DAY-3
#100DAYSOFRTL

Problem statement :-

1. Create one instance of module mod_a, then connect the module's three pins (in1, in2, and out) to your top-level module's three ports (wires a, b, and out). The module mod_a is provided for you — you must instantiate it.

Write your solution here

[Load a previous submission] Load

```
1 module top_module ( input a, input b, output out );
2
3     mod_a in1(.out(out),.in1(a),.in2(b));
4
5 endmodule
6
```

Submit | Submit (new window)

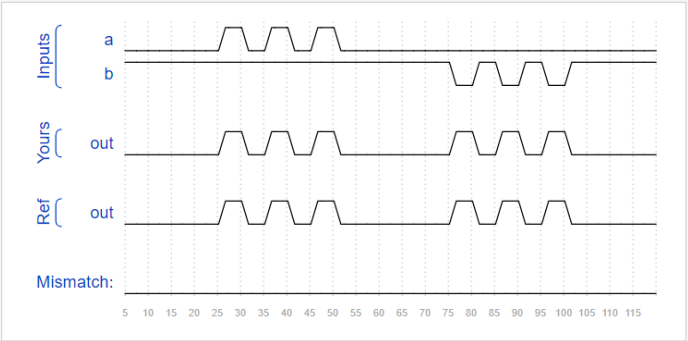
Upload a source file...

Status: Success!

You have solved 20 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).



2. This problem is similar to the previous one You are given a module named `mod_a` that has 2 outputs and 4 inputs, in that order. You must connect the 6 ports *by position* to your top-level module's ports `out1`, `out2`, a, b, c, and d, in that order. You are given the following module:

```
module mod_a ( output, output, input, input, input, input );
```

Write your solution here

[Load a previous submission] Load

```
module top_module (
    input a,
    input b,
    input c,
    input d,
    output out1,
    output out2
);
    mod_a(out1,out2,a,b,c,d);

endmodule
```

Submit | Submit (new window)

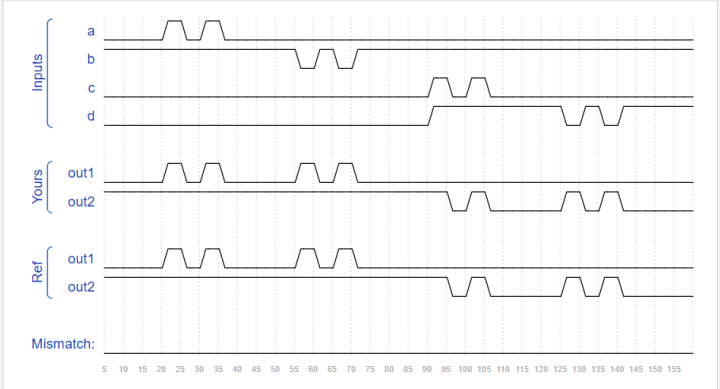
Upload a source file...

Status: Success!

You have solved 21 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).



3. Given a module named mod_a that has 2 outputs and 4 inputs, in some order. You must connect the 6 ports by name to your top-level module's ports: module mod_a (output out1, output out2, input in1, input in2, input in3, input in4)

Write your solution here

[Load a previous submission] Load

```
module top_module (
    input a,
    input b,
    input c,
    input d,
    output out1,
    output out2
);

    mod_a(.out1(out1),.out2(out2),.in1(a),.in2(b),.in3(c),.in4(d));

endmodule
```

Submit | Submit (new window)

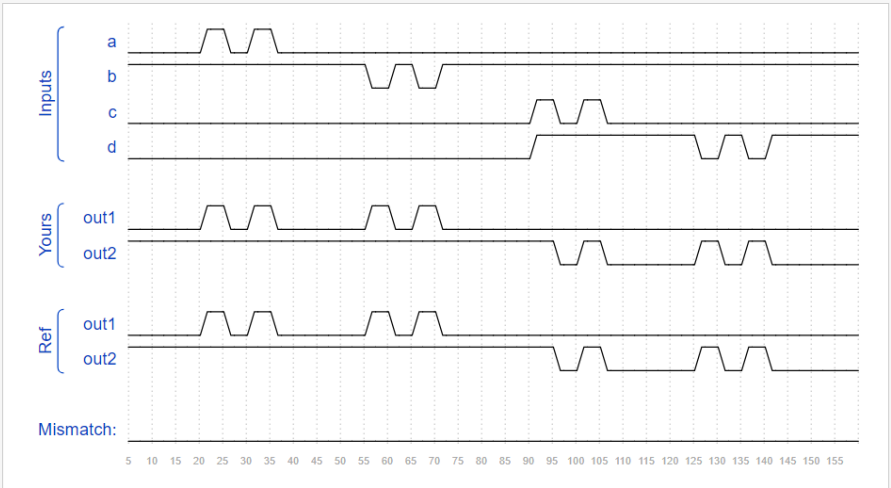
Upload a source file...

Status: Success!

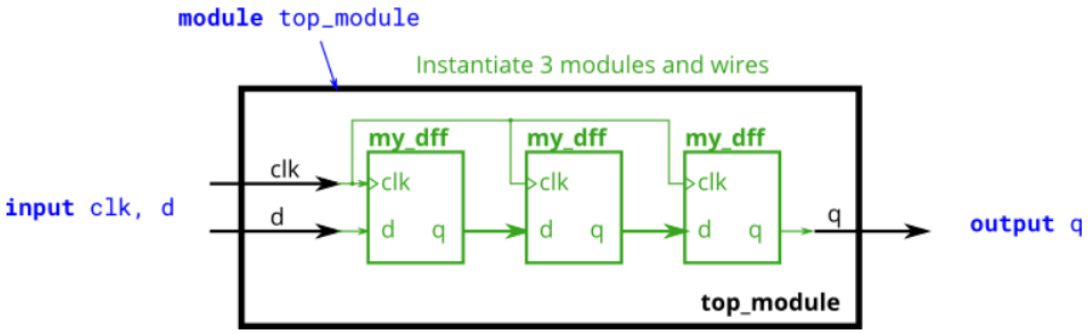
You have solved 22 problems. See my progress...

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).



4. You are given a module `my_dff` with two inputs and one output (that implements a D flip-flop). Instantiate three of them, then chain them together to make a shift register of length 3. The `clk` port needs to be connected to all instances.The module provided to you is: `module my_dff ( input clk, input d, output q );`



Write your solution here

[Load a previous submission] Load

```
module top_module ( input clk, input d, output q );

    wire d1_out,d2_out;

    my_dff d1(.clk(clk),.d(d),.q(d1_out));
    my_dff d2(.clk(clk),.d(d1_out),.q(d2_out));
    my_dff d3(.clk(clk),.d(d2_out),.q(q));

endmodule
```

Submit | Submit (new window)

Upload a source file...

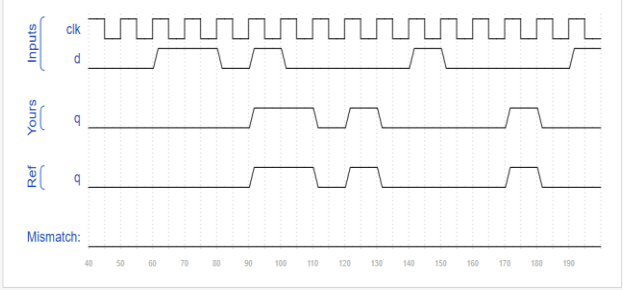
Status: Success!

You have solved 23 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).

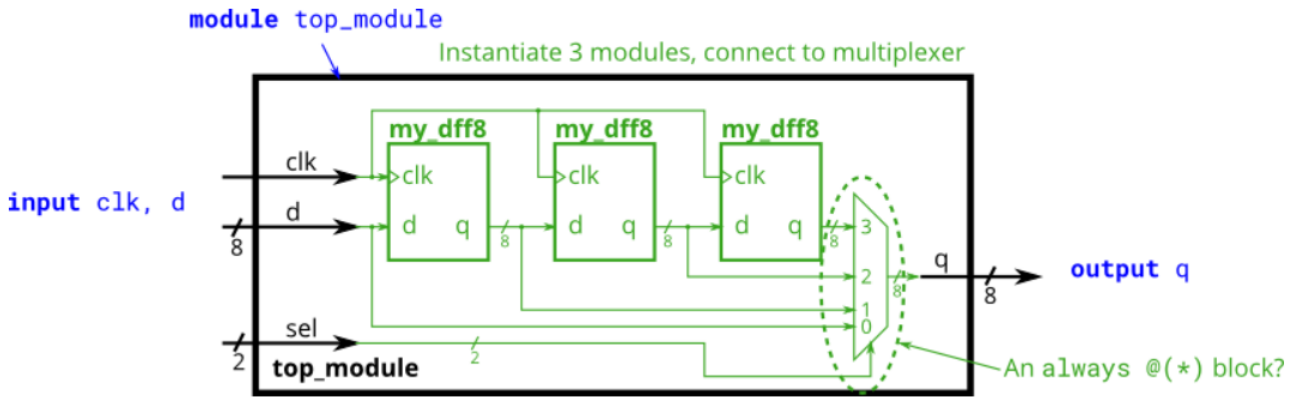
Shift register



5. This exercise is an extension of module_shift. Instead of module ports being only single pins, we now have modules with vectors as ports, to which you will attach wire vectors instead of plain wires. Like everywhere else in Verilog, the vector length of the port does not have to match the wire connecting to it, but this will cause zero-padding or trucation of the vector. This exercise does not use connections with mismatched vector lengths.

   You are given a module `my_dff8` with two inputs and one output (that implements a set of 8 D flip-flops). Instantiate three of them, then chain them together to make a 8-bit wide shift register of length 3. In addition, create a 4-to-1 multiplexer (not provided) that chooses what to output depending on `sel[1:0]`: The value at the input d, after the first, after the second, or after the third D flip-flop. (Essentially, `sel` selects how many cycles to delay the input, from zero to three clock cycles.)

   The module provided to you is: `module my_dff8 ( input clk, input [7:0] d, output [7:0] q );`



Write your solution here

[Load a previous submission] Load

```
module top_module (
    input clk,
    input [7:0] d,
    input [1:0] sel,
    output [7:0] q
);

    wire[7:0]w1,w2,w3;

    my_dff8 d1(clk,d,w1);
    my_dff8 d2(clk,w1,w2);
    my_dff8 d3(clk,w2,w3);

    always@(*)
        begin
            case (sel)
                2'b00:q=d;
                2'b01:q=w1;
                2'b10:q=w2;
                2'b11:q=w3;
            endcase
        end


endmodule

```

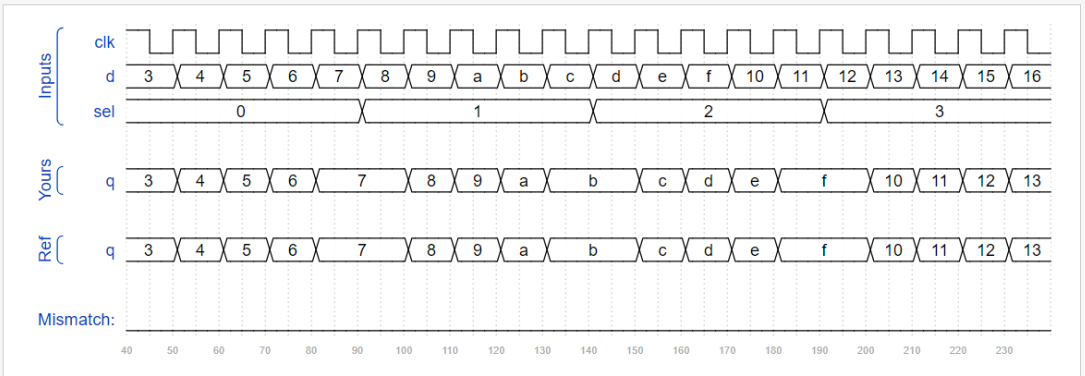
Status: Success!

You have solved 24 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).

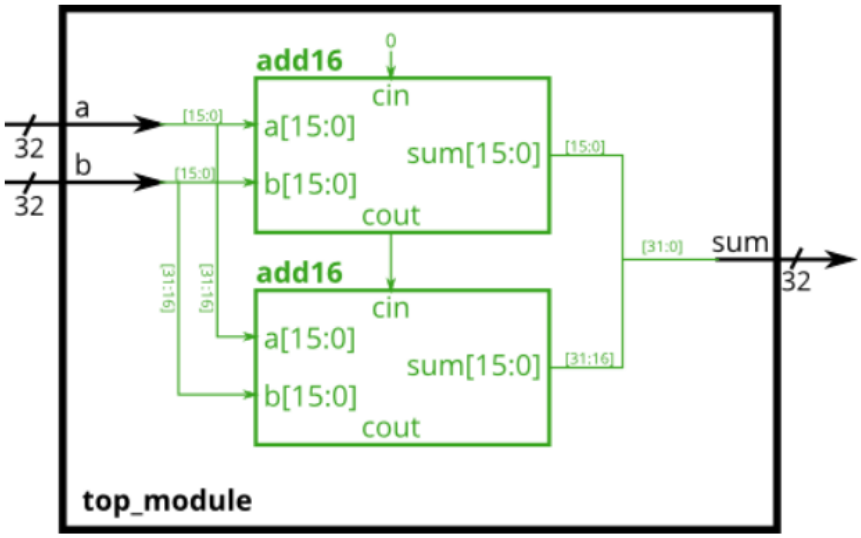
Shift register



6. You are given a module `add16` that performs a 16-bit addition. Instantiate two of them to create a 32-bit adder. One add16 module computes the lower 16 bits of the addition result, while the second add16 module computes the upper 16 bits of the result, after receiving the carry-out from the first adder. Your 32-bit adder does not need to handle carry-in (assume 0) or carry-out (ignored), but the internal modules need to in order to function correctly. (In other words, the `add16` module performs 16-bit a + b + cin, while your module performs 32-bit a + b).

   Connect the modules together as shown in the diagram below. The provided module `add16` has the following declaration:

```
module add16 ( input[15:0] a,
input[15:0] b, input cin, output[15:0] sum,
output cout );
```



Write your solution here

Last success: 10/9/2023, 8:10:03 PM Load

```
module top_module(
    input [31:0] a,
    input [31:0] b,
    output [31:0] sum
);
    wire carry;
    add16 block1(.a(a[15:0]), .b(b[15:0]), .cin(1'b0), .sum(sum[15:0]), .cout(carry));
    add16 block2(.a(a[31:16]), .b(b[31:16]), .cin(carry), .sum(sum[31:16]) );
endmodule

```

Submit Submit (new window)

Status: Success!

You have solved 26 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).

32-bit adder