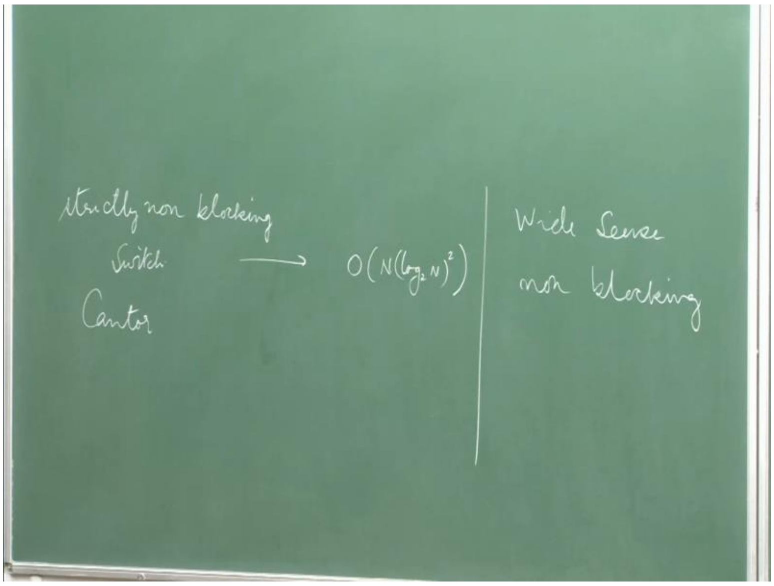
Digital Switching
Prof. Y. N. Singh
Department of Electrical Engineering
Indian Institute of Technology, Kanpur

Lecture - 12

So, in the previous lecture, what we had done was cantor network and its cross-point complexity.

(Refer Slide Time: 00:29)



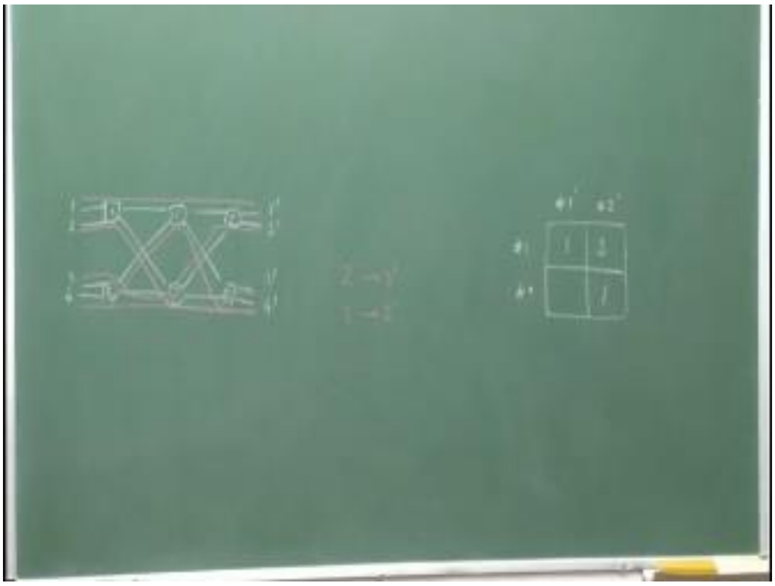
And, we ultimately came up with the bound that, for any strictly non-blocking switch, you will have number of cross points, which are required, will always be less than or equal to this. And if the number of cross points are more than this, you can always build up a strictly non-blocking switch. That was the thing. And this was to a cantor network. And I think one of the things, which I think should appreciate that, the way this was made a strictly non-blocking, is you took something, which is really a non-blocking. And you repeated it in multiple times vertically, so that you create more alternate paths till it becomes strictly non-blocking. And only if basically the contribution was that, to figure out how many number of times that has to be repeated in vertical direction.

So, now, we move on to something, which is different. And we call it wide-sense non-blocking. I have not defined it so far. So, let me first of all explain what is it – wide-sense non-blocking. So, technically, the meaning of this is you will have a switch and important thing is the algorithm to set up the paths. Usually, what happens whenever the switch is in one particular state; and a new path has to be set up. All paths are not set up simultaneously. That is also possible if all paths have to be set up simultaneously, relatively non blocking switch is always better actually; you can anyway find out the all non-conflicting arrangements and use that; and relatively non-blocking switch can be used if all paths are set up together. But, if they are not, it is being done one by one. Or, suddenly a request comes; you do not know what to… Because what happens – a connection setup request will come; you set up the connection; you set up few more. And then some connection will go away. So, coming and going means that sequence cannot be guaranteed; that sequence is random, because users are setting up the connections; they are making the request and they are also relinquishing the connection.

So, the question is sometimes, now, each one of this configuration, whereas the path is set up, is known as the state of the switch. So, with one single… With no connections, there is only one state of the switch that, there is no connection. With one possible connection between an input and output port, for one particular IO pair, there may be many possible ways; you can set up even one single path. All those form in a state actually. And then all pairs can actually have one single path. So, the idea is that, when we want to set up the connections; out of you, all these possible paths, which particular option should I choose whenever a new connection has to be set up. When a connection is going away; it is being relinquished, I cannot do much actually. So, I am just moving on to a state with lower number of connections. But, whenever a new connection comes, I need to have a definite algorithm to ensure that, the switch always remains in non-blocking state; that non blocking state means in that state, whatever free input ports, whatever free output ports are there; if I want to set up a connection between them, I can always set up the connection. So, that should be possible. So, that is the idea.

So, we are going to now look at an example. There is no mathematics; I am actually now going through what we call state space. So, I am going to build up the states of the switches; going to iterate them that, how you move from one state to another state. And based on that, we will figure out that, the switch is wide-sense non-blocking if I avoid certain states of the switch. There is going to be exactly one state. Now, before going to this, I have to understand that, many switches states are equivalent. So, I will give few examples of them. And then what we will do is all equivalent switches states will be represented by only one single state. So, by some certain transformation, I can actually converge from those set of switches to only one state. So, it is like forming sets of equivalent states. From any state, you can move to any other state by just doing transformation. They are equivalent in the sense that, functionally or mathematically, they are same actually. But, when you draw them, they look different; they only look different, but they are not different. So, let us start with that.

(Refer Slide Time: 05:35)

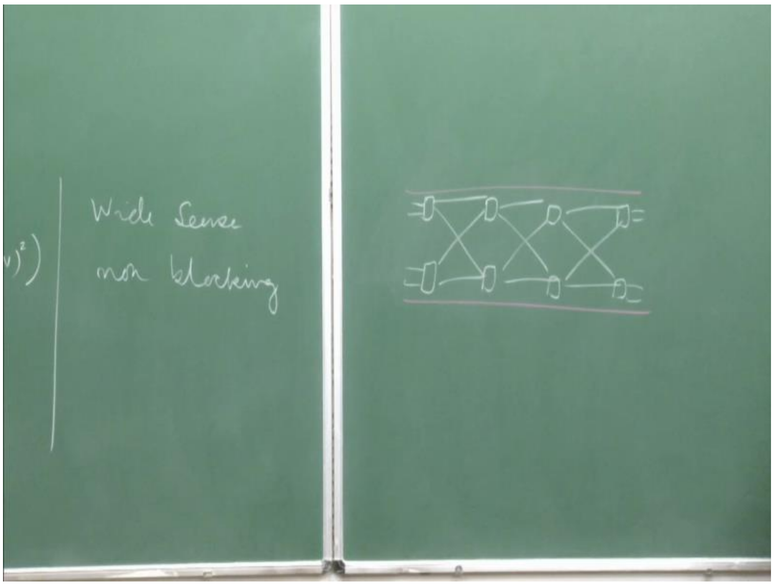


Now, the wide-sense non-blocking switch – you already know this 4 by 4 switch is a ((Refer Time: 05:41)) network actually. And this thus satisfies ((Refer Time: 05:53)) theorem. This is a relatively non blocking switch. I can always setup the paths; I can give an example for this one. So, I set up the path say from 1 to 1 prime in this fashion; and 4 to 4 prime in this fashion. This path is already set up in this case. Now, can I set up a path from 2 to 3 prime and 3 to 2 prime; it cannot be done. You can observe that, I cannot set up a path from 2 to 3 prime. This is not possible in this case. So, these are blocking switches. But, I can always do the rearrangements. So, once I do the rearrangement, it is possible to set up the path. So, rearrangement will be pretty simple; you can actually use…

Let me do even that Paull's matrix, so that it becomes an example how this is done. So, Paull's matrix for this will contain only… So, this is switch number 1, switch number 2. These are not port number. So, this is 1, 1 prime and 2 prime; 1 prime… 1 and 2. Middle-state switches are also written as same as 1 and 2. So, in this case, 1 is already connected to 1 using 1; and 2 is already connected to… The switch 2 is connected to 2 again by 2. This is what has does happened. When I want to set up a connection between 2 to 3 prime; so, what happens is I am not able to set up, because 2 to 3 prime means from here to here; I need to put up an element here which is not… which is free. But, you look into this column and this row; you will find out all elements are consumed – 1 and 2. But, I can…

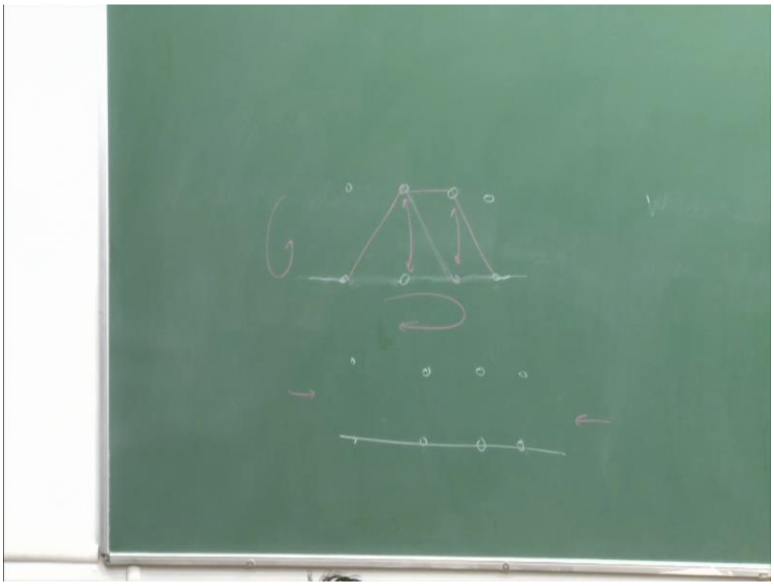
As there is ((Refer Time: 08:22)) theorem says, I can always find out a pair. One is available here, but it is not available here. Two is used here, but it is not used here; there is 1 and 2 pair. So, I can simply put the 2 here and 1 here. That is the way, because chaining cannot be done; it is a very small Paull's matrix. And then I can set up a connection. So, this technically means 2 to 2 prime. This one should be done; obviously, this was a choice actually. So, I can do it this way. Then, 2 to 3 can actually very done very easily. And then of course, 3 to 2 prime also can be done very easily. So, by rearrangements, I can always set up the connection. But, I can now make it wide-sense non-blocking by adding one more stage.

(Refer Slide Time: 09:29)



So, this is what I am going to prove. And I am going to leave one question unanswered actually after this. I add one more stage. I am creating one more alternate path; that is the only thing, which I am doing. This switch becomes wide-sense non-blocking – the proof we will come across. My question is for which even I do not know the answer. Can I strictly add few more stages and make it strictly non-blocking? Is it possible? And I do not think it is possible. There is in for that. I have already given you the hint when I solved earlier. See so far if you can get a switch state in which it is not possible to say connect to one pair of free input and free output port; if you can even find out one state switch can go into blocking state if that end up in that state. And this is always going to happen if you set up a connection like this and this. These two pairs will always connect to the specific pairs here; alternate cannot be done, because they always have to do criss-cross across the switch. So, this cannot be made converted into a strictly non-blocking by a ((Refer Time: 10:59)) states actually further. You cannot keep on adding more stages and make it strictly non-blocking. But, it becomes wide-sense non-blocking with four stages. That is one thing, which is sure.

(Refer Slide Time: 11:18)

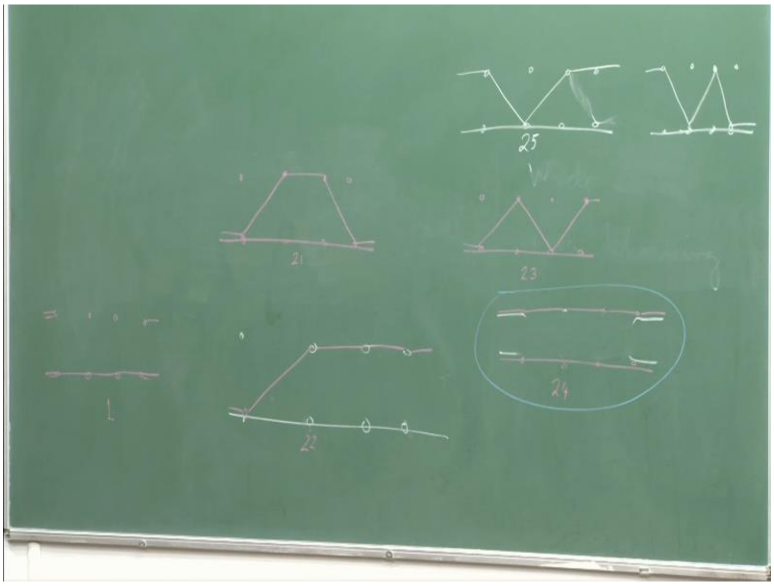


So, let us start with that. So, I will make the state diagrams. And then we will set up the equal… So, I am just going… because I have to draw lot of these figures now. So, one possible state is no connection. And there is only one state of this kind. When there is no connection, this is the state. The connection between these switches is the one, which I have already drawn earlier. But, now, we will only draw if a connection has been set up to be more clear. So, if I set up only one connection, I can set up connection in this fashion. That is one possible state. I can also make another state actually if with only one connection, even this is also the state. But, as I told, is a matter of visualization – where I put the node; it is like graph theory. So, node placement gives us the visual picture.

But, mathematically, if I swap these two, I will again get back the same thing. So, this and this are equivalent. Instead of this, if I have something like this; I can swap these two; I will again get back this thing. So, whatever connection pattern you take from whichever input port, whichever output port; so far there is only one connection. All those connections can be always converted to this. And how this is done is by simply moving this node here and this node here; links are not disturbed. When I move this node here and this node here and this one on this side, what happens? This node will come here and this node will come here; these two remain as it is. And this straight line has to be now connected to this. That is one kind of transformation.

Second transformation is I am looking from this side; I can start looking from this side; it is all the same visually. Whatever is true from the input is also true from output. So, I can also swap; I can rotate this figure like this. So, I rotate the board and you see from the back side of the board. That transformation is also fine. So, I can do this thing; I can do the translation of the whole switch; I can do it this way; the switch state would not change; it remains same. So, I will be only representing one single state, which is equivalent of all these possible states. So, this is I think a very commonly done stuff when we try to merge all equivalent states and represented by only one single state; otherwise, I will not be able to draw actually. This switch does not have large number of states; that is why we can draw and do it graphically. So, that is the one single connection state.

(Refer Slide Time: 14:47)



Now, let us set up two connections; and then what happens. So, I want to set up the second connection. How it can be done? I have to systematically do kind of all possible combinations; that is the only way. So, let me do it actually. So, I will start, because there is another port on this side. But, I cannot go straight; that is not possible. There is only one link. I have to go up. Now, here there are two possibilities: going up and going down. So, I will search for both actually. I will create all possible combinations and then try to do the merger. I will go first of all up. Again I can go up and down. I will go up actually. And then find out the combination. This is one possibility with two connections.

Second possibility – let me draw it here. The connection starts from here no option goes to only up. I have to use even these remaining two also later on and draw the equivalences actually. I have gone up; I will still go up; but, I can come down. That is a next possibility. And remember this merge state is only the one, which I am considering for one connection. But, same combinations can exist even with other equivalent states of the single connection thing. But, that does not matter; I can represent them by this. This defines the complete state space actually this way for two connections. So, this has been up and down I have taken care of. So, now, the next one I have to go. For the down, there is no other option for him; it has to go up. And I think for all two connections, I have figured out these states. ((Refer Time: 16:44)) actually; none of them are equivalent. All three are unique. By any transformation, I cannot convert any one of these to anything else; that is not possible. These are unique. So, the first one, which was there with wide; I call this state as 1 and I call this state as – this one as 2 1. You can give numbering anyway you want; I am going as per my notebook actually; the way I have solved it. So, I have taken the second connection from same port.

Now, let me take it from a different switch and find out if there are more connections, which are feasible. So, let me take it from here. I can go up and down both. So, always I will go for up first actually. Go this way; I will again go up first and up first. That is one state. And we know this is going to be a blocking state, because now remaining two ports can only be connected into only certain map. All possible combinations between two free input ports and two free output ports cannot be made. So, this is a blocking state. But, this is not true with the remaining other states remember. Here I can take these two and connect with anyone of these. Now, this can, because there is a cross connection, which is happening here; they are sharing one single switch. Just by changing the state, here itself can be done. Same is true here. But, this is not possible in this case. So, this one is a blocking state. So, let me put a circle around it. And ultimately, this is what we want to avoid. I should not be able to come to… Not this one; I should not be able to come to this state either by removal of a connection or by creation of a new connection; that is the only thing, which we have to do. And that is what will give the algorithm.

There is one more, which is left now – the fifth one. Let me draw it here. So, I start from the top – goes to bottom; I have to first of all go with the upper one. And then the last one will be this.

Student: ((Refer Time: 20:09)) … Criss-cross switches.

Which one?

Student: ((Refer Time: 20:15))

You can. So, from here you have to set up this switch; that is the only option you have. So, this can only be connected to this. But, this cannot be connected to this three port. That is why it is a blocking.

Student: It can be connected… One switch has two ports; no sir?

There is one port, three port here; one ((Refer Time: 20:35)) to connect this and this.

Student: Even these measured switches can…

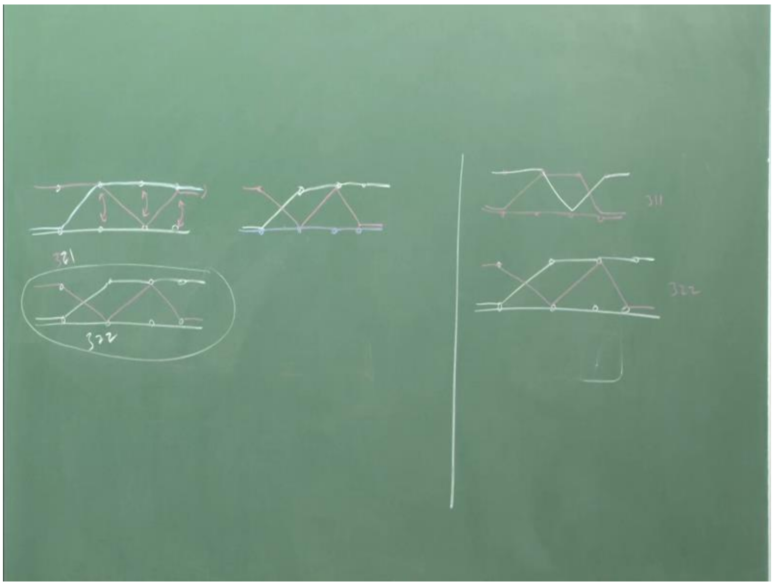
There is only one link between every stages. It is not possible. These are already occupied by this red color correction. So, this can only be connected to this and this one can only be connected to this actually. So, it is a blocking state. For other cases, this is not true.

Now, coming to the equivalence, because I need not number everything; this one I called 2 5. So, fifth kind of state. Now, this state if you carefully observe is nothing but 2 3.

Student: ((Refer Time: 21:23))

Rotate. So, from this side, instead of that, you start looking from the right side. We will get the same switch actually; same switches state. So, this need not be considered; I have already taken care of. So, all possible switches states, which have been contracted by equivalence into five possible combinations have now been present. So, this I can remove, because this is already being covered by 2 3. So, next will be… So, I have done one; I have done with two connections. Now, the question is of three connections; how this will be done. Now, this slightly elaborate. Ultimately, we will end up only with very few states actually. But, every time there will be equivalence; you should be able to find out. So, let me take up first of all 2 1 and find out all possible things, which can happen.

(Refer Slide Time: 22:28)



So, what I will do is – the final states I will maintain here. Here I will do all homework. And whatever is the final state, I will just copy it there. So, let me take 2 1 first. This is fine? This is a switch 2 1. So, I call it 3 1 and I have to now set up the path into this – third label – the first one. Now, remaining two ports, I will first of all try to go up and then down while trying to setup the connection; and find out this state. So, first is this. So, go straight; there is no possibility of going this thing; this is the only option available. This is one state; I call it 3 1 1 – next possibility. So, this is state I am going keep it as a final one. That is what is 3 1 1. Next possibility is I can do it on this itself. Instead of going up, I can go down actually. So, if I do that, the only option is going this way and this way. But, instead of looking from left, we are looking from right; it is again 3 1 1 only. Instead of looking from left, look from right; it is 3 1 1; there is no other option. So, that is already been covered; so, need not bother. Now, is there any other option in this case? So, no other option. I think all I have covered with this. So, if I am starting from this state.

I have to take the next one, which is 2 2. So, that is the next one. So, I call it 3 2. So, I will take the first connection and let me try for this – state 1 first always. This will be the first connection. So, I call it 3 2. And this will be equivalent to something actually. And that equivalent will be… This is 3 2; this is one combination. Next one, let me try full; then I will give the numbering. I have numbered it slightly differently. Then, I can go for a bottom one, which is this. And what I am trying is I am trying to make sure that… because one of the ports will have two inputs and two outputs; I am trying to convert it to a standard form by transformation, because otherwise, it will become difficult to track, so that both… There has to be one straight line connection. So, it is as if emanating from here. And both the connections – two ports will be… – should come from the bottom side. I have to transform every situations to that thing.

So, now, I have to transform this one. So, these two lines has to come to bottom. So, that is why I have to do this swap. Once I do this, only this particular line is going to be the straight line in the bottom. Now, I have to even do the swap of these. Once I do that… So, that is this particular line. Let me use a different coloring ((Refer Time: 27:16)). So, this is this. When I do this transformation, all three nodes have been swapped actually. So, the white color ((Refer Time: 27:39)) go what way? It will go up. This node also goes up; this node also goes up. This will become this. And the red one – this has gone down. So, it will come here. This has gone up. So, it has to go up. This has gone down. Now, these two are exactly same. So, these two combinations are equivalent again. So, I called this as 3 2 2. This is 3 2 1. And I take this as a standard, because that is what I am referring. That bottom line has to be straight connection.

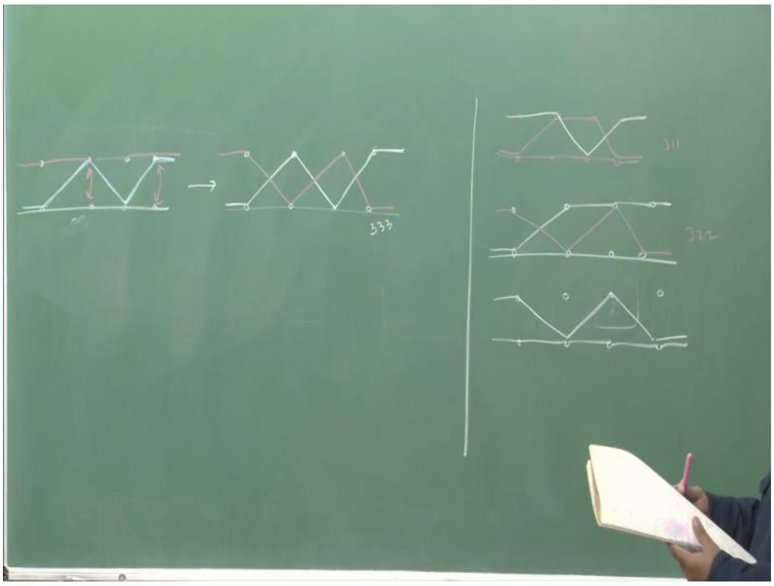
And, two ports – both the ports in the bottom; switch has to be occupied as input and output. But, that was my thing; you can take any reference and try to do the transformation as per that. So, the standard connection will be now this. I think all possible combinations, which can emanate from 22 has been taken care of by this. And that is being represented by 3 2 2, because all states are equivalent to that. So, next is coming to 2 1, 2 2, 2 3. Let me check what happens with that.

Student: ((Refer Time: 29:36))

This one – you should do swap; this will become this; and these two are same. This is also 3 2 2. So, these two are equivalent actually. Only thing you… I actually have done it very simply; I said always the bottom line has to be straight and both ports have to be occupied on the bottom side. So, once I try to make a comparison, then I can immediately make a comparison. And then you can always look from input side and output side and try to make a comparison for the equivalence. So, transformation becomes simple. I can decide very easily; switch nodes to swap, because these two – this port has to come down. And there has to be a straight line, which can only happen is – if this connection comes down; so, this means all three have to be swapped; or, this has to be swapped and then you look from the top actually. Instead of the bottom, you look from the top. Then also, it turns out to be same.

So, now, coming to the next one, I think you all have to do this thing once. Unless you sit down and do it on your own, you would not be able appreciate how this actually happens; because while doing here, you may be grasping something, but not everything. And appreciating everything is not possible actually. Still I have been trying actually.

(Refer Slide Time: 31:19)



So, next one is – let me take 2 3 – this particular switch. And let us put all combinations. See if the equivalence can be set up or not. So, I start the connection; I start from the… As usual, I always try to take the upper part; upper part and upper part. There is one possibility. And because I want these two ports; when they are occupied, this node has to be in the bottom. So, I will transform these things. So, swap between these two. And this is an only straight line going up. So, this has to remain here. So, only this has to be swapped. But, in the bottom, I can get a straight line.

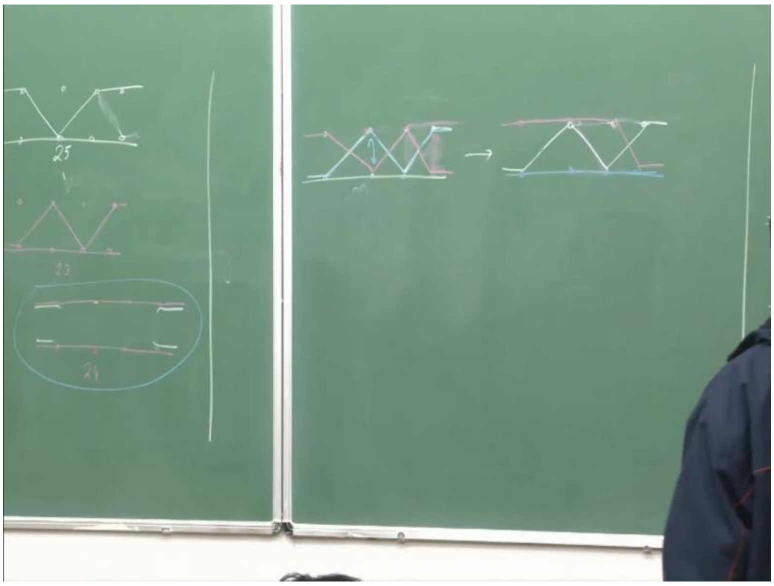
So, the transformation will look… Let me change the color, so that… So, with the swaps, now, you will have this thing as a straight. The white colored one will… Because of the swap, this remains down. So, it remains as it is here in the bottom. This goes up. And what happens to the red one? So, this is again one of the standard states and I called this as 3 3 3. It is basically being derived from 23. That is why I am calling it. 23 basically being used. So, this is the one state after transformation, which you will get. And this is also one of the standard states actually.

Student: ((Refer Time: 33:43)) Only two nodes…

Yeah, because I want two ports to be occupied here; two ports to be occupied here; and one horizontal line for a reference. That was the purpose. You could have taken an alternative, not an issue; that is possible. But, be consistent. We will end up in some transformed version of these states only. So, this one is… And I call it 3 3 3.

So, next possibility is… So, I have gone up. Actually, first of all, let me try downwards from here. Now, is it equivalent to something already existing? This is nothing but can be transformed to 322. So, instead of looking from input side, look from the output side. So, instead of looking from this side, look from this side; it is nothing but 322. So, it is already being covered. So, need not bother about it actually. So, next possibility is let us look… This can only go horizontally; does not matter. So, where the next possibility is? When this goes down. When this goes down, it has to go up. Now, it can either go this way; there is one option. I have to do a transformation again, because I want both the occupied ports to come down. So, this also has to be swapped; this need not be swapped for the transformation.

(Refer Slide Time: 35:51)



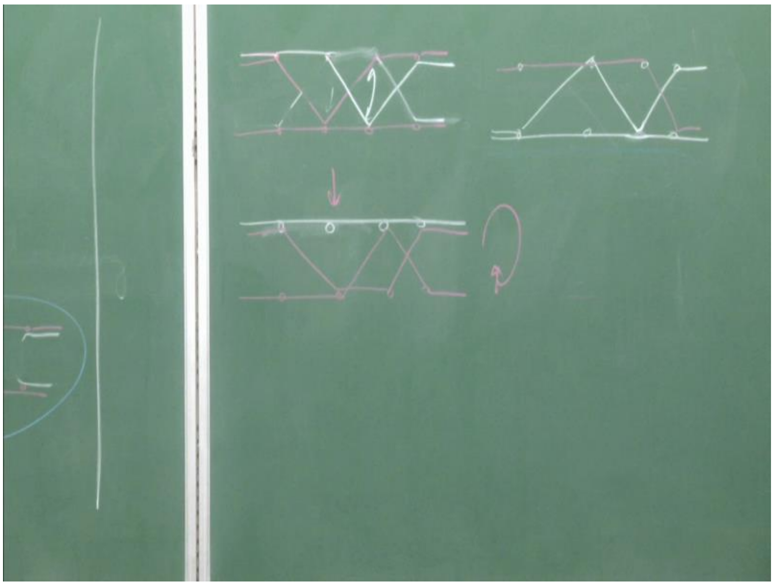
And, the transformation will give me… Or, all three have to be swapped. All three have to swapped; otherwise, it cannot be done. So, this will go up; this will also go up.

Student: The only change will be in the last station, because there is only one output busy; there is only… Sir, if you see second stage and third stage, the bottom line – bottom nodes – both are busy sir – busy links.

No, I have to… I will just figure it out. I will just need… Some swapping problem actually has happened. So, this one, this one… Yes. So, this need not be swapped actually. This swapping will not be there; then it makes sense. That is a mistake, which I made. So, that need not be swapped. Then, it will become a straight line. So, this one will come here. This has not been swapped… It will come here. This is nothing but this; just turn around; make input to output and output to input same as 322.

So, all the possibilities have been covered already now in this case. One only possibility which has been left is this. Now, this will give you nothing but 333. This will give you 333. So, this is also covered. So, all possibilities have been covered for… from 2 3 and from 2 5. Now, only thing, which I have not kept the record here is from which state you are able to move to which state with three connections. But, that… I think you can again… You will be… Actually you must be remembering by this time; you may not have noted it down.

(Refer Slide Time: 39:00)



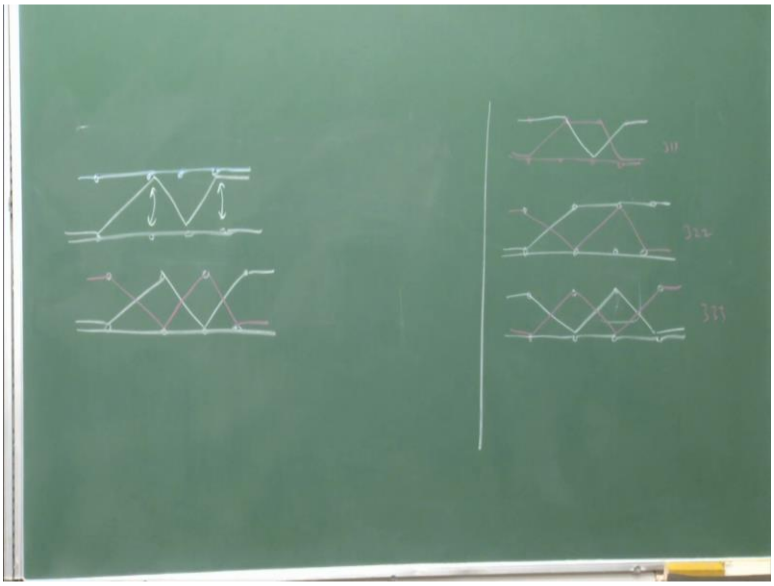
So, last state is now 2 5. In this state only you setup a connection. So, it has to be start from the bottom; or, top does not matter actually. If this starts from the bottom, you will… I will come to this, what it will look like. You can also come from here – 2 3… One of these ((Refer Time: 39:28)) Not this one, but the upper one will come. This is 2 5. From 2 5, I am just starting. 2 4 I have left; I will do it later. So, in this case, you can start from any side. I can start from the bottom first. This is the only option I have. I can go straight, come down; this is I think looks to be very similar to 3 1 1. That combination is taking care of 3 1 1. This is nothing but 3 1 1 actually. Go this way; you have to do a swapping to find out the equivalence. You swap this and swap this. And of course, the last one – this is also nothing but 322. So, that is also taken care of. So, these two: top and bottom – both options have been taken care of. And there is no other option in this case. So, already this can be mapped to one of those three states.

And, next you can try from that top side from here. Only possibility is go up; you do this. This is nothing but already covered state – 322. So, input, output – you have to just swap. Next possibility is this. You have to do a swap of this and then I can just look from the upside. Let me do it here. So, this one is straight line; one is straight line. Look from the top; this will be also nothing but 3 1 1. So, instead of from bottom, you look from the up – this way.

Student: This top have to be sent to ((Refer Time: 43:18))

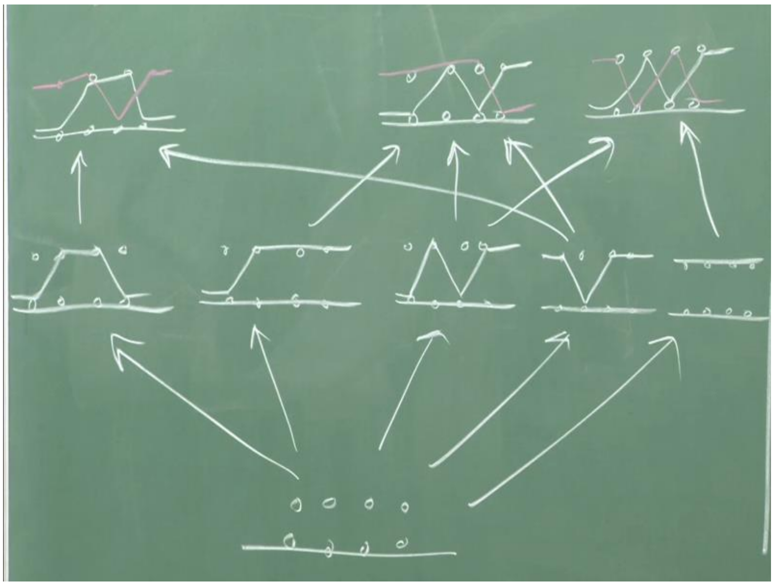
You swap now whole thing; you rotate it; you rotate it actually. You look from the top, instead of from the bottom. Currently, I am looking with this reference. So, you stand on your head and then see. This will be same as 3 1 1. So, it is already taken care of. There is no other possibility; I have consumed all the possibilities now. I have consumed all the possibilities. So, bottom one I had already taken care of with all possible combinations; the top one also I have taken care of all possible routes, which are there. So, I end up in one of these three. In fact, there are only three possible states with three connections; they are not more.

(Refer Slide Time: 44:12)



So, the last one we have to take 2 4. So, 2 4… So, you set up a connection this way. And I want a straight line in the bottom as well as both connections. So, I will do the swap of this; I will do the swap of this. And this is nothing but 333. This one always lead to 333. So, now, I have to just draw the arrows – the transition diagram.

(Refer Slide Time: 45:25)



So, from here I can go to this state; I can go to this state; I can go to this state; and I can go to this state; as well as I can go to this state. There are total five states in which I can go. So, I think now I can draw it once you understand this. So, from here you can come to either this; you can just keep on verifying from there actually. So, these five states we should be able to come. From these you should be able to go to this one. From here also you can come to this one actually. So, from this, you can come here; from this, you can come here; from this also you can come here – all three. So, from here you can come here; from here you can come here; from here again there is only one state, which is possible. So, remember that does not matter; all these three states will be unique; fourth state does not matter; only important thing is that, this switch is in a blocking state.

Now, if a switch has to operate in this state and a connection goes away, you might end up and fall back into this particular state, which is a blocking thing. You want to avoid to come into this state in any case. So, when we want to set up a path from here, never come to this state. So, this is not permitted for you. And since because this is falling back, is not in your hand, you should not also enter into this state. So, this is also not permitted. So, if you know algorithm, you avoid this transition; you will always end up in these four and these two and remaining two, which is on top of only in these states you will be operating. And you will always be able to set up the connection without rearranging the existing ones. Henceforth, you got the algorithm and so its switch is wide-sense non-blocking. So, this is the simple argument, which proves it.

So, important thing, which we have learnt is the states; how to contract them into very few states, all possible combinations; and then how to build up a state transition diagram; and basically from there to identify the algorithm of operation for a wide-sense non-blocking switch. So, different configurations again you will have different algorithms. So, this is one of those examples.

Student: ((Refer Time: 50:17))

This one, because if the connection goes off, you might end up in this state, which is a blocking one. See not only here; moving in the direction of arrow; you are also going back when the connections are released. So, there is no guarantee that, connections will be released and you will be always moving from here to here. You can also go from here to here and then come back here. And you might come back here. And from there, there is only one state in the ground when there is no connection.

So, you are moving from one state to another state whenever connections are set up or they are released. So, avoid coming to this; always ensure that, you cannot come to this state. And that will ensure that, switch is non-blocking. But, there is an algorithmic constant here; it is not strict-sense non-blocking; you cannot set up anything arbitrarily; you have to follow certain rules. So, rule is if you are in these states, do not bother. If you are in this state and you want to set up a connection, never try setting up that connection, so that you end up in this state. Always go to this configuration. There are two possible configurations, where you can go. So, never go to this one; always go to this one; you can always set up the path. So, next class, we will see how to implement a time switch using random-access memory, so that we can understand the control structures of the switches.

Thank you.