ARC-ORG **2nd Term 2017-2018**

Case Project

INTRODUCTION

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

Common instructions: LD, SD, DADDIU, DADDU, BC

Group 1: BEQZC/ANDI; Group2: BNEZC/ORI; Group 3: BEQC/XORI; Group 4: BNEC/SLTI;
Group 5: BGTC/ANDI; Group 6: BLEC/ORI; Group 7: BLTC/XORI; Group 8: BGEC/SLTI;
Group 9: BGEZC/XORI

The microMIPS processor is based on the MIPS64 architecture.

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: The program is stored starting from address **0100-01FF** while data segment is from **0000-00FF**. Also, provide a "GOTO Memory" option to go to target memory location
4. Simulator should support **single-step instruction execution**
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 internal MIPS64 registers as follows:
   IF Cycle: IR, NPC
   ID Cycle: A,B,IMM
   EX Cycle: ALUOUTPUT, COND
   MEM Cycle: PC, LMD, actual memory affected
   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: Upload source code on CANVAS

Deadline: April 13, 2018; demo during class time