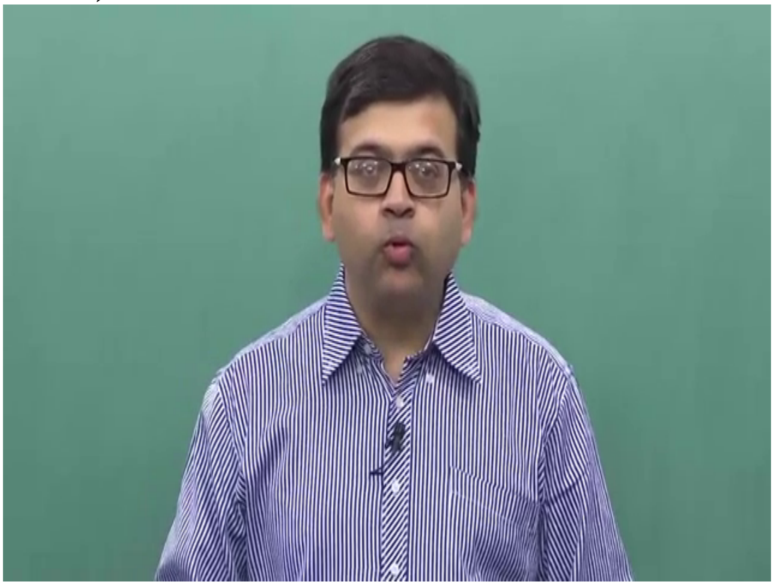
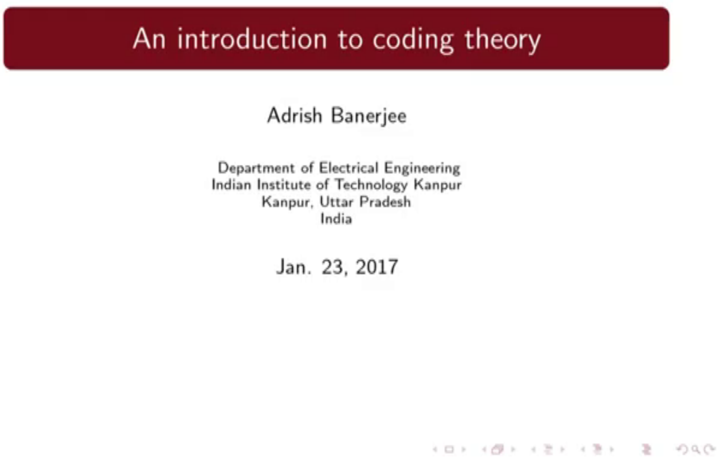
An Introduction to Coding Theory
Professor Adrish Banerji
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Module 01
Lecture Number 02
Introduction to Error Control Coding-II

(Refer Slide Time 00:13)



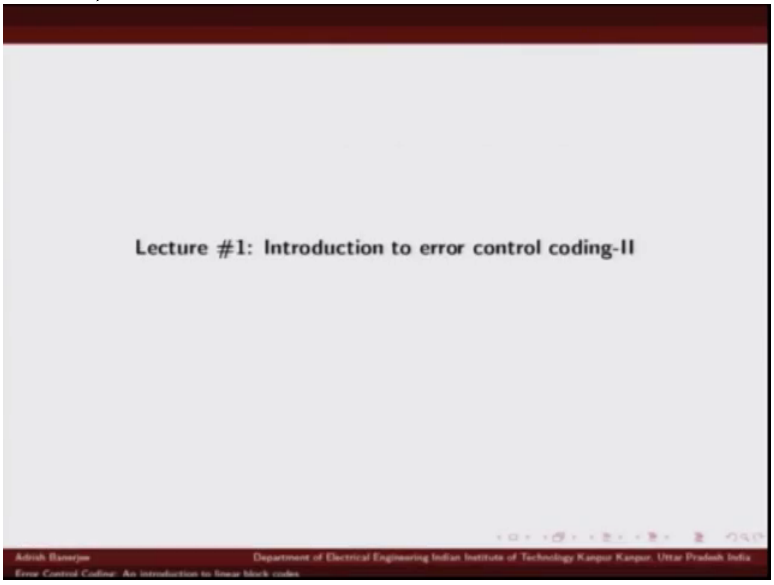
Welcome to the course on Coding Theory.

(Refer Slide Time 00:15)



I am Adrish Banerji. We are going to continue our introduction

(Refer Slide Time 00:21)

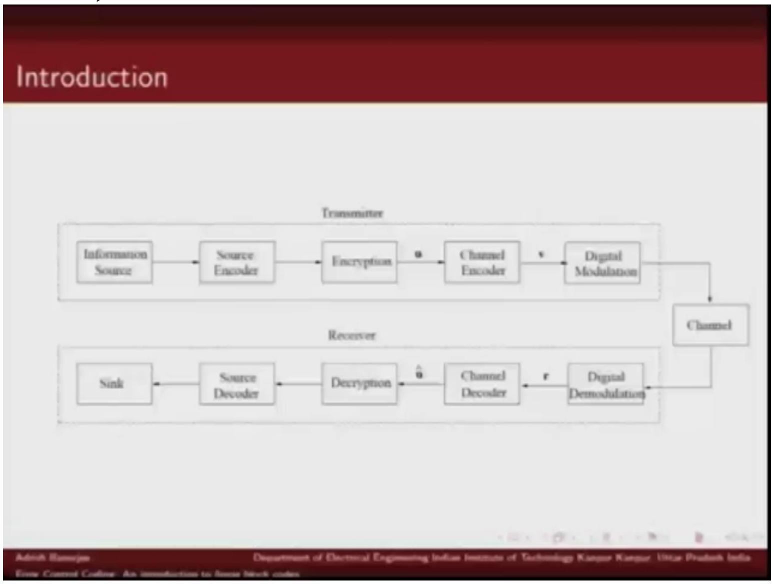


to error control coding This time we are going to show you

(Refer Slide Time 00:26)

where does error control coding fits in our digital communication system So this is a

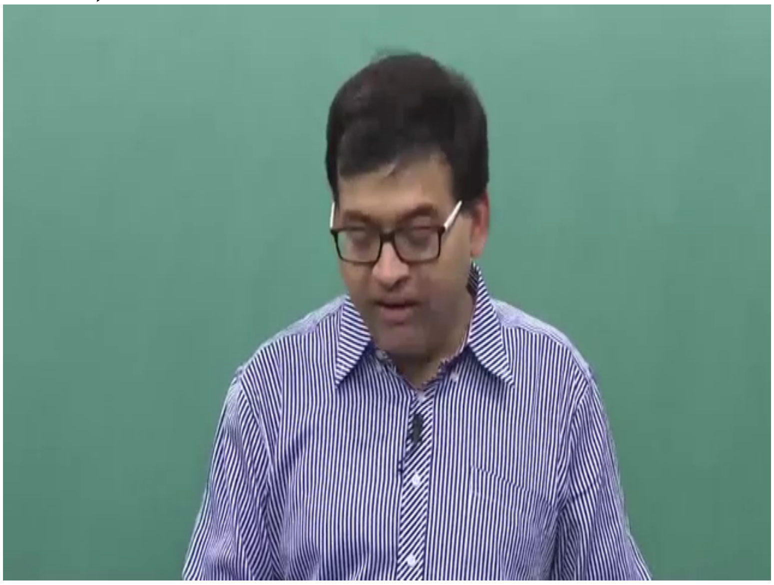
(Refer Slide Time 00:34)



block diagram of a digital communication system. So we have our information source which is basically the message that we want to transmit. The first block that you see is source encoder. Now source encoder essentially does data compression. It tries to represent the source efficiently in terms of minimum number of bits required to represent the source. So that's basically done by this block source encoder. Once you have represented, you have compressed your source, the next block is basically data encryption. This is for secrecy and then the next block which is of our interest here in this course is what is known as channel encoder. So it takes into account as input information bits and gives us as output the coded bits. We typically use the symbol u to denote information sequence and the symbol v to denote the coded sequence.

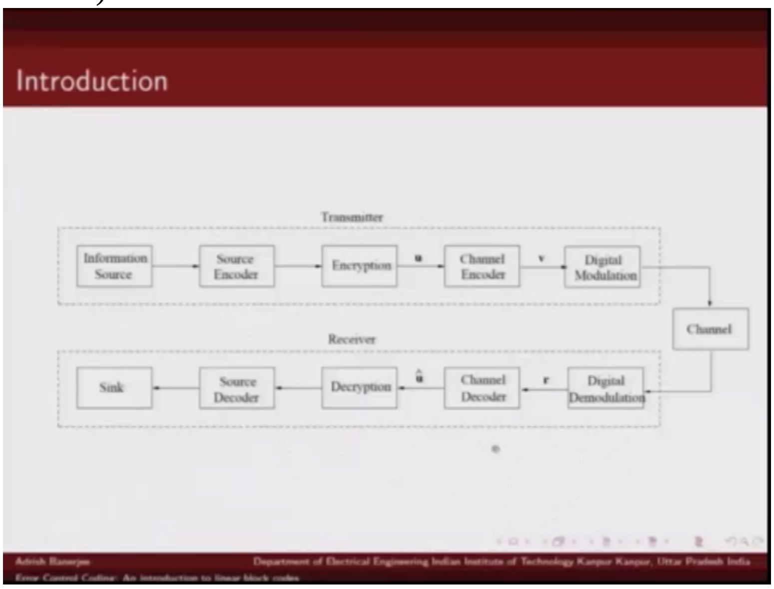
So this course essentially will deal with this block of digital communication system essentially how to design good error correcting codes and properties of error correcting codes. Once you have this encoded sequence, you want to send these bits over a communication link so you need to do digital modulation. Once you have modulated your signal you want to send it over the transmission medium which is a channel. And then at the receiver you just have the reserve, reverse operation. So once you get a noisy version of the received sequence you would like to demodulate the signal and the demodulated output is then fed to channel decoder whose job is to estimate

(Refer Slide Time 02:50)



information sequence

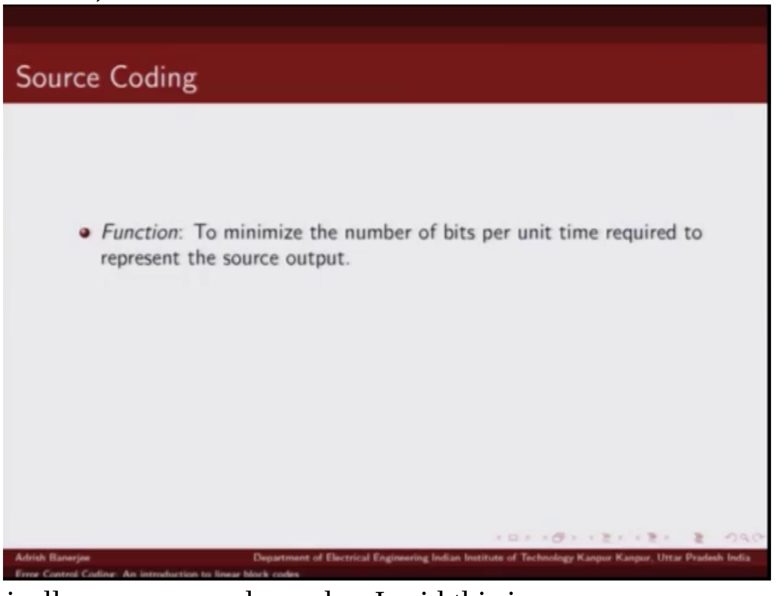
(Refer Slide Time 02:52)



from the received sequence We use the notation r for the received signal, received input's demodulated signal and we will use u hat to denote the estimated information bits. Now if u is equal to u hat there is no error. However if u is not same as u hat then there are decoding errors. Now once you decode information bits then you would like to de-encrypt to get back your plain text data. So this de-encryption is opposite action of encryption and once you get back your plain text you would like to decompress the signal to get back the original source signal. So that's basically in a nutshell, the broad block diagram of a digital communication system. So we can see basically in this course we are interested in this block, we are interested in this block and of course we are interested in this block as well because our error control coding strategy depends on what sort of channel we are sending our information bits over, Ok.

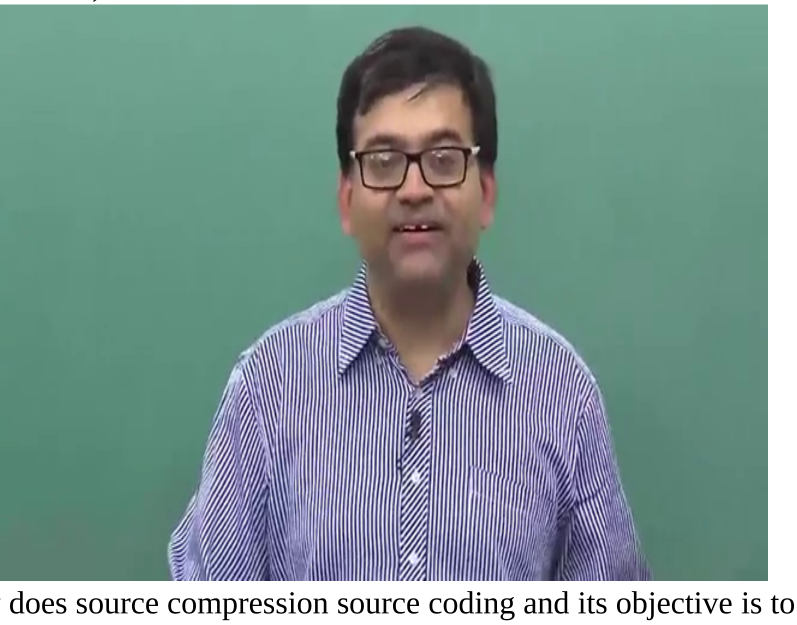
So I will now explain little bit more detail of each of these block diagrams.

(Refer Slide Time 04:23)



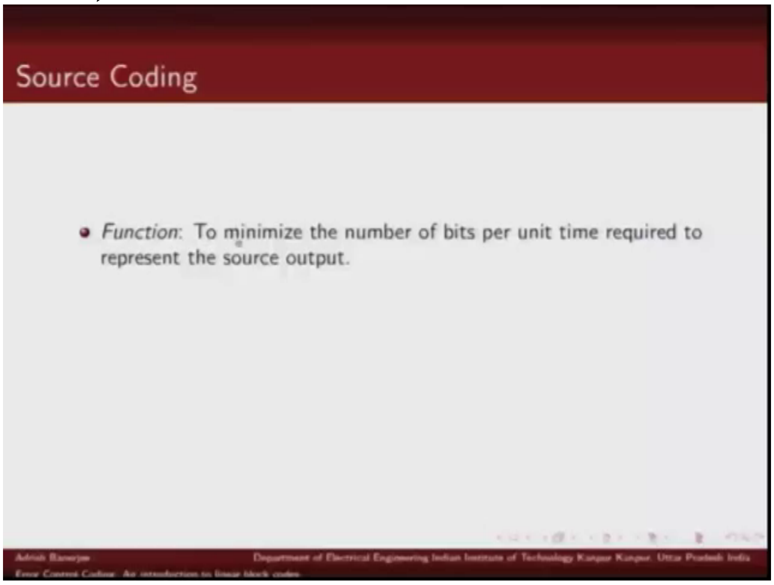
So the first is basically source encoder and as I said this is

(Refer Slide Time 04:28)



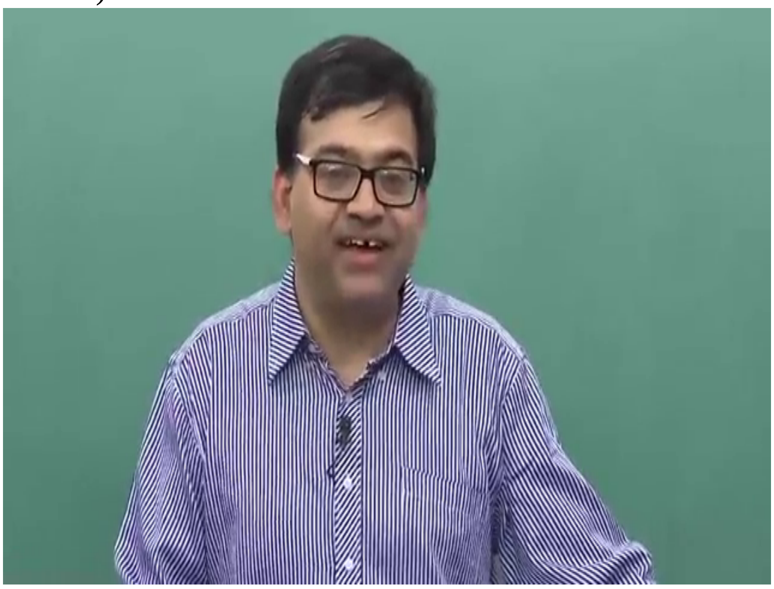
this is essentially does source compression source coding and its objective is to

(Refer Slide Time 04:33)



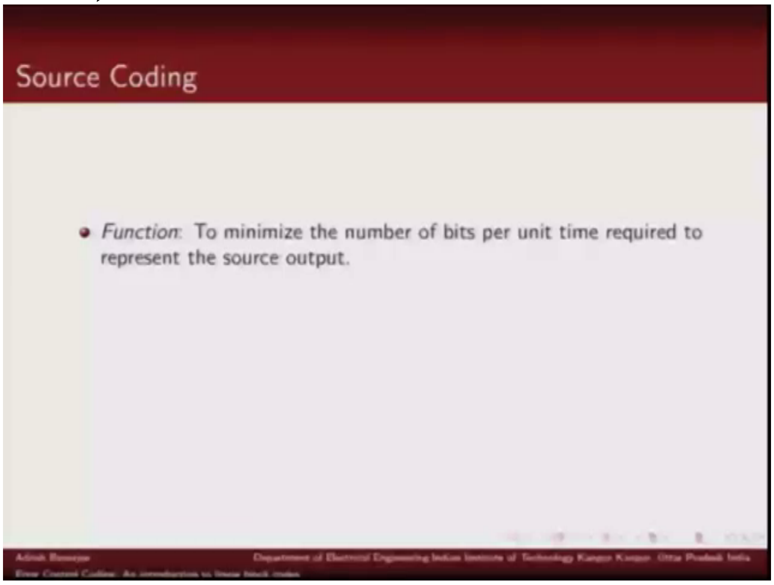
represent your source in minimum number of bits required. As we know that our source to be for example English text, there are

(Refer Slide Time 04:46)



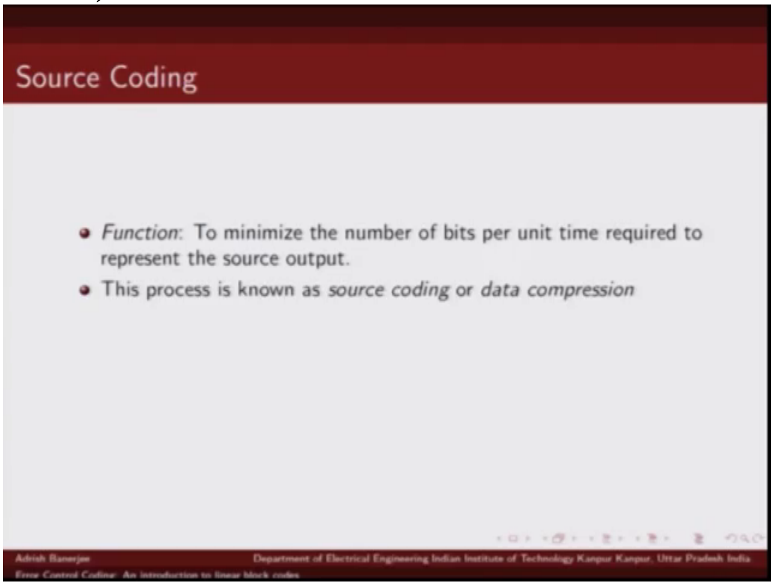
lot of inbuilt redundancies in the text For example, q; the letter q is always followed by u. So if you want to transmit, let's say some word starting with q we don't need to transmit u because in English text, because q will be always followed by u. So like that there are lot of unnecessary redundancy build in into the source which we would want, we would like to get rid of and that is the job of source encoder. So the source encoder will like to represent our source compactly in a minimum number of bits

(Refer Slide Time 05:27)



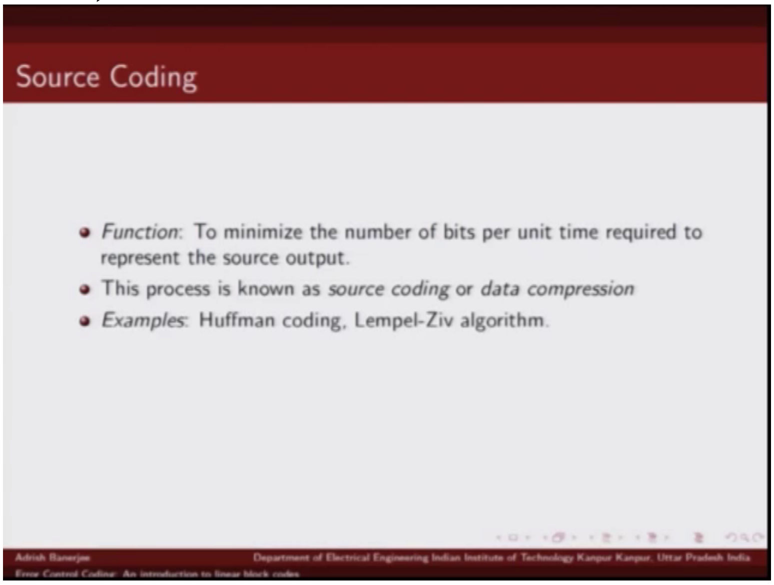
required, that's basically the objective of source encoder So it does basically data

(Refer Slide Time 05:35)



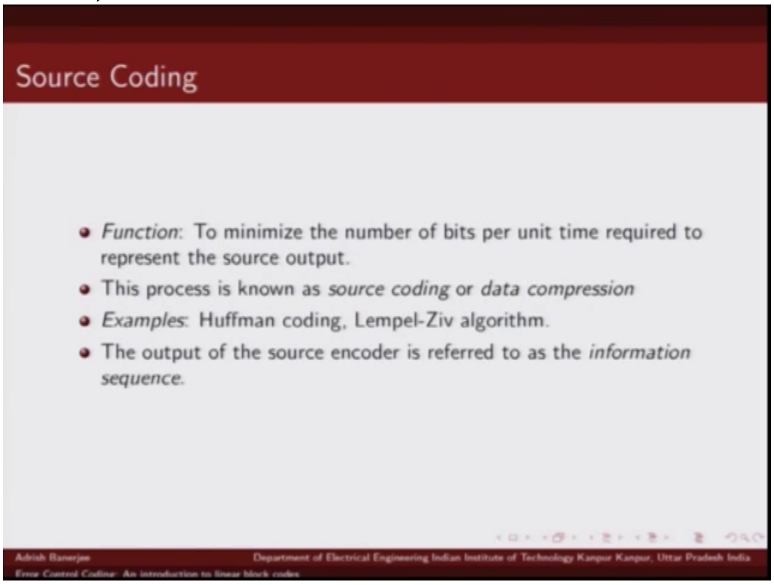
compression Some

(Refer Slide Time 05:38)



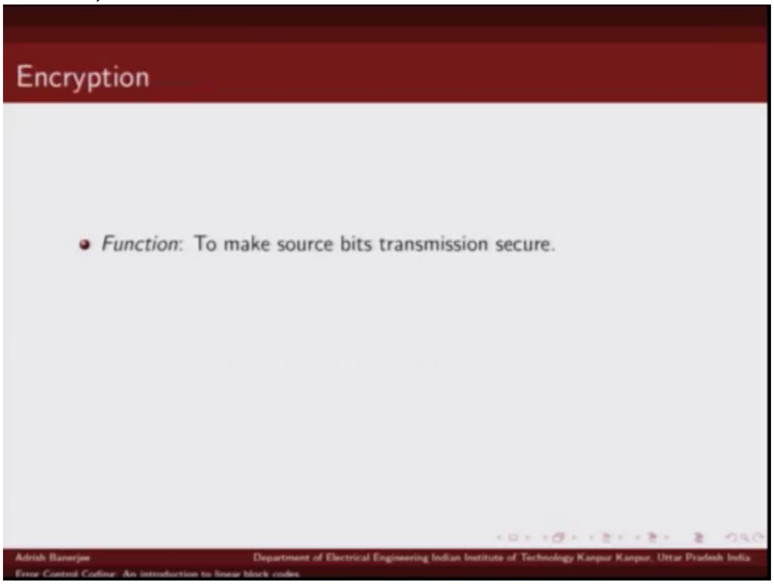
examples of data compression techniques, basically Huffman coding, Lempel's algorithm we would not be talking about this because these are topics of interest in Information theory. We will restrict our discussion to coding theory topics in this course.

(Refer Slide Time 05:59)



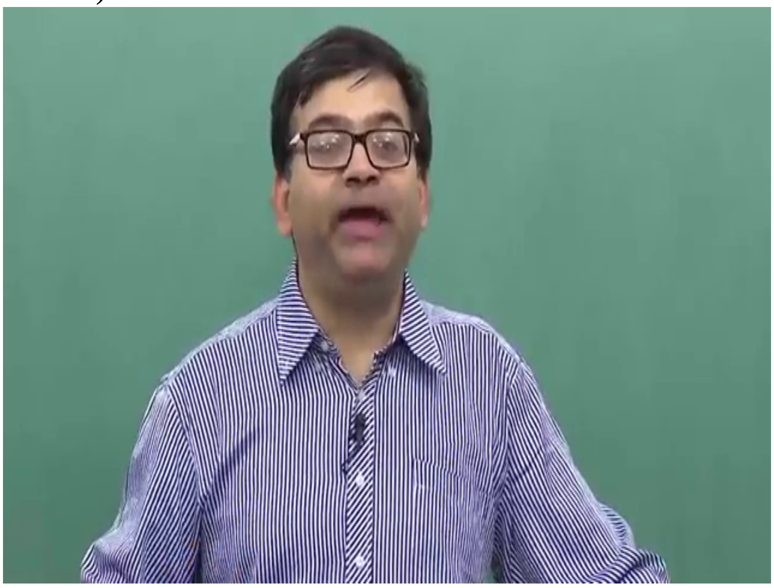
And as I said basically the input to our channel encoder, we are referring it as information sequence. Now the next block

(Refer Slide Time 06:12)



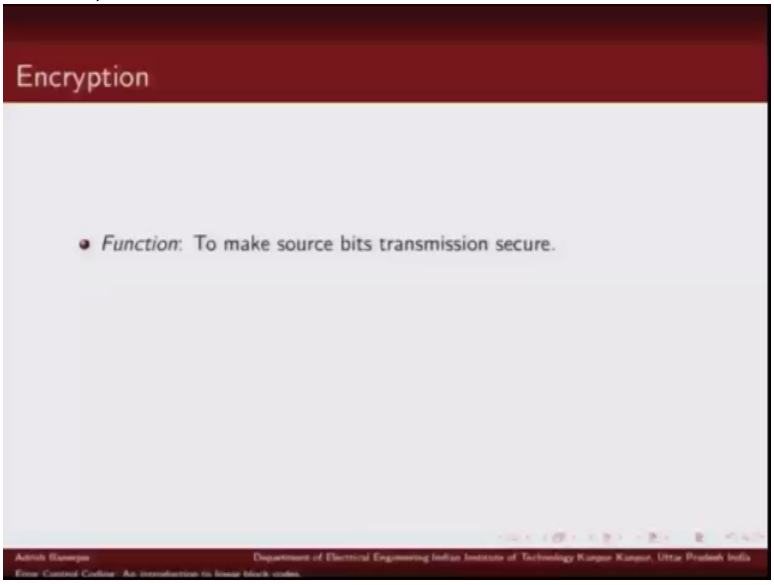
was of encryption and objective of encryption is to make the

(Refer Slide Time 06:17)



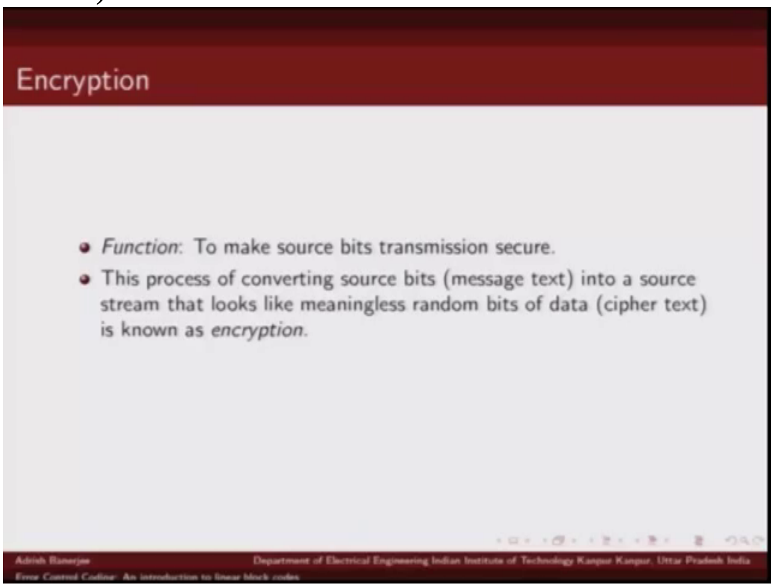
data, or make the bits secure so that unwanted users should not be able to deco/decode, find out what has

(Refer Slide Time 06:27)



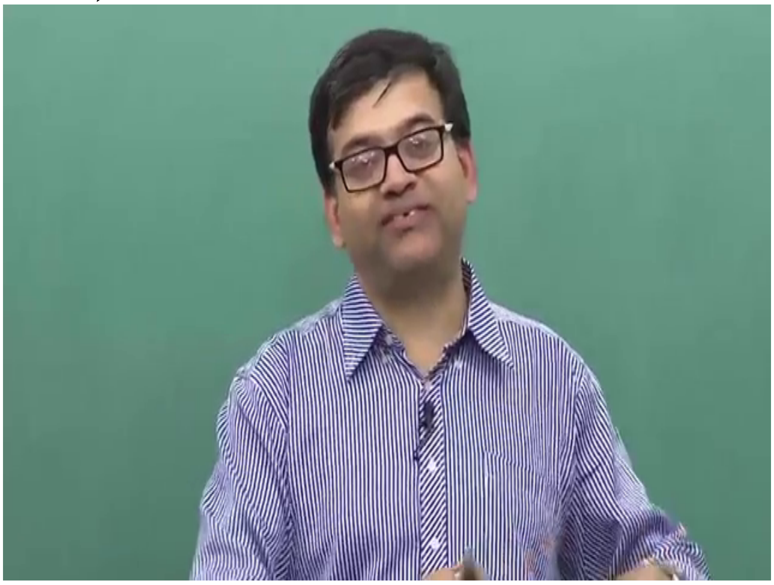
been transmitted So

(Refer Slide Time 06:29)



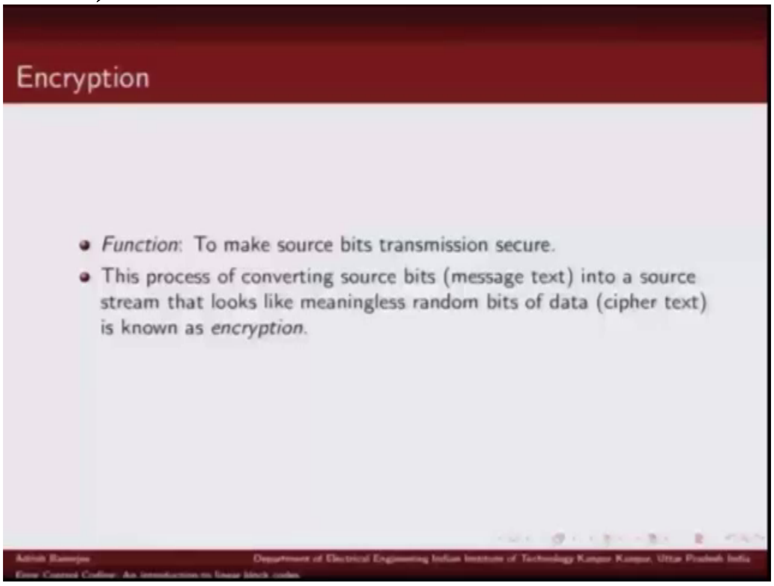
in this encryption process what we do, we convert our source bit into what we call a cipher text. So we basically, encryption takes our source bits and it

(Refer Slide Time 06:45)



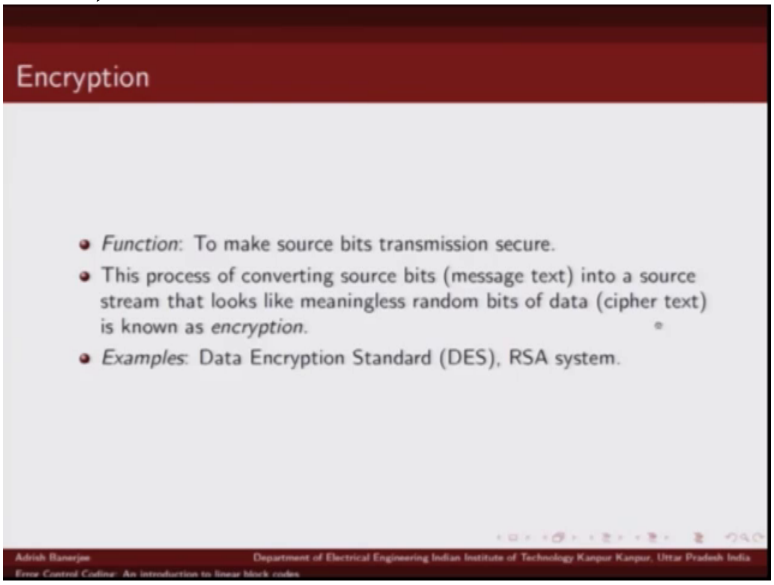
converts it into some random looking bits which, for

(Refer Slide Time 06:50)



somebody who does not have key would appear as a random meaningless data. So it basically converts your message text into what we call, cipher text. That's basically encryption

(Refer Slide Time 07:04)



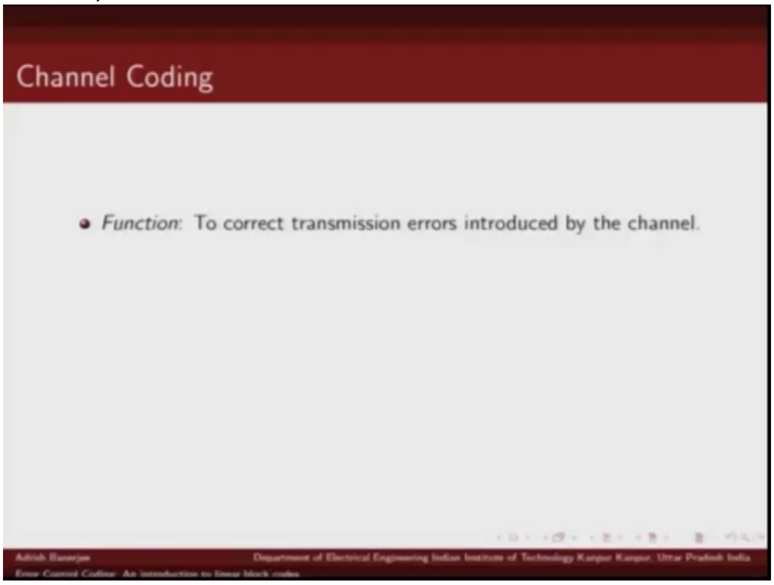
and examples of encryption is Data Encryption Standard, Advanced Encryption Standard, RSA System

The next block

(Refer Slide Time 07:13)

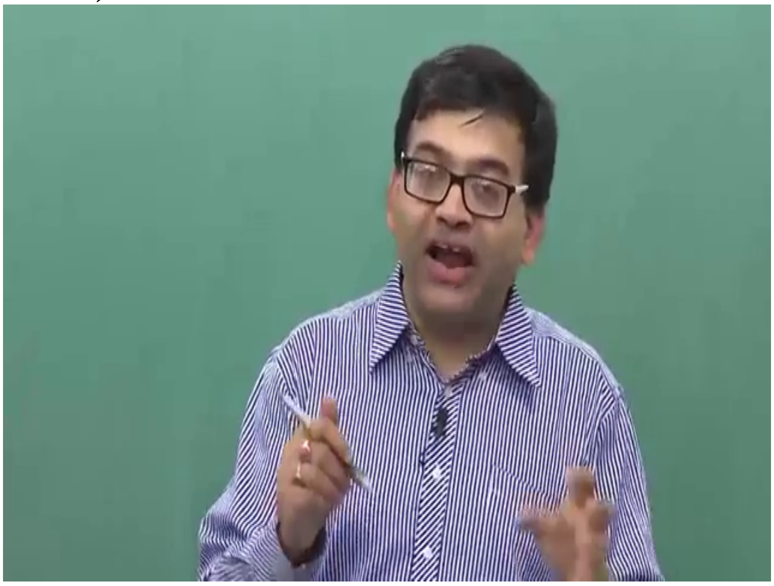
was of channel encoder

(Refer Slide Time 07:16)



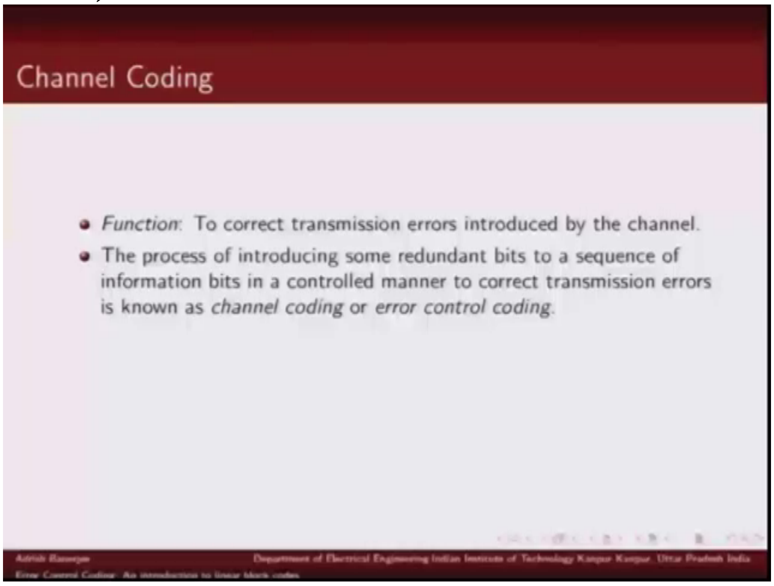
Now this is of interest in this course. And what's objective of this block? We would like to

(Refer Slide Time 07:23)



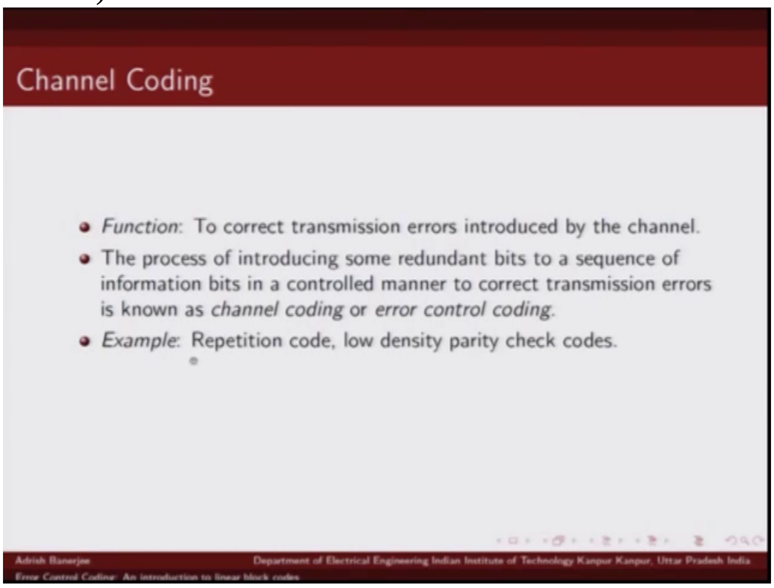
introduce additional redundancies into the source and with the help of these redundant bits, we would like to detect and correct errors induced by transmission errors or storage errors. Basically we would like to correct or detect those errors. Now if you are wondering why can't we use the inherent redundant bits which are available in a source and why, why don't we just get rid of source encoder and channel encoder, just use the plain source which has inherent some redundant bits? The problem with this is we don't have any control over those redundant bits which are inherent in a source where as in our channel encoder; the redundant bits that we are adding have been carefully designed in such a way so as to detect or correct errors. So

(Refer Slide Time 08:17)



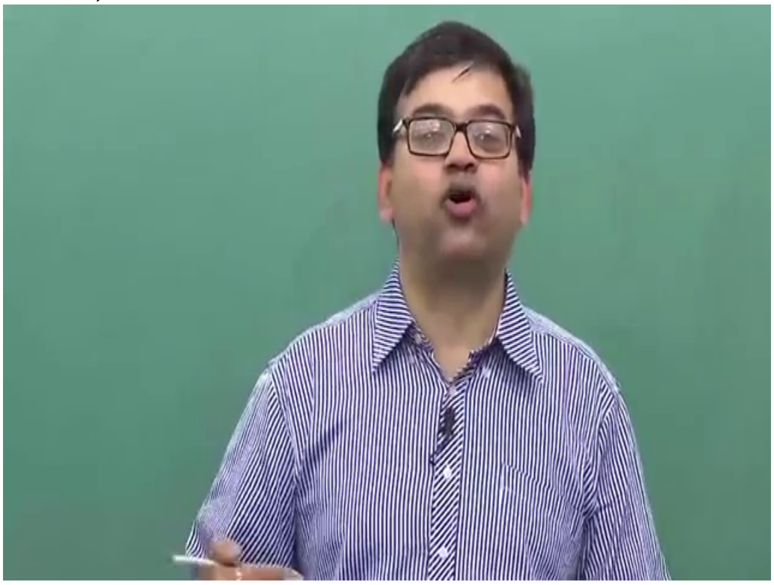
this process of basically adding some redundant bits which we call as parity bits for message bits is basically, this process is known as channel coding or error control coding and this is essentially done to detect or correct error which has caused by transmission medium.

(Refer Slide Time 08:39)



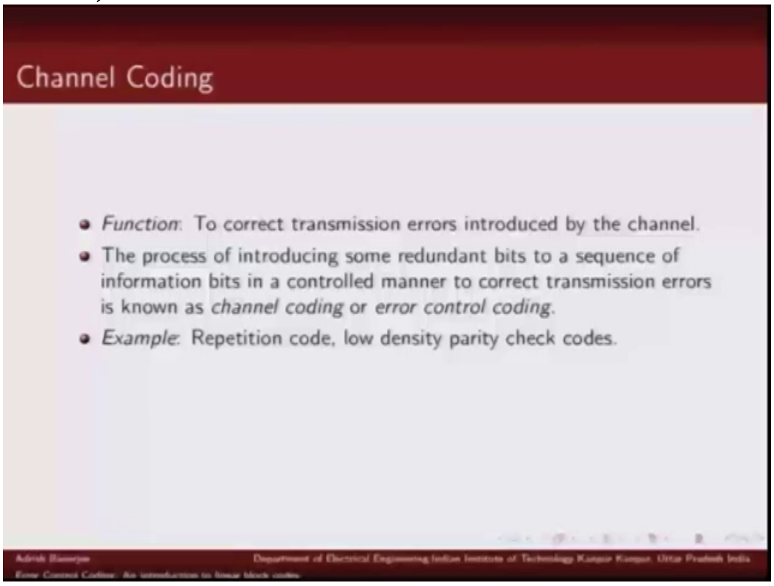
In the last lecture I already gave an example for simple, very simple code, repetition code. There are, throughout this course we are going to talk about

(Refer Slide Time 08:48)



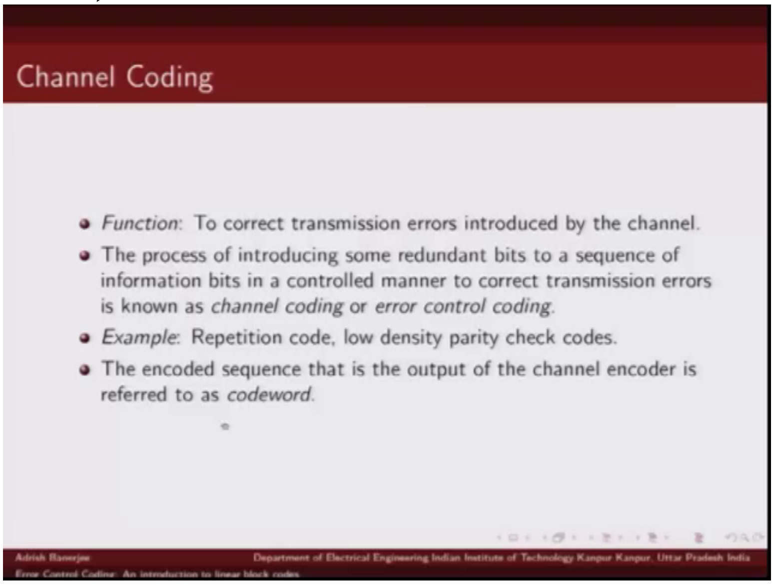
some other examples of error correcting codes

(Refer Slide Time 08:54)



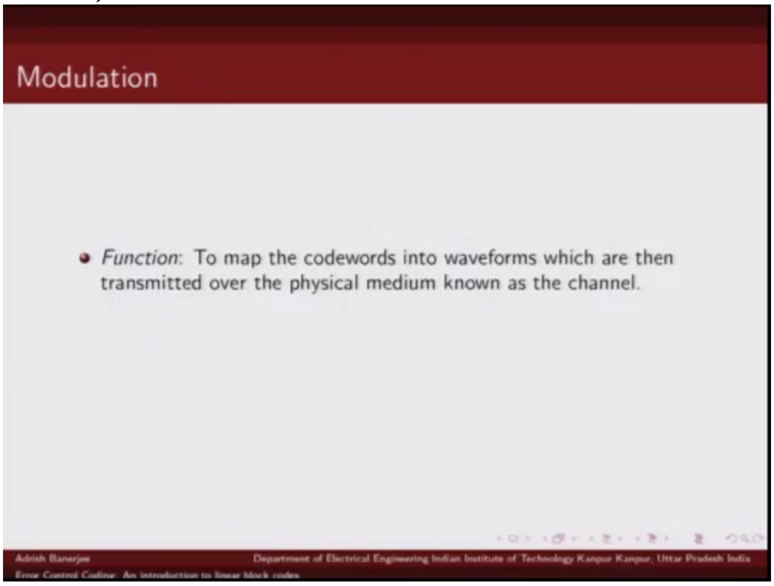
We use the term

(Refer Slide Time 08:56)



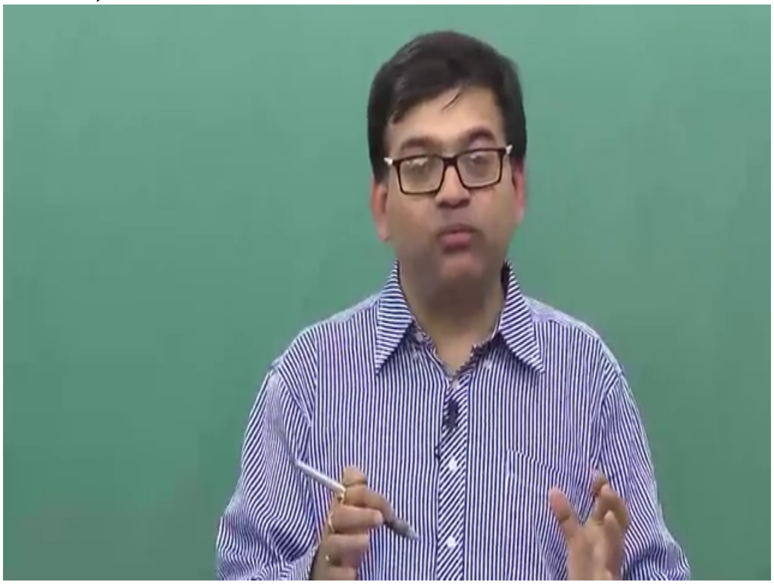
codeword to denote the encoded sequence out of the channel encoder

(Refer Slide Time 09:03)



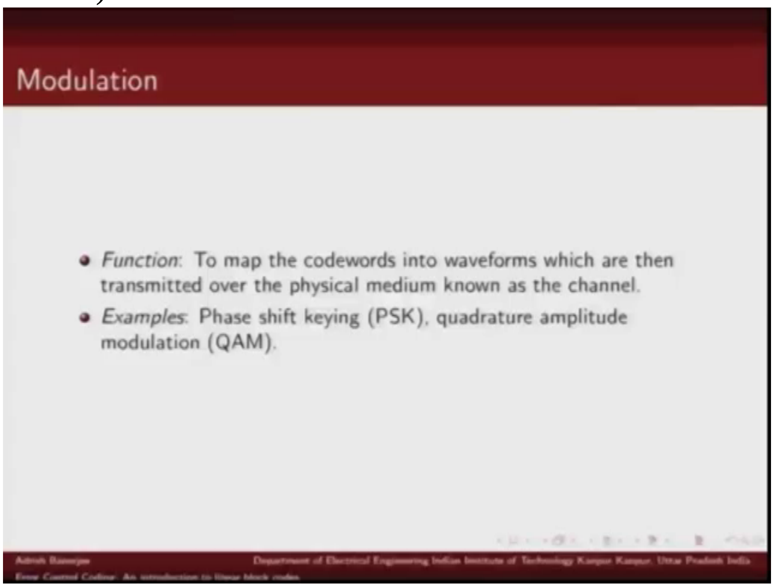
Now once we have the

(Refer Slide Time 09:06)



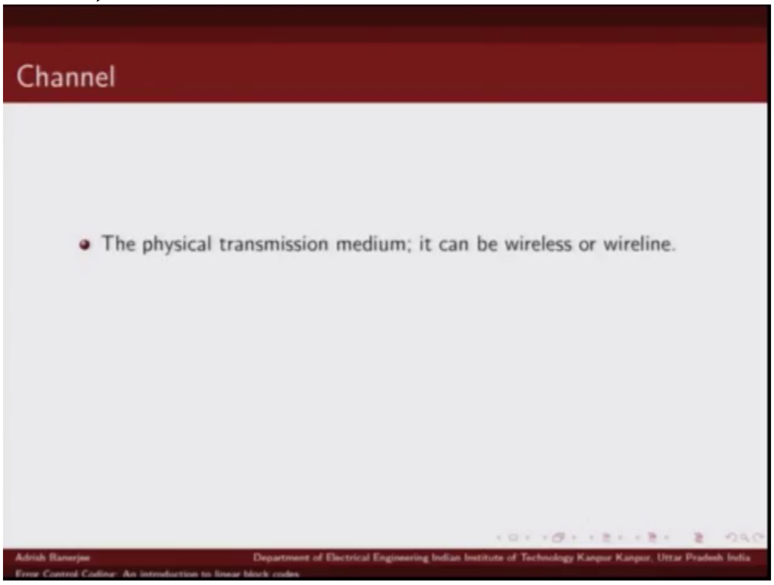
encoded sequence we would like to transmit that sequence over a communication link. So what we need to do is modulation; so modulation maps these codewords into actual waveforms which are sent over a communication link. And you are familiar with, I am sure you are familiar with digital modulation techniques such as B P S K binary phase shift keying, quadrature amplitude modulation or you have also studied about

(Refer Slide Time 09:35)



analog modulation schemes also. The next block

(Refer Slide Time 09:39)

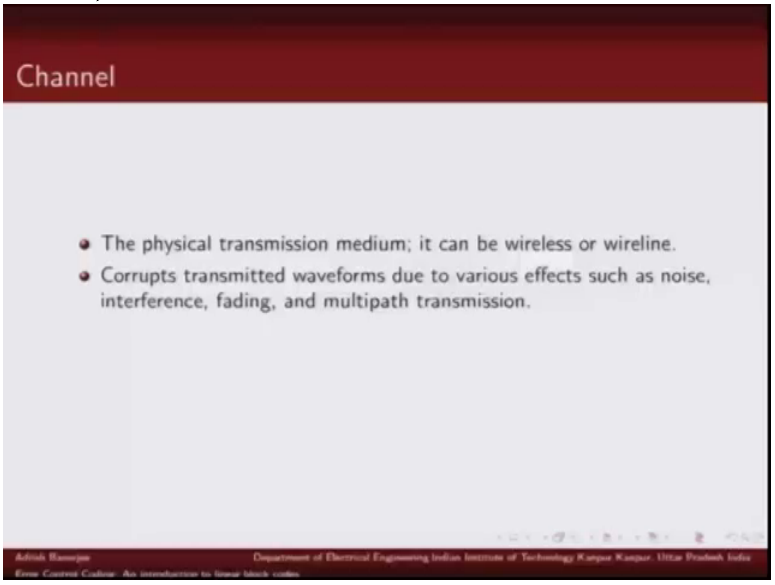


in our block diagram was a channel. So what was channel?

(Refer Slide Time 09:44)

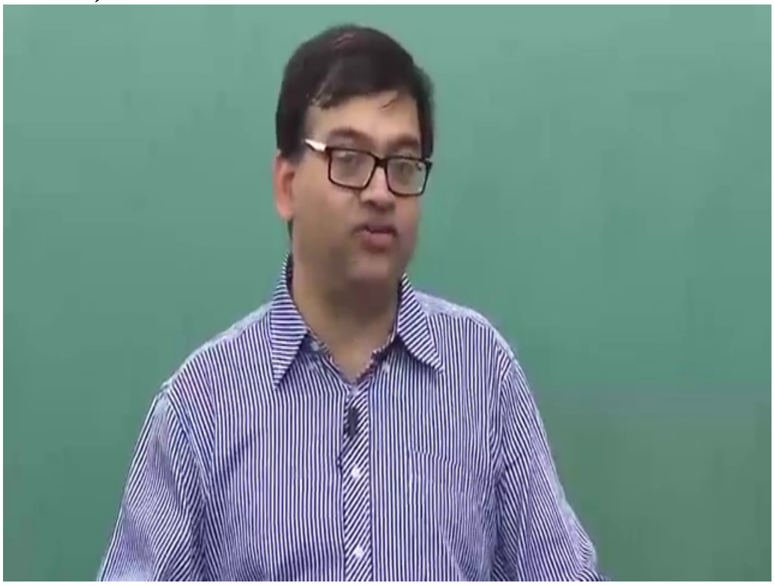
Channel was essentially our transmission medium over which we were sending our coded bits. So the physical transmission medium is known as channel and this could be a wired medium, for example our old-fashioned phones or it could be wireless medium.

(Refer Slide Time 10:08)



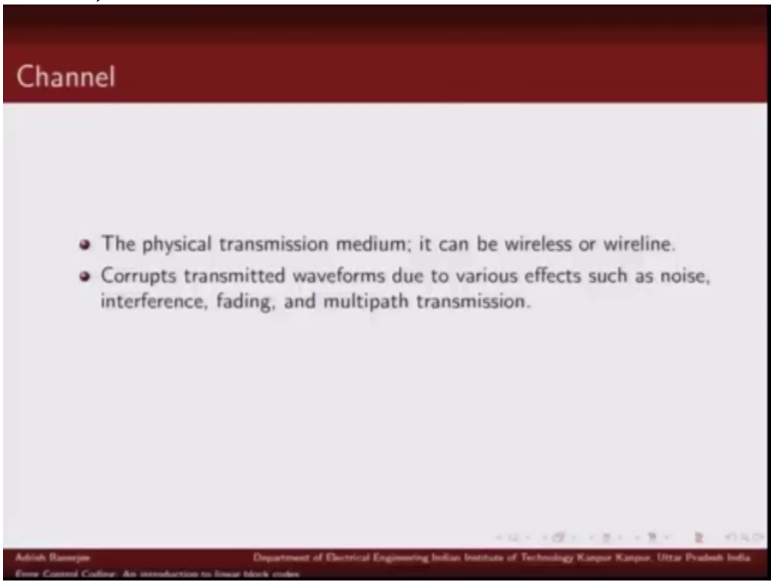
Now what effect does channel has on

(Refer Slide Time 10:11)



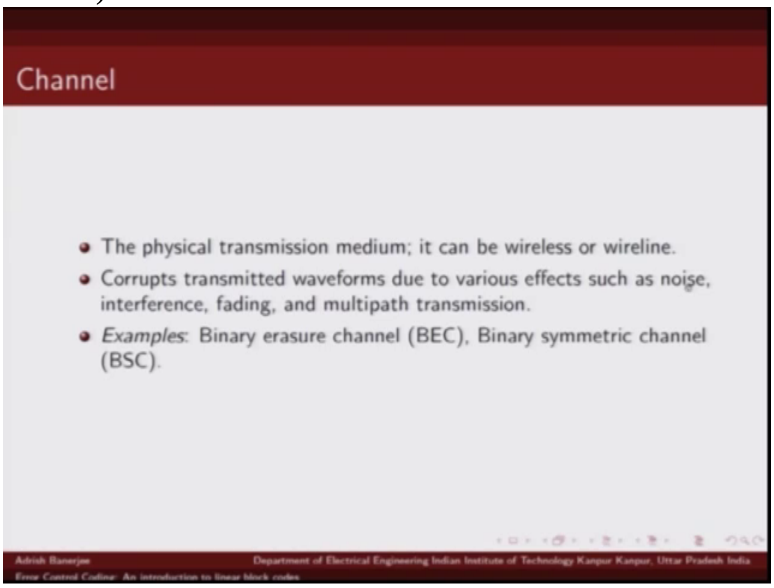
the transmitted sequence? As we know, basically the

(Refer Slide Time 10:15)



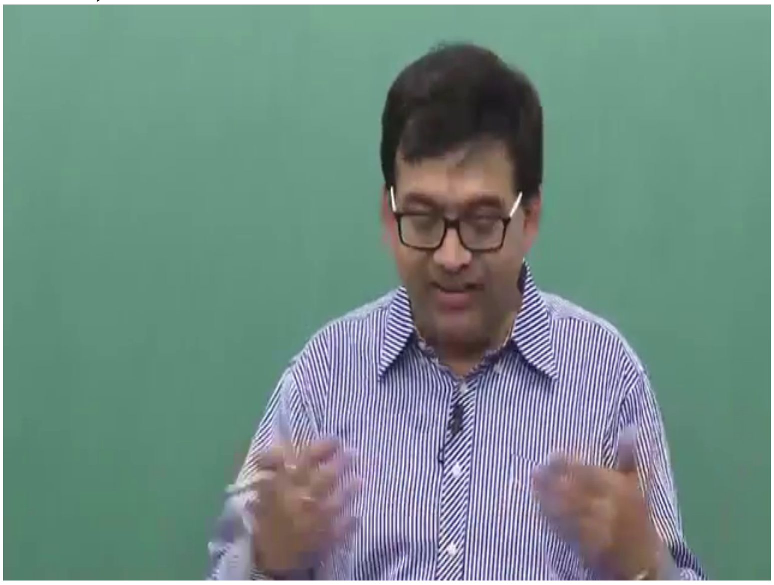
channel corrupts a signal in multiple ways. For example noise, this basically, noise detoriates.

(Refer Slide Time 10:22)



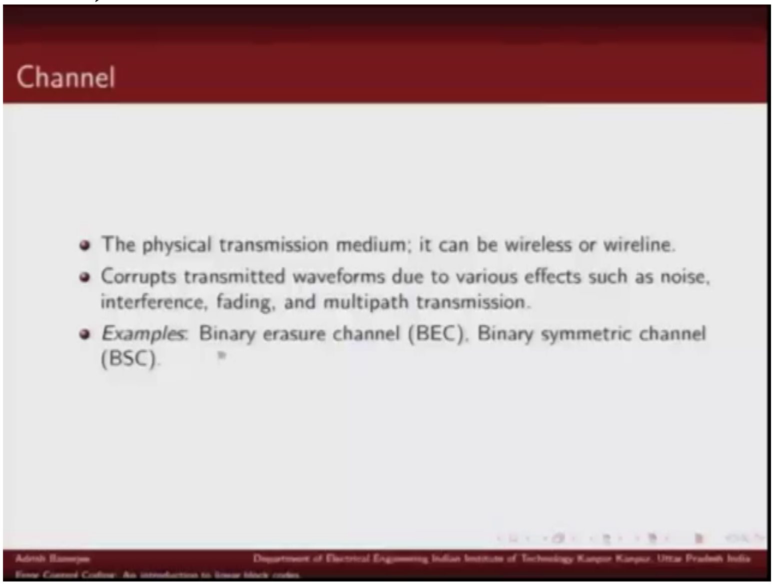
The signal interference is basically because multiple signals clashing with each other, fading basically if you are let's say under some building or something, you may not get a very good signal. That's an example of fading shadowing; multipath transmission when you have reflections from multiple buildings what you receive you send only one signal you get multiple copies with varying attenuation and delay spread. So essentially channel

(Refer Slide Time 10:53)



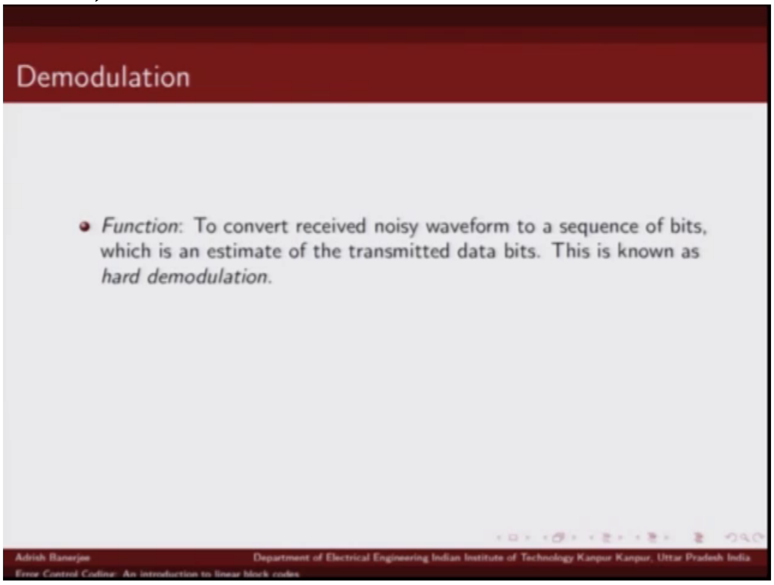
corrupts our signal that we are transmitting and it results in errors. And there are various models of channels which are very popularly used. In the last lecture we did talk

(Refer Slide Time 11:12)



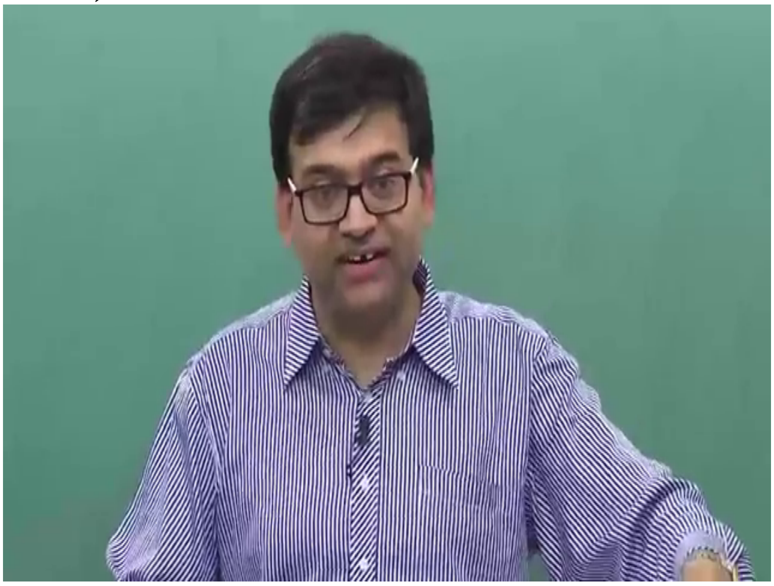
about binary erasure channel which is a very good example for, to model basically packet data communication system and another channel model we talked about was binary symmetric channel, it's a very simple channel model typically used to model additive white Gaussian noise channel with hard demodulation. So these are some examples of channel models. Now the next block

(Refer Slide Time 11:41)



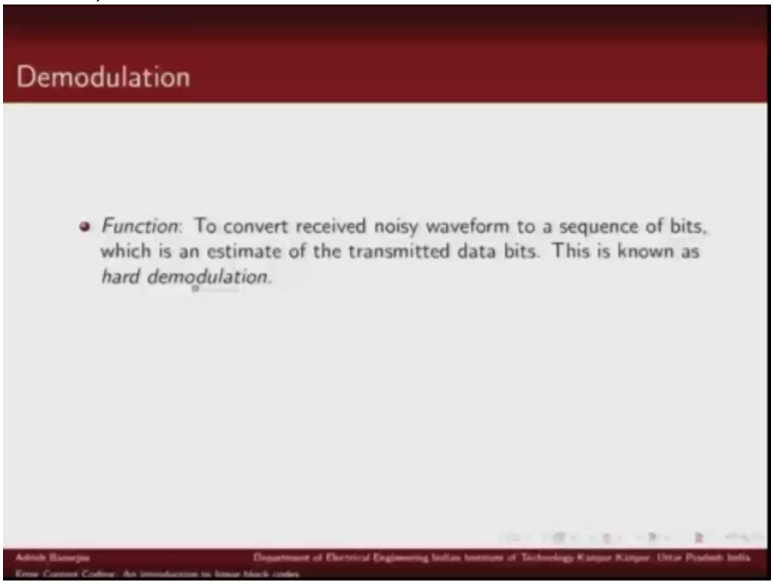
in our digital communication system was demodulation. So what does demodulation does? So as a result of passing through the communication channel

(Refer Slide Time 11:52)



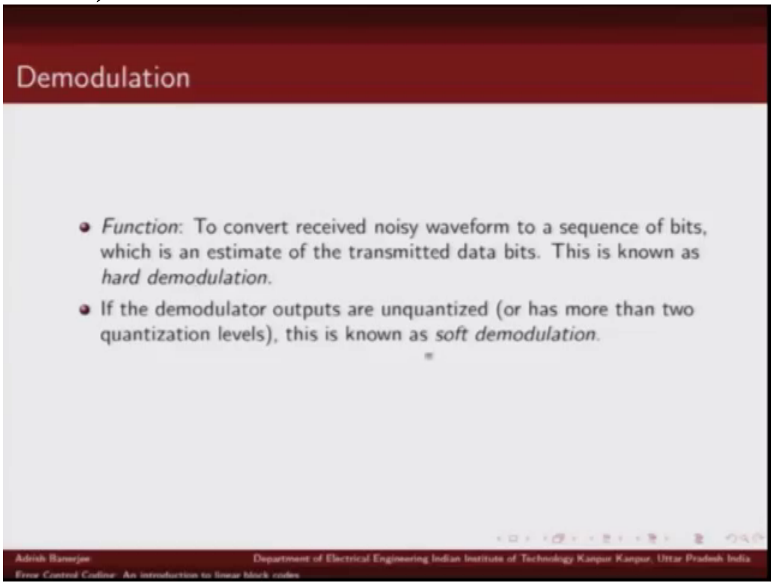
what we get is a very noisy waveform and from that noisy waveform essentially you want to get an estimate of what the transmitted bits are. Now when we try to demodulate or get back original bits zeroes and ones, that's basically what is known as

(Refer Slide Time 12:12)



hard demodulation There is also something called

(Refer Slide Time 12:15)

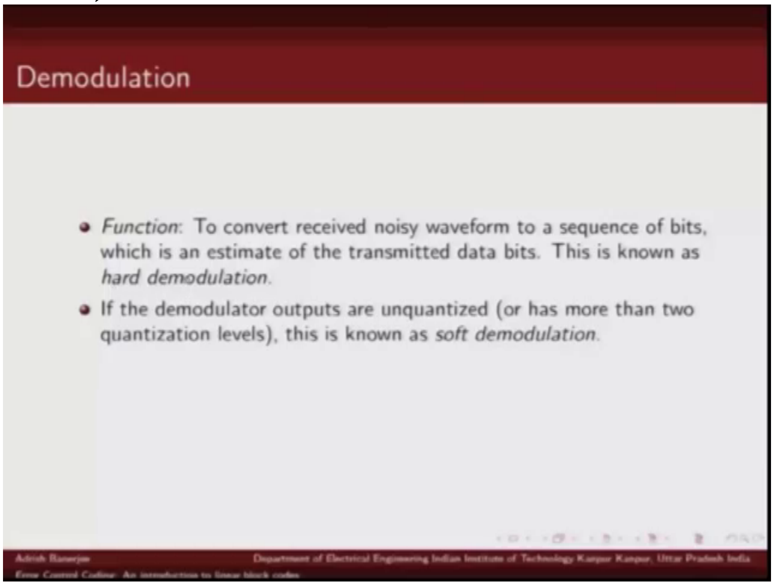


soft demodulation What if we are getting received noisy values of the received sequence, instead of quantizing them directly to zeroes and ones, if we keep it unquantized then this type of demodulation is known as soft demodulation. Now what is the advantage of soft demodulation? In hard demodulation we basically lose lot of information. For example let's say

(Refer Slide Time 12:42)

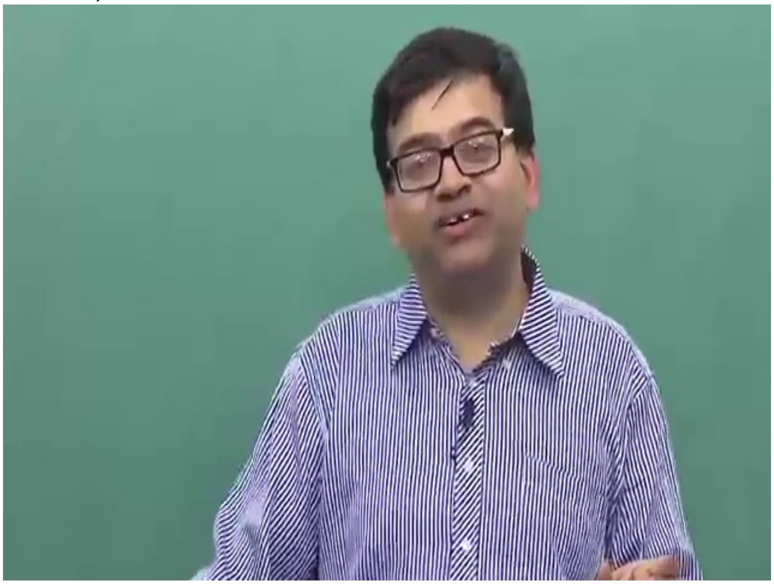
you consider binary transmission. You are sending zeroes and ones. And let's say you are using binary phase shift keying. So let's consider scenario where 0 is mapped to plus 1 and 1 is mapped to minus 1. And let's say you received, received symbol was point 0 1. Now if you were doing hard demodulation,

(Refer Slide Time 13:11)



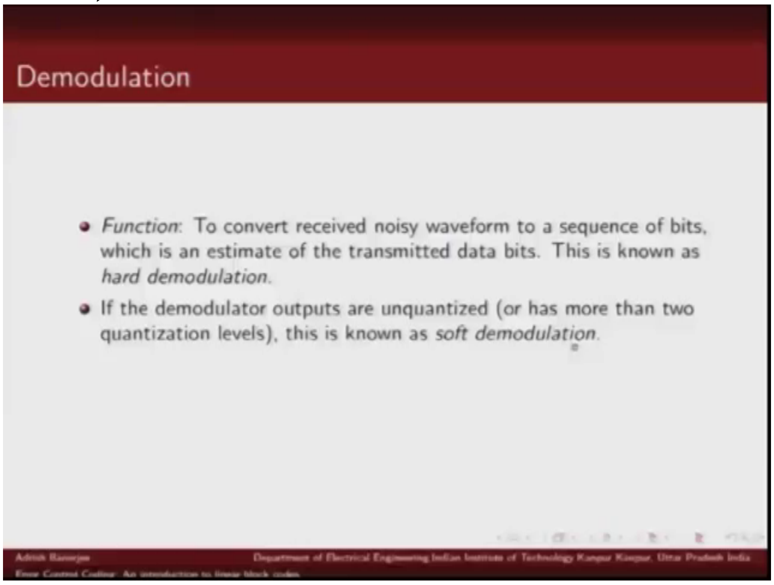
you would demodulate it to be, because its decision threshold would be 0, you will demodulate it to be 0 bit transmitted. But point 0 1 could very well be, could have been minus point 0 1 if the noise would have been little bit, like this thing, it's very close to 0, so in hard demodulation you lose that

(Refer Slide Time 13:35)



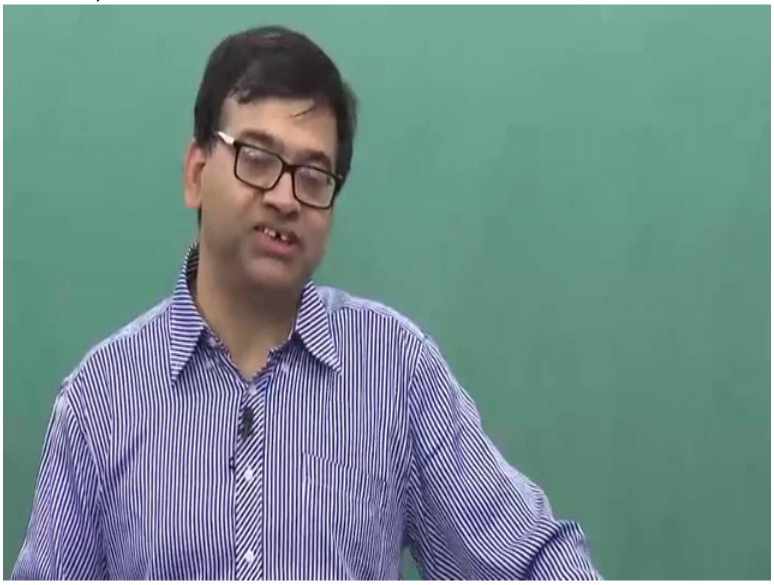
information So instead of just directly mapping our received sequence, it was 0 point 0 1 to 0, if you would have also kept some idea about of how sure we are, how close we think it is to 0, or 1 then that information is useful. And that is captured in soft demodulation.

(Refer Slide Time 13:57)



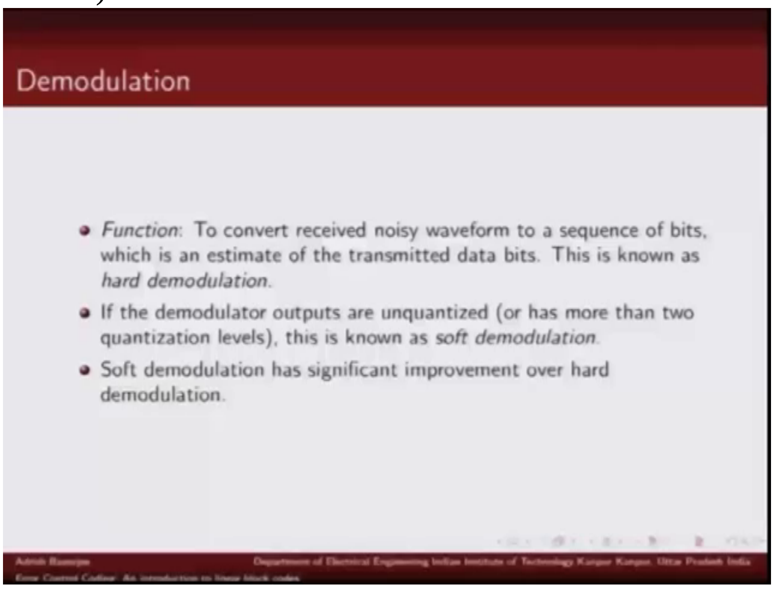
So in soft demodulation, we would not just map point 0 1 to 0, we will just say it is 0 with probability point 5 1 or 1 with probability point 4 9. Just like, I am just giving an example. So we lose some information when we go for hard demodulation. So typically, basically when you will see basically, if we do soft demodulation

(Refer Slide Time 14:31)



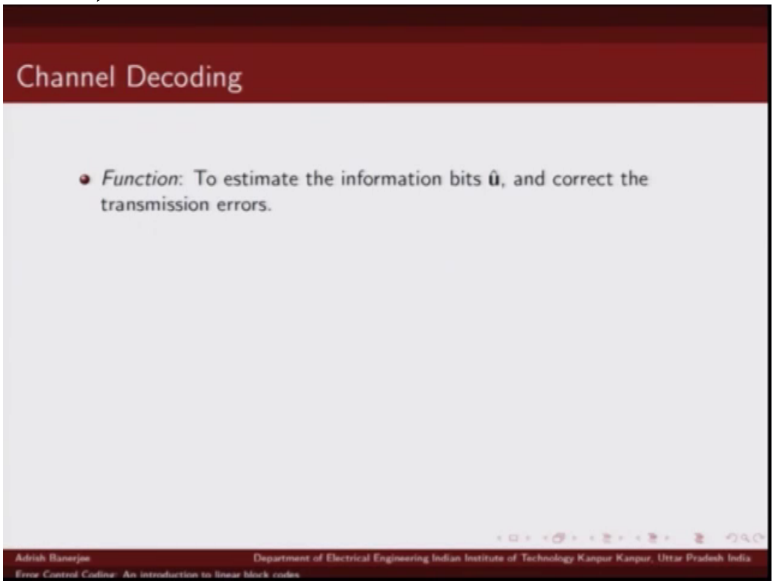
our performance is roughly 2 d B better than if we were just doing hard demodulation. So most of the

(Refer Slide Time 14:39)



modern coding techniques we will basically be doing soft demodulation and we will pass on these bits, soft values to the decoder, channel decoder and we will then ask it to do error correction or error detection. Software demodulation

(Refer Slide Time 14:57)

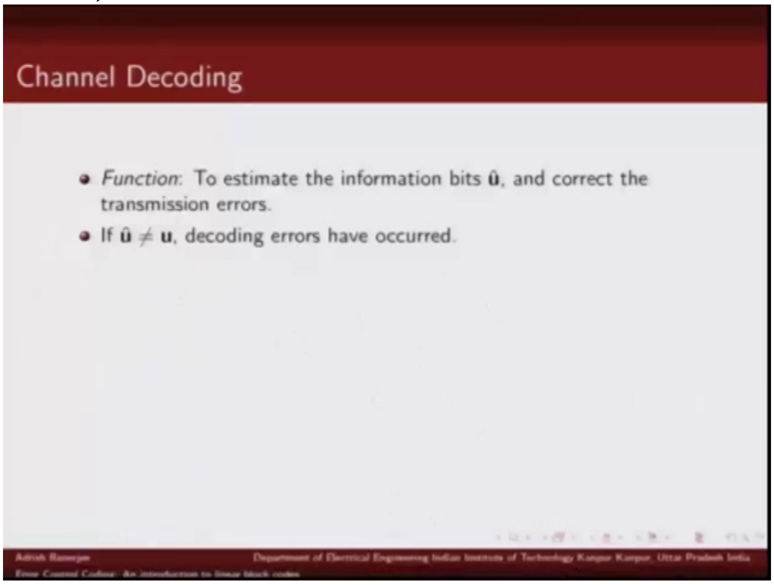


we have channel decoding as I said; the channel decoder objective is to estimate what was

(Refer Slide Time 15:07)

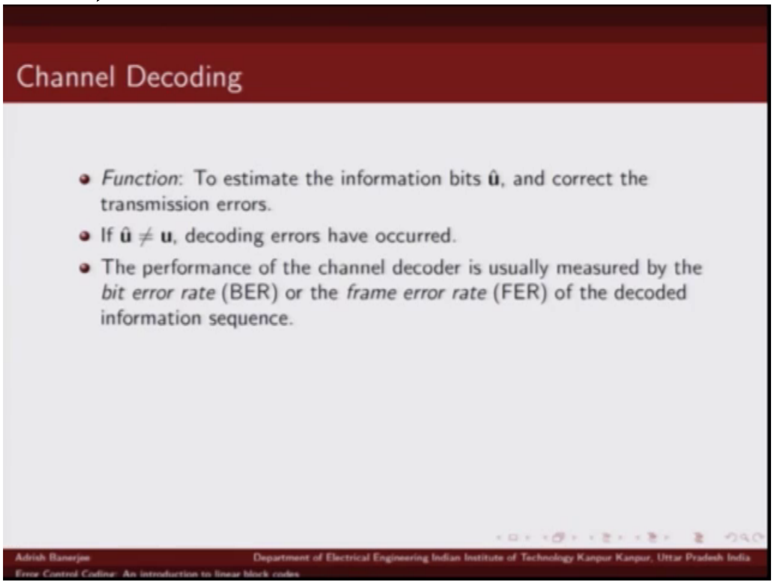
the transmitted information sequence from the received coded sequence and

(Refer Slide Time 15:16)



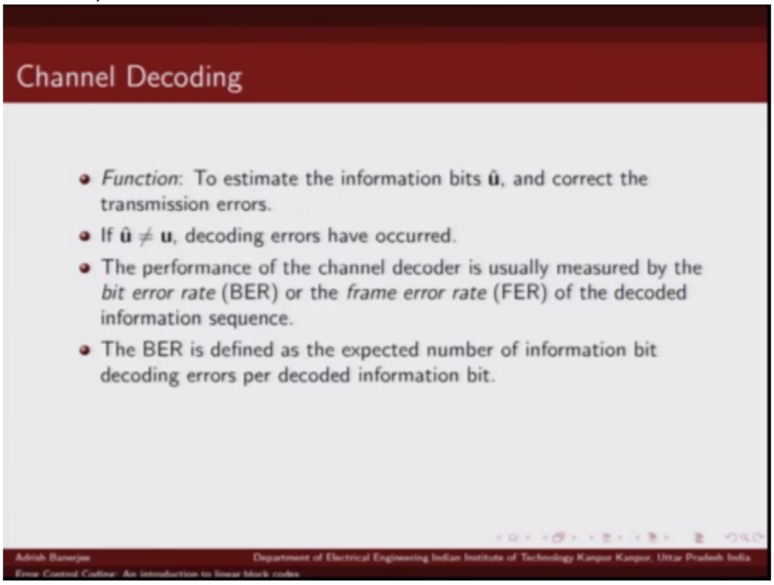
its objective is also to correct any errors which was caused by transmission error. So when do we say an error has occurred, if our decoded bit is not same as transmitted bit, we say that decoding errors have occurred.

(Refer Slide Time 15:40)



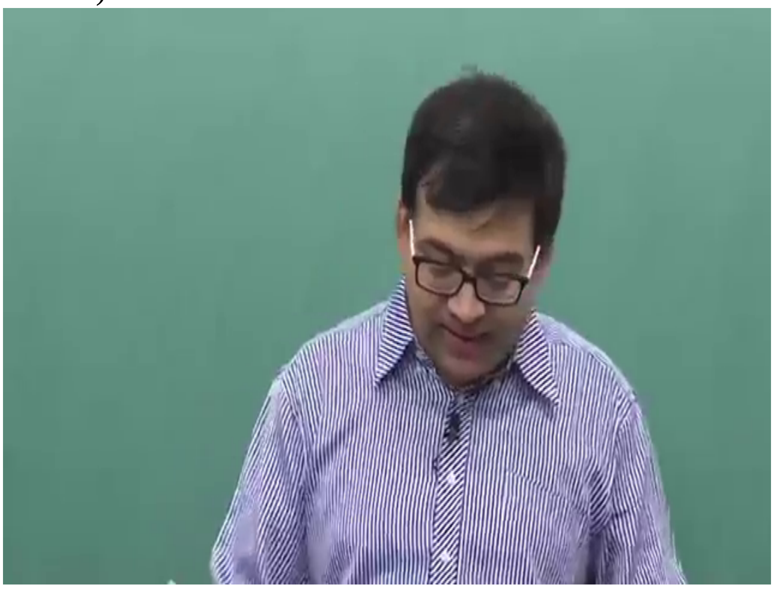
And how do we quantify these decoding errors? We quantify these decoding errors using what we call bit error rate or frame error rate. So performance of error correcting codes is typically evaluated using these bit error rate and frame error rate. So what is bit error rate and frame error rate?

(Refer Slide Time 16:05)



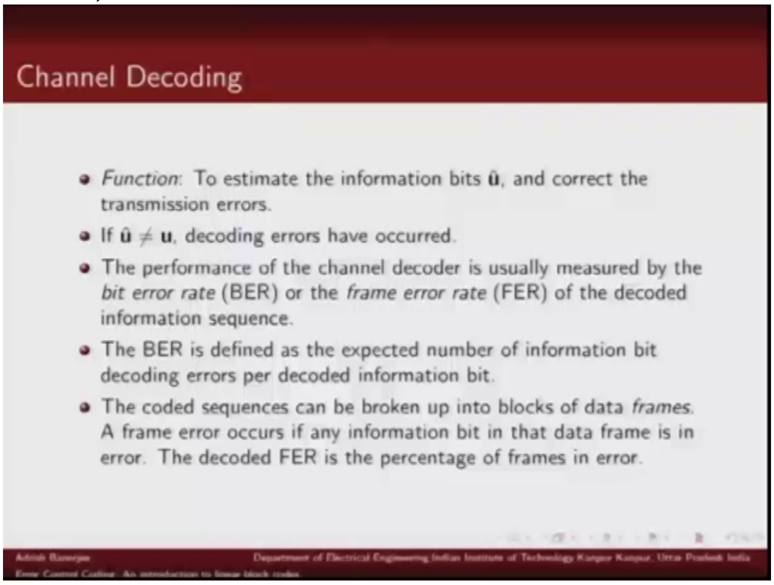
So we define bit error rate as

(Refer Slide Time 16:08)



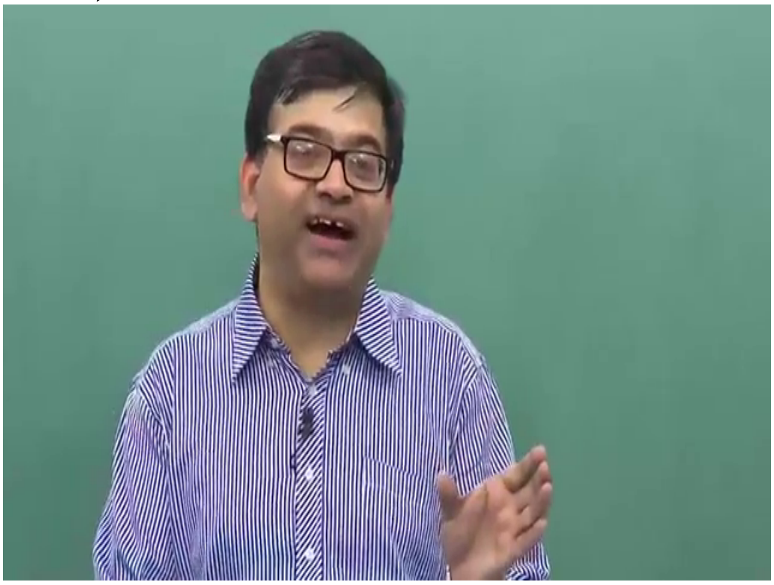
expected number of information bits that are decoded to be in error. So for example if you transmit n information symbols and out of them x basically bits are received in error. So the bit error rate would be x by n. Similarly

(Refer Slide Time 16:34)



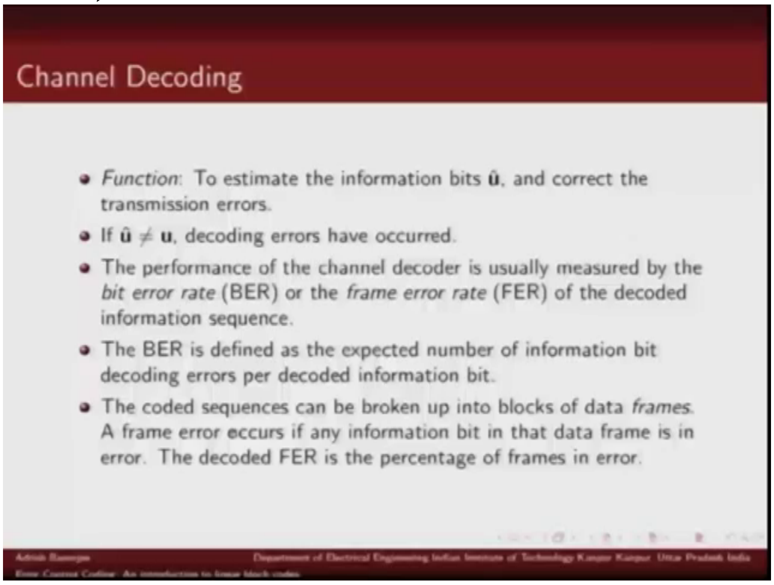
when we are talking about packet data communication, we are interested in, so for example if any of the bits in the packet, whole packet is in error we say packet is in error, or the frame is in error; so frame error rate is defined as

(Refer Slide Time 16:50)



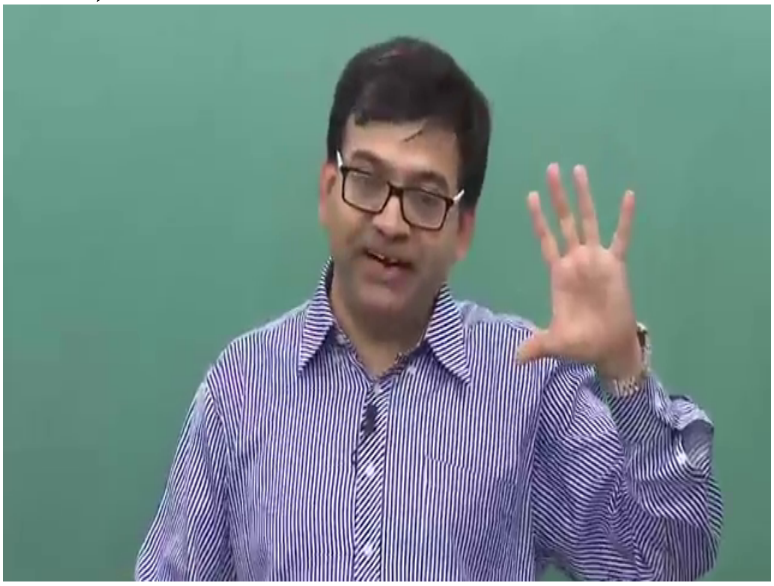
number of frames or number of packets in error divided by total number of packets that we have transmitted. So a frame error rate occurs

(Refer Slide Time 17:01)



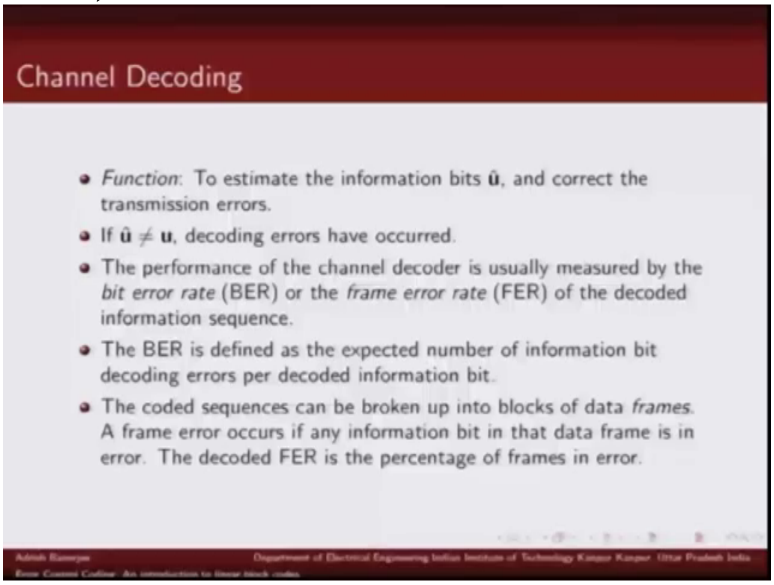
if any of the bits in your data frame or in your packet is received in error. So it's a percentage of frames that are in error, this is known as frame error rate. So when we compare performance of different error correcting codes what we do is we plot on the y axis bit error rate

(Refer Slide Time 17:25)



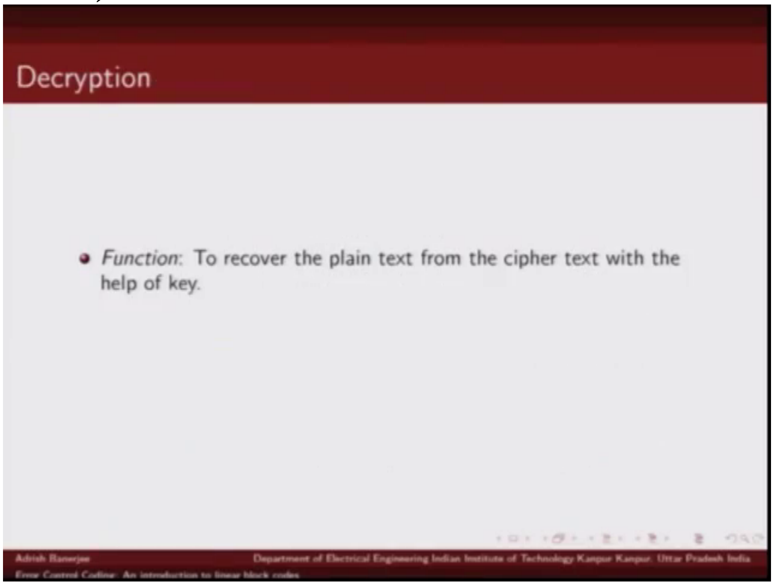
or frame error rate versus S N R, signal to noise ratio basically. So how is the performance of, when we increase signal to noise ratio, how does our error correcting codes perform, that's how basically we compare two error correcting; that is one way of

(Refer Slide Time 17:48)



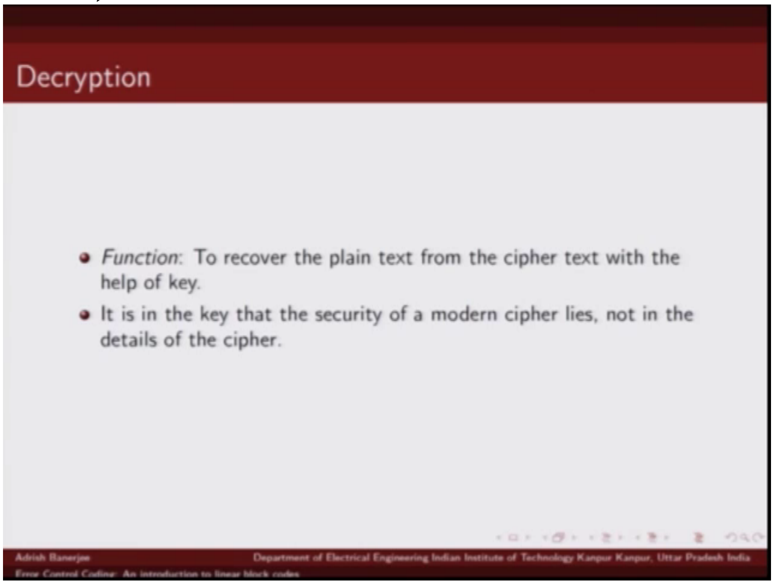
comparing two error correcting codes After

(Refer Slide Time 17:52)



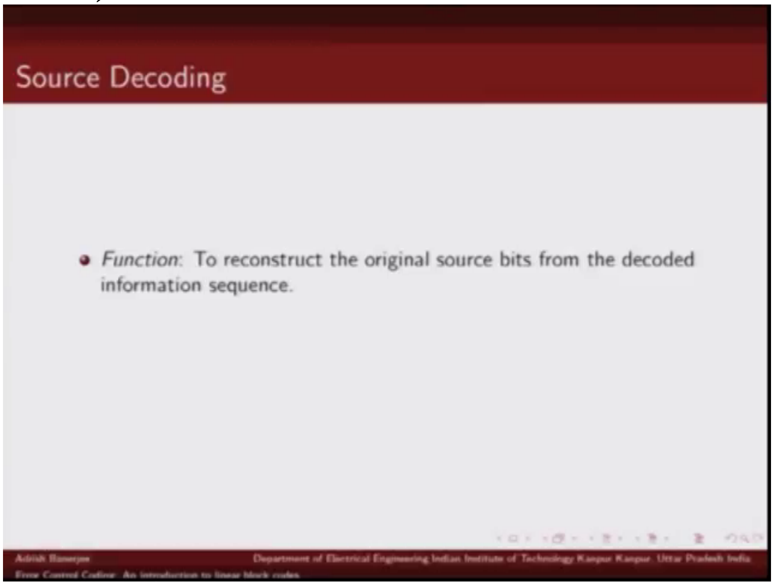
channel decoder, we have this de-encryption block and as I said its objective is to recover back the plain text from the cipher text

(Refer Slide Time 18:03)



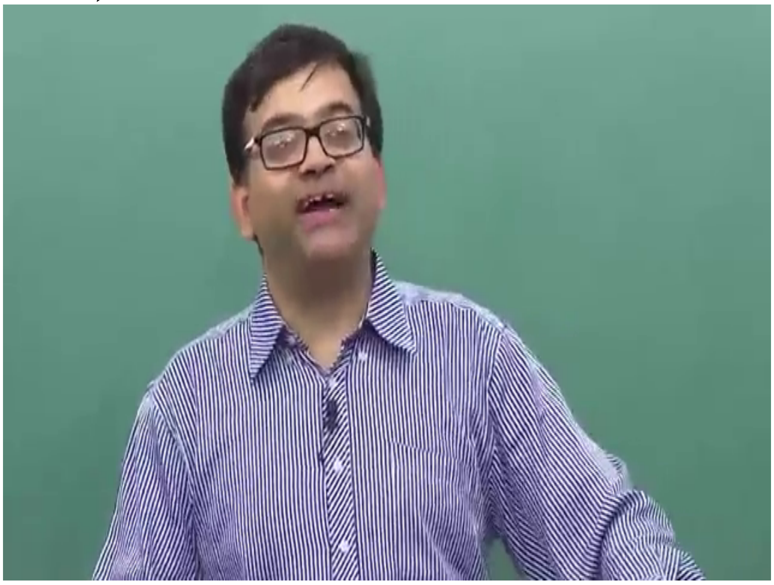
and essentially the security of the system lies in the key. And

(Refer Slide Time 18:10)



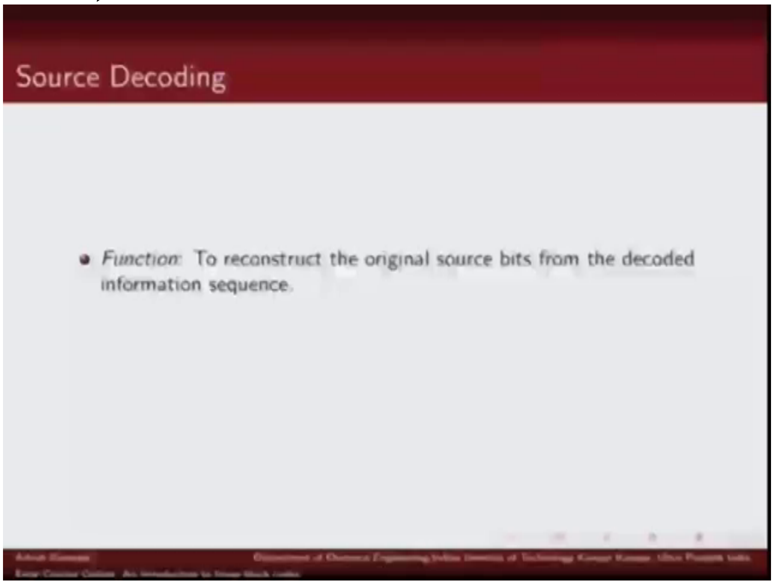
finally we would

(Refer Slide Time 18:12)



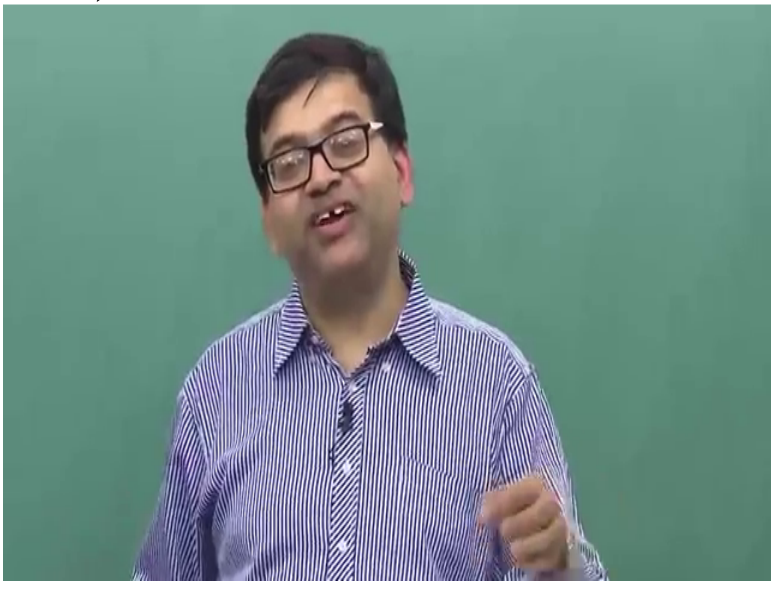
like to get back our original source because remember, in source encoder what we did was we did data compression. So we have been processing this compressed data. So from the compressed data, we would like to get back our original source and that's

(Refer Slide Time 18:27)



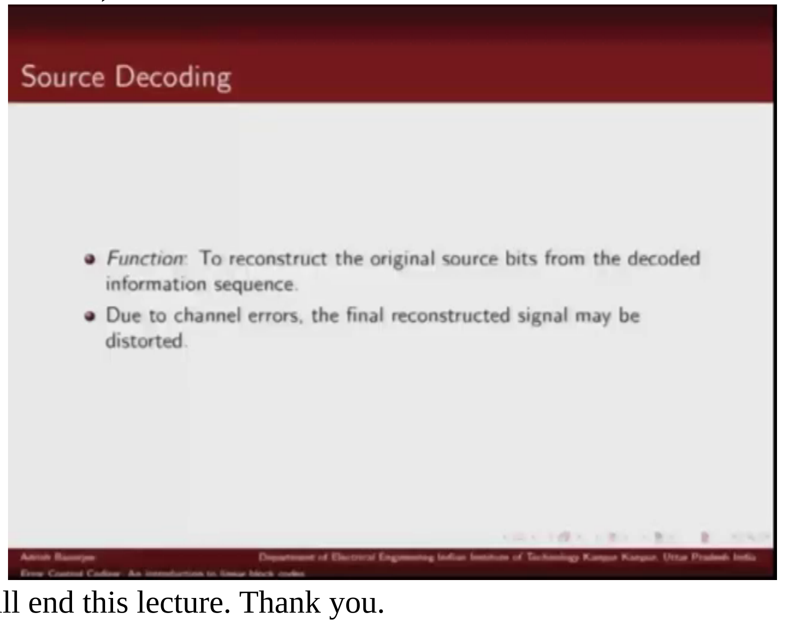
basically the source decoder So it

(Refer Slide Time 18:30)



reconstructs back the original source bits from the decoded information sequence And of course if there are errors then our final you know decoded sequence would not be same as what was transmitted.

(Refer Slide Time 18:46)



So with this I will end this lecture. Thank you.