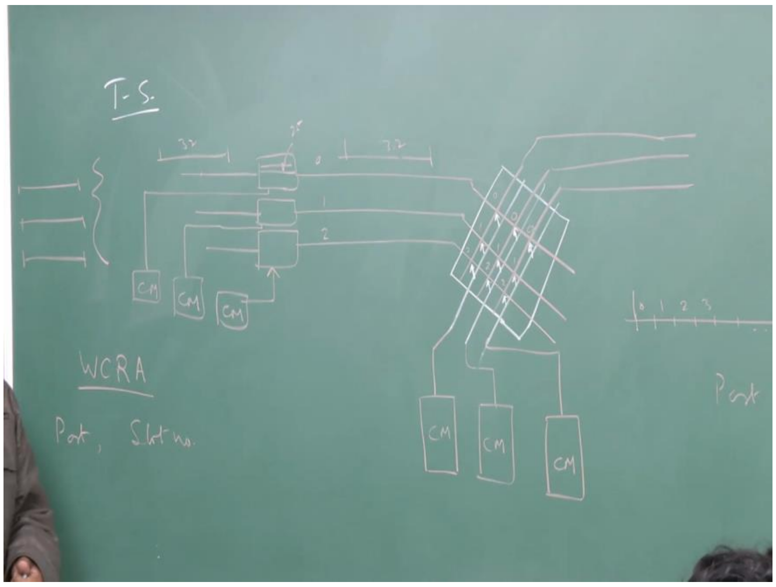
Digital Switching

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

Lecture - 14

We will now move ahead with whatever we are doing. We are looking at again time switches. Earlier we have done super multiplex time switch.

(Refer Slide Time: 00:27)



Now, we will actually look into time space first and then timing, space time later on and then how the controlling structure of these kind of switches going to be implemented. So, let us take a time space switch and obviously I require a space switch. So, I am taking a 3 by 3. So, these are the three outgoing ports. The space part I have done so far, these white lines are nothing but the control lines. They will control the cross points.

So, if the cross point is on or not on, it will be governed by these and interestingly, I do not require a row and control a row, and column control will not be required the way it was there in the cross bar in this case. I will tell how actually this happens? I do not require a row and this thing and these are the input lines, but I also require a time switch, so far which I want a cross connection, it is ok, 3 by 3. So, when I mean time and space, this actually means time is already slotted and each y slot, there will be a separate direct connection which should be operating for this switch. So, for even frame, every 125 micro second at 32 out of these two does not. You would never bother about them actually.

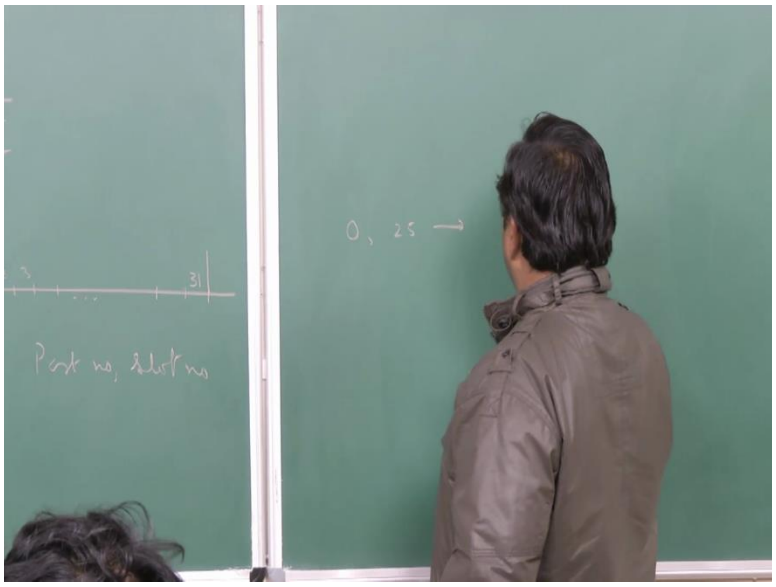
So, basically we thought that at 30 slots, for each of those 30 slots in which we want the switching to happen, there will be specific cross connection pattern which have to be have there will be 30 patterns. So, they have to change every time slot. So, what we will do is, we will actually now address these as 0, 1 and 2. That is it will be done and I will use this. There will be control memory and I will also have a time switch. So, this I am showing is speech memory. Each one of them will be controlled through a control memory. So, I am not drawing all the circuits which are there as per the time switch design which we have done earlier. So, it is the same thing.

So, this is serial line. This will be converted to parallel time switching will happen. It will be converted to a serial line again. So, it is not a parallel bus. It is a serial line that is what I am assuming. That is why there three ports here on timing scale; this will be 1 to 125 micro seconds. There will be first slot, second slot, third slot and so on like this and these are frame boundaries. So, you will have 8 bits coming in the first slot and then you will have again 8 bits coming corresponding to second wire circuit and so on. So, there will be going from 1 1 2 3 and 3 1 2 slots. I mean we really will not be bothered about whatever switching you do for those does not matter because anyway it will be is stiffed of at the input, and they will be inserted back at the output inside the switch.

Now, this kind of configuration because what we have done was the write cyclic and read a cyclic. So, write cyclic and read a cyclic, this kind of configuration was discussed by me in the earlier lectures, one of them. So, this actually means now you want to implement a switching. There will be corresponding three frames coming here. So, what you have to specify always for a switch map input port number which is 1 1 and 2, and slot number and you will be having an outgoing port number and slot number. The map has to be generated for this.

Now, let us look first of all the switch will operate in the first time slot because I am doing right cyclic kind of configuration, and this need not be enable more number of ports, the more number of slots for frame and keeping it same in this case. So, what will happen is you will have 30 frames, 30 slots. Here you will still have 30 slots, here 32 basically. So, you need to create, first of all you will be given these things. These things are given by your switching requirement. These are not given by the switch. Now, you have to just generate map for this.

(Refer Slide Time: 06:43)



So, once I give you this something. So, for example, you say that for 1 number port slot number of 25, this should be mapped on to port number 1 slot number 2, this kind of scenario. So, how this will be done? So, one thing is sure that when I am doing this, only this time switch can change the slot here. Actually only the physical port number can be changed. You cannot change the time slot because this is a space switch. This is a two stage switch so far. So, in this case, what you require is your time slot should actually map from Zeroth port from 25th port to second slot, and remembering the speech memory I will be writing in which location I will be writing in 25 fifth location because that is a right cyclic. While I am writing, I am writing in the location corresponding to the time start number that counter which will be there which is counting time slot.

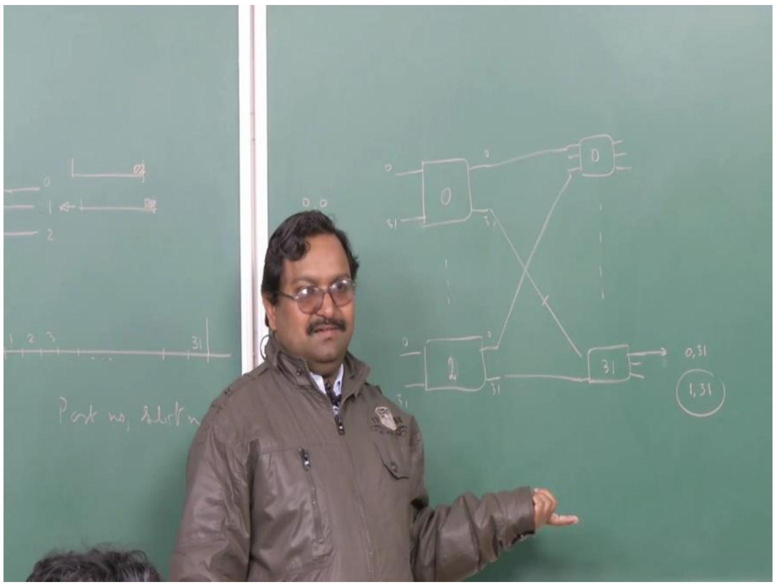
So, I will be writing here somewhere in 25th location, but while reading I have to read in this second slot 1 1 2. In the second slot, I have to do the reading. So, who will give the reading? Reading will be given by the control memory. Now, it actually means in the second location of the control memory I should write what value here. I should be written 250. So, in the second slot read from the 25th location, we call this as cyclic read. So, in the second slot if I am reading from the 25th location, this location was filled in the 25th slot of the input. So, I have already done the time switching. I have moved thing from 25th slot to second slot. Once it is done, I need to know output thing to 1 1 and 2.

So, in the first slot I have to put it. Now, remember in this column, only one of these can be active, even two can be active. This not an issue, but I am assuming there is no multi-cost. Now, there can be a multi-cost, but all these three never connect it at the same time because three inputs cannot go to the same output at any point of time. This like mixing the signals they will interfere with each other and you cannot count anything. One input, I can go to multiple outputs. So, for example, if I have what have done this kind of scenario, this control is implemented in such a way because I know that single input can go to multiple outputs, but multiple inputs cannot go to single output. That is not possible. So, I have to select the inputs, not the outputs actually. So, for each column, each output I am selecting corresponding input. Only one input has to be selected.

So, this is 1 1 2. So, now, I am reading in say I am worried about the second slot. So, in the second location because whenever the same counter is counting 1 1 2 in the second slot, I will be reading out from this control memory. From the second location, this should actually have a value 1, once this have the value 0. So, during what value it should have 0? Yeah, 1 this 1 cross point will be connected for the second slot duration, and the second slot is what was your octate was coming here. So, this octate will be rooted to this output. If you want to implement multi-costs, this actually should go to second slot. So, all outgoing ports, then all second locations in all three control memory will contain 0. So, you require the number of beds which is required for memory what will depend on how many cross points are there for each outgoing port or in each column. Column corresponds to n outgoing port row corresponds to an incoming port in this cross bar.

Now, this is the time multiplex which remembers this is one kind of 3 by 3 which one slot takes one configuration. First slot, one slot is no more existing time which will be exclusive. When slot one is happening, any other slot cannot happen. So, this is like 30 different switches which is operating in time multiplexed mode of 32. Again two configurations are useless because you will not be switching them. So, this is like 32 times space switch is, but I am using one single switch to immolate those two 32 space switches I could have done at d m here and then could have created a space output, 30 space outputs. Then, I would have required 32 space switches and of course, this obviously is a blocking switch. You always will have to do configuration where you cannot connect even when input and output is free. You can think of some configurations where it will be blocking. It tends to be, it is a two stage network. It is not three stages. It was a particular time slot. Suppose more than one channel want to send on the same output channel. No, that is an output contention that is certainly only one will be permitted, but there has to be case where input and output are available, but you cannot make the connection.

(Refer Slide Time: 13:13)



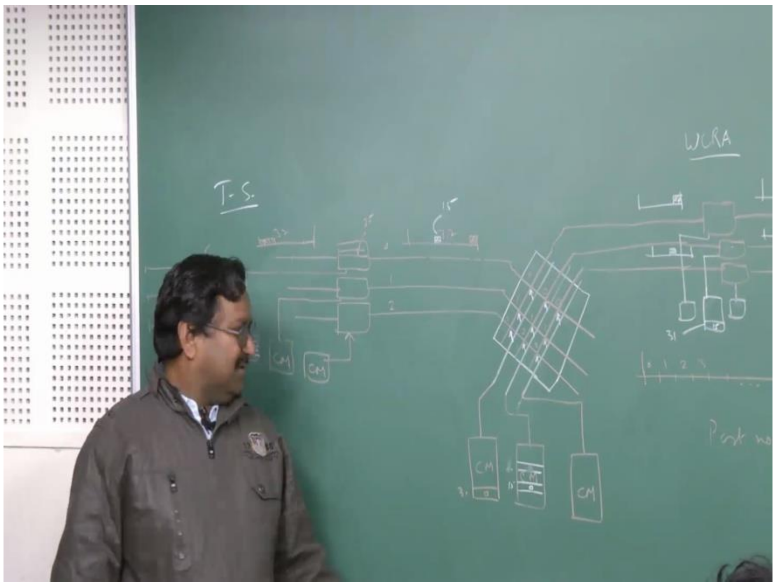
I think it would become clear if you can build it. If I build up a space switch because that is, so this is the 30 was able to 31. If I switch three of them and then you have 3 by 3 this equivalent to this. So, I just converted my time switch to this space switch here. So, then required 1 to 31 and I think I should put 1 to 2. That is a best way. So, both are equivalent actually except now I am not using a time switch. Now, I can find out one corresponding blocking configuration here is very simple. Let us one be connected to this particular port. This actually means Zeroth port time slot number 1 getting connected to. Now, this corresponds to the physical port and switch number corresponds to the time switch slot, time slot.

So, this corresponds to 1 31 slot. If this mapping is done, this particular connection is gone. Now, you want to connect slot number 1 to 1 31. Now, is it possible? It is not possible because we have to first of all put that thing into 31st slot. That 31st slot is already occupied. So, now, look at the same thing here. Now, one thing shows that I cannot set up this link. This gets blocked. So, coming here what I am saying is, this is the last slot, this 1 0. Actually 1 0 is this one. This one was mapped to 31 and then during the 31st slot, I have actually connected this as 1 and 31st location. This gets rooted on this output. You will get this.

What I want is now on this, on the first one, on the second output, I want to connect root from this second slot. This can be only done at the second can be moved to 31. 31 is not available that is like this line is. There is only one connection which is possible here, and it is a blocking system.

Now, immediately probably should be able to cognate between the two scenarios when you want to. For example, this is equivalent scenario that 1 1 has to be connected to 1 31. If you want to do that, this one has to be moved to slot number 31 here, but this slot number 31 on this line has been already occupied. Slot number 31 is already occupied because of previous connection. This makes this line has been occupied one line connecting this switch to this switch. This one, this one only output conjunction. This will be switch blocking input and output. Both are available. Input is free, output is free, but you cannot setup a root. Exactly same principles will follow in this configuration also, in timing space configuration also. So, with two stages, it is not possible. So, what we do is, we go for the three stage configurations because I can use whatever logic which we have learnt earlier.

(Refer Slide Time: 18:14)



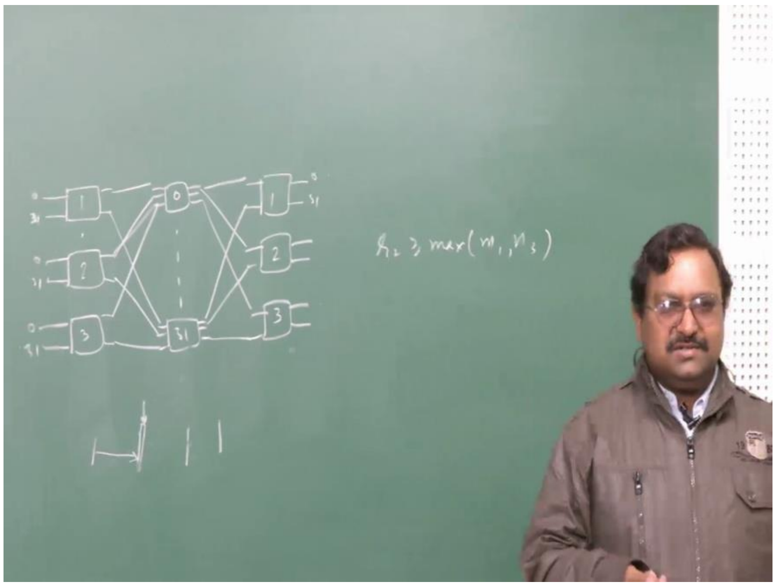
So, let me put a switch here, let me put a switch here, let me put a switch here and there will be correspondingly the control memories, fine and once you have this, I will be able to take care of that contention. How that will be done? So, I have done 1 to 31 switching over same thing. So, from here this 31 remains 31 when I want to switch to 1 31. So, from there I can take to any slot which is available in this case. So, this might be this one say 15 which is available. So, I map from 1 to 15 to this time switch.

So, in this case at 15th location, I will write 1 at 15th location. So, during the 15 time slots, it would be to beat from location one which is written during the slot 1. So, I needed to do time slot change actually to this t s i is known as time slot interchanger also. So, when this comes, so at that point I can also change. So, I can output say at 15th location, I have to now connect switch number 1 that 31 1 remains here, 15 1 which has to be put. So, 15 slots both will be coming here and now, I can swap over using this control memory. So, in location number again if it is write cyclic and read cyclic, same kind of switch what will happen is, it will be writing on the 15th location inside the speech memory, and this has to be written over the 31st.

So, here 31st location, I will read from 15 th. My only problem is that I cannot use one single clog, which is derived at the input of this whole block. This is one thing is known as switch block. This whole unit is known as switch block actually, and each of these elements is a switch basic unit. So, there is a complication because here it is 15, here it is

15, here it is 31. This is also 15. That will keep the life very simple actually. So, whatever is current slot running, I can just use that. Of course, there will be a problem. If I expand the number of slots here, then this will not be true in that case because the number of locations here will be larger. Currently this is rearrangeably non-blocking switch. So, I will change this.

(Refer Slide Time: 22:06)



So, this is equivalent to 1 to 31. Another one, 1 to 31, another one 1 to 31, here 31 outputs. So, this 1 2 3 1 2 and 3. Both switches are equivalent. Now, this is a space configuration and as per the theorem, this satisfies the condition of free non-blocking property. Here, I have 2 is greater than or equal to maximum of m 1 and m 3. So, 31 32 are there and r 2 is equal to 32. So, it is rearrangeably non-blocking configuration.

The time space is both are equivalent. All three satisfies with this. I am saying this configuration and this configuration are equivalent. You have these two 32 cell space which is these 32 are be immolated by one single 3 by 3 here because no time is multiplexed. So, everyone 25 micro second, it has 32 possible configurations immolated these switches. So, important thing is the amount of hardware reduces drastically. Each of them is now converted to time that is only thing, but this is a time multiplexed thing. So, instead of 32, you require only 1.

So, it is a big advantage actually. So, big advantage here now this column still has to be resolved, otherwise switch block control cannot be simplified because remember when I am going to build up a packet, switch packets will keep on coming at the input and packets come for a very nerved duration. So, depending on whatever are the input packets which are there, I have to immediately figure out for that slot. I interconnection input to output mapping which has to be implemented for that slot. All packets will be rooted out to the outgoing ports and by that time you to compute for the next incoming slot what should be a new configuration.

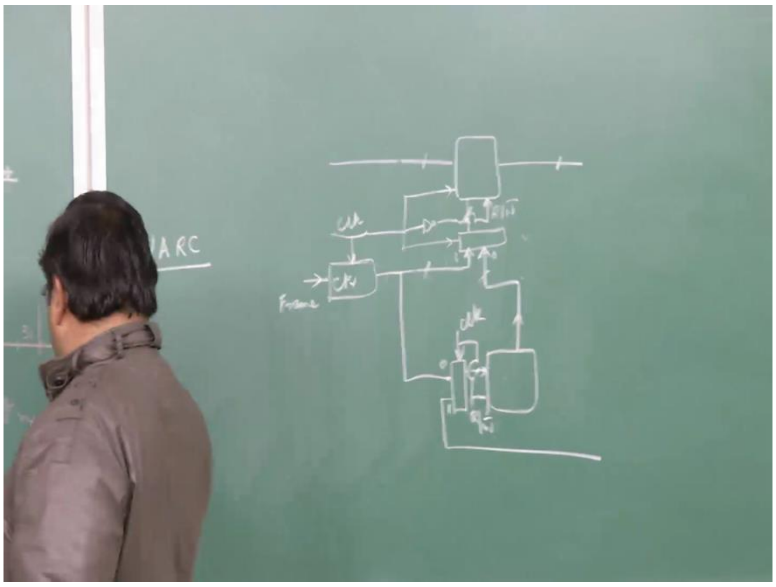
So, it will be very fast switching which will be done. It cannot be like circuit switch which you set up a path and inter (( )) and there is also one very important phenomenon which will happen is, in case of circuit switching, there is already some connections are setup. You setup some new connection without disturbing the older ones. You release some connections. Some connections will still remain. Now, that situation is not there in packet switching system because at the beginning of the slot, it is like fresh. All connections are made fresh depending on how the packets have to be rooted to the output and of course, my assumption is that there will not be in that stage. There will not be any line problem. The only problem which can happen is the output contention. When two packets try to go to the same output and same slot simultaneously that will not be permitted, but now this kind of switch will operate at much higher rates, and must have efficiency actually.

So, every time slot, it will change every packet slot if it is a t m system. So, whatever is the 40 53 byte, a t m cell for every a t m cell, there will be different switch configuration. Usually switches will be synchronous. All input packets will be a line. They will be usually of same size. Output will be also of in same way line, but in real life packets are actually not of same size. No i p packet are of same size, but that reduces efficiency of switching. If I start using variable length packets, then it becomes usually which operate of make it everything equal and synchronously, I can start operating the switch. The only problem is all packets have to be analyzed. They have to be aligned and all action has to be done. All control memories have to be configured and then the packet will go out and before the next packet is pushed and then again you have to reset, change all the control memories correspondingly have to be later. The only problem is for the old slot.

All these thirty locations have to be rewritten, all these thirty locations. So, before a frame starts, then that is a complication because this whole memory. So, in this case, they have actual total nine memories. Each one consisting of 32 locations. All these word has to be returned before the next batch of packets can be inserted, all the packets. Now, this is the only complication which is there. You can still use the switch, but it is going to be complicated because you have got there is not that 125 micro seconds thing is not there anymore. Usually it will be shorter duration when packets come into picture. So, what will happen is this is the packet slot duration. So, switching has happened in this very small fraction of time. All the control memories need to be rewritten parallely. So, you have to write all 32 words inside the memory within the short period.

So, this switching can happen. There will be some period where you cannot do anything; you cannot pass through any packet. There will be a small gap or guard band which has to be there. Again this is to be done in very short thing. Now, this is one of the crucial things to implement circuits switching. It is fine when you want to setup a circuit. Only one memory location you have to write or something is fine, and then circuit remains for a long period. So, packet switching system that is the reason why packet switching systems are not usually implemented in this fashion, unless you can write this control memory in a large burst. The whole memory has to be written and not just a single word. Actually now coming to solving this particular problem for circuit switching, still it becomes important. So, how you will do that? I have given a clue actually yesterday.

(Refer Slide Time: 29:33)



I asked you to write draw a schematic for write cyclic and read cyclic configuration. So, what was the duty in that? In that case, you had a speech memory. You are writing into it and you are also having your counter which is generating the address, but remember now you have to read cyclically. This address has to be for reading purpose; not for writing. For writing, you will be taking the address from the control memory. So, control memory whatever is written actually tells where it has to be, in what location it has to be written actually. So, I think it was pretty simple. There was a clock here which were also driving the counter and there was the frame clock which was driving, which was resetting the counter every frame period and once this thing comes, this clock during positive cycle, I can write ending with negative cycle. I can read and I have to select a rate. So, output of the control memory goes into this. This counter address actually goes into this. This actually means the outgoing thing. This clock will also be driving. This particular ram, it has to if the numbers of frames are same, otherwise you have to go for dual system, dual memory.

So, the thumb rule is that whenever this clock rates input and output, both are same. You can actually go for single memory option. Whenever the clock rates are different, we have to always go for dual memory option. You cannot work with the single memory in that case. So, this clock will be selecting. So, when it is 1, it has to be for writing. It has to take from this address. So, one will be here, one will be here in the design which I have taught earlier which was write cyclic and read cyclic. This was reverse. Actually I have just inverted ones and zeros circuit actually remains the same only. One had to be swapped. So, that was also done said to the quiz which we read yesterday is exactly said circuit has to be copied zeros. Ones have to be swapped, nothing else. So, this read rows goes here. Clock goes in this case.

So, whenever this clock is positive, we are going to use this and read write bar. So, you are in the write. Only thing is with which address you are going to write. It is going to change. That is only thing that is. Everything else remains the same and same counter also goes. I can write, but actually I will not be using even because once I make this change there, actually my block control, switch block control I will not be using a separate address coming from the control. That will not be done, but I am still drawing it now during this has to be 1 again. This is controlled from same clock. So, when it is 1, this address is being from the earth, whatever it is coming out, it is 1. It goes into this. What it means is doing 1, you will be writing here and again correspondingly there will be a read write bar. So, that is how it is going to work. It is simplest that you have to again change the earth you have to see. So, earlier it was 1, it was 1, it was 1 and it was 1.

Sir, instead of that read bar, instead of 1 and 1 if we start for read bar write control. That is also fine, but circuits are always rewrite bar 80 85 circuit. You might have done. It was always read write bar. It was never read bar or read bar w. So, I just use the convention. We would have all solutions are correct. So, far it is going to function on the chip or hardware system is fine. So, let should function without any column and other people who are using everything as a module should actually get the same module obstruction independent of how you implemented. It should never actually implement something in such a way that is different than the obstruction because when they are building up their system, some module sub system they always conceive. They always think of an obstruction and if it is different, their system will become wrong because they always consider know your box is a black box to set in designated area. So, that black box whatever is visible to outside should remain same.

Internally is your problem system should work consistently as per the obstruction module. It has a lot of obstruction. You must be observing which I have been also using here. Now, using this, what will be address here I have to use now because in 31st any way will be reading in 15 slots. I will write 31st location. So, once I do this, I got 15, I got 15, I got 15 here also. All three. So, you use write cyclic, read cyclic here instead of w c r a. So, I go the same address is actually. So, now, coming to the switch block control, how that will be done? Once this is clear, I can move to switch block control.

Now, we are converting all the time switching is w. No, the first stage is write a cyclic read a cyclic, for all three of them. The third stage, it is write a cyclic and read cyclic. The things which are not getting algorithm or these are example which I am showing these two for any combination. If you take any mapping from here to here, you will find out, you will require same location everywhere. You take any mapping. Now, you can take anywhere. For example, one say 128 have to map to say 3 or 40. Let us take this out. So, I could do 128. So, 128 is going to be in this thing somewhere here.

Now, I only know the input port and slot number, and output port and slot number which particular slot number will be used in this and this. For transporting I have to decide whatever is available and if it is not available, there is no common slot available which can be used. I can make the rearrangements and always find one again using that theorem and then the matter by which we do the rearrangements, we can find that because remember this switch is equivalent to that rearrangement of blocking a space switch. Anytime you have a confusion, write that, draw that thing because theorems have been done and then from there you find out the rearrangements. Do the slot change, whatever it is. Find out the vacant slot. We can always find a vacant slot which is common to this as well as this third one. Basically you need a common slot because some slots will be occupied here which have a slot which is free in both of these, and which is same number only that you need. So far you can find. One you can use that.

Sir, initially when the third stage was also w c or write cyclic, then also it was rearrangeably not known. Then also we can. Yes, then also the same procedure. When the advantage are proved because you get changing into. No, with this memory addresses, all three will have now 15, otherwise you will be using 15, 15 and 31 here. This will be the implementing control. It is a control implementation issue. Now, switching will be used to function. Switching has nothing, no issues mathematically. Everything is correct, ok.

Sir, what we are getting two stages we are converting from time slot, you know time because I am trying to reduce the hardware. So, why we have to completely remove this. Which one? Second stage. Super multiplex time switch was that see instead of that 3 1, even I was using 16 evens on this side. I can write a super multiplex time switch structure and I got 16 evens out. So, I am not here doing super multiplexing. So, super multiplexing also has a limitation. You cannot keep on doing it ultimately forever in large size. Sometimes you will go for a space. Hardware requirement is more in the time switch. It is not displaced always. It is only this. Only things are clocks we can switching.

So, what advantage and disadvantage in t s t? Compared to whom? Time switch and one time switch. This is just another option. If given I can implement everything using a super multiplex time switch. I will always go for that, but the clock issue synchronization issue if you can maintain that, that is the best you are using in something which is in time. Only thing cost is arrived actually there even storing, reading and writing in different sequence. See this cost actually becomes huge if you start using from number optical which is optical, which if I want to do all optical implementation.

(Refer Slide Time: 41:00)

I require a delay line, a fiber delay line for memory. It is pretty simple for you. It is, this is a flip flop say electronic flip flops fiber. There is no flip flop. You can actually build it. It is not you can build optical flip flops also, but these will be there. Actually switching speed is an issue and then of course for how long you can store. Usually this delay lines is the most common technique which is used, but this made the switch cumbersome. If you want to build up time slot interchanger, you can do optically. It is not that you cannot do it, but then it will actually have problems. So, building up a space in optical domain is going to be much better because it is more compact in size. So, that is the reason instead there is nothing like a best solution. I cannot give you the best solution which will fit everywhere as an engineer. You have to look at all of possible options depending on the application for which you are implementing. You have to choose the best thing out of all possibilities. So, here I was doing this 128. So, maybe you find out say 11 is free slot number 11.

My only question is how do you find out that eleventh slot number? No, that is again theorem. See when you are setting up connections here, how many inputs are there? 32, but there is one slot which is free, which we are trying to connect one input. So, only 31 can be occupied here. So, there is at least one slot which will be free here. Look at this output again. Your one output is free. Let say rest everything is busy. So, it has to be one slot which is free. Here also in first case scenario, the slot which is free here and slot which is free here are not same, this same element. So, the row which is present actually element in the column, you cannot find out a common element which is missing from both. If you take both the sets, all are occupied. That is what is happening.

All slots we are taken here, all slots taken here make millions of those slots are occupied. Then, you can always find a pair, the element in the row which is not present here, but present in the column. The element in the column which is not present in the column, but present in the row. There is already pair I have proved that and then I can use that same algorithm row column thing and then find out and do all those swaps. Then, you can always build up a slot which is free. In both, use that later for them in the back and you keep on running. Algorithm is going to be same. What was there for space switches? It will not change as for the software obstruction is concerned. Obstruction remains the same. Your implemented switch is different. It is almost depending on data or algorithm.

No, there is rearrangement algorithm which comes because of in the (()) theorem. That will read where a controller will be writing separately. You know there is separate command control. There is a separate processor which has to figure out. Somebody has to tell when you are skipping the signaling that is going to come to that computer. There is a micro processor setting which is running the algorithm. So, once signaling information come, he will come here. Signaling information will tell that which particular part has to be set up to connect with particular port pair. You put map and will be undertaken by that. It always happens in telephone when I would dial a number. It goes to exchange and then exchange looks into table. It has to go to particular line. Any port on this on the output side, it has to go to only specific outgoing port in any incoming port. It may take depends whether you are on transmit or both users to connected to same exchange. If both users are on the same exchange, the ports are fixed which has to be connected and then they have to run what you call outing algorithm and switching algorithm is required after that.

So, I am assuming is already running in some other processor and that processor through a bus mechanism is able to write into this. So, I am coming to that actually. Now, how this will be done? There will be three kinds of matches which have implemented which are actually used for this kind of communication. Here one doubt is there. What is algorithm depends on? It only says that h2 has to be greater than or equal to maximum of this. Theorem is there. Then, after there is a rearrangement algorithm which I told. Looping is only for 2 by 2 elements. Looping cannot be said in general. In general, we cannot form loops actually. So, you have to find out c d and then corresponding c d. This algorithm I am talking about. All is depended. Yeah. All.

Again that is the Paul's theorem. Paul's theorem only says that how many rearrangements are required. That is again a theorem, but the rearrangement procedure is part of this (( )). So, it not only says that what minimum is required, but how it also will be done. For example, find out a start number say 10 which is available here as well as here. Once you know this, I am going to write cyclically. So, I will be writing at twentieth location, but while reading, it has to read at the tenth location. The value will be 28 tenth locations. Here switching has to happen. Sorry, this one tenth location switching has to happen at this point one and at tenth location. I should write on to location number 14. In this case, this is the tenth location. I have through directed location number 14. Time will be reading out cyclically. So, it will be ultimately coming at the fourteenth location here. So, this will get mapped on to fourteenth location. This one. There all locations are 10 10 10.

Suppose this side then only 2 have to be replaced for same. 7 is there. Instead of this time you do 7, but then this also has to switch at the 7 only. So, here also the 7 will come and also, then you have to move at the 7, because you cannot do times switching here. So, there also it will be at the 7 th location. So, everywhere it will change. So, the location at the control memory location is identified by the slot which is being used as the intermediate to transport the y sample. I think this already 10 955, I have to close. So, I think switch block control is what I will do in the next class, and then we move with the packet switching system.