**ECE 4680**  
Computer Architecture and Organization

**Lecture 1: A Short Journey to the World of Computer Architecture**

Basic Ideas and Definition

Major Components of Software/Hardware

Computer Revolution

---

**What is "Computer Architecture"**

- Co-ordination of levels of abstraction
  - Application
  - Operating System
  - Instruction Set Architecture
  - Compiler
  - I/O system
  - Digital Design
  - Circuit Design

- Merits of Abstraction/Layers/Hierarchy
- Under a set of rapidly changing Forces: technology, applications, Programming Languages, operating systems, history

---

**Technology Trend: Clock rate**

- 30% per year --> today’s PC is yesterday’s Supercomputer

---

**Technology Trends: Transistor Count Growth**

- 40% per year, order of magnitude more contribution in 2 decades
- More and more functions can be performed by a CPU
- Similar story for storage: capacity increased by 1000x over ten years, speed only 2x

---

**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**

- Supercomputers
- Mainframes
- Minicomputers
- Microprocessors

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

**Moore’s Law (1965)**
- The number of transistors on a microchip doubles about every 18-24 months.
- The speed of a microprocessor doubles about every 18-24 months.
- The price of a microchip drops about 48% every 18-24 months, assuming the performance metric (processor speed or memory capacity) of the chip stays the same.

**Notations and Conventions for Numbers**

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Abbreviation</th>
<th>Meaning</th>
<th>Numeric Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>milli</td>
<td>m</td>
<td>One thousandth</td>
<td>$10^{-3}$</td>
</tr>
<tr>
<td>micro</td>
<td>μ</td>
<td>One millionth</td>
<td>$10^{-6}$</td>
</tr>
<tr>
<td>nano</td>
<td>n</td>
<td>One billionth</td>
<td>$10^{-9}$</td>
</tr>
<tr>
<td>pico</td>
<td>p</td>
<td>One trillionth</td>
<td>$10^{-12}$</td>
</tr>
<tr>
<td>femto</td>
<td>f</td>
<td>One quadrillion</td>
<td>$10^{-15}$</td>
</tr>
<tr>
<td>atto</td>
<td>a</td>
<td>One quintillion</td>
<td>$10^{-18}$</td>
</tr>
<tr>
<td>kilo</td>
<td>K (or k)</td>
<td>Thousand</td>
<td>$10^3$ or $2^10$</td>
</tr>
<tr>
<td>mega</td>
<td>M</td>
<td>Million</td>
<td>$10^6$ or $2^{20}$</td>
</tr>
<tr>
<td>giga</td>
<td>G</td>
<td>Billion</td>
<td>$10^9$ or $2^{30}$</td>
</tr>
<tr>
<td>tera</td>
<td>T</td>
<td>Trillion</td>
<td>$10^{12}$ or $2^{33}$</td>
</tr>
<tr>
<td>peta</td>
<td>P</td>
<td>Quadrillion</td>
<td>$10^{15}$ or $2^{38}$</td>
</tr>
<tr>
<td>exa</td>
<td>E</td>
<td>Quintillion</td>
<td>$10^{18}$ or $2^{42}$</td>
</tr>
</tbody>
</table>

Even the measure unit is changing!!!

**How they predict the future**
- Popular Science, 1949 “Computers in the future may weight no more than 1.5 tons”
- Thomas Watson, Chairman of IBM, 1943 “I think there is a world market for maybe five computers”
- Ken Olsen, founder and president of Digital Equipment Corp, 1957 “There is no reason anyone would want a computer in their home”
- Charles H. Duell, Commissioner, U.S. Office of patents “Everything that can be invented has been invented”
- Bill Gates, 1981 “640K ought to be enough for anybody”

**Computer Arch. = Instruction Set Arch. + Organization**

- **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"
**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, Blaw, and Brooks, 1964

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

**MIPS R3000 Instruction Set Architecture**

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

**Instruction Format**

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>OP</td>
<td>Operation</td>
</tr>
<tr>
<td>rs</td>
<td>Register</td>
</tr>
<tr>
<td>rt</td>
<td>Register</td>
</tr>
<tr>
<td>rd</td>
<td>Register</td>
</tr>
<tr>
<td>sa</td>
<td>Source Address</td>
</tr>
<tr>
<td>funct</td>
<td>Function Code</td>
</tr>
<tr>
<td>target</td>
<td>Target Address</td>
</tr>
</tbody>
</table>

**Example Organization**

TI SuperSPARC™ TMS380Z50 in Sun SPARCstation20

**Levels of Representation**

- **High Level Language Program**
  - Compiler: `lw $15, 0($2)`
  - Assembler: `lw $16, 4($2)`
  - Machine Language Program: `sw $16, 0($2)`, `sw $15, 4($2)`

**Measurement and Evaluation**

Architecture is an iterative process

- searching the space of possible designs
- at all levels of computer systems

Bad Ideas  
Creativity  
Cost / Performance Analysis  
Good Ideas
ECE468: Course Overview

Computer Design
- Instruction Set Design
  - Machine Language
  - "Computer Architecture"
  - "Instruction Set Processor"
- "Building Architect"

Computer Hardware Design
- Machine Implementation/Logic Design
- "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.

ECE468: 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 implementation hardware
- Experience with the design process in the context of a large complex (hardware) design.
- Functional Spec → Control & Datapath
- Learn how to completely design a correct single processor computer.
- No magic required to design a computer
- Foundation for students aspiring to work in computer architecture.
- Others: solidifies an intuition about why hardware is as it is.

The SPARCstation 20

Memory Controller Memory Bus
- Slot 1
- Slot 0
- Slot 1
- Slot 0
- Slot 3
- Slot 2
- Slot 1
- Slot 0
- Memory SIMMs

SCSI Bus
- Keyboard & Mouse
- Floppy Disk
- Tape
- External Bus

Levels of Organization

Computer
- SPARC Processor
- Memory
- Devices
- Control
- Datapath
- Input
- Output

The Underlying Network

Memory Controller Memory Bus
- Slot 1
- Slot 0
- MBus
- SEC MACIO
- Standard I/O Bus:
- SCSI Bus
- Processor Bus:
- MBus
- Sun’s High Speed I/O Bus:
- SBus
- Low Speed I/O Bus:
- External Bus

Processor and Caches

MBus Module
- SuperSPARC Processor
- Registers
- Datapath
- Internal Cache
- Control
- External Cache
**Memory**

- Memory Controller
- Memory Bus
- SIMM Slot 0
- SIMM Slot 1
- SIMM Slot 2
- SIMM Slot 3
- SIMM Slot 4
- SIMM Slot 5
- SIMM Slot 6
- SIMM Slot 7
- DRAM SIMM
- DRAM
- DRAM
- DRAM
- DRAM
- DRAM
- DRAM
- DRAM

**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**

- The 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**

- ISA—Principle of abstraction
  - Hiding details from the level above
  - Both software designers and hardware designers comply with
- 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 has the messiest organization
  - Wide range of speed: graphics vs. keyboard
  - Wide range of requirements: speed, standard, cost ... etc.
  - Least amount of research (so far)