COSE222 Computer Architecture
Final Project: 1st Milestone

Follow the synthesis & simulation tutorial linked on the class web. The example code in the tutorial will display a number 5 on the HEX0 7 Segment of DE0 Board.

The RISC-V (RV32I) System design on the class web implements only a few RV32I instructions: `add, addi, sub, and, or, sltu, lui, lw, sw, beq` and `jal`. Check out the Verilog source `(/RV32I_System/RV32I_CPU/rv32i_cpu.v)` to figure out its implemented instructions. (Look at the `main_dec` and `aludec` modules). It means that you are allowed to use those instructions in your software program for now.

The RISC-V System also has a peripheral device called GPIO (General-purpose Input/Output). GPIO is used to connect various I/Os and has several memory-mapped registers. LEDs, Switches, and 7 Segments on DE0 board are connected to our RISC-V System via GPIO. Check out the GPIO hardware design at `/RV32I_System/GPIO/GPIO.v` and decoder (for creating the memory-map) hardware design at `/RV32I_System/Decoder/Addr_Decoder.v`. Our RISC-V system has the following memory map shown in Table 1.

Table 1 Memory Map of Our Computer System

Components	Size	Base (Start) Address	End Address
Reserved			
GPIO	4KB	0xFFFF_2000	0xFFFF_2FFF
Reserved			
Timer Counter	4KB	0xFFFF_0000	0xFFFF_0FFF
Reserved			
Memory	8KB	0x0000_0000	0x0000_1FFF

Table 2 shows the GPIO registers and their allocated addresses. To display a number on HEX0, for example, you should write data to 0xFFFF_200C using `sw` instruction.

Table 2. GPIO Registers

Registers	offset	Absolute Address	Note
Button Status	0x0	0xFFFF_2000	Button 0 ~ 2
Switch Status	0x4	0xFFFF_2004	Switch 0 ~ 9
Green LEDs	0x8	0xFFFF_2008	
HEX0	0xC	0xFFFF_200C	7 Segment
HEX1	0x10	0xFFFF_2010	7 Segment
HEX2	0x14	0xFFFF_2014	7 Segment
HEX3	0x18	0xFFFF_2018	7 Segment

There are 4 7-segments on DE0 board. Using the instructions supported in the RV32I CPU design, display the last 4 digits of your student ID (학번) on 7 segments. Note that if you use C-language, make sure that the compiler-generated machine code is supported in the provided RV32I CPU design (You can check it with dump file).

What and How to submit:

- Create a 1-min video clip (with your smartphone or any other convenient means), showing
  - Your smiling face to camera
  - 7 segment output on DE0 board

  AND explaining the followings:
  - Your C source code
  - What you have done to display your ID on 7 segments
- Upload the video clip to the instructed place (from TA).

Note: This is an individual assignment. You are welcome to discuss, but DO NOT COPY solutions. If you are found to copy solutions from others or slightly modify the solutions from others, both of you will be given 0 credits.