CMPE 212

Principles of Digital Design

Lecture 1

Introduction and Overview

January 25, 2016

www.csee.umbc.edu/~younis/CMPE212/CMPE212.htm

Lecture's Overview

- Course resources, syllabus and work load
- Grade structure and policy
- Teaching style and philosophy
- Course overview.
- Von Neumann and System Bus models.
- Different levels of abstractions (programmer's view, operating system view, hardware designer view, etc.)
- Some historical perspective
- Digital versus analog systems

Course Resources

Instructor: Dr. Mohamed Younis

Office: ITE 318 E-mail: younis@cs.umbc.edu

Office hours: Monday and Wednesday 10:30 -- 11:30 AM

Research interest:

Wireless networks, Fault tolerant computing, Security, Real-time systems, Underwater communication, Tool support for embedded systems

TA: Mr. Ahmed Shahin

Office: ITE 242 E-mail: ashahin1@umbc.edu

Office hours: Tuesday (1:00-2:00 PM) and Friday (4:00 – 5:00 PM)

Textbooks:

- **Digital Logic Circuit Analysis and Design, 1/e,** 1995
  V. P. Nelson, H. T. Nagle, B. D. Carroll, and D. Irwin, *Prentice Hall,*

Web page: www.csee.umbc.edu/~younis/CMPE212/CMPE212.htm

Instructor will stay after class to answer questions

Course Syllabus

1. *Number Systems and Codes*

- Introduction to number systems
- Number systems conversion
- Representation of binary numbers
- Arithmetic and computer codes

2. *Algebraic Analysis and Synthesis of Logic Circuits*

- Fundamentals of Boolean Algebra
- Switching functions and circuits
- Analysis and synthesis of combinational circuits
- Computer-aided design of logic circuits

3. *Simplification of Switching Circuits*

- Characteristics of minimization methods
- Methods: Karnaugh maps, Quine-McCluskey and Patrick's algorithms
- Dealing with timing hazards
- Computer-aided minimization of switching functions

Course Syllabus

4. Modular Combination Circuits

- Decoders, encoders, multiplexers, De-multiplexers comparators
- Binary arithmetic elements

5. Combinational Circuit Design with Programmable devices

- Logic-array circuits
- Field-programmable logic arrays
- Programmable read-only memory

6. Introduction to Sequential Devices

- Memory devices (Flip-flops, latches, etc.)
- Modules: Shift registers, counters, etc.

7. Analysis, Synthesis & Simplification of Sequential Circuits

- State machine representation
- State transition table
- Synchronous sequential Circuit synthesis

8. Asynchronous Sequential Circuits (time permitting)

Course Workload

❑ Assignments

- Five assignments will be given contributing 15% to the final grade (3% each)
- An average assignment requires about 2-3 hours to perform
- Assignments are due in class on the due date (not later)

❑ Lab/Project

- Participating and performing lab work contributes 5% towards the final grade
  ➔ attendance is very important
- Lab work involves hardware design and implementation
- Two design projects will be assigned contributing 5% and 10%, respectively

❑ Exams

- Two midterm exams are planned; each contributes 20% to the final grade
- The midterms are not accumulative and are scheduled on March 9th and April 18th during the regular class time
- The final exam is scheduled on May 18th at 1:00 pm - 3:00 pm
- Final Exam is comprehensive covering all what is taught in the class

Grade structure and policy

	Grade distribution
Final Exam	25%
Midterm Exams	40%
Lab/projects	20%
Homework Assignments	15%

Course grade	Range
A	90% - 100%
B	80% - 89.9%
C	70% - 79.9%
D	60% - 69.9%

- Assignments/Projects are due in class (*Late assignments are not accepted*)
- UMBC rules apply to cheating/copying
  - You may work together and discuss homework and the project.
  - You must do your own work and not copy from anyone else
  - You better off skipping an assignment or get a partial credit
- Copying/cheating will result in a minimum punishment of a zero grade for the assignment or project

Teaching Style and Philosophy

❑ *Instructor's role*

- Facilitate and guide the students to the fundamental concepts
- Make it simple and elaborate with examples
- Relate as much as possible to the big picture and other courses
- Prepare class notes to be as rich and comprehensive as possible

❑ *Student's role*

- Focus on understanding and digesting the concept
- Do not worry about the grade more than concepts, soon will be a professional
- Slow down the instructor if you do not understand and raise questions

❑ *TA's role*

- Help students with questions related to their assignments
- Conduct lab sessions and supervise students during their experiments
- Resolve computer and tool issues related to the lab assignment and projects
- Grade the assignments and projects

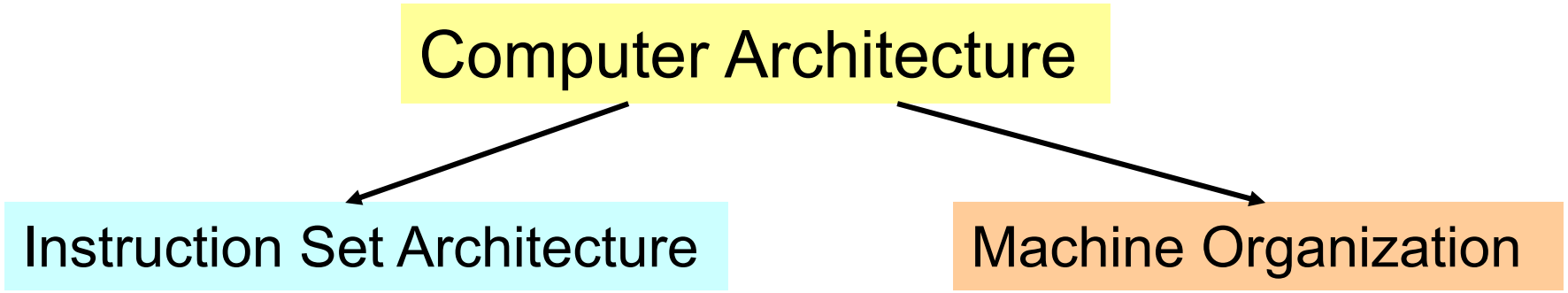
Exams will question the level of understanding of fundamental concepts

Introduction & Motivation

- Computer systems are responsible of 5-10% of the gross national product of the US
- Has the transportation industry kept pace with the computer industry, a coast to coast trip would take 5 seconds and cost 50 cents
- WWW, ATM, DNA mapping, … are among the applications that were economically infeasible and became practical
- Cashless society, anywhere computing, automated intelligent highways… are next computer science fiction on their way to become a reality
- Advanced computer design has been at the core of such technological development and is still on a forward move

What is "Computer Architecture"?

- Instruction set architecture deals with the functional behavior of a computer system as viewed by a programmer (like the size of a data type – 32 bits to an integer).
- Computer organization deals with structural relationships that are not visible to the programmer (like clock frequency or the size of the physical memory).
- The Von Neumann model is the most famous computer organization



- Interfaces
- Compiler/System View
- "Building Architect"

- Hardware Components
- Logic Designer's View
- "Construction Engineer"

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

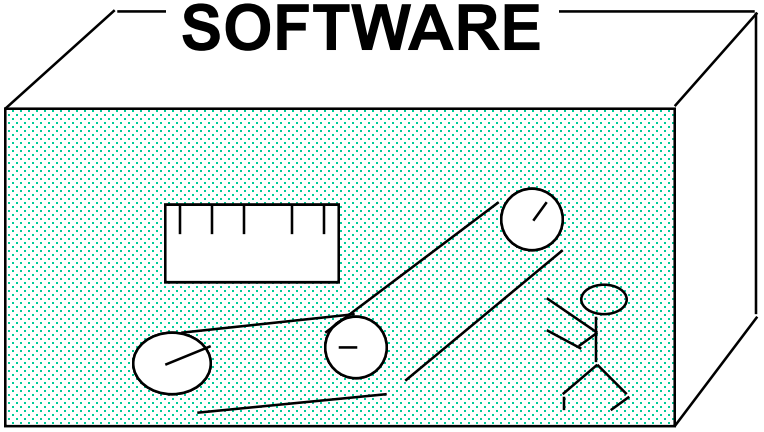
-- Data Types & Data Structures: Encoding & Representation

-- Instruction Set

-- Instruction Formats

-- Modes of Addressing and Accessing Data Items and Instructions

-- Exceptional Conditions



The instruction set architecture distinguishes the semantics of the architecture from its detailed hardware implementation

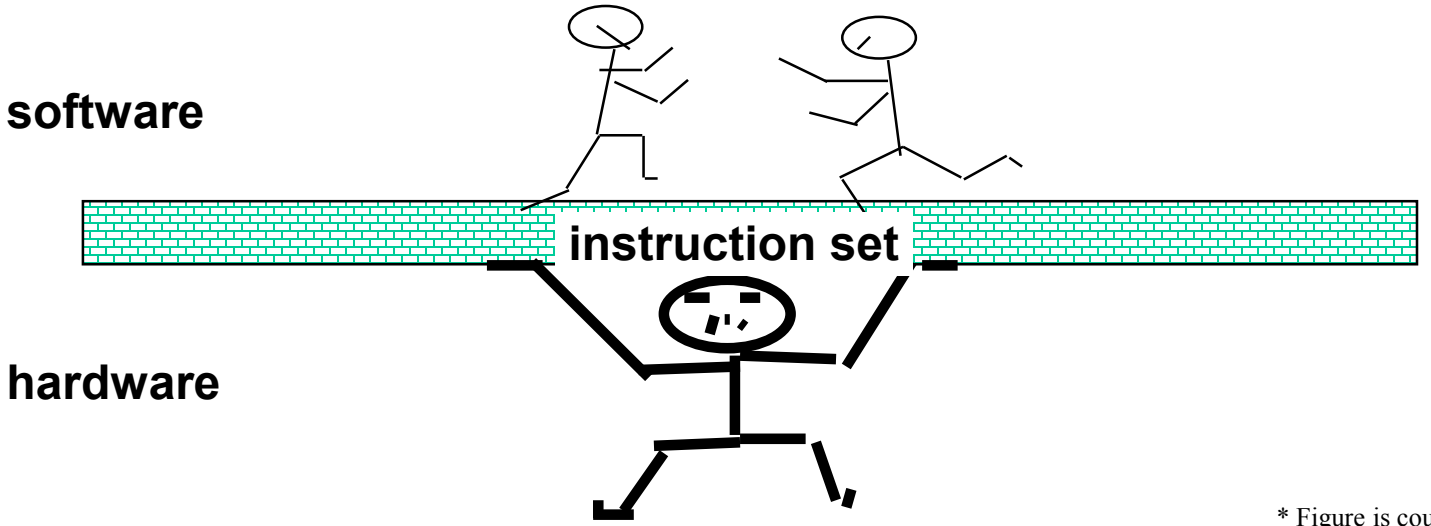
* Slide is courtesy of Dave Patterson

The Instruction Set: a Critical Interface

Examples:

- DEC Alpha (v1, v3) 1992-1997
- HP PA-RISC (v1.1, v2.0) 1986-1996
- Sun Sparc (v8, v9) 1987-1995
- SGI MIPS (MIPS I, II, III, IV, V) 1986-1996
- Intel (8086,80286,80386, 80486,Pentium, MMX, ...) 1978-2000

The instruction set can be viewed as an abstraction of the H/W that hides the details and the complexity of the H/W



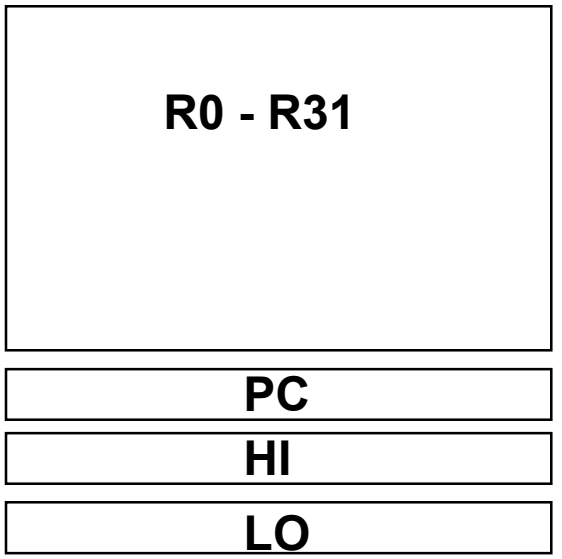
* Figure is courtesy of Dave Patterson

MIPS R3000 Instr. Set Arch. (Summary)

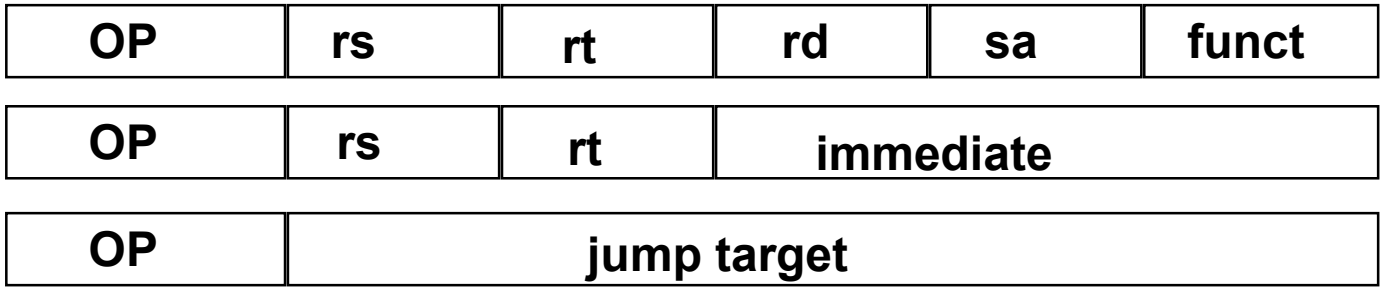
Instruction Categories

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

Registers

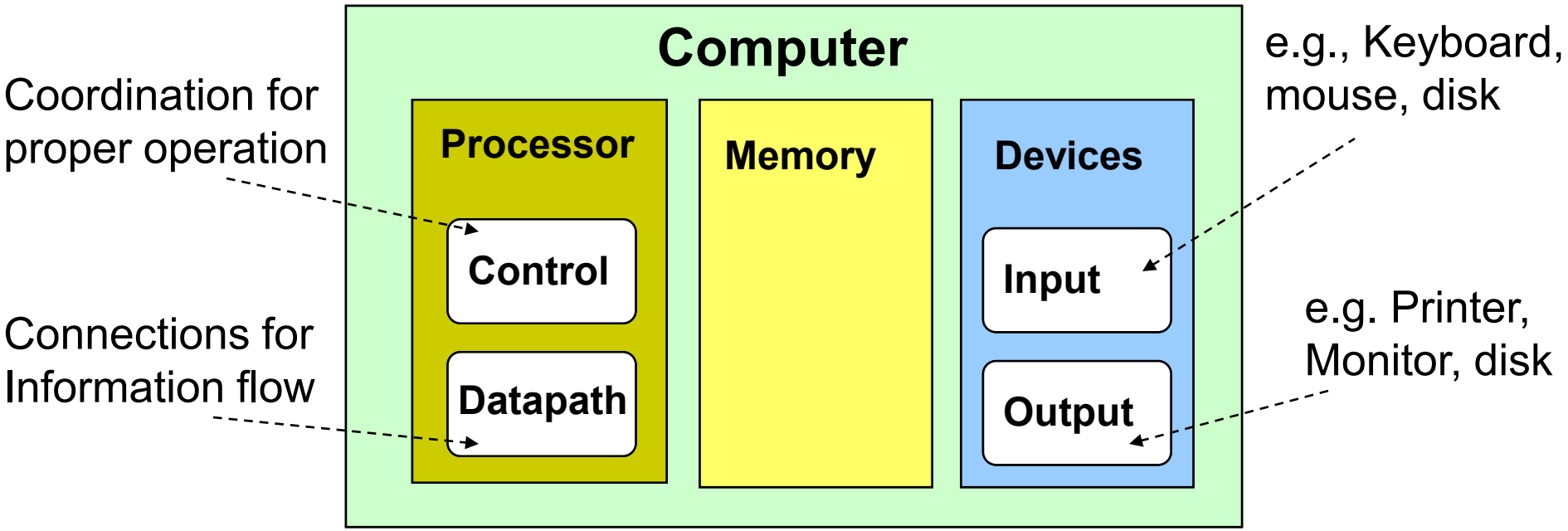


3 Instruction Formats: all 32 bits wide



* Slide is courtesy of Dave Patterson

General Computer Organization



- Every piece of every computer, past and present, can be placed into input, output, memory, datapath and control
- The design approach is constrained by the cost and size and capabilities required from every component
- An example design target can be 25% of the cost for Processor, 25% of the cost for minimum memory size, leaving the remaining budget for I/O devices, power supplies, and chassis

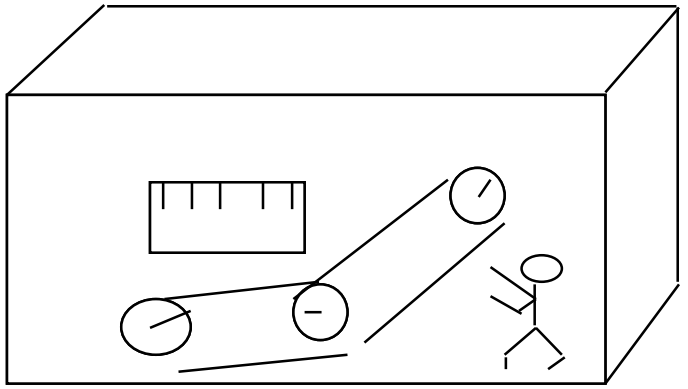
Machine Organization

- 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 functional units to realize the instruction set architecture
- Register Transfer Level Description

Logic Designer's View

ISA Level

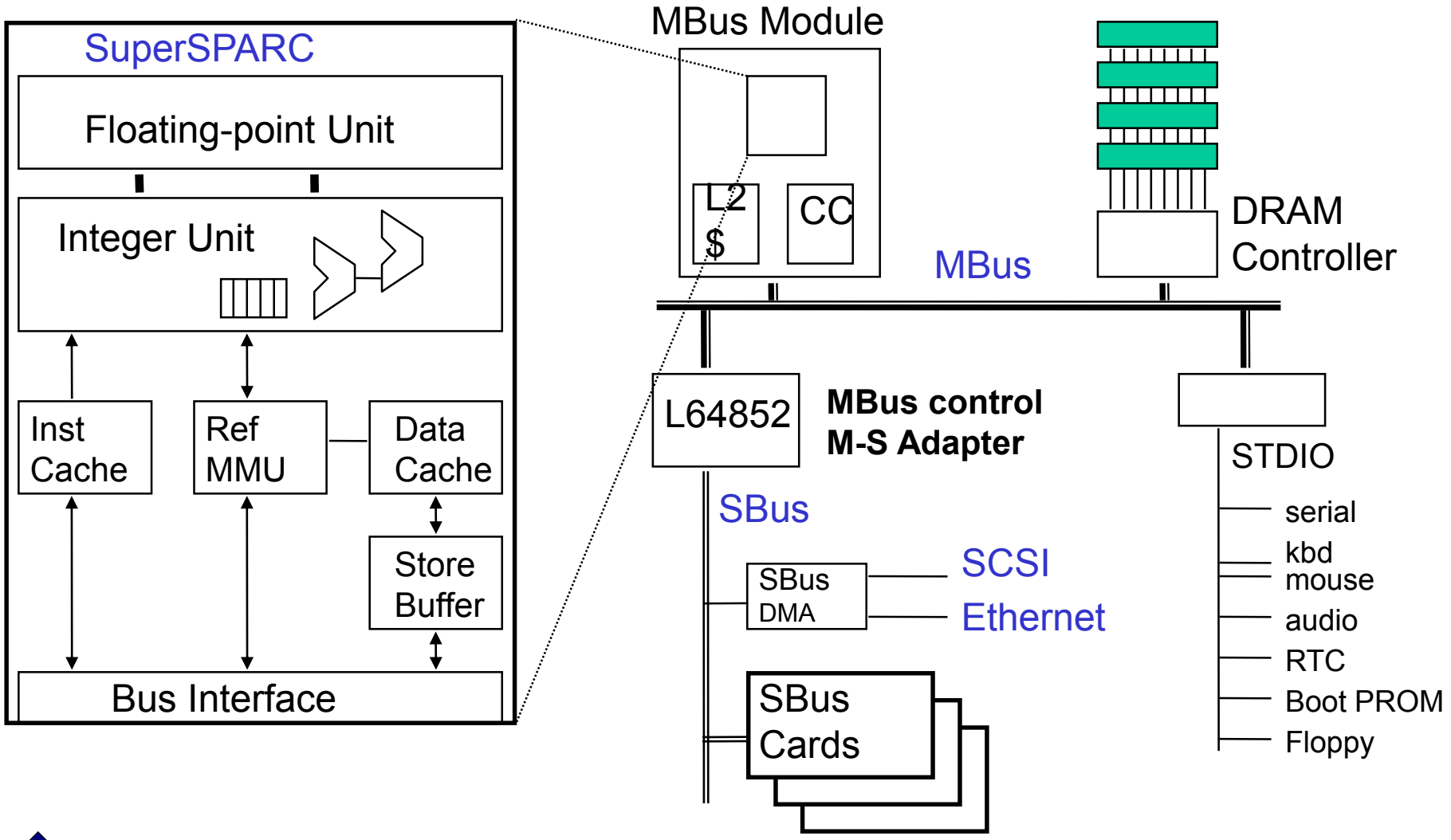
Functional Units & Interconnect



* Slide is courtesy of Dave Patterson

Example Organization

- TI SuperSPARC^{tm} TMS390Z50 in Sun SPARCstation20



* Slide is courtesy of Dave Patterson

The von Neumann Model

- **The Von Neumann model consists of five major components:**

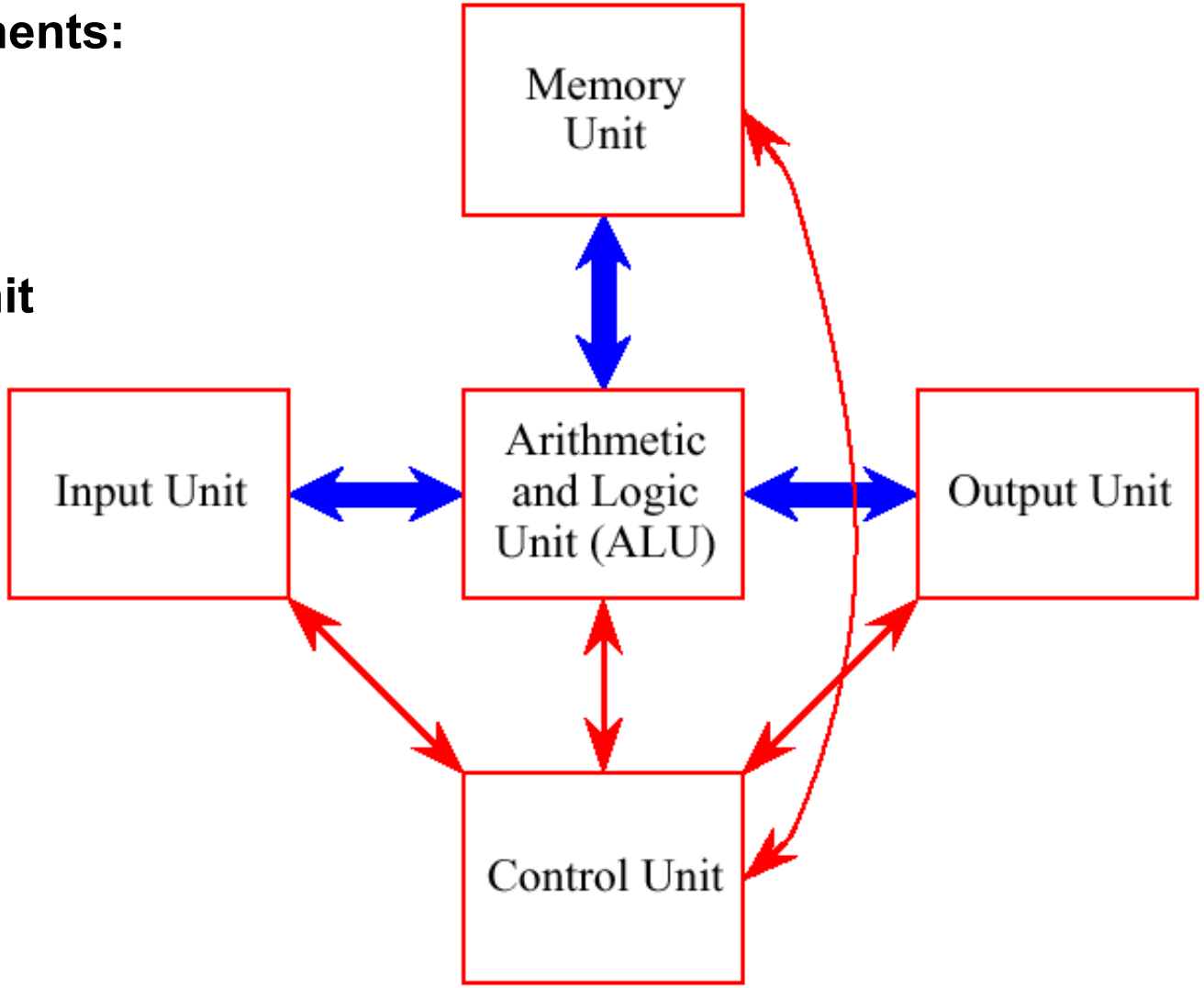
(1) Input unit

(2) Output unit

(3) Arithmetic logic unit

(4) Memory unit

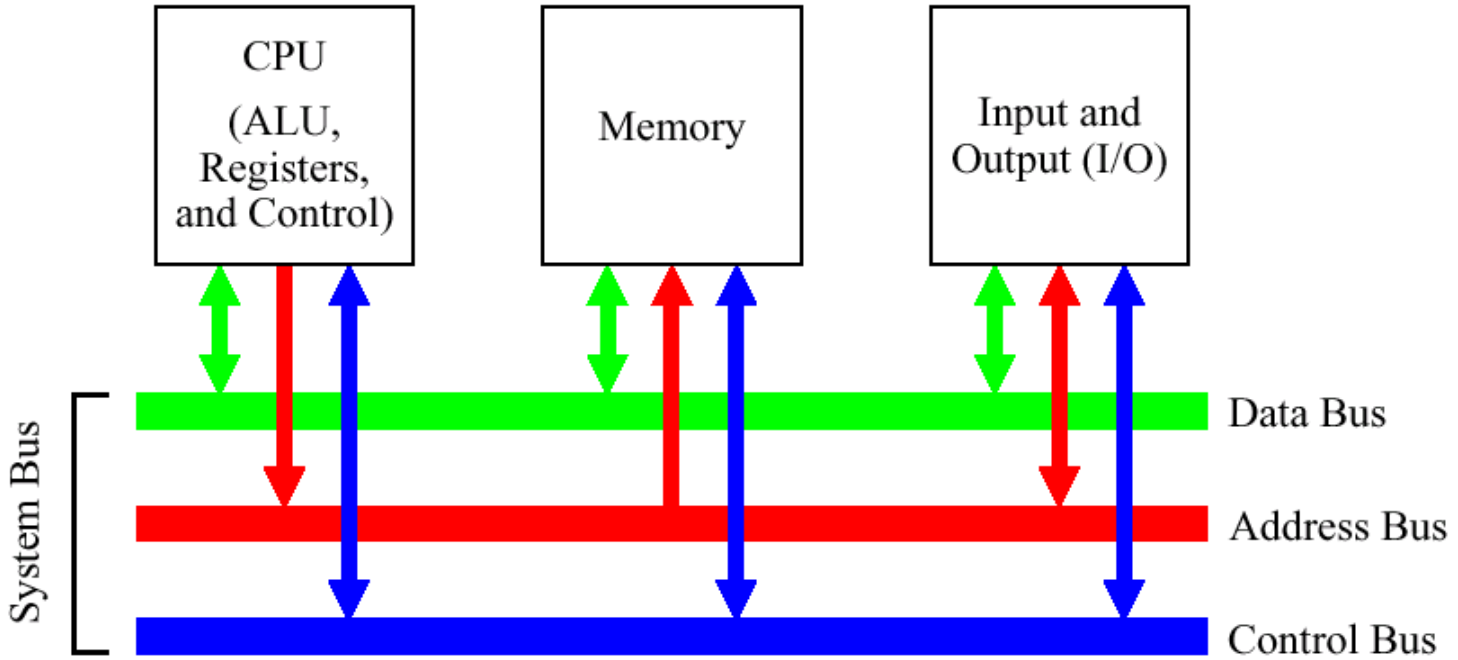
(5) Control unit



* Slide is courtesy of M. Murdocca and V. Heuring

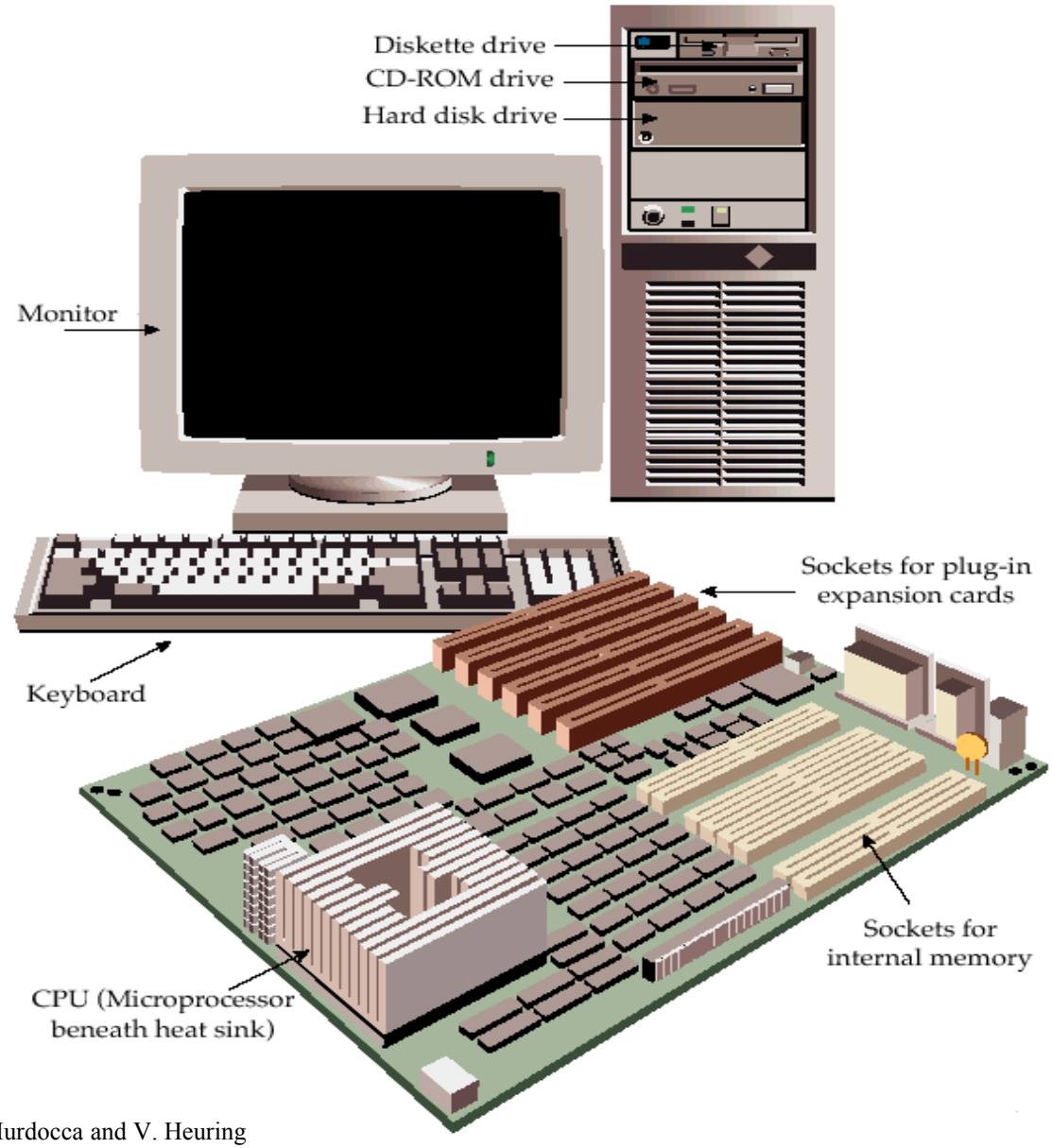
The System Bus Model

- A refinement of the von Neumann model, the system bus model has a CPU (ALU and control), memory, and an input/output unit.
- Communication among components is handled by a shared path-way called the system bus, which is made up of the data bus, the address bus, and the control bus. There is also a power bus, and some architectures may also have a separate I/O bus.



* Slide is courtesy of M. Murdocca and V. Heuring

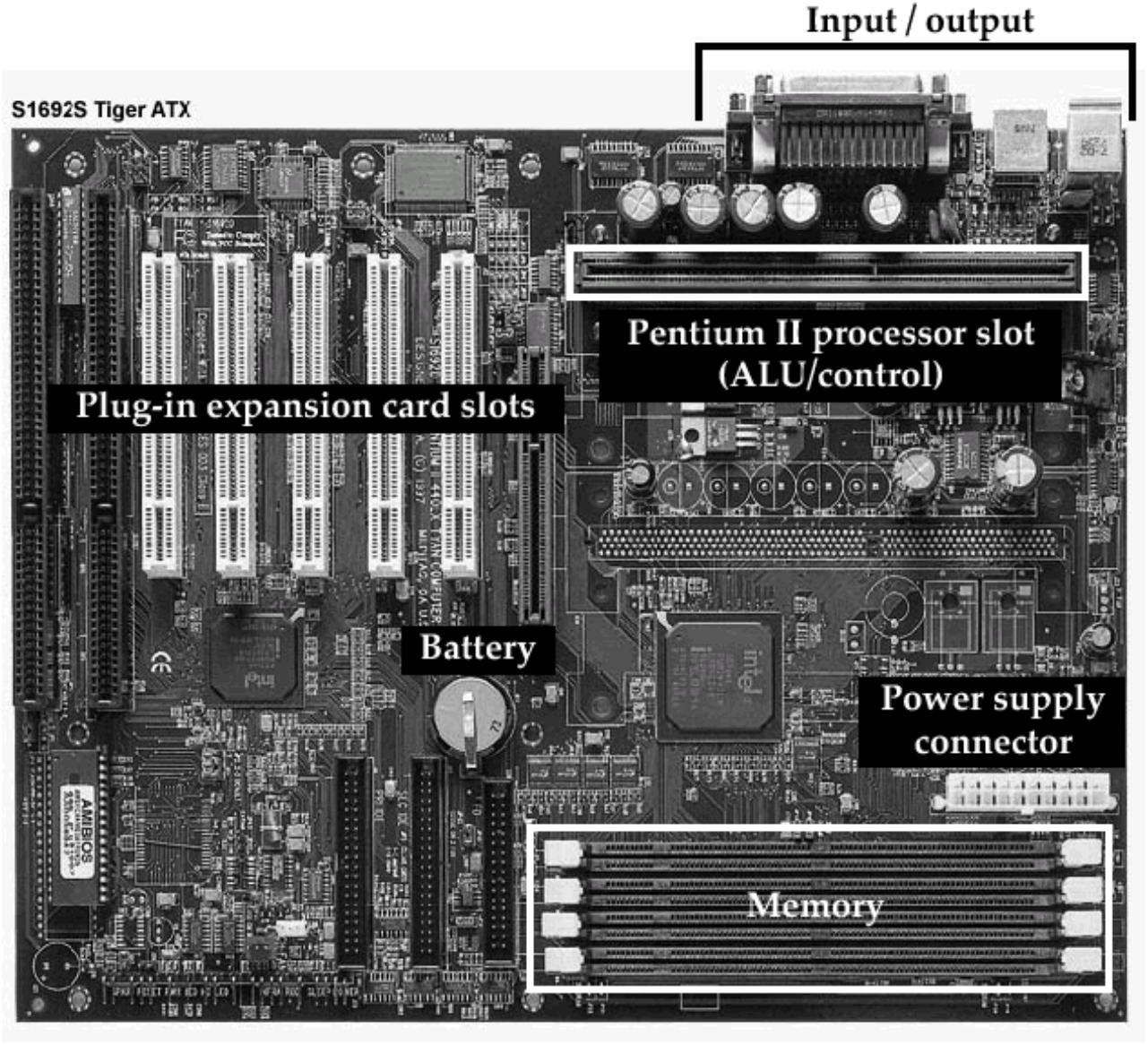
A Typical Computer System



* Slide is courtesy of M. Murdocca and V. Heuring

An Example: Motherboard

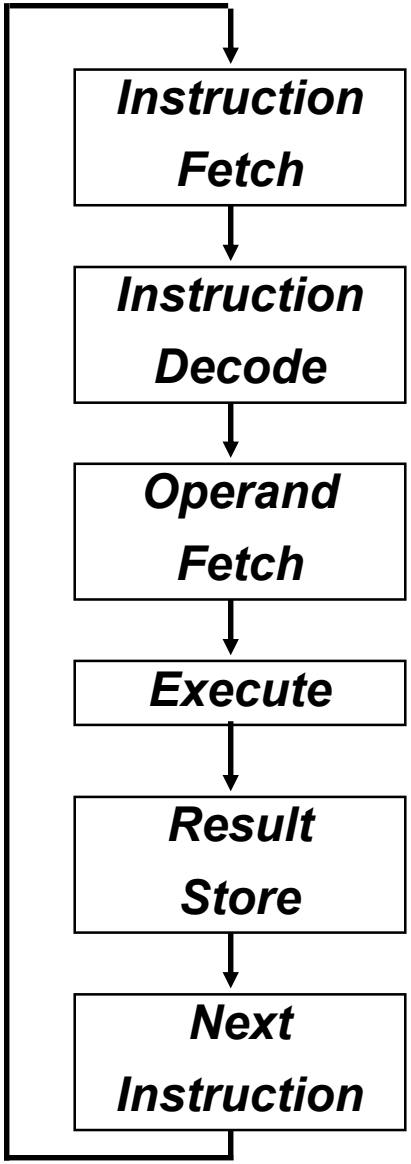
The five von Neumann components are visible in this example motherboard, in the context of the system bus model.



* Slide is courtesy of M. Murdocca and V. Heuring

(Source: TYAN Computer, http://www.tyan.com)

Execution Cycle (Von Neumann Model)



Obtain instruction from program storage

Determine required actions and instruction size

Locate and obtain operand data

Compute result value or status

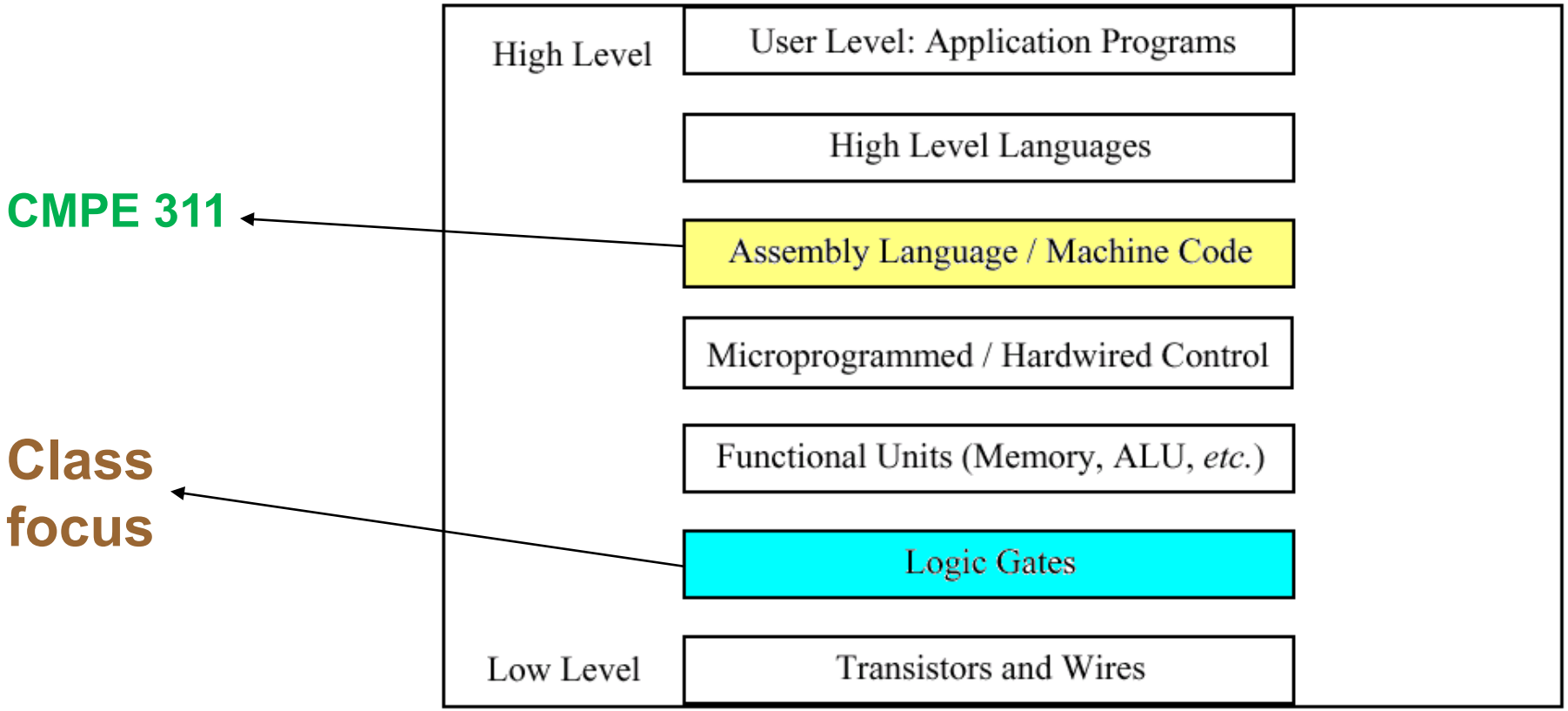
Deposit results in storage for later use

Determine successor instruction

* Slide is courtesy of Dave Patterson

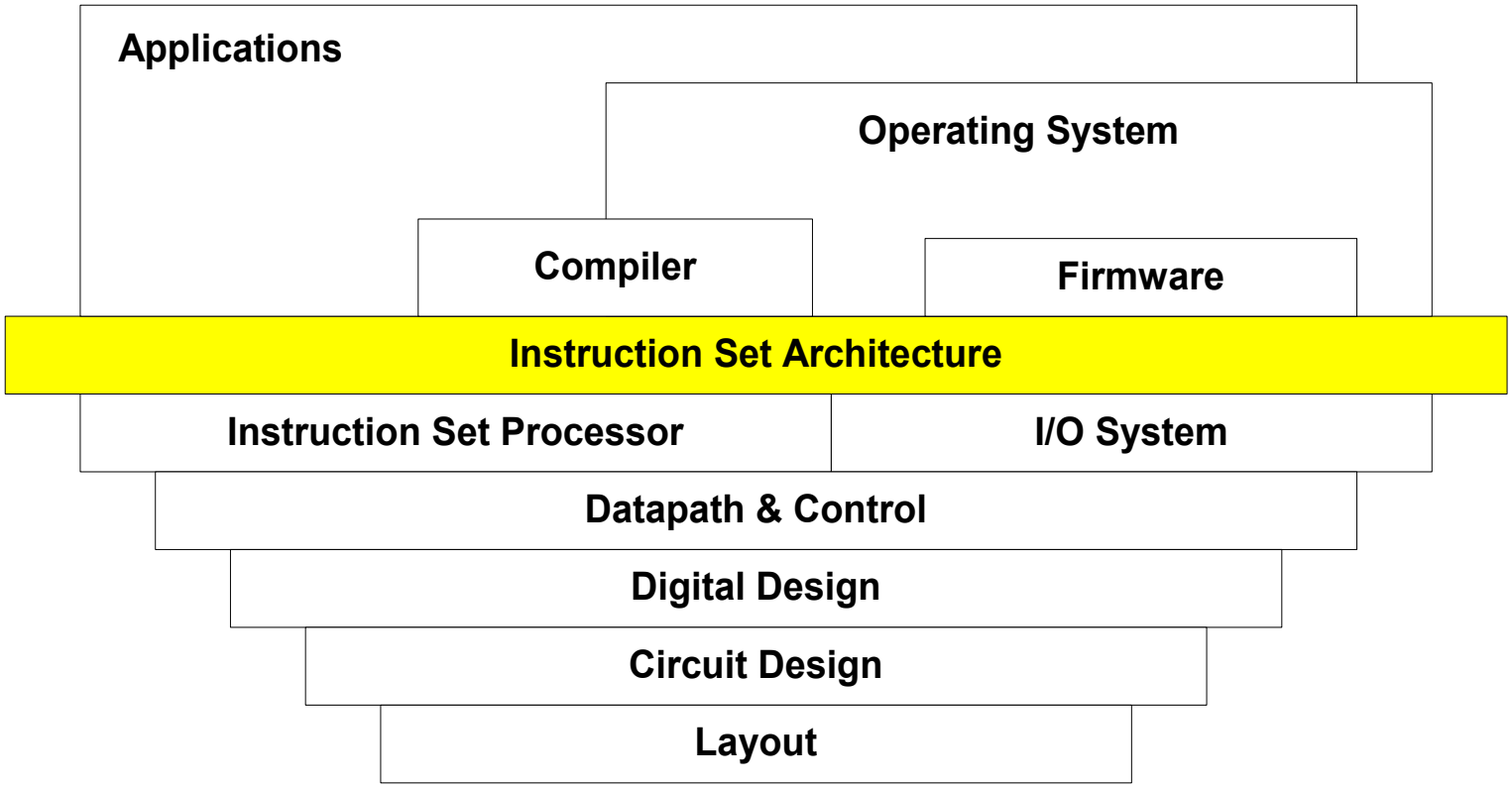
Levels of Machines

- There are a number of levels in a computer (the exact number is open to debate), from the user level down to the transistor level.
- Progressing from the top level downward, the levels become less abstract as more of the internal structure of the computer becomes visible.



* Slide is courtesy of M. Murdocca and V. Heuring

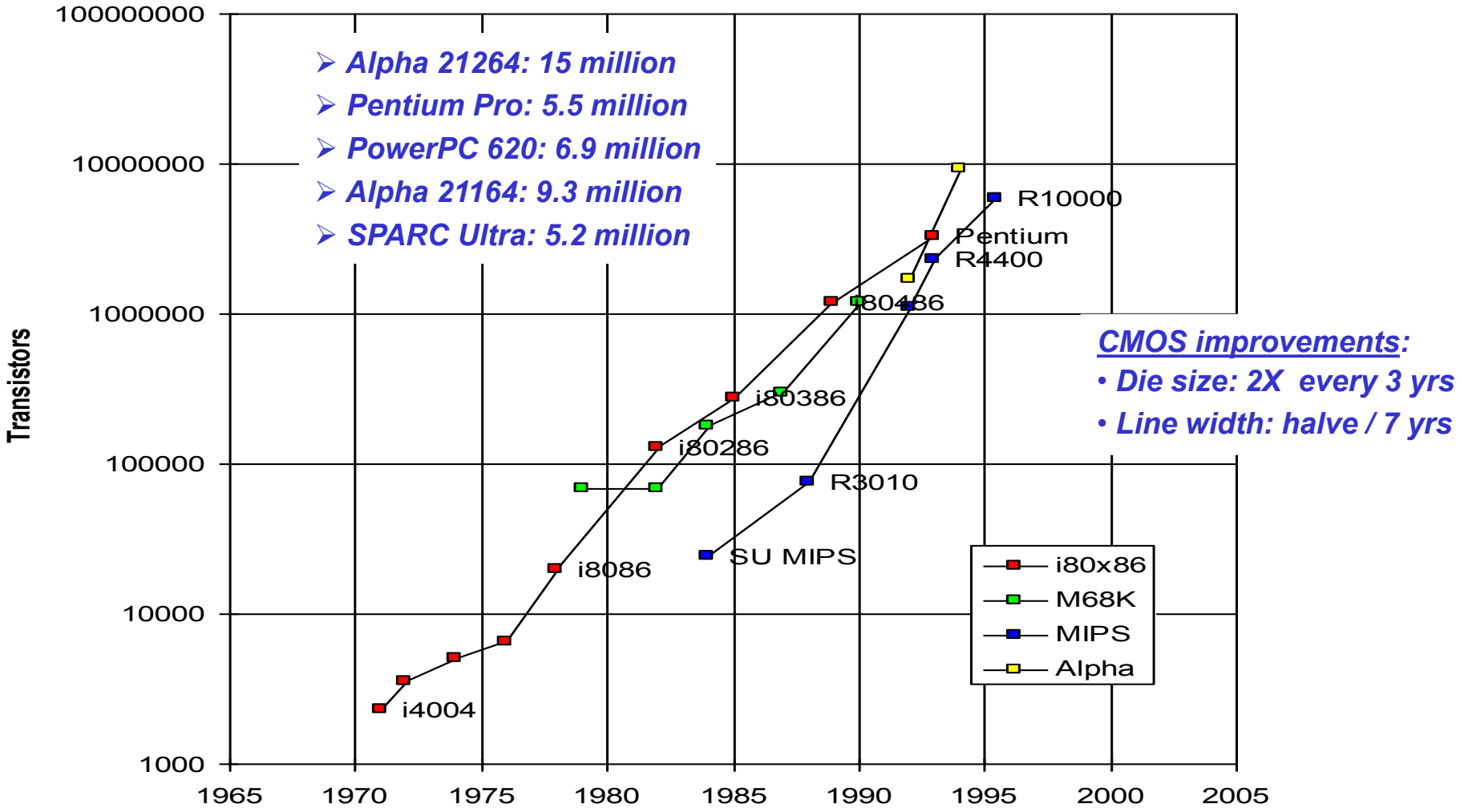
Levels of Abstraction



- S/W and H/W consists of hierarchical layers of abstraction, each hides details of lower layers from the above layer
- The instruction set arch. abstracts the H/W and S/W interface and allows many implementation of varying cost and performance to run the same S/W

* Figure is courtesy of Dave Patterson

Technology Impact on Processors



- In ~1985 the single-chip processor and the single-board computer emerged
- In the 2004+ timeframe, old mainframes have become single-chip computers

* Figure is courtesy of Dave Patterson

Computer Generations

- Computers were classified into 4 generations based on revolutions in the technology used in the development
- By convention, electronic computers are considered as the first generation rather than the electromechanical machines that preceded them
- Today computer generations are not commonly referred to due to the long standing of the VLSI technology and the lack of revolutionary technology in sight

Generations	Dates	Technology	Principal new product
1	1950-1959	Vacuum tube	Commercial electronic computer
2	1960-1968	Transistor	Cheaper computers
3	1969-1977	Integrated circuits	Minicomputer
4	1978- ?	LSI and VLSI	Personal computers and workstations

Historical Perspective

Year	Name	Size ($Ft.^3$)	Power (Watt)	Perform. (adds/sec)	Mem. (KB)	Price	Price/ Perform. vs. UNIVAC	Adjusted price 1996	Adjusted price/perform vs. UNIVAC
1951	UNIVAC 1	1000	124K	1.9K	48	$1M	1	$5M	1
1964	IBM S/360 model 50	60	10K	500K	64	$1M	263	$4.1M	318
1965	PDP-8	8	500	330K	4	$16K	10,855	$66K	13,135
1976	Cray-1	58	60K	166M	32,768	$4M	21,842	$8.5M	15,604
1981	IBM PC	1	150	240K	256	$3K	42,105	$4K	154,673
1991	HP 9000/ model 750	2	500	50M	16,384	$7.4K	3,556,188	$8K	16,122,356
1996	Intel PPro PC 200 Mhz	2	500	400M	16,384	$4.4K	47,846,890	$4.4K	239,078,908

After adjusting for inflation, price/performance has improved by about 240 million in 45 years (about 54% per year)

Digital vs. Analog Systems

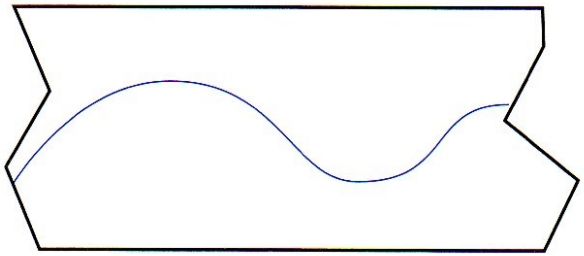
- Advances in the computer industry led to massive digitization of equipment control and electronic devices

Analog systems:

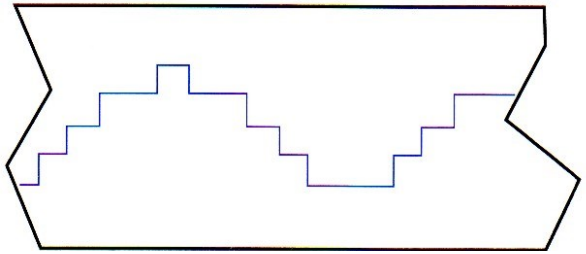
- Sinusoidal signal representation with low precision feedback control
- Fixed functionality that cannot be changed without circuit redesign

Digital Systems:

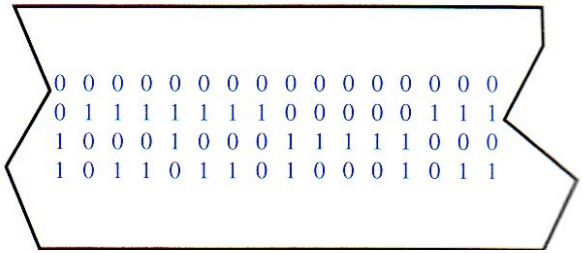
- Store and process information in digital format ➔ change on the fly
- Leverage advances in miniaturized micro-controllers and processors



Analog recording



Sampled form



Digital representation

Digital design is advantageous: (1) High speed (advanced processors)
(2) Flexibility (programmability), (3) Noise immunity (error correction code),
(4) Reproducibility (stored bits), (5) Small (miniaturized & low power chips)

Conclusion

❑ *Summary*

- ➔ Class overview and polices
  (resources, syllabus and work load. Grade structure and policy)
- ➔ Von Neumann and System Bus models.
- ➔ Different levels of abstractions
  (programmer's view, operating system view, hardware designer, etc.)
- ➔ Some historical perspective
- ➔ Digital versus analog systems
  (concepts covered in this course goes beyond computers)

❑ *Next Lecture*

- ➔ Different representations of numbers
- ➔ Arithmetic in different radix
- ➔ Converting numbers between bases (whole and fractions)

Reading assignment: Chapter 1 in the textbook