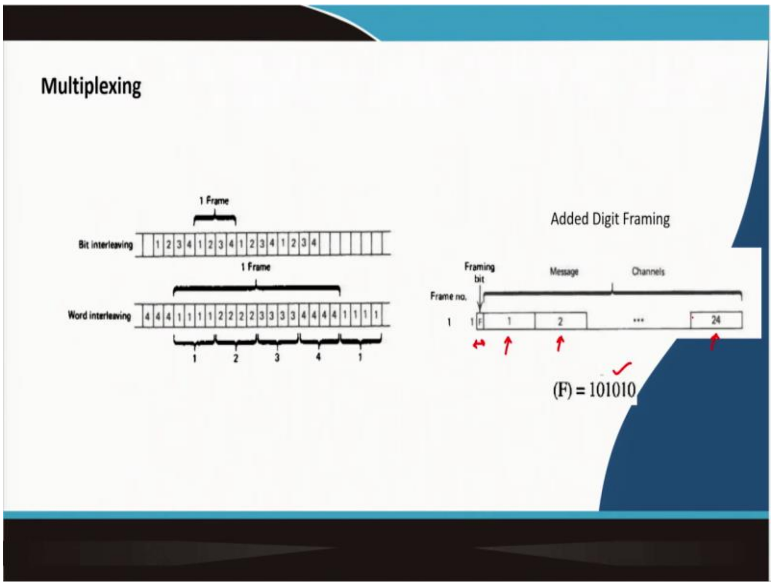
Communication Networks
Prof. Goutam Das
G. S. Sanyal School of Telecommunication
Indian Institute of Technology, Kharagpur

Module - 03
Circuit Switched Networks
Lecture - 10
Time Switch

So, today we will discuss Time Switching. So, a quick recap of what we have done so far is the discussion about space switching and their design. So, now, we will see the other dimension of switching, which is time switching. Before time switching last class, we have already discussed multiplexing.

(Refer to Slide Time: 00:50)



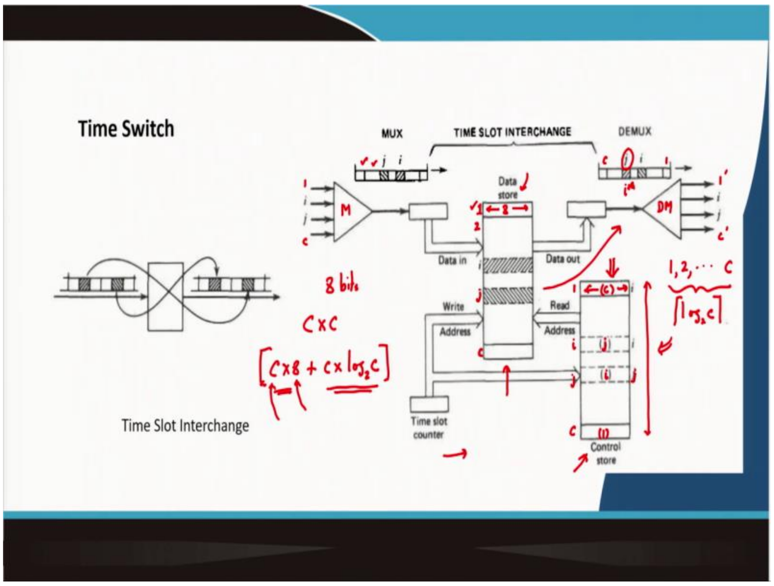
So, let me go. So, this was where we have been. So, we have been discussing about multiplexing. We have already told a T 1 framing which is actually the 24 PCM encoded voice signal, is being actually done; the multiplexing is done with the word. So, this one word interleaving ok. So, this is what we have done, actually.

So, each word is 8 bits, whatever the PCM encoded bit numbers that you have. So, each sample is encoded by 8 bits. So, it is generally encoded by 8 bits. So, as you can see over here in this diagram. So, you have this channel 1 followed by channel 2 and, like this, up to channel 24.

And we have also discussed this framing bit which is an important criterion actually. So, what can we see? So, 24 channels of 8 bit 192 bit and 1 bit for framing, which is having a sequence we have already discussed this also, 101010. So, that is, the 4-kilo Hertz signal may not be present in any of the other bit patterns of any of those voice signals because they do not have that frequency of 4 kiloHertz.

So, that is where we do frame synchronization. So, 193 bits per 125-microsecond frame structure. So, this one is 125 microseconds, and each one of them carries 1 sample and 1 PCM encoded sample from every voice signal ok. So, that is exactly what is happening.

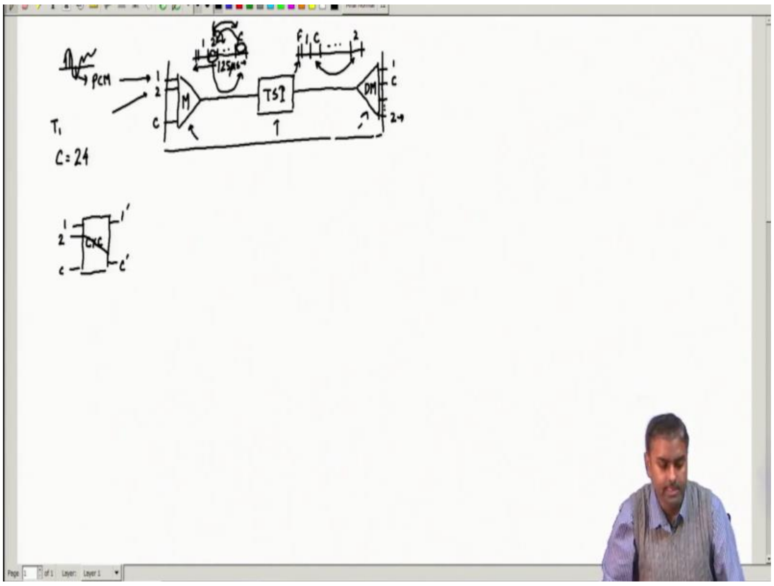
(Refer to Slide Time: 02:36)



Now, after we have understood this time switching, we will be now talking about this. Sorry, this multiplexing will now be talking about this time switching. So, it is actually called it is the name is known as TSI or Time Slot Interchange, or it is called time swap; the switching is done in terms of swapping the particular bit location. You know that now this means frame location within a frame that is the address.

So, the first one first voice is at location 1; the second is at location 2, and so on. So, now, if you interchange the time means location, that will probably make switching. We will try to see we will try to appreciate how this actually makes a switch.

(Refer to Slide Time: 03:31)



So, let us try to see. So, what exactly is happening over here? Let us try to draw that. So, this is our multiplexer; multiplexer has 1 2 up to let us say C number of voice samples, right? So, this is the multiplexer output. So, this is actually the frame structure that is coming out.

So, each one of them is actually having a corresponding voice signal which is being sampled at the Nyquist rate. Each of those samples means 10 PCM encoded, and that is exactly coming over here, channel 1.

Similarly, another voice is coming to channel 2, and so on. Then the multiplexer actually does that frame structure. So, it actually takes each byte from each one of them in a round Robbin fashion, one after another. This means if there are C, C is actually 24 for our T 1. So, if it is T 1 standard, so, it depends on the standard.

So, if you go to this standard, then it is 24; there might be 30 channels. So, there might be other standards also. So, whatever it is. So, C number or 24 number of such PCM encoded 8-bit voice samples will be taken, or word will be taken, and then the frame structure will be built upon after this multiplexes.

So, this multiplexer actually gives you that frame structure output with 1 framing bit and then followed by 1 2 up to C ok. And this is actually your 125 microsecond, and this gets repeated after another. So, if this is a multiplexer, now let us try to see what is time switching actually.

So, let us try to understand that. So, in a switch, what happens is I have a box, I have this 1 to, let's say, C number of this one; let us say I want to create a C cross C switch. So, in a space switch, what happens? I can connect. So, let us name that 1 dash to C dash. So, anyone to anybody I can connect with. So, that facility I will have to give that somehow. So, here again, I have the C number of voice signals which are coming to the multiplexer.

So, first, any time switch will have a multiplexer at the input because it has to construct this particular frame structure. So, that is the most important part, basically ok. So, that is something you will have to do. After that, that particular structure will be talking about time slot interchange, or TSI will be putting.

So, basically, what does it do? It does the swapping. So, let us say I pick some 2 and C I want to swap. So, basically, I want to do switching between 2 to C dash. So, this switching, I want to do ok. So, suppose this is the switch I want to do. So, what I will be doing this TSI will actually swap these things. How it does the swapping that we will see later, but it intacts the framing bit, and then 1 remains in his place in the place of 2 it keeps.

So, this data will actually put it over here, and this data it puts it over here. So, C will come over here, dot dot dot, and 2 will come over here, ok. Now followed by a demultiplexer. Now demultiplexer, what does it mean? Exactly with the frame synchronized frame, it actually keeps the output according to this sequence.

So, 1's data comes out from here now, but C's data will come out from here, and so on, and from here, 2's data will come out. Now if you just see this structure, it acts like a switch, as you can see, due to this interchanging. So, because I have done this interchanging, so, due to this interchanging, what is happening is that 2's data is coming from this last output. So, that means that switching has been done.

So, this is the technique of doing time switching. So, time switching means there should be a swapping of these slots because slots are the address now; once I swap the slot because of this multiplexer and demultiplexer, you will see the effect of switching in the spatial domain. So, basically, if you talk about a time switch, it is a combination. Combination of a multiplexer, a time slot interchanger, and a demultiplexer. So, all together, it makes a switch ok.

So, this is another dimension of switching. You will see that this switching you do not require a means special port or any special switch matrix. Over here, all you have to do is logically swap the location of the bits or bytes, let us say, in the sequence of the frame. So, whichever sequence they are in the frame, you just swap that.

And if you do that, what will happen? That is the logic you will always have. So, basically, what you do every frame, you will be swapping these things. So, that is the logic; we will have to keep it.

So, for every frame, you have the frame synchronization. So, you know from this bit. So, this is the starting of the frame, and then from that synchronization, you will keep on doing this swapping. So, for every frame, you will be doing the same swapping as long as this particular switching is required.

If you change this, so later on, after some time, probably after, let us say, 10,000 frames, you need to change this. So, now, if you want to swap between 2 and 3, then what will you do? So, from then onwards, you will change this particular configuration. You will swap between 2 and 3 instead of 2 and C; immediately, the switching will be done between port 2 and port 3. So, this is a phenomenon that is termed as time switching ok.

Now let us go a little bit deeper into this TSI unit. So, regarding the time slot interchange, let us try to understand that part. So, the TSI unit ok. So, this is particularly a TSI unit. In that TSI unit, as you can see, you have this. Let us say 1 to C number of elements you have a multiplexer, as we have discussed.

This is the whole unit of time slot interchange this whole thing, and this is followed by the multiplexer, and then you have 1 to C that is going out, or you can call it 1 dash to C dash. So, suppose now you want to switch between the i and j location. So, jth location and its location. So, you want to swap this; this is what you want to do. So, basically, what time switch does is something like this, or time slot interchange what it does?

Whatever data these frames are coming because you know the synchronization. So, basically, you read byte by byte. After the synchronization is done, so you know one frame. So, take that entire one-frame data and byte by byte data you read, ok. After reading it, you actually store it in data storage. So, as you can see, the data storage facility has basic storage starting from 1 to C, and many storage locations are there.

Each one of these data stores can actually store a byte. So, if it is PCM encoded 8-bit, then it can store 8-bit. If your encoder is different, let us say 10 bits or any other things, then that data store length also should be of that many bits. So, over here, it is 8-bit. And how many storage places are there? C storage places are there.

Now, what is happening? Suppose you want to; there are two ways of doing these things. So, let us talk about one of them. So one of them is when you are writing it into the data storage. So, basically, from this side, from this left-hand side, you are actually writing it into the data storage, ok, and on the other side, you will be actually reading it from there, ok.

So, there are two operations actually eventually, you are writing, and then eventually, you are reading. So, when you are writing, what you have to do is all those bytes that are there; you can sequentially write them. That means, the first one this first one you can write it over here in the first location. The second one, you can write it in the second location, and so on up to C, you can do that ok.

So, you can also do other things. So, initially, let us think about that is what I will be doing. So, I will be storing them sequentially, but while reading, I need to do swapping. So, suppose I and J I want to swap, so basically, what will I do? I also have a memory or control bit. So, this is the control bits. What do I do? Again I will be having a 1 to C number of specifications where the control information will be there ok.

In that 1 to C, suppose i to j, I want to swap. That means ith data should go to jth data, and jth data should come to ith data. So therefore, in the control of what I will write, in the ith location, I will actually store the value j, and in the jth location, I will store the value i. What does this signify? So, while reading, I will mean this from this one when I am reading them to the multiplexer now; what will I do?

I first do not read it from here. I first go to the control location. So, control location in the 1st this one, whatever is stored over here that data I will be reading. So, I do a random reading over here guided by, of course, the control bit. So, this control buffer, whatever he is guiding me accordingly, I read it ok.

So, if I go to suppose the ith data has to be fetched. So, in the ith location, what do I need to put? So, in the ith location, I first see which data I have to fetch. So, jth data. So, I go to the jth location whatever data is stored that I put in the ith location. So, this is the ith location, and I put the jth data. And similarly, when I have to read the jth one, first, I go to the jth control bit or control information, and I see ith data has to read has to be read.

So, basically, I go to the ith location of this data storage, and I read. Immediately what happens? I will get this swapping. Whatever swapping I want, all these things can be stored over here in the memory location, ok. So, I can simultaneously be swapping multiple things; i to j, i can in between 1 to C, suppose I want to do. So, over here, I will store C, and over here, I will store 1.

So, accordingly, he will read. Whenever he sees the first data, he sees which one to read; the Cth one, so he will fetch this data. So, over here, the Cth data will be stored ok. and similarly, when he goes to the Cth location, he will actually fetch the first data. So, accordingly, C and 1 also will be fetched. And immediately, you will see after demultiplexer that switching will be the corresponding switching will be done.

So, this is how switching time switching is actually understood ok, or realized in a particular circuit. So, as you can see, it is a digital circuit; you have j, just two storage; one is the data storage, and one is the control storage ok. Now if you try to ask how much storage is required overall. So, over here, as you can see, I do not actually need any kind of switching element.

So, switching elements are not required; it is just through logic and memory reading, and memory writing I am capable of doing the switching. So, it is about how much storage is required. So, as you can see, if I have C things multiplexed. So, if I wish to actually construct now, C cross C switch ok, and each one of them is actually multiplexed with an 8-bit PCM encoded voice signal.

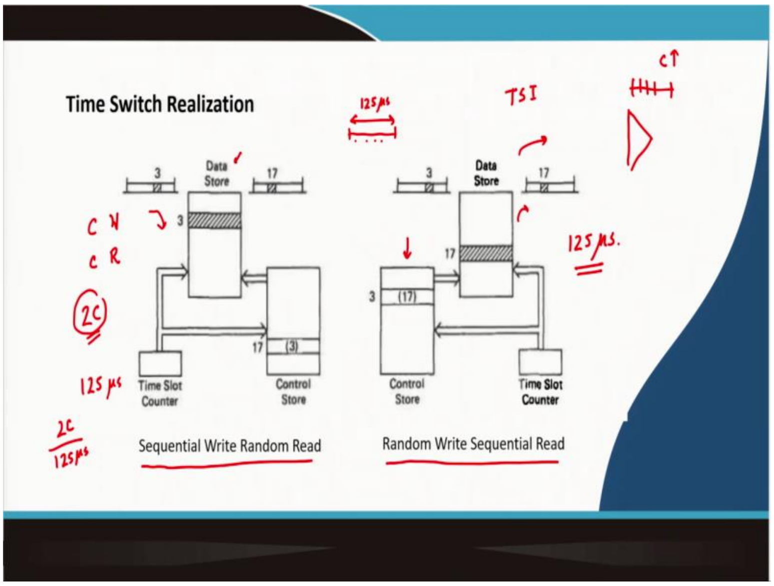
So, if that is happening, then how much storage for data do I require? So, C places of 8 bit. So, C into 8 that many bits are required for data storage. How much or how many bits are required for control? So, how many control locations are there? That is still C. How many bits per this one per control information?

So, now as you can see what the information is, I will have to store, maximum, up to C. So, basically, 1 2 up to C, how can they be represented by binary bits? So, that should be log 2 C; this many numbers of bits are required to represent all of them. So, basically, each one of them will require that many bits; of course, if log 2 C is not an integer, I will have to take the ceiling of that ok.

So, let me generally say C is taken, so that it is an. If it is an integer, so I can put it as log 2 C ok. So, these many bits are required for my data, which means overall control realization or overall time slot interchange realization. This is for the data storage, and this is for the control ok.

As you can see, if I represent a different PCM encoder, then 8 will be changed. If I represent a different number of multiplexed data streams, then C will be changed, and accordingly, my data requirement will be changed ok.

(Refer to Slide Time: 18:28)



I hope this is understood. So, now, as we were discussing, we can have this time switch realization, or time switch can be realized in two different ways. One is, as we have said that we will do sequential writing and random read the one I was doing. So, earlier, in the earlier demonstration of what we have done. So, you are reading them sequentially. So, one after another, you are reading them, and then while writing sorry, you are writing them sequentially.

So, sequential write, while reading them from the storage data storage, then you are reading them randomly according to your buffer location. You can also do the reverse. What can you do? While storing them, you actually store them from memory; this is a control requirement.

So, you store them accordingly, and then while reading, you do a sequential read. So, that will be random writing and sequential reading; both of them are ok. So, either of them you can implement ok. So, this is two ways of implementing this time slot interchange; this TSI ok. So, we have already understood this part.

Now this is very good because earlier, we were actually seeing that this switching element; is a major, probably cost-consuming element, and that was giving a stray capacitance effect and all those things we wanted to get rid of. We were trying to do a space switch, which was sharing, so the number of switching elements was reduced.

But now we have got a time switch which is not requiring any of those elements. So, it is just pure digital logic, reading to the memory, writing into the memory, and reading from memory; this is all that is required, but is that all rosy? There must be there are no free lunches, right? In engineering, always, if you get to achieve something, you must be paying somewhere.

So, where exactly are we now paying? Let us try to understand that part. What is happening? As you can see over here, in time switching. So, whenever you are switching them, so, basically a frame is coming to you of duration 125 microseconds ok. So, a full frame is coming to you. Now what do you have to do to switch?

First, you will have to store the entire frame because each of these byte locations has to be stored in the data storage; then, only you can do a switching operation. Or, while storing, if you do a random write, then while doing the storing, whatever it is, you have to store them, you can do the switching, but after that, you will have to read from that memory.

So, whatever you do, as you can see, this frame will be only delivered after 125 microseconds because for the entire 125 microseconds, you will be storing them, and then the next 125 microseconds, you will be actually putting them into the channel or putting them into the demultiplexer.

So, that means every time switch eventually will always give you a 125-microsecond delay. That is always something you will always see, ok. So, this is inevitable; you cannot actually get rid of this particular part. So, any this kind of TSI element, you put as many such elements as you will be putting every one of them will give you a 125-microsecond delay.

So, now earlier your space switch which was not giving you any delay, it was very nicely designed. It was, of course, bulky, it was costly, and it was probably giving you all those stray capacitance effects, but now we have designed a switch which is good, gets rid of all these things, but now it introduces delay.

So, now how far you can withstand delay that is up to you, ok. Second thing, is there a limitation? Can I start putting as many suppose this multiplexer? This multiplexer duration, I can actually make it very big, ok. So, for this multiplexer size, I can make it very big or the duration of the frame or within the frame how many such things will be there. So, that is the number C, I can make it very big, and that might increase my switch dimension.

But is it possible that infinitely, I actually increase this dimension? Is it possible at all? So, this is something next we will have to see that there is a restriction on this particular number or value C let us try to understand that part. Let us try to see within these 125 microseconds how many things we are actually doing.

So, basically, what we are doing at the input we are writing to the memory, so that is one operation. So, byte by byte, we are writing. So, if there are C bytes, then the C number of writing operations you will have to do. And at the same time, within that 125 microseconds, you will also you will have to read from memory.

So, another C-read operation you will have to do. So, overall 2 C operation you will have to do within this 125 microsecond. Every 125 microseconds, you will eventually have to do this many operations. So therefore, there is a speed restriction. If you increase C, your speed of reading and writing has to be very high ok.

So, for memory writing and memory reading, that speed has to be very high, and there is a restriction hardware restriction for that. So therefore, as you can see, every 125 microseconds, I will have to do a 2 C operation; ok, so, if you see that this 2 C is divided by 125 microseconds. So, that many operations per second you will actually have to perform, and that should be the speed of your memory read and write.

If you increase C, of course, it will hit the hardware restriction, and you will not be able to go beyond some C value. So, that is another restriction that is coming over the time switch. So, we have now seen that there are two restrictions of this time switching ok. So, what are these two restriction? So, one is that whatever you do, you will always be having 125-microsecond delay; that is one of the restrictions.

The second restriction is there is a speed limit; you cannot indefinitely increase the value C to make the switch dimension very high because there is a hardware restriction on that. So, we come back to this again. So, we can now see that we have designed a time switch, we have designed a space switch. Space switch has some typical advantages; time switch has some typical advantages.

Now, what we wish to do, we want to actually add the virtue of these two or the advantage of these two. Can we do that? Can we take advantage of the time switch as well as the space switch while designing an overall switch? So, that will be the discussion of our next class, where we will try to actually combine these two switches, the time and space switches, to come up with a better design. So, this is something that we will be discussing next, ok.

Thank you.