Relative Performance

- Define Performance = 1/Execution Time
- “X is \( n \) time faster than Y”

\[
\text{Performance}_X / \text{Performance}_Y = \frac{\text{Execution time}_Y}{\text{Execution time}_X} = n
\]

- Example: time taken to run a program
  - 10s on A, 15s on B
  - \( \text{Execution Time}_B / \text{Execution Time}_A = 15s / 10s = 1.5 \)
  - So A is 1.5 times faster than B
    (we do NOT say that "B is 33.3% slower than A")
CPU Time

CPU Time $= \text{CPU Clock Cycles} \times \text{Clock Cycle Time}$

$= \frac{\text{CPU Clock Cycles}}{\text{Clock Rate}}$

Performance improved by
- Reducing number of clock cycles
- Increasing clock rate
- Hardware designer must often trade off clock rate against cycle count
Instruction Count and CPI

Clock Cycles = Instruction Count × Cycles per Instruction

CPU Time = Instruction Count × CPI × Clock Cycle Time

\[
\text{Instruction Count} \times \text{CPI} \over \text{Clock Rate}
\]

- Instruction Count for a program
  - Determined by program, ISA and compiler
- Average cycles per instruction
  - Determined by CPU hardware
  - If different instructions have different CPI
    - Average CPI affected by instruction mix
CPI Example

- Computer A: Cycle Time = 250ps, CPI = 2.0
- Computer B: Cycle Time = 500ps, CPI = 1.2
- Same ISA
- Which is faster, and by how much?

\[
\text{CPU Time}_{A} = \text{Instruction Count} \times \text{CPI}_{A} \times \text{Cycle Time}_{A}
\]

\[
= l \times 2.0 \times 250\text{ps} = l \times 500\text{ps} \quad \rightarrow \quad \text{A is faster…}
\]

\[
\text{CPU Time}_{B} = \text{Instruction Count} \times \text{CPI}_{B} \times \text{Cycle Time}_{B}
\]

\[
= l \times 1.2 \times 500\text{ps} = l \times 600\text{ps}
\]

\[
\frac{\text{CPU Time}_{B}}{\text{CPU Time}_{A}} = \frac{l \times 600\text{ps}}{l \times 500\text{ps}} = 1.2 \quad \rightarrow \quad \ldots\text{by this much}
\]
CPI in More Detail

- If different instruction classes take different numbers of cycles

\[
\text{Clock Cycles} = \sum_{i=1}^{n} (\text{CPI}_i \times \text{Instruction Count}_i)
\]

- Weighted average CPI

\[
\text{CPI} = \frac{\text{Clock Cycles}}{\text{Instruction Count}} = \sum_{i=1}^{n} \left( \text{CPI}_i \times \frac{\text{Instruction Count}_i}{\text{Instruction Count}} \right)
\]

Relative frequency
### CPI Example

Alternative compiled code sequences using instructions in classes A, B, C

<table>
<thead>
<tr>
<th>Class</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPI for class</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>IC in sequence 1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>IC in sequence 2</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Sequence 1: IC = 5**
- Clock Cycles
  \[= 2 \times 1 + 1 \times 2 + 2 \times 3\]
  \[= 10\]
- Avg. CPI = 10/5 = 2.0

**Sequence 2: IC = 6**
- Clock Cycles
  \[= 4 \times 1 + 1 \times 2 + 1 \times 3\]
  \[= 9\]
- Avg. CPI = 9/6 = 1.5
Performance Summary

The BIG Picture

CPU Time = \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Clock cycles}}{\text{Instruction}} \times \frac{\text{Seconds}}{\text{Clock cycle}}

- Performance depends on
  - Algorithm: affects IC, possibly CPI
  - Programming language: affects IC, CPI
  - Compiler: affects IC, CPI
  - Instruction set architecture: affects IC, CPI, $T_c$