Introduction to Computer Organization

Introduction

October 2023

Overview

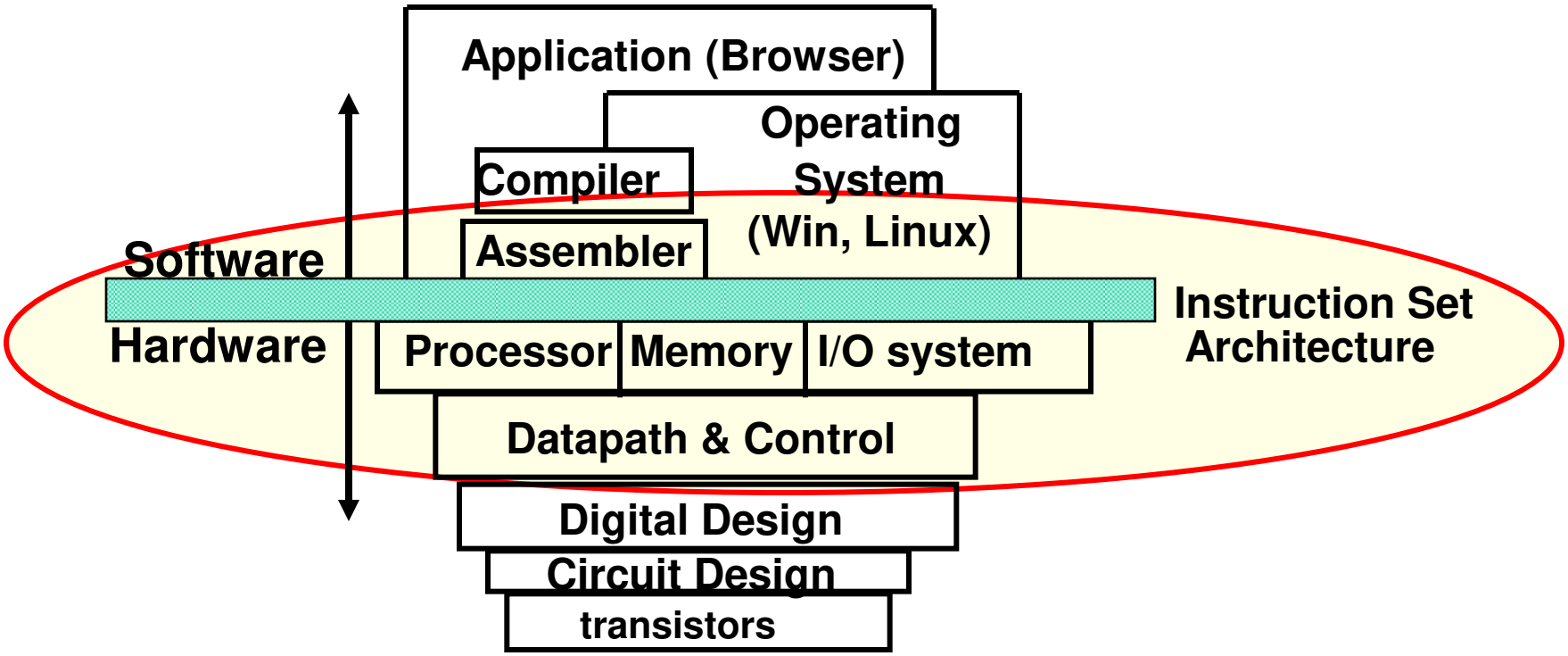
- Introduction to Computer Organization
- Course overview
- Housekeeping details

- Computer abstraction
- Anatomy of a computer
- Conclusion

Introduction

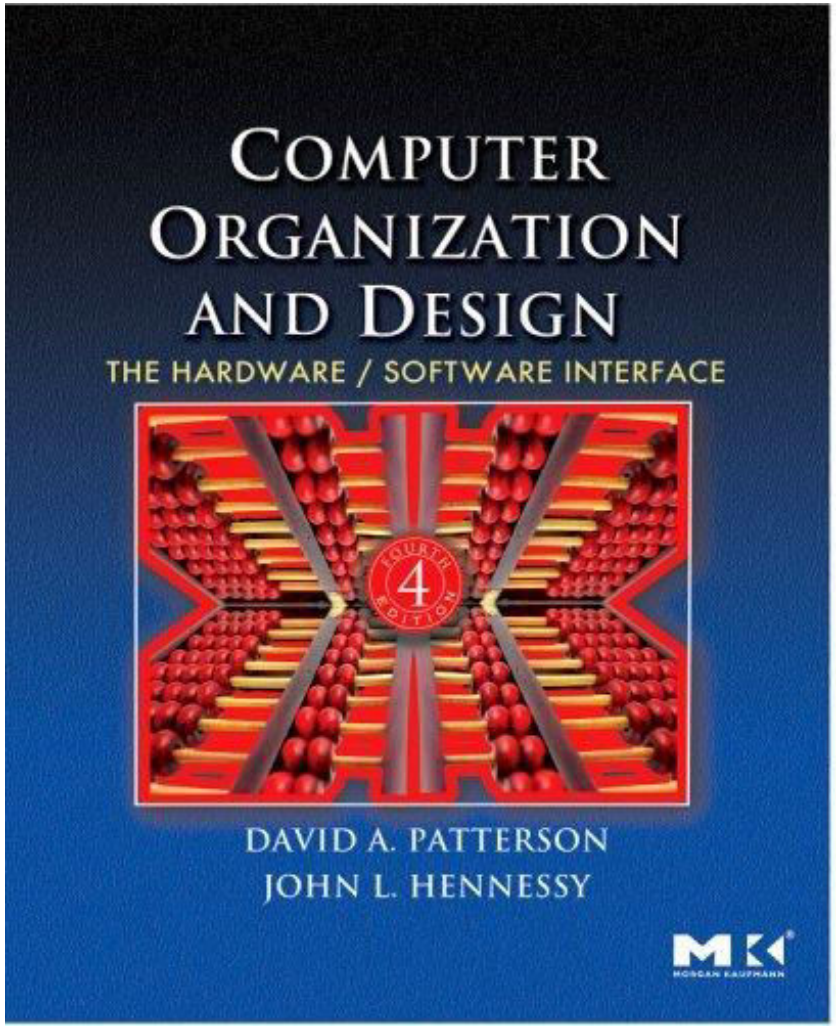
- **Rapidly changing field:**
  - vacuum tube -> transistor -> IC -> VLSI
  - doubling every 1.5 years (Moore's law):
    - *Memory capacity*
    - *Processor speed (Due to advances in technology and organization)*
- **Things you'll be learning:**
  - how computers work, a basic foundation
  - how to analyze their performance (or how not to!)
  - issues affecting modern processors (caches, pipelines)
- **Why learn this stuff?**
  - you want to call yourself a "computer scientist"
  - you want to build software people use (need performance)
  - you need to make a purchasing decision or offer advice

Computing System



- Coordination of many *levels of abstraction*

Computer Organization



Textbook: "P&H"

Computer Organization and Design

Patterson and Hennessy

Fourth Edition

Morgan Kaufmann

ISBN: 978012374493

Course Overview

- Performance issues (Ch 1 – P&H **4th Edition**)
- A specific instruction set architecture (Ch 2)
- Arithmetic and how to build an ALU (Ch 3)
- Constructing a processor to execute our instructions (Ch 4)
- Pipelining to improve performance (Ch 5)
- Caches, main, and virtual memory, I/O (Ch 6,7)
- Multiprocessor Architectures (Ch 8)
- Future Computing Technologies (instructor)

Computer Organization **Big Ideas**

- **5 Classic components of a Computer**
- **Data can be anything** (integers, floating point, characters): a program determines what it is
- **Stored program** concept: instructions just data
- **Principle of Locality**, exploited via a memory hierarchy (cache and virtual memory)
- **Greater performance** by exploiting parallelism
- **Principle of abstraction**, used to build complex systems as layers
- **Compilation v. interpretation** thru system layers
- Principles/Pitfalls of **Performance Measurement**

Course Administration

- Instructor:
  - Pr. Cherif TOLBA (cherif.tolba@**univ-annaba.dz**)

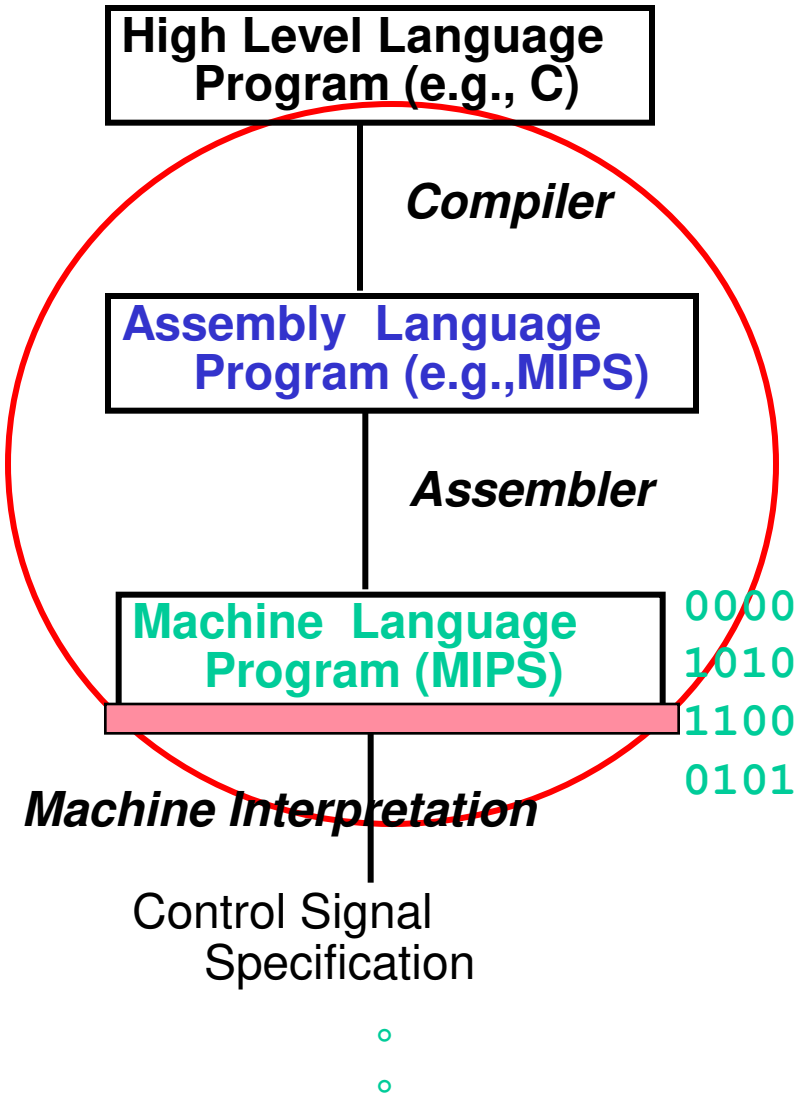
- Text: ***Computer Organization and Design: The Hardware Software Interface***, **Fourth Edition**, Patterson and Hennessy

Course Evaluation

- Grade breakdown
  - Final Exam 60%
  - Tutorial Works 20%
  - Practical Works 20%

TOTAL 100%

Levels of Representation

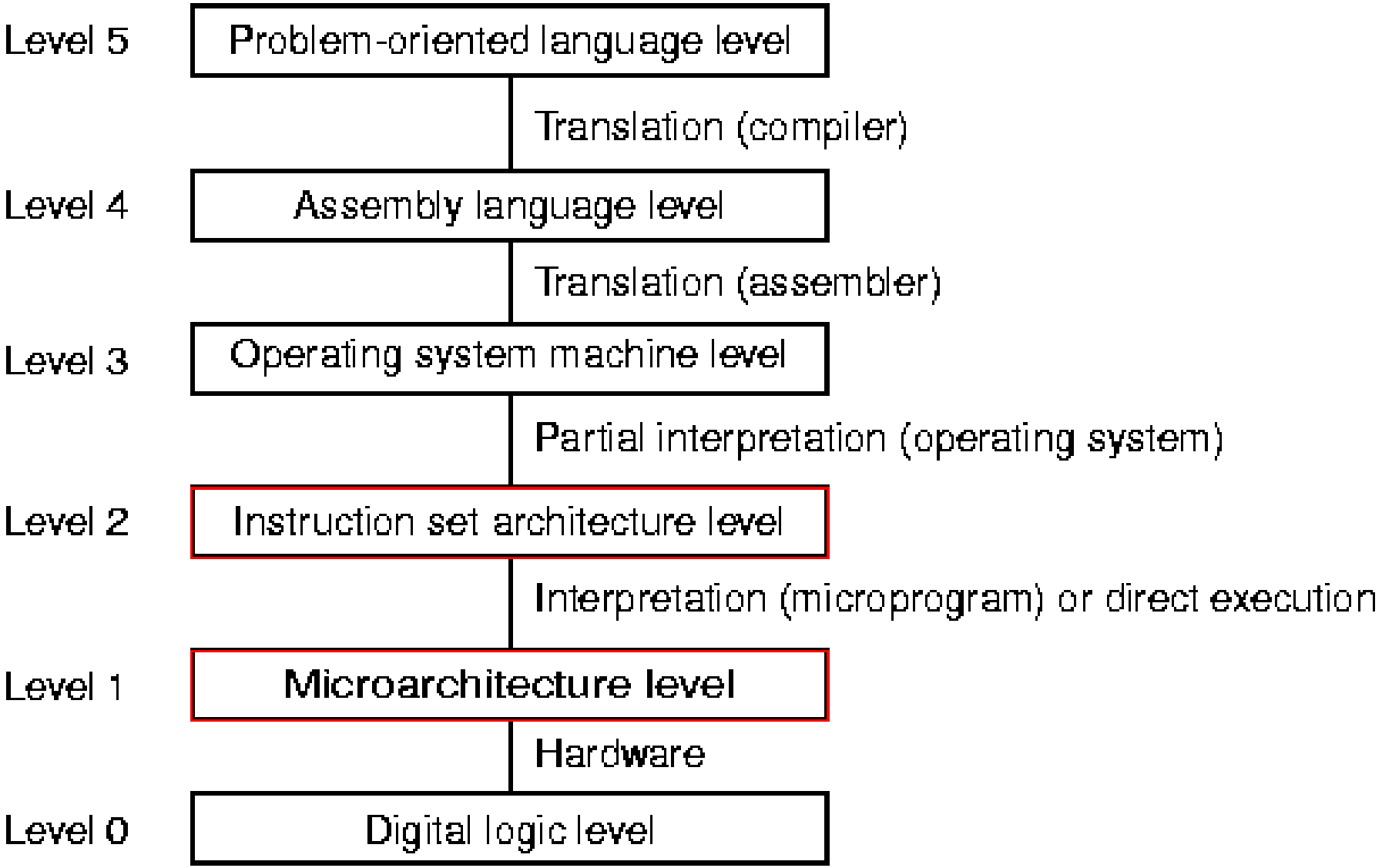



```
temp = v[k];
v[k] = v[k+1];
v[k+1] = temp;
```

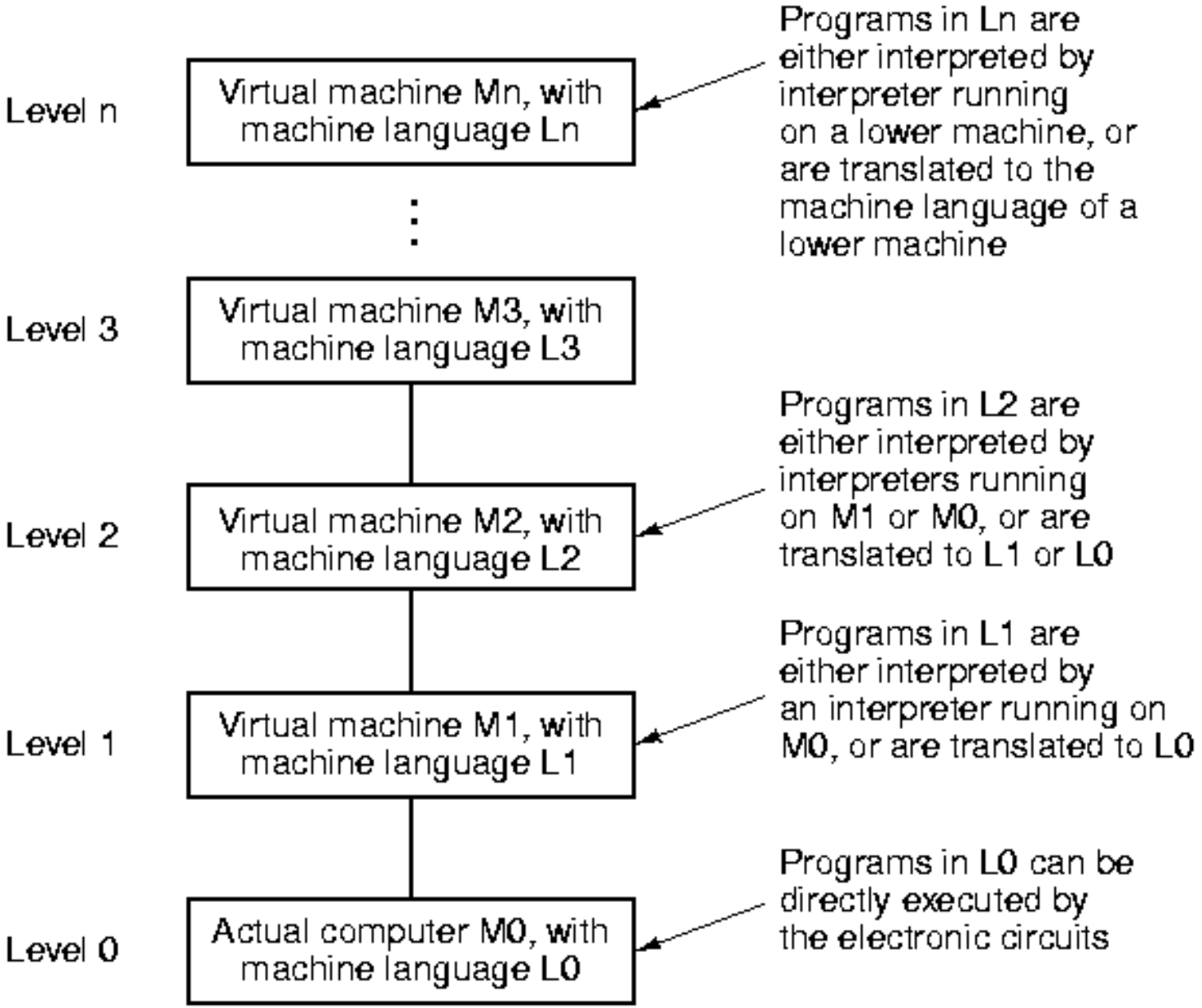
```
lw $t0,   0($2)
lw $t1,   4($2)
sw$t1,    0($2)
sw$t0,    4($2)
```

```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

A Six-Level Computer



Big Idea: Multilevel Machine



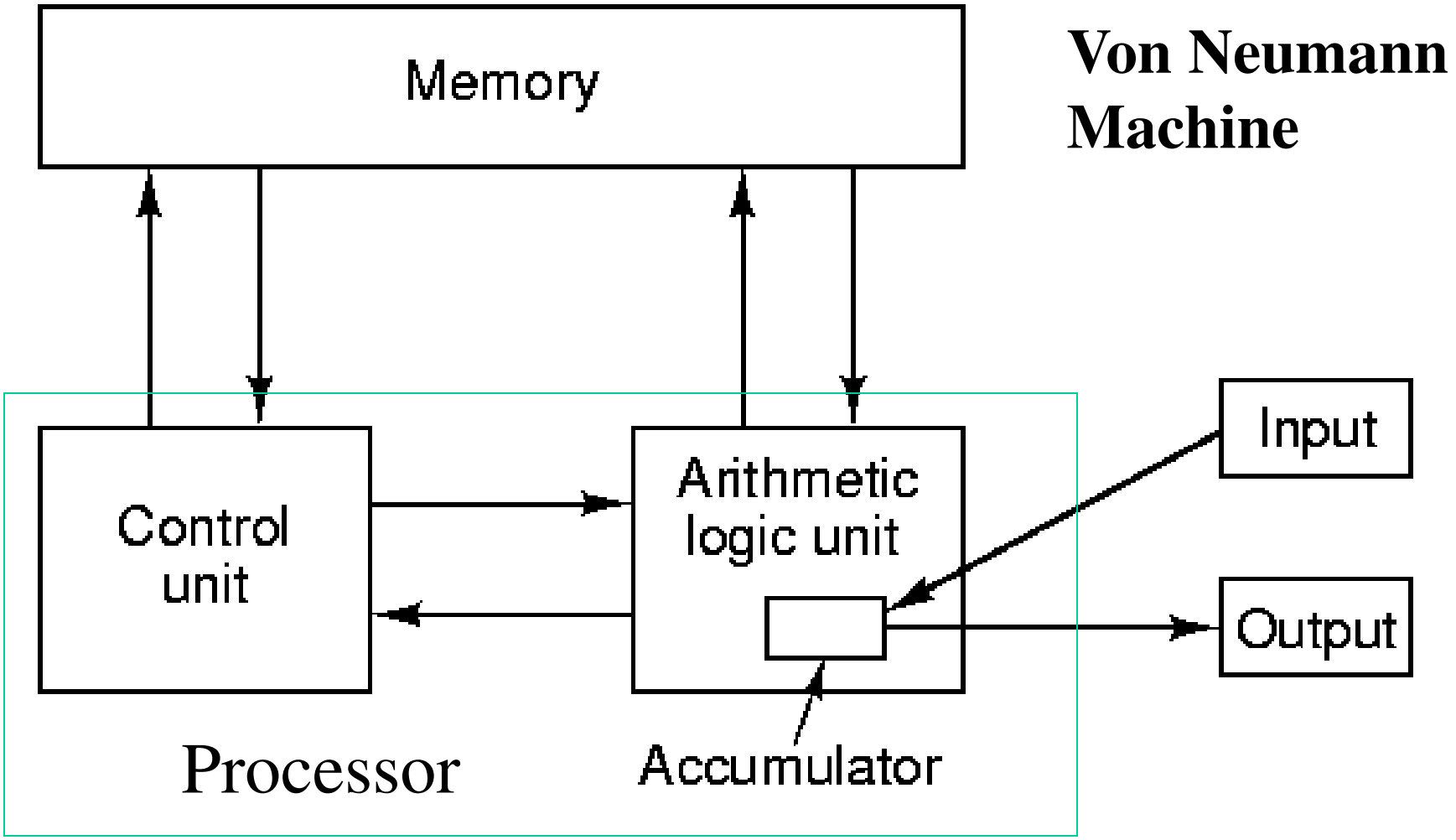
Evolution of Multilevel Machines

1. Bare hardware
2. Microprogramming
3. Operating system
4. Compilers
5. **Hardware / software interface**
   - Simple ISA
   - CISC
   - RISC

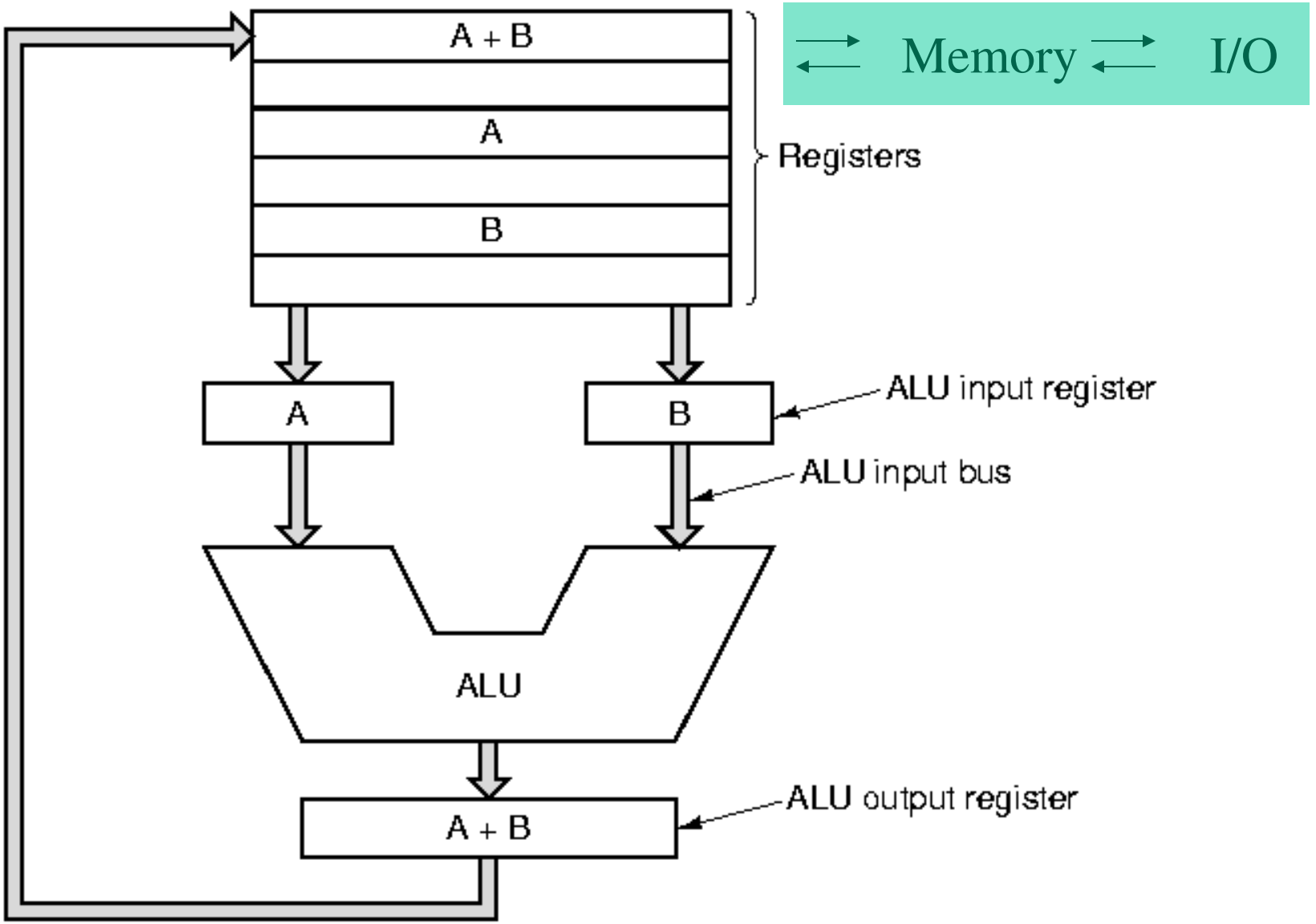
Design Principles

- CISC vs. RISC
- Instructions directly executed by hardware
- Maximize instruction issue rate (ILP)
- Simple instructions (easy to decode)
- Access to memory only via load/store
- Plenty of registers
- Pipelining

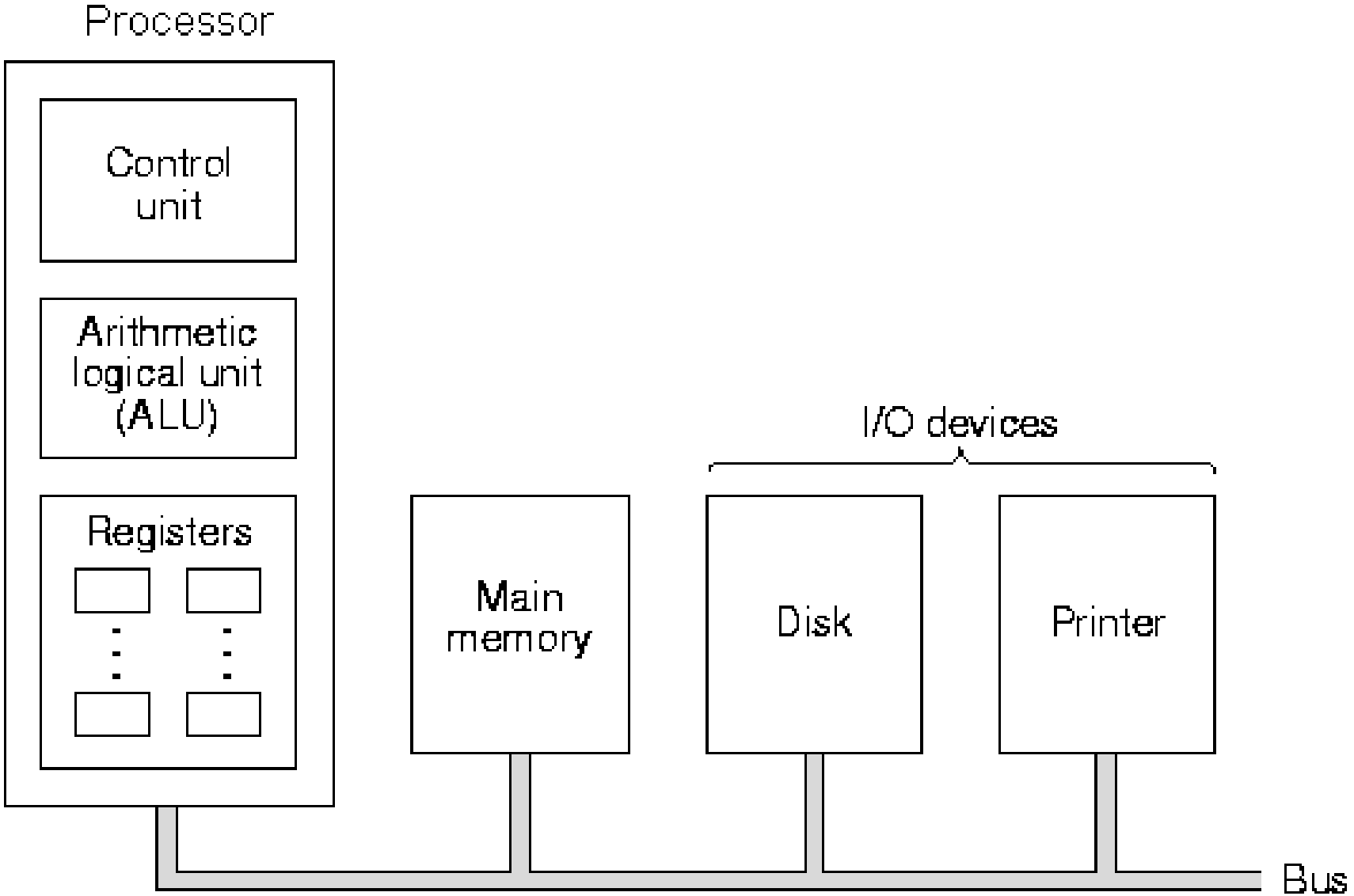
Computer Organization



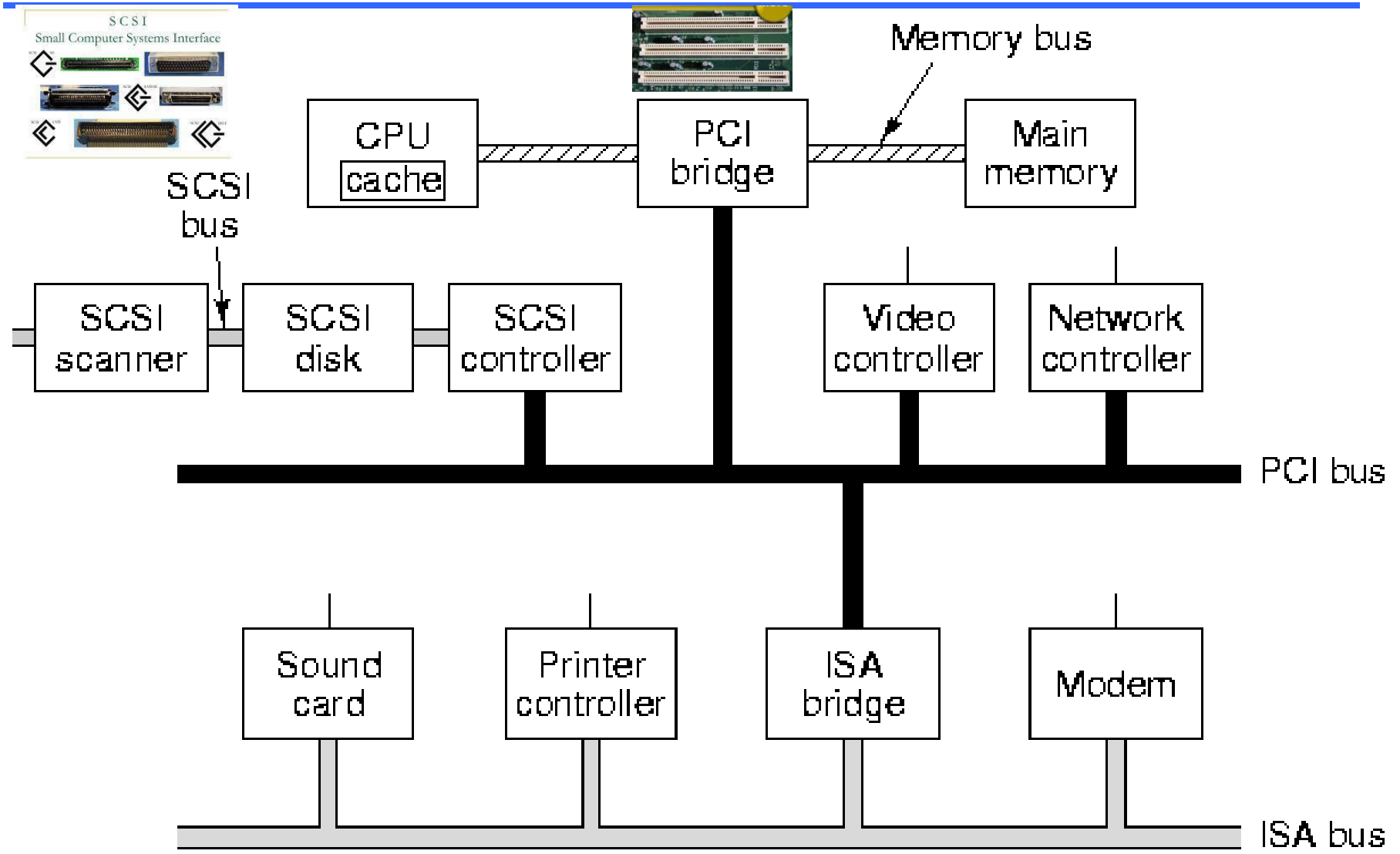
Datapath


A + B
A
B
Registers
Memory
I/O
A
B
ALU input register
ALU input bus
ALU
A + B
ALU output register

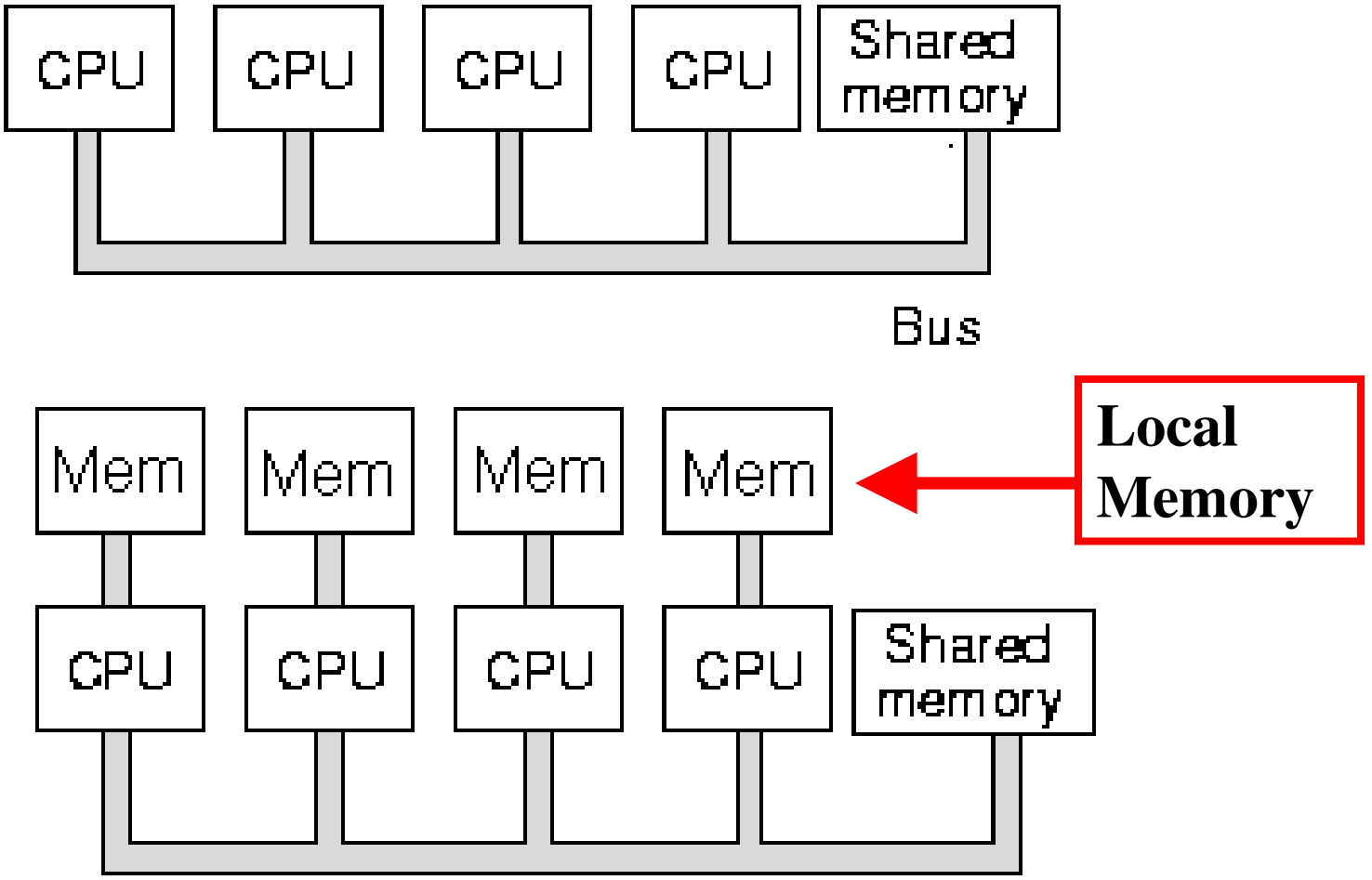
Bus-Based Computer



Anatomy of a Modern PC



Multiprocessors



Conclusion

- Principle of *abstraction*, used to build systems as layers
- *Pliable Data*: a program determines what it is
- *Stored program* concept: instructions are just data
- Principle of *Locality*, exploited via memory hierarchy
- Greater performance by exploiting *parallelism* (pipeline)
- *Compilation v. interpretation* to move downward through layers of system
- Principles/Pitfalls of *Performance Measurement*