Slide 1

Administrivia

- Reminder: Homework 1 due Friday.

Slide 2

Minute Essay From Last Lecture

- Question:

  We said a program as stored in memory is a sequence of ones and zeros. Name three ways a programmer could produce this sequence. (Hint: some ways involve the use of other programs.)

- Intended answer:
  - Write ones and zeros directly.
  - Write in assembly language and use assembler program.
  - Write in HLL and use compiler program.

Slide 3

A Little About Integrated Circuits, Review

- Conceptual view of hardware:
  - "Transistor" — on/off switch controlled by electrical current.
  - Combine/connect a lot of transistors to get "circuit" that does interesting things (e.g., addition).
  - Put a bunch of circuits together to get a "chip" / "integrated circuit" (IC). If lots of transistors, "VLSI chip".

Slide 4

A Little More About Integrated Circuits

- Manufacturing process starts with a thin flat piece of silicon, adds metal and other stuff to make wires, insulators, transistors, etc.
- Of course, this is all automated! Low-level chip designers use CAD-type tools, which save designs in a standard format, which the chip designers simulate/test with other software, and then send off to be "fabricated".
- Typically make many "chips" on a "wafer", discard those with defects, bond each good one to something larger with "pins" to allow connections to other parts of computer.

Defining Performance

Slide 5

- What does it mean to say that computer A "has better performance than" computer B?
- Really — "it depends". Some answers:
  - Computer A has better response time / smaller execution time.
  - Computer A has higher throughput.
- We'll use execution time, and say

$$\frac{\text{Performance}_A}{\text{Performance}_B} = n$$

exactly when

$$\frac{\text{Execution time}_B}{\text{Execution time}_A} = n$$

Measuring Performance

Slide 6

- If we use execution time as criterion, how to measure?
- Wall-clock time seems fair, since it includes
  - Time for CPU to execute instructions.
  - Any waiting for memory access.
  - Any waiting for I/O.
  - Any waiting for operating system.
- Is that easy to measure reliably / repeatably?

Slide 7

Measuring Performance, Continued

- No — to get repeatable measure of wall clock time, need an otherwise unused system.
- So instead we could use "CPU performance" — amount of time CPU needs to run program. Easier to measure, more consistent.
- Or we could try "clock speed". Can define in terms of "clock period / cycle" or "clock rate" (inverse of clock period).
- Example — for 1GHz processor, what's its clock cycle?

Slide 8

How These Metrics Relate

- CPU execution time for program X is given by

$$\text{CPU cycles} \times \text{clock cycle}$$

- How would you write that using clock rate instead of clock cycle?
- How would you write it if you know number of instructions and (average) number of cycles per instruction?
- What if you can define different classes of instructions, each with a different number of cycles per instruction?
- So, to double performance for a program, is it enough to double the clock rate?

How These Metrics Relate, Continued

- Not necessarily —
  - Could number of instructions change?
  - Could cycles per instruction change?
- Well, but at least it's better to have fewer instructions?

Slide 9

How These Metrics Relate, Continued

- Also not necessarily — e. g., if you replace instructions that take a few cycles each with a few that take a lot of cycles.

  (Textbook example on p. 64+.)

Slide 10

Slide 11

Evaluating / Comparing Performance

- Trickier than it sounds to come up with one number that means something.
- Approaches include
  - Use the actual workload, on the actual hardware platform(s), and compare times.
  - Put together a representative simulated workload — "benchmark"; run and compare times.
  - Compare code size.
  - Compare number of instructions per second ("MIPS" or "MFLOPS").
- Alas, all of these are flawed in some way.

  (Paraphrasing someone whose name I don't remember, "peak MIPS is just the number you can't go any faster than.")

Slide 12

Minute Essay

- Suppose for a given program you have

	Instructions	*Avg cycles/instr*	*Cycle time*
Machine X	1 million	1.5	1 ns
Machine Y	1 million	2	0.5 ns

  (1 second = 10^9 ns)

- Which machine is faster? by how much? (e.g., "X is twice as fast as Y".)