Comp E 475

Microprocessors

HW7: ALU Module

Student: Davit Barbakadze

Red ID #: 822164584

4.12.2020

Contents

[Task Description 2](#_Toc58244244)

[Solution 3](#_Toc58244245)

[Simulation & Verification 4](#_Toc58244246)

[Conclusion 6](#_Toc58244247)

# Task Description

For the assignment 7 we had to implement ALU module, the scheme is the same as the one used in previous assignment PC module, see scheme in figure 1.

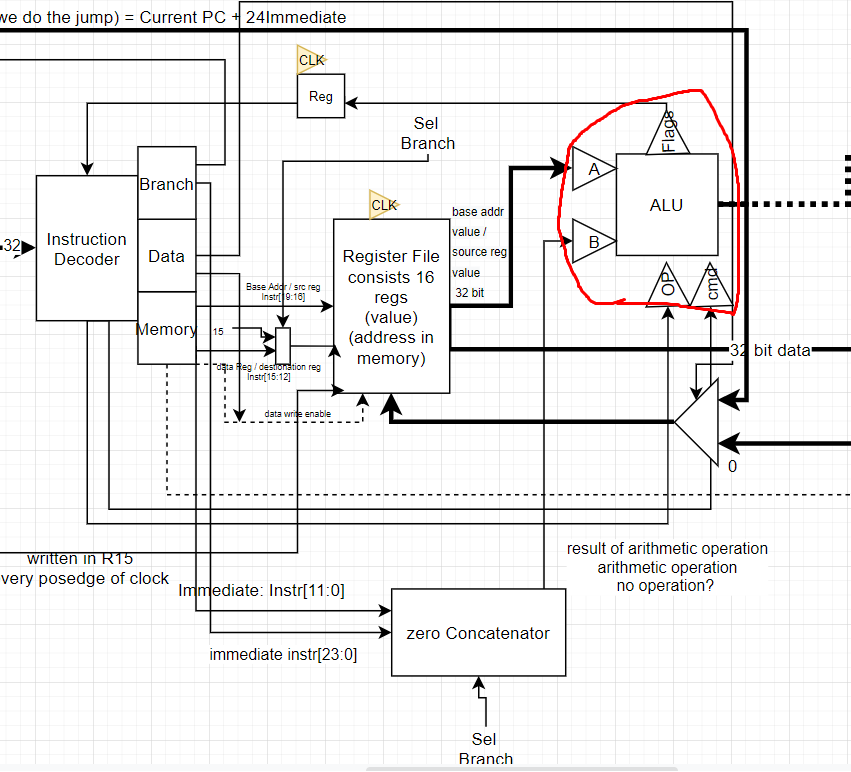


Figure scheme

On the scheme we see ALU module inside the red line that is the part we had to implement in Verilog code.

# Solution

As seen on the scheme, ALU module consists of 4 inputs: A and B operands, operation type – op, and operation – cmd. We also can see that the module has 2 outputs: flags, which consist of 4 bits (4 flags) and the result from ALU. Inputs A and B, as well as output from ALU – ALU\_out had to be 32 bit. We learned how ALU makes operations from lectures and slides and created logics for flags and outputs, so that ALU takes two inputs, makes desired operation and sends the output on ALU\_out and sets the proper flags.

# Simulation & Verification

------------------------------------------- VERILOG CODE -----------------------

I implemented logic where I have two always blocks. First always block is responsible for choosing operation based on cmd and setting the value for output – ALU\_out. See figure 1.

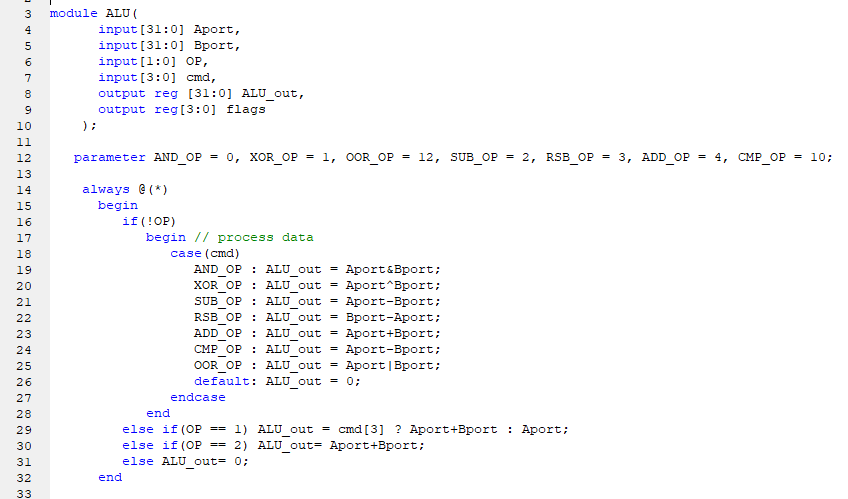


Figure instantiation and output setter

On the second always block I implemented logic for setting proper flags (see figure 2).

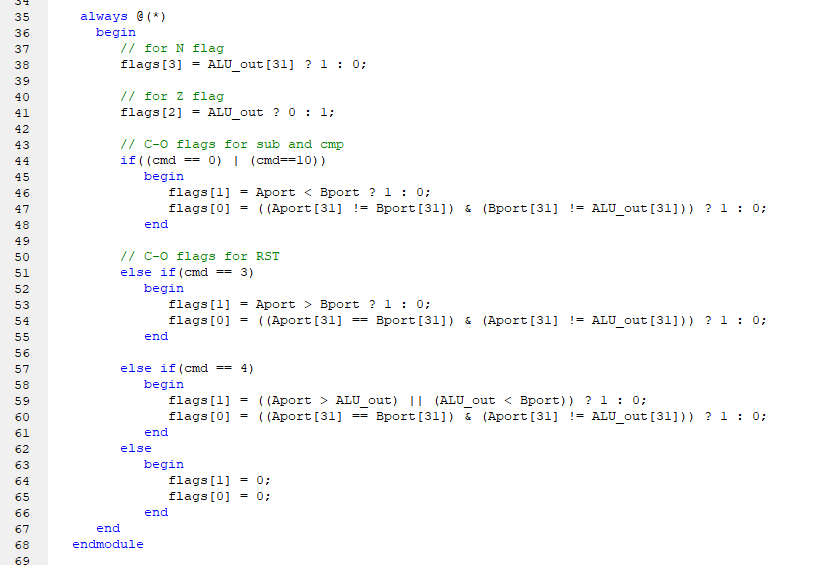


Figure logic for setting flags

------------------------------------- testbench code-----------------------------

In testbench I tried all the commands and saw the outputs as well as flags if they were set correctly (see testbench code in figures 4-5 and in attached V file).

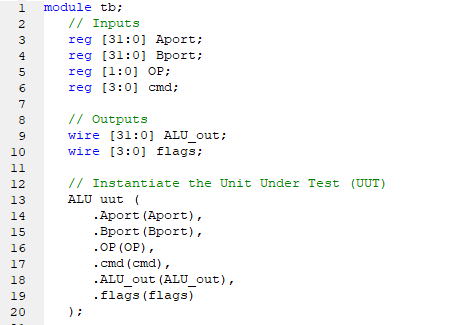


Figure testbench signal instantiations

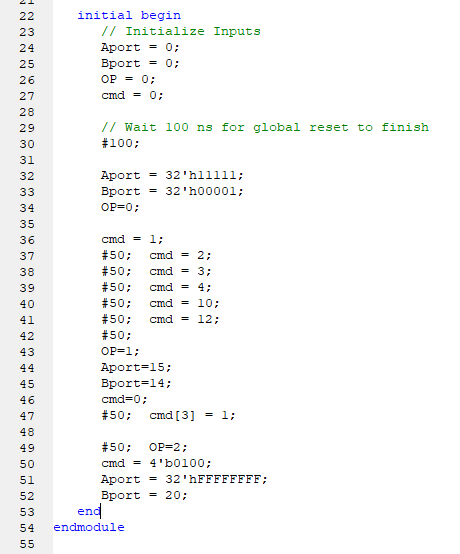
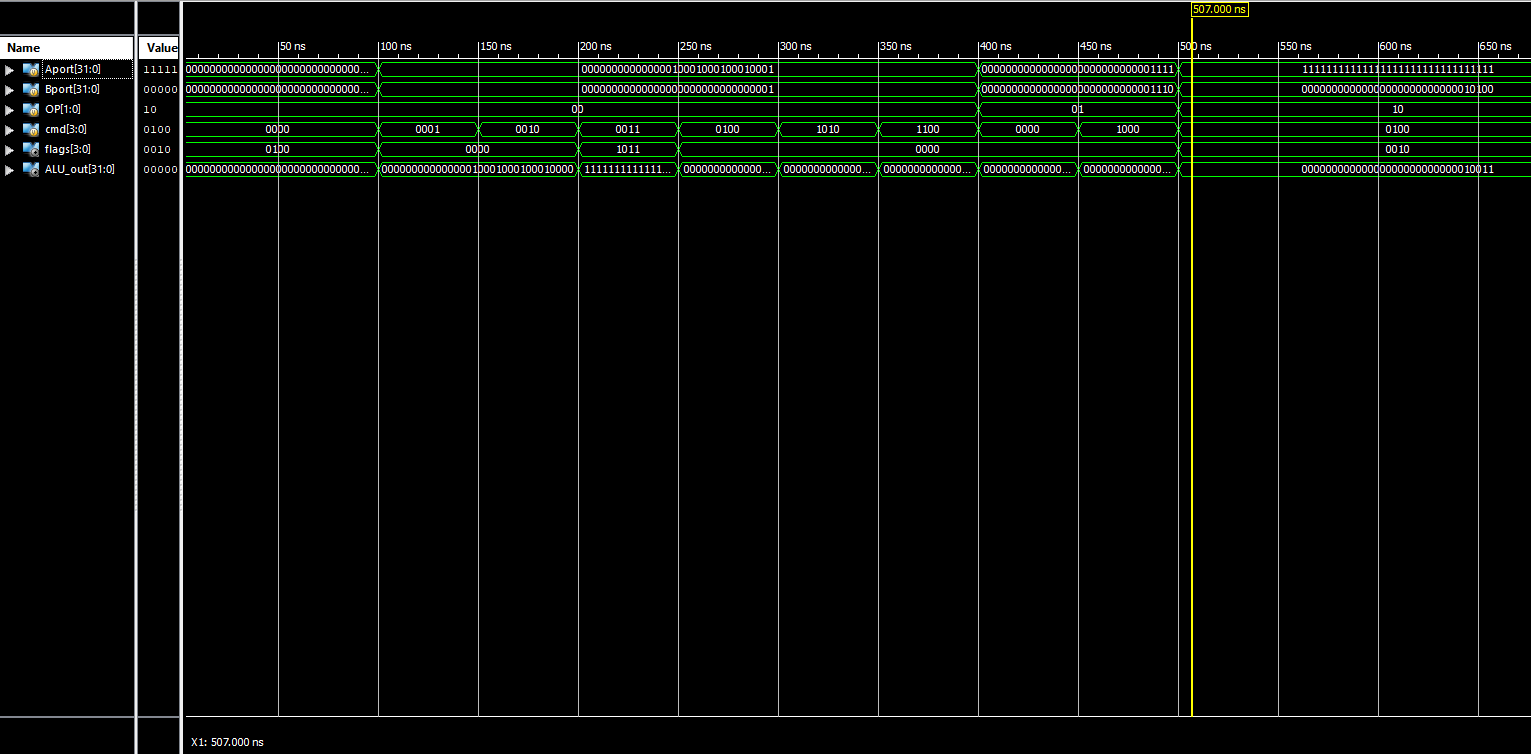


Figure testbench signal modifications

------------------------------- SIMULATION ---------------------------------



# Conclusion

The lab was quite interesting as we had to implement ALU module by ourselves and see how it works. We had to follow instructions from the lecture slides which made it easier to do this lab. The project has 1 warning, More than 100% of Device resources are used, but since we have not learned what it is based on yet I left it as it was.

https://github.com/nikanika0221/ALU\_Module.git