03-IBFW-DSMV	Digital Systems Modeling and Verification using SystemVerilog

Lehrform *(teaching format)* / **SWS** *(hours per week)*: 2K

Kreditpunkte *(credit points)*: 3

Turnus *(frequency)*: i.d.R. jedes WiSe

Inhaltliche Voraussetzungen *(content-related prior knowledge/skills)*: KEINE

Sprache *(language)*: English

Lehrende *(teaching staff)*: AG Rechnerarchitektur (Prof. Dr. Rolf Drechsler, u.a.)

Studiengang *(degree program)*	Module	Semester
Informatik (Bachelor)	Freie Wahl	ab 4.Sem.
Informatik (Master)	General Studies	

Lernergebnisse:

- Die Bedeutung der RTL-Designmethodik (Huffman-Codierungsstil) verstehen
- Die Bedeutung der mehrschichtigen Testbench-Methodik verstehen und eine robuste Testbench einschließlich der wichtigen SystemVerilog-Konstrukte modellieren können
- Verstehen, wie die Testbench-Qualität und der Verifizierungsprozess mithilfe von SystemVerilog-Verifizierungsmechanismen (z.B. Random Constraint-based Verifikation und Functional Coverage) verbessert werden können
- In der Lage sein, Assertions für eine bestimmte RTL-Designspezifikation basierend auf SVA zu schreiben.

Learning Outcome:

- To understand the importance of RTL design methodology (Huffman coding style)
- To understand the importance of layered testbench methodology and to be able to model a robust Testbench including the important SystemVerilog constructs.
- To understand how to improve the testbench quality and verification process using SystemVerilog verification mechanisms such as Random Constraint-based verification and functional coverage
- To be able to write assertion for a given RTL design specification based on SVA

Inhalte:

- Bedeutung der RTL-Designmethodik (Datenpfad, Controller, Zustandsmaschine usw.)
- Aufbau robuster Testbenches mit SystemVerilog-Konstrukten (Prozesse, Aufgaben, Klasse usw.)
- Einführung wichtiger Verifikationsmechanismen von SystemVerilog (Constraint, funktionale Abdeckung usw.)
- Grundlagen der Erstellung transaktionsbasierter und geschichteter Testbenches
- Gute Praktiken bei der Verifizierung digitaler Systeme mit SystemVerilog Assertions (SVA)

Contents:

- Reviewing the importance of RTL design methodology (datapath, controller, state machine, etc)
- Building robust Testbenches using SystemVerilog constructs (Processes, tasks, class, etc)
- Introduction of SystemVerilog important verification mechanisms (Constraint, functional coverage, etc)

- Fundamental of creating transaction-based and layered Testbenches.
- Good practices in Digital systems verification using SystemVerilog Assertions (SVA).

Hinweise *(remarks)*: The table lists only the primary / most specific modules to which this course is assigned.