VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Week 03

Lecture 16

STA considering OCV and CRPR (Hold check)

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss about static timing analysis considering on-chip variation and CRPR for min path or hold check. So, the content of this lecture includes basically the launch flip-flop and the capture flip-flop, and we have three different cases. One is without variation. So here what we are doing is that we are doing mean timing analysis or the hold check without variation that is the first case. Then we will do min timing analysis or the hold check with on-chip variation or OCV. And the third category is basically how my timing analysis, the min timing analysis or the hold check will change with OCV plus CRPR. So, we will discuss all the three types of timing analysis with one example. So, we will first of all we will discuss how my this sequential circuit will look like for a minimum timing analysis. So static timing analysis or minimum timing minimum timing analysis. This is also called to check for hold violation in any path. So now first we will discuss this how I can derive the equation for the hold path. So, this is my launch flip-flop. This is my capture flip-flop. So, this is launch flip-flop 1. This is capture flip-flop. This is flip-flop 2. So, this is D. This is Q. This is D. This is Q. I have some combinational logic is there. So, since I am doing minimum path timing analysis, I need to find the short path or min path in the combinational logic. Now in the previous hold analysis, we consider the clock tree as a ideal clock tree. But here we are not considering the clock tree as a ideal clock tree.

We have clock buffers introduced inside the clock tree. So here you have a clock source is there. This is a clock source. Now that is going through some buffer. Why the buffers are needed? To make the input slew because the clock is going to multiple flip-flops. So, if the clock is going to multiple flip-flops, it is driving a large amount of capacitance. So, in order to make the rise time same, we need some buffers in the clock tree. Otherwise, your clock signal will be very bad if you do not have buffers in the path. So, this is the clock pin. So, I have, this is the common point. Now this is the basically sequential circuit

I need to analyze for the min timing. So, I have two points, two things is there, two definitions I need. One is data arrival time. So, data arrival time for the hold check and the setup check will be different. So here we are doing the data arrival time analysis for the hold check. So, for hold check, my data arrival time data arrival time DAT is basically launch clock path plus mean data path. I need to consider the minimum path; this is very important point. I need to consider the minimum path in the combination circuit. If I have a clock to Q delay is mentioned, I need to take the t clock to queue minimum. So, this basically the mean data path whatever it is written, it is basically t clock to Q minimum plus t combinational minimum.

I am using one term here for this. So now I have data required time. So, data required time DRT is basically capture clock path plus t hold. So now for hold check what is my constraint? For hold check, these are the arrival time for hold check. Your data arrival time should be greater than equals to because that path should take more time compared to data required time. This is very important. Your data arrival time should be greater than data required time to avoid hold violation. So, this is my, then the slack here is basically your data arrival time minus data required time. So, there are two cases here. So, if I have slack is given, there are two cases.

Case 1, slack is positive. What will happen if your slack is positive? Basically, your data arrival time is greater than data required time. Data arrival time is greater than data required time means that your data is arriving late compared to your clock edge. Data arriving late compared to your clock edge, so slack is positive. So, there is no hold violation, implies no hold violation. This is very important. This has always confusion with the hold violation. So, slack is positive means data is arriving late to the D pin, this pin there is no hold violation. But case 2, your slack is negative implies that your data arrival time is small compared to data required time. Means what is happening is that in the same clock edge your old data whatever is there, it is sampled by the new data which is not correct. So, you have hold violation. So, you have hold violation. Basically, we are checking the hold violation in the same clock edge. But whenever you are checking the setup violation, you are using the two-clock edge. So, this slack is negative means data arrival time is less than data required time, you have a hold violation.

So let us consider a special case. So, the special case is that there is no buffer in the clock path or clock tree. So, your launch clock path delay will be 0 and capture clock path delay will also be 0. Now if that is the case, your data arrival time should be greater than equals to data required time. So, data arrival time equation, the launch clock path this will be 0. Then you will have min data path only. And data required time we have this capture clock path is there. So that will be 0. Now we have t hold. So, this min data path is nothing but your clock to Q delay minimum plus t combinational minimum greater than equals to t hold. So, we discussed this equation earlier. So, this is a special case equation because this is an ideal condition equation because there is no buffer in the clock tree. But in general,

whenever you implement a chip, we have buffers in the clock tree. We need to consider those buffer and when we do the timing analysis. So now we will take one example and explain how my basically timing analysis will happen with clock tree with buffers. So, I have a flip-flop here. So, this is a launch flip-flop. This is capture flip-flop. Then I have min delay path. This is the min path I am considering because I am doing minimum timing analysis.

This is clock. This is clock. This is D. This is Q. This is D. This is Q. Now I have buffers in the clock path because I am not, I am considering the actual condition inside a chip, and I want to do the timing analysis considering the clock buffers.

So, I am showing in this example one-one buffer, but in reality, that can be multiple buffers. We need to consider all of them. So, this is your clock source, and this is the delay of this one is 0.4 nanosecond. This is 0.6 nanosecond, and this delay is basically the min data path delay. Whatever we have considered here is basically 1.5 nanosecond. Then this delay is basically 0.9 nanosecond. Now your t hold is basically 0.9 nanosecond. So now I have three cases. I will check how the things are happening.

Case one without variation. So, the launch clock path delay is basically 0.4 plus 0.6. This is 1 nanosecond from here to here. This is your launch clock path. Now I have min data path. Min data path is basically 1.5 nanosecond. Now I have captured clock path. The capture clock path is basically is your 0.4 plus 0.9 which is basically 1.3 nanosecond and your t hold is already given which is 0.9 nanosecond. Now data arrival time is found out to be your 1 nanosecond plus 1.5 nanosecond. It comes out to be 2.5 nanosecond. Data required time is basically 1.3 nanosecond plus 0.9 nanosecond which is comes out to be 2.2 nanosecond. So, my slack in this case, slack is basically data arrival time minus data required time. So, this is 2.5 minus 2.2 basically 0.3 nanosecond. Your slack is positive. Conclusion is that slack is positive, hold requirement is met. Now there is no issue, there is no hold violation without variation. Now we have case 2 where we have with OCV, on-chip variation. With OCV how my timing will change. If I consider the variation, local variation inside the chip. This OCV stands for local variation inside the same die or chip. So let us say I have to set some derating factor. These derating factors are taken from the foundry or doing basically Monte Carlo simulation or something.

So basically, set timing derate - early. This is basically 0.85 and set timing derate late. So, this is 1.1. Now set timing derate early 0.9 - cell check. Now I have launch clock path. So, this cell check is used for hold case because I am doing minimum timing path. So, launch clock path delay, launch clock path is basically 1 nanosecond multiplied by 0.85. Why 0.85? Because I am considering minimum path. So, this one should be from here to here, from here to here is considered minimum or early, early in the derating factor is was early and from here to here it is max or late, max or late which will create a worst case condition. My data is arriving early, my clock is reaching late to mimic a worst-case

condition in we need to consider the worst-case delay. Now so this value is comes out to be 0.85 nanosecond. And min data path is 1.5 nanosecond multiplied by 0.85, okay, is 1.275 nanosecond. Then the capture clock path delay should be 1.3 nanosecond multiplied by 1.1 because I have to take the max or basically late for the capture clock path. So, it comes out to be 1.43 nanosecond. So, this will be early, this will be late. Now data arrival time is basically 2.125 nanosecond and data required time is found to be 2.24 nanosecond. Then the slack in this case is basically data arrival time minus data required time if I do the subtraction it is comes minus 0.115 nanosecond. So, which implies the slack is negative implies just hold violation, okay. So, the case 2 with OCV we have hold violation. Now we will consider case 3. So, the case 3 is basically with OCV plus CRPR. The CRPR is already defined in the setup case. So, we are just finding the CPP, the common path pessimism factor whatever it is there.

So, this is 0 point which is the common path here. The common path is basically from here to here till this middle point is your common path. So, the delay is 0.4 which is taken as min or early when I am considering the launch clock path. So, this is considered min but in case of capture clock path it was taken as late. So, one buffer it is there in a single same chip how can it can behave two differently if the same buffer is there inside the chip. So, this is something mistake we did in the OCV calculation. We need to correct that using CPP. So how much error we did in the OCV is basically we calculate that value using CPP. So, 0.4 multiplied by 1.1 minus 0.4 multiplied by 0.85 which is comes out to be 0.1 nanosecond. So earlier the slack is negative, and the value was what is the slack value? This is the slack value previously there. So, this is the slack. So, this minus 0.115 then the plus CPP. So, the plus CPP is basically it will this will 0.1 nanosecond CPP is 0.1 nanosecond. It comes out to be minus 0.015. So, it is reduced compared to the OCV but still it is 15 picosecond basically a negative slack is there. So, hold violation is still there but it is reduced compared to the analysis with OCV. Hold violation is there. Hold violation exists. It can be solved using buffer insertion. It can be solved using buffer insertion and even if it change any time period you cannot solve the hold violation. So, your hold violation that main important point is that hold violation or hold check or hold constraint is independent of the time period of the clock. Even if we change the time period your hold violation problem cannot be solved. So, this buffer whatever we will insert it should be inserted in this path data path.

So, we need to do the buffer insertion in the data path. So, there is one more important point here is that your slack whatever we defined earlier the slack is basically what is the slack definition. Your slack is basically in this case is data arrival time minus data required time. So, this slack is the basic definition of the slack. So, the case 2 whenever I am doing basically case 2 whenever I am doing within OCV let us say the slack is basically slack OCV is the data arrival time minus data required time. This is for case 2. The case 3 what is happening? We are doing OCV plus that CRPR where the slack the modified slack

actually we can say the slack for OCV plus CRPR is what is the case is your data arrival time minus data required time plus CPP. The CPP whatever the term that is added with your data arrival time and that is added with the data arrival time to find out the actual slack in the path. So, the new slack is data arrival time plus CPP minus DRT. Because that path was taken mean data arrival time path was taken mean for the common path actually for this path from here to here. So that factor should be added in the data arrival time to make the actual calculation of the slack whenever you are doing the OCV plus CRPR.

So, the hold is very crucial thing and many of the designs if you do not calculate the buffer delay properly or interconnect delay properly, we have more chance of hold violation. We need to do the mean timing analysis very carefully because if you have hold violation inside a chip it cannot be modified after the manufacturing. But if you have a set of violations which can be solved by slowing down the clock speed, but hold violation cannot be solved basically without by changing the clock frequency. So, you should put more attention while checking the hold in all the paths in the all the short path in the design all the short path in the design.

So, in this lecture we discussed about the minimum timing analysis. So we consider three different cases one is without variation the case one the first case is without variation and the second case is how to do minimum timing analysis considering the local variation or the on-chip variation and the third case is basically how to do timing minimum timing analysis considering the OCV or on-chip variation plus the common path the common clock path which is creating the error in our timing analysis that considering OCV plus CRPR we did the timing analysis. We discussed three different timing analysis for the hold check the or minimum path minimum timing analysis. So, this is very important for doing the hold violation checks in the short path. Thank you for your attention.

Thank you. Thank you.