CSM151B
Computer Systems Architecture

Discussion 1E

1/12/2018

Contact Info

- TA: Yuchen Hao
- Email: haoyc@cs.ucla.edu
- Office hours
  - Thursdays 4:30 - 6:30 pm
  - BH 2432

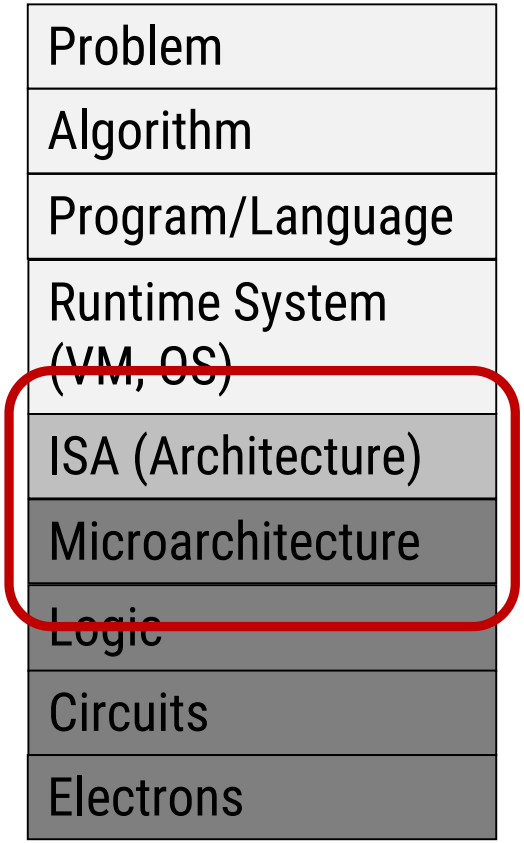
Logistics

- **HW1 due next Friday**
  - **Please submit raw text (i.e. a .txt file) or a PDF file**

Agenda

- Performance

Levels of Abstraction



How to Evaluate Performance

$$\text{Execution time} = \frac{\text{time}}{\text{program}}$$

$$= \frac{\#\ \text{instructions}}{\text{program}} \times \frac{\#\ \text{cycles}}{\text{instruction}} \times \frac{\text{time}}{\text{cycle}}$$

Algorithm		uarch
Program	ISA	Logic design
ISA	uarch	Circuit implementation
Compiler		Technology

Improving Performance

- **Reducing instructions/program**
  - More efficient algorithms and programs
  - Better ISA?
- **Reducing cycles/instruction (CPI)**
  - Better microarchitecture design
    - Execute multiple instructions at the same time
    - Reduce latency of instructions (1-cycle vs. 100-cycle memory access)
- **Reducing time/cycle (clock period)**
  - Technology scaling
  - Pipelining

Sample Question 1

- **A program is written in C. We execute this program on two different computers:**
  Computer A: has a 3GHz x86 processor
  Computer B: has a 3GHz x86 processor

  When we execute this program and measure its CPI in x86 instructions, we find the following results:
  On computer A: CPI = 10
  On computer B: CPI = 8

  What can you say about on which computer this program runs faster?

Sample Question 1

- We don’t know.

- Because we don’t know how many instructions are actually executed for the program on either machine, we cannot conclude which computer runs faster. Although B has lower CPI, but it might be executing 2 times more instructions than A due to a less optimized compiler.

Sample Question 2

- **You are on the design team for a new processor. The following table gives instruction frequencies for a benchmark, as well as how many cycles the instructions take, for the different classes of instructions.**

Instruction Type	Frequency	Cycles
R type	40%	5 cycles
Load	30%	4 cycles
Store	10%	3 cycles
BEQ	20%	2 cycles

- **The compiler expert says that if you double the number of registers, then the compiler will generate code that requires only half the number of loads & stores. What would the new CPI be on the benchmark?**

Sample Question 2

- **If we say that there are 100 instructions, then**
  40 R types
  30 loads
  10 stores
  20 BEQs
- **We will reduce the number of loads and stores by half, so the new instruction mix will be:**
  40 R types
  15 loads
  5 stores
  20 BEQs
- **The total number of instructions is now 80, so the answer is:**
  (40 * 5 + 15 * 4 + 5 * 3 + 20 * 2) / 80