VLSI Design Flow

Mid Semester Exam (2nd March 2024)

Time allowed: 1 hour *Maximum Marks: 30*

Note:

I. ***All questions are compulsory.***
II. ***Marks are indicated in bold at the end of each question.***
III. ***Be crisp and precise in your answer so that you can complete the answers in 1 hour.***
IV. ***Cheating or use of unfair means will be dealt with as per institute policy.***

1.

a. What are inputs and outputs of Behaviour Synthesis or High-level Synthesis from the perspective of VLSI Design Flow? **[2 Marks]**

b. What do you understand by the Engineering Change Order (ECO) fixes/changes in VLSI Design Flow? **[2 Marks]**

c. What is the effect of increasing the area of a die on the yield of manufacturing that die? Give reason in support of your answer. **[0.5+1.5 Marks]**

d. Which of the two types of RTL simulator (cycle-based or event-based) can report within-cycle glitches for combinational circuit elements/logic. Give reason in support of your answer. **[0.5+1.5 Marks]**

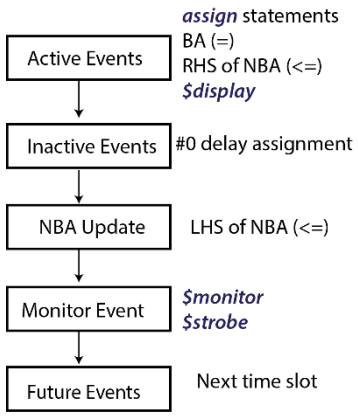
e. What is the typical effect of reducing the slew at the data pin of a D flip-flip in a technology library on: A) setup-time of that flip-flop B) clock-to-Q delay of that flop-flip. **[1+1 Marks]**

2. Consider the following piece of Verilog code. What will be the output when simulation is performed? **Give a very brief explanation for your answer.**

```
module top();
 reg a, b;

 initial begin
            a = 1'b1;
            a <= 1'b0;
            b = a;
            b <= a;

            $display("Display a=%b b=%b", a, b);
            $monitor("Monitor a=%b b=%b", a, b);
 end
endmodule
```



The stratified Verilog event queue is shown for your convenience. **[4 Marks]**

3. Draw the schematic of the typical circuit that will be generated by an RTL synthesis tool for the following Verilog codes (No need of any explanation). Label all the design entities (including pins and ports) appropriately (marks will be deducted for wrong/missing labeling). **[3+3 Marks]**

a.

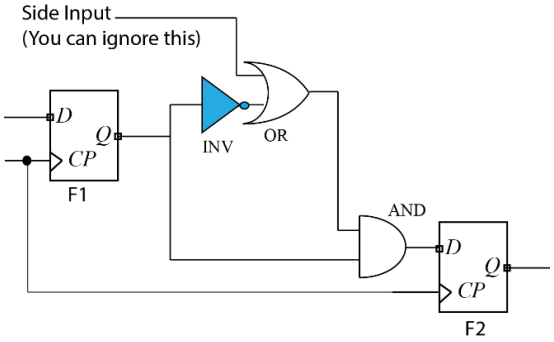
```
module top(clk, in1, out1);
        input clk;
        input [1:0]in1;
        output [1:0]out1;
        reg [1:0]cur_state;
        reg [1:0]next_state;
        always @(posedge clk) begin
                cur_state <= next_state;
        end
        always @* begin
           case (in1)
              2'b00 : next_state = 2'b00;
              2'b01 : next_state = 2'b01;
              2'b10 : next_state = 2'b10;
              2'b11 : next_state = 2'b11;
           endcase
        end
        always @* begin
                out1 <= cur_state;
        end
endmodule
```

b.

```
module top(clk, in, out);
        input clk;
        input in;
        output out;
        reg [3:0]q;
        reg i;
        always @(posedge clk) begin
                for (i=0; i<4; i=i+1) begin
                    if (i == 0) begin
                        q[i] <= in;
                    end else begin
                        q[i] <= q[i-1];
                    end
                end
        end
        always @* begin
                out = q[3];
        end

endmodule
```

4. Consider the portion of a synchronous circuit shown below.



Assume that the clock period is 1000 ps. Assume that the arrival time of the clock signal at the clock pin of all the flip-flops is 0 ps. For data not given for the problem (such as wire delay), assume that ideal conditions exist.

Assume that:

For the flip-flop F1: setup time = 22 ps, hold time = 12 ps, clock to Q delay = 17 ps

For the flip-flop F2: setup time = 37 ps, hold time = 15 ps, clock to Q delay = 34 ps

For the AND gate: delay = 23 ps

For the OR gate: delay = 44 ps

For the inverter: delay = 14 ps.

Consider the timing paths between F1 and F2. Compute the a) setup slack at F2 and b) hold slack at F2. [**2+2 Marks]**

5. Consider the following two Verilog netlists:

```
module top(in1, in2, in3, in4, clk, out1);
        input in1, in2, in3, in4, clk;
        output out1;
        wire y1;
        wire y2;
        wire z1;
        wire q1, q2;
        NAND2 N1( .A1(in1), .A2(in2), .ZN(y1));
        BUF B1 ( .I(in3), .Z(y2));
        DFF F1 (.CLK(clk), .D(y1), .Q(q1));
        DFF F2 (.CLK(clk), .D(y2), .Q(q2));

        NAND2 N2( .A1(q1), .A2(q2), .ZN(z1));
        NAND2 N3( .A1(z1), .A2(in4), .ZN(out1));
endmodule
```

```
module top(in1, in2, in3, in4, clk, out1);
        input in1, in2, in3, in4, clk;
        output out1;
        wire y1;
        wire y2;
        wire z1;
        wire q1, q2;
        NAND2 N1( .A1(in1), .A2(in2), .ZN(y1));
        INV I1 ( .I(in3), .ZN(y2));
        DFF F1 (.CLK(clk), .D(y1), .Q(q1));
        DFF F2 (.CLK(clk), .D(y2), .Q(q2));

        NAND2 N2( .A1(q1), .A2(q2), .ZN(z1));
        NAND2 N3( .A1(z1), .A2(in4), .ZN(out1));
endmodule
```

The cells DFF, NAND2, BUF, and INV are D flip-flop, NAND gate, buffer and inverter, respectively. These cells have typical pin names found in libraries. These two netlists are subjected to combinational equivalence checking (CEC). The ports and registers are matched by names.

a) Draw the schematic of both the netlists. Label the ports and instances (no other label needed). **[2 Marks]**

b) Draw the schematic of all the miters separately that the CEC tool will internally create. Label the ports and instances (no other label needed). **[3 Marks]**

c) Which of the miters drawn above will show inequivalence? Give the input pattern that will cause that miter to indicate inequivalence. **[1 Marks]**