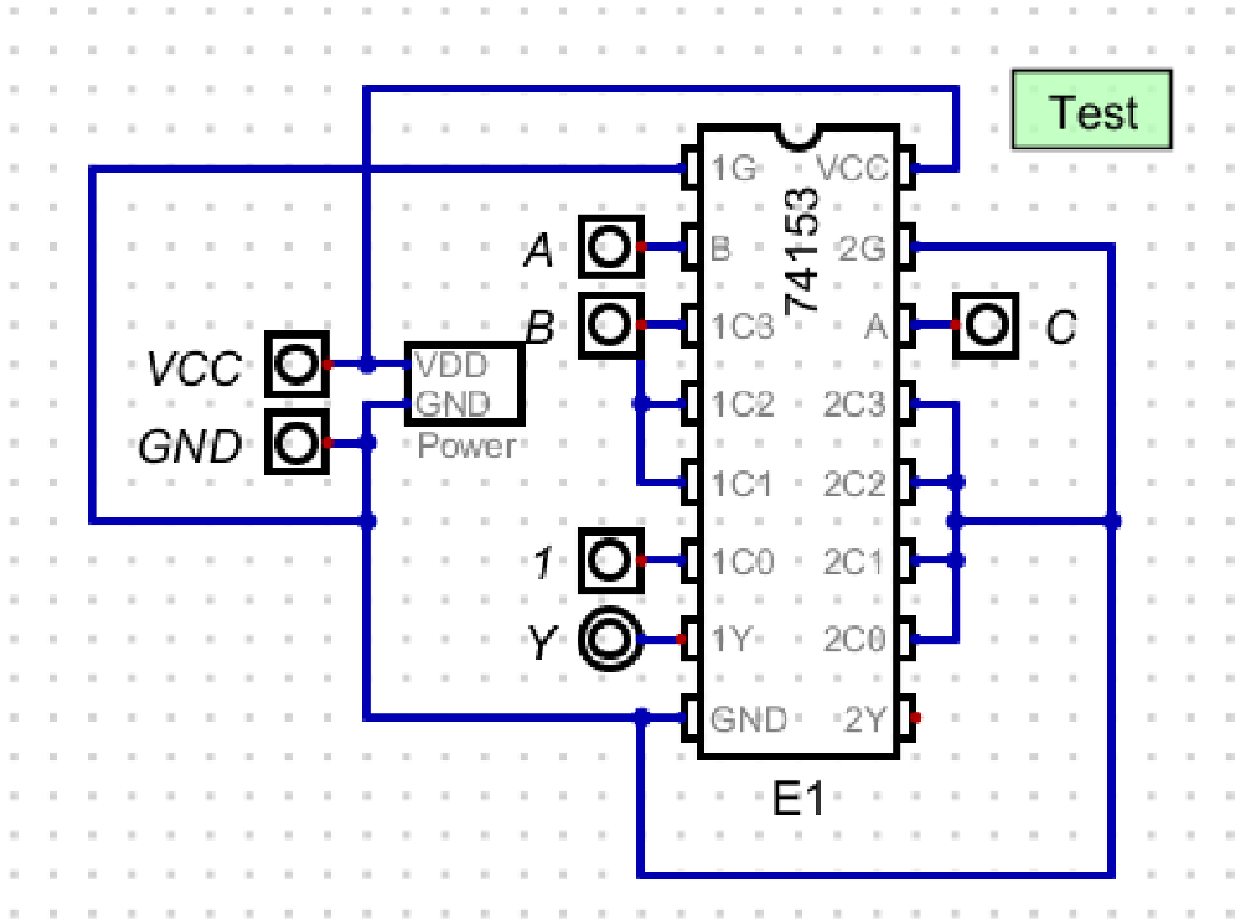
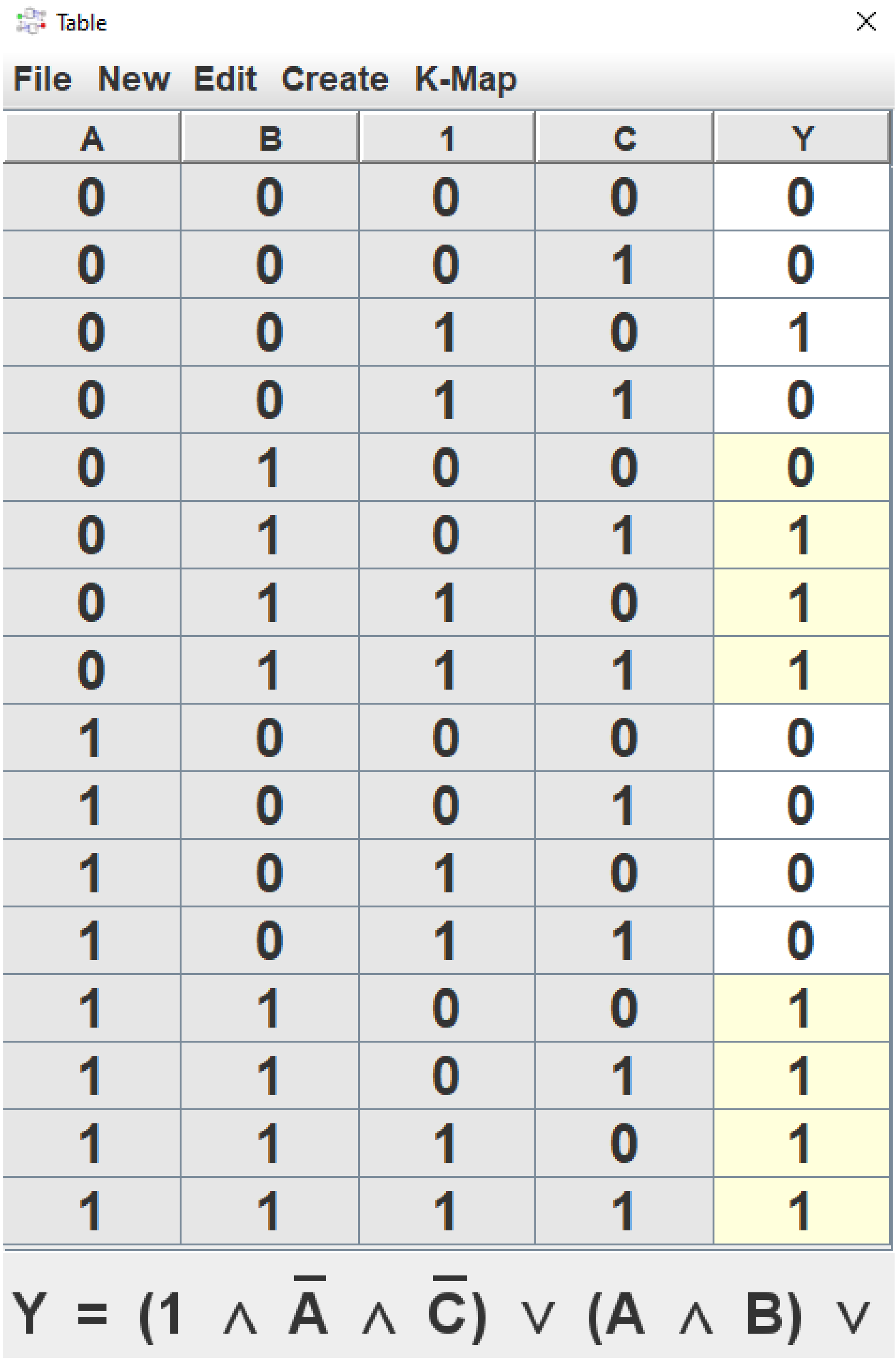
Zen Park
WXP190001
CS 4141.6U1
6/22/21

Postlab 2

1)

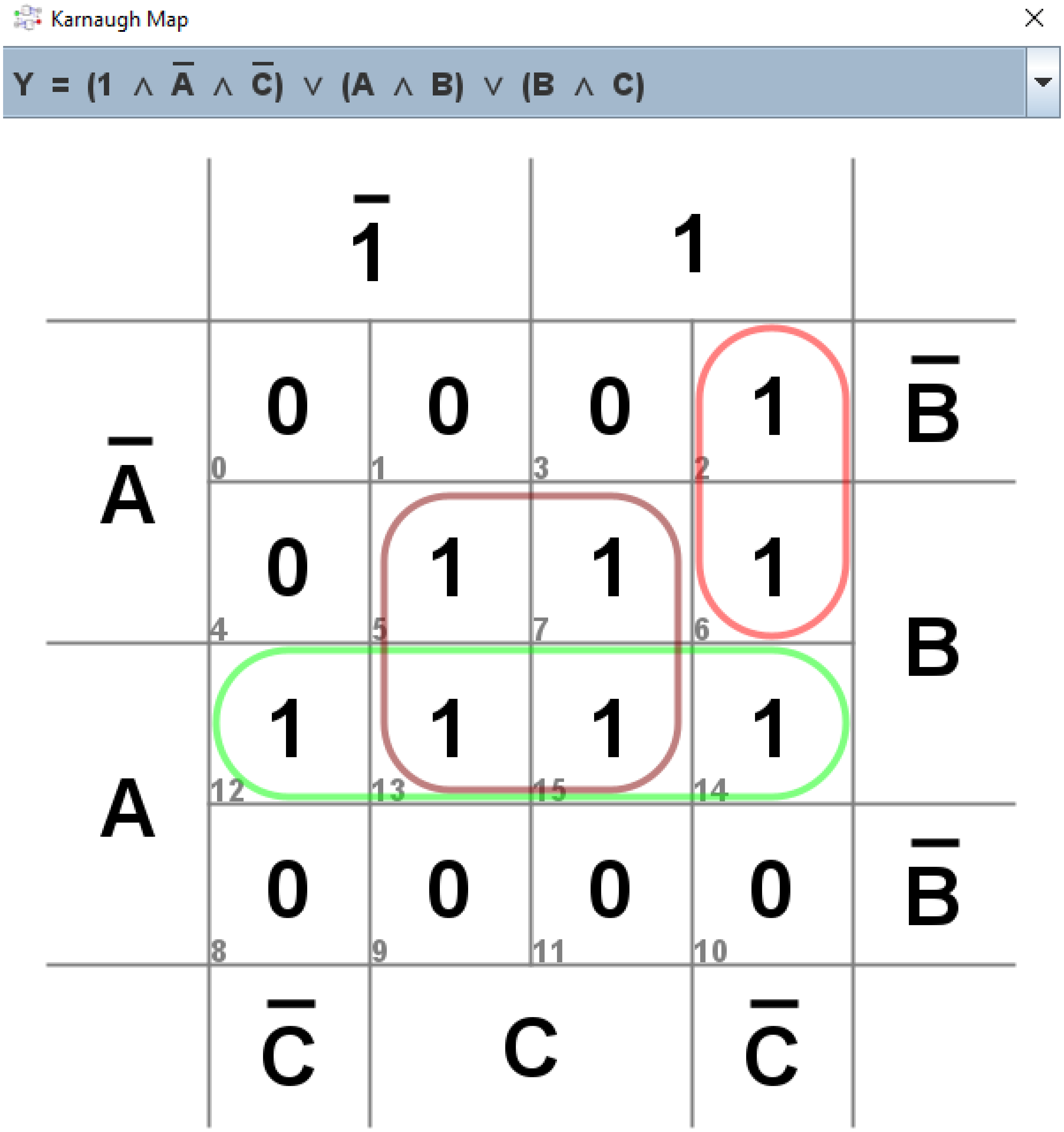


Analysis: Using the 74153 chip that uses a 4:1 mux, we get several inputs and the Y output, which represents a variety of variables. Through various connections such as A and B connecting to 1C2 and 1C1, we also see that the output is connected with 1Y as well. The right side of the chip has majority of its sections grounded.



A	B	1	C	Y
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	1
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

$$Y = (1 \wedge \overline{A} \wedge \overline{C}) \vee (A \wedge B) \vee$$

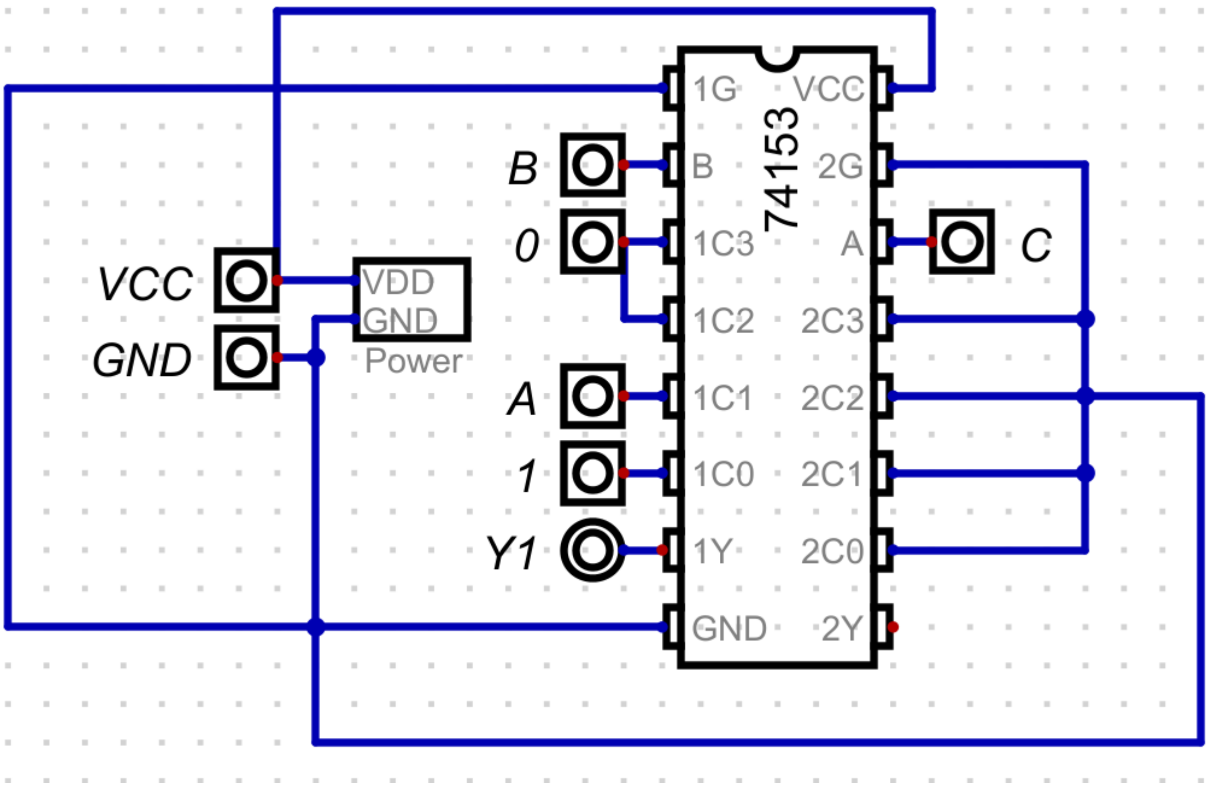


Karnaugh Map

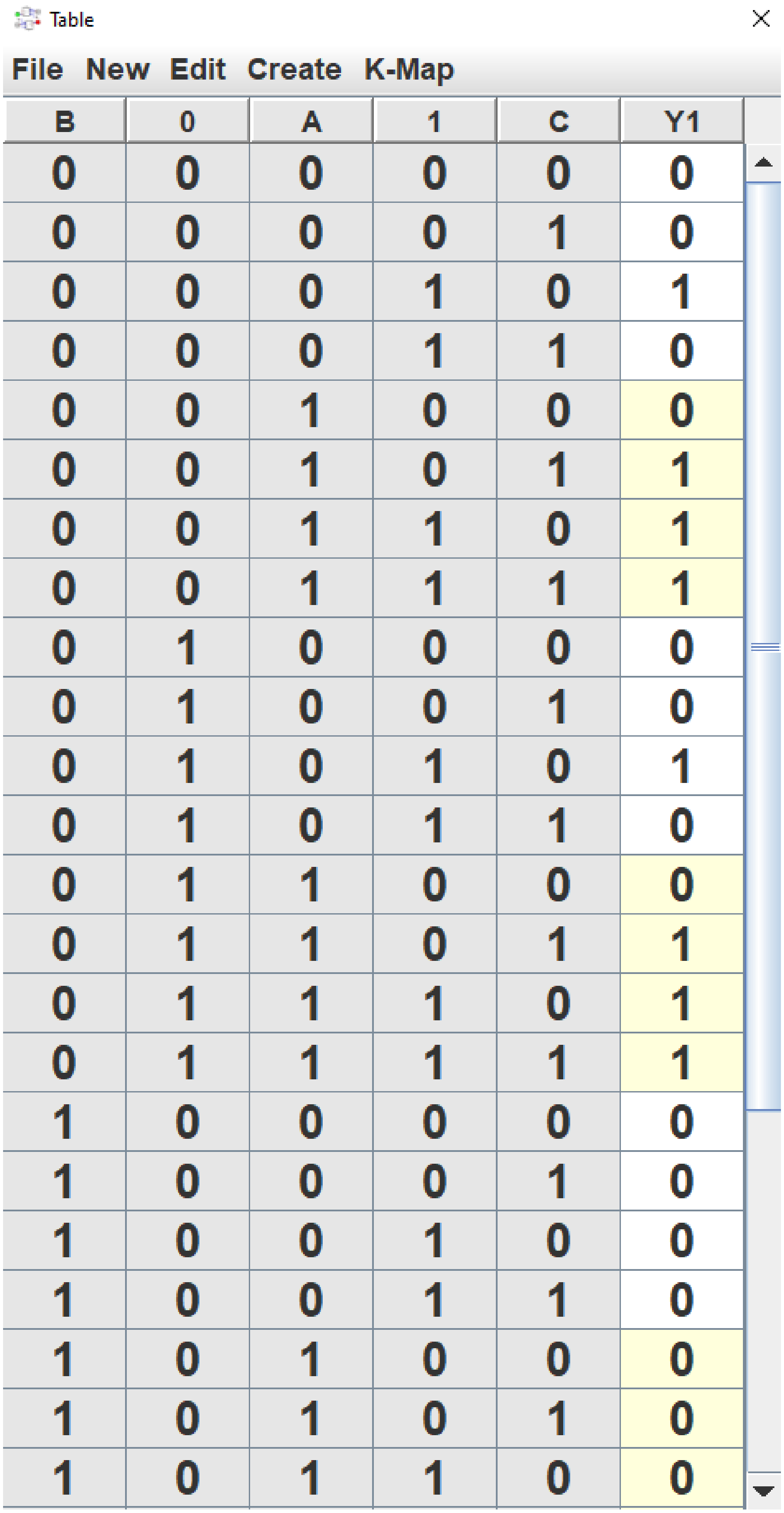
$Y = (1 \land \bar{A} \land \bar{C}) \lor (A \land B) \lor (B \land C)$

	$\bar{1}$		1		
$\bar{A}$	0 0	0 1	0 3	1 2	$\bar{B}$
	0 4	1 5	1 7	1 6	B
A	1 12	1 13	1 15	1 14	
	0 8	0 9	0 11	0 10	$\bar{B}$
	$\bar{C}$	C		$\bar{C}$	

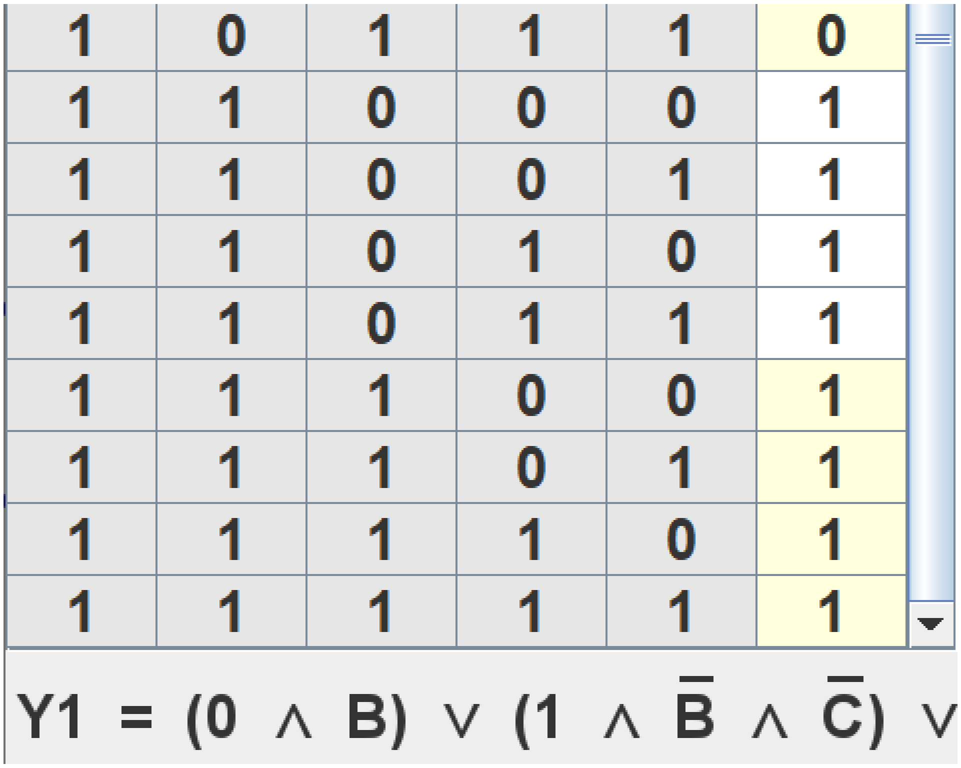
2)



Analysis: Here, we also use the 74153 chip, just as we did in the previous one. Although both formatted chips look very similar, do not let this fool you. Here, we have B and 0 which connects with 1C2. A and 1 have their own individual inputs. 1Y is an output, just as the first one. We ground nearly every right side of the chip as well.



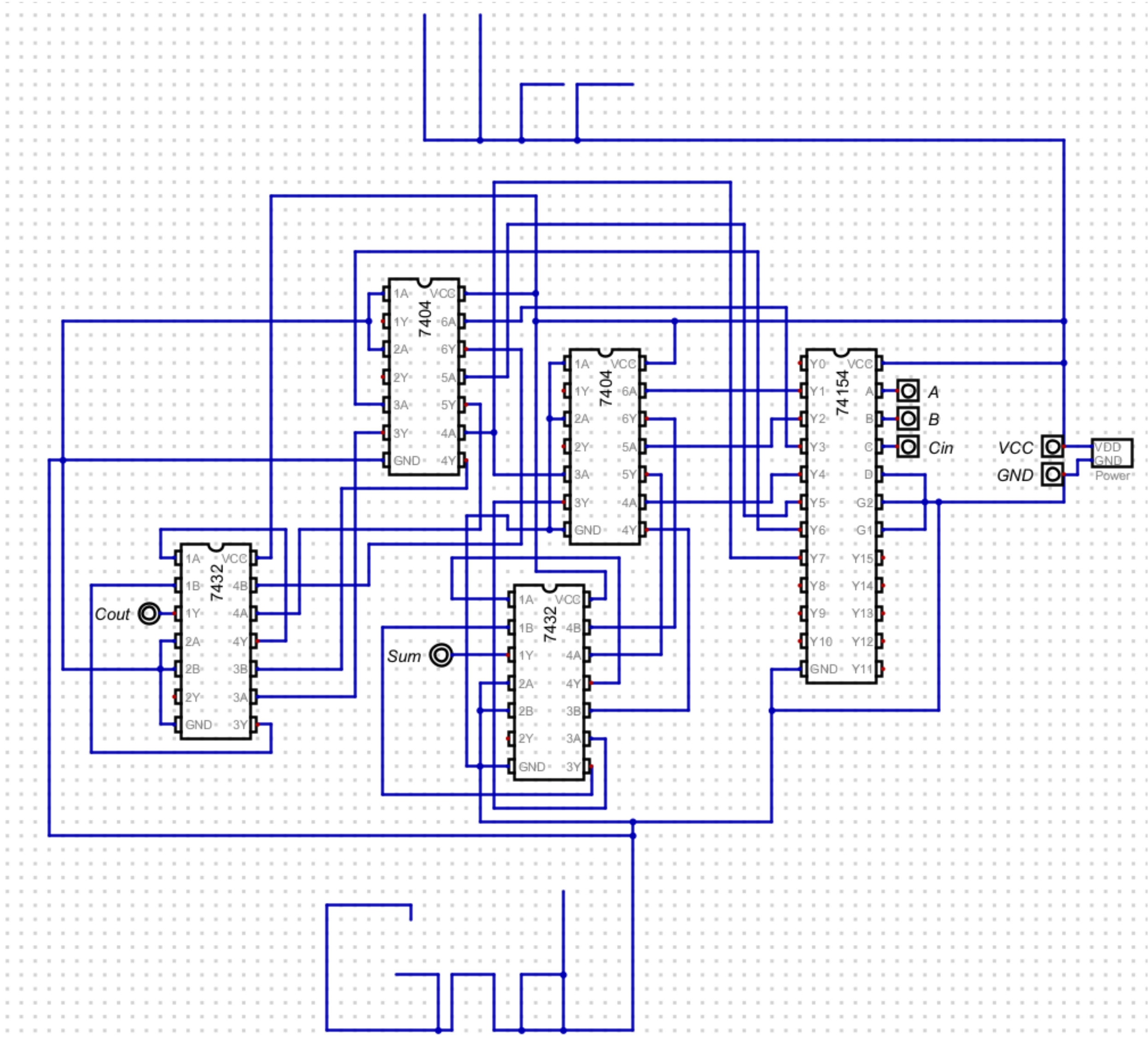
B	0	A	1	C	Y1
0	0	0	0	0	0
0	0	0	0	1	0
0	0	0	1	0	1
0	0	0	1	1	0
0	0	1	0	0	0
0	0	1	0	1	1
0	0	1	1	0	1
0	0	1	1	1	1
0	1	0	0	0	0
0	1	0	0	1	0
0	1	0	1	0	1
0	1	0	1	1	0
0	1	1	0	0	0
0	1	1	0	1	1
0	1	1	1	0	1
0	1	1	1	1	1
1	0	0	0	0	0
1	0	0	0	1	0
1	0	0	1	0	0
1	0	0	1	1	0
1	0	1	0	0	0
1	0	1	0	1	0
1	0	1	1	0	0



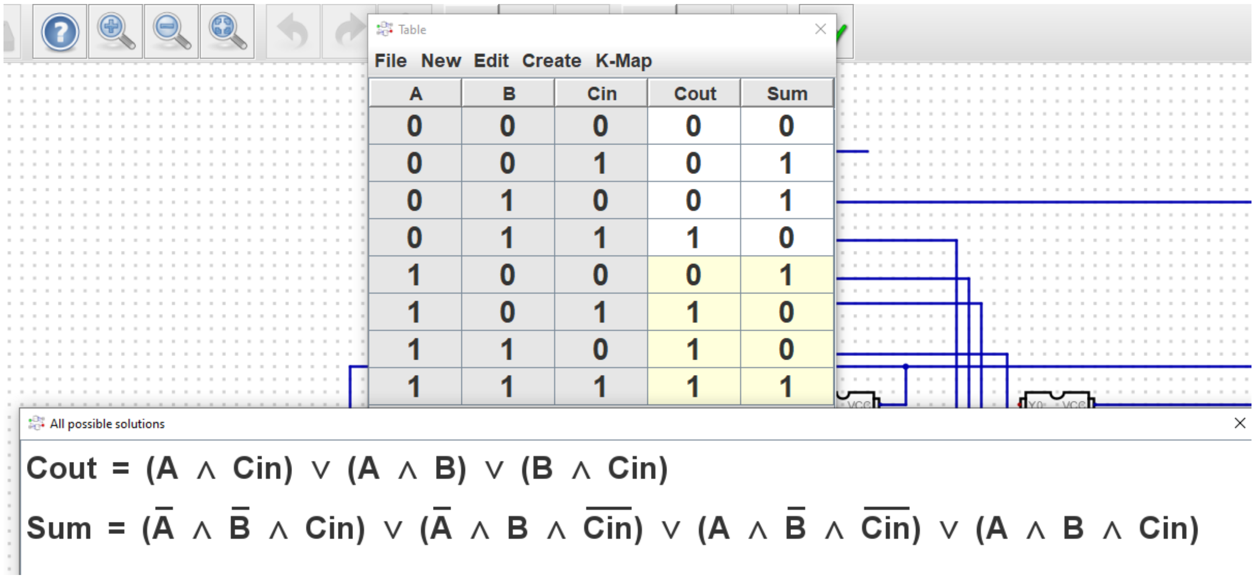
1	0	1	1	1	0
1	1	0	0	0	1
1	1	0	0	1	1
1	1	0	1	0	1
1	1	0	1	1	1
1	1	1	0	0	1
1	1	1	0	1	1
1	1	1	1	0	1
1	1	1	1	1	1

$Y1 = (0 \wedge B) \vee (1 \wedge \overline{B} \wedge \overline{C}) \vee$

3) Full adder decoder



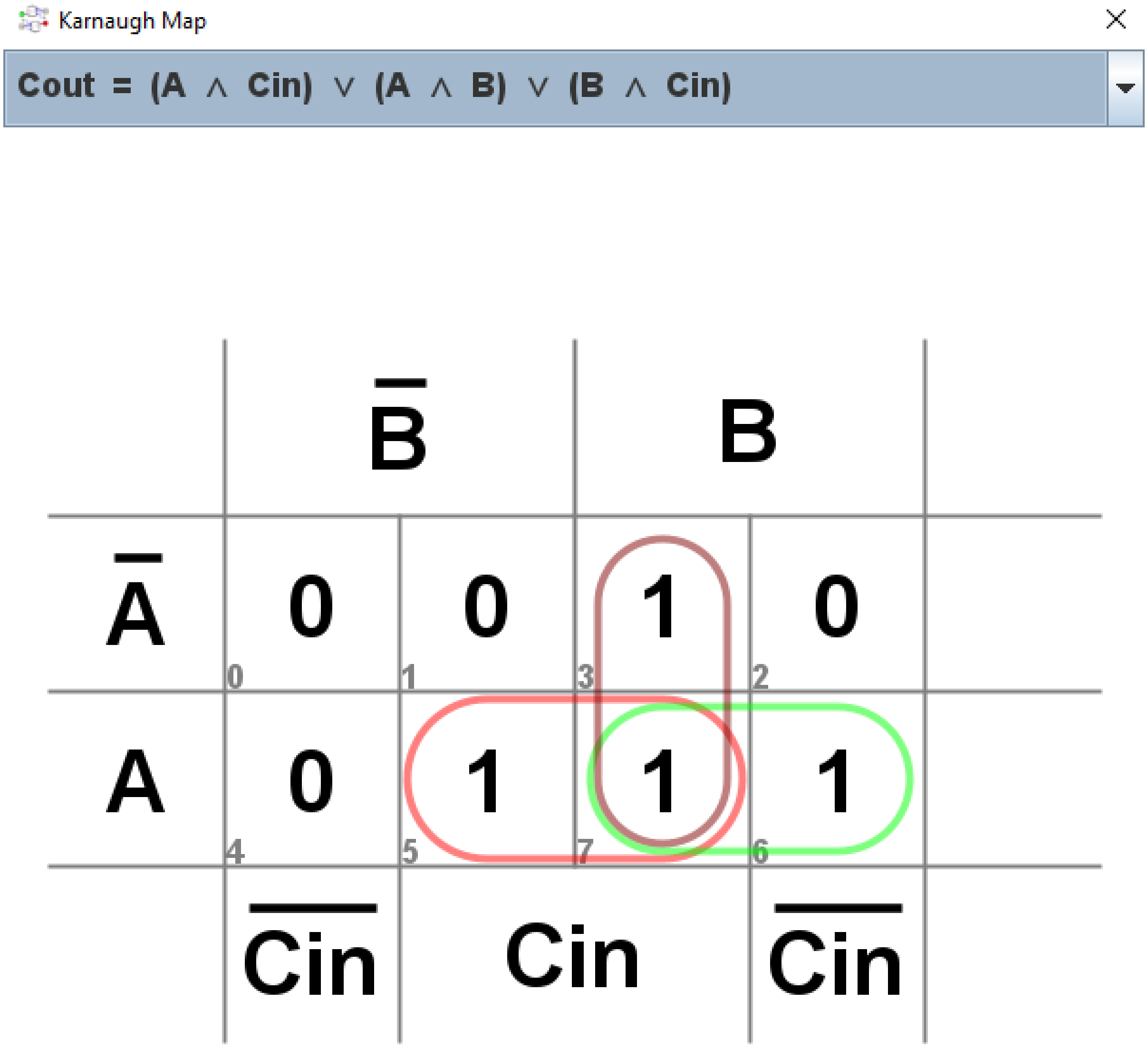
Analysis: The 74154 chip is a decoder chip, where we use 2 types of each chip that connect to the decoder. 7404 represents NOT and 7432 represents OR. We start with 3 inputs in the decoder of A, B, and Cin. We will not be using every single port, only the ones that we sufficiently need. The Sum can be shown with the 4 numbers, and Cout is represented with those numbers as well. The decoder is especially configured to work with Low. As we do this, we can get the appropriate Sum and Cout.



A	B	Cin	Cout	Sum
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

$$Cout = (A \land Cin) \lor (A \land B) \lor (B \land Cin)$$

$$Sum = (\overline{A} \land \overline{B} \land Cin) \lor (\overline{A} \land B \land \overline{Cin}) \lor (A \land \overline{B} \land \overline{Cin}) \lor (A \land B \land Cin)$$



Cout = (A ∧ Cin) ∨ (A ∧ B) ∨ (B ∧ Cin)

	$\overline{B}$		B	
$\overline{A}$	0 0	0 1	1 3	0 2
A	0 4	1 5	1 7	1 6
	$\overline{Cin}$	Cin		$\overline{Cin}$