Name: Laiba Fatima Khan

CA Lab09 Report

TASK 1:

```
module Program_Counter
(
	input clk, reset,
	input[63:0] PC_In,
	output reg [63:0] PC_Out
);


	always@(posedge clk)
	begin
	if (reset)
		PC_Out = 0;
	else
		PC_Out = PC_In;
	end

endmodule
```

TASK 2:

```
module Adder
(
	input[63:0] a, b,
	output reg[63:0] out
);

	always@(*)
	begin
		out = a + b;
	end

endmodule
```

TASK 3:

Using PC and Adder modules from task1 and task2 resp. and Instruction Memory module from lab8

```
module Instruction_Memory
(
	input[63:0] Inst_Address,
	output reg[31:0] Instruction

);
```

```
	reg[7:0] Inst_Mem[15:0];


	//initialize acc to fig:8.3
	initial
		begin
			Inst_Mem[0] = 8'b10000011;
			Inst_Mem[1] = 8'b00110100;
			Inst_Mem[2] = 8'b00000101;
			Inst_Mem[3] = 8'b00001111;
			Inst_Mem[4] = 8'b10110011;
			Inst_Mem[5] = 8'b10000100;
			Inst_Mem[6] = 8'b10011010;
			Inst_Mem[7] = 8'b00000000;
			Inst_Mem[8] = 8'b10010011;
			Inst_Mem[9] = 8'b10000100;
			Inst_Mem[10] = 8'b00010100;
			Inst_Mem[11] = 8'b00000000;
			Inst_Mem[12] = 8'b00100011;
			Inst_Mem[13] = 8'b00111000;
			Inst_Mem[14] = 8'b10010101;
			Inst_Mem[15] = 8'b00001110;
		end

	always@(*)
	begin
		//concatenate Inst_Mem[Inst_Address+3:Inst_Address] as output Instruction
		assign Instruction = { Inst_Mem[Inst_Address+3], 
Inst_Mem[Inst_Address+2] ,Inst_Mem[Inst_Address+1], Inst_Mem[Inst_Address]};
	end

endmodule
```

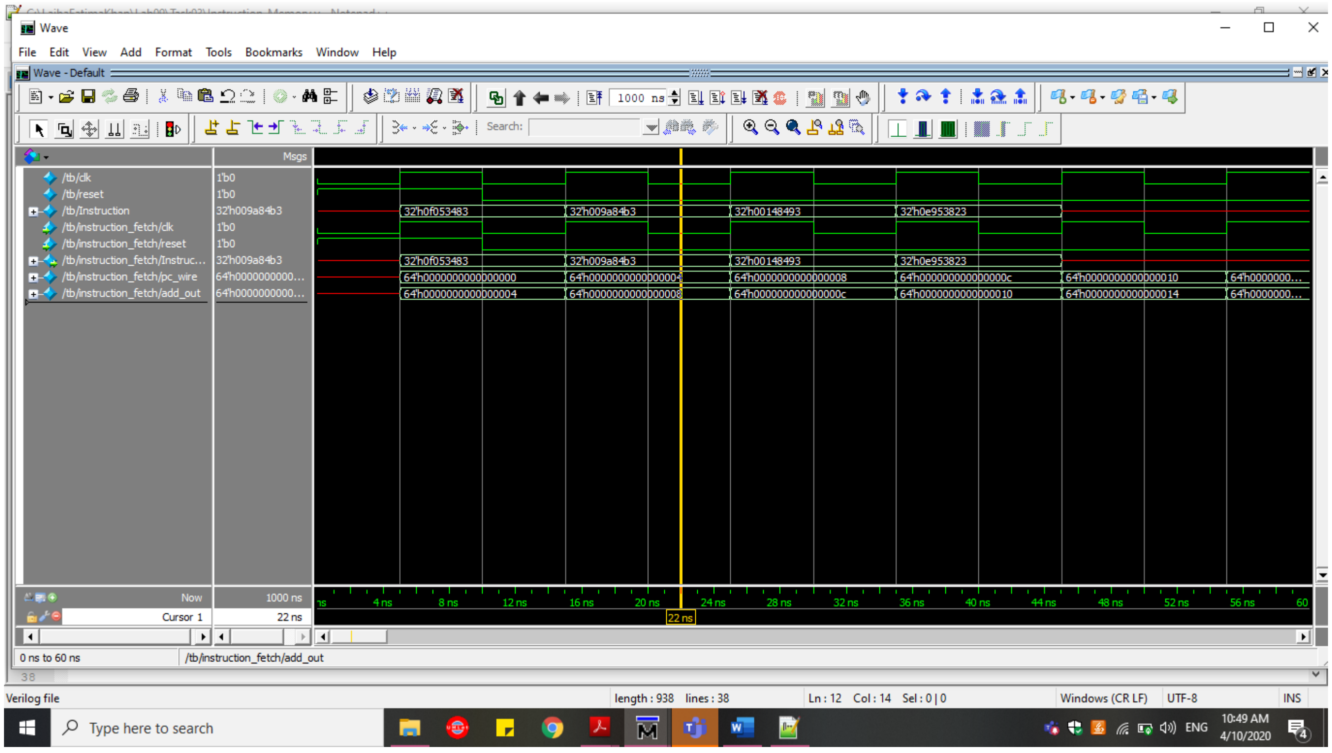
module Instruction_Fetch

```
(
	input clk, reset,
	output[31:0] Instruction
);

	wire[63:0] pc_wire;
	wire[63:0] add_out;

	Program_Counter PC
	(
		.clk(clk),
		.reset(reset),
		.PC_In(add_out),
		.PC_Out(pc_wire)
	);

	Instruction_Memory ins_mem
	(
		.Inst_Address(pc_wire),
		.Instruction(Instruction)
	);
```

```
        Adder adder
        (
                .a(pc_wire),
                .b(64'd4),
                .out(add_out)
        );

endmodule
```

module tb

```
(
);
        reg clk, reset;
        wire [31:0] Instruction;

        Instruction_Fetch instruction_fetch
        (
                .clk(clk),
                .reset(reset),
                .Instruction(Instruction)
        );
        initial
        begin
         clk = 0;
         reset = 1;
         #10 reset = ~reset;
        end


        always
        #5 clk = ~clk;
endmodule
```



```
/tb/clk                          1'b0
/tb/reset                        1'b0
/tb/Instruction                  32'h009a84b3      32'h0f053483        32'h009a84b3        32'h00148493        32'h0e953823
/tb/instruction_fetch/clk        1'b0
/tb/instruction_fetch/reset      1'b0
/tb/instruction_fetch/Instruc... 32'h009a84b3      32'h0f053483        32'h009a84b3        32'h00148493        32'h0e953823
/tb/instruction_fetch/pc_wire    64'h0000000000... 64'h0000000000000000 64'h0000000000000004 64'h0000000000000008 64'h000000000000000c 64'h0000000000000010 64'h0000000...
/tb/instruction_fetch/add_out    64'h0000000000... 64'h0000000000000004 64'h0000000000000008 64'h000000000000000c 64'h0000000000000010 64'h0000000000000014 64'h0000000...

Now 1000 ns
Cursor 1 22 ns
0 ns to 60 ns    /tb/instruction_fetch/add_out
```