CSC 230 Midterm Reading List (Summer 2023)

The midterm will take place **in-class on Monday June 12, 2023**. The duration of the exam will be 75 minutes. The following material will be covered (references are class notes slides posted on Birghtspace):

1. Slides 20 to 59 from "1_Csc230_Intro"
2. Slides 1 to 30 and 41 to 74 from"2_CSc230_Numeration_Systems" (focus will be decimal, binary and Hex)
3. All Slides from "3_CSc230_Assembly_Language"
4. Slides from 1 to 78 from "4_CSc230_Functions_in_AVR"

Go through some of the lab exercises and assignment questions related to these topics. A small subset of assembly instructions will be provided to you. The general topics are:

Computer Architecture

- Major components of a CPU
  - ALU, PC, CIR, Memory, Registers
- Memory:
  - byte addressable, word addressable
- Fetch-Decode-Execute cycle

Data Representation and Number Systems

- bits required to represent a range of numbers
  - alternately, given *n* bits, range of values that can be represented
- integer representation
  - unsigned
- base conversions: base 2, base 10, base 16

Assembly Language Programming

- PC and SREG
- Load and Store
- Branches and Jumps
- Arithmetic operations
- Logical operations
  - bit manipulations and Masks
- CALL and RET instructions
- Stack
- Assembler directives: .org, .cseg, .dseg, .equ, .include, .def

AVR Specifics

- Instruction encoding
- program memory and data memory
- instruction timing
- PORTS and Data Direction Registers

- I/O area
- Memory mapped I/O
- IN/OUT instructions vs LDS/STS

Midterm Exam Objectives

In order to successfully complete this midterm exam, you must be able to:

1. Define Computer Architecture and Computer Organization
    a. List and identify the various parts of a computer.
    b. Describe the internal parts of a CPU: ALU, registers, control unit, buses, input/output, etc.
    c. Explain von Neuman and Harvard models of a Computer System.
2. Count in decimal, hexadecimal and binary.
3. Convert between all combinations of decimal and hexadecimal and binary.
4. Explain how arithmetic operations produce overflow. Identify overflow in results.
5. Determine the range of integers available using a given integer representation and number of bits.
6. Store numbers in and read from memory using little and big endian conventions.
7. Determine the size of buses based on the capacity of a system.
8. Perform logical operations (And, Or, invert) on binary values.
9. Shift, rotate binary values
10. Apply Masks on bit sequences
11. Identify parts of the Fetch-Decode-Execute cycle.
    a. Describe the details of the Fetch-Decode-Execute cycle on a typical system.
12. Identify the opcode and operands of an AVR assembly language instruction
13. Determine the operation of an AVR assembly language program (i.e., trace through a program and determine what is stored in registers and/or memory at the end of the execution.)
    a. Determine the values in the V,Z,N,C bits of the status register after arithmetic, compare or logic operations.
14. Distinguish between addressing modes (e.g., Absolute, Immediate, Direct or Indirect) of an AVR assembly language program.
15. Complete an already started AVR assembly language program.
16. Write an entire AVR assembly language program that accesses data stored in both program and data memories.
17. Be able to convert control structures such as if, if/else, switch, while, do/while and for into AVR assembly language.
18. Explain purpose and differences of various AVR ATMega2560 memories such as SRAM and Flash.

19. Memory sizes and their mapping on various Address Space ranges
20. Be able to use common assembler directives.
21. Be able to use ports and port related instructions
22. Able to differentiate between memory-mapped I/O and channel (port)-mapped I/O.