Assignment 6
Due November 28, 12:59pm

NOTE: Late submissions will **NOT** be accepted. Please put your solutions in the CENG 355 **drop-box** (ELW, second floor) – they will be collected at **13:00**.

1. [3 points] Consider a <u>pipelined</u> datapath consisting of <u>five stages</u>:

> **F** – fetch the instruction from the memory,
> **D** – decode the instruction and read the source register(s),
> **C** – execute the ALU operation specified by the instruction,
> **M** – execute the memory operation specified by the instruction,
> **W** – write the result in the destination register.

Identify data hazards in the code below and <u>insert **NOP** instructions</u> where necessary.

```
ADD    R2, R4, R1         // R1 = R2 + R4
ADD    R4, R6, R5         // R5 = R4 + R6
ADD    R0, R2, R3         // R3 = R0 + R2
MOV    R6, (R1)           // MEMORY[R1] = R6
MOV    (R3), R6           // R6 = MEMORY[R3]
MOV    R4, R2             // R2 = R4
ADD    #4, R4, R4         // R4 = R4 + 4
ADD    R0, R2, R1         // R1 = R0 + R2
MOV    R2, R0             // R0 = R2
```

2. [10 points] Solve Problem **12.7** from the textbook. **Hint:** Declare the shared counter variable as "**volatile int thread_id_counter**", initialize it to 0 in **main()**, and check it by each thread as follows: "**while (thread_id_counter != my_id);**". Each thread must increment **thread_id_counter** after updating global **dot_product**.

3. [2 points] Solve Problem **12.8** from the textbook.

4. [10 points]
  (a) Show **decimal** number **+5.25** in the 32-bit <u>IEEE-754</u> floating-point format.
  (b) Show 32-bit **IEEE-754** number **0 00000000 10000000000000000000000** in the <u>decimal</u> format.
  (c) Show 32-bit **IEEE-754** number **0 01111111 00000000000000000000000** in the <u>decimal</u> format.
  (d) Given two 32-bit <u>IEEE-754</u> floating-point numbers **X** and **Y** below, calculate (in the binary format) **Z = X–Y**, and then convert **Z** to the <u>decimal format</u>:
       X = **1100 0001 1001 0100 1111 0000 0000 0000**,
       Y = **0011 1110 0100 0000 0000 0000 0000 0000**.