Comp E 475

Microprocessors

HW6 : PC module

Student: Giorgi Tchkoidze

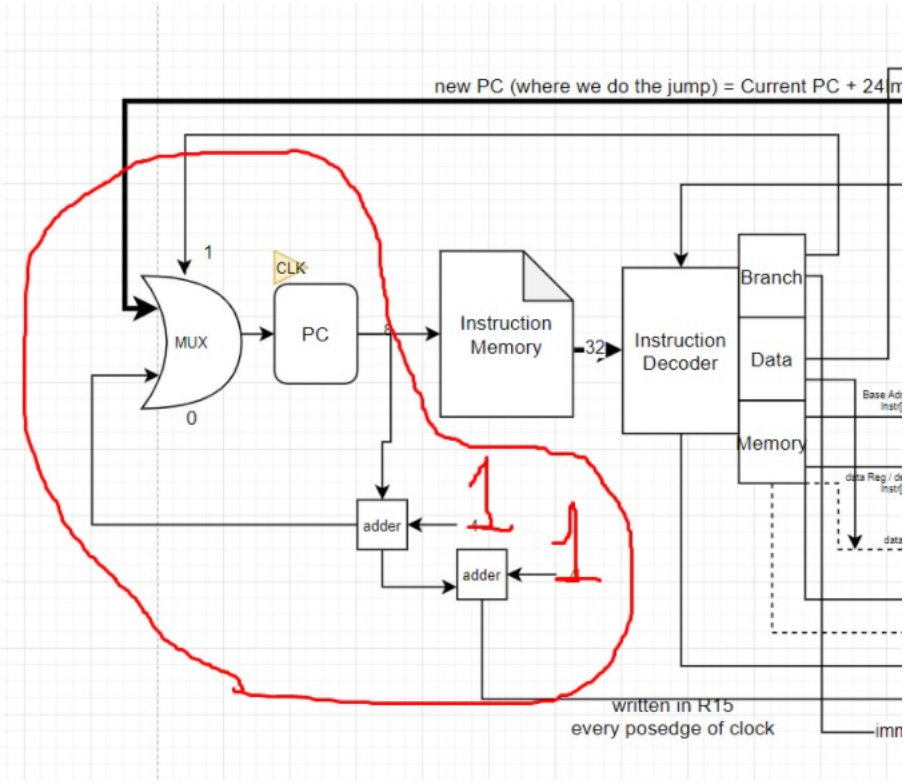
Red ID #: 822089301

31.12.2019

Contents

Task Description

For this assignment we have to implement following controller module in verilog:



Solution

On the diagram we have two most important element: multiplexer and adder, since every output depends on them.

Multiplexer gets 3 input: select signal, jump location and incremented PC output by one. So we have to implements this according to following table:

Sel	output
0	jmp
1	pc_out+1

So we have:

```
assign mux_in = sel ? jmp : pc_out+1;
```

for adder we do not need any special solution, we can just increment registers.

Entire solution code:

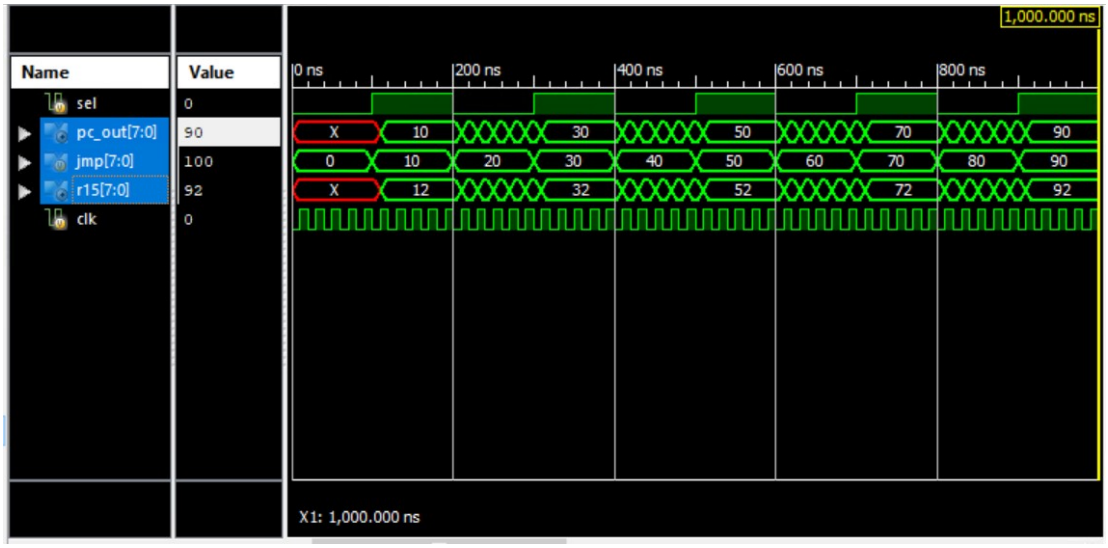
```
`timescale 1ns / 1ps

module pc_module(
    input clk,
    input sel,
    input [7:0] jmp,
    output reg [7:0] pc_out,
    output [7:0] r15
    );
    wire [7:0] mux_in;

    always @ (posedge clk) begin
            pc_out <= mux_in;
    end

    assign mux_in = sel ? jmp : pc_out+1;
    assign r15 = pc_out + 2;

endmodule
~
~
~
```

Simulation & Verification

#Tb code:

```
module tb;
        reg clk;
        reg sel;
        reg [7:0] jmp;
        wire [7:0] pc_out;
        wire [7:0] r15;

        pc_module uut (
                .clk(clk),
                .sel(sel),
                .jmp(jmp),
                .r15 (r15),
                .pc_out (pc_out)
        );

        always #10 clk <= !clk;
        always #100 sel <= !sel;
        always #100 jmp <= jmp + 10;

        initial begin

                clk = 0;
                sel = 0;
                jmp = 0;

        end
endmodule
~
~
```

#Result:



Conclusion

In this part, solution was very straightforward without any complicated codes.