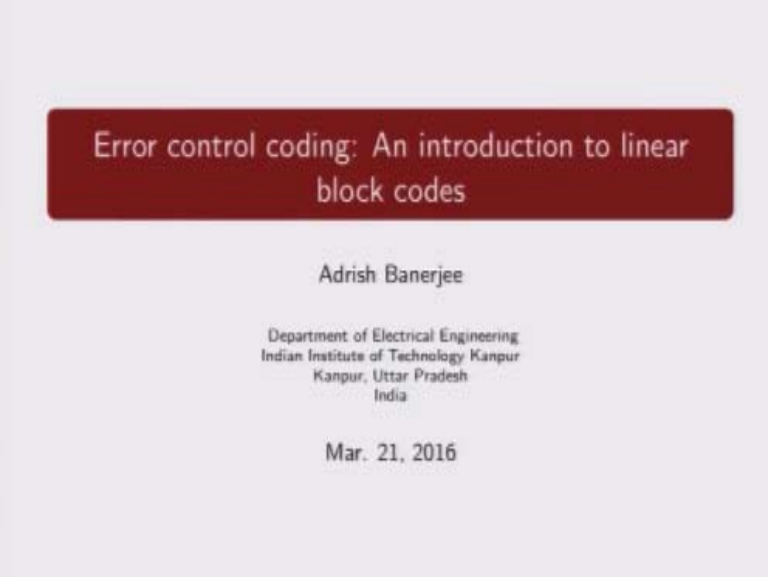
Indian Institute of Technology Kanpur
National Programme on Technology Enhanced Learning (NPTEL)
Course Title
Error Control Coding: An Introduction to Linear Block Codes

Lecture – 5B
Distance Properties of Linear Block Codes – II

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

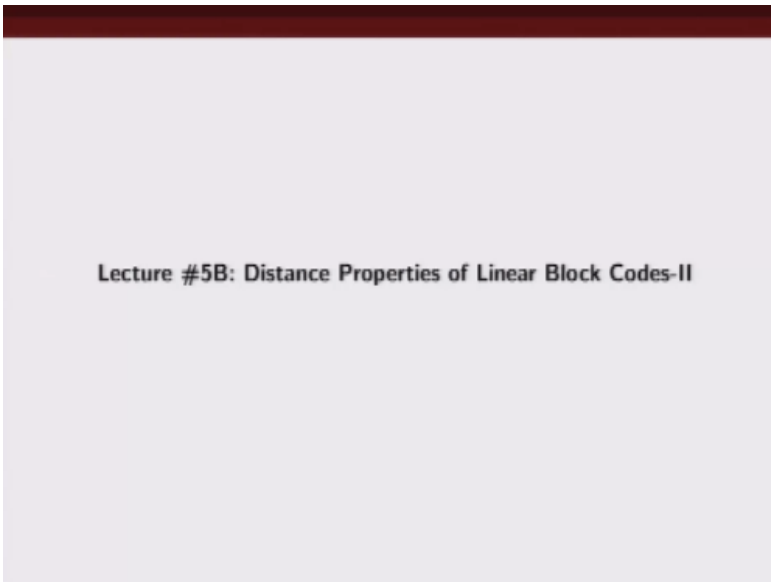
Welcome to the course on error control coding an introduction to linear block codes.

(Refer Slide time: 00:20)



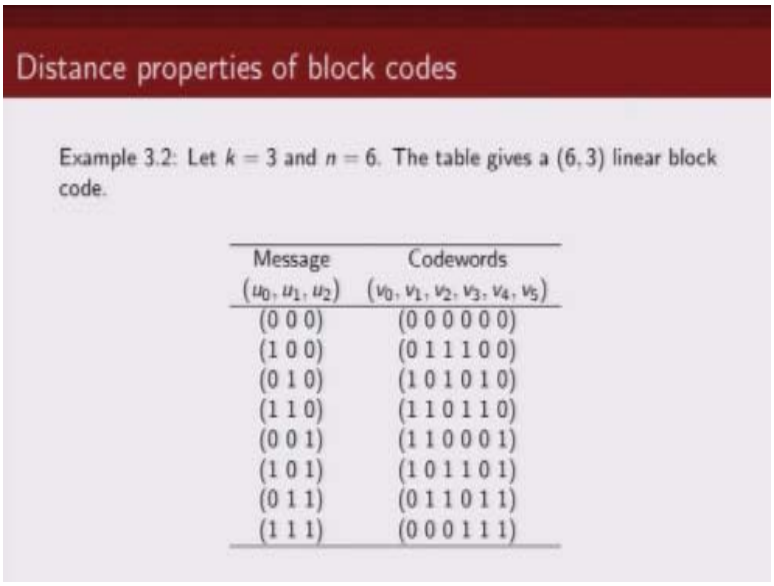
In this lecture we are going to talk about what do we mean by weight distribution of a linear block code and then we are going to talk about how is the error correcting capability and error detecting capability of a linear block code dependant on the minimum distance of a code.

(Refer Slide time: 00:45)



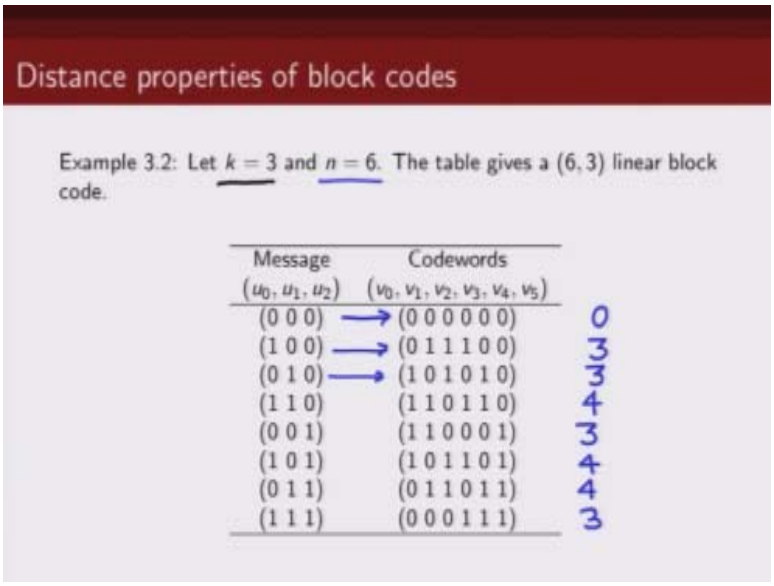
So we will continue basically our discussion on distance properties that we started last time.

(Refer Slide time: 00:49)



So this is one example of a linear block code where number of information bits is three.

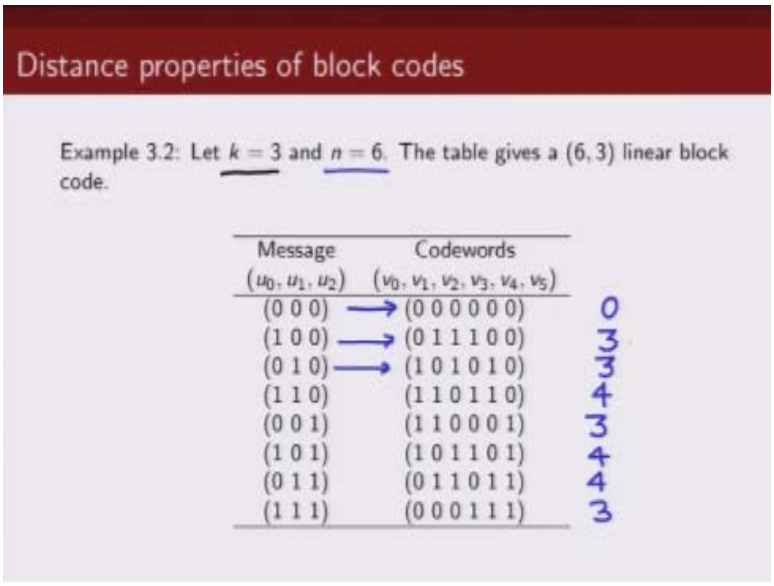
(Refer Slide time: 00:57)



And number of coded bits is six, this is a list of 2k code words which is eight code words, a message bits and these are their corresponding code words. So these are the from 000 to 111 these are our 2k message bits and corresponding to each of these message bits these are the corresponding code words, okay. Now let us look at what is the weight distribution of these code words, so this code word this is all zero code word so the weight, hamming weight for this is basically zero, what about this?

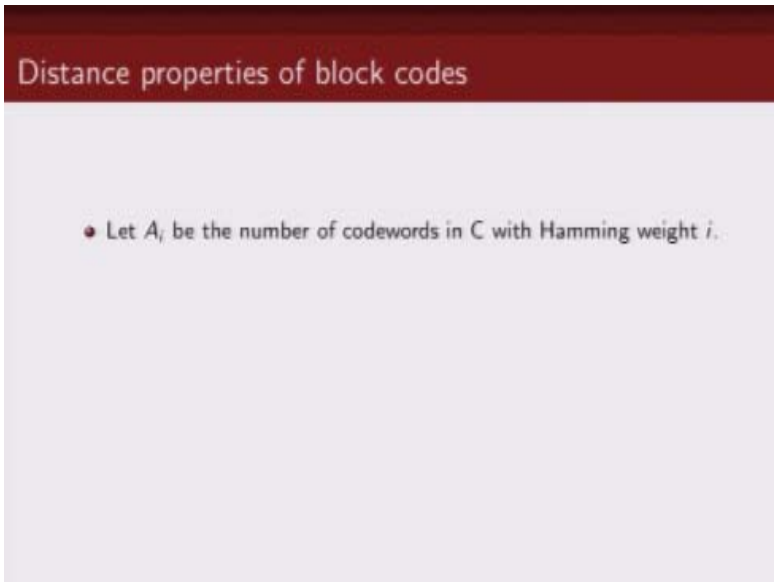
This code word has three ones so hamming weight is three, this code word has three ones so its hamming weight is three, this code word has four ones so the hamming weight is four, this code word has three ones so hamming weight is three, this one similarly has hamming weight four, this one hamming weight four and this one has hamming weight three. Now what is the minimum distance of the code? As you recall we define the minimum distance of a code as minimum weight of a non - zero code word.

(Refer Slide time: 02:41)



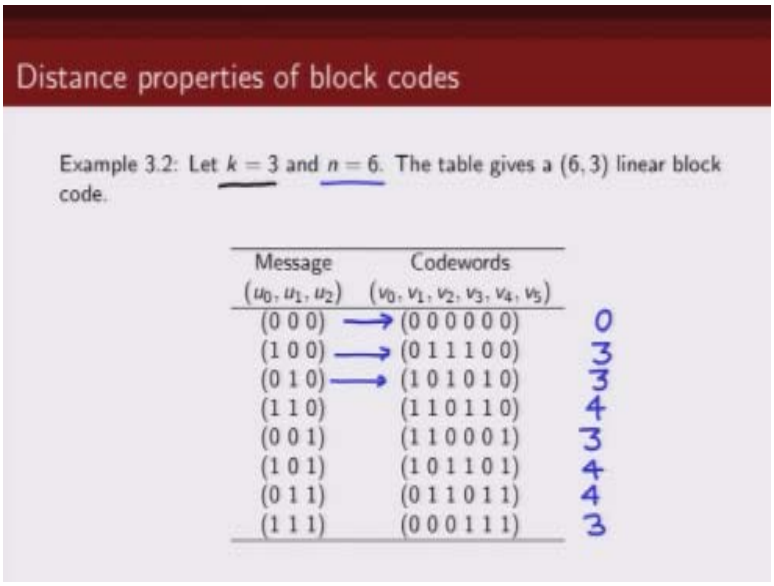
So what is the minimum weight of a non - zero code word in this case? It is three, so the minimum distance of this code is three.

(Refer Slide time: 02:56)



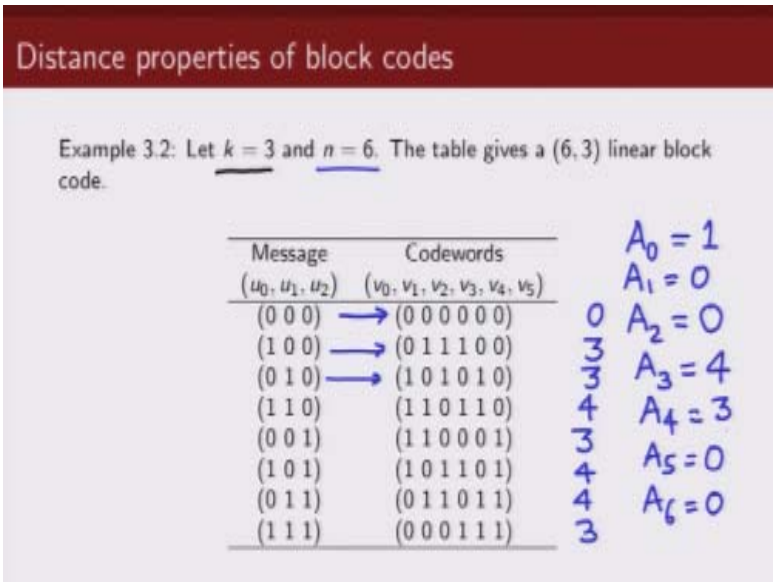
So let A_i denotes the number of code words in C with hamming weight i.

(Refer Slide time: 03:07)



So is you look here if i, so I will use A_0.

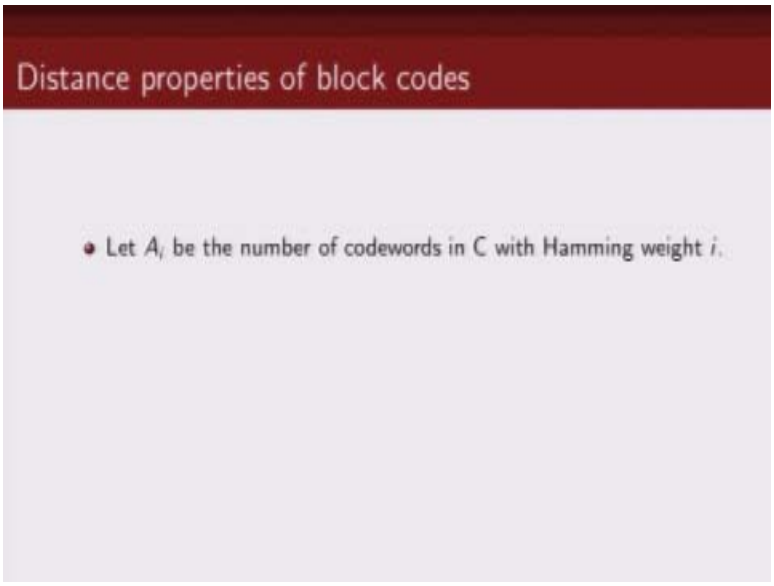
(Refer Slide time: 03:41)



To denote number of code words which have hamming weight zero and that number is 1, do we have any code word with hamming weight one? No, so A_1 is going to be 0, what about A_2? How many code words we have with hamming weight two, again that is zero, what about A_3?

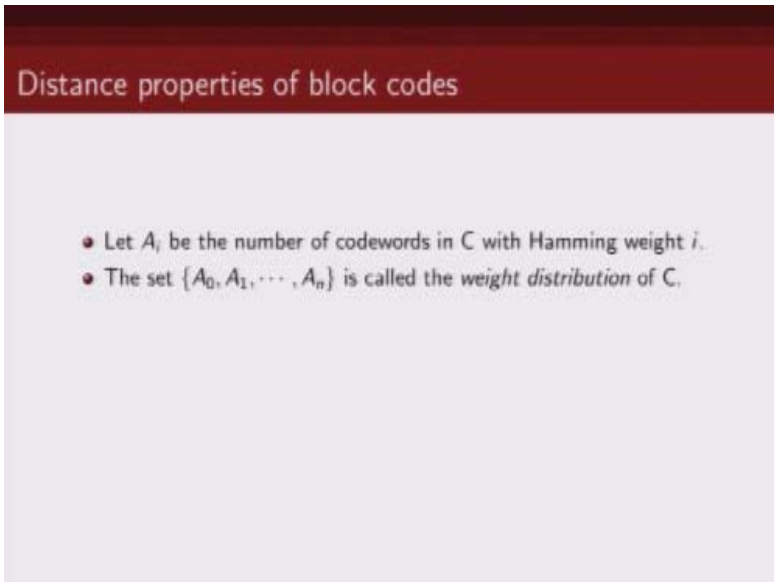
That is basically one, two, three, four, you have four code words with hamming weight three, A_4 one, two, three okay, we do not have any code with hamming weight five or hamming weight six and you can do a quick check, the number of code words should add up to number of code words that we have which is eight 1+4+3, okay.

(Refer Slide time: 04:29)



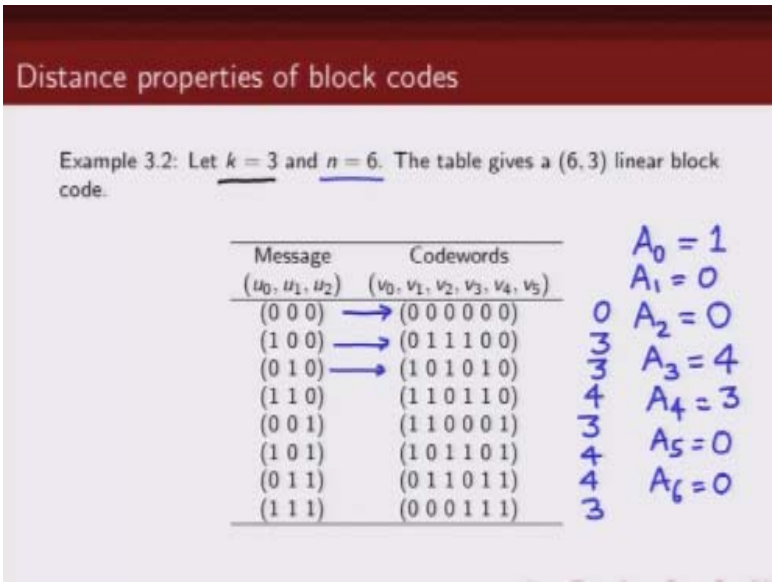
So we are denoting by A_i the number of code words in this linear block code with hamming weight i.

(Refer Slide time: 04:38)



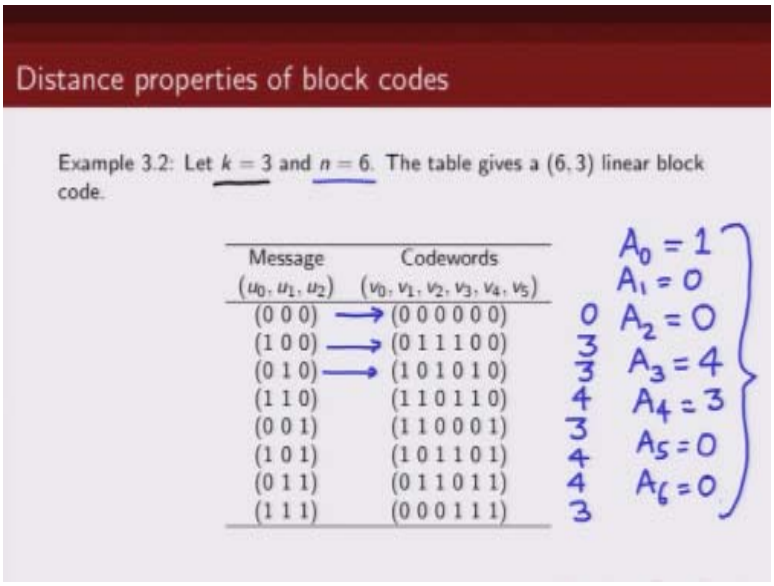
Now this set which describes how many code words we have of particular weight, this is basically known as weight distribution of a linear block code c.

(Refer Slide time: 04:55)



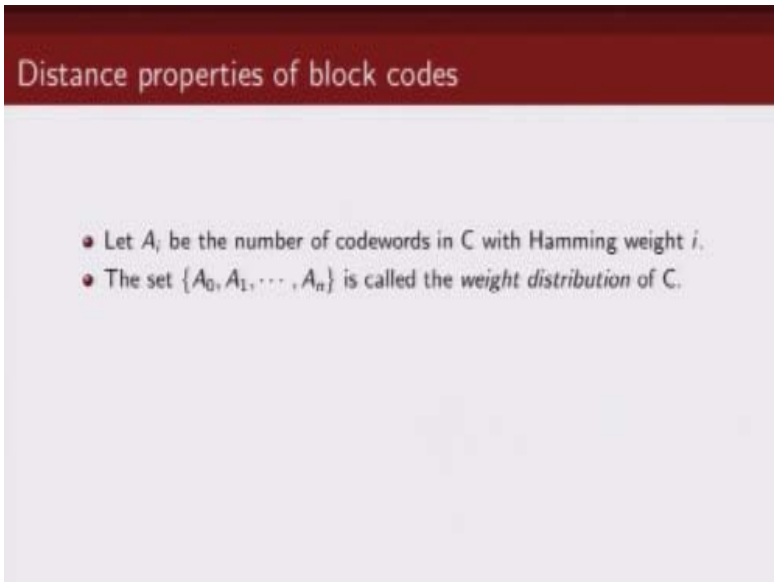
So for this block code the weight distribution is given by this.

(Refer Slide time: 05:01)

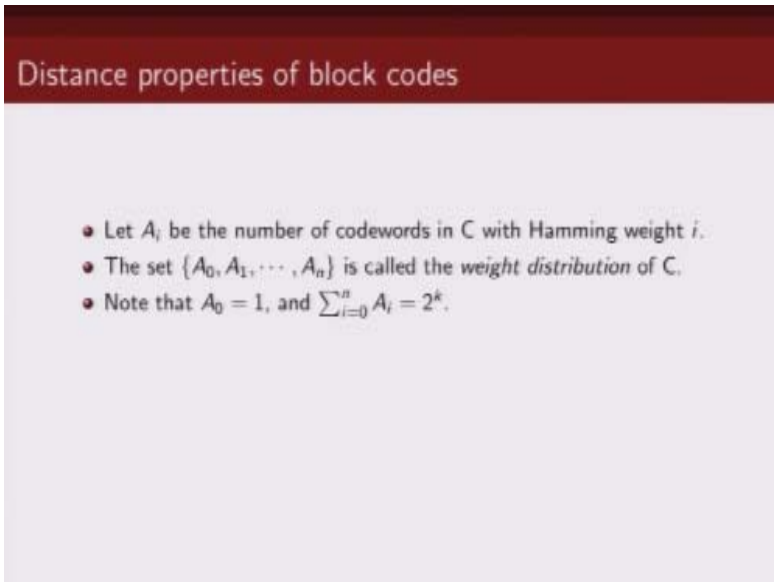


This completely specifies the weight distribution of this particular (6, 3) linear block code.

(Refer Slide time: 05:11)

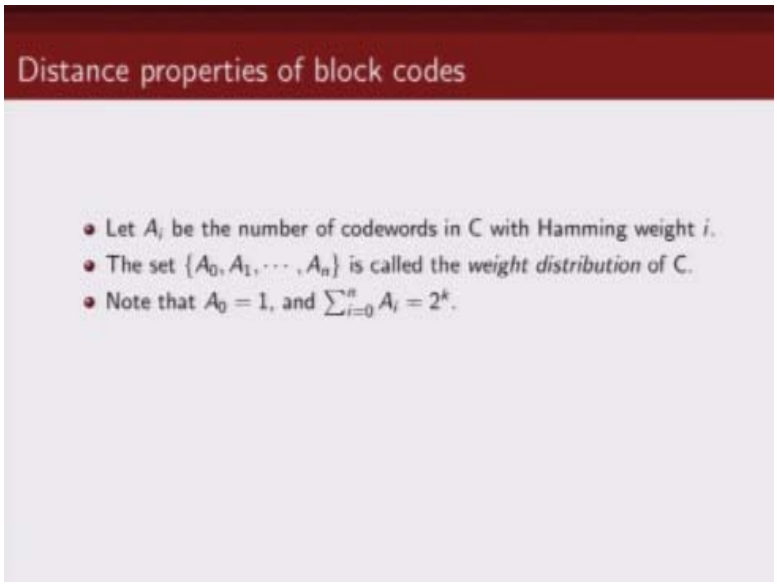


(Refer Slide time: 05:12)



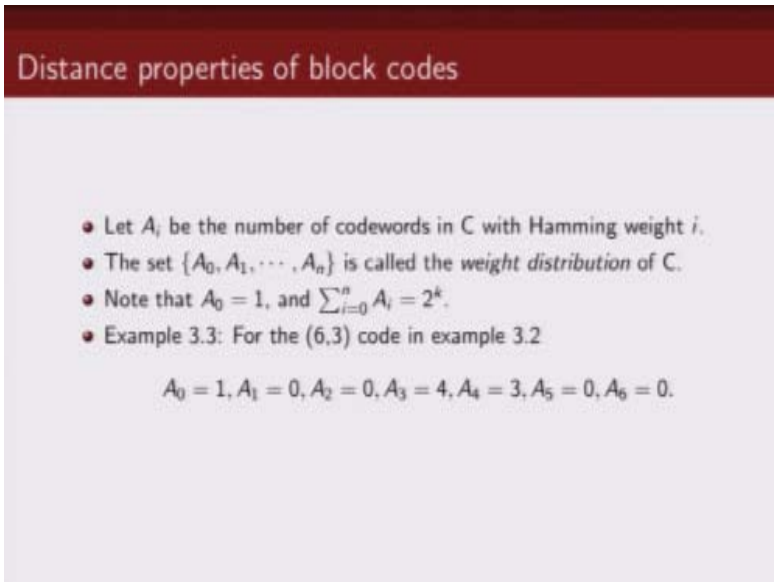
And since we have said a linear block code will have a non - zero code word.

(Refer Slide time: 05:17)



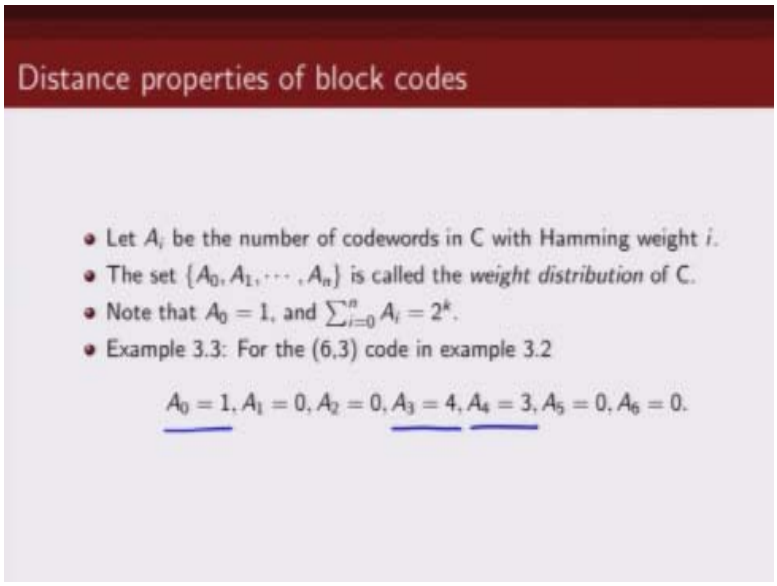
So A_0 will be one and sum of all these code words they should all add up to total number of code words which is 2^k.

(Refer Slide time: 05:29)



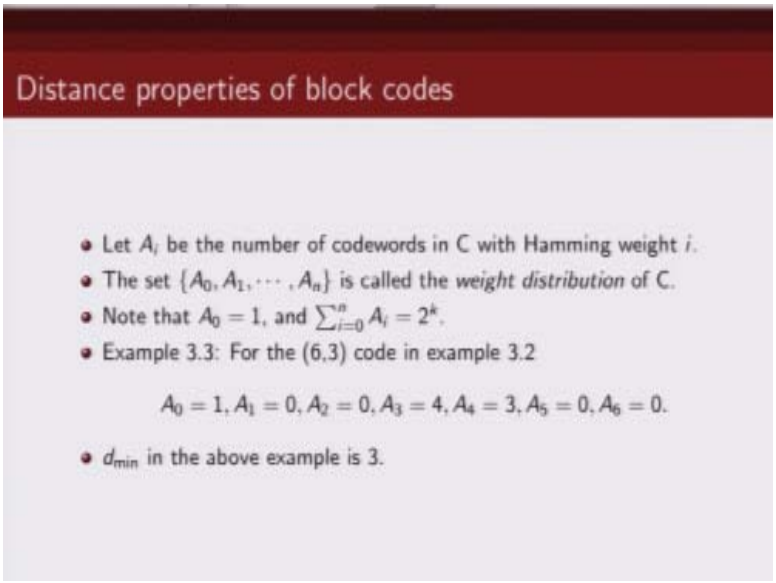
I just worked out this example for the (6, 3) code that we have shown in the previous slide and I showed you that in this particular example.

(Refer Slide time: 05:40)



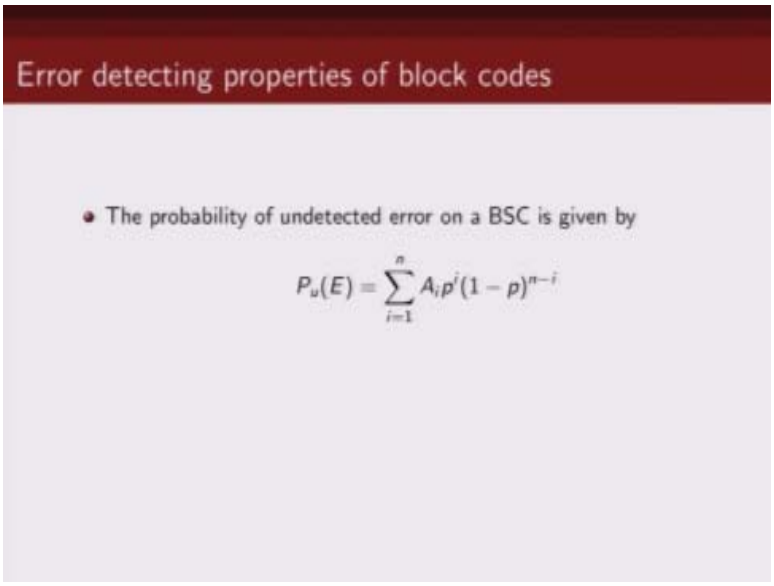
A_0 is 1, A_3 is 4, A_4 is 3, rest all others are zero.

(Refer Slide time: 05:48)



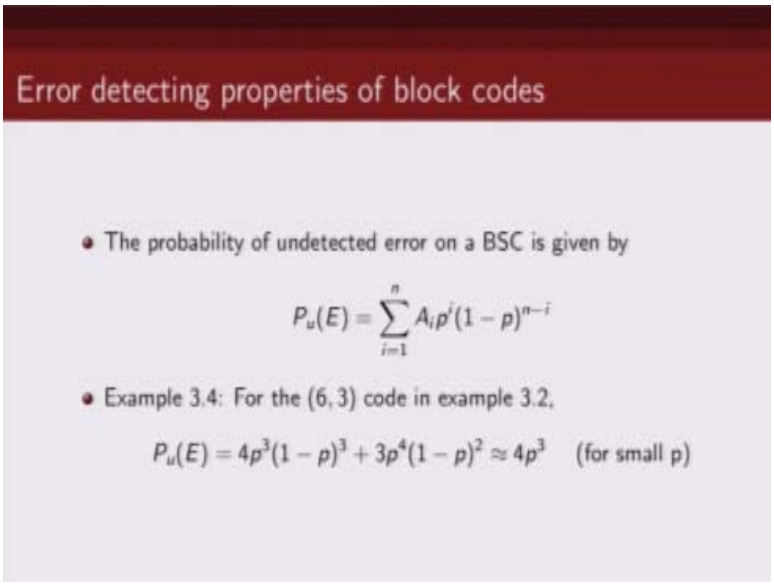
And I also showed you that the minimum distance of this code is three because minimum weight of a non – zero code word in this example is three.

(Refer Slide time: 06:03)



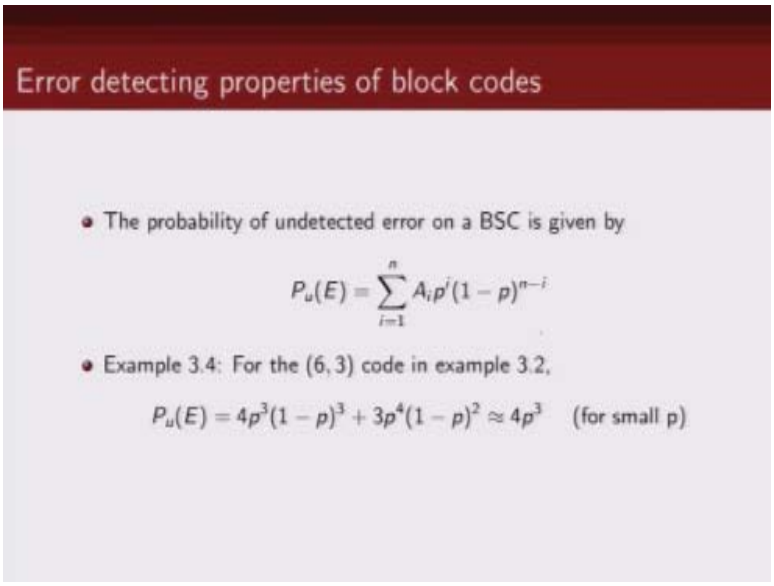
Now the probability of undetected error for a linear block code over a binary symmetric channel is basically related to the weight distribution of the code.

(Refer Slide time: 06:16)



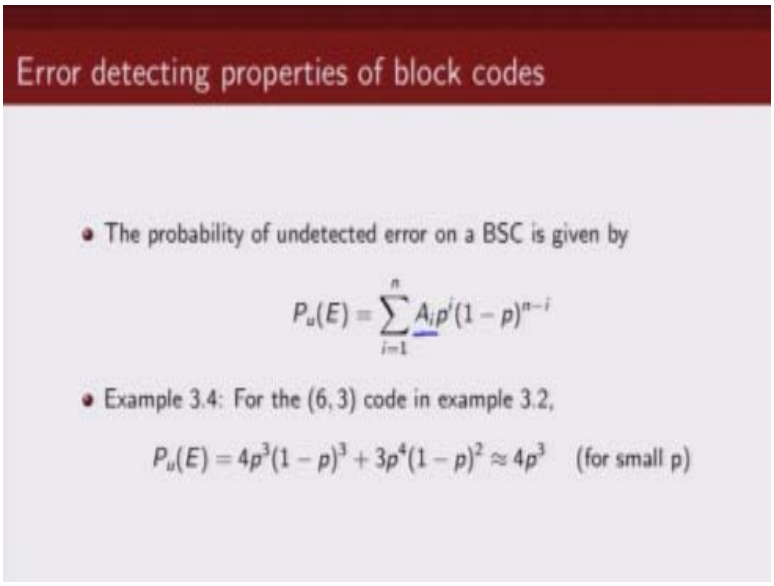
So for a (6, 3) linear block code and so when does an, when does a undetected error happens? An undetected error happens if let us say you send one particular code word and at the receiver you receive some other code word, so without loss of generality let us assume that we sent a all zero code word and at the receiver you received any other non – zero code word.

(Refer Slide time: 06:58)



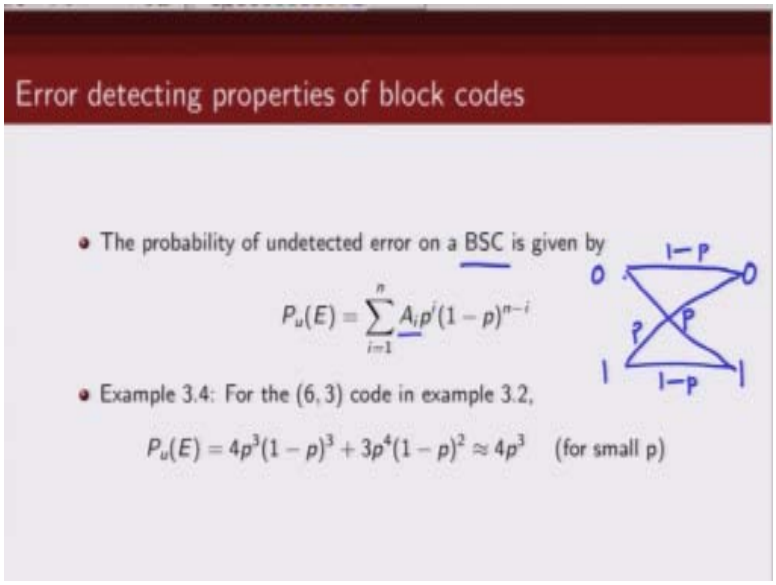
So if I send an all zero code word at the transmitter and at the receiver you receive any other non – zero code word then that will be the case of undetected error, so you can see basically.

(Refer Slide time: 07:15)



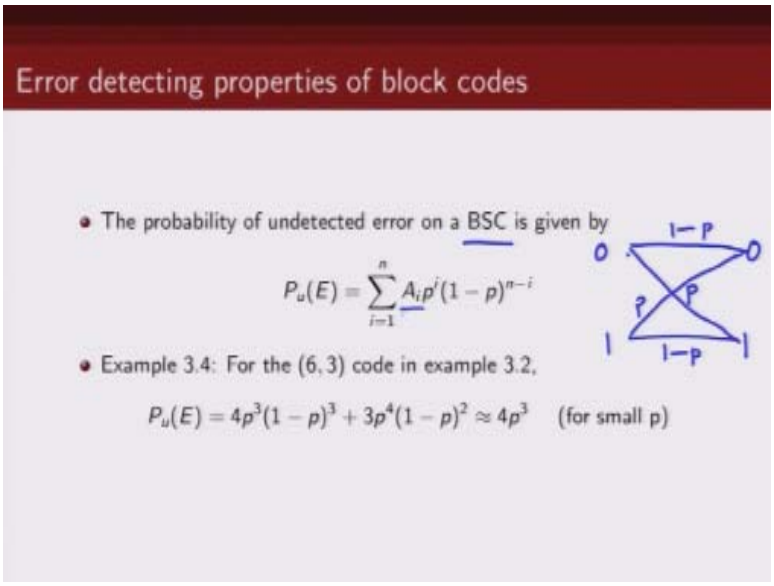
That is why I have written it as, so what is the probability of when you are sending an all zero code word what is the probability of getting another code word of weight A_i of weight i, what is the probability that when I am sending an all zero code word and you receive a code word which has weight i?

(Refer Slide time: 07:44)



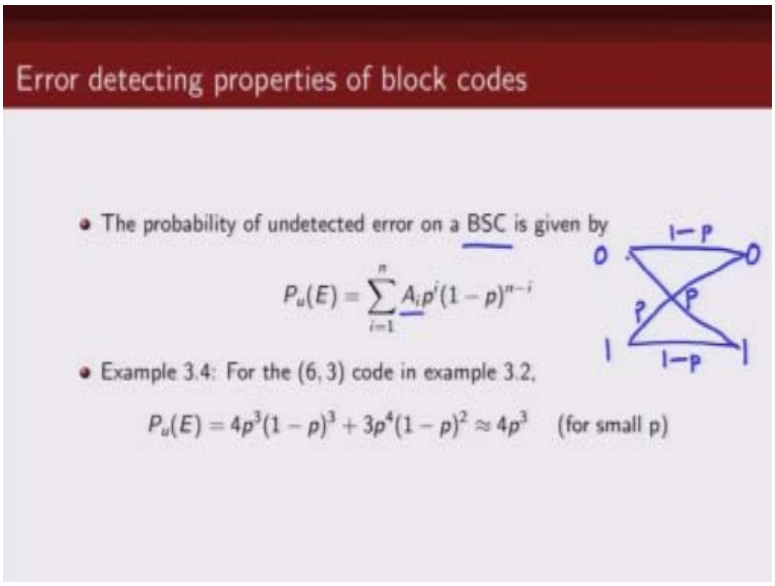
Now that probability is given by, since we are considering a binary symmetric channel now recall what happens in binary symmetric channel, two inputs 0 and 1, two outputs 0 and 1, and what is the crossover probability? That is basically given by P, so with probability P, 0 can get flipped to 1, 1 can get flipped to 0, and the probability of correct detection is 1-p so you are sending a code word which is an n bit tuple. Now what is the probability that you are sending an all zero code word which is of all zero bits?

(Refer Slide time: 08:34)



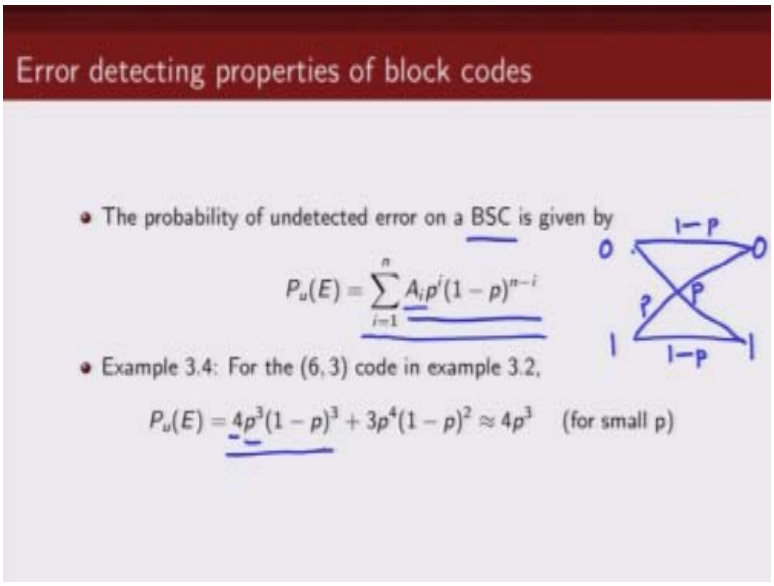
You receive another code word of weight i, now that probability is given by P^i this will happen when i bits get flipped and n-i bits do not get flipped, so that probability is given by $p^i (1-p)^{n-i}$ and how many such code words exist? That number is given by Ai so the probability of getting a weight i code word at the receiver when you send an all zero code word.

(Refer Slide time: 09:14)



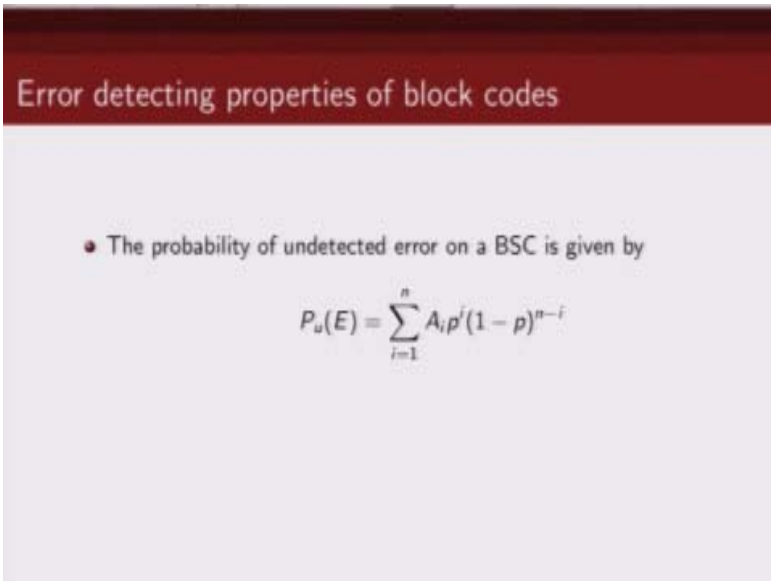
That probability is basically given by this okay, now an undetected error will happen if the receiver receives any non – zero code word.

(Refer Slide time: 09:28)



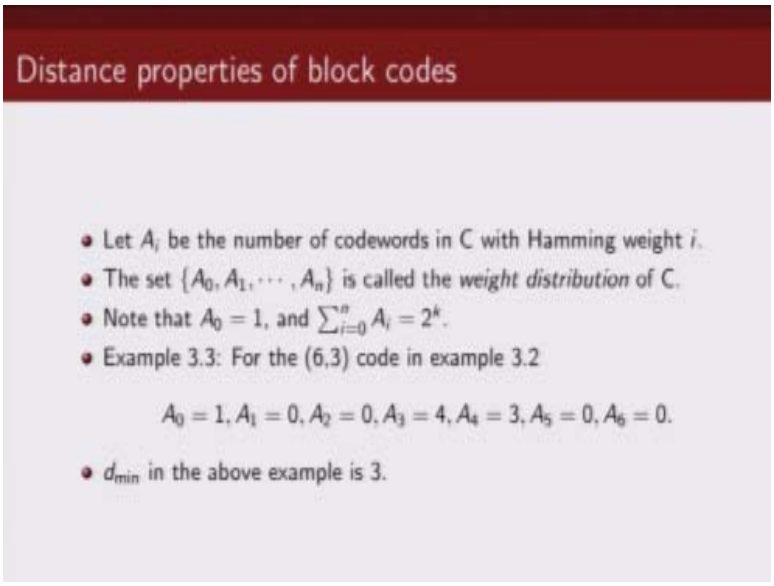
So I have to sum up this probability for all i going from 1 to n, so this is my overall undetected error probability if I send a linear block code over a binary symmetric channel, so for the example that I have considered I know the weight distribution, so if I plug that in here what I get is, so there were four code words with weight three so this is $4p^3$ and what was n, n is six so 6-i which is three in this case is three so first term that I will get is this.

(Refer Slide time: 10:22)

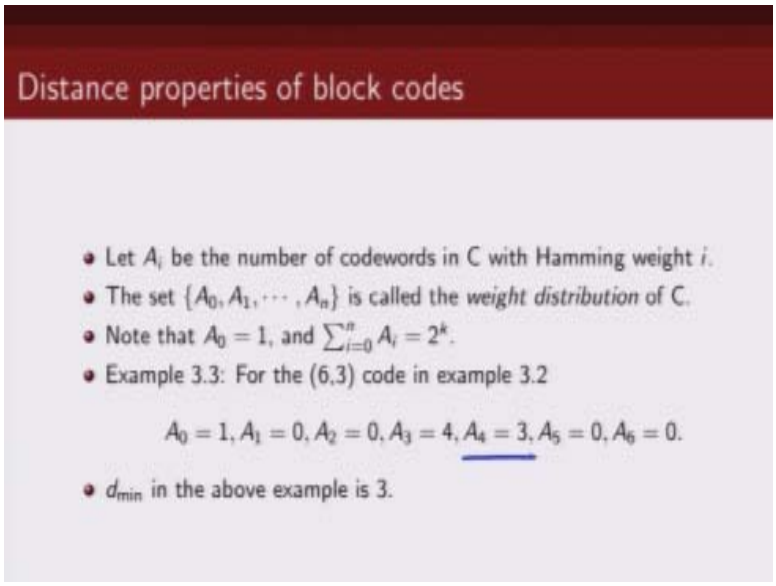


The next term corresponding to these code words is given.

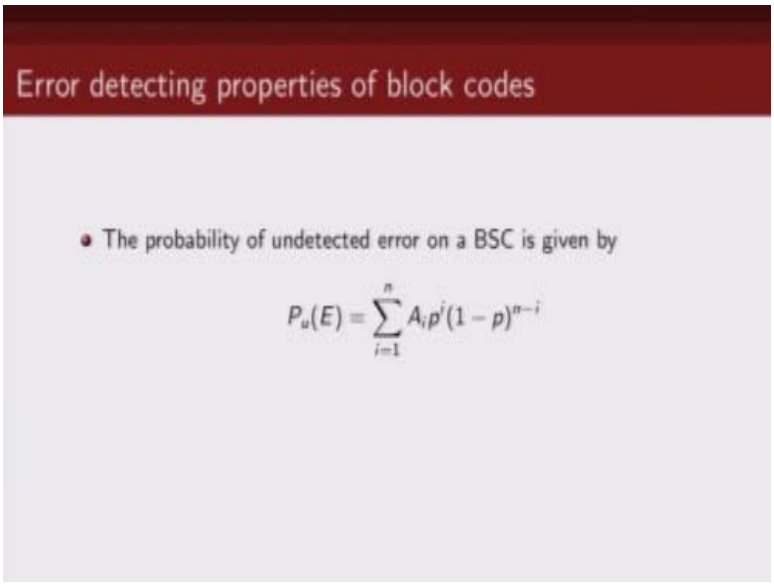
(Refer Slide time: 10:23)



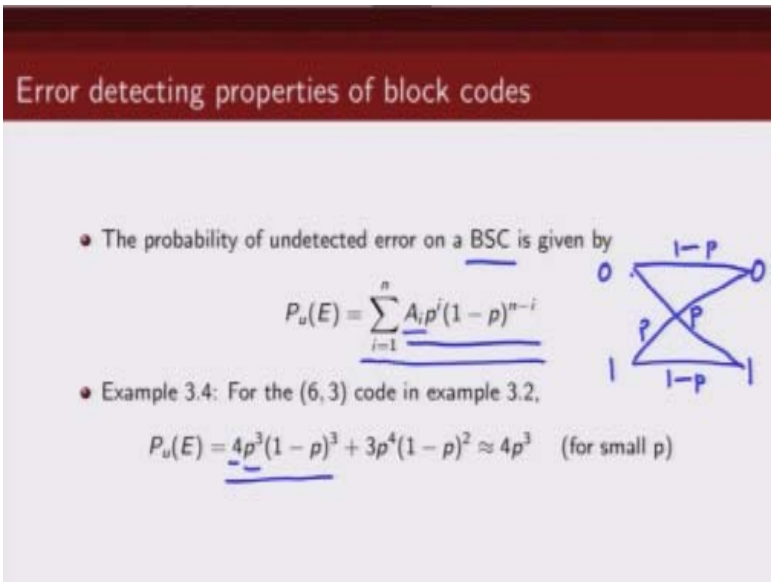
(Refer Slide time: 10:25)



(Refer Slide time: 10:29)

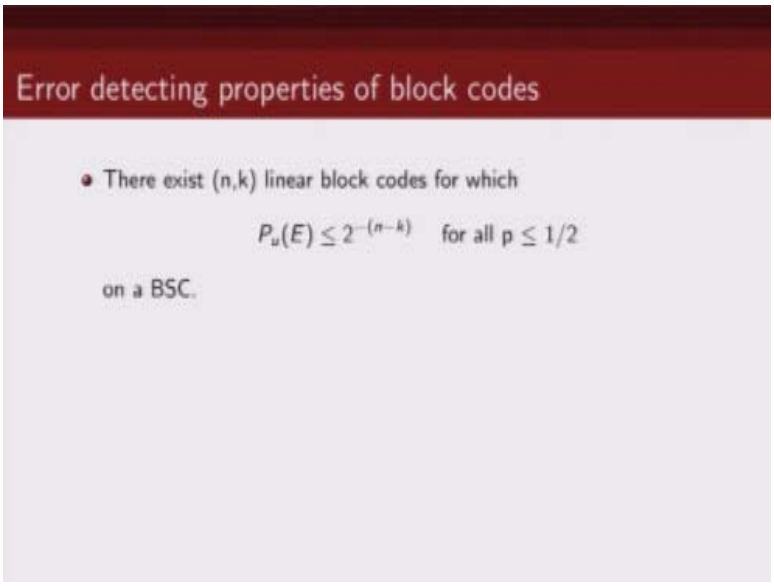


(Refer Slide time: 10:31)



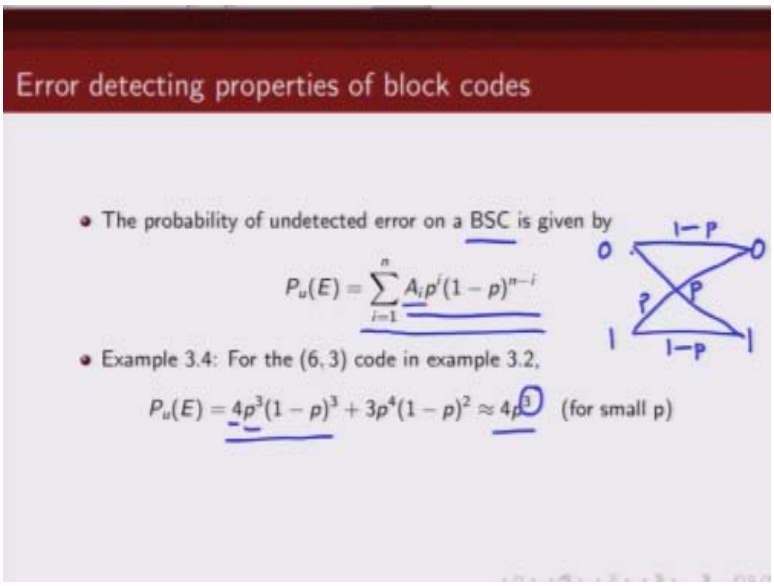
So there are three code words of weight four probability of four bits getting flipped is p^4 and probability of the other two bits not getting flipped is $(1-p)^2$ and since p is typically small I mean I can approximate it for a small p, I can approximate this undetected error probability as $4p^3$ because this will be close to one and since p is small a small number p^4 will be a small number so this be roughly equal to $4p^3$ this is for the case when p is small.

(Refer Slide time: 11:11)



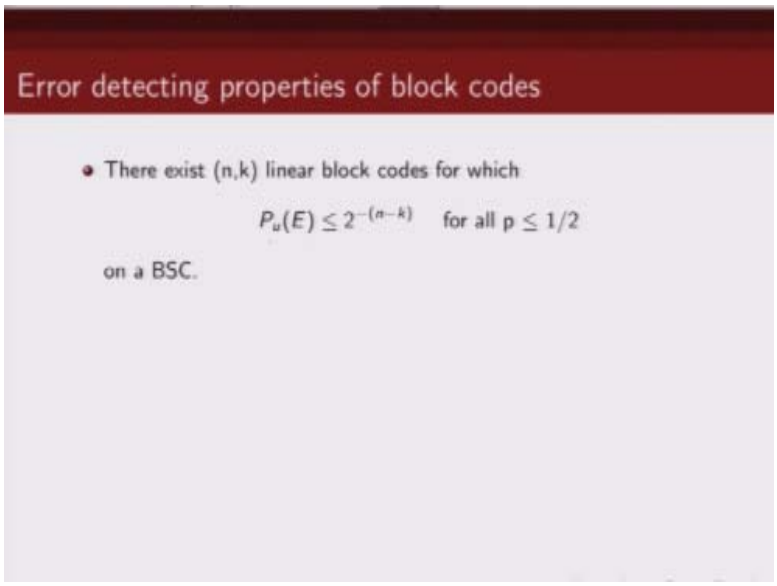
So you can see in general.

(Refer Slide time: 11:17)

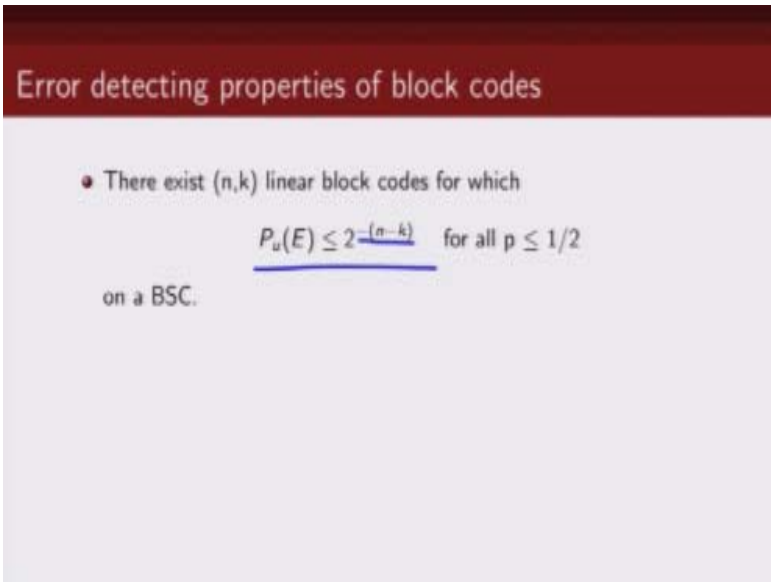


So in this particular example the undetected probability basically varies as p^3 which is basically same as n-k, in general we can show that.

(Refer Slide time: 11:28)

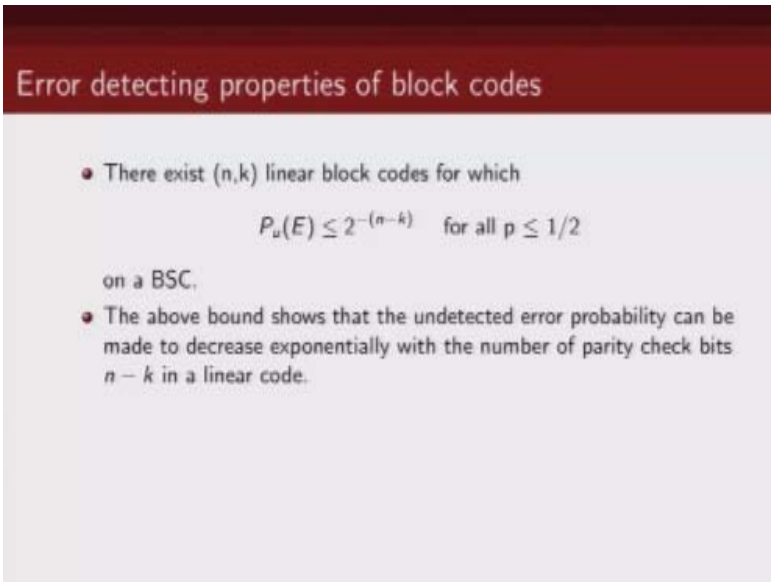


(Refer Slide time: 11:31)



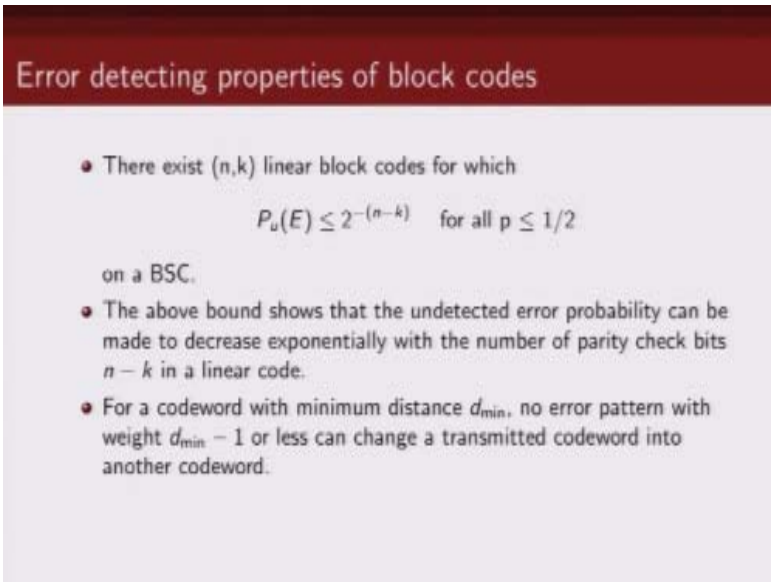
That undetected error probability is dependent on how many parity bits that we have, so more the number of parity bits lesser will be the undetected error probability.

(Refer Slide time: 11:43)



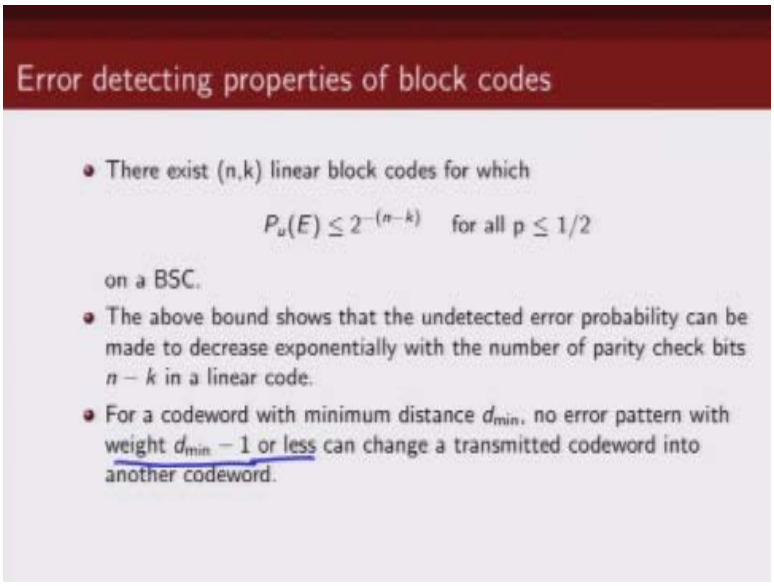
So we can make the undetected error probability go small by increasing the number of parity bits.

(Refer Slide time: 11:50)



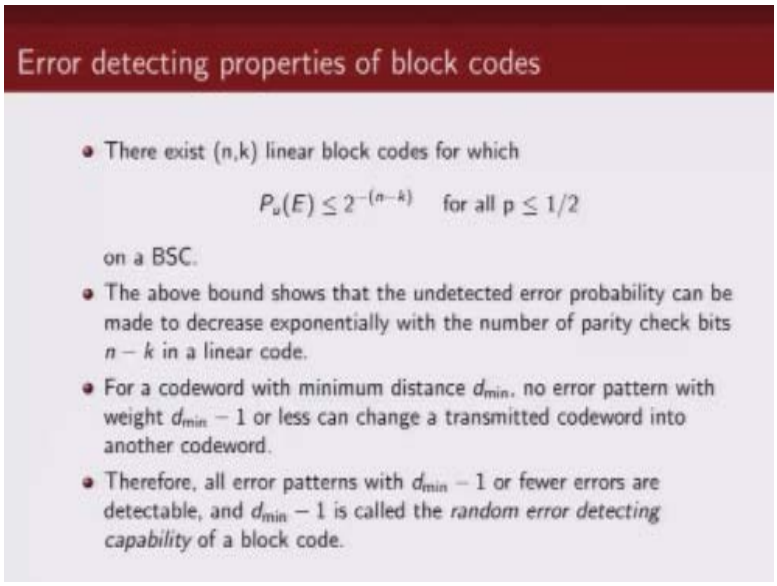
Now if we have a code word with minimum distance d_{min} we know that any error pattern, a weight less than equal to d_{min} -1 is not going to change that code word into any other valid code word, so in other words if there is an error pattern.

(Refer Slide time: 12:20)



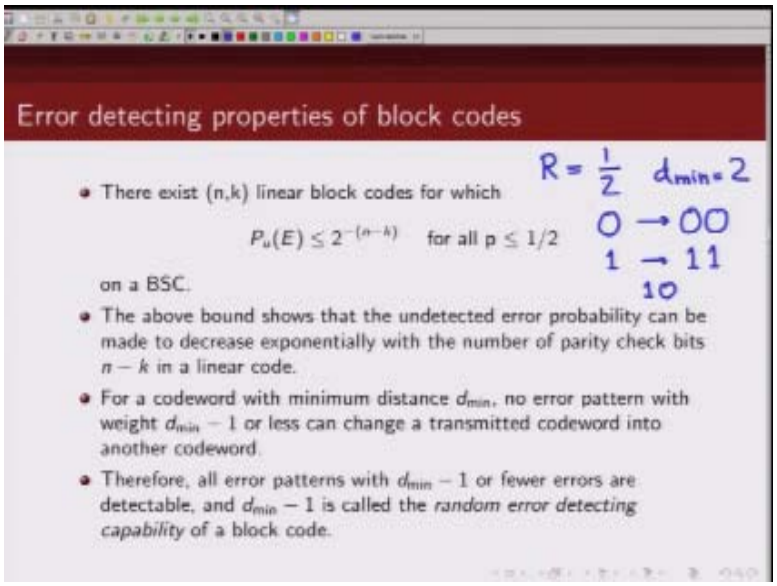
Of weight d_{min} -1 or less, then it cannot change a valid code word into another valid code word what does that mean? It means that we can actually detect any error pattern of weight upto d_{min} - 1.

(Refer Slide time: 12:48)



So all error patterns of weight d_{min} -1 or fewer errors are basically detectable and this is.

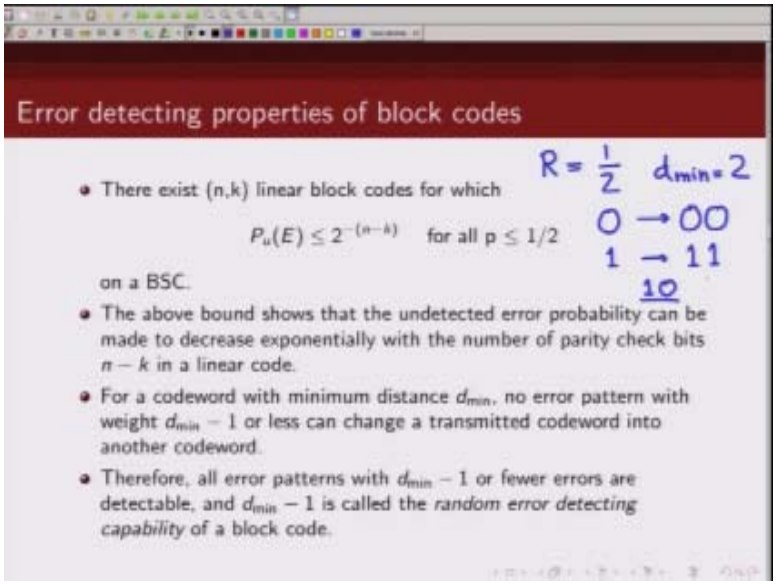
(Refer Slide Time: 13:00)



Also known as random error correcting capability of a linear block code, now take an example of a repetition code that we did in the first class so let us say we have a rate one 1/2 repetition code so then for 0 we are sending 00 and for 1 we are sending 11, now let us assume because of error in the channel some other bits got flipped so let us say this a what we received in when these what we – let us say what we received is 1 0 if you receive 1 0 can you detect, so what is the minimum distance of first answer this question?

What is the minimum distance of this code, this rate one ½ repetition code, we can see basically a minimum distance is 2, minimum distance of this code is 2 so according to this we should be able to detect all error patterns of weight 1, so let us take an example, let us say we receive 1 0, can you detect the error, yes we can because since it is a rate one ½ repetition code what we expect is we expect to receive either 00 or 11.

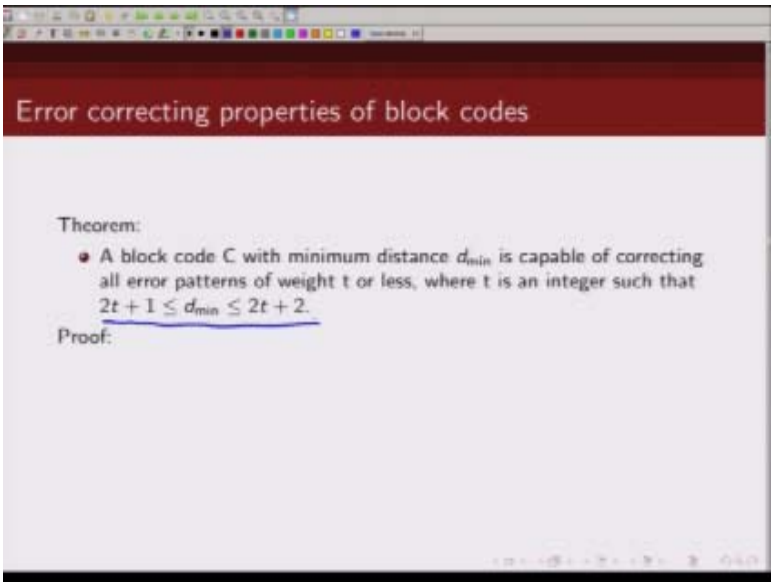
(Refer Slide Time: 14:07)



If we transmit these code word over a binary symmetric channel but what we have received is 1 0 which is neither 00 nor 11, so we are able to detect single error, so to repeat basically if you have a linear block code whose minimum distance is d_{min} you will be able to detect all error random errors of error pattern up to $d_{min} - 1$. Next we are going to show how is the error

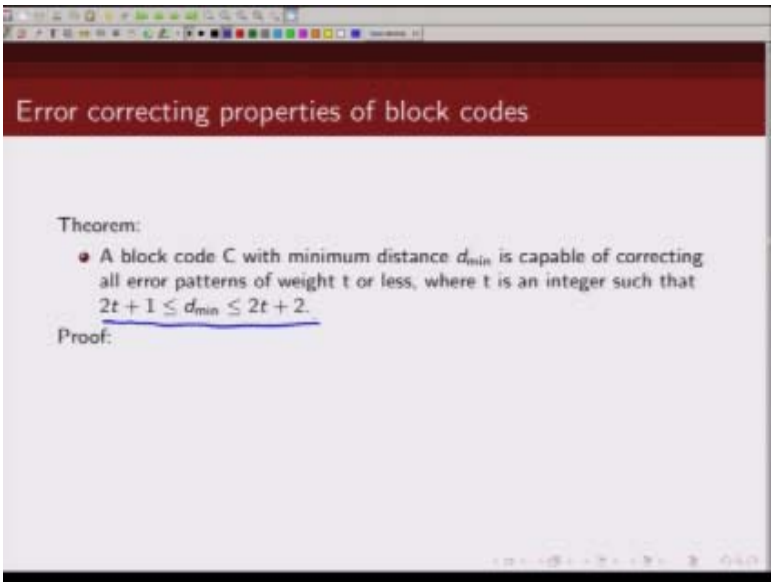
detecting capable, error correcting capability of a linear block code related to the minimum distance of a code.

(Refer Slide Time: 15:37)



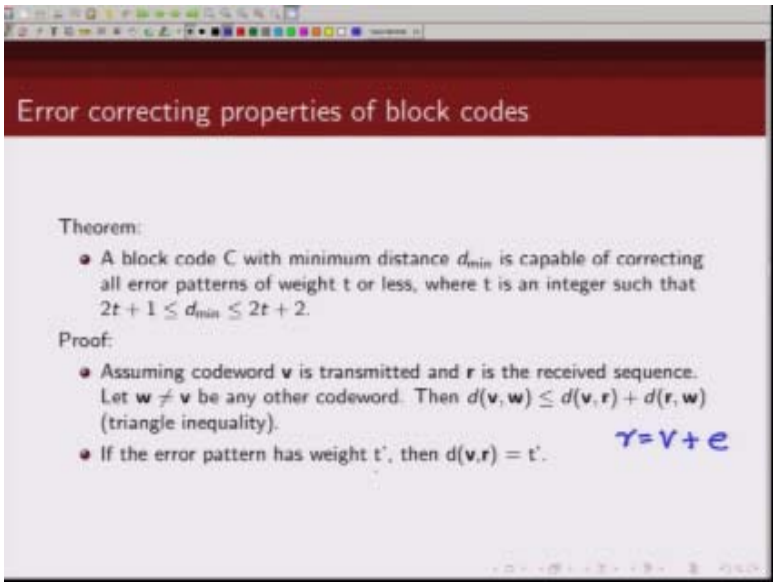
So if we have a linear block code C whose minimum distance is d_{min}, where d_{min} satisfies this relation d_{min} is > than equal to 2t + 1 where t is an integer and it is < than equal to 2t + 2. If d_{min} satisfies this relation and if we have a linear block code with minimum distance d_{min}.

(Refer Slide Time: 16:13)



Then it is capable of correcting all error patterns up to weight t, so let us prove this result.

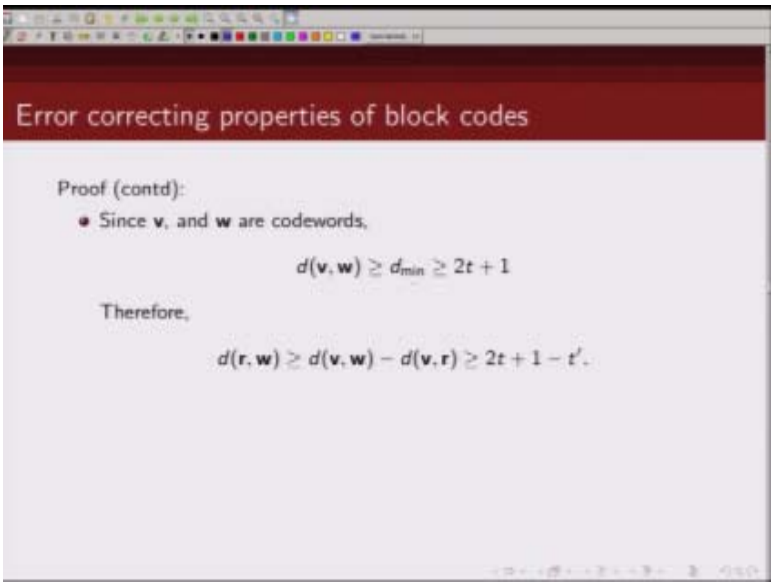
(Refer Slide Time: 16:20)



Let us assume the code word that is transmitted is given by v and what we received is this n-tuple r. Let us assume there is another code word w which is not same as v, now we know from triangular inequality that hamming distance between v and w will be < than equal to hamming distance between v and r + hamming distance between r and w. Now let us assume that the error pattern has weight t‘ and what is r, r is nothing but v+ this error pattern.

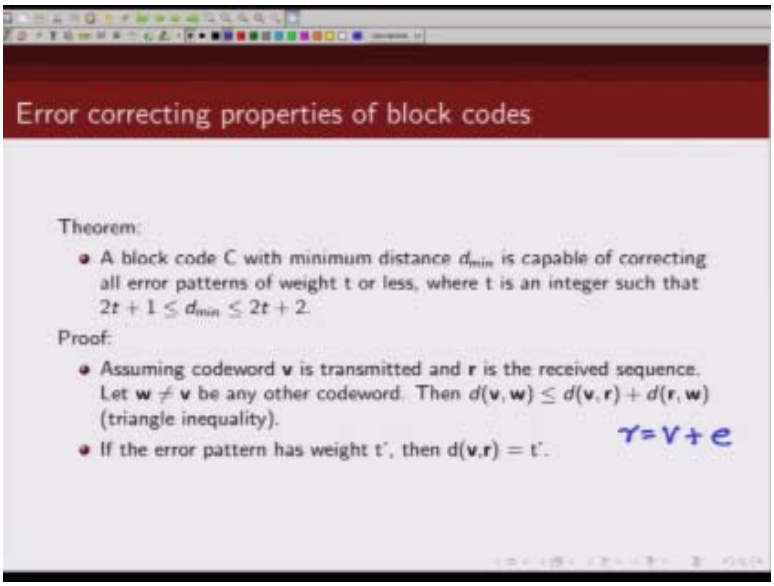
Correct, so the hamming distance between v and r is going to be the weight of this error pattern and which we are denoting by t -.

(Refer Slide Time: 17:34)



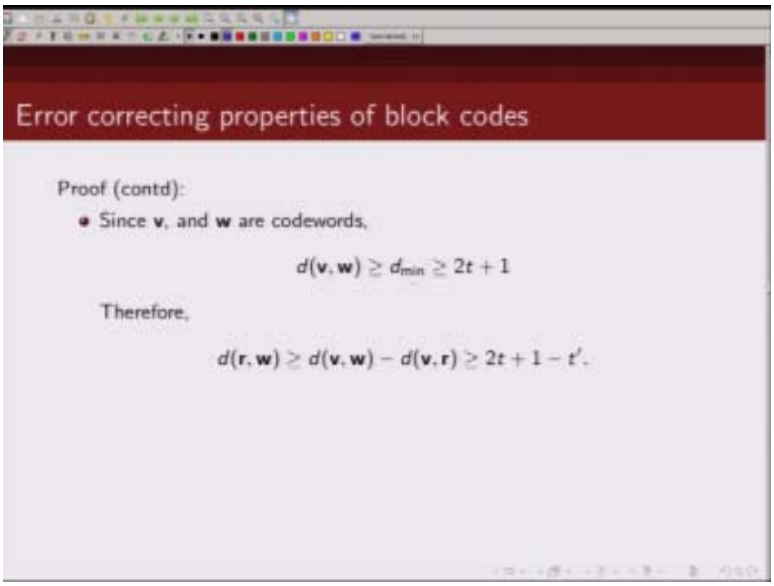
Now since v and w are valid code words so the hamming distance between v and w will be at least equal to minimum distance of the code, so the hamming distance between v and w is > than equal to minimum distance of the code and in the beginning we define.

(Refer Slide Time: 17:59)



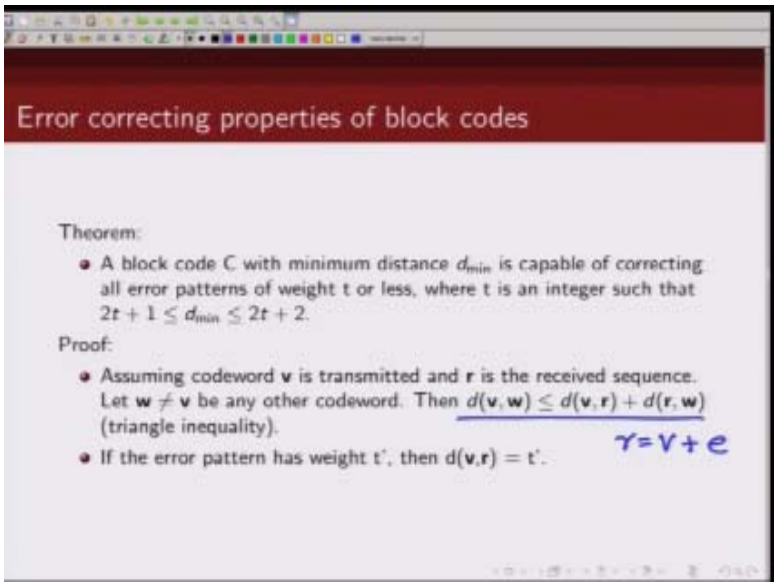
That our minimum distance is at least 2t + 1.

(Refer Slide Time: 18:04)



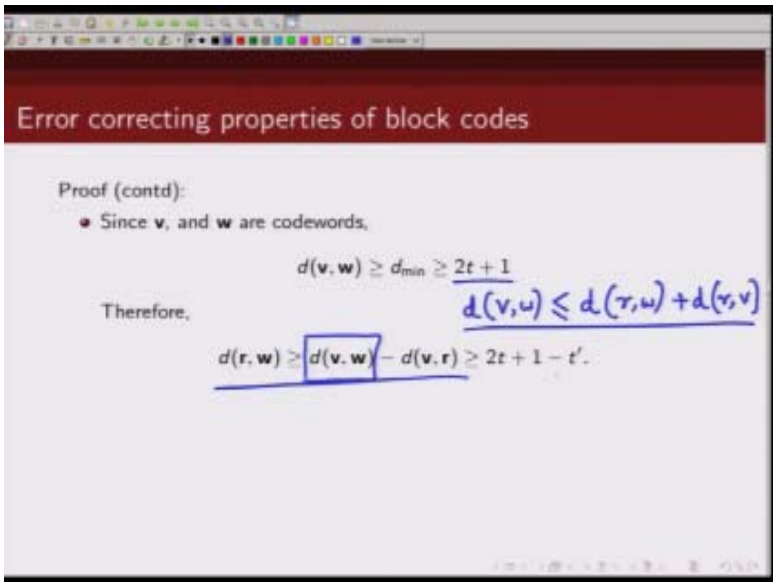
So from these 2 we can write that hamming distance between v and w is > than equal to 2t + 1. Now from the triangular inequality we know that hamming distance between r and w this we can see from here.

(Refer Slide Time: 18:30)



This relationship basically triangular inequality, what we have is.

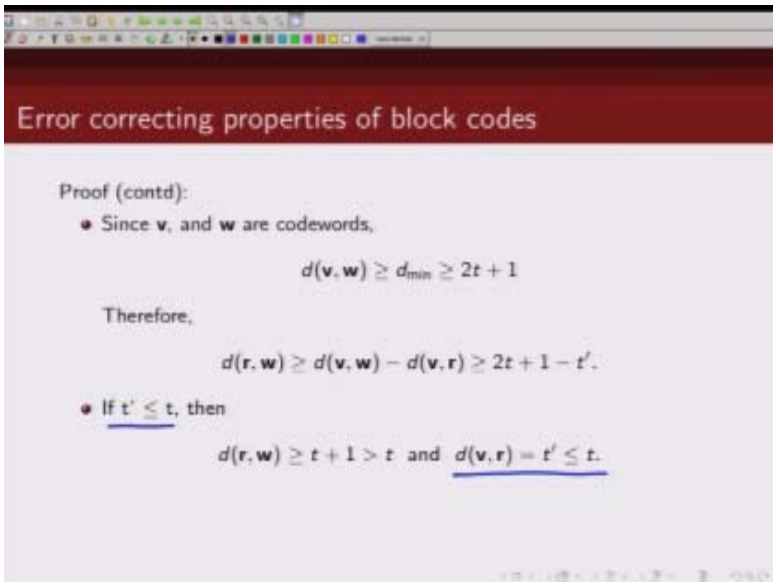
(Refer Slide Time: 18:35)



The hamming distance between v and w to be < than equal to hamming distance between r and w + hamming distance between r and v right? Now this we can write as, we can bring this here and we can bring this here, what we can write this as let us say we can write this, this relation in this particular form okay? Now what is this quantity, hamming distance between v and w, the hamming distance of between v and w is at least equal to 2t+ 1 and what is the hamming distance between the transmitted code word and the receive code word?

This is, we denoted by t – so then hamming distance between r and w is given by 2t + 1 – 2-.

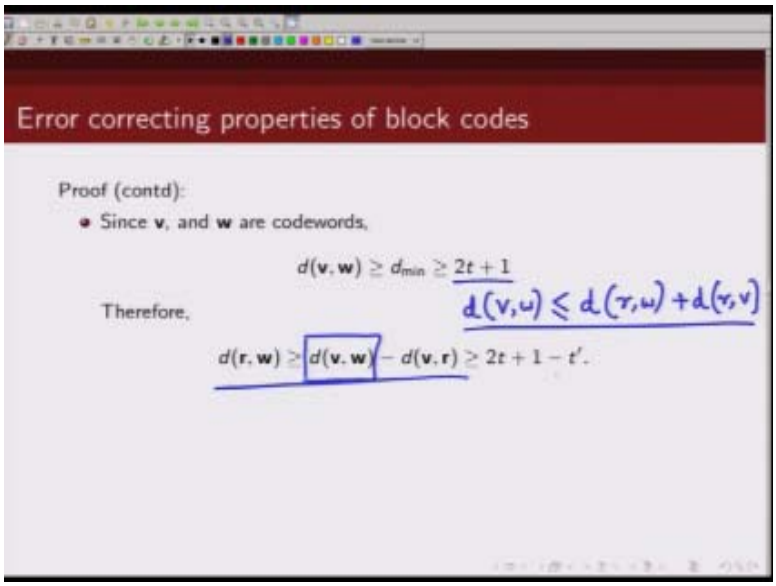
(Refer Slide Time: 20:00)



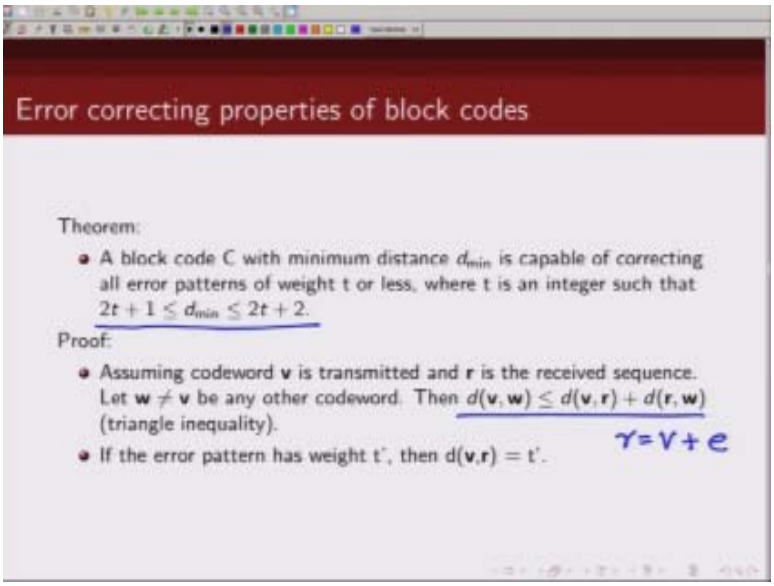
Now as long as your error pattern is < than equal 2t the weight of the error pattern is < than equal to t, in that case the hamming distance between r and w will be, we can plug that value of t here and what we will get is hamming distance between r and w is > than equal to t +1 which is > than equal to t, whereas the hamming distance between transmitted code word and the received code word is t hat which is < than equal to t what does it mean, it means that the received code word is closer to v then any other code word w.

So what will your maximum likelihood decoder for binary symmetric channel will decide in favor of? It will decide in favor of v so you will correctly decode this receive sequence to be v and this was our transmit code word, so you will not make an error. So what we have shown here is as long as your error pattern has weight u to t those error patterns are correctable provided

(Refer Slide Time: 21:42)

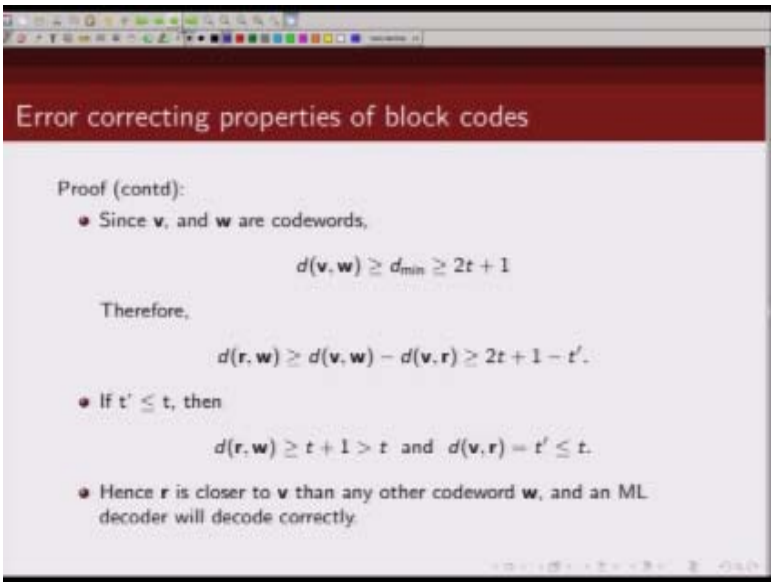


(Refer Slide Time: 21:43)



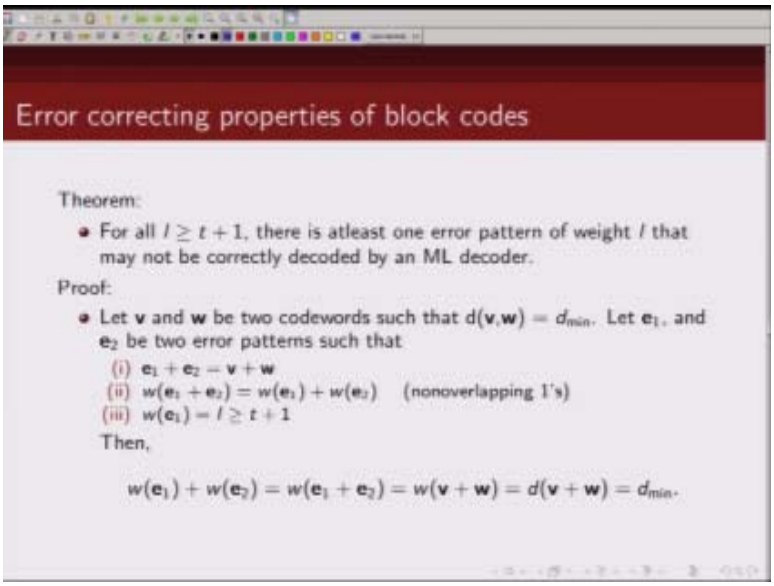
The minimum distance of your code is d_{min} and it satisfies this relationship, so if minimum distance of the code is at least 2t + 1 and it is $<$ than equal to 2t + 2 then it can correct all error patterns of weight t or less.

(Refer Slide Time: 22:09)



So as we can see here the received code word is closer to v then any other code word w so it will decide in favor of v and this r will be decoded as v.

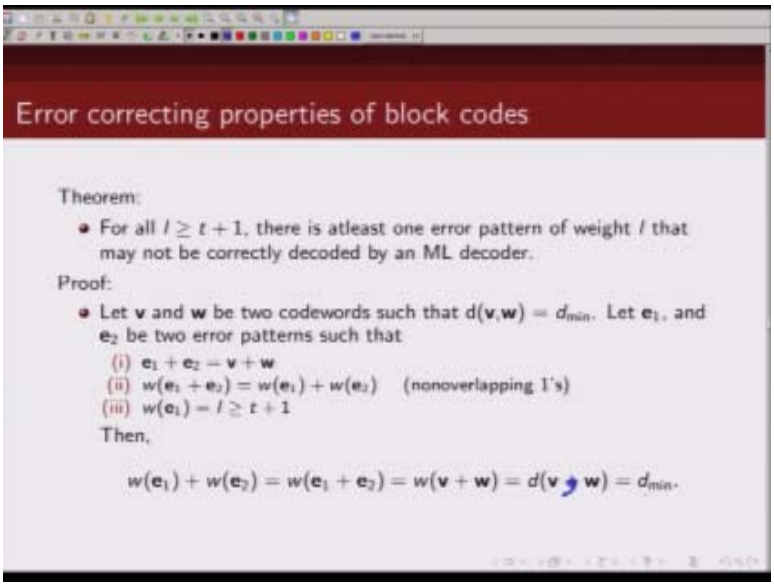
(Refer Slide Time: 22:23)



Next we are going to show that if there exists an error pattern of weight greater than equal to t +1 then our decoder whose minimum distance is at least 2t+1 but less than 2t + 2 this decoder will make an error, in other word it would not be able to correct this error pattern of weight t + 1so for all error patterns of weight L, if L is at least t + 1 then our maximum likelihood decoder may not be able to correctly decode.

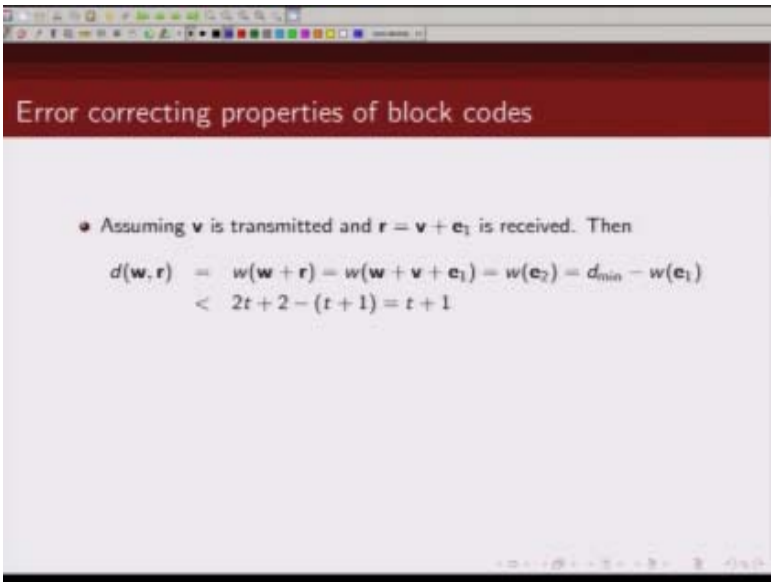
Or correct that error so let us prove this, if v and w are 2 code words and let us assume that the hamming distance between v and w is equal to the minimum distance of the code which is denoted by $t_{min,}$ and let e_1 and e_2 are two error patterns which satisfies these 3 properties and what are these 3 properties? The sum of e_1 and e_2 is same as v + w, the second property is e_1 and e_2 they do not have any over lapping ones so weight of e_1 +e2 can be written as.

(Refer Slide Time: 24:02)



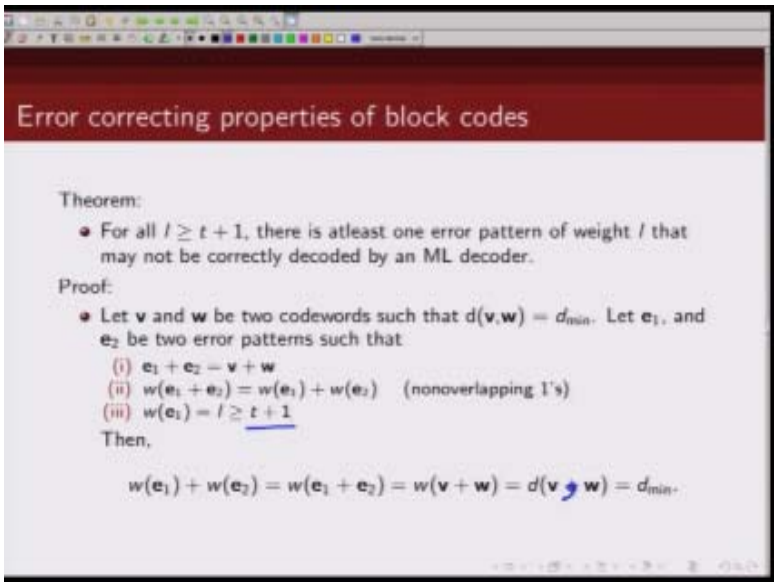
Weight of e_1 + weight of e_2, and we will show that if there is an error pattern of weight L where L is at least t + 1 then our maximum likelihood decoder will make an error in decoding. So the way we have chosen our error pattern weight of e_1 + weight of e2 is given by weight of e1 + e2 this is from 2, and from 1 we know e1 + e2 is nothing but v + w so this is same as weight of v + w and this is nothing but this is hamming distance between v and w and we have said the hamming distance between v and w is the minimum distance, so this is equal to minimum distance.

(Refer Slide Time: 25:06)



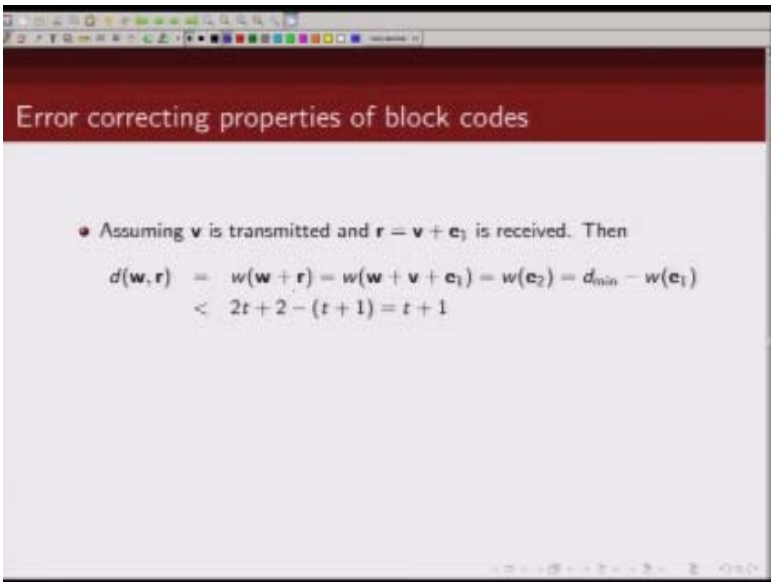
Now let us assume that we transmitted this code word v and what we received is r so this v got corrupted by this error pattern e1.

(Refer Slide Time: 25:20)



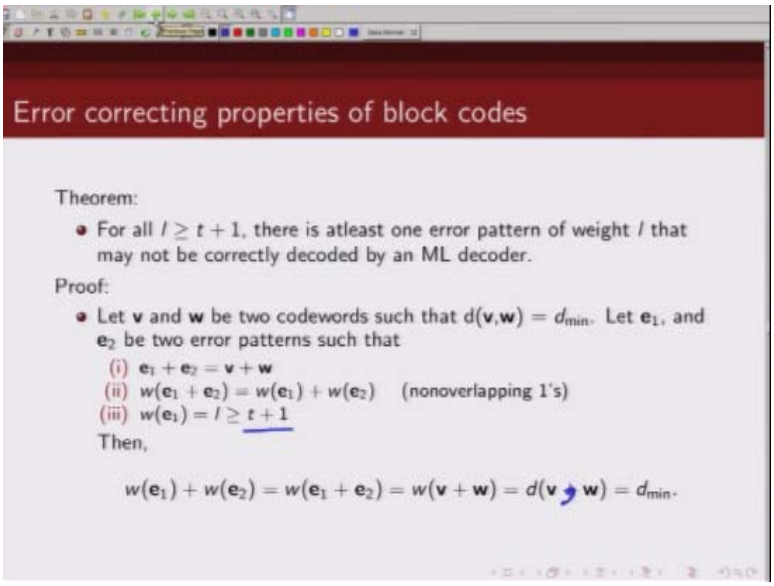
Which has hamming weight of at least t + 1.

(Refer Slide Time: 25:28)



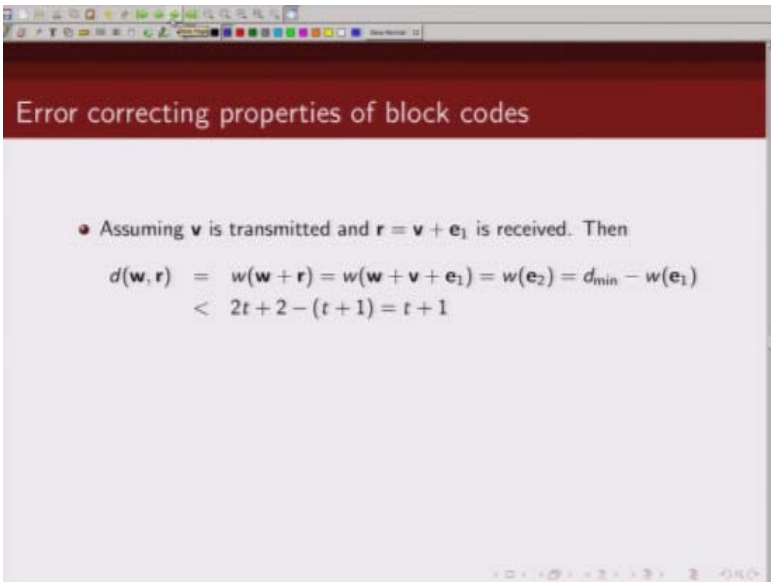
Now we will repeat the same exercise, we will try to find out the hamming distance of this received code word from the correct transmitted code word v and hamming distance between the received code word and any other code word w. So if we calculate the hamming distance between w and the received code word we know that hamming distance between w and r is nothing but hamming weight of w and r. And what is r? r is my received code word $v+e_1$. So I can write this as $w+v+e_1$. Now what is w+v?

(Refer Slide Time: 26:15)



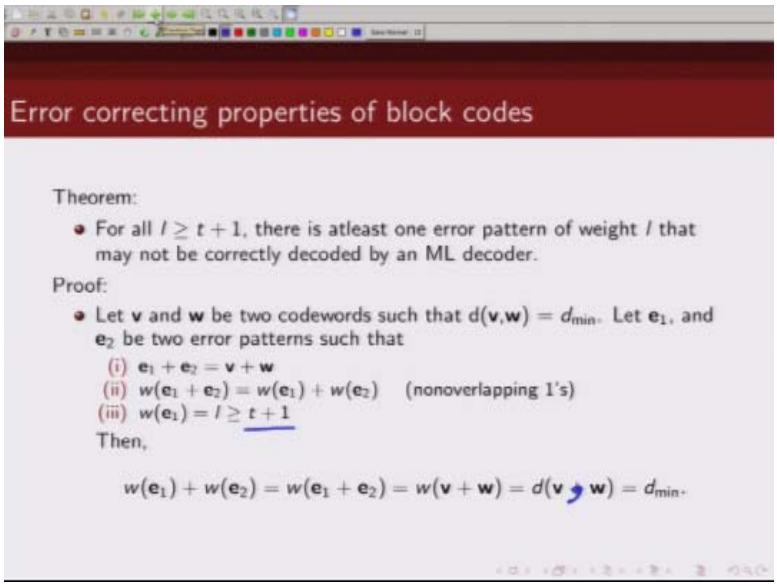
From one I have w+v is same as e_1+e_2.

(Refer Slide Time: 26:23)



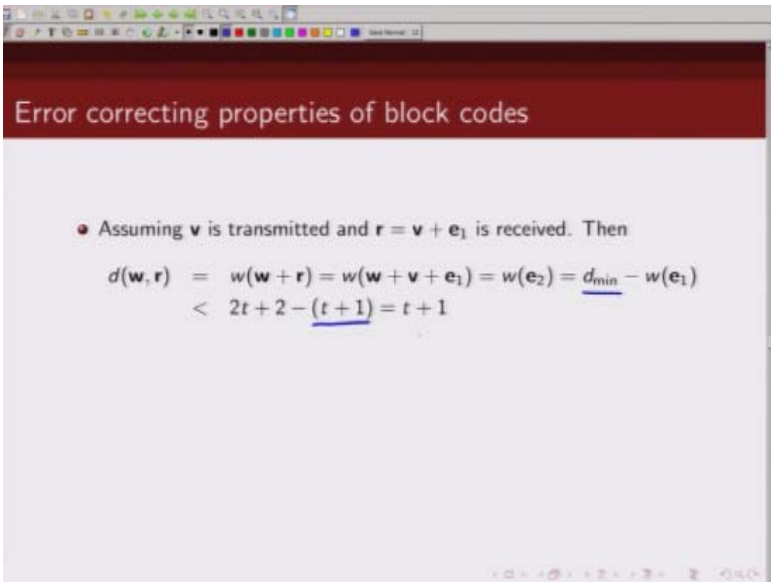
So then this is $e_1+e_2+e_1$ so e_1+e_1 will be zero so this will be e_2 $w(e_2)$ and what is $w(e_2)$?

(Refer Slide Time: 26:37)



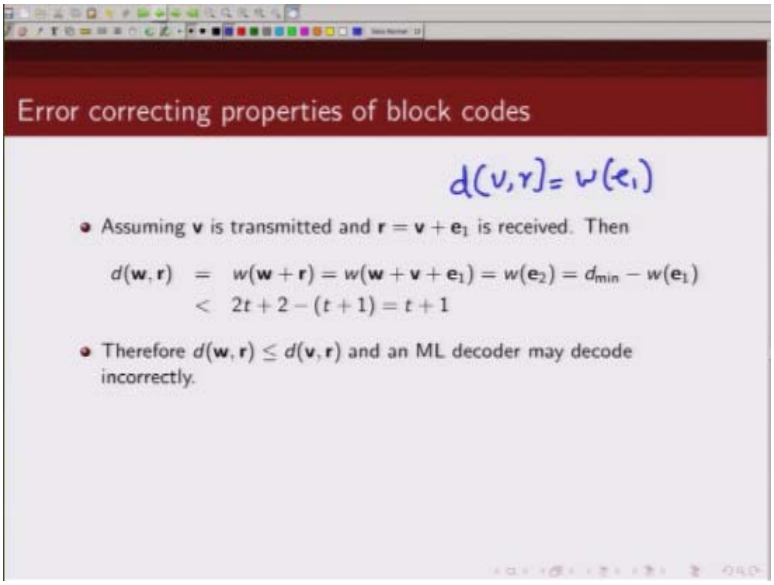
From this relation we can see $w(e_1) + w(e_2)$ is d_{min}. So $w(e_2)$ is $d_{min} - w(e_1)$.

(Refer Slide Time: 26:50)



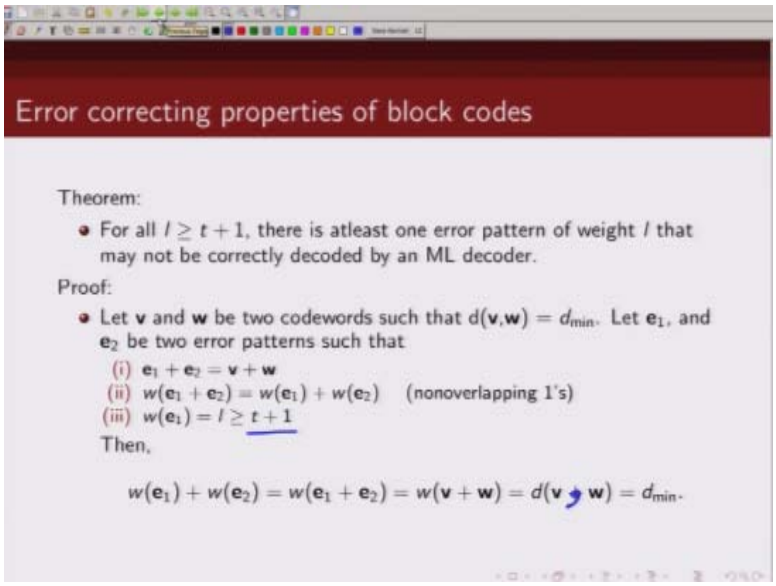
So this we can write as $w(e_2)$ as $d_{min} - w(e_1)$. So d_{min} is less than equal to $2t+1$ and $w(e_1)$ is atleast $t+1$ so $w(e_2)$ will be less than $2t+2-(t+1)$ which is $t+1$. So the hamming distance between w and r is less than $t+1$.

(Refer Slide Time: 27:21)



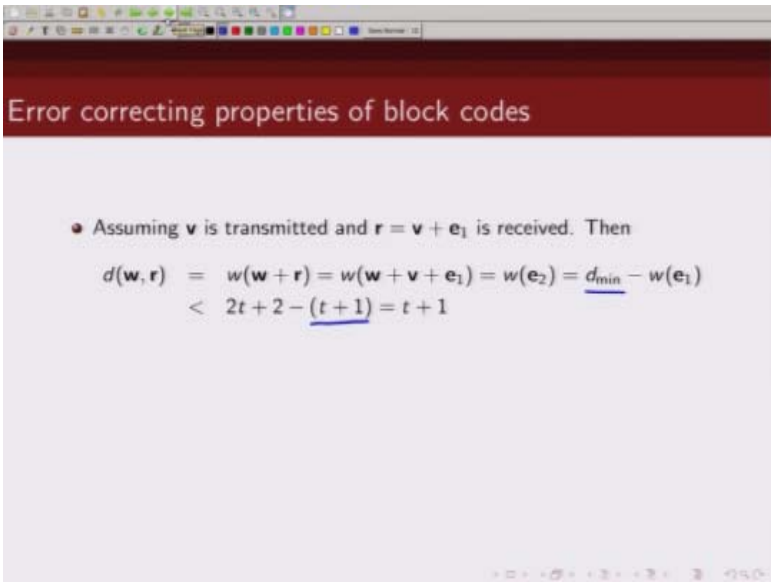
And what is the Hamming distance between v and r, this is $w(e_1)$ okay. And what is $w(e_1)$, $w(e_1)$ is given by

(Refer Slide Time: 27:41)



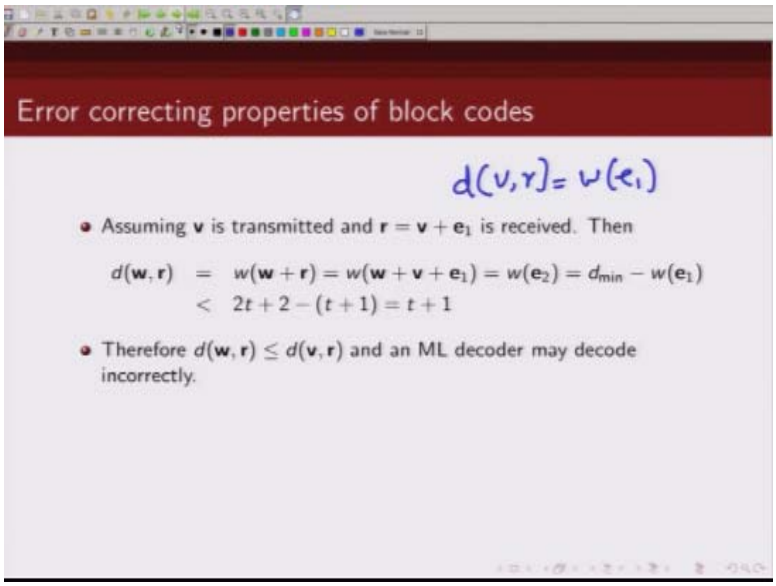
l which is atleast t+1.

(Refer Slide Time: 27:46)

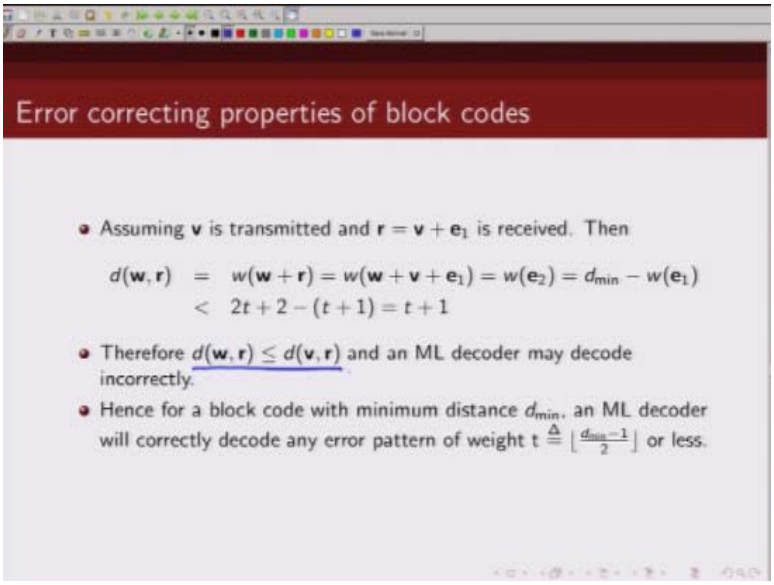


So what we have shown here is

(Refer Slide Time: 27:48)



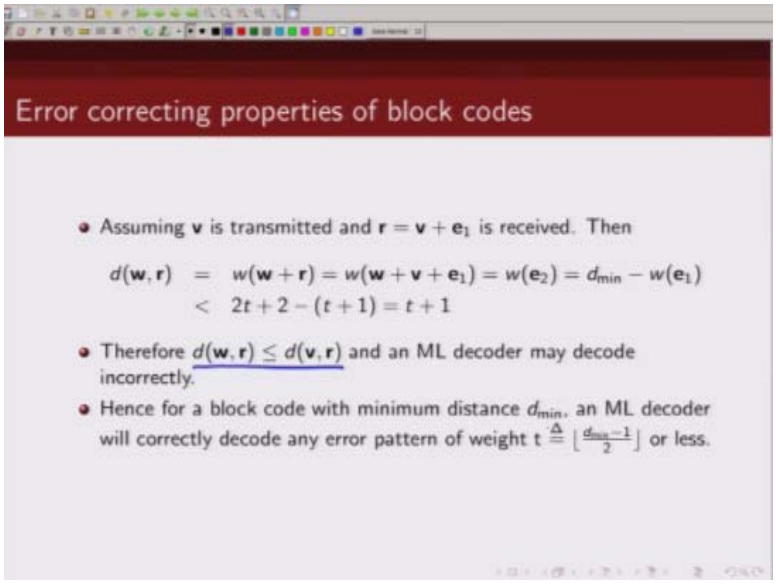
(Refer Slide Time: 27:49)



Weight of w, hamming distance between w and r is less than t+1 whereas hamming distance between v and r is greater than equal to t+1. So what we have shown is hamming distance w and r is less than equal to hamming distance between received code word r and the true code word which was actually transmitted which is v. So in this case the maximum likelihood decoder will

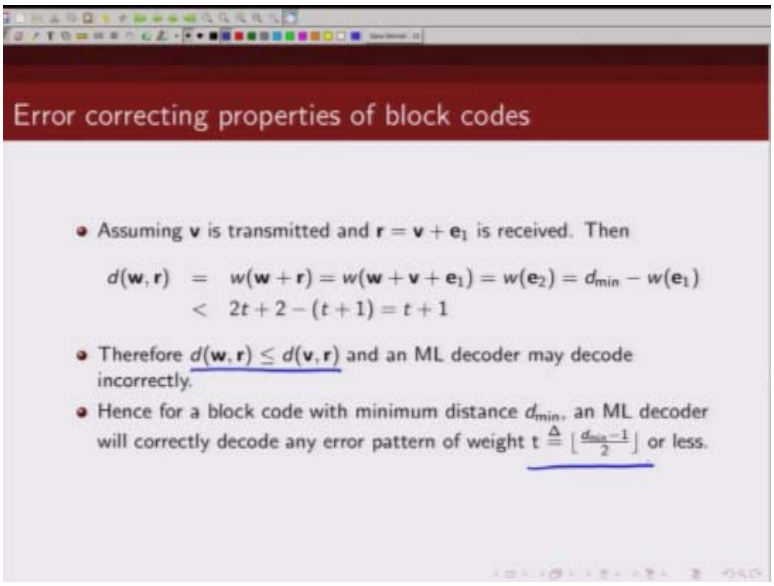
decode in favor of w and not v and will make a mistake. So through this construction we have shown that if your error pattern is a weight t+1 then you are not guaranteed to correct that error. So from this and the previous result

(Refer Slide Time: 28:42)



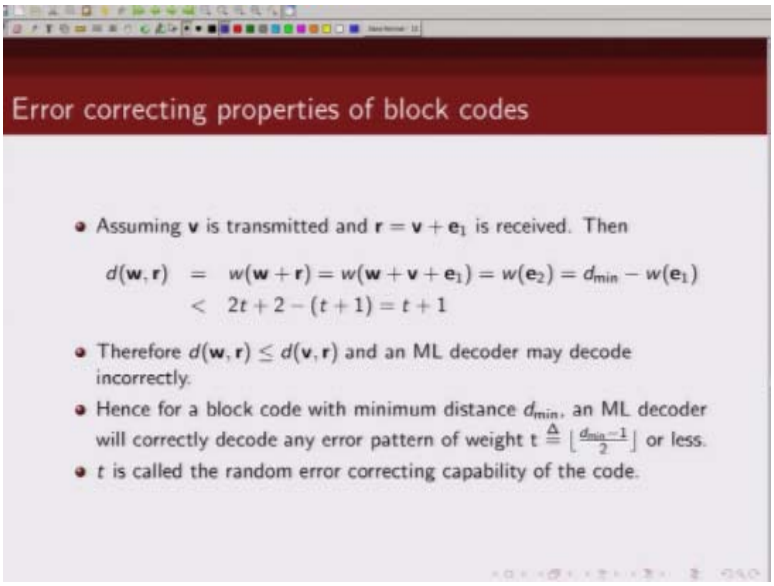
We can conclude that if we have a block code with minimum distance d_{min} which satisfies relationship that d_{min} lies between 2t+1 and 2t+2 then this linear block code with minimum distance d_{min} should be able to.

(Refer Slide Time: 29:02)



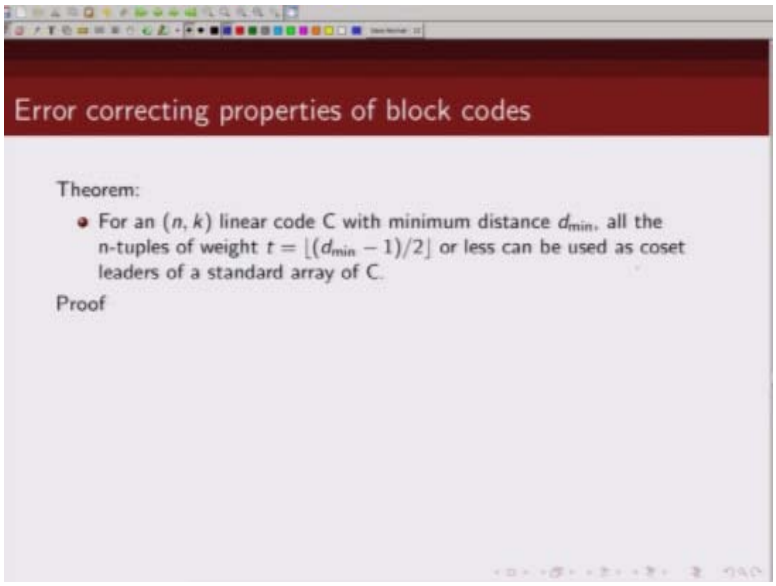
Correct all error patterns upto weight t, where t is given by this.

(Refer Slide Time: 29:12)



So this t is known as random error correcting capability of the linear block code. Next we are going prove a result which is as follows.

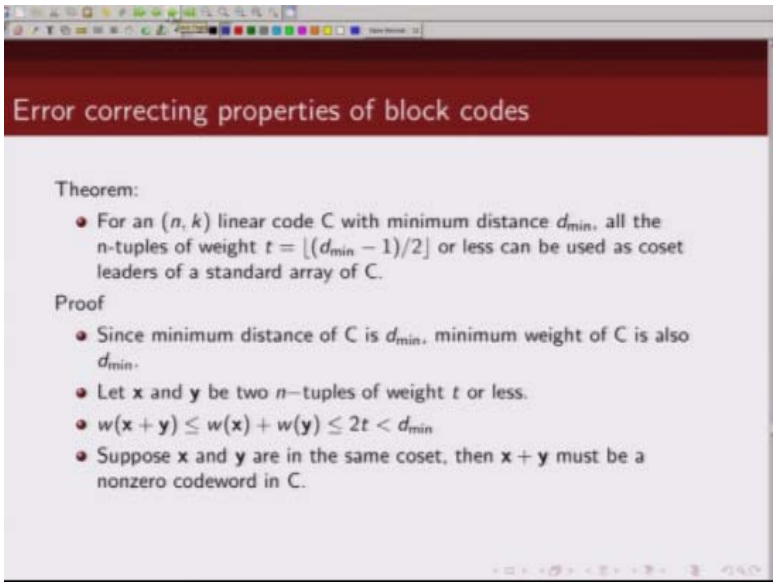
(Refer Slide Time: 29:25)



So if we have an (n ,k) linear block code whose minimum distance is given by d_{min} then we can show where d_{min} lies between 2t+1 and 2t+2, then we can show that all n-tuples of weight t or less can be used as coset leader in our standard array. So we are going to prove this result using method of contradiction. Now let us say, so how does method of contradiction work, so we will say let us say they are all error patterns are weight upto t. Let us say they are not coset leader.

So let, let us we will assume our scenario where there are two such n-tuples with weight up to t which are not coset leader. In other words, they lie in the same coset or same row. And then later on we will show that, that is not possible. So that is how this

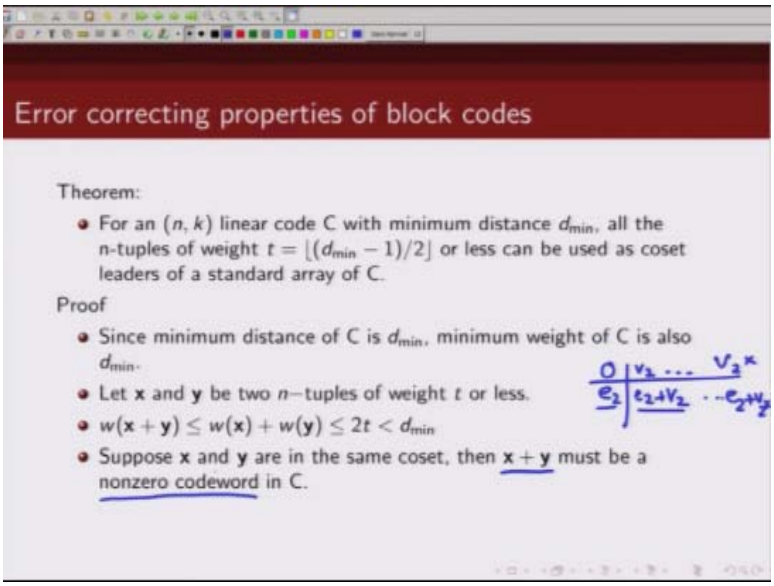
(Refer Slide Time: 30:32)



Method of contradiction will work. So minimum distance of a code is d_{min} so minimum weight of the code is also d_{min}. Let x and y are two n-tuples of weight t or less. Now w(x+y) will be less than equal to w(x)+w(y), why because there might be some over lapping ones at some locations of this n-tuple x and y, and we are given that the weight of x and weight of y is at most t.

So then w(x)+w(y) will be less than equal to 2t and this is less than minimum distance because minimum distance of a code is atleast 2t+1. Now let us assume that these x and y which are error patterns of weight t or less, let us assume that they are not coset leader, so if they are not coset leaders let us assume they are in the same coset, they are in the same row.

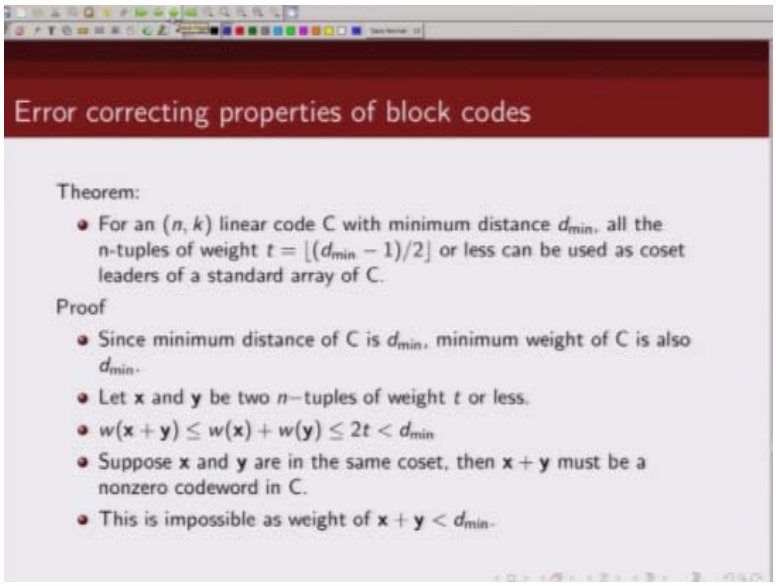
(Refer Slide Time: 31:50)



So if we assume x and y are in the same row or same coset then x+y must be a code word, why this is so, if you recall your standard array we had something like this, first row first column was all zero vector and then we had other code words. And then we had error pattern let us say e_2 so this was e_2+v_2 like this was $e_2+v_2^k$. Now if you look at any two elements in the same coset or same row and if you add them up what do you get?

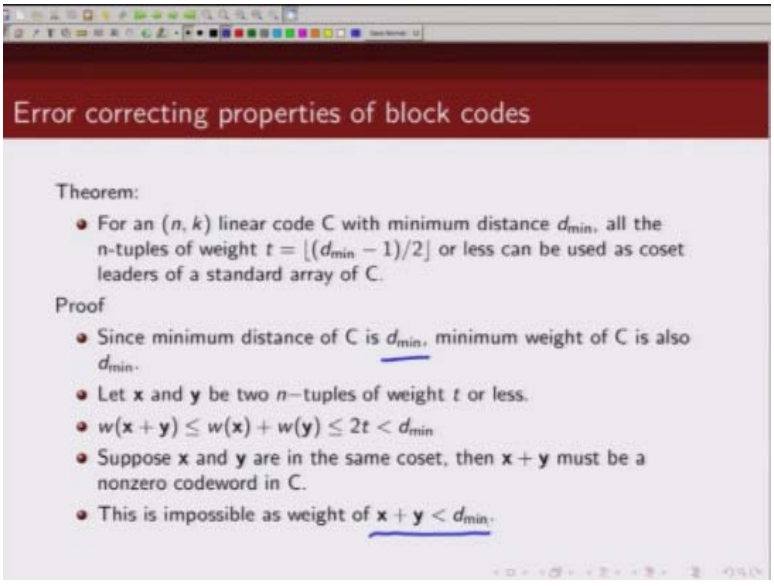
Let us add this and this, what do we get, $e_2+e_2+v_2$ we will get v_2. If we add this and this we will get $v_2+v_2^k$ which is another code word v_s. So if we take any two elements in the same coset and we add them up, we are going to get a non zero code word. So if x and y are in the same coset then x+y must be a code word.

(Refer Slide Time: 33:20)



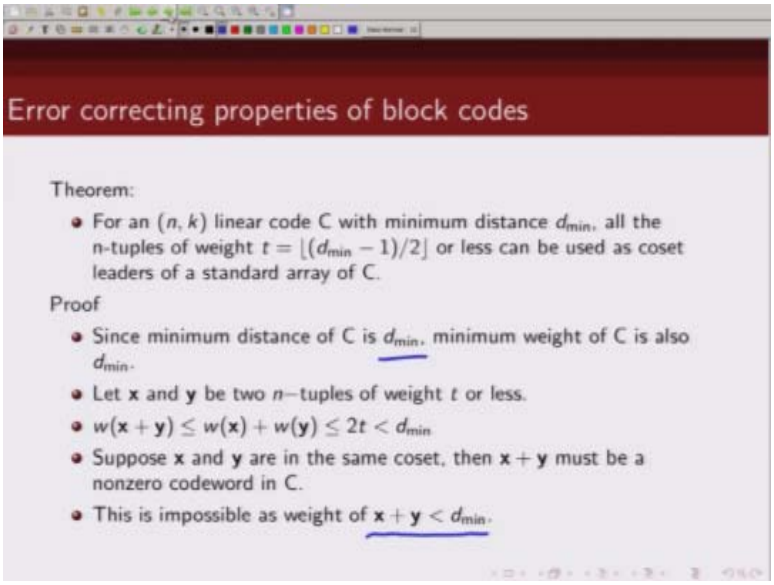
This is impossible, why, if x+y is a code word then what is the minimum distance of x+y, x+y minimum distance of that must be d_{min}.

(Refer Slide Time: 33:34)



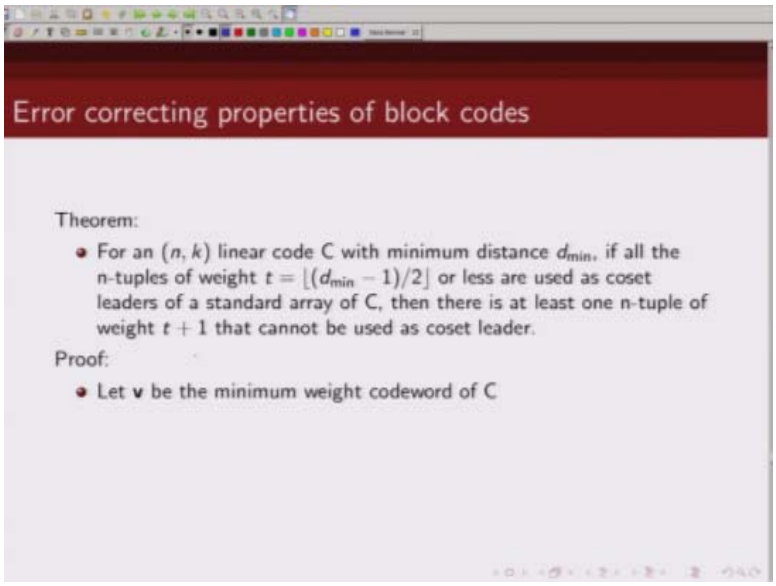
But what is the, what is the weight of x+y we just showed in this bullet that weight of x+y is less than d_{min}, that means weight of x+y is less than d_{min}. If weight of x+y is less than d_{min} then x+y cannot be a non zero code word, because the weight of a non zero code word should be atleast d_{min}. So our assumption that x and y are in the same coset is wrong. In other words then x and y must be in different cosets, different rows and we can always make these x and y as coset leaders. So this proves our result that.

(Refer Slide Time: 34:34)



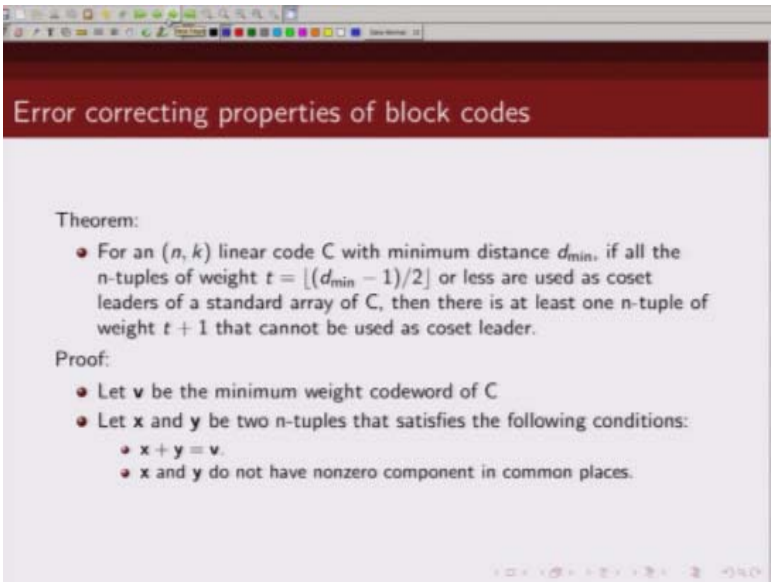
All n-tuples of weight n of weight t or less can be used as coset leaders in the standard array. And we know that if we use them as coset leaders we, those are our correctable error patterns.

(Refer Slide Time: 34:53)



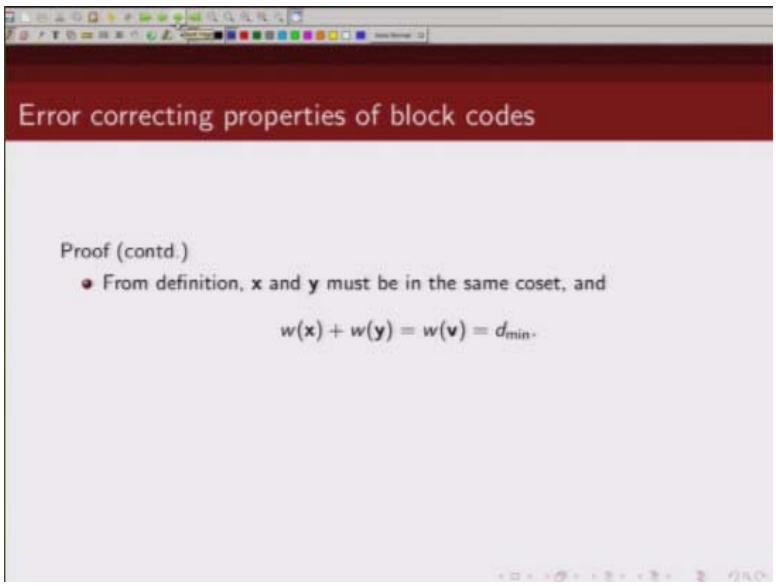
Next I am going to show your result which is as follows. So if you have an (n, k) linear block code whose minimum distance is d_{min} and if all n-tuples of weight t or less are already used as coset leader, then there is atleast one n-tuple of weight t+1 which cannot be used as coset leader. So this essentially is going to show us again the same result that any weight pattern of weight pattern, error pattern or weight t+1 is not guaranteed to be corrected.

(Refer Slide Time: 35:37)



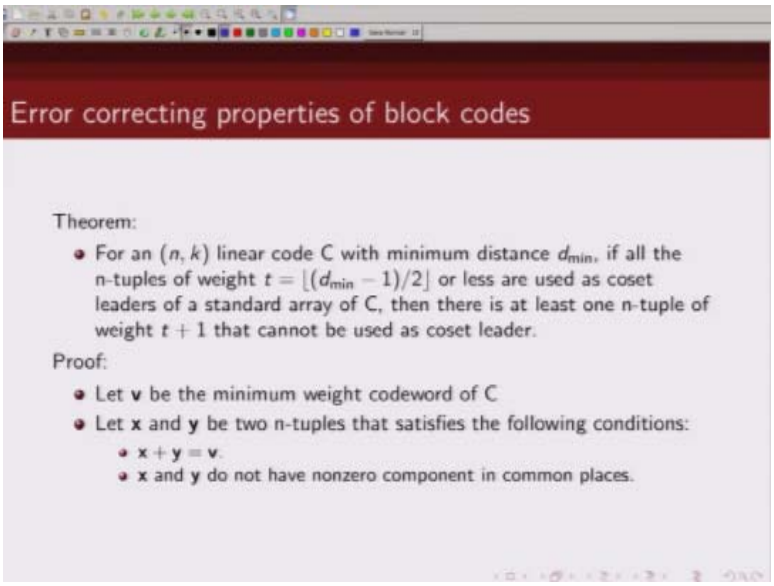
So how do we prove it, so let us assume v is the minimum weight code word of C and we have two n-tuples x and y which satisfies these following conditions. First x+y=v and x and y do not have any component common, so they do not have ones common in same position.

(Refer Slide Time: 36:10)



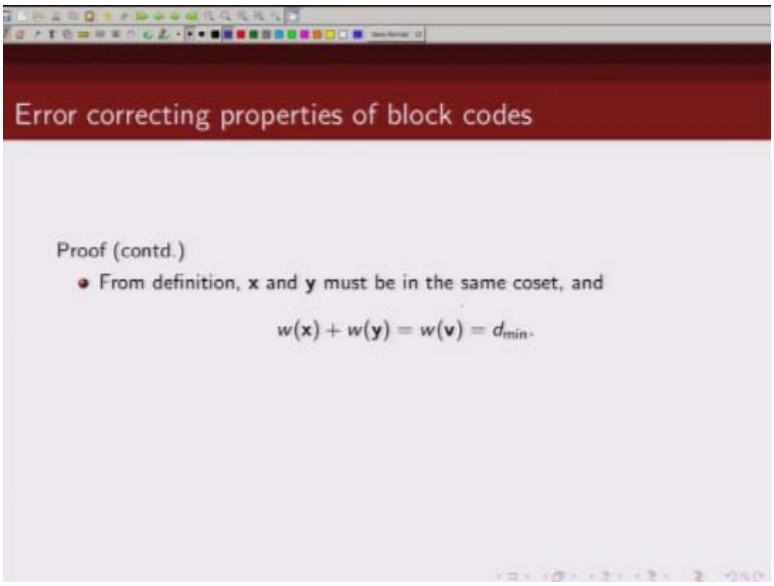
So from the definition

(Refer Slide Time: 36:13)



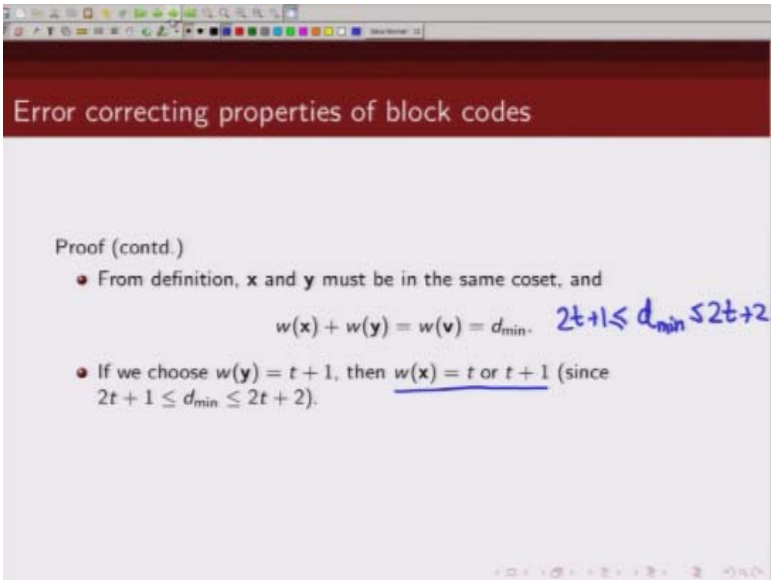
x and y must be in the same coset, why because we know if two elements are in the same coset and if we add them the sum is a valid code word. So if x+y=v which is a valid code word then x and y must be in the same coset.

(Refer Slide Time: 36:38)



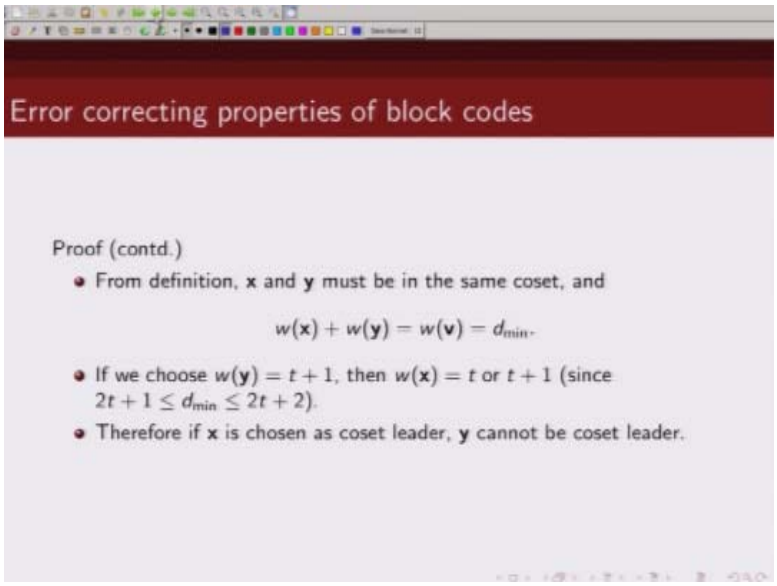
So that is what I said from definition x and y must be in the same coset because x+ y is v which is a valid code word. And we know that if we add any two elements in a coset their sum is a valid code word.

(Refer Slide Time: 37:01)



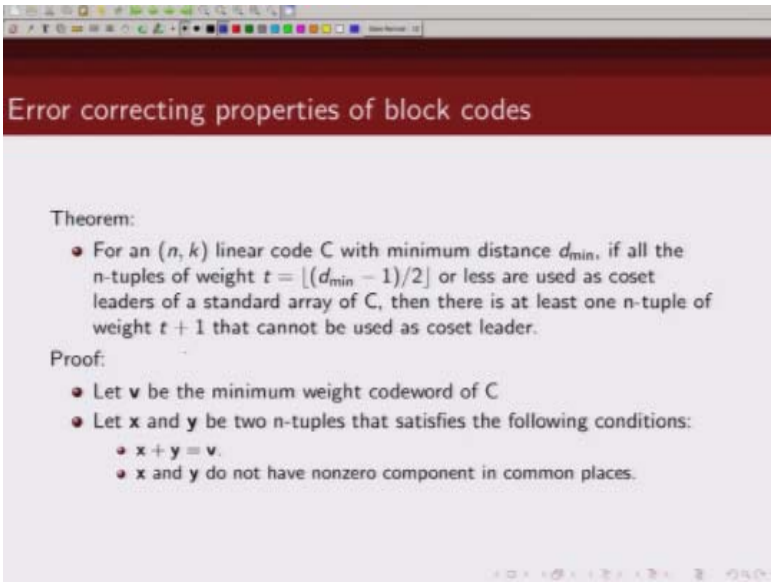
And similarly w(x) + w(y) = w(v) and we have chosen v to be the minimum distance code word so this is given by d_{min}. Now if we choose our y to have a weight of t+1 then we can see from here d_{min} is greater than equal to 2t+1 but less than equal to 2t+2. So from this and using the fact that d_{min} lies between 2t+1 and 2t+2 using these two results what we get is w(x) can be t or t+1.

(Refer Slide Time: 37:59)



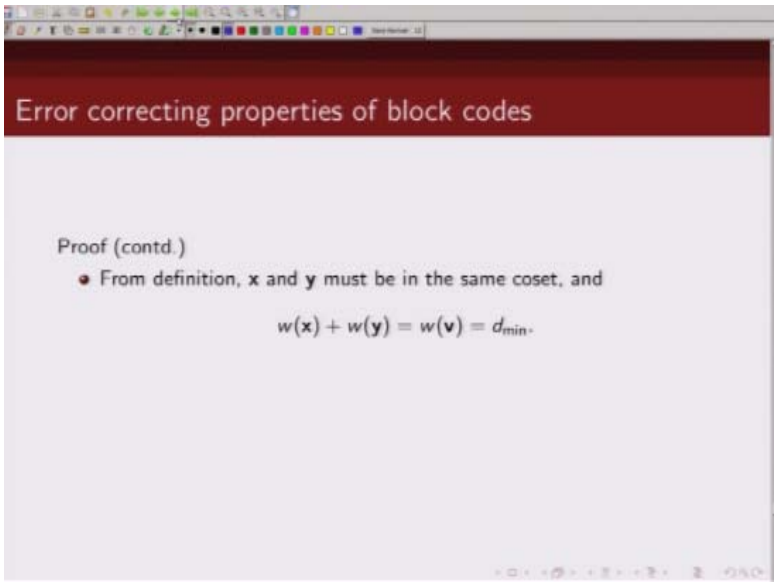
So therefore if we choose x to be our coset leader then we cannot choose y as our coset leader. You can see, because x and y are in the same coset and w(x) is t or t+1, whereas w(y) is t+1 so I will choose x as my coset leader and if I choose x as my coset leader then I cannot choose y as my coset leader.

(Refer Slide Time: 38:33)

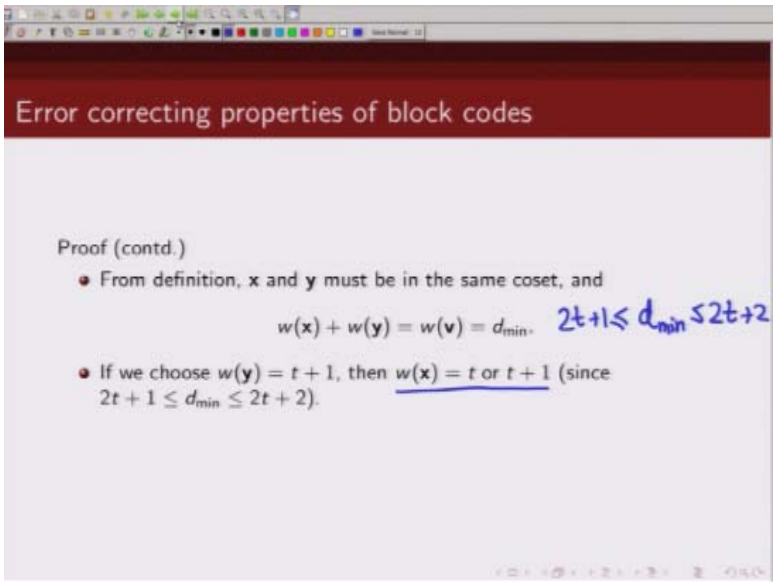


Which proves my result which says that if all n-tuples of weight t or less are used as coset leaders then there exists atleast one error pattern of weight t+1 which cannot be used as coset leader.

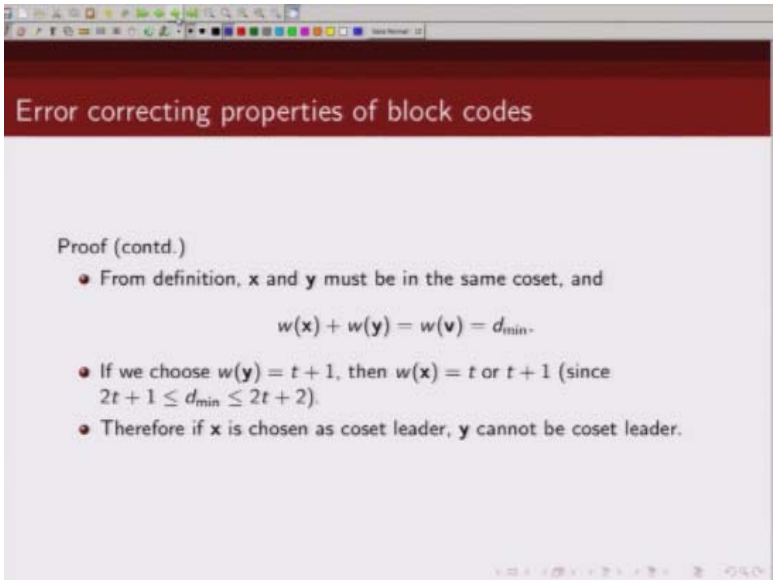
(Refer Slide Time: 38:50)



(Refer Slide Time: 38:51)

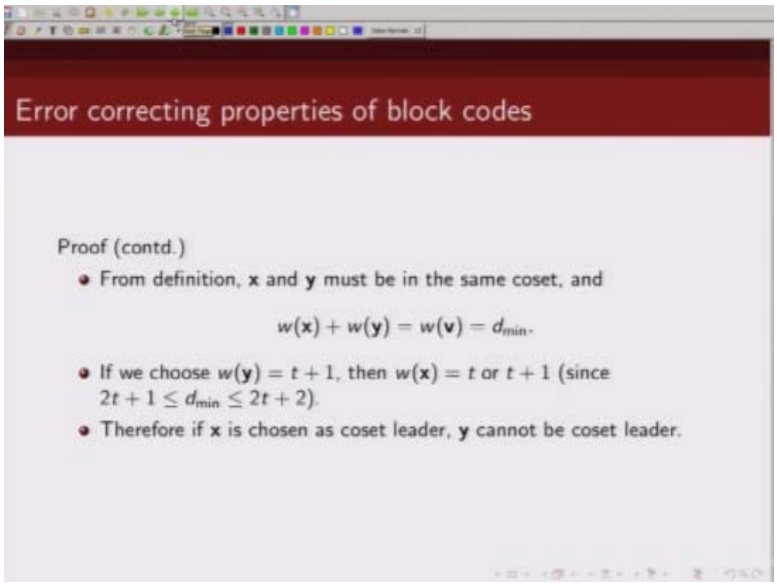


(Refer Slide Time: 38:52)



And if this error pattern of weight t+1 cannot be put as coset leader then this is not a correctable error pattern.

(Refer Slide Time: 39:00)



So with this I will conclude my lecture on random error correcting and random error detecting properties of 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