

CPS104
Computer Organization
Lecture 1

August 30, 1999

Dietolf (Dee) Ramm

Slides available on:
<http://www.cs.duke.edu/~dr/cps104.html>

CPS104: Computer Organization

Instructor: Dietolf (Dee) Ramm

Office: LSRC D226, 660-6532 ramm@cs.duke.edu

Office Hours: Mon. 10:15; Thu., 10:00, or by appt.

TA: Sathish Govindarajan

Office: LSRC D208 660-4008 gsat@cs.duke.edu

Office Hours: Mon., 1:30; Tue. 10:00, or by appt.
or by appt.

UTAs:

Text: Patterson & Hennessy, **Computer Organization & Design: The Hardware / Software Interface (2nd edition).**

Web page: <http://www.cs.duke.edu/~dr/cps104.html>

Newsgroup: duke.cs.cps104

Course Outline:

- **1.Introduction to Computer Organization.**
 - ◆ What is in the box.
 - ◆ Integer and Floating point representation.
 - ◆ Basic data structures.
- **2.Instruction Set Architecture.**
 - ◆ The MIPS Processor.
 - ◆ Assembly level programming.
 - ◆ Instructions and data types representations.
 - ◆ Addressing, procedure calls and Exceptions.
 - ◆ Linking & Loading.
- **3.Digital Logic:**
 - ◆ Introduction: Digital Gates and Boolean Algebra.
 - ◆ Arithmetic and Logic circuits,
 - ◆ Other Functional Units
 - ◆ Flip-flops, Registers and Tristate drivers

Course Outline (continue):

- **4. Single Cycle Per Instruction Processor.**
 - ◆ The Datapath.
 - ◆ Executing Instructions
 - ◆ Control
- **5. Interrupts.**
- **6. The Memory Hierarchy.**
 - ◆ Cache Memory.
 - ◆ Virtual Memory and Paging.
- **7. I/O Devices.**
 - ◆ I/O storage devices.
 - ◆ I/O buses and arbitration
 - ◆ LANs and WANs.
- **8. Advanced processors:**
 - ◆ Pipelined Processor.
 - ◆ Super-Scalar processor.
- **9. Advanced Computer Architecture. (If there is time).**
 - ◆ Fast Interconnects
 - ◆ Parallel Machines

Grading

- **Grade breakdown**

- ◆ **Midterm Exams: 30%**
- ◆ **Final Exam: 20%**
- ◆ **Homework Assignments 50%**

- **Late homework policy:**

- ◆ **No sad stories!**
- ◆ **No “cooperation” on homework (Unless specified in the assignment).**
- ◆ **10% reduction for each day late.**
- ◆ **No credit after the homework was graded and handed back.**

- **Grades posted on home page:**

- ◆ **Password protected Access**
- ◆ **Written/email request for changes to grades.**

Homework-0

- Send me (ramm@cs.duke.edu) email message with: your name, year, major and a short description of your computer science / engineering / science / math background.
- Readings: Chapter-1, next time we start on data representations (Chapter-4).

Course Problems

- **Can't make midterm**
 - ◆ Tell us early and we will schedule alternate time
- **Forgot to turn in homework/ Dog ate the computer, network down.....**
 - ◆ **I do not accept excuses!**
 - ◆ If you have a legitimate problem. Talk to me early, email me a reminder.
- **What is cheating?**
 - ◆ Studying together in groups is encouraged
 - ◆ All written work must be your own. **Programs that are substantially the same as others will receive a grade of 0!**
 - ◆ Common examples of cheating: running out of time on a assignment and then pick up someone else's output, person asks to borrow solution "just to take a look", copying an exam question, ...

What You Will Learn

- **The basic operation of a computer**
 - ◆ no “magic”
 - ◆ primitive operations (instructions)
 - ◆ arithmetic
 - ◆ instruction sequencing and processing
 - ◆ memory
 - ◆ input/output
 - ◆ etc.
- **Understand the relationship between abstractions**
 - ◆ interface design
 - ◆ high-level program to control signals (SW -> HW)
- **Software performance depends on understanding underlying HW**

CPS104: Course Overview

Computer Design

Instruction Set Design

- Machine Language
- Compiler View
- "Computer Architecture"
- "Instruction Set Processor"

"Building Architect"

Computer Hardware Design

- Machine Implementation\
- Logic Designer's View
- "Processor Architecture"
- "Computer Organization"

"Construction Engineer"

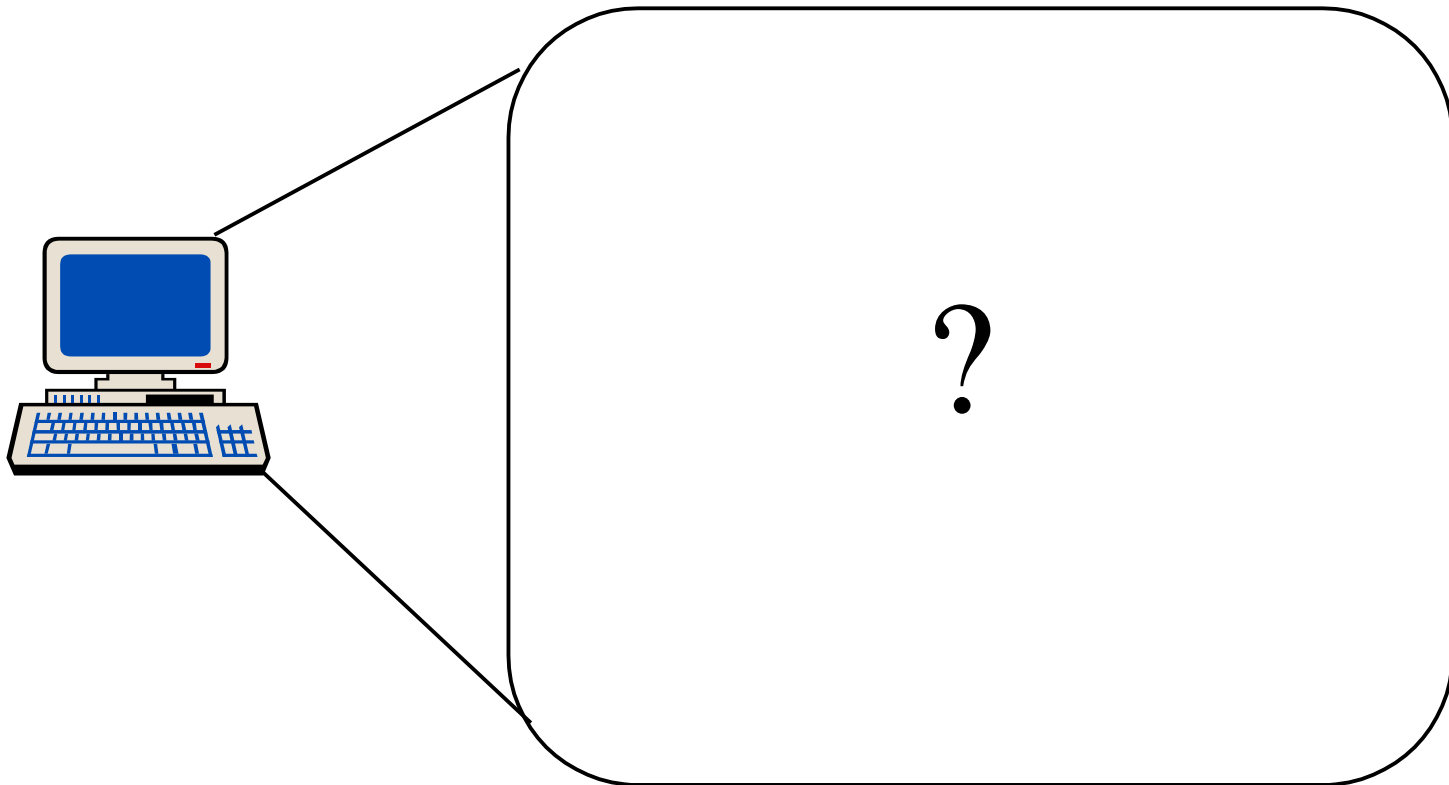
Few people design computers! Very few design instruction sets!

Many people design computer components.

Very many people are concerned with computer function, in detail.

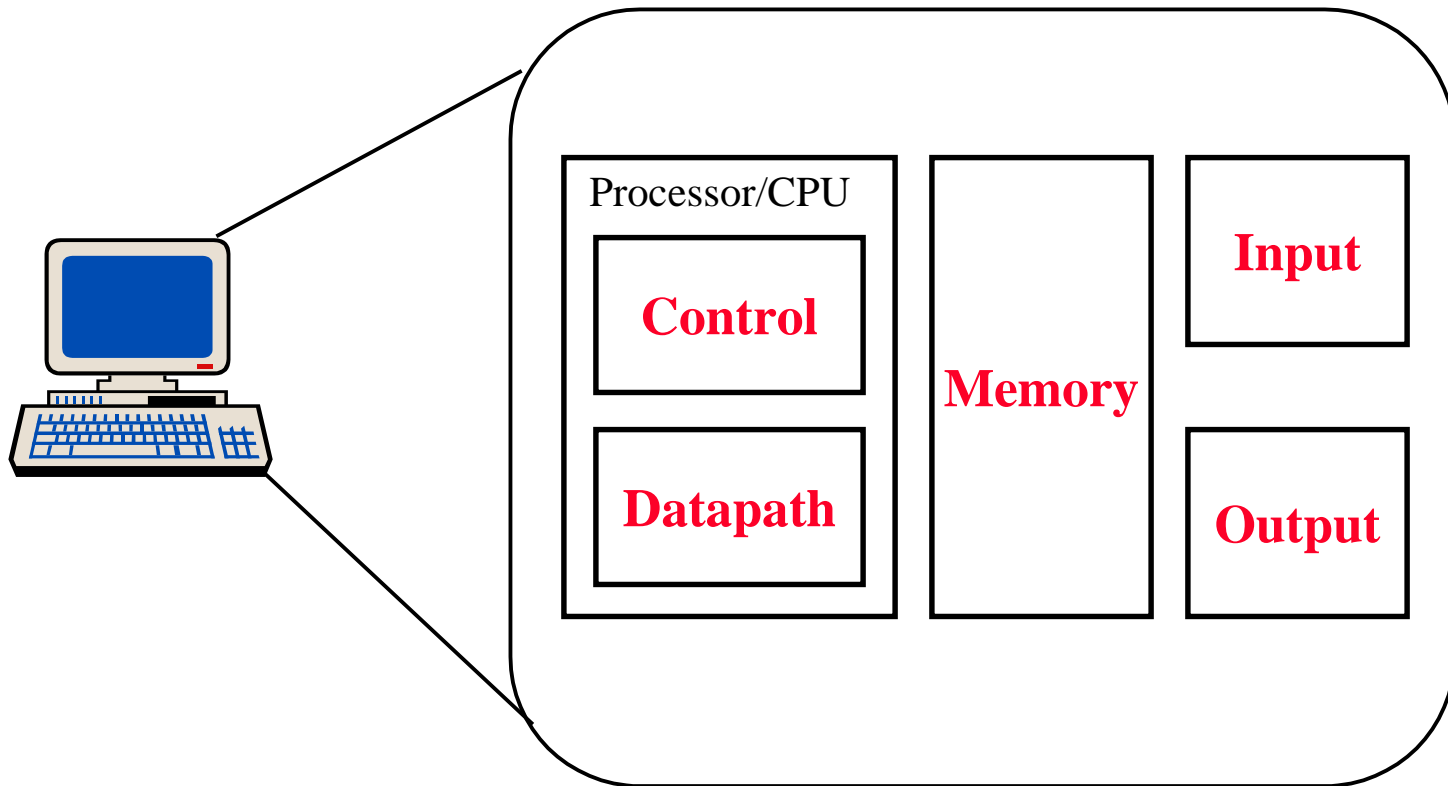
The Big Picture

- What is inside a computer?
- How does it execute my program?

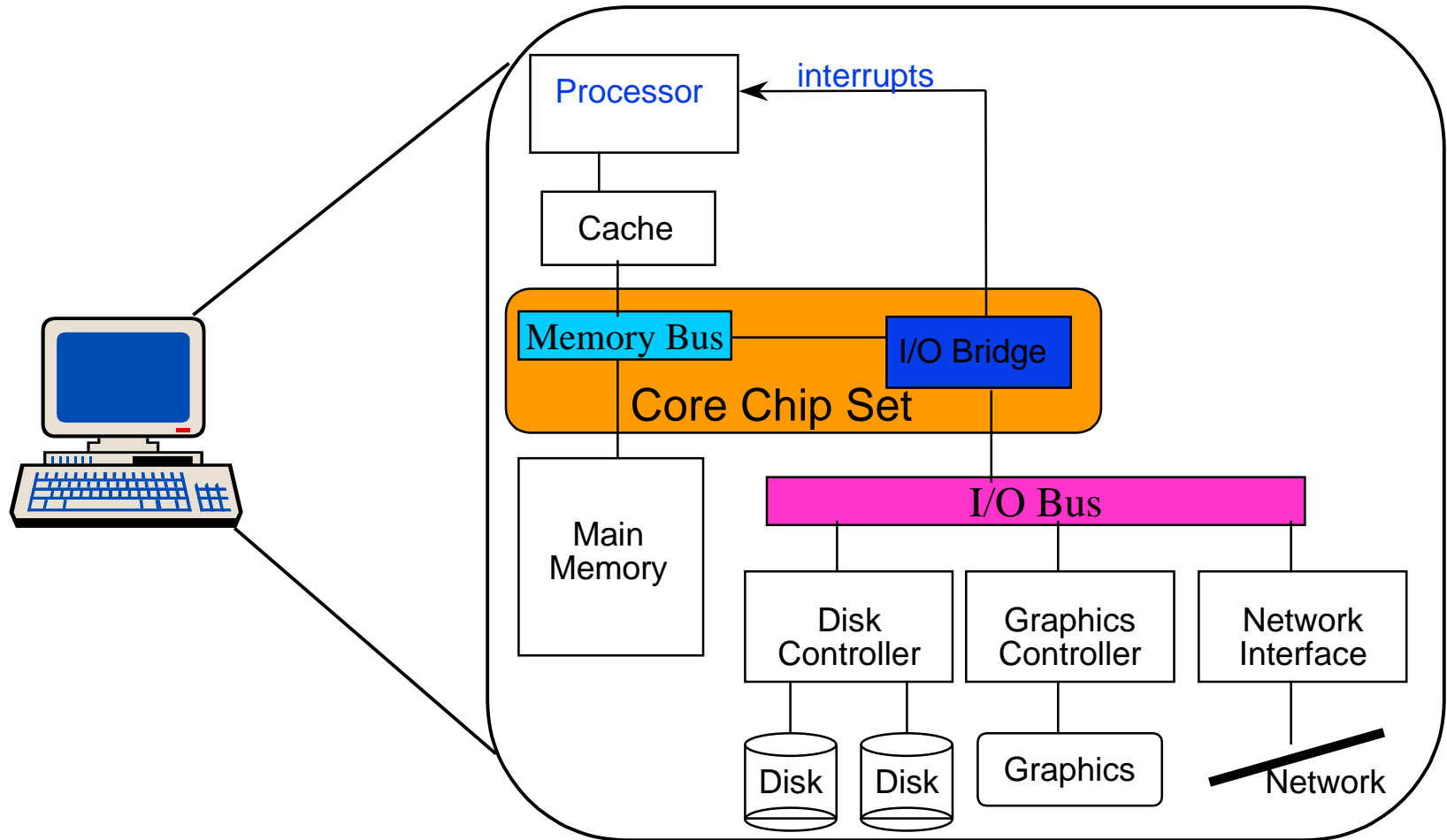


The Big Picture

- The Five Classic Components of a Computer

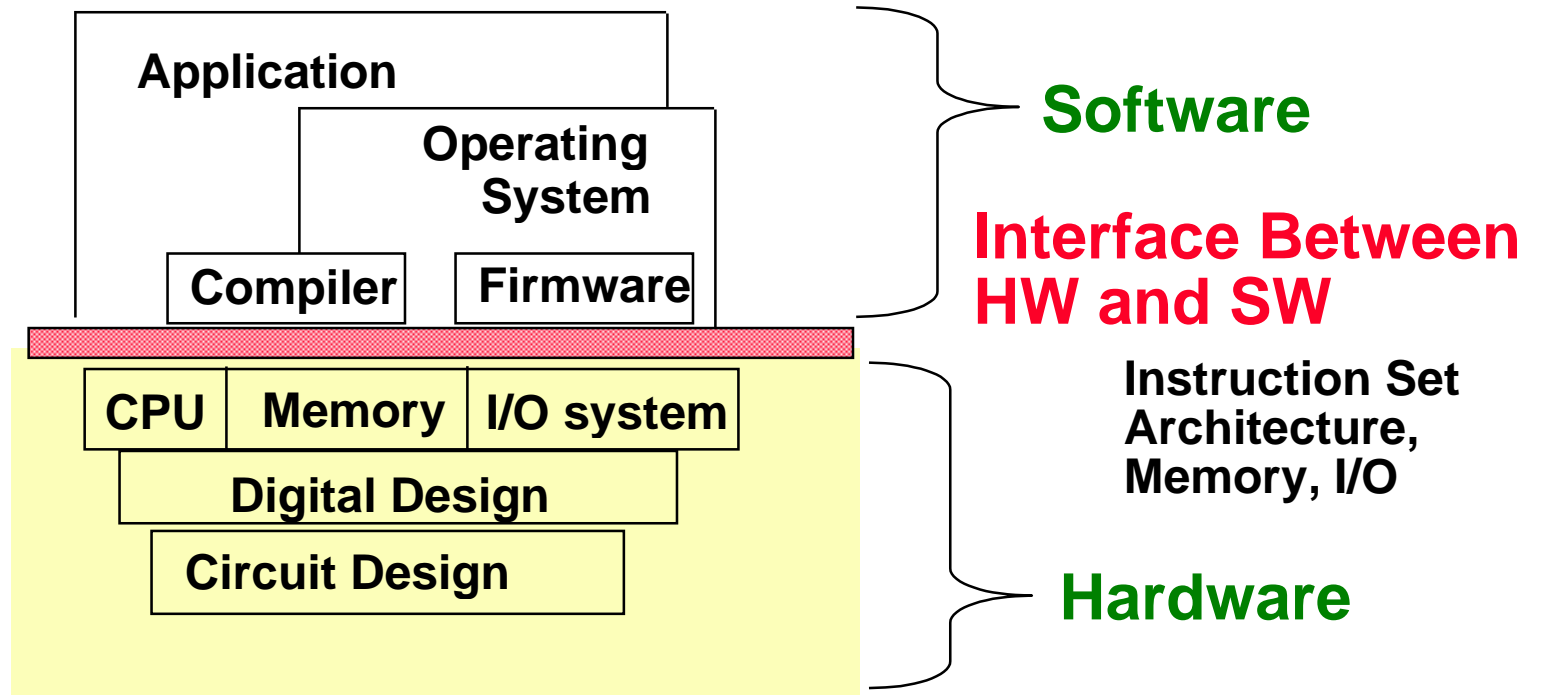


System Organization



What is Computer Architecture?

- Coordination of levels of abstraction



- Under a set of rapidly changing *Forces*

Levels of Representation

High Level Language Program

Compiler

Assembly Language Program

Assembler

Machine Language Program

Machine Interpretation

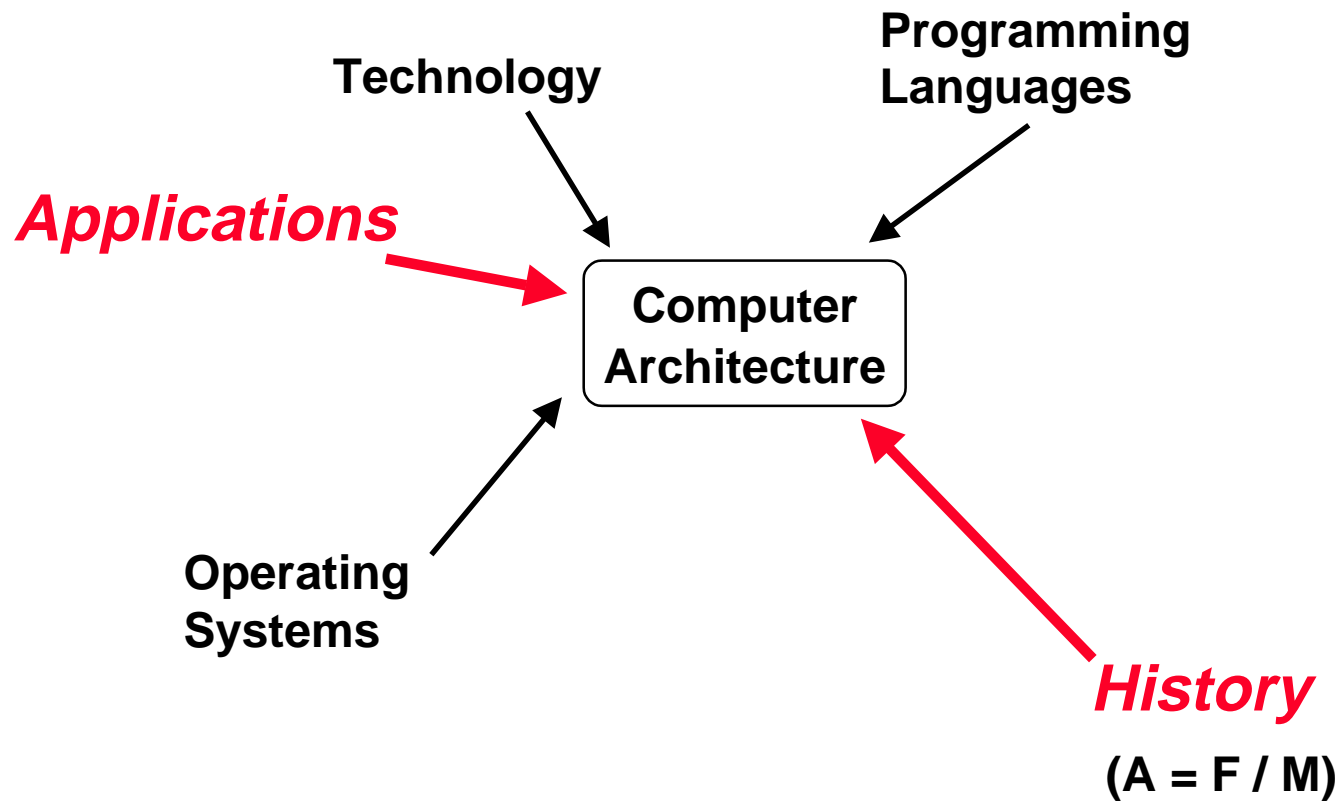
Control Signal Specification

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

```
lw $15, 0($2)  
lw $16, 4($2)  
sw $16, 0($2)  
sw $15, 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
```

Forces on Computer Architecture

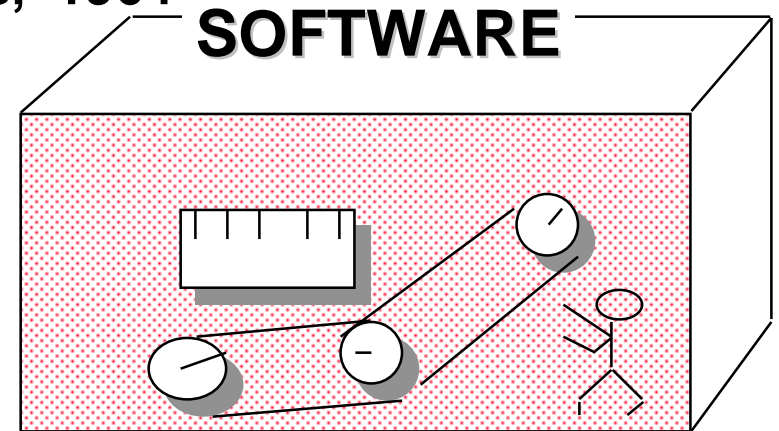


Instruction Set Architecture

- . . . the attributes of a [computing] system as seen by the programmer, i.e. the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls the logic design, and the physical implementation.

Amdahl, Blaaw, and Brooks, 1964

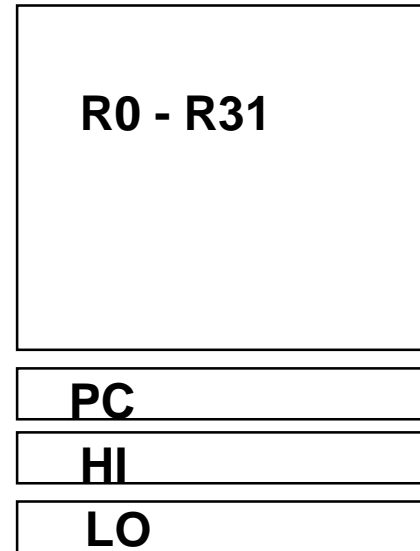
- Organization of Programmable Storage
- Data Types & Data Structures: Encoding & Representations
- Instruction Formats
- Instruction (or Operation Code) Set
- Modes of Addressing and Accessing Data Items and Instructions
- Exceptional Conditions



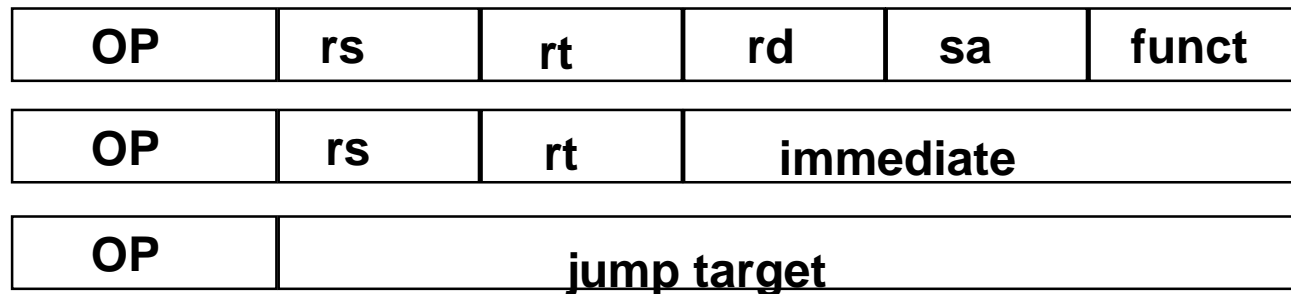
MIPS I Instruction Set Architecture

- **Instruction Categories**

- ◆ **Load/Store**
- ◆ **Computational**
- ◆ **Jump and Branch**
- ◆ **Floating Point**
- ◆ **Memory Management**
- ◆ **Special**



3 Instruction Formats: all 32 bits wide



Organization

ISA Level

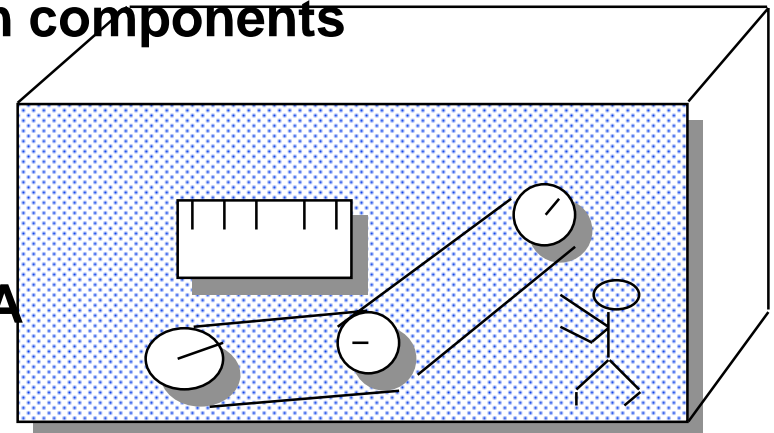
FUs & Interconnect

Logic Designer's View

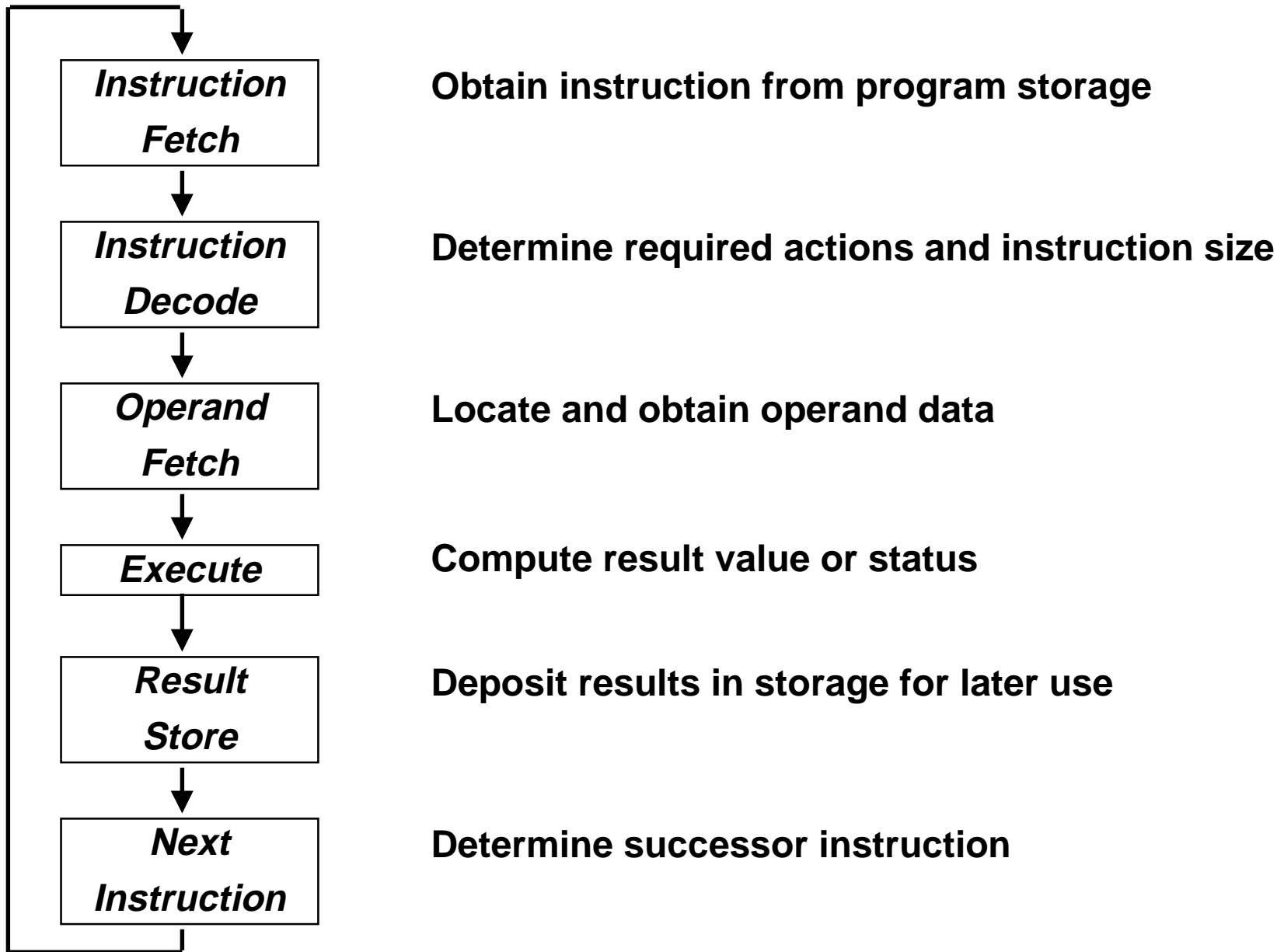
- Capabilities & Performance Characteristics of Principal Functional Units
(e.g., Registers, ALU, Shifters, Logic Units, ...)
- Ways in which these components are interconnected
- nature of information flows between components
- logic and means by which such information flow is controlled.

Choreography of FUs to realize the ISA

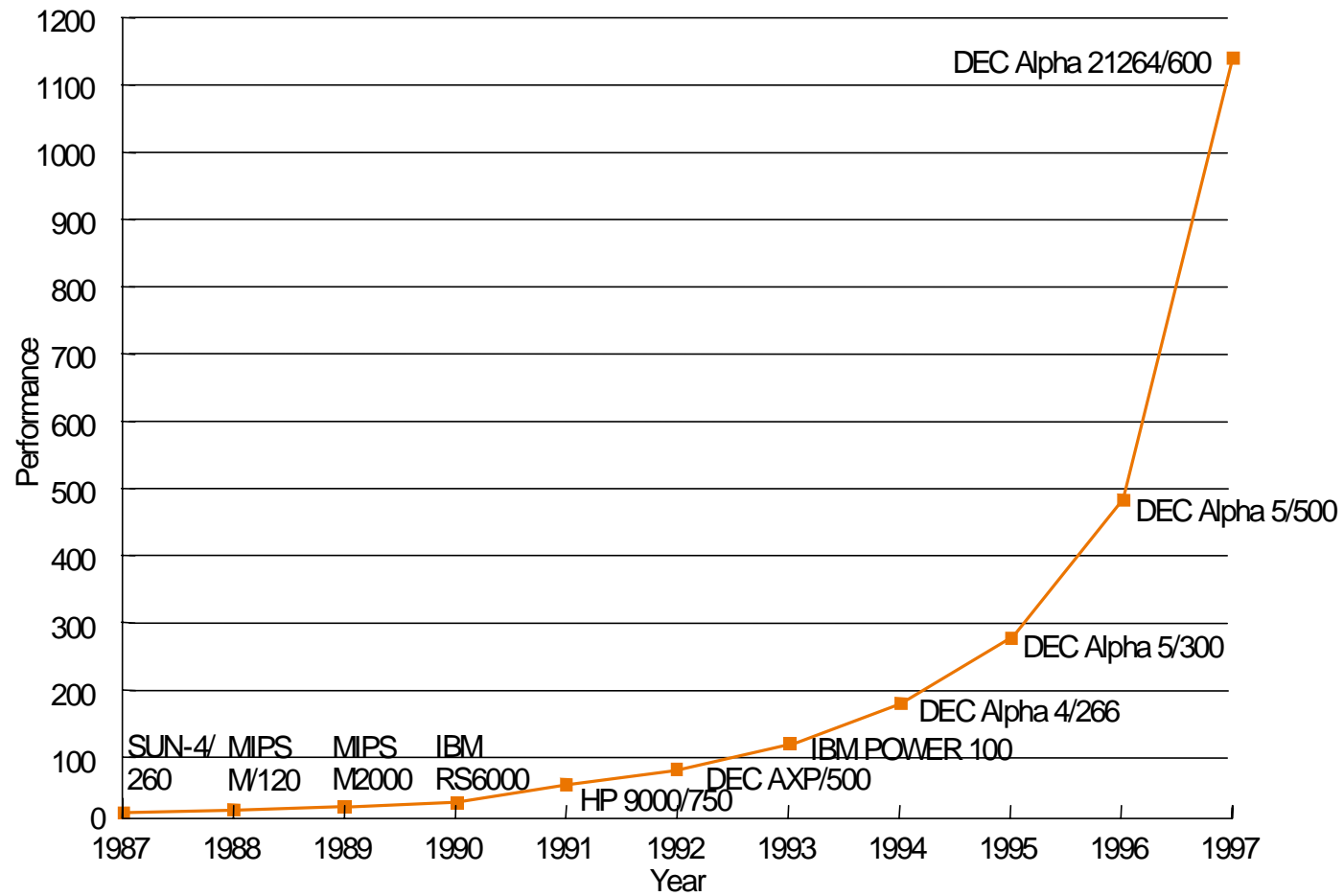
Register Transfer Level Description



Execution Cycle

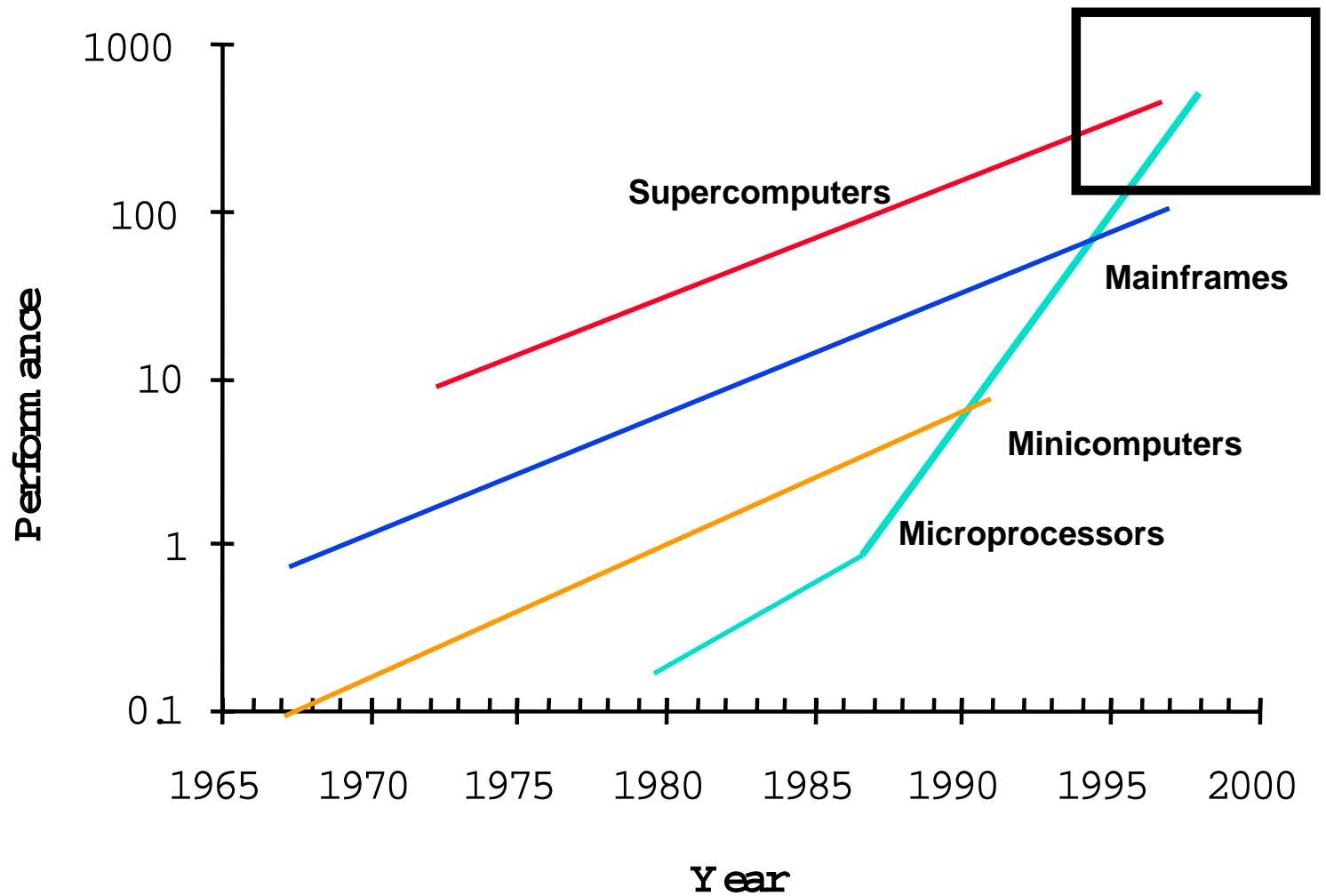


Processor Performance

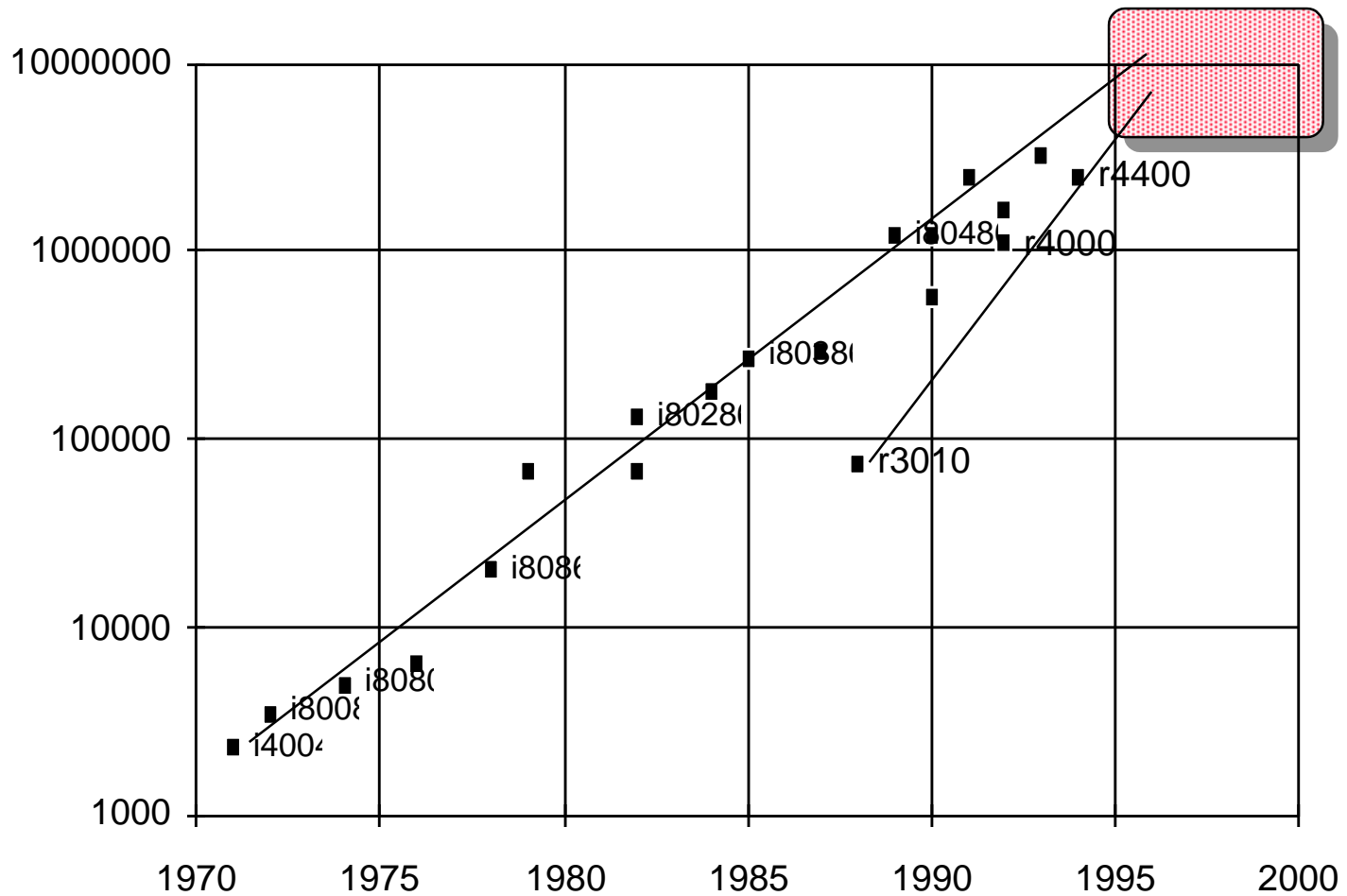


Copyright 1998 Morgan Kaufmann Publishers, Inc. All Rights Reserved

Performance Trends

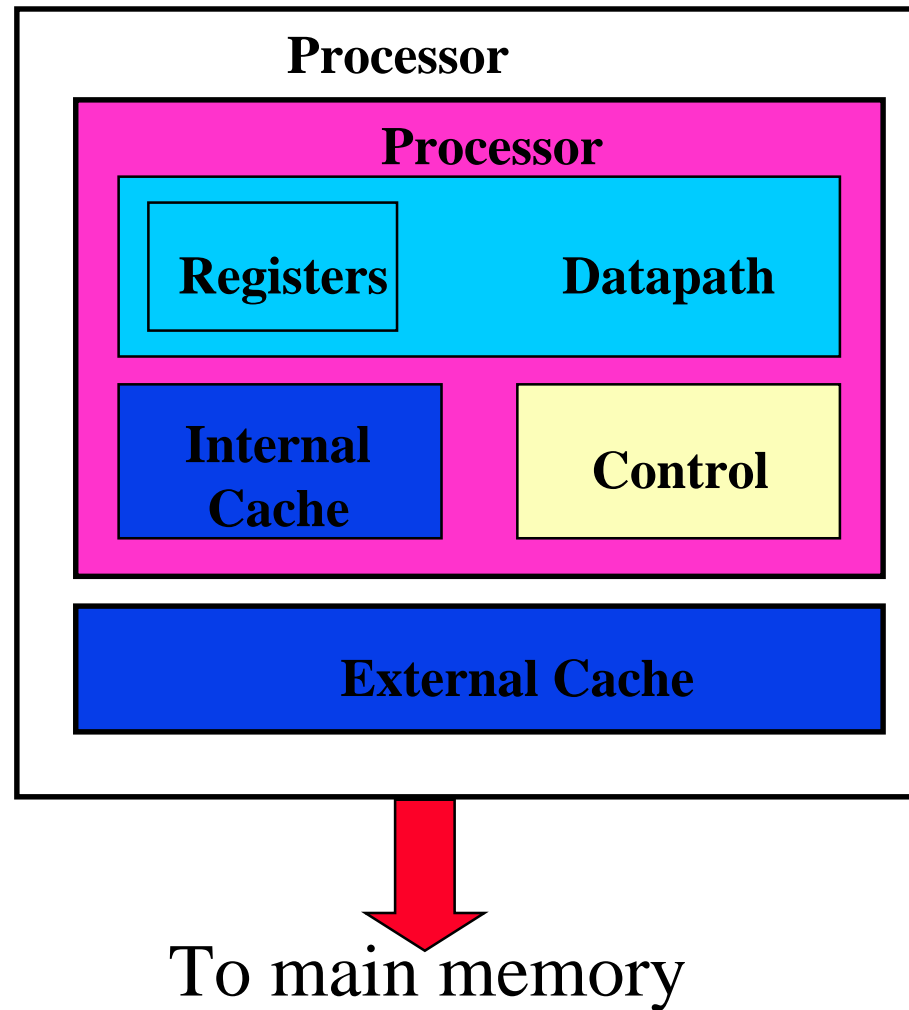


Technology: Microprocessor Logic Density

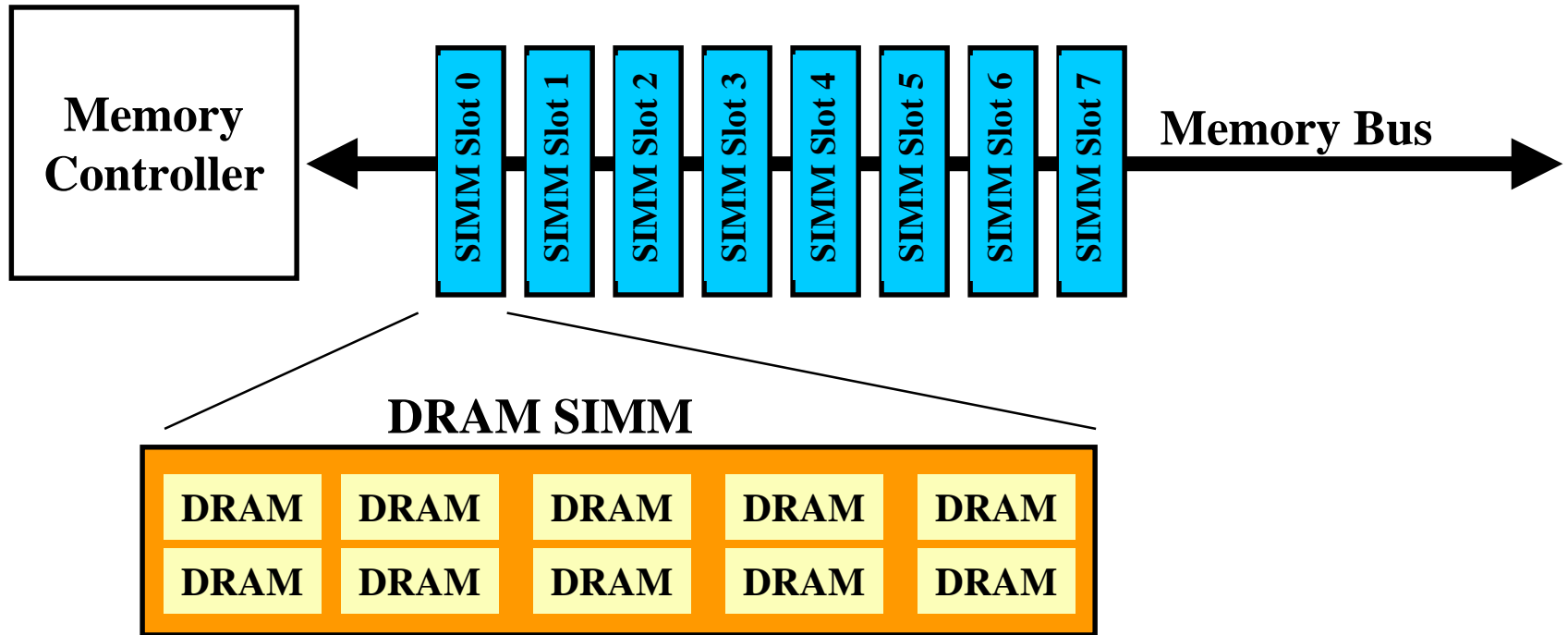


Memory: 4x every 3 years

Processor and Caches



Memory



Summary

Goal

- Understand basic operation of a computer

Why?

- Software performance is affected/determined by HW capabilities
- Future Computer Architects (Processor or System)

Summary (Continued)

Agenda

- **Map “high-level” software to instructions**
- **Instructions are composed of hardware primitives**
 - ◆ how to use them
 - ◆ how to implement them
 - ◆ why a particular primitive
- **Memory for storing instructions and data**
 - ◆ Main memory
 - ◆ Caches
 - ◆ interaction with operating system
- **Input/Output**

Summary (Continued)

- **All computers consist of five components**
 - ◆ **Processor: (1) datapath and (2) control**
 - ◆ **(3) Memory**
 - ◆ **(4) Input devices and (5) Output devices**
- **Not all “memory” created equally**
 - ◆ **Cache: fast (expensive, small) memory close to the processor**
 - ◆ **Main memory: slower, cheaper, larger memory farther from processor**
- **Input and output (I/O) devices has the messiest organization**
 - ◆ **Wide range of speed: graphics vs. keyboard**
 - ◆ **Wide range of requirements: speed, standard, cost ... etc.**

Next Time

- **Data Representations**

Reading

- **Chapter 4.1-4.3, 4.8 pages 275-280**
- **Read Chapter 1, Skim 2**
- **Start reading Chapter 3**