Lab 07: Conditional Codes

The purpose of this lab is to be able to save a value that we can access later. In this lab, we choose an appropriate flip flop and build a logisim subcircuit that saves a value until another operation sets it once more. This subcircuit will be used later on in the semester for the final project.

There are two incoming signals, A and B. When A is high, the value of the flip flop should be set to the value of B. When A is low, the value does not change. For this particular problem, I choose to use the JK flip flop because there are two input values A and B and thus it will be better suited. Thus, I make the following truth table with A, B, and Q as my input and Q_1, J, and K as my outputs:

A	**B**	**Q**	**Q_1**	**J**	**K**
0	0	0	0	0	X
0	0	1	1	X	0
0	1	0	0	0	X
0	1	1	1	X	0
1	0	0	0	0	X
1	0	1	0	X	1
1	1	0	1	1	X
1	1	1	1	X	0

The SOP formulas that I come up with are:

$$J = ABQ'$$

$$K = AB'Q$$

Using my knowledge of Karnaugh maps and algebraic proofs, I reduce my SOP expressions:

	BQ			
J	00	01	11	10
A				
0	0	X	X	0
1	0	X	X	1

For J, I get the expression:

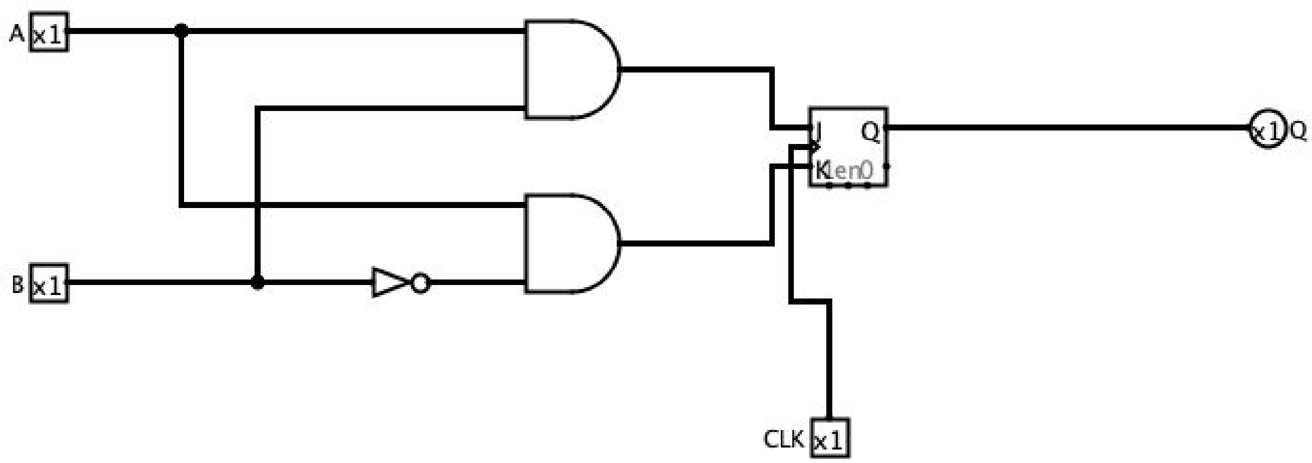
$$J = AB$$

K	BQ 00	01	11	10
A				
0	X	0	0	X
1	X	1	0	X

For K, I get the expression:

$$K = B'A$$

I sketched out the following combinational circuit based on the expressions I got:



This subcircuit as you can see is only for 1-bit. In logisim, I further replicate the subcircuit 8 times to get the 8-bit register. I split the 8-bit input B using a splitter and then input it to my 8 subcircuits. A is fed as a 1-bit input. I further use one clock for all my subcircuits and then the output Q is combined via a splitter. The main circuit in my .circ lab shows my 8-bit register.

Thus after this lab, we are able to better understand how JK flip flops work and when to use them. The lab also enhances our knowledge of truth tables and how to use Karnaugh maps to get simplified boolean expressions. We also learn how to use multiple subcircuits to make an 8-bit register while increasing our skills in Logisim.