

# SYSTEM MODELLING AND SYNTHESIS WITH HDL

DTEK0078

2023 Lecture 1



UNIVERSITY  
OF TURKU

# Course Information

- Lecturer:
  - Professor Tomi Westerlund  
[tomi.westerlund@utu.fi](mailto:tomi.westerlund@utu.fi)
- Credits: 5 ects
- Course Requirement:
  - Exercises
- Pre-Requirements: none
- Helpful knowledge: digital system design,  
basic knowledge of HDL and  
object-oriented programming

# Moodle

- Course name is

**System Modelling and Synthesis with HDL S2023**

- Course id in Moodle is

**14534**

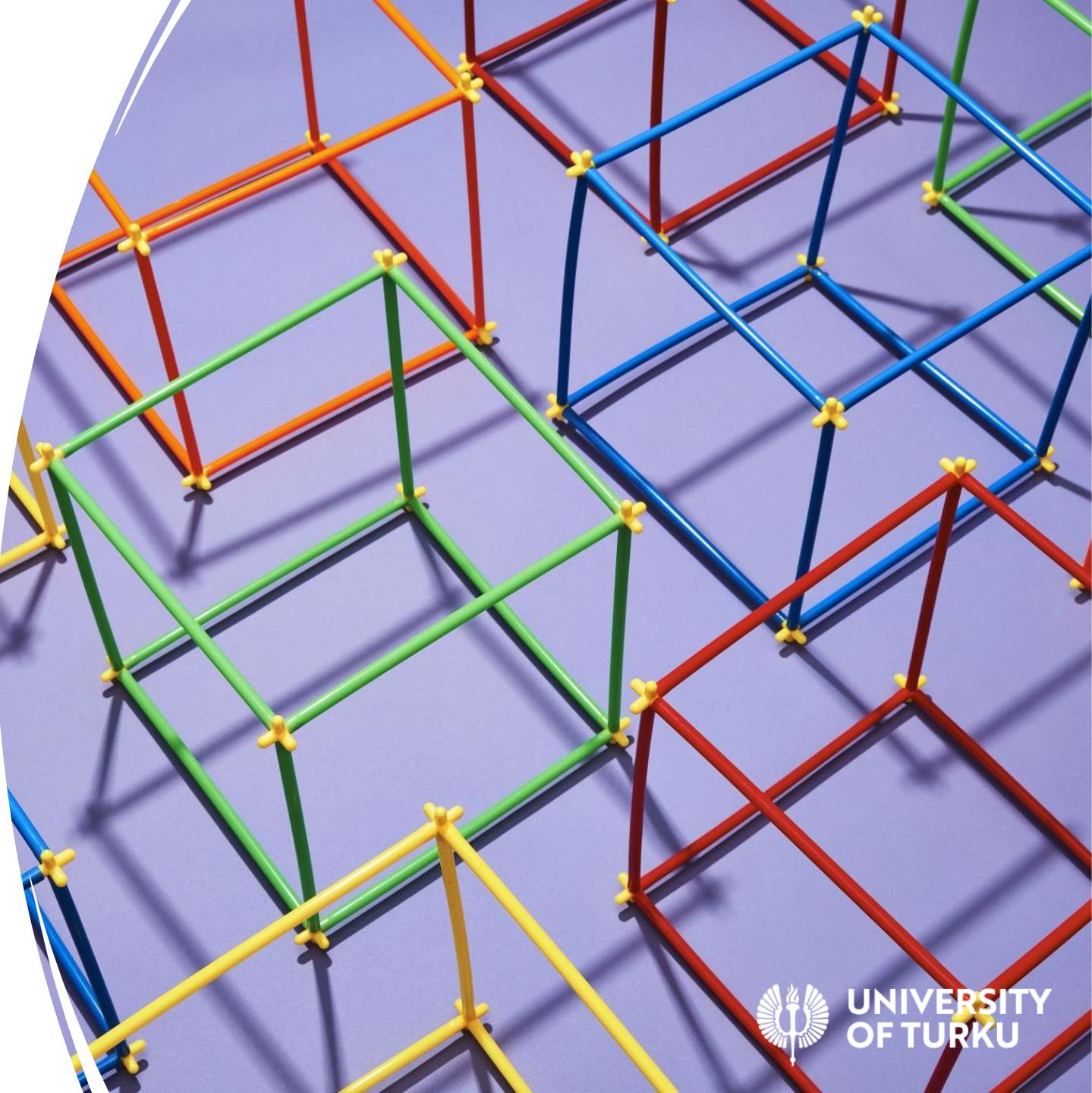
- Enrollment key is

**smsHDL23**

# Lectures

---

- Intro, modelling, very basic VHDL syntax
- Main building blocks, Coding Styles, objects and data types,
- Hierarchical design, components, testbenches
- Statements, processes, control structures
- Statements, signals
- Simulation, timing, synchronous design,
- Storage
- Finite state machines, generics
- Routines (functions and procedures)



# Lecture Schedule

- Lectures in the beginning twice a week on Tuesdays and Thursdays
  - Tue 14-16, 110C
  - Wed 30.8. 12-14, 110C
- Detailed schedule and possible changes in Moodle
- We have more lectures in the beginning less afterwards

DTEK0078-3004 System Modelling and Synthesis with HDL

01.08.2023 - 17.12.2023

## Teacher

Tomi Westerlund

## Teaching language

English

## Enrollment time

01.08.2023 00:00 - 22.08.2023 23:59

## Learning materials and recommended literature

Lecture slides, instructions for exercises, reference material

## Group

- Robotics and Autonomous Systems (DI)

## Teaching

Tue 29.08.2023 14:15-16:00 110C, Agora 110C  
Wed 30.08.2023 12:15-14:00 110C, Agora 110C  
Thu 31.08.2023 14:15-16:00 110C, Agora 110C  
Tue 05.09.2023 14:15-16:00 110C, Agora 110C  
Wed 06.09.2023 12:15-14:00 110C, Agora 110C  
Thu 07.09.2023 14:15-16:00 110C, Agora 110C  
Tue 12.09.2023 14:15-16:00 110C, Agora 110C  
Wed 13.09.2023 12:15-14:00 110C, Agora 110C  
Thu 14.09.2023 14:15-16:00 110C, Agora 110C  
Tue 19.09.2023 14:15-16:00 110C, Agora 110C  
Wed 20.09.2023 12:15-14:00 110C, Agora 110C

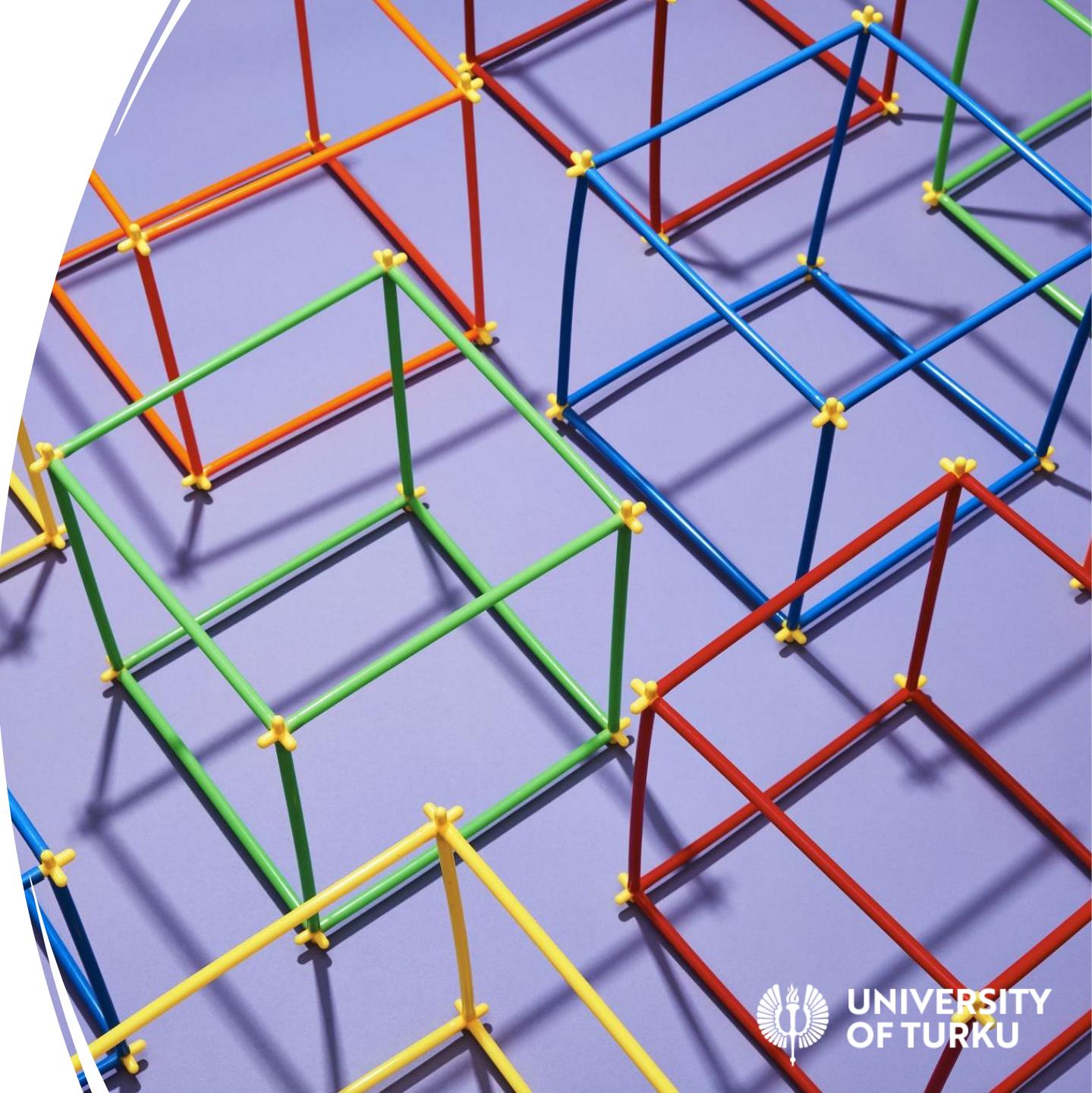


UNIVERSITY  
OF TURKU

# Labs

---

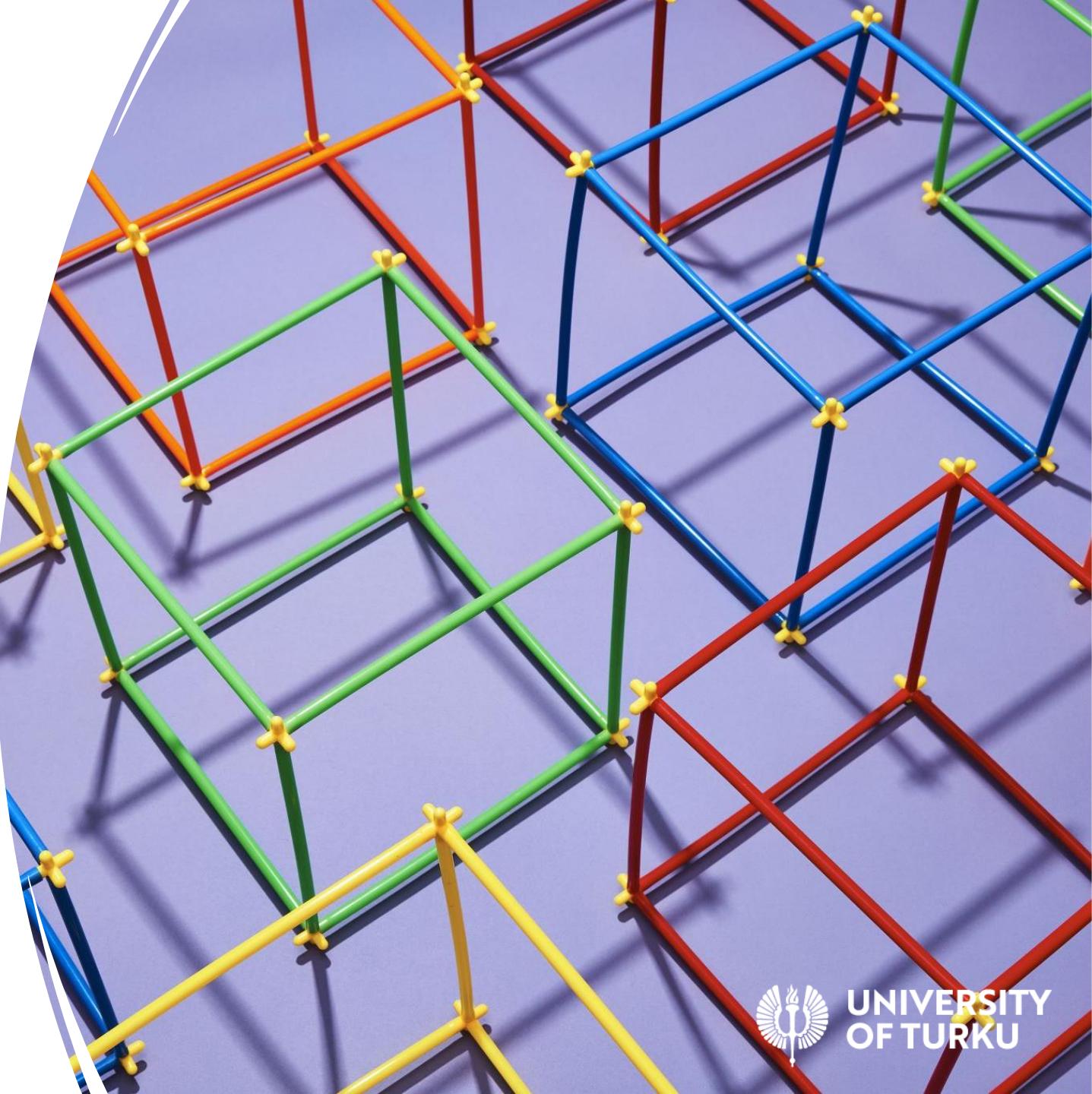
- Modelling Concepts
- Encoders, Decoders and Memories
- Behavioural and Timing Constraints
- Latch and FlipFlop
- Registers and Counters
- Finite State Machines
- Functions and Procedures



# Exercises

---

- All exercises are obligatory
- In total 5 exercises
  - EDIT: might be 5-7 but keeping the workload the same
  - Only affect exercise deadlines



# Exercise Schedule

- Starts on Wednesday 13.9.
  - 12-16 in Agora 110C
- Exercises will be assessed  
-> grade of the course
- Always remember to comment on your code
  - Insert comment blocks on files, classes and class members (variables and member functions)
  - Comment should describe the intent of the code

DTEK0078-3004 System Modelling and Synthesis with HDL

01.08.2023 - 17.12.2023

## Teacher

Tomi Westerlund

## Enrollment time

01.08.2023 00:00 - 22.08.2023 23:59

## Group

- Robotics and Autonomous Systems (DI)

## Teaching language

English

## Learning materials and recommended literature

Lecture slides, instructions for exercises, reference material

## Teaching

Tue 29.08.2023 14:15-16:00 110C, Agora 110C  
Wed 30.08.2023 12:15-14:00 110C, Agora 110C  
Thu 31.08.2023 14:15-16:00 110C, Agora 110C  
Tue 05.09.2023 14:15-16:00 110C, Agora 110C  
Wed 06.09.2023 12:15-14:00 110C, Agora 110C  
Thu 07.09.2023 14:15-16:00 110C, Agora 110C  
Tue 12.09.2023 14:15-16:00 110C, Agora 110C  
Wed 13.09.2023 12:15-14:00 110C, Agora 110C  
Thu 14.09.2023 14:15-16:00 110C, Agora 110C  
Tue 19.09.2023 14:15-16:00 110C, Agora 110C  
Wed 20.09.2023 12:15-14:00 110C, Agora 110C



UNIVERSITY  
OF TURKU

# Exercises

- In the exercises, we will use Xilinx's tools and FPGA development boards. Therefore, **download Vivado ML Standard Edition (free)**
- <https://www.xilinx.com/products/design-tools/vivado/vivado-ml.html>



# Passing the course

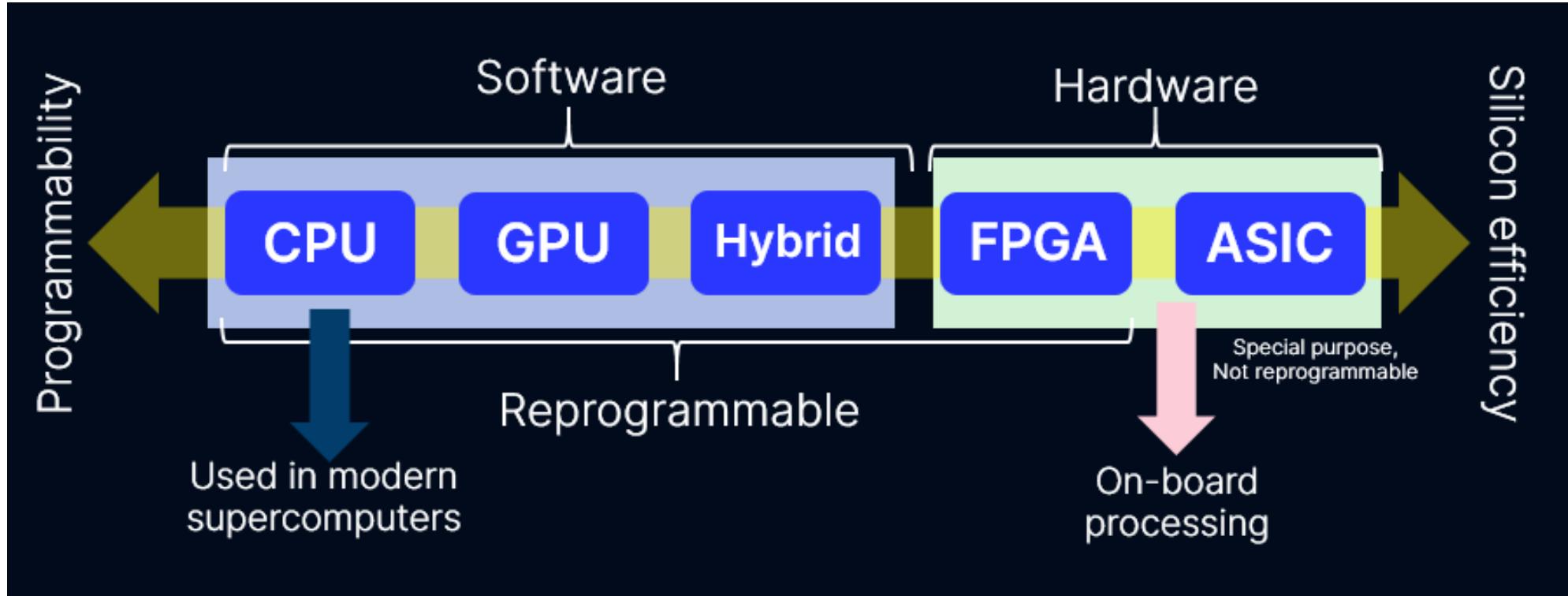
- Do ALL the exercises  
(NO examination)



# SYSTEM MODELLING AND SYNTHESIS WITH HDL

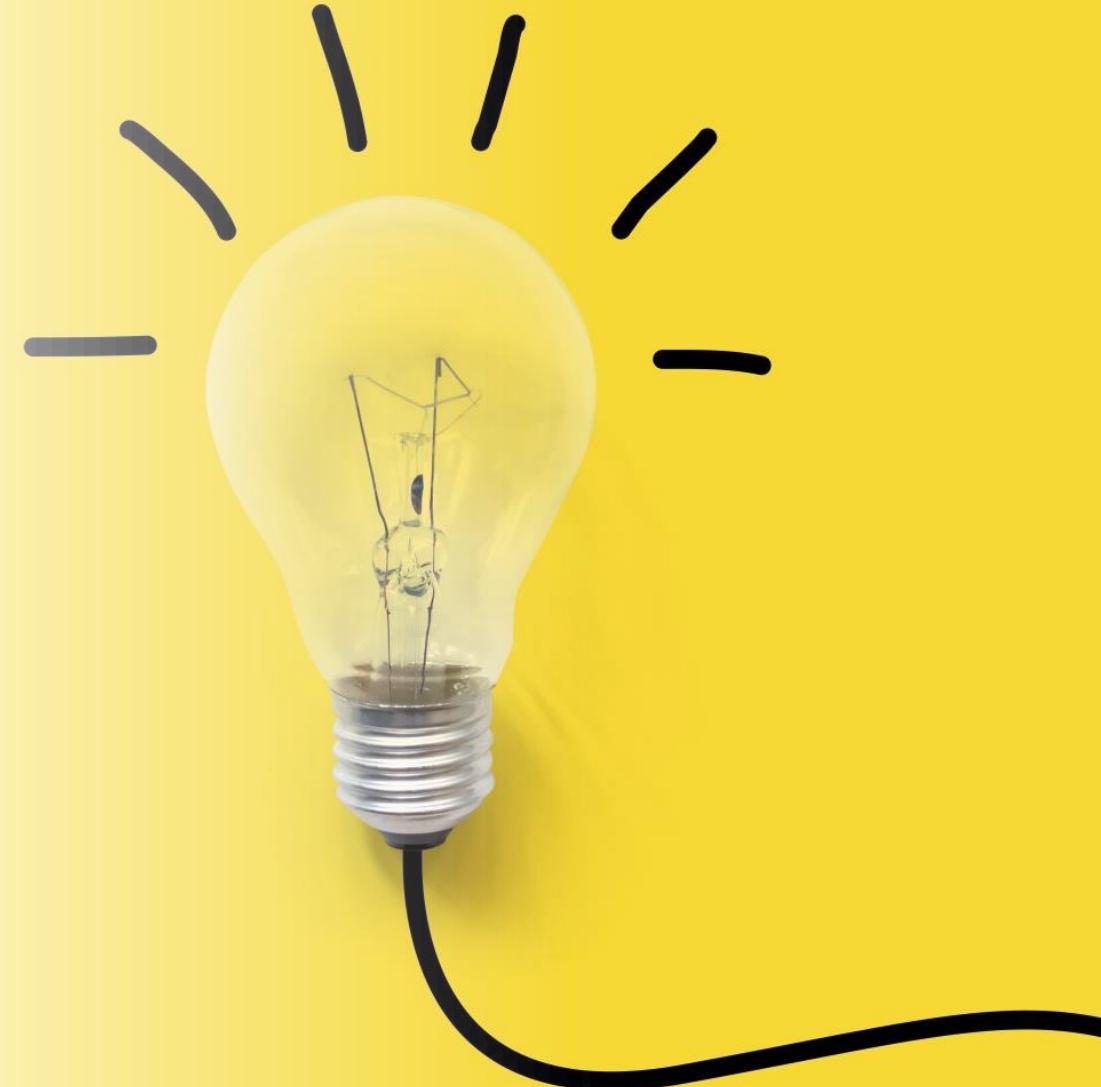


UNIVERSITY  
OF TURKU



# Efficiency

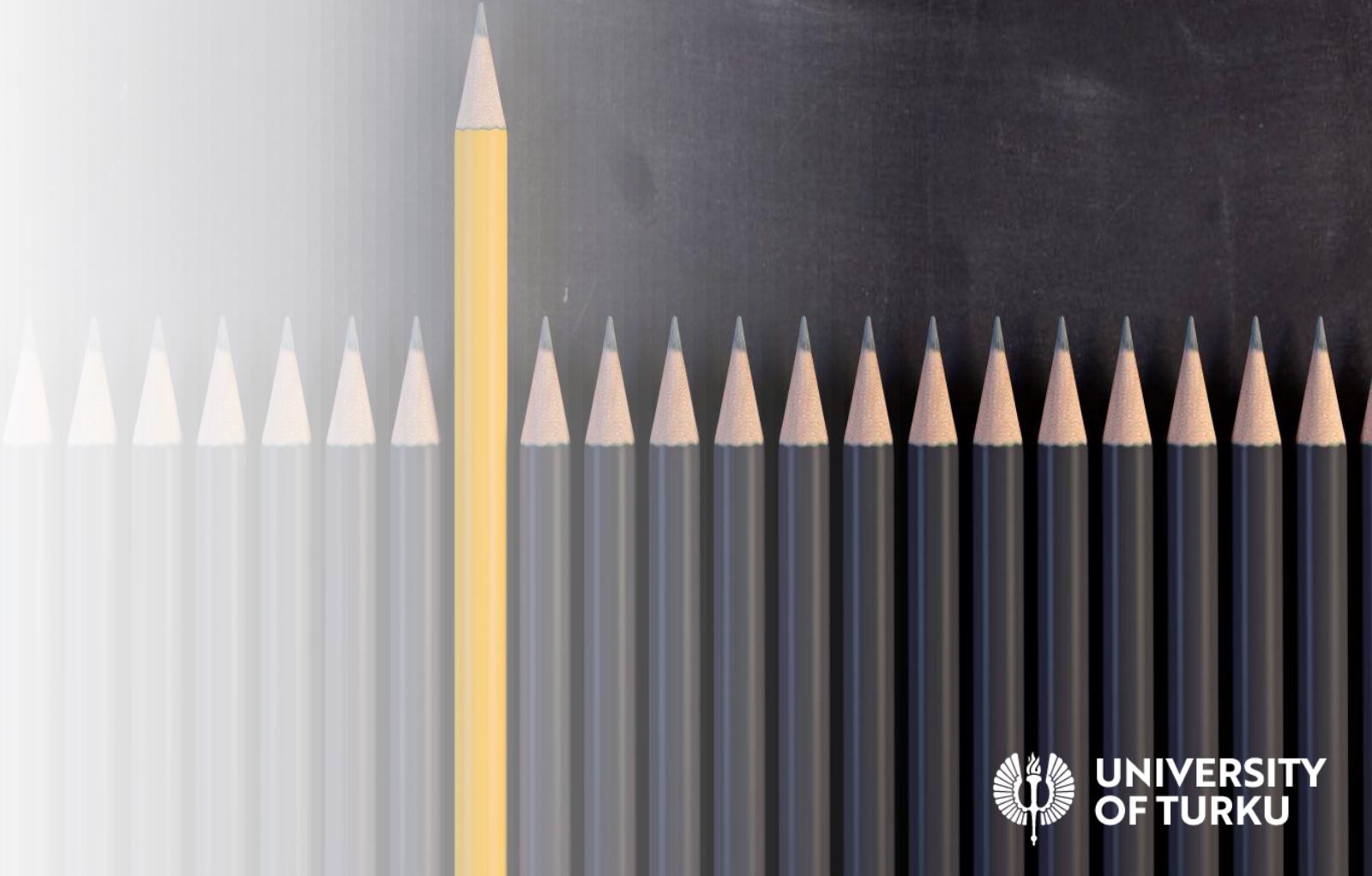
Efficiency means  
making good use of  
scarce resources



# Quality

---

The quality of the code  
we write has a deep  
influence on the quality  
of the finished system



# Design Qualities

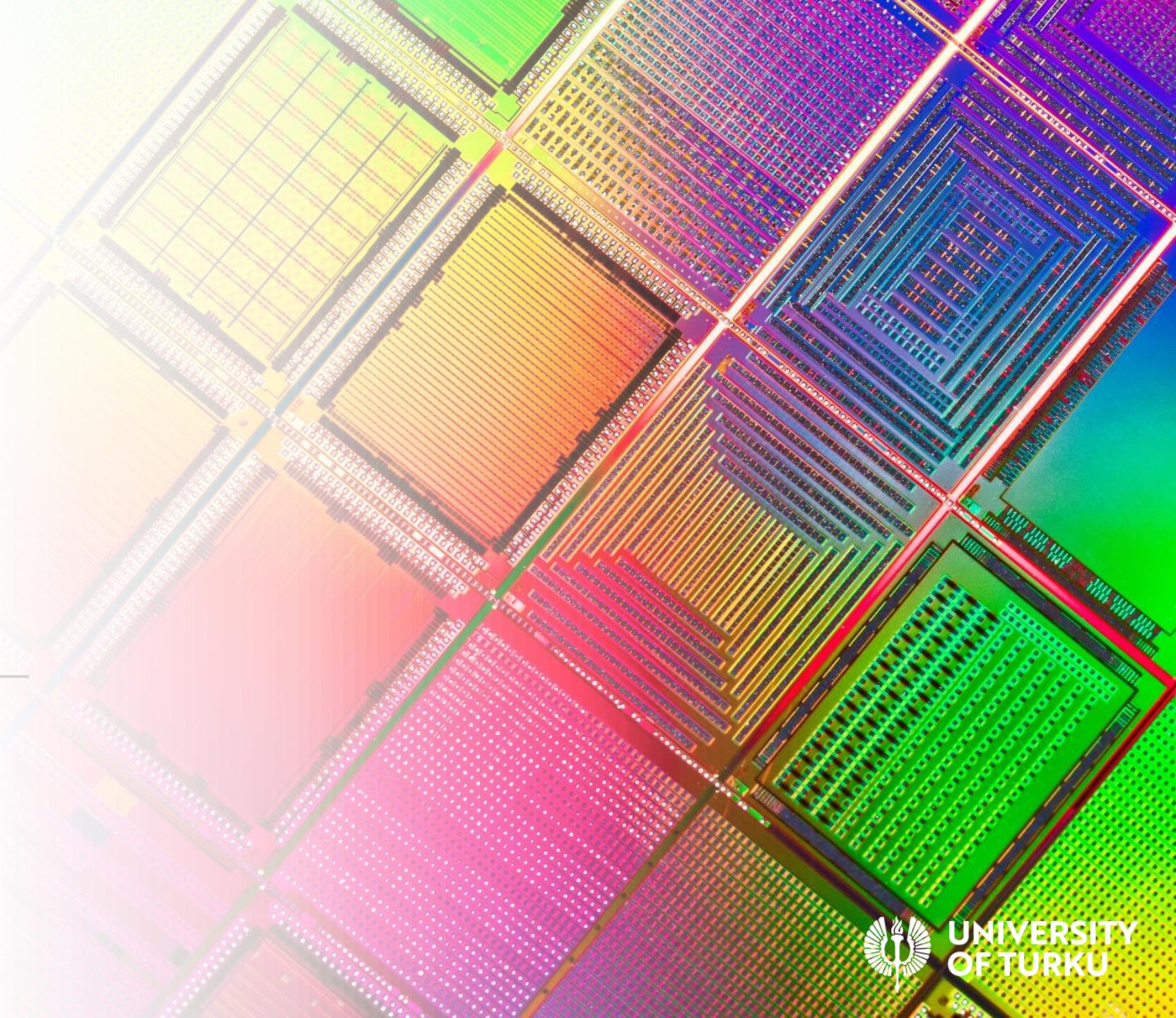
- Simplicity
  - It does not come naturally; it must be designed
- Understandability
  - In every realistic situation, code is read much more often than it is written
  - The clever code we write today may be hard to understand tomorrow
- Modifiability
  - A highly modifiable system has two desirable characteristics: it enables localized changes, and it prevents ripple effects
- Testability
  - There is a strong correlation between ease of testing and good design

# Design Qualities

- Extensibility
  - How easy it is to accommodate new functionality in a system
- Reusability
  - A measure of how easy it is to use the code in a place other than where it was designed for
- Portability
  - How easy it is to migrate a system to a different environment
  - To increase portability, do not use nonstandard language features
- Maintainability
  - Combined result of other qualities, such as modifiability, understandability, extensibility and portability

# Design and Modelling

---



UNIVERSITY  
OF TURKU

# What is Design?

- As a verb, **to design** means to create a plan for turning specifications into a working solution
- As a noun, **design** can mean either a set of such plans or the way in which something has been made
  - the sum of all design decisions made while creating the solution
- We need to do some design before coding because a good plan improves the odds of achieving our goals
  - The quality of the source code deeply influences the design qualities of the finished system

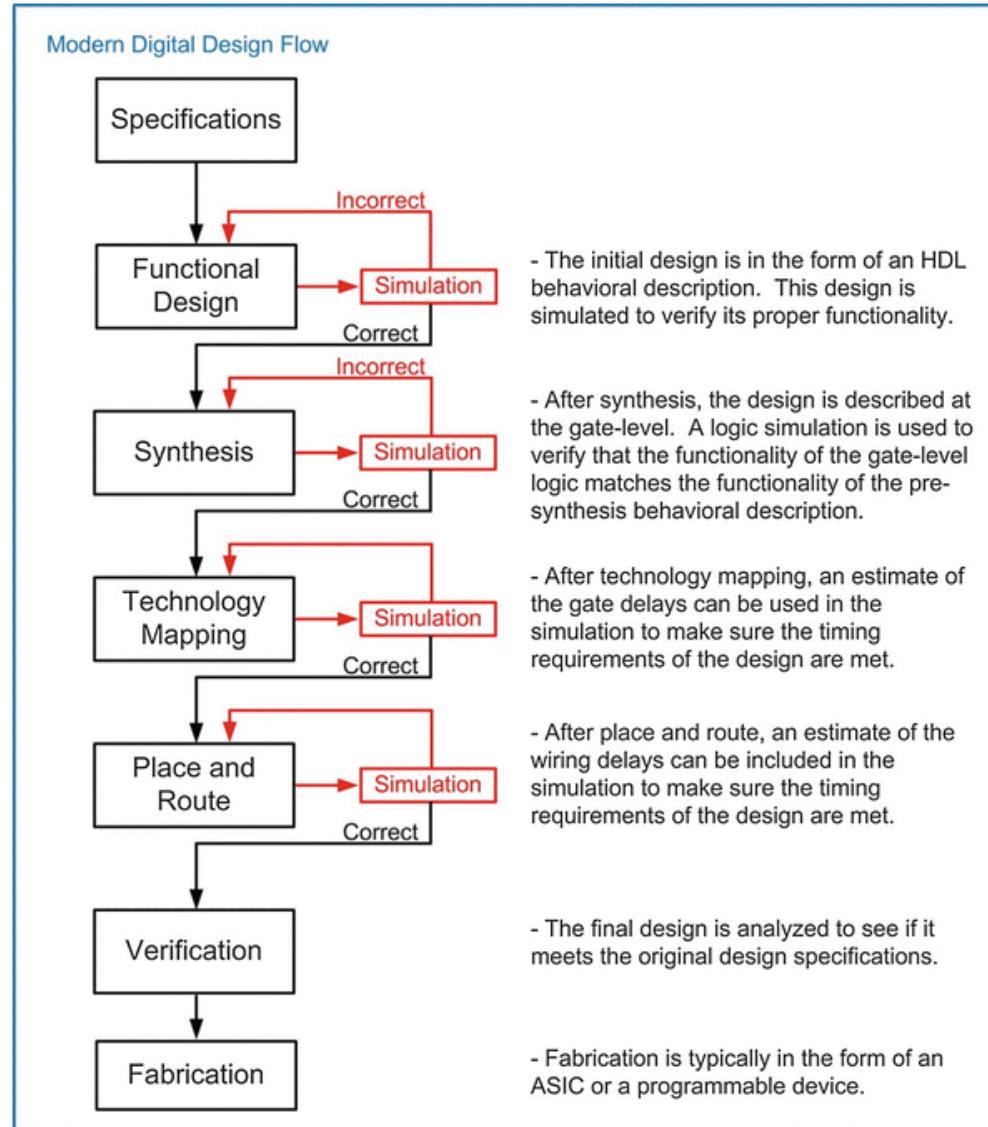
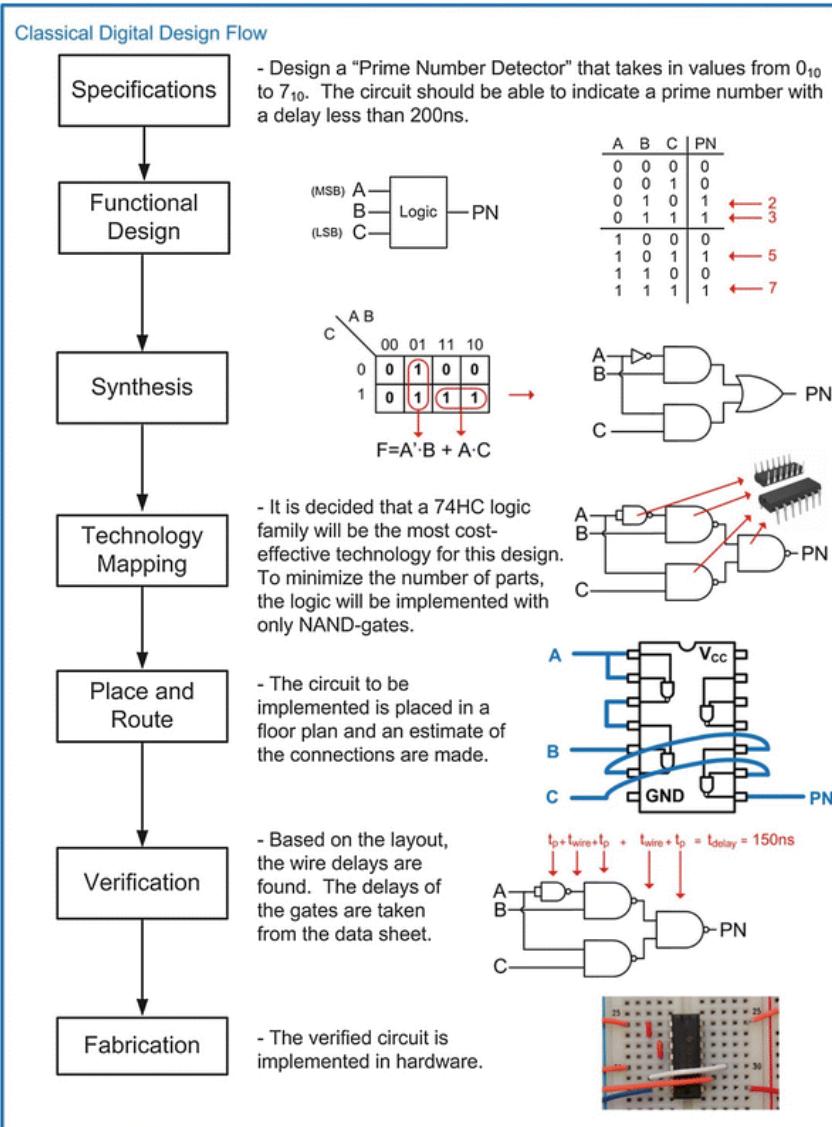
# In Digital System Development ...

- ... you can have a role as a
  - Design Engineer
- 
- Verification Engineer

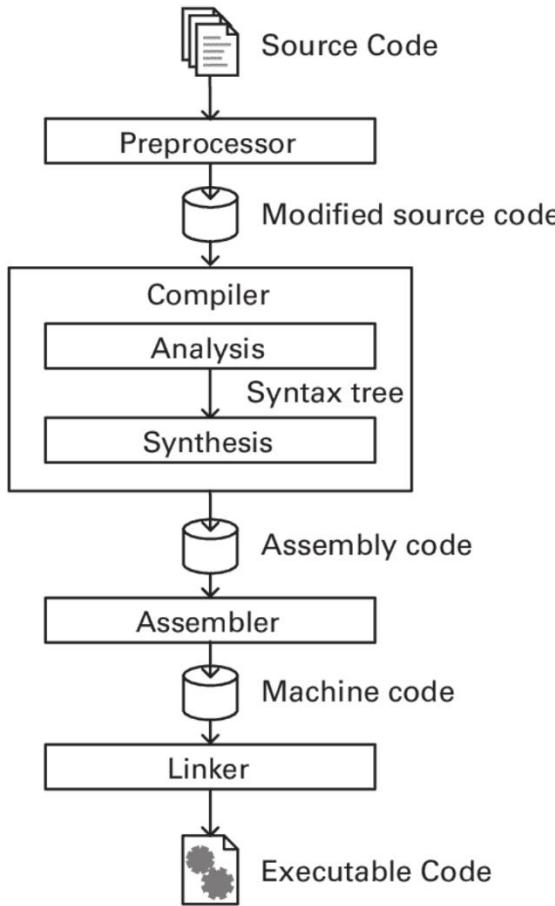
# In Digital System Development ...

- ... you can have a role as a
- Design Engineer
  - To create a device that performs a particular task based on a design specification
- Verification Engineer
  - To ensure that the model / design works correctly and successfully based on a design specification

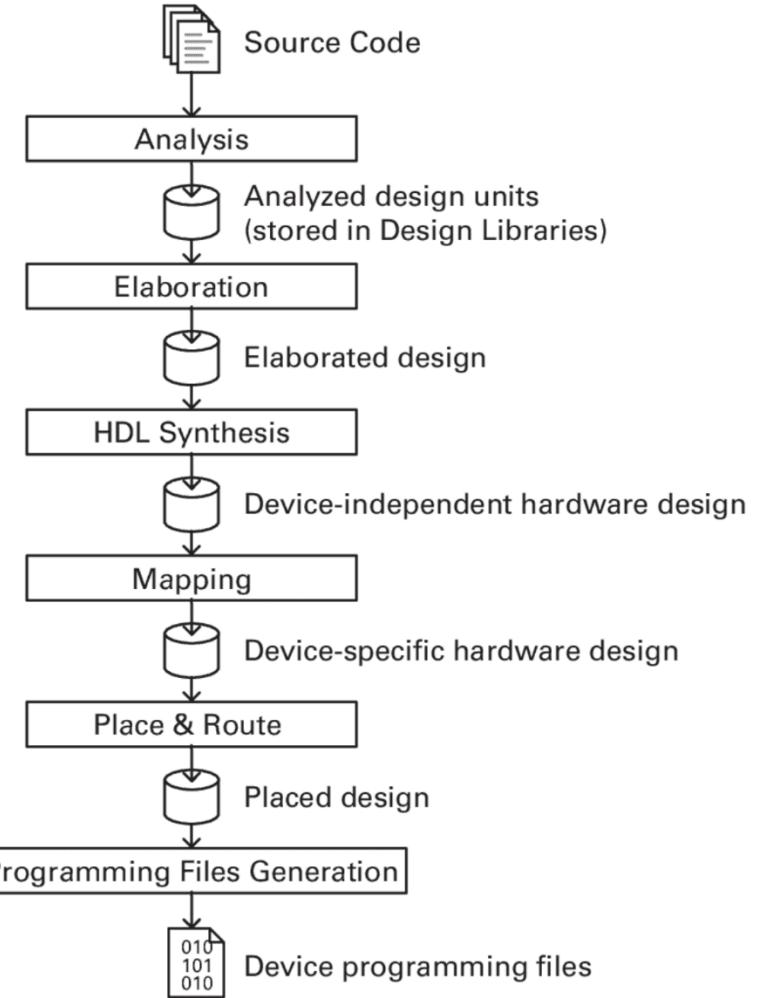
# Digital Design Flow



# Digital Design Flow



(a) Software case  
(programming language).

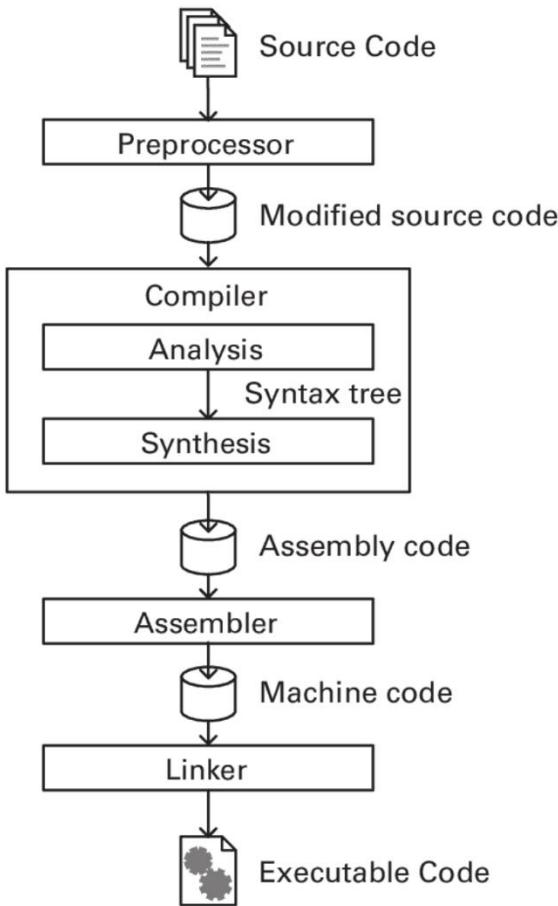


(b) Hardware case  
(hardware design language).

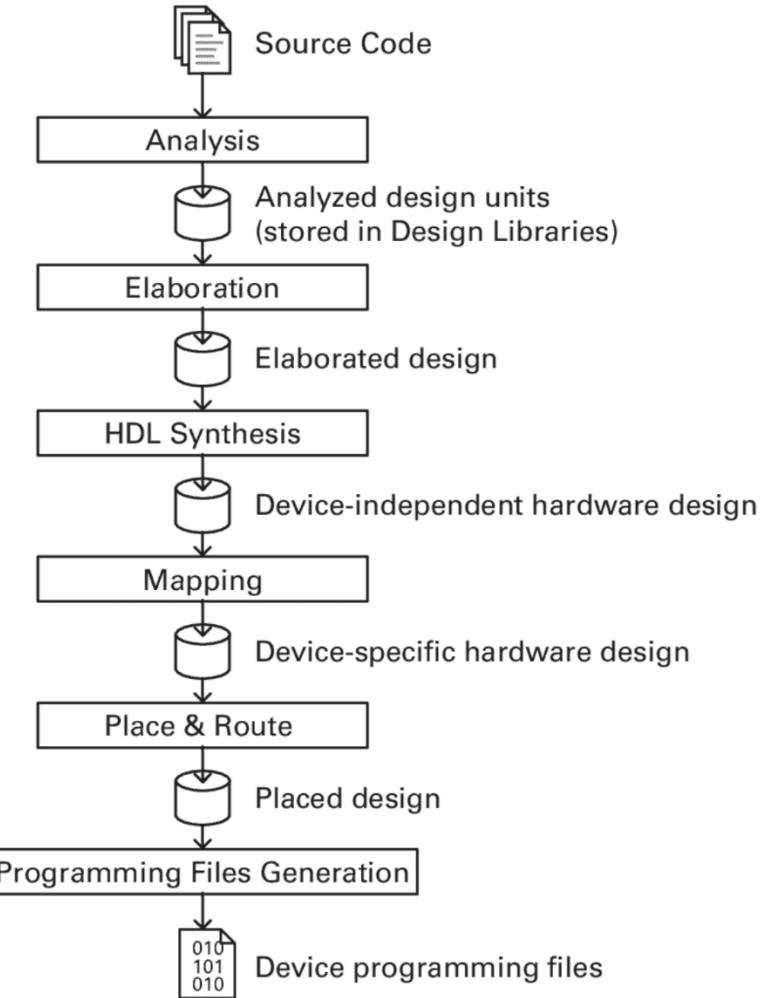
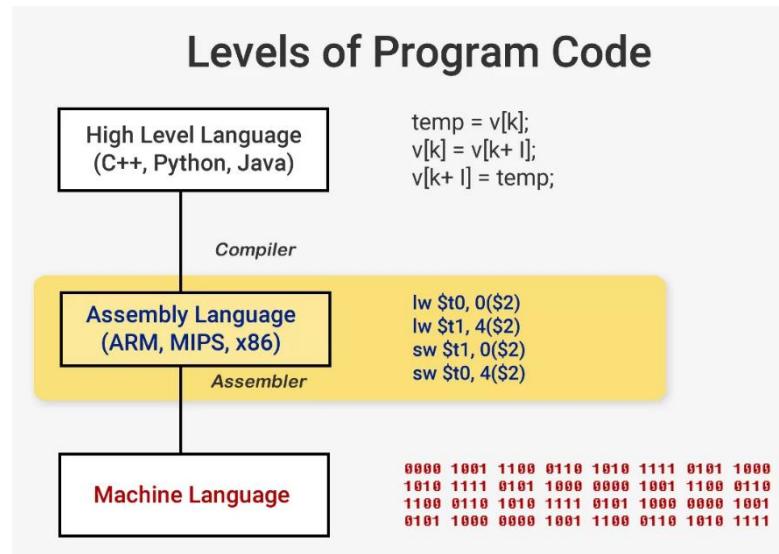


UNIVERSITY  
OF TURKU

# Digital Design Flow

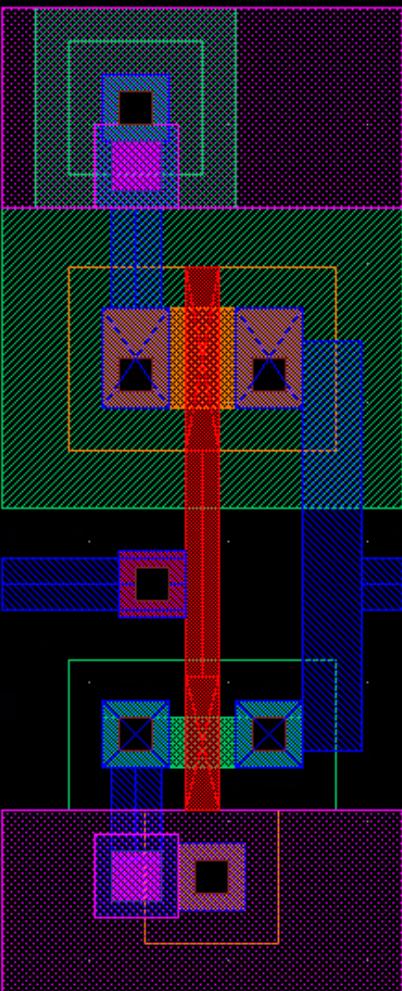


(a) Software case  
(programming language).



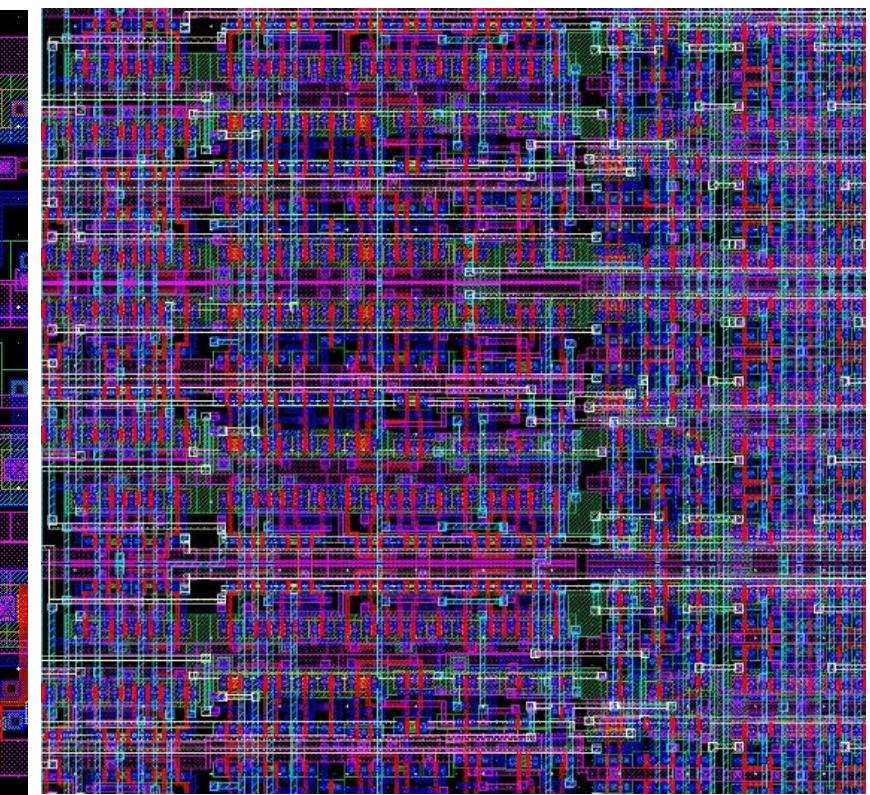
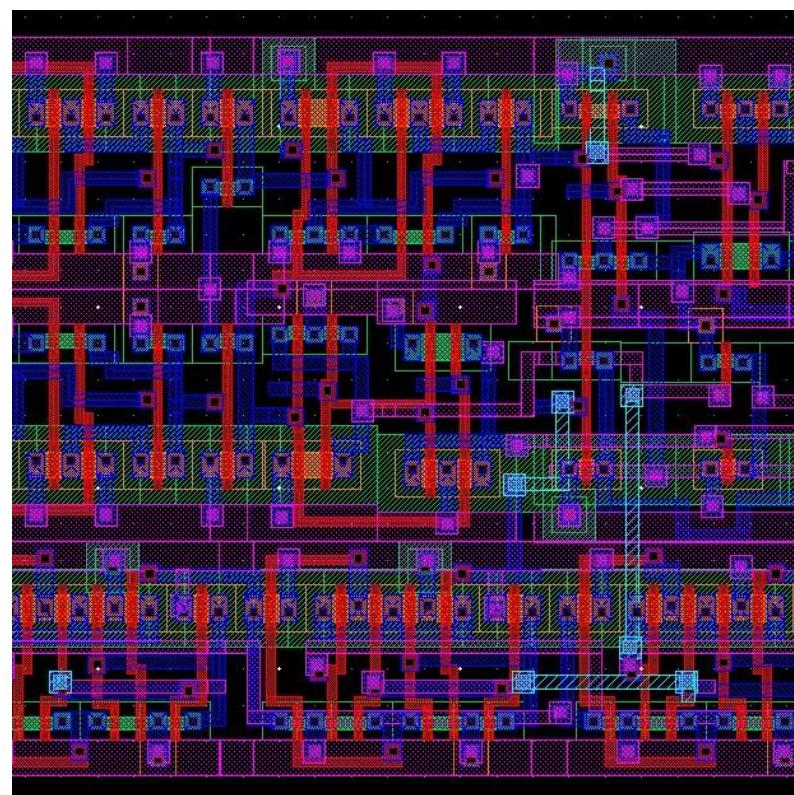
(b) Hardware case  
(hardware design language).

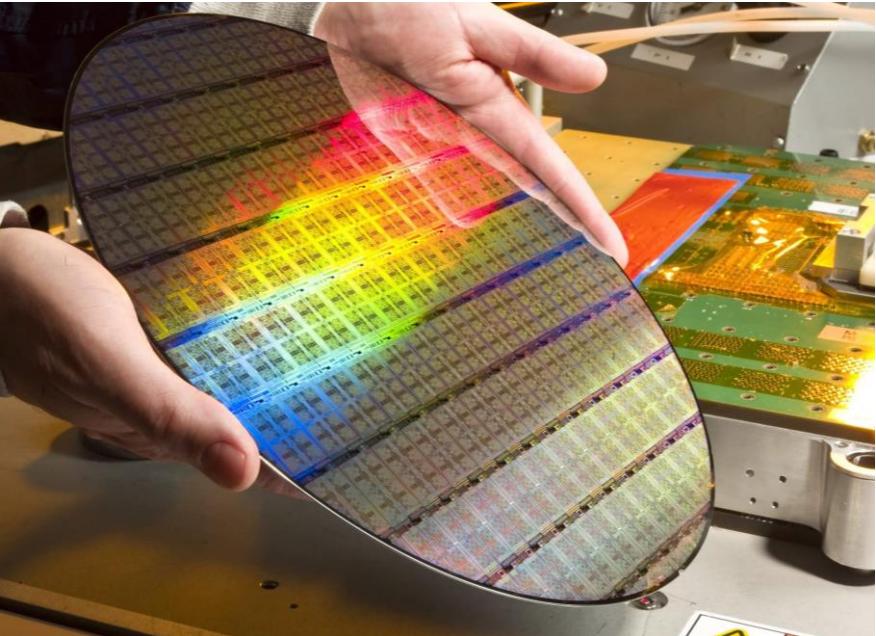
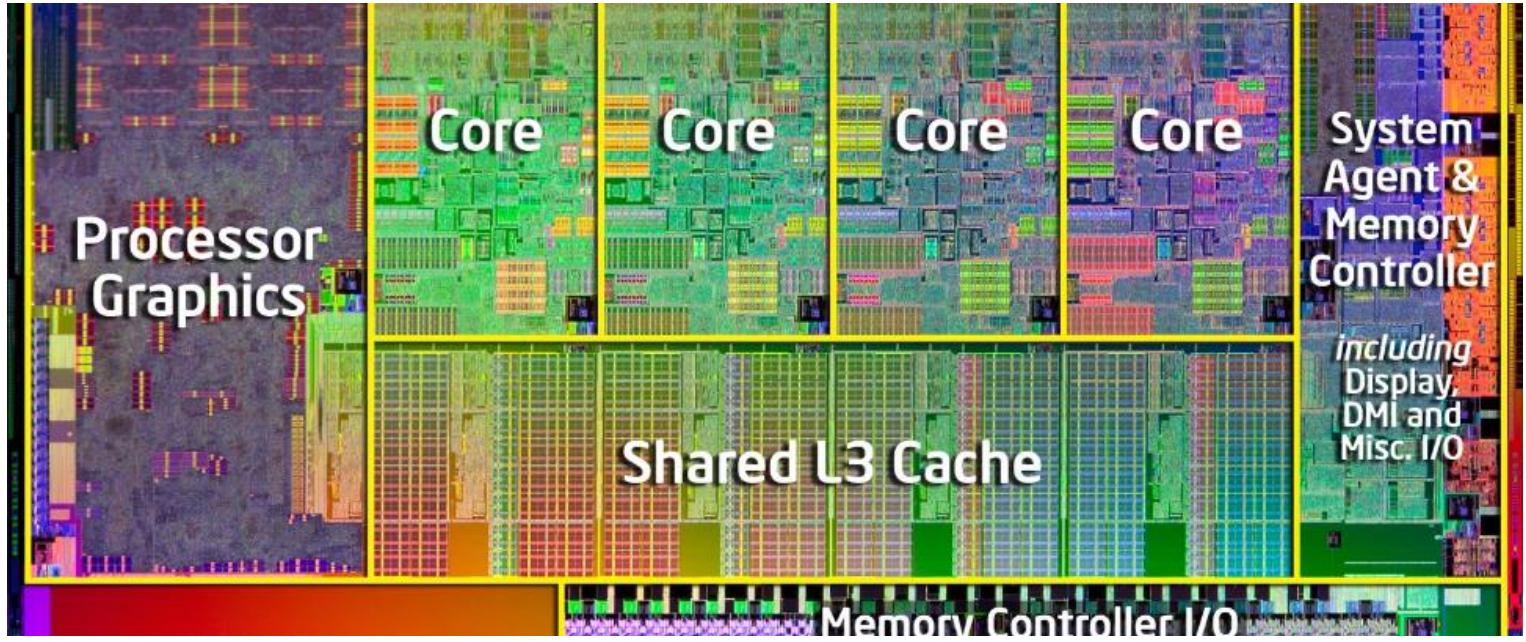
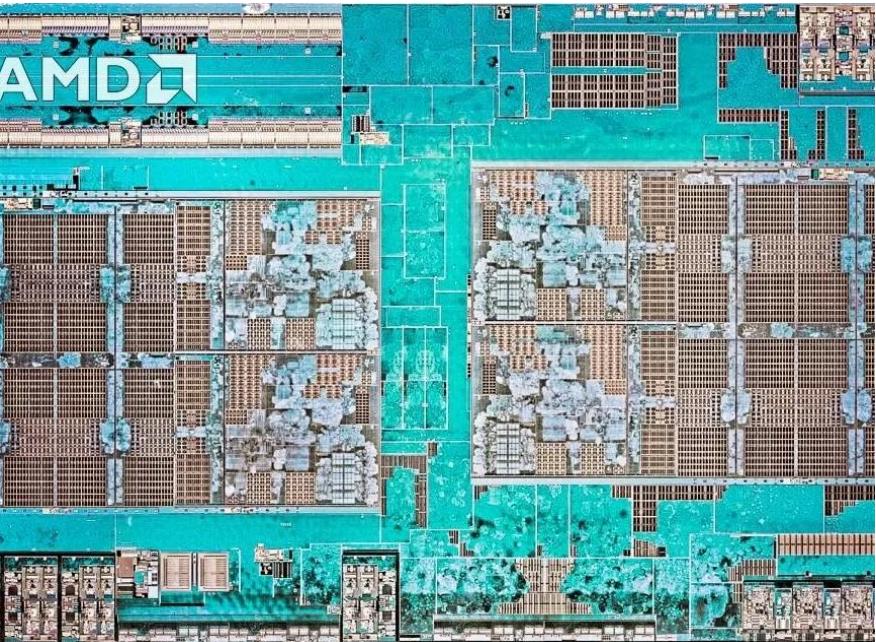
vdd!



and!

## Technology Mapping / Place and Route

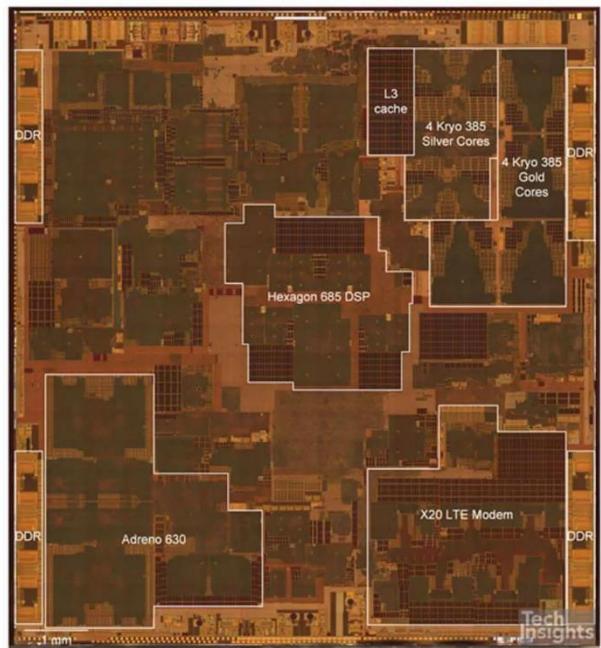




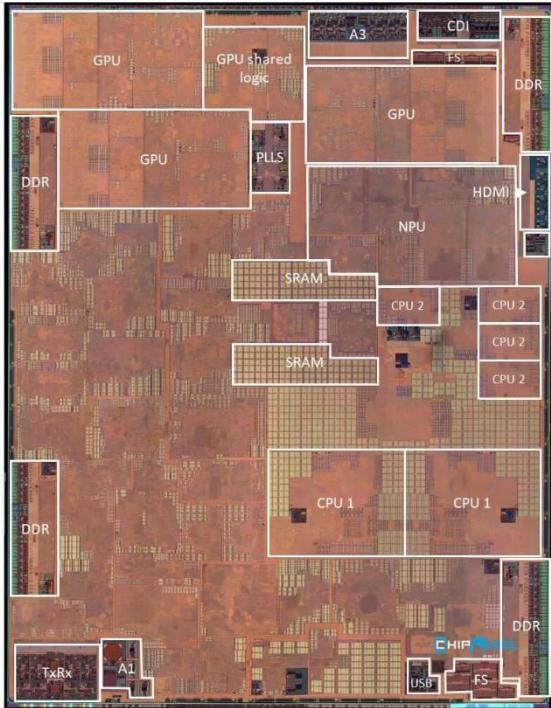
# Fabrication

---

# Fabrication



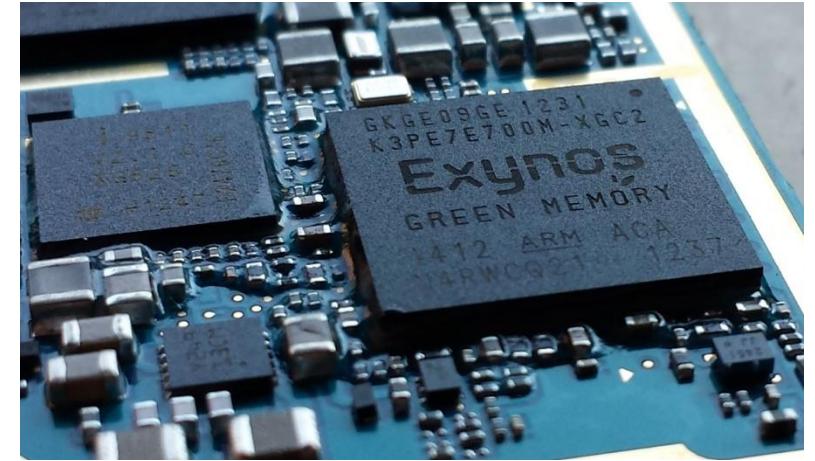
Snapdragon 845  
(~90 mm<sup>2</sup>)



A11 Bionic  
(87.66 mm<sup>2</sup>)



Exynos 9810  
(118.94 mm<sup>2</sup>)



---

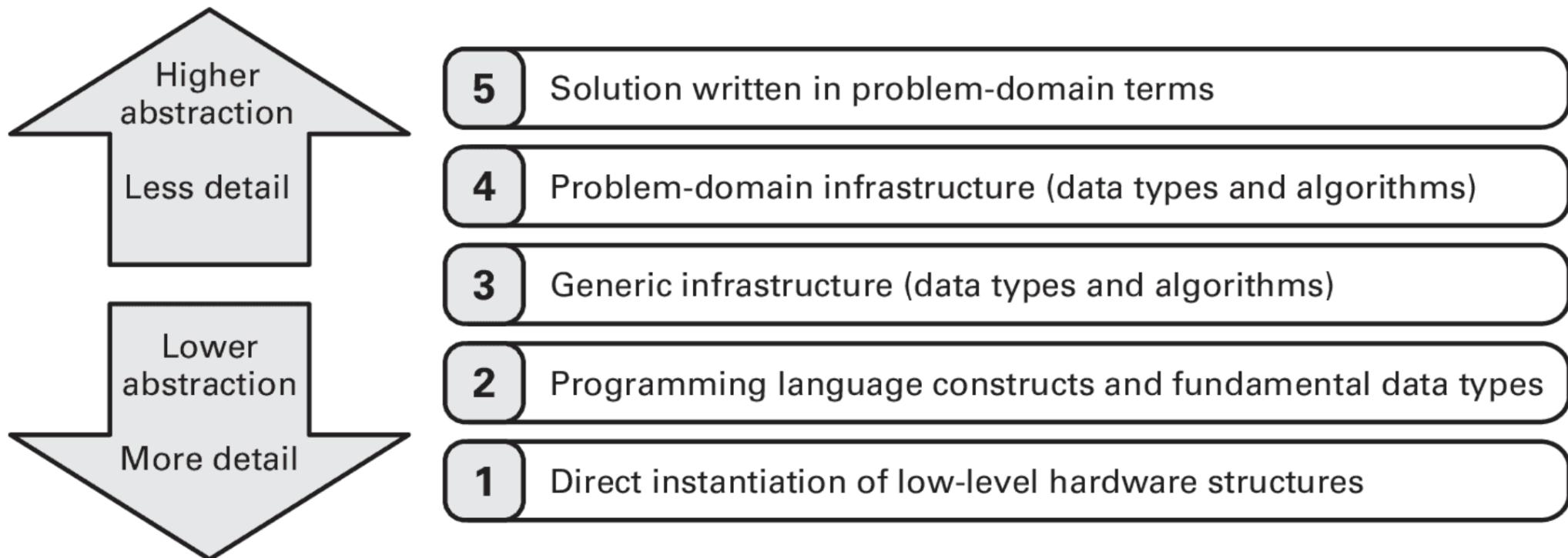
# Three Main Design Principles

---

- Abstraction
- Modularity
- Hierarchy



# Layers of Abstraction



# Levels of Abstraction

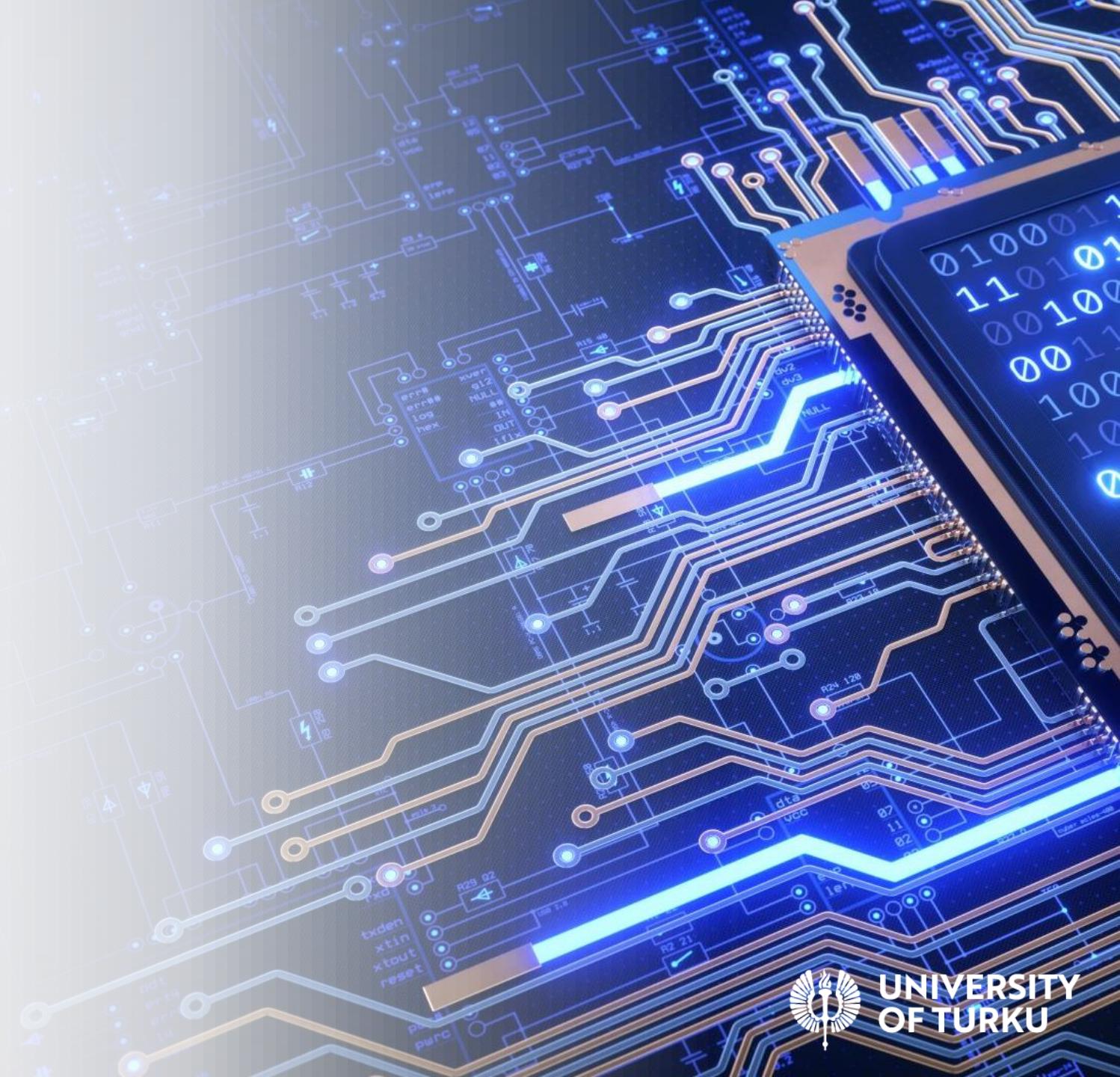
Level	Components	Representation
Behavioral	Algorithms, FSMs	Algorithmic representation of system functions
System (architectural)	Processors, memories, functional subsystems	Interconnection of major functional units
Register Transfer (dataflow)	Registers, adders, comparators, ALUs, counters, controllers, bus	Data movement and transformation and required sequential control
Gate (logic level)	Gates, flip-flops	Implementation of register level components using gates and flip-flops
Transistor	Transistors, resistors, capacitors	Implementation of gates and flip-flops using transistors, capacitors, and resistors





# Modelling

---



UNIVERSITY  
OF TURKU

# Modelling

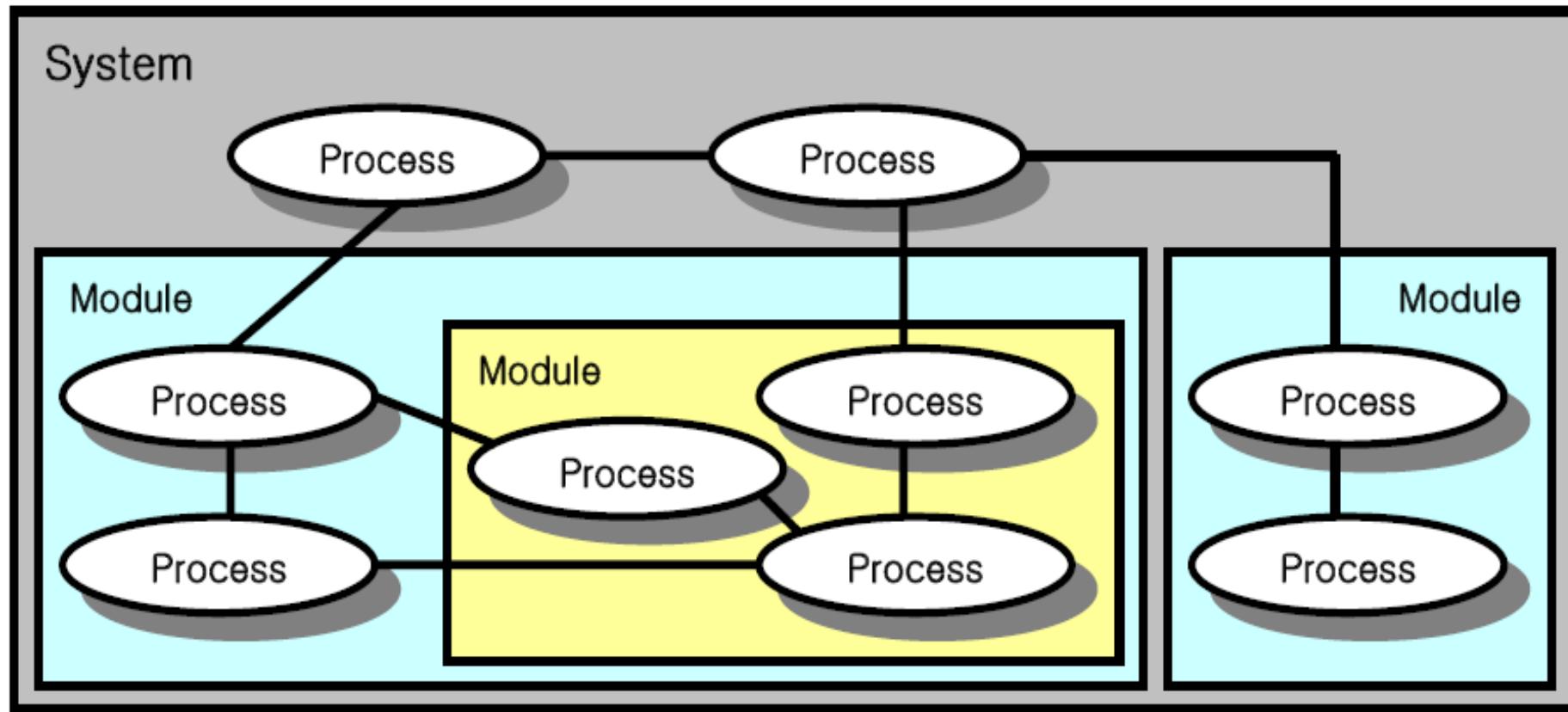
- A model helps us to understand the important aspects of a complex system *before* its implementation
- A model attempts to simulate an *abstract* model of a particular system
- The abstraction level of a model defines how much low-level details are included in the model

# Can't see the forest for the tree

In Finnish: Nähdä metsä puilta



# What is the forest?





UNIVERSITY  
OF TURKU