1. Implement the MIPS register file that reads simultaneously from two registers and write into one register.

   Main module should be called "RegisterFile".

   Entity should look as follows:

   read_sel1 : in std_logic_vector(4 downto 0)

   read_sel2 : in std_logic_vector(4 downto 0)

   write_sel : in std_logic_vector(4 downto 0)

   write_ena : in std_logic

   clk: in std_logic

   write_data: in std_logic_vector(31 downto 0)

   data1:out std_logic_vector(31 downto 0)

   data2: out std_logic_vector(31 downto 0)

   Test before submit (attached test case).

2. Implement 32 bit full ALU.

   Main module should be called "ALU" ALU functional specifications:

ALUOp	Function
0000	AND
0001	OR
0010	ADD
0110	SUB

   ALU RTL should match the desired RTL (refer to lab06). Entity should look as follows:

   data1 : in std_logic_vector(31 downto 0)

   data2 :in std_logic_vector(31 downto 0)

   aluop : in std_logic_vector(3 downto 0)

   cin : in std_logic

   dataout: out std_logic_vector(31 downto 0)

   cflag:out std_logic

   zflag: out std_logic

   oflag:out std_logic

   Test before submit (attached test case).

Delivery method:

Will announced soon ☺

Deadline: Thursday, 09 April, 2020 23:59.

Good luck!