CSC 343, Fall 2021

Very SIMPLE Add-Sub unit with Register file (3-ported RAM)

Instructor: Professor Izidor Gertner

Screenshots with brief explanations due: OCTOBER 13, 2021, BY 6:15 PM

Report, Video, QAR file_ with README : October 17 TH, 2021, by 10:00 PMN

Components used in the lab:

1. You can use 3 ported register file you have designed in the previous lab, or design it with LPMs.
2. You can use 32 bit registers you have designed the previous lab, or LPM.
3. You can use ADD/SUB unit you have designed in the previous labs, or LPM, pleaser make sure all flags areset..

Description of instruction format used:

Initialize instruction register IR with 32 bit simple instruction (MIPS like) . In the instruction register IR specify two operand registers, destination register, and the operation code: add or sub.

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

- Instruction Register Bits [21:26] = 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.

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 in the previous LAB, alternatively you may use LPM module, pleaser add required flags.
- For Instruction Register you may use LPM module.
- Use VHDL CODE (NO BLOCK DIAGRAMS APPROACH) to describe the ADD/SUB unitr
- Use key as a rising edge event to write to register,

CSC 343, Fall 2021

Very SIMPLE Add-Sub unit with Register file (3-ported RAM)

Instructor: Professor Izidor Gertner

Screenshots with brief explanations due: OCTOBER 13, 2021, BY 6:15 PM

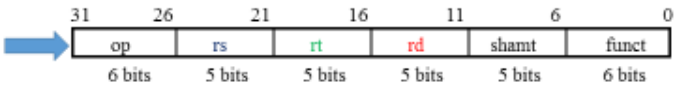
Report, Video, QAR file_ with README : October 17 TH, 2021, by 10:00 PMN

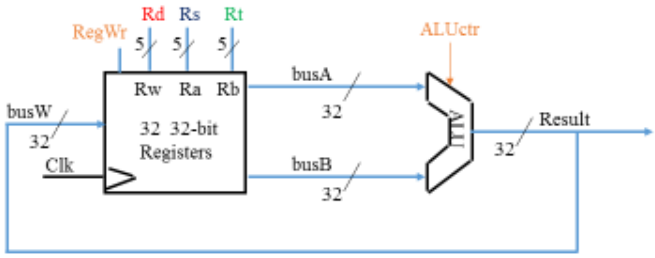
MINI ADD/SUB Unit with register file

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. Initialize 3 Ported RAM (REGISTER FILE) using MIF file to the numbers used in ADD/SUB lab.
2. Input 32 bit MIPS R-TYPE instruction (ADD or SUB) to Instruction register IR.

3. Execution:

1. Display THE RESULTS USING waveforms in MODEL-SIM.
2. Integration step: Read DATA from two different addresses and feed their obtained data to the two inputs of the adder/subtractor unit, and thereafter write the result of the adder/subtractor unit into the third address in 3 PORTED RAM BLOCK. We call this block register file.