COMP311-5: Logic Circuit Design

Spring 2019, Prof. Taigon Song

Quiz #1: March 26, 9:00 – 10:00am [Total: 105 points]

1. Design an inverter that has the module name of "inverter".
You must have an "assign" statement inside your design. [10 points]

2. Describe the period of clk1 and clk2 in nanoseconds (ns). [10 points each, 20 total]

```
`timescale 1ns / 1ns

module XORgate_tb();
  wire out1234567890;
  reg clock1, clock2;

  always begin
    #2 clock1 = ~clock1;
  end

  always begin
       #4 clock2 = ~clock2;
  end

  initial begin
       clock1 = 0;
       clock2 = 0;
  end

  XORgate tgxor1(clock1, clock2, out1234567890);

endmodule
```

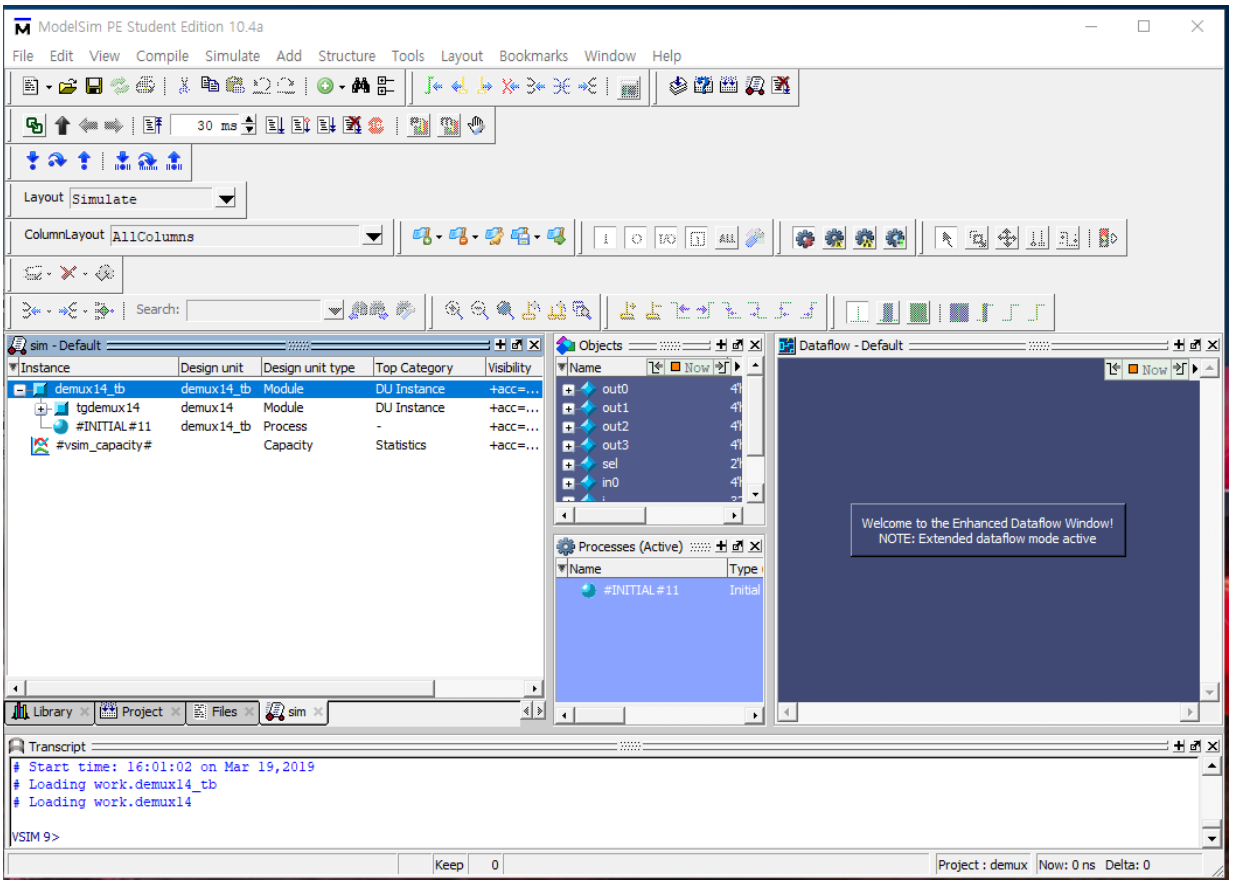
3. Convert '13' (decimal) to binary, then express this in Verilog. [20 points total]

(1) how many bits do you need? [10 points]

(2) use the Verilog expression to represent 13. [10 points]

4. Please describe the next step to be performed based on the Modelsim tutorial. [10 points]

(Hint, I finished compile, then ran "simulate" in Library→work→demux14_tb)



5. Please design a 2:1 mux using the following design methodologies.

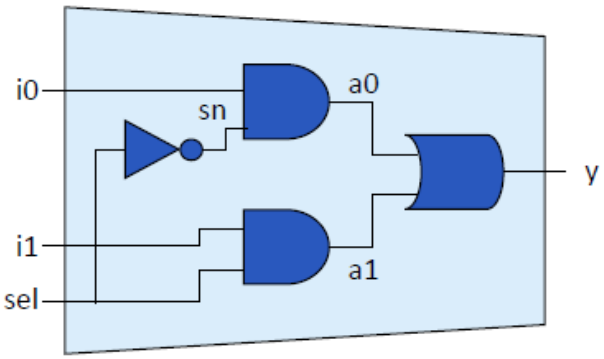
Below will be a common syntax this problem. If you need additional statements to detail your methodology, you are welcome to describe them.

[3 points each]

```
module mux21(output y, input i0, i1, s);

(fill in here)

endmodule
```



(1) Using primitive gates

(2) Using primitive gates, but not using INV

(3) Using only one AND keyword

(4) using the "assign" statement, no ternary operator
(hint, Y = I0◦S′ + I1◦S)

(5) using the "assign" statement, with ternary operator

(6) Using "if/else" statement

(Hint: you may need something more for (6) and (7)).

(7) Using "case" statement

5. Find the inappropriate statements in the following designs. You are to design an optimized 1:4 demultiplexer.

[12 errors in total, 2 points each]

```
module demux14(in0, sel, out0, out1,
out2, out3)
  input [3:0] in0;
  input [2:0] sel;
  output reg [3:0] out0, out1, out2, out3;
  wire select;

  always @(!) begin
    case(sel)
        2'b11 :  begin
                        out0 <= in0;
                out1 <= 4'b0000;
                out2 <= 4'b0000;
                out3 <= 4'b0000;

                        end
    2'b01 :  begin
                        out0 <= 4'b0000;
                out1 <= in0;
                out2 <= 4'b0000;
                out3 <= 4'b0000;
                       end
        2'b11 : begin
                        out0 <= 4'b0000;
                out1 <= 4'b0000;
                out2 <= in0;
                out3 <= 4'b0000;
                       end
        default : begin
                    out0 <= 4'b0000;
                out1 <= 4'b0000;
                out2 <= 4'b0000;
                out3 <= in0;
                       end
    endcase

endmodule
```

```
`timescale 1ns / 10ns
module demux14_tb(inout leon);
  wire [3:0] out0, out1, out2, out3, out4;
  reg [1:0] sel;
  reg [3:0] in0;
  wire i;

  demux14 tgdemux14(.in0(in10), .sel(sel),
.out0(out0), .out1(out1), .out2(out2),
.out3(out3));

  initial begin
    in0 <= $random;
        sel <= 0;

    for(i=1; i <4; i++)
          #10000 sel <= i;

  end

endmodule
```