Professional Development Course on VLSI System on a Chip Design (Full-time)

Sandeepani offers the **16-week full time (Monday – Friday)** Professional Development Course for recent graduates and post-graduates in Electronics/Electrical/Telecommunication engineering. This program is specifically designed with an objective to provide a sound platform for the students and prepare them for a successful career in the fields of VLSI Design and verification with FPGAs.

The PDC offers the right blend of classroom teaching, quality hands-on training from 'concept-to-project', covering design methodology using industry standard tools and practices. The course includes a project work as well.

Placement assistance is provided to those who complete all modules of this course and a pre-placement test.

Course Duration: 16 Weeks		Course Structure and Outline
Mod.	Module Title	What You Learn
P1 (3 Days)	Engineering Primer	• Number systems • Logic gates • Boolean expressions • Introduction to registers and counters • Introduction to Embedded systems
C1 (10 Days)	Advanced Digital System Design	• Synchronous Finite State Machine Design • Data-path elements – Arithmetic Structures • Introduction to Programmable Platforms • Design Capture and Simulation • Practical Digital System Design Examples
C2 (10 Days)	Verilog	• Hardware Modeling Overview, • Verilog language concepts • Modules and Ports • Dataflow Modeling • Introduction to Test benches • Operators • Procedural Statements • Controlled Operation Statements • Coding for Finite State Machines • Coding For Synthesis • Tasks and Functions • Advanced Verilog Test benches
C3 (15 Days)	FPGA Design	• FPGA Architecture - Basic Components of FPGA (LUT, CLB, Switch Matrix, IOB), FPGA Architecture of different families: 7-series and UltraScale devices, Zynq • FPGA Design Flow – Xilinx Vivado tool Flow, Reading Reports, Implementing IP cores, Debugging Using Vivado Analyzer • Optimal FPGA Design – HDL Coding Techniques for FPGA, FPGA Design Techniques, Synthesis Techniques, Implementation Options • Static Timing Analysis – Global Timing Constraints, Path specific timing constraints, Achieving Timing Closure,

		Introduction to Reset techniques, Clock Domain Crossing, Multiple Clock Domains
C4 (5 days)	Tcl Scripting	• The need for scripting • Basic syntax of the Tcl Language • Introduction to the Vivado Tcl Environment • Tcl-based project flow and non Project batch flow • Data types, variables and expressions, conditional loops, data structures and Xilinx-specific Tcl commands • Tcl procedures and packages • Creating your own scripts
C5 (10 days)	FPGA design using Zynq	• SoC design flow • System on Chip with Zynq • Processing system and Programmable logic • Designing with Zynq – Zedboards, Zybo • Zynq vs FPGA • Microblaze, Picoblaze and ARM Cortex A9 processor types • IP Block design • IP reuse and Integration • AXI interfacing • Adventures with IP integrator • Introduction to Vivado HLS/SDSoC/MPSoC
Integrated in the course	Course Project	Design/Implementation
Elective 1 (10 Days)	Functional Verification with SystemVerilog	• Introduction to Verification and Verification Plan • Verification Tools • Stimulus and Response • SystemVerilog Basics – Introduction to SystemVerilog, Enhancement made in SystemVerilog over Verilog, Interface and Modports • Introduction to Bus Functional Models • Verification environment and its components • SystemVerilog for Verification - SystemVerilog Event Ordering, Clocking block and Program block, OOP's Concept of SystemVerilog - Parameterized classes, Virtual interface, Constrained Randomization techniques, Functional Coverage (Coverage Driven Verification), SystemVerilog Assertions
Elective 2 (5 Days)	UVM	• Introduction to UVM • UVM Classes • UVM Factory • Sequence Item, Sequencer, Virtual Sequences • Transaction Level Modeling • UVM Reporting Methods • Development of Reusable Verification Environment

Note:

1) The contents listed above is a representative outline and is subject to change at short notice in compliance with the current industry demands
2) Legend: P# - Primer Module, C# - Core Module