Architectural Design of Digital Integrated Circuits
Prof. Indranil Hatai
School of VLSI Technology
Indian Institute of Engineering Science and Technology, Shibpur, Howrah

Lecture - 48
Hazards in Digital Circuits

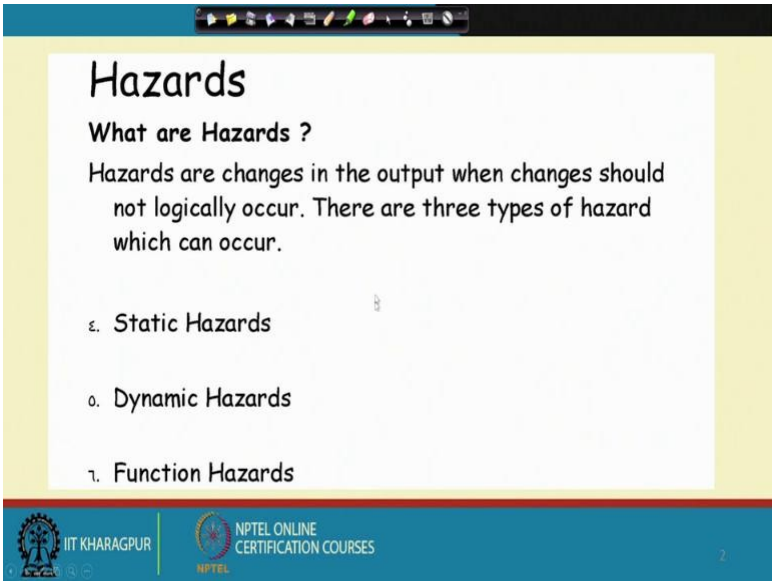
Hello everyone, welcome back to the course on Architectural Design of IC's. So, in the last class we are seeing that the issues which are related with the timing closure; that means, whenever we are doing this digital VLSI design so, at that time we have to do this timing analysis. So, static timing analysis only we have consider, we have not set or we have not seen this dynamic timing closure at that time or the dynamic timing analysis.

But the concept of dynamic timing analysis is that we have to put the timing; that means, the some set of or selected set of input vectors and in then we have to do this timing analysis. So, as that is simulation based. So, that is why; that means, you have to need the simulated to do that on the real time or that perform the dynamic timing analysis you need the logic simulator tool; not only the timing analysis tool set.

Then today again we will talk about another; that means, another concept which is basically presented in the circuit design or while we are doing these architectural designs of digital integrated circuits; so, that is the Hazards. What is hazards? So, hazards means is some unwanted signals; that means, actually how one digital circuit works. We gave some of the input and then we are we based on the logic or based on the; that means, operation of the function we know that this will be my expected output.

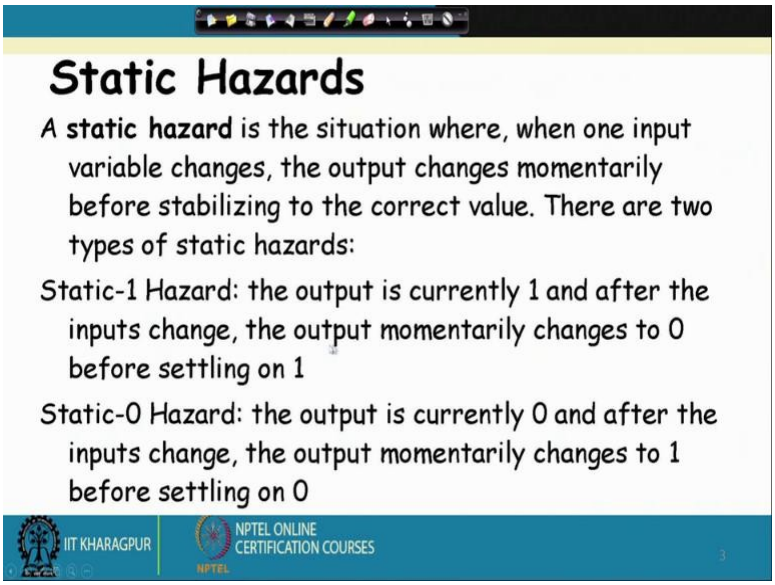
But sometimes it happens that though we have done the logic; that means, though we have not that means, done the logic means suppose some missing of the logic it creates some unwanted signal presented at the output. So, that type of that means that type of condition is known as hazards in digital integrated circuits. So, here I just wrote the thing that hazards in digital circuits or digital integrated circuit both are same.

(Refer Slide Time: 02:43)



So, the first thing is that what is hazards? Hazards are changes in the output when changes should not logically occur, means this is unwanted changes. So, there are three type of that means, hazards are present which is the first one is static hazard, then dynamic hazard, then function hazards. So, what are these hazards? When this is occurring and how we can remove we will see in today's lecture. So, first we will talk about what is static hazards.

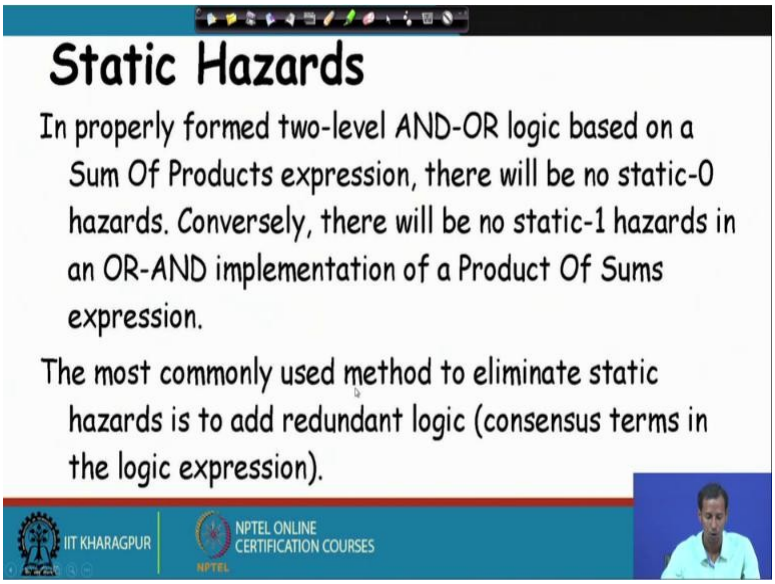
(Refer Slide Time: 03:19)



So, static hazard is the situation where and when one input variable changes, the output changes momentarily before stabilizing to the correct value. There are two types of static hazards: one is the static 1 hazards, another one is static 0 hazards. So, what is static hazard? Static hazard says that sometimes it happens that your; that means, one input you are changing, but the output is not that means, it is before stabilization that means, whatever is the correct value or whatever what I said. That whenever we put some of the input or the test vectors as the input for testing of the corresponding circuit, the output behavior also I know or the pattern of the output behavior also we know that this is the expected behavior for that particular circuit.

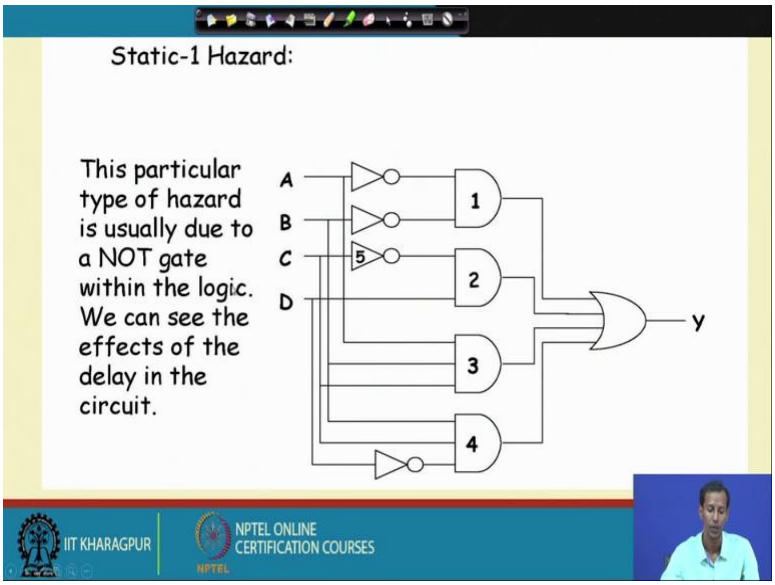
But, sometime it does not happen. The results which we will get, though the proper correct value it does not stabilized to that, it gives you some unwanted signal, some unwanted values on that for some corresponding positions. So, whenever you are getting the output is currently 1, but after the input changes the output momentarily changes to 0 before settling to 1; that type of hazard is known as static 1 hazard. The just the opposite one; that means, the output is currently 0, but after the input changes the output momentarily changes to 1 before the settling to 0; that is static 0 hazards. Mean what you will get some of the glitch, glitch means what some unwanted signal or spurious signal which is present at the output.

(Refer Slide Time: 05:21)



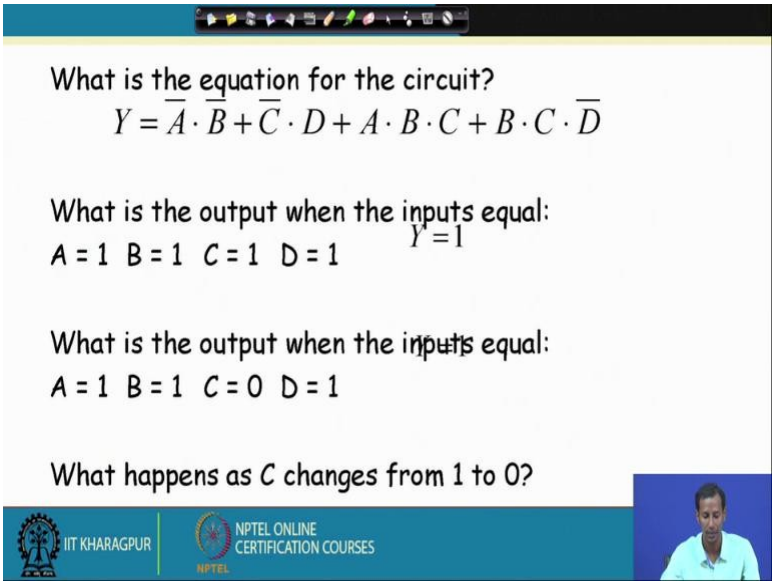
So, the static hazards mostly it is formed in two level AND-OR logic based on a Sum of Products expression and there will be no static 0 hazards. Conversely, there will be no static 1 hazards in an OR-AND implementation of a Product of Sum expression. The most commonly used method to eliminate the static hazard is to add redundant logic. So, how to add redundant logic, that also we will see.

(Refer Slide Time: 05:58)



Suppose what it says that AND-OR logic, in AND-OR logic there will be no static 0 hazard, but static 1 hazard will be there. Just opposite in OR-AND implementation or this product of sum AND-OR means some of a product and then this OR-AND means this product of sum. So, in this particular two case this static 1 hazards occurs, static 1 hazard occurs what static 1 in some a product and in product of some this static 0 hazard. So, static 1 hazard if I consider this particular example so, this particular type of hazard is usually due to a NOT gate within the logic. We can see the effects of the delay in the circuit.

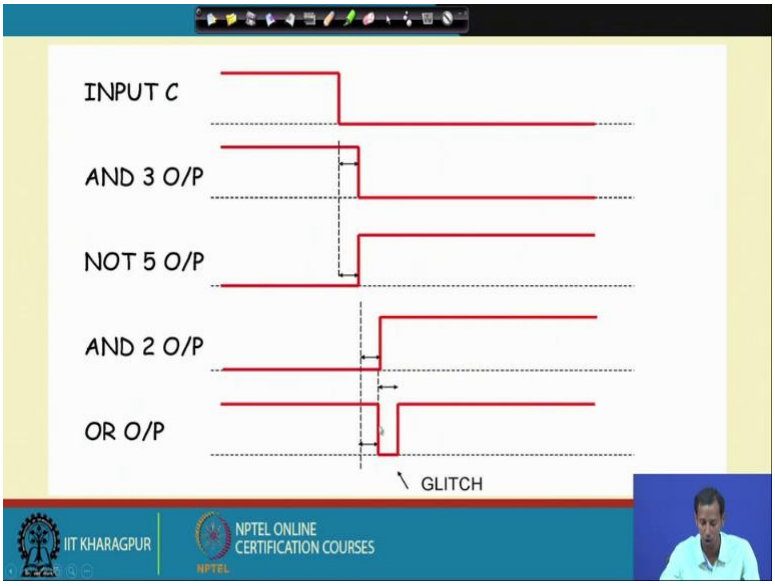
(Refer Slide Time: 07:07)



Now, suppose if I take the corresponding example of this so, at the time we can visualize the corresponding logic 1 hazard at the output which is Y. So, let us start with that. So, this suppose consider one equation where is Y equals to A bar and B bar C bar and with D and A B C and then B C D bar.

Then what is the output, when the input is equals to A equals to 1 B equals to 1 C equals to 1 and D equals to 1. When all these A B C D are 1 so, at that time the Y output value is 1. Again when all the; that means, this A B D are 1, C is 0, Y is again 1. What happens as C changes from 1 to 0? So, in these two particular cases you see in both the case Y is 1 as C is changing from 1 to 0.

(Refer Slide Time: 08:20)



If you just see, suppose if I consider the corresponding changes in this. So, input C is changing input C is basically at 1. So, then AND 3 output means this is the AND 3 output. C is 1 A B D all are 1 right so, A B D all are 1 means C is 1. So, in AND gate what happens? If the other input is 1 then it depends upon the corresponding value of the other input. Here, if you see this is the AND output, this is the third gate or I have just numbered it as 3.

So, here what is connected? B C sorry A then B and C, A B C is connected to 3. So, A and B is 1 means what it is depending on the values of C. So, here you see so, as this is 1 this is also 1. Then NOT 5 output. What is NOT 5 output? NOT 5 is the just the inversion of C. So, inversion of C means as this is 1 so, this will be 0. Then AND 2 output. What is AND 2 output? AND 2 output is C bar into D. So, D is 1 C 1 of the input is 0 means this C output 2 output, as long as C is 0 this output will be 0. So, as long as this AND output sorry NOT output from the 5 C bar is 0 AND output 2 is also 0.

Then what will be the corresponding OR output? OR output means here what I am considering this is basically, if I do not just consider these two. So, if I consider only these 2 and 3, 2 and 3 if we just consider, considering and then calculate this OR output final OR. So, at the time 2 and 3 means ORing of these 2 and 3 means 1 and 0, this is 1. Then input C is changed to 0, AND 3 is after sometime it will be 0 because, this has

some delay. This gate has some delay so, after that time now, this is basically coming to 0 because, of this change in the C.

Then NOT gate also it will have some that means, delay and then again it is becoming 1 over here. That means, the changes is occurring at these particular time, but because of the gate delay after sometime this is the AND output is changing to 0 and the NOT output is basically changing to 1. So, as long as this is 0 so, this is not that case. So, whenever this NOT output is 1, whenever this NOT output is 1 so, AND gate, AND 2 output will be what?

AND 2 output AND 2 output is one and these one. Means, what so, after that again after sometime AND 2 output is at 0 and OR output is considering these 1 and 0 that is 1.

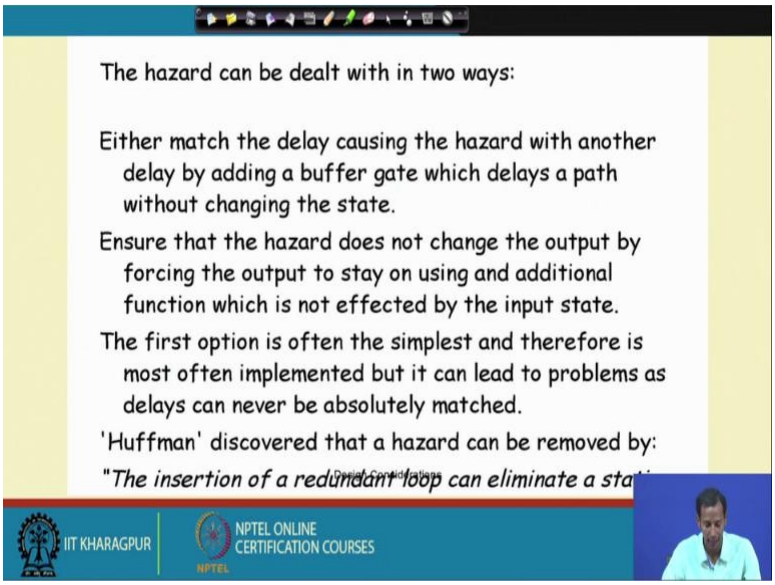
So, here you consider that after these particular time period, this AND gate now it will try to change. Why this AND 2 is having extra delay element over here because, of this is passing 2 gates. So, till now because of this NOT gate it is getting the delay. So, whenever these output is changing so, after sometime now this AND 2 will be changed. So, AND 2 is now changed to because, of this NOT which is changing to 1. So, 1 and 1 this will be 1. So, OR output this will be again change to what this AND 3 & AND 2. So, which is this 0 over here and 1 over here, sorry yeah sorry this 0 over here.

And, then 0 for 0 this will be up to after this will be 0. Then again after sometime what will happen because, of this particular terms. So, this time period it will be remain at 0, after that again it will be change to 1. Why? Because OR is passing through 3 delay element; that means, 1 then 2 then 3. Means what? If you just see so, this is at level 4 right. So, here you see AND 2 is basically changed to 1 and AND 3 is 0. So, OR will be change at what? Or will be changed after some delay related to OR gate which is this; that means, after again sometime it is changing to which is this 1 value of this.

So, it is arriving this 1 and 0 then again it is becoming 1. So, that means, but this particular 0 it will not happen. Why this will not happen? If you see in both the case the Y should be 1. So that means Y will continue its value to 1, but because of this gate delay because of this gate delay I am getting 1 0 which is happening in between of this. So, this is basically the glitch and this is static 1 hazard. Why? Because, though the output should

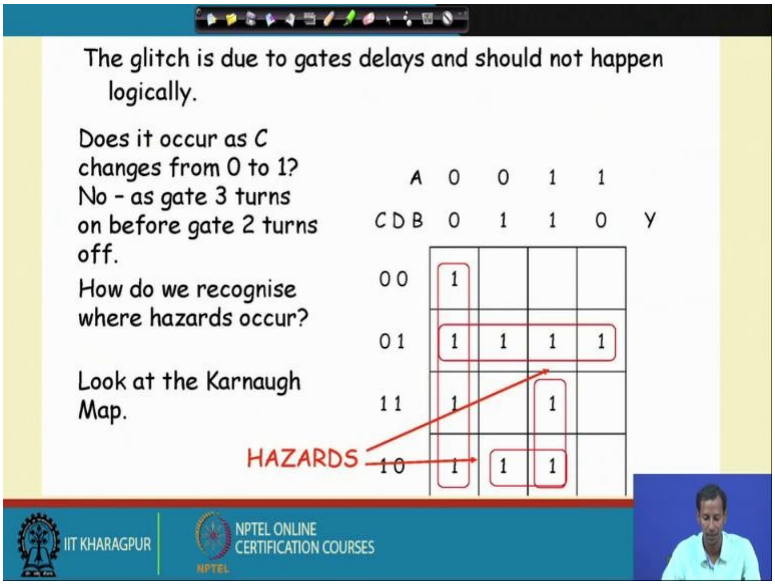
remain 1 but it is momentarily changes to 0. So, that is why this is static 1 hazard which is happening in case of sum of product. The same example if you just; that means, same kind of example if you take for product of sum; that means OR-AND case so, at the time you will get this as static 0 hazard.

(Refer Slide Time: 16:05)



So, how to deal with this hazard? So, this hazard can be dealt with in two ways, either match the delay causing the hazard with another delay by adding a buffer gate which delays a path without changing the state. That means what? Here in this particular path as because of this particular NOT gate this is happening. So, add here some of the other buffer over here so, that these two paths become balanced. So, if these two paths become balanced so, at the time you will not get any delay over here. The same thing if you just take another thing is that what I said that you put some redundant logic.

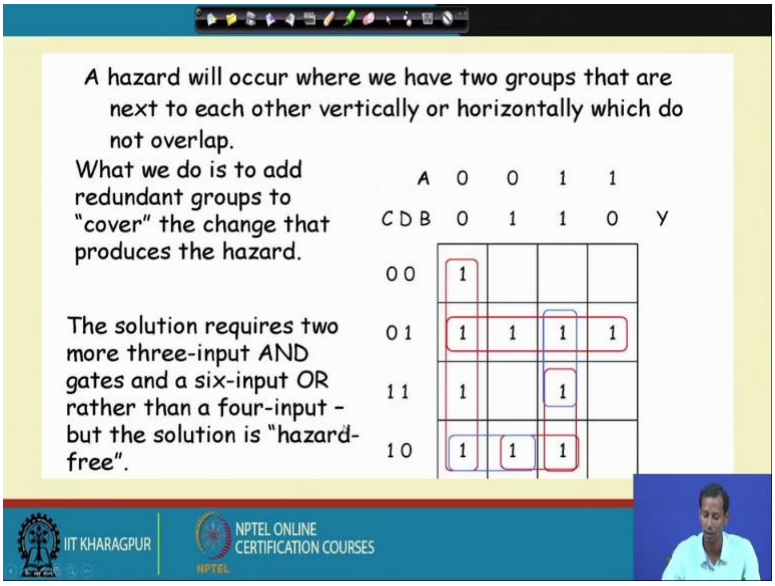
(Refer Slide Time: 17:06)



So, redundant logic means suppose the corresponding Karnaugh map if you just; that means, for the corresponding logic if you just put the expression of Y into a Karnaugh map. So, at that time you will get the expression or you can find out the expression based on this right. So, here you see the glitch is due to the gate delays and should not happen logically. So, does it occur as C changes from 0 to 1? No, if only it will occur whenever this C is basically changing from 1 to 0 or 0 to 1 change there will not be any glitch.

So, why no because as gate 3 turns on before the gate 2 turns off; that means, gate 2 turns off means, what is gate 3? Because, this gate is turning on before of this gate 2 is turn on. Why? That is why what I said as there is no balanced path in between so, that is why it is happening. So, after that what I can do? So, this particular terms; that means, these are these expression is basically creating the hazards.

(Refer Slide Time: 18:36)

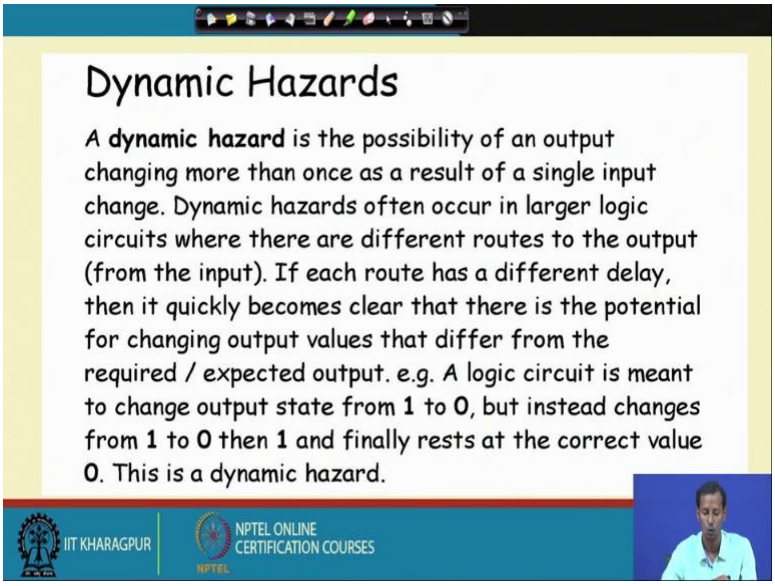


So, for that reason to reduce that what I have to do? I have to take here what is happening so; that means, all the 1's are basically covered. So, that that means, this is the minimal logical expression you can take from the Karnaugh map right but, if you take another extra of this term that means, another extra of this combination. So, if you just take that this particular glitch will be minimized. So, you can take this examples or you can you can just calculate the corresponding expression for that.

And, then you can just simulate in any simulator and then you can check whether you are getting any corresponding error for; that means any glitch or not. So that means, then what is the meaning of that is this kind of things unintentionally it may happen in your particular circuit. And when it will be occurred whenever you will do timing simulation the, for the functional simulation, you will not get this kind of things will not be recognized or that cannot be noticed. Only when you are doing this timing simulation at the time, this kind of this logic this hazards will be noticeable.

So, that is why whenever we are getting violation at; that means, you are getting this some kind of glitch or this hazard. So, at that time by putting these two kinds of things you can remove the hazards. So, one of the way is that what? One of the ways is that you make the path balanced; another thing is that you just add some redundant logic to nullify the corresponding glitch which is occurring.

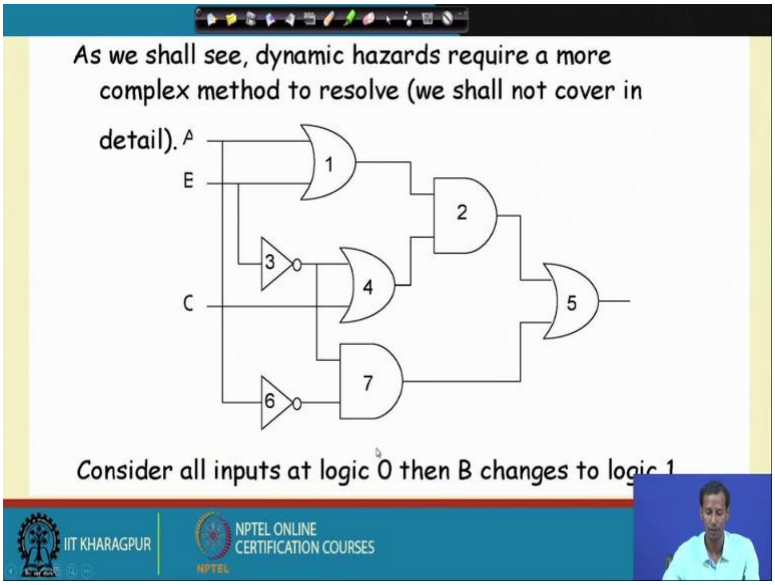
(Refer Slide Time: 20:38)



So, then you this is the; that means, in static hazard we are we can have two hazards: one is static; that means, this static 1 and another one is static 0. Then dynamic hazard, what is dynamic hazard? Dynamic hazard is the possibility of an output changing more than once as a result of a single input change. Dynamic hazard often occur in larger circuit where there are different routes to the outputs.

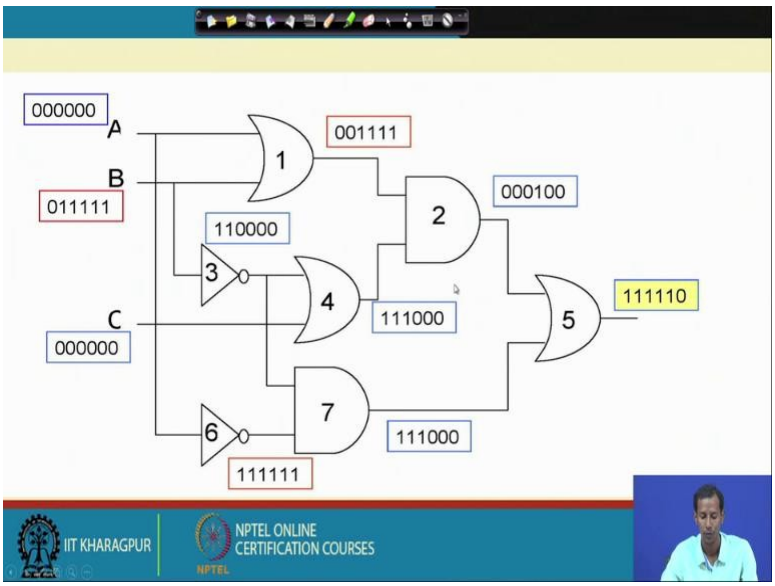
If each route has a different delay, then it quickly becomes clear that there is the potential for changing output values that differ from the required or expected output. As an example, a logic circuit is meant to change output state from 1 to 0, but instead changes from 1 to 0 then 1 and finally, rests at the correct values to 0. This is a dynamic hazard.

(Refer Slide Time: 21:44)



So, then if we take the example of dynamic hazard then you see suppose this is the; that means, there is several paths which are basically going from input to the output. So, here you see now as we shall see dynamic hazard requires a more complex method to resolves. So, consider all inputs at logic 0 then B changes to logic 1.

(Refer Slide Time: 22:21)



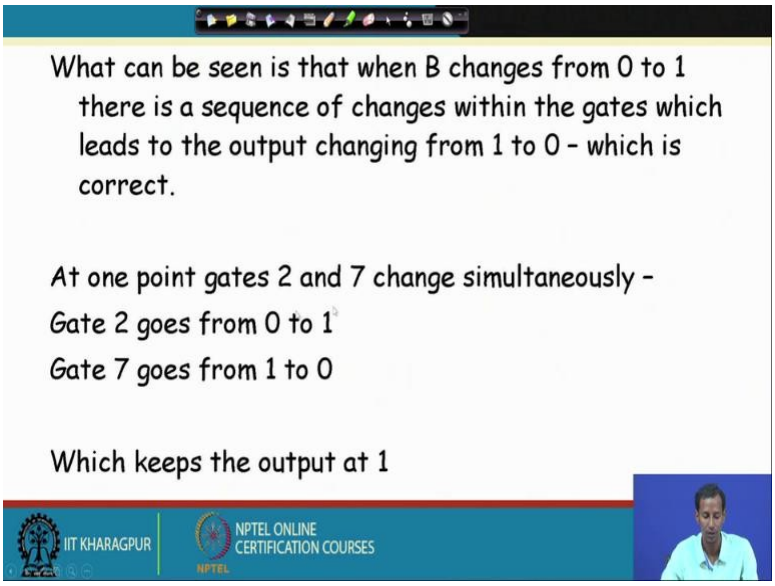
That means all to A C to 0 and B changes, initially A B C all are 0 and then this B changes to logic 1 what will happen. So, initially this all is 0. So, all this 0 means OR is 0; so, this is 1, this is 1 then after that what is happening? So, 1 and then 0, this is 1, this

is 1 over here and 1 over here. So, this is 1, then this is 0, this is 1 0, this is 0, this is 1, this is 1. Now, next what will happen? Now, next what I did B is change to 1. So, B is change to 1 means what? Now, initially this was 1; so, after sometime this will be 1 sorry this will be 0. But for that particular time whatever is the gate delay so, during that time it will be remaining as 1 1.

So, then again if I just see then again it is changing to 1; that means, now 1 is basically continued. So, at the time it is it will become 0, here 0 is continued, here 0 is continued so, 1 will be continued. So, here it will get at this particular point what it will get? It will get 0 0 1. Why 0 0? Because, this gate is also having some delay. So, initially this 0 is passes through 0 0 so, 1 1 here, 1 1 here considering that. So, now whenever these 1 is now continuing so, now it will try to affect this as 0 and this as 1. And, because of this because of this now, this output is basically changing accordingly.

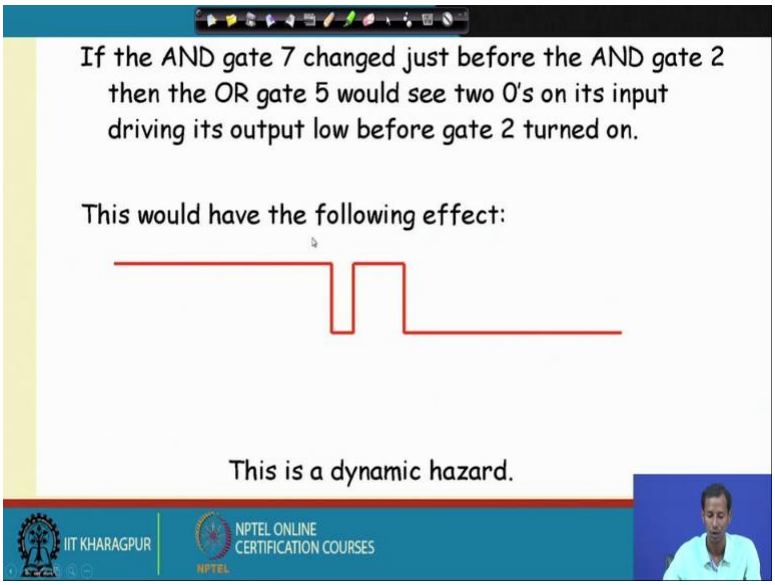
Then next it will become 0 0, this will become 1 1, this all are 1. So, now on based on that 1 1 0 1 1 0 0 0 1 1 1 1. Then again this will be now this will be just try to continued. Whenever these particular terms all; that means, are becoming that means, for 5 clock cycle if I just consider at that time you see though this is this 1 will be continued, but I am getting this as 0. Why? Because, this particular because of this inversion at the B. So, 1 2 3 4 after 4 level this delay that corresponding changes because, of this is tries to reflect at the output.

(Refer Slide Time: 25:39)



So, if you just see what can be seen is that when B changes from 0 to 1 there is a sequence, there is a sequence of changes within the gates which leads to the output changing from 1 to 0 which is correct. At one point gates 2 and 7 changes simultaneously; that means, gate 2 changes from 0 to 1 and gate 7 changes from 1 to 0, which keeps the output at 1.

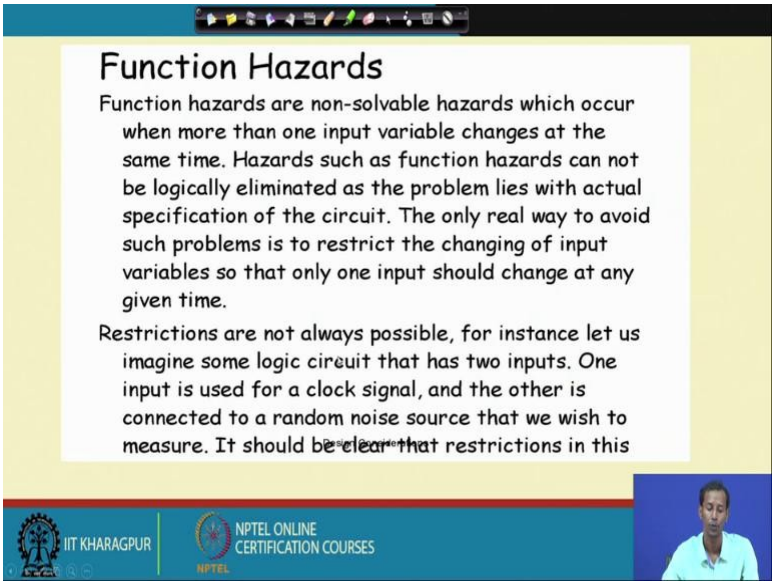
(Refer Slide Time: 26:10)



But, if the AND gate 7 changed just before the AND gate 2 then the OR gate 5 would see a two 0's on its input driving the output low before the gate 2 turned on and this would this would effect the following effect. That means, if you just try to continue at the time I will get 0, then again as there are two inverter in these two particular case. So, based on this we will get the following hazards. It is not that only one glitch I am getting, I am getting the glitch twice because of 2 inverter I have used over here.

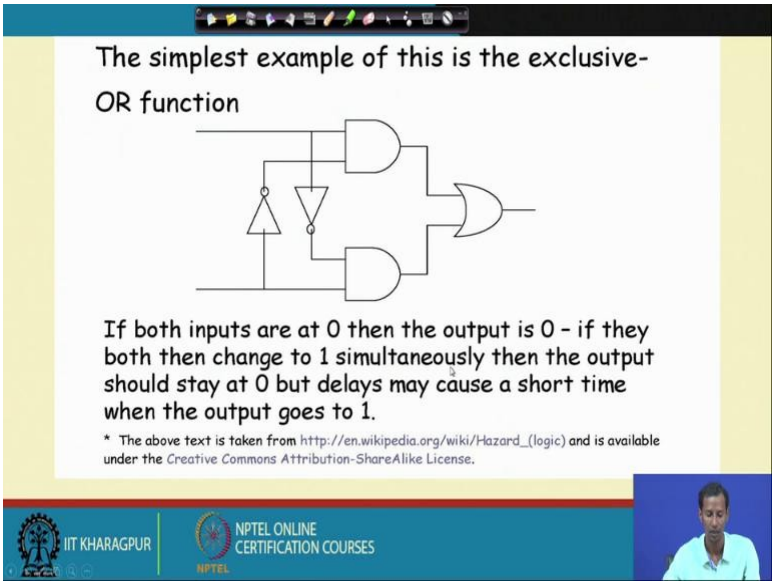
This is normally because of this B as well as for the C; that means, for 3 and 6 both are basically try to changes in this particular circuit and I will get these kind of dynamic hazard. So, in static hazard what is happening that is because of 1 gate, that one glitch I am getting. So, here at the output if multiple paths are basically effecting, at the time I can get multiple glitch and that is why this is known as dynamic hazards.

(Refer Slide Time: 27:31)



So, then the third thing is that function hazards. So, what is function hazards? Function hazards are non solvable hazards which occurs when more than one input variable changes at the same time. Hazards such as function hazards cannot be logically eliminated as the problem lies with the actual specification in the circuit. The only real way to avoid such problem is to restrict the changing of input variables so, that only one input should change at any given time. These restrictions are not always possible, for instance let us imagine some logic circuit that has two inputs. And one input is used for clock signal and the other is connected to a random noise source that we wish to measure.

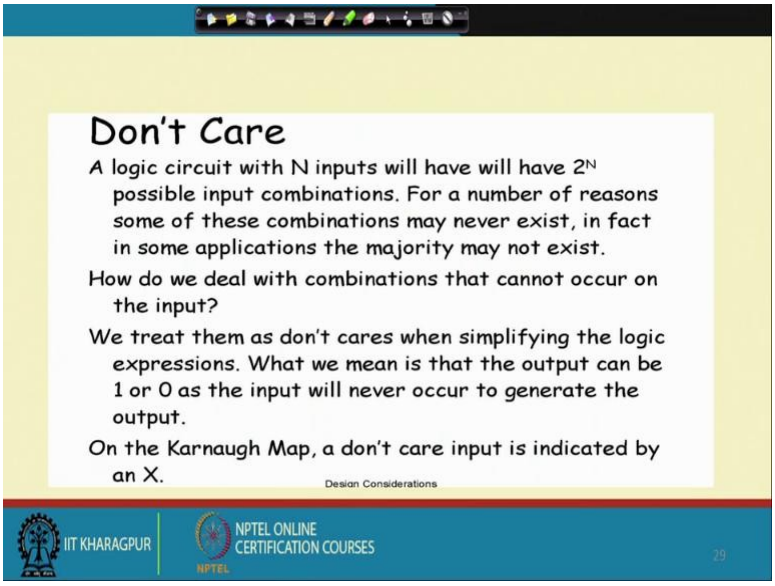
(Refer Slide Time: 28:26)



So, it should be clear that restriction in this case; suppose you consider one of this particular function. So, in both the inputs are at 0 then the output is 0, if both the inputs is 0 at the time what will be. So, 0 means this AND gate whatever, is the other input this is 0, this is 0; so, this is 0. If they both then changes to 1 simultaneously, then the output should stay at 0, but delays may causes short time when the output goes to 1. Means, what? When both are 0 this is 0. If now, from 0 to 1 if I changed so, at the time what will happen, after sometime this will be coming to this and this will be coming to this. So, depend during that particular time this will be so, this is change to 1 and what is this value.

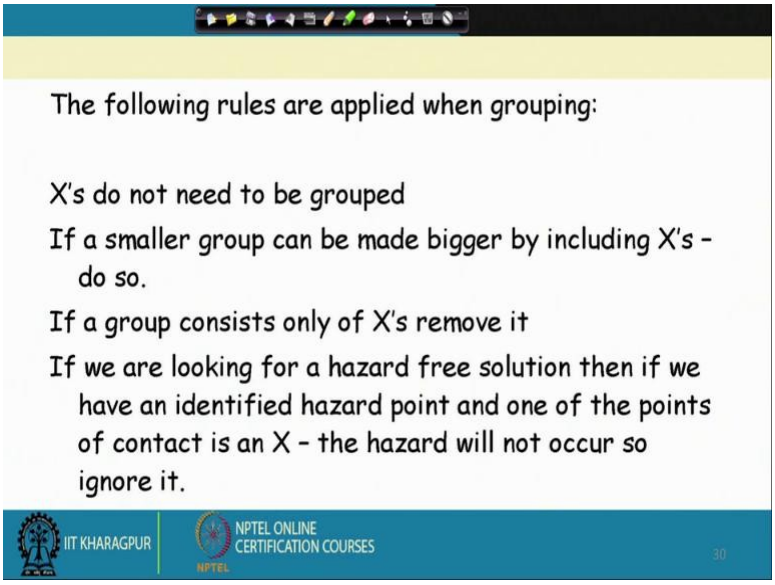
This value was 0 after that after at that time this value was 1, whatever delay I am getting over here. That means, so whenever there is a change this value is now, changed to 1, but for the corresponding delay for these is the delay amount till then it is 1. So, at the time I am getting 1 over here the same thing is happening over here and 1 over here. So, at the time I am getting 1 over here, any of the input if I change to 1 so, this will remain at 1 depending on the corresponding delay of these two inverters. So, this is basically known as the function hazard.

(Refer Slide Time: 03:30)



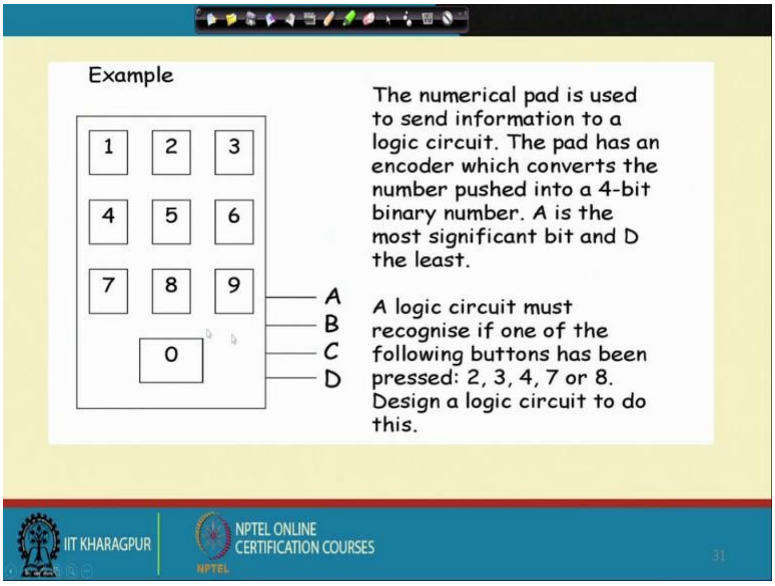
So, how to that means, deal with this? We can deal with this considering that don't care things. So, in Karnaugh map whenever there is no such state available so, at that time we have to consider that as a don't care.

(Refer Slide Time: 30:57)

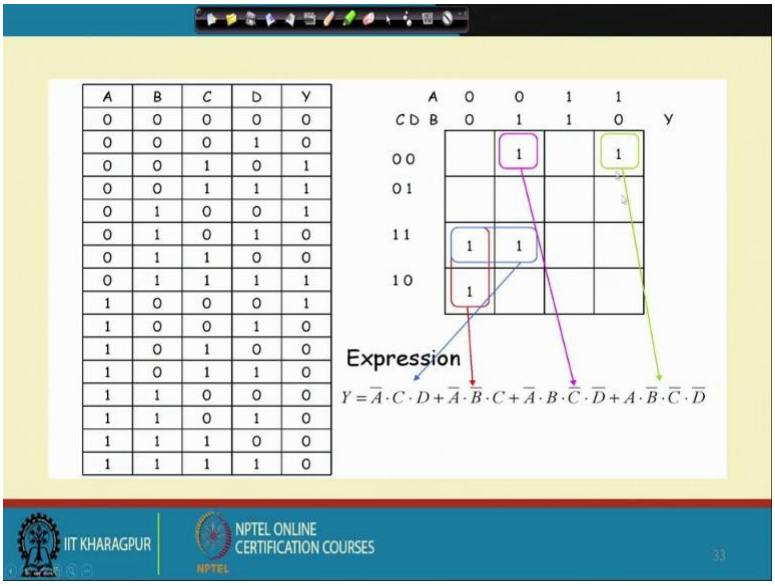


So, if we just take that corresponding position as don't care, at that time this kind of function hazards will not be occurred.

(Refer Slide Time: 31:08)

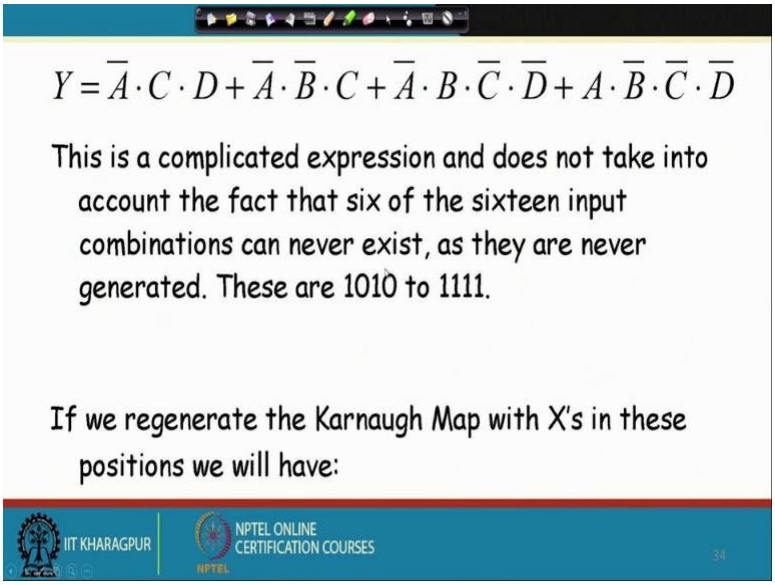


(Refer Slide Time: 31:13)



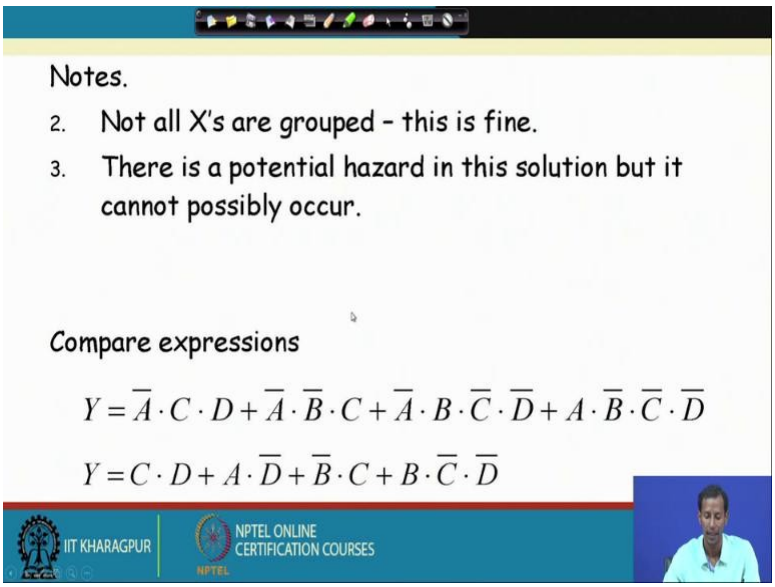
So, suppose one of the examples if I take so, at the time you see that all the that means, this is the corresponding truth table of one particular circuit right. So, now Y 1 5 1; so, 5 1 is represented as something like this.

(Refer Slide Time: 31:36)



So, now for these each of the expression is something like this. So, whenever I am getting this expression so, at that time this is a complicated expression and does not take into account the fact that six of the sixteen input combination can never exist, as they are never generated. These are this 10 to 15. What is that? This 10 to 15, this 10 to 15 they have been never generated. So, if we regenerate the Karnaugh map with don't care in these position, we will have so that means, for this don't care position if we don't just feel with this corresponding don't care; that means, this position if I just don't fill with this don't care. So, at that time the expression will be something like this.

(Refer Slide Time: 32:35)



And, from here you just note that not all X's are grouped this is fine, but there is a potential hazard in this solution, but it cannot possibly occur. So, this two expression if we just compare; that means, if you do not consider this don't cares and if you consider the don't care at the time the two expression is different. So that means, because of considering this inverters over here so, you may get the glitch in this particular expression. So, to remove that you just group with don't care. So, which will that means you just put some extra logic to remove those glitches.

So, this is the that means, end of hazard and this is this setup time violation, hold time violation then this glitch whenever we will actually work on the digital VLSI design or this integrated circuit design. So, at the time this we will get or this we will face in your circuit. So, this is just to give the overview of that if this kind of things happens; so, at the time this is the logic which you can think of to remove this. This is not only the case, suppose if you are getting hold time violation. So, whole time violation how we can remove? You can put some extra buffers to the clock period so, that as you said as you see that if put extra buffers so, at that time you can avoid the clock buffer; that means, the whole time violation.

So, now suppose you are getting setup time violation. So, at that time how you can reduce or how you can remove that setup time violation. So, those kind of that is not at all a part of your optimization; that means, doing the logical optimization I can remove it is not that thing. You can do, but if you are not getting that so, at that time some of the tricks are there, by which you can remove those setup time violation or hold time violation. For a simpler circuit this type of violation or this is not that much important or that will not be seen.

But whenever you are working on a complex circuit design so, at that time this timing analysis is one of the very much important issue because, so many violation you will get. Here, we are not considering any memory based design right. Suppose, memory based design so, if whenever we are suppose fetching the data from the memory and then again storing the data in the memory so, those memory base design again it creates very much problem; very much problem in the sense of getting the violations. In 1 clock cycle you have to write it or you have to write it fast then you can read. So, this kind of things you have to be remembered.

So, there are several that means, tips or techniques are available in digital integrated circuit design. It is not at all possible to cover up all the that means, all of those in these particular course. But, I can just give you or the intension; proper intension of this course is to that there is so, many things this is just to show you the path; that this is the way you can go. Now, you can if you are interested any of this path, you can explore it and it has a very deeper way. So, that is why I cannot in this particular course that that means, the that within the time limit, I cannot go into the deeper in each of this path. I am just giving you the direction in this path.

Now, if you are interested then please let me know, more on to this then we will discuss it discussion forum that we can do or else I will provide you my email ID if you are interested then also you please let me know via my email ID. But, the thing is that whenever you are working in this digital VLSI design, it is one of that very interesting topics or interesting that means, subject not only for this learning. The all the circuits are basically dependent on this digital VLSI circuit design, mostly not all mostly there are dependent on this digital signal processing. Nowadays all the things are basically digitally processed.

So, that is why this course is very much; that means, if you are this particulars direction is very much interesting. And, people are working a lot in these each of this particular path or in each of this direction. And, it is not for one that to get mastery in each of this direction. So, if you are interested for any of the direction, then please go through that. How much help you need from me; obviously, surely I will do that. With that I am just thanking you all for attending the course. So, this is the end of this. This is the last lecture of this course.

Thank you all.