Lab 1. Building Circuits using Logisim Evolution.

Student Name: Zewen Ma

Student #: 1005968375.

Part 1: Given the following Boolean function, which is a logic expression for a 2-to-1 multiplexer:

$$f = xs' + ys$$

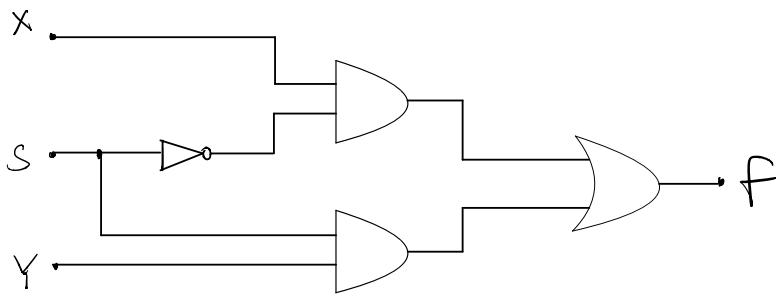
Performing:

1. Draw the gate diagram using the AND. OR. and NOT gates.

- We express the function using logic words:

  f = (X AND (NOT S)) OR (Y AND S).

Therefore, according to the expression above. We can draw:



2. Truth Table.

x	s	y	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

Part 2: Given the following Boolean expression:

$$f=(a+b')'+cb'.$$

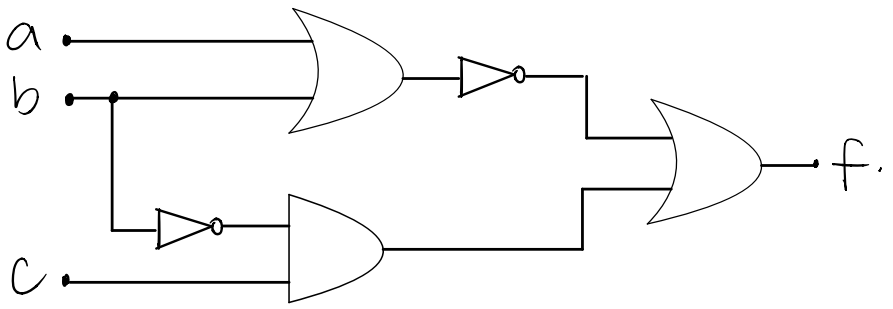
Performing:

1. Draw the gate diagram using the AND, OR and NOT gates.

- We express the function using logic words:

  f = (NOT (a OR B)) OR (c AND (NOT b))

Therefore, according to the expression above, we can draw:



2. Truth Table:

a	b	c	f
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. Yes, there is a cheaper implementation for the design that use fewer gates:

$f = (a+b)' + cb'$

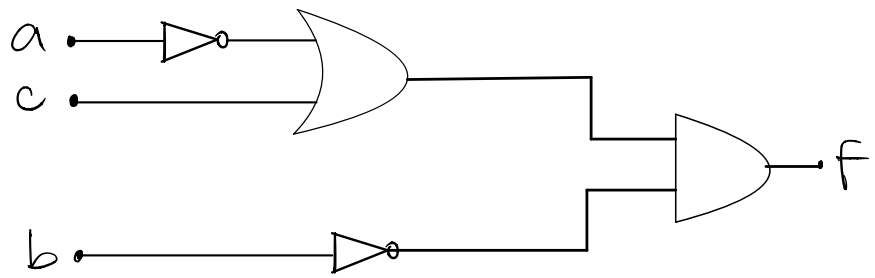
$= (a'b') + cb'$ (Since we know NOT (a OR b) has the same output as (NOT a) AND (NOT b))

$= b'(a'+c)$

Therefore $f = b'(a'+c)$ i.e

$f =$ (NOT b) AND ((NOT a) OR c)

- According to the expression above, we can draw:

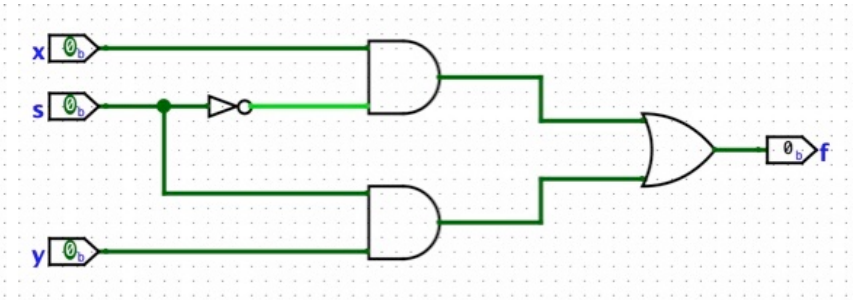


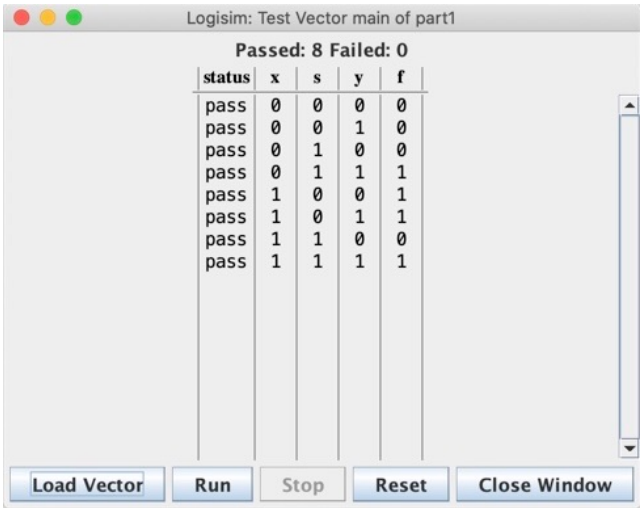
- Truth table:

a	b	c	f
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

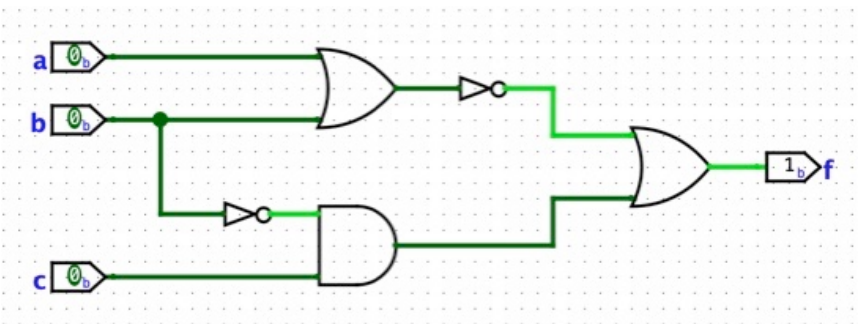
Part 3. Circuits from Part 1 and Part 2 implemented in Logism and their test result.

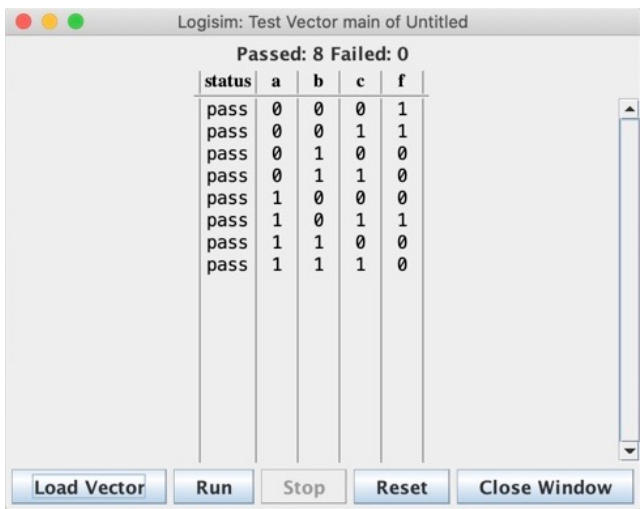
①. Circuit from Part 1 and its test result.





②. Circuit from Part 2 and its test result.





③. Cheap Implementation of Circuit in Part 2. and its test result.