Digital Design and Computer Architecture

227-0003-10L (7 ECTS), BSc in CS, ETH Zurich

- **Lecturers:**
  Mohammad Sadrosadati, D-ITET, ETH Zurich
  Frank K. Gürkaynak, D-ITET, ETH Zurich

- **Labs:**
  The SAFARI group
  digitaltechnik@lists.inf.ethz.ch

- **Website:**
  https://safari.ethz.ch/ddca
  https://safari.ethz.ch/digitaltechnik

In This Lecture

- **Motivation for the lecture series**
  - Why should you care about
    Digital Design and Computer Architectures
- **The Art of Managing Complexity**
  - How can extremely complex systems be designed?
- **Organization of the Course**
  - Information on the lectures
  - Laboratory Exercises
  - Exam
  - How to get help when you are stuck

Introduction

Digital Design and Computer Architecture
Mohammad Sadrosadati
Frank K. Gürkaynak

http://safari.ethz.ch/ddca

First things first: How to follow the lecture?

- **Lectures in HG F7**
  - An overfill room in HG F5

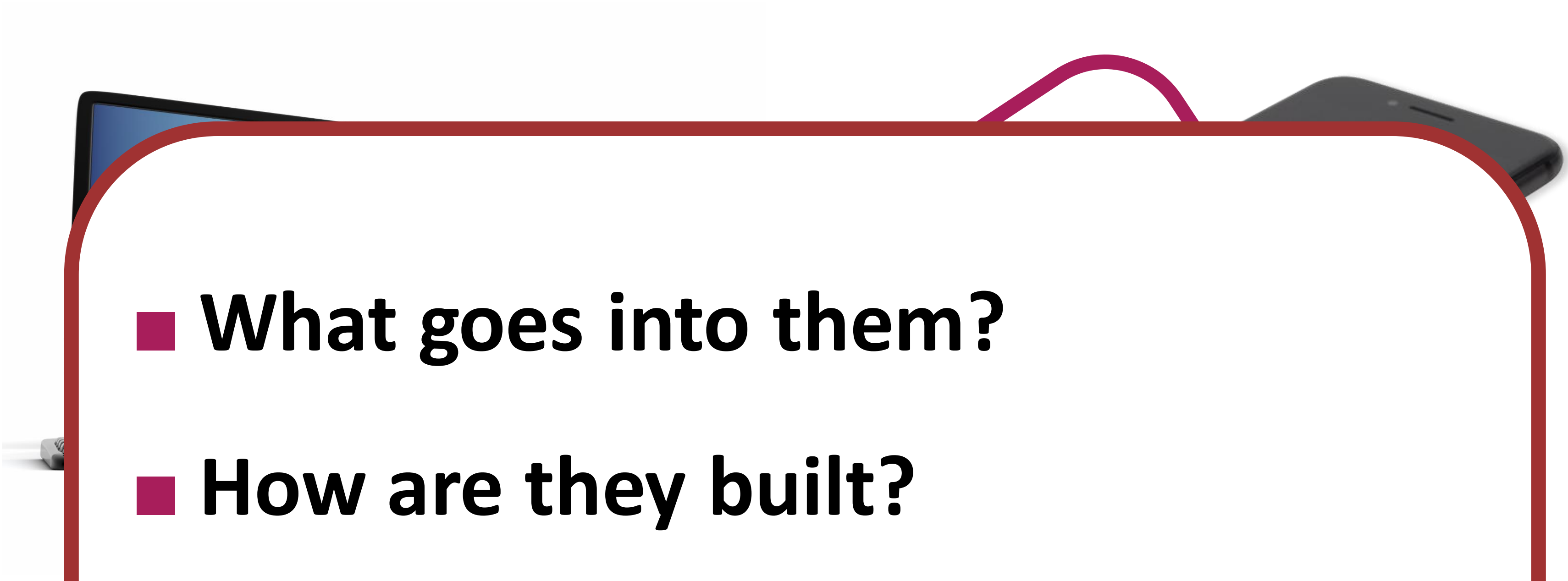
- **Livestream of the lecture on Youtube**
  - https://youtube.com/live/7DWHFBb4sJk

- **Recorded lectures available on the class WWW site**
  - http://safari.ethz.ch/digitaltechnik/

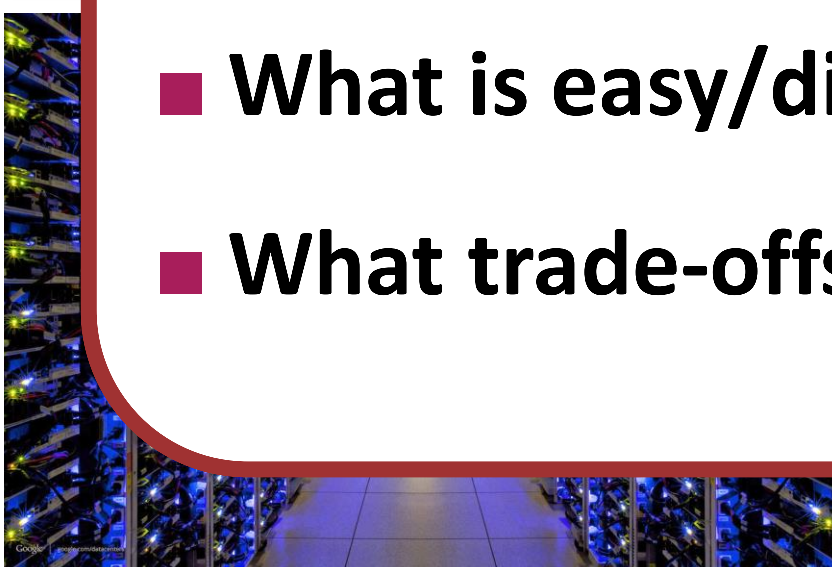
- **Also on the class site**
  - Presentation used in the class
  - All information from previous editions (slides, videos, exams)

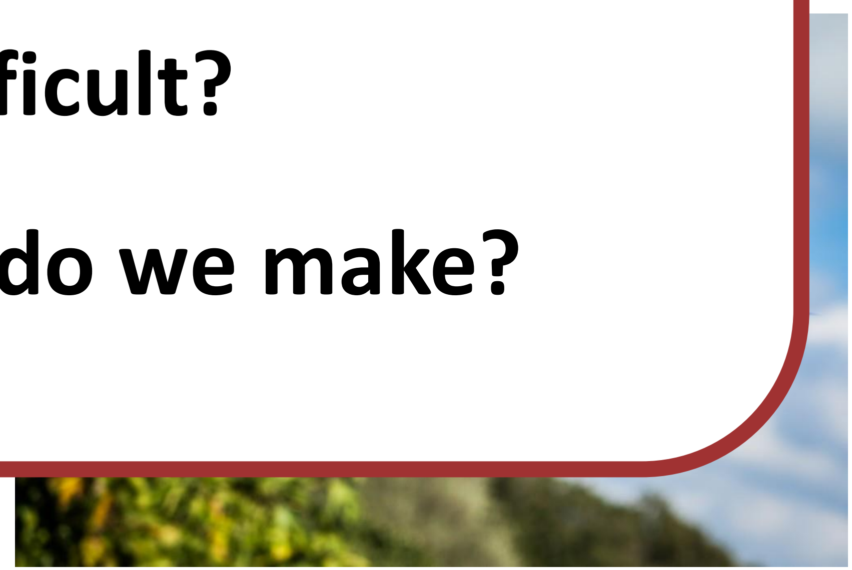
- **Moodle page for discussions**

Computers are everywhere



- What goes into them?
- How are they built?
- What is easy/difficult?
- What trade-offs do we make?





The Purpose of This Course Is That You:

- **Learn what's under the hood of a computer**
- **Learn the principles of digital design**
- **Learn to systematically debug increasingly complex systems**
- **Design and build a microprocessor**

Goal

- **Be able to understand a statement such as:**

  "The microarchitecture is a three-way superscalar, pipelined architecture. Three-way superscalar means that by using parallel processing techniques, the processor is able on average to decode, dispatch, and complete execution of (retire) three instructions per clock cycle. To handle this level of instruction throughput, the P6 processor family uses a decoupled, 12-stage superpipeline that supports out-of-order instruction execution."

Taken from:
http://www.intel.com/content/dam/www/public/us/en/documents/manuals/64-ia-32-architectures-software-developer-manual-325462.pdf

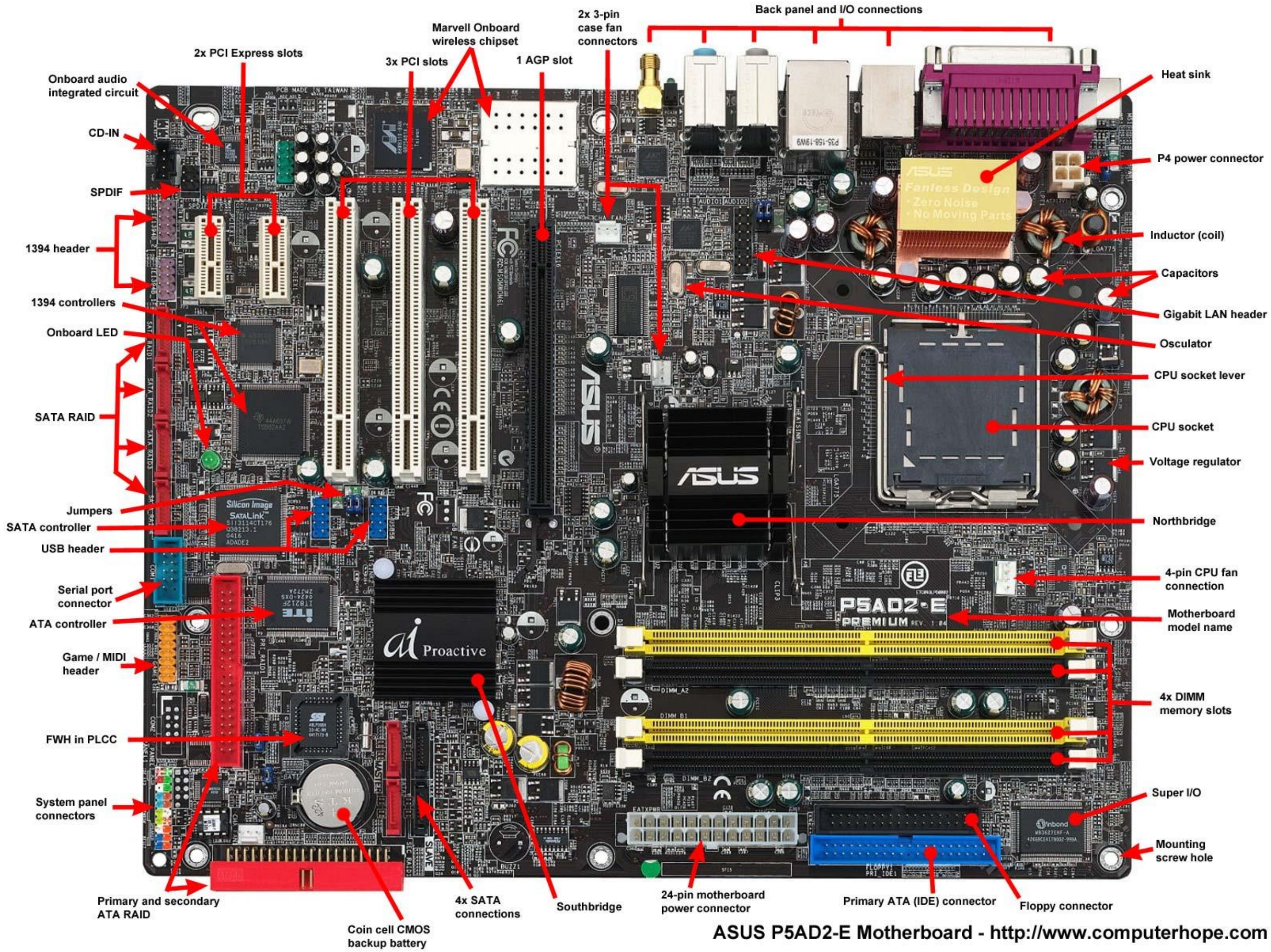
What Is This Lecture About?

Computers

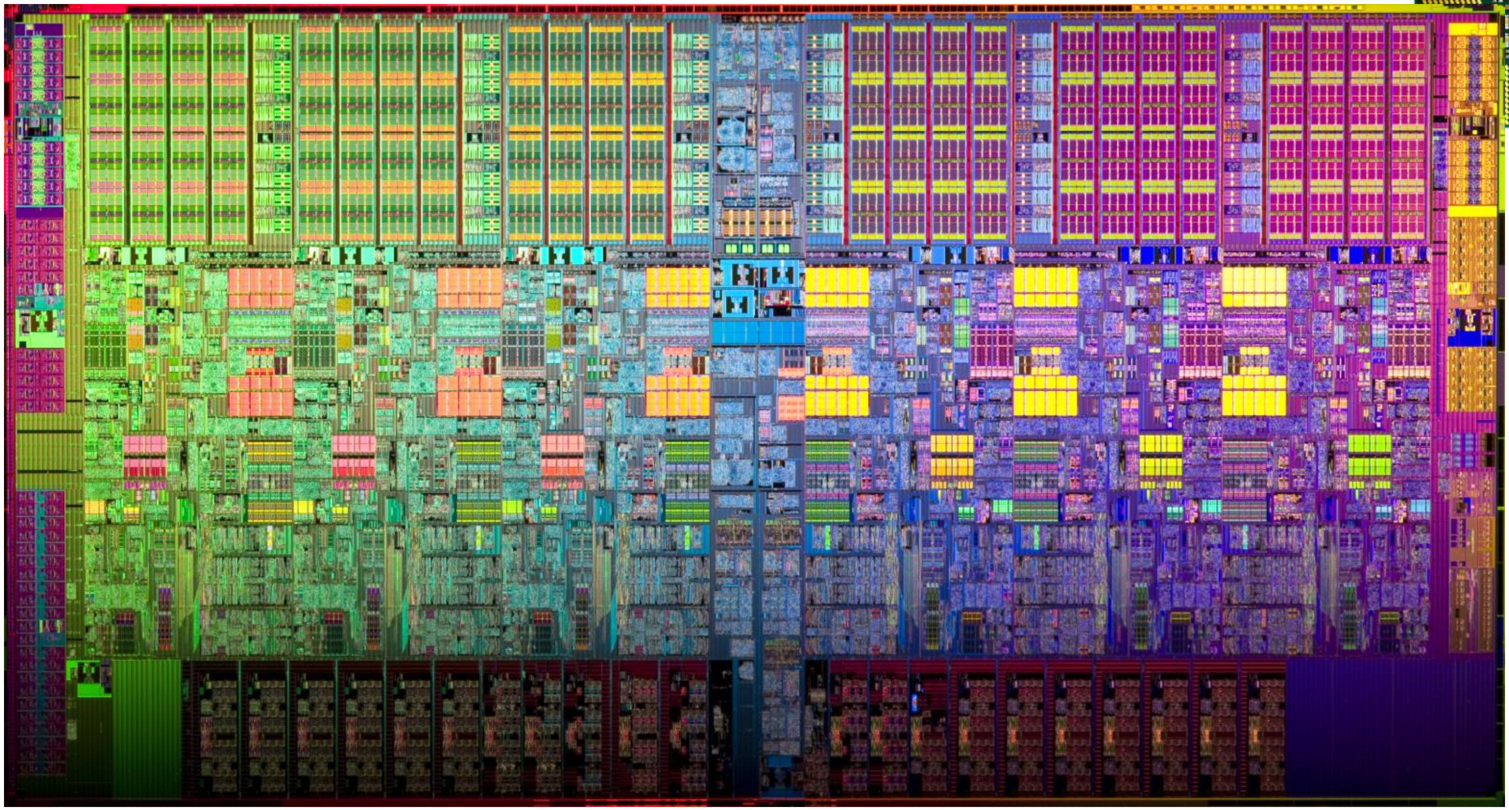
What Is This Lecture About?

What's inside?

What Is This Lecture About?



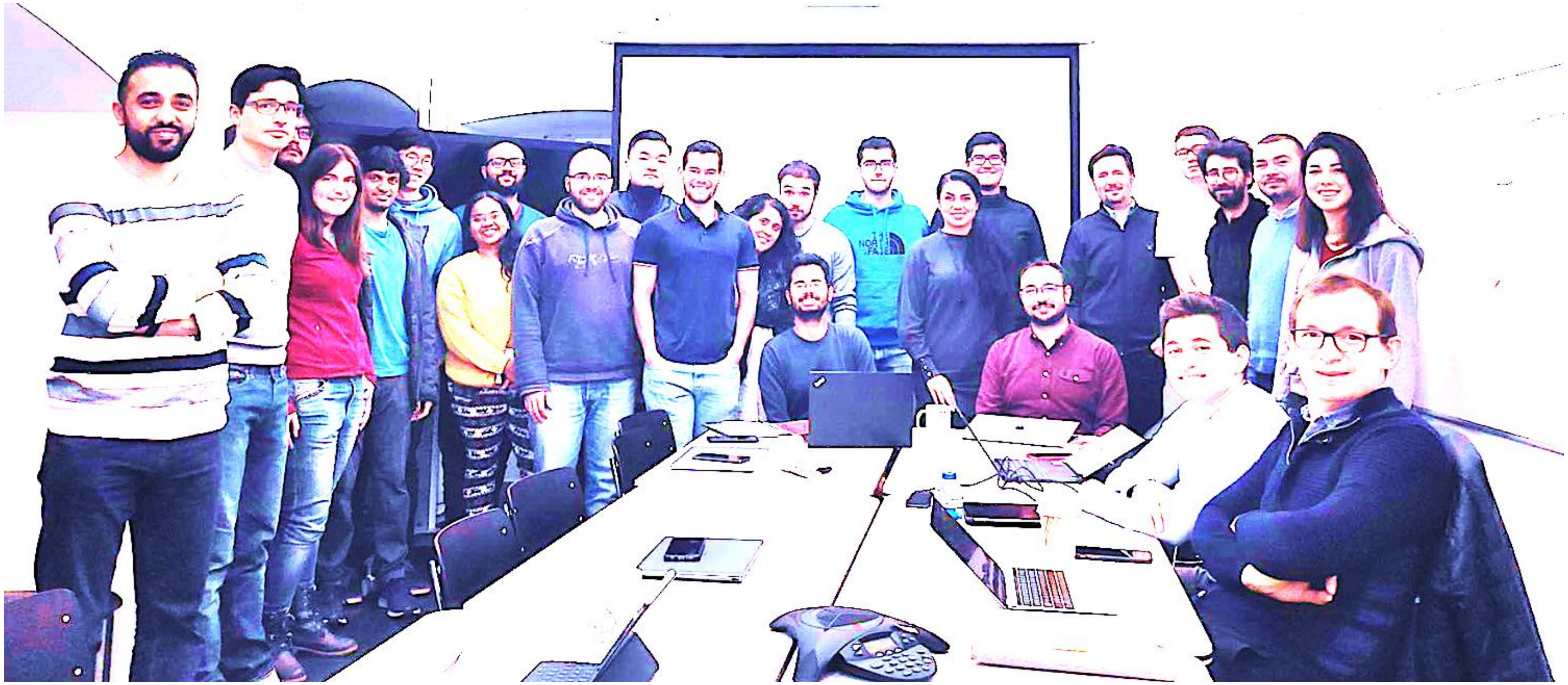
What Is This Lecture About?



How can we build them ?

Intel Westmere: http://www.intel.com/pressroom/archive/releases/2010/20100316comp_sm.htm

This lecture is brought to you by



Onur Mutlu's SAFARI Research Group

Computer architecture, HW/SW, systems, bioinformatics, security, memory

SAFARI https://safari.ethz.ch

Prof. Onur Mutlu

Prof. Onur Mutlu

- **Career**
  - Full Professor @ ETH Zurich ITET (INFK), since Sept 2015
  - Strecker Professor @ Carnegie Mellon University ECE (CS), 2009-2016, 2016-…
  - Started the Comp Arch Research Group @ Microsoft Research, 2006-2009
  - Worked @ Google, VMware, Microsoft Research, Intel, AMD
  - PhD in Computer Engineering from University of Texas at Austin in 2006
  - BS in Computer Engineering & Psychology from University of Michigan in 2000
  - https://people.inf.ethz.ch/omutlu/ omutlu@gmail.com

Currently at a sabbatical leave in the US

- **Research and Teaching in:**
  - **Computer architecture, systems, hardware security, bioinformatics**
  - Memory and storage systems
  - Robust & dependable hardware systems: security, safety, predictability, reliability
  - Hardware/software cooperation
  - New computing paradigms; architectures with emerging technologies/devices
  - Architectures for bioinformatics, genomics, health, medicine, AI/ML
  - …

NOT Prof. Onur Mutlu

- **Frank Kağan Gürkaynak**
- **Senior scientist at D-ITET**
  - Group of Luca Benini
  - B.Sc. M.Sc. In Istanbul Technical University
  - Ph.D. at ETH Zürich
  - At ETH Zurich/EPFL since 2000
  - Runs the Microelectronics Design Center
- **Involved in this lecture for 10+ years**
  - Also teaches at D-ITET
    - **VLSI1**: HDL Design (elective in CS)
    - **VLSI2**: Integrated Circuit Design (elective in CS)
    - VLSI4: Testing of Integrated Circuits

NOT Prof. Onur Mutlu

- Mohammad Sadrosadati
- Senior Researcher & Lecturer
  - Prof. Mutlu's SAFARI Research Group
  - B.Sc., M.Sc., and Ph.D. at Sharif University of Technology
  - At ETH Zurich in 2017-2018 & since 2021
  - mohammad.sadrosadati@safari.ethz.ch
- Involved in this lecture for 2 years as the Head-TA
- Also teaches
  - Computer Architecture
  - Seminar in Computer Architecture
  - P&S Understanding and Designing Modern Storage Systems

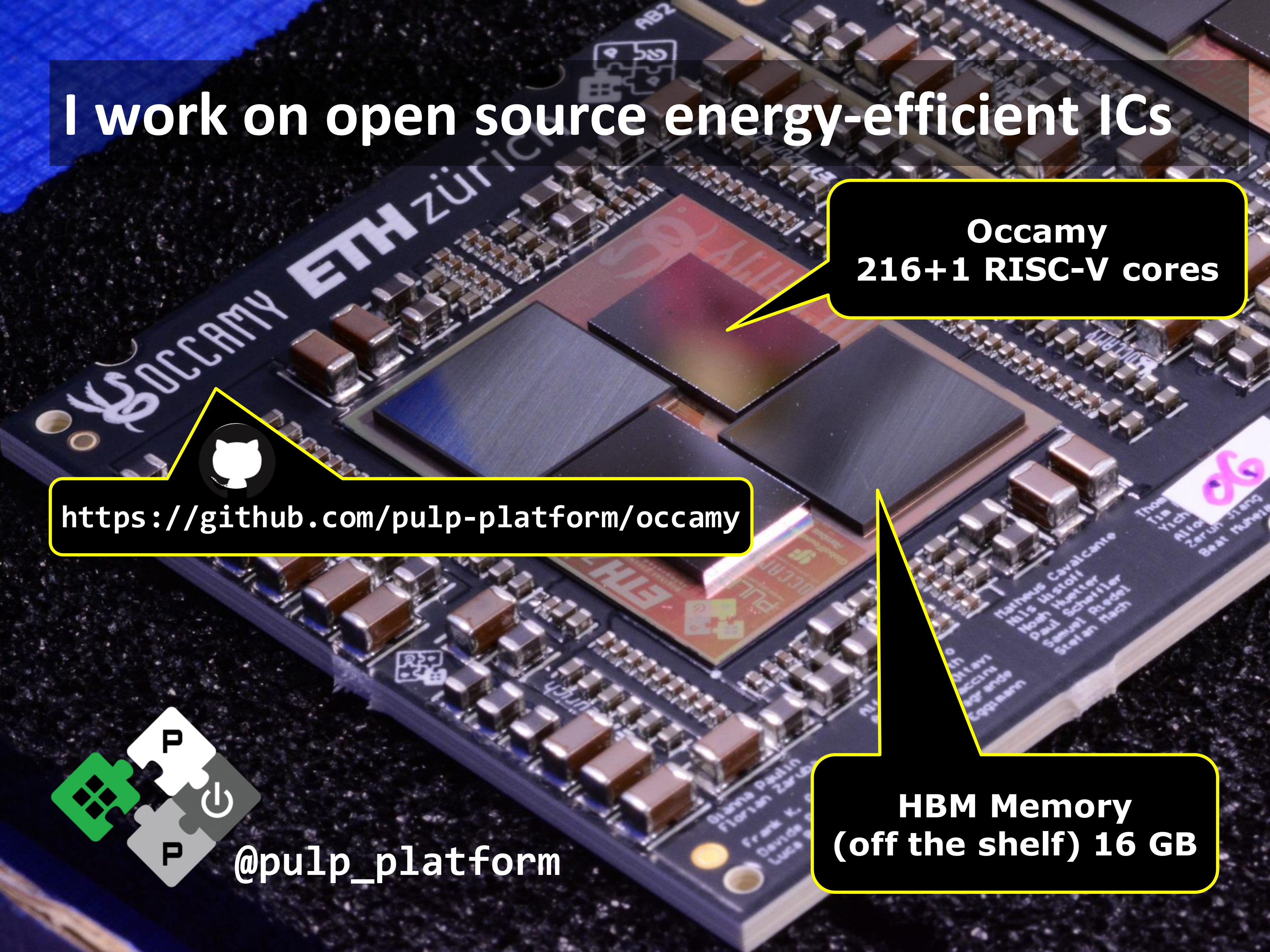
Changes this year

- **The exercises are identical**
- **The teaching goals of the lecture remains the same**
- **The recommended textbook is the same**

BUT

- **Frank can not keep the same speed as Prof. Mutlu**
  - Slightly different (and a bit slower) style
  - Leaves less time for some of the more advanced topics
  - Exam (August 2024, Jan 2025) will consider this
- **Prof. Mutlu will be back for FS2025**

I work on open source energy-efficient ICs



Occamy
216+1 RISC-V cores

https://github.com/pulp-platform/occamy

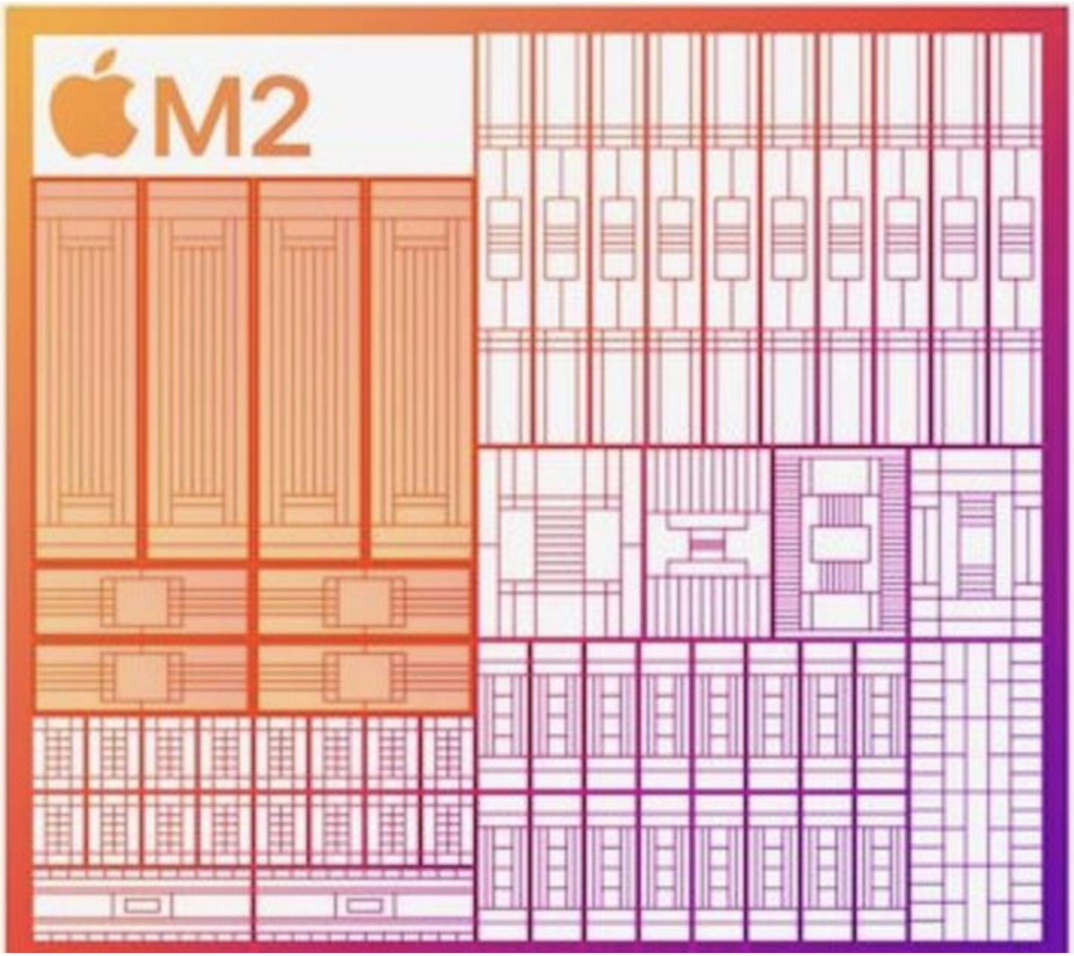
HBM Memory
(off the shelf) 16 GB

@pulp_platform

The Art of Managing Complexity

- **Abstraction**
- **Discipline**
- **The Three -Y’s**
  - Hierarchy
  - Modularity
  - Regularity

An example: Apple M2



~ 13.5mm

SoC	M2
CPU	4x High Performance (Avalanche?) 16MB Shared L2 4x High Efficiency (Blizzard?) 4MB Shared L2
GPU	"Next Generation" 10-Core 3.6 TFLOPS
Neural Engine	16-Core 15.8 TOPS
Memory Controller	LPDDR5-6400 8x 16-bit CH 100GB/sec Total Bandwidth (Unified)
Memory Capacity	24GB
Encode/ Decode	8K H.264, H.265, ProRes, ProRes RAW
USB	USB4/Thunderbolt 3 2x Ports
Transistors	20 Billion
Mfc. Process	"Second Generation 5nm" TSMC N5P?

https://www.anandtech.com/show/17431/apple-announces-m2-soc-apple-silicon-updated-for-2022

Abstraction

- **Hiding details when they are not important**

Abstraction Levels	Examples
Application Software	Programs
Operating Systems	Device drivers
Architecture	Instructions, Registers
Micro architecture	Datapath, Controllers
Logic	Adders, Memories
Digital Circuits	AND gates, NOT gates
Analog Circuits	Amplifiers
Devices	Transistors, Diodes
Physics	Electrons

Abstraction

- **Hiding details when they are not important**

	Abstraction Levels	Examples
	Application Software	Programs
	Operating Systems	Device drivers
This Course	**Architecture**	Instructions, Registers
	Micro architecture	Datapath, Controllers
	Logic	Adders, Memories
	Digital Circuits	AND gates, NOT gates
	Analog Circuits	Amplifiers
	Devices	Transistors, Diodes
	Physics	Electrons

Discipline

- Intentionally restricting your design choices to that you can work more productively at higher abstraction levels

The Three -Y’ s

- **Hierarchy**
  - A system is divided into modules of smaller complexity
- **Modularity**
  - Having well defined functions and interfaces
- **Regularity**
  - Encouraging uniformity, so modules can be easily re-used

The Digital Abstraction

- **Most physical variables are continuous, for example:**
  - Voltage on a wire
  - Frequency of an oscillation
  - Position of a mass

- **Instead of considering all values, the digital abstraction considers only a discrete subset of values**

Digital Discipline: Binary Values

- **Typically consider only two discrete values:**
  - 1's and 0's
  - 1 = TRUE = HIGH
  - 0 = FALSE = LOW

- **1 and 0 can be represented by specific voltage levels, rotating gears, fluid levels, etc.**

- **Digital circuits usually depend on specific voltage levels to represent 1 and 0**

- ***Bit*: *B*inary dig*it***

How Much Can We Do with 1s and 0s?

- **How do you write**

2024

if you only had 1s and 0s to write with?

How Much Can We Do with 1s and 0s?

- How do you write

2024

if you only had 1s and 0s to write with?

- How about:

$$-4.7913 \cdot 10^{-12} \times 8.1653 \cdot 10^{-3}$$

- We will learn how to calculate with only 1s and 0s.

From Real Problems to 1s and 0s

- **Example Problem: You are going to the cafeteria for lunch**
  - You won't eat lunch ($\bar{\mathbf{E}}$)
  - If it is not open ($\bar{\mathbf{O}}$) or
  - If they only serve cabbage (**C**)

From Real Problems to 1s and 0s

- **Example Problem: You are going to the cafeteria for lunch**
  - You won't eat lunch ($\bar{\mathbf{E}}$)
  - If it is not open ($\bar{\mathbf{O}}$) or
  - If they only serve cabbage (**C**)

O	*C*	*E*
0	0	0
0	1	0
1	0	1
1	1	0

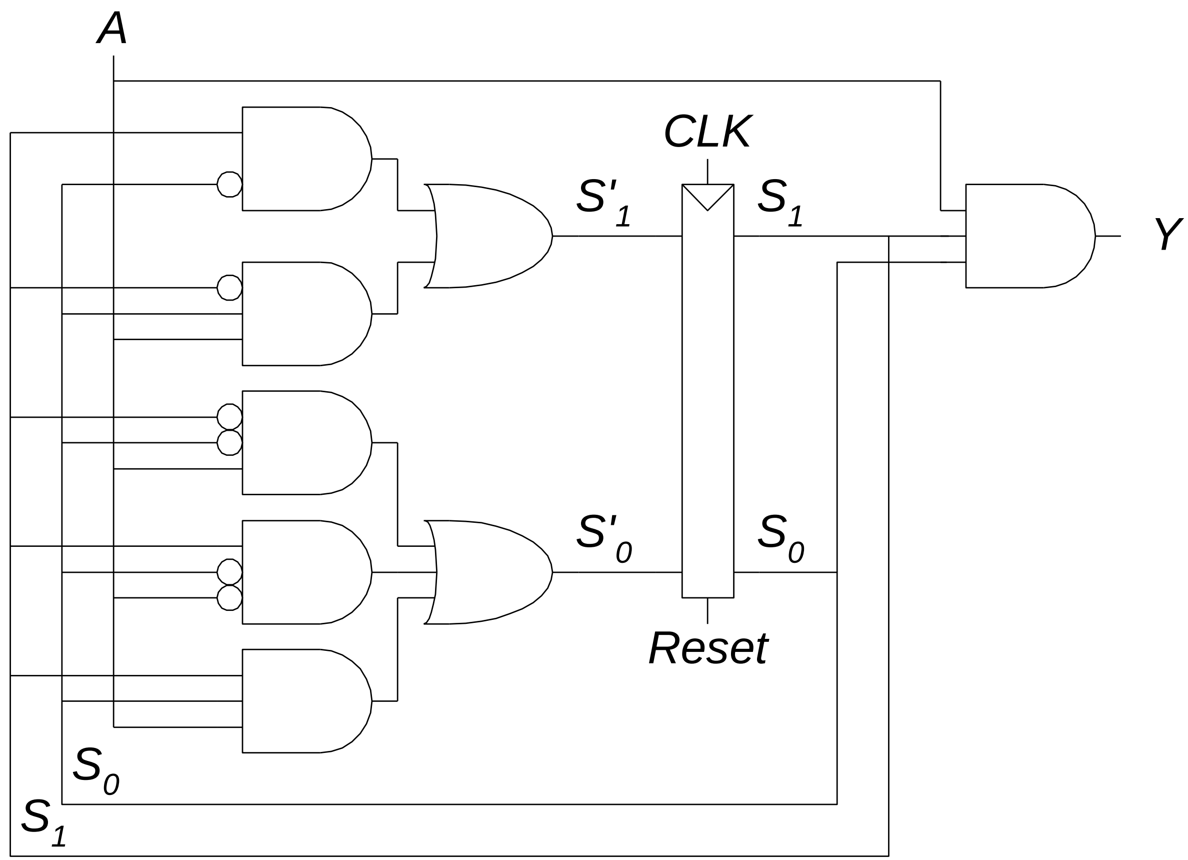
- **We will learn how to formulate problems in truth tables**

From Logic to Circuits

Example: Alyssa P. Hacker has a snail that crawls down a paper tape with 1's and 0's on it. The snail smiles whenever the last four digits it has crawled over are 1101.

From Logic to Circuits

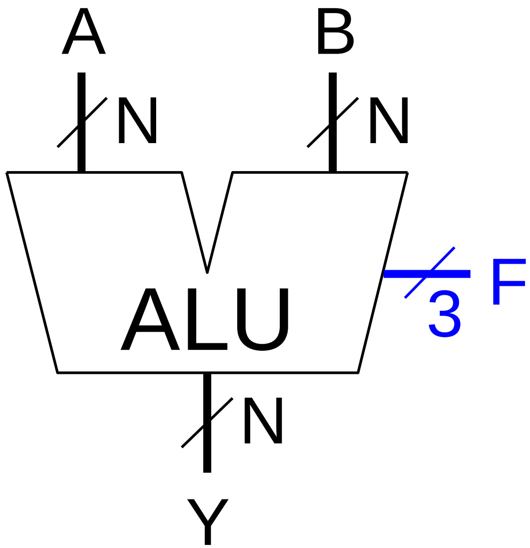
Example: Alyssa P. Hacker has a snail that crawls down a paper tape with 1's and 0's on it. The snail smiles whenever the last four digits it has crawled over are 1101.



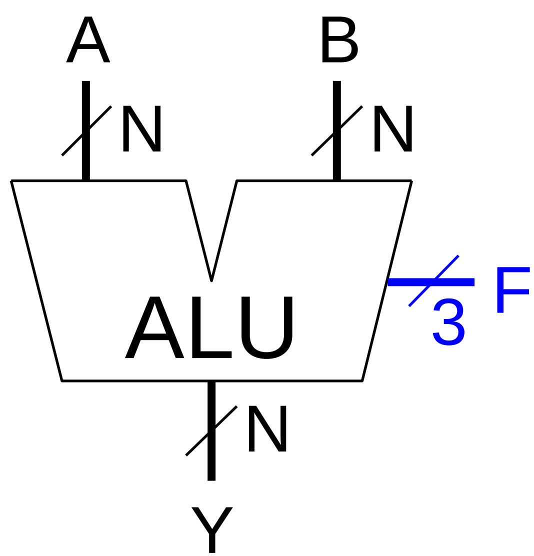
Verilog Will Help us Describe Circuits

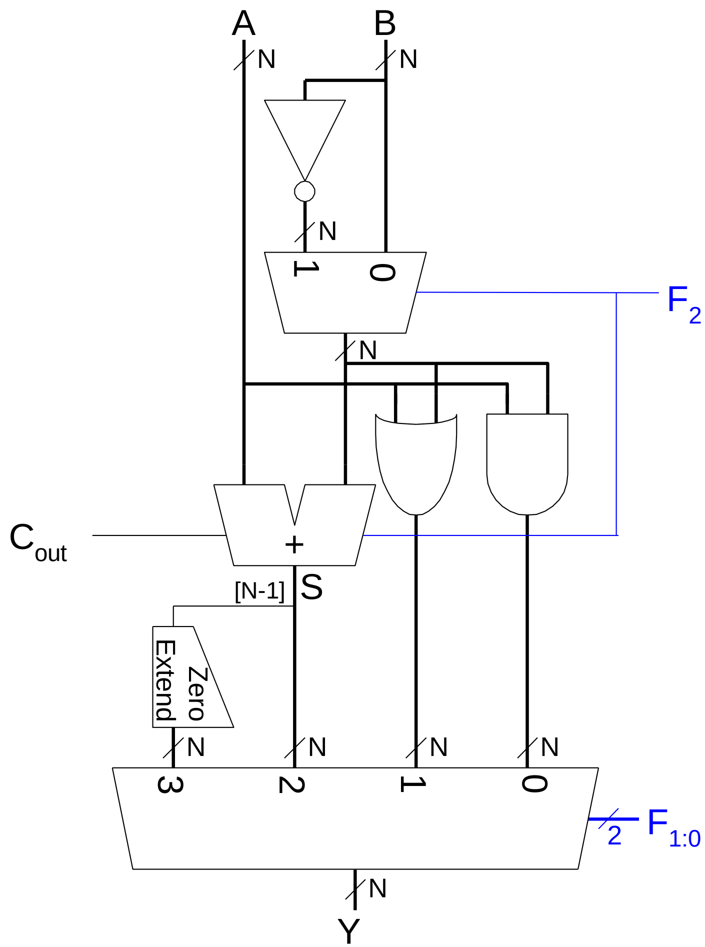
```
module divideby3FSM (input clk, input reset, output q);
   reg  [1:0] state, nextstate;

   parameter S0 = 2'b00;
   parameter S1 = 2'b01;
   parameter S2 = 2'b10;

   always @ (posedge clk, posedge reset) // state register
      if (reset) state <= S0;
      else       state <= nextstate;
   always @ (*)                          // next state logic
      case (state)
         S0:      nextstate = S1;
         S1:      nextstate = S2;
         S2:      nextstate = S0;
         default: nextstate = S0;
      endcase
   assign q = (state == S0);             // output logic
endmodule
```

How Can We Add/Multiply Binary Numbers?

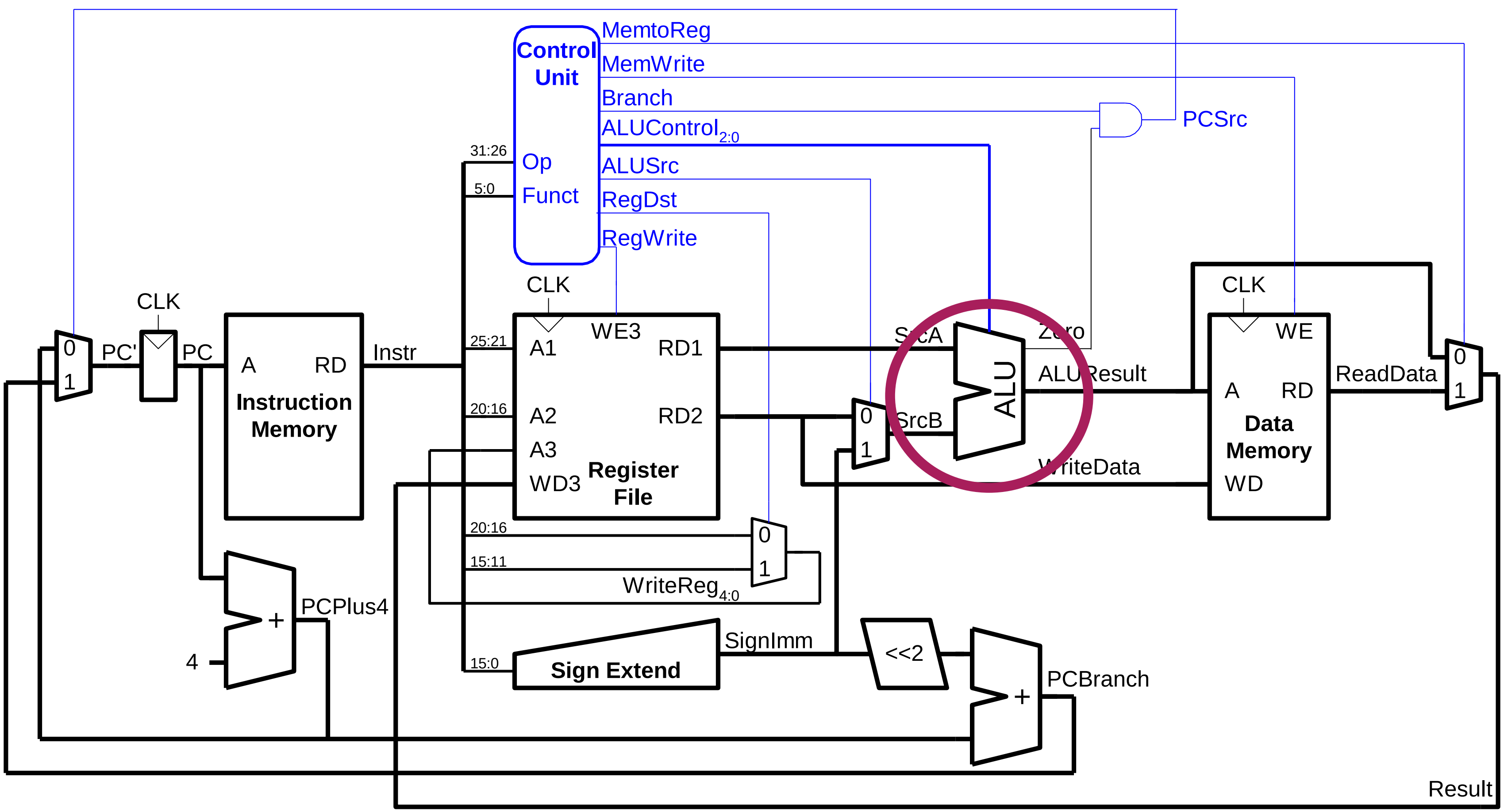


How Can We Add/Multiply Binary Numbers?





The Single-Cycle MIPS Architecture

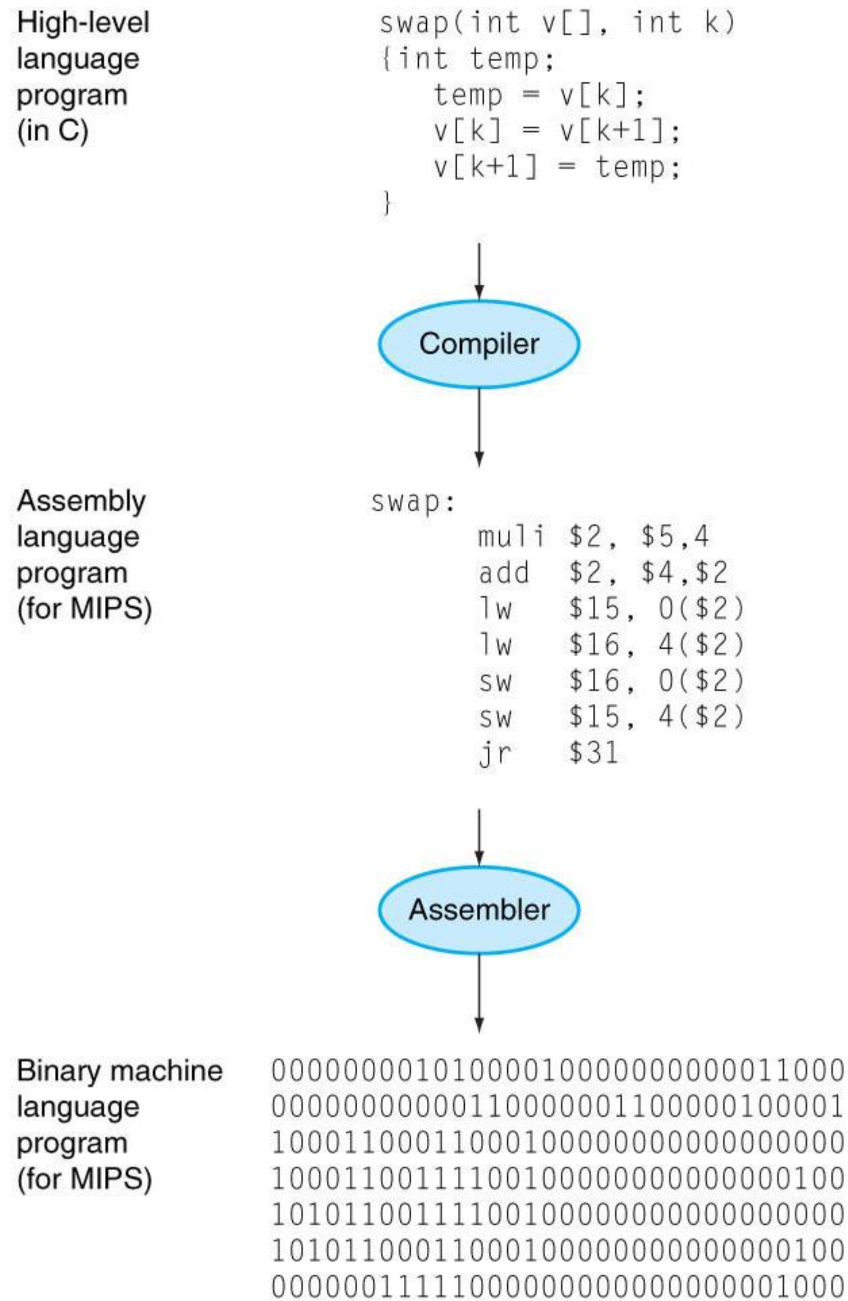


Program Running on MIPS

```
# MIPS assembly code
# $s0 = array base address, $s1 = i
# initialization code
  lui  $s0, 0x23B8        # $s0 = 0x23B80000
  ori  $s0, $s0, 0xF000   # $s0 = 0x23B8F000
  addi $s1, $0, 0         # i = 0
  addi $t2, $0, 1000      # $t2 = 1000

loop:
  slt  $t0, $s1, $t2      # i < 1000?
  beq  $t0, $0, done      # if not then done
  sll  $t0, $s1, 2        # $t0 = i * 4 (byte offset)
  add  $t0, $t0, $s0      # address of array[i]
  lw   $t1, 0($t0)        # $t1 = array[i]
  sll  $t1, $t1, 3        # $t1 = array[i] * 8
  sw   $t1, 0($t0)        # array[i] = array[i] * 8
  addi $s1, $s1, 1        # i = i + 1
  j    loop               # repeat
done:
```

C to Machine Code

- Start writing program in a high-level language
- Compiler translates this into simple instructions that the processor will understand
- Assembler translates this into 1s and 0s that will control the processor



Abstraction

- **Hiding details when they are not important**

	Abstraction Levels	Examples
	Application Software	Programs
	Operating Systems	Device drivers
This Course	**Architecture**	Instructions, Registers
	Micro architecture	Datapath, Controllers
	Logic	Adders, Memories
	Digital Circuits	AND gates, NOT gates
	Analog Circuits	Amplifiers
	Devices	Transistors, Diodes
	Physics	Electrons

In This Lecture

- **Motivation for the lecture series**
  - Why should you care about Digital Circuits?
- **The Art of Managing Complexity**
  - How can extremely complex systems be designed?
- **Organization of the Course**
  - Information on the lectures
  - Laboratory Exercises
  - Exam
  - How to get help when you are stuck

Schedule

- **Lectures:**
  Thursday, 14:15-16:00, HG F7 (overfill HG F5)
  Friday, 14:15-16:00, HG F7 (overfill HG F5)

  No lectures on: 29th March, 4th , 5th of April (Easter)
  9th of May (Ascension)

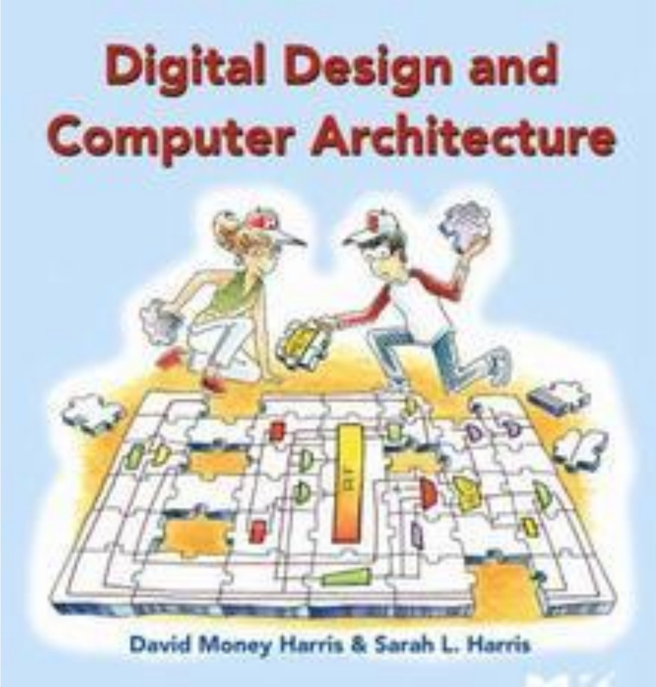
  Lab exercises:
  - Will start in the third week of the semester (5th of March)
  - 10 lab sessions
  - 2 hours each
  - Lab sessions on Tuesday, Wednesday and Friday
  - In computer rooms in HG (details in the next hour)

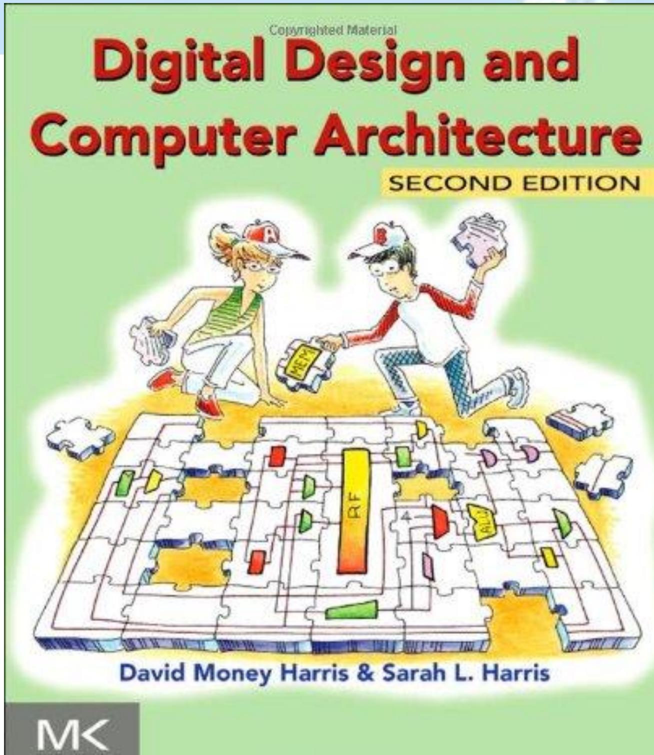
Course Book

- **Literature:**
  Digital Design and Computer Architecture,
  David Harris and Sarah Harris
  ISBN-10: 0123704979

- **The course closely follows this book:**
  - https://www.sciencedirect.com/book/9780123704979/digital-design-and-computer-architecture `(first edition)`
  - Multiple later editions available
  - https://www.sciencedirect.com/book/9780123944245/digital-design-and-computer-architecture `(second edition)`
  - First edition closer to the lecture

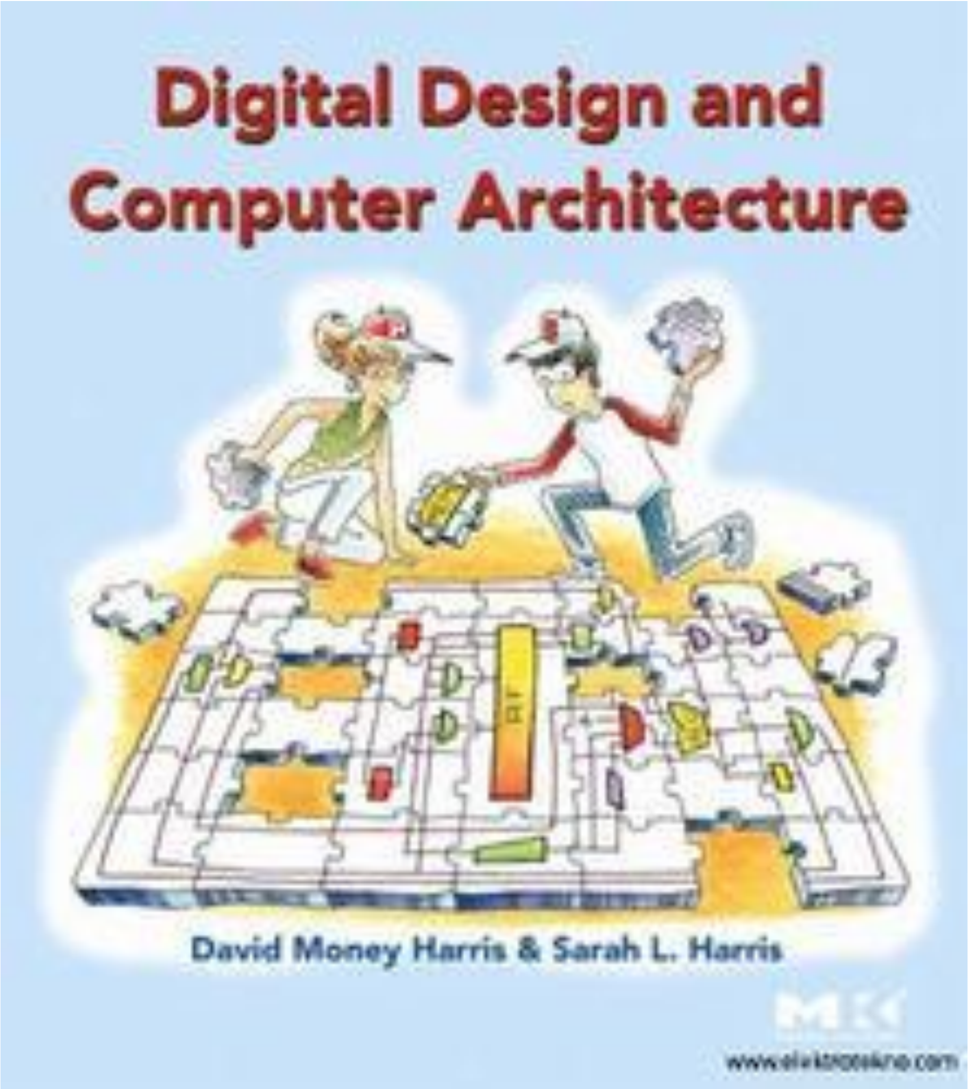
- **PDFs for download from ETH or with ETH VPN from home**





Content

- The story of 1s/0s (ch1)
- Logic Design (ch2-3)
- Verilog (ch4)
- Digital Building Blocks (ch5)
- Assembly (ch6)
- Microarchitecture (ch7)
- Memory Systems (ch8)



The older (2007) edition of the book is more aligned to the lecture

Laboratory Exercises

Goal:
By the end of the lecture (with a little help) design your own processor and make it work!

- **Understand how processors are built**
- **Get an insight of how everything comes together**
- **This is an ambitious goal, so some of the exercises will be challenging**

Laboratory Exercises

- **Every group will work using an FPGA* board**
  - We will have some boards available to take home and experiment on your own if you want
- **We will work in groups of two**
  - You will be able to enroll yourself electronically
- **Four sessions per week**
  - Tuesday 16-18
  - Wednesday 16-18
  - Friday 08-10
  - Friday 10-12

Ataberk will talk about exercises in the second hour

** Field Programmable Gate Array: a generic system that can be programmed to perform any digital function*

Examination

- **180 minute exam within the exam period**
  - Scheduled by the school, we have no influence on the exam time.
- **6 to 8 questions, related to the lectures and labs**
  - Everything covered in the lectures can be part of the exam
- **Six pages of hand-written notes allowed**
  - *No books, papers, computers, phones, calculators, or other electronic devices are allowed. Maximum 6 A4 hand-written pages (i.e., 3 double-sided A4 sheets or 6 one-sided A4 sheets) with notes are allowed.*
- **Accounts for 70 out 100 points of the final grade**
  - 30 points will come from the exercises
- **Previous exams available on class www site**
  - `https://safari.ethz.ch/ddca`

From Prof. Mutlu: Learning & Exam

- **We will enable you to learn + prepare you for the exam**

- **My suggestions:**
  - focus on **understanding, learning**, mastering the material
    - lectures, readings, labs, HWs all enable this and prepare you
  - reinforce problem solving skills with homeworks
  - do **not** worry about the exam while listening to lectures
    - most of you will pass this course (historically >80%)

- **We will release a lot of material to help you with the exam**
  - Problem solving sessions
  - Exam guidance
  - All past exams (and basic solutions) are already online

If You Need Help

- **Write an e-mail to:** **digitaltechnik@lists.inf.ethz.ch**
  - All lecturers and assistants will receive this e-mail

preferred

- **Moodle forums**
  - Possible to post anonymously
- **Write directly to (put DDCA in subject):**
  - Frank Gürkaynak: kgf@iis.ee.ethz.ch

Further Information

- **Lecture web page**

  `https://safari.ethz.ch/ddca`
  `https://safari.ethz.ch/digitaltechnik`

- **Now is the time for questions**