Computer Organization and Networks

(INB.06000UF, INB.07001UF)

Welcome

Winter 2020/2021

Stefan Mangard, www.iaik.tugraz.at

COVID-19 Prolog

Lecture and Practical will be All Digital Events

- **Lecture**
  - Links for Microsoft Teams will be sent by email before each class
    - → Let's have an **interactive lecture**
    - → The lecture should not be one-way streaming
  - Youtube streaming is available as backup
- **Practical**
  - Video tutorials
  - Discussions and tutorials via discord

Let’s Try Interaction

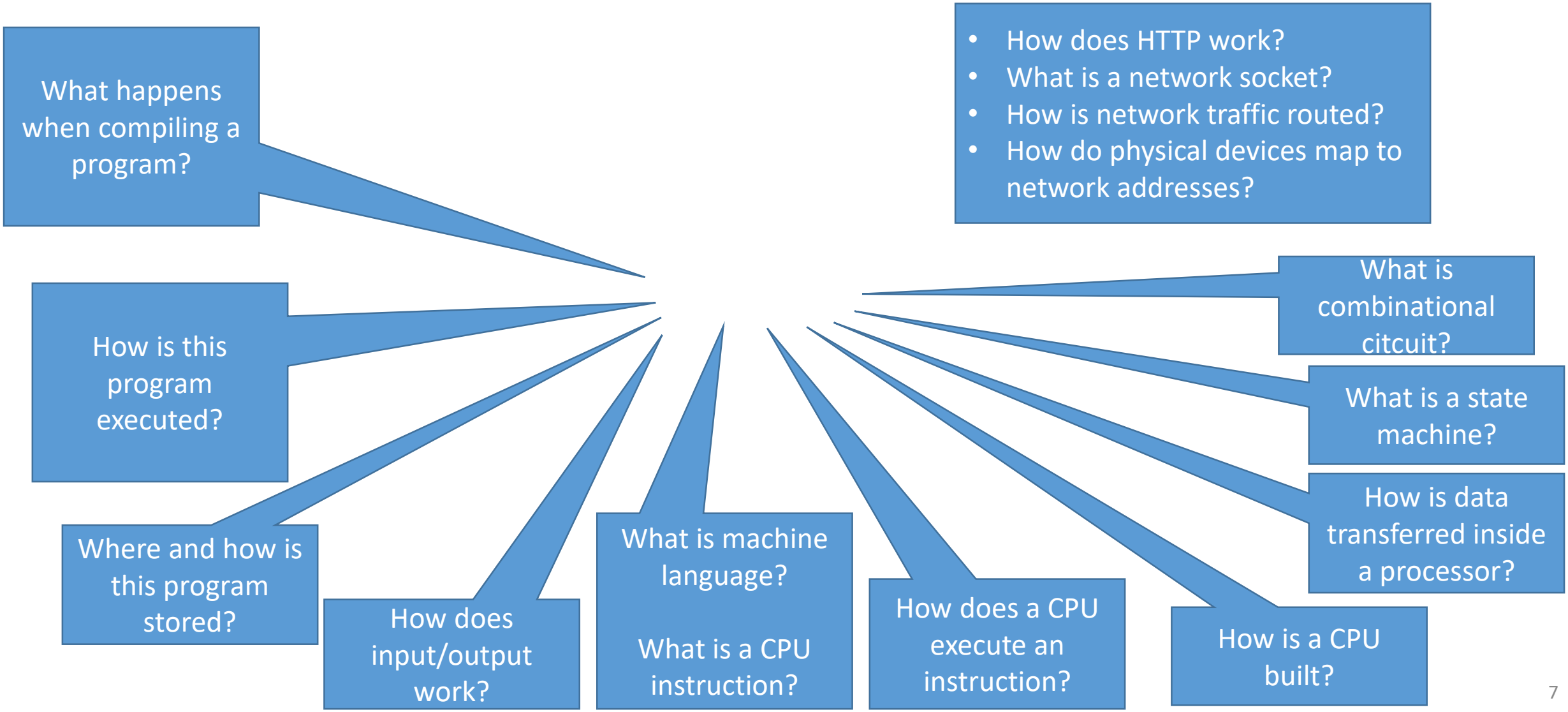
Content

What this course is about

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

```
include <stdio.h>

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

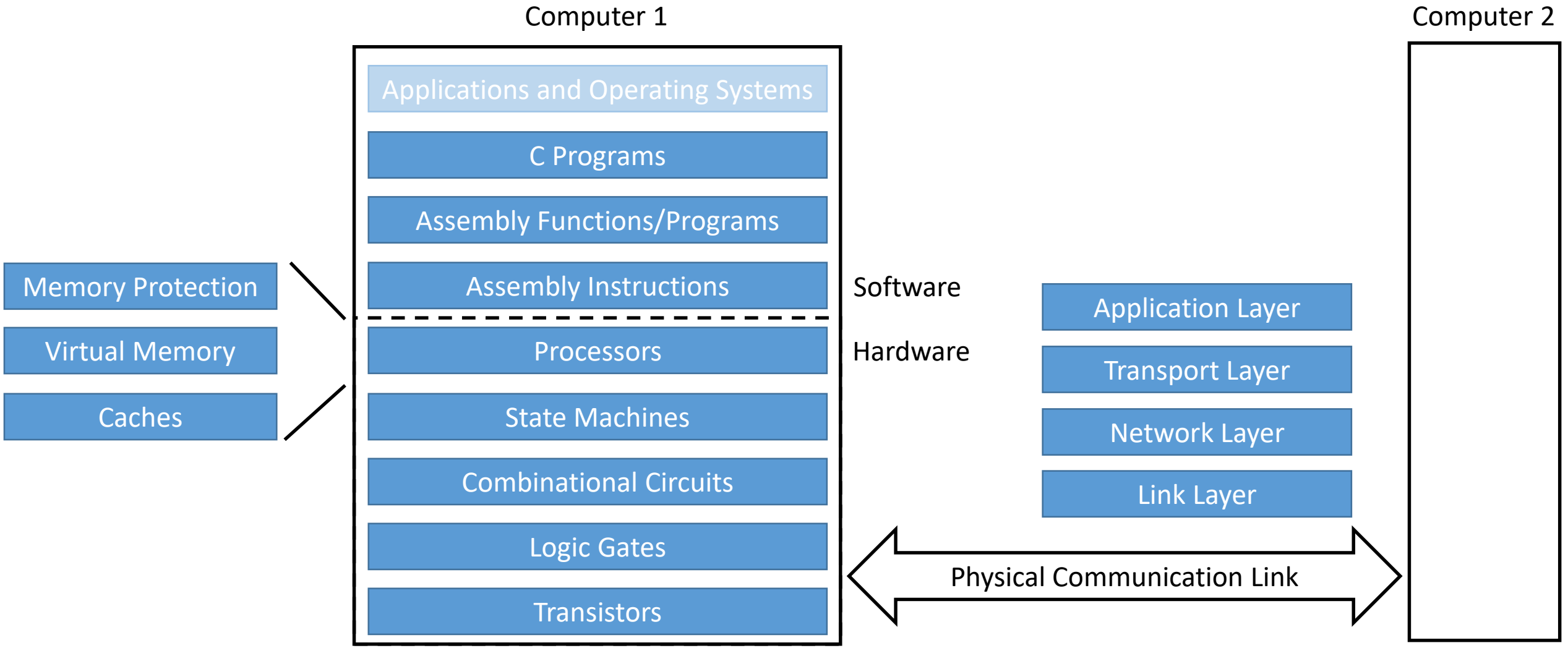
Hardware and Software - It's all one Thing



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



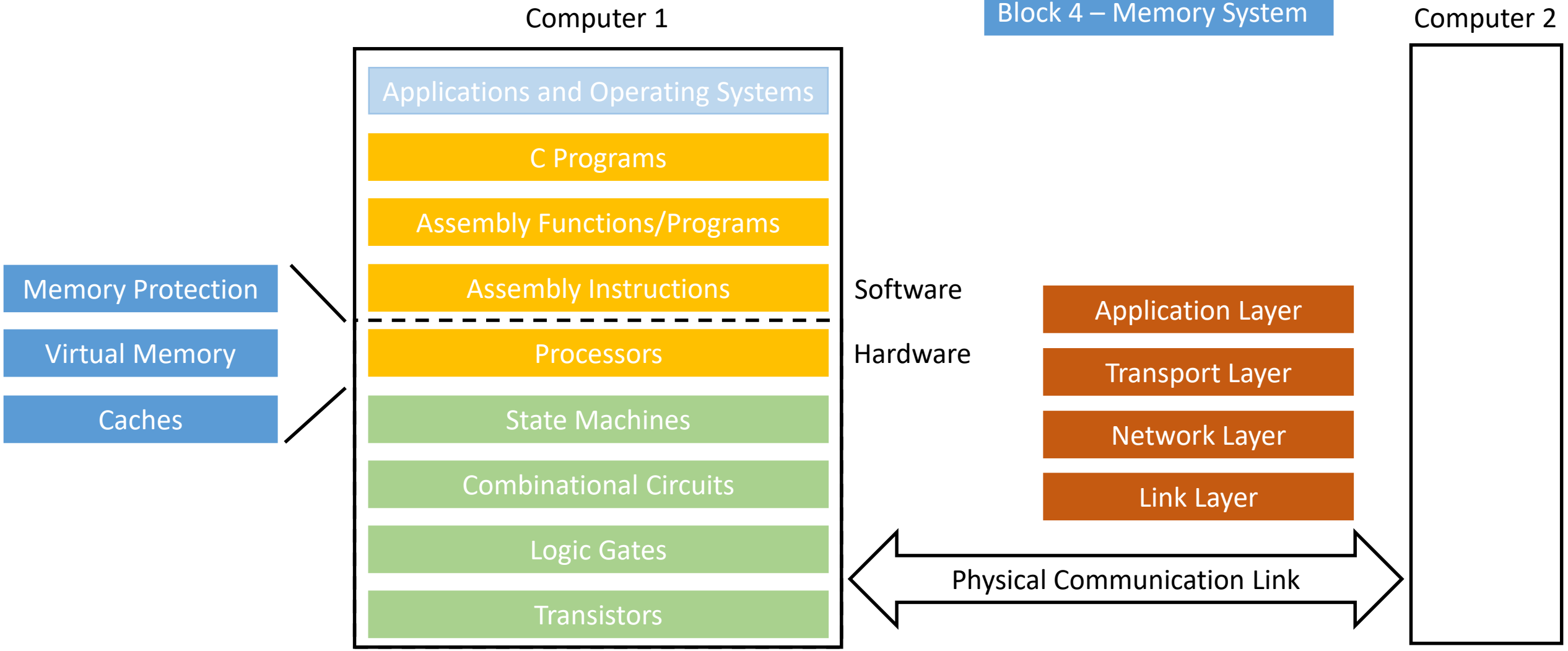
The Big Picture

Block 1 - Basics

Block 2 - Processors

Block 3 - Networks

Block 4 – Memory System



Goal

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

- Understand the specifications of your device
  - Which device does this spec belong to?

    - 64-bit six-core CPU implementing ARMv8.4-A ISA
    - Two high performance cores @2.65 GHz (Lightning), four energy efficient cores (Thunder)
    - 6 ALUs and three FP/vector pipelines
    - Lightning: 128 KiB L1I and 128 KiB L1D; Shared L2 with 8 MiB
    - TSMC 7nm, 8.5 billion transistors

ACM Turing Awards

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

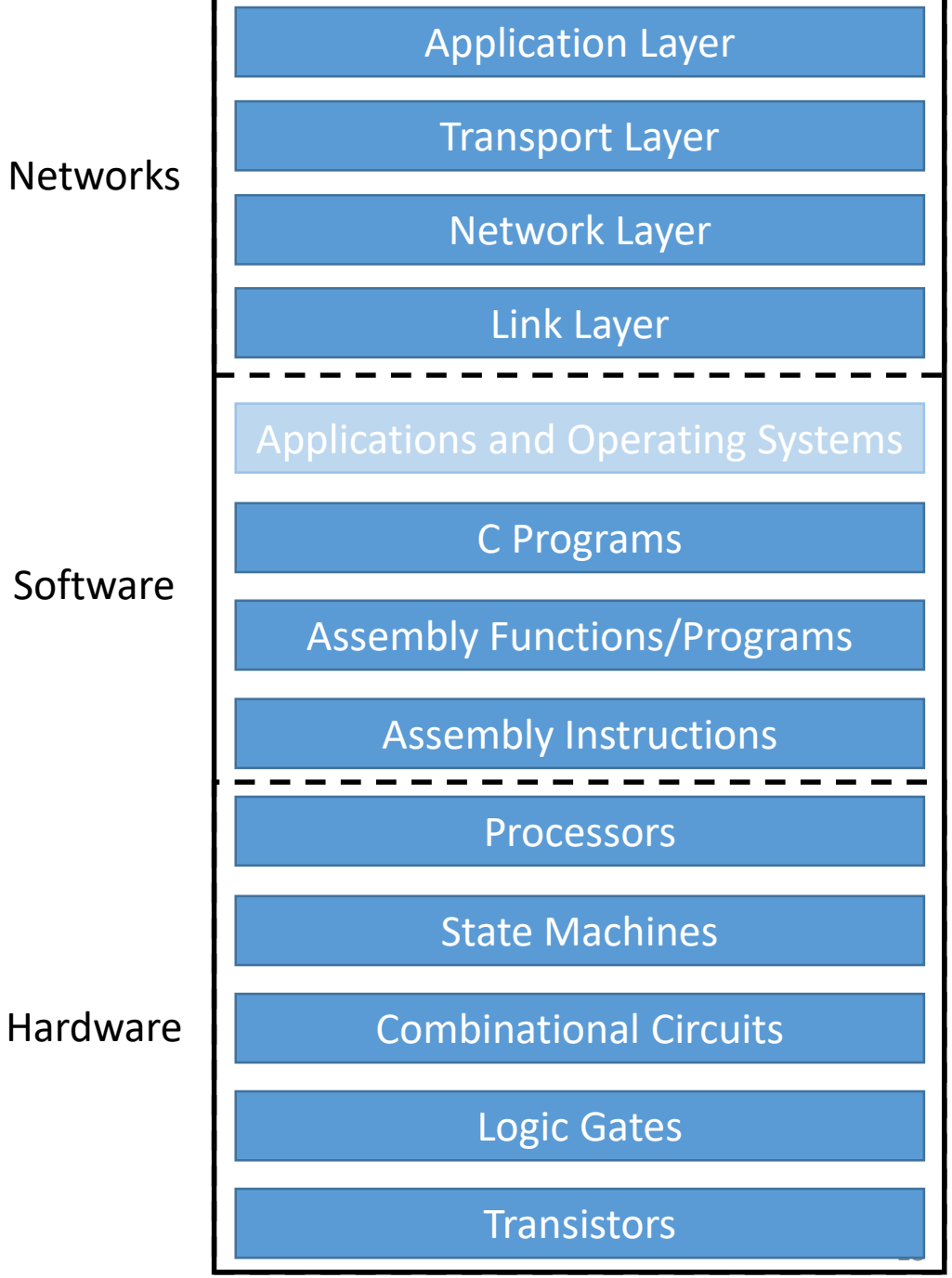
- David A. Patterson and John L. Hennessy received the Turing Award 2017 for their work on computer architectures and organization

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!

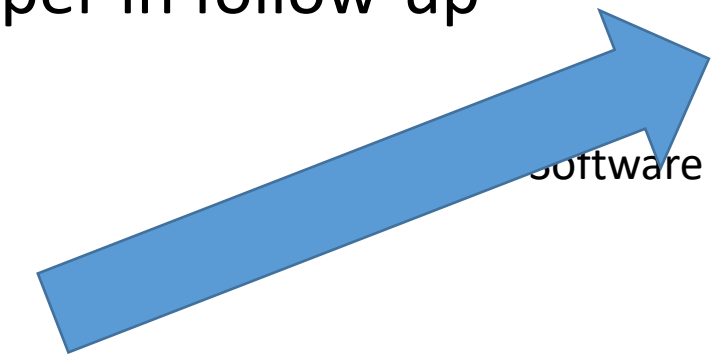


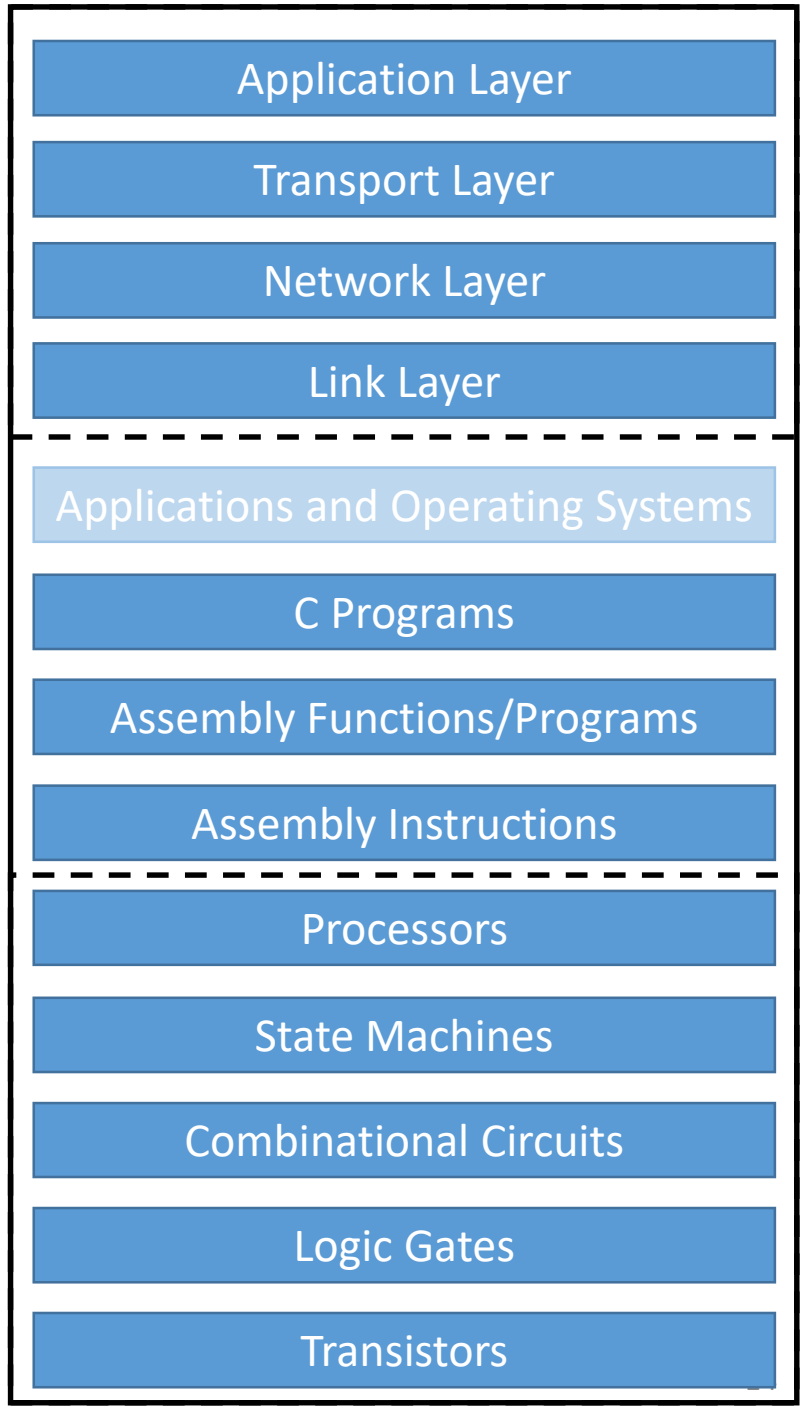
More?

- 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"





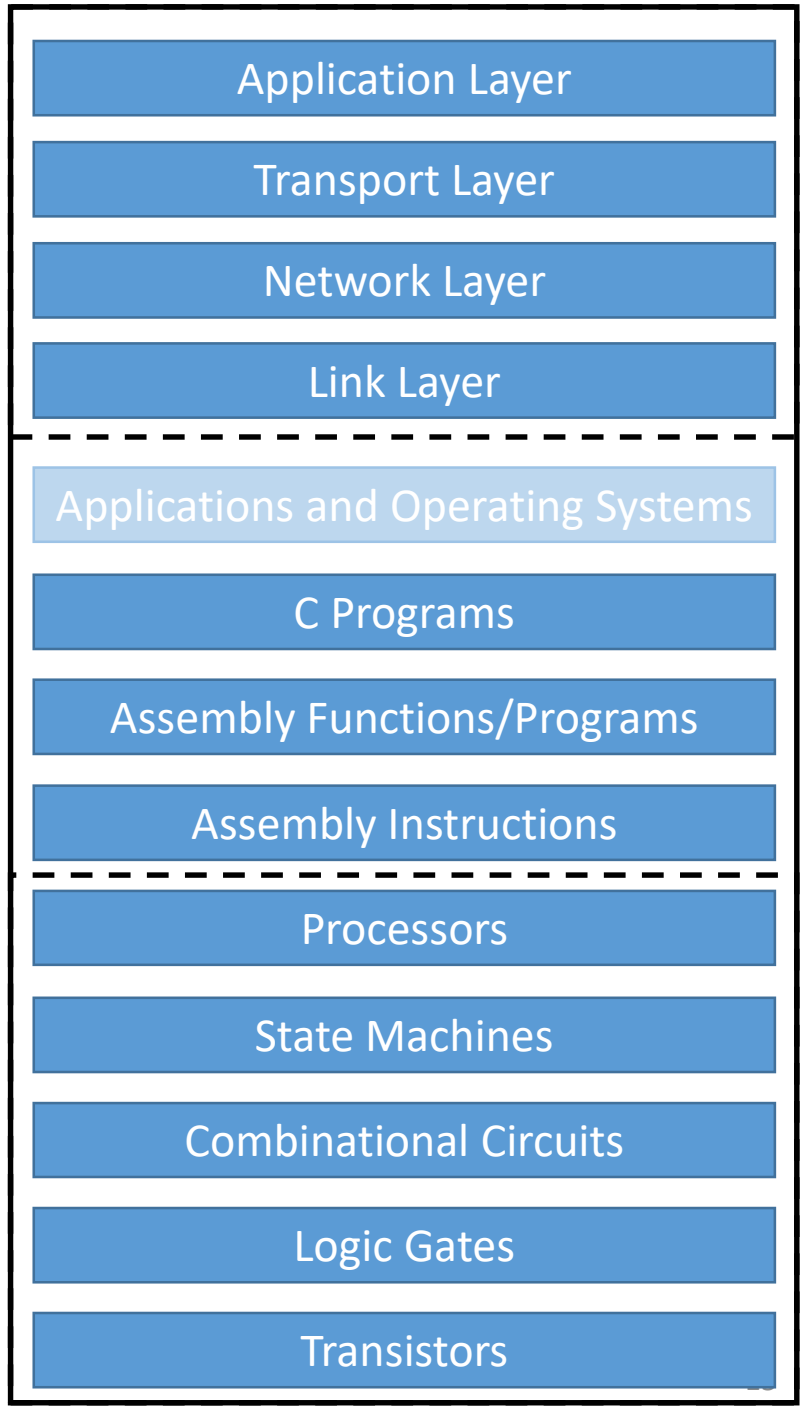
More?

- 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/

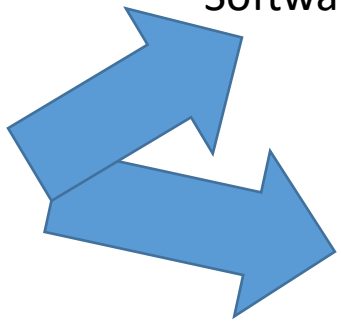


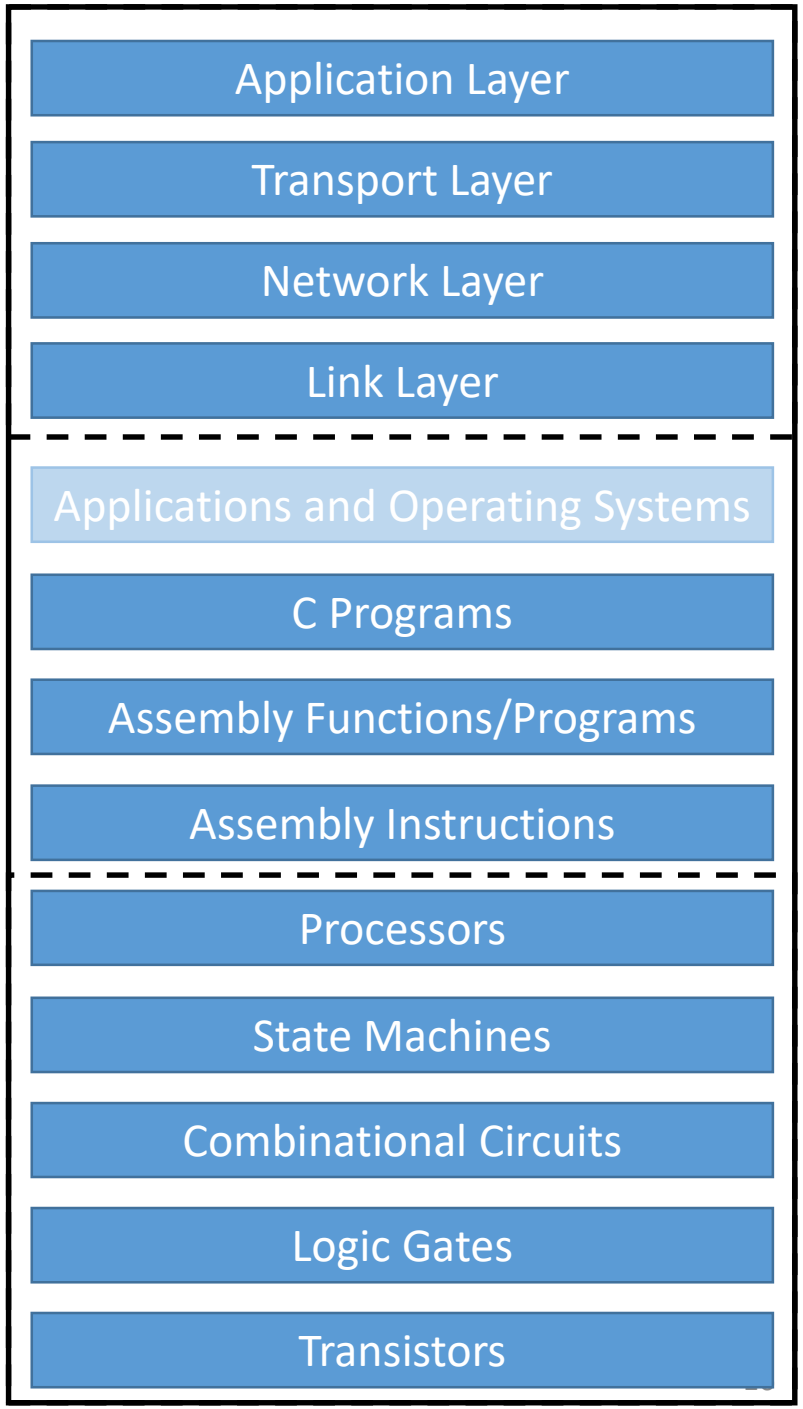
More?

- 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"



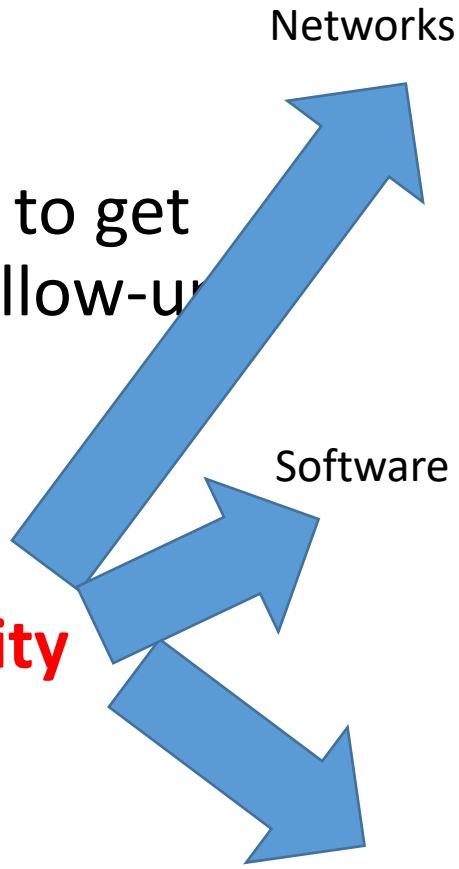


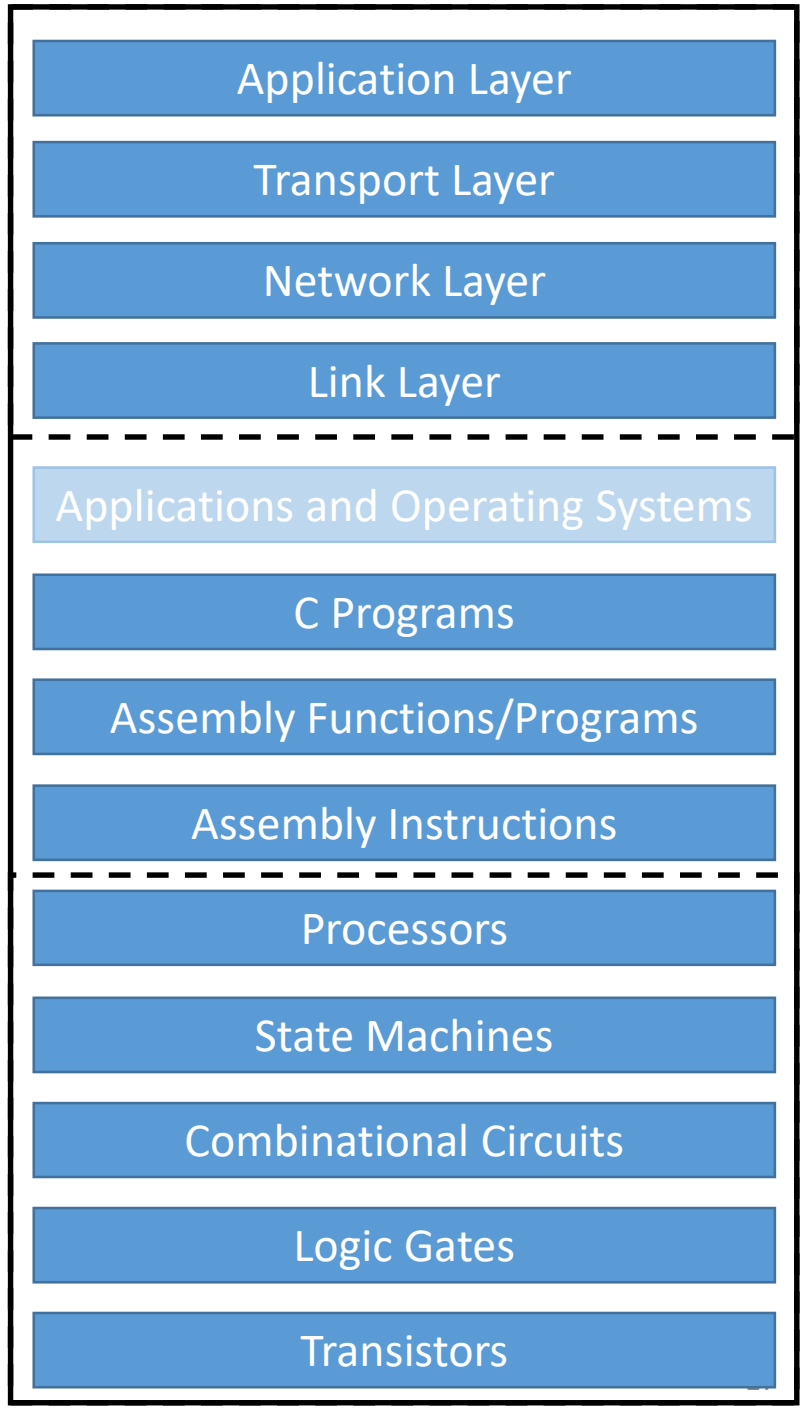
More?

- 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"





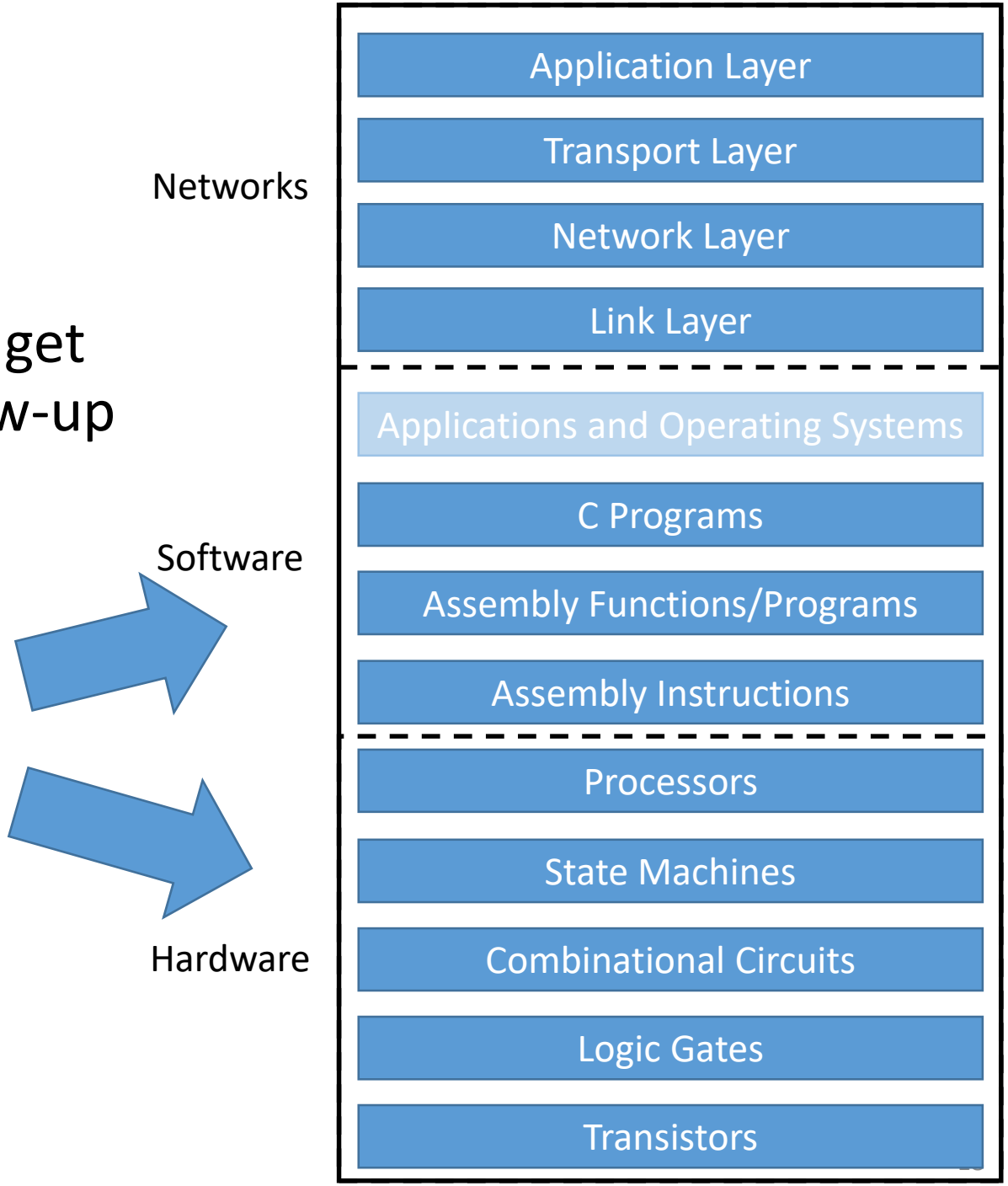
More?

- 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 (curriculum 2019)
  - 521 Computer Science (curriculum 2019)
  - 524 Software Engineering and Management (curriculum 2019)
- **Elective compulsory** course in **semester 3** for
  - 054, 414 Supplementary Bachelor's program Teacher Training: Secondary Schools (General Education), Subject: Informatics (curriculum 2019)
  - 198 Teacher Education Programme for Secondary Level (curriculum 2019)

Team

Stefan Mangard

Johannes Feichtner

Lukas Prokop

Robert Schilling

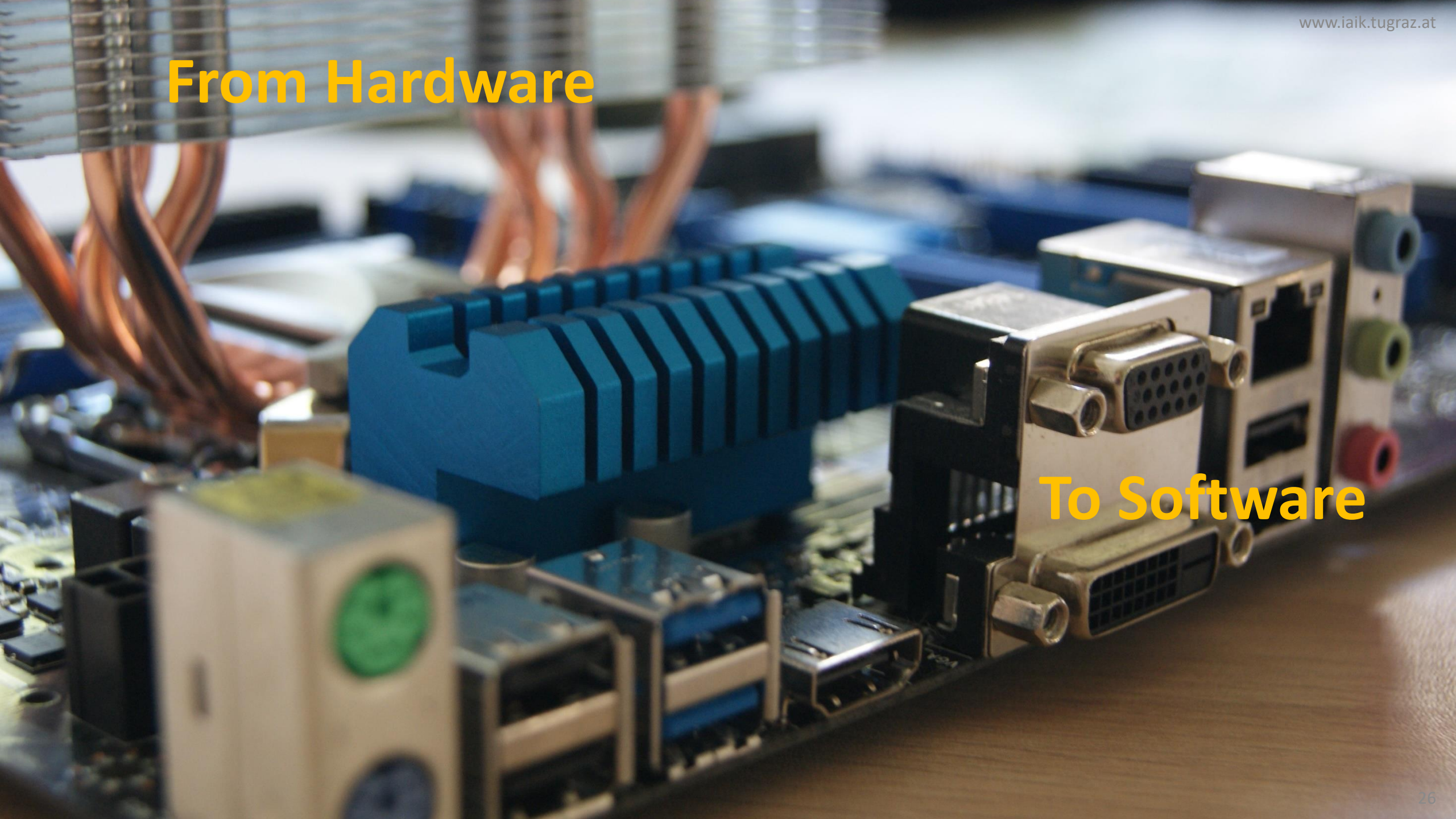
Teaching Assistants

- Ferdinand Bachmann
- Cagdas Baris Demirtas
- Nikolaus Grogger
- Richard Heinz
- Amir Mujacic
- Michael Neubauer
- Lukas Pertoll
- Crt Stajnko
- Martin Unterguggenberger
- Moritz Waser

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
- Newsgroup
  tu-graz.lv.con

Lecture


From Hardware
To Software

Lecture

- Location
  - Online in MS Teams (Backup via Youtube)

- 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-2020.git

Lecture Content and Timeline

- **Block 1: Basics (Stefan Mangard)**
  - Chapter 1: Combinational Circuits
  - Chapter 2: Number representation and arithmetic
  - Chapter 3: Finite State Machines

Block 1 - Basics

- **Block 2: Processors (Stefan Mangard)**
  - Chapter 4: Basics of Processor Design
  - Chapter 5: Pipelining
  - Chapter 6: Pipelining Issues
  - Chapter 7: Hardware/Software Contract, Stack

Block 2 - Processors

- **Block 3: Networks (Johannes Feichtner)**
  - Chapter 8: Network Basics
  - Chapter 9: Network Layer
  - Chapter 10: Transport Layer
  - Chapter 11: Application Layer

Block 3 - Networks

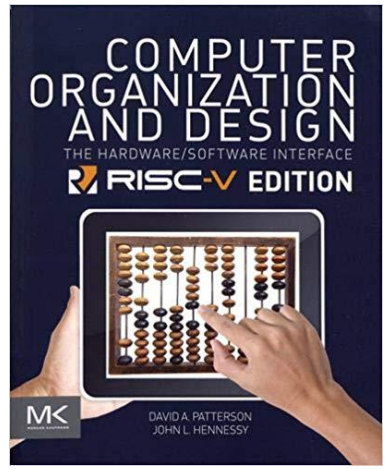
- **Block 4: Memory System (Stefan Mangard)**
  - Chapter 12: Caches
  - Chapter 13: Virtual Memory
  - Chapter 14: Security

Block 4 – Memory System

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/



- Textbook:
  - Computer Organization & Design: The Hardware/Software Interface (David A. Patterson / John L. Hennessy)
  - We partly cover the content of this book

Practical

Tasks

Deadline	Topic	Toolchain	Points
23.10.2020	AddSub Machine	Logisim	15
06.11.2020	PicoRISC-V CPU	SystemVerilog	20
20.11.2020	Multiplier incl. CPU Integration	SystemVerilog	20
04.12.2020	XGCD	RISC-V Assembly	15
18.12.2020	Ping me	C/C++	15
22.01.2021	Talk UDP to me	C/C++	15

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 for each task
- There is a video tutorial for each assignment (+ extra video tutorial for SystemVerilog)
- All tutorials take place online via Discord → organized as Q & A sessions
- There is approximately two weeks for each assignment

Assignment

- PDF Assignment for Task 1 is online

- Video tutorials for the next task will be published no later than with the deadline of the task before

Publication Date	Tutorial Video Content
09.10.2020	Git and Logisim
23.10.2020	SystemVerilog: toolflow, example implementations
06.11.2020	SystemVerilog: finite state machines
20.11.2020	RISC-V assembly and calling conventions
04.12.2020	C and network basics
18.12.2020	Network protocols

Question hours

- 10 groups, 10 teaching assistants (TAs)
- Question hours start next week (13.10.2020)
- Weekly question hours specific for each group

	Tuesday	Wednesday	Thursday
08:00	Richard, Michael	Nikolaus	Amir
09:00	Cagdas, Martin		
10:00			
11:00		Črt	
12:00		Lukas	
13:00			Ferdinand
14:00			Moritz

Submissions

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

Have you worked with git before?

Interviews

- Interviews will happen on Discord (microphone needed!)
- TA will pick you up from #con-waiting-room

Week	Interview scope
23–27.11.2020	Task 1–3
25–29.01.2021	Task 4–6

Resources

- Course Web:
  - https://www.iaik.tugraz.at/con
- Virtual machine with all tools installed
  - https://seafile.iaik.tugraz.at/f/1c53add7a9bb4af3afe3/
- Upstream Repository and Assignment:
  - https://extgit.iaik.tugraz.at/con/practicals-2020
- Newsgroup:
  - tu-graz.lv.con
- Tutorials:
  - Regularly, almost every week

Plagiarism

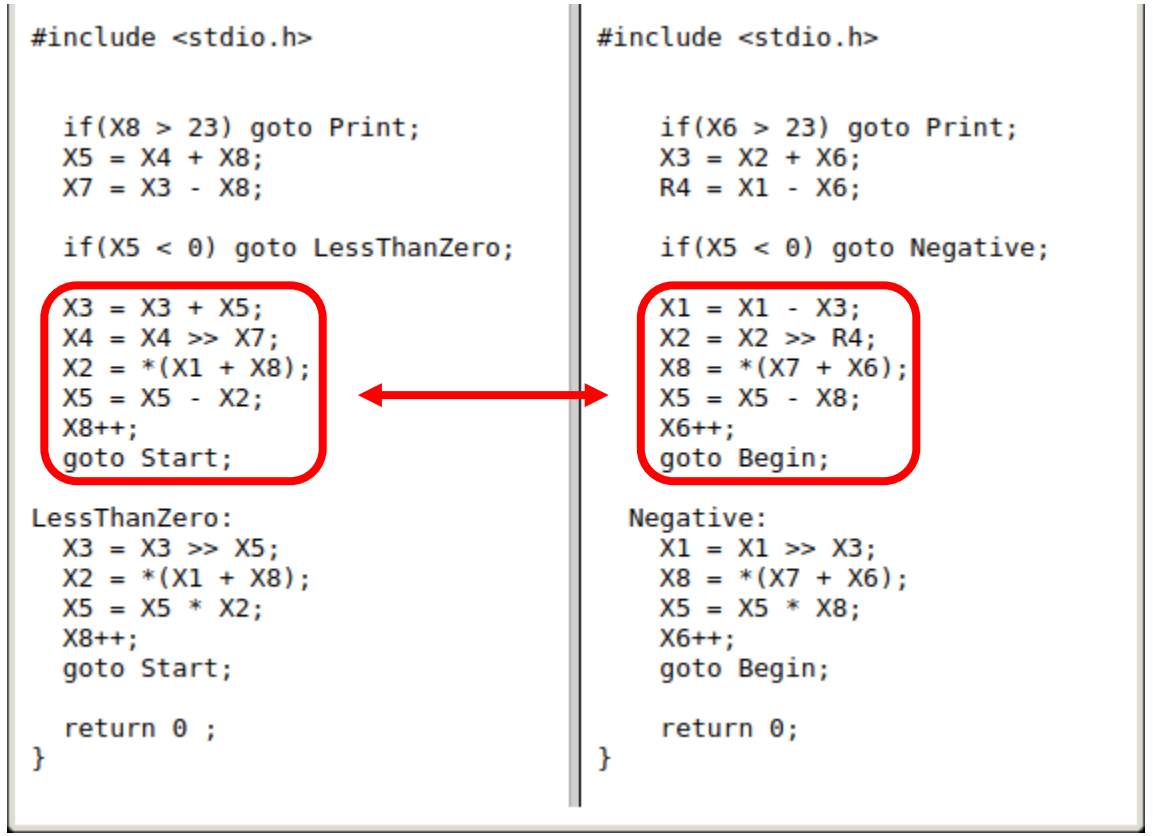
- **We perform plagiarism checks!**
- **All involved people** receive the grade “U (Ungültig / Täuschung)”
  - We will not invest time on researching who copied from whom
- If you plagiarize parts of the program, it is still a case of plagiarism
- How to avoid plagiarism?
  - **Do not share code!**
  - 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 for one of the groups in TUGRAZonline **(deadline: this week)**
- Read the assignment sheet
- Install the development environment and get going
- Clone the upstream repository
- Watch the video tutorial for assignment 1
- Attend the online tutorials next week

Effort

- This is a 7 ECTS course – this is approx. 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”)