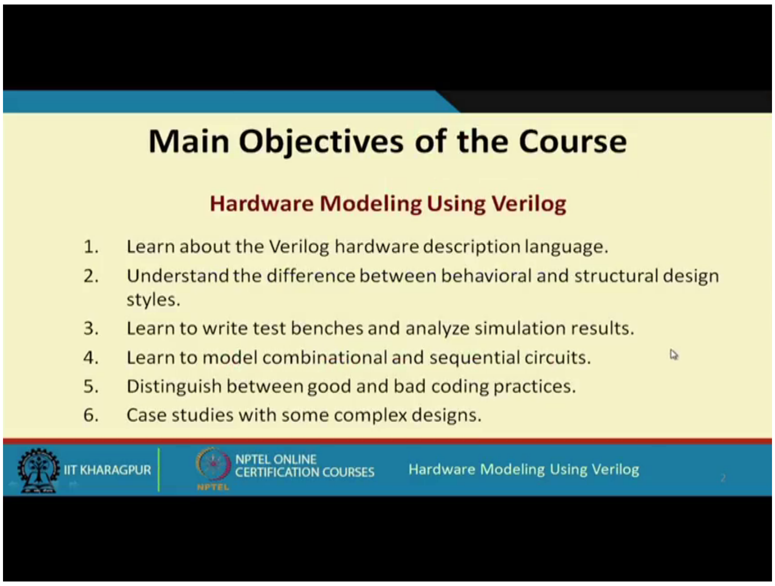
# Hardware Modeling using Verilog
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

## Lecture - 01
## Introduction

Welcome to the course on Hardware Modeling using Verilog. Now in this course over the next 8 weeks, we shall be discussing the various features of the verilog hardware description language, and see that how as a designer you can utilize the facilities and the features that are they are as the part of the language to the best possible extent.

So, today we start with some of the basic introductory topics.

(Refer Slide Time: 00:58)



Main Objectives of the Course

Hardware Modeling Using Verilog

1. Learn about the Verilog hardware description language.
2. Understand the difference between behavioral and structural design styles.
3. Learn to write test benches and analyze simulation results.
4. Learn to model combinational and sequential circuits.
5. Distinguish between good and bad coding practices.
6. Case studies with some complex designs.

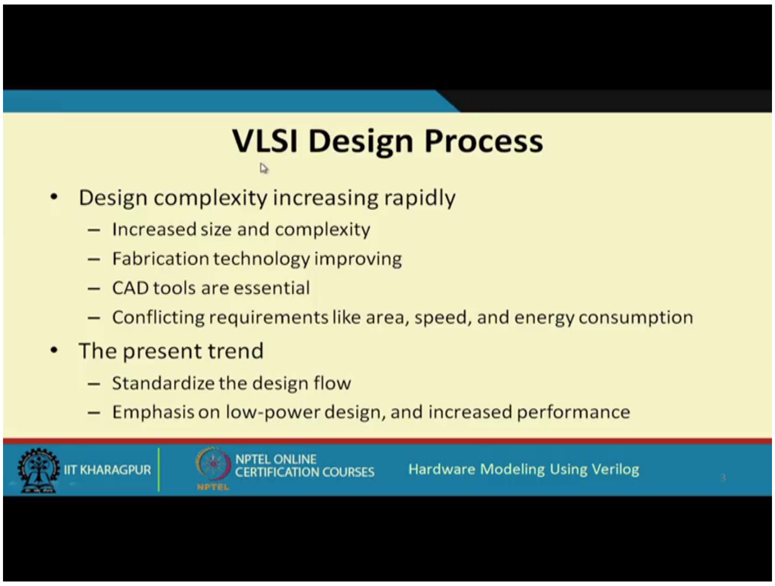
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

So, let us start by talking about the main objectives of this course. So, as you know the name of this course is hardware modeling using verilog. Now verilog is one of the so called hardware description languages that you may already be knowing, it is a language using which a designer can specify the behavior, or the functionality, or the structure of some given hardware; some specified hardware circuit. Now in this as part of this course well we shall of course, be learning about the verilog hardware description language, its various features, the syntaxes, and so on.

Specifically we shall be looking at, two different distinct way of modeling the functionality of a circuit this so called behavioral and the structural design styles. So, we shall be explaining the differences. And from the point of view of verifying whether the design is correctly working or not, its correct or not, we have to write something called test benches, or test harness.

So, we shall also see how to write such test benches and evaluate the results of simulations, we shall be learning about modeling both combinational and sequential circuits. And during the course of this we shall be learning also what are the good practices and what are they so called avoidable practices that a designer should be evade off. And of course, we shall be looking at some of the case studies. Specifically at the end we shall be looking at designer for complete processor and we see how using verilog, we can design the data path and also the control path of the processor, ok.

(Refer Slide Time: 03:06)



VLSI Design Process

- Design complexity increasing rapidly
  - Increased size and complexity
  - Fabrication technology improving
  - CAD tools are essential
  - Conflicting requirements like area, speed, and energy consumption
- The present trend
  - Standardize the design flow
  - Emphasis on low-power design, and increased performance

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

So, we start by talking a few things about the VLSI design process. Because you see whenever you are talking about the hardware description language, you are directly or indirectly talking about some piece of hardware which you are trying to design. Now the first and the most natural kind of hardware building block that comes to our mind is a chip, it is an IC. So, today we are in the area of very large scale integration or VLSI, so we talk about a VLSI chip as our basic hardware building block.

So, when you talk about the VLSI design process there are a few things that we need to keep in mind. First thing is that over the years the complexity of the VLSI circuits and consequently there deign they have increased dramatically. So, means when I say it is incase dramatically then means in fact there has been an exponential increase over the years, I shall show you a slide just depicting the kind of increase that has taken place. But the point to notice that earlier few decades back we use to design some chips, which consisted or contained few 100 or 1000 of gates or transistors, but now today we are talking about circuits or chips consisting of billions of transistors. So, you can just see the difference the dramatic advancements, and improvements that has been taken place over the years, ok.

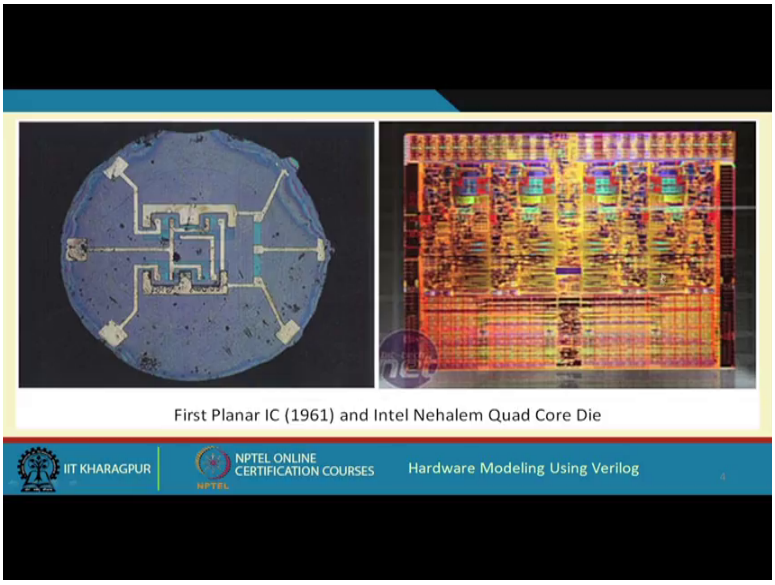
So, because of this increased size and complexity, this has been met possible of course because of improvement in the fabrication technology. The VLSI fabrication technologies have improved dramatically. And as a consequence because of the complexity of the circuits manual design is simply ruled out. So, earlier when the circuit was smaller you could have designed your circuits on a piece of paper layout on a piece of graph paper and so on and so forth, but now when we are talking about millions and billions of transistor you have to make use of a computer system and use some so called computer aided design tools ok they cad tools.

And in the process there is some conflicting requirements that often come, in front of the designer. Like for example, the designer may want to reduce the area, the designer may want to increase the speed the designer may also want to reduce the energy consumption, because as you know most the circuits today are working on battery. So, it is quite natural to try and conserve the power or the energy in the battery for a longer period of time.

So, it is very important to come up with some design ideas or principles that will consume less energy, right. But often these requirements are conflicting. When you try to reduce area may be your delivery increase, may be if you want to reduce the power, your area will increase and so on. So, these requirements are not independently controllable. If you try to optimize one you may make the other one worse, right.

So, the present trend is to standardize something called design flow means that steps that you need to follow to create a VLSI circuit or a chip. And as I have said the present emphasis is number 1 on low power design and number 2 on increased performance.

(Refer Slide Time: 07:25)



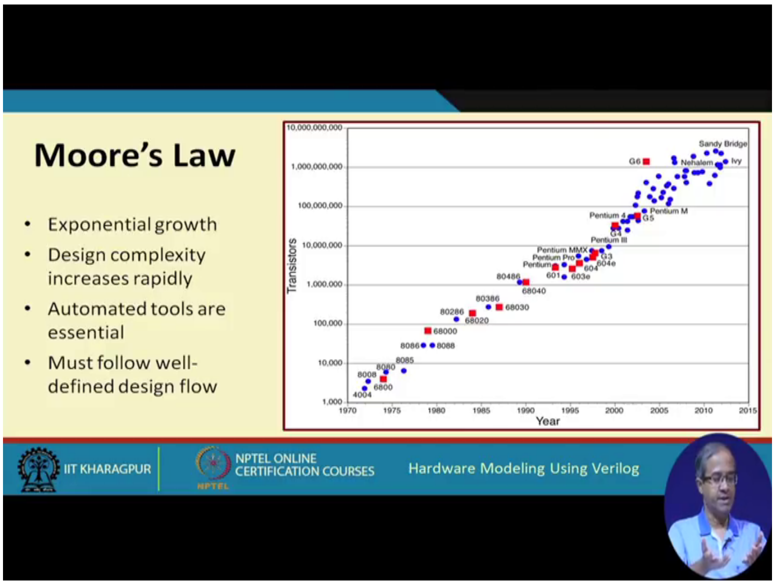
First Planar IC (1961) and Intel Nehalem Quad Core Die

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

So, here in this diagram I am showing 2 circuits side by side. So, on the left you have the first IC planar means it is laid down in a single plan this is called a planar IC. So, you can very easily see the connections the metals and the devices which are prepared, this was a very simple circuit. And on the right side you think of one of the; you see one of the modern processor chips the Intel Quad Core Nehalem series processor. So, here as I have said you have something of the order of billion transistors packed in a single chip.

When you look out the layout in a very compacted way you see a colorful picture like this, where of course the different colors indicate the different layers of the circuit, right,

(Refer Slide Time: 08:23)



## Moore's Law

- Exponential growth
- Design complexity increases rapidly
- Automated tools are essential
- Must follow well-defined design flow

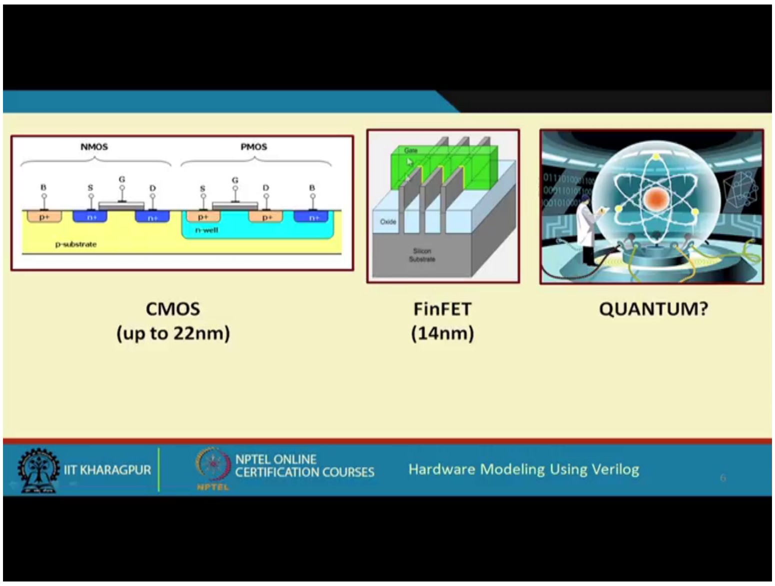
This is the very interesting plot Moore's Law is a very important law in semi conductor design you can say. The persons who are into semi conducted designs, they all know about Moore's Law. There was a person called Gordon Moore who as early as in the 1960s predicted some behavior about the growth in the semiconductor industry. So, what he was said at that time was that- the number of transistors that you can put inside the chip would be increasing exponentially with time, with the number of years that pass.

So, there have been some refinements to this basic you can say law or postulate. So what it is accepted more or less today is something like this- it says that every 18 months or so the number of transistors in a chip single chip will get doubled. Now there were people who had doubted this principle or law since quite a long time in the past. This said that well the devices are becoming smaller and smaller the transistors you are making smaller, so there will be a time, a time will come where you will not be able to make the devices any further smaller.

So, there will be a limit and beyond that Moore's Law will cease to exist. But actually what was happened till today is that, because of semiconductor fabrication advances we are able to fabricate bigger chips, in the process we have been able to sustain Moore's Law; we have been put more circuits in the chip. If you look at this graph, so on the x axis we are plotting year started from 1970 so here it shows up to 2015, and on the y axis you see the number of transistors which is in a log scale see 1000 here up to here it is 1 billion here it is 10 billion. So, you can see there is a straight line kind of a behavior which indicates exponential growth, and here the blue dots refer to the processors which are manufactured by the processor major Intel Corporation, the red dots are the processors manufactured by some other companies ok.

So, Moore's Law as you can see this has continued to hold and this trend is still a straight line behavior which indicates exponential growth over the years.

(Refer Slide Time: 11:30)



CMOS
(up to 22nm)

FinFET
(14nm)

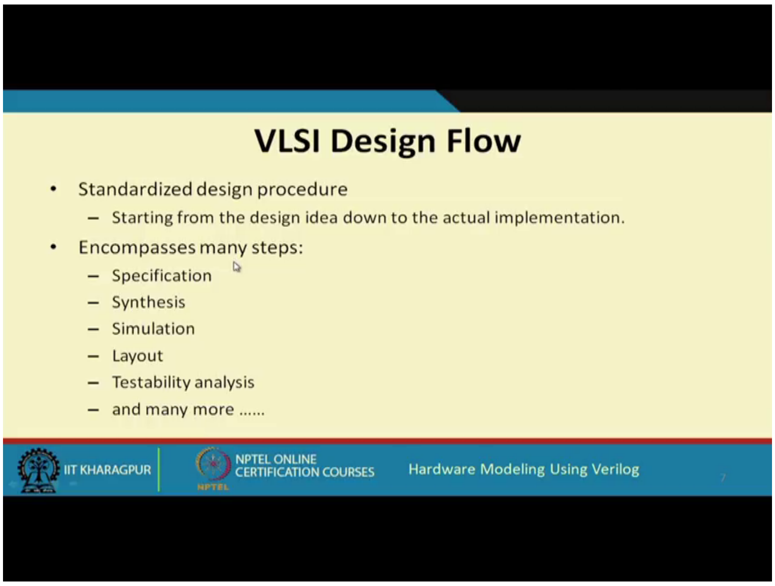
QUANTUM?

Well, the technologies that have made this possible are well CMOS. You may be knowing that CMOS is the most dominant technology today, with which we are manufacturing our VLSI chips, and the CMOS transistors are becoming smaller and smaller and smaller over the years. There is something called feature size which we talk about, that is roughly that is equal to this smallest feature or the transistor that you can fabricate.

Well, the state of that CMOS technology today you can go down up to 22 nanometer; this is traditional CMOS fabrication. But there has been some very innovative kind of CMOS designs also, there is something called FinFET, where the gate drain and the source they are staked vertically instead of horizontally as in the traditional case, and in this way you can tact transistors in a smaller area.

So, today in the FinFET state of the technology you can go down to 14 nanometer. And many of the modern chips that are coming in the market, they are actually manufactured using this FinFET technology, ok. And the picture which is show in the right, this is of course futuristic, this we do not have today, tomorrow quantum computer may come so you may be having a new technology the quantum technology.

(Refer Slide Time: 13:19)



VLSI Design Flow

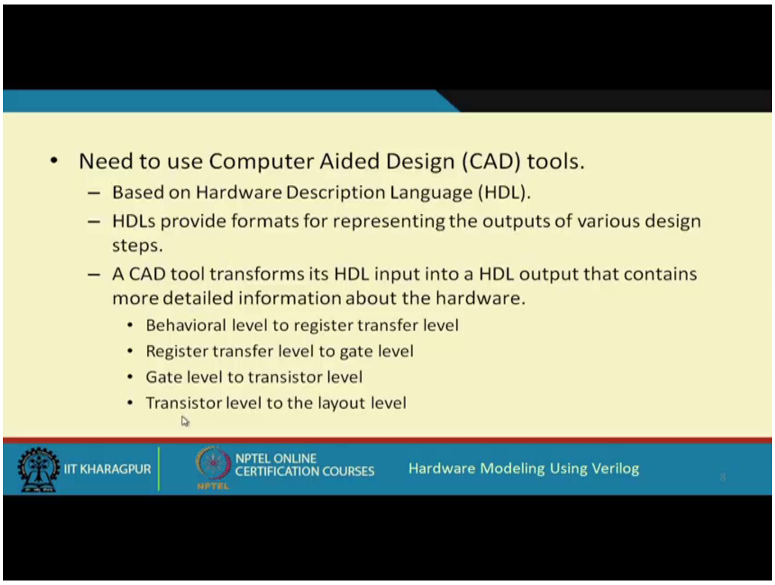
- Standardized design procedure
  - Starting from the design idea down to the actual implementation.
- Encompasses many steps:
  - Specification
  - Synthesis
  - Simulation
  - Layout
  - Testability analysis
  - and many more ......

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

So, looking at the VLSI design flow again; so what is a VLSI design flow? VLSI design flow is nothing but a standardized set of design procedure. Means, here we specify the step by step procedure to be followed starting from our given specification, what you want to do, down to the actual hardware circuit, that you want to built or manufacture. This is the overall design procedure. And this typically encompasses many steps; just a few of them are shown here. Starting from the specification, you go through some steps called synthesis, simulation, layout generation, testability analysis, and then many more steps in between, ok.

So, these steps have to be followed before we can actually get a design fabricated, or manufactured. Now because of the complexity of design as I have said that we need the help of computers. So, it is just beyond the capability of human being to carry out the design in a manual way the more.

(Refer Slide Time: 14:40)



- Need to use Computer Aided Design (CAD) tools.
  - Based on Hardware Description Language (HDL).
  - HDLs provide formats for representing the outputs of various design steps.
  - A CAD tool transforms its HDL input into a HDL output that contains more detailed information about the hardware.
    - Behavioral level to register transfer level
    - Register transfer level to gate level
    - Gate level to transistor level
    - Transistor level to the layout level

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

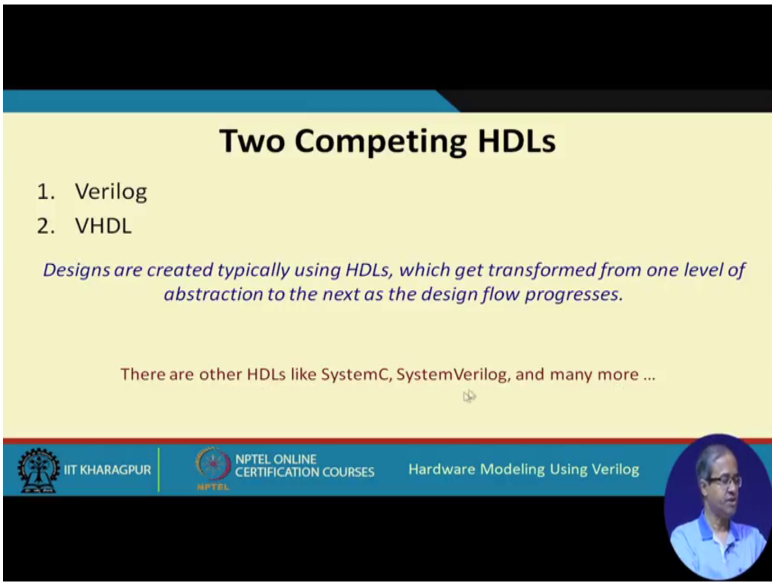
So, you have to rely on computer aided design tools. And this computer aided design tools today are all based on some hardware description language. See just like the high level languages like C, C++ or Java we have a set of language, languages with which we can specify our hardware, and after specifying that we give it to our cad tools and the cad tools will be doing the rest for us, ok.

These tools are based on hardware description language as I have said. These description languages they provide ways to represent designs. Not only the initial design, so as the cad tools translate or transform the designs they will represent this specification at the different steps of transformation as well; this we will see later. So, here is exactly what I was trying to mean. So the cad tool will transform some input which is specified in the hardware description language and generate a output which will also be a hardware description language, but the output will contain more detailed information about the hardware than the input.

Like some of the typical steps in the cad tool transformation as follows. From the behavior you can translate or transform design into a register level design, register transfer level which means from the behavior, you convert it into a form where you have the registers, counters, adders, multipliers, multiplexers, a design at that level ok. Now, once you have done that may be the next step will be to convert each of those functional blocks into gate levels, then the gate level you convert the transistor levels, then each transistor you convert to the final layout level. So, once you have done this your design is ready for fabrication.

So, once you have carried out sufficient analysis and simulation to find out, that your design is meeting your requirements in terms of power consumption and delay, you can send it for fabrication.

(Refer Slide Time: 17:16)



Two Competing HDLs

1. Verilog
2. VHDL

*Designs are created typically using HDLs, which get transformed from one level of abstraction to the next as the design flow progresses.*

There are other HDLs like SystemC, SystemVerilog, and many more ...

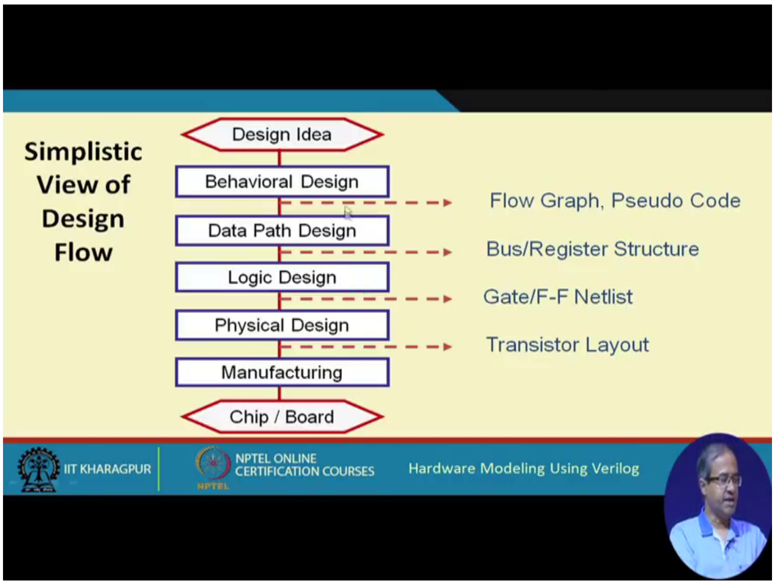
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Hardware Modeling Using Verilog

So, there are two computing HDLs today most popular: so one is Verilog other is VHDL. So, in this course as I have told you we shall be looking at the language verilog. Now earlier as I have said that the designs are created using HDLs, so verilog or VHDLs are very typical examples of these HDLs.

So, you can specify a design in either verilog, or in VHDL and as the cad tools transformed these designs from one level to the next; so the transform design is also expressed in similar kind of hardware description languages.

Now these are not the only ones there are other hardware description languages as well, some of the popular languages are like SystemC, SystemVerilog and so on, but here in this course we shall be concentrating only on verilog.

(Refer Slide Time: 18:23)

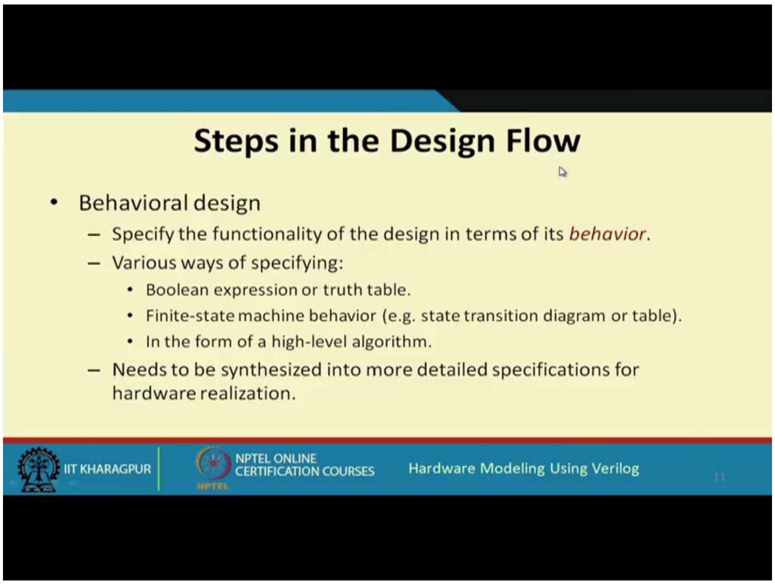


Simplistic View of Design Flow

Talking about the design flow the simplistic view is as follows: starting from a design idea we have to finally come down to our chip design or a board design. So, we typically start with the behavioral design which is in the form of a pseudo code in a hardware description language or some kind of a flow graph notation. In the first level of translation we can convert the behavioral design into something data path design which is the so called register transfer level design, where the basic building blocks are buses, registers, multiplexers, adders, and so on. Then in the next step we convert it in to logic design where you have gates and flip flops, then physical design where you have transistors, then we go for the last step of manufacturing, where the transistors are finally laid out and they are ready to be fabricated on silicon.

Now we can send out designs, to the fabrication house where they can actually fabricate our chip for us, ok.

(Refer Slide Time: 19:39)



## Steps in the Design Flow

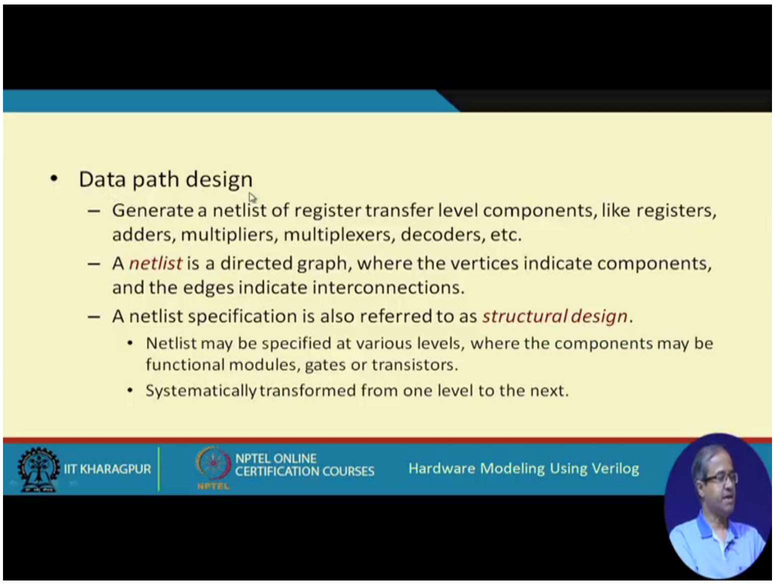
- Behavioral design
  - Specify the functionality of the design in terms of its *behavior*.
  - Various ways of specifying:
    - Boolean expression or truth table.
    - Finite-state machine behavior (e.g. state transition diagram or table).
    - In the form of a high-level algorithm.
  - Needs to be synthesized into more detailed specifications for hardware realization.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

So, talking about the steps in the design flow, the first was the behavioral design. As I have said here, we just specify the functionality of the design in terms of the behavior we do not say, how it is doing it, we just say what we want. Some examples, in the behavioral style; so we can express a Boolean expression (Refer Time: 20:02) function and the form of Boolean expression or in the form of a truth table. If it is a sequential circuit we can express it as a finite state machine: for example, just as a state transition diagram, or as a state transition table. Or you can even specify it a very high level, just like a program written C or C++ or Java in a very high level of abstraction you can write a pseudo code just in the form of an algorithm.

Now, during the design flow the steps, this behavioral design have to be transformed which is called synthesis, synthesized in to more detailed specification as a result of which you will be finally getting your hardware.

(Refer Slide Time: 20:54)

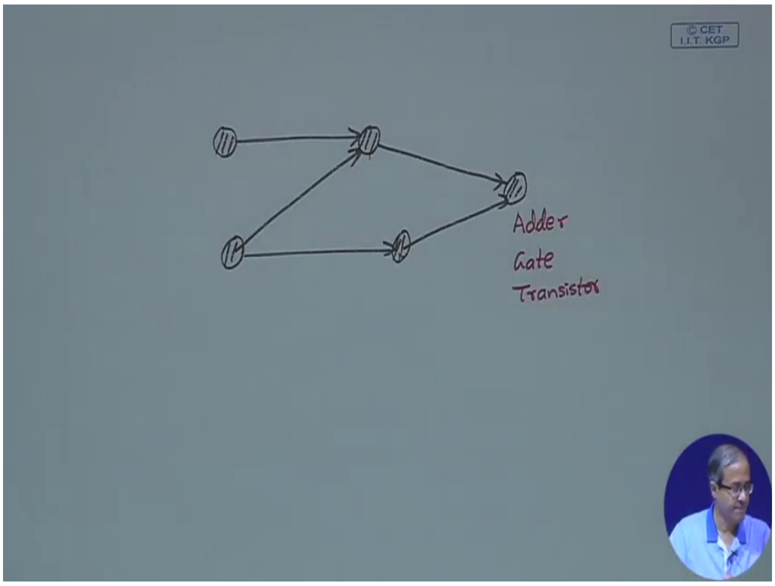


- Data path design
  - Generate a netlist of register transfer level components, like registers, adders, multipliers, multiplexers, decoders, etc.
  - A *netlist* is a directed graph, where the vertices indicate components, and the edges indicate interconnections.
  - A netlist specification is also referred to as *structural design*.
    - Netlist may be specified at various levels, where the components may be functional modules, gates or transistors.
    - Systematically transformed from one level to the next.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

So, data path design as I have said here, we talk about register transfer level components like registers, adders, multipliers, multiplexers, decoders, bus etcetera. Now when you call a netlist; netlist is nothing but some kind of a graph where the vertices indicate components and the edges indicate interconnects.

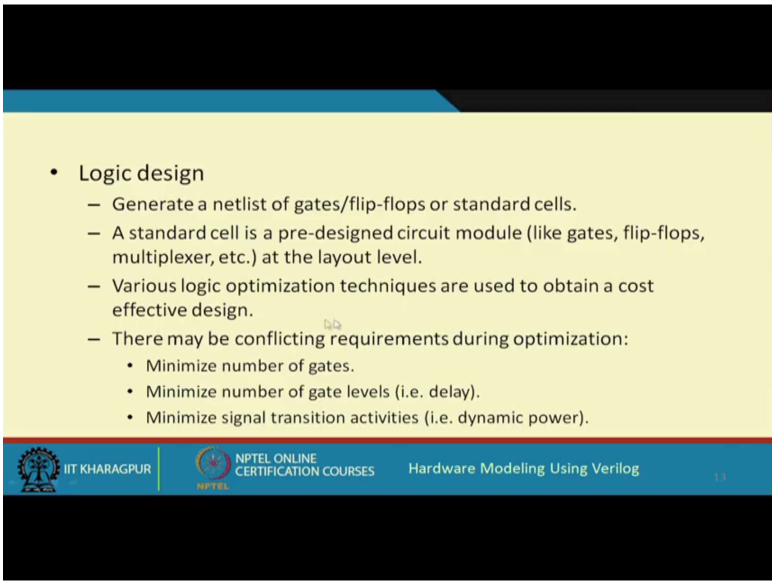
(Refer Slide Time: 21:20)



Like you think of a graph there are some vertices and there are some edges. So, when you say a netlist I have a number of such blocks, and I also specify how these blocks are interconnected.

Now this blocks can be at various different levels. Now this block can be very high level, high level like, it can be adder, it can be a gate, it can be a transistors for example. So I can specify a netlist at various different levels of abstraction, right. So, whenever we specify a netlist this kind of a design specification is also referred to as a structural design. Now this term we shall be using repeatedly when we will be discussing some details about verilog in the next classes. So, netlist as at said, it can be specified at various levels, functional level, gate level, transistor level and so on. And during this synthesis process there will be systematically transformed from one level to the next right.

(Refer Slide Time: 22:46)



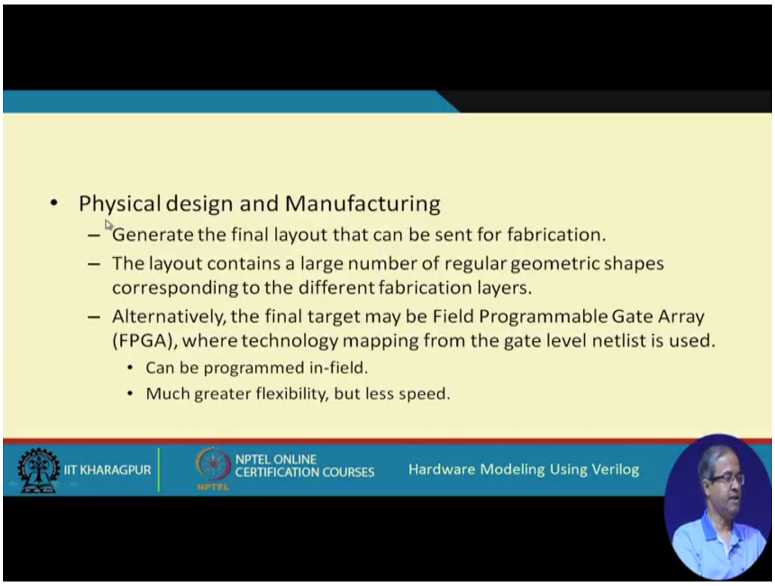
- Logic design
  - Generate a netlist of gates/flip-flops or standard cells.
  - A standard cell is a pre-designed circuit module (like gates, flip-flops, multiplexer, etc.) at the layout level.
  - Various logic optimization techniques are used to obtain a cost effective design.
  - There may be conflicting requirements during optimization:
    - Minimize number of gates.
    - Minimize number of gate levels (i.e. delay).
    - Minimize signal transition activities (i.e. dynamic power).

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Hardware Modeling Using Verilog

Now, when you come to the logic design level; now here we have here again a netlist but now your blocks are gates and flip flops, or something called standard cells. Standard cell like; what is standard cell we should discuss it later. Well a standard cell basically is a pre designed circuit module like, it can be gates, and flip flops, small circuit like a multiplexer, whose layout is already given to you. So, you have this standard cell already present in a library, so in your design if you want you can pick up one of those standard cell you can put them in your layout directly, ok. In that way you create your layout and this standard cell library contains the most commonly used small functional modules in a highly optimized layout form, fine.

And in this type of logic design, various logic optimization techniques are used to minimize your design, to create a so called cost effective design as for as possible ok. Now, as I said earlier that during this process there can be some conflicting requirements, like minimizing number of gates, minimizing delays; that means, number of gate levels, minimizing powers, which means number of signal transitions that are taking place at the outputs of the gates. Now these requirements are often conflicting, if you want to minimize one of these, possibly the others can be increase right, so this is something that you have to keep in mind.

(Refer Slide Time: 24:27)



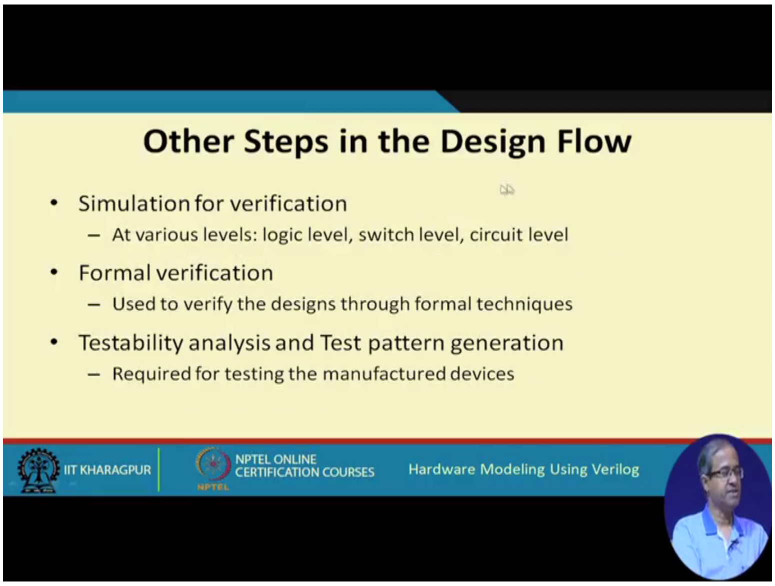
- Physical design and Manufacturing
  - Generate the final layout that can be sent for fabrication.
  - The layout contains a large number of regular geometric shapes corresponding to the different fabrication layers.
  - Alternatively, the final target may be Field Programmable Gate Array (FPGA), where technology mapping from the gate level netlist is used.
    - Can be programmed in-field.
    - Much greater flexibility, but less speed.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Hardware Modeling Using Verilog

And lastly physical design and manufacturing: here, the final layout that is to be sent for fabrication is generated. Now, at this level the layout will contain the large number of regular geometric shapes, because ultimately when you are going for fabrication, you are actually fabricating some patterns on the surface of silicon: metal layer, poly silicon layer, diffusion layer, and all of them have a regular polygonal shapes, that polygons typically rectangular. So, at this level your specification will consist of a very large number of such regular polygonal shapes.

Or suppose you do not want to go for a chip to be fabricated as an alternative, you can also go for something called a field programmable gate array, or FPGA, where from the design you can directly program the device which can be done in field in your laboratory, and as a result you can have much greater flexibility. But as compared to a chip you are fabricating, here the speed will be less ok. This is the trade off you have to realize.

(Refer Slide Time: 25:52)



## Other Steps in the Design Flow

- Simulation for verification
  - At various levels: logic level, switch level, circuit level
- Formal verification
  - Used to verify the designs through formal techniques
- Testability analysis and Test pattern generation
  - Required for testing the manufactured devices

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
NPTEL
Hardware Modeling Using Verilog

So, there are some other steps in the design flow also, these are not the only ones. Like simulation is very important step for verification. Like for example in this course we shall be extensively using simulation to check and verify the verilog modules that we will be writing. We will also be informing you telling you how to do this simulation so that you can do or carry out the simulation yourself, right.

So, this simulation can be carried out at various levels of specification, logic level, transistor level, circuit level and so on. There is a step called formal verification, of course this will be beyond the scope of this course, where using some mathematical and formal techniques, you can check whether your designs are meeting the specifications or not. And again testability analysis text pattern generation is also very important. So, when you manufacture or designs some hardware, you will also have to test whether your final manufactured hardware is working correctly or not. Again this step is beyond the scope of this course we shall not be also talking about testability and testing here, ok.

So, with this we come to the end of this first lecture. In this lecture we have basically tried to give you an overview about what are the things we have expected to cover in this course and some basic concepts of VLSI design flow. Because understanding VLSI design flow the process that is embedded there in, it will allow you to have a better understanding of how you can create a design using a so called HDL it can be either verilog VHDL as I said, so that the final hardware that will be generated in the process can be better in some sense. So, over the course of this lecture that you will follow we shall being trying to address these issues.

Thank you.