COMPARC **2nd 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:

Set A:

1. R-type instructions: DADDU, DMULU, DMUHU, SLT, SELEQZ
2. I-type instructions: BEQC, LD, SD, DADDUI
3. J-type instruction: BC

Set B:

1. R-type instructions: DSUBU, DDIVU, DMODU, SLT, SELENEZ
2. I-type instructions: BNEC, LD, SD, DADDUI
3. J-type instruction: BC

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 / Pipeline flush / predict not taken / pipeline #2

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

1. Utility program to input the MIPS program.
2. Utility program to input value for registers R1 to R31
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 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 and affected memory should contain the actual value.
Note: The program should be in an "Integrated Development Environment (IDE)" interface
Note: The program is assume to be stored at address 0

Milestone #1: March 9, 2016: Program input w/ error checking and opcode (#1,5,6)
Milestone #2: March 14, 2016: GUI, register and memory input, pipeline map (#2,3,7)
Milestone #3: March 30, 2016: Complete program

***Group assignment

Freeze (Set A): Castillo et al	(Set B): Santos et al
Flush (Set A): Olan et al	(Set B): Arevalo et al
Predict Not Taken (Set A): Lim et al	(Set B): Lalic et al
Pipeline #2 (Set A): Bautista el al	(Set B): Wong et al