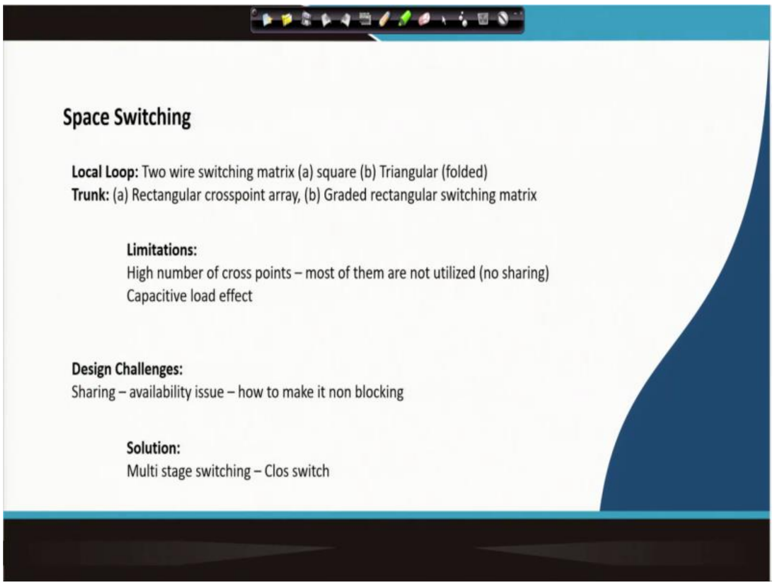
Communication Networks
Prof. Goutam Das
G. S. Sanyal School of Telecommunication
Indian Institute of Technology, Kharagpur

Module - 02
Circuit Switched Networks
Lecture - 09
Space Switching Architecture cont'd

So, this is Lecture number 10. We have already discussed some details of Space Switching. So, we will take it forward, and we will also explore more time switching, which is another important dimension of switching; we will see that ok, but before that probably, we will talk a very little bit about multiplexing.

(Refer to Slide Time: 00:46)



Space Switching

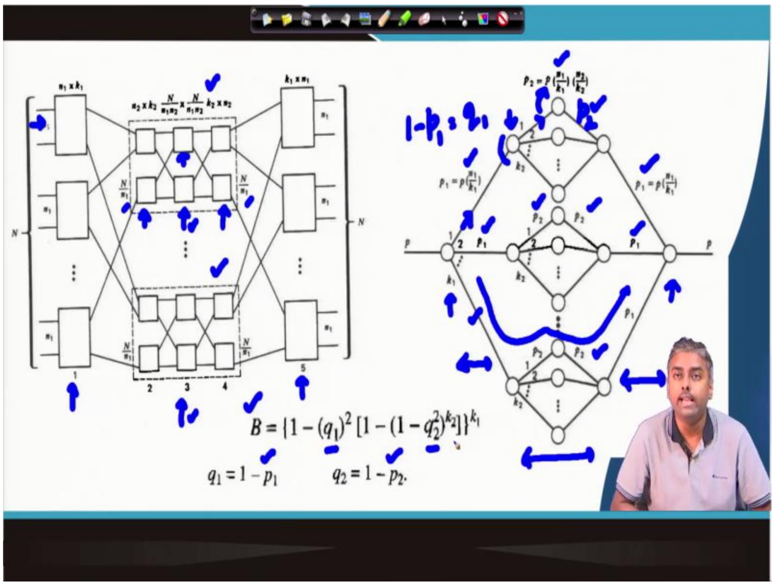
Local Loop: Two wire switching matrix (a) square (b) Triangular (folded)
Trunk: (a) Rectangular crosspoint array, (b) Graded rectangular switching matrix

Limitations:
High number of cross points – most of them are not utilized (no sharing)
Capacitive load effect

Design Challenges:
Sharing – availability issue – how to make it non blocking

Solution:
Multi stage switching – Clos switch

(Refer to Slide Time: 00:47)



$$B = \{1-(q_1)^2[1-(1-q_2^2)^{k_2}]\}^{k_1}$$

$q_1 = 1 - p_1$ $\quad$ $q_2 = 1 - p_2.$

So, if we go forward, we are. So, we have already explored the 3-stage switching, the Clos switching we have seen how that can reduce the number of space switch elements by sharing them. So, this is something we have argued we have talked about different kinds of blocking configurations. So, re-arrangeably non-blocking strictly non-blocking those things we have already talked about.

We have also given something on Lee's graph to evaluate how, if we introduce some amount of blocking, how do we analyze the performance; that means introducing the switching matrix and then correspondingly calculating the blocking probability. So, while designing, we can guarantee that this will be the blocking performance for these kinds of switches, and we have also seen how that further reduces the switch dimension.

Now, what we will do, we will take that forward to see if these 3 stages can be, in general, extended to more multi-stages and whether that can give further benefit in reduction. So, as you can see, this is a complex diagram, but it is not as complex, so what has happened if you think about it is like this. So, right now, probably you have this is the first stage, this entire thing is the second stage, and this is the third stage, actually.

So, it is a general 3-stage switch again, but what we have done now. In the middle stage, which is the sharing stage each of this middle stage, there were k elements. So, we have already discussed that there can be k elements. So, those k elements we further subdivide into multiple 3 stages. So, let us say we have actually meant N number of input ports are there that have been subdivided into small n 1 blocks ok.

So, this is n 1 cross k 1, k 1 is the middle number of stages ok. So, now, if this is n 1, we know that this switch will be capital N divided by n 1 cross capital N divided by n 1. So, this is the switch dimension. Now, this can be this particular switch; you can again see it as a 3-stage switch. So, for which this will be the first stage. So, if you see that is the first stage, that is the second stage, and that is the third stage, right?

And then again, what is happening is we subdivide into some n 2 input block-wise switches, and there will be middle stage k 2 number of stages ok of course, for our strictly non-blocking switches, that k 2 will be according to 2 2 n 1 minus 1 and similarly k 1 will be 2 n 1 minus 1 so, something like that.

So, those non-blocking switches, if we wish to design that, will be the case, but we can also design blocking switches. So, that is something we will see later on. So, what we are doing is that middle 3 means 3 stages middle stage switch; we are again sub-dividing into 3 stages, and for each of these elements, we are actually sub-dividing. So, this element, this element, and all the k-1 elements we are subdividing into multiple 3-stage switches ok.

So, again what will be happening is there will be sharing in this middle stage. So, in this middle stage, there will be sharing again. So, similarly, we can take this particular method forward and take this middle stage switch again that can be further subdivided into 3 stage switches. So, like this, we can go from 3 stages to 5 stages, 7 stages to 9 stages, and so on; as the switch dimension increases, this will give us further better benefits, ok.

Now, if we now ask that ok after this, how do I evaluate suppose I do not want to do a strictly non-blocking switch because if it is a strictly non-blocking switch, we know how to design k 2, k 2 will be strictly 2 n 1 minus 1. So, that is a clause argument. Similarly, k 1 k 1 must be 2 n 1 minus 1.

So, these things we already know if it has to be strictly non-blocking, but suppose we want to do a blocking switch, and we want to take this concept towards a modified Lee’s graph for this multi-stage switches; how do I do that?

So, again we drew Lee’s graph, but now you can see earlier Lee’s graph was there. So, there was a single stage. So, that was represented by this one input; this was the output and all the middle stages, now the middle stage; each of these middle stages has been further subdivided into 3 stages.

So, that is exactly what is happening in each of these middle stages; they have been further subdivided into 3 stages, and that is the input of these 3 stages that are the output of those 3 stages this is the these are the middle stages.

So, what we can see, is there will be how many numbers of these things should be k 1, how many numbers of these things should be k 2 ok. So, k 2 stages, k 1 stages, and now we have to see how do I evaluate the blocking probability; again, we will take the same concept of load.

So, let us say each of these links is ok, which is input to one of these input stages ok. So, for each of these links, if they are putting p amount of load so; that means p fraction of time that particular link is occupied the way we have discussed that ok. So, p fraction of time if that input link is occupied so, as you can see again, each of these output links what should be the occupancy. So, according to our previous discussion, there is n 1 such link over here you can see n 1 links.

So, n 1 into p will be the overall load input to this switch, and this is subdivided as they are equally divided into k 1 number of ports. So, it should be p 1 should be at this link; it should be p into n 1 divided by k 1 ok, with a similar argument. So, now that load is being fed into this switch, if you concentrate on this switch again, we will be able to discuss it. So, how many input ports are there over here that are actually n 2?

So, n 2 number of ports. So, p 1 into n 2 divided by k 2 so, p 1 which is this one, p n 1 by k 1 into n 2 by k 2. So, that should be this links occupancy probability and because of symmetry each of the links. So, if this is p 2, that should be p 2 as well, and this should be p 1, and so on.

So, you now populate all this p 1, p 2, so like this p 1, p 2, p 2, p 1, something like that everywhere, and then you just do a similar thing as we have done for 3 stage switches. So, you calculate and evaluate the blocking probability ok.

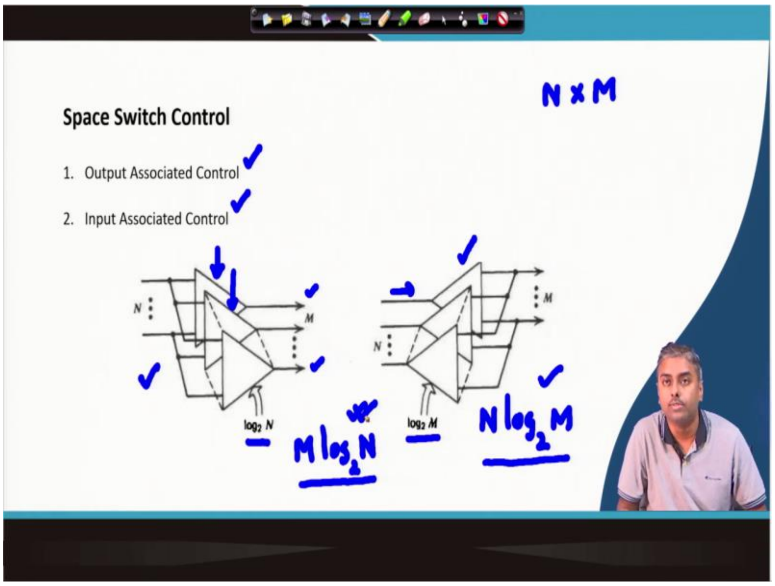
So, for blocking probability, first of all, you need to try that each of these links is blocked ok. So, this link is blocked this entire link; this link is blocked, this link is blocked, then only it will be blocked, right? So, for this link to be blocked, this should be blocked means this, this. So, there are three segments, right this segment, this segment, and this segment; one of them should be blocked, ok.

So, that is the probability we are trying to find out, and from there, we are trying to find out the overall blocking probability. So, if you just evaluate the way we have done for Lee's graph with the algebraic manipulation and the probability calculation, we can find out the overall blocking probability should be this one where q 2 and q 1s are given by this formula. So, this is if this is p 1 1 minus p 1 is q 1, if this is p 1 1 minus p 1 is actually q 1 as mentioned over here, and 1 minus p 2 is q 2 ok.

So, from there, you can evaluate the blocking probability, and now you can actually start designing your n 1 n 2 or, for a given n 1 n 2, what should be the overall blocking probability? So, this is something you will be able to evaluate. So, like this, we can take this forward towards the 7th stage, 9th stage, 11th stage, or whatever stage you want to make it.

So, it is very easy now you know how to generalize a switch structure for multiple stages and how to actually evaluate the blocking probability by putting into an appropriate Lee's graph and then evaluating the blocking probability. So, this is the way you design your switches, ok.

(Refer to Slide Time: 09:38)



Space Switch Control

1. Output Associated Control
2. Input Associated Control

So, after knowing the space switch design now what we will talk about another very important thing; how do I control these switches, because see the switches have to be automatically controlled; we have learned the story of Strowger he wanted to make the switches automatically controlled, should not be manually controlled so, this automatic controlling of switches.

So, suppose I want to reconfigure the switches one time somebody is talking to somebody, and then I want to reconfigure it so that it gives a facility that he can talk to some other guy, ok. So, this auto-configuration or auto-reconfiguration requires control in the switch. So, we will now talk about the switch control ok.

So, we will first talk about the space switch control, of course, because we have been discussing space switch; later on, we will see how to do a time switch control later on once we do the architecture of the time switch and we understand how the time switches are controlled. So, let us try to understand how the switches can be controlled. So, the space switch can be controlled in two different ways. One is called, as I have listed. So, output-associated control and input-associated control.

So, basically, as you can see, what is happening? So, there are let us say I want to design a switch, which is actually N cross M; ok, I have just taken 2 arbitrary numbers; it can be m equal M might be equal to N, then it will be a symmetric switch N cross N, but I am just making it more generalized.

So, the N input port is going to the M output port, and I want to take control of the switches ok. So, how do I do that? How do I actually design the structure of the switches so that they can be controlled?

So, it is generally done by a digital logic circuit. So these are digital switches and digital logic circuits. So, you know about multiplexers and demultiplexers. So, it is in the digital circuit. So, it is done by that only. So, as you can see, this N input ports, you actually subdivide it ok. So, it is multiple fans out, you take from each of the input ports, and you actually organize for each input port, sorry each output port one, one selector switch ok.

So, what are these selector switches? It is actually this you can think about as a multiplexer; ok, so; that means this multiplexer, whatever input it will have depending on the selector logic ok, whatever it will have some logic like the multiplexer has. So, it will choose which particular port to connect to the output ok.

So, what now, eventually, we are doing, each port we are actually putting at the input of each of the multiplexers ok like this, each of these input ports will go to one of the multiplexers. Now, the selector logic there will be selector logic for each of these multiplexers, ok. So, that will decide which particular port because, as you can see, this guy is getting input from each of these input ports, all N input ports.

Now I can, through the logic, I can select which particular port I should connect over here, and that makes my switch because, for a particular input port, I am giving a particular output port ok; as you can see, it is as simple as that. So, all I need is a selector circuit and associated selector logic; this is how many bits are there; as you know, for a multiplexer, as many inputs are there, log to N number of inputs are required.

So, as you can see, there are M multiplexers. So, M logs 2 N number of bits will be required to control this whole switch ok. So, as you can see, we are selecting the input ok. So, basically, we are selecting the output. So, that is why it is called; from the input, I am not selecting anything; I am giving input feeding to everybody, and at the output, I am selecting which one to take. So that is why it is called output-associated control.

If we just do the reverse logic, ok. So, instead of doing a multiplexer, we put a demultiplexer over here, and we do the same thing over here; from the demultiplexer, it will be selecting which output port it should this input this particular input-output port it should go. And like this, for every input, if I select a corresponding output, then that is the switch or switch control which is input-associated control because of the input I am selecting.

Again, because as many output ports are there, log 2 of that that many switch control will be required or demultiplexer control bit will be required. So, how many bits will be required over here, N number of demultiplexer? So, N logs 2 M in each of these demultiplexers, ok.

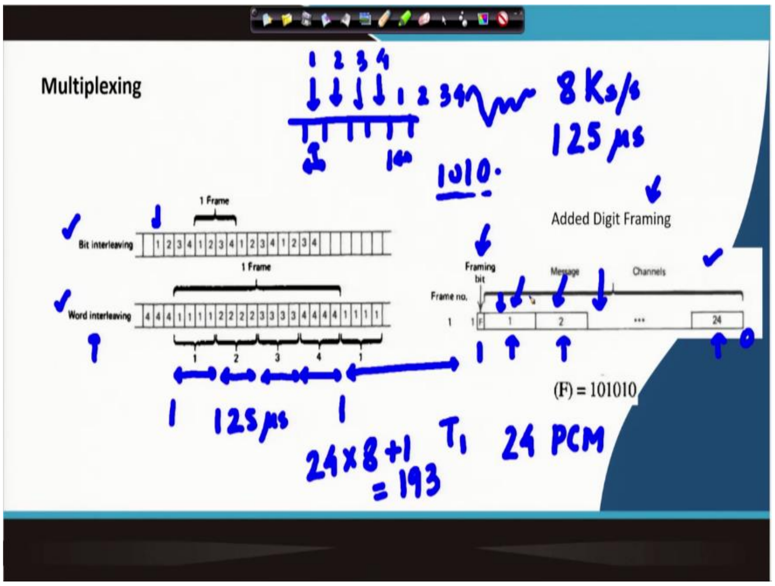
So, depending on the input associated or output associated, if M and N are the same, it will require the same number of bits to control them, but if the M and N are not symmetric, then accordingly, if you have an output associated or input associated accordingly this will be the amount of switching bits or control bits that will be required ok.

So, this is something that will be required later on; we would also like to see how many memory bits are required to control a switch. Now, you have understood for a particular N cross M space switch which can be very easily realized by banks of either multiplexer or demultiplexer, the digital logic circuits, and, accordingly, how many control bits are required that is also very clear.

So, we know now that this should be the associated switch architecture, and this should be the associated number of control bits that are required for that kind of switch if it is output associated; if it is input associated, the other one will be happening over here. So, this is something we have understood. So, basically, what we can see now our switch design is complete because we now know how to design an associated digital space switch of N cross M ok.

And we have also seen how to use those blocks to design a multi-stage switch; this is also something we know; we also know how to evaluate the blocking performance or design a switch that is non-blocking while sharing the switching elements. So, these things we have already explored, we right now we have also understood how to control a switch ok.

(Refer to Slide Time: 16:44)


Multiplexing

So, once this is being done, ok. So, the next thing that we should understand is probably the space switch is being done right. So, now, we want to go towards time switching; there is another domain of switching which is called time switching. We will see that, that is also very interesting, we will see the complexity of that, we will see the contrast between the space switch and the time switch, why the time switch is so important, what benefit it gives, what are the disadvantages of it.

So, we will see all those things can we combine these switches. So, all these things we will try to explore. But before going to the concept of time switching may be something more is required, which is called multiplexing ok. So, as you might have seen, that means whenever we are trying to transmit the PCM bits ok PCM encoded bits. So, what happens is that I have a voice channel. So, let us say I have a voice signal, sorry.

So, I have a voice signal, ok. So, that gets encoded so; which means I do Nyquist sampling ok with 4 kilohertz assumed to be the maximum voice this one or the bandwidth and then doing 8-kilo samples per 2 of that. So, 8-kilo samples and each one of these samples are encoded into 8 bits by PCM encoding. So, we get 64 kbps or something like that.

So, basically, what happens whenever we talk about these 8-kilo samples per second? Ok. So, basically, the inter-sample spacing will be 125 microseconds. So, what is happening? When we do PCM encoded, then every 125 microseconds, there will be 8 bits coming from that particular voice signal ok. So, this is what we generally do.

If now, we want to multiple, multiple such voice signals in TDM multiplexing; we have already talked about TDM multiplexing. So, that is exactly what we will be trying to do. So, how do we do this TDM multiplexing? So, when we try to do this TDM multiplexing so, what do we do? We actually repeat these things. So, suppose I have this 125 microseconds. So, what I do, let us say 2 of these or 4 of this TDM 4 of this voice signal, we want to do TDM multiplexing.

So, what can we do? Within these 125 microseconds, I can subdivide it into 4 blocks ok. So 1, I can put it over here from the voice sample 1, whatever samples we have. So, for all 8 bits, I can put it over here, then for 2, I can put it over here, and then 3, and then 4, ok. So, in 125 microseconds from this voice sample 1, there will be 8 bits. So, all those 8 bits I will put over here.

Then it will be interleaved. So that is why it is called interleaving. So, it will be interleaved; that means, next, it will give a chance to put the block or 8-bit from the second voice signal and so on. And this will be after that once this 125-microsecond boundary goes away, then, or it is over, then I can again repeat 1 2 3 4 and so on; this pattern gives keeps on repeating if I wish to demultiplex them, so what I do.

Suppose I want only voice signal 1. So, I just take this portion and then skip this whole portion; again, I take this portion and skip the next 3 parts, and I keep doing that. I will be getting only means of voice signals or samples from actual voice sample 1. And then I can reconstruct it through low pass filtering the way from PCM encoded these things actually goes back to actual voice. So, I can do that.

So, there will be D 2 A converter and then low pass filtering. So, all those things we can do are ok. So, reconstruction we can do. So, this interleaving or multiplexing can be done in two different ways, as you can see. So one is called bit interleaving, and the other one is called word interleaving. So, in bit interleaving what we do, instead of doing this, this was actually a word interleaving that we have talked about.

So, as you can see, if it is represented by 4 bits, so, all 4 bits from 1 have been put in 1 block. So, that is why it is called word interleaving; if it is 8-bit PCM encoded, then all 8 bits will be put in 1 block next, followed by from 2, followed by from 3, and followed by 4. So, this is actually 125 microseconds, and then the pattern gets repeated next onwards.

So, I can do this word interleaving or I what I can do, I can keep on interleaving every bit. So, bit 1 followed by 2 3 4 and then again bit 1 followed by 2 3 4. So, like this four times, I will repeat. So, that makes my 125 micro-second frame, and I keep repeating this; as you will see later on, word interleaving is one of the most popular ones. So, we will probably be taking them as my interleaving circuitry or multiplexing ok.

So, let us try to see on top of this; you might be requiring some more thing which is called the framing bit. Ok, this is part of synchronization, but because we are discussing it, let us try to see or understand one of the multiplexings that have been popularly used or standardized in the US, which is called T 1 multiplexing ok. So, in T 1 multiplexing, what happens is basically 24 PCM encoded voices are multiplexed together.

So, it is actually trying to multiplex 24 such PCM-encoded voices. So, if I now try to see this is the frame structure of 125 microseconds. So, this is 125 microseconds; this is the word interleave. So, basically, 1 that 8 bit followed by 2 words followed by up to 24, and then this will be this pattern will be repeated ok. So, we have to do that.

Now, as you can understand this, after interleaving, it will be a stream of bits, ok. Now, at the other end, I have to understand because the location is the address. So, where from this will a stream of bit 1010 come now? I need to understand very clearly where exactly the particular frame is starting.

So, the starting point or initial point of this particular frame is very important. So, I need to really understand how this is overall frame is being structured, where is the starting point and where is the end point of a frame. So that I can understand very clearly those 125 microseconds; once I understand that, then I can actually know that, or I know, according to the standard, there are 24 such things. So, every 8 bits, I can get the next one ok.

So, identifying the frame starting point is very important. So, that is why they have added an extra framing bit ok. So, basically, what is happening now? If you see the whole frame structure, there are 24 bytes, so; that means 24 into 8 bits are there, plus 1 frame bit has been added for identifying the frame structure. We will talk about how this one frame bit actually helps us to identify the frame structure, but overall what we can see now is that 193 bits are being formed ok within 125 microseconds.

So, every 125 microseconds, there will be 193 bits; out of them, 192 are actual data bits or actual voice bits ok, or PCM encoded voice bits; there are 24 channels that are multiplexed together in a TDM fashion, and one additional bit has been added for framing.

Now, how do I do this framing? So, basically, what will be happening? As you can see over here, every 125 microseconds, there will be a 1-bit ok. So, this 1 bit now, if I give in every bit, means every frame there will be this 1 bit, this 1 bit if I encode it in this fashion that if this is 1, the next frame it will be 0 and next frame it will be 1 so, 101010.

So, as you can see, this pattern gives me a 4-kilo Hertz signal because this pattern is repeating every 125 microseconds; what is happening is you are actually getting 1 followed by 0 and another 1 followed by 0. So, is it actually giving you a sinusoidal ok, or does it mean it’s first harmonic sinusoidal will be of 4 kiloHertz?

This 4-kilo Hertz, as you can see, nowhere this PCM encoded bits will have 4 kilohertz because 4-kilo Hertz generally the voice signal will be up to 3 kilos 3.3-kilo Hertz, right? So, none of the PCM-encoded voices will have that pattern. So therefore, this 1 0 1 0 pattern repeated will be a unique pattern that will never be repeated in any of these subsequent bits at its corresponding places in subsequent frames.

So, this is a unique pattern. So, we wanted to actually identify a unique pattern; this is how the unique pattern has come. So, the maximum frequency that can be there will never be there in the voice signal. So, because of the bandpass filtering that will be done before encoding it, PCM encoding it.

So, will always be putting 3.3 kilo Hertz probably, which is the maximum we want. So therefore, this 4 kiloHertz signal will never be there, and the pattern we are creating that is actually coming from 4 kiloHertz.

So therefore, that particular pattern will never be repeated in any of the subsequent bit locations. So, that makes this bit location unique; if we go through a subsequent number of frames, after many frames, we will be able to identify this pattern. So, what a frame synchronizer does?

Actually, every 193 bits, it takes a bit. Suppose I randomly choose a bit because, at the initial beginning, I do not know. And every 193 bits, I go and try to see whether I am getting that pattern 1 0 1 0 this pattern; if I am not getting that, I keep shifting to the left or right, ok. So, I keep doing this shift till I get this 1 0 1 0 pattern.

So, there will be a few initial frames initially. Once I go through this frame, I will get my synchronization because of this framing bit; once I get the synchronization, then everything is known. I will go every 8 bits. I will get my multiplexer, ok. So, that is the technique by which we will do the multiplexing. So, once we have understood this multiplexing in the next class, what we will try to do is we will try to appreciate how this multiplexing can be used for time switching.

Thank you.