

Computer Engineering

Prof. Dr. Jochen Rust
Prof. Dr. Michael Schäfers
Prof. Dr. Tim Tiedemann

Agenda

- Vorstellung
- Organisatorisches
- Motivation, Grundlagen des „Computer Engineering“
- Grundlagen Digitalschaltungen

PAUSE

- Einführung VHDL

0. Vorstellung

Vorstellung

Werdegang

2023 – heute
2023 – heute

HAW Hamburg
DSI Aerospace GmbH

Professor Technische Informatik
Deputy Head of Pre-Development

2020 – 2023
2014 – 2020
2008 – 2014
2006 – 2007
2001 – 2008

DSI Aerospace Technologie GmbH
ITEM, Universität Bremen
ITEM, Universität Bremen
IBM Deutschland Entwicklung GmbH
Universität Hannover

Head of Studies
Senior Research Group Leader
Research Associate
Masterarbeit, Internship
Studium Technische Informatik



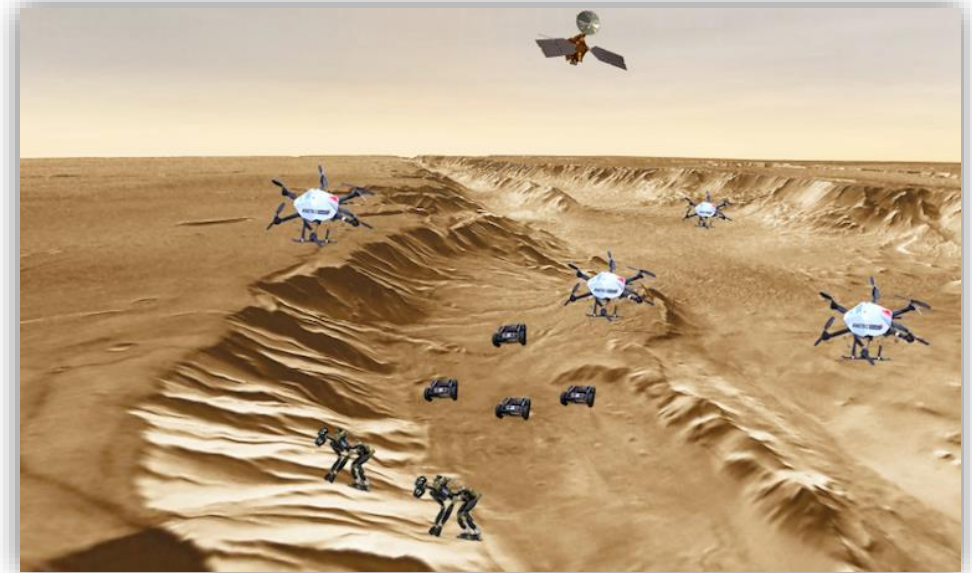
Jochen Rust
Prof. Dr.-Ing.

Erfahrungen und Interessen

Industrial Radio/SDR

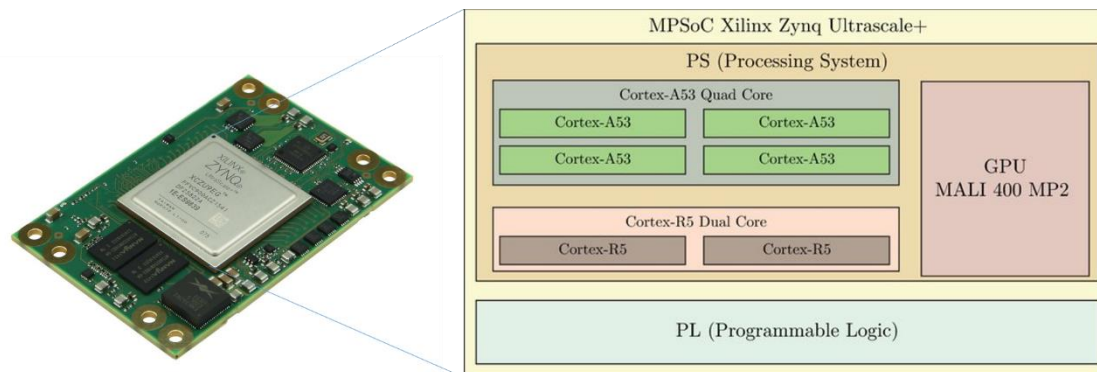


Space Robotics



© Universität Bremen

System-on-Chip Design



© Trenz Electronics

...und Sie?

Was ist für Sie Computer Engineering?

Haben Sie schonmal was von

- VHDL
- Hardwareentwurf
- FPGAs

gehört?

Was erwarten Sie von der Vorlesung?

Computer Engineering

1. Organisatorisches und Einführung



Organisation

Allgemeines

• Vortragsfolien	➡	Präsenz/TEAMs,	PDF in TEAMs (vor der VL)
• Vertiefungen	➡	Tafel, FPGA-board	während der VL
• Implementierungsbeispiele	➡	Labor, FPGA-board	Dateien in TEAMs (nach der VL)
• Kleine „Projekte“	➡	Labor, FPGA-board	Selbständige Bearbeitung im Praktikum

Besondere Daten (Stand jetzt)

- | | | |
|------------------------|---|---------------------|
| • Pfingsten (21.05.24) | ➡ | keine Veranstaltung |
|------------------------|---|---------------------|

Unterlagen sowie aktuelle Informationen finden sich im TEAMs-Raum
„SoSe24_CE“

Einschreibeschlüssel: **ssjr901**

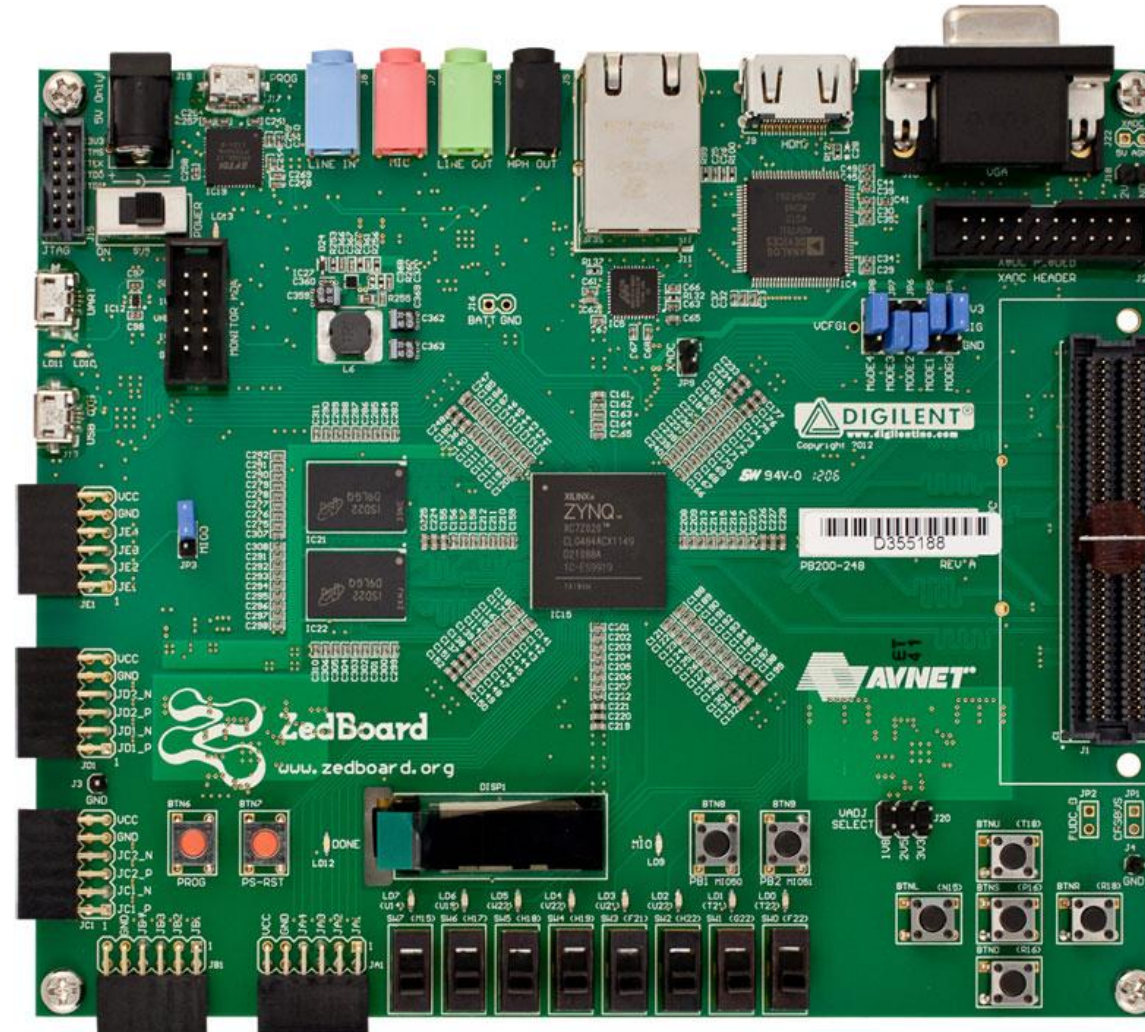
Prüfungsform

- mündlich

Übersicht CE SS24

Semester- woche	Kalender- woche	Woche im Stundenplan	Wochen- beginn	Bemerkungen
0	13	13	25.03.2024	
1	14	14	01.04.2024	Beginn WiSe Prüfungswoche nur INF Ostermontag 01.04.2024
2	15	15	08.04.2024	Grundlagen / VHDL
3	16	16	15.04.2024	
4	17	17	22.04.2024	
5	18	18	29.04.2024	Praktikum
6	19	19	06.05.2024	Grundlagen FPGA Design
7	20	20	13.05.2024	
8	21	21	20.05.2024	MO 20.05. Pfingstmontag am DI *
9	22	22	27.05.2024	Praktikum
10	23	23	03.06.2024	Digitale Signalverarbeitung
11	24	24	10.06.2024	
12	25	25	17.06.2024	Praktikum
13	26	26	24.06.2024	Anwendungen
14	27	27	01.07.2024	Praktikum
15	28	28	08.07.2024	Prüfungswoche nur E+I
16	29	29	15.07.2024	Prüfungswoche
17	30	30	22.07.2024	Prüfungswoche

Zielhardware: Das Avnet ZedBoard



Wo findet „FPGA-Design“ statt?

Hochsprachenebene

- Rechner wird als abstrakte Maschine aufgefasst
- Die konkrete Arbeitsweise ist uninteressant

⇒ Java, C/C++, Python, etc.

Maschinennahe Ebene (Registerniveau)

- Verstehen des Rechners auf Registerniveau
- Typisch: Technische Informatiker, Applikationsingenieure

⇒ Assembler, C (bare-metal)

Gatter und Schaltwerkniveau

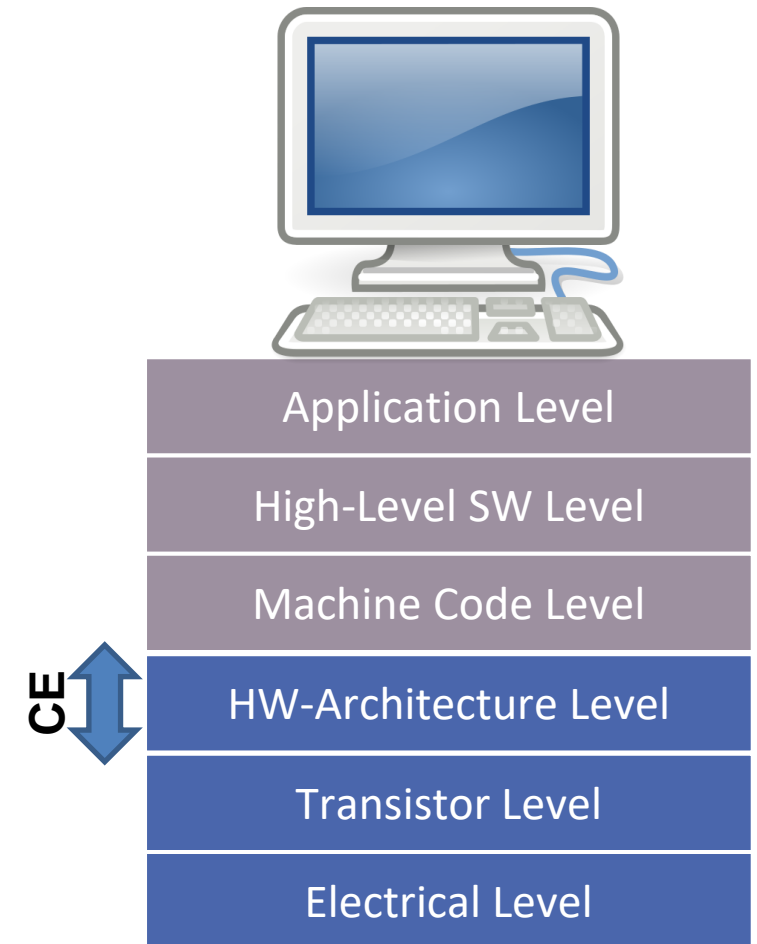
- Verstehen der Prozessorkomponenten
- Typisch: Hardwareentwickler

⇒ Hardware-Beschreibungen, z.B. VHDL, Schaltpläne (Schematics)

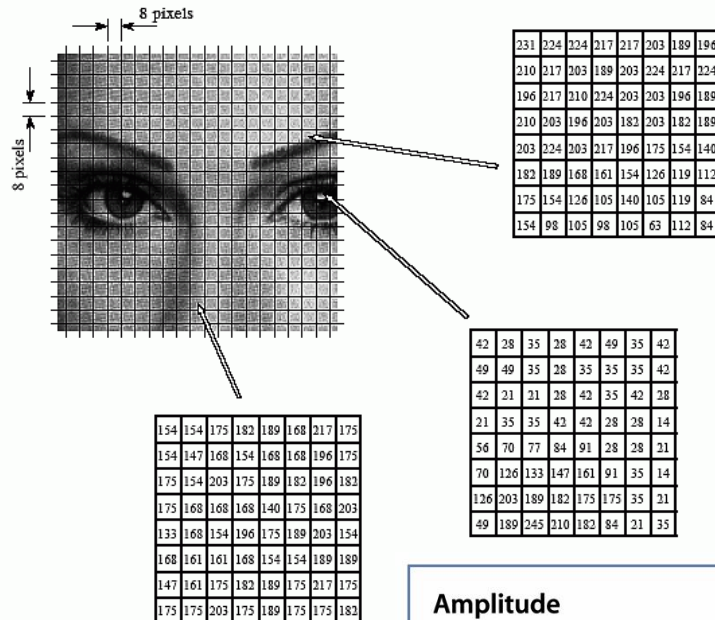
Transistorniveau

- Verstehen der integrierten Schaltkreise
- Typisch: theoretische Elektrotechniker, Physiker

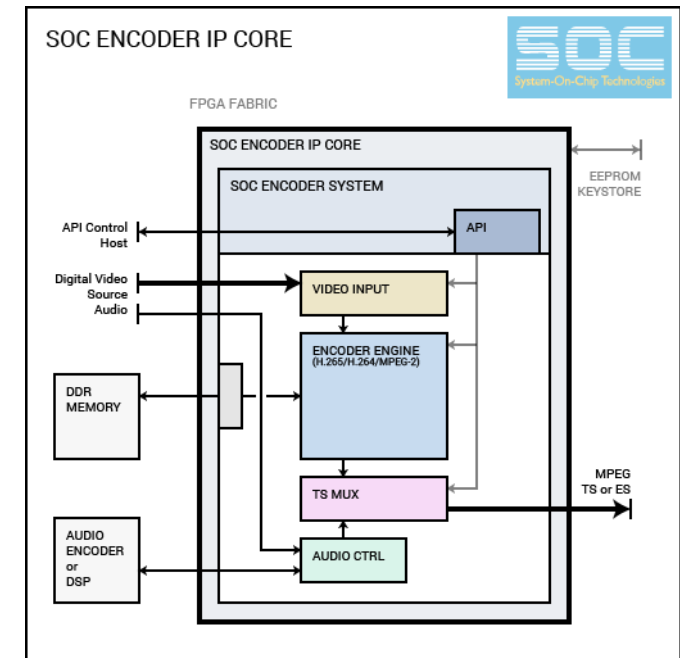
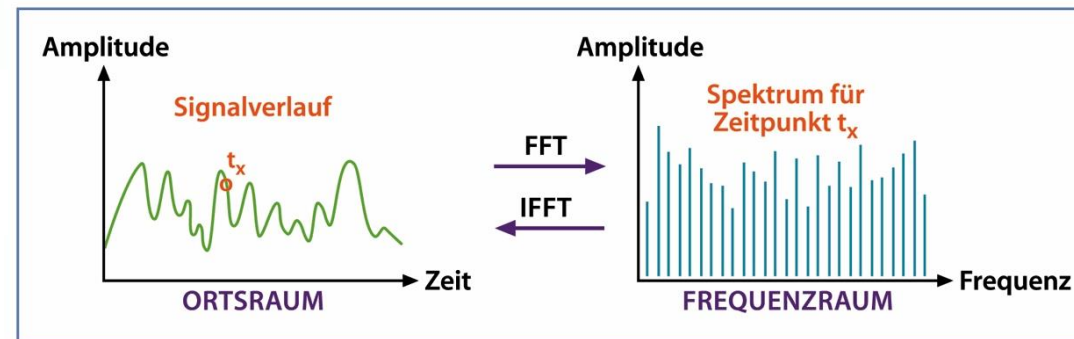
⇒ Layout- und Transistortools: SPICE, Cadence



Anwendungsbeispiel: Image-/Video-/Audio-Encoding



- Kompression des Speicherbedarfs
- Trade-Off: Qualität versus Speicherbedarf
- Beispiel: JPEG Encoding via Fouriertransformation



© soctechnologies.com

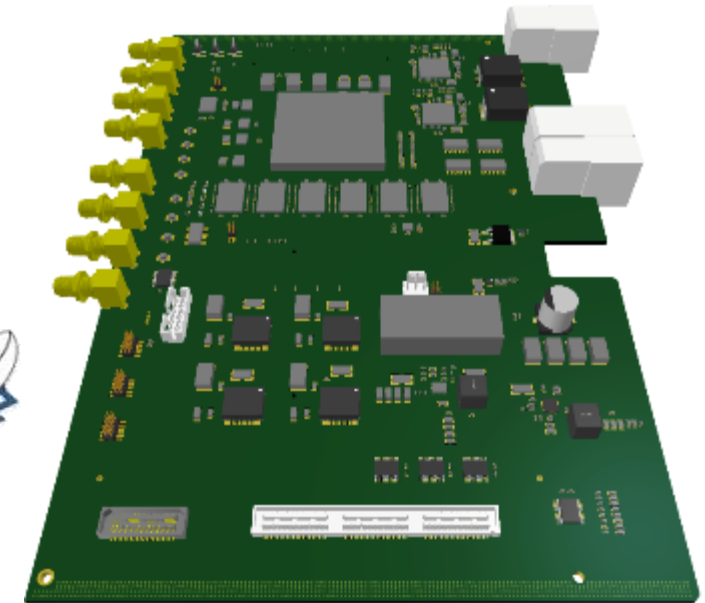
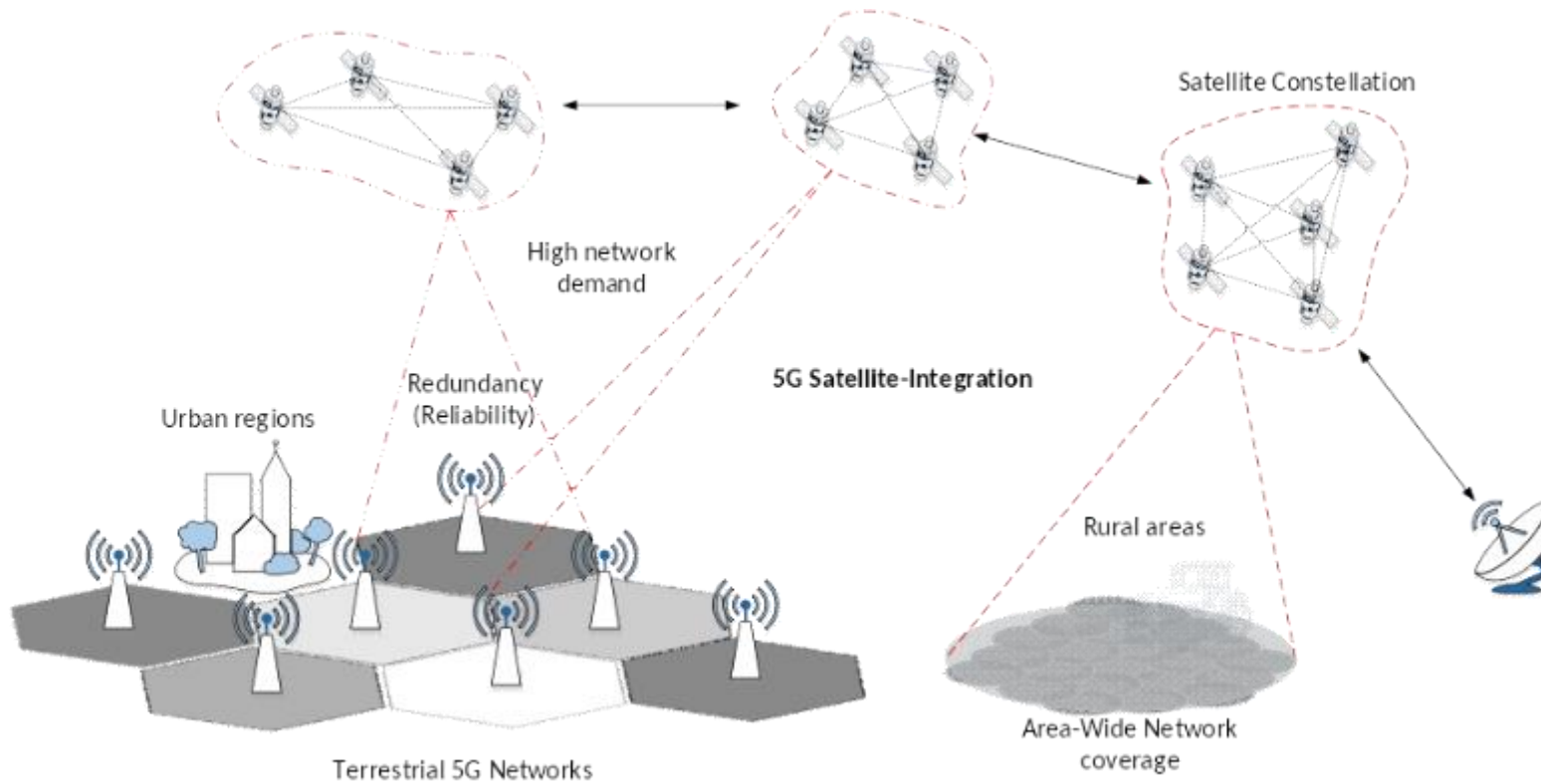
Anwendungsbeispiel: High-End Oszilloscope und Signalgeneratoren

- Erzeugung hochfrequenter Signale
- Sampling hochfrequenter Signale

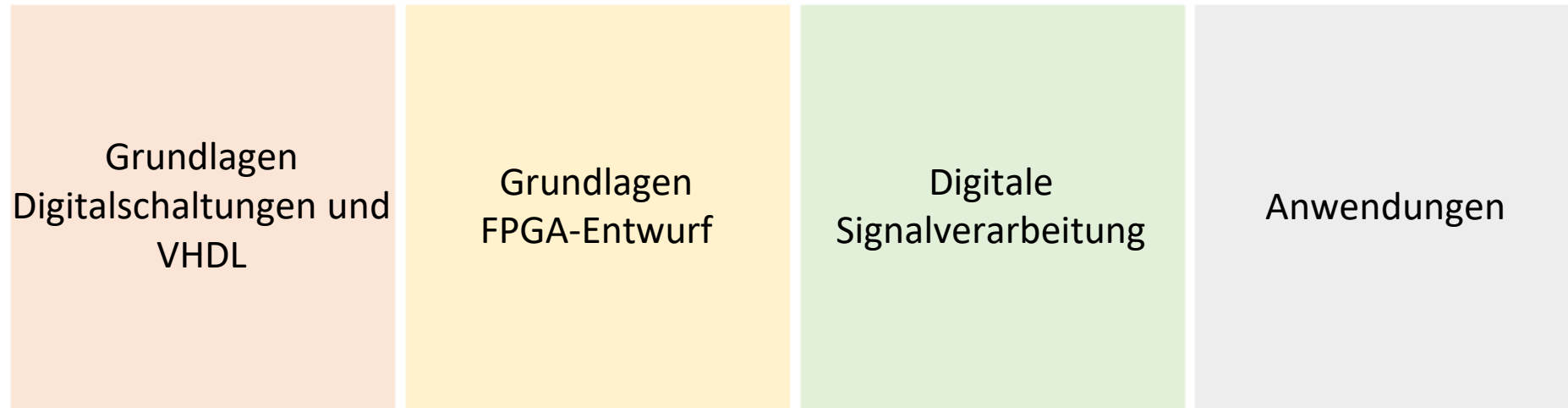


Anwendungsbeispiel: Low-Layer Protokolle

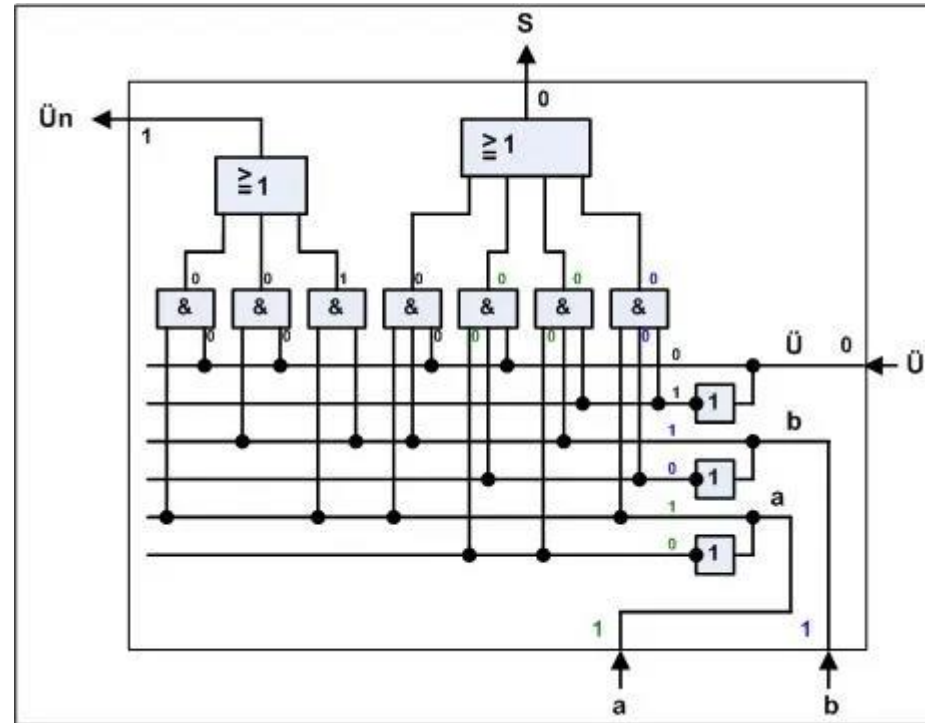
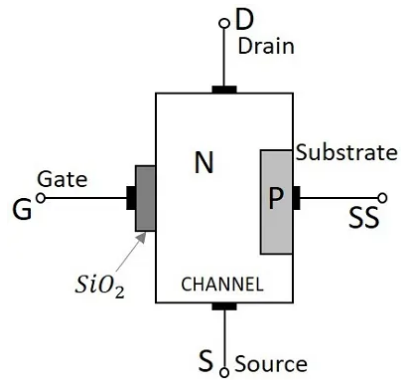
- Beispiel 6G/5G NG Mobile Communications
- Hier: Rad-Tolerant FPGA



Übersicht Vorlesungsinhalte



1. Grundlagen und VHDL



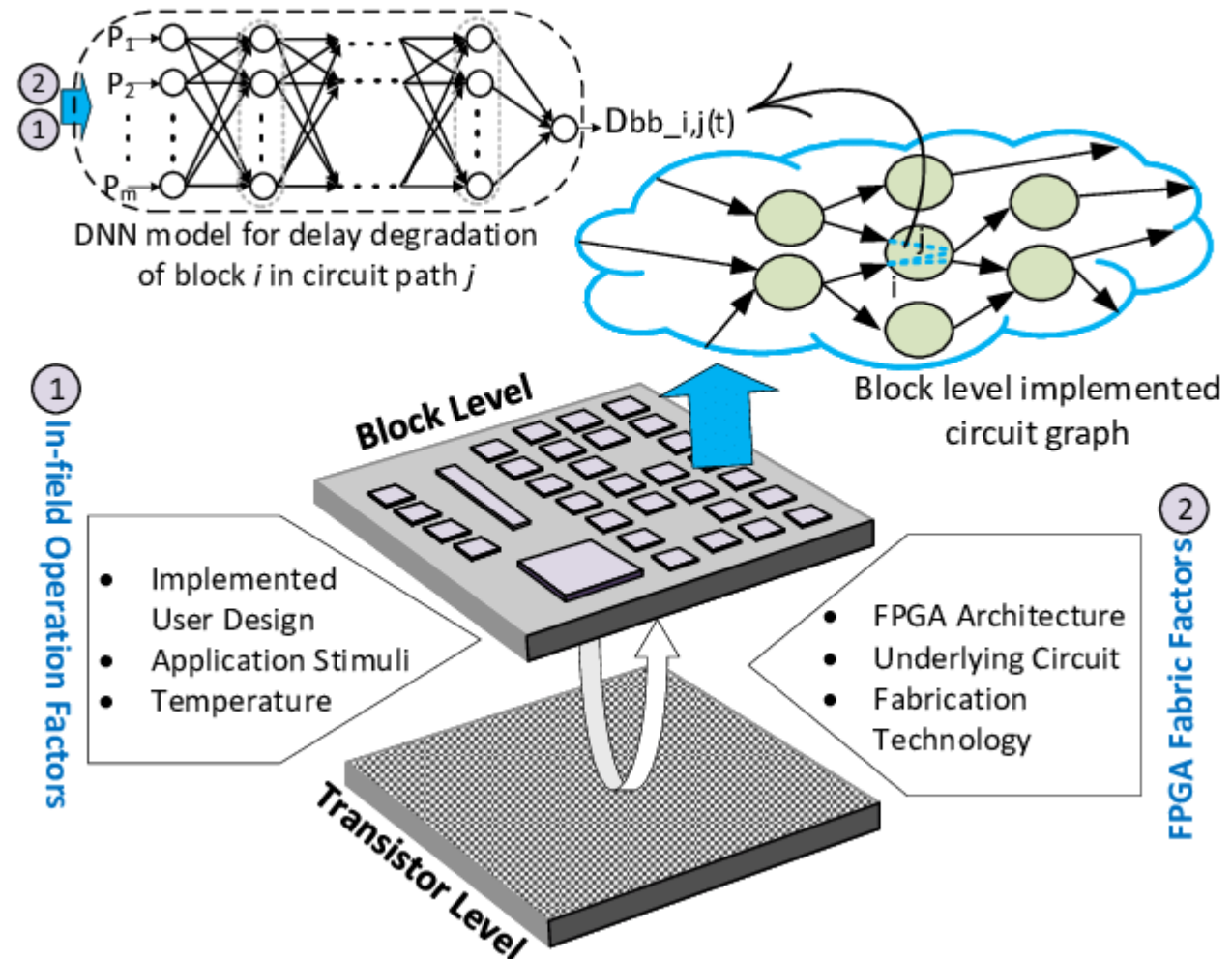
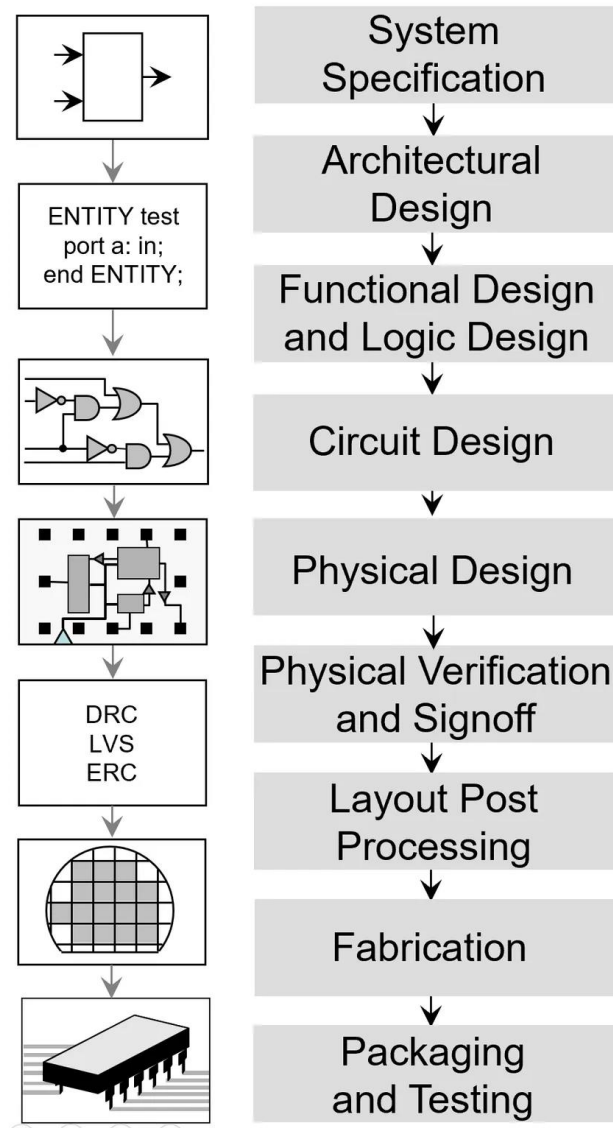
© u-r-rennert.de

```

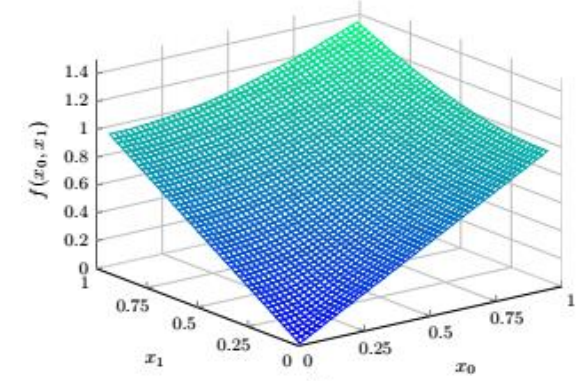
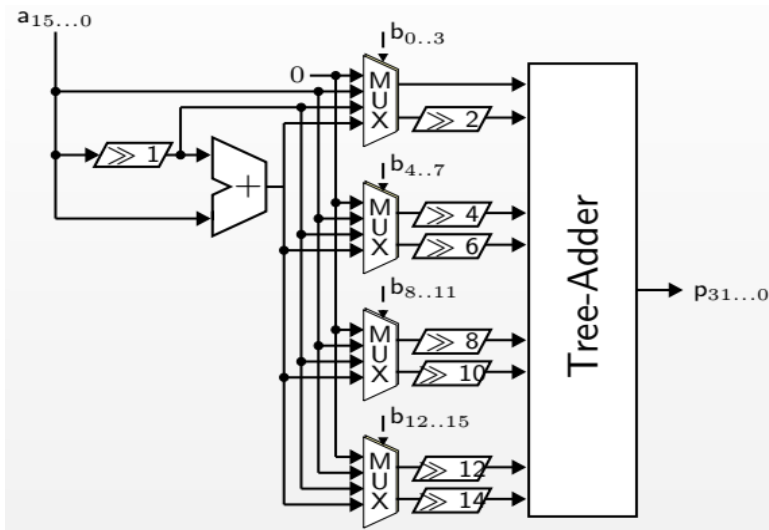
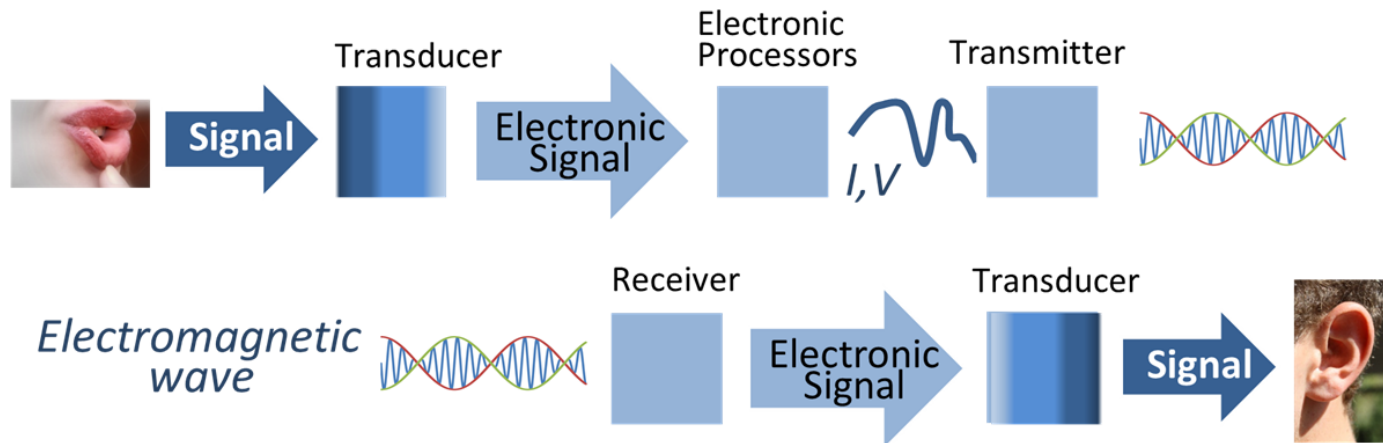
1 library ieee;
2 use ieee.std_logic_1164.all;
3 use ieee.numeric_std.all;
4
5 entity signed_adder is
6   port
7   (
8     aclr : in  std_logic;
9     clk  : in  std_logic;
10    a     : in  std_logic_vector;
11    b     : in  std_logic_vector;
12    q     : out std_logic_vector
13  );
14 end signed_adder;
15
16 architecture signed_adder_arch of signed_adder is
17   signal q_s : signed(a'high+1 downto 0); -- extra bit wide
18
19 begin -- architecture
20   assert(a'length >= b'length)
21     report "Port A must be the longer vector if different sizes!"
22     severity FAILURE;
23   q <= std_logic_vector(q_s);
24
25   adding_proc:
26   process (aclr, clk)
27   begin
28     if (aclr = '1') then
29       q_s <= (others => '0');
30     elsif rising_edge(clk) then
31       q_s <= ('0' & signed(a)) + ('0' & signed(b));
32     end if; -- clk'd
33   end process;
34
35 end signed_adder_arch;

```

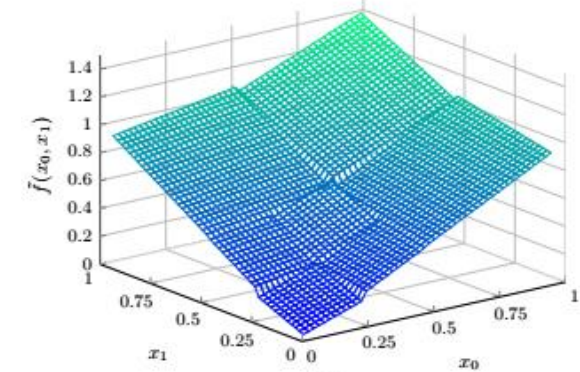

2. Grundlagen FPGA-Entwurf



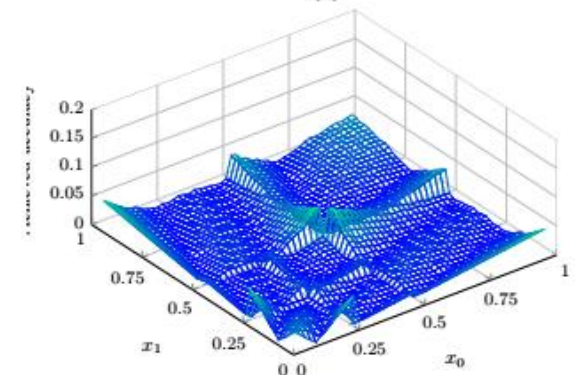
3. Digitale Signalverarbeitung



(a)

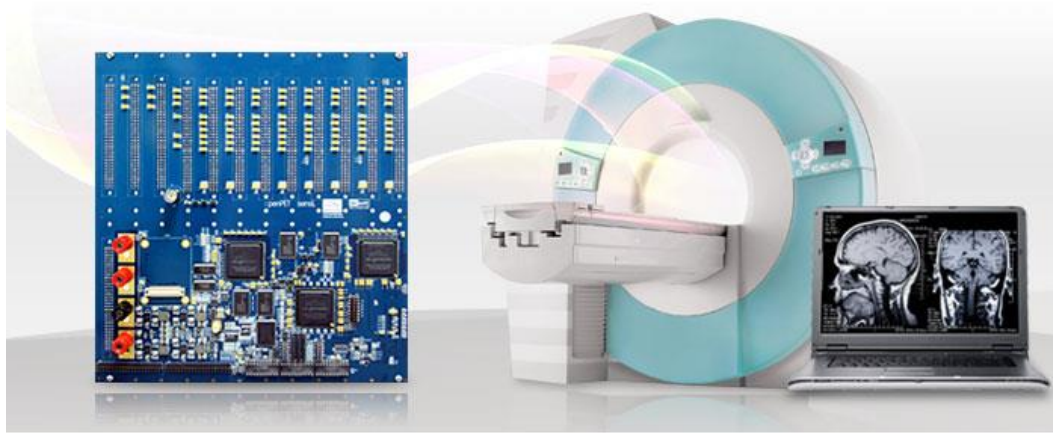


(b)



(c)

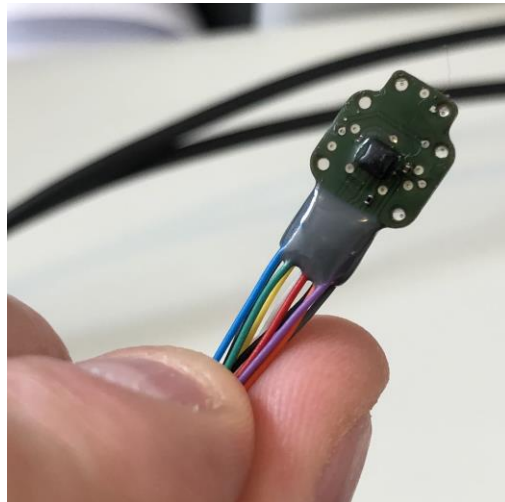
4. Anwendungen



© <https://www.terasic.com.tw/>



© AEON Robotics

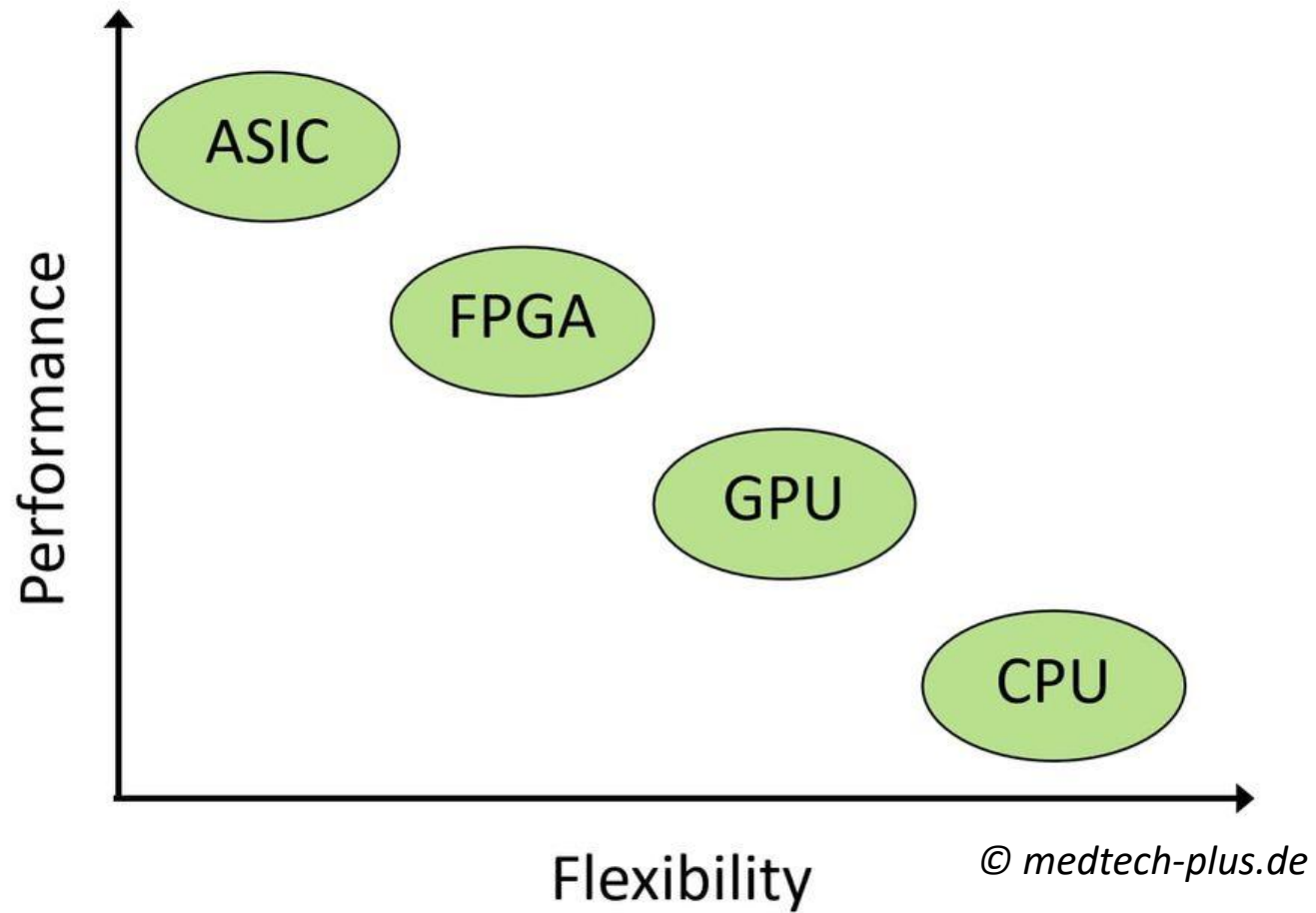


© DSI/HAW/TUHH



© DSI Aerospace GmbH

...in a nutshell



Literatur und Software

Zum Nachlesen:

- H. Kaeslin : „Digital Integrated Circuit Design: From VLSI Architectures to CMOS Fabrication “, Cambridge University Press, 2008.
- G. Lehmann, B. Wunder und M. Selz: „Schaltungsdesign mit VHDL“ online:
https://www.itiv.kit.edu/downloads/Buch_gesamt.pdf
- Universität Hannover, IMS: „Skript zur Vorlesung Electronic Design Automation (EDA)“, URL:
<http://edascript.ims.uni-hannover.de/de/index.html>
- J.-M. Muller: „Elementary Functions: Algorithms and Implementation“, Birkhäuser, 2. Aufl., 2006, ISBN-13: 978-0817643720
- J. Hennessy, D. Patterson: „Computer Architecture: A Quantitative Approach“, Morgan Kaufmann, 2. Aufl., 1996, ISBN: 1-55860-329-8
- J. Hennessy, D. Patterson: „Computer Architecture: A Quantitative Approach“, Morgan Kaufmann, 4. Aufl., 2006, ISBN: 0-12-370490-1
- J. Gustafson: „The End of Error: Unum Computing“ , Chapman & Hall CRC Computational Science, 2015, ISBN-13: 978-1482239867

Literatur und Software

Zum Nachschlagen:

- Online repository für VHDL: <https://tams.informatik.uni-hamburg.de/research/vlsi/vhdl/index.php?content=03-documentation>
- AVNET ZedBoard technical datasheet:
https://www.avnet.com/wps/wcm/connect/onesite/922900e3-3d57-4cc7-883f-a8b9fbea0cd0/ZedBoard_HW_UG_v2_2.pdf?MOD=AJPERES&CACHEID=ROOTWORKSPACE.Z18NA5A1I41L0ICD0ABNDMDDG0000-922900e3-3d57-4cc7-883f-a8b9fbea0cd0-nxyWMFS

FRAGEN

F.1a Wozu müssen ITS-/AI-ler FPGAs kennen und benutzen?

F.1b Was könnten Vor- und Nachteile des FPGA-Entwurfs gegenüber Microcontroller-/Prozessor-Entwurfs.

F.1c Kennen Sie ein Beispiel, in dem Prozessoren „nicht schnell genug“ sind?