Performance

How do we evaluate computer architectures?

- Think of 5 characteristics that differentiate computers?

1) processor, GPU, mother board
2) memory, storage (
3) O/S

performance

4) I/O, drivers
5) color, form factor, screen res, portability
6) power
7) price

8) reliability

i>clicker

Can some computer processors compute things that others can't?

a) Yes

b) No

Two notions of performance

	Aircraft	DC to Paris	Passengers	
	Aircraft	DC to Paris	Passengers	
A)	747	6 hours	500	– throughput
B)	Concorde	3 hours	125	– latency

- Which has higher performance?
- From a passenger's viewpoint: latency (time to do the task)
  - hours per flight, execution time, response time
- From an airline's viewpoint: throughput (tasks per unit time)
  - passengers per hour, bandwidth
- Latency and throughput are often in opposition

Some definitions for performance

- Relative performance: “x is *N* times faster than y”

$$\frac{\text{Performance}(x)}{\text{Performance}(y)} = N$$

- If we are primarily concerned with latency,

$$\text{Performance}(x) = \frac{1}{\text{Latency}(x)}$$

- If we are primarily concerned with throughput,

$$\text{Performance}(x) = \text{Throughput}(x)$$

The Iron Law of Computing: The time it takes to run a program depends on three factors

1. The number of *dynamic* instructions *N* in the program $\frac{I}{P}$
   - Executing more instructions tends to take longer.
2. The kind of instructions in the program $\frac{CC}{I}$
   - Some instructions take more CPU cycles than others
   - Let *c* be the *average* number of cycles per instruction (CPI)
3. The time *t* per CPU clock cycle (clock-cycle time) $\frac{S}{CC}$

CPU time = Instructions executed × CPI × Clock cycle time

$$\frac{\text{Seconds}}{\text{Program}} = \frac{\cancel{\text{Instructions}}}{\text{Program}} \times \frac{\cancel{\text{Clock cylces}}}{\cancel{\text{Instruction}}} \times \frac{\text{Seconds}}{\cancel{\text{Clock cylce}}}$$

Iron Law Component (1) Dynamic instruction Count is determined during runtime

Not the number of lines of code!!!!

```
for (int i = 0; i < N; i++) {
   //10 lines of code;
}
```

~~Static lines of code: 11~~

Dynamic instructions: ~12*N + 1

Iron Law Component (2) The Average Cycles per instruction depends on both the machine and the program

- Example: CPI(floating-point operations) > CPI(integer operations)
- Example: Improved processor may execute same instructions in fewer cycles
- Single-cycle machine: each instruction takes at least 1 cycle (CPI = 1)
  - CPI can be > 1 due to memory stalls and slow instructions
  - CPI can be < 1 on superscalar machines

Iron Law Component (3) Clock Cycle Time is determined by the worst-case path delays between clocked state elements

- 1 cycle = minimum time it takes the CPU to do any work
- clock cycle time = 1/ clock frequency
  - Example: 500MHz processor has a cycle time of 2ns (nanoseconds)
  - Example: 2GHz (2000MHz) CPU has a cycle time of just 0.5ns
- Higher frequency is usually better for performance

How do we improve performance for each component of the Iron Law of Computing?

CPU time = Instructions executed × CPI × Clock cycle time

complex arch march transitor

- We can improve performance by making any component smaller

	Program	Compiler	ISA	Organization	Technology
Instruction Executed	X	X	X		
CPI	X	X	X	X	
Clock Cycle Time			X	X	X

	a)	b)	c)	d)	e)
	X	X	X	_	_
	_	X	X	X	_
	_	_	X	X	X

i>clicker

Example 1: ISA-compatible processors

- Let's compare the performances two x86-based processors.
  - An 800MHz AMD Duron, with a CPI of 1.2 for an MP3 compressor.
  - A 1GHz Pentium III with a CPI of 1.5 for the same program.
- Compatible processors implement identical instruction sets and will use the same executable files, with the same number of instructions.
- But they implement the ISA differently, which leads to different CPIs.

$$\text{CPU time}_{AMD,P} = \text{Instructions}_P * \text{CPI}_{AMD,P} * \text{Cycle time}_{AMD}$$
$$=$$
$$=$$

$$\text{CPU time}_{P3,P} = \text{Instructions}_P * \text{CPI}_{P3,P} * \text{Cycle time}_{P3}$$
$$=$$
$$=$$

Example 2: Comparing across ISAs

- Intel's Itanium (IA-64) ISA is designed facilitate executing multiple instructions per cycle. If an Itanium processor achieves an average CPI of .3 (3 instructions per cycle), how much faster is it than a Pentium4 (which uses the x86 ISA) with an average CPI of 1?

a) Itanium is three times faster

b) Itanium is one third as fast

c) Not enough information

i>clicker