# COSE222 Computer Architecture
# Final Project 'Last' Milestone

Update your pipelined version of RISC-V, so that it is able to execute the code in the zip file. It will repeatedly display 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9 on HEX0 with a certain time interval.

http://esca.korea.ac.kr/teaching/cose222_CA/milestones/RISCV/final-project_last.7z

You should probably follow the steps below;

1. Compile the code under Eclipse and generate the RV32I binary.

2. Open `labcode.dump` to see what kinds of RV32I instructions you should implement
    a. Figure out which instructions are already there in the single-cycle CPU and what are not there
3. Implement those new instructions to your single-cycle CPU
4. **Simulate with ModelSim** to debug and validate your design.
5. Synthesize the pipelined CPU with the `mif` file using Quartus-II.
6. Download the bitstream to the DE0 board to make sure that it displays the numbers.

7. Implement the pipelined version of the RISC-V.
8. **Simulate with ModelSim** to debug and validate your design.
9. Synthesize the pipelined CPU with the `mif` file using Quartus-II.
10. Download the bitstream to the DE0 board to make sure that it displays the numbers.

`RV32I_System.v` is an overly simplified version of a computer system, consisting of RISC-V CPU, Timer, and GPIO. Nevertheless, it teaches you almost everything you need to know about computer systems. Please think about the followings;

1. We have extensively used the GPIO to display numbers on 7 segments. Check out how it is done. (Hint: Is it a memory-mapped I/O or I/O-mapped I/O? why and how?)
2. Timer is extensively used in a computer system as well. Check out the Verilog code (`TimerCounter.v`) of the Timer.
3. How the pointer in the C code is converted to assembly code.

**What and How to submit:**

- Create a (up to) 3-min video clip (with your smartphone or any other convenient means), showing
  - Your smiling face to camera
  - 7 segment output on DE0 board

  AND **verbally** explaining the followings:
  - **What instructions** you added to the CPU, and **how you figured out those instructions** to be added to the CPU
  - **CPU design change** (please elaborate this!)
  - **ModelSim simulation output**

- Upload **both** the video clip **and** zipped Verilog source to the instructed place (from TA)