CSE 331 - Computer Organization
Project-4: Welcome MIPS

Due date: December 28, Monday – 17:00

In this project, you will use Altera Quartus II with Verilog. You will the 32-bit MIPS processor. The block that you will design will get no inputs from outside. You will design two additional memories: Data Memory and Instruction Memory. The instructions must be loaded to the instruction memory and the data must be in data memory. You will add **xor, xori, slt, sltiu, lw, lb, sw, sb, j, jal, jr, beq** and **bne** instructions to your previous design. Remember the previously supported instructions: **add, sub, and, or, sra, srl, sll, sltu** and **addi, addiu, andi, ori, slti, lui** instructions. Insert a **new instruction** on your own to MIPS. **Find a suitable new instruction, define it and design it.**

You will write test bench and simulate your design for verification. You will write the register and memory contents before and after the execution of instructions using **writememh** in your test bench verilog code. You will initialize memory contents using **readmemh**.

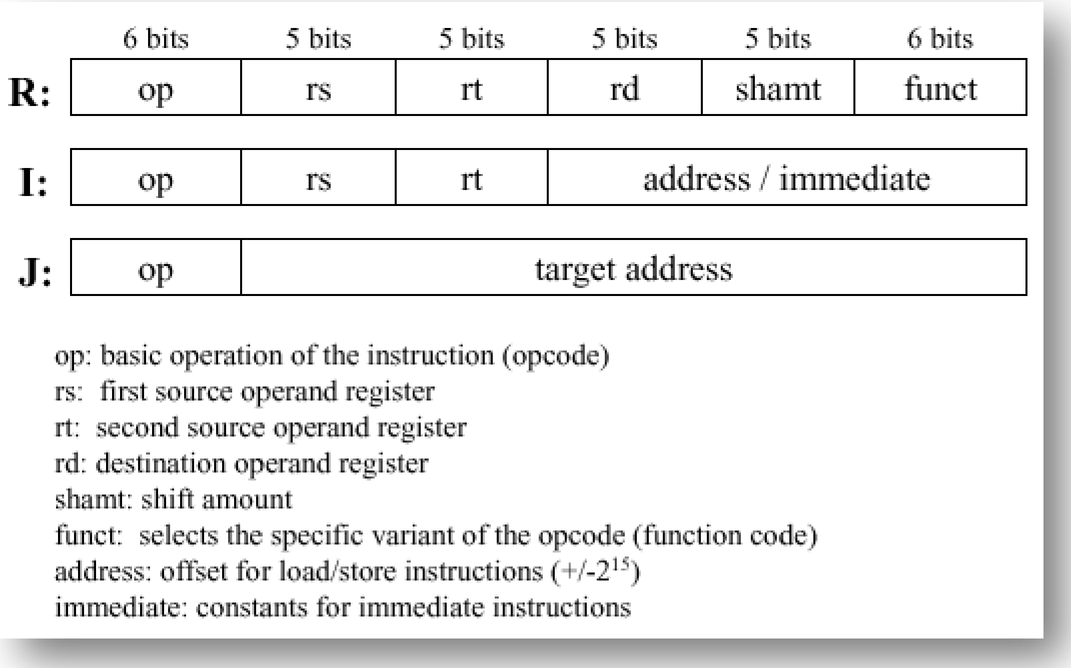
The data memory size will be 2KB whereas the instruction memory size will be 1KB. Remember that addressing for a 2KB memory only requires 11 bits instead of 32 bits in regular MIPS. Update your design accordingly.

(Bonus) Pipelined CPU design brings up additional 35pts. All design must belong to you.

(Bonus) Forwarding unit additional 15pts. Hazard detection unit additional 15pts.

(Bonus) Multicycle CPU design brings up additional 35pts. All design must belong to you.

(Bonus) Each new instruction produced by you brings up additional 5pts until 20pts.



	6 bits	5 bits	5 bits	5 bits	5 bits	6 bits
R:	op	rs	rt	rd	shamt	funct
I:	op	rs	rt	address / immediate		
J:	op	target address				

op: basic operation of the instruction (opcode)
rs: first source operand register
rt: second source operand register
rd: destination operand register
shamt: shift amount
funct: selects the specific variant of the opcode (function code)
address: offset for load/store instructions (+/-2^{15})
immediate: constants for immediate instructions

Please be sure that your design simulates correctly. Designs that are not even simulating can get at most 20 points.

Submit your Altera Project folder as a zip file to Moodle. We will simulate your design using not only your testbench but also our testbench to see whether all instructions are executing correctly or not.

No late submissions even if it is 1 minute. No medical reports. No excuses. No cry. So start early.

Any cheating attempt with the previous years' projects or with your friends or Internet will result in at least -100 and at most -300. No matter you gave or take the code. Protect your code. Do it yourself for your own good.