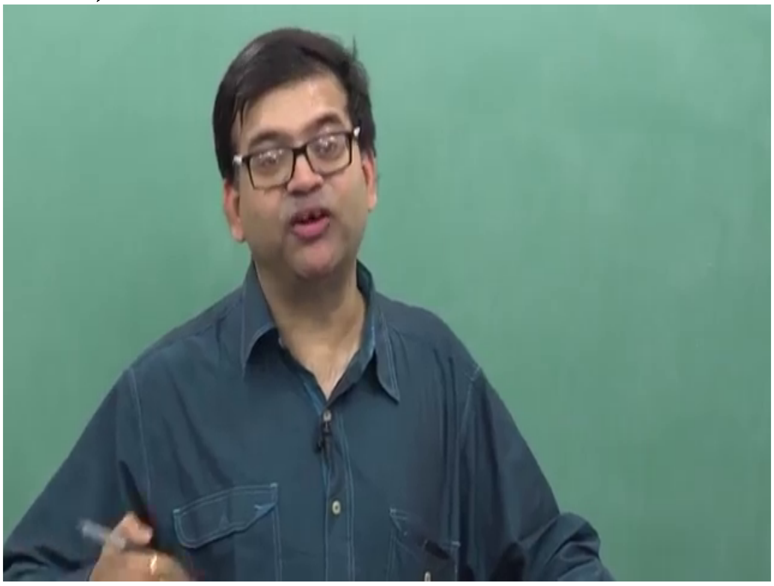
of a generator matrix of a systematic code This is systematic code we can see, we can separate out this generator matrix as some matrix P and some identity matrix. So this we can write as, H matrix we can write as identity matrix and P transpose. So then this can be written as 1 1 0 is 1 1 0; 0 1 1, 0 1 1; 1 1 1, 1 1 1; 0 1 0 so this is my H matrix

(Refer Slide Time 28:25)



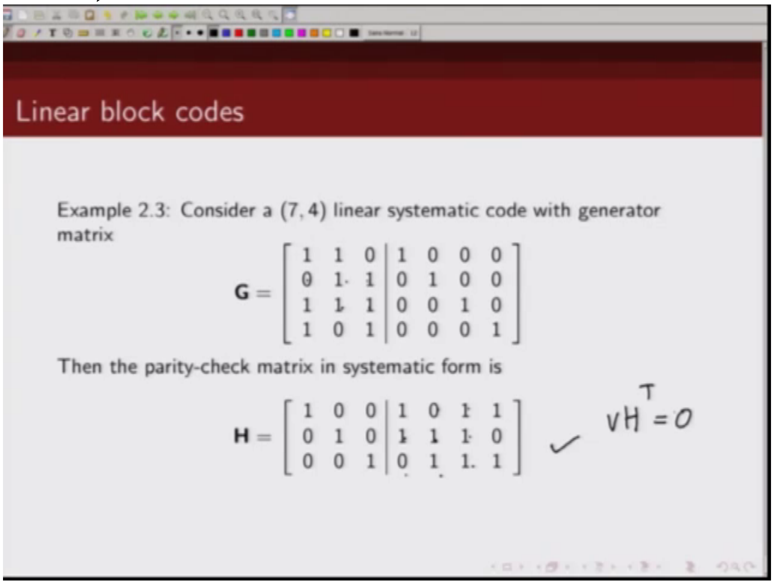
corresponding to this So whether you are given a generator matrix or a parity check matrix, your linear block code is completely specified by either of them. And as I said, we use the generator matrix to generate our code,

(Refer Slide Time 28:42)



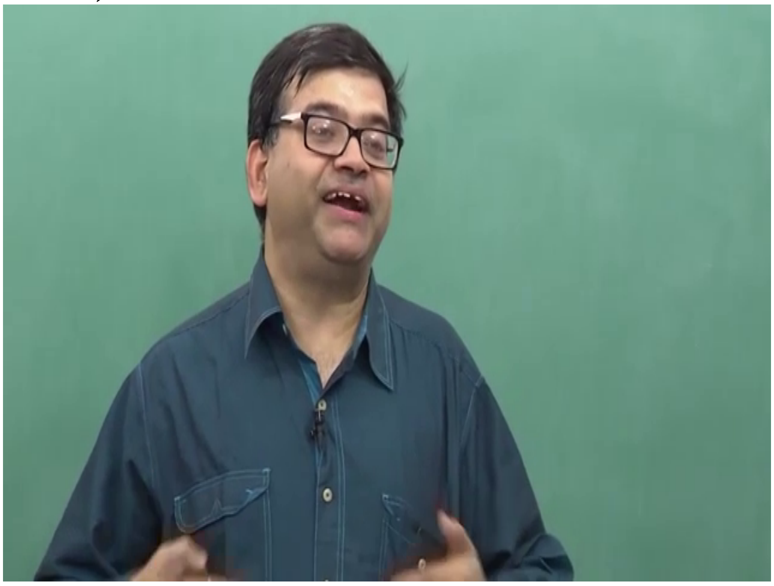
set of codewords where as parity check matrix as the name suggests is used to check whether the parity check constraints are satisfied. As we said basically parity check matrix has this property that, if v is

(Refer Slide Time 28:58)



a valid codeword, if and only if v H transpose is zero and we use this property in decoding, so that's why you see the name parity check

(Refer Slide Time 29:08)



matrix because this matrix H is essentially used to, in some sense check whether the parity check constraints of the code are satisfied or not Thank you.