Computer Architecture

Virendra Singh
Associate Professor
Computer Architecture and Dependable Systems Lab
Department of Electrical Engineering
Indian Institute of Technology Bombay
http://www.ee.iitb.ac.in/~viren/
E-mail: viren@ee.iitb.ac.in

CP-226: Computer Architecture

Lecture 1 (25 Jan 2013)

CADSL

Acknowledgement

- Prof. Kewal Saluja, Univ. of Wisconsin Madison
- Prof. Mikko Lipasti, Univ. of Wisconsin Madison
- Prof. Vishwani Agrawal, Auburn Univ.
- Prof. Erik Larsson, Lund Univ.
- Prof. Sarita Advi, UIUC
- Prof. Matthew Jacob, IISc
- Prof. Hideo Fujiwara, NAIST

Computer Architecture

- Instruction Set Architecture (IBM 360)
  - *… the attributes of a [computing] system as seen by the programmer. I.e. the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls, the logic design, and the physical implementation. -- Amdahl, Blaauw, & Brooks, 1964*
- Machine Organization (microarchitecture)
  - ALUS, Buses, Caches, Memories, etc.
- Machine Implementation (realization)
  - Gates, cells, transistors, wires

Running Program on Processor

$$\text{Processor Performance} = \frac{\text{Time}}{\text{Program}}$$

$$= \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Time}}{\text{Instruction}}$$

(code size)

Architecture

Compiler Designer

Computer Architecture

- Instruction Set Architecture (IBM 360)
  - *… the attributes of a [computing] system as seen by the programmer. I.e. the conceptual structure and functional behavior, as distinct from the organization of the data flows and controls, the logic design, and the physical implementation. -- Amdahl, Blaaw, & Brooks, 1964*

Running Program on Processor

$$\text{Processor Performance} = \frac{\text{Time}}{\text{Program}}$$

$$= \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Time}}{\text{Cycle}}$$

(code size) (CPI)

Architecture --> Implementation

Compiler Designer **Processor Designer**

Running Program on Processor

$$\text{Processor Performance} = \frac{\text{Time}}{\text{Program}}$$

$$= \frac{\text{Instructions}}{\text{Program}} \times \frac{\text{Cycles}}{\text{Instruction}} \times \frac{\text{Time}}{\text{Cycle}}$$

(code size) (CPI) (cycle time)

Architecture --> Implementation --> Realization

Compiler Designer **Processor Designer** **Chip Designer**

Iron Law

- Instructions/Program
  - Instructions executed, not static code size
  - Determined by algorithm, compiler, ISA
- Cycles/Instruction
  - Determined by ISA and CPU organization
  - Overlap among instructions reduces this term
- Time/cycle
  - Determined by technology, organization, clever circuit design

Computer Architecture's Changing Definition

- 1950s to 1960s:
  Computer Architecture Course = Computer Arithmetic
- 1970s to mid 1980s:
  Computer Architecture Course = Instruction Set Design, especially ISA appropriate for compilers
- 1990s onwards:
  Computer Architecture Course = Design of CPU (Processor Microarchitecture), memory system, I/O system, Multiprocessors

Thank You