

---

**CS152**  
**Computer Architecture and Engineering**  
**Lecture 1**

**August 27, 1997**

**Dave Patterson (<http://cs.berkeley.edu/~patterson>)**

**lecture slides: <http://www-inst.eecs.berkeley.edu/~cs152/>**

# Overview

---

- **Intro to Computer Architecture (30 minutes)**
- **Administrative Matters (5 minutes)**
- **Course Style, Philosophy and Structure (15 min)**
- **Break (5 min)**
- **Organization and Anatomy of a Computer (25) min)**

# What is “Computer Architecture”

---

**Computer Architecture =  
Instruction Set Architecture +  
Machine Organization**

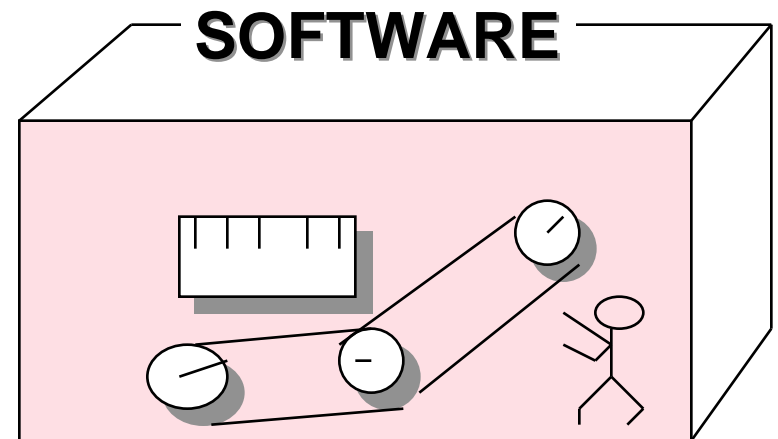
# Instruction Set Architecture (subset of Computer Arch.)

---

... 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.

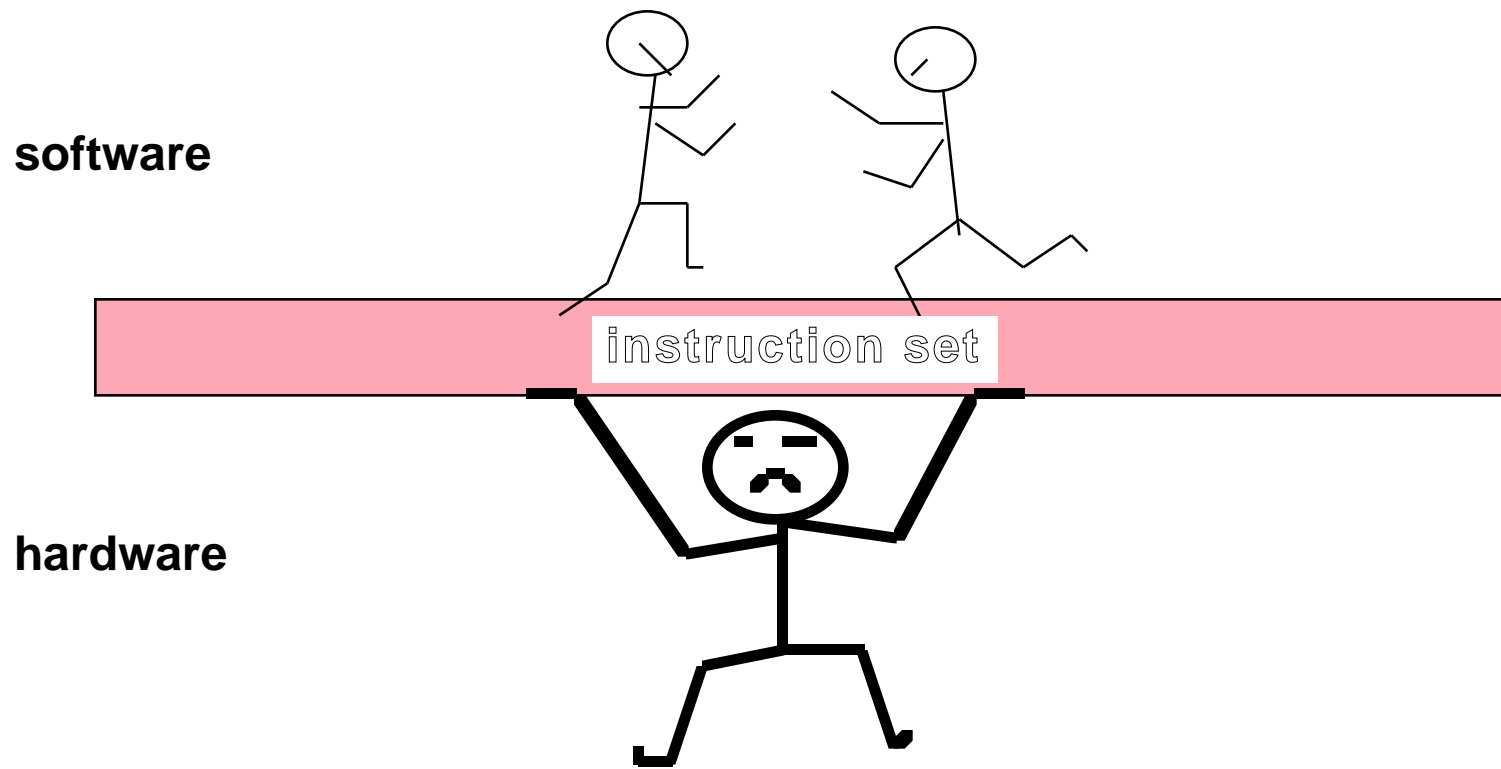
– Amdahl, Blaaw, and Brooks, 1964

- Organization of Programmable Storage
- Data Types & Data Structures: Encodings & Representations
- Instruction Set
- Instruction Formats
- Modes of Addressing and Accessing Data Items and Instructions
- Exceptional Conditions



# The Instruction Set: a Critical Interface

---



## Example ISAs (Instruction Set Architectures)

---

- **Digital Alpha** (v1, v3) 1992-97
- **HP PA-RISC** (v1.1, v2.0) 1986-96
- **Sun Sparc** (v8, v9) 1987-95
- **SGI MIPS** (MIPS I, II, III, IV, V) 1986-96
- **Intel** (8086, 80286, 80386, 80486, Pentium, MMX, ...) 1978-96

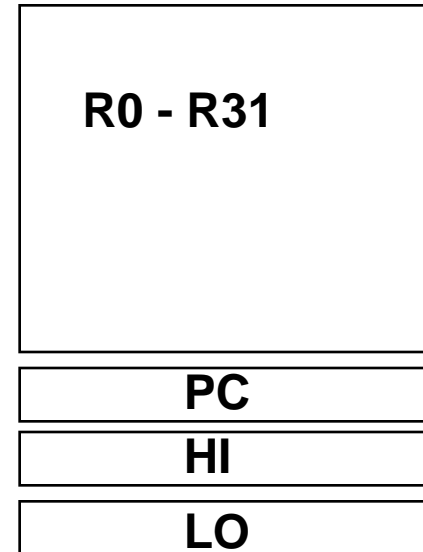
# MIPS R3000 Instruction Set Architecture (Summary)

---

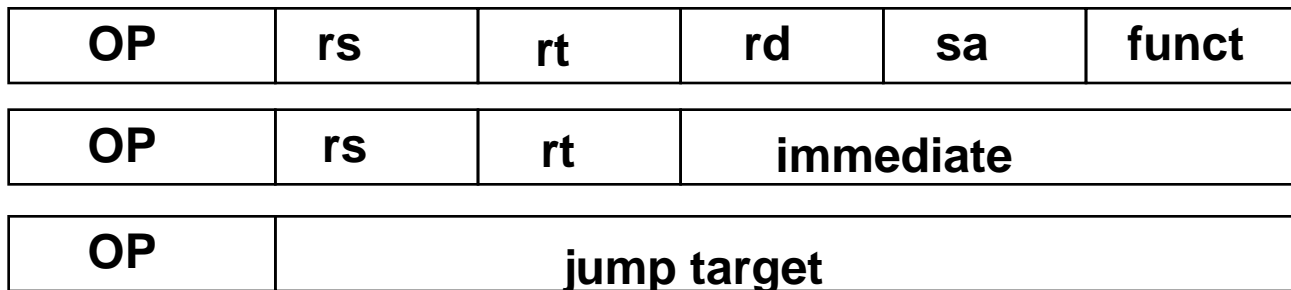
## ◦ Instruction Categories

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

## Registers



## 3 Instruction Formats: all 32 bits wide



Q: How many already familiar with MIPS ISA?

# Organization

---

- **Capabilities & Performance Characteristics of Principal Functional Units**

- (e.g., Registers, ALU, Shifters, Logic Units, ...)

- **Ways in which these components are interconnected**

- **Information flows between components**

- **Logic and means by which such information flow is controlled.**

- **Choreography of FUs to realize the ISA**

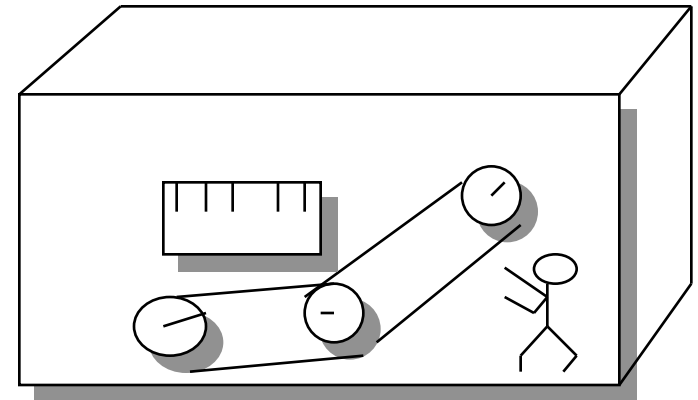
- **Register Transfer Level (RTL) Description**

*Logic Designer's View*

ISA Level

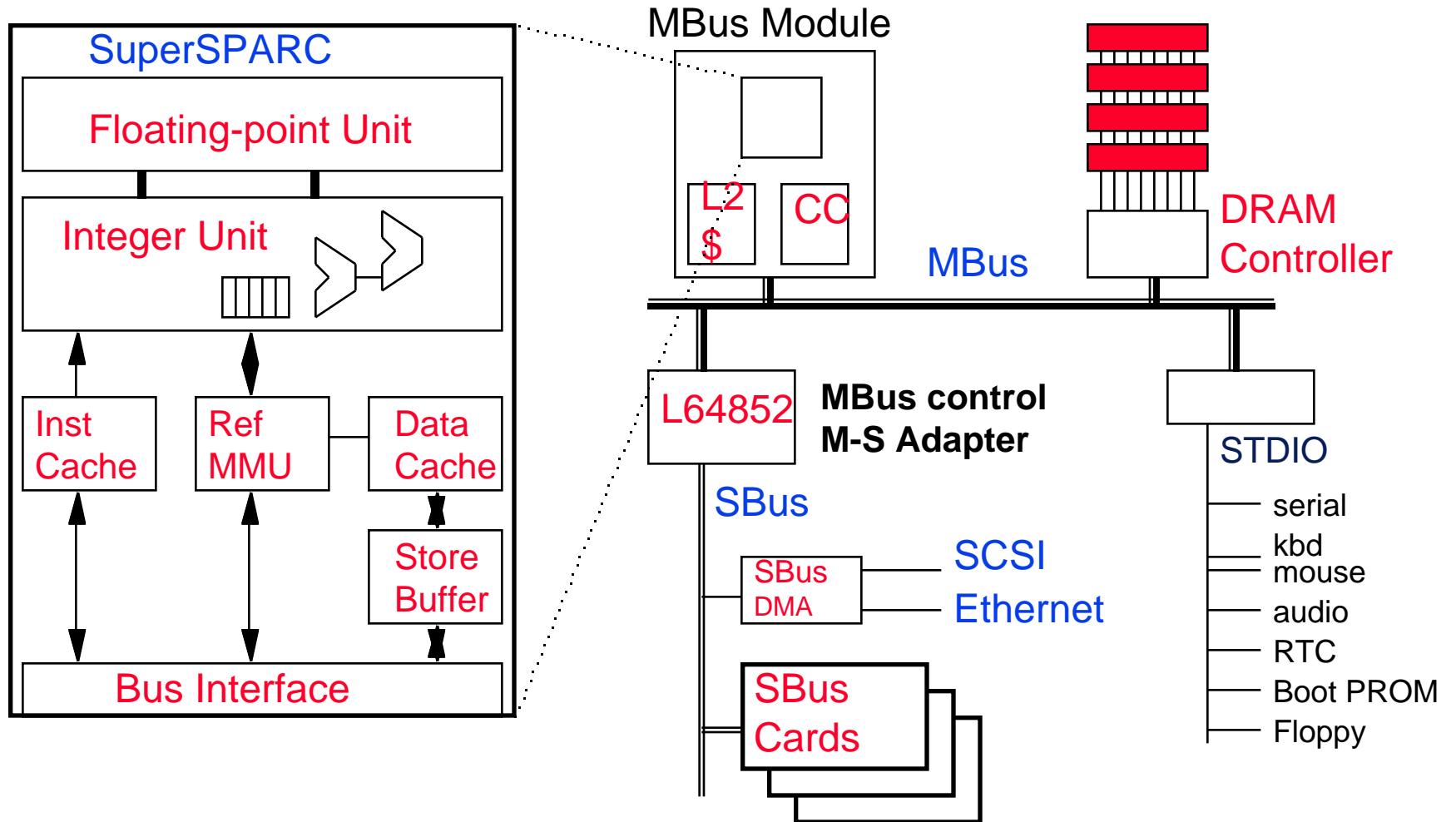
---

FUs & Interconnect



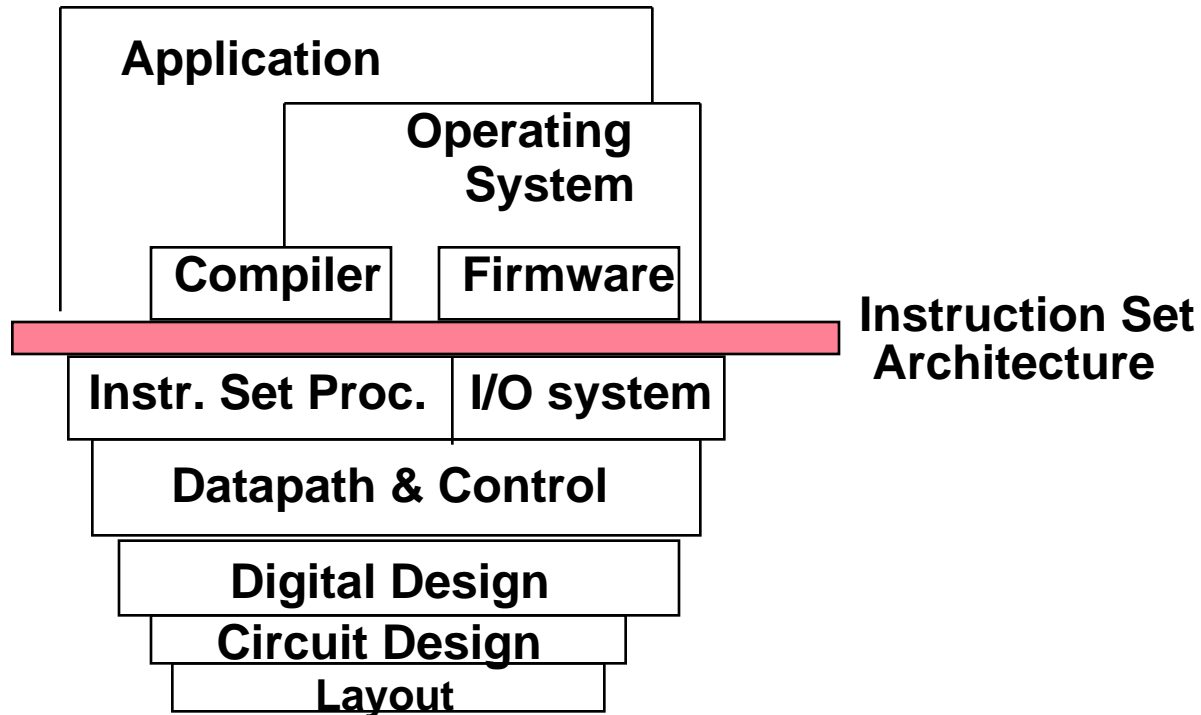
# Example Organization

## ◦ TI SuperSPARC™ TMS390Z50 in Sun SPARCstation20



# What is “Computer Architecture”?

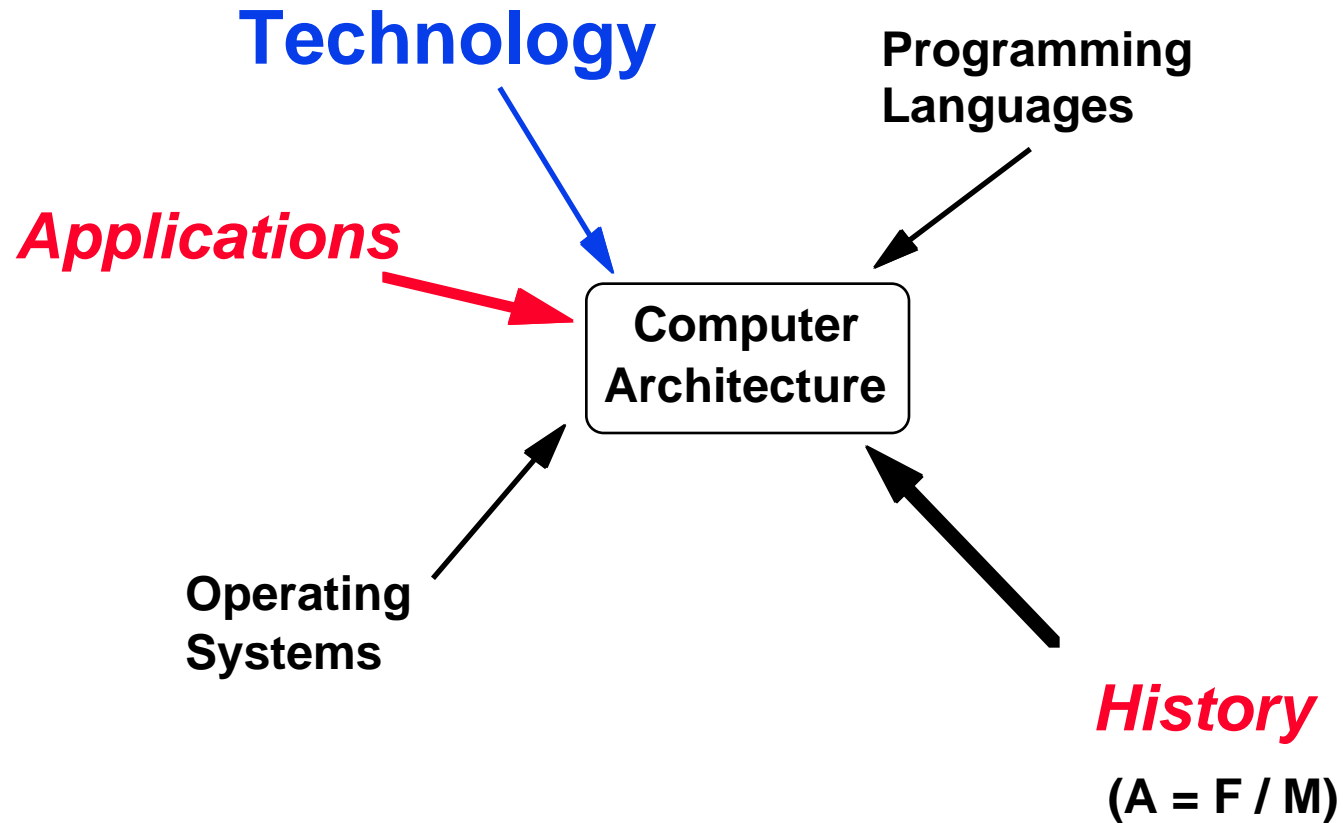
---



- Coordination of many *levels of abstraction*
- Under a rapidly *changing set of forces*
- Design, Measurement, *and* Evaluation

# Forces on Computer Architecture

---

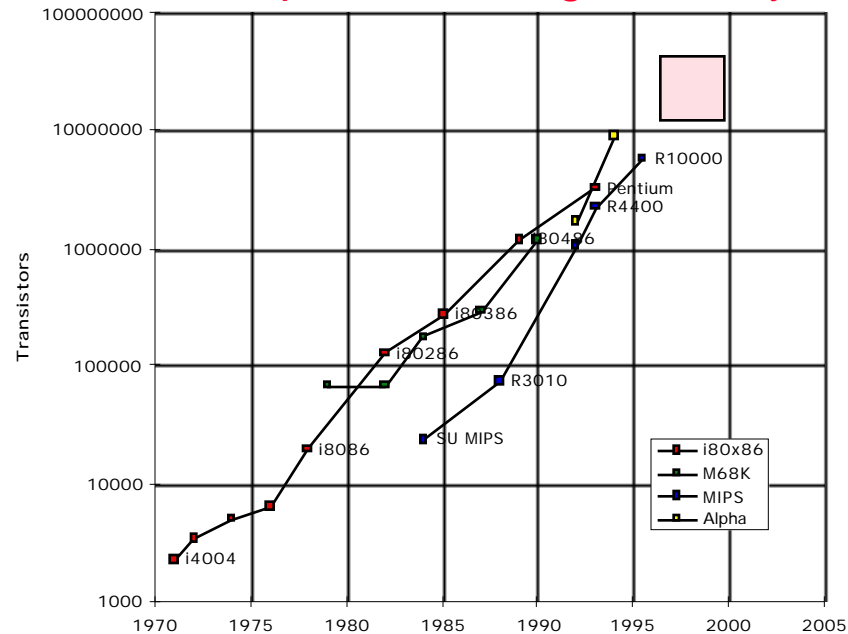


# Technology

## DRAM chip capacity

DRAM	
Year	Size
1980	64 Kb
1983	256 Kb
1986	1 Mb
1989	4 Mb
1992	16 Mb
1996	64 Mb
1999	256 Mb
2002	1 Gb

## Microprocessor Logic Density



◦ In ~1985 the single-chip processor (32-bit) and the single-board computer emerged

- => workstations, personal computers, multiprocessors have been riding this wave since

◦ In the 2002+ timeframe, these may well look like mainframes compared single-chip computer (maybe 2 chips)

# Technology => dramatic change

---

## ◦ Processor

- logic capacity: about 30% per year
- clock rate: about 20% per year

## ◦ Memory

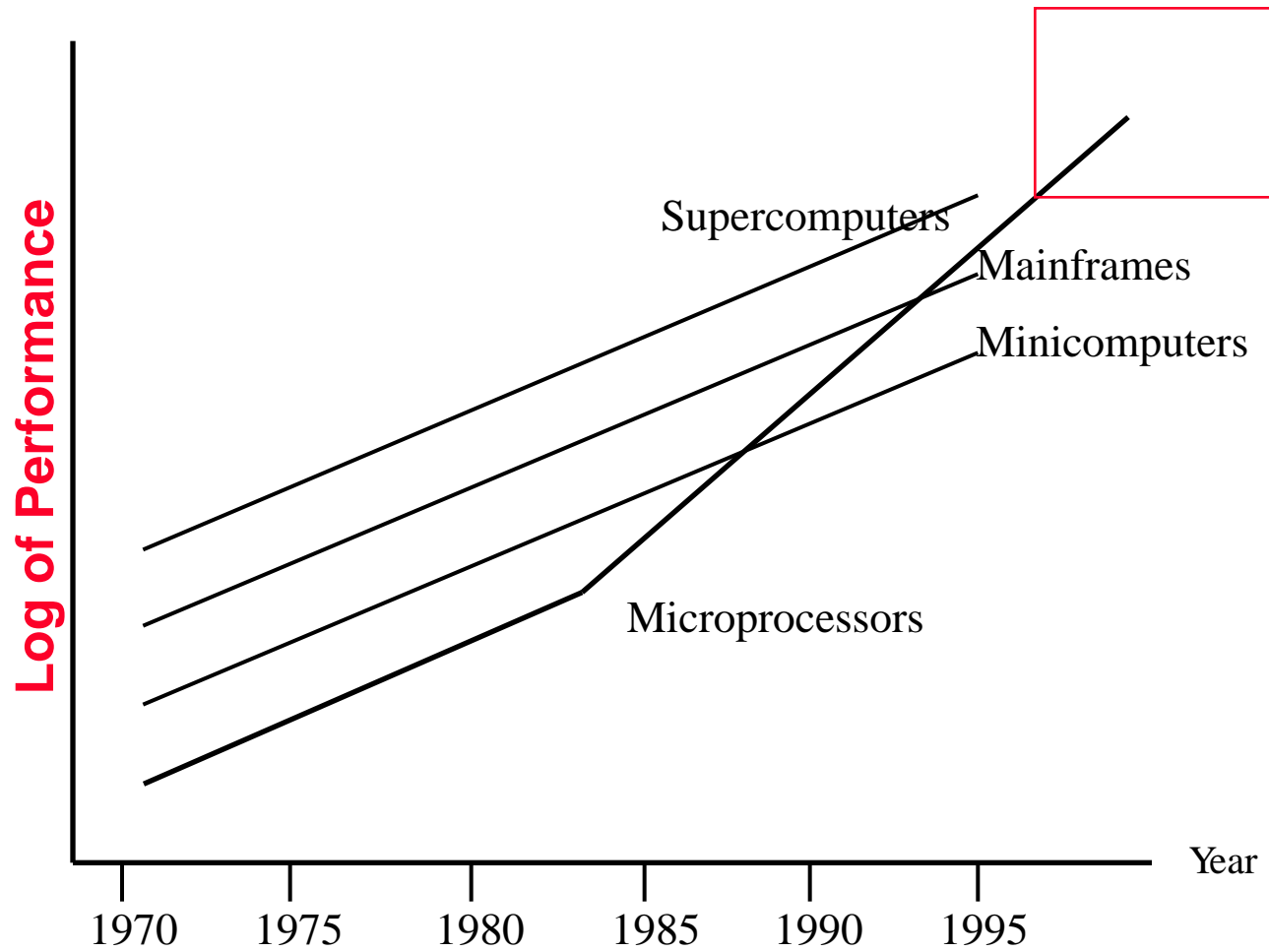
- DRAM capacity: about 60% per year (4x every 3 years)
- Memory speed: about 10% per year
- Cost per bit: improves about 25% per year

## ◦ Disk

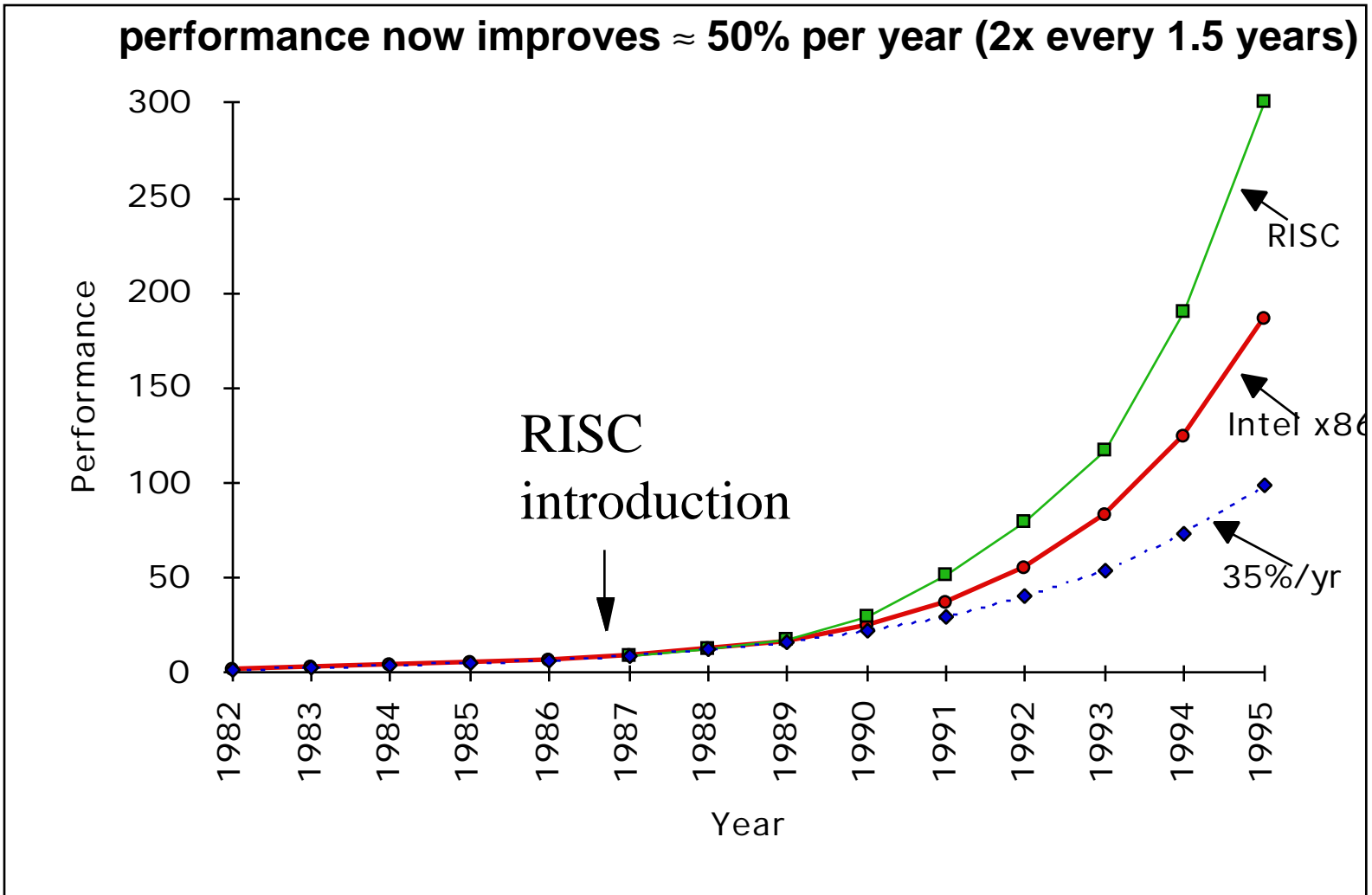
- capacity: about 60% per year

# Performance Trends

---



# Processor Performance (SPEC)



Did RISC win the technology battle and lose the market war?

# Applications and Languages

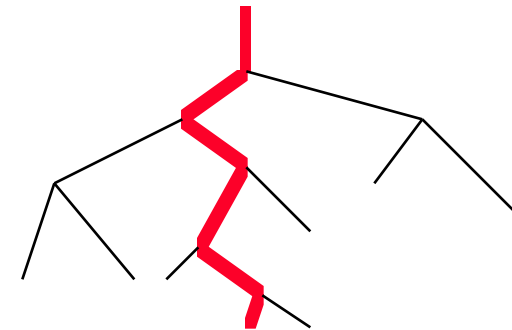
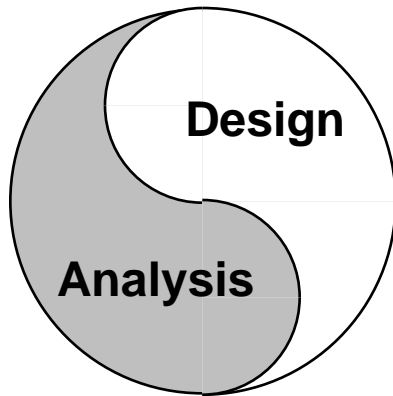
---

- CAD, CAM, CAE, . . .
- Lotus, DOS, . . .
- Multimedia, . . .
- The Web, . . .
- JAVA, . . .
- ???

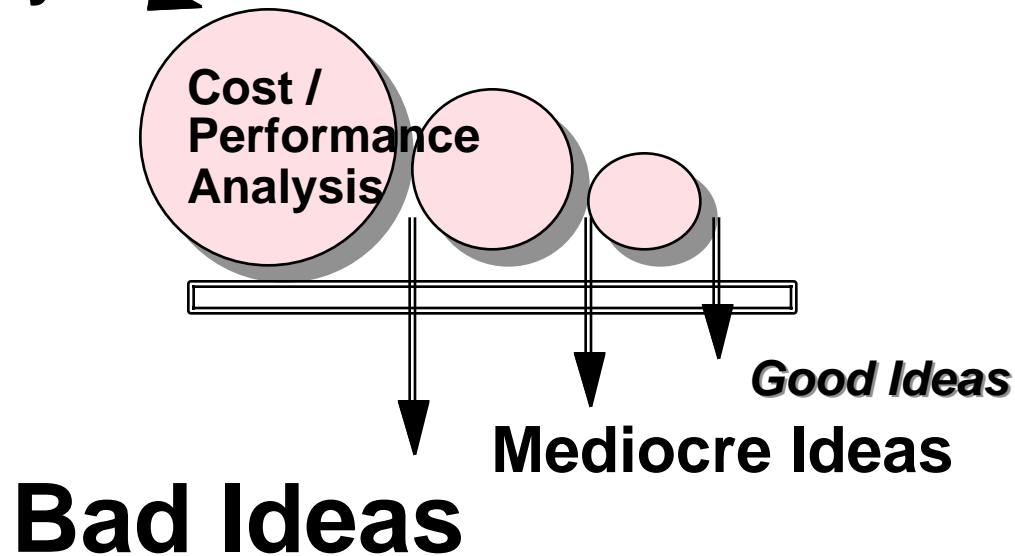
# Measurement and Evaluation

---

Architecture is an iterative process  
-- searching the space of possible designs  
-- at all levels of computer systems



**Creativity** →



# Why do Computer Architecture?

---

- **CHANGE**
- **It's exciting!**
- **It has never been more exciting!**
- **It impacts every other aspect of electrical engineering and computer science**

# CS152: Course Content

---

## Computer Architecture and Engineering



Instruction Set Design

Interfaces

Compiler/System View

≈“Building Architect”

Computer Organization

Hardware Components

Logic Designer’s View

≈“Construction Engineer”

# CS152: So what's in it for me?

---

- **In-depth understanding of the inner-workings of modern computers, their evolution, and trade-offs present at the hardware/software boundary.**
  - Insight into fast/slow operations that are easy/hard to implement in hardware
- **Experience with the *design process* in the context of a large complex (hardware) design.**
  - Functional Spec --> Control & Datapath --> Physical implementation
  - Modern CAD tools
- **Designer's "Conceptual" toolbox.**

# Conceptual tool box?

---

- **Evaluation Techniques**
- **Levels of translation (e.g., Compilation)**
- **Levels of Interpretation (e.g., Microprogramming)**
- **Hierarchy (e.g, registers, cache, mem,disk,tape)**
- **Pipelining and Parallelism**
- **Static / Dynamic Scheduling**
- **Indirection and Address Translation**
- **Synchronous and Asynchronous Control Transfer**
- **Timing, Clocking, and Latching**
- **CAD Programs, Hardware Description Languages, Simulation**
- **Physical Building Blocks (e.g., CLA)**
- **Understanding Technology Trends**

# Course Structure

---

- **Design Intensive Class --- 75 to 150 hours per semester per student**  
**MIPS Instruction Set ---> Standard-Cell implementation**

- **Modern CAD System (PowerView):**

## **Schematic capture and Simulation**

**Design Description**

**Computer-based "breadboard"**

- **Behavior over time**
- **Before construction**

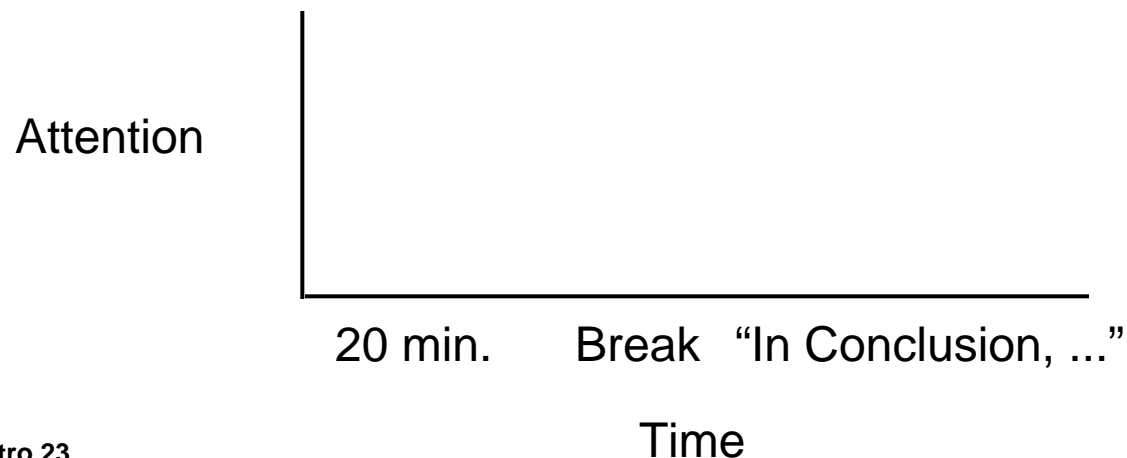
- **Lectures:**

- **Review: 2 weeks on ISA, arithmetic**
- **1 week on technology & HDL**
- **4 weeks on Proc. Design**
- **3 weeks on Memory and I/O**
- **1 week on fast networks and multiprocessors**
- **2 week by guest lecturers, field trip**
- **2 weeks exams, presentations**

# Typical Lecture Format

---

- **25-Minute Lecture**
- **5- Minute Administrative Matters**
- **20-Minute Lecture**
- **5-Minute Break (water, stretch)**
- **25-Minute Lecture**
- **Instructor will come to class early & stay after to answer questions**



## Course Administration

---

- Instructor: **David A. Patterson (patterson@cs)**  
**635 Soda Hall**  
**Office Hours(Tentative): WF 1-2**
- TAs: **Joe Gebis (gebis@eecs)**  
**Christoforos Kozyrakis (kozyraki@cs)**  
**Kirby Zhang (zhang?@eecs)**
- Labs: **Class Accounts on Soda Machines**
- Materials: **<http://www-inst.eecs/~cs152>**
- Newsgroup: **ucb.class.c152**
- Text: ***Computer Organization and Design:  
The Hardware/Software Interface,  
Second Edition, Patterson and Hennessy***
  - **Q: Need 2nd Edition?**  
**yes! >> 50% text changed, all exersizes changed all examples  
modernized, new sections, ...**

# Course Exams

---

- **Reduce the pressure of taking exams**
  - **Midterms: (approximately) Wednesday Oct 1 and Wed. Nov 5**
  - 3 hrs to take 1.5-hr test (5-8 PM, Sibley Auditorium)
  - Our goal: test knowledge vs. speed writing
  - Review meetings: Sunday before
  - Both mid-terms can bring summary sheets
- **Students/Staff meet over pizza after exam**
  - Goal: change to meet everyone I teach in person
  - 1st time: pay back royalty (1 pizza/book)

# Course Workload

---

◦ **Reasonable workload (if you have good work habits)**

- No final exam: Only 2 mid-terms
- Final project has been simplified
- Every lab feeds into the project
- Project teams have 4 or 5 members

◦ **Spring 1995 HKN workload survey  
(1 to 5, 5 being hardest)**

CS 150	4.2	CS 164	3.1
CS 152	3.4/3.5	CS 169	3.6
CS 162	3.9/4.0	CS 184	4.6

◦ **Spring 1997 HKN workload survey  
(1 to 5, 5 being hardest)**

CS 150	3.8	CS 164	4.0
CS 152	3.2	CS 169	3.2
CS 162	3.3	CS 184	3.3

◦ **Revised Science/Design units: now 3 Science, 2 Design**

# Homework Assignments and Project

---

- **Most assignment consists of two parts**
  - Individual Effort: Exercises from the text book
  - Team Effort: Lab assignments
- **Assignments go out on Friday**
  - Exercises due on a later Monday 10 AM in 283 Soda box
  - Labs due at beginning of discussion section
- **Lab Homeworks returned in discussion section**
  - To spread computer workload
  - put section time on them homeworks
- **Discussion sections this week**
  - 101            Th 2-4    405 DAVIS
  - 102            Th 12-2   75 EVANS
  - 103            Th 4-6    87 EVANS
- **Must turn in survey to be considered enrolled**

# My Goal

---

- **Show you how to understand modern computer architecture in its rapidly changing form.**
- **Show you how to design by leading you through the process on challenging design problems**
- **NOT to talk at you**
- 
- **SO...**
  - **ask questions**
  - **come to office hours**
  - **find me in the lab**
  - **...**

# Project/Lab Summary

---

- **Powerview runs on all HPs, but 349 and 273 Soda machines have local copies**
- **Lab assignments:**
  - Lab 0 C -> MIPS, SPIM (1 week)
  - Lab 1 Performance measurement, diagnostics (1 week)
  - Lab 2 Powerview / Fast ALU Design (1 week)
  - Lab 3 Single Cycle Processor Design (2 weeks)
  - Lab 4 Pipelined Processor Design (2 weeks)
  - Lab 5 Cache & DMA Design (2 weeks)
  - Lab 6 *Open ended work for final project*
- **2-hour discussion section so that can have project meetings in second hour (attendance required at project meetings); team in same section!**
- **Oral presentation and written report**

# Grading

---

## ◦ Grade breakdown

- Two Midterm Exams: 40% (combined)
- Design Project: 30%
- Homework Assignments 20%
- Project Group Participation 5%
- Class Participation: 5%

## ◦ No late homeworks or labs: our goal grade, return $\leq 1$ week

## ◦ Grades posted on home page

- Written/email request for changes to grades
- **Nov 26 deadline to correct scores**

## ◦ CS Division guideline upper division class GPA between 2.7 and 3.1.

- average 152 grade will be a B or B+; set expectations accordingly

# Course Problems

---

- **Can't make midterm**
  - Tell early us and we will schedule alternate time
- **Forgot to turn in homework/ Dog ate computer**
  - As a result of feedback, going to grade almost immediately so that can give results back quickly => late homeworks a hassle
- **What is cheating?**
  - Studying together in groups is encouraged
  - Work must be your own
  - Common examples of cheating: running out of time on a assignment and then pick up output, take homework from box and copy, person asks to borrow solution “just to take a look”, copying an exam question, ...
  - Better off to skip assignment (6 Labs + 6 homeworks 20% of grade)

# **Class decides on penalties for cheating; staff enforces**

---

## ◦ **Exercises (book):**

- 0 for problem
- 0 for homework assignment
- subtract full value for assignment
- subtract 2X full value for assignment

## ◦ **Labs leading to project (groups: only penalize individuals?)**

- 0 for problem
- 0 for homework assignment
- subtract full value for assignment
- subtract 2X full value for assignment

## ◦ **Exams**

- 0 for problem
- 0 for exam

# Project Simulates Industrial Environment

---

- **Project teams have  $\geq 4$  members in same discussion section**
  - video lecture by Sun expert on working in groups
- **Communicate with colleagues (team members)**
  - What have you done?
  - What answers you need from others?
  - You must document your work!!!
  - Everyone must keep an on-line notebook
- **Communicate with supervisor (TAs)**
  - How is the team's plan?
  - Short progress reports are required:
    - What is the team's game plan?
    - What is each member's responsibility?

# Things We Hope You Will Learn from 152

---

## ◦ **Keep it simple and make it work**

- Fully test everything individually and then together
- Retest everything whenever you make any changes
- Last minute changes are big “no nos”

## ◦ **Group dynamics. Communication is the key to success:**

- Be open with others of your expectations and your problems
- Everybody should be there on design meetings when key decisions are made and jobs are assigned

## ◦ **Planning is very important:**

- Promise what you can deliver; deliver more you promise
- Murphy's Law: things DO break at the last minute
  - Don't make your plan based on the best case scenarios
  - Freeze you design and don't make last minute changes

## ◦ **Never give up! It is not over until you give up.**

## What you should know from 61C, 150

---

- **Basic machine structure**
  - processor, memory, I/O
- **Read and write basic C programs**
- **Read and write in an assembly language**
  - MIPS preferred
- **Understand the steps in a make file and what they do**
  - compile, link, load & execute
- **Understand the concept of virtual memory**
- **Logic design**
  - logical equations, schematic diagrams, FSMs, components

## Getting into CS 152

---

- If not preenrolled, Fill out petition form
- Fill out survey and return Friday
- **Know the prerequisites**
  - CS 61C - assembly language and simple computer organization
  - CS 150 - Logic design
- **Prerequisite quiz on Friday Sept 5; Pass/Fail**
  - UC doesn't always enforce prerequisites
  - TA's will hold review sessions tomorrow+1 week
  - Need to pass prerequisite quiz to take CS 152

# Levels of Representation (61C Review)

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)**

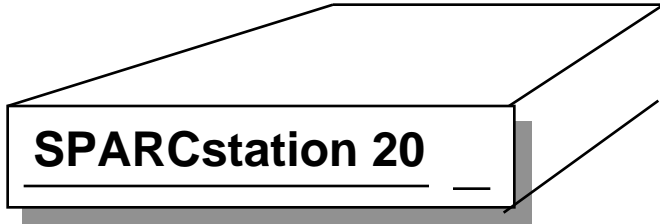
```
0000 1001 1100 0110 1010 1111 0101 1000
1010 1111 0101 1000 0000 1001 1100 0110
1100 0110 1010 1111 0101 1000 0000 1001
0101 1000 0000 1001 1100 0110 1010 1111
```

ALUOP[0:3] <= InstReg[9:11] & MASK

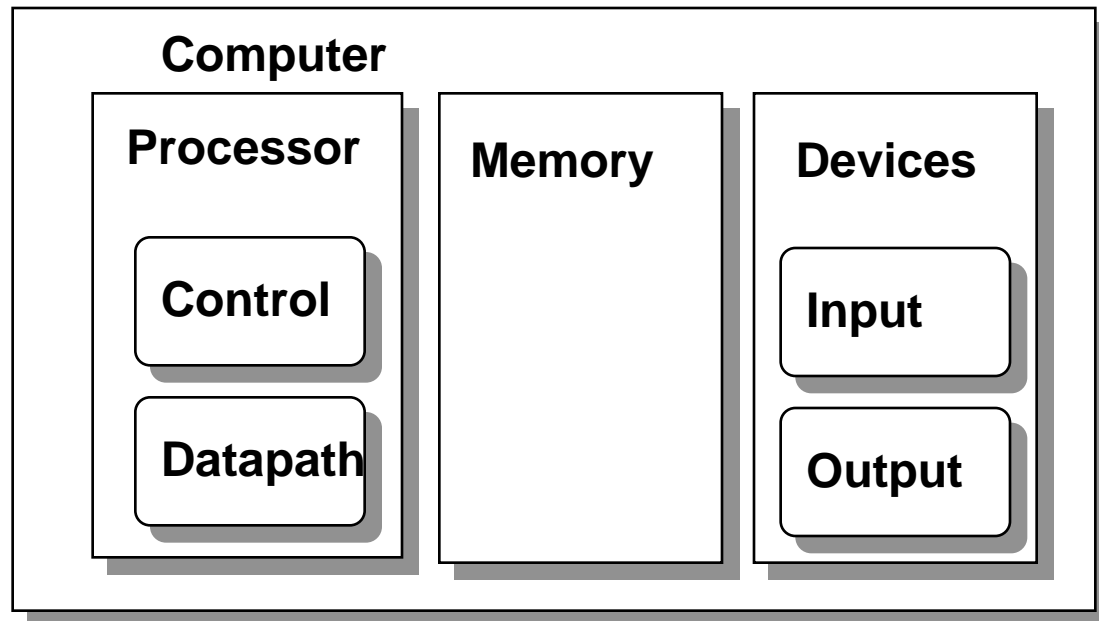
- o
- o

# Levels of Organization

---

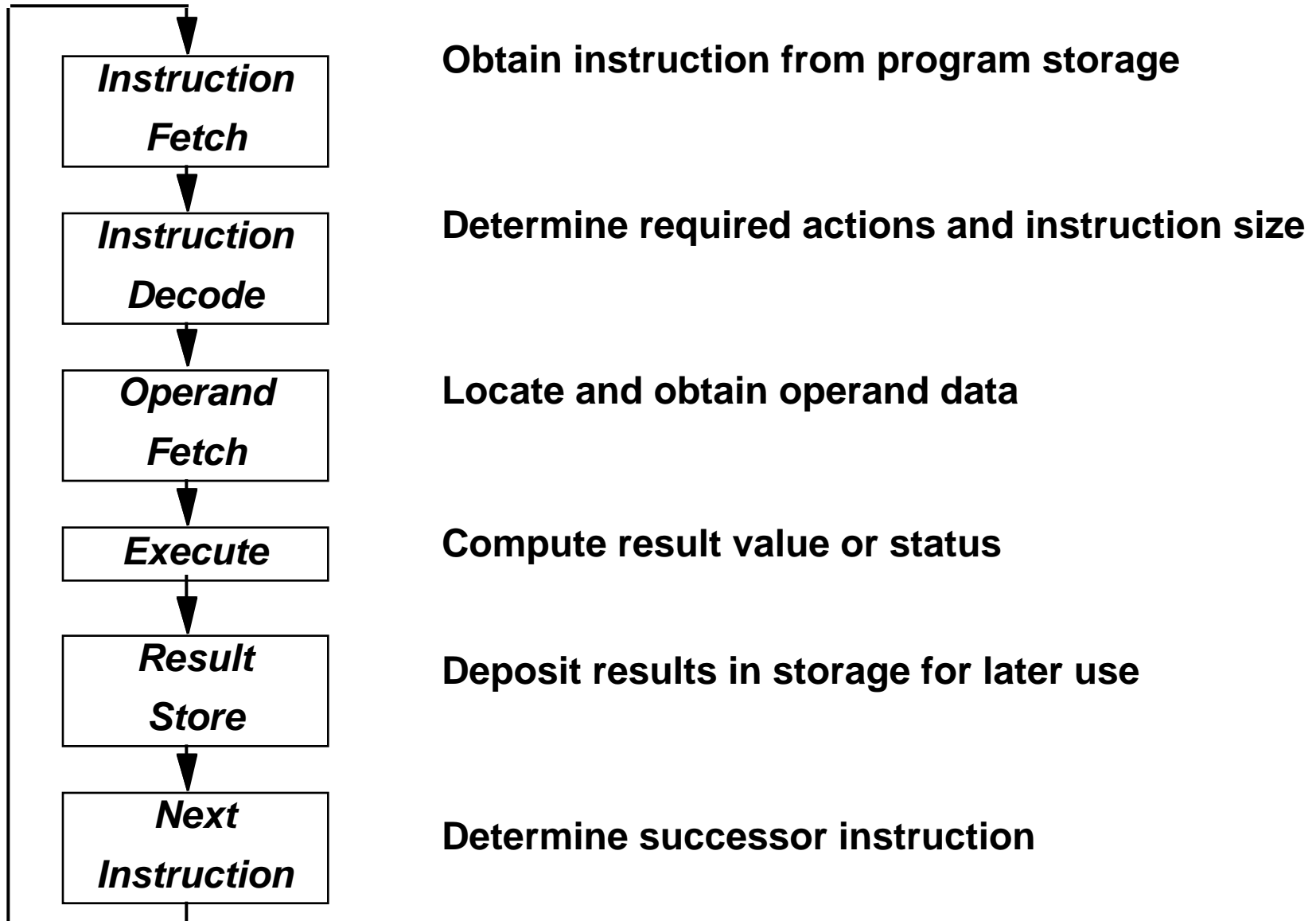


Workstation Design Target:  
25% of cost on Processor  
25% of cost on Memory  
(minimum memory size)  
Rest on I/O devices,  
power supplies, box

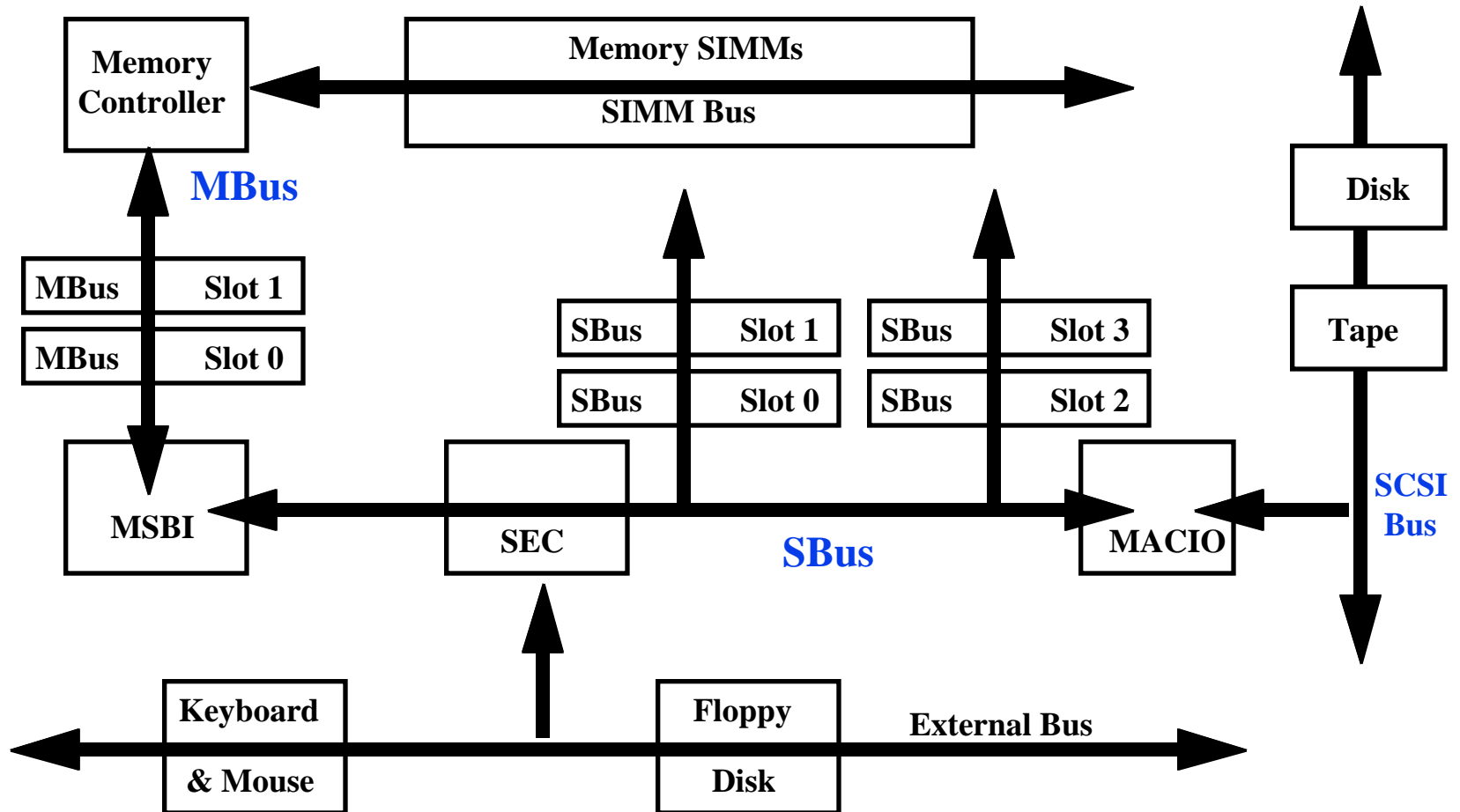
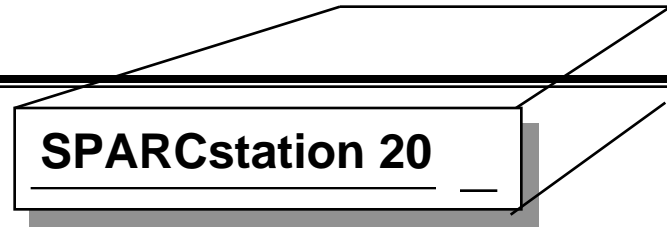


# Execution Cycle

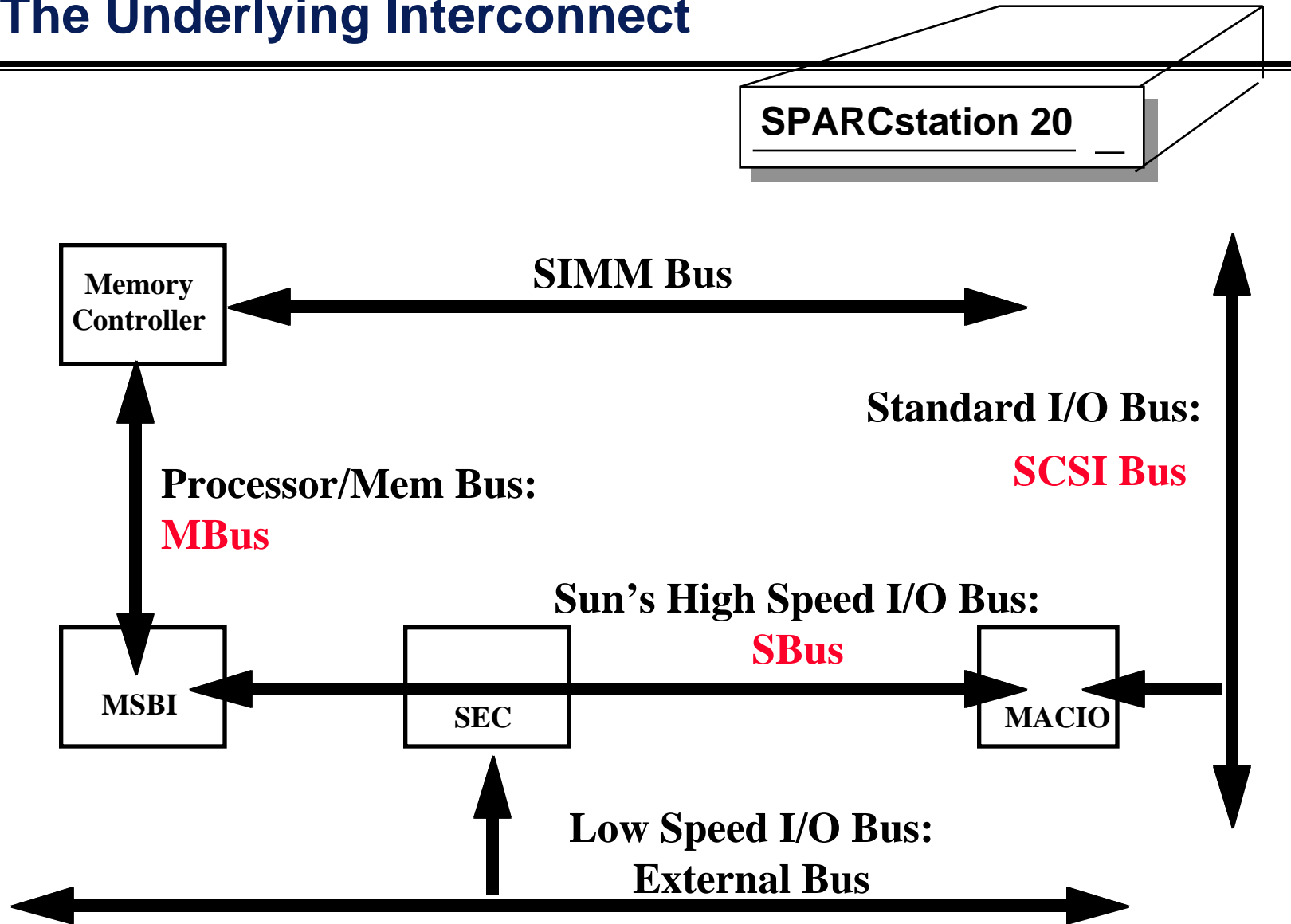
---



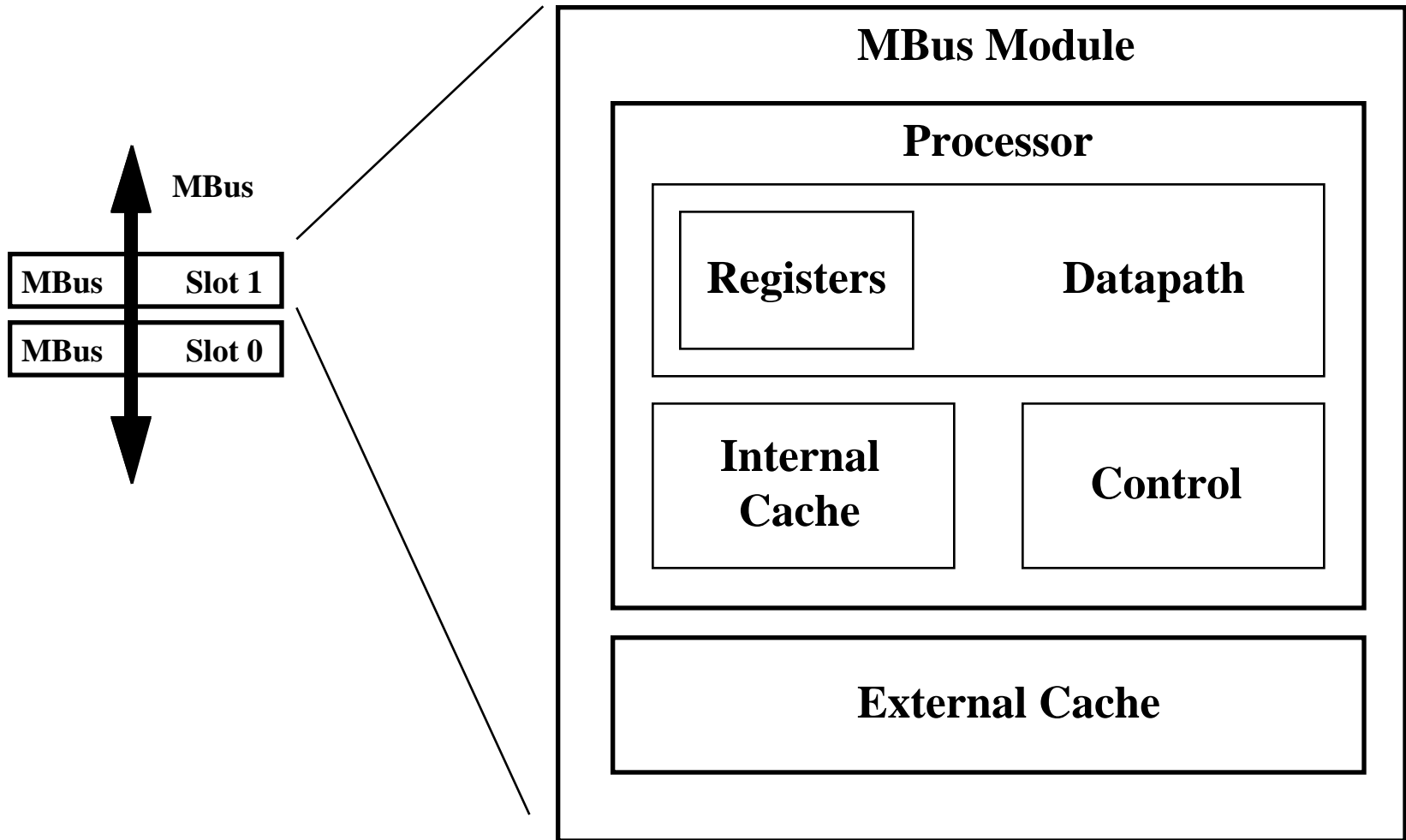
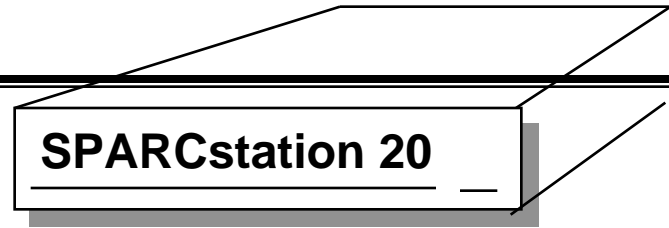
# The SPARCstation 20



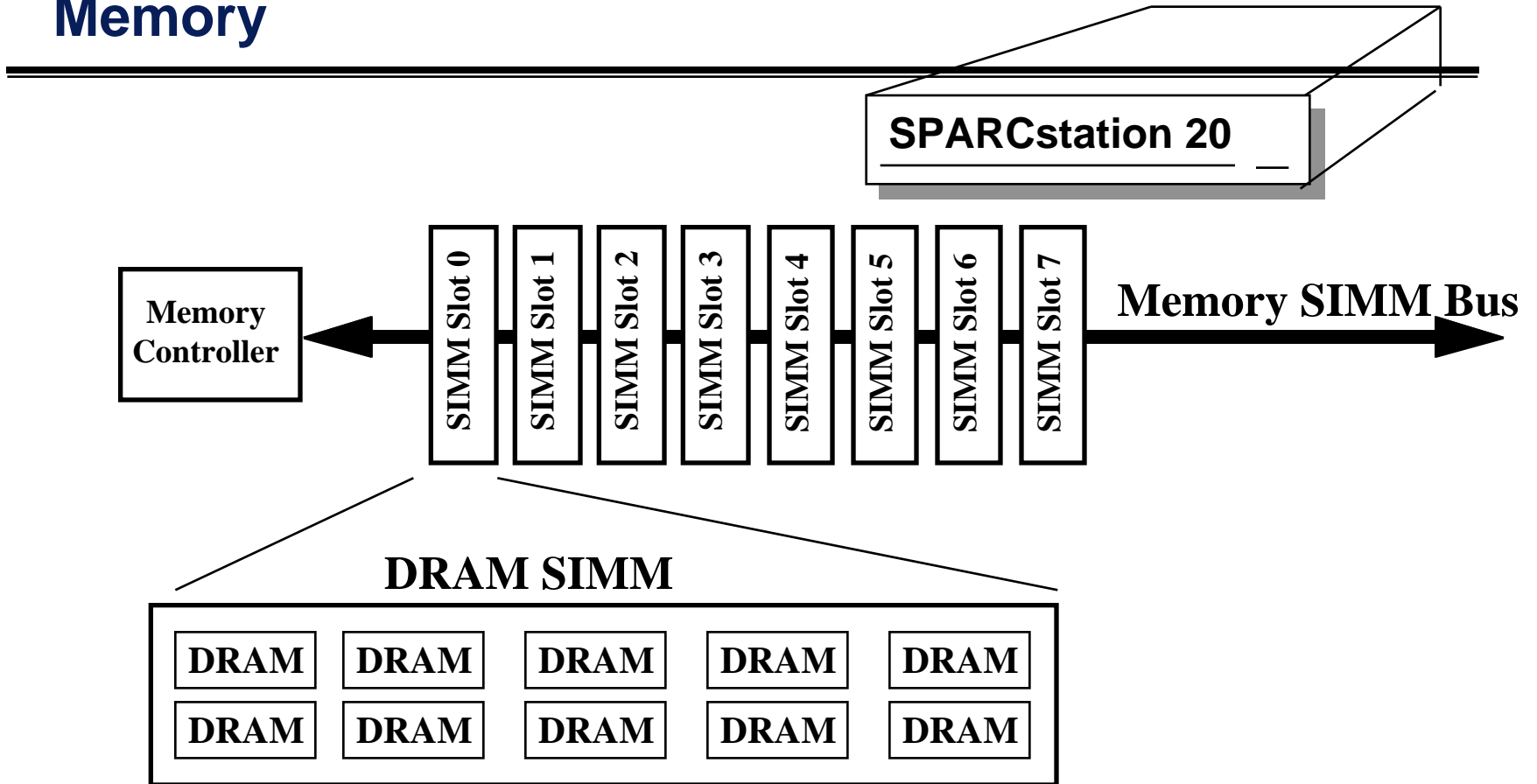
# The Underlying Interconnect



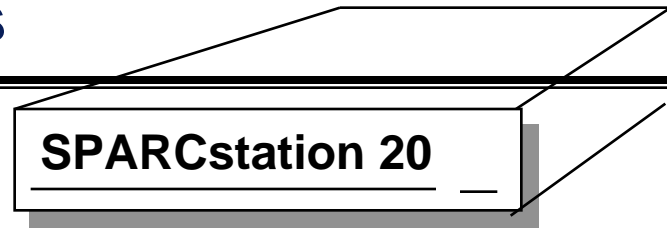
# Processor and Caches



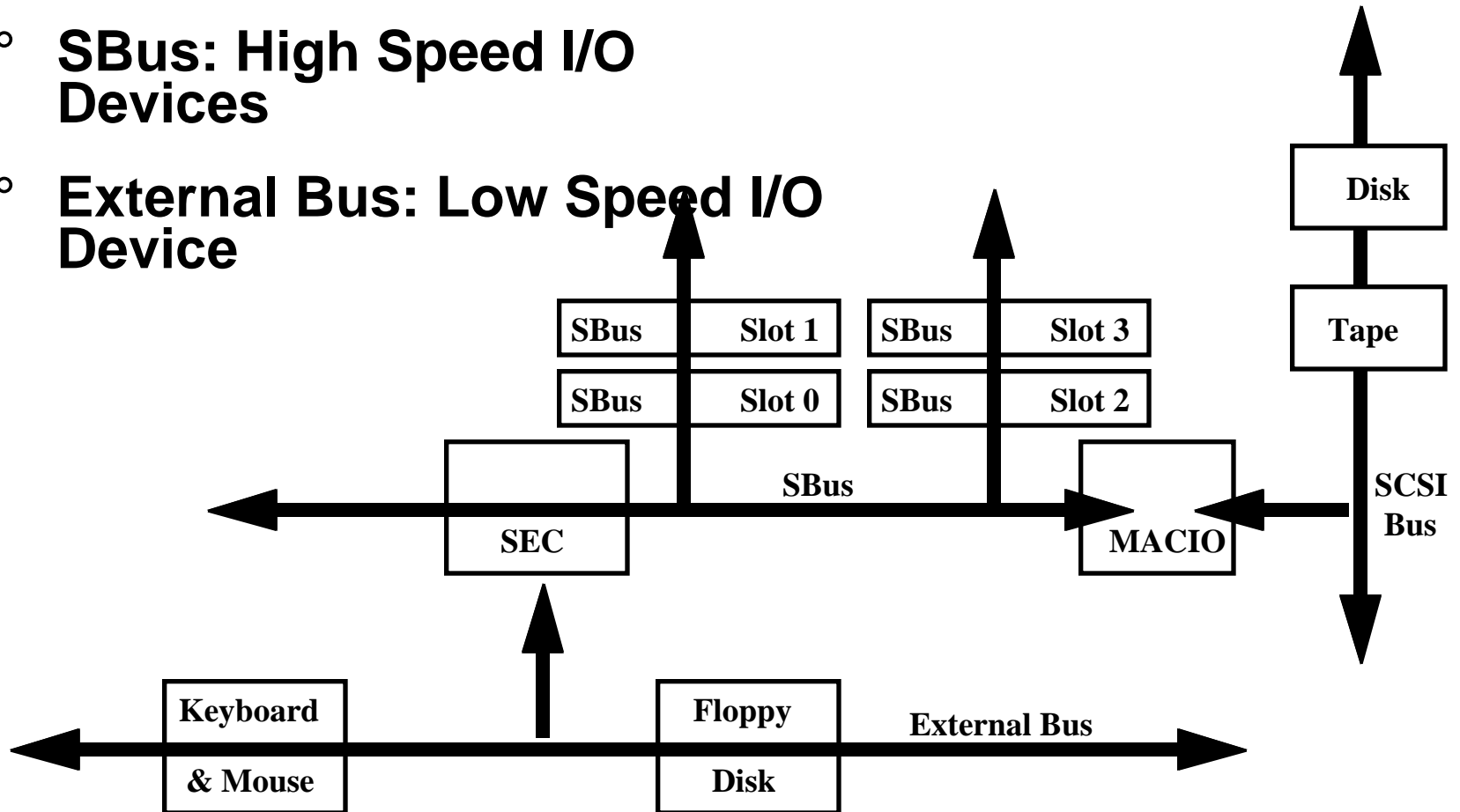
# Memory



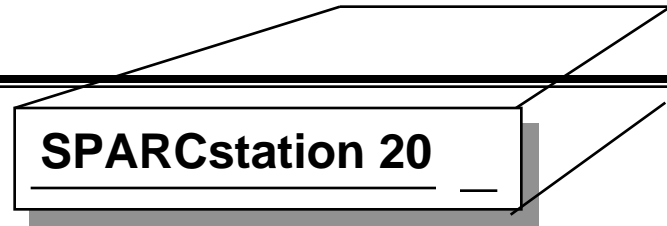
# Input and Output (I/O) Devices



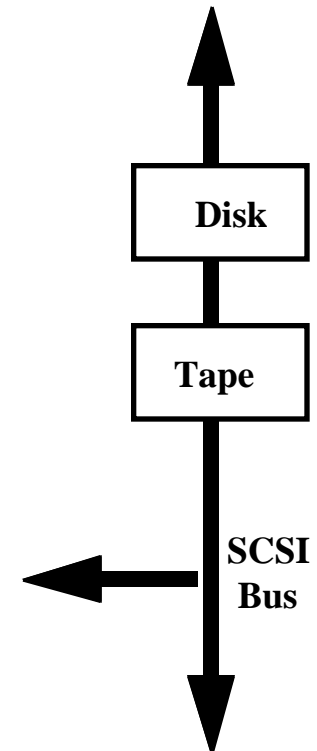
- **SCSI Bus: Standard I/O Devices**
- **SBus: High Speed I/O Devices**
- **External Bus: Low Speed I/O Device**



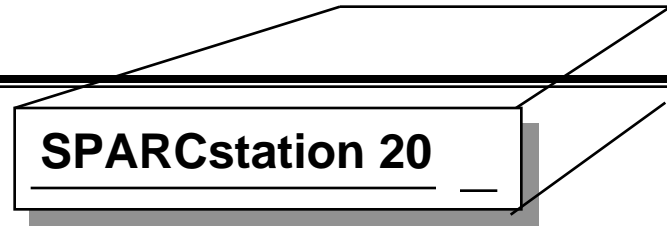
# Standard I/O Devices



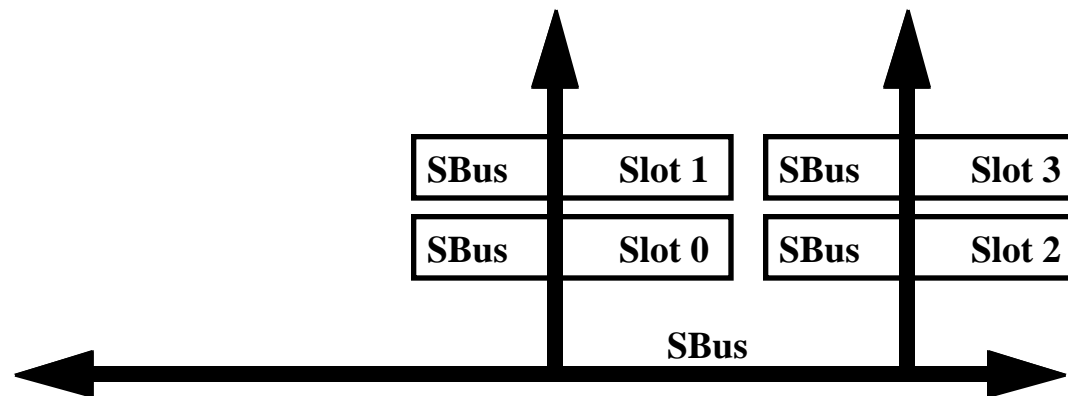
- **SCSI = Small Computer Systems Interface**
- **A standard interface (IBM, Apple, HP, Sun ... etc.)**
- **Computers and I/O devices communicate with each other**
- **The hard disk is one I/O device resides on the SCSI Bus**



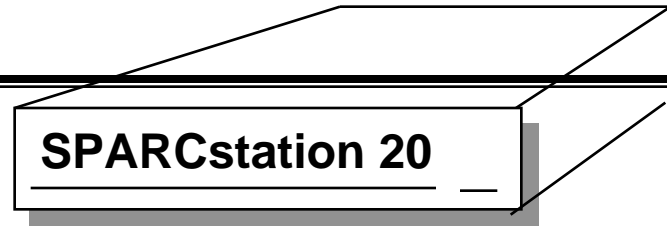
# High Speed I/O Devices



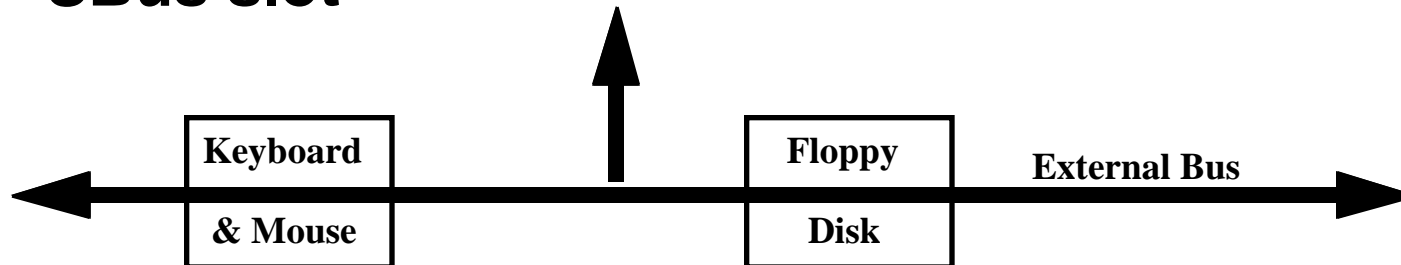
- **SBus is SUN's own high speed I/O bus**
- **SS20 has four SBus slots where we can plug in I/O devices**
- **Example: graphics accelerator, video adaptor, ... etc.**
- **High speed and low speed are relative terms**



# Slow Speed I/O Devices



- There are only four SBus slots in SS20--"seats" are expensive
- The speed of some I/O devices is limited by human reaction time--very very slow by computer standard
- Examples: Keyboard and mouse
- No reason to use up one of the expensive SBus slot



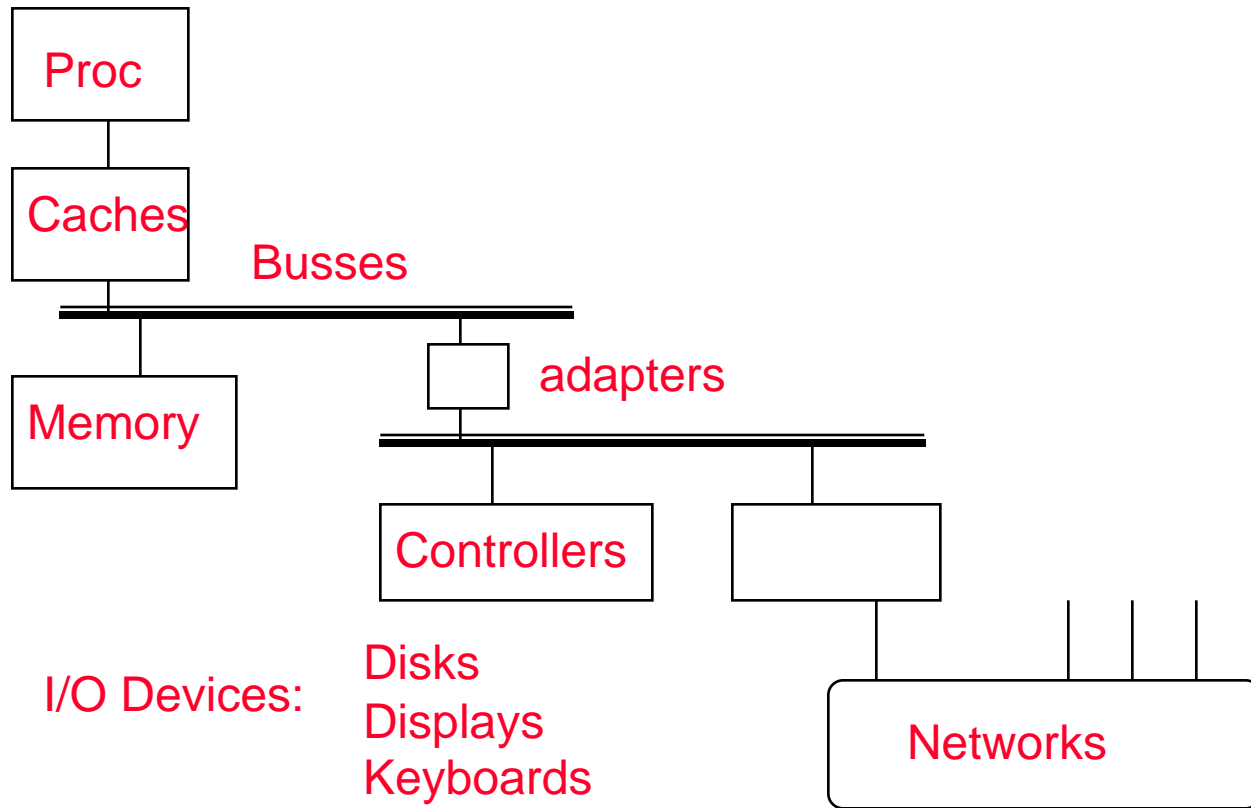
# Summary: Organization

---

- **All computers consist of five components**
  - Processor: (1) datapath and (2) control
  - (3) Memory
  - (4) Input devices and (5) Output devices
- **Not all “memory” are created equally**
  - Cache: fast (expensive) memory are placed closer to the processor
  - Main memory: less expensive memory--we can have more
- **Input and output (I/O) devices have the messiest organization**
  - Wide range of speed: graphics vs. keyboard
  - Wide range of requirements: speed, standard, cost ... etc.
  - Least amount of research (so far)

# Summary: Computer System Components

---



° All have interfaces & organizations