CSC612M **1st Term 2015-2016**
Case Project

INTRODUCTION

In this case project, you will implement a simulator for a simplified MIPS64 processor, miniMIPS. The miniMIPS processor offers the following subset of MIPS64 instructions:

1. R-type instructions: DADDU, DMULT, OR, SLT
2. I-type instructions: BEQ, LW, LWU, SW, DSLL, ANDI, DADDIU
3. J-type instruction: J
4. Floating-point instruction: L.S, S.S, ADD.S, MUL.S

The miniMIPS processor is based on the MIPS64 architecture.

The objective this project is to "execute" the program using pipelining with the following schemes to solve the hazards:

- Structural Hazard: Separate Memory
- Data Hazard: No Forwarding
- Control Hazard: Pipeline freeze [Group 1/5/9], pipeline flush [2/6/10], predict-not-taken [3/7], pipeline #2 [4/8]
- Floating point add = 4 clock cycles, floating-point multiply = 8 clock cycles

In this case project, you will create the following modules:

1. Utility program to input the MIPS program. You have the option to use **drop-down menu** method or to **let user input the program**
2. Utility program to input value for registers R1 to R31; F0 to F11, HI and LO
3. Utility program to input value for memory (data segment). Note: code segment is from address 0000-1FFF while data segment is from 2000-3FFF. Also, provide a "GOTO Memory" option to go to target memory location
4. Write a simulator program using pipeline. Simulator should support (a) **single-step instruction execution** mode and (b) **full execution** mode
5. Output screen #1: the equivalent opcode of the MIPS program (in Binary and HEX)
6. Output screen #2: Error message screen
7. Output screen #3: the "pipeline map"
8. Output screen #4: the internal MIPS64 registers as follows:
   IF Cycle: IF/ID.IR, IF/ID.NPC, PC
   ID Cycle: ID/EX.IR, ID/EX.A, ID/EX.B, ID/EX. IMM
   EX Cycle: EX/MEM.IR, EX/MEM.ALUOUTPUT, EX/MEM.B, EX/MEM.cond
   MEM Cycle: MEM/WB.IR, MEM/WB.ALUOUTPUT, MEM/WB.LMD, MEM[ALUOUTPUT]
   WB Cycle: Registers affected

Note: The affected registers (including HI and LO) and affected memory should contain the actual value.
Note: The program should be in an "Integrated Development Environment (IDE)" interface

Milestone #1: October 26, 2015: Program input w/ error checking and Opcode (#1,5,6)
Milestone #2: November 16, 2015: GUI, register and memory input, pipeline map (#2,3,7)
Milestone #3: December 1, 2015: Complete program

1. Dela Cruz – Kalaw
2. Lam – Sapalo
3. Laron – Delos Santos
4. Xu – Go
5. Rivera – Adolfo
6. Sanhi – Macatangay
7. Tan – Lim
8. Cayabyab – Octaviano
9. Ricafort – Abergos
10. Lu - Paner