DAY-43
#100DAYSOFRTL

PROBLEM STATEMENT:--

1. A D flip-flop is a circuit that stores a bit and is updated periodically, at the (usually) positive edge of a clock signal. Create a single D flip-flop.

Write your solution here

[Load a previous submission] Load

```
module top_module (
    input clk,    // Clocks are used in sequential circuits
    input d,
    output reg q );//

    always@(posedge clk)
     begin
         q<=d;
     end

endmodule
```

Submit | Submit (new window)

Upload a source file...

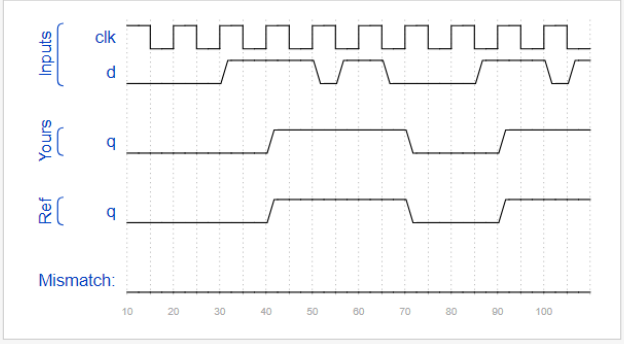
Status: Success!

You have solved 69 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).

Positive-edge triggered DFF



2. Create 8 D flip-flops. All DFFs should be triggered by the positive edge of `clk`.

Write your solution here

[Load a previous submission] Load

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

    always@(posedge clk)
        begin
            q<=d;
        end

endmodule

```

Submit | Submit (new window)

Upload a source file...

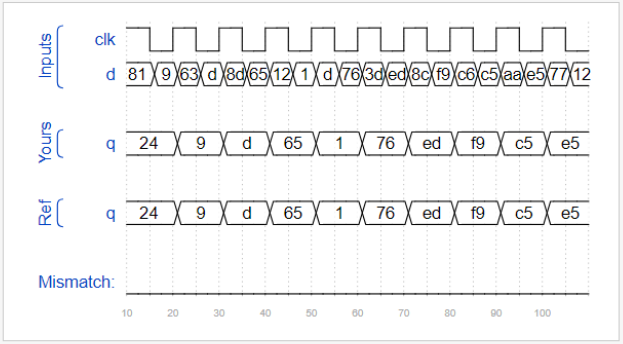
Status: Success!

You have solved 70 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).

Positive-edge triggered DFF



3. Create 8 D flip-flops with active high synchronous reset. All DFFs should be triggered by the positive edge of `clk`.

Write your solution here

[Load a previous submission] Load

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

    always@(posedge clk)
        begin
            q<=d;
        end

endmodule
```

Submit | Submit (new window)

Upload a source file...

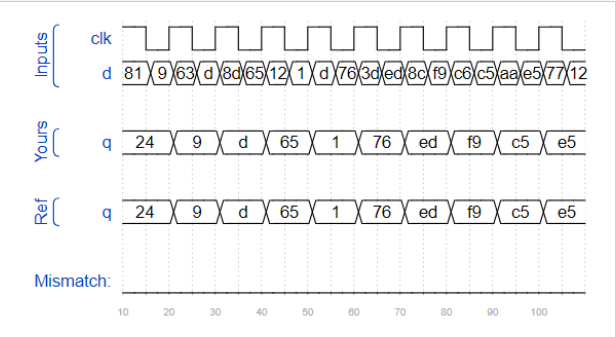
Status: Success!

You have solved 70 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).

Positive-edge triggered DFF



4. Create 8 D flip-flops with active high synchronous reset. The flip-flops must be reset to 0x34 rather than zero. All DFFs should be triggered by the **negative** edge of `clk`.

Write your solution here

[Load a previous submission] Load

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

    always@(negedge clk)
        begin
            if(reset)
                q<= 8'h34;

            else
                q<=d;

        end

endmodule

```

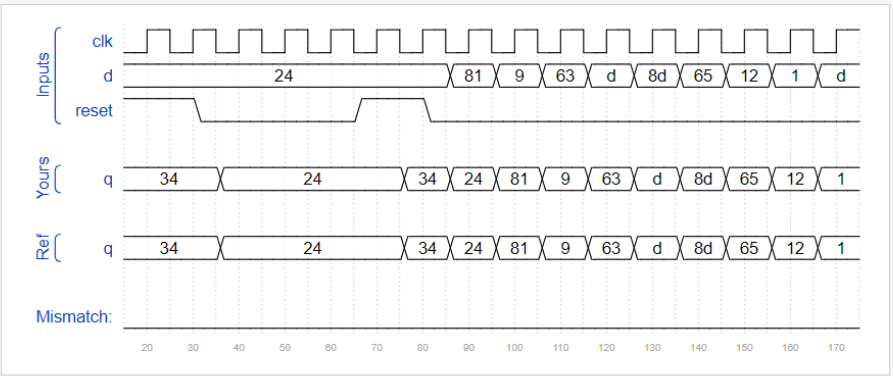
Status: Success!

You have solved 72 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-high reset



5. Create 8 D flip-flops with active high asynchronous reset. All DFFs should be triggered by the positive edge of `clk`.

Write your solution here

[Load a previous submission] Load

```
module top_module (
    input clk,
    input areset,   // active high asynchronous reset
    input [7:0] d,
    output [7:0] q
);

    always@(posedge clk or posedge areset)
        begin
            if(areset)
                q<=0;

            else
                q<=d;

        end

endmodule

```

Status: Success!

You have solved 73 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).

Asynchronous active-high reset