

**Digital Communication**  
**Prof. Bikash Kumar Dey**  
**Department of Electrical Engineering**  
**Indian Institute of Technology, Bombay**

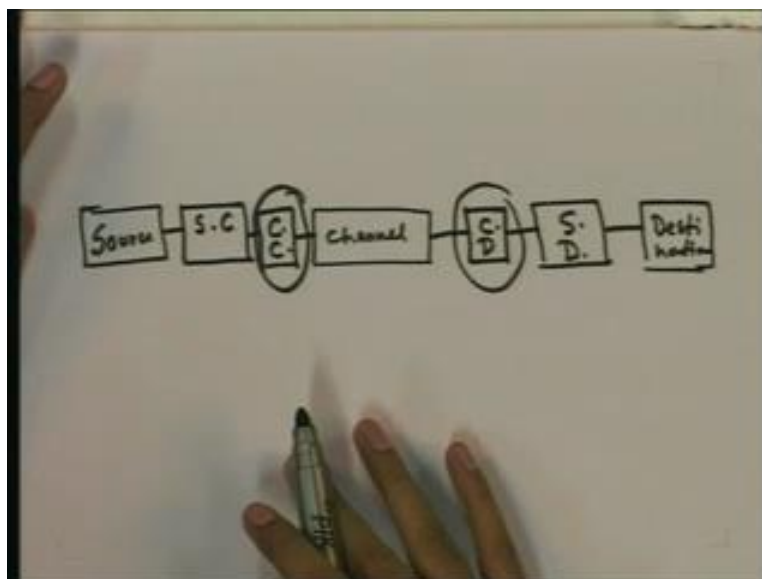
**Lecture - 30**  
**Channel Coding**

For the last few classes, we have discussed source coding purpose of which was to remove redundancy from the source generated message. And that resulted in compression of the source message. And in this class we will discuss another component of a communication system called channel coding. Here the purpose is to introduce redundancy just the opposite to source coding in such a way that, we can correct errors or distortion introduced by the channel at the receiver.

So, we have discussed already before in this course that, a communication system will usually have in the transmitter side a source encoder a channel encoder and then the signal will be transmitted and then the signal will go through the channel. It will be superimposed with noise and then when the receiver, receives the signal it has already some noise.

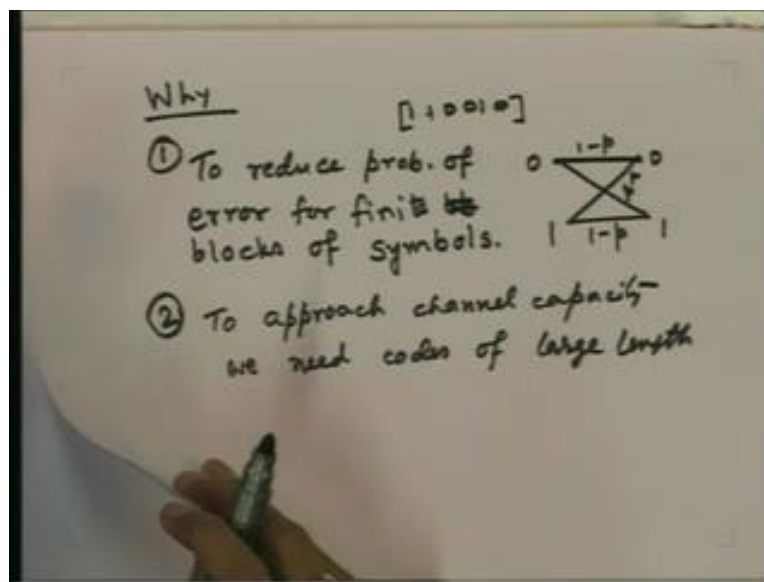
Now, the noise distorts the signal or it introduces error to the message. Now, if we want to correct errors to some extent we need to have some redundancy in the transmitted signal. And that, is the purpose of channel coding.

(Refer Slide Time: 02:48)



So, we have here, the channel, source then we have here source encoder which we have discussed before, source coding and then we have already discussed, a particular type of source coding technique called as known as lossless source coding techniques. And in this code in this class, we will discuss another component called channel coding. And then here we will have the decoders for these coding in the reverse order. So, here first we will have channel decoder then we will have source decoder then destination. So, we are going to discuss these components in this class. So, what is the purpose of channel coding.

(Refer Slide Time: 04:19)



Let us see, why should there be any channel encoder and channel decoder at all in a communication system. So, the purpose of channel coding is to suppose, that we transmit a block of symbols we are transmitting a block of symbols let us say this, the channel is a binary symmetric channel that, we have already discussed different kind of channels

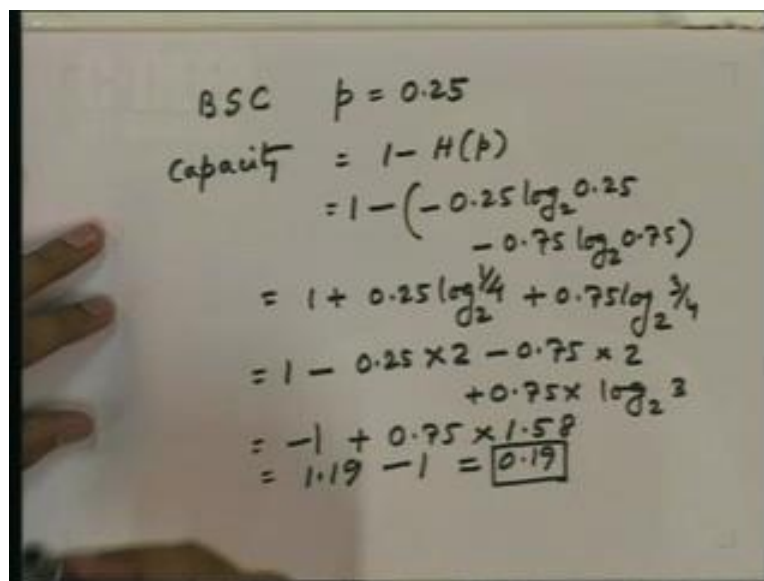
So, it is a binary symmetric channel, 0 goes to 0 with some probability  $1-p$  0 goes to 1 with some probability of error  $p$  1 goes with probability of error  $p$ . This is a binary symmetric channel. 0 to 0 is  $1-p$  this is also  $1-p$ . So, if we are transmitting a sequence of bits like this through this channel then there will be some errors introduced by this channel in this block. Now, if we want to be able to correct some errors in this block then we should have some redundancy in the, in this block. So, if we have redundancy in this block, we will be able to correct some errors and

as a result we will be able to reduce the effective resulting probability of error after correction of errors the remaining probability of error will be less than  $p$ .

So, to reduce probability of error for finite blocks of symbols. So, this is 1 purpose. Another way to view, it is in terms of Shannon's work we have already discussed these concepts before, in the little bit of information theory we have discussed. Then that is we have seen there are 2 the every channel has a capacity has a maximum limit, which actually limits the amount of information that, can be transmitted through the channel per use.

So, there we have also seen that, that channel capacity can be achieved by using coding and there also we have said that to actually approach channel capacity, we should have large block length codes and that block length should be very large to approach channel capacity. So, that also motivates us to do channel coding. So, this is to approach channel capacity we need codes of large length. So, we will see with example, what we mean by this.

(Refer Slide Time: 07:45)



The image shows a handwritten calculation on a whiteboard for the capacity of a Binary Symmetric Channel (BSC) with a probability of error  $p = 0.25$ . The steps are as follows:

$$\begin{aligned} \text{BSC } p &= 0.25 \\ \text{Capacity} &= 1 - H(p) \\ &= 1 - (-0.25 \log_2 0.25 - 0.75 \log_2 0.75) \\ &= 1 + 0.25 \log_2 \frac{1}{4} + 0.75 \log_2 \frac{3}{4} \\ &= 1 - 0.25 \times 2 - 0.75 \times 2 \\ &= -1 + 0.75 \times 1.58 \\ &= 1.19 - 1 = \boxed{0.19} \end{aligned}$$

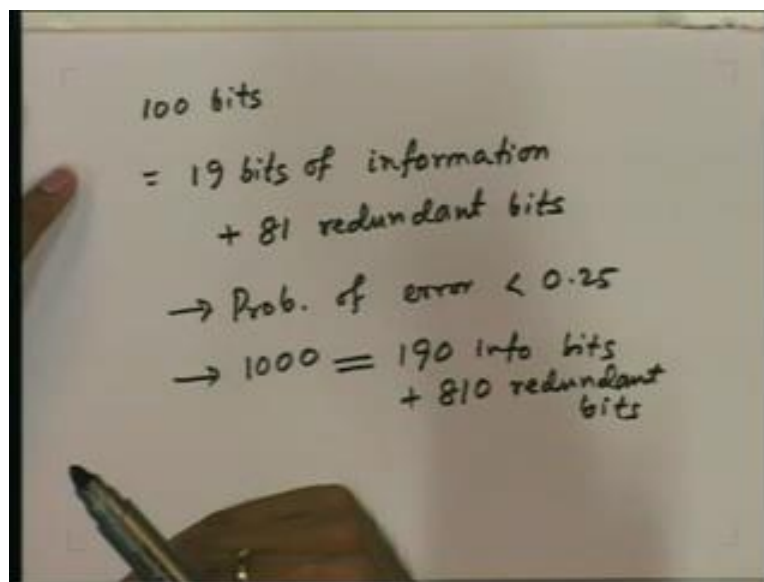
Let us say, we have a BSC binary symmetric channel with probability of error  $p$  equal to 0.25 that means; on average every fourth bit goes in error. So, the capacity of this channel is known to be 1 minus  $H_p$ , where  $H_p$  is the entropy of entropy function of  $p$ . So, this is  $H_p$  is minus 0.25 log 0.25 minus 0.75 log 0.75 and that is same as 1 plus 0.25 log 4 1 by 4 and 0.75 log 3 by 4. So, log

1 by 4 is minus 2. So, 1 minus 0.25 into 2 then again we have here minus 0.75 multiplied by 2 that is for the 4 and then 3 we have plus 0.75 times log 3.

Now, this is 0.5 and this is 1.5. So, 1.5 and 1.5 is 2. So, 1 minus 2 is that is minus 1 plus 0.75 times log 3 base 2 and that is 1.58. So, we get here this term is 1.19 minus 1 that is 0.19. Now, what do you see here the though this channel is a binary channel meaning by it can a bit can be transmitted and bit a will be received. The channel does not really have 1 bit capacity meaning by, you cannot transmit 1 bit just 1 bit and receive 1 bit correctly always.

So, there will be error sometimes. So, if you want to reduce the error and reduce the error to almost 0. How to do it? If we want to reduce the error to almost 0, we will not be able to transmit 1 bit of information per use of the channel. So, we can achieve according to Shannon's result we can achieve this capacity for that, for this channel meaning by for every bit we will be able to transmit 0.19 bits of information. What does it mean? It simply means that out of say every 100 bits we transmit, we will be able to transmit nineteen bits of information all the other bits should carry no information.

(Refer Slide Time: 11:25)



So, this motivates us to say design a code. So, we will have say 100 bit 100 length code out of which 19 bits of information and 81 redundant bits because, you have seen that for this channel to reduce the probability of error to almost 0 we cannot transmit more than 19 bits out of 100 for

19 bits of information per 100 bits of transmission. So, that means; if we want to use transmit 100 bits blocks then out of 100 bits nineteen bits should be information and 81 bits should be redundant.

They should depend on those 19 bits, they should carry no more no extra information.

Now, even if we use this way we, if we construct codes, codes means basically the possible blocks that we will transmit. So, all possible 100 bit length blocks bit sequence is that we will transmit. And out of all those blocks only 19 bits carry information all the other bits are redundant. So, even if we use such a code we will have some probability of error even after correction of the errors because we will not be able to correct all errors.

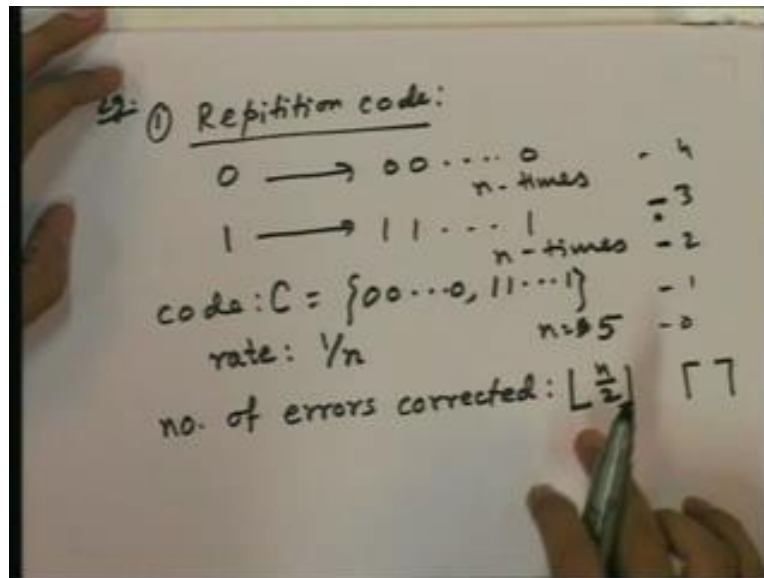
We are not saying that whatever, is the number of errors channel introduces we will be able to correct that is never possible.

So, with this particular code, if we construct a code like this then we will be able to correct many errors, but still there will be many types of errors we will which we will not be to correct. So, for those also we will still have a bit a probability of error which will not be zero, but the probability of error bit error probability will be, now reduced to less than 0.25. So, with this the probability of error is less than 0.25. Now, Shannon's work also says that, actually to bring probability of error to very almost 0.

So, this will reduce to some extent, but if we want to bring to even less value then we will need to increase this length. Let us say, we take 1000 lengths and then we take 19 bits of information and then 810 redundant bits. If we use such a code then our probability of error will be further reduced. So, this way as we increase length we will see that the we can feel that, the probability of error will reduce and this way we can go to almost 0 probability of error.

So, this is what actually Shannon's work says and this is actually this results Shannon's channel coding theorem and its proof actually laid to more research in this area. And people try to find out construct how to construct codes and how to do encoding and decoding efficiently and also such research in the area of coding theory. Now, let us see some examples of codes the simplest codes which will not require any rigorous background we will first discuss those. Say the first the most simple example is repetition code.

(Refer Slide Time: 15:52)



Where, if you want to transmit 0 just transmit repeat 0 n number of times n may be 10 it may be 5 it may be 100. So, transmit 0 n times and if we want to transmit 1 transmit 1 n times. So, the code, code is actually the set of blocks that will be transmitted. So, in this case the code is all 0 and all 1 it has only 2 code words all 0's and all 1's. So, you can see that; obviously, using such a code you can transmit only 1 bit of information using the code. So, because there are 2 possibilities.

So, you can transmit only bit if there are 4 code words you can transmit 2 bits if you have 16 code words you can transmit 4 bits and so, on. So, these are this is called the code there is set of all such blocks that will be transmitted actually and these individual blocks will be called code words. So, using this we can see now, what is the rate of the code meaning, by what is the rate of information we are transmitting per use of the channel. To transmit a single code word we are using the channel n times because we are transmitting n bits, but how much information are you transmitting using those n bits.

We are transmitting only 1 bit of information though it has we are transmitting n bits.

It is carrying only 1 bit of information. So, the rate of the code is 1 by n that is part bit we are transmitting 1 by n bits of information, rate is 1 by n and then how many errors can we correct if so, how would we decode we know that, the channel will introduce some errors. So, 1 obvious

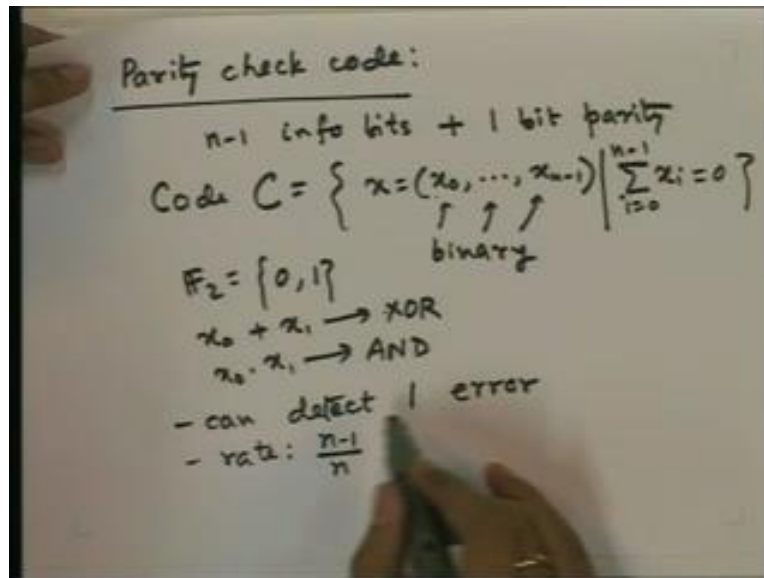
way 1 intuitively satisfying way is to see how many 0s we have received and how many 1s we have received in the block.

If the number of 0s is more than number of 1s then we will assume that actually 0 was transmitted and 1s are because of the channel introduced errors. So, if we actually do the decoding at the receiver that way, that is take the majority number of bits. Then how many errors can we correct you can see that, if the number of errors is more than  $n/2$  actually the floor of  $n/2$  that is; if suppose, we have  $n$  equal to 5 then if the number of errors is less than equal to 2. Then we will still have majority number of correct bits 3 bits will be correct and 2 bits will be wrong.

So, we will be able to decode correctly whereas, if we have 3 or more bits of errors then majority number of bits will be in error. And so, we will be we will not be able to decode correctly we will decode wrongly. So, the number of errors that we can correct is obtained by dividing 5 by 2 and taking the integer part of that. So, what is result of dividing five by 2 it is 2.5. So, 2 bits can be corrected. So, only take the integer part and that number is called floor of  $n/2$ .

So, take this is, if we plot integers 0 1 2 3 4 then suppose, you have number here. The floor of that is this ceiling of this number is this. So, ceiling of it is 3 floor of it is 2. So, this is this notation this is floor notation and this ceiling notation. Now so, we have seen here that, if  $n$  is the length of the code length of the repetition code then the number of errors that we can correct is up to floor of  $n/2$ . This for repetition code we will also see another very commonly used code called parity check code.

(Refer Slide Time: 20:40)



So, here parity check code. So, here what is done is you take  $n$  minus 1 bits information bits and add 1 bit parity. So, we will have  $n$  bits block out of  $n$  bits  $n$  minus 1 number of bits are information bits and the 1 bit is redundant that depends on all the information bits and it is the parity of the information bits. What is parity? Parity is nothing but count the number of 1s in the  $n$  minus 1 information bits if the number of 1s is odd the parity is 1 if the number of 1s is even the parity is 0.

So, you can get this parity of actually simply by taking XOR of all the bits. XOR  $n$  number of bits is nothing but, it is 1 if the number of 1s is odd and it is 0 number of 1s is even. So, the parity bit can be obtained from the information bits that way just by taking XOR of all the information bits. So, now if we do use this code the what is the code the code now code  $C$  in this case is set of all vectors  $x$  naught to  $x_{n-1}$  set of all vectors are these are binary, these are components are binary.

We will call the set here  $F_2$  we will discuss, we will I will tell you in a moment why we denote it by this symbol. It is a set of 0 and 1. So, these  $x$  naught till  $x_{n-1}$  are from this set each 1 is either 0 or 1. So, take all such binary  $n$  such that the XOR of all these bits is 0. So, take, so now, XOR of all these bits XOR of 2 bits  $x$  naught plus  $x_1$  this XORing operation will be



denoted by plus because it satisfies all the conditions satisfied by the usual plus operation of integers or real numbers.

Similarly, multiplication of 2 bits suggest the AND operation. So,  $x_n$  times  $x_{n-1}$  is AND and this is XOR. So, parity of this is nothing but, summation of  $x_i$  equal to 0 and minus 1. So, this is basically XOR of all the bits. So, this is 0. So, we take all possible n-tuples of bits such that the XOR of all the bits is 0, that means;  $x_{n-1}$  is the parity of all the other bits because if the number of 1s here all the other bits is 1 then  $x_n$ , if the number of 1s is even then  $x_{n-1}$  has to be 0 because the total there are even number of bits and if the number of 1s in this  $n-1$  bits is odd then  $x_{n-1}$  must be 1 to make the total number of 1s even.

So, we see that this condition implies that,  $x_{n-1}$  is the parity of all the other bits and similarly any bit is the parity of all the other bits. So, this is really parity check code we can say that, we take these  $x_n$  to  $x_{n-2}$  as the information bits and the  $x_{n-1}$  bit is the parity bit of all those information bits. So, this is our parity check code now how many errors can we correct actually, we cannot correct any error using this because the distance of the there are too many code words and there is not much difference between the code words.

We see in detail what we mean by this, but we will just discuss the capability of this code here. So, this code can detect at least at most 1 error. What does it mean. If the channel introduces 1 error or no error no error there is no question of detecting. If the channel introduces 1 error we will be able to detect that is no where at the receiver that there is some error in this block. So, how do we know that, we will simply take this received block and see how many 1s we have that is calculate the parity of the block and if it is even that is: if the parity is 0 then there is no error, that is, it is a code word and if the number of bits is number of 1 bit is odd then we know that there is some error because such a code word no code word has odd number of 1s.

So, the what we have received was not transmitted something else was transmitted. So, we will be able to detect 1 error up to 1 error, if there are 2 errors again we may not be able to detect because it may happen that 2 errors are introduced and the parity of the code word is even now 0. That is the number of 1s is still even because 2 bits have changed and it was even number of 1s before.

So, after changing 2 bits also the number of 1 bit will be still even. So, we say here that, it is it can detect 1 error it can detect 1 error. And what is the rate of the code. How many bits of information we are transmitting per bit. We have n number of bits and out of those n number of bits we are transmitting n minus 1 bits of information. The only 1 bit is redundant. So, rate is n minus 1 by n per bit we are transmitting n minus 1 number of n minus 1 by n bits of information because total n minus 1 bits of information is transmitted in n bits.

So, the rate of this code is n minus 1 by n and we can detect 1 error up to 1 error using this code. Now, to discuss these techniques in a more systematic manner we need to introduce some terms, 1 is hamming distance suppose, we have 2 vectors x and y.

(Refer Slide Time: 28:27)

$$\underline{x} = (x_0, x_1, \dots, x_{n-1})$$

$$\underline{y} = (y_0, y_1, \dots, y_{n-1})$$

$$d_H(\underline{x}, \underline{y}) = \text{no. of components that are different in } \underline{x} \text{ \& } \underline{y}.$$

$$\underline{x} = (1001)$$

$$\underline{y} = (1100)$$

$$d_H(\underline{x}, \underline{y}) = 2$$

x is naught x1 xn minus 1, we will assume binary codes all these are binary and y another y naught y1 yn minus 1 2 vectors. Hamming distance between those these 2 vectors denoted by this is defined as the number of components that are different in x and y. So, means we will check whether x naught is same as xy naught, if they are same we do not count that. If x 1 is not equal to y1 we count 1 then we will see how many are different that is the hamming distance between those 2.

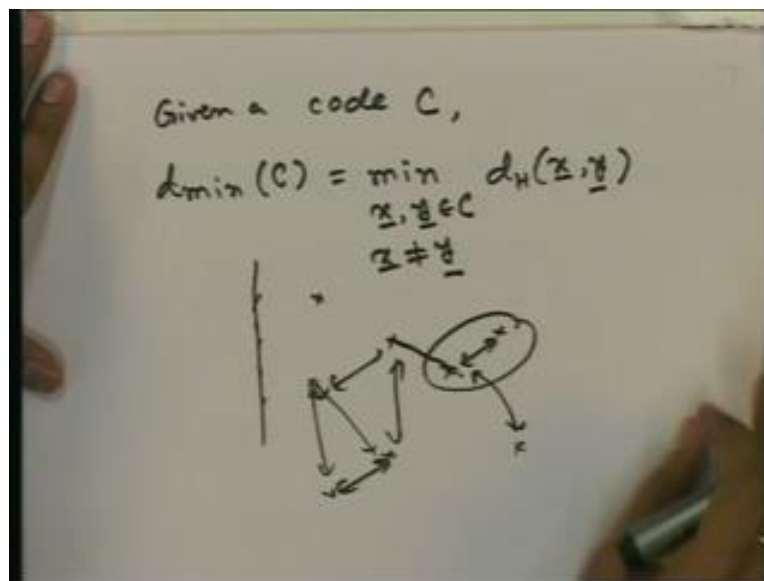
So, that is measure of how different these 2 vectors are if there are more number of bits that are different in these 2 vectors then the they are more different. So, this is called the hamming

distance. Let us take an example suppose  $x$  is 1 0 0 1 1 and  $y$  is 1 1 0 0 1 let us see what the hamming distance. And  $y$  is 1 1 0 0 1 let us see what is the hamming distance of these 2. This first bit is same second bit is different. So, 1 third bit is same fourth bit is different and fifth bit is same. So, we have 2 bits which are different second and fourth. So, the hamming distance is 2.

So, we have seen what is hamming distance between 2 vectors of binary numbers. Now, if you are given a code, given a code what is a code the code a code is a set of  $n$  tuples

A code of length  $n$  is a set of  $n$  tuples of binary numbers 01. So, binary  $n$  tuples set of binary  $n$  tuples not all binary  $n$  tuples, but some of the binary  $n$  tuples. Then given a code what is the minimum distance  $d$  minimum of the code is take all possible vectors take any 2 vectors from  $C$ .

(Refer Slide Time: 30:40)

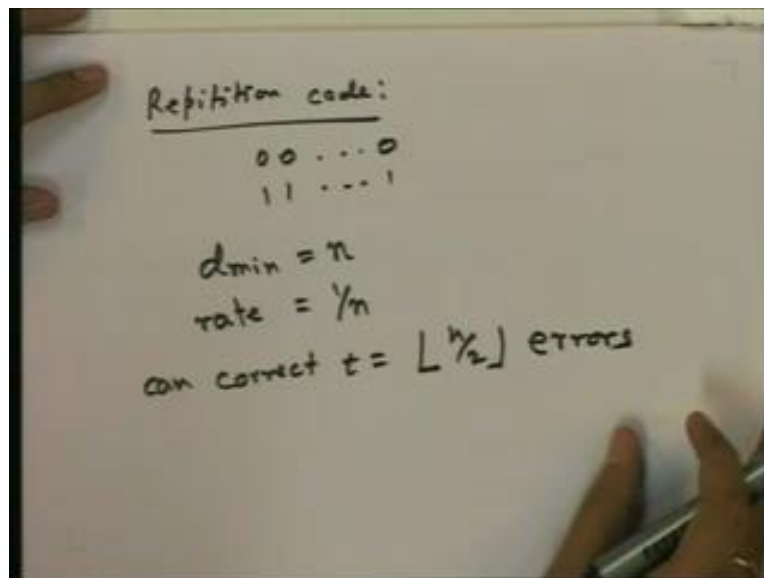


Say  $x, y$  in  $C$  and  $x$  not equal to  $y$  do not take the same codeword for  $x$  and  $y$  take 2 different code words and see what is the minimum distance see what is the distance between  $x$  and  $y$ . We will get some minimum get some distance take some other pair of code words take the distance and that way we take all pairs of code words. And take the distances and take the minimum of all those distances that is the minimum distance of the code. What does it mean. Pictorially it means that in the  $n$  dimensional space. So, you have  $n$  tuples of binary number. So, in that space you have some code words non not all points are codeword not all  $n$  tuples are code words some of them are code words. So, we have some points like this which are code code words.

So, that set is the code there are other points here which are not code words. Now, take this minimum distance means take 2 points which are nearest to each other. So, for example, take this distance you can see the this distance is less than distance between this and this. So, take C, if there are closer points this distance may be nearer the this distance may be smaller. So, take that distance take the minimum of the distances. So, then this distance take all these distances all possible distances then take the minimum that is: the minimum distance of the code.

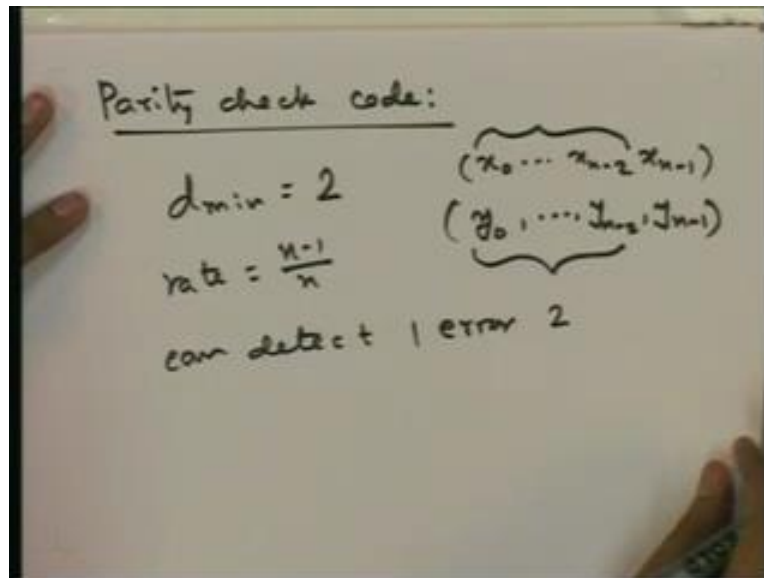
Now, once we have decided once we have defined this term let us see for the 2 codes we have already discussed what is the minimum distance? First repetition code here you take 2 code words they are nothing but 0 and all 0s. And all 1s that is all.

(Refer Slide Time: 33:16)



So, number of bits that are different there is the hamming distance between these those 2 code words is simply n. So,  $d_{min}$  for the repetition code of length n is n. And we have already seen the rate is 1 by n and numbers of errors that we can correct t equal to floor of n by 2 errors. And for parity check code what is the minimum distance it is 2. Let us see, suppose we take any 2 vectors any 2 code words any 2 code words 2 different code words must be different in the first n minus 1 bits itself. So, in first n minus 1 bits x nought xn minus 2 xn minus 1.

(Refer Slide Time: 34:30)



This is 1 code word and we take another code word  $y_0, \dots, y_{n-2}, y_{n-1}$  another vector. What is the distance between these 2 we want to find. The minimum distance will come only when the information bits here are different from information bits here only by 1 bit, if the difference between this and this, the distance between this and this is 1. Then these 2 also will be different because either it has this has if this has even number of 1s. This will have odd numbers of 1s because there is only 1 bit that is different.

So, then this will be 1 and this will be 0 because of parity of this will be 0 and parity this will be 1 So, this bit also will be different. So, the hamming distance will be 2. Similarly, this is odd this even if this is odd and that only 1 bit is different in these 2. Then this will have even numbers of 1s. So, this will be 0 this will be 1 So, these 2 this bits this bit will be different.

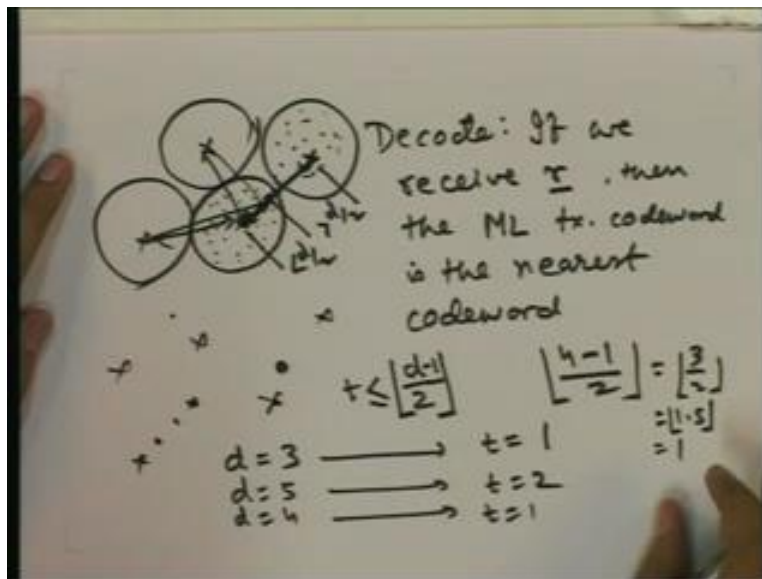
So,  $n$  the distance is 2, but if these  $2, n-1$  tuples till here are different by more than 1 bits. Then we know any way that the minimum the distance between them must be at least 2 because here itself there are 2 bits which are different 2 or more bits which are different. So, if the distance between these 2, if there are more than 1 bits that are different than these 2 then also the hamming distance between these 2 code words will be at least 2. So, as a result we can say that the minimum hamming distance of the code is 2 because there are there are there is a pair of

code words we can find between which the distance is 2. And there is no other pair of code words which has distance more than 2 which has distance less than 2.

So, the minimum hamming distance is 2. And rate we have already seen rate is  $n$  minus 1 by  $n$  numbers of errors we can detect only 1 error cannot correct, Can detect 1 error. So, this is about minimum distance. Now, from if you know the minimum distance of a code can we say how many errors can be corrected. We should be able to do that because minimum distance tells us that the code words in the code are separated by this much distance they are far apart if the distance is large they are far apart.

So, if the if they are some errors then we can still this will not go near other code words. If this is a code word and there are code words far from this. Then if there is some error then this will not the probability that this will go near any other code word will be small because this distance is too much. So, for that we need so, many errors. And that probability is low. So, we will be able to correct more errors if the minimum distance is more, but we can exactly specify how many errors we can correct.

(Refer Slide Time: 38:00)



Let us say these are the code words. And this is a code we transmit this code and suppose, the minimum distance is this, this these 2 these has the minimum distance in the code. So, the what is think in can happen when this code word is transmitted and the number of errors is more the

above this. So, this is the circle in which all the points for all the points here this code is the nearest. This code word is the nearest.

So, how do you decode? Sorry, decode. Take, if you receive  $r$  the vector  $r$  of binary numbers then the maximum likely that is ML transmitted codeword is the nearest codeword that is if you receive this then take the nearest codeword and this is the nearest codeword then most probably this was transmitted. The probability that this was transmitted is less than probability that this was transmitted. So, the maximum likely codeword that was transmitted is the nearest codeword that can be proved also.

So, we see that this is the way to do decoding that is: if you receive a vector here take the nearest codeword nearest in the sense of hamming distance. So, calculate the hamming distance from each codeword and take the nearest 1. So, if we take the nearest 1 then what will happen is that we can form the points from then points from which this codeword is nearest. So, this is the decision region of this codeword that is; if you receive all these vectors then we will decode this vector this codeword.

If you receive all these points inside the this circle this will be the nearest to all these points. So, we will decode this. So, this will be decoding regions. Now, if you do the it that way we can see that suppose, there are less than  $d/2$  number of errors number of errors is less than  $d/2$ . Then this distance from what you receive is less than  $d/2$ . So, distance to this which is which has distance  $d$  from here this distance will be greater than  $d/2$  because this is less than  $d/2$  this extra will be greater than  $d/2$  because this total is  $d$ .

So, this much is less than  $d/2$ . So, this much must be greater than  $d/2$ . So, we see that be and any other codeword you take this distance is at least  $d$  where  $d$  is the minimum distance. So, again this distance from here to all these distances will be greater than  $d/2$  that is greater than here this distance. So, this will be the nearest. So, we have seen that in the number of errors is less than  $d/2$  where  $d$  is the minimum distance of the code then we will be able to decode the codeword correctly by decoding this way taking the nearest codeword we will take the correct codeword.

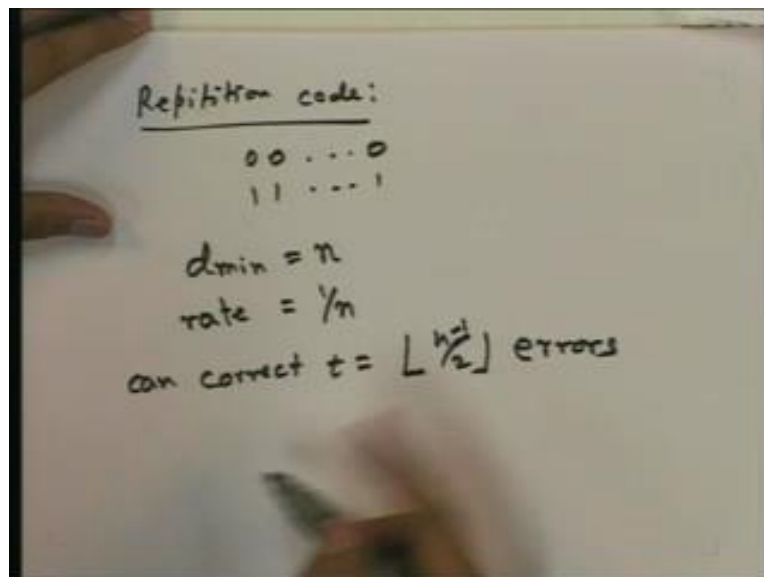
So, we have seen in other words that, if we have a code with minimum distance  $d$  then we can correct all the errors, if the number of errors is less than  $d/2$ . So, how many errors can you

correct take  $d$  by 2 take the integer part of  $d$  by 2 that is floor of  $d$  by 2 and all errors number of errors less than equal to floor of  $d$  by 2 can be corrected because these numbers are less than  $d$  by 2 less than equal to  $d$  by 2. This is actually  $d$  minus 1 by 2 we want it to be strictly less than  $d$  by 2. So, we need to add here subtract minus 1. So, there we can put equality here.

So, using this concept now, if we have a code with  $d$  equal to 3 minimum distance equal to 3 then how many errors can you correct, we can correct 1 error. If you have  $d$  equal five how many errors can you correct, we can correct 2 errors. If we have  $d$  equal to 4 how many errors can you correct 4 minus 1 by 2 floor of this. So, this is 3 by 2 floor that is 1.5 floor that is 1. So, we can still correct  $t$  equal 1  $d$  equal to 3 also we will give us in this formula will give us 1 error.

So, we can see if the distance is at least 3 if they are 2 code words at distance 3 if there is 1 error it will come here, but distance from here is 2 1 2. So, this will be still nearest. So, you will be able to correct the error successfully. Now, according to this we see that the repetition code which has repetition code has distance we said it has distance 2 So, it has distance repetition code has distance  $n$ . So, we can see that we can correct is not  $n$  by 2, but  $n$  minus 1 by 2 according to this formula.

(Refer Slide Time: 43:41)



And that is correct, if  $n$  is for example, even like we consider 4 we will be able to correct only 2 errors not  $n$  by 2. So,  $n$  minus 1 by 2 floor number of errors. Similarly, for parity check codes



what was the distance was 2. So, we cannot correct any error because 2 minus 1 by 2 floor is half floor that is 0. We can correct 0 errors, but we can detect 1 error.

(Refer Slide Time: 44:32)

$\{0, 1\} \rightarrow \mathbb{F}_2$   
 $+ \rightarrow \text{XOR}$   
 $\cdot \rightarrow \text{AND}$   
 Hamming code:  $(7, 4)$   $H \begin{bmatrix} \end{bmatrix} = \underline{0}$   
 $n=7$   
 $H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix}$   
 $C = \{ \underline{x} = (x_0, \dots, x_6) \mid H \underline{x}^T = \underline{0} \}$

Now, we have just now said that we will denote we will denote a denote the set 0 and 1 as  $\mathbb{F}_2$  why because this is the notation for a field. We have discussed already about field in a different module of this course that this has 2 operations plus and multiplication, addition and multiplication so, that it satisfied certain conditions just like: real numbers and complex number, but this field is not like real number for complex number this is a finite set, but it is still satisfies some conditions which are nice.

So, in this field now we are considering codes with components from this field. So, we can do still addition, multiplication, addition is addition is nothing, but XOR multiplication is nothing, but AND under these 2 operations this satisfies nice conditions like: real numbers and complex numbers. We can have vectors we can do matrix multiplication and all that can we done with this additional multiplication. So, the code defined this code we are considering is called hamming code. And the code is constructed this way we have considering particular code called a 7 4 hamming code we will see why it is called 7 4 length is 7 n equal to 7 and it is defined this way take a matrix H which has all non 0 3 tuples.

So, take 1 0 0 0 1 0 0 0 1 then 1 1 0 1 0 1 0 1 1 then 1 1 1. There are 7 columns all possible non 0 3 tuples of binary numbers is they are in the columns. So, there are total 2 to the power 3 that is eight 3 tuples out of them 1 is all 0 that is, excluded all the other are here as columns of H. Now, the code is defined as take all vectors of length 7 that is,  $x$  is  $x$  naught to  $x$  six such that  $Hx^T$  consider,  $x$  as a vector write it this way.

So, this will be a column vector and then  $Hx^T$  is this then  $H$  that should be 0 vector. This is again a column vector of length 3 This is 0 vector of length 3 take all such vectors such all those vectors which we multiply this way will you give you 0. So, those vectors from a code and this code is called is hamming code hamming 7 4 hamming code. Now, we will see that, this code can correct 1 error. How. Suppose, we have transmitted  $x$  it is code word it is in the code  $C$ .

Now, we have suppose, 1 error if there is no error then this will receive the same codeword. So, we can we need not correct an error, but if there is 1 correct error then also we will see that, we will be able to correct. How do you do that suppose we receive an error where the third bit is in error.

(Refer Slide Time: 47:40)

Handwritten mathematical derivation on a whiteboard:

$$\underline{x} \in C$$

$$\underline{y} = \underline{x} + (0010000)$$

At the receiver,

$$H\underline{y}^T = H(\underline{x}^T + (0010000)^T)$$

$$= H\underline{x}^T + H \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$= 0 + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Then we will receive this we will receive  $x$  plus 0 0 1 0 0 0  $x$  plus these vector because this plus remembered is XOR. So, the first bit the this addition the first component of this will be first component of this  $x$  naught plus 0  $x$  naught XOR 0 is  $x$  naught itself, if you XOR with any bit

XOR 0 with any bit it will give you same weight. So, 1s do not change the components where as 1 changes the component So, that particular the third bit will be in error because we have added 1 here So, we can represent the received bit as  $x$  the transmitted sequence received code transmitted code word plus a vector like this, if there is only 1 error.

So, now we will at the receiver, will do this operation we have received  $y$  we will simply multiply  $y$  transport with transport with  $x$ . So, we can write it as  $x$  plus  $0\ 0\ 1\ 0\ 0\ 0\ 0$  then  $H$  this can be written as this transpose, this transpose. So, plus  $H\ 0\ 0\ 1\ 0\ 0\ 0\ 0$ . Now, this is 0 because we have transmitted  $x$   $x$  is a code word. So, it satisfied  $H\ x$  transpose equal to 0 we have transmitted such a codeword such a vector. So, this is 0 plus now what is  $H$  times this column vector column vector has 1 only in the third column third row.

So, it will the result of multiplying  $H$  with  $0\ 0\ 1\ 0\ 0\ 0\ 0$  will be  $0\ 0\ 1$  the third column of this please check it afterwards, but you can see that if you multiply any matrix by a vector like that  $0\ 0\ 1$  and then all again 0 then it will pick up the third column. So, the result will be  $0\ 0\ 1$  this will  $0\ 0\ 1$ . Now, once we take this we will take this result this result will be  $0\ 0\ 0\ 1$ . If the fourth bit was in error the result will be  $1\ 1\ 0$ . If the sixth bit is in error we will have  $0\ 1\ 1$  as this multiplication. So, we will just take this result and see which column of  $H$  it is, if it is 0 then we know that it is a codeword the  $y$  itself is codeword.

So, in there is no error. If there is this is not 0 then we will see where which column of  $H$  it is because all the non 0 columns are here all the non 0 3 tuples are here. So, we will pick that particular column and we will know that that bit was in error. So, if this was here we will see that this is the third column. So, we know the third bit is in error  $x\ 2$  is in error. So, we will invert that and that is our codeword that was the thing which was transmitted. So, we will know which bit was in error this way and then we can correct that bit.

So, this is way the hamming code can be used to correct up to 1 bit. So, we have discussed so, for 1 error detecting code where up to 1 error can be detected, but not corrected that is the parity check simple parity check code. Here in the hamming code seven 4 hamming code we have seen that some we have seen this the this is a code which can be used to correct up to 1 error. So, hamming distance of this code we can guess that hamming distance because numbers of errors that can be corrected is 1 and we can check that actually 2 errors cannot be corrected.

So, 1 is the number bits that can be corrected. So, 1 is greater than equal to  $d$  minus 1 by 2 floor.

(Refer Slide Time: 51:45)

$$1 \leq \left\lfloor \frac{d-1}{2} \right\rfloor$$
$$\Rightarrow d \geq 2 \cdot 1 + 1 = 3$$

Hamming distance = 3  
dimension of  $C = 4$   
no. of codewords =  $2^4$   
rate =  $\frac{4}{7}$   
 $d_{\min} = 3$ ,  $t = 1$

So, the from here you will get  $d$  is  $d$  this is less than equal to  $d$  is greater than equal to 2 times 1 plus 1 that is 3. So, it hamming distance of hamming code the minimum hamming distance is greater than equal to 3, but again we can check very easily that it is actually equal to 3 So, we cannot correct more error that is obvious from here to correct 2 errors we need minimum distance at least 5. So, for 3 we can correct exactly 1 error at most 1 error.

Now, how many code words are there in the code. We see here the code is the all code words or all vectors which 1 you multiply with  $H$  gives you 0.

(Refer Slide Time: 52:58)

Handwritten notes on a whiteboard:

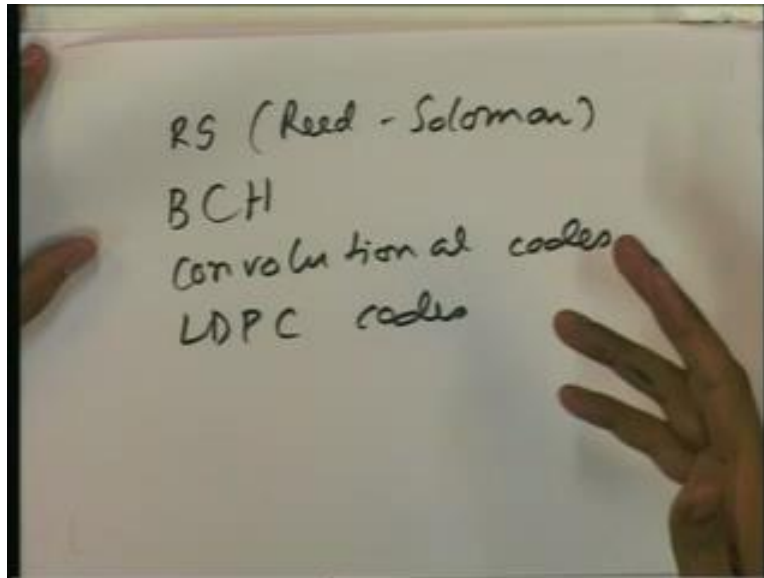
- $\{0, 1\} \rightarrow \mathbb{F}_2$
- $+ \rightarrow \text{XOR}$
- $\cdot \rightarrow \text{AND}$
- Hamming code:  $(7, 4)$   $H \begin{bmatrix} \end{bmatrix} = 0$
- $n = 7$
- $H = \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix}$
- $C = \{ \underline{x} = (x_0, \dots, x_6) \mid \underline{x}^T H = 0 \}$

So, how many solutions of this matrix are there that is the question those many code words are there. So, you can see from here the rank of this matrix is 3 because there is a identity matrix here and there are 7 here. So, the dimension of the solution will be 4 and; that means, there will be 2 to the about 4 number of code words. So, if you do not understand what is dimension of the solution, how to find that it is basically number of the dimension of the solution space will be the n minus the rank 3.

So, dimension is 4. So, dimension of C is 4. So, number of code words is 2 power 4 because it is. So, it is like set of all 4 tuples. So, we have 2 to the power 4 number of codes the rate of the code. So, we can transmit 4 bits using those 7 bits. So, rate of the code is 4 by 7. So, this 4 is the dimension is the parameter which we said here 7 4 7 is the length and 4 is the dimension. So, rate of the code is 4 by 7 d min as you said before is 3 number of errors that can be corrected is 1.

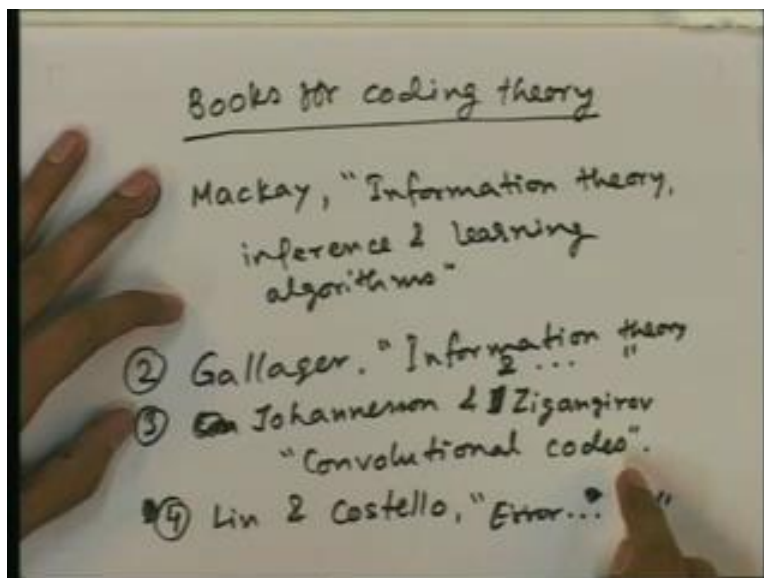
So, this is all about hamming code for in this course, we will not discuss more and there are some other names of course, I will just mention if you are interested you can study from books later, but background regard for that is more than what we can handle in this short span of this course.

(Refer Slide Time: 54:44)



So, most of very popular codes are like RS that is Reed Solomon code. BCH and then convolutional codes LDPC code, these are now, all these codes we have discussed in this class and also the codes which we have not discussed, but just mentioned in the last slide that is, Reed Solomon code, BCH code, convolutional codes and LDPC codes. These can be found in different books for all the basic material and coding theory that we have discussed in this class and.

(Refer Slide Time: 55:26)



we can look at any information theory book like: Mackay, Gallager which we have also mentioned before while discussing source coding. And you can also look at the basic introduction chapter of Johannesson and Zigangirov title is: Convolution codes and. Then there is book on coding theory error correcting codes the title is error correcting codes or something similar and the authors are Lin and Costello.

So, in this book you can find more on coding theory like RS code, BCH code, convolutional code and LDPC codes those can be found in this book all and this is specially for convolutional codes, but the introduction chapter of this will give us some block codes also. That's all Please, I will encourage you to read these books these are very good books.

Thank you.