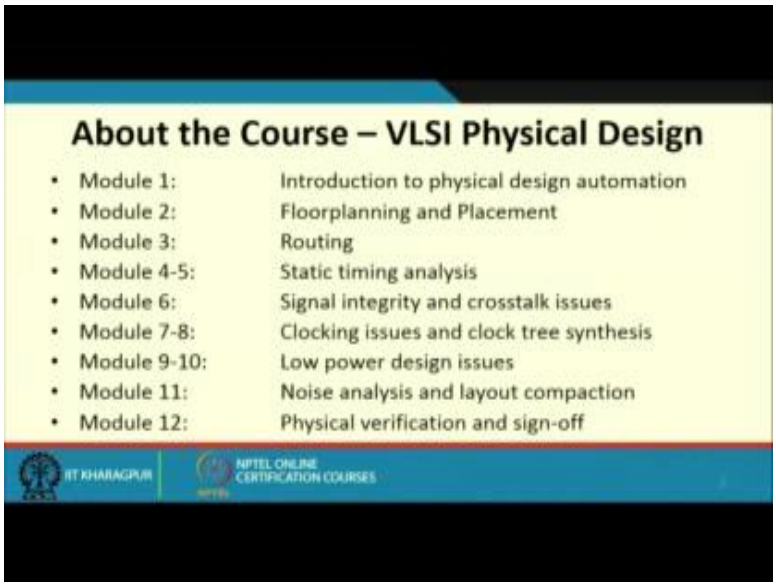
VLSI Physical Design
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 01
Introduction

So, let me welcome you to this MOOC course on VLSI physical design. In the first module that we shall be starting now, we shall be discussing some of the basic concepts on VLSI physical design automation. So, this first lecture you shall be looking at some of the introductory topics. I would first like to talk about the rough coverage of this course.

(Refer Slide Time: 00:50)



So, as you can see in the first module we shall be talking about some of the introductory concepts. So, we shall try to motivate you regarding the need and the importance of physical design automation in the VLSI design cycle.

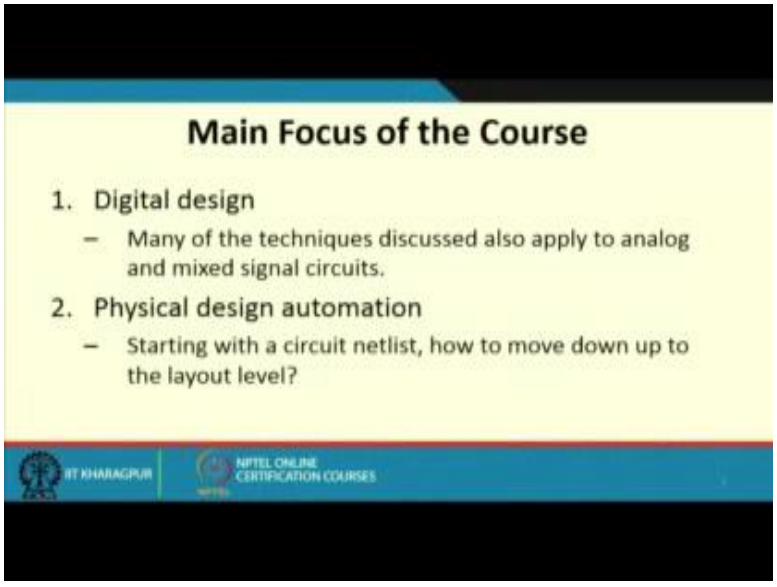
Then, in the successive modules we shall be looking at some of the very important components or steps in this physical design automations step. For instance, in the second module we shall be talking about floor planning and placement. And, as we shall see partitioning of a design is also equally important at this step. Module 3, would discuss the various aspects of routing. So, here we shall see that there are many

different kinds and types of routing algorithms and routing problems involved in the design cycle.

Then, in the next two modules we shall be looking at a very important topic of modern day VLSI design namely static timing analysis. So, here we shall be talking about the various timing issues and the different ways in which you can enhance the performance of a design of the circuit. Module six will talk about the signal integrity and cross talk, which are related. Then, we shall be discussing the various clocking issues. And, in particular how the clock nets are routed, which are sometimes typically called clock tree. Then, in the next two modules we shall be talking about the very important issue of reduction of power in circuits. Then, in the last two modules we shall be considering noise analysis layout compaction then, physical verification with final sign off before the chip is designed.

So, essentially what we are talking about in this course is that starting from the behavioral specification, we can arrive at the circuit net list in some form. And, from the net list, what are the different steps that we need to follow in order to finally arrive at the physical layout at the lowest level. So, during this course we shall basically be talking about the various aspects and the tools and technologies that reused in the process.

(Refer Slide Time: 03:48)

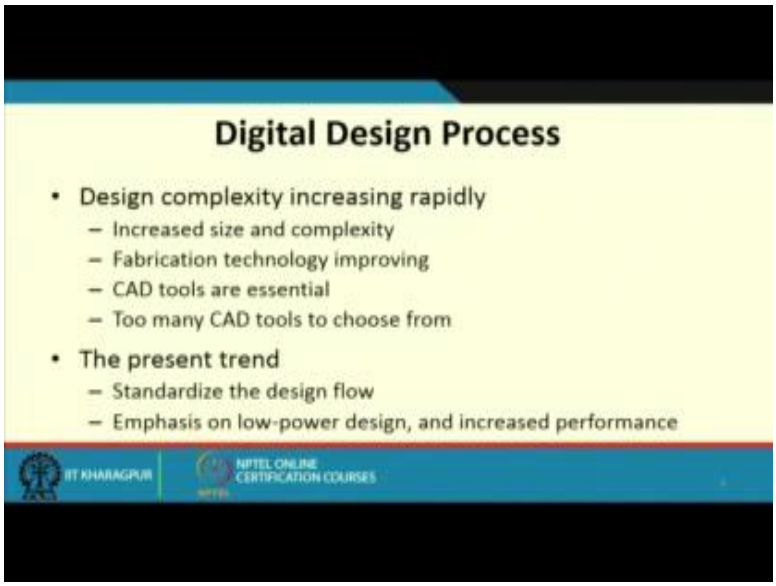


Now, talking about the main focus of the course; well, as you can see we would be talking about digital design and we would be talking about physical design automation. So, digital design means as you know that we are only considering circuits and sub systems, which work on digital signals in this, you can say topic, digital design.

But, as you know in a modern day systematic chip, let us take our mobile phone, for example. So, in this mobile phone we have some VLSI chip, which contains not only the digital sub systems or circuit components, but also other analog and communication circuitry embedded in a single silicon package. So, we call them mix signal digital and analog combined kind of ICs.

So, whatever discussions and concepts we would be talking about, they would apply not only to digital circuits, but also to this kind of mix signal and analog circuits as well. And, physical design automation as I had mentioned, it basically consists of the different intermediate steps that need to be followed to translate a given circuit net list into the final layout, which can be given for final fabrication of the chip. This is the scope of physical design automation and this is the scope of the discussions that we are expected to cover as part of this course.

(Refer Slide Time: 05:50)



So, let us briefly talk about the digital design process. So, as you all must be knowing that the complexity of circuits have been increasing very rapidly over the years, primarily because of improvements in semiconductor and fabrication technologies.

So, in the beginning in the 70s and 80s we used to talk about 5 micron technology and things like that. But, today we are into deep sub-micron technology. So, here very recently devices with 14 nanometer feature sizes have already come to the market. So, we are seeing a fantastic growth in the semi conducted technology and the way we are able to pack more and more devices in a single circuit or a chip.

So, in the digital design process we are not only looking at the increased sizes of the circuits, but also the associated complexity involved. What does this mean? You see, handling a circuit with 100 gates and handling a circuit with 1 million or 1 billion gates is of course not the same. We need much more effort, much more sophisticated tools and technologies and techniques to handle this circuits of extremely large sizes. Say, earlier the circuits were small. Some of the steps we could have done manually. But, in today's world manual design and manual, you can say analysis is almost ruled out. This circuits are too large to be. Even you can say, I mean, it is by difficult to even think about doing this steps in a manual fashion because it will take years to do even a very simple small part of it.

So as I said, the fabrication technologies have improved. And, the computer aided design or automation tools have become extremely essential in the present day context because of the increasing size and complexity of the circuits. But, the issue is that if you look in to the available CAD tools that there in the market, you will find that there are many such available tools. So, one question that naturally would come into mind, "Which tool should I choose for my design?" So, of course the kind of tool that you will be choosing in your design may depend on number of factors, cost can be one of the most important issues. Some of the tools, though may not be the best possible among the alternatives, but the cost may be affordable for you to go for that. Some people, they also do some kind of a mix and match.

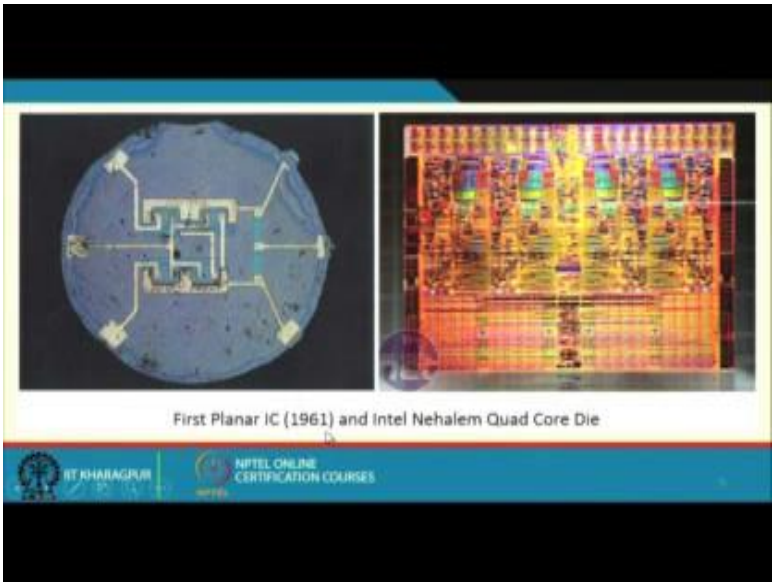
Let me give an example. I mean, there is a research group in our college who actually carry out the VLSI digital design process, design flow, it is called. As a combination of two parts, the high level part that is called the front-end design, that is carried out

using the CAD tools available from synopsys, which is quite well known. And, the later part which is called the back-end design or the physical design that is carried out using the cadence CAD tools.

So, as you can see depending on the convenience and the way, I mean, you are able to use the tools, you can have a mix and match. So, you can take the best of both worlds for example in this case. So, the present trend as I had mentioned is to standardize the design flow. Design flow means what are the tools I would be using. Like I had said, in our case we would be using synopsys in the first few steps and cadence in the later steps.

And, of course with the increasing emphasis on battery operated devices and various mobile computing platforms, the greater and greater emphasis have been laid on low power design without sacrificing performance fine.

(Refer Slide Time: 10:41)



So, let us look at these two pictures. So, so much, you will get a very rough idea. So, the picture in the left shows one of the first ICs that are manufactured in the year 1961. So, this IC comprised of only a few transistors. But, in contrast you look at this photograph on the right side. This corresponds to the Intel Nehalem Quad Core processor, which houses more than 1 billion transistors. So, you can see the, see of change that has taken place over the years.

(Refer Slide Time: 11:23)

And, here we have another very important; you can say empirical rule or law that has driven us over decades. Gordon Moore in the 1970s, even earlier than that, he predicted the way the circuit; you can say the circuit design and fabrication technology would evolve over the years.

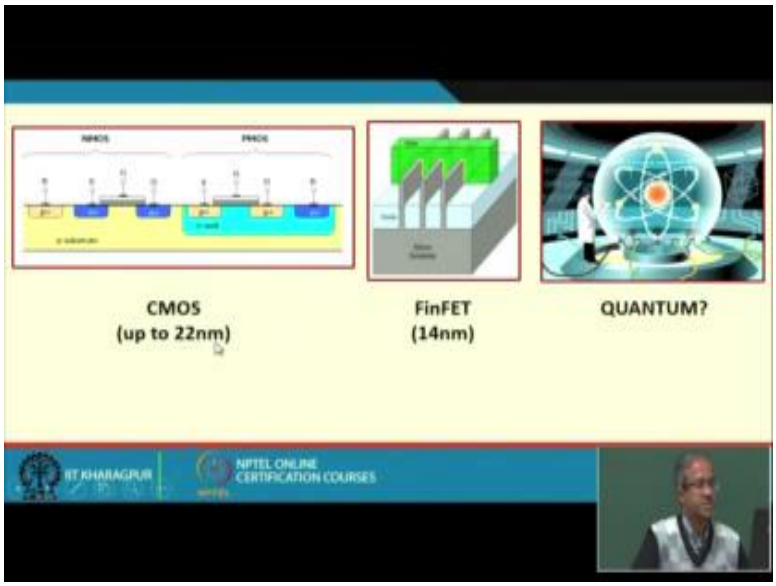
So, what you predicted was that the number of transistors of the basic components that you can pack inside a single integrated circuit would be double in every eighteen months or so. So, this means some kind of an exponential growth over the years. So, initially people thought that, well it was fine; at least for a few years we would be able to sustain this exponential growth. But, finally this has to taper off. You cannot continue this for long. But surprisingly with the developments not only in semiconductor technology, but also with enhancements and improvements in the architecture of the systems; the way we are building the chips multi core architectures that we see today.

So, we have been able to sustain this exponential growth over the years till today. So, this slide if you see, so on the y axis you see the number of transistors in an exponential scale. Around the x axis you see the years starting from 1970 up to 2015, which is shown here. So, you can see in this graph that here from the first microprocessor that came to the market, it is here 4004 up to the latest (Refer Time: 13:20) IV microprocessor multi code systems, which Intel is coming up with. So,

mostly you predict that this curve will be a straight line. So, as you can see this straight line behavior is still being maintained.

So, because of this exponential increase in growth, one thing you can easily see that design complexity is also increasing exponentially; which means automated computer aided design tools are essential. It is not an optional. I mean, this is not optional thing anymore. So, you do not have any choice. You will have to use such tools and of course you will have to follow some well-defined design flow in order to control and manage the complexity of the designs. Now, I had said that over the years the circuit complexity in terms of the feature sizes have reduced in sizes.

(Refer Slide Time: 14:27)



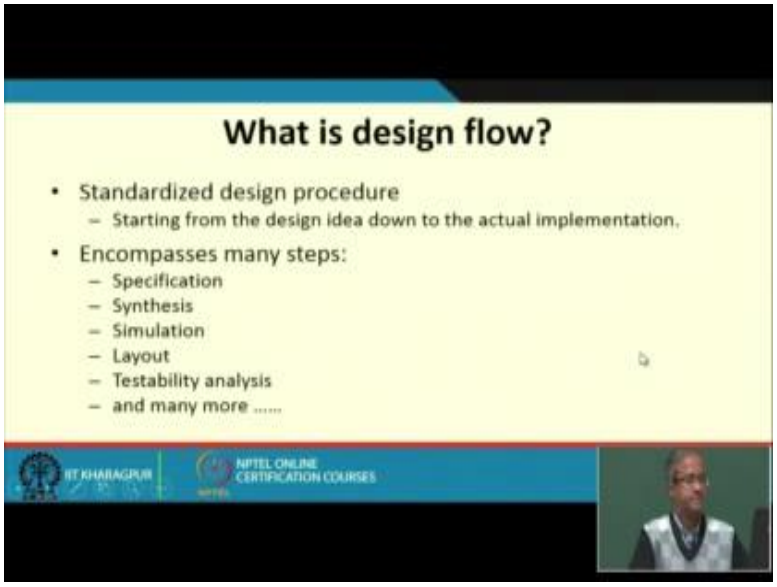
So, I will show slide quickly very; you can see the pictures. So, on the left I have shown classical CMOS transistor consists of an NMOS CMOS gate. In fact, consist of an NMOS transistor and a PMOS transistor.

So, they are laid out on the same silicon substrate. So, with this classical CMOS kind of design, we have, here we have been able to sustain the growth up to as small as 22 nanometer technology. But, today we have newer ways of fabricating this MOS transistor. So, one of the most popular ways is called FinFET, where the transistors look like fins. Now, they are not laid out in a flat fashion, but they are laid out as fins which require much smaller area to fabricate. So, FinFET can go down to much smaller feature sizes. And, as you can see FinFET has designs have been reported,

which has gone down up to 14 nanometer. And very recently, some communication chips from Qualcomm have been reported. They have used FinFET 14 nanometer technology.

Looking into the feature, we really do not know can quantum computing be an option for us. So, this only time (Refer Time: 16:00). But, of course many researchers are walking in this direction. So, let us hope that we will see something very interesting in the near future.

(Refer Slide Time: 16:11)

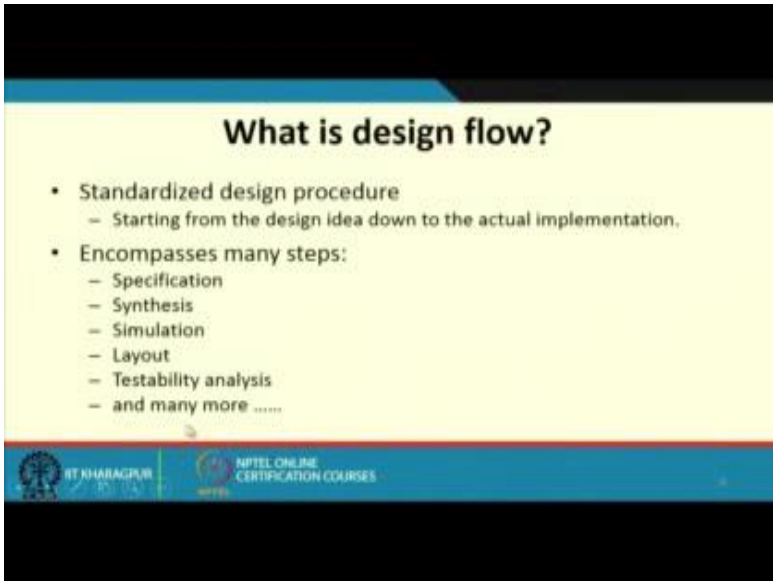


Let us now talk about the design flow. Now, as I said design flow is essentially some standardized design procedure, where you will be using some set of defined steps that are to be followed in order to translate given design into the final fabricated chip that you are targeting for. Now, there are many steps in this design flow. It is not just a single step which is automated. There are a large number of intermediate steps. Some of which are mentioned here. You typically start with this specification of a system what your system is supposed to do.

In the next step, you go for synthesis. Synthesis means you are trying to translate your, this specification, into some kind of circuit net list. Now, again this synthesis may not be in a single step, there can be a multiple levels of synthesis as you will see little later. Then, in order to verify whether the synthesis process is correct, we usually use simulation to verify that whether the circuit net list that we have obtained behaves

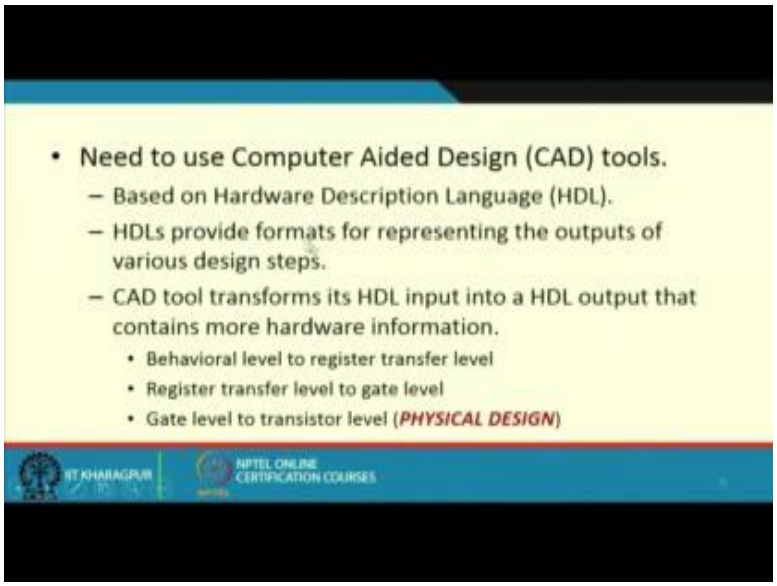
in the same way as for the original specification. Then, finally we go down to the layout, which is now quite ready for fabrication. But, of course in the process we also carry out testability analysis because whatever we design, there can be manufacturing defects. And after fabrication, each and every device has to be subjected to elaborate testing before it can go to the market.

(Refer Slide Time: 18:16)



So, and there are many other step. This is a very simplified design flow. Few of this steps that are shown and, as I said that we must use computer aided design tools and, let us see how this tools typically work. Now, one thing the inputs and the outputs of this tools are now fairly standardized, so that you can say interoperate multiple number of tools because they work on the same input and output formats. So, we typically give our specification in terms of a description language, which you call as hardware description language or HDL.

(Refer Slide Time: 19:11)



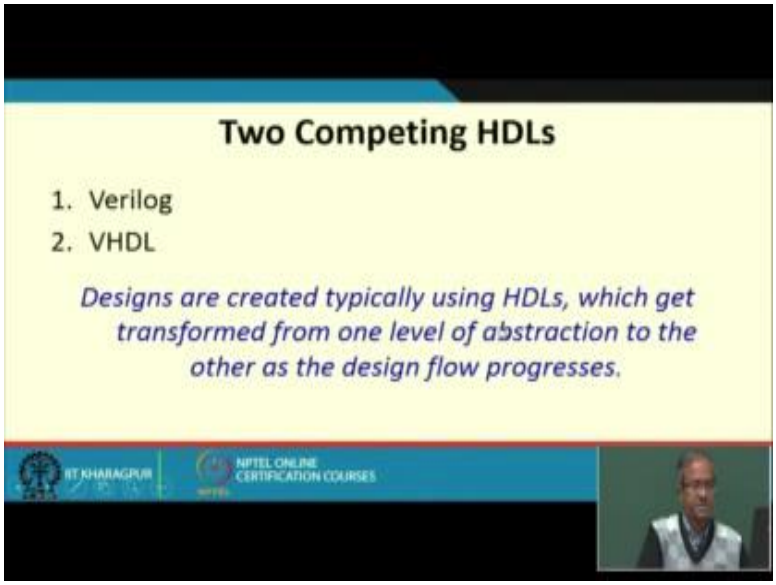
So, this computer aided design tools are based on hardware description languages. These HDLs provide ways to represent both the inputs as well as the outputs of the various steps in the design flow.

So, essentially a CAD tool will transform its input, which is in a HDL format into an output, which is also in a HDL format which will contain more detailed hardware information. Like some examples are here. You can translate from a behavioral level specification to register level specification. So, what is register level specification? Register level specification means a kind of circuit net list which consist of components or modules like say busses, registers, adders, multipliers, multiplexers. Some bigger functional modules of blocks, this is called register transfer level specification. Now, again in the next step you can translate this RTL or register transfer levels specification to gate level specification, where you translate everything into basic logic gates and flip flops. And, in the last step this gate level specification might get translated to the final transistor level specification, which is sometimes called physical design.

Now, talking of the hardware description languages, usually most of the people use one of two languages Verilog and VHDL. Of course, the choice is up to the designer or the person uses it. Both of these languages are equally capable. They can be used to

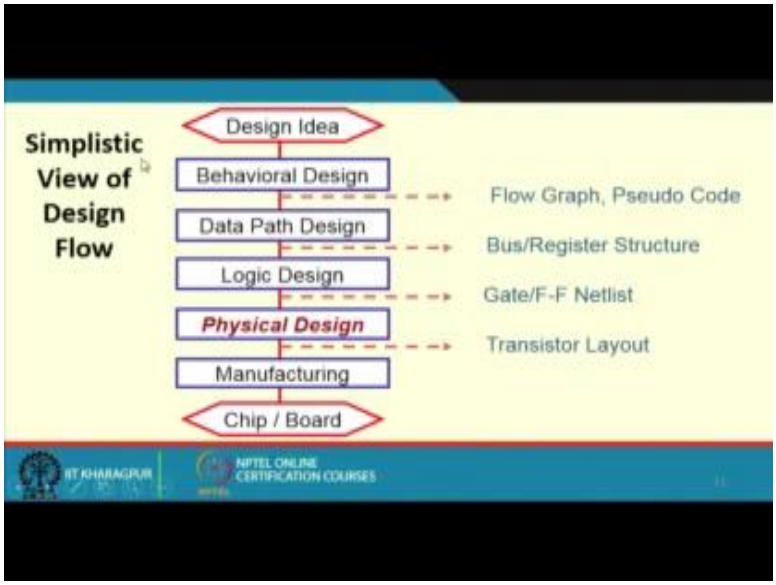
model any designs, any circuit. And, also can be used in the synthesis flow, in the design flow along with the CAD tools.

(Refer Slide Time: 21:28)



So, as I had said designs are created typically using these languages, which can be transformed from some higher level of abstraction to a lower level of abstraction as time progresses.

(Refer Slide Time: 21:50)

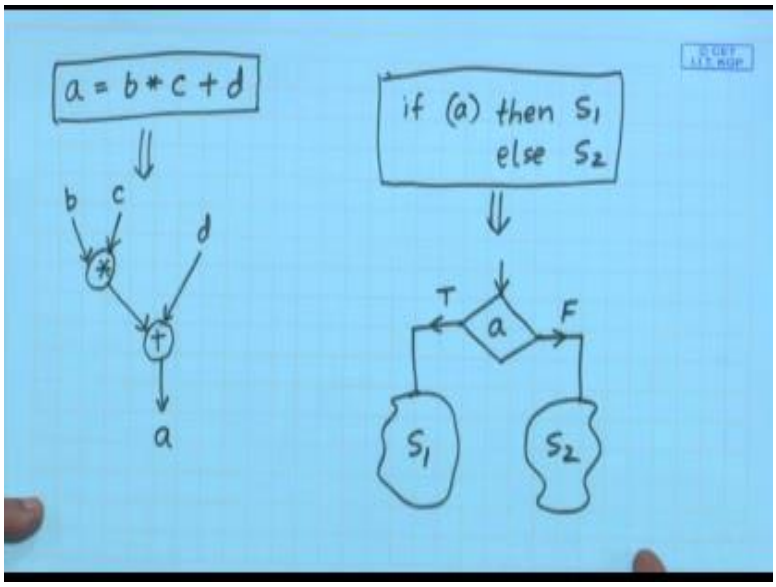


So, this diagram gives you a simplistic view of design flow. Simplistic means I am not showing all the steps involved. So, what I am showing? I am showing some steps

which starting from a design idea. You first carry out behavioral design where you model your idea in terms of flow graphs and pseudo codes. I shall give some examples. Then you go for data path design, where do you translate your design into a register transfer level architecture; bus, registers, adders, etcetera. Then, you go for logic design where you would translate them into gates or flip flops. Then, you go into something called physical design where you translate these into the final transistor level layout, which is now ready for fabrication or manufacturing.

So, once you have fabricated, you can finally get your chips using which you can create your circuit boards. Now, let us briefly look at some of this steps to just understand how this works. (Refer Time: 23:04) say behavioral design, what does it do?

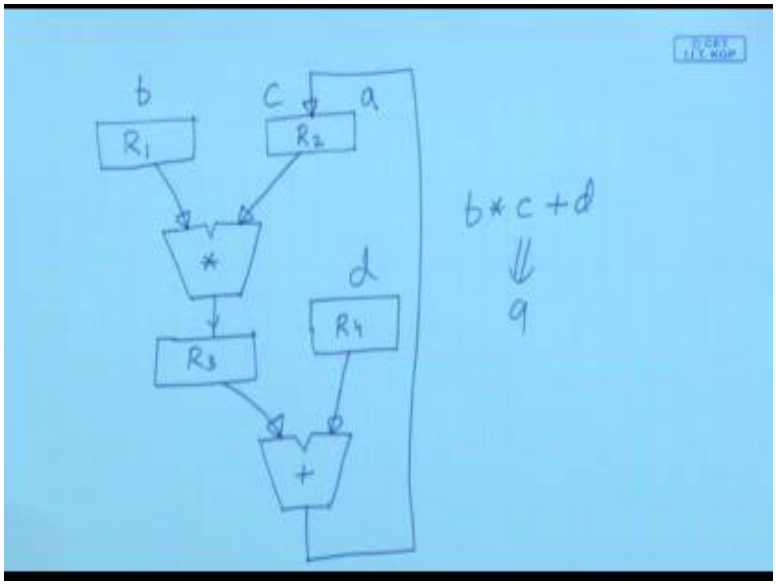
(Refer Slide Time: 23:08)



Let me just try to take an example. Suppose, I have a some operation to do, say, so I am multiplying two numbers b and c and I add it to d. So, this is one way of specifying what I want to do. But, when I talk about behavioral level representation, so this same specification I can translate it into a flow graph, which might look like this. This multiplication is carried out with b and c as the inputs and there will be another addition step, where this will come and also d will come and finally you get a. This is one example. Suppose, I can have an example like this, if some condition is true, then you carry out some step or set of steps s 1, else carry out s 2.

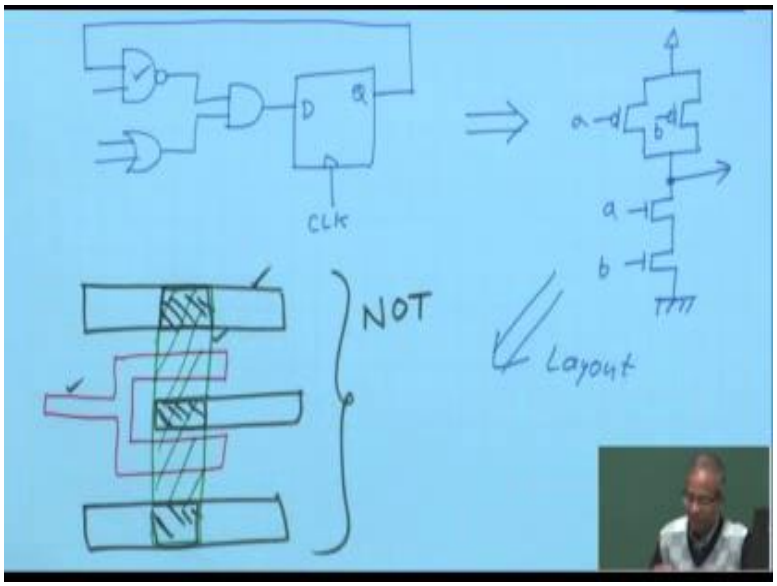
So, again this can be translated into some kind of a data flow graph which can be modelled like this. So, you check the condition a. If this is true, then you go for s 1; if this is false, you go for s 2. So, what I essentially want to say here is that given any high level description in Verilog or VHDL, the first step would be to translate those language constructs into data flow graphs, which can be translated into, you can say, data paths and control paths for the next levels of design. Now, in this diagram you see that your next step is your data path design, where you come to the bus or the register levels structures. Let us see how this register level to structures look like. And, here I am giving an example again.

(Refer Slide Time: 25:31)



Suppose I have a register R 1, which hold some data. So, at this level we are not talking about the variables, but we are talking about the hardware registers. Then, we talk about a multiplier. This is the multiplier circuit. So, it can take the inputs from R 1 and R 2. The output of this multiplier can go to another register say R 3. Then, we can have an adder. So, one input of this can come from R 3 and the other one come from another register R 4. And, suppose the output of this adder, I want to store back in register R 2, then I will have a connection like this. So, this is one way in which we can implement that operation as I had mentioned b into c plus d; where b would be stored here, initially c would be stored here. They would get multiplied, then added with d and the result will be stored in a, which again would be stored here. I will also store a in this register.

(Refer Slide Time: 27:06)



So, this is an example of a register level design. In the next step, we will come to the gate level design. Gate level design: you already know where you can have gates or flip flops. Let me take one very simple example of a gate level design. Let us say I have some gates. Then, I can also have some flip flops. Let us say I have a deep flip flop with the output q. The output can come here, let us say. So, there will be clock input to clock this flip flop. So, gate level design will look more like this. So, the each of the register level components like your registers, multiplexers, then adders, they we will get translated into a gate level design like this. And finally from this gate level design, each of this gates and flip flops can be translated into equivalent transistor representation.

For example, this NAND - this NAND can get translated into this equivalent CMOS gate. These are two n-type transistor that say a and b are the inputs. This will be p-type, this is also a, this is also b and this is the output. This is the CMOS representation of a NAND gate to input NAND gate. And, at the end when you translate this into layout, then you can have something like this. You can have metal layers, two metal layers, then you can have some diffusion layers, then you can have some polysilicon layers, like this. And, the output you can possibly take from here through some connection or contact.

So, this is very rough sketch of a CMOS NOT gate. Now, here you see that the final layout consist of nothing but various rectangular shapes. And, some of the rectangular shaped can be on metal, can be on diffusion layer, can be on polysilicon layer. And, also there are some contact, connections; here there is a contact, here is a contact, here is a contact and so on.

So, in the final layout it is nothing but sequence or a set of large number of rectangular shapes. This is the; you can see in this step when you talk about transistor layout, it is nothing but those rectangular shapes at the end.

(Refer Slide Time: 30:11)

So, this steps in the design flow. I have already mentioned them. Behavioral design, data path design, logic design, physical design and finally, once you have those rectangular shapes, you can create the masks and you can go for fabrication of the chip.

(Refer Slide Time: 30:37)

So now in addition to this, you have a few other step also, which you often carry out. For example, to check whether some step or translation is correct, you typically carry out simulation.

Simulation can be carried out at various levels like logic levels, switch level or circuit level. Formal verification is also very important tool to verify the correctness of designs. And, of course testability analysis and test pattern generation are also very important because you need to test the manufactured devices. So, with this we come to the end of this first lecture.

Thank you.