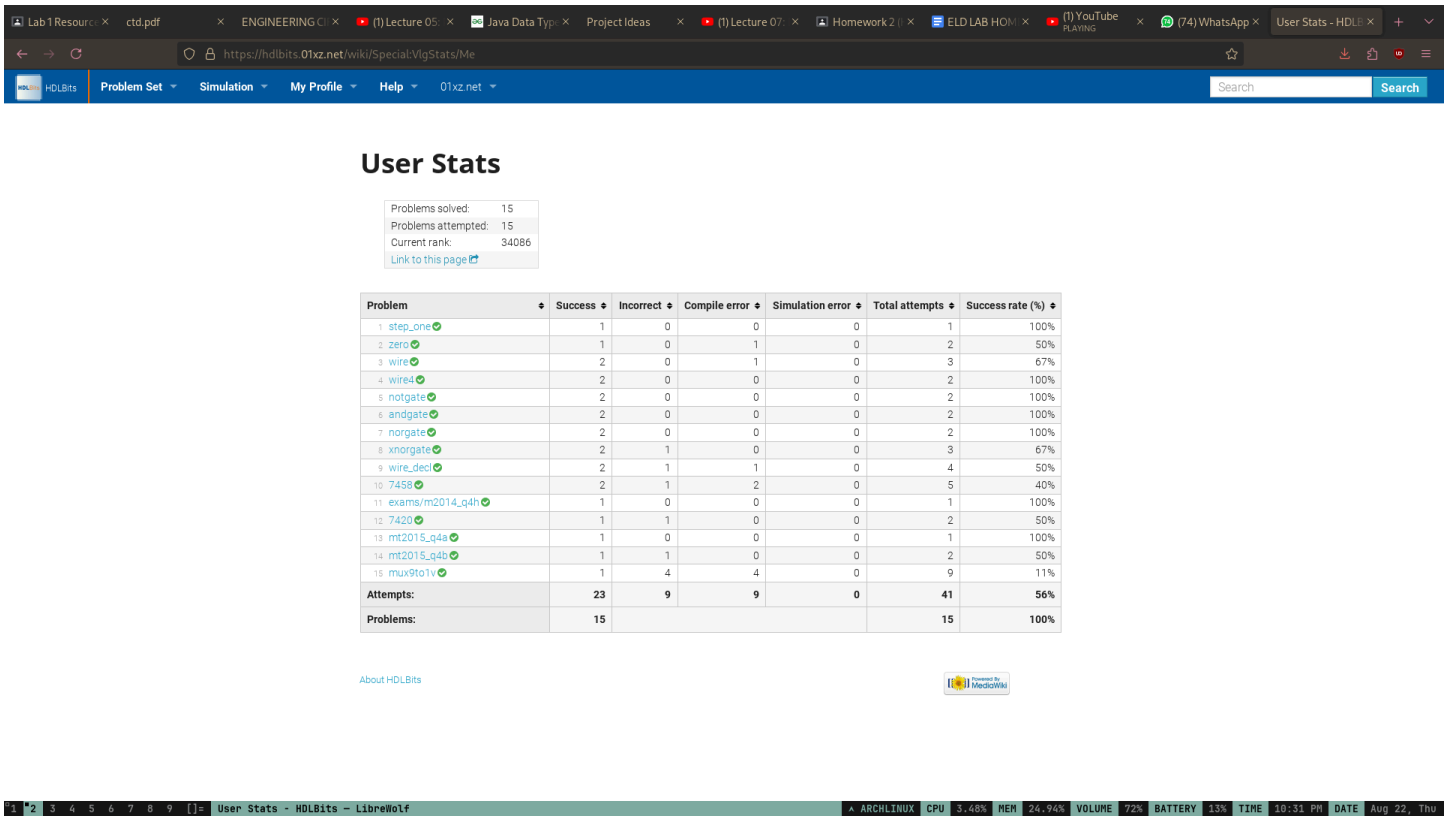
# ELD LAB HOMEWORK - 2

Aditya Gautam
2023043

- Note 1: I am on Arch Linux operating system, and hence the status bar does not resemble the one on Windows operating system
- Note 2: If the final Success Rate will be taken into account for grading, kindly take into consideration that I had attempted questions which were not mentioned in the question list.

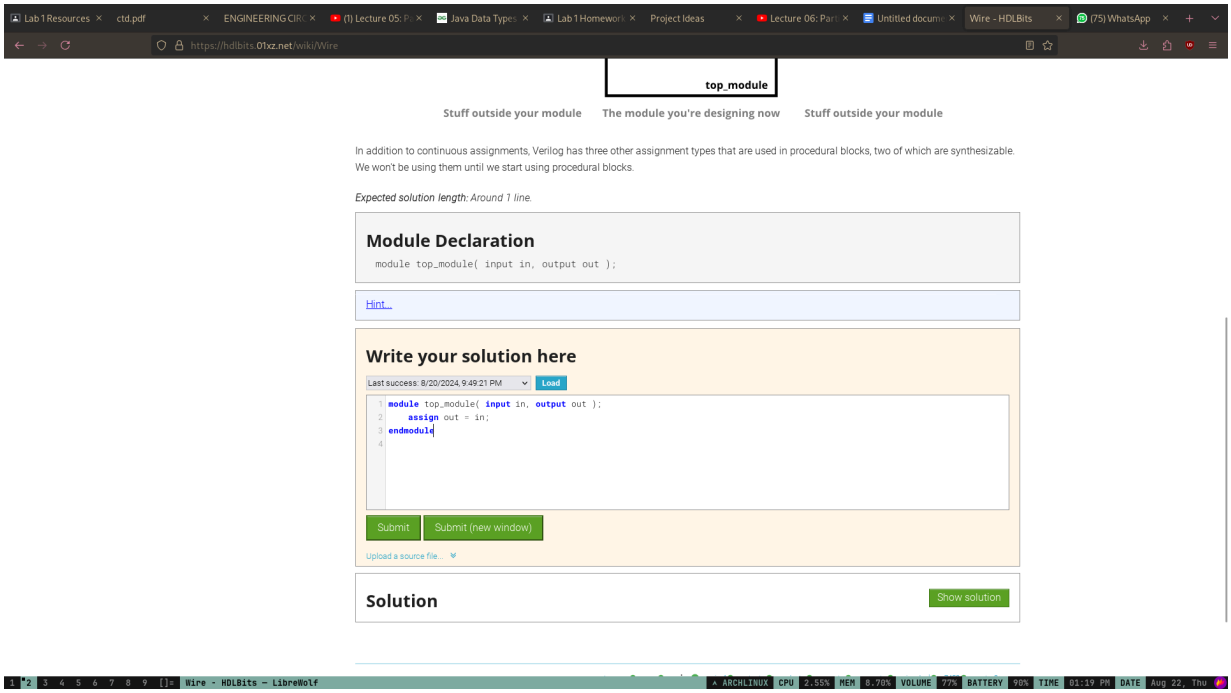
## MY STATS

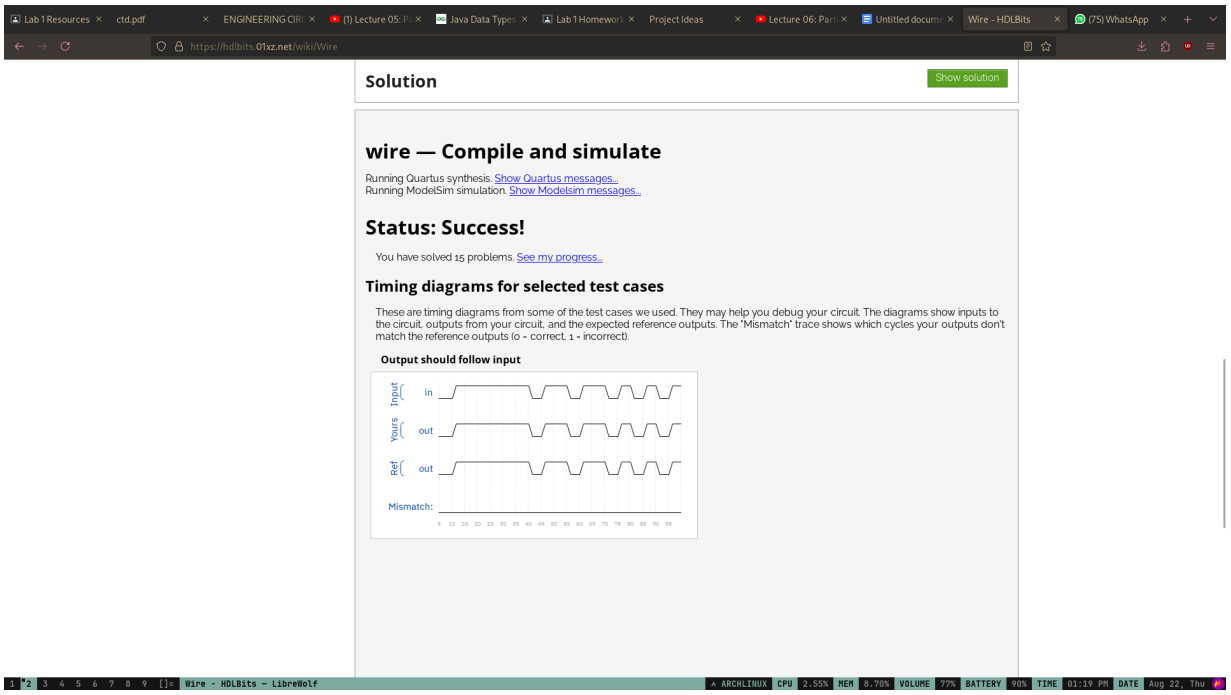
https://hdlbits.01xz.net/wiki/Special:VlgStats/EA35A63A256E325F



# Q1. Wire

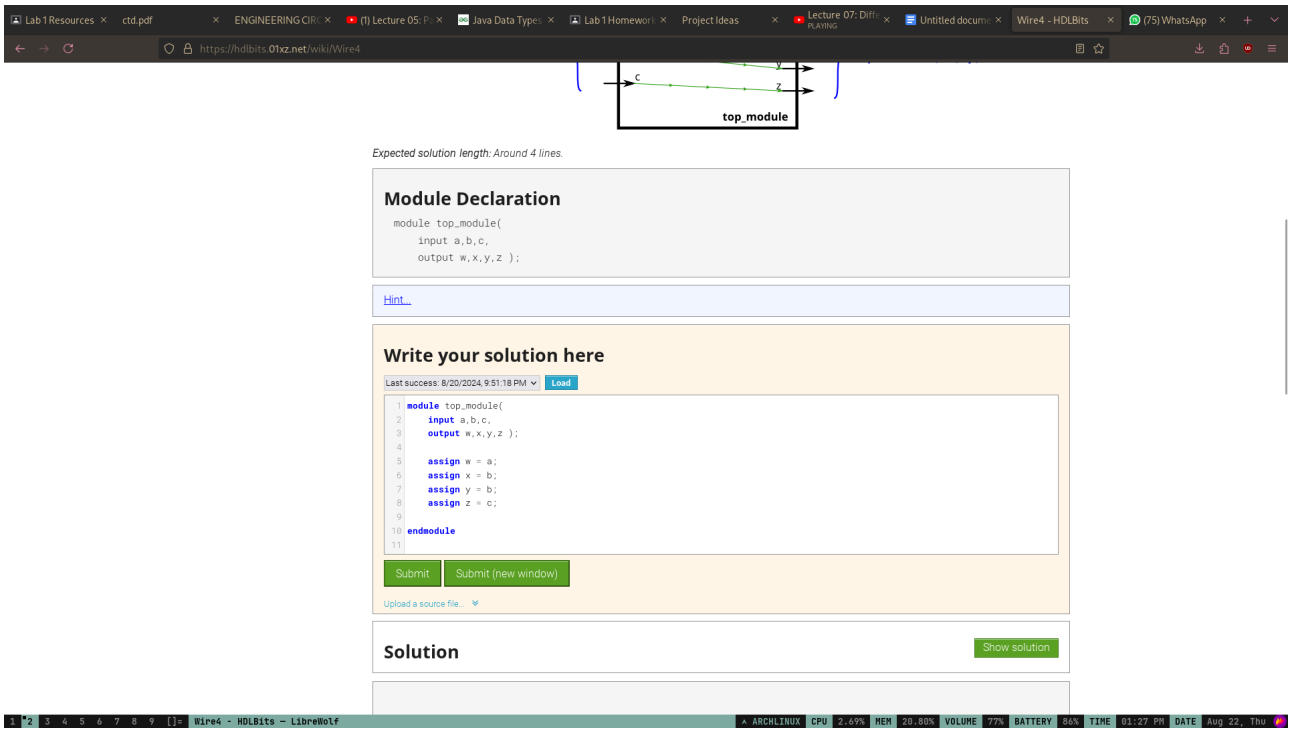
```
module top_module( input in, output out );
        assign out = in;
endmodule
```

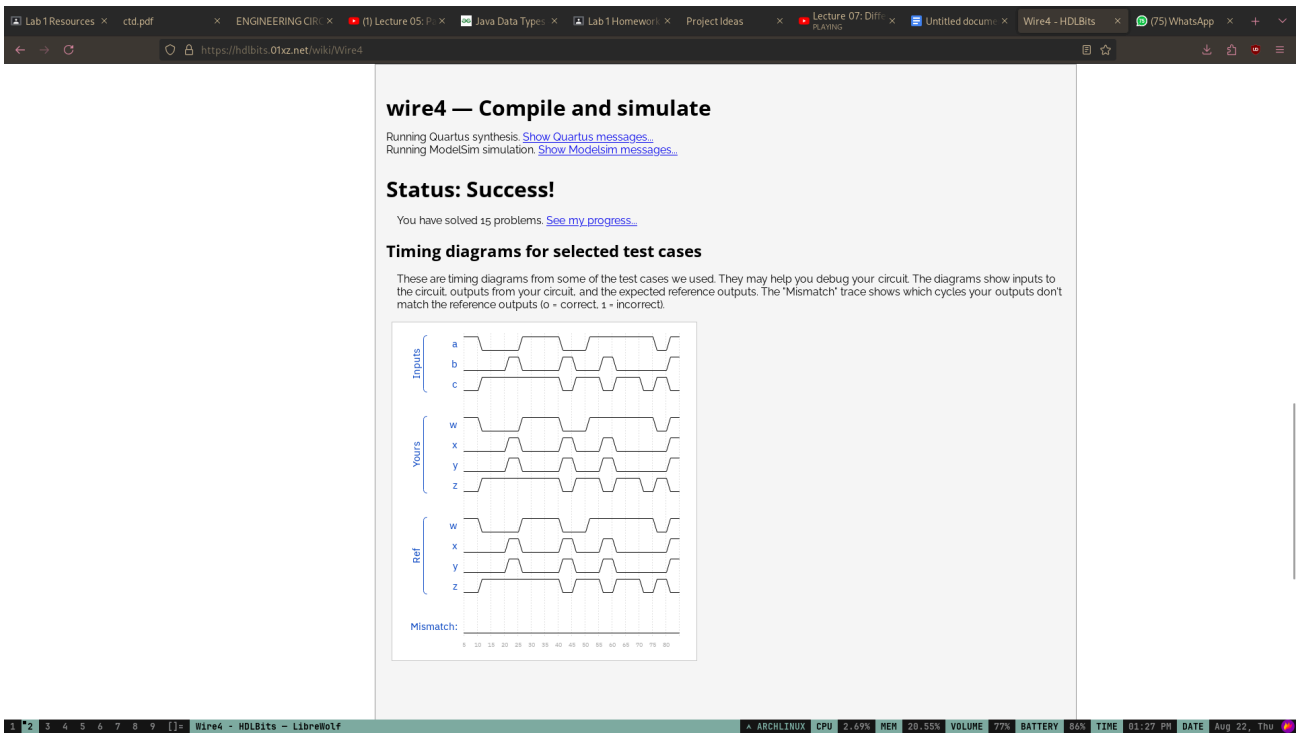


## Q2. Wire4

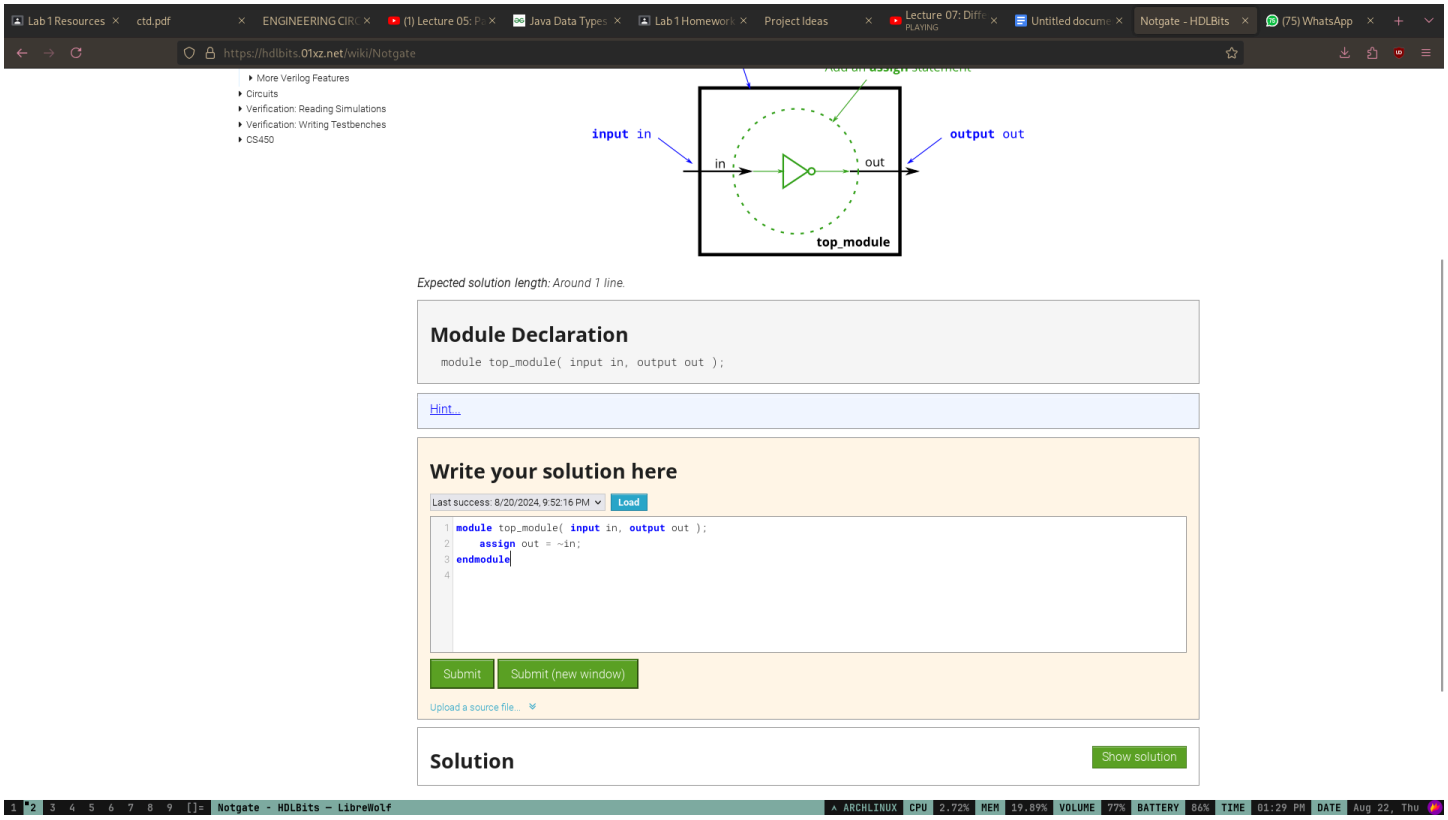
```
module top_module(
        input a,b,c,
        output w,x,y,z );
        assign w = a;
        assign x = b;
        assign y = b;
        assign z = c;
endmodule
```

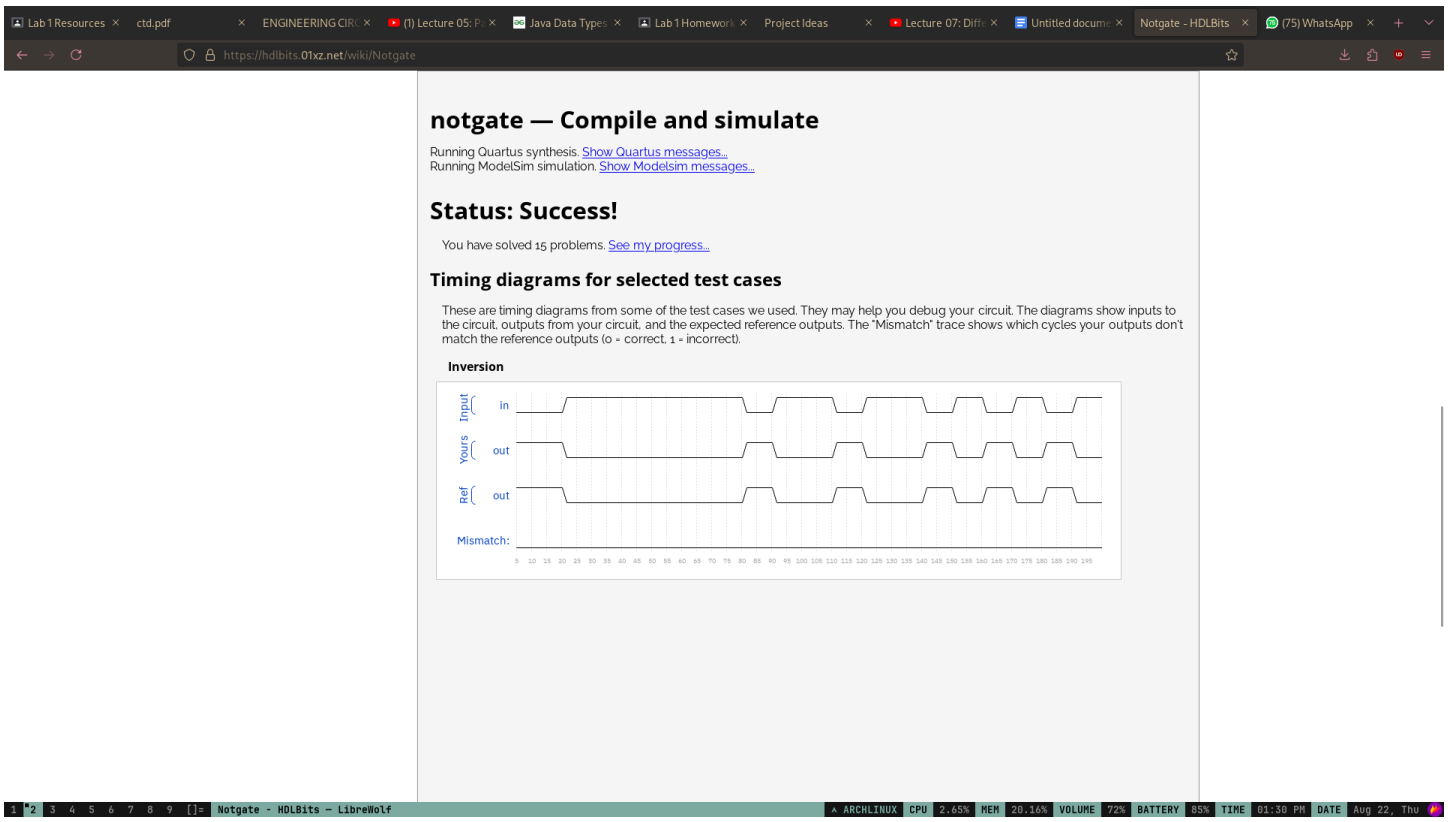


## Q3. Notgate

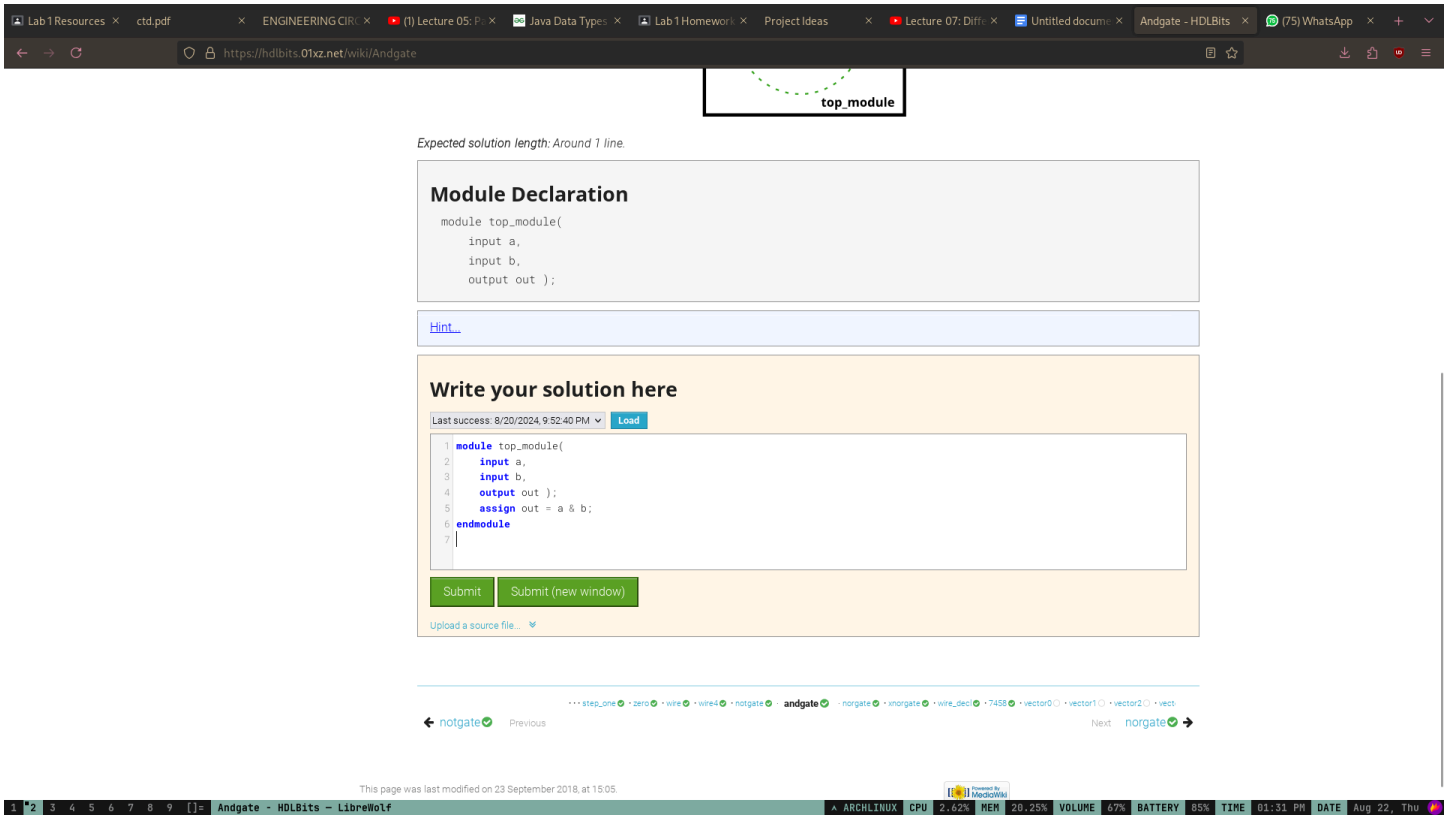
```
module top_module( input in, output out );
        assign out = ~in;
endmodule
```

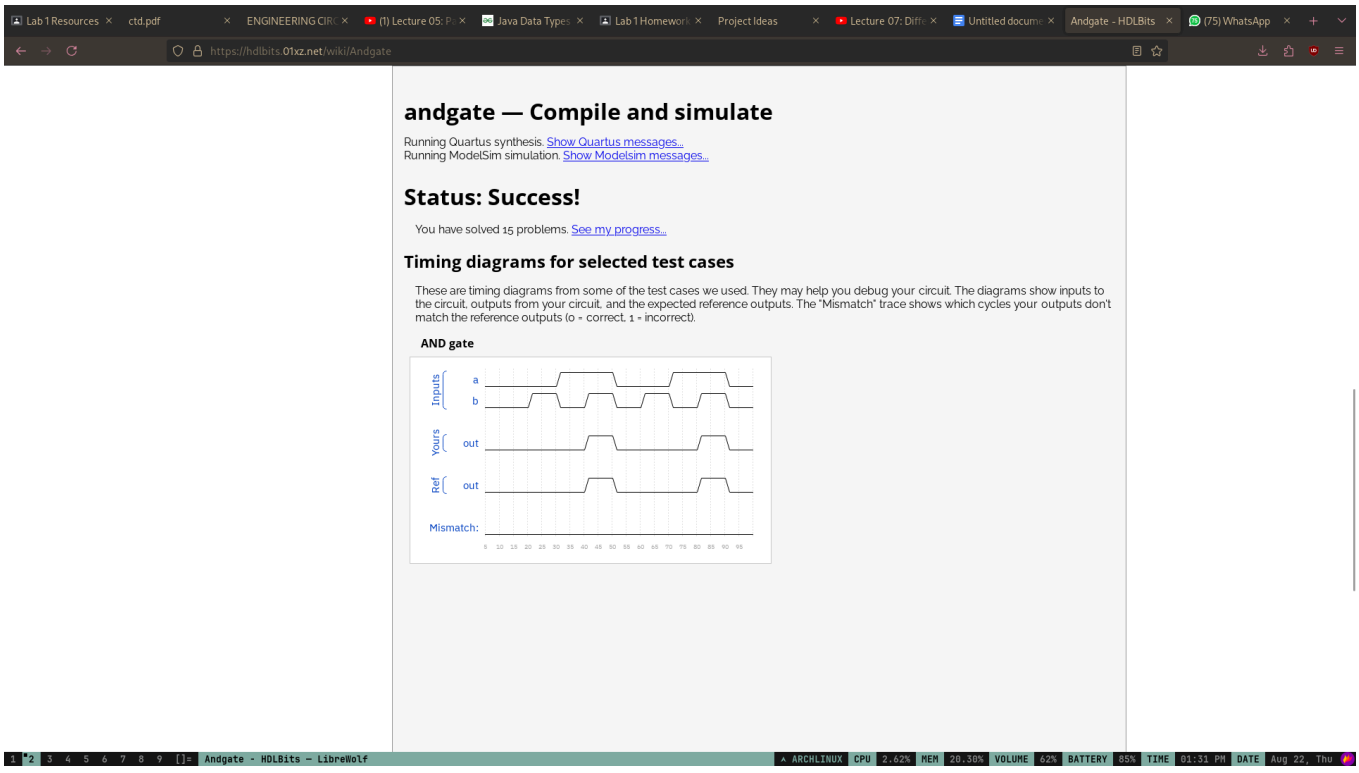


## Q4. Andgate

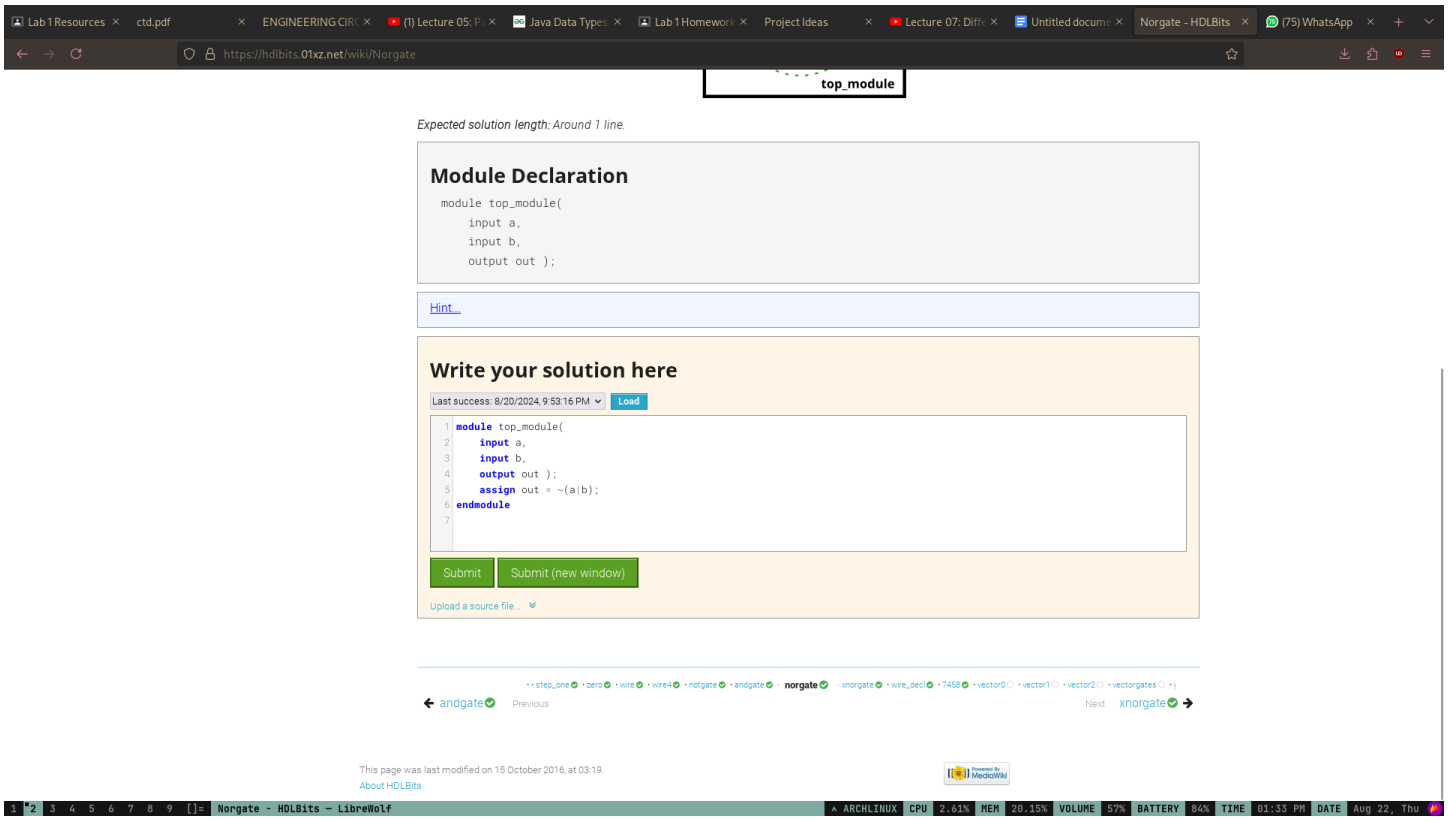
```
module top_module(
        input a,
        input b,
        output out );
        assign out = a & b;
endmodule
```

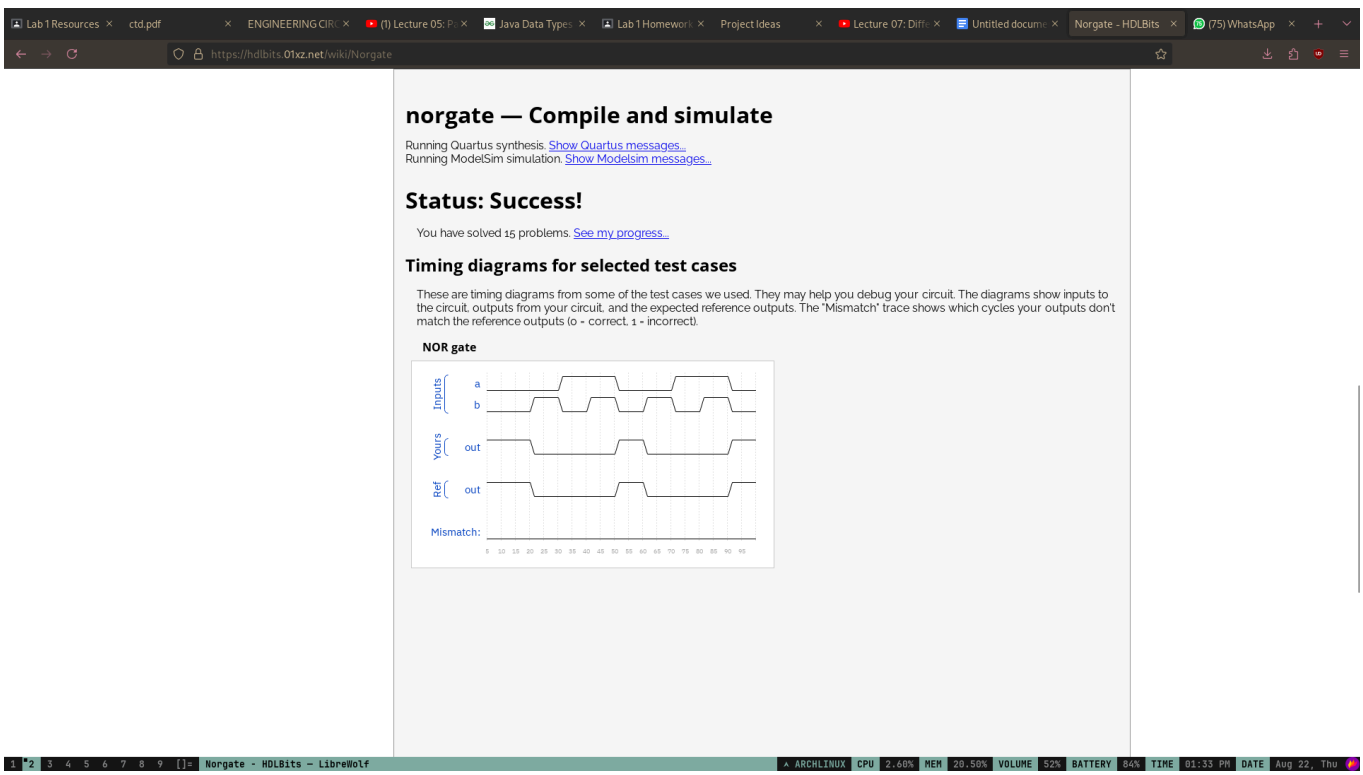


## Q5. Norgate

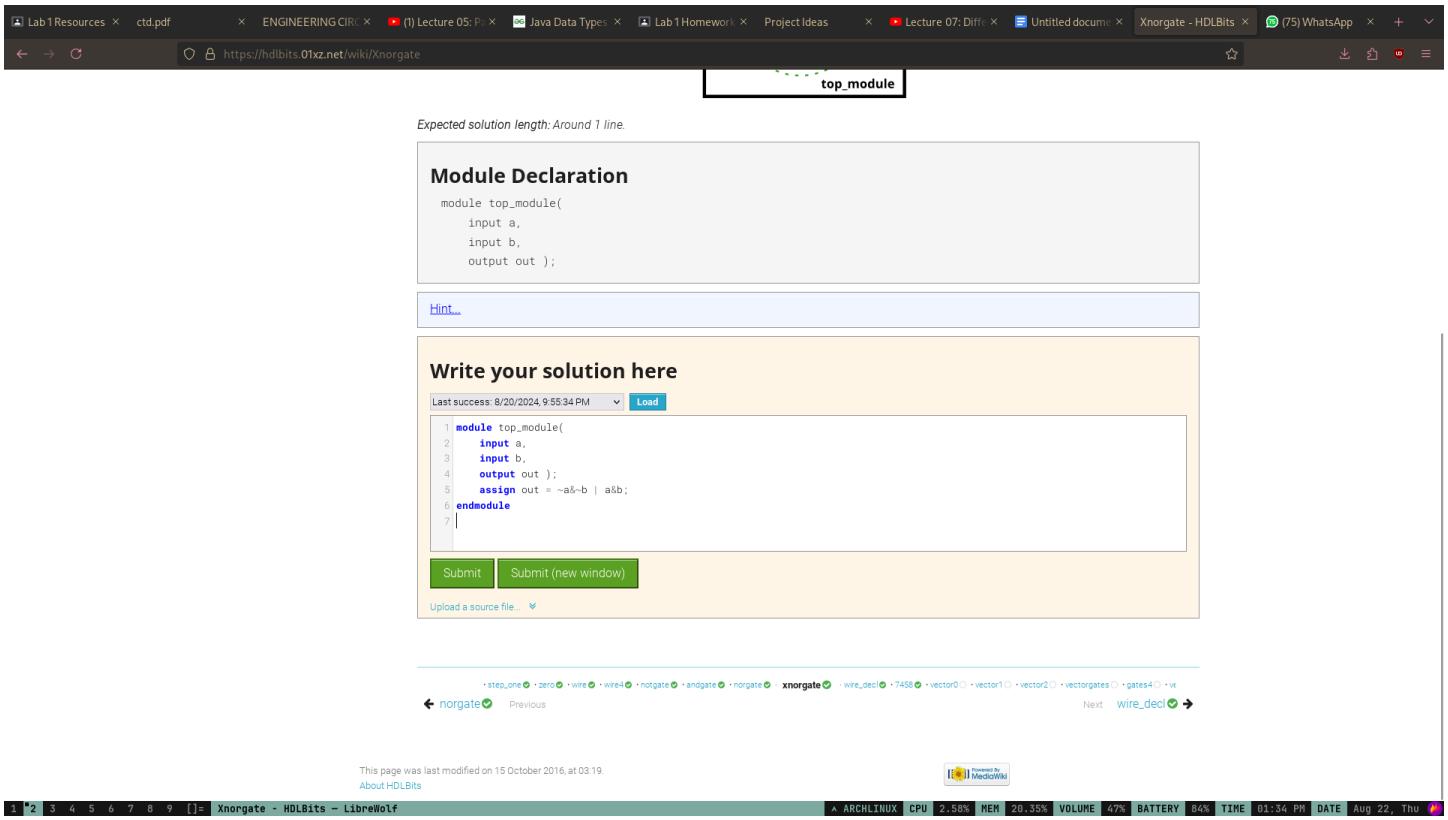
```
module top_module(
        input a,
        input b,
        output out );
        assign out = ~(a|b);
endmodule
```

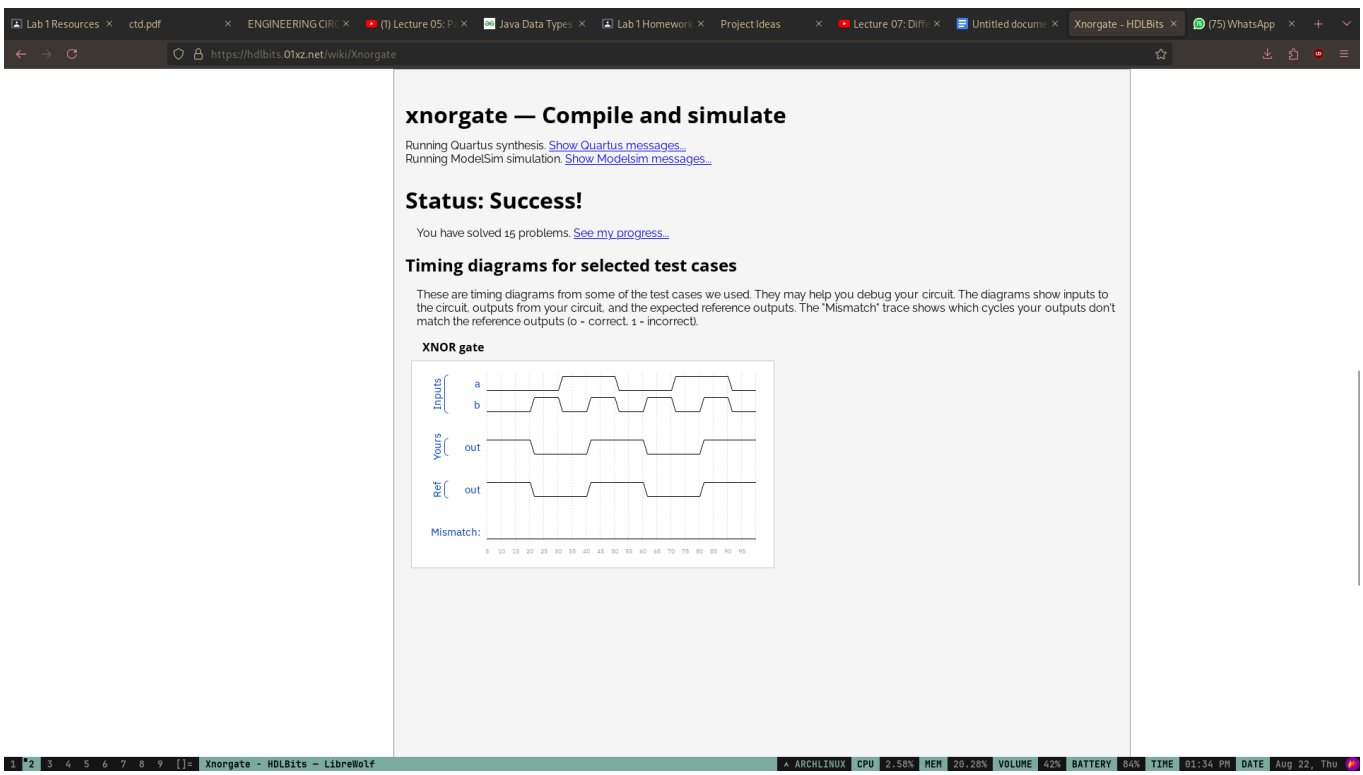


## Q6. Xnorgate

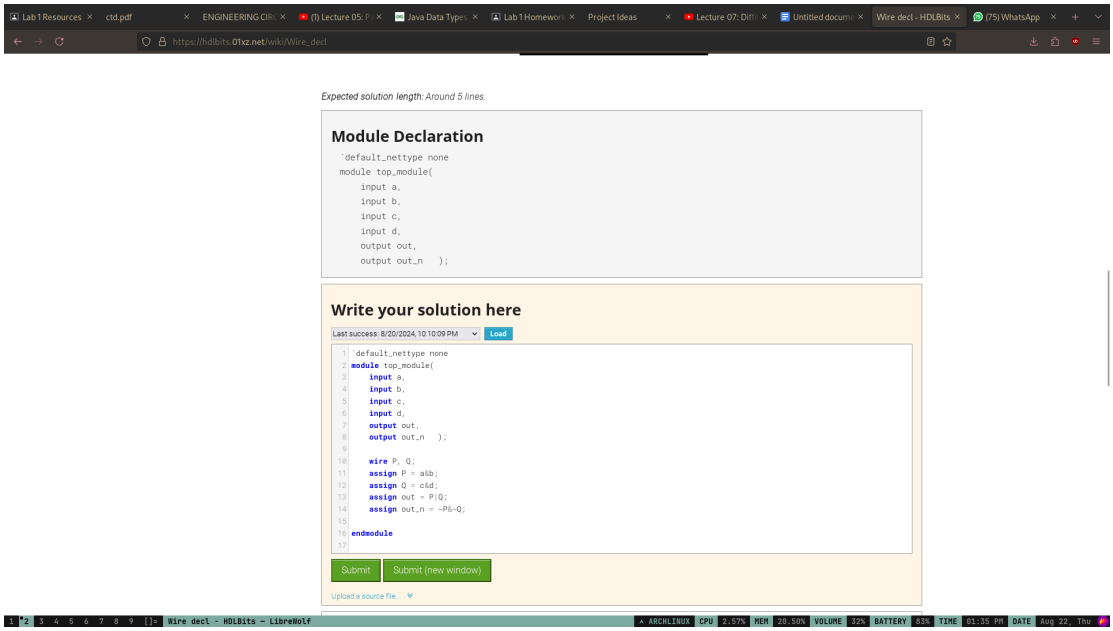
```
module top_module(
        input a,
        input b,
        output out );
        assign out = ~a&~b | a&b;
endmodule
```

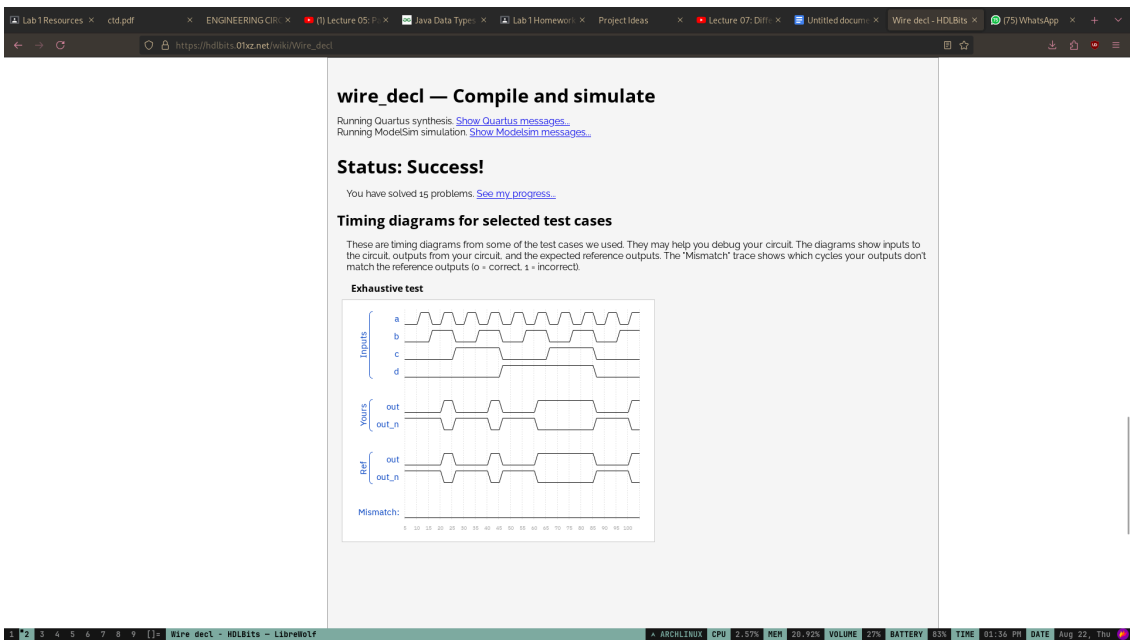


## Q7. Wire_decl

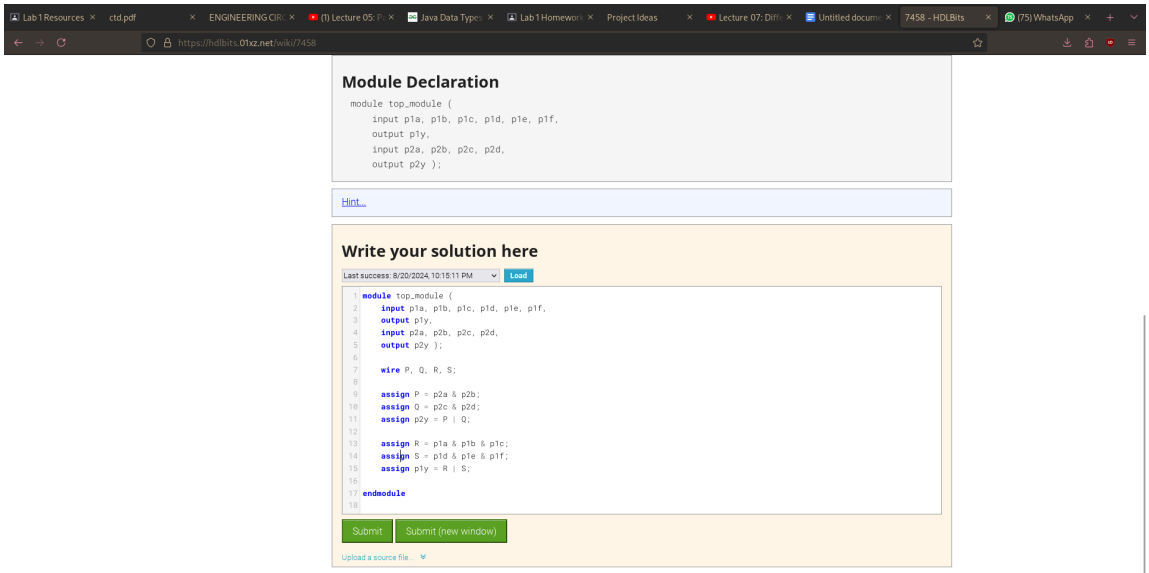
```
`default_nettype none
module top_module(
        input a,
        input b,
        input c,
        input d,
        output out,
        output out_n   );
        wire P, Q;
        assign P = a&b;
        assign Q = c&d;
        assign out = P|Q;
        assign out_n = ~P&~Q;
endmodule
```

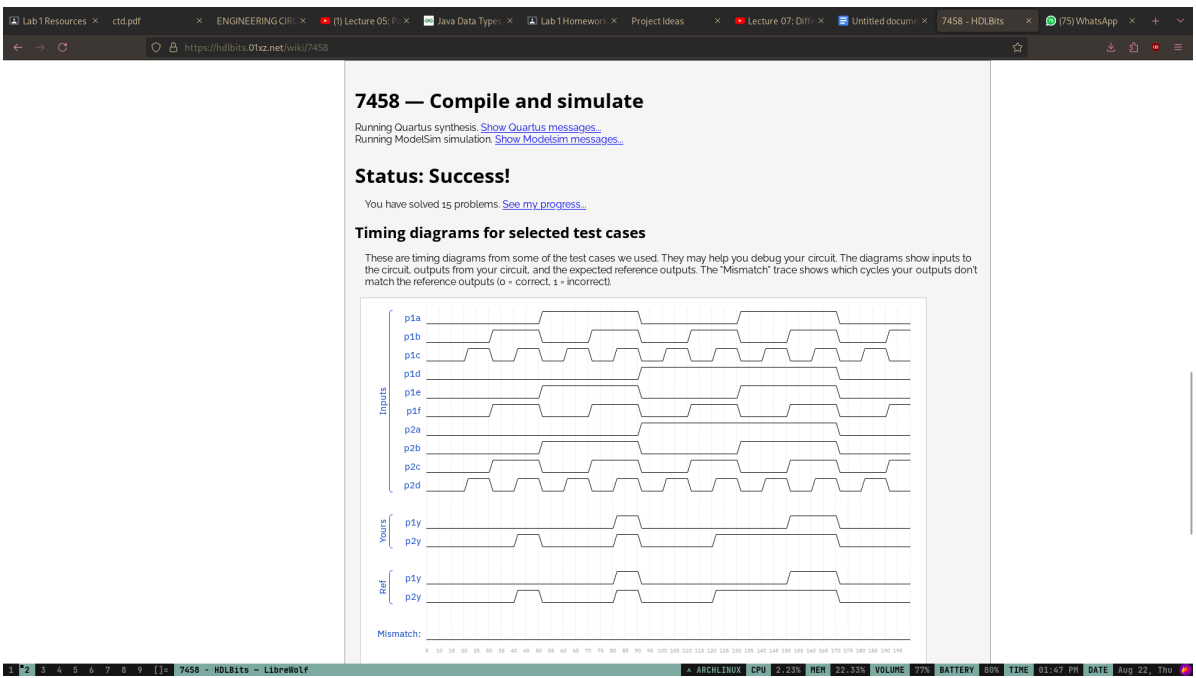


## Q8. 7458

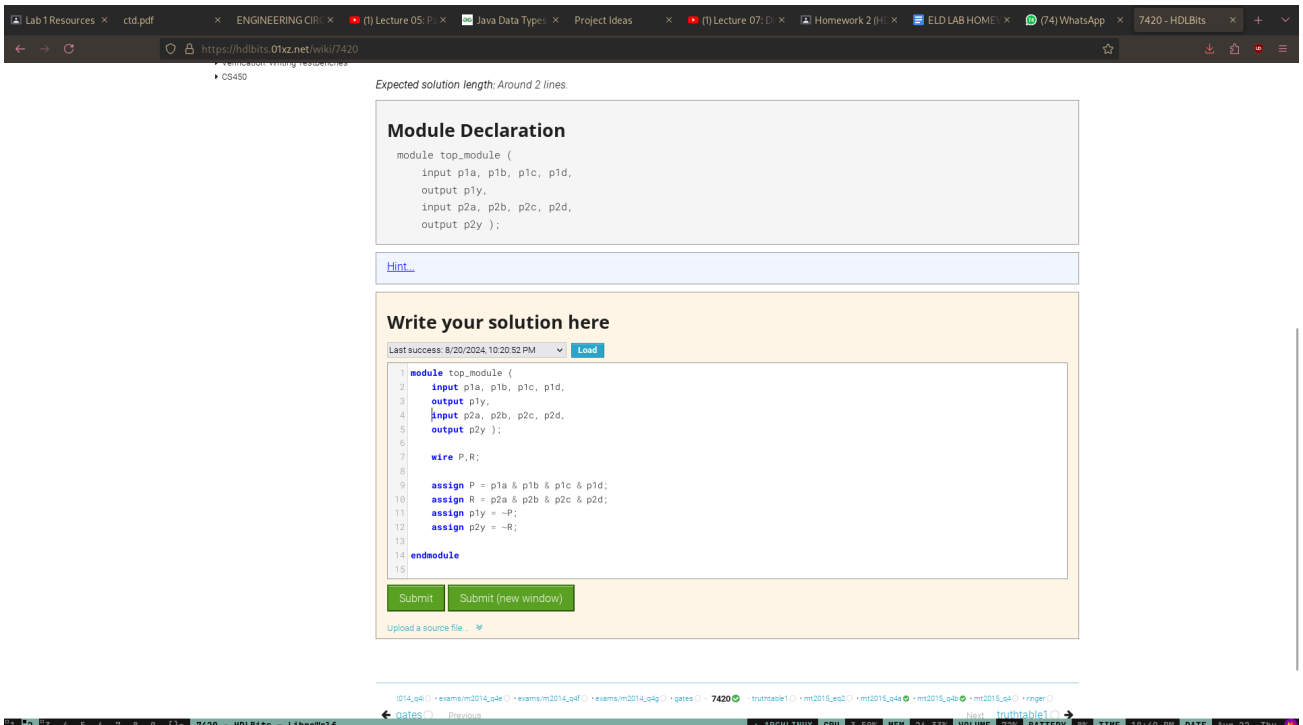
```
module top_module (
        input p1a, p1b, p1c, p1d, p1e, p1f,
        output p1y,
        input p2a, p2b, p2c, p2d,
        output p2y );
        wire P, Q, R, S;
        assign P = p2a & p2b;
        assign Q = p2c & p2d;
        assign p2y = P | Q;
        assign R = p1a & p1b & p1c;
        assign S = p1d & p1e & p1f;
        assign p1y = R | S;
endmodule
```

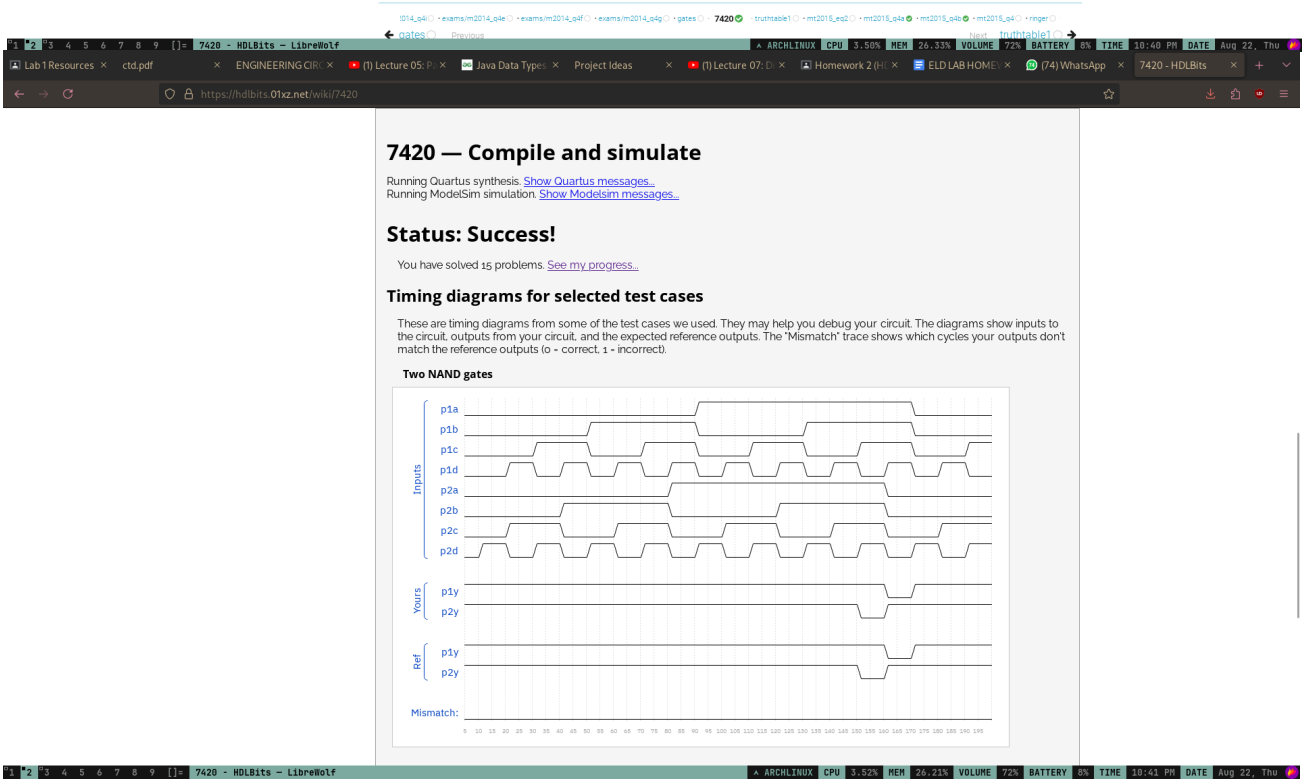


## Q9. 7420

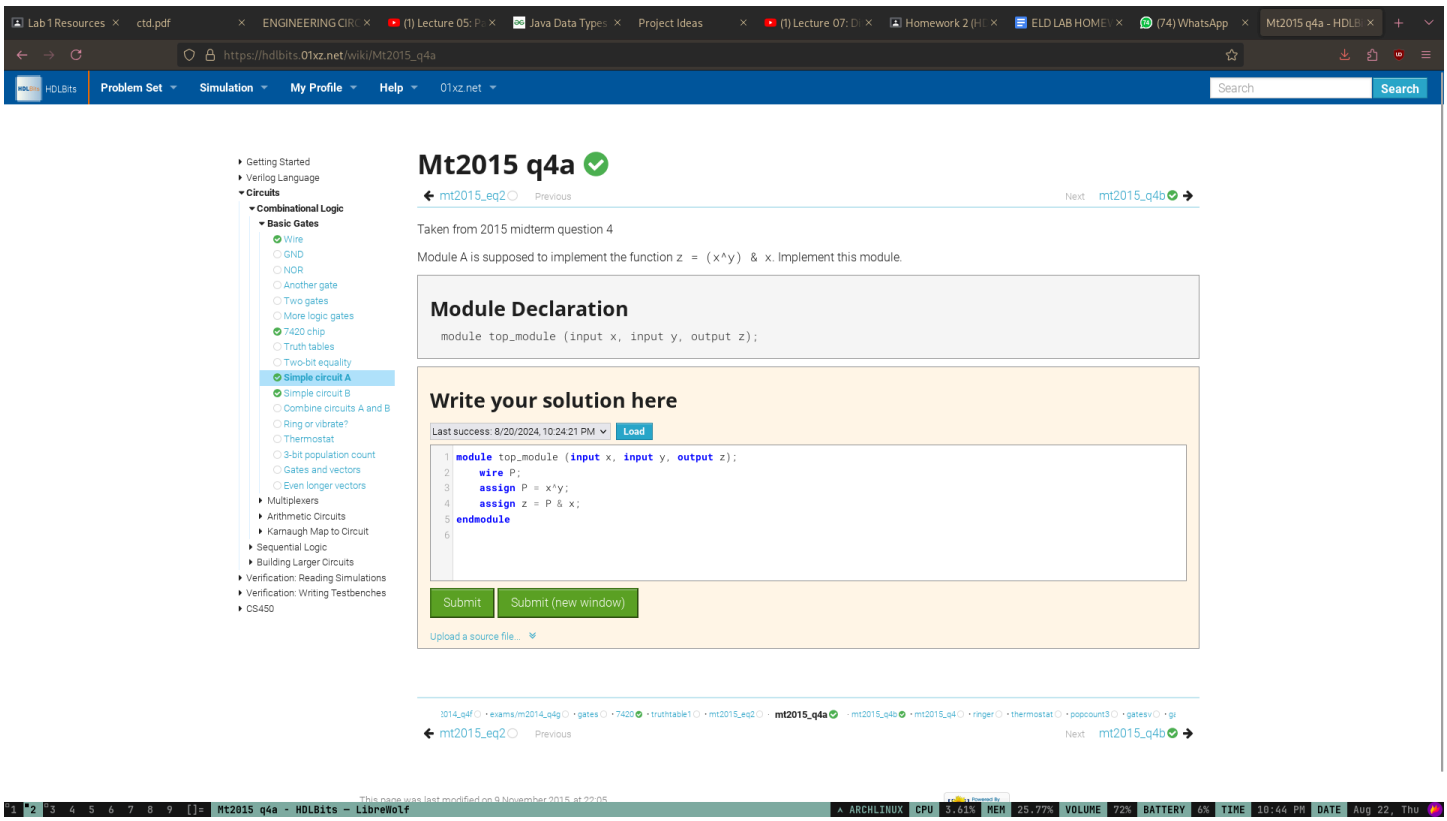
```
module top_module (
    input p1a, p1b, p1c, p1d,
    output p1y,
    input p2a, p2b, p2c, p2d,
    output p2y );
    wire P,R;
    assign P = p1a & p1b & p1c & p1d;
    assign R = p2a & p2b & p2c & p2d;
    assign p1y = ~P;
    assign p2y = ~R;
endmodule
```

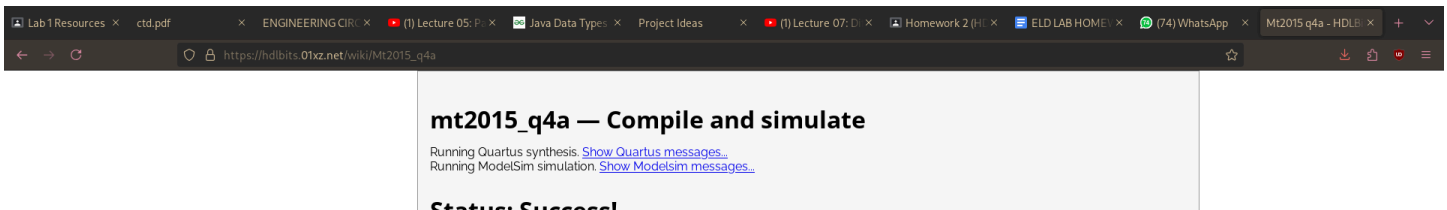


## Q10. mt2015_q4a

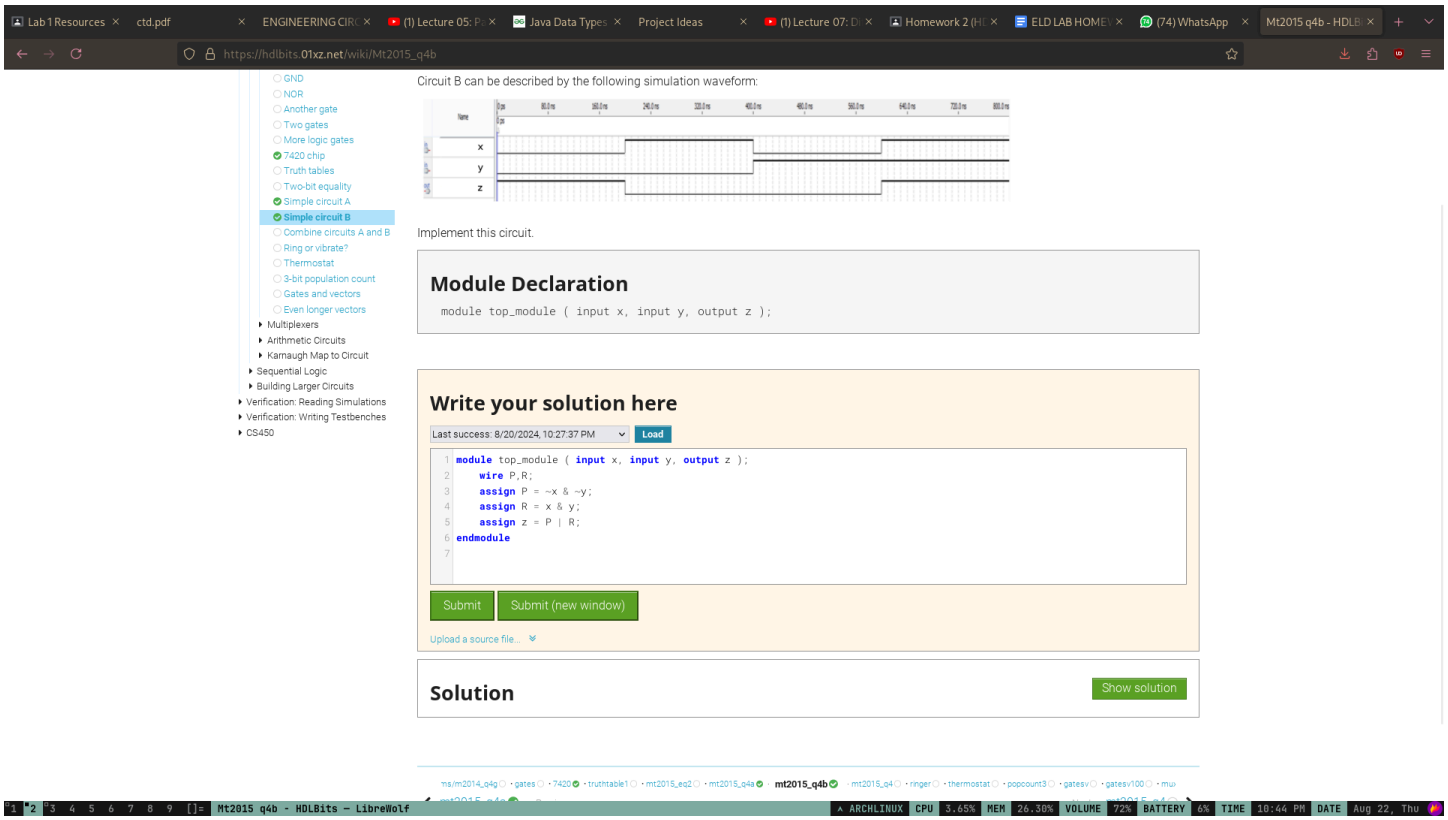
```
module top_module (input x, input y, output z);
        wire P;
        assign P = x^y;
        assign z = P & x;
endmodule
```

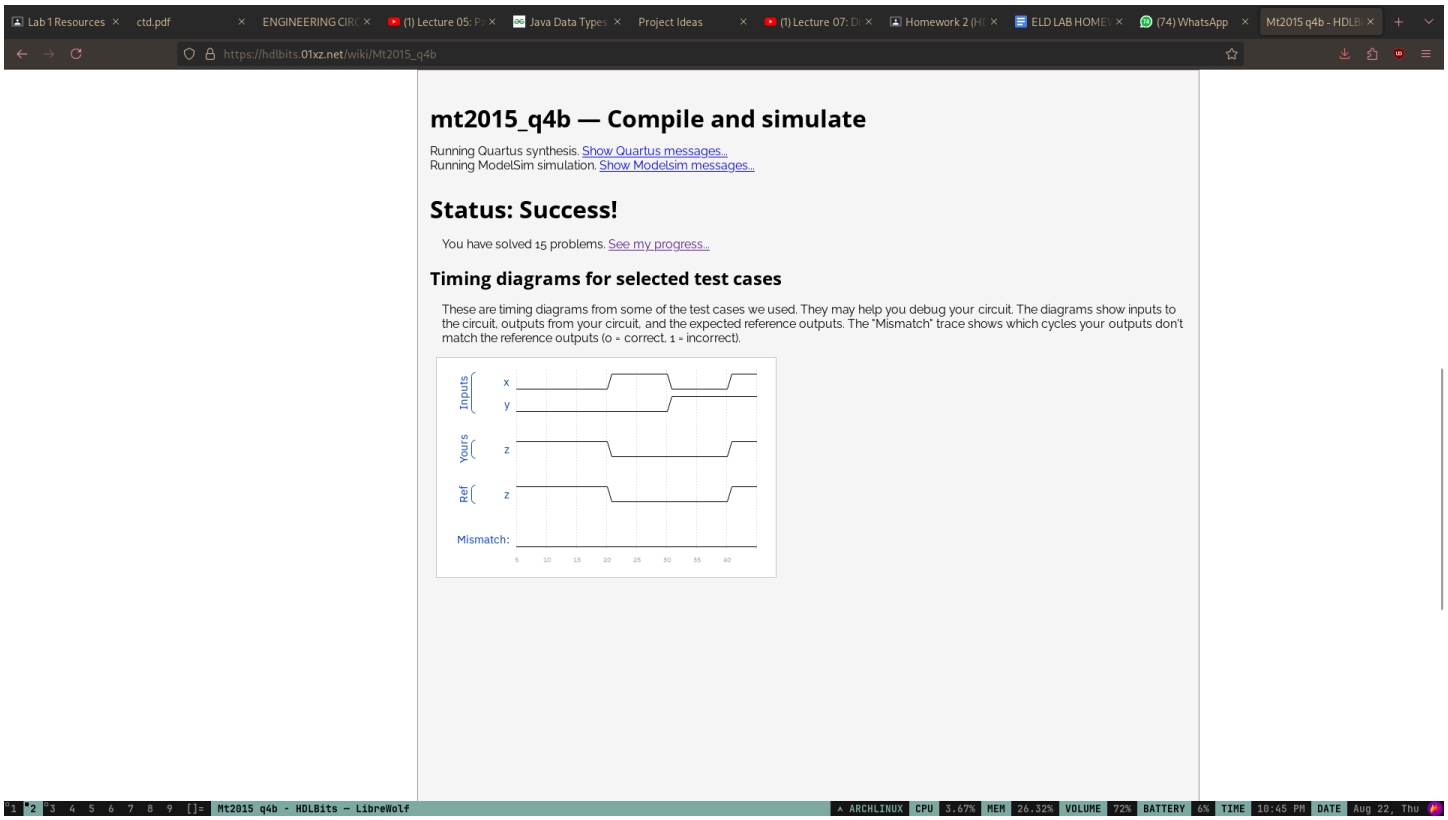


## Q11. mt2015_q4b

```
module top_module ( input x, input y, output z );
        wire P,R;
        assign P = ~x & ~y;
        assign R = x & y;
        assign z = P | R;
endmodule
```



**Q12. mux9to1v**

```
module top_module(
	input [15:0] a, b, c, d, e, f, g, h, i,
	input [3:0] sel,
	output [15:0] out );

	always @(*) begin
	if(sel == 4'b0000)
	out = a;
	else if(sel == 4'b0001)
	out = b;
	else if(sel == 4'b0010)
	out = c;
	else if(sel == 4'b0011)
	out = d;
	else if(sel == 4'b0100)
	out = e;
	else if(sel == 4'b0101)
	out = f;
	else if(sel == 4'b0110)
	out = g;
	else if(sel == 4'b0111)
	out = h;
	else if(sel == 4'b1000)
	out = i;
	else
	out = {16{1'b1}};
	end

endmodule
```

```
    input [3:0] sel,
    output [15:0] out );
```

Hint...

## Write your solution here

Last success: 8/20/2024, 10:41:32 PM Load

```
module top_module(
    input [15:0] a, b, c, d, e, f, g, h, i,
    input [3:0] sel,
    output [15:0] out );

    always @(*) begin
        if(sel == 4'b0000)
            out = a;
        else if(sel == 4'b0001)
            out = b;
        else if(sel == 4'b0010)
            out = c;
        else if(sel == 4'b0011)
            out = d;
        else if(sel == 4'b0100)
            out = e;
        else if(sel == 4'b0101)
            out = f;
        else if(sel == 4'b0110)
            out = g;
        else if(sel == 4'b0111)
            out = h;
        else if(sel == 4'b1000)
            out = i;
        else
            out = {16{1'b1}};
    end

endmodule

```

Submit | Submit (new window)

Upload a source file...





# mux9to1v — Compile and simulate

Running Quartus synthesis. Show Quartus messages...
Running ModelSim simulation. Show Modelsim messages...

## Status: Success!

You have solved 15 problems. See my progress...

### Timing diagrams for selected test cases

These are timing diagrams from some of the test cases we used. They may help you debug your circuit. The diagrams show inputs to the circuit, outputs from your circuit, and the expected reference outputs. The "Mismatch" trace shows which cycles your outputs don't match the reference outputs (0 = correct, 1 = incorrect).