How do we evaluate computer architectures?

- Think of 5 characteristics that differentiate computers?
  - Can some processors compute things that others can't? No*








How do we evaluate computer architectures?

- Think of 5 characteristics that differentiate computers?

1. architecture / ISA ← O/S
2. memory size / capacity or storage capacity
   address space / word length
3. performance / parallelism ←
4. power consumption
5. price
6. size / portability
7. reliability

Two notions of performance

Aircraft	DC to Paris	Passengers
747	6 hours	500
Concorde	3 hours	125

- 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

x is 2.15 times faster than y

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

~~x is 115% faster~~

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

- If we are primarily concerned with latency,

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

- If we are primarily concerned with throughput,

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

CPU performance

- The obvious metric: how long does it take to run a test program? This depends upon three factors:

1. The number of *dynamic* instructions *N* in the program
   - Executing more instructions tends to take longer.

2. The kind of instructions in the program
   - 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)

~~freq~~

CPU time = Instructions executed × CPI × Clock cycle time

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

iron law of cpu perf

The three components of CPU performance

- Instructions executed:
  - the dynamic instruction count (#instructions actually executed)
  - not the (static) number of lines of code
- Average Cycles per instruction:
  - function of the machine and program
    - CPI(floating-point operations) > CPI(integer operations)
    - Improved processor may execute same instructions in fewer cycles
  - Single-cycle machine: each instruction takes 1 cycle (CPI = 1)
    - CPI can be > 1 due to memory stalls and slow instructions
    - CPI can be < 1 on superscalar machines
- Clock cycle time: 1 cycle = minimum time it takes the CPU to do any work
  - clock cycle time = 1/ clock frequency
  - 500MHz processor has a cycle time of 2ns (nanoseconds)
  - 2GHz (2000MHz) CPU has a cycle time of just 0.5ns
  - higher frequency is usually better

Execution time, again

CPU time = Instructions executed × CPI × Clock cycle time

"architecture" "micro architecture"

- Make things faster 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

- Often easy to reduce one component by increasing another

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

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.

$$\text{CPU time}_{AMD,P} = \text{Instructions}_P * \text{CPI}_{AMD,P} * \text{Cycle time}_{AMD}$$
$$= I \times 1.2 \times \frac{1\ s}{.8 \times 10^9\ \text{cycles}} =$$
$$= \frac{1.2}{.8 \times 10^9} I = \frac{12}{8 \times 10^9} I = 1.5 \times 10^{-9} I$$

$$\text{CPU time}_{P3,P} = \text{Instructions}_P * \text{CPI}_{P3,P} * \text{Cycle time}_{P3}$$
$$= I \times 1.5 \times \frac{1\ s}{1 \times 10^9\ \text{cycles}} =$$
$$= 1.5 \times 10^{-9}\ I\ s$$

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

Cycle time is equal between both processors