VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology Roorkee

Week 07

Lecture 36

Clock Routing Algorithms – II

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss some more clock routing algorithms. The content of this lecture includes method of means and medians approach for clock tree synthesis. Then the second approach is recursive geometric matching based clock tree synthesis. So, the clock tree synthesis is one of the important task for a VLSI physical design person. So, it is very important to understand how the clock tree is inserted in a VLSI physical design flow. So, what is the purpose here is to distinguish different approaches because it will help us to solve the skew problem in the clock tree. So, the first most interesting algorithm is called the method of means and medians. So, here it was proposed to overcome the topology limitations of the edge tree. Here what is the basic idea is basically we will recursively partition all the sink terminals.

So, the sink terminals are basically your location of your flip flops and the latches. So, what we are doing here is that we are partitioning the number of flip flop and latches in the design into subsets. And whenever we are creating the subsets of the overall set, we maintain the same number of sink nodes in each subset. So, then we will find the centre of mass of the set. That is connected to the centre of mass of the two subsets. So, we will find the centre of mass of the set and we will find the centre of mass of the two subsets and connect them. So, in this method we will find the clock tree. So, we will discuss the method. This is the pseudo code for the MMM algorithm, method of means and medians. So, first of all, we will see what is input to the algorithm and what is coming out of the algorithm.

The input to the algorithm is the set of sink node that is denoted by S. So, the sink node whatever I told earlier is the number of flip-flops and latches in the design. Then we have a empty tree which is having no element in that tree. So, initially we have a empty tree and set of sinks going as input to the algorithm and what is coming out we are getting the finally the clock tree, how it is routed. So, here the condition is that if the cardinality or

the mod of S is less than 1 means that there is no element in that set then we will terminate the algorithm. Otherwise, we will continue this algorithm. So, first of all, we will find the center of mass of S. So, what is the center of mass? Let us say I have 4 points. So, let us say this is x1, y1, this is x2, y2, let us say this is x3, y3, then it is x4, y4, then the center of mass of these 4 points can be found out in both the direction x and y. So, let us say this x0 is the center of mass in the x direction. So, this will be x1 plus x2 plus x3 plus x4 divided by 4. So, we can write this one as summation i equal to 1 to 4 xi by 4. So, similarly this is for x coordinate. So, this is for x coordinate. Similarly, I can do it for y coordinate. y0 will be y1 plus y2 plus y3 plus y4 divided by 4. So, similarly I can write summation i equal to 1 to 4 yi divided by 4. So, this is corresponding to the y coordinate. So, for this one we found out x0 and y0 which is given here. Now what we have to do is that we partition the set into two sub sets like SA and SB.

What is input to the partition is S which is basically the union S is union of SA and SB. So, what we are doing here is that we are partitioning based on finding the median. Median means we are partition the set S into two parts such that each partition will have equal number of sink nodes. So, I have SA and SB. I have a set SA then I will find the center of mass of SA then which is found out to be XA and YA. Similarly the center of mass of SB is basically XB, YB. Now what we are doing? We have found three center of mass. One is corresponding to S that is x0, y0. Then we have SA. We have a center of mass SA, YA. Then SB which is basically center of mass is basically is XB, YB. So, now I have three center of mass then I will connect. So, how should I connect? I will connect the center of mass basically x0, y0 with XA, YA using a metal segment. So, basically, I will repeat for set B. So, this first one, let us say this is the first one is given here, the first one.

This is the one step. The second step is the, this one is the second step is done here. Now I did the routing and connect the center of masses. Then I will do this thing recursively and I call this same function, same basic MMM recursively till I find the basically the leaf nodes whatever the constraint it is given in the starting of the algorithm. So, this is the basically recursive algorithm and it use the concept of means and medians. So, we have already discussed this one. Again I am repeating. So, here for each subsets I am finding the mean of x coordinate and y coordinate and which is nothing but my center of mass point of the set S. So, this is giving corresponding to x coordinate, this is corresponding to my y coordinate. Then I will show you the procedure here.

So, basically if you can see I have, these are my sink nodes actually. These black dots are the sink nodes. Then I find the center of mass or center of gravity which is found to be this point. So, this is my center of gravity or center of mass. So, now what I have to do? I have to divide this whole set into two partitions, partition A and partition B. So, how do I do the partition based on cardinality of each of the subsets. So, the cardinality of the each subset should be equal. So, what I did? I have here 1, 2, 3, 4, 5, 6, 7, 8, 8 sink node in

partition A. Let us say this is partition A and this is partition B. It has also 1, 2, 3, 4, 5, 6, 7, 8. So, here I have 8 nodes, sink nodes. Here I have also 8 sink nodes. Now I have to repeat the same center of mass concept for the sub-partitions. So, I found for partition A, so this is my center of mass and this partition B, this is my center of mass. So, now I find the center of mass for both the partitions. Now we have a center of mass of the overall set. So, I need to connect them. I need to connect all of them. So, this is x0, y0 whatever I showed earlier, this is xa, ya. Then this is xb, yb. So, now I connect them. I will repeat the same procedure for rest of the nodes and create this kind of clock tree. So, this is one method of finding the clock tree which is called method of means and medians. So, if we can look into this MMM method, it is basically top-down methodology. It works from the top sink node to the bottom sink nodes.

There is another approach is there which is called recursive geometric matching algorithm which is working from bottom up approach. So, here what is happening is that what is the basic concept behind this RGM is that we are recursively finding the minimum cost geometric matching of the n sink nodes. So, once we find that, then we find basically set of n by 2 line segment that match n endpoints. So, we will locally match two of the sink node and create n by 2 line segments such that my total wire length is minimum. So, I have an objective to minimize the wire length that is my objective. What is my goal? I have to create n by 2 line segment. So, how the line segments are chosen? If two sink nodes are near to each other such that the minimum wire length can be achieved. So, I have to create n by 2 line segment from n endpoints. So, after I created this n by 2 line segment for each matching step, a balance or tapping point is found. So, what I have to do? Whenever I have a line segment, I have to find a tapping point. So, tapping point will connect one line segment with the other line segment that is one thing and it cannot be middle of the line segment also. It depends upon how I can basically maintain 0 clock skew. So, these lines whatever it is that line segments are there. Let us say I will give an example here. So, these are the sink nodes let us say.

These are the sink nodes, 4 sink nodes are there. So, what I have to do? First of all I will do 2 line segments. So, then how I can tap a node in between the line segment? I have to tap a middle of this one and middle of this one such that my 0 skew is achieved. So, this is called basically I need to find n by 2 tapping points from the input. So, which one is a tapping point here? These are the tapping points and this is the line segment.

This is the line segment and these are the actually sink nodes. So, very interesting, it is not very lot of math is involved here. It is very simple concepts. My goal is to achieve 0 skew. How should I draw the interconnect such that I can achieve 0 skew or minimum skew? 0 skew may not be possible all the case so as minimum skew as possible. So, here basically n by 2 tapping points will be created from the input line segment to the next matching step. This process will be repeated till I will get to connect all the nodes in the set. So, basically this is my pseudo code of the RGM algorithm recursive geometric

matching algorithm. So, if you can see here what is input to the algorithm like the previous algorithm MMM algorithm we have the same inputs and similar outputs. We have set of sink node S and empty tree T and the output the basically sink node with the clock tree synthesized.

The T is basically sink nodes with the clock tree synthesized. The first one is obvious whenever mod is less than 1 then there is nothing to route then we will complete the algorithm. So, what I am doing here is that I am finding M, I am finding M the minimum cost geometric matching over S. So, this S is the set of sinks.

So, then I am finding the M. So, initially we have a set S prime which is phi there is no element in that one. Then we will add elements to that set. So, the M denotes the set of matching pair Pi, Pj over S. So, that procedure will be repeated. So, what we are doing here is that for each pair in that M matching set we have a matching set actually we are taking each pair of points add to the tree. So, we have a already we have a sub tree. We have a sub tree of T rooted at Pi. Then we have a Pj which is a sub tree rooted at Pj. Then we are creating a tapping point between Pi and Pj. Then we create a updated tree. So, this tree T, TP is basically the updated tree clock tree where we have this sub trees are there.

This is one sub tree. This is another sub tree and we are adding one more branch to that sub tree and this branch is connected with the sub tree through this tapping point that is the main point. So, this branch will connect to the tree through the tapping point. So, what is the objective? We have to minimize the clock skew. So, now the TP will be connected to your S prime. So, this step the last step what it is doing is that we are calling this same RGM recursively till all the elements in the set S is 1. This process will continue till element in the set S becomes 1. So, then we will stop the algorithm. So, let us look into how things are happening here. First one this is the set of sink nodes actually. This is our set of sink nodes. Then what we are doing here is that we will create the line segment. The second step is basically create the line segment such that my wire length is minimum. My wire length is minimum cost geometric matching we are doing. So, here let us say how many sink nodes are there. So, now we have n by 2 line segments. So, then what we have to find the tapping point in each line segment. Then I need to find the tapping point in each line segments. So, this red dots are the tapping point in the line segment. Remember that this tapping point may not be at the middle of the line segment. So, it depends upon how I can make my clock skew as minimum as possible.

So, this point tapping point may not be at the middle of the line segment. Now what I am doing here I am connecting the tapping points nearby and create a tree quite similar to H3 but it is not exactly H3. So, this is the minimum cost geometric matching possible. Then these soft trees, these are the sub trees actually. Let us say this is S1 sub tree, this is S2 sub tree, S3 sub tree, S4 sub tree. Then again we need to find the tapping point in the sub trees. So, these red lines in these line segments we need to find the tapping points such

that my clock skew will be 0. Now these tapping points is created and connected. These are the tapping point of the sub trees. Then this is created where my total clock skew should be as minimum as possible. It may not be 0. So, that is the goal. So, in practice actually this RGM improves the clock tree balance and wire length over MMM compared to MMM algorithm it improves the wire length and it also balances the clock skew because it works from the local level to the global level. It works from the local sink nodes towards the global clock tree synthesis CTS. So, that is the reason it is better than this MMM algorithm. However, we can achieve 0 clock skew in case of MMM algorithm. However, RGB may not guarantee actually 0 skew in this approach.

So, that is one of the limitations, but it will create a balance clock tree with minimum wire length. So, if you can see here the comparison between our MMM in the left hand side. So, this is basically the problem statement. I have 4 sink nodes are there. Here I have in this case I have 4 sink nodes are there. Now what I am doing? I am doing division and I can do basically this is the final clock tree after using MMM. Then the same set of problem is given to the RGM algorithm and this is the final clock tree synthesis. If you can see here the wire length is minimum in case of RGM. So, wire length is minimum in RGM. So, here if you can see interconnect length is minimum in case of RGM. However the interconnect length in the clock tree is large in case of MMM algorithm. If your interconnect is larger then you have more delay in the clock path in case of MMM algorithm. But it is more balanced and it can produce minimum clock skew, but this RGM may not lead to minimum clock skew, but it is targeted to minimize the total wire length in the clock tree synthesis procedure. In this lecture we discuss about two popular clock tree synthesis algorithm. One is called MMM which stands for methods of means and medians. The second method is called RGM recursive geometric matching algorithm. We discuss both the algorithm with their advantages and disadvantage with one example.

Thank you for your attention.