

Topic X

# Final Design Project

April 8<sup>th</sup>, 2021

Haojun Zhao

400179164

zhaoh52

## I. Analytical Solution

In the final design project, we are asked to display our student number using sequential logic design. We are going to dealing with the 7-segment display and the decoder chip. In my situation, my student number is 400179164. First thing is to convert these numbers into binary numbers. Here are the conversions:

Decimal number	4 Binary numbers
4	0100
0	0000
0	0000
1	0001
7	0111
9	1001
1	0001
6	0110
4	0100

In here, I use  $q_0, q_1, q_2, q_3$  to represent my 4-binary code. For example, in the case 0100,  $q_0 = 0, q_1 = 1, q_2 = 0, q_3 = 0$ . In this stage, we may can do some simplification for our truth table if we do not have 8 and 9 in student number. Unluckily, I have 9 in my student number which means that I cannot eliminate the  $q_0$  from my truth table. Furthermore, I have repeated numbers in my student number. I have two four, two one and two zero. It means that I need extra bit to track the repeated number in my truth table. Here is the picture to show the part of my truth table:

current state					cs memory
q0	q1	q2	q3	q4	
0	1	0	0	0	0
0	0	0	0	0	0
0	0	0	0	1	1
0	0	0	1	0	X
0	1	1	1		
1	0	0	1		X
0	0	0	1	1	
0	1	1	0		X
0	1	0	0	1	

From the previous graph, I use q4 as my extra bit to track repeated number in my student number. I use 0 to track when the number is showed up first time. I use 1 to track when the number is showed up second time. I use X (do not care situation) to show the numbers are not repeated. The advantage to use X in here is that it can improve some flexibility for my K-mapping afterward. Next thing I need to do is to identify the next state for each current state. For example, the next state for 01000 is 00000. Here is the table for the current state and their corresponding next state.

current state					cs memory	Next State				NS Memory
q0	q1	q2	q3	q4		Q0	Q1	Q2	Q3	Q4
0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0	0	1	0
0	0	0	1	0	0	1	1	1	1	X
0	1	1	1	X	1	0	0	1	1	X
1	0	0	1	X	0	0	0	1	1	1
0	0	0	1	1	0	1	1	0	0	X
0	1	1	0	X	0	1	0	0	0	1
0	1	0	0	1	0	1	0	0	0	0

I use Q0, Q1, Q2, Q3, Q4 to represent the next state for the truth table. The next thing is to use current state and their next state to identify the action for each flip flop will do. We will use the excitation table from the lecture note.

## JK Excitation Table

<b>q (current state)</b>	<b>Q (next state)</b>	<b>J</b>	<b>K</b>
0	0	0	X
0	1	1	X
1	0	X	1
1	1	X	0

I will do a sample calculation to show how I determine the action for the JK flip flop by using the excitation table. Let us consider the current state is 00010. According to the table, the next state should be 0111X. the following table can be concluded:

<b>q (current state)</b>	<b>Q (next state)</b>	<b>J</b>	<b>K</b>
0	0	0	X
0	1	1	X
0	1	1	X
1	1	X	0
0	X	X	X

In here, since q0 is 0, the next state for q0 is also 0. The excitation table states that the J0 and K0 should be 0 and X. if K is 1 or 0, the output for this JK flip flop is either hold or reset. It will return 0 for the next state. That is reason why K is X here. Similar reason for q1 and Q1. The current state for q1 is 0, the next state is 1. The JK flip flop will either set to 1 or toggle. The same reason for the q2. q3 has a different reason. We can tell the current state for q3 is 0, the next state is X. Since we know the next state is X.,

we do not really care the output the JK flip flop. That is the reason why J3 and K3 are X because we do not care the output for that. It can be anything. To conclude this step, here is the table included all the JK input for my student number. They are all following the similar analysis process showed above.

Flip flop input required									
J0	K0	J1	K1	J2	K2	J3	K3	J4	K4
0	X	X	1	0	X	0	X	0	X
0	X	0	X	0	X	0	X	1	X
0	X	0	X	0	X	1	X	X	1
0	X	1	X	1	X	X	0	X	X
1	X	X	1	X	1	X	0	X	X
X	1	0	X	0	X	X	0	1	0
0	X	1	X	1	X	X	1	X	X
0	X	X	0	X	1	0	X	1	0
0	X	X	0	0	X	0	X	X	1

The next thing I need to do is the K-mapping. K-mapping will give me the specific logic we need to implement the student number. From the lecture, we know that there are two methods for the K-mapping. One is called SOP, the other one is called POS. in here, I will use sum of products (SOP) to perform my K-mapping process. The key for SOP is to find all one in  $2^n$  boxes. The boxes can overlap with each other and it can go the other side in the specific situations. In here, I have 5 different variables. The drawing boxes will be different from the normal K-mapping which has 4 variables. I will show how I draw the boxes below:

**J0: q2\*q3**

q0q1q2\q3q4 <sup>↓</sup>	00 <sup>↓</sup>	01 <sup>↓</sup>	11 <sup>↓</sup>	10 <sup>↓</sup>
000 <sup>↓</sup>	0 <sup>↓</sup>	0 <sup>↓</sup>	0 <sup>↓</sup>	0 <sup>↓</sup>
001 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>
011 <sup>↓</sup>	0 <sup>↓</sup>	0 <sup>↓</sup>	1 <sup>↓</sup>	1 <sup>↓</sup>
010 <sup>↓</sup>	0 <sup>↓</sup>	0 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>
100 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>
101 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>
111 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>
110 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>

From the lab 6, we have discussed the way to find the logic behind K-mapping. I will not be discussed that in here anymore. I will directly give the answer for each input. J0 (SOP) is  $(q_2 * q_3)$ .

### K0: 1

$q_0 q_1 q_2 \setminus q_3 q_4$	00 $\leftarrow$	01 $\leftarrow$	11 $\leftarrow$	10 $\leftarrow$
000 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
001 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
011 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
010 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
100 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	1 $\leftarrow$	1 $\leftarrow$
101 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
111 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
110 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$

Since for K1 we only have do not care situation and 1. In conclusion, K0 is ALWAYS 1.

### J1: $!q_0 * q_3$

$q_0 q_1 q_2 \setminus q_3 q_4$	00 $\leftarrow$	01 $\leftarrow$	11 $\leftarrow$	10 $\leftarrow$
000 $\leftarrow$	0 $\leftarrow$	0 $\leftarrow$	1 $\leftarrow$	1 $\leftarrow$
001 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
011 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
010 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
100 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	0 $\leftarrow$	0 $\leftarrow$
101 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
111 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
110 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$

K1:  $q3 + (!q2 * !q4)$

$q0q1q2\backslash q3q4^{\leftarrow}$	$00^{\leftarrow}$	$01^{\leftarrow}$	$11^{\leftarrow}$	$10^{\leftarrow}$
$000^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$001^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$011^{\leftarrow}$	0 $\leftarrow$	0 $\leftarrow$	1 $\leftarrow$	1 $\leftarrow$
$010^{\leftarrow}$	1 $\leftarrow$	0 $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$100^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$101^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$111^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$110^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$

J2 :  $!q0 * q3$

$q0q1q2\backslash q3q4^{\leftarrow}$	$00^{\leftarrow}$	$01^{\leftarrow}$	$11^{\leftarrow}$	$10^{\leftarrow}$
$000^{\leftarrow}$	0 $\leftarrow$	0 $\leftarrow$	1 $\leftarrow$	1 $\leftarrow$
$001^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$011^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$010^{\leftarrow}$	0 $\leftarrow$	0 $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$100^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	0 $\leftarrow$	0 $\leftarrow$
$101^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$111^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$110^{\leftarrow}$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$

## K2: 1

$q_0 q_1 q_2 \setminus q_3 q_4 \Leftarrow$	$00 \Leftarrow$	$01 \Leftarrow$	$11 \Leftarrow$	$10 \Leftarrow$
$000 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$001 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$011 \Leftarrow$	1 $\Leftarrow$	1 $\Leftarrow$	1 $\Leftarrow$	1 $\Leftarrow$
$010 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$100 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$101 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$111 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$110 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$

Since for K2 we only have do not care situation and 1. In conclusion, K1 is ALWAYS

1.

## J3: !q1 \* q4

$q_0 q_1 q_2 \setminus q_3 q_4 \Leftarrow$	$00 \Leftarrow$	$01 \Leftarrow$	$11 \Leftarrow$	$10 \Leftarrow$
$000 \Leftarrow$	0 $\Leftarrow$	1 $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$001 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$011 \Leftarrow$	0 $\Leftarrow$	0 $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$010 \Leftarrow$	0 $\Leftarrow$	0 $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$100 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$101 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$111 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$110 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$

K3:  $(!q0 * !q1 * q4)$

$q0q1q2\backslash q3q4 \Leftarrow$	$00 \Leftarrow$	$01 \Leftarrow$	$11 \Leftarrow$	$10 \Leftarrow$
$000 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	1 $\Leftarrow$	0 $\Leftarrow$
$001 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$011 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	0 $\Leftarrow$	0 $\Leftarrow$
$010 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$100 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	0 $\Leftarrow$	0 $\Leftarrow$
$101 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$111 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$110 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$

J4:  $!q1 + q2$

$q0q1q2\backslash q3q4 \Leftarrow$	$00 \Leftarrow$	$01 \Leftarrow$	$11 \Leftarrow$	$10 \Leftarrow$
$000 \Leftarrow$	1 $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$001 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$011 \Leftarrow$	1 $\Leftarrow$	1 $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$010 \Leftarrow$	0 $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$100 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	1 $\Leftarrow$	1 $\Leftarrow$
$101 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$111 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$110 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$

K4:  $!q0 * !q2$

$q0q1q2\backslash q3q4 \Leftarrow$	$00 \Leftarrow$	$01 \Leftarrow$	$11 \Leftarrow$	$10 \Leftarrow$
$000 \Leftarrow$	X $\Leftarrow$	1 $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$001 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$011 \Leftarrow$	0 $\Leftarrow$	0 $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$010 \Leftarrow$	X $\Leftarrow$	1 $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$100 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	0 $\Leftarrow$	0 $\Leftarrow$
$101 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$111 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$
$110 \Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$	X $\Leftarrow$

I conclude a table to show the action for the each JK input:

J0 : $q_2 * q_3$	K0: 1
J1 : $\overline{q_0} * q_3$	K1: $q_3 + (\overline{q_2} * \overline{q_4})$
J2 : $\overline{q_0} * q_3$	K2: 1
J3: $\overline{q_1} * q_4$	K3: $\overline{q_1} * \overline{q_0} * q_4$
J4: $\overline{q_1} + q_2$	K4: $\overline{q_0} * \overline{q_2}$

In the actual JK flip flop, we do not have K directly. We only have !K. It will simplify our logic a little bit and we can use De Morgan's Laws:  $\overline{XY} = \bar{X} + \bar{Y}$  and  $\overline{X+Y} = \bar{X} + \bar{Y}$  to simplify the logic even further. Here is the final simplified version can directly used for the JK flip flop (SOP):

J0 : $q_2 * q_3$	!K0: 0
J1 : $\overline{q_0} * q_3$	!K1: $\overline{q_3} * (q_2 + q_4)$
J2 : $\overline{q_0} * q_3$	!K2: 0
J3: $\overline{q_1} * q_4$	!K3: $(q_0 + q_1) + \overline{q_4}$
J4: $\overline{q_1} + q_2$	!K4: $q_0 + q_2$

Next, let us check whether this implementation is right or wrong. We can use the current state and next state to check that.

J0	K0	J1	K1	J2	K2	J3	K3	J4	K4
0	X	X	1	0	X	0	X	0	X
0	X	0	X	0	X	0	X	1	X
0	X	0	X	0	X	1	X	X	1
0	X	1	X	1	X	X	0	X	X
1	X	X	1	X	1	X	0	X	X
X	1	0	X	0	X	X	0	1	0
0	X	1	X	1	X	X	1	X	X
0	X	X	0	X	1	0	X	1	0
0	X	X	0	0	X	0	X	X	1

To conclude, it is a correct way to represent the logic. Next thing I need to consider is the number of gates. The following table conclude how many gates and chips we need to implement this SOP logic.

Gate name	AND gate	OR gate
quantity	4	5
Gate name	AND chip	OR chip

Gate name	AND chip	OR chip
quantity	1	2
Gate name	AND gate	OR gate

Next, I will perform the same logic but using the POS instead of using SOP. There is a key difference between POS and SOP. POS is trying to find all the 0 but SOP is to find all 1 in the table. Here shows how I group the boxes for POS solution:

**J0 : q1 \* q3**

q0q1q2\q3q4 <sup>↓</sup>	00 <sup>↓</sup>	01 <sup>↓</sup>	11 <sup>↓</sup>	10 <sup>↓</sup>
000 <sup>↓</sup>	0 <sup>↓</sup>	0 <sup>↓</sup>	0 <sup>↓</sup>	0 <sup>↓</sup>
001 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>
011 <sup>↓</sup>	0 <sup>↓</sup>	0 <sup>↓</sup>	1 <sup>↓</sup>	1 <sup>↓</sup>
010 <sup>↓</sup>	0 <sup>↓</sup>	0 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>
100 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>
101 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>
111 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>
110 <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>	X <sup>↓</sup>

**K0: 1**

$q_0 q_1 q_2 \setminus q_3 q_4 \leftarrow$	$00 \leftarrow$	$01 \leftarrow$	$11 \leftarrow$	$10 \leftarrow$
$000 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$001 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$011 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$010 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$100 \leftarrow$	X $\leftarrow$	X $\leftarrow$	1 $\leftarrow$	1 $\leftarrow$
$101 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$111 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$110 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$

**J1: q3 \* !q0**

$q_0 q_1 q_2 \setminus q_3 q_4 \leftarrow$	$00 \leftarrow$	$01 \leftarrow$	$11 \leftarrow$	$10 \leftarrow$
$000 \leftarrow$	0 $\leftarrow$	0 $\leftarrow$	1 $\leftarrow$	1 $\leftarrow$
$001 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$011 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$010 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$100 \leftarrow$	X $\leftarrow$	X $\leftarrow$	0 $\leftarrow$	0 $\leftarrow$
$101 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$111 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$110 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$

K1:  $(q_2 + !q_4)^* (!q_2 + q_3)$

$q_0q_1q_2\backslash q_3q_4 \leftarrow$	$00 \leftarrow$	$01 \leftarrow$	$11 \leftarrow$	$10 \leftarrow$
$000 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$001 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$011 \leftarrow$	0 $\leftarrow$	0 $\leftarrow$	1 $\leftarrow$	1 $\leftarrow$
$010 \leftarrow$	1 $\leftarrow$	0 $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$100 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$101 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$111 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$110 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$

J2:  $!q_0 * q_3$

$q_0q_1q_2\backslash q_3q_4 \leftarrow$	$00 \leftarrow$	$01 \leftarrow$	$11 \leftarrow$	$10 \leftarrow$
$000 \leftarrow$	0 $\leftarrow$	0 $\leftarrow$	1 $\leftarrow$	1 $\leftarrow$
$001 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$011 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$010 \leftarrow$	0 $\leftarrow$	0 $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$100 \leftarrow$	X $\leftarrow$	X $\leftarrow$	0 $\leftarrow$	0 $\leftarrow$
$101 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$111 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$110 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$

K2: 1

$q_0q_1q_2\backslash q_3q_4 \leftarrow$	$00 \leftarrow$	$01 \leftarrow$	$11 \leftarrow$	$10 \leftarrow$
$000 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$001 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$011 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$010 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$100 \leftarrow$	X $\leftarrow$	X $\leftarrow$	1 $\leftarrow$	1 $\leftarrow$
$101 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$111 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
$110 \leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$

J3: !q1 \* q4

$q_0 q_1 q_2 \setminus q_3 q_4 \leftarrow$	$00 \leftarrow$	$01 \leftarrow$	$11 \leftarrow$	$10 \leftarrow$
$000 \leftarrow$	$0 \leftarrow$	$1 \leftarrow$	$X \leftarrow$	$X \leftarrow$
$001 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$011 \leftarrow$	$0 \leftarrow$	$0 \leftarrow$	$X \leftarrow$	$X \leftarrow$
$010 \leftarrow$	$0 \leftarrow$	$0 \leftarrow$	$X \leftarrow$	$X \leftarrow$
$100 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$101 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$111 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$110 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$

K3: q4 \* !q2\* !q0

$q_0 q_1 q_2 \setminus q_3 q_4 \leftarrow$	$00 \leftarrow$	$01 \leftarrow$	$11 \leftarrow$	$10 \leftarrow$
$000 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$1 \leftarrow$	$0 \leftarrow$
$001 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$011 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$0 \leftarrow$	$0 \leftarrow$
$010 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$100 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$0 \leftarrow$	$0 \leftarrow$
$101 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$111 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$110 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$

J4: !q1 + q2

$q_0 q_1 q_2 \setminus q_3 q_4 \leftarrow$	$00 \leftarrow$	$01 \leftarrow$	$11 \leftarrow$	$10 \leftarrow$
$000 \leftarrow$	$1 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$001 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$011 \leftarrow$	$1 \leftarrow$	$1 \leftarrow$	$X \leftarrow$	$X \leftarrow$
$010 \leftarrow$	$0 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$100 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$1 \leftarrow$	$1 \leftarrow$
$101 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$111 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$
$110 \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$	$X \leftarrow$

K4:  $\overline{q_3} * \overline{q_2}$

$q_0 q_1 q_2 \backslash q_3 q_4$	00 $\leftarrow$	01 $\leftarrow$	11 $\leftarrow$	10 $\leftarrow$
000 $\leftarrow$	X $\leftarrow$	1 $\leftarrow$	X $\leftarrow$	X $\leftarrow$
001 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
011 $\leftarrow$	0 $\leftarrow$	0 $\leftarrow$	X $\leftarrow$	X $\leftarrow$
010 $\leftarrow$	X $\leftarrow$	1 $\leftarrow$	X $\leftarrow$	X $\leftarrow$
100 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	0 $\leftarrow$	0 $\leftarrow$
101 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
111 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$
110 $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$	X $\leftarrow$

I conclude a table to show the action for each JK input (POS):

J0 : $q_1 * q_3$	K0: 1
J1 : $\overline{q_0} * q_3$	K1: $(q_2 + \overline{q_4}) * (\overline{q_2} + q_3)$
J2 : $\overline{q_0} * q_3$	K2: 1
J3: $\overline{q_1} * q_4$	K3: $\overline{q_2} * \overline{q_0} * q_4$
J4: $\overline{q_1} + q_2$	K4: $\overline{q_3} * \overline{q_2}$

We can also use similar process to simplify the POS solution by using the similar process which I used for the SOP simplification. Here is the final simplified version can directly used for the JK flip flop (POS):

J0 : $q_1 * q_3$	$\overline{K}0: 0$
J1 : $\overline{q_0} * q_3$	$\overline{K}1: (\overline{q_2} * q_4) + (q_2 * \overline{q_3})$
J2 : $\overline{q_0} * q_3$	$\overline{K}2: 0$
J3: $\overline{q_1} * q_4$	$\overline{K}3: (q_0 + q_2) * \overline{q_4}$
J4: $\overline{q_1} + q_2$	$\overline{K}4: q_3 + q_2$

Next, let us check whether this implementation is right or wrong. We can use the current state and next state to check that.

J0	K0	J1	K1	J2	K2	J3	K3	J4	K4
0	X	X	1	0	X	0	X	0	X
0	X	0	X	0	X	0	X	1	X
0	X	0	X	0	X	1	X	X	1
0	X	1	X	1	X	X	0	X	X
1	X	X	1	X	1	X	0	X	X
X	1	0	X	0	X	X	0	1	0
0	X	1	X	1	X	X	1	X	X
0	X	X	0	X	1	0	X	1	0
0	X	X	0	0	X	0	X	X	1

To conclude, it is a correct way to represent the logic. Next thing I need to consider is the number of gates. The following table conclude how many gates and chips we need to implement this POS logic.

Gate name	AND gate	OR gate
quantity	6	4

Gate name	AND chip	OR chip
quantity	2	1

In the following discussion, I will discuss the other implantation method. I will use only NAND gate to represent the logic I need to do for the design lab. I will transfer the SOP solution into NAND only solution. From the previous discussion, we know the SOP solution is the following table:

J0 : $q_2 * q_3$	$!K_0: 0$
J1 : $\overline{q_0} * q_3$	$!K_1: \overline{q_3} * (q_2 + q_4)$
J2 : $\overline{q_0} * q_3$	$!K_2: 0$
J3: $\overline{q_1} * q_4$	$!K_3: (q_0 + q_1) + \overline{q_4}$
J4: $\overline{q_1} + q_2$	$!K_4: q_0 + q_2$

I will use the De Morgan's Laws to do each transformation. Here is the completed solution for the NAND version:

J0 : $\overline{\overline{q_2} * \overline{q_3}}$	$!K_0: 0$
J1 : $\overline{\overline{q_0} * \overline{q_3}}$	$!K_1: \overline{\overline{q_3} * \overline{q_2} * \overline{q_4}}$
J2 : $\overline{\overline{q_0} * \overline{q_3}}$	$!K_2: 0$
J3: $\overline{\overline{q_1} * \overline{q_4}}$	$!K_3: \overline{\overline{q_0} * \overline{q_1} * q_4}$
J4: $\overline{\overline{q_2} * q_1}$	$!K_4: \overline{\overline{q_0} * \overline{q_2}}$

The following is the check whether this NAND implementation is right or wrong:

J0	K0	J1	K1	J2	K2	J3	K3	J4	K4
0	X	X	1	0	X	0	X	0	X
0	X	0	X	0	X	0	X	1	X
0	X	0	X	0	X	1	X	X	1
0	X	1	X	1	X	X	0	X	X
1	X	X	1	X	1	X	0	X	X
X	1	0	X	0	X	X	0	1	0
0	X	1	X	1	X	X	1	X	X
0	X	X	0	X	1	0	X	1	0
0	X	X	0	0	X	0	X	X	1

In conclusion, it is a right implementation, here are the tables conclude that the number of gates and chips we need to use for this NAND version.

Gate name	NAND gate
quantity	16

Gate name	NAND chip
quantity	4

By considering I decide to use the SOP solution for my following Multisim and Physical solution. There are couple reasons why I choose SOP solution. First, it uses only 4 AND gates and 5 OR gates. It uses least gates to represent the same logic compared to the POS solution and NAND solution. Secondly, I only need to use three extra chips to do this logic. It is also doable for the physical build. Thirdly, the expression for the SOP solution is easy to follow. It does not include some tricky and hard expression for the SOP solution.

To conclude the analytical part for my final design project. I changed my student number 400179164 into 4 binary numbers first and then generate a state transition table and truth table. I also set extra bit to track the repeated number in my student number. Then I use K-mapping to get the algebra expression. After comparing the SOP, POS, and NAND solution, I decided to use SOP solution. Here are the main results to conclude my analytical solution:

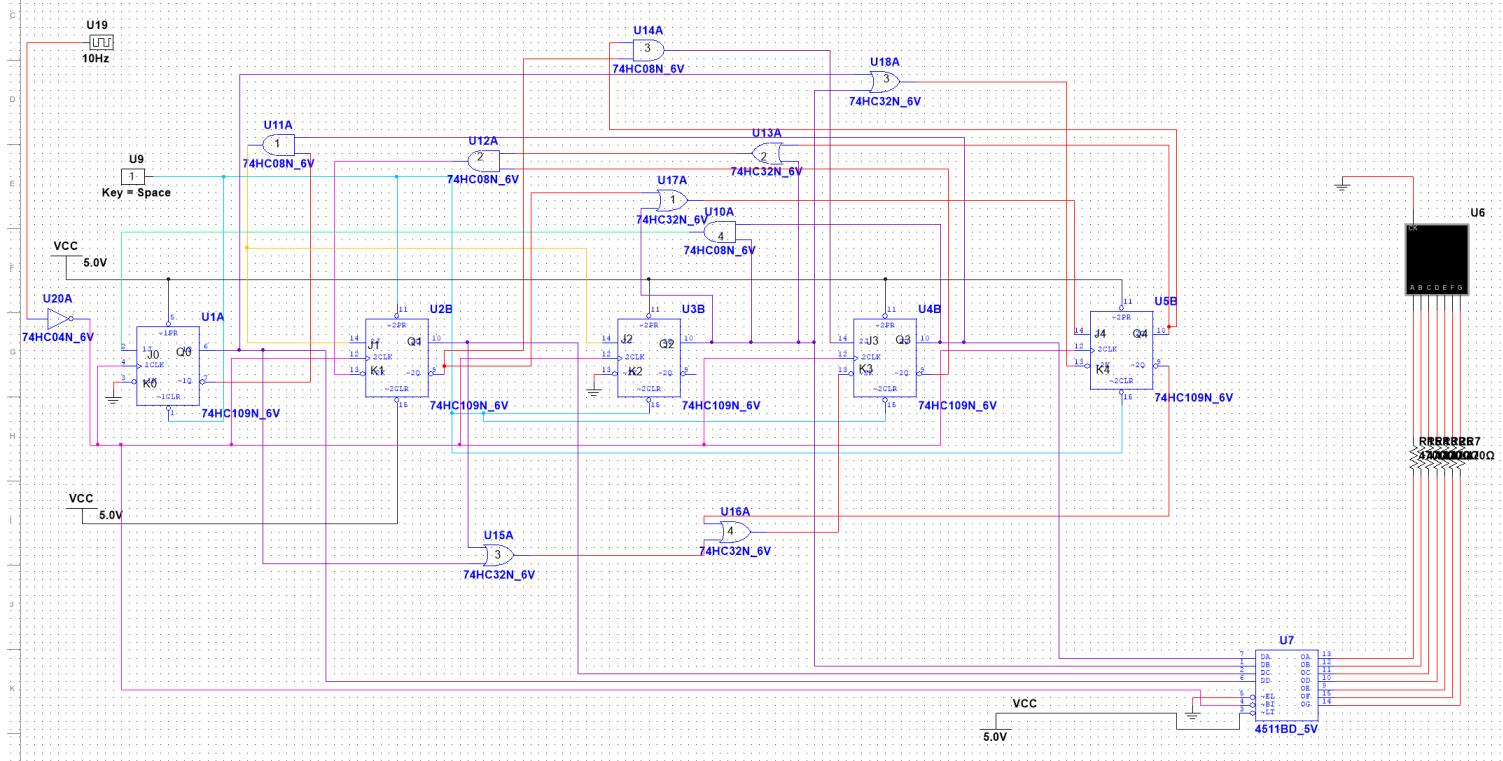
current state					cs memory	Next State				NS Memor
q0	q1	q2	q3	q4	Q0	Q1	Q2	Q3	Q4	
0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	
0	0	0	0	1	0	0	0	1	0	
0	0	0	1	0	0	1	1	1	X	
0	1	1	1	X	1	0	0	1	X	
1	0	0	1	X	0	0	0	1	1	
0	0	0	1	1	0	1	1	0	X	
0	1	1	0	X	0	1	0	0	1	
0	1	0	0	1	0	0	1	0	0	

Flip flop input required									
J0	K0	J1	K1	J2	K2	J3	K3	J4	K4
0	X	X	1	0	X	0	X	0	X
0	X	0	X	0	X	0	X	1	X
0	X	0	X	0	X	1	X	X	1
0	X	1	X	1	X	X	0	X	X
1	X	X	1	X	1	X	0	X	X
X	1	0	X	0	X	X	0	1	0
0	X	1	X	1	X	X	1	X	X
0	X	X	0	X	1	0	X	1	0
0	X	X	0	0	X	0	X	X	1

J0 : $q_2 * q_3$	!K0: 0
J1 : $\overline{q_0} * q_3$	!K1: $\overline{q_3} * (q_2 + q_4)$
J2 : $\overline{q_0} * q_3$	!K2: 0
J3: $\overline{q_1} * q_4$	!K3: $(q_0 + q_1) + \overline{q_4}$
J4: $\overline{q_1} + q_2$	!K4: $q_0 + q_2$

## II. Multisim solution:

From the analytical solution, I decide to use the SOP solution for my Multisim and physical build solution. Here is the screenshot shows my Multisim circuit:



First, I want to address the color code for cables. Here is the table conclude the color code:

Cable colour	function
Yellow	Connect number one AND gate to J1 and J2
Blue	Connect pre-set and clear to clock signal
Green	Connect number 4 AND gate output to J0
Black	Connect pre-set and clear to VCC
Pink	Every JK flip flop's clock signal
Purple	Output for Q0, Q1, Q2, Q3

Red	General connect each logic gates and decoder chip
-----	---

In this paragraph, I will discuss how I generate this Multisim solution. From the analytical solution, I have the following table:

J0 : $q_2 * q_3$	$\neg K_0: 0$
J1 : $\overline{q_0} * q_3$	$\neg K_1: \overline{q_3} * (q_2 + q_4)$
J2 : $\overline{q_0} * q_3$	$\neg K_2: 0$
J3: $\overline{q_1} * q_4$	$\neg K_3: (q_0 + q_1) + \overline{q_4}$
J4: $\overline{q_1} + q_2$	$\neg K_4: q_0 + q_2$

We do not need any inverter in this lab because JK flip flop has the inverted output automatically. The only gates I need to add is the AND gate and OR gate. From the analytical solution, I know I need 5 OR gates and 4 AND gates. I numbered each gate to make everything clear and easy to follow. The output of number 4 AND gate is the input for J0. The inputs for number 4 AND gate are  $q_2$  and  $q_3$ . Since  $K_0$  is always 1,  $\neg K_0$  will always connect to the ground. To represent J1, I use number one AND gate which inputs are  $\neg q_0$  and  $q_3$ . I use yellow line to represent the output. For accomplish  $\neg K_1$ , I use number 2 AND gate and number 2 OR gate in my circuit. First,  $q_2$  and  $q_4$  will connect to the number 2 OR gate directly. The output for that OR gate will become one of input for the AND gate. The other input for AND gate is  $\neg q_3$ . The final output for number 2 AND gate is  $\neg K_1$ . Since J2 has the same logic as J1, I just use the other yellow wire to connect them.  $K_2$  is always zero, so  $\neg K_2$  is always wired into

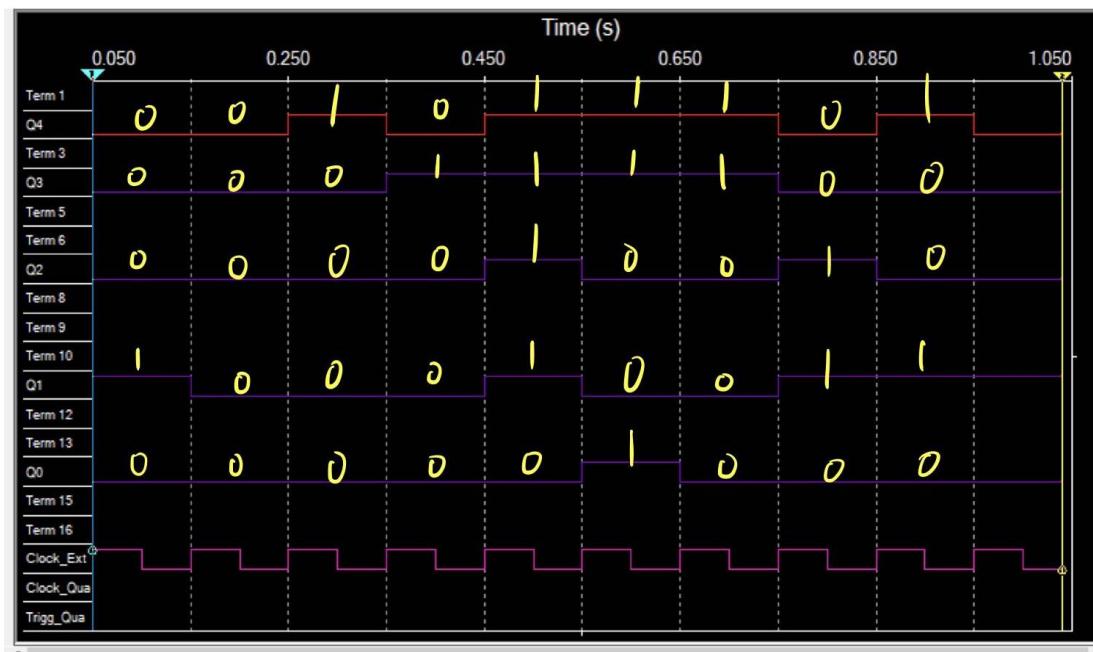
the ground. For J3, I use the number 3 AND gate. The input for number 3 AND gate is  $\neg q_1$  and  $q_4$ . To implement  $\neg K_3$ , I use number 4 and number 3 OR gate to do that. The inputs for number 3 OR gate are  $q_0$  and  $q_1$ . The inputs for number 4 OR gate are  $\neg q_4$  and number 3 OR gate's output. For J4, I use number 1 OR gate to do that. The input for that is  $\neg q_1$  and  $q_2$ . Finally, for K4, I use the second number 3 OR gate to do that. The input is  $q_0$  and  $q_2$ .

Next, I will attach the link for my Multisim solution:

<https://youtu.be/d2DJSAkdfEg>

from the video, the result is that: 4 → 0 → 0 → 1 → 7 → 9 → 1 → 6 → 4.

The following is the timing diagram for my whole circuit:



The pattern can be concluded as follow:

$$\begin{aligned}
 & 01000 \rightarrow 00000 \rightarrow 00001 \rightarrow 00010 \rightarrow 01111 \rightarrow 10011 \rightarrow 00011 \\
 \rightarrow & 01100 \rightarrow 01001
 \end{aligned}$$

Since the fifth digit is to trace the repeated number so the 4-binary numbers show in my timing diagram are the following:

$$\begin{aligned}
 & 0100 \rightarrow 0000 \rightarrow 0000 \rightarrow 0001 \rightarrow 0111 \rightarrow 1001 \rightarrow 0001 \\
 \rightarrow & 0110 \rightarrow 0100
 \end{aligned}$$

If we change these binary number into decimal ,we have:

$$4 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 7 \rightarrow 9 \rightarrow 1 \rightarrow 6 \rightarrow 4$$

To conclude my Multisim solution, the simulation from the Multisim shows that the numbers are 400179164. My student number is exactly 400179164. The timing diagram also shows the output for the Multisim solution is 400179164. That means the Multisim solution are consist with the design goal for this lab. I also can check if

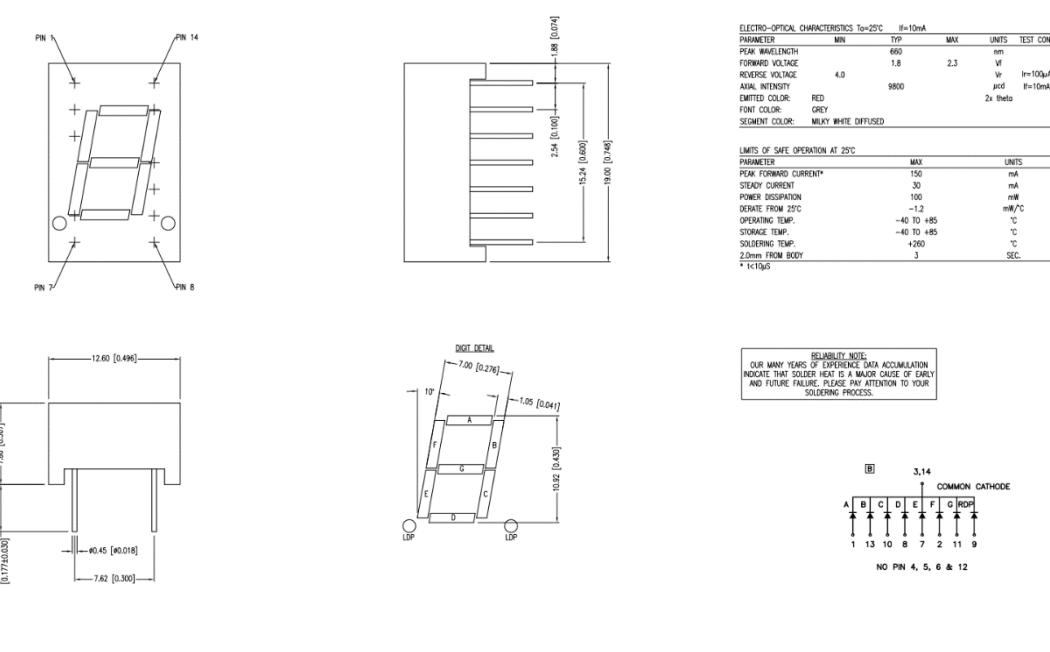
current state				cs memory	Next State				NS Memor
q0	q1	q2	q3	q4	Q0	Q1	Q2	Q3	Q4
0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0	1	0
0	0	0	1	0	0	1	1	1	X
0	1	1	1	X	1	0	0	1	X
1	0	0	1	X	0	0	0	1	1
0	0	0	1	1	0	1	1	0	X
0	1	1	0	X	0	1	0	0	1
0	1	0	0	1	0	1	0	0	0

The state transition table shows that the Multisim solution and analytical solution are matched with each other.

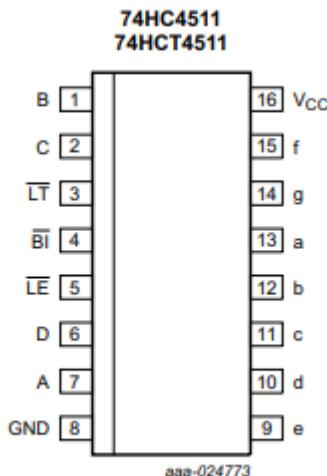
### **III. Physical build**

In this paragraph, I will discuss the process for building this final design project. As you can see, I use three breadboards to make this final design project happen. I have couple reason why I want to use three breadboards. Firstly, one breadboard does not have enough space to hold so many chips. Since I have 5 variables that means that I need 5 JK flip flops in the physical build. In conclude, three SN74HC109N chips will be used. I also need 2 OR gate chip and 1 AND gate chip. For the clock, I need one NOT chip to make sure clock is working. Finally, I need a decoder chip and 7 segment display to make the whole circuit. It is impossible to do this on only single breadboard. That are the reasons why I want to more than one breadboard. Since I bought three breadboards in accidently, I thought that it was better to separate things into three parts. One breadboard can hold the JK flip flops and clock signal, one breadboard can hold AND gates and OR gates to represent all logic staff. The final breadboard can hold the 7-segment display and decoder chip for my whole circuit.

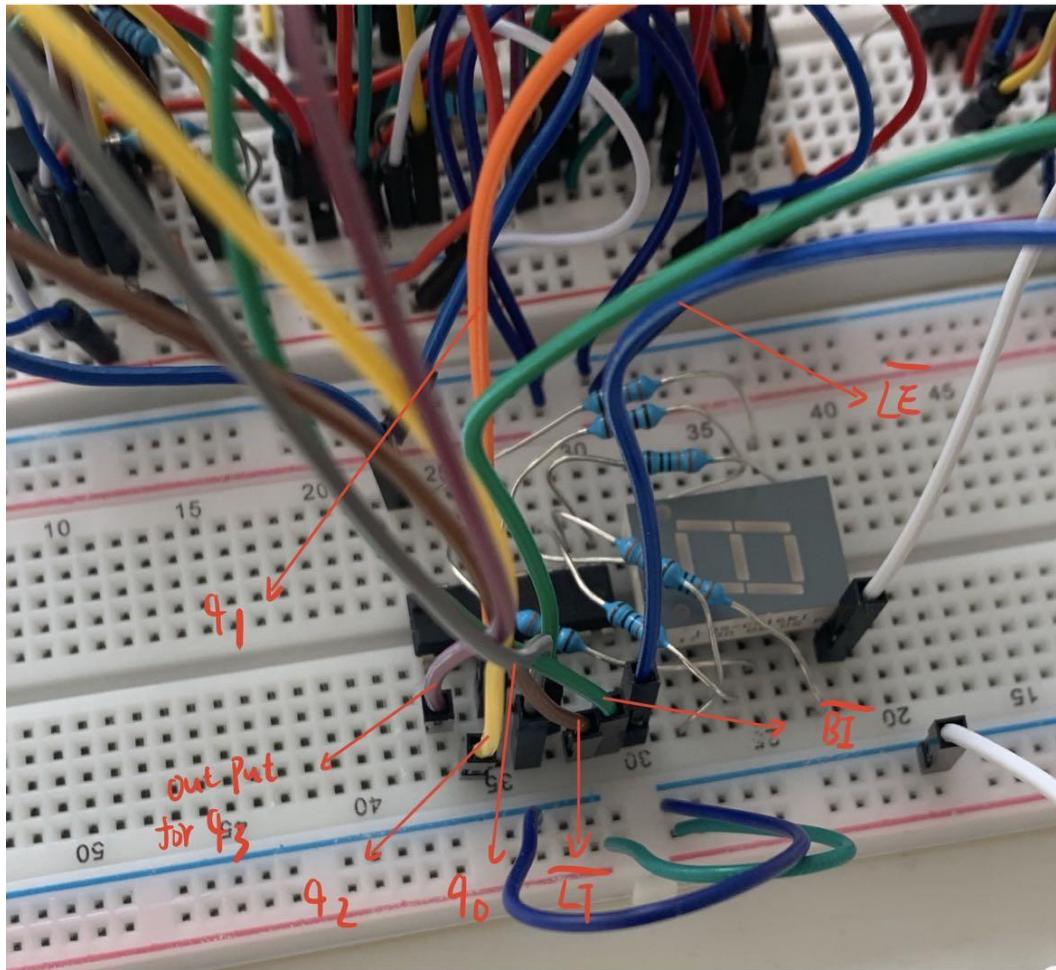
Let us start from the easiest part for this circuit. The setup for the 7-segment display. From the lab video from Dr. Minnick, we need to add resistors to each pin on the 7-segment display because we want to limiting the current through the 7-segment display. In here, I use  $330\Omega$  resistor instead of using  $470\Omega$  resistor because I do not have  $470\Omega$  resistor in my own lab kit.  $330\Omega$  can also make sure the current through 7 segment display is low enough. Here is the datasheet for the 7-segment display:



From the datasheet, we can tell the ground for this 7-segment display is 3 pin and 14 pins. They are common ground. I use 14 pins as my ground. In here, the other pins are responsible for the different segment for the 7-segment display. They will be connected to our decoder chip. Talking about the decoder chip, I got this decoder chip. SN74HC4511N from Peter's office. Here are the data sheet for that chip:

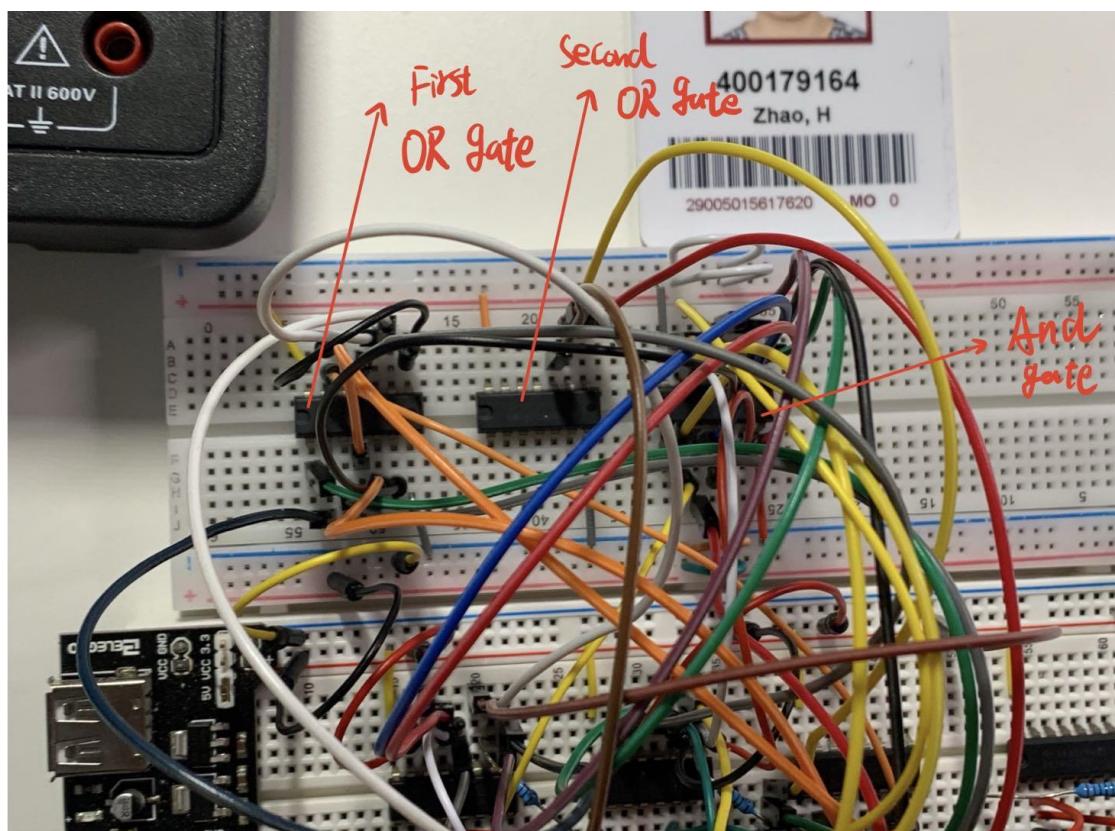
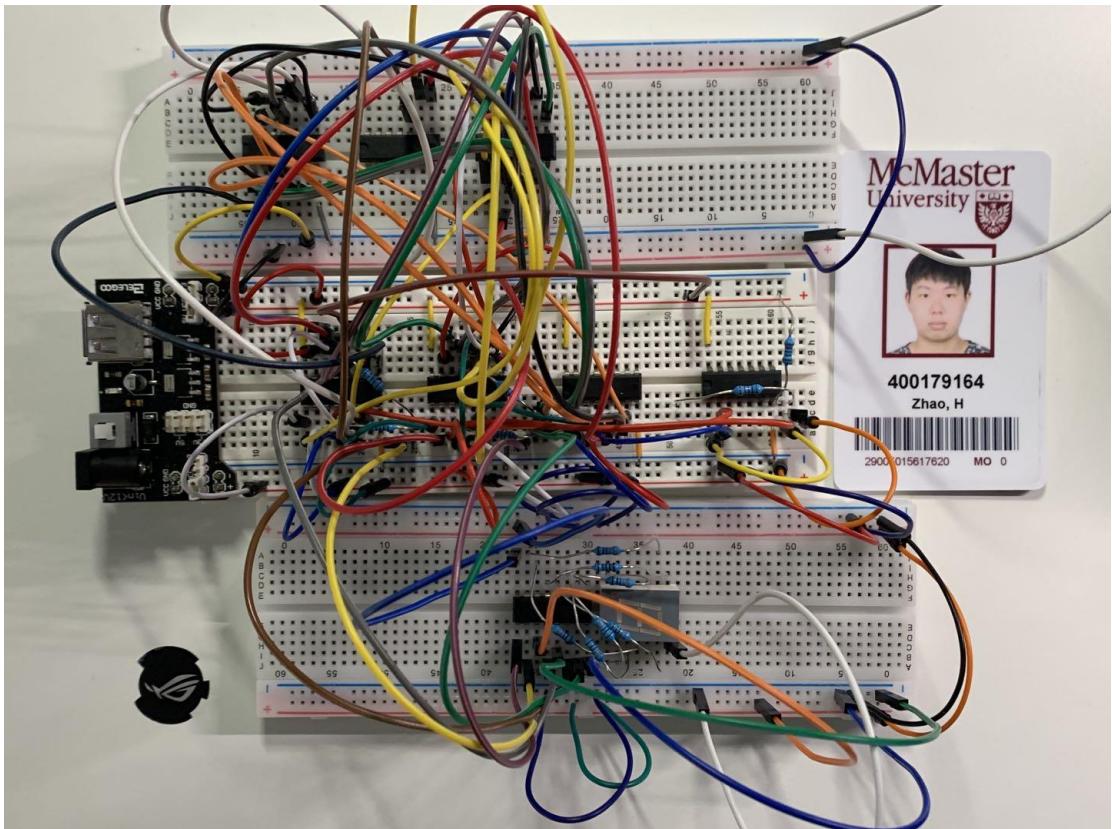


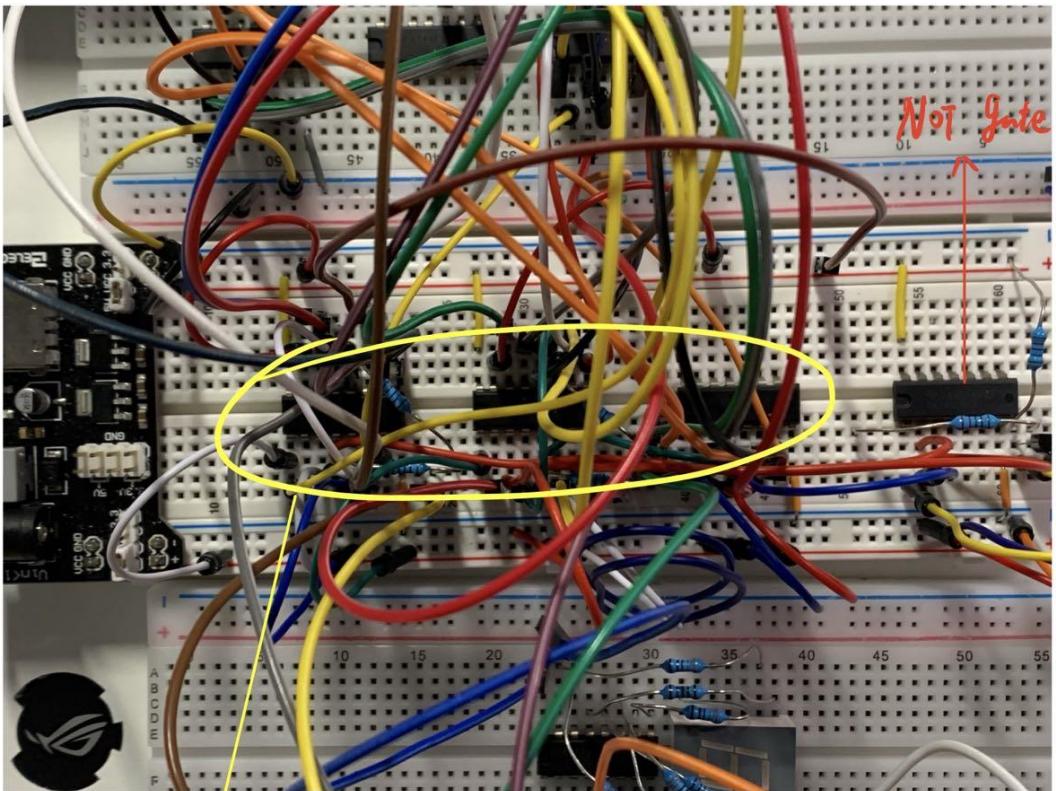
The ABCD from the left side from the decoder chip is the general input for the 7-segment display. I connect the ABCD pins to the Q3,Q2,Q1,Q0( the output for JK flip flop).  $\overline{LT}$  is connected to the VCC and  $\overline{LE}$  is connected to the ground.  $\overline{BI}$  will be connected to my clock signal to make sure the final output will blank every single time. Here is the final implementation for the 7-segment display set up.



That conclude how I complete the set up the 7-segment display.

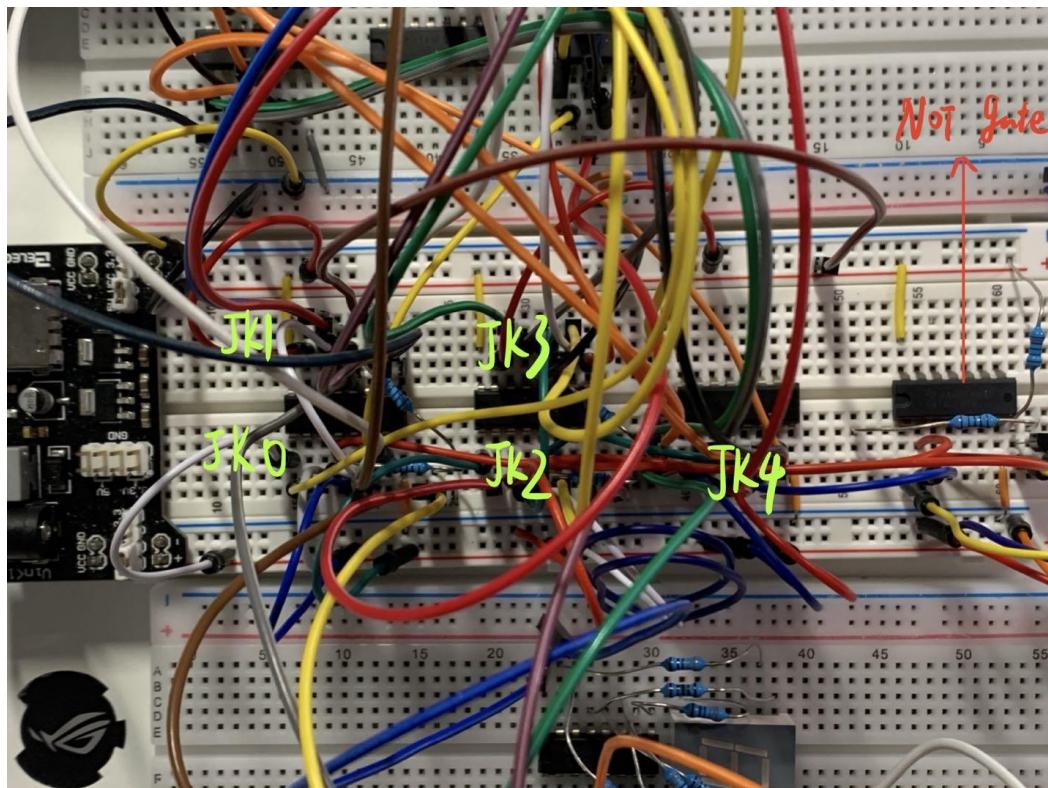
In this part, I will focus on how to complete the main logic design for the physical build. Here are the pictures for the overview for entire physical circuit:

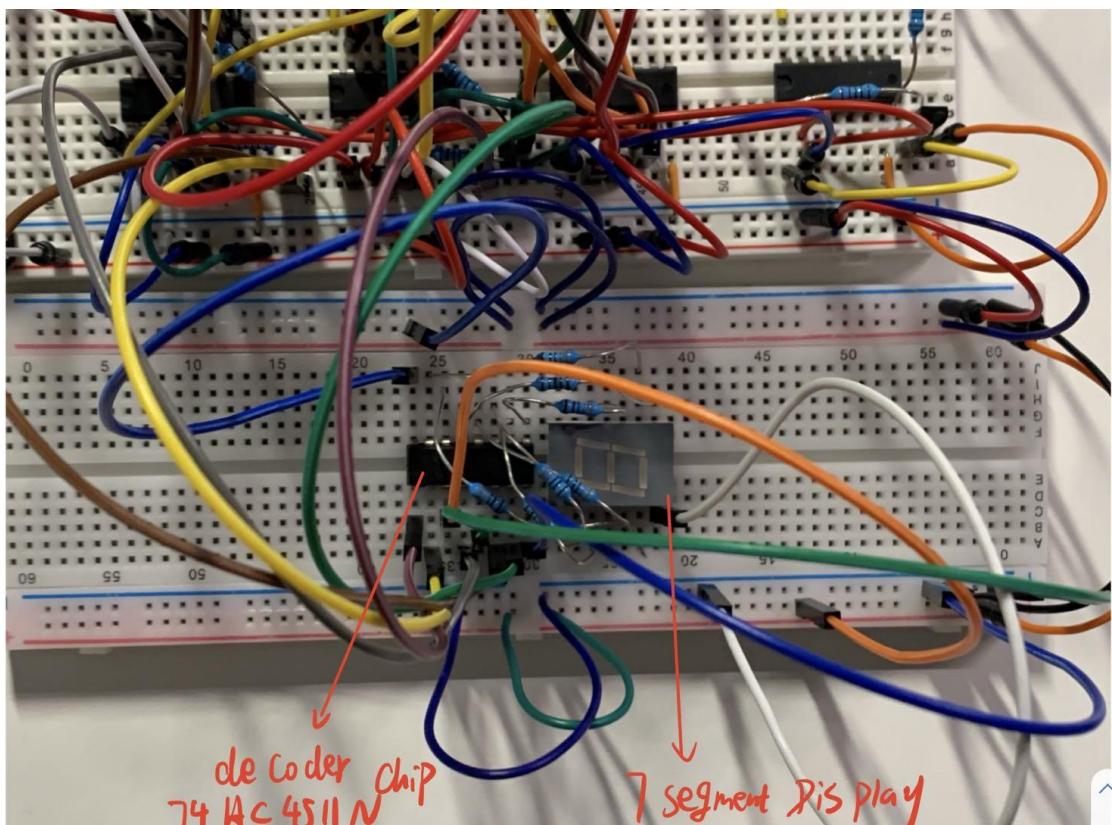
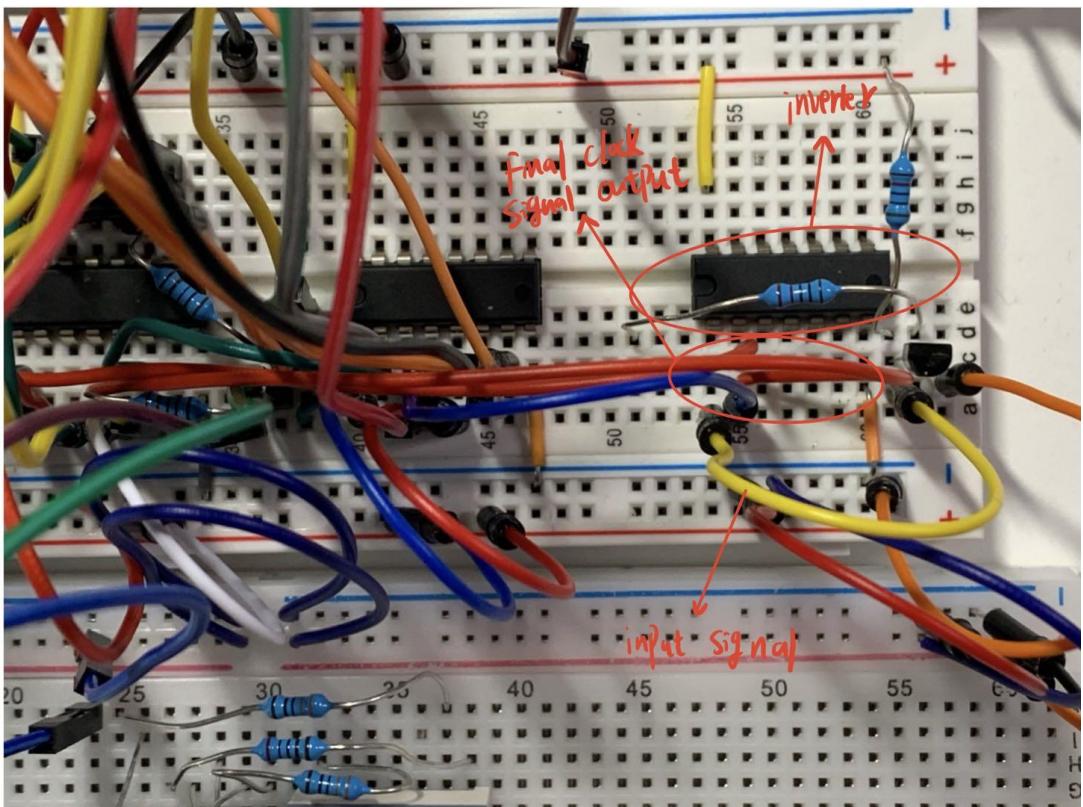


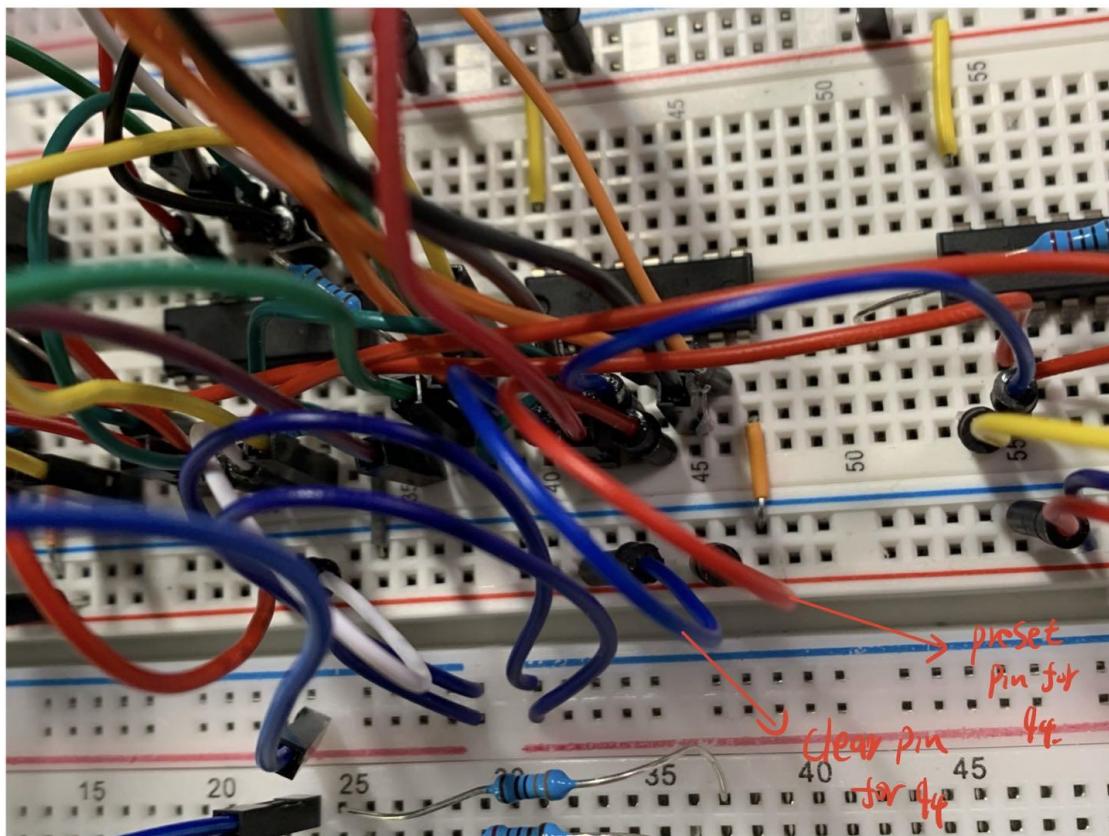
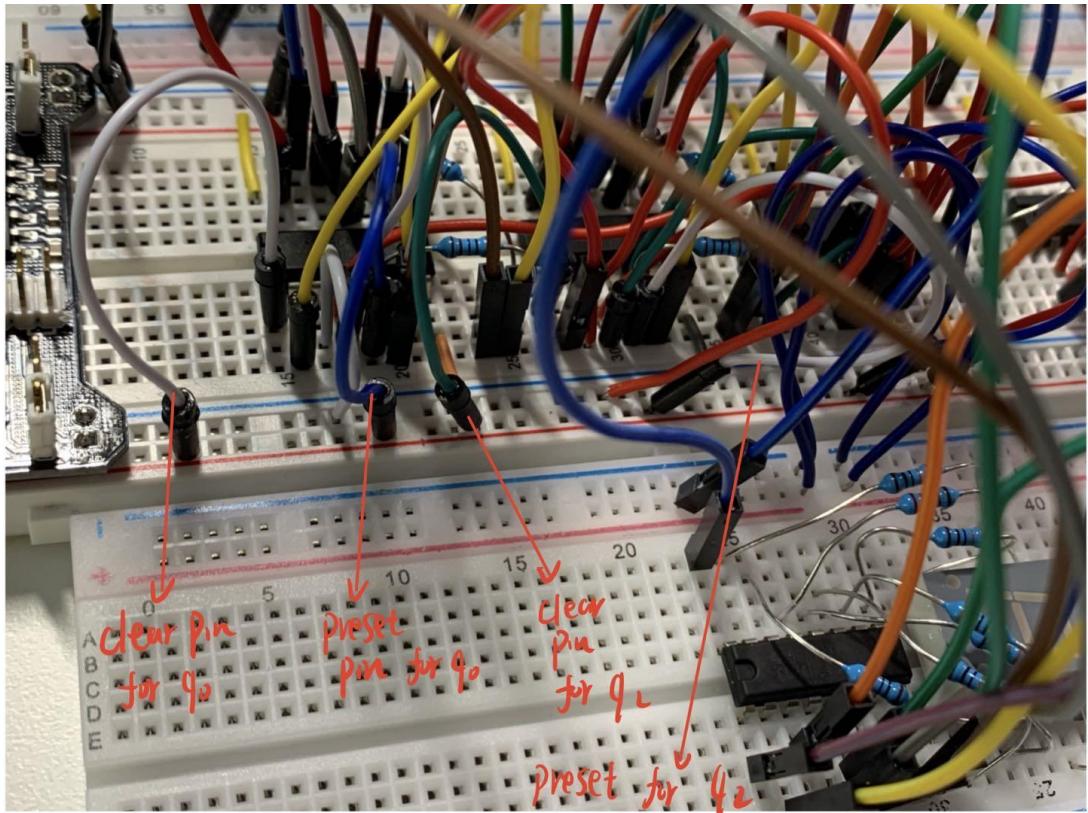


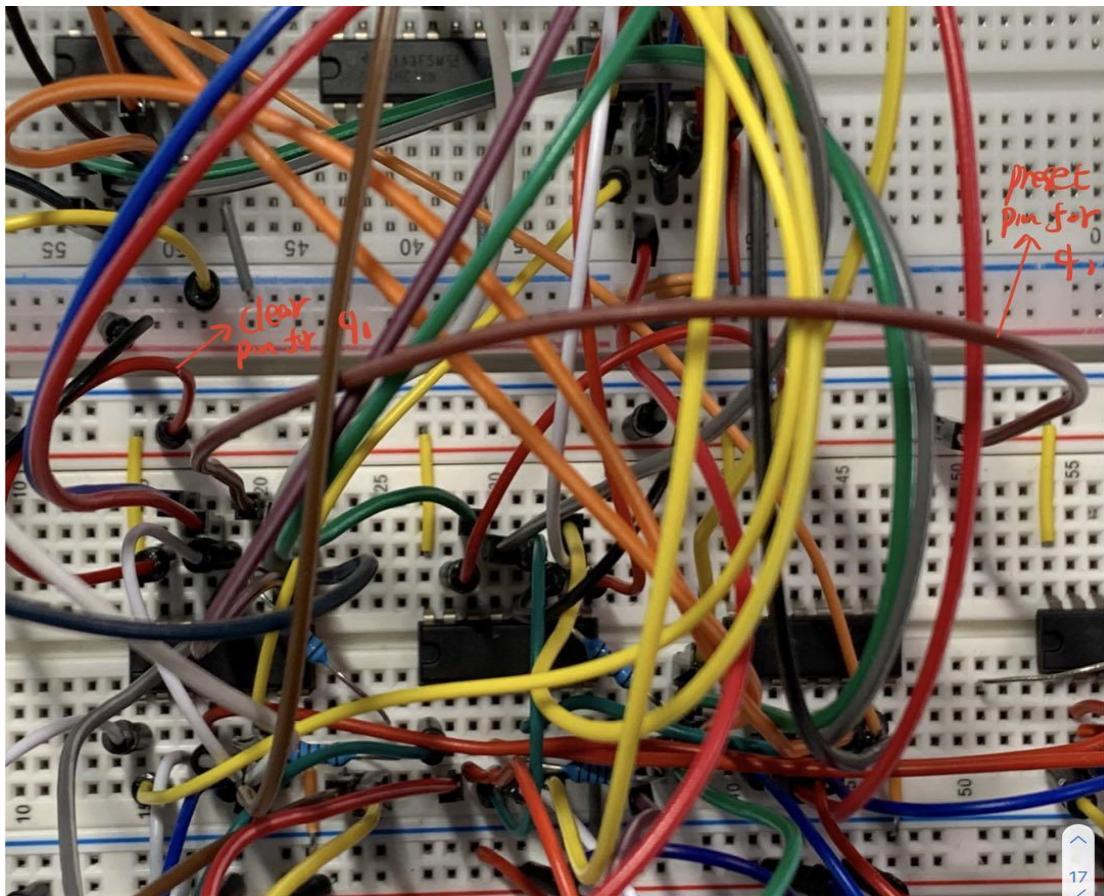
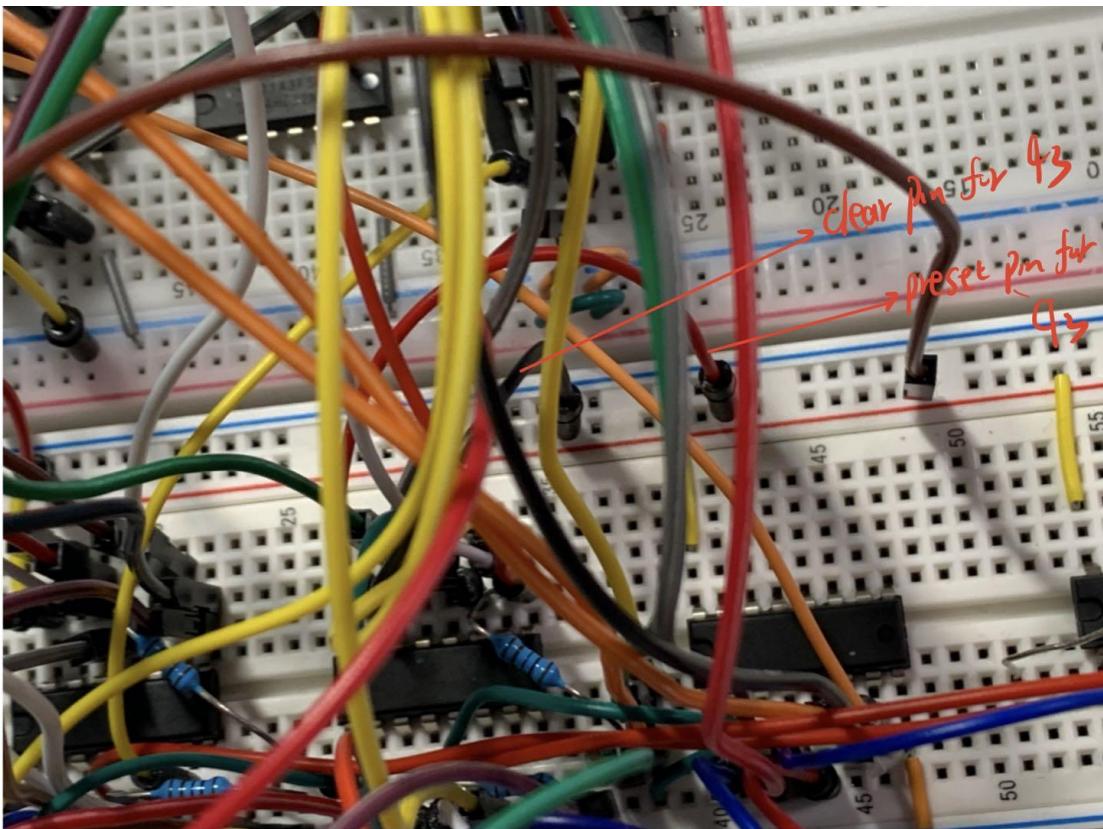
JK flip flops

^  
11 / 19









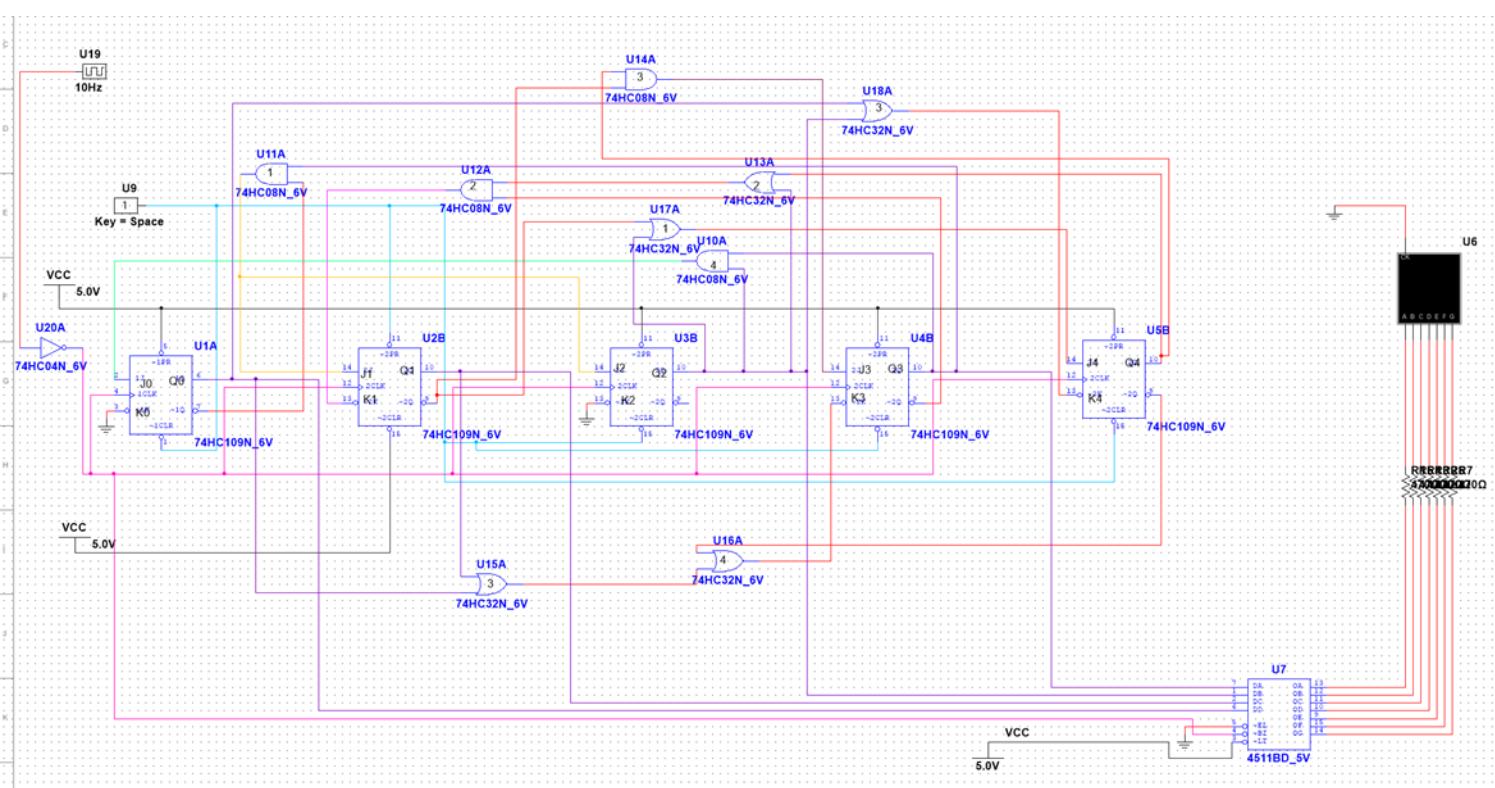
From previous pictures, I showed the overview for my circuit and inputs and clock signal. Next, I will discuss how I get to this stage.

From the previous discussion, I need 5 OR gates and 4 AND gates to represent the logic I have. That means that I need two SN74HC32N chips and one SN74HC00N chip. Since I have 5 variables ( $q_0, q_1, q_2, q_3, q_4$ ). I need 5 JK flip flop. That means that I need three SN74HC109N chips. Form the overview pictures for my circuit, I put all OR gates and AND gates on the top breadboards. At the middle breadboard, I put all JK flip flops and my clock signal. At the bottom breadboard, I put my LED output. That concludes the basic set-up for the circuit.

First thing I need to do is connect each chip's VCC and ground. I use some small jumper wire from our lab kit to do this. They are useful in this situation because I will use a lot of wires in this project. These small jumper wires can make the space as clean as possible. Next, the thing I need to do is to connect breadboards with each other. Since I only have one power supply, I can not give the power to three breadboards in separately. The solution to this is to use jumper wires from the breadboard which has power supply to the breadboard which does not has power. After these steps, I have already made sure each chip can work properly. Next thing is to redo the logic for the physical part.

Here I will give the pictures from My Multisim solution and analytical solution which can help me redo the logic.

J0 : $q_2 * q_3$	$\neg K_0: 0$
J1 : $\overline{q_0} * q_3$	$\neg K_1: \overline{q_3} * (q_2 + q_4)$
J2 : $\overline{q_0} * q_3$	$\neg K_2: 0$
J3: $\overline{q_1} * q_4$	$\neg K_3: (q_0 + q_1) + \overline{q_4}$
J4: $\overline{q_1} + q_2$	$\neg K_4: q_0 + q_2$



Since I have number ordered the each AND gate and OR gate. My physical build will follow to these order numbers. From my analytical solution,  $\neg K_0$  and  $\neg K_2$  is always connected to the ground. I can connect those first. Next thing I can do first is to connect clock signal for each JK flip flop. From lab7, I know how to make a clock signal for the circuit. I will not discuss the detail how to do that. From the clock signal

picture, I use three jumper wires at 1Y. They will directly connect the clock signal to pin 4 for each JK flip flop. I use other jumper wires to connect that pin 4 clock signal to pin 12. That will make sure every JK flip flop will have the same clock signal. We should make sure the circuit I design is synchronous. After these steps, I can represent J0, J1, J2, J3, J4, !K1, !K3, !K4.

From the analytical solution, the logic for J0 is  $q_2 * q_3$ . That means that I need one AND gate. From the Multisim numbering, I will use number 4 AND gate. I connect one jumper wire at pin 2 (1J) from the first JK flip flop and pin 11 (4Y) in the AND gate. Regarding the inputs for the AND gate, I use one jumper connect pin 13 (4B) to pin 6 ( $q_2$ ). I also use one jumper wire connect pin 12 (4A) to pin 6 ( $q_3$ ). From these steps, the logic for J0 can be represented. For J1, the analytical solution shows that the logic for that is  $\overline{q_0} * q_3$ . Since for each JK flip flop, it has  $\overline{q}$  automatically, I do not need extra NOT gate to invert the output. From the Multisim solution, I need number 1 AND gate to do this. I use one jumper wire connect pin 7 ( $\overline{q_0}$ ) to pin 2 (1B). I use one jumper wire connect pin 6 ( $q_3$ ) to pin 1 (1A). Then, I use a jumper wire from pin 3 (1Y) to pin 2 (J1). That are steps how I make J1. For J2, the logic is also  $\overline{q_0} * q_3$ . So, the only thing I need to do is to use a jumper connect pin 3 (1Y) to pin 2 (J2). For J3, the logic for that is  $\overline{q_1} * q_4$ . In the Multisim solution, I use number 3 AND gate to do this logic. The building process is like the previous one. I use a jumper wire connect pin 9 for  $\overline{q_1}$  from first JK flip flop chip. It will go to pin 10 (3B) as one input for the AND gate. the other input is  $q_4$ . I use one jumper wire connect pin 6 ( $q_4$ ) from third JK flip flop to pin 9 (3A). Finally, I use one jumper connect 3Y from AND

gate to pin 14 (J3) from second JK chip. For !K1, the logic is a little complex. It is  $\overline{q3} * (q2 + q4)$ . In here, we need one AND gate and one or gate to do this. From here, I have noticed one question. My empty pin for q2 will not enough. The following logic will use q2 for couple time therefore I use one jumper wire connect q2 into other empty spots. As a result, I can get q2 as many as I want. To get  $(q2 + q4)$ . I choose use number 2 OR gate. I use one jumper wire connect q2 to pin 4 (2A) from OR gate and the other jumper wire connect q4 to pin 5 (2B) from OR gate. The output 2Y will be  $(q2 + q4)$ . Next, I use number 2 AND gate to represent  $\overline{q3} * (q2 + q4)$ . I find the position for  $\overline{q3}$  (pin 11) from second JK flip flop chip. Connect it to 2A (pin 4) from AND gate. Then, I will use jumper wire connect 2Y from OR gate to 2B (pin 5) from AND gate. 2Y output for AND gate should become !K1. For !K3, its logic is  $(q0 + q1) + \overline{q4}$ . I should use two OR gates to redo this logic. I choose to use number 3 and number 4 OR gate to do this. Number 3 OR gate will perform  $(q0 + q1)$ . I first find the position for q0 and q1 which are pin 6 and pin 10 from the first JK flip flop chip. I use two jumper wires connect them into 3A and 3B. Next, I will find the position for  $\overline{q4}$ . It is located at pin 7 from third JK flip flop. I will connect  $\overline{q4}$  to 4A as the input for number 4 OR gate. 3Y from number 3 OR gate will connect to 4B. The final output for 4Y is connected to pin 13 for the second JK flip flop. Finally, !K4 is simple. The logic is  $(q0 + q2)$ . I use the second OR gate's number 3 gate to represent this. I use two jumper wires to connect q0 and q2 to 3A and 3B, respectively. The final output 3Y is connected to pin 3 from the third JK flip flop. At this point, all logic has been connected. Here is the link for the final design result:

<https://youtu.be/PIcKK4Ny-T4>

the next video briefly explains my build and what does what, here is the link:

<https://youtu.be/0CuMCqiarU>

from the result video, the 7-segment display does not start at 4 at first but it starts at number 1. After 4 seconds, it starts at 4. The result can be concluded as follow:

4 → 0 → 0 → 1 → 7 → 9 → 1 → 6 → 4

Here is my student card picture:



It shows that my physical build is successful. That concludes my final design project.

## IV. Discussion

From analytical solution, my final design project should be:

$$4 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 7 \rightarrow 9 \rightarrow 1 \rightarrow 6 \rightarrow 4$$

Multisim solution show the result is:

$$4 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 7 \rightarrow 9 \rightarrow 1 \rightarrow 6 \rightarrow 4$$

From my physical build, the result is:

$$4 \rightarrow 0 \rightarrow 0 \rightarrow 1 \rightarrow 7 \rightarrow 9 \rightarrow 1 \rightarrow 6 \rightarrow 4$$

From here, I can conclude that analytical, Multisim and physical build solutions are consistent with each other.

## V. Reflection

From this final design project, I really enjoyed the process for building and design the logic circuit. They are really interesting. I am glad I can learn that from this course. I learned the excitation table for this lab. I knew the process to build a small project. I definitely will apply these knowledges to my engineering career. At last, I want to say thank you to all people who helped me before. I appreciate that a lot. That conclude this final design project and my 2E04 as well. Thank you all again.