Computer Systems Architecture
Fall 2014

Huzefa Rangwala, PhD
http://www.cs.gmu.edu/~hrangwal

*Slides adapted from Computer Organization and Design by Patterson and Henessey
WHERE YOU SIT IN CLASS/SEMINAR
And what it says about you:

- **Mid-Center:** "Bring it on."
- **Nearest Exit:** Uncommitted
- **Back Row:** "Too cool for school"
- **Front Row:** Teacher's pet wannabes
- **Second-row sleepers:** Good intentions, bad narcolepsy
- **Against the wall:** "I’m sensitive. Please ignore me."

**Proximity to Lecturer:**

\[ X = \frac{How\ much\ you\ care}{How\ sleepy\ you\ are} \]
Today ..

- Welcome
- Syllabus
  - Grading
  - Policies
  - Things you want to know and care about.
- Why should you be in this class?
- What is computer architecture?
- Demos
Location and Office Hours

- **Class Time:** T – 4:30-7:10 pm
- **Class Location:** Arts Building, 2026
- **Instructor:** Huzefa Rangwala
  - Office Hours: Tuesday (2:00-4:00 pm) in Engineering Building #4423.
  - Email: rangwala@cs.gmu.edu
- **TA:**
  - **Office Hours:** W-F(2:00-3:00 pm) in Engineering Building #4457
    - Email: tbd@gmu.edu
Grading

- HW 0 0%
- HW 1 10% (Use of MIPS)
- HW 2 10% (Use of MIPS)
- HW 3 15% (Use of MIPS)
- HW 4 15% (Cache/MIPS)
- Mid-Term 20% (Closed Book)
- Final Exam 25% (Closed Book)
- Class Participation 5%
- Quizzes (Extra) 5%
Class Website and Announcements

- Keep Track of

- Specific Readings will be posted after each lecture. Note, the book is detailed and I will make clear what you are responsible for exams/assignments and also what is extra or “if-interested” reading.
Write-Pair-Share Activity*

- What does computer architecture mean for you?
  - Think about this individually and write 2-3 bullets. (2 minutes)
  - Meet your closest neighbor and discuss this with him/her (3 minutes)
Why? Besides it being required ....

- CS majors
Moore’s Law

The graph illustrates Moore’s Law, which states that the number of transistors on an integrated circuit doubles every 18 months. The x-axis represents the years from 1971 to 2004, and the y-axis represents the number of transistors, ranging from 2,300 to 10,000,000,000. The graph shows the exponential growth of transistors over time, with each major milestone marked by the release of a new processor model, such as the 8086, Pentium, Pentium II, Pentium III, Pentium 4, Itanium, and Itanium 2 (with 9 MB cache).
The Computer Revolution

- Progress in computer technology
  - Underpinned by Moore’s Law
- Makes novel applications feasible
  - Computers in automobiles
  - Cell phones
  - Human genome project
  - World Wide Web
  - Search Engines
- Computers are pervasive
Classes of Computers

- Desktop computers
  - General purpose, variety of software
  - Subject to cost/performance tradeoff
- Server computers
  - Network based
  - High capacity, performance, reliability
  - Range from small servers to building sized
- Embedded computers
  - Hidden as components of systems
  - Stringent power/performance/cost constraints
The Processor Market

[Bar chart showing processor market share for Cell Phones, PCs, and TVs from 1997 to 2007.]
What You Will Learn

• How programs are translated into the machine language
  ◦ And how the hardware executes them
• The hardware/software interface
• What determines program performance
  ◦ And how it can be improved
• How hardware designers improve performance
• What is parallel processing
Understanding Performance

- Algorithm
  - Determines number of operations executed

- Programming language, compiler, architecture
  - Determine number of machine instructions executed per operation

- Processor and memory system
  - Determine how fast instructions are executed

- I/O system (including OS)
  - Determines how fast I/O operations are executed
Below Your Program

- **Application software**
  - Written in high-level language

- **System software**
  - Compiler: translates HLL code to machine code
  - Operating System: service code
    - Handling input/output
    - Managing memory and storage
    - Scheduling tasks & sharing resources

- **Hardware**
  - Processor, memory, I/O controllers
High-level to Machine Language

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

High-level language program
(in C)

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
```

Binary machine language program
(for MIPS)

```
00000000011010000100000000000110000
0000000001100001110000110000010001
1000110001110001000000000000000000
1000110001111100100000000000000010
1010110001111100100000000000000000
1010110001100001000000000000000010
0000001111100000000000000000010000
```
Components of a Computer

- The BIG Picture

- Same components for all kinds of computer
  - Desktop, server, embedded

- Input/output includes
  - User-interface devices
    - Display, keyboard, mouse
  - Storage devices
    - Hard disk, CD/DVD, flash
  - Network adapters
    - For communicating with other computers
Organization of a computer
Anatomy of Computer

5 classic components

- **Datapath**: performs arithmetic operation
- **Control**: guides the operation of other components based on the user instructions

**Computer**

**Processor**
- **Control** ("brain")
- **Datapath** ("brawn")

**Memory**
- (where programs, data live when running)

**Devices**
- **Input**
- **Output**

**Keyboard, Mouse**
**Disk** (where programs, data live when not running)
**Display, Printer**
The von Neumann architecture is a design model for a stored-program digital computer that uses a central processing unit (CPU) and a single separate storage structure ("memory") to hold both instructions and data.
Why Learn Computer Architecture?

- You want to call yourself a “computer scientist”
  - Computer architecture impacts every other aspect of computer science
- You need to make a purchasing decision or offer “expert” advice
- You want to build software people use – sell many, many copies-(need performance)
  - Both hardware and software affect performance
    - Algorithm determines number of source-level statements
    - Language/compiler/architecture determine machine instructions
    - Processor/memory determine how fast instructions are executed
    - Assessing and understanding performance
    - New Trends in Architecture: Multicores, Multiprocessors, GPUS
Demos —

- Demos
- Etc...