INEL 4215: Computer Architecture and Organization

Instructor: Nayda Santiago

Fall 2004

Introduction

Department of Electrical and Computer Engineering
University of Puerto Rico - Mayaguez

What is Computer Architecture?

- Instruction Set Architecture
  - "…the attributes of a system as seen but 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
- Machine Organization
  - ALUs, Buses, Caches, Memories, etc.

Why study Computer Architecture?

- To become a computer designer
  - After this class you can design the computers you use
- To learn what is under the hood of your computer
  - Understand when things break
  - How to design high-performance applications
  - To aid the design of system software (OS, compilers, libraries, etc.)
- To demystify the computer
- It is fascinating!!!!!!!!!!!!

Why is Computer Architecture so Dynamic?

- Everything is changing
- Technology Push
- Application Pull

Technology Push

- What do these two intervals have in common?
  - 1776-1999 (224 years)
  - 2000-2001 (2 years)
  - Answer: Equal progress in absolute processor speed
- Consider salary doubling
- Driven by Moore's Law
  - Device per chip doubles every 18 months to 2 years
- Computer architects work to turn the additional resources into speed.

Some INTEL Numbers

Date	What	Comments
1947	1st Transistor	Bell Labs
1958	1st Integrated Circuits	Texas Instruments
1971	1st Microprocessor	Intel
1974	Intel 4004	2300 transistors
1978	Intel 8086	29K transistors
1989	Intel 80486	1.2M transistors
1995	Intel Pentium Pro	5.5M transistors
2006	Intel Estimate	350M transistors
2011	Intel Estimate	1G transistors

Application Pull

- Corollary to Moore's Law: Cost halves every two years
  - In a decade you can buy a computer for less than its sales tax today --- Jim Gray
- Computers cost-effective for
  - National Security – weapons design
  - Enterprise computing – banking
  - Departmental computing – CAD
  - Personal computing – Spreadsheets, email, web
  - Embedded computing – microcode in electric shavers

Application Pull

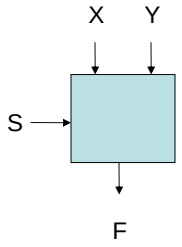
- What about the future?
- Must dream up applications that are not cost-effective today
  - Virtual reality
  - Wireless
  - Ubiquitous computing
- This is your job

Abstraction

- Difference between interface and implementation
  - Interface: WHAT something does
  - Implementation: How it does so

Abstraction

- Example 2-to1 MUX
- Interface:



S	F
0	X
1	Y

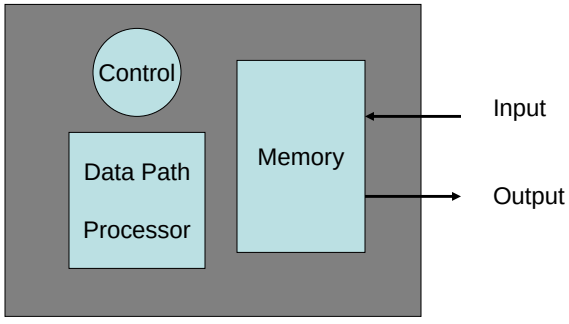
- Implementation
  - Gates, transistors

What is the Big Deal?

- Example, a processor interface book
- Worse for computers, in general - a tower of abstraction
  - Application software
  - System software (OS and compiler/assembler/linker)
  - Hardware (CPU, memory, I/O)
- Each interface is complex and implemented with layer below
- ABSTRACTION KEEPS UNNECESSARY DETAILS HIDDEN
- Hundreds of engineers to build one product

Basic Division of Hardware

- In space and time
  - In space



Basic Division of Hardware

- In time
  - Fetch the instruction from memory
  - Decode the instruction
  - Read input operands
  - Perform operation
  - Write results
  - Determine the next instruction

Classes of Computers

- Supercomputer – $5 to $20 million
- Mainframe – $.5 to $4 million
- Server - $10 to $ 200 thousands
- PC/Workstation - $1 to $10 thousands
- Network Computer - $300 to $1000
- Embedded Computer - $1 to $10