DAY-44
#100DAYSOFRTL

PROBLEM STATEMENT:--

1. Create 16 D flip-flops. It's sometimes useful to only modify parts of a group of flip-flops. The byte-enable inputs control whether each byte of the 16 registers should be written to on that cycle. `byteena[1]` controls the upper byte `d[15:8]`, while `byteena[0]` controls the lower byte `d[7:0]`.

```
module top_module (
    input clk,
    input resetn,
    input [1:0] byteena,
    input [15:0] d,
    output reg [15:0] q
);

    always @(posedge clk) begin
        if (!resetn) begin
            q <= 16'b0;
        end
        else
            begin
            if (byteena[0])
                begin
                q[7:0] <= d[7:0];
            end

            if (byteena[1])
                begin
                q[15:8] <= d[15:8];
            end
        end
    end

endmodule

```

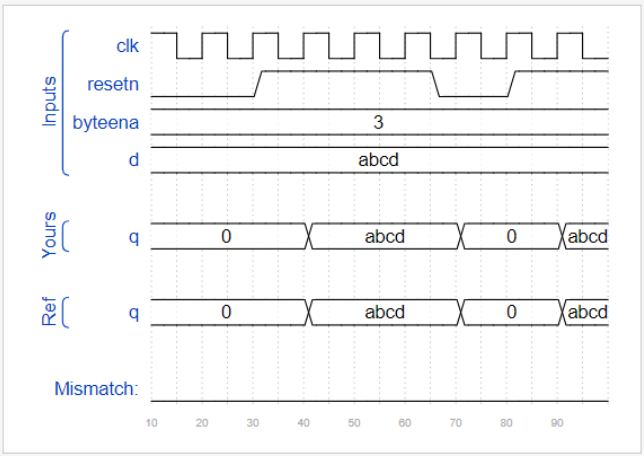
Status: Success!

You have solved 74 problems. See my progress...

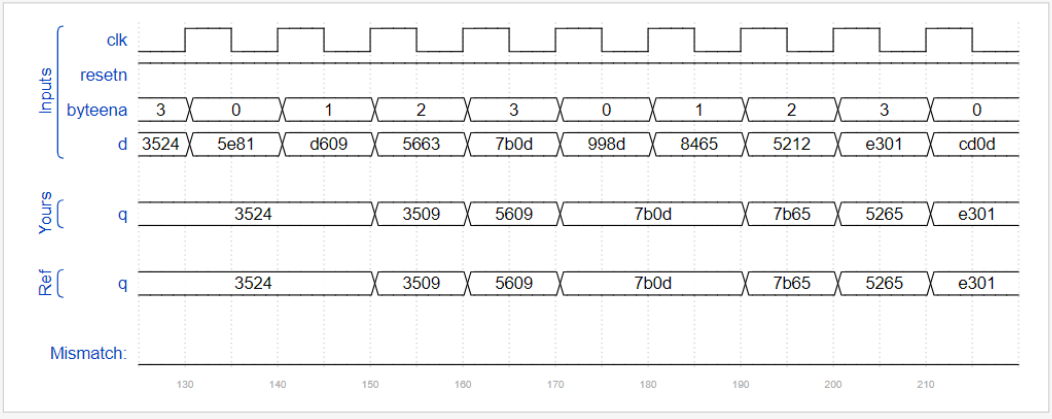
Timing diagrams for selected test cases

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).

Synchronous active-low reset



DFF with byte enables



2.

Implement the following circuit:

Note that this is a latch, so a Quartus warning about having inferred a latch is expected.

Write your solution here

[Load a previous submission] Load

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


    always@(*)

        if(ena)
            q<=d;

endmodule

```

Submit | Submit (new window)

Upload a source file...

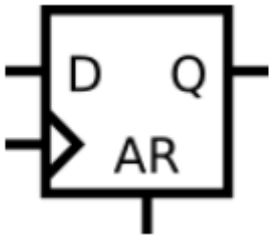
exams/m2014_q4a — Compile and simulate

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

Status: Success!

You have solved 75 problems. See my progress...

3. Implement the following circuit:



Write your solution here

[Load a previous submission] Load

```
module top_module (
    input clk,
    input d,
    input ar,   // asynchronous reset
    output q);

    always@(posedge clk or posedge ar)
        begin
            if(ar)
                q<=0;
            else
                q<=d;

        end
endmodule
```

Submit | Submit (new window)

Upload a source file...

exams/m2014_q4b — Compile and simulate

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

Status: Success!

You have solved 76 problems. See my progress...