COMPSYS 304: Computer Architecture

Lecture Notes

Nicholas Russell

2024-03-09

Table of contents

Introduction **2**
Organisation of the Notes . . . 2
How I Use These Notes . . . 2

I Lectures 1 - 3 **4**

1 Introduction & Course Overview **5**
1.1 Background . . . 5
1.1.1 What do we mean from *Performance*? . . . 5
1.1.2 Intel X86 Processor from 1978 to 2018: . . . 5
1.2 Course Details . . . 5
1.2.1 Learning outcomes . . . 5
1.2.2 Course Overview (learning outcomes) . . . 6
1.2.3 Assessment . . . 6

2 Basics of Computer Architecture **7**
2.1 What is Computer Architecture? . . . 7
2.2 Review: Memory Organization . . . 7
2.3 An Overview of Memory Speed vs. CPU Speed . . . 7
2.4 Processor and Memory Hierarchy . . . 8
2.5 Example of Intel Processors . . . 8

3 Instruction set Architecture (ISA) **9**

Introduction

These notes are compiled from the lectures of COMPSYS 304. They are intended as a personal reference to help with assignments, exam preparation, and understanding key concepts in Computer Architecture.

The notes are organised by lecture and cover a wide range of topics from basic computer architecture principles to more advanced subjects like MIPS implementation and performance analysis. Feel free to add personal insights, additional readings, or questions as you review this material.

Organisation of the Notes

The notes are divided by lecture content, with each section corresponding to a specific set of topics. Here's how they're organized:

- **Lecture 1-3: Basics of Computer Architecture**
  Key topics include Instruction Set Architecture (ISA), memory hierarchy, and basic CPU organisation.
- **Lecture 4-6: MIPS Architecture**
  Covers the MIPS instruction set, control flow, and subroutine handling.
- **Lecture 7-9: CPU Implementation**
  Focuses on different methods for implementing CPUs and the trade-offs involved.
- **Lecture 10-12: Digital Circuits and Datapath Design**
  Reviews fundamental digital circuit concepts and discusses the design of a MIPS datapath.
- **Lecture 13-15: Performance Analysis**
  Provides an in-depth look at how different design choices impact CPU performance.

How I Use These Notes

These notes are a living document, and I intend to update them as I gain a deeper understanding of the material. Here's how I use them:

- **Quick Reference**: For quick lookups, the Table of Contents will help me jump directly to the relevant section.
- **In-Depth Study**: For exam preparation, I'll revisit each section, ensuring I understand each concept before moving on.
- **Personal Insights**: I'll be adding my own thoughts, additional notes from readings, and potential questions for further study.

I might also include exercises or practical examples that help solidify my understanding of the more

complex topics.

Part I

Lectures 1 - 3

Chapter 1

Introduction & Course Overview

This chapter provides an introduction to the course, some background, and an overview of the course itself.

1.1 Background

In the last six decades, computer technology has made incredible progress due to innovations in both **semiconductor technology** and computer **architecture**.

1.1.1 What do we mean from *Performance*?

The relative performance can be measured by standard benchmarks, which depend on the specific target applications.

Performance depends on clock frequency and other factors, as discussed more in later chapters.

1.1.2 Intel X86 Processor from 1978 to 2018:

- **Intel 8086** - (1978): 1 core, 1 W, 5 - 10 MHz
- **Core i7-8086K** - (2018): 6 cores, 95 W, 4 GHz

1.2 Course Details

1.2.1 Learning outcomes

The main learning outcomes of this course are:

- **To understand the basics of modern computer architectures and quantitative principles of computer design in order to develop a conceptual understanding of issues involved in designing a high performance computer system.**
- **To use and apply this knowledge to design computer systems or select computers for specific tasks. This course will give you an understanding of the effects of design decisions on system performance and makes you a well-informed consumer in addition to a processor designer.**

Recommended Textbook:

- David A. Patterson and John L. Hennessey, Computer Organization and Design: The Hardware/Software Interface, Fifth edition, 2013 by Elsevier/Morgan Kaufmann Publishers (or 3rd or 4th editions).
- Lecture notes provided on canvas (these will be summarised in these notes).

1.2.2 Course Overview (learning outcomes)

Part 1: *In this part you will learn to:*

- Design and evaluate the instruction set architectures (both RISC and CISC) and how it can be related to the hardware/software interface in a computer system with a quick review of assembly programming.
- Understand different processor implementation methods including the basic single-cycle implementation and how it can be extended to a multi-cycle, pipelined and superscalar implementations.
- Understand performance evaluation techniques and their relation to the target applications and the processor workload.

Part 2: *In this part you will learn to:*

- Understand the memory hierarchy in a modern computer system and its impact on the performance of the system. This includes physical and virtual memory systems and basics of cache memories.
- Understand some basic principles of parallel computing using special topics in this course (more advanced materials covered in some elective courses).

1.2.3 Assessment

Three assignments and one test in addition to the final exam. The first assignment is mainly on the instruction set architecture design, hardware/software interfacing and review of assembly programming. The second assignment is related to processor implementation and performance issues. The third assignment is related to memory hierarchy system and multiprocessing.

The test only covers the first part of the course. The final exam covers the whole course.

- **Assignment 1: 8% due Fri. 9 August.**
- **Assignment 2: 7% due Fri. 23 August.**
- **Assignment 3: 15% due Fri. 4 October.**
- **Test: 20% (in week 7, Wednesday 11 September)**
- **Final exam: 50%**

Chapter 2

Basics of Computer Architecture

This chapter begins the journey into the content of this course.

2.1 What is Computer Architecture?

- **ISA:** The programmer-visible instructions — ISA serves as the boundary between the hardware and software.
- **Organization:** high-level aspects of computer design (e.g. the internal design of the CPU, the memory system, the interconnection structure, …)
- **Hardware:** detailed logic design, circuit-level design, packaging technology, etc.

2.2 Review: Memory Organization

- Memory is viewed as a large, one-dimensional array, which has an address range.
- A memory address is an index into the array
- The index usually points to a byte of memory.
- To access a larger word (e.g., a 32-bit word or 4 bytes), four consecutive memory locations are accessed.

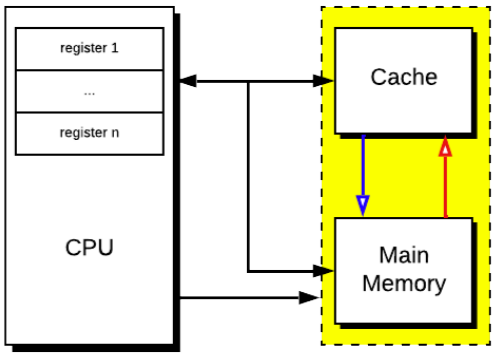
Memory access time is the amount of time required to read (usually one byte) or write data (usually one byte) from/to memory.

2.3 An Overview of Memory Speed vs. CPU Speed

The rate of memory speed increase has been less than processors speed increase

Fast memory is more expensive per bit than slow memory, so a **memory hierarchy** is often used to give a performance close to the fastest memory with a lower average cost per bit.

2.4 Processor and Memory Hierarchy



Processor Registers: The smallest and fastest memory for CPU.

Typically, 32 to 64 registers, each of them may be 32 or 64 bits with typical access time of *nanoseconds or less*

Cache Memory: Slower than registers.

Typically 8 to 256 Kbytes with an access time about a *few nanoseconds or less*

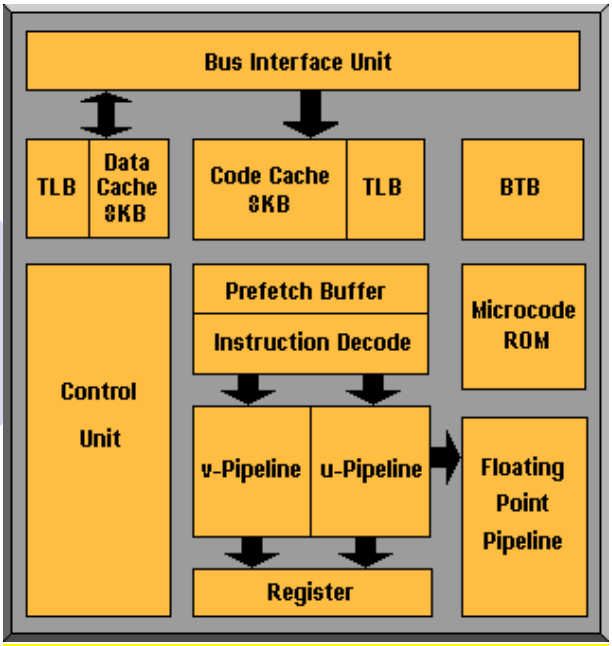
Main Memory: Slower than cache memory.

Hundreds of Mega bytes with an access time about *tens of nanoseconds*

2.5 Example of Intel Processors

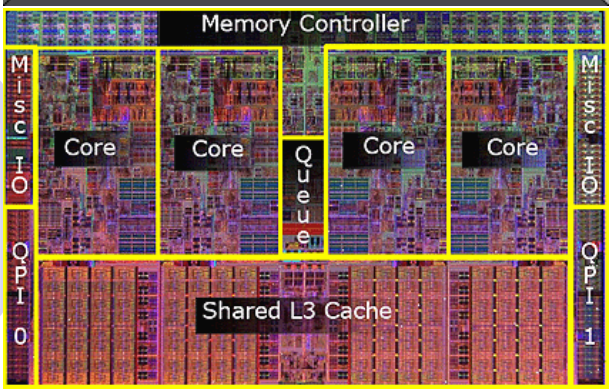
Pentium Processor Architecture (simplified)

An example of architectural innovations made by processor designers (to improve computer systems performance) (Source: Intel)



Intel Core i7

The Core i7 die and major components. (more than 700 million transistors in a die with area 263 mm^2 in 45 nm technology). (Source: Intel)



Chapter 3

Instruction set Architecture (ISA)

ISA is the interface between hardware and low-level software

Some modern instruction set architectures:

80x86, Itanium, PowerPC, **MIPS**, SPARC, HP, **ARM**, **RISC-V**