

**CSE331**

**COMPUTER ORGANIZATION**

**HOMEWORK 2 REPORT**

**Burcu Sultan ORHAN**

**1901042667**

## MODULES:

- **ALU32bit.v**

This module takes 2 inputs which are 32 bit and used as operands, 1 input which is 3 bit and used as operation code, creates 1 output which is 32 bit and used as result. This module calls all the other modules which create our ALU and actually make the arithmetic work. Finally, this module sends the result to 8x1 mux which creates the output.

- **\_32bit\_adder.v**

This module takes 2 inputs which are 32 bits and used as operands, 1 input which is Carry In, performs addition operation through \_8bit\_adder module and creates 2 outputs, one is sum which is 32 bits, and the other is Carry out.

- **\_8bit\_adder.v**

This module takes 2 inputs which are 8 bits and used as operands, 1 input which is Carry In, performs addition operation through \_4bit\_adder module and creates 2 outputs, one is sum which is 8 bits, and the other is Carry out.

- **\_4bit\_adder.v**

This module takes 2 inputs which are 4 bits and used as operands, 1 input which is Carry In, performs addition operation through full\_adder module and creates 2 outputs, one is sum which is 4 bits, and the other is Carry out.

- **full\_adder.v**

This module takes 2 inputs which are 1 bit and used as operands, 1 input which is Carry In, performs addition operation through half\_adder module and creates 2 outputs, one is sum which is 1 bits, and the other is Carry out.

- **half\_adder.v**

This module takes 2 inputs which are 1 bit and used as operands, performs addition operation and creates 2 outputs, one is sum which is 1 bit, and the other is Carry out.

- **\_32bit\_xor.v**

This module takes 2 inputs which are 32 bits and used as operands, and creates an output which is 32 bits and used as Result. This module performs xor operation on operands 1 by 1 bit and stores the results in output.

- **\_32bit\_sub.v**

This module takes 2 inputs which are 32 bits and used as operands, 1 input which is carry in and creates an output which is 32 bits and used as Result, 1 output which is carry out. This module performs subtraction operation on operands with adding them and 1 after xor'ing second operand with 32-bit 1 value so that it would be negative.

- **\_32bit\_slt.v**

- This module takes 2 inputs which are 32 bits and used as operands, and creates an output which is 32 bits and used as Result. This module performs set-less-than operation on operands.

- **\_32bit\_nor.v**

This module takes 2 inputs which are 32 bits and used as operands, and creates an output which is 32 bits and used as Result. This module performs nor operation on operands 1 by 1 bit and stores the results in output.

- **\_32bit\_and.v**

This module takes 2 inputs which are 32 bits and used as operands, and creates an output which is 32 bits and used as Result. This module performs and operation on operands 1 by 1 bit and stores the results in output.

- `_32bit_or.v`

This module takes 2 inputs which are 32 bits and used as operands, and creates an output which is 32 bits and used as Result. This module performs or operation on operands 1 by 1 bit and stores the results in output.

- mux8x1.v

This module takes 8 inputs which are 32 bits and used as result registers, 1 input which is 3 bits and used as operation code (selection bit), and creates an output which is 32 bits and used as Result. This module performs mux operation on operands through mux4x1 module.

- mux4x1.v

This module takes 4 inputs which are 32 bits and used as result registers, 1 input which is 2 bits and used as operation code (selection bit), and creates an output which is 32 bits and used as Result. This module performs mux operation on operands through mux2x1 module.

- mux2x1.v

This module takes 2 inputs which are 32 bits and used as result registers, 1 input which is 1 bit and used as operation code (selection bit), and creates an output which is 32 bits and used as Result. This module performs mux operation on operands.

- test bench.v

[illegible]