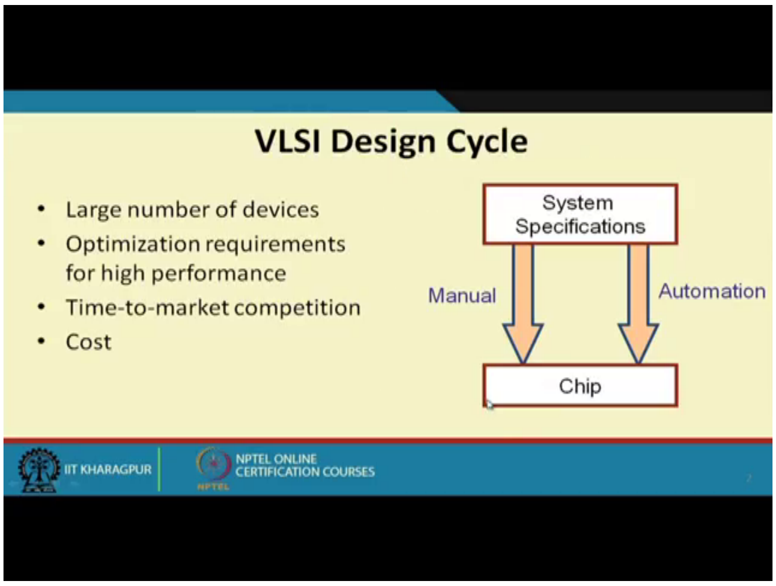
Hardware Modeling using Verilog
Prof. Indranil Sengupta
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur

Lecture - 04
VLSI Design Styles (Part 1)

So, here we now take a break from Verilog and look at some of the so called VLSI design styles. Well, when you talk about VLSI design styles, it means some of this I already talked about in the last lectures like I talked about ASIC application, specific ICI, talked about FPGA, there is something called semi custom and full custom designs, there is something called gate array which falls between FPGA and ASICs.

So, will see some of the features of these design styles, the reason I would be discussing this is that, some of you may be actually designing this kind of circuits. So, you may want to see a correlation between the Verilog coding and the design that you will be doing and your final target hardware. So, there are a few differences. So, unless you understand these differences very clearly, it will be difficult for you to customize or modify your design as you move from one design style to another. So, we start with our discussion on VLSI design styles.

(Refer Slide Time: 01:47)

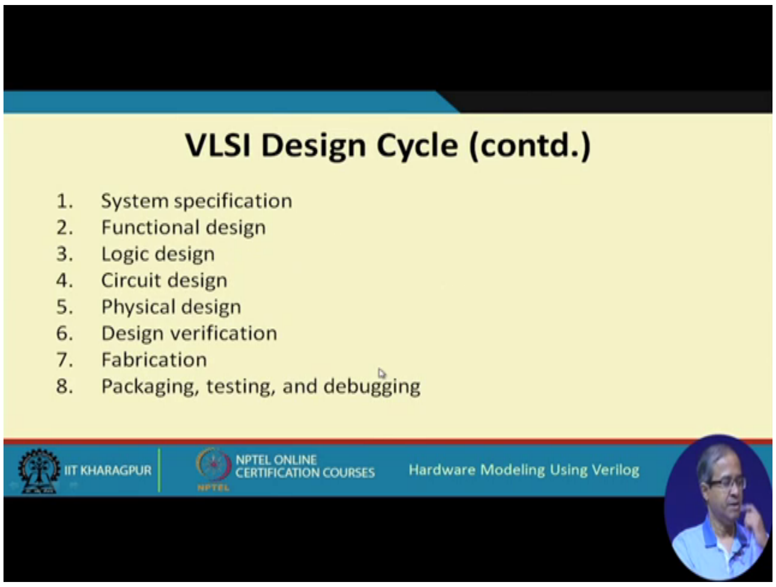

VLSI Design Cycle

- Large number of devices
- Optimization requirements for high performance
- Time-to-market competition
- Cost

System Specifications
Manual
Automation
Chip
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES

Now, VLSI design cycle, already I mentioned a little bit earlier that the VLSI design complexity has increased immensely over the years, manual design or synthesis is simply out of the question, now you have to use computer aided design tools which means automation.

(Refer Slide Time: 02:20)



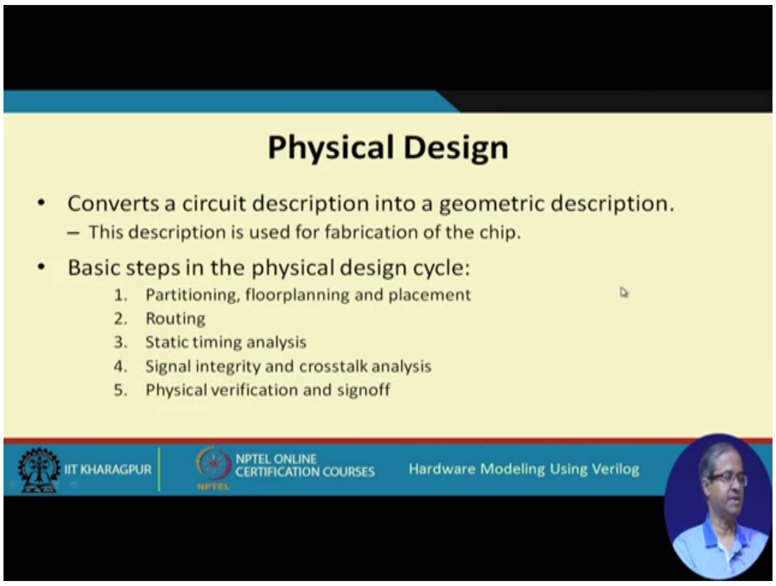
VLSI Design Cycle (contd.)

1. System specification
2. Functional design
3. Logic design
4. Circuit design
5. Physical design
6. Design verification
7. Fabrication
8. Packaging, testing, and debugging

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

So, there are many reasons for doing this time to market computation cost optimization and so on. Talking about the VLSI design cycle, here I am showing you a slightly more detailed stepwise breakup. So, starting from the system specification; you can go to functional design which is the register transfer level design, logic design, gates flip flops circuit design in terms of transistors, physical design. Physical design is something which will again look at a little more design verification, that whether our design is correct finally, fabrication and after fabrication packaging the chip testing and debugging these are the overall steps in the VLSI design cycle.

(Refer Slide Time: 03:06)



Physical Design

- Converts a circuit description into a geometric description.
  - This description is used for fabrication of the chip.
- Basic steps in the physical design cycle:
  1. Partitioning, floorplanning and placement
  2. Routing
  3. Static timing analysis
  4. Signal integrity and crosstalk analysis
  5. Physical verification and signoff

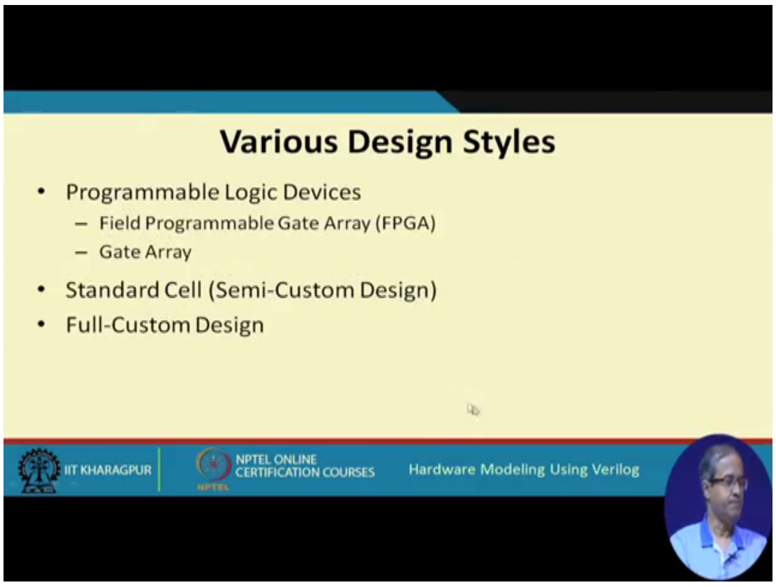
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

But talking about physical design which is somewhere here; so, after your circuit net list is done, we go for physical design before you can actually go for fabrication, right. So, physical design roughly says that I start with some kind of a net list typically at the gate level or the transistor level and from the net list; I ultimately translate it in to my manufacture hardware.

So, my manufacture hardware can be as sake, it can appreciate can anything, but this step is like this I start with a net list at a sufficiently lower level gate level or transistor level and I map it in to my hardware. Now in this steps of physical design, we again may have to go through a number of intermediate steps like there are steps called partitioning, floor planning and placement like you have to decide if it is a large design how to break it up in to smaller partitions and on the surface of silicon where to place them how to plan my total chip floor plan and so on, then comes a very important step of routing.

So, how to interconnect this blocks that I have already placed static timing analysis extremely important in the modern day context. So, after I have completed placement and routing, I need to check that whether my design is meeting my timing constraints or not if not, I may have to go back and change these things. Signal integrity cross stock analysis these are related to timing analysis, this are also very important and when everything is fine, I see that everything is meeting the requirements then I complete my physical verification process and do a step called sign off. Sign off means I am satisfied with my physical design, now I can proceed to fabrication fine.

(Refer Slide Time: 05:22)



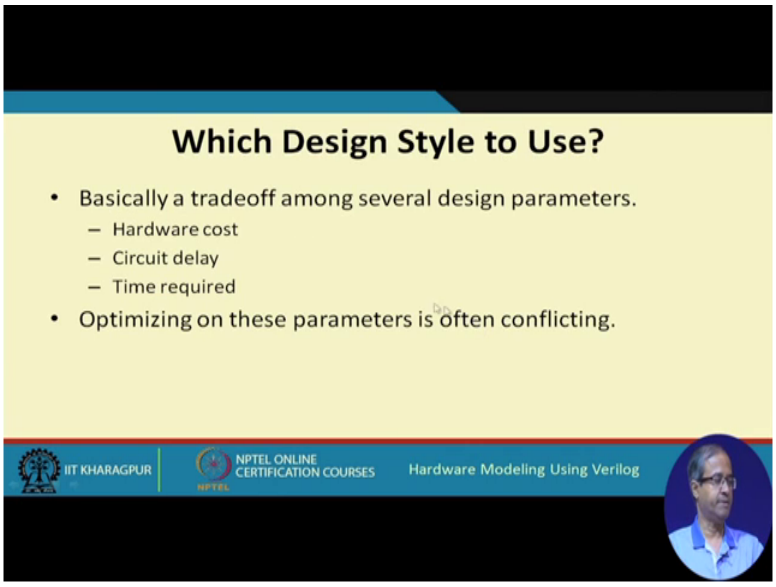
Various Design Styles

- Programmable Logic Devices
  - Field Programmable Gate Array (FPGA)
  - Gate Array
- Standard Cell (Semi-Custom Design)
- Full-Custom Design

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

Now, talking of the design styles, well, I shall be looking briefly in to this 4 design styles there is the first category of programmable devices, I will talk about field program will get our FPGA and also simple gate array is then for standard cell design or semi custom design and full custom design. The last 2 they fall under the category of ASICs application specify I see fine.

(Refer Slide Time: 05:54)



Which Design Style to Use?

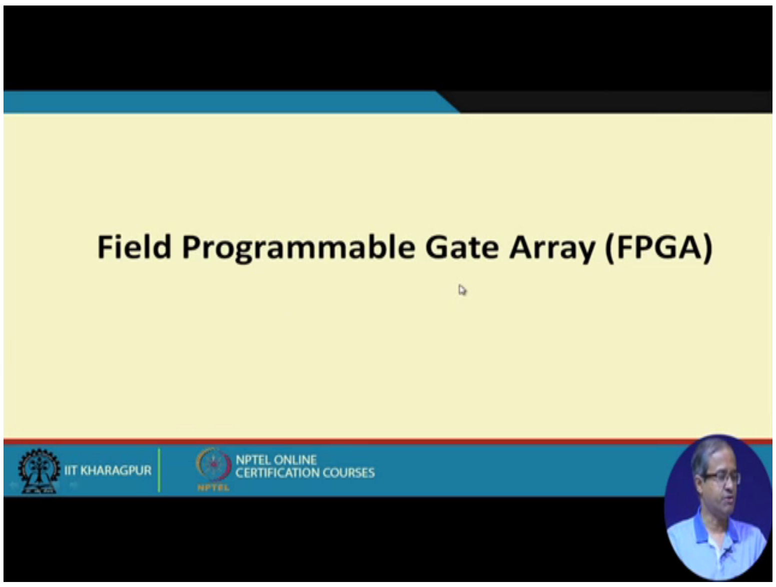
- Basically a tradeoff among several design parameters.
  - Hardware cost
  - Circuit delay
  - Time required
- Optimizing on these parameters is often conflicting.

IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

Now, which design style to use; this is a matter of deciding on part of the designer, there is a tradeoff, there is a tradeoff between hardware cost performance and the total time required for the design. You see FPGA is easiest to design, but the circuit delay may be is bad, but ASIC performance is very good, but hardware cost and time required will be very high.

So, these parameters are often very conflicting. So, when you are going for a particular kind of design we have to look in to the bigger context that exactly for whom we are designing it, who are the potential customers and what are the main objectives that need to be satisfied. So, in that way we can decide on the optimizing criteria in a much better and concise way fine.

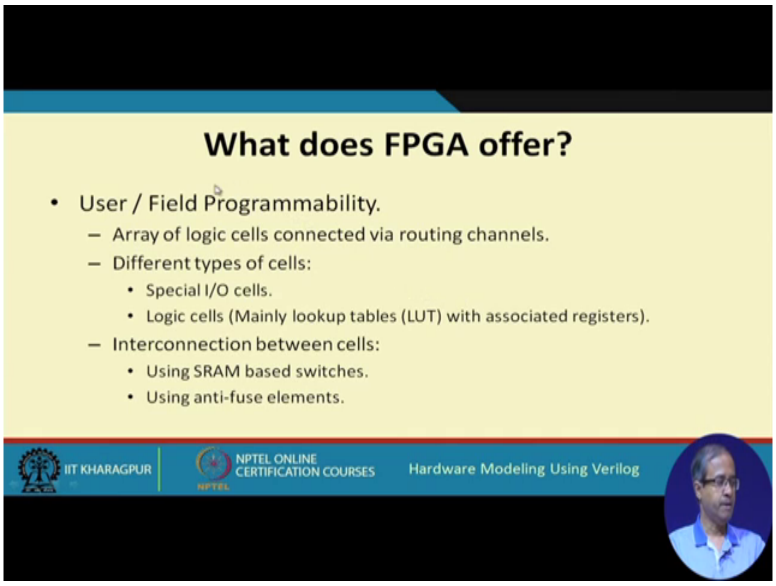
(Refer Slide Time: 07:00)



Field Programmable Gate Array (FPGA)

So, start with FPGA field programmable gate array.

(Refer Slide Time: 07:04)



What does FPGA offer?

- User / Field Programmability.
  - Array of logic cells connected via routing channels.
  - Different types of cells:
    - Special I/O cells.
    - Logic cells (Mainly lookup tables (LUT) with associated registers).
  - Interconnection between cells:
    - Using SRAM based switches.
    - Using anti-fuse elements.

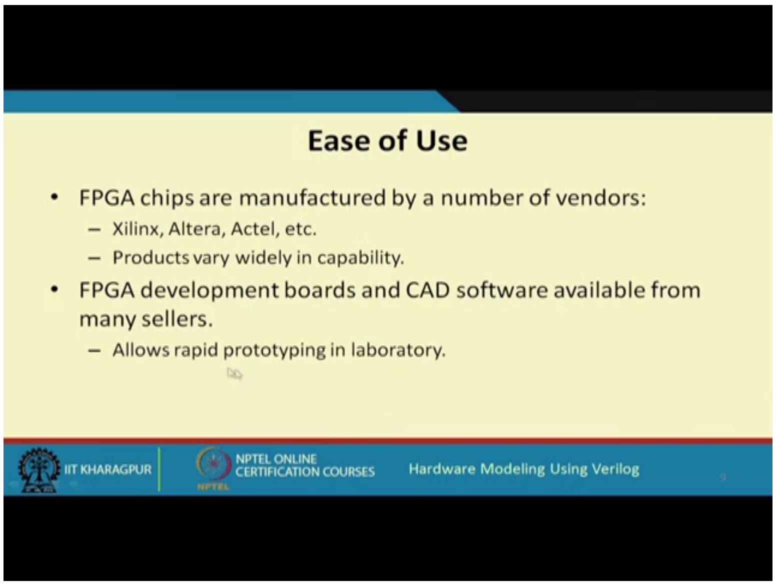
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

Now, FPGA as I mentioned earlier this offers user programmability or field programmability means we can do the programming in our lab sitting on a table, sitting in front of table, we can do it. Now what an FPGA is really FPGA as I said just said it is a programmable device, but inside there is an array of logic cells well array means many thousands of logic cells, they are placed in a regular array and they are interconnected via routing channels. Now both this logic cells and the routing channels are programmable means their functionality can be modified.

Now, in addition to it, there are some other cells called I O cells. Now this logic cells can be either I O cells or they can be something called lookup table blocks and this routing channels interconnections, they are manufactured in different way; either using static RAM or using something called anti fuse static RAM means inside, there are small memories from outside I can store some bit pattern in this memory 0s and 1s. So, if I store a 0 some switch will be open, if I store a one some switch will be closed. So, that way I can the program my interconnection now anti fuse is something which is a little different it is one time. So, by passing a high current between 2 points; so, either I can blow out or I can connect a connection now anti fuse means normally there is no connection.

So, if a high current is flowing then the material will be fusing it will be melting and a connection will be established.

(Refer Slide Time: 09:17)



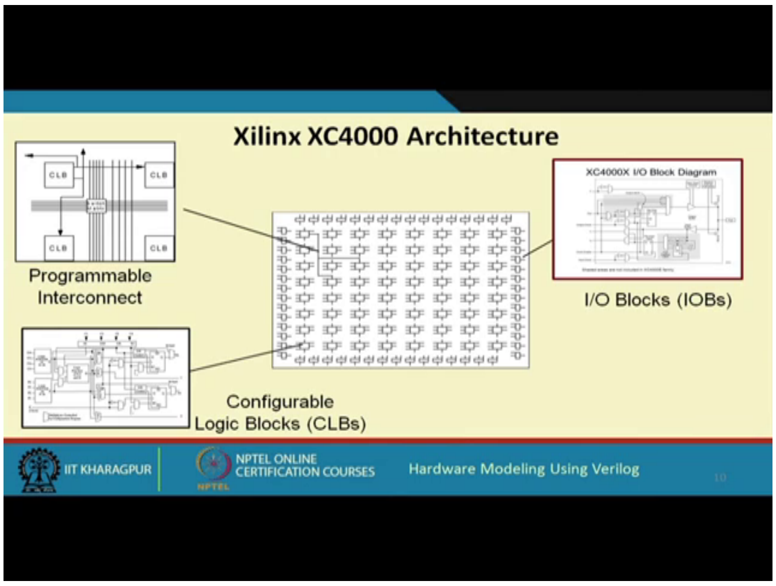
Ease of Use

- FPGA chips are manufactured by a number of vendors:
  - Xilinx, Altera, Actel, etc.
  - Products vary widely in capability.
- FPGA development boards and CAD software available from many sellers.
  - Allows rapid prototyping in laboratory.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Hardware Modeling Using Verilog

So, there is some FPGA manufactures which also use this kind of empty fuse technology fine. So, means; obviously, FPGA's are very easy to use this, this chips are manufactured by a number of vendors like Xilinx, Altera, Actel; the products vary widely in capability they are our FPGA chips now available which are very fast very complex radiation hardened. So, you can map very complex and large designs in to those FPGA chips as well and development boards and cad software are available.

(Refer Slide Time: 09:55)



Xilinx XC4000 Architecture

Programmable Interconnect
CLB
XC4000X I/O Block Diagram
I/O Blocks (IOBs)
Configurable Logic Blocks (CLBs)

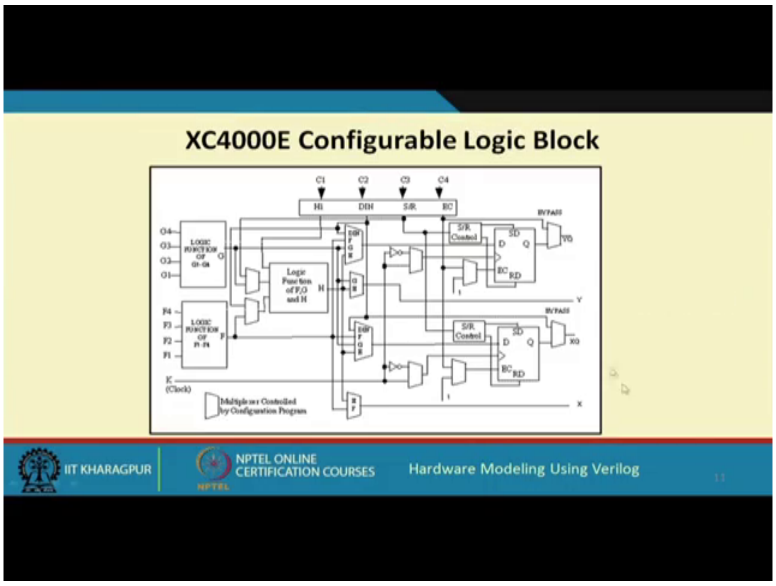
IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Hardware Modeling Using Verilog

Now, here we have a very quick look at a particular FPGA architecture Xilinx XC4000, you see this is a bird's eye view of the whole chip you see at the periphery there are some small rectangles these are the so called I O blocks or I O cells.

This I O block contains some multiplexers flip flops tri state buffers; you see from some external pins, there are signals which can be coming in or going out some of these blocks can be input pins some of these blocks can be output pin. So, by programming these blocks you can configure them, then you have these rectangular blocks in between these are something called configurable logic blocks. So, again by programming this you can make them work in various ways. So, shall we should see a little later. So, how we can do this and in between the configurable logic blocks there is some spaces there are some programmable interconnects are there by again programming the switches here I can connect make connection from one block to the other as per my requirement.

So, whatever design I have, I can map them on to this fabric and I can program them in such a way that appropriate functionality and appropriate interconnections are made.

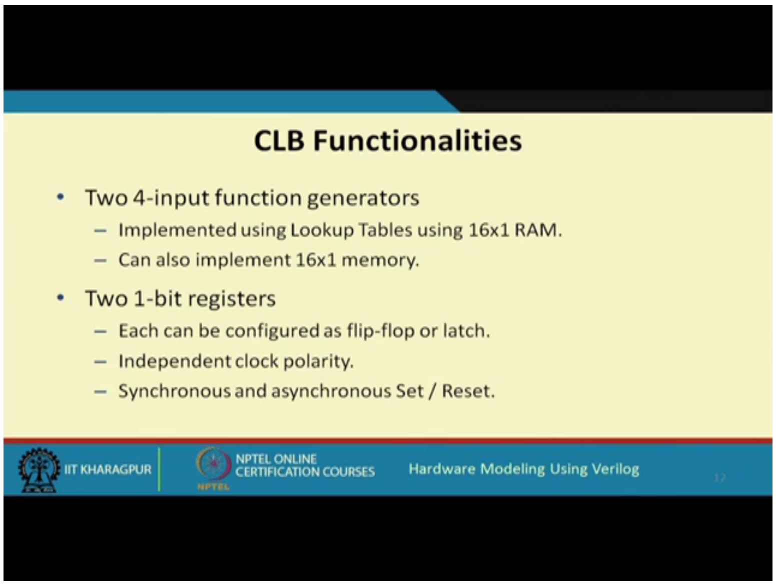
(Refer Slide Time: 11:41)


XC4000E Configurable Logic Block

C1
C2
C3
C4
H1
DIN
S/R
EC
LOGIC FUNCTION OF G1-G4
LOGIC FUNCTION OF F1-F4
Logic Function of F,G and H
S/R Control
BYPASS
YQ
XQ
Y
X
K (Clock)
Multiplexer Controlled by Configuration Program
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

So, that whatever circuit I want to design I want to implement that gets implement implemented on the FPGA chip this is a basic function. So, the configurable logic block it looks like this you see I am not going to detail, but the more interesting thing is that there are 2 blocks here that 2 blocks here which are called look up tables. So, as you can see there are 4 inputs and one output this look up tables can realize any 4 variable function. So, I just explain a little later how and there is also 2 flip flops in the output. So, by properly selecting these multiplexer there are many multiplexers you can see. So, you can actually connect them in a variety of ways right some someway you can program them.

(Refer Slide Time: 12:35)



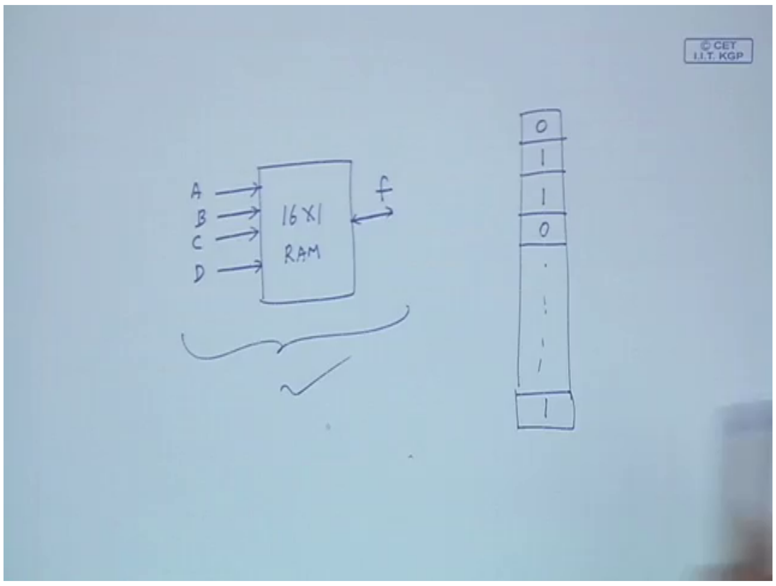
CLB Functionalities

- Two 4-input function generators
  - Implemented using Lookup Tables using 16x1 RAM.
  - Can also implement 16x1 memory.
- Two 1-bit registers
  - Each can be configured as flip-flop or latch.
  - Independent clock polarity.
  - Synchronous and asynchronous Set / Reset.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Hardware Modeling Using Verilog

So, CLB as I said, there are 2 4 input function generators here these 2, then implemented using look up table using 16 by 1 RAM.

(Refer Slide Time: 12:55)

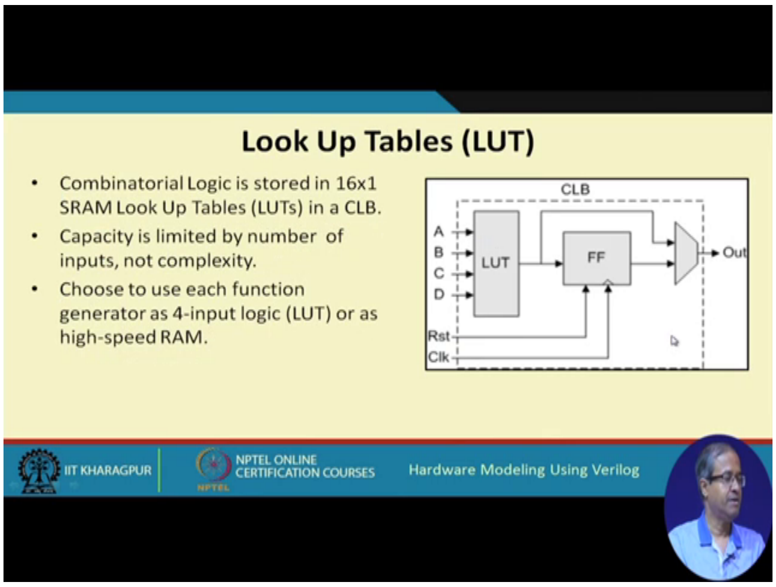


So, how it is done; let us try to explain, suppose I have a 16 into 1 memory RAM, there are 16 words each containing 1 bit. So, there will be 4 address lines and then the 1 data output or input whatever you say. So, in this memory location there will be 16 locations you consider a 4 variable function A, B, C, D. So, any 4 variable functions, if you look at the truth table, there will be 16 columns. Suppose I want to realize some function f of 4 variables I construct the truth table and I just fill up these memory locations by the output column of the truth table. So, once I do this my function is ready. So, I apply a input that corresponding location will be selected and that corresponding output will be generated.

So, just by modifying this memory I can implement any arbitrary function of 4 variables, this is how the programmability comes into the CLBs, right.

So, in addition to implementing 4 functions; 4 input functions, it can also be used as a memory if you require 16 by 1 memory and also inside you have seen that there are 2 flip flops there 2 to 1 bit resistance, they can be configured as a flip flop or a latch clock polarity you can have leading edge triggered falling edge triggered set reset all these facilities are there look up table as I mentioned one look up table is shown here.

(Refer Slide Time: 14:45)



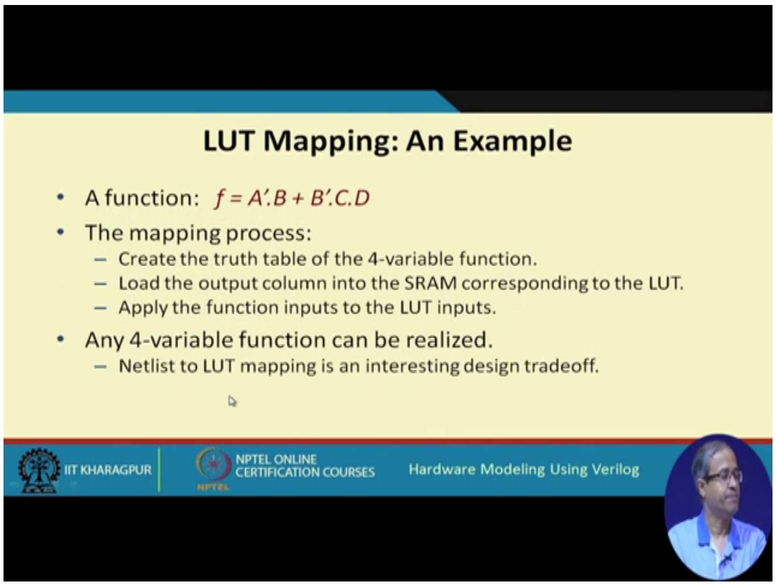
Look Up Tables (LUT)

- Combinatorial Logic is stored in 16x1 SRAM Look Up Tables (LUTs) in a CLB.
- Capacity is limited by number of inputs, not complexity.
- Choose to use each function generator as 4-input logic (LUT) or as high-speed RAM.

CLB
A
B
C
D
LUT
FF
Out
Rst
Clk
IIT KHARAGPUR
NPTEL ONLINE CERTIFICATION COURSES
Hardware Modeling Using Verilog

I am showing 1 look up table flip flop multiplexer. So, this lookup table can implement any function of 4 variables which gives its power as I said.

(Refer Slide Time: 15:05)



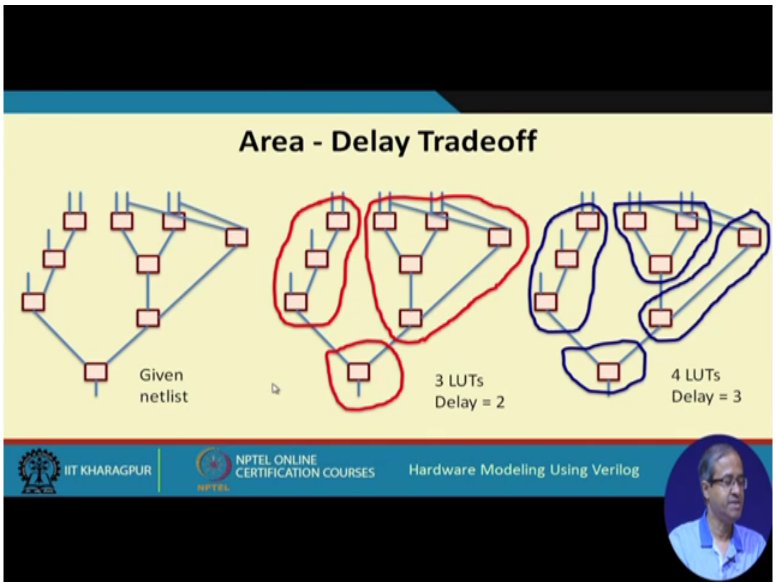
LUT Mapping: An Example

- A function: $f = A'.B + B'.C.D$
- The mapping process:
  - Create the truth table of the 4-variable function.
  - Load the output column into the SRAM corresponding to the LUT.
  - Apply the function inputs to the LUT inputs.
- Any 4-variable function can be realized.
  - Netlist to LUT mapping is an interesting design tradeoff.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Hardware Modeling Using Verilog

So, just one small example; so, if you are given a means function we create the truth table of the function just as I said, load the output column of the truth table in to the SRAM. So, my function is ready. So, I apply A, B, C D to the inputm I get f as the output of the memory.

(Refer Slide Time: 15:35)



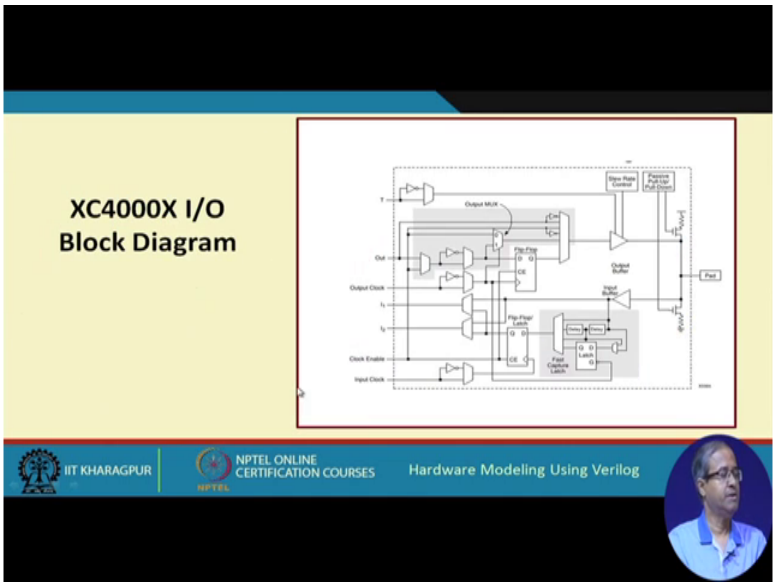
Area - Delay Tradeoff

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Hardware Modeling Using Verilog

So, any 4 variable function can be realized mapping there can be area delayed like means I am just showing an example suppose these are some gates some gates and these are the inputs lines are inputs this is a given net list. So, I want to map them in to those look up tables. So, one way of mapping may be like this, this can be 1, this can be 1, this can be y; you see if I take this part of it, this whole thing there are total 4 inputs, you see 1, 2, 3 and 4. So, overall this is a 4 input and 1 output circuit; this part similarly, this whole thing is 1, 2, 3, 4 and 1 output.

So, any sub circuit with any number of gates with maximum up to 4 inputs and one output can be mapped to tell LUT and here whatever remains I can map into 1 one more LUT. So, so here there is only 2 inputs. So, there are 3 LUTs which are required, here delays 2, this is 1 level of delay, 2 level of delay. Now you think of an alternate mapping; let us say I take this as one, let us; I take this as 1, there is also 4 inputs; 1 output and I take this as 1; 1, 2, 3 input and 1 output and this as 1. So, here there will be 4 LUTs, delay will be 3 because this will be one delay this output is coming here, this will be another delay and then this 3. So, this is a bad mapping this is a better mapping, right. So, this LUT mapping is a big problem this a challenge.

(Refer Slide Time: 17:25)

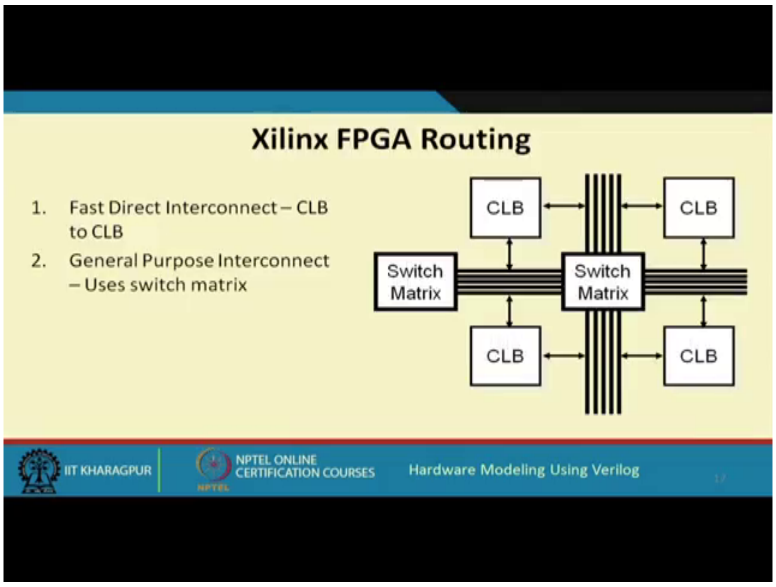


XC4000X I/O Block Diagram

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Hardware Modeling Using Verilog

So, the I O cells are also quite complex as you can see the I O blocks there are also latches there. There are flip flops, these I O pads the tri state controls buffers pull ups, there are a lot of things you can again; there is some RAM, SRAM inside by loading them you can configure the output pins that whether it is an input pin or an output pin whether you require tri state control and so on whether it is a stroked output you need clock leading edge falling edge all these things you can program. So, these are extremely flexible cells. So, by specifying a few bits you can specify exactly how these this individual I O blocks will work.

(Refer Slide Time: 18:17)



Xilinx FPGA Routing

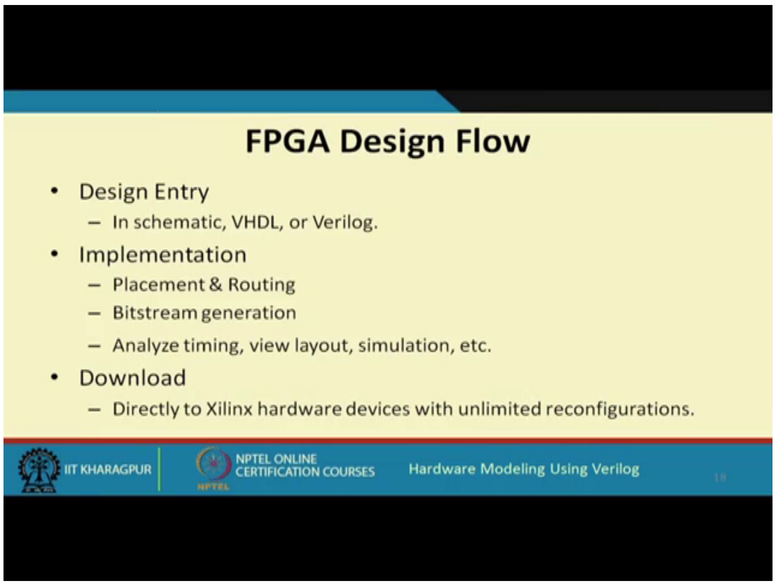
1. Fast Direct Interconnect – CLB to CLB
2. General Purpose Interconnect – Uses switch matrix

IIT KHARAGPUR NPTEL ONLINE CERTIFICATION COURSES Hardware Modeling Using Verilog

So, and talking about the routing see this CLB is there inside which those LUTs are there and in between, there is some space, there are some switch matrixes with some finite number of wires that are connected and this switch matrix is also programmable like one wire of this can be connected to a wire here say a wire here can be connected to a wire here and the CLBs are connected to these wires.

So, by suitably, programming the switch matrix and connecting the CLBs to these; so, we can make any kind of connection say as the output of this, CLB can be connected to the input over this CLB. So, this is also programmable routing interconnection.

(Refer Slide Time: 19:05)



FPGA Design Flow

- Design Entry
  - In schematic, VHDL, or Verilog.
- Implementation
  - Placement & Routing
  - Bitstream generation
  - Analyze timing, view layout, simulation, etc.
- Download
  - Directly to Xilinx hardware devices with unlimited reconfigurations.

IIT KHARAGPUR | NPTEL ONLINE CERTIFICATION COURSES | Hardware Modeling Using Verilog

So, then FPGA you say everything is programmable the logic is programmable logic in terms of the look up table interconnection is program means interconnection is programmable in terms of the switch matrixes and also the I O blocks can be programmed they are also programmable. So, when I say in that board, I have shown earlier that I am downloading some data on the FPGA board where actually downloading all this programming data depending on my circuit net list the synthesis software which is there it will be generating this programming data and if I download it on my chip I will be getting my desired functionality on that chip fine.

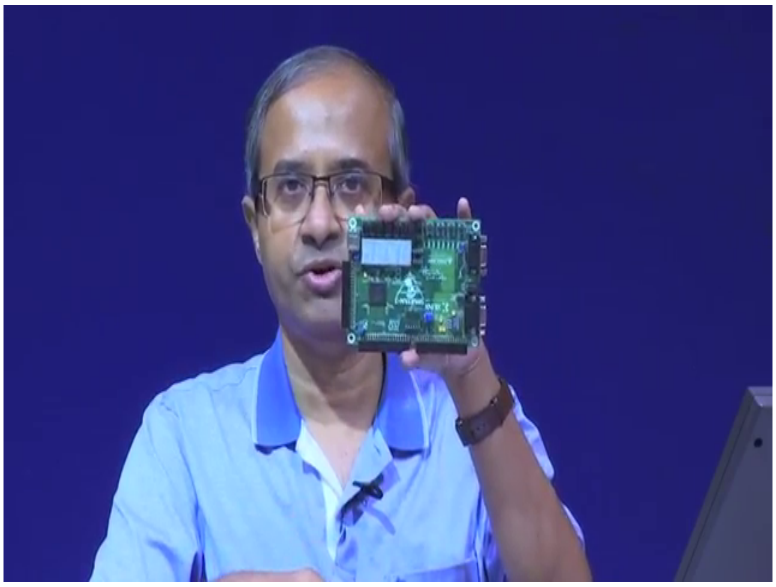
So, FPGA design flow if you just look at it. So, it consists of design entry to start with you can do it in Verilog then the software that is provided by the vendor like I talked about Xilinx, IAC or Vivado; they do placement and routing partitioning, but here it is different here partitioning means partition in to sub circuit with maximum 4 inputs and one output because each partition I will be mapping to one of the LUTs and this LUTs will be placed in to one of this CLBs configurable logic blocks.

Now if you talk about the switch matrixes, you see there is a finite number of wires, right, but if you have a very bad kind of a placement, you have placed them in such a way that lot of wires need to be connected across a switch matrix, but may be one matrix is allowing only 4 wires to be laid. So, you cannot complete the routing see you may have to change the placement and again try the routing.

So, FPGA placement and routing is also a big challenge. So, if we have a good routing, if you have a good placement, then routing will be easy if a placement is not good may be during routing it will fail you may have to move some blocks around again try and may ultimately will be getting out some solution. So, this placement and routing bit stream generation is that programming.

So, whatever you want to program ultimately it is generated in the form of stream of bits and these software, they allow you to analyze the timing behavior layout that in that within the layout which part is heavily utilized which part is not utilized of course, you can also do simulation and check with a function it is correct or not then finally, you can do the download you can directly map it to the Xilinx hardware device like just talking about that that FPGA both that I showed you earlier again.

(Refer Slide Time: 22:15)



So, here I told that there is a port here some pins through which you connect a cable to the p c and you can download that bitmap file here to the FPGA board and this FPGA chip will be automatically configured in that way right. So, with this way come to the end of this lecture and the next lecture which we will look at the other design styles namely gate array semicustom and the full custom design styles.

Thank you.