MINDFUL LEARNING

FPGA Design with Verilog HDL

Introduction:

This is a highly hands-on workshop in which you will learn the FPGA internal architecture and optimize your RTL code according to it. You will learn to use FPGA design tools to create new designs using Verilog and download it on to the FPGA by following the industry standard FPGA design flow. You will also learn to configure new boards (if you already have any) and run desired designs on it. FPGAs can also be used for developing Embedded Systems and in this course you will touch upon those concepts as well. Finally, we will discuss a few case studies and projects which can become a part of your resume. This course will assure to enhance the quality of your resume to match industry standards in a very less timespan.

Objectives:

By the end of this course, you should be able to:

1. Write a hardware friendly RTL design in Verilog HDL
2. Write a test-bench for RTL design verification in Verilog HDL
3. Explain the FPGA design flow in detail
4. Constrain the design according to the timing needs and pin configuration
5. Achieve optimal hardware results by fine tuning the synthesis setting
6. Understand various reports generated by the development tools
7. Use Xilinx IP cores in your design
8. Debug the design on chip using Chipscope
9. Start your embedded systems development on FPGA using Microblaze
10. Learn to use protocols such as UART, SPI and I2C
11. Learn to use DDR memories

Pre-requisites:

1. Logic design fundamentals

Level:

Intermediate

Who should attend?

This course can be taken up by any student or professional who would like to do a systematic study of the FPGA technology.

Package:

1. Verilog HDL ramp-up

   A good design on FPGA starts from writing a hardware friendly RTL design and thoroughly verifying it using a test bench. In this section, we will touch upon those aspects. It is assumed that you have some basic knowledge of Verilog.

2. Guidelines for writing a good HDL design and obtaining desired hardware inference

   Timing and area optimized RTL design is crucial for successfully implementing a design on FPGA. In this section, we will discuss regarding some of the universally followed guidelines for writing an industry standard RTL design.

3. Timing concepts in Digital Systems

   Delay estimation and getting maximum frequency performance is an important step in the development flow. Therefore, it is necessary to get the fundamentals right in this domain. We will discuss how to analyze timing parameters such as delay and slacks in digital circuits. This also helps in understanding the reports generated by the development tools and constraining the design appropriately.

4. FPGA Architecture

   Digital logic is implemented in the FPGA using LUTs, multiplexers and flip-flops. We will discuss in this section various elements present in the FPGA using which digital design gets implemented. We will discuss the Xilinx family of FPGAs.

5. FPGA design flow

   In this section, we will discuss the steps involved from RTL design to final FPGA implementation.

6. IPs in Xilinx ISE

   FPGA vendors, in this case Xilinx, provide design IPs along with their tools which can be readily used in our design to expedite the design development. We will how this is done in this section.

7. Debugging using Chipscope

   Once the design is implemented on the FPGA, there might be several occasions when we may have to probe the internal signals and control a few of them as well. Xilinx provides an excellent solution for this by providing us a debug core that can be inserted in our design. It helps in on-chip run time debug.

8. Embedded Systems design using FPGA

   These days, FPGAs are so capable that a complete SoC can be implemented. In this section we will see how we can configure such a system using a Xilinx provided Microblaze processor. We will take a look at how the interfaces such as UART, SPI and DDR memories can be programmed to meet the specific requirements.

Case studies:

1. Implementing the UART transceiver module on the FPGA
2. Case study on implementing AES encryption and decryption unit on FPGA: uses open source IP cores
3. Case study on implementing crypto-system on a Microblaze system
4. Case study on working with DDR memories

PS: All the above case studies will be demonstrated live on a FPGA board and complete working code will be shared with all participants for further improvements and customization. Appropriate assignments will also be provided.

PS: For online coursework, we suggest that the participant to have access to a simple FPGA board such as NEXYS series from Digilent Inc. or any other low-cost FPGA board. For suggestions please contact us.

Hardware and software:

FPGA boards: Spartan 3A and Spartan 6 boards
Software: Xilinx ISE 13.2 and above

Cost

INR 15000 per head
(Minimum 3 participants for on-site training with in Bangalore city limits)

Duration:

24 hours