What is Computer Architecture?

Architecture:

- Interface between the hardware & the lowest levels of software (compilers & OS)
  - model of the hardware
  - contract between the hardware & software
- An architecture is comprised of:
  - instruction set design
  - where data is located
  - software conventions
  - some portions of the microarchitecture

Microarchitecture (organization, implementation): high level implementation

- components
- connections

Architecture & Implementation

Why separate architecture & implementation?

Architecture & Implementation

Why have different architectures?

Principles of Computer Design

Design for the common case & make it fast

- Amdahl's Law quantifies the performance benefit of introducing new features

- Performance gain due to the new feature is limited by the fraction of time the feature is used

$$\text{exec Time}_{with} = \text{exec Time}_{wo} \times \left((1 - \text{frac}_{with}) + \frac{\text{frac}_{with}}{\text{speedup}} \right)$$

$$\text{speedup} = \frac{\text{execution Time}_{without}}{\text{execution Time}_{with}}$$

Principles of Computer Design

Its corollaries

- Smaller is faster

- Keep it simple, stupid

- Good design demands compromise

- Understand the requirements of the application area

Principles of Computer Design

Design to withstand changes in technology (2005 figures)
- performance: 55% per year but falling
- IC technology

  transistor density: 25% denser each year

  die size: about the same each year

  transistor count of combined effect: 25% increase per year
- DRAM technology

  density: 30% denser each year

  cycle time: 30% shorter each year

  bandwidth increases 2X faster than latency decreases
- magnetic disk technology:

  density: 100% denser each year

  access time: 30% faster in ten years

Don't tie the architecture to the technology!