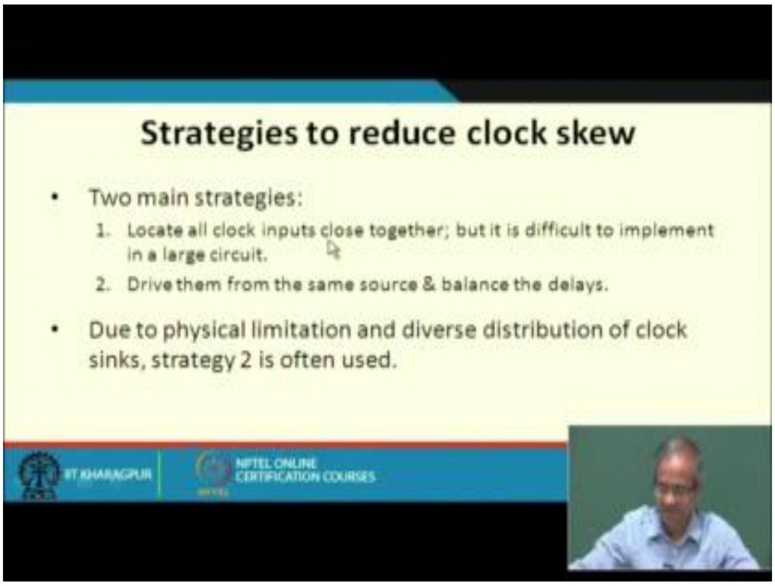
VLSI Physical Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 28
Clock Network Synthesis (Part 2)

So, we continue with our discussion. So, in this lecture we shall be talking about some general strategies used to control the clock skew because ultimately our problem if you recall we said that from a clock pin, we have to lay out a clock net which will be feeding the clock signal to all the terminal points, and we have to control this skew in this overall network. So, across any 2 terminal points, the maximum difference in the arrival time of the clock signal or the clock edges should be kept within a limit, right.

(Refer Slide Time: 01:00)

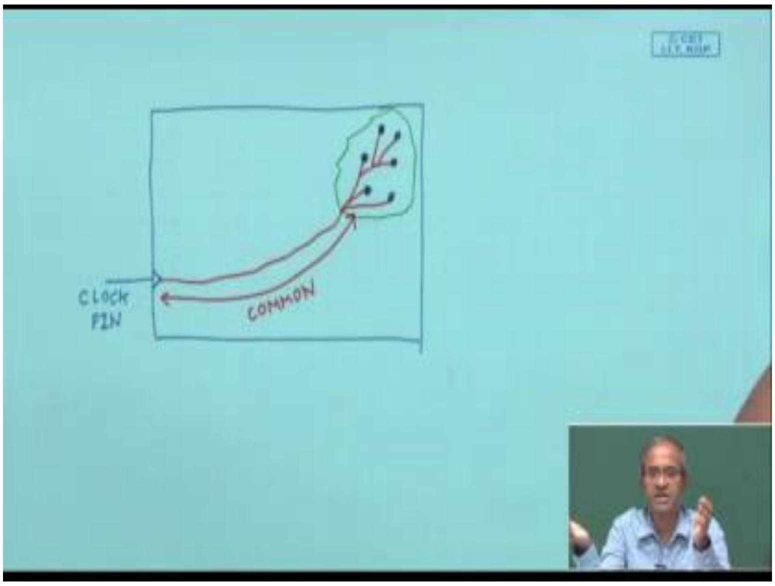


Strategies to reduce clock skew

- Two main strategies:
  1. Locate all clock inputs close together; but it is difficult to implement in a large circuit.
  2. Drive them from the same source & balance the delays.
- Due to physical limitation and diverse distribution of clock sinks, strategy 2 is often used.

So, first we look at the broad strategies to reduce the clocks skew. So, let me try to explain what are these 2 main strategies are. The first strategy says locate all clock inputs close together. So, what does this mean? So, let me just illustrate.

(Refer Slide Time: 01:25)



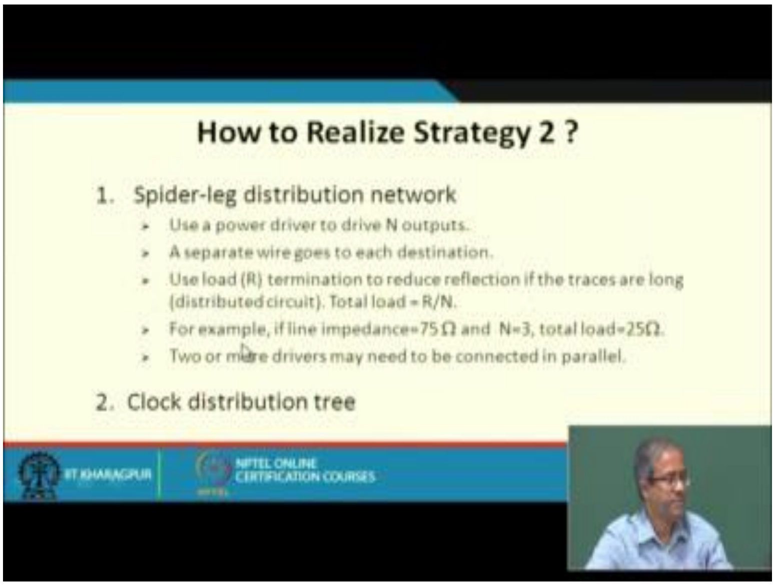
Suppose I have a chip, let us assume that all my clock means clock input the places where I have to feed the clock signal, they are all located close together let say like this and let say this is my so called clock pin, from where the clock signal is entering my chip. So, whatever strategy I use to carry this clock signal up to this, say I can use some connections like this. Now one thing is clear that there is a major part of this net this part is common and here in this locality there is some variation 2 wire lengths are little different.

So, overall this skew will be limited to the variability of the wires in this local region. Now if this clock pins are constrained to be located within a small region, then this maximum variation that we are talking about that variation can also be controlled. So, this strategy means although it looks simple, but practically I not easy to incorporate because in a general chip you will be needing clocks almost everywhere, you cannot assume that clocks will only be located or concentrated in a small geometrical area. So, this was the first strategy as I had said that this strategy is difficult to implement in general.

The second strategy is that, well, let us not try to bring them close together, but let us try to balance the delays. So, the same clock pin is feeding the signal, but let us route the wires in such a way that the delays are getting balanced in some way right. So, assume as

I had said because of difficulty in implementing the first strategy, we typically use the second strategy. So, let us concentrate on the second strategy first.

(Refer Slide Time: 04:07)

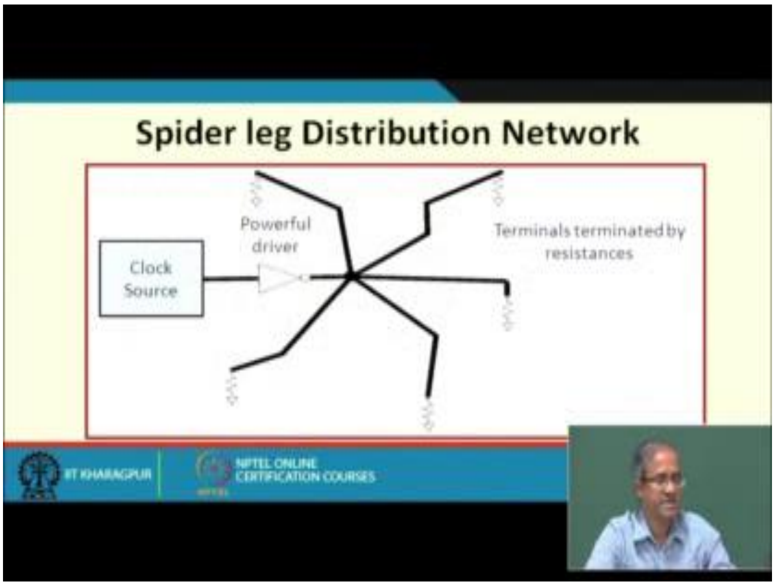


How to Realize Strategy 2 ?

1. Spider-leg distribution network
   - Use a power driver to drive N outputs.
   - A separate wire goes to each destination.
   - Use load (R) termination to reduce reflection if the traces are long (distributed circuit). Total load = R/N.
   - For example, if line impedance=75 Ω and N=3, total load=25Ω.
   - Two or more drivers may need to be connected in parallel.
2. Clock distribution tree

Now, this second strategy again broadly if you think we can try to do a couple of things. The first means approach maybe like I have a certain number of pins where I have to send the clock signal. So, I try to draw independent lines to each of the clock signals with some calculations such that the delays are approximately equal or I try to layout a general kind of a network, later on I will try to connect the pins to the network in some way. The general network will be easier to lay, it will be more generic in the sense that clocks will be available almost all throughout the chip and then the exact pin location wherever they are you try to connect to the nearest clock point something like that, we will go into the detail of this later.

(Refer Slide Time: 05:39)



Spider leg Distribution Network

Clock Source
Powerful driver
Terminals terminated by resistances
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, the first alternative that I talked about here, this is sometimes called spider leg distribution well. Spider leg distribution has a number of characteristic features, I shall be explaining this well first let see, how are spider leg distribution network looks like; it looks like something like this, you see that here you have a clock source this might be coming from outside the chip, you have a powerful driver because this driver should be able to supply current to all these branches, the number of branches can be large, right.

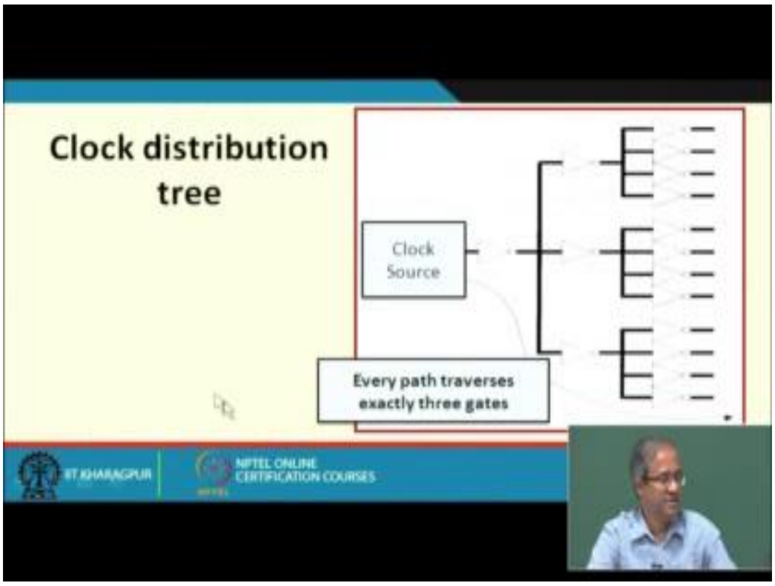
So, these are the final points where the clock signals are required, the clock pins I call them. And because they are located arbitrarily these arms will look like legs of the spider, they are typically and another thing this clock signal one thing let me tell you because of this skew and jitter constraints and requirements, this clock signal as also later on we will see the power and ground signals, they are laid out on the same layer as compared to signal routing, where you know either 2 layer 3 layer this kind of routing is done. So, whenever we switch from one layer to another layer, there is a wire connection. But in clock whenever you have a wire connection that will incur additional delays; so those delays we are trying to minimize right. So, in clock or power we do not use such wires normally, to means we can use via only when required not otherwise and we try to lay the wires on the same layer as much as possible.

So, the rest I am talking about this. So, this driver has to be powerful enough. So, when I say I a powerful driver there are several ways to create a powerful driver we shall talk about this a little later, so going back. So, we use a drive let us say N points there are N clock pins, we are driving all these N pins. So, a separate wire will be going from the clock source to each of the destinations, right. Now one thing now for those who are familiar with transmission lines will be knowing that whenever you are driving a long transmission line, there are some effects like reflection and other problems because of which we normally have to use a terminating resistance at the end of the line. See earlier when you laid out the network cables for those of you who have seen those thin and thick Ethernet cables. So, at the end there was typically a 75 ohm or 300 ohm termination resistance which was connected, that is required with respect to transmission line requirements.

So, here also so it will be something like this these legs can be treated as transmission lines. So, at the each of the end of this transmission line you can you have to use a resistance for termination right. So, these are additional overheads, but if you do this, so the total load resistance R, there are N outputs see this resistance is R, let say R R R R see all these resistances are appearing in parallel because these are parallel connection. So, the effective load that this driver will be driving will be this; will be R divide by N. So, if your R is 75 ohm and you are driving 3 points then an effective load will be 75 divided by 3 25 ohm; and this is a point I shall be coming again later that for higher power driving capability, you often may require to connect 2 or more drivers in parallel. So, this is one strategy you can use this.

But obviously, you can understand there are some drawbacks you have to use or terminate every clock pin with a resistance, which will incur additional area additional overhead, which may not always be feasible. So, although this method is an alternative where you can lay out the clock to different pins and again another point to balance skew you have to ensure that all the legs of the spiders are approximately equal. So, for that purpose you may have to route it in a special way, you have to make a detour sometime it is called snaking like a snake lies like this, instead of going through straight you make a line longer deliberately to match it with another line, such things you may have to do.

(Refer Slide Time: 10:45)



Clock distribution tree

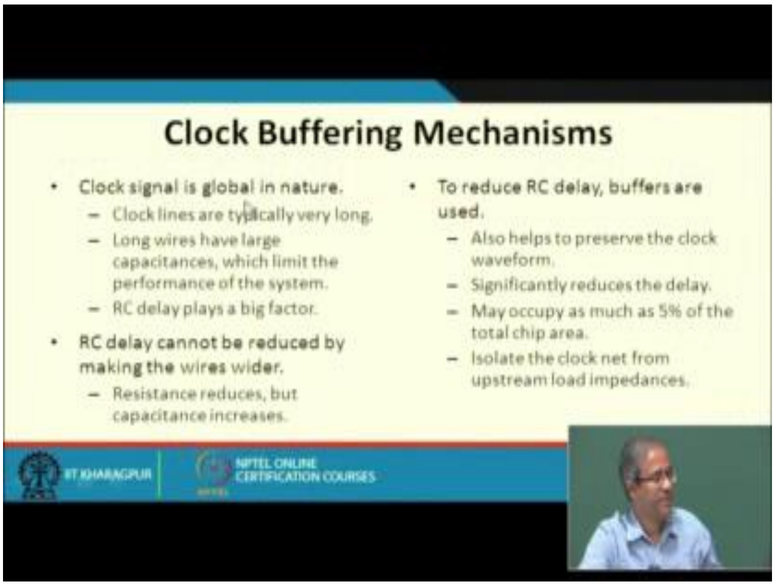
Every path traverses exactly three gates

So, the second alternative is the clock distribution tree. So, clock distribution tree looks something like this. So, here you see instead of a single driver I have several drivers, these are the drivers and these drivers are placed along the branches of a tree. So, I am laying out the clock network like a tree. Now assume that there are 12 terminal points as shown in this diagram. So, starting from a clock source I am first going to 3 clock source with respect to this driver, then the each of this driver is driving 4 lines and each of this driver I am showing one it may be driving again 4, I am just showing one line.

So, now in this way the advantage that you gain is of 2 types; first you do not need very large drivers, you need small drivers and all the drivers can be of very similar dimension and size. Secondly, if you do this you see the main delay of the signals will be delay of the drivers; because maybe these inter connection delay will be shorter, so the delay of this inter connection will be much shorter than the delay of the driver, but also we shall be trying to balance layout the wires in such a way that the wire lengths will be approximately equal. So, from the clock source to the clock terminals, you need exactly 3 drivers to be traversed.

So, the idea is that you are laying out your clock network in such a way that automatically the delay from the clock source and the clock terminal points become approximately equal. The number of drivers you are traversing are equal in number and also for one driver to then next the way you layout, it has to be some kind of a regular layout the wire length should be all equal. So, that delay will also be approximately equal right. So, this is the basic principle behind clock distribution tree.

(Refer Slide Time: 13:11)



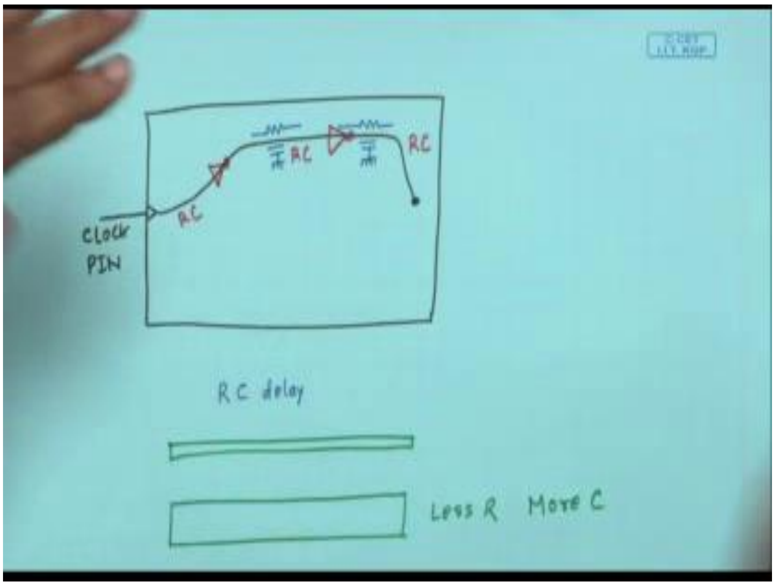
Clock Buffering Mechanisms

- Clock signal is global in nature.
  - Clock lines are typically very long.
  - Long wires have large capacitances, which limit the performance of the system.
  - RC delay plays a big factor.
- RC delay cannot be reduced by making the wires wider.
  - Resistance reduces, but capacitance increases.
- To reduce RC delay, buffers are used.
  - Also helps to preserve the clock waveform.
  - Significantly reduces the delay.
  - May occupy as much as 5% of the total chip area.
  - Isolate the clock net from upstream load impedances.

Ok. So, now another thing we use the clock buffers here I am showing. So, why do we you need the clock buffers well. Clock buffers one obvious reason I told you earlier that you need to drive more current, maybe the clock source cannot drive the current to a 100 or 1000 points. So, I need buffers to amplify the current, but there are other reasons also.

First thing is that clock signal is global in nature, clock lines are typically very long from one end of the chip to the other it might go and if there are long lines, there will be distributed capacitances and resistances and there will be RC delay. So, what I mean to say is that suppose I have a chip like this. So, I have a clock pin out here, from this clock pin I am laying out a long clock wire let us say to a pin out here.

(Refer Slide Time: 14:03)



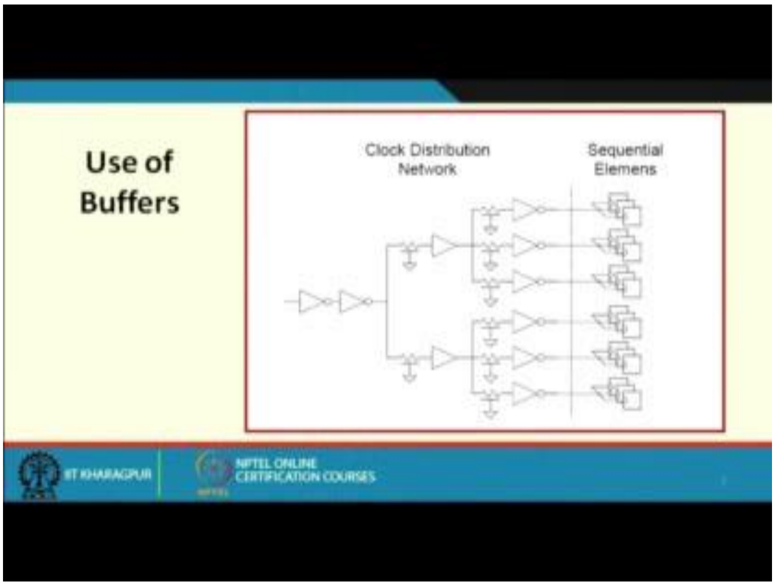
Now, because of transmission line factors means along this line, there will be resistive effects there will be capacitive effects, not in one place all throughout. So, longer the line longer will be the values of RC, and as you know for any RC kind of a network we refer to as it as RC delay. So, if we can either reduce R or reduce C, the delay will be reduced right. But it we have a long wire both this R and C will be more, but in contrast if we do like this instead of single long wire, if we break it up into smaller pieces by inserting buffers then you effective RC delay of each of these segments will be much less and the contribution for a long wire it increases much more rapidly as compared to shorter wire. So, this sum of the delays of these shorter segments will be much less then the delay of this long segment if you have a single segment. So, the idea is that if we use buffers to break a long interconnection into shorter interconnections, your overall delay will be less that is one reason.

And well you can argue that well I can reduce my RC delay by making the wires wider; like see I am talking about the RC right resistance and capacitance. So, suppose I am laying a wire like this I am showing it as a rectangle. So, as an alternative I could have made it wider. So, if I make it wider it will mean less resistance, but it will also mean increase in the capacitance, because a if the wire is wider, then the capacitance to another layer will also be more right; because the wider wire will be having lot more coupling than another layer; see the value of the capacitance if you recall the formula is proportional to the area of the plate, and if you make the wires wider the area of the plate is increasing. So, the capacitance is increasing. So, just by making the wires narrower or wider will not help you, will have to include or introduce buffers in between, right.

So, the net conclusion that we have is that we have to include buffers to reduce RC delay of course, but there is another advantage it helps to preserve the clock waveform; what is the meaning of this? So, whenever a signal a digital signal we are talking about 0 1, it traverses over long distance. So, again due to this RC effect, the nature of the signal gets distorted; the more the length of the communication line, the distortion will be more. So, if I introduce a buffer after some time, so the output of the buffer this signal will again become distortion free. So, we are controlling the maximum amount of signal noise or distortion, that might go in buffers also help in that.

Delay will be increased, but the point to note is that in terms of the total area, there can be a large number of buffers are introduced and it can occupy as much as 5 percent of the total area right. And this buffers help in isolating the clock net from upstream load impedances. So, what does this point mean? Suppose I have a buffer, which is directly driving let say 100 clock loads, so the total load that I driving is those 100 loads. So, each of the loads will be having some capacitance value, they will all get added up 100 c, but if we use several distributed way then each driver will not be driving 100, but maybe only up to 4 or 5. So, the load I driving also gets limited and hence it will be fast.

(Refer Slide Time: 19:00)

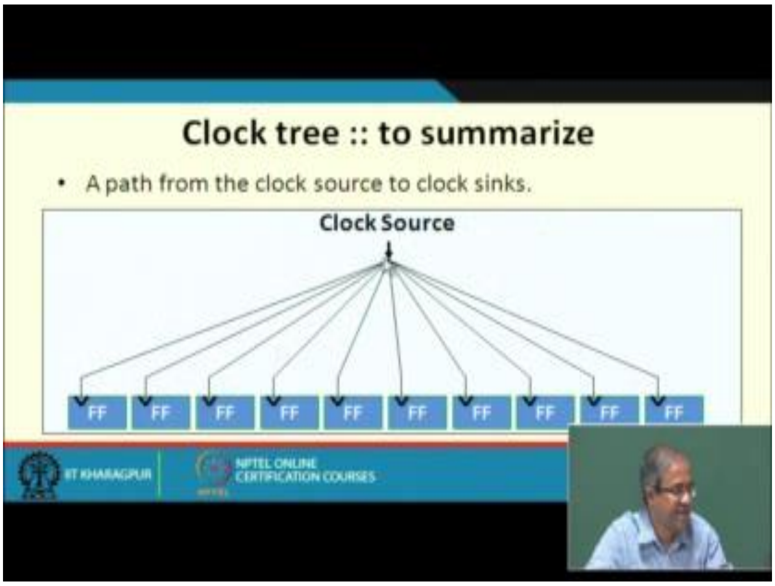


Use of Buffers

Clock Distribution Network
Sequential Elemens
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

So, this diagram actually summarizes whatever I have said diagrammatically, that if you distribute the buffers along the branches of the clock tree. So, along each branch there will be a resistive and capacitive effects shown like this, but this will restricted to shorter wire segments.

(Refer Slide Time: 19:36)

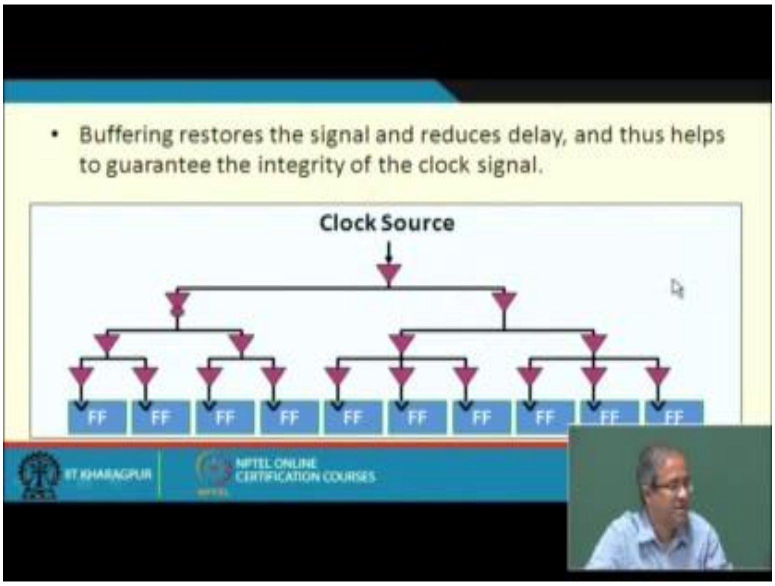


Clock tree :: to summarize

- A path from the clock source to clock sinks.

So, the RC delays for each of the segments will be much less and finally, your drive the actual circuits, where clocks are required sequential elements. So, to summarize you need to send a clock source to a number of flip flops that can be spread all across the chip these are called clock sinks. So, we do not send or route this nets by connecting them through long wires, rather we introduce buffers in the form of a tree like this, and then we these buffers can be used to drive these flip flops directly right; this is what we talked about so far.

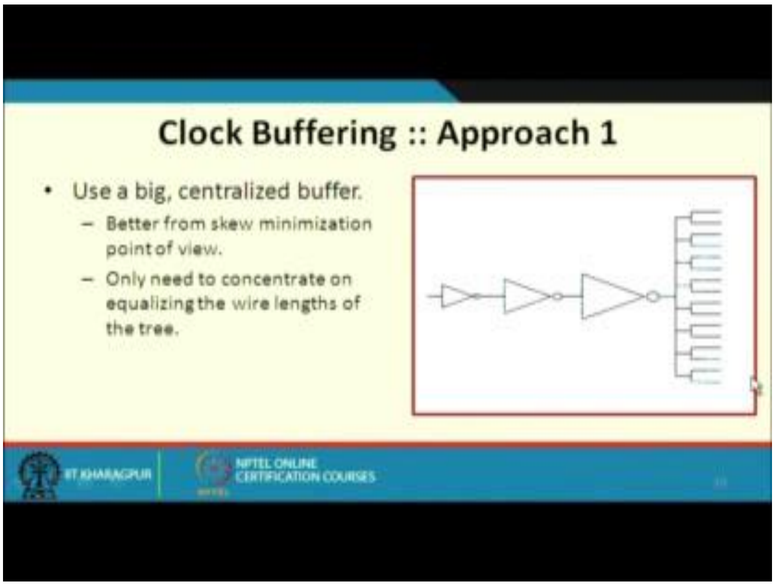
(Refer Slide Time: 20:00)



- Buffering restores the signal and reduces delay, and thus helps to guarantee the integrity of the clock signal.

Now, with respect to clock buffering let see the broad approaches like. Suppose we are not using buffers in the branches of the tree, we are trying to design a single very big buffer which will be powerful enough to drive all the clock sinks; it will be having sufficient cluck clock driving capability.

(Refer Slide Time: 20:19)

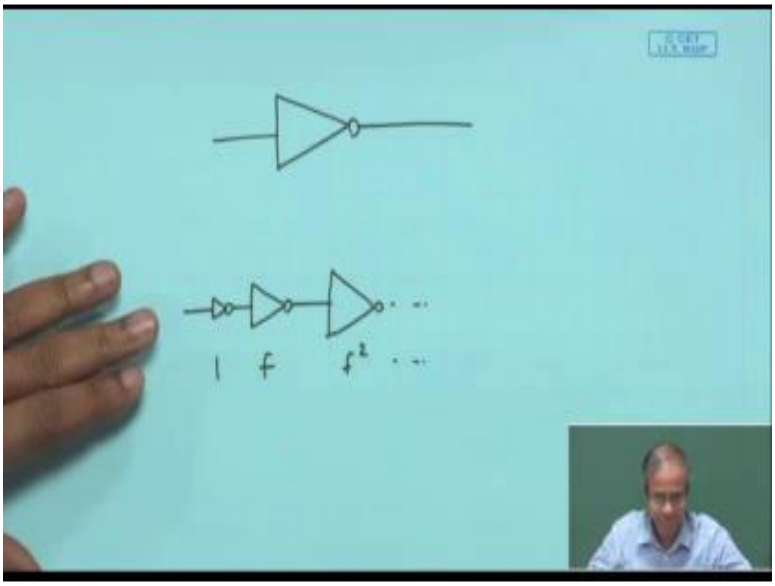


Clock Buffering :: Approach 1

- Use a big, centralized buffer.
  - Better from skew minimization point of view.
  - Only need to concentrate on equalizing the wire lengths of the tree.

Now, there is some known results from CMOS technology again, when you design such buffers; if you design a single very large buffer that is always possible, but that buffer will require very large dimension transistors.

(Refer Slide Time: 21:30)



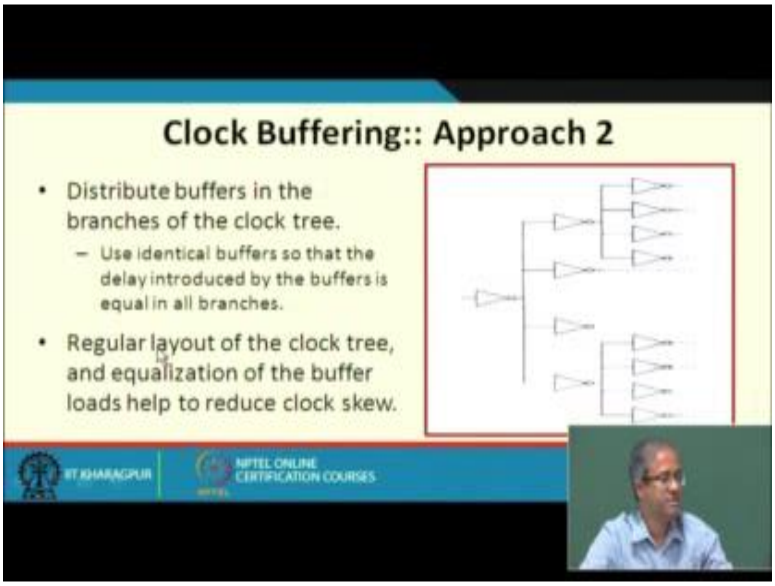
Now, when you make the transistors larger the resistance of the capacitance value of the transistors will increase that may increase the switching delay of the transistors, the on 0 to 1 or 1 to 1 delay or 1 to 0s delays. So, because of that a general design guideline is that instead of using a single large buffer, you use something like this.

Firstly, use a very small buffer, then you use a slightly longer buffer, then you use a slightly bigger buffer like this. So, each buffer is larger than the other by a factor of f. So, if it is 1 size of this will be f, this will be f square and so on, this is the standard practice it has been shown that if you can select f in a proper way the total delay in this method will be quite optimized, as compared to a single buffer kind of a thing.

So, in this diagram, so although we are seeing big centralized buffer, that centralized buffer we are showing as a chain of progressively larger buffer. So, as I have said the reason I have mentioned. But this single centralized buffer is powerful enough to drive all the sinks. So, if you can layout the wires in such a way that the lengths are all approximately the same then skew minimization will take place.

Now this approach if you can incorporate or if you can implement, this will give you better skew minimization because your buffer is in one place; only wires need to be matched in terms of the lengths and you can see the parasitics R n c, but you are not worried about buffer you are only worried about equalizing the lengths. But if you distribute the buffers in many place many small, small buffers there will also be small variability in the buffers, 2 buffers can never be identical when they are fabricated. There will be small difference in their parameters right so some variations will also go in there. So, here as I had said we need to concentrate only on equalizing the wire lengths of the tree.

(Refer Slide Time: 23:44)

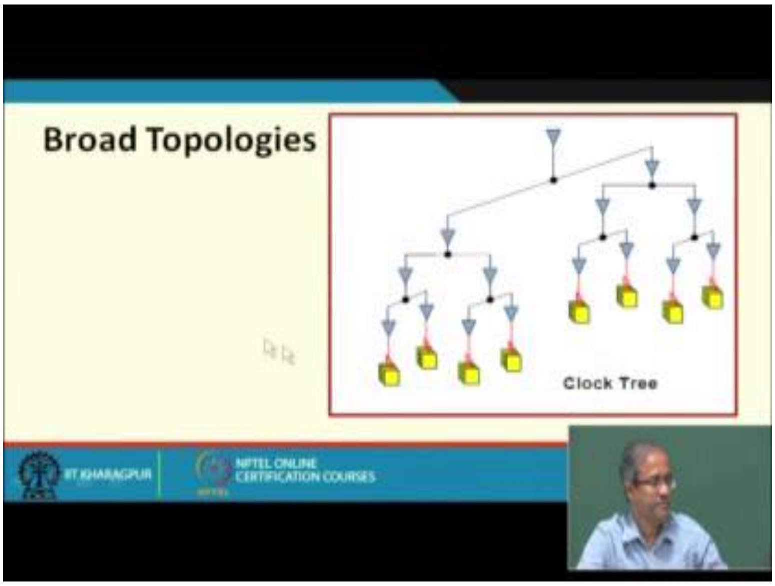


Clock Buffering:: Approach 2

- Distribute buffers in the branches of the clock tree.
  - Use identical buffers so that the delay introduced by the buffers is equal in all branches.
- Regular layout of the clock tree, and equalization of the buffer loads help to reduce clock skew.

So, approach 2 where you are introducing buffers along the edges of the tree. So, we are trying to use identical buffers like say here I am assuming that every buffer will be driving 4 other buffers. So, their sizing their current driving capability will be that way identical. So, if we can ensure this then you can ensure that the delay along all the branches will be equal, because all the buffers will be having equal delays and if we can layout this kind of a network, we shall see later how. We can have a very regular layout of the clock tree, which can ensure the equalization of not only the buffer delays, but also the delays of the interconnections. So, that way you can ensure your overall delay will be means equalized and skew will be minimized right.

(Refer Slide Time: 24:48)

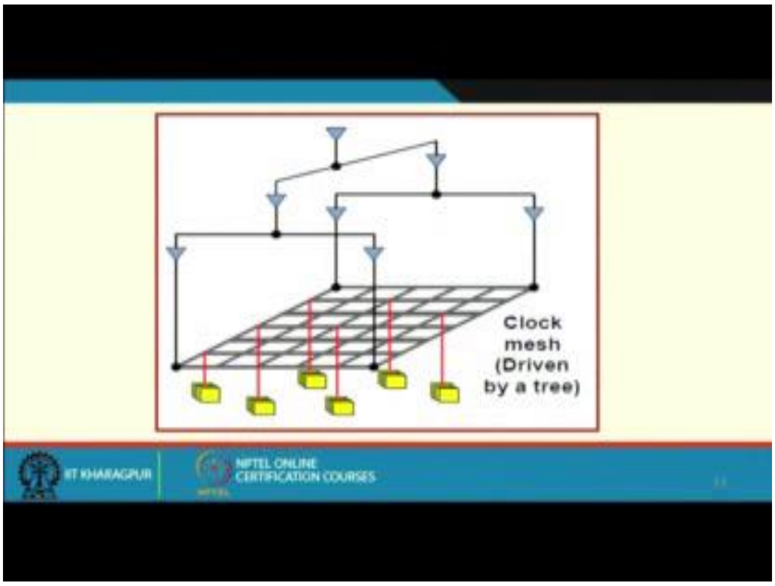


Broad Topologies

Now, let us come to the broad topologies; you see so whatever we talked about was somewhat philosophical in nature, I said that there are many clock sinks either use a single clock buffer, distribute them in a nice way, such that q is balanced or during distribution instead of a single large buffer, use small, small buffers, but put them along every branch of the tree. Now in a general chip you think it can be a processor chip, it can be a system on chip and not only processor there is processor, there is memory there are other function units there are so many different kind of circuits in the chip. So, the requirements will not be the same for every part of the chip. So, how do you distribute the clock? So, sub broad approaches we are talking about now one by one.

These are called the topologies the clock topologies; the first topology we already talked about this is the clock tree. So, just to refresh your memory we are basically laying out a tree like structure, it can be a binary tree it can be a tree of higher dimension for example, every buffer can drive 4 buffers so here I am showing 2. So, the last buffer will be driving, driving a set of clock pins, clock terminals right. This is how clock tree can be laid out, but here in this topology we are assuming that the last level of buffers wherever we have put them, the clock terminal should be closer to those. So, the last level of buffers should be close to the clock terminals, so that you can connect them by shorter wires and those wires also should also be approximately equal in length. But again this is possible if your layout is some sort of regular, your circuit is regular. So, from each of this terminals of the clock tree, you can layout equal wires to connect the clock signals this is one.

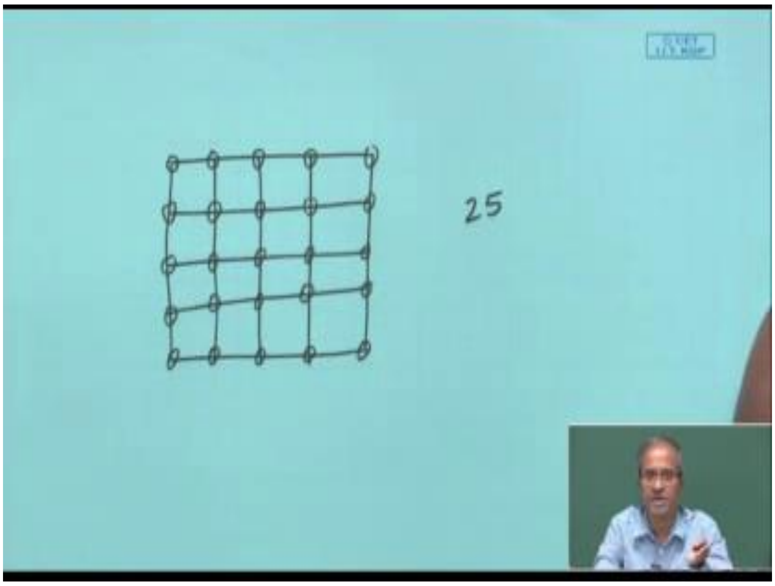
(Refer Slide Time: 27:06)



The second one is more general and many processor chips actually use this kind of a strategy. So, what they suggest is that like we shall again come back to this later when we talk about some case study, this is some kind of a 2 level distribution; means in the top level distribution we use a clock tree as you can see, this is also a binary tree I am showing it can be some other tree of higher dimensions.

Now, you have a binary tree like structure like normally what we said earlier, that we have a clock tree the clock tree is finally, feeding all the pins of the storage cells. So, storage cells wherever we require, but now we are saying no, storage cells you forget for the time being.

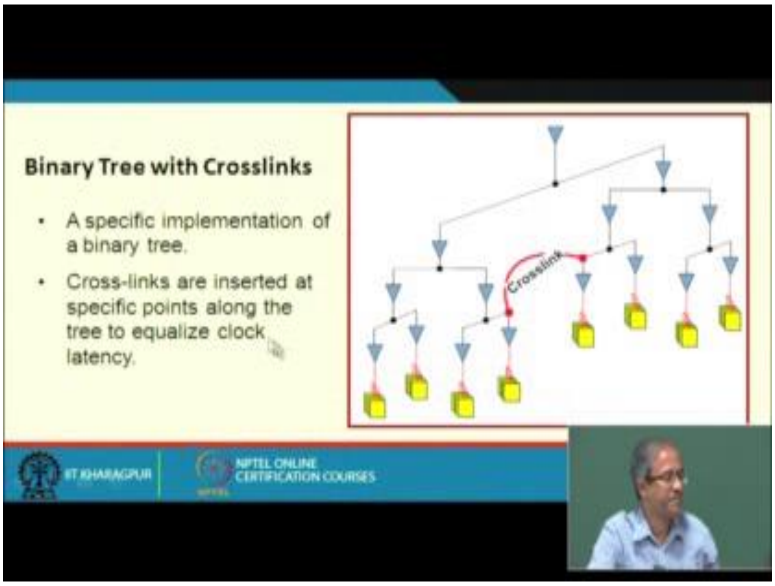
(Refer Slide Time: 28:03)



For the time being we assume that there is a grid that we create all along our chip, this is this is a metallic gird connection they are all shorted together; this solid lines are wires they are all connected together and the tree that I create the tree is on top, that clock tree is on top at every terminal is not driving or clock terminal, but driving one point of the grid. So, with respect to this diagram maybe one point is driving this point, one point is driving this. So, each of these grid terminals that we have so the clock tree is driving these terminals all of them; which means I have some kind of a parallel driver connection there can be for example, in this case 5 by 5 there can be 25 such drivers, driving 25 pins in this grids.

So, the current driving capability will be 25 times more, and once we have this grid on a metal layer, now at the lower level where you have the circuits wherever you need the clock you draw it from the nearest grid point. You are separating the clock tree from the grid, clock tree will be regular in nature they will be feeding the grid, they will be parallelly driving the grid and at the lower level where you have the circuits you tap the clock from the nearest grid point; this is the principle behind the so called clock net mesh driven by a tree.

(Refer Slide Time: 29:50)

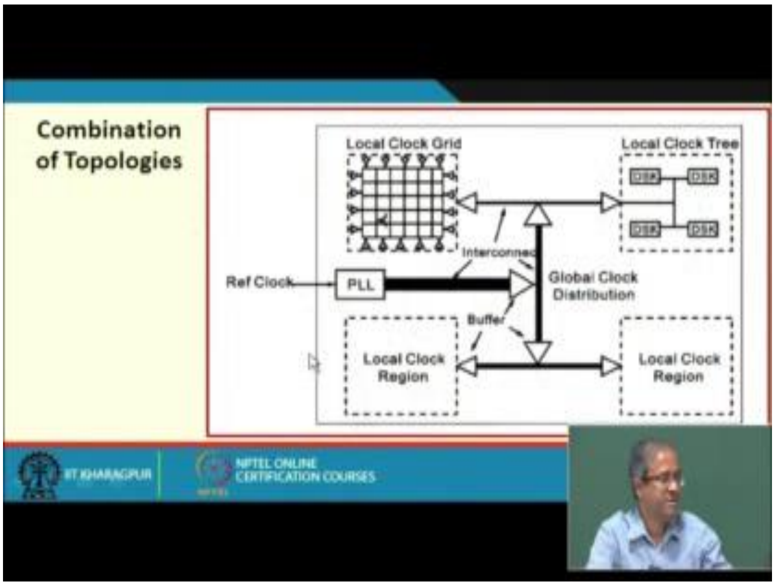


Binary Tree with Crosslinks

- A specific implementation of a binary tree.
- Cross-links are inserted at specific points along the tree to equalize clock latency.

And the third alternative we have again a binary tree, but for delay equalizations sometimes you can add some cord links across sub tress, like this red link is a cross link sometimes you do this to equalize the clock delay, maybe in some part of it because of longer wires are longer RC delays the delay is more; so to reduce delay you can connect 2 branches together so that they come in parallel and the delays become less right.

(Refer Slide Time: 30:28)



Combination of Topologies

So, in general you can combine all these topologies together like you can have the; this is a chip, you can have the clock pin which is feeding, well in many clock circuitry you have a PLL inside it phase lock look kind of a circuit which will be synchronizing the clock signal internally, and from this you can distribute it globally to different regions and each of the regions you can have a grid structure that I have talked about. So, you can have a tree that will be feeding a grid. So, you can have a clock tree, but grid is not required very few connections are required. So, you can customize your clock tree design like this.

So, with this we come to the end of this lecture. So, we continue with our discussion on various tree distribution strategies in our next lecture.

Thank you.