Indian Institute of Technology Kanpur

National Programme on Technology Enhanced Learning (NPTEL)

Course Title
Digital Switching

Lecture – 28

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

In the earlier video what we were discussing was a cross bar and how it can be used as a packet switch and we specifically analyzed if we can actually put the packets in the input q because they are bound to be conflicts when more than one packet would like to go to the same output port okay and in that case only one of them can go out other one cannot go out if the speed a factor is one so we defined a speed of factor we defined input q we defined output q which happens when a speed the factor is = number of ports.

And then of course we compared these two and figured out what will be maximum through put which we will have to get especially input q has a limited through put you cannot achieve one while output q you can get the maximum through put okay the buffers at the output side will take care of the staging but one average the same number of packing will be coming for each outgoing port so it the q will always a stable for output queued system for a input queued then we though okay can which we modify in a strategy and then further improve through put we in fact investigated that.

And came of within idea that if a the load crosses a certain threshold then I should move over to dropping the packets which are contending and which are not successful in the in out queued system so that was analysis switched but that was a still a single cross bar if I want to build up a very large packet switch then what is going to happen I need a very, very large size cross bar I also need a separate centralize controller which has to do very fast comparison of all the headers

scheduling them and then configuring the switch so the load on the one this particular switch controller will be pretty much high.

In circuit switching system if we can recall what we were we actually had done is we have tried creating multiple stages multiple switches smaller once to be used and a still build up a very large switch now can I do the same thing in the packet switching that is question? Yes we certainly can do that but then how the load on the controller will go down so one possibility that I may actually use a smaller switch.

(Refer Slide Time: 02:39)

I connect them in multiple stages maybe I can take up an idea of Clos network kind of thing now what I do is I do not actually use centralize controller these are likely all small cross bars all of them so there is certain input so at each input port I can actually identify I can put a board a electronic board this is going to actually analyze all the packet which is coming from the input side and was the packet has been analyze it will figure out from there that to which outgoing port a packet has to go.

Once it was to go to a certain outgoing port it should be able to insert in front of that packet something called tag okay and based on this, this tag will contain certain bits are characters are buy it is and has this packet now is being injected every switch will have at the input a interface board which will then analyze this particular tag and decide to which outgoing input has to go basically it will tell through which particular switches should be handed over this particular switch will now analyze a different part of the tag.

Will decide where it should go so basically the complete path which this packet will take through this matrix through this switch can be decide by the tag which is going to be inserted was packet reaches at the output I can remove the tag and then let the packet go through as it out once I do this I do not require centralize controller so every interface every input actually has this a small controller which inserts the tag after computing it.

And then a switch is independently controlling itself it has soon dedicated controllers which will analyze the tag set the input and based on that is starter outing the packet there is s a possibility that you may end up in getting multiple packets here which would like to go to the same port so if that possibility happens either you have to drop the packet or you have to buffer it.

Okay but what kind of network I would like to have if I am going to have network which will have more than one path for example Clos network I we did a three stage Clos and from one place to another place I was actually having multiple routes so from one input I can go wire three routes now that is where the problem which will be combine the tag switching so normally these kind of networks should not be preferred but of course they certainly can be used okay but some unit a coordination that.

To which outgoing port I should actually route the packet to because once it will reach here there made be conflict with this they are maybe conflict here they maybe conflict with equally likely probability so these can be use but these are going to be compression so that the good idea is that if I have determinist path through which a packet will follow when I will injected into the switches very important so we actually use category of switches which are known at baniyan networks for routing the packet switches.

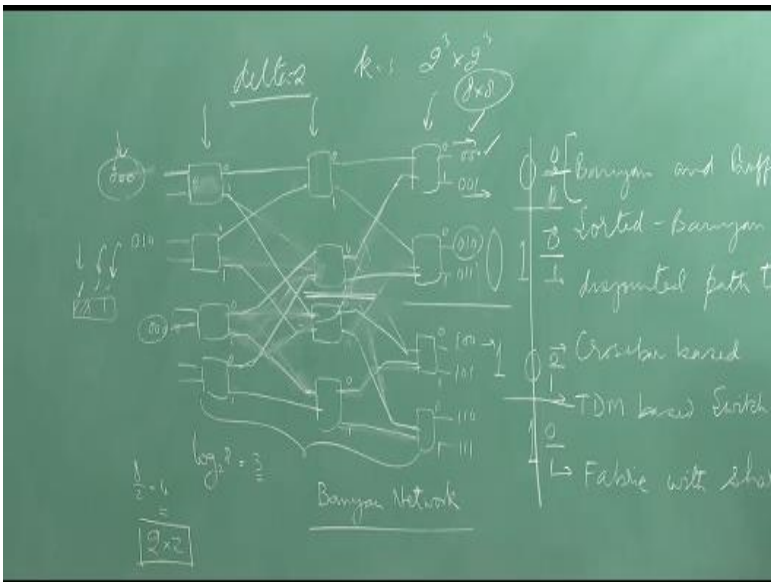
These are also multi stage packets multi say switches but the most important thing then let me define what the baniyan network is.

(Refer Slide Time: 06:34)

So we will have switches so this can be b/b kind of thing or n/ a/b or whatever you want or a 2/2 also but once I have this so I will contain multiple of them so you will be the condition is that you should be able to go take go from any input to any output so from many input I should be able to go to anyone out or any output which I want and they should be exactly path from this input to every other output.

So take any input output pair and they should be exactly one path so let us try to create one such switch so I am actually take it say 8/8 system and let see the eight outputs and one input how I am going to create this kind of switch so this is in fact gives the structure that how the baniyan network to be form.

(Refer Slide Time: 07:45)



So I am taking 2/2 so I am technically creating what we call a de multiplexer tree so from input I should be able to go to any of the 8 outputs so I am even not bother about second one so if I can go from one I can go also from the second one to the eight output so I will split and take it into two switches then they are further two inputs these inputs I will look into these thing later on and if I can further split this so I am able to get de multiplexer tree there is exactly one path from this input to every other output.

From this input to all the eight outputs there is exactly one path what about the second one if I do this of course from this again there is one exactly one path from this input what is going to happen if I have another switch I can create one here and one here again there is exactly one path I have actually essentially these de multiplexer tree of these two have these elements in common okay so I just plug it in I can similarly actually put few more.

Okay so I create another de multiplexer tree and keep these four elements as common okay so from here also I created de multiplexer tree I can use this here also de multiplexer tree so in general for this one for example it is a log2 of it number of stages which will be required number of a stages which will be required so which is three in this case three stages each stage will

contain 8/2 four switches and a switch will be of 2/2 cross bar so from every input to every output there is a unique path which is available.

I will make slide change here what I will do is okay I will have if it is so I just created a switch of this kind now I have to have I am actually now going to put the address so there eight thing so I am actually taking each will be represented by three bits so for the first one I will got get it 0, 0 I will make it 0 0 1 0 1 0 0 1 1, 1 0 0, 1 0 1, 1 1 0 and 1 1 1 now I can actually push in a packet and I can inserted tag and the way we will put it that there will be three tags so the first value of the first bit of the tag will be sue by this row.

The second will be used by this one and the third bit will be used by the third row and based on that whether it is 1 or 0 they will go up and down so I can actually make a configuration where it will be 01, 01, 01…..So if I actually insert for example 000 from here then where it will go, okay. So at every place I can put 000C and see where it is going, so if I put 000 here the packet will be looking at the first bit will be moving to this 0 coming all the way the second bit will decide It will again come here.

Third bit will decide it is coming here actually, so if I insert a packet bit 000 it is going to go to the output 000 but if I insert the same tag here then what is going to happen? This one will ne now routed to9 the word thing which will come here so first zero it will bring it here second 0 will bring it here and third 0 will bring it here. So this 00 when inserted here will actually reach here.

So I just need a unique tag if I know this port I know where I am based on these two input port and output port address I should be able to figure out a tag and insert that tag into the packet and then using self routing the packet will actually ultimately reach here, okay. So this kind of networks are what is known as Banyan networks, there is exactly one path from an input to an output, okay.

So now in this case what actually we are doing, we are taking one packet at a time in doing it, now this not the only way of doing the switching there is another way we call it fast this is

known as fast packet switching there is another method we call it fast circuit switching, so earlier we had been doing circuit switching and now we moved to packet switching we should also know what is a fast circuit switching.

So in this case idea is in the network where I actually have to set up a path is basically not one network there multiple networks in a big network there many switches and each one of them is a switching element so normally from the input side I will actually send a small header first and then after a gap I will send my chunk of bites which has to go this header will actually has its going to move through.

It will keep on setting up a virtual circuit or a circuit for a duration when this data will be passing through, okay. So this header will keep on moving every where it is going to be processed so gap between these actually will reduce, so next time the gap will be slightly less by the time it reaches here the gap will be still smaller, when it reaches in the final it can be very small value but if this gap actually gets finished up here itself.

Because the header the header processing time itself is pretty large then this burst you will not be able to route through this is also known as burst switching, what I am talking about is, a fast packet switching, the packet is coming and is being all packets are of equal length and there are being synchronously moved from one stage to another stage till the end, okay. So we are not taking about this one.

So this needs a different kind of switching design. And you have to build up a reservation structure here but optical burst switched network since they will be coming up so that also is an important thing but we will not be talking about optical burst switches in this particular part of the course, so typically these fast packet switching systems as I told is going to be Banyan network.

So there are two configurations one is a Banyan configuration as I told the two packets they are trying to go to same single output of a switch there is a contention even if the packet which is

coming from here is circuit for this one, packet coming from here is direct for this one there is not conflict for the outgoing port but then I need to go through a same link, they will get blocked.

Because of this the performance of this banyan networks is not very good, okay. It is actually limited because of the central link blocking, so we do have Banyan networks but people then have invented methods to take care of these kinds of conflicts so we also have buffered Banyan we will be looking this into the buffer banyan in lesson in the next week. Then because even if you do buffering I can remove this internal conflicts, okay.

I can actually somewhat handle it improve my throughput but it still cannot go to 100% so is it possible when the packets are coming here they do not get actually get into conflict remember even if the packets all the packets are for different ports then also they can get blocked in the system this not a strictly non blocking system, okay. While cross bar is strictly a non blocking system.

So you are actually now compromising you are going from cross bar to this having a simplified control implementing self routing but you are losing on any arbitrary permutation map which can happen from input to output that will not be possible in this case, okay. In this case only number of permutations which are allowed will be equal to number of these switches raise power 2 if these are 2/2 elements in that case.

So I will do that calculations once I go through this review, so to sort of this particular thing there is a mechanism which would which you can do is you can actually build up a again a similar kind of network where you put packets for various addresses anywhere and there is going to be an algorithm which is going to be used depending on which addresses smaller and lower the packets will; be shuffled actually here.

So they will be either if the always the smaller packet will go up bigger one will go or the larger address will go here and so on they will have a different sorting sequences, I will be able to actually sort out all these packets and they will come in order here and then if I can do an

shuffling and insert into a Banyan network there will not be any conflict all of them will pass through without any problem.

There will not be any internal conflict so far all packets are directly to a different outgoing ports, so this category of networks are known a as sorted banyan, so technically what you are doing is, you have increased the number of stages in a banyan network and made is a strictly non blocking switch so far if there is no conflicts for the outgoing port, okay. So it will just pass through as it is but it need more delay.

So that is another category then it is possible that you can actually create disjoined path topologies so multiple path which can be there and if there is a conflict we actually pass through multiple paths that also is possible at any point of time, so this is so disjoined path topology basically based switches, in this case all the packets which are even having conflicts for the output will reach simultaneously to the output port through multiple paths.

And one they reach there they have to be queued up because all of them cannot be pushed out on the outgoing link at the same time, to you require output buffering in this case, then of course we had already talk about in last time is a cross bar based systems which are the most versatile because you can create any input to any output kind of possible maps, so I will give the number of these maps which are possible in cross bar.

It is possible to build up TDM based switch fabrics with common memory, so idea is that all the packets which are going to come in from the input all of them will be stored in a common memory and they will be stored at one place and then they will be read out at a faster phase, okay. So you can read and sequence =, write and sequence and read synchronously the packets and the packets will then get switched of to the various outgoing ports.

And then of course he can actually have a shared medium is basically broadcast since select so these are the basic categories but we will be focusing only on the banyan con this particular configuration, one you have this banyan again I have given you example that if you put 000000

but they move to different outgoing ports, now is it possible that if I put I can use the output port address itself as a tag here.

So from whichever port I will put it will always issue the same outgoing port, yes that is possible and these categories of switches are known as delta networks, okay. So is normally you call it delta B use the base B system so B/B switches will be used in a delta B network and number of if the number of stages will, be k in this case number of inputs and outputs will be B^k So that will be equal to n, n/n switch can be formed.

So delta B is basically sub class of banyan, now how you are going to create it? In fact if you must have observed that I actually made some changes here whether I was drawing if I go back to the original design if I move it like this if I now insert 000 it will still reach to the same outgoing port, you put 000 in any one of the input it will always come to this one does not matter, put 001 or 010 from any input you will always end up in the same output there. Now why this happens as to be understood so issue put for example 010 as I was telling you put 010 here what is going to happen? Since the first bit 0 it goes up

Next bit is 1 it goes down it is 0 it comes here, put from here first bit is 0 it goes up second bit is 1 comes here third bit it again comes here, what exactly is happening is you can actually partition this into two halves these two partitions are being identified by 0 and 1 the first bit. I can further partition this into two halves so first half is being identified by 0 second half by 1 and then there being further partition depending on their port so it is 01010101.

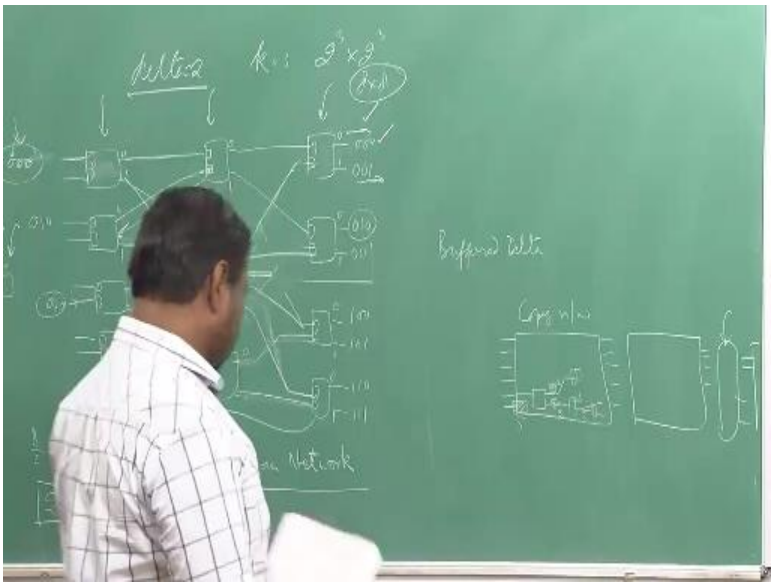
So when and actually selecting a 0 and if here I am invariable selecting where it is going to the upper half or the bottom half and now to understand why this is going to happen if I just simply swipe these two I just take this box and bring it down I take this box and bring it up it will immediately clarify what actually I am doing, so if I do that so I have to just this rewiring. So now you can clearly see when I am actually choosing 0 in both cases I am going to the upper half, if I am choosing 0 I am going to the upper half, choosing 1 going to this bottom of this where the partitioning is happening between 0 and 1.

Just simply swapping this, of course this is the same structure which I had done earlier if I just rotated around input becomes output and output becomes input, so I am able to partition into two halves and then depending on 0 I am going to either upper or bottom, the second division same here and the last one division will be happening within the switches and that is the reason it is able to do a self routing and this is delta2 actually, delta2 network of size of case with k=3, s0 $2^3/2^3$ which is 8/8 this 8/8 switch.

So before moving further, let me just talk about how we will do the buffering here, so as I told that for improving performance for this particular switch one possibility is that I start increasing the transmission rate or link speeds which is the speeder factor we have done it in the cross bar we can also do it in this place that is one possibility. Second possibility is that, because there maybe contention two packets trying to go to same port I may provide one buffer at each one of the ports, this I do for every switch.

So when I'm actually providing these buffers at for every switch this become buffer but and on this case it become a buffered delta network.

(Refer Slide Time: 27:56)



Now the problem is if two packets have come to this port they both wanted to go to the same port this one, so one of them we will pass through what we will do the other one, other one if you keep it here and the another packet being transmit in the next cycle when this packet cannot pass through or this buffer is not empty then what you will do, one packet either it has to be dropped, your technically assuming that they will not be any flow from the back, so we use something called a back pressure mechanism in this case.

So normally if this buffer is full, it will tell to the switch in the back that you cannot transmit to me because my buffer is occupied, so if this is occupied so any packet which is just in for this will not be transmitted for further, so it will be doing backward propagation so backward basically whenever a packet moves out the signal will flow backward and the packet will move in the forward direction, so no packet technically will get lost inside the switch they can only get lost outside, so there will be input buffers through which the packet will keep on moving that is how the buffered delta actually operates.

There is another way we can actually do it, these kind of conflicts if these are because of internal links if I can distribute the packets in properly see in a random order so most of them if these two are directing towards same port if I would have put this packet somewhere here, it would have gone through the other link, so internal conflicts can be avoided so that is done through a distribution network.

So randomizing and distributing so that is what is done through sorting and then distribution network so that is again a kind of Banyan network itself which does the distribution, randomization of the packet so that internal link conflicts will not be happening. So far we have been assuming unique cast, but in any internet switch or a packet or a packet router I should provide a multicast capability so one packet goes in and multiple packets from multiple outgoing ports will be coming out how that will be done.

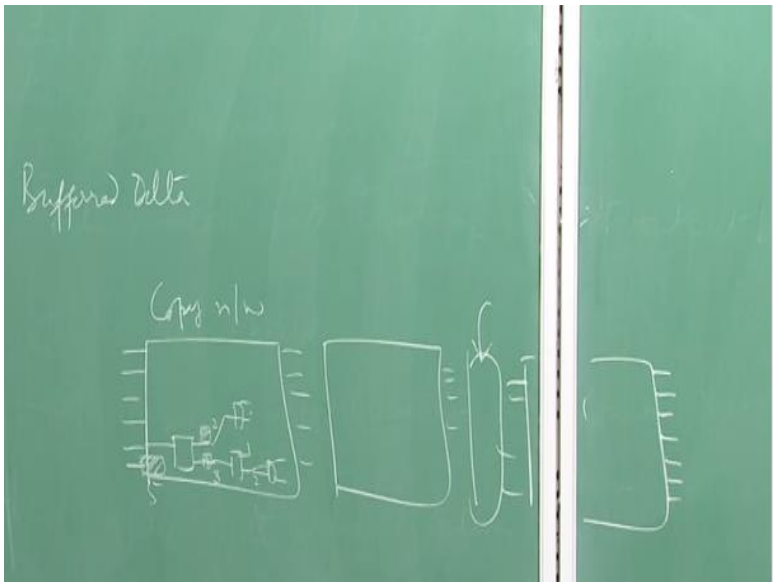
So normally for that you actually have a small what we call a copy network which is put the packets are being put and then each packet how many copies have to make as to be inserted in the tag. As the packet goes through a small switch it main take need if it makes two copies so it

can actually put the half of whatever is the value in the tag here and remaining half here, so as it goes along more copies will keep on happening ultimately this value will reach to 1.

If it is an odd number so say if it is 5 so you may put 3 here and 2 here so there will be only two copies next time which will be make here, so when this is the copies are going to made here so 1 and 2 so next time there will be two copies being made by the next switch, so you will have 1,2,3,4,5, copies which will be create. So a copy network distributed, distribution network after this, okay and the of course you will do a shuffle interconnection and then putting a routing network which is basically this delta network will give you almost, this will give you non-blocking system with multicast capabilities, multicast as well as broadcast.

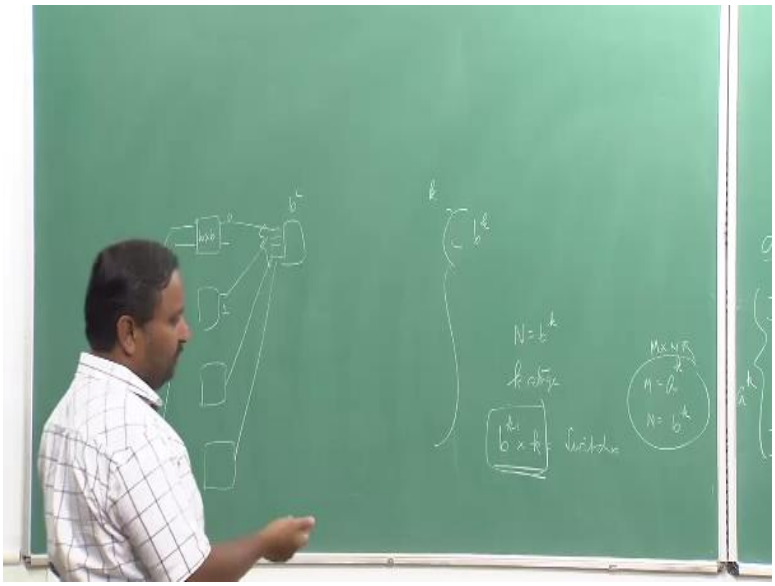
If you do not need this using these two only you will be able to do, this becomes a sorting network technically, okay. so sorting plus this becomes a distribution network and to give you almost non-blocking performance for the packet switches.

(Refer Slide Time: 31:56)



So let us move to the enlists of delta network now, so how many networks, how switches will be required that is one important criteria in, so if you are going to build up.

(Refer Slide Time: 32:13)



Now the trick for building delta network is very simple that whenever you build a take a switch all inputs which are coming to this switch should come from the same level of the previous switch, so if it is coming from 0 here and this being connected here all these should also come from 0 only from which ever switch that does not matter, so far you do it you will be able to create a delta network. But you take from the 0^{th} output this particular input, but from one output you take this one then it will no more remain delta, okay so that is the only characteristic for a delta network actually.

So we can try out and of course if it is a b/b network in that case you will require b outputs and this will become b^2 next time and it will be able to reach and the k^{th} stage it will outputs will be b^k, okay and if this is also b^k so for a switch which is $N=b^k$ there will be k stage is, okay each stage will require b^{k-1} basically n/b those main number of switches so b^{k-1}x k so this will be total number of switches which will be required for a Banyan network or a delta network, delta network only this routing thing we have to take care of, okay.

And if you have a x b switch then what is the condition, I can also build with a x b so first stage will contain a second one will be b I can have this basically becomes M x N where m is equal to

some number so if it is a case test system so next output will be containing so these, okay. so after the k^{th} stage let me just see how many of this, so my outputs will be b^k will be equal to m, so number of switches will be 1, b^{k-1}, okay so this will be 1, this will be a^{k-1} total number of elements inputs will be a^k so m will be a^k n will be b^k

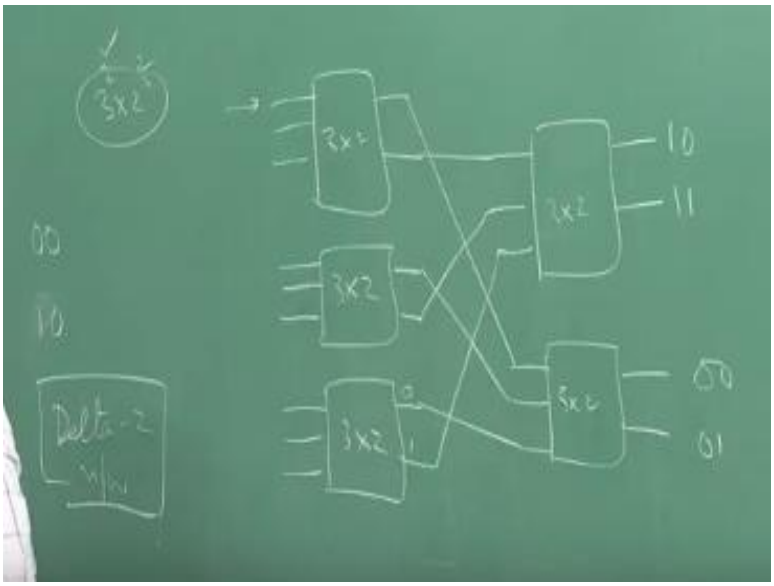
So the number of output ports here will be now I can write it here a^{k-1}x b so this should be equal to this input ports in a Banyan network or delta network which is a plus that of that, none of these input ports will be left free, then none of these output ports will be left free and no input will be terminating on these internal input ports and output ports they will be only inputs will be coming here output only going out from here and since these are not free this same number of ports has to be there on this side okay so I will take each one of these and connect keep on doing it for every switch every time okay.

And then on this side number of switches now will be a^{k-1}/b divided by so this one this will be a^{k-2b} so these when the switches will be there so first stage a/b switches number of them will be a^{k-1} second stage a/b switches will be a^{k2xb}. Third stage a/b and so on so by the time we reach the last stage which is the k^{th} one here you will be having b^{k-1} switches so total number of switches will be nothing but some of these.

So you can actually get sum from here itself so this is what we will get when a is not equal to b this is what is the total number of modules when a will be equal to b the number of modules will be these one okay. So and of course we can actually see take few examples and try it out whether self outing happens or does not happen only thing this rule that all the inputs to a switch in stage should actually come from the same level same output port level of the previous switches will be taken care of and you will be able to do the self outing.

So the output ports be not in the order but those all also always can be made a alter occur shuffling the output wires, so we can try out for example 3/2.

(Refer Slide Time: 39:36)



$3^2/2^2$ kind of configuration so this actually means I will have 3/2 two of so I need actually in this case 2 stages so I will be requiring 3^2 so which is 9 and second stage I will be requiring 23/ 2 switches as per the construction give me of the output ports there will be only two stage k=2 3 inputs all these three should come either from 0 or they should come all from 1 level so I can take all of them from 1 I can take all of them from 0 so should be a self routing switch.

So you put from anyway anywhere now the kind of address which you are tag which you will be putting will be vote it should be either 0 or 1 first bit will be 0 1 second will be 0 so it is a basically two digits both of them will be binary numbers, so if it is a a/b switch I will be using b area digit k number of them to identify the address of the outgoing port okay so in this case that will be a two binary switch which will be there.

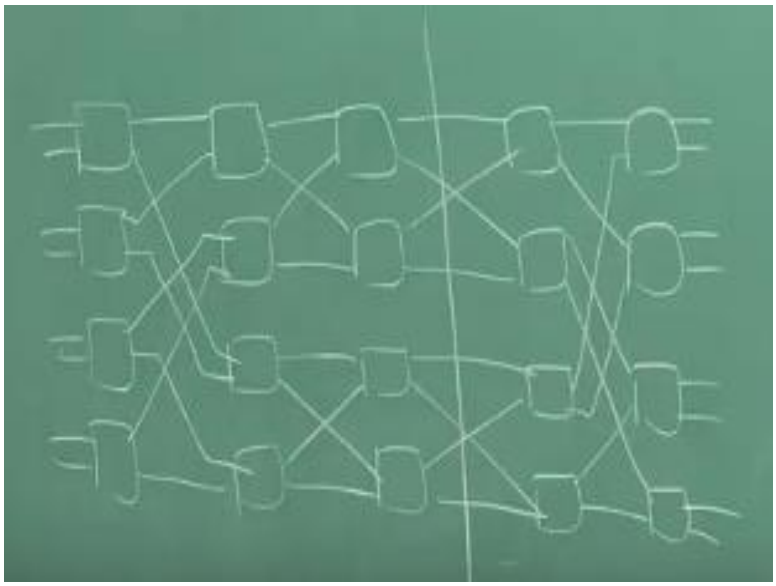
So for example you want to put 00 you put from anywhere if you put 00 here the first 0 will cause the packet to come here second 0 will come here so this is a 00 output port. Take it from anywhere so if you take it from anywhere 00 all of them will be coming here and ultimately on to this particular output line, put 10 in that case take it from any one of the inputs one will cause in to the bottom and they will all come here and then next one 0 will come and give a 0 here.

So this will be 00011011and it is a δ routing network it is a δ2 but these an astigmatic system this is the astigmatic network but it is a δ network. So we can actually have many kind of combinations inter connections which can be done but now the problem is I need something for a symmetric system so this inter connection can be replicated very easily when you do fabrication when they case I just it is.

If I do I am not able to repeat this particular system again and again and again every time I have to make a different int3er connection pattern and this going to be while fabricating will become a difficult stuff so the question is if I actually can make a switch of a/b, a/b, a/b switches the inter connection pattern from first to second stage second to third stage third to fourth stage should be exactly same.

And if I can do that and that I should be able to build up a still a self outing network. So thus that kind of switch is possible or not possible okay so far what we have done is technically we call it base line network remember base line actually has come base line or inverse base line both are self outing networks they actually have come from base network.

(Refer Slide Time: 43:34)



So remember for a 4/4 the base network which we have done earlier this has come from regional none blocking thing so this will be requiring five stages so there about two more stages the full base if I draw so remember this was something like this so these the base line this is the base network from the base network if I take only this particular part this become the outgoing port remaining I forget these the base line network.

And you can actually note this is also self outing system you take this port I am either going up or down and then further from there only this, this is the self routing system that is what actually I tried in the beginning if I take not this one but the other one other part is taking this onward on the way here that is also self routing network if you can actually carefully observe okay so if you put in.

So both of them are self routing but remember the inter connection pattern is not same they are different in same stage this one is for can built through de multiplex try base line one the inverse base line can be converted to this form by actually swapping the notes here and there moving them around now the question is what can I do it with uniform pattern between consecutive stages yes we can do that.

So there is one particular network we call it shuffle so shuffle inter can shuffler network actually does this so but for this we need to understand what is the shuffle inter connection.

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