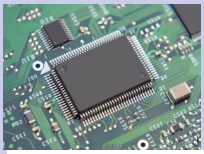

Digital Design Concepts LLC
Verilog Level 1 Classroom Training Description

Course Description (doc version = 1.0.0)

The Verilog Level 1 Classroom Training Course is a three day course that is taught in a traditional classroom environment. At the conclusion of the course, students will have the ability to design and verify digital logic circuits using the Verilog language and commercial EDA tools. Approximately 70% of classroom time consists of teacher instruction, while the other 30% of classroom time is dedicated to laboratory exercises. Not only will students simulate and place and route designs, but also they will download their designs onto development boards to observe the output in real-time.

In addition to providing hands-on training, this course provides more comprehensive details with regards to finite state machine design, synthesis, and simulation than the Verilog Online Level 1 Training Course.

At the conclusion of the course, students will be able to perform the following:

- write efficient code using the latest Verilog constructs, including features of Verilog-2001 and System Verilog
- create sophisticated synchronous designs using Verilog primitives, LSI components, and finite state machines.
- create hierarchical designs
- write testbenches and simulate designs using Modelsim simulator
- synthesize and place and route their designs using Altera Quartus
- download their designs to a development board and view their operation
- write code that follows good programming style

Students that attend this course receive the following:

- Hard copies of all course notes and labs
- A development board identical to the one used in class
- Access to downloadable reference designs

Course Topics

- Introduction to RTL languages
- The typical digital design workflow
- Verilog basics

- Data types
  - Wires, registers, vectors, integer, real
  - Enumerated types
  - Vectors
  - Arrays
- Combinational logic design
  - Continuous Assignments
  - Operators
- Procedural logic design
  - The always block
  - Sensitivity list
  - Blocking and non-blocking statements
  - Combinational logic using always blocks
  - Sequential logic using always blocks
  - Flip Flops
  - If-else statements
  - Case, casex, and casez statements
- Modules and Ports
- Synthesis
  - Characteristics
  - The FPGA logic cell
  - Synthesis optimization techniques
- Case studies -- coding for basic LSI modules
  - Flip/Flop register with reset, clock enable
  - Multiplexer implementations
  - Shift registers
  - Counters
  - Comparators
- Hierarchical design
  - Advantages of hierarchical design
  - Instantiation methods
- Integrating tool generated IP into designs
  - Instantiating
  - Simulating
- Parameters
  - Local parameters
  - Passing parameters through hierarchy
  - Defparams
  - `define
- Finite State Machines
  - Mealy, Moore and mixed type machines
  - Two/Three Process formats

    - Encoding/synthesis strategies
- File I/O
- Testbench concepts
    - Initial blocks
    - Timescales
    - Tasks
    - Functions
    - Loops
    - $system tasks
- Functional Simulation
    - Compiling designs
    - Simulating designs
    - Verifying the results
- Compiling a Programmable Design Using EDA tools
    - Synthesis
    - Place and Route
    - Timing Analysis
    - Programming
- Good design practices
- Future Topics