VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology Roorkee

Week 07

Lecture 35

Clock Routing Algorithms – I

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss about different clock routing algorithms. The content of this lecture includes problem formulation for clock routing. Then we will discuss about some of the clock routing algorithm, for example, most popular H-Tree based algorithm. So, first of all, we will discuss about this, how we can formulate the problem for basically clock routing. We have basically zero skew problem means we want the skew is basically we have discussed in the lecture number 12 regarding skew.

If you have not gone through that lecture, please go and revisit the lecture number 12. The skew is basically we want the clock skew should be ideally zero. So, if the clock tree exhibit a zero skew, then it is called a zero clock skew or ZST and given a set S of sink locations, we need to construct ZST of T of S with minimum cost. In this context, a connection topology G is also given. Now we will discuss about the bounded skew problem. So, the bounded skew problem due to manufacturing variation or process variation, so the both the transistors and interconnect will basically have variation in its behavior. So, this variation of its behavior will translate to the variation in the delay in the interconnects and the delay of the logic gates. In actual case, it is not possible to get zero clock skew. So, there should be some bound on the clock skew.

So, in practical clock tree routing does not typically achieve exact zero skew, but it has a range of the skew. So, in this case, we have given a set S of sink locations. Sink locations are the locations of the flip flops and the latches where our clock is going to be connected. Then we have a skew bound U of B which is a greater than zero. We need to construct the clock tree T of S with the skew which is bounded by T of S basically should be less than equals to U B having minimum cost. Minimum cost is cost of the delay through the clock tree or interconnect delay. Those delays should be as minimum as possible because whenever you are inserting the buffers in the clock tree that will also consume some amount of power. So, this cost will include all the delay through the interconnect, power consumption through the buffers all these into account. So, now we have something called useful skew problem. So, basically whenever we are looking for this clock tree routing problem, if you are using the global concept of optimizing the skew, it is not a good idea.

Because the useful skew information is based on the analysis of the local skew constraints. So, basically, we need to look into the local skew constraint while basically analyzing the skew instead of dealing this one as a global problem. So, now we will introduce the useful skew in a clock tree by perturbing the clock arrival times at the sink nodes. We will see that how much clock arrival time is changing at each of the sink nodes. So, clock skew is basically a spatial variation because of the location of the let us say I have a chip, I have flip flops sitting here.

So, the arrival time of the clock at all the flip flop is different. So, because of the space position of the flip flop inside the chip, the clock can arrive at each of the nodes at different time. So, based on that we can define the clock skew. So, the arrival time difference between the two flip flops will translate to the clock skew. These are basically the sink nodes actually.

These are the sink nodes. All the flip flops and the latches are the sink nodes. Then we are either minimizing the clock period or maximizing the safety margin whenever we are discussing about the useful skew problem. We will discuss with one example here. So, basically we have discussed about the clock skew in the previous lecture 12, but for reference I am just writing the two equations, max constraint equation and the mean constraint equation for the positive skew.

So, I am discussing this one for the positive skew. So, your max constraint equation which is basically determining your time period.

$$T_{clk} \geq t_{cq}^{\max} + t_{comb}^{\max} + t_{setup} - \delta$$

This delta is your basically clock skew. Here the defined delta is your clock skew. So, similarly the mean constraint which is for the hold violation. So, that is detected by your basically

$$t_{hold} + \delta \leq t_{cq}^{\min} + t_{comb}^{\min}$$

So, these are the two equations which is useful for doing the timing analysis and here we are using the same concept in a different manner. If you can see we will consider the case 1. This is the first approach.

This is approach 1. So, if you can see you have flip flop 1 and flip flop 2 and flip flop 3 same time you are getting the clock. And here to do the analysis simplified the clock to q delay and the setup time is ignored. Basically, this clock to q delay and setup time is ignored in this analysis. Similarly, the clock to q minimum is also ignored in the analysis.

But what is the time period between the flip flop 1 and flip flop 2 is denoted by the p. So, this is the time period should be greater than equals to 2 nanosecond is the delay because of the combinational delay. What is this combinational delay is coming? 2 nanosecond minus skew is 0 here. So, it is 2 nanosecond. And in the second case between the flip flop 2 and flip flop 3 the time period is should be greater than equals to this is the 6 nanosecond.

So, this comes to be 6 nanosecond and the skew is 0. So, your time period is 6 nanosecond. So, we need to satisfy both the constraints. So, our time period should be greater than equals to 6 nanosecond. This is a concept called useful skew concept. So, what it is doing is that you are giving the flip flop 2 with 0 skew. But the flip flop 1 is getting at time t equals to 2 nanosecond. Because skew is delta in this case is minus 2. And delta in this case is basically 0 minus 2 it is minus 2 in this case nanosecond. And if I use here 2 nanosecond minus minus 2 nanosecond it will be 4 nanosecond. This is a negative skew. In this case it is a negative skew. Between flip flop 1 and flip flop 2 it is basically a negative skew. And between flip flop 2 and flip flop 3 this is a positive skew. So, the time period will be 6 nanosecond minus 2 it is comes out to be 4 nanosecond.

So, this type of skew is basically useful because in the earlier case my time period maximum time period is basically said to be 6 nanosecond in the approach 1. In the approach 2, in approach to my time period is basically equals to 4 nanosecond. So, my since the time period is less my speed will be higher. So, this concept of skew is called useful for improving the speed of the design. So, to avoid zero clocking or set up violation basically we have the same equation whatever we discussed in this equation the same equation was written here is the max constraint equation.

So, here what is there is a your set up time is there. This max is basically this max i, j basically the longest path in the combinational logic actually between the flip flop 1 and flip flop 2 and this xi minus xj, xj minus xi is your delta. Delta is basically arrival time of the flip flop at flip flop j which is xj minus xi. So, this is your skew. So, if you can write this one so this is the final equation. So, xi to consider the worst case condition we should consider the lattice time at which the clock is can arrive at flip flop 1. Since this is the latest time and for flip flop j I have to take the early arrival time. So, I have to check for the early arrival time because I am considering the worst case. So, the clock signal should arrive to the xi basically in late this is a late arrival and here it is a early arrival. So, this will create a worst case conditions.

So, now this equation is coming from the same previous equation whatever I explained here. So, and t clock to q is not taken into account. So, our previous equation is

$$T_{clk} \geq t_{cq}^{\max} + t_{comb}^{\max} + t_{setup} - \delta$$

Your delta is basically xj minus xi. So, here in this thing the t clock is your p

$$P \geq t_{comb}^{\max} + t_{setup} - (x_j - x_i)$$

So, if I simplify this one this will comes out to be

$$(P + x_j) \geq t_{comb}^{\max} + t_{setup} + x_i$$

So, this is this expression and this expressions are the same p is your time period. These two expressions are the same. Now we can do the same thing for the hold. So, hold is basically

$$t_{hold} + \delta \leq t_{cq}^{\min} + t_{comb}^{\min}$$

Now what is happening here is that your

$$t_{hold} + (x_j - x_i) \leq t_{cq}^{\min} + t_{comb}^{\min}$$

So, now if you simplify this then you will get

$$t_{hold} + x_j \leq x_i + t_{comb}^{\min}$$

Here in this expression this one is basically not considered for simplicity. So, these two equations are now same. So, this is a different way of representing the clock skew.

So, here what is the case is that for worst case conditions xi should arrive early because I am checking the same rising edge of the clock to see that whether there is a race condition happening or not. So, the xi should come early and xj should come late. So, this will come early. Now there will be some buffer or something is there so that will come late.

So, this edge should be considered late. So, now this will create a worst case condition for the hold violation. So, till this path we have considered early and this is late. So, now we will discuss about the useful skew problem. So, what we are doing here is that we are applying the linear programming can be used to find the optimal clock arrival time xi at all the sync nodes to satisfy either one of them. One objective is to minimize the clock period that is LP underscore speed actually and the second one is basically we are adding some margin. So, maximize the safety margin LP underscore safety. So, both are linear equations. So, we can use a linear programming solver to solve that. So, here what we have is that we have some constant value like setup and hold and we have propagation delay through the combinational logic is denoted by max i, j min i, j. This is for setup check and this is for hold check between the two flip flops or the latches. The minimum source sync delay is basically t min. So, then we can formulate the two problems one for the speed one for the safety. The first one is for the speed problem LP speed actually. What we are doing here is that determine the clock arrival times xi. So, what is the objective here is to find the clock arrival time xi for all syncs to minimize the clock period p.

So, minimize the clock period p. Why you are minimizing the clock period? Because I want higher speed. So, the clock period is minimized and such that we can also find out the all clock arrival times at the input of the all the sync nodes. So, here we have two constraints. One is related to hold this is related to hold and this is related to setup. For all nodes all i, j for all i, j. So, now we have one xi which is greater than t min for all i. So, this can be solved using a linear programming solver. Now we have a second problem which is called LP underscore safety. What it does is that it also the same objective finds the arrival time of the arrival time xi for all the sync nodes actually. All the sync nodes are the flip flops and the latches and latches in the design. So, our objective is to minimize the clock period p. If your clock period will be minimized then that will basically maximize our speed of operation. So, you have this term is added. These two terms are added in the expression. So, there are several clock routing algorithms are there. We can divide that into two categories. One is called a single phase clock. Whenever we are using single phase clock in the system we have multi phase clock then you have different algorithm. In case of single phase we have a delay based. We have basically path length based algorithms are there. So, the delay based algorithm will be the exact zero skew method and the second is that methods of means and medians we will discuss.

Then we will discuss weighted center and geometric matching. So, these are the three basically the path length based optimization algorithms are there which is used for a clock tree synthesis. Basically, we have multi-phase clocking is also there. A square tree based routing is also employed for multi-phase clock. Now we will discuss with H tree based algorithm. This H tree is basically a self similar fractal structure. Fractal means it is very systematic structure and it is a detailed structure actually, geometric structure with exact zero skew due to its symmetry. So, it has exact zero skew due to its symmetry if I do not have any kind of process variation. If the process variation will be there obviously there will be some variation in the skew if there is no process variation because of the structure of the H tree it will produce you exactly zero skew. It is frequently used for top level clock distribution and cannot be employed directly for the entire clock tree due to some of the reasons because we have blockage.

Some of the places there is blockage the router cannot route. So, because of the blockage basically this entire symmetry cannot be maintained. Irregularly spaced sync blocks the flip flops may not be placed in the design in a regular manner and sometimes excessive routing cost because it has a symmetry it also encounter excessive routing costs. Let us discuss this H tree based algorithm. In this example we have four sync nodes these are the sync nodes actually. These are four sync nodes are there. Now how can we do the clock tree synthesis in this case? So, we have these are the locations of the pins four pins and in a routing plane length is six actually this is 1, 2, 3, 4, 5, 6 and the width is also 6 1, 2, 3, 4, 5, 6. Then we have a clock entry point P0. So, this is the clock entry point P0. So, the clock will enter to the design at this point.

This is the clock entry point. Now what we have to do is that in the H tree the P1 is connected to P3 this P1 is connected to P3. Now we have P2 is connected to P4 using a two vertical segments. So, now we will find the midpoint middle point of this vertical segments. This is the middle point of the vertical segments. Now what we are doing that we will find the midpoint P1, 3, 1, 3, P2 for 5, 3.

These middle points are also called the tapping points actually. Here we do the tapping for the clock tree because we need to connect to that point. So, that is why it is called tapping points. So, now we will create another horizontal segment. This horizontal segment will create an exact H in the layout. So, that is why it is called H tree based algorithm. So, now what we have to take the middle of that horizontal segment. Now we will finally connect the clock entry point P0 to that midpoint of the horizontal segment. So, we can extend the same approach to create several H tree in the design.

If it is a two level H tree design algorithm. So, if you can see here you have a very interesting points here like let us say I have a K there are two factors are there. K is one factor, n is another factor. How they are related? K is the number of levels in the H tree. In the previous example number of levels because how many H are possible. This first one, this one is one level, this one is one level and these are the second level.

This is a second level. So, K in this case is 2. So, if K is 2 how many sink nodes is possible? Sink node means how many number of flip flops it can give the clock signals. So, here it is 4 to the power K is n which is basically number of sink nodes. So, in this case it will be 4 to the power 2 which is 16. If you can see in the previous example here it is 1, 2, 3, 4, 5, 6, 7, 8 similarly 8 in other direction. So, here 8, here 8 so together it is 16 sink node or flip flop or latches that are connected to this H tree which will provide you zero skew signal to the flip flops. And the total overhead here is basically total wirelength overhead is basically 3 into the root over of n by 2 and this H tree construction has been extensively used for clock routing in regular systolic arrays also. It is very popular algorithm H tree based clock tree synthesis algorithm is very popular in industry. In this lecture we discuss about the problem statement of the clock tree synthesis then we discuss about the H tree based clock tree synthesis algorithm.

Thank you for your attention.