

VLSI Design Flow: RTL to GDS

Dr. Sneh Saurabh

**Department of Electronics and Communication Engineering
IIT-Delhi**

Lecture 51 Clock Tree Synthesis (CTS)

Hello everybody, welcome to the course VLSI design flow. In this lecture, we will be discussing clock tree synthesis. In the earlier lectures, we had seen that the physical design is broken into multiple tasks which are shown in this figure. And in earlier lectures, we had looked into chip planning and placement. So in this lecture, we will be discussing the next step in the physical design that is clock tree synthesis. So specifically in this lecture, we will be first looking at some terminologies related to clock tree synthesis.

Then we will be looking at various types of clock distribution networks and then we will be looking at clock network architectures and a concept of useful Skews. So before going into physical design, we had done logic synthesis. And in logic synthesis, we assume that the clock signals are ideal. Now what do we mean by ideal clock signals? So by an ideal clock signal, we mean that the clock signal has the same waveform at all points in the circuit.

By waveform, we mean the value of the voltage of the clock signal at various instantaneous times . So ideally, we expect that if there is a clock source and it is generating a clock signal, the exact clock signal is available at all the sinks or all the clock pins of the flip flops act instantaneously. And at all the points of the circuit, the clock signal is behaving in the same manner. So in particular, we expect that there is no clock skew. And what do we mean by clock skew? The clock skew is the difference in the arrival time of the clock signal at the flip flop.

So if there is a clock source, then this is a clock source and there are multiple clock pins in our design. We expect that at all the clock pins in the design, the clock signal is being received simultaneously at the same time. So it means that there is no skew or difference in the arrival time, arrival time meaning that how much time it takes for the clock signal to start from the source and reach the clock pin, that is the arrival time. So the arrival

time, we expect that the arrival time at all the flip flops of our design is the same. Now this is the ideal behavior and if we go into physical design, the ideal behavior can never be ensured.

Why can it never be ensured? Because we are making the layout and in the layout, the clock signal will be generated from one particular point, maybe say from a clock generator and from there it will be fed to all the flip flops in our design. Now as the clock signal goes from one point to the other or it propagates through the network of clock, network of various components in the clock network, the clock signal will encounter delays and these delays can be different for different clock pins. So for some clock pins, the clock signal can reach very quickly, for others it may take more time because the distance between the clock source and that pin is more and so on. And therefore, in physical design we cannot ensure we can we can we can have non-idealities because of the delays in the clock path. Now what does the clock tree synthesis try to do? So clock tree synthesis tries to make or implement a clock distribution structure such that it mimics an ideal clock or if the layout of the clock components are made in the clock network such that the behavior of the clock signal is similar to or to the ideal clock.

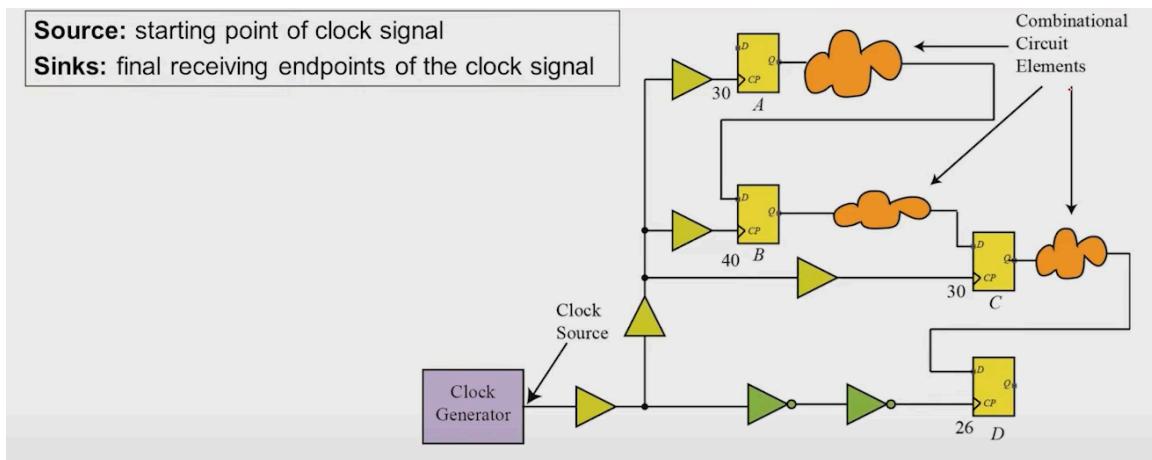
Meaning that the ideal clock is the one which we want right, that is what what what what is targeted. But due to no delay effects in the clock network that can never be ensured that that clock network can never be implemented such that the clock is ideal meaning that there is no skew. But nevertheless what clock distribution, clock tree synthesis tries to ensure is that the clock skew is minimized right. So it tries to build a clocking structure such that the ideal clock network is mimicked right. It can never achieve the full ideality, but it tries to achieve a 0 skew or or non or similar to an ideal clock right.

So it minimizes the clock skew and it inserts clock buffers inverters and performs routing of the clock distribution network. Now the clock the the clock tree synthesis also tries to reduce power dissipation in the clocking structure. And why does it do it? Because the clock network consumes a significant portion of the active power it can be as high as 25 percent to 70 percent right. And therefore, the CTS or clock tree synthesis tries to reduce the power dissipation in the clocking structures. So, the clock signal is a very active signal meaning that it transitions every a in one clock cycle it transitions one rising and one edge one falling edge it will have right.

So, the clock signal transitions very rapidly and therefore, dynamic power dissipation in the clock network is very high and therefore, CTS tries to reduce the dynamic power dissipation in the clock network. And then it also tries to fix timing violations and signal

integrity issues because the clock signal is a signal which is rising very rapidly right. Ideally its rise transition should be as abrupt as possible right. Now because the rise time is very small or the signal is the clock signal is transitioning very quickly it can impact neighboring lines right and therefore, the signal integrity issues related to clock signals are also important in CTS. Now let us look at a few terminologies related to CTS.

So, what is a source? So the source for a clock is the starting point of the clock signal and what are sinks? Sinks are the final receiving end points of the clock signal. For example, if we look into this ah structure so here there are these flip flops right and these are being clocked by this clock generator right. So, the clock generator is basically generating the clock signal and therefore, this pin of the clock generator can be considered as the clock source right. Typically the information of the clock source is obtained through the SDC file. So, remember that when we discuss constraints we discuss how sources of the clock sources are defined in the SDC file using create clock and create generated clock constructs.



So, those information are used by the tools to infer what are the clock sources in our design and these are flip flops and the pins, the clock pins of these flip flops are the sinks of the clock network. Now in the clock network there is a tree kind of structure. If we look into the clock generator right the clock generator is the root of the tree and from there we have various branches emanating and finally, the clock pins are the leaves of the tree right. So, we can think of the clock source as the root. From there the branches emanate right and then finally, the endpoints are the clock sinks right. And in the branches that we have we will have many buffers and inverters. So, these are buffers, there could be inverters and so on right which are basically used in the clock network to deliver the clock sink right.

Now let us understand a very basic question: why do we have so many buffers and inverters in the clock network? What is the role of these buffers and inverters? So, to understand it let us take an example where say there is an inverter and this inverter is

driving a very long wire right and at the end there is another inverter right. Now this wire is long now why we are taking a long wire because in the clock distribution network from the source to the sink you have a long path right the the flip flops can be on the layout at any point right and therefore, there will be long paths from the clock source to the sink right. Now let us assume that the length of this wire is say $2L$ and let us assume that the resistance per unit length is r and capacitance per unit length is a small c . So, what will be the resistance and the capacitance of this wire? The resistance R can be given as $R = r \cdot 2L$.

$$R = 2rL$$

$$C = 2cL$$

$$T = RC = 4rcL^2$$

So, we can write $2rL$ right, that is the resistance and what is the capacitance again $2cL$ is the capacitance per unit length and $2L$ is the total length of this wire right. Now as a first approximation what will be the delay of this interconnect we can think of this delay as a first approximation as a product of R and c right and therefore, if we take the product of these two terms it will be $4rcL^2$ right. Now let us assume that we have divided this wire into two parts: take L driven by one side and L on the other side. So, suppose we have the same structure and in this we insert a buffer at the length of L and then another on the other side we have left with L . So, we have just transformed this circuit into another circuit in which the length is so we have extra buffer right.

Now let the resistance per unit length and capacitance per unit length be the same right. Now in this case what will be the delay that we expect the delay that we expect in this case will be $\tau = rcL^2$ for the first section plus the delay of this buffer right. Let us call it B plus the delay of the second section that is rcL^2 . Now if we assume that the buffer delay is very small compared to the wire delay we can neglect this term right. And the total delay in this second circuit will be approximately 2 times rcL^2 .

$$T = rcL^2 + rcL^2 = 2rcL^2$$

Now if we compare with the original circuit it has reduced by half right. So, if there is a long wire and if we insert a buffer in the middle then the interconnect total interconnect delay is expected to reduce right. Given that the delay of the buffer itself is small, right. So, that is the first reason why we want to insert buffers in the clock network because there are long wires and to reduce the delay we can add buffers to add difference or we divide the long nets long wires into smaller wires and have buffers in between right. The second reason is that if there is a long wire and the signal starts an ideal clock signal

which is rising very quickly, the right transition time is very small . when it reaches the end point the signal quality will be very bad meaning that it will be rising very slowly right.

Because of the high resistance and capacitors offered by the interconnect right. So, the slew in this case may be say 2 picoseconds here the slew will increase to say 100 picoseconds right and this is very large right. So, we want the clock signal to be very very sharp. Now if we introduce a buffer in the middle as in this second circuit what we are doing is that in a sense these buffers will have VDD also and ground line also. So, this buffer is basically pulling up the signal to the to make it more sharper and as a result the quality of the signal that finally reaches the end point this buffer the final buffer here that improves.

So, if the signal was very sharp here here we expect that it will be the signal quality will be much better because of the driving effect of this buffer right and that is why we have in the in the clock tree we we have lots of buffers and inverters to restore the quality of the clock signal and also reduce the delay of the clock signal in propagating through the clock network right. Now we have what is insertion delay in a clock clock tree. It is the time taken by the clock signal to propagate through a clock tree and reach the clock signal right. Because a clock sinks right, how much time it takes for a signal to go from the clock source to the clock sink is known as insertion delay. Typically we call this as latency during synthesis, but when we go to clock tree synthesis we get actual value and ah and there we call it as insertion delay. Latency is a kind of an estimate of the insertion delay.

Now what is the clock skew between two sinks? Clock skew is the difference in the arrival time of the clock signal between a pair of sinks S1 and S2 right. If there are two sinks S1 and S2 we ah if the difference in the arrival time of the clock signal is known as clock skew right. So, mathematically $\Delta S_1 S_2$ that is the the skew between two sinks S1 and S2 is equal to $t_{S_1} - t_{S_2}$. t_{S_1} is the arrival time of the clock signal at S1 and t_{S_2} is the arrival time of the clock signal at S2. So, if we look into this clock tree or that we discussed in the last slide, if they say that the clock signal was starting at a 0 point and the delay of insertion delay was a 30 picosecond then the arrival time at a will be 30 picoseconds right.

Clock Skew between two sinks: difference in the arrival time of clock signal between a pair of sinks S_1 and S_2 , i.e.,

$$\delta_{S_1 S_2} = t_{S_1} - t_{S_2}$$

Similarly, 40 is the arrival time at b c, 30 is the arrival time at c and 26 is the arrival time at d. So, assume that this is in the circuit. Then in this case what is delta a b that is the skew between the sink a this clock pin and sink b this clock pin this will be equal to 30. Let us assume that the time unit is picosecond. So, that delta a b will be 30 minus 40 is equal to minus 10 picoseconds right.

$$\begin{aligned}\delta_{AB} &= 30 - 40 = -10 \text{ ps} \\ \delta_{BA} &= 40 - 30 = 10 \text{ ps}\end{aligned}$$

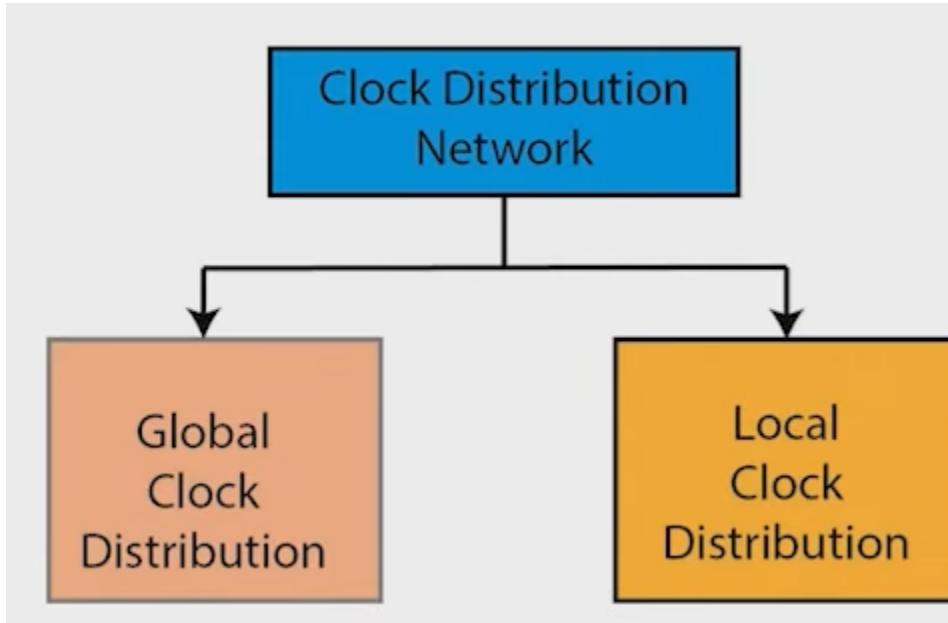
Now, if we take delta b a then it will be 40 minus 30 is equal to 10 picoseconds right. So, the sign of the skew is also to be noted. Now, what is the global clock skew? So, global clock skew is the maximum value of the clock skew between

any pair of sinks. Now, if there are say 4 clock sinks in this case then ah what will be the maximum ah clock skew right that will be that is defined as the global clock skew right. So, if we take the maximum value of the clock delta s1 s2 where s1 and s2 are any sinks then that we get the global clock skew.

$$\Delta_{global} = \underline{\text{Max}} |\delta_{S_1 S_2}| \text{ where } S_1 \text{ and } S_2 \text{ are sinks}$$

So, in this case we see that this one is having the if we take the difference between b right and d right then we will get the maximum right. So, because d is having the minimum arrival time and b is having the maximum arrival time. So, in this case delta global is equal to 40 minus 26 is equal to 14 picoseconds right. So, these are some of the definitions ah you should ah you should be aware of and ah typically the tool tries to ah CTS tool tries to reduce the clock skews between ah between flip flops right. And sometimes for easy evaluation the clock treat synthesis tool can also target improving the global clock skew rather than looking at individual clock skew between 2 6.

Now, let us look into how clock distribution networks are created. So, we create a clock distribution network in 2 steps or in 2 hierarchies right. So, at the top we have what is known as the global clock distribution network and using this global clock distribution network we drive the local clock distribution network. So, what it means is that suppose we have a chip right, we have a chip and we have a clock source. So, from this clock source we will first make a global clock distribution network right which will basically distribute the clock signal throughout our chip right.



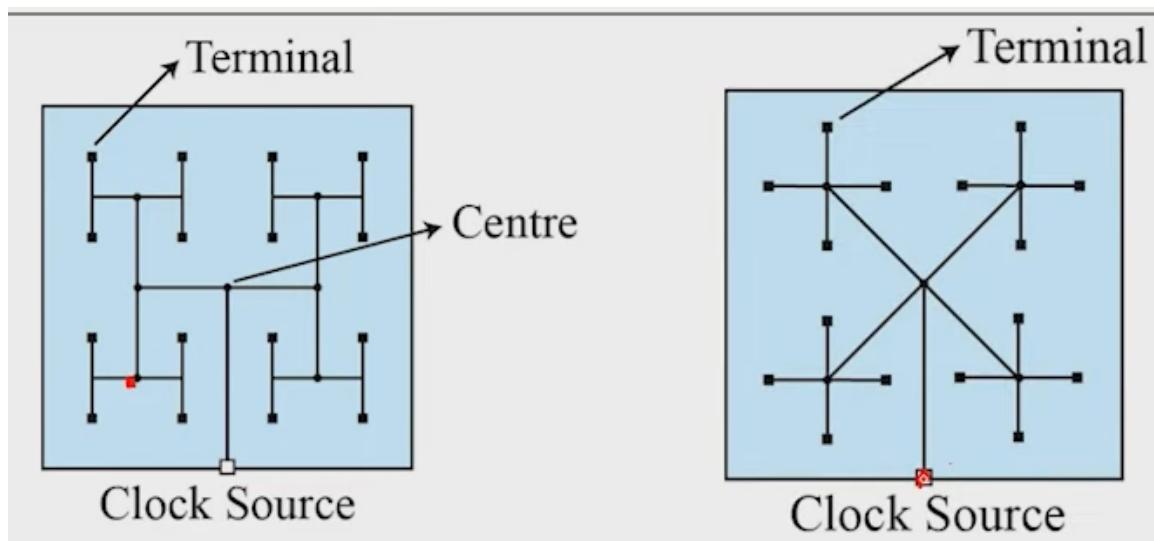
And wherever this clock global clock distribution network ends right these are the end points of the global clock distribution network. I am not showing the buffers and inverters in this global clock distribution network just for easy illustration right. So, from the point where the global clock distribution network ends there the local clock distribution network will start and this global local clock distribution network will basically supply clock signals to the flip flops right. So, the responsibility of the global clock distribution network is to distribute or propagate the or propagate clock throughout the chip right. Now, from the end points of the global clock distribution network the local clock distribution network starts and those local clock distribution networks deliver the clock signal to individual flip flops ok.

So, let us see the differences between these 2 networks. So, the global clock distribution network distributes the clock to various parts of the chip and over a larger area while the local clock distribution network distributes the clock to smaller parts of the circuits. And the global clock distribution network will have larger buffers and it will be numerous right. Why because the clock global clock distribution network needs to propagate clock

signal to over a longer length and therefore, will need more buffers and larger buffers right. And the global clock distribution network will consume larger power because of larger buffers and also the larger capacitances associated with a global ah global clock distribution network.

And another point to note is that the top level clock distribution network or the global clock distribution network needs to be planned and a lot of manual intervention is made in the designing of the global clock distribution network. While the automated clock distribution network sorry while the local clock distribution network those are created automatically by the CTS right. So, now let us look into how the global clock distribution network is created. So, typically we create a global clock distribution network in a symmetric manner or in a symmetric tree form is one of the ways in which we can implement a global ah clock distribution network. So, in symmetric tree architecture what is done is that the clock is first routed to a central point in the chip right from from the clock ah source the clock is first routed to the center point in the chip and from center point central point another symmetric architecture fogs out.

So, ah there can be many types for example, H-tree type or X-tree type of symmetric tree net tree architecture is very popular. So, let us understand what H-tree or X-tree architecture is. So, this is an example or this is a figure of H tree architecture. So, this is the clock source from here the clock signal goes and first routed to the center of the central point of the central point in the chip right and from the central point we have another symmetric structure that is H right and then wherever the end is there of H we have another symmetric structure and so on right. Similarly, we can have an X type of architecture right.

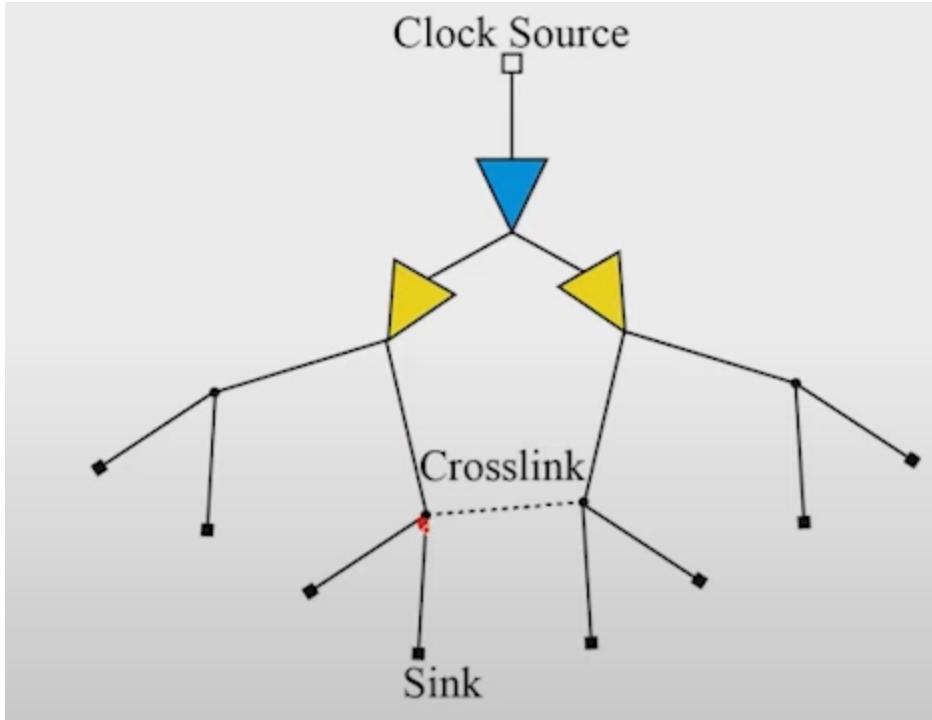


So, what we were from the clock source first we routed to the center point of the chip right then from the center point we have a symmetric X architecture right and where this

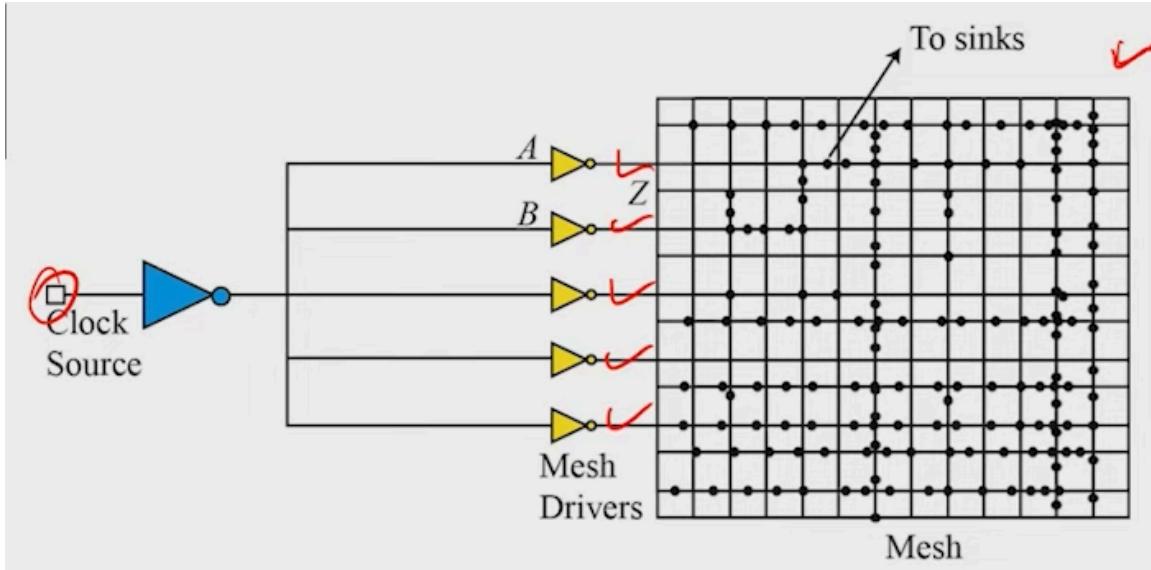
X ends there we have another symmetric architecture right and so on. So, if we look into this this architecture then we these both architecture we can see that from the clock source to any of these endpoints these all are endpoints right these endpoints all have the same wire length right same length right and therefore, we expect that the clock signal will encounter exactly the same delay in all the paths right because the end points all the endpoints have the same distance from the clock source right. So, ideally if each path is balanced skew will be 0 right ideally that will be the case we will of course, we will add buffers and inverters in this in this network to enhance the drive strength and restore the quality of clock signal and if we add matched buffers and inverters then ideally the the clock skew will be 0 at each of the end points in our clock distribution network right. However, due to PVT variations skew will still be observed right for example, when we fabricated there can be small differences in say the thickness of the wire right because of unavoidable process induced variations right. So, even though we have designed the length to be exactly same the capacitances can vary for different sections of the interconnects and even if we if we have matched in buffers some buffer can be can have more delays some can have less delay and so on maybe because of the variation in the gate oxide thickness and so on.

And as a result what can happen is that the delay at different endpoints may be slightly varying right at for example, here we have say a delay of say 10 picosecond and at this point we have say 98 picosecond and so on right. So, just for illustrative purposes do not take it at an exact number right. So, this is so due to PVT variation the skew will still be observed in this kind of architecture right. Now, how to reduce even that skew right even if that is observed in symmetric architecture. So one of the solutions which is widely used is to use non-tree architecture.

So, even though we call it clock tree synthesis, many times the clock distribution network is not a tree, it is a different architecture. So, instead of a tree purely tree architecture we can have a non tree architecture which can reduce the PVT variations. Now what can be this non-tree architecture? A simplest case can be that we have cross links between nodes. For example, between these two nodes right if we ignore this cross link then the distance from this clock source to the end points all are the same and therefore, ideally the skew should be 0. But because of PVT variations there can still be skew, but if we introduce cross links as this as it is shown between two two points then the delay at this point and say delay at this point will be correlated. Why because of the cross links they are sharing some paths right.



And therefore, that if the path between the two end points is being shared right then the delay will be correlated and that is what will be observed in these two cases and as a result of that the PVT impact due to PVT variations is expected to be lower in this architecture compared to a purely tree architecture. Now we can take this concept of cross link further up. What we can do is that instead of trees with cross links we can make a mesh architecture which is where there are cross links everywhere right. So, what we do is that we create 2D structures with redundant wires and devices right. So, we first create a mesh architecture over the entire chip area right over the entire chip area. We create a mesh architecture something like this and drive this mesh architecture from the clock source at multiple points. So, we have shown only 1, 2, 3, 4, 5 points where the driving clock source can be driven from this side and so on all right.



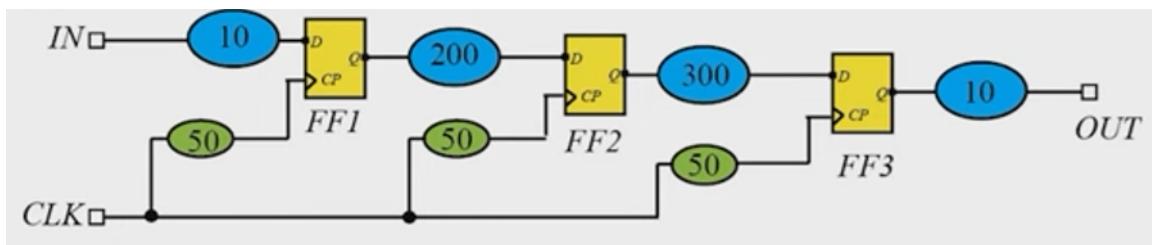
So, we can drive it from multiple points right and if we look into any one point at any one point in this grid there will be multiple paths from the clock source to this point through the mesh right. Similarly, if we take any other point for example, this one there will be many paths of that will be shared between this point A and this point B right from the clock source to points A and B there will be many paths which will be shared and therefore, their delay will get correlated and therefore, the impact of process induced variation is expected to come down. So, the mesh architecture ensures more paths between mesh drivers and clock sinks and therefore, a very small skew is observed and decreased impact of process induced variation is also observed in this kind of architecture. And it is also robust, robust in the sense that if it is a mesh architecture then even if one link gets broken the clock can reach through another path right and therefore, this architecture is more robust than a tree kind of architecture. But what are disadvantages of mesh architecture? So, the first disadvantage is that it will have more capacitance because of more wires being used in the network right and it will lead to increased power consumption because of increased capacitance.

Remember that power is dynamic power is proportional to capacitance being driven and therefore, it will consume more power. Another reason for more power being dissipated in a mesh architecture is because of short circuit power dissipation. Now, in this case we can think that this driver A and driver B right these two are driving the same points in the sense that output of two inverters are being shorted right. Now, if these inputs have some slight skew in input because of say process induced variation in these path right and because of a small skew what can happen is that the one input is 1 and the other input can be 0 right. And therefore, for a short duration there can be a short circuit path between the power and the ground through the nMOS and pMOS internal to these inverters and therefore, short circuit power dissipation can be higher in mesh architecture.

So, we should be careful about the short circuit power dissipation also in the mesh architecture of the clock distribution network. Now, let us look into a concept which is known as a useful skew. Till now what we discussed is that the clock tree synthesis tries to minimize the skew right ideally it should be 0 right. But in some situations useful skews can actually help us improve the system performance. So, introducing well controlled skews can improve the system performance and when that happens then those skews are known as useful skewed right.

Now, when can clock skews be useful? It can be useful if there is a significant difference in the slack on the two sides of the flip flops. So, we will see an example then it will become more clear. So, if there is on the two sides of the flip flop there is a significant difference in slack then what can be done is that we weave the clock signal at the two different endpoints or different start and launch and capture points. And as a result what happens is that we will be able to allocate excess margin on one side of the flip flop to the more criticals right. How can it be done? We will see with an example then it will become more clear because it helps if we do some clock skewing then what we are basically doing is that we are allowing more time for data to propagate on the critical path by delaying by changing the clocks skew right.

Now, how can it be done? Let us see with the help of an example. Suppose this is a portion of circuit that is the which or the portion of circuit that is critical which is shown here right.



Assume that our flip flops are ideal because assume that the setup time of these flip flops are 0 and clock to q0 clock to q delay is also 0 right. Let us assume that and let us assume that this input delay is very small at this point and therefore, this path is not critical similarly the output delay is small and therefore, this path is also not critical right. So, the critical portion of our design is the central path right.

Now, if we look into this fitflop ff2 it has got two paths right. First path is this one in which ff2 is basically the capture flip flop and the second path is basically this one in which ff2 is the launch flip flop right. Now, in this circuit what we have done is that the clock skew has been made 0 the ideal condition is that the delay for all the clock prints is

50 time units. Let us assume that the time units given here are in picoseconds right.

So, we have 50 picoseconds delay on all the clocks to power right. So, therefore, since the delay is the same everywhere where clocks skew is basically 0 right. Now, let us consider or evaluate that at what maximum clock frequency this circuit can operate right. So, there are two paths. First let us consider this path in which ff2 is working as the capture flip flop right. So, in this case if ff2 is the capture flip flop the arrival time will be 50 plus 200 right that is the arrival time and what will be the required time the clock period we are considering setup analysis right or.

$$50 + 200 < T_{period} + 50$$

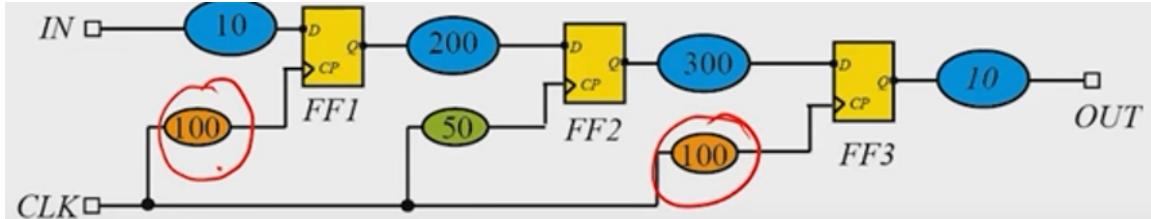
So, setup constraints. So, t period and plus 50 is the delay on the capture clock path right. Now, we have assumed the clock to q delay of flip flop is 0 and also the setup time of the flip flop ff2 is 0 for the sake of easy analysis right. So, this is the constraint for the first part right. So, we can simplify it. We can just cancel this out and we can say that t period should be greater than 200 picoseconds. Now, let us look into the other path in which ff2 is working as the launch flip flop and ff3 is working as the capture flip flop.

$$50 + 300 < T_{period} + 50$$

So, what is the arrival time 50 plus 300 right and required time similarly is t period plus 50 50 is the clock is the delay of this path right. So, we can write if we can cancel out this 50 50 we get t period should be greater than 300 right. So, we have two constraints: one constraint is coming for this path and the other is coming for this path right. Now, one constraint is saying that t period should be greater than 200 the other is saying t period should be greater than 300. So, the most stricter should be considered right if we follow the stricter then the lenient that is greater than 2 t period should be greater than 200 will automatically be fulfilled right.

So, the maximum so the minimum value of t period can be 300 it should be at least more than 300 right. So, if that is the case then we can compute the maximum clock frequency as 1 divided by 300 picoseconds that turns out to be 3.33 gigahertz right. Now, this is the case when the skew was 0 the ideal case.

$$F_{max} = 1/300ps = 3.33 \text{ GHz}$$



Now, let us introduce some useful skew right. So, what we did is that in this circuit we introduced instead of taking a 50 picosecond as the delay on the clock path we have increased it to 100 picosecond and on the other side also on this path we also increased from 50 to 100 right. Now, let us compute the timing requirements first for FF 2 acting as capture and then FF 3 acting as the capture right. So, when FF 2 is acting as the capture the launch path is this right. So, that the arrival time is 100 plus 200 and the t period it should be less than t period plus this is the delay on the on the capture side capture clock path right that is what 50 is here right. So, we can simplify it and write say 50 goes on the other side it becomes 250 right.

$$100 + 200 < T_{\text{period}} + 50, \text{ and } 50 + 300 < T_{\text{period}} + 100$$

So, 250 so t period should be greater than 250 that is the constraint for FF2 working as the capture flip flop. Now, what will be the case for FF3 acting as the capture right we can easily evaluate again. So, the arrival time will be 50 plus 300 right and the required time will be t period plus 100, 100 is the delay on the capture clock path right. So, from this what we get we can take 100 to the other side and you can easily simplify we can write t period is greater than 250 right. Now, in this case both these constraints are showing that t period should be greater than 250.

So, what should be the maximum minimum value of t period it can be minimum is 250 or and from that we can compute the maximum clock frequency that this circuit can operate right. So, we can get f max is equal to 1 divided by 250 picosecond is equal to 4 gigahertz right. So, we have enhanced the maximum operable frequency of our circuit by introducing skews in our clock distribution network right. So, what basically we have done is that by introducing extra delay on the path which was critical earlier this path was critical right by introducing extra extra delay on this clock path we are allowing the clock signal to propagate for a longer duration on the critical path. And as such we are getting some surplus slack which was observed on this side we are utilizing it on the other side of the flip clock right.

$$F_{\max} = 1 / 250\text{ps} = 4\text{GHz}$$

Now, a couple of things are important to note here. The first thing is that we have

increased the delay of say this ah this ah clock delay element from 50 to 100. What if we increase it further if we increase it further what will happen is that this ff2 will become more critical right f f earlier the critical path was related to ff3 right the path ending at ff3 or the when ff3 was acting as the critical ff3 was acting as the as the capture flip flop right. If we make this delay element larger than 100 then the ff2 path ending on ff2 will become critical right. And therefore, there is some optimum value of the clock of the delay in the clock network that will give us the maximum operable clock frequency right. So, we need to so the tool which is trying to introduce a useful skew needs to come up with an optimum value right. It is not that we can just add a delay element and it will lead to improvement right.

Another important thing is that when we introduce delay on the clock path right for example, this clock path we are improving the set of requirements or we are improving the ah or relaxing the setup constraints or setup requirement, but we are making the hold requirement more restricted right. For example, if we add from 50 to 100 if we are increasing the delay of this circuit element right then the clock is reaching the capture flip flop later right and therefore, ff3 is more likely to have hold violation than earlier right. So, while reducing the or improving the setup requirement of our circuit or improving the maximum operable frequency for circuit we should be careful that the hold violation does not get introduced because of clock skew. So, the CTS tools calculate useful skew targets for all critical registers and then it increases the arrival time of the clock for target registers by adding delay buffers or extra wires. And another side effect of introducing skew in the ah in the clock distribution network is that it helps us decrease the problem of voltage drop right.

So, in the earlier lecture when we discussed ah chip planning we discussed that there can be a problem in power ah power VDD voltage dropping when it reaches the standard sense right. Now, if we skew the clock at the clock signal what happens is that the flip flops get triggered at different time intervals and all of them do not get triggered at the same time. And therefore, the activity of the clock of the signal that was there in the circuit earlier which was unbreak concurrently gets skewed over the time and therefore, the the ah the simultaneous current drawn from the power supply that can be reduced. And therefore, the voltage drop ah drop problem can be reduced by introducing the useful skew right. Now, after we have done the clock tree synthesis then what happens is that the clock network has been built and now we have got a realistic clock network.

Before clock tree synthesis our clocks were ideal. That was the assumption we made in synthesis. We added some constraints like latency uncertainty to model as some effect of realistic clock right. By using some estimates right though, those were very crude measures. But once we have built the CTS then we can actually compute the delay of the

ah delay of the circuit elements in the clock network and based on that we can do the timing analysis right. So, now, the timing analysis that will be done will be more realistic because the clock network has already been built and actual delays on the clock path can be accounted for and therefore, what can happen is that once we do CTS new violations can come up right. And to do the timing analysis we also need to instruct the static timing analysis tool that now the clock tree synthesis has been done using the propagated delays of the clock signal rather than the set set clock latency and the and and network latency and source latency that were used earlier right.

So, source latency can still be there in our SDC file, but the network latency that was earlier introduced before CTS and during synthesis that needs to be removed and use instead of the network latency we should use the actual value of the delays of the clock ah of the circuit elements which are there in the clock network and to ah to instruct the STA tool that use the the realistic the realistic delay we use a command set underscore propagated clock in the SDC file and we can give all clocks if all the clocks have been built right. Now, new timing violations may show up after CTS then we need to fix it due to and why this new timing violations can come because of the additional skew that we will see now because of the realistic clock network and also because some logic component components on the data path they might be displaced right to make some space for the clock buffers and the structures needed for the clock ah clock tree. So, some circuit elements might be moved from the from the circuit elements which were on the data paths those might have been moved away from the from their respective position and therefore, the the the delay of the data path might have increased and therefore, we the the new timing violations can come up and therefore, and the CTS tool needs to fix those timing violations. Now, once those timing violations are fixed may be using some optimization techniques like resizing or making incremental changes in the placement and so on then the the clock network is frozen meaning that we have done the inserted all the circuit elements on the clock path we have done done the routing also of the clock signal on the clock path and therefore, we say that we we our our clock network is fully built and then we do not allow any change to happen on the clock network unless and until it is a very important change without changing that we cannot move ahead and those changes are very well controlled and are used you and and are introduced in our design using ecoo or engineering change order otherwise after CTS is done our clock network is completely built and it is frozen right. So, if you want to know more about the topics we discussed in this lecture we can look into this book.

Now, to summarize in this lecture what we have done is that we have looked into clock tree synthesis right, we looked into various types of clock distribution network and also clock clock network arty architecture and we also looked into the concept of useful

skills. In the next lecture we will be looking into the routing of a design. Thank you very much.