Review - CPU Performance

We will use “n times faster”, which means both increased performance and decreased execution time.

- For some program running on machine X, Performance _X = 1 / Execution time _X
- "X is n times faster than Y" Performance _X / Performance _Y = n

Review - How to improve performance

\[
\frac{\text{seconds}}{\text{program}} = \frac{\text{cycles}}{\text{program}} \times \frac{\text{cycles/program}}{\text{clock rate}}
\]

- So, to improve performance (everything else being equal) one can either
  - Decrease the # of required cycles for a program, or
  - Decrease the clock cycle time or, said another way, Increase the clock rate.

Review - Different instructions take different number of cycles

![Figure 6.3](image.png)

- Note that "lw" is longer than "add" and "beq"
- lw~800 ps, R-type~600 ps, beq~500ps [Fig. 6.2]
Review - *Different instructions take different number of cycles*

![Program execution order diagram]

- Instruction fetch
- Reg
- ALU
- Reg

- Note that "lw" is longer than "add" and "beq"
- lw~800 ps, R-type~600 ps, beq~500ps [Fig. 6.2]
- What is the worst-case clock cycle in this pipelined execution?

---

**CPI - Cycles per Instruction**

- Number of clock cycles to execute an instruction
- Average CPI for a program can be computed:

\[
\text{CPI}_{\text{ave}} = \frac{\text{CPU clock cycles}}{\text{instruction count}}
\]

- Average CPI is useful for comparing two implementations of the same ISA since the instruction count for a program will be the same.

---

**Performance**

Review some ideas and terms...

- A given program will require
  - some number of instructions (machine instructions)
  - some number of cycles
  - some number of seconds

- The vocabulary that relates these quantities:
  - cycle time (seconds per cycle)
  - clock rate (cycles per second)
  - CPI (cycles per instruction)
  - MIPS (million of instructions per second)
    - MIPS (Microprocessor without Interlocked Piped Stages)
    - Software that simulates MIPS code is called SPIM

---

**Performance Question...**

- Performance is determined by execution time
- Do any of the other variables equal performance? If the answer is yes, which one(s)?
  - # of cycles to execute program?
  - # of instructions in program?
  - # of cycles per second?
  - average # of cycles per instruction?
  - average # of instructions per second?

- Common pitfall: thinking *only one* of the above variables is indicative of performance.
Performance

### CPI example 1
- Suppose we have two implementations of the same instruction set architecture (ISA). For some program,
  - Machine A has a clock rate of 4 GHz and a CPI of 2.0.
  - Machine B has a clock rate of 2 GHz and a CPI of 1.2.
- Which machine is faster for this program?

### MIPS as a Performance Measure
- MIPS is an instruction execution rate
- MIPS is Million Instructions Per Second
- MIPS is easy to understand

\[
\text{MIPS} = \frac{\text{instruction count}}{\text{CPU time} \times 10^6}
\]

- **Problems with using MIPS as a measure for comparing computers:**
  - Cannot use MIPS to compare two machines with different ISA's!
  - MIPS varies between programs on same computer
  - MIPS can vary inversely with performance!

<table>
<thead>
<tr>
<th>Instruction Count</th>
<th>Computer C1</th>
<th>Computer C2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10 billion</td>
<td>8 billion</td>
</tr>
<tr>
<td>Clock Rate</td>
<td>4 GHz</td>
<td>4 GHz</td>
</tr>
<tr>
<td>CPI</td>
<td>1.0</td>
<td>1.1</td>
</tr>
</tbody>
</table>

Performance

CPI example 2

Two computers, C1 and C2, have the following metrics when running the same program. Which computer is faster? Which has higher MIPS?
Performance

In terms of instructions...

- CPU time = Instruction count x CPI x Clock cycle time
- CPU time = Instruction count x CPI / Clock rate

- What about instructions with different cycle count?

Performance example

In terms of instruction count...

A compiler designer is trying to decide between two code sequences for a particular machine. Based on the hardware implementation, there are three different classes of instructions: Class A, Class B, and Class C, and they require one, two, and three CPI (respectively).

- The first code sequence has 5 instructions:
  - 2 of A, 1 of B, and 2 of C
- The second sequence has 6 instructions:
  - 4 of A, 1 of B, and 1 of C

Which sequence will be faster? How much? What is the CPI for each sequence?

Performance

Amdahl's Law

Pitfall: Expecting the improvement of one aspect of a computer to increase performance by an amount proportional to the size of the improvement!

- Execution Time After Improvement = Execution Time Unaffected + (Execution Time Affected / Amount of Improvement)

Example (pg. 267):
“Suppose a program runs in 100 seconds on a machine, with multiply operation responsible for 80 seconds of this time. How much do we have to improve the speed of multiplication if we want the program to run five times faster?”
Performance

**Remember...**

- Performance is specific to a particular program(s)
  - Total execution time is a consistent summary of performance
- For a given architecture performance increases come from:
  - Increases in clock rate (without adverse CPI effects)
  - Improvements in processor organizations that lower CPI
  - Compiler enhancements that lower CPI and/or instruction count

**They work together...**

The performance of software systems is dramatically affected by how well software designers understand the basic hardware technologies at work in a system. Similarly, hardware designers must understand the far-reaching effects their design decisions have on software applications.

---

**Hypothetical news release**

Question 4-51 pg. 277

“The company will unveil the industry’s first 5 GHz version of the chip, which offers a 25% performance boost over the company’s former speed champ, which runs at 4 GHz…”

Comment on the definition of performance that you believe the company used. Is the news misleading?
Next class...

- Benchmarks