Introduction to HPCA

Introduction

1. How do modern processor cores work?
2. How do cores access memory?
3. How do we combine cores into a multicore chip?

What is Computer Architecture?

1. Architecture: Designing a building that is well-suited for its purpose
   - Different requirements for apartments vs houses
2. Computer Architecture: Designing a computer that is well-suited for its purpose
   - Different requirements for desktop, laptop, phone

Why Do We Need Computer Architectures?

1. Improve performance
   - Speed, battery life, size, weight, energy efficiency
2. Improve abilities
   - 3D graphics, developer support, security
3. Translate improvements in fabrication technology and circuit design into faster, lighter, cheaper, and more secure designs

Computer Architecture Quiz

1. Computer architecture is about:
   - How to build faster computers
   - How to design more energy-efficient computers
   - How to build computers that fit better into buildings

Computer Architecture & Technology Trends

1. If we design with current technology and parts, our computer will be obsolete by the time it goes to market due to improvements in technology
2. Must anticipate future technology and what will be available to produce future computers that take advantage of current technology

Moore's Law

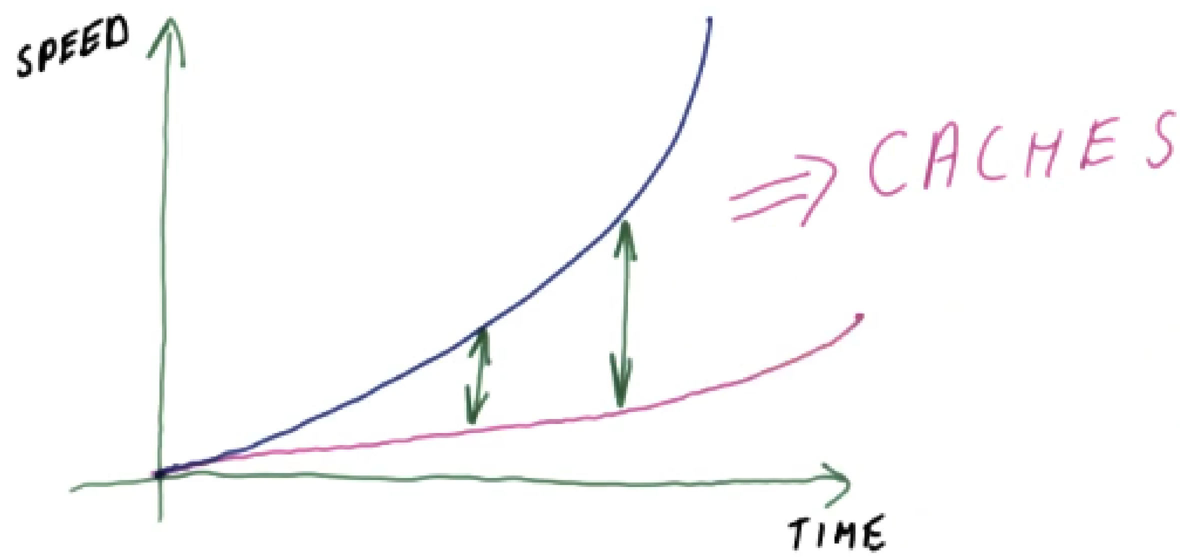
1. Moore's Law: Every 18-24 months, 2x transistors on same chip area
   - Computer architects attempt to double processor speed every 18-24 months
   - Computer architects attempt to halve energy/operation every 18-24 months
   - Computer architects attempt to double memory capacity every 18-24 months
2. Distinction between what technology is expected to do and what architects are expected to do with that technology

Speed Doubling Quiz

1. Trains: 1971 record was 380 km/h
   - In 2007, doubling every 2 years, this would result in a speed of 99614720 km/h
   - 380 * 2 ^ ((2007 - 1971) / 2)
   - Voyager is fastest manmade vehicle at 62000 km/h
2. Computers have actually doubled in speed every two years since 1971

Memory Wall

1. Instructions/second -> 2x every 2 years
2. Memory capacity -> 2x every 2 years
3. Memory latency -> 1.1x every 2 years
4. Gap between memory and processor performance is called the memory wall
    - Use caches (which are fast) to avoid going to slow main memory



Memory Wall

Processor Speed, Cost, and Power

1. Typically talk about instructions/second of a processor
2. Also need to consider cost for applications like refigerators
3. Power consumption results in longer battery life
4. These are tradeoffs that need to be considered for specific application

Speed	Cost	Power
2x	1x	1x
1.1x	0.5x	0.5x

Speed vs Power vs Weight vs Cost Quiz

MODEL	PERF	BAT. LIFE	WEIGHT	COST
Crawlium	0.1	30 hours	0.5 oz	$10
Slowium	0.5	15 hours	2 oz	$30
Laptium	1	5 hours	4 oz	$100
Fastium	1.5	1 hour	1 lb	$200
Hotium	2	15 min	3 lb	$500
Burnium	4	2 min	20 lb	$5000

1. Burnium, although fastest, would make the worst laptop battery due to high cost, high weight, and short battery life

Power Consumption

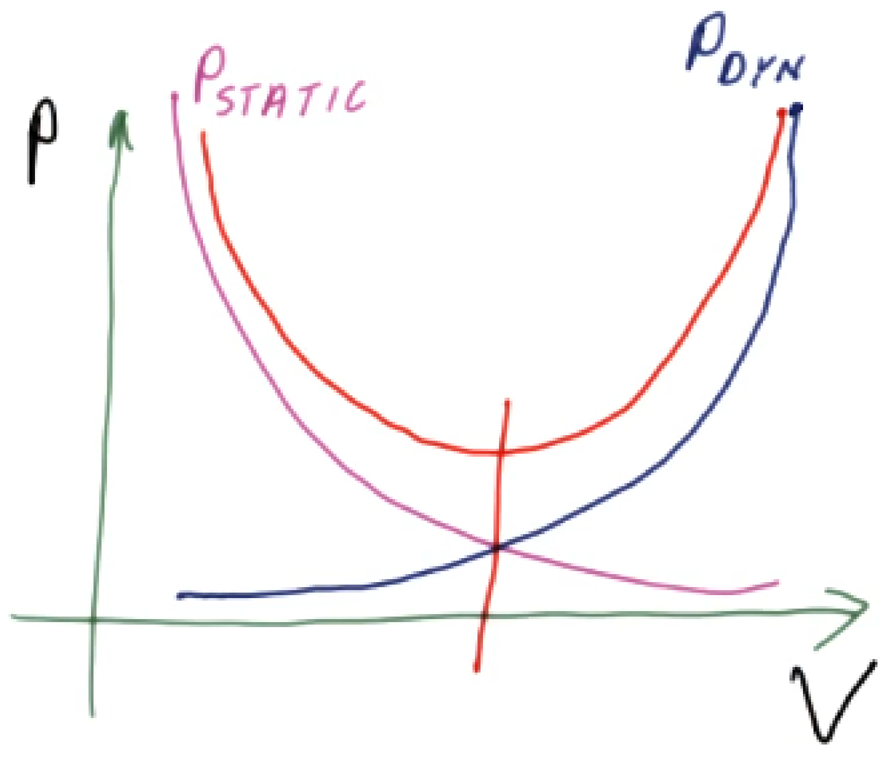
1. Dynamic power: Consumed by activity in a circuit
2. Static power: Consumed when a circuit is powered on but idle

Active Power

1. P = 0.5 * C * V ˆ 2 * f * alpha
    - C: Capacitance (proportional to chip area)
    - V: Voltage (Power supply)
    - f: Clock frequency
    - Alpha: Activity factor (some percentage of transistors are actually active)
2. Lowering voltage is most important in getting improvements in power usage

Static Power

1. Static power prevents us from lowering voltage too much
2. Leakage: Expending power even when a transistor is off
    - Transistors act like valves; low voltage is like a leaky valve
    - Lowering voltage increases leakage



Leakage

Active Power Quiz

f	v
1.8	0.9
2.0	1.0
2.2	1.1
2.4	1.2
2.6	1.3
2.8	1.4
3.0	1.5

1. What is P for the most power efficient setting given P = 30W at 1.0V, 2 GHz
    - Peff = 30 * (0.9/1) ˆ 2 * (1.8/2) = 21.9 W
2. What is P for the highest performance setting?
    - Pperf = 30 * (3/2) ˆ 2 * (1.5/1) = 101.3 W

Fabrication Cost

1. Chips are manufactured by printing layers onto a silicon wafer
    - The cost of the wafer and manufacturing process is relatively fixed, so the ability to decrease the footprint of the chip will decrease overall cost
    - If a chip doesn't work, we have to throw it away

Fabrication Yield

1. Yield = Working chips / chips on wafer
2. Wafer is expected to have some number of defects
    - The smaller the chips are, the fewer bad chips there will be
3. Assume a wafer has 2 defects
    - With small chips, 62/64 may work (97%)
    - With large chips, 6/8 may work (75%)

Fabrication Cost Example

1. Assumptions:
    - $5000/wafer
    - Small (400/wafer)
    - Large (96/wafer)
    - Huge (20/wafer)
    - 10 defects per wafer
2. Small wafer: 5000 / (400 - 10) = $12.80
3. Large wafer: 5000 / (96 - 10) = $58.20 (5x cost for 4x size of small)
4. Huge wafer: 5000 / (20 - 9) = $454.55 (9x cost for 4x size of large)
    - Perhaps both defects are on the same chip, so yield is 9
5. Moore's Law
    - Smaller chips -> Reduced cost
    - Same area -> Faster for same cost

Manufacturing Cost Quiz

1. 1mmˆ2 processor for watches (WP)
2. 100mmˆ2 processor for laptops (LP)
3. Because WP is smaller and will give better yield, cost(LP) > 100 * cost(WP)

Conclusion

1. Goal of computer architecture is to design a computer that is better suited for its intended use
    - Need to develop metrics to discuss our intuitive notion of "better"