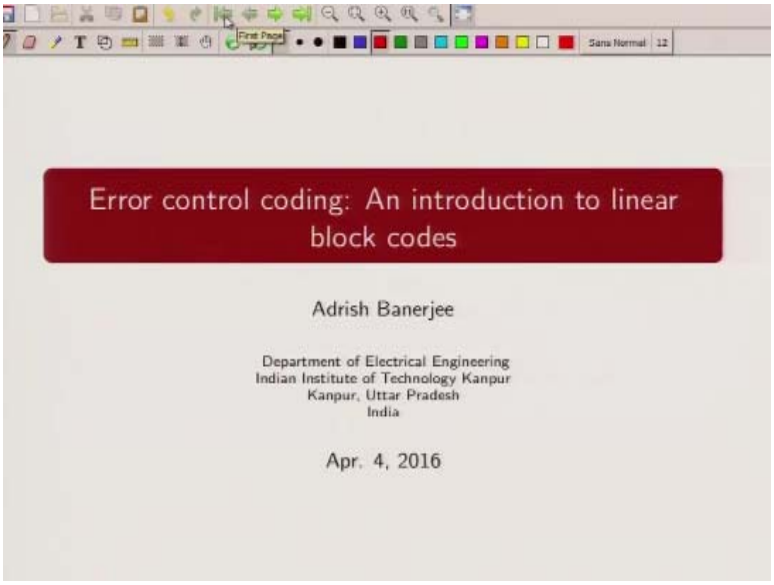
Indian Institute of Technology Kanpur
National Programme on Technology Enhanced Learning (NPTEL)
Course Title
Error Control Coding: An Introduction to Linear Block Codes

Lecture – 9A
Decoding of low density parity check codes-I

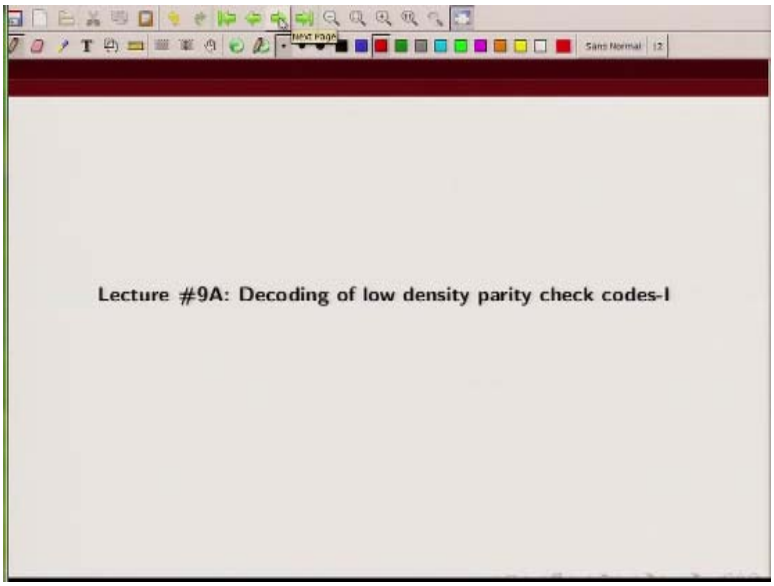
by
Prof. Adrish Banerjee
Department of Electrical Engineering, IIT Kanpur

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

(Refer Slide Time: 00:20)

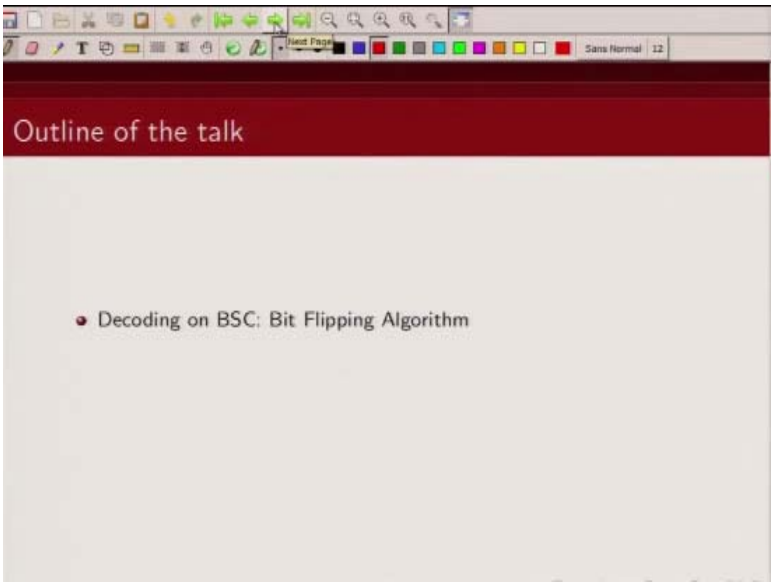


(Refer Slide Time: 00:22)



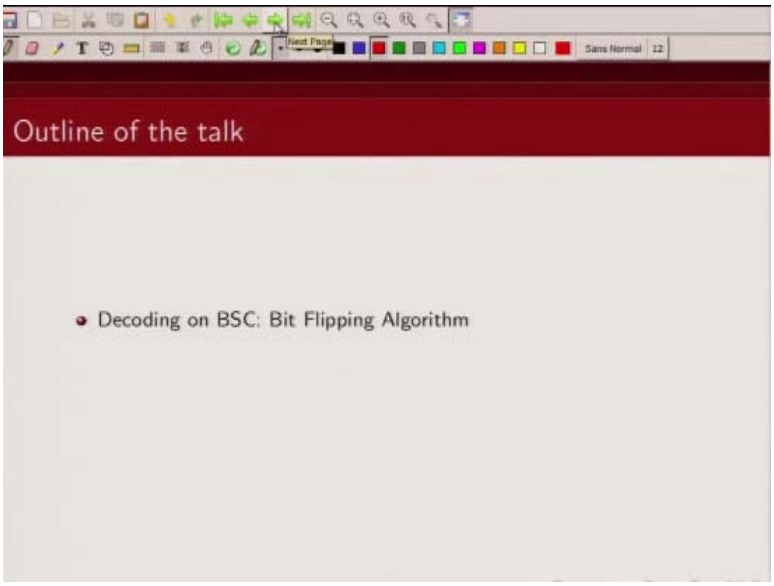
Today we are going to discuss decoding of LDPC codes.

(Refer Slide Time: 00:28)



So to start with, let us first take a simple example of transmission over a binary symmetric channel and we are going to talk about a bit flipping algorithm to decode LDPC codes. And then in the next lecture we will talk about probabilistic decoding algorithm based on belief propagation.

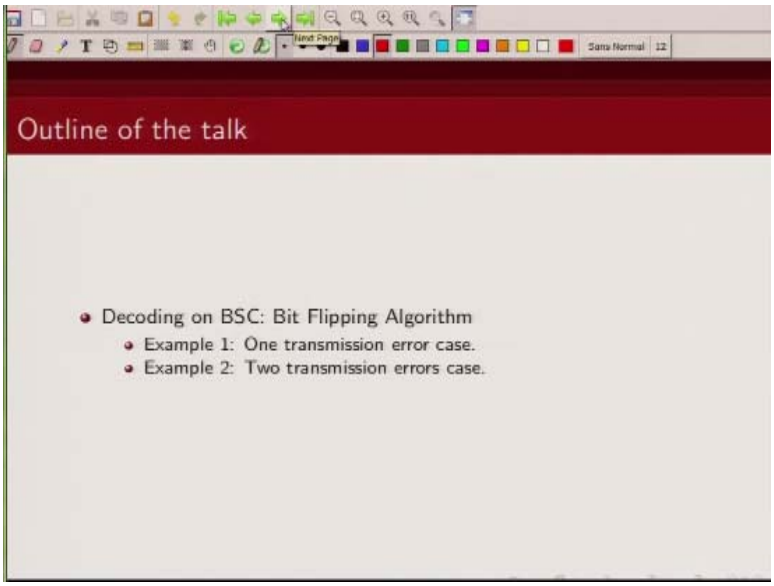
(Refer Slide Time: 00:48)



(Refer Slide Time: 00:49)

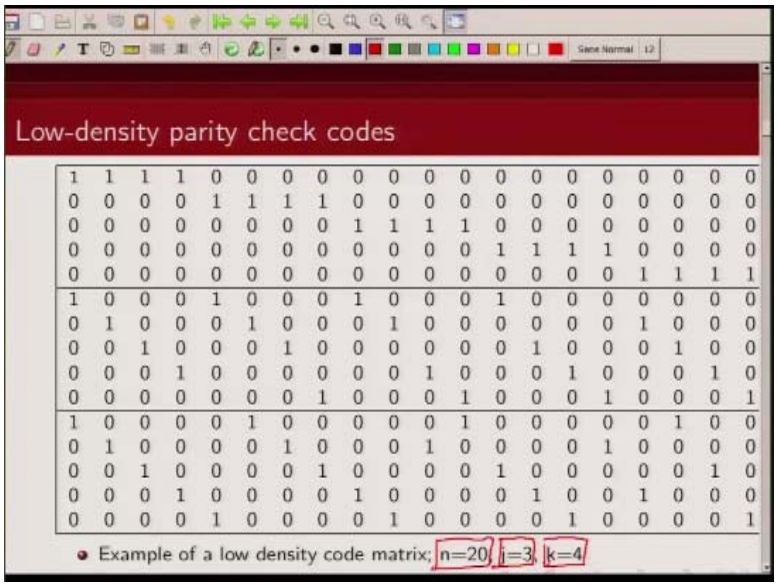
So we will consider two cases today, first where there is only error has happened.

(Refer Slide Time: 00:56)



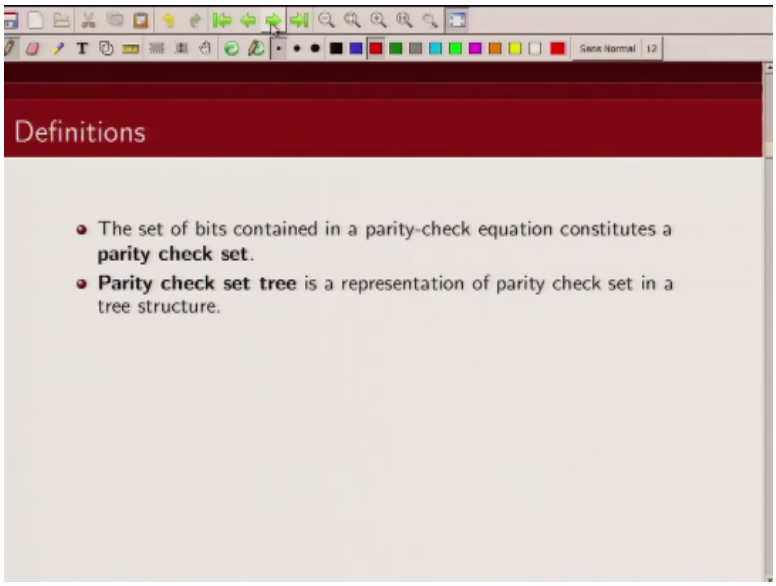
And second where there are two errors have happened. And we will show how we can correct these errors using LDPC codes.

(Refer Slide Time: 01:04)



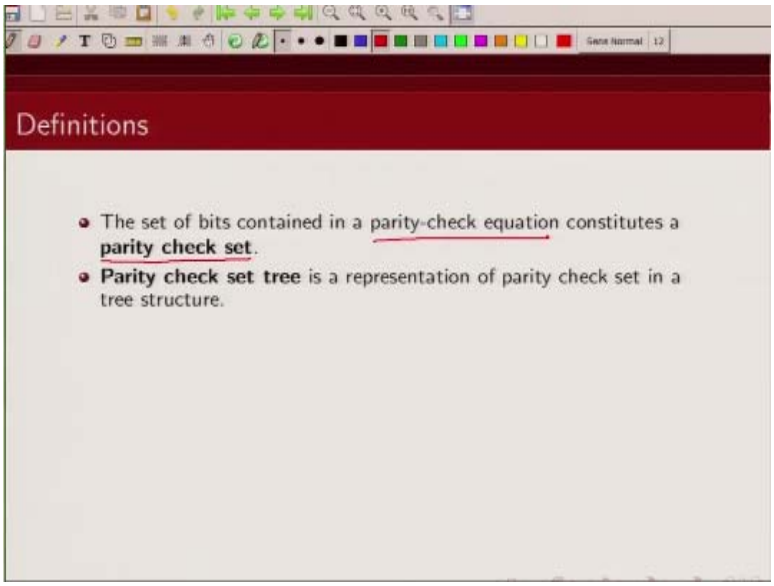
So recall this is an example of a low-density parity-check code, a flock length 20, the column weight is 3, and row weight is 4.

(Refer Slide Time: 01:24)



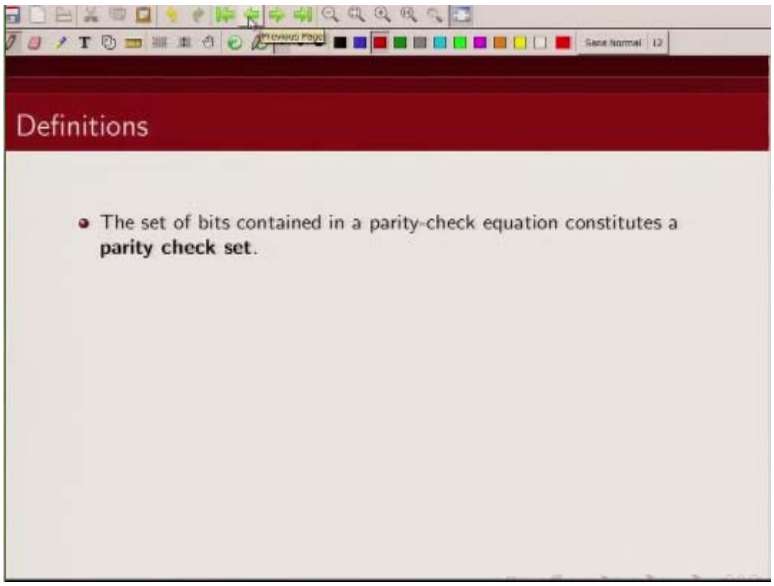
We will first define few terms and then we will come to the decoding of that.

(Refer Slide Time: 01:35)



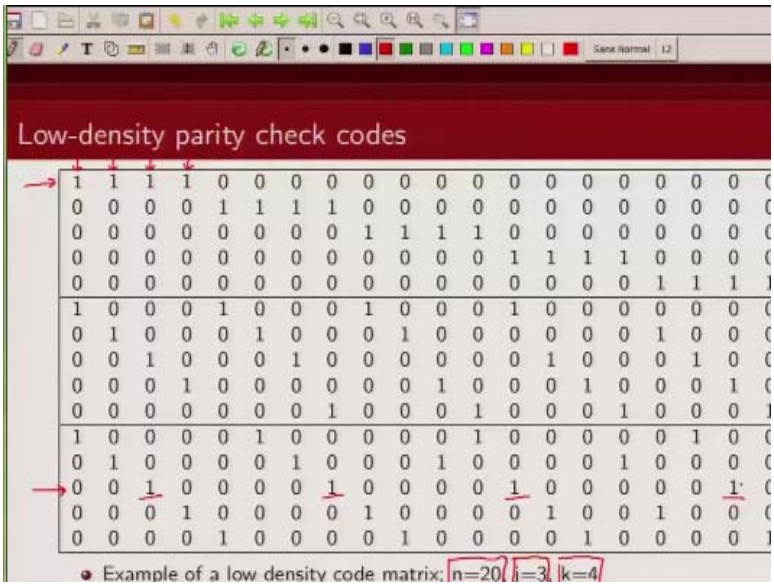
So first thing we will define is what is a parity-check set, so what is a parity-check set? It is the set of bits that are participating in the parity-check equation. So set of bits that participate in a parity-check equation, they constitute a parity-check set.

(Refer Slide Time: 01:54)



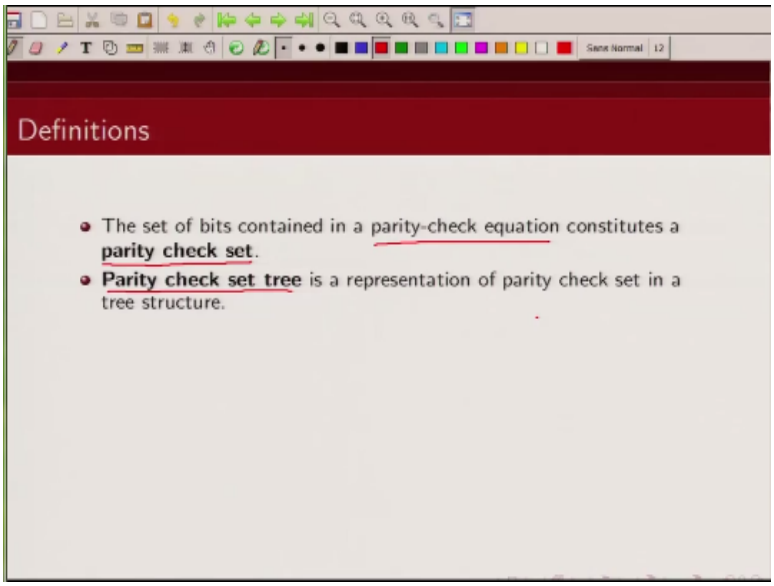
So for example

(Refer Slide Time: 01:56)



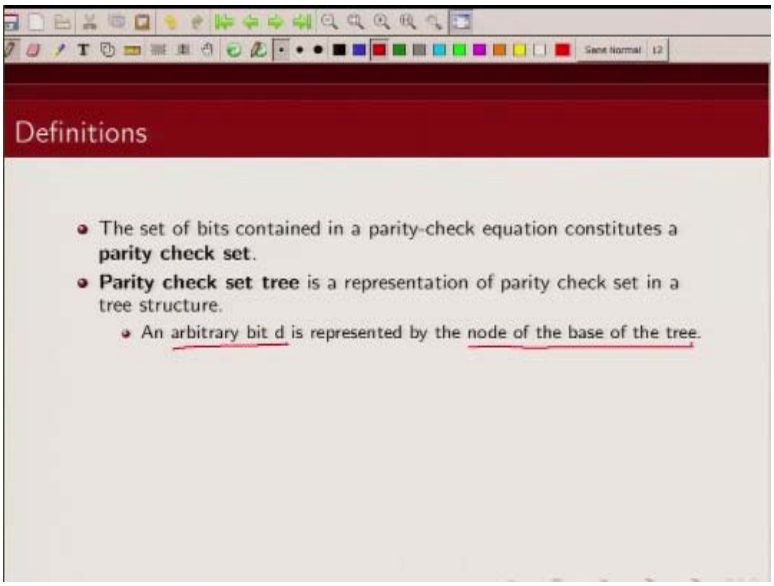
If you look at this particular parity-check equation, now these are the bits that are participating in this parity-check equation. So these bits will form a parity check set, if we look for example at this particular row, now this bit, this bit, this bit, and this bit, these are the four bits that are participating in the parity-check equation. So these bits will form a parity check set.

(Refer Slide Time: 02:32)



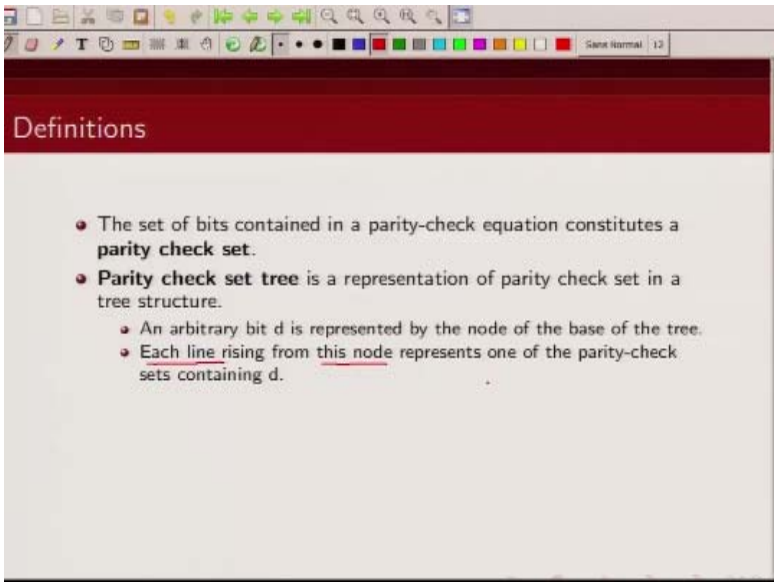
So what is a parity check set tree, it is a graphical representation of the parity-check set in a tree like structure. How? We will explain.

(Refer Slide Time: 02:46)



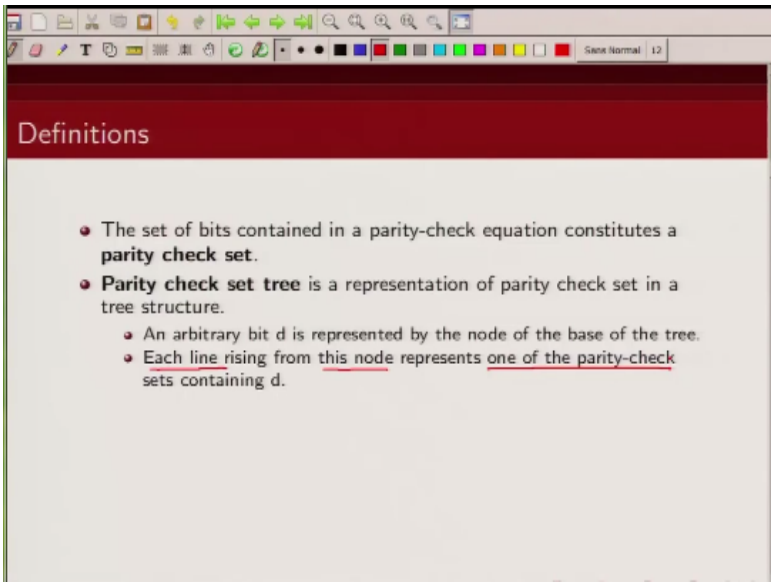
So in any arbitrary bit is represented as node of the base of the tree.

(Refer Slide Time: 02:57)



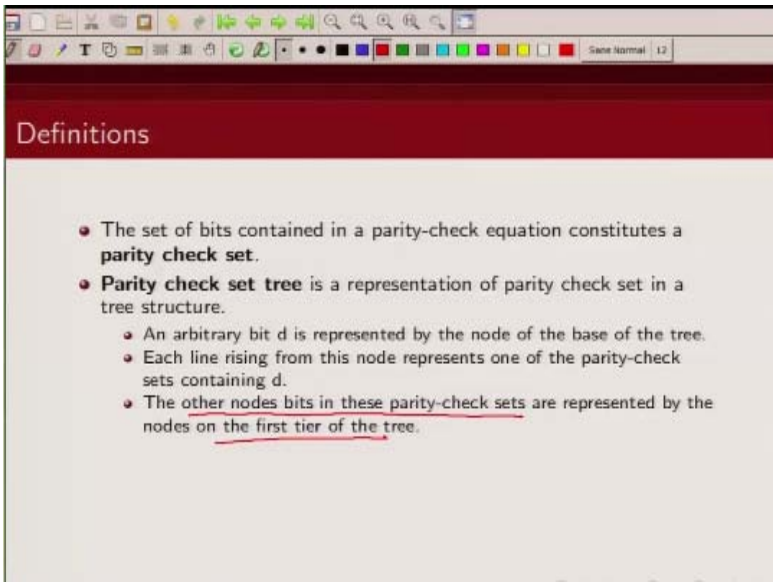
There is a line arising from this node and each of these line represent one parity-check equation where this particular bit is participating.

(Refer Slide Time: 03:12)



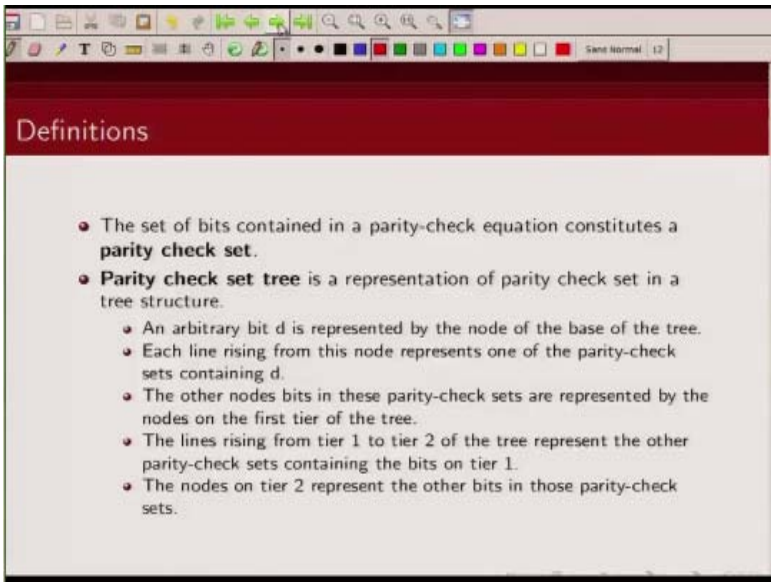
So each line arises from the node and it represents one of the parity-check equations or one of the parity-check sets where this particular node is participating.

(Refer Slide Time: 03:25)



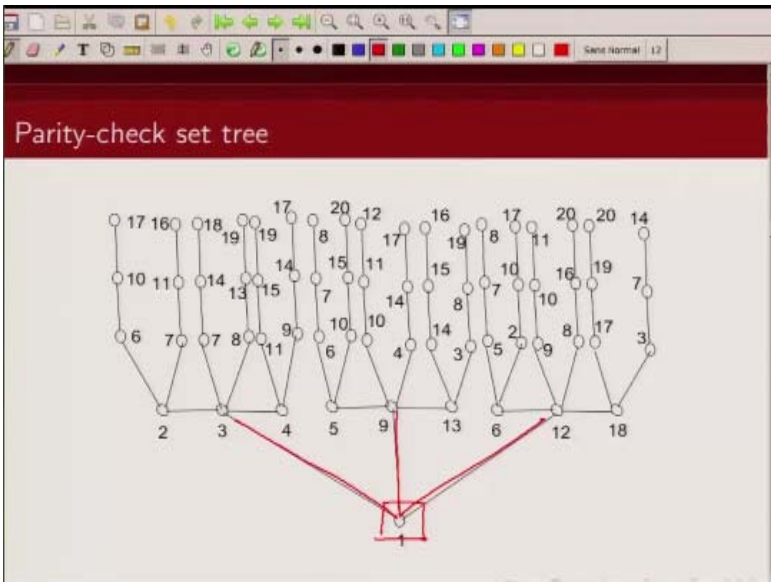
Now other nodes in these parity-check constraints are represented as nodes in the first tier of the tree.

(Refer Slide Time: 03:36)



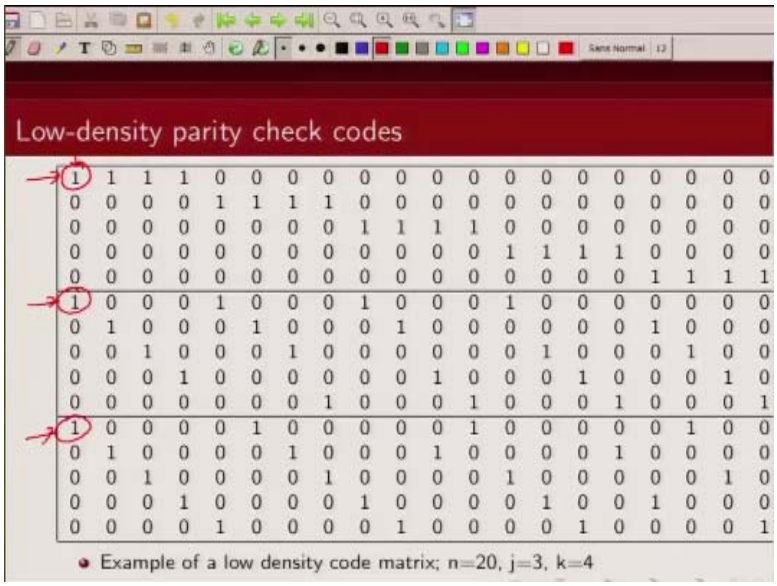
Now what do I mean by this, so let us just look at.

(Refer Slide Time: 03:39)



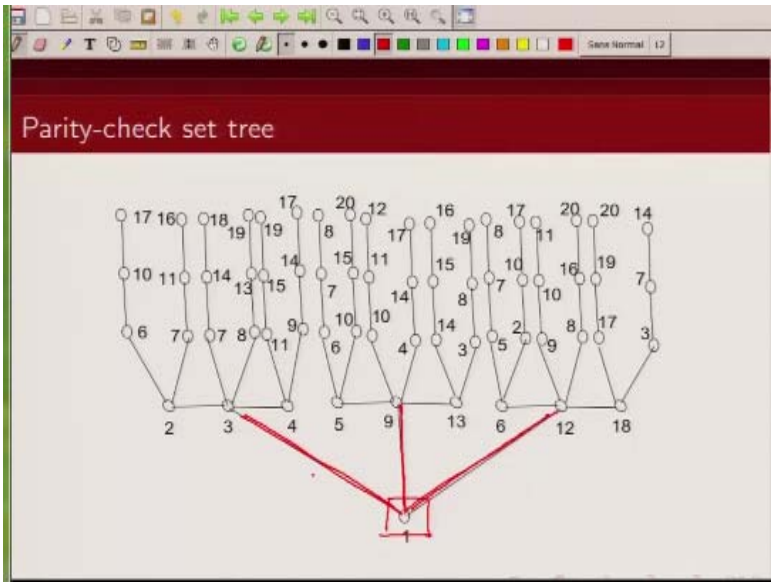
So let us say I have this node, first node calling it node 1. Now this node participates in three parity-check equations, you can see 1, 2, 3, go back to our.

(Refer Slide Time: 04:00)



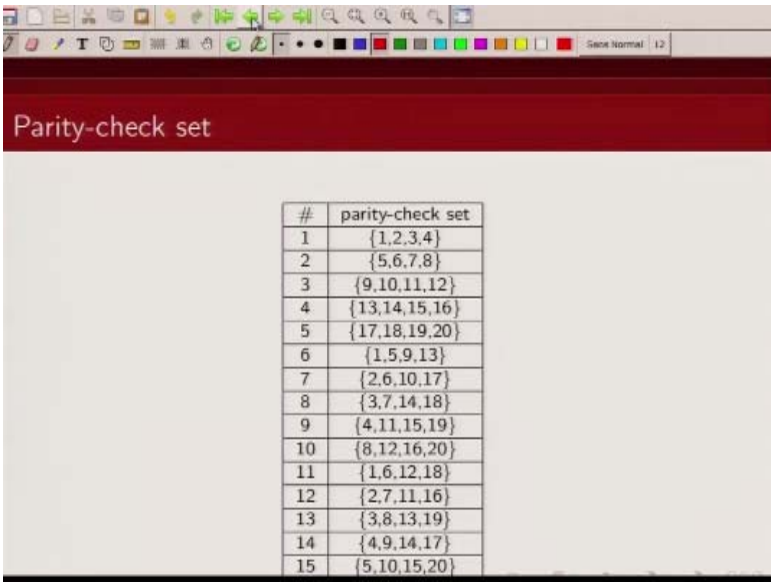
So we are looking at first bit, it participates in this parity-check equation, this parity-check equation, and this parity-check equation.

(Refer Slide Time: 04:14)



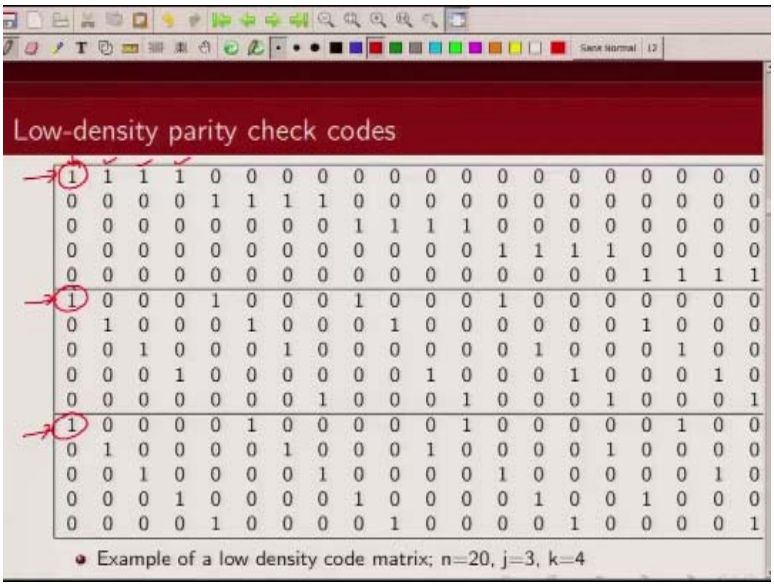
So there is one line corresponding to each of these parity-check equation okay.

(Refer Slide Time: 04:24)



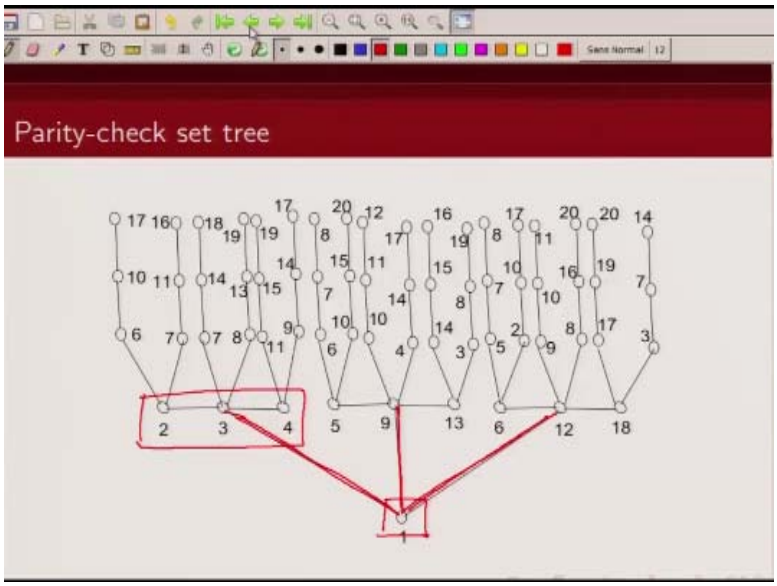
Now in this parity-check equation you can see which are the other bits participating.

(Refer Slide Time: 04:27)



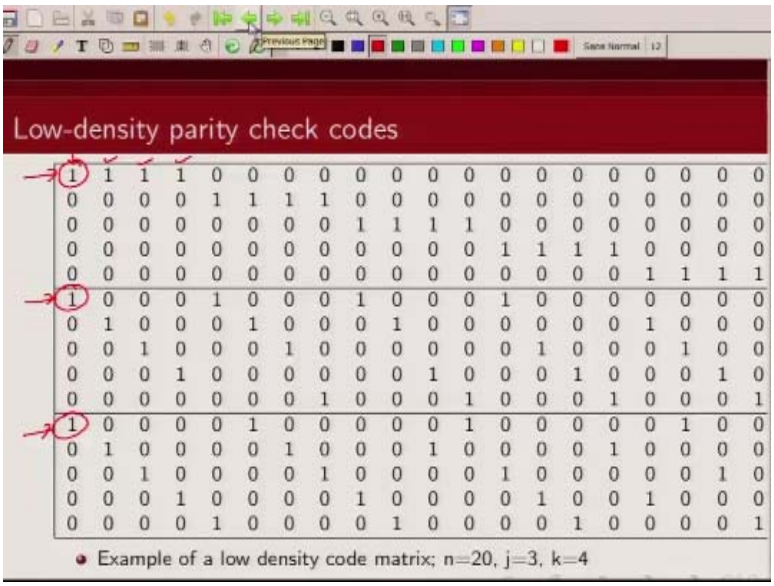
So bit number two, bit number three, bit number four, so how did we write that?

(Refer Slide Time: 04:34)



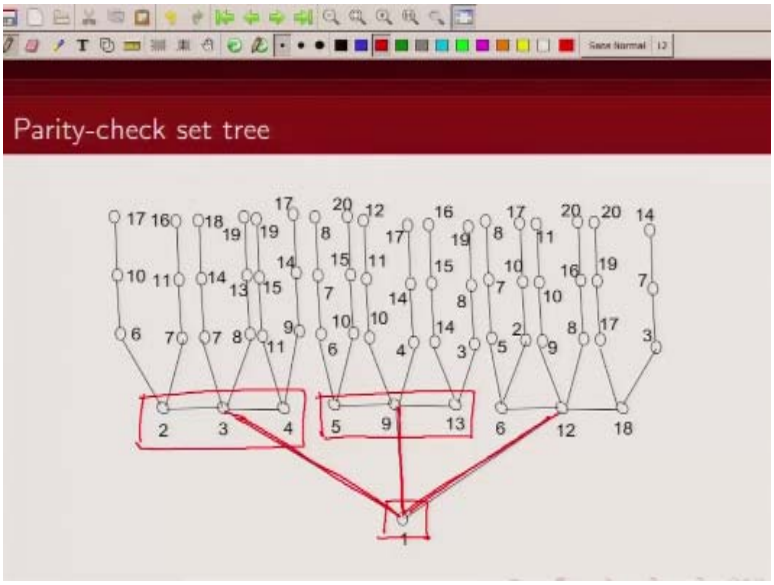
So the other bits that are participating in the parity-check constraints they are written like this. So one, so this is one parity-check constraint and two, three, four bits are participating.

(Refer Slide Time: 04:50)



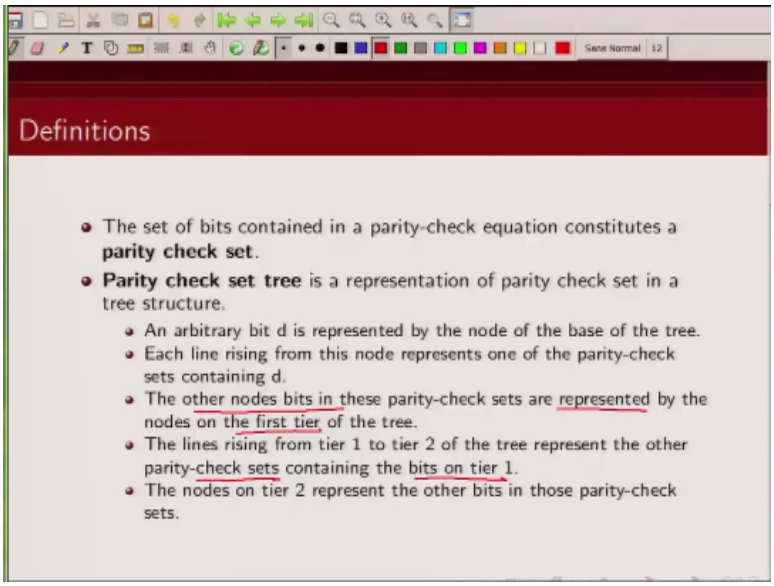
Similarly if you look at here this bit number 5, 9 and 13 are participating in this particular parity-check equation.

(Refer Slide Time: 04:56)



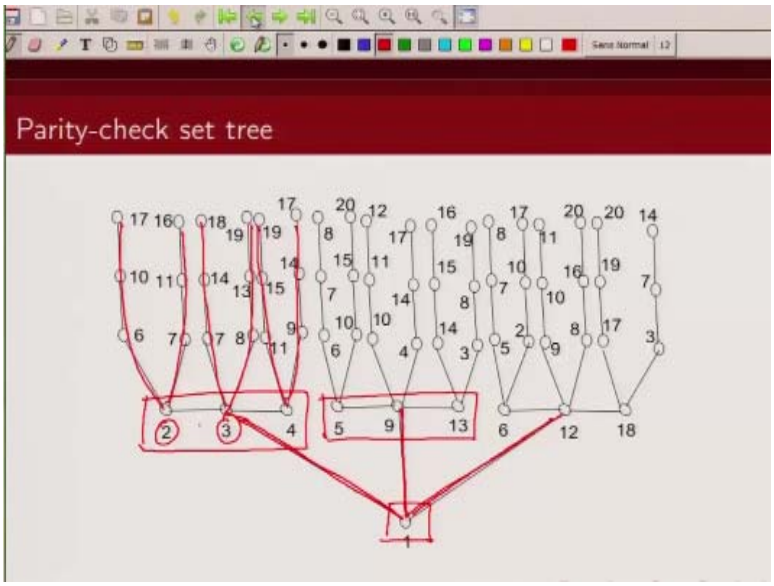
So that is represented by this.

(Refer Slide Time: 05:03)



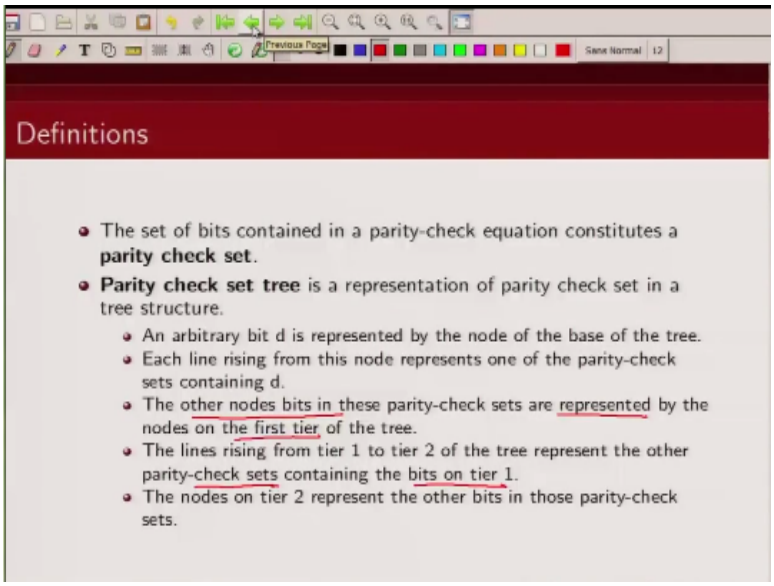
So that is what I mean, when I said other nodes are represented as nodes in the first tier. Now line arises from tier 1 to tier 2 represent the other parity constraints containing bits on tier 1.

(Refer Slide Time: 05:25)



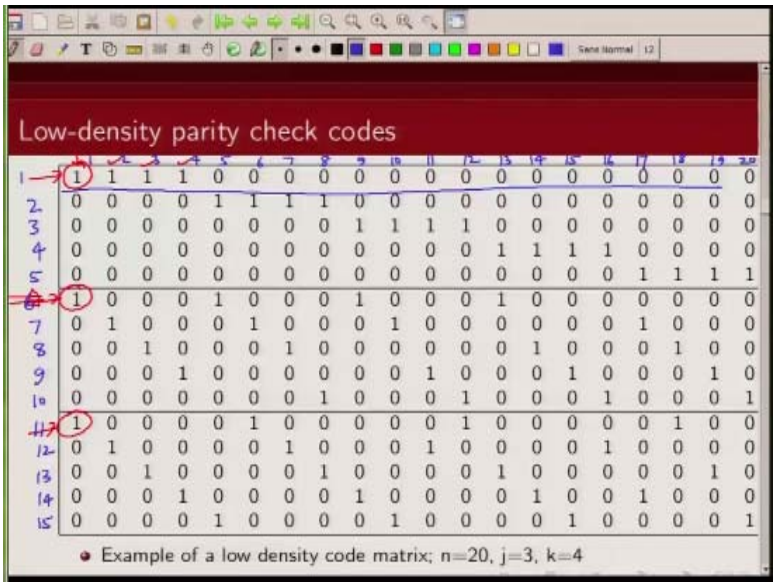
So this is my tier 0, this is tier 1. Now what is that – what are the connections coming here? These are the parity-check constraints involving these bits, involving 2, involving 3 is here, involving 4, these are the parity constraints okay.

(Refer Slide Time: 05:48)



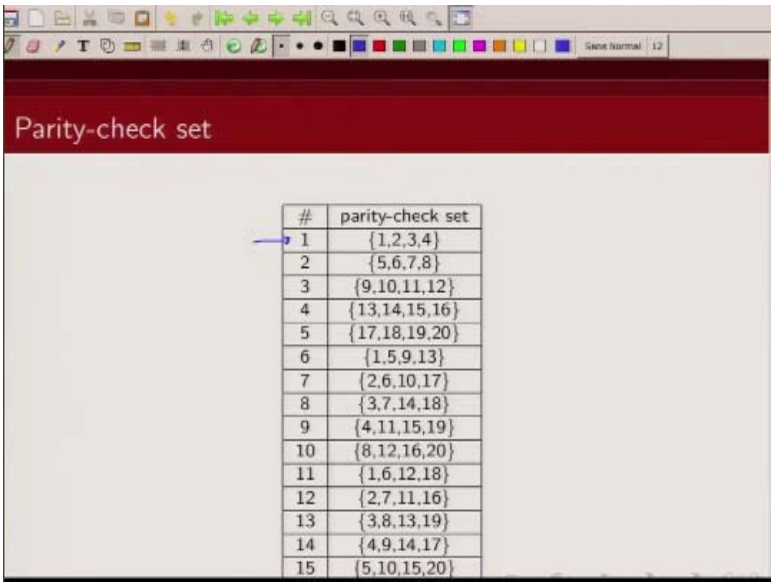
So this is how I am drawing my parity-check set tree.

(Refer Slide Time: 05:52)



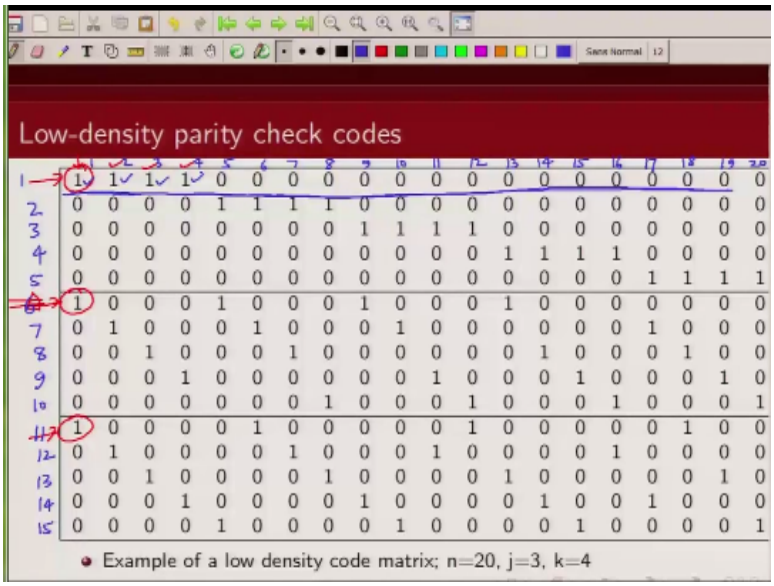
So again pay attention to this parity-check matrix. Let us label each of them like this, let us say 1, 2, 3, 4, 5, 6, 7, let us just label these columns, so that way it will be easier for us to refer to them. Similarly I am labeling these rows. So you can see there will be 15 parity check sets, each corresponding to each of the rows okay. So let us look at the parity check set.

(Refer Slide Time: 06:50)



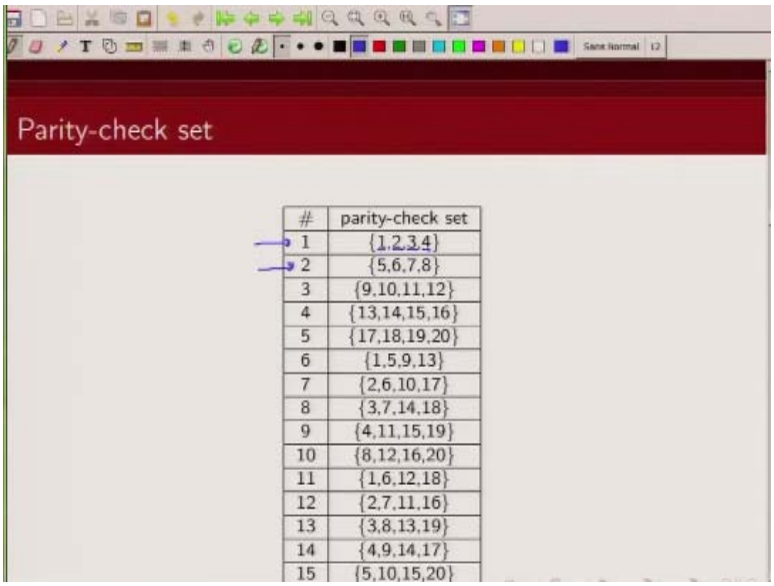
So let us first look at this first parity-check set.

(Refer Slide Time: 06:58)



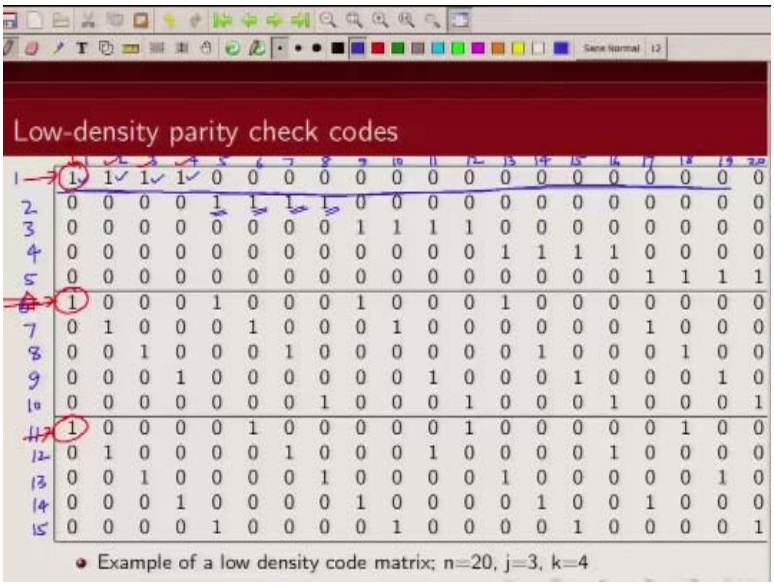
Which corresponds to this first row. So note here book number 1, 2, 3, and 4, these are participating in the parity-check equation.

(Refer Slide Time: 07:11)



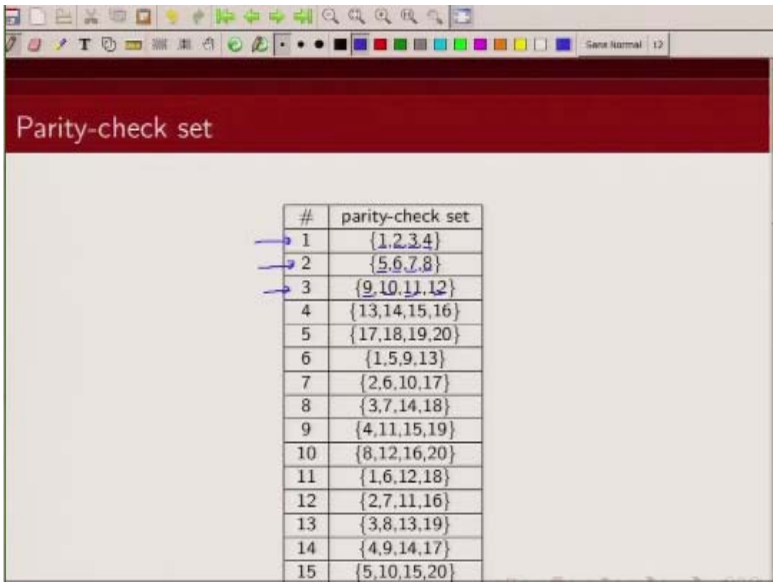
So that is why this first parity-check consists of 1, 2, 3, and 4. Similarly parity-check set 2.

(Refer Slide Time: 07:23)



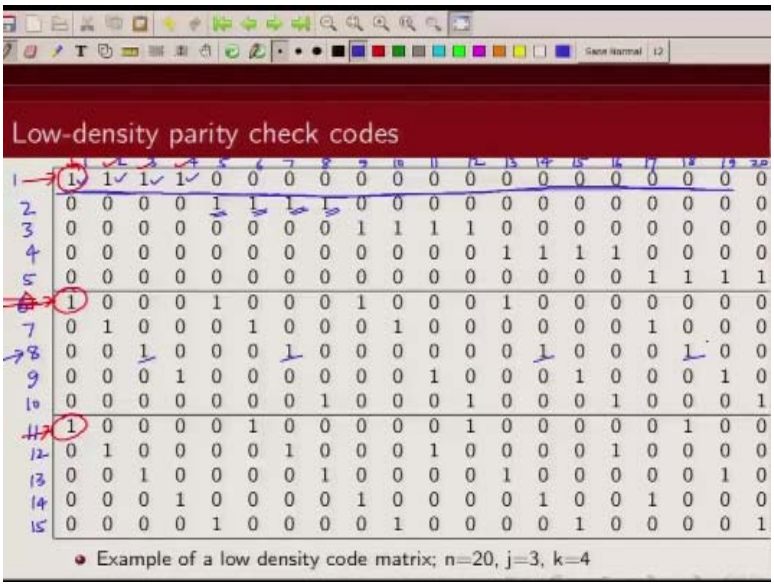
If you look at second parity check equation this bit number 5, bit number 6, bit number 7, bit number 8 are participating.

(Refer Slide Time: 07:34)



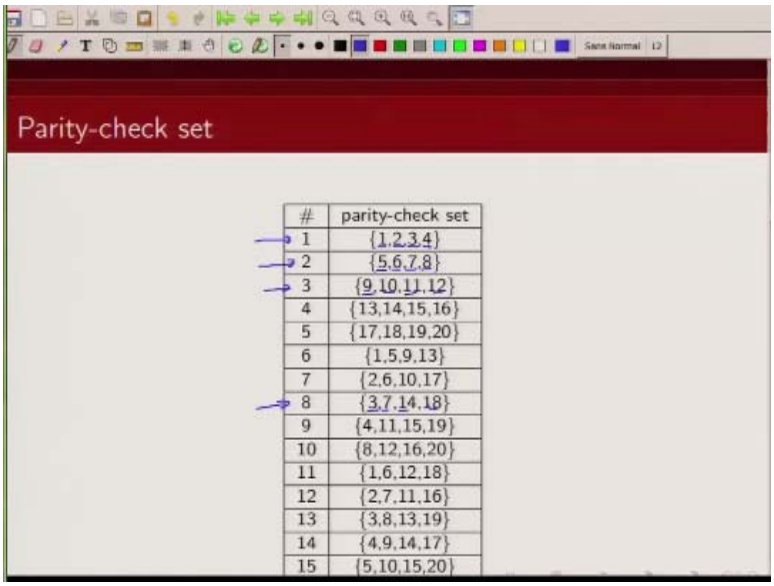
So then parity-check set will have 5, 6, 7 and 8. Similarly parity-check set third has 9, 10, 11, 12.

(Refer Slide Time: 07:47)



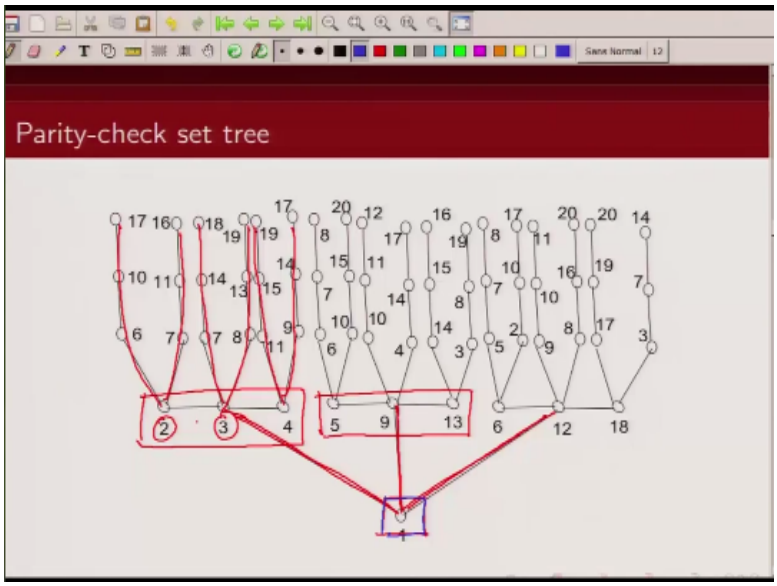
So we can take any example, let us just take this one, 8th one, bit number 3, 7, 14 and 18, 3, 7, 14 and 18 these are participating in the parity-check equation.

(Refer Slide Time: 08:04)



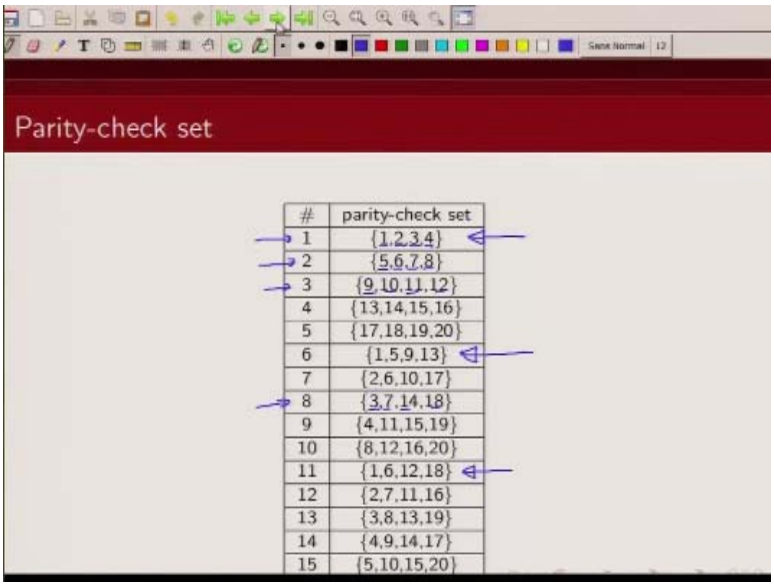
So bit number 3, 7, 14 and 18. So this is how for each of the parity-check equations we create this parity-check set, so there are 15 such parity-check set for this particular example.

(Refer Slide Time: 08:24)



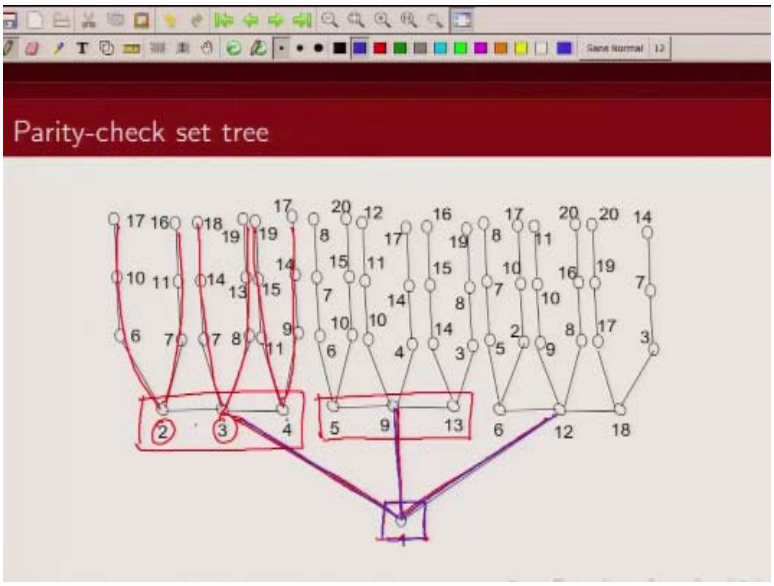
And how do we draw the parity-check set tree, as I said we pick one bit, let us say, I picked bit number 1.

(Refer Slide Time: 08:36)



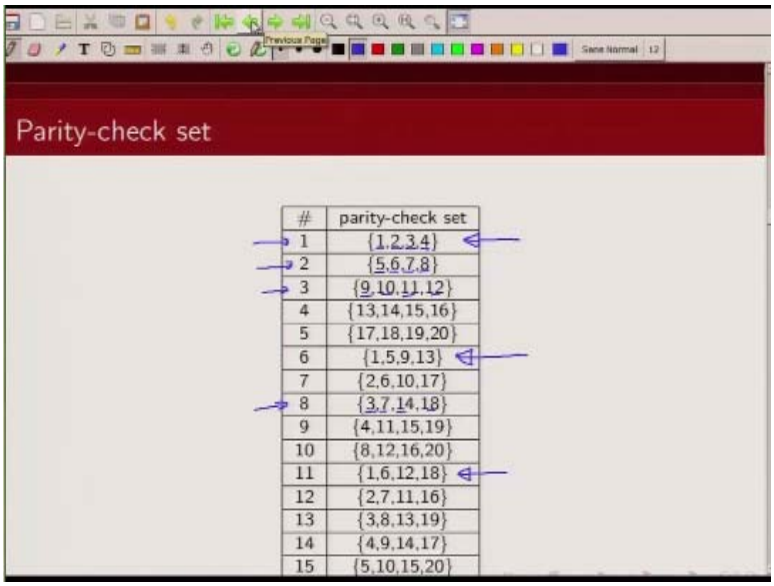
Now bit number 1 appears in which parity check set, how many parity-check equations, look here, bit number 1 appears 3 bits, bit number 1 appears here, bit number 1 appears here, that is it. It appears in these 3 parity-check sets. So we are going to draw three lines corresponding to each of these parity-check sets.

(Refer Slide Time: 09:07)



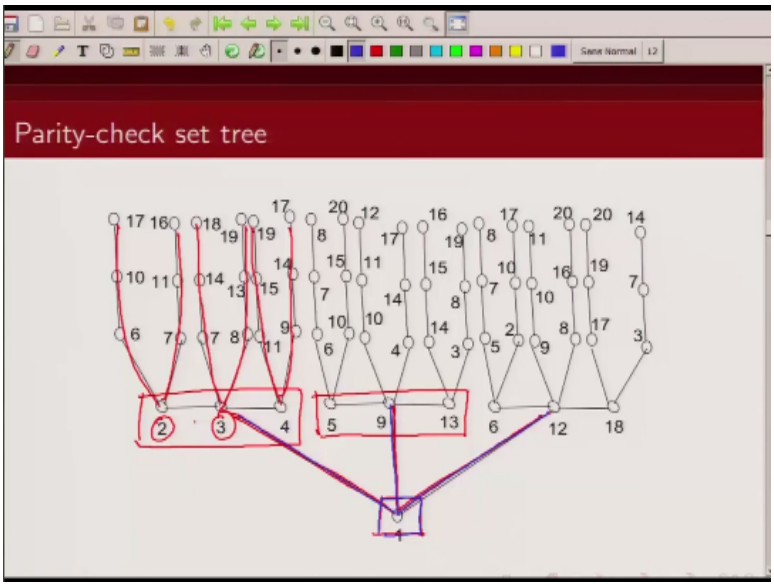
So that is what we have done, this is 1 line, this is another line, this is another line. Now next what we have done is we have written all the nodes that participate in the parity-check set.

(Refer Slide Time: 09:21)



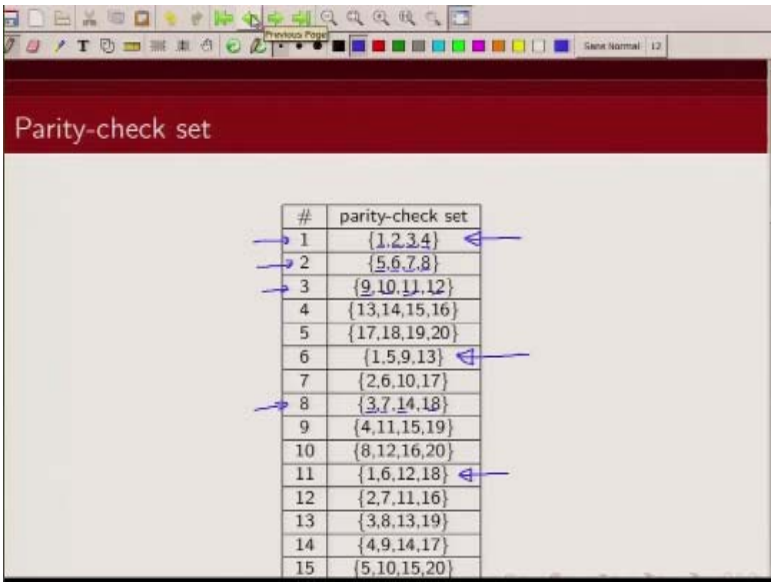
So if you look at this one, in addition to 1, the other bits are 2, 3, and 4.

(Refer Slide Time: 09:28)



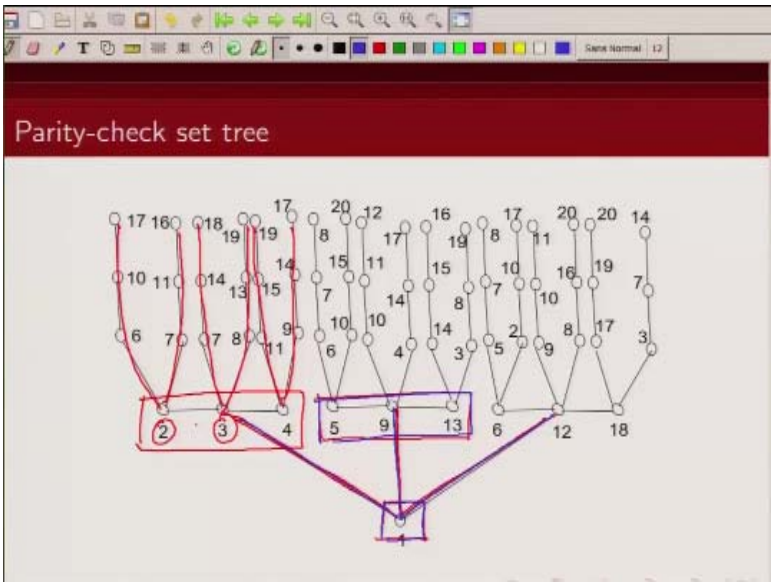
So that we are writing like this, 2, 3, and 4.

(Refer Slide Time: 09:31)



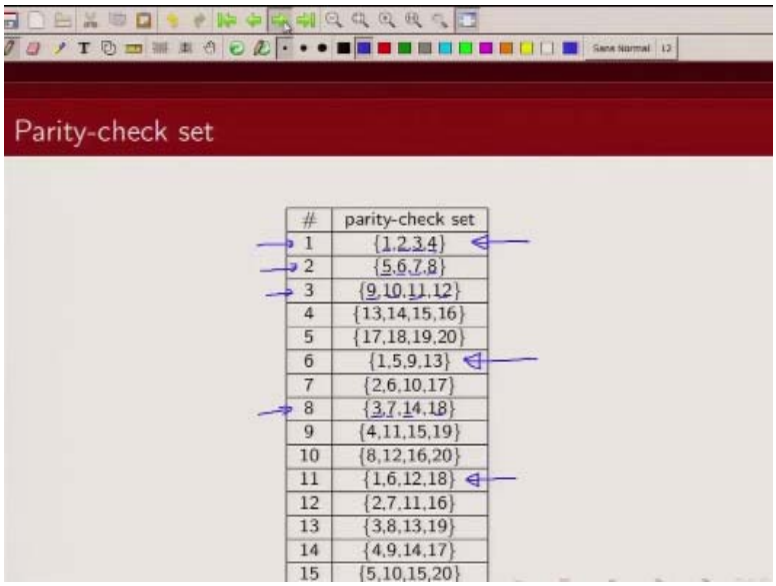
Similarly here bit 5, 9 and 13 are participating in addition to bit number 1.

(Refer Slide Time: 09:41)



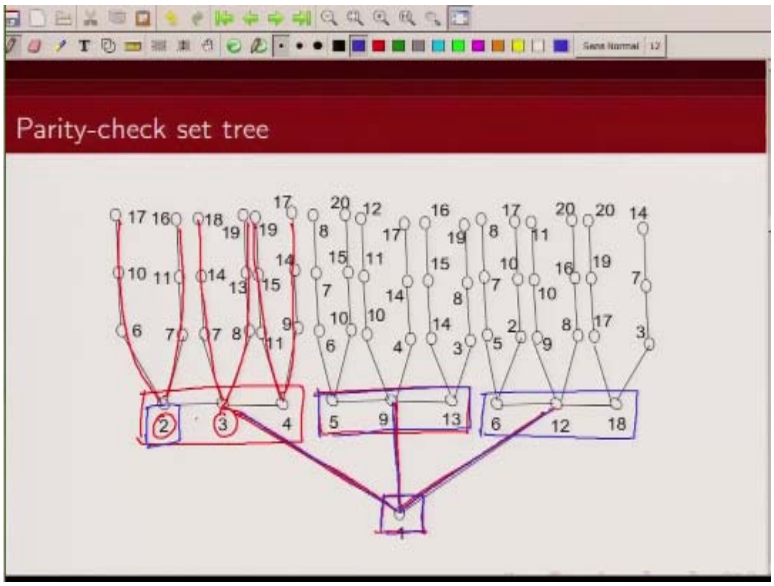
So these are 5, 9, and, 13.

(Refer Slide Time: 09:46)



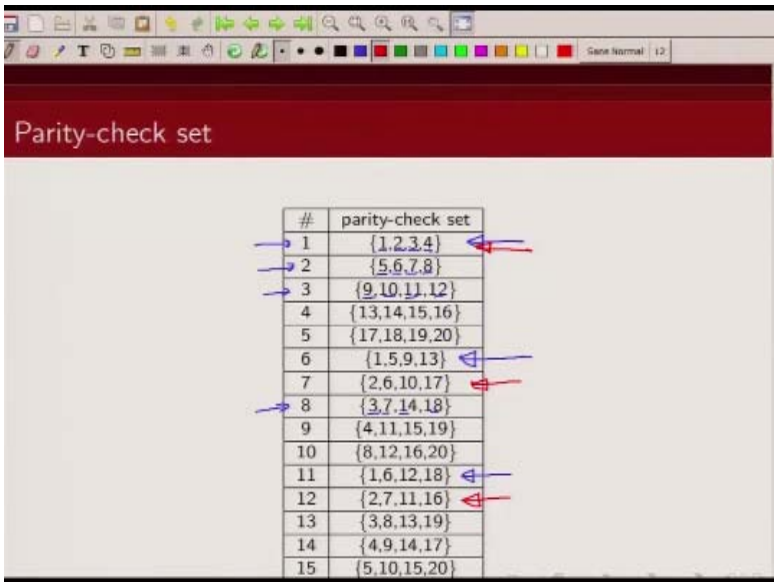
And here 1, 6, 12, and 18 are participating.

(Refer Slide Time: 09:51)



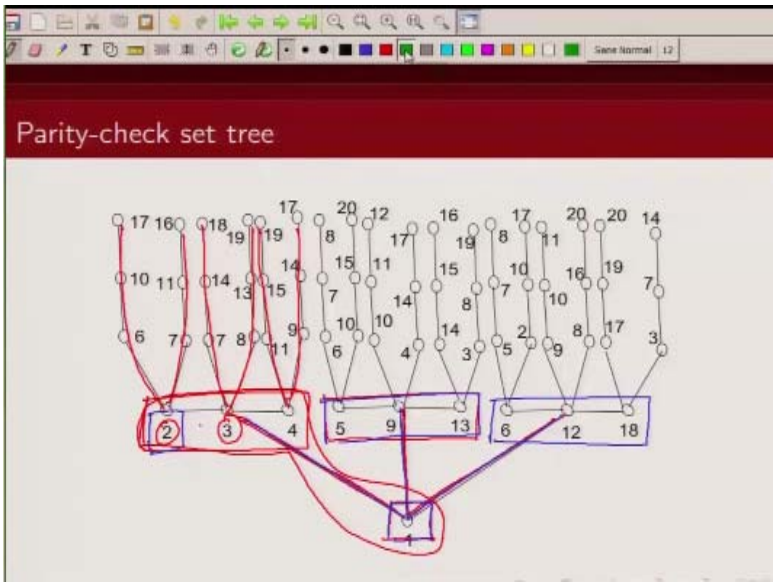
So then we have 6, 12, and 18. So this is how a tier 1. Now how do we draw a tier 2, now you can think of this, look at this.

(Refer Slide Time: 10:04)



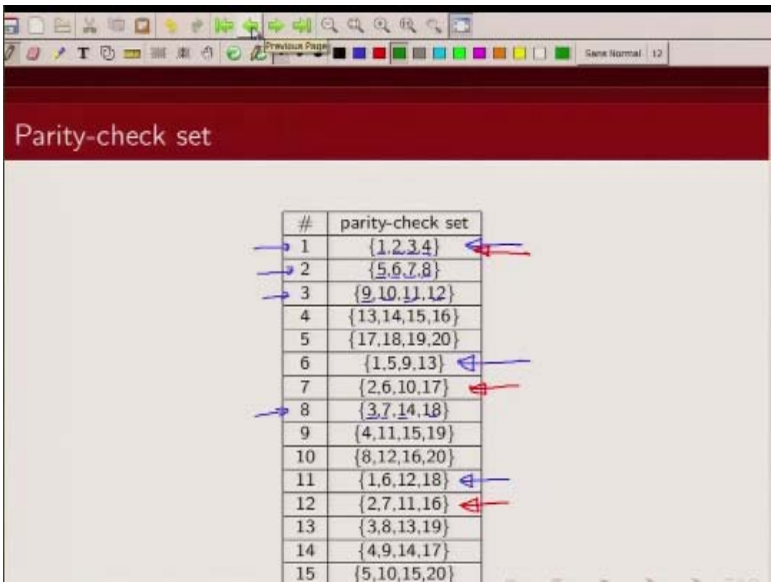
Now 2 appears in which, 2 appears in parity-check set 1, 2 appears in parity-check set 7, 2 appears in parity-check set 12 right.

(Refer Slide Time: 10:22)



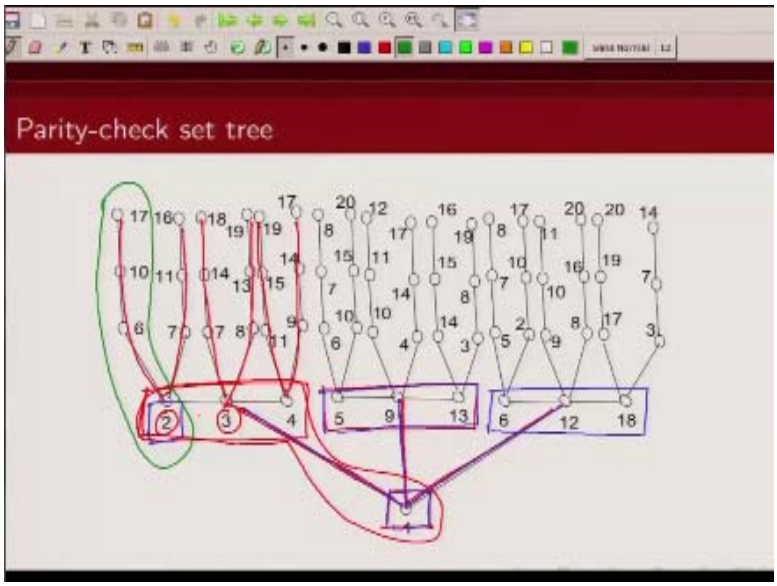
Now this 2 appears in parity-check set 1 that is already captured here, this is already captured here that 2 appears in parity-check set 1.

(Refer Slide Time: 10:35)



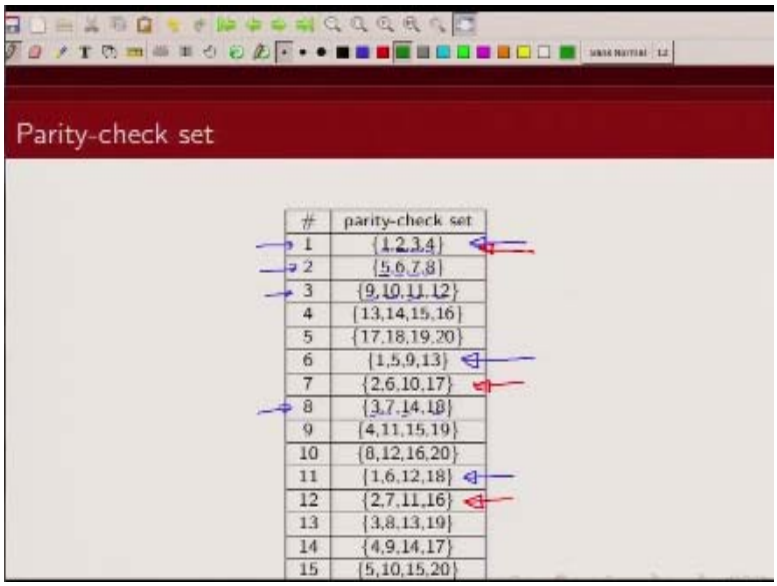
So what are the other two parity-check set, this is 1 is this other is this. So 2 appears with 6, 10, and 17.

(Refer Slide Time: 10:45)



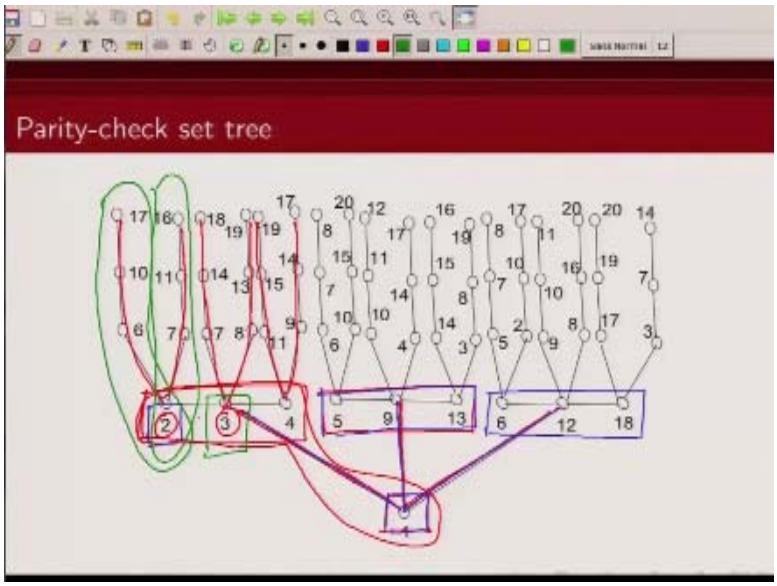
How do we show that, so we are showing this by this particular edge.

(Refer Slide Time: 10:53)



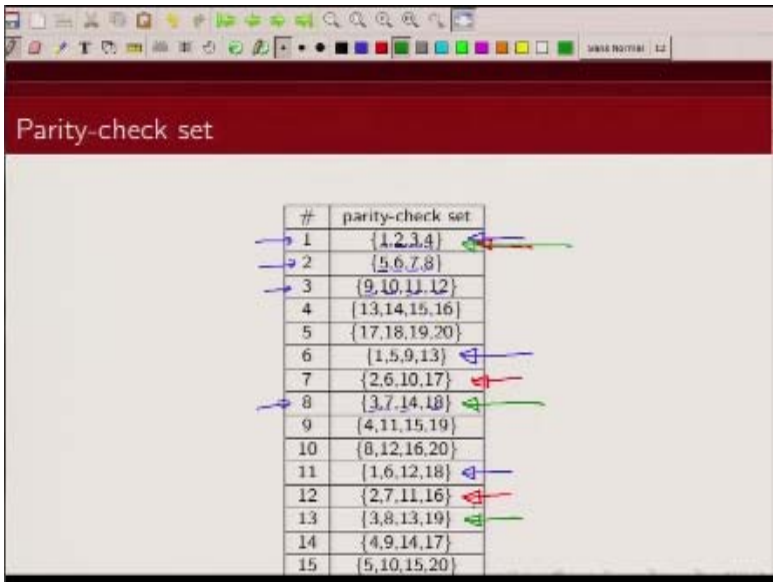
How do we flow this parity check set 2, 2 appears with 7, 11, and 16, how do we show that?

(Refer Slide Time: 11:00)



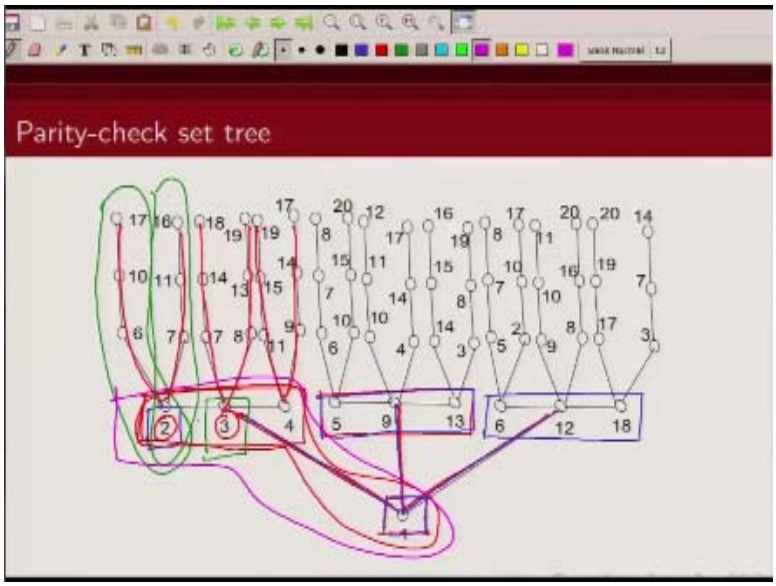
We show that moving this. Similarly we do the same thing for other bits, so for example bit number 3.

(Refer Slide Time: 11:15)



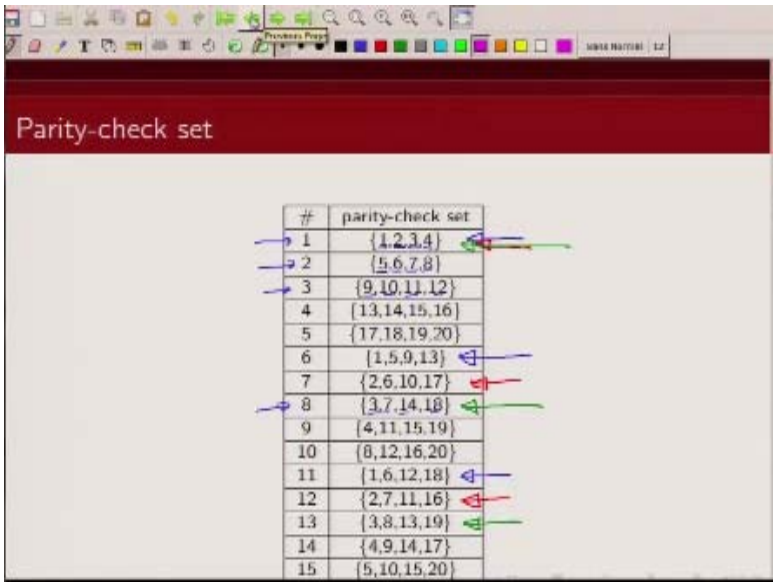
Now look at bit number 3, bit number 3 appears in parity-check set 1, it appears in parity-check set 8, it appears in parity-check set 13. Now this parity-check set 1 that is already captured.

(Refer Slide Time: 11:36)



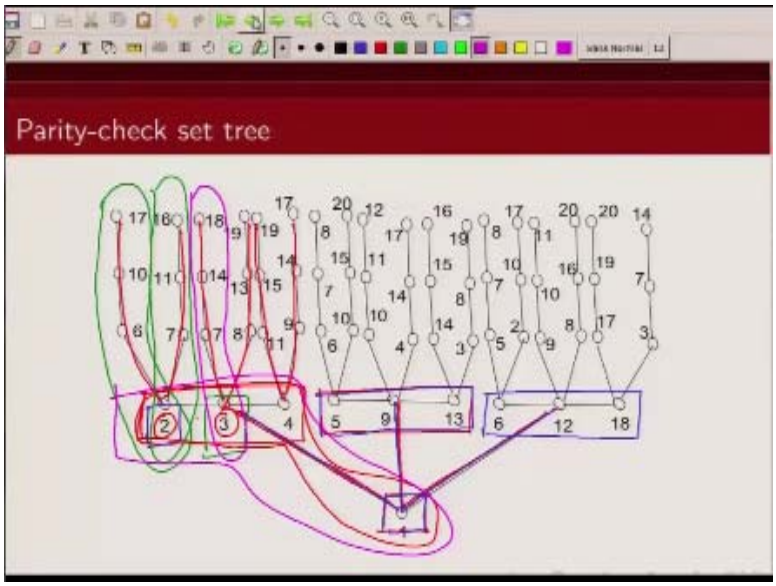
Because that is this one, it is already captured.

(Refer Slide Time: 11:43)



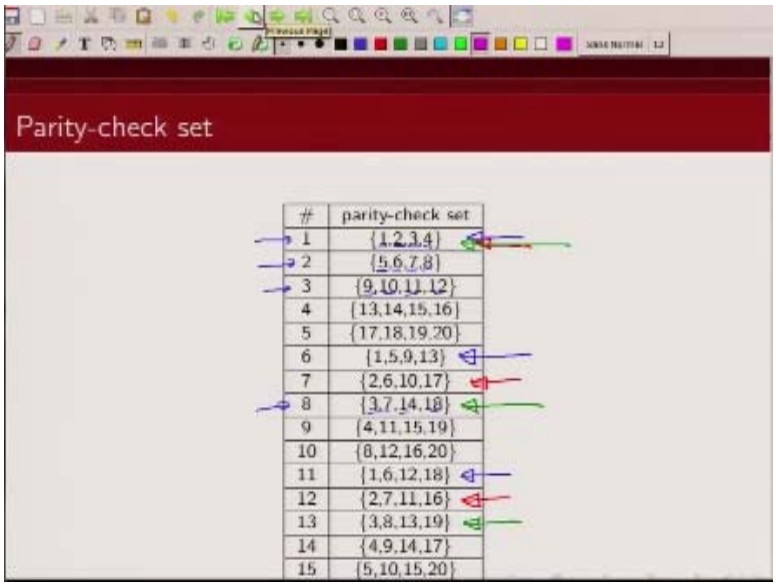
So what are the other two parity-check sets, the one involving 3, 7, 14, and 18.

(Refer Slide Time: 11:50)



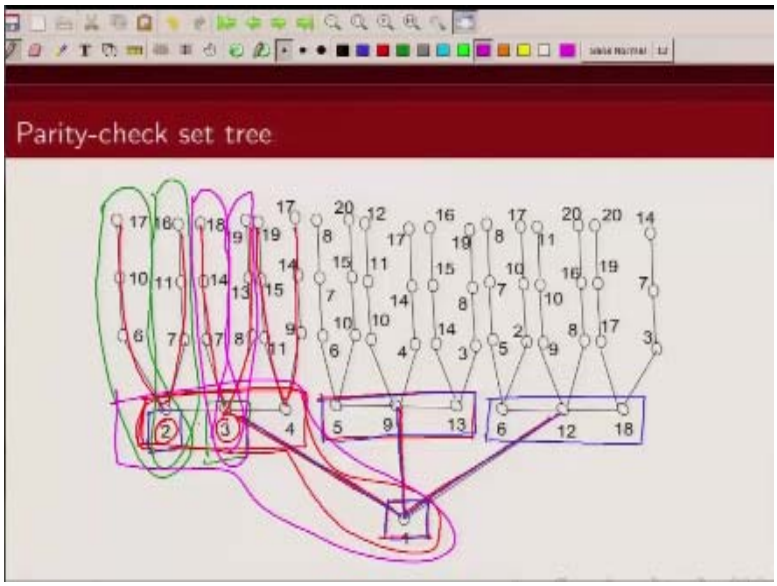
So this is 3, 7, 14, and 18, that is this one.

(Refer Slide Time: 12:00)



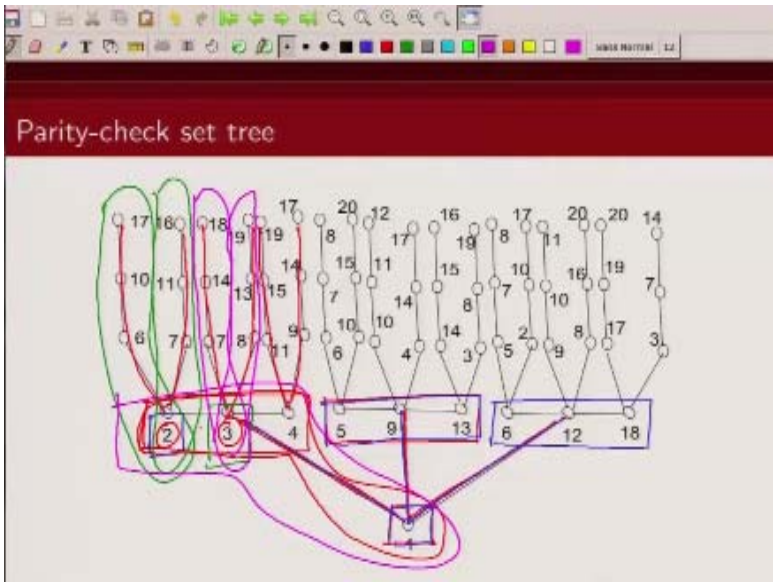
And the other one is 3, 8, 13, and 19.

(Refer Slide Time: 12:05)



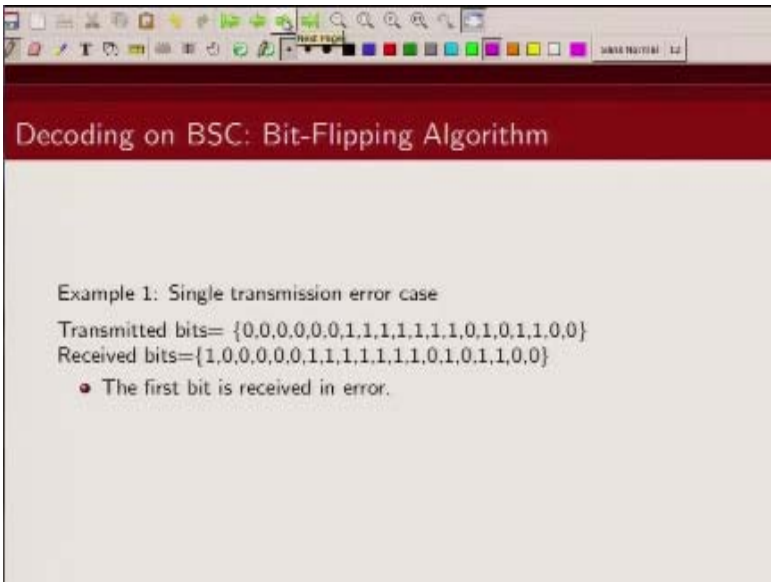
So this is this one 3, 8, 13, and 19 okay. So we are basically connecting by edges all these parity-check sets. So that is how we are representing parity-check set tree. Now we can do this with other bits as well.

(Refer Slide Time: 12:27)



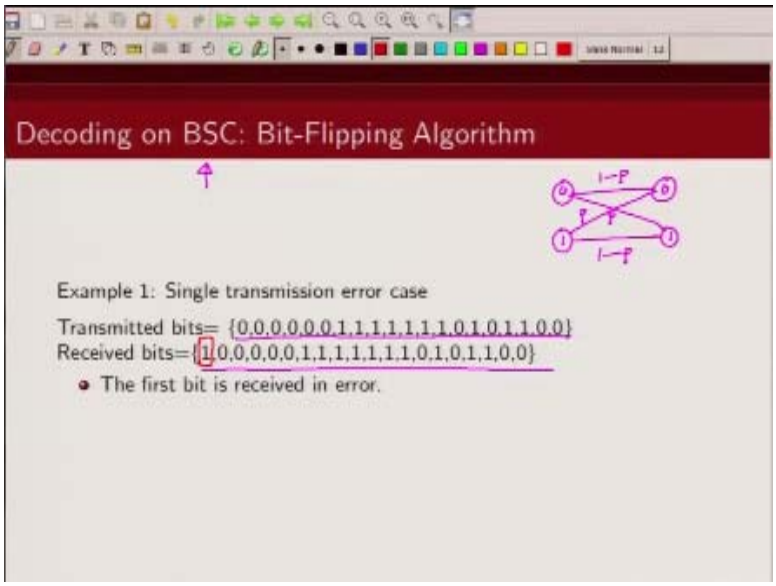
We can for example instead of making 1, if I can make this as 2 and construct a tree around this node 2, same procedure.

(Refer Slide Time: 12:35)



Now let us look at how we can correct error.

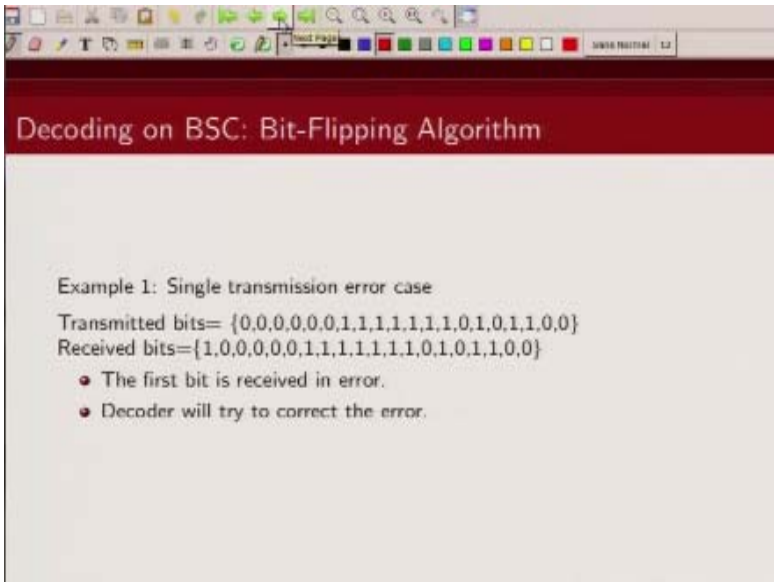
(Refer Slide Time: 12:42)



So we are considering a binary symmetric channel, again recall what is a binary symmetric channel. So there are two inputs 0 and 1, 0 and 1 with probability 1-P you receive the bits correctly and there is a crossover probability of bits getting flipped. So let us consider that we have transmitted this information, this we have transmitted this coded sequence and what we received is this.

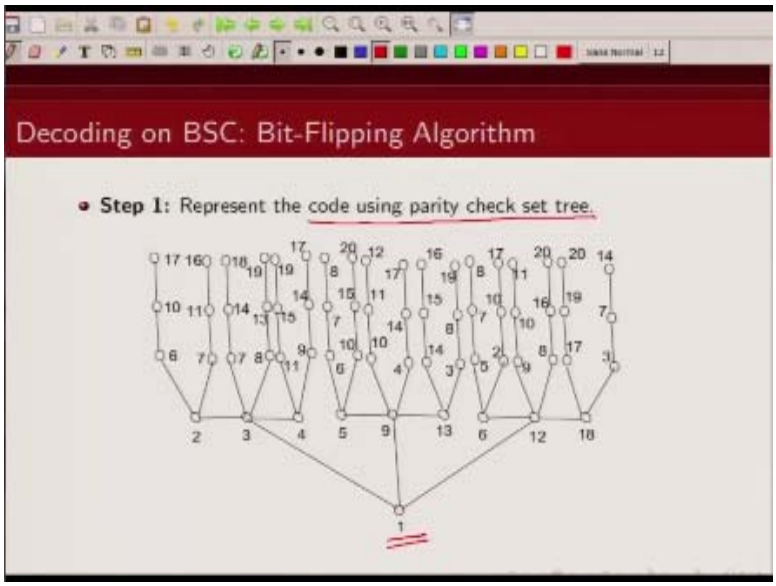
So there is an error in the first bit location. Now how do we correct this error, so to decode this what we are going to do is, we are going to construct a parity-check set tree around each of these bits and use that for our decoding purpose.

(Refer Slide Time: 13:37)



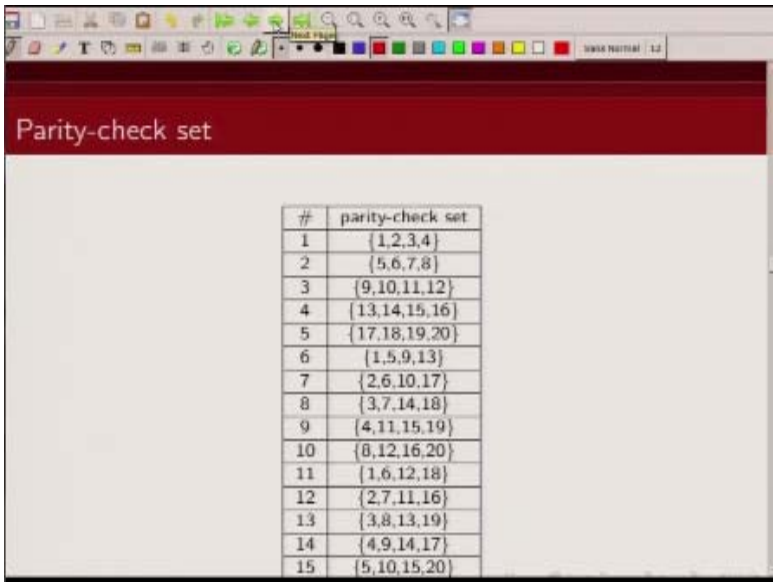
So let us see how we do that.

(Refer Slide Time: 13:40)



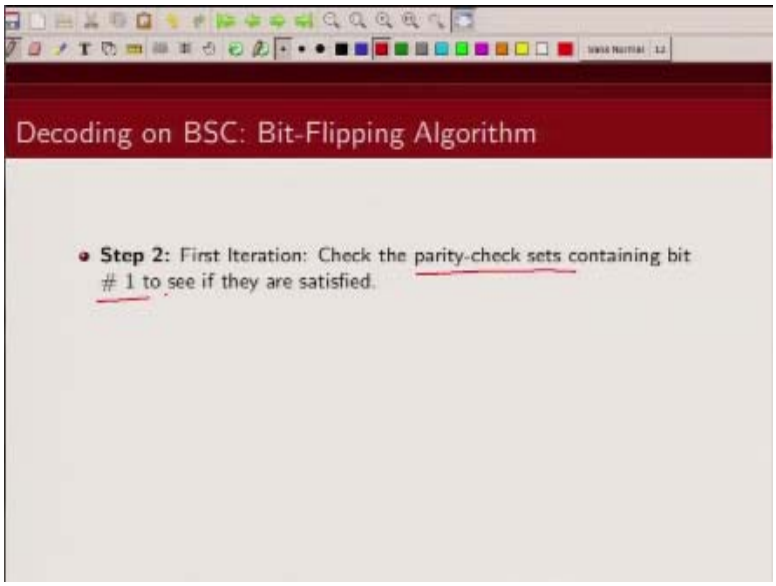
So the first step is we construct a – we represent the code using parity-check set tree and we have explained in the previous slide, how this parity check set tree is constructed. So this is a parity check set tree for the bit number 1.

(Refer Slide Time: 13:59)



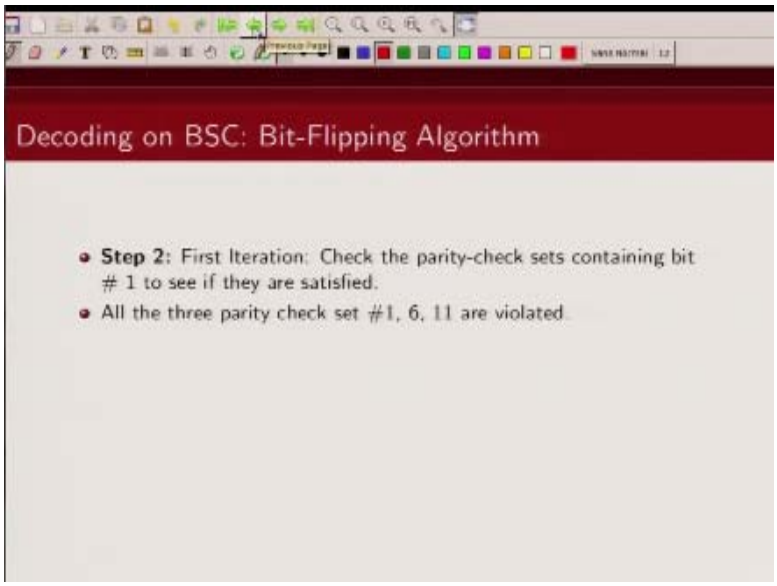
And remember this is the parity check set, corresponding to this we have drawn this parity check set tree.

(Refer Slide Time: 14:08)



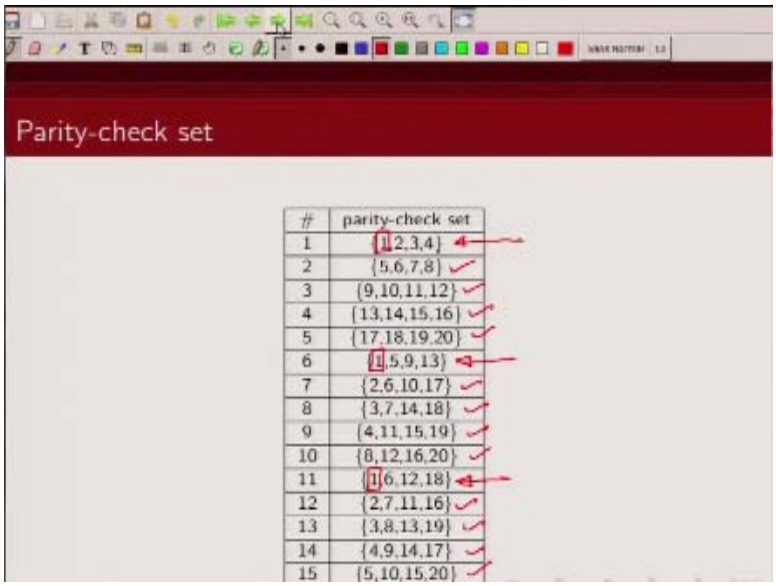
Now what we are going to see first check is, whether all the parity-check sets containing bit number 1 if they are satisfied. If they are satisfied it is likely that bit number 1 is received correctly. If majority of them are not satisfied it is likely that bit number 1 is an error.

(Refer Slide Time: 14:32)



So let us see, now which are the parity-check sets in which bit number 1 is participating?

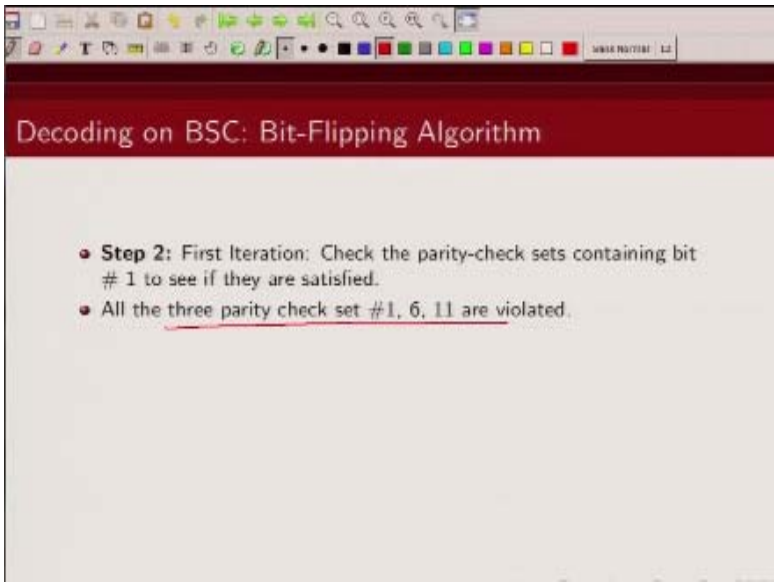
(Refer Slide Time: 14:39)



That is this, this one, and this one. Now note in our example there was single error in bit number one location so all other bits were received correctly only bit number 1 was an error, then what is going to happen? This parity-check set would not be satisfied, because this bit is an error. This will be satisfied, this will be satisfied, this will not be satisfied, because this particular bit was an error.

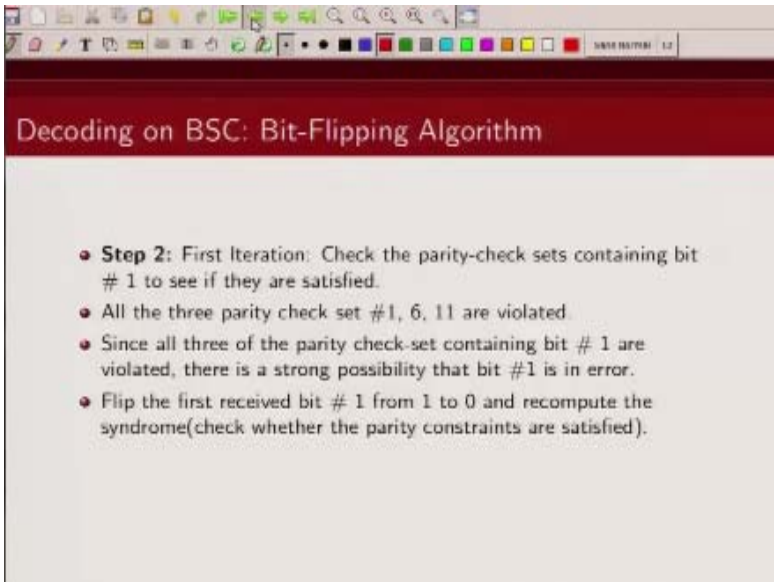
These are all satisfied, this will not be satisfied, again these are all satisfied. So you can see all the three parity-check sets involving bit 1 are not satisfied in this particular example.

(Refer Slide Time: 15:32)



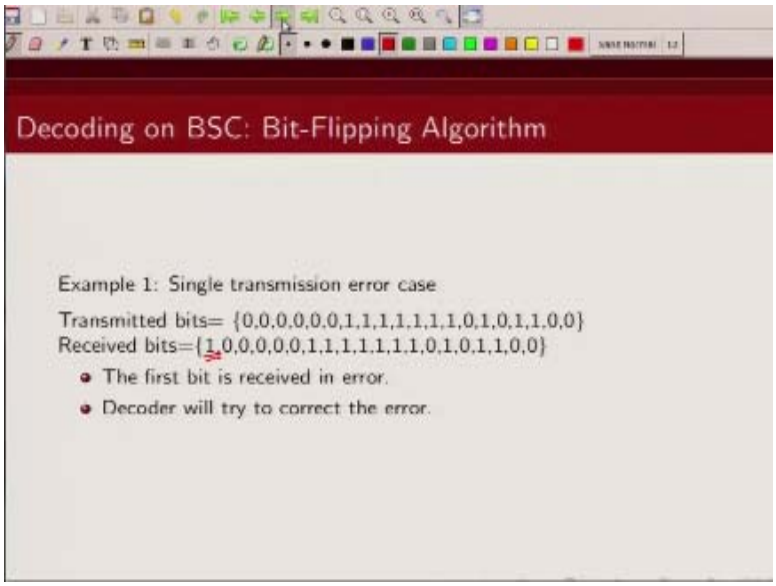
So all the parity-check sets containing 1, 6 and 11 are violated, now what does that mean, it means that there is a very large likelihood of this particular bit being received in error.

(Refer Slide Time: 15:49)



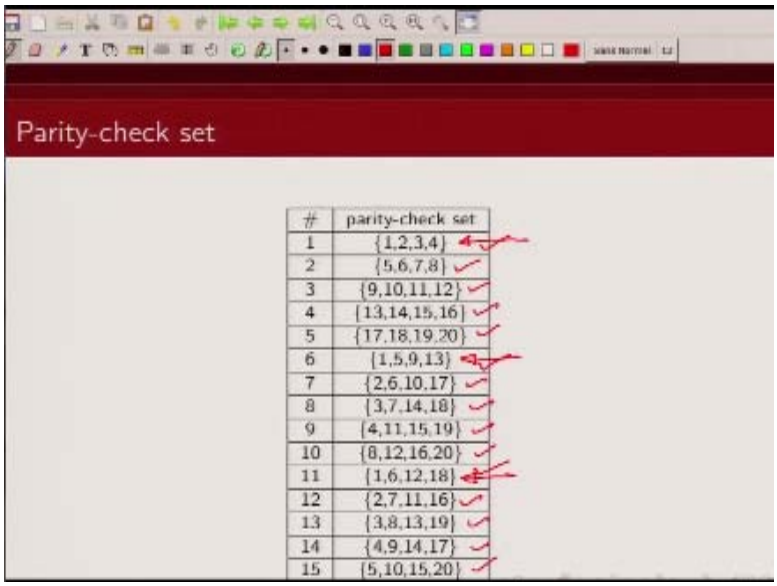
Hence what do we do, then we are going to flip this bit 1, whatever this bit was, we are going to flip it and then again check the parity check constraints.

(Refer Slide Time: 16:11)



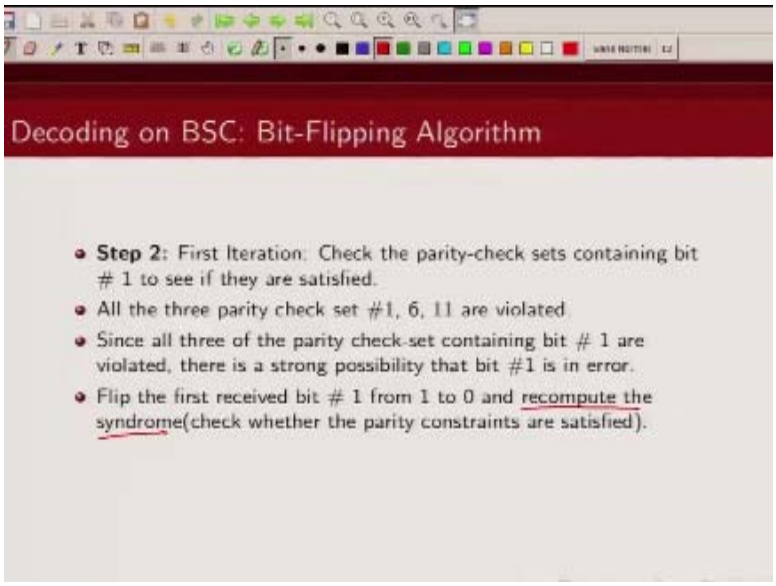
So earlier this bit was received as 1, we are going to flip it to 0 and again try to check the parity-check equations.

(Refer Slide Time: 16:18)



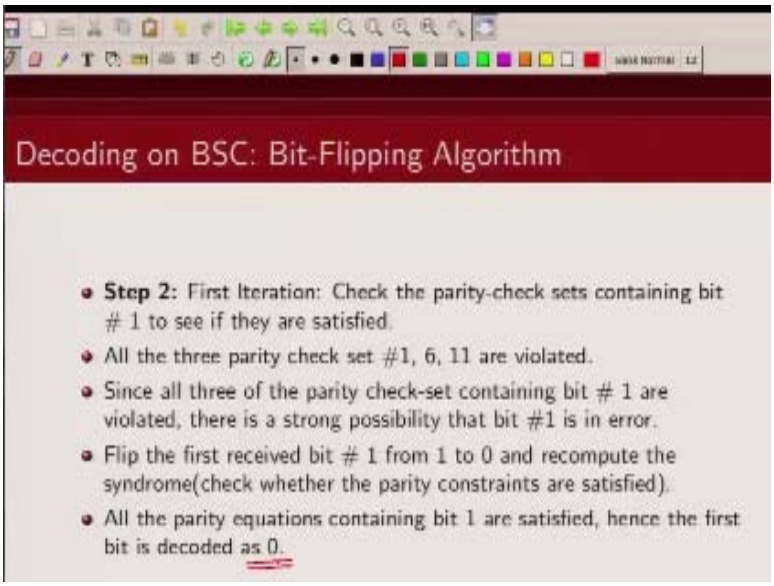
Now note that when we flip this bit, this bit is now no longer in error, these bits are no longer in error, so then these parity-check constraints will also be satisfied. Hence we are able to correct single error.

(Refer Slide Time: 16:38)



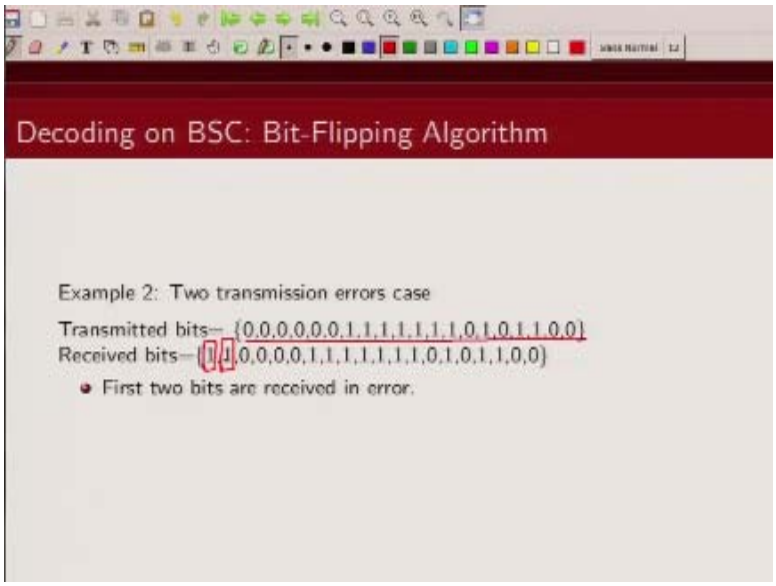
So when we recompute the syndrome we will see that all the parity-check constraints are satisfied, because there was only single error.

(Refer Slide Time: 16:47)



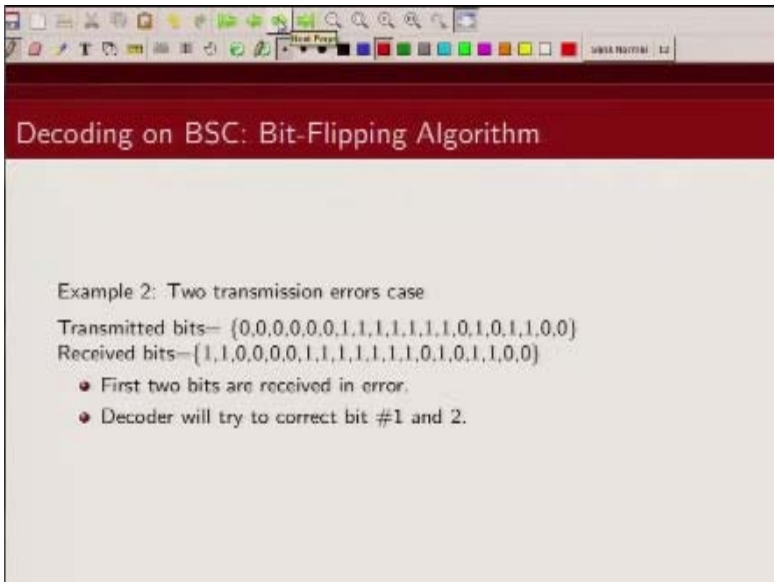
Which we are able to detect and we were able to correct it. So hence the first bit will be decoded as 0 and same procedure we will follow for other bits as well and since there was no error in other bits, so all the parity-check sets involving those bits will already be satisfied. So we will be able to successfully decode it okay.

(Refer Slide Time: 17:10)

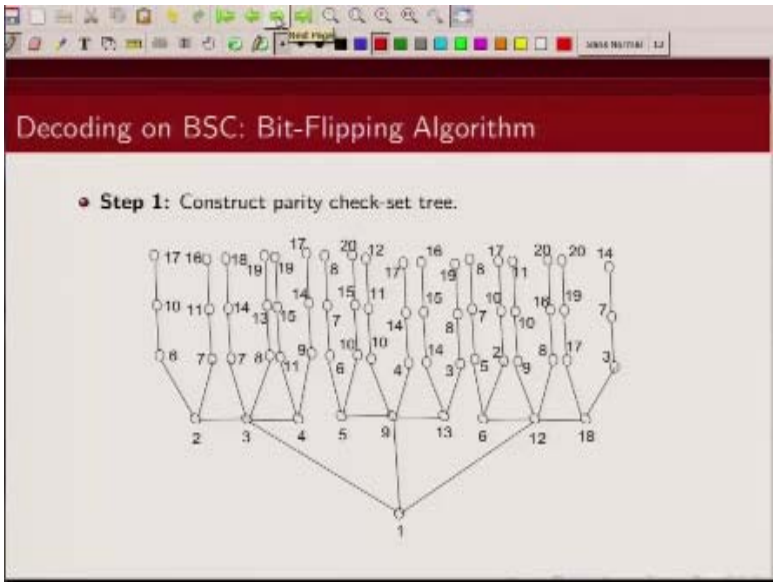


Now let us look at the case when there are two errors. So the same transmitted code word we have considered. In this case now we have considered the two errors, in bit location 1 and bit location 2. Now let us see how our LCPC decoder will be able to decode this.

(Refer Slide Time: 17:32)

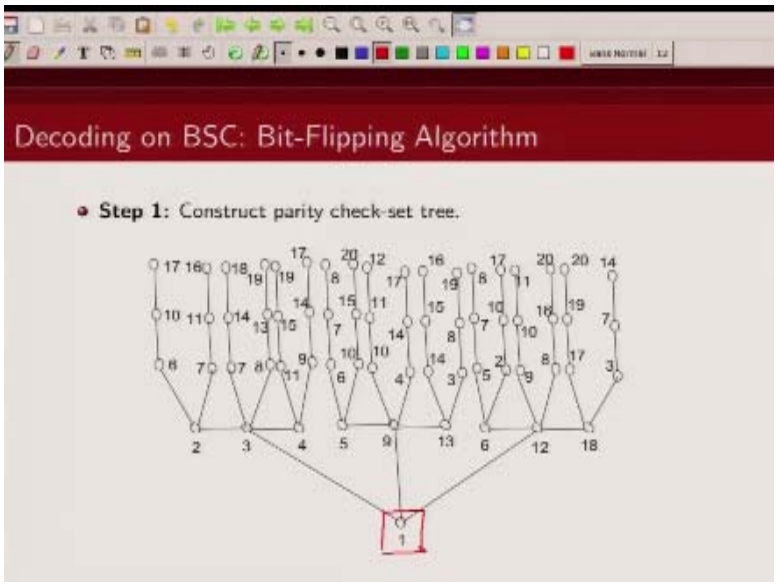


(Refer Slide Time: 17:34)



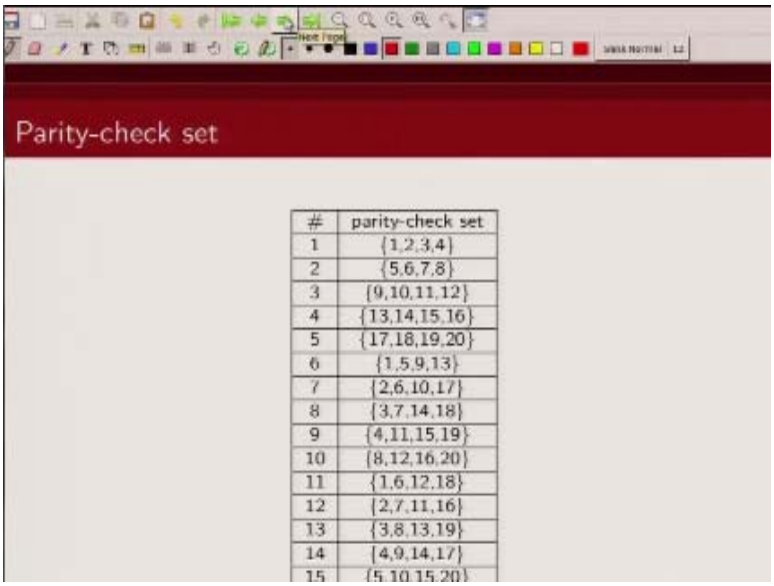
So again we follow the same procedure, we draw the parity-check set tree with each node at its space.

(Refer Slide Time: 17:44)



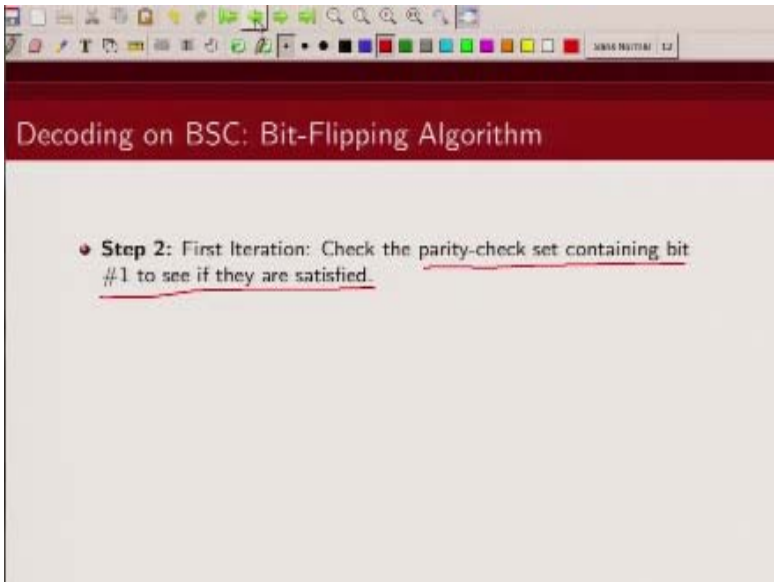
So we start with node number 1, we construct the parity-check set tree.

(Refer Slide Time: 17:50)



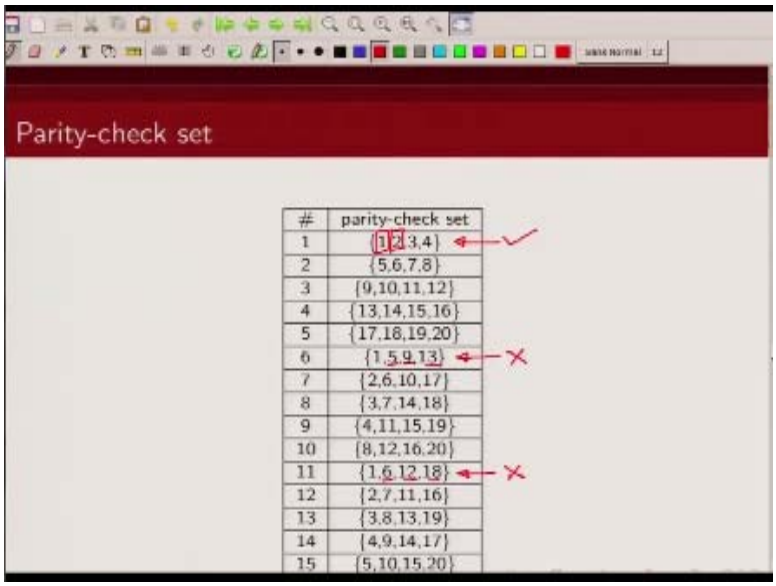
And these are the parity-check set, 15 parity-check sets corresponding to the parity-check matrix given to us.

(Refer Slide Time: 18:03)



Now in the first step what we do is we check the parity-check sets containing bit number 1, and we see if all the parity check constraints are satisfied.

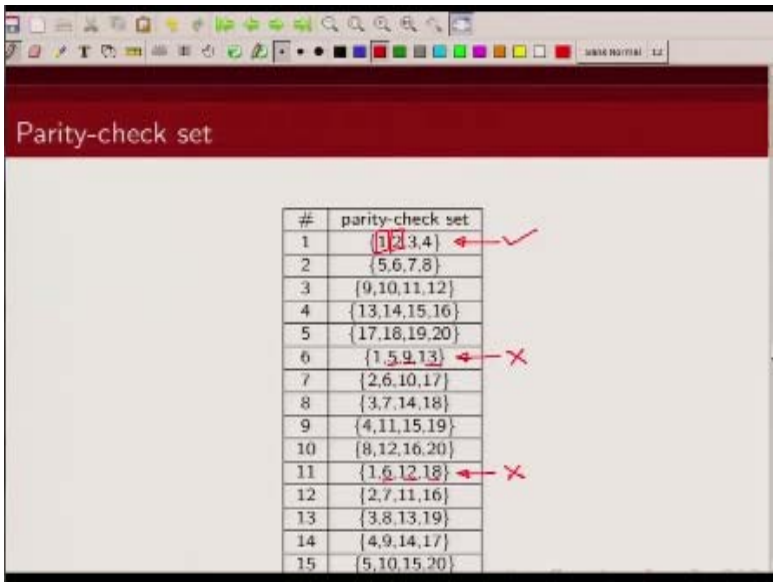
(Refer Slide Time: 18:18)



So what are we going to do, we are going to look at all these parity-check sets which have one in them. So this is our parity-check set 1, 6, and 11. How will this be satisfied? Yes, it will be satisfied, why? Because this was also in error and this was also in error and this was also an error. So this parity-check equation will be satisfied, because two bits are in error okay.

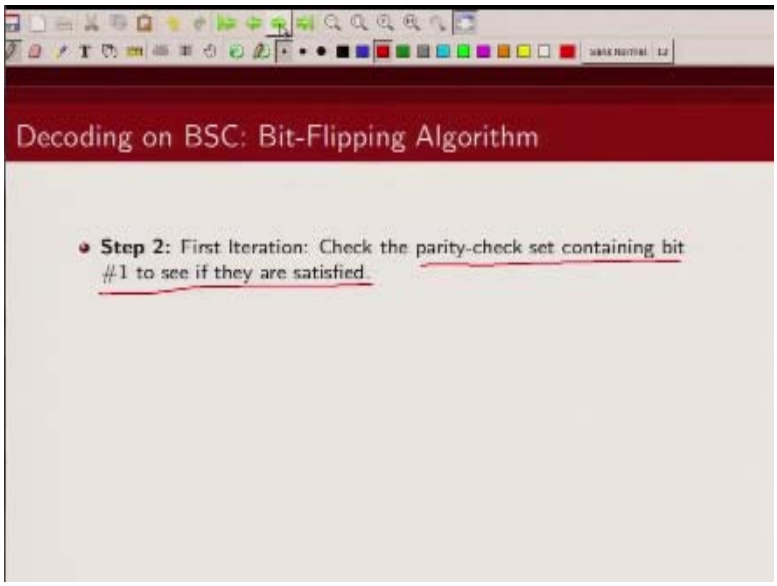
What about this? This parity-check set will not be satisfied, why? Because 5, 9, and 13 were received correctly, but 1 was not received correctly, so this parity-check set will not be satisfied. Similarly here 6, 12, and 18 are received correctly, but 1 is not. So then this parity-check set will not be satisfied. So what we have seen here in the case of double error is, 2 of the parity-check set involving 1 is not satisfied whereas 1 is satisfied.

(Refer Slide Time: 19:28)



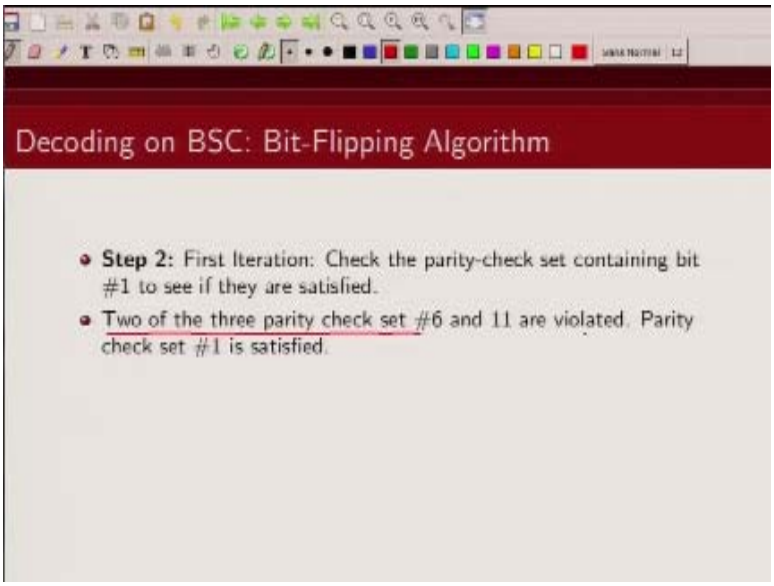
Now what does that tell us? It tell us since majority of them are not satisfied it is likely that bit 1 was in error so we are going to flip it.

(Refer Slide Time: 19:39)



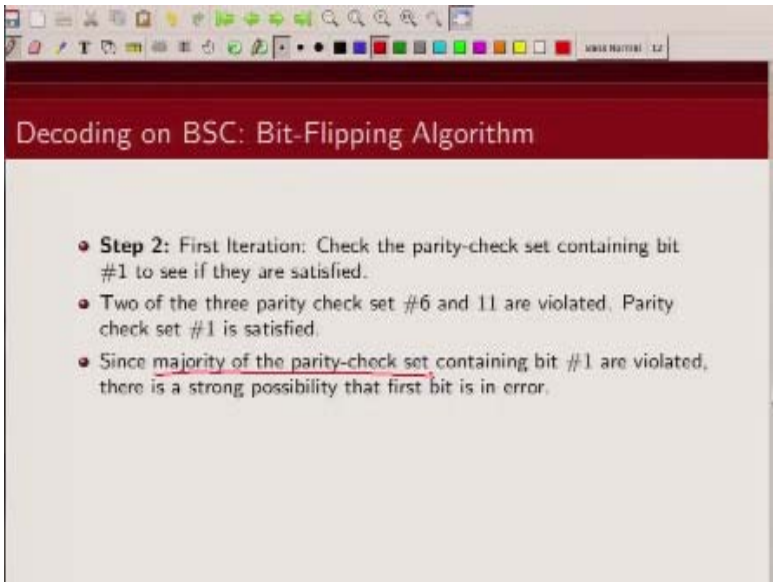
And try to do the same thing again.

(Refer Slide Time: 19:41)



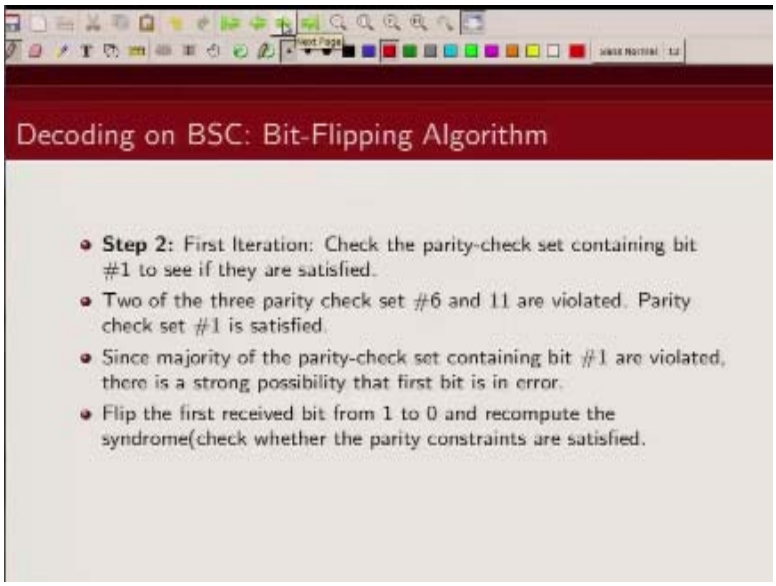
So since two of the parity-check sets are violated.

(Refer Slide Time: 19:50)



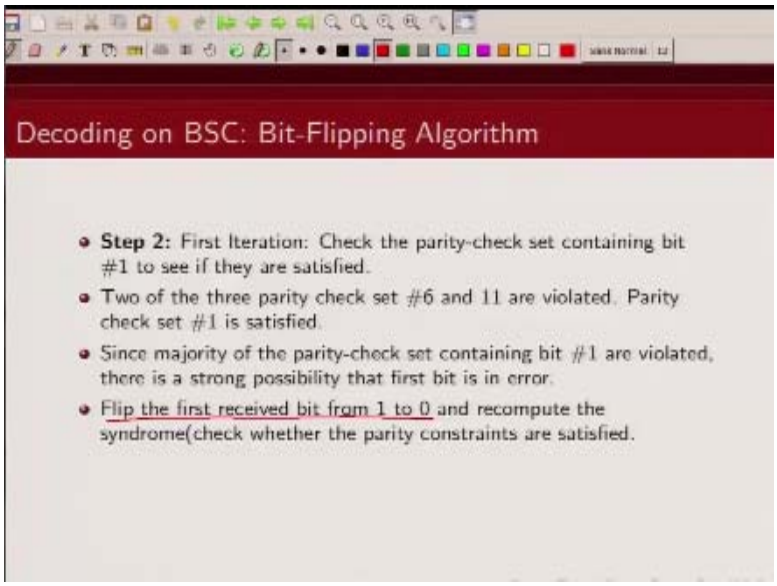
Is likely that bit 1 is in error, because majority of the parity-check set containing 1 are not satisfied.

(Refer Slide Time: 20:00)



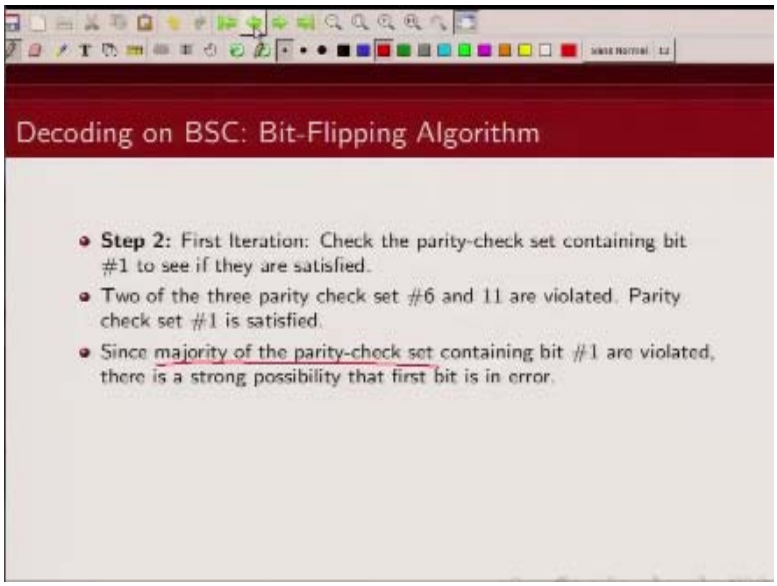
So what do we do if majority of them are saying they are not satisfied, we are going to flip that bit.

(Refer Slide Time: 20:09)

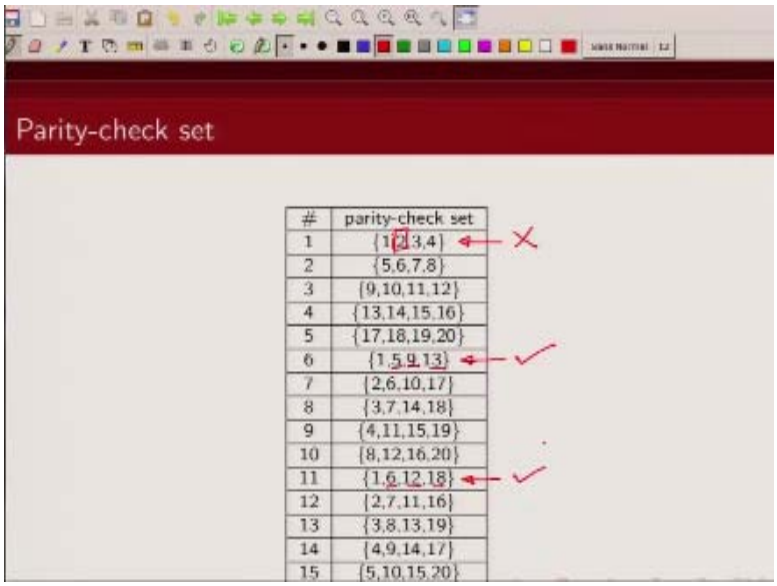


So we are going to flip the first bit from 1 to 0 and again recompute our parity-check constraints. So let us do that.

(Refer Slide Time: 20:16)

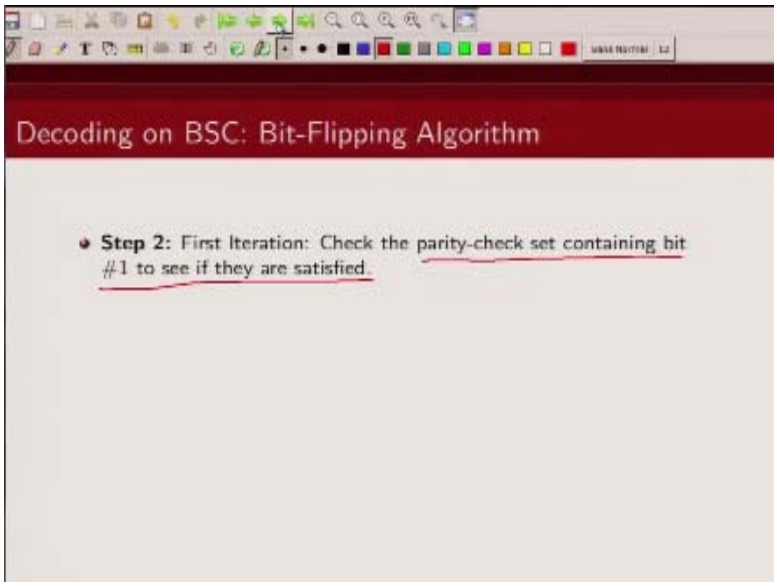


(Refer Slide Time: 20:17)

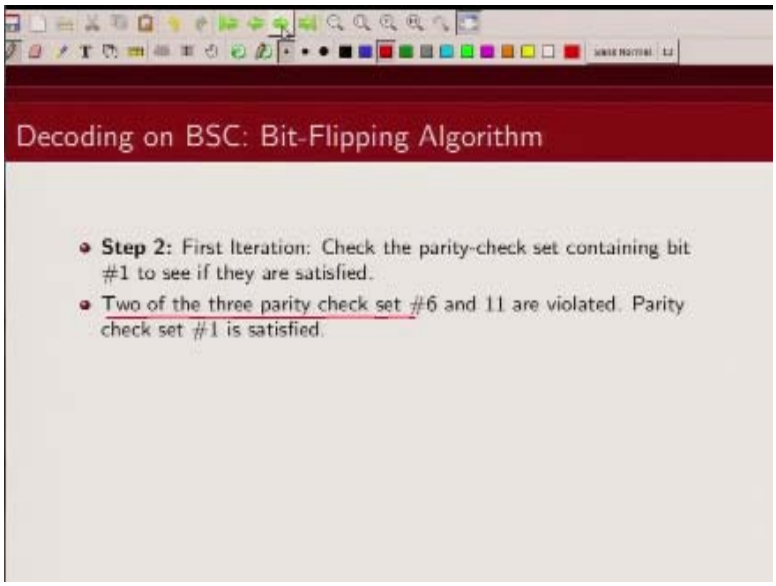


So this bit has been flipped, now if this bit is being flipped what is going to happen? If this bit is flipped now this bit has been corrected, but 2 was in error. So this parity-check set which was earlier getting satisfied is now not getting satisfied, what about this? It is getting satisfied, what about this? It is getting satisfied. So two of them are getting satisfied while one of them is not getting satisfied, so then the first iteration is not enough to decode this bit.

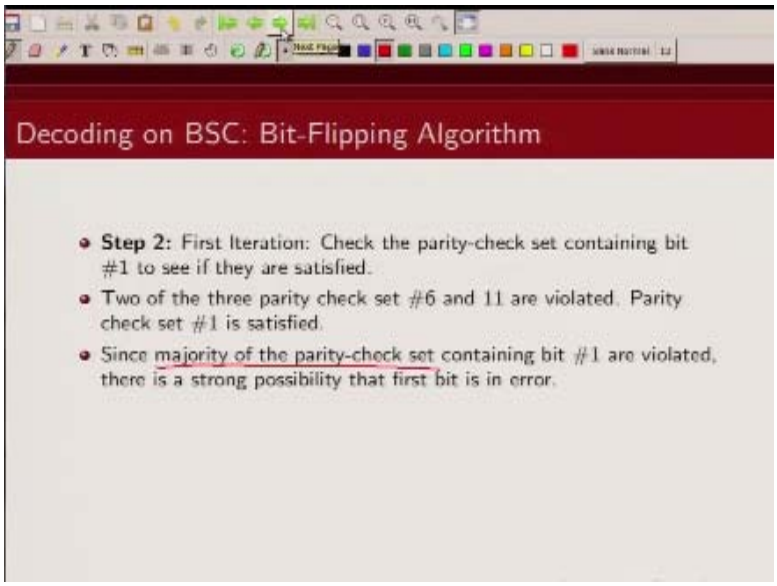
(Refer Slide Time: 20:52)



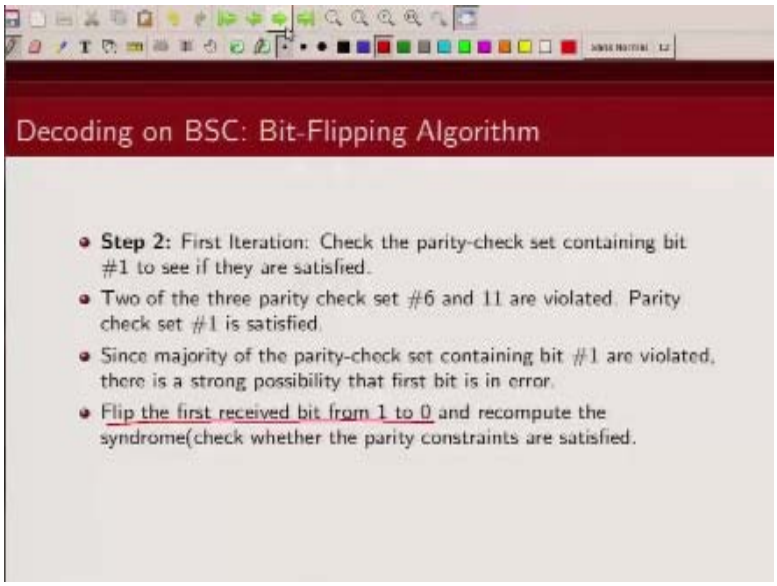
(Refer Slide Time: 20:53)



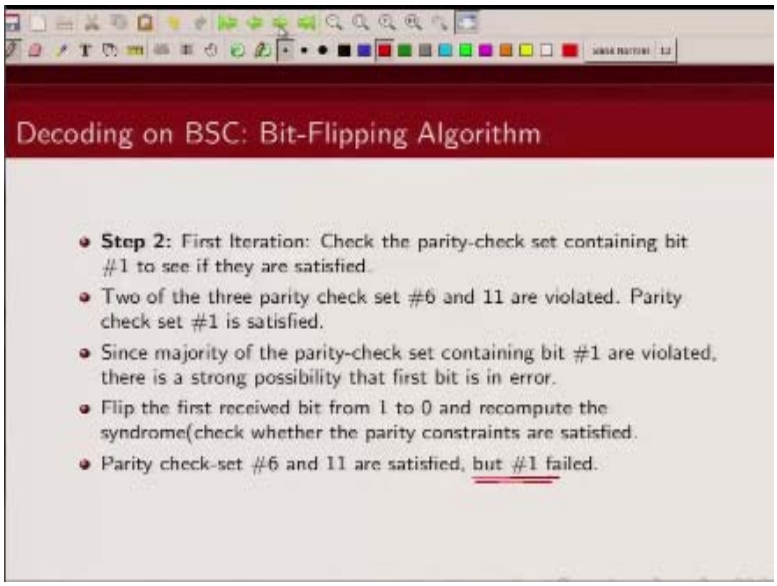
(Refer Slide Time: 20:54)



(Refer Slide Time: 20:55)

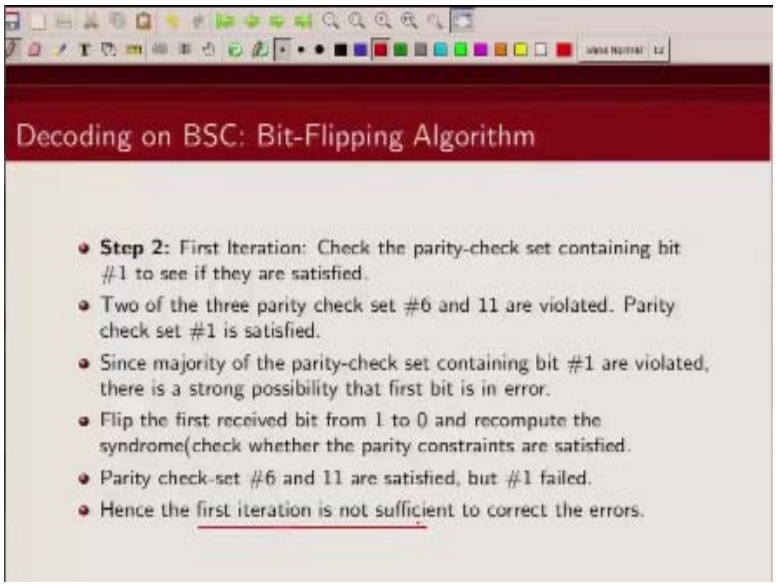


(Refer Slide Time: 20:57)



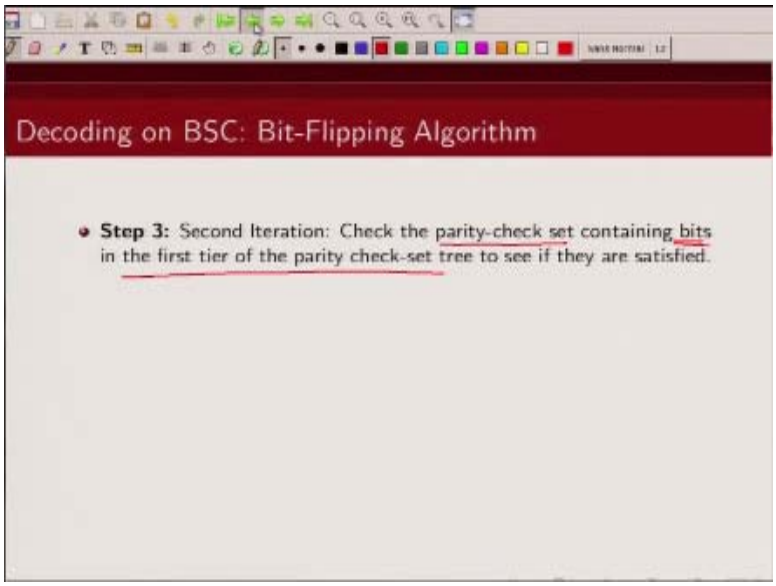
Because parity-check set first fail, there were two single error.

(Refer Slide Time: 21:04)



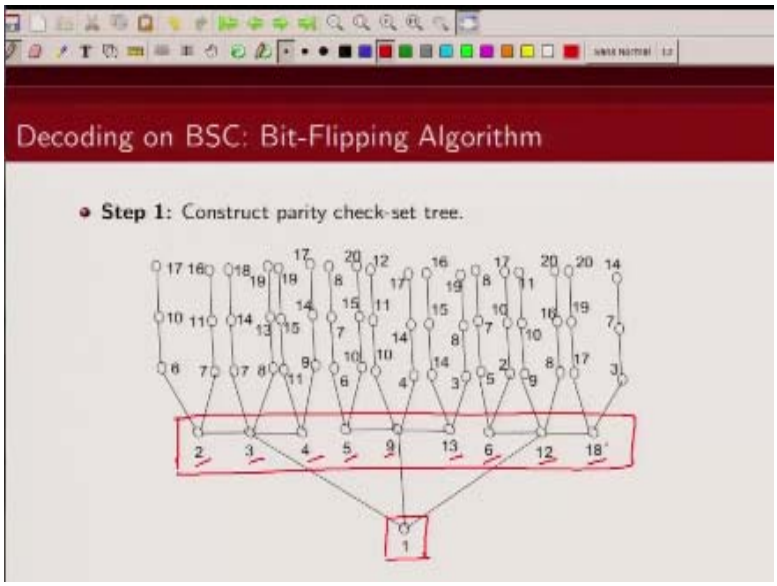
Two errors, so first iteration is not sufficient to correct the errors.

(Refer Slide Time: 21:11)



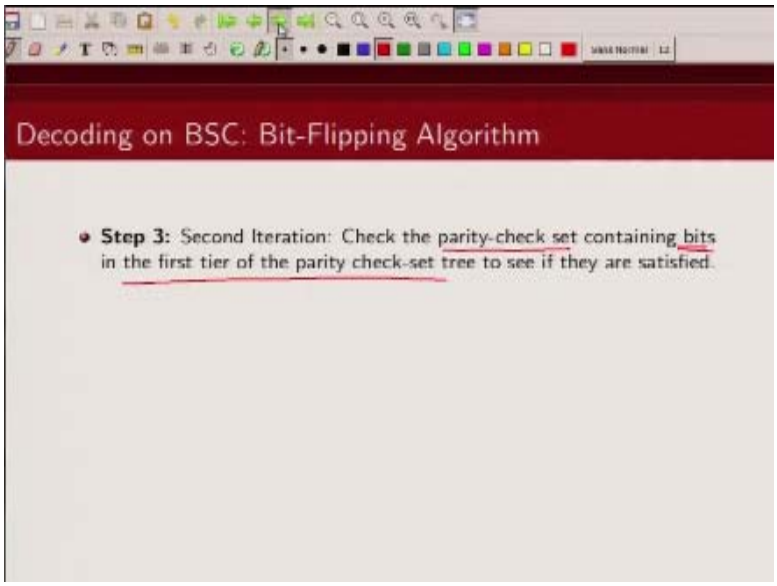
So then we will go to the next tier. So next iteration we will check parity-check set containing bits in the first tier of the parity-check constraints of the tree. And we will see if they are satisfied, so what we are going to do is we are going to go into the first tier.

(Refer Slide Time: 21:30)

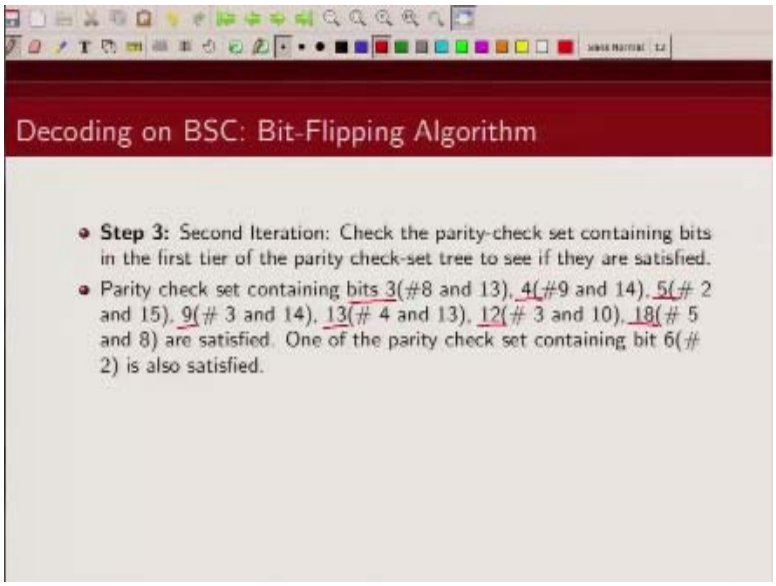


So we are now going to look at these bits, and we are going to see if the parity-check sets involving these bits 2, 3, 4, 5, if the parity-check sets involving these bits, are they getting satisfied. If they are getting satisfied fine, if they are not getting satisfied then we will again have to flip the bit to make them satisfy. So this is how we are going to proceed, so let us look at second iteration.

(Refer Slide Time: 21:59)

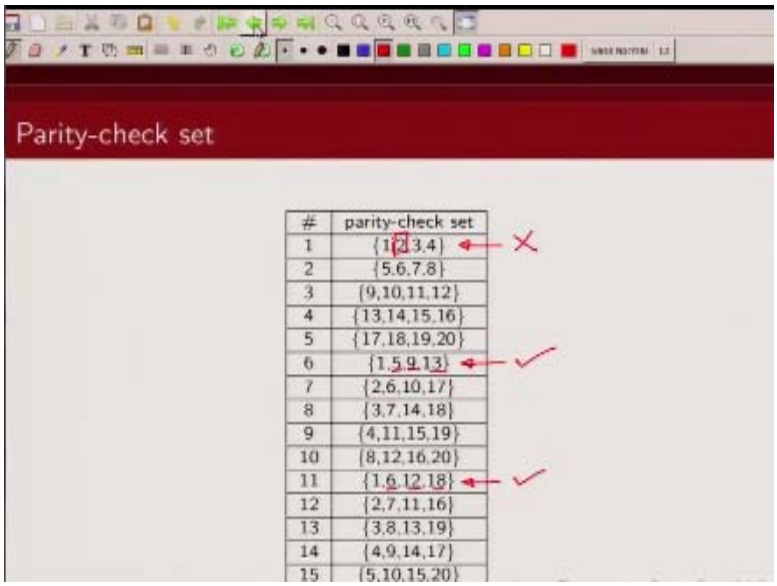


(Refer Slide Time: 22:00)



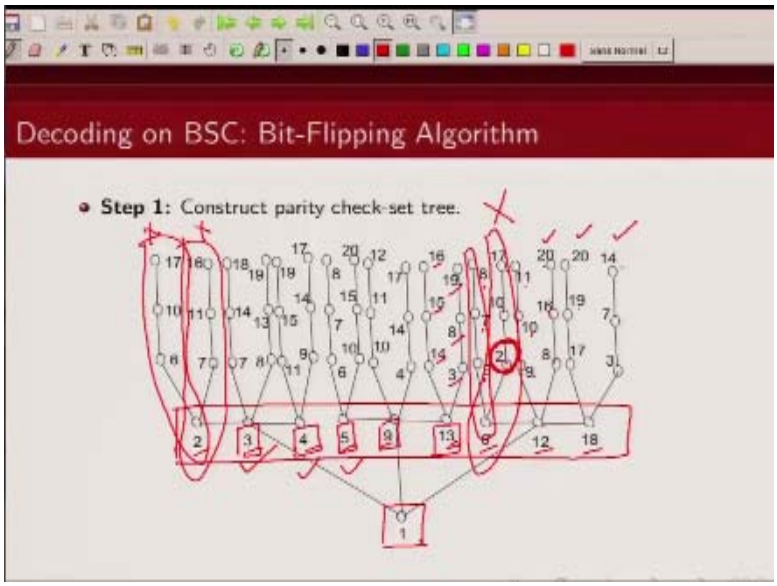
Now what we are going to notice it is that, since bit 3 was not in error, bit 3, 4, 5, 9, 13, 12, and 18 these are – these will get satisfied.

(Refer Slide Time: 22:17)



So if you go back to the parity-check set diagram.

(Refer Slide Time: 22:22)

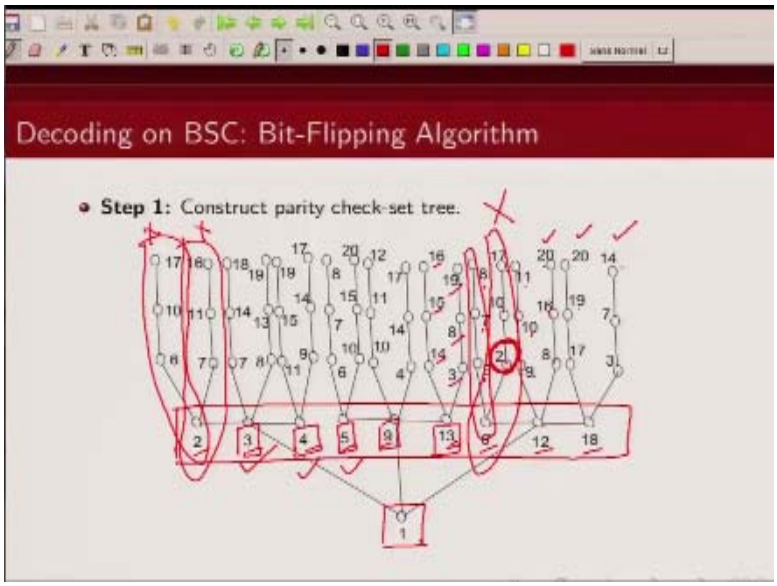


This was not in error and this involves 18, 14, 7, none of these were in error, similarly this involves 8, 13, 19, these were not in error, so all the parity-check set containing 3 will be satisfied. What about 4? 4, 11, 15, 19 they were not received in error, similarly 4, 9, 14, 17 were not received in error. So this parity-check sets will be satisfied 5, 6, 7, 8, this will be satisfied, 5, 10, 15, 20, again these will be satisfied.

Similarly 9, 10, 11, 12, no error in any of the bits so this parity-check equation will be satisfied, similarly 17, 14, 4, so this will be satisfied. 13 this has 16, 15, 14, and 13, none of the bits are in error, so this will be satisfied, then 3, 8, 19, 13, again this will be satisfied. What about this, 6, 5, 7, and 8, none of those bits are in error, so this will be satisfied, this will be satisfied, but what about this? 6, 2, 10, and 17, now this bit is in error, this bit is in error.

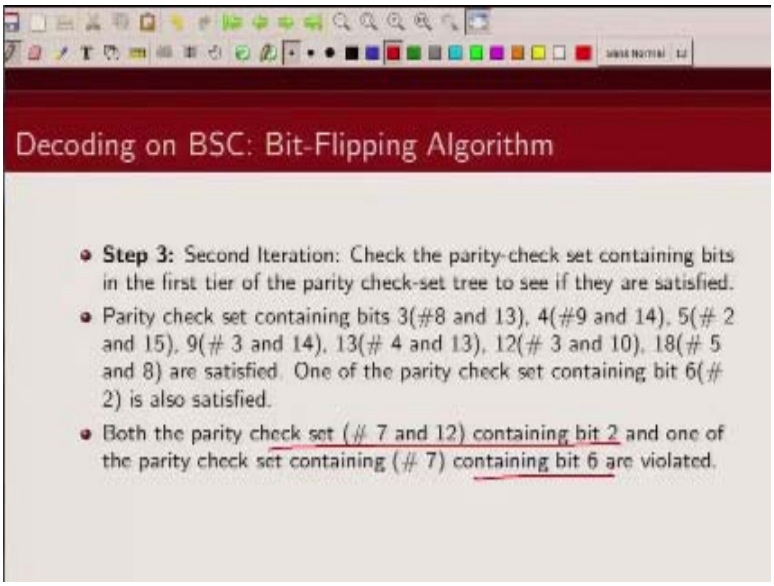
So this particular parity-check equation will not get satisfied. What about this? 12, 9, 10, 11 this will be satisfied, 12, 8, 16, 20, this will be satisfied. Similarly 18, 17, 19, 20, this will be satisfied, 18, 3, 7, 14, this is also satisfied. What about 2? This will be not satisfied, and similarly this will be not satisfied. So what we can see is, the parity-check sets involving 2 is not getting satisfied.

(Refer Slide Time: 24:23)



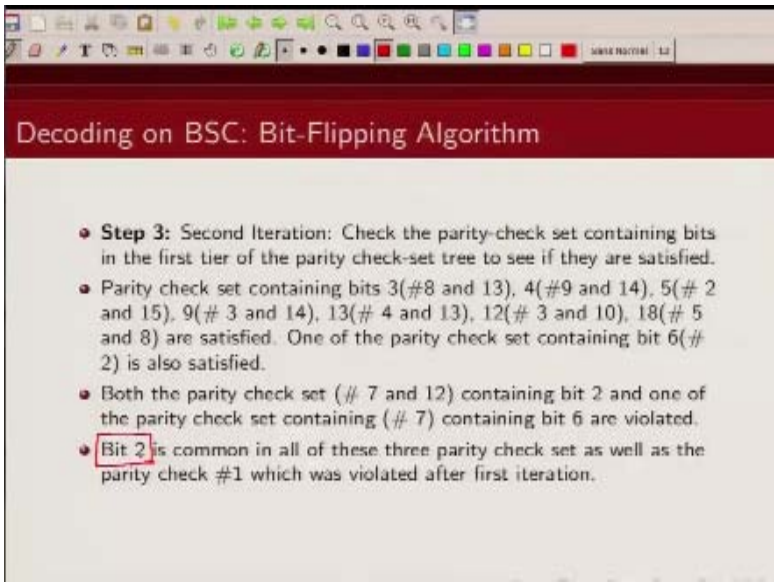
Because here there was 2, here there was 2, here there was 2, this is not getting satisfied, 2 of them are not getting satisfied. And the third one is this one, which involves 1, 1, 2, 3, 4, this is getting satisfied. This is also not getting satisfied, because 1 was corrected, 3 and 4 are correct, so this is also not getting satisfied. So what we notice is parity-check sets containing 2 are not getting satisfied.

(Refer Slide Time: 24:54)



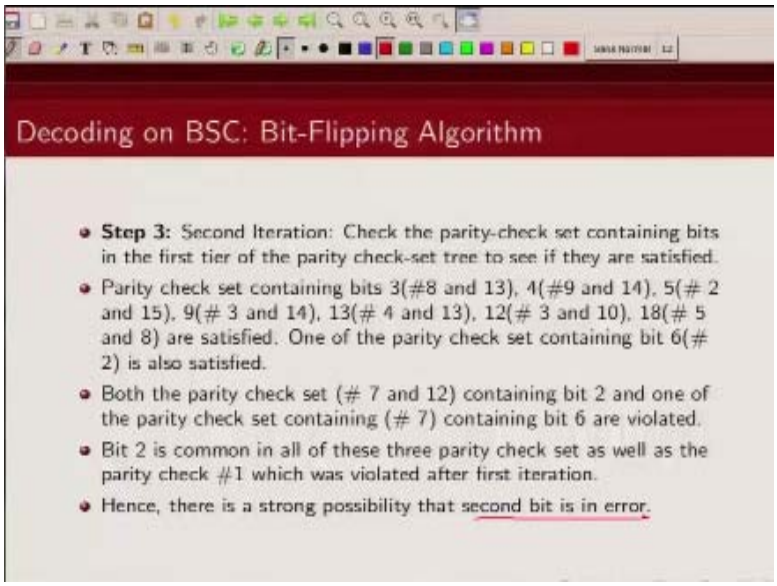
Then in that case what do we do, we are going to flip the bit, so parity-check sets containing bit 2 are not getting satisfied, again one of the constraints containing bit 6 has bit number 2 and it was not getting satisfied.

(Refer Slide Time: 25:11)



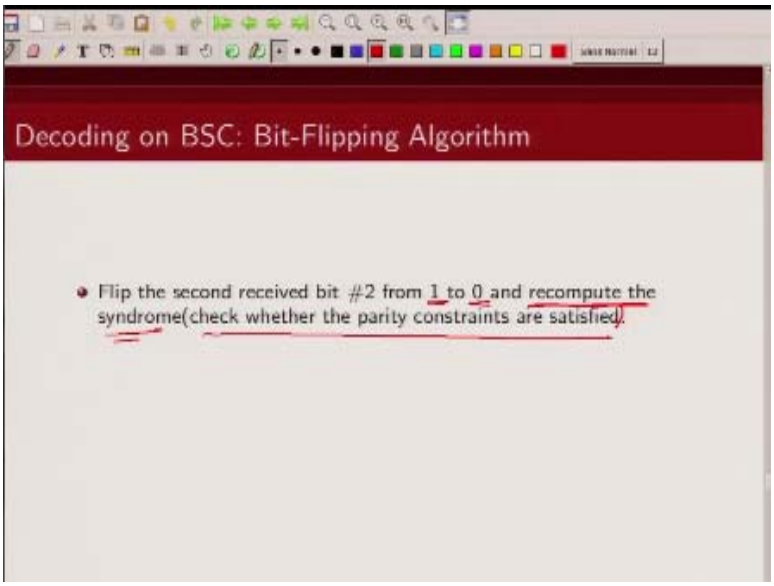
So bit 2 was common in all the parity-check sets which were not getting satisfied.

(Refer Slide Time: 25:22)



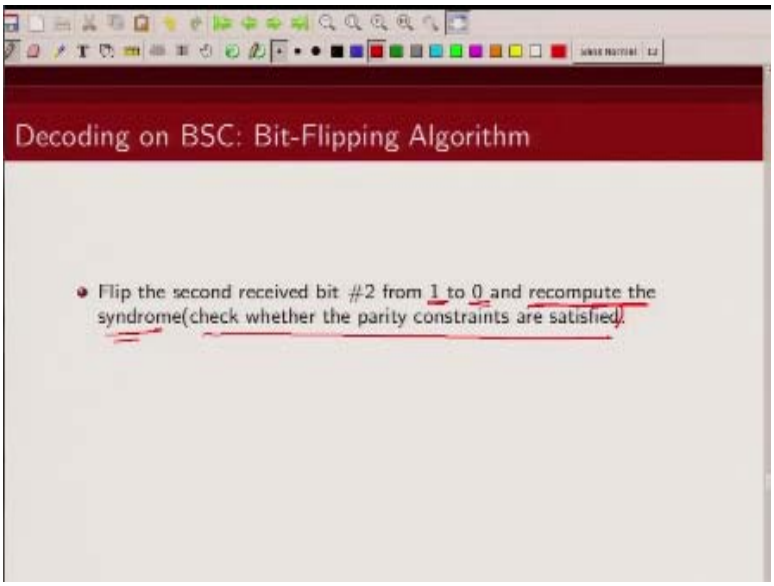
So what we do is we think that second bit is in error.

(Refer Slide Time: 25:27)



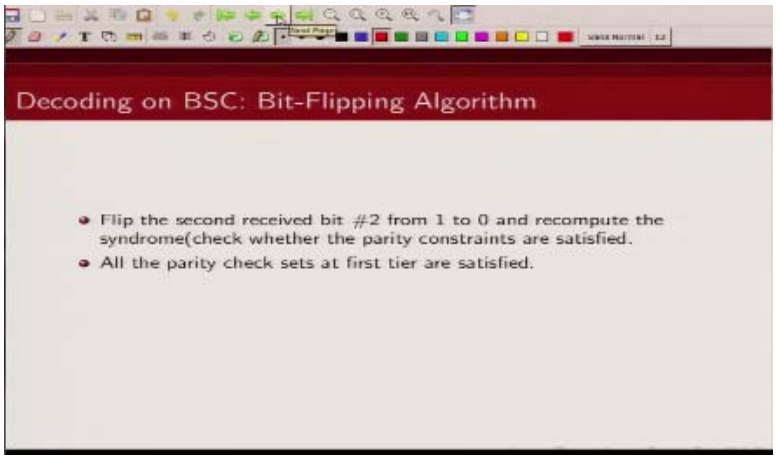
And we are going to flip the second bit, so we flip the second bit was 1, we flip it to zero and we are going to recompute all the syndrome. And now we notice that the parity-check constraints are satisfied, because the 2 bit was in error, after we have flipped it we will see that all the bits involving 2.

(Refer Slide Time: 25:52)



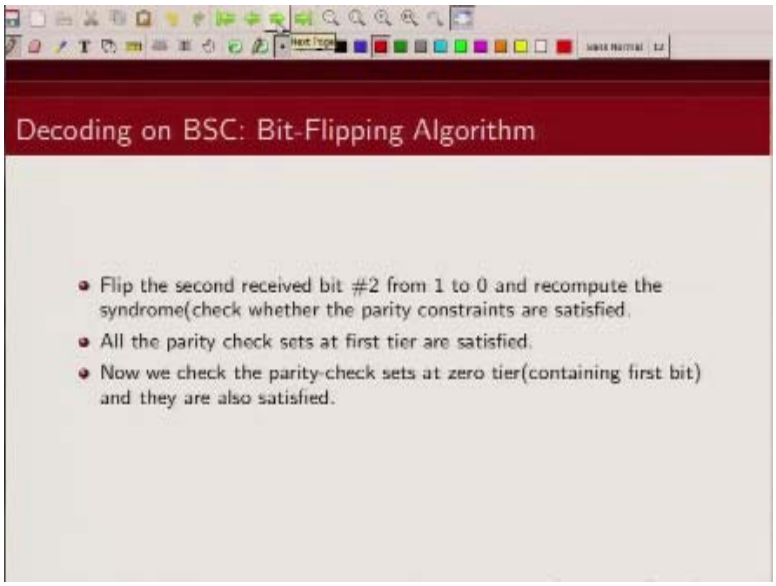
All the parity-check sets involving bit 2 are now getting satisfied.

(Refer Slide Time: 25:55)



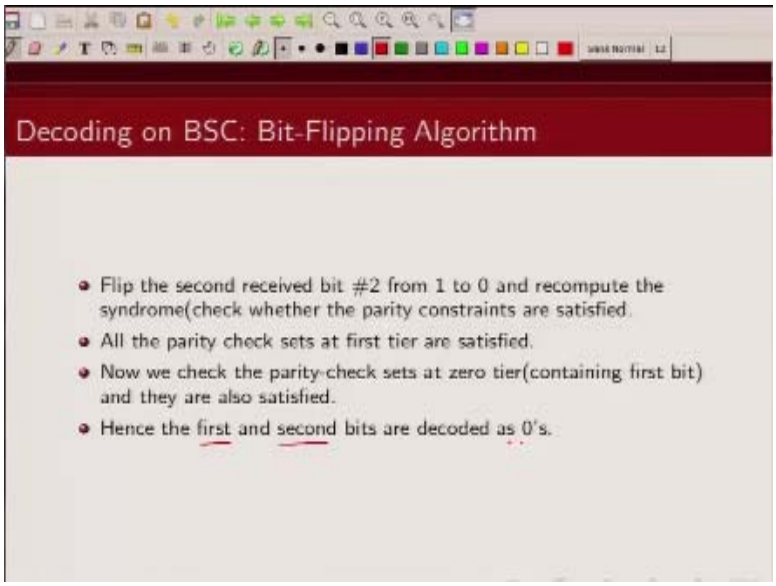
And hence we are able to correct all errors. So if there are two errors you can see that fun iteration was not enough.

(Refer Slide Time: 26:05)



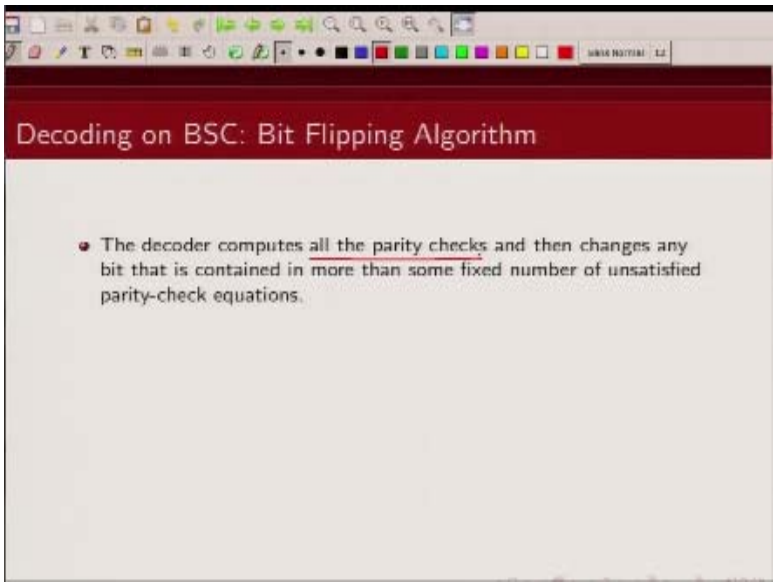
We had to go for two iterations okay. Now we go back and check at zero tier and we see that at zero tier also all the parity-check sets are satisfied.

(Refer Slide Time: 26:18)



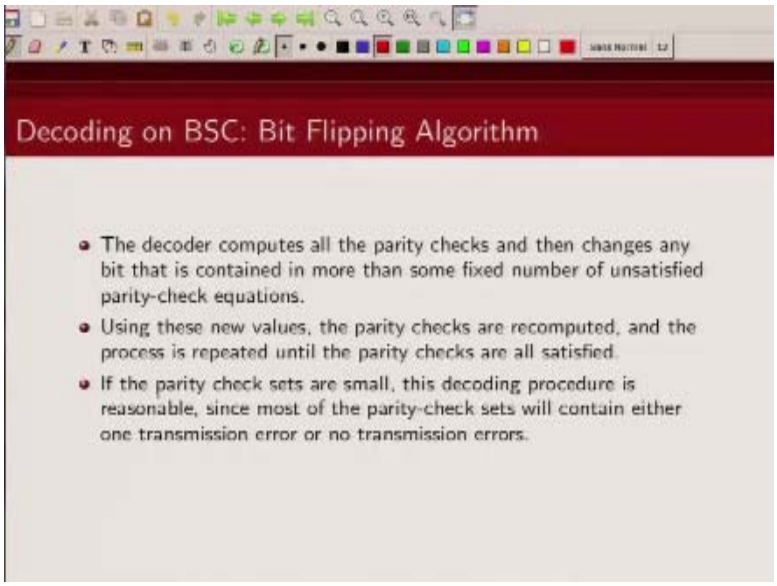
So hence we have successfully decoded the first and the second bit to be zeros.

(Refer Slide Time: 26:25)



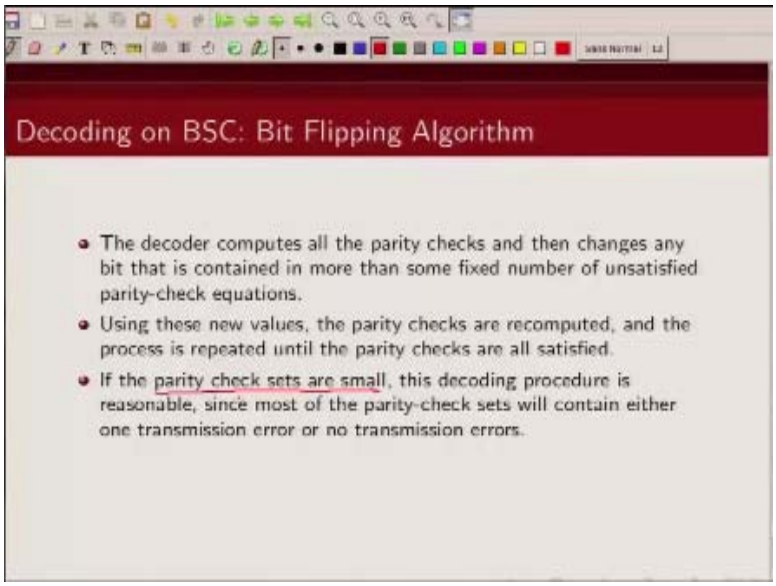
And all other bits were received incorrectly so there is no error. So then what the decoder does it, it basically computes all the parity-check sets and then changes any bit that are contained in more than a fixed number of unsatisfied parity-check equations. And then we recompute the syndrome, recompute the parity-check constraints and hopefully by flipping the bits which are common in most of the parity-check constraints that are getting violated.

(Refer Slide Time: 26:58)



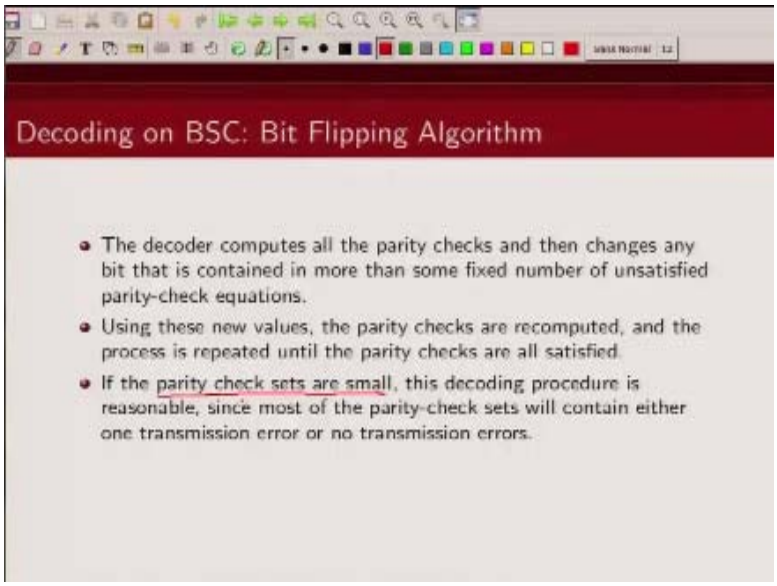
We will be able to finally correct those errors. And each time after we flip the bits we recompute the syndrome, check whether the syndromes are satisfied, when all the syndromes are getting satisfied we have successfully decoded the LDPC code.

(Refer Slide Time: 27:17)



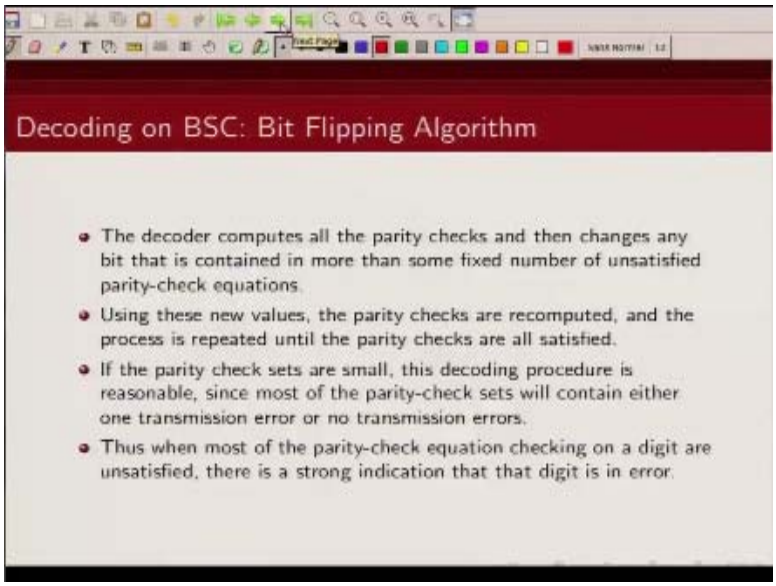
And since the size of the parity-check set is small this decoding is reasonable it does not – it is not very hard and we can also do this process parallelly, we can have a – for each parity-check set tree for each of these bits.

(Refer Slide Time: 27:31)



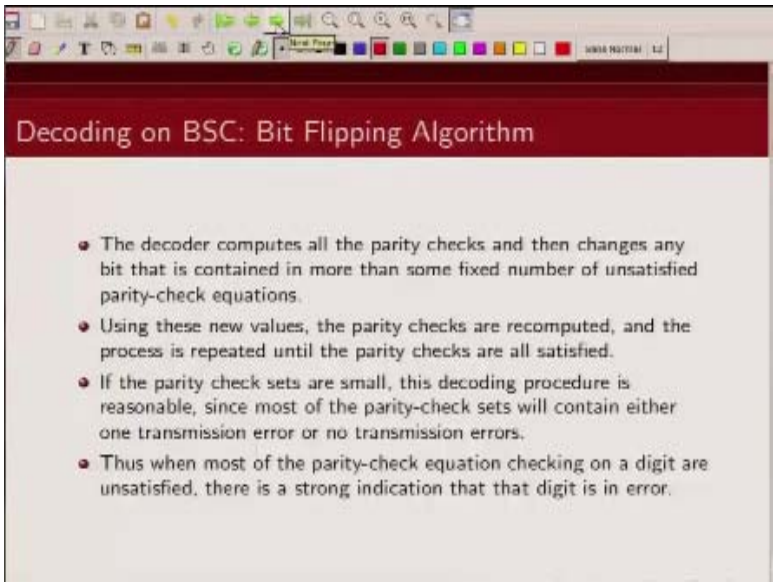
And we can try to do this decoding in a parallel fashion.

(Refer Slide Time: 27:34)



And again this relies on the logic that a bit that is appearing in most of the unsatisfied parity-check equation that is most likely culprit that is the one which is appearing most likely to be in error.

(Refer Slide Time: 27:55)



And we are flipping that bit to correct it okay. So with this I am going to conclude our discussion on decoding of LDPC codes over a binary symmetric channel, we will continue the discussion on decoding of LDPC codes in the next lecture by discussing the probabilistic decoding algorithm, thank you.

Acknowledgment
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 Gairola
Dilip Katiyar
Sharwan
Hari Ram
Bhadra Rao
Puneet Kumar Bajpai
Lalty Dutta
Ajay Kanaujia
Shivendra Kumar Tiwari

an IIT Kanpur Production