Defining Performance

Which airplane has the best performance?

- Passenger Capacity
- Cruising Range (miles)
- Cruising Speed (mph)
- Passengers x mph
Response Time and Throughput

- **Response time**
  - How long it takes to do a task

- **Throughput**
  - Total work done per unit time
    - e.g., tasks/transactions/… per hour

How are response time and throughput affected by

- Replacing the processor with a faster version?
- Adding more processors?

We’ll focus on response time for now…
**Relative Performance**

- Define Performance = \(1/\text{Execution Time}\)
- “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
Measuring Execution Time

- **Elapsed time**
  - Total response time, including all aspects
    - Processing, I/O, OS overhead, idle time
  - Determines system performance

- **CPU time**
  - Time spent processing a given job
    - Discounts I/O time, other jobs’ shares
  - Comprises user CPU time and system CPU time
  - Different programs are affected differently by CPU and system performance
CPU Clocking

- Operation of digital hardware governed by a constant-rate clock

- Clock period: duration of a clock cycle
  - e.g., 250ps = 0.25ns = 250×10^{-12}s

- Clock frequency (rate): cycles per second
  - e.g., 4.0GHz = 4000MHz = 4.0×10^9Hz
CPU Time

CPU Time = 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
CPU Time Example

- Computer A: 2GHz clock, 10s CPU time
- Designing Computer B
  - Aim for 6s CPU time
  - Can do faster clock, but causes $1.2 \times$ clock cycles
- How fast must Computer B clock be?
CPU Time Example

- Computer A: 2GHz clock, 10s CPU time
- Designing Computer B
  - Aim for 6s CPU time
  - Can do faster clock, but causes $1.2 \times$ clock cycles
- How fast must Computer B clock be?

\[
\text{Clock Rate}_B = \frac{\text{Clock Cycles}_B}{\text{CPU Time}_B} = \frac{1.2 \times \text{Clock Cycles}_A}{6s}
\]

\[
\text{Clock Cycles}_A = \text{CPU Time}_A \times \text{Clock Rate}_A
\]

\[
= 10s \times 2GHz = 20 \times 10^9
\]

\[
\text{Clock Rate}_B = \frac{1.2 \times 20 \times 10^9}{6s} = \frac{24 \times 10^9}{6s} = 4GHz
\]
Instruction Count and CPI

Clock Cycles = Instruction Count \times \text{Cycles per Instruction}

CPU Time = Instruction Count \times CPI \times \text{Clock Cycle Time}

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

- 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?
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?

\[
\begin{align*}
\text{CPU Time}_A &= \text{Instruction Count}_A \times \text{CPI}_A \times \text{Cycle Time}_A \\
&= l \times 2.0 \times 250\text{ps} = l \times 500\text{ps} \\
\text{CPU Time}_B &= \text{Instruction Count}_B \times \text{CPI}_B \times \text{Cycle Time}_B \\
&= l \times 1.2 \times 500\text{ps} = l \times 600\text{ps} \\
\frac{\text{CPU Time}_B}{\text{CPU Time}_A} &= \frac{l \times 600\text{ps}}{l \times 500\text{ps}} = 1.2
\end{align*}
\]

A is faster…

…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

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

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</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>IC in sequence</td>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

- Which code sequence executes the most instructions? Which one will be faster? What is the CPI for each sequence?
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

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

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

\[
\text{Power} = \text{Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}
\]

- \( \times 30 \)
- \( 5V \rightarrow 1V \)
- \( \times 1000 \)
Reducing Power

- Suppose we developed a new, simpler processor that has 85% of the capacitive load of the more complex older processor. Further, assume that it has adjustable voltage so that it can reduce voltage 15% compared to processor B, which results in a 15% shrink in frequency.

- What is the impact on dynamic power?
Reducing Power

- Suppose a new CPU has
  - 85% of capacitive load of old CPU
  - 15% voltage and 15% frequency reduction

\[
\frac{P_{\text{new}}}{P_{\text{old}}} = \frac{C_{\text{old}} \times 0.85 \times (V_{\text{old}} \times 0.85)^2 \times F_{\text{old}} \times 0.85}{C_{\text{old}} \times V_{\text{old}}^2 \times F_{\text{old}}} = 0.85^4 = 0.52
\]

- The power wall
  - We can’t reduce voltage further
  - We can’t remove more heat

- How else can we improve performance?