Indian Institute of Technology Kanpur

National Programme on Technology Enhanced Learning (NPTEL)

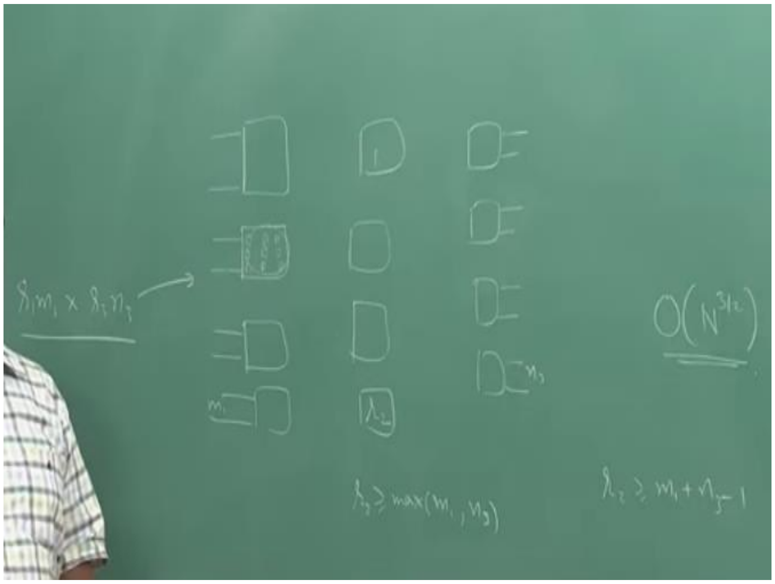
Course Title
Digital Switching
Lecture – 20

By
Prof. Y. N. Singh
Dept. of Electrical Engineering
IIT Kanpur

Okay so in the previous video what we had discussed was, Slipean theorem and which gave the condition for rearrange non blocking switch okay. And we had also done a few examples where, I have shown that when we you end up in a blocking, how to identify the rearrangements .So we actually had looked at the chains are algorithm so over the once chains are chain then alternate chains search and then based on that to identify what all reconfigurations has to be done okay and of course then I also.

Discussed about how the looping algorithm is same as doing the chain search algorithm okay so in fact these are use any of those two it is going to function so far it is fine I am able to get.

(Refer Slide Time: 01:06)



Get a strictly non blocking switch by using Clos network which is a three stage system so any number of switches I can put and I am able to actually even show that for rearrange non blocking switch rearrange non blocking Clos network how though what will be the condition I have shown that that r_2 has to be equal to maximum of m_1 or n_3 so where this is m1 this is n_3 and number of switches are r_2.

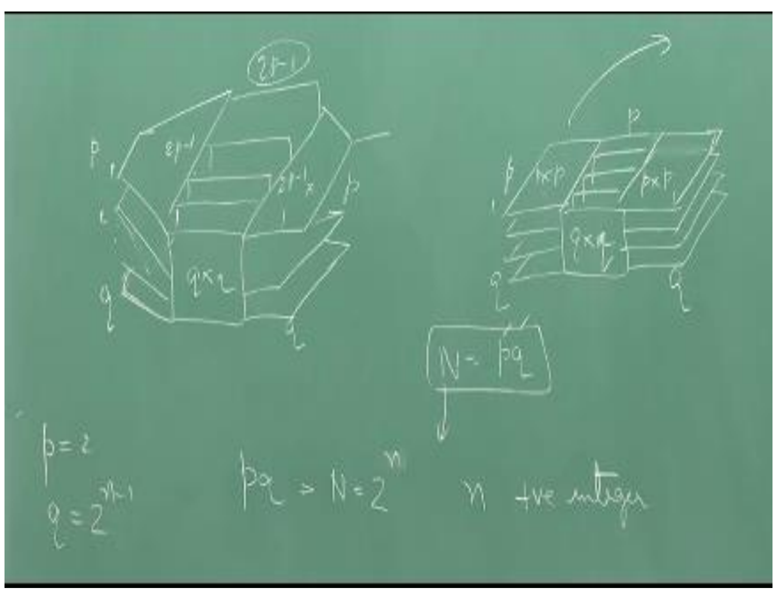
So this for rearrangibly non blocking switch for a sit non blocking condition we had shown that r_2 has to be greater than $m_1 + n_3 – 1$ okay and at some point of time I had also for if that is not look at this one as of no let is look at the strictly non blocking switch number of cross points which will be required will turn out to the $O(N)^{3/2}$ but remember when I computed this $O(N)^{3/2}$ this was actually is a cross bar a still of them are cross bar elements but I have replaced a bigger cross bar which was r_1 m_1 x r_3 n_3 this cross bar was replaced by a Clos network.

And I got a reduction in the cross point complexity so if I can actually take this alimented and also construct this thing by also a three stage network so number of cross points which will be required in side this rectangular switch or a cross bar will get reduced the way I have reduce for a bigger by putting a three stage to 13^2 I can recursively also reduce this then I can take a switch inside this also a cross bar I can also construct this thing by a three stage Clos network I can do it for all the switches actually.

In the same thing so I expect that my cross point complexity will be reduced from this particular value to something is still better okay now this kind of construction were actually I take a bigger switch and constructed using a Clos network then take each element and again further constructed using a Clos network within that particular thing if there is a smaller switch again constructed using a Clos network is the recursive construction so I am recursively constructing a bigger switching a smaller switch.

Okay so now we will be looking into the recursive construction and what will be cross point complexity because of that okay so first fall we will look into rearrangibly non blocking switch and that two ever is simple version okay so now these actually now can be return also drawn in this sense that I get.

(Refer Slide Time: 04:46)



So this p has been explained into 2p – 1 because I wanted a strict non blocking switch and then I had so this 1 2 till q of them so I will have q/q switches and these will be 2p of them 2p – 1 q/q switches which will be there again I will be using 2p – 1 by p switch the p outputs which are coming and they are q switches so this is another way of representing the same Clos network okay so any feeds going to rearrange one so this condition is what is going to be satisfied in this case it is a symmetric system.

That is what I have taken so I have taken n3 = m1 and in that case this is how it will look like so p is the equal to m1 here = n3 okay so similarly I can actually draw it for a rearrangable switch so rearrangable switch it will be like this squares so this will be p/ so this p/p switch so p of them will be their these are q/q so these are q switches of p/p so again these are p/p and these are going to be q switches so this will be rearrangibly non blocking configuration ,okay .So when this condition will be satisfied I am again taking a symmetric condition ,so we are going to solve it for a symmetric condition.

Not for the symmetric one so for this switch what is a number of cross points which will be required okay so when I have to create recursively I have to decide what will be the value of p and q when I do the first break up when I write because I am essentially been given n/n I have to break it up into p x q basically p x q will be equal to n here so what will be p and q I need to choose.

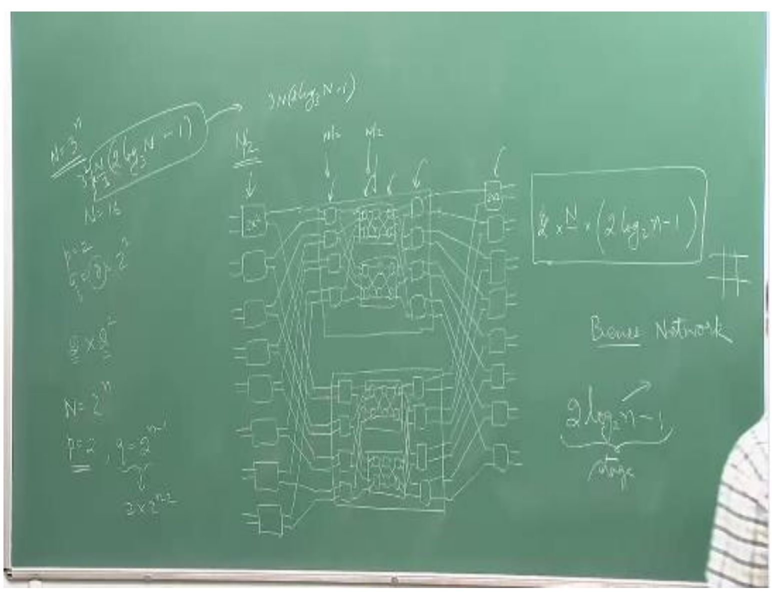
Once I choose that I will get p/p switches in q/q and p/p which will be required each one of them will be further broken down into smaller segments so by fragmenting in further way I'm doing it for n/n okay so x in this case number of cross points so this one required p2 cross points and q say switches next one require q2 it is q/q p is a switches last one again require p^2q so this will be x = 2p2 q+ q2p so these mean number of cross points will be required so the p and q need to chosen so I can actually take p as a small as prime factor because this the indication which is coming from here this twice the number.

P2 is twice so p should square should be chosen as the small as possible okay so q2 can be large because this twice actually here okay. So I can actually do the prime factorization of p and q and then chose the smaller one this is okay for rearrangibly blocking switch but this is not okay for restricted on blocking switch okay what the problem which will be happening in a strict on blockings if I choose to small a p okay the q will become very large in this case okay if you small p is chosen.

If you chose a smaller q the p will become very large so this width will become actually very large in that case okay so you have to essentially find out some optimality we will come to what will be the optimal value what could be the best possible choice which we can have so for rearrangable network what we can do is we can actually take n as n actually of course can anything but I am going to take $n = 2^n$ so where n is going to be an integer n is going to be a positive integer.

So in this case I can actually this I have convert into p x q so I can take p as to so q will become 2n – 1 okay so I will get p/p basically 2/2 elemental block I need not fragment it any okay I have to take only this q and then further do it by recursive constructions this can be further factorized by 2 x 2^{n-2} and so on, okay. So let us try doing it or some switch and see what we are going to get, so if we take for example.

(Refer Slide Time: 11:15)



N is taken as N as to be 2^{n} so I can take n = 2^4 which is 16, okay. So if n is 16 what will happen. So n is 16 so I have taken p = 2 so Q = 8 which is 2^3 now, so 2^{4-1} 2^3 so I will have 2/2 blocks. So I have got but I have to also do the same thing in the third stage so I will be building up 8 of them on the third stage also. Yeah i9 have got 8 of them so what will be the Q/Q which is going to be there as per that it will be 8/8.

So how many of them 2 of them will be required. So there will be 8 lines, so I can connect like this, Okay same way I am going to do it on this side. So this is what will be the structure, so I have got 2/2 here which is P/P and I have Q = 8, now because I wanted to recursive thing so I can actually take again this Q in N and further do a prime factorization, so this 8 has to be further prime factorize.

So it will become 2 x 2^2 which actually implies I am going to now create 2/2 4 such switches and I need 4/4 here so I will connect, okay. So again using same Clos concept same I will be doing it here. So I had one stage here so every time I am actually reducing I am actually getting two stage then these two have been added now I can actually interlay take this 4/4 and again create a 4/4 block.

So I can connect like this, so I have created two more stages here and then the middle one, now this particular network which is been created recursively when N = 2^{n} and P is being taken 2 every time and Q is taken every time is equal to 2^{n-1} this further broken 2 x 2^{n-2n} so on, what we get is a Ben's network, okay. So this is what we call Ben's network. So this built with basic building block of 2/2 cross bar.

Now if I actually this is the middle stage so the total number of stages will be every time there is two so it will be log 2n comes from 1, 2, 3 and 4 the middle stage if you count, so middle stage this four on this side also counting the same thing, so twice of log 2m and since the middle stage is common minus 1 these many stage will be there and this will be the structure of a Ben's network, so it can be done using a recursive construction and this will be rearrangeable in non blocking because remember when I did the outer one using two sided by 8 it was re-arrangeably non blocking.

I was satisfying this slip into it condition, each one of the internal block is gain done using the same technique, satisfying the same condition therefore it is also rearrangeable non blocking, so this is what is a recursively constructed re-arrangeably non blocking switch, total number of stages will be this much, how many switches will be there in each stage because I am dividing by 2.

So there total 8 here so there n/2 switches which are present here, this was a 4/4, 8/8 block so there are four switches here four here again there is 8 n/2, set third stage n/2, so every stage has n/2. So in total, total number of switches are 2 log 2m- 1 and h 2/2 cross bar is a basic cross bar so it requires 4 x 1 so I can put 4 here, okay. So this will be the number of cross points which will be required in a rearrangeably non blocking switch.

Of course I am assuming the condition is the prime factor is 2, so if n is 3^n if nh is 3^n then every time you will be actually you will be dividing it n/3 switches which will be present and number of stages will be twice log 3m – 1n/3 so that will be number of switch is which will be present each one of them required 3^2 so you will end up in with th3 3/3 as a prime factory it will be 3 and 2 log 2_n – 1 cross point which will be required in this case it becomes 2n 2log 2n -1 in general you can actually observe the number of cross points x the order of complicity her in this cases and log n of course the be value is can be 2 or 3 does not matter I can actually put in general it is log n .

Okay so because I can always convert log 2 n to something log of 10n in to something it can always be done log 10_2 which can be actually multiplied divided by that okay. So in general the complexity of here recursively built re arrangable non blocking switch will be of the form o and log n now interestingly if I actually I will be using this structure remember this is from any input I have can go to any output and there are more than one possible paths which are there.

But if I actually buy for gate if I just simply look at the middle stage and remove this particular path which in the right side whatever I will get some number of input and output will have exactly one path between them okay and this particular network which is there on the left side is known as base line if I just remove this particular part, okay and instead of drawing a line here if I draw a line on the other side if I draw a line here and remove this particular part then what I will get whatever is on this side in known as inverse base line.

These cannot be use for circuit switching because there is they are not multiple part they are all blocking switches base line is a blocking switch inverse base line is also a blocking switch but they actually can be used a self routing switches for the packer switching will come back to the self routing switches in one of the week we actually have shed hold it, okay so this is as for as the rangable non blocking switch which is concern.

But we can also do similar thing re cursive construction for excepted non blocking switch of course you know how must we puzzle that how this will be done in this case live I simple I took n= 2^n to prime factor p but if I take p to be very small okay so this will be small but q will become very large okay if I take p very large okay this middle switch will become very large but q/q will become small okay.

So we have to figure out what should be p and q if I want to create a recursively constructed expected to non blocking switch so let us try something actually let us try to find out what I should I am anyway computing a bound so whatever value of p and q I will chose here so whatever is my limit I will be always lower than that, okay I am trying to find out what is my best possible number of cross point bound which I can get.

So if I even deviate from there I will became only verse than that okay so it is only a bound so it is possible to got better but we do not know how will go actually okay but certainly we know the limit key okay at least this much I will be able to get how to go below this we do not know but we know this at least this much we will be able to go the verse performance can always be there okay so I am trying to find out whatever best I can do.

So the cross point for this one so how many cross point will be required for a p x2p-2 switch so this will be px2p – 1 this many cross point show many switches are there q of them middle switch there are 2p-1 switches each one require q^2 and then just trying to make an estimate what p and q I choose okay and then of course I can put q^2 switches 2p-1 x p so this will be total number of cross point if I am not doing any recursive construction.

Let me try to optimize my p and q her it is self so I can now do it first thing which I will do is instead of 2p if I use to 2p-1 this I have replace by 2p I am only adding extra switch okay so I am trying to find out the best performance best possible bound on cross point the what is the minimum by which I can get this I am only making a verse it is okay so I will be estimate straightly here which is fine.

But it will be make my life easier so I will replace this 2p-1 /2p on both sides everywhere so I will get q $2p^2 + 2p\ q^2 + 2p^2q$ it becomes $4p^2q + 2p^2$ I can just try taking the re votive and try to find out what should be p and q for optimizing the value of x so I can do that I can actually put l x (δ p) = 0 so which tern and remember now n= p x q so q I have to put n/ p actually if I need to solve this equation because p and q are both related.

So I will put $\delta/\delta p$ of $4pn + 2n^2/p$ so this will be $4_n + 2_n^2 - 1(p^2=0)$ so this Trans out to be 2 in fact 2 will come out I will have $2 - np^2 = 0$ which implies either this should be 0 which is not the case which is not our solution or this should be 0 so this actually indicates that so let me write it here $2 - 2/p^2$ is 0 which transfer to $2 = n/p^2$ so p is n/2 which actually will give me $q = \sqrt{2}n$ some kind of a square root kind of factor but this mostly like should give me optimum.

But we cannot get a close from solution if I pout these values of p and q I have to remember recursively re constructed the whole show. So what we will do is I will use something else only thing is that if gives me the best I will be only doing slightly worst performance but at least I know that my best is going to be a better than this okay it cannot be worst than whatever I am estimating okay.

So I know that this whatever I am going to estimating my best is going to be always lower than this okay I know I am going to get a bound so this is the same reason why I did this particular thing so I wherever is the optimum I try to move upward by actually putting more cross points here by removing this n -1 so my best was always going to be lower than this okay if something becomes better I need to make a proof here I am making an approximation.

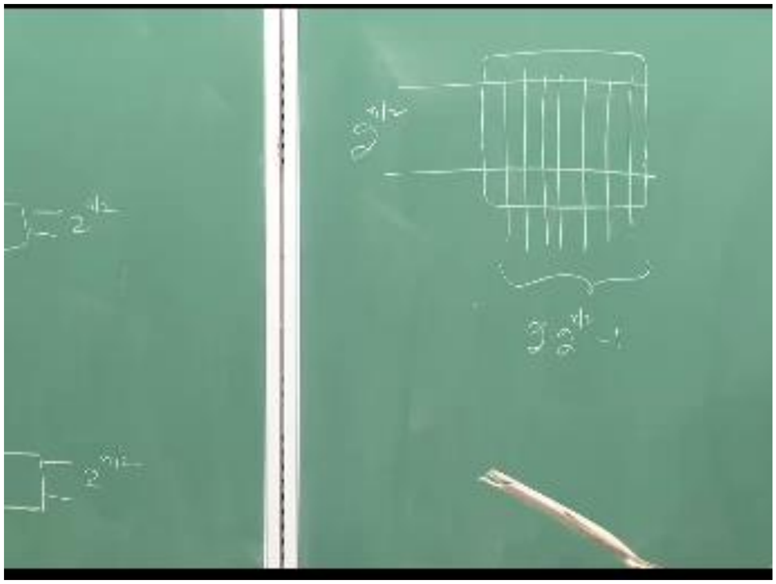
So that I can solve the problem so instead of taking this $p = \sqrt{n/2}$ and $q = \sqrt{2n}$ I will take $p = q = \sqrt{n}$ so now p x q will be n in this case okay so I am going to work with this scenario and I will get a close form expression for the cross point complicity so at least I know that whatever is my best is going to be lower than this okay at least in this, that much I will be able to say, so let's start doing this exercise, so we can actually again as a convenience, take n is equal to 2^n, of course it can be anything, it can be 3 less power n but then I'm taking it this way so that, life is easier for me. And I will also take this is small n, such that n is 2^n, it actually solves, simplifies my situation.

So where l is equal to be an integer, l is a positive integer, so now what is my p, square root of n which is 2^n, so what is going to be my q, is going to be same thing, so my switch will be first state switches will be of size $2^{n/2}$, what's going to be the output, $2*2^{n/2-1}$ and I will create, so how many switches, $2^{n/2}$ switches will be present. That's how I affected it actually. These are again number of inputs.

This $2*2^{n/2-1}$, so then I will require switches here, so I will require $2^{n/2} * 2^{n/2}$ switches, how many of them, so last one will be 2 into whatever is this number, them I can go to the third stage and what will be the outcome, $2^{n/2-1}$ and then I can connect it, each one can be connected in this pattern, I got the cross network, so how will be constructing this particular switch, let's see. Remember here numbers of outgoing ports are more; numbers of incoming ports are less.

(Refer Slide Time: 36:23)

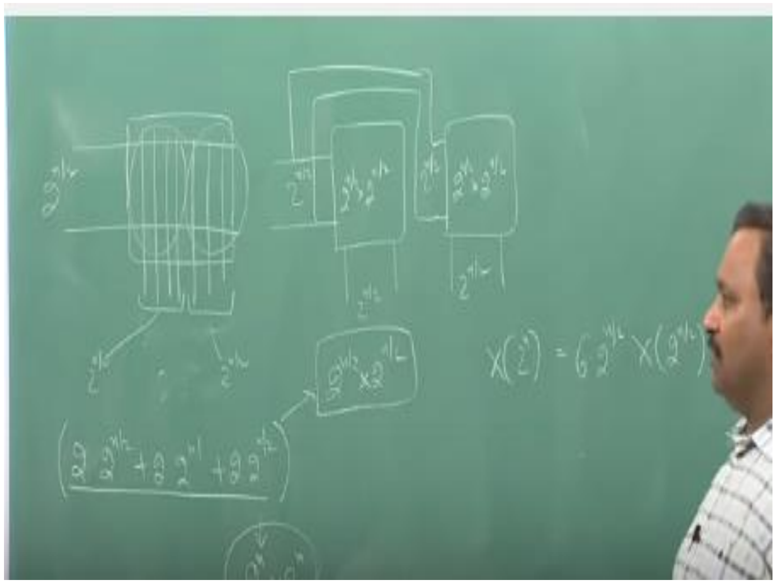
(Refer Slide Time: 36:50)



The way I can do is, I can take a cross bar, so number of lines is $2^{n/2}$ and these number of lines are $2^{n/2-1}$, I can now take a simplification step, see I'm anyway trying to estimate the bound, If I actually put one extra switch does not matter, so let me put one extra switch here, so total number of switches is $2^{n/2}$. So whatever I'm going to estimate will not be better but slightly worse. So I know my best bound is going to be better than this.

So I can, this will simplify my situation, this -1 will go away and this -1 will go away, and this is one extra switch, which is going to be put, so you will count from here, 1, 2, 3, 4, 5 to n/2, so that's the first simplification, so -1 will also go away and this -1 will also go away. How I'm going to build this? So I'm going to remove this, so I'm going to remove this, so half of them will become $2^{n/2}$, half of them is $2^{n/2}$, but you can clearly observe, that this is one square cross bar and this is another square cross bar.

(Refer Slide Time: 38:17)



I can very well actually make it, using you take the input $2^{n/2}$, take the output $2^{n/2}$, take the same input here and get, so both are exactly same, only thing now this inputs which are coming, I just actually found them out and put into another square switch. Here they were, actually internally coming and shared across, these many numbers of lines. So I need two of these, 2 of $2^{n/2 \text{ into}}$ $2^{n/2}$, so this is the switch size, two of them to build this element.

This can be build by the taking two such switches, so first stage requires how many such switches, so these many numbers, so $2^{n/2,}$ so each one of them will require two of these, so these many number of switches will be required and size of each switch will be this size, how many will be required in the middle stage, the same size which I have used here, these many numbers. Here also taking the same logic, which I have used here.

I require $2*2^{n/2}$ two switches and each one of them will be in size $2^{n/2}$ and $2^{n/2}$. And this is strictly non blocking, this is satisfying the condition and middle switches are more than the twice of the input or output whichever is the maximum or we say the sum of these two, this plus this -1, greater than or equal to that. So, when I remove -1, it actually is greater than or equal to, this is greater than the value.

So it still remains a strictly non blocking, so in general how many switches are required? So I require $2^{n/2} + 2^{n/2} + 2^{n/2}$ switches of size $2^{n/2} * 2^{n/2}$ so these many switches of this size will give me a switch of $2^{n/2} * 2^n$. So cross point complexity of this particular switch $x(2^n)$, now can be represented by whatever the complexity of this into the number of switches, so number of switches are here $6 * 2^{n/2}$.

So once I have got my recursive formula, I can actually use this and compute what is going to my total number of cross points in a recursively constructed switch. So let's try to solve this, so I have got my cross points, so it's a wonderful thing actually I know this number, I can find out the cross points, I can actually convert this cross points here in terms of cross points of smaller switch. So I can write down, so this should be, it is a recursion, which I'm doing.

Let's do it further, I keep on doing it, ultimately what I will get, and $x(2^2)$, $x(2)$. Of course just before this, I can write, so it forms a chain, it forms a geometric progression actually, so 2^2, 2^4, $2^{n/4}$ and so on, so I can actually now3 I start replacing all these what is going to come so I can put this value into this so what I will get if i put here I will get 2 * 2 x(2) if in keep on doing it so this will become $2*2^2\ 2^{4}*2*2^2*2*2^1x2$ so if I keep on doing it what i will be getting here I should get $2^{n/8}$ so now I can write them in brackets so like its simpler .

So on after just solve what is this value now remember the way I have return it earlier that $N = 2^n$to get $n= 2^l$ then l is an positive integer okay ,so now let me try to solve it so how many terms will be there we need to identify that so n/2is 2^{l-1} n/4 is 2^{l-2} and so on so I can very well write this is $2^{l-1,l-2}$ this should be 2^1 okay so l is an integer so total how many terms which will be there ,this will be 2^{l+1} and so infect better way of doing is actually don't do it now .

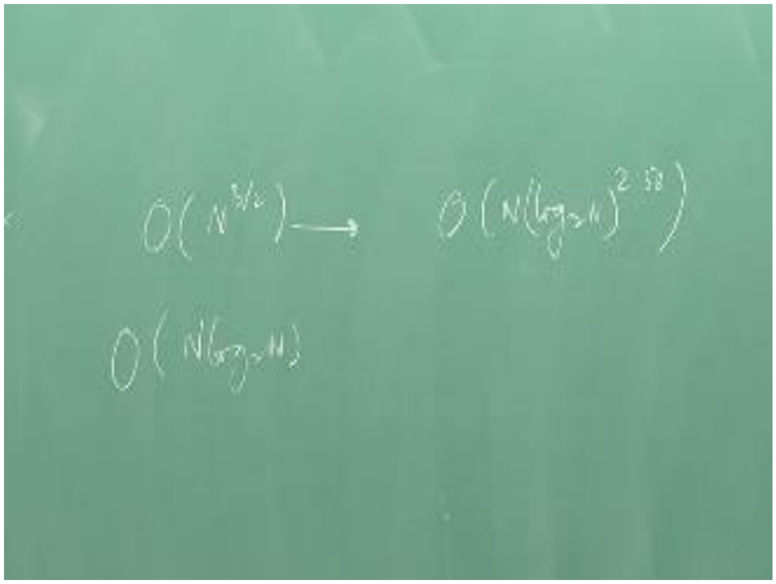
So let me do it so I will solve it from there itself so I will take the number of terms here so all these two's are I will take out , so I have to find out it depends on how many terms are there second is $2^{n/2}$ +n/4+n/8 so on +1 okay and since N is 2^l I can write this thing is this particular series now can be written as $2^{l-1}+2^{l-2}+2^{l-1}$,so there will be total l terms in this case there will be total l terms so this actually becomes 1 2^0 l-1 ,L terms I have to find out this summation .

This is nothing but a geometric progression and solution for this is nothing but the first term whatever is the multiplying factor every time that is 2 total number of terms are l-1/2-1 which terms to be 2^{l-1} so I can replace this thing so now these are l terms so this also must be coming l times actually so I can replace this by 2^{l} and this I can write as from here to 2^{l-1} okay, I made a mistake so there should be $6*2^{n/2}$ so every where this should have been 6 in the re correction okay not 2 .

So which actually implies this every where this has to be 6 so this term has to be 6^{n} okay so that was the mistake and that I think should be corrected so what I will have is now $6l*2^{l-1}$that should be the cross point complexity of n/n switch in terms of a cross point now this is the constant this will be a 2/2 cross bar technically that's what I am looking at and this cross point complexity will be 4 okay there will be 4 cross points which are required.

Solving it further now this 2^{l}=n ,so I will get 6^{l},2^{n-1}x2 6^{l} this is nothing but n/2 x2now for 6^{l} I can actually write this thing as now we have to understand the 2log2 of 6=6this is the universal condition so when I am going to put 6^{l} this will be 2^{log26}in to L okay which turn out to be nothing but 2^{l} is how much so 2^{l} is n which is nothing but log2(n) so we can now write this thing is log2(n) log2(6) n/2 x2,log2(6) actually can be computed and this value is a constant which is 2.58 so which implies that may cross point complexity is now is n log2n $^{2.58}$, so that's the cross point complexity for recursively constructed strictly non blocking Clos network okay so what we actually have got that we started.

(Refer Slide Time: 53:32)



With non recursively built it on non blocking that was on3/2 re arrange non blocking switch O(N log2n) and I improved on this bound and I have converted to O(N(log2n)$^{2.58}$) okay that's because of the recursive construction I will able to improve y number of cross points required to build the restrictive non blocking switch.

Acknowledgement

Ministry of Human Resources & Development

Prof. Satyaki Roy

Co-ordinator, NPTEL, IIT Kanpur

NPTEL Team

Sanjay Pal

Ashish Singh

Badal Pradhan

Tapobrata Das

Ram Chandra

Dilip Tripathi

Manoj Shrivastava

Padam Shukla

Sanjay Mishra

Shubham Rawat

Shikha Gupta

K. K. Mishra

Aradhana Singh

Sweta

Ashutosh Gairola

Dilip Katiyar

Sharwan

Hari Ram

Bhadra Rao

Puneet Kumar Bajpai

Lalty Dutta

Ajay Kanaujia

Shivendra Kumar Tiwari

an IIT Kanpur Production