CSEE 3827: Fundamentals of Computer Systems, Spring 2011

8. Processor Performance

Prof. Martha Kim (martha@cs.columbia.edu)
Web: http://www.cs.columbia.edu/~martha/courses/3827/sp11/

Outline (H&H 7.1)

- Performance Analysis

Microarchitecture

Application Software	programs
Operating Systems	device drivers
Architecture	instructions registers
Micro-architecture	datapaths controllers
Logic	adders memories
Digital Circuits	AND gates NOT gates
Analog Circuits	amplifiers filters
Devices	transistors diodes
Physics	electrons

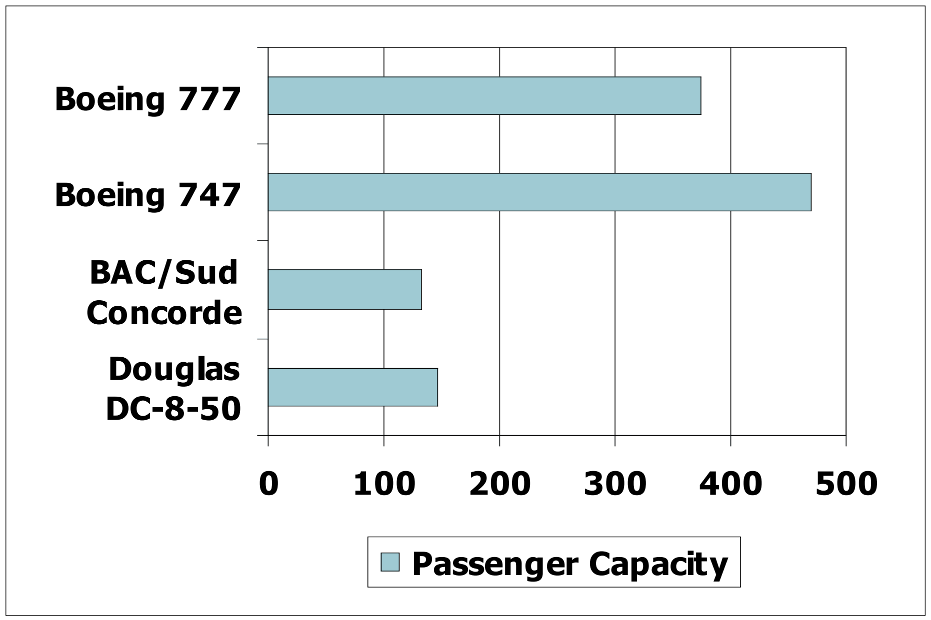
- Multiple implementations for a single architecture
  - Single-cycle: *Each instruction executes in a single cycle*
  - Multi-cycle: *Each instruction is broken up into a series of shorter steps*
  - Pipelined
    - *Each instruction is broken up into a series of steps*
    - *Multiple instructions execute at once*

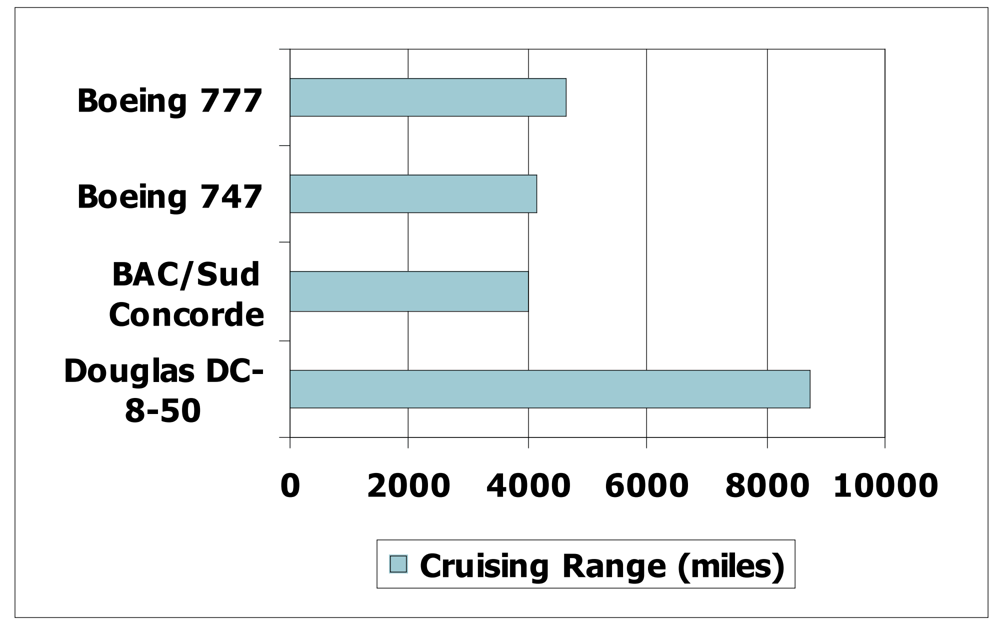
Understanding Performance

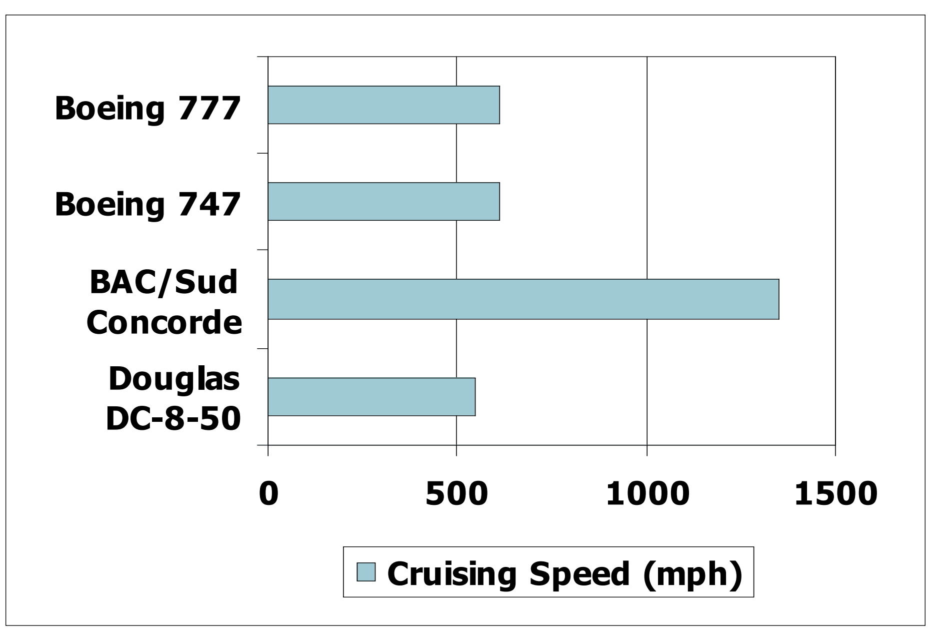
- Algorithm → number of operations executed
- Programming language, compiler, architecture → determine number of machine instructions executed per operation
- Processor and memory system → determines how fast instructions are executed
- I/O system (including OS) → determines how fast I/O operations are executed

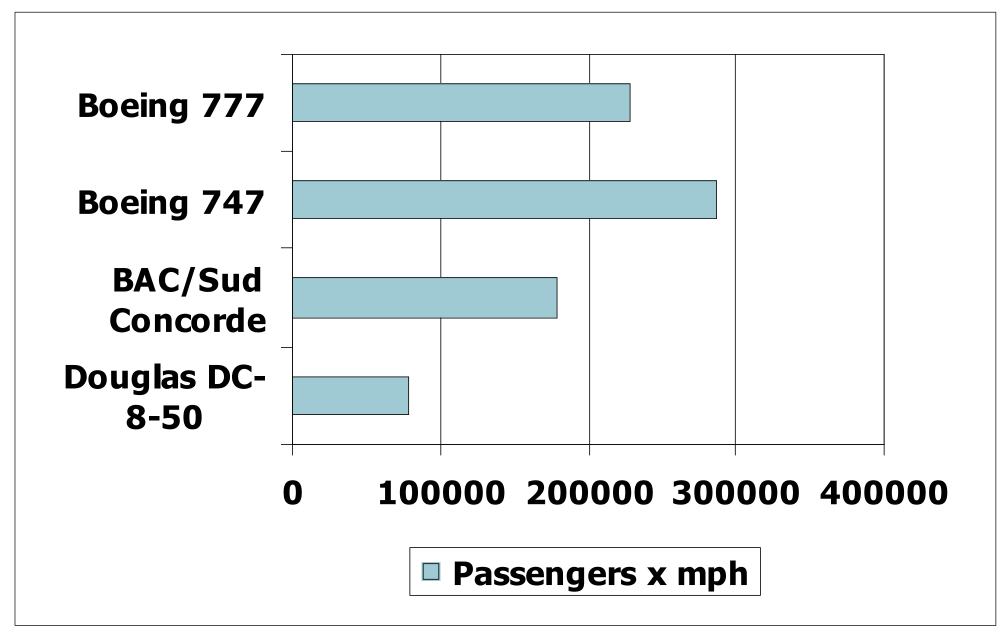
Defining Performance

- Which airplane has the best performance?









Response Time and Throughput

Response time:
how long it takes to do a task,
sometimes also called latency [time/work]

Throughput:
total work done per unit time [work/time]

How are response time and throughput affected by. . .
Replacing the processor with a faster version?

Adding more processors?

For now, we'll focus on response time

Processor Performance, In a Nutshell

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

$$\left(\begin{array}{l} \text{Cycles/instruction} = \text{CPI} \\ \text{Seconds/cycle} = \text{clock period} \\ \text{Instructions/cycle} = \text{IPC} = 1/\text{CPI} \end{array} \right)$$

Relative Performance

Define: Performance = 1 / Execution Time

"X is n times faster than Y" →
Performance X / Performance Y =
Execution Time Y / Execution Time X =
n

Example:

Program takes 10 s to run on machine A, 15 s on machine B
Execution Time B / Execution Time A = 15 / 10 = 1.5
"A is 1.5 times faster than B"

Measuring Execution Time

Define: Elapsed Time

Total response time including all aspects
(Processing, I/O, overhead, idle time)

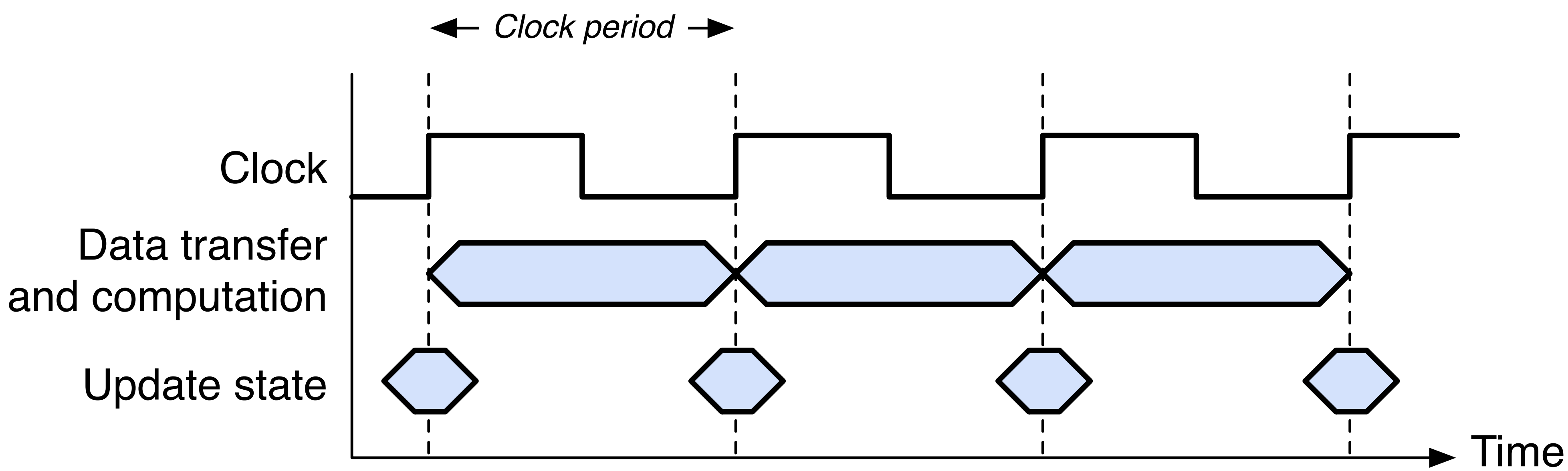
Define: CPU Time

Time spent processing a given job
(discounts I/O time, other jobs shares)

Elapsed Time > CPU Time

CPU Clocking

Operation of digital hardware governed by a constant-rate clock



Clock period: duration of a clock cycle
e.g., 250ps = 0.25ns

Clock frequency (rate): cycles per second
e.g., 4.0GHz = 4000MHz

CPU Time

CPU Time = CPU Clock Cycles * Clock Cycle Time

= CPU Clock Cycles / Clock Rate

Performance improved by:

1. Reducing number of clock cycles
2. Increasing clock rate (reducing clock period)

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
- Clock rate increase requires 1.2x the number of cycles

How fast must Computer B's clock be?

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

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

$$= 10s \times 2\text{GHz} = 20 \times 10^9$$

$$\text{Clock Rate}_B = \frac{1.2 \times 20 \times 10^9}{6s} = \frac{24 \times 10^9}{6s} = 4\text{GHz}$$

Instruction Count and CPI

Instruction count

Determined by program, ISA, and compiler

Average cycles per instruction (CPI)

- Determined by CPU hardware
- If different instructions have different CPI, can compute a weighted average based on instruction mix

Clock Cycles = Instruction Count * Cycles per Instruction

CPU Time = Instruction Count * CPI * Clock Cycle Time

= (Instruction Count * CPI) / Clock Rate

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?

$$\text{CPU Time}_A = \text{Instruction Count} \times \text{CPI}_A \times \text{Cycle Time}_A$$
$$= I \times 2.0 \times 250\text{ps} = I \times 500\text{ps}$$

A is faster...

$$\text{CPU Time}_B = \text{Instruction Count} \times \text{CPI}_B \times \text{Cycle Time}_B$$
$$= I \times 1.2 \times 500\text{ps} = I \times 600\text{ps}$$

$$\frac{\text{CPU Time}_B}{\text{CPU Time}_A} = \frac{I \times 600\text{ps}}{I \times 500\text{ps}} = 1.2$$

… by this much

Amdahl's Law

Be aware when optimizing. . .

$$T_{improved} = \frac{T_{affected}}{\text{improvement factor}} + T_{unaffected}$$

Example: On machine A, multiplication accounts for 80s out of 100s total CPU time.

How much improvement in multiplication performance to get 5x speedup overall?

Corollary: make the common case fast

Performance Summary

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

Algorithm, programming language and compiler compiler affect these terms.

ISA affects all three.

Performance depends on all of these things.