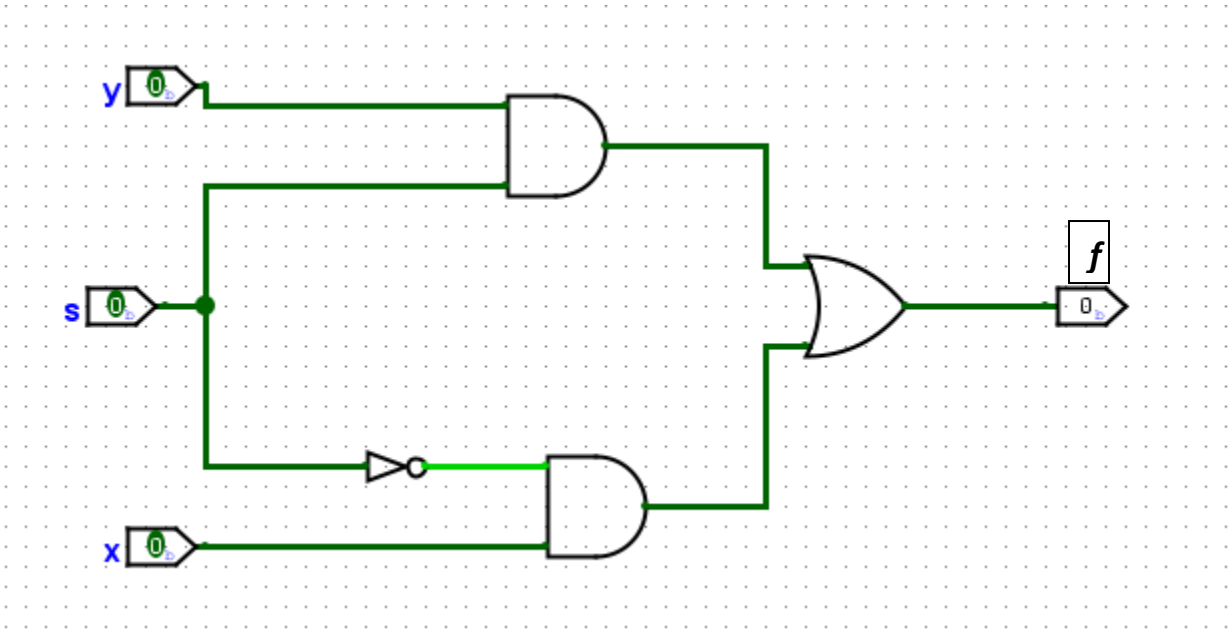
Tony Hong
1005109482
Sep. 20, 2020

CSC258 Prelab (Lab 1)

Part 1: 2-to-1 Multiplexer

1. Using only the AND, OR and NOT gates, below is a circuit with input *y*, *s* and *x* and output *f=xs'+ys.*

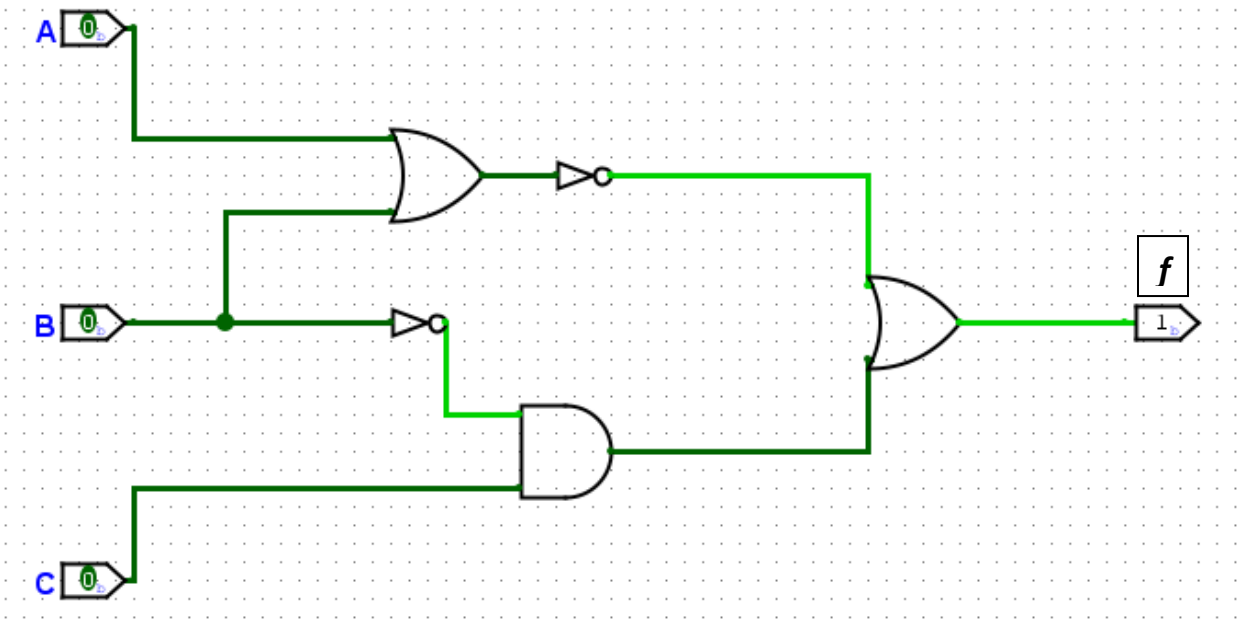


2. Below is the truth table of *xs'+ys*.

X	Y	S	XS'+YS
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

Part 2:

1. Below is a circuit implementation of *f = (a + b)' + cb'.*



2. Below is the truth table of *(a + b)' + cb'.*

A	*B*	*C*	*(A+B)' + CB'*
0	*0*	*0*	*1*
0	*0*	*1*	*1*
0	*1*	*0*	*0*
0	*1*	*1*	*0*
1	*0*	*0*	*0*
1	*0*	*1*	*1*
1	*1*	*0*	*0*
1	*1*	*1*	*0*

3. There is a cheaper implementation for the design. The Boolean expression above required 5 gates as shown in the circuit diagram. Using the following operations, we can show that only 4 gates are required.

(a + b)' + cb'

= a'b'+cb' (DeMorgan's Law)

= b'(a'+c) (Distributive Law)

We see that *b'(a'+c)* is an equivalent Boolean expression compared to *(a + b)' + cb'* and only requires 4 gates to implement as a circuit.

Below is the cheaper implementation of the circuit in step 1.