COSE222 Computer Architecture
Assignment #4

No late turn-in accepted

1. Execute the sorting program you wrote for the assignment #3 on the real RISC-V CPU you have designed for the class project

    - For the correctness check, add the following pseudo code to your sorting program. Either assembly or C code would be fine.

```
    // Pseudo code for validating the sorting output

      int result;

      result = -16;

      for (i=0; i<32; i++) {
          if (*Output_data != result) {goto ERROR;}
          Output_data++;
          result++;
      }

      Display 1 on HEX0;
Smile:
      j Smile

ERROR:
      Display 2 on HEX1;
Frown:
      j Frown
```

    - Run your final sorting program on your (or reference) single-cycle CPU to make sure it works fine.
    - Run the final sorting program on your pipelined CPU

2. Assume that a direct-mapped **data** cache with the following configuration is placed between your RISC-V CPU and memory.

    - Cache size: 64-byte (so, super tiny cache)
    - Cache line size (== memory block size): 16-byte
    - Write-allocate, Write-back policy

    a) Draw the detailed cache structure specifying the cache size, fields (cache line, valid, …) in the cache entry and the number of cache entries.

    b) After executing your final sorting program,

        a. How many memory blocks are replaced from the cache and why?
        b. What is the final hit ratio in the cache and why?
        c. Draw the final state of the cache

What and How to submit:

- For Question #1:
  - 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 the video clip to YouTube and provide the link to Blackboard
  - Upload the zipped Verilog source, **and** sorting program to Blackboard

- For Question #2:
  - Upload pdf (for question #2) to Blackboard

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.