Congratulations on your hard work and dedication to complete your own MIPS-32 processor implementation! Implementing a working processor is not a trivial task, but you have done it. You should be proud of yourself that you come this far, considering that you had no experience in vhdl and processor design.

Lab:

This last lab is to test the MIPS machine codes as the final step. Write and test a MIPS assembly code that adds from 1 to 200 (C8 hex) and then stores the result at a memory location on your own processor. This time connect the clock to the system clock (50MHz) or a divided slower clock. In order to display the final result, use an infinitive loop of loads. This can be achieved through displaying the signal read_data to the seven segment display output. The seven segment display is used because it is a faster display. Your assembly code should look like

MIPS code that computes 1+2+3+, ….,+ 200=20100=4e84 hex.

L1: ....
    beq $1,$2, L2  -- “bne” can be implemented by beq and j combination
    j   L1
L2:  --- Assume that the addition result is in $5
    sw $5, 3($0)  --- Save the addition result at mem[3]
L3: lw  $5, 3($0)   --- Create an infinitive loop to display the result stored in memory
    j    L3   --- to the seven segment display.

Above method should allow you to test your processor’s loop (branch) capability.

One common mistake students make is recognizing the instructions implemented. Remember that you only implemented lw, sw, rformat, bew, jmp, and addiu. Your program can use only these instructions.

Another important feature you should test is the reset capability. Resetting your processor should produce the same result, that is, it should be able to repeat the execution of the addition application program. This is because the processor starts from PC=0 by the reset and your code ends up executing an infinitive loop.

Demo:

Show the task described above, i.e., the addition result, and reset.

Report

- Objective of this lab and intro.
- Show your MIPS source code and explain how it works
- Attach fetch.vhd, execute.vhd, memory.vhd, and maintest.vhd.
- Your conclusion
• Your experience on this lab, wishes, and comments that would help improve this lab for the future ECE students