Technische Hochschule Augsburg

THA

FPGA-Hardwareentwicklung mit Open-Source-Tools

MARCO MILENKOVIC & LUKAS BAUER
MICHAEL SCHÄFERLING & PROF. DR. GUNDOLF KIEFER

Gliederung

FPGA-Hardwareentwicklung
mit Open-Source-Tools

Technische
Hochschule
Augsburg

THA

1. Was sind FPGAs?
2. Entwicklungsboards
3. Open-Source-Tools für Hardware Design
4. SystemC und ICSC
5. Beispiel: ParaNut
6. Ausblick: PicoNut
7. Aktuell: THANNA
8. OSS in der Lehre
9. Fazit

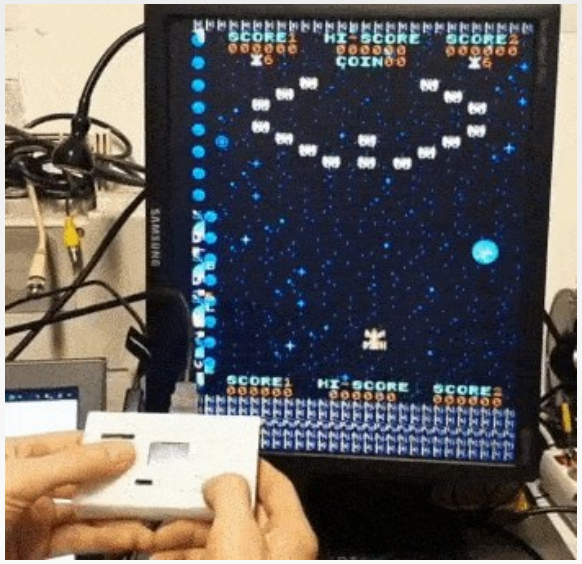
Gliederung

FPGA-Hardwareentwicklung
mit Open-Source-Tools

1. Was sind FPGAs?
2. Entwicklungsboards
3. Open-Source-Tools für Hardware Design
4. SystemC und ICSC
5. Beispiel: ParaNut
6. Ausblick: PicoNut
7. Aktuell: THANNA
8. OSS in der Lehre
9. Fazit

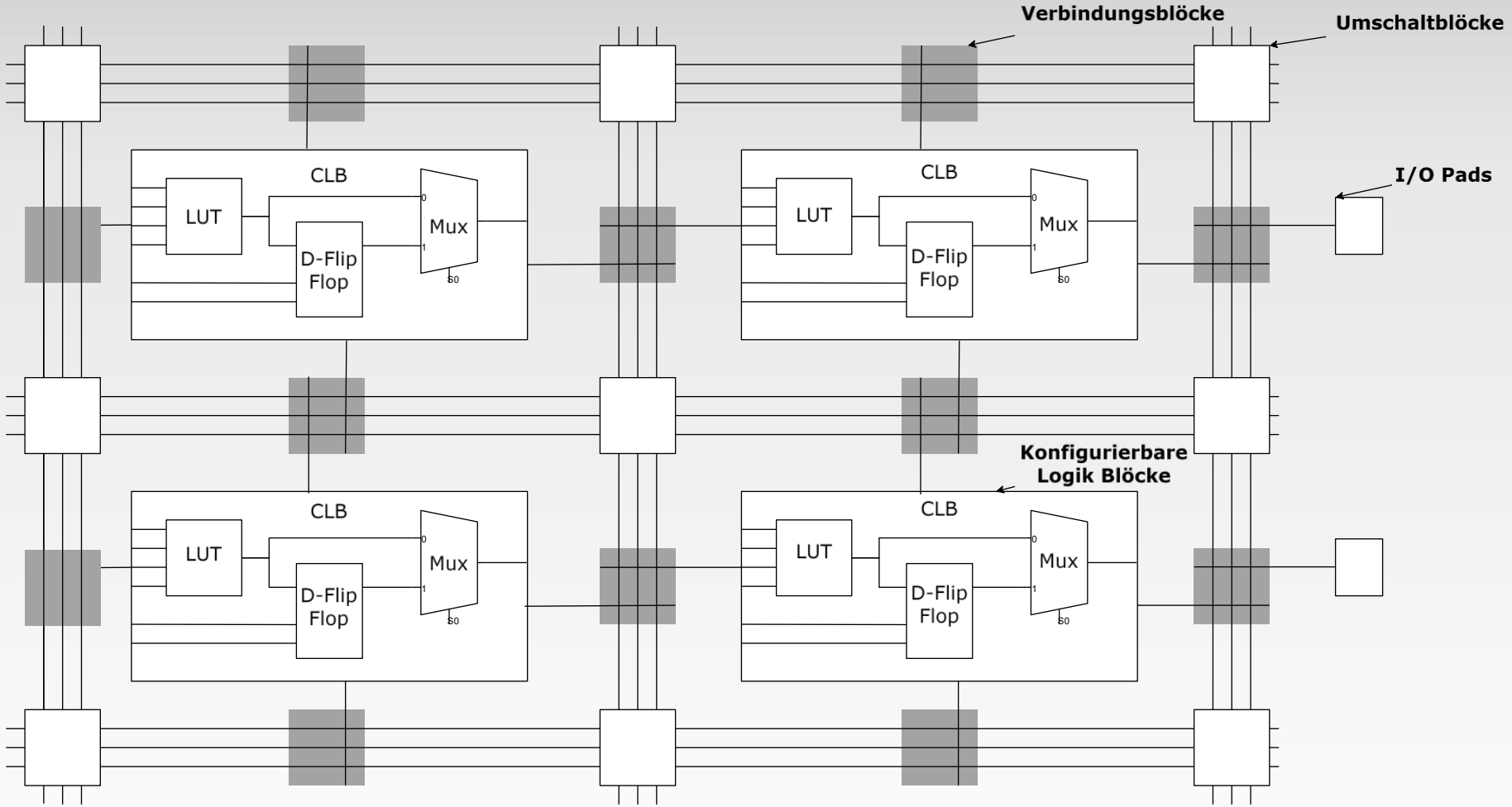
Was ist ein FPGA?

- **F**ield **P**rogrammable **G**ate **A**rray (FPGA)
- Programmierbare Logik-Blöcke
- Verbunden durch programmierbare Leitungen



https://radiona.org/ulx3s/

FPGA-Aufbau


Verbindungsblöcke
Umschaltblöcke
I/O Pads
Konfigurierbare Logik Blöcke
CLB
LUT
D-Flip Flop
Mux
0
1
S0

Wofür FPGAs?

- Wenn Universal-Rechner scheitern
- Ideal für Spezialanwendungen
- Bilderverarbeitung
- Sicherheitsanwendungen
- Prototypen Anfertigung

➢ Ändert sich die Anforderung, ändert sich die Hardware!

HiCoVec

- **Hi**ghly**Co**nfigurable**Vec**torprocessor
- Open Source Hardware
- Komplett von Studenten entwickelt
  - Prozessor
  - Befehlssatz
  - Schnittstellen

Gliederung

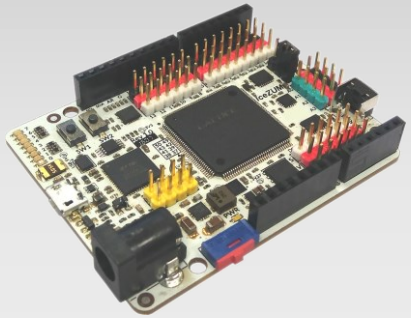
FPGA-Hardwareentwicklung
mit Open-Source-Tools

Technische
Hochschule
Augsburg

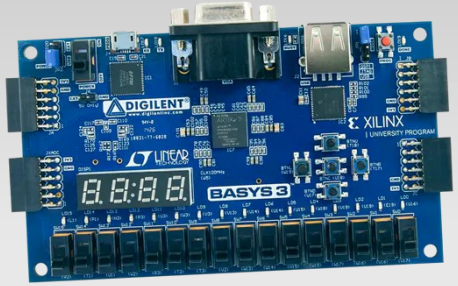
THA

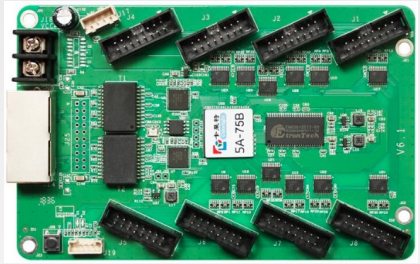
1. Was sind FPGAs?
2. Entwicklungsboards
3. Open-Source-Tools für Hardware Design
4. SystemC und ICSC
5. Beispiel: ParaNut
6. Ausblick: PicoNut
7. Aktuell: THANNA
8. OSS in der Lehre
9. Fazit

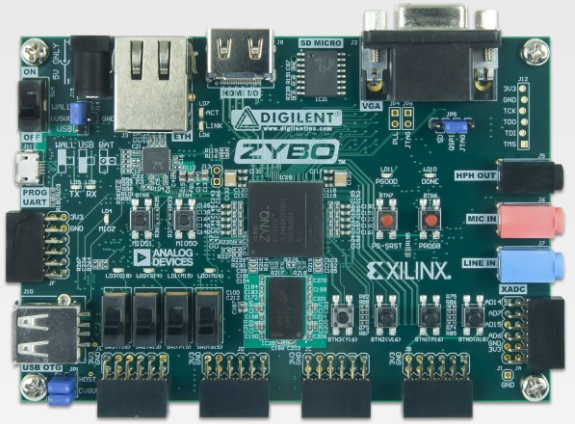
FPGA - Boards

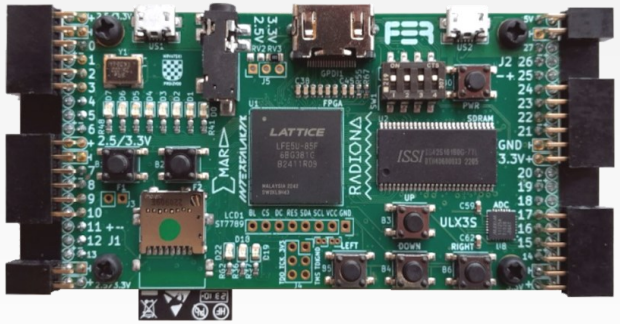


Vielzahl an Boards







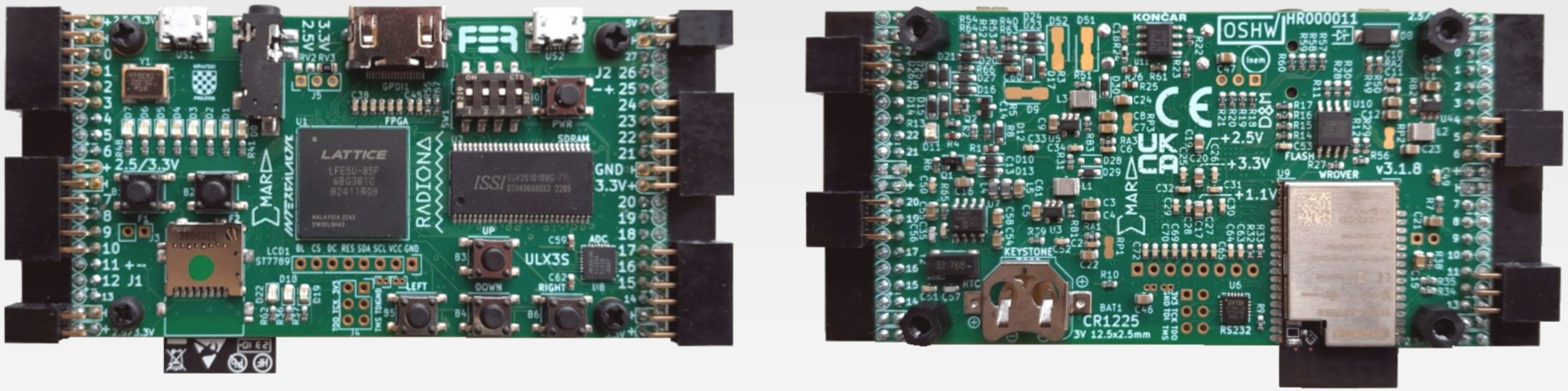


FPGA-Hardware

Board Name	FPGA Chip	LUTs / FFs	Preis ca.
PicoIce	ICE40 (Lattice)	5,3k LUT-4 / 5,3k FFs	34 €
Orangecrab 25F	ECP5 (Lattice)	25k LUT-4 / 25k FFs	95 €
Orangecrab 85F	ECP5 (Lattice)	84k LUT-4 / 84k FFs	195 €
GateMateA1-EVB	GateMate (Cologne Chip)	20k LUT-4 / 20k FFs	50 €
Cologne Chip CCGM1A	GateMate (Cologne Chip)	20k LUT-8 oder 40k LUT-4 / 41k FFs	223 €
ULX3S-12K	ECP5 (Lattice)	12k LUT-4 / 12k FFs	160 €
ULX3S-85K	ECP5 (Lattice)	84k LUT-4 / 84k FFs	260 €
Cmod A7-35T	Arctix 7 (Xilinx)	20k LUT-6 / 41k FFs	120 €
XC7Z010	Zynq (Xilinx)	17,6k LUT-6 / 35k FFs	300 €
XC7Z020	Zynq (Xilinx)	53k LUT-6 / 106k FFs	400 €

Technische
Hochschule
Augsburg

Das ULX3S



ULX3S Spezifikationen

- **FPGA** Lattice ECP5 LFE5U-85F-6BG381C (84K LUT)
- **USB** FTDI FT231XS (500kbit JTAG and 3Mbit USB-serial)
- **GPIO** 56 pins (28 differential pairs), PMOD-friendly with power out 3.3V/1A 2.5V/1.5A
- **RAM** 32MB SDRAM 166 MHz
- **Flash** 4–16MB Quad-SPI Flash for FPGA config and user data storage
- **Mass storage** Micro-SD slot

ULX3S Spezifikationen

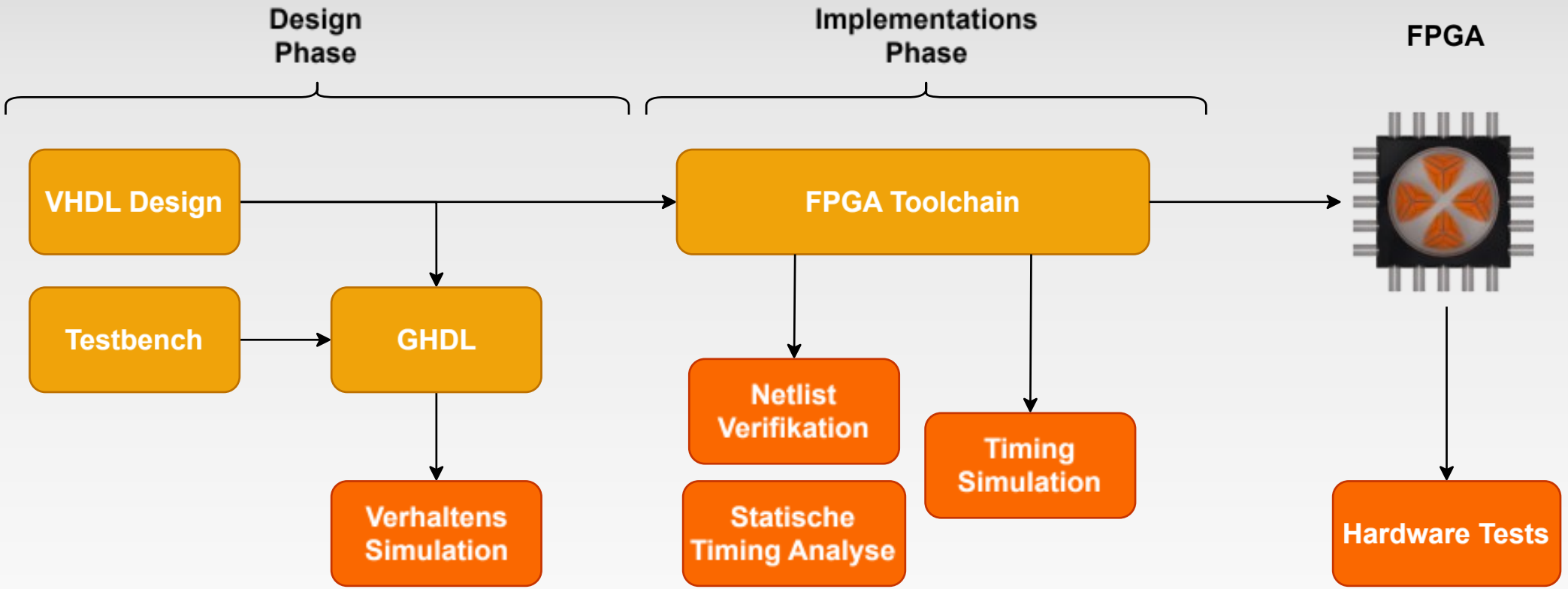
- **Video** Digital video (GPDI General-Purpose Diferential Interface) with 3.3V-5V I2C bidirectional level shifter
- **WiFi+bluetooth** placeholder for ESP-32 (Standalone JTAG web interface over WiFi)
- **ADC** 8 channels, 12 bit, 1 MSa/s MAX11125
- **Clock** 25 MHz onboard, external differential clock input
- **Low power sleep** 5uA/5V standby, RTC MCP7940N clock wake-up, power button, 32768 Hz quartz with CR1225 battery backup

Gliederung

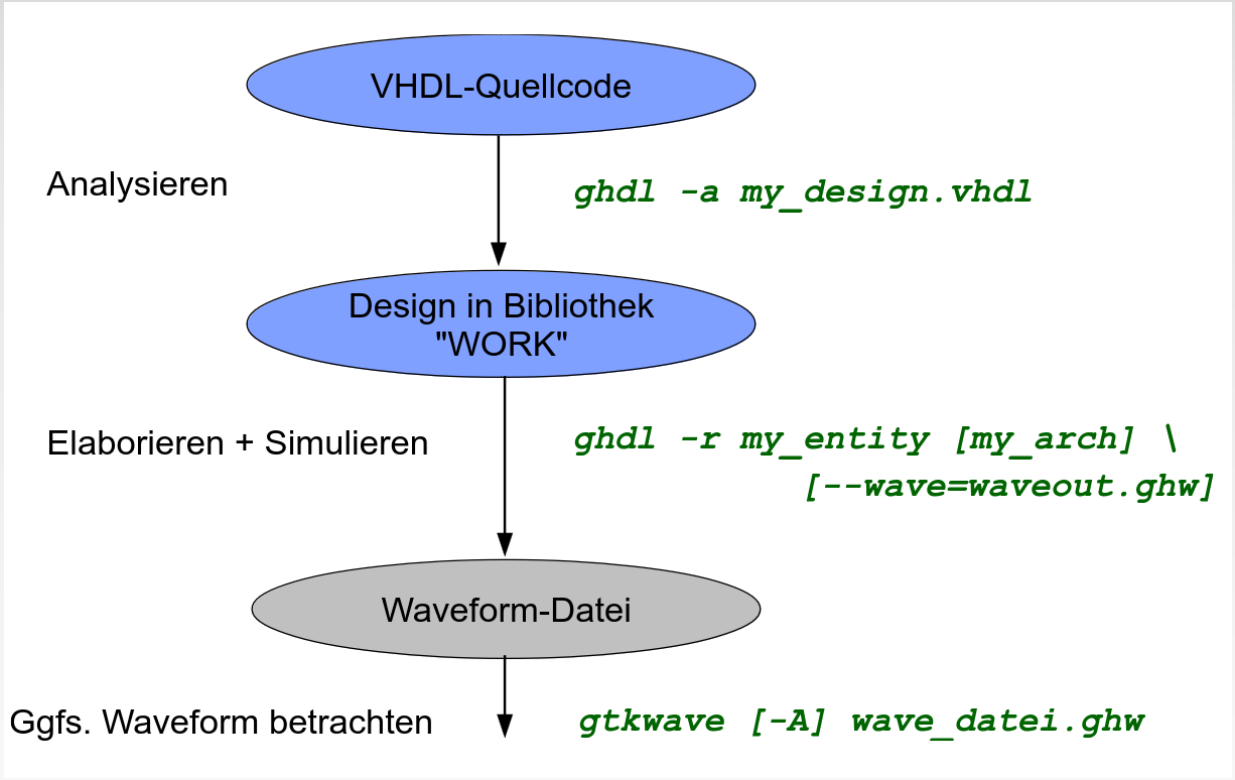
FPGA-Hardwareentwicklung
mit Open-Source-Tools

1. Was sind FPGAs?
2. Entwicklungsboards
3. Open-Source-Tools für Hardware Design
4. SystemC und ICSC
5. Beispiel: ParaNut
6. Ausblick: PicoNut
7. Aktuell: THANNA
8. OSS in der Lehre
9. Fazit

Workflow in der Hardwareentwicklung

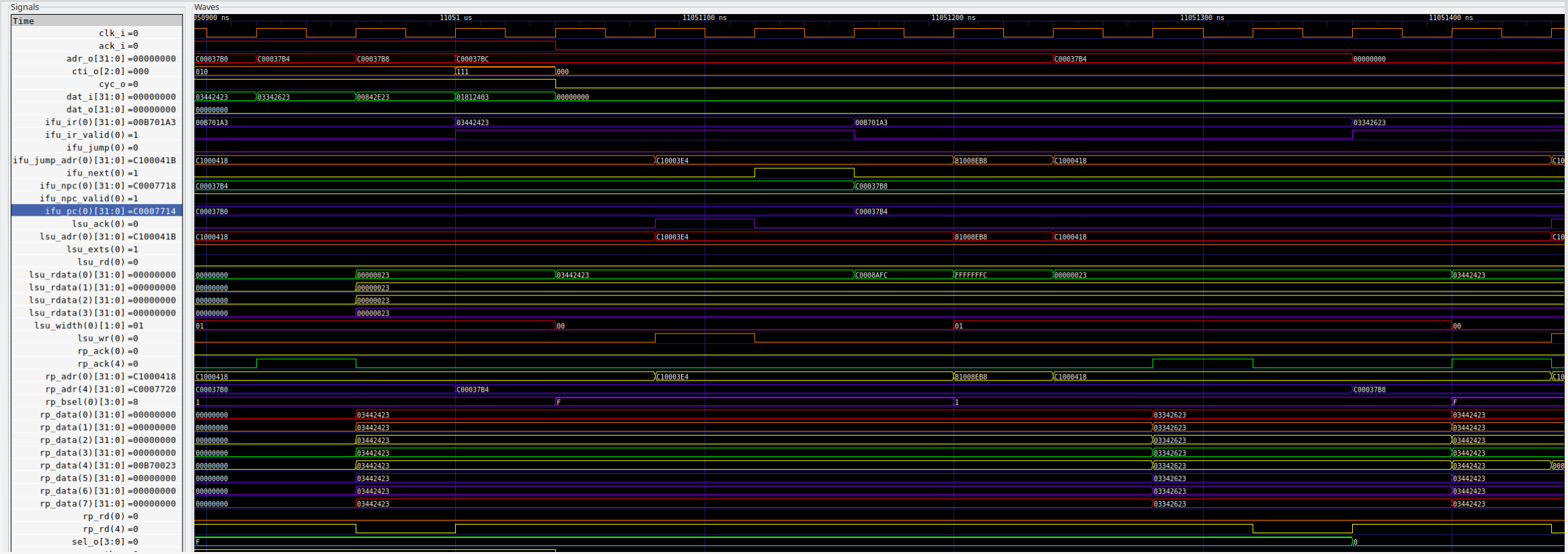


Workflow mit VHDL



- GHDL ist ein Simulator für VHDL
- Simulation über eine Testbench

GTKWave Analyse

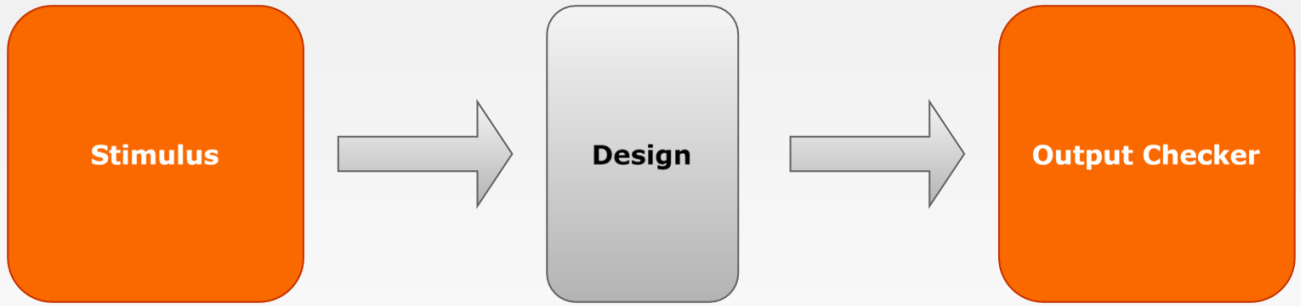


SystemC

- Bibliothek für C++ von Accellera
- Hauptsächlich für Modellierung auf Systemebene
- Ziel: Schnelles Prototyping

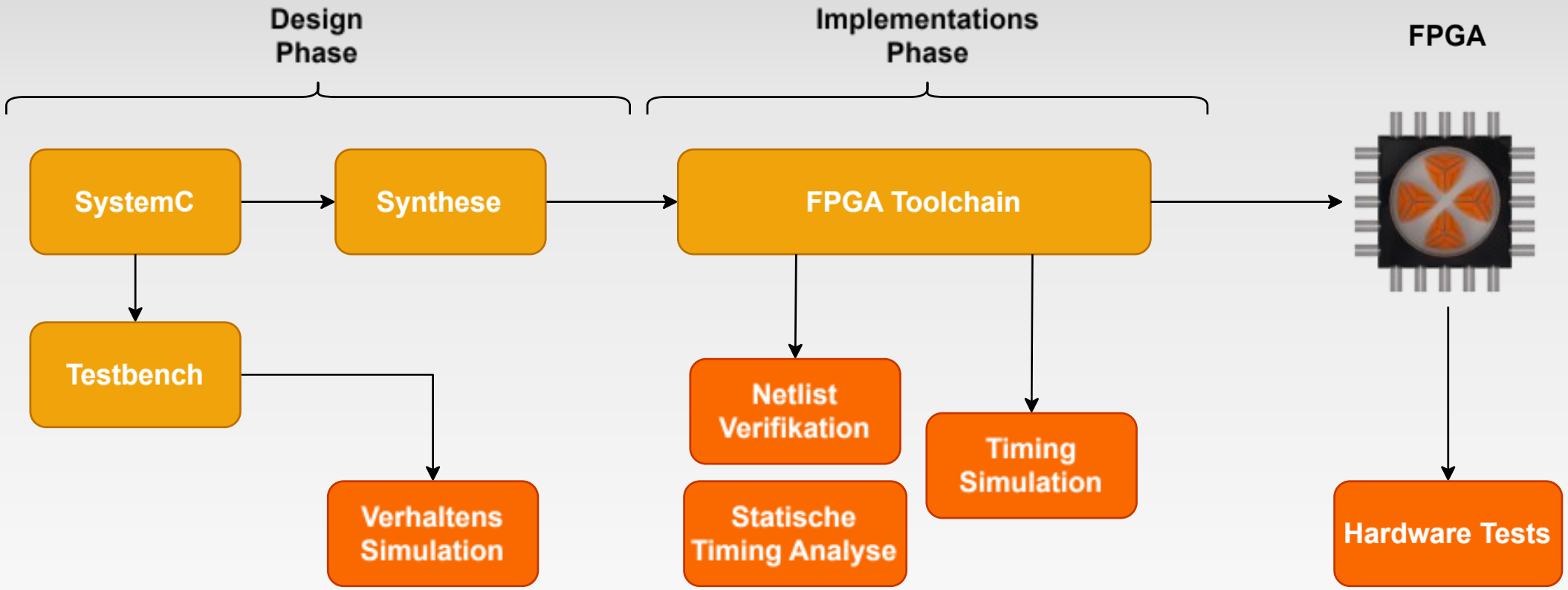
Workflow mit SystemC

- Design in SystemC auf System-, Architektur- und Registerebene möglich
- Testbench und Software als Stimulator möglich
- GTKWave Analyse auch hier möglich



Testbench

Workflow mit SystemC



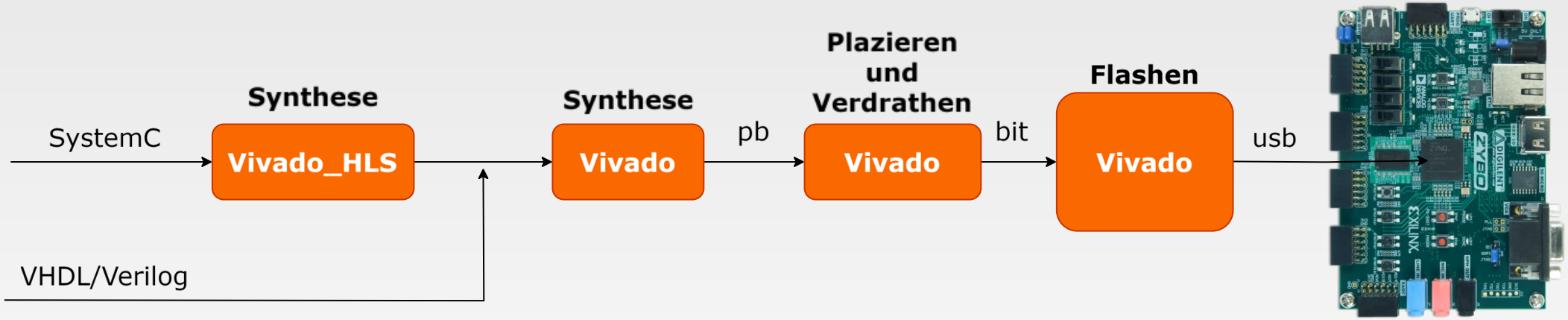
Vorteile

- Software und Hardware in der gleichen Sprache
- Performantere Simulation möglich
- Hardware und Simulation aus einem Source-Code
- Bekannte C++-Syntax & Hardware-Beschreibung
- Taktgenauer Code kann synthetisiert werden

Xilinx Vivado

- Design Software von Xilinx
- Propritäre Software
- Minimale Variante ist kostenlos
- Grafische IDE verfügbar

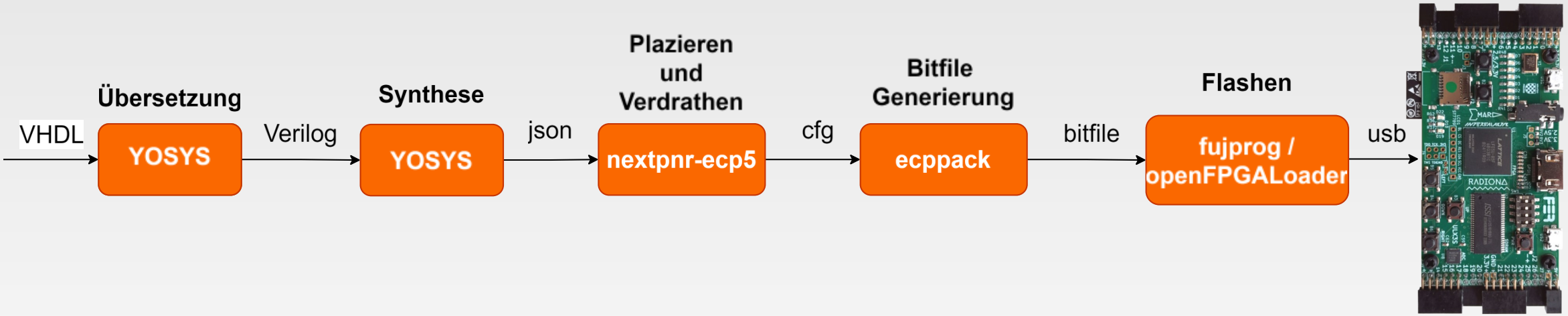
Synthese mit Vivado



OSS-CAD-Suite

- Sammlung von Open-Source Tools für:
  - Simulation
  - Synthese
  - Platzieren und Verdrathen (ecp5, ice40, nexus, gowin…)
  - FPGA Tools (ULX3S, Orangecrab, TinyFPGA, …)
- Meist keine Grafischen Tools

Synthese mit OSS-CAD-Suite



Gliederung

FPGA-Hardwareentwicklung
mit Open-Source-Tools

1. Was sind FPGAs?
2. Entwicklungsboards
3. Open-Source-Tools für Hardware Design
4. SystemC und ICSC
5. Beispiel: ParaNut
6. Ausblick: PicoNut
7. Aktuell: THANNA
8. OSS in der Lehre
9. Fazit

Umstellen der Toolchain

Vivado HLS

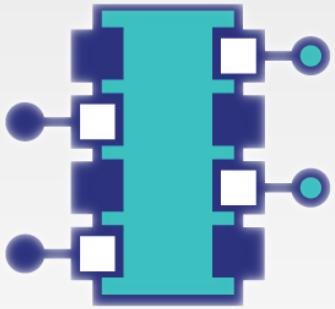
ICSC
intel

ICSC

Intel Compiler for SystemC (ICSC)

- ICSC ist Teil der Chips Alliance
- CA gehört zur Linux Foundation
- Synthetisiert SystemC zu System-Verilog
- Ermöglicht Weiterverarbeitung mit gängigen FPGA-Tools

ICSC
intel



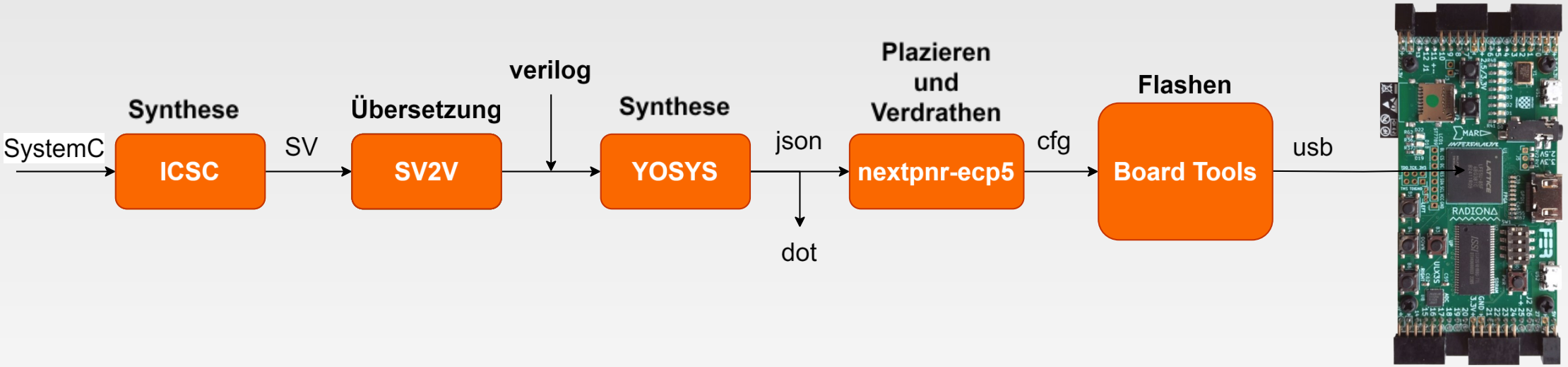
CHIPS ALLIANCE

LINUX FOUNDATION

ICSC

- Benötigt Einhaltung von Designregeln
- Einhaltung des Synthesizable Subsets
- Benötigt Clang & LLVM zum Synthetisieren
- Wird aktiv weiterentwickelt

Toolchain

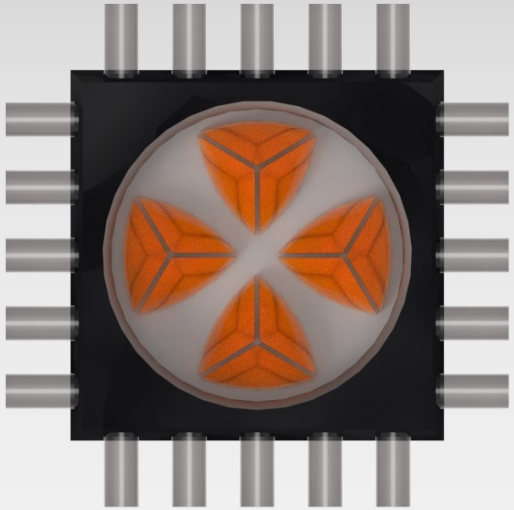

Synthese
Übersetzung
verilog
Synthese
Plazieren
und
Verdrathen
Flashen
SystemC
ICSC
SV
SV2V
YOSYS
json
dot
nextpnr-ecp5
cfg
Board Tools
usb

Gliederung

FPGA-Hardwareentwicklung
mit Open-Source-Tools

1. Was sind FPGAs?
2. Entwicklungsboards
3. Open-Source-Tools für Hardware Design
4. SystemC und ICSC
5. Beispiel: ParaNut
6. Ausblick: PicoNut
7. Aktuell: THANNA
8. OSS in der Lehre
9. Fazit

Beispiel: ParaNut



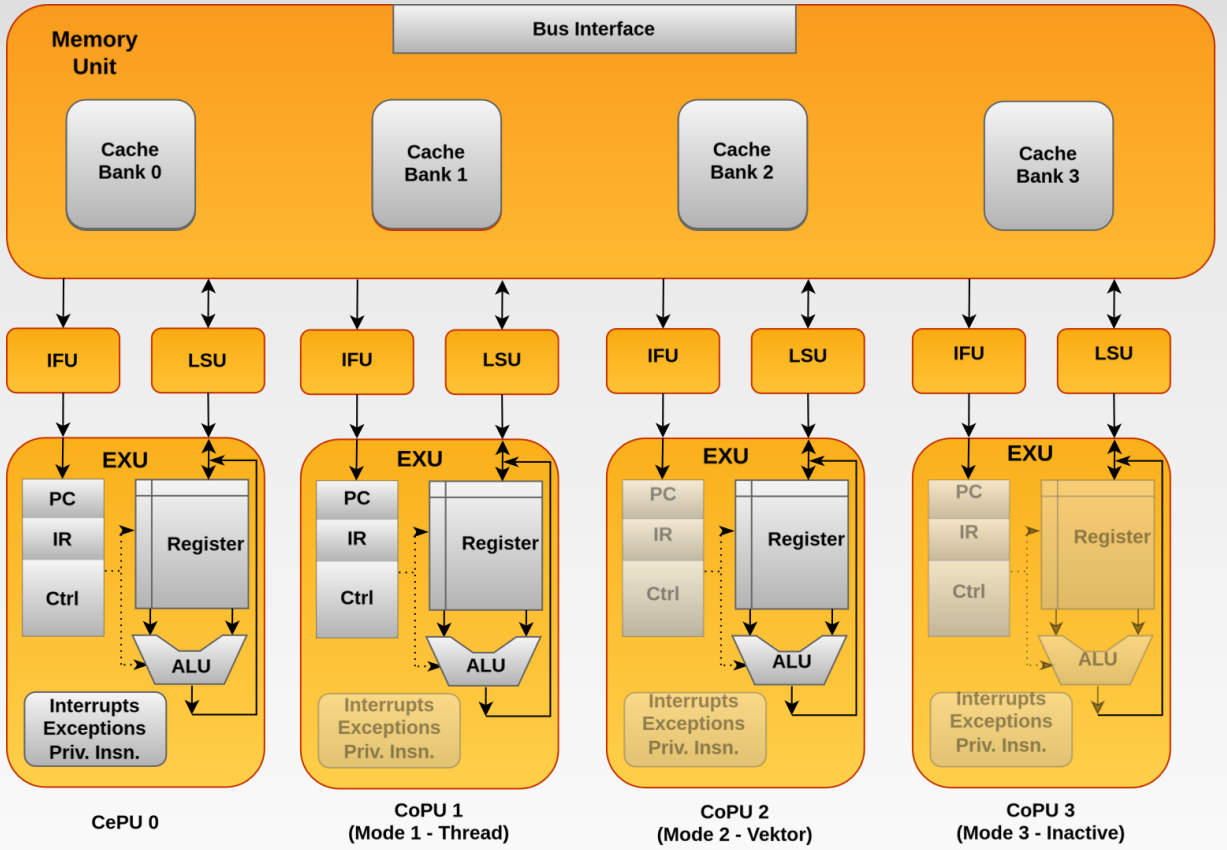
- In SystemC modellierter RISC-V-Prozessor
- Entwickelt an der THA
- Besitzt spezielles Parallelitäts-Konzept
- Wird bereits seit vielen Jahren weiterentwickelt.

Was ist RISC-V ?

- Offene Befehlssatzarchitektur
- Entwickelt an der UC Berkley
- Weiterentwicklung von der RISC-V Foundation

- Mehrere Varianten:
  - RV32/64/128I oder RV32E
- Mehrere Erweiterungen:
  - **M**ultiplikation, **F**loating-Point, ...

Das System


Memory
Unit
Bus Interface
Cache
Bank 0
Cache
Bank 1
Cache
Bank 2
Cache
Bank 3
IFU
LSU
IFU
LSU
IFU
LSU
IFU
LSU
EXU
PC
IR
Ctrl
Register
ALU
Interrupts
Exceptions
Priv. Insn.
EXU
PC
IR
Ctrl
Register
ALU
Interrupts
Exceptions
Priv. Insn.
EXU
PC
IR
Ctrl
Register
ALU
Interrupts
Exceptions
Priv. Insn.
EXU
PC
IR
Ctrl
Register
ALU
Interrupts
Exceptions
Priv. Insn.
CePU 0
CoPU 1
(Mode 1 - Thread)
CoPU 2
(Mode 2 - Vektor)
CoPU 3
(Mode 3 - Inactive)

Parallelität mit libparanut

- Zwei Modi der Parallelität
  - SIMD-Vektor
  - Multithreading
- Sind durch die libparanut in C verwendbar

Beispiel: Linked-Mode

```
int n, a[4], b[4], w[4], wsum[4];

for (n = 0; n < 4; n++)
  wsum[n] = a[n] * w[n] + b[n] * (100 - w[n]);
```

```
n = PN_BEGIN_LINKED (4); /* returns core id (0..3) */
wsum[n]=a[n] * w[n] + b[n] * (100 - w[n]);
PN_END_LINKED ();        /* switch back to single-thread
```

Beispiel: Threaded-Mode

Technische Hochschule Augsburg

THA

```
int n, a[4], b[4], w[4], wsum[4];
int threshold = get_threshold();

n = PN_BEGIN_THREADED (4);  /* returns core id (0..3) */
wsum[n] = a[n] * w[n] + b[n] * (100 - w[n]);t
if (wsum[n] > threshold) ...
PN_END_THREADED ();          /* switch back to single-thread mode */
```

Herausforderungen

- Anpassen von bestehendem SystemC-Code
  -> Design-Guide für Synthesizable-Subset
- Bestehende Designfehler
  -> Bessere Fehlermeldungen
- Kinderkrankheiten des ICSC
  -> Guter Austausch mit Maintainer möglich

Gliederung

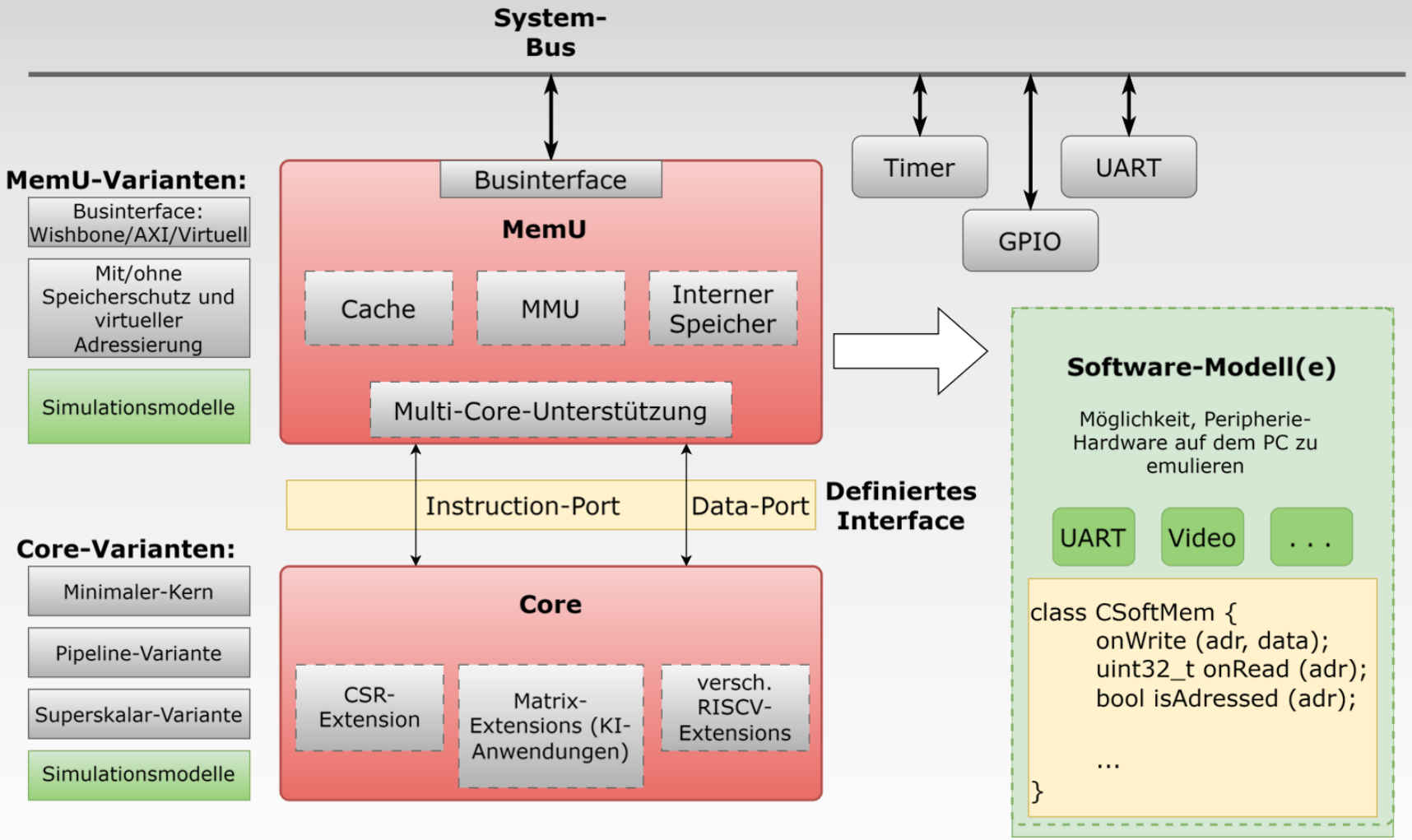
FPGA-Hardwareentwicklung
mit Open-Source-Tools

1. Was sind FPGAs?
2. Entwicklungsboards
3. Open-Source-Tools für Hardware Design
4. SystemC und ICSC
5. Beispiel: ParaNut
6. Ausblick: PicoNut
7. Aktuell: THANNA
8. OSS in der Lehre
9. Fazit

Ausblick: PicoNut

- Minimaler RISC-V Prozessor (Open-Source)
- Modularität und Erweiterbarkeit
- Vielvältige Simulationsmöglichkeiten
- Hauptsprache: SystemC
- Design für Linuxfähigkeit
- Projektbegleitendes CI/CD mit Jenkins
- Projekt von Studierenden für Studierende

PicoNut Konzept



Gliederung

FPGA-Hardwareentwicklung
mit Open-Source-Tools

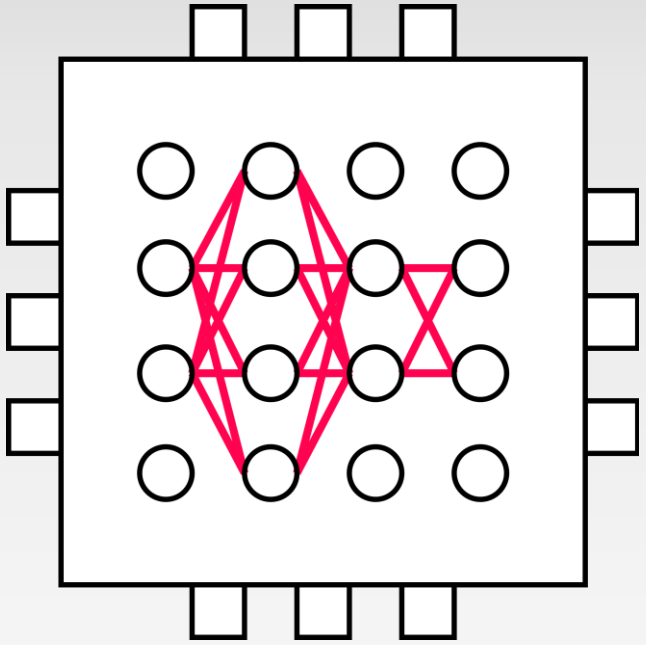
1. Was sind FPGAs?
2. Entwicklungsboards
3. Open-Source-Tools für Hardware Design
4. SystemC und ICSC
5. Beispiel: ParaNut
6. Ausblick: PicoNut
7. Aktuell: THANNA
8. OSS in der Lehre
9. Fazit

Aktuell: THANNA

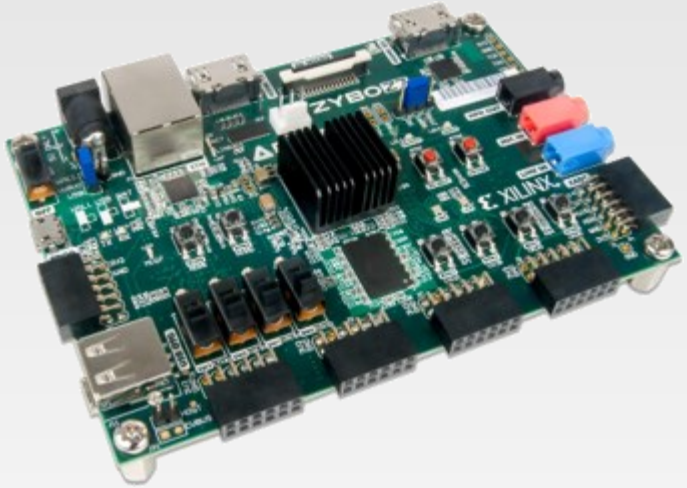
- Ziele:
  - Optimierte Neuronale Netze
  - Effiziente Hardware für KI-Anwendungen
  - Optimale Anpassung an gegebene Neuronale Netze
  - Einfache Benutzung

Aktuell: THANNA System

THANNA Quantizer



THANNA Prozessor



PetaLinux
THANNA Treiber
THANNA Library

Gliederung

FPGA-Hardwareentwicklung
mit Open-Source-Tools

Open-Source-Hardware und -Software in der Lehre

Technische Hochschule Augsburg

THA

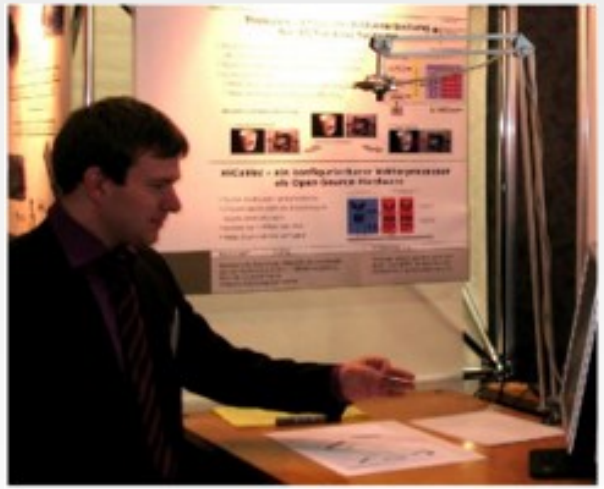
Ziel: Erlernen der Design-Schritte mit "Hands-on" im Praktikum

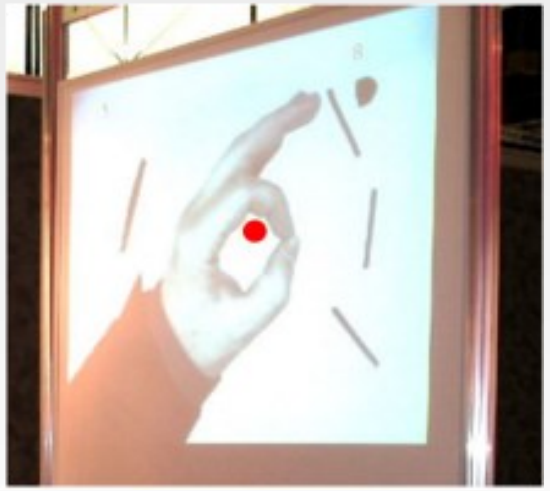
- Simulation
- Synthese
- Platzierung und Verdrahtung
- Programmierung / Inbetriebnahme der Hardware

Beispiel: ASTERICS-Praktikum

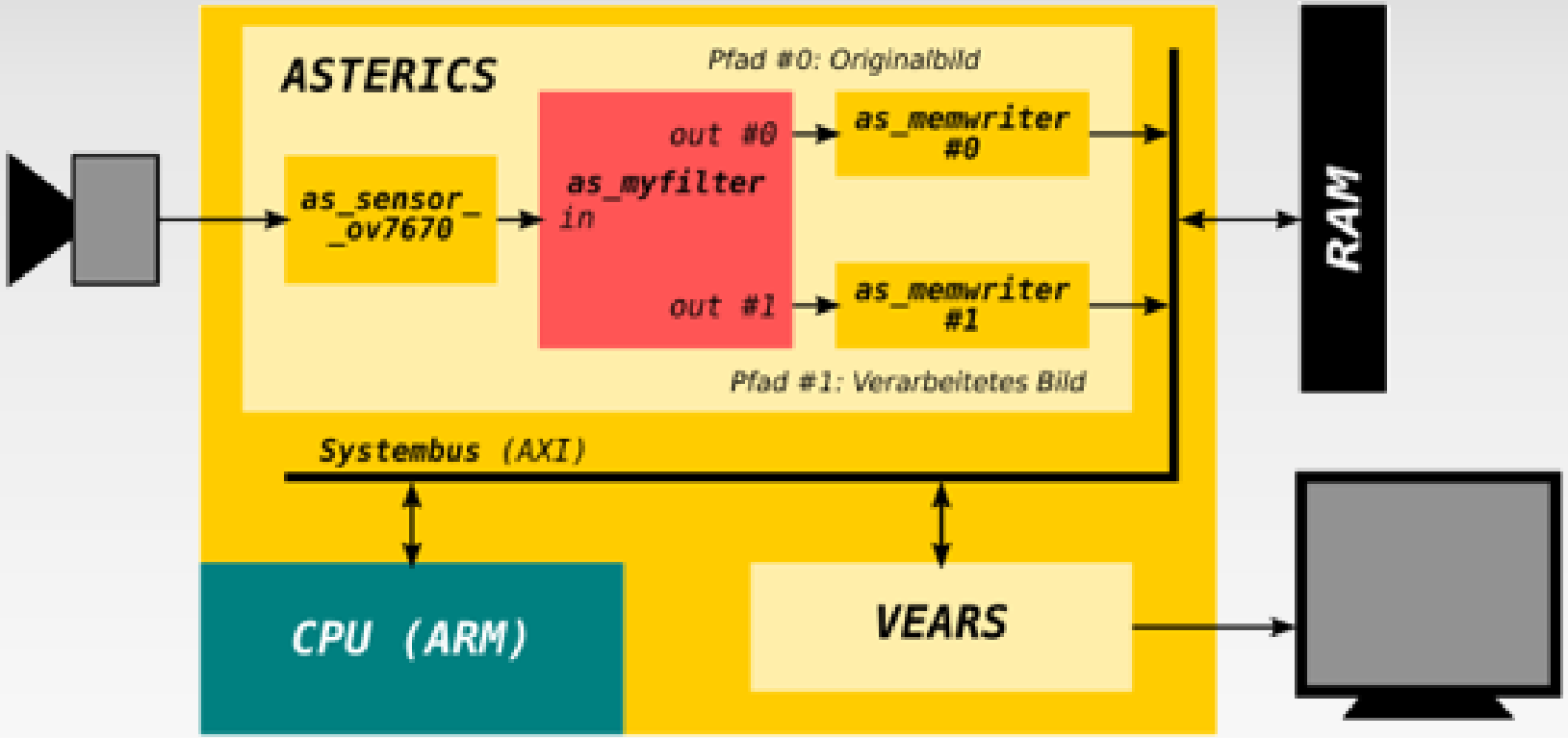
Technische Hochschule Augsburg

- Studenten entwickeln ein System-on-Chip, das mittels Computer-Vision Kanten in einem Kamerabild erkennt und einen Ball als Overlay darstellt, der von den Kanten reflektiert wird.



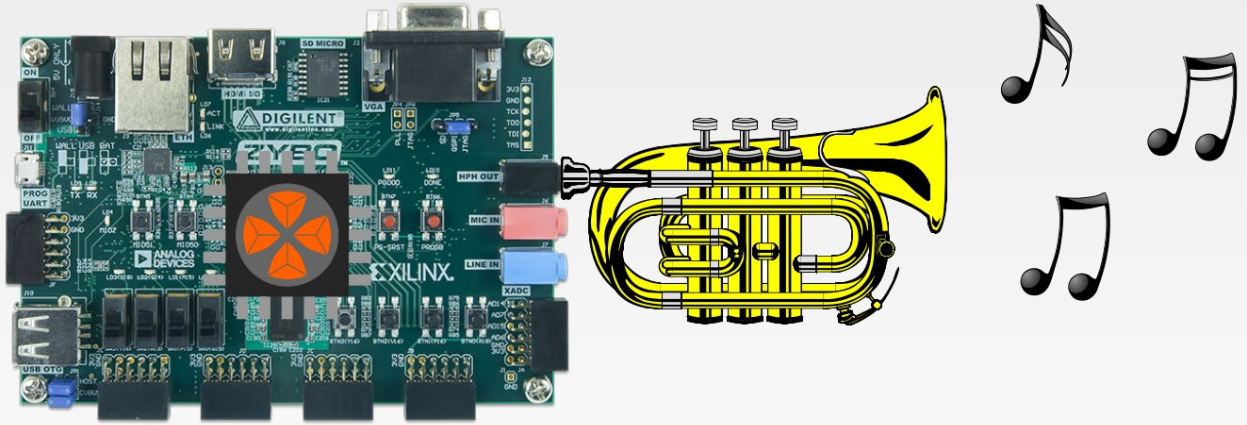


Beispiel: ASTERICS-System

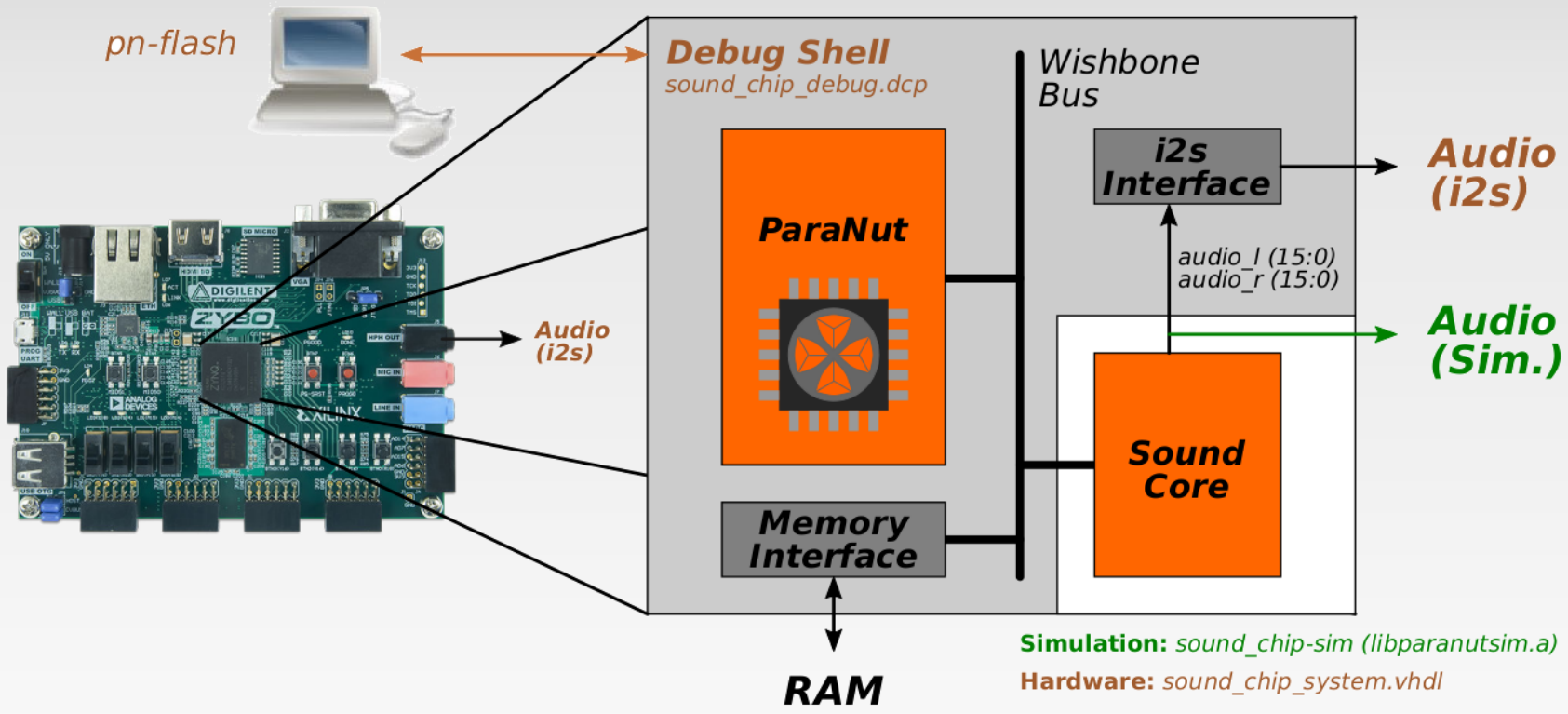


Beispiel: ParaNut-Praktikum

- Studenten bauen ein System-on-Chip mit einem ParaNut-Prozessor, das mit einem selbst entwickelten Tongenerator Musik abspielt.



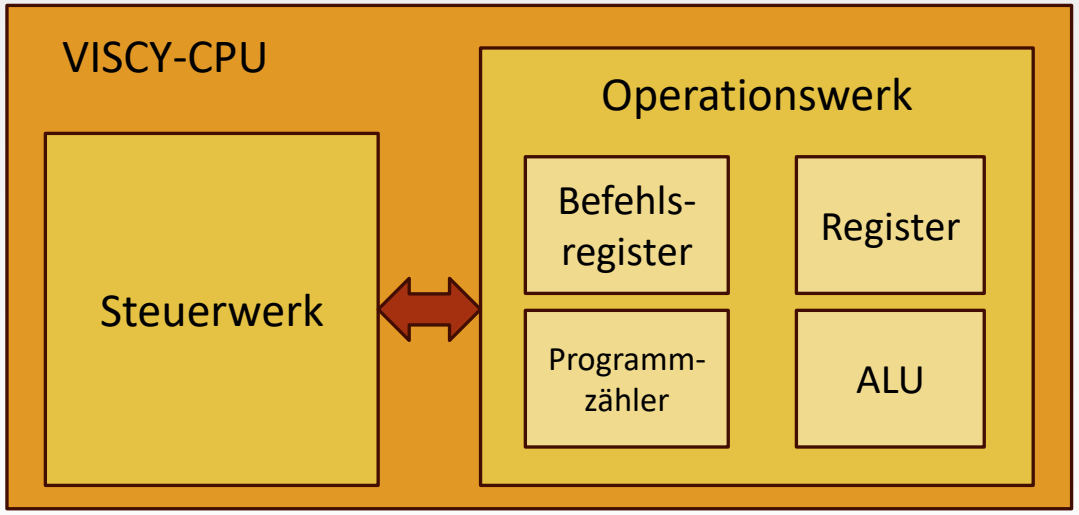
Beispiel: ParaNut-System



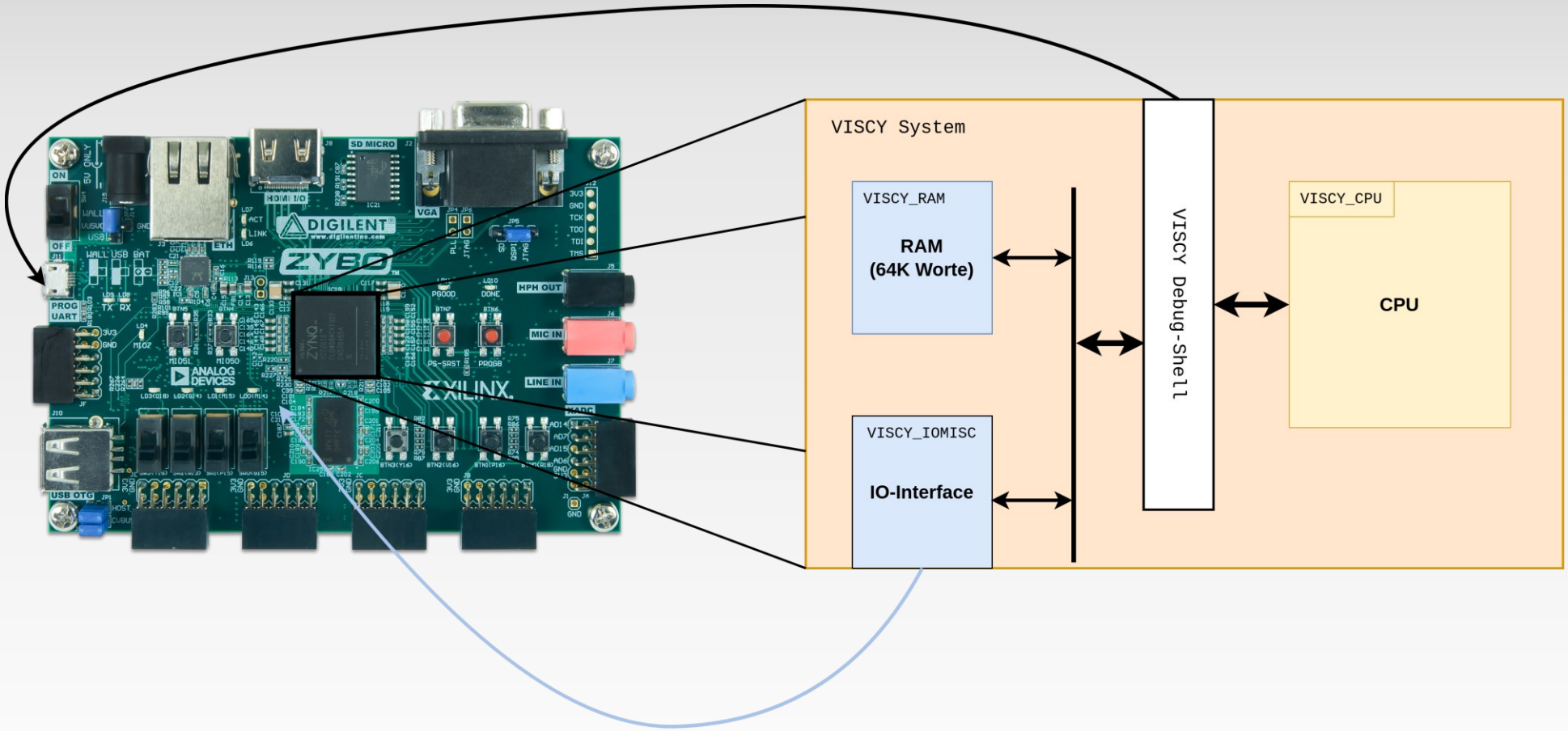
Beispiel: VISCY

Der VISCY-Prozessor …

- ist eine kleine und einfache CPU.
- wird von Studenten innerhalb von einem Semester gebaut und programmiert.



Aktuell: Praktikumsaufbau

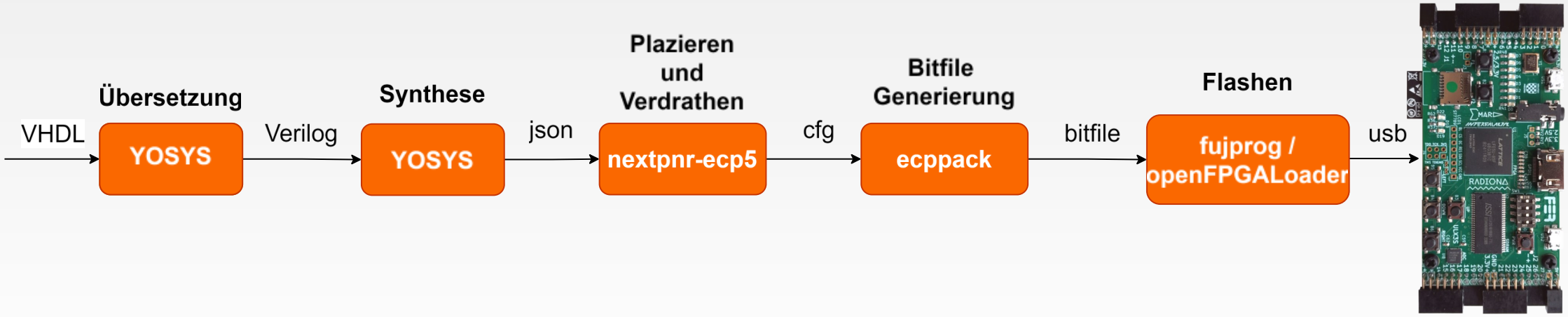


Aktuell verwendete Tools

- Entwurf und Simulation der CPU-Teilkomponenten bereits mit OSS-Tools:
  - GHDL & GTKWave
- Synthese der CPU und Einbau in das System (Implementierung):
  - Xilinx Vivado (Synthese, PnR, BitGen) -> Wrapper-Skripte zur Vereinfachung
- Inbetriebnahme auf dem Zybo-Board (Digilent)

Geplant: VISCY-V und ULX3S

- Der VISCY wird RISC-V-kompatibel
  - OSS-Toolchain für Entwurf der CPU ***und*** Implementierung:
  - GHDL, GTKWave und weitere Tools aus der OSS-CAD-Suite
  - Inbetriebnahme auf dem ULX3S-Board



Gliederung

FPGA-Hardwareentwicklung
mit Open-Source-Tools

1. Was sind FPGAs?
2. Entwicklungsboards
3. Open-Source-Tools für Hardware Design
4. SystemC und ICSC
5. Beispiel: ParaNut
6. Ausblick: PicoNut
7. Aktuell: THANNA
8. OSS in der Lehre
9. Fazit

Vorteile der OSS-Toolchain

- Simulation
  - GHDL, SystemC und GTKWave seit vielen Jahren erfolgreich im Einsatz
- Synthese
  - Toolchain (OSS-CAD-Suite für Lattice) ist komplett quelloffen
  - wesentlich kleiner als Xilinx-Vivado: 2GB << 45GB
  - Synthese-Zeiten sind wesentlich kürzer
- Allgemein
  - Einstiegshürden werden gesenkt:
    - Arbeitsumgebung leicht einrichtbar und verwendbar
  - Erschwingliche Boards sind verfügbar

Herausforderungen bei OSS-Tools

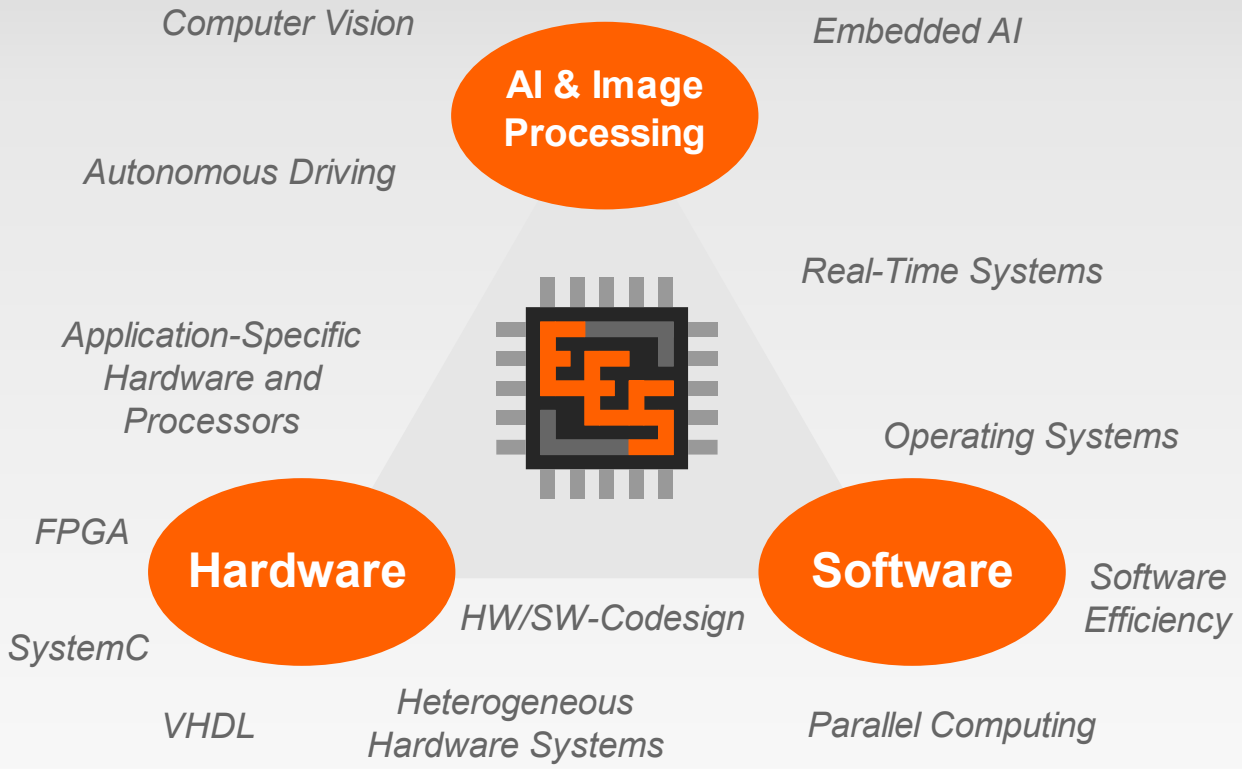
Technische Hochschule Augsburg

THA

- Teils kryptische Fehlermeldungen bei YOSYS
  - Mit ICSC dagegen sehr gute Erfahrungen gemacht
- Integration in IDE (Feedback etc.)
- Grafische Darstellung von Informationen für Fehleranalyse bei der Synthese
  - Schematic-Viewer mit Auflösung von Hierarchien
  - Timing-Analyse: welcher ist der kritische Pfad, wie Verlauf?

Bei Interesse…

ees.tha.de



PicoNut