VLSI Design Flow: RTL to GDS

Dr. Sneh Saurabh
Department of Electronics and Communication Engineering
IIIT-Delhi

Lecture 28
Static Timing Analysis- I

Hello everybody. Welcome to the course VLSI Design Flow RTL-2 GDS. This is the 22nd lecture. In this lecture, we will be discussing static timing analysis. So static timing analysis is a very important component of VLSI design flow because it is used throughout the design flow and it ensures the timing safety of our circuit. So we will be discussing static timing analysis in four lectures.

So in this lecture, we will be looking into the basic concepts related to static timing analysis and in subsequent lecture, we will be looking into the mechanics of static timing analysis, meaning how static timing analysis is done and then we will be looking at some advanced concepts related to static timing analysis and then we will be also be looking at constraints or meaning how we specify the required behavior for our design in terms of its timing. Now let us look into what is done in static timing analysis. So the most important purpose of static timing analysis is to ensure that the circuit is in a valid state at each clock cycle. Now what is meant by valid state and how does an STA tool ensure that the circuit is in the valid state at each clock cycle.

We will be looking at it in the later part of this lecture. Then the second important task that is carried out by the STA tool is to ensure that the given design is capable of working at a given frequency. So how do we specify frequency for an STA tool? So we specify it in the constraints file which is typically written in Synopsys design constraints format or SDC format. So in the SDC file we as a designer specify that at what clock frequency our design is expected to run and the STA tool ensures that the design can indeed work at the given frequency in the worst case. So what it means is that an STA tool does not try to come up with a frequency at which a design can work and run.

So these are important things that need to be understood that the purpose of STA is not to find a frequency at a design at which a design can run. But the purpose of STA is to ensure that at the given frequency it is safe to run a design and to ensure that an STA tool will take a pessimistic view wherever required. And the third important thing that an STA tool does is that it ensures that the setup time and hold time for all the flip flops in our design are met . So in the last lecture we had looked into the fact that for each flip for

a flip flop there is a forbidden window around the clock edge where the data signal cannot change. So an STA tool will ensure that the constraint of setup and hold time for each flip flop in a design is indeed met.

And to ensure or to while performing these three tasks what the STA tool does is that it takes a worst case view wherever required. So the purpose of STA is to ensure the timing safety of our circuit . To ensure the timing safety, sometimes an STA tool can take a pessimistic view and it can throw some unnecessary violations. But thus an STA tool will never suppress a real violation wherever it can throw some extra violations but once a design is passing then STA tool means that an STA tool says that where there are no timing violations then the design is indeed safe to operate as per its timing requirement. So while ensuring that these three import while performing these three tasks the static timing analysis does not apply any test vectors .

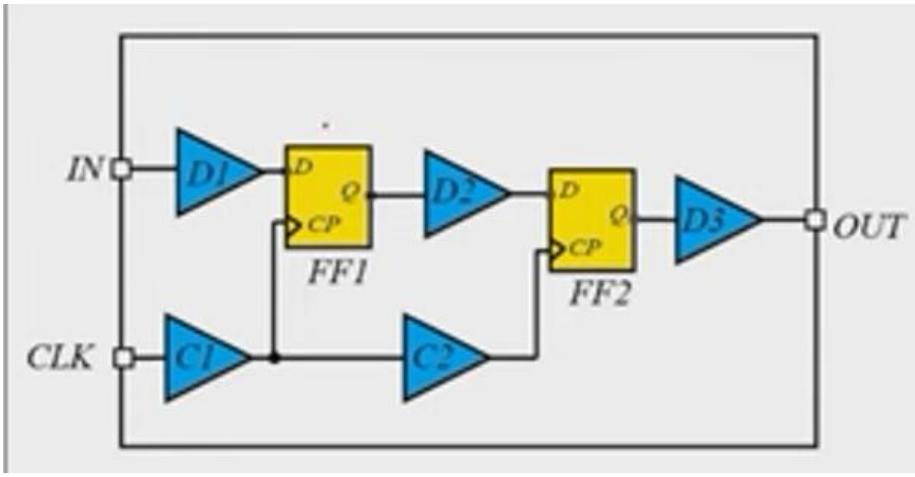
And in that sense it is a static kind of analysis, it is not a dynamic timing analysis it is not that it is applying a test vector and then seeing the response and then checking whether it is timing safe or not. So what the tool does is that it only or an STA tool does is that it only concentrates or focuses on the worst case behavior of the design. And in the worst case most of the input vectors that we can apply to our design do not contribute to the worst case scenario and therefore we can safely neglect those test vectors. So the static timing analysis does analysis without using test vectors. So while doing its analysis it considers inputs inputs and logic one values at the inputs of the gates such that the worst case scenario is automatically captured.

So an STA tool ensures that the design will not have set up or hold violations for any test vector why because if it is verifying it is taking pessimistic view throughout our design then the the one which was considered by STA will actually be a superset of whatever test test vectors can be applied to our design and in therefore the if the tool is able to establish the timing safety of a design for the worst scenario that it has or the pessimistic scenario that it has considered it will automatically be timing safe for the other situations or all the possible test vectors that can be applied to our design and therefore we can say that that design is safe for any test vector. So we can differentiate between what is different or so we have in the earlier lectures we have looked into another verification technique and that was simulation. Now we can differentiate or we should understand the differences between the simulation and static timing analysis. So in a static timing analysis we do not use any test vector we assume the pessimism pessimistic view of timing everywhere and based on that we ensure the timing safety and in simulation what we do is that we apply test vectors zeros and ones at the inputs and then see its response with respect to clock cycles and then if we have model delay in our circuit then we with respect to delays and so on. So for simulation test vectors will be

required for STA no test vector is required and then for the STA does not check the functionality meaning that it does not consider that if I apply zero a sequence of zero whether the sequences of zeros and one produced at the output that is correct or not that purpose of STA is not to verify the functionality .

It totally ignores the functionality details of or Boolean function detail of the circuit which works; it only considers the timing arcs and based on that it does that or does the timing analysis of our circuit. While the purpose of simulation is to check the functionality meaning that if I give a sequence of zeros and ones at the input then whether we are getting the sequences of zeros and ones inside in or from the circuit or not that is what the purpose of simulation is. So here we are verifying only the timing in the timing in a static timing analysis not verifying the functionality of our design . So in STA what is the mechanism ? In STA the analysis is performed taking a pessimistic or pessimistic view of delays and other attributes of the design and simulation is done based on the specified test vectors and delays. So these are some of the important differences between STA and simulation.

Now before moving further let us understand what is expected out of a synchronous design and what are the valid states of a synchronous design. So let us consider this circuit.



So in this circuit there are say two flip flops for simplicity let us assume that these flip flops are ideal meaning that the clock to q delay is 0 the setup time of this flip flop is 0 and the hold time is also 0 . So we are assuming that our flip flops are ideal. And then we are saying that there are buffers in our circuit so there are buffers on the data path d1, d2, d3 and there are two buffers on the clock path c1 and c2.

These buffers have some delay for simplicity. We have just taken a buffer . So these buffers can have delays and suppose we apply input at the port on the . So we are applying input at the port in and what are the inputs the input in the first clock cycle is a in the second clock cycle is b then in third c and in fourth we have d . Typically these numbers the input that will be coming in will be a digital value meaning that it will be either 0 or 1 . But for our analysis I have represented using a, b, c, d so that we can easily understand what is happening in our circuit.

So we have given different identifiers to signals at different clock cycles . Also assume that the initial value at the point at the output of this flip flop 1 q is p and the output value at the output of the second fip-flop ff2 is q . Suppose these were given initial conditions and this sequence of inputs were applied at the input port on the . Now the state is defined by the combination of values at the q pin. Let us assume that we are defining states by the values that are there at the q pin for these two flip flops.

So what is the initial state? Initially this q pin had value p this q pin had value q so the initial state is pq . So the initial state is p. Now let us understand the behavior of the circuit in different clock cycles for various cases of delay of the buffers . So now we are saying that now what will happen if this is a perfectly synchronous circuit what will be happening is what will be the state of the system meaning what will be these values in different clock cycles . Will take different delays of d1 d2 d3 c1 c2 and c different scenarios.

So let us take the first scenario in which we say that the delay of d1 d2 d3 has some finite values but it is less than one clock period . So d1 d2 d3 are having some delay but less than the one clock period and the delays of c1 and c2 are negligible. Let us assume that the delay of these two are negligible so we can remove them from the circuit. So equivalently we can represent the circuit like this. Now in this case suppose we applied the input as a b c d at the input in what will be the stage at these clock cycles .

What will be the value at these clock cycles at ff1 q and ff2 q . So suppose what will happen is that if a was here then a will come and propagate here at d pin and whenever the clock edge will come we will get a at this point . So this will come at this point. Now once this comes here it will propagate through d2 and delay of d2 is finite less than one clock period. So when in the next clock edge this by that time this a will reach here at the

d pin and in this third clock cycle a will go to the output .

Similarly b will go to this and then in the next clock edge it will go this and this whatever was p initially here here we assume that at 0 clock cycle we had p at this point. So that will go and propagate to d and in this clock cycle will have p. Similarly this c will come here . So we will get something like this . So the value if d1, d2 and d3 have some finite delay less than the clock period then the value at the q pin will be as shown in this field .

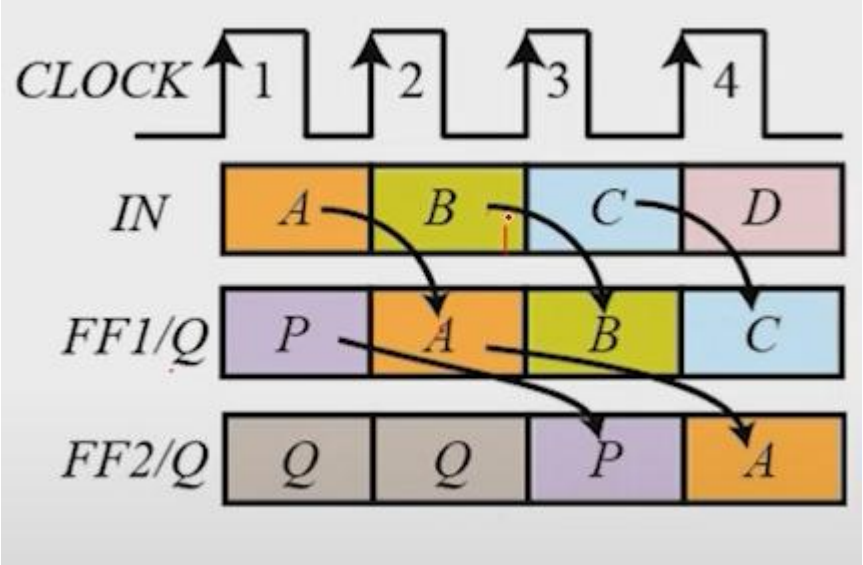
Then what are the valid states that will be there for this circuit. So the valid state will be pq initial state then ap then ba then cp. So these are the valid states . So these are the valid states of a synchronous circuit. This is how a synchronous circuit behaves and these are the valid states.

Now what can go wrong in a synchronous circuit? Let us look into that. So the first first case can be that the delay of this d2 becomes very large . So now let us assume that c1 and c2 are negligible we can remove them and the delay of the circuit element d2 is more than 1 clock period. This is more than 1 clock period and less than 2 clock period and d1 and d3 are minimum. There are very small delays on this . So equivalently we can draw something like this.

So we have d2 which is taking more than 1 clock period and d1 and d3 are having small delays. Now if we apply the same stimulus at the input a input in the ab so the same thing what do we expect . So at this FF1 q there will be no change as similar to the synchronous behavior meaning that this is a small delay so whatever will be there in the next clock edge it will be captured. So whatever was at a it will go to in the next clock cycle it will go to FF1 q at this point and then whatever was at whatever comes in the second clock cycle it will become available in the clock it will come at the output in the third clock cycle at FF1 q and in the fourth clock cycle and so on. So there will be no change at the output of FF1 q but what will happen at FF2 q at that point what will happen is that if there was a value p here it will propagate through d2 and it will take more than 1 clock cycle to reach the point d.

So it will not be captured in the second next clock edge but in the next to next clock edge and therefore it is not captured ideally it should have been captured at this point but instead of being captured here it will be captured in the next to next clock cycle why because the delay of element d2 is more than 1 clock period but less than 2 clock period . Similarly a will not be captured in the next clock cycle but in the next to next clock cycle . So the values at FF2 q will get disrupted so the states that will now will get disrupted as pq and then aq bp and ca . So these are the states that will be getting in our

circuit but in the last slide we saw that what are the value states these are ap ba and cb for the synchronous circuit . So the states that we get in our circuit are different from what we wanted as in a synchronous circuit and therefore the synchronicity of our design is lost.



The valid states are not existing in our circuit even though we assume that flip flops are ideal and many ideal conditions prevail in our circuit . So in effect clock fails to capture the data due to late arrival of the data and this is known as zero clocking we call this as a case of zero clocking the circuit goes into an invalid states these are these are invalid states why because it is different from the one that we expect in a synchronous circuit. Now let us look into another scenario in in this case what we are saying is that let the delay of d1 d2 d3 c1 all of them be very very small or negligible while the delay of this element c2 is somewhat positive and call it let us call let us call it as delta. Now in this situation we can draw an equivalent circuit where d1 d2 d3 are very small or negligible and then we have a delay delta amount delta amount on the on the for the element c2. Now in this case what will happen? In this case the data that was there at the a that came here in the next clock edge will go at this point but when it reaches this point the delay from this point c2 through d2 is very small.

So a reaches very quickly to the input of f f2 d and there is a delay on the clock path and the clock will reach later on . So what will happen is that in the second clock cycle we will have the data in two different flip flops . So the same data got last by two flip flops in the same clock cycle . So this is the k so then the circuit is still on in the invalid state. So this happened for this cycle similarly it will happen for the other cycles also .

So the states will be p p a a b b c c for this case and while the valid states were what we saw in the synchronous circuit . So now the circuit has gone into the wrong state. So in effect the data gets captured by two flip flops by the same clock edge and therefore we call this scenario as we have these double clocking scenarios ok. So this scenario is known as double clocking and the circuit goes into invalid states ok.

So we need to avoid this. So we know what the tool does to ensure that double clocking and the zero clocking does not take place. So the NSTA tool avoids zero clocking by setup analysis or the late analysis. Why is it late analysis? Because it is looking into the data path and checking whether the data delay between two flip flops is not too high or not at the destination at the capture flip flop that the data does not reach too late so that the synchronicity of the data is lost . So we avoid zero clocking by setup analysis or late analysis and we avoid double clicking by hold analysis or early analysis. What it means is that in order to avoid double clocking the tool will ensure that the data does not reach the capture flip flop too early so that it gets captured in the same clock cycle .

So that is what is checked in in hold analysis or early analysis. So a synchronous circuit can now a synchronous circuit can contain many flip flops in the in the in the in the figure that we just saw there were only two flip flops by in in in in a real design there may be thousands of flip flops. Now in that case how the timing constraints are checked. So in that case what the tool does is that it considers constraints between pairs of adjacent synchronously adjacent flip flops. It considers flip flip flops pairwise .

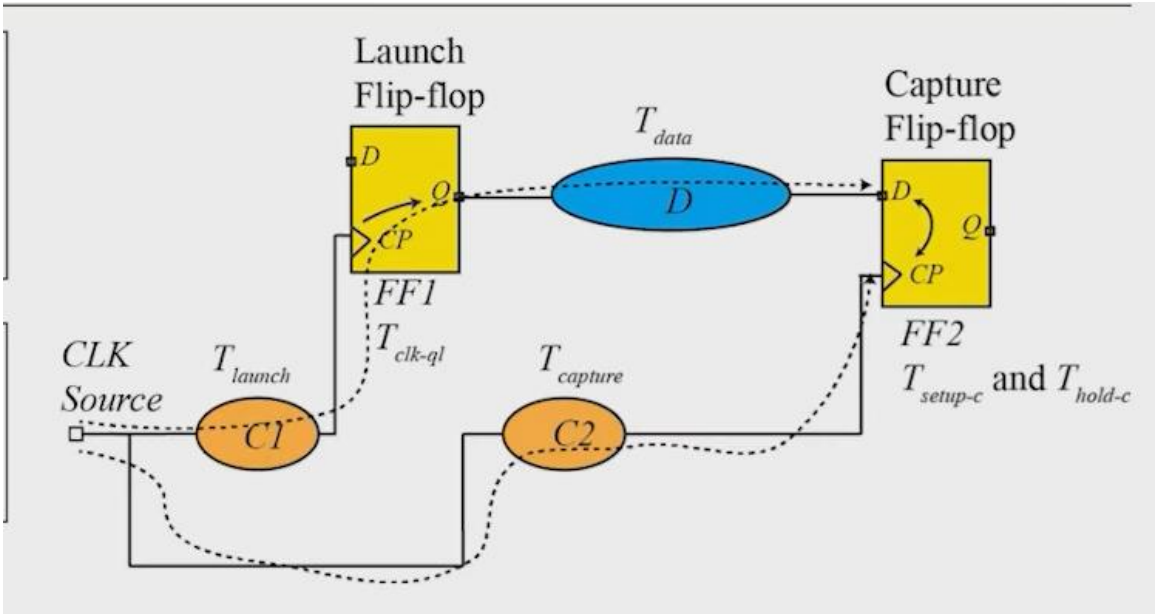
So in a synchronous circuit which contains many flip flops data can propagate sequentially through a pipeline before through a pipeline before reaching the output. So the STF tool examines each pair of launch and capture flip flops separately . So the tool will consider the path from one flip flop to the sequentially adjacent flip flop meaning from the launch flip flop to the capture flip flop. It will consider all pairs of such combinations that are possible in the circuit. So now there are various types of combinational gates that can be encountered in the path.

So we have taken a simplistic case in which we have said that only the buffers in our path are . What if there are say AND gate, NAND gate, XOR gates and other kinds of complicated gates. In those cases what the tool will do is that it will add delay of the combination of circuit elements in the path and the check for the delay requirements. For example there was a NAND gate followed by a NAND gate and so on. Then it will add together the delay of all the circuit elements which is encountered from say Q to D and based on that it will ensure that the delay between Q and D is such that the synchronicity of the data design is maintained .

So now another important thing is that we have considered the ideal flip flop . Now what if our design has a non-ideal flip flop . Typically our design will have some set up time . It will also hold time . So it will also have a clock to Q delay .

In those cases how is the analysis done? So in those cases the analysis is similar to what we have done in the previous lecture. Only thing is that the constraints become more restricted. There is the need to consider the time when the signal arrives at the deep end, slightly taking into account the margins of the set up time and hold time of the flip flops also. So it makes the verification a bit more pessimistic. Yet the same analysis can be used to ensure that the states of the synchronous circuit are valid and also the flip flops that are in our design. The constraints of setup time and hold time for those flip flops are also maintained or those are also met.

So now let us look into the set up requirement that needs to be met for a realistic circuit . So in the set up requirement check what is done by an STA tool is that it ensures that the data sent by the launch flip flop in any given clock cycle is captured reliably by the capture flip flop in the next clock cycle . And while ensuring this the tool will also check for the set up requirement of the flip flop at the capture side. So let us understand how the tool does this . Let us consider this a part of a circuit in which there is a launch flip flop is the capture flip flop and these two flip flops are synchronously adjacent meaning that there are only combinational circuit elements in between them .



So let us assume that the combinational circuit element is D and its delay is T data and there are two clock there are two clock two delay elements on the clock pass C 1 and C 2 the delay the delay of of of the element C 1 is T launch and the delay of the element C 2

is T capture ok. And then let us assume that the clock is generated at this clock source and the clock to q delay of this flip flop is T clock ql and the setup and hold time at the capture flip flop is T set up and T hold . Now understand that we are considering the setup and hold requirement at the capture flip flop not at the launch flip flop . At the launch flip flop the check is not happening the check the check of timing is happening at the capture side at the capture cap of a of the synchronous at or the for a pair of flip flop the check happens the check for setup and hold time happens at the the at the capture side of the pair of set a pair of flip flops ok. Now let us understand what are these constraints or requirements that must be met for this circuit.

$$t\ (\ arrival\ ) = T(launch) + T(clk\text{-}ql) + T(data)$$

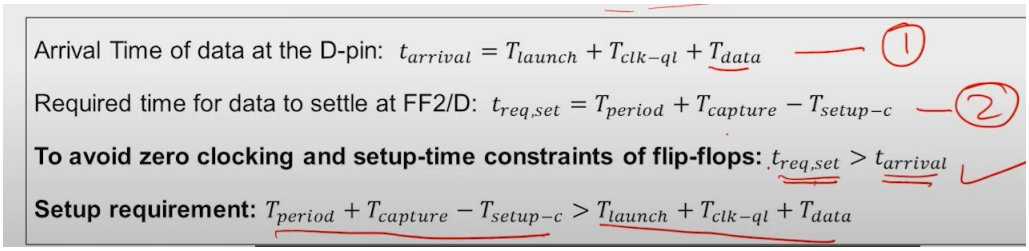
Now the arrival time now let us assume that the clock edge is generated at time t is equal to 0 at this point . Now this this clock signal will go and it will it will it will undergo a delay of T launch and it will undergo a and it reaches this clock input and a data will be captured will be whatever was at d it will be moving to q and it will start propagating in this combination and it will propagate and finally, reach the d input. Now when this data reaches the d input we can add the delay and just find out . So, the delay will be of this element and then the delay of the clock to q delay of the launch flip flop and the delay of the combinational circuit element. So the arrival time of the signal at the d pin will be T launch plus T clock to q delay this is the clock to q delay of ff1 and T data of the combinational circuit element ok.

$$t(req, set) = T(period) + T(capture) - T(setup\text{-}c)$$

And then the required time for data to settle at f f 2 d when this data should settle. So, we have seen that if there is a clock edge we are not allowed to make any change in data for setup for the time that is known as T setup for T setup duration we cannot make change in the data. So, data should arrive before this it cannot arrive later than this and therefore, the required time is T period. So, now, when will the next clock edge come? So, we want the data that was launched by this clock, this flip flop flip flop 1 it should get captured in the flip flop 2 in the next clock cycle .

Now when will the next clock edge be generated at this point it will be generated after one clock period that is what T period is . And then it will start traversing and it will undergo a delay of T capture . And then it should reach and and then it reaches at this point and data should reach T setup time before this clock edge. And therefore, the required time is T period plus T capture minus T setups , that is the time when we expect

that the data should come before this time . So, we can write this constraint as T required setup is greater than T arrival .



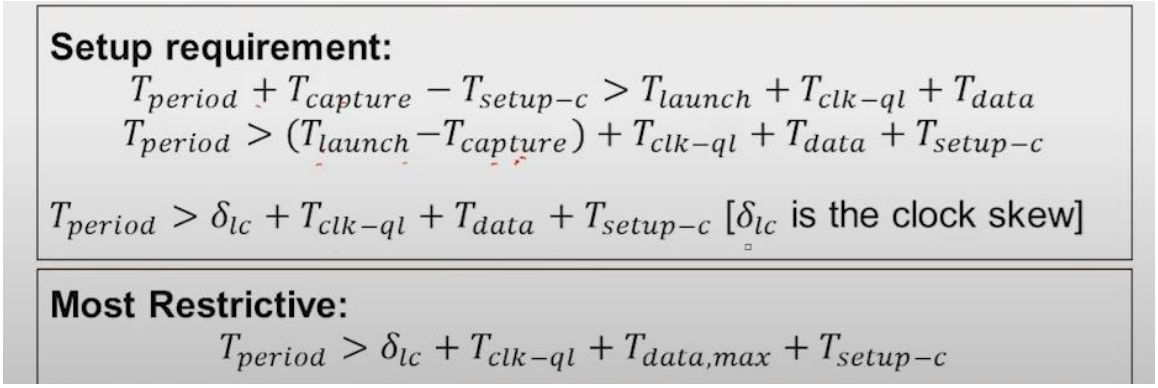
So, this is the arrival time and this is what we want. that this required time should be greater than. So, the arrival time should be less than this required time. This is what the constraint of setup time is . So, we can write this now. We can just put this expression, this equation and this equation in this equation and we get this expression . So, T period plus T capture minus T setup that is T required should be greater than T launch plus T clock to queue delay plus T data ok. Note that clock to queue delay is considered for the launch flip flop not for the capture flip flop.

Now, we can write the same equation in by moving this T capture on the other side . So, we get T launch minus T capture. This moves up to the other side and that is why it becomes minus T capture plus T clock to queue delay and the other terms remain the same . So, what we have done is that we have moved T capture on the other side and also T set up on the other side . So, this gives us the constraint on the clock period how much will be the minimum value of the clock period that we expect for which this constraint will be valid . So, now the term T launch minus T capture means what is the difference of the arrival time of the clock signal at this point and this point .

This is known as the clock skew or the clock skew meaning what is the difference in the arrival time of the signals at the clock. So, in terms of if we denote clock skew as delta L c L L means for launch and c is for capture. So, delta the T period should be greater than delta L c plus T clock to q L ah clock to q delay at the launch flip flop and T data . So, this is the T data and T plus T setup c that is the setup time of this capture flip flop . Now what the tool will do is that the tool will pick this T period from the constraint file or SDC file where we have specified the clock frequency .

And then this these terms this quantity will be decided by the tool based on the clock path the attributes of the flip flop clock to q delay of the flip launch flip flop and the setup time of the of the capture flip flop will be picked from the lab rate from the technology lab rate this information will be picked. And then T data will be computed for this for the path between q and d how those T data will be computed we will see in

the later lectures. It will just be the sum of delays encountered for each of the gates in the path . Now because of many reasons these numbers can vary . So, for example, due to process induced variations the clock to q delay and say setup time of flip flops and also the skew of the signals or skew of the clock signal can vary because of say process induced variations.



And also there can be multiple paths between q and d there can be multiple paths. Now this constraint should be valid in a circuit for the worst case scenario and therefore, what we should do is that we should consider this data to be the max because then if we take the maximum of suppose there were three paths one was having a delay of 100 other as a 200 and third one was 150. So, out of various path we should pick the path which has got the maximum delay of T 200 why because if we take the maximum then this constraint is most likely to fail this because this is coming on the less than side if we take the maximum value then it is most likely to fail and therefore, we should take the maximum of all we consider maximum of all these values. Now, setup violations can occur if the clock period is decreased .

So, if we decrease this, this constraint can be violated. So, if we decrease the clock period then we are basically increasing the clock frequency. So, there will be a limit on the clock frequency at which a given design can work beyond that it cannot work . So, if we go on decreasing the clock, increasing the clock frequency or decreasing the clock period at some point this inequality will break and will be violated. Similarly, if the delay of the capture clock path is decreased if the delay of the capture clock path is decreased or the delay of the data path is increased or the delay of the launch clock path is increased then this constraint is more likely to fail ok.

Now, let us consider a scenario in which we have an ideal flip . So, suppose there was an ideal flip flop. What will happen instead of these flip flops suppose these were ideal flip flops. Now, if the flip flops were ideal then the clock to q delay will be 0 we can remove this and setup time will be 0. So, in that case what is the constraint? So, the constraint is t period should be greater than the clock q delta l c plus t data max . So, in a

if the and what will happen if we have an ideal clocking structure meaning that if the clock reaches at this point and this point at the same time that means the clock q is 0 we say that we this is an ideal clock. If the ideal clock's clocking structure was 0 then this term will also vanish.

So, then what the final constraint remains for the ideal case t period is greater than t data max . So, now what does this constraint signify? This constraint basically signifies that there is no no 0 clocking in our circuit if that is what we discussed earlier . So, if we saw that 0 clocking can happen only when the one we saw when the the delay of one between the two flip flops were more than one clock cycle, but this constraint says that the if the delay is less than one clock cycle for the ideal situation that is the flip flops are ideal clocking structure is ideal then there will be no 0 clocking So, this constraint basically avoids 0 clocking in our circuit for the ideal case. Now let us similarly compute the hold requirement . So, the hold check is from one active edge of the clock in the launch flip flop to the same clock edge at the capture flip flop .

So, in what we check in the hold hold requirement we check that the data that is launched by a flip flop does not get captured by the capture flip flop in the same clock cycle . So, it is the same clock cycle, not the next clock cycle . So, it also ensures that the hold requirement of the flip flop is also met . So, with the circuit the only thing is that instead of the setup time at the capture flip flop we now consider the hold time. Now what will be the arrival time of the signal at this point it will be the same.

So, if a clock edge is generated at this point at time 0 then it will reach here after t launch at this point and then it will reach here after t clock to q delay at the of launch flip flop and then from this point to this point it will take t data or amount of delay . This is the arrival time at this point. Now what will be the required time? Now data arrives at f f 2 d after the required time and that required time is t capture plus t holds. So, let us understand what this required time signifies . Now there was a clock edge that was generated at time 0 at this clock source.

Now that same clock's edge will reach this point after a delay of t capture to capture . So, this is and for the hold requirement we are trying to check that data is not being captured by the same clock edge. Therefore, we are not taking a clock period in this case . So, the clock period is immaterial or not and not need not be required to be considered in this case .

Now how does this term come? That second term t holds c . Now we have seen that if there is a flip flop these are forbidden windows. So, after the clock edge there is a time to hold for which the data should not change . This is to hold or rather to hold . So, this is

the duration for which the time data cannot change at the Q pin and that at the D pin or at the D pin we are not allowed to change. So, this is on after the clock edge the data should remain steady for the time to hold.

Now the we want that the required time is required time is such that the data will data should not come before this and also within this window if it should not come . If it comes within this window then we will get a hold violation and data should come any time after this window and that is why we need to add this to hold c in this . So, to avoid double clocking and hold time constraints of flip flop we need to ensure that the arrival time is greater than t required meaning that the arrival the data can come anything any time after this and therefore, t arrival is greater than t required hold . So, that is what we have computed .

Now using these two equations we can expand this inequality. So, this equation and this equation we just put it and we get this hold requirement for these two pairs of flip flops. Now we similar to what we had done for setup analysis we can take this t capture on the on the on the LHS and we get this delta s that is t launch minus t capture is defined as a clocks q and we denote it as delta s and t clock to q delay and t data that should the the sum of this to be greater than the t hold or the hold time of the capture flip flop. Note that we are considering the hold time of the capture flip flop not of launch flip flop. Now what will be the most restrictive requirement for this. So, we know this inequality is more likely to fail if this number is very very small .

So, as we go on decreasing this as we go on decreasing it this inequality is likely to fail and therefore, t data should be taken as minimum . So, if there are multiple paths between this we should consider the path which has got the minimum delay for most restrictive or pessimistic analysis . Similarly, if there are PVT variations if the clock q is changing or clock to q delay is changing or hold time is changing. So, for these two thing quantities that are clocks q and the and the and the clock to q delay we should take the minimum value and for the hold time we should take the maximum value because then this constraint is most likely to fail . Now hold violation can occur if delay a delay of the data path is decreased if we decrease this it is more likely to fail ok or delay of the launch clock path is decreased if we decrease this or delay of capture path is increased if we increase this then that then the then the this inequality is likely to break down.

Now what will happen for an ideal flip flop let us consider an ideal flip flop what will be the constraint for that . So, for the ideal flip flop what we can say is that for an ideal flip flop we can say this clock to q delay is 0 and hold time is 0 . So, what constraint do we get? We get delta L c plus t data min should be greater than 0 this will be the constraint. Now when is this constraint so when is this constraint likely to fail to to to get violated

typically t data min cannot be negative . But delta L c that is the q can be negative because this is the difference of these two quantities t capture and t launch and t capture and if t capture is more the data is likely to be if the if the the clock signal encounters more delay on the capture path then on the launch path then this quantity can become negative .

And that is what we saw when we were looking at the double clocking scenario. We made a situation in which the clock q was negative and as a result of this this constraint was violated and we assume that this is negligible and that is why it was violated in our case . So, in effect if we assume an ideal flip flop what the whole requirement is checking it is checking that the design there is no double double clocking in our design . And other important thing that can be inferred from this equation is that there is a the the the the whole requirement strongly depends on the strongly depends on the the clocks q and if in during logic synthesis we have assumed say clocks q to be 0 typically we assume it an ideal structure then clocks q is 0 in that case this t data the the the whole time constraint just translates to t data min is greater than 0 . It is typically more likely to be to this followed or no or this constraint is likely to be met most of the time because the delay will be there will be some delay on the data path. Thus the corner cases will be the situation in which we have a kind of shift register in which we have a flip flop and then another flip flop and there is no logic between them. A shift register kind of scenario if there is in our design then the whole violation is more likely to happen in those kinds of situations.

Another thing that we should infer is that the the during the logic synthesis step it is typically not that much important to look into the to the the whole check why because the whole check depends a lot on the on the on the on the on the clocks q and that picture we get only when we have done clock tree synthesis or we are into the physical design. Therefore, fixing of whole time violations during the logic synthesis does not makes much sense because most of these whole violations which we see in logic synthesis will typically be will be automatically fixed because of say wire delay in on the data path and new whole violations can appear after clock tree synthesis because of the clocks q. And therefore, fixing whole violations typically we postpone up till up till the time we have gone into the physical design we have made the actual clocking structure and then after that we will have a better picture of whole violations and then we fix the relevant whole violations in our design. So, if you want to look into this concepts in more detail you can refer to this book. Now to summarize what we have done in this lecture is that we have looked into that why why a STA is needed in a design, how does STA ensures that synchronous behavior is maintained in a design by analyzing the cases for zero clocking and double clocking and throwing violations whenever it finds that zero there will there can be there can be a situation of zero clocking or double clocking in our design.

And also we looked into how the setup constraint or requirements and whole requirements for a circuit is derived and how those requirements ensure that the given circuit will meet the setup and hold time of each flip flop in our circuit and we also be capable of operating at the specified clock frequency. So, in the next lecture we will be looking into how these constraints or the timing requirements of setup time set and and hold time and early and late analysis how these requirements are modeled by the tool internally and then evaluated and checked. So, we will be looking into the mechanism of STA in the next lecture. Thank you very much.