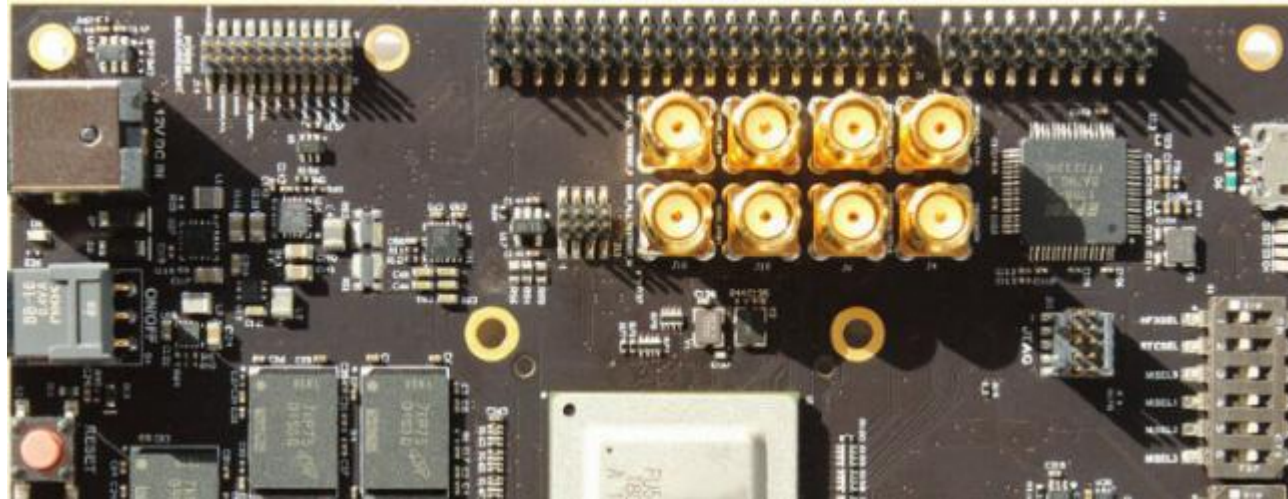# Computer Architecture and Operating Systems
Lecture 1: Introduction

Andrei Tatarnikov

atatarnikov@hse.ru
@andrewt0301

Course Resources

- **Wiki**

http://wiki.cs.hse.ru/ACOS_DSBA_2020/2021

- **Web site**

https://andrewt0301.github.io/hse-acos-course/

- **Telegram channel**

https://t.me/joinchat/AAAAAFDXhCd-WvYYZwBPGQ

Course Team

Instructors

Andrei Tatarnikov

Assistants

TODO

Course Outline

Syllabus (see details in the web site)

- Module 3: Computer Architecture
  - Computer architecture
  - Assembly language programming (RISC-V)
  - Home works, quizzes, and test.
- Module 4: Operating Systems
  - Operating System Architecture (Linux)
  - System programming in C
  - Home works, quizzes, and test
- Final Exam

Course Motivation

Brief History

Eight Great Ideas

- Design for **Moore's Law**

- Use **abstraction** to simplify design

- Make the **common case fast**

- Performance *via* **parallelism**

- Performance *via* **pipelining**

- Performance *via* **prediction**

- **Hierarchy** of memories

- **Dependability** *via* redundancy

MOORE'S LAW

ABSTRACTION

COMMON CASE FAST

PARALLELISM

PIPELINING

PREDICTION

HIERARCHY

DEPENDABILITY

Abstraction

- Hiding details when they are not important

Focus of this course
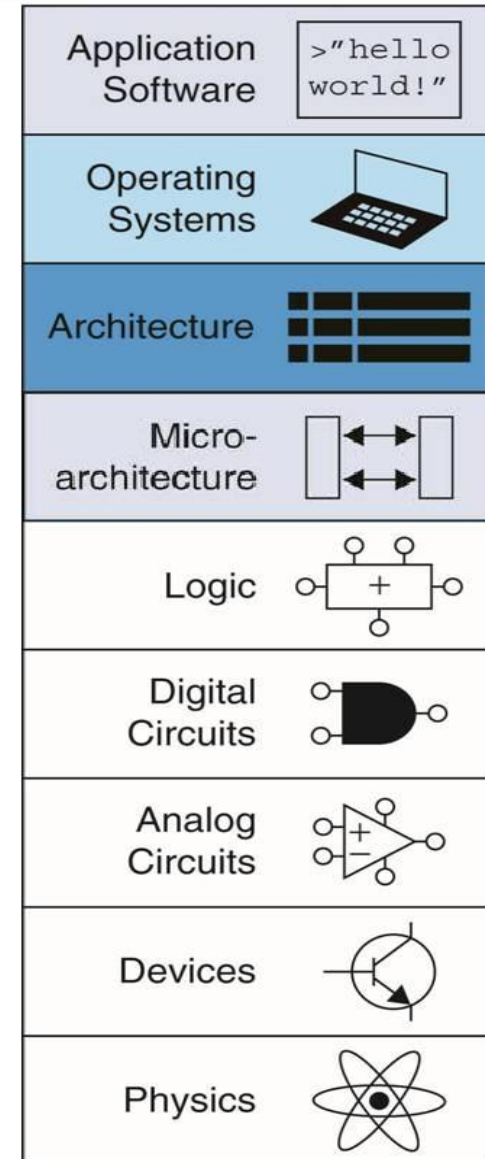
Any Questions?

```
                    .text
        __start:    addi t1, zero, 0x18
                    addi t2, zero, 0x21
     cycle:         beq t1, t2, done
                    slt t0, t1, t2
                    bne t0, zero, if_less
                    nop
                    sub t1, t1, t2
                    j cycle
                    nop
   if_less:         sub t2, t2, t1
                    j cycle
  done:             add t3, t1, zero
```