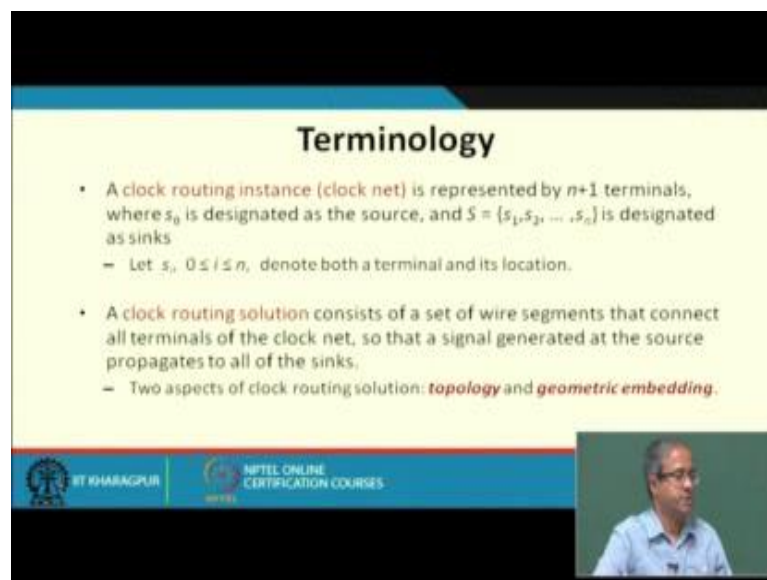


VLSI Physical Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 29
Clock Network Synthesis (Part 3)



We continue with our discussion on clocking and clock distribution again. So, we start now by introducing some terminologies, then we shall be moving on to some actual clock distribution or we can say layout algorithm or strategies how it is done so.


(Refer Slide Time: 00:43)



Terminology

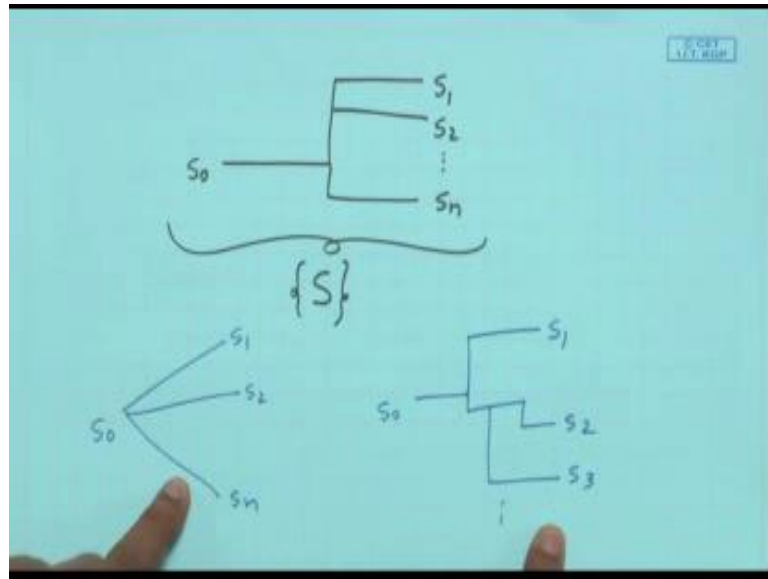
- A **clock routing instance (clock net)** is represented by $n+1$ terminals, where s_0 is designated as the source, and $S = \{s_1, s_2, \dots, s_n\}$ is designated as sinks
 - Let $s_i, 0 \leq i \leq n$, denote both a terminal and its location.
- A **clock routing solution** consists of a set of wire segments that connect all terminals of the clock net, so that a signal generated at the source propagates to all of the sinks.
 - Two aspects of clock routing solution: **topology** and **geometric embedding**.

 IIT KHARAGPUR  NPTEL ONLINE CERTIFICATION COURSES



So, we introduce some terminologies which we shall be using in some discussions later. So, we define a clock network a clock net or a clock routing instance as represented by a net with n plus 1 terminals, where there is one source called S_0 and there are n terminals S called sinks. So, diagrammatically it will look like this.

(Refer Slide Time: 01:21)



So, I have a source S_0 and there are n sinks S_1, S_2 to S_n . So, I have to layout a net like this the S_1 has to be connected to all S_1, S_2 to S_n this is a net which is I am indicating by S , this is a set.

So, given a clock net; so we are looking for a clock routing solution because ultimately from the clock source S_0 we have to use a set of wire segments to connect the terminals let see. This is the problem that S_0 has to be connected to S_1, S_2 to S_n , but there are so many ways like for example, I can say that S_0 , I can connect by straight lines to all of them or I can connect by vertical and horizontal line segments something like this and so on. So, there are so many ways right. So, this is the final solution I want to arrive at and this I refer to as the clock routing solution. So, the clock routing solution is actually the set of wire segments that will allow us to make this connection.

Now, in this clock routing solution there are 2 different sub problems you can think of; one is referred to as the topology, other is referred to as the geometric embedding. Like I shall be illustrating with the name of the example, but let me explain it first intuitively, topology means the kind of clock tree I talked about. So, how the clock tree will look like that will be my topology, like every driver is driving 2 other drivers or 4 other drivers that will be the topology that I am talking about. Once I have the topology I have to actually connect my physical clock terminals the sinks, to some lead node of this tree that is the geometrically embedding. So, exactly geometrically where do I place this, the

tree is general in nature and geometrical embedding is more specific depending on the exact location of the clock terminals, right.

(Refer Slide Time: 04:10)

The slide contains the following text:

- The clock-tree topology (clock tree) is a rooted binary tree G with n leaves corresponding to the set of sinks.
 - Internal nodes = Steiner points

The slide also features the IIT Kharagpur and NPTEL logos at the bottom left and a video inset of a speaker at the bottom right.

So, when I say a clock tree topology, it is a rooted binary tree with n leaves corresponding to the set of sinks; and internal nodes as I had said that I have to connect them by horizontal and vertical line segments, so it will like a Steiner tree. So, all the internal nodes of the tree will be referred to as Steiner points, they will have to be connected.

(Refer Slide Time: 04:35)

The slide is divided into three sections:

- Clock routing problem instance:** A diagram showing a set of sinks $s_0, s_1, s_2, s_3, s_4, s_5, s_6$ scattered on a plane.
- Connection topology:** A rooted binary tree with root u_0 and internal nodes u_1, u_2, u_3, u_4 . The leaves are the sinks $s_1, s_2, s_3, s_4, s_5, s_6$.
- Embedding:** A diagram showing the tree structure from the previous section embedded in a 2D plane using horizontal and vertical line segments to connect the sinks.

The slide also features the IIT Kharagpur and NPTEL logos at the bottom left and a video inset of a speaker at the bottom right.

Just diagram diagrammatically let me illustrate; let say I have a clock routing problem instance, let say this is a small chip. So, I am showing these are the terminals which have to be connected S 1, S 2, S 3, S 4, S 5, S 6 and suppose my clock source is somewhere in the centre S 0 here, this is S 0 and the remaining ones are the clock sinks, this is my clock routing problem instance.

Now, first thing is that we just arrive at a topology, we are saying that let us have a topology like this, like S 0 will be connecting to 2 intermediate nodes now each of these intermediate nodes can refer to drivers. This U 1 and U 2 can be drivers, so this U 1 can drive this S 1 S 2 directly, this U 2 can drive 2 more drivers U 3 U 4 and each of them is driving 2 of them. So, in this connection topology, so one thing we are assuming that every node is driving up to maximum 2 other nodes. So, whenever we are designing the buffers the buffer design will be like that. So, it will be driving a maximum of 2 other signal lines. So, the fan out will be two. So, the buffers will be relatively smaller in size.

Now, on this connection topology; so this connection topology will be doing embedding on our actual problem. So, this is one possible solution of embedding like what I was saying is that, these are the points which were already there and this is the connection topology that we want and this is one possible interconnection that we do to achieve this. Let say this one from S 0 to U 1 then S 1, S 2. So, I am showing it like this, this node maybe 1 and this S 0 and this connects to S 1 and S 2, then U 2 goes to U 3 and U 4. Let say this one is U 2 this goes to U 3 here and U 4 here. U 3 goes to S 3 and S 4, S 5 and S 6. So, this embedding you see here you get not only the exact connections, but also the location where the buffers have to be placed U 1, U 2, U 3, U 4 this is solution which shows the location of the buffers as well as the geometry of the interconnection lines, right.

(Refer Slide Time: 07:28)

Terminology

- **Clock skew:** (maximum) difference in clock signal arrival times between sinks.
$$\text{skew}(T) = \max_{i, j \in S} |t(s_0, s_i) - t(s_0, s_j)|$$
- **Local skew:** maximum difference in arrival times of the clock signal at the clock pins of two or more related sinks.
 - Sinks within distance d .
 - Flip-flops or latches connected by a directed signal path.

NPTEL ONLINE CERTIFICATION COURSES

So clock skew you already know of. So, it is the maximum difference in clock signal arrival times, like you see I say that I have a clock source S_0 , so I have. So many sinks S_1, S_2 in general S_i there are n sinks.

(Refer Slide Time: 07:41)

The diagram shows a clock source S_0 on the left. A list of sinks $S_1, S_2, \dots, S_i, \dots, S_n$ is on the right. An arrow points from S_0 to S_i with the label $t(s_0, s_i)$. A bracket on the right side of the sink list is labeled "max skew". Below this, a box contains four small circuit diagrams showing clock signal paths from a source to multiple sinks, illustrating how skew is calculated in a network.

So, the delay between the source and the sink I refer to as $t(S_0, S_i)$. So, if I calculate this delay across all S_0 and S_i pairs $S_0, S_1, S_0, S_2, S_0, S_3$ and so on and take the maximum of them that will give me the maximum skew right. So, mathematically we are expressing it like this, the total skew will be maximum of the differences. So, what is the

maximum difference in the delays between any pair of i and j . So, the delay between S_0 and S_i S_0 and S_j their difference maximum across all S_i S_j pairs that will be defined as my maximum clock skew.

Now, we define 2 terminologies local skew and global skew. See one thing we are talking about the clock signal which is going to all parts of the chip. Let say so I have a chip, let us suppose these are some of the clock terminals here, there are some clock terminals here, there are some clock terminals here; there are some clock terminals here right. See these clock terminals refer to some flip flops or some storage elements; now let us assume that these 5 dots I have shown here, they correspond to some storage cells which are connected among themselves, these 4 refer to some storage cells again which are connected among themselves, they refer to storage cells which are connected among themselves.

So, what I mean to say is that for correct operation. So, we must ensure that in every zone the flip flops which depends on each other, there should not be too much skew among themselves. Similarly in this zone or this zone this skew has to be limited, but if you take a point here and a point here they correspond to 2 storage elements, but they are not directly connected. So, even if there is a considerable skew between these 2 points that will not affect your circuit operation. So, what you mean to say is that although theoretically skew seems to be not a good thing a bad thing in fact, but all skews may not be equally important; skews that correspond to some local area of the circuit which corresponds to suppose a pipeline storage, stages with combination circuits in between they skew between those stages should be important for you, but one pipeline and another pipeline someone else, they may have some significance skew among them also that will not be a problem, right.

So, local skew we define like this, this is the maximum difference in the clock arrival time. Locally means at the clock pins of a set of related sinks, related sinks means the storage cells which are somehow connected among themselves. So, typically local skew corresponds to the sinks which are within some distance limited distance; they often refer to as I had said storage elements that are connected by a directed signal path. What is a directed signal path? The output of a flip flop is going to the input of a flip flop, via some combinational circuit, that is referred to as a directed signal path right. So, this is local skew.

(Refer Slide Time: 12:28)

- **Global skew:** maximum difference in arrival times of the clock signal at the clock pins of any two (related or unrelated) sinks.
 - Difference between shortest and longest source-sink path delays in the clock distribution network.
 - The term "skew" typically refers to "global skew".

So, if you now look at global skew, global means I am looking at it across the chip, maximum difference in arrival times correspond to any 2 sinks, they may be related they maybe unrelated. So, here with respect to the whole clock distribution network we are looking at the difference between the shortest and the longest source sink paths. So, when you refer to a skew we actually refer to as global skew, but for a practical problem local skew can be more important to you because you need not minimize skew globally for better performance you can concentrate on the local elements and try and limit skew on those on those local elements only right fine.

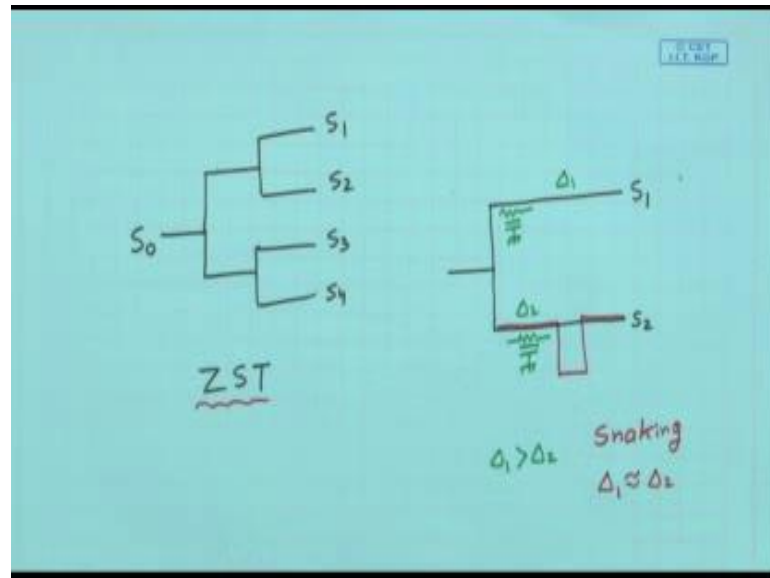
(Refer Slide Time: 13:29)

Terminologies for Clock-Tree Routing

- **Zero skew:** zero-skew tree (ZST)
 - ZST problem
- **Bounded skew:** true ZST may not be necessary in practice
 - Signoff timing analysis is sufficient with a non-zero skew bound.
 - In addition to final (signoff) timing, this relaxation can be useful with intermediate delay models when it facilitates reductions in the length of the tree.
 - Bounded-Skew Tree (BST) problem.

So, some terminologies for clock skew clock tree routing well; first sub problem this is of course, desirable and the ideal situation 0 skew. So, we try to create a 0 skew tree like when I say that from a source I create some kind of a tree, so I am not showing the buffers here the buffers will also be there.

(Refer Slide Time: 13:53)



So, let us take a small example with 4 sinks, when I say it is a 0 skewed tree ZST. it means that the delays from these S0s to each of S1, S2, S3, S4 are all equal of course, as you can understand this is an idealized situation, you can try to make the delays equal, but in practice because of variations in parameter and variations in parasitics there will be small differences in delay right, but you make your best effort to make the wires equal or you can consider the parasitics the R and C. So, we your wire length equalization may not be the main parameter, but RC delay equalization that can be more accurate in that sense we try to equalize them. So, if you are able to do that you will get something which is referred to as 0 skew tree, but as I had said computing 0 skew tree is not easy it is difficult and some variation in the parameter during fabrication can deviate this 0 skew characteristics in an actual tree design, right.

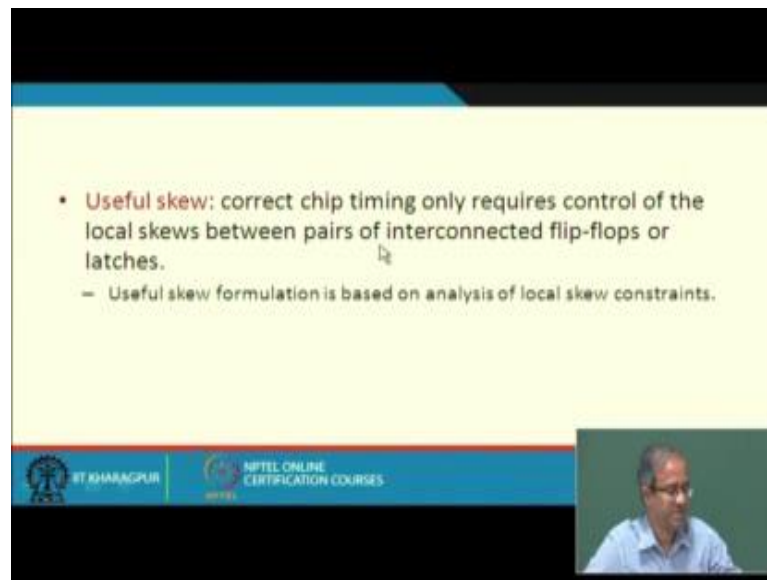
So, in practice what you typically may want to have not exactly 0 skew, but bounded skew. Let us have a clock tree where skew is limited to certain maximum limits well, we shall later see that during the backend design we have a process called signoff means we analyze all timing delays, the relevant timing delays in the circuit and we say that

everything is in order, if the maximum delays are within certain limit. So, this bounded skew model helps us not only for the final signoff, but also during intermediate steps that can help in reducing the length of the tree. Because you see to make a clock tree exactly 0 skew, maybe you may have to make some tree length longer.

Like it may so happen that let us take hypothetic example, that from 1 point you are drawing a line to a point S 1 and another line to a point S 2. So, although it looks like the lengths are equal, but there can be some parasitic resistance and capacitance effects right and these parasitics may not be the same for both the lines, because it depends on the lines which are running parallel to those lines on the same layer across different layers so many other factors are there. So, although the wire lengths are the same the capacity and resistive effects may be different. So, it may so happen that the delay of the second line let us say the delay of this line is δ_1 and this line is δ_2 , it may so happen δ_1 is greater than δ_2 .

So, for that purpose to equalize, sometimes we may have to do something called snaking this is a term which is used, which means instead of laying this wire like this maybe we will be laying it like this, make this wire length longer. So, as to match the 2 delays try to make δ_1 and δ_2 approximately equal right fine. So, here the problem with bounded skew is sometimes referred to as BST bounded skew tree problem and useful skew means as I had said that although we talked about skew across arbitrary points it is the global skew, but in practice local skews is more important.

(Refer Slide Time: 18:14)



• **Useful skew:** correct chip timing only requires control of the local skews between pairs of interconnected flip-flops or latches.

- Useful skew formulation is based on analysis of local skew constraints.

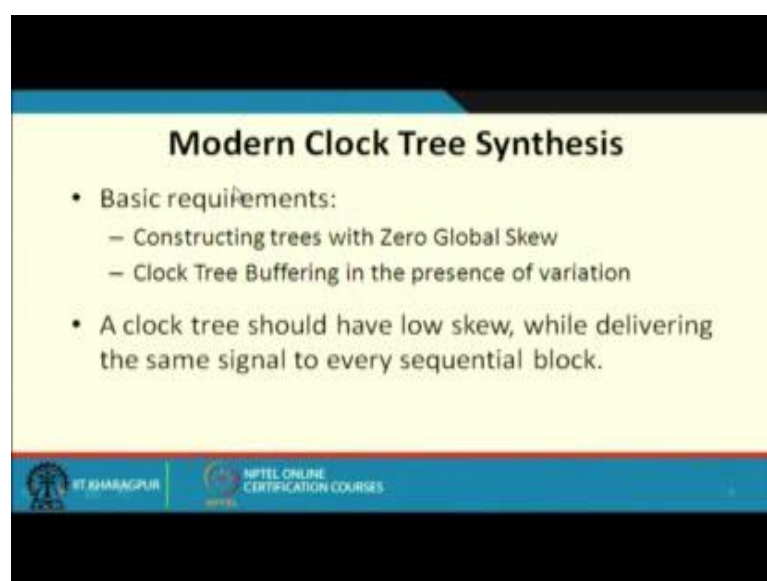
IT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

The slide features a yellow background with a blue header and footer. A small inset video of a man is visible in the bottom right corner.

So, if you can handle the local skew and try to reduce them, that will perhaps provide us with a solution where we can run a circuit with a very high clock frequency, global skew may not be that important in that respect.

Now coming to the modern clock tree synthesis, let us look at what are the basic steps which are carried out and some suggested algorithms some important algorithms which are used in the process let us look at.

(Refer Slide Time: 19:07)



Modern Clock Tree Synthesis

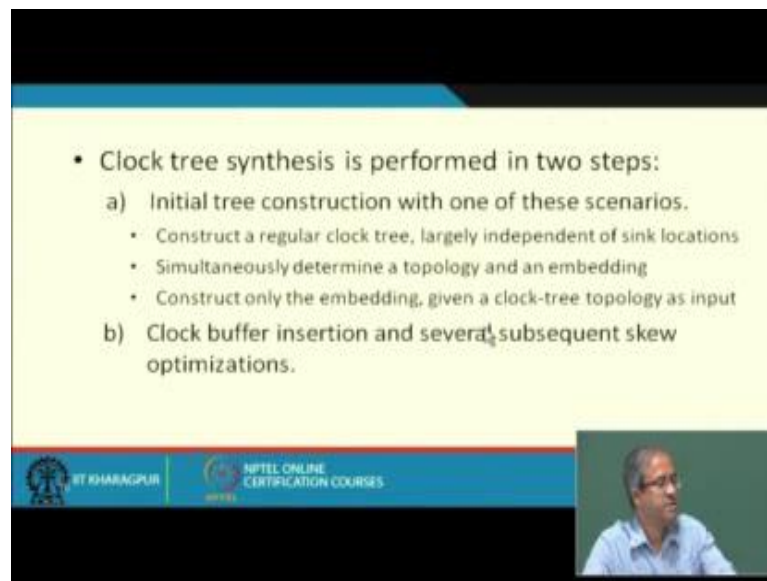
- **Basic requirements:**
 - Constructing trees with Zero Global Skew
 - Clock Tree Buffering in the presence of variation
- A clock tree should have low skew, while delivering the same signal to every sequential block.

IT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

The slide features a yellow background with a blue header and footer.

So, for modern clock tree synthesis there are 2 main requirements; the first requirement is of course, relatively easier and we shall be looking at this in some detail that how we can construct a tree, where this skew is either 0 or limited bounded, and how we can introduce the buffers in the presence of variations. So, the clock tree should have low skew so while delivering the signal to every sequential block. This is the main requirement; so we shall be looking at various ways of designing the clock tree such that this skew is either 0 or very close to 0 bounded.

(Refer Slide Time: 20:05)

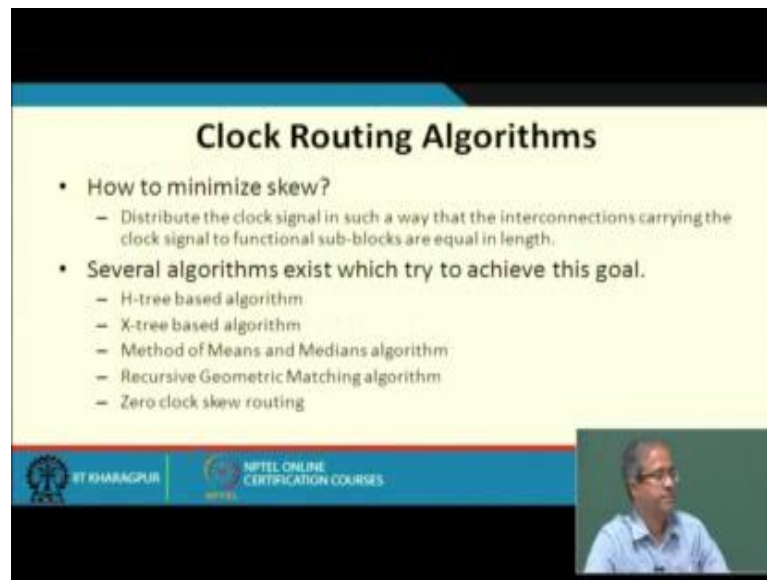


- Clock tree synthesis is performed in two steps:
 - a) Initial tree construction with one of these scenarios.
 - Construct a regular clock tree, largely independent of sink locations
 - Simultaneously determine a topology and an embedding
 - Construct only the embedding, given a clock-tree topology as input
 - b) Clock buffer insertion and several subsequent skew optimizations.

So, when we talk about clock tree synthesis. So, it is performed in 2 steps. So, we construct the initials tree. So, either we construct the tree not considering the sink locations; it is a very general tree which is applicable to designs which are independent of sink locations. Second alternative maybe we consider the sink locations, topology and embedding both we are considering together.

Third we consider them in as 2 separate steps; we assume that a clock tree has already been built. Now we are talking about embedding, just like that grid structure I told you about that I make a clock tree first then I have a mesh or a grid, the clock tree is driving that mesh and embedding means I have to tap the grid from appropriate points to take my clock signals right. So, after you do this. So, you can insert the clock buffers whenever required and you can carry out several skew optimizations like that snaking example lie we told you about, you may have to make a wire slightly longer to match with the others.

(Refer Slide Time: 21:35)



Clock Routing Algorithms

- How to minimize skew?
 - Distribute the clock signal in such a way that the interconnections carrying the clock signal to functional sub-blocks are equal in length.
- Several algorithms exist which try to achieve this goal.
 - H-tree based algorithm
 - X-tree based algorithm
 - Method of Means and Medians algorithm
 - Recursive Geometric Matching algorithm
 - Zero clock skew routing

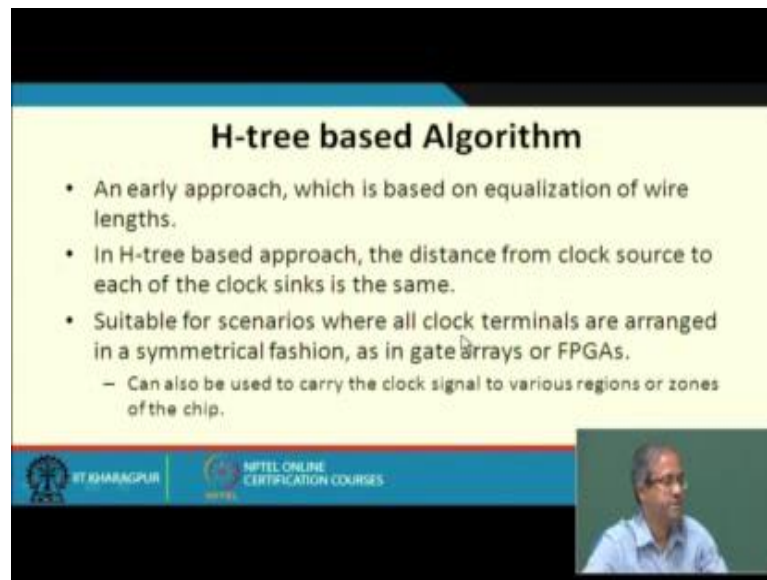
IT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES

(Speaker video inset showing a man in a blue shirt)

So, now let us look at some of the clock routing algorithms, which are used in practice. So, the main objective behind these clock routing algorithms is to minimize this skew, which means how to distribute my clock signal such that from the clock source down to the sink, how I can ensure that my inter connection lengths are all equal in length well. The first approach will consider only inter connection length, but later on we shall see that not only inter connection length we can also calculate the parasitic, we can measure the actual distance or estimate the distance and you can try to estimate actually the delay. We can estimate the delay based on the distance and the parasitics and you can try to balance the delays, right. So, to start with the first 2 algorithms we talked about they will be looking at only the distances, they will try to equalize or match the distances, right.

So, we shall be talking about several algorithms, the first 4 of them for instance they try to minimize the length the last one of course, they look at actual 0 clock skew by carrying out more accurate estimates of the delays. So, let us see this one by one, right.

(Refer Slide Time: 23:17)



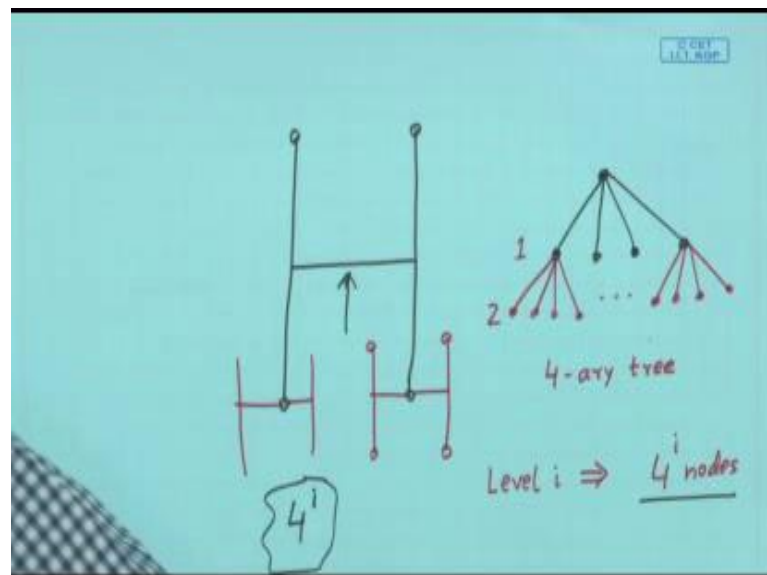
H-tree based Algorithm

- An early approach, which is based on equalization of wire lengths.
- In H-tree based approach, the distance from clock source to each of the clock sinks is the same.
- Suitable for scenarios where all clock terminals are arranged in a symmetrical fashion, as in gate arrays or FPGAs.
 - Can also be used to carry the clock signal to various regions or zones of the chip.

NPTEL ONLINE CERTIFICATION COURSES

The first algorithm we look at this is referred to as H-tree based algorithm, see H-tree based algorithm the name comes from the letter H in the English alphabet.

(Refer Slide Time: 23:47)



So, clock routing takes place in a way that look like the letter H like suppose I have an H like this, this is what I am referring to this is like a tree for example, I feed my clock and I get my clock in these 4 places.

So, this H logically speaking is like a tree with 4 child nodes. Now what you can have after that from each of these terminal points, you can have a smaller H like this. So, you

get another 4 points from each of them right. So, from each of them you get 4 child nodes and so on. So, your clock tree will look like this, this is called a this is not a binary tree this is 4-ary tree, which means every vertex has up to 4 children and 1 constraint is that as you go up in the various levels, suppose this is your level 1, this is level 2 then level 3 you go, so in level i will have you can check 4 to the power i nodes.

First levels there are 4 nodes 4 to the power 1 . Second level there are 16 nodes 4 to the level 2 ; third level there will be 64 nodes 4 to the power 3 and so on. So, this kind of a clock network you can build only for those cases where there will be 4 to the power i nodes, but there are some advantages we shall see let us come back to the slides.

So, in this approach we shall see that the distance the geometric distance from the clock source to each of the terminal points or sling sinks, is guaranteed to be the same well. Loosely speaking this kind of a approach can be used for scenarios where you need a large number of clock terminals, and they are all arranged regularly all across the chip like for example, in a gate array or an FPGA, but also you can use this for the 2 level as I had said, you have a clock tree which feeds a grid or a mesh, you can have the H tree as the first level tree. So, it regularly feeds the clock to regular places, which will be connected to the grid that is also one way. So, this can also be used to carry the signal to various regions or zones.

(Refer Slide Time: 27:00)

- In (a), all points are exactly 7 units from the point P_0 , and hence the skew is zero.
- This ensures minimum-delay routing as well.
 - P_2 and P_2 are at a distance 7 (rectilinear distance).
- Can be generalized to 4^m points, where m is an integer.

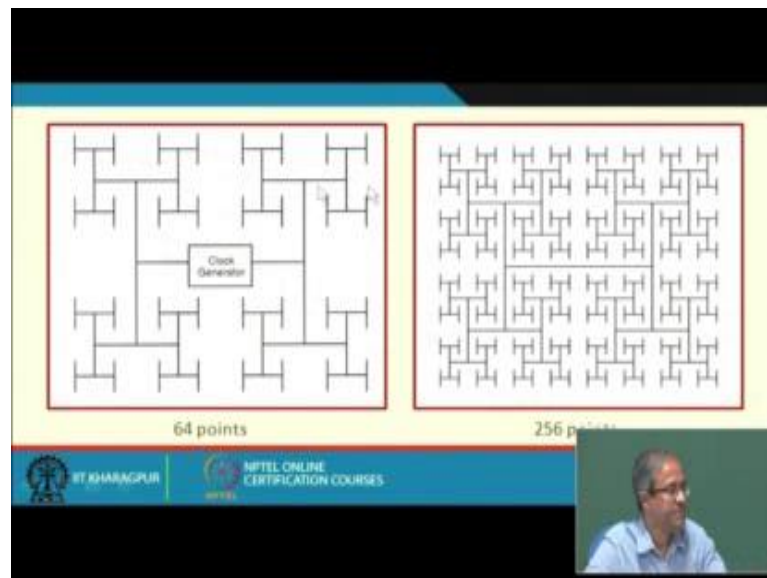
(a) P_0 (distance = 7)

(b) P_0 (distance = 19)

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES

So, let us look at this H shapes in terms of the geometry. So, here the grid lines are shown, let say these this H the centre points the terminal points are placed like this, then from this point up to each of the terminal points you can check the distance will be 1 2 3 4 5 6 7 and if you look at the centre point, it will be 4 - 1 2 3 4. So, either from the centre point of from this entry point, the distance up to each of these nodes will be 7. So, if you construct this H up to another level, and if you calculate the distance similarly the distance will be 19; 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 15, 16, 17, 18, 19 like this. So, from this point down to each of the terminals the total length of the wire is guaranteed to be equal and as I had said this can be generalized to any power of 4, this is 4 terminals 16 terminals, so you can go upto 64 terminals, 256 terminals like this.

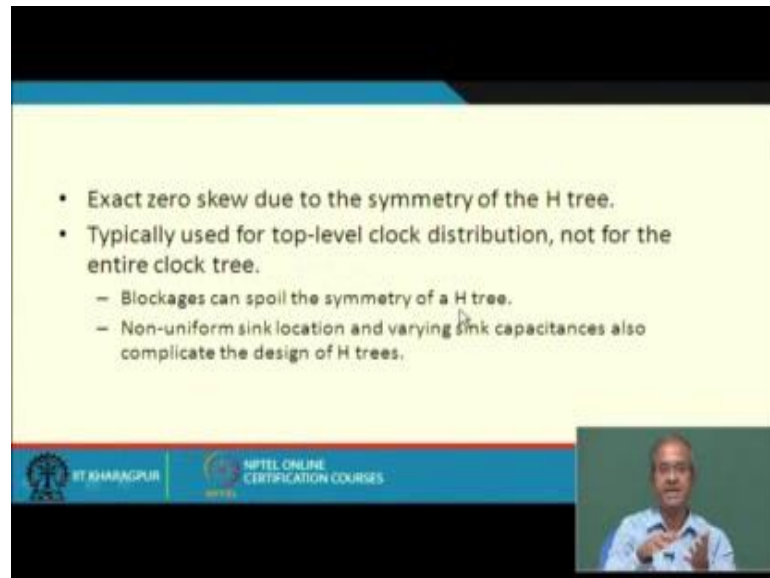
(Refer Slide Time: 28:15)



So, as you can see as you go up and up you get a structure where the terminals are very regularly spread all across the chip. So, in any scenario where you need the clock terminals to be distributed regularly across the chip, you can use this method or this principle.

Now, another thing you can also check that this connection or the routing is such that all the wires are connected on the same metal layer, you do not need to move from horizontal to vertical, vertical to horizontal from 2 layers. It is a purely planar routing, the entire connection you are carrying out on a single plane and you are guarantying equal distance from the source to each of the sinks, this is a big advantage for H tree.

(Refer Slide Time: 29:19)

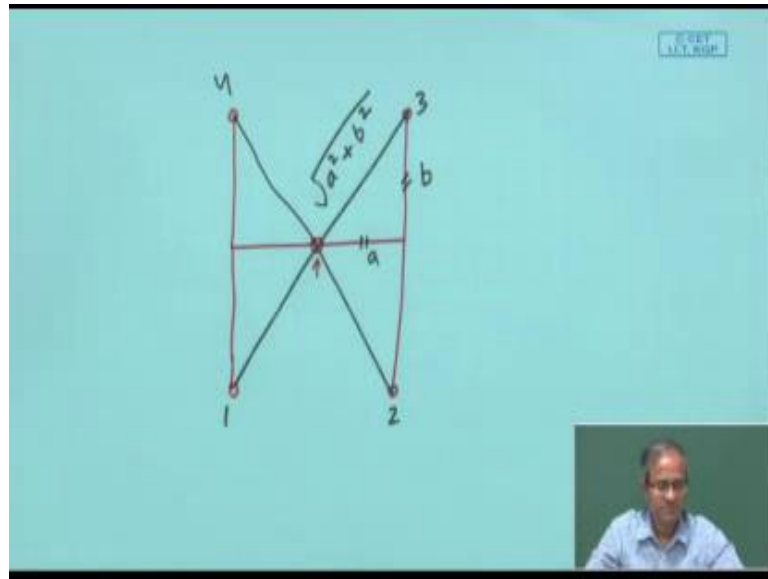


- Exact zero skew due to the symmetry of the H tree.
- Typically used for top-level clock distribution, not for the entire clock tree.
 - Blockages can spoil the symmetry of a H tree.
 - Non-uniform sink location and varying sink capacitances also complicate the design of H trees.

So, what do you guarantee here? Exact 0 skew well in terms of the distance, here we are ignoring the parasitics RC for the time being, exact 0 skew with respect to the length of the wires because of the symmetry of the tree. But as I had said this H tree is typically not used for the entire clock distribution, this is used for the top level clock distribution. The top level this H tree can feed that grid structure and the grid can in the second level use to feed the other points or other pins of the circuit.

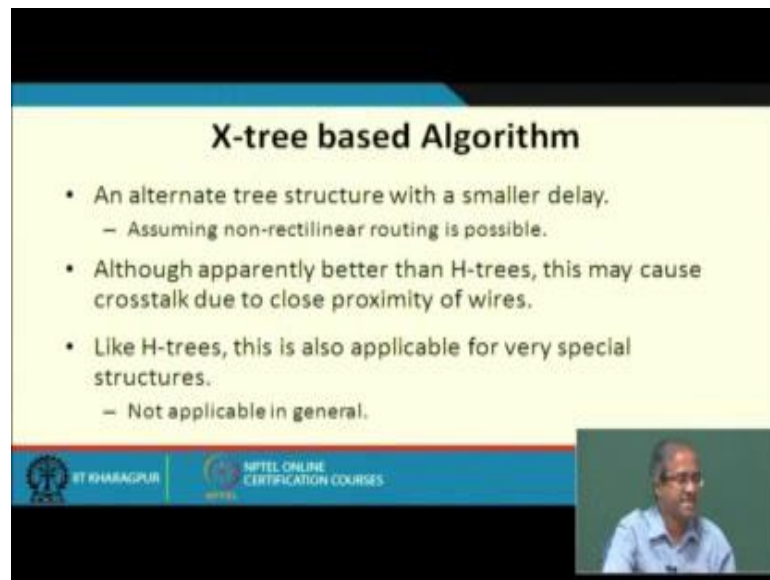
Now, if there are blockages on the layer, then the design of the H tree will be spoiled; that is why I said so whenever you are laying out a clock tree that has to be laid out on a separate metal layer, where there are no blockages you can layout the lines as you want. You know also if the sink locations and sink capacitances are vary then the design of the H tree will be different, because here you are saying exact 0 skew only with respect to the distances.

(Refer Slide Time: 30:49)



Well now another thing you just observe suppose you consider an H. So, this an H tree which was connecting these 4 points, and suppose this is the central point from where the point is coming let us say. Now what I am saying is that because we are laying out all the connections on the same layer, so I do not have any compulsion that the wires up to horizontal and vertical only, so I can also run the wires diagonally if I want. So, here you see just inside this edge my main requirements was to connect the points 1 2 3 and 4. So, by doing this I am making the wires longer why do not I connect them by straight lines like this, they will obviously, be shorter because this was this, if this is a and this is b this will be square root of a square by b square Pythagoras theorem this will be shorter. So, if I want shorter connection I can make it as an X instead of a H.

(Refer Slide Time: 32:10)



X-tree based Algorithm

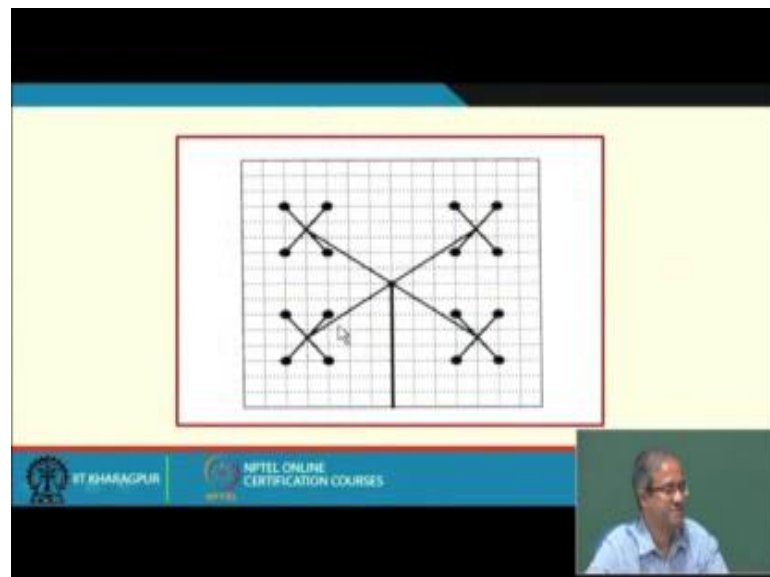
- An alternate tree structure with a smaller delay.
 - Assuming non-rectilinear routing is possible.
- Although apparently better than H-trees, this may cause crosstalk due to close proximity of wires.
- Like H-trees, this is also applicable for very special structures.
 - Not applicable in general.

BT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES

(Video inset shows a man speaking)

So, our next alternate algorithm this is called X-tree. Same thing instead of the X type structure we are using an X type structure like this. So, I use a big X and with each of the terminal points I use smaller Xs, from each of them I will be using even smaller Xs like that, right.

(Refer Slide Time: 32:22)



The diagram illustrates an X-tree structure on a grid. A central vertical line (the root) splits into two diagonal lines (the first level of Xs). Each of these diagonal lines further splits into two more diagonal lines (the second level of Xs), resulting in a total of four terminal points. The structure is symmetric and non-rectilinear.

BT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES

(Video inset shows a man speaking)

So, we are assuming that we are allowed to layout the wires which are non horizontal and vertical non vertical also.

So at first look you may say that well this looks better, the wire lengths are shorter; but there is a problem that is also there. Like you see in an H tree the wires were so regular they were symmetrical, but now here you see between this line and this small line, the wires are running closer together parallel so there will be more capacitances between them, but between this wire and this wire there will be much less capacitance they are much further away. So, because of the slanted kind of connection, the capacitance between the wires which are running closer to each other will be varying from 1 segment of the X to other. So, it will not be exactly 0 skew in that sense if you consider the parasitics with respect to R and C. So, this is one drawback apparently better it may, but there can be cross talk due to close proximity of the wires.

So, again H trees just like H trees you can use it for special structure like creating a top level tree than feeding it to the next level like that you can use this also. So, this H and X and 2 alternative ways, which typically people build the first level clock tree before they are distributed to the different parts of the trip.

So, with this we come to the end of this lecture number 29. So, we continue with other implementations of clock tree routing in our next lecture.

Thank you.