CSC 342, Spring 2025

Very SIMPLE ADD/SUB processor

Instructor: Professor Izidor Gertner

Demo simulation report, video on March 12th, 2018, at 5:00Pm.
Please prepare no more than 2 min video on your project to show in in class.
Please submit to me before 5:00PM your video, report, archived project file and how to run in a REAME file. Your Last Name should be in all file names and on every page of the report. In video you must show your face talking, title page should specify your, project title.

Components used in the project:

1. You can use 3 ported register file you have designed, or design it with LPMs.
2. You can use 32 bit registers you have designed or LPM.
3. Write and simulate VHDL code for 32 bit add/sub- adder subtractor unit you for this part you can not use LPM.

Description of instruction format used:

Input 32 bit simple R-type instruction (MIPS like)
. In the instruction specify two operand registers, destination register, and the operation code***: add or sub, addu, and subu (from "mips green pages")***
The fields should be as following:

- Instruction Register Opcode Bits [26:31] = 000000 encode addition operation
- Instruction Register Opcode Bits [26:31] = 000001 encode subtraction operation
    - Take the opcodes FOR ***: add or sub, addu, and subu*** from "green pages'

- Instruction Register Bits [21:25] = destination register index ranging from {00000} to {11111}
- Instruction Register Bits [16:20] = source register index ranging from {00000} to {11111}
- Instruction Register Bits [11:15] = source register index ranging from {00000} to {11111}

The remaining bits are set to zero.
Hint; YOU HAVE TODESDIGN A PROCESSOR that executes EACH MIPS INSTRUCTION ***: add or sub, addu, and subu. You have to design digital control circuit that based on the opcode the processor controller decodes the opcode and executes corresponding instruction.***

TEST only MIPS **R type (ADD/SUB)instructions.**

- Use Register File component (Read two registers and write to a third register at the same time) you designed in the previous lab.
- Use ADD/SUB unit you have designed. For add, sub, addu, subu
- For Instruction Register you may use LPM module.

ADD/SUB Unit with register file

CSC 342, Spring 2025

Very SIMPLE ADD/SUB processor

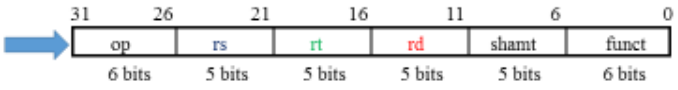
Instructor: Professor Izidor Gertner

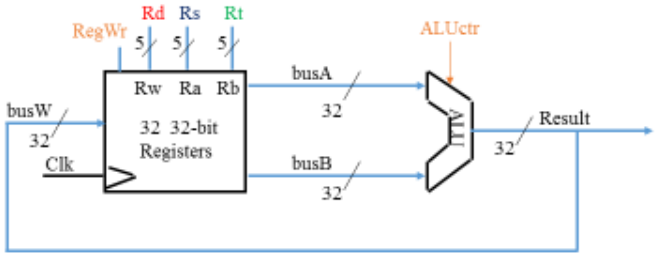
Demo simulation report, video on March 12th, 2018, at 5:00Pm.
Please prepare no more than 2 min video on your project to show in in class.
Please submit to me before 5:00PM your video, report, archived project file and how to run in a REAME file. Your Last Name should be in all file names and on every page of the report. In video you must show your face talking, title page should specify your, project title.

Add & Subtract Instruction

- R[rd] <= R[rs] op R[rt]
  Example: addU rd, rs, rt
  - Ra, Rb, and Rw come from instruction's rs, rt, and rd fields
  - ALUctr and RegWr: control logic after decoding the instruction

Instruction Register
Stores the instruction
During the execution
time





Initialization:

1. Input operand to a specified 32 bit register in the Register file (hint:
2. Input operand 2 using to a specified register .
3. Write to INSTRUCTION REGISTER 32 bit MIPS R-TYPE instruction (ADD or SUB, ADDU,SUBU
   Execution: In simulation inusing Model-SIM. Show waveforms for BUSA, BUSB, Result for all cases.