VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Week - 12

Lecture 59

OpenROAD Physical Synthesis Flow – I

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss about the open road physical synthesis flow. It is divided into two parts. First of all, we discuss about the part 1. The content of this lecture includes what is physical synthesis. Then we will discuss about the open road, which is an open source physical synthesis tool. Then we will go for a demonstration. So, the physical synthesis happens here. So, basically it will take the gate level netlist as input and it produces a GDS as output. This is the GDS as output. So, this is your physical synthesis. So, it takes gate level netlist as input and produces GDS as output. So, what is the input to our physical synthesis tool here is that timing library file and this is the gate level netlist coming from the logic synthesis tool. Then we are setting some timing constraint. This is a user defined. Then we have a script for basically doing the physical synthesis.

Then we have standard cell lef file and you have a basically a tag file which carries the information related to RC resistance and capacitance of different metals. Now outputs are gate level very long netlist which carries the gate level netlist and other information like buffer insertion in the clock tree all this information is there and this netlist will act as schematic whenever you are doing LVS. Then we have LEF, DEF or GDS which will be used for basically physical verification and if the design is correct then you can send this for tape out or fabricating the chip. Then we have a parasitic information file which will go to the static timing analysis tool, STI tool.

Then we have a timing reports coming from physical synthesis tool. Then finally, we will take the output of the physical synthesis for logic equivalence checks between the RTL to netlist or netlist to netlist. There are lot of other sign of checks like low power checks, EM and IR and many more can be done with the outputs from the physical synthesis files. So, this is very top level of a physical synthesis tool. So, if you can look into this it is a very complex picture, but it is very simple. So, if you can see you have a RTL which is going to basically Yosis and it does logic synthesis here. It produces the gate level

netlist which is going as input to the physical synthesis. It has floor plan, power plan, placement optimization, clock tree synthesis, routing optimization and finally, we will get the layouts which can be viewed in a open source tool like magic layout editor tool like magic. So, here if you can see you have a synthesis to complete finishing of the design all the things are there, but in this lecture we are discussing floor planning. Then we are discussing I O placement, random I O placement, timing driven placement, macro placement, tap cell and wheel tie cell insertion and routing of the power network all these will discuss in the part 1 of the lecture. So, this is the top level script actually we discuss in the demonstration. So, finally, we will get the layout physical synthesis layout. So, in this lecture we will discuss about the open source tool that is called open road. This open road is basically a tool for doing the VLSI physical design where we have basically using the gate level netlist what is generated from the logic synthesis tool like Yosis that will go inside this open road to generate the layout final layout of the design. So, how can you open that tool we use open road hyphen GUI, hyphen log and you can give any log file, but here we are giving this log file where the output of the all the information are stored in the log file. Now after running this one we will get this window. So, this window will run all the commands and if you can see here, So, if you can see this window you have display control is there this is one part of it then you have clock tree and viewer is this is one portion, but there is one most important part is the TCL command actually. This TCL commands are very important where we basically run all the commands in this window. So, we will look into the first file and this file is the complete file which is used for the physical synthesis actually. So, we will start with the first file there are some source files are needed for setting the variables. So, here helper dot TCL flow helper dot TCL this is a NAND 45 nanometer technology node. So, that one and our design name is map 9 V3.

So, top level module is map 9 V3. So, then we have this very log file which is we got it from the basically Yosis this map 9 V3 underscore final dot V it is a gate level netlist. So, the then we have a STC file which is the synopsis design constraint here we have that file is provided to give the constraint to the design. Then we have a chip area or the die area is 0 0 is one of the corner and this 0 0 is the LLX, LLY and URX and URY is 82.13 and 110.8. So, this is giving the LLX, LLY and URX and URY two corners of the chip. So, similarly this is a core area inside that die area we have a core area which is 10.07, 11.2 which is the core area LLX and LLY and similarly 70.25 and 101 are the URX and URY of the core area. So, till this point we will run the script. So, it is basically setting the variables. Now if you go here okay we are running this variables from here design then you have yeah. So, from here first one is the helper dot TCL if you can see this side alSo, both the side the helper dot TCL is a first command. Then we are going to flow helpers article then we are going inside the NAND gate library NAND 45 dot var then we are going to the design what is the design name MAP9V3 then the top level module and this synthesized Verilog then SDC file then the die area and the core area finally the core area.

So, till this point we will run okay. So, now then this is all setting the variables okay after that we will go into the flow this flow dot TCL is another file which is having lot of information okay. We will go into that flow dot TCL file we will go to the flow dot TCL file this is a file. So, if you can see it has all the steps of VLSI physical design whatever we learnt in this course okay. So, we will set some of the things here okay this is the basically graphical user interface for the tool and this is the script. If you can learn all the steps of in this script you will get a understanding about how this physical synthesis is working okay. So, first one is the basically read library okay and we are linking the library and we are reading the Verilog file using these commands actually okay. So, here we are reading the library reading the Verilog file linking the design to the top level module and reading the synopsis design constraint file okay. So, if you can see here So, I am running this the first set of commands. So, now this Verilog file is read by the tool then we will after this is very important because this Verilog file whatever it is written here that should be understood by the tool. If you have some mistake in the Verilog file then you will get lots of error in this first step itself okay.

So, this is very important step then we will go to the basically two commands like you have ORD and STA those are related to open STA tool okay. So, these are passing some of the variables to that open STA tool okay. So, we are running this those commands here. So, if you go up this is the utility metric IFP then this is the second ORD for underscore version then the second is the instance count. Instance count is basically giving the number of instance in that Verilog netlist okay. So, now this all are related to initialization till this point we are basically giving the inputs to the design. Now the first step starts with the floor plan okay or whatever we discussed in the due course actually the chip planning okay. So, if you can see here you have initialized floor plan okay. So, this initialized floor plan will come okay.

So, if you can go here we have initialized floor plan then in this one we are having a site coordinates actually then you have a die area and the core area. See this die area and core area is mentioned in that file okay. I can go back again to show you first of all I will run this and I will go back where the die area and core area is mentioned in which file okay. This is the my floor plan if you can go and see that there is a file. So, if you can go here this here the die area and the core area whatever I discussed just a few minutes before is mentioned here and this is acting as a variable dollar die area whenever I am writing in TCL it will copy these coordinates to that file okay. So, dollar die area will be assigned to this die underscore area variable. Similarly dollar core area is assigned to this core underscore area variable. So, here we are mentioning that thing actually. So, this one will be copied actually okay.

So, now we have already discussed this part how to do the floor plan. Now if you can go to that open road you can see a floor plan here. So, if you can see here So, there is a control and middle scroll if you do you can zoom in and zoom out okay. So, if this corner

is the die area this corner is the die area but if you zoom out this is your core area this coordinate is basically the core area LLX LLY and this is the die area LLX and LLY. Then similarly the core area URX and URY is this point okay and die area URX and URY is this point okay. So, this is the die area this boundary is the die area and this boundary is the core area. Now what we are doing we are going and So, these rows whatever the green color lines what is there I can zoom it out. So, these rows are basically the placement rows okay where our standard cells will sit okay. So, here I can show you where the standard cells are there. If you can go and if you can see our Verilog file you can see this is our Verilog generated from the Yosis. So, your inverter all these are generated from the Yosis. So, now these are the XOR gate and MOCs all these are standard cells actually and you have all sorts of gate which is implementing your design actual design what you are going to design in your chip. So, now this in the flood planning is done then I will go to the track file actually.

So, if I go here okay So, I am sourcing the track file. So, here basically if you can go and see this track file okay So, this track file is basically if you can see here you have metal 1, metal 2, till metal 10. So, you have 10 routing metals are there in this technology of 45 nanometer NAND gate process. So, if you can see here you have metal 1 track with offset of this much, pitch is basically middle of the metal to the middle of the metal then similarly the offset in the X direction, there is a offset in the Y direction, the pitch in the Y direction all these are given in this file metal 1, metal 2, metal 3, metal 4 all these are basically defined minimum value is defined in the technology you can little bit play with the offset. But you cannot change the pitch actually if you change the pitch then there will be a problem with the routing congestion okay. So, we have to follow the technology files to find the metal 1 to metal 1 pitch and offset you can be changed based on our requirement okay. So, now what we have to do is that we are going to that previous place where we have left out basically we have done the sourcing of the tracks it is done. We have done the sourcing of the tracks.

So, these tracks actually if you can see here for example if I can go So, what we can do is that we have these tracks are there. So, if you can go here there is a track is written here. If you click here all the metal tracks are visible. If I do not click it is not visible. So, if you can see the tracks of each of the metal okay I will try basically let us say only to show the metal 1 metal 10 track. I do not want to show the other metal let us say okay. It is just showing this purple color thing is the metal 10 track. If I remove this that is gone okay. So, you can see the tracks by just clicking on this thing. Let us say I want to see the metal 9 track. So, I will off the metal 10 track and make the metal 9 track on. It is visible okay. So, now I can make all the things on by clicking here okay. So, all the metals are on now. So, now we will do this track part is determined by this whether you want to see this or not is defined here okay.

So, you can make it on or off okay. So, basically each of the metal has a preferred direction of routing and non-preferred direction of the routing. Let us say metal 1 is horizontal okay that is a preferred way of routing and the metal 2 will be vertical because metal 1 is used in the power grid usually the metal 1 is horizontal and the metal 2 will be opposite to the horizontal. So, metal 2 will be vertical okay. Metal 1 is horizontal, metal 2 will be vertical, metal 3 again will be horizontal, metal 4 will be vertical like that. But in some cases if the router could not route the signals they can use the non-preferred manner provided there is no shorts in the design or there is no DRC violation okay. Now I have to go into the next part. So, if there is any kind of buffer inserted during the synthesis So, those can be removed using this command. If you do not want you can comment this command alSo, but you can play with this command to remove any extra buffer there in your design which can be corrected during the physical synthesis. So, after running this command 26 buffer has been removed okay 26 buffers are removed from the design. Then we have basically there is two types of IO placements is there in this tool. One is called a random IO placement and the second latter part will do the actual IO placement. So, the first one is basically random IO placement.

So, where we are placing the defining the layer in which your pins will be placed okay basically input output pins will be placed okay. So, here we are running this okay So, the random this is a rate in random that is why it is called random placement and horizontal layers then the vertical layers if you can go here horizontal layers and the vertical layers okay. So, those layers need to be mentioned which metal layer your IO should be connected to the internal pins okay. So, now if you can see here these if I can show you in this area these green color dots are your pins actually pin locations. These green color dots are the pin locations actually okay. I am removing the tracks to show this is now visible this green color points are the pin location for the pins whatever we place just now. After doing this we have a macro placement we can have some of the things macro which need to be placed okay.

So, if you can go here we have macro placement okay. So, this macro placement is done in this step okay. So, we can run this script macro means let us say if you have any kind of memory block or any kind of IP we are taken it from a third party company and you are using in your design those are the macro and those have the fixed location. So, we have to fix their location before doing the other standard cell routings actually okay. So, this basically was done in this step then we have next step. So, now we will do that tap cell insertion. Why these tap cells are needed? Tap cells are basically the body contacts which connects the body. If you do not add these tap cells let us say NMOS body and the PMOS body for NMOS it should be connected to ground and for the PMOS body should be connected to VDD. If you do not place those cells then there will be a concern of latch up okay. So, this tap cell is essential in your design. If you do not place that the tool will

not tell you that there is an error in your design until we specifically write a report on the tap cell checks.

However this will not the chip will never work. There are two types of design based on the standard cells what is there. In one type of standard cell the body contacts is placed in the VDD and ground lines. So, in those type of cells standard cells you do not need to insert the tap cells. But in this case that tap is not placed in the supply line So, we need to add those tap cells. So, this is the tap cell placement procedure. So, basically we have basically 128. So, what we are showing here if you can see we remove the basically rows to show how the tap cells are inserted in the design okay. If you can see here this end caps are inserted 128 end caps are inserted okay in this design because the area of the design is small this end caps are alSo, like a tap cells only to give a body contact. But if your design size is wider then this tap cell will be inserted in the actual design okay. Now if you can see this physical only cell if I remove this one then their tap cell will go away. So, this basically what we discuss in the input files your tap cell is a physical cell actually. So, this tap cell is basically inserted by this procedure okay. Now after you do the tap cell placement then there is a procedure called power distribution network insertion. How you can do the power grid design okay.

So, this power grid design is one of the crucial factor when you are doing a chip design because if your power grid is good then it will provide uniform supply to all the location inside the chip. However if your power grid is not good then there will be lots of higher drop and each of the part will see a different supply voltage. So, that is why this part is very very crucial while going for a chip design. So, if you can see here this one this is the config file and the power grid but you can see here.

So, if you can see this the changes in the layers in the diagram if you can go okay I can show the power file. You can see this this is the actually file for adding the power grid. Here if you can see we have only two supply voltage VDD and VSS. So, we have connection for VDD net and the ground net and if you can see this is the most important part is that what are the metals it will be used for the power grid. If you can see here your metal 1 what is the width of the metal, what is the pitch of the metal, what is the offset those things are needed for this power grid actually. And if you can see this is the one of the power grid, this is for metal 4, this is metal 7, metal 1 and metal 4 is in a group it will be inserted in the design. So, these are the commands used to create the power grid okay. You need to look into this add pdn underscore connect to create the power grid in your design okay. So, in this method these commands are very important to create the power grid. So, you need to go through it and to create the proper power grid for your design okay. So, this is one time process okay and it was done before the basically any kind of routing because the power has the most dedicated network okay. So, this dedicated network should be placed with highest priority.

So, that is the thing the power distribution or the supply line should be placed first before doing any kind of placement. Then the tool will know that So, and So, metals are there in certain location based on that then they will place the standard cells then they will do the global and detail routing because the power has a most important part So, that is done first okay. So, we discussed about open source tool open road in this lecture. Hope you enjoyed this lecture. Thank you for your attention.