VLSI Physical Design with Timing Analysis

Dr. Bishnu Prasad Das

Department of Electronics and Communication Engineering

Indian Institute of Technology, Roorkee

Week 01

Lecture 02

Introduction to VLSI Physical Design

Welcome to the course on VLSI Physical Design with Timing Analysis. In this lecture, we will discuss about VLSI design flow and also we will discuss about the various steps of the VLSI physical design. So, first of all we will discuss what are the steps of a VLSI design flow, then we will mainly focus on the physical design flow and there are several verification flow are there and then basically we are talking about the physical verification. For example, DRC, LVS, ERC and ARC all these will be discussed in detail in this lecture. So, the VLSI design flow first we starts with a system specification, how we can define the system specification basically the functionality, the performance area and input output.

For example, let us say we want to design a processor. So, the processor can be targeted for various applications. Let us say it will be used for a mobile application or a smartphone application or it will be used for a basically general purpose system. So, in that case let us say if you look for a processor design for a smartphone, the power is one of the critical factor, but if you design a processor for a regular PC or a personal computer in that case your power is not a constraint , performance is a constraint or you need a basically processor for a server, then in those cases power is not important, but the performance is most important parameter, and also if you look into the functionality what kind of what is the performance, what is the area all these things need to be specified here.

So, the functionality, performance target, physical dimension, what is the size of the chip is also important because in some cases the size of the chip will not fit to the area of your device. So, in that case we need to think of different types of packaging to fit that one. So, the physical dimension of the chip need to be provided in the system specification and which technology node we are implementing that one that is also need to be specified. So, it involves this one is one of the most difficult task because it requires various persons inputs. For example, chip architect, circuit designer, product marketer, operation managers, layout, and library designers all these will come together to whether the specification whatever you are targeting we can meet it in the stipulated time to market or not. So, based on that we need to define the specification.

So, now we will go into the architecture level design. So, in what type of architecture we should use. Let us say if you are designing a processor whether we will use a RISC architecture or a CISC architecture or MIPS this is a MIPS processor basically. And so, here you have a data path controller and ALU. So, here what we are doing here is that how they are interconnected with each other, what kind of communication protocol, bus protocol is used, AXI bus is used or what kind of communication protocol is used that need to be defined at this stage.

So, the main thing here is that integration of analog and digital blocks. In your processor you need also a PLL to be integrated because it will give you clock to the processor, high speed clock to the processor. So, the integration of the analog and digital blocks, memory management is crucial thing how the memory will be managed inside the processor, addressing scheme, serial and parallel communication between the memory management unit. Then we have different multiple cores are integrated in the single chip right now.

So, you may need a additional DSP core for performing the DSP applications. So, we also need to look into different communication protocol like AXI bus protocol all these that should be defined in case of architectural level design. Now look into the RTL, the Verilog need to be designed that functionally how can you implement that design. So, the functional design we need to functionally implement using some form of a register transfer level. This is the Verilog code which will be written in the high-level language where we can implementing the system specification, then the architectural design into more detail which is basically represented in terms of hardware description language.

So, here the functionality need to be satisfied, connectivity of the module needs to be defined and it will be represented in the high level, it will be represented using hardware description language. We need to whenever we do the RTL, we need to basically simulate it multiple times whether it is working at each and every point of requirement. So, we need some kind of simulator to verify that. So, for the students, we need to use some kind of open-source tools like Xilinx Vivado, which is a good simulator for RTL. You can write a code in RTL, you can simulate using Xilinx Vivado.

So, that is to simulate and verify the module thoroughly. We need to thoroughly verify the design whether it is meeting your architecture level description or not. So, then we use a logic synthesis tool to convert your HDL to gate level netlist. So, we need a logic synthesis tool which will convert your HDL to gate level netlist. So, this transformation from this to this is called logic synthesis.

So, now we have a gate level netlist. The gate level netlist is specific to a technology node. So, let us say I have a 65 nanometer technology node, the gate level netlist will contain the transistor of 65 nanometer technology node, logic gates which is implemented in a 65 nanometer technology node. So that is technology. Now this circuit design is basically technology specific, but your RTL is independent of technology.

RTL there is no information about which technology it is implemented. So, here we are implementing everything in the transistor level. We have memory, analog circuits and high speed designs. If you have analog and digital together, analog blocks will be implemented separately and it will be added as a black box to your RTL and which can be implemented in the physical design flow, but those digital flow we will use the standard cell library for implementation.

So, now you have a design which consists of gate level netlist, transistors specific to a particular technology node, then we need to simulate that one to verify whether it is really working what I intended, whether it is meeting my speed or not. So, let us say if you do a spice level simulation, if your design is smaller, you can do a spice level simulation that is using a LT spice you can do. Or if you are using logic simulator where you can do logic level simulation because you have a gate level netlist, you can do also logic level simulator to use the, to verify the functionality is working or not properly. Then you need to have, consider the transistor size, power and manufacturing variability, all these whenever you have a gate level netlist. Then you have the physical design, the physical design this transformation is called from circuit level design to this one, from circuit level to the physical design this transformation is called physical synthesis.

So, what it does? You have a gate level netlist, from the gate level netlist we are creating the layout of your design using some physical synthesis tool. So, here we are going through some of the steps like placement, routing all these two and also at the same time we need to satisfy the fabrication constraint provided by the fab and then we need to also meet the performance area, reliability, power and yield whenever you are going for a physical design or the lay when we generate the layout of the design. Then finally, we need to do the physical verification and sign up. The physical verification include design rule check, layout versus schematic, electrical rule check, antenna rule check all these things we need to do. Then we need to also basically do the timing sign up to check whether our timings are met or not. So, you have DRC which will satisfy the technology constraint, you have LVS to verify our functionality is working or not, our layout is matching to the schematic or not functionally. Then we extract the parasitics and simulate that whether the extracted netlist is functionally meeting my speed or not. Then we need to check the antenna effects to proper power and ground connections. All these things need to be checked in the stage of physical verification and sign up. After that we need to send the design to the fab and that GDS2 which is a file which contains all the layout information that is sent to the foundry for fabrication.

Then the foundry will print your design into silicon to make your chip and send you back the chip. So, in case of a fabrication process or the DRC clean layout is sent to the foundry for manufacturing that is called a tape out. So, we have different sizes of wafers typically 200 mm or 300 mm wafer size is used. So, this we have different photo mask to print your design into the silicon wafer. Finally, we have chip in your hand and this chip if you see internally this is your die and this is your package. So, this is called a dual inline package. So, here it is 40 pin. So, this package why it is essential? It is used to test our design. We can design a PCB and test that one the package chip we can test it actually.

So, there are several types of packages like dual inline package, pin grid array, ball grid array, all these packages are available based on your requirement and cost and performance requirement all these things we need to choose one of the package. More the number of pin cost will be higher. So, depending upon the number of pins and performance all these we need to choose the package. So, we need also package we have wire bonding, solder bumps and we basically test then the functionality timing power after the chip will back to us. Then we will get the chip back to us for testing. Then we will finally get the chip.

Now we will go into physical design cycle. The physical design cycle is basically whatever we are talking about VLSI physical synthesis. This is basically VLSI physical synthesis. So, what we are doing here we have a gate level netlist. We have a logic gate level netlist which is coming out of logic synthesis. We are using that logic gate level netlist as input to VLSI physical design tool to create the layout of your design. This is the final layout. So, you have logic circuits as input layouts as the output which is done automatically with few of the steps we will discuss in this in the next slides.

One is partitioning. First one is the you have a bigger design will partition into smaller part that is called a partitioning. Then we have chip planning, then placement, clock tree synthesis, signal routing and timing closure. We will discuss one at a time. Let us basically the partitioning means what we have the complete design here. We have partition into smaller parts and such that the number of interconnect from one part to the other is minimum. So, here in case of floor planning we need to define the steps of each of the blocks sub blocks such that we have less means we are the module steps and arrangements are fixed during the floor planning. Then we need to space yourselves at different locations. That is called placement. This is we need to find out the proper location of each of the cells inside the chip using the placement phase. How we can place it such that my interconnect distance will be minimum, my performance will be improved, my speed will be improved. Then we have clock tree synthesis. The clock is most important signal inside the chip. So, how we can design the clock tree inside the chip such that our routing is basically optimized. Main purpose of the clock skew main purpose main aim of clock tree synthesis is to reduce the skew between two leaf nodes. Your clock is coming from onto the chip from this side it is going to different units. Ideally it reach to all the units at the same time, but because of different interconnect distance from the source to the sink they will reach at different times.

However, there are some advanced clock tree synthesis techniques are there if you use those techniques the all the points the clock will reach at the same time. So, in case of signal routing we have basically with the power routing the power is VDD ground routing and also you have global routing. The global routing is defines the routing tracks in the channel and switch boxes actually. The resources in the routing tracks in the channels and switch boxes that will also is defined it is not give you the actual routing it will collectively give you one location how the routings will go. But the next step is the detail routing will specify the actual specific metal layers that will go to the tracks.

So, the routing is two parts one is the detail which is assigned the routing to the metal lines and the global which is basically specific to a region. So, the global routing is the first step, detail routing is the second step. Then you have timing closure which optimizes the circuit performance through placement and routing and during the design phase we define some timing and we need to meet that timing through optimizing the placement routing and all these optimizations. So, the timing closure is one of the most critical part of your VLSI physical design flow.

Then we have physical verification, how we can verify our design. So, one is called the design rule check, layout versus schematic, electrical rule check, antenna rule check. Let us discuss about the design rules. Basically, whenever we have specified a technology node, the foundry or the fabrication unit will supply you one technology file to the designer and we need to that rule file, the technology rule file is the golden model for the designer. We need to follow those rules in order while doing the layouts. So, the geometrical constraints on the layout to ensure successful fabrication and must be avoid any kind of manufacturing defects. It will also tell you all the minimum design rules, we need to optimize your design. So, let us take what is inputs to your DRC tool. One is the layout GDS and your DRC files coming from foundry, it is coming from the foundry. Obviously, both are technology specific. Then you have layout, it is coming from the designers, layout designer, both is going to the DRC tool, then it will generate some DRC report. Whether my layout design by the designer matches the rules defined by the foundry or not. DRC rule provided by the foundry is the golden rules. We need to follow while doing the layouts. If it is not matched, then it will generate some error in the DRC report. We will discuss some of the violations in this report in this lecture.

For example, let us say I have a minimum width defined in a technology DRC tool. Let us say this one is satisfied, however, this is not satisfied, this is not correct. So, then it will create a DRC violation in this area. Then you need to increase the metal to satisfy that violation. Then you have minimum spacing actually, there is a metal-to-metal minimum spacing need to be there. If the metal to metal minimum spacing is not maintained, so, then basically it will show a violation. So, how can we solve this? We need to basically move the one of the part to the other such that the minimum rules what is specified in the DRC rule provided by the foundry will be satisfied. So, here what is the rule is a minimum enclosure or overlap. This is basically a poly; this is a metal one. So, this metal one is not covering the contact, this is a contact. If your poly and poly is connected to metal one through a contact and the metal one is not covering the contact completely. So, it will show as an error here. Basically, the metal one enclosure is not sufficient, then we need to solve it by moving the metal and completely covering the contact.

Then you have a layout versus schematic. So, here we have two netlists. One is coming from the schematic, one is coming from the layout. So, what is the thing here is that we have two netlists, one is coming from the layout GDS, one is coming from the schematic which is going to the LVS tool. This is a EDA tool, electronic design automation tool. So, then you have a LVS rule file, it is also provided by the foundry to you, and we need to check that whether the layout GDS and layout netlist is matching with the schematic netlist or not. If it is not, then it will show LVS errors. So, there are several LVS errors. If your device is not matched, number of transistors is not same in both the designs, then it will show a device mismatch error. Number of nets is not matching, then it will show the net mismatch error. Then if your size of the transistor is not same, then it will show property error. Then port swap error, basically if two of the ports are swapped, then it will show port swap error. So, let us say here if you have some extraction error, for example, missing device terminal in case let us say you have a transistor, this is a NMOS transistor, this should be grounded and this is a gate, this is a source, this is drain, then we have a body, if the body connection is not connected to ground here, then it will show a missing device terminal error. So, it should be grounded. If it is showing that, then you need to connect that actually. So, then if you have a extra device terminal, if you connected those that will also show as error. Then you have device extraction error, take short, if two of the net is sorted or open those things will also show as error. So, for example, here let us say this is a layout. I have a schematic for this one, this is a inverter layout. I have the basically this is a PMOS, this is in, this is out, this is VDD, this is ground. Then this body is connected to ground, this body is connected to PMOS body is connected to VDD.

Here if you can see body connection is not given here, forget about this body connections are present. But if you can see here, whenever you are thinking of layout versus schematic or few things I need to point it out here, your schematic is a golden reference. Means that you are doing the schematic carefully, there is no mistake in the schematic, then and you have simulated it. Then only you can compare in case of a in that to layout. And here this is a layout of inverter. What is happening here? We extract a netlist from the schematic, we will extract a netlist from the schematic. So, that is a schematic netlist. Then I have a layout netlist. So, if both are same, then only it will pass the LVS, otherwise it will show error, ok. In this case body terminal, I am removing because here it is not there, ok. So, here this golden model, sorry. So, this is a golden model. If you do any mistake in the schematic that will translate to the layout. So, you just should be more careful should be given while designing the schematic.

Then you have ERC which verifies the correctness of your supply lines, the power and ground connections, your signal transition times, capacitive loads, fan outs, all these need to be verified during the design process. So, in case of antenna rule check, we have a long metal which is connected to the gate of your transistor. We have a long metal connected to the gate of the transistor. What it does is that it accumulates lot of charges in the gate which damages the gate oxide, ok. And this gate oxide if it is damaged then there is no conduction, this is called plasma induced gate oxide damage, ok. So, this effect we need to check that during the VLSI physical design also you need to correct that. So, what is the consequence of that one? Your gate leakage increases, changes the threshold voltage and also it reduces the reliability of your transistor because your threshold voltage changes, leakage current increases. So, the reliability lifetime of your transistor also reduces. So, we need to fix those one during the design time. So, let us say you have a long metal connected to the gate of your transistor, we need to make it smaller actually, ok. We need to make it smaller. So, basically the M1 is more likely to have damaged the gate oxide than M2. So, in case of antenna rule check here, we have a long interconnect here which is going to the gate of your transistor. So, it will accumulate more charges and it will damage the gate oxide which will have more leakage current which is degrade the reliability of your transistor. So, this can be avoided by adding a jumper, metal to jumper. So, here we need to add a jumper. So, the metal 1 is changed to metal 2 near to the gate actually. So, since the near to the gate it is changed, the distance of the metal is reduced. The distance of the metal is reduced means the number of charge accumulation will also reduce and there will be no damage of the gate oxide and your volition can be fixed. And the next method is that you can connect a diode here to discharge the charge. So, that this is called an antenna diode which can also be inserted automatically using the VLSI physical design flow. So, you should have a proper antenna rule file to add in the VLSI physical design to fix the antenna error.

So, in this lecture we will discuss about the VLSI design flow and different stages of VLSI physical design for example, partitioning, floor planning, placement and clock tree synthesis and finally, the signal routing and the final stage is basically physical verification and sign up and we discuss about different physical verification steps in this lecture. Thank you very much.