COMP311-5: Logic Circuit Design

Spring 2019, Prof. Taigon Song

Mid-term exam: April 23, 5:00 – 6:20pm [Total: 125 points]

Guidelines:

1. Read the questions carefully and pay attention to the special instructions.
2. Show all your mark to receive the full credit.
3. State any assumptions you make on your solution.
4. Total number of pages in this exam: 6 (including this cover page.)
5. Write your name at ALL pages.

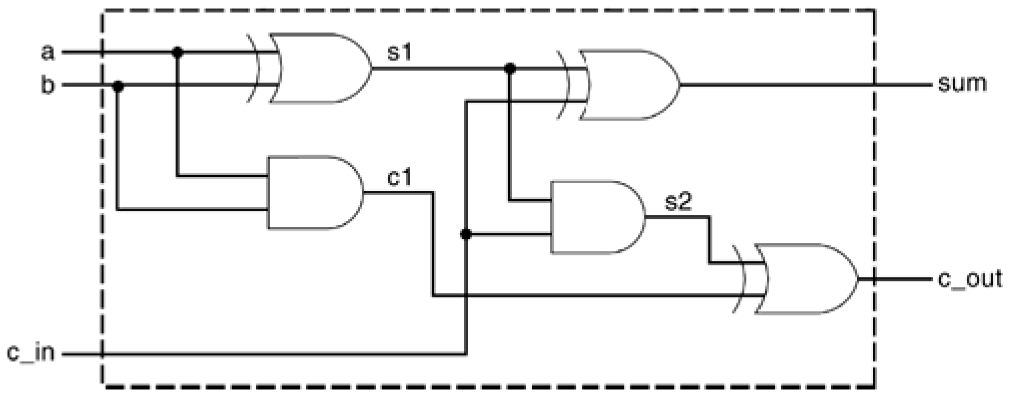
#ID / Name:____

Prob. 1 (15pts)	
Prob. 2 (15pts)	
Prob. 3 (20pts)	
Prob. 4 (20pts)	
Prob. 5 (20pts)	
Prob. 6 (25pts)	
Bonus (10pts)	
Total (125pts)	

Student ID#: Name:

1. Design a 1-bit full adder using primitive gates. The schematic is as the below (15 points)

Figure 5-6. 1-bit Full Adder



```
module onebit_adder(input a, b, cin, output sum, cout);

endmodule
```

Student ID#: Name:

2. I want to exit my loop when "Encountered a TRUE bit at element number %d" is printed. (Total: 15 points)

```
module exitingloop();
  parameter TRUE=1'b1;
  parameter FALSE=1'b0;

  reg [15:0] flag;
  integer i;
  reg continue1;

  initial begin
    flag = 16'b 0010_0000_0000_0000;
        i=0;
        continue1 = TRUE;

        while((i < 16) && continue1  )
        begin
          if (flag[i]) begin
            $display("Encountered a TRUE bit at element number %d", i);
            continue1 = FALSE;
        end
      i = i+1;
    end
    $display("i at the end is %d", i);
  end

endmodule
```

(2.1) When running simulation in Modelsim, describe the "i" values in the command prompt. (10 points)

```
VSIM 39> run
# Encountered a TRUE bit at element number ________ (write your answer here)
# i at the end is                         ________ (write your answer here)
```

(2.2) fix the source code to make the same thing happen by exiting the initial block. Remove statements related to the "continue1" variable. You may recycle the source code above. (5 points)

3. Find the correct answer. (Total: 20 points, each 4 points)

(3.1) A function can enable another function but not another task.	O	X
(3.2) Functions may execute in non-zero simulation time	O	X
(3.3) Functions may contain delay, event, or timing control statements	O	X
(3.4) Tasks must have at least one input argument. They can have more than one input.	O	X
(3.5) Tasks always return a single value.	O	X

4. Judge the following source codes. Is it correct? Then mention "correct" and do nothing. If not, fix the source codes. (Total: 20 points)

(4.1) a function that multiplies two inputs and returns an output based on the module description.

```
module mul_8b(
  input [3:0] in1, in2,
  output [7:0] out1
    );

    assign out1 = mul4b (in1, in2);
    //assign out1 = in1 * in2;

    function  mul4b(input input1, input2);
      begin
        mul4b = input1 * input2;
      end

    endfunction

endmodule
```

(4.2) a task that checks the even parity bit on every rising clock edge.

```
`timescale 1ns / 1ns
module even_parity_tb();
  reg [7:0] in1;
  wire out1;
  reg clk, out2;

parameter something = 15;

  even_parity u1(in1, clk, out1);

  initial begin
    in1=0; clk = 0;
    #15 in1 = $random; //eparity;
    #20 in1 = $random; //eparity;

 end

  always
    #(10 + something) clk = ~clk;

  always
    eparity;

 task eparity;
  if(clk == 1)
    @(posedge clk) out2 <= #5 ~(^in1);
   endtask

endmodule
```

Student ID#: Name:

5. Design the following flip-flops. Your common syntax for inputs and outputs are as below: (Total 20 points)
inputs: d, clk, rst, preset, ld
output: q

(5.1) Positive-edge FF.	(5.2) Negative-edge FF w/ async reset (rst) at 0
(5.3) Positive-edge FF w/ sync preset at 1.	(5.4) Positive-edge FF w/ sync reset at 0, enable at 1

6. Based on Modelsim, describe out1, out2, out3, out4, and out5 at 12ns in binary digits (e.g., 110011). (Total: 25 points)

```
`timescale 1ns/1ns
module q5check();
  reg [5:0] in1, in2;
  reg [5:0] out1, out2, out3, out4, out5;
  initial begin
    //out1 <= 0;
    in1 = 5'b11010; in2 = 5'b11000;
    out4 = in1 + in2;
    out2 = in1 ^ in2;
    #1  in1 = 5'b11010; in2 = 4'bx1100;
        out5 = in1 + in2;
    #5  in1 = 5'b1101x;
    #4  out1 = &in1;
    #10 out3 = |in1;
  end
endmodule
```

Bonus. What would happen in modelsim when running the following source code? Describe in terms of the "#4.3" term. (Total 10 points)

(Bonus 1.1) Would it fail to compile? Or would it fail during runtime? Or if it runs successfully, when would "out1" change its value to in1 + in2? (5 points)

(Bonus 1.2) Fix the source code to correctly print 4.3ns delay in Modelsim. (5 points)

```
`timescale 1ns/1ns
module q1addcheck();
  reg [5:0] in1, in2;
  reg [5:0] out1;
  initial begin
    out1 = 0;
    in1 = 5'b11010; in2 = 5'b11000;
#4.3 out1 = in1 + in2;
  end
endmodule
```