Course Information

CST 250 Microcomputer Architecture and Programming

Catalog Description

Microcomputer architecture, instruction set, assembly language programming and debugging, I/O considerations, memory interface, peripherals and busses, exception/interrupt handling.
Credit Hours : 3

Prerequisites by Topic, Course

- CST 100; SER 232

Textbooks and Other Resources

- Computer Organization and Design By D. Patterson, J Hennessy, Morgan Kaufmann

Course Description and Goals

Welcome to CST 250! Think about all the things in the world that have a microprocessor in them. Your laptop, your refrigerator, car, TV, phone, pacemaker... This class is the study of these systems and the microprocessors that are the brains of these systems. That includes topics such as number systems, assembly language programming, organization and design of a microcomputer, and interfacing with sensors, A/D converters, etc.

This course introduces you to core, low-level, computing systems concepts including assembly programming and the organization of computing systems. We’ll look at tradeoffs in various contemporary architectures (ARM, Intel), practical limitations of computing systems, and some of the rationales behind their designs. Additionally, we’ll be interfacing a lot with other systems through sensors and controls, which is what real engineers do. Throughout the course we will work on multiple labs and micro projects some of which will come together as a final class project.

It has been my personal experience as a student and as a teacher for many years that we often think we have understood a concept when someone explains it to us but when we try to apply it to solve a problem or build a project, we realize that we really didn’t get it. This course is based on the pedagogy of active learning, where you apply your knowledge to solve problems and reflect on your learning. In that process you gain procedural knowledge and also gain and strengthen conceptual knowledge that is required in the context of your projects and activities. Be prepared to work hard. What you gain from this course will be proportional to how hard you are willing to work. Remember, a 3 credit-hour course entails 9 hours of work/study per week.

Major Topics Covered
...
Assignments 10%
Midterm 10%
Final Exam 10%

Course Outcomes

1. CO1 Students are able to identify problems in familiar and unfamiliar programs, and correct and improve the code.

   **Student Outcomes Supported**
   Software Engineering Practice, Technical Competence

2. CO2 Students are able to interface with different types of I/O devices with a range of interface protocols (serial, parallel, digital, analog).

   **Student Outcomes Supported**
   Technical Competence

3. CO3 Students are able to show (through drawings and short descriptions) how numbers and program statements are stored and operated on by the CPU.

   **Student Outcomes Supported**
   Technical Competence

4. CO4 Students are able to utilize the advanced features of the CPU (stacks, interrupts, advanced instructions) to accomplish complex tasks.

   **Student Outcomes Supported**
   Technical Competence

5. CO5 Students are able to work with other students to develop teamwork skills and an appreciation for alternate approaches to a problem.

   **Student Outcomes Supported**
   Teamwork

6. CO6 Students are able to compare different approaches/solutions to a specific assembly programming situation/problem and evaluate which one is better.

   **Student Outcomes Supported**
7. CO7 Students are able to define basic terminology related to computing (e.g. Program Counter, Stack, Algorithm)

   Student Outcomes Supported
   Technical Competence

8. CO8 Students are able to draw a block diagram of the main parts of a CPU and describe each part.

   Student Outcomes Supported
   Technical Competence

9. CO9 Students are able to generate original assembly code using the software development cycle (analyze problem, create algorithm, draw flowchart, write program, and debug program).

   Student Outcomes Supported
   Problem Solving, Technical Competence