CSE301 – Computer Organization

Lecture 1 Notes

Architecture vs. Organization

Architecture

- The **conceptual (functional) design** of the computer system as seen by the programmer.
- Describes **what the system does**.
- Examples:
  - Instruction Set Architecture (ISA)
  - Data types
  - Registers
  - Addressing modes
  - I/O mechanisms visible to software

Organization

- The **implementation details** of the architecture.
- Describes **how the system is built internally**.
- Examples:
  - Datapath, control signals
  - Memory hierarchy
  - Pipelining
  - Physical arrangement of components

Critical Thinking 💡

- When we say *single-cycle* or *multi-cycle*, is that an **architecture** or an **organization**?
- Is MIPS considered an **architecture** or an **organization**?

Structure vs. Function

Structure

- How the components are related to each other.

Function

- The operation of individual components.
- Four core functions of any computer:
  1. Data Processing
  2. Data Storage
  3. Data Movement
  4. Control

Computer Evolution and Performance

Parameters affecting performance:

1. System clock speed (f)
2. Instruction count (I_c)
3. Cycles per instruction (CPI)

Since CPI varies across instructions, we use the **average CPI**:

$$\text{CPI}_{\text{avg}} = \frac{\sum \text{CPI}_x \cdot I_x}{I_c}$$

Performance Metrics

1. **Instructions per second (IPS, R_i)**

$$T_i = \frac{\text{CPI}}{f}, R_i = \frac{1}{T_i}\text{instructions/sec}$$

2. **Million instructions per second (MIPS, (R_m))**

$$R_m = \frac{R_i}{10^6}$$

3. **Programs per second (PPS, R_p)**

$$T_p = T_i \cdot I_c, R_p = \frac{1}{T_p}$$

Info:
These performance metrics are valid for analyzing a **single system**.
They cannot be directly used to compare **different systems** because of differences in manufacturers, architectures, and instruction sets.
To compare across systems, we use **benchmark programs**.

Benchmark Programs

- Written in high-level language.
- Designed in different styles and widely distributed.
- Used to compare machines with a common workload.

Definitions:

- T_{ref} = runtime on the **reference machine**
- T_{sut} = runtime on the **system under test**

Relative speed:

$$r_x = \frac{T_{\text{ref}}}{T_{\text{sut}}}$$

Geometric mean speed:

$$r_g = \left(\prod_i^n r_{x,i}\right)^{\frac{1}{n}}$$

Info:
The larger the r_g, the faster the **system under test** compared to the reference machine.

Amdahl's Law

If a program takes time (T) on a single processor and only a fraction (f) can be executed in parallel on (N) processors:

$$\text{SpeedUp} = \frac{1}{(1 - f) + \frac{f}{N}}$$

Maximum speedup (as $N \to \infty$):

$$\text{SpeedUp}_{\text{max}} = \frac{1}{1 - f}$$