FPGA Embedded System Design Course 2024

Course content (Lectures):

1. **Binary System**: Understanding binary representation, binary arithmetic, and binary logic.
2. **Basic Logic Gates**: AND, OR, NOT, NAND, NOR, XOR, XNOR gates and their truth tables.
3. **Flip-flops / Registers / Memory / Latches**: Basic building blocks of sequential logic, including D flip-flops, registers, memory elements like SRAM, and latches.
4. **Karnaugh Maps (K-Maps)**: A graphical method for simplifying Boolean expressions.
5. **Introduction to PAL, PLA, CPLD, and FPGA Architectures**: Programmable logic devices and their architectures, including PAL (Programmable Array Logic), PLA (Programmable Logic Array), CPLD (Complex Programmable Logic Device), and FPGA (Field-Programmable Gate Array).
6. **SRAM & Anti-fuse Technology**: Different types of memory technologies used in digital systems.
7. **Layout Design using Microwind**: Software tools used for digital layout design.
8. **N-MOSFET Design Flow**: Design considerations for NMOS transistors.
9. **FPGA Architecture**: Detailed understanding of FPGA architecture, including I/O blocks, CLBs, and programmable interconnects.
10. **Introduction to VHDL Language**: Hardware description language used for modeling digital sytems.
11. **VHDL Modeling**: Style of modeling, data types, operators, variables, constants, packages, libraries, signals, and process characteristics.
12. **Data Conversion, when-else, with-select, if statements, case statements**: Control structures in VHDL.
13. **Tri-state Drivers / Buffers (CMOS Transmission)**: Circuit elements used for bus sharing.
14. **Comparator, Multiplexer, Shift Register**: Common digital building blocks.
15. **Process Statements**: Combinational and sequential processes.
16. **Component Instantiations**: Designing and instantiating components using VHDL, such as a full adder using half adders.
17. **Serialiser Design**: Designing serial data transmission circuits.
18. **Block RAM Design**: Designing memory elements in FPGA.
19. **Finite State Machine (FSM) Design Flow**: Designing state machines for control logic.
20. **UART Operation (RS232 Protocol)**: Understanding UART communication and its implementation in VHDL.

Demonstrations:

1. **Introduction to Quartus II Software Tool**
    - Overview of Quartus II features and capabilities.
    - Design Methods: VHDL code design and Block diagram (Schematics designs).
    - Simulation and synthesis basics.
2. **Cyclone II Development Board Overview & Schematics Analysis**
    - Detailed examination of the Cyclone II FPGA board components.
    - Understanding the schematics and connectivity.
    - Power supply and pin configuration.
3. **Basic Gate Interface**
    - Implementing basic logic gates (AND, OR, NOT, XOR) using VHDL.
    - Practical examples and testing using Quartus II.
4. **Push Button Input & LED Output**
    - Interfacing push buttons and LEDs.
    - Basic input/output handling in VHDL.
5. **LED Blinking**
    - Creating a simple LED blinking project.
    - Understanding clock dividers and timing constraints.
6. **Pulse Width Modulation (PWM)**
    - Introduction to PWM and its applications.
    - Generating PWM signals using FPGA.
    - Controlling LED brightness with PWM.
7. **LED Shift Operation**
    - Implementing LED shifting patterns (left, right).
    - Using shift registers in VHDL.
8. **Button Debounce Operation with LED Shifting**
    - Combining button debounce logic with LED shifting operations.
    - Advanced input handling techniques.
9. **Seven-Segment Display (SSD) Interface**
    - Interfacing a seven-segment display with FPGA.
    - Displaying numbers and characters on SSD.
    - Multiplexing multiple SSDs.
10. **Synchronous 3-bit Up/Down Counter and 4-bit Counter using SSD**
    - Designing a 4-bit counter and displaying output on a seven-segment display.
    - Designing a counter with up and down counting capabilities.
    - Using push buttons to control counting direction.
11. **Implementation of Phase-Locked Loop (PLL) and Clock Division Techniques in FPGA Designs**
    - Adding a PLL block diagram.
    - Clock Division from 50 MHz to Lower Frequencies.

12. **Multiplexer Example**
    - Implementing multiplexers in VHDL.
    - Practical examples of data selection and routing.
13. **Full Adder and Half Adder Design**
    - Designing and implementing full adder and half adder circuits.
    - Testing and verifying adder functionality.
14. **Finite State Machine (FSM) Example**
    - Introduction to FSM concepts.
    - Designing and implementing FSMs for various applications.
15. **16x2 LCD Interface**
    - Interfacing a 16x2 LCD with FPGA.
    - Displaying custom messages on the LCD.
    - Handling LCD commands and data.
16. **Active Buzzer Interface**
    - Interfacing and controlling an active buzzer.
    - Generating sound patterns and alarms.
17. **Analog to Digital Converter (ADC) Using DC Geared Motor & Potentiometer**
    - Understanding ADC concepts and interfacing.
    - Using an ADC to read potentiometer values and control a DC motor.
18. **Serial Communication (UART)**
    - Implementing UART communication.
    - Sending and receiving data between FPGA and a computer.
19. **Block RAM Memory Interfacing**
    - Interfacing block RAM memory module.
    - Reading from and writing to RAM.
20. **Final Project – FPGA-based Line Follower Robot**
    - Integrating various skills learned.
    - Designing and implementing a line follower robot using FPGA.
    - Sensor interfacing and control algorithms.

Don't miss this opportunity to become proficient in FPGA design! Enroll now and unleash your creativity in digital systems.

Limited seats available. Reserve yours today!