Machine Internals / Performance Quick Recap

- **CPI** = cycles per instruction
  - determined by CPU hardware (CPU = central processing unit)
  - different instructions might have different CPI

- **Instruction count**
  - determined by program, instruction set, compiler

- **Clock cycles** = instruction count * CPI
- **CPU time** = instruction count * CPI * cycle time
Machine Internals / Performance: Example

- Using the formulas on the previous slide, let's try an example:

Consider two different implementations of the same instruction set architecture. There are four classes of instructions, A, B, C, and D. The clock rate and CPI of each implementation are given in the following table.

<table>
<thead>
<tr>
<th></th>
<th>Clock Rate</th>
<th>CPI Class A</th>
<th>CPI Class B</th>
<th>CPI Class C</th>
<th>CPI Class D</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1.6 GHz</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>P2</td>
<td>2 GHz</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

a. Given a program with $10^6$ instructions divided into classes as follows: 10% class A, 20% class B, 50% class C and 20% class D, which implementation is faster?
Machine Internals / Performance: Solution

- Solution to the previous example

\[
\text{Since the clock rate for P1 is 1.6 GHz, this means:}
\]
\[1.6 \times 10^9 \text{ cycles} = 1 \text{ sec}\]
\[1 \text{ cycle} = 1/(1.6 \times 10^9) \text{ sec}\]
\[1 \text{ cycle} = 0.625 \times 10^{-9} \text{ sec OR 625 ps}\]

\[
\text{Since the clock rate for P2 is 2 GHz, through a similar calculation, we see that}\n\]
\[1 \text{ cycle} = 500 \text{ ps}\]

\[
\text{Total CPU time for the program using P1 is:}\n\]
\[(0.1 \times 625 \text{ ps} + 0.2 \times 2 \times 625 \text{ ps} + 0.5 \times 3 \times 625 \text{ ps} + 0.2 \times 4 \times 625 \text{ ps}) \times 10^6\]
\[= 1750000000 \text{ ps}\]
\[= 1.75 \text{ milliseconds}\]

\[
\text{The total CPU time for the program using P2 is:}\n\]
\[(0.1 \times 2 \times 500 \text{ ps} + 0.2 \times 2 \times 500 \text{ ps} + 0.5 \times 2 \times 500 \text{ ps} + 0.2 \times 2 \times 500 \text{ ps}) \times 10^6\]
\[= 1000000000 \text{ ps}\]
\[= 1 \text{ millisecond}\]

\[
\text{This means that the P2 implementation is faster.}\]
Machine Internals / Performance: Example 2

- Building off the previous example (same details):

Consider two different implementations of the same instruction set architecture. There are four classes of instructions, A, B, C, and D. The clock rate and CPI of each implementation are given in the following table.

<table>
<thead>
<tr>
<th></th>
<th>Clock Rate</th>
<th>CPI Class A</th>
<th>CPI Class B</th>
<th>CPI Class C</th>
<th>CPI Class D</th>
</tr>
</thead>
<tbody>
<tr>
<td>P1</td>
<td>1.6 GHz</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>P2</td>
<td>2 GHz</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

b. What is the global CPI for each implementation?
Solution to the previous example

Soln:
The program clock cycles are divided as 10% A, 20% B, 50% C, and 20% D.
For P1 we know A = 1CPI, B = 2CPI, C = 3CPI, and D = 4CPI so we can multiply.
\( (0.1 \times 1 + 0.2 \times 2 + 0.5 \times 3 + 0.2 \times 4) = 2.8 \text{ CPI} \)

We do the same for P2, knowing that A,B,C,D all have 2CPI on P2.
\( (0.1 \times 2 + 0.2 \times 2 + 0.5 \times 2 + 0.2 \times 2) = 2 \text{ CPI} \)

OR
Since all of the individual instructions in P2's set have a CPI of 2, then P2's global CPI is also 2.
Machine Internals: Practice 1

Instruction set A takes 4 cycles for a lw or sw, and 1 cycle for everything else. Instruction set B takes 2 cycles for a lw or sw, and 2 cycles for everything else.

Consider a CPU of type A at 1GHz, and a CPU of type B at 2GHz, which runs a program with 30% lw/sw and 70% other instructions faster?

\[
\text{CPU Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}
\]
Instruction set A takes 4 cycles for a lw or sw, and 1 cycle for everything else. Instruction set B takes 2 cycles for a lw or sw, and 2 cycles for everything else.

Consider a CPU of type A at 1GHz, and a CPU of type B at 2GHz, which runs a program with 30% lw/sw and 70% other instructions faster?

1GHz = 1ns/cycle = 1000ps/cycle
2GHz = 500ps/cycle

\[(1000ps \times 4 \text{ cycles} \times 0.3) + (1000ps \times 1 \text{ cycle} \times 0.7) = 1200 + 700 = 1900\text{ps/inst.}\]
\[(500ps \times 2 \text{ cycles} \times 0.3) + (500ps \times 2 \text{ cycles} \times 0.7) = 300 + 700 = 1000\text{ps/inst.}\]

CPU B will run such a program much faster \((1900/1000 = 1.9\text{ times faster})\).
Instruction set A takes 4 cycles for a lw or sw, and 1 cycle for everything else.
Instruction set B takes 2 cycles for a lw or sw, and 2 cycles for everything else.

Consider a type A CPU and a type B CPU running different programs. CPU A is running a program with 1000 instructions that is 20% lw/sw. CPU B is running a program that is 1200 instructions, but is only 15% lw/sw. They both complete their programs in the same amount of time: 2μs. What are the clock frequencies of each of the CPUs in GHz?

\[
\text{CPU Time} = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}
\]
Machine Internals: Practice 2

Instruction set A takes 4 cycles for a lw or sw, and 1 cycle for everything else.
Instruction set B takes 2 cycles for a lw or sw, and 2 cycles for everything else.

Consider a type A CPU and a type B CPU running different programs. CPU A is running a program with 1000 instructions that is 20% lw/sw. CPU B is running a program that is 1200 instructions, but is only 15% lw/sw. They both complete their programs in the same amount of time: 2μs. What are the clock frequencies of each of the CPUs in GHz?

CPU A: 1000 * 0.2 = 200 lw/sw and 800 everything else so we have 200 * 4 + 800 * 1 = 1600 cycles total
CPU B: All instructions are 2 cycles so: 1200 * 2 = 2400 cycles total

1600cycles / 2μs = 800MHz = 0.8GHz for CPU A
2400cycles / 2μs = 1200MHz = 1.2GHz for CPU B

Or the more difficult way:
2μs / 1600cycles = 0.00125μs = 1.25ns/cycle => 0.8GHz for CPU A
2μs / 2400cycles = 0.000833μs = 833ps/cycle => 1.2GHz for CPU B
Pipeline Stages

- Consider the following table of pipelining performance for a 5GHz CPU

- Assume time for stages is
  - 100ps for register read/write
  - 200ps for other stages

<table>
<thead>
<tr>
<th>Instr</th>
<th>Instr fetch</th>
<th>Register read</th>
<th>ALU op</th>
<th>Memory access</th>
<th>Register write</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>lw</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td>200ps</td>
<td>100 ps</td>
<td>800ps</td>
</tr>
<tr>
<td>sw</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td>200ps</td>
<td></td>
<td>700ps</td>
</tr>
<tr>
<td>R-format</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td></td>
<td>100 ps</td>
<td>600ps</td>
</tr>
<tr>
<td>beq</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td></td>
<td></td>
<td>500ps</td>
</tr>
</tbody>
</table>
Non-Pipelined Performance Example

Use the table solve the following:

How long does it take a non-pipelined CPU to execute the following instructions assuming register $10$ starts with value $5$?

```
loop:  lw $2, 0($30)
      sw $1, 0($30)
      beq $0, $10, loop
      add $10, $0, $0
      beq $0, $10, loop2
      add $10, $10, $0
loop2: add $10, $10, $10
```

- assume time for stages is
  - $100ps$ for register read/write
  - $200ps$ for other stages

<table>
<thead>
<tr>
<th>Instr</th>
<th>Instr fetch</th>
<th>Register read</th>
<th>ALU op</th>
<th>Memory access</th>
<th>Register write</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>lw</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td>200ps</td>
<td>100 ps</td>
<td>800ps</td>
</tr>
<tr>
<td>sw</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td>200ps</td>
<td></td>
<td>700ps</td>
</tr>
<tr>
<td>R-format</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td></td>
<td>100 ps</td>
<td>600ps</td>
</tr>
<tr>
<td>beq</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td></td>
<td></td>
<td>500ps</td>
</tr>
</tbody>
</table>
Non-Pipelined Performance Example

How long does it take a non-pipelined CPU to execute the following instructions assuming register $10$ starts with value $5$?

```
loop:   lw $2, 0($30)
        sw $1, 0($30)
        beq $0, $10, loop
        add $10, $0, $0
        beq $0, $10, loop2
        add $10, $10, $0
loop2: add $10, $10, $10
```

- Assume time for stages is:
  - $100ps$ for register read/write
  - $200ps$ for other stages

<table>
<thead>
<tr>
<th>Instr</th>
<th>Instr fetch</th>
<th>Register read</th>
<th>ALU op</th>
<th>Memory access</th>
<th>Register write</th>
<th>Total time</th>
</tr>
</thead>
<tbody>
<tr>
<td>lw</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td>200ps</td>
<td>100 ps</td>
<td>800ps</td>
</tr>
<tr>
<td>sw</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td>200ps</td>
<td></td>
<td>700ps</td>
</tr>
<tr>
<td>R-format</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td></td>
<td>100 ps</td>
<td>600ps</td>
</tr>
<tr>
<td>beq</td>
<td>200ps</td>
<td>100 ps</td>
<td>200ps</td>
<td></td>
<td></td>
<td>500ps</td>
</tr>
</tbody>
</table>

$800 \text{(lw)} + 700 \text{(sw)} + 500 \text{(beq, branch not taken }$10 \neq $0) + 600 \text{(add)} + 500 \text{(beq, branch is taken, }$10 = $0) + 600 \text{(add)} = 3700ps$
Assignment and midterm reminders

- Always test your code on the university servers before you hand it in.
- Hand in the .asm file (*not the .mips file*)