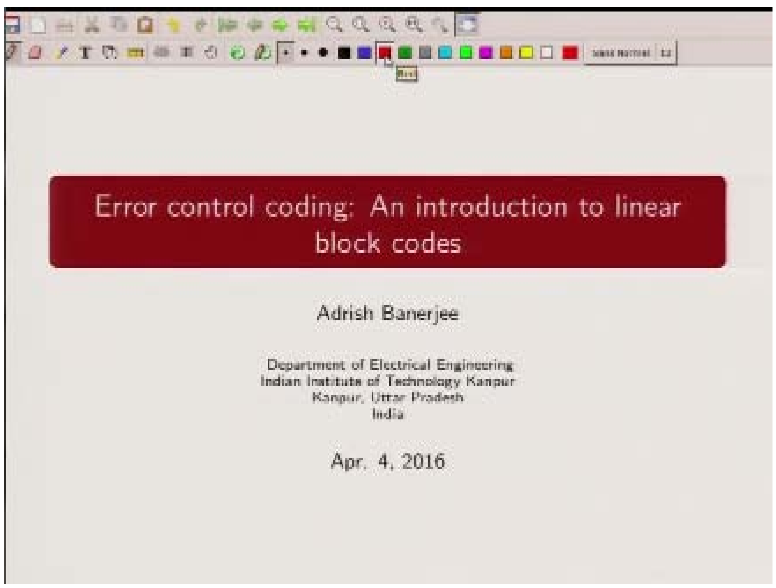
Indian Institute of Technology Kanpur
National Programme on Technology Enhanced Learning (NPTEL)
Course Title
Error Control Coding: An Introduction to Linear Block Codes

Lecture -10
Applications of Linear Block Codes

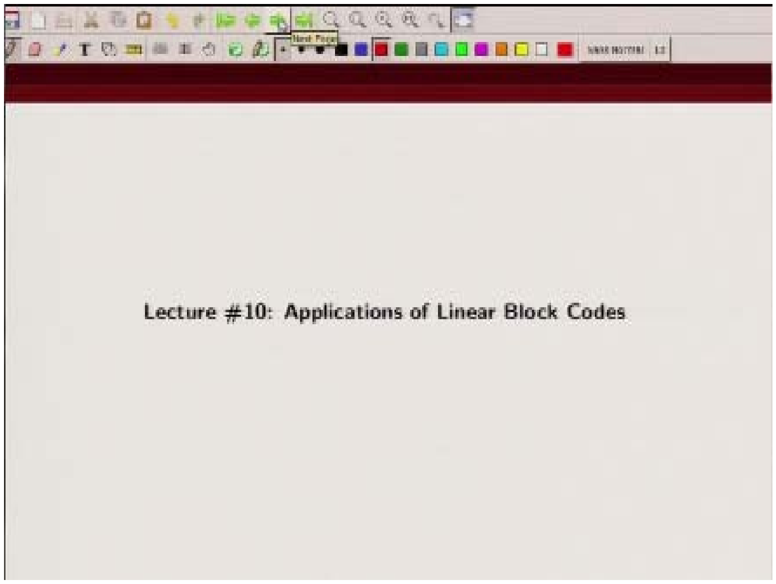
by
Prof. Adrish Banerjee
Department of Electrical Engineering, IIT Kanpur

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

(Refer Slide Time: 00:19)

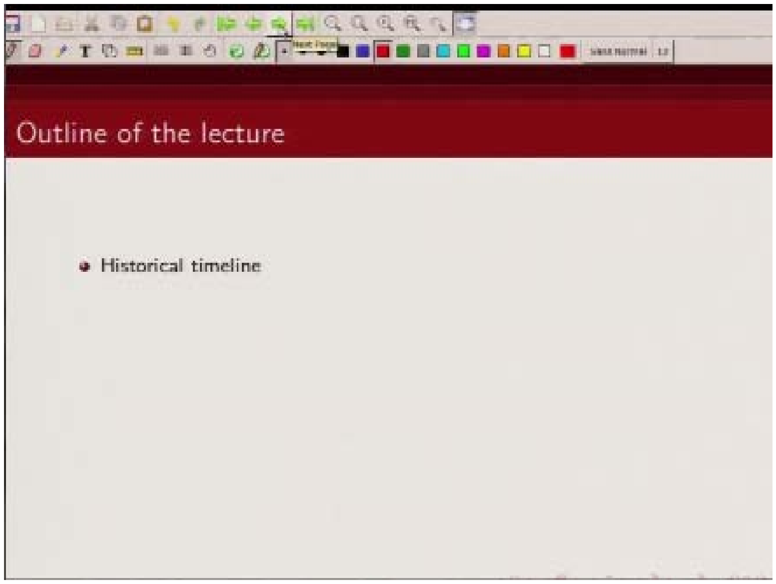


(Refer Slide Time: 00:22)



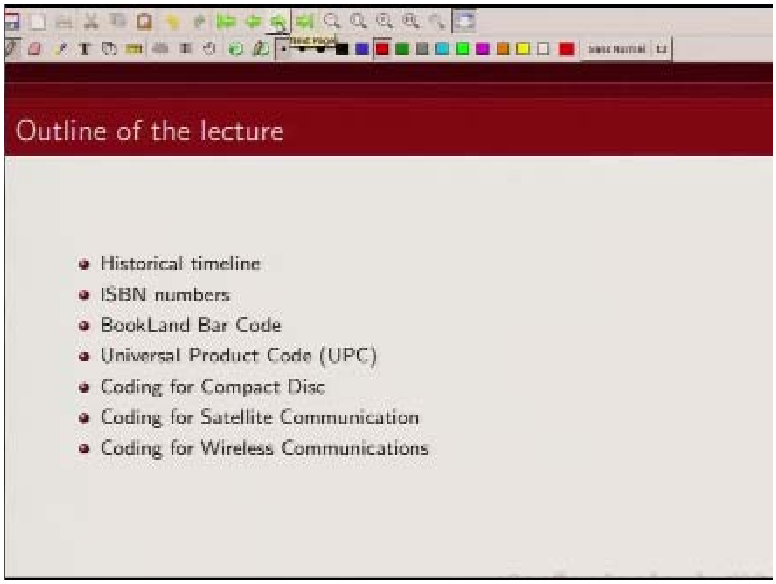
In this lecture we are going to talk about some of the applications of linear block codes.

(Refer Slide Time: 00:25)



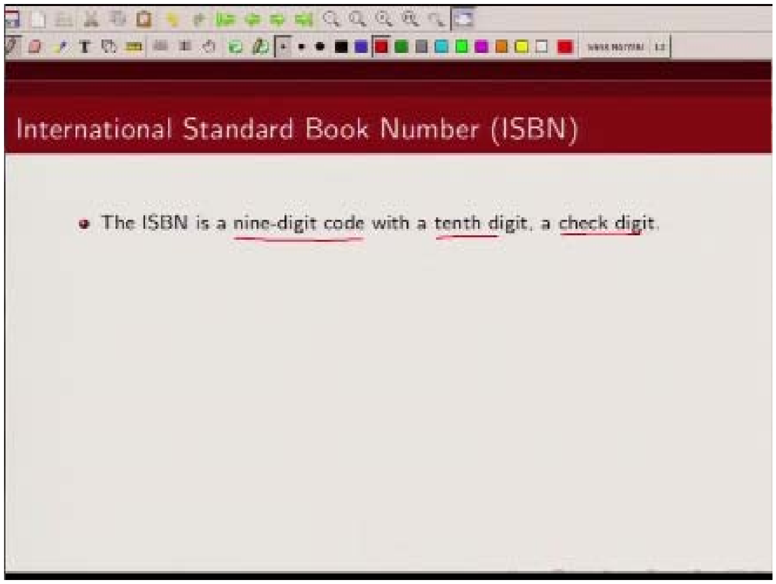
So we will first start with a brief historical timeline of how coding theory has progressed over years.

(Refer Slide Time: 00:35)



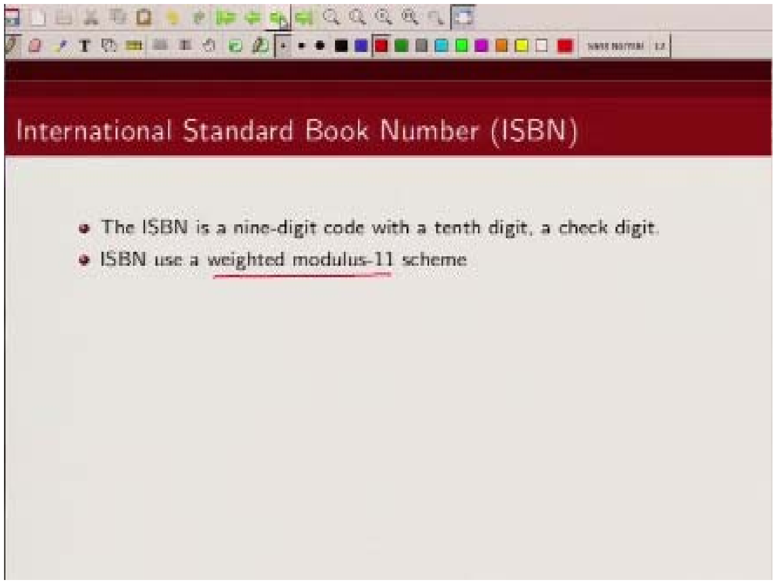
Then we will start with some basic examples, we start off with ISBN number and BookLand bar code. Then we will talk about Universal Product Code and European article number. Then we will talk about some of the applications of error control coding in practice. So we will start off with applications in compact disc, then we will talk about applications in satellite communication as well as deep space explorations and finally we will conclude with applications and wireless communications.

(Refer Slide Time: 01:18)



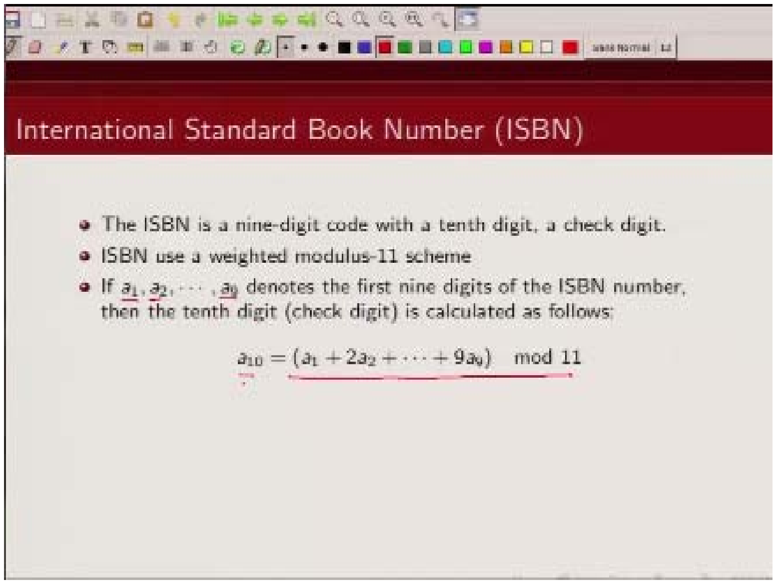
So what is an ISBN number? An ISBN number is a nine digit code with a 10 digit which is a check digit. So it is a 10 digit number, you have a nine digit code plus one bit which is a check bit. So how is this check bit calculated?

(Refer Slide Time: 01:40)



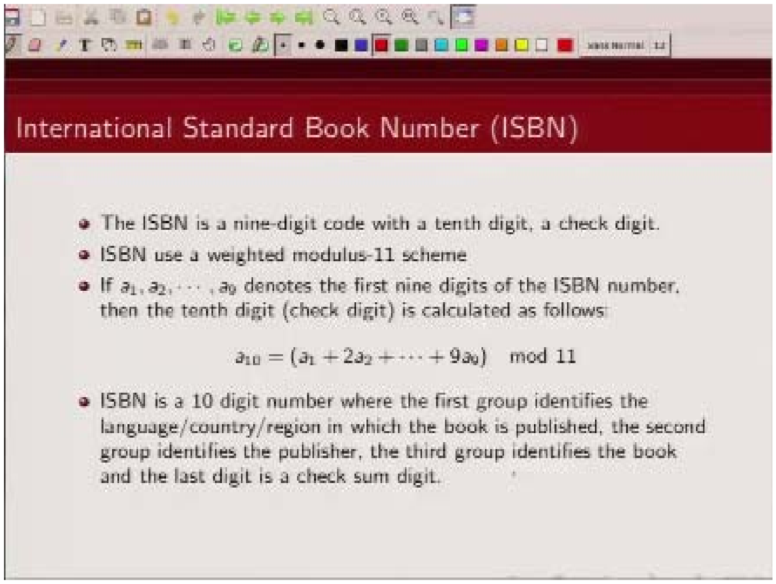
This check bit is calculated as follows. It uses a weighted modulus-11 arithmetic.

(Refer Slide Time: 01:47)



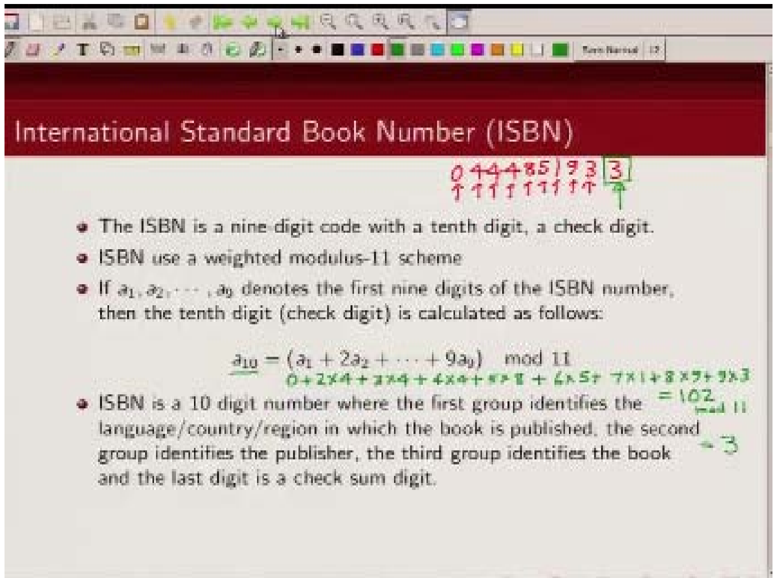
So let us denote the nine bits of ISBN number by a_1, a_2, a_3, and a_9, then the tenth bit is calculated as follows. So $a_1+2a_2+3a_3+4a_4$ up to $9a_9$ modulus-11 will give us the tenth bit. So the tenth bit is actually a check bit.

(Refer Slide Time: 02:20)



An ISBN is typically a 10 digit number where the first group identifies the country region where the book is published, the second group identifies the publisher and third group identifies the specific book and finally the last bit is actually a check bit. So let us take an example, let us take this book. This is one of our reference book that we are using, The Theory of Error Correcting Codes, by MacWilliams and Slone. So if we look at the ISBN number for this book.

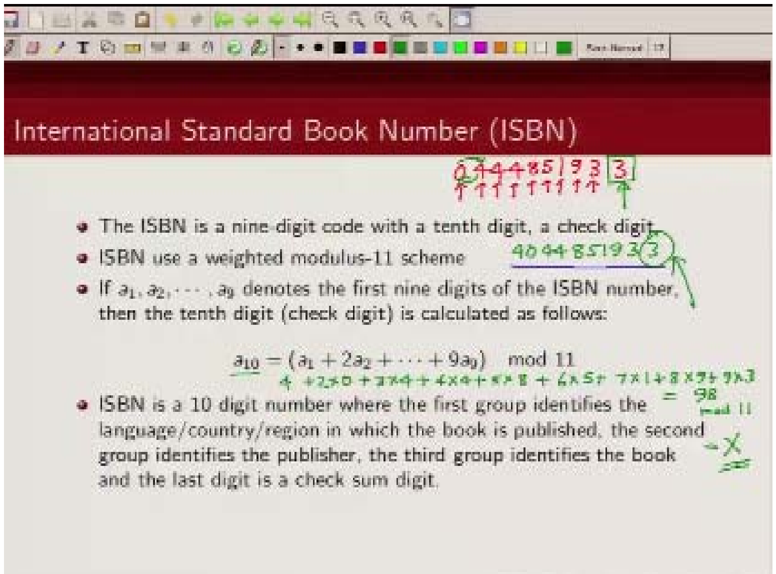
(Refer Slide Time: 03:00)



The ISBN number for this book is 0444851933 okay. So let us check whether – so this is the first bit, second bit, third bit, fourth bit, fifth bit, sixth bit, seventh bit, eighth bit, and this is the ninth bit, and this is our check bit okay. So let us check whether this is correct or not. So how is the check bit calculated? It is a_1 so a_1 in this case is 0, a2a_2 that is 4, 3a_3 that is 4, 4a_4 which is again 4 plus 5a_5+6a_6+ 7a_7+ 8a_8 and 9a_9 okay.

And this comes out to be 102, now 102 modulus 11 is, so 11x9=99 so it will be 3 modulo 11. So this 102 modulos 11 is going to be 3 and that is precisely our check bit is, okay. Now the interesting part about these ISBN numbers are that they can detect transposition error.

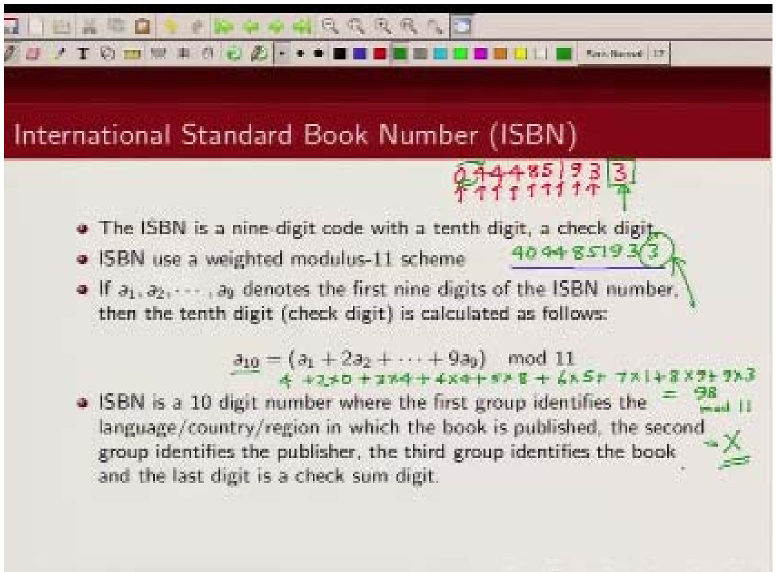
(Refer Slide Time: 04:59)



So what is the transposition error? So let us say when you are reading this ISBN number with a bar code reader the most common mistake that happens is the adjacent bits are read wrongly. So for example this is 0444851933, if it is read like 4044851933 that means these two bits are changed, this is transposition error. Then this particular code can detect such errors, we can verify this, let us say this is the number which is being read by the bar code reader.

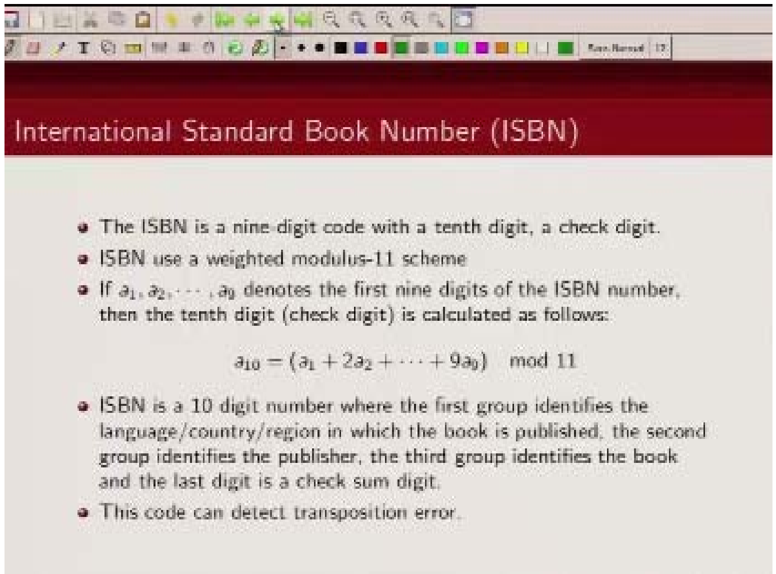
We try to find out the check bit in this case, so this would be the difference would be here it is these two things will be changed, because these two bits got, so it will be $4+2_0$ and that would be 98. So this will be 98 mod 11 and what is 98 mod 11, 11x8=88 so that is 10, so that would be X. Note that this check bit is not matching with the check bit calculated this way. So you can see this particular ISBN code can detect transposition error.

(Refer Slide Time: 06:40)



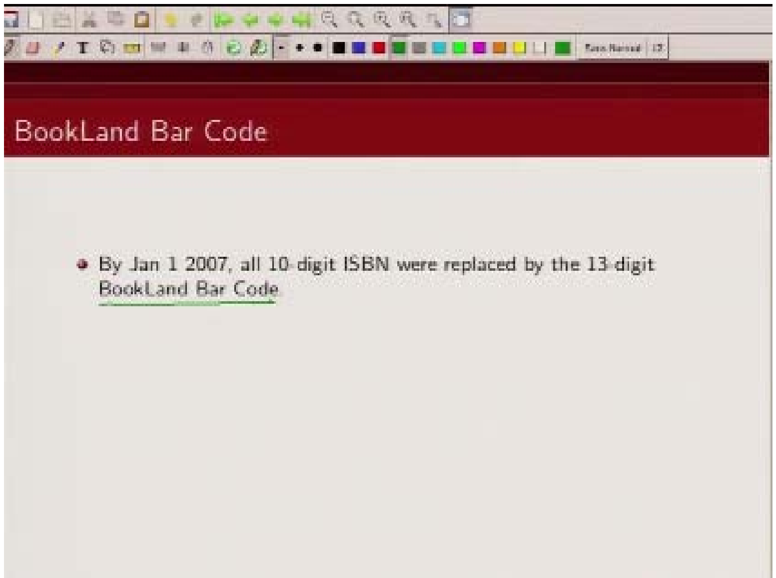
The correct number was 0444851933 and these two bits got interchanged, transposition error happened and you can see using this ISBN number we are able to detect such errors.

(Refer Slide Time: 07:00)



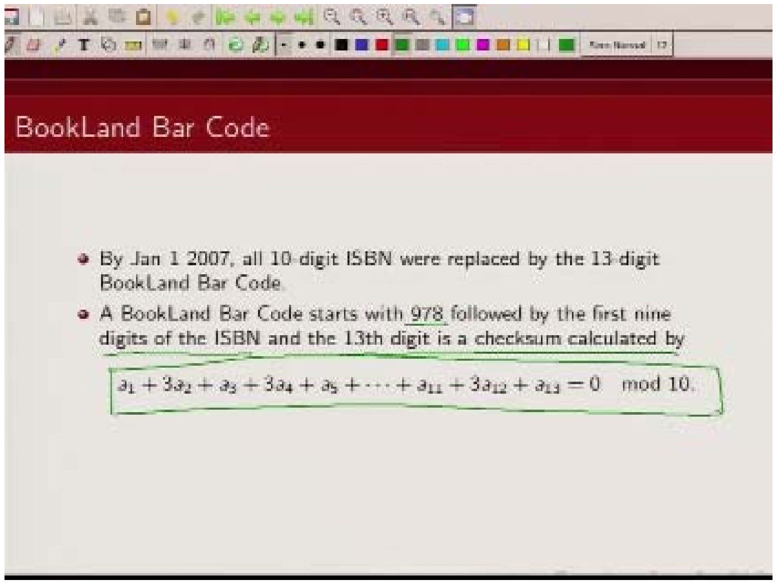
Okay.

(Refer Slide Time: 07:01)



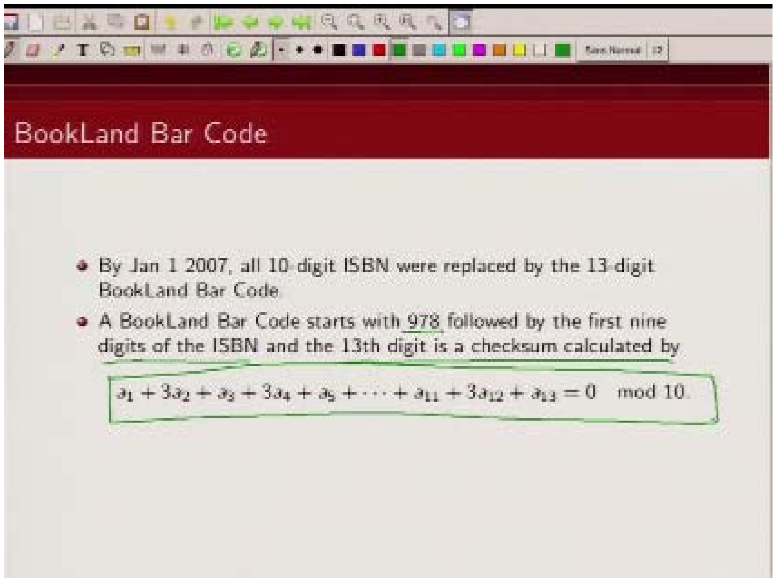
Now by 2007 all these ISBN numbers were replaced by this 13 digit BookLand Bar Code number and what are these BookLand Bar Code numbers?

(Refer Slide Time: 07:16)



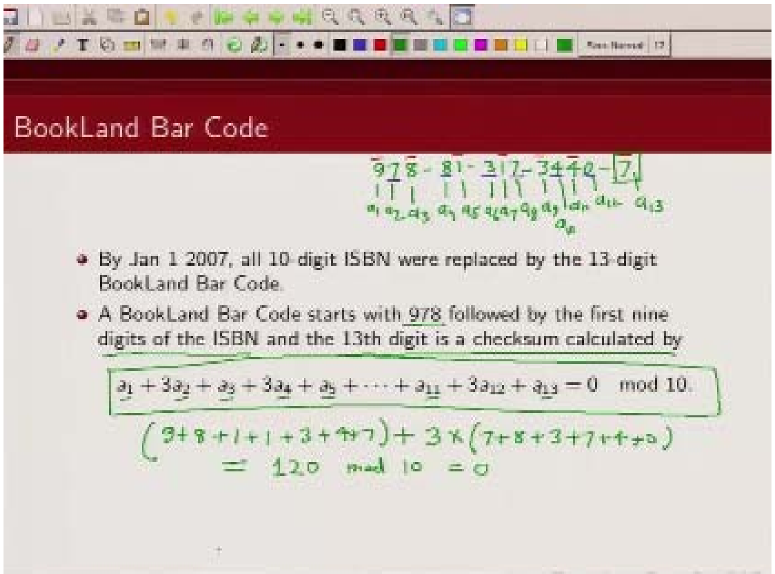
So a BookLand Bar Code number starts with 978 and then it is followed by nine digits of ISBN number and the 13th bit is a check sum which is calculated in this particular way okay. This is how we are computing the 13th bit which is the check sum bit.

(Refer Slide Time: 07:43)



Now let us look at an example, so let us consider our book. This error control coding by Lynn Costello and what is its BookLand Bar Code number?

(Refer Slide Time: 07:58)

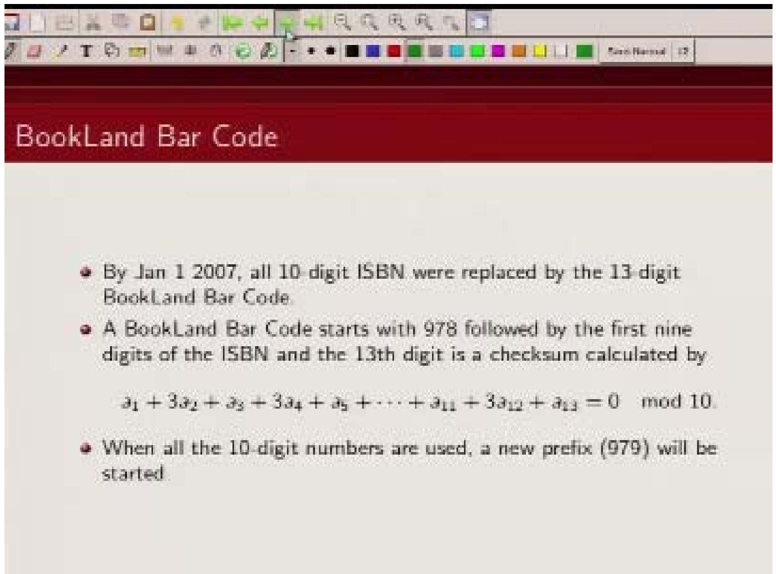


That is given by 978-81-317-3440-7 so this is the 13 bit digit number which is given by – which is the ISBN number 13 digit BookLand Bar Code number of this text book that we are using. Now so this is my a_1, this is my a_2, this is a_3, this is my a_4, a_5, this is a_6, a_7, a_8, a_9, this is a_{10}, this is a_{11}, this is a_{12}, and this is a_{13}. So let us look at whether we are getting the same check bit as plotted by this.

So let us – so you can see here what we are doing is, all odd digit numbers we are adding them up right. And all even numbers we are adding them up and multiplying them by 3. So this would be then 9+ -- let me first take all the odd bit numbers. Odd bit numbers are this, the ones I am marking by, these are my odd locations right. So this is (9+8+1+1+3+4+7)+3 times the even numbers, even numbers are this (7, 8, 3, 7, 4, 0) so this is (7+8+3+7+4+0) okay.

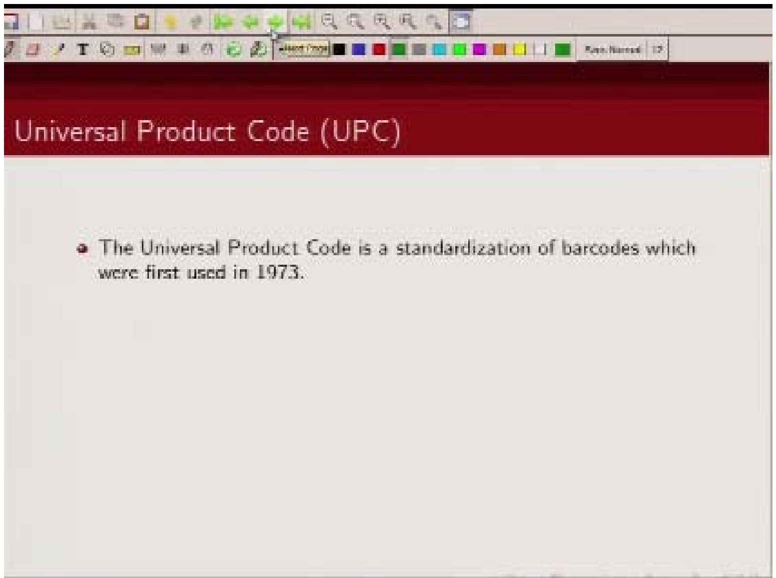
And this comes out to be 120 mod 10 which is 0. So as you can see from this example that this check bit is correct, because a_{13}+ this should be 0 mod 10. So this is the example of a BookLand Bar Code. And then all the books now have these 13 digit ISBN number.

(Refer Slide Time: 10:55)



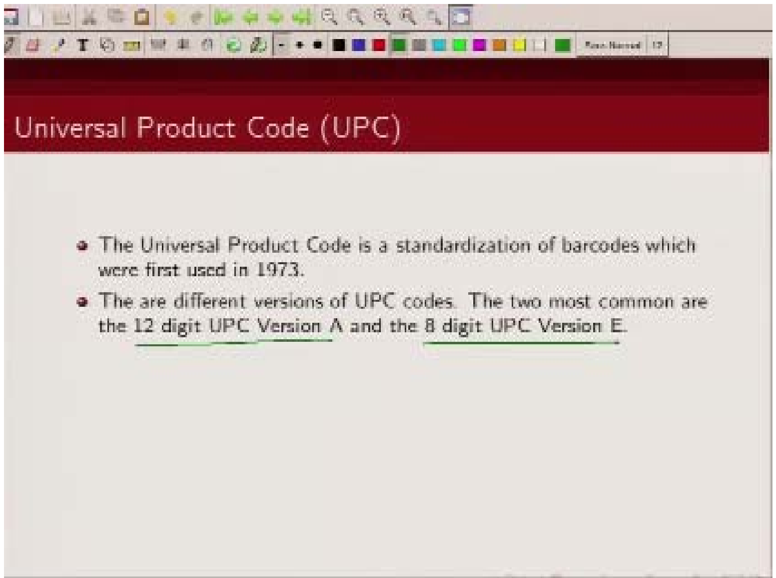
Now once they will get exhausted with this 978 number they will start numbers with 979.

(Refer Slide Time: 11:02)



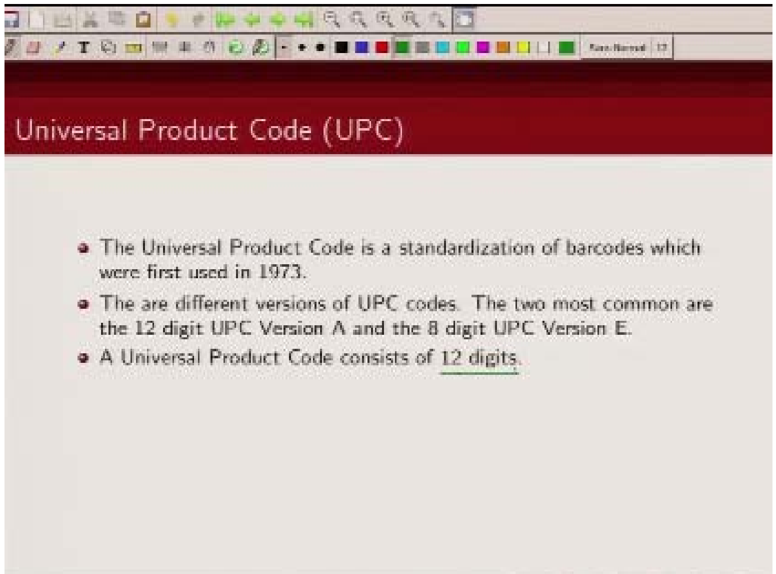
Now next application that we are going to talk about is this Universal Product Code. So this was basically a standardization of all bar codes which was first used in 1973.

(Refer Slide Time: 11:18)



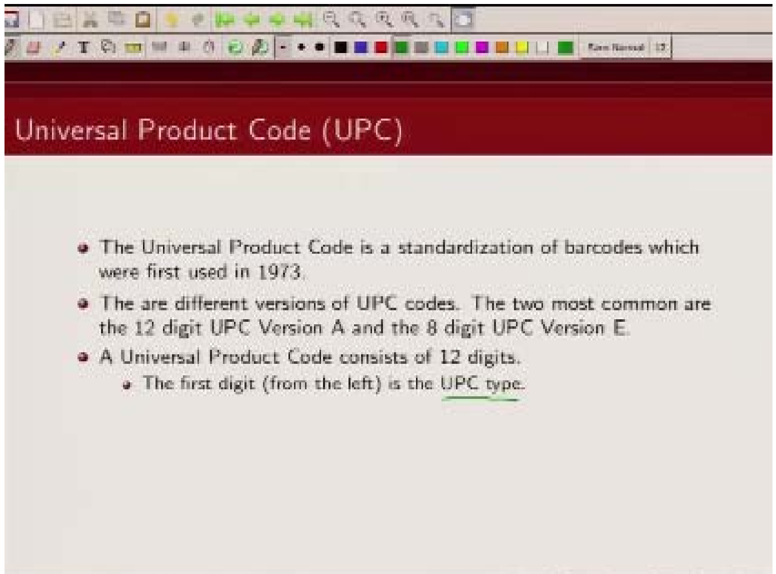
So there are two versions of it, one is this 12 digit UPC version A and the other one is this 8 bit UPC version E.

(Refer Slide Time: 11:28)



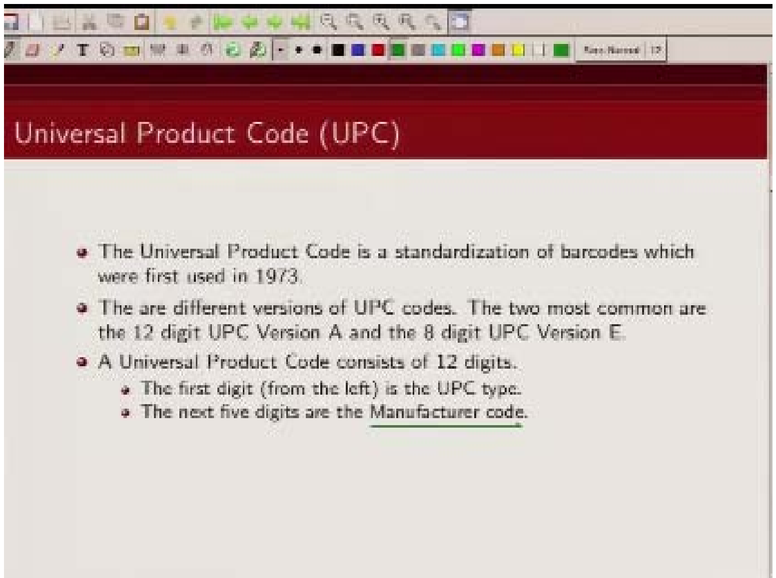
So a 12 digit UPC version has consist of 12 digits.

(Refer Slide Time: 11:35)



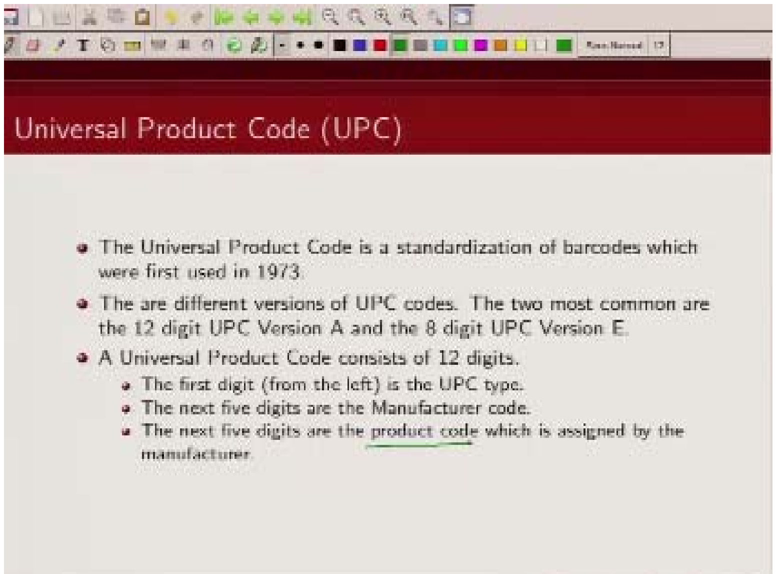
The first digit from the left is of UPC type.

(Refer Slide Time: 11:40)



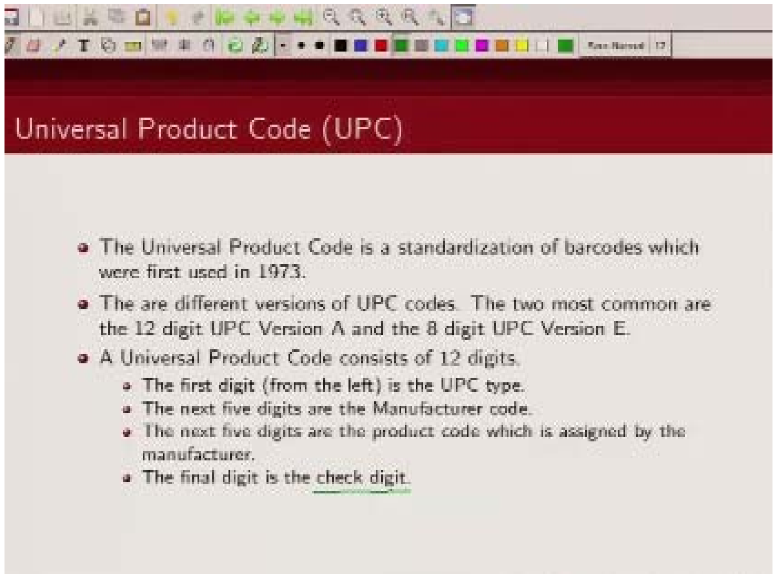
Next five digits are manufactured code.

(Refer Slide Time: 11:43)



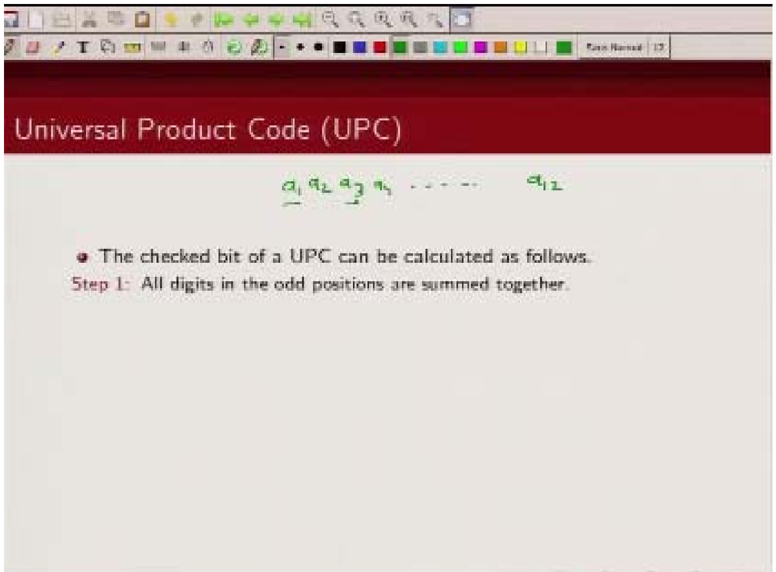
The next five digits are actually the product codes which are assigned by the manufacture.

(Refer Slide Time: 11:49)



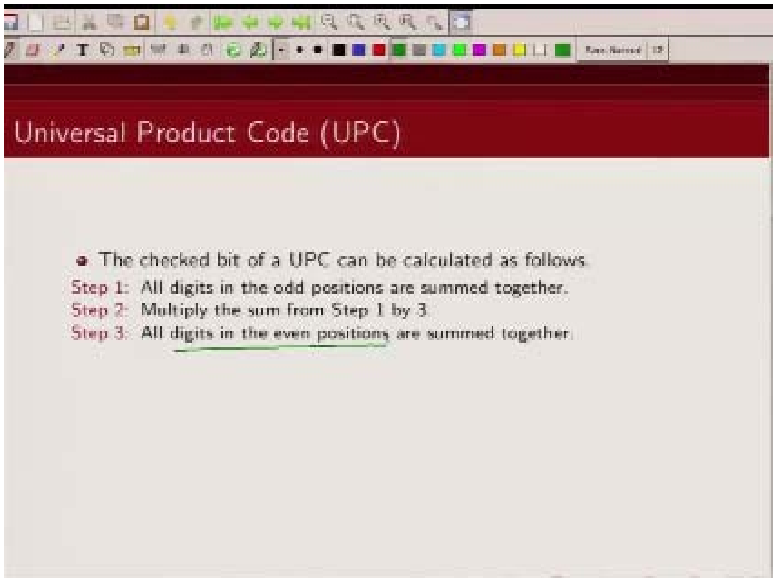
And the final digit is the check bit. So that is what we are going to use for error detection. So how is this check bit computed?

(Refer Slide Time: 12:03)



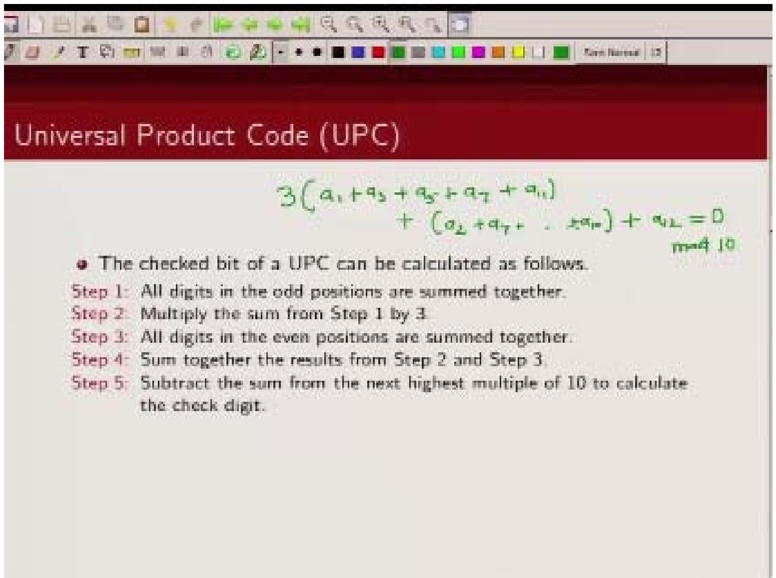
So this check bit is computed as follows. All digits in the odd positions are summed up. So you have this UPC number which is like a_1, a_2, a_3, a_4 up to a_{12}, then what you are going to do is you are going to add up all the odd numbers in the odd location it is a_1, a_3, a_5 like that.

(Refer Slide Time: 12:33)



And multiply the sum by 3, so you are going to multiply the sum of the odd numbers at the odd location by 3, and to that you are going to add all the digits in the even position. So it will be three times $a_1+a_2+a_3+a_5$ and + $a_2+a_4+a_6$ like that. So you are going to add the number, so essentially you are weighing the numbers multiplying by 3, 1, 3, 1, like that.

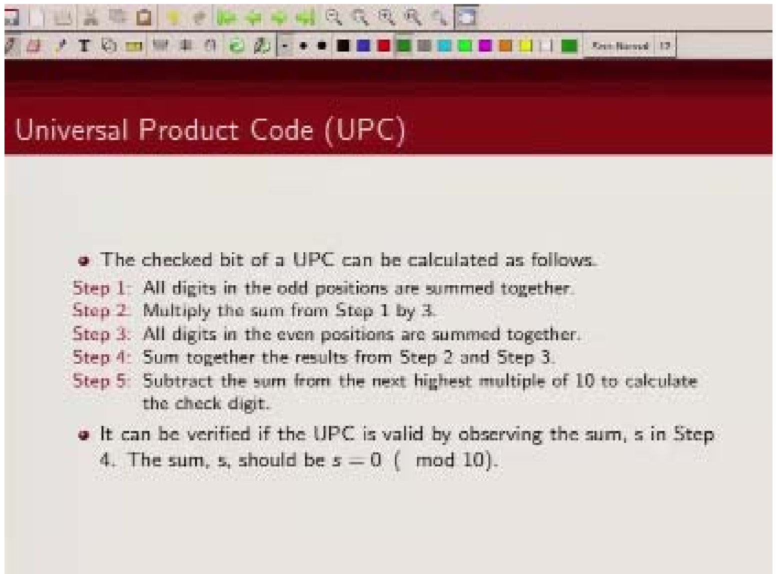
(Refer Slide Time: 13:02)



And you are going to add them all up and subtract the sum that you get from the next highest multiple of 10. So what I am saying is, so you have this $3(a_1+a_3+a_5+a_7+a_{11})+$ you have this $(a_2+a_4+\ldots\ldots)$ you have these when you add them all up plus your 12 bit number, this is your – that should add up to 0 modulo 10 okay. So the odd bit numbers are multiplied by 3, even bit numbers are just added up.

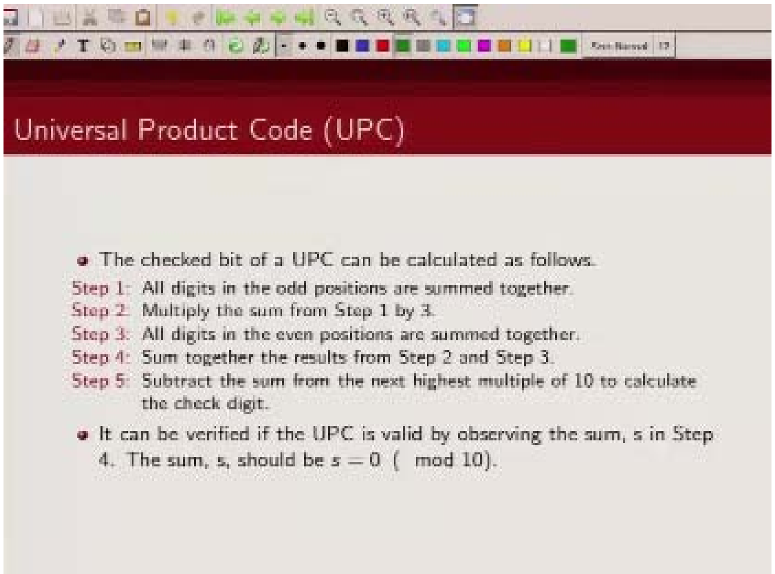
If you add those 11 bits, then the 12[th] bit which is the check bit should be such that sum of this weighted sum should be 0 modulo 10.

(Refer Slide Time: 14:19)

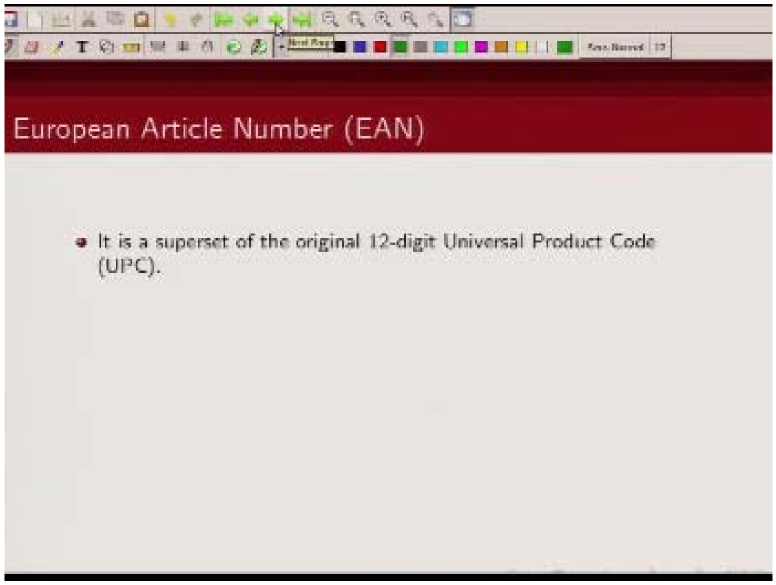


And that is your UPC code. So the sum that you should get should be 0 modulo 10, sum of all the 12 bit numbers. So three times $a_1+a_2+3a_3+a_4$ up to $3a_{11}+a_{12}$ should be zero modulo 10 in case of UPC 12 bit, UPC code.

(Refer Slide Time: 14:44)

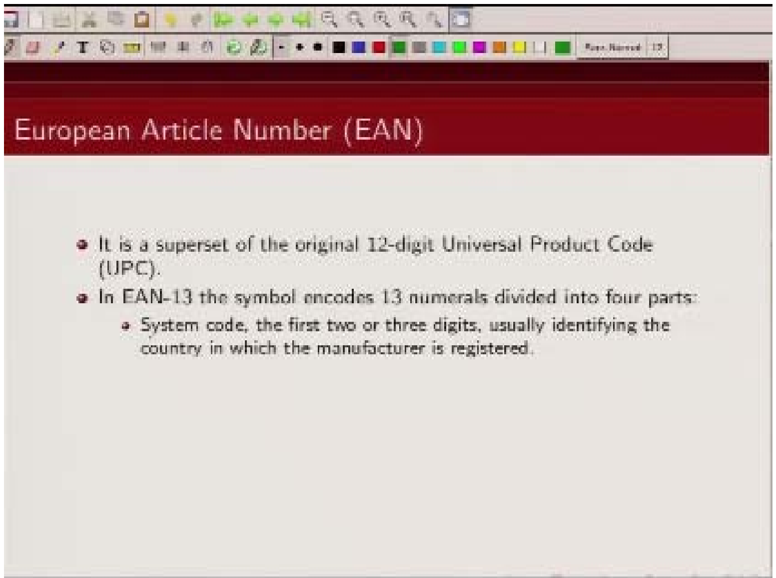


(Refer Slide Time: 14:45)



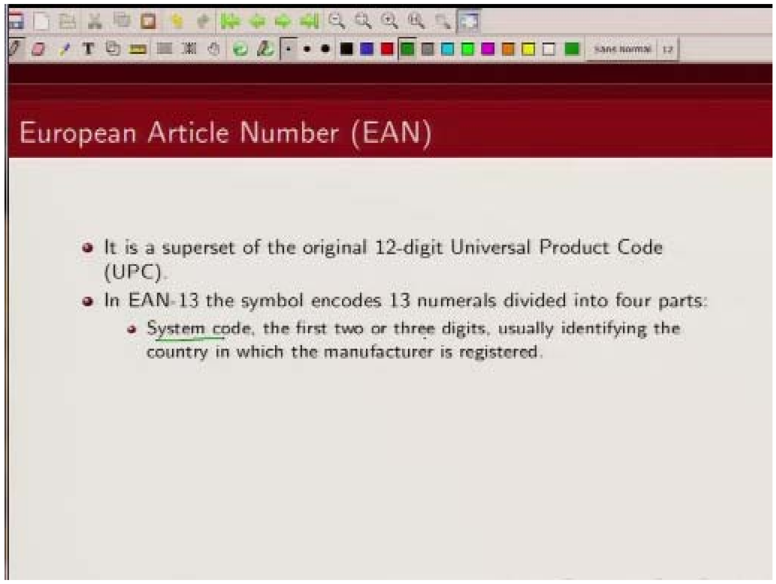
Now a superset of this UPC code 12 bit UPC code is this 13 bit European article number.

(Refer Slide Time: 14:56)



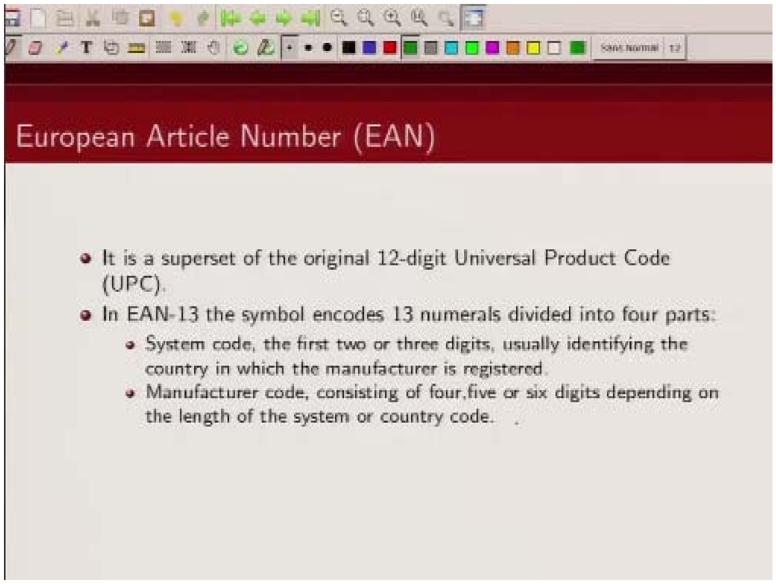
So what is this 13 bit European article number, so it has four parts?

(Refer Slide Time: 15:04)



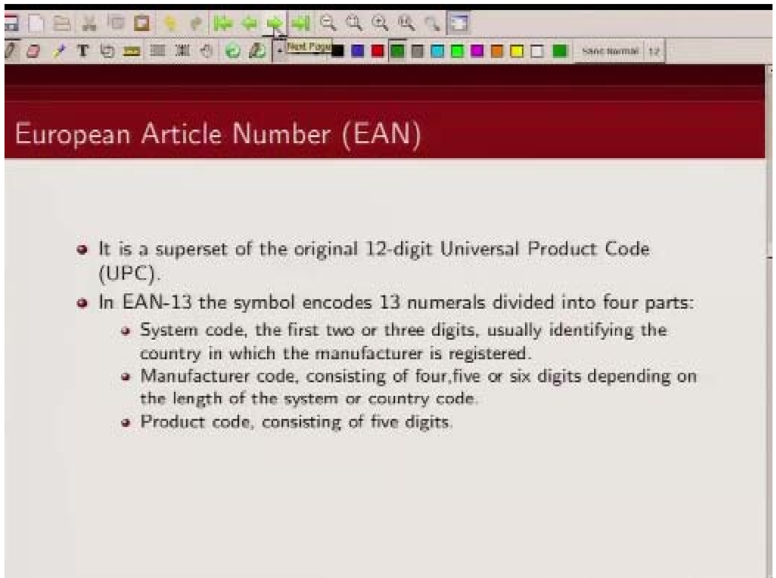
So fist is system code which is first two or three bits which identities a country where the product has been manufactured

(Refer Slide Time: 15:12)



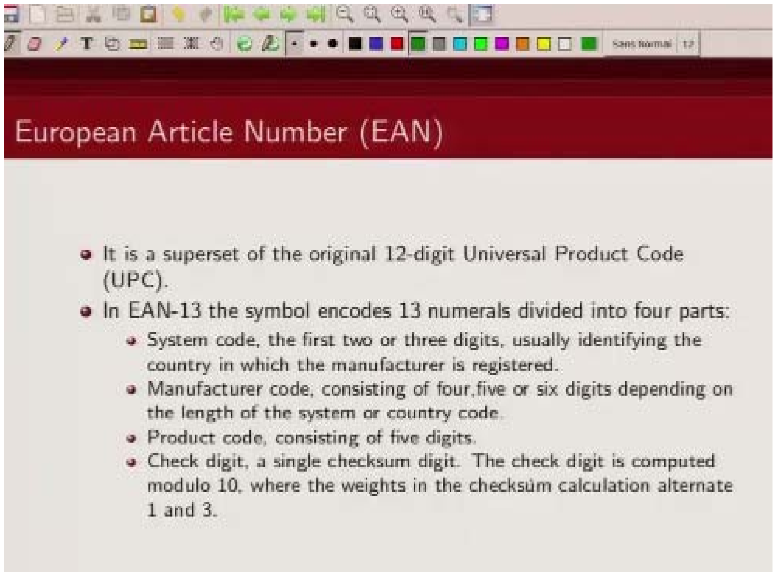
Then you have manufacturer code which is four to six bits

(Refer Slide Time: 15:16)



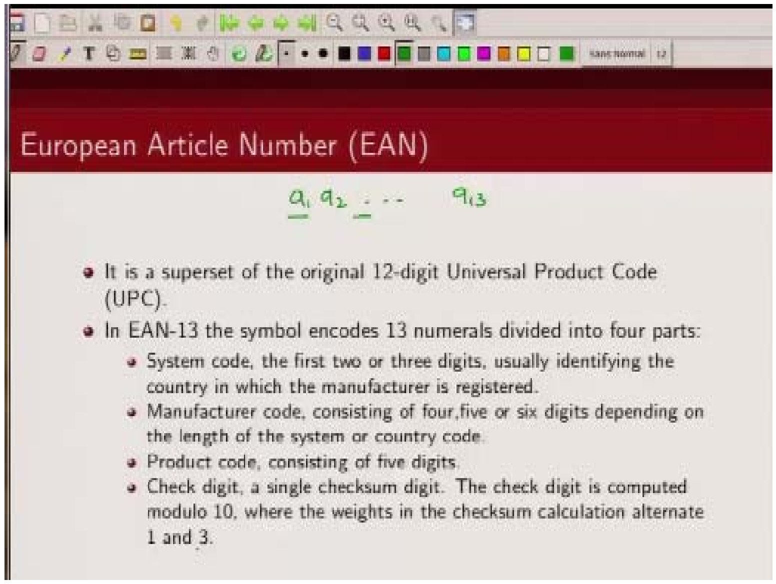
And then you have the product code which consists of five bits

(Refer Slide Time: 15:20)

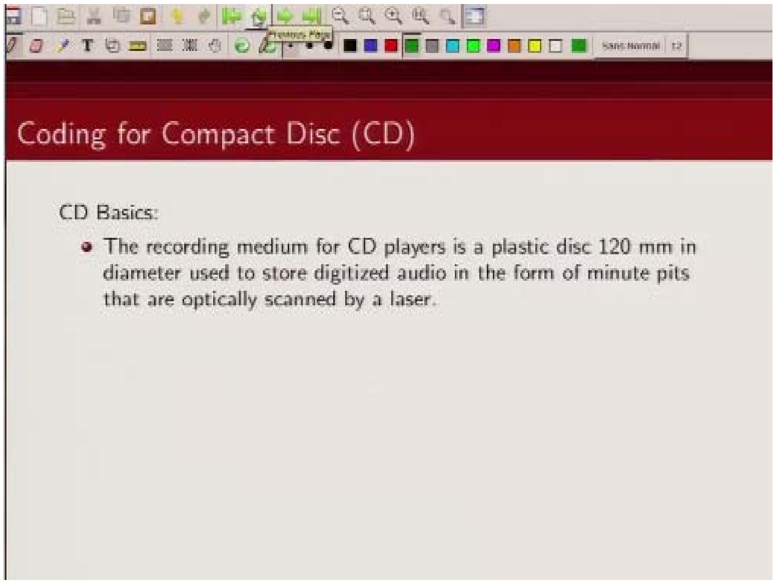


And finally you have one check bit and this check bit is computed modulo ten where the weights in the checksum calculate, calculation alternate between 1 and 3 now what do I mean by tha,t so you have this 13 bit number, $a_1, a_2, a_{13},$ so what you are going to do is these bits in the odd location you multiply by 1 and bits in the even location you multiply by 3, and the modular sum of that should add up to 0 mod 10.

(Refer Slide Time: 15:38)

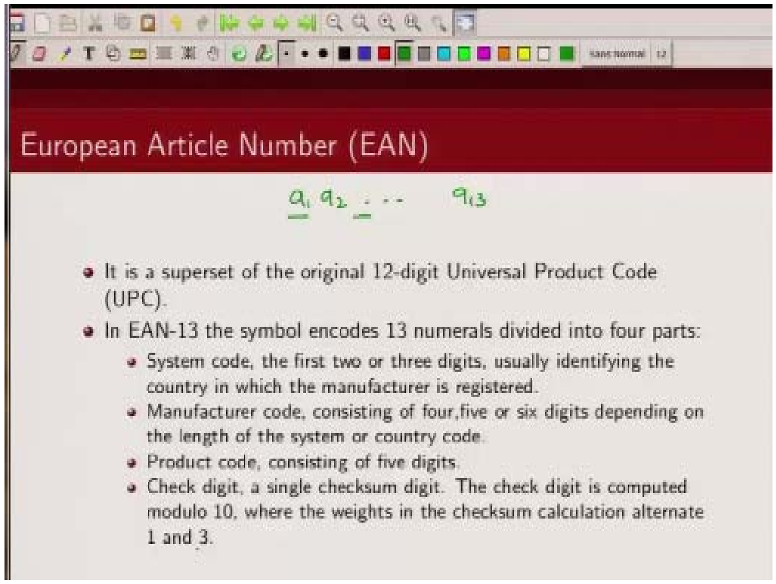


(Refer Slide Time: 16:03)



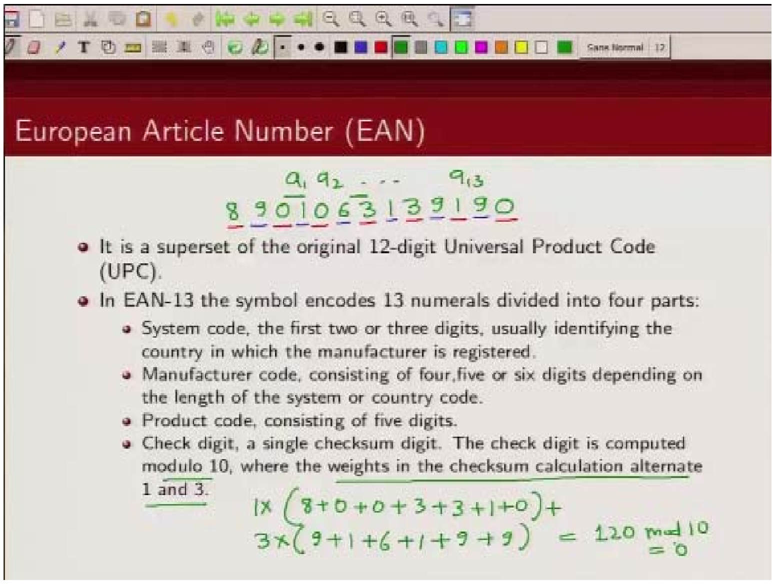
So let us look at an example to

(Refer Slide Time: 16:07)



Illustrate that, okay let us look at an example to illustrate that so I have this Britannia bonbon biscuit right and what is this European article number for this,

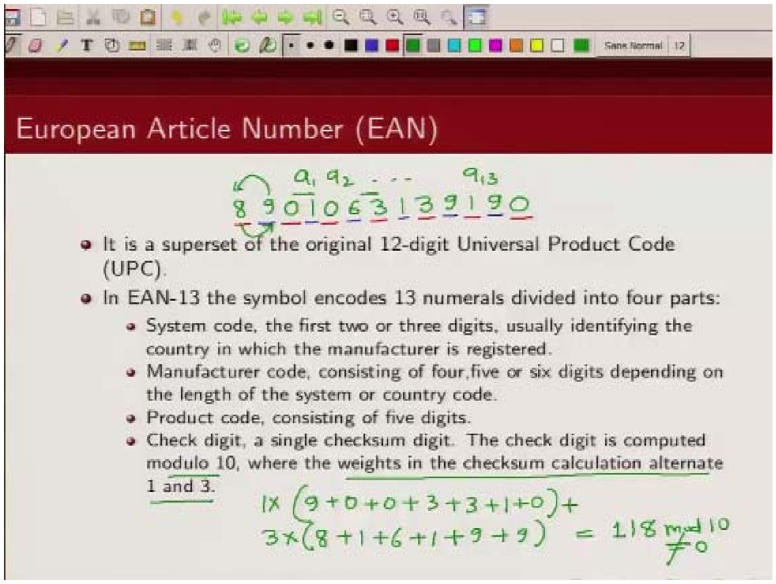
(Refer Slide Time: 16:23)



So European article number for this is 8 9 0 1 0 6 3 1 3 9 1 9 0, so this is my 13 bit, you can see 1,2,3,4,5,6,7,8,10,11,12,13 so these 13 bit European article number. Now let us see whether this checksum bit is correct so what do we need to do, we need to add odd bits and we need to add the even bits, the bits in the even location and multiply them by three, so let us first mark the bits in the odd location, this is 0, 3, 3, 1, 0, okay and what are the bits in the even location, that is marked with this blue pen okay. So let us add up so again the check bit is computed modular ten where weights in the checksum calculation alternate between 1 and 3, so what do I mean by that so every odd bit I multiply by 1 and every even bit I, bits at the even location I multiply by three.

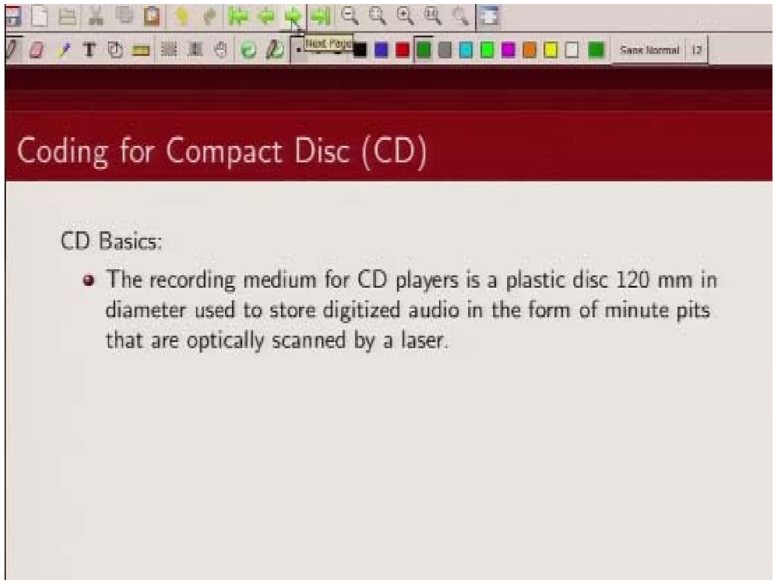
So let me first add up the numbers in the odd location, so that is 8+0+0+3+3+1+0, this will multiplied by 1 + 3 times numbers in the even location, that is 9+1+6+1+9+9. 9, 1, 6, 1, 9, 9, and this comes out to be 120 and 120 mod 10 is 0, so you can see that, that checksum bit is correct. So you can see from the examples that we have done so far the last bit in all these are our check bit which can be used for error deduction.

(Refer Slide Time: 18:17)



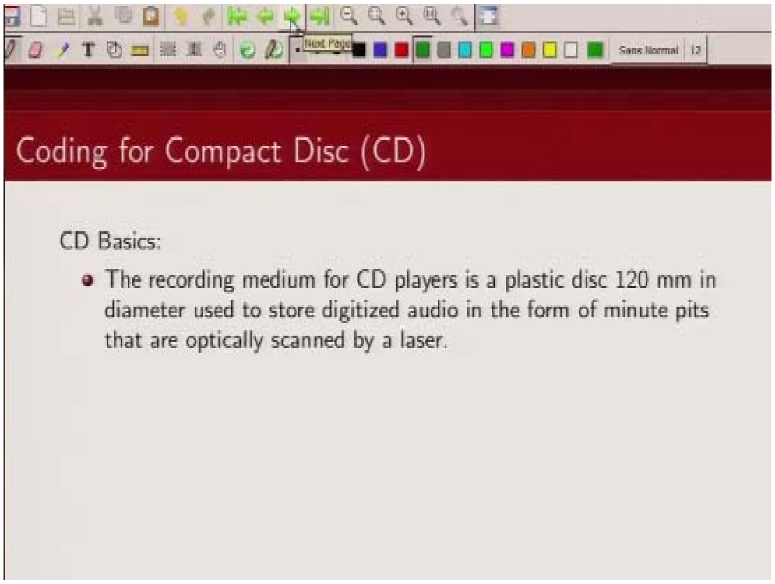
Now this code also can easily find out the transposition error, for example if these two bits would have got exchanged then what would have happened, then here this would have been these things would have changed this would have become 9 and this would have become an 8, so this would have been, this would have changed the sum would have changed so earlier it was 9 times 3, now it is 24 three less and there is one more, so now two less, so this would have become 118 and this is not 0 modulo 10. So you can see this code can correct, can, can detect transposition error and these are the most common error which happen when you are trying to read these bar codes okay. Now let us now move to some other.

(Refer Slide Time: 20:10)



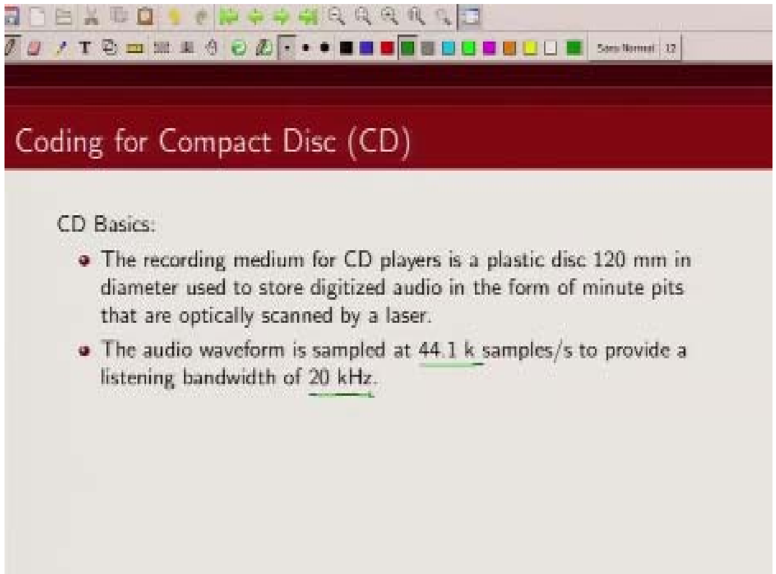
Applications, so we start with application of error control coding in CD's, you are familiar with CD disc, it is basically.

(Refer Slide Time: 20:14)



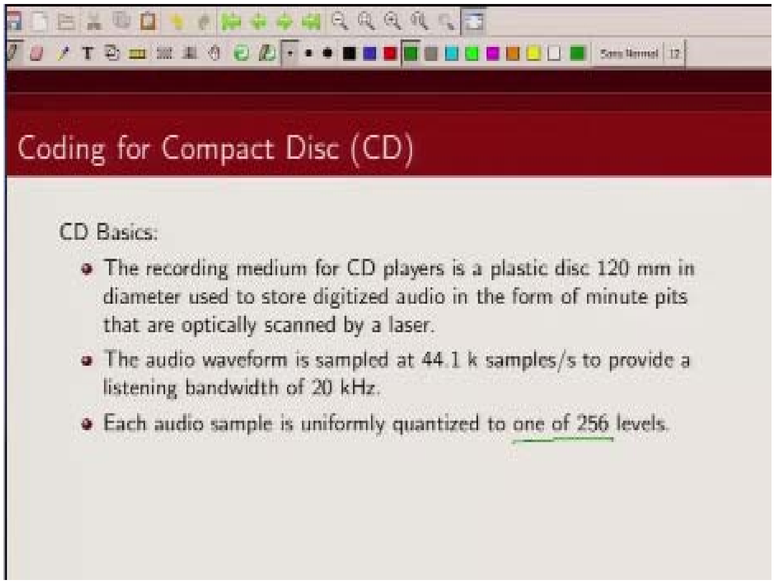
A plastic disc of 120 mm in diameter and it can be used for storing digital audio, video data

(Refer Slide Time: 20:28)



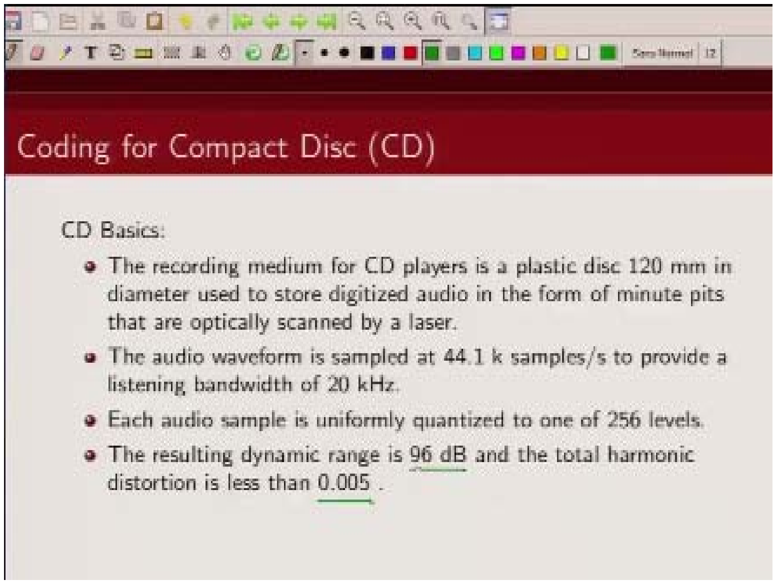
Now the audio signal is sampled at 41.1 k samples/s and it provides a listening bandwidth of 20 kHz.

(Refer Slide Time: 20:40)



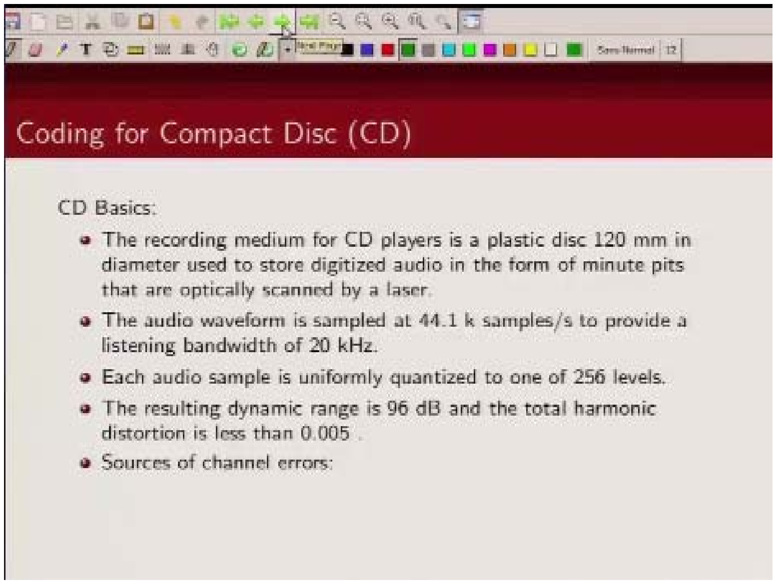
Each audio sample is sample to quantized to one of the 256 levels

(Refer Slide Time: 20:50)



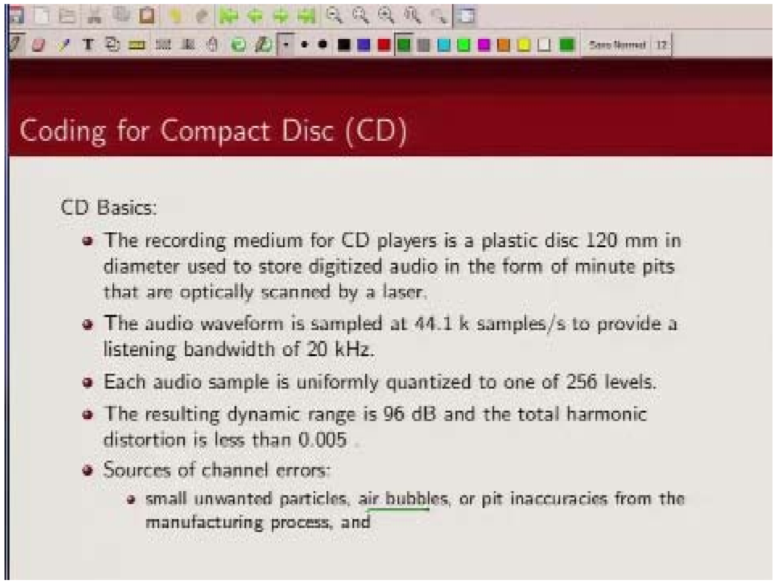
Which provides a dynamic range of 96dB and harmonic distortion of less than .005

(Refer Slide Time: 20:59)



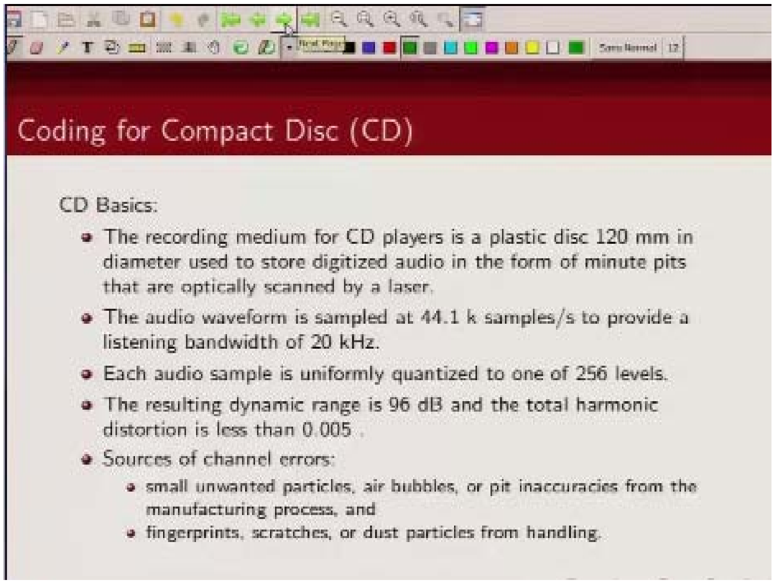
Now what are the sources of error in a CD disc?

(Refer Slide Time: 21:04)



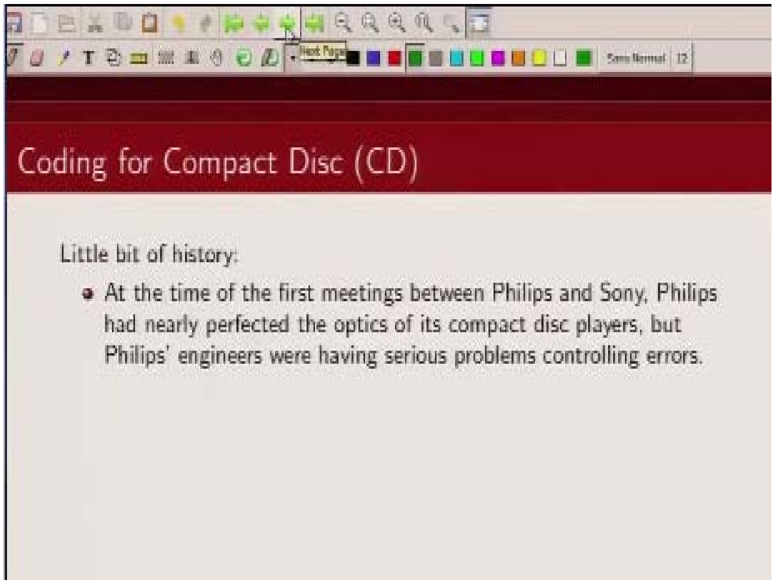
So there could be small unwanted particle, dust particles or there could be some air bubbles or things like that which could have happened during manufacturing of these CD's

(Refer Slide Time: 21:16)



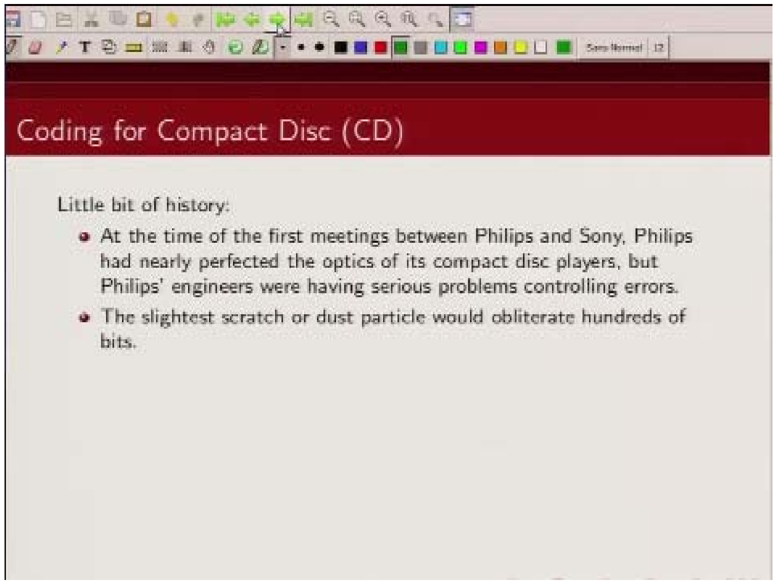
Or fingerprints, scratches, dust particles from handling the CD, so how does a CD corrects or these kinds of errors? You must have noticed even if you have, your CD has small scratches but still your CD plays, so how does it happen?

(Refer Slide Time: 21:35)



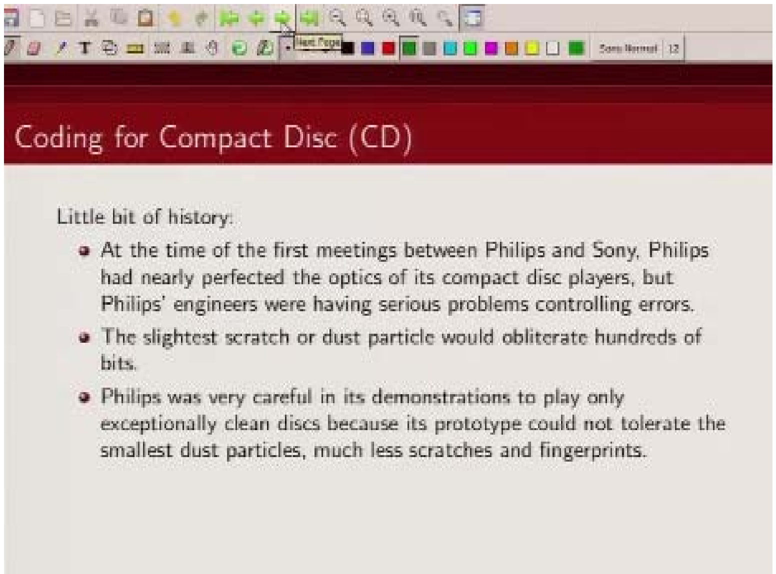
This happens because of a very powerful error control coding which is used in the CD's, so little bit of history basically Philips and Sony were experimenting with this optical disc and Philips engineers had a lot of problems controlling errors.

(Refer Slide Time: 21:53)



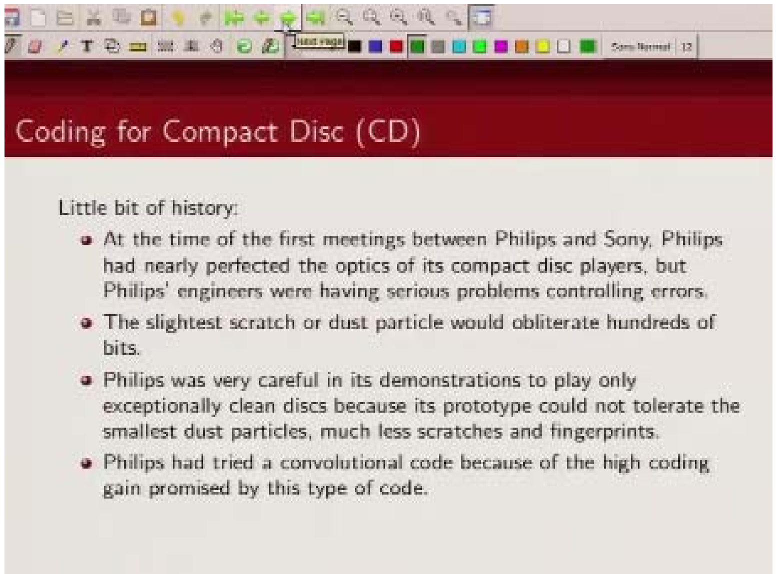
In the disc and even if there would have been small scratches or dust particles the whole data would essentially you know go away

(Refer Slide Time: 21:53)



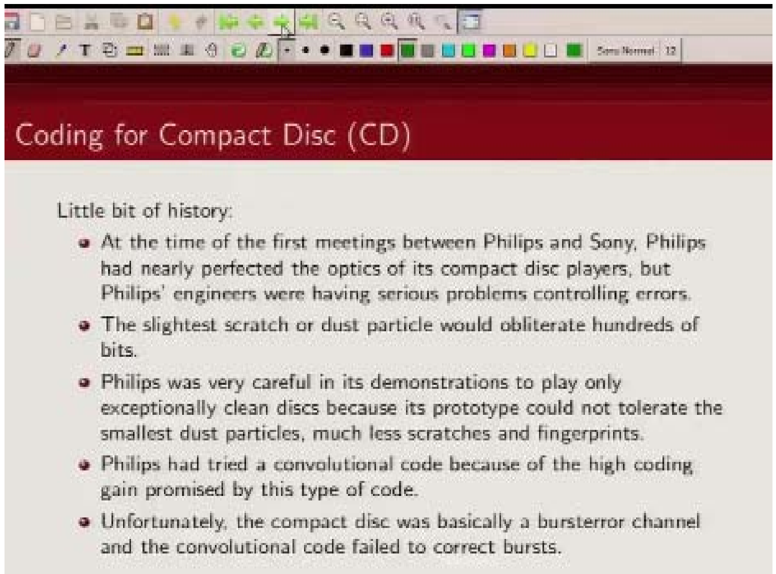
So whenever Philips were demonstrating the CD they were very careful, they were just demonstrating with very, very clean CD's and which did have any scratches and things like that and what Philips.

(Refer Slide Time: 22:14)



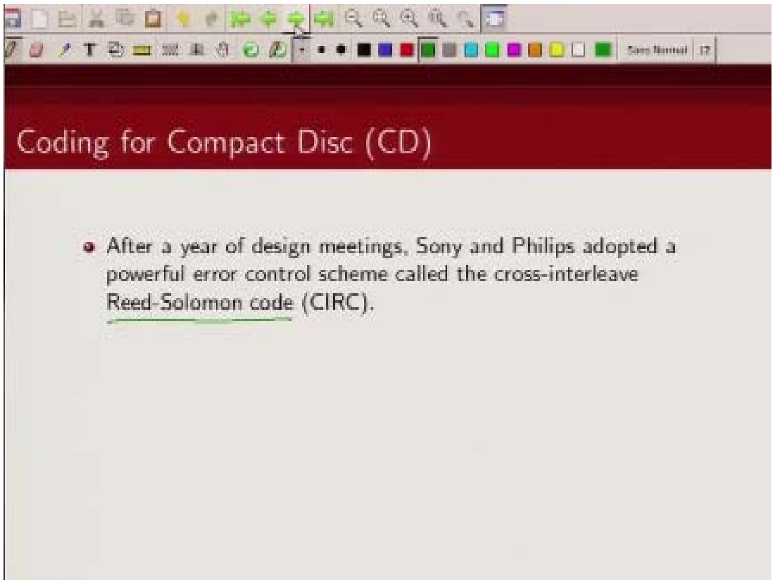
Was actually doing was they were trying a convolutional code to correct errors because of high coding gain provided by that but the problem is this storage channel is like an erasure channels, if there is scratch or something like that there are neighboring bits which get affected and the data there is erased, so convolutional codes are not good in correcting erasures and that is why they were initially not successful.

(Refer Slide Time: 22:44)



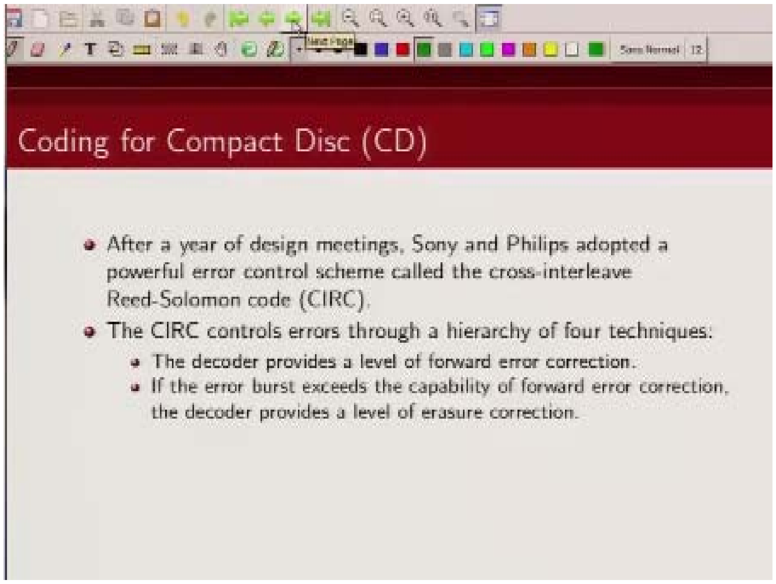
So as I said the problem was this storage media was a burst error correct burst channel, and convolutional codes were not good in correcting burst error, so there comes

(Refer Slide Time: 22:59)



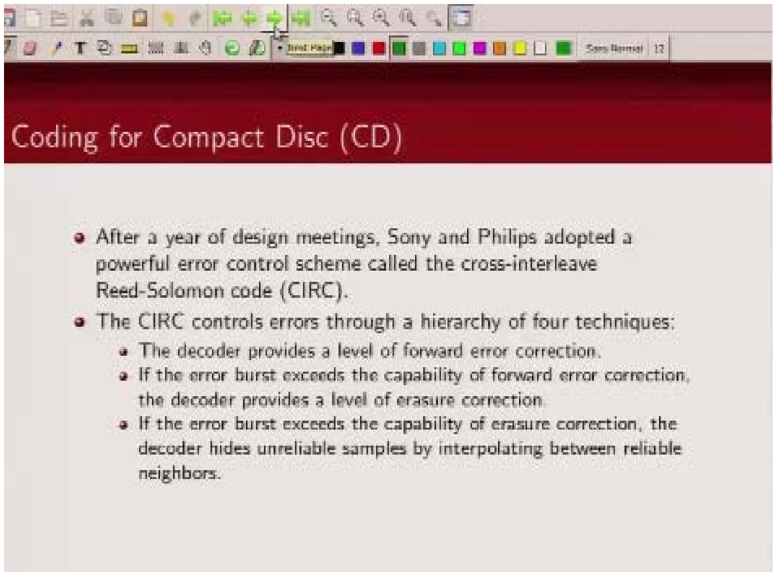
Reed-Solomon code and so the CD's used are basically across interleave Reed-Solomon codes

(Refer Slide Time: 23:10)



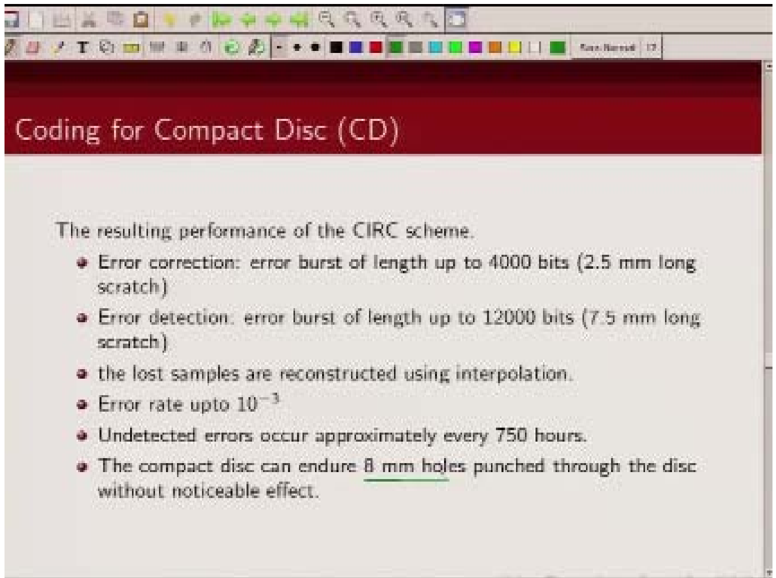
And there is you know four set of kind of hierarchy as far as error control and deduction is concerned, so there is an inherent forward error correction because of these Reed-Solomon code, but this the error burst exceeds the capability of the error correcting capability of the code, then the decoder also provides a level of erasure correction so there is a, in addition to error correction there is also an erasure correction and erasure is you know bunch of neighboring bits just getting erased and not just not being able to recover, so the decoder has this capability, if the error corrections fails this, at second level the decoder can do erasure correction and try to recover

(Refer Slide Time: 23:57)



Those data, third is if the error burst exceeds the capability of erasure correction then what it does is it tries to do some sort of an interpolation between reliable neighbors so as to interpolate what the data is, and if even interpolation fails then basically the decoder just blanks out and you will hear, like kind of a you know mute sound when the error is big enough that even the interpolation cannot work.

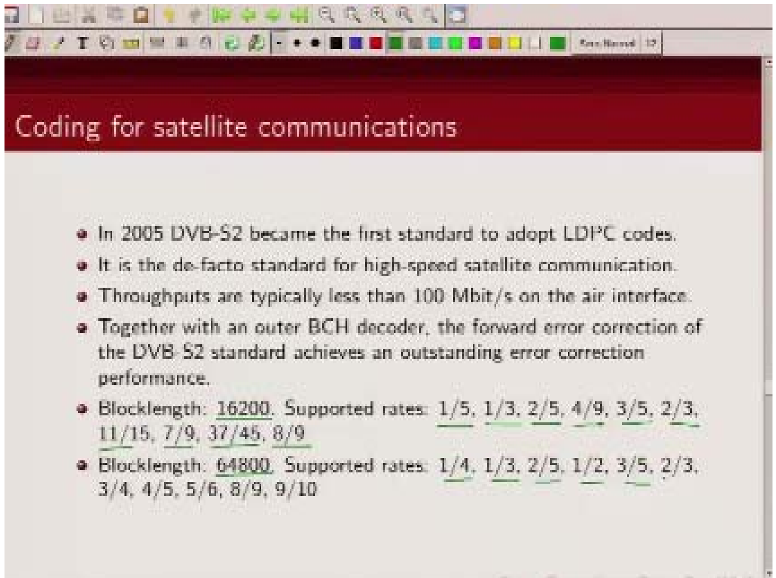
(Refer Slide Time: 24:29)



So a CD disc can correct error burst up to 4000 bits, so try putting a scratch of 2.5mm, your CD will still work, I can guarantee that, it can detect error up to 12000 bits so it can detect errors of up to 27.5mm long scratch, and of course as I said if the error burst exceeds the capability of erasure correction then it tries to do interpolation to get back those steps, and unrated error happens very rarely, it happens every like 750 hours so. So there is a very powerful error correcting codes which are used in, in CD.

And interestingly even these compact disc scan withstands small punch holes in it, you can try putting a 8mm hole into your CD and still it would be able to you know correct the errors in the data and the CD will still work.

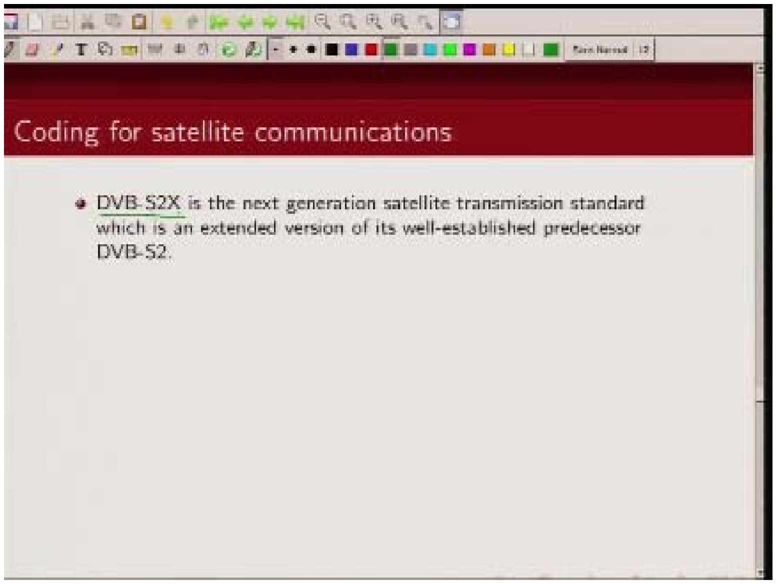
(Refer Slide Time: 25:41)



Now let us move to application for in satellite communication, so there is a standard D, digital video broadcasting is too standard which was one of the first standards to adopt low density parity check codes as their standard, so it is a de-facto standard for high speed satellite communication where typical through put is less than 100 Mbits/s, now it uses as an outer code BCH code along with the inner LDPC codes and it can support variety of block sizes so it arise, it can support a block size of 16,200.

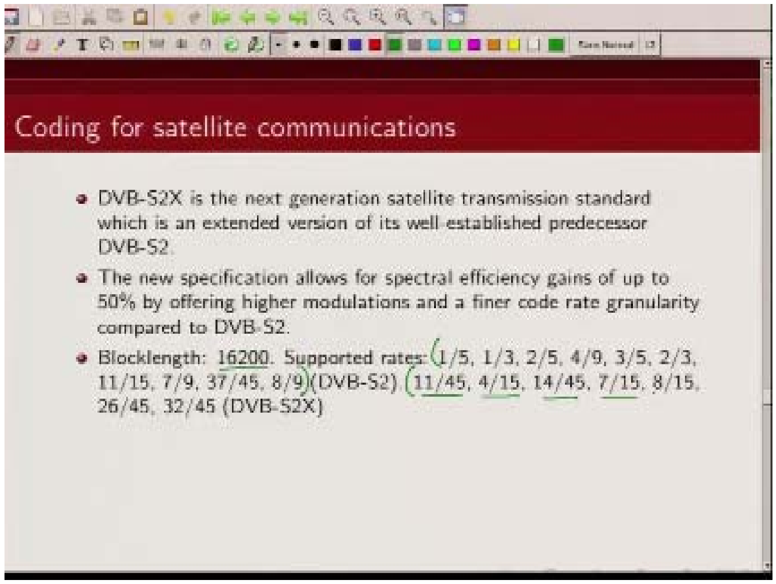
And these are the various code rates that have been supported, so LDCP codes of rate so these are the various rates which have been supported by this standard, similarly for block size of 64,800 these are the various code rates that has been supported in this standard.

(Refer Slide Time: 26:48)



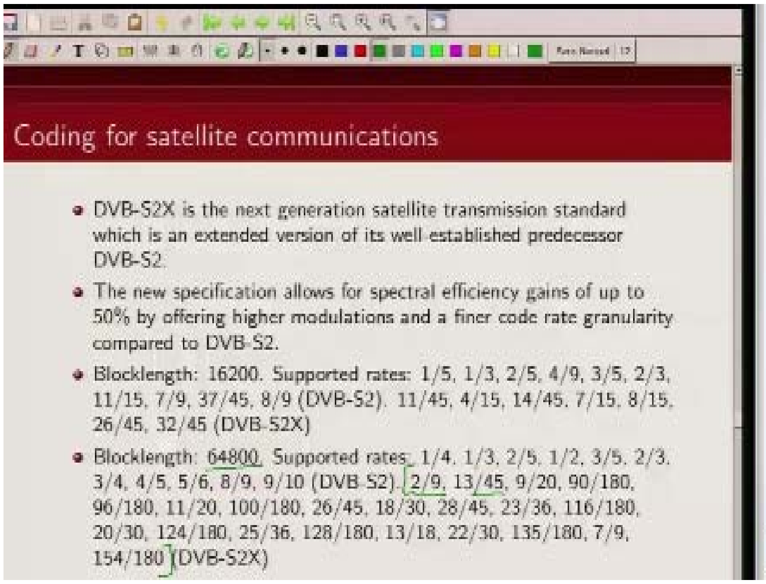
Now there is another standard which is abradation of this DVB is 2 standards quality VB is 2x which is an extended version of DVB-S2 standard.

(Refer Slide Time: 27:00)



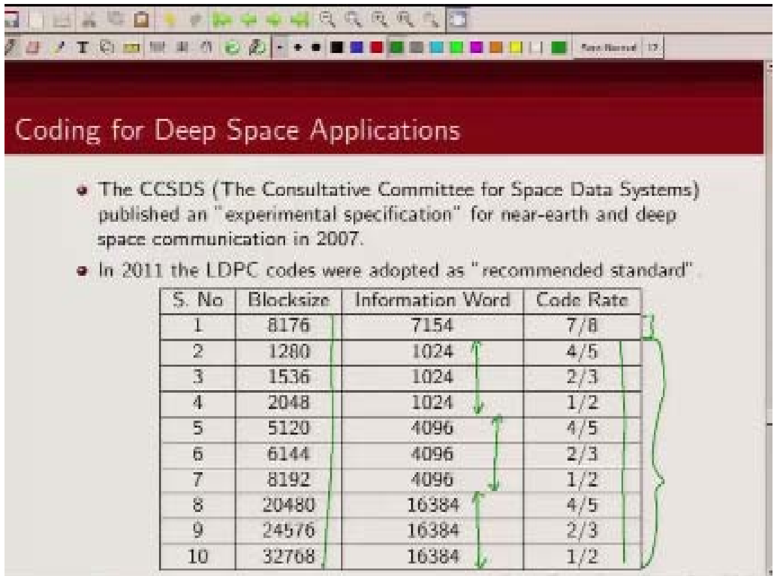
And it allows more final rates of LDCP codes, so you can see for the block length of 16,200 these were the rates which were supported by DVB- S2, now the DVB-S2x can support further these rates, so it provides a much finer code rate granularity compared to DVB-S2 standard.

(Refer Slide Time: 27:29)



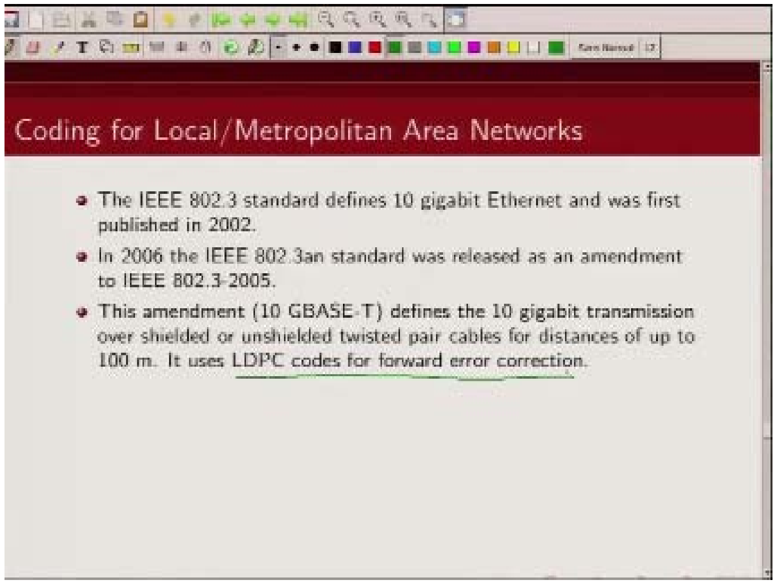
Now similarly for block size of 64,800 you can see these rates that you see here from here starting from here, these rates were not supported in DVB-S2 which are supported in DVB-S2x.

(Refer Slide Time: 27:44)



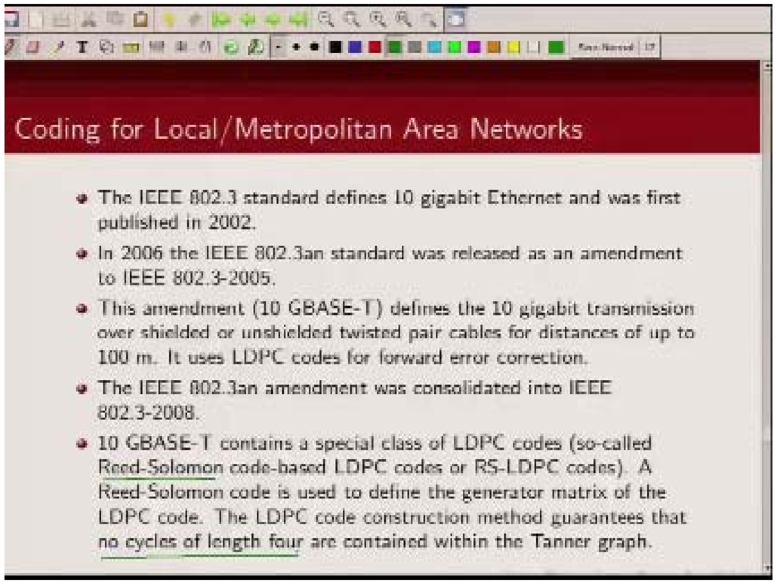
What about these space exploration, now NASA again has specified as recommended standard so these codes that you see here, these are space up, have been prescribed for D space communication and this is for near earth communication so it is for D space communication it up, it supports 3 different block length, information block length 1024, 4096 and 16,384, so these 3 different information block length can be supported and these are the various rates which have been supported, so correspondingly these are the block sizes that you will get.

(Refer Slide Time: 28:28)



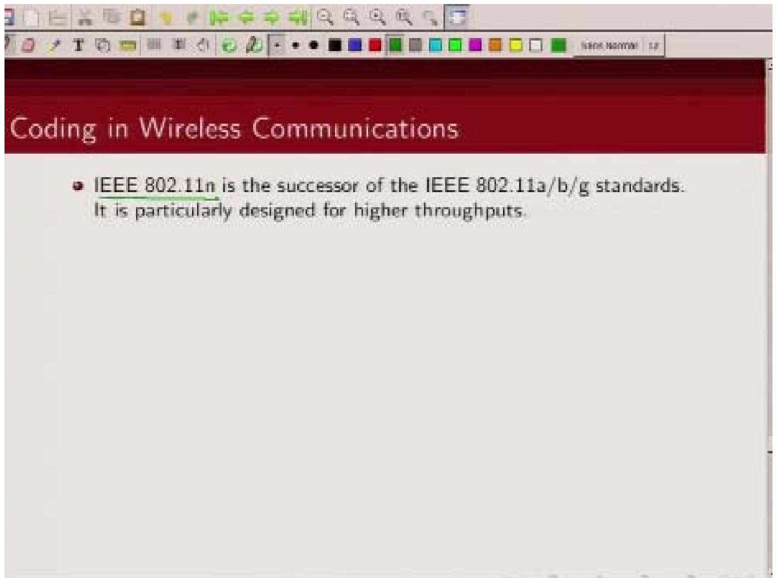
Now for 10 gigabits Ethernet also LDPC codes have been suggested, so LDPC codes have been use for forward error correction here, for 10 gigabits Ethernet transmission over Ethernet.

(Refer Slide Time: 28:45)



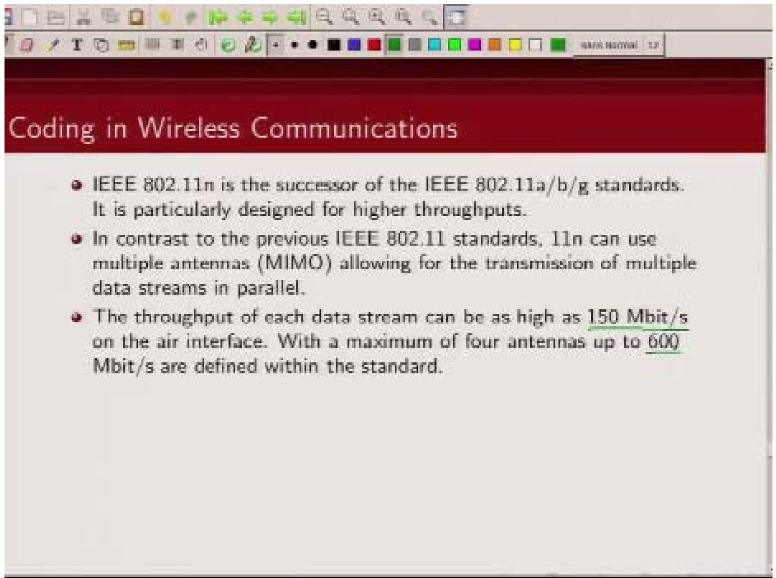
And it uses Reed-Solomon code based LDPC code so the Reed-Solomon code is used to define the generator matrix of LDPC code, and this construction and it shows that there is no cycle, 4 cycles in the LDPC code.

(Refer Slide Time: 29:07)



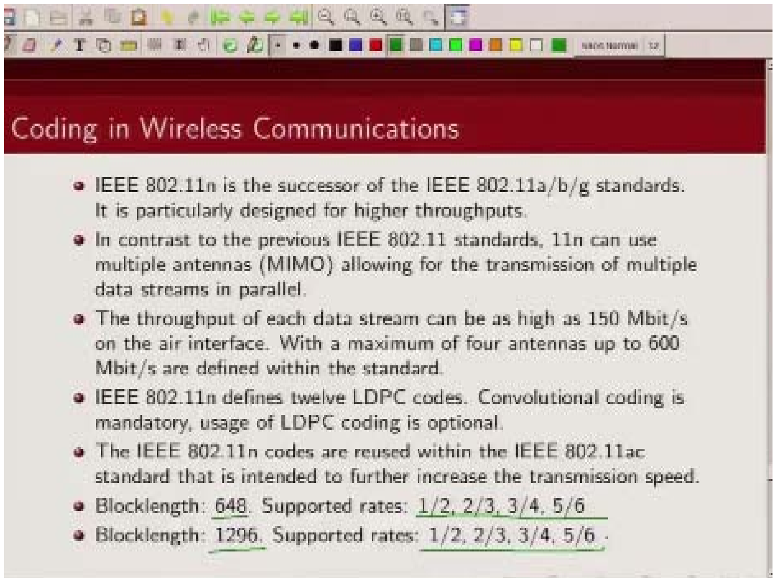
Now let us move to applications in Wifi, so IEEE 802.11n which is a successor of IEEE 802.11a/b and g. and it supports.

(Refer Slide Time: 29:22)



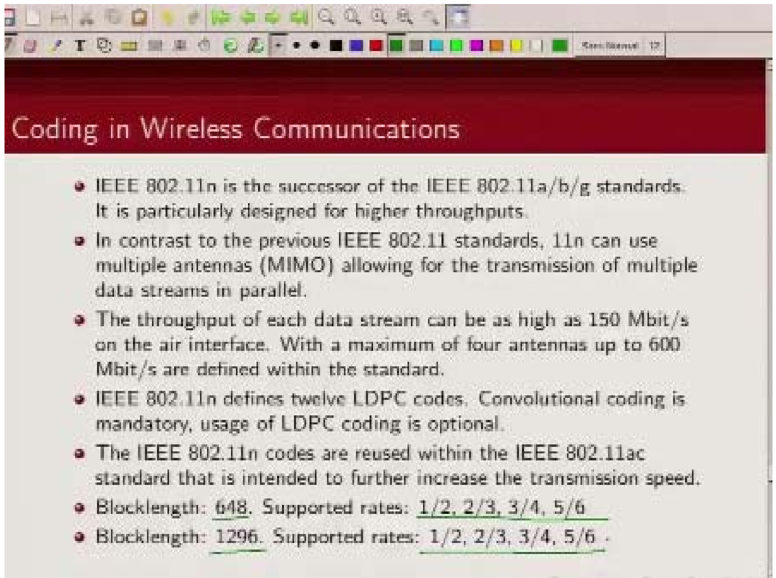
Multiple antennas for transmission, so you could get about 150Mbps in each data stream, if you use four antennas can get up to 600 Mbps

(Refer Slide Time: 28:35)



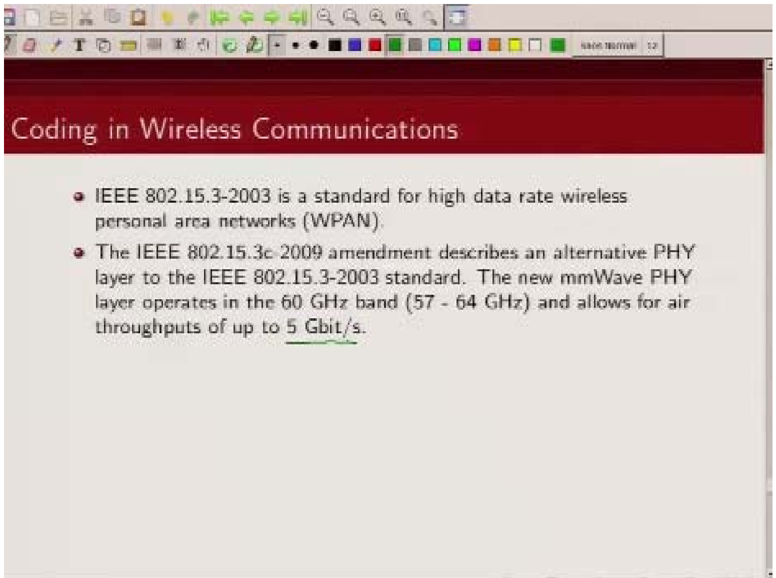
Now here twelve different LDPC codes have been suggested, now use of LDPC codes in, in this standard is optional, the mandatory requirement is convolutional code. But they have specified LDPC codes which, which are optional which can be used and then these codes have been further reused in this standard 802.11ac to further increase the transmission rate.

(Refer Slide Time: 30:25)



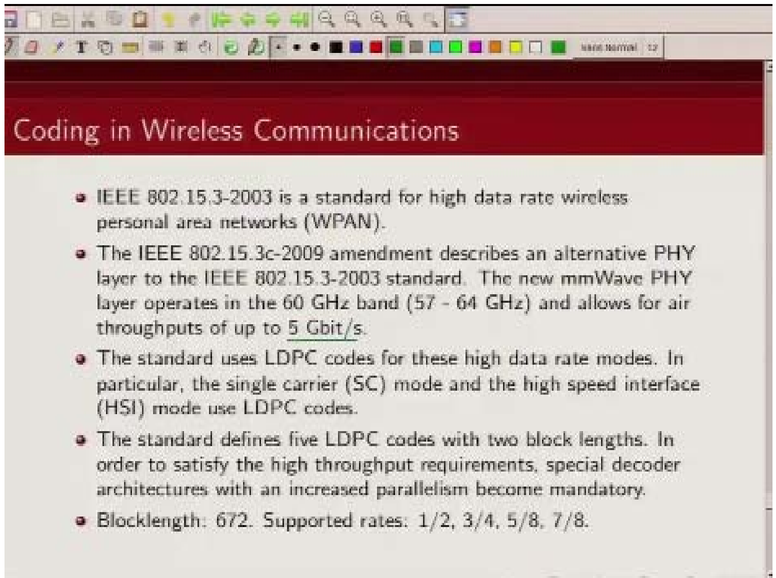
So I list here some of the block lengths which have been supported. So block lengths 648 and these are the various code rates supported here. Similarly, for a block length of 1296 these are the various code rates which have been supported.

(Refer Slide Time: 30:18)



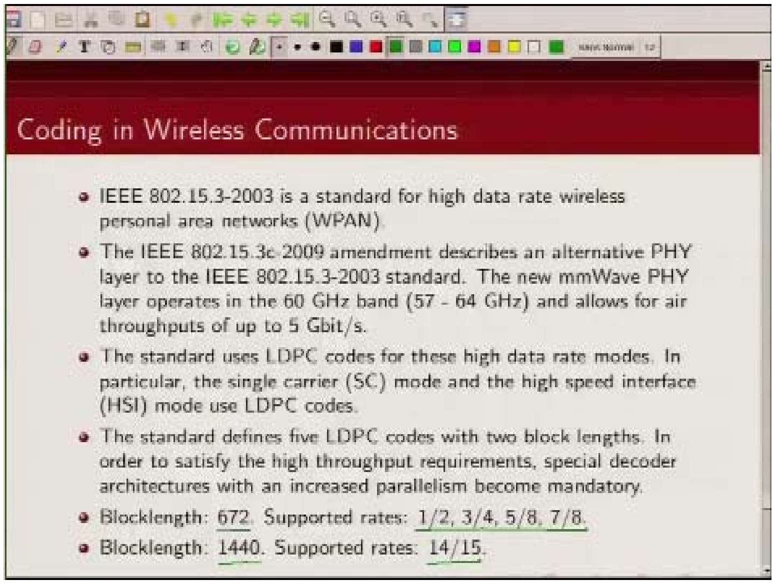
Then we have IEEE 802.15 which is a standard for wireless personal area network. Here also there is a new PHY layer which is based on millimeter wave in 60GHz band which allows throughput up to 5 Gbit/s.

(Refer Slide Time: 30:39)



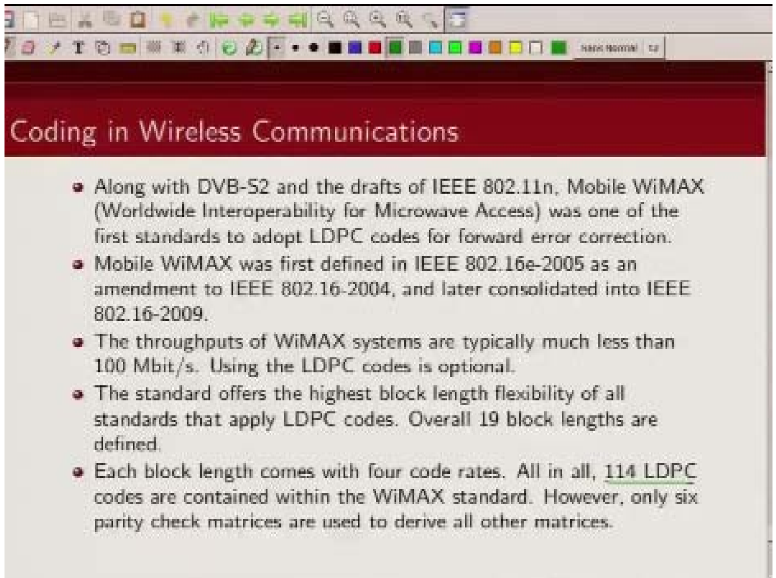
Now here LDPC codes have been suggested, now five different LDPC codes for two different block length have been suggested. Now typically the block size here is smaller so as to ensure and also there is architecture research that because we have to get five throughput up to 5 Gbit/s, so architecture research that there is lot of parallelism and the decoder.

(Refer Slide Time: 31:04)



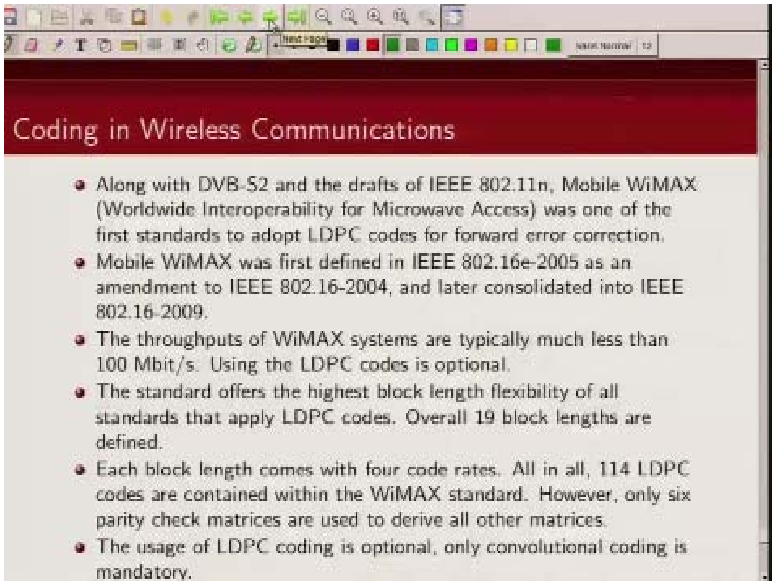
Can be implemented in parallel, so the rates which have supported, so block size 672 and these are the rate code rates which are supported here. Similarly for block length of 1440 a code rate of 14/15 has been supported.

(Refer Slide Time: 31:20)



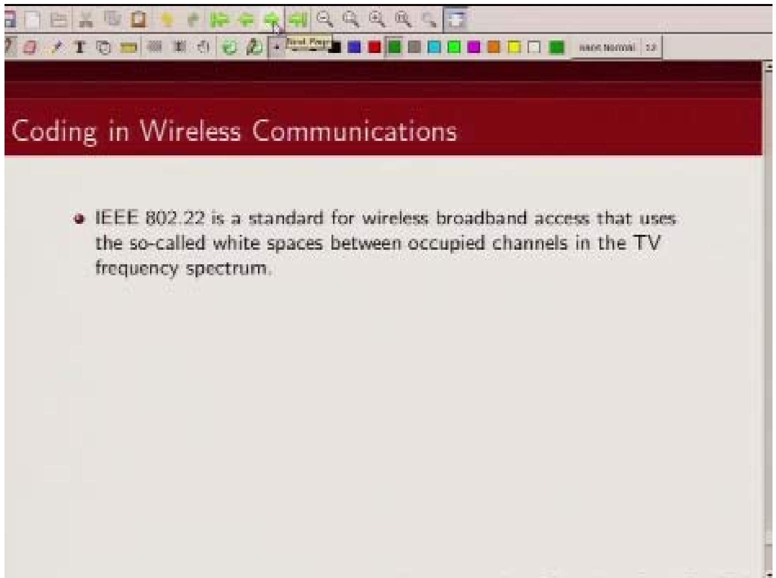
Now another standard where LDPC codes have been used has been from very beginning is this WiMax and there are lots of codes which have been defined in this WiMax standard, again use of LDPC code is optional here, but they have suggested lot of flexibility as far as block length is concerned and 19 different block lengths have been considered and various different rates have been considered. Total of 114 LDPC codes have been you know defined or contained in this WiMax standard.

(Refer Slide Time: 31:58)



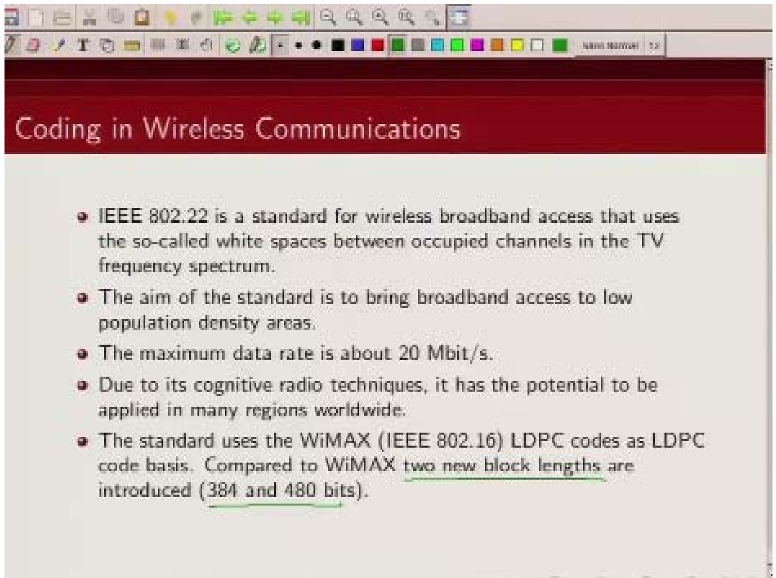
And as I said, here also the use of LDPC codes is optional; the mandatory requirement is convolutional code.

(Refer Slide Time: 32:08)



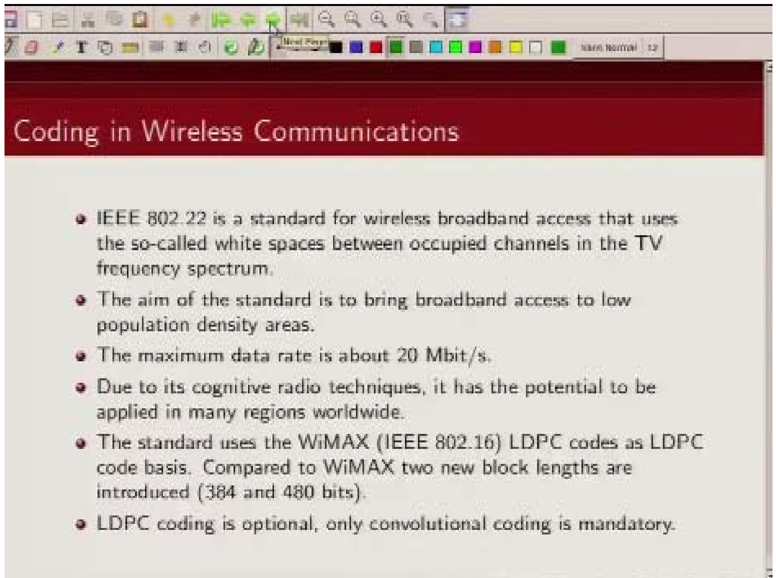
And finally I will conclude this discussion with IEEE 802.22 standard which is basically for cognitive radio opportunistic access over TV white spaces.

(Refer Slide Time: 32:44)



The TV spectrum which has not been used for coming which is not, which can be opportunistically used for communication and the whole idea of IEEE 802.22 2 is to provide rural connectivity. So here also they have specified lots of various LDPC codes and in fact they have two different, two new block lengths which were not there in WiMax have been added, one corresponding to 384 bits and another corresponding to 480 bits. So this is another application where

(Refer Slide Time: 32:56)



LDPC code is optional; convolutional code is mandatory but LDPC code is optional has been suggested. So this is not an exhaustive list of application, I just wanted to give you some flavor of where in practice we use linear block codes. Thank you.

Acknowledgement
Ministry of Human Resource & Development

Prof. Satyaki Roy
Co-ordinator, NPTEL IIT Kanpur

NPTEL Team
Sanjay Pal
Ashish Singh
Badal Pradhan
Tapobrata Das
Ram Chandra
Dilip Tripathi
Manoj Shrivastava
Padam Shukla
Sanjay Mishra
Shubham Rawat
Shikha Gupta
K. K. Mishra
Aradhana Singh

Sweta
Ashutosh Gairola
Dilip Katiyar
Sharwan
Hari Ram
Bhadra Rao
Puneet Kumar Bajpai
Lalty Dutta
Ajay Kanaujia
Shivendra Kumar Tiwari
an IIT Kanpur Production