VLSI Design Flow: RTL to GDS

Dr. Sneh Saurabh

Department of Electronics and Communication Engineering
IIIT-Delhi

Lecture 36
Technology Library and Constraints

Hello everybody. Welcome to the course VLSI Design Flow RTL-2 GDS. This is the tutorial for the 8th week. In this tutorial, we will be discussing technology libraries and constraints. Specifically, the objective of this tutorial is to understand how delay calculation and static timing analysis are impacted by technology libraries and constraints. And we will be using the open source tool OpenSTA for this.

So what are the requirements for this tutorial? The first requirement is that the OpenSTA should be installed on your machine. So the installation and how to run OpenSTA was described in tutorial 7. So if you have not yet installed OpenSTA on your machine, I will suggest that you install it by taking help of tutorial 7. And we will be needing the following files, the design file test.

v, the script file test.tcl, the sdc file test.sdc and the technology library toy.library. So all these files are available on the NPTEL website as study material for week 8.

So you can download these materials and use them for your experiments. So in this tutorial, we will be covering the concepts of the library which was discussed in lecture 21. So let me just recap the nonlinear delay model or NLDM which was discussed while we were covering technology libraries. So NLDM basically models the delay of a timing arc. For example, the timing arc between an input pin i and the output pin Zn of an inverter.

And the delay is considered as a function of input slew or input transition and the output load or the output capacitance Cn. And it is modeled as a two dimensional table. On one axis, we have the loads, various loads or capacitances and on the other axis, we have the input slew or transitions. And in this tutorial, we will also be covering the concepts of constraints. So we will be covering the concept of creating a clock, set input delay, set output delay, set input transition, set load and set clock uncertainty.

Now let us run OpenSTA and study the impact of library delay and constraints on static timing analysis. So first let us see whether we have all the files that are needed. So we have the SDC file, the TCL file, a script file and the test.v, the design file and the library toy.

lib. Now let us open the design file test.v. So it contains one module that is named at top which has got two ports, one input port A and the output port out. And it contains only one instance, an instance of a cell named inv. So inv will be a cell inside the technology library toy.

lib and the name of the instance is i. And the input pin of the inverter is driven by the input port A and the output pin of the inverter drives the output port out. So this is a very simple design. Now let us look into the technology library toy.lib and see the delay model corresponding to the inverter inv.

So I am using the editor gwim, you can use any other editor to open these files. So I go to the cell inv, so the cell inv or inverter is having a timing arc. So this is a timing arc and it is defined at the output pin Zn and the related pin is i, meaning that this is a timing arc from input i to the output pin Zn. And the delay are specified here and at what values or what are the characterization points. So the characterization points are defined in this template, so timing underscore template.

So these templates are described at the top of the library. So let us go to the top of this library and see the template. So the characterization points are for the input net transition or input slew; the values are 0.1 and 100 in library timing units and the characterization point for output net capacitance or output load is 0.1 and 100 in library units.

Now let us go into the NLDM table and see what the values are. So we go to the cell inv and corresponding to that we see the NLDM table for the delay. The other NLDM table this one is for the slew output slew, so we are not considering that in this experiment. So the delay table is this one and the values are 184, 200. So this is a toy library and therefore the characterization points are only 2 for slew and 2 for load.

In realistic design there will be say 8 characterization points or more for slew and similarly for load. So now corresponding to this NLDM table we can draw a table and for our easy analysis. So I have drawn that table and it will look something like this. So on the rows we have various transitions 0.1 picosecond, 100 picosecond, so picosecond is the library timing unit and on the column we have the capacitance 0.

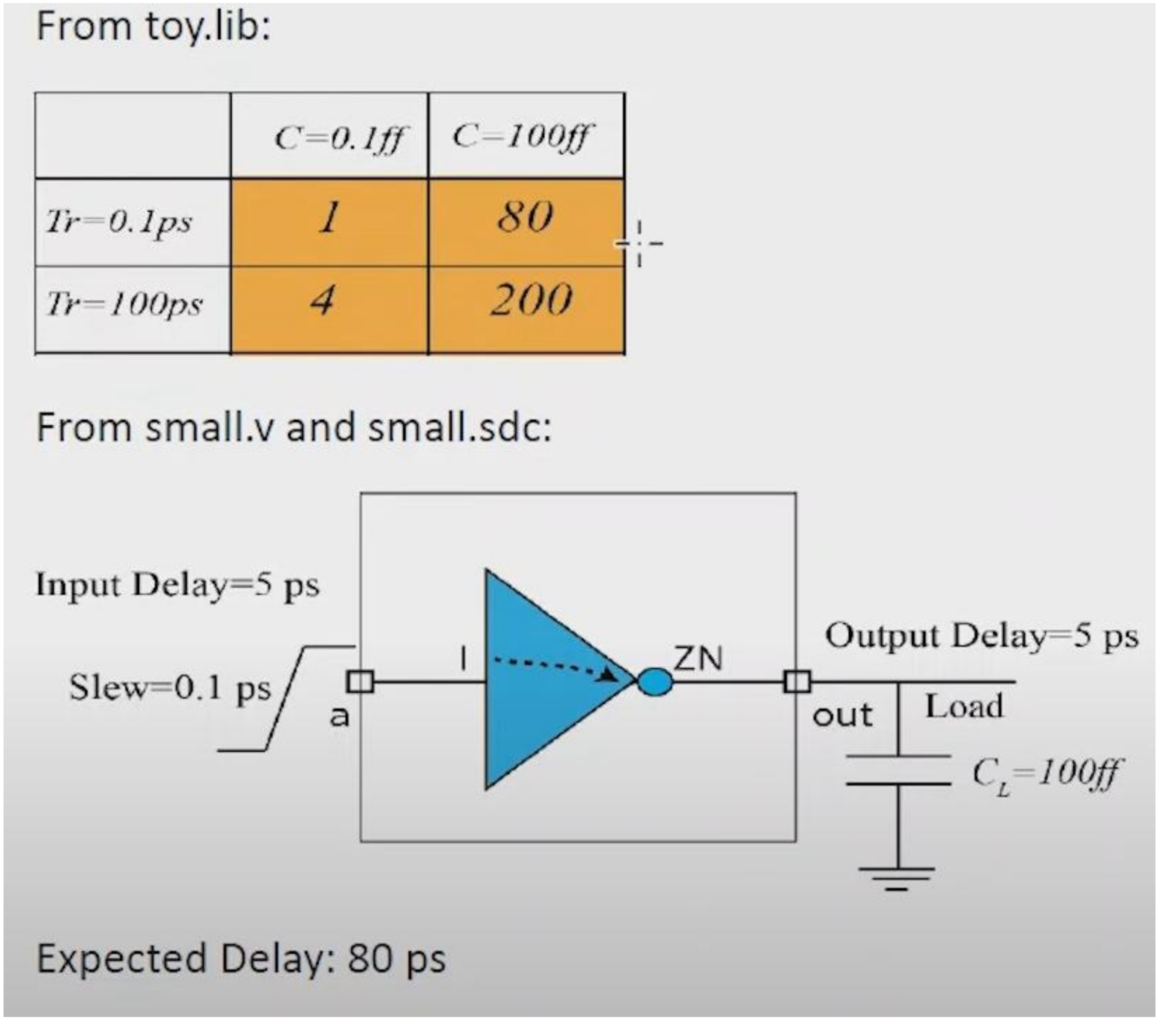
1 femtofarad and 100 femtofarad. So femtofarad is the unit of capacitance in the library

and 184, 200 these were the values that were defined in the NLDM table. Now let us look into what are the constraints that we are specifying for our design. So we open the file test.

stc. So the first constraint is to create a clock. So we are creating a clock, the name of that clock is capital CLK and the period or time period is 1000 picoseconds. So there is no port or pin associated with this clock and therefore this kind of clock is known as virtual clock and we have created this virtual clock just for constraining our input and output. So we are specifying the delay of 5 time units at the input port A and a delay of 5 time units, delay of 5 time units, 5 picoseconds at the output port out. And we are also setting input transitions of 0.

1 at the input port A and load of 100 femtofarad at the output port out. So now using this information, this information input transition and load and NLDM we can compute the expected delay of the inverter. So let us see how we can do this. So we have the circuit, the design and the constraints represented here. So we have an inverter and the input slew is 0.

1 and output load is 100 femtofarad. Now if we refer to the NLDM table corresponding to input slew of 0.1 picosecond, the row, the first row is there and corresponding to the load of 100 femtofarad there is the second column. So the first row, second column the value is 80. So the expected delay of the inverter or the dimming arc I to Zn will be 80 picosecond.



So now we can carry out an experiment and see that indeed the delay is coming out to be 80 picoseconds or not. So let us see what the script file is. So that we open a file test.

tcl. So in the test.tcl first we are reading the lab library using the command read liberty and the name of the library is toy.lib. Then we are reading the Verilog design test.v and then we are linking the instances of which are there in test.v or design with the cells which are present in the toy.

lib that is the inv cell is being linked that is for that we are using the command link design. And then we are reading the sdc file test.sdc and then we are reporting the timing for our design. So now let us run the STA tool, open STA by using the command STA since STA is open STA is already installed on my machine I can run using the command STA. Then what I do is that I take a test.

tcl file. So all the commands that were in the test.tcl file will be run one by one. So it will load the design link with the library and then report the timing based on the timing constraints. So this is the timing report. So we can see that because of the set input delay there is a delay of 5 time units coming here and then we see that the inverter delay is 80 picoseconds as we had computed.

So 5 plus 80 is 85 that is the arrival time we are getting at the output port out. And what is the required time since we expect that the output should come by the next clock edge and the time period of the clock is 1000 picosecond the required time is 1000 picosecond. Now out of this required time of 1000 picoseconds we need to subtract the time of the output delay that we expect will be the external delay that will be there when our circuit will be integrated at the system level. So we subtract 5 picosecond from 1000 and we get the effective required time as 995 and the arrival time was 85 picosecond. So if we subtract arrival time from the required time we get the slack as 910 picoseconds.

Now we can carry out many experiments by changing the constraint and other things. So let us carry out a few experiments for example let us consider that instead of input transition as 0.1 what will be the effect or what will be the delay if we change the input transition to 100. So if we do that then what do we expect so we can easily understand it with the help of an NLDM table.

So we see that instead of 0.1 picosecond if it is 100 picosecond then we have to refer to the second row and the load is still 100 femto farad so the column is second so the second row second column value is 200. So we expect that the delay should not change to 200 picoseconds. So now we copy this command set input transition and paste it. Now transition is 100 at input and then we do report checks to report that so that tool will report the time. So we see that now the delay of the inverter has changed from 200 to 80 picoseconds, the arrival time has increased and therefore the slack has decreased.

Now let us carry out one more experiment: what will happen if we set load to 0.1 instead of 100 femto farad if we set it to 0.1 what will be its effect. So using an NLDM table we can see that if the output load changes from 100 femto farad to 0.

1 we have to refer to the first column. So the first column is second row second row because the input slew is 100 picosecond so the value here is 4. So we expect that the delay should decrease to 4 picoseconds. So we copy this command and paste it and then do report checks. So we see that the delay has now decreased from 200 picosecond to 4 picosecond as a result the arrival time has decreased and slack has increased. Now let us carry out one more experiment related to set input delay.

So earlier the set input delay was 5 time units. Now if we change to 25 time units then the input will be delayed by 20 time units. So we are specifying set input delay 25. So in this case what will happen is that the input will be delayed by 25 time units. 20 time units more than earlier and therefore the slack will decrease by 20 time units that is what we expect.

So let us copy this command and paste it. And then do report checks which expect that the slack should decrease from 986 to 966. So we run this command. So we see that now the input delay has increased to 25, arrival time has increased and the slack has decreased to 966. Now what will happen if we increase the output delay right in from 5 picosecond to 35 picosecond. So in this case again what will happen is the required time will be adjusted in this case.

So we expect that instead of 5 picoseconds here at the out will have 35 picoseconds and therefore the required time will decrease and therefore slack will decrease by 30 picoseconds. Here it was 5. Now we are specifying 35 so the difference is 30. So slack is decreased by 30 it should become 936. So we set this command, we run this command and then do report checks. So we see that the output delay is changed to minus 35 and the slack has decreased to 936.

Now let us carry out one more experiment. What if we apply a clock uncertainty of 100 picosecond right. So if we apply that then we expect that the timing analysis will become more pessimistic right. So the time it will take or required time will be decreased by 100 picosecond and therefore the slack should decrease by 100 picosecond it should become 900 and sorry 836 picosecond. So we do report checks. So we see that there is a contribution of clock uncertainty of minus 100 right and to the required time the required time is decreasing and the slack has decreased.

So this concludes the experiments that we wanted to carry out in this tutorial. Now let me summarize what we have done in this tutorial. So in this tutorial we studied the impact of technology libraries and the constraints on delay and also on static timing analysis. Now in this tutorial we have taken a very simple design consisting only of an inverter right. Why have we taken a simple design? The reason is so that we can analyze it easily right.

But I will suggest that you carry out more experiments with realistic design and do the analysis of the result. Once you do the analysis of the result then you will be able to link the concepts that are discussed in the lectures with what we observe in the experiments using tools. And once you are able to establish that linkage then you will become an excellent VLSI designer. So all the best. Thank you very much.