

EECS 151/251A Homework 1

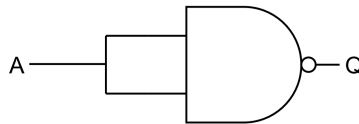
Due Friday, Sept 10th, 2021

Submit your answers directly on the assignment on Gradescope.

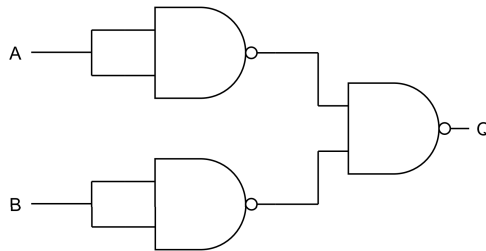
Problem 1: Logic Warm-up

Identify the Boolean logic function that is equivalent to the operation of each of the following circuits. Please make sure to write your answer clearly in the box on the assignment. The answer in each box will be a standard Boolean operator that is operating on the inputs A and B. For example, if the circuit output, Q, were to be the $\text{NAND}(A, B)$, you should write NAND in the box.

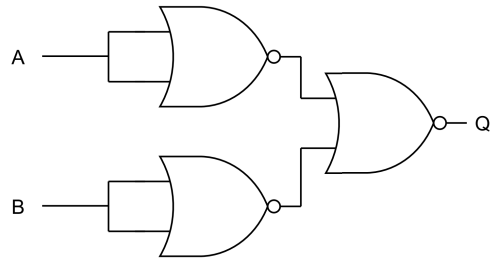
(a)



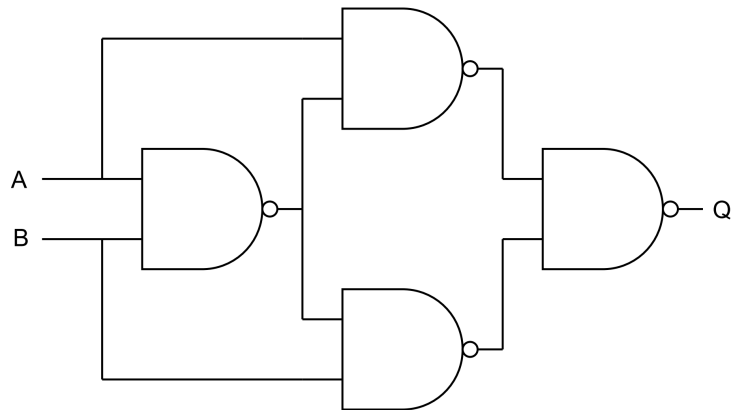
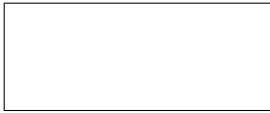
(b)



(c)



(d)



Problem 2: Boolean Algebra

- (a) Simplify the following expression: $((\overline{A + B}) + \overline{A})B + A$

Simplified Expression:

- (b) Simplify the following expression: $(A + BC)\overline{(A + B)}$

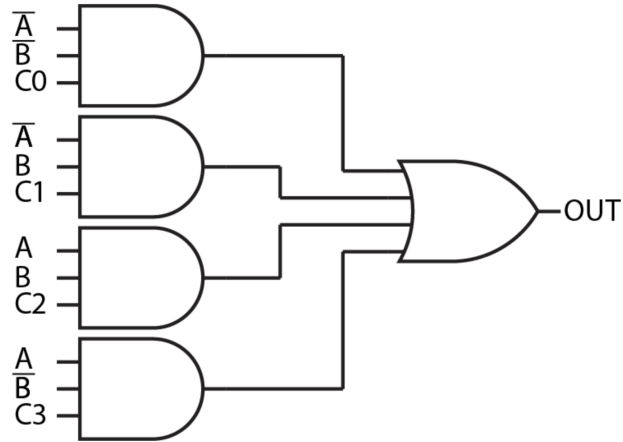
Simplified Expression:

- (c) Simplify the following expression: $(\overline{A}B + A)((\overline{A})(\overline{B}) + C)\overline{(\overline{B} + AC)}$

Simplified Expression:

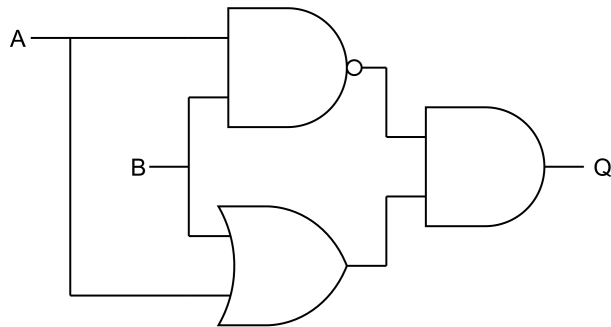
Problem 3: More Boolean Logic

- (a) Consider the given circuit. All inputs (A , B , $C0$, $C1$, $C2$, $C3$) must be either 0 or 1. What must $C0$, $C1$, $C2$, and $C3$ be such that the circuit computes the function $\text{AND}(A, B)$?



$C0$: $C1$: $C2$: $C3$:

- (b) For the following logic circuit, write out the truth table. What is the equivalent Boolean operation of this circuit?



Boolean operator:

- (c) Consider the following truth table. Write out the sum-of-products expression and simplify it, recording your simplified expression in the box at the bottom of the page. Write the name of the useful digital block that the truth table is equivalent to.

A	B	S	Out
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

Simplified Expression:

Name of block:

Problem 4. Elementary Verilog

- (a) For each of the following Verilog modules, write *yes* in the box below it if it will compile, or *no* in the box if it will not compile.

```
module module_1(  
    input a, b,  
    output reg out  
);  
    assign out = a & b;  
endmodule
```

```
module module_2(  
    input a, b,  
    output out  
);  
    always @(*) begin  
        assign out = a & b;  
    end  
endmodule
```

```
module module_3(  
    input a, b,  
    output reg out  
);  
    always @(*) begin  
        out = a | b;  
    end  
endmodule
```

- (b) Fill in the following blanks in the following snippet of Verilog on the dotted lines (eg. _____) so that *out* matches the output of a 2-to-1 mux with inputs *a* and *b*, and a select bit, *sel*, equal to AND(*c*, *d*). Each blank should have one word or symbol.

```
module mux2to1(  
    input a, b, c, d,  
  
    output _____ out  
);  
  
    _____ sel;  
  
    _____ sel = c _____ d;  
  
    always @(*) begin  
        if(sel)  
            out _____ a;  
        else  
            out _____ b;  
    end  
endmodule
```