

Essential Verification Methodology

Foundation Level - 2 Tage

Essential Verification Methodology ist ein umfassendes Methodiktraining, dessen Ziel ist, einen Überblick über den Prozess der funktionalen Verifikation sowie der aktuellen Bandbreite von verfügbaren Technologien zur praktischen Anwendung zu geben.

Die Trainingsinhalte werden unabhängig von bestimmten Design- oder Verifikationssprachen und EDA Tools bestimmter Hersteller vermittelt. Der Kurs kann daher als vorbereitendes Training für weitere verifikationsspezifische Doulos-Trainings, wie zum Beispiel 'Assertion-Based Verification with PSL', 'Comprehensive *e*' oder 'Modular SystemVerilog', dienen. Jedes dieser Trainings bietet eingehendes Studium und praktische Workshops in den entsprechenden Sprachen und Methodiken.

15% der Unterrichtszeit bestehen aus schriftlichen Übungen, die das Gelernte verfestigen und herausfordern.

Zielgruppe

- Design- und Verifikationsingenieure, die einen Überblick erlangen und ihr Verständnis der vielfachen Aspekte von moderner funktionaler Verifikation aktualisieren möchten.
- Ingenieure mit wenig oder keiner Erfahrung in Verifikation. Dazu gehören auch:
- Ingenieure aus anderen Fachgebieten (z.B. Softwaredesign oder digitales Hardwaredesign) zur Umschulung bzw. Fortbildung in funktionaler Designverifikation.
- Technische Führungskräfte, denen es an Geläufigkeit mit dem Verifikationsprozess und aktuellen Aspekten funktionaler Verifikationsmethodik mangelt.

Kursinhalte

- Ein tiefgehendes Verständnis der Bedeutung, Definition, Terminologie und Motivation für den Gebrauch von funktionaler Verifikation
- Erstellung und Instandhaltung eines Verifikationsplans
- Simulation und simulationsspezifische Technologien, wie z.B. Entwicklung und Gebrauch von Design-Properties und Assertions, Testbench Automation und Transaction-Level-Modelling
- Formale Verifikationsmethoden – Verifikation eines Designs ohne Testvektoren, sowie der Gebrauch von Design-Properties
- Code Coverage und funktionale Coverage – Fortschrittserfassung im Verifikationsprozess
- Von Hardware unterstützte Verifikationsmethoden und –technologien – Möglichkeiten, den Verifikationsprozess zu beschleunigen

Voraussetzungen

Die Kursteilnehmer benötigen keine Vorkenntnisse in Verifikation, ihnen sollte allerdings der digitale Designprozess geläufig sein. Kenntnisse in Hardwarebeschreibungssprachen (HDL) oder anderen Sprachen in Verbindung mit Verifikation, wie z.B. *e* oder PSL, sind hilfreich aber nicht unerlässlich.

Continued...

Essential Verification Methodology

Foundation Level - 2 Tage

Kursunterlagen

Die Doulos Kursunterlagen sind für ihren umfassenden Informationsgehalt und die äußerst benutzerfreundliche Präsentation allgemein bekannt. In den Kursgebühren ist enthalten:

- Kursskripte mit vollständigem Stchwortverzeichnis, die ein komplettes Referenzhandbuch darstellen
- Ein Arbeitsbuch mit schriftlichen Übungen als Unterstützung bei der Wissensanwendung

Struktur und Inhalt

Overview

Definitions, terminology • Functional verification flow and methods • Linting • Simulation • Debugging • Modelling • Coverage • Assertion-based verification • Formal methods

Verification Process

Verification Plan: creation and maintenance • Verification strategy • Test definition, generation and execution • Re-usable verification IP • Milestones and code reviews • Regression and stress testing • Verification metrics and bug-tracking

Testbenches

HDL testbenches • Testbench architecture • Bus Functional Models and data modelling • Testbench automation • Hardware Verification Languages (HVLs) • Object-oriented and aspect-oriented programming • Stimulus generation, directed, random and constrained-random • Response checking and self-checking testbenches • Variable latency, FIFOs and scoreboarding • Coverage-driven methodology and types of coverage

Properties and Assertions

Properties, their definition and use • Temporal properties • Assertions • Authoring of properties and assertions • Observability and functional coverage • Re-using properties

Formal Verification

Definitions, motivation and terminology • Equivalence checking • Property checking • Coverage • Safety, liveness, invariant • Assumptions • Dynamic formal verification

Hardware Prototyping

Hardware acceleration • Emulation and In-Circuit-Emulation (ICE) • FPGA Prototyping • Observability of internal design nodes • Synthesizable assertions

Appendix - Temporal Logic

Boolean algebra • Temporal logic: CTL and LTL • Fairness

Verwandte Trainings

- Comprehensive SystemVerilog
- Comprehensive *e*
- Expert SystemC Verification
- Assertion-Based Verification with PSL
- Expert VHDL Design & Verification
- Expert Verilog Design & Verification