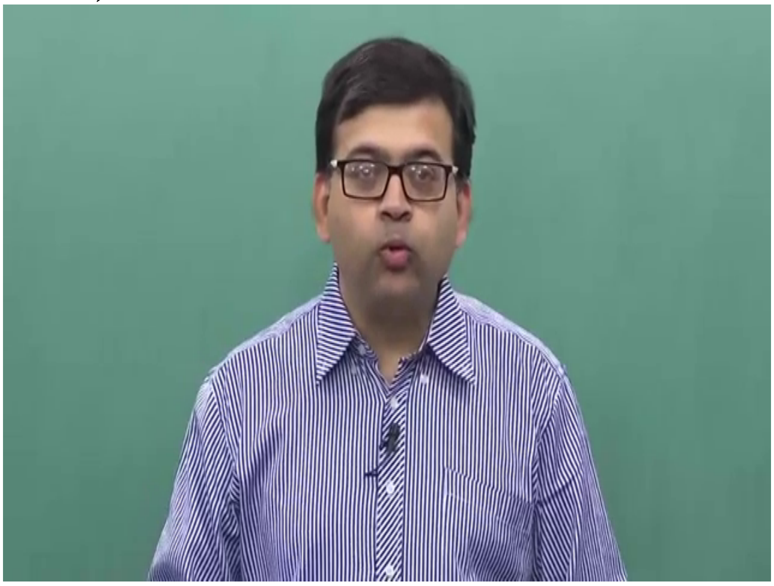
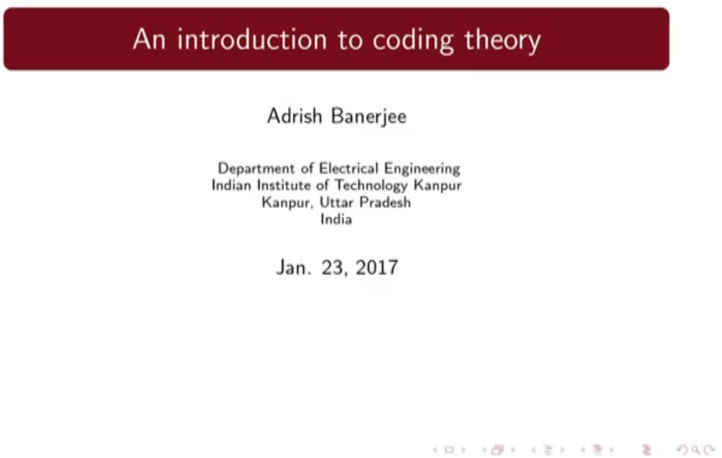
**An Introduction to Coding Theory**
**Professor Adrish Banerji**
**Department of Electrical Engineering**
**Indian Institute of Technology, Kanpur**
**Module 01**
**Lecture Number 01**
**Introduction to Error Control Coding-I**

(Refer Slide Time 00:13)



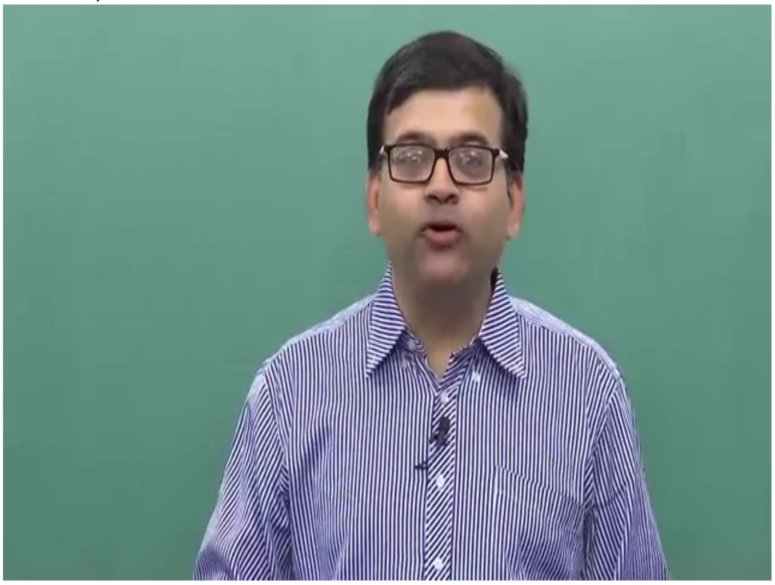
Welcome the course on Coding Theory.

(Refer Slide Time 00:15)



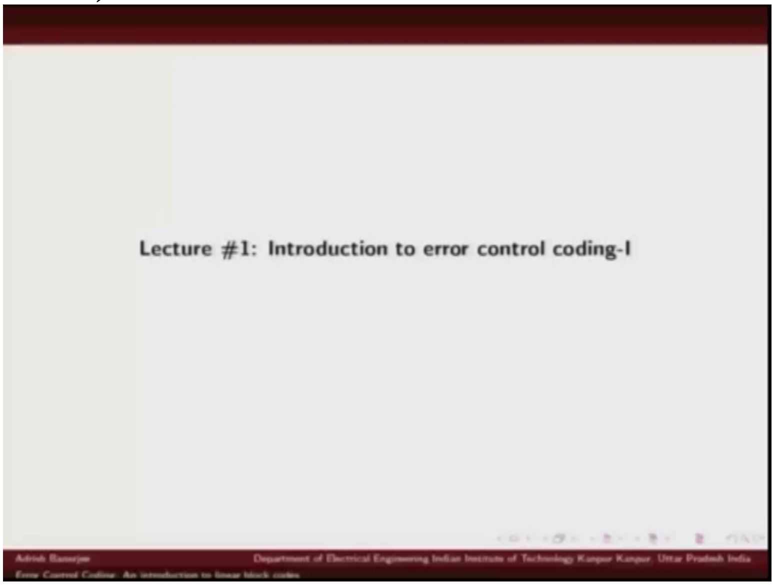

I am Adrish Banerji from I I T Kanpur. Today we are going to talk

(Refer Slide Time 00:19)



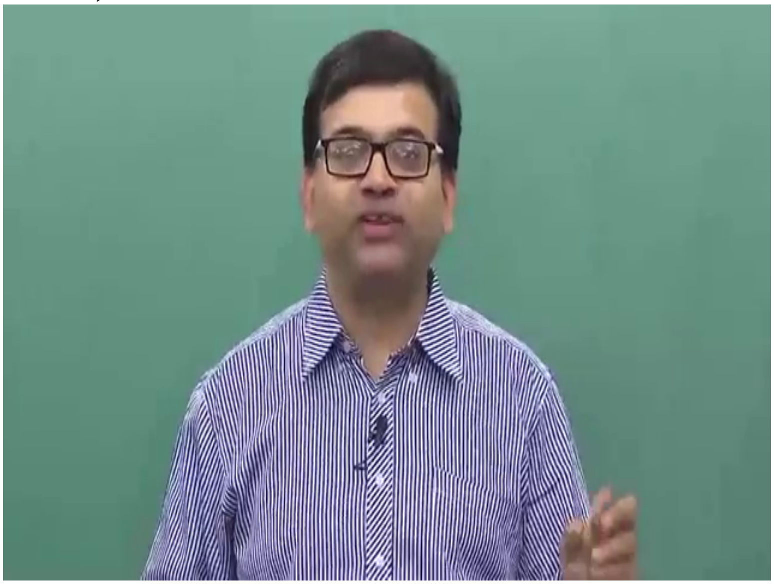
about basic introduction to what coding theory is all about So we will start our lecture.

(Refer Slide Time 00:27)



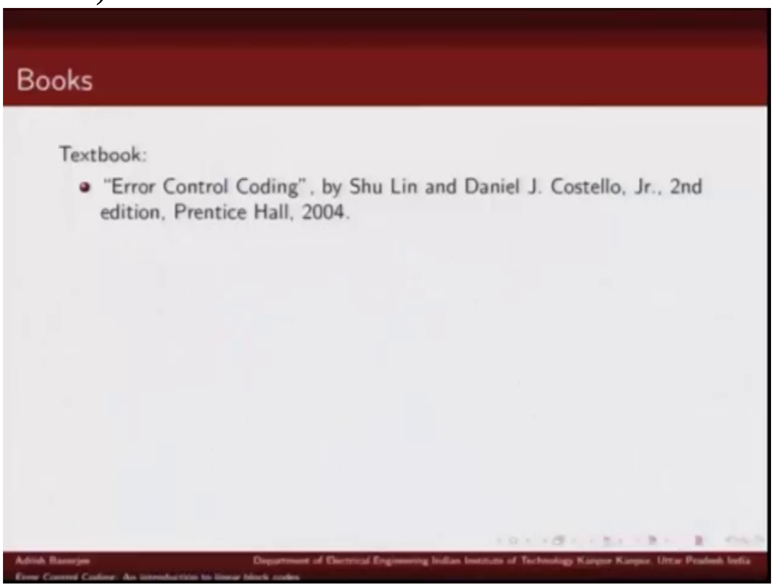

So in introduction, as I said, we will talk about what is coding theory, we will illustrate with a very

(Refer Slide Time 00:34)



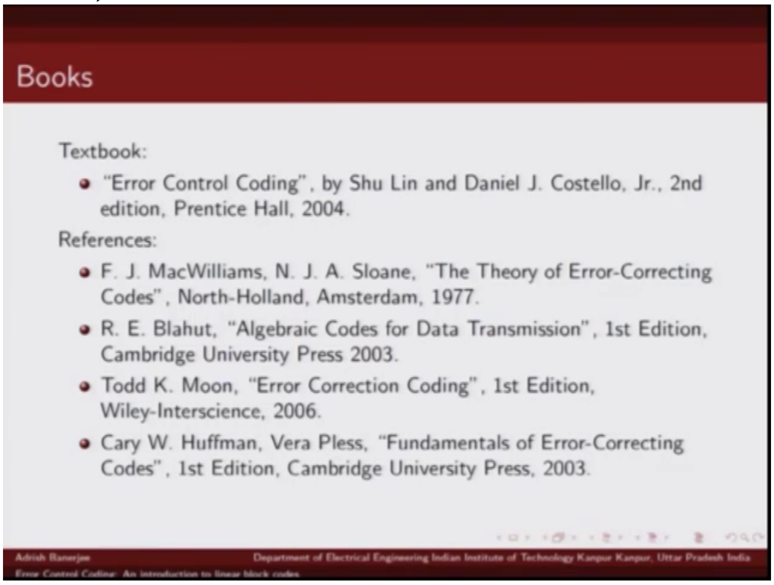
simple example how error correcting codes can be used for error detection and error correction Before I start my lecture, I would like to talk about the books that we are going to use for this course, so we are going to

(Refer Slide Time 00:52)



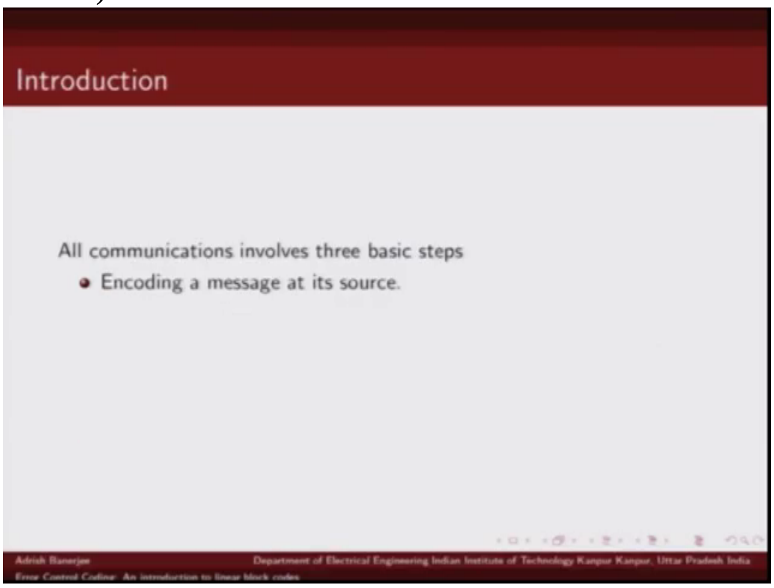

follow this book Error Control Coding by Lin and Costello. It’s a second edition of this book. We are going to follow this book as our textbook. And there are some very nice books which

(Refer Slide Time 01:07)



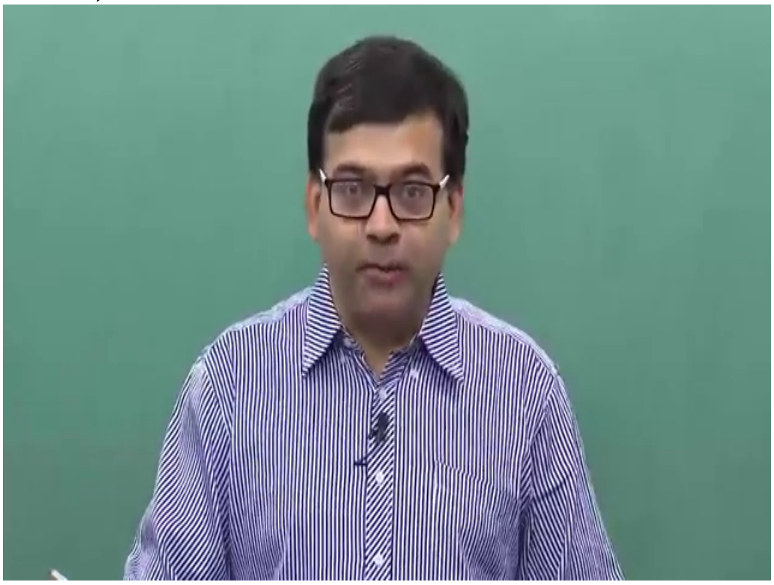

you can use as reference book, for example this book by Sloane and McWilliams is a very nice book on block codes. You could also follow this book by Blahut on Algebraic Codes for Data Transmission, this book Error Control Coding by Todd K. Moon, this also gives a very nice introduction to error correcting codes or you could use this book by Huffman and Pless called Fundamentals of Error-Control Codes.

(Refer Slide Time 01:39)




So when we talk about

(Refer Slide Time 01:41)



communications, communications basically involves three basic steps. The first is encoding a message. You have a message

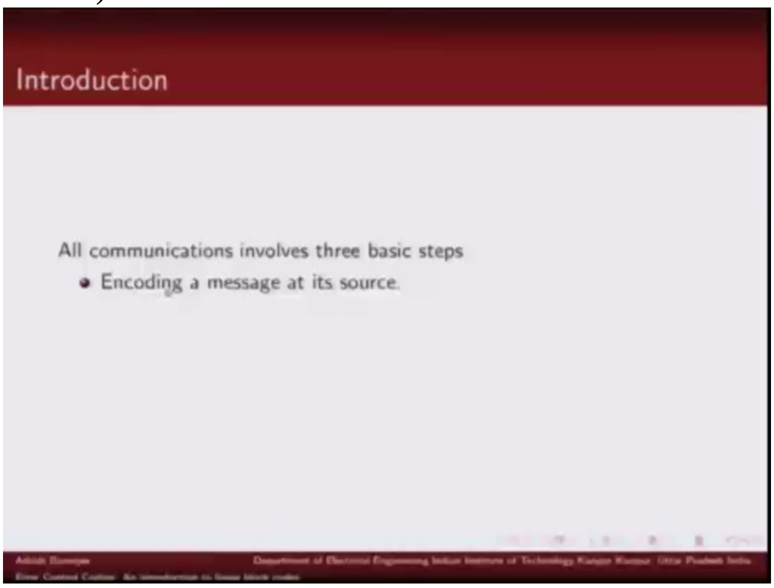
(Refer Slide Time 01:50)


source that you want to represent efficiently

(Refer Slide Time 01:54)

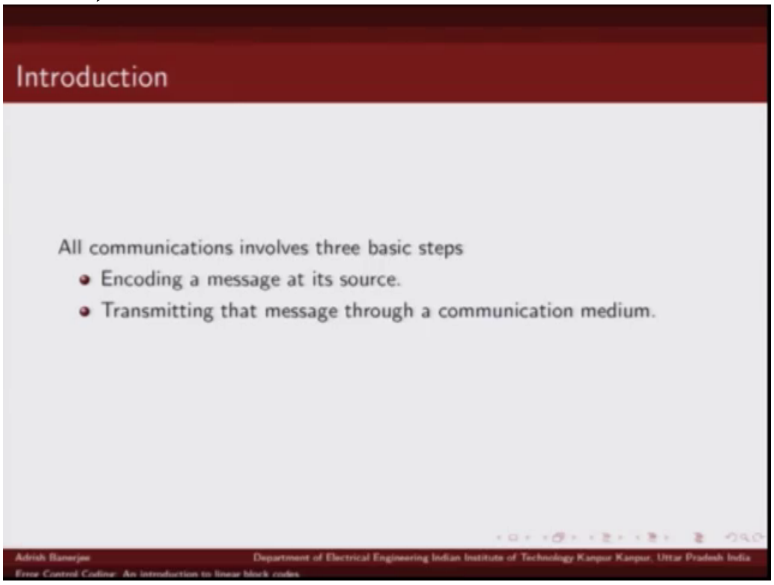
Now you can for example consider a speech signal. Now if you want to transmit a speech signal you first have to convert analog signal to digital signal and then you need to get rid of useless redundancies. Why, because we want to transmit basically useful information. We want a, source inherently has lot of redundancy and when we try to represent a source, we would like to represent a source efficiently in minimum number of possible bits. So first step involved in any communication

(Refer Slide Time 02:31)



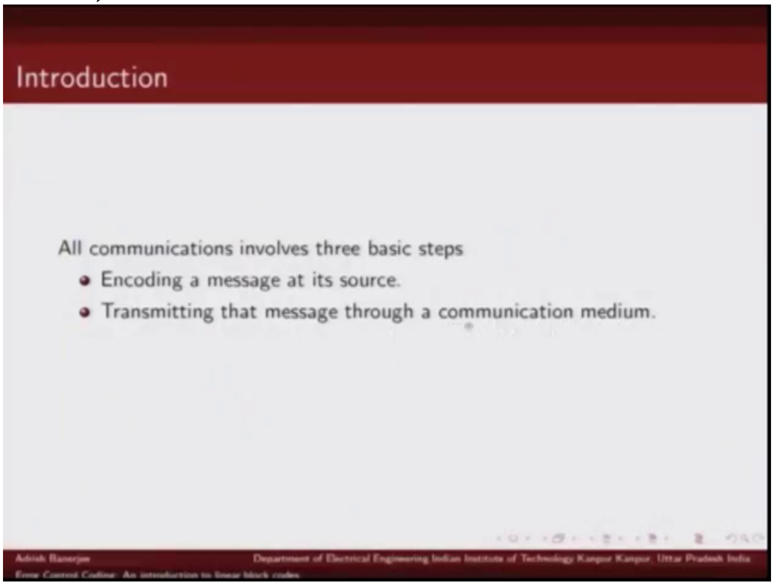

is basically encoding a message Second is, once you have

(Refer Slide Time 02:36)



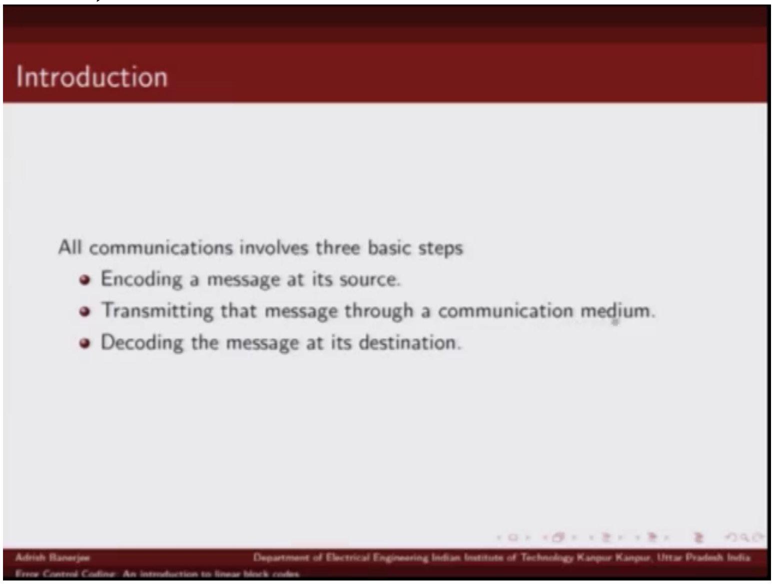

represented your source you want to transmit that source over a communication channel. So the second thing

(Refer Slide Time 02:43)




is transmission of the message through a communication

(Refer Slide Time 02:47)

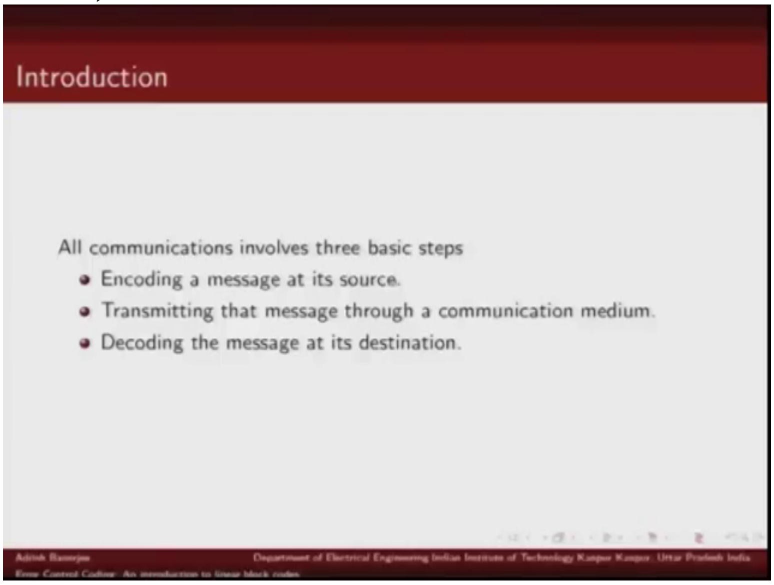



channel and finally once the receiver received the message, it has to decode to find out what was the information that was transmitted. So broadly there are three steps

(Refer Slide Time 03:01)

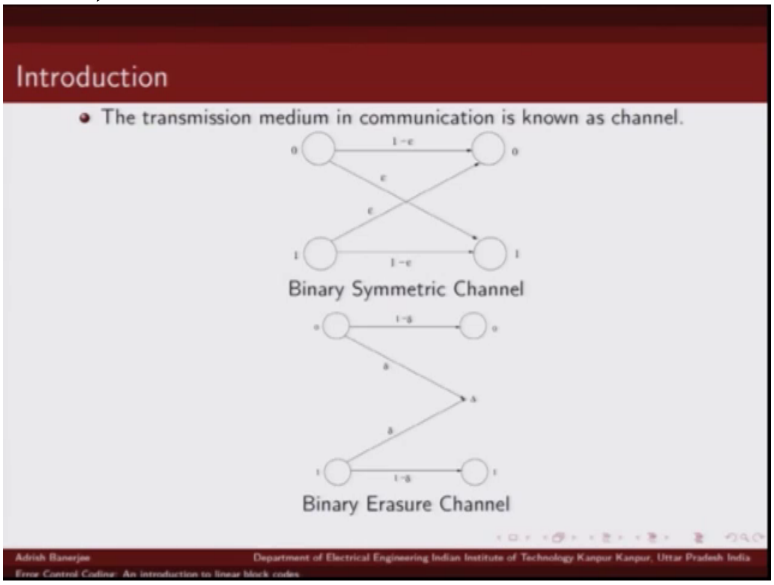
involved in communication; encoding, transmission and then finally decoding So information theory basically

(Refer Slide Time 03:10)



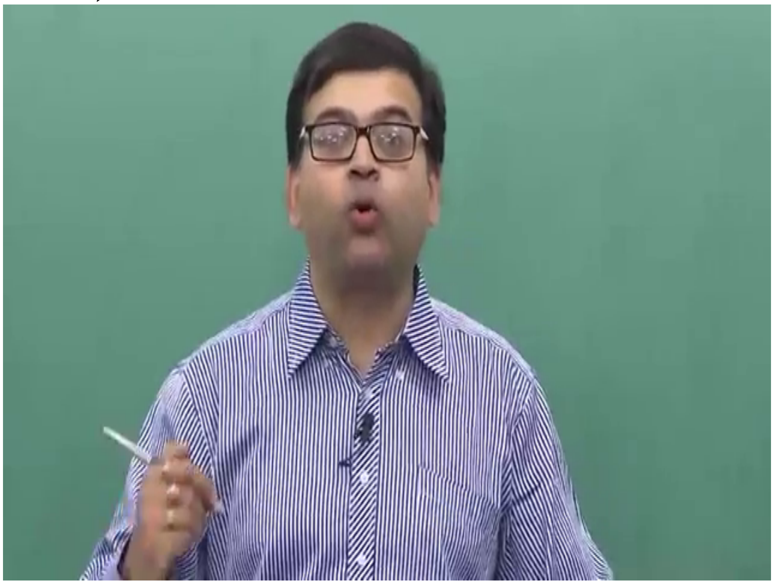

gives us fundamental limits of what is a maximum limit of like what is the best compression, fundamental limit and compression that we can achieve. It also gives us fundamental limits on what is the maximum transmission rate possible over a communication channel.

(Refer Slide Time 03:31)



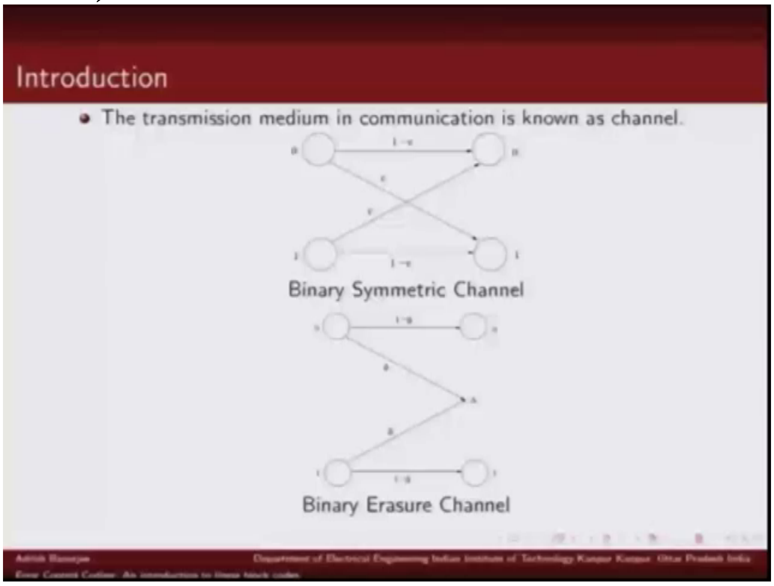

So let's spend some time on what is our transmission medium. So the transmission medium over which

(Refer Slide Time 03:38)



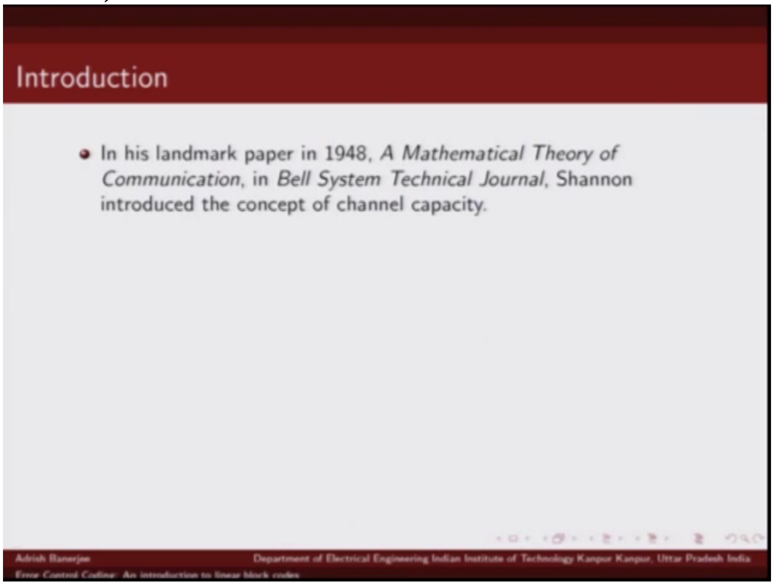
we want to send a packet, that is known a channel and here I have illustrated

(Refer Slide Time 03:44)



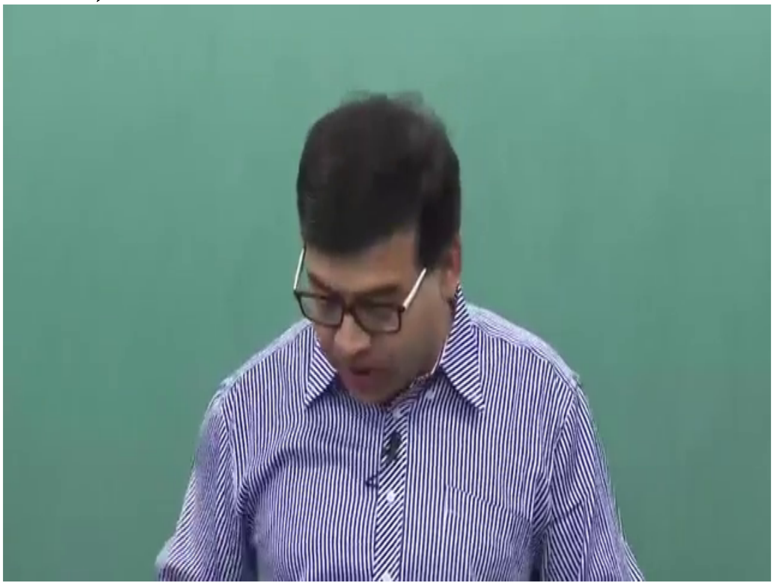

two very simple channel models, the first which is binary symmetric channel; now you can see there are, it has binary inputs, zeros and one and similarly it has binary output 0 and 1. Now with probability 1 minus epsilon, basically whatever you transmit is received correctly at the receiver. So this is a transmitter and this is a receiver side. So if you transmit 0 with probability 1 minus epsilon, you will receive it correctly. Similarly if you transmit 1 with probability 1 minus epsilon you will receive it correctly. And this crossover probability of error is basically given by epsilon. So this is basically a symmetric channel and it's a binary channel because the binary input binary output, it's known as binary symmetric channel. Another channel which basically is very commonly used to model packet data networks is what is known as binary erasure channels. So they are binary inputs, 0s and 1s and the outputs are either you see whatever has been transmitted you receive it correctly or whatever you have transmitted is basically erased. So this delta that you see basically, we are denoting an erased bit using this symbol. So with probability 1 minus delta you receive the bit correctly and with probability delta the bit is erased or lost.

(Refer Slide Time 05:17)



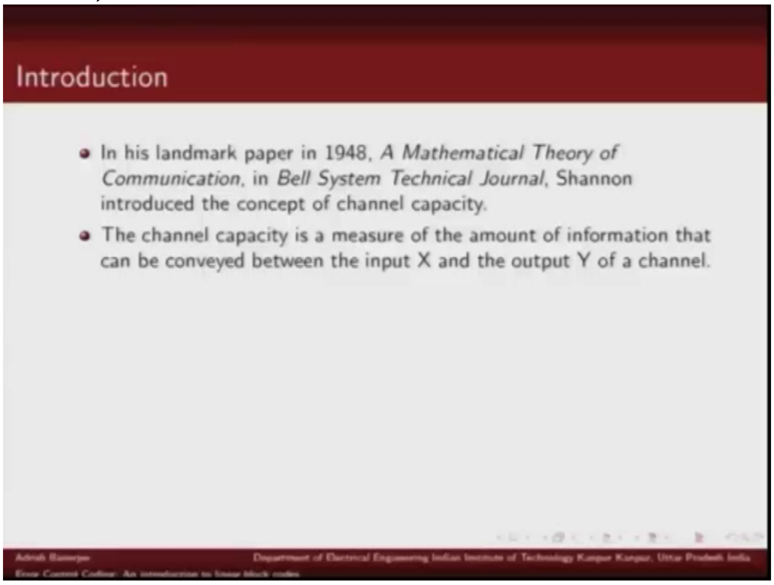

So in his landmark paper in 1948, Shannon introduced this concept of channel capacity, that what is the maximum rate at which we can communicate

(Refer Slide Time 05:27)



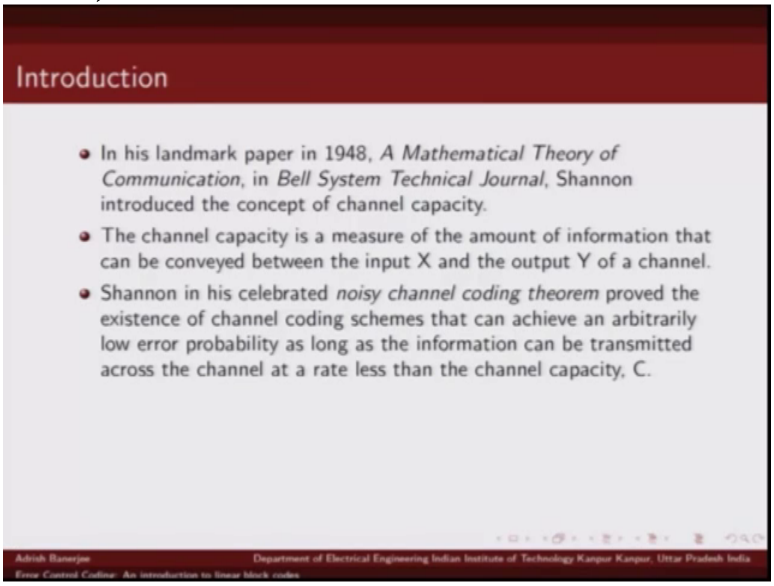
over a communication link; so a channel

(Refer Slide Time 05:31)



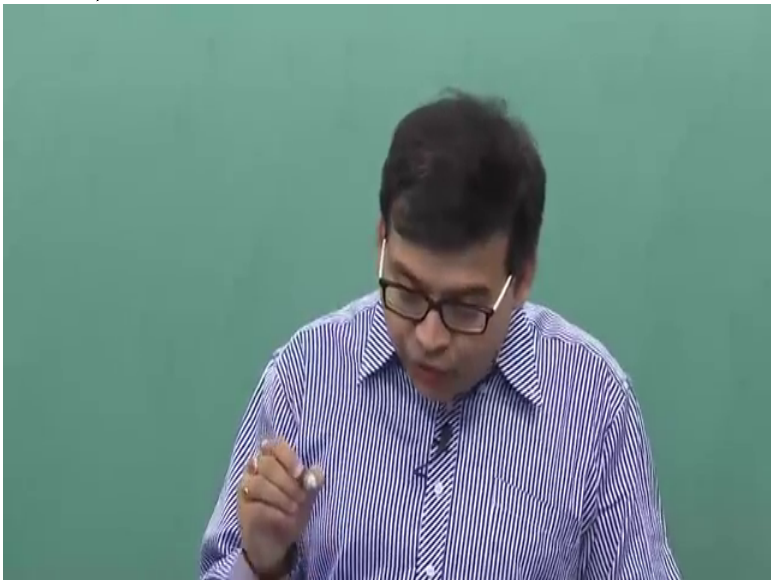

capacity is defined as the maximum amount of information that can be conveyed from the input to the output of a channel. Shannon

(Refer Slide Time 05:40)



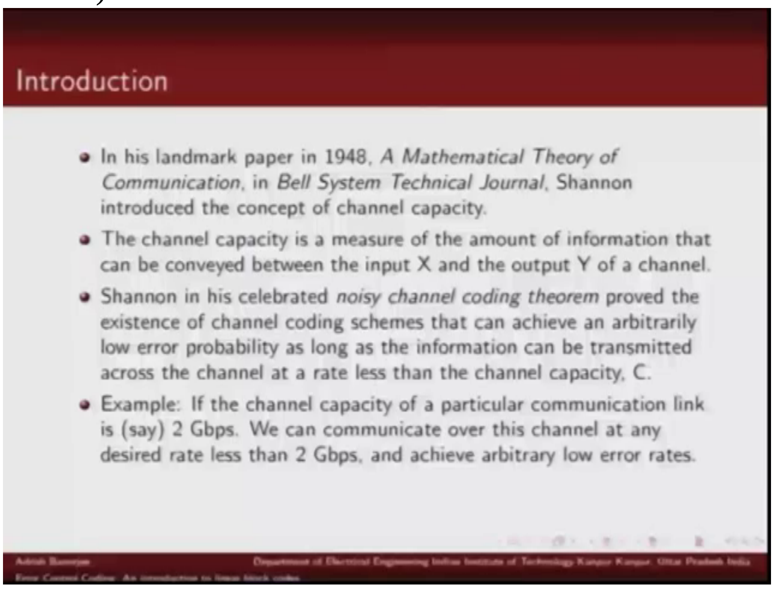

in his theorem also proved that there exist channel coding schemes that can achieve very low, arbitrarily very low probability of error as long as the transmission rate is below channel capacity. So Shannon showed

(Refer Slide Time 06:04)



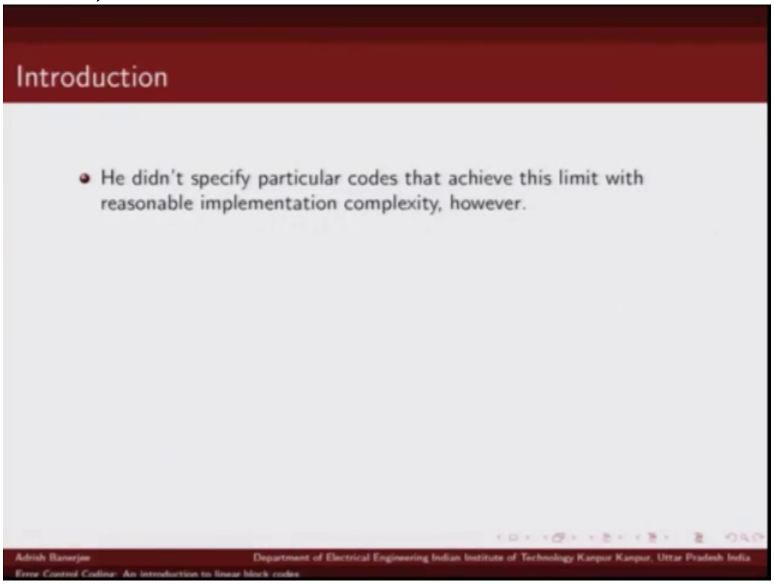
that there exist good channel codes as long as the transmission rate is below channel capacity we can achieve arbitrarily low probability of error. For example

(Refer Slide Time 06:16)



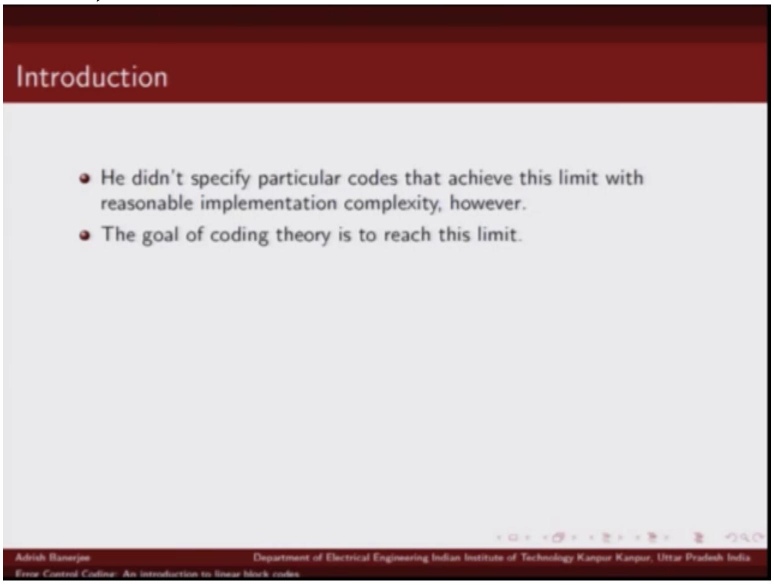

if we talk of a channel capacity of a particular link to be two Giga bits per second then basically we should be able to communicate at rate, any rate up to two Giga bits over this communication link without basically, and can achieve very low probability of error at the decoder.

(Refer Slide Time 06:35)



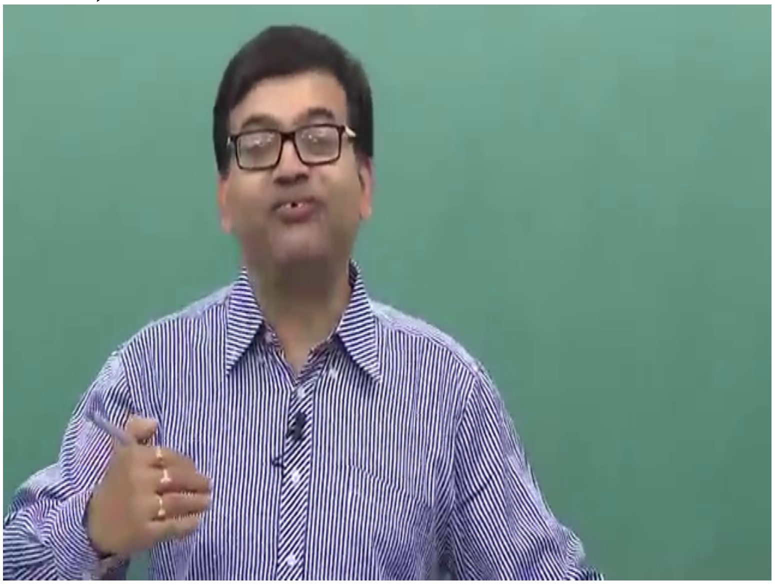

Now in this theorem Shannon did not specify how to design such codes which have rate close to capacity and that's where basically error control coding comes into picture. So the goal of error correcting

(Refer Slide Time 06:53)



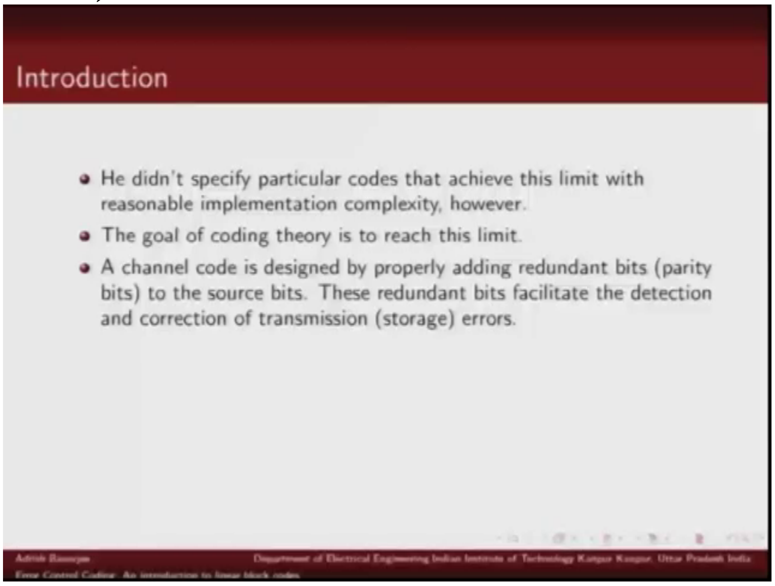

So the goal of error correcting coding theory is to achieve this, to design

(Refer Slide Time 06:57)



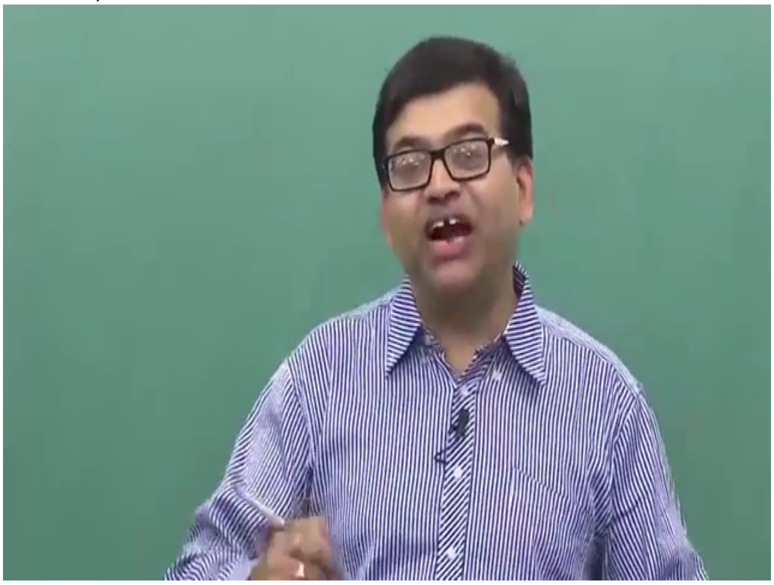
codes which can achieve this limit so basically Shannon has mentioned that we could transmit, we could design, as long as we design error correcting codes which have rate less than channel capacity, we can achieve arbitrarily low probability of error. So the goal of the coding theory or the error control coding is to design such error correcting codes with rates as close to capacity which can achieve arbitrarily low probability of error. And Shannon

(Refer Slide Time 07:32)




did not specify how to design such code; so basically where coding theories come into picture. So how do we design an error correcting code? An error correcting code is designed by adding some redundant bits to the message bits. Message bits, we call them information bits and those additional redundant bits that we add, those are known as parity bits. So error correcting code is designed by properly adding some redundant bits to your message bit and then send this coded message over a communication link. Now we use these additional redundant bits

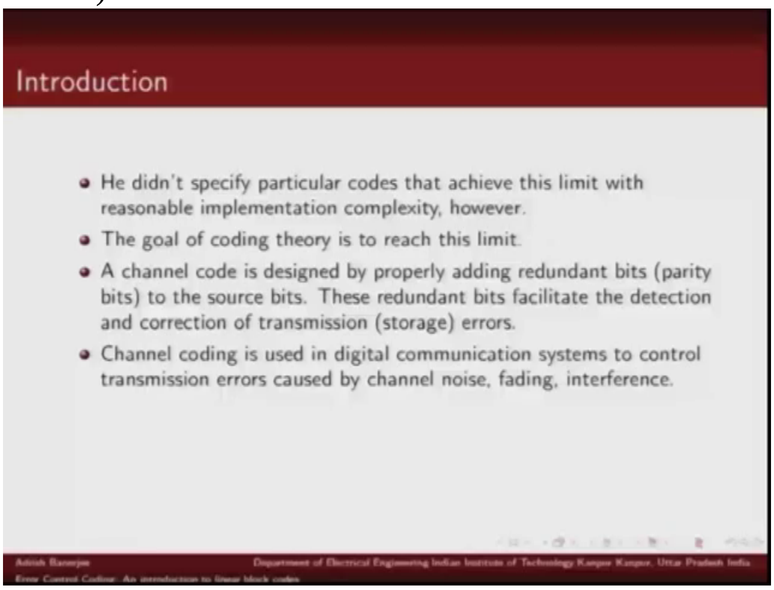
(Refer Slide Time 08:15)



to detect error and correct error

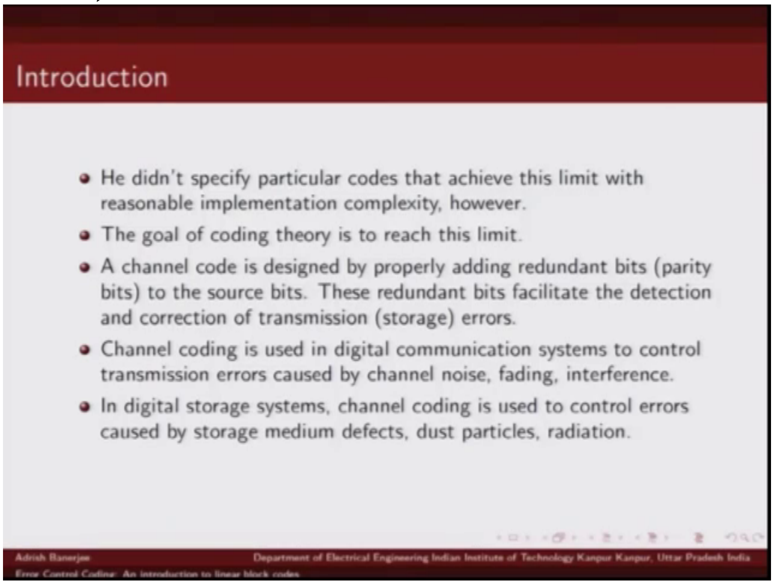
Error correcting

(Refer Slide Time 08:22)



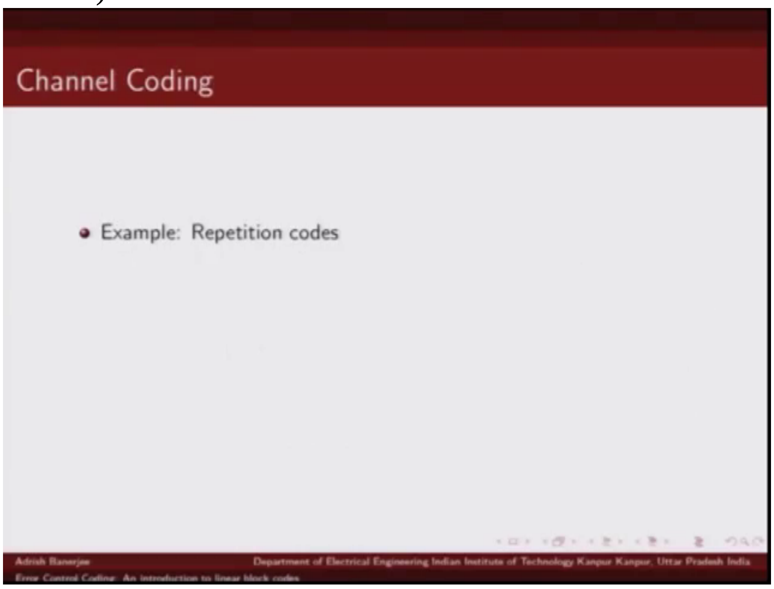

code has wide range of applications in digital communications and storage. I have listed few of the uses, for example when we send a signal over communication link; it gets corrupted by noise, fading, interference. So to combat the interference of all these basically we use the error correcting codes to correct the errors. Similarly

(Refer Slide Time 08:47)



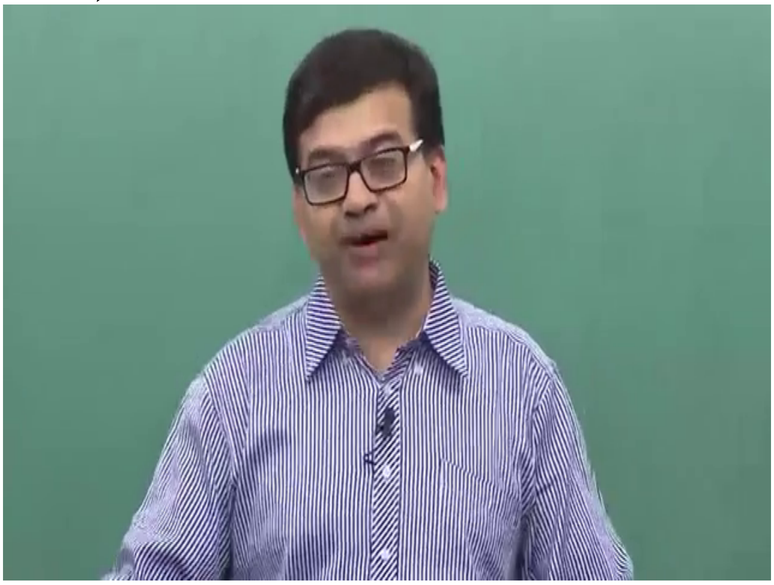

in digital storage system you want to correct the error caused due to storage media defect, dust particles, radiations we use error correcting codes there.

(Refer Slide Time 09:01)



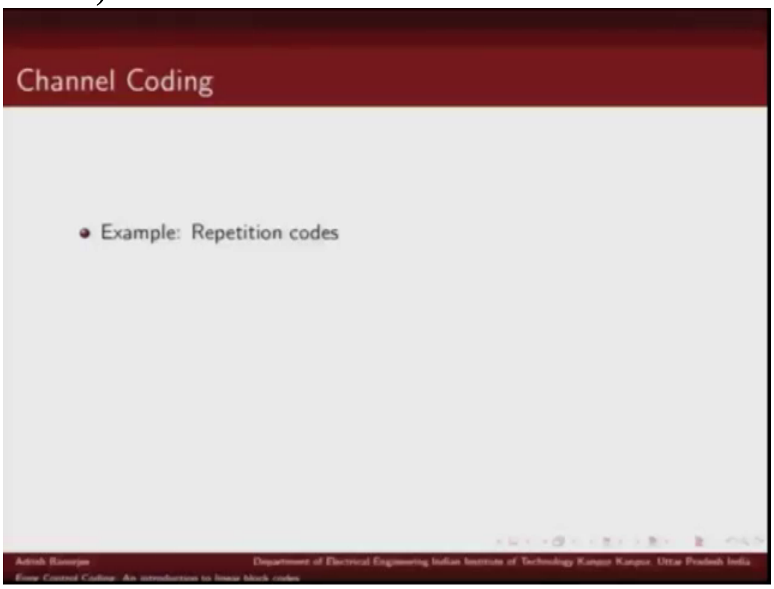

So let us take a very simple example of

(Refer Slide Time 09:05)



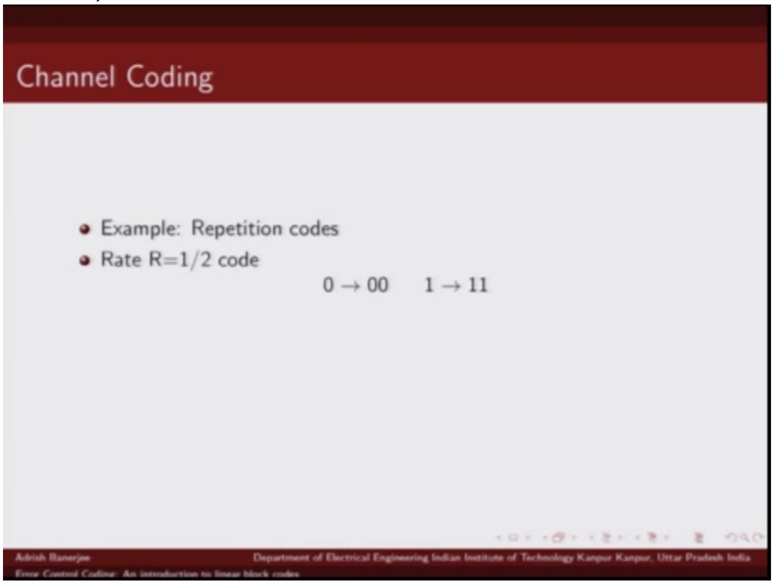
error correcting codes and illustrate

(Refer Slide Time 09:09)



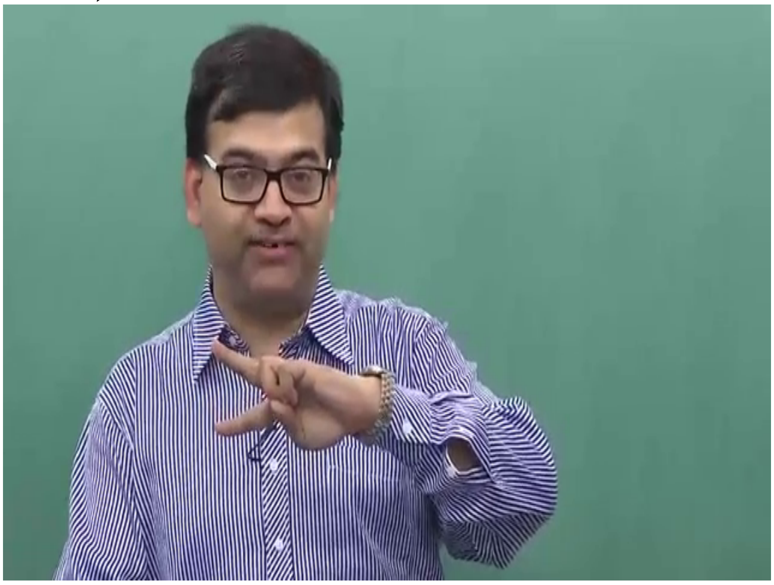

how we can use error correcting codes to detect and correct errors. So example I am going to show you right now is of what is known as repetition code.

(Refer Slide Time 09:25)



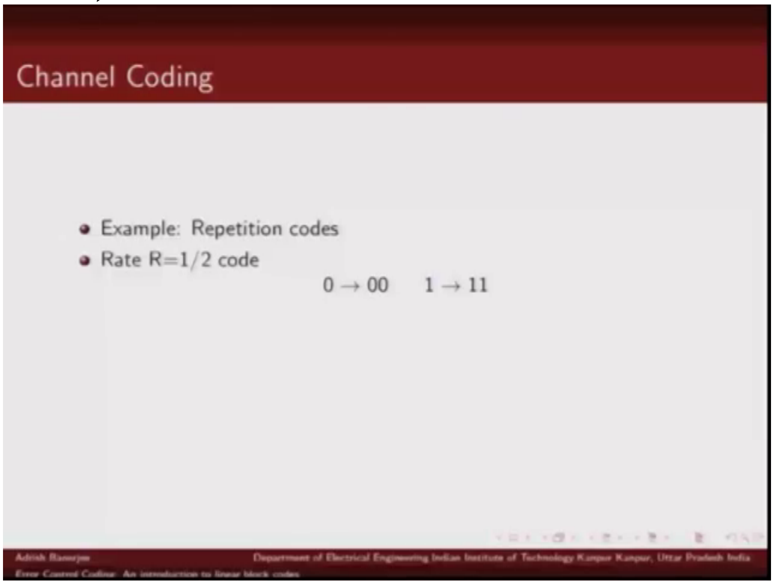

So the rate is defined as the ratio of number of information bits to number of coded bits. So when I say rate one half code, I mean there is one information bit or one message bit and there are two coded bits.

(Refer Slide Time 09:41)



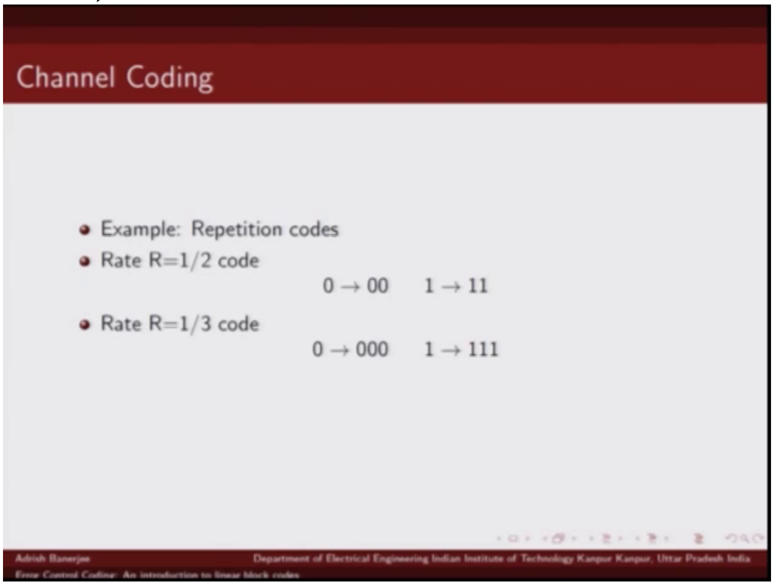
For example in a repetition code,

(Refer Slide Time 09:43)




in a binary repetition code basically we repeat whatever information bit is. So a rate of repetition code would be; would look like this. A binary rate of repetition code would look something like this, so for 0, we would be transmitting 0 0 and for 1, we would be transmitting 1 1. Similarly for a rate

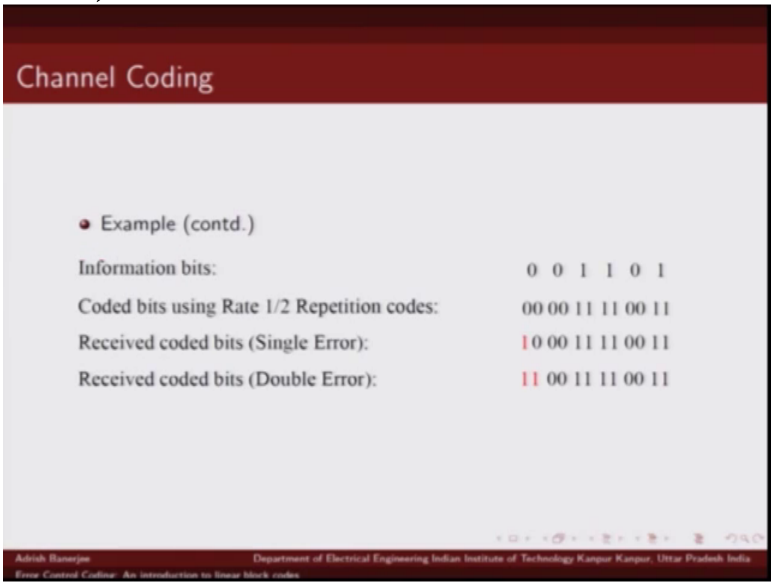
(Refer Slide Time 10:08)




one third repetition code, for 0, we will be transmitting 0 0 0, and for 1, we will be transmitting 1 1 1. So you can see here, in this, in rate one half is we are adding one additional redundant bit and for rate one third code basically we are adding two additional redundant bits. Now how we are going to make use of these redundant bits for error correction and error detection that will be explained in the next slide.

So let's take

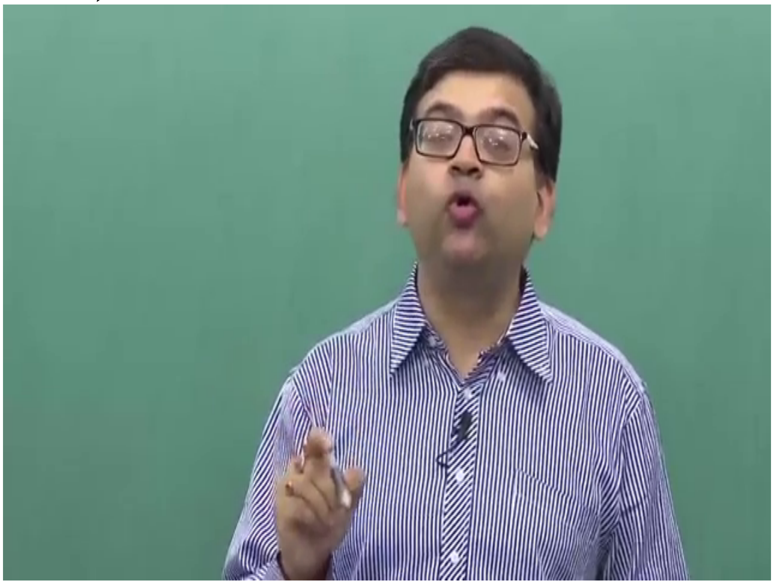
(Refer Slide Time 10:46)




this example Let's say I want to transmit these set of bits. So I want to transmit 0 0 1 1 0 1. Now if I use a rate one half repetition code what would be my coded bits? For 0, I would be transmitting 0 0, for 0, I would be encoding as 0 0, for 1 I would be encoding them as 1 1, and for 1 I would be sending as 1 1, for 0 I will be sending as 0 0 and for 1 I will be sending as 1 1. So this will be my coded sequence, Ok. Now here I have illustrated one case where there is a single error. So this was basically the sequence which was transmitted. Think of it that this bit sequence has been transmitted over a binary symmetrical channel and this is what the received sequence I received. So you can see here, this is a case of a single error. The first bit which was transmitted 0 was received as 1. Now how can I use error correcting codes to detect error?

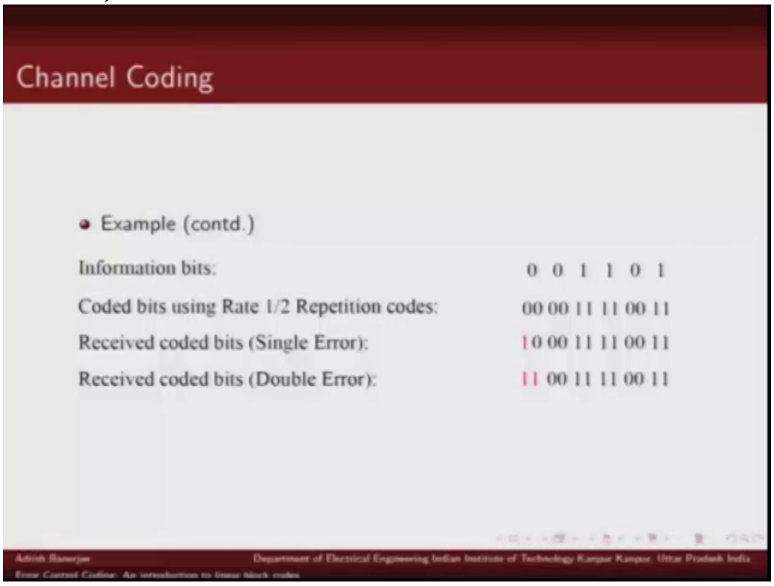
So since it is a rate one half code, for each information bit I am sending two coded bits. So at the receiver I will look at two bits at a time. So I will look at, first I will look at this 1 0. Now since it is a repetition code what do you expect? I expect that both the bits should be same, right? But here in this case first bit is 1; second bit is 0 which means there is a transmission error. So I am able to detect single error. How? Because these bits were encoded using rate half repetition code; I expect these two bits to be same. So I know there is an error in the first bit but I don't know whether this is bit 0 or bit 1. Let's look at other received bits, 0 0 this will be decoded as 0, 1 1 this will be decoded as 1, 1 1 this will be decoded as 1 there is no ambiguity, 0 0 this will be decoded as 0 again there is no ambiguity, and 1 1 this would be decoded as 1. So we can see that using one additional redundant bit we are able to detect

(Refer Slide Time 13:36)



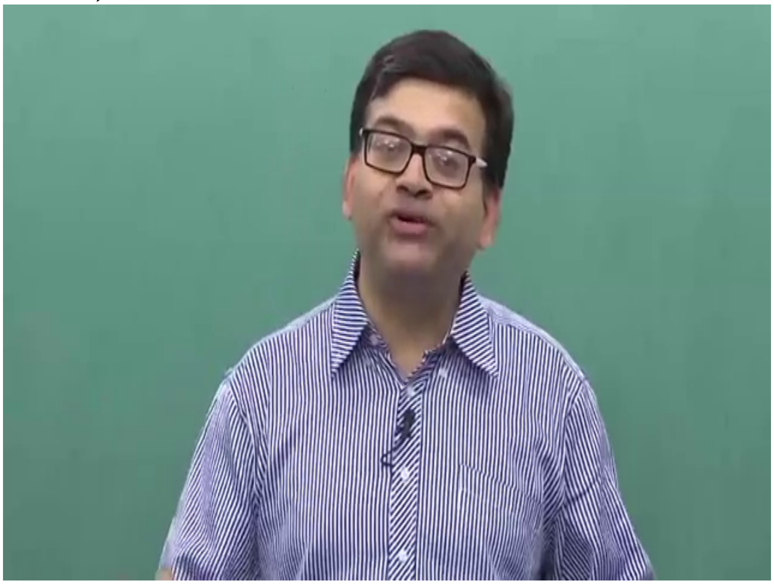
single error

(Refer Slide Time 13:39)



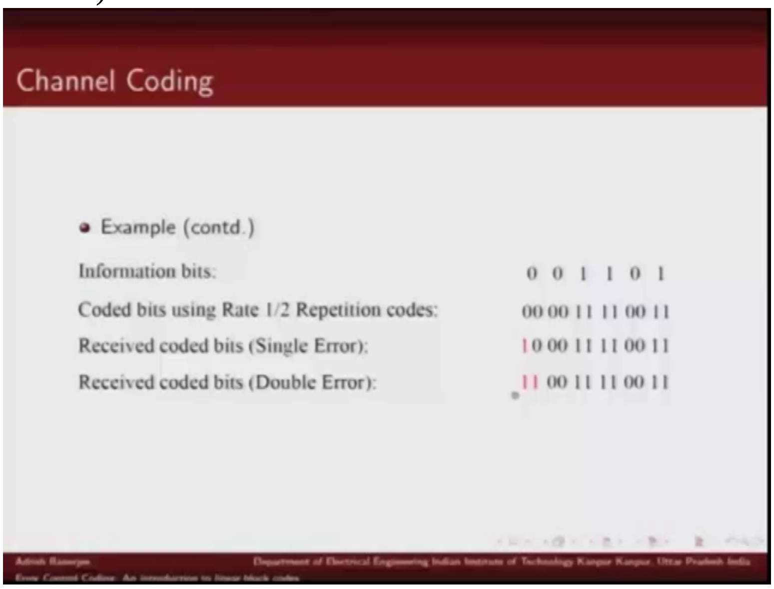

Now let's look example for double error. So let's say the first and second bit are received in error. So what we have received is basically 1 1 0 0 1 1 1 1 0 0 1 1. So the first two received bits are in error 1 1. Now let's see whether we can detect using this rate half repetition code. So again we will follow the same logic for decoding. We will look at two bits at a time. So first two bits are 1 1; now since these bits are same, we will decode them as 1, but what was transmitted, it was 0. So we can see that this is a case of undetected error. Even those these two bits were received in error, the decoder is not able to detect this

(Refer Slide Time 14:46)



error So this kind of thing happens when the error pattern is such that it transforms one code word into some other code word. So since

(Refer Slide Time 14:56)

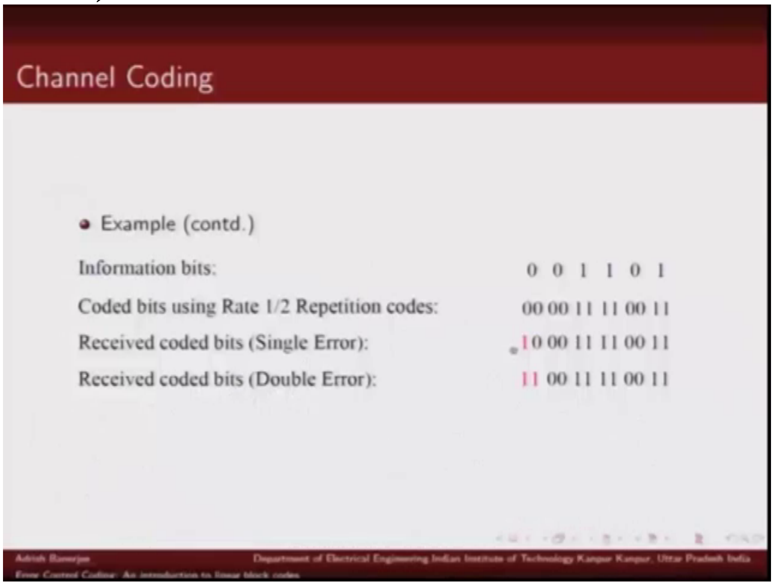



1 1 is a valid code word for 1; the decoder is not able to detect this error. So this rate one half repetition is able to detect single error but it is not able to detect double errors.

(Refer Slide Time 15:14)

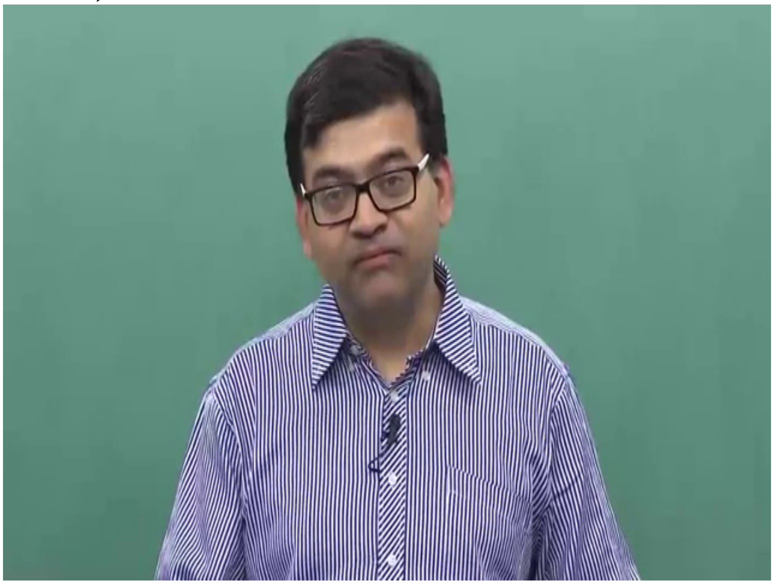
Now let's look at whether it can correct any errors. So let's look at

(Refer Slide Time 15:22)



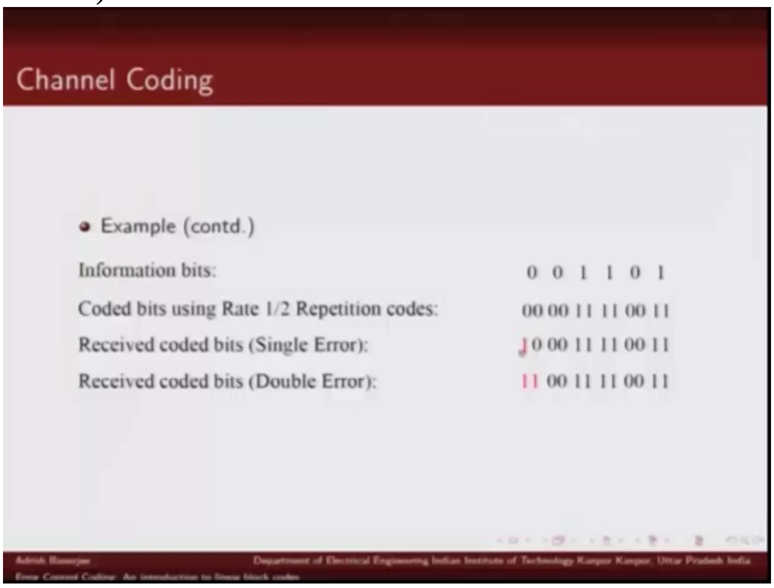

this example when we had single error. So note here, so what we received was 1 0. So we were able to detect error that there was an error. But can we correct it? No

(Refer Slide Time 15:36)



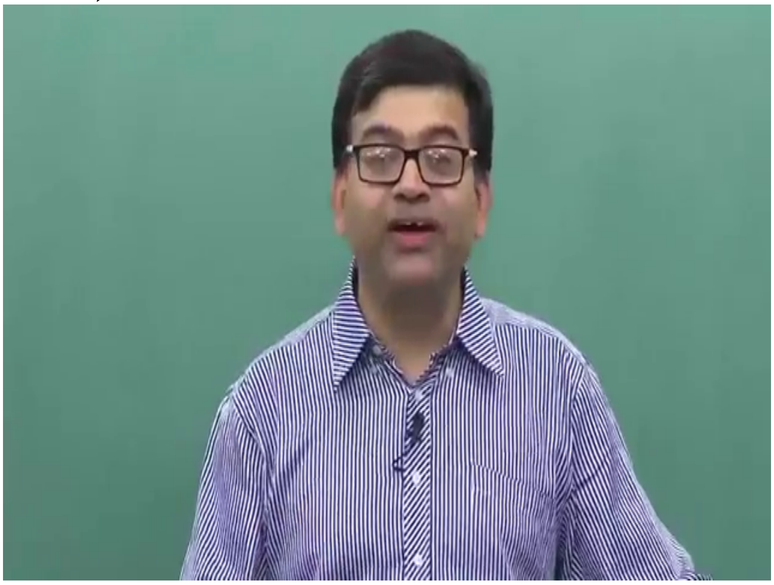
we cannot. Why? It is equally likely that this

(Refer Slide Time 15:40)




1 that we received was 0 or this 0 that we received was 1 if we are talking about binary symmetrical channel, right? So we do not know whether the first bit got flipped to 0, first bit got flipped to 1 instead of 0 or the second bit got flipped to 0 instead of being 1. So this particular rate half repetition code

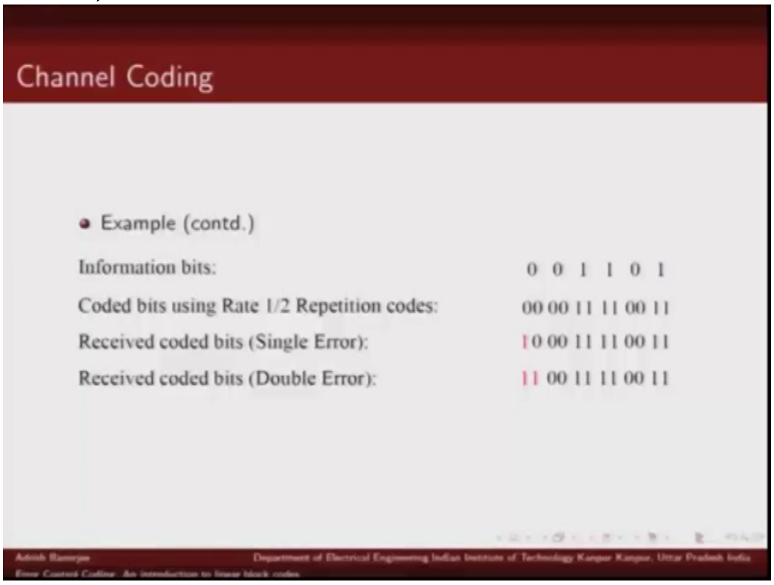
(Refer Slide Time 16:09)



cannot correct any errors It can only detect single errors.

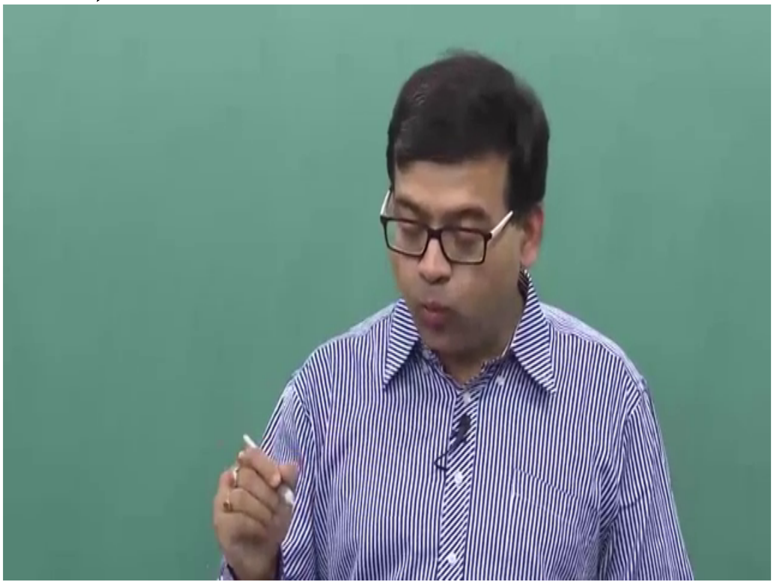
Now look at

(Refer Slide Time 16:21)



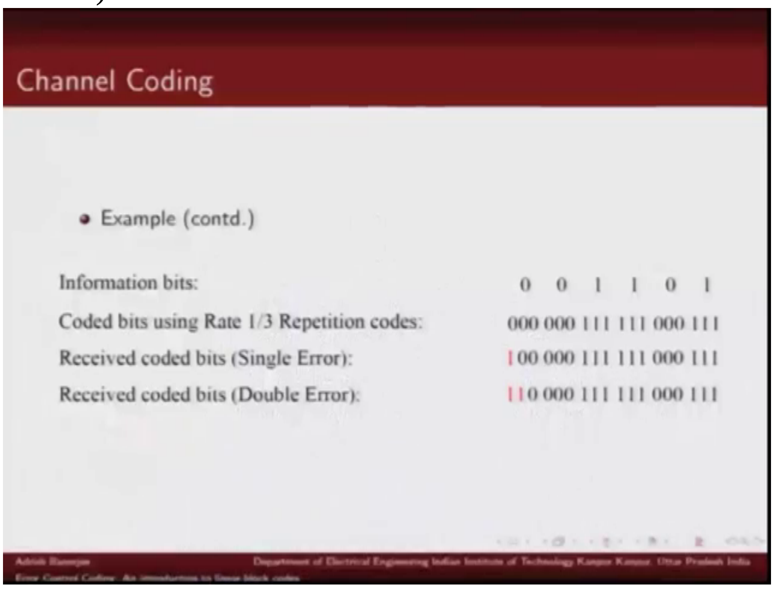

another example This time we are considering a rate one third repetition code. So what does rate one third repetition code means? For each, and again we are considering binary code, so for each bit we are adding two parity bits and we are repeating the same bit. So for 0, we will be transmitting, we will be coding it as 0 0 0, for 1 we will be coding it as 1 1 1.

(Refer Slide Time 16:53)



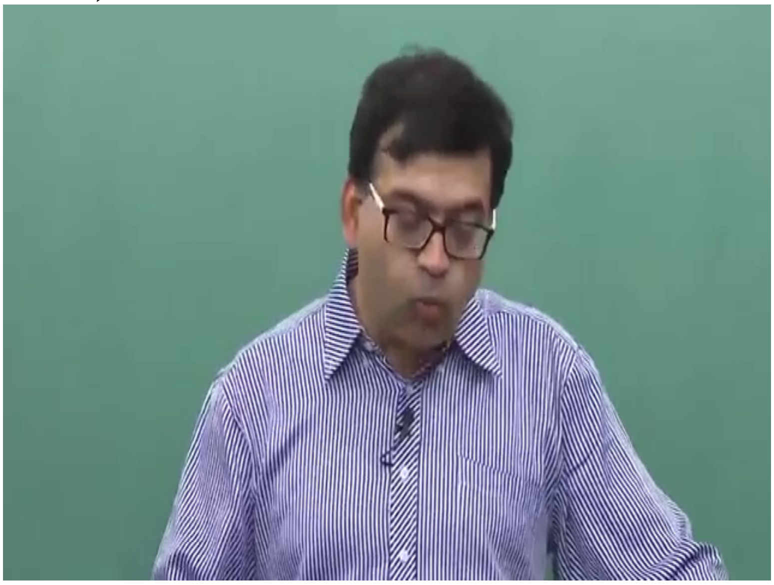
So again we consider the same example of transmitting 0 0 1 1 0 1. So we are transmitting the same

(Refer Slide Time 17:02)



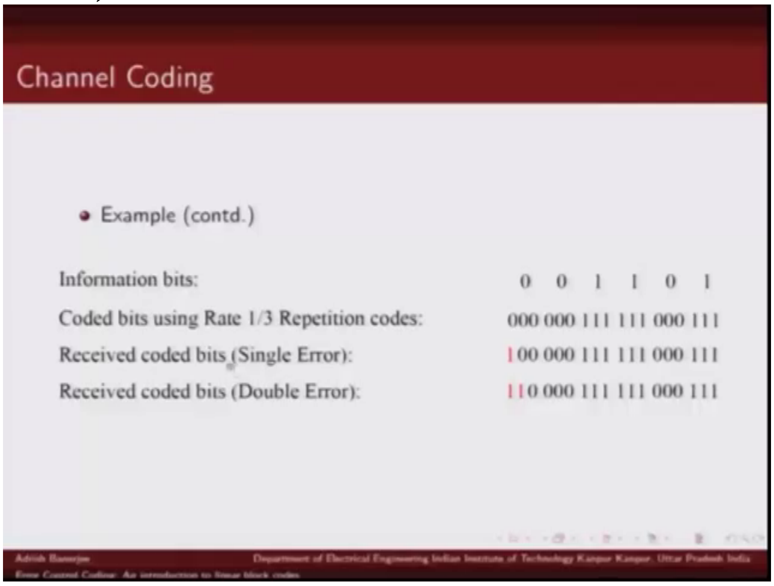

information sequence This time we are encoding them using rate one third repetition code so this 0 will be encoded as 0 0 0. Similarly 1 will be encoded as 1 1 1 so we will be transmitting this. So this information sequence will be coded in this particular way. Now we will again look at what happens

(Refer Slide Time 17:31)



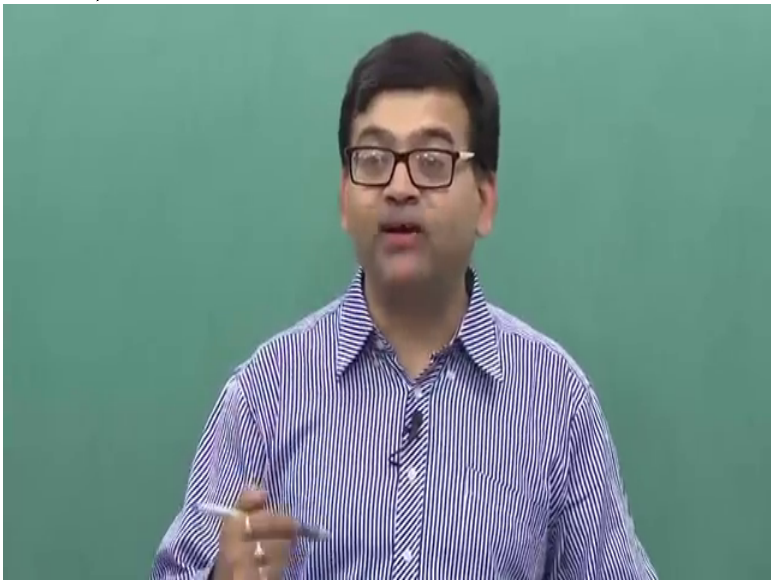
when there are errors in the received sequence, like we did for rate one half repetition code. So let's again look at

(Refer Slide Time 17:39)



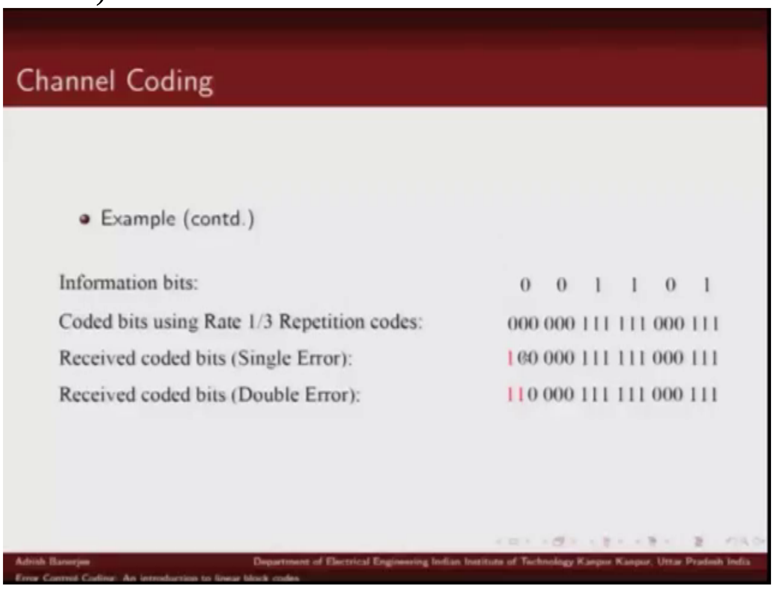

example for single error scenario So let's say the first bit was received in error. So instead of 0, we received a 1. Now let's see whether our rate one third repetition code can detect single error. So since it is rate one third code for each information bit we are sending 3 coded bits. So we are going at the receiver. We are going to look at 3 bits at a time. At the decoder we are going to look at 3 bits at a time. So we will first look at these 3 bits, 1 0 0. Now what do you expect? We expect, since we are using our repetition code, we expect

(Refer Slide Time 18:26)



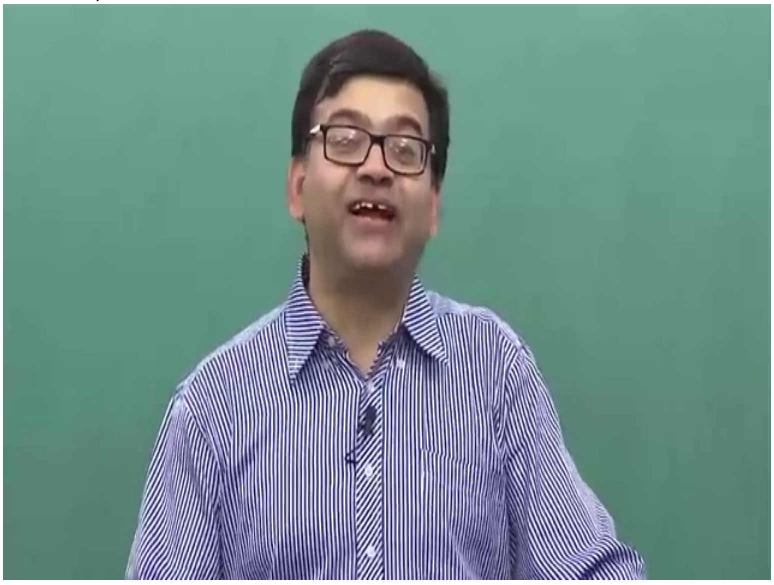
all these 3 bits to be same But here in this case,

(Refer Slide Time 18:31)



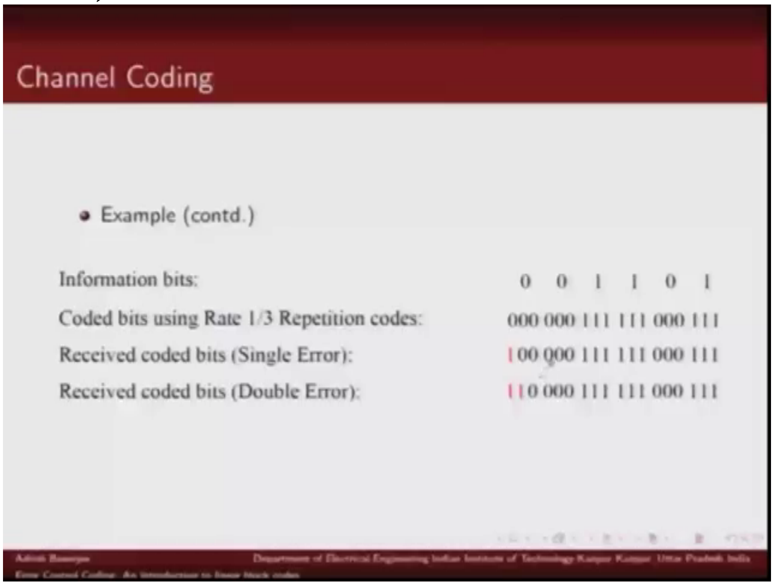

they are not, because what we have received is 1 0 0. Now what does that mean?

(Refer Slide Time 18:37)



That means there is a transmission error. So we are able to detect single error using a rate one third repetition code.

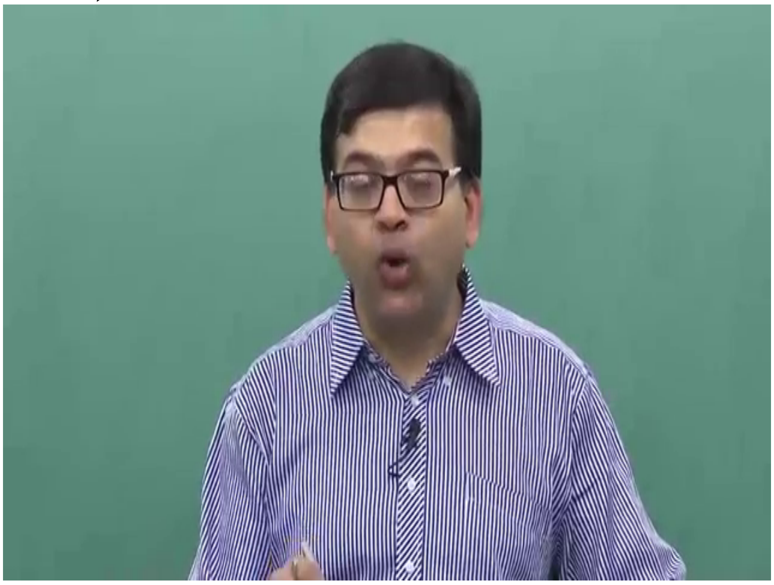
(Refer Slide Time 18:53)




If we look at other sets of bits 0 0 0, there is no error here; 1 1 1 no error, again no error here, no error here, no error here. So rate one half repetition code was able to detect single error, even rate one third code is also able to detect single error.

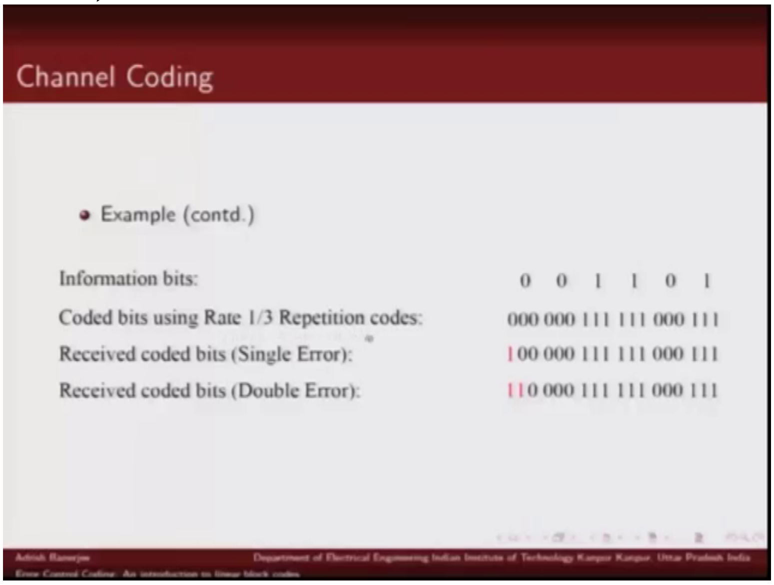
Now let’s look at double error. So let's consider scenario when first two bits are received in error. So we have 1 1 and rest of the sequence is this, Ok. Now can we detect double error? We just look at; we again look at 3 bits at a time. So if you look at 3 bits at a time, the first 3 bits are 1 1 0. Now we could see that there is an error. Why? Because either this should have been 0 0 0 or 1 1 1

(Refer Slide Time 19:56)



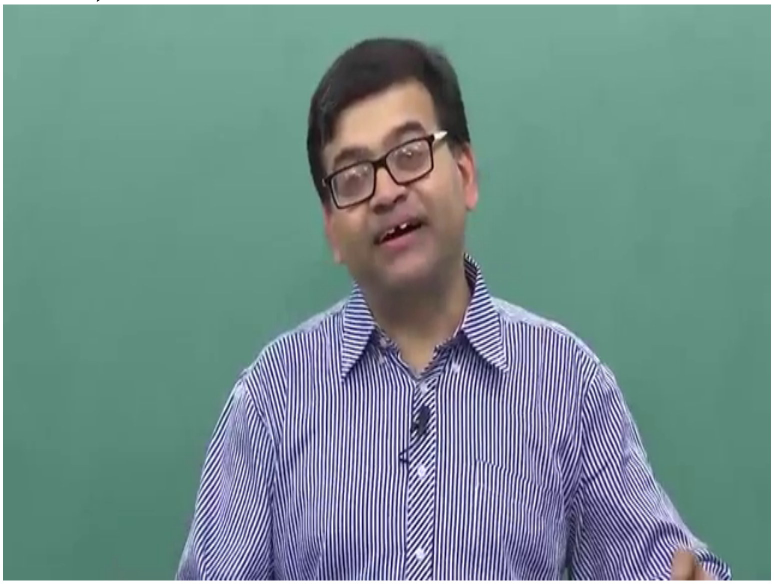
but we received 1 1 0. So we are able to detect using rate one third repetition code

(Refer Slide Time 20:03)



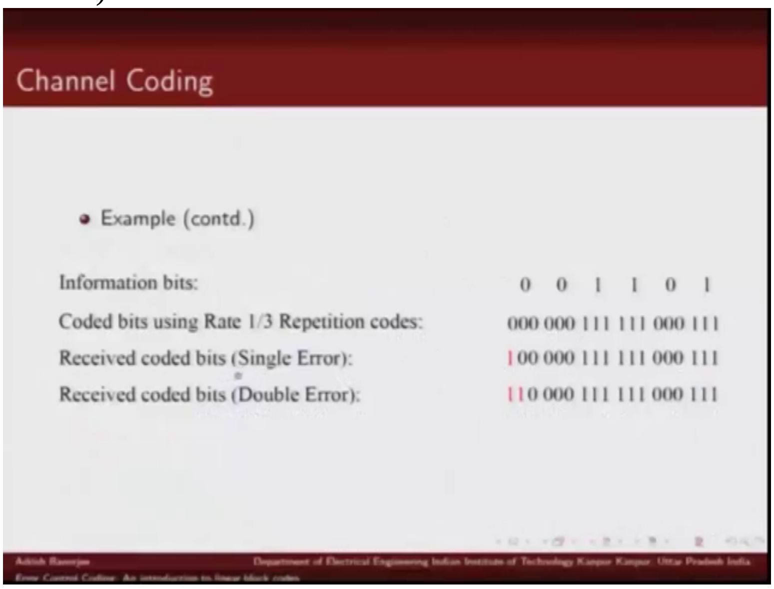

we are able to detect double errors as well which we were not able to detect using rate one half repetition code. Now let's look at the error correcting

(Refer Slide Time 20:16)



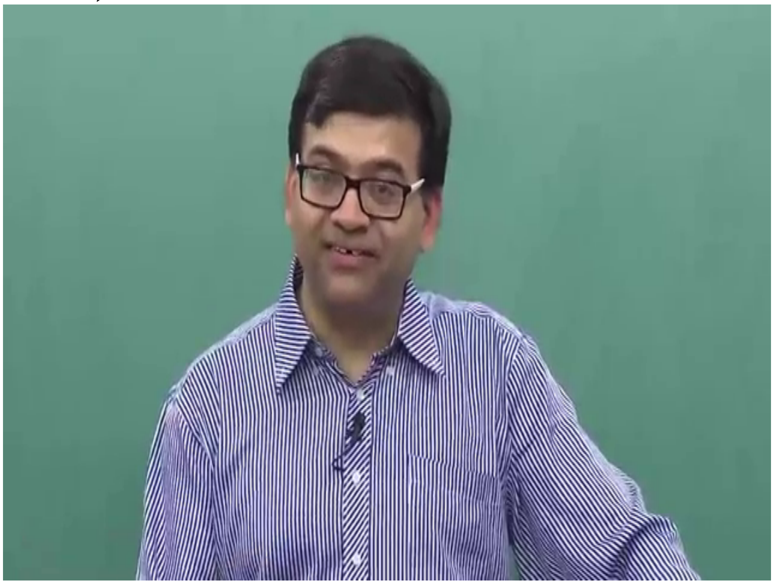
capability of this code So let's go back again and look at

(Refer Slide Time 20:20)



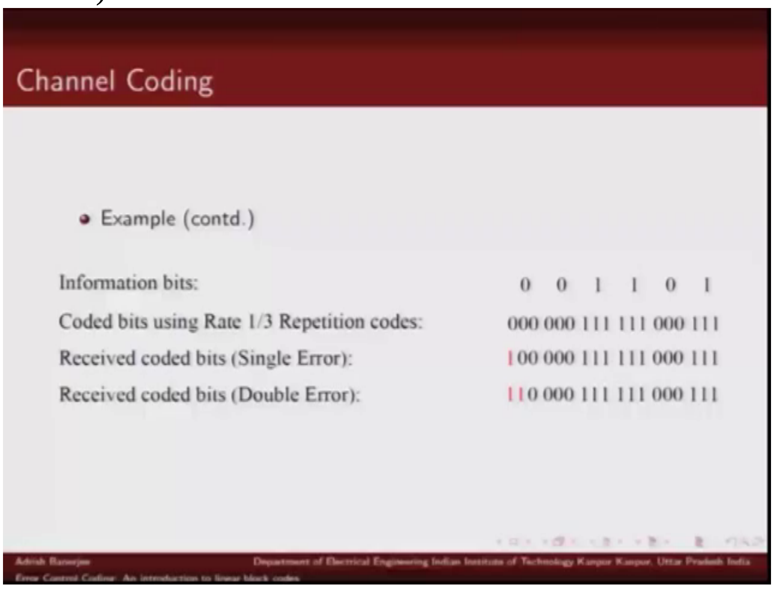

single error situation So when single error happens, so something like this; let’s say one of the bit got flipped, 1 0 0. Now can we correct single errors?

(Refer Slide Time 20:33)



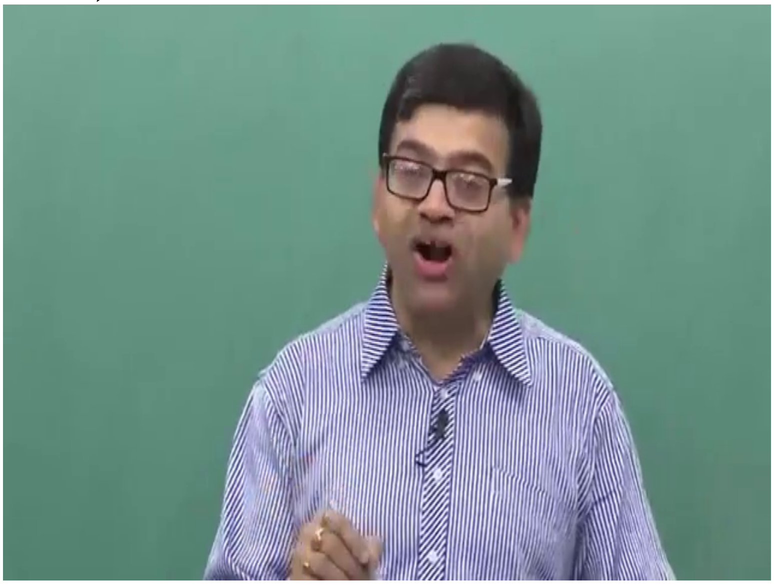
And the answer in this case is yes. Why? Because if you look at these 3 bits, two bits are already 0 and one bit is 1. So it is, and what are the possible outcomes?

(Refer Slide Time 20:48)



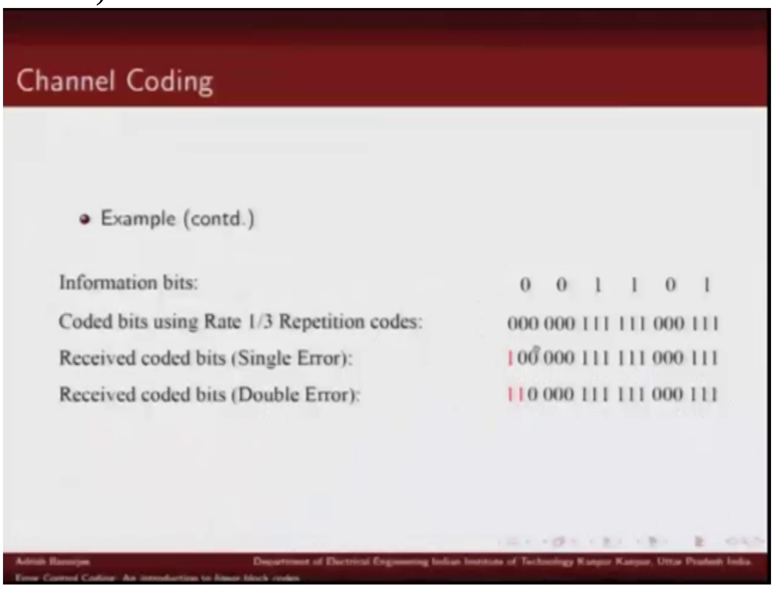

This could be either 0 0 0 or 1 1 1 and it is more likely

(Refer Slide Time 20:55)



that one bit got flipped. It is more likely that 0 got flipped to 1 rather than two 0s, two 1s getting flipped to 0. So it is

(Refer Slide Time 21:05)

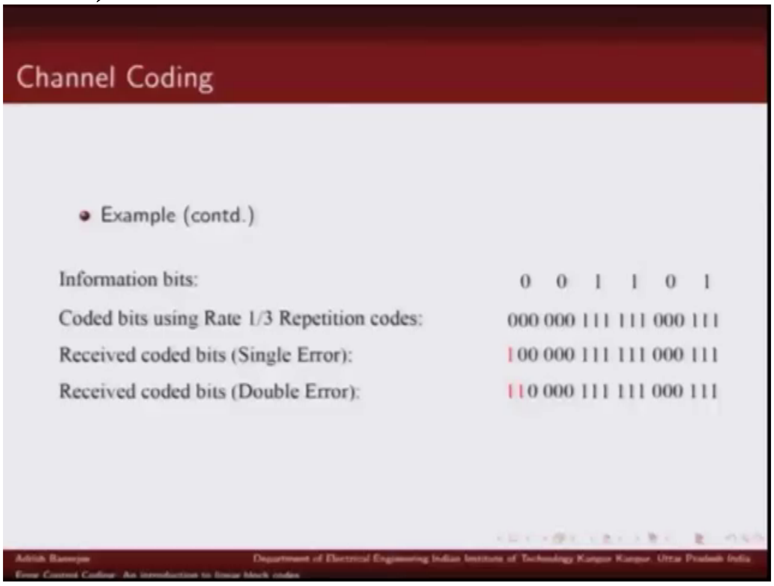



more likely that this bit got flipped from 0 to 1 instead of these two bits getting flipped from 1 to 0. So using majority logic, these majority of the bits are 0; we will decode this as 0. So we can see this rate one third repetition code can correct single error. This was not possible for rate one half repetition code. Now can it correct double errors? Now if you look at this 1 1 0, it will think that this particular bit got flipped from 1 to 0 so it will decode this as 1. So this cannot correct double errors. So to summarize, we saw that rate one half repetition code can detect single error but cannot correct single error.

(Refer Slide Time 22:09)

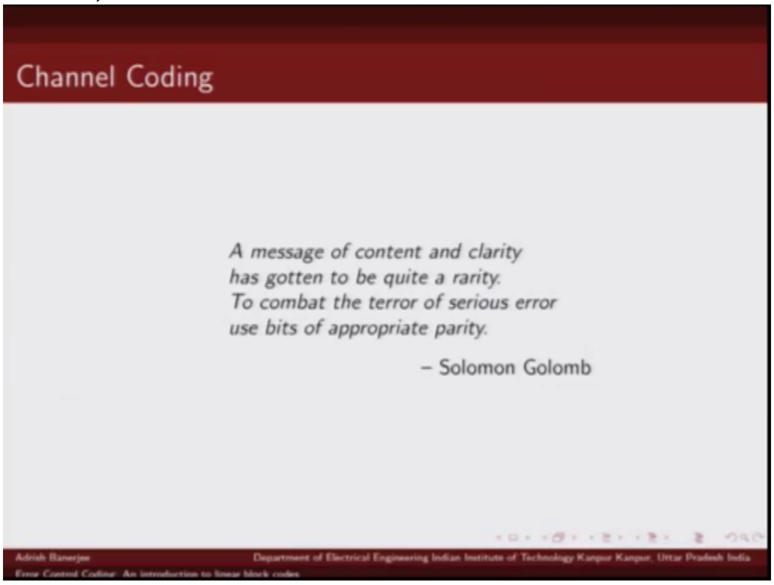
It cannot detect double errors whereas rate one third repetition code can correct single error

(Refer Slide Time 22:15)



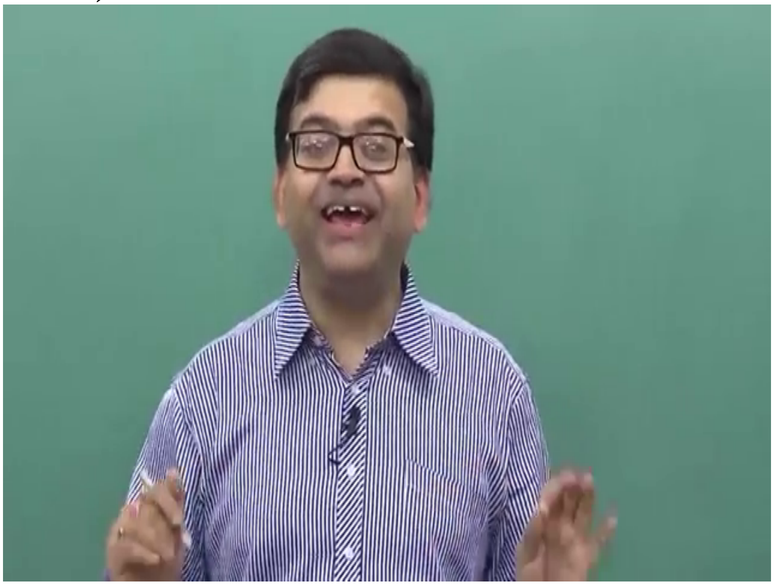

and can detect single error. It can detect double error but it cannot correct double errors. So why is this code better than the first code? It has certainly has better error detecting capability than the rate one half code. This we will discuss in subsequent lectures. It has to do with separation between, the distance separation between the code words and you can see basically in this particular code we are using two redundant bits and in the previous case we were just using one redundant bit. So the error correcting capability and error detecting capability of the code is depending, is dependent on the distance properties of the code and we will talk about in subsequent lectures. So to summarize it I think

(Refer Slide Time 23:05)



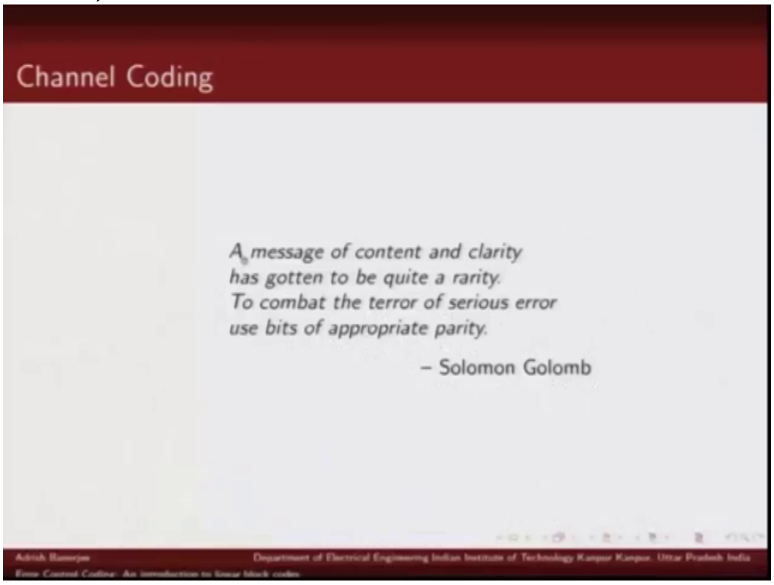

this quotations by Solomon Golomb rightly

(Refer Slide Time 23:10)



captures what error correcting code is all about So I will read it. A message

(Refer Slide Time 23:16)




of content and clarity has got to be quite a rarity. To combat the terror of serious error, use bits of appropriate parity. Thank you.