

Computer Architecture
Lecture Week 1 (Part I)

Introduction to Computer Architecture

Artem Burmyakov

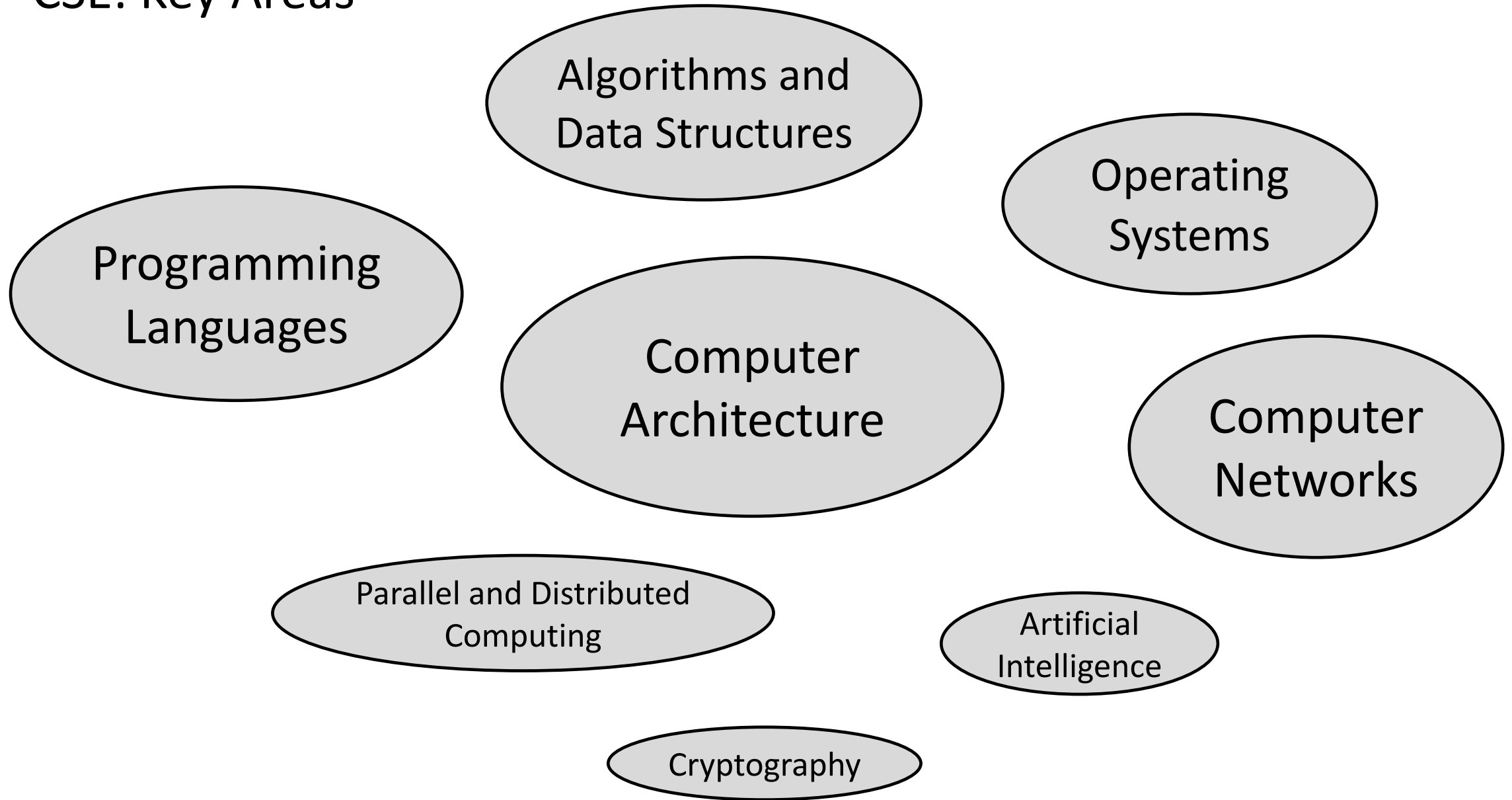
September 04, 2026



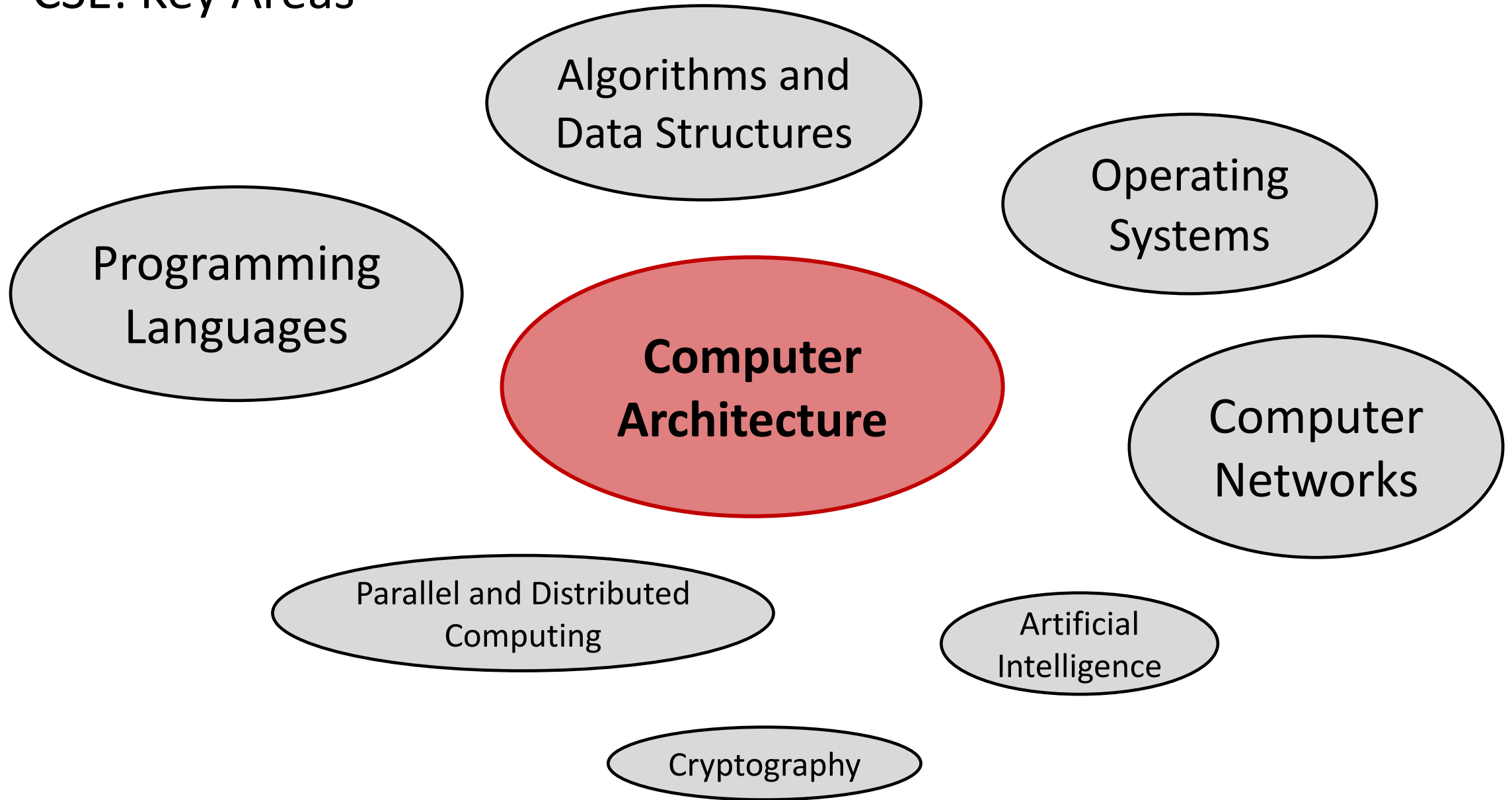
INNOPOLIS
UNIVERSITY

Computer Science and Engineering (CSE): Key Areas

CSE: Key Areas



CSE: Key Areas



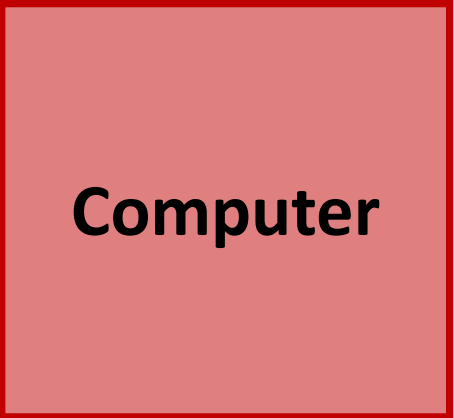
What is a “computer”?

What is a “computer”?

A machine that is programmed to automatically perform sequences of arithmetic/logic instructions with some input data and to output result *[Wikipedia]*

What is a “computer”?

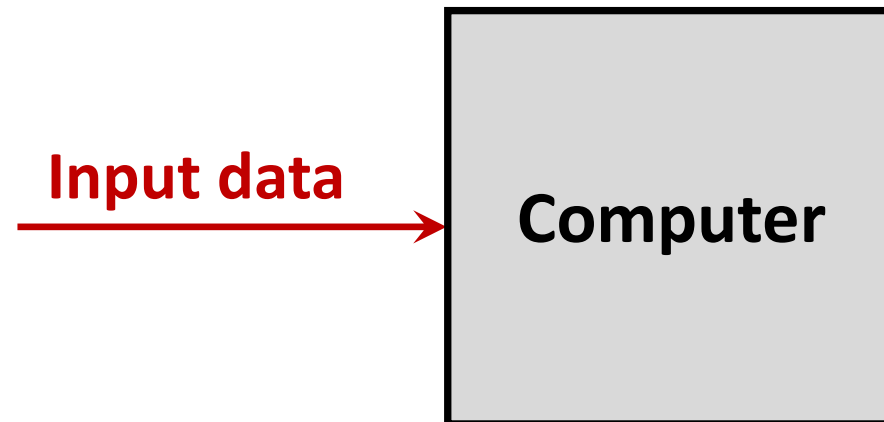
A **machine (= physical device)** that is programmed to automatically perform sequences of arithmetic/logic instructions with some input data and to output result *[Wikipedia]*



Computer

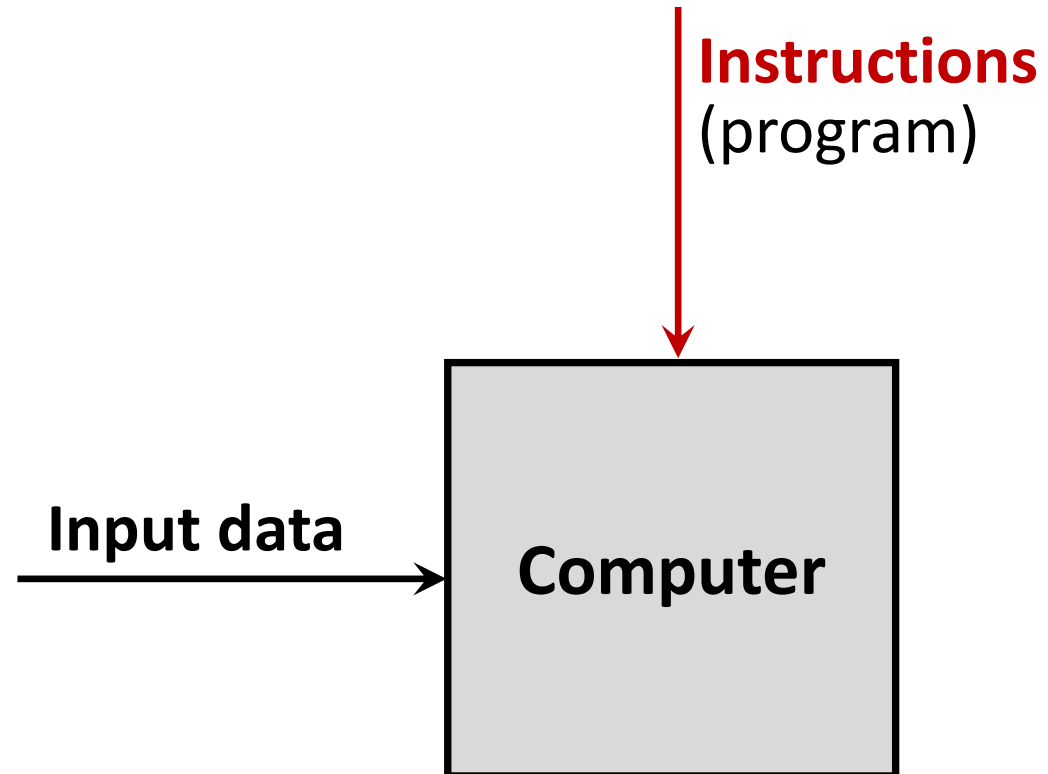
What is a “computer”?

A **machine** that is programmed to automatically perform sequences of arithmetic/logic instructions with some **input data** and to output result *[Wikipedia]*



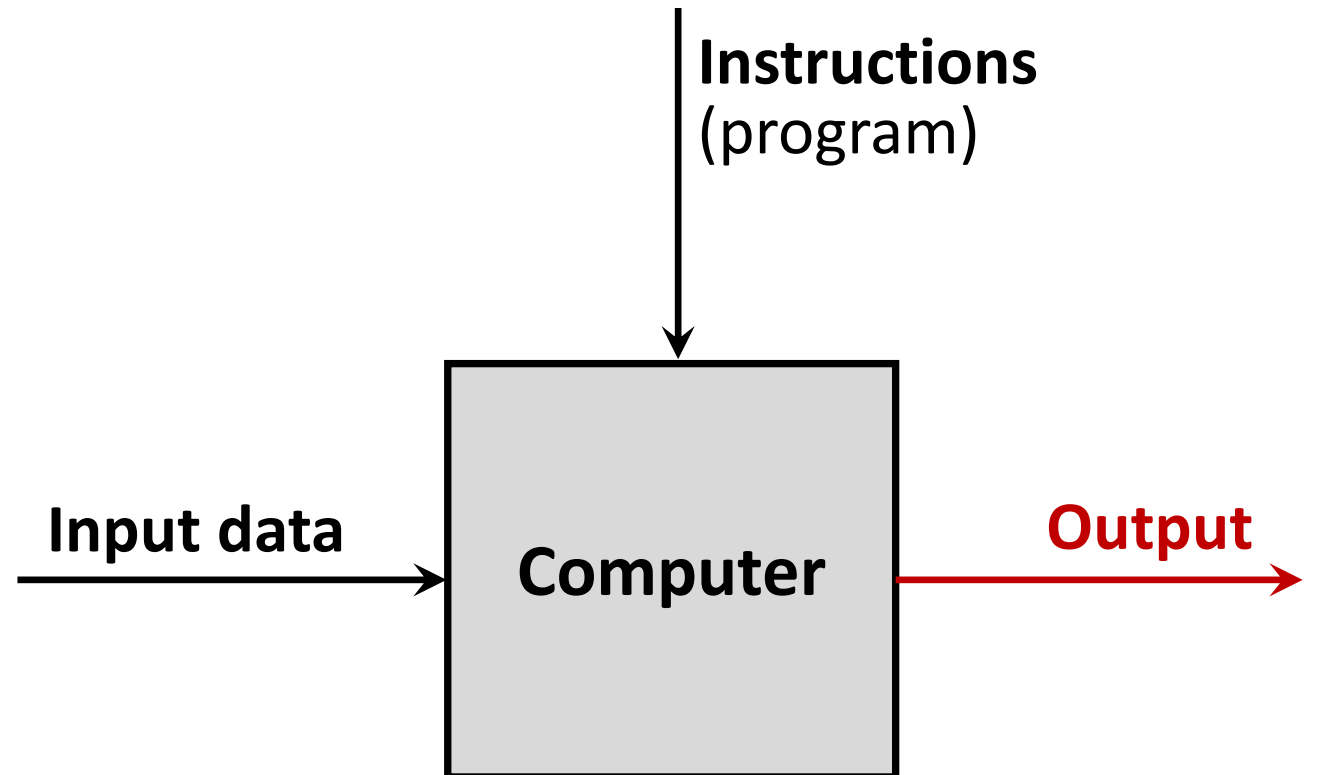
What is a “computer”?

A **machine** that is programmed to automatically perform sequences of arithmetic/logic **instructions** with some **input data** and to output result *[Wikipedia]*



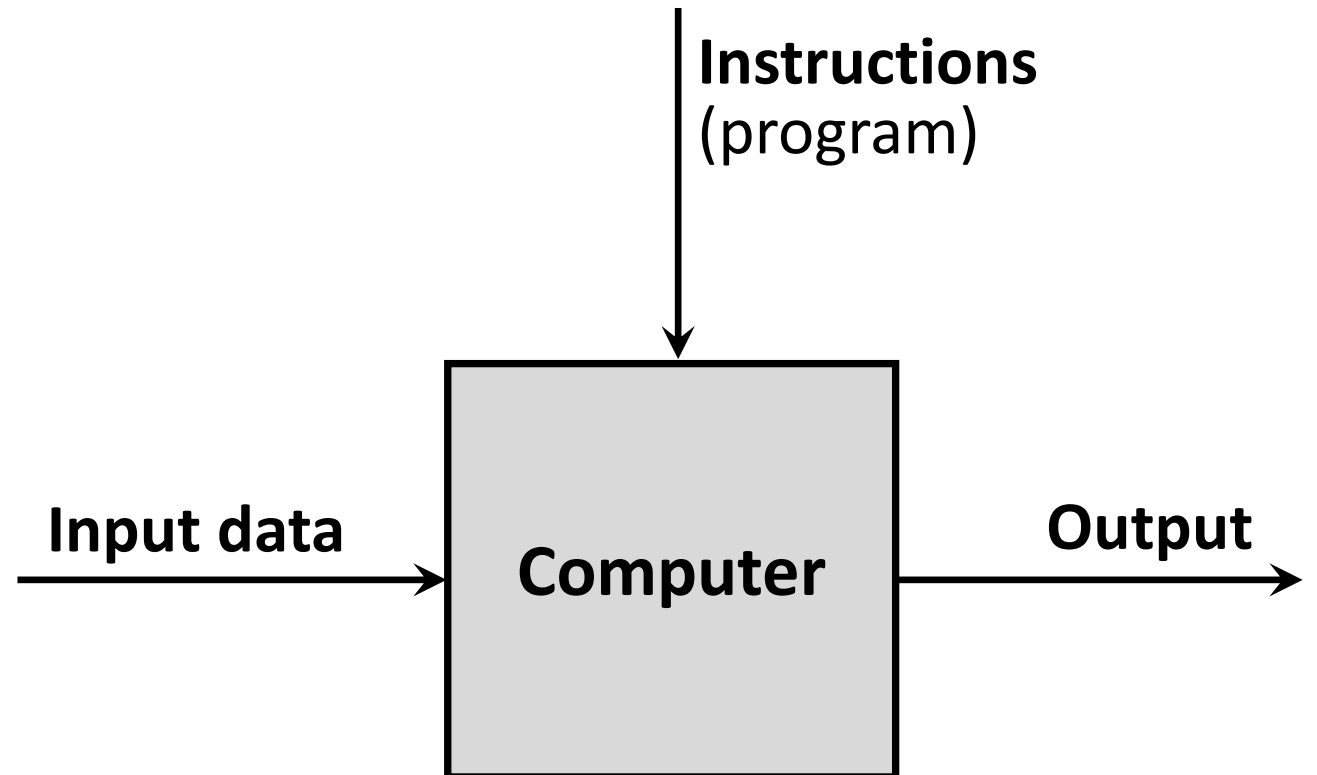
What is a “computer”?

A **machine** that is programmed to automatically perform sequences of arithmetic/logic **instructions** with some **input data** and to **output** result *[Wikipedia]*



What is a “computer”?

A **machine** that is programmed to automatically perform sequences of arithmetic/logic **instructions** with some **input data** and to **output** result *[Wikipedia]*



“Computer”: some definitions

A machine that is programmed to automatically perform sequences of arithmetic/logic instructions with some input data and to output result *[Wikipedia]*

An electronic device that operates under the control of a program stored in its memory

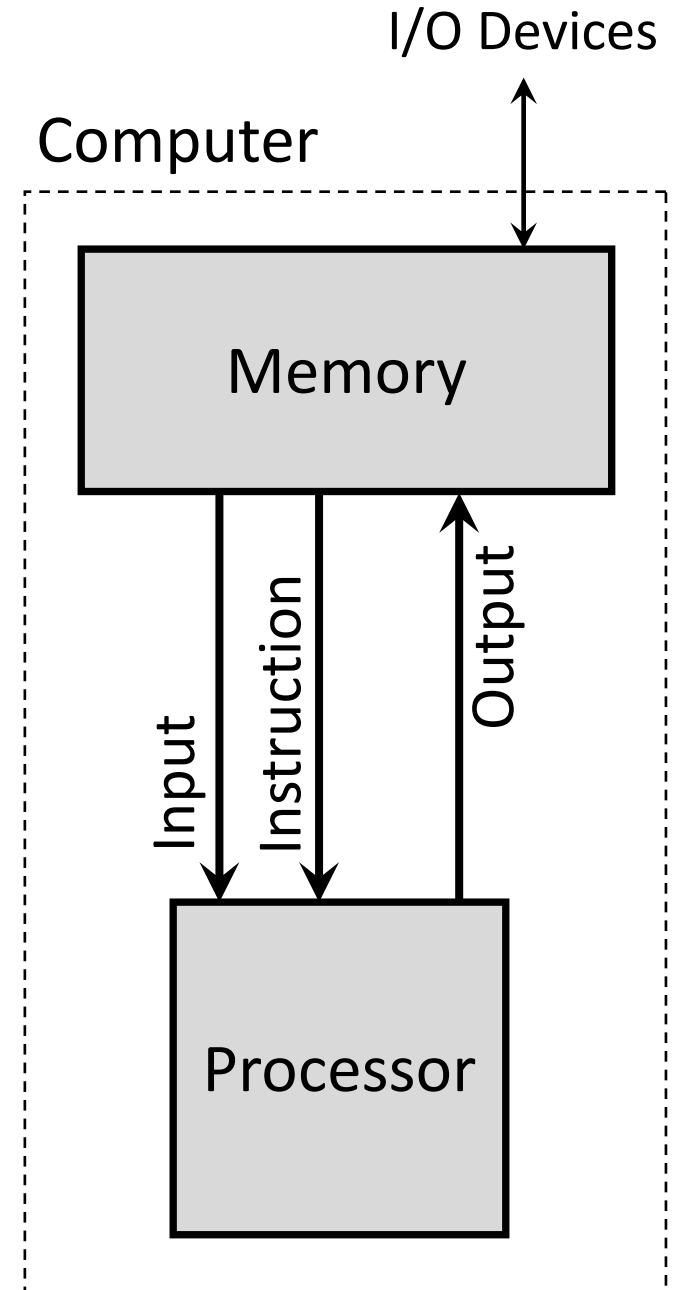
[Britannica]

“Computer”: some definitions

A machine that is programmed to automatically perform sequences of arithmetic/logic instructions with some input data and to output result *[Wikipedia]*

An electronic device that operates under the control of a program stored in its memory

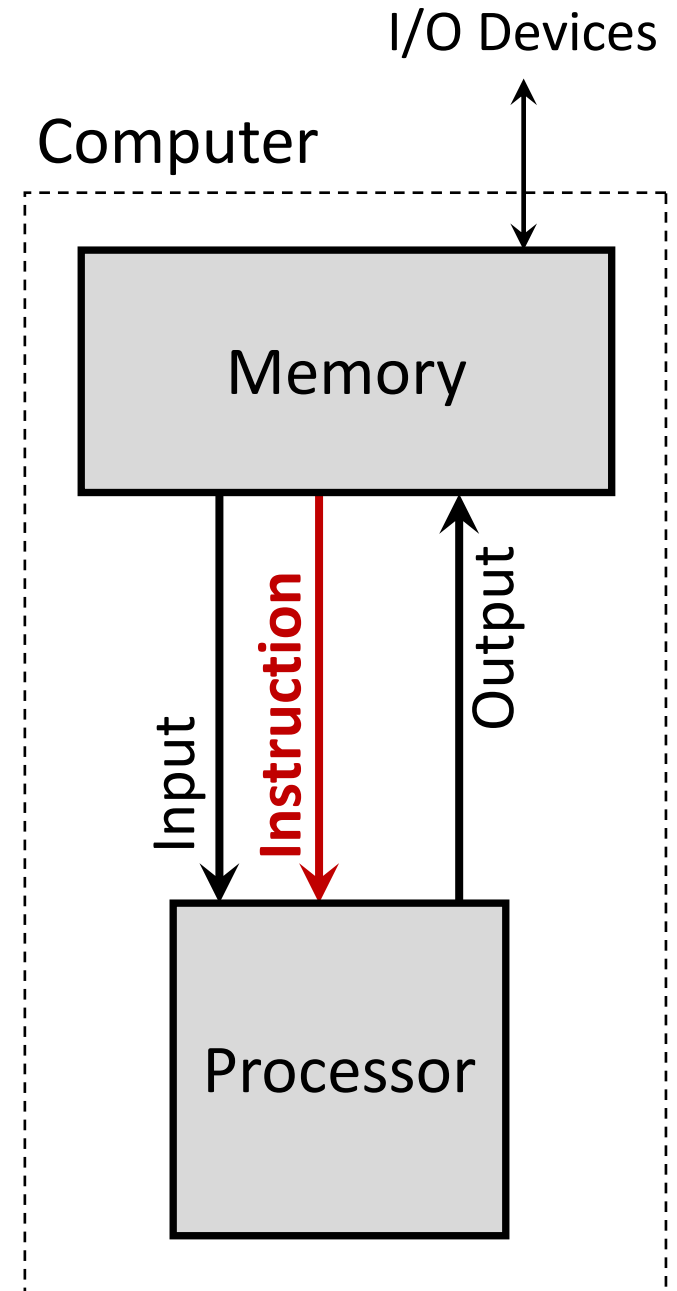
[Britannica]



“Computer”: some definitions

A machine that is programmed to automatically perform sequences of arithmetic/logic instructions with some input data and to output result *[Wikipedia]*

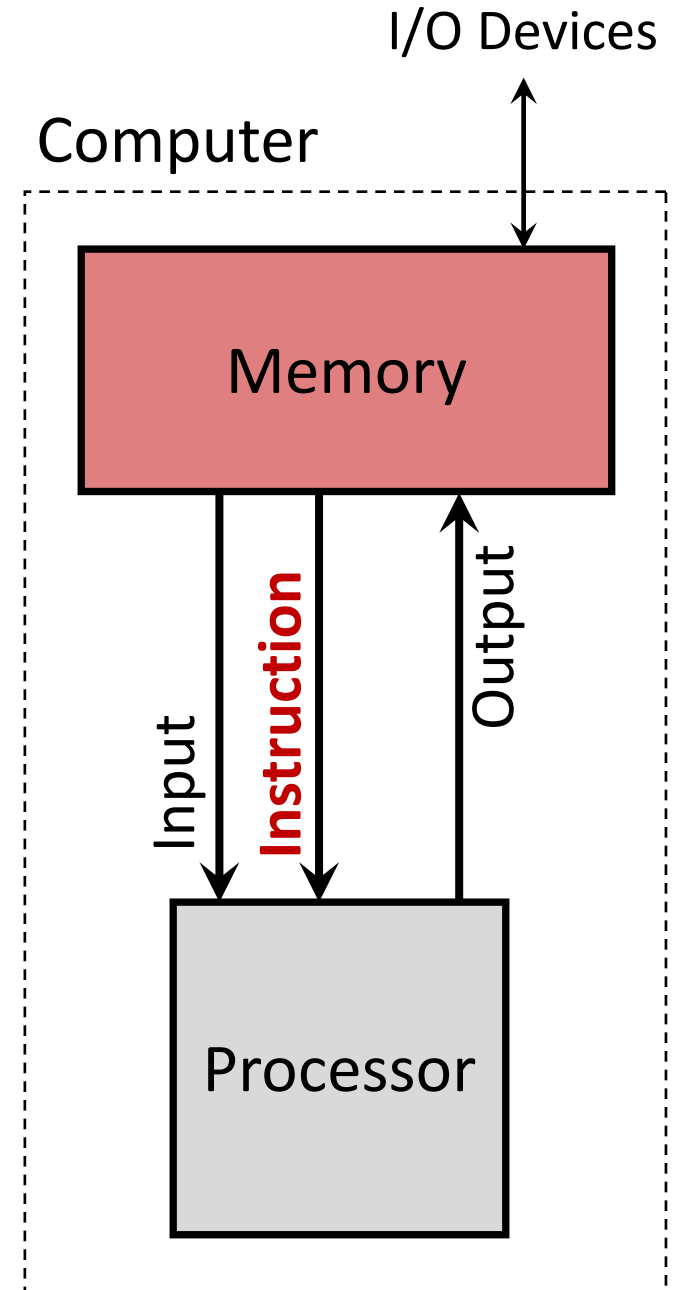
An electronic device that **operates under** the control of a **program** stored in its memory *[Britannica]*



“Computer”: some definitions

A machine that is programmed to automatically perform sequences of arithmetic/logic instructions with some input data and to output result *[Wikipedia]*

An electronic device that operates under the control of a **program** stored **in its memory** *[Britannica]*



“Computer”: some definitions

A machine that is programmed to automatically perform sequences of arithmetic/logic instructions with some input data and to output result *[Wikipedia]*

An electronic device that operates under the control of a program stored in its memory *[Britannica]*

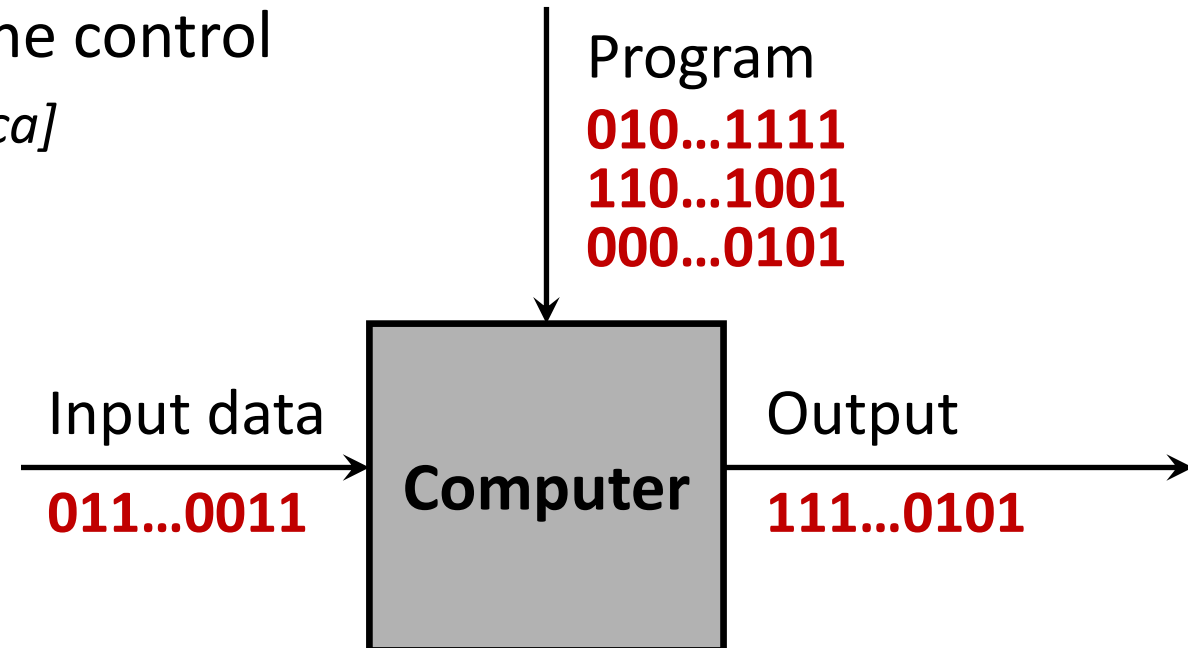
A device that accepts binary input and manipulates it for some binary result following a sequence of instructions
[TechTarget]

“Computer”: some definitions

A machine that is programmed to automatically perform sequences of arithmetic/logic instructions with some input data and to output result *[Wikipedia]*

An electronic device that operates under the control of a program stored in its memory *[Britannica]*

A device that accepts binary input and manipulates it for some binary result following a sequence of instructions
[TechTarget]

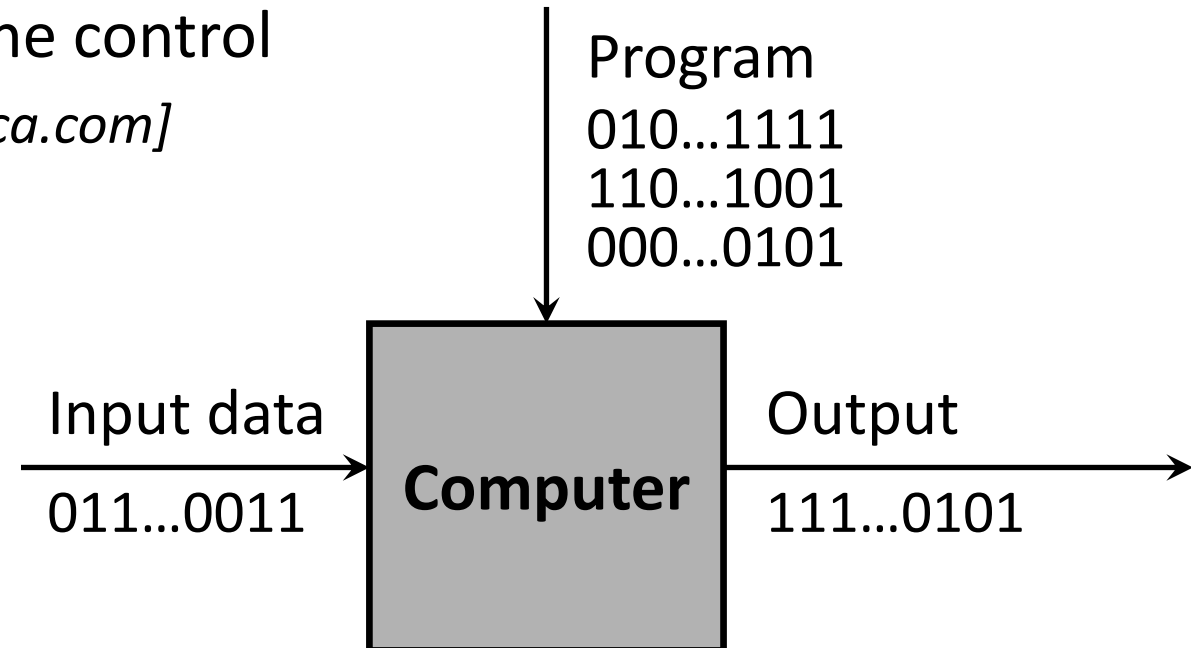


What is a “computer”?

A machine that is programmed to automatically perform sequences of arithmetic/logic instructions with some input data and to output result *[Wikipedia]*

An electronic device that operates under the control of a program stored in its memory *[Britannica.com]*

A device that accepts binary input and manipulates it for some binary result following a sequence of instructions *[TechTarget]*



What is “Computer Architecture”?

Computer Architecture is about

- **Hardware organization of computers**



Computer Architecture is about

- **Hardware organization of computers**



Name	MacBook Pro (2)
Processor	2.6 GHz 6-Core Intel Core i7
Graphics	Intel UHD Graphics 630 1536 MB
Memory	16 GB 2667 MHz DDR4
Serial number	C02CN0JJMD6M
Coverage Expired	Details...

Computer Architecture is about

- **Hardware organization of computers**



System Information File Edit Window Help

Hardware Overview:

Model Name:	MacBook Pro
Model Identifier:	MacBookPro16,1
Processor Name:	6-Core Intel Core i7
Processor Speed:	2.6 GHz
Number of Processors:	1
Total Number of Cores:	6
L2 Cache (per Core):	256 KB
L3 Cache:	12 MB
Hyper-Threading Technology:	Enabled
Memory:	16 GB
System Firmware Version:	2075.140.4.0.0 (iBridge: 22.16.16083.0.0.0)
OS Loader Version:	583~2210
Serial Number (system):	C02CN0JJMD6M
Hardware UUID:	A887F2B1-050A-5883-A03B-4DB7E745656B
Provisioning UDID:	A887F2B1-050A-5883-A03B-4DB7E745656B
Activation Lock Status:	Enabled

Hardware

- ATA
- Apple Pay
- Audio
- Bluetooth
- Camera
- Card Reader
- Controller
- Diagnostics
- Disc Burning
- Ethernet
- Fibre Channel
- FireWire
- Graphics/Displays
- Memory
- NVMeExpress
- PCI
- Parallel SCSI
- Power
- Printers
- SAS
- SATA
- SPI
- Storage
- Thunderbolt/USB4
- USB

Network

- Firewall
- Locations
- Volumes
- Wi-Fi

Software

- Accessibility
- Applications
- Developer
- Disabled Software
- Extensions
- Fonts
- Frameworks
- Installations
- Language & Region
- Legacy Software

Computer Architecture is about

- Hardware organization of computers
- **Hardware/Software integration principles**



Computer Architecture is about

- Hardware organization of computers
- **Hardware/Software integration principles**

What is “integration”?



Computer Architecture is about

- Hardware organization of computers
- **Hardware/Software integration principles**



```
1 #include <stdio.h>
2
3 int calculate_sum(int a[5]) {
4     int s = 0;
5     for (int i = 0; i < 5; i++) s += a[i];
6     return s;
7 }
8
9 int main() {
10     int numbers[5];
11     for (int i = 0; i < 5; i++) {
12         scanf("%d", &numbers[i]);
13     }
14     ...
```

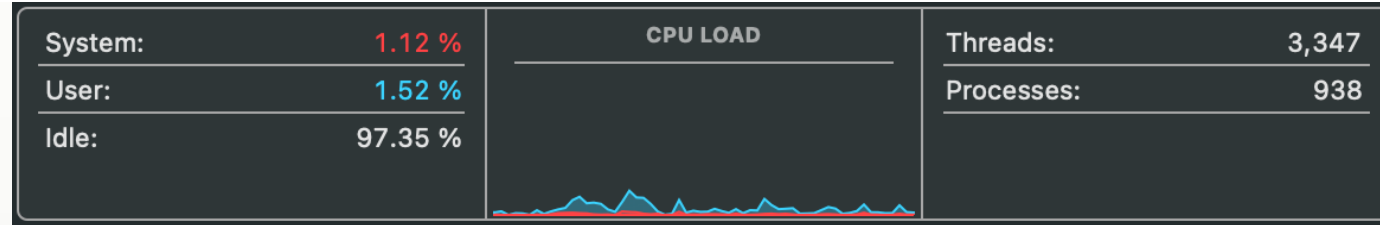
Computer Architecture is about

- Hardware organization of computers
- Hardware/Software integration principles
- **Performance-related aspects of computers**



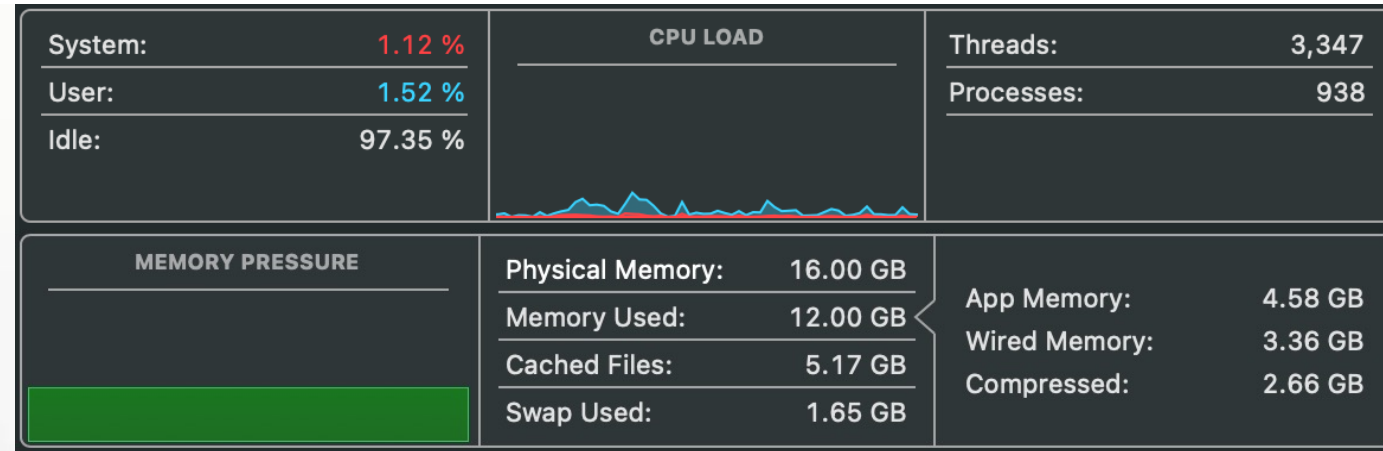
Computer Architecture is about

- Hardware organization of computers
- Hardware/Software integration principles
- **Performance-related aspects of computers**



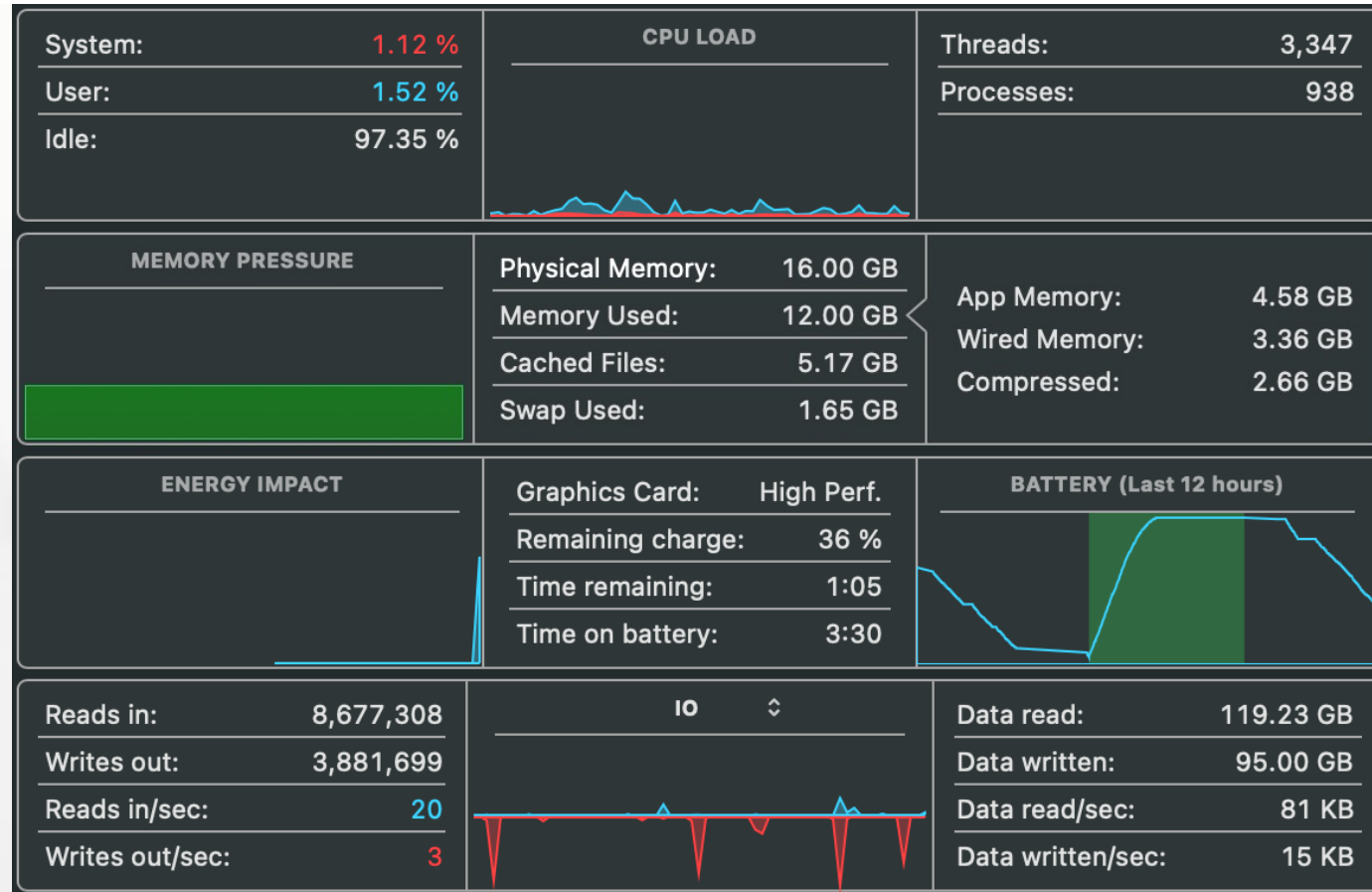
Computer Architecture is about

- Hardware organization of computers
- Hardware/Software integration principles
- **Performance-related aspects of computers**



Computer Architecture is about

- Hardware organization of computers
- Hardware/Software integration principles
- **Performance-related aspects of computers**



Computer Architecture is:

- Hardware organization of computer systems
- Hardware/Software integration principles
- Performance-related computer aspects

Why to study Computer Architecture?

Computer Architecture is:

- Hardware organization of computer systems
- Hardware/Software integration principles
- Performance-related computer aspects

Computer Architecture provides essential background to:

- **Implement runtime and memory efficient programs**

Computer Architecture is:

- Hardware organization of computer systems
- Hardware/Software integration principles
- Performance-related computer aspects

Computer Architecture provides essential background to:

- Implement runtime and memory efficient programs
- **Develop operating systems and drivers for hardware**

Computer Architecture is:

- Hardware organization of computer systems
- Hardware/Software integration principles
- Performance-related computer aspects

Computer Architecture provides essential background to:

- Implement runtime and memory efficient programs
- Develop operating systems and drivers for hardware
- **Configure hardware for specific purposes**

Computer Architecture is:

- Hardware organization of computer systems
- Hardware/Software integration principles
- Performance-related computer aspects

Computer Architecture provides essential background to:

- Implement runtime and memory efficient programs
- Develop operating systems and drivers for hardware
- Configure hardware for specific purposes
- **Design own custom processing systems**

Computer Architecture is:

- Hardware organization of computer systems
- Hardware/Software integration principles
- Performance-related computer aspects

Computer Architecture provides essential background to:

- Implement runtime and memory efficient programs
- Develop operating systems and drivers for hardware
- Configure hardware for specific purposes
- Design own custom processing systems
- **Design own hardware circuits**

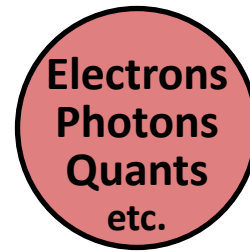
Computer Architecture is:

- Hardware organization of computer systems
- Hardware/Software integration principles
- Performance-related computer aspects

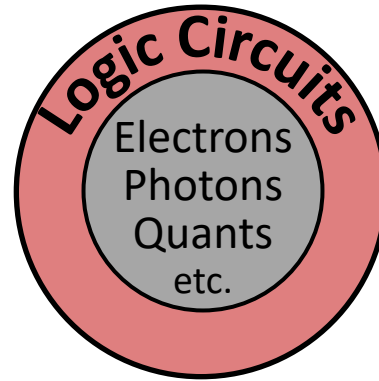
Computer Architecture provides essential background to:

- Implement runtime and memory efficient programs
- Develop operating systems and drivers for hardware
- Configure hardware for specific purposes
- Design own custom processing systems
- Design own hardware circuits

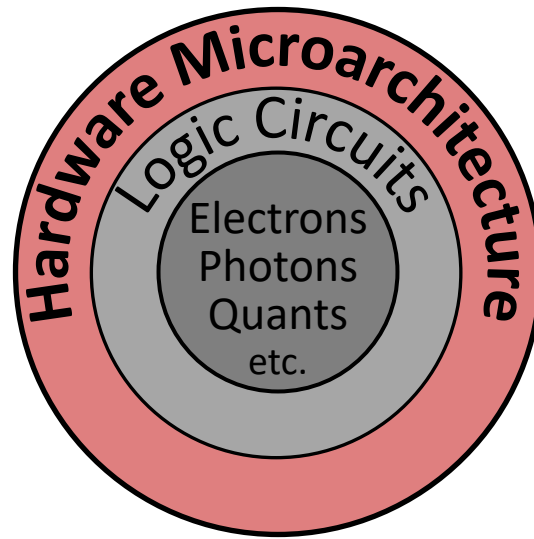
Problem Solution Stack with a Computer



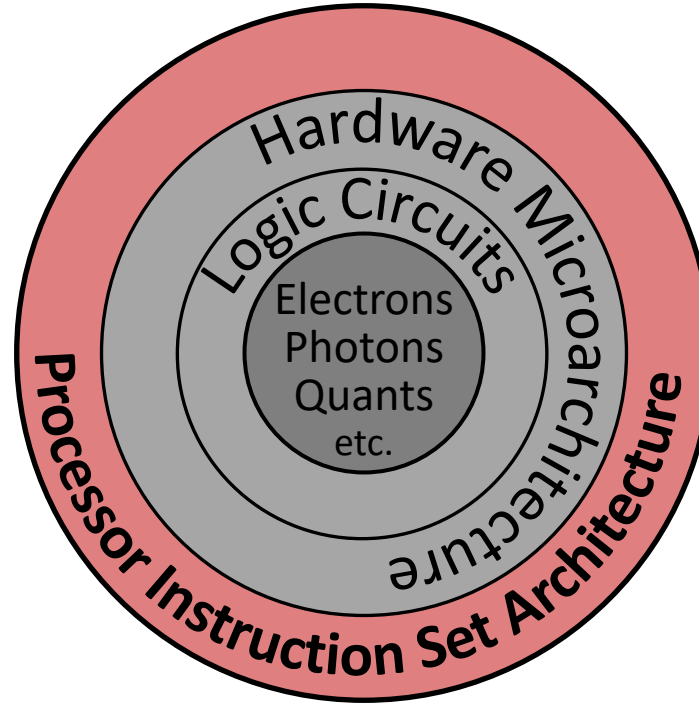
Problem Solution Stack with a Computer



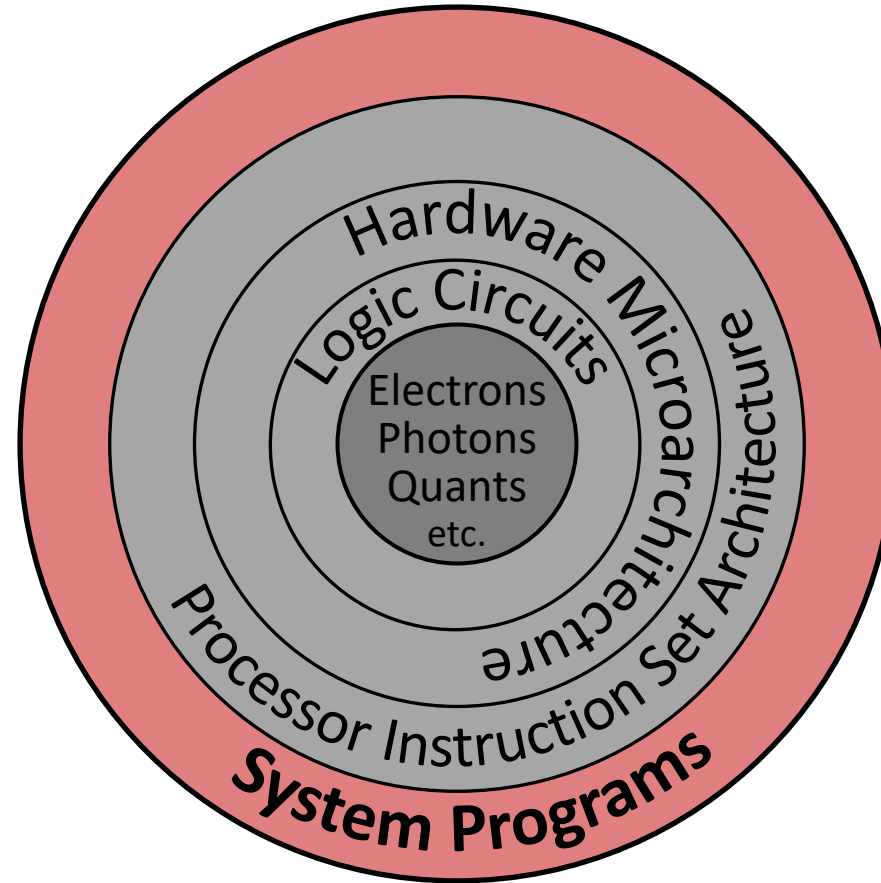
Problem Solution Stack with a Computer



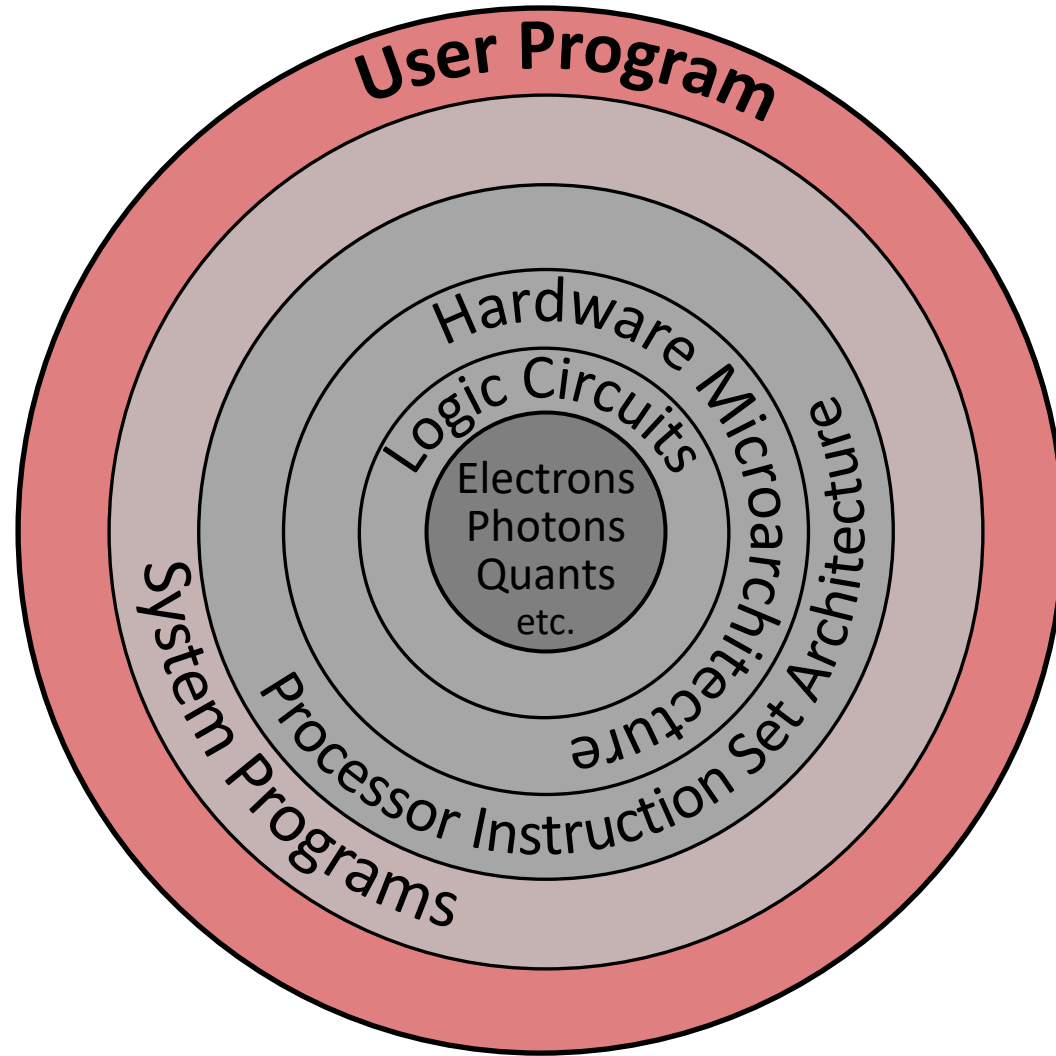
Problem Solution Stack with a Computer



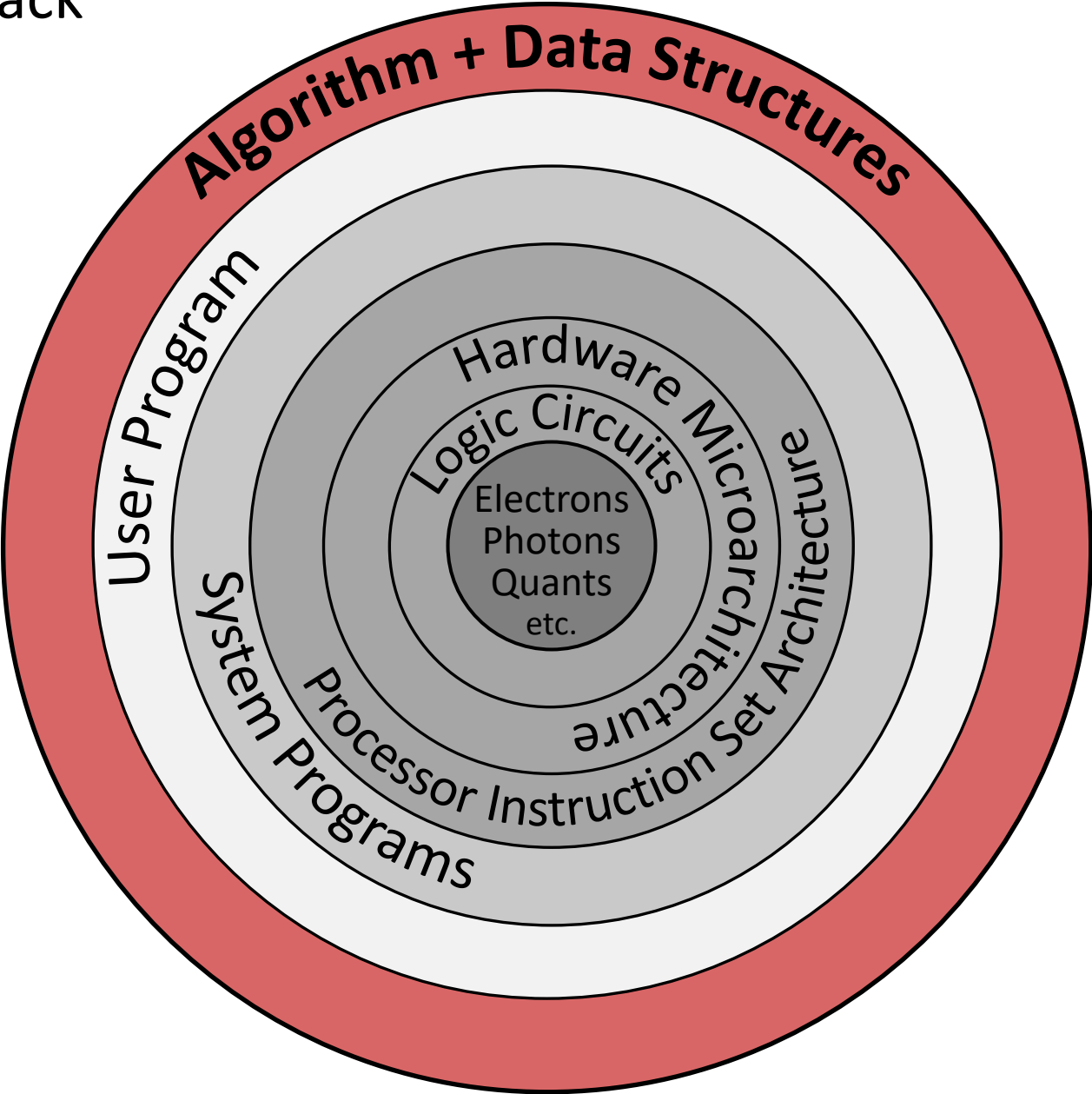
Problem Solution Stack with a Computer



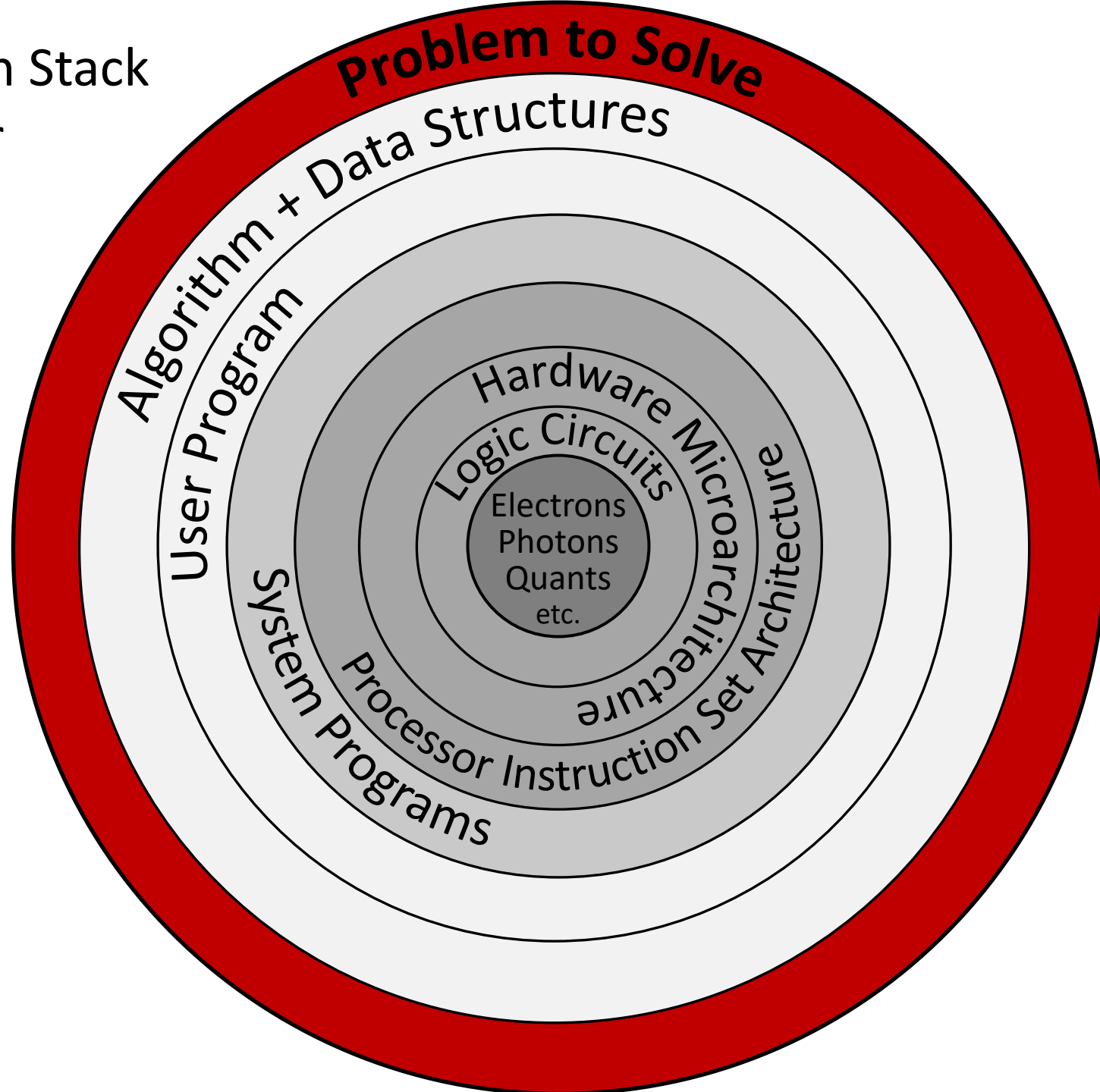
Problem Solution Stack with a Computer



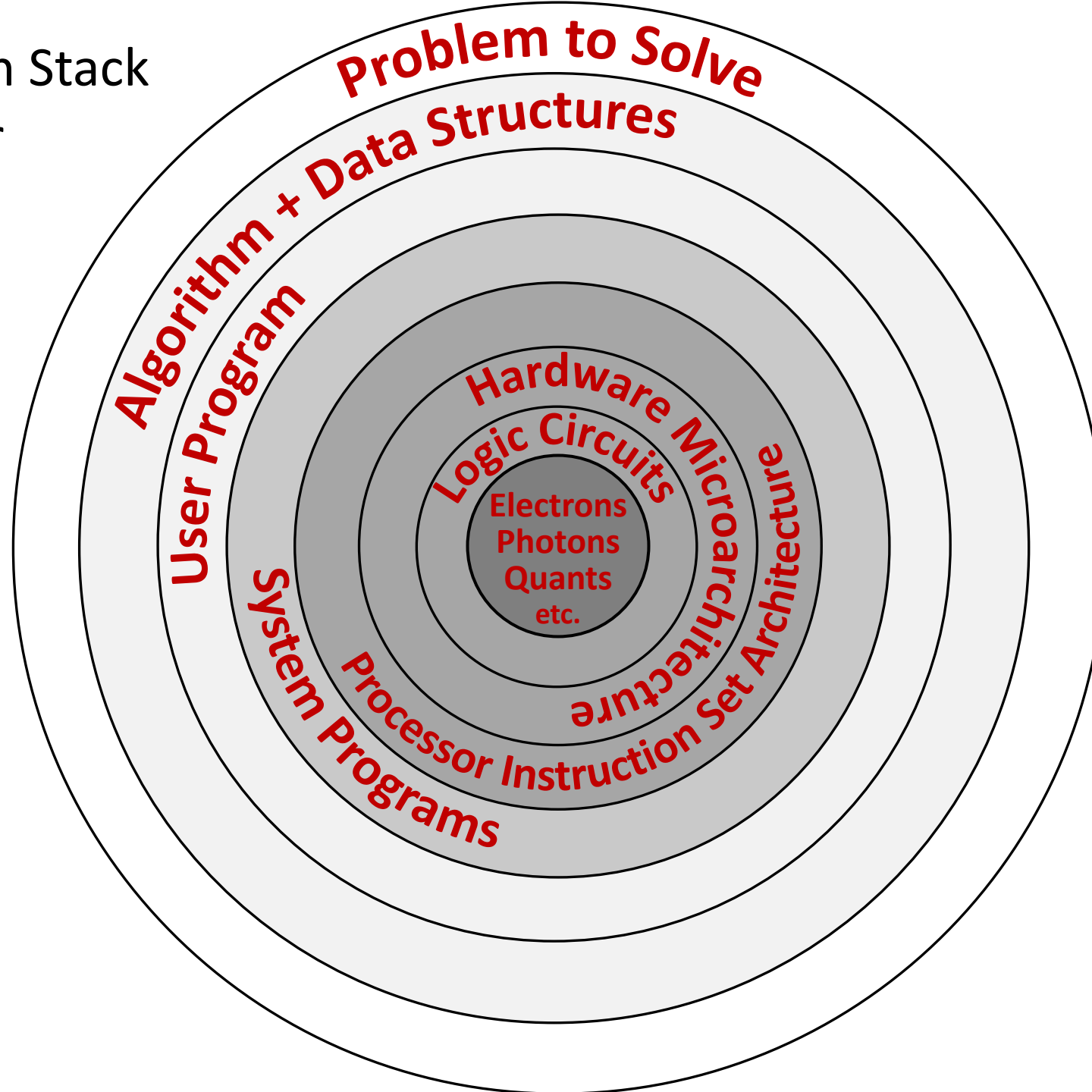
Problem Solution Stack with a Computer



Problem Solution Stack with a Computer



Problem Solution Stack with a Computer



Course Outline (can be adjusted)

Course Outline (can be adjusted)

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA

Course Outline (can be adjusted)

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project

Course Outline (can be adjusted)

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates

Course Outline (can be adjusted)

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors

Course Outline (can be adjusted)

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits

Course Outline (can be adjusted)

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits
6	Synchronous and asynchronous circuits. SRAM and DRAM	Sequential logic circuits; latches

Course Outline (can be adjusted)

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits
6	Synchronous and asynchronous circuits. SRAM and DRAM	Sequential logic circuits; latches
7	Processor: overview of key design principles	Verilog HDL

Course Outline (can be adjusted)

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits
6	Synchronous and asynchronous circuits. SRAM and DRAM	Sequential logic circuits; latches
7	Processor: overview of key design principles	Verilog HDL
8	Midterm Exam (week estimated)	

Course Outline

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits
6	Synchronous and asynchronous circuits. SRAM and DRAM	Sequential logic circuits; latches
7	Processor: overview of key design principles	Verilog HDL
8	Midterm Exam (week estimated)	
9	System functions, call stacks, related hardware aspects	Verilog design for an ALU

Course Outline

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits
6	Synchronous and asynchronous circuits. SRAM and DRAM	Sequential logic circuits; latches
7	Processor: overview of key design principles	Verilog HDL
8	Midterm Exam (week estimated)	
9	System functions, call stacks, related hardware aspects	Verilog design for an ALU
10	Processor pipelining and hazards	Further practice with Verilog

Course Outline

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits
6	Synchronous and asynchronous circuits. SRAM and DRAM	Sequential logic circuits; latches
7	Processor: overview of key design principles	Verilog HDL
8	Midterm Exam (week estimated)	
9	System functions, call stacks, related hardware aspects	Verilog design for an ALU
10	Processor pipelining and hazards	Further practice with Verilog
11	Introduction of RISC-V instruction set architecture	RISC-V assembly and RARS simulator

Course Outline

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits
6	Synchronous and asynchronous circuits. SRAM and DRAM	Sequential logic circuits; latches
7	Processor: overview of key design principles	Verilog HDL
8	Midterm Exam (week estimated)	
9	System functions, call stacks, related hardware aspects	Verilog design for an ALU
10	Processor pipelining and hazards	Further practice with Verilog
11	Introduction of RISC-V instruction set architecture	RISC-V assembly and RARS simulator
12	Heterogeneous computer architectures. GPU, NPU, other	RISC-V conditional instructions

Course Outline

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits
6	Synchronous and asynchronous circuits. SRAM and DRAM	Sequential logic circuits; latches
7	Processor: overview of key design principles	Verilog HDL
8	Midterm Exam (week estimated)	
9	System functions, call stacks, related hardware aspects	Verilog design for an ALU
10	Processor pipelining and hazards	Further practice with Verilog
11	Introduction of RISC-V instruction set architecture	RISC-V assembly and RARS simulator
12	Heterogeneous computer architectures. GPU, NPU, other	RISC-V conditional instructions
13	General principles of caching. A simple cache design	RISC-V floating point instructions

Course Outline

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits
6	Synchronous and asynchronous circuits. SRAM and DRAM	Sequential logic circuits; latches
7	Processor: overview of key design principles	Verilog HDL
8	Midterm Exam (week estimated)	
9	System functions, call stacks, related hardware aspects	Verilog design for an ALU
10	Processor pipelining and hazards	Further practice with Verilog
11	Introduction of RISC-V instruction set architecture	RISC-V assembly and RARS simulator
12	Heterogeneous computer architectures. GPU, NPU, other	RISC-V conditional instructions
13	General principles of caching. A simple cache design	RISC-V floating point instructions
14	Recap of selected topics	Delivery of missing labs (subject to penalisation)

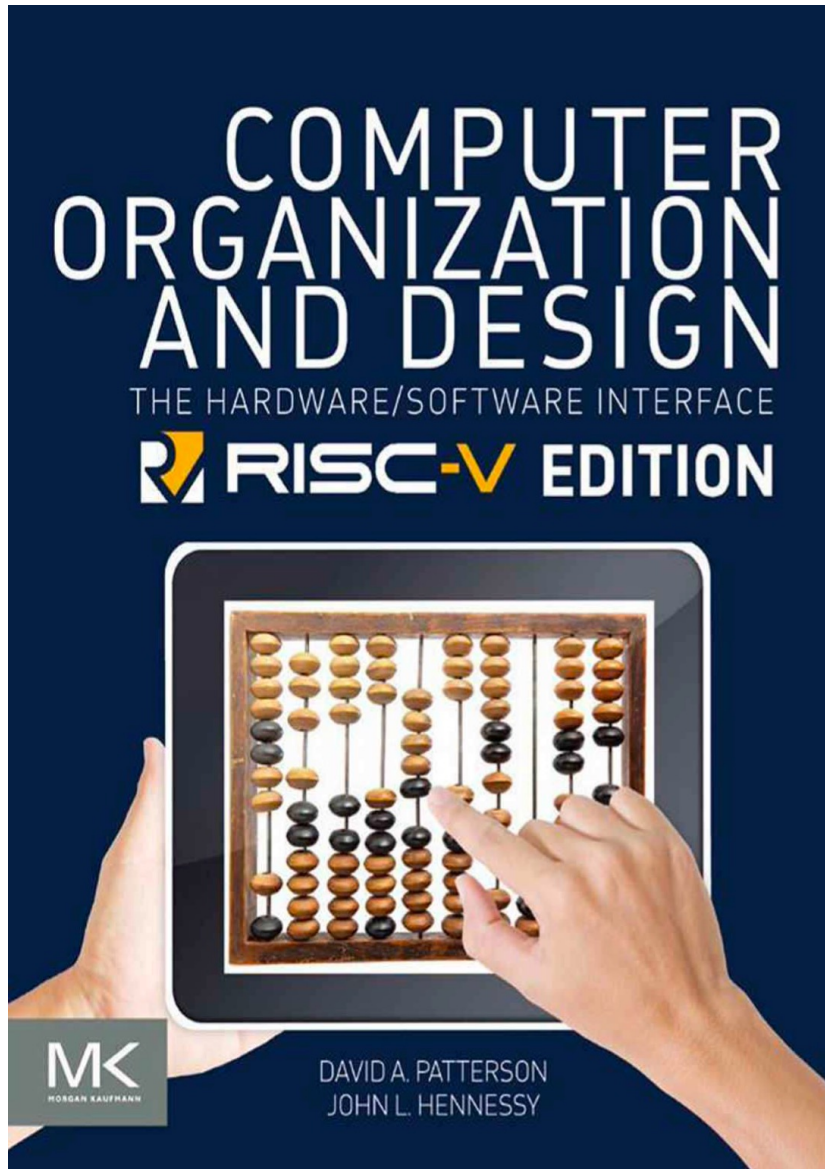
Course Outline

W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits
6	Synchronous and asynchronous circuits. SRAM and DRAM	Sequential logic circuits; latches
7	Processor: overview of key design principles	Verilog HDL
8	Midterm Exam (week estimated)	
9	System functions, call stacks, related hardware aspects	Verilog design for an ALU
10	Processor pipelining and hazards	Further practice with Verilog
11	Introduction of RISC-V instruction set architecture	RISC-V assembly and RARS simulator
12	Heterogeneous computer architectures. GPU, NPU, other	RISC-V conditional instructions
13	General principles of caching. A simple cache design	RISC-V floating point instructions
14	Recap of selected topics	Delivery of missing labs (subject to penalisation)
15,16	Pre-final (optional) and final exams	

Course Outline (can be adjusted)

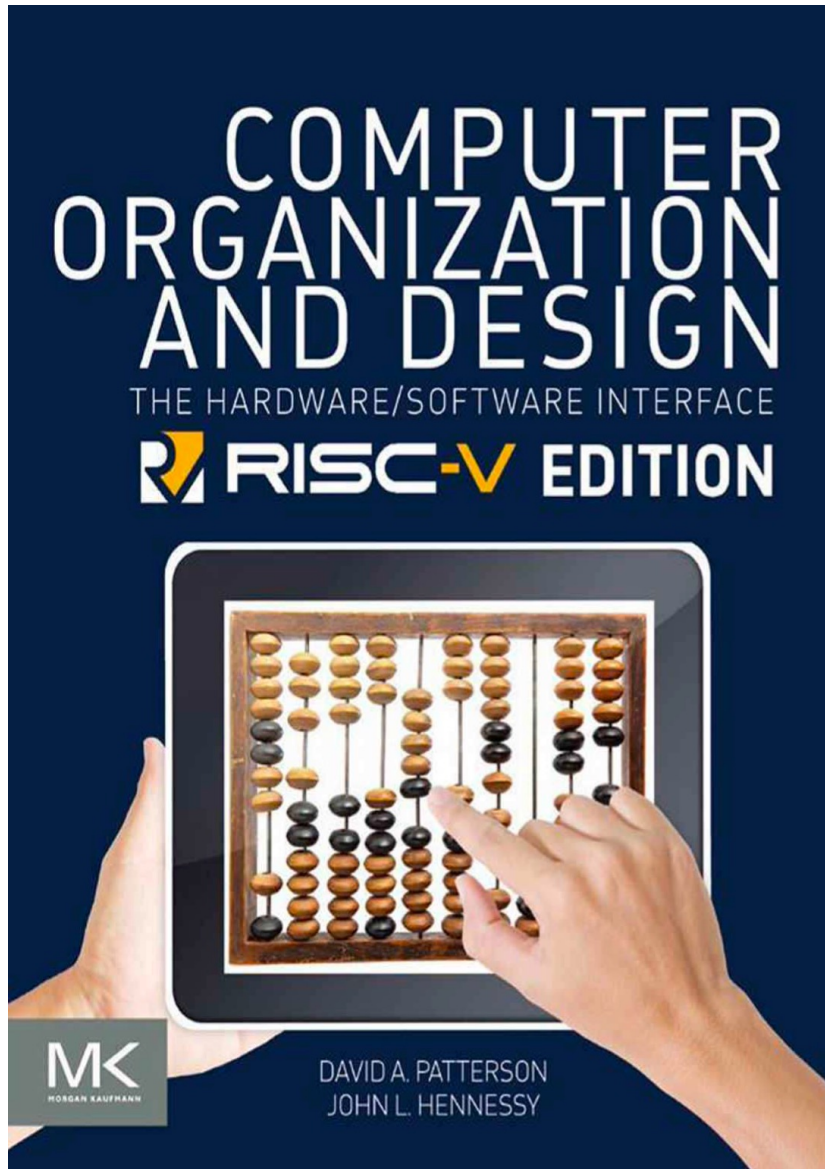
W	Lecture	Lab
1	Introduction. A brief overview of a computer layout	Introduction to FPGA
2	9 fundamental ideas of Computer Architecture	First FPGA project
3	Performance metrics of a computer. Logic gates	Performance testbenches. Universal logic gates
4	Combinational circuits, hardware building blocks	Adders and subtractors
5	The design of a simple ALU. Sequential logic circuits	Combinational logic circuits
6	Synchronous and asynchronous circuits. SRAM and DRAM	Sequential logic circuits; latches
7	Processor: overview of key design principles	Verilog HDL
8	Midterm Exam (week estimated)	
9	System functions, call stacks, related hardware aspects	Verilog design for an ALU
10	Processor pipelining and hazards	Further practice with Verilog
11	Introduction of RISC-V instruction set architecture	RISC-V assembly and RARS simulator
12	Heterogeneous computer architectures. GPU, NPU, other	RISC-V conditional instructions
13	General principles of caching. A simple cache design	RISC-V floating point instructions
14	Recap of selected topics	Delivery of missing labs (subject to penalisation)
15,16	Pre-final (optional) and final exams	

Our Major Course Book



(1st ed. 1993)

Our Major Course Book

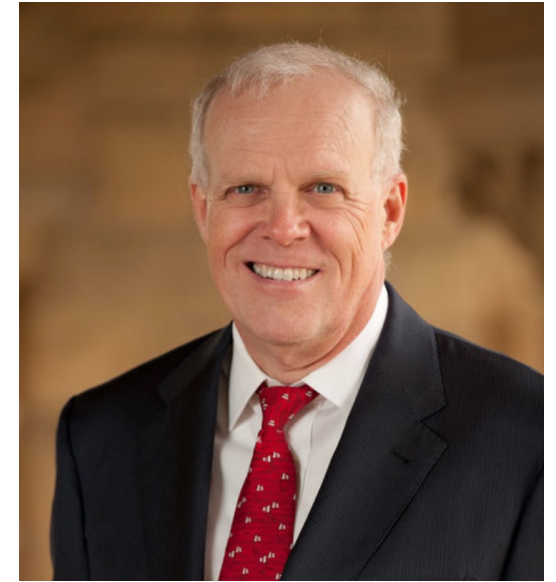


(1st ed. 1993)



David A. Patterson

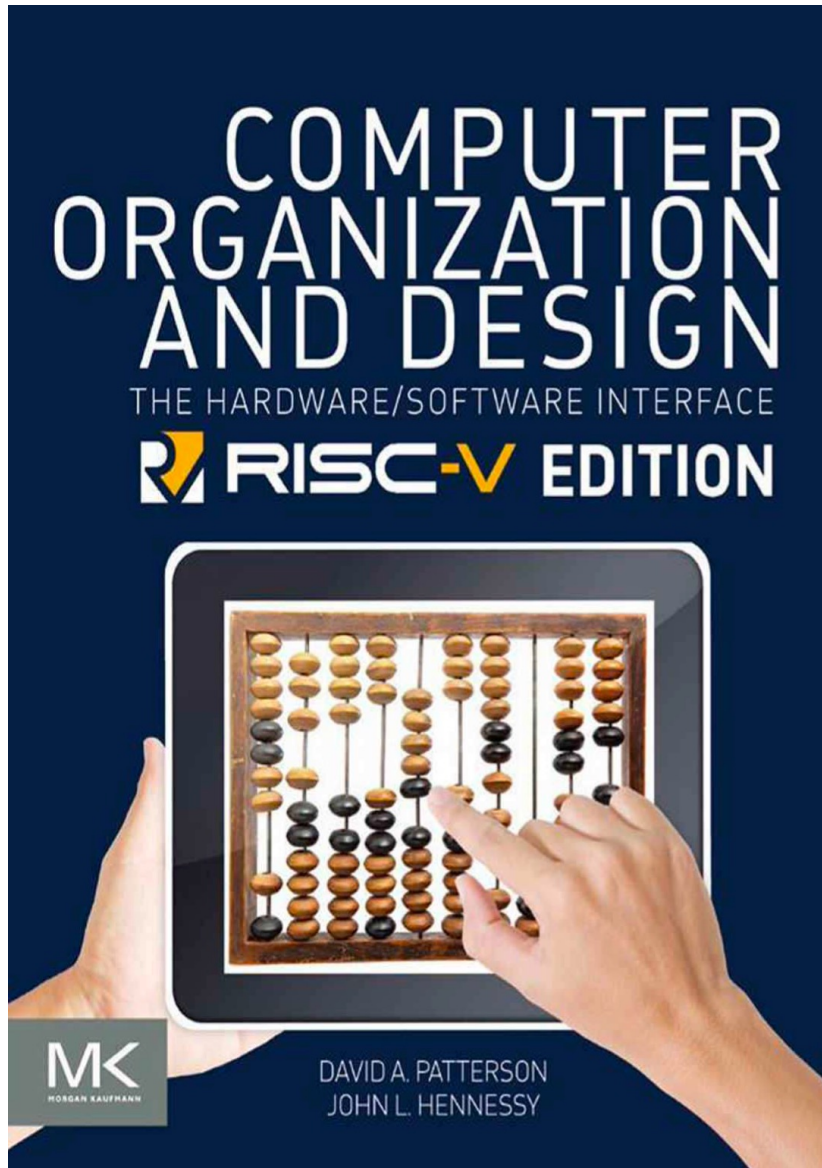
University of California, Berkley
Distinguished Software Engineer,
Google



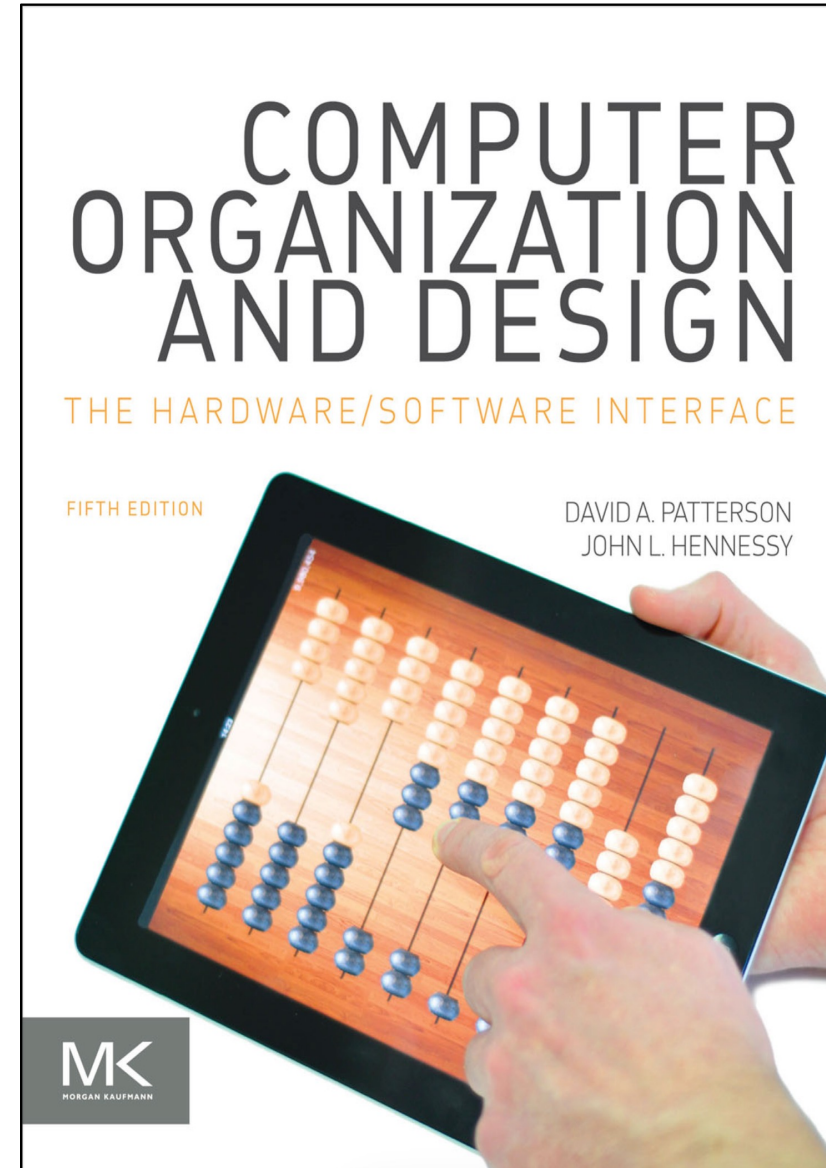
John L. Hennessy

Stanford University
Chairman of Alphabet Inc.

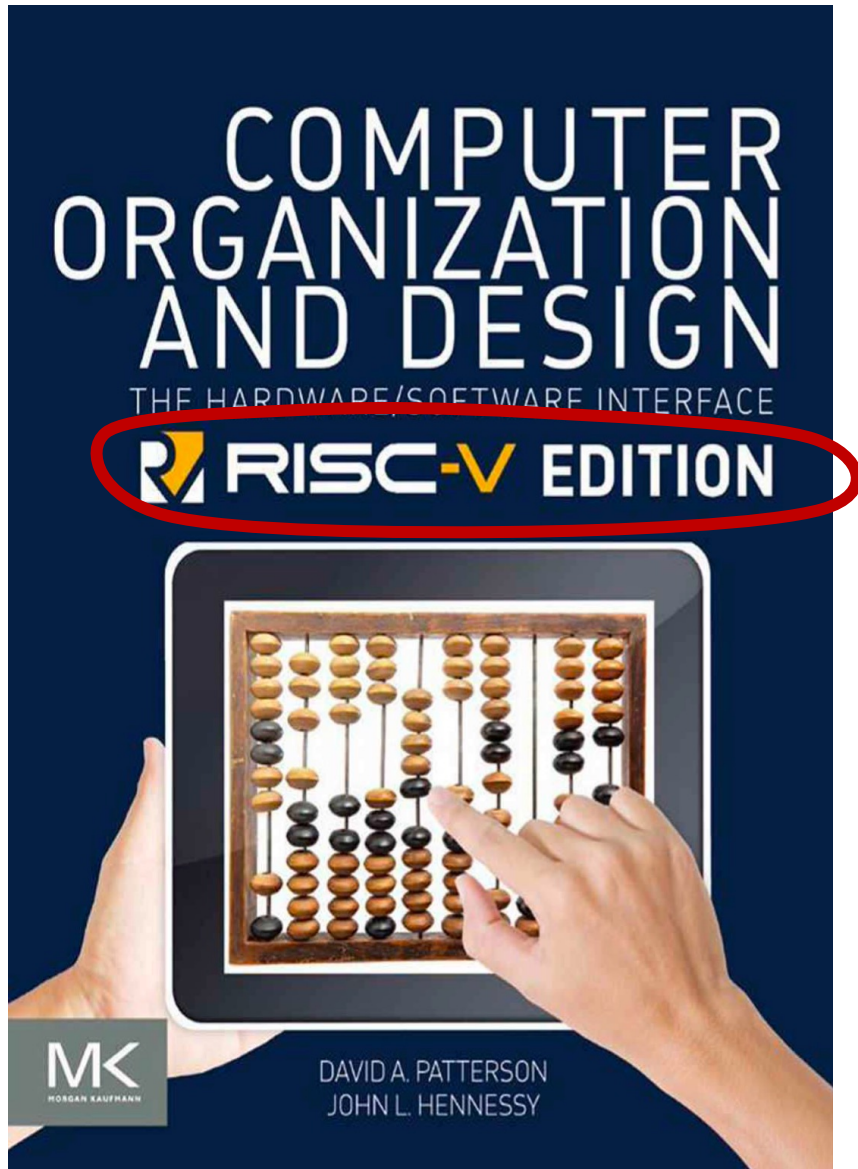
Our Major Course Book



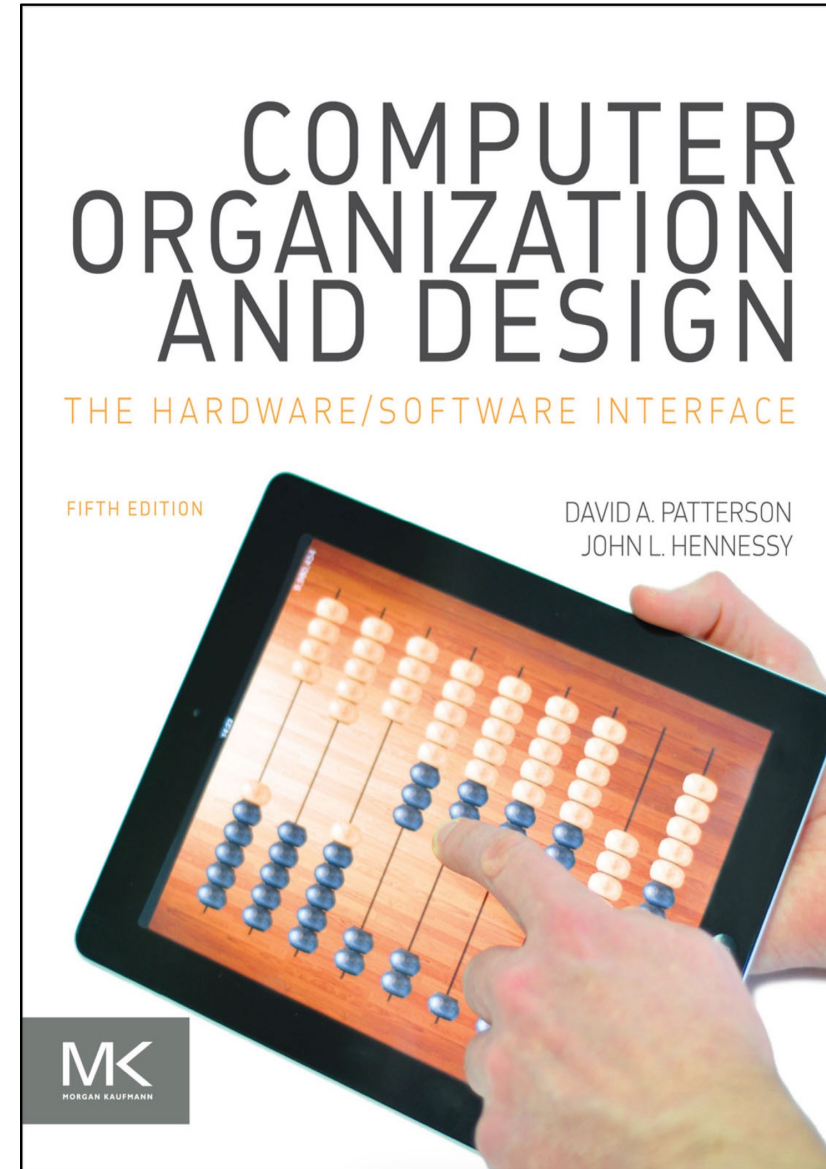
Older Book Version



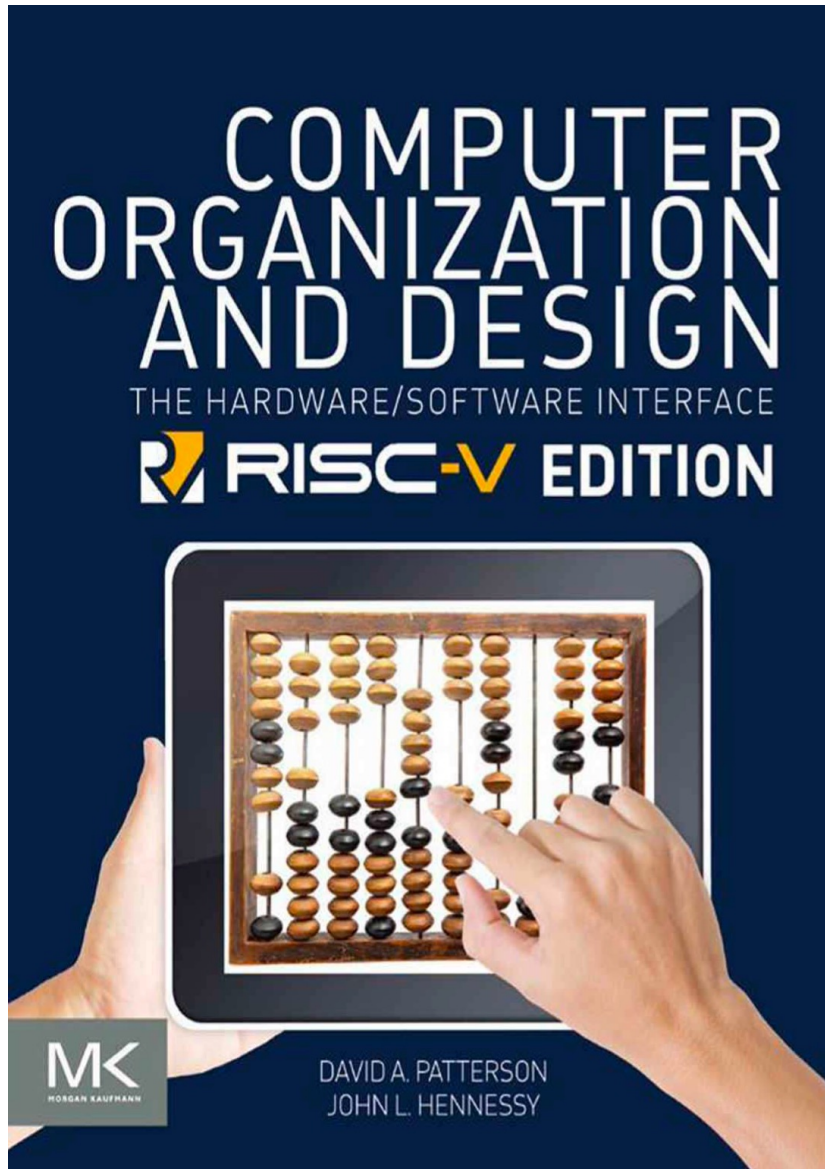
Our Major Course Book



Older Book Version

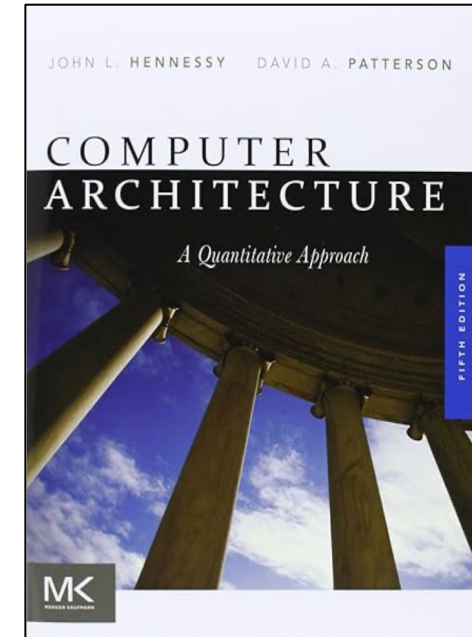
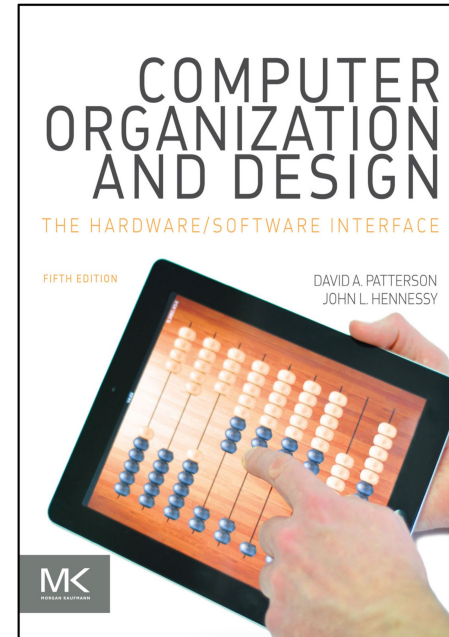


Our Major Course Book



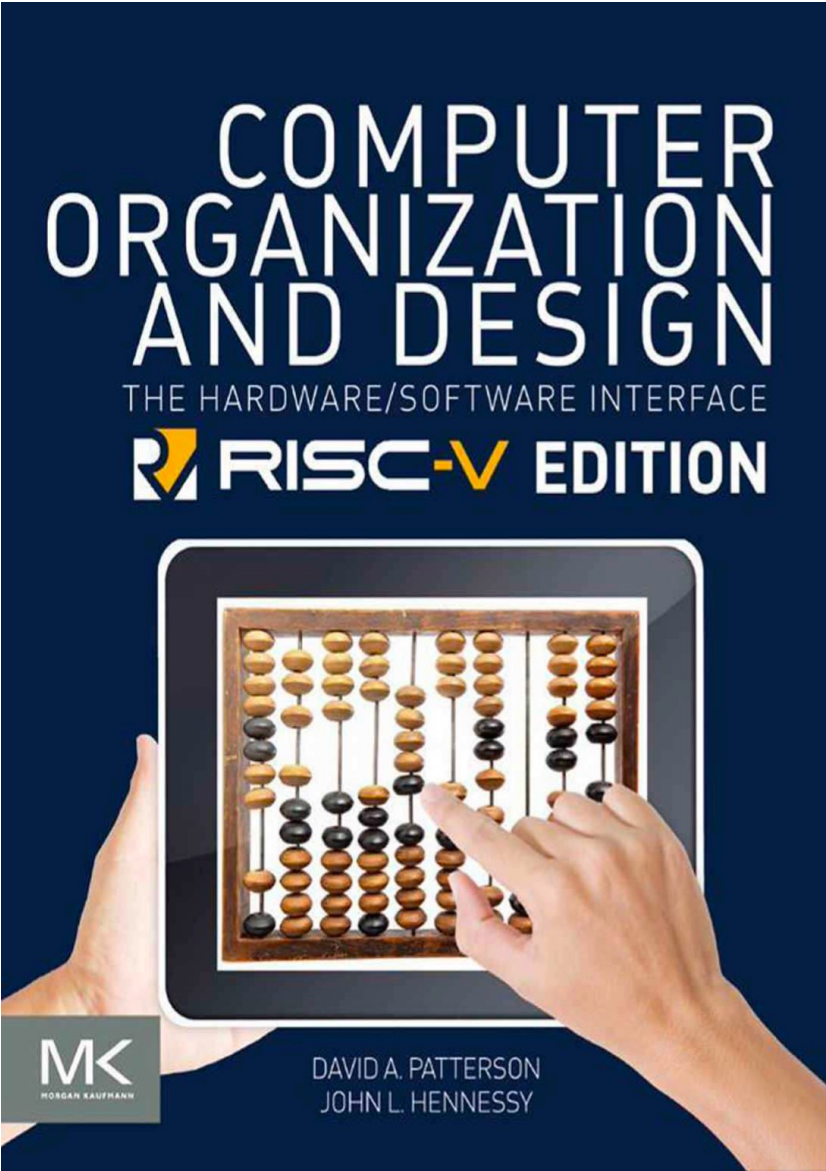
(1st ed. 1993)

Other Well-Known Books



(1st ed. 1990)

Our Major Course Book



(1st ed. 1993)

Other Well-Known Books



(Tanenbaum; 1st ed. 1976)

COMPUTER ORGANIZATION AND DESIGN

THE HARDWARE/SOFTWARE INTERFACE

RISC-V EDITION



MK
MORGAN KAUFMANN

DAVID A. PATTERSON
JOHN L. HENNESSY

**COMPUTER
ORGANIZATION
AND DESIGN**

THE HARDWARE/SOFTWARE INTERFACE

FIFTH EDITION

DAVID A. PATTERSON
JOHN L. HENNESSY

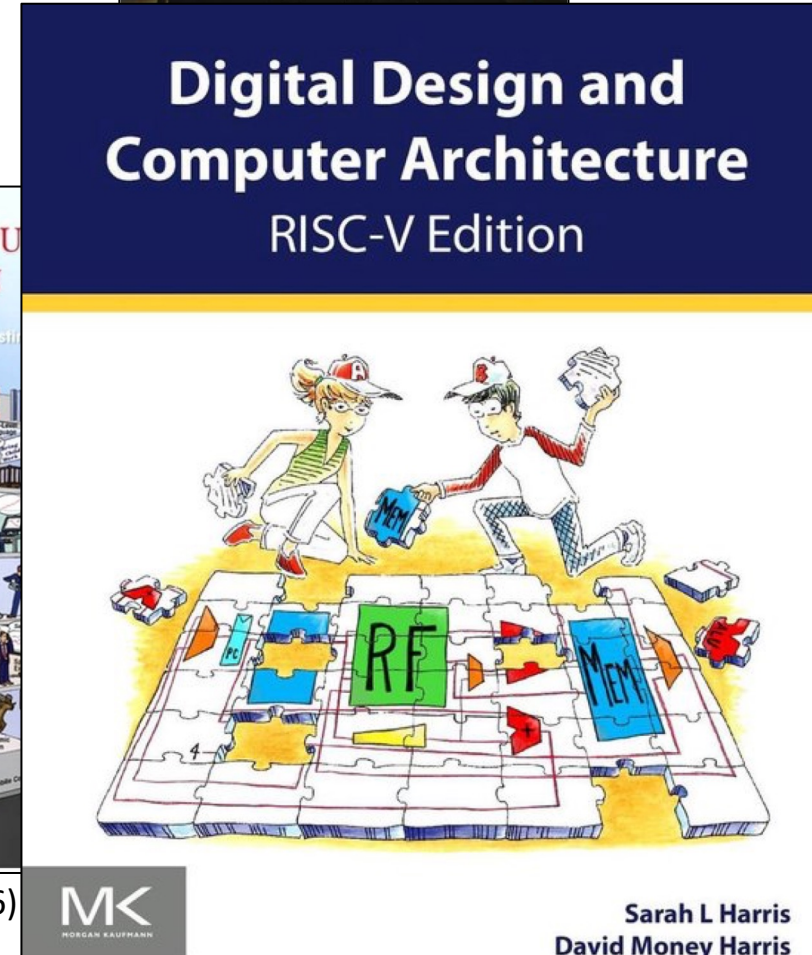
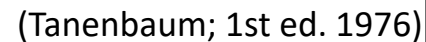
**STRUCTURED
ORGANIZATION**

SIXTH EDITION

Andrew S. Tanenbaum

COMPUTER ARCHITECTURE

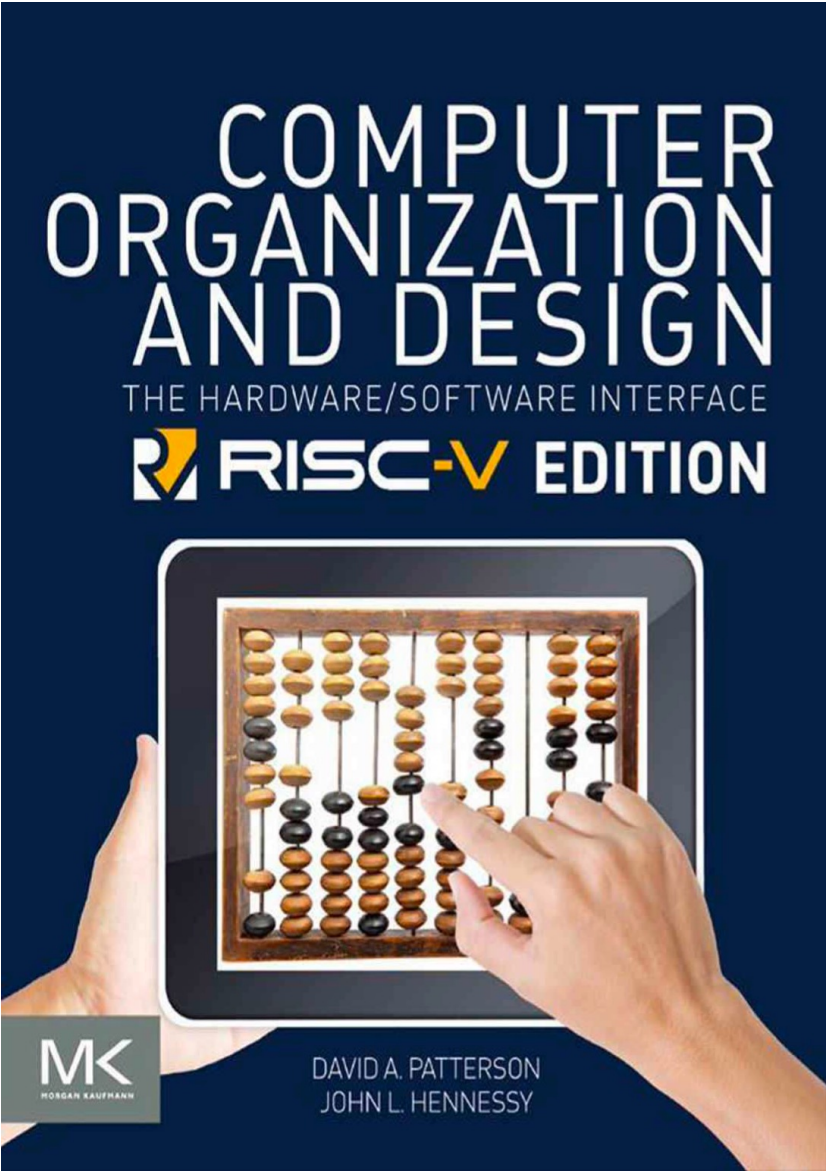
MK
MORGAN KAUFMANN



MORGAN KAUFMANN

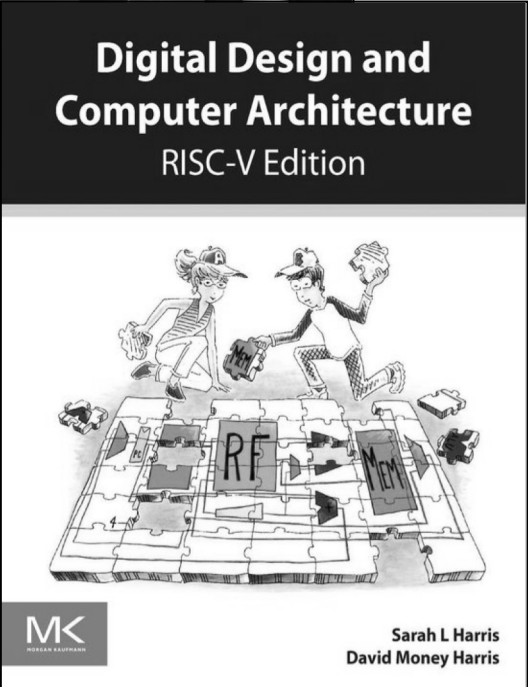
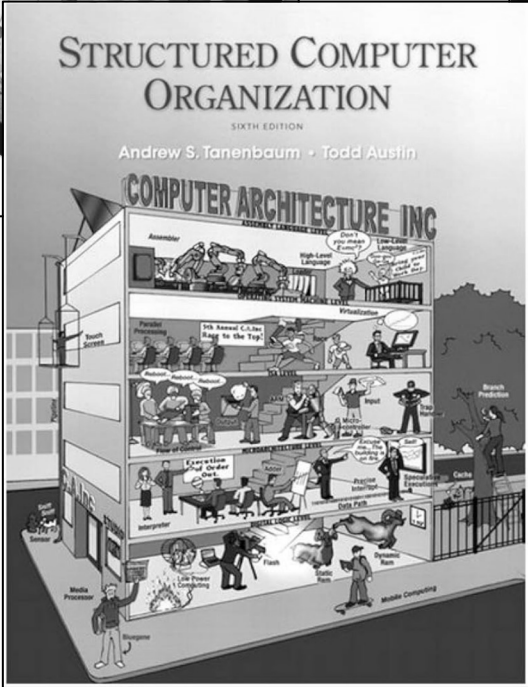
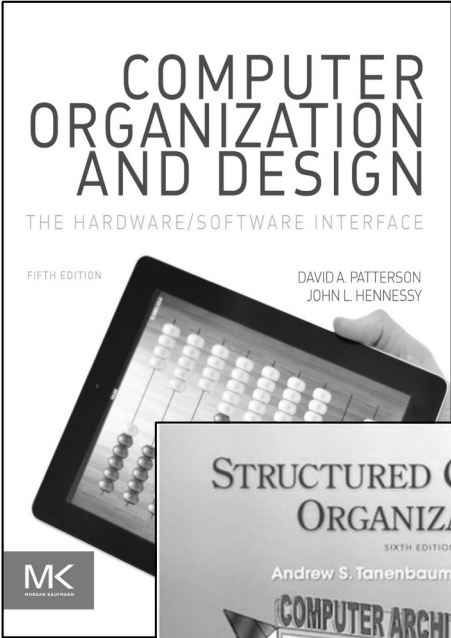
Sarah L Harris
David Money Harris

Our Major Course Book



(1st ed. 1993)

Other Well-Known Books



(1st ed.)

Course Grading Policy

Course Component	Weight to Final Grade
Regular quizzes (considering only 4 best quizzes)	5%
Lab assignments <i>Note on late delivery (assuming no reasonable justification is provided)</i> <i>50% penalty, if delivered later than 1 week exclusive;</i> <i>Not graded after 2 weeks exclusive</i>	15%
Midterm Exam	20%
Final exam (written format, no oral part)	60%

Letter Grades:

A $\geq$ 90, B $\geq$ 70, C $\geq$ 60, D $<$ 60

Acknowledgment

Over years, the following people have contributed to this lecture material:

- Alexander Tormasov
- Giancarlo Succi
- Muhammad Fahim
- Yuri Panchul
- Artem Burmyakov