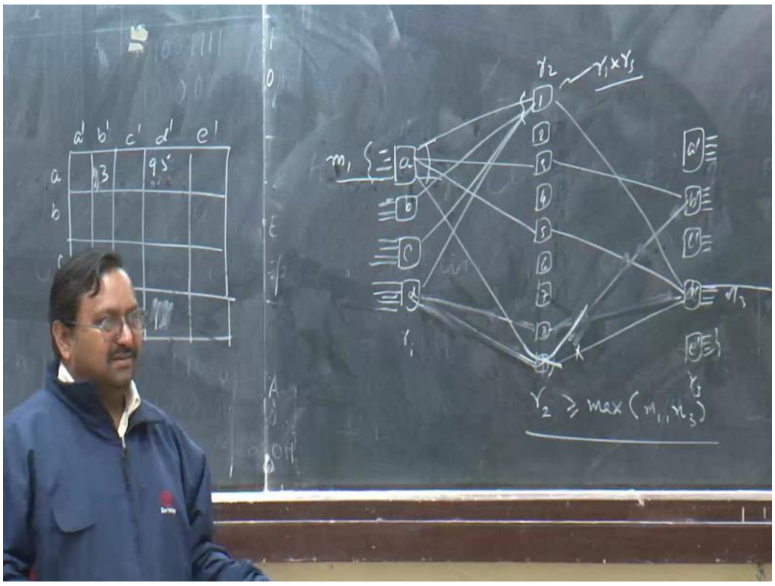
Digital Switching

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

Lecture - 8

So, actually we will go ahead with the clos network now further, and we will now formally proves that in, what cases it will be rearrangeably non blocking? And how you will actually do the rearrangement? That also terms will come we will also prove now formally when a switch will be strictly non blocking in a 3 stage clos network. Intuitively we do have an idea that, number of middle stage switch has to be equal to twice the number of inputs, per switch minus 1. If it is equal to that or greater than then it will be strictly non blocking, but before doing that we have to understand the formalism of Paull's matrix that is we are I think I left in the previous class.

(Refer Slide Time: 01:01)



So, the Paull's matrix is nothing, but it is a matrix it is going to have rows and columns so; obviously, and the number of rows is equal to number of input state switches, not the ports remember it is equal to number of switches in the first stage. So, if I am going to have say for example, a switch like this which has 4 things, there might be many number of incoming ports per switch. So, I will only write a b c d and there will be only 4 rows in this case.

So, I will have only row a, row b, row c and row d. Similarly, on the output side on the 3^{rd} stage basically I will have number of switches, and they need not be same. I am not whatever; I am now going to do formally is for any kind of clos configuration, this need not be symmetric, number of input ports need not be equal to outgoing ports. It has it can be anything arbitrary. So, I may even end up in having 5 switches nobody stops me, I may end up in having 5 switches the number of ports here can be different, the number of ports per switch there that is also permitted.

So, it is a clos network I will be able to deal with it. Earlier case was I took everything symmetric, and then through some rationally or logic I was able to drive when the switch will be strictly non blocking. So, it is a general now. So, if you have for example, now a prime, b prime, c prime, d prime, e prime. So, in this case correspondingly I will have 5 columns, and I called it a prime, b prime, c prime, d prime and e prime. In the middle stage again there will be many switches.

So, fundamental rule is if the outgoing ports 1 thing is very obvious that these outgoing ports in the first stage has to be more than the incoming ports. So, at the switch itself will not create any blocking, is very important should not create any blocking. So, because of somebody trying to coming outgoing port is free, but these ports are less you cannot do it. So, blocking happens because of this n 1 is a smaller than the m 1. So, that condition should do not happen. So, n 1 should be always be greater than m 1, if that is there and m 2 m 3 is again greater than n 3 then also the blocking will not happen, because of the switch it only because of interconnection.

Now, what I can do is; only thing which I need to manipulate now here is, number of switches in the middle stage. If I only keep it 1 unfortunately; there is only 1 line coming from that it will be blocking switch, because these ports are less than the input. So, 1 thing which is sure is, if I have this as an input and this as an output. And for example, this number of outputs n 3, I am just going from certain fundamentals now.

So, m 1 and n 3 if they are equal it is ok, then we know that if the number 1 outgoing port is less than m 1, number of incoming ports is less than n 3 it is blocking, which actually means when r 2 is going to be less than m 1 or n 3, which are as minimum of 2 it is going to be blocking. In fact, that is what the result, which you will get. Now, if I want to set up the connection from here, I want to set up the connection from here. So, one thing which is sure that r 2 should be greater than or equal to maximum of I am sorry not minimum, but maximum this is this has to be ensured.

So, that the blocking does not happen, because of these switches. It happens because of the interconnection. So, that is the first condition. So, if you are below this, your switch is blocking switch, it cannot be rearrangeably block non blocking it also cannot be infected has to be… If r 2 is less than maximum of m 1 and n 3 the switch is blocking 100 percent blocking. So, form certain combination you will find switch cannot get through and blocking is not because of interconnection,, but because of switches itself, because the number of ports will become smaller than the outgoing ports on this side or the incoming ports on this side. So, first condition.

Now, next is, once this thing is clear; that means, now, where are my interconnection? Going to be in this way there is r 2. So, when you make r 2 is going to be greater than or equal to this condition, in the beginning r 2 will satisfy some thing, as I keep on increasing r 2 there must be r 2 after which I should be able to set up all connections, all possible connections.

Because, r 2 is infinity switch has to be strictly non-blocking, number of I o ports are limited, number of I o ports are limited. So, at certain value of r 2 it will become, strictly non blocking at certain value of r 2 it should rearrangeably non blocking, and below this it should be blocking. But, still I am not sure whether it is going to be rearrangeably non blocking, if this condition is satisfied this is. So, I am not sure as of now.

So, this I do not know about that thing, but at certain value of r 2 it will become rearrangeably non blocking, and after certain threshold it has to be strictly non blocking. Strictly non blocking you are not rearranging any connections, you should able to in every state should be able to set up the connections. In rearrangeably non blocking you have to make the rearrangement of existing connections I o map or I o pairs still are maintained at same, and then you can set up the connection.

Intuitively we actually get reduced, that rearrangeably non blocking switch will require lesser amount of resources compare to strictly non blocking, which is going to be true actually. So, that is the case. Now, let us look at how the Paull’s matrix actually helps us. So, you will now put some number or whatever it is on some symbols actually on this, middle stage switches, and when a connection is set up.

So, when I take for example, this particular port I set up a connection, I set up to this particular port. Between these 2 ports a prime and d prime I use a switch 9. So, I am going to route the call through 9, I can now put in this cell that I am now set up a call. I am not worried about which particular port here, because that exactly 1 call which can be set up from there can be multiple. But, through 9 1 can be set. So, if there is another port, which is free which also wants to go to d prime, I will leave have to some other switch. It cannot be through nine because there is only 1 line here there is only 1 line here.

Student: ((Refer Time: 09:25)).

No it is not 1 by 1 there are other switches also, and everybody's connected. So, the number of switches here is r 1 r 3. So, this is r 1 by r 3 switch, all middle stage switches are r 1 by r 3 they have to be. Because, you cannot leave any port open here, every port is connected in the middle stage. All inputs are coming at this port all outgoing ports are on this stage. All middle stages have to be connected to somewhere that is as far as the clos interconnection thing.

So, the next 1 for example, this same guy wants to set up a call from here through 5 that is the next 1, this guy a 1 wants to set up a call to say b. It can use something like 3. So, I with can put 3 here. Some other switch wants to set up this call to d prime. So, this can be through this cannot be through this route, because this is occupied. So, d can be connected to d prime through 8, that is what the paull's matrix is now, what are the validity conditions?

Now, 1 of the simple thing, if I would have I have liked to set up the condition at this particular connection this line was free, but could have used this. I could not have used this line, because this link is already occupied. Because, a connection from a to d prime has been set up. So, which actually means 9 cannot come in the column corresponding to d prime. Because, it is already being occupied, because of d to d prime I would have contented to 9. I would have put 9 here, but nine cannot repeat in the same column, but if I am in a multicast configuration, multicasting is a 1 input is being connected to multiple outputs. I am trying to make copies of the signal to multiple outputs

Student: ((Refer Time: 11:47)).

I am not estimating any probability yes find out whichever free and meet the connection.

Student: Sir, I want to make a connection from d to c prime how.

D to…

Student: C prime.

Exactly right.

Student: Sir why cannot I use that input with 9 and from pair inputs.

From here to 9 and from here to you can certainly use who stops.

Student: Actually multicast.

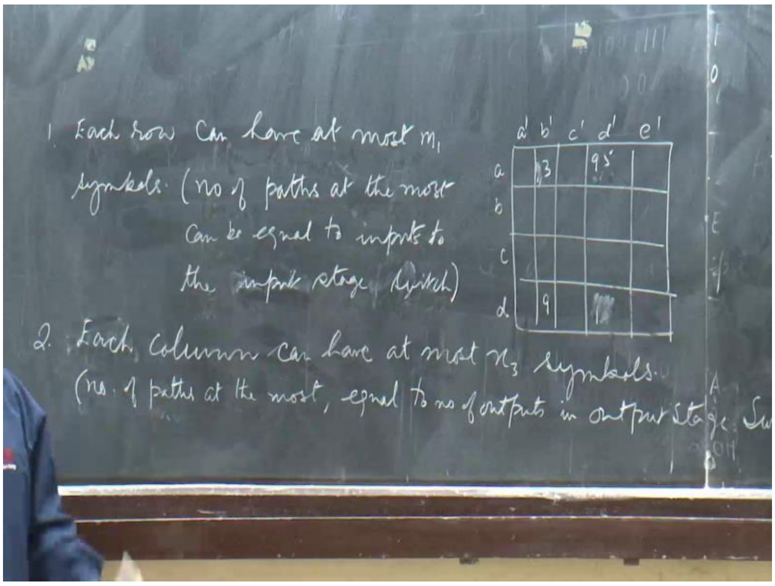
No, there is no multicasting, because this he is setting up 1 connection this way, and 1 connection this way this is permitted by this crossbar this permitted. So, I can actually put d to b if I want to put it I can put 9 here. So, conditions actually are coming on columns and rows, what elements can come, whether; elements can be duplicated in a row or it can be duplicated in column or not. So, those are the validity conditions. So, I am now coming to a case, where this port whatever m was giving to b also wants send it to somebody else.

For example, b you forget this particular connection, same input is coming here this is a crossbar I am able to put a copy on this side also, that is the multicast case. In that case a to d prime has 9, a to b prime also can have a 9, which actually means; I can have duplication of elements in the rows that is possible because of the multicasting. But, duplication within column was not permitted, I cannot combine this and this signal and send it over signaling to d prime that is not possible, it will create interference.

So, I am trying to emphasize, get whether elements can be repeated again in rows or columns or not, they can be repeated in rows within rows they can be, but within columns they cannot be. And this repetition only happen if it is a multicast scenario, not in unique cast case, unique cast it will be only coming exactly Once why it is going to come exactly once?

If you look at a 95 has already come, there is no multicasting if 9 has already come, I cannot use this link to set up any other connections. So, this 9 cannot repeat anywhere in the in the row. Similarly, in column I cannot repeat nine have this link is already being used, to set up some path from b prime to some input port. If this 9 is already use, I cannot also repeat if it is a unique cast case.

(Refer Slide Time: 14:51)



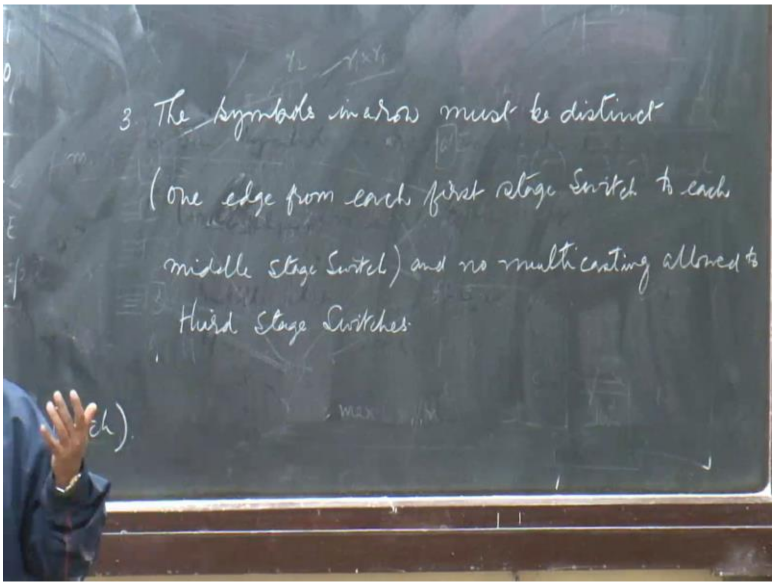
So, now let us write down what we will the legitimate Paull’s matrix conditions. I am formally writing them now, I explained by example. Each row will have how many elements tell memory that at most, number of elements r 2, row 2 can be larger than usually going to be larger or equal to [larger]- greater or equal to the number of incoming ports m 1. But, how many entries you can make here, can you put r 2 entry, you cannot set up r 2 connections only m 1 connections at most you can set up.

So, m 1 entries from those row 2 possible symbols any m 1 you can use, and they can be put in the row. So, that is a maximum size of elements in the row, except if it is a multicast then it will be duplicated. But, that actually means there can be exactly, only m 1 symbols which will be there the symbols can be repeated in a row if it is a multicast. Out of r 2 only m 1 symbols will be used, and they can be any combinations there is no restriction on that. If they all depends how you are setting up the connection.

So, first case is each row, can have at most m 1 symbols. This actually means number of max the path, number of paths which can be set up from a switch from input stage switch can at most the m 1, that is what it means. So, I can write that; number of paths can be equal to inputs to this the switch input stage actually. And which is the stage 1, I write simply in this way, input stage switch actually means only this switch not the whole switch, only this switch which is m 1. So, that is the first condition.

Next I have already told you, what about column? Give me the equal inter statement for a column? m 3 exactly, it has to be only m 3. So, what I will do is; the whole statement is going to be same. So, wherever there is going to be a, let me write it down, because each columns can have at most m 3 symbols. And this actually implies that the number of paths, at the most equal to number of outputs in output stage switch. So, you agree with these 2 conditions.

(Refer Slide Time: 19:05)



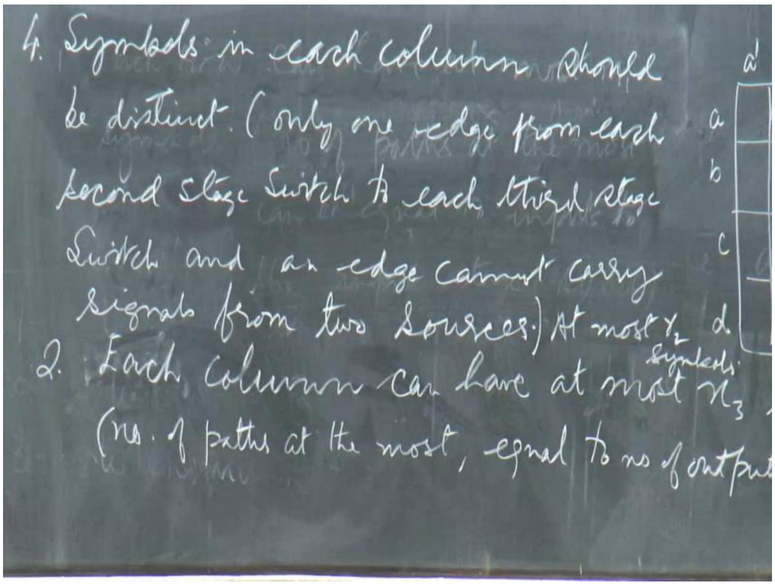
Now, 3rd condition is; the symbols in each row. So, I am not worried about weak symbols, there will be only m 3 symbol, m 1 symbols or n 3 symbols. But, they all symbols have to be distinct, if it is a unique cast case. So, this actually means 1 edge from each 1st stage switch to each middle stage switch, and no multicasting is allow in this stage. So, this actually means at the most r 2 symbols can be there. So, whatever the minimum of m 1 and r 2 that has to be here.

So, if do not bother about that condition at most r 2 can be this, because only r 2 symbols are there. And you are not having distinct thing, because there is no multicasting, so but usually it will be because I have already told you that m 1 is going to be smaller than r 2. So, usually m 1 symbols will be there m 1 distinct symbols will be there, and they will be chosen from r 2 possible symbols. And they will all be distinct they will not be repeated actually that what it means, because there is no multicasting

Student: ((Refer Time: 21:38)).

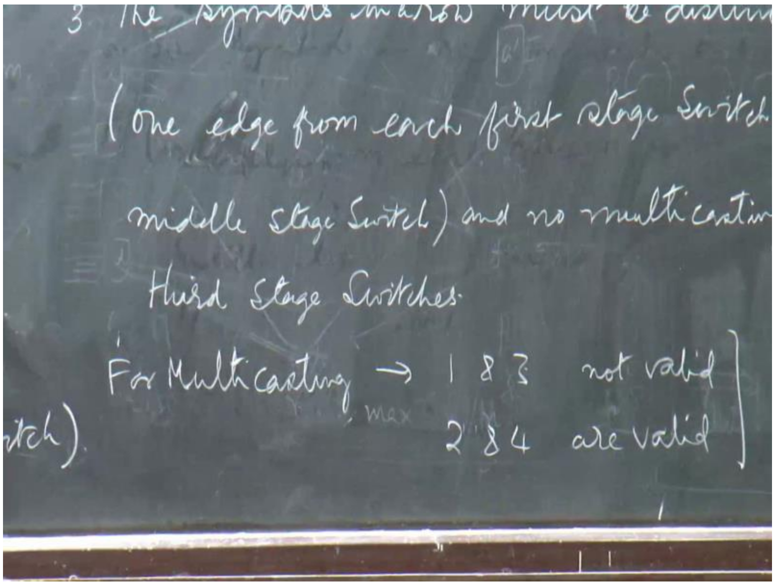
There is no relation, they can be independent they can be independent. You can generally you can make a general proof on that. Usually, in most of the actually implement this is they will all be equal m 1 and n 3 will always be equal, r 1 will always equal to r 3 usually. Symmetricity will be always maintained in actually implementing, because this is, but there may be cases where you do not want asymmetry, asymmetric cases are possible.

(Refer Slide Time: 22:25)



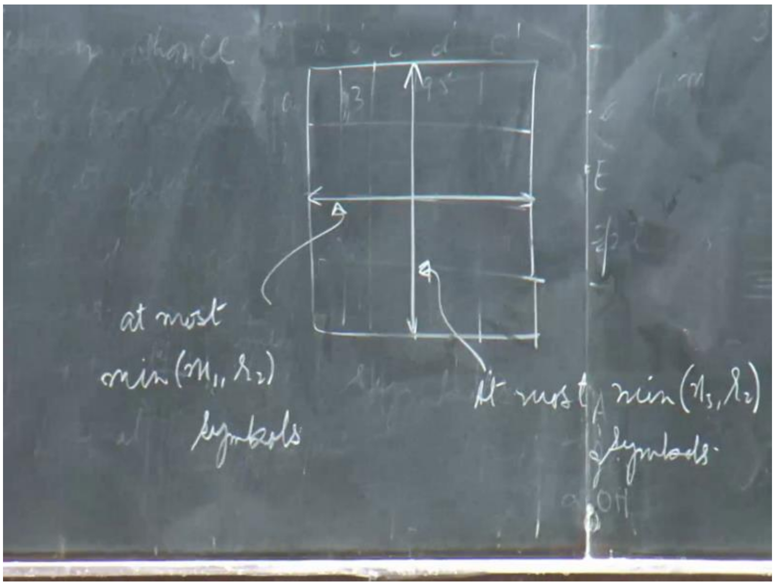
So, now let us come to 4 th condition. Symbols in each columns should be distinct. Only one edge from each 2 nd stage switches to each 3 rd stage switch. And an edge cannot carry signals from 2 sources, and that also means implies at most r 2 symbols if you do not bother about the other statement, when you combine you get mini max all kind of combinations. Independently, at most r 2 symbols can be there distinctness. Only if the multicast is there condition number 1 and 3 will not be valid, but 2 and four will strictly valid that is the final thing.

(Refer Slide Time: 24:19)



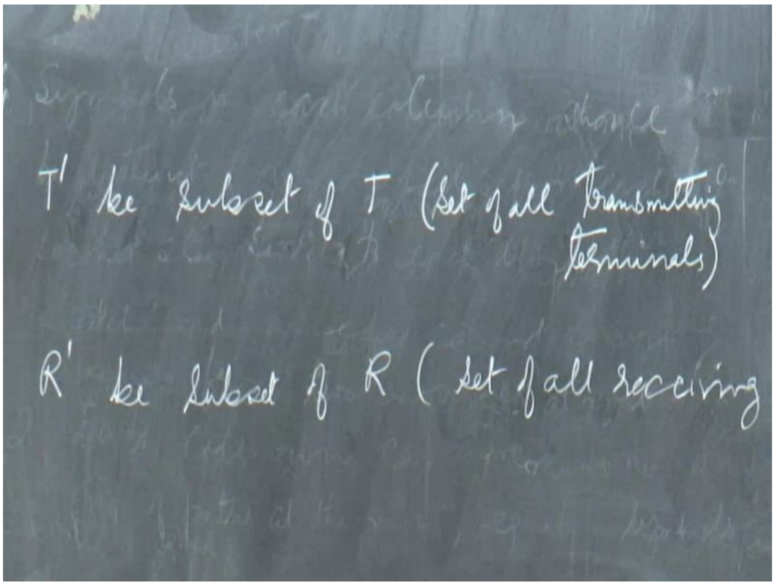
For multicasting 1 and 3 not valid, but 2 and 4 are valid. That is the only change, you can have more than m 1 symbols in the row. Once it is multicasting and they may not be distinct you can repeat them. So, they will not they will be more than r 2, that actually what it means. They can become more than r 2. So, as a consequence, if I put everything in a figure I can simply write this thing else.

(Refer Slide Time: 25:10)



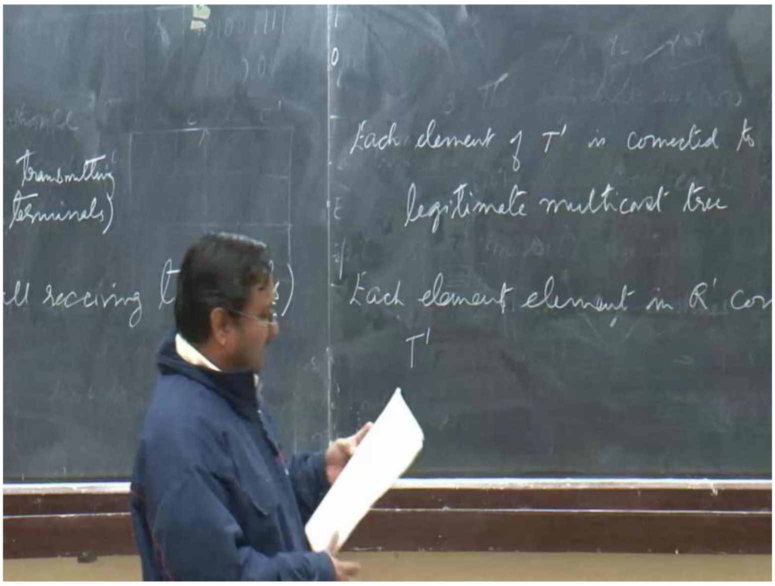
So, in a row you will have minimum of m 1, r 2 symbols. At most minimum of m 3 and r 2 symbols, that is what it means actually. Now, formally writing down the definition for a strictly non blocking switch, and re-arrangeably non blocking switch, I have defined the sets of the ports. So, let me give the definition and then state the statement.

(Refer Slide Time: 26:14)



So, let T prime be subset of T, where T is nothing, but set of all transporting terminals or input ports. And there will be R prime this can be any subset, where many possible values of T prime and R prime. So, R prime be subset of R, where R is set of all receiving terminals, and there is a state of the switch.

(Refer Slide Time: 27:15)



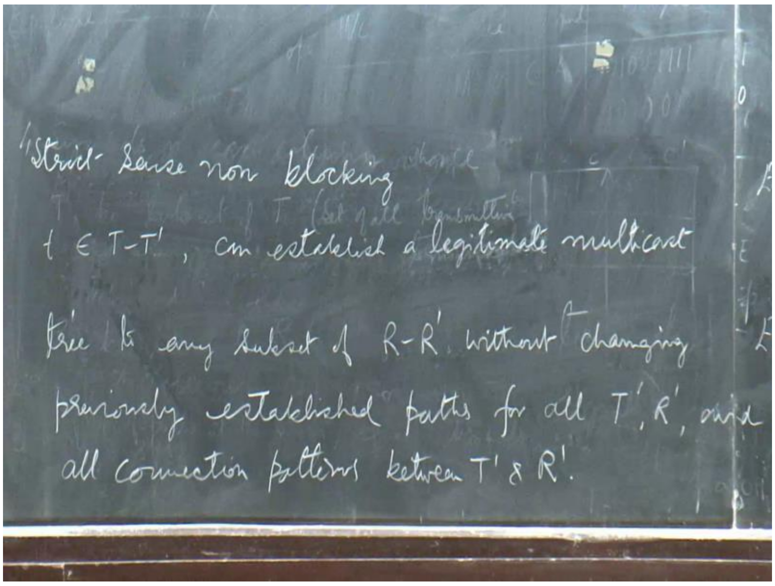
Now, state of the switch is nothing but this that each element of T prime this is basically T prime means all ports, which are busy, R prime set of all ports which are also busy. So, what will do is; each element of T prime is connected to. So, connection how we write we write a legitimate multicast tree is there from each element of T prime to elements in R prime, that is the way it has been written actually. So, that is only formal language we do whatever I am saying otherwise; is actually means the same thing. R prime by a now this is legitimate all possible conditions are satisfied.

So, if it is multicast even those conditions have satisfied. So, multicast unique cast everything is taken care of by this. And usually number of members in R prime most of the people think the cardinality of R prime and T prime is going to be same. The number of members in that set, because 1 port is connected to 1 outgoing port, but in multicast this may not be true.

So, cardinality may not be equal actually. So, that is why it is a multicast tree actually, which has been mentioned. That is 1 important 3 that each element in T prime is connected to R prime through an estimated multicast tree. So, but R prime can be more. So, there should not be any element in R prime, we should be left out. So, for that there is n we have to think that each element in R prime connected to n element in T prime. So, an element in T prime can be connected to multiple elements in R prime.

But, an element in R prime is always going to be connected to only 1 element in T prime. So, you can only split in the forward direction, while doing you cannot combine actually; combination is not permitted, but splitting of signal is permitted. That is what it means, and that is what the multicast essentially actually also implies. Now, with this set definition now, let us define what strict sense non blocking network is? Then we will go to the proof of that.

(Refer Slide Time: 30:03)



So, strict sense non blocking, and element t which belongs to T minus T prime a free incoming port, that is what it means symbols and everything looks very dangerous. But it is not this only means a free incoming ports, any free incoming port is what it implied by this thing. T prime is a set of all busy ports T is set of all ports you remove, these all busy ports this is set of all free ports, T is a member of that, t can establish.

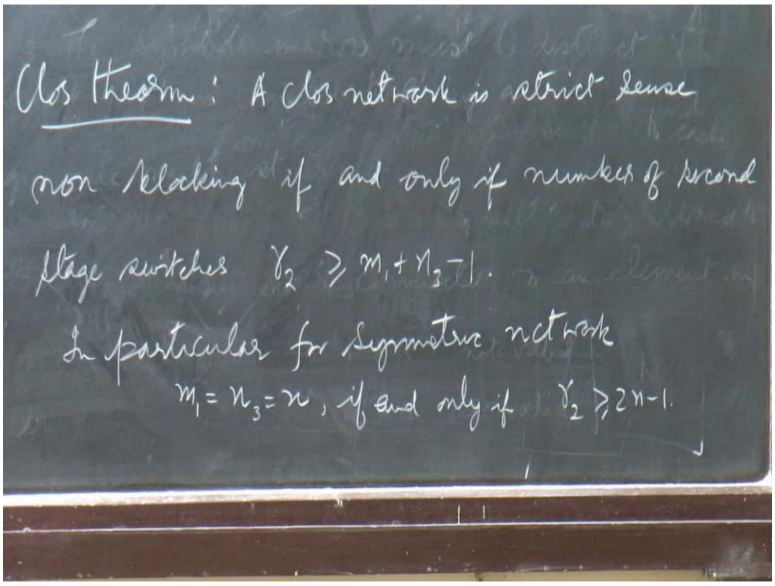
Now, that is very important it cannot not only establish 1 to 1 connection, the way I defined it earlier I have told you that there was 1 input port there was 1 outgoing port both are free, I should able to set up the connection. No, that is not the only condition, if this is a free input port there are many outgoing free ports, for in this free ports for any set any subset of the free ports, I should be able to set up a multicast tree. That is the condition for strictly non blocking, not point to point connection.

So, I am now deviating from whatever I had said earlier. So, this more this will be much more proper definition of strictly non blocking network. So, I should be able to set established legitimate multicast tree to any subset of R minus R prime that is more appropriate definition. But, this is still not defined what is a strict sense strict sense now I am defining without, changing previously established paths for.

And this should be true for all possible T prime. Because T prime also can be all various different possibilities which are exist, for all. So, I have to now qualified T prime, R prime and all connection patterns, between T prime and R prime. This is the I think a complete definition to the very much complete definition. And instead of without changing, a few say by rearranging all the existing connections from T prime to R prime. So, that all legitimate existing multicast trees are still maintained.

But, the connection pattern may be rearranged or maybe redone again, and then if you are able to do this establish a new an estimate multicast tree all the time, then that is the rearrangeably non blocking switch. And there is something called wide sense non blocking, I do not have a formal definition, but you can try it writing it out. I will give you an example of wide sense non blocking switch, I think later some time. And that is the only example which I am aware of as of now.

(Refer Slide Time: 34:45)

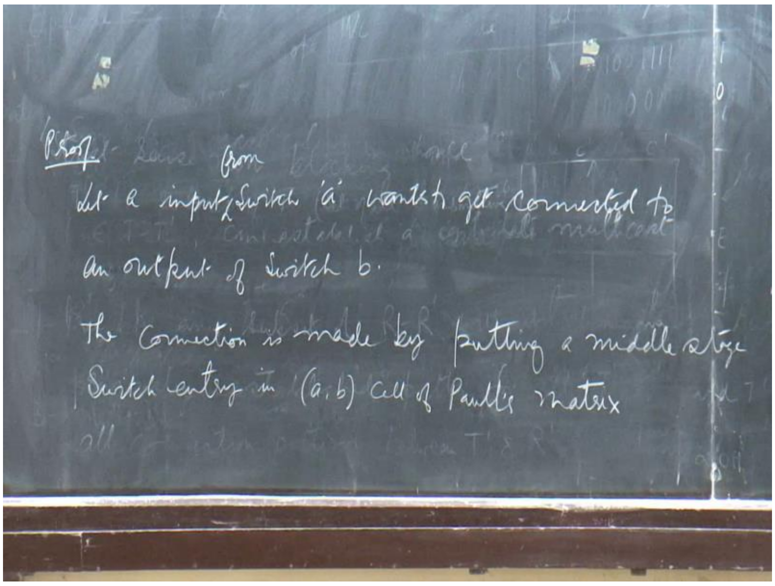


So, the clos theorem, I can now state the clos theorem, once I given this definition. So, the a clos network is strict sense non blocking. If and only if necessary, and sufficient only if means its sufficient, if it means it is a necessary condition. So, both we have to prove, necessity as well as sufficiency both. This I have not stated earlier I only told that this happen this will be there, but this is the sufficient condition. I have not mentioned that. If number of 2 nd stage switches r 2 is greater than or equal to, I think you will should be able to appreciate that new expression.

It is for asymmetric switch remember, it is a asymmetric it is a generalized formulation. And I can write in particular that is what I have told you earlier, for symmetric network m 1 is equal to n 3 is equal to n, if and only if, r 2 is greater than or equal to 2 n minus 1.

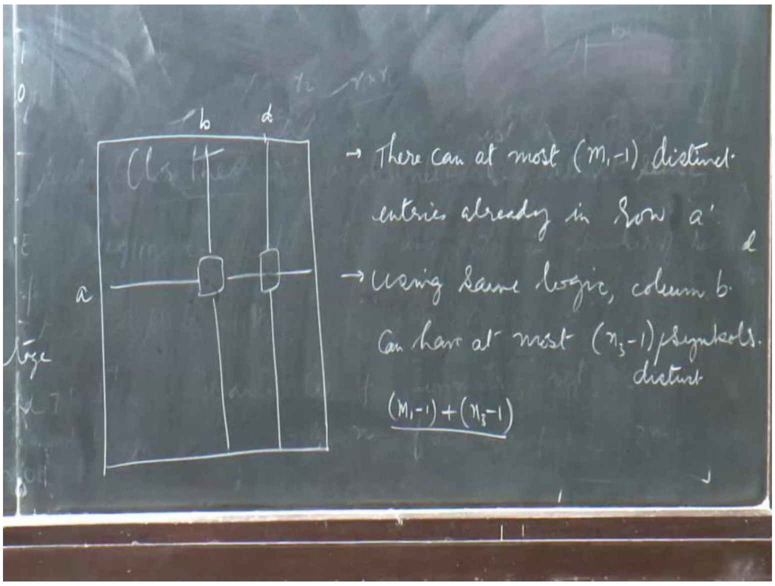
So, that is the definition. Now, formally I can just prove it, because I have now Paull's matrix I have Paull's validity condition everything there with me. So, let us use that and quickly prove it. So, at least we should be able to close with the clos theorem today.

(Refer Slide Time: 37:31)



So, proof I am going to write it down I could have only spoken it out that is fine. So, let there be the input from switch a, who wants to get connected to an output of switch b. So, a input from. So, how you will set up the connection? We will take the Paull's matrix I think, I can remove this particular stuff, and this figure that will be required.

(Refer Slide Time: 38:38)



So, there is something which is common here, where I have put the entry. Write b does not matter or you can put b dash also it is ok. I am just putting a b. So, you will set up the connection is by you take any middle stage switch and that entry has to be put here. Any middle stage switch I have to put an entry without violating the Paull's matrix condition. So, if I am able to find out an element a free element or free symbol from the middle stage switch, which can be put without violating the thing I can always set up the connection.

Now, you have to only figure out that when all possible scenarios you should be able to build up this. And you are actually figure out what I am doing. So, let me just you can actually note it down, I think it is better to write, because this text is not there anywhere else. So, for the people, who will be looking at the video recording this will be important, you have that text other people do not have. So, connection is made by putting a middle stage switch entry in a b cell of Paull's matrix. That is what have to be done, I am not worried about now duplicating intern, because this is a multicasting entries will be repeated in your row.

So, now when I am going to write this statement, I will use a word distinct before the entries. So, there are duplications 1 is repeated 3 times, I will took only 1 distinct entry. So, that becomes very important. So, the previous statement when I written the distinct word you have to always figure out whenever; I write distinct what it means. Because, you are going to even handle for the multicast.

So, presume very important here, there can be at most because there is 1 input in switches, which is free which we are trying to connect. So, worst case scenario m 1 input ports must be busy, m 1 minus 1 input ports must be busy. So, that actually means there can be at most m 1 minus 1 I am going to now, put a word distinct to take care of multicast scenario now.

Earlier, I would have say m 1 minus 1 symbols would have been there at most, m minus 1 distinct symbols can be read at most, because in multicasting the symbols will repeat. So, number of symbols not important, number of distinct symbols what is important. So, I am going to put word distinct entries already in row a. So, this is equal to number m 1 input minus 1 input which wants to get connected. m 1 input minus 1 input, which wants to get connected.

Using the same logic, I can say that column b can have at most n 3 minus 1 symbol. And n goes to worst case, what can happen is; this m minus m 1 minus 1, and this n 3 minus 1 distinct symbols they will always be distinct actually, does not matter because there will not be any repetition in the column. 2 and 4 conditions for the Paull's matrix are always valid, irrespective of whether it is unique cast or multicast.

So, total number of symbols in worst case scenario, no symbol in m 1 minus 1 in this set. And in this set there is no symbol which is common. There is distinctness they all are distinct. So, total number of symbols, which are occupied are m 1 minus 1 plus m 3 minus 1, I think you can understand what I am doing, what is this I am just writing whatever I have said earlier in earlier lecture.

So, these many total symbols are already occupied or busy. If number of available symbols going to be more than this, you will be able to find ones symbols 1 those the symbols, which are not consumed here. Because, if the number of symbol available symbols are more than this, there is hardly some symbol which is available now, and that symbol can be put in this cell. And I can set up the connection, and this can always be done this can always be done.

Now, as far as a multicast tree is concern, what actually probably instead of this I can further say let a has to connected b and d somewhere. So, I have to put entries here, same entry also should repeat here, same entry should also repeat here. that is most important. And then I can create a multicast tree.

Student: ((Refer Time: 45:31)).

It is for to creating for the multicast.

Student: ((Refer Time: 45:39)).

Entries and symbols are same, any symbol which is entered there that is; what I am talking about. Otherwise; it is a distinct entered symbol or distinct symbol is also fine not an issue. These 2 even for multicasting, clos theorem is to even for multicasting. So, this actually means r 2 has to be greater than m 1 plus m 3 minus 2, first switch to be strictly non blocking, which actually implies that r 2 to be has greater than or equal to m 1 plus m 3 minus 1. I have put the equal 2 conditions that is I write 1 there. And if it is symmetric case it becomes 2 m minus 1, because I am using m not n, so, its 2 m minus 1. So, this is the same condition, which I had told you earlier. So, you find out the worst case offset a and then maximum overlapped, and based on that you find out. And since this entry this new entry has not been used by, nether in this column or in this column. If you look at for d similarly, you will be able to find out, because d is also free in worst case offset you can find out always some entry which also can be put here.

I need not put the same entry here and here remember. If I am trying to set up a connection with d. So, whatever number of symbols which are used here, and number of symbols which are used here, worst case scenario will be m 1 minus 1 and n 3 minus 1 and whatever is missing that actually can be used here. I can even repeat the same symbol if it is available otherwise; I will use some other symbol.

Student: ((Refer Time: 48:07)).

Multicasting enable repetition is permit limitation repetition actually means that. So, now let us that free switch with the clos theorem only thing is the multicasting, I will try to elaborate on that in the next lecture slightly more. I have only told about this as for unique cast function. So, let me also look into more into the multicast scenario, specifically, because that sufficiency and condition has not been still proved on formality cast. And then we will go for slepen duad theorem re-arrangeably non blocking switch condition.

Student: ((Refer Time: 48:46))

Yes, this condition currently for unique cast, I have not told about anything about multicast, but this will be true even for multicast, but I have to formally prove it. So, I have to at least give an argument, how it will be. Because, you should actually ask a question m 1 minus 1 are occupied here, there is exactly 1 extra entry which is free which we have used here already.

These entries are there now, question is it is possible that this particular entry, which have been put here, have already been use somewhere here. But, then in that case the condition of distinctness between this and this will not be maintained there is something, which is already common. So, that entry cannot be, but I have to just actually articulate that. That articulation, so far has not been done for multicasting scenario.

So, far proof this stands only for unique cast. So, we have to figure out whether this condition is going to be true for multicast or not. So, I also leave it to you, because this is as of now till now whatever; we have discussed its only showing for unique cast. Multicast I have to give the case that it is always going to be possible. So, we will go further in the next class from here onward.