VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Week 03

Lecture 15

STA considering OCV and CRPR (Setup check)

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will study about Static Timing Analysis considering On-chip variation and Clock-reconvergent-pessimism removal, CRPR. So, the content of this lecture includes what is the launch flip-flop, what is capture flip-flop and then we will discuss about three different types of timing analysis. One is called max timing analysis or the setup check considering without considering any process variation. And the second type of timing analysis where we do the max timing analysis or the setup check considering the on-chip variation. So, we will consider the on-chip variation and what is its impact on setup check, we will discuss that in detail. Finally, we will discuss about the max timing analysis considering the both on-chip variation and CRPR. So, in this lecture, first of all we will discuss about what is launch flip-flop and what is capture flip-flop. We will discuss this with the example itself. So, we have a sequential circuit, we have a sequential circuit here the flip-flop, two flip-flop, flip-flop 1 and flip-flop 2, this is the clock.

So, this is your launch flip-flop which is basically launching the data, and this is the capture flip-flop which is capturing the data. Now, in the previous lectures, we discussed that we have an ideal clock tree means there is no buffer in the clock tree, but in reality, there is a buffer. So, we will introduce buffer in the clock tree. So, here we have one buffer, then we have another buffer which is going to this clock, this is a clock pin, then we are going then there is another buffer going to the capture flip-flop. So, this is clock, so this is D pin, this is Q pin, this is D pin, this is Q pin, in between we have combinational delay. So, here we are doing max timing analysis. So, we are considering the maximum path in the combinational logic. So, this is my circuit diagram, I want to do the timing analysis of this circuit. So, here we have this is the launch flip-flop, the first one flip-flop 1 is your launch flip-flop and flip-flop 2 is your capture flip-flop. So, in one clock edge launch flip-flop will sample the data, in the second clock edge the capture flip-flop will sample the data. So, now we have a few terms there, if you can see here this is a common point, this is a

common point and this is the clock source, this is clock source clock actually and this is basically you have till this point is common to both the clock paths. So, you have there are few delay terms are involved, one is called from here from this point till this point you have a launch clock path. Now from here from this point, so this point to this point is your launch clock path and from this point till this point is your max data path. Now you have one more delay component is there till this point is the capture clock path. So, these are the three parameters launch clock path, max data path then captures clock path. So, these are the three definition we need to remember. So, now I introduce two things one is called data arrival time, data arrival time which is basically launch clock means how much time it will take to take the data to available at the input of the capture flip-flop that is your launch clock path plus max data path. Then you have basically data required time, data required time which is basically clock period plus capture clock path minus t setup, t setup of the flip-flop 2. For setup time check, basically your data arrival time is less than equals to data required time. So here basically your data arrival time should be less than equals to data required time for successful capture of the data at the flip-flop 2 otherwise there will be setup failure.

So, basically if I look into this data arrival time, your data arrival time should be less than equal to data required time. So, your launch clock path plus max data path should be less than equals to clock period plus capture clock path minus t setup. So, this is the basically the same definition. So, this is the general case, this definition what is there is the general case, in general case. General case means that considering the delay of the clock buffer, considering the delay in the clock buffers in the clock path, but we have a special case is also there. The special case will be, what is the special case in this case? The special case is that if there is with clock buffer here, general case is with clock buffer in clock tree and the special case is no buffer in clock tree. If there is no buffer in the clock tree means it is an ideal condition. So, this is basically ideal condition. If ideal condition what is the my launch clock path delay will be? Launch clock path delay will be 0 and capture clock path delay will also be 0. So, if this is the case my equation will be, my this equation 1 whatever it is there, the equation 1 will be max data path should be less than equals to clock period minus t setup. Now this max data path has two components, one is called t clock to Q delay whatever we discussed earlier max plus t combinational delay max less than equal to clock period is T clock minus t setup. So, this implies that t clock to Q delay max plus t combinational delay max plus t setup should be less than equals to T clock. So, this is my equation whatever I discussed earlier. This is my equation of whatever I discussed earlier. I will put this one in a box because this is special case equation what we discussed earlier when there is no buffer in the clock tree.

So, this is the case. Now this is the general case actually. We will put these two equations in a box, equation 1 and equation 2. Equation 2 is basically the special case of equation 1. Now we will define few more terms here. So, we will define one term called slack actually.

The slack, what is the slack actually? Actually, how much delay margin is available for the data path actually means how much delay margin we have basically that we determined by the slack. How much delay margin we have. So that one we need to find it out. So how I can find the slack? So, slack should be data required time minus data arrival time. If my slack is 0, my slack is equals to 0 imply that your data required time and data arrival time is same. So, your design, what is the conclusion here? This is basically the case 1. So, the conclusion is that your design is basically we have minimum clock period, we have running the design with minimum clock period or maximum clock frequency. Because there is no margin there. We are running at very high speed the design, when slack is 0 your design is running at very high speed.

And the case 2 is that if my slack is basically positive. So, if slack is positive means I have extra margin. So now if I have extra margin means data is arriving early. What is extra margin? Data is arriving early to this point and waiting for the rising edge of the clock means the your data arrival time basically is smaller compared to data required time. So, the amount is smaller, the data is waiting for the clock is that much of time.

Let us say for example, I have a class at 11 o'clock and the students will come 5 minutes earlier. They have to wait for 5 minutes before the class starts. So, the data is basically that 5 minutes is your extra margin, and you have that extra margin is there for the student to wait for the teacher. So that is called your 5 minutes margin. So similarly, here if a data is coming early here means you have basically your slack is positive then what will happen is that the data has to wait till the clock edge will arrive.

We have extra margin and it is safe there is no issue then but you are wasting some amount of time. Case 3 is that if the slack is negative. Slack is negative means data is coming late. Your data is arriving late to the flip-flop too. If data arrives late means what will happen? Your setup violation will happen. So, it will lead to setup violation. So now we will go to an example and see how these things are happening. So basically, I have a circuit I have a basically this launch flip-flop and capture flip-flop then I have to check that how I can do the timing analysis here. The flip-flop 1 launch flip-flop flip-flop 2 is your capture flip-flop. I have a clock this is clock this is clock this is D Q this is D Q then I have combinational logic. Now I have to some parameters are given here. So, this delay of this buffer is 1.5 nanosecond, and this is 0.9 nanosecond. Now I have the delay of this one is 1.1 nanosecond. My delay from this clock to Q is and is let us say together this from here to here is basically 6 nanoseconds. And my clock period T clock is given I will write here T clock is given as 7.2 nanosecond. And here my setup time t setup is also given which is 0.5 nanosecond. Now I will do the all the calculations. So, I will see for three cases. The case 1 is basically without variation. So, I have the launch clock path which is basically 1.5 nanosecond plus 0.9 nanosecond which is equals to 2.4 nanosecond. The max data path here it is combined it is 6 nanoseconds. Now you have capture clock path. The capture

clock path is basically 1.5 plus 1.1 nanosecond, it is 2.6 nanosecond. Now t setup is already given t setup is given is basically 0.5 nanosecond. So, data arrival time will be launch clock path plus max data path 2.4 plus 6 it is 8.4 nanosecond. Then data required time is basically 7.2 which is your basically time period T clock. This is your T clock actually plus capture clock path. Capture clock path is 2.6 minus 0.5. So, this will comes out to be basically 9.3 nanosecond. So now I have space here. So now the slack is basically data required time minus data arrival time which is comes out to be 9.3 minus 8.4 which is basically 0.9 nanosecond. So, slack is positive setup requirement is met. So, slack is positive and setup requirement is met is satisfied. So, this is the conclusion in this case. So, this is the conclusion. The conclusion is in this box. You say slack is positive and setup requirement is satisfied. Now we will go to the case 2 with on-chip variation. The case 2 with OCV. OCV stands for on-chip variation. So, on-chip variation is the variation inside the same die. So, it is a local variation actually local variation. This is the local variation inside the same die. So now I have a few parameters need to be specified here. Set timing derate - early. So, it is 0.85. Set timing derate - late. So, this is 1.1. Set timing derate - late 1.05 cell check. Cell check is applicable for setup time and hold time for the flip-flop level. So, this is cell checks are applicable for setup and hold times. This is applicable for only set up t setup and t hold. Since we are doing max timing analysis it will only applicable for setup time.

Now we have basically two paths. One is data arrival time and data required time. If I go here, so this path starting from this point let us say from this point to this point is basically your data arrival time DAT. So, the DAT should be considered max to make it worst case violation. You check for worst case from this star the clock source to this point should be considered for the worst. Basically, the delay should be max, max delay should be considered here or the late. You can use the term late. And the other case that from here if I have two star from here, from here to this point, so this point that path should be basically earlier minimum from here to here it should be minimum or early for worst case. For worst case analysis, your capture clock path should be treated early or minimum and the launch clock path and the max data path that should be treated as maximum or late. So, we will do those analysis here. So here we will do those analysis.

The launch clock path should be now it is 2.4 nanosecond multiplied by 1.1. So, the derating factor 1.1 because it is considered late. So late or maximum. So, it is 2.64 nanosecond. Now I have max data path. It should be considered late. So, it is 6 multiplied by 1.1 it is 6.6 nanosecond. Now I have capture clock path, capture clock path. The capture clock path should be 2.6 this is nanosecond multiplied by it should be treated early because that should be minimum for a worst case 0.85 it should comes to be 2.21 nanosecond. Now my setup for setup I need to use this third parameter t setup. So, t setup should be 0.5 nanosecond multiplied by 1.05. So, it is 0.525 nanosecond. So now I have to find data arrival time. If I have to do data arrival time 2.64 plus 6.6 it comes out to be 9.24

nanosecond. Now data required time is basically 2.21 minus 0.525 this is coming from capture clock path, and this is coming from the t setup these are the things plus your clock period. Clock period there is no variation is introduced because it is a designer's parameter there should not be any variation. So, this is 7.2 nanosecond. So, if I substitute if I calculate this one it comes out to be 8.885 nanosecond. My slack in this case the slack is basically data required time minus data arrival time. So, it is comes out to be -0.355 nanosecond. So, you have conclusion is that you have setup violation. We have setup violation. Now I have case 3. So, the case 3 is basically takes into account OCV on chip variation whatever I discussed earlier plus CRPR. What is CRPR? CRPR is called clock reconvergent pessimism removal. So, the clock reconvergent pessimism removal. So, what is the point here is that we have a common path here these points from here to here. So, this point this path if we can see we have calculated whenever I consider the launch clock path, I multiplied that path with one point consider that one as late. So, for launch clock path it is considered as late, but for the capture let say this launch clock path it is considered as late and for capture clock path this path is considered as early, but that is not correct. So, this is considered late for the launch clock path early for the capture clock path. So that is not correct, because it is inside the same chip, we cannot consider two variations here. So, we will introduce a term called CPP. So, we will introduce a term called CPP. What is the CPP? Actually, CPP is basically, CPP is the delay difference, delay difference along the, along this common portion, along the common portion of the clock tree, ok, due to the difference in different de-rating for launch and capture path. So, this is the CPP actually. So, basically the CPP we need to find out because we are using two different derating factors. This derating factor is different for this common path, which is not correct in the previous analysis, we need to make it correct considering this CRPR actually. So, your CPP is basically, your CPP is basically your latest arrival time at the right common point minus earliest arrival time at the right common point, ok. So, these things we need to calculate. What is this value? This value is 1.5 into 1.1, ok, this is for the late, minus 1.5 nanosecond. So, this is 1.5 nanosecond multiplied by 0.85, this is early case. So, this difference is comes out to be, this difference is comes out to be 0.375 nanosecond. So, this much of error we are doing. So, we need to add that one, we need to add that one in the previous slack whatever you got in the case 2. The case 2 slack is what? The case 2 slack whatever you got it is basically this much. We need to correct that one, we need to correct that one. So, that slack whatever it is we got it there. So, minus 0.355 nanosecond plus 0.375 nanosecond which is basically 0.02 nanosecond which is 20 picoseconds. Now, the slack is positive, slack is positive, hence implies no setup violation, ok. So, no setup violation, this is the conclusion. The main point here is that slack definition, the slack is basically data required time minus data arrival time, ok. So, here what we are doing is that whatever the slack we got, slack for both OCV plus CRPR is equals to slack we got due to OCV plus the CPP, ok. So, what the CPP is doing here? CPP is added with the, so added with the which one? This is a plus term, this will be added with data required time. So, we

are doing a error in the data required time while we are doing this common path as two derating factors, but we are correcting that one by adding this CPP in the data required time. So, since we are adding in the data required time, then the common path is treated with the same derating factor, with the same derating factor and the timing analysis is accurate. In this lecture, we discussed about static timing analysis for setup check in three different cases. One is without variation and the second is with OCV or on chip variation and the third one is the OCV plus CRPR. How my slack changes in all the three cases was explained with an example. Thank you for your attention.

Thank you very much.