

Midterm1 Preparation

CS 3853 Computer Architecture, Spring 2024

Wei Wang

Computer Science
University of Texas at San Antonio

Goals and Topics

- ▶ The goal is to help you systematically review the basic knowledge in Computer Architecture.
 - Only basic knowledge is tested.
 - No trick questions.
- ▶ Topics for this exam:
 - Introduction
 - Instruction Set Architecture
 - Computer Arithmetic
 - Performance Metrics

Location, Time and Logistics

- ▶ Feb 21st 2024, Wednesday, online, in-class
- ▶ close-book, close-notes
- ▶ You can bring a calculator, but no cellphones.
 - The math is simple, you probably won't need a calculator.
- ▶ Do not waste time – if you stuck on a problem, move forward and revisit the problem later.
- ▶ Do not leave any question unanswered.
- ▶ The exam have Conceptual questions and Problems.
 - The problems will be similar as those in the assignments.

Materials to Review

- ▶ Slides, all questions are from slides
- ▶ Assignment 1 and 2.
- ▶ You can check out the textbooks, but it is not required.

Introduction

- ▶ The three topics in Computer Architecture
 - ISA, micro-arch and system architecture.
- ▶ The definitions of Moore's Law and Dennard Scaling.
- ▶ The impact of the failure of Dennard Scaling.
- ▶ Design metrics for computer architectures:
 - Performance, cost, availability and power dissipation
- ▶ The definition of Von Neumann architecture and Harvard Architecture.

Instruction Set Architecture

- ▶ Be able to read basic instructions with source and destination operands.
 - Know the basic syntax.
- ▶ Four source devices for operands, and their examples.
 - Stack, x86 FP instructions.
 - Accumulator, x86 multiplication/division instructions.
 - Register-Register, most RISC ISAs.
 - Register-Memory, most CISC ISAs.
- ▶ Know all addressing modes and their examples from the slides.
 - Be able to write and recognize an addressing mode.
 - Especially the example for addressing elements in two-dimensional arrays.
- ▶ Fixed-length and variable length ISAs.
 - Their definitions, advantages and disadvantages.

Instruction Set Architecture cont'd

► CISC and RISC

- The full names of CISC and RISC.
- Examples ISAs of CISC and RISC.
- The features of CISC and RISC (slide 51 from the ISA lecture)
- CISC vs. RISC
 - CISC has better programmability, smaller code sizes
 - RISC is easier to implement, thus needs fewer transistors and has better energy efficiency.
- Modern processors mostly use RISC internally.

► SIMD instructions, definition and examples

- Examples: MMX, SSE, AVX, 3DNow

Computer Arithmetic

- ▶ Remember the binary representations of numbers 0 to 15.
- ▶ Two's complement encoding.
 - Be able to write two's complement encoding given a decimal number.
 - Why two's complement?
 - The problem is similar to the one from Assignment 2.
- ▶ Floating point encoding.
 - Be able to write 32-bit encoding given a binary real number.
 - The problem is similar to the one from Assignment 2.
- ▶ Know the four logic gates, and know that logic gates are used to construct functional units.

Performance Metrics

- ▶ Why use benchmarks and simulators?
- ▶ Know the definition of MIPS, MFLOPS, CPI and IPC.
- ▶ Problems, very similar to those in Assignment 2.
 - Be able to compute average, weighted average, geometric and harmonic mean given some execution times.
 - Amdahl's Law
 - ▶ Review the equations and examples in the slides for Amdahl's Law.
 - ▶ You should be able to compute the overall speedup given a percentage of enhance-able part and a speedup.
- ▶ Know the relationship of instructions per program, cycles per instruction (CPI) and cycles per second (frequency).

Example Questions

- ▶ Conceptual question: What is the impact of the failure of Dennard scaling?
 - Answer: The growth of CPU frequency has stopped, and the growth of single core performance has significantly slowed down.
- ▶ Problem: If the 60% of an application can be improved by a speedup of 6, which is the overall speedup for the whole application?
 - Answer: $Spd = \frac{1}{(1-60\%) + \frac{60\%}{6}} = \frac{1}{40\% + 10\%} = 2.$