❖ Problem Set 5: Block Design

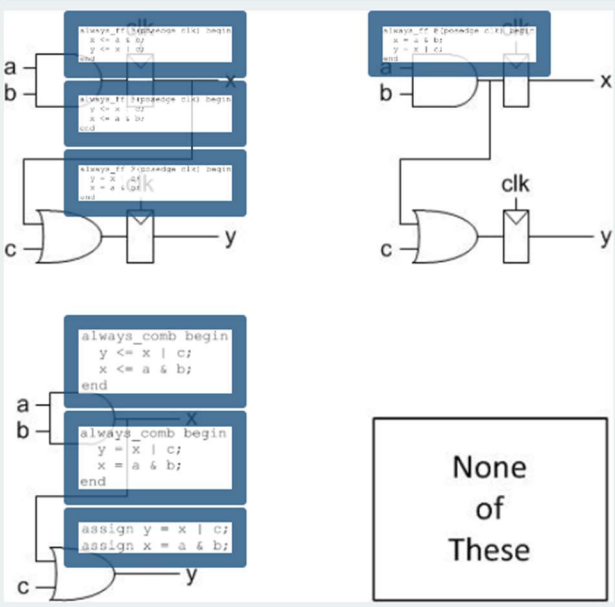
First problem is just jigsaw puzzle to try. Picture to show number of answer per category:

SystemVerilog Assignments

1/1 point (graded)

Keyboard Help

Drag each of the following snippets of SystemVerilog code to the circuit it implies. You may assume that all inputs, outputs, and internal signals have been declared as single-bit logic signals.



None of These

Reset

Consider the following code that shows up as the 4th hit on a Google search for "verilog counter"

https://riptutorial.com/verilog/example/8307/simple-counter

```
module counter(
    input clk,
    output reg[7:0] count
)
initial count = 0;
always @ (posedge clk) begin
    count <= count + 1'b1;
end
```

Counter Syntax Errors

1/1 point (graded)

Find the syntax error in this counter.

- ○ No space after the word reg
- ◉ No semicolon at the end of the module declaration
- ○ Illegal space after @
- ○ None of the above

✔

Submit Try again (1 attempt remaining) Show answer

Logic Bug

1/1 point (graded)

Find the logic bug in this counter.

- ○ 1'b1 is only one bit, so the counter will only count between 0 and 1 before returning to 0.
- ○ The code does not have always_ff, so there will be no flip-flops and the counter will not remember its value.
- ◉ The initial statement will cause the counter to start at 0 in simulation. However, there is no sequential circuit element that magically starts at 0 when first powered up, so a real counter chip will start at a random number. Hence, physical hardware will not match the behavior of the simulation.

✔

Submit Try again (1 attempt remaining) Show answer

Problem Set due May 6, 2023 10:37 PDT Completed

Sign-magnitude addition is harder than two's complement addition. Consider adding N-bit numbers Y = A + B. Let SY, SA, and SB be the sign bits and MY, MA, and MB be the N-1 bit magnitudes.

An algorithm for sign-magnitude addition (disregarding overflow) is:

```
if (SA == SB) begin  // if the signs are the same

    {Cout, MY} = MA + MB     // just add the magnitudes

    SY = SA          // and the sign of the result matches the sign of the inputs

end else begin       // if the signs are different

    MBN = ~MB

    {Cout, MR} = MA + MBN + 1 // Negate MB using 2's complement, and then add A + (-B)

    if (Cout) begin  // If there is a carry out, MA >= MB

        MY = MR      // Result has correct magnitude

        SY = SA      // And sign matches A

    end else begin

        MY = ~MR + 1 // Otherwise negate result with 2's complement

        SY = SB      // and sign matches B

    end

end
```

Overflow

1/1 point (graded)

When does Overflow (V) occur in sign-magnitude addition using the algorithm above?

- ○ Whenever there is a carry out
- ○ Whenever there is no carry out
- ◉ Whenever the signs of A and B are the same and there is a carry out
- ○ Whenever the signs of A and B are the same and there is no carry out
- ○ Whenever the signs of A and B differ and there is a carry out
- ○ Whenever the signs of A and B differ and there is no carry out
- ○ Whenever the signs of A and B are the same and the sign of Y is different

✔

Submit Try again (1 attempt remaining) Show answer

A SystemVerilog implementation and associated test vectors are here:

smadd.sv

smadd.tv

Component delays are given below. Remember that the assign statements at the beginning and end of small just imply wires and cost no area or delay. Note that the design described in the SystemVerilog does not use NANDs or NORs.

Cell	t_{pd} (ps)	t_{cd} (ps)	Relative Size
NOT	6	4	2
AND2	14	10	5
AND3	18	14	6
OR2	16	12	5
OR3	20	16	6
XOR2	24	18	9
NAND2	8	6	3
NAND3	12	10	4
NOR2	10	8	3
NOR3	14	12	4

Sketch out the circuit at the schematic level for your own analysis, and count the number of each type of cell used in the design, and identify the longest and shortest paths through the sign-magnitude adder.

NOT gate count

1/1 point (graded)

How many NOT gates are in this design?

9 ✔

9

Submit Try again (1 attempt remaining) Show answer

AND2 gate count

1/1 point (graded)

How many AND2 gates are in this design?

18 ✔

18

Submit Try again (1 attempt remaining) Show answer

OR2 gate count

1/1 point (graded)

How many OR2 gates are in this design?

5

5

Submit Try again (1 attempt remaining) Show answer

OR3 gate count

1/1 point (graded)

How many OR3 gates are in this design?

3

3

Submit Try again (1 attempt remaining) Show answer

XOR gate count

1/1 point (graded)

How many XOR2 gates are in this design?

12

12

Submit Try again (1 attempt remaining) Show answer

Circuit Cost

1/1 point (graded)

What is the total cost of the circuit, measured as the sum of all of the components weighted by their relative size?

259

259

Submit Try again (1 attempt remaining) Show answer

Critical Path

1/1 point (graded)

What is the critical path through the sign/magnitude adder?

- (●) NOT + 3AND2 + OR2 + 2OR3 + 4XOR
- ( ) 2NOT + AND2 + 2OR2 + XOR
- ( ) NOT + 4AND2 + OR2 + 3OR3 + 2XOR
- ( ) 4NOT + 7AND2 + 4OR2 + 3OR3 + 2XOR
- ( ) 5NOT + 18AND2 + 5OR2 + 3OR3 + 12XOR

✔

Submit Try again (1 attempt remaining) Show answer

Critical path delay

1/1 point (graded)

How long is the critical path? Express your answer in ps.

200 ✔

200

Submit Try again (1 attempt remaining) Show answer

Short Path

1/1 point (graded)

What is the shortest path through the sign/magnitude adder?

- (●) AND2 + OR2
- ( ) AND2 + OR2 + 2XOR
- ( ) NOT + AND2 + 2OR2 + XOR
- ( ) 2NOT + AND2 + OR2 + 4XOR
- ( ) NOT + 3AND2 + OR2 + 2OR3 + 4XOR

✔

Submit Try again (1 attempt remaining) Show answer

Short path delay

1/1 point (graded)

How long is the short path? Express your answer in ps.

22

22

Submit Try again (1 attempt remaining) Show answer

Optimization

1/1 point (graded)

Suppose you were building the system with CMOS gates and could redesign circuits using NAND and NOR instead of AND and OR. Which module would you redesign to obtain the greatest speedup?

- not_3
- mux_2
- fulladder
- incr

Submit Try again (1 attempt remaining) Show answer