VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology Roorkee

Week 07

Lecture 34

Introduction to Clock Tree Synthesis

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss about the clock tree synthesis. So, the content of this lecture includes introduction to the clock tree synthesis, then we will discuss about what are the design consideration we should follow for clocking system, then we will discuss about how we can effectively calculate the delay of a clock tree. So, this is the physical design flow where we are now focusing on clock tree synthesis. It comes after the placement of the blocks and so, clock tree synthesis is one of the very important step of VLSI physical design. So, basically in most of the digital systems and digital designs are synchronous.

So, it is synchronous with respect to the clock. This clock signal is used to maintain synchronization of all computation that takes place inside a chip. So, it is a very important signal. It should be routed inside the chip with utmost care. The clock signal is generated outside the chip. So, let us say if you have a chip where the basically PLL is not inside the chip, then the clock signal is generated outside the chip. So, this generated signal should come in inside the clock pin. It should pass to the chip through the clock pin and it was routed throughout all the blocks inside the chip, all the flip-flops inside the chip, all the latches inside the chip. So, basically if you can look into this synchronous system, this chip performance is directly proportional to the clock frequency.

This clock net, clock nets are the clock the interconnects carrying the clock signal is the clock nets that needs to be routed with great precision. So, what it basically determines the maximum frequency of the clock signal on which we can operate the chip. So, the maximum frequency will determined by this clock signal. So, basically, we have clock routers. So, the clock routers is the router or the algorithm which is used to route the clock signals. It needs to take several factors into account. For example, the first one is the resistance and capacitance of the metal layers. Then the second one is basically noise and crosstalk in the wires and the third one is basically what is the type of load it is

driving. How many flip-flops it is driving. Based on that the router will insert buffer whenever it is necessary. So, if you can see this clock signal is a global net. It is a global net and this clock line is usually very long. Let us say if you have a big chip and you have a clock pin here, it has to route to all the flip-flops inside the chip. So, let us say I have flip-flop here, I have flip-flop here, I have flip-flop here, I have flip-flop here, then it can have branches and the clock will route to them. So, the clock net is very long in nature because it is going to all the flip-flops inside the chip.

So, the delay caused by this long wire is due to the capacitance and resistance of that interconnect. So, there are two types of integration level, one is called the low-level integration and one is called the high level integration. When we are doing the low-level integration, the gate capacitance is the dominant factor compared to our interconnect capacitance because in low level integration the area is small and interconnect distance will be small and the interconnect capacitance will be small. So, this interconnect capacitance are ignored in case of low level integration. In case of high level integration, this gate capacitance is smaller compared to the interconnect capacitance as the area of the chip is increases, the interconnect distance passing through the different location of the chip also increases.

Then the interconnect capacitance must be taken into account when the clock wires are routed. So, basically we need to consider the combined effect of resistance and capacitance of the interconnect. So, this basically lead to the RC delay. So basically your RC delay increases which increases with the other square of the scaling factor. So, if you go from one technology to the other, you have certain scaling factor is there, your interconnect distance or delay will be increased by square of that.

So, in a given technology node, RC delay cannot be reduced by making the wire wider. If I increase the wire wider, your R will reduce, resistance will be reduced. However, your C will increase. So, it is not an effective way to make the wires or the interconnect wider. So, one effective way to reduce the delay is use of the buffer. We have to use the specialized buffer in the clock tree. So, it helps to preserve the clock waveform because if you pass the signal for a longer distance without buffering, thus transition time or the sleeve of the clock signal will be degraded. So, nothing comes freely. So, the disadvantage of using the buffer is that buffer has some inertial delay. By default, there is some delay of the buffer. It also consumes one thing is the area and second thing the power. So, it leads to increase in the clock power. So, increase in clock power. So, this leads to increase in clock power. So, need to be considered in the overall layout of high performance design.

So, this basically the power consumption and the area needs to be considered when you are designing a high performance chip. So, this clock tree or the buffers can be used in two different way. So, this buffers can be used in two different way. So, if you can see

here, you have a big centralized buffer, we have a big centralized buffer and you have a small branching out of it. And your flip flop will sit in each of the branches like this to get the clock signal. So, now this is called big centralized buffer approach. Then another is called distributed buffer approach. We have branches, different types of branches. Let us say this is the main branch, then this sub branch, then these are the sub sub branches and the flip flop will sit at these points. So, this is called distributed buffer approach. Here basically advantage is that the size of the buffer should not be weak. In this case, we will discuss how we can compute the interconnect delay of the clock tree. So, exact computation of RC delay of the clock tree is very difficult. So, there are some kind of approximations are used to find the delay through the clock tree in a reasonable accuracy. So, we do some kind of linear approximation to get the result in less time. So, one of the most popular method of finding the interconnect delay is called Elmore delay model which is used to calculate the delay of the RC tree. We will discuss about this Elmore delay model. So, if you have a clock network with interconnects, then we can analyze that network in different manners using different models, RC models. One of the model is called distributed RC line, which is more accurate, but it takes more time to simulate. Distributed RC line will take more time for simulation, but it is more accurate, but time is also one of the important factor when you are designing the chip.

So, there are some kind of equivalent pi model is there and equivalent T models of the network is created to do the analysis in less time. So, let us consider a RC network given here. And we have a basically step response is given here. This is a step response is given at S point. Then I am interested to find the delay from the source node to the I node.

So there are some property need to be satisfied in case of whenever we are applying the Elmore delay. So, what are the properties are basically the network should have a single input node that is the first point. So, all capacitance would be between the node and the ground. For example, I have a resistance, the capacitance would be between this node and the ground.

This is one assumption. This is one assumption. So, this kind of capacitance are allowed. However, this is allowed. But if I have a capacitance like this between the two nodes and there is a resistance here like a Miller cap or something, then you cannot use Elmore delay model. So, this is not allowed. This is not allowed. And all the network does not contain any resistive loop. It does not contain any resistive loop which makes it a tree. Total resistance calculation formulas are there. For example, you have to find the total resistance from the source node to any node in the network.

Basically, this is called the path resistance. There is a concept called path resistance from the source node to that node. For example, let us say I will do an analysis here from the source node to my node 4. What is my resistance? If I can see here, I have R1 in the path, I have R3 in the path, I have R4 in the path. So, I have R44 is basically the sum of R1, R3

and R4. So, this is basically the path resistance from node S to the node 4. So, this R44 is the path resistance from node S to node 4. So, in this one, we are discussing about the shared path resistance. The shared path resistance Rik is the resistance shared among the path from the root node. Root node is S here from root node to I and from root node to k. This is called as what is the common resistances among them. So, if you can see here, Rik is basically Rj which belongs to intersection of the path between S to I and path from S to k. So, if I take an example, so it will collect all the resistance which is intersection of both of them. So, if you can see Ri4, I24 then you have basically the resistance from S to I is R1, R3 and Ri. So Ri4 will be R1 plus R3 plus Ri intersection of R1 plus R3 plus R4. So, we need to consider which are the common among them, which is the common among them is basically R1 plus R3. So, this Ri4 is basically R1 plus R3. Ri4 is R1 plus R3. Then Ri2 is basically R1. Ri2 will be basically R1. So, what we are doing here is that we are calculating the Elmore delay at node I is given by this capacitance at that node multiplied by the shared path resistance. This shared path resistance, first we have to calculate the shared path resistance at a node then multiplied with the capacitance of that node. We have to do the summation over all the nodes k equal to 1 to n. So here in this case what we are doing here is that Elmore delay of node I, node I is this one from the source node S is given in the RC tree is basically R1. You have R1 is the shared resistance at node 1 multiplied by C1. Similarly for the node 2, the shared resistance is R1. Ri2 is basically R1 and the capacitance at node 2 is basically C2. Similarly, you can apply for all the nodes. Similarly, you can do the calculation for all the nodes. Now this is the Elmore delay from the source node to the node I in the given RC tree. Now what we are doing is that we are doing the or let us say if I have unbuffered tree what is happening is the total capacitance of the sub tree can be defined as recursively. So, we have to calculate the each of the capacitance separately add with the present capacitance then to find the total capacitance. Ci is the node I capacitance and Ri is basically the resistance of the edge I. So, the Is of I is a set of nodes adjacent to the node and does not content in its parent node. So, all the adjacent nodes are denoted by Is of I. So, we are interested there are two approaches are there one is unbuffered tree one is called buffered tree. In case of unbuffered tree all the capacitance are added because all the capacitance are in parallel. Let us say I have a capacitance here I have a capacitance here I have a capacitance here all are in parallel in parallel all the capacitance will add up. So that is the concept is used here. So, all the capacitance will be added up in the unbuffered tree because there is no buffer inserted in the interconnect.

So here this Is of I is the this Is of I is saying that set of node adjacent to the nodes and does not content in its parent node. This is the parent node. This is the parent node and these are the calculation for the adjacent node. So, we can do this calculation to add all the capacitances. Then if you have a buffered tree if you have a buffered tree how we can do the delay calculations.

So, it is very interesting here because you have several different equivalent circuit for the model and what we are doing is that we are adding a delay. So, you have an extra delay introduced due to the buffer and that one if you can see here the buffered tree is and its equivalent model. So let us say I have a clock source and I have a latch and I have two wires now. One is wire one and wire two. So, in this case we are adding a buffer in between. So, if you are adding a buffer in between all the capacitance and resistance will not add up and they will be isolated by the buffer. And this buffer has another important point it helps in improving the slew. Let us say I have a degraded input slew given to the input of the buffer. The output of the buffer will have a basically fast transition time. So, the slew at this point is bad but the slew at the output of the buffer is good.

So, this is one of the requirement for the clock tree. So, it solves two purpose introduction of the buffer helps in reducing the delay through the interconnect and second point is that it improves the transition time of the clock signal. So, Db is denoted the internal delay of the buffer and Rb is denotes the output resistance and Cb is basically the input capacitance. So, we have two basically independent RC tree. Independent RC tree is there and this independent RC tree will contribute to the delay and the buffer of the delay of the buffer will also added in the while doing the interconnect delay. So here what is happening is that if I have a node I is a buffer node then this is the capacitance.

Its capacitance small here. If I have a buffer in the path otherwise the capacitance will be increasing by this amount. So, introducing a buffer will help in reducing the capacitance of the overall interconnect. Now we will discuss about the clock skew is basically a maximum difference in the arrival time of the signal between two sinks. So, if you have two flip flop here the arrival time difference of the clock signal will introduce skew. So let us say this here is t1 and here it is t2 the difference of t2-t1 is just skew.

So, in this case it was defined by basically t s0 to si. Let us say I have a source node s0 and this is let us say i and this is sj this is the j node. So this time t s0, si is the delay from here to here and then the delay from here to here is basically t s0 to sj. So you take the difference take the mode of that one that will give you my clock skew. So here I am giving a one instance if you have multiple instance point you take the max of that it will give you the maximum bound on the clock skew. What is the maximum skew that we need to take into account while doing the timing analysis.

So, this is the clock skew of a clock tree. So, the t u, v denotes the signal delay between the nodes u and v. So, if you can see here I have two cases case one here okay this is case one and this is case two. In case one the skew is basically if you can see here maximum delay is 20 here and the minimum delay is 9 here. So, as I told you this is the max value so the difference basically skew here is 20-9 is 11. But if you can see in the second example it is a balanced tree in all the nodes all the points arrival time is same arrival time is 20 then the clock skew is 0 here.

So, this is a better design. Case two is a better design. So, what is the main objective of our clock routing? First objective is to reduce the clock skew, reduce the clock skew improve the circuit performance we discussed that in our timing analysis lecture. Then our second objective is to reduce the delay through the interconnect which basically guarantees faster clock and high frequency operation of the overall chip. So, we discussed about clock tree synthesis some of the terminologies and some of the objective of the clock tree synthesis.

Thank you for your attention.