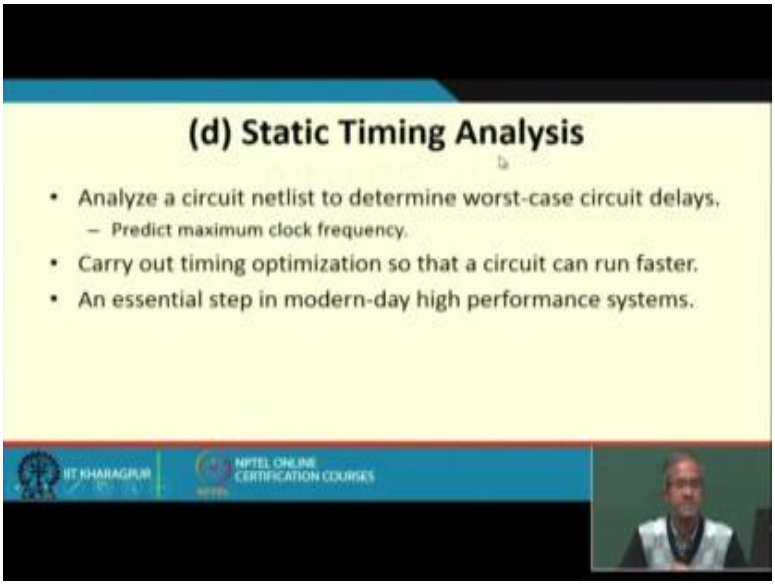
VLSI Physical Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 06
VLSI Physical Design Automation (Part 2)

So, we continue with our this previous lecture, where I talked about some of the VLSI design automation steps. So, in this part 2 we continue our discussion and look at some of the other steps which are also very important in the physical design cycle.

(Refer Slide Time: 00:45)



So, we will start with static timing analysis before going into what static analysis of timing analysis is; let us look at a design from a general perspective. So, when we create a design we have some design specifications to meet, typically designs specifications may mean that I want to run this circuit at a minimum clock frequency of certain things may be you say 600 megahertz, it should be minimum clock frequency this should be the maximum possibility delay between input and output lines. A few such things can be there these are timing related constraints, which has to be satisfied to qualify a design to be satisfiable with respect to the specifications.

Now, what make happen is that once you go though the design process, we arrive at a design which is correct in terms of function functionality, but (Refer Time: 01:58) to find is that delay wise it is not performing as per your expectation or requirement. You see

the cat tools the tools there were using for translation synopsis (Refer Time: 02:12) mentor, graphics whatever you use, those are extremely complex pieces of software and you can never expect that whatever they are generating will always be the best possible or will be able to meet your requirements all the time. So, you need continuously interact with the tools or the systems in order to access on judge the kind of design that you are getting. Now static timing analysis is a very important step in this respect, because of the simple reason is that today we are talking about circuit designs where we have very stringent timing requirements, real time requirements, minimum clock frequency requirements so on.

So, we must be able to analyze the circuit netlist before hand before going into the detailed fabrication steps, to access whether we are going grossly wrong something somewhere. So if so we can take some corrective steps, you can modify your netlist, we can try to make our clocks faster in that way and you can do something using some kind of worst case analysis. So, static timing analysis basically talks about that.

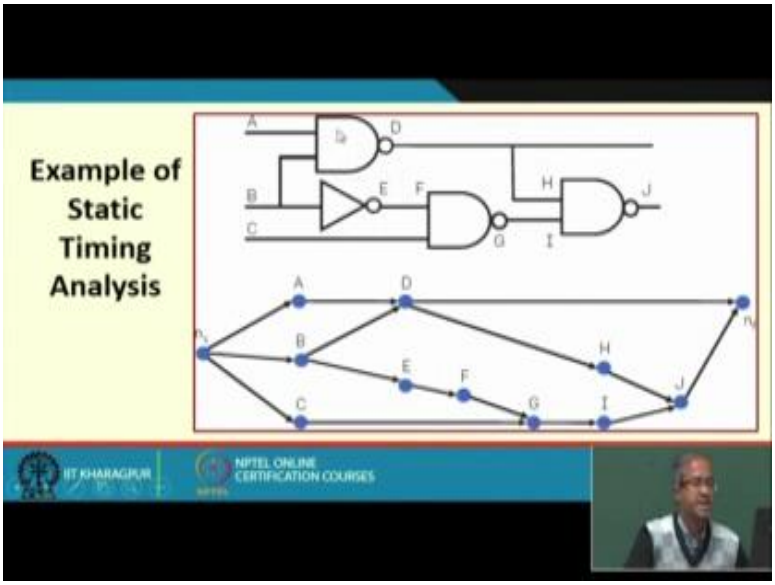
So, this I mentioned it basically tries to analyze a circuit netlist, to determine worst case circuit delays. See here when you calculate worst case circuit delays, you not only consider the delays of the gates as some single numbers; like I can say to simplify things that the delay of this get is 5, the delay of that get is 4, this is 3 and so on according like calculate the total delay, but things are not that simple. So, when gates are fabricated in the VLSI chip using the kind of deep submicron technology that we use today, there can be lot of variations in parameters - like one gate can be having a delay of 5, or the other gate can be having a delay of 4.7 other can have a delay of 5.2. So, there will be lot of delay variations. So, we call it a design slack some minimum value and maximum value.

So, the gate delay has to remain within that, if we find that it is not within that then possibly or circuit will fail; but the slack has to be considered it enough that taking care of the variations during fabrication it should be lying within those regions. Similarly rising time and falling time of the signals; so maximum or the slowest rising and slowest falling times we have to take care in order to determine this slack values, so this is one.

Using this you can estimate the worst is delays and hence you can predict the maximum frequency with which you can feed the clock signal. Not only that you can do some analysis on the circuit and you can suggest some circuit modifications, so that the circuit

can run faster, the clocks can be made faster as I said this is an essential step in modern day systems.

(Refer Slide Time: 05:51)



Now, let us take a simple example here. So, here as you can see here we have a simple combination as circuit comprising of a 4 gates, this A B C are the primary inputs, and this J and this other one D these are the outputs. So, we can model this as a signal flow graph, where we use 2 dummy notes in the input side and the output side and every, you can say every signal line or every gate can be represented by a verdicts.

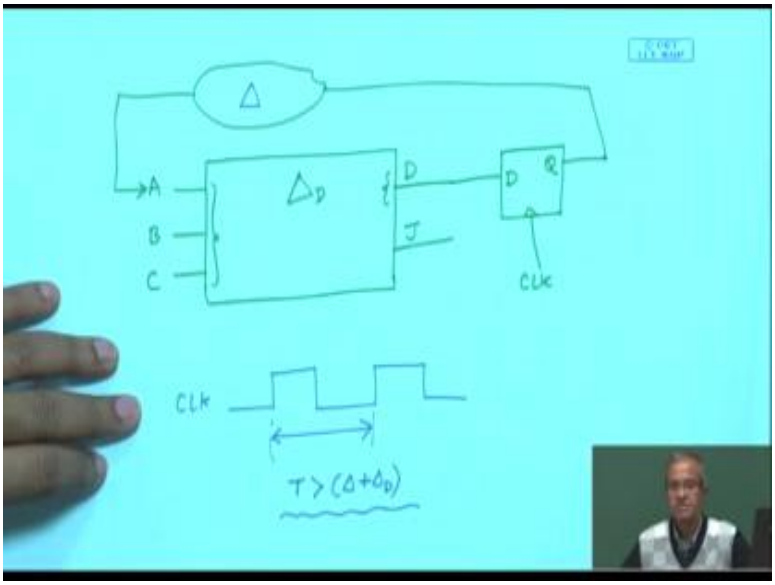
So, every signal line lets represented as a verdicts in this graph and this dummy note here let us call it source elaborate n s and the dummy note here this is the final n f. At the input side we have notes for A B and C here D E, F you see this E and F apparently their represents this same line they are electrically there equivalent, but suppose I want to model this interconnection may be these are long interconnection line. So, I use E and F at 2 separate notes, similarly G and I, D and H, H J G I like this and in this graph I am not shown here each of this edge can have a weight, this weight might indicate either the delay of the gate, delay of the interconnection or both.

Like for example, when you go from A to D, you go through this interconnection as well as the delay of this gate then you reach D, but when you reach go from E to F it is just interconnection, G to I just interconnection. So, if you model this signal flow graph, one

thing when you model this signal flow graph you have to have some realistic values for these weights.

Now, once you have this then you can have some kind of a shortest path or the longest path algorithms running through this graph, so that you can find out what is the longest delay between n s and n f the longest delay.

(Refer Slide Time: 08:34)

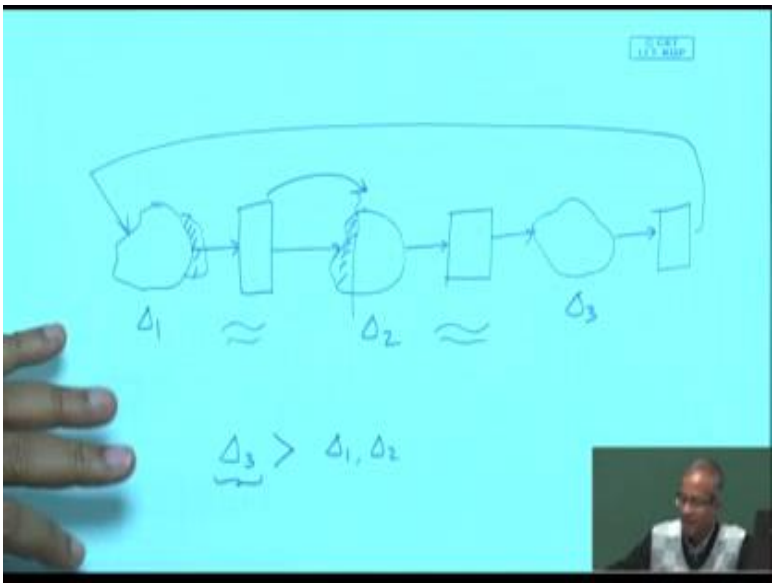


Now, suppose this entire circuit I have in a scenario like this. So, I am showing this circuit as a box black box. So, A B C are the inputs right. So, D and J are the 2 outputs, if suppose these are fed to some storage element let say flip pops these are clock. So, this output can be going to some other circuit element which intern is feeding A let say.

Now, through timing analysis what I can find out that in this circuit the maximum delay from the input to the output D is let us say delta D, then I can also see I can evaluate that this circuit module which is available here it is the delta is delta. So, the clock signal that you are applying if you are look at the clock period T, T obliviously must be greater than delta plus delta D, this is the hard timing constrain that this circuit like this will be having. So, if you do a static timing analysis of each of the circuit blocks between the flip pops then you can have a very fear estimation of the worst case timing delay between clock pulses, so that you can know at least to a fear degree of accuracy that how much or what is the maximum frequency of the clock that you can apply right.

So, basically this is the main purpose of static timing analysis, but there are many other issues that we shall be discussing when we looking to this topic in more detail later on, there are the issues like determining determination of false path adjusting a circuit like.

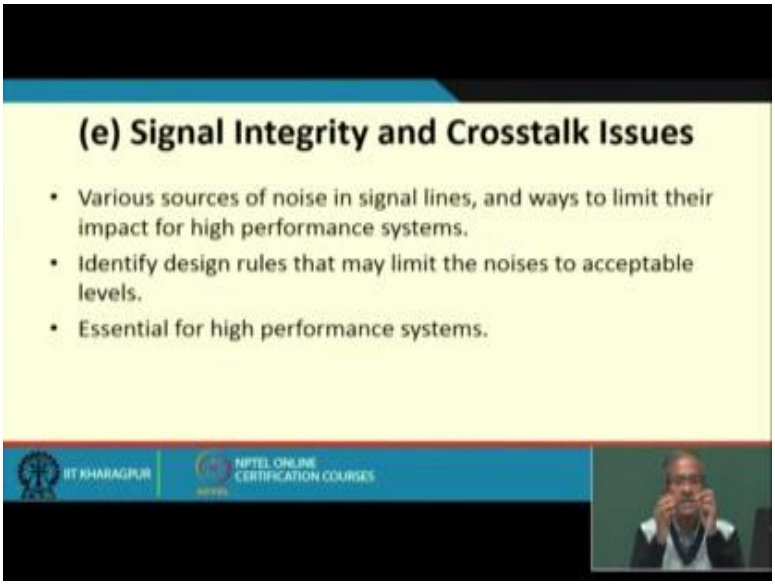
(Refer Slide Time: 10:58)



For example let me also tell you another thing, you can have a scenario like this we have some combinations circuit then a flip pop feeding like this then a combination circuit then a flip pop may be you have something like this, may be something your feeding back like here.

Now, it is possible to move say this combination can consist of a number of gates, it is possible to bring some part of this circuit along here what the reverse and move this flip pop a little ahead in the change, because what we trying to do these delays of this circuits delta 1, delta 2, delta 3 if this is the miss match, suppose delta 3 is significantly greater than delta 1 or delta 2 then delta three will be the dominating delay which will be determining the clock frequency maximum clock frequency. But you we if you can balance this out by moving this flip pops forward and backward in the change, such that this deltas can become approximately equal then you can carry out some optimization with respective to the clock frequency.

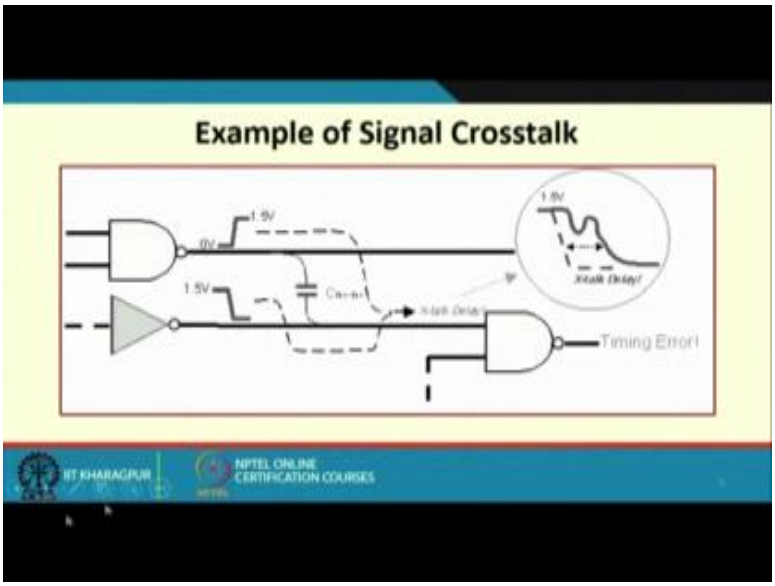
(Refer Slide Time: 12:19)



So, all this things we shall be discussing later; and another important issue is considering signal integrity and crosstalk in circuits. Now you see in a submicron VLSI chip, the gaps between the 2 interconnection words are very very less; due to fabrication error they can even come closure. So, because of that if there is a signal transition on one line that transition can get coupled to the other line and if are not careful enough this can lead to an error in the calculation or computation which those signal lines are leading to. So, signal integrity means, the signal what is suppose to be, but because of crosstalk this signal is getting frequency polluted or there is some error introduced there in. So, you should be able to as a should be able to model analyze this kind of signal integrity and crosstalk issues and come up with a set of approaches, which if you follow would expected to overcome or miss eliminate this kind of problems in practical circuits.

So, typically designed rules are used which are quite conservative, but the advantages that if you follow this design rules, your final design is expected to be free from this kind of problems. Design rule means you can say that the minimum separation of the line should be this. So, in all the lines that were laying out you make sure that the minimum separation should be this. In (Refer Time: 14:14) this is just as an example and also another example can be you can also limit the maximum length of the lines, because longer an interconnection line the distributed restive and capacitive effects can be coming, which can lead to not only signal delays, but also signal degradation. The signal if you just feeding to some other circuits, because of degradation there can be errors means a one might be recognize as a 0 something like that, ok.

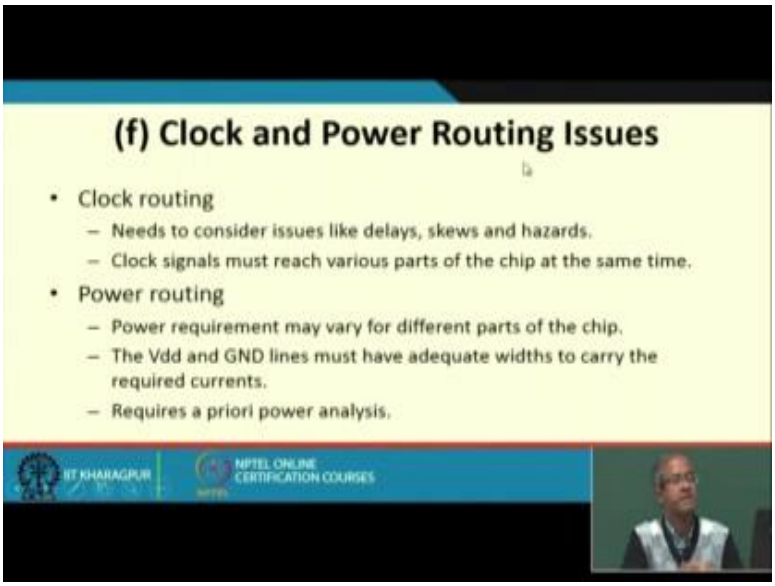
(Refer Slide Time: 14:45)



So, let us take an example here. So, here you show some gates where some signal transitions are taking place of the output of these gates. The second gate transition has happening this gate this transiting a little later than this transition, because these 2 interconnection lines are running parallel to each other and quite close let us say there will be a capacity coupling. Because of the capacity of coupling, so what might happen is that in this signal line second one your expectative have a clean transition like this, but because of this rising sudden spike here, you can encounter a sudden noise signal injected here like this; and if this noise signal is sufficiently high in amplitude and this signal is feeding some other gate, so it might temporarily recognize it as a 0 and then again 1 like that.

o, that might lead to a timing error and some error in calculation. So, this is just a very simple example I have given, but here we shall be looking at much more delay related issues and other problems there in, our next modules where we discussed these things in detail.

(Refer Slide Time: 16:11)



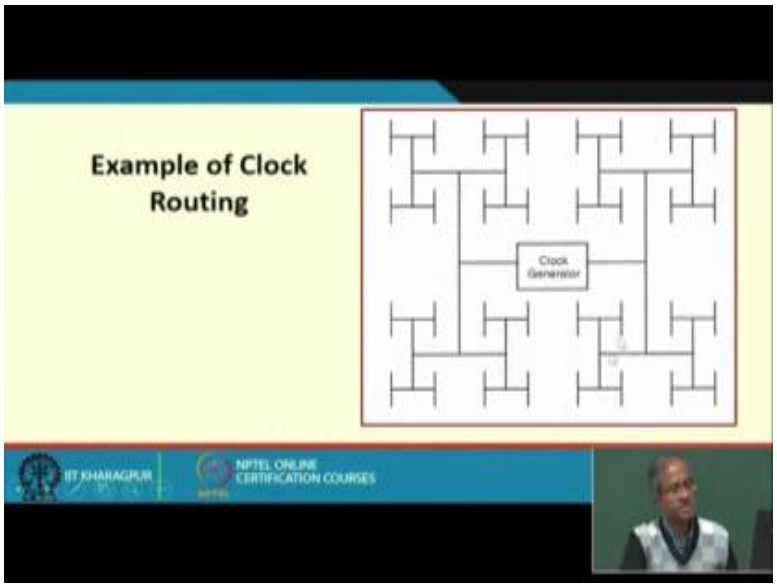
This clock I have just mentioned earlier this clock and power routing requires some special attention, because when you have the clock signals it is not just the issue of just allowing the signals to be distributed to the different points were should be. There are lot of other clock related parameters that need to be satisfied for correct operation of this storage elements; rise time, fall time, setup time, whole time there are lot of such delay related parameters which needs to be looked at very carefully and if your design is good those delays will also remain within limits. You have to take all of them into account, so as to come up with the realistic value of clock frequency, which is expected to run this circuit in a correct way.

So, for clock routing you need to consider issues like delays, skews and hazards; earlier the example I took we are talking about clocks skews, clock skew means same clock is same clock is reaching 2 different points in 2 different times, but may using that H tree kind of an approach by making all the interconnection lengths equal, the interconnection delays are also expected to be equal. So, we are expecting to reduce this cube to a great extent. So, this is so for a clock is concerned, but for power routing with Vdd and ground, this also is a very important problem. Because one thing is that we should not want to mix the power signals along with the normal signals that carry data, because data variations must not couple into the power signals and generate power supply variations that may not be acceptable. So, usually the power signals the Vdd and ground, they run on separate dedicated metal layers. So, interference and crosstalk are minimum there.

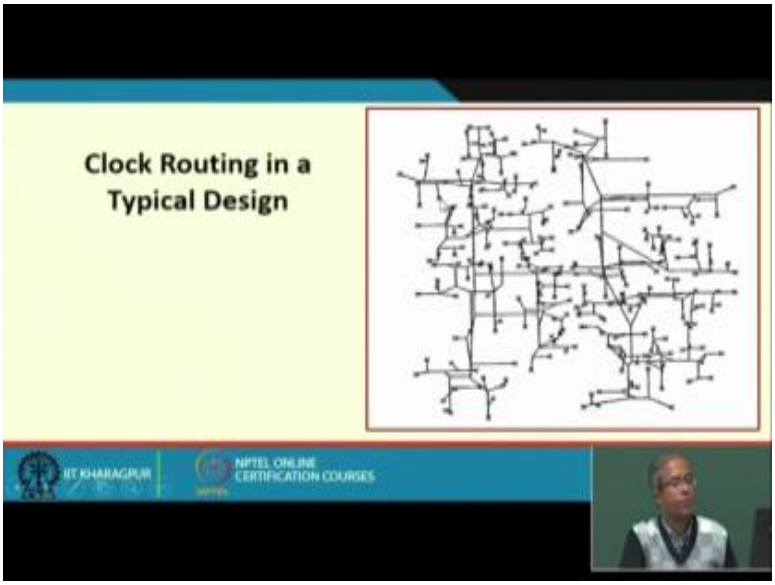
Not only that because of variations in power requirements in different parts of the chip in some places the lines may be wider in some places the lines may be require to be much less wider. So, the width of the lines is not fixed unlike the signal lines, somewhere where a lot of current is expletive flow the line can be wider; later on the lines can become thinner and thinner and thinner like that as the power signals are getting distributed in power supply.

So, this will be require a priori power and this is to be done to decide on the exact widths of this lines ok.

(Refer Slide Time: 19:18)

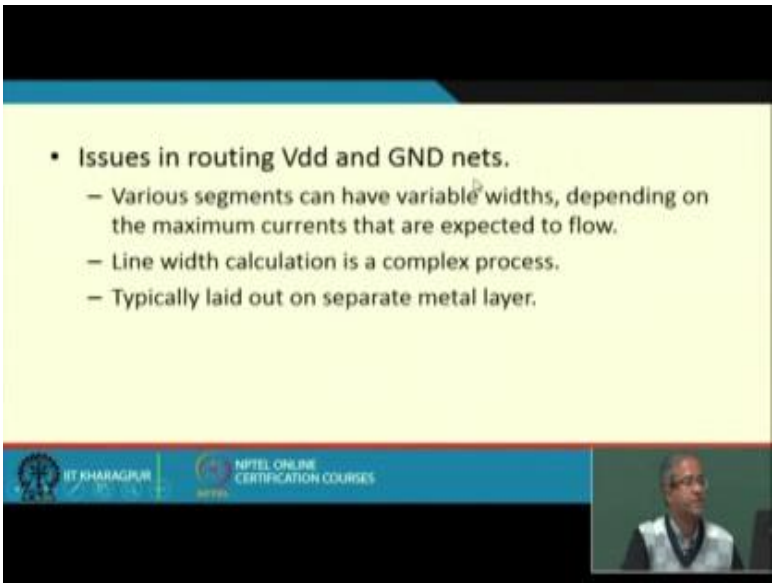


So, this is the same diagram we saw earlier clock generator, but as I said in a real clock distribution scenario, the clock distribution points may not be so uniformly located.
(Refer Slide Time: 19:40)



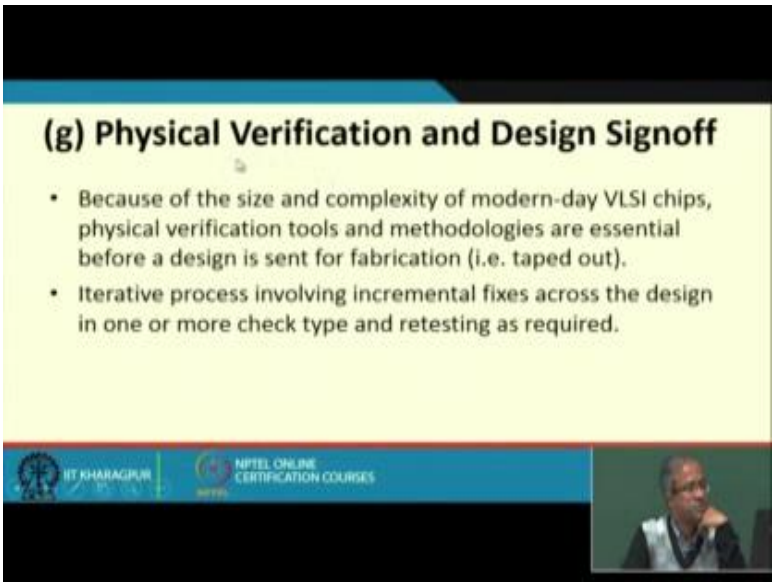
So, we can have a scenario like where this small process are the points where I need to distribute this signal and may be the clock is generated somewhere centrally. So, there are lot of I mean sharing need to do a lot of complex analysis, in order to determine various parameters, so as to reduce the clock skew in these cases. So, here we shall be looking at some algorithms on this later on.

(Refer Slide Time: 20:10)



And related to Vdd and ground net as I said various segments can have a variable width, line width calculation is required which is complex and they are typically laid out on a separate metal layer.

(Refer Slide Time: 20:27)



Now, the last thing that that we shall be discussing with respect to VLSI physical design, is the process called physical verification that is follow up by design sign off.

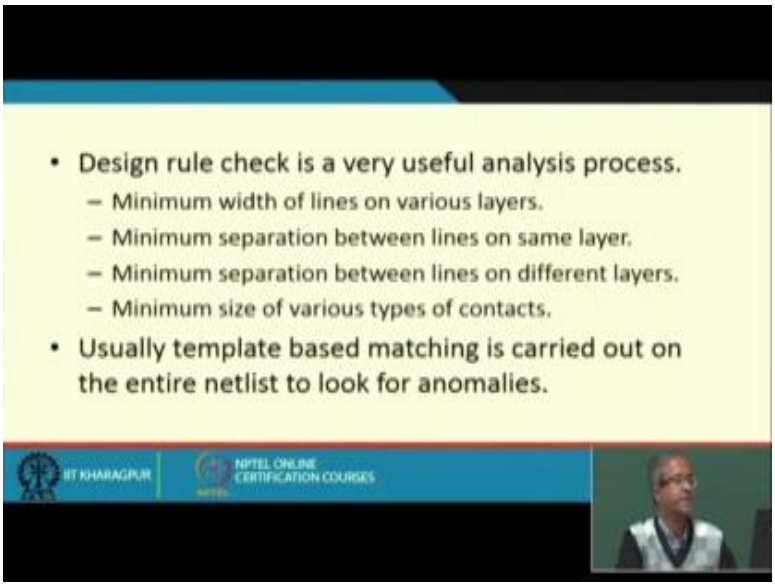
You see what is physical verification? As the name implies we are trying to verify something through physical inspection. Now see we have gone through all these steps of placement, routing etcetera we have arrived at a layout. And once you arrived at a layout well, those inspection you cannot see with your eyes it has to be done is an automated way using an extreme means say use using a microscopic mechanism, you have to look into the layout. You can look into the layout and you can find out anomalies there in, as I said you can have some design rules like 2 wires running parallel to each other must have a minimum separation of this. Now I refine this a little bit, if the 2 wires are running on the same layer separation should be this, if the 2 wires are running on to different layers separation must be this and so on.

Similarly, minimum widths of the layers; if it is metal what should be the minimum width? If it is polysilicon what should be the width; diffusion what should be the width; if it is a contact connection what should be the minimum dimension? These are some design rules which are defined apriori and when you are creating the layout, there are lot of optimization steps that are carried out, it might so, happen that due to some bug in one of the steps, some of the design rules a getting violated. So, instead of letting those violations remain with your layout, it is always the case that you do a physical design

verification to do some layout extraction, verify the layout overall to look for some anomalies in those design rules and only if an your satisfy with that then only you can go for the so called design sign of where you can actually send your this layout data for fabrication. As I said this layout data is usually send in some standard format for fabrication in the foundry, GDS to is one such very popular format which is used. So, here as you can see as I just no upset because of the size and complexity of the designs, physical verification and methodologies are essential.

See sending a design for fabrication sometimes we call it as taping out; taped out is a term which is used very popularly in the design houses and this physical verification is not a very well defined process that is why it may require a number of iterations. These are more like some adopt checks optimizations we are trying to carry out, some incremental fixes. So, you in order to fix one error may be some other error might get injected, you have to do it continuously. So, and this iterative process which may need rechecking and retesting if required as an when required. So, this is the last process that you need to be done and you should be satisfied about it before you tape out your design for fabrication.

(Refer Slide Time: 24:31)

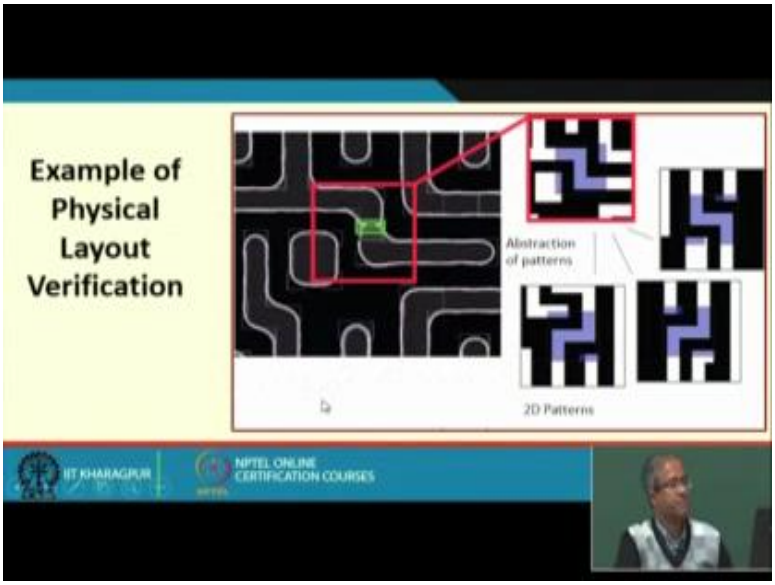


So, as I said design will check is one process which is very important. Minimum width, minimum separation same lay different layers minimum size of contact, all this have to be specified. Now these separations and these widths these are typically specified in

terms of the basic feature size we refer to as lambda. So, as technologies skills down say from say 0.25 micron to 0.18 micron to 0.12 micron to today's same it is around 22 nano meter and beyond, they design rules keep changing, but all the rules are in terms of the basic feature size lambda, for 0.2 to micron point for 0.022 micron 22 nano meter technology, were lambda will be 0.022 micron or 22 nano meter.

So, in terms of that lambda that parameter lambda you define the design rules and you try to verify the designs like that and what you do is usually some kind of template based matching, you defined a lot of templates because just design rule checking maybe difficult, you try to match those templates against the entire netlist by sliding a window kind of a thing.

(Refer Slide Time: 26:03)



That I showing an example that here you have a window shown by this red box; this red box has a something like this say zigzag kind of a connection, where some minimum width is expected; but due to some fabrication error the width of this line has been reduced. So, this is our design rule violation.

So, you can have some inspection in the layout and try to find out whether such design violations do exist or not. So, this patterns there can be different abstraction of different kinds of patterns, you move it and see that whether what you see under this window is matching with any one of these. This is typically what is done and this trusts me that this is the very time consuming process, because you have to look through the entire layout it

is a quite time consuming process, but it make sense because it is only after this will be sending out your design for fabrication, which is again very expensive you cannot afford to have any error before that.

(Refer Slide Time: 27:20)

So, once you are satisfied with this you can finally, send the design for fabrication taping out and you get your final chip fabricated after that. So, what you have seen is that we have looked at a very quick review from a very high level a birds of a view you can say about the different steps that are required in the physical design process, which we shall be dealing with in much more detail in the successive modules. And another thing also that you will be discussing about is some special consideration about low power design, because low power design because low power designing will also extremely important now a days, because most of the computing gadgets that you use now a days are operated on battery other than the desktops one or offices, laptops, mobiles, tablets they all operate on batteries. So, reducing the power consumption is of paramount important. So, that is one thing that also we shall be discussing, and this way come to the end of this lecture 6.

Thank you.