Why should I learn computer architecture?
Reason #1: It’s fun

- Moore’s Law means the field is always radically changing
  - where else do you get an exponentially larger number of legos to play with every year?
- New application domains lead to totally new designs
  - GPUs
  - Phones
  - Data Center
  - Wearable
  - Implantable processors
  - Quantum, Biological, etc..
- CS ideas are increasingly applicable to hardware design
- Making things faster, smaller, more energy efficient is a rush
Reason #2: Performance Matters

100,000 computers + your code + 2X faster
= 50,000 computers saved
= 1 MW of electricity saved

1 phone + your code + 2X faster
= “fast enough”
= runs on 4 M more ipads

How can you speed up code if you don’t know how a computer works?
Reason #3

Great computer scientists *know the whole stack.*

Which of these people didn’t know how a computer works?
Reason #4

Employers want employees that are generalists and *know the whole stack*. Who knows what problems you might end up having to innovate on?
What we will learn in this class

- **Basic architecture:**
  - Instruction Sets
  - Performance Analysis
  - Pipelining
  - Caches
  - Virtual Memory
  - In-order processors
  - How to build your own all of the above.

- **Advanced Topics:**
  - Multicore
  - Data centers
  - Mobile Processors
  - GPUs
  - Out-of-order Processors
  - How x86 / ARM / NVidia combines all of the above