Introduction to Computer Architecture
Chapter 1

Computer Abstractions and Technology

Hyungmin Cho

Department of Computer Science and Engineering
Sungkyunkwan University

Goal of the Course

- Understand how computer hardware runs software
- Understand what determines the performance of a computer

Class Schedule

Week 1	Computer Abstraction and Technology	Chapter 1
Week 2	Performance	Chapter 1
Week 3	Instructions: Language of the Computer	Chapter 2
Week 4	Instructions: Language of the Computer	Chapter 2
Week 5	Instructions: Language of the Computer	Chapter 2
Week 6	The Processor	Chapter 4.1 – 4.4
Week 7	The Processor	Chapter 4.1 – 4.4
Week 8	Midterm Exam (Exact date: TBD)	Chapter 1, 2, 4.1 – 4.4
	Pipelining	Chapter 4.5
Week 9	Pipelined Datapath and Control	Chapter 4.6, 4.7
Week 10	Pipeline Hazards	Chapter 4.7, 4.8
Week 11	Memory Hierarchy	Chapter 5
Week 12	Memory Hierarchy	Chapter 5
Week 13	Memory Hierarchy	Chapter 5
Week 14	Arithmetic for Computers	Chapter 3
Week 15	Final Exam (Exact date: TBD)	

Class Logistics

- Grading criteria
  - Midterm exam: 30%
  - Final exam: 40%
  - Programming assignments (3~4 assignments): 30%

- Class attendance
  - Each student can miss class up to 4 times without any penalty (No need to provide excuse)
  - Fail to attend more than 4 classes: 10% deduction
  - Fail to attend more than ¼ of the total classes: F

- **If you fail to attend one of the exams, you will get "F" automatically.**

Class Communications

- Questions related to the lecture contents:
  - Questions are welcomed anytime during the lecture.
  - You may ask questions through i-Campus messages, but we recommend using the open Q&A board rather than private messages.

- Homework (programming assignment) questions:
  - Homework questions are allowed only in the designated “discussion” thread in i-Campus.

- Office hour:
  - Tuesday, 4PM – 5PM
  - Room 85470 (산학협력센터)

1만라인 프로젝트 과목?

- If you're 소프트웨어학과 2022+학번, one of the conditions for your graduation is fulfilling the "1만라인프로젝트" requirement.

- To satisfy the requirement, you need to get "B" grade or higher from at least two "1만라인프로젝트" courses.

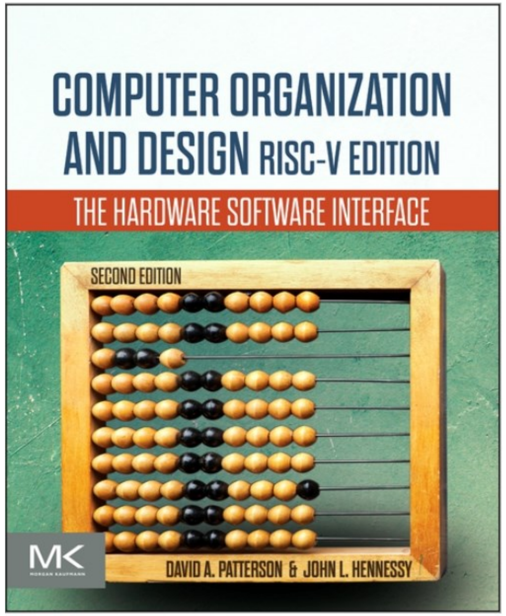
- In those courses, one or more class assignments will be based on a large (more than 10K lines of code) open-source based project.

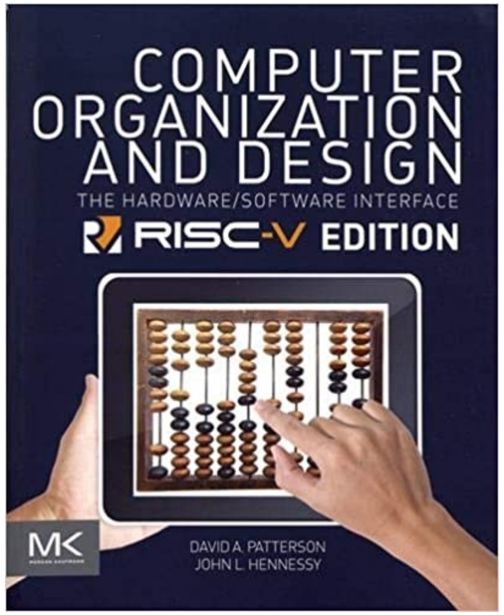
- In this computer architecture class, the last assignment will be the 1만라인프로젝트 assignment.
  - **If you do not submit this last assignment, you cannot get grade B or higher.**

Textbook

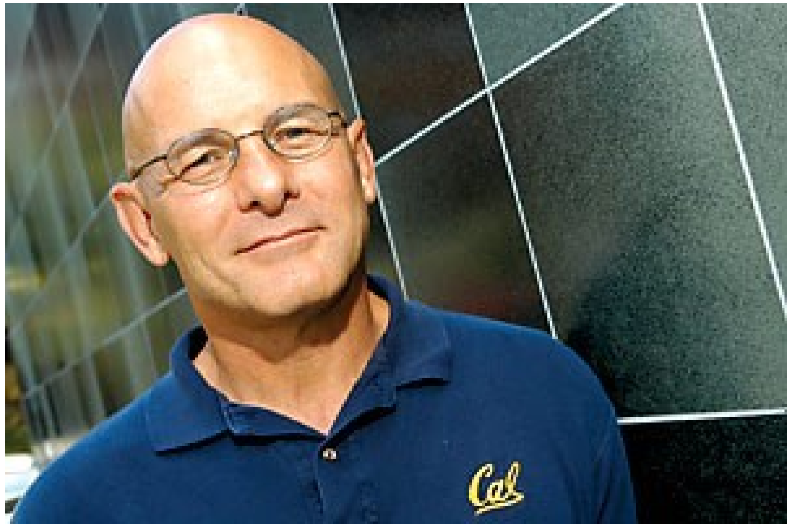
- Computer Organization and Design **RISC-V Edition**: The Hardware/Software Interface
  - 2nd edition
  - 1st edition is also okay

- The “Patterson & Hennessy” book





John L. Hennessy / David A. Patterson



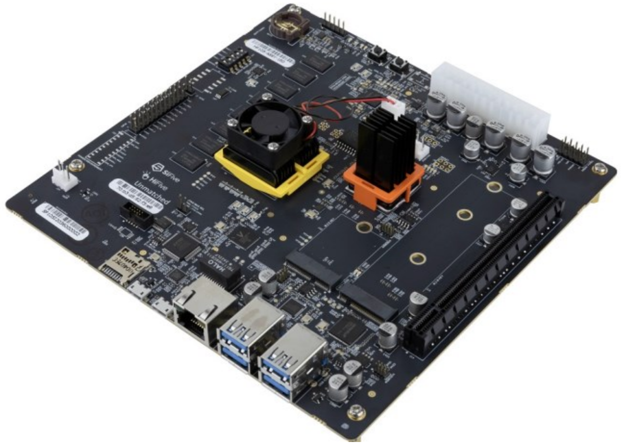
- RISC processor architecture
- **Turing award 2017** – “For pioneering a systematic, quantitative approach to the design and evaluation of computer architectures with enduring impact on the microprocessor industry”

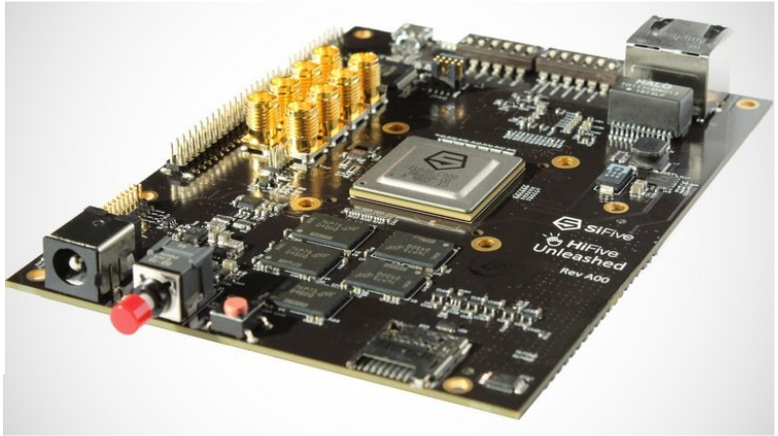
RISC-V CPU

- Modern x86-based computers are very complicated
  - Complex instruction set

- This course is based on RISC-V CPUs
  - Open standard
    - Many open-source & commercial implementations exist
    - Simple (reduced) instruction set

RISC-V®





SweRV Core

RISC-V CPU

- I don’t have a RISC-V computer! How can we test the RISC-V software?
  - Simulators are available!

- Why not ARM?
  - ARM is little more complex than MIPS
    - The basic principles are similar
  - ARM ISA is not free

x86?

- Processor family from Intel and AMD
  - Derived from the model numbers of the first few generations:
    - 8086, 80286, 80386, 80486 → x86
  - x86-16: 16-bit processor
  - x86-32 (aka IA32): 32-bit processor
  - x86-64 (aka AMD64): 64-bit processor

- How intel (and AMD) maintained the market dominance?
  - Backward compatibility with legacy software
  - Complexity

Programming Assignment (Plan)

- Building a RISC-V **simulator**
  - A software that mimics the behavior of a RISC-V CPU

- Multiple-stage assignment
  - We will keep adding functionalities to the simulator
  - If you fail an assignment, it will be hard to complete the next assignment.

- Details TBA

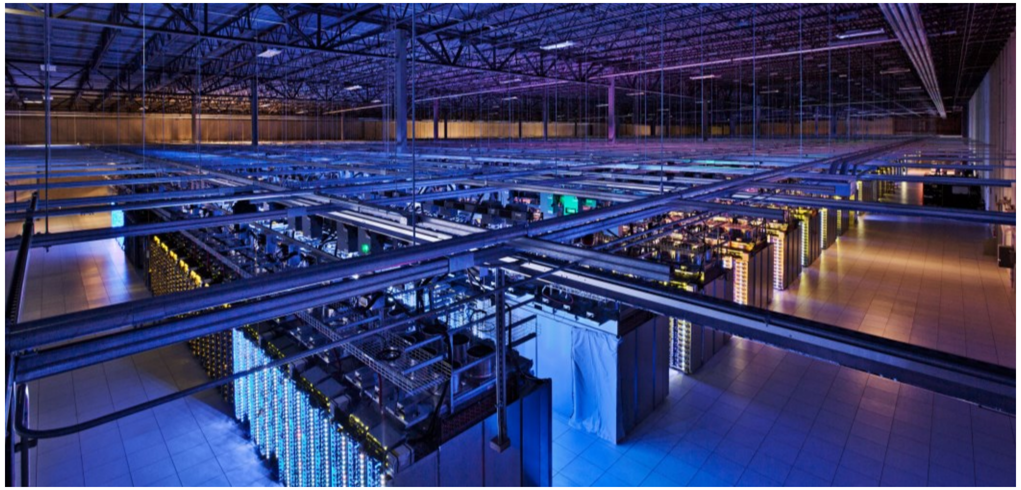
Classes of Computers

- Personal computers
  - General purpose
  - Human interaction




Classes of Computers

- Server computers
  - High capacity, reliability
  - Basic building block is not so different from personal computers



Google datacenter

Classes of Computers

- Supercomputers
  - Specialized in high computational performance
    - Scientific experiments, weather forecast, etc.

- Fastest supercomputer in the world (as of June 2022)
  - **TOP 500** list (top500.org)
  - “Frontier” @ Oak Ridge National Laboratory
  - HPE Cray EX235a
  - 9,248 × 64-core AMD EPYC
  - 36,992 × 220 AMD Instinct MI250X GPUs
  - Performance
    - 1,100,000,000,000,000,000 calculations per second (1.1 Exa FLOPS)
    - Compared to a normal PC (Intel i5): 11,000,000×

Classes of Computers

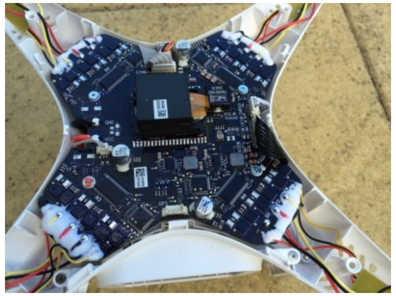
- Most energy-efficient supercomputer in the world (as of June 2022)
  - **GREEN 500** list (top500.org)
  - “Frontier TDS” @ Oak Ridge National Laboratory
    - A smaller version of the Frontier supercomputer
  - Performance: 19.2 petaFlops
  - Power consumption: 309 kW
  - Energy efficiency = 19.2peta / 309k = 62 GFlops/watts

- Second energy-efficient supercomputer in the world
  - “Frontier” again..
  - Energy efficiency = 1.1exa / 21M = 52 GFlops/watts

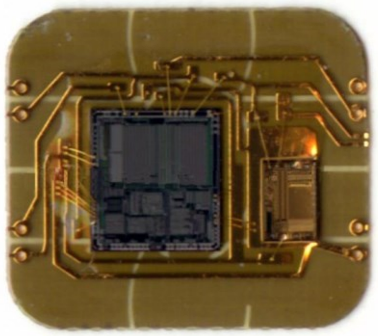
- 6^{th} place?
  - SSC-21 Scalable Module @ Samsung Electronics
  - Energy efficiency = 2.27peta / 103k = 34 GFlops/watts

Classes of Computers

- Embedded computers
  - Operates as a part of the other systems (automobile, home appliances)
  - Significant restrictions in size, power consumption, performance, etc…



USIM



8bit or 16bit CPU
Simple operating system
1~2KB RAM
EEPROM: Storage

[www.extremetech.com]

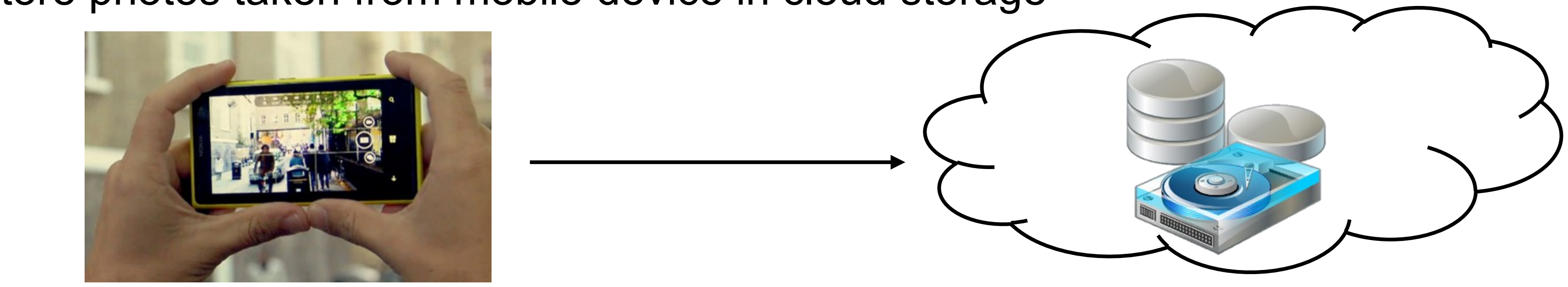
The PostPC Era

- Personal Mobile Device (PMD)
  - Battery operated
  - Connects to the Internet
  - Hundreds of dollars
  - Smart phones, tablets, electronic glasses

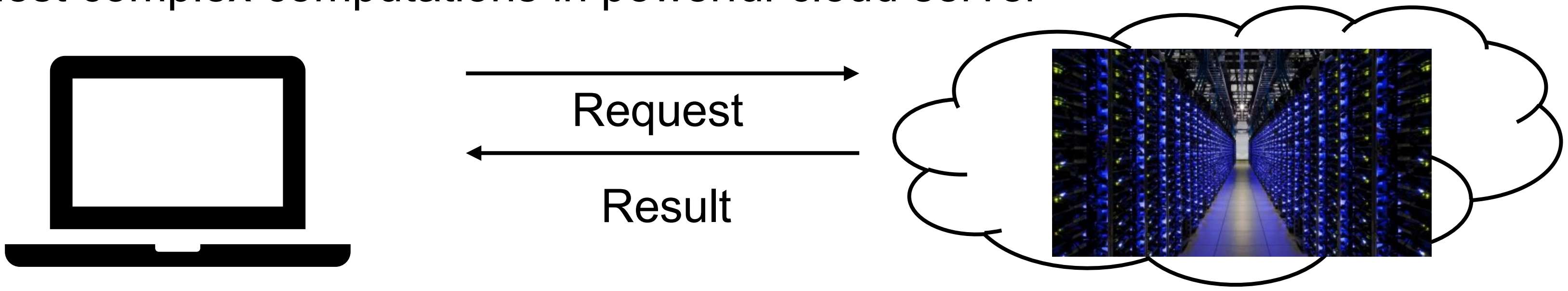
- Cloud computing
  - Information processing & storage is not limited in specific hardware
  - Processing with the many computer systems over the network

Cloud Computing Examples

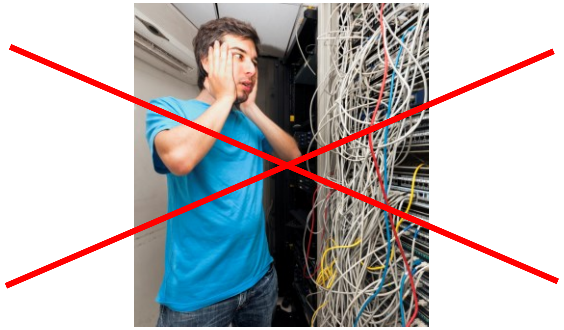
1. Store photos taken from mobile device in cloud storage



2. Request complex computations in powerful cloud server

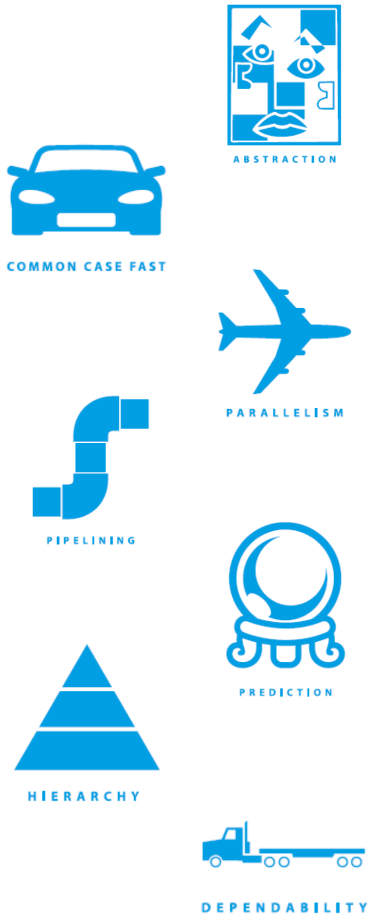


3. Use cloud-based virtual machines instead of managing your own servers

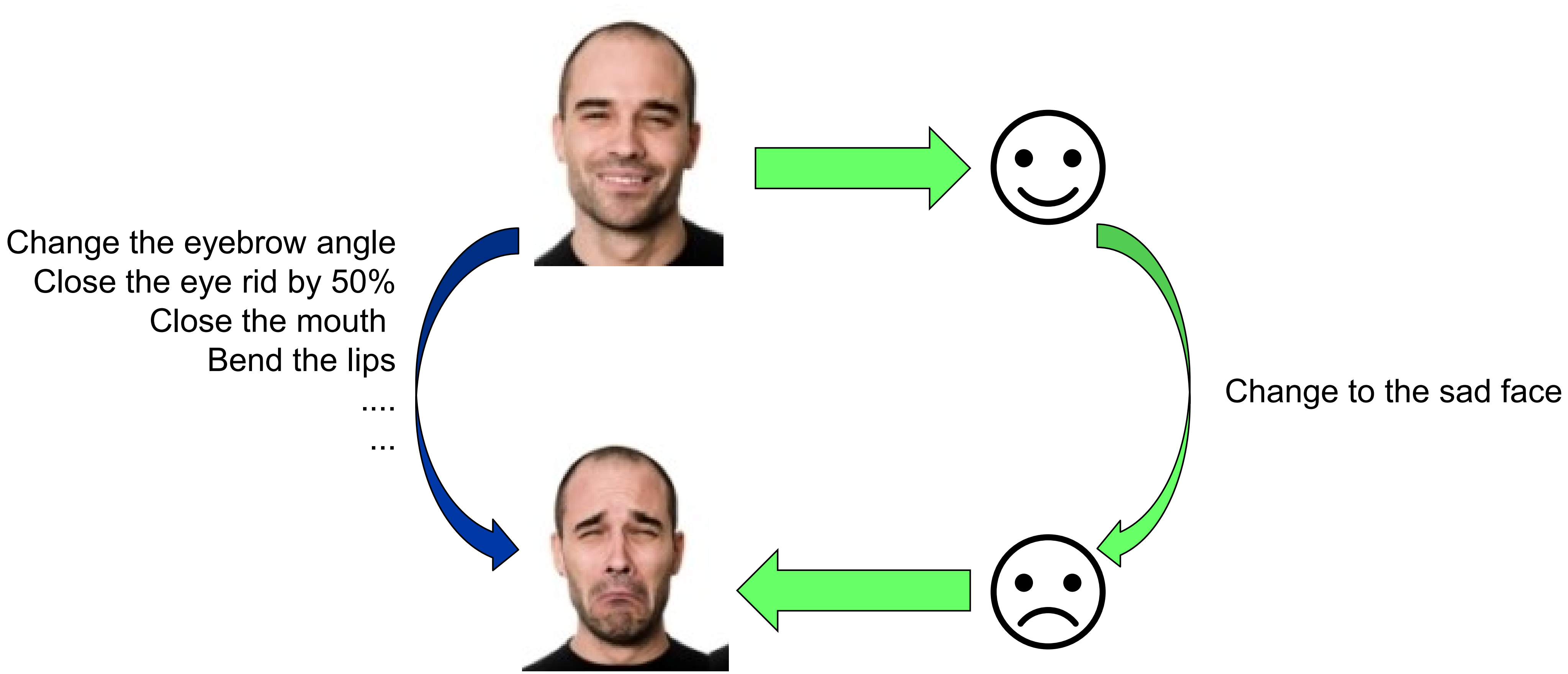


Seven Great Ideas in Computer Architecture

- 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

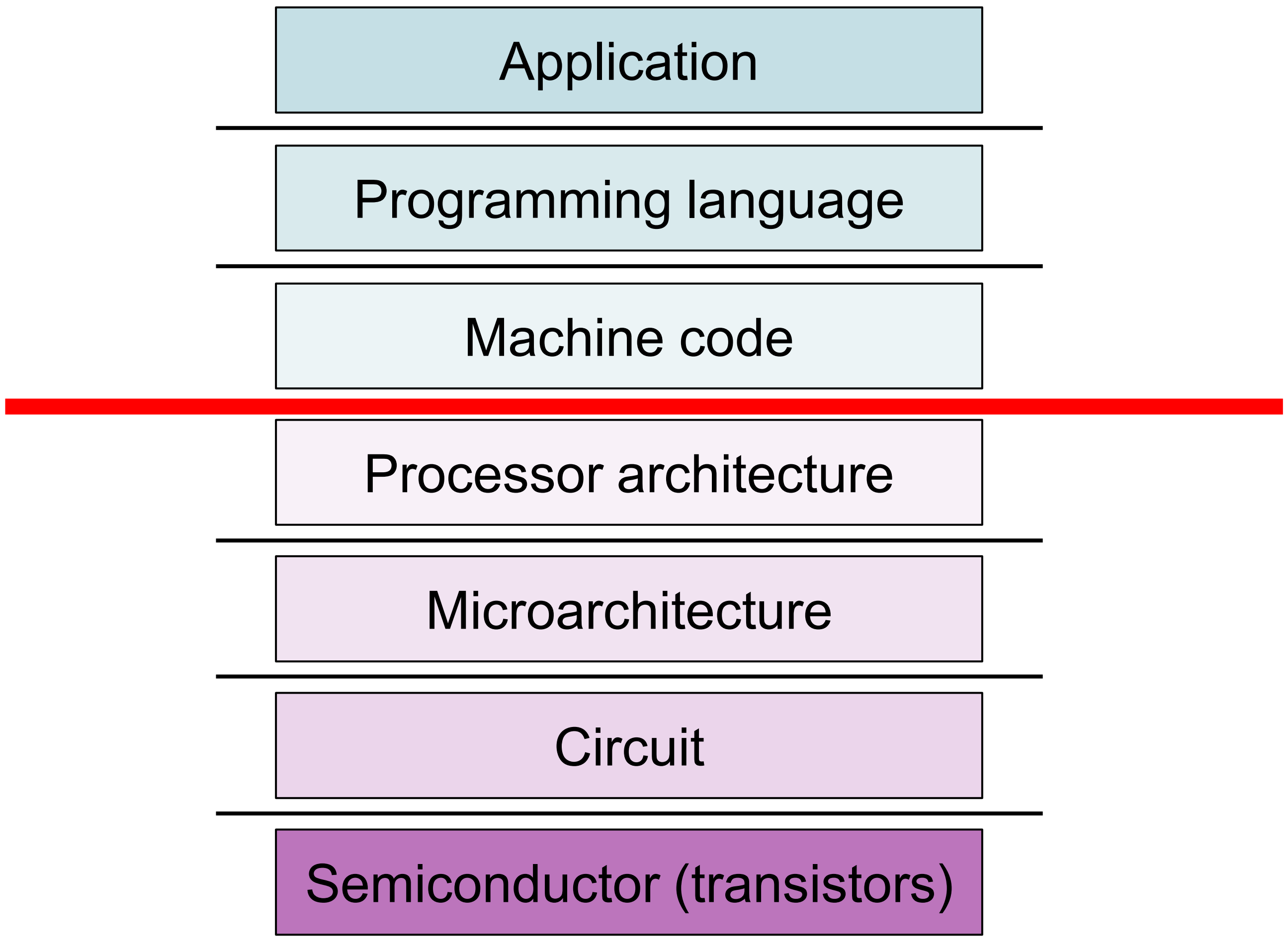


1. Use *Abstraction* to Simplify Design



- Modern computer designs are very complex
- Abstraction: Hide the lower-level details to simply the model

Abstractions in Computer System



Abstractions

2. Make the *Common Case Fast*

- Make the common case fast

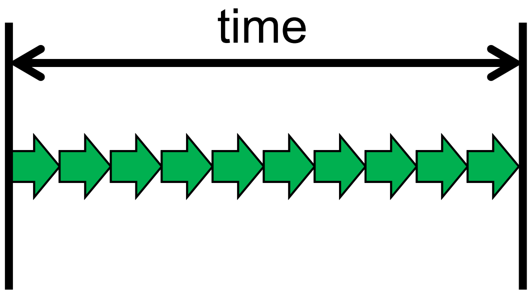
  vs.

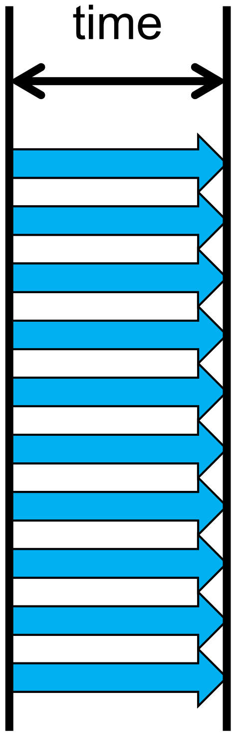
  Optimize the rare case

- Common case is often simpler than the rare case.

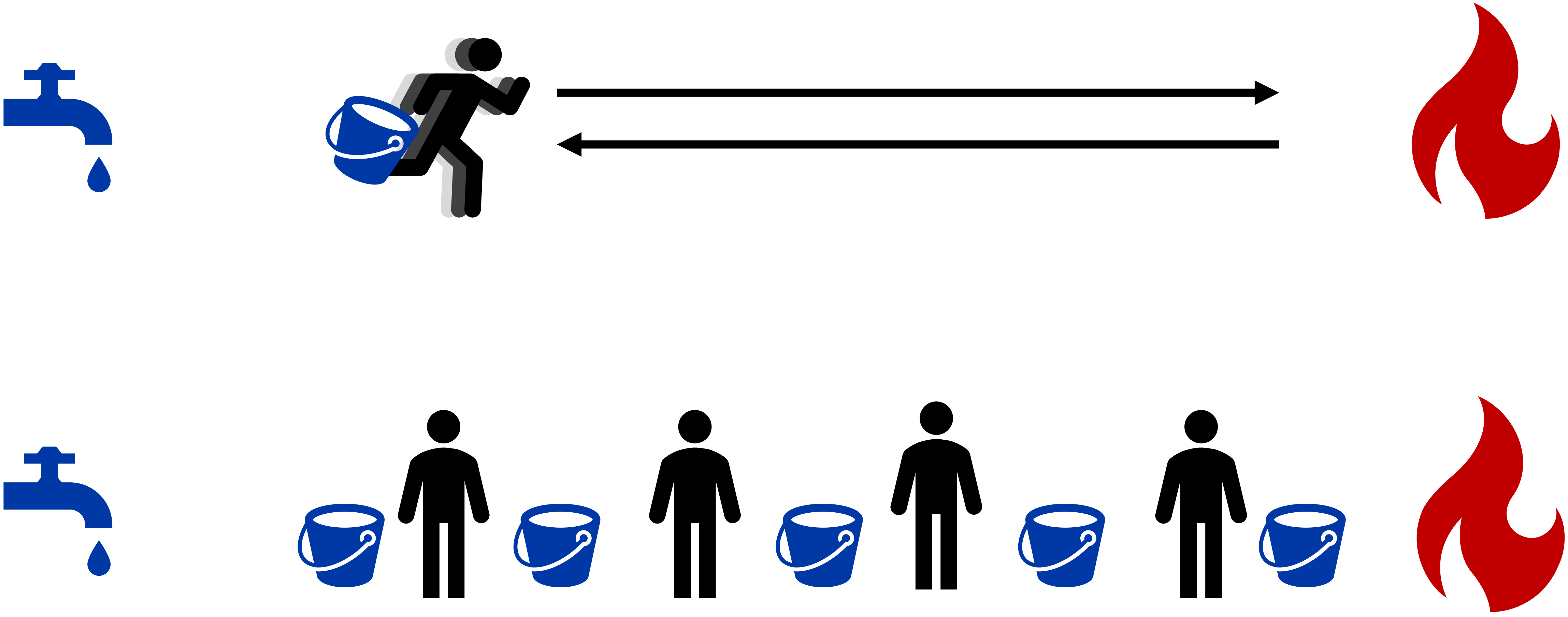
3. Performance *via Parallelism*

PARALLELISM

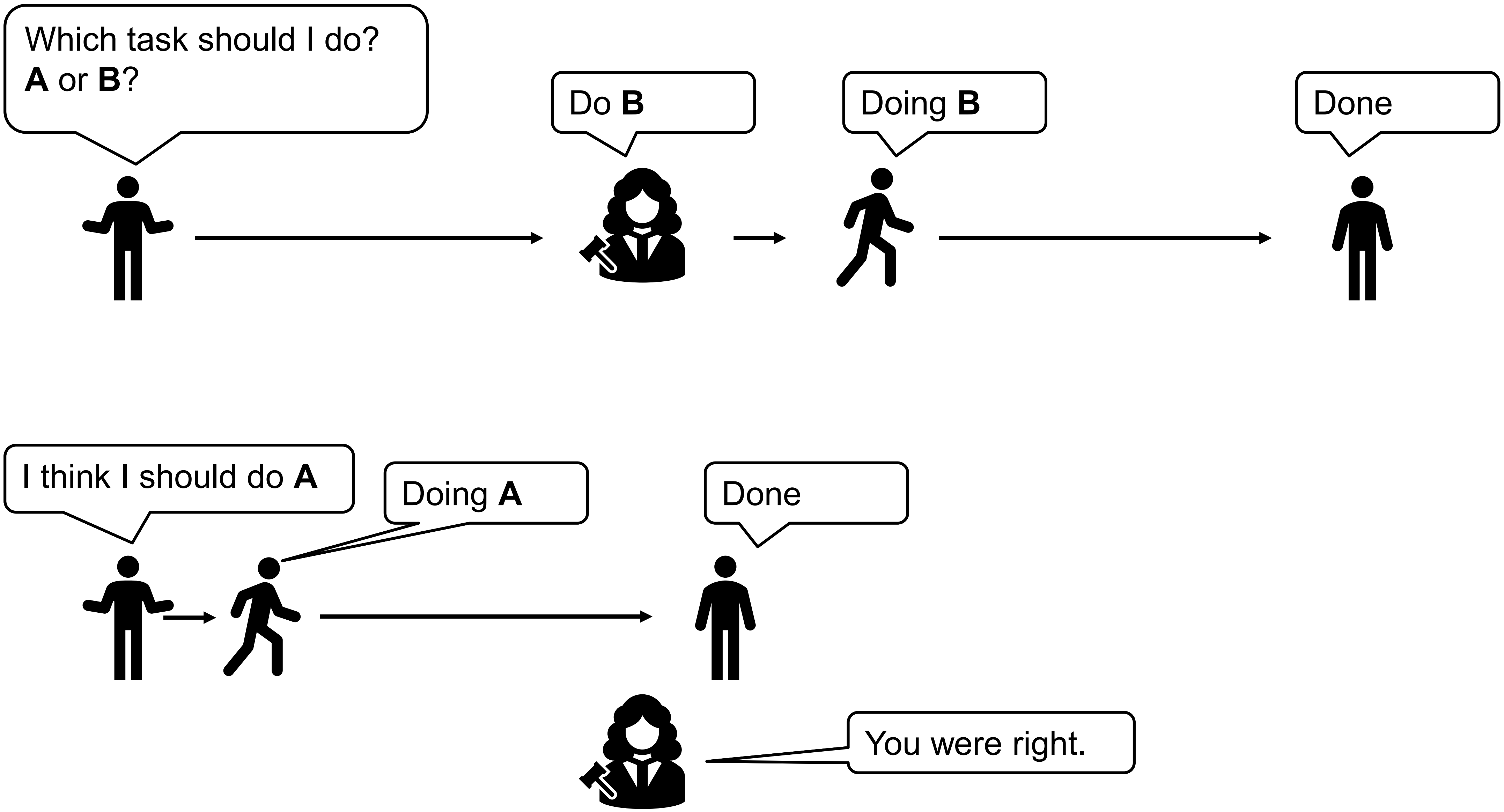




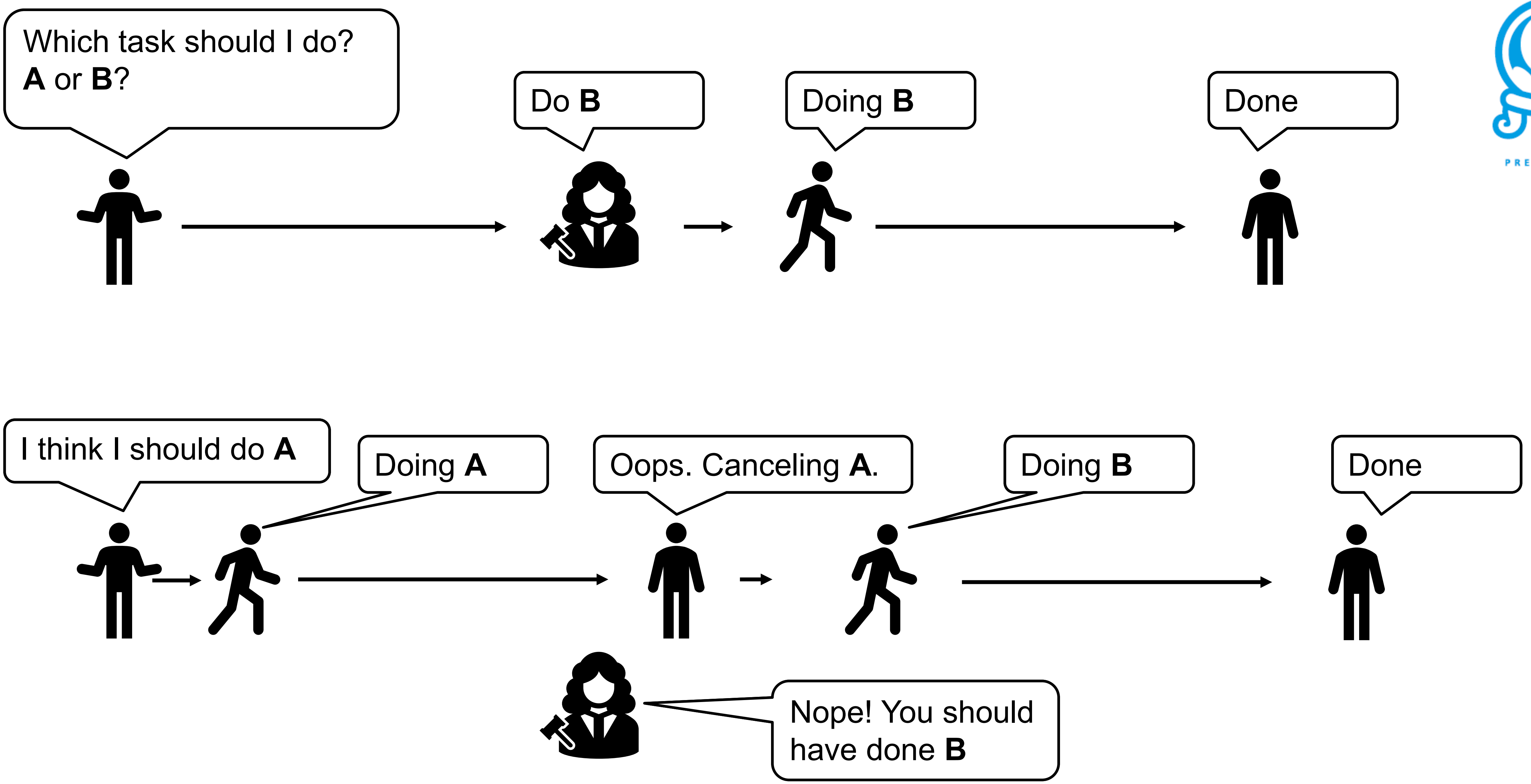
4. Performance *via Pipelining*



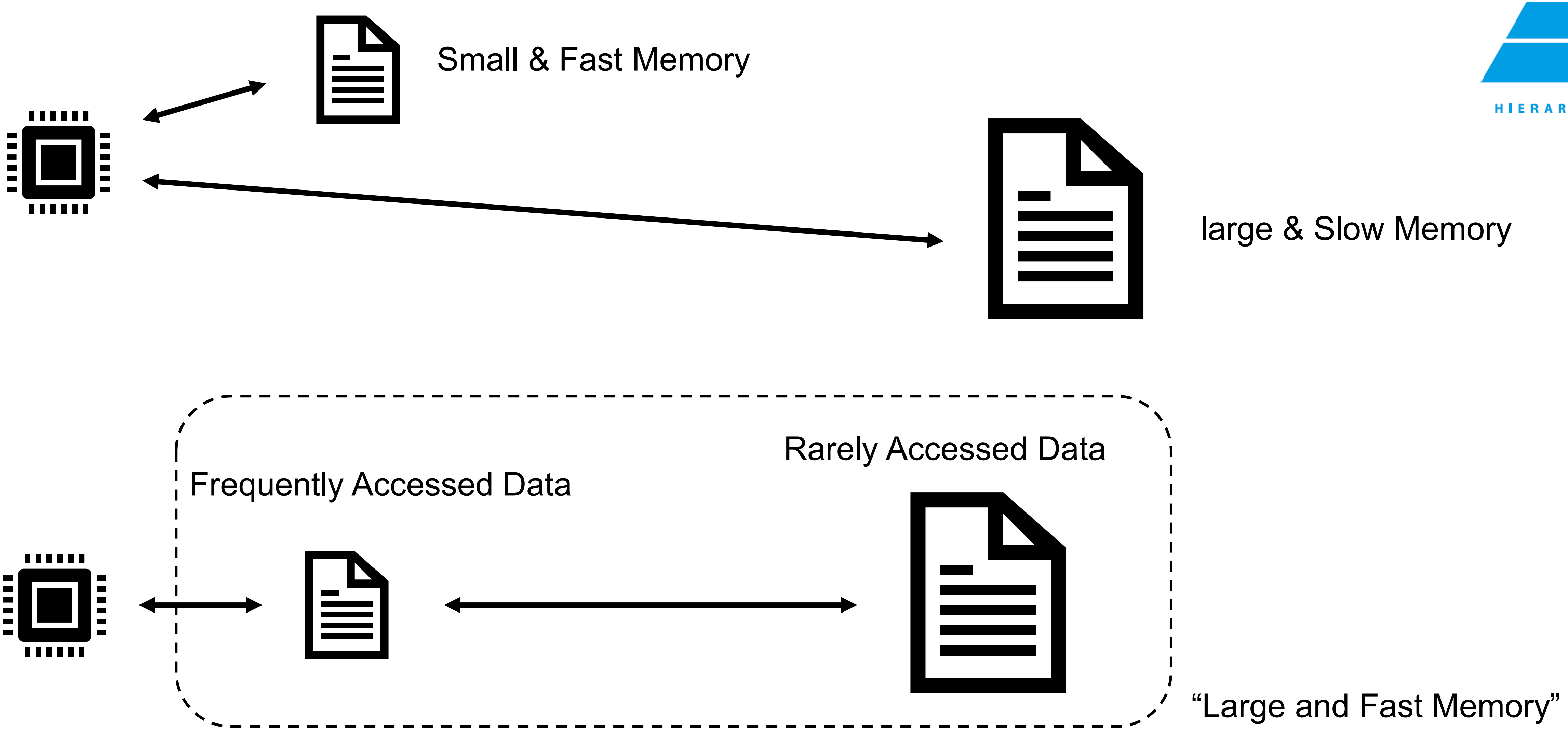
5. Performance *via Prediction*



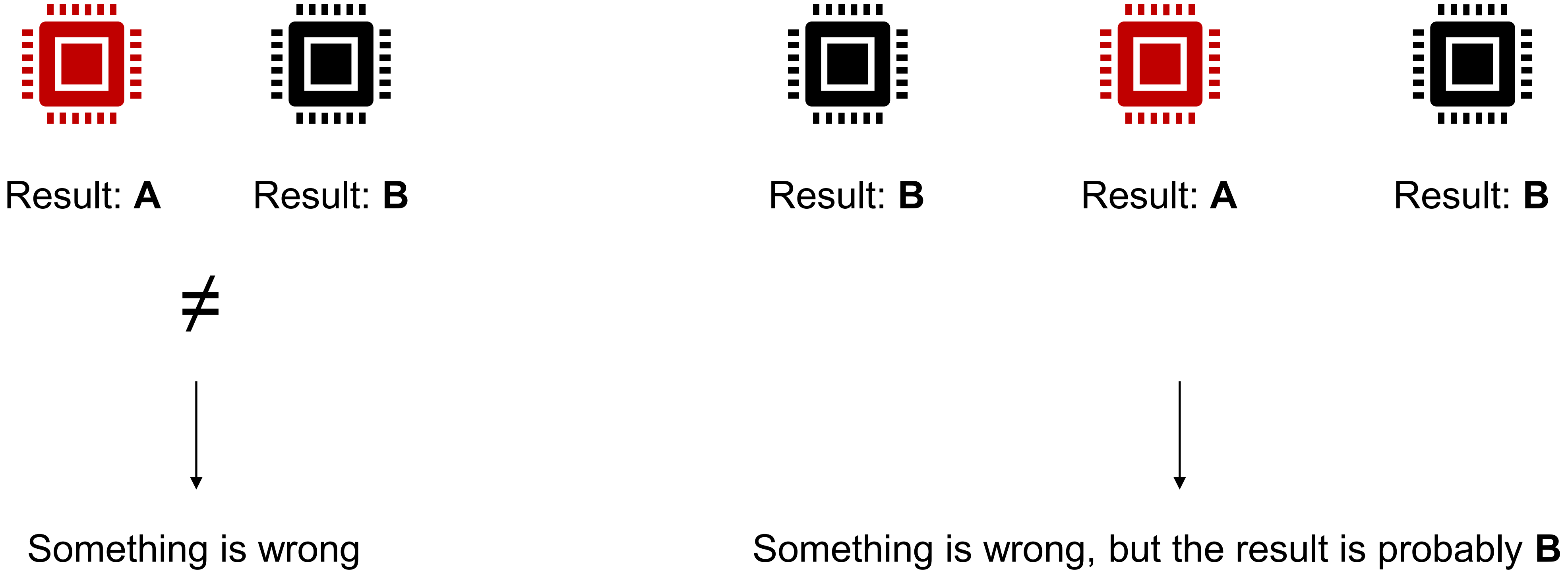
5. Performance *via Prediction*



6. *Hierarchy* of Memories



7. *Dependability via* Redundancy



Hardware Abstractions

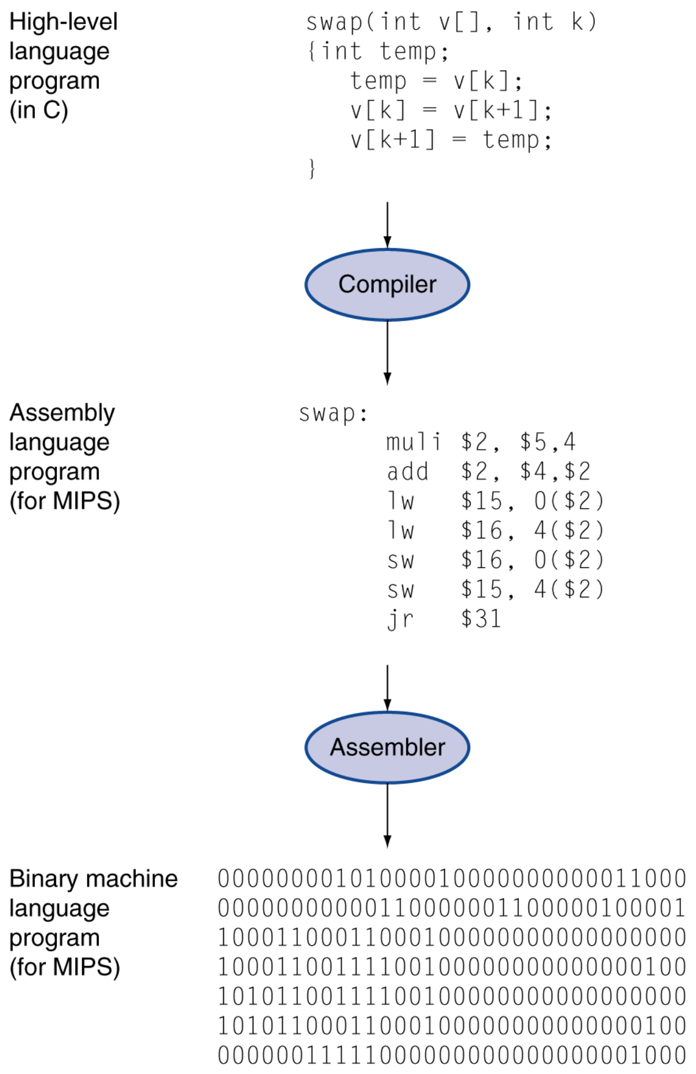
- Instruction set architecture (ISA)
  - How software abstracts hardware
  - The hardware/software interface
    - CPU instructions
  - x86 (Intel), ARM, SPARC, etc…
- Application binary interface (ABI)
  - The ISA plus system software (OS) interface
  - e.g., x86 Windows application vs. ARM iOS application

ISAs

- x86
  - Used in PC and servers
- POWER, PowerPC
  - Used in Mac, but now it is replaced by Intel Processor
- SPARC
  - Workstations & database servers (Sun microsystems is acquired by Oracle)
- ARM
  - Most popular in embedded world
  - Designed by ARM, but implemented by many manufactures (Apple, Samsung, etc…)
- RISC-V
  - Similar to MIPS
  - Open ISA (no license fees), Many open-source implementations

Levels of Program Code

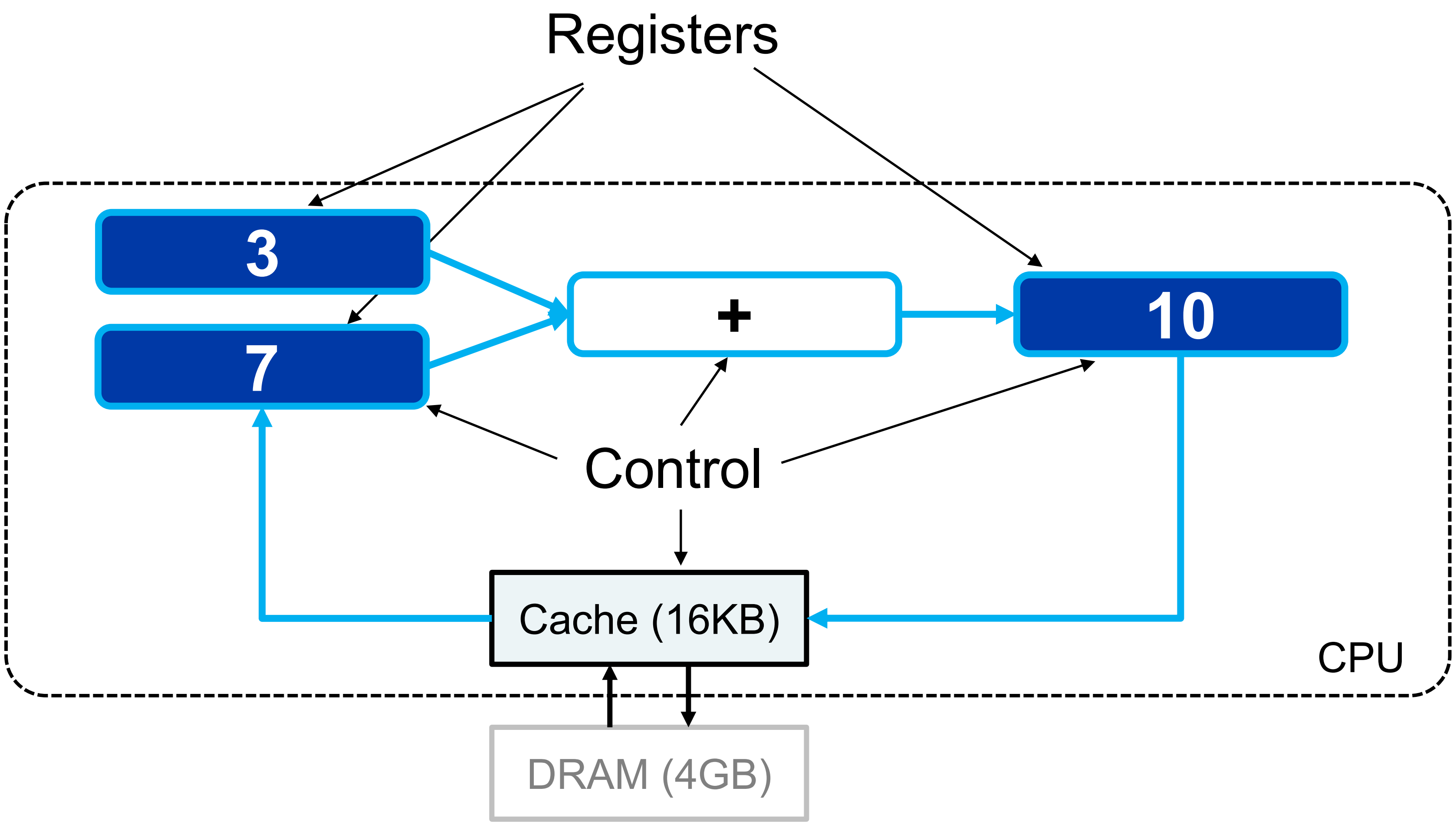
- High-level language
  - Level of abstraction closer to problem domain
  - Productivity
  - Portability
- Assembly language
  - Instructions is human-readable format
- Machine code
  - How the instructions are stored in hardware



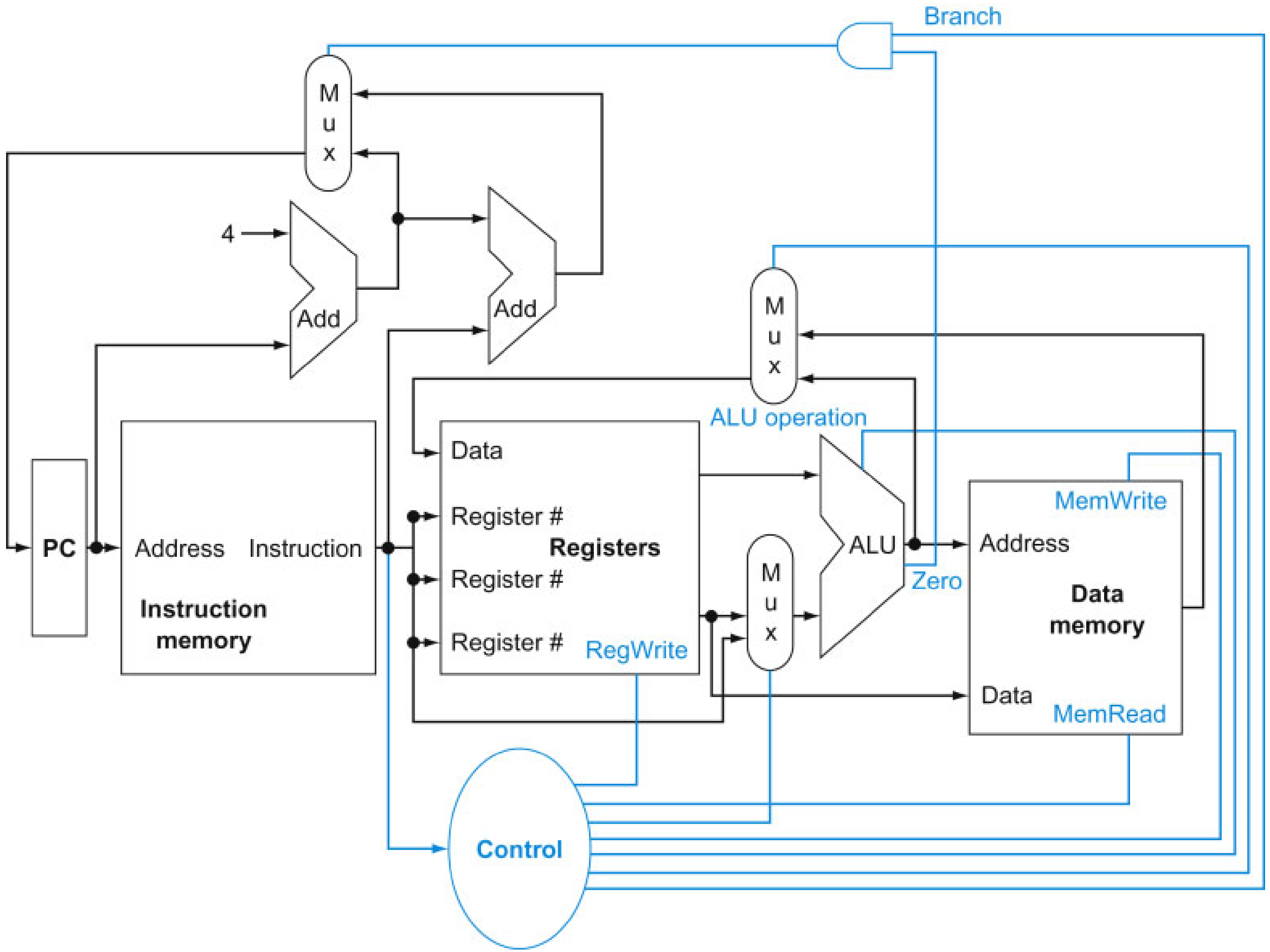
Inside the Processor (CPU)

- **Datapath**: Data moves between the components
- **Control**: Manages how the components process the given data
- **Registers**: Temporary data storage inside the CPU
- **Cache memory**:
  - DRAM Memory: Main data storage during program execution
  - Frequently accessed data is brought into the CPU memory

Inside the Processor (CPU)

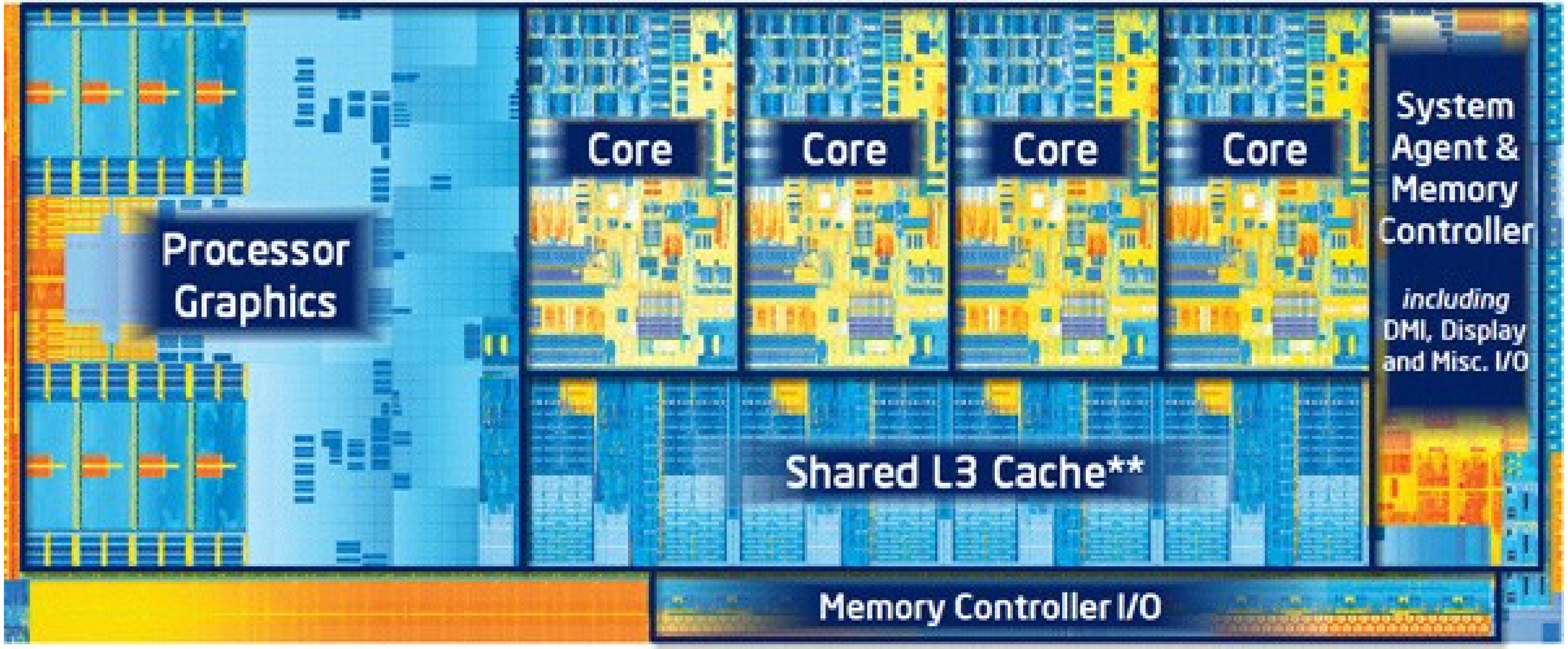


Processor Architecture



Inside the Processor

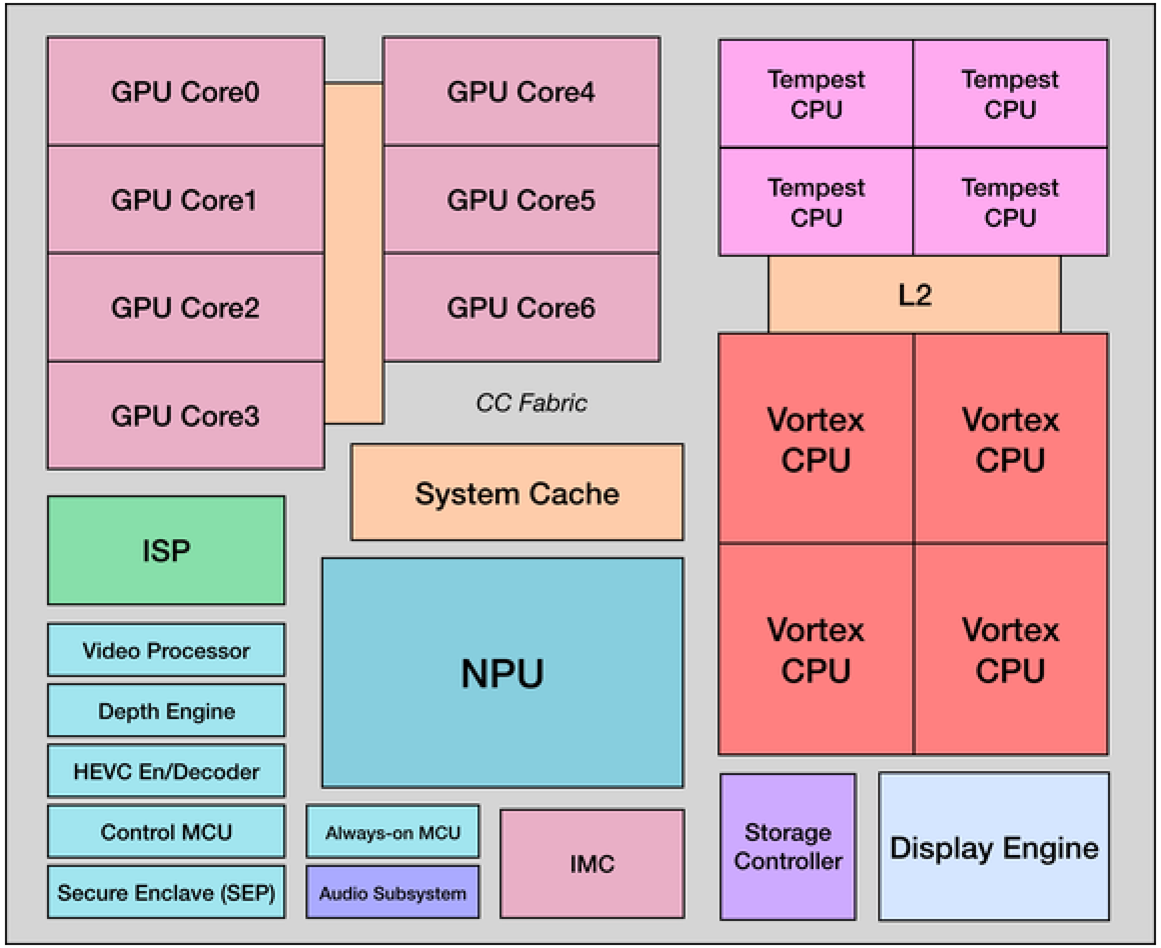
- Intel i7 "Ivy bridge"



[anandtech.com]

Inside the Processor

- Apple A12X



[wikichip.org]