What is Computer Architecture?

Architecture: hardware interface to compilers & OS

- instruction set design
- software conventions
- some portions of the implementation

Organization: high level implementation

- components
- connections

Why separate architecture & implementation? (IBM 360)

- many implementations for same architecture
    - includes different process technologies
- share software development, binary compatibility
- 3rd party software if open architecture

Why have different architectures?

Principles of Computer Design

Design for the common case & make it fast

- Amdahl's Law defines speedup due to particular design improvement

$$\text{speedup} = \frac{\text{executionTime}_{\text{without}}}{\text{executionTime}_{\text{with}}}$$

$$\text{execTime}_{\text{with}} = \text{execTime}_{\text{wo}} \times \left\langle (1 - \text{frac}_{\text{with}}) + \frac{\text{frac}_{\text{with}}}{\text{speedup}_{\text{with}}} \right\rangle$$

Its corollaries

- smaller is faster
- keep it simple, stupid (the KISS principle)
- good design demands compromise
- understand the requirements of the application area

Principles of Computer Design

Design to withstand changes in technology (1996 figures)

- Performance: 55% per year (RISC; Intel little less)
- IC technology
  - transistor density: 50% per year
  - die size: 10% to 25% per year
  - combined effect: 60% to 80% per year
- DRAM technology
  - density: 60% per year
  - cycle time: 30% in ten years
- Magnetic disk technology:
  - density: 50% per year
  - access time: 30% in ten years

Don’t tie the architecture to the technology