# Syllabus

## General Information

**CS1541 – Introduction to Computer Architecture**

Instructor: Yong Zhao, 532N

Contact: yong.zhao@scupi.cn

Teaching Assistant: Hejia Zhang

Contact: 2020141520183@stu.scu.edu.cn

## Course Meeting Times

Lectures:

Two session / week, Monday 8:15am - 11am, 1:50pm – 4:30pm

Office Hours:

Tuesday 9:30am -12:30 pm

Wednesday 2pm – 5:30pm and by appointment

## Overview

This course aims to provide the foundation for students to understand the modern eras of computer architecture (i.e., the single-core era, multi-core era, and accelerator era) and to apply these insights and principles to future computer designs. It is a study of the evolution of computer architecture and the factors influencing the design, performance, cost and security of hardware and software elements of computer systems. Topics include: performance, pipeline; instruction/task/data/thread level optimization, cache and virtual memory organizations; I/O and interrupts; in-order and out-of-order superscalar architectures; vector supercomputers; multithreaded architectures; symmetric multiprocessors; memory models and synchronization; embedded systems; parallel computers; GPU, and cloud computing.

This course also includes a course project to help students get hands-on experience on the design, implementation, testing, and evaluation of a computer system, and to better understand the concepts involved in computer architecture.

## Course Objectives

The goals of the course are to help the students understand the underlying architecture of computers, have basic knowledge of instruction set architectures, and look into details of the design and implementation of the key architectural components in modern computers, and learn the key concepts and approaches to improve the performance and efficiency of modern computers and processors, such as pipelining, ILP, caching, virtual memory hierarchy.

The course will prepare students for working in the computer engineering industry and can act as a footstep to more advanced material in graduate-level courses. This course can also provide a foundation for students interested in performance improvement, compilers, and operating systems; and it can provide system-level context for students interested in emerging technologies and artificial intelligence.

## Prerequisite

Digital Logic or equivalent

## Class Schedule

Lecture slides will be available for copying or posted on blackboard.

1. Introduction and review
2. Assembly language
3. Performance
4. Pipeline
5. Instruction Level Parallelism
6. Task/Data/Thread Level Parallelism
7. Memory hierarchy
8. Cache
9. Virtual memory
10. Networks
11. Security
12. Multiprocessing
13. GPU
14. Large Language Models

## Learning Outcome

At the end of the class, the students should be able to understand the key concepts related to computer architecture.

The students should also be able to:

- describe computer architecture concepts and mechanisms related to the design of modern processors, memories, and networks and explain how these concepts and mechanisms interact.
- apply this understanding to new computer architecture design problems within the context of balancing application requirements against technology constraints
- write concise yet comprehensive technical reports that describe designs implemented at the register-transfer-level, explain the testing strategy used to verify functionalities.
- evaluate various design alternatives and make compelling quantitative and/or qualitative comparisons.

The students will also have practical experience of basic architectural design and prototyping digital circuits though course projects and relevant tools.

## Grades

Grades will be based on homework, course project and final exam.

Homework and attendance: 30%

Course project (1 major project or 2 small projects): 30%

Final Exam: 40%

## Collaboration and Academic Honesty Policy

Individual work on all homework and examinations is required, Cheating and copying other students' homework/exam are strictly prohibited. Any violation of this policy will be treated severely.

Collaboration amongst students to understand the course material and to work on course projects is strongly encouraged, however each student should take on different/distinguishable responsibilities in the course projects.

## Course Reading Material

Patterson, D. A., and J. L. Hennessy. *Computer Architecture: A Quantitative Approach*, 6th ed.

This is the main textbook used in this course.

To review the basic material, you may also want to refer:

Hennessy, J. L., and D. A. Patterson. *Computer Organization and Design: The Hardware/Software Interface*, 6th ed.

Supplemental readings from selected papers may also be assigned throughout the semester.