Comp E 475

Microprocessors

Lab 6 : PC Module

Student: Davit Barbakadze

Red ID #: 822164584

31.12.2019

Contents

[Task Description 2](#_Toc22460835)

[Solution 2](#_Toc22460836)

[Simulation & Verification 2](#_Toc22460837)

[Comparison 3](#_Toc22460838)

[Conclusion 3](#_Toc22460839)

# Task Description

In this laboratory assignment students had to implement PC part of the CPU code.

# Solution

First of all we learned about how CPU works, with its parts one by one. We were given scheme where all parts and the input/outputs are visible and we had to implement specific part as shown in Figure 1.

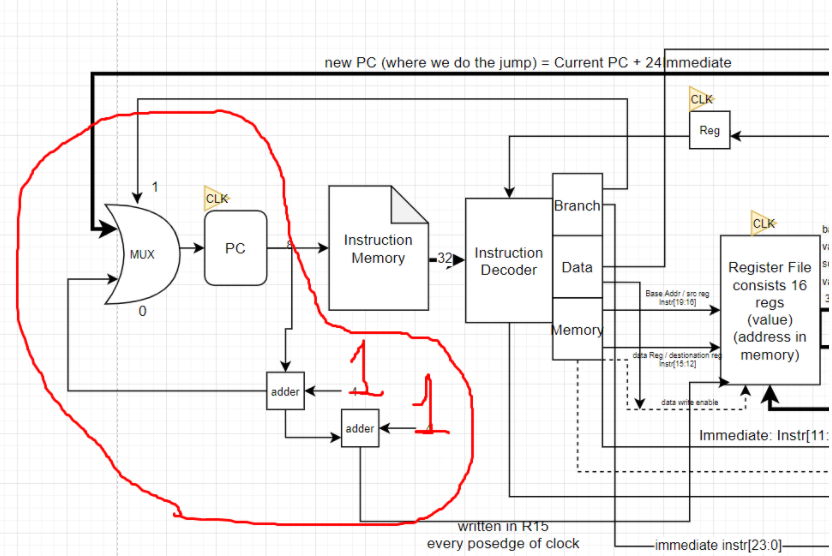


Figure Schema

As shown on the scheme PC has 2-to-1 multiplexer, 2 inputs and 1 selector with one output going into PC. PC sends signal to multiplexer and to adders. The project itself, as shown in the scheme, has 3 inputs one as a clock signal, another is input for mux and the last is selector for mux. The project should also have two outputs one coming out of PC and going to instruction memory, and another for R15 value, which is PC value + 2.

As PC is working with clock signal, we have put its value to mux out at every posedge signal (Line 17).

# Simulation & Verification

-----------------------Verilog Code--------------------------

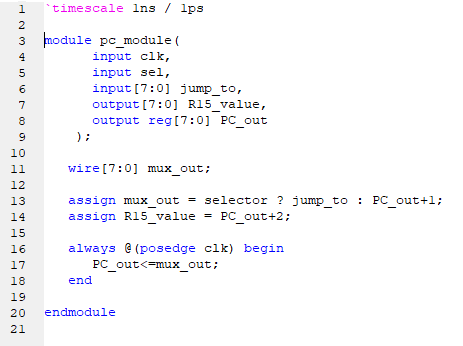


Figure Verilog module code

-----------------------------------Test bench code------------------------------------

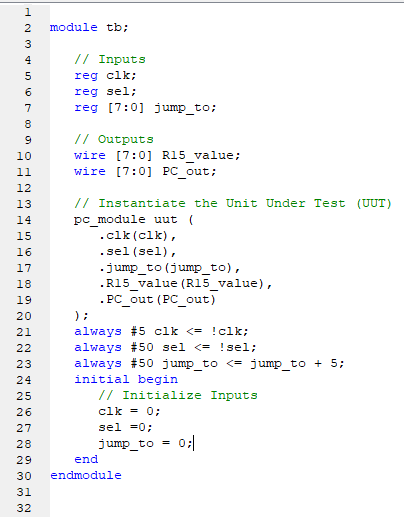


Figure Test Bench code

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

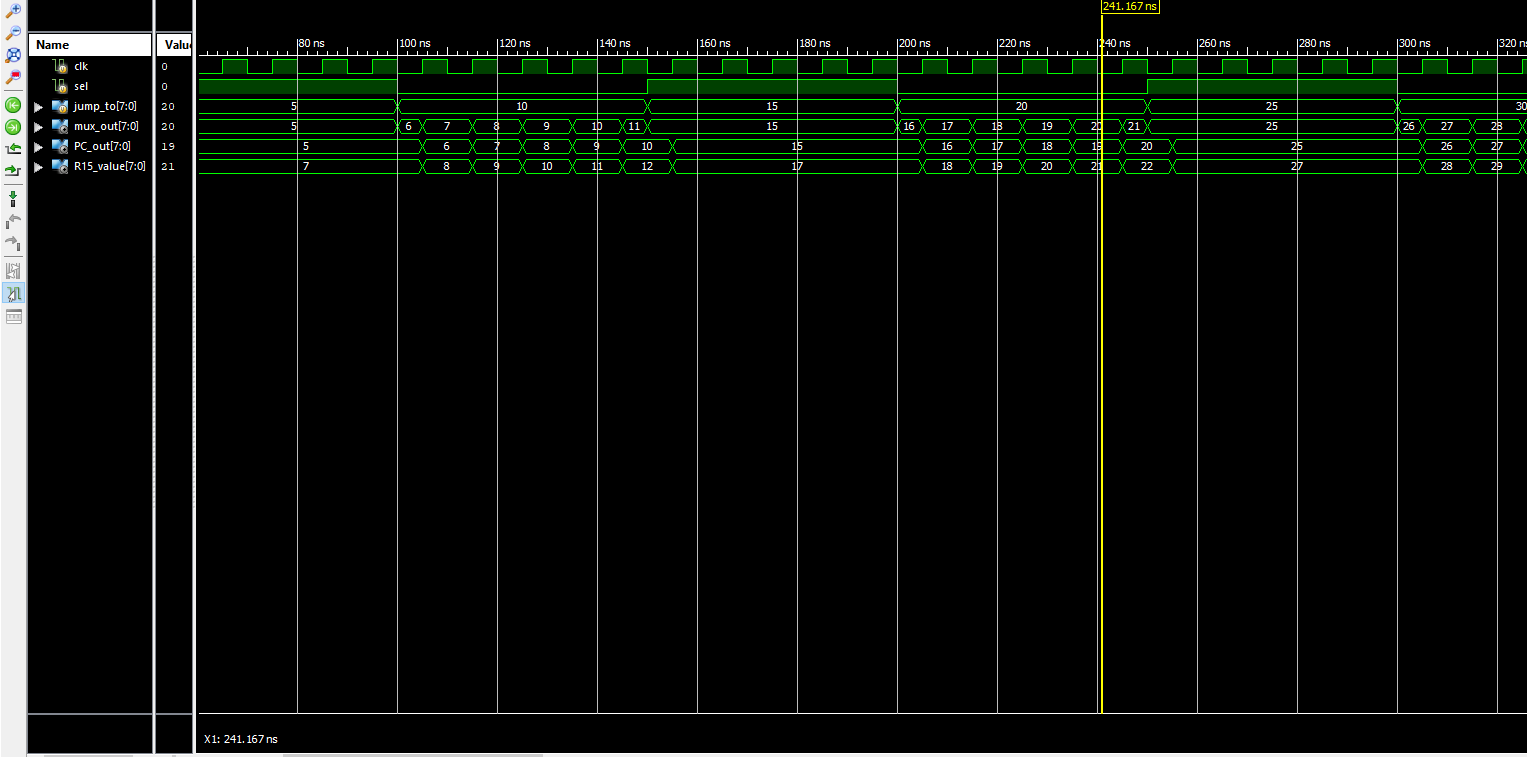


Figure Simulation

As we see on simulation (figure 4) if sel signal is 1 mux out gets the value of jump\_to signal, and when sel signal is 0 mux out gets the value of PC\_out+1. And R15\_value signal follows PC\_out + 2.

# Conclusion

The code was simple to write, the difficulty was to learn how CPU itself works, but pc part was quite small and simple.

Git repository:

https://github.com/nikanika0221/PC\_part