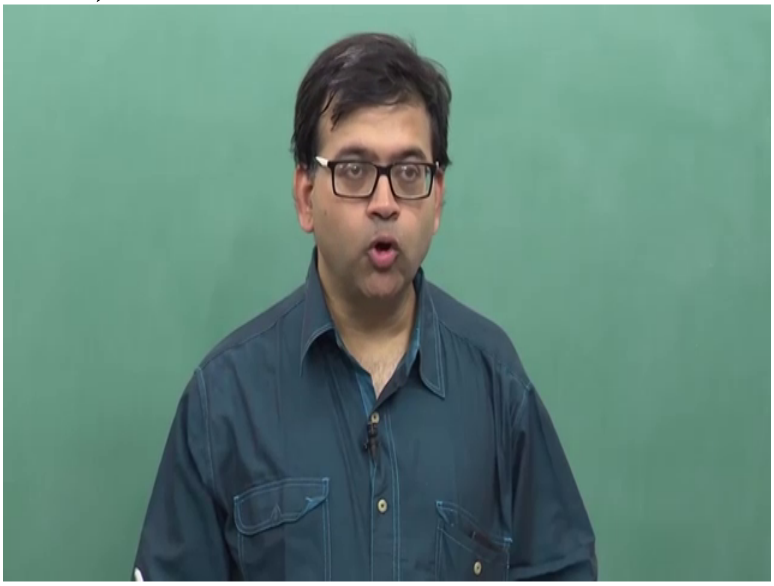
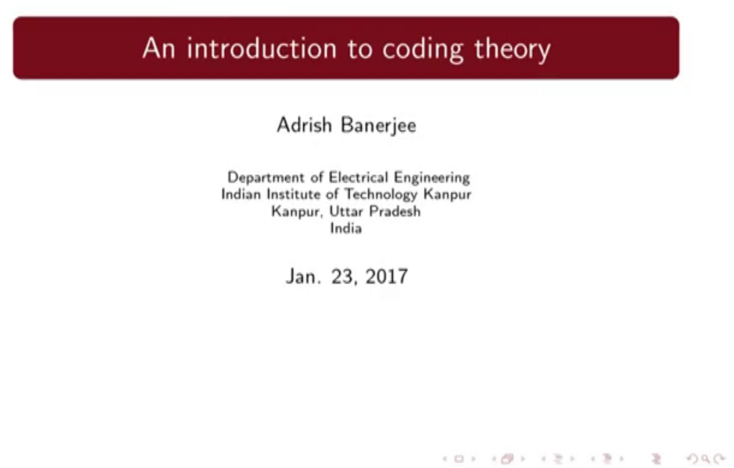
An Introduction to Coding Theory
Professor Adrish Banerji
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Module 01
Lecture Number 05
Syndrome, Error Correction And Error Detection

(Refer Slide Time 00:13)



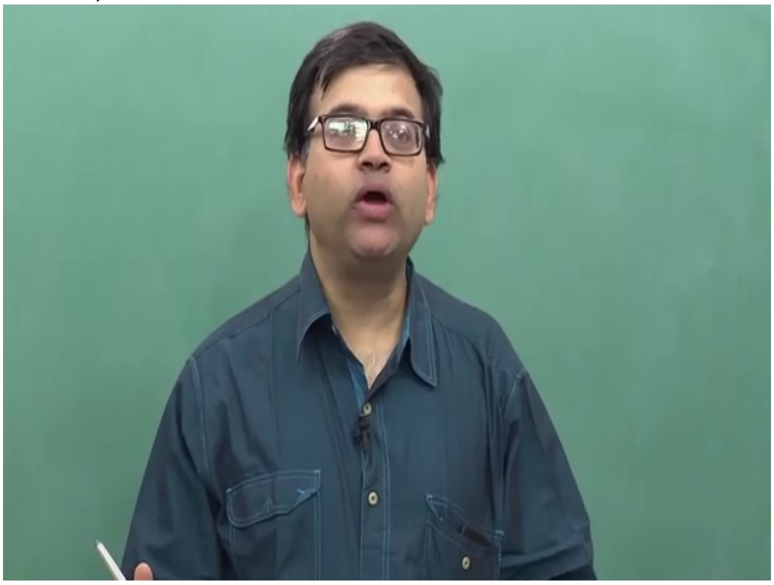
Welcome to the course on

(Refer Slide Time 00:15)



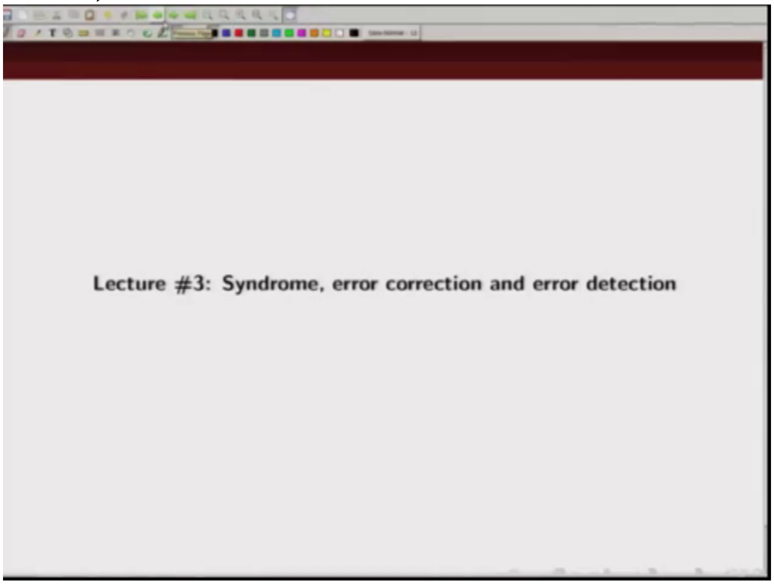
Coding Theory In this lecture, we are

(Refer Slide Time 00:18)



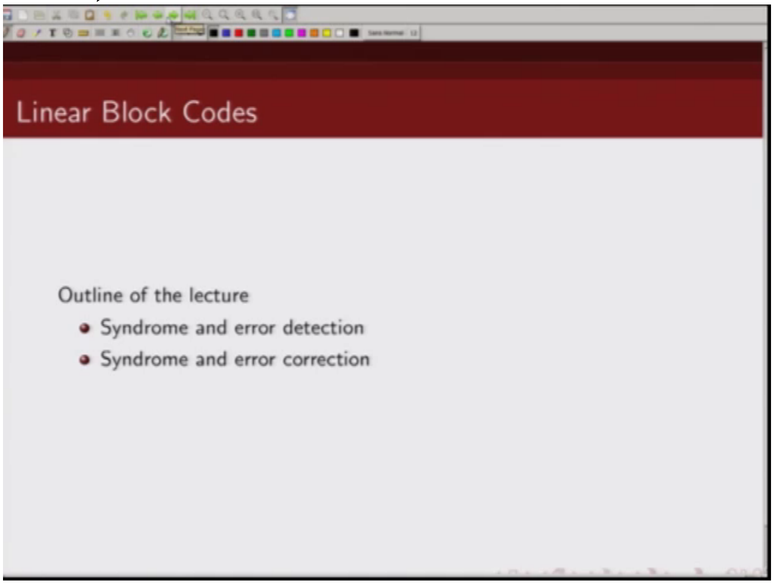
going to describe how we can use error correcting codes for error detection and error correction. So we will first describe what we mean by Syndrome and then we will show how we will use this Syndrome to do error correction and error detection. So this lecture

(Refer Slide Time 00:43)



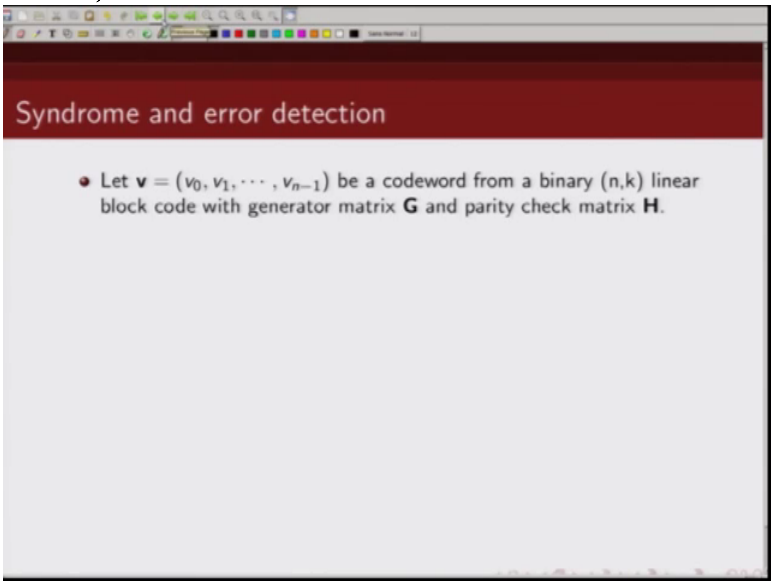
is about Syndrome and error correction and error detection We will first talk about what is a Syndrome and how we can use it for error detection. And then we will talk

(Refer Slide Time 00:53)



about how the Syndrome can be used for error correction

(Refer Slide Time 00:58)

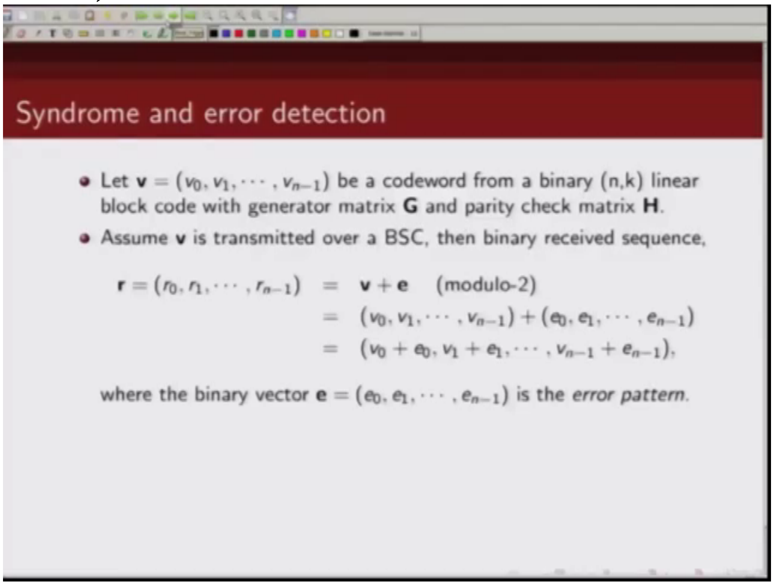


So we know so far that if we have an n k linear block codes, so that means we have k information

(Refer Slide Time 01:07)

bits and n coded bits and this n k linear block code is completely described by a generator matrix and a parity check matrix.

(Refer Slide Time 01:22)

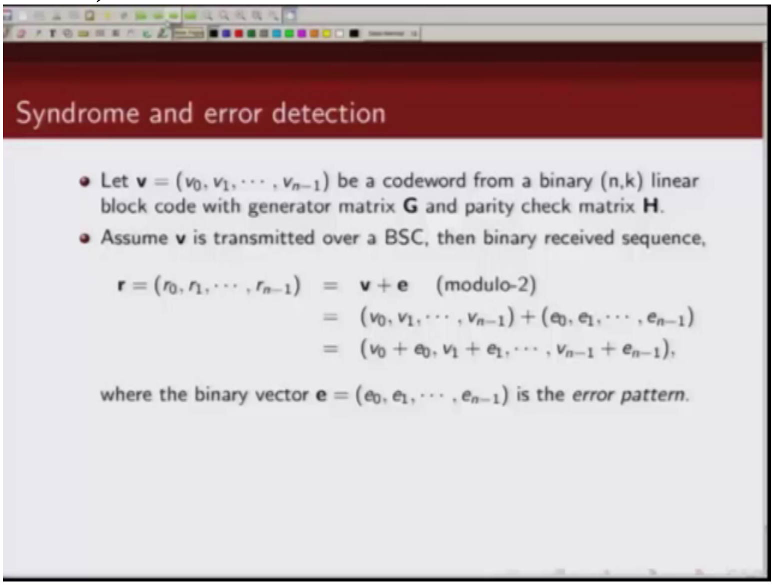


So let us assume that we have codeword which is encoded using a linear n k block encoder. So we have an output of a encoder which is of coded bits of length n. Now we want to transmit this

(Refer Slide Time 01:47)

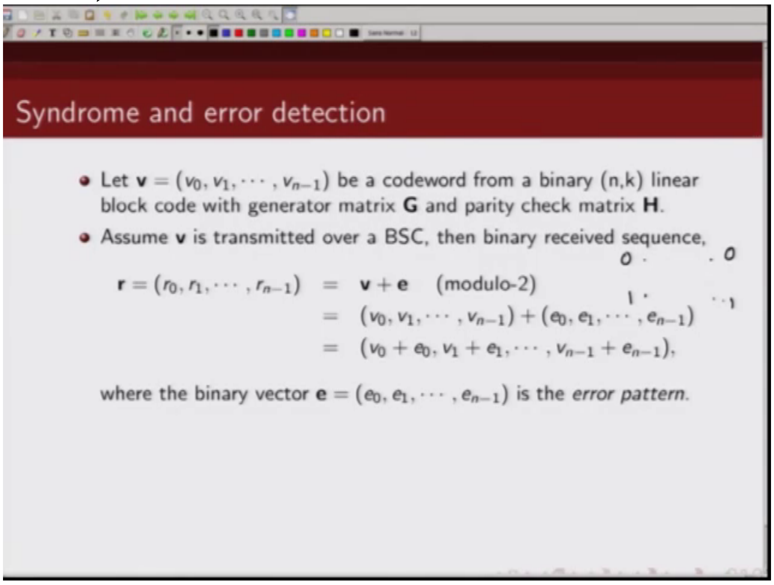
codeword over a communication channel For simplicity let's consider a

(Refer Slide Time 01:52)



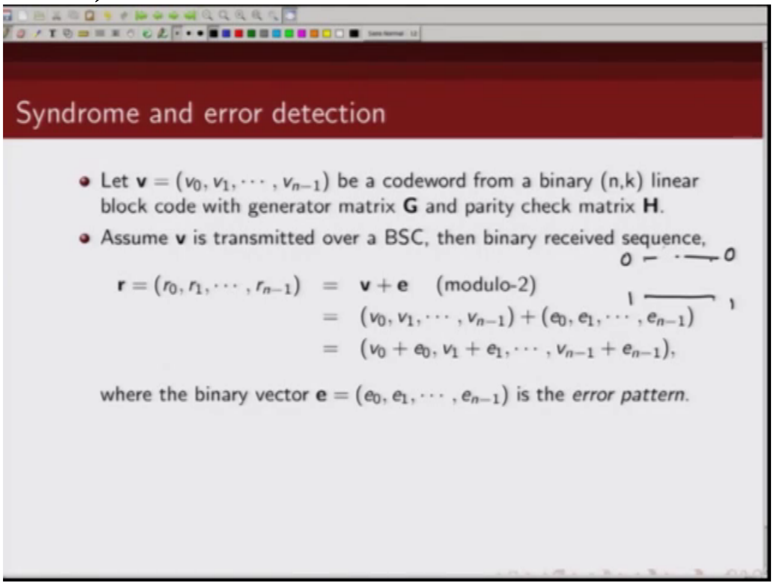
binary symmetric channel Again let's recall what is a binary symmetric channel. So in a binary symmetric channel we have 2 inputs and 2 outputs, so binary inputs 0s and 1, binary output

(Refer Slide Time 02:06)



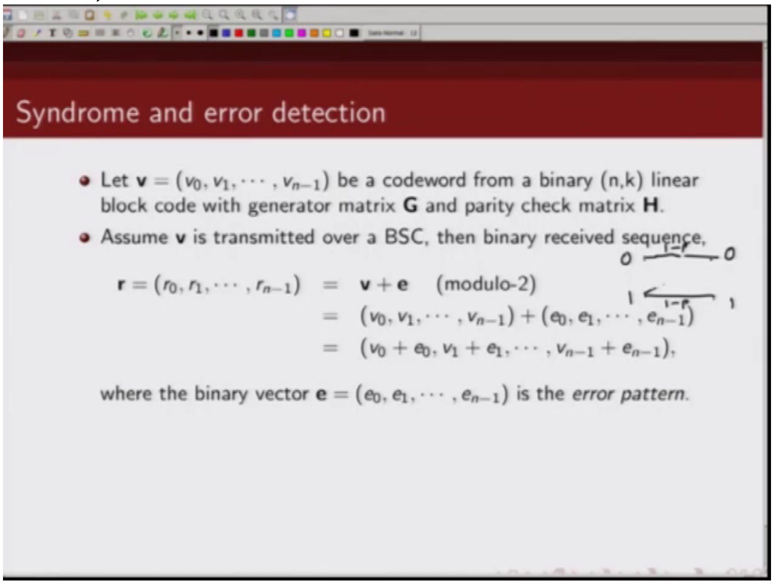
0s and 1 and with probability let's say 1 minus, probability

(Refer Slide Time 02:11)



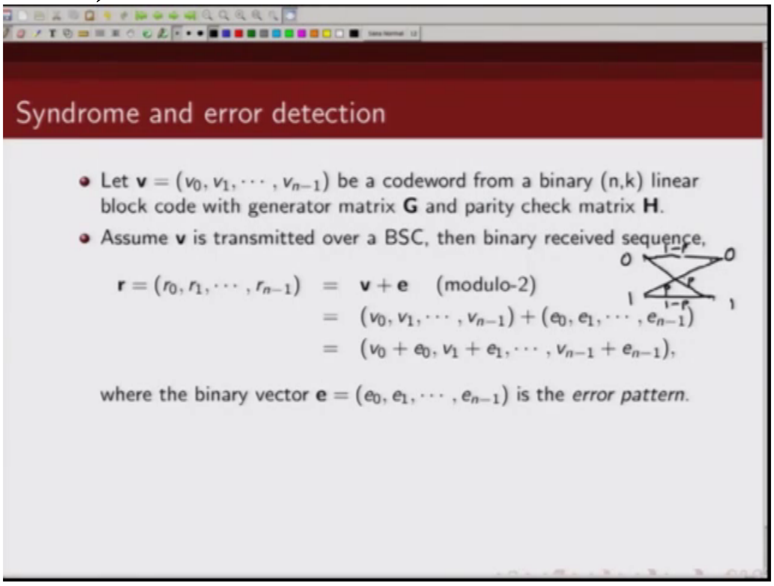
1 minus p, we receive the bits correctly and there is

(Refer Slide Time 02:16)



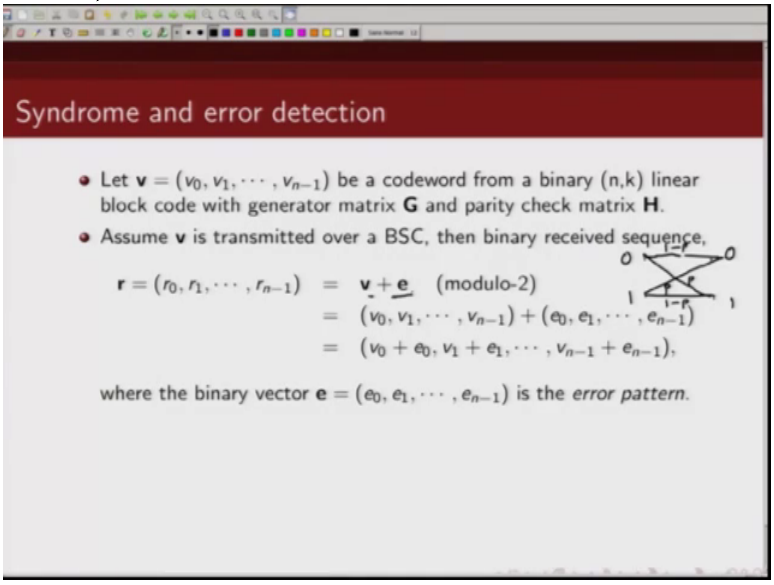
a crossover probability of p that the bits get flipped So this is a binary symmetric

(Refer Slide Time 02:22)



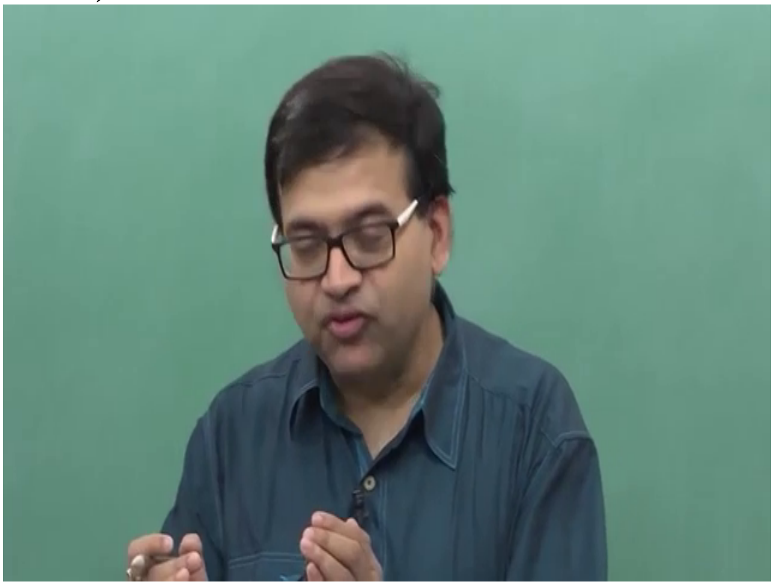
channel Let's denote by r our received sequence, received codeword which we sent over this binary symmetric channel. So the types of, as I said, the outputs of binary symmetric channel is also 0 and 1, so we can describe the output of a binary symmetric channel r as our transmitted codeword plus some

(Refer Slide Time 02:53)



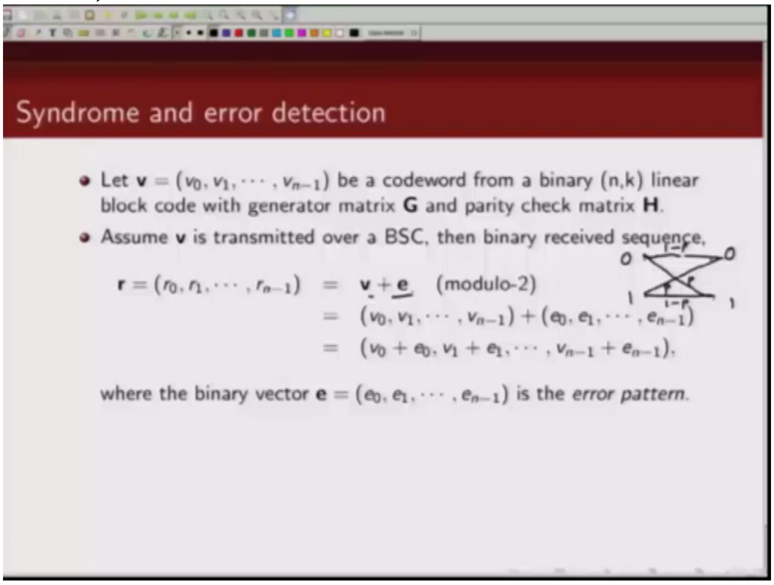
error vector So our transmitted codeword is an n-bit tuple. Similarly

(Refer Slide Time 03:00)



our error vector is also an n-bit.

(Refer Slide Time 03:03)

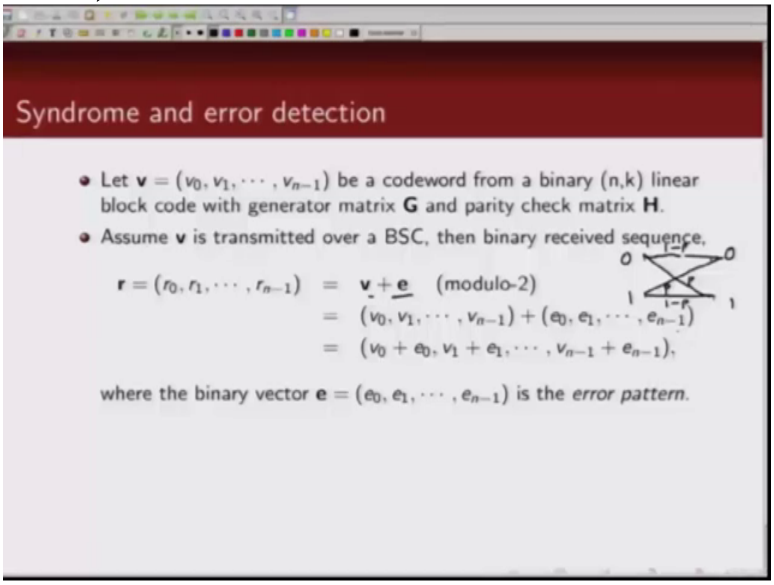


So we describe our error vector by e 0, e 1, e 2, e n minus 1 and whenever e i term is 1, that means that particular bit was not received correctly.

(Refer Slide Time 03:18)

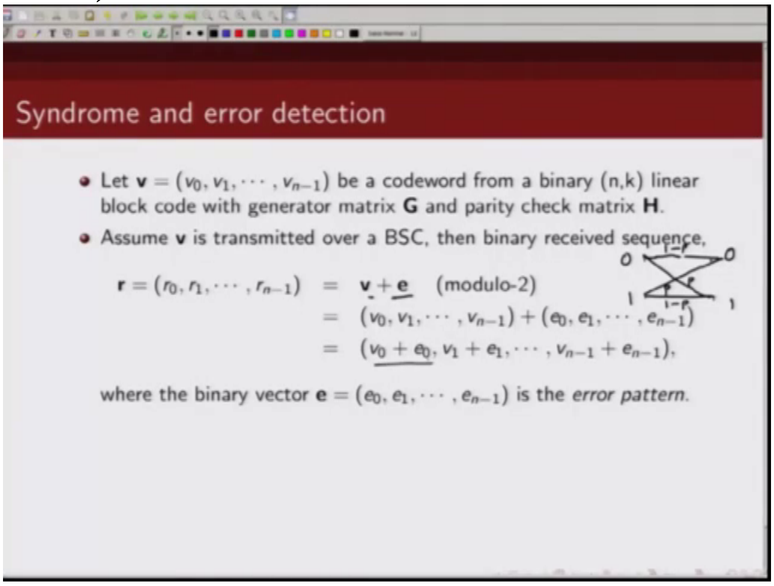
So we can write an output of a binary symmetric channel.

(Refer Slide Time 03:23)



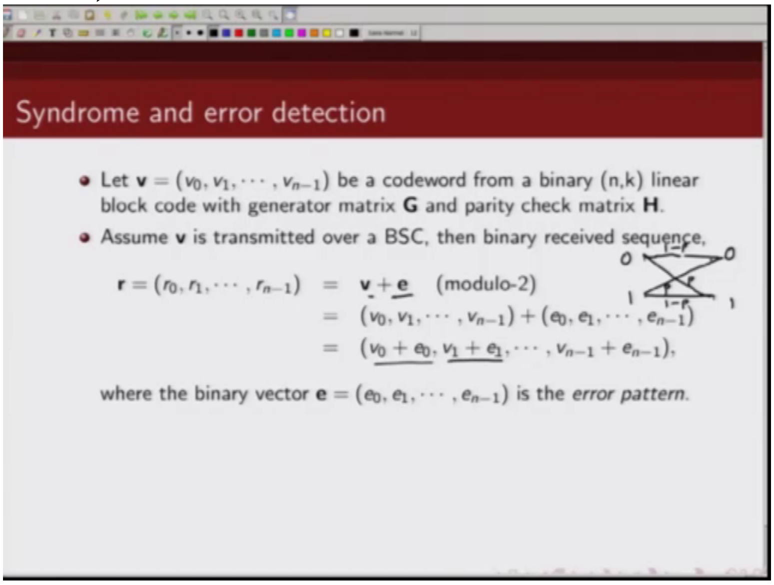
So the first bit that we would receive is basically

(Refer Slide Time 03:26)



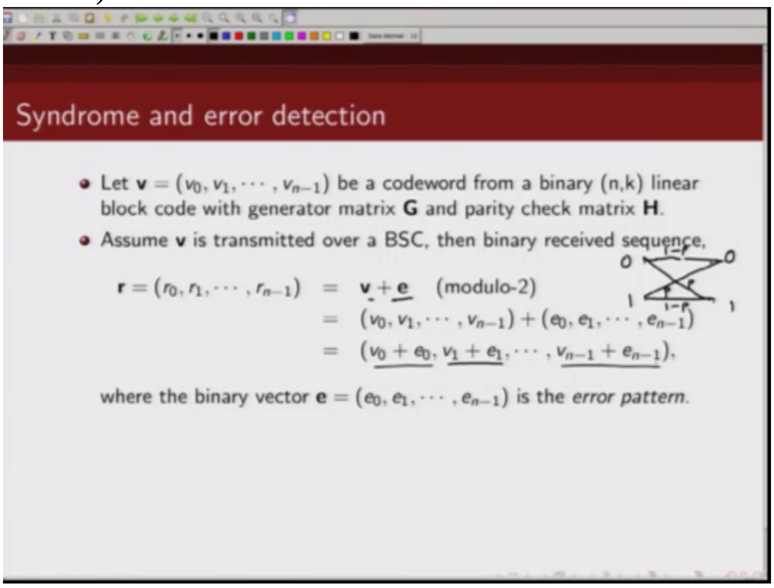
v 0 plus e 0, v 1 plus, second

(Refer Slide Time 03:30)



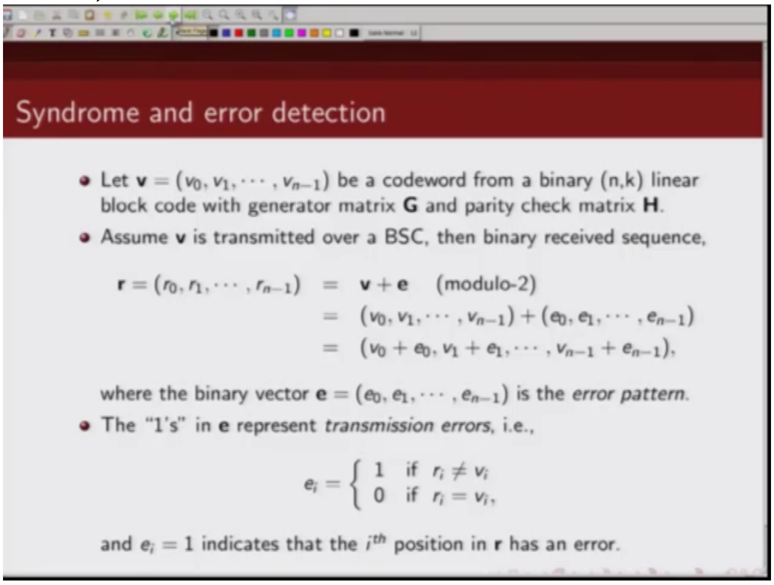
bit will be v 1 plus e 1, similarly v 2 plus e 2 and last finally we will get v n minus 1

(Refer Slide Time 03:36)



and e n minus 1 where this e 0, e 1, e 2, e n minus 1 is my error pattern. Now when does an error

(Refer Slide Time 03:49)

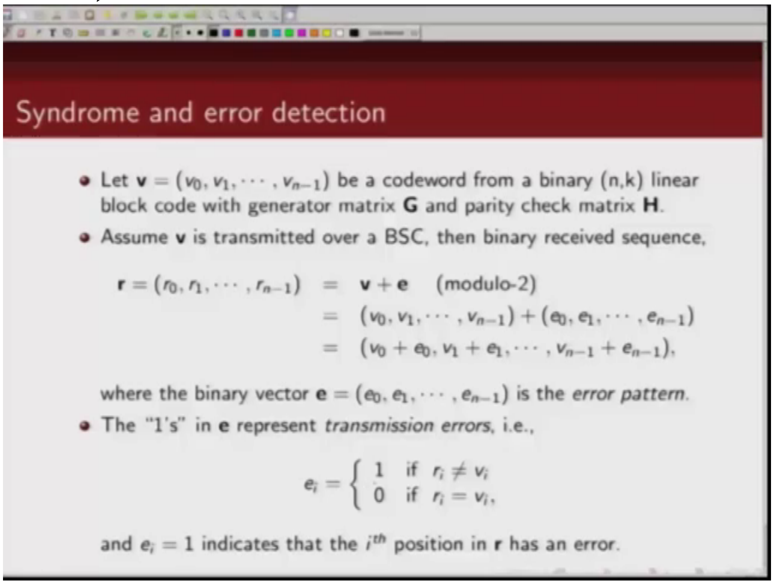


occur? If my received sequence is not same as my

(Refer Slide Time 03:53)

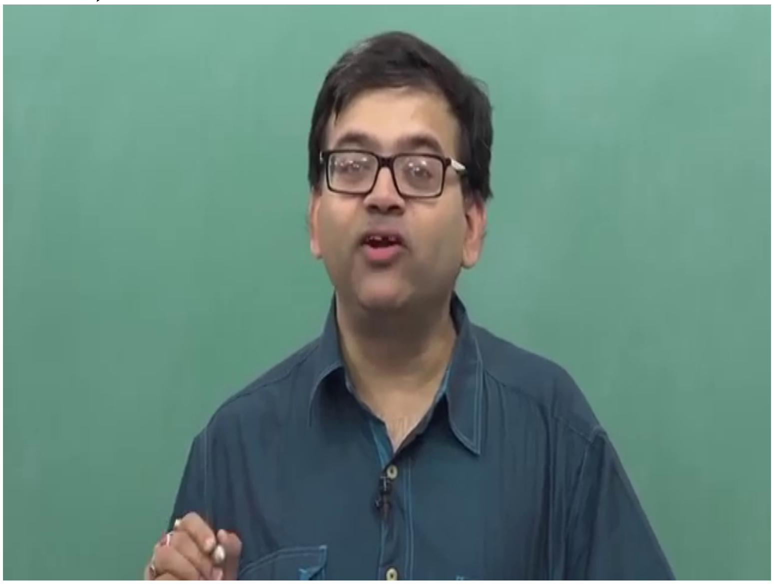
transmitted codeword, then there is an error. So the error is 1

(Refer Slide Time 03:58)



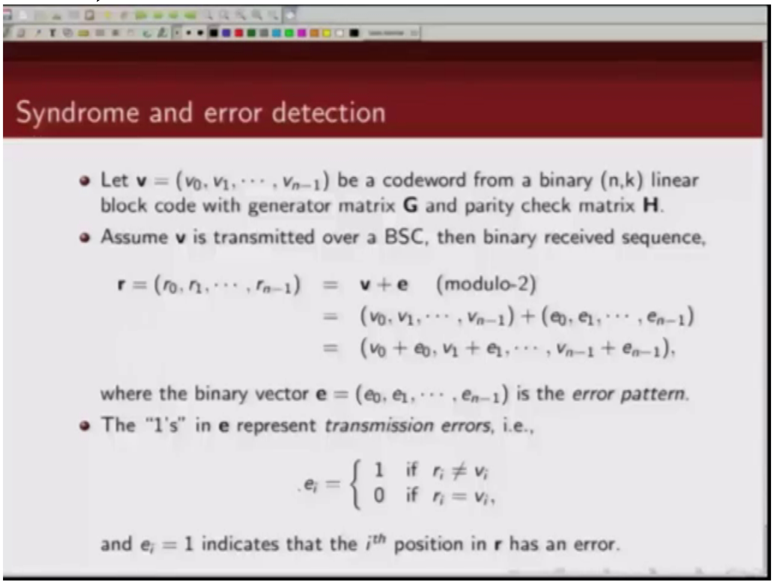
only if my received bit is not same as my transmitted bit. If the received bit is same as my transmitted bit, there is no error. That I will keep that bit e i as 0, Ok; now if a particular bit e i is 1, what does it denote? It denotes that,

(Refer Slide Time 04:23)



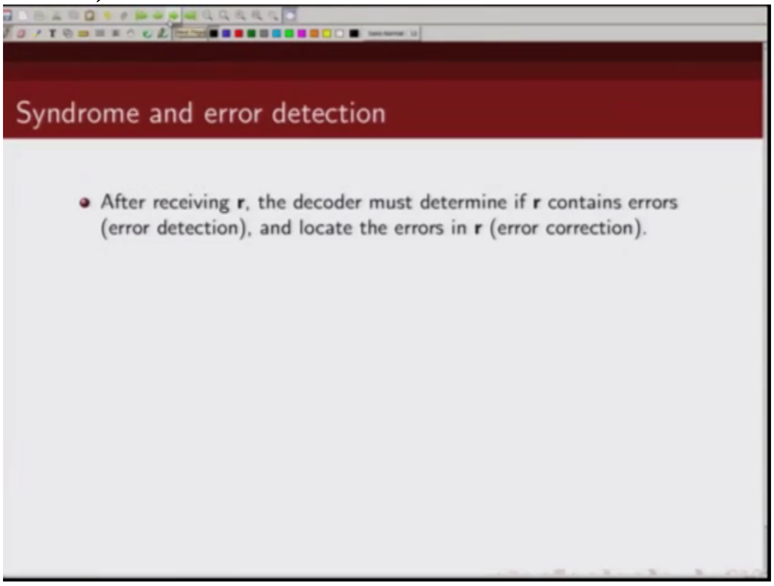
that i'th bit is in error. Now when we are sending this codeword over a communication channel, what are we interested in? We are interested to find out whether any error has occurred. If any error has occurred, we are interested to find out the location where the error has occurred so that we can correct those errors, Ok.

(Refer Slide Time 04:53)



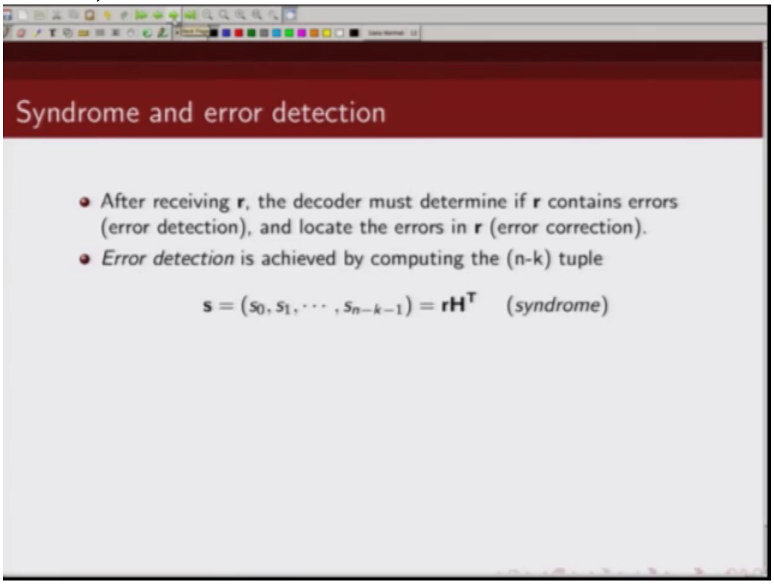
So first we are going to

(Refer Slide Time 04:55)



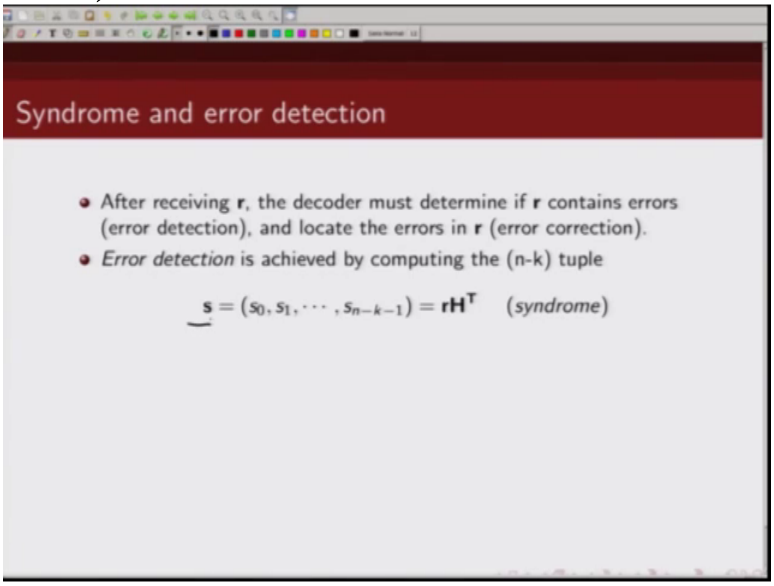
show how we can detect error. So we

(Refer Slide Time 05:01)



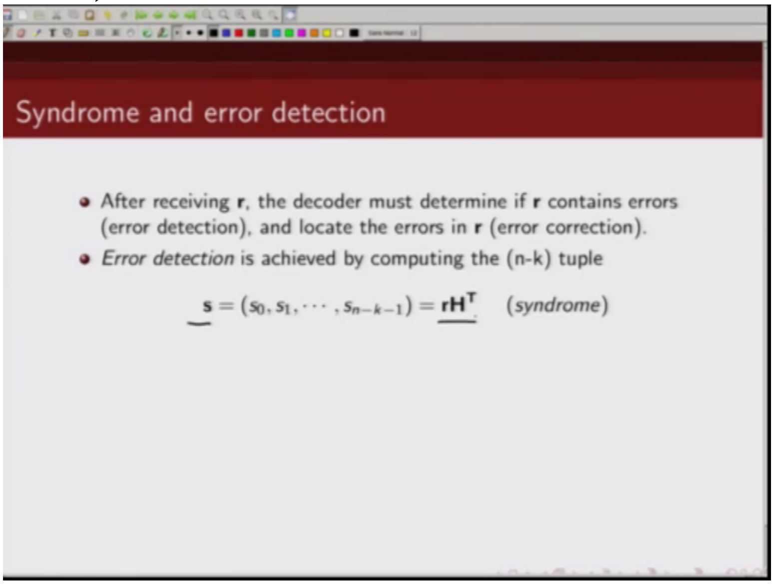
define a term which we call as Syndrome.

(Refer Slide Time 05:04)



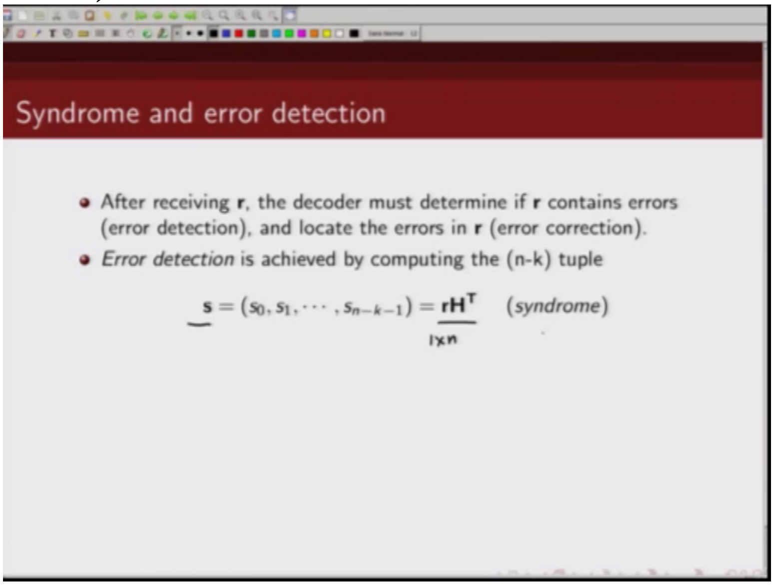
What is a Syndrome? Syndrome is computed by computing this

(Refer Slide Time 05:11)



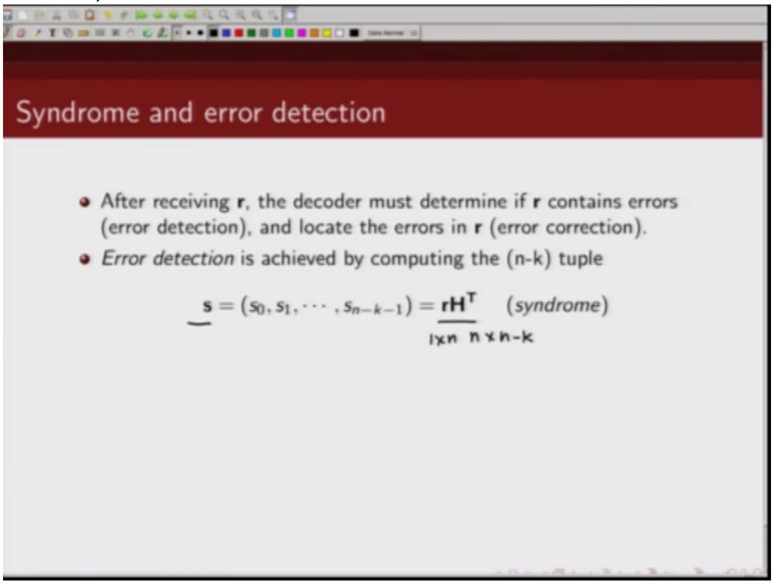
r H transpose So r is basically 1 cross n vector. H is n minus k

(Refer Slide Time 05:18)



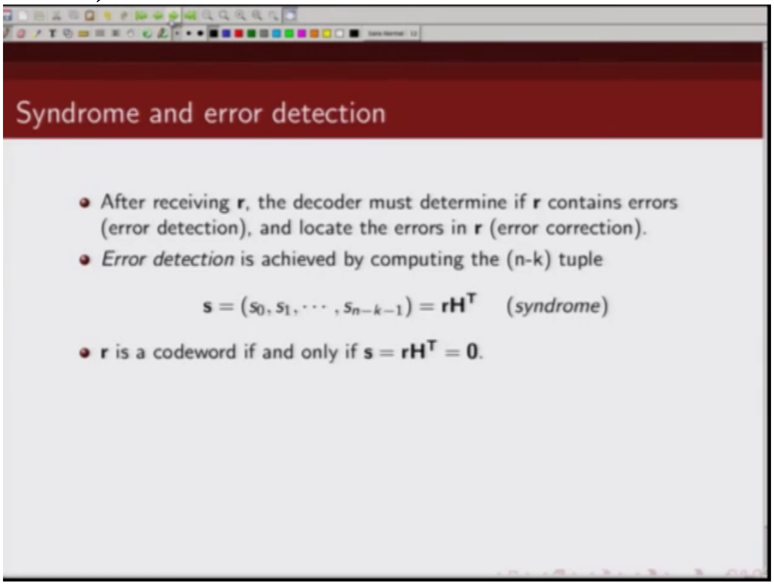
cross n vector, so H transpose is n into n minus k

(Refer Slide Time 05:24)



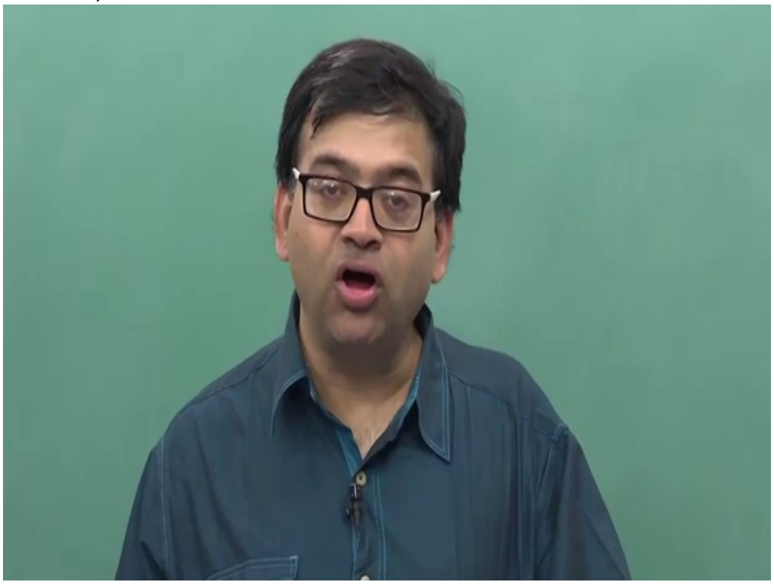
matrix So this term r H transfer is known as Syndrome and this is, when this is non zero,

(Refer Slide Time 05:34)



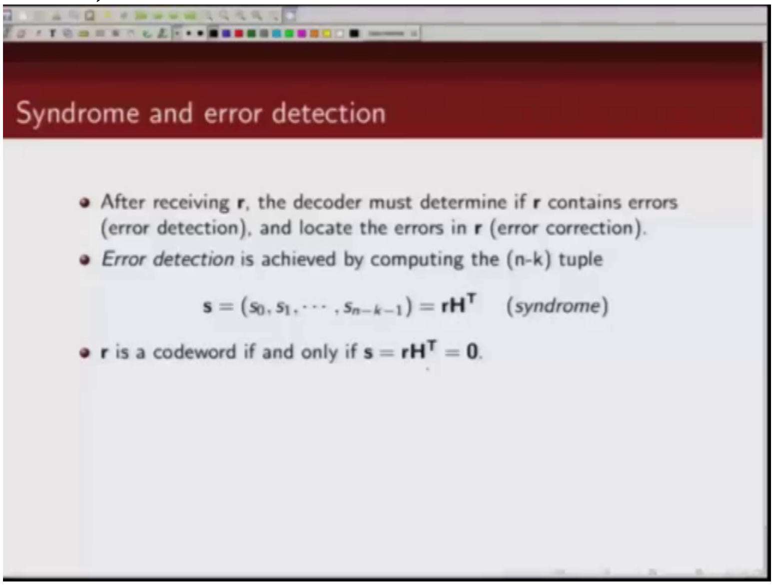
it indicates there is an error. So r is a codeword if and only if the Syndrome is 0. So whenever the Syndrome is 0, basically if the Syndrome is 0, then r is a codeword if and only if Syndrome is 0 and this is easy to check because we know

(Refer Slide Time 06:03)



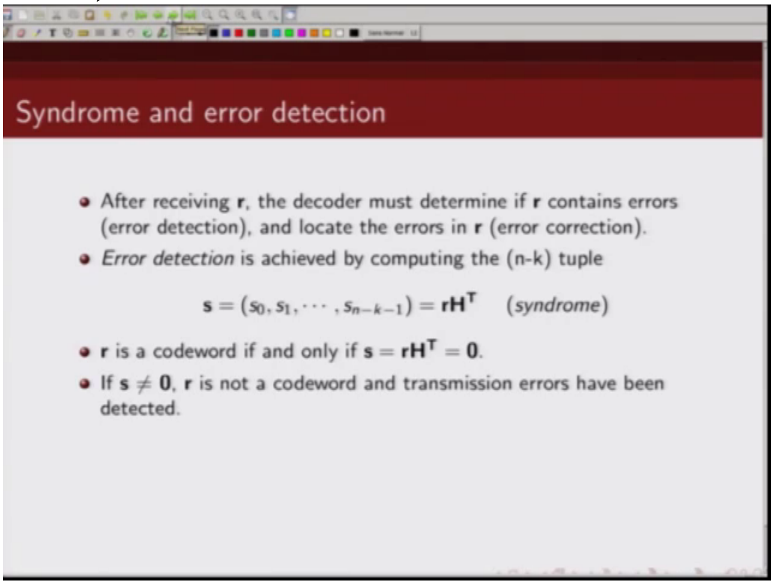
that if v is a valid codeword, then v H transpose will be 0. So if there is no error, my received sequence r would be just equal to v. And then the Syndrome would be

(Refer Slide Time 06:19)



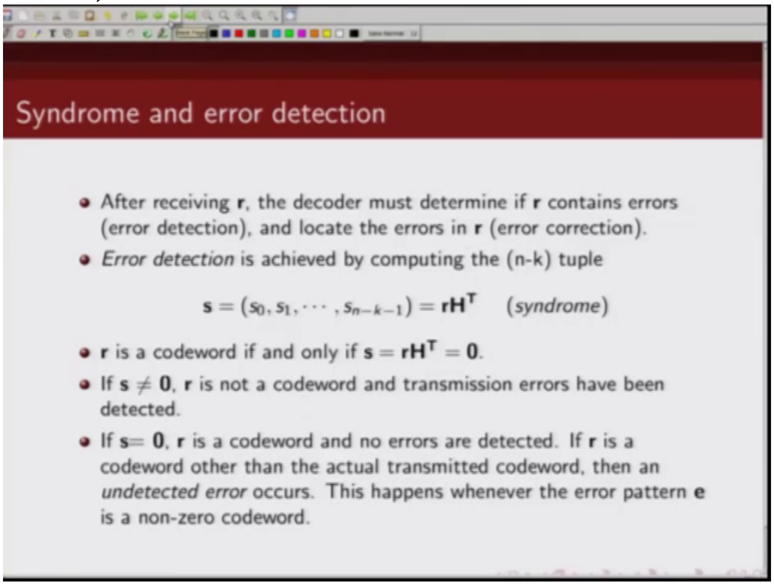
v H transpose which is 0. And if Syndrome

(Refer Slide Time 06:25)



is not equal to 0, then it means there is an error. Now if the Syndrome is 0, does it always mean that there is no error? No. There is a category

(Refer Slide Time 06:44)

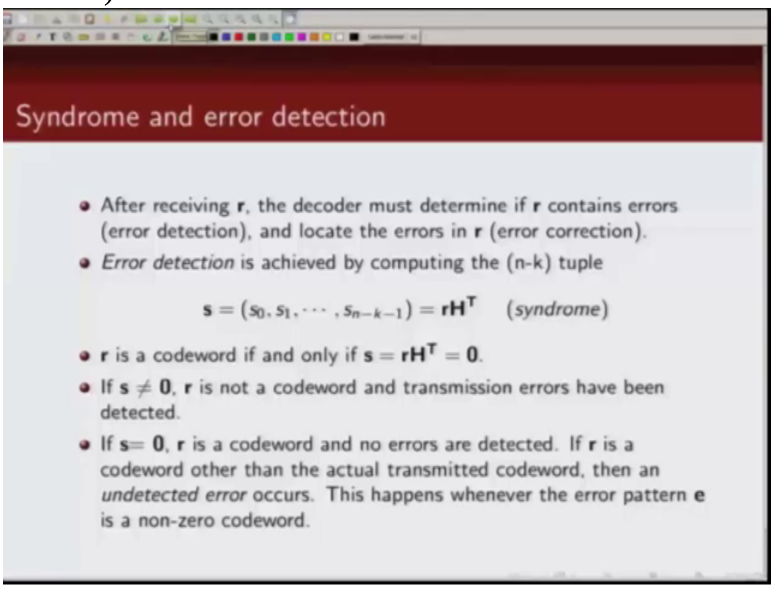


of error which we call as undetected error

(Refer Slide Time 06:50)

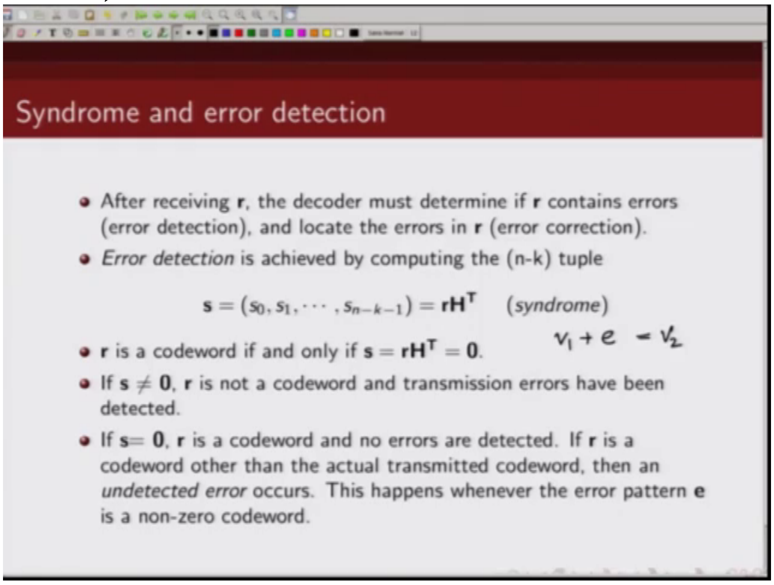
Now when does an undetected error happen? If your s

(Refer Slide Time 06:56)



is 0 and your received sequence r is not the codeword that you transmitted but some other codeword, let's say I transmitted a codeword, v 1 and my error e was such that that it transformed it into another codeword v 2. So received sequence is v 2.

(Refer Slide Time 07:22)

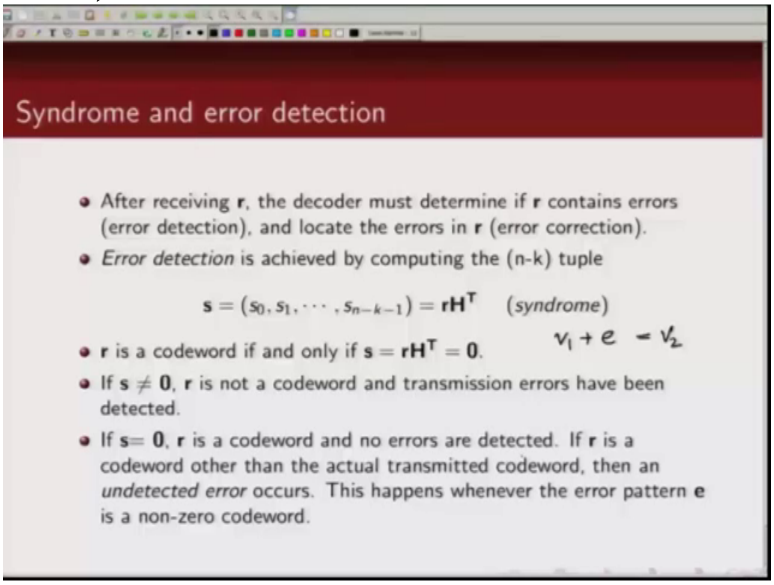


if I compute Syndrome because v 2 is a valid codeword then v H transpose will be 0. So an undetected error happens when your error pattern is such that it transforms your one codeword to

(Refer Slide Time 07:40)

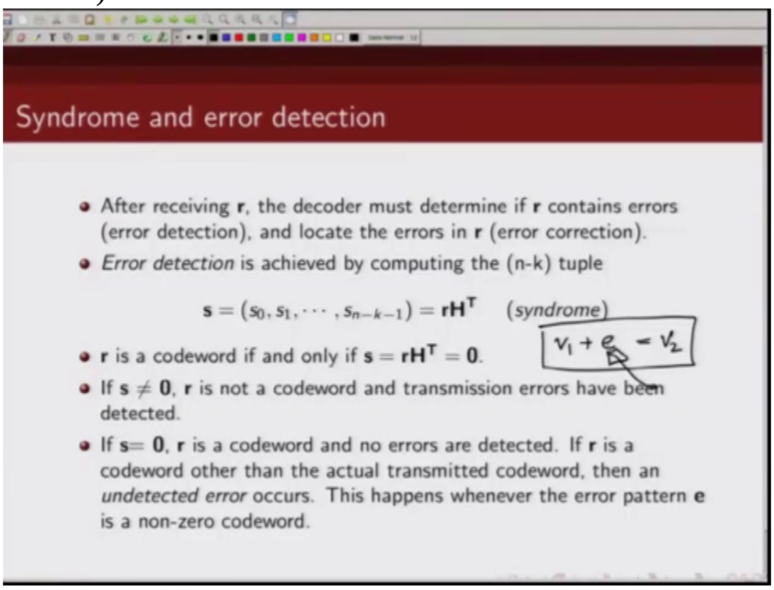
another codeword And that's what we have written here.

(Refer Slide Time 07:43)



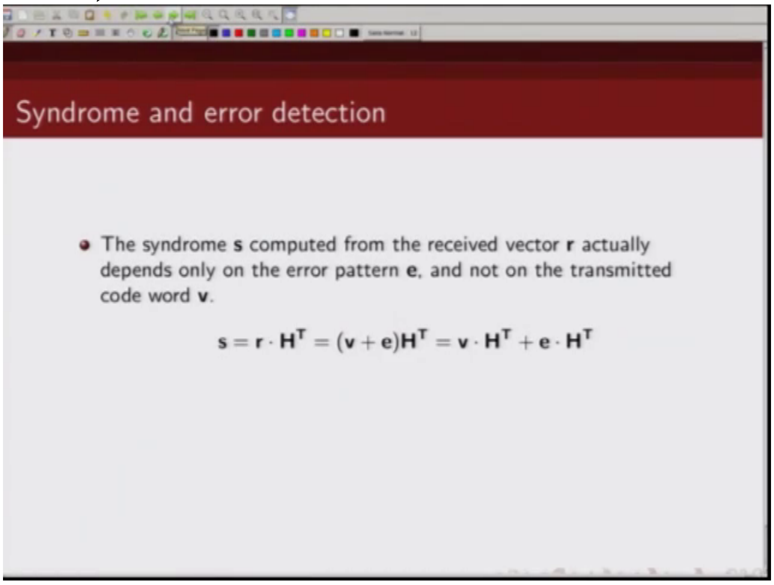
So if s is 0 and r is a codeword that means no errors are detected. However if r is a codeword other than the actual codeword transmitted then an undetected error has happened. And this happens when an error pattern is a non-zero codeword; because we have said a property of linear block code that sum of 2 codewords is also a codeword. So for this scenario to happen, this e has to be a

(Refer Slide Time 08:20)



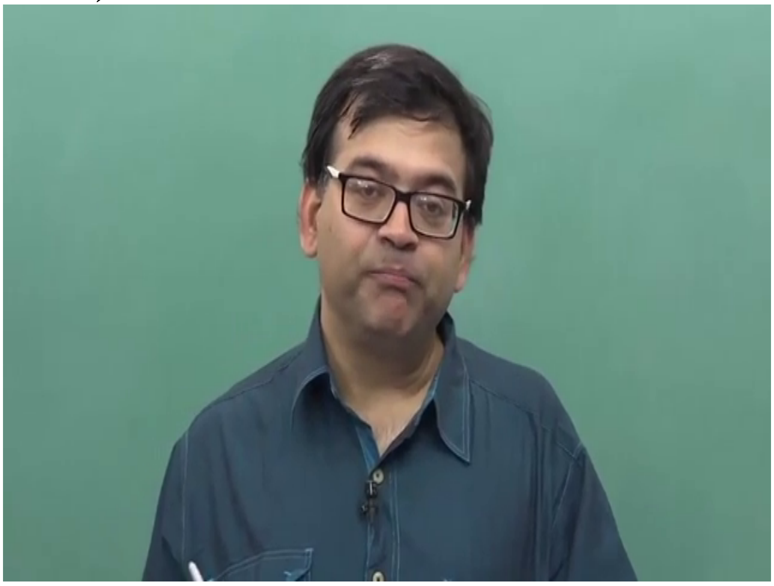
valid non-zero codeword So as I said

(Refer Slide Time 08:25)



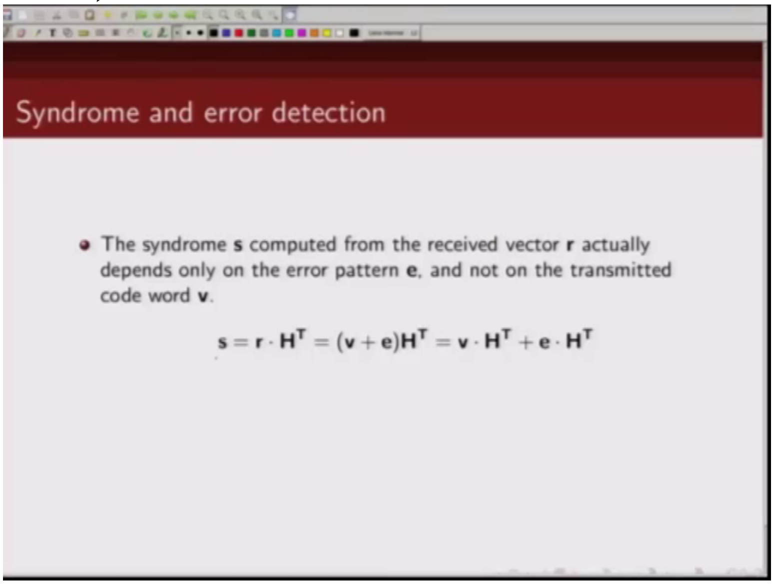
we compute the Syndrome from the received sequence r, the interesting part is

(Refer Slide Time 08:32)



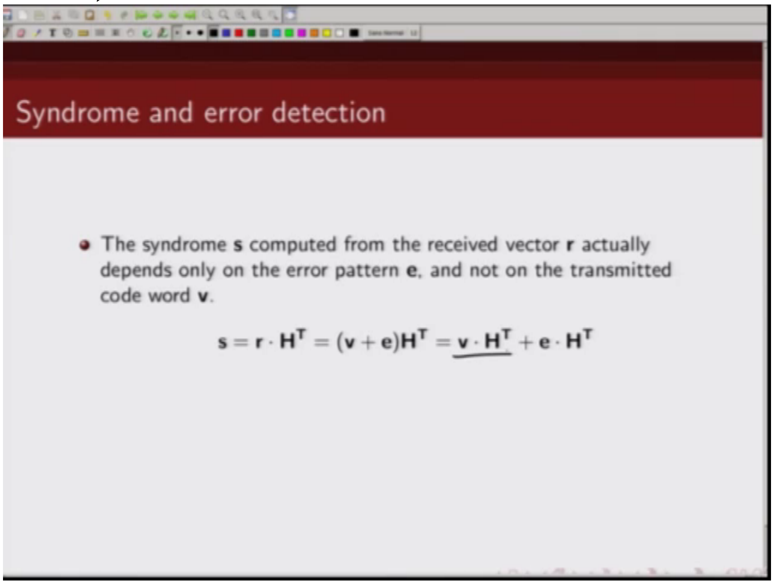
Syndrome depends only on the error pattern. It does not depend on what codeword was transmitted. And this is easy to see.

(Refer Slide Time 08:43)



So Syndrome is r H transpose which we can write r is my transmitted codeword plus error vector, this whole multiplied by H transpose. So this I can write as v H transpose and e H transpose. Now what is v H transpose?

(Refer Slide Time 09:04)

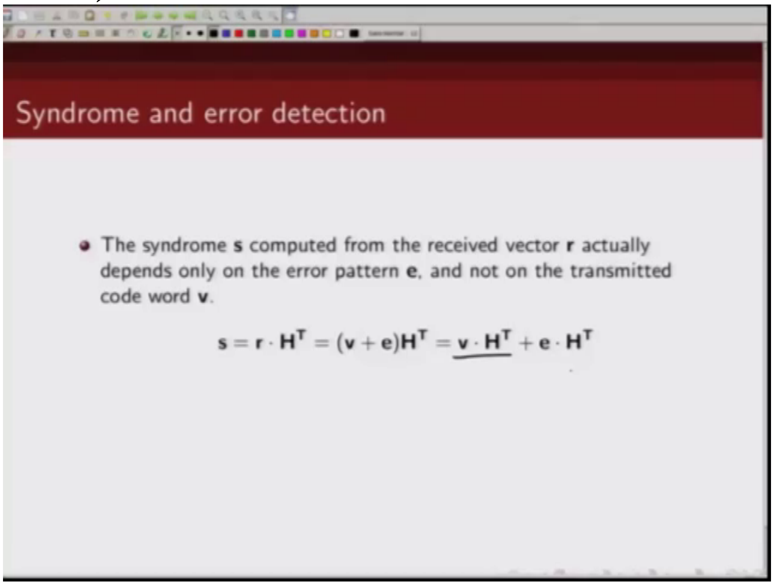


v H transpose is 0, because v is a valid codeword.

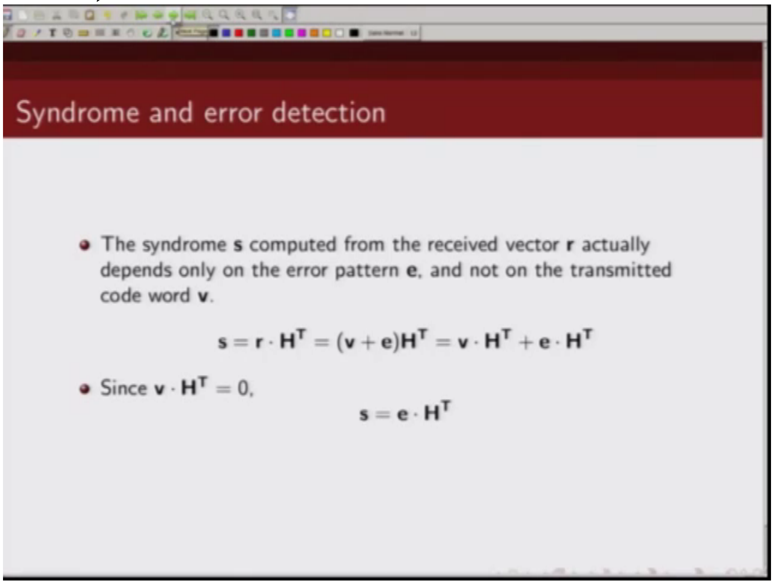
(Refer Slide Time 09:08)

Then Syndrome is nothing but e H transpose.

(Refer Slide Time 09:12)

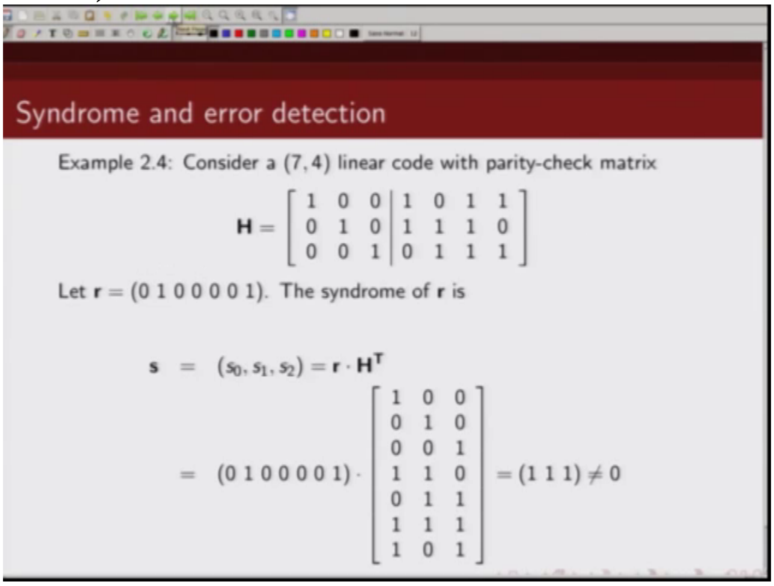


(Refer Slide Time 09:14)



So Syndrome does not depend on transmitted codeword. It only depends on the error pattern.

(Refer Slide Time 09:23)

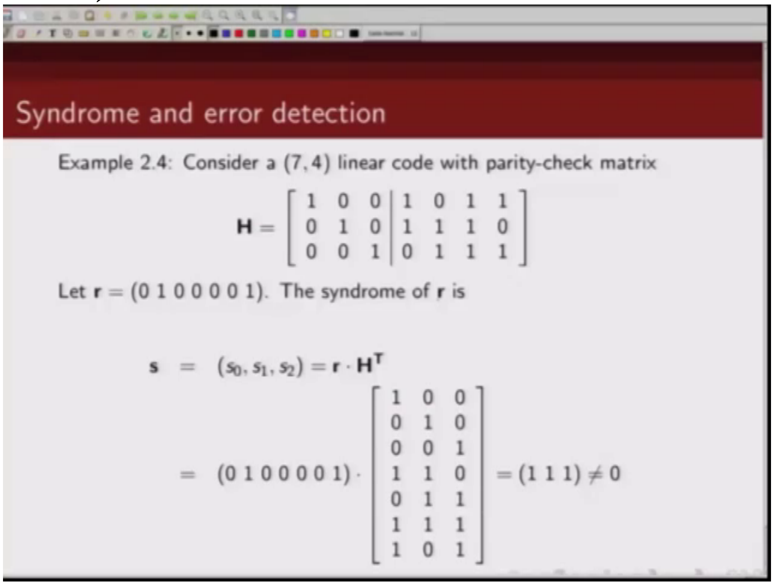


So I have an example here. For 7 4 linear block code whose parity check matrix is given by this. Now let's say my received sequence is, received coded sequence is this. And I am interested to find whether there is any error in this received sequence. So how do I do that?

(Refer Slide Time 09:45)

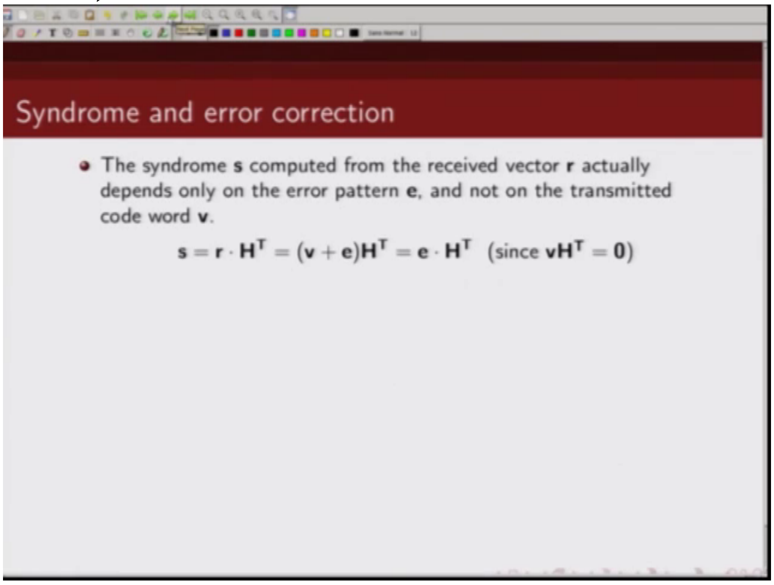
So first I will compute the Syndrome. So what is Syndrome

(Refer Slide Time 09:50)



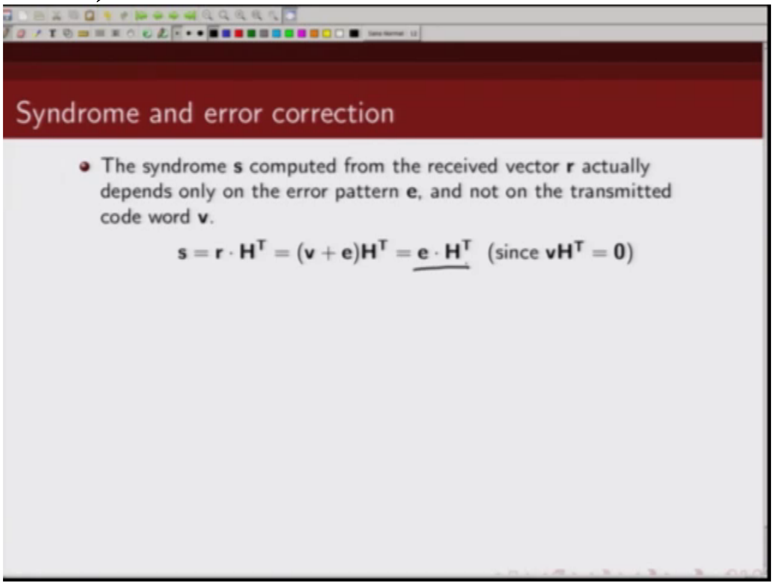
of r? This Syndrome is r H transpose, so r is this and H is given, so H transpose is basically 1 0 0 0 1 0 0 0 1, so this is my H transpose and when I multiply this, I multiply this by this, multiply this by this, multiply this by this what I get is 1 1 1 which is not 0 that means there is an error in my received sequence.

(Refer Slide Time 10:27)



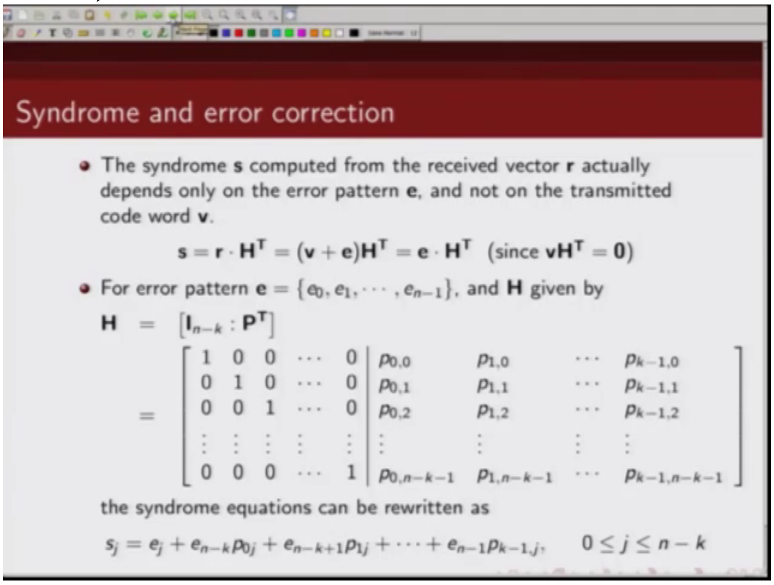
As I said the Syndrome only depends on the error pattern. It does not depend

(Refer Slide Time 10:32)



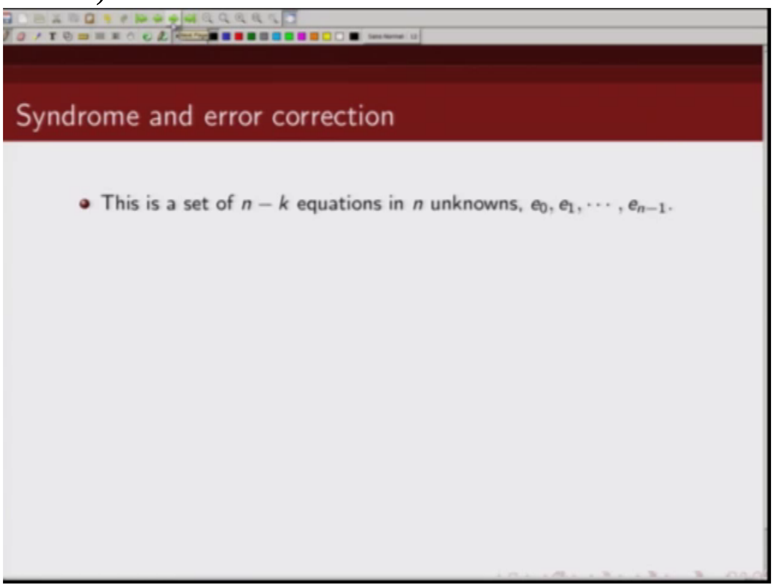
on transmitted codeword.

(Refer Slide Time 10:35)



And if you have your parity check matrix in a systematic form then you can write your Syndrome equations in terms of error patterns like this. So these are your basically n minus k Syndrome equations.

(Refer Slide Time 10:54)

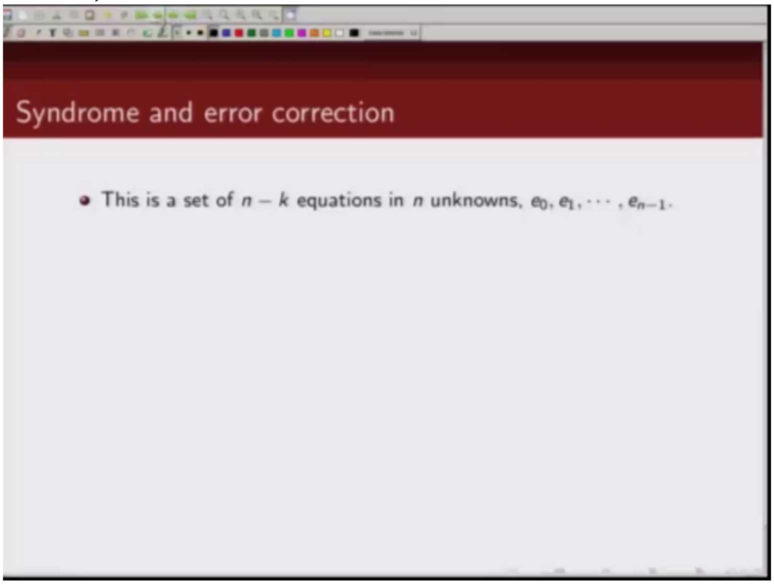


So to recap what we have said so far, we have said that for error detection we need to compute the Syndrome.

(Refer Slide Time 11:08)

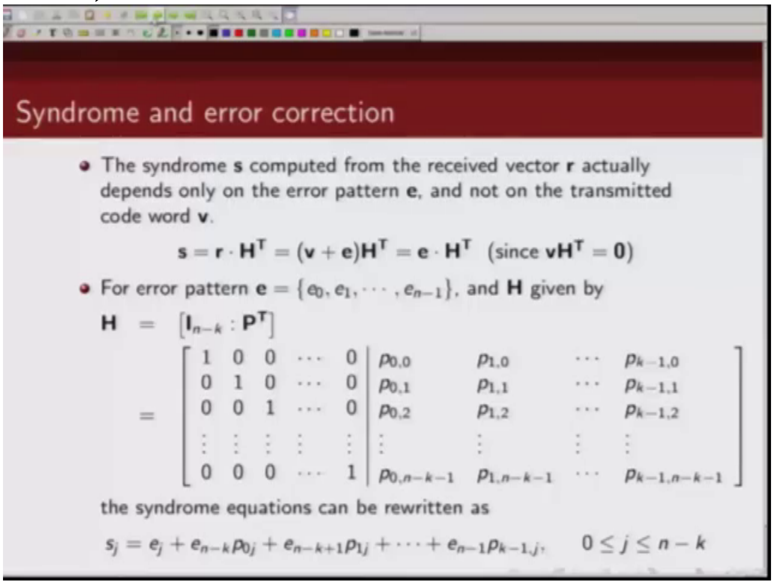
And if the Syndrome is non-zero that means an error has occurred. Now we will move into error correction and we will talk about how we can use the Syndrome for error correction. So as we saw basically from the Syndrome

(Refer Slide Time 11:29)



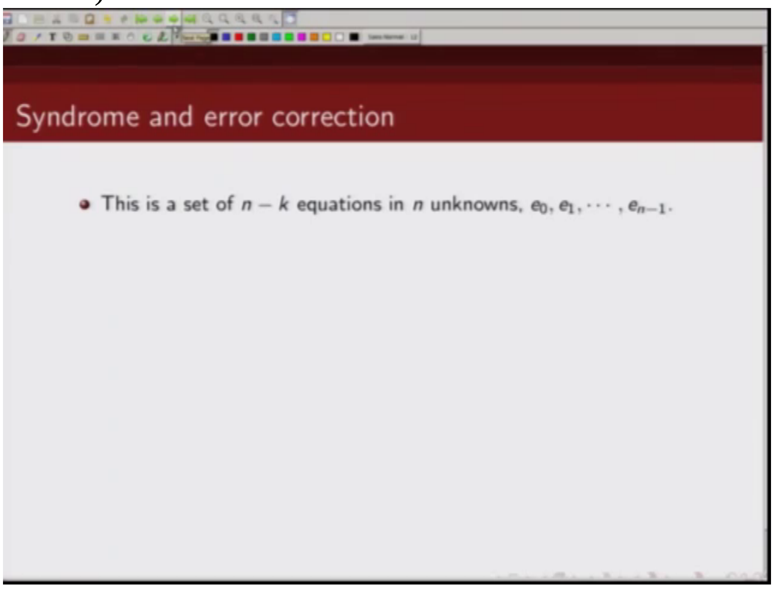
equations

(Refer Slide Time 11:30)



we had basically one cross n and h transpose was n cross n minus k. So we essentially had

(Refer Slide Time 11:41)

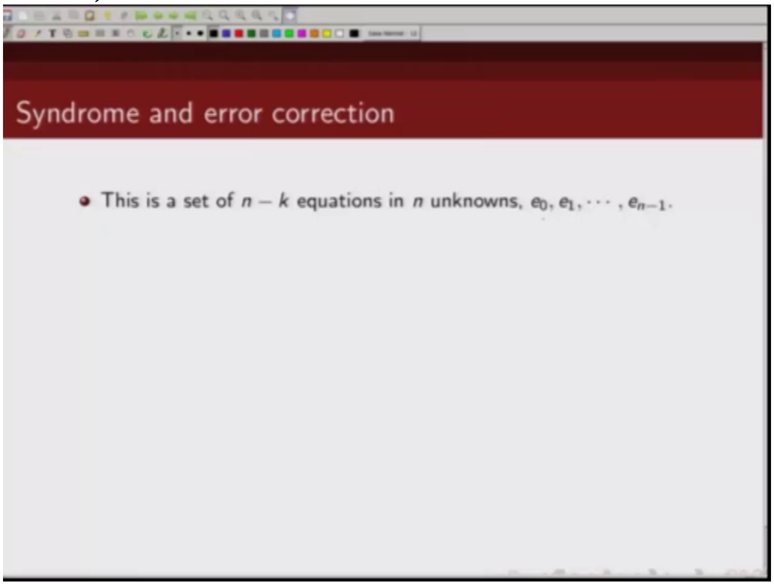


n minus k Syndrome equations So we had total n minus k equations and how many unknowns we have, we have n unknowns from location error at location 1 to error at location

(Refer Slide Time 11:55)

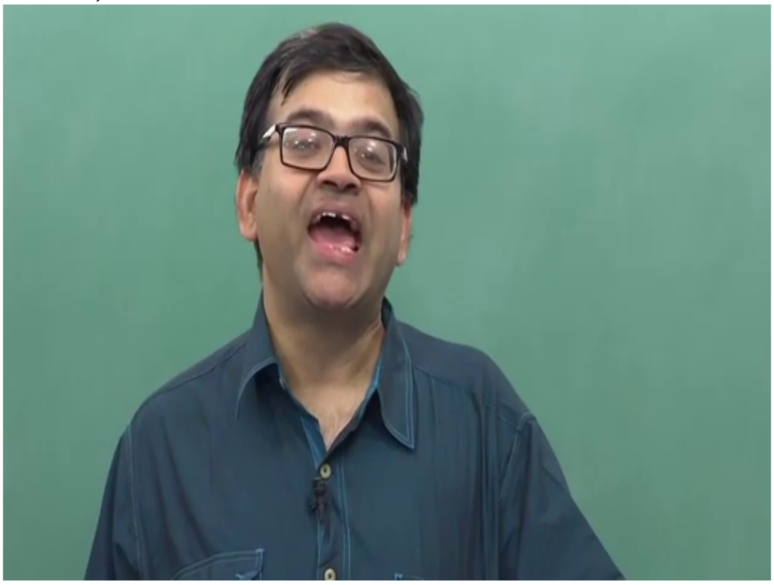
n. So we have n unknowns, e 0, e 1,

(Refer Slide Time 11:59)



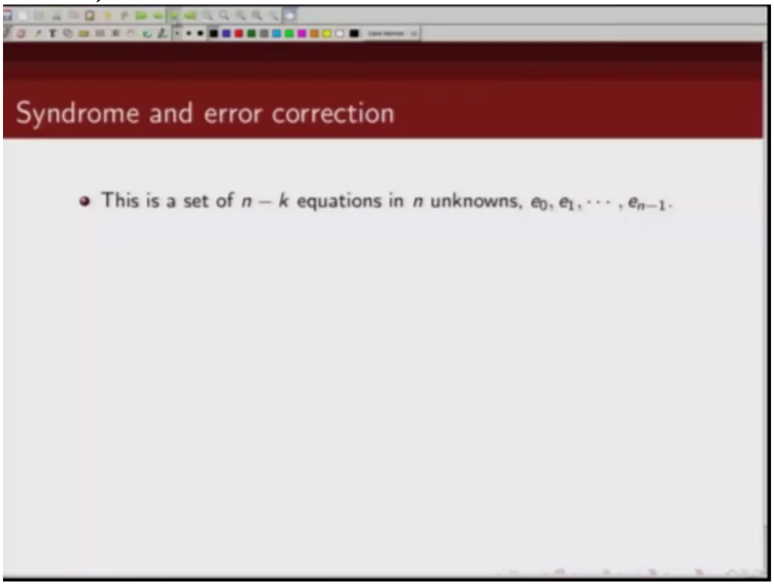
e 2, e n minus 1 where as we have only

(Refer Slide Time 12:04)



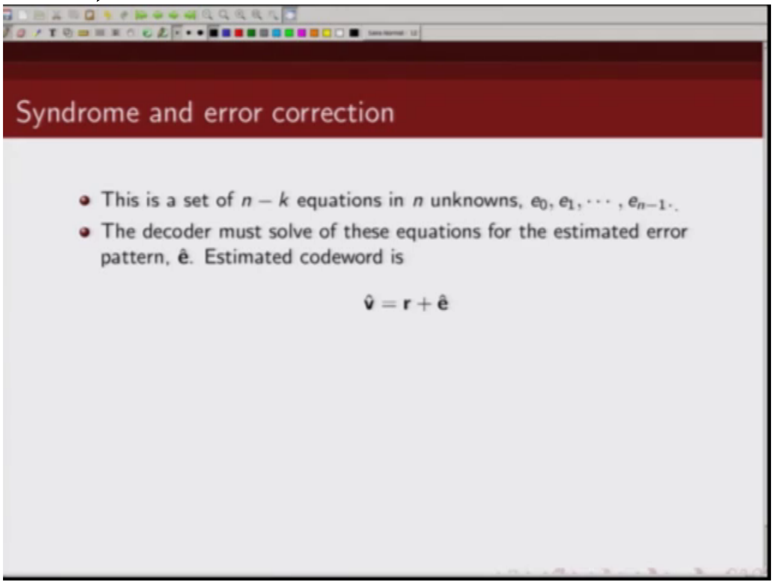
n minus k equations So we have less equations, more unknowns.

(Refer Slide Time 12:10)



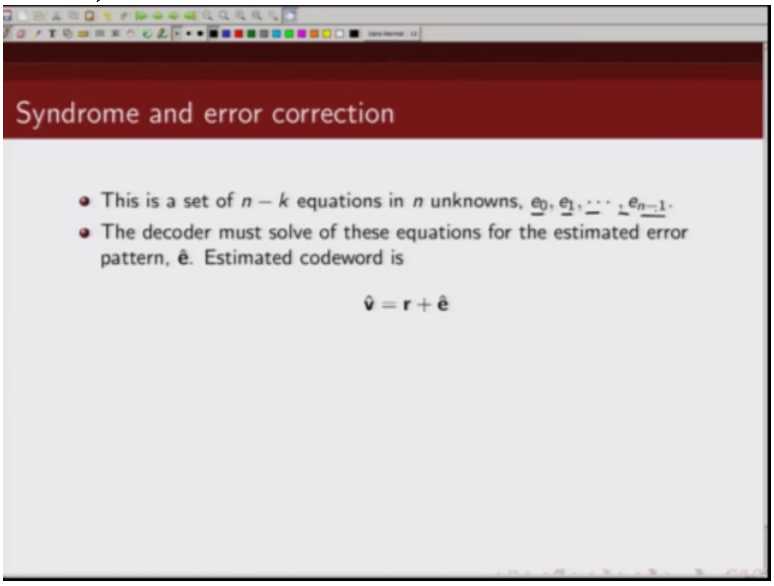
And what we need to do is we need to solve this set of

(Refer Slide Time 12:16)



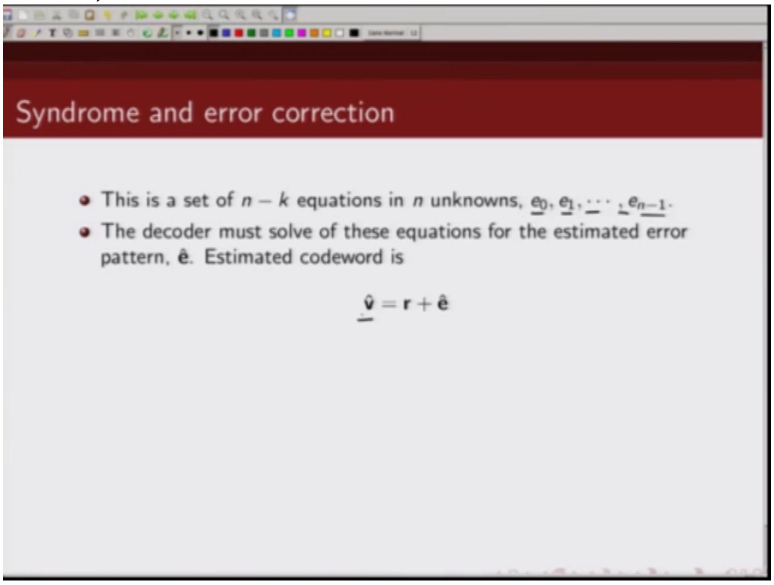
equations to find out what these unknown quantities are Because to find out the error pattern, we need to know what these e i's are,

(Refer Slide Time 12:29)



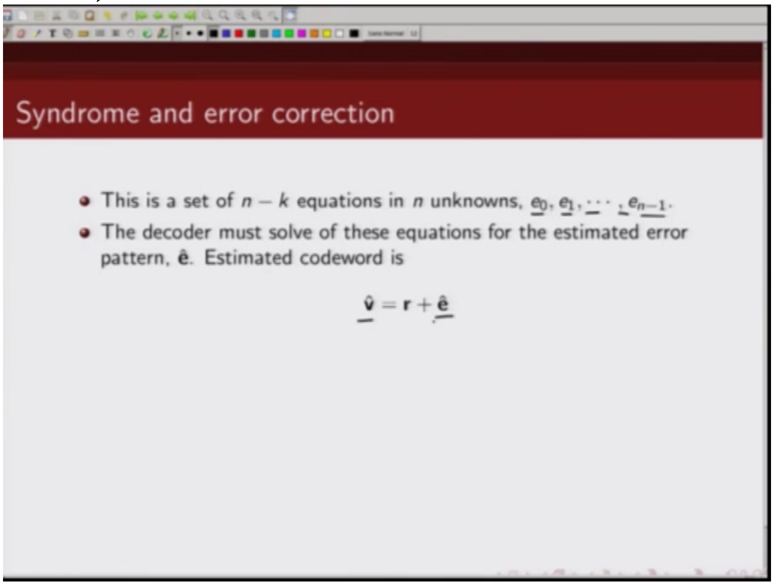
Ok. So we need to solve these set of n minus k equations to get back our corrected sequence. And what would be our corrected sequence?

(Refer Slide Time 12:42)



Estimated codeword would be nothing but r received sequence plus estimated error pattern. So we need to, when we do error

(Refer Slide Time 12:56)

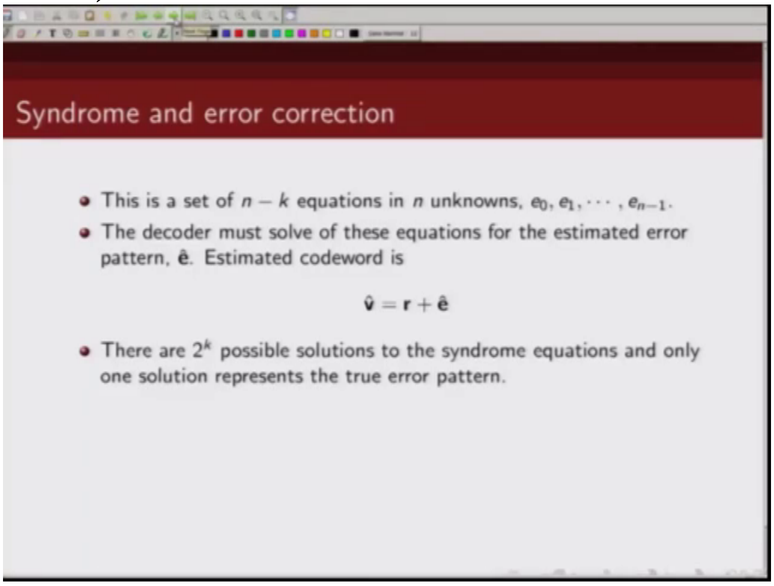


correction, essentially what we are trying to do is

(Refer Slide Time 12:59)

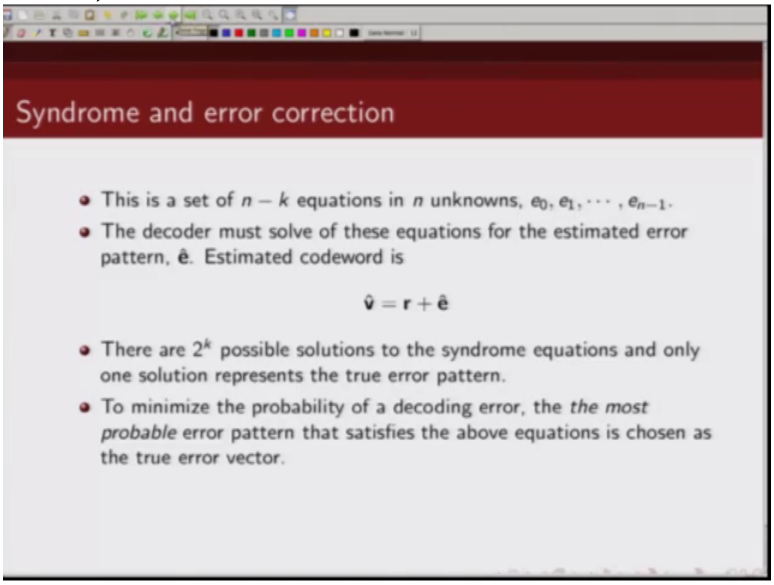
we are trying to estimate what the error pattern is. So as we can see

(Refer Slide Time 13:05)



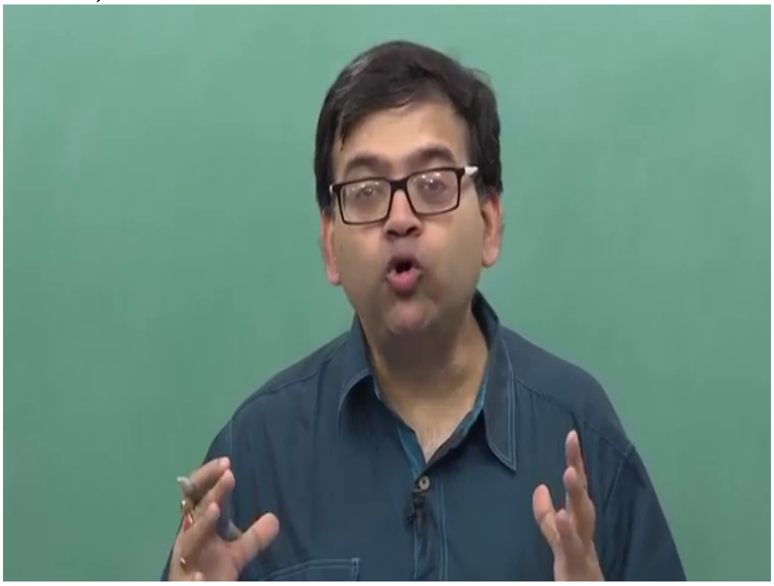
from here, because we have n minus k equations and n unknowns, we have total 2 to the power k solutions of these n minus k equations. So there are total 2 k solutions to these equations and out of them there is only one which is the correct error pattern. Out of those 2 k solutions there is only 1 error pattern

(Refer Slide Time 13:35)



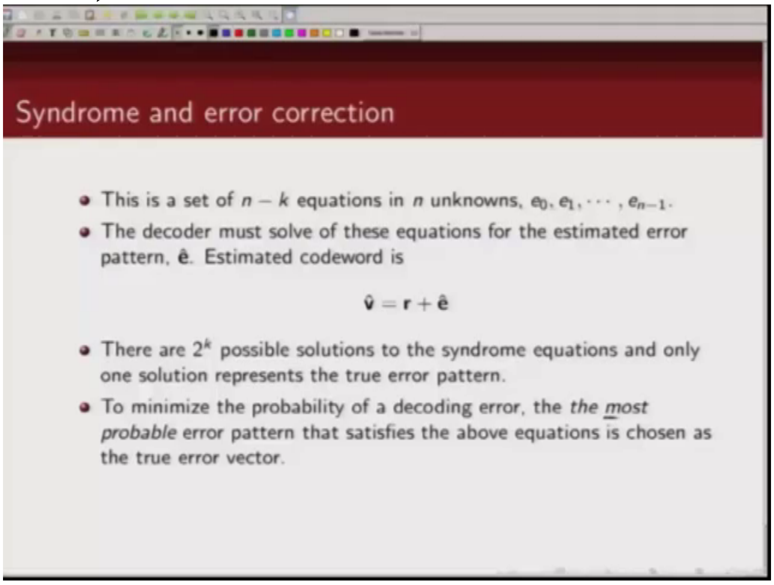
which is correct Now how do we choose

(Refer Slide Time 13:40)



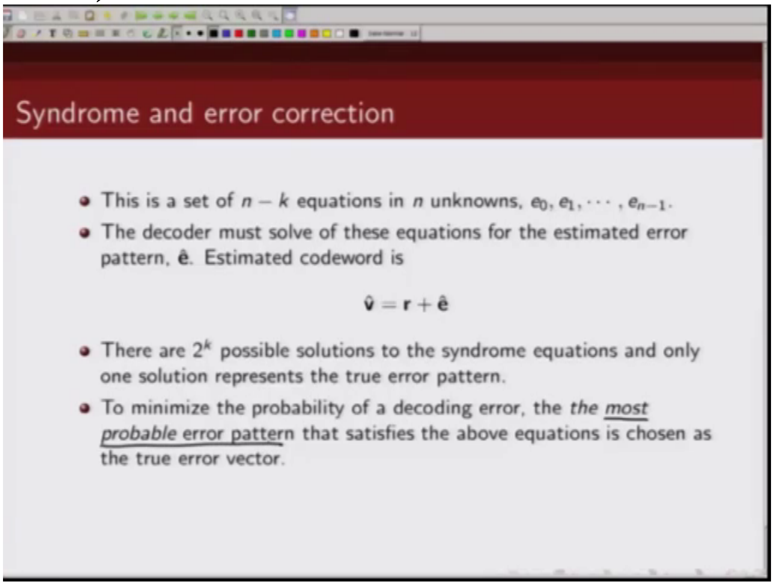
the most likely error pattern from these 2 k solutions? How do we choose the most likely pattern? That's basically what our objective is. So when we try to minimize probability of error we want to choose

(Refer Slide Time 13:57)



the most probable error pattern

(Refer Slide Time 14:00)

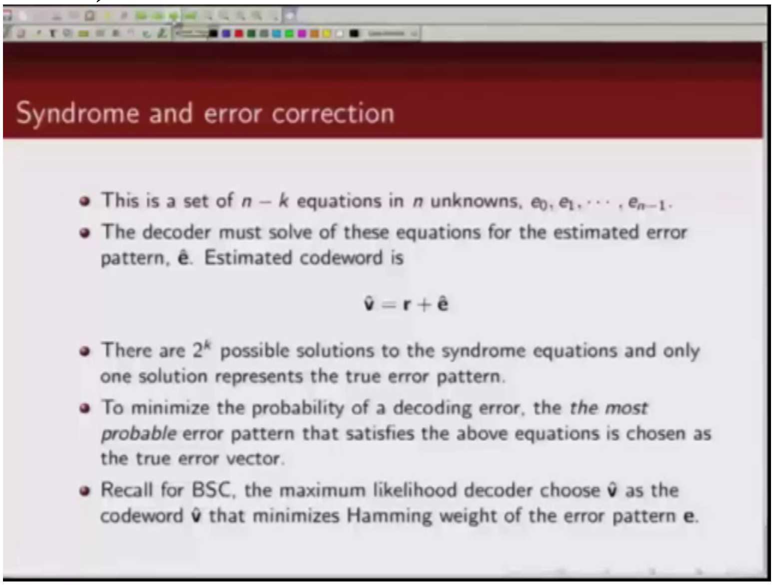


We want to choose the most probable error pattern from those 2 k solutions of this set of equations.

(Refer Slide Time 14:11)

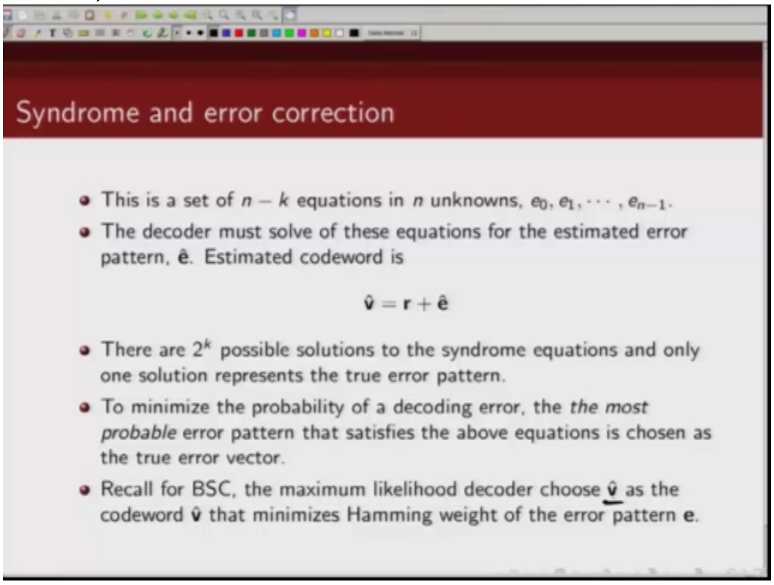
And as we said, we did an exercise, you know,

(Refer Slide Time 14:17)



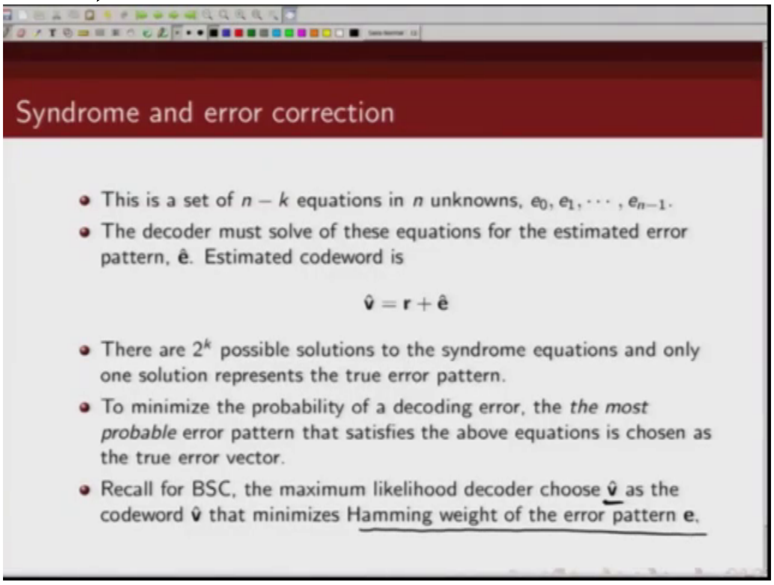
in previous lectures we have shown the decoding rule for maximum likelihood decoding rule for a binary symmetric channel and we have shown that for maximum likelihood decoder we will choose

(Refer Slide Time 14:35)



our estimated codeword as one that will minimize the Hamming distance between the received codeword and the transmitted codeword. In other words it would minimize the Hamming weight

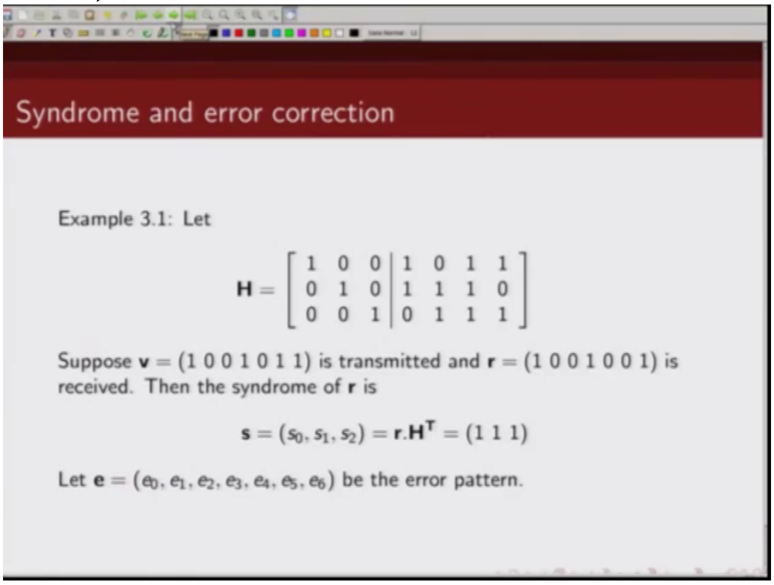
(Refer Slide Time 14:49)



of the error pattern e So from the 2 k different solutions basically, the one which has the least Hamming weight, that's the best solution for the maximum likelihood decoding rule for a binary symmetric channel.

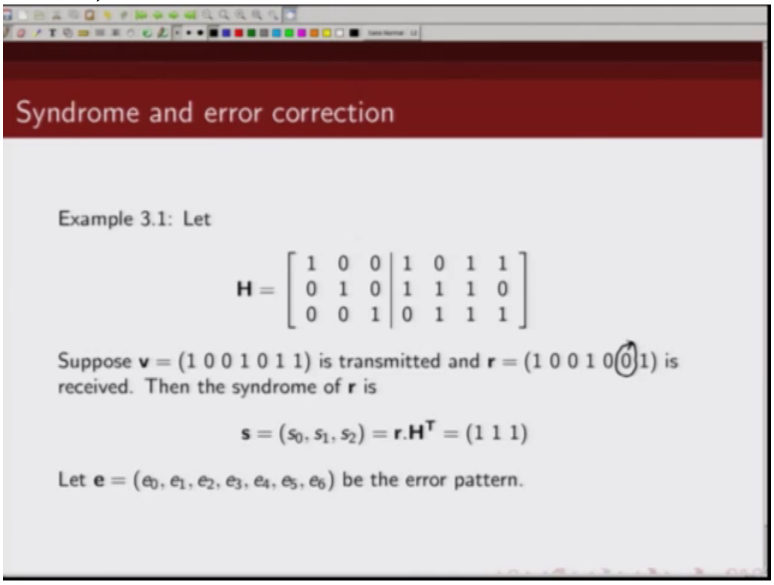
So let's take

(Refer Slide Time 15:09)



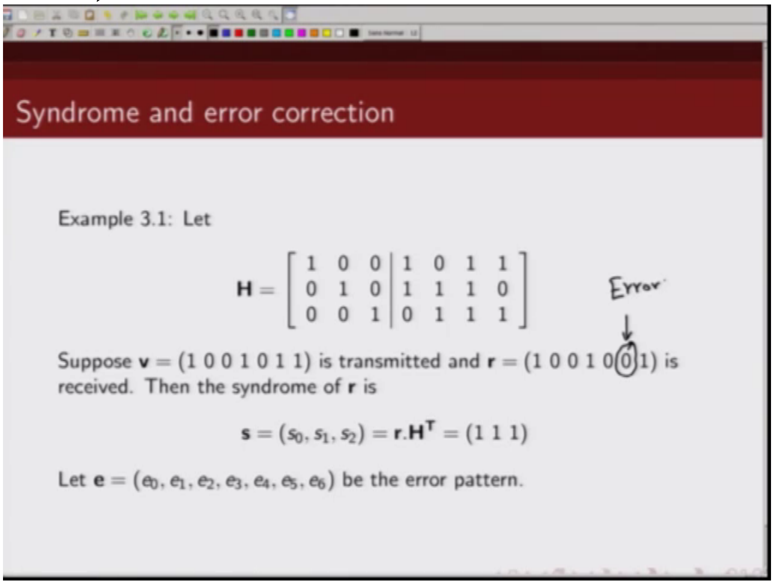
an example now So we have a 7 4 code whose parity check matrix is given by this and our transmitted codeword is this and received codeword is this. We can see that there is an error in this location.

(Refer Slide Time 15:26)



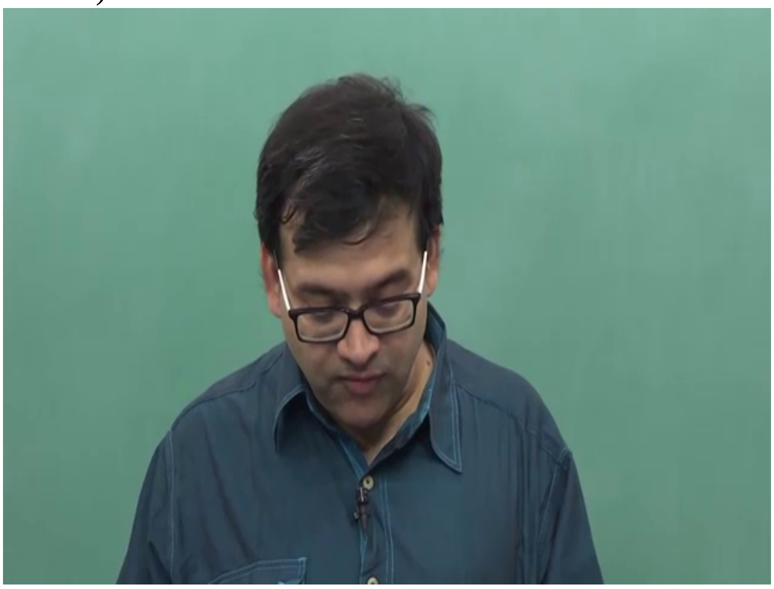
This was transmitted as 1 and this was received as 0, so there is an error in this location. Now how do we find out

(Refer Slide Time 15:38)



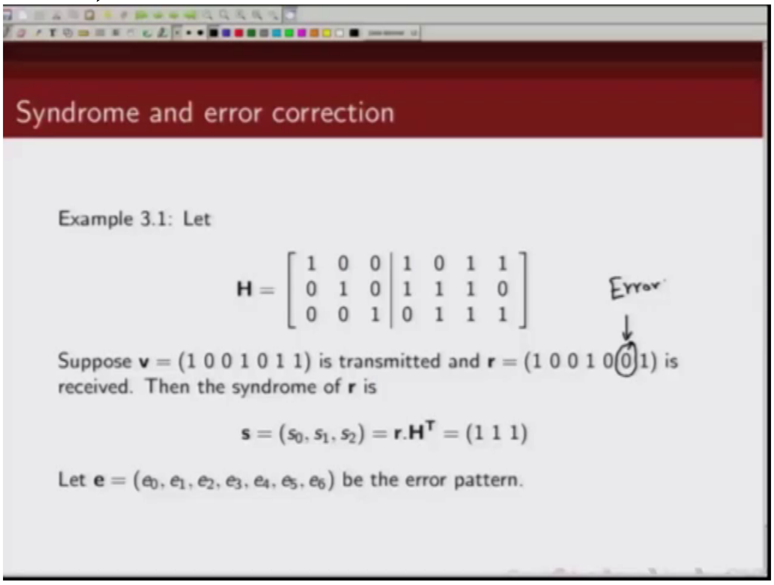
that there is an error and there is an error in this location? So first thing that we will do is we will compute the Syndrome. When we will compute

(Refer Slide Time 15:48)



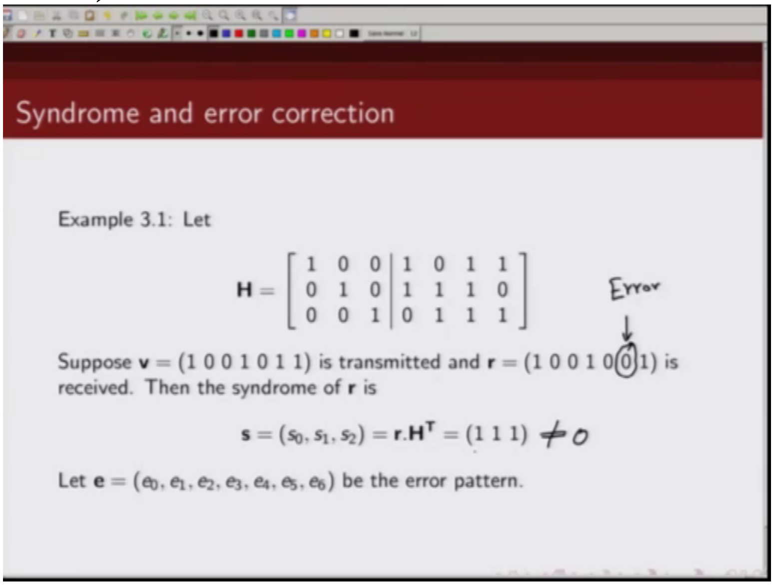
the Syndrome which is r H transpose what we get is a

(Refer Slide Time 15:53)



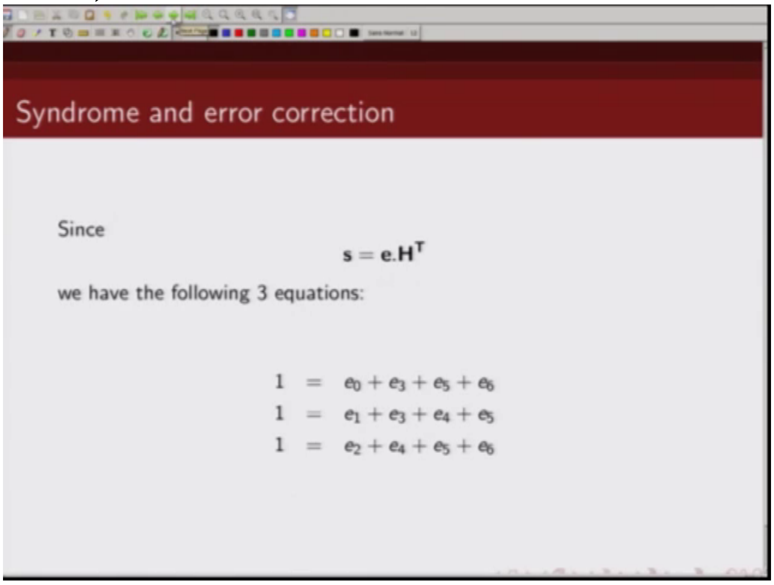
non-zero and since this is non-zero this means

(Refer Slide Time 15:59)



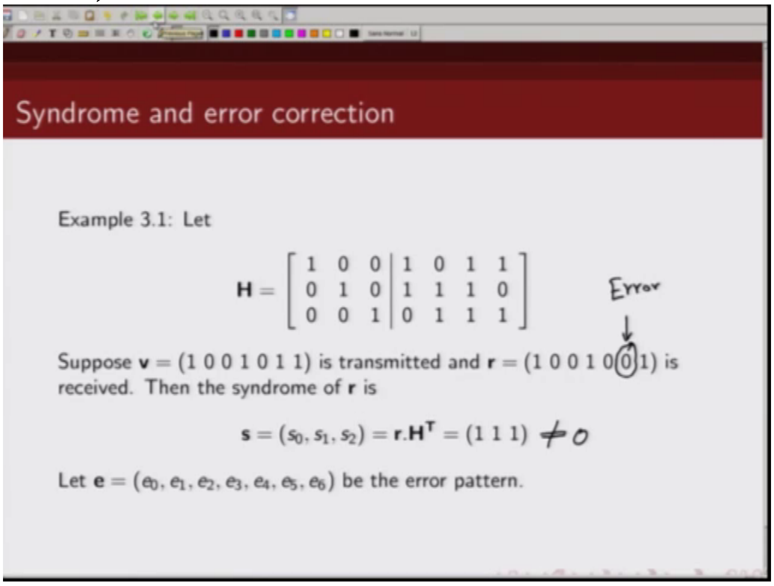
there is an error. Now next step is to find out where the error has occurred. So in this case

(Refer Slide Time 16:11)



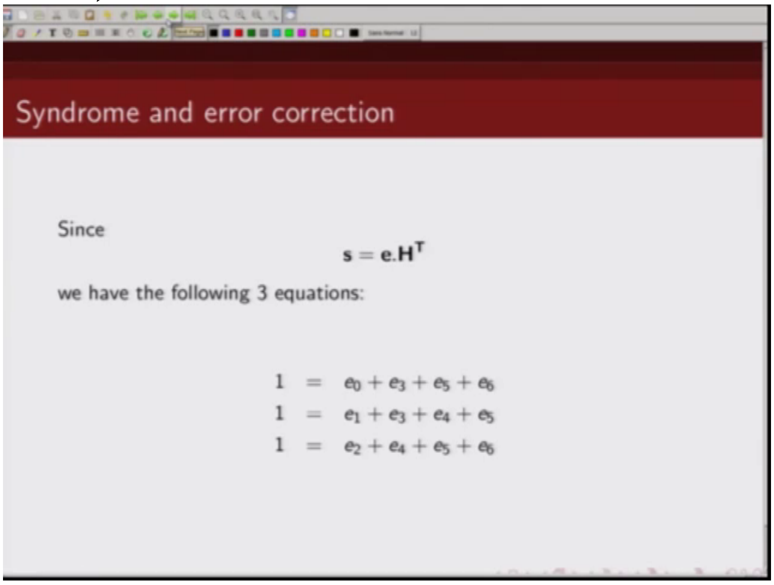
we can write the Syndrome in terms of error bits. So we have 3 equations and we have total 7 unknowns, right? And these equations are basically given by this. We have our

(Refer Slide Time 16:31)



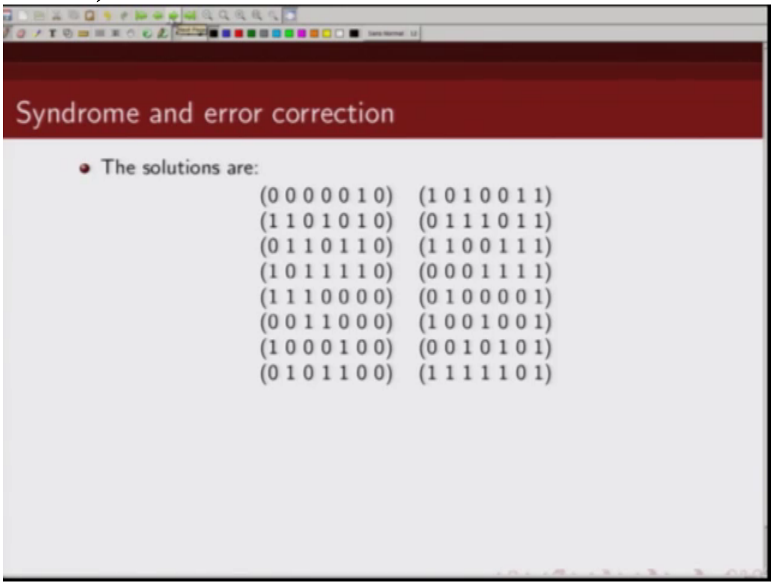
H matrix given by this This is our error pattern, so when we compute

(Refer Slide Time 16:37)



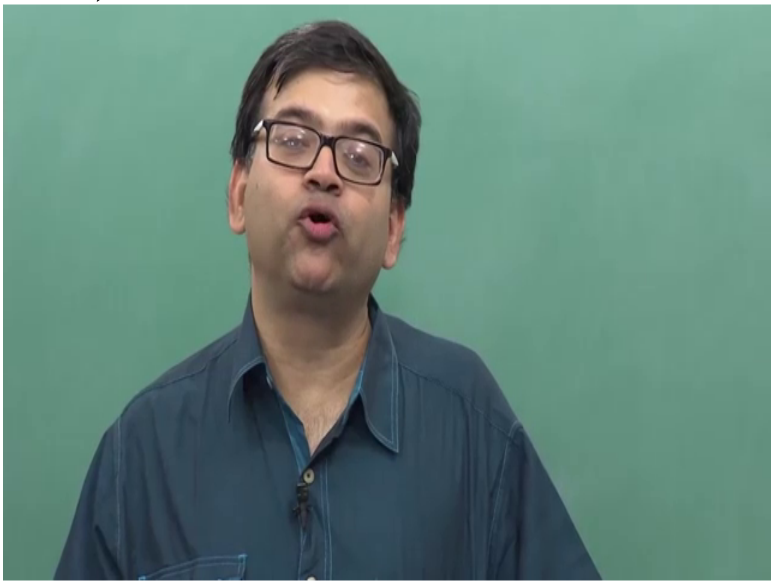
e H transpose we get set of these 3 equations. And

(Refer Slide Time 16:43)



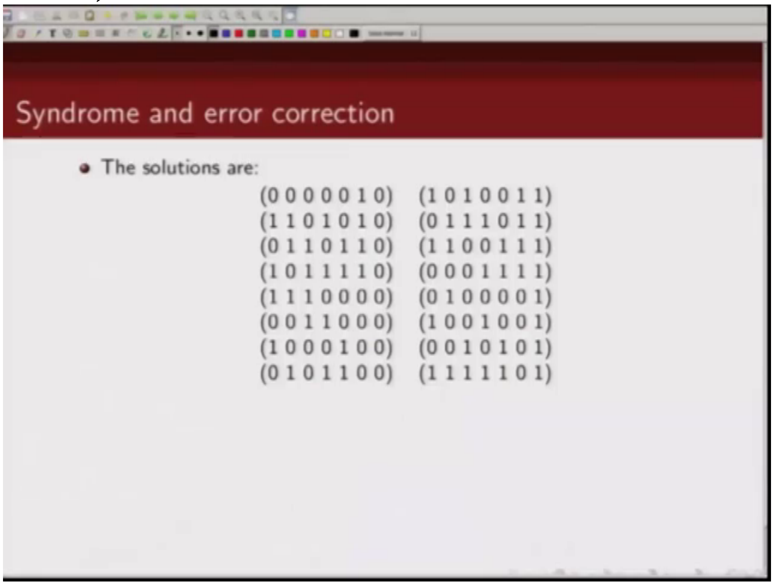
there are 16 solutions to this set of equations. This set of 3 equations where there are 7 unknowns; there are total 2 to power 4 different solutions. These are the 16 different solutions and as we said, the maximum likelihood decoding rule

(Refer Slide Time 17:07)



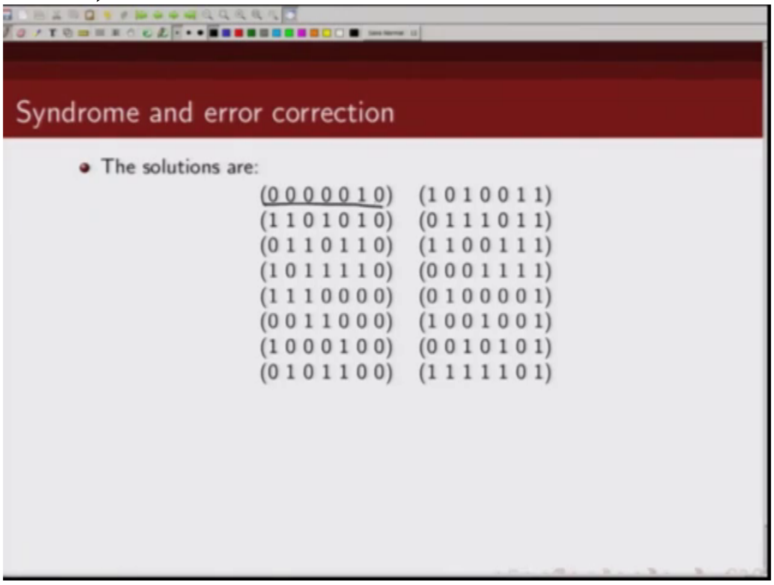
for a binary symmetric channel will chose an error pattern that has minimum Hamming weight so, which has minimum

(Refer Slide Time 17:15)



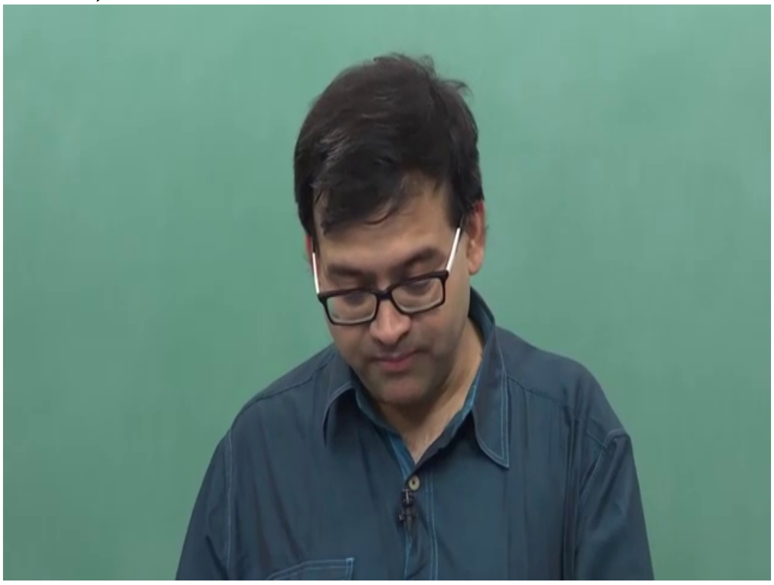
number of 1s So you can see among these 16

(Refer Slide Time 17:19)



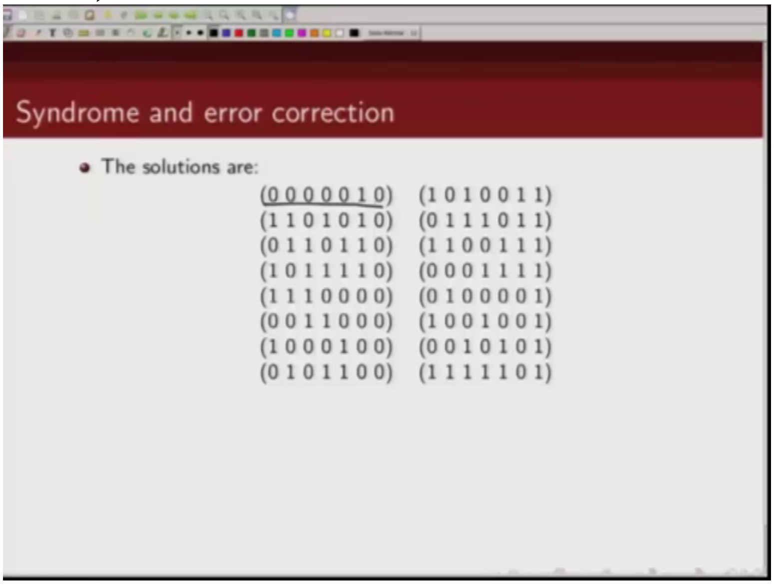
solutions the one that has minimum Hamming weight, minimum number of 1s is this. All others have, this has four 1's, this has four 1's, this has five 1's, three 1's, two 1's, you can check basically this one has the least number of 1's. So this

(Refer Slide Time 17:39)



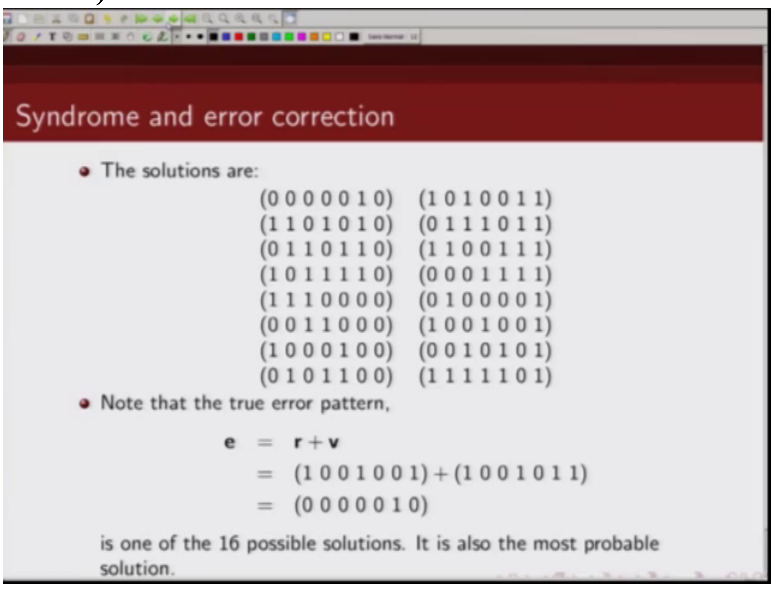
error pattern has the least of Hamming weight. So this is out of

(Refer Slide Time 17:44)



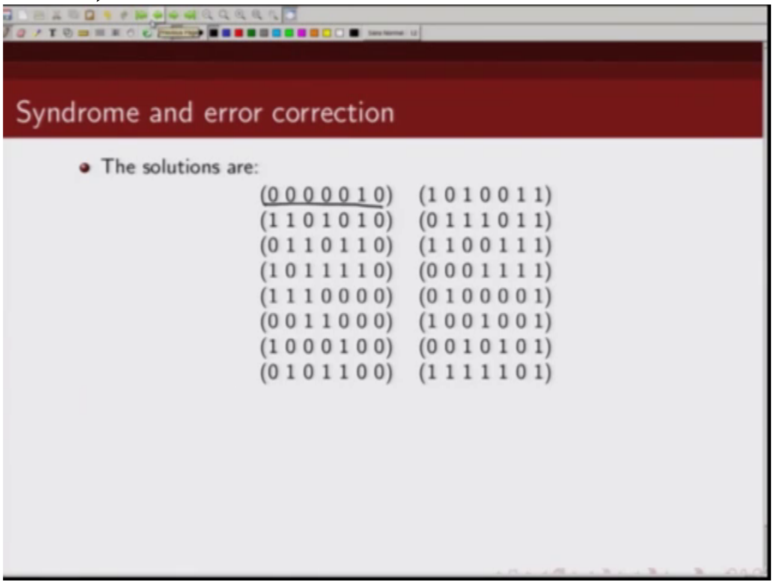
those 2 to the power 4 solutions, this is the most likely solution. And this we can verify also

(Refer Slide Time 17:52)



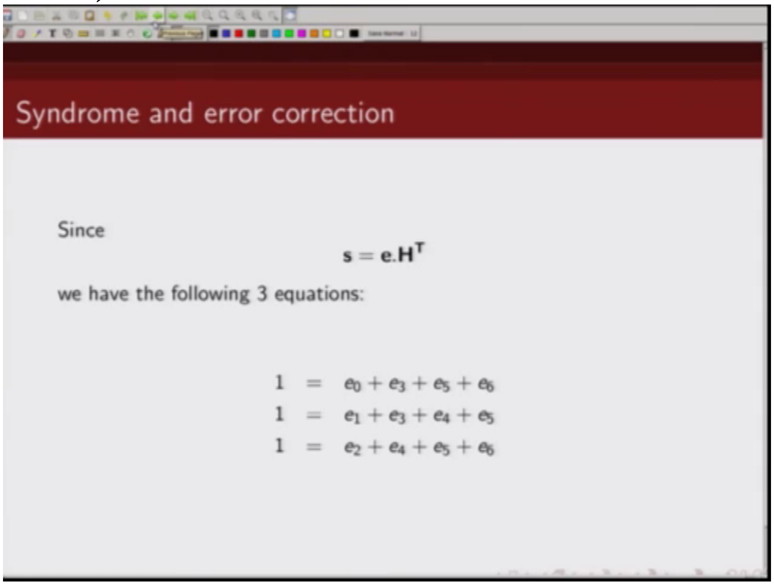
because we were given the received sequence and we knew what was the transmitted sequence was error pattern indeed was this, which we found out from, by solving these set of equations. So to recap then, now when we want to do error correction, what do we need to do, we need to solve

(Refer Slide Time 18:22)



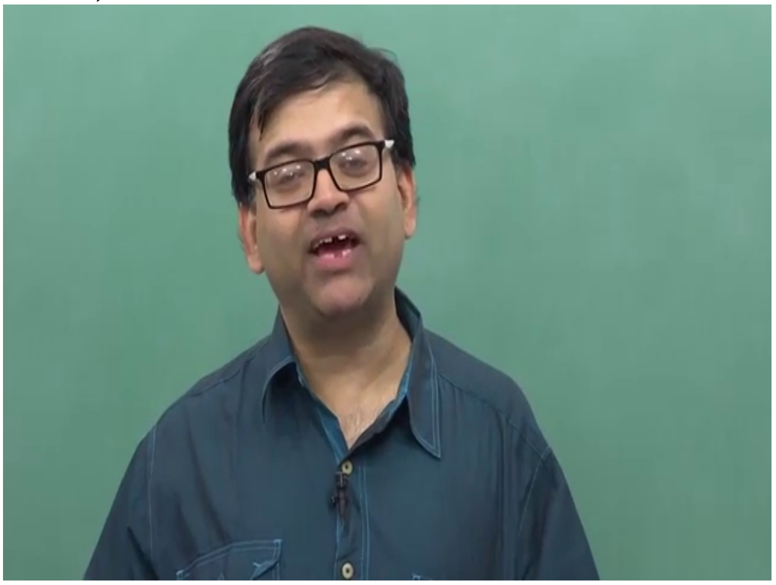
for

(Refer Slide Time 18:25)



this Syndrome equations And there are n minus k equations

(Refer Slide Time 18:30)



but there will be n unknowns so this will have 2 k solutions and we will have to pick the most probable solution from these set of 2 k solutions. The next lecture we are going to talk about a general decoding algorithm for a linear block code, thank you.