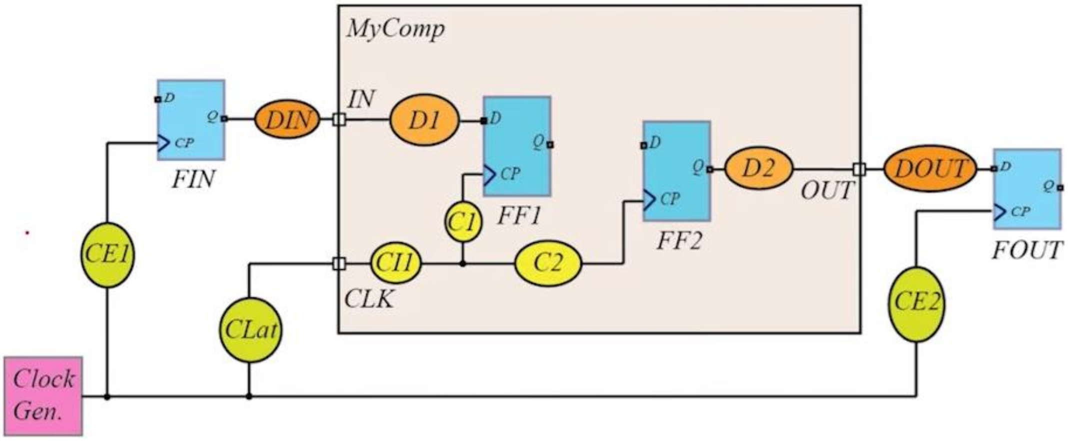
VLSI Design Flow: RTL to GDS

Dr. Sneh Saurabh

Department of Electronics and Communication Engineering
IIIT-Delhi

Lecture 33
Constraints II

Hello everybody, welcome to the course VLSI Design Flow RTL II GDS. This is the 26th lecture, in this lecture we will be continuing with constrict.



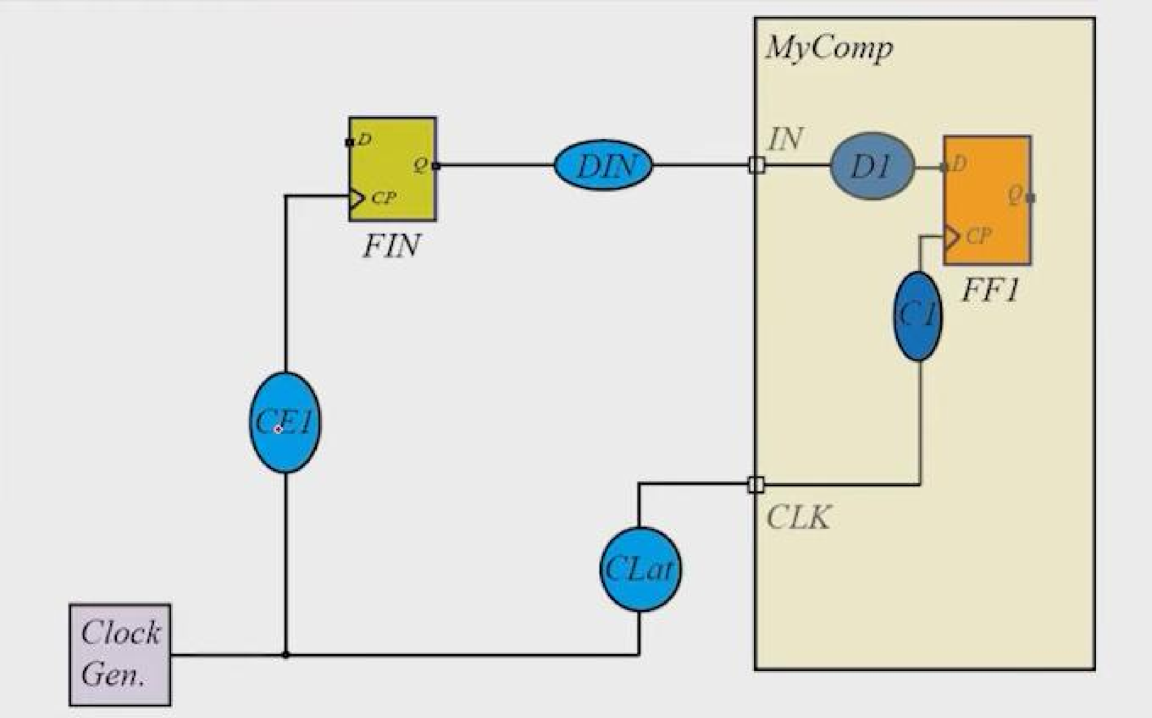
First let us look into what is the expected environment in which our design works . So this is the design, this is the design which we are considering or for which we are writing the SDC file, this is what we are designing at this moment. Now this design will be receiving signals at its input port from the external sources and it will also be producing signals which will go to the external circuit . Now when this is the environment in which the circuit is working that will put additional constraint why? Because we are assuming that the whole system is synchronous there is a clock source there is a clock source which is generating the clock signal and it is used both in the external world and it is also used in our design .

So remember that many times we partition our design into multiple blocks. Then this my comp will be actually our block and the external that we are referring to may be the top level design in which our block is being instantiated . Now this external circuit or the top level design might be operating at the same clock generator and it would be using the same clock generator if you are using a synchronous system . So, now in this system the signal that will be coming at the input port in at this point will already go undergo some delay because of this C1 element the flip flops and the in .

And if they delay the time period that will be available for our internal flip flop to capture the data that will be reduced . That information must be captured otherwise what will happen this D1 delay the implementation tool may make it much larger or allow it to be much larger. And the timing when it is this system my comp is put and made combined with the overall system because of the external delays and that part of the clock cycle that is used by the external circuit elements though because of those delays the setup and hold violation will happen in our flip flop that is in FF1 . So, the signal entering the design at the input port encounters delay and we need to model that. Similarly signals leaving our port should leave the port out and will also encounter delay because of the circuit elements in our design that is my comp and also it will undergo delay externally before being captured .

Now the gen signal that we are generating at the output port will be captured externally . So, and what we want is that wherever it is being captured externally there also the setup and hold requirement should be met by this flip flop f out . So, to meet this requirement we need to inform the tool that is doing timing analysis for our tool that is my comp we need to inform that there must be some constraint on the output port it is not that this delay D2 can be arbitrary . So, to model the effect of data being captured externally we need to add some constraints at the output port . Now what is that constraint we will just see.

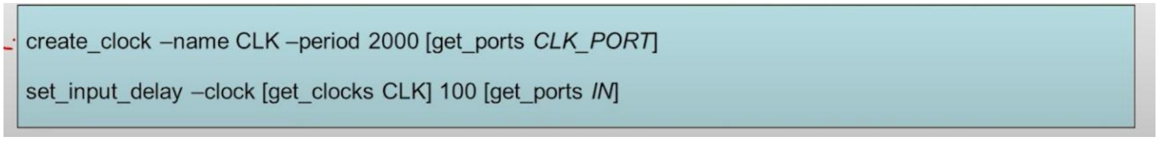
So, in this slide what we are showing is that we need to consider or we need to be to bother about what is the environment in which our circuit is operating or our circuit in my comp in this case is operating. Because the external environment will also impose constraints within our design and that needs to be captured while we are doing the static timing analysis of our design. Because if we do not do that then when our design will be plugged in at the top level there will be a failure ok. To avoid that failure we have to model the external environment and put that in constraint in the HTC file. Now let us look into how we can put those constraints in the HTC file.



So suppose this is our module mycomp . Now signals which are entering our design will get delayed before entering the design . So, it will be delayed. So the lesser part of the clock period is available for the signal to reach the flip flop within the design. So some part of the clock clock period will be taken by this D in and C E1.

Note that the effect of this C at is already being taken into account by the source latency that we saw in the previous slide. So here when we define the clock and specify the clock latency then the effect of C lat is already taken. So we need not bother about C lat but we need to bother about C delay of C E1 delay of this flip flop that between clock to cupid and D in because the data that will be generated by this flip flop externally that must be captured in the flip flop FF1 in the next clock range . So there is only a clock only, so there is one clock period which is available for the data to traverse from here to here and meet the required time requirement at the D pin . So this delay that will be encountered by the signal which is coming at the port in that delay is modeled using the command SDC command set input delay.

Value of the input delay = Delay of CE1 + CP ➔ Q delay of FIN + Delay of DIN

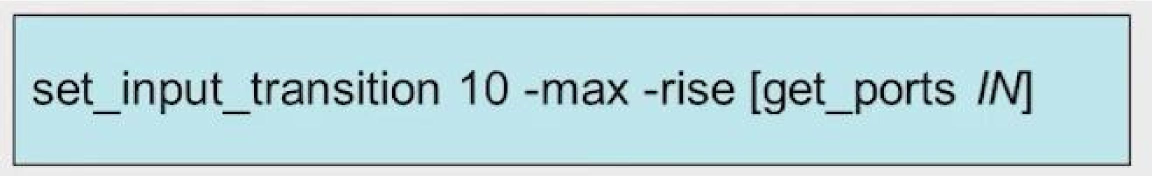


```
create_clock –name CLK –period 2000 [get_ports CLK_PORT]
set_input_delay –clock [get_clocks CLK] 100 [get_ports IN]
```

And how do we model it? So we model it by adding the delay of C E1 the F in of F in the delay bit in between a bit in of the timing arc between clock pin and the Q pin and the delay of D . So these three components will be taken that much then the delay will be there. Suppose this delay was supposed to be 100 clock 100 time units. So first we define the create clock first we define the create clock and the frequency that it is being generated. So the create clock is defined on the port so this port name should be actually a clock underscore port .

So we define the clock to create a clock on this underscore port and we give the name of the clock signal as CLK . And then we write the command set input delay and then with reference to which clock we are referring to the set input delay we may give the name of the clock, get clock CLK and use the option minus clock to define that we are saying that we are defining the clock's clock signal. And then we define the value 100 so 100 represents the delay of C E1 plus clock to Q delay plus D in and then we say that on which input port we are defining this constraint. So this constraint is specified on the input port in and therefore we have written to get underscore ports in . Now at the input side or in the input port we also need to define the transition of the signal.

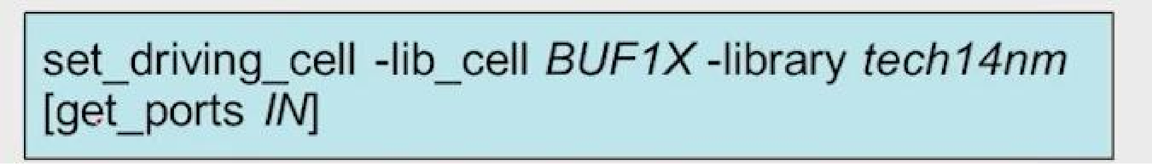
Why do we want to specify the transition of the signal because if we have a design, if we have a design and a signal is coming at this input port to compute the delay of the instances which are directly connected to this input port we need to know the slew of the signal which is coming at it . Now let us remember from the earlier discussion that the delay of say this inverter depends on two things the first is the input slew the input slew that is the slew of the signal how fast it is transitioning and the load the load at this out at this load seen by this invert . So the load can be computed internally by the tool because with this it is known that the attributes of the instances and other things are known by the tool which is doing SDC for which is doing SDA for our design. But the behavior of the external signal that the tool cannot comprehend or cannot derive on its own. Therefore we need to define what is the slew at the input port from the input port then it will do the kind of slew propagation that we have discussed in the earlier lectures and the slew will be known inside our circuit by slew propagation.



```
set_input_transition 10 -max -rise [get_ports IN]
```

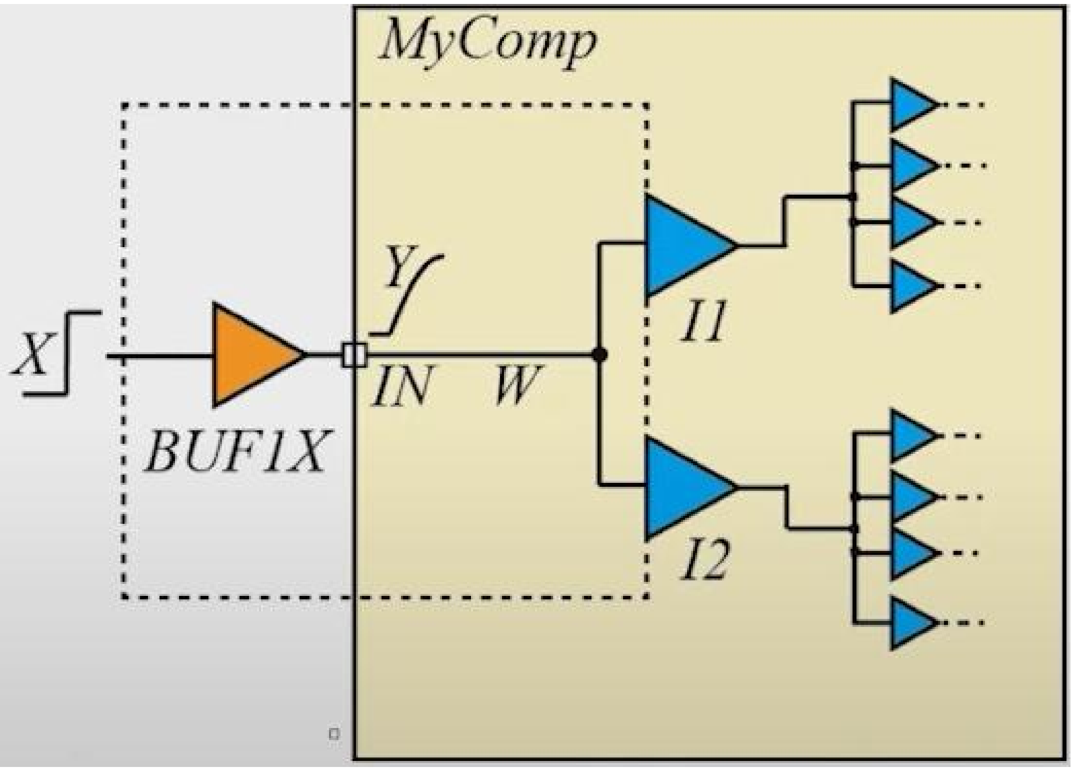
But at the input port we have to define it because the tool cannot derive the characteristics of the signal which is coming . So we use the command set input transition to model slew of the incoming signal . For example we say that input transition 10 on the input port in suppose this the name of this port was in so we say that the input transition is 10 time units meaning that slew of this signal is 10 time units for the that needs to be used for the max case or the setup analysis and that it is for the rise. Similarly we can do for the min case and also for the or for the fall transition so we can define 4 different values if you want . Now sometimes the behavior of what will be the input transition is not known but what we know is that what will be the driver that will be driving the input port in that case we can use the command set driving cell .

So suppose we give the command set driving cell and we say that the cell that is used to drive our input port is buff 1 x and the library name is tech 14 nanometer we can so the set driving cell command is attached to this object that that is port in.



```
set_driving_cell -lib_cell BUF1X -library tech14nm
[get_ports IN]
```

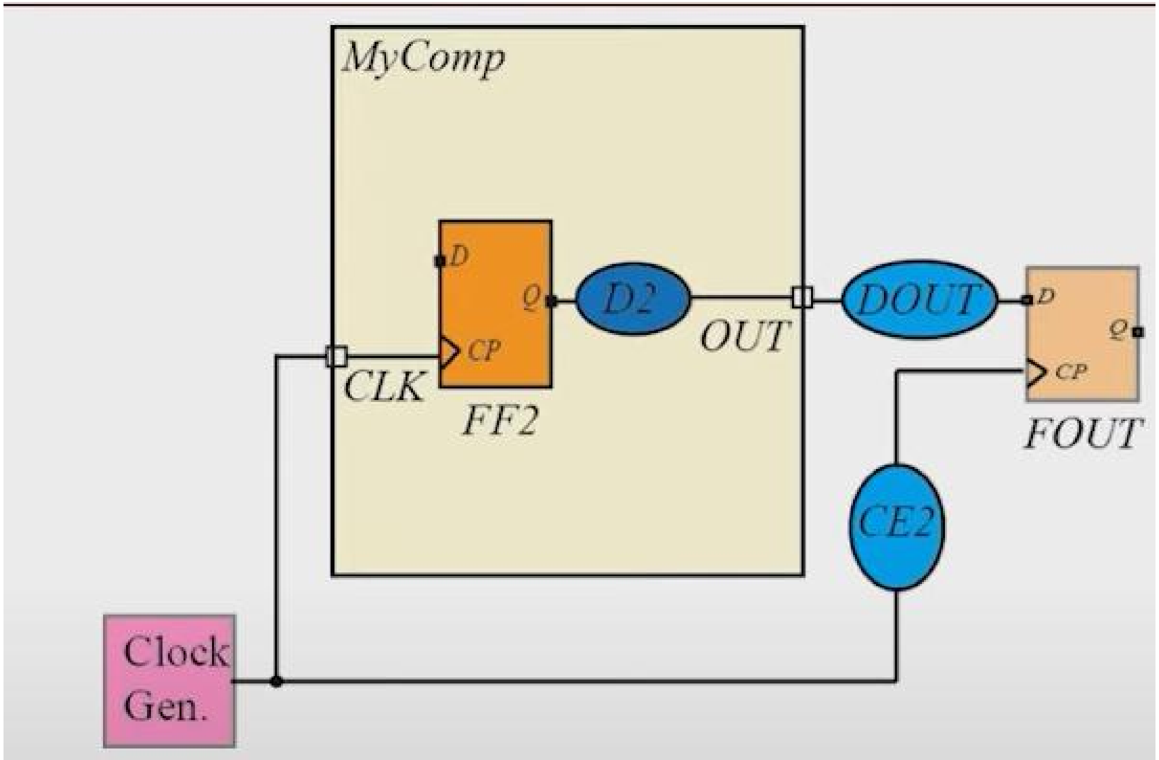
So what it means is that we will have our design in our design there will be an input port like in and then on this input port there is a driver whose name is buff 1 x and it is contained in the library tech 14 nm and that is driving it .



Now while we are discussing the delay calculation we saw that the tool computes delay by creating stages. So in this case a stage will be created based on the wire that we have inside our circuit: the driven pins that is I 1 of the pins of the I 1 and I 2 and a driver which is what is specified in the set driving cell . Now the input waveform at the at x that the tool can can use some default value or an ideal case and using that and the load that is seen by this wire w and the load seen at the input port and the capacitance of the wire w and the input loads of I 1 and I 2 that is pin loads of I 1 and I 2 the tool will compute the waveform at this out input port by making this a stage and creating this stage and doing the delay calculation.

And that this note is this waveform that we obtain at the input port that is y in this case that is we gave an input as x and we got the out the behaviour at the or the waveform at the input port as y. Now this y can be non-linear also the limitation of set input transition command is that it assumes a linear behavior for only for a linear signal we can define a slew . But for a non-linear signal we cannot define a slew and the signal that may come to our design may be non-linear also and in that case a set driving cell is a better option to use rather than set input transition. Because in that case non-linear behavior of buff 1 x can be automatically computed by the delay calculator by using advanced delay models such as ECCS or ECSM. Now let us look at the output side.

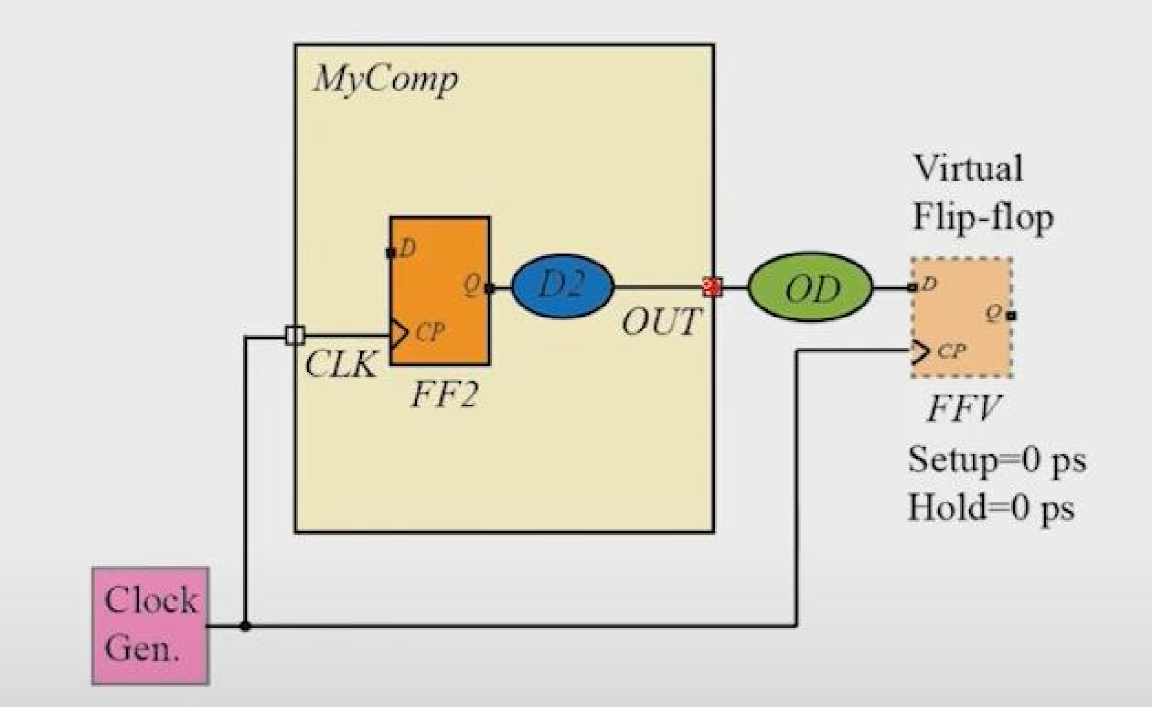
Now at the output side the signal leaving a design must meet the setup and hold requirement of the flip flop that captures that signal . So, the constraints of the external flip flop are modeled using the command set output in and out. So, let us understand how we can define the command set output delay to model the setup and hold requirement of the external flip flop .



So, suppose this is our design, this is our design, these are designs in which we have my comp . So, my comp is our design and it is producing a signal at the output port .

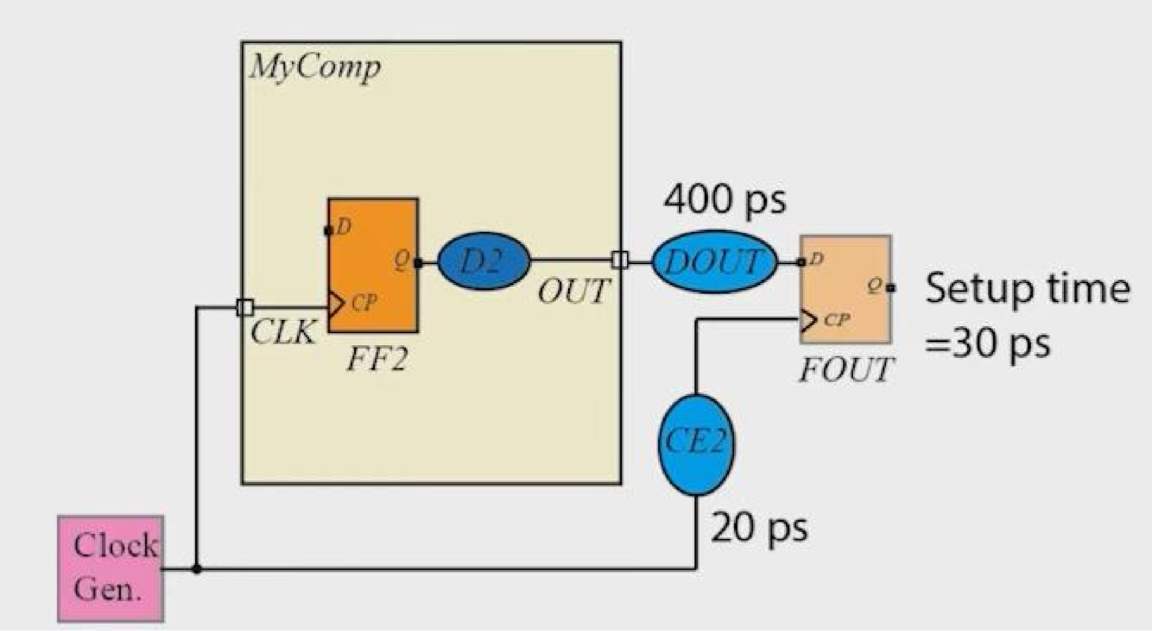
Now in this output port the signal generated at the output port undergoes delay of the out and then it is captured by this flip flop f out . Now the setup and hold requirement of f out must be met . So, that the setup and hold requirement of f out may be met because the entire system consisting of our comp and the external element is a synchronous design or is a synchronous system. So, we need to meet the setup and hold requirement of f out . Now if we do not model it what can happen is that the the the delay d 2 will be unconstrained and therefore, a major part of the clock cycle may be consumed by d 2 and when it reaches d out and then d pin and the d pin of this f out then the required time may not be met .

The required time at f out may not be met and therefore, we have to somehow model the setup and hold requirement of f out at the output port . But the problem is that we add the the SDC file the SDC file is or whenever the ST tool is doing STA the tool is only seeing this part of our circuit this is only our circuit is the tool is seeing when it is doing an STA it has no information of the external environment that what type of flip flop it is that will capture our the data produced at out what will be the setup time what will be its hold time what will be the delay d out and so on. So, the external behavior is totally unknown to the to to the STA tool when it is doing STA of mycomp . So, we must model those effects somehow in the SDC file. So, the effect is captured while doing STA of our design mycomp.



So, what the tool will do in that case is that it will do the timing analysis at the output port out by considering an ideal flip flop which is sitting close to the output port and having a delay of OD which is specified by the command set output delay . So, what we need to do is that while providing the set output delay command we need to choose the value of the set output delay set set output delay or this element OD such that the timing requirement of this circuit and this circuit both are equivalent . So, note that when the tool is doing STA of this circuit it is assuming that there is a virtual flip flop connected adjacent to the output port . What is a virtual flip flop or what and what it is? The characteristic says that its set up time is 0 and its hold time is 0 . So, the simplest thing that the tool can assume is that the set up time is 0 hold time is 0.

So, whatever the effect needs to be modeled for this flip flop that has to be contained in this the value that we specify for set output D . So, the delay of OD this element this element needs to be chosen such that the set up hold requirement in the actual circuit this is the actual circuit and the equivalent circuit this is the equivalent model that the tool will use for doing the timing analysis for the output port out these two should be exactly same . So, the delay of OD is specified in this set output delay command.



So, let us look into this as an example and understand that how we can choose the definition determines the value of OD . Let us assume that this is our circuit, this is our circuit and there is an external circuit which is having a delay of say 400 picosecond on this of D out and the set up time of this flip flop was 30 time units and the the delay on the clock path was say 20 time units .

$$T_{int} + 400 < T_{clk} + 20 - 30$$

$$T_{int} + T_{OD} < T_{clk}$$

Now the same circuit will be analyzed using the set output delay command . Now the set output delay command has got a value of delay which we will be there in the set output delay command and that delay we have to find out we have to find the delay of OD such that the timing analysis of the original circuit and the model that will be considered by the SDA tool those match exactly. So, let us first understand what will be the timing requirement in the first circuit, the original circuit . So, we consider the set up analysis the late analysis set up analysis or late analysis the earlier hold analysis we can do this in the similar manner . So, in this case if we consider the flip flop f out f out then what is the arrival time of the signal at this D in .

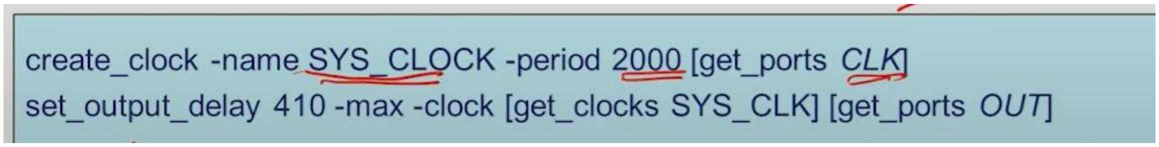
So, suppose the time the clock was generated or the signal was generated at time 0 then it will undergo a delay in this path . We let us call the delay between this clock and this output port as T in internally and then it will undergo a delay of 400 picosecond this is the arrival time of the signal at D . Now for the set up analysis the arrival time should be less than the required time what is the required time. So, if the clock time period is TCLK then the next clock edge will be generated at TCLK then it will undergo a delay of 20 picosecond and then the clock edge should or the data should be stable at least 20 30 picosecond before the clock edge that is why we have to use minus 30 in the required time . So, this is the constraint that we have for the actual set and for the circuit that will be used by the STA tool in the analysis of what will be the arrival and required time.

So, the arrival time will be the same from the internal part and then it will have a delay of whatever that set output delay command specifies, let us call it TOD . So, the arrival time at this point will be T int which is delay that is the signal that undergoes internal to our module my com and TOD is the time that will be specified in the set output delay command . And then what is the required time required time the next clock edge will be generated in TCLK time and we are assuming the virtual clock has set up time of 0 there is no no circuit element on the clock path and therefore there is no delay here and the set up time is set up time is 0 and therefore the required time is same as TCLK . Now what we are saying is that we have to find the value of TOD such that this constraint and this constraint are exactly the same . Now how do we find it? Now T int are the same in both cases .

Now let us assume that the TCLK is on the RHS . So we take this 20 and minus 30 on the LHS. So this constraint becomes 400 minus 20 plus 30 if we take it on the other side and it becomes 410 less than TCLK . And what we have constraint on this side we have tau TOD is less than TCLK. So what it means TOD should be equal to 410 picoseconds .

$$T_{OD} = 400 - 20 + 30 = 410$$

So TOD is 410 picoseconds and we will write the constraints set output delay as follows. So first we will be writing the name of the clock . So say we write the command, create the clock name as sys clock and minus period 2000 picosecond and then the port name clock in this case . So clock here and then we write a set output delay and then we give the value 410 that we derived here and then we say that max. So max will be used for the setup analysis and then we give the clock which is being referred to in this which is basically generating this clock sorry this signal at the output port out.

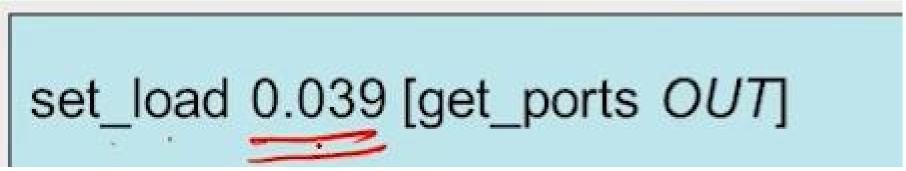


```
create_clock -name SYS_CLOCK -period 2000 [get_ports CLK]
set_output_delay 410 -max -clock [get_clocks SYS_CLK] [get_ports OUT]
```

So the name of the clock is sys clock. the name that we have given here and then we write get underscore ports out where we want to apply this set output delay. So this is how we specify set output delay for the case when the for the setup analysis. Similarly we can do the hold analysis and in that case we will give the option minus min . And this we can also define separately for rise and fall and so on . So these things you can look into or the more options that are available for the command set output delay create clock create generated clock and other commands that we are discussing you can look into the manual of this tool that manual of the tool that you will be using for STA or logic optimization or design implementation.

Now at the output port we also need the information of the load . So suppose this is the design that we are making . Now in this design if we consider the input filter and inverter which is driving an output port out . Now to compute the delay of this inverter we must know what load it is driving externally for example there may be a wire which is connected to the output port before this is being driven by the other or that before it is connected to an inverter which is reading the signal out . So there will be a capacitance associated with this wire and the input pins.

So to do a realistic delay analysis of the instances which are directly connected to the output port we need to specify this external load and to specify this is external load we use the command set underscore load.



```
set_load 0.039 [get_ports OUT]
```

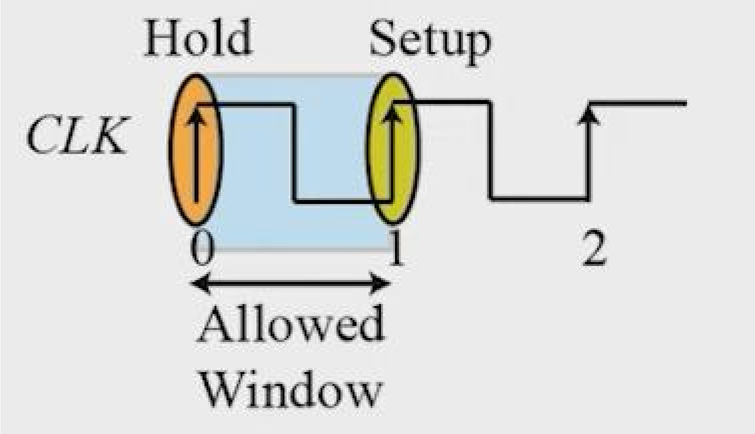
So we can use it for example set underscore load this 0.039 is basically specifying the external load external capacitive load in terms of library unit it may be picofarad or femto farad or whatever the library what is the unit of capacitance in the library. And then we say that on which port we are applying this constraint out in this case so we are applying at this port out. Now let us look at a few constraints which define the functionality of the or which are related to the functionality of our design.

So there are some situations in which we do not want the STA tool to do an analysis, that is that is a traditional STA analysis. So traditional STA analysis is that we are checking for the setup k setup time requirement at the next clock edge and the whole timing requirement at the same clock edge that is the conventional conventional timing analysis or for a synchronous circuit. But in some cases we want the tool to not do that kind of analysis. For example we know that there is a there is a there is a flip flop and there is another flip flop and between these two flip flops there is a path which is connected connection connection is there but the behavior of the function or the behavior of our circuit is such that the signal that is generated from this Q pin will never will be reaching this D pin because of the the functional Boolean function or the or some relationship that exists between the the the logic elements in the path from Q to D pin . And therefore that signal cannot propagate in this path.

Now if that is the case then we can specify a command set false path between these two flip flops and can also mention the name of the instances through which this or the name of the pins through which this path go through the map. So we can specify that or we can write set false command set false path command in our SDC file to specify that we do not do timing analysis for this particular path because the signal is not expected to traverse or propagate synchronously or on this path . So we are making an exception . So if we know that there is a path which where where the where where signal cannot propagate then we can use this command set false path it will relax the timing for that path and may help in improving the improving the PPFR design or it will also help in in in suppressing false timing violations . So if we know that the signal will not propagate from one point to another then if we do not use this set false path command then we may get lots of violations which are actually not needed .

So you know we can use the set false path command to suppress those violations. And then there are some cases where we want the signal or we allow a signal to go from one flip flop to the next flip flop or next sequentially adjacent flip flop in say more than one clock cycle. For example, suppose there was a flip flop and it was producing data and it was traversing through the combinational circuit element and then reaching another sequentially adjacent flip flop. But we know from our functionality that this quipin will change only every set four clock cycles . It need not be held constant for say four clock cycles and then only it will change a value .

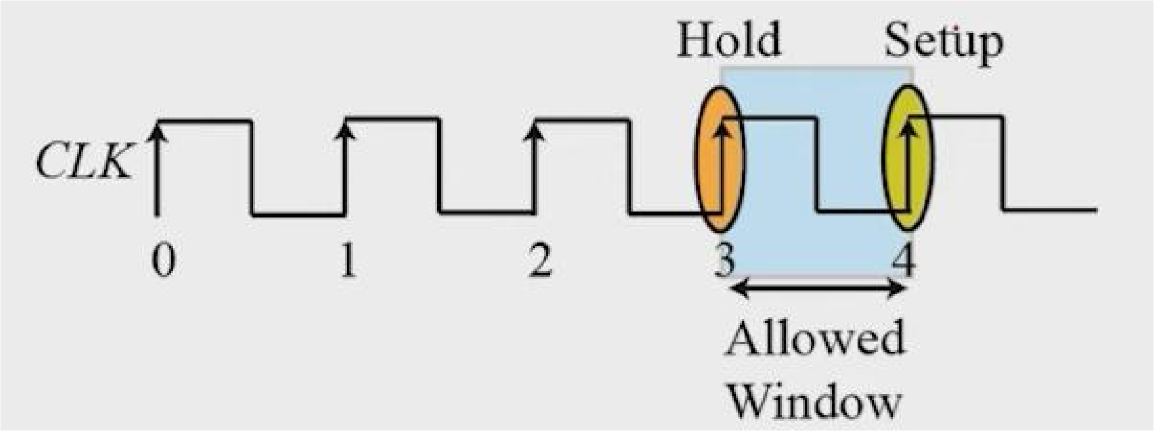
So if we know that kind of functional behavior then we can allow the tool or we can inform the implementation tool and STA tool that for this particular path do not do conventional STA analysis but make an exception. Allow four clock cycles of delay in this path . If we do that then probably the combinational logic between these two flip flops will be not highly optimized for timing and therefore the area savings can be done or P P F of the circuit can be improved. So this is basically relaxing the timing constraints and therefore some Q & R benefit can be expected . So let us see how this set multi cycle path command is used to specify that for a given path we allow more than one clock cycle for data to propagate.



So if we do not use set up multi cycle paths then the behavior of the tool is something like this which is shown here. So the setup analysis will happen in the next clock edge and the hold analysis will happen in the same clock edge. So the window allowed window in which the signal can propagate is from this wind the window that is we are that is highlighted in this figure . Now if we say that set multi cycle path four with a setup of setup time with with the option of manner setup from this FF1 C P this is FF1 so this is FF1 C P pin and to FF2 this is D pin to FF2 D we say that for the setup analysis we allow four cycles to for the data to traverse in in this path . So if we give this command then what will the tool do? It will shift the window of checking as shown in this figure.

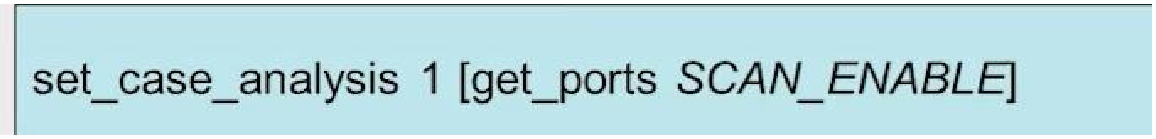
It will check for the setup analysis at the fourth clock edge not at the same as the first clock edge at but at the fourth clock edge. But what will happen to the hold analysis? It will be checked just one cycle before that similar to the behavior of the synchronous circuit. It will be checked at the third clock cycle. So what it means is that if the data starts from this Q pin and reaches D pin say in this in in the second clock cycle then there will be a hold violation because the data is coming before the hold the hold check edge . But ideally what we want is that or typically what we want in our design is that we want to relax the timing constraint.

We want that the data can traverse any any time between the 0 clock edge and the fourth clock edge. We want to expand the window of checking . So if we want that behavior and most of the time we want this kind of behavior then what we want is this kind of

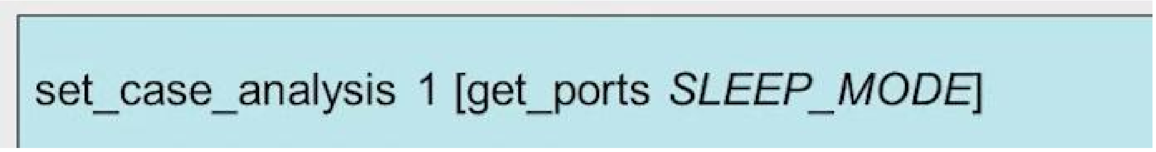


behavior that the setup check happens at the fourth clock edge but hold clock edge so hold checking should happen at the the 0th clock edge only . So in that case we have to write two commands. The first is related to the mannus setup where we give the multiplier as a factor of 4 as earlier.

But we also need to add a command which has an option mannus hold and where we have the path multiplier as 1 less than what we have specified in the setup . So here we have specified 4 and then we specify 3 for the hold check then what it means is that move the hold check 3 clock cycles to the left side and it will come at the side. So if we want our window to expand for the checking then we should use both the setup option and hold option for the set multi cycle path command as shown. Now sometimes we need to apply constant value to the port or pin . So in the earlier discussion we had seen that we do what is known as MMMC multi mode multi corner analysis to account for variations that will happen in our circuit .



```
set_case_analysis 1 [get_ports SCAN_ENABLE]
```



```
set_case_analysis 1 [get_ports SLEEP_MODE]
```

Now we said that for multi mode we have different SDC files and in SDC files we say that some value of or some signal will assume some constant value. For example we may say that there is an SDC file which is for a design say scan dot SDC SDC. In that case in the scan dot SDC file the scan enable signal is taking a value of say 1 and there may be other other changes also involved in the scan dot SDC file. Maybe the clock frequency may be different from what we have there in the functional mode and so on. So this scan SDC file is a separate SDC file and in that if we say that in the scan in the scan mode if there is some signal for example scan enable is assuming some constant value for example the input port scan enable is always having a value of 1 . So in order to specify that we can write in a scan SDC file the command set case analysis value is 1 and the name of the signal where we want to apply that constraint.

Similarly there can be a design a mode in our circuit which is known as sleep mode and we can write a sleep dot SDC for that particular mode and in that case suppose there is a signal or a port whose name is sleep mode that that that signal might be might need to be kept at constant value 1 during the sleep mode . So we can specify it using the command set case analysis. So if we want that some signal may be a port or a pin in our design that is held to a constant value for a certain mode in our SDC file then we specify it using the set case analysis command. So if you want to go deeper into the topics that we have discussed in this lecture we can refer to these references.

So now let me summarize what we have done in this lecture. So in this lecture we have looked into the constraints and the SDC file and how we can apply various types of constraints such as clock constraints, the constraints related to input and output delay and also the constraints related to input output port such as load and transition and also specify some functional constraints such as false path and multi cycle path . Now with this we complete one important part of our course that is timing analysis and timing constraints and in earlier lectures we had looked into the RTL synthesis and logic optimization and at the end of logic optimization we said that we got a netlist in terms of generic logic gates. Now from the next lecture onward we will look into how these generic logic gates are mapped to standard shells and how the timing and power and other considerations are taken into account in the logic in the design implementation or the logic implementation . So we will be looking into technology mapping in the next lecture. Thank you very much. .