This document is for the sole use of Giovanni Ferreira of Aluno

Module 1

About This Course

cadence

This page does not contain notes.

© Cadence Design Systems, Inc. All rights reserved.

This document is for the sole use of Giovanni Ferreira of Aluno

Course Prerequisites

Before taking this course, you need to have:

- Basic computer literacy – you must know how to use a shell and editor of your choice and navigate the file system.
- A basic understanding of digital hardware design and verification.
- Knowledge of a procedural programming language will facilitate your learning experience.

 © Cadence Design Systems, Inc. All rights reserved.

cadence

This page does not contain notes.

© Cadence Design Systems, Inc. All rights reserved.

This document is for the sole use of Giovanni Ferreira of Aluno

Course Objectives

In this course, you:

- Use fundamental Verilog language constructs required for design, verification and logic synthesis.
- Analyze and ensure that Verilog designs meet the requirements for synthesis.
- Compare mismatches that can happen between pre-synthesis and post-synthesis simulation and the process of synthesis.
- Debug digital designs by developing Verilog test environments of significant capability and complexity.
- Apply system tasks and functions to verify digital designs using the Xcelium™ Simulator.

 © Cadence Design Systems, Inc. All rights reserved.

cadence

This page does not contain notes.

© Cadence Design Systems, Inc. All rights reserved.

This document is for the sole use of Giovanni Ferreira of Aluno

Course Agenda

Day 1

- Describing Verilog Applications
  - Lab: Exploring the VeriRISC CPU Design
- Verilog Introduction
  - Lab: Modeling an Address Multiplexor
- Choosing Between Verilog Data Types
  - Lab: Modeling a Data Driver
- Using Verilog Operators
  - Lab: Modeling the Arithmetic Logic Unit

Day 2

- Making Procedural Statements
  - Lab: Modeling a Controller
- Using Blocking and Nonblocking Assignments
  - Lab: Modeling a Generic Register
- Using Continuous and Procedural Assignments
  - Lab: Modeling a Single-Bidirectional-Port Memory
- Understanding the Simulation Cycle
  - Lab: Modeling a Generic Counter
- Using Functions and Tasks
  - Lab: Modeling the Counter Using Functions
  - Lab: Modeling the Memory Test Block Using Tasks
- Directing the Compiler
  - Lab: Verifying the VeriRISC CPU Design

Day 3

- Introducing the Process of Synthesis
  - Lab: Exploring the Synthesis Process
- Coding RTL for Synthesis
  - Lab: Using a Component Library
- Designing Finite State Machines
  - Lab: Coding State Machines in Multiple Styles
- Avoiding Simulation Mismatches
- Managing the RTL Coding Process
- Managing the Logic Synthesis Process
- Coding and Synthesizing an Example Verilog Design
  - Lab: Coding a Serial-to-Parallel Interface Receiver

Day 4

- Using Verification Constructs
  - Resolving a Deadlocked System
- Coding Design Behavioral Algorithmically
- Using System Tasks and System Functions
  - Adding System Tasks and System Functions to a Beverage Dispenser Design
- Generating Test Stimulus
  - Verifying a Serial Interface Receiver
- Developing a Testbench
  - Testing a VeriRISC CPU Model
- Example Verilog Testbench
  - Developing a Script-Driven Testbench Using Verilog 1995
  - Developing a Script-Driven Testbench Using Verilog 2001

 © Cadence Design Systems, Inc. All rights reserved.

cadence

This page does not contain notes.

© Cadence Design Systems, Inc. All rights reserved.

This document is for the sole use of Giovanni Ferreira of Aluno

Software and Licenses

For the software and licenses used in the labs for this course, go to:

https://www.cadence.com/en_US/home/training/all-courses/82110.html

If there is additional information regarding the specific software, it is detailed in the lab document and/or the README file of the database provided with this course.

 © Cadence Design Systems, Inc. All rights reserved.

cadence

This page does not contain notes.

© Cadence Design Systems, Inc. All rights reserved.

This document is for the sole use of Giovanni Ferreira of Aluno

Become Cadence Certified by Earning a Digital Badge

Digital badges indicate mastery in a certain technology or skill and give managers and potential employers a way to validate your expertise.

- Cadence Training Services offers digital badges for our popular training courses.
- Your digital badge can be added to your email signature or social media platforms like LinkedIn or Facebook.

Benefits of Cadence Certified Digital Badges

- Validate expertise
  - Expand career opportunities
- Professional credibility
  - Stand apart from your peers
- For more information, go to www.cadence.com/training or email es_digitalbadge@cadence.com.

How do I register to take the exam?

- Log in to our Learning Management System, click on the course in your transcript, and go to the Content tab to locate the exam.

How long will it take to complete the exam?

- Most exams take 45 to 90 minutes to complete. You may retake the exam multiple times to pass the exam.

How do I access and use the digital badge?

- After you pass the exam, you get a digital badge and instructions on how to place it on social media sites.

How is the digital badge validated?

- Credly validates the digital badge as issued to you by Cadence and includes the details of the criteria you completed to earn the badge.

© Cadence Design Systems, Inc. All rights reserved.

cadence

This page does not contain notes.

© Cadence Design Systems, Inc. All rights reserved.

This document is for the sole use of Giovanni Ferreira of Aluno

Icons Used in This Class


Best Practice


Language Syntax


Concept/ Glossary


Frequently Asked Questions / Quiz


Error Message


Problem & Solution


Quick Reference


Tool Command


How To


Lab List

Throughout this class, we use icons to draw your attention to certain kinds of information. Here are the icons we use, and what they mean.

© Cadence Design Systems, Inc. All rights reserved.

cadence

This page does not contain notes.

© Cadence Design Systems, Inc. All rights reserved.