

# Introduction

Rodolfo Azevedo

MO801



# Why Processors Evolve?

- New microarchitectural findings
  - Caches, branch predictions, renaming, ...
- Technology scaling
  - Relative speeds of gates vs wires
- Workload evolution
  - New software (desktop, mobile, server, UI, ...)

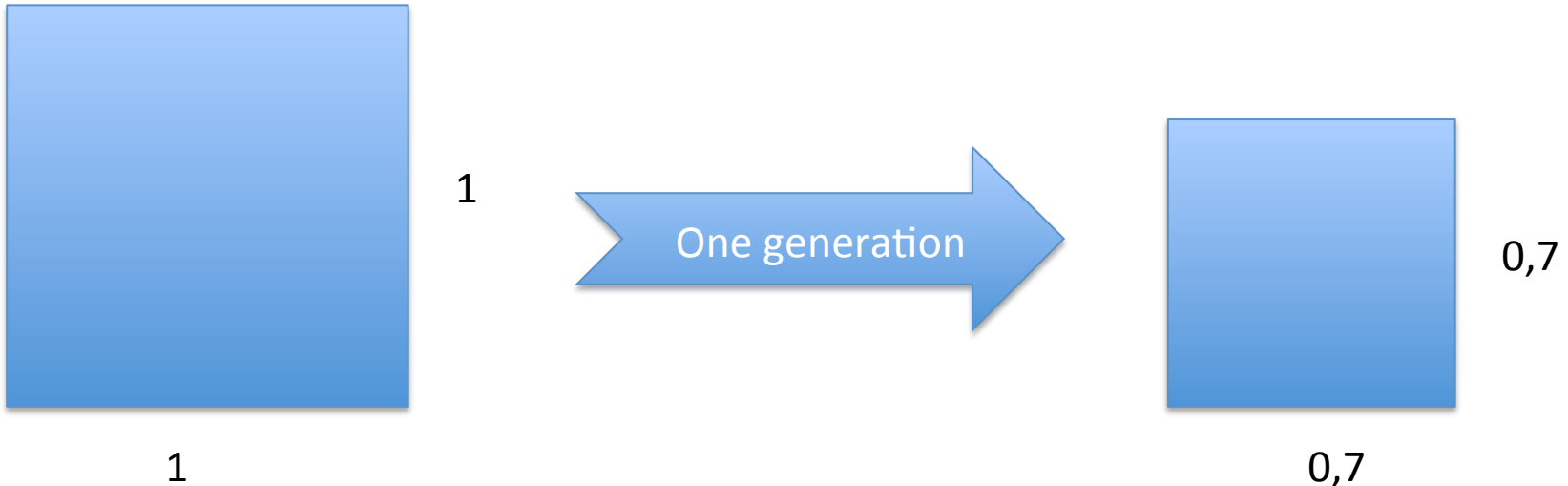


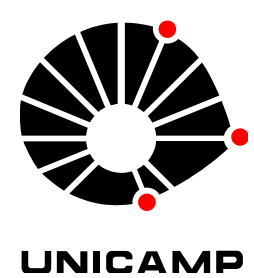
# Architecture vs Microarchitecture

- Architecture is the model
  - x86, ARM, ...
- Microarchitecture is implementation
  - Intel i7, AMD K8, ...
- Instruction Set Architecture can be seen as the border
  - ISA can make it easier or more difficult do implement microarchitectures

# Moore's Law

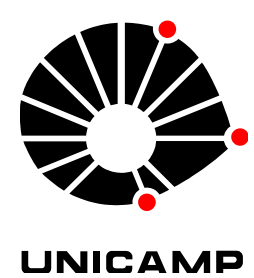
- Transistor density double every 2 years
  - 90nm, 65nm, 45nm, 32nm, 22nm, 14nm, ...





# Microarchitecture Classification

- Pipeline vs Nonpipeline
- In-Order vs Out-of-Order
- Scalar vs SuperScalar
- Vector
- Multicore
- Multithreaded



# Pipeline vs Nonpipelined

- Split execution into phases
- Increase Instruction Level Parallelism (ILP)
- Cost effective
- Almost all processors



# In-Order vs Out-of-Order

- In-Order executes instructions in the order they appeared in the binary
- Out-of-Order executes instructions as soon as they are able to
- How can a processor execute one instruction before the previous one?
  - What is necessary?
  - What about ILP?
  - What about hardware complexity?



# Scalar vs Superscalar

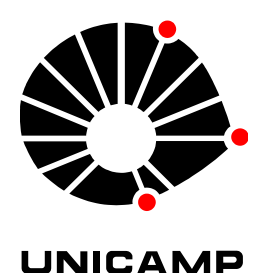
- Scalar cannot execute more than 1 instruction in at least one of the pipeline stages
  - $IPC \leq 1$
- Superscalar is able to execute more than 1 instruction in any of its stages
  - IPC can be  $> 1$
- VLIW
  - In-order
  - Instruction order stated in the binary
  - Latencies exposed to programmer/compiler





# Vector

- Instructions that operate on vectors
  - Traditionally big processors with big vectors
  - Currently instruction extensions for vector operations
- SIMD (Single Instruction, Multiple Data)
- Intel AVX is vectors can hold 8 single-precision FP elements



# Multicore

- Multiple cores in the same die
- Can run multiple programs or multiple threads
  - Process vs Threads?
- Each core is, usually, completely independent
- Cache coherence



# Multithreaded

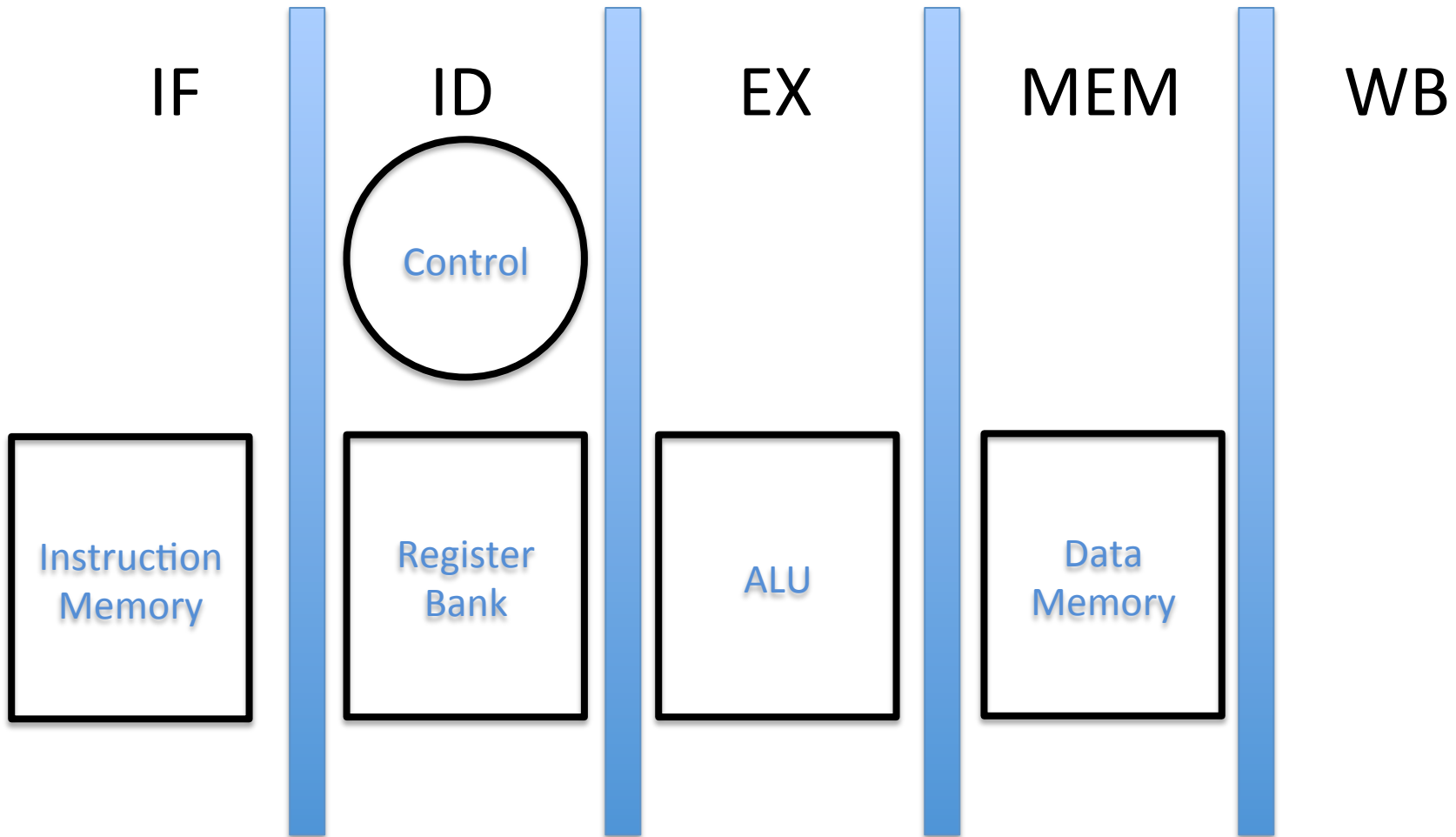
- Share physical resources among different threads on the same core
- Only need to replicate a few set of resources
  - PC
  - Register files
  - Some control registers
- Uses less resources than multicore
- A multicore processor can be multithreaded



# Market Segments

- Servers
  - Multiple processors, computing power and power dissipation
- Desktop
  - Computing power and noise
- Mobile
  - Energy consumption, computing power
- Ultramobile
  - Energy consumption
- Embedded
  - Multiple aspects such as performance vs power consumption

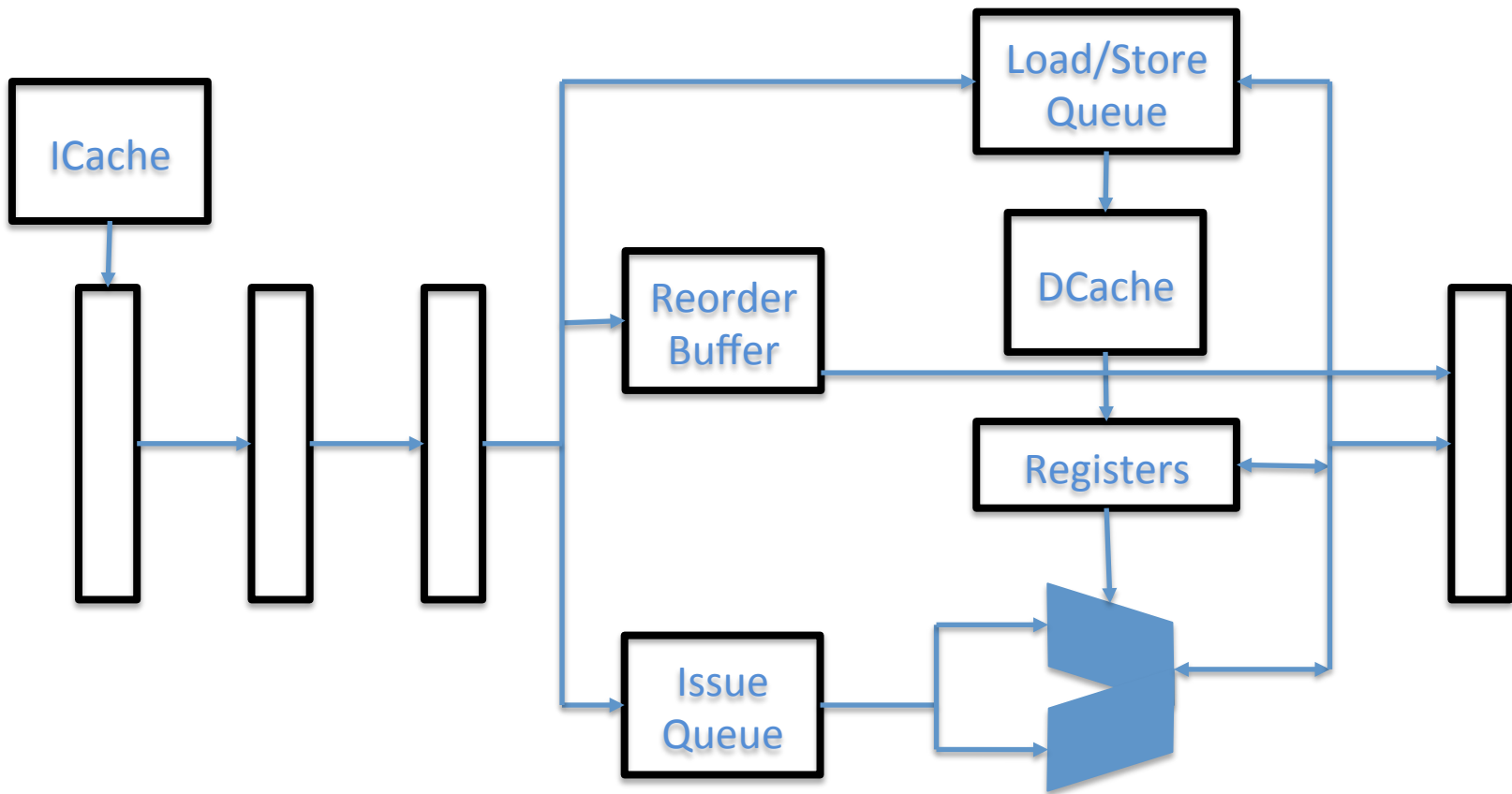
# Conventional 5-stage Pipeline



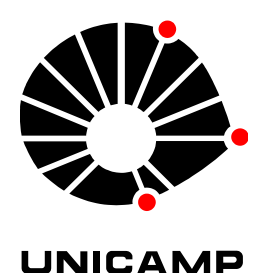


UNICAMP

# Detailed View



Fetch    Decode    Rename    Issue    Execution    Write-back    Commit



# Short Presentations

- 3 minutes
- Topics discussed in last class
- Any question?