

# Verilog crashcourse

Steffen Reith, Thorsten Knoll



Hochschule **RheinMain**  
University of Applied Sciences  
Wiesbaden Rüsselsheim

- 1 Introduction
- 2 Synthesis tool: Yosys
- 3 Verilog elements
- 4 A few basic circuits in verilog
- 5 Selected features in verilog

# Section 1

## Introduction

# Introduction

Verilog wurde 1983/1984 zunächst als Simulationssprache entwickelt, von Cadence aufgekauft und 1990 frei gegeben.

Erste Standardisierung 1995 durch die IEEE (Verilog 95). Neuere Version IEEE Standard 1364–2001 (Verilog 2001).

- Syntax vergleichbar mit C (VHDL ist an ADA / Pascal angelehnt) mit kompakten Code
- Verbreitet in Nordamerika und Japan (weniger in Europa)
- Kann auch als Sprache von Netzlisten verwendet werden
- Unterstützung durch Open-Source-Tools
- Die Mehrheit der ASICs wird in Verilog entwickelt.
- Weniger ausdruckstark als VHDL (Fluch und Segen)

Die Nähe zu C und Java führt evtl. zu Verwechslungen! Auch in Verilog können z.B. Zeilen, die eine kombinatorische Schaltung beschreiben vertauscht werden.

## **Verilog ist eine Hardwarebeschreibungssprache!**

In diesem Abschnitt legen wir auf auf einen Subset von synthetisierbaren Sprachkonstrukten fest.

Ziel unserer Auswahl sind nicht kommerzielle Tools, sondern offene Entwicklungswerkzeuge wie OpenRoad<sup>1</sup> oder Toolchains für bekannte FPGAs, d.h. wir verwenden auch einige Sprachkonstrukte von SystemVerilog, die durch das Synthesewerkzeug yosys unterstützt werden.

---

<sup>1</sup><https://theopenroadproject.org/>

## Literature

- Donald E. Thomas, Philip R. Moorby, Hardware Description Language, Kluwer Academic Publishers, 2002
- Blaine Readler, Verilog by example, Full Arc Press, 2011

# Contributions, mentions and license

- This course is a translated, modified and ‘markdownized’ version of a Verilog crashcourse from Steffen Reith.

<https://github.com/SteffenReith>

- The initial rework (translate, modify and markdownize) was done by:

<https://github.com/ThorKn>

- The build of the PDF slides is done with pandoc:

<https://pandoc.org/>

- Pandoc is wrapped within this project:

<https://github.com/alexeygumirov/pandoc-beamer-how-to>

- License:

GPLv3

## Section 2

Synthesis tool: Yosys



# Synthesis tool: Yosys

Man sollte sich auch mit den Eigenheiten des Synthesetools beschäftigen! Das bekannte Open-Source-Synthesetool yosys schreibt dazu

Yosys is a framework for VerilogRTLsynthesis. It currently has extensive Verilog-2005 support and provides a basic set of synthesis algorithms for various application domains. Selected features and typical applications:

- Process almost any synthesizable Verilog-2005 design
- Converting Verilog to BLIF / EDIF/ BTOR / SMT-LIB /simple RTL Verilog / etc.
- ...

## Section 3

# Verilog elements

# Structure of a verilog module

```
1  module module_name (port_list);  
2  // Definition der Schnittstelle  
3  Port-Deklaration  
4  Parameter-Deklaration  
5  
6  // Beschreibung des Schaltkreises  
7  Variablen-Deklaration  
8  Zuweisungen  
9  Modul-Instanzierungen  
10  
11 always-Bloেকে  
12  
13 endmodule
```

In modernem Verilog können Portliste und Portdeklaration zusammengezogen werden. *//* leitet einen Kommentar ein.

## Section 4

### A few basic circuits in verilog

# Combinational circuits

# Sequential circuits

## Section 5

### Selected features in verilog

# Parameterized Hardware



# The preprocessor

# Yosys and Systemverilog