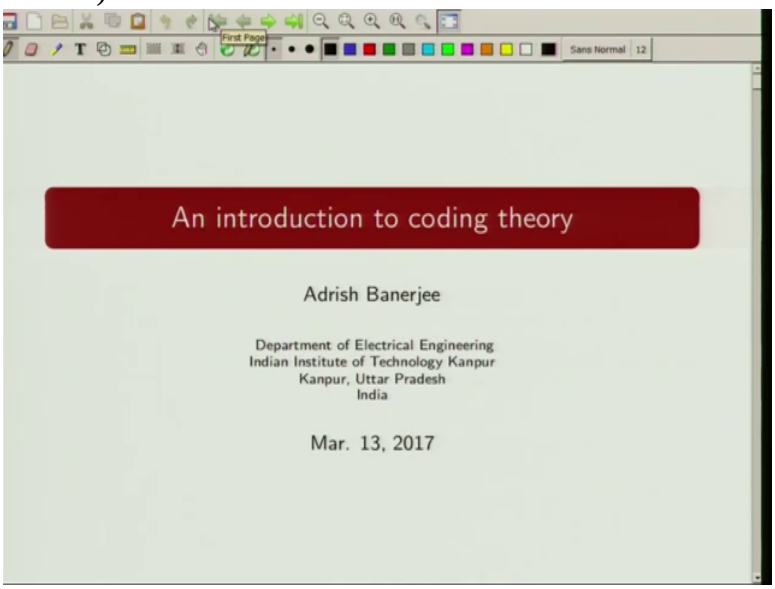
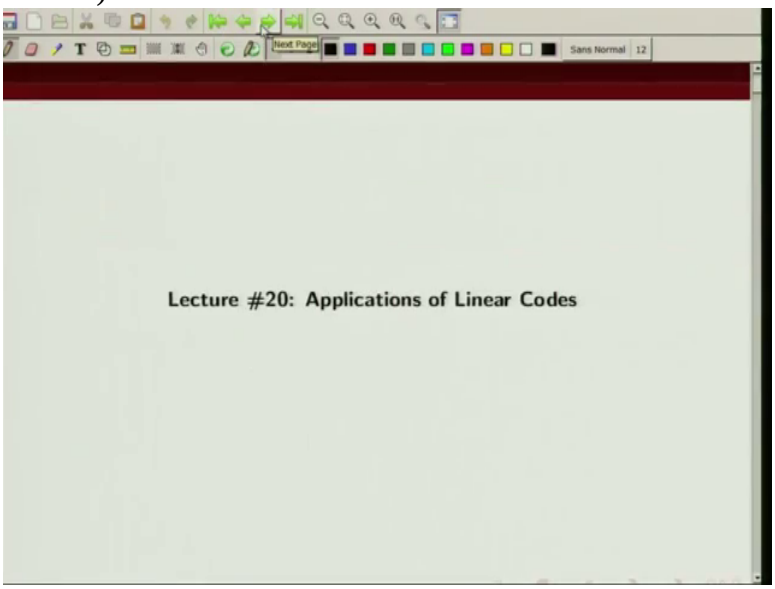
An Introduction to Coding Theory
Professor Adrish Banerji
Department of Electrical Engineering
Indian Institute of Technology, Kanpur
Module 08
Lecture Number 33
Applications of Linear Codes

(Refer Slide Time 00:14)



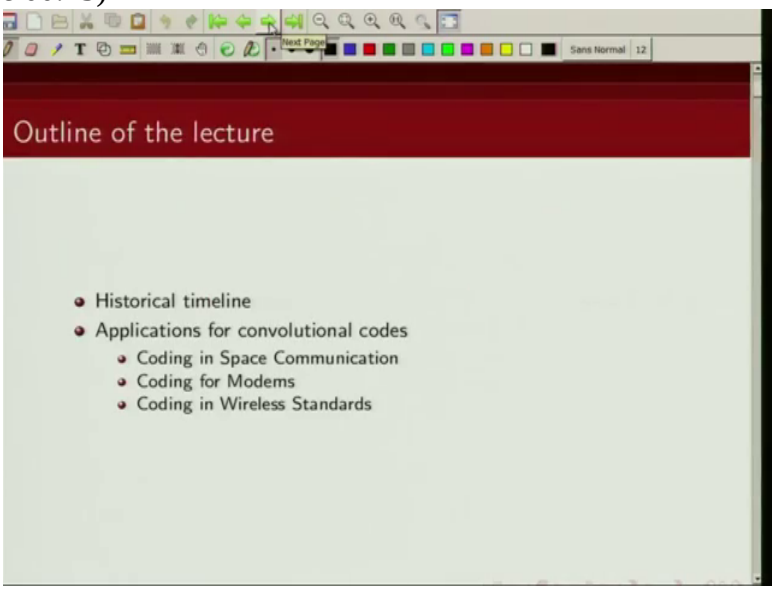
Welcome to the course on An introduction to coding theory. In this lecture we are going to talk about

(Refer Slide Time 00:21)



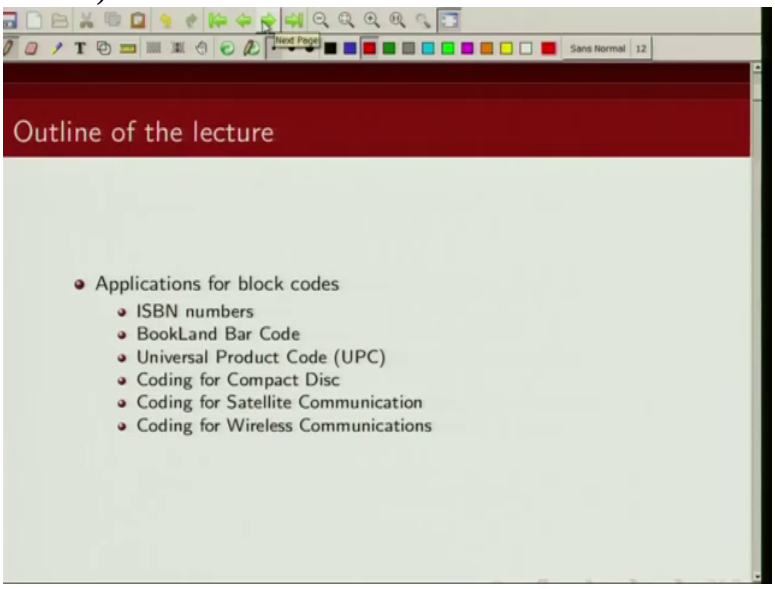
applications of linear block codes.

(Refer Slide Time 00:25)



So before we do that, we will first talk about a brief historical timeline of how error control coding has progressed over years. Then we are going to talk about applications for convolutional codes, in particular we will talk about applications of convolution code in space communication, applications of convolution code in wireline modems and applications of convolution code and turbo code in wireless communication standard.

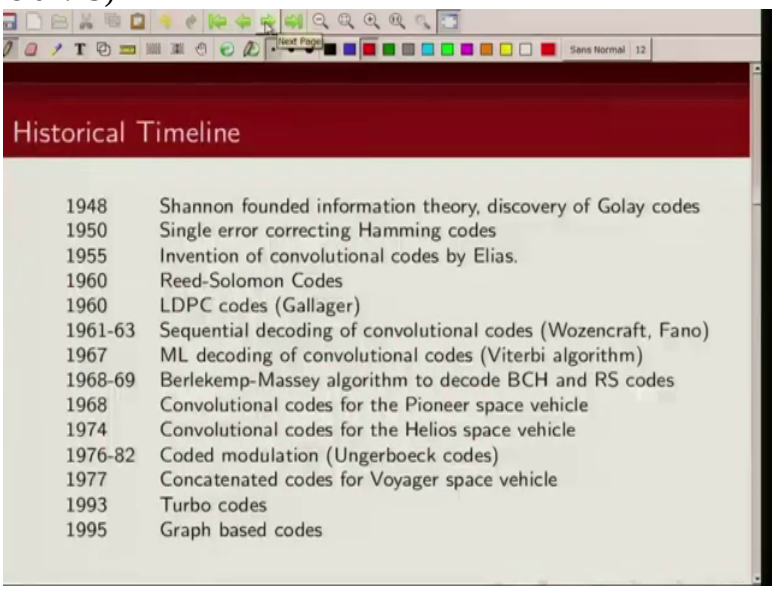
(Refer Slide Time 01:01)



Next we are going to talk about some applications of linear block codes. In particular we are going to talk about I S B N number, how an error detecting code is used in I S B N number. Similarly we will talk about application of error detecting codes in BookLand Bar Code. We are also going to talk about application of error detection code in U P C code as well as European Article Number and then we are going to talk about applications of linear block

codes for compact disk, applications of linear block code for satellite communication and applications of linear block code for wireless communication.

(Refer Slide Time 01:49)

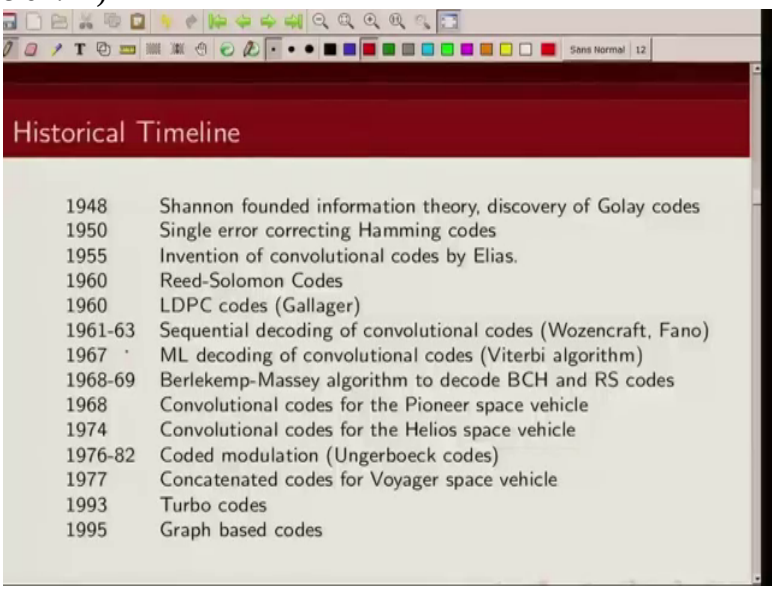


So as we know, 1948 Shanon founded Information Theory and then subsequently around 1955 Elias came with convolutional codes. Around 1960s Reed Solomon code and L D P C codes were discovered and around 61-63, Wozencraft and Fano came up with sequential decoding of convolutional code which allowed us to decode convolutional code with very large memory order.

(Refer Slide Time 02:29)

Around 1967-68 Viterbi algorithm for Maximum Likelihood decoding of convolutional code came and then around 74, B C J R algorithm was proposed.

(Refer Slide Time 02:41)

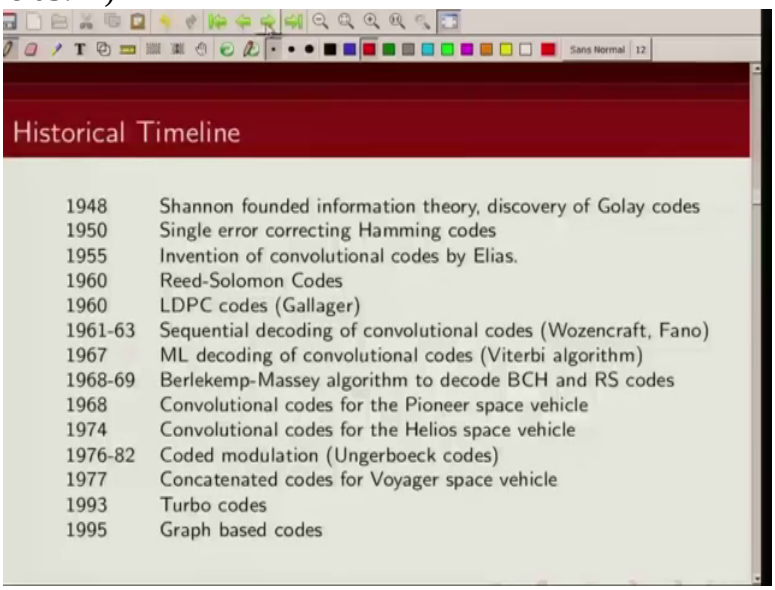


In late 60s and early 70s, convolutional codes and

(Refer Slide Time 02:45)

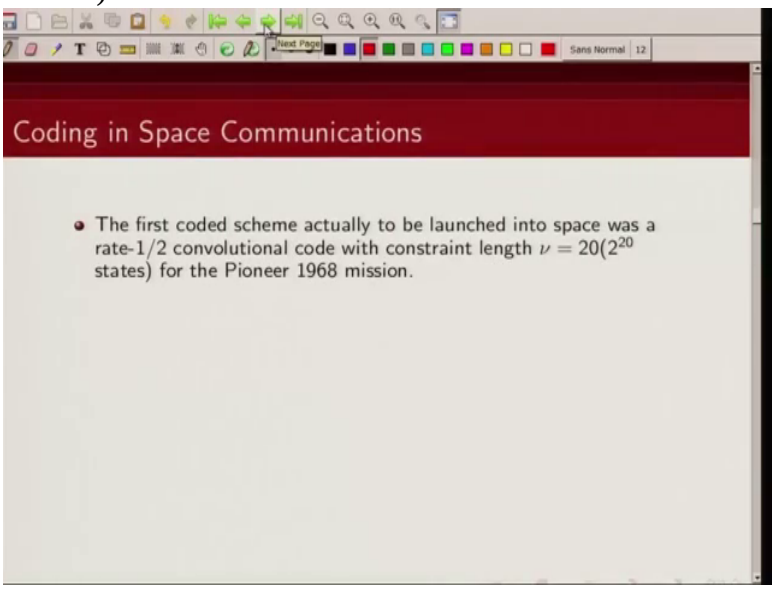
concatenated codes found applications for deep space applications and around 1977, coding and modulation was combined together uh design Trellis coded modulation and they subsequently found lot of applications in designing coding for modems. And then finally around 1993, turbo codes were discovered and L D P C codes were rediscovered and then from late 90s onwards to early 2000 and people started proposing these codes for various standards and they found applications in many communication standards. So that's a brief outline of how things have evolved over

(Refer Slide Time 03:42)



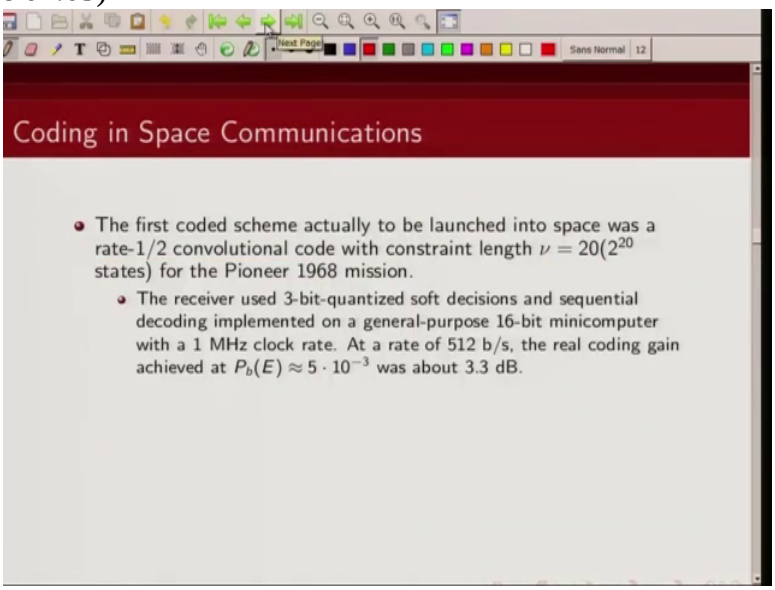
50, 60 years.

(Refer Slide Time 03:43)



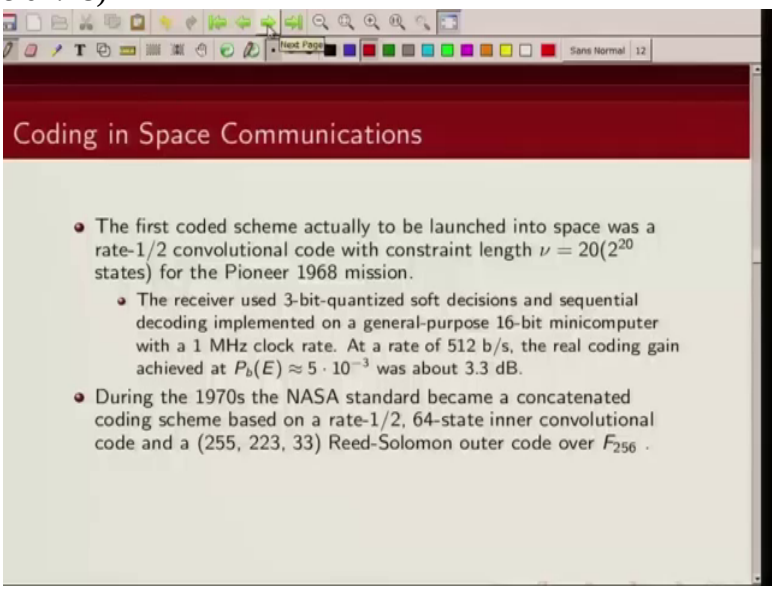
So let's start with applications of convolutional code. Now Pioneer Mission was in 1968 and they used rate half convolutional code which has constraint length of 20, so there were 2 to power 20 states and they used sequential decoding to decode

(Refer Slide Time 04:03)



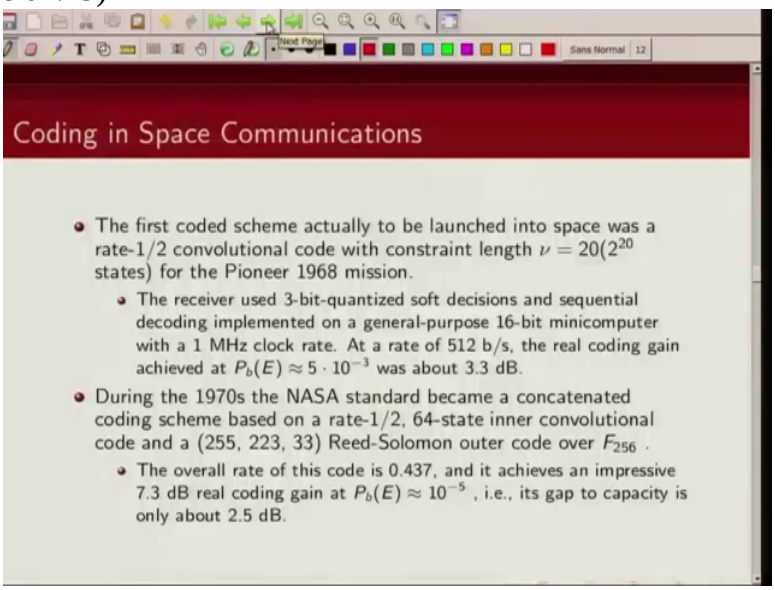
this code. The receiver used a 3 bit quantized soft decisions and it was implemented on a 16 bit minicomputer. It provided the coding gain of roughly 3 point 3, it got like for a probability of around this it gave a coding gain of about 3 point 3 d B.

(Refer Slide Time 04:28)



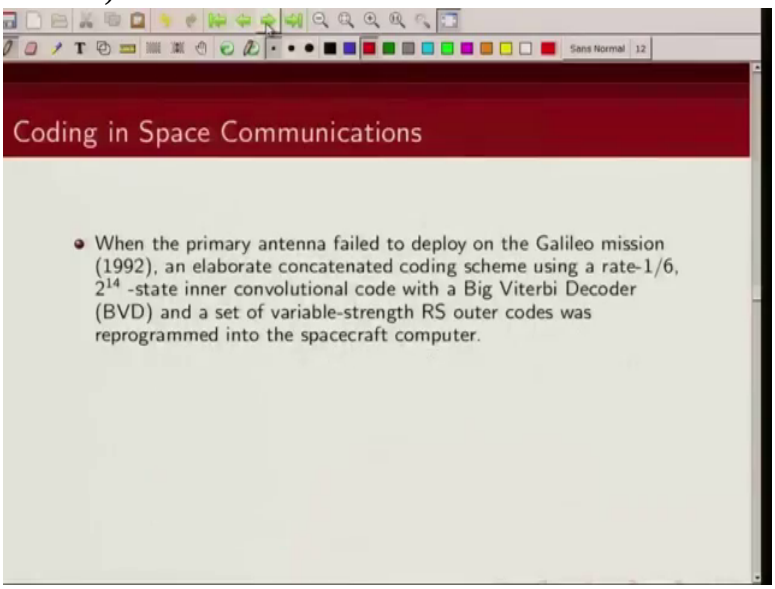
Next around 70s, NASA started using this concatenated code. Outer code was a Reed Solomon Code and inner code was 64 state convolution code, rate half convolutional code and it found lot of applications.

(Refer Slide Time 04:49)



The overall code rate of this was point 4 3, and gap to capacity was only 2 point 5 d B. It provided coding gain of around 7 point 3 d B at bit error rate of to 10 to minus 5.

(Refer Slide Time 05:06)

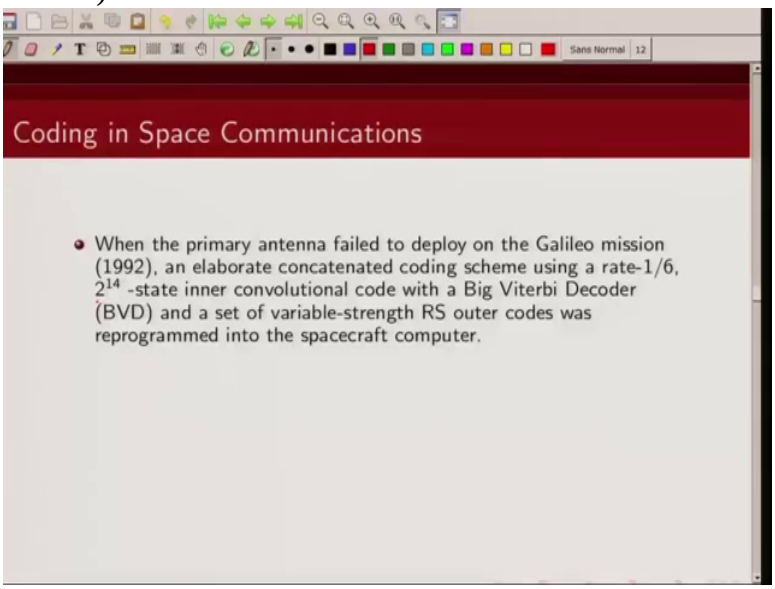


Another big development was when in Galileo's mission this antenna failed to deploy, primary antenna when it failed to deploy so there was an elaborate

(Refer Slide Time 05:20)

concatenated coded scheme which used

(Refer Slide Time 05:23)

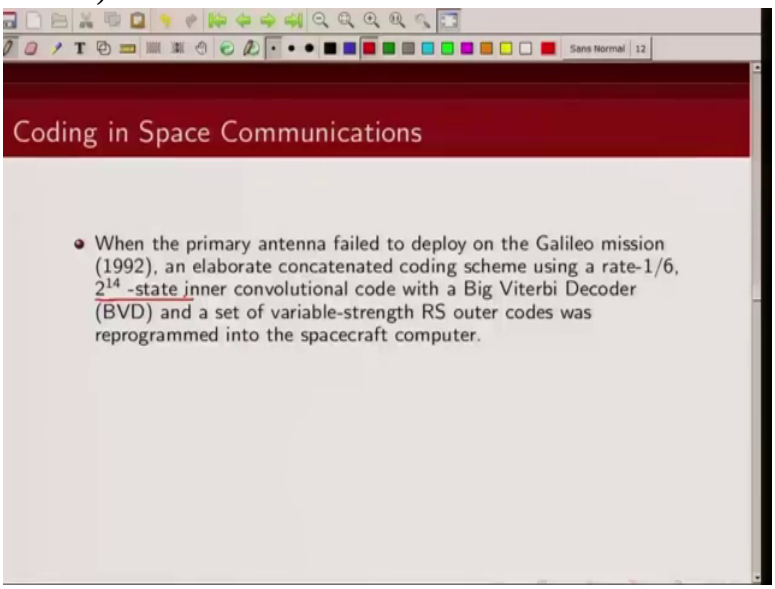


a 2 raised to power 14 state convolutional code as inner code

(Refer Slide Time 05:30)

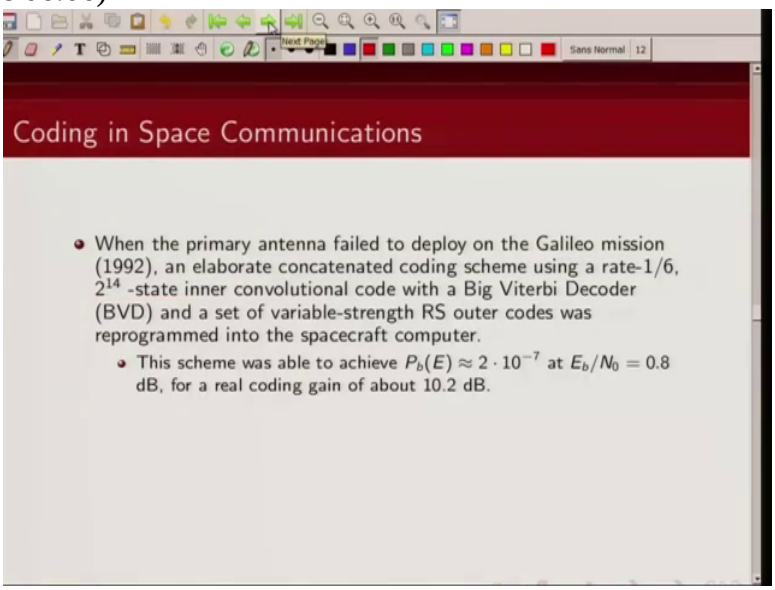
and Reed Solomon code as outer code and uh a big Viterbi decoding a decoding, Viterbi decoding algorithm was proposed which could work over such a large number of states

(Refer Slide Time 05:46)



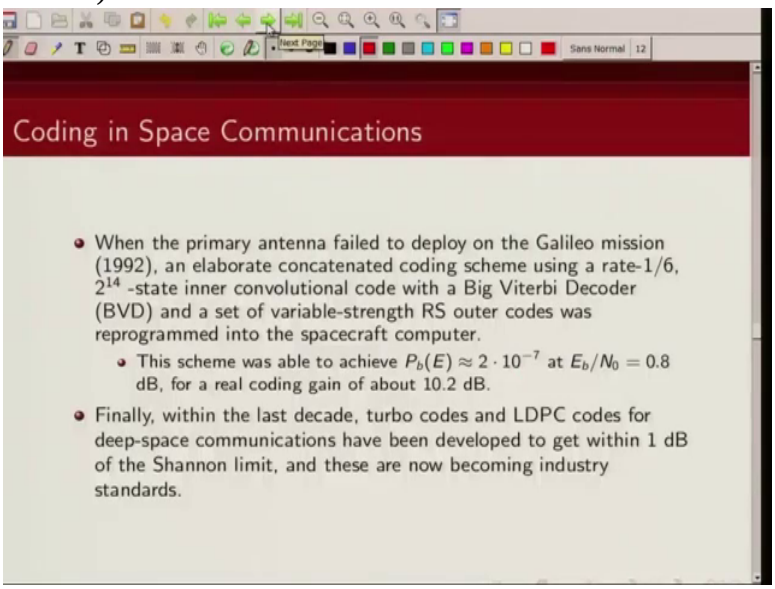
and that was also a very impressive step forward in terms of coding application was concerned because it was able to salvage this mission because of this very nice concatenated code used and it was successfully

(Refer Slide Time 06:06)



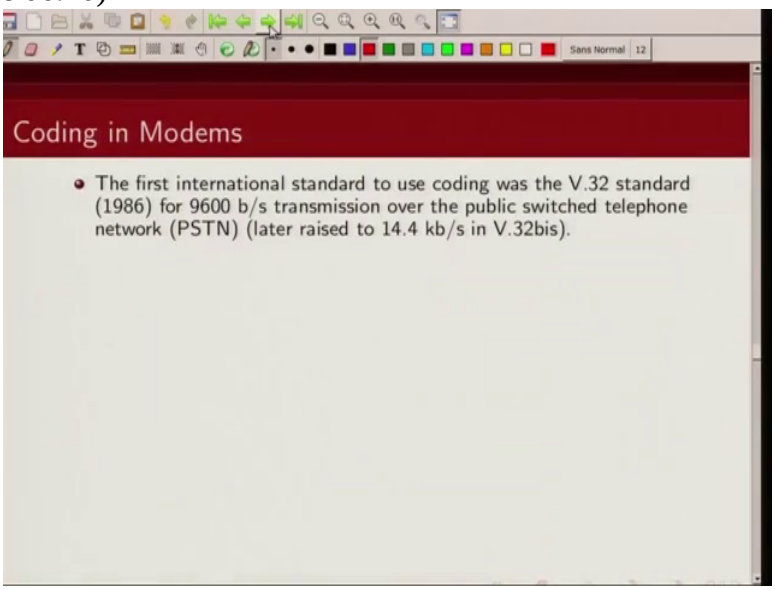
deployed. It provided a coding gain of around 10 point 2 dB at, and it was able to achieve bit error rate close to 10 to minus 7.

(Refer Slide Time 06:19)



And finally as I said, from early 2000 now, this is era of turbo code and L D P C codes and they are finding applications from deep space to mobile communication standards, TV broadcasting and things like that.

(Refer Slide Time 06:40)

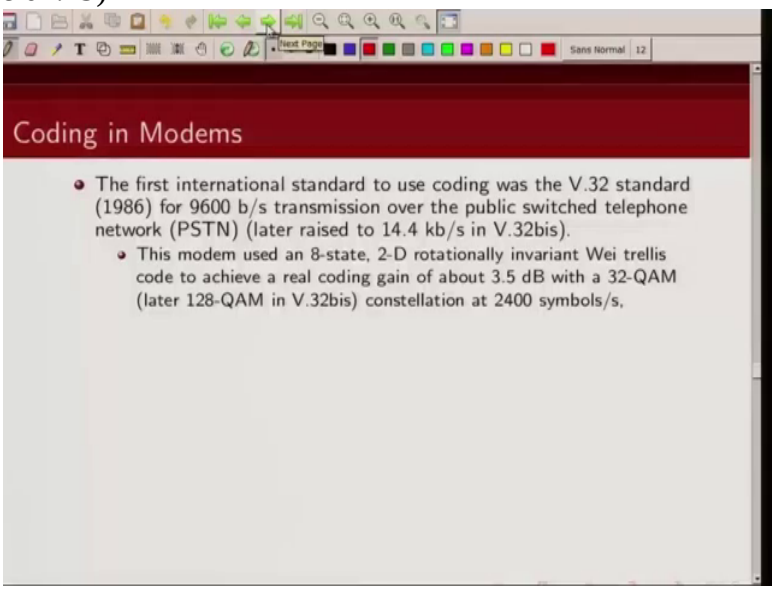


Now another area where error control coding and

(Refer Slide Time 06:46)

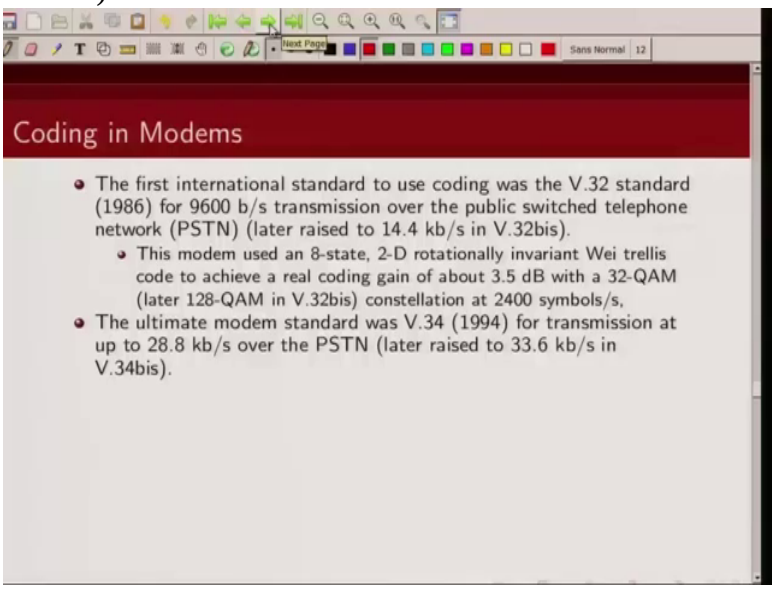
this convolutional code made major impact was in coding for modems, so around mid 70s Ungerboeck came with how to combine coding and modulation and he came up this Trellis coded modulation and these codes were heavily used in designing coding schemes for data transmission basically over public switched telephone network. So some of the

(Refer Slide Time 07:19)



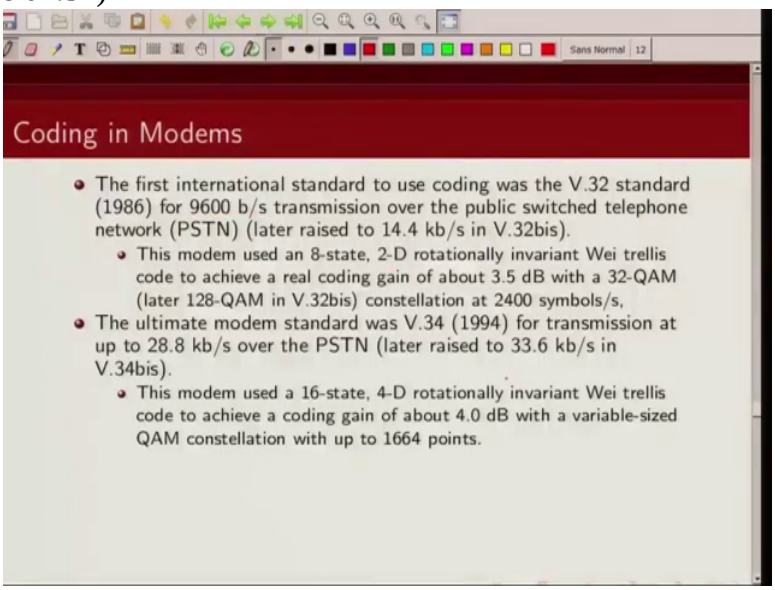
applications I have listed here,

(Refer Slide Time 07:21)



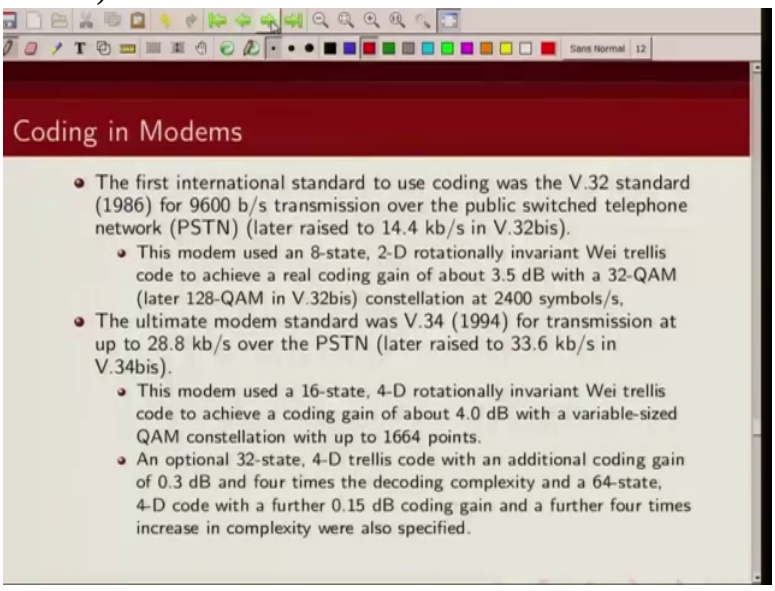
so initially we had 9 point 6 k b p s transmission that used an 8 state code, then finally for a

(Refer Slide Time 07:32)



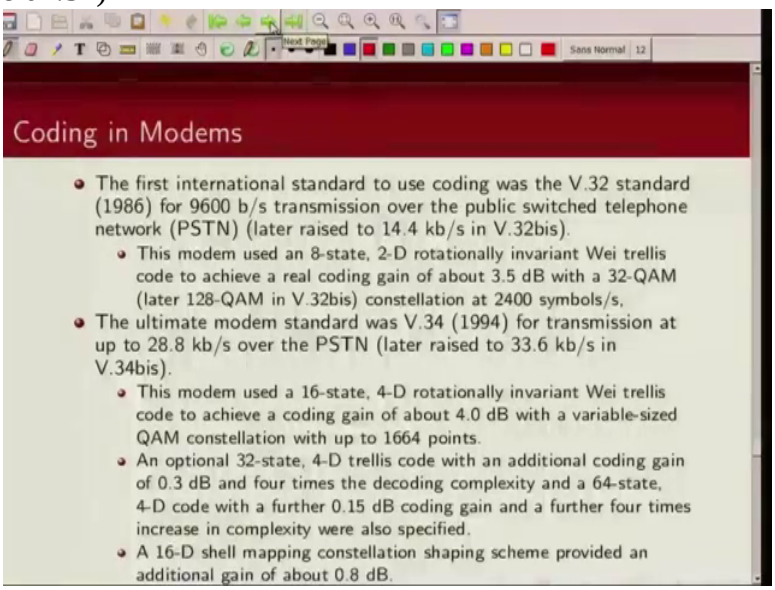
33 k b p s transmission they use a 16 state code dimensional Trellis coded modulation scheme and

(Refer Slide Time 07:42)



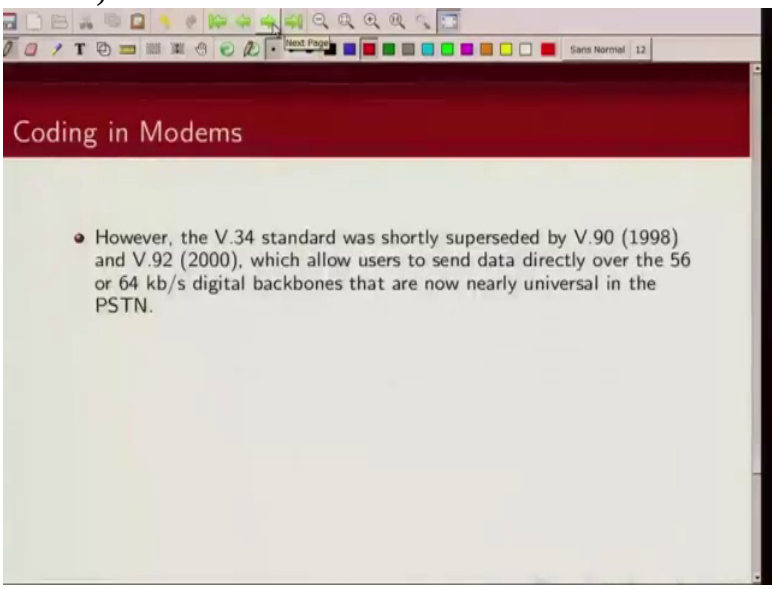
there was an optional 32 state 4 dimensional Trellis code which was also specified in this standard. But

(Refer Slide Time 07:52)



when we

(Refer Slide Time 07:54)

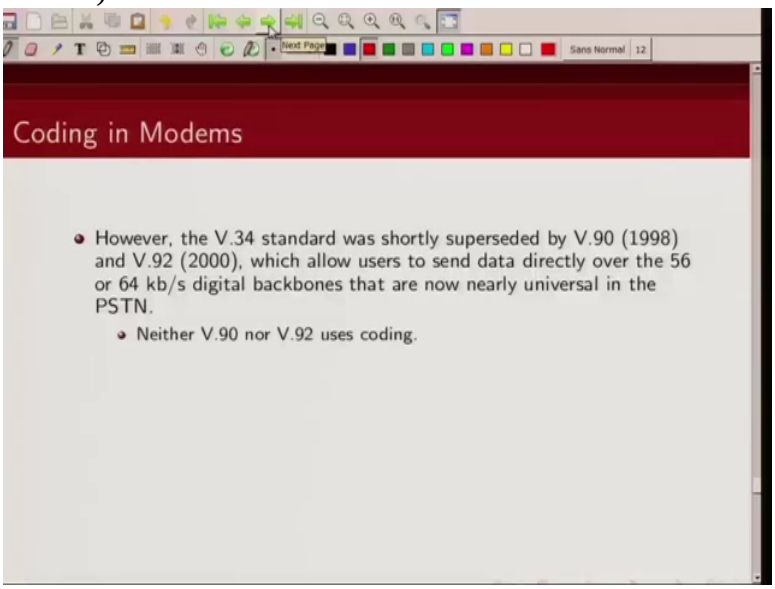


progressed to digital backbone then basically there was

(Refer Slide Time 08:00)

no need for these codes and so the V 90

(Refer Slide Time 08:06)

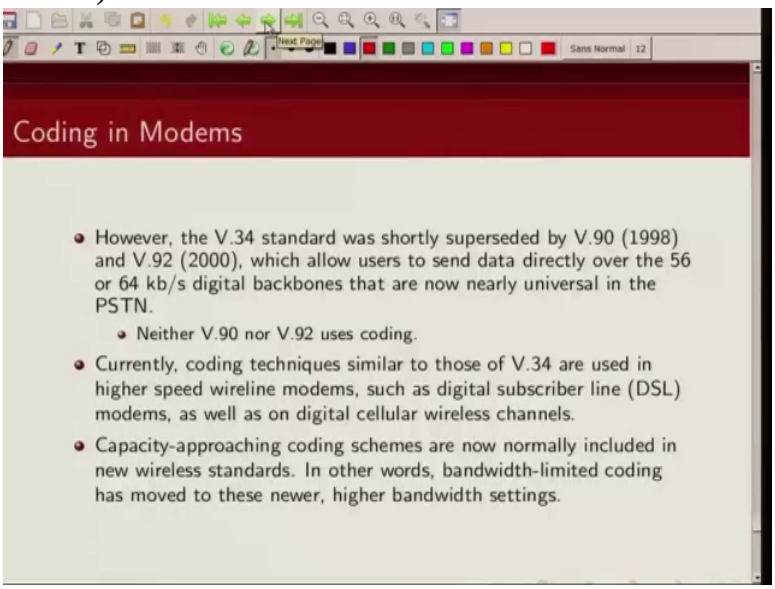


or V 92 standards, they essentially use any of these coding techniques. But in

(Refer Slide Time 08:12)

mid 80s and early 90s, there were lot of, there were very successfully these codes were applied for coding for modems.

(Refer Slide Time 08:24)

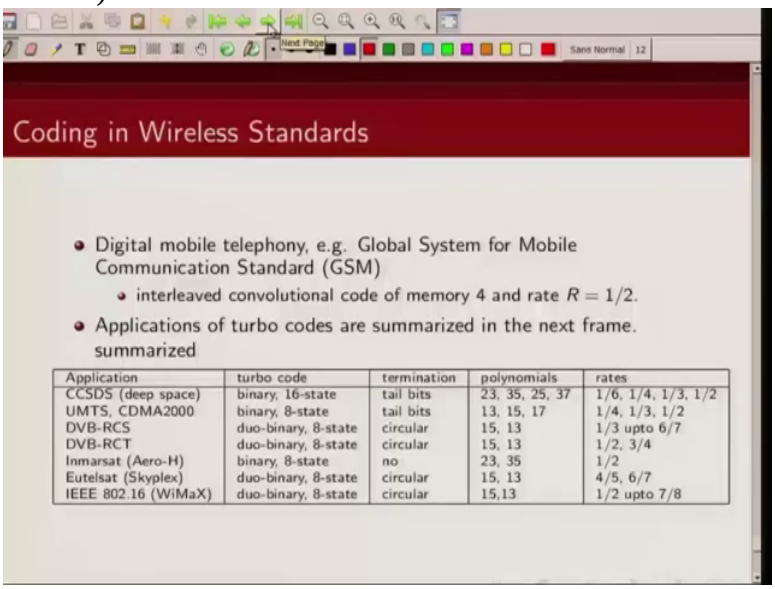


Now currently these techniques are also used for high speed wireline modems like A D S L and other applications and of course turbo codes and their variants are also finding applications

(Refer Slide Time 08:38)

here.

(Refer Slide Time 08:39)

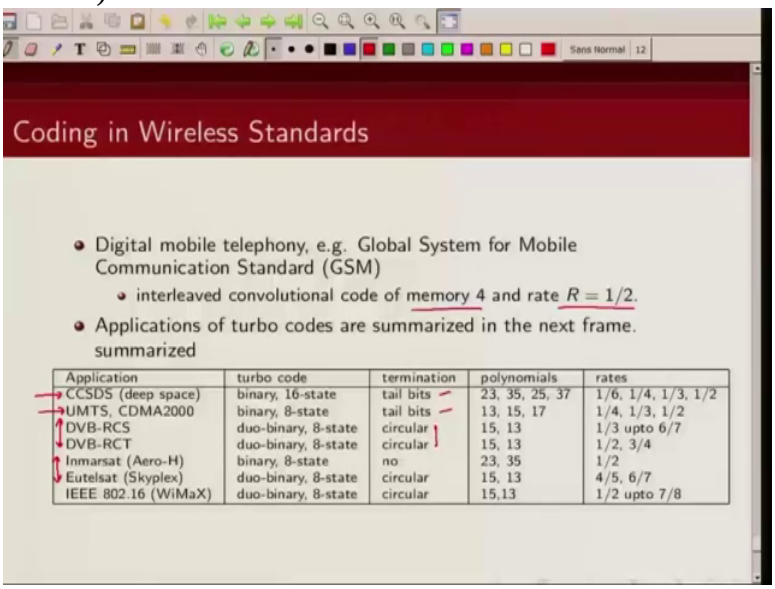


So I would like to conclude basically with giving some examples of communication system, so our G S M uses convolution code of memory 4 and rate one half code and in this table I have listed some of the applications of turbo codes in recent standards, I mean so starting from digital video broadcasting standard to third generation wireless standard to deep space exploration to satellite communication applications and you can see there are this duo binary turbo code that we talked about, this duo binary turbo code essentially uses like rate k by code which is greater than 1 and there are additional bits, tail bits that have been added in some of the cases for termination of this encoder

(Refer Slide Time 09:40)

and in some cases

(Refer Slide Time 09:41)

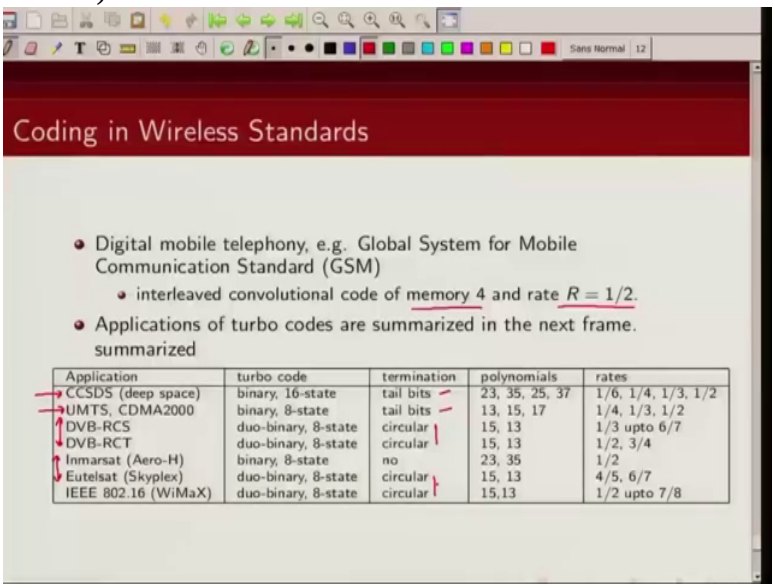


where you see this circular thing, they have used a circular

(Refer Slide Time 09:45)

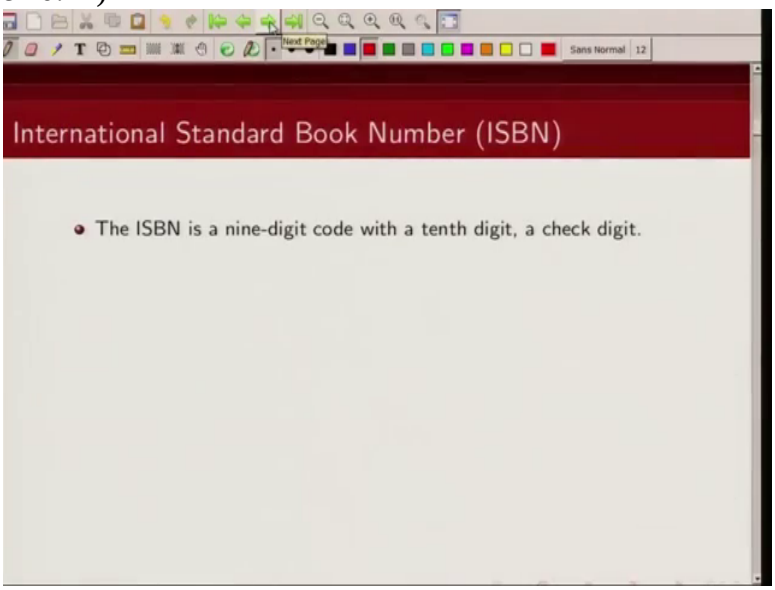
recursive encoder so these are what they call, what is known as tail biting encoder so the starting and ending state are same. So the Trellis is like a circular Trellis. So they don't need any termination bits.

(Refer Slide Time 10:01)



And these are the code generator polynomials in octal notation and some of the rates that have been used in these codes.

(Refer Slide Time 10:12)

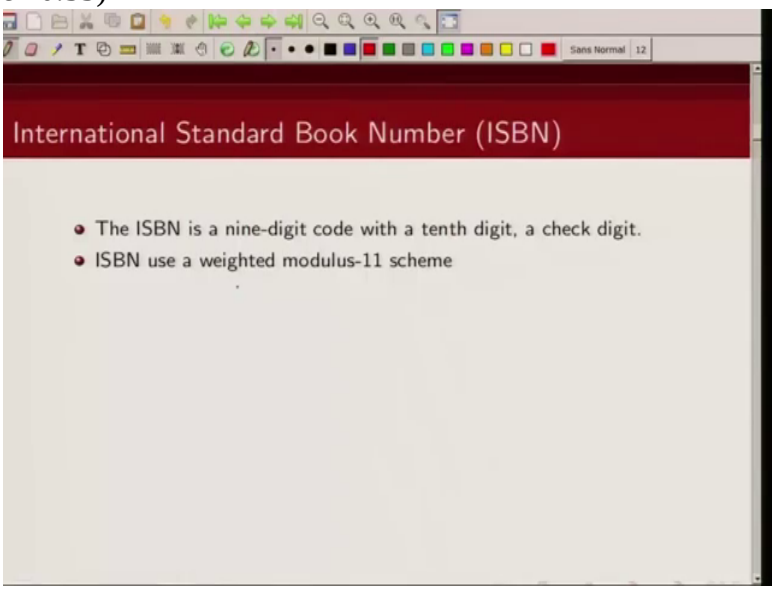


So what is an I S B N number? I S B N number is a 9 digit code with a ten digit which is a check digit. So it is a ten digit number. You have a nine digit code plus one bit which is a check bit. So how

(Refer Slide Time 10:30)

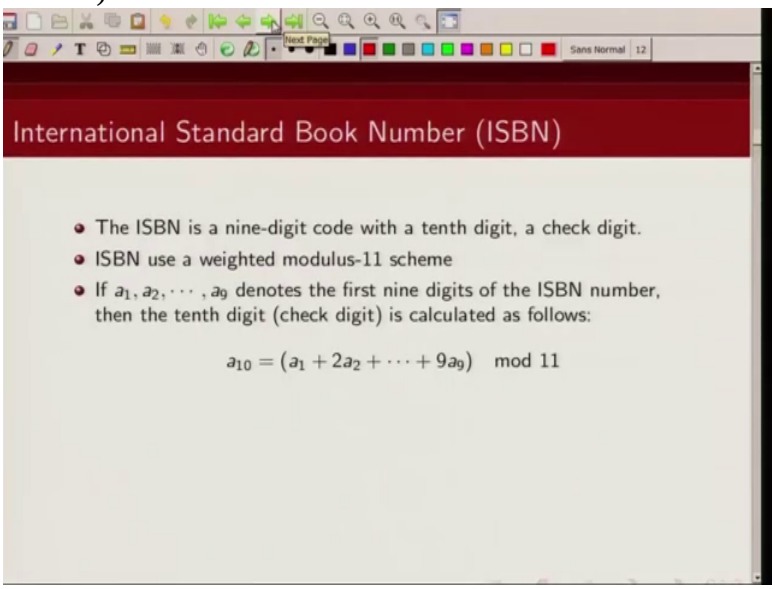
is this check bit calculated? This

(Refer Slide Time 10:33)



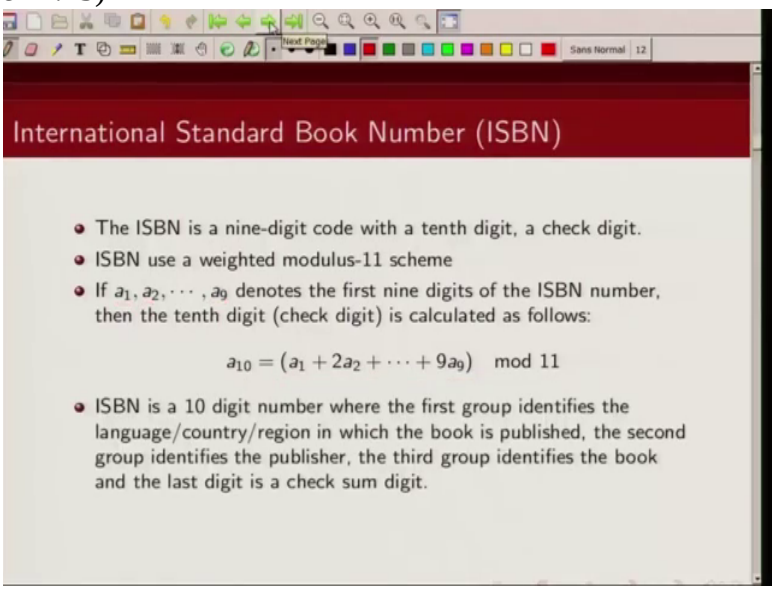
check bit is calculated as follows. It uses a weighted modulus 11 arithmetic.

(Refer Slide Time 10:40)



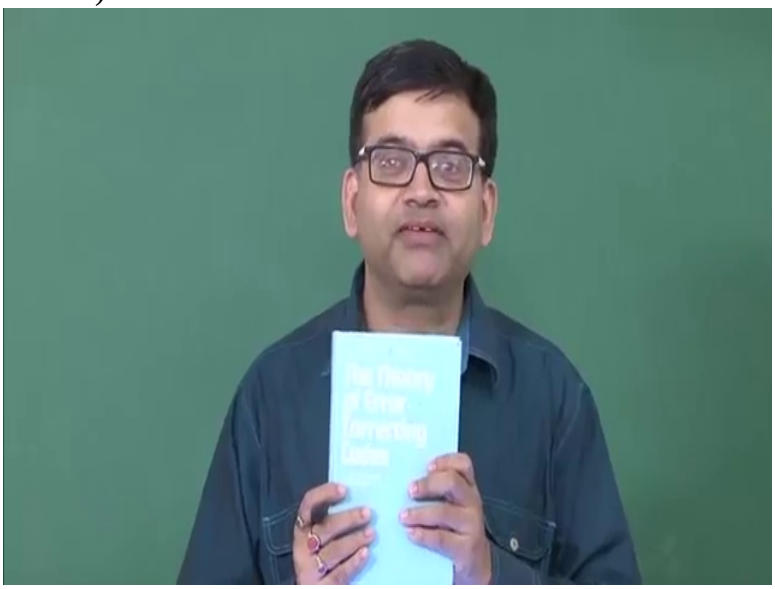
So let us denote the nine bits of I S B N number by a 1, a 2, a 3, and a 9. Then the tenth bit is calculated as follows. So a 1 plus 2 times a 2 plus 3 times a 3 plus 4 a 4 up to 9 times a 9 modulo 11 will give us the tenth bit. So tenth bit is actually a check bit.

(Refer Slide Time 11:13)



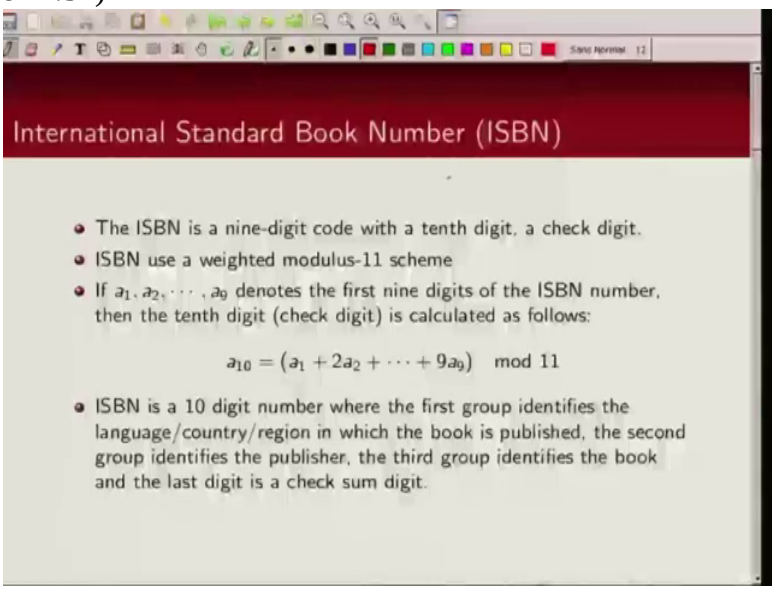
I S B N is typically a ten digit number where the first group identifies the country, region where the book is published, the second group identifies the publisher, and third group identifies the specified book and finally the last bit is actually a check bit. So let's take an example. Let's take this book.

(Refer Slide Time 11:39)



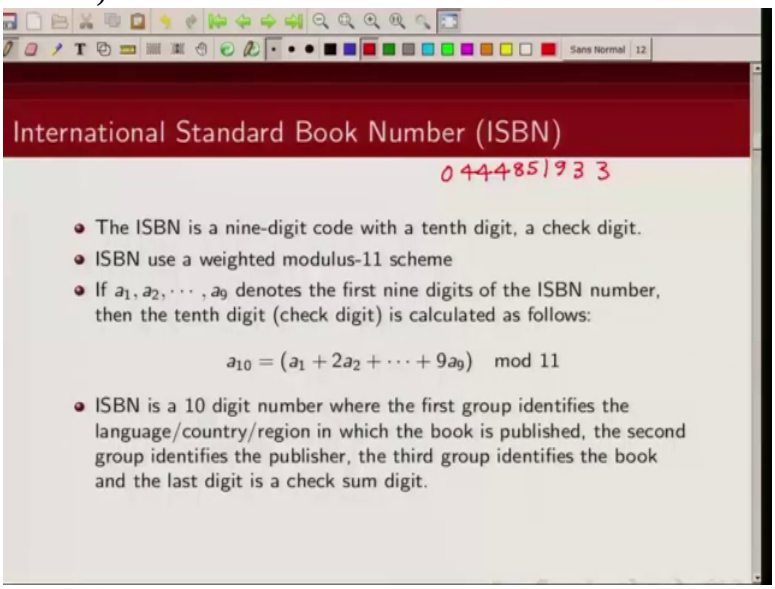
This is one of the reference books that we are using, The Theory of Error correcting Codes by MacWilliams and Sloane. So if we look at the I S B N number for this, this book, the I S B N number for this book is

(Refer Slide Time 11:54)



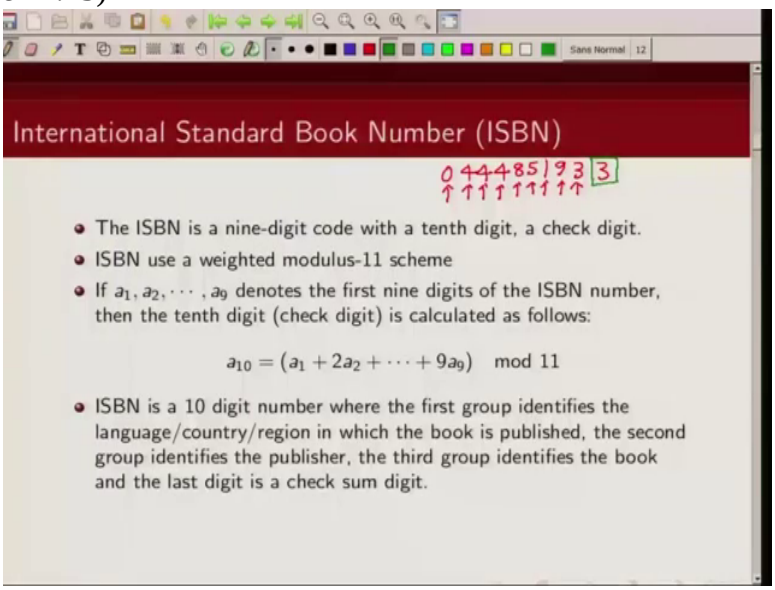
0 4 4 4 8 5 1 9 3 3,

(Refer Slide Time 12:06)



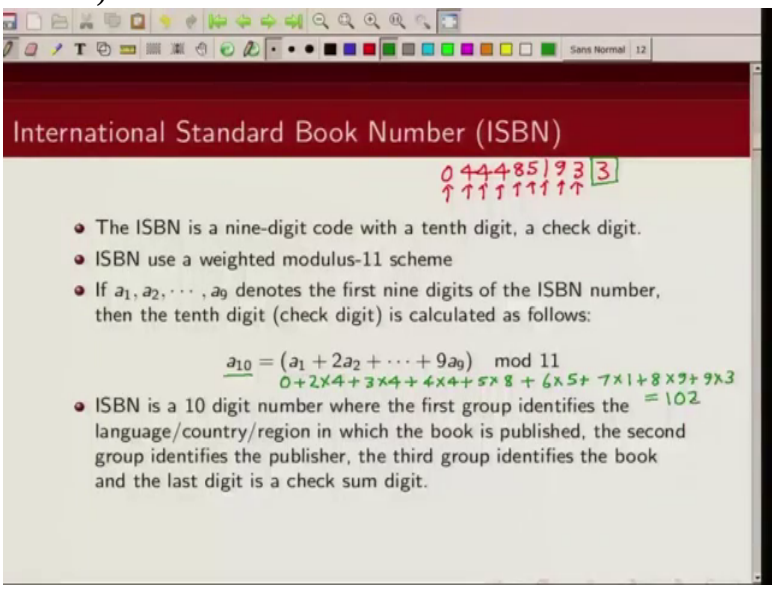
Ok. So let's check whether, so this is the first bit, second bit, third bit, fourth bit, fifth bit, sixth bit, seventh bit, eighth bit, and this is the ninth bit. And this is our check bit,

(Refer Slide Time 12:23)



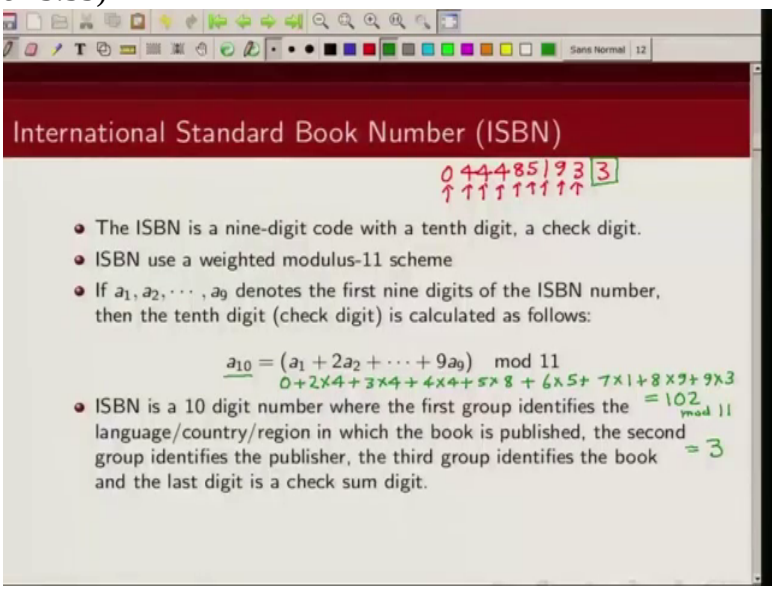
Ok. So let us check whether it is correct or not. So how is the check bit calculated? It is a 1, a 1 in this case is 0, a 2 times a 2 that is 4, three times a 3 that is 4, four times a 4 which is again 4 plus five times a 5, six times a 6, seven times a 7, eight times a 8 and nine times a 9, Ok and this comes out to be 1 0 2.

(Refer Slide Time 13:20)



Now 1 0 2 modulus 11 is so 11 nines are 99 so it will be 3 modulo 11. So this 1 0 2 modulus 11 is going to be 3

(Refer Slide Time 13:35)

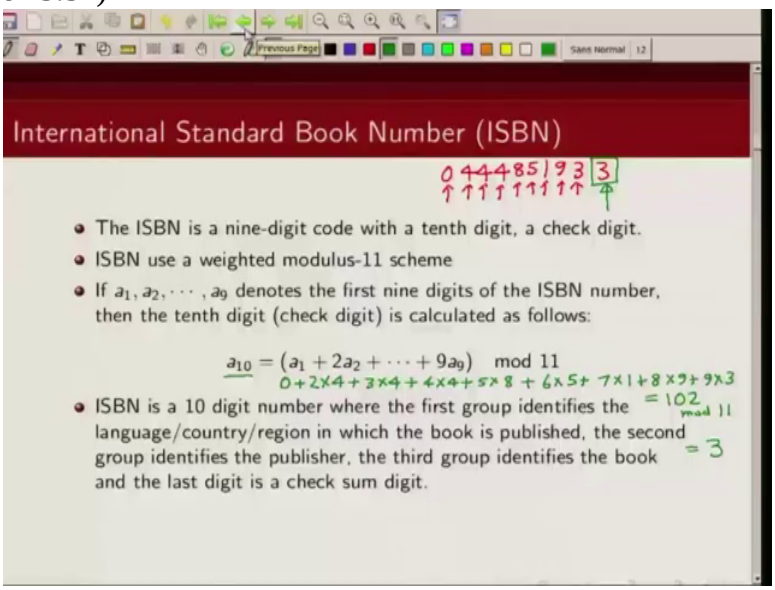


and that's precisely our check bit is, Ok. Now the interesting part about these I S B N numbers are that they can

(Refer Slide Time 13:49)

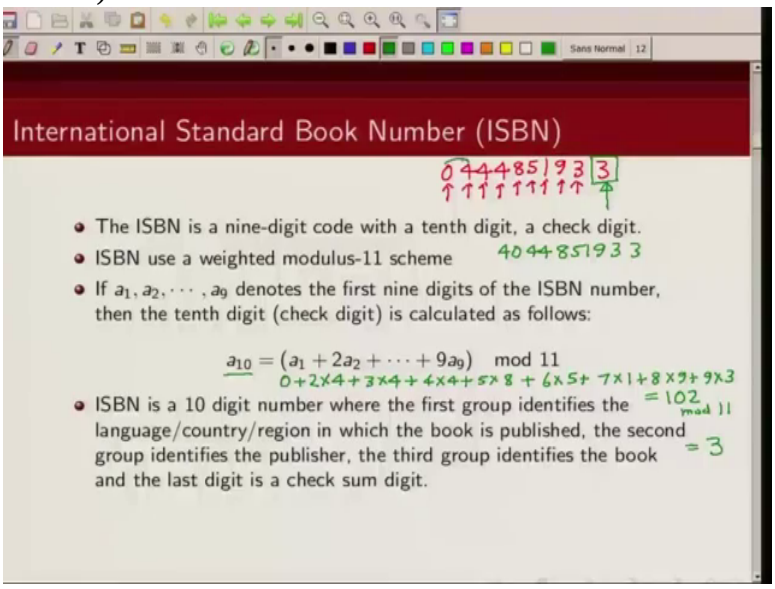
detect transposition error. So what is a transposition

(Refer Slide Time 13:52)



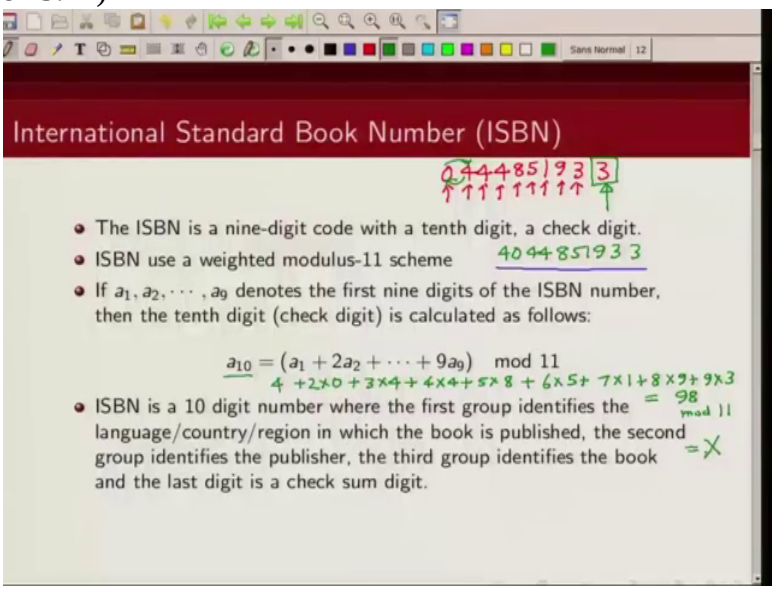
error? So let's say when you are reading this I S B N number with a bar code reader the most common mistake that happens is, the adjacent bits are read wrongly. So for example, this is 0 4 4 4 8 5 1 9 3 3. If it is read like 4 0 4 4 8 5 1 9 3 3, that means these 2 bits are

(Refer Slide Time 14:22)



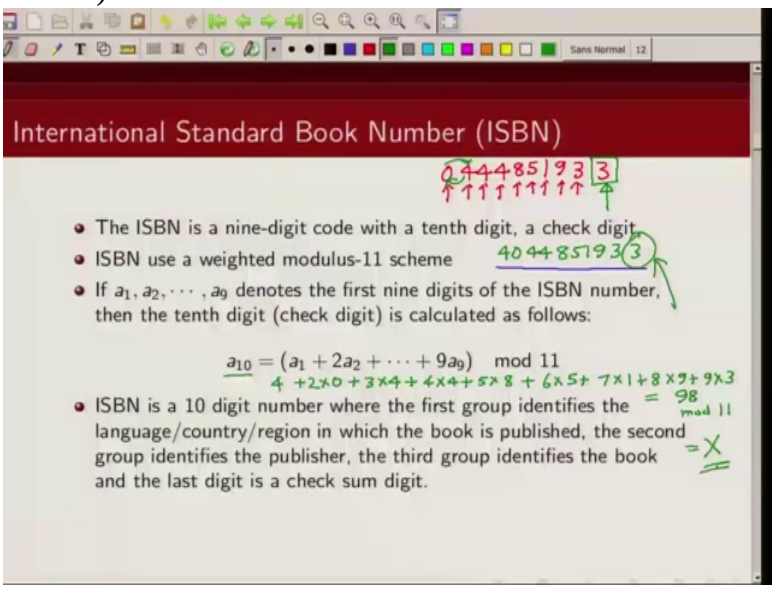
changed, this is transposition error. Then this particular code can detect such errors. We can verify this. Let's say this is the number which has been read by the barcode reader. We try to find out the check bit in this case. So this would be, the difference would be, here it is, these 2 things will be changed because these 2 bits got, so it will be 4 plus 2 times zero and that would be 98. So this will be 98 mod 11 and what is 98 mod 11, 11 eights are 88, so that is 10, so that would be x.

(Refer Slide Time 15:14)



Note that this check bit is not matching with the

(Refer Slide Time 15:22)

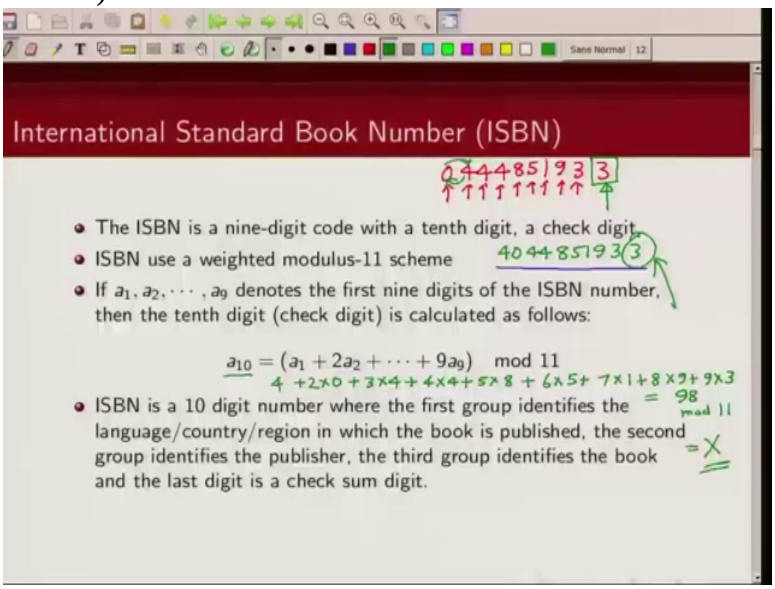


check bit calculated this way. So you can see this particular I S B N code can detect transposition error.

(Refer Slide Time 15:31)

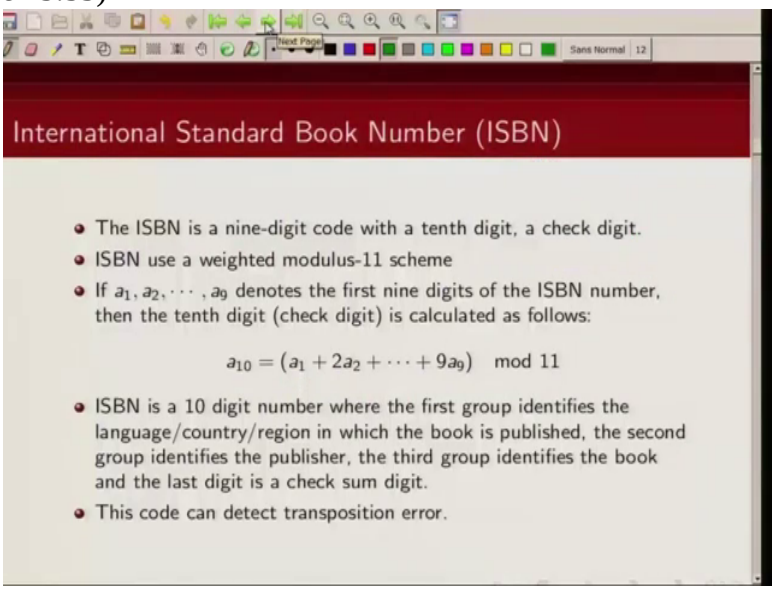
The correct

(Refer Slide Time 15:33)



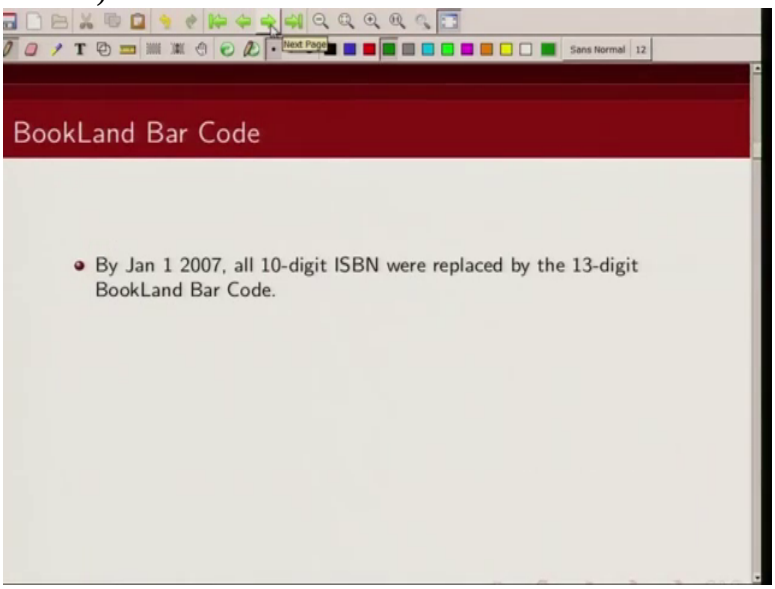
number was 0 4 4 4 8 5 1 9 3 3, and these two bits got interchanged. Transposition error happened and you can see using this I S B N number, we are able to detect such errors,

(Refer Slide Time 15:53)



Ok. Now

(Refer Slide Time 15:54)

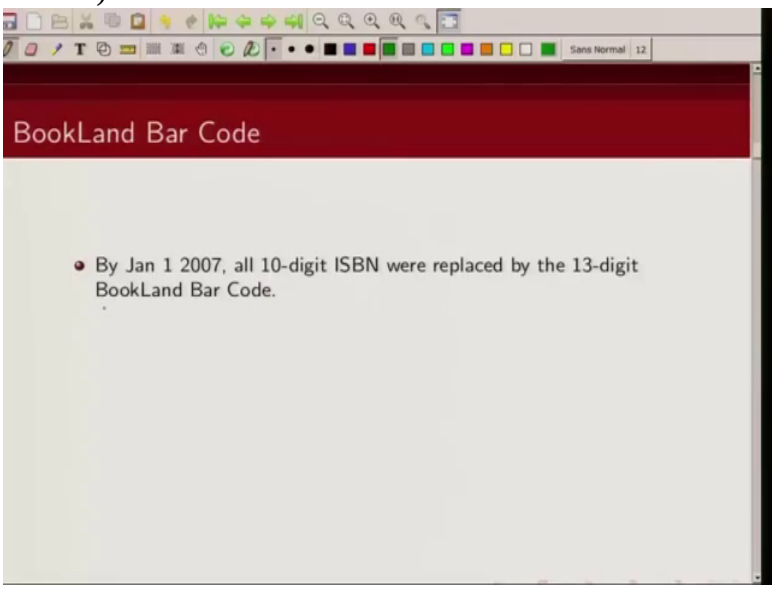


by 2007

(Refer Slide Time 15:56)

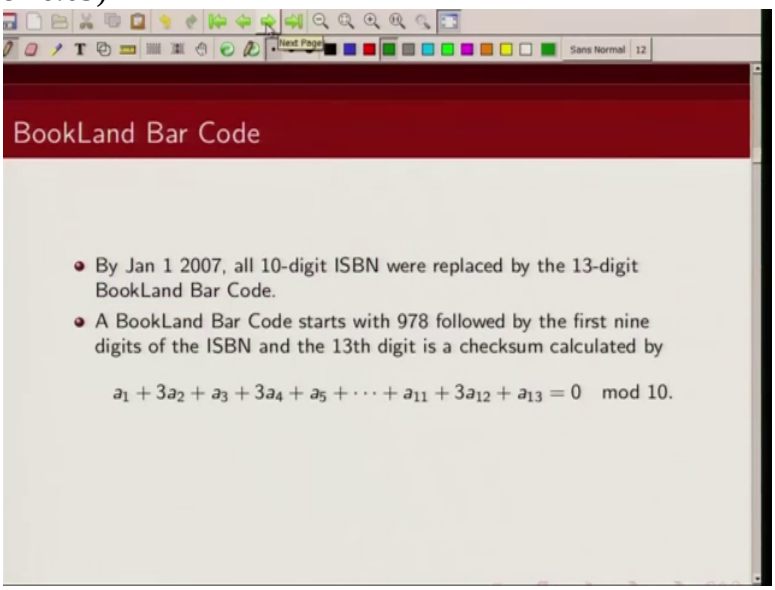
all these I S B N numbers were replaced by

(Refer Slide Time 15:59)



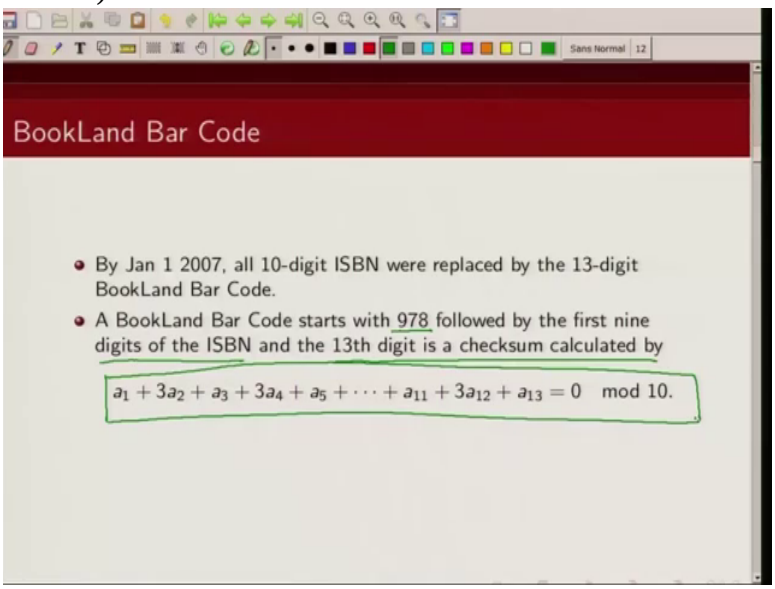
this 13 digit BookLand Barcode Number and what are these BookLand Barcode Numbers?

(Refer Slide Time 16:09)



So a BookLand Barcode Number starts with 9 7 8 and then it is followed by 9 digits of I S B N number and the thirteenth bit is a checksum which is calculated in this particular way, Ok.

(Refer Slide Time 16:28)

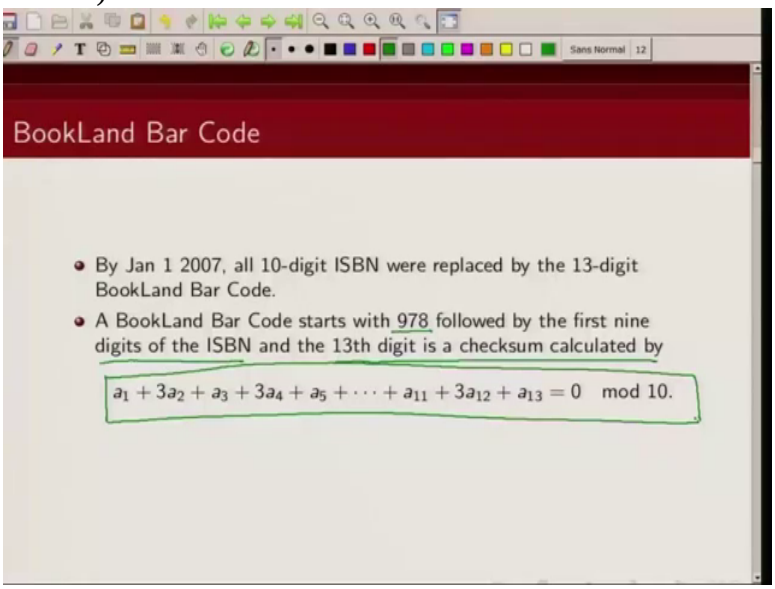


This is how we are

(Refer Slide Time 16:30)

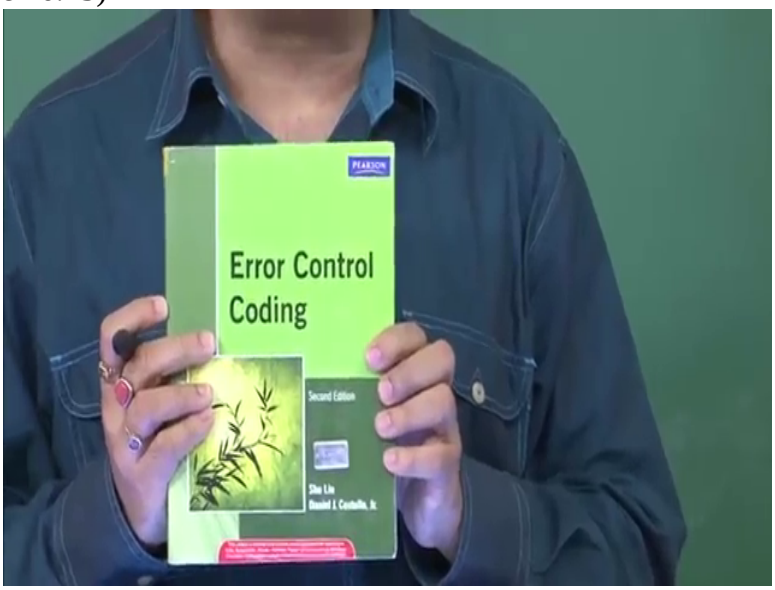
computing the thirteenth bit which is the checksum bit. Now

(Refer Slide Time 16:35)



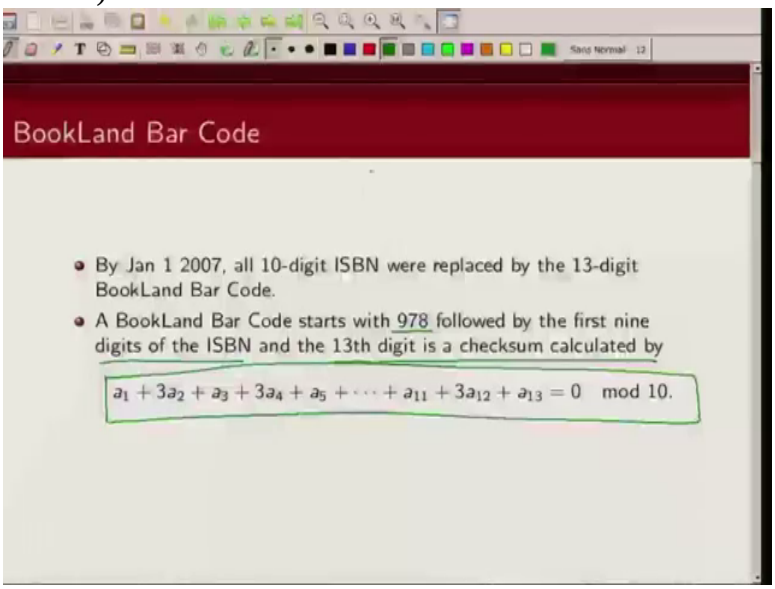
let us look at an example. So let's consider our book. This Error Control Coding by

(Refer Slide Time 16:43)



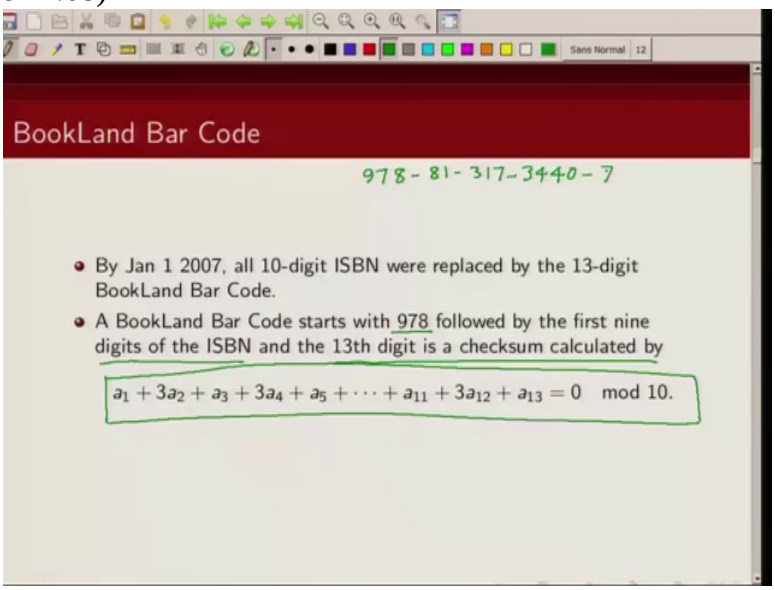
Lin, Costello and what is its BookLand Barcode Number? That's given by

(Refer Slide Time 16:51)



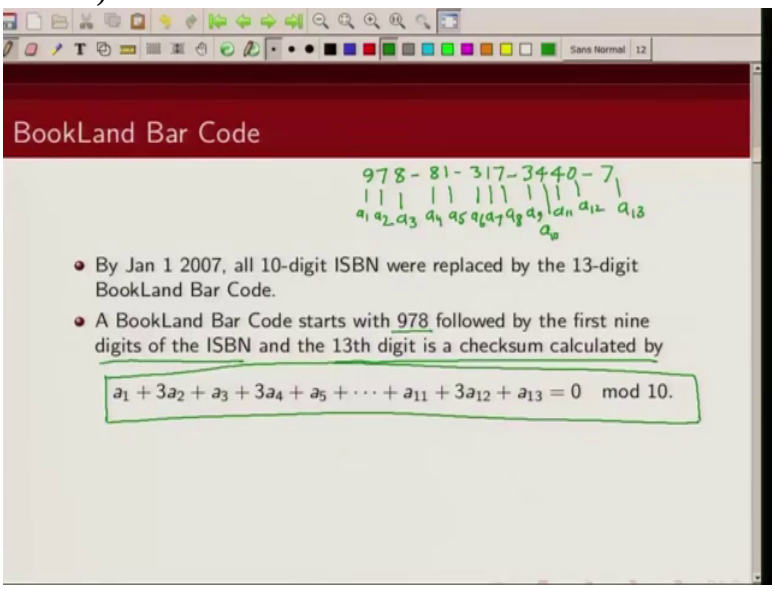
9 7 8 8 1 3 1 7 3 4 4 0 7. So this is the 13 bit

(Refer Slide Time 17:08)



digit number which is given by, which is the I S B N number, 13 digit Bookland Barcode Number of this text book that we are using. Now this is my a 1, this is my a 2, this is my 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's

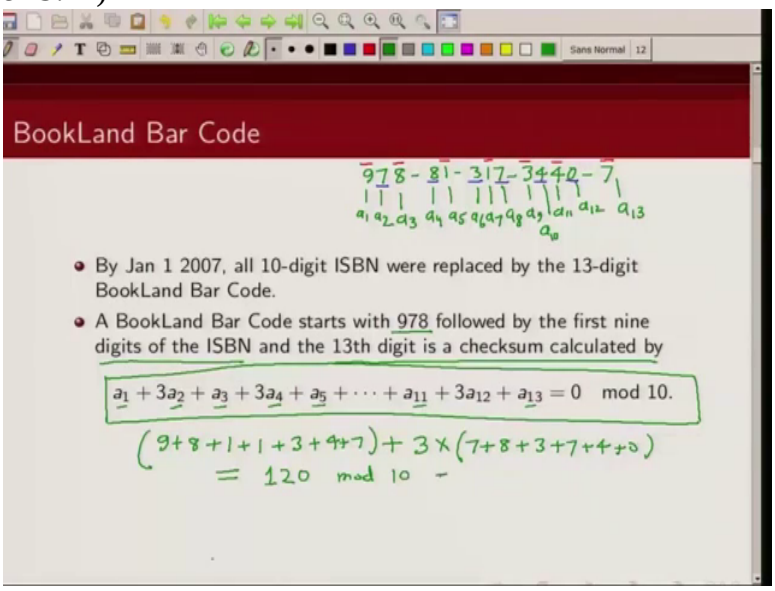
(Refer Slide Time 17:45)



look at whether we are getting the same check bit as rated by this. So let's, 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 plus, let me first take all the odd bit numbers. Odd bit numbers are these. Now once I am marking by, these are my odd locations, right? So this is 9 plus 8 plus 1 plus 1 plus 3 plus 4 plus 7 plus three times

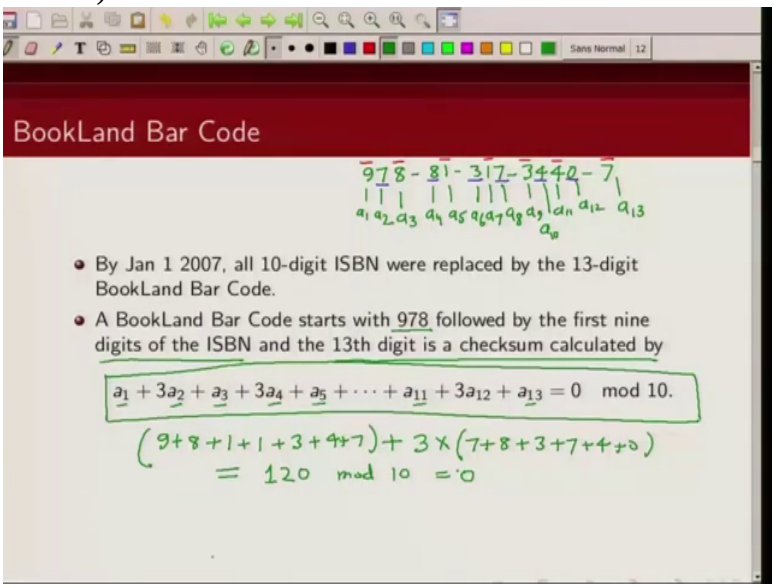
even numbers; even numbers are these, 7, 8, 3, 7, 4, 0. So this is 7 plus 8 plus 3 plus 7 plus 4 plus 0, Ok and this comes out to be 120 mod 10

(Refer Slide Time 19:22)



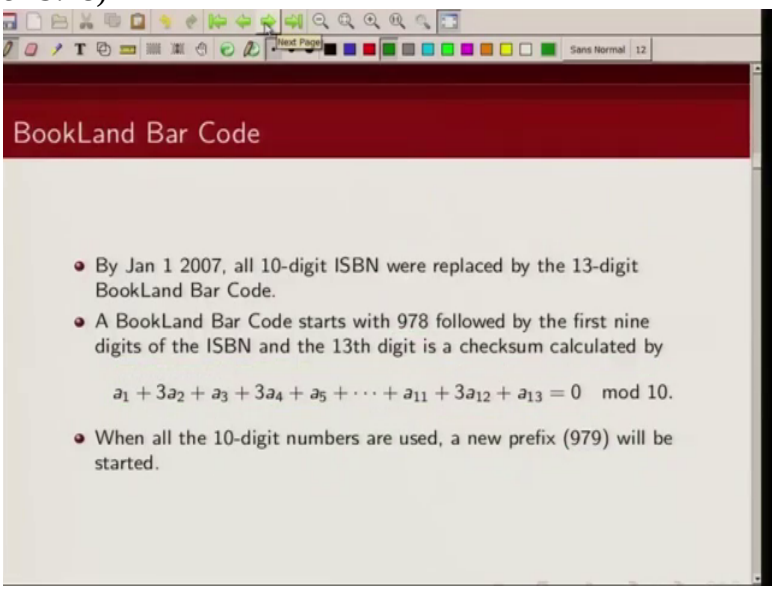
which is 0. So you can see

(Refer Slide Time 19:25)



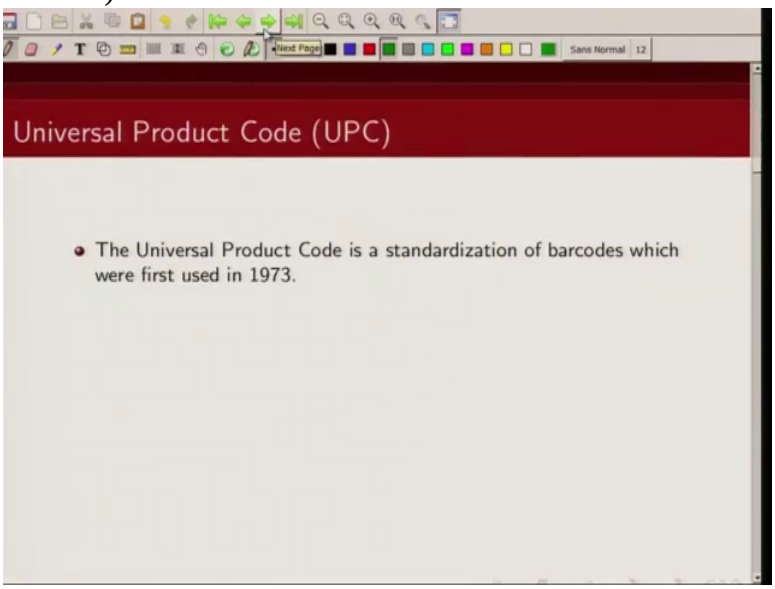
from this example that this check bit is correct because a 13 plus this should be 0 mod 10. So this is example of a BookLand Bar Code. And all the books have these 13 digit I S B N number.

(Refer Slide Time 19:48)



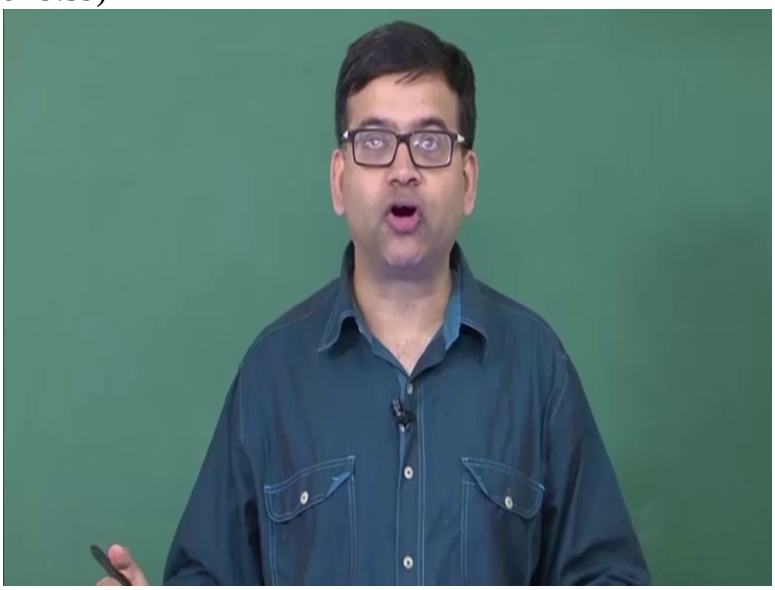
Now once they will get exhausted with this 9 7 8 number they will start numbers with 9 7 9.

(Refer Slide Time 19:55)



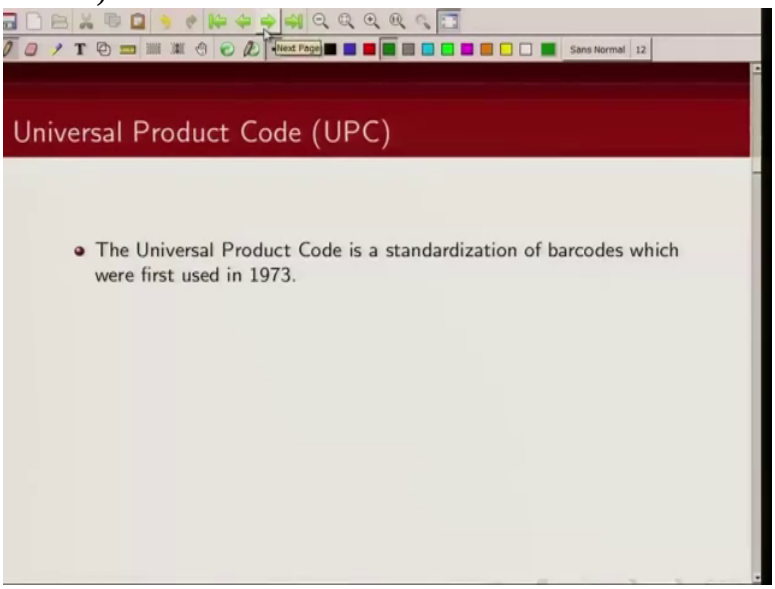
Now next application that we are going to

(Refer Slide Time 19:59)



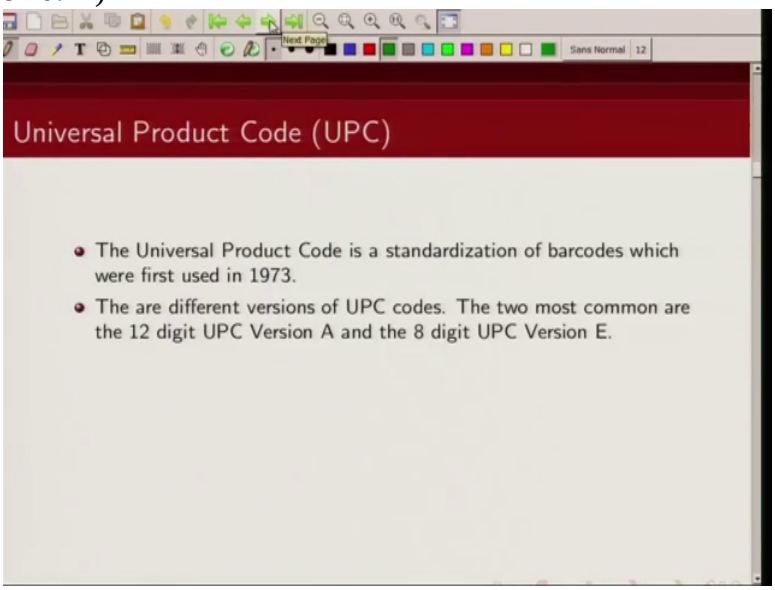
talk about is this Universal

(Refer Slide Time 20:02)



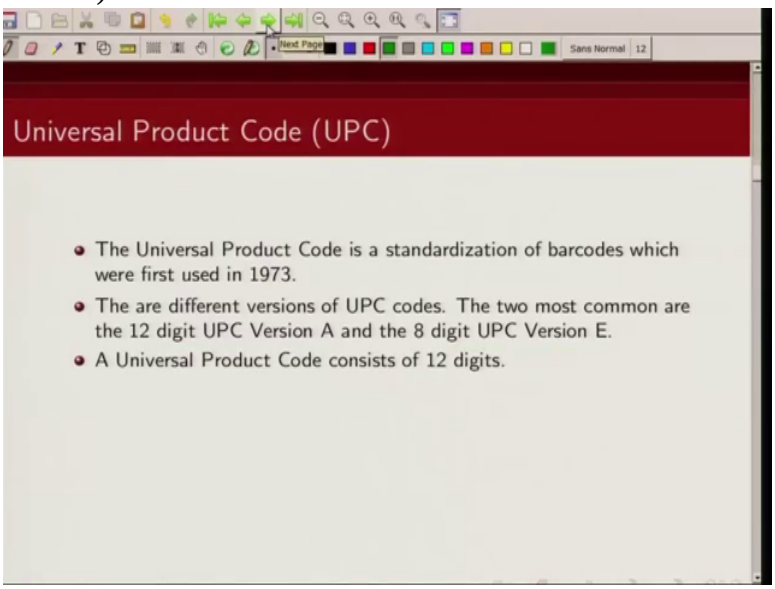
Product Code. So this was basically a standardization of all bar codes which was first used in 1973. So there are

(Refer Slide Time 20:11)



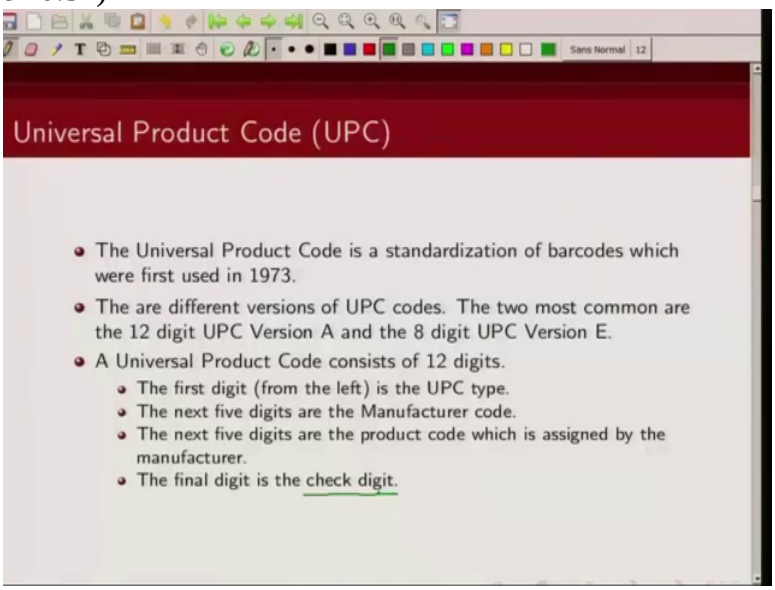
2 versions of it. One is this 12 digit U P C version A and the other is this 8 bit U P C version E.

(Refer Slide Time 20:21)



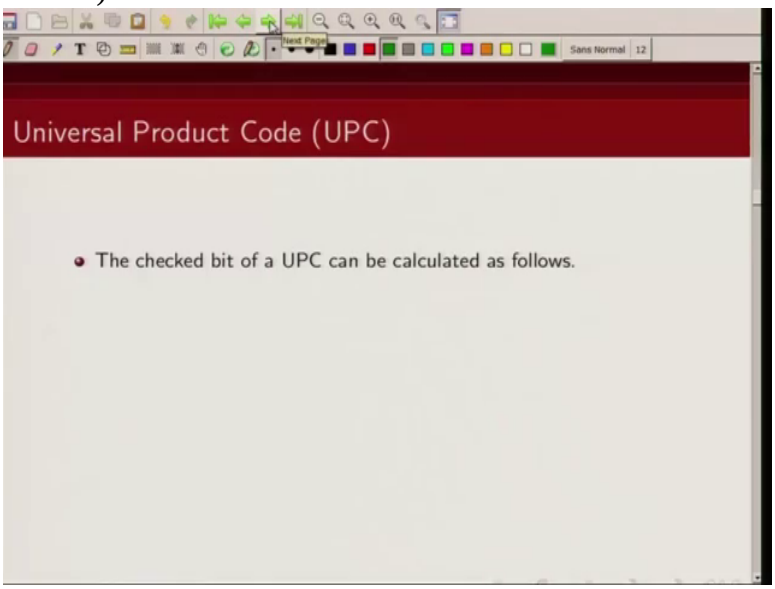
So a 12 digit U P C version has, consists of 12 digits. The first digit from the left is of U P C type, next 5 digits are manufacturer code, then next five digits are actually the product code which are assigned by the manufacturer and the final digit is the check bit. So that's what we are going to use for error detection. So how is

(Refer Slide Time 20:52)



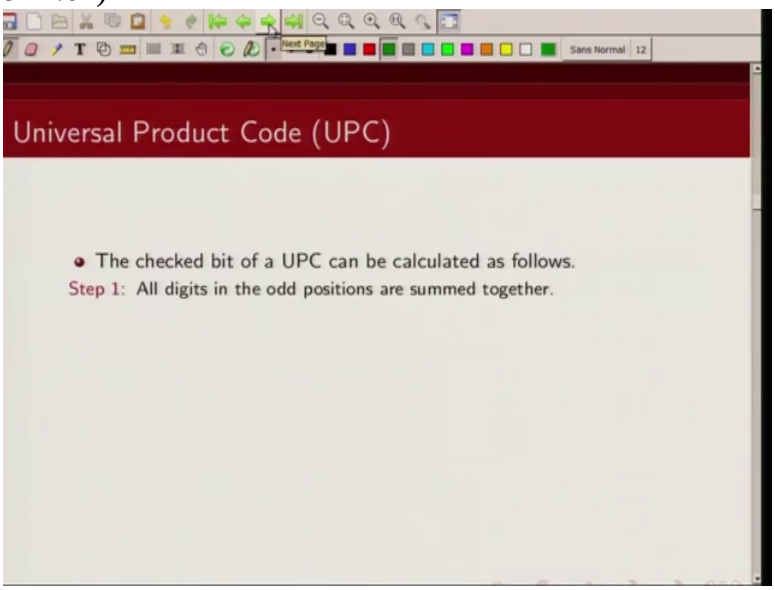
this check bit computed?

(Refer Slide Time 20:56)



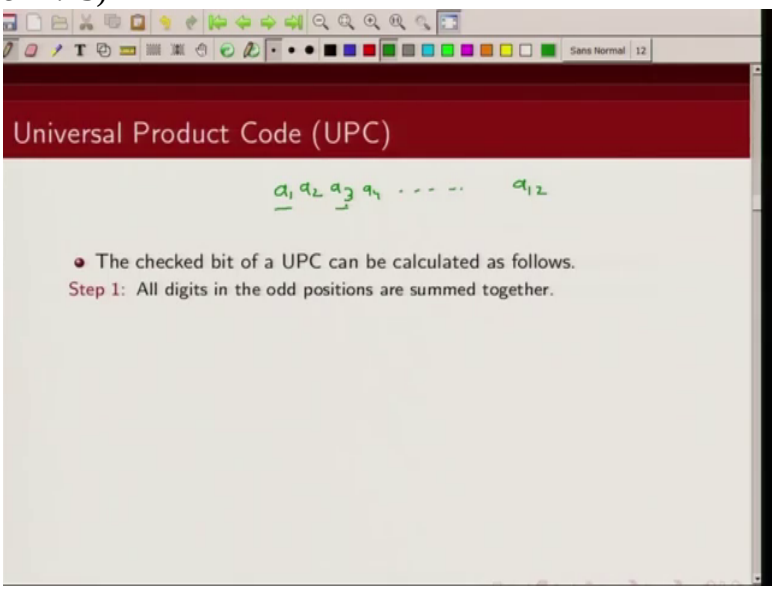
So this check bit is computed as follows.

(Refer Slide Time 21:01)



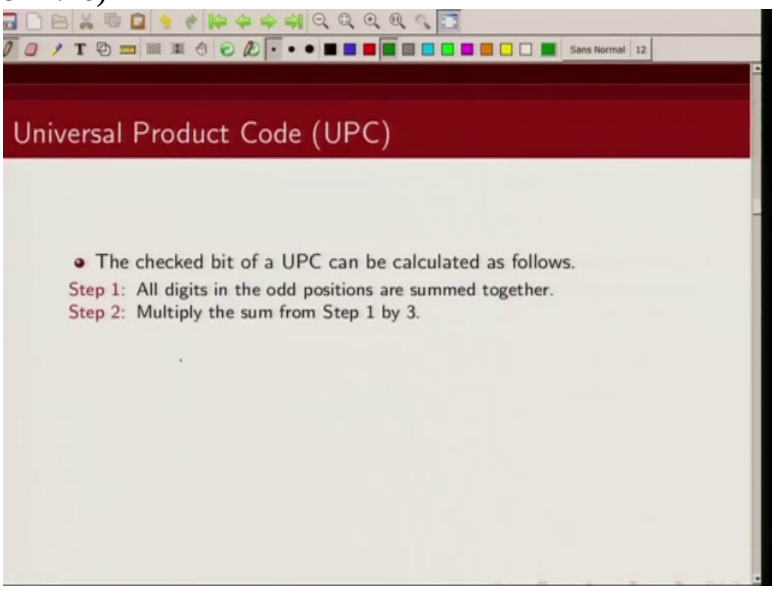
All digits in the odd position are summed up. So you have U P C number which is a 1, a 2, a 3, a 4 up to a 12 then what you are going to do is you are going to add up all the odd, numbers in the odd location, that is

(Refer Slide Time 21:23)



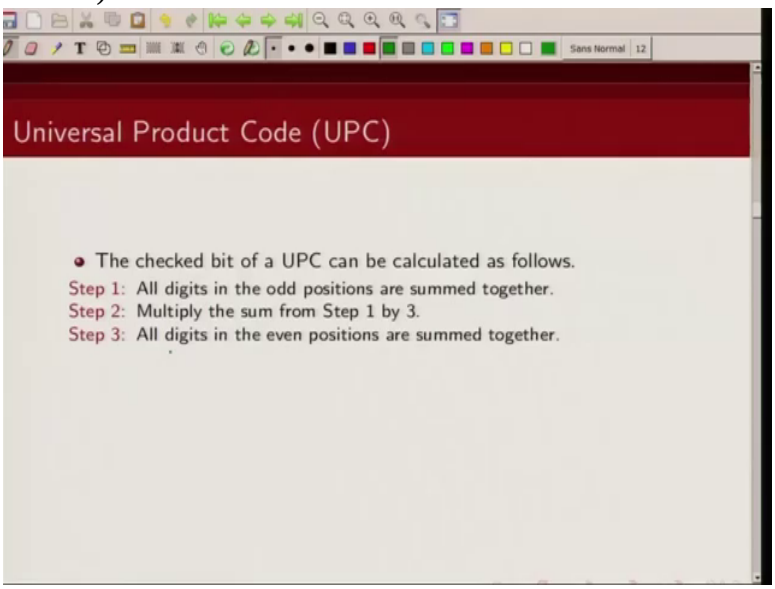
a 1, a 3, a 5 like that and

(Refer Slide Time 21:26)



multiply the sum by 3. So you are going to multiply the sum of the numbers of the odd location by 3 and to that you are going to add

(Refer Slide Time 21:39)

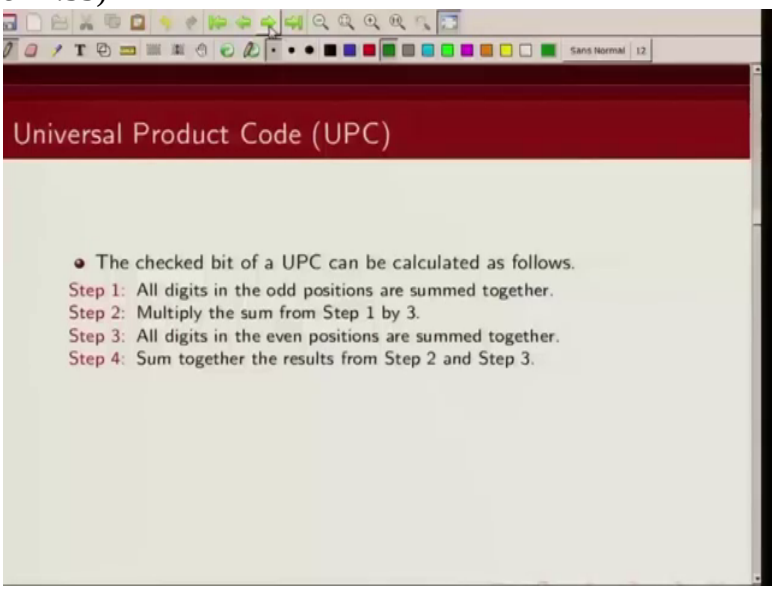


all the digits in the even position. So it will be three times a 1 plus a 3 plus a 5 and plus a 2, a 4, a 6

(Refer Slide Time 21:52)

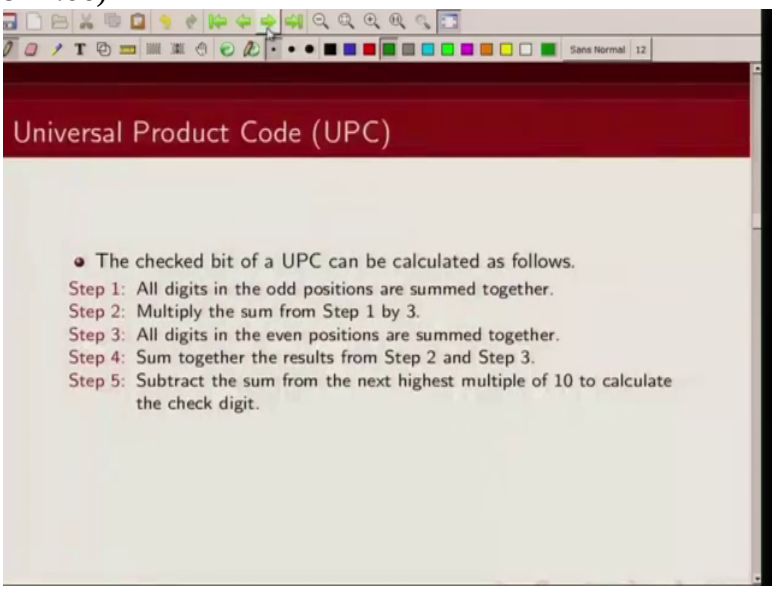
like that. So you are going to add

(Refer Slide Time 21:55)



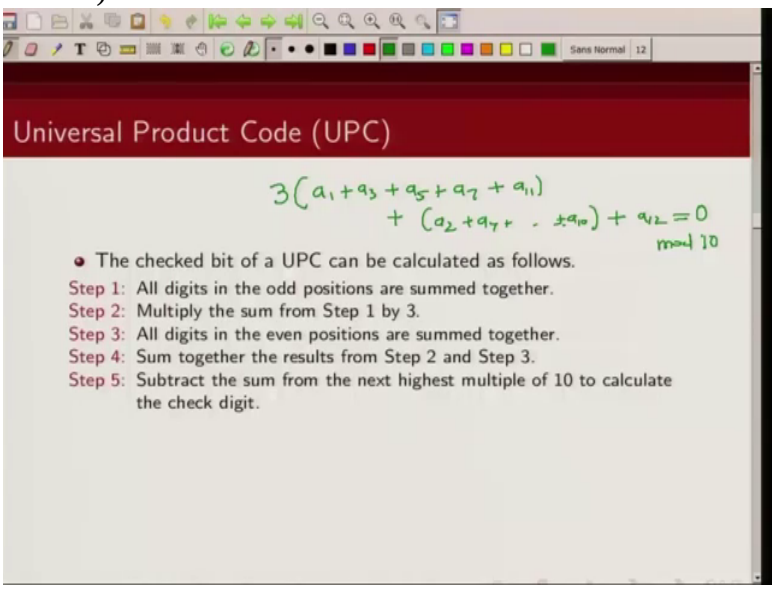
the number, so essentially you are weighing the numbers, multiplying by 3, 1 3, 1 like that and you are going to add them all up

(Refer Slide Time 22:06)



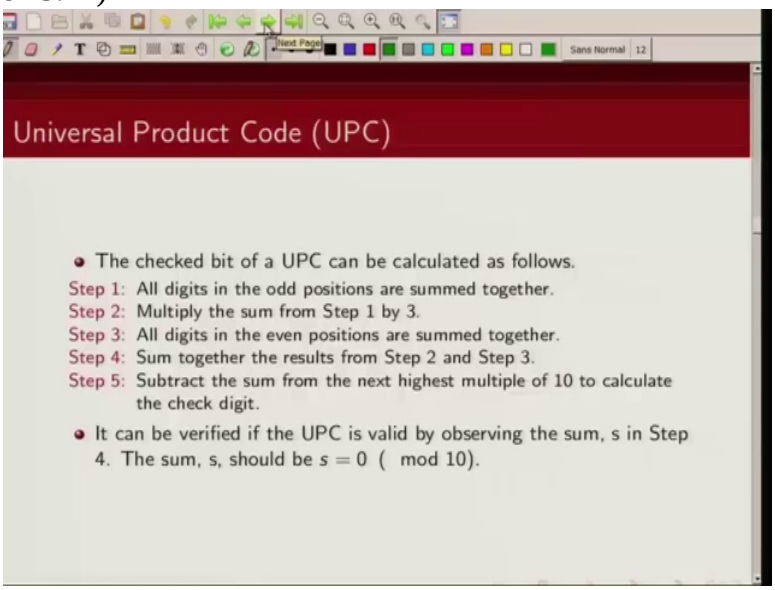
and subtract the sum that you get from the next highest multiple of 10. So what I am saying is, you have this three times a 1 plus a 3 plus a 5 plus a 7 plus a 11 plus you have this a 2 plus a 4, you have these, when you add them all up, plus your 12 bit number this is your a 12, so that should add up to 0 modulo 10, Ok. So the odd bit numbers

(Refer Slide Time 22:52)



are multiplied by 3, the even bit numbers are just added up, if you add those 11 bits, then the twelfth bit which is the check bit should be such that sum of this, weighted sum should be 0 modulo 10 and that's your U P C

(Refer Slide Time 23:12)

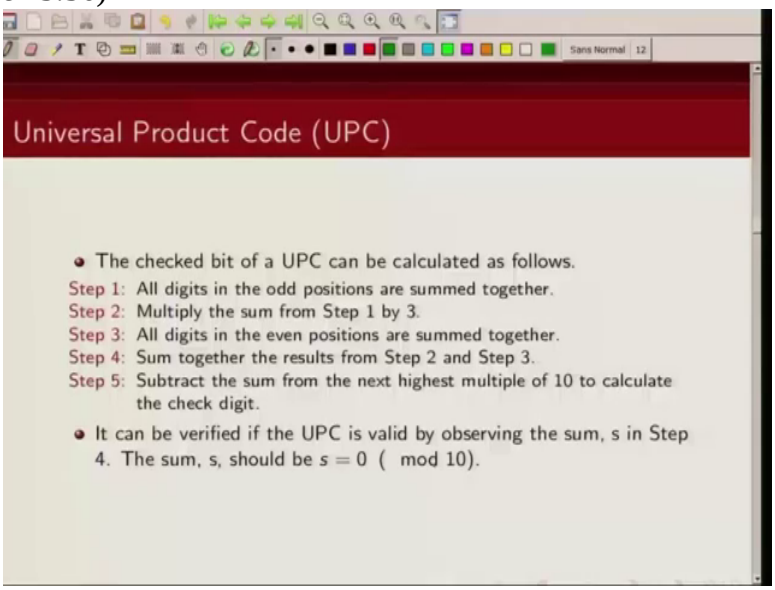


code. So the sum that you should get should be 0 modulo 10, sum of all the 12

(Refer Slide Time 23:20)

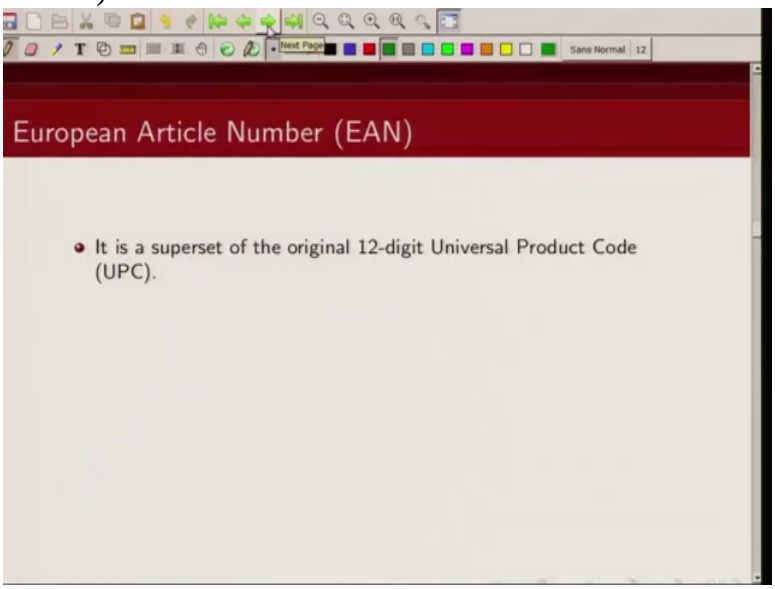
bit numbers, so three times a 1 plus a 2 plus three times a 3 plus a 4 up to three times a 11 plus a 12 should be 0 modulo 10 in case of U P C, 12 bit

(Refer Slide Time 23:36)



U P C code.

(Refer Slide Time 23:38)

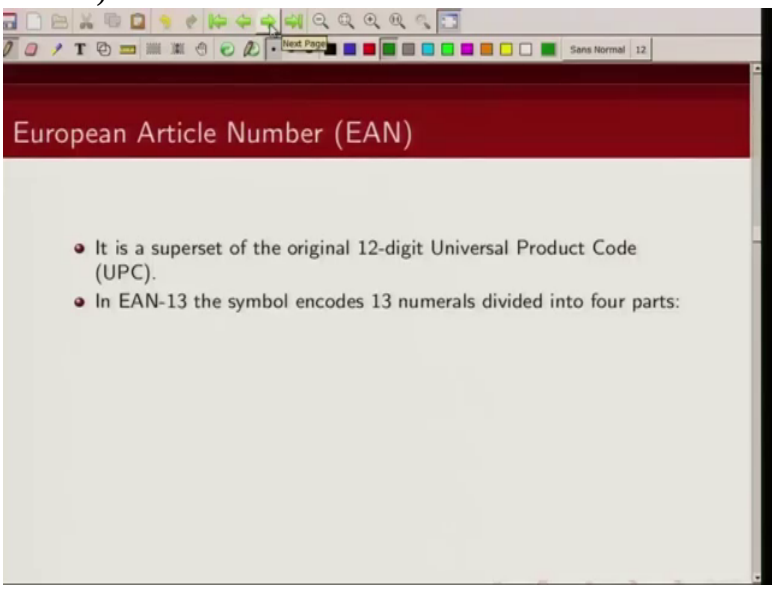


Now a superset of this U P C code, 12 bit U P C code is this 13 bit European

(Refer Slide Time 23:47)

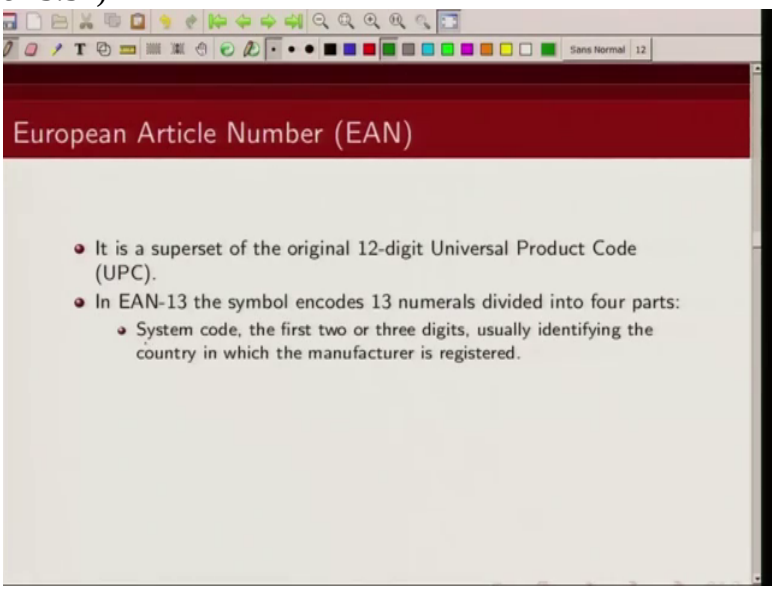
Article Number. So what is this

(Refer Slide Time 23:51)



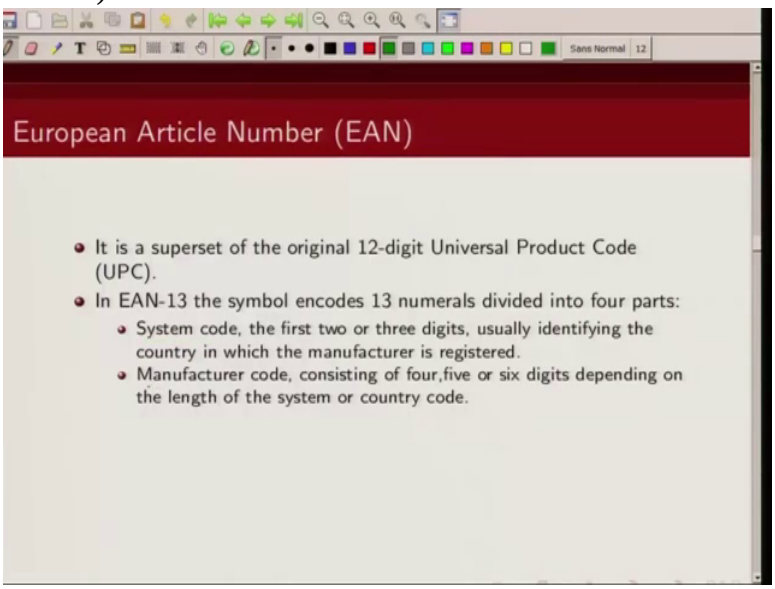
13 bit European Article Number? So it has 4 parts. So first

(Refer Slide Time 23:57)



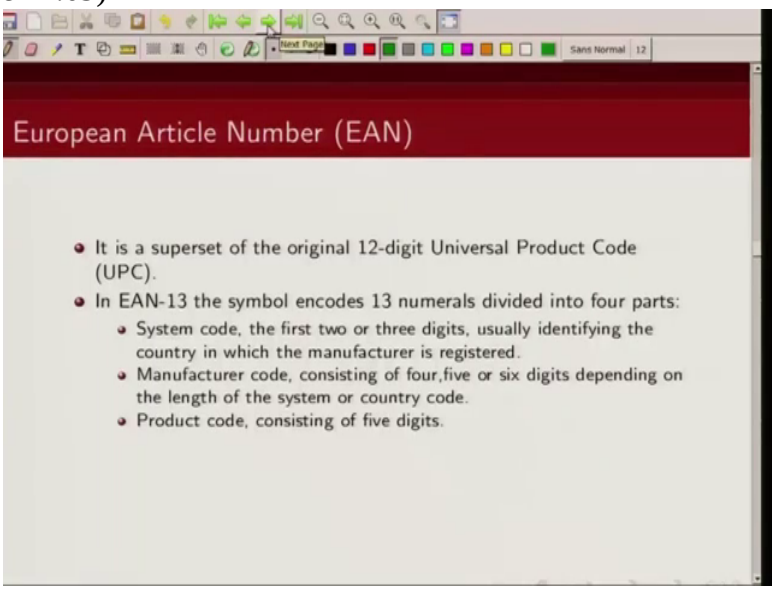
is system code which is first 2 or 3 bits which identify the country where the product has been manufactured. Then you

(Refer Slide Time 24:05)



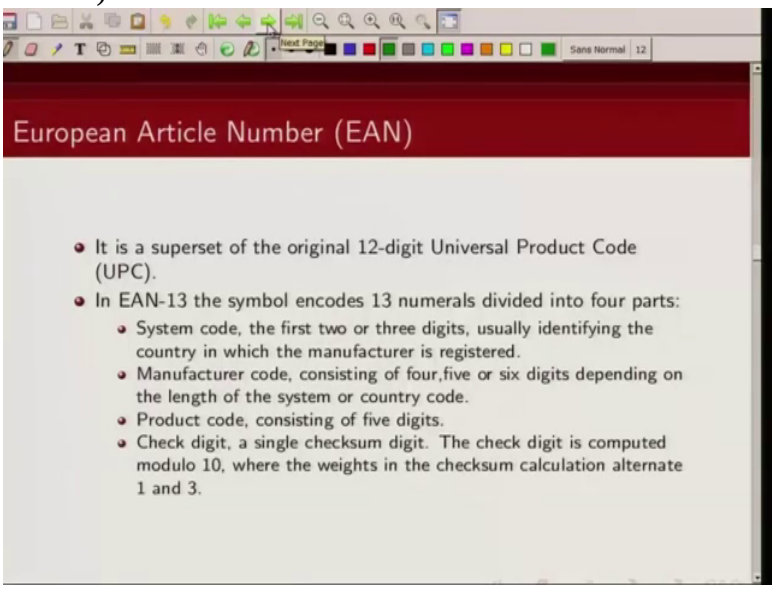
have manufacturer code which is 4 to 6 bits

(Refer Slide Time 24:09)



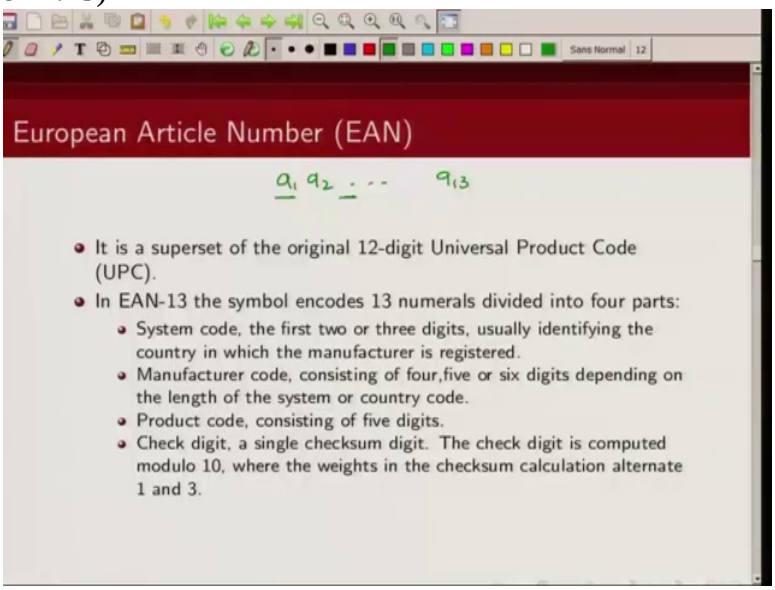
and then you have the product code which consists of 5 bits and finally

(Refer Slide Time 24:13)



you have this check bit and this check bit is computed modulo 10 where the weights in the checksum calculation alternate between 1 and 3. Now what do I mean by that? 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

(Refer Slide Time 24:43)



and bits in the even location, you multiply by 3 and the modulo sum of that should add up to 0 mod 10. So let us look at an example to illustrate that. Ok, so let us look at an example to illustrate that. So I have this Britannia

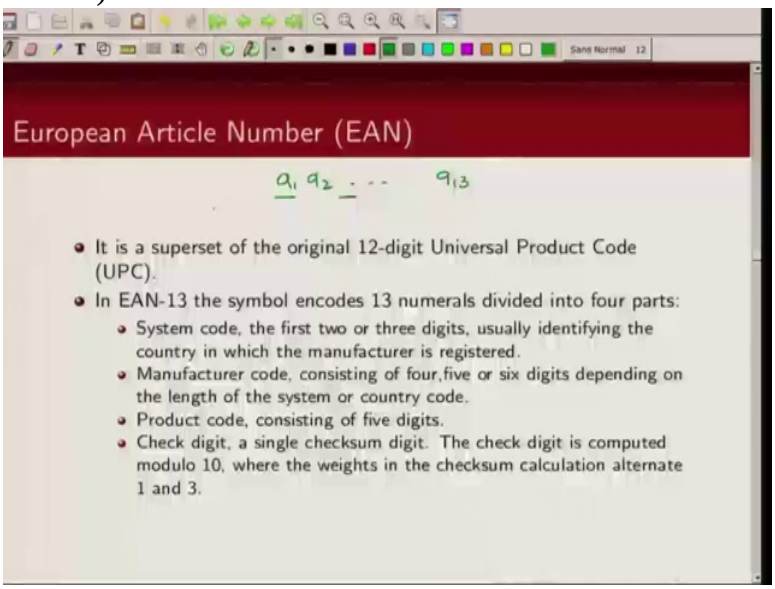
(Refer Slide Time 25:05)

Bourbon biscuit, right and what is

(Refer Slide Time 25:10)

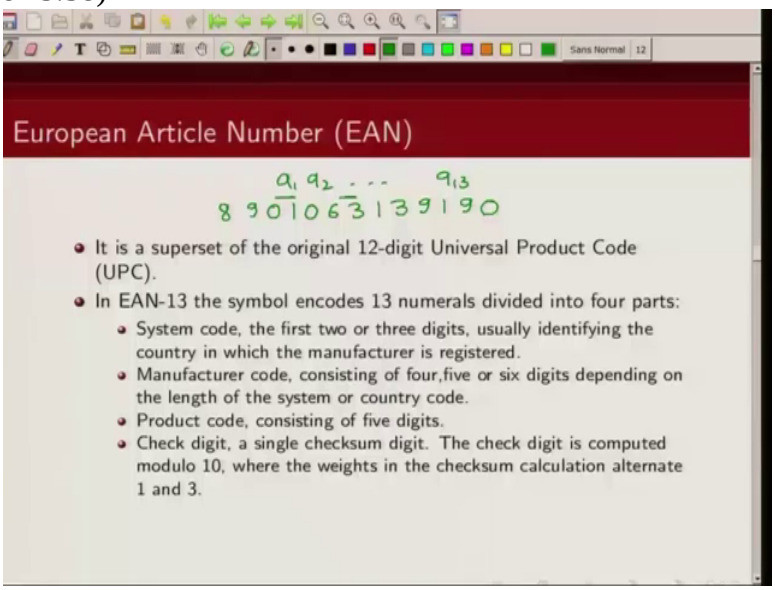
this European Article Number for this? European Article Number for this is

(Refer Slide Time 25:15)



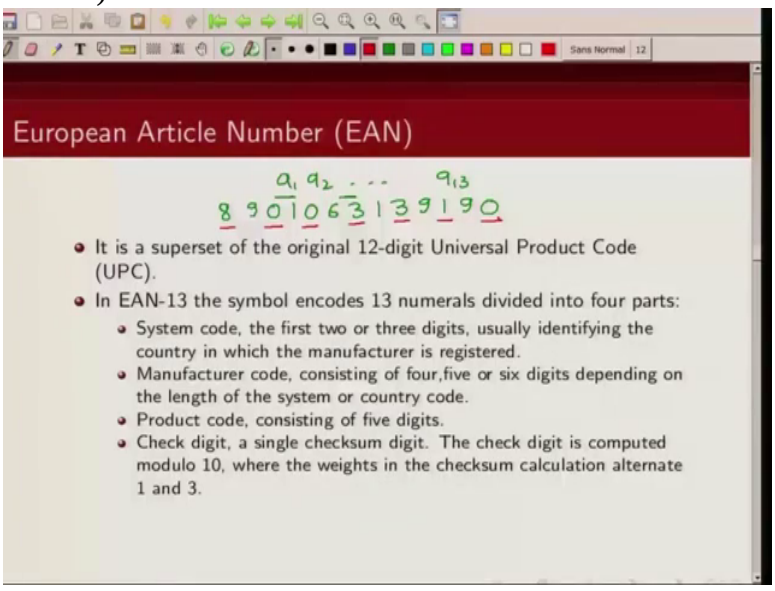
8 9 0 1 0 6 3 1 3 9 1 9 0. So this is my 13 bit, you can see, 1,2,3,4, 5, 6, 7 8, 9, 10, 11, 12, 13, this is 13 bit

(Refer Slide Time 25:38)



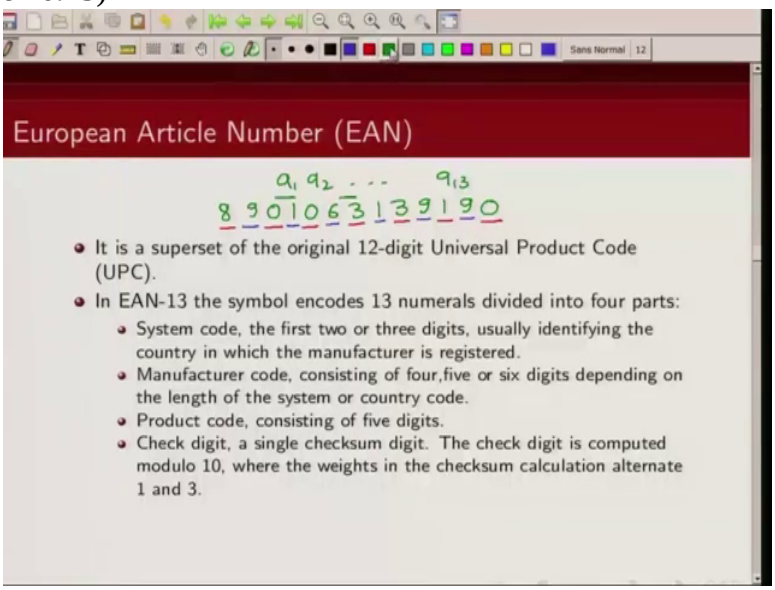
European Article numbers. Now let's see whether this checksum bit is correct. So what do we need to do? We need to add odd bits and we need to add the even bits, bits in the even location and multiply them by three. So let's first mark the bits in the odd location. This is 0, 3, 3, 1, 0, Ok

(Refer Slide Time 26:13)



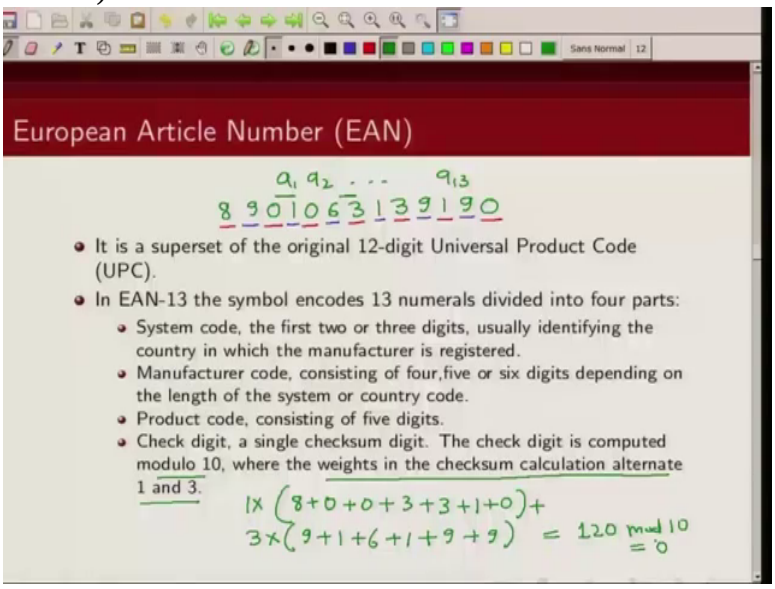
and what are the bits in the even location? That's marked with this blue pen, Ok.

(Refer Slide Time 26:25)



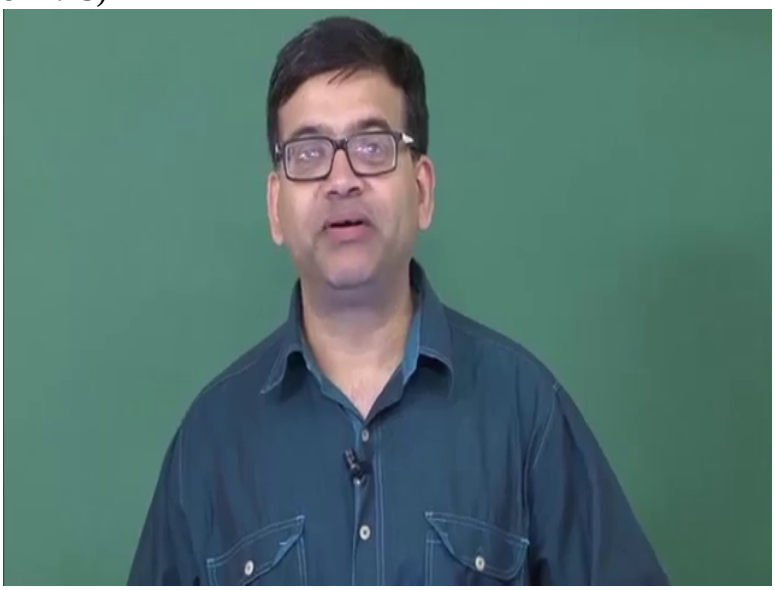
So let's add up, so again the check bit is computed modulo 10 where weights in the checksum calculation alternate between 1 and 3. So what do I mean by that? So every odd bit I multiply by 1 and every even bit

(Refer Slide Time 27:42)



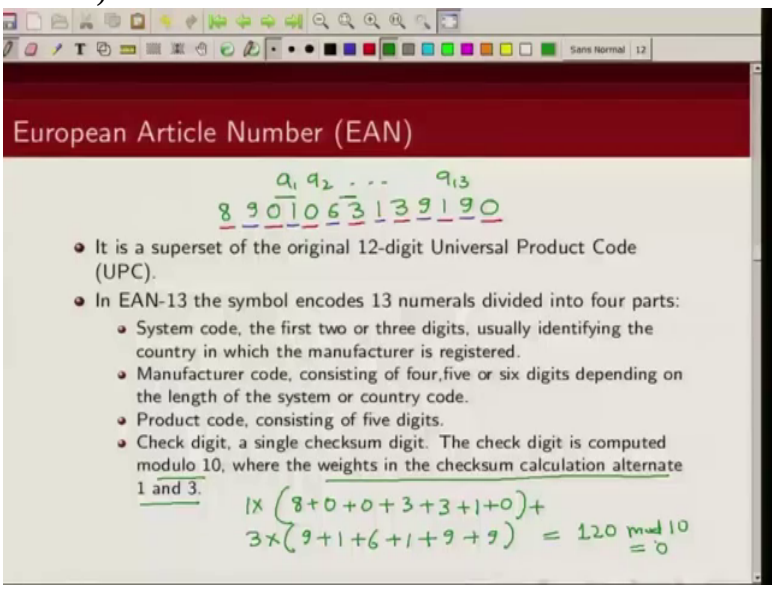
I, the bits in the even location I multiply by

(Refer Slide Time 27:45)



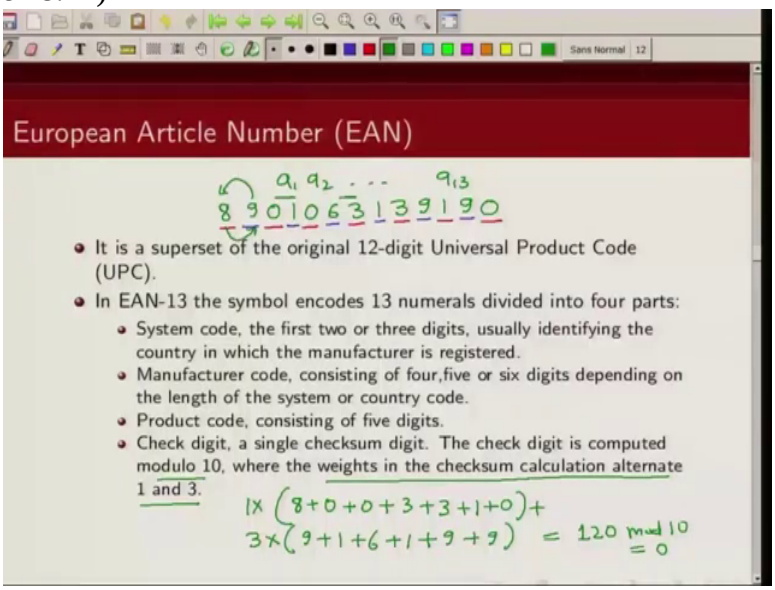
3. So let me first add up the numbers in the odd location, so that's 8 plus 0 plus 0 plus 3 plus 3 plus 1 plus 0. This we multiply by 1, plus 3 times numbers in the even location, that's 9 plus 1 plus 6 plus 1 plus 9 plus 9; 9, 1, 6, 1, 9, 9. And this comes out to be 120 and 120 mod 10 is 0. So you can see that the 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 detection. Now this code also

(Refer Slide Time 28:04)



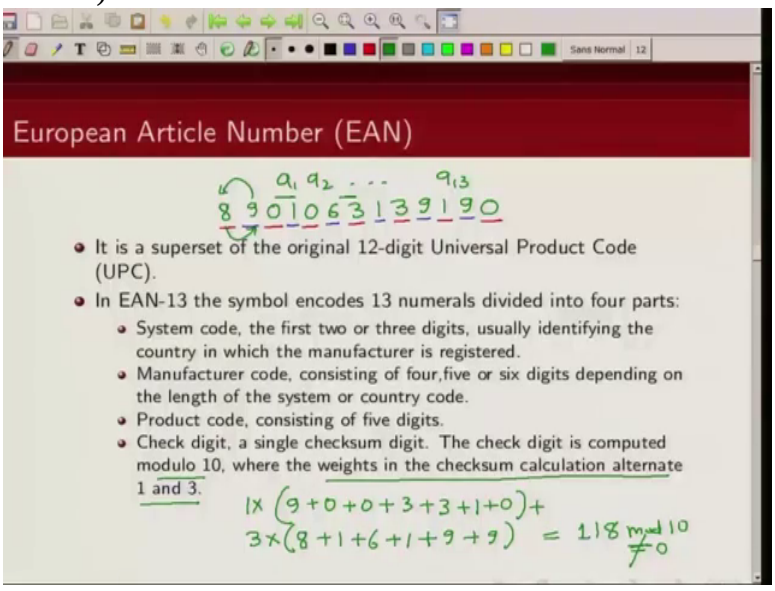
can easily find out the transposition error. For example if these two bits would have got exchanged then what

(Refer Slide Time 28:14)



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. This sum would have changed. So the earlier it was nine times 3 27, 3 less and there is 1 more, so now 2 less. So this would have become 118 and this is not

(Refer Slide Time 28:45)



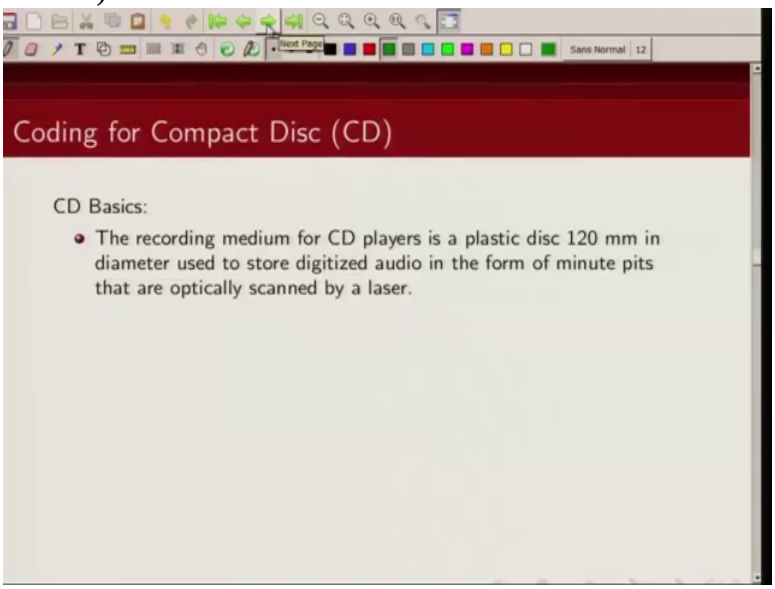
0 modulo 10, so you can see, this code can correct, can detect transposition error. And these are the most common error which happen when you

(Refer Slide Time 28:56)

are trying to read these bar codes, Ok.

Now let us now move to some

(Refer Slide Time 29:03)

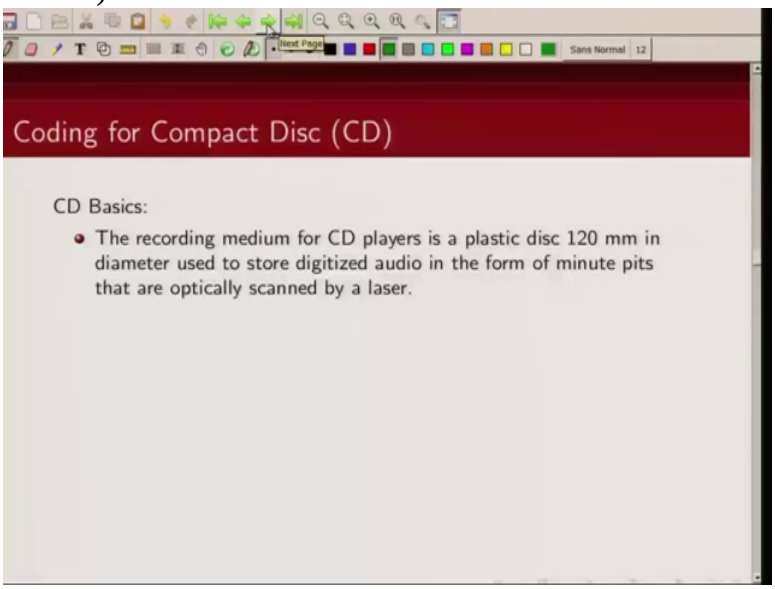


other applications. So we start with

(Refer Slide Time 29:06)

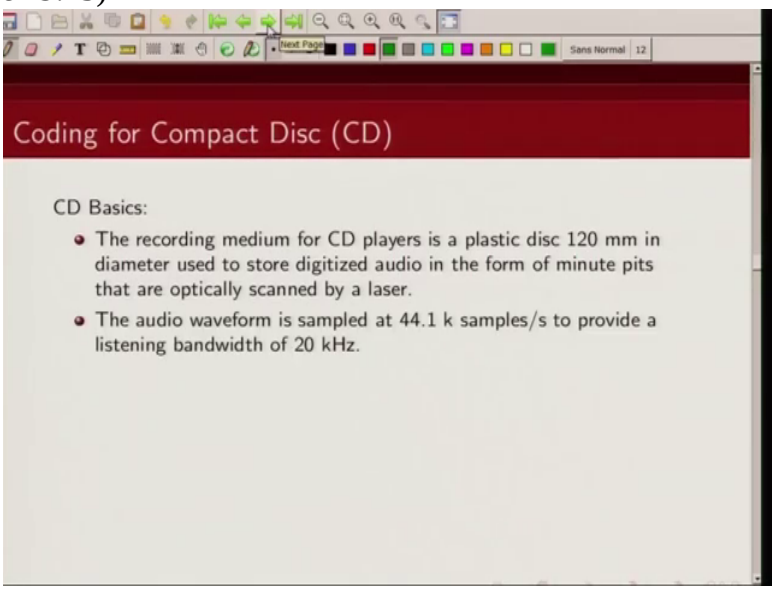
application of error control coding in C Ds. You are familiar with the C D disk. It is basically

(Refer Slide Time 29:12)



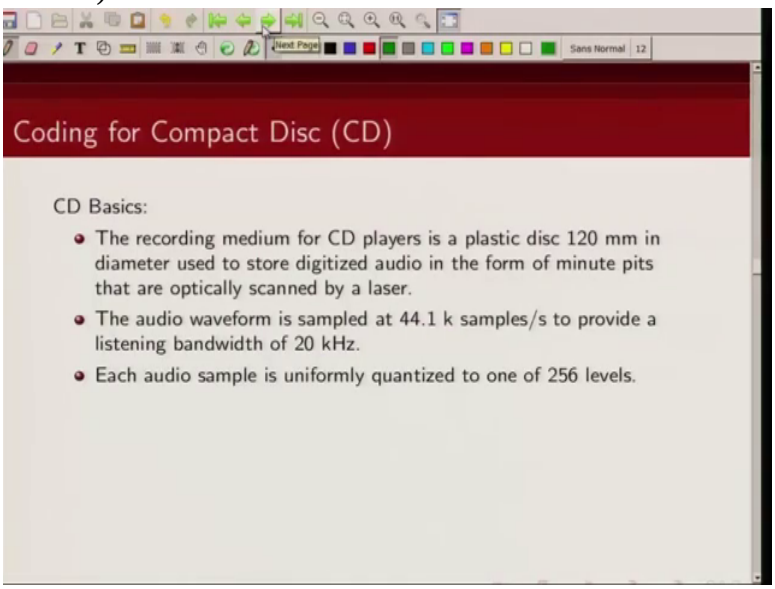
a plastic disks of 120 m m in diameter and it can be used for storing digital audio, video data. Now

(Refer Slide Time 29:23)



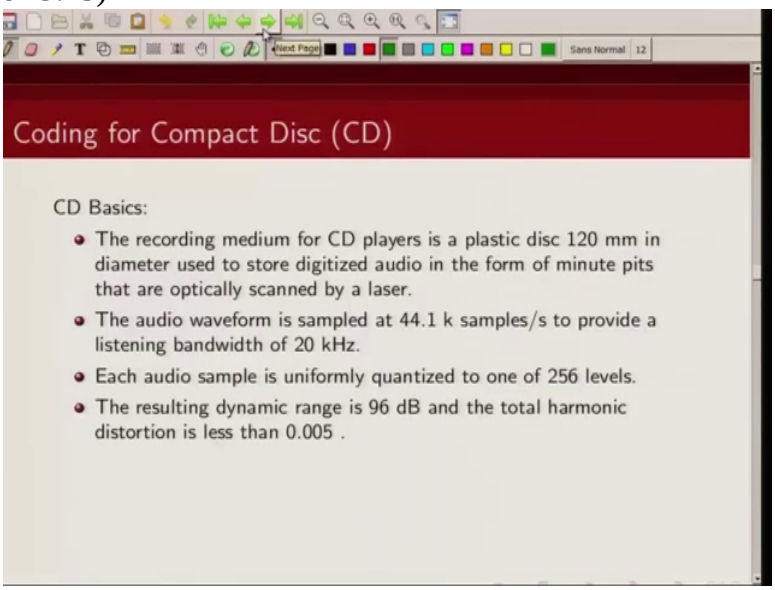
the audio signal is sampled at 41 point 1 kilo samples per second. And it provides a listening bandwidth of 20 kilo Hertz.

(Refer Slide Time 29:34)



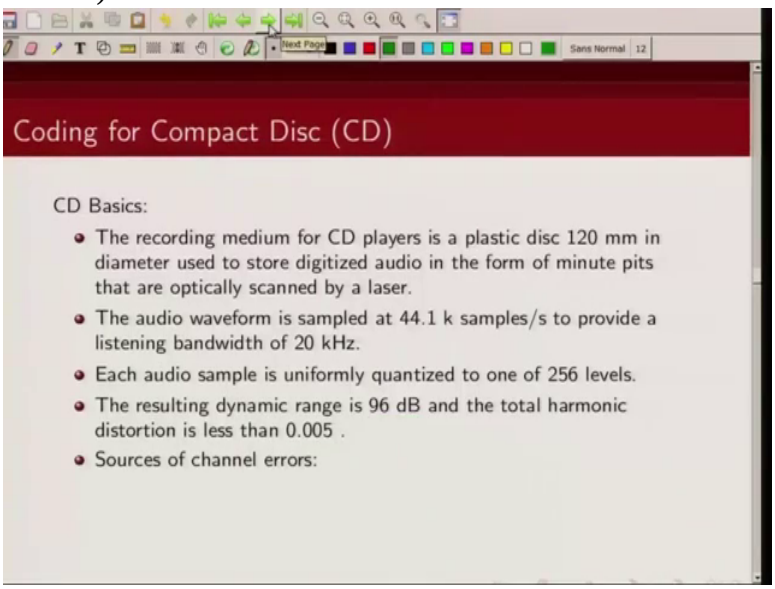
Each audio sample is sampled to, quantized to one of the 256 levels which

(Refer Slide Time 29:43)



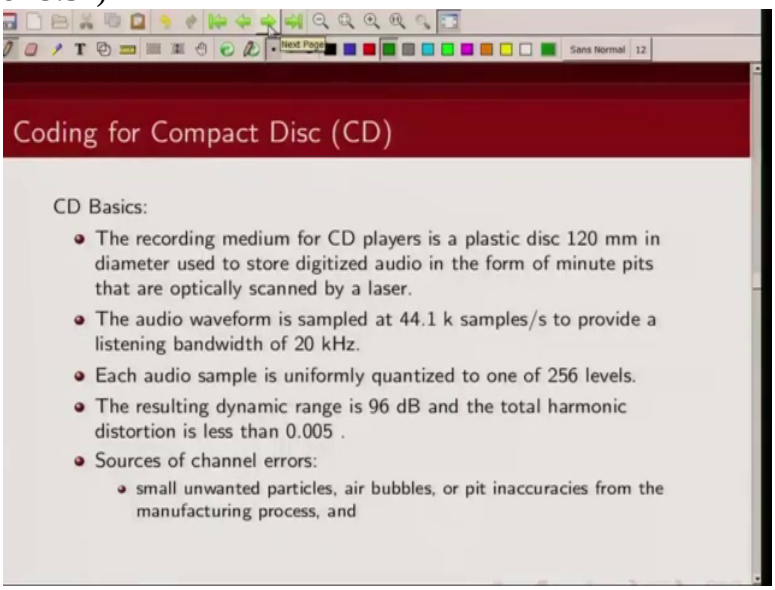
provides a dynamic range of 96 d B and harmonic distortion of less than point 0 0 5. Now what

(Refer Slide Time 29:52)



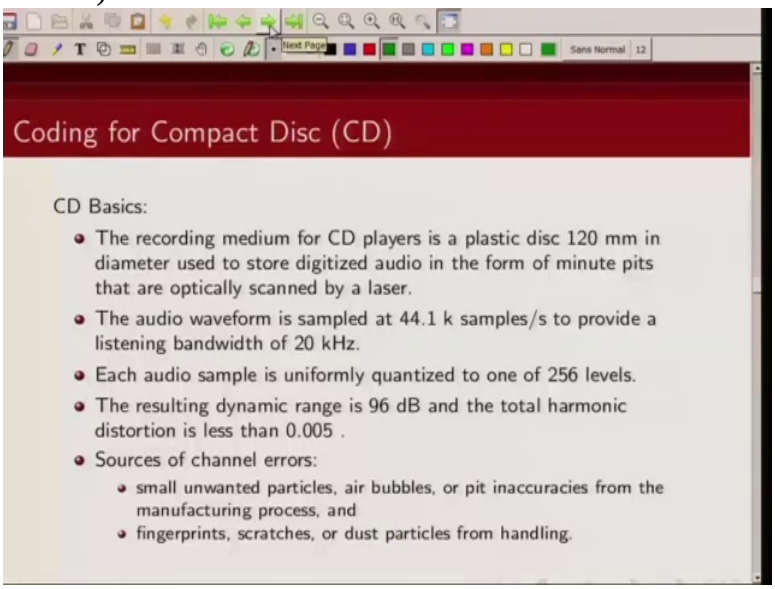
are the sources of error in a C D disk?

(Refer Slide Time 29:57)



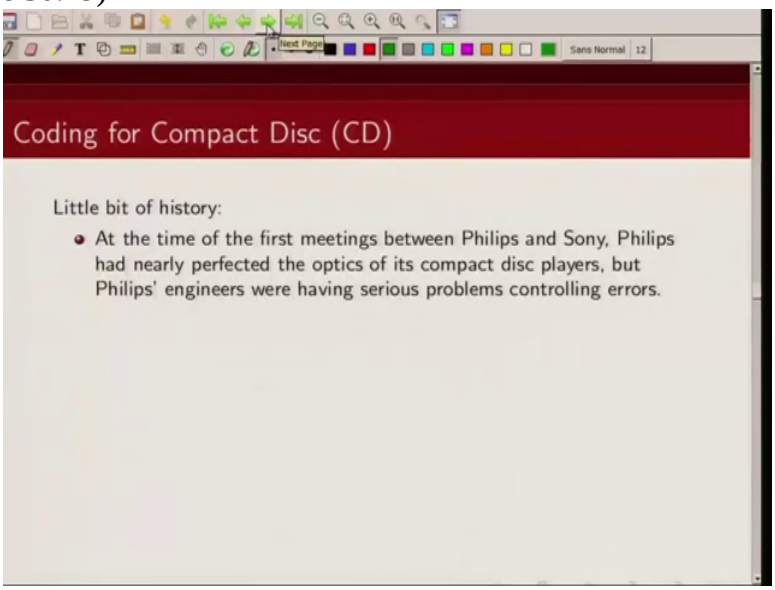
So there could be small unwanted particles, dust particles or there could be some air bubbles or things like that which could have happened during manufacturing of these C Ds, or

(Refer Slide Time 30:09)



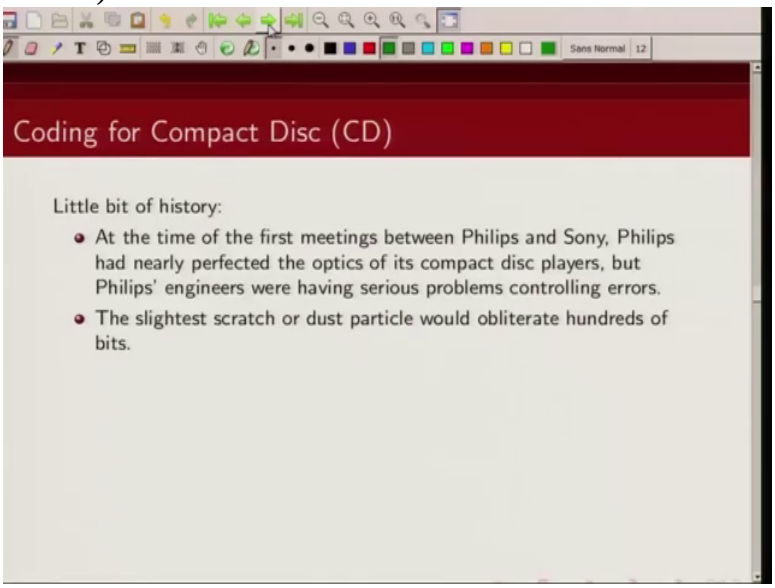
fingerprints, scratches or dust particles from handling the C D. So how does C D corrects all these kinds of errors? You must have noticed, even if your CD has small scratches but still your C D plays. How does this happen? This happens because

(Refer Slide Time 30:29)



of a very powerful error control coding which is used in the C Ds. So little bit of history, basically Philips and Sony were experimenting with this optical disk and Philips engineers had lot of problems controlling errors

(Refer Slide Time 30:45)

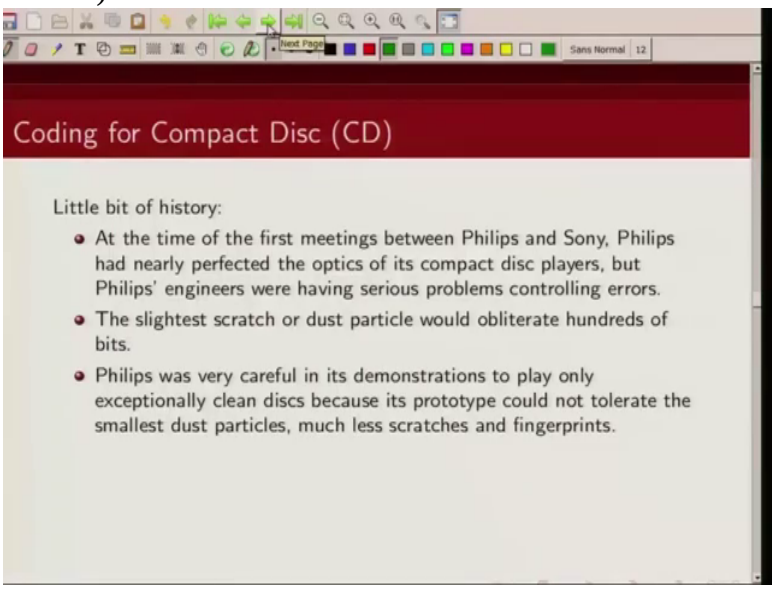


in the disk and even if there would have been small scratches

(Refer Slide Time 30:49)

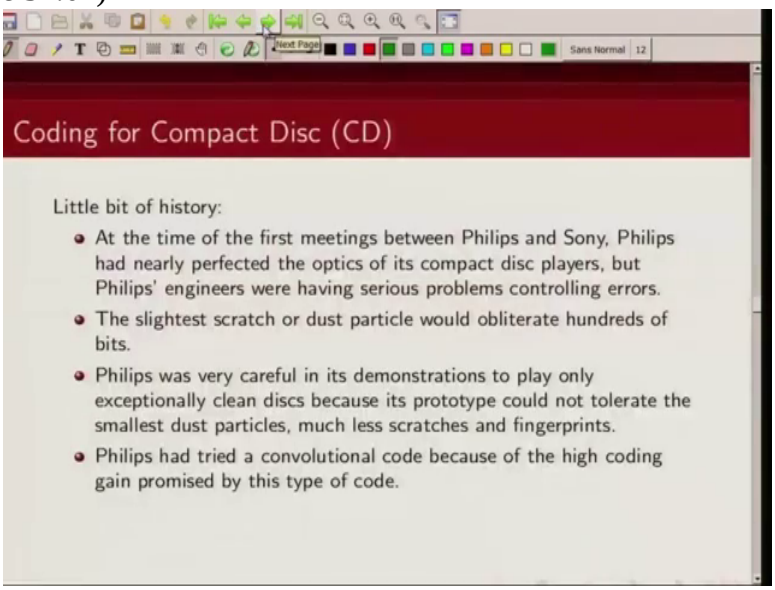
or dust particles, the whole data would essentially, you know, go away. So

(Refer Slide Time 30:55)



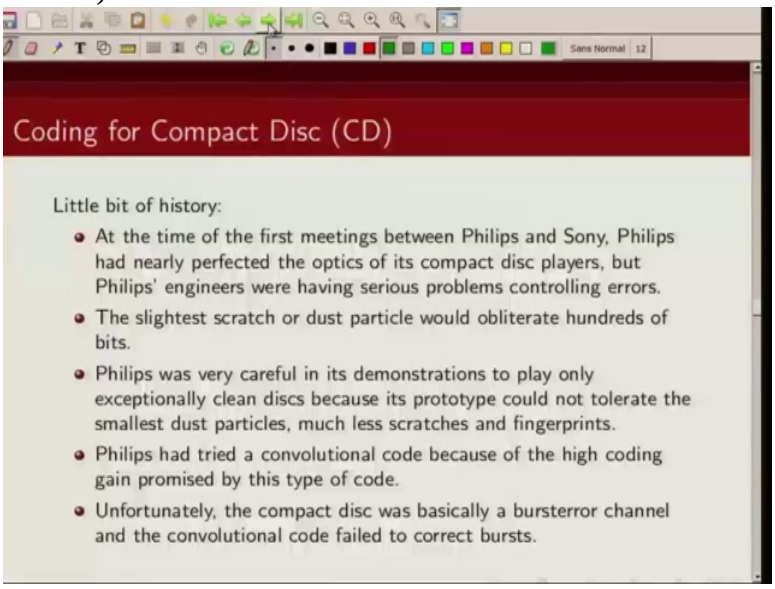
whenever Philips were demonstrating the C D, they were very careful. They were just demonstrating with very, very clean C Ds and which did not have any scratches and things like that. And what

(Refer Slide Time 31:07)



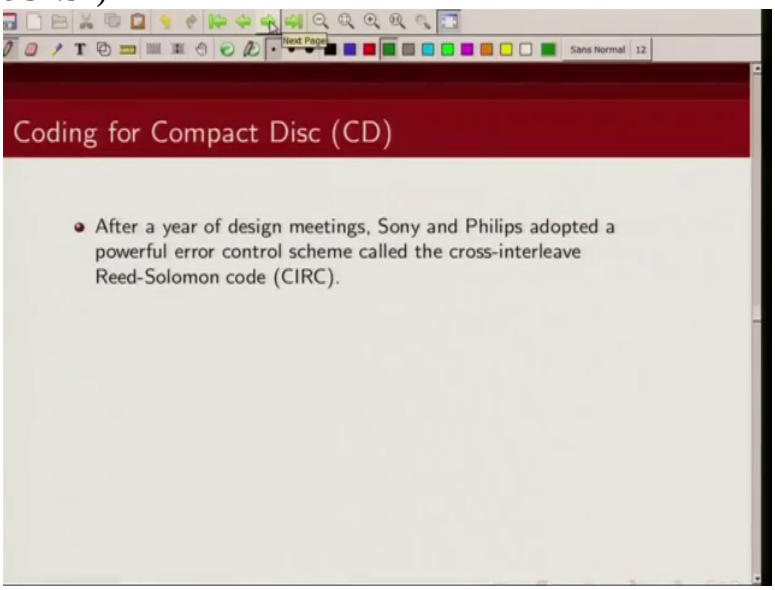
Philips was actually doing was, they were trying a convolutional code to correct errors because of high coding gain provided by that. But the problem is that this storage channel is like an erasure channel. If there is scratch or something like that, there are neighboring bits which get affected and the data there is erased. So convolutional codes are not good in correcting erasures and that's why they were initially not successful. So as I said,

(Refer Slide Time 31:39)



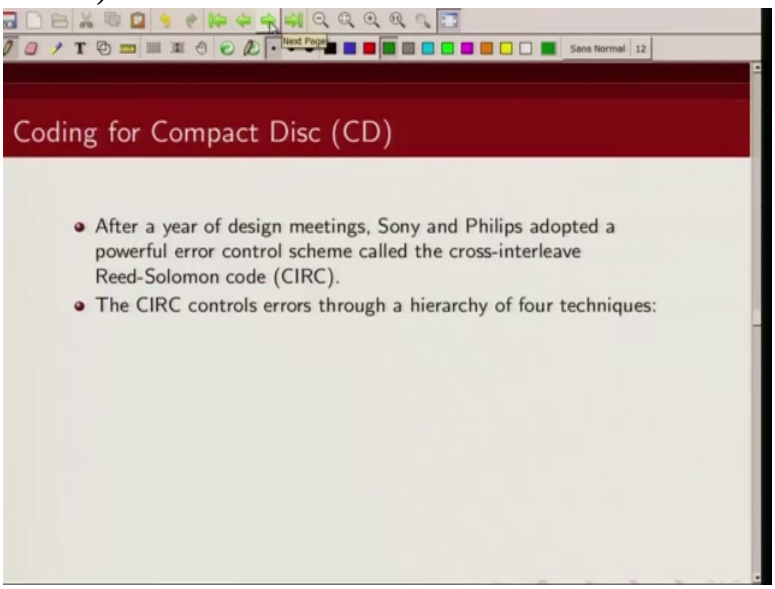
the problem was this storage media was burst error correct, burst channel and convolutional code was not good in correcting burst error. So there comes

(Refer Slide Time 31:52)



Reed Solomon code and so the C Ds used, a basically a cross interleaved Reed Solomon codes

(Refer Slide Time 32:03)

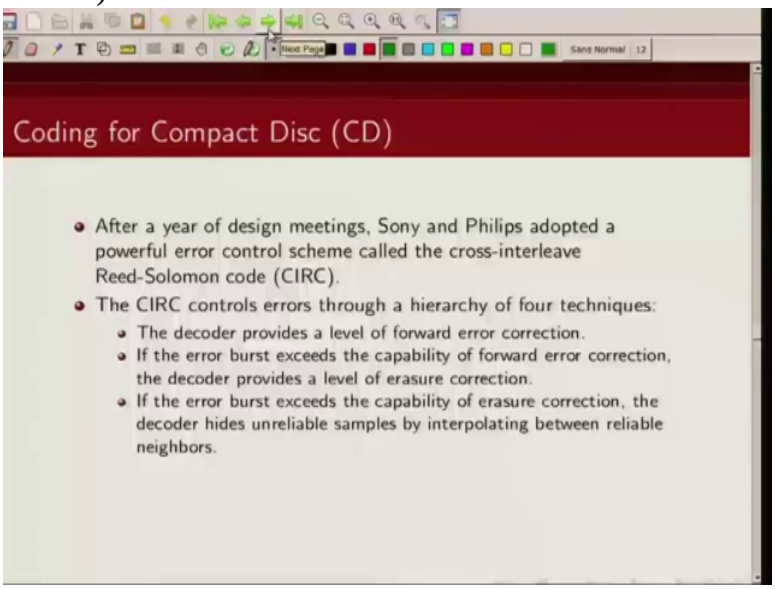


and there is four set of hierarchy as far as error control and detection is concerned. So there is an inherent forward error correction because of these Reed Solomon code but if the error burst exceeds the capability of the error correcting capability of the code, then the decoder also provides a level of erasure correction. So there is a, in addition to error correction there is also an erasure correction. An erasure is, you know a bunch of neighboring bits just getting erased and not, just not been able to recover. So the decoder has this capability that if the error correction fails, at

(Refer Slide Time 32:43)

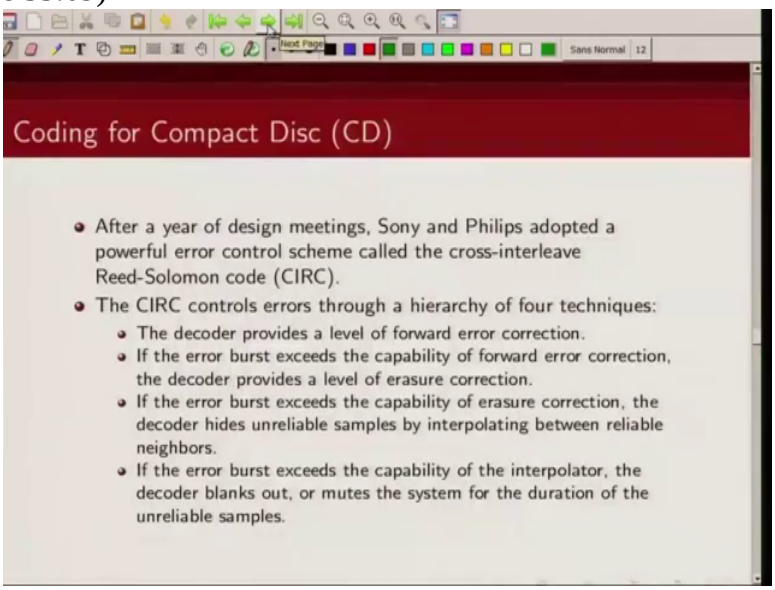
second level the decoder can do erasure correction and try to recover those data.

(Refer Slide Time 32:50)



Third is, if the error burst exceeds the capability of erasure correction then what it does is it tries to do some sort of interpolation between reliable neighbors so as to interpolate what the data is and if even

(Refer Slide Time 33:09)

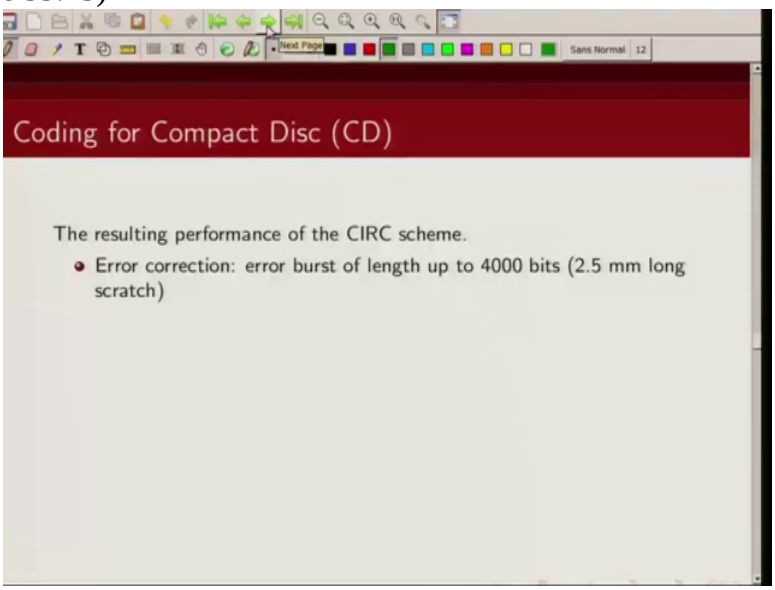


if interpolation fails, then basically the decoder just blanks out and you will hear [tuk/ short beep] like kind of a, you know mute sound when

(Refer Slide Time 33:18)

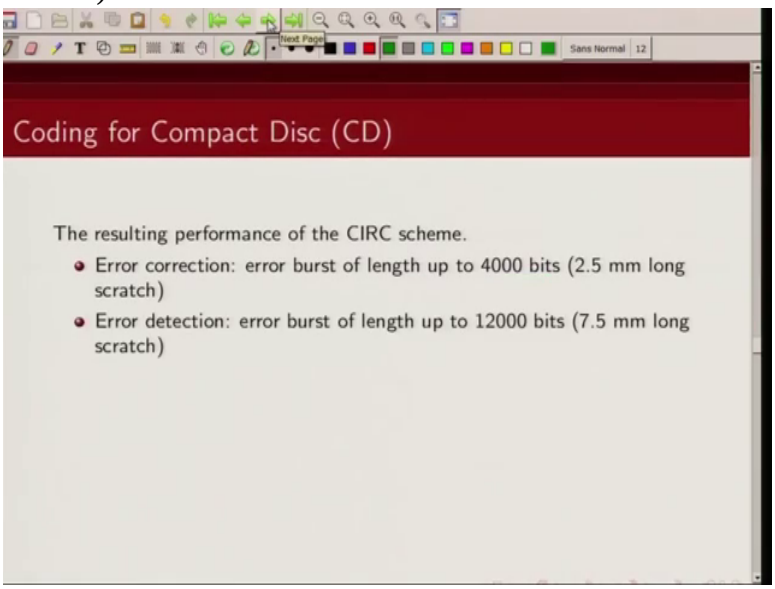
the error is big enough then even interpolation cannot work.

(Refer Slide Time 33:23)



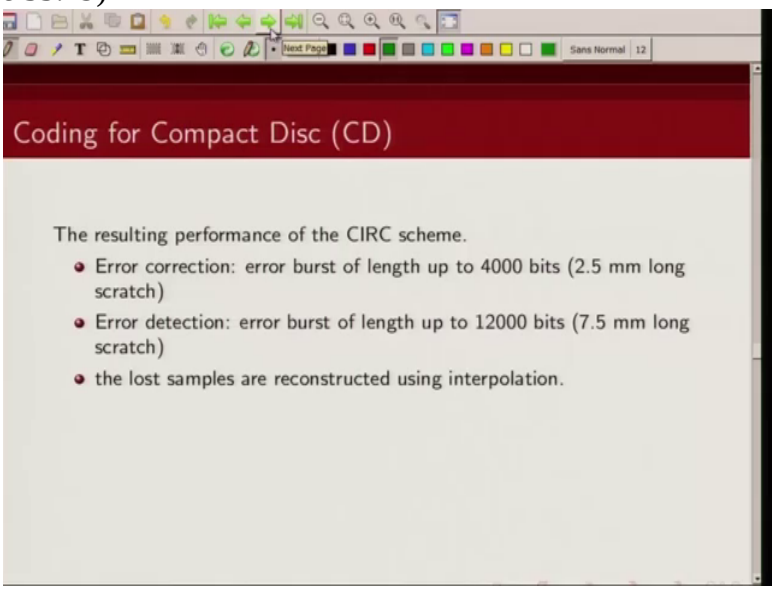
So a C D disk can correct error burst up to 4000 bits. So try putting a scratch of 2 point 5 mm, your C D will still work, I can guarantee that.

(Refer Slide Time 33:38)



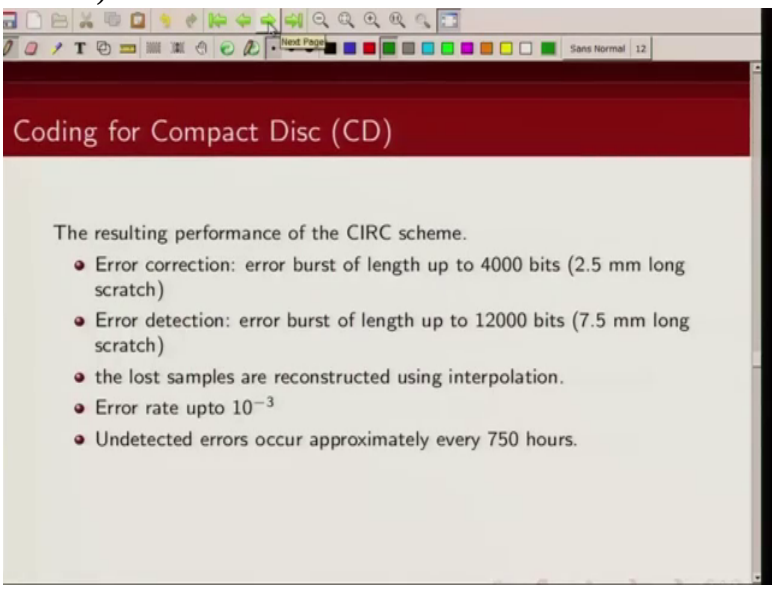
It can detect error up to 12000 bits so it can detect errors of up to 7 point 5 m m long scratch.

(Refer Slide Time 33:49)



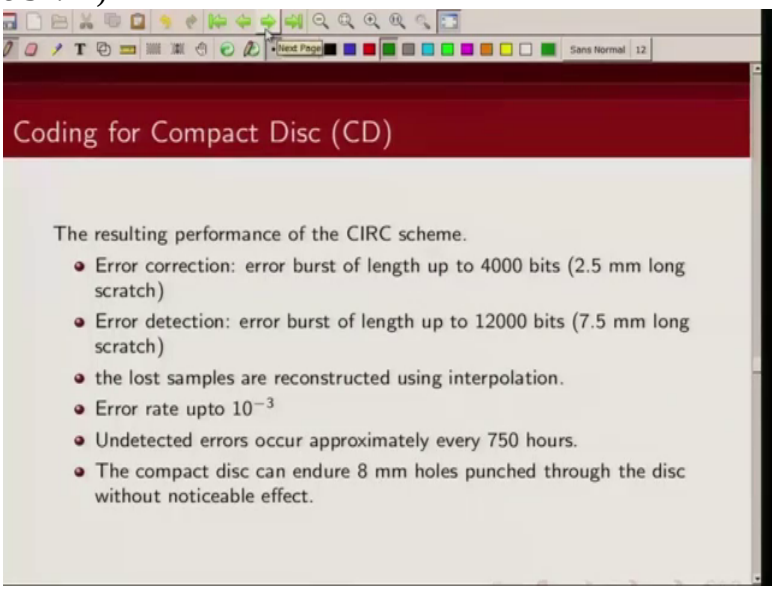
And of course as I said, if the error burst exceeds the capability of erasure correction then it tries to do interpolation to get back those bits.

(Refer Slide Time 34:00)



And undetected error happens very rarely. It happens every like 750 hours or so. So there is a very powerful error correcting code which are used in C D.

(Refer Slide Time 34:14)



And interestingly even compact discs can withstand small punch holes in it. You can try putting a 8 m m hole into your C D and still it would be able to, you know correct the errors in the data and

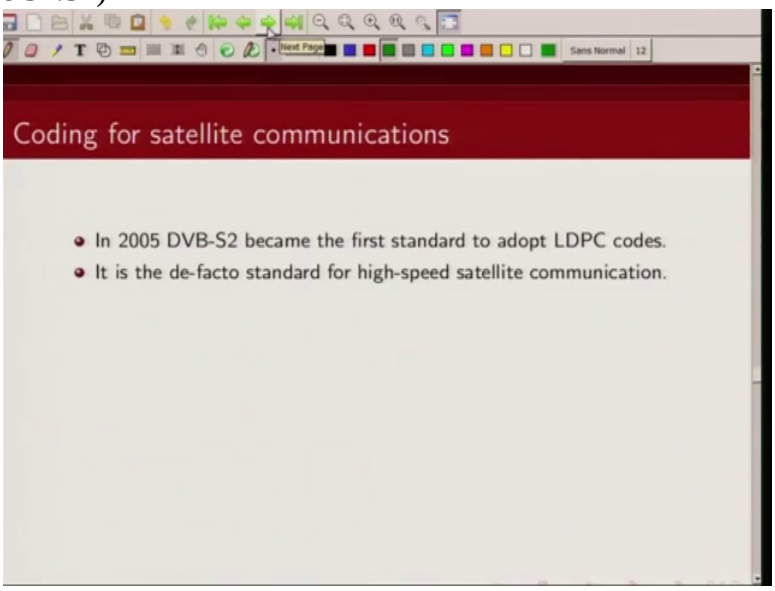
(Refer Slide Time 34:33)

C D will still work.

(Refer Slide Time 34:34)

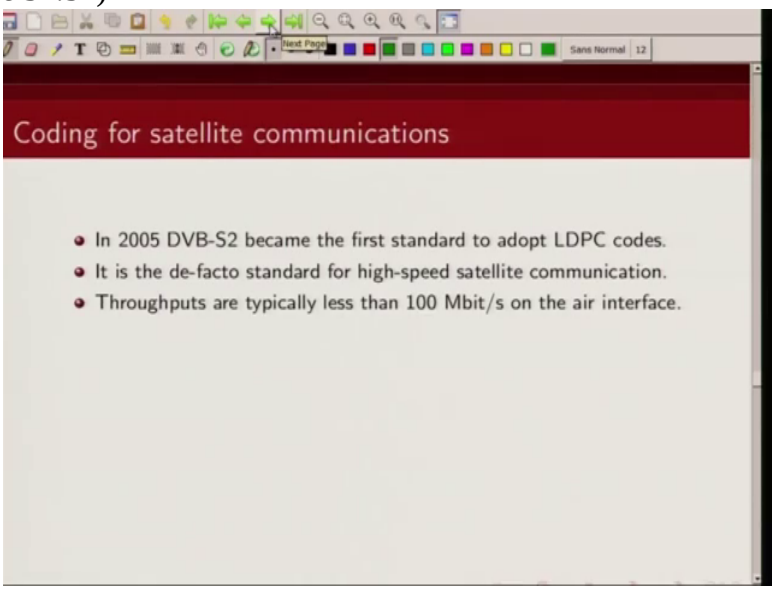
Now let's move to application for, in satellite communication. So there is the standard video digital video broadcasting S 2 standard which was one of the first standards to adopt low density parity check codes as their standard so it is a de facto

(Refer Slide Time 34:52)



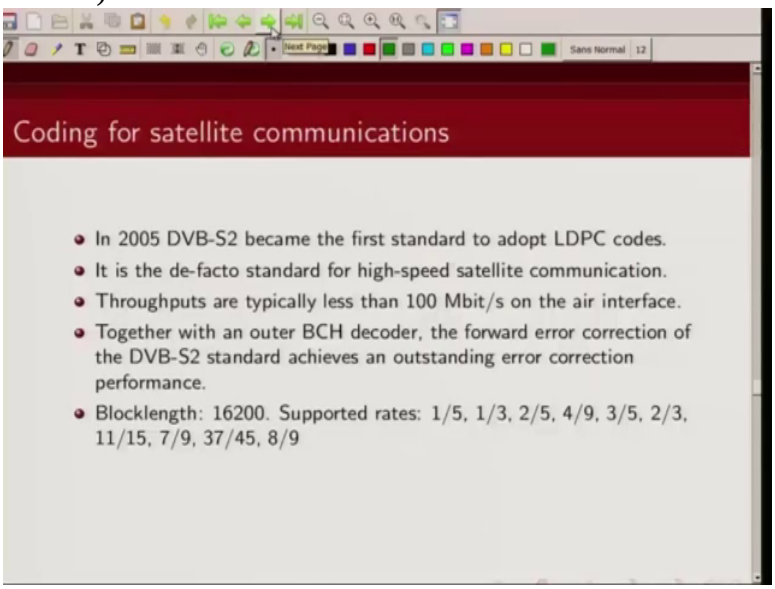
standard for high speed satellite communication

(Refer Slide Time 34:57)



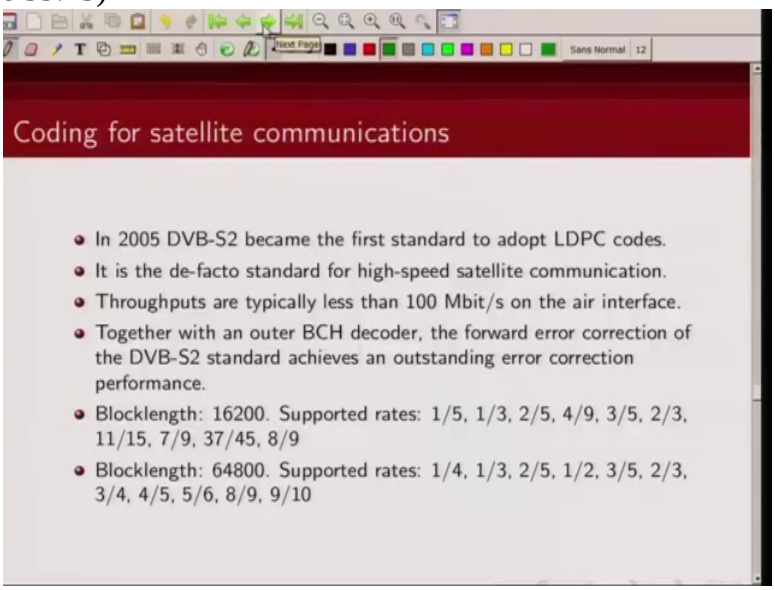
where typical throughput is less than 100 mega bits per second. Now

(Refer Slide Time 35:03)



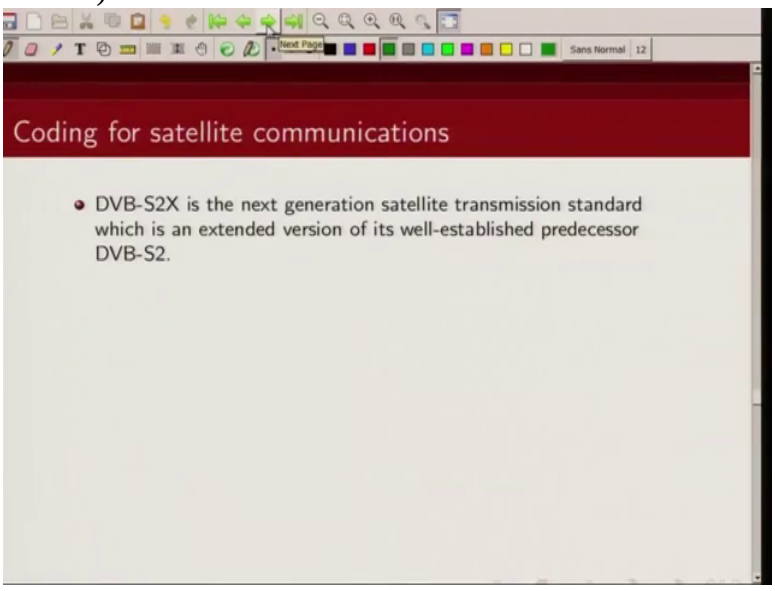
it uses, as an outer code B C H code along with the inner L D P C codes and it support variety of block sizes.

(Refer Slide Time 35:13)



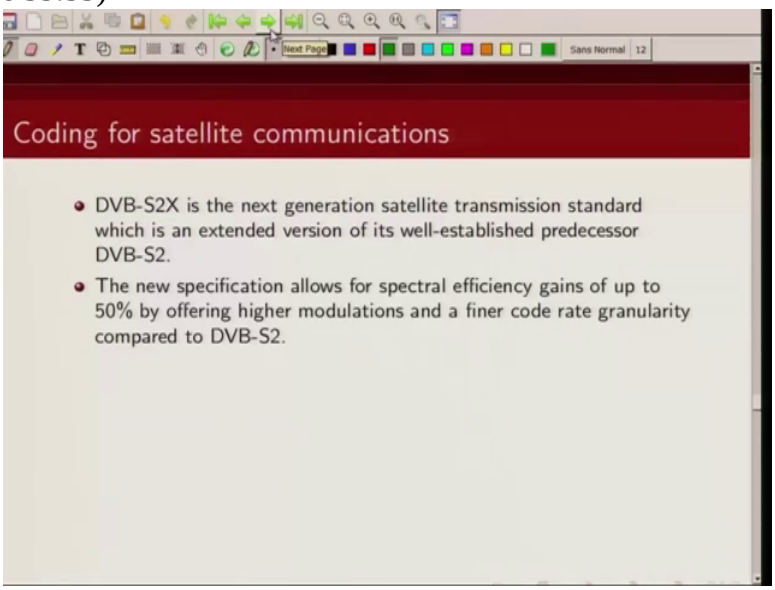
So arise, it can support block size of 16200 and these are the various code rates that have been supported. So L D P C codes of rate, so these are the various code rates that have been supported by this standard. Similarly for block size of 64800, these are the various code rates that have been supported in this standard. Now there is

(Refer Slide Time 35:41)



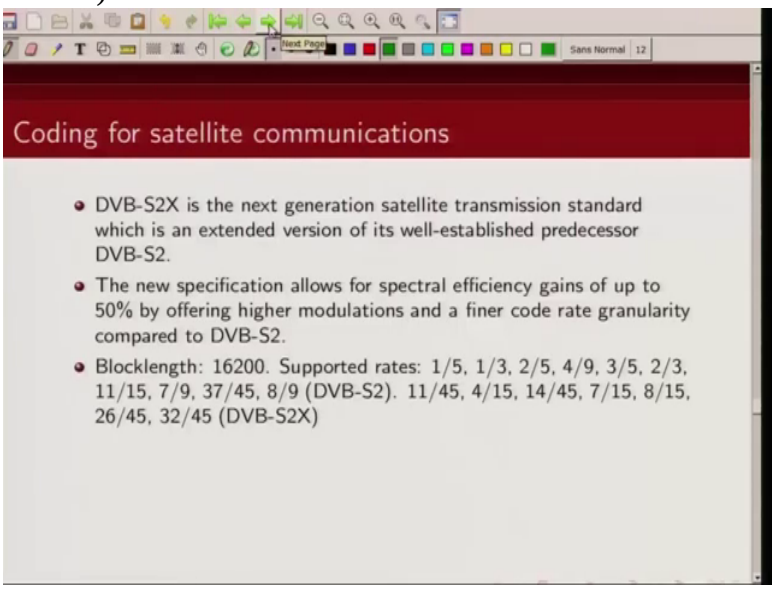
another standard which is upgradation of this D V B S 2 standard called D CV B S 2 X which is an extended version of D V B S 2 standard, and

(Refer Slide Time 35:53)



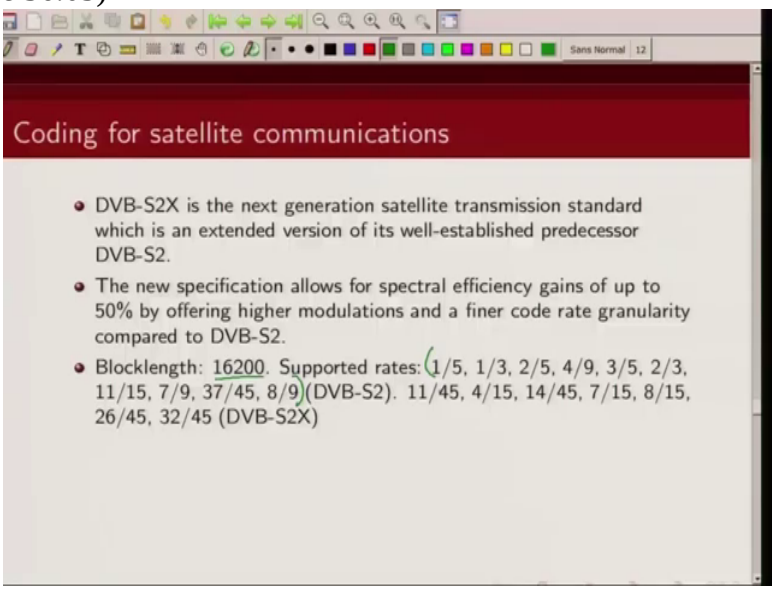
it allows more finer rates of L D P C codes.

(Refer Slide Time 35:58)



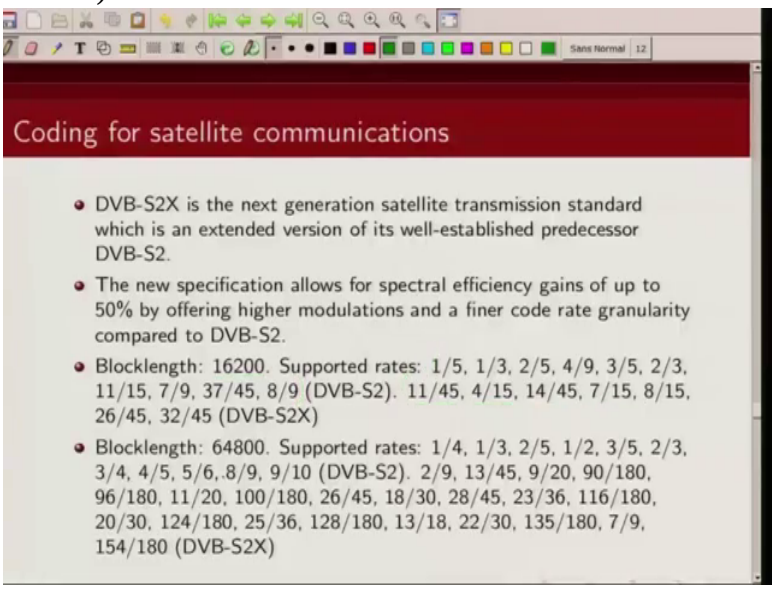
So you can see for the block length of 16200, these were the rates which were supported

(Refer Slide Time 36:05)



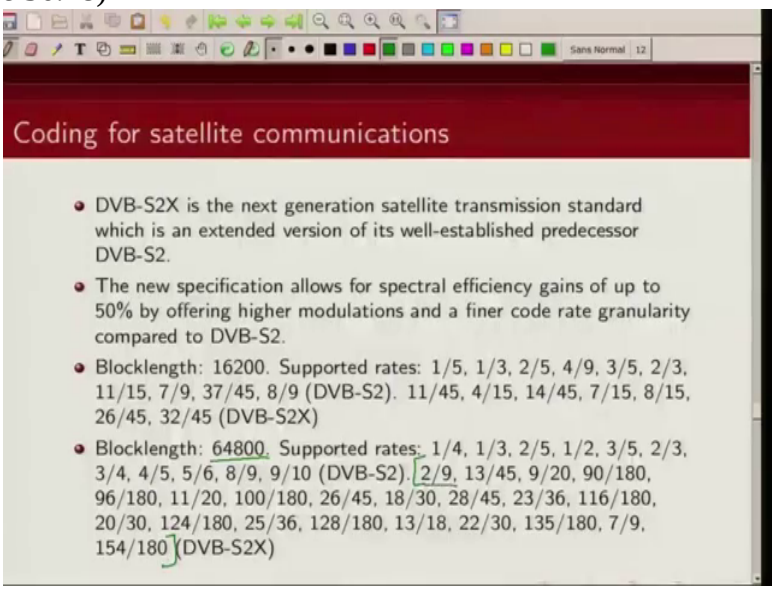
by D V B S 2. D V B S 2 X can further support these rates. So it provides much finer code rate granularity compared to D V B S 2 standard and similarly

(Refer Slide Time 36:22)



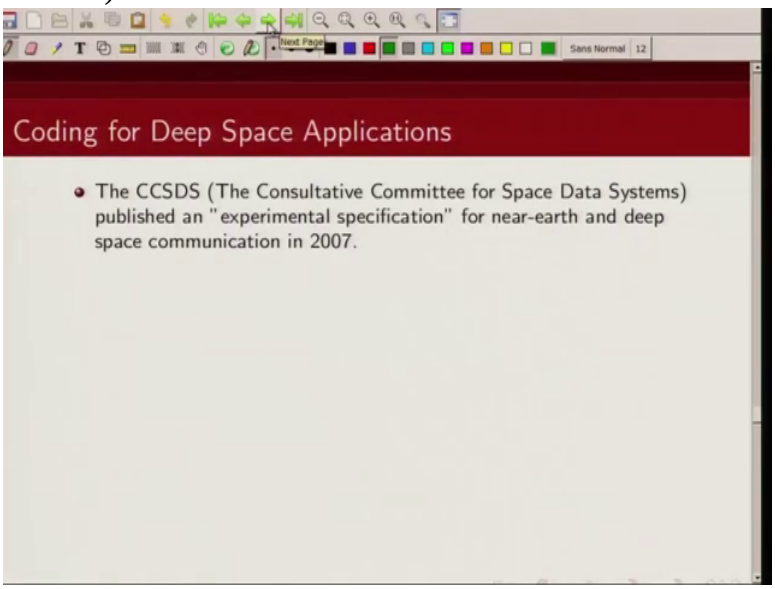
for block size also, 64800, you can see these rates that you see here,

(Refer Slide Time 36:28)



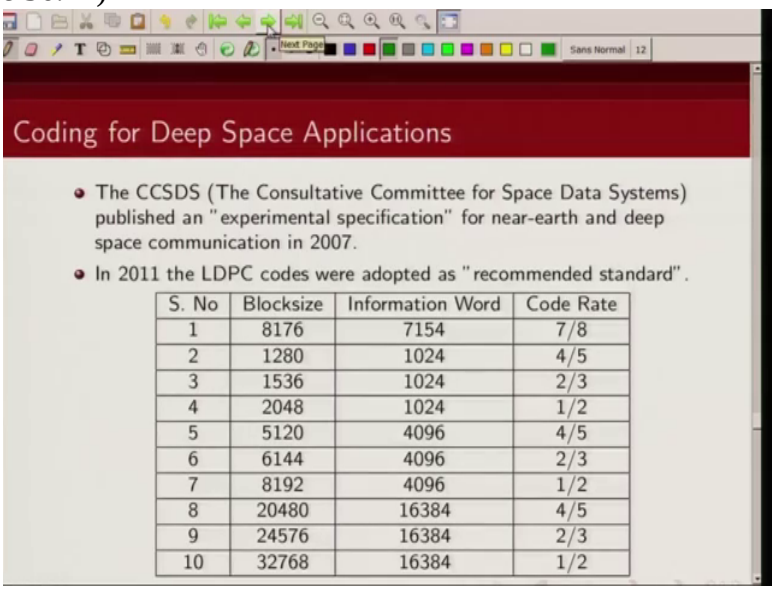
from here, starting from here these rates were not supported in D V B S 2 which are supported in D V B S 2 X.

(Refer Slide Time 36:37)



What about deep space exploration? Now NASA again has specified, has recommended

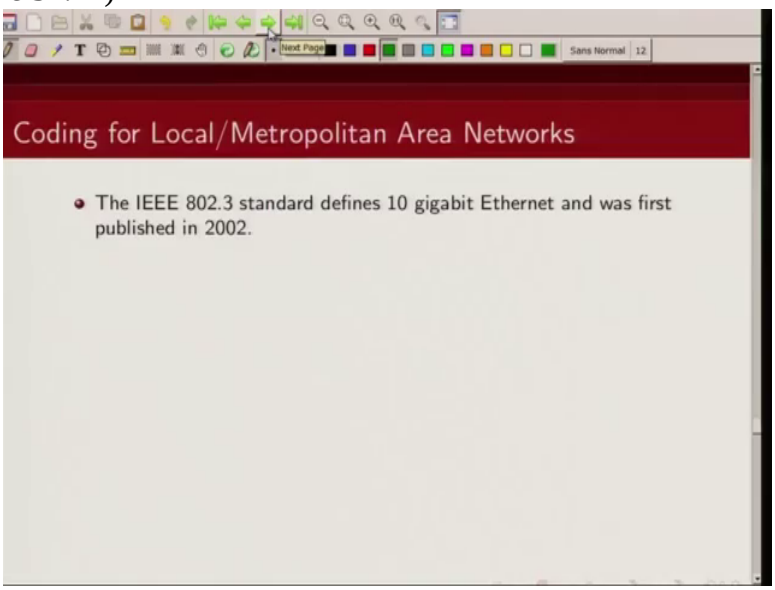
(Refer Slide Time 36:44)



S. No	Blocksize	Information Word	Code Rate
1	8176	7154	7/8
2	1280	1024	4/5
3	1536	1024	2/3
4	2048	1024	1/2
5	5120	4096	4/5
6	6144	4096	2/3
7	8192	4096	1/2
8	20480	16384	4/5
9	24576	16384	2/3
10	32768	16384	1/2

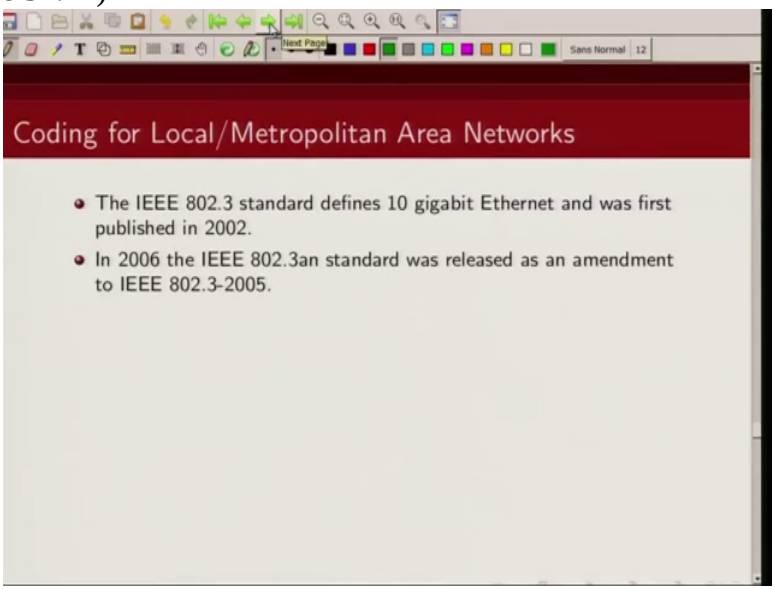
standard. So these codes that you see here, these are, been prescribed for deep space communication and this is for near earth communication. So it's for deep space communication, it supports 3 different block length 1024, 4096 and 16384 so these 3 different information block length will be supported and these are the various rates which have been supported. So correspondingly these are the block sizes that you will get.

(Refer Slide Time 37:21)



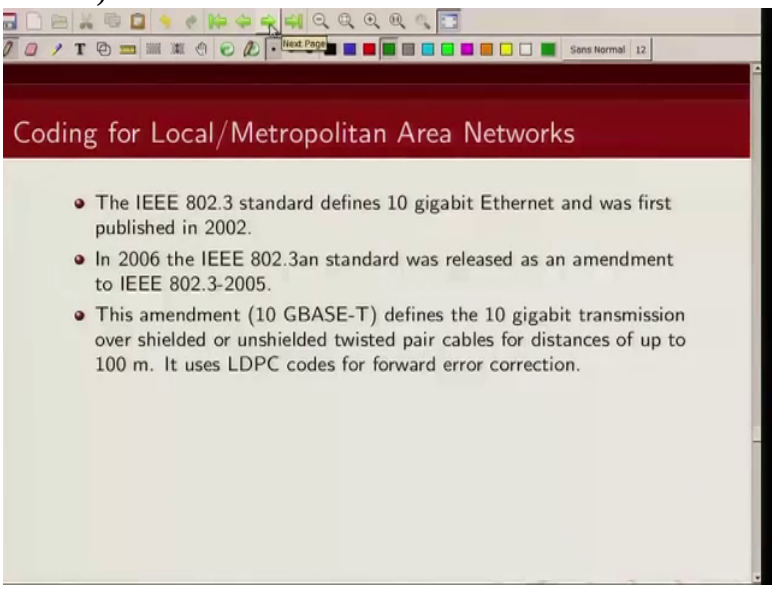
Now for 10 Giga bit Ethernet also L D P C codes have been

(Refer Slide Time 37:27)



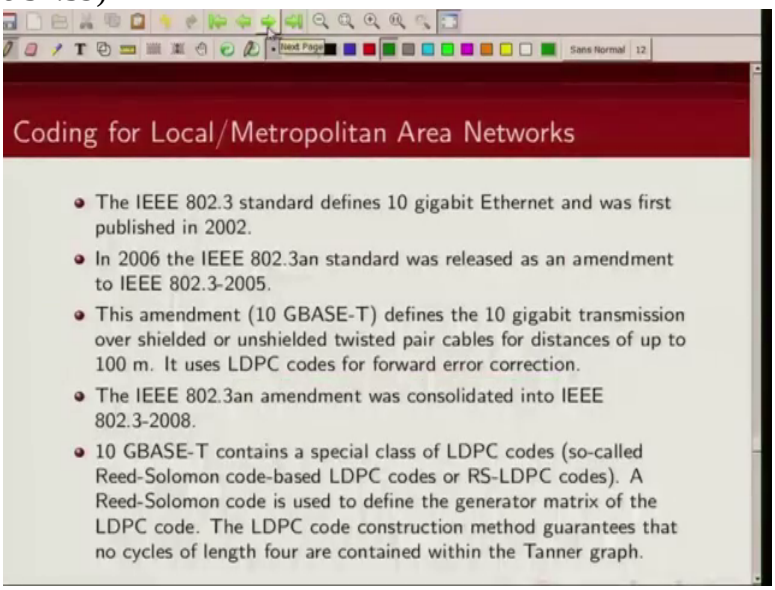
suggested. So

(Refer Slide Time 37:29)



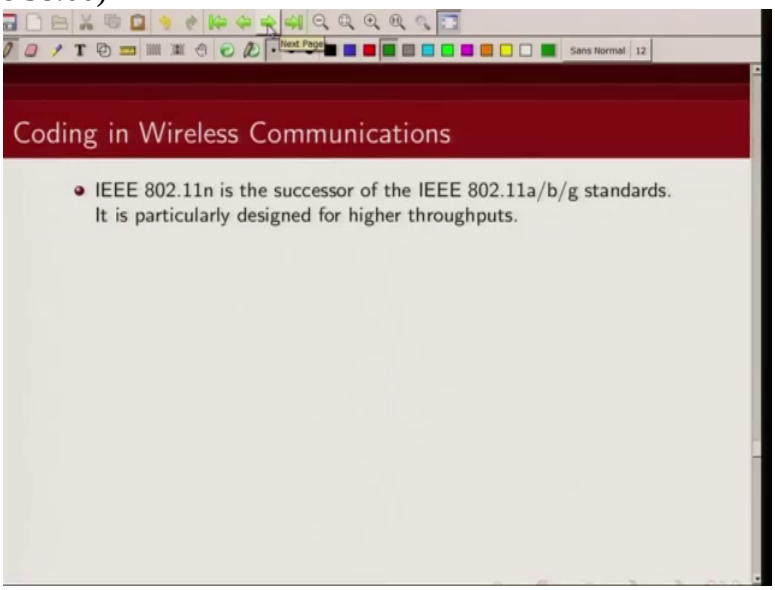
L D P C codes have been used for forward error correction here. For 10 Giga bit transmission over Ethernet and

(Refer Slide Time 37:39)



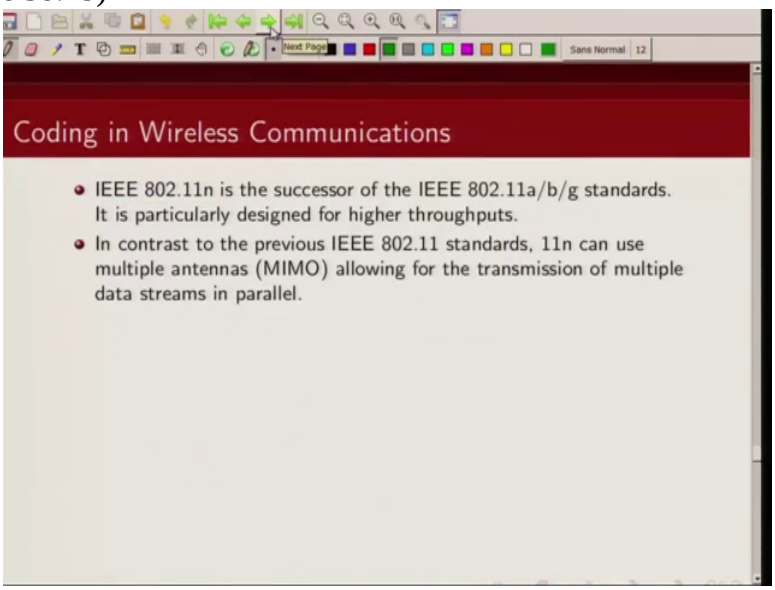
it uses Reed Solomon code based L D P C code. So Reed Solomon code is used to define the generator matrix of L D P C code and this construction ensures that there is no cycle, four cycles in the L D P C code.

(Refer Slide Time 38:00)



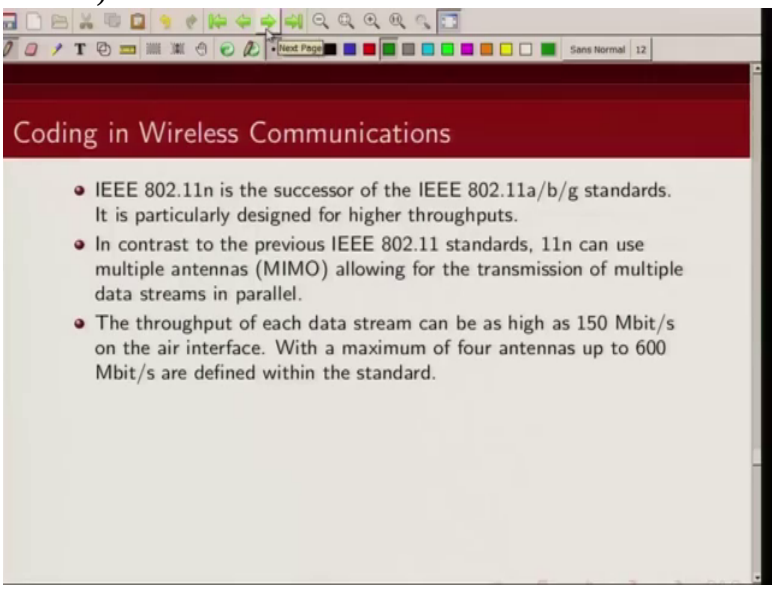
Now let's move to applications in Wi Fi. So I E E E 8 2 2 dot 11 N which is the successor of I E E E 8 2 2 dot 11 a b and g and it supports multiple

(Refer Slide Time 38:15)



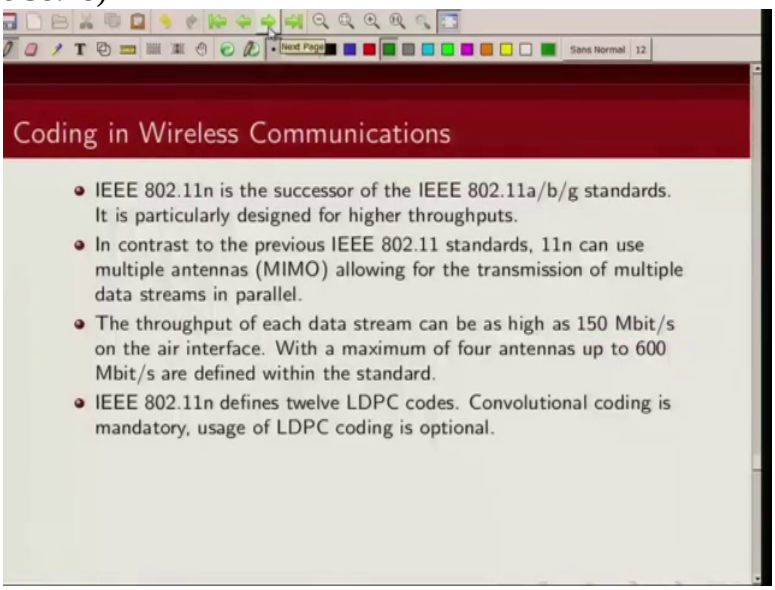
antennas for transmission

(Refer Slide Time 38:18)



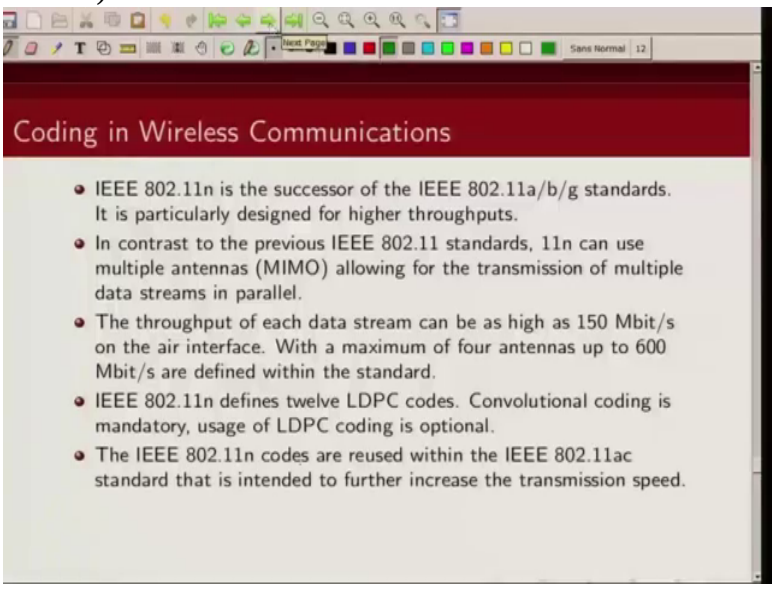
so you could get about 150 M b p s in each data stream. If you use 4 antennas, you can get up to 600 M b p s. Now

(Refer Slide Time 38:28)



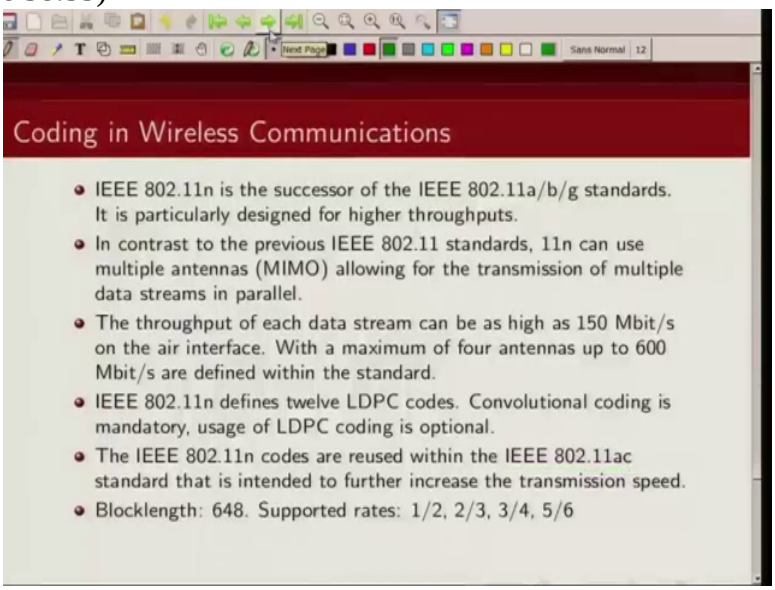
here 12 different L D P C codes have been suggested. Now use of L D P C codes in the standard is optional. The mandatory requirement is convolutional code but they have specified L D P C codes which are optional, which can be used

(Refer Slide Time 38:46)



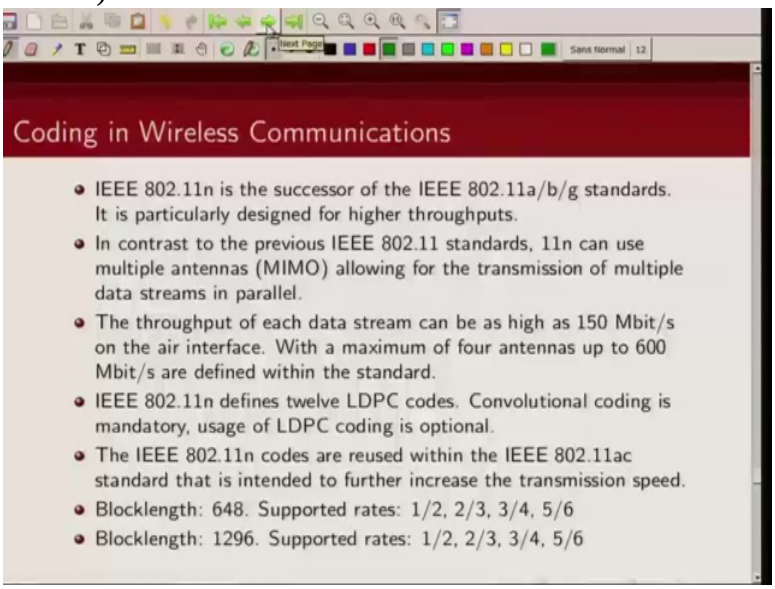
and then these codes have been further reused in this standard 8 2 2 dot 11 a c to further increase the transmission rate.

(Refer Slide Time 38:55)



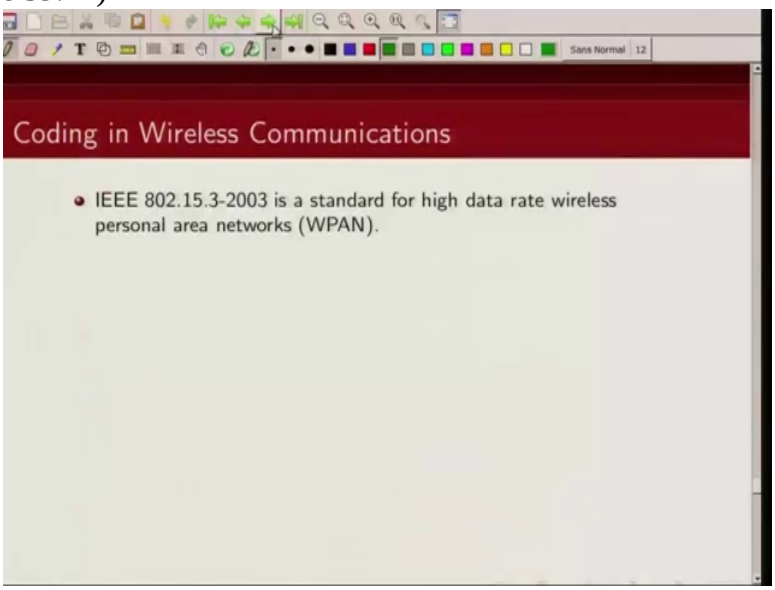
So I will list here some of the block lengths which have been supported. So block lengths

(Refer Slide Time 39:00)



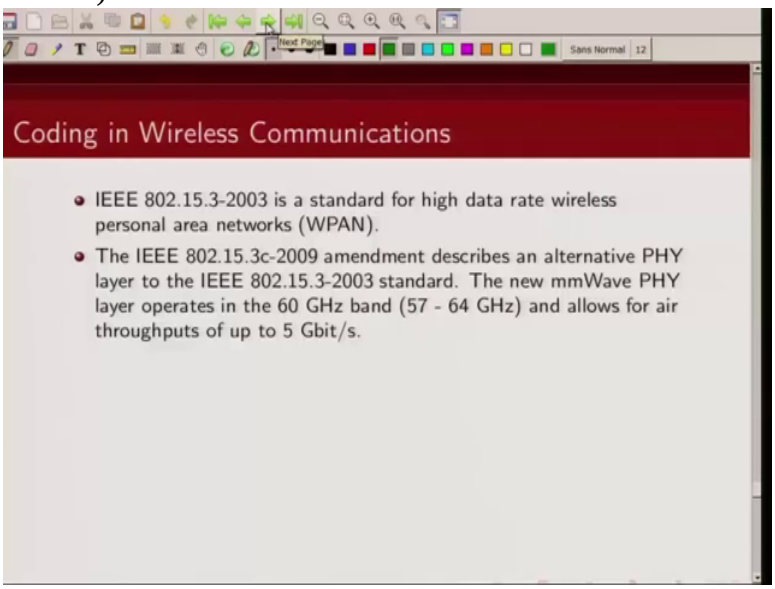
648 and these are the various code rates supported here. Similarly for block length of 1296, these are the various code rates which have been supported. Then we

(Refer Slide Time 39:12)



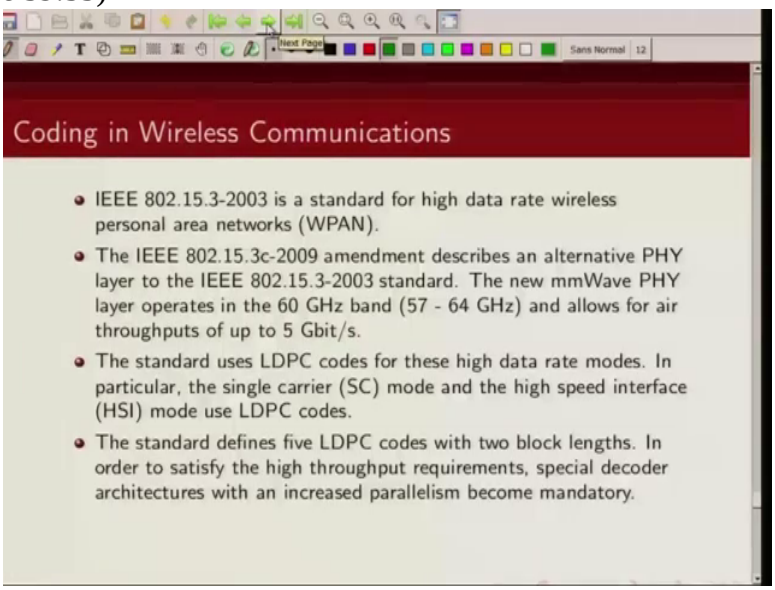
have I E E E 8 2 2 dot 15 which is a standard for wireless personal area network.

(Refer Slide Time 39:19)



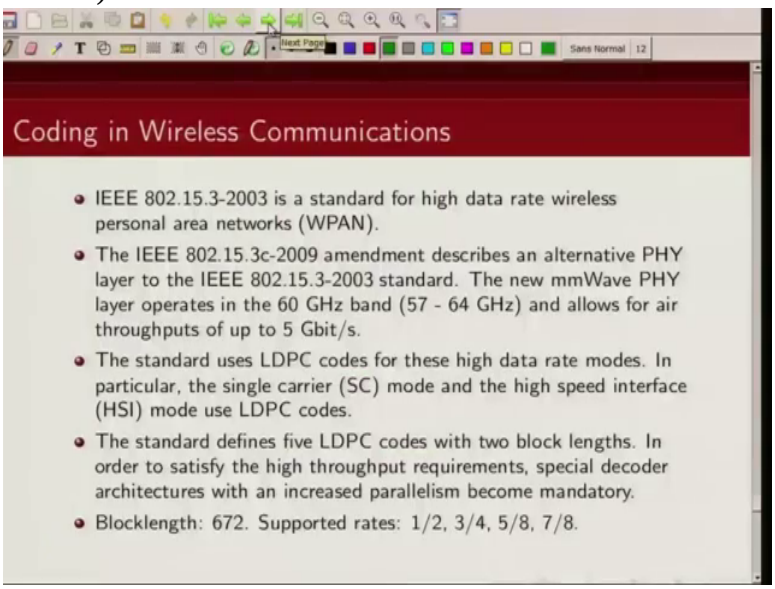
Here also there is a new PHY layer which is based on millimeter Wave in 16 Giga Hertz band which allows throughput up to 5 Giga bits per second. Now here

(Refer Slide Time 39:33)



L D P C codes have been suggested. Now 5 different L D P C codes with two different block length has been suggested. Now

(Refer Slide Time 39:43)

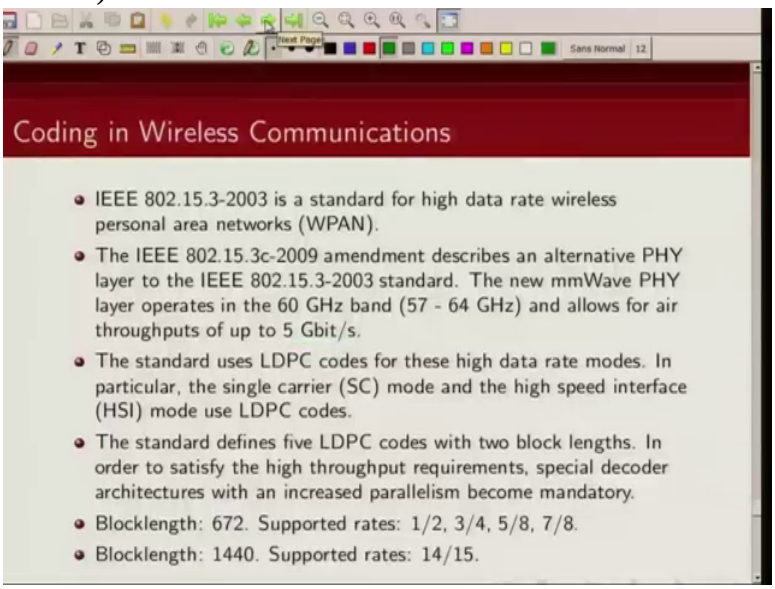


typically the block size here is smaller so as to ensure and also its architecture is such that because you have to get 5; throughput up to 5 gigabits per second so architecture is such that

(Refer Slide Time 39:54)

there is lot of parallelism and the decoder can be implemented in parallel. So the rates which are

(Refer Slide Time 39:59)

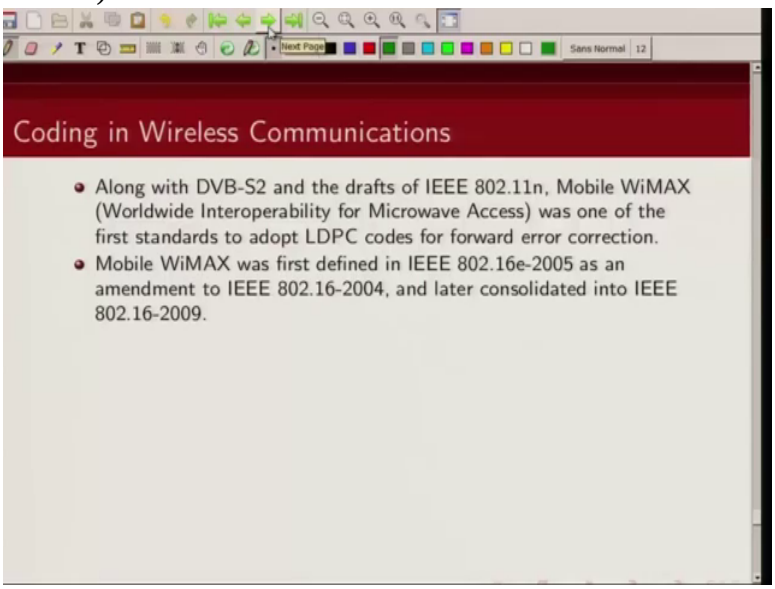


supported, so block size 672 and these are the rate, code rates which are supported here. Similarly for block length of 1440, a code rate of 14 by 15 has been supported

(Refer Slide Time 40:14)

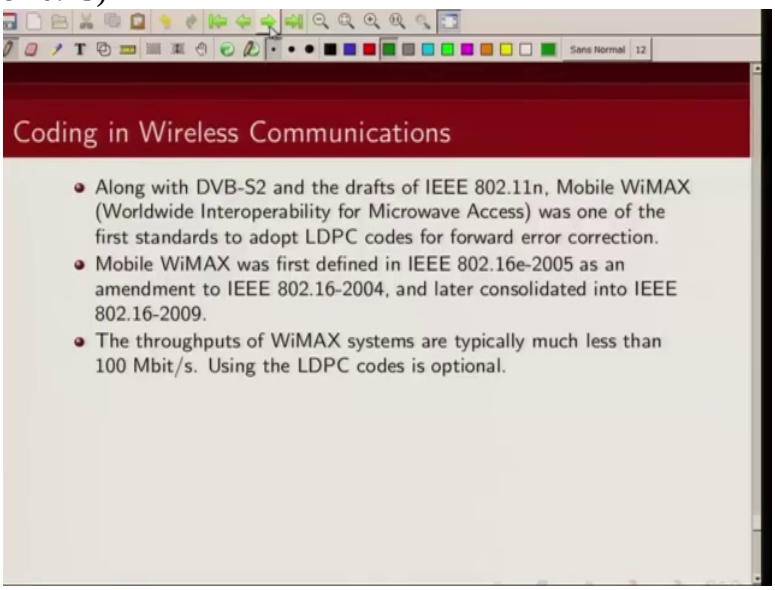
Now another standard where L D P C codes have been used has been, from the very beginning is this Wi MAX. And

(Refer Slide Time 40:21)



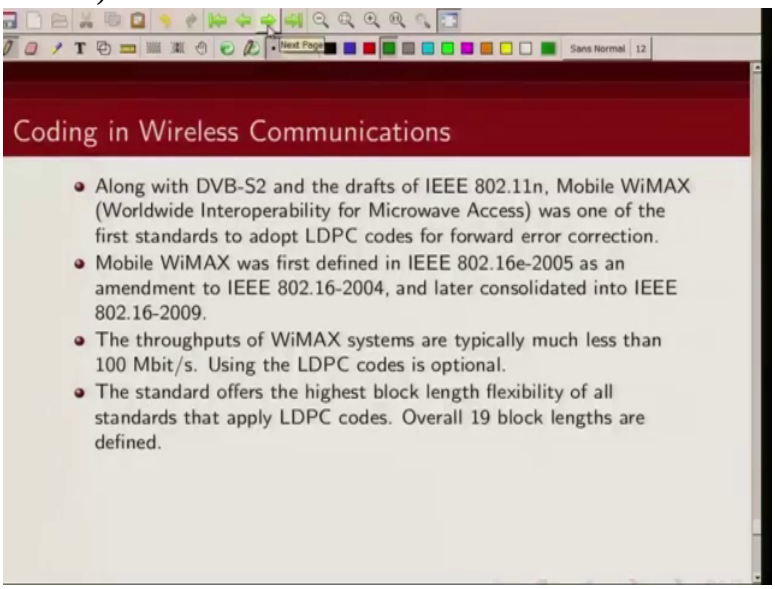
there are lots of codes which have

(Refer Slide Time 40:23)



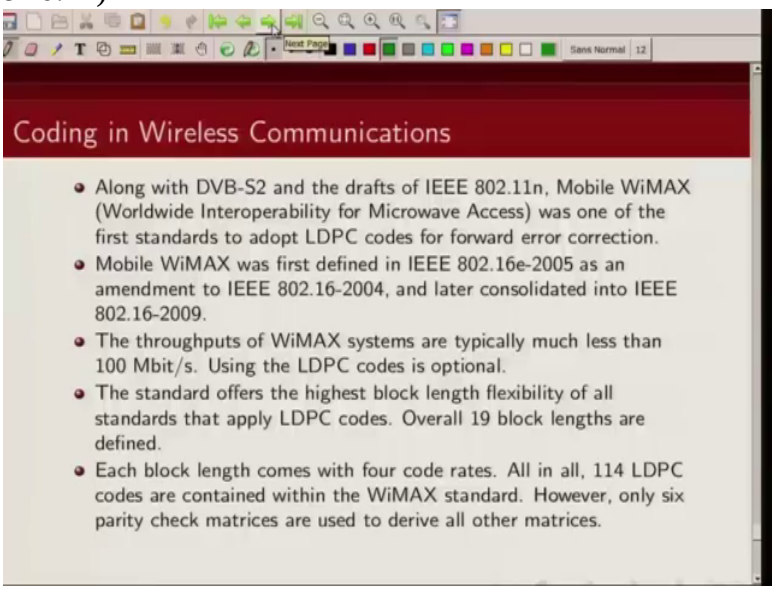
been defined in this Wi MAX standard; again, use of L D P C code is optional here but they have

(Refer Slide Time 40:31)



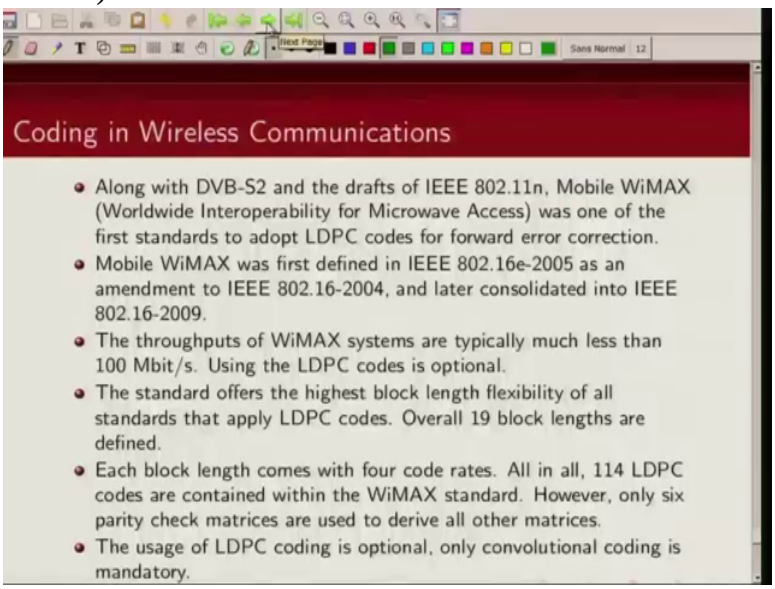
suggested lot of flexibility as far as block length is concerned and 19 different block lengths have been considered, and various different rates have been considered.

(Refer Slide Time 40:42)



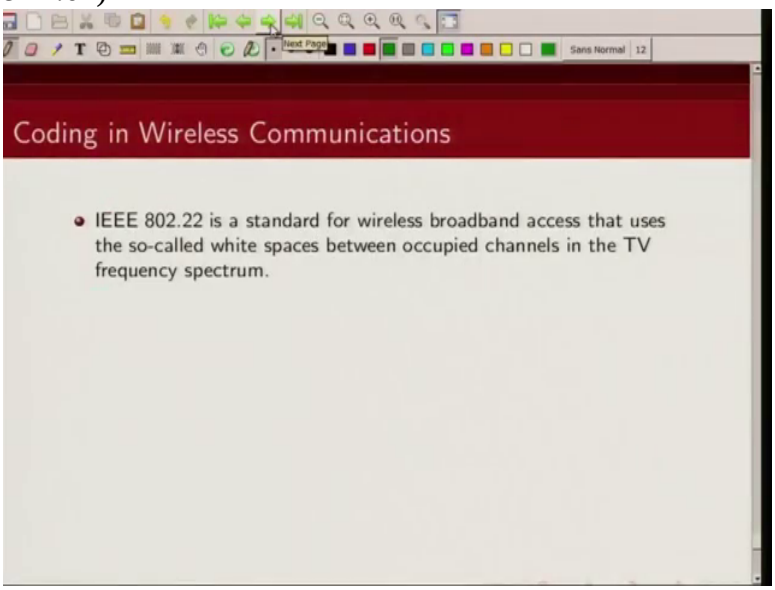
Total of 114 L D P C codes have been, you know defined or contained in this Wi MAX standard. And as

(Refer Slide Time 40:53)



I said, here also the use of L D P C codes is optional; the mandatory requirement is convolutional code.

(Refer Slide Time 41:01)

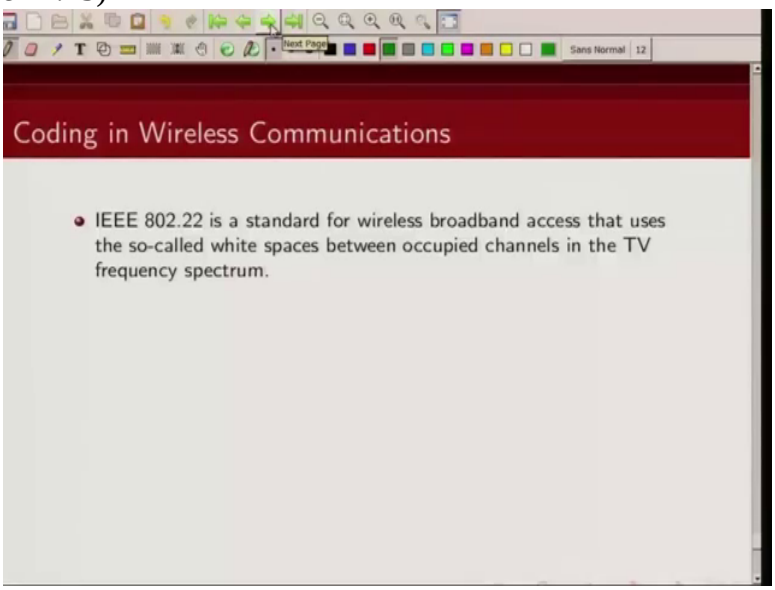


And finally I will conclude this discussion with I E E E 8 0 2 dot 2 2

(Refer Slide Time 41:07)

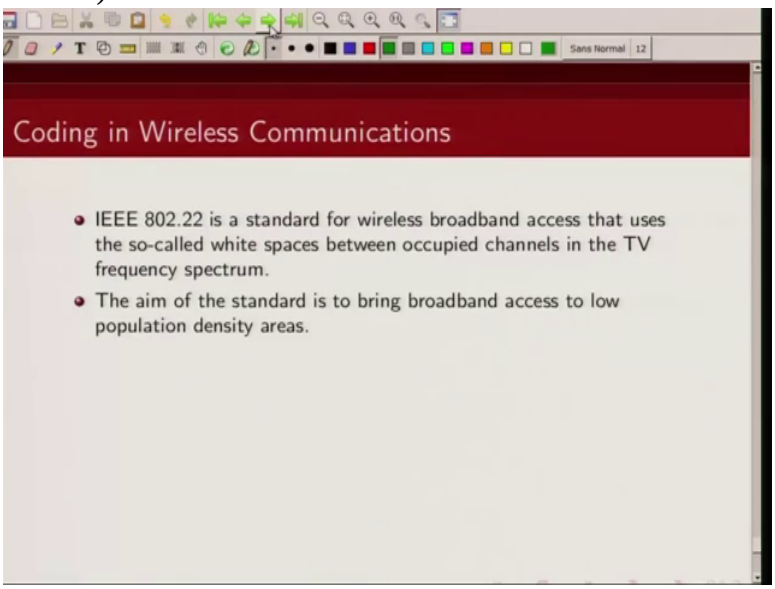
standard which is basically for cognitive radio, opportunistic access over T V white spaces,

(Refer Slide Time 41:15)



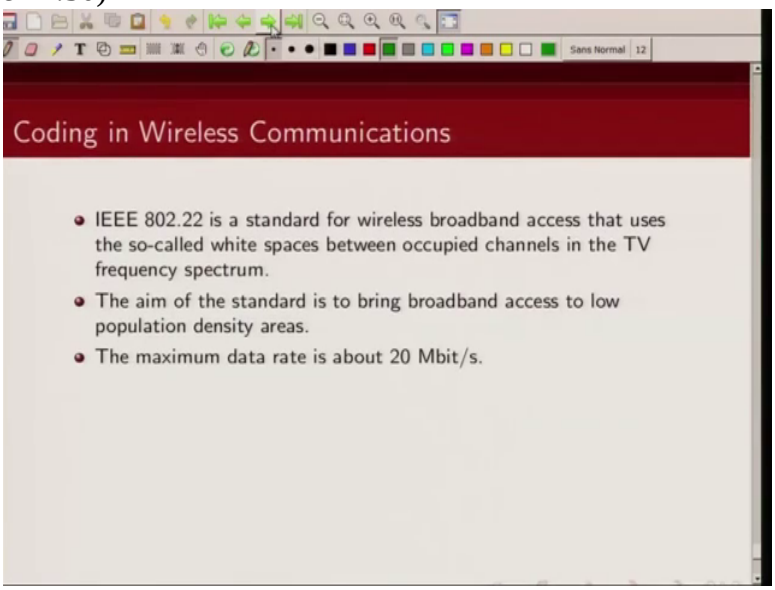
the T V spectrum which is not being used for communi, which is not, which can be opportunistically used for communication and

(Refer Slide Time 41:23)



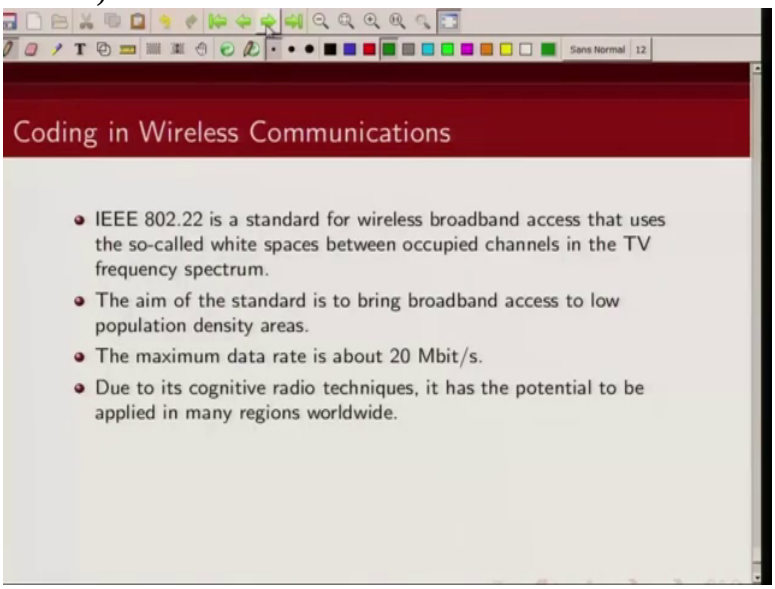
the whole idea of I E E E dot 8 2 2 is to provide rural connectivity. So here also they have specified

(Refer Slide Time 41:30)



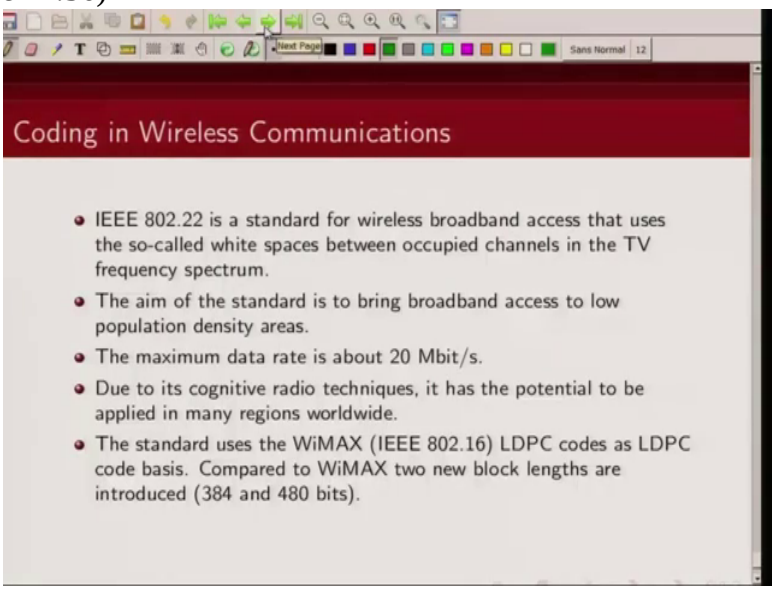
lots of various L D P C codes

(Refer Slide Time 41:33)



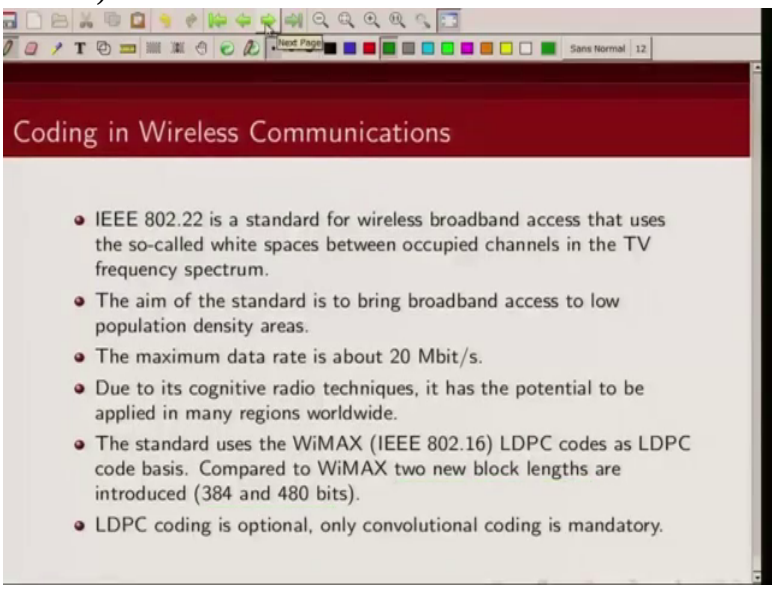
and in fact they have 2 different,

(Refer Slide Time 41:36)



2 new block lengths which were not there in Wi MAX has been added, one corresponding to 384 bits and another corresponding to 480 bits. So this is another application where

(Refer Slide Time 41:49)



L D P C code is optional, convolutional code is mandatory but L D P C code is optional has been suggested.

(Refer Slide Time 41:57)

So this is not an exhaustive list of applications, I just wanted to give you some flavor of where in practice we use linear block codes, thank you.