CENG311 - Computer Architecture **Fall 2023**

PROGRAMMING ASSIGNMENT #3 **Due date:** 02.01.2023 23:55

In this assignment, you are required to create your own ISA and its implementation by adjusting/updating the MIPS-lite single-cycle implementation (discussed in your laboratory sessions). You will use ModelSim simulator to develop and test your code.

In this design, your processor will still be 32-bits, but your register file will have **16 registers**. All the students will have the same set of ISA, which is given in Table 1. You can change the order of the instructions as you wish, the table is provided to you only as an example. For the R-type instructions, you have the flexibility to modify the function bits and ALU control lines as you wish. (or you can keep them as it is.) Opcodes of the instructions are changing according to your student id. Least significant 3 bits of your id will be the number of the first instruction. For example;

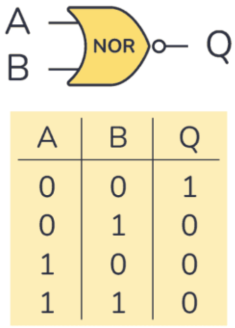
Student id: 230062**100** -> 100 is the first instruction's opcode.

opcode	instruction
100	R-type
101	lw
102	sw
103	jump
104	beq
105	bne
106	addi

Table 1. Instruction Set Architecture

Additionally, you should add 4 instructions to your design.

1) **R-type:** nor gate: You can choose any proper function and Alucont bits for this new instruction.



A	B	Q
0	0	1
0	1	0
1	0	0
1	1	0

2) **I-type:** addi : performs an addition on source register's data and 16bits immediate number, stores the result in the destination register.

3) **J-type:** jump : jump **directly to given address**. If the given address is 0x4, pc should be updated with pc = 0x4 directly.

4) **B-type:** bne (branch not equal): compares rs and rt register's contents, if the values in the two registers are not equal, branch to given 16bits address+(PC+4).

Accordingly, the instructions will be as given:

J-type:

8 bits	24 bits
Opcode [31:24]	address [23:0]

R-type:

8 bits	4 bits	4 bits	4 bits	6 bits	6 bits
Opcode [31:24]	rs[23:20]	rt[19:16]	rd[15:12]	shamt[11:6]	funct[5:0]

I-type:

8 bits	4 bits	4 bits	16bits
Opcode [31:24]	rs[23:20]	rt[19:16]	address[15:0]

B-type:

8 bits	4 bits	4 bits	16bits
Opcode [31:24]	rs[23:20]	rt[19:16]	address[15:0]

You must design revised single-cycle datapath and control units that make a processor that executes all instructions as well as the instructions implemented already in the design. You also need to implement the nor, addi, jump and bne instructions.

After designing the updated version of the MIPS processor, you will implement it in Verilog hardware description language.

You are required to submit a report that just includes your ISA and your Verilog code. You are going to demonstrate your work to the TA. During the demo section, you will write the instructions to the Instruction Memory and execute them. Therefore, ensure that all instructions work before the demo. Additionally, it would be beneficial for you to prepare a document with the binary and hex representations of the instructions.

Requirements:

- You must submit the source code and a report.
- You need to work individually, no group work is allowed.
- No late homework will be accepted.

Submission: You are required to submit all of your files to Teams. Please create a compressed file including all source files and report, and name it as yourstudentnumber_P3.zip (e.g. If your student number is 202112345678, the file name must be 202112345678_P3.zip). Your report should just include your ISA as given in Table1.