Computer Architecture | CPTR 308

Preliminaries

- Me: Raheel Ahmad
    - Ph.D., Southern Illinois University
    - M.S., University of Southern Mississippi
    - B.S., Zakir Hussain College, India
- Contact: Science 116, rahmad@manchester.edu, 982-5314
    - Tues: 9:00 - 12:00 am; Thu: 10:00 - 12:00 am
- Email me with subject starting with **CPTR308**
- http://users.manchester.edu/Facstaff/RAhmad/classes/308/index.htm
    - Also, Angel’s course webpage has a link to above

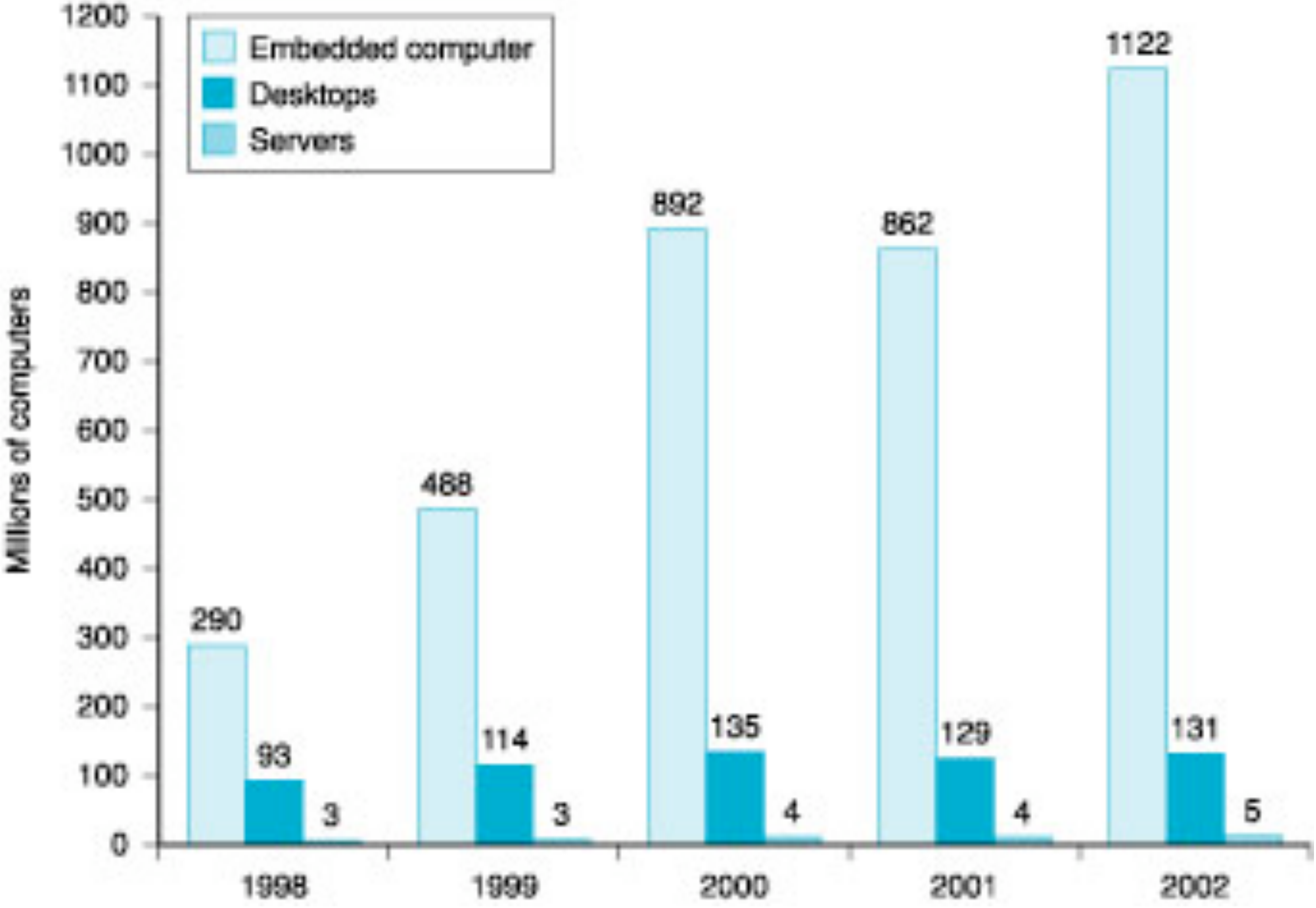
Preliminaries

- Course schedule
  - Science 142, MWF: 11:00 - 11:50 am
- Computer Architecture
  - Crucial
  - Insightful
  - Challenging
- Discuss problems early, often
- Assignments, quizes, tests
- Keep up to date with the deadlines and due dates

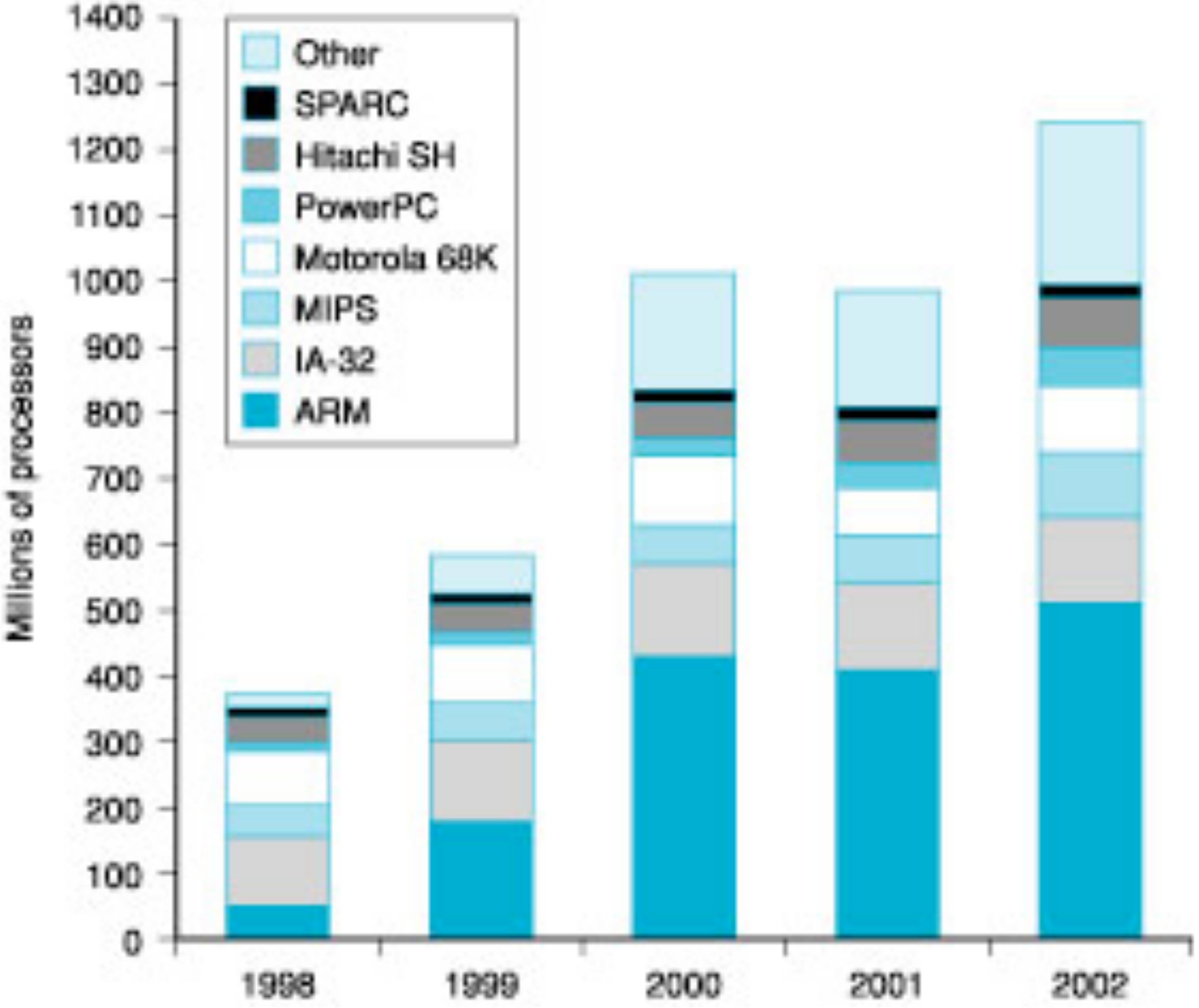
Introduction

- This course is all about how computers work
- But what do we mean by a computer?
  - Different types: desktops, servers, embedded devices
  - Different uses: automobiles, graphics, finance, genomics…
  - Different manufacturers: Intel, Apple, IBM, Microsoft, Sun…
  - Different underlying technologies and different costs!
- Analogy: Consider a course on “automotive vehicles”
  - Many similarities from vehicle to vehicle (e.g., wheels)
  - Huge differences from vehicle to vehicle (e.g., gas vs. electric)
- Best way to learn:
  - Focus on a specific instance and learn how it works
  - While learning general principles and historical perspectives

Number of different types of processors sold



Sale of microprocessors by architecture



Objectives

- Study
  - how do Java/C++ programs execute?
  - the interface between s/w & h/w
  - what determines program performance & how to improve it
  - techniques for improving hardware performance

What is a computer?

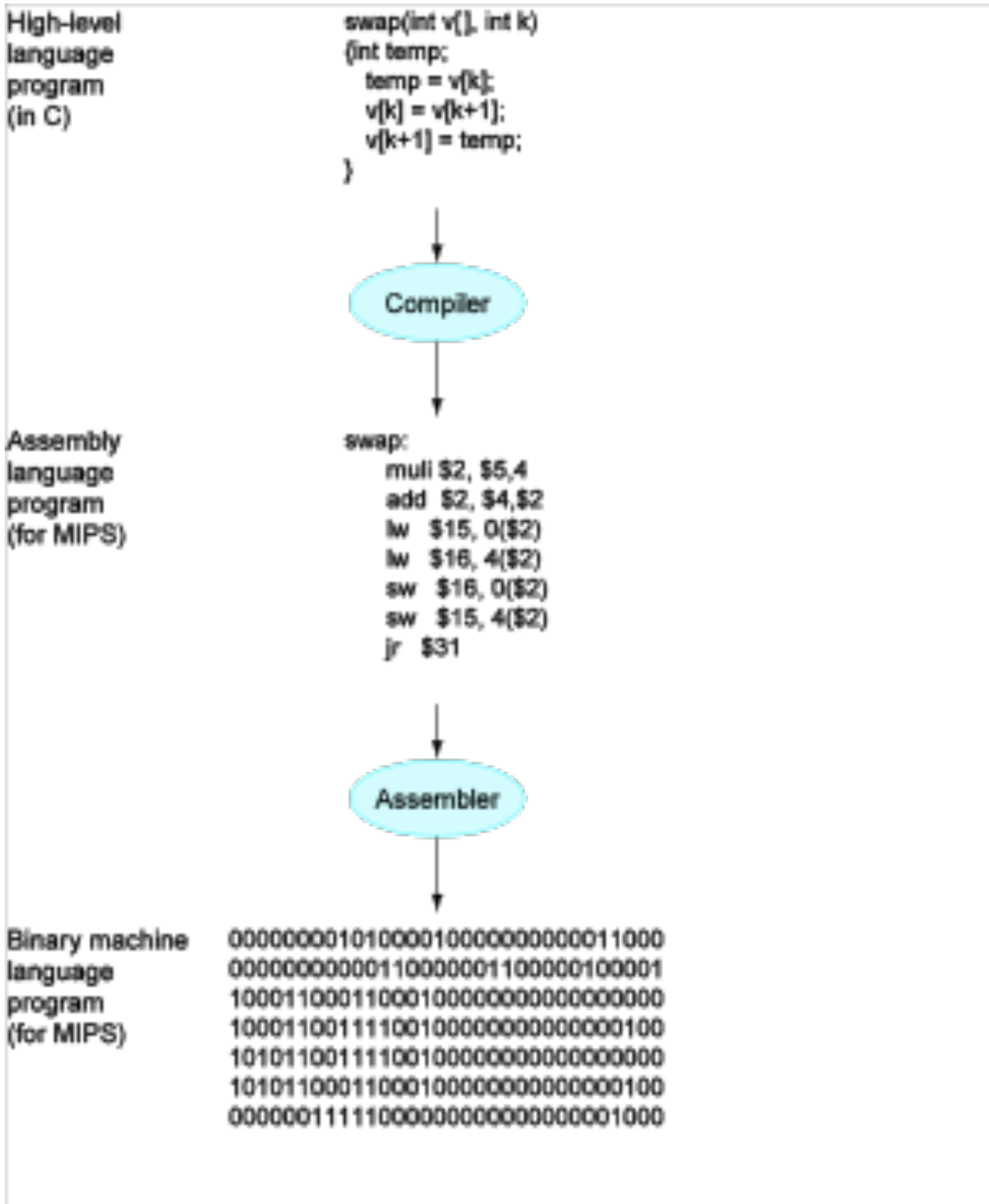
- Components:
  - input (mouse, keyboard)
  - output (display, printer)
  - memory (disk drives, DRAM, SRAM, CD)
  - processor
- Our primary focus: the processor (datapath and control)
  - implemented using millions of transistors
  - Impossible to understand by looking at each transistor
  - We need...

Abstraction

- Delving into the depths reveals more information

- An abstraction omits unneeded detail, helps us cope with complexity

What are some of the details that appear in these familiar abstractions?



High-level language program (in C)

```
swap(int v[], int k)
{int temp;
   temp = v[k];
   v[k] = v[k+1];
   v[k+1] = temp;
}
```

Compiler

Assembly language program (for MIPS)

```
swap:
     muli $2, $5,4
     add  $2, $4,$2
     lw   $15, 0($2)
     lw   $16, 4($2)
     sw   $16, 0($2)
     sw   $15, 4($2)
     jr   $31
```

Assembler

Binary machine language program (for MIPS)

```
00000000101000010000000000011000
00000000000110000001100000100001
10001100011000100000000000000000
10001100111100100000000000000100
10101100111100100000000000000000
10101100011000100000000000000100
00000011111000000000000000001000
```

How do computers work?

- Need to understand abstractions such as:
  - Applications software
  - Systems software
  - Assembly Language
  - Machine Language
  - Architectural Issues: i.e., Caches, Virtual Memory, Pipelining
  - Sequential logic, finite state machines
  - Combinational logic, arithmetic circuits
  - Boolean logic, 1s and 0s
  - Transistors used to build logic gates (CMOS)
  - Semiconductors/Silicon used to build transistors
  - Properties of atoms, electrons, and quantum dynamics
  - So much to learn!