Computer Organization and Networks

(INB.06000UF, INB.07001UF)

Welcome

Winter 2023/2024

Stefan Mangard, www.iaik.tugraz.at

What This Course is About

- How does a computer work?
- What does actually happen when I compile and execute this code?

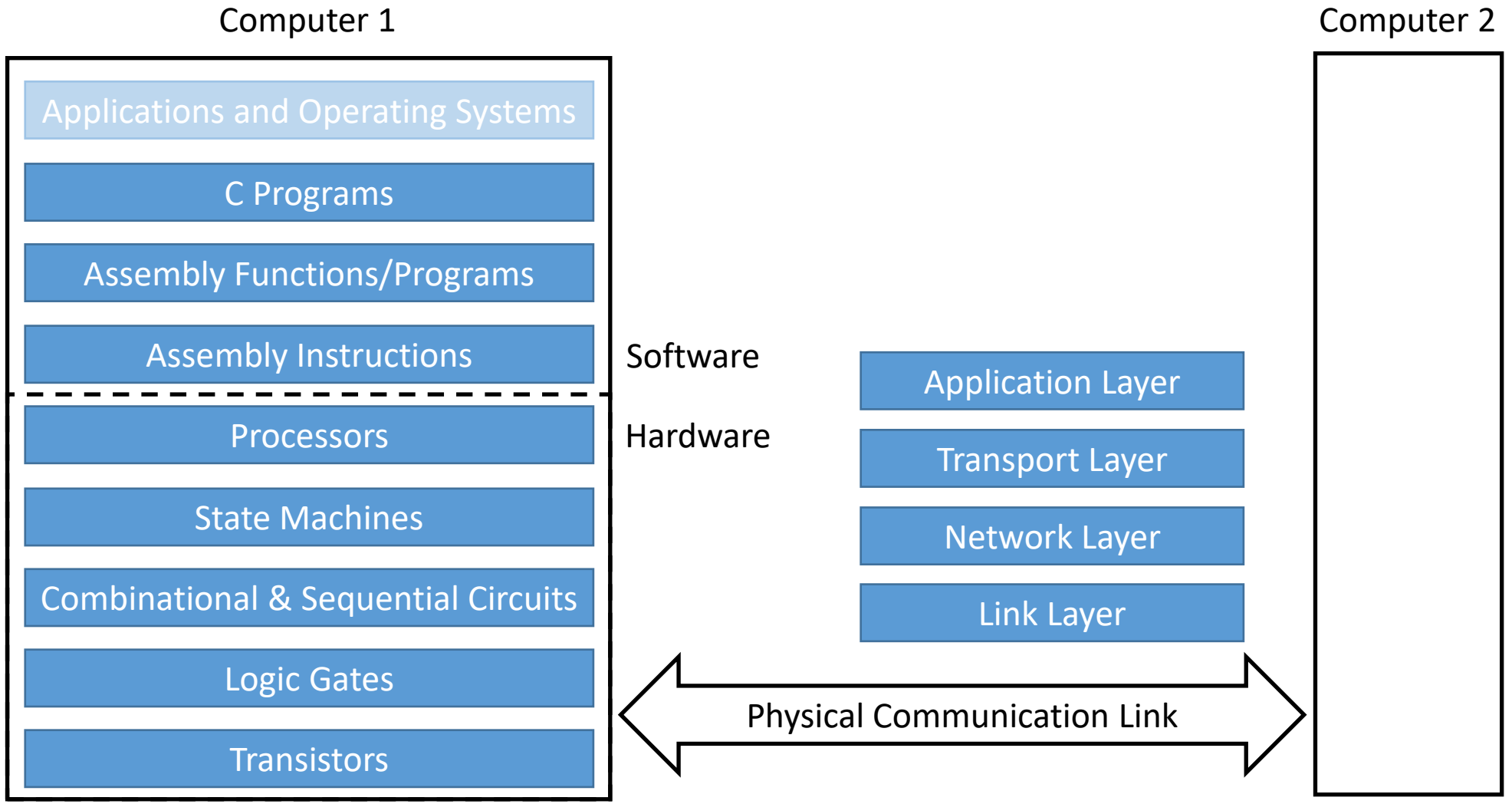
```
include <stdio.h>

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

- How do computers communicate?

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



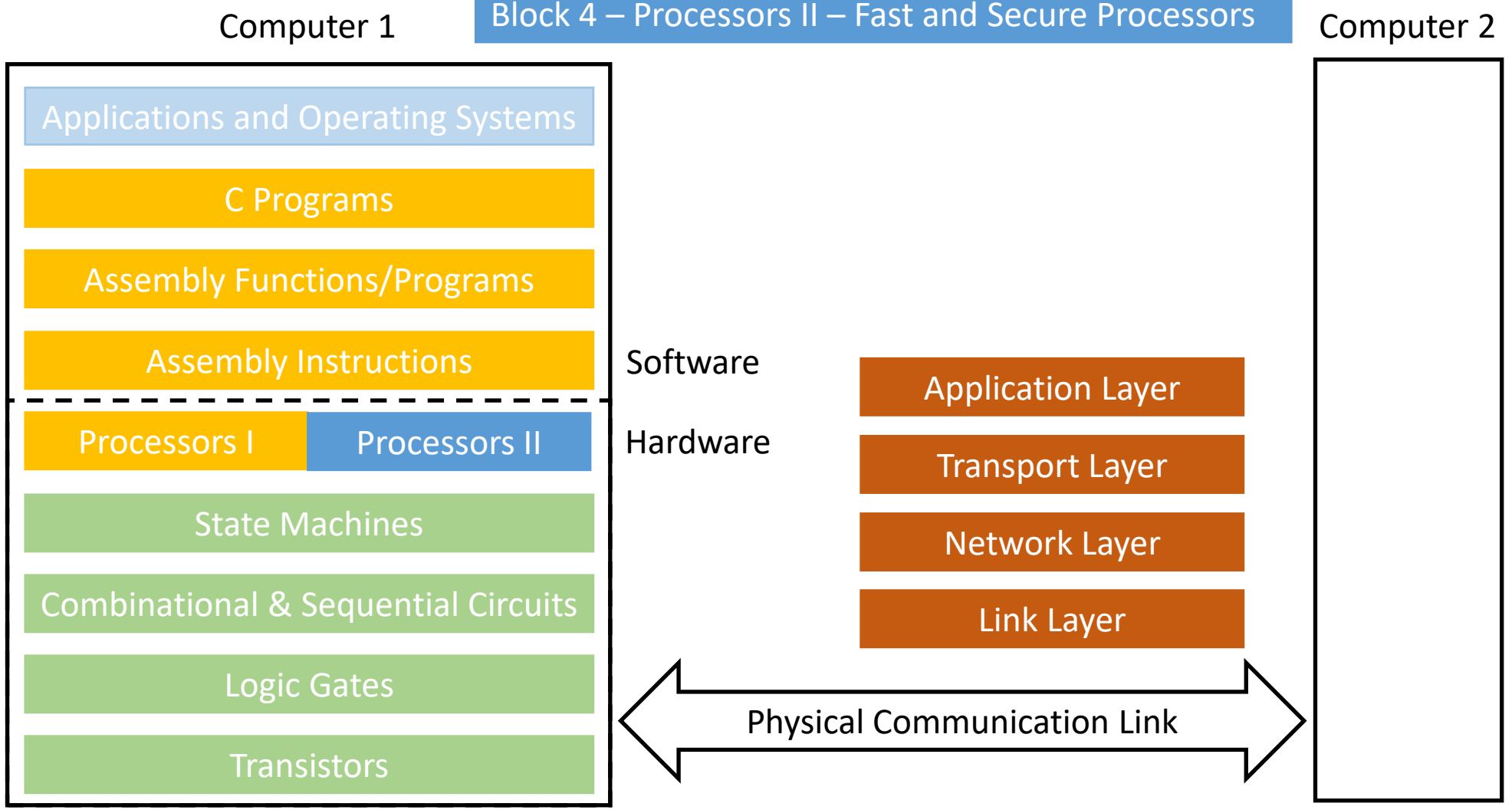
The Big Picture

Block 1 – Basics on Hardware

Block 2 – Processors I – HW/SW Interface

Block 3 - Networks

Block 4 – Processors II – Fast and Secure Processors



Goal

- Get to know the machine you program

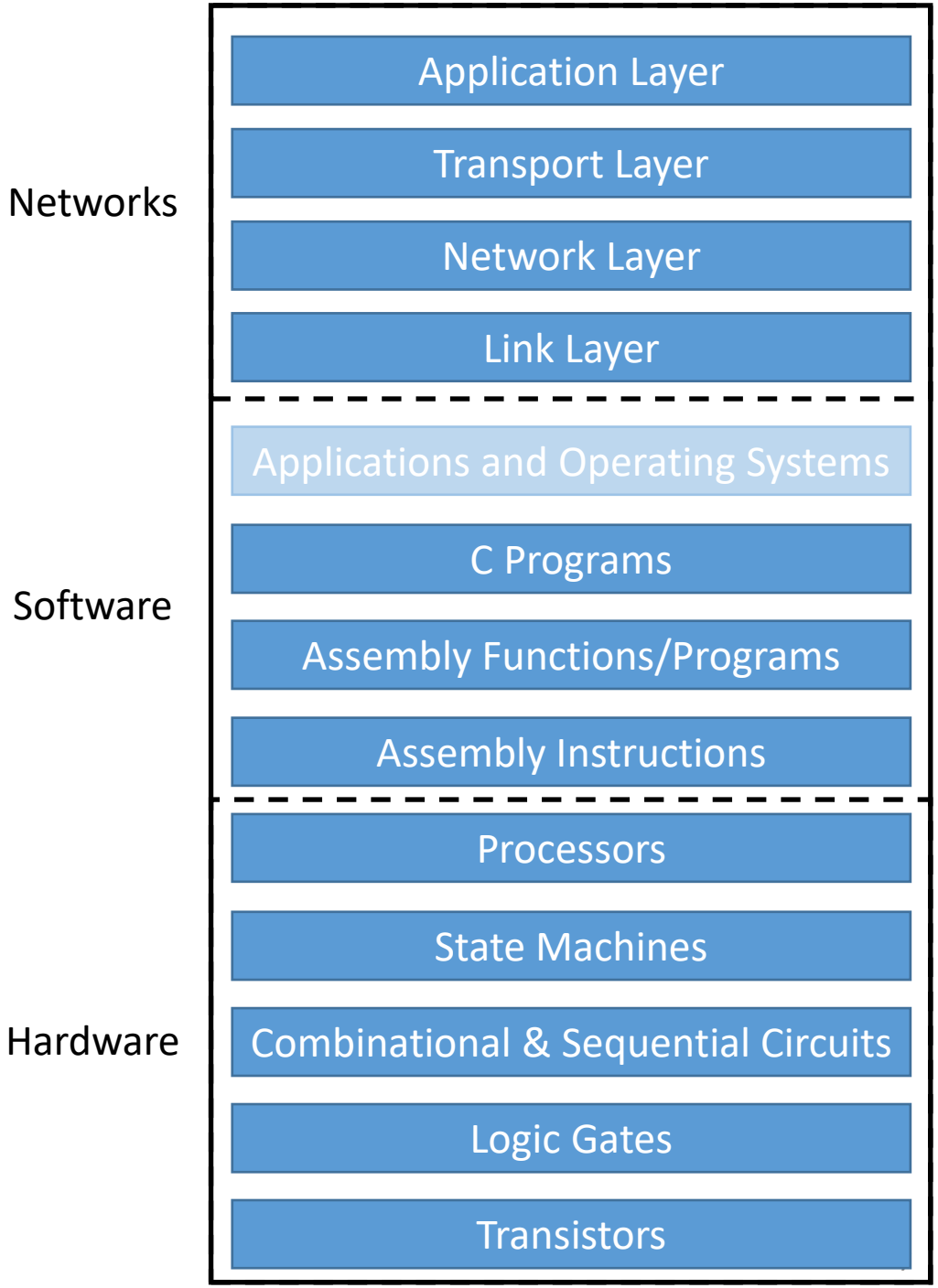
  →Understand what happens when you execute a program

  →Understand how to optimize code

  →Understand the specifications of your device
  - Clock Rates
  - Caches
  - Cores
  - Pipeline Stages
  - Execution Units
  - …

Computer Organization and Networks

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

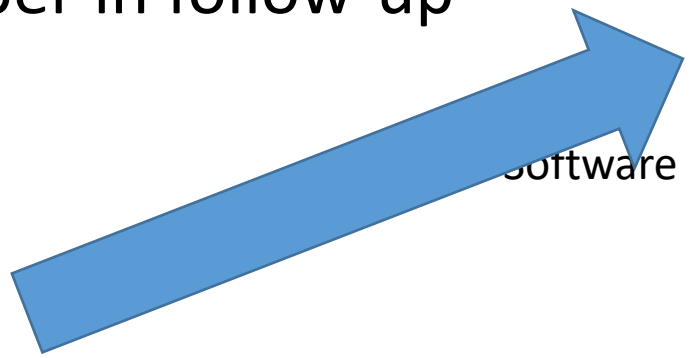


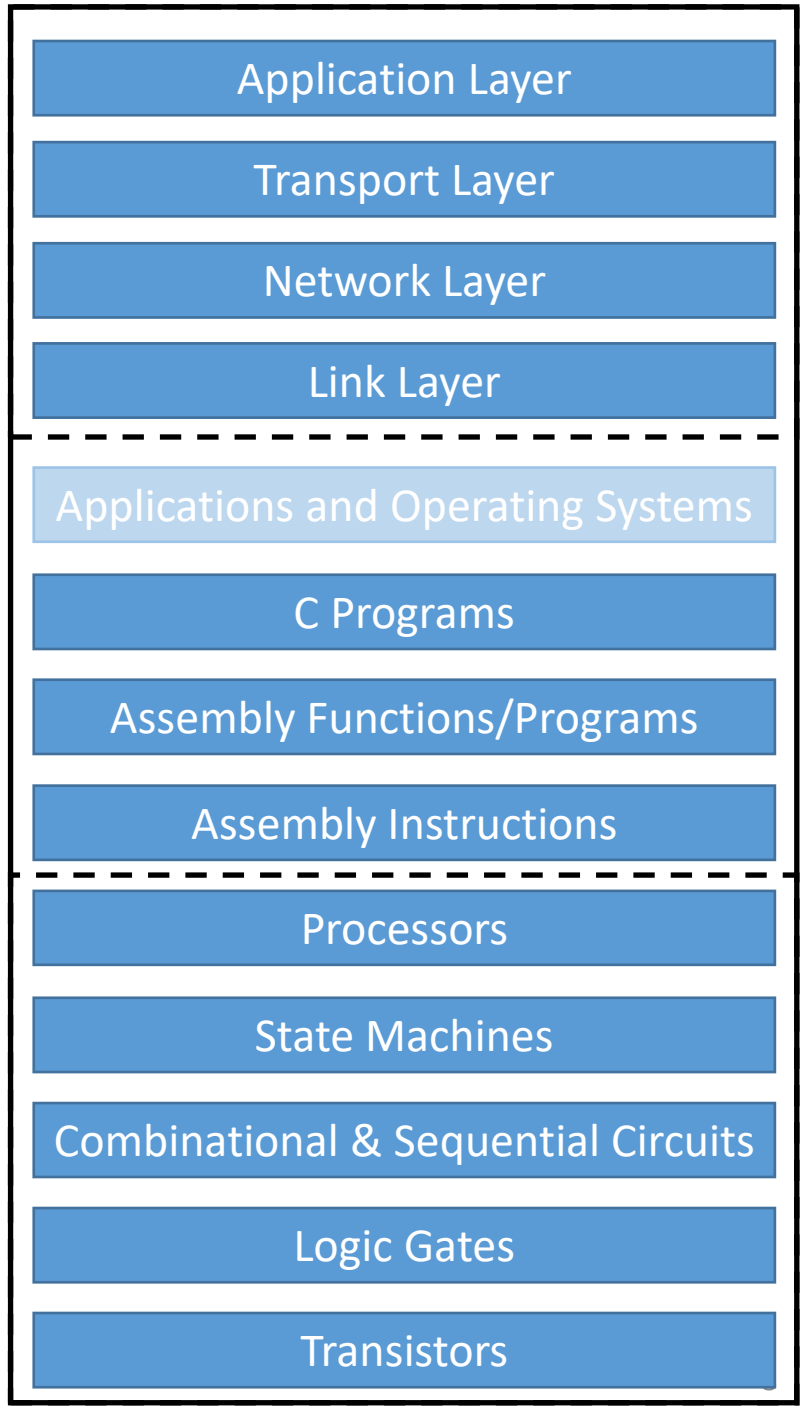
More?

- 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"





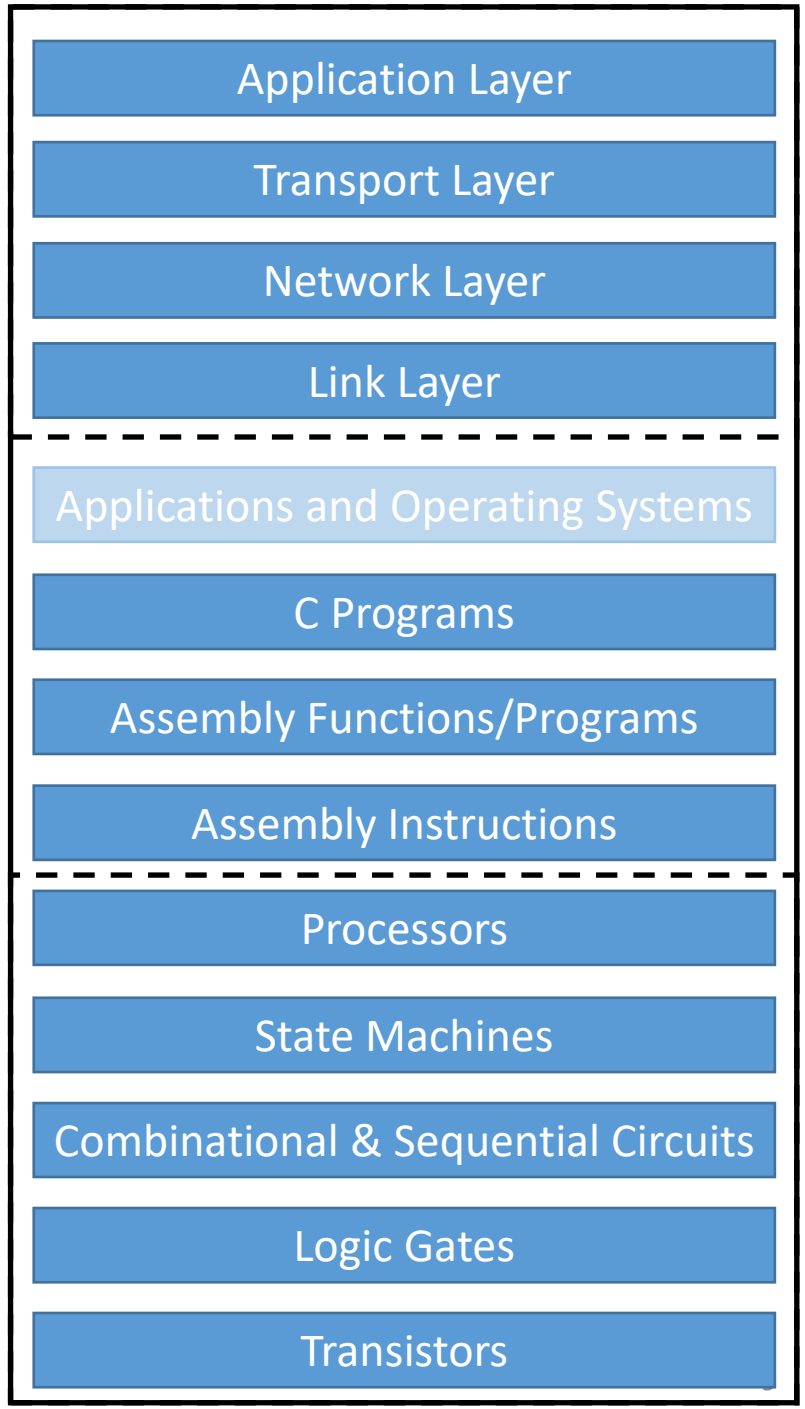
More?

- 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/

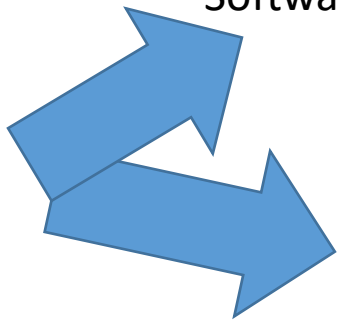


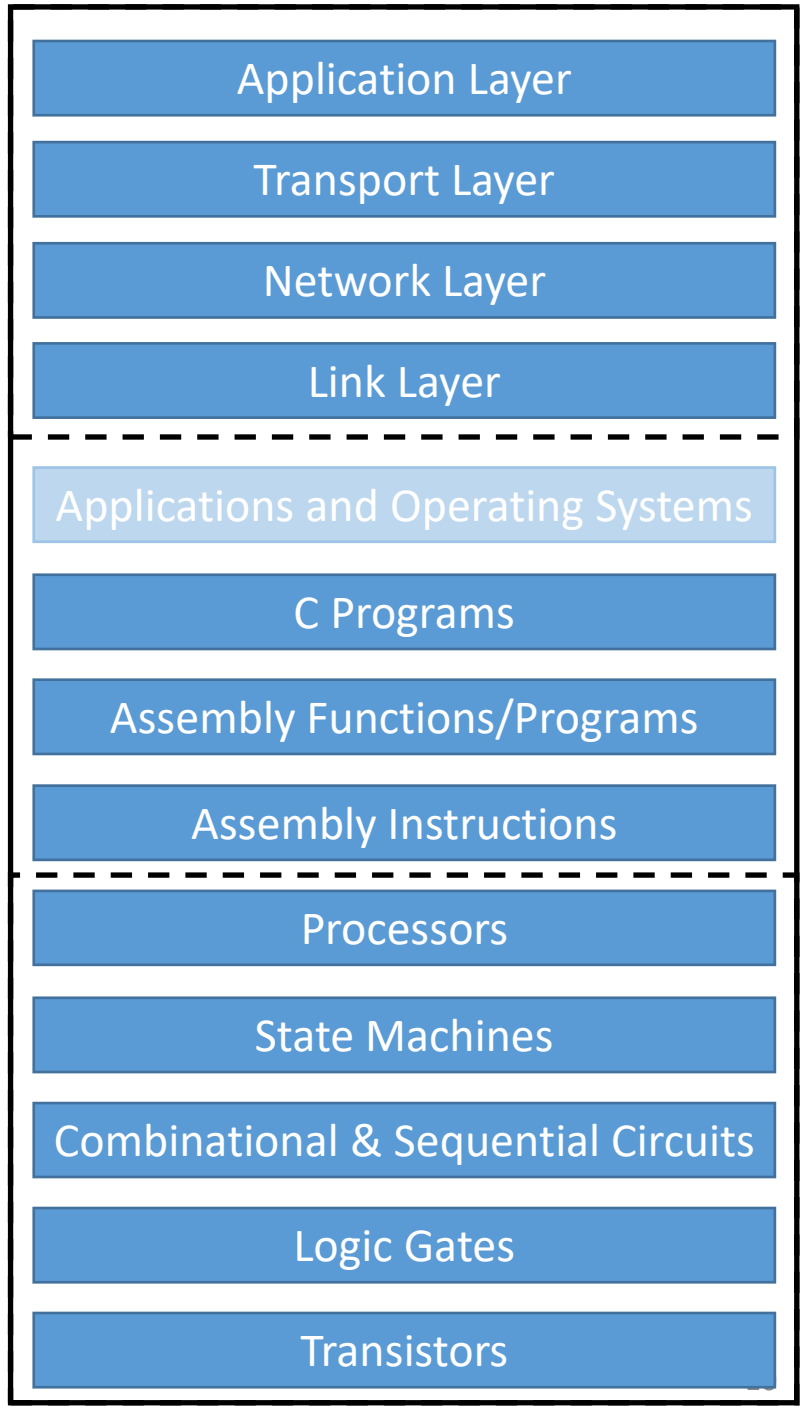
More?

- 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"



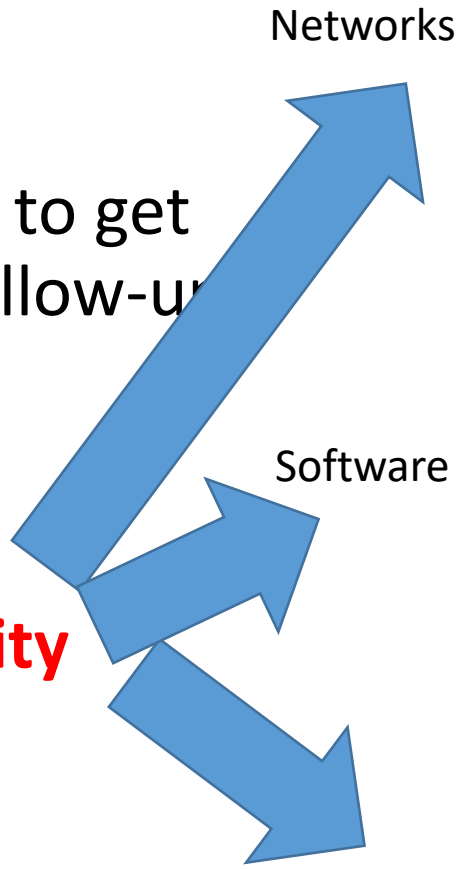


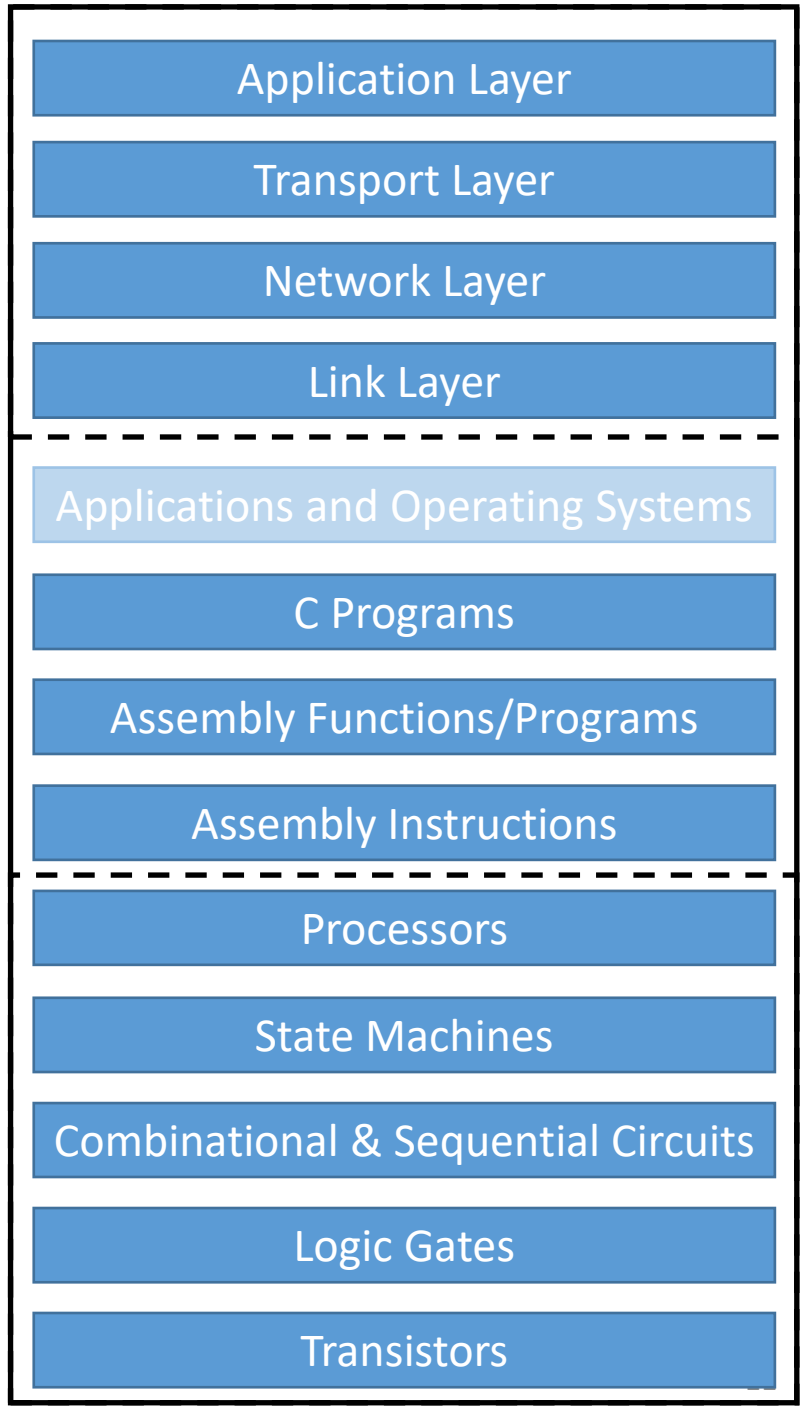
More?

- 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"





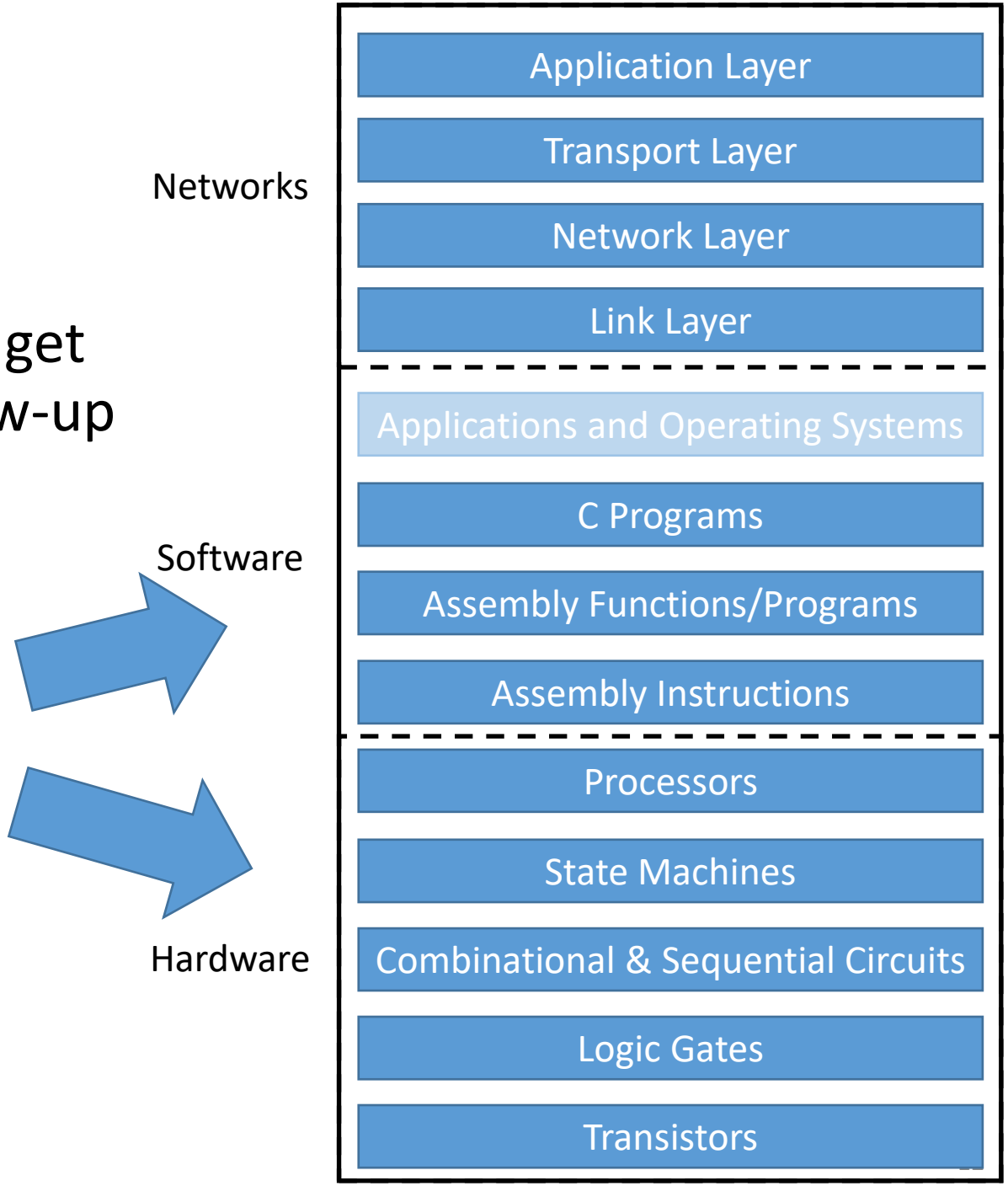
More?

- 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**
  - 211 Information and Computer Engineering
  - 285 Digital Engineering
  - 521 Computer Science
  - 524 Software Engineering and Management
- **Elective Compulsory Course**
  - 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

Lukas Lamster

Moritz Waser

Teaching Assistants

- Sebastian Felix
- Matthias Fischer
- Markus Grebien
- Philipp Kraft
- Benjamin Mörth
- Constantin Piber
- Oliver Popa
- Verena Schaffer
- Daniel Scharf
- Andreas Schlager

Teaching Assistants

Sebastian
Felix

Matthias
Fischer

Markus
Grebien

Philipp
Kraft

Benjamin
Mörth

Constantin
Piber

Oliver
Popa

Verena
Schaffer

Daniel
Scharf

Andreas
Schlager

Material and Contact

- Email

  con@iaik.tugraz.at

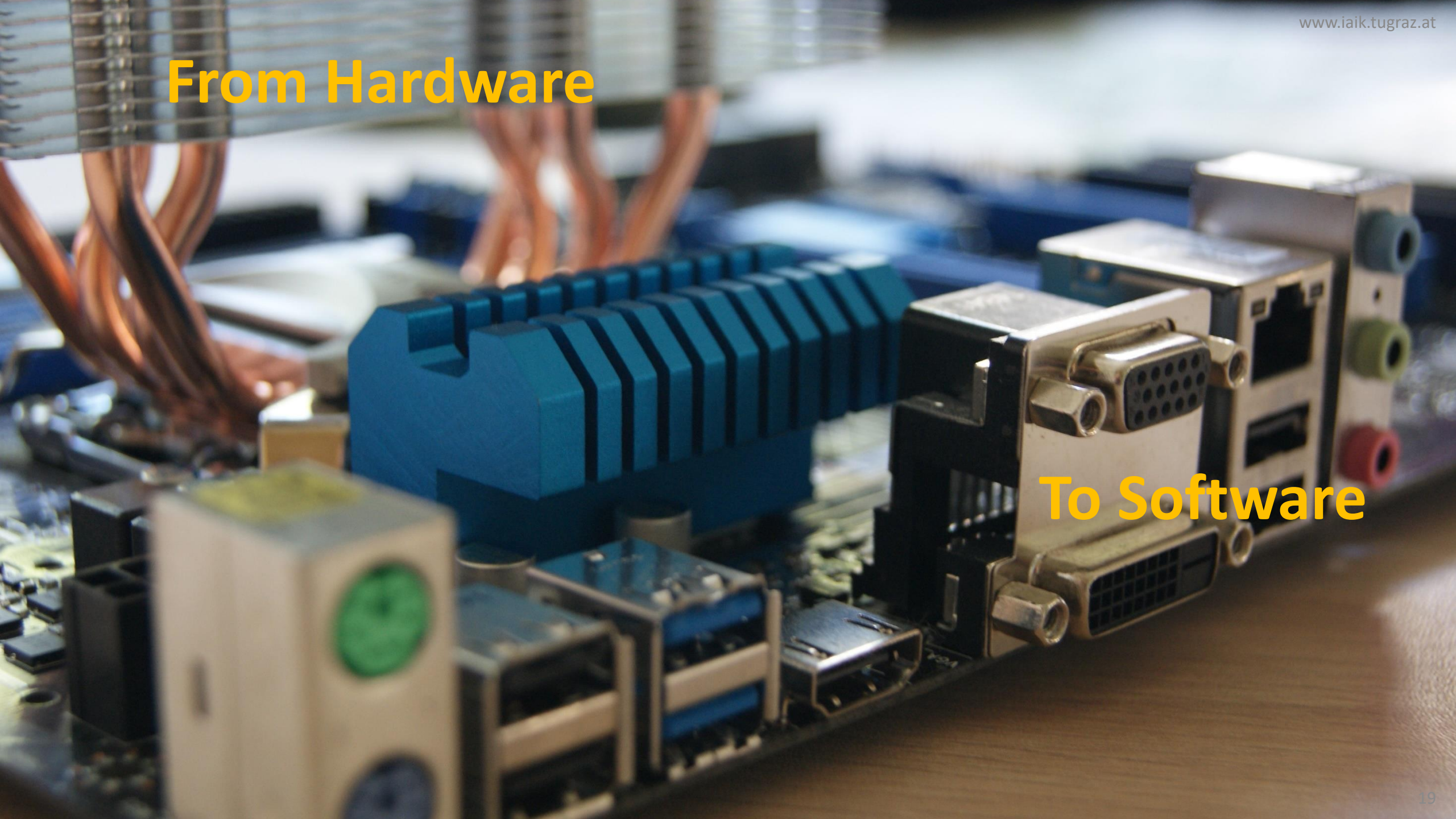
- Course website including all material

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

- Discord invitation link

  https://discord.com/invite/mxuUnjP

Lecture


From Hardware
To Software

Lecture

- Location
  - HS i13
  - Recorded lecture available 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-2023/

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 – HW/SW Interface (Stefan Mangard)**
  - Basics of Processor Design
  - 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 – Fast and Secure Processors (Stefan Mangard)**
  - Caches, Virtual Memory
  - Out-of-Order Execution, Multiprocessor Systems
  - Security

Block 4 – Processors II

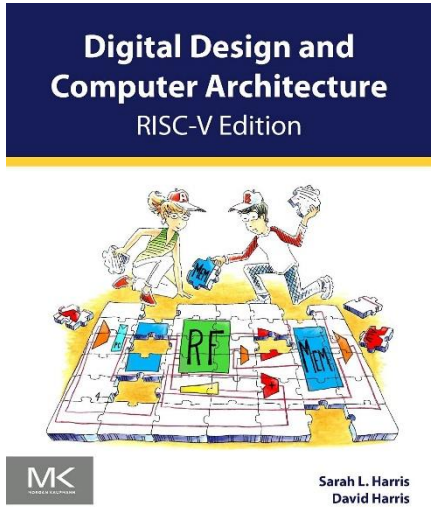
Material

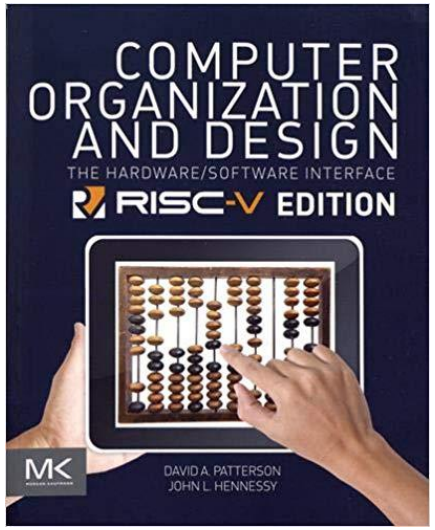
- Slides
  - Central source

- 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

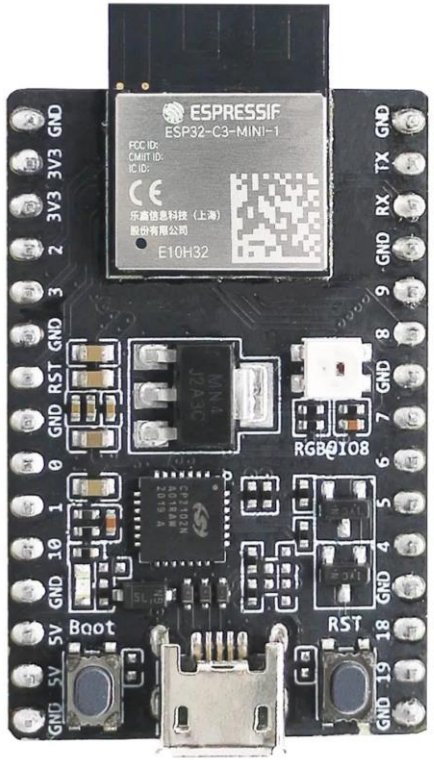
- Textbooks:
  - Digital Design and Computer Architecture, RISC-V Edition
    (Sarah L. Harris and David Harris)

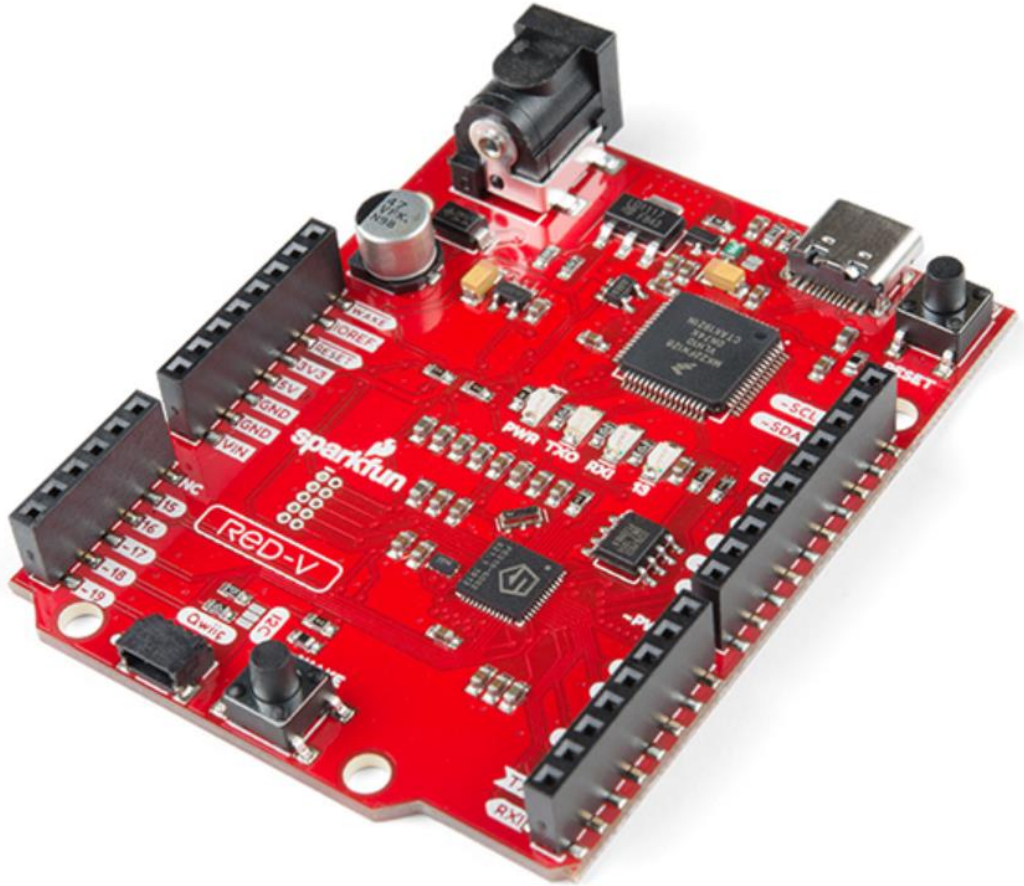
  - Computer Organization & Design: The Hardware/Software Interface, RISC-V Edition
    (David A. Patterson and John L. Hennessy)





Example Platforms

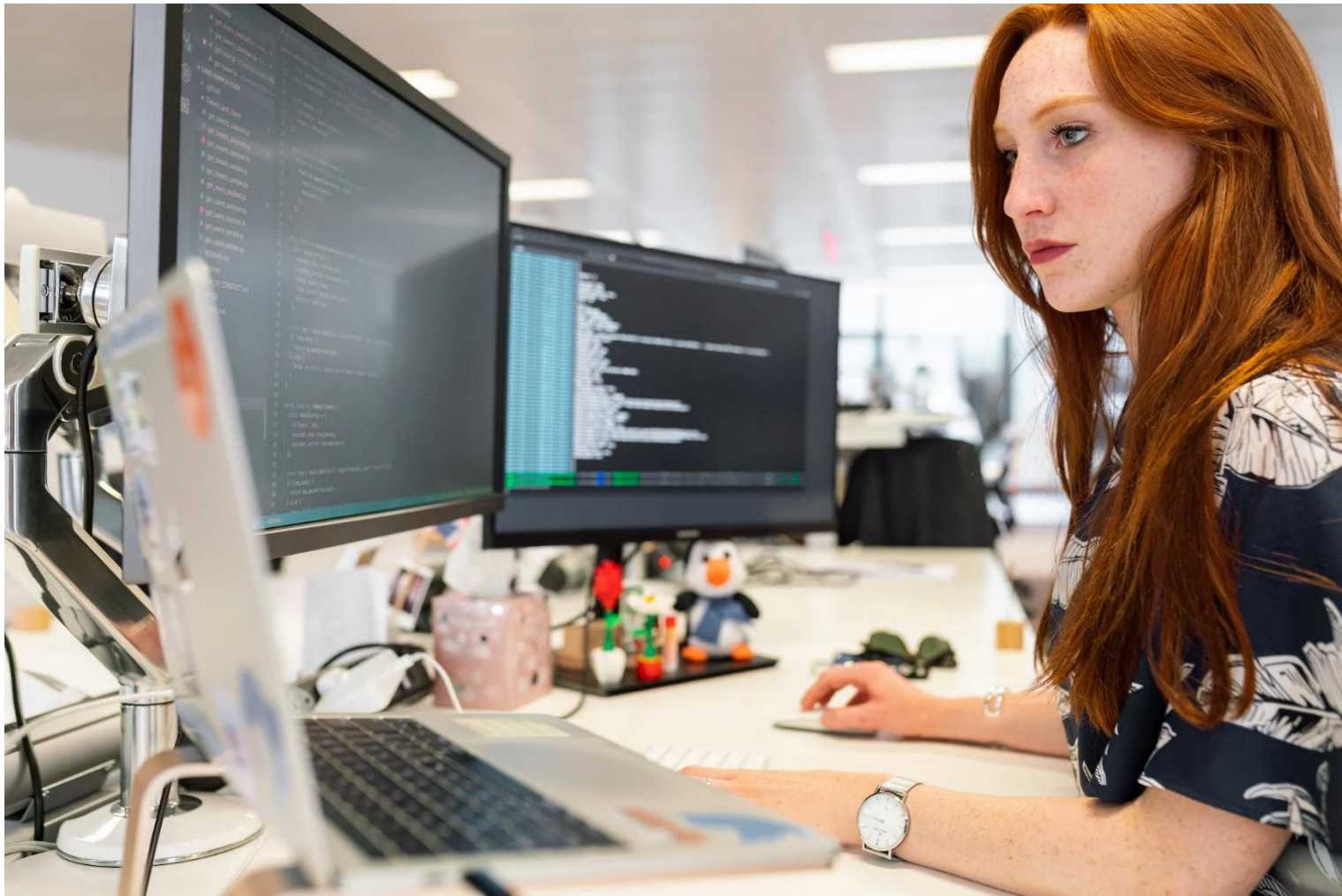




https://www.espressif.com/en/products/socs/esp32-c3

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

Practical



Tasks, Interviews, Grading

Deadline	Topic	Toolchain	Points
20.10.2023	Task 0 - Getting Started	SystemVerilog	5
10.11.2023	Task 1 - Communication Interface	SystemVerilog	25
01.12.2023	Task 2 - Binary GCD in RISC-V	RISC-V Assembly	20
22.12.2023	Task 3 - IPv4 Routing & NAT	C/C++	25
19.01.2024	Task 4 - Building a Cache	SystemVerilog	25

Expected Timeframe	Interview scope
04–13.12.2023	Task 1, Task 2
22–31.01.2024	Task 3, Task 4

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 sheet containing all tasks
  - Will be published **Monday, October 9**
- There is a video tutorial for each assignment (+ extra video tutorial for “SystemVerilog”)
  - Tutorial 0: https://seafile.iaik.tugraz.at/f/56b299d7f6f645abb574/
  - Tutorials 1-4 will be published Monday, October 9
- All tutorials take place online via Discord → organized as Q & A sessions
- There are two task interviews in person

Question hours

- 10 groups
- Question hours start next week (9.10.2023)
- Weekly question hours are specific for each group

	Monday	Tuesday	Wednesday	Thursday	Friday
09:00	X	X	X		
10:00	X	X	X		
11:00	X	X	X		
13:00	X				

Submissions

- Submission via GitLab

- GitLab repositories will be distributed via email to all registered students beginning of next week

Resources

- Course web:
  - https://www.iaik.tugraz.at/con
- Code examples shown during the lecture
  - https://extgit.iaik.tugraz.at/con/examples-2023
- Virtual machine with all tools installed (user: con password: con2023)
  - https://seafile.iaik.tugraz.at/f/62c8b40e46e74bc2b90e/
- Tutorials:
  - Regularly, 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

Use of AI Technology

- We treat the involvement of ChatGPT and similar tools the **same way as the involvement of another natural person**.

- The involvement that qualifies as plagiarism or an impermissible level of assistance, the **consequences will be the same** in both cases to the strictest extent possible.

Your First Actions for the Practical

This Week:

- Register in TUGRAZonline **(deadline: tomorrow)**
  - Beginning of this week all registered students receive their git repositories
- Install the CON2023 Virtual Machine (user: con password: con2023)
  - https://seafile.iaik.tugraz.at/f/62c8b40e46e74bc2b90e/
- Clone the examples repository and familiarize yourself with the environment
  - https://extgit.iaik.tugraz.at/con/examples-2023

Next Week:

- Watch the video tutorial for assignment 0
- Read the assignment sheet
- Clone your repository
- 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

Aerospace Team Graz

Aerospace Team Graz

- **Raketenentwicklung**: Wir konzipieren und bauen fortschrittliche Raketen.
- **Internationaler Wettbewerb**: Stolze Teilnehmer der European Rocketry Challenge in Portugal.
- **Technologie & Software**: Spezialisiert u.A. auf die Entwicklung von Elektronik und Flugsoftware.
- **Vielfältige Projekte**: Neben Raketen bieten wir auch Projekte im Bereich Kleinstsatelliten an.
- **Akademische Chancen**: Gelegenheiten für Bachelor- und Masterarbeiten in der Luft- und Raumfahrt.

Join us!

- **Praktische Erfahrung**: Werde Teil unseres erfahrenen Teams mit 87 engagierten Mitgliedern und sammle wertvolle praktische Erfahrungen.
- **Open-House Event**:
  - **Datum**: 23. Oktober
  - **Uhrzeit**: Ab 13:00 Uhr
  - **Ort**: ASTG Büro, Inffeldgasse 16b, ICEG094
- **Sturmstand Event**:
  - **Datum**: 25. Oktober
  - **Uhrzeit**: Ab 16:00 Uhr
  - **Ort**: Campusplatz Inffeldgasse, TU Graz

www.astg.at

LosFuzzys

Local CTF team

Interested in everything security related

Weekly meetups

↳every Wednesday 18:00 at the FuzzyLab

Scan Me