VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Week - 12

Lecture 60

OpenROAD Physical Synthesis Flow – II

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss about open road physical synthesis flow part 2. So, this lecture will discuss about remaining part of the open road flow. Basically, we will concentrate on some of the part which is left out in the part 1. If you can basically look into this picture, we discussed all these in the part 1 of the lecture. This is all discussed in the part 1 of the lecture and the part 2 will basically cover placement, then the clock tree synthesis, then the routing, then we have all these layout generation all these. So, we discussed all the four points in this lecture.

There are some shortcuts which is used for open road tool. Basically, for copy you need to select the left mouse plus select command from the TCL. Paste TCL command, you use the mouse center bottom or the wheel middle wheel actually. And for physical layout, there are two shortcuts are there. One is for full screen, you have to press F, which is easy and for zoom in, you press Z or control scroll, it will zoom in. Then the third one is basically shift plus small z or control plus scroll, wheel backward mouse, it will do the zoom out. So, these three are very useful doing editing and doing the basically being viewing the layout in the open road. Now basically in this part, what we are doing is that we are discussing the global placement. So, the global placement what we are doing here is that we have some basically layer we need to provide. For example, here if you can see here global routing layer adjustment.

So, what it does is that it is a command that sets the routing resource adjustment in the routing layers of the design. If you give these commands, then that will reduce the number of routing tracks that the global router basically assumes to exist actually. So, it does an adjustment of the routing tracks actually. So, this basically sets this one using this command. Then here what we have to set that set routing layer, what are the routing layers are needed for doing the global routing. So, that why this is needed, if you set here whenever you are doing the global placement, it will estimate the wire length. This wire length will help us to find the basically timing of the design actually. So, it will do a estimation while doing the global placement.

So, that is the reason we are setting the routing layers in this place. So, the global placement can be of various type, but here it is basically routing driven. It wants to reduce the wire length actually, but it can alSo, be timing driven also. It can be have another point you can make it timing driven, but here in this case it was routing driven. Then you have density. This is regarding self-padding information. How much left and right self-padding to be placed or preserved to avoid routing condition. Padding means that how much area left and right of the cells should be kept empty to overcome the congestion issues due to the cells high pin density. So, first we will run till this global placement. So, I can remove the metal lines. So, we have removed the metal lines. Now if you run this one what is happening? If you run this one what is happening? So, main target is the global placement. If you can see here some of the cells are placed in the design. Lot of space are empty. We can optimize the area alSo, if you can, but if you can see you see a very good point here what we taught in our class. This is HPWL, half perimeter wire length model is used to do the estimate of the wire length.

So, we discussed about the half perimeter wire length model when we are discussing the wire length estimation. So, this tool is alSo, using the HPWL half perimeter wire length model. So, if you can see this command is not always essential, but if you want to add some padding in the left and right side of standard cells of the design. So, in that case you have to add what is the advantage of using this hyphen pad command is that it will help in global routing or detailed routing because sometimes there is a congestion. So, this padding will allow more tracks for availability for routing. So, this is not a mandatory thing, but you can add it based on your requirement.

So, now basically we have to do this pin placement. We did a random pin placement earlier if you can see here. This is the random pin placement. We did it earlier, but now we are doing the final pin placement. If you can go here place underscore pin, then which routing track horizontal, HOR is horizontal, VER is vertical. So, then we will run this command. So, if you run earlier locations are here and it will change after doing this pin placement. You see this pin locations are changed after this placement. So, it is more proper location for the pins compared to the previous random assignment of the pins. So, this pin placement what is happened here, it is based on the signal routes available in the standard cell library, which is basically closer to the cell. However, if you have any kind of constraint to keep the pin in one direction, then you can write a file where you can specify the coordinate of the pins.

There you can write the coordinate and the tool will do the placement of the pin based on your requirement, based on the designer's requirement. Sometimes there is a requirement of routing a pin to the another blocks. In that cases, there is a requirement to put all the pins in one direction. So, during that time you have to specify the pin locations in a file. So, now, then we have some checkpoint where we can save the complete design using a DB file. So, this is the basically we are storing that place global placement database in a file here. After the global placement, we are saving the data in a file. So, this is very essential sometimes if your design is very big, you do not need to repeat the steps, you can work from here itself, you can load this database and work from here.

So, it is very essential to save the design in the intermediate step. If you have any kind of issue, you go back and modify, then you do not need to redo all the steps again and again. Now we will go to the next step. Here you are doing the repair max and sleeve cap, handout violations and normalize sleeve. So, now we have completed the global placement. Now we are entering the detail placement. The second phase is the detail placement. So, in the detail placement, we need parasitic extraction to estimate the timing, whether our timing is meeting or not. So, that was done using this first one basically RC extraction.

So, we are doing the RC extraction of the estimated wire length, whatever it is coming, we run that. Then estimate the parasitics. We are estimating the parasitics here. After we estimate the parasitics, now we can repair the design using this command to improve the timing. So, here what it is doing is that it modified the placement of the shell to repair the design such that my timing is met or not. You can see some of the cells have changed after running this. Some of the shell sizes are alSo, changed. It will not change the functionality of the design also. It will just change the sizes to fix the timing. So, here as I discussed in the course, we have tie high, tie low cells are there. See tie high and tie low cells are basically added in this flow. So, this will be inserted in the design. So, this is a step for inserting the tie high and tie low shell. So, this part is done.

So, what is in the detail placement it does is that it legalise the placement. If you run this command, all the cells, if you can see there is a overlap of the cells, but that overlap of the cells are not allowed in the final layout. So, then it will placed in a particular row and do the routing. So, the legalisation of the placement what we discuss in the course that is done in this step actually. Now, none of the cells are above other actually. There is no overlap of cells. So, this is a half perimeter, one length estimation model, legalised half perimeter, one length estimation model. So, this is basically these are the reports whatever we found from the detail placement. So, that this is the detail placement is over now. Now we will basically do that some of the timing checks here basically to check that what is the basically whether we do not need to resize some of the buffers or not. Post resize timing report.

So, here we are doing the, So, what we discuss in our course we have discussed the hold violation and setup violation. So, it will report the mean timing violation in the first line. So, these reports will be useful for resizing the cells, resizing the cells to fix the any kind of violation. So, this is for this is for the hold slag, worst case hold slag. Hold check is normally done post CTS as it is independent of clock period. This pre CTS that is before CTS hold check can be done to understand with zero skew how good the hold slag and quality of the hold timing constraints are defined. So, here we are checking the timing before clock tree synthesis. However, we will again do the same thing after the clock tree synthesis. So, now we will do the for the max timing constraint. So, this is the worst case slag. Then we have total negative slag in 3 digits. Total negative slag is 0.

So, then what we are doing is that we will run this part to repair any kind of resizing of the cells because this report whatever it is generated that will be used to repair the design to fix the slags negative slags actually. So, now this part is over. Now we will go to the CTS, clock tree synthesis. So, now we will look into the clock tree synthesis. So, here what we are doing is that we are adding buffers, inverters to make our basically skew in the design as minimum as possible. There are several trees are there. Each tree based routing is there. There are several routing algorithms there like MMM methods of means and medians. Then you have geometric matching technique. All these are there for clock tree synthesis. But clock tree plays a very vital role while designing any kind of chip because the skew will take away our margin. So, we want to make sure that the skew is as minimum as possible in our design.

So, here this is basically the steps to insert the clock tree. So, if you can see here your clock tree is inserted in the design. 5 clock buffers are inserted and minimum buffers in the clock path is 2, maximum buffers in the clock path is 2 like that 5 buffers are inserted in the path. Now we will go and repair the nets. This is the repair clock nets. This step will do the repair clock nets. Now you have to do that detail placement again. Now then again we will save the design after the CTS. So, we are saving the design at different stages of the flow such that if I want to do the design again, I can reload the design from for that point and do the corrections because the physical synthesis flow is not possible to close in a single run.

So, more runs are needed in doing that flow this flow. So, we are saving it in the intermediate step. Now we will come to the setup and hold repair step actually. If there is any kind of violation is there, how we can repair that. So, in this case we have basically set propagated clock for all clock we are setting that then we will do this step then there is a repair timing. So, this is for basically repair any kind of errors in the timing. Now after this we have post timing repair. What is the report timing report? This is the max, min and total negative slack, max, min, first will tell. So, you have this worst lag is there. This is for the hold and this is for the setup, min and max. Then we have report total negative slack. This is zero that is good there is no negative slack. Then we have basically report and if any kind of repair is needed this will be done by this commands actually.

So, this is all about setup and hold timing repair step. Now we will move to detail placement. After the clock tree synthesis any kind of legalization all these we can do it here. So, this step basically here there is two things after you legalize and all these things are done then you can save the design again after this detail placement and which is already the clock tree is already synthesized and the design is saved. Now there is a step we can generate a verilog file. So, whatever it is designed we can write that thing to a verilog file. So, it will create a verilog file with the clock tree and the actual design together in the single verilog file. When we are using the basically gate level netlist generated from the logic synthesis tool like YOSIS there is no clock tree was there. But now this generated verilog will have more gates compared to the netlist generated from the logic synthesis tool. Now we will discuss about the global routing. Now we will discuss about the global routing.

So, we have global routing is basically here what we are doing is that we set the minimum and maximum routing layers and we will do that global routing in this stage and then write the output to a verilog file. This step is done. After this we discussed about antenna violation. If a long interconnect is connected to the gate of the transistor then antenna violation will come. So, we can solve that issue using this command actually. So, in this design there is no violation after the antenna solution.

So, if you can see here there are lots of empty spaces there in the design. So, that need to be filled with some shell. So, that is called the filler shell. The filler shell is a physical shell actually. So, it will create a continuity in the supply line and alSo, base and lower metal density. All these will be maintained using the filler cells. You can see the design is changed and all the gaps are filled with them filler cells actually. Then you will do that write the database to a file after the filler shell insertion. Now we will go to detail routing. Detail routing basically has several steps. So, we can have run pin access again after inserting the diodes and moving the shell. So, we will do that pin access. That is the first step. Then this is actually the detail route detail underscore route. This is a command.

So, we will set this one. We will set this thing. Then we will do the detail routing. It will take some time. Now it is done. You can see So, many metals are connected. It is very wonderful to see all the lines are connected to the required places to do the complete routing of the design. It is very interesting to see that how the algorithms inside the tool is useful in doing the route or connect the routes in such a less time. So, this is the very time consuming work if you do. Do this one manually. It is related to some fixing some of the errors. Now we have some check antenna and write the data into another database actually. Like we are intermediate step we are writing the data to the database. Now the detail routing is over. Now we have two more steps left out. One is basically related to extraction of spec file which is useful for your STA tool. So, we need to do spec file extraction which will be needed for timing analysis tool. So, we will do that extraction in this step. It is basically parasitic extraction which will be taken into account while doing the timing analysis.

So, this is very important to close the timing. After this then we have to check the reports all the reports like if you have a min, max, min is related to hold, max is related to setup. All these we need to check. So, we need to report all these. Now if you have any kind of floating nets all these will be reported in this command. Total area is this much. So, then you have these are the some of the fixes for setup and hold violation. If we are running incremental optimization this metric values are feedback to the detail route engine and does incremental optimization. So, here some of the files which is generated like spec file and the verilog file and the GDS file what is generated from here that will go to let us say the GDS file will go and you can check in some kind of layout editor to check that whether it is DRC, LVS, that is one step related to physical verification like DRC, LVS check. Then there is some kind of thing called timing verification what you need to do we need to take that spec file what is generated which contains the parasitics of the interconnect all these information that will go into a static timing analysis tool, a state tool to do the timing analysis including the interconnect into account.

So, these are the files which is come out from this physical synthesis that will be useful for your further analysis. So, we can have some features here like we can see the clock tree how it is inserted and what is the skew in the clock tree all these information. So, this is the leaf nodes actually of the clock tree and the clock is entering through this buffer and it is going to the leaf nodes. So, you can see the clock tree graphically in this location. Then alSo, you can see the timing report alSo, in a graphical format. So, you can check the timing report one by one by clicking one of them.

So, like that you have to go one clock at a time like this you can go change that and the corresponding signals are alSo, you can check online if you can see here the signals are alSo, changing how the signals in the GUI it is changing that you can see. And second third point is that you can see all what is the top level actually what is the physical only cell, what is the filler cell, how many filler cells are instantiated, how many taps cells are instantiated, what is the number of combinational gates, what is the number of sequential gates, what is the number of clock buffer, all these information are provided in here. So, there is one more concept called heat map actually you can see this heat map for example placement density you can see here. Then you can go to the power density this is one map then you can see the routing condition as a map. So, these things are essential for doing the analysis.

In this lecture we discussed the remaining part of the open rod demonstration.

Thank you for your attention.