Computer Architecture

Some questions & answers

Prof. Nizamettin AYDIN, PhD
naydin@yildiz.edu.tr

http://www3.yildiz.edu.tr/~naydin

AcA-00-Introduction

- What is Computer Organization?
- What is Computer Architecture?
- Brief history of computing systems
- What are the classes of computers?
- What are the constituents of a computer?
- What are the constituents of a CPU?
- What are the constituents of a Control Unit?
- What are the levels of program code?
- What is a program?
- How do you describe The Computer Level Hierarchy?

AcA-01-Fundamentals

- What is informatics?
- What is data?
- What is information?
- What is knowledge?
- What is a system?
- What is an information system?
- What are to components that implement information system?
- What is a computing system?
- What is a digital system?
- What is a signal?

AcA-01-Fundamentals

- Compare analog and digital signals
- Why do we sample a signal?
- Why do we quantize a signal?
- Describe continuous, discrete, and digital signals.
- Describe the process of obtaining digital signals
- What is sampling theorem?
- What are the fundamental data types represented in a computing system?
- Boolean algebra, digital circuit functions

AcA-02-InstructionSet-rev

- What is an instruction?
- What is instruction set?
- What is meant by Instruction Set Architecture? Explain
- What are the general instruction types in a computing system?
- What are the elements of an instruction?
- Classify instruction set in terms of number of operands.
- What types of operand an instruction can take?
- What is Big/Little Endian?

AcA-03-Performance

- List common performance metrics used in a computing system.
- Describe the Forces on Computer Architecture.
- What type of parallelisms exist in a computing system?
- What are the classes of computers?
- What is Flynn’s Taxonomy?
- Power consumption in a processor
- How to reduce power consumption?
- What are the basic performance metrics?
- What are the measurement tools?
- What is Amdahl’ law?

AcA-04-MemoryHierarchy

- What is Memory Hierarchy?
- What is the Principle of Locality?
- What is a Cache?
- Why a Cache Memory is used?
- How many cash types exist?
- What is Main Memory
- What is Virtual Memory
- Why a Virtual Memory is used?
- Classify memory types
- Differences between SRAM and DRAM?
- Memory organization
- Virtual machines

AcA-05-Instruction-Level Parallelism

- Explain Instruction-Level Parallelism
- What is pipelining?
- What is main constraint in paralellism?
- How many dependences exist?
- What are data hazards?
- What techniques exist to avoid dependences
- What is purpose of Tomasulo's algorithm?
- Compare the processors interms of pipelining

AcA-06-Data-Level Parallelism

- What are the classes of parallelizm? Briefly explain
- Classify computers in terms of the Data-Level Parallelism
- Brifly describe Vector Architecture
- How Vector Processors work? Explain with an example
- Brifly describe Graphics Processing Units Architecture
- What is heterogen computing system?

AcA-06-Data-Level Parallelism

- Brifly describe NVIDIA Instruction Set Architecture
- What are the Challenges for the GPU programmer
- Compare Graphics Processing Units and vector Architectures
- Dependences in Loop Level Parallelism
- How to find dependences in Loop Level Parallelism

Computer Architecture Formulas

1. *CPU time* =
   Instruction count × Clock cycles per instruction × Clock cycle time

2. X is *n* times faster than Y:
   n = Execution time$_Y$ / Execution time$_X$ = Performance$_X$ / Performance$_Y$

3. *Amdahl's Law*:
   $$\text{Speedup}_{\text{overall}} = \frac{\text{Execution time}_{\text{old}}}{\text{Execution time}_{\text{new}}} = \frac{1}{(1 - \text{Fraction}_{\text{enhanced}}) + \frac{\text{Fraction}_{\text{enhanced}}}{\text{Speedup}_{\text{enhanced}}}}$$

4. *Energy*$_{\text{dynamic}}$ ∝ 1/2 × Capacitive load × Voltage^2

5. *Power*$_{\text{dynamic}}$ ∝ 1/2 × Capacitive load × Voltage^2 × Frequency switched

6. *Power*$_{\text{static}}$ ∝ Current$_{\text{static}}$ × Voltage

7. *Availability* = Mean time to fail / (Mean time to fail + Mean time to repair)

8. *Die yield* = Wafer yield × 1 / (1 + Defects per unit area × Die area)^N
   where Wafer yield accounts for wafers that are so bad they need not be tested and *N* is a parameter called the process-complexity factor, a measure of manufacturing difficulty. *N* ranges from 11.5 to 15.5 in 2011.

Computer Architecture Formulas

9. *Means—arithmetic (AM), weighted arithmetic (WAM), and geometric (GM):*
   $$\text{AM} = \frac{1}{n}\sum_{i=1}^{n}\text{Time}_i\,, \qquad \text{WAM} = \sum_{i=1}^{n}\text{Weight}_i \times \text{Time}_i\,, \qquad \text{GM} = \sqrt[n]{\prod_{i=1}^{n}\text{Time}_i}$$
   where Time$_i$ is the execution time for the *i*th program of a total of *n* in the workload, Weight$_i$ is the weighting of the *i*th program in the workload.

10. *Average memory-access time* = Hit time + Miss rate × Miss penalty

11. *Misses per instruction* = Miss rate × Memory access per instruction

12. *Cache index size*: 2^{index} = Cache size /(Block size × Set associativity)

13. $Power\ Utilization\ Effectiveness\ (PUE)$ of a Warehouse Scale Computer $= \frac{\text{Total Faciliy Power}}{\text{IT Equipment Power}}$

Rules of Thumb

1. Amdahl/Case Rule:
   - A balanced computer system needs about 1 MB of main memory capacity and 1 megabit per second of I/O bandwidth per MIPS of CPU performance.
2. 90/10 Locality Rule:
   - A program executes about 90% of its instructions in 10% of its code.
3. Bandwidth Rule:
   - Bandwidth grows by at least the square of the improvement in latency.

Rules of Thumb

4. 2:1 Cache Rule:
   - The miss rate of a direct-mapped cache of size *N* is about the same as a two-way set-associative cache of size *N*/2.
5. Dependability Rule:
   - Design with no single point of failure.
6. Watt-Year Rule:
   - The fully burdened cost of a Watt per year in a Warehouse Scale Computer in North America in 2011, including the cost of amortizing the power and cooling infrastructure, is about $2.