Computer Organization and Networks

(INB.06000UF, INB.07001UF)

Welcome

Winter 2022/2023

Stefan Mangard, www.iaik.tugraz.at

Content

What This Course is About

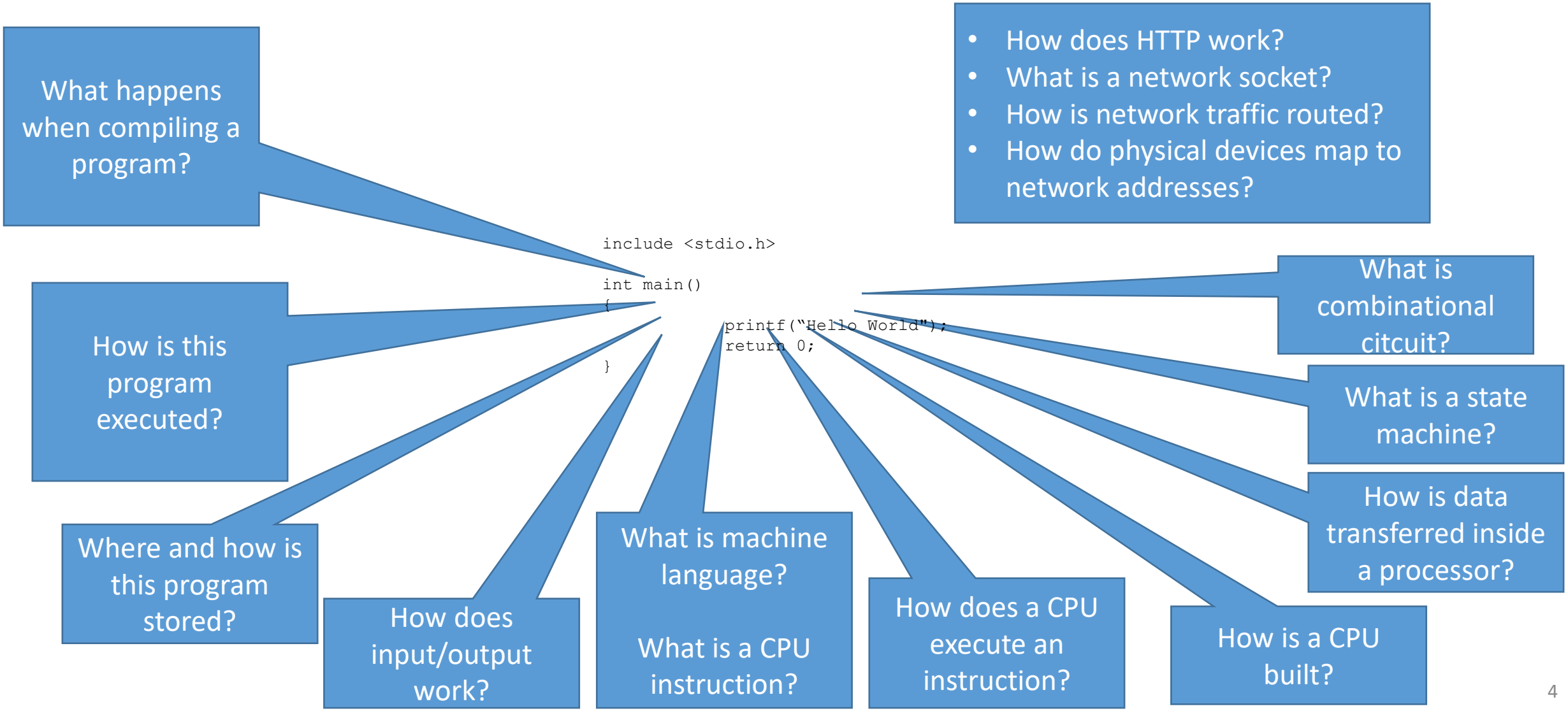
- How does a computer work?
- What does actually happen, if I compile and run this code?

```
include <stdio.h>

int main()
{
            printf(“Hello World");
            return 0;
}
```

- How do computers communicate?

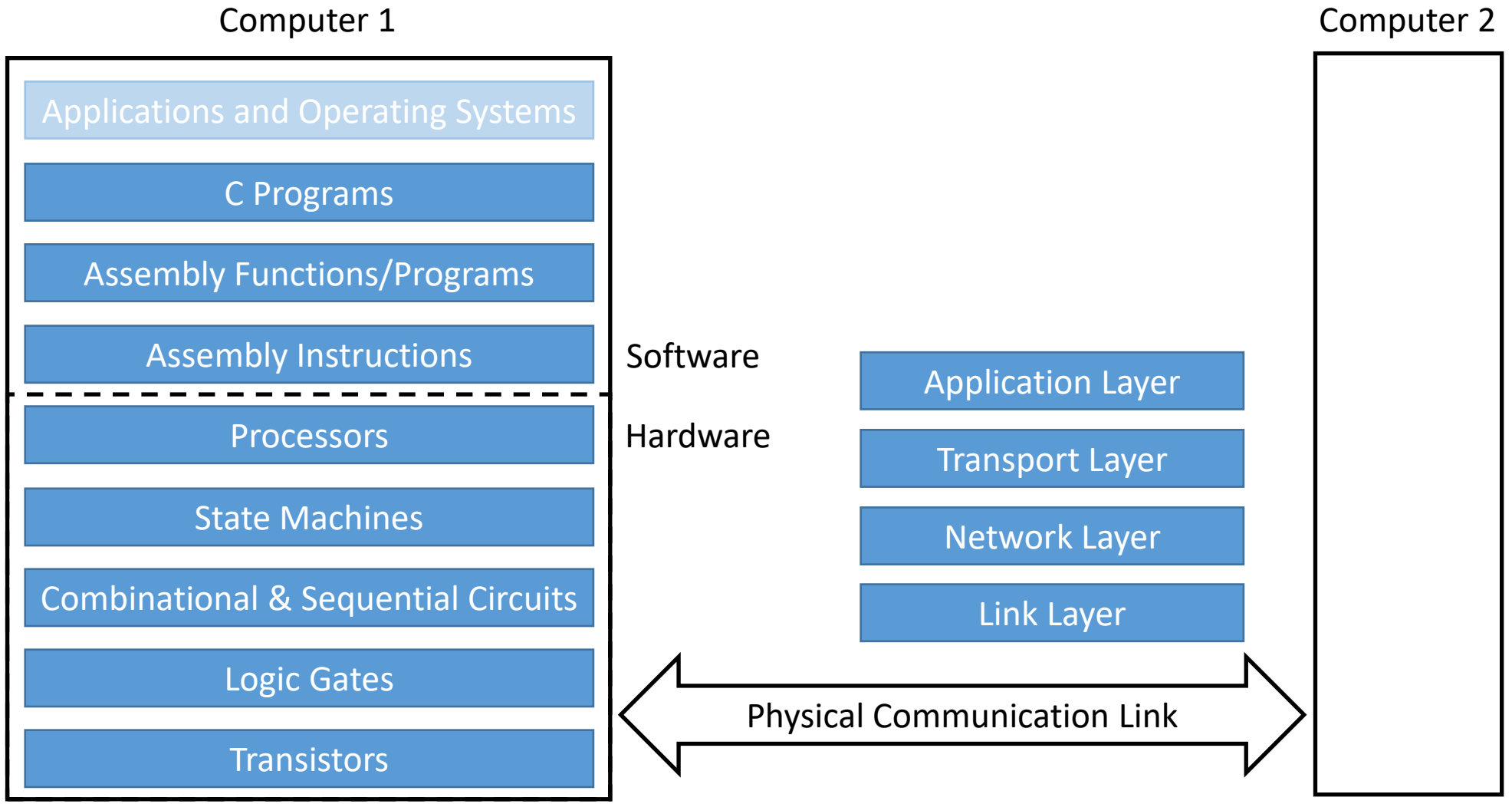
Hardware and Software - It's all one Thing



The Lecture follows a Bottom-up Approach

- Abstraction will be our most important tool
- We “play Lego” and we constantly build larger and more powerful bricks

The Big Picture



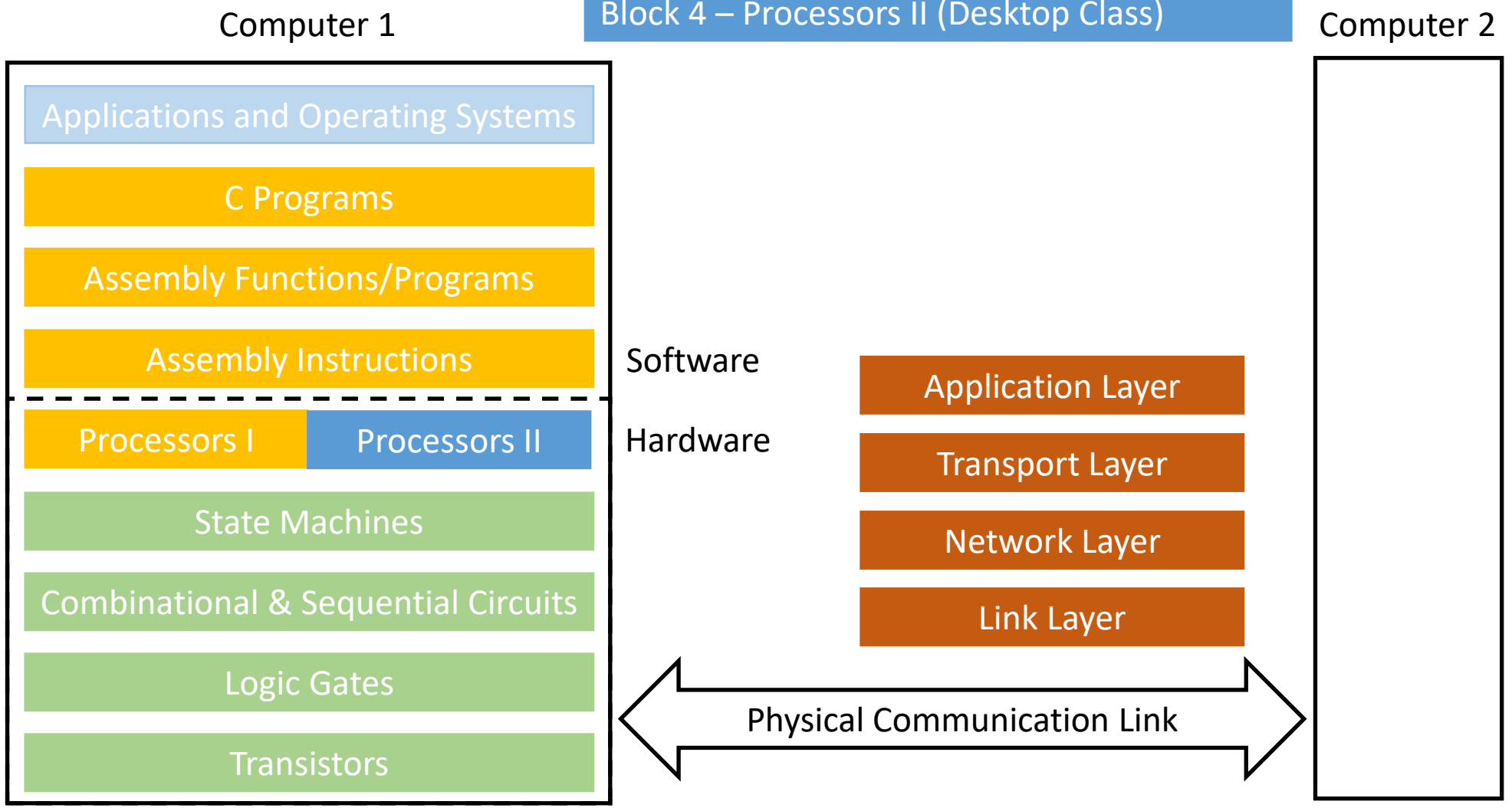
The Big Picture

Block 1 - Basics

Block 2 – Processors I (Microcontroller Class)

Block 3 - Networks

Block 4 – Processors II (Desktop Class)



Goal

- Get to know the machine you program → only this allows to write optimized code

- Understand the specifications of your device

ACM Turing Awards

- The Turing Award is the most prestigious award in computer science – it is the Noble Price of Computer Science

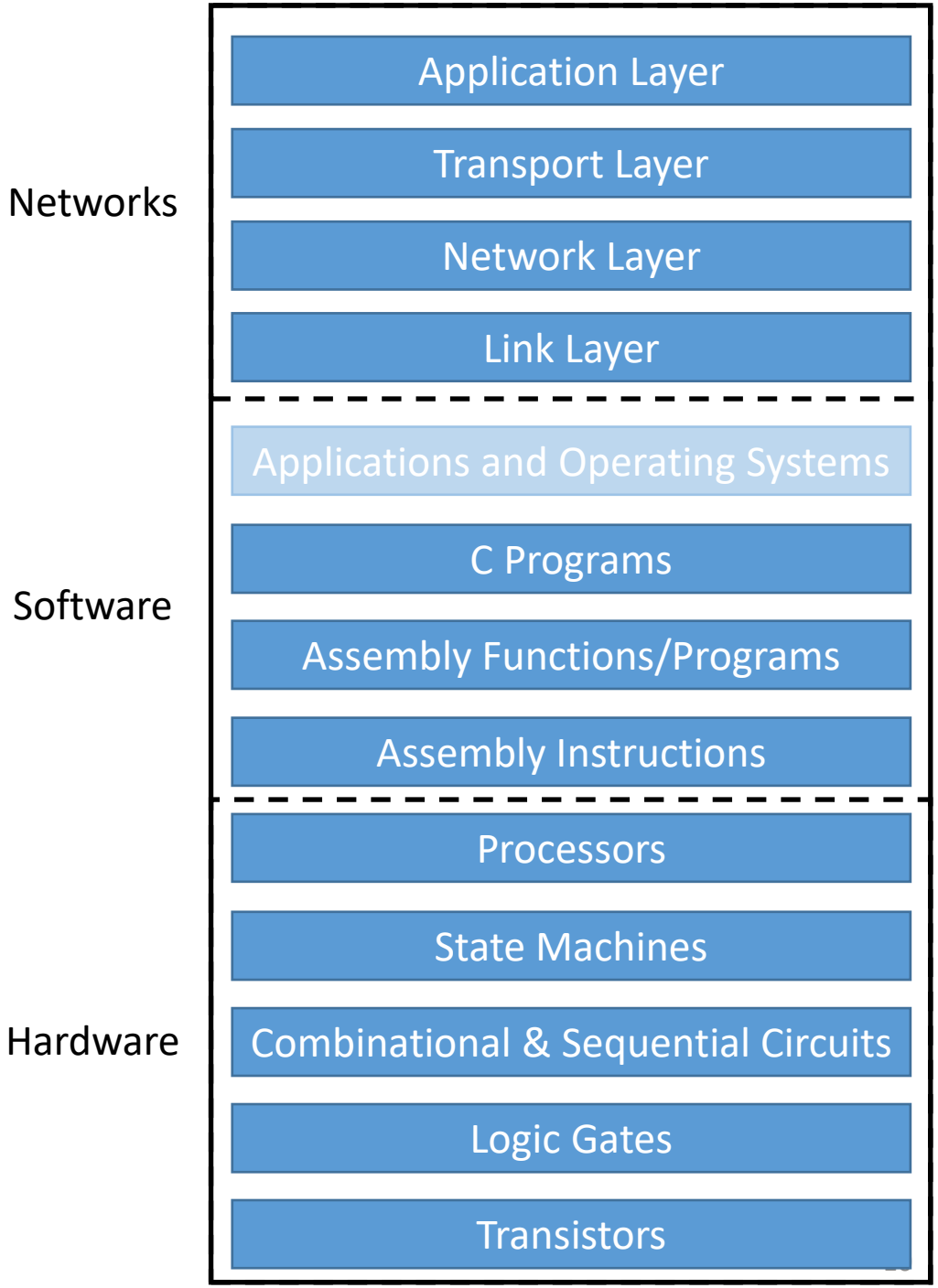
- David A. Patterson and John L. Hennessy received the Turing Award 2017 for their work on computer architectures and organization

Watch their Turing Lecture:

https://www.acm.org/hennessy-patterson-turing-lecture

Computer Organization and Networks

- In this course, we learn the basics to get the **big picture** → dig deeper in follow-up courses!

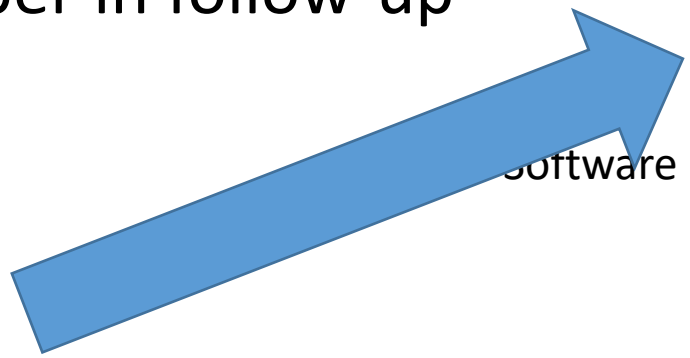


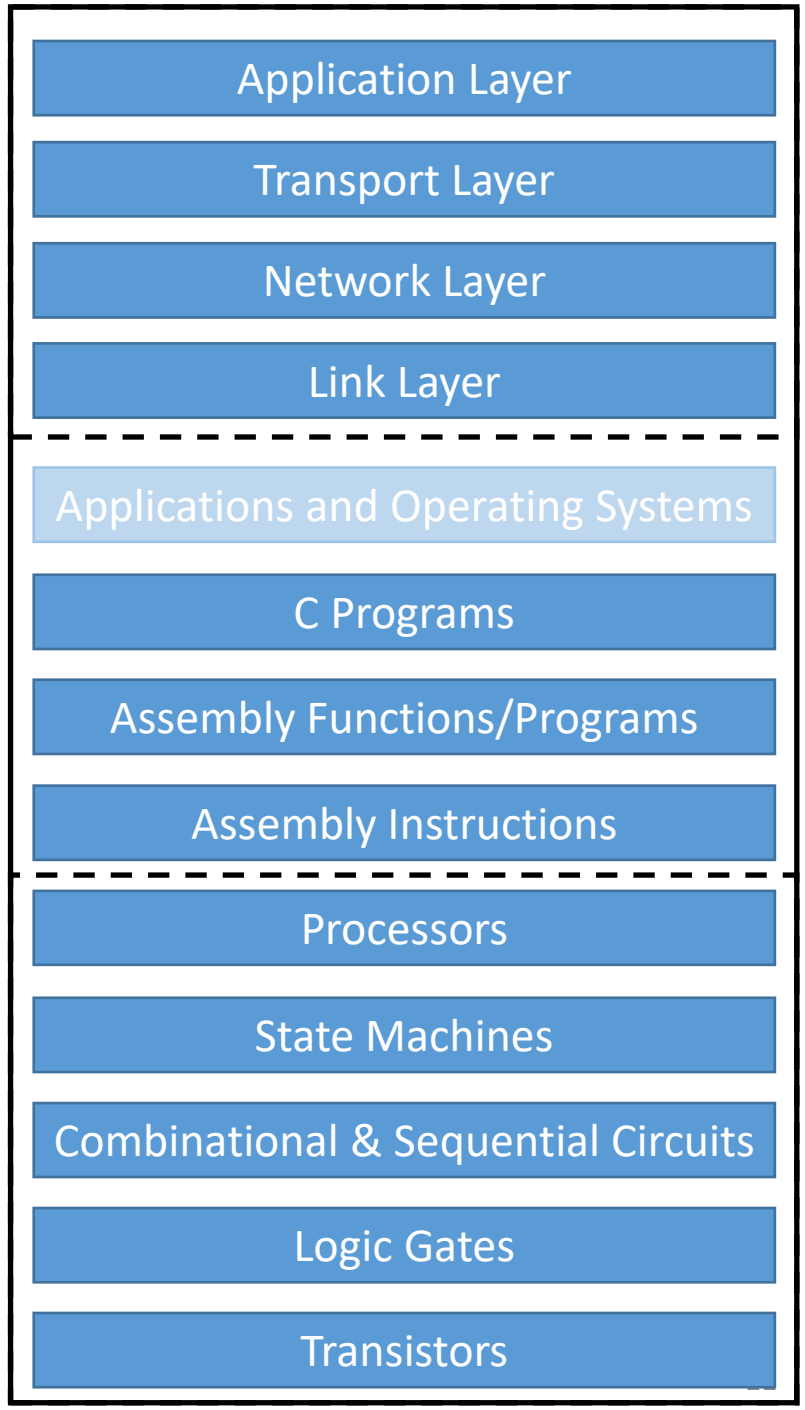
More?

- In this course, we learn the basics to get the big picture → dig deeper in follow-up courses!

- **System-Level Programming**
- **Operating System**

"Build your own OS"





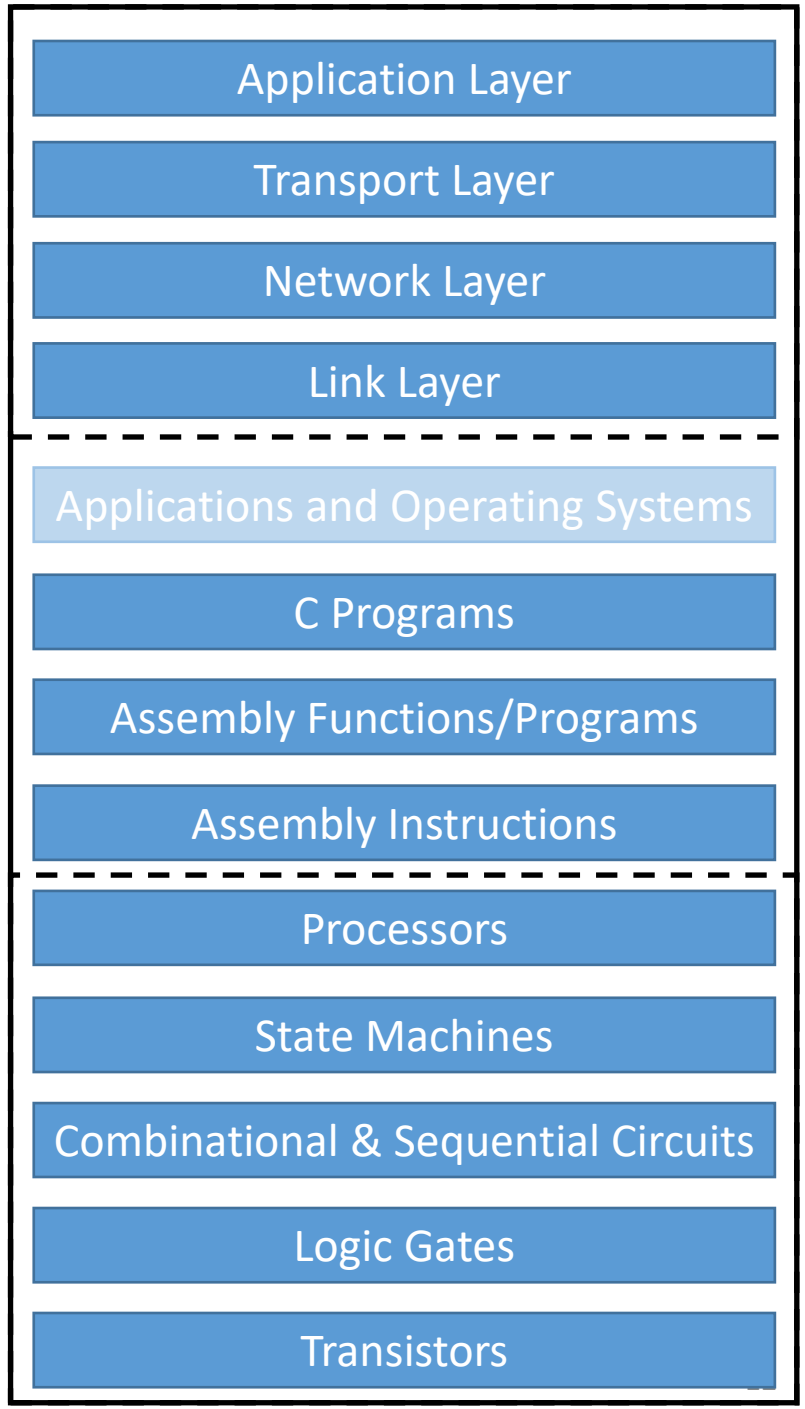
More?

- In this course, we learn the basics to get the big picture → dig deeper in follow-up courses!

- **Digital System Design**

"Build your own hardware"

https://opentitan.org/

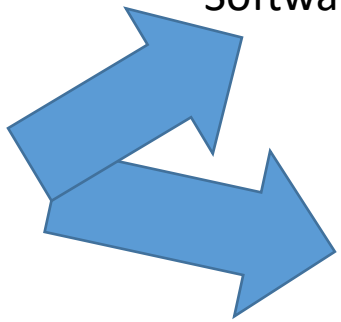


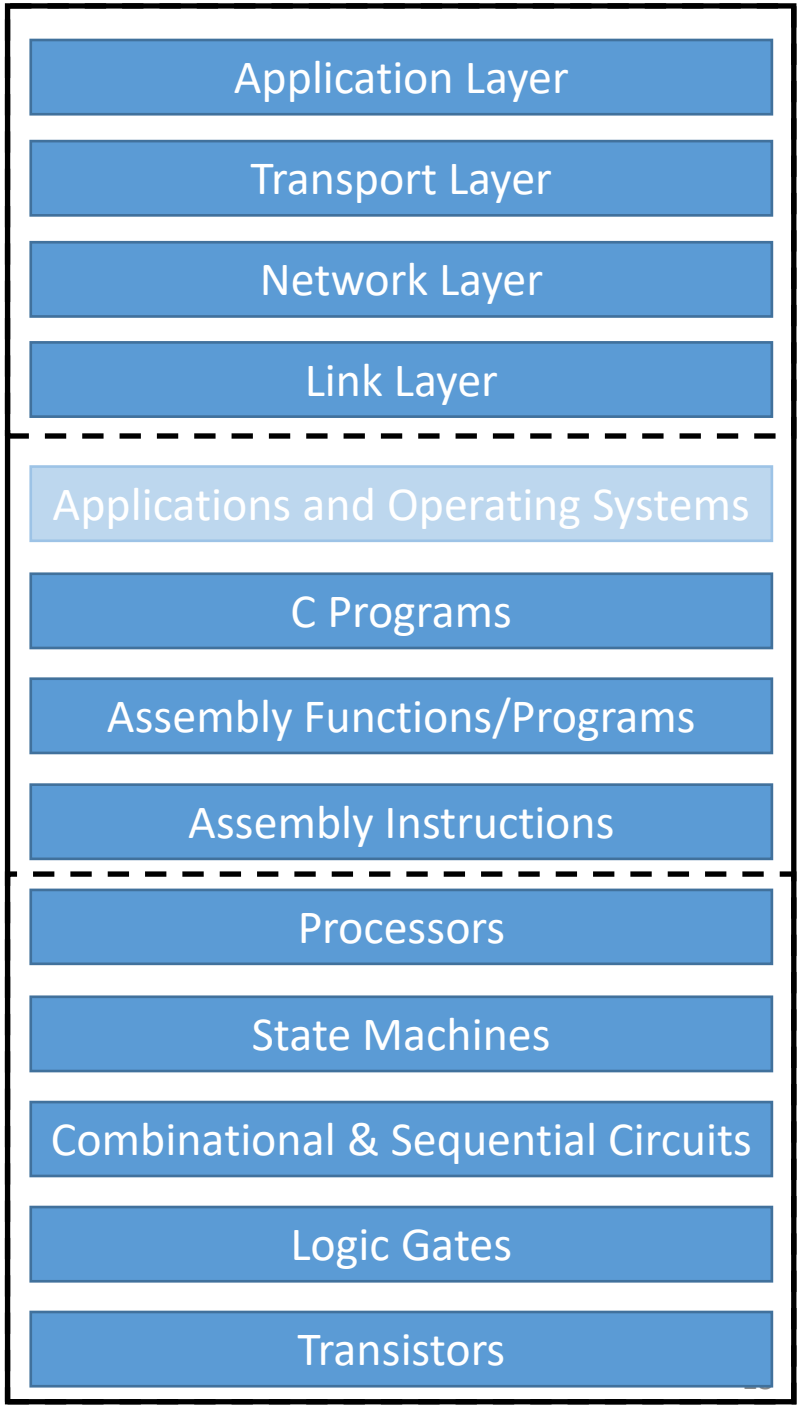
More?

- In this course, we learn the basics to get the big picture → dig deeper in follow-up courses!

- **System Integration and Programming**

"Build your own hardware and integrate it in Linux"



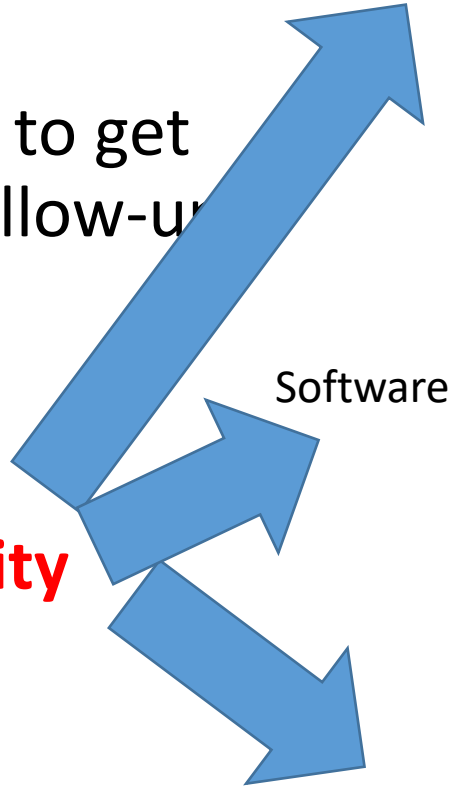


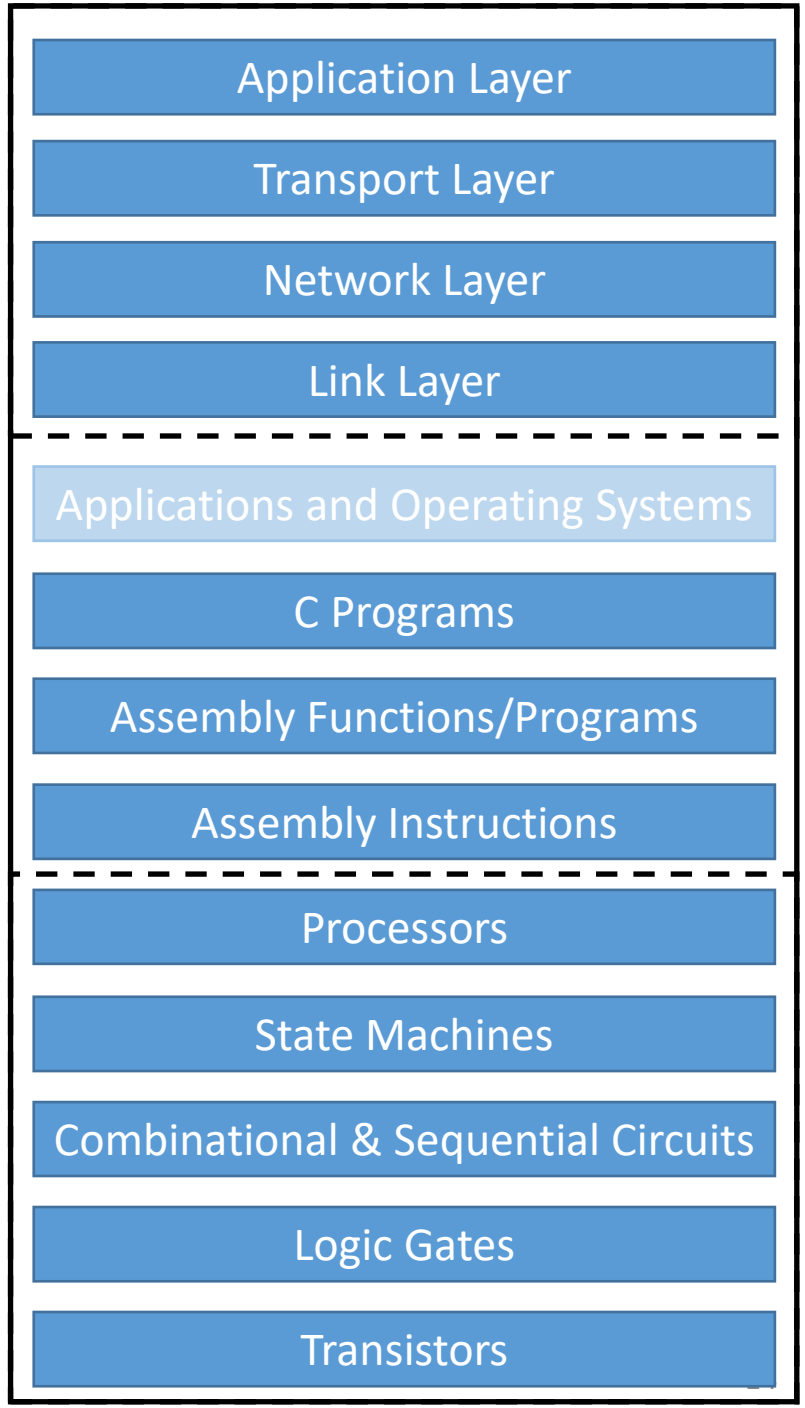
More?

- In this course, we learn the basics to get the big picture → dig deeper in follow-u courses!

- **Introduction to Information Security**

"Learn about Security on all Layers"





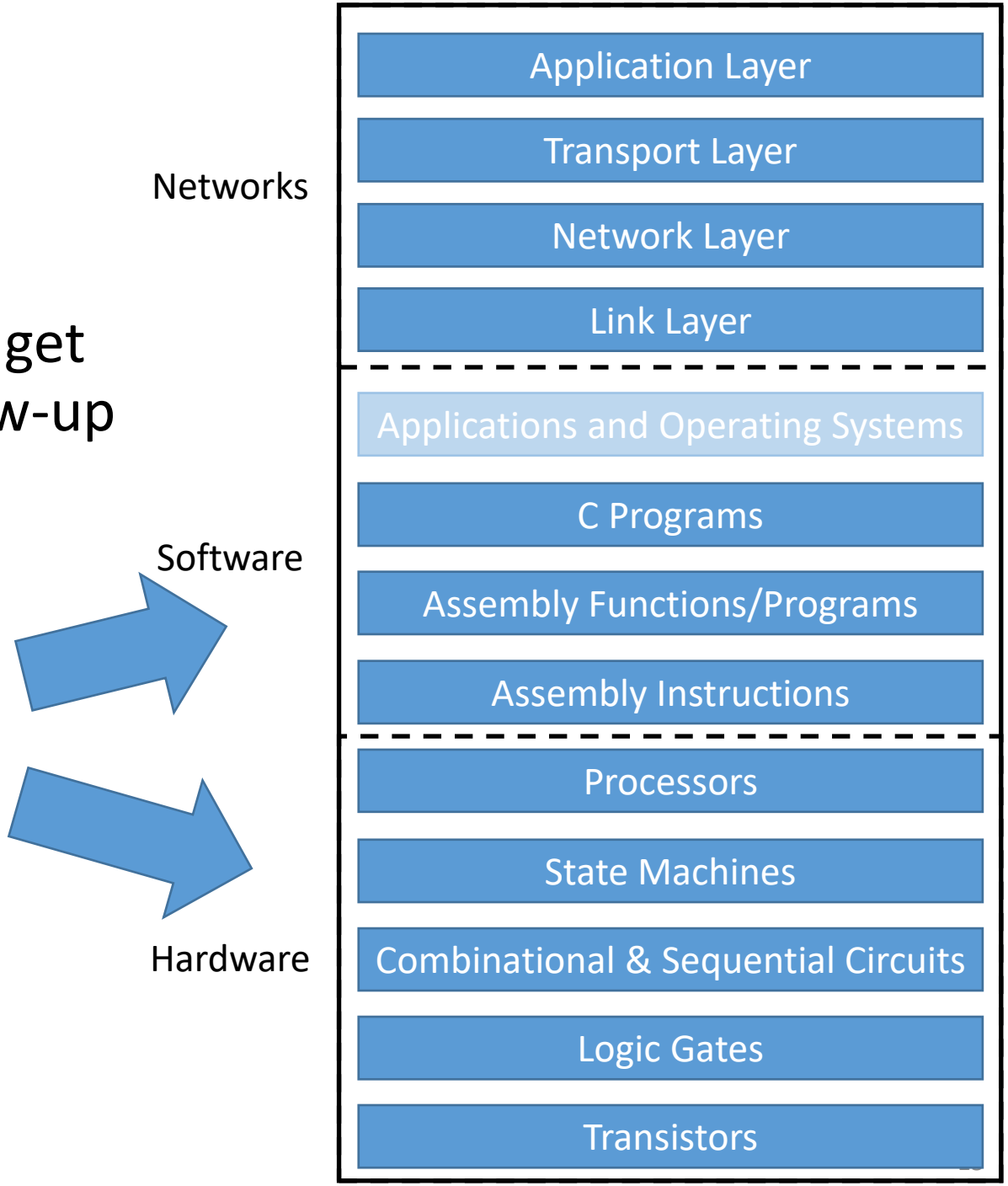
More?

- In this course, we learn the basics to get the big picture → dig deeper in follow-up courses!

- **Side Channel Security**

"Learn about the fatal consequences of side channels"

https://meltdownattack.com/



Administrative Stuff

Position in Curricula

- **Compulsory course** in **semester 3** for
  - 211 Information and Computer Engineering
  - 285 Digital Engineering
  - 521 Computer Science
  - 524 Software Engineering and Management
- **Elective compulsory** course in **semester 3** for
  - 054, 414 Supplementary Bachelor's program Teacher Training: Secondary Schools (General Education), Subject: Informatics
  - 198 Teacher Education Programme for Secondary Level

Team

Stefan Mangard

Jakob Heher

Martin Unterguggenberger

Robert Schilling

Teaching Assistants

- Sebastian Felix
- Matthias Fischer
- Markus Grebien
- Sarah Hörtnagel
- Nives Krizanec
- Patrick Krumpl
- Constantin Piber
- Oliver Popa
- Alexander Schalk
- Daniel Scharf
- Felix Schatzl
- Patrick Schuster

Teaching Assistants


Sebastian Felix

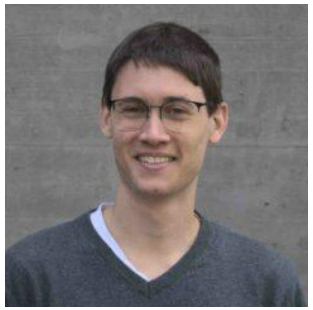

Matthias Fischer


Markus Grebien


Sarah Hörtnagel

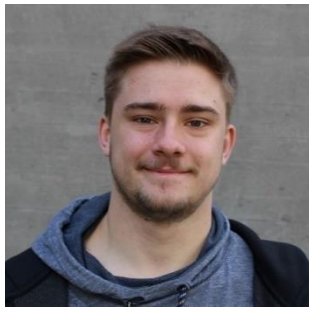

Nives Krizanec

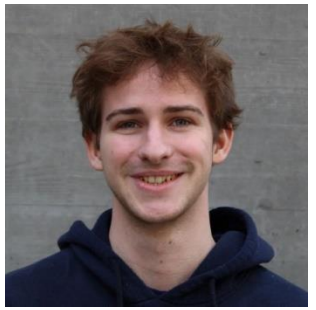

Patrick Krumpl


Constantin Piber


Oliver Popa


Alexander Schalk


Daniel Scharf


Felix Schatzl


Patrick Schuster

Material and Contact

- Email

  con@iaik.tugraz.at

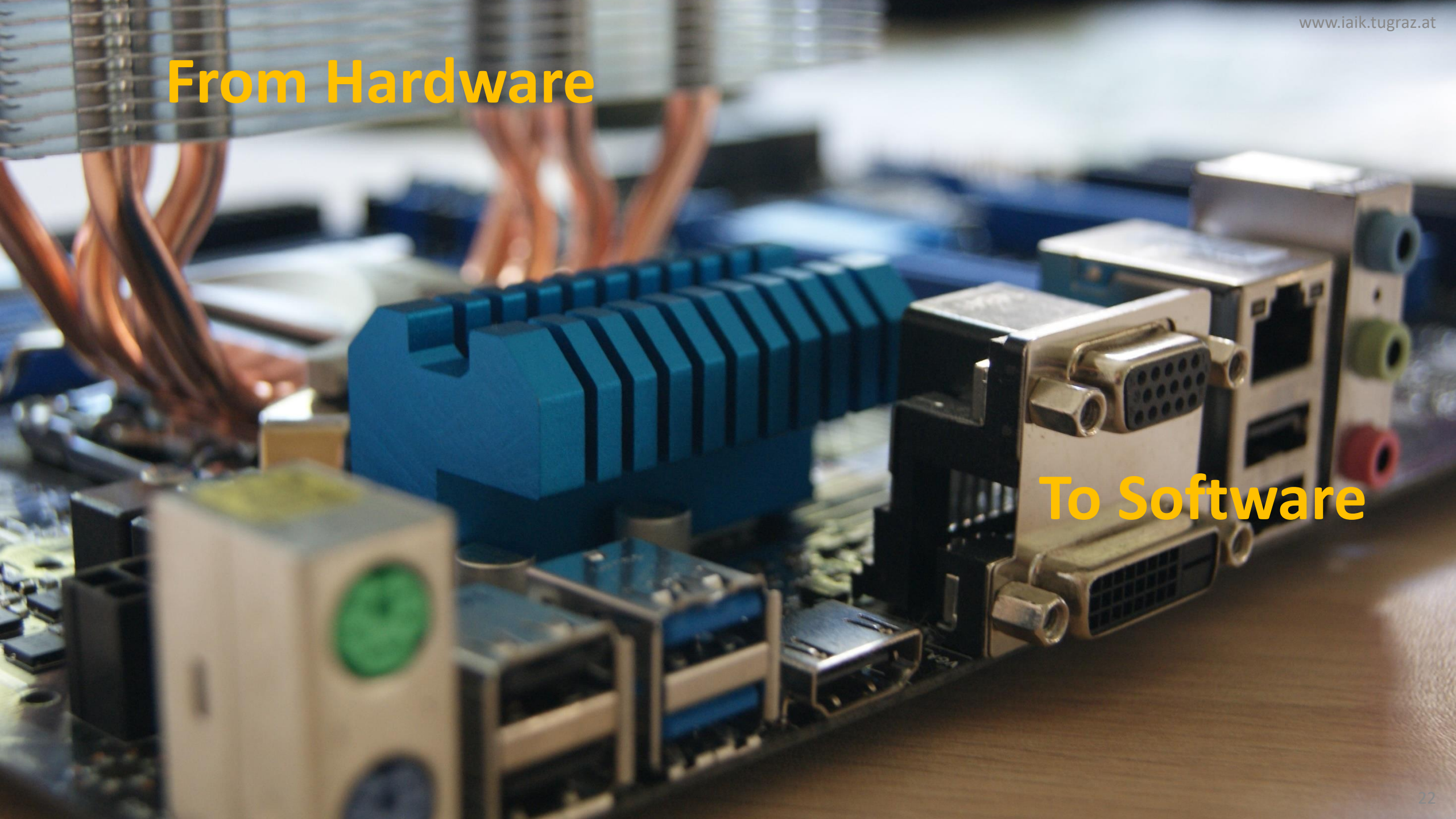
- Course website including all material

  http://www.iaik.tugraz.at/con

- Discord invitation link

  https://discord.com/invite/mxuUnjP

Lecture


From Hardware
To Software

Lecture

- Location
  - HS i13
  - Online Streaming via TU Graz Tube

- Time

  Each week on Wednesday there is a block of 120min lecture plus a 10 min break

  - 13:00 – 14:00 lecture block 1
  - 14:00 – 14:10 break
  - 14:10 – 15:10 lecture block 2

  The positioning of the break may vary ;-)

- Programming examples are available from
  https://extgit.iaik.tugraz.at/con/examples-2022/

Lecture Content and Timeline

- **Block 1: Hardware Basics (Stefan Mangard)**
  - Combinational and Sequential Circuits
  - Number Representation and Arithmetic
  - Finite State Machines

Block 1 – Hardware Basics

- **Block 2: Processors I – Microcontroller Size (Stefan Mangard)**
  - Basics of Processor Design
  - Pipelining
  - Peripherals
  - Hardware/Software Contract, Assembly Programming, Stack

Block 2 – Processors I

- **Block 3: Networks (Jakob Heher)**
  - Network Layer
  - Transport Layer
  - Application Layer

Block 3 – Networks

- **Block 4: Processors II – Large Processors (Stefan Mangard)**
  - Out-of-Order Execution, Multiprocessor Systems
  - Caches
  - Virtual Memory

Block 4 – Processors II

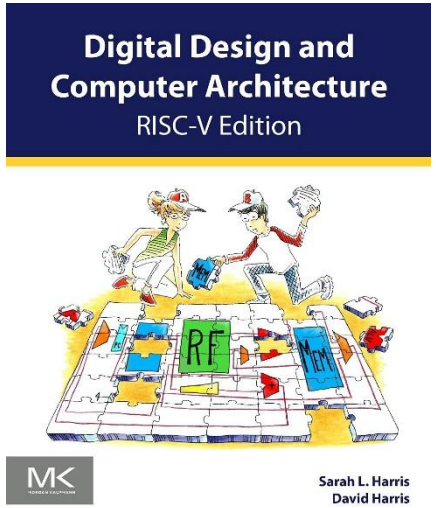
Material

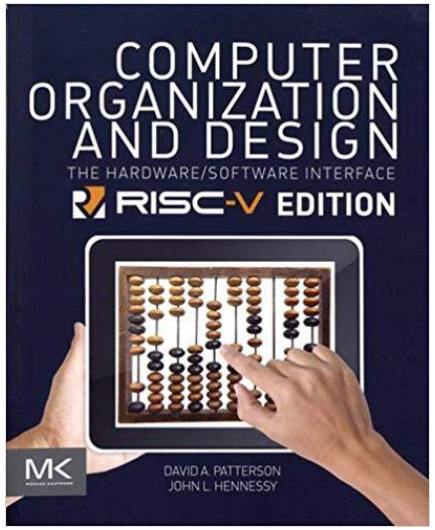
- Slides
  - Central source

- This course is based on the RISC-V instruction set
  - Many tutorials and materials can be found on the web
    https://riscv.org/
    https://riscv.org/exchange/?_sft_exchange_category=learning

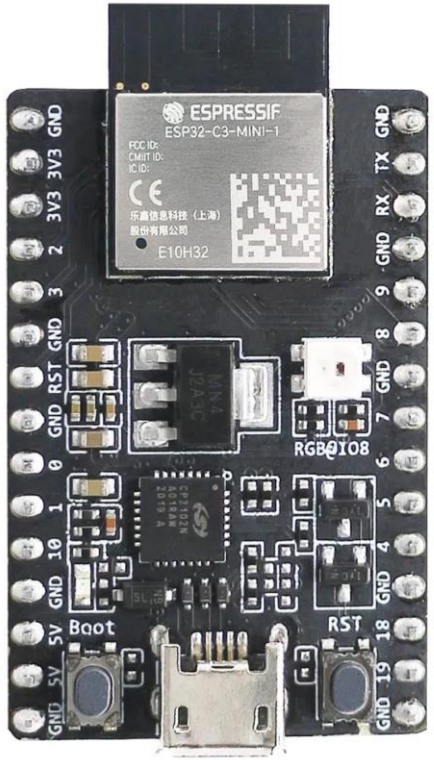
- Textbooks:
  - Digital Design and Computer Architecture, RISC-V Edition
    (Sarah L. Harris and David Harris)

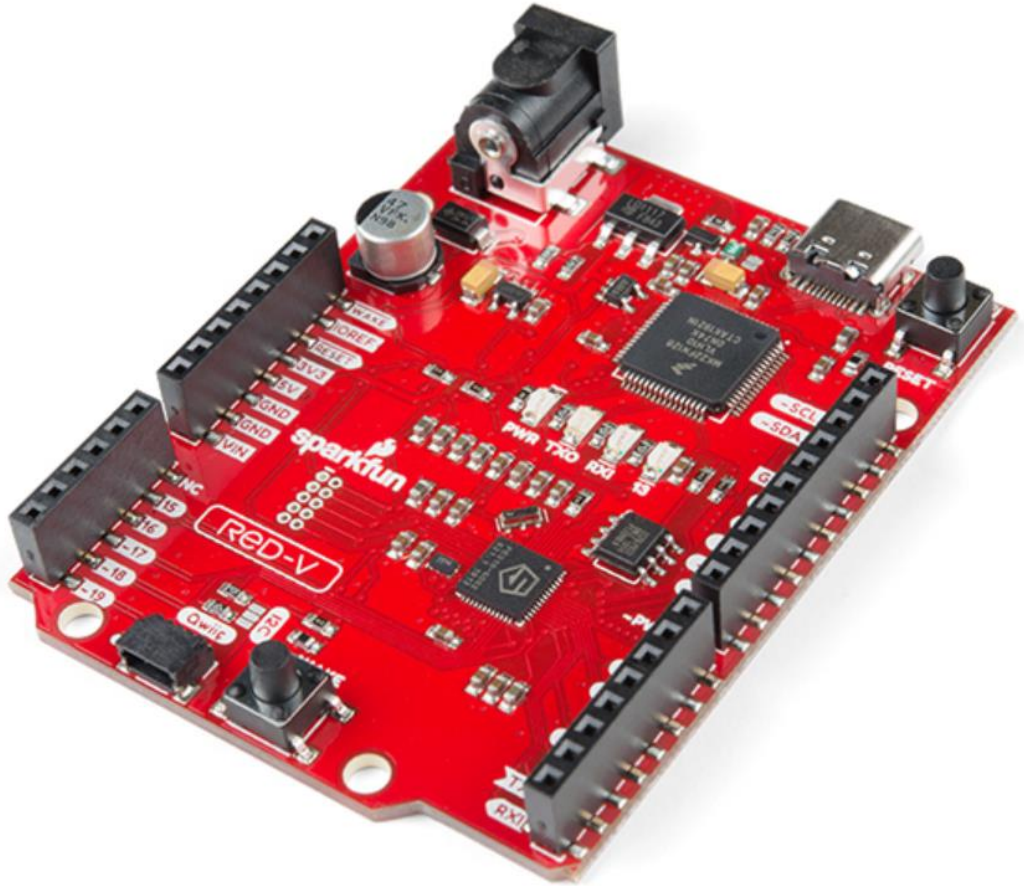
  - Computer Organization & Design: The Hardware/Software Interface, RISC-V Edition
    (David A. Patterson and John L. Hennessy)





Example Platforms





https://www.espressif.com/en/products/socs/esp32-c3

https://www.sparkfun.com/products/15594

Practical

Tasks

Deadline	Topic	Toolchain	Points
28.10.2022	Multiplier	SystemVerilog	15
04.11.2022	GCD Accelerator	SystemVerilog	20
25.11.2022	Peripheral Integration and Interrupts	SystemVerilog	20
02.12.2022	Binary Insertion Sort	RISC-V Assembly	15
13.01.2023	Firewall	C/C++	10
20.01.2023	Internet Radio	C/C++	20

88–100	Sehr gut (1)
76–87	Gut (2)
63–75	Befriedigend (3)
51–62	Genügend (4)
0–50	Nicht genügend (5)

Mode of Operation

- There is a PDF assignment containing all tasks
- There is a video tutorial for each assignment
  (+ extra video tutorial for “Getting Started” and “SystemVerilog”)
- All tutorials take place online via Discord → organized as Q & A sessions
- Three tasks with 2 subtasks each. One week between deadlines of subtasks.

Assignment

- PDF Assignment is distributed with your git repository this week
- Video tutorials for the next task will be published no later than with the deadline of the task before

Publication Date	Tutorial Video Content
06.10.2022	Getting started
06.10.2022	SystemVerilog
06.10.2022	Task 1.a
06.10.2022	Task 1.b
04.11.2022	Task 2.a
04.11.2022	Task 2.b
02.12.2022	Task 3.a
02.12.2022	Task 3.b

Question hours

- 11 groups
- Question hours start next week (10.10.2022)
- Weekly question hours specific for each group

	Monday	Tuesday	Wednesday	Thursday	Friday
09:00	Sebastian	Patrick S.	Markus		
10:00	Patrick K.	Daniel	Alexander		
11:00	Sarah	Nives	Felix		
13:00	Oliver	Constantin			

Submissions

- Submission via GitLab
- GitLab repositories will be distributed via Email by the end of this week

Interviews

- Interviews will happen on Discord (microphone needed!)
- TA will pick you up from #con-waiting-room
- Tentative schedule (may slighty vary per TA group)

Week	Interview scope
05–9.12.2022	Task 1.a, 1.b, 2.a
23–27.01.2023	Task 2.b, 3.a, 3.b

Resources

- Course web:
  - https://www.iaik.tugraz.at/con
- Code examples shown during the lecture
  - https://extgit.iaik.tugraz.at/con/examples-2022
- Virtual machine with all tools installed
  - https://seafile.iaik.tugraz.at/f/4af9802977b24525b447/
- Tutorials:
  - Regularly, almost every week

Plagiarism

- **We perform plagiarism checks!**
- **All involved people** fail the practical
  - We will not invest time on researching who copied from whom
- If you plagiarize parts of the program, it is still a case of plagiarism
- How to avoid plagiarism?
  - **Do not share code!**
  - Do not tell/dictate others your solution!
  - Commit regularly to your git repository!
  - The practical is **no group work!**

Plagiarism example: Identical program

```
#include <stdio.h>


  if(X8 > 23) goto Print;
  X5 = X4 + X8;
  X7 = X3 - X8;

  if(X5 < 0) goto LessThanZero;

  X3 = X3 + X5;
  X4 = X4 >> X7;
  X2 = *(X1 + X8);
  X5 = X5 - X2;
  X8++;
  goto Start;

LessThanZero:
  X3 = X3 >> X5;
  X2 = *(X1 + X8);
  X5 = X5 * X2;
  X8++;
  goto Start;

  return 0 ;
}
```

```
#include <stdio.h>


  if(X8 > 23) goto Print;
  X5 = X4 + X8;
  X7 = X3 - X8;

  if(X5 < 0) goto LessThanZero;

  X3 = X3 + X5;
  X4 = X4 >> X7;
  X2 = *(X1 + X8);
  X5 = X5 - X2;
  X8++;
  goto Start;

LessThanZero:
  X3 = X3 >> X5;
  X2 = *(X1 + X8);
  X5 = X5 * X2;
  X8++;
  goto Start;

  return 0 ;
}
```

Plagiarism example: Variables renamed

```
#include <stdio.h>


  if(X8 > 23) goto Print;
  X5 = X4 + X8;
  X7 = X3 - X8;

  if(X5 < 0) goto LessThanZero;

  X3 = X3 + X5;
  X4 = X4 >> X7;
  X2 = *(X1 + X8);
  X5 = X5 - X2;
  X8++;
  goto Start;

LessThanZero:
  X3 = X3 >> X5;
  X2 = *(X1 + X8);
  X5 = X5 * X2;
  X8++;
  goto Start;

  return 0 ;
}
```

```
#include <stdio.h>


   if(X6 > 23) goto Print;
   X3 = X2 + X6;
   R4 = X1 - X6;

   if(X5 < 0) goto Negative;

   X1 = X1 - X3;
   X2 = X2 >> R4;
   X8 = *(X7 + X6);
   X5 = X5 - X8;
   X6++;
   goto Begin;

  Negative:
   X1 = X1 >> X3;
   X8 = *(X7 + X6);
   X5 = X5 * X8;
   X6++;
   goto Begin;

   return 0;
}
```

This is still the same program!

Plagiarism example: Branches flipped

```
#include <stdio.h>


  if(X8 > 23) goto Print;
  X5 = X4 + X8;
  X7 = X3 - X8;

  if(X5 < 0) goto LessThanZero;

  X3 = X3 + X5;
  X4 = X4 >> X7;
  X2 = *(X1 + X8);
  X5 = X5 - X2;
  X8++;
  goto Start;

LessThanZero:
  X3 = X3 >> X5;
  X2 = *(X1 + X8);
  X5 = X5 * X2;
  X8++;
  goto Start;

  return 0 ;
}
```

```
#include <stdio.h>


  if(X6 > 23) goto Print;
  X3 = X2 + X6;
  R4 = X1 - X6;

  if(X5 >= 0) goto Positive;

  X1 = X1 >> X3;
  X8 = *(X7 + X6);
  X5 = X5 * X8;
  X6++;
  goto Begin;

Positive:
  X1 = X1 + X3;
  X2 = X2 >> R4;
  X8 = *(X7 + X6);
  X5 = X5 - X8;
  X6++;
  goto Begin;

    return 0;
}
```

This is still the same program!

Do not invest time on trying to bypass detection of plagiarism

Invest your time on the assignments

Your First Actions for the Practical

- Register in TUGRAZonline **(deadline: today)**
- Install the CON2022 Virtual Machine

By the end of this week all registered students receive their git repositories including assignment sheet

- Clone your repository
- Read the assignment sheet
- Watch the video tutorial for assignment 1
- Attend the online tutorials next week

Effort

- This is a 7 ECTS course – this is approximately one quarter of your semester (approx. 200 working hours)
- There are 120 minutes lecture per week
- This lecture and the practical runs through many abstraction layers with many different tools – work on the course every week (“Am Ball bleiben”)

Selected Related Student Teams

FORMULA STUDENT

International Design Competition established to promote cooperation between students and automotive industry.

JOIN THE TUG RACING TEAM!

Have we caught your interest?

More details are provided at our KickOff Meeting on **11.10.2022** or at the open workshop days on **4./6./13.10**!

Scan the QR code and stay up to date!

SCAN ME

Aerospace Team Graz

- Development of Rockets
- International Competitions
- 75 members – 14 fields of studies

- Sturmstand → October 25, from 16:00
- Open-House → October 27, from 14:00

www.astg.at

LosFuzzys

Local CTF team

Interested in everything security related

Weekly meetups

↳every Wednesday 18:00 at the FuzzyLab