Project 4: Single Cycle MIPS Processor

Due Date: 30.12.2013 Monday 23:59

In this project you will complete the design of your MIPS processor by adding most of the remaining MIPS instructions. Your previous memory, register and ALU projects should contain most of the major components, with the exception of Data RAM for the load/store instructions. For this project, we will use a split-memory design: *The Program ROM* will store a read-only copy of the instructions, and a separate *Data RAM* will be used to store data for the program's execution. You should design Data RAM similar to your previous project.

Begin by making a pencil-and-paper design showing all of the components you will need, and create a list of all control signals in the processor. You will not submit this paper design, but the TA will demand to see it if you ever have a question about anything. Next, decide for each instruction what the value of each control signal must be, and use this information to design the instruction decode logic. Lastly, implement your decoder and the rest of the processor in Xilinx ISE, and test your circuit again in Xilinx ISE by testbenches.

Use the book for the machine codes of the instructions and the single cycle CPU picture, which will show you the path to design the processor.

Your design must implement the following subset of the MIPS architecture:

Jumps	J, JR
Branches	BEQ, BNE, BLEZ, BGTZ
Memory Load/Store	LW, SW
Immediate load	LUI
Immediate arithmetic	ADDI, ADDIU, ANDI, ORI, XORI
Register arithmetic	ADDU, SUBU, AND, OR, XOR, NOR
Shifts	SLL, SRL, SRA

How to submit: A single zipped project file containing your processor and simulation files will be sent to Moodle. Send the file named with your number as "**yourID_Project4.zip**".

For the Adventurous:

Note: These suggestions for an extra challenge will be examined (and commented on, if your project works well) and also will be graded by bonus points.

- Implement the jump-and-link instructions JAL, JALR (barely challenging). (10 pts)
- Implement multiply (MULT and/or MULTU) (moderately challenging). (20 pts)

Deviation from the MIPS Standard:

You can ignore any MIPS instruction or feature not mentioned in this document, and can assume that your processor will never encounter anything but legal instructions from the table above. In addition:

- Assume loads and stores will be word aligned and to valid Data RAM addresses.
- Assume all jumps will be word aligned and to valid Program ROM addresses.
- Assume traps or exceptions will not occur (i.e., do not implement them).
- Do not implement floating point, multiply/divide, or the HI and LO registers.

Ask anything you did not understand to TA or instructor before start.