Qais Abu El Haija , abuelhaq
400294443

Final Design Project - ENG PHYS 2E04

Qais Abu El Haija, abuelhaq - 400294443

Aim

The aim is to display student number 400294443 on the 7-segment display.

Analytical Method

The table below shows the 4-bit binary numbering for each of the digits in the student number 400294443.

Number	**Binary**
4	0100
0	0000
0	0000
2	0010
9	1001
4	0100
4	0100
4	0100
3	0011

Qais Abu El Haija , abuelhaq
400294443

- The table below shows the transition table for the FSM which contains 4 outputs with Q1 being the most important bit and with decreasing importance and Q5, and Q6 is the counter which is used for seeing if any repeating digits appear.

CURRENT STATE				REPEATED DIGIT MEMORY (before)		NEXT STATE				REPEATED DIGIT MEMORY (after)	
q1	**q2**	**q3**	**q4**	**q5 (counter)**	**q6 (counter)**	**Q1**	**Q2**	**Q3**	**Q4**	**Q5 (counter)**	**Q6 (counter)**
0	1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	1	0	0	1	0	X	X
0	0	1	0	X	X	1	0	0	1	X	X
1	0	0	1	X	X	0	1	0	0	0	1
0	1	0	0	0	1	0	1	0	0	1	0
0	1	0	0	1	0	0	1	0	0	1	1
0	1	0	0	1	1	0	0	1	1	X	X
0	0	1	1	X	X	0	1	0	0	0	0

- The below excitation table was used to find the inputs for each of the flip-flops.

q (current)	**J**	**!K**	**Q (next)**
0	0	X	0
0	1	X	1
1	X	0	0
1	X	1	1

The above Excitation table has the Xs, which indicates don't care cases. When q (current state) is 0, and we want the next state to stay at 0, we can either hold or reset. Holding the switch would keep the output at zero which is the desired output. Similarly, a reset would also do the same thing. Therefore, it doesn't matter what the value of !K is, both 0 and 1 will have the same effect.

Qais Abu El Haija , abuelhaq
400294443

- The before and after were used for the output Q1 from the transition table to figure out the inputs J1 and, !K1

q1 (current)	Q1 (Next)	J1	!K1
0	0	0	X
0	0	0	X
0	0	0	X
0	1	1	X
1	0	X	0
0	0	0	X
0	0	0	X
0	0	0	X
0	0	0	X

- The values of the input J1 are compared with the outputs of the 6 flip-flops q1, q2, q3, q4, q5, and q6.

q1	q2	q3	q4	q5 (counter)	q6 (counter)	J1
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	1	0	X	X	1
1	0	0	1	X	X	X
0	1	0	0	0	1	0
0	1	0	0	1	0	0
0	1	0	0	1	1	0
0	0	1	1	X	X	0

Qais Abu El Haija , abuelhaq
400294443

- The K-mapping for J1

q1q2q3 / q4q5q6	**000**	**001**	**011**	**010**	**100**	**101**	**111**	**110**
000	0	0	X	X	X	X	X	X
001	1	1	1	1	0	0	0	0
011	X	X	X	X	X	X	X	X
010	0	0	0	0	X	X	X	X
100	X	X	X	X	X	X	X	X
101	X	X	X	X	X	X	X	X
111	X	X	X	X	X	X	X	X
110	X	X	X	X	X	X	X	X

The Kmap for the input J1 is done using SOP and the boolean expression obtained is:

J1 = (Q3 * !Q4)

- The same thing is also done for the other flip-flop input for the first flip flop !K1.

q1	**q2**	**q3**	**q4**	**q5 (counter)**	**q6 (counter)**	**!K1**
0	1	0	0	0	0	X
0	0	0	0	0	0	X
0	0	0	0	0	1	X
0	0	1	0	X	X	X
1	0	0	1	X	X	0
0	1	0	0	0	1	X
0	1	0	0	1	0	X
0	1	0	0	1	1	X
0	0	1	1	X	X	X

Qais Abu El Haija , abuelhaq
400294443

- The K-mapping for !K1

q1q2q3 / q4q5q6	**000**	**001**	**011**	**010**	**100**	**101**	**111**	**110**
000	X	X	X	X	X	X	X	X
001	X	X	X	X	X	X	X	X
011	X	X	X	X	X	X	X	X
010	X	X	X	X	X	X	X	X
100	X	X	X	X	0	0	0	0
101	X	X	X	X	X	X	X	X
111	X	X	X	X	X	X	X	X
110	X	X	X	X	X	X	X	X

The Boolean expression acquired for the input !K1 using POS is:

!K1 = 0

- The before and after were used for the output Q2 from the transition table to figure out the inputs J2 and !K2

q2 (current)	**Q2 (next)**	**J2**	**!K2**
1	0	X	0
0	0	0	X
0	0	0	X
0	0	0	X
0	1	1	X
1	1	X	1
1	1	X	1
1	0	X	0
0	1	1	X

Qais Abu El Haija , abuelhaq
400294443

- The values of the input J2 are compared with the outputs of the 6 flip-flops q1, q2, q3, q4, q5, and q6.

q1	**q2**	**q3**	**q4**	**q5 (counter)**	**q6 (counter)**	**J2**
0	1	0	0	0	0	X
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	1	0	X	X	0
1	0	0	1	X	X	1
0	1	0	0	0	1	X
0	1	0	0	1	0	X
0	1	0	0	1	1	X
0	0	1	1	X	X	1

- The K-mapping of J2

q1q2q3 / q4q5q6	**000**	**001**	**011**	**010**	**100**	**101**	**111**	**110**
000	0	0	X	X	X	X	X	X
001	0	0	0	0	1	1	1	1
011	X	X	X	X	X	X	X	X
010	X	X	X	X	X	X	X	X
100	X	X	X	X	1	1	1	1
101	X	X	X	X	X	X	X	X
111	X	X	X	X	X	X	X	X
110	X	X	X	X	X	X	X	X

- The K-map for the input J2 is done using SOP and the Boolean expression obtained is

J2 = Q4

Qais Abu El Haija , abuelhaq
400294443

- The same thing is also done for the other flip-flop input for the first flip flop !K2.

q1	q2	q3	q4	q5 (counter)	q6 (counter)	!K2
0	1	0	0	0	0	0
0	0	0	0	0	0	X
0	0	0	0	0	1	X
0	0	1	0	X	X	X
1	0	0	1	X	X	X
0	1	0	0	0	1	1
0	1	0	0	1	0	1
0	1	0	0	1	1	0
0	0	1	1	X	X	X

- The K-mapping for !K2,

q1q2q3 / q4q5q6	000	001	011	010	100	101	111	110
000	X	X	X	X	X	X	X	X
001	X	X	X	X	X	X	X	X
011	X	X	X	X	X	X	X	X
010	0	1	0	1	X	X	X	X
100	X	X	X	X	X	X	X	X
101	X	X	X	X	X	X	X	X
111	X	X	X	X	X	X	X	X
110	X	X	X	X	X	X	X	X

The k-mapping for the input !K2 is done using both the SOP and POS and the Boolean expression obtained is,

!K2 = (!Q5 * Q6) + (Q5 * !Q6)

Qais Abu El Haija , abuelhaq
400294443

- The before and after outputs Q3 of the third flip flop are compared to attain the inputs for the flip flop.

q3 (current)	**Q3 (next)**	**J3**	**!K3**
0	0	0	X
0	0	0	X
0	1	1	X
1	0	X	0
0	0	0	X
0	0	0	X
0	0	0	X
0	1	1	X
1	0	X	0

- The values of the input J3 are compared with the outputs of the 6 flip-flops q1, q2, q3, q4, q5, and q6.

q1	**q2**	**q3**	**q4**	**q5 (counter)**	**q6 (counter)**	**J3**
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	1	1
0	0	1	0	X	X	X
1	0	0	1	X	X	0
0	1	0	0	0	1	0
0	1	0	0	1	0	0
0	1	0	0	1	1	1
0	0	1	1	X	X	X

Qais Abu El Haija , abuelhaq
400294443

- The K-mapping for J3

q1q2q3 / q4q5q6	**000**	**001**	**011**	**010**		**100**	**101**	**111**	**110**
000	0	1	X	X		X	X	X	X
001	X	X	X	X		X	X	X	X
011	X	X	X	X		X	X	X	X
010	0	0	1	0		X	X	X	X
100	X	X	X	X		0	0	0	0
101	X	X	X	X		X	X	X	X
111	X	X	X	X		X	X	X	X
110	X	X	X	X		X	X	X	X

- The Kmap for the input J3 is done using the POS and the Boolean expression obtained using POS is

J3 = (Q6 * !Q4) * (!Q2 + Q5)

- The same thing is also done for the other flip-flop input for the first flip flop !K3.

q1	**q2**	**q3**	**q4**	**q5 (counter)**	**q6 (counter)**	**!K3**
0	1	0	0	0	0	X
0	0	0	0	0	0	X
0	0	0	0	0	1	X
0	0	1	0	X	X	0
1	0	0	1	X	X	X
0	1	0	0	0	1	X
0	1	0	0	1	0	X
0	1	0	0	1	1	X
0	0	1	1	X	X	0

Qais Abu El Haija , abuelhaq
400294443

- The K-map for !K3

q1q2q3 / q4q5q6	**000**	**001**	**011**	**010**		**100**	**101**	**111**	**110**
000	X	X	X	X		X	X	X	X
001	0	0	0	0		0	0	0	0
011	X	X	X	X		X	X	X	X
010	X	X	X	X		X	X	X	X
100	X	X	X	X		X	X	X	X
101	X	X	X	X		X	X	X	X
111	X	X	X	X		X	X	X	X
110	X	X	X	X		X	X	X	X

The Kamp for the input !K3 is done using both the SOP and POS and the Boolean expression obtained is

!K3 = 0

- The before and after outputs Q4 of the counter flip flop are compared to attain the inputs for the flip flop.

q4 (current)	**Q4 (next)**	**J4**	**!K4**
0	0	0	X
0	0	0	X
0	0	0	X
0	1	1	X
1	0	X	0
0	0	0	X
0	0	0	X
0	1	1	X
1	0	X	0

Qais Abu El Haija , abuelhaq
400294443

- The values of the input J4 are compared with the outputs of the 6 flip-flops q1, q2, q3, q4, q5, and q6.

q1	**q2**	**q3**	**q4**	**q5 (counter)**	**q6 (counter)**	**J4**
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	1	0	X	X	1
1	0	0	1	X	X	X
0	1	0	0	0	1	0
0	1	0	0	1	0	0
0	1	0	0	1	1	1
0	0	1	1	X	X	X

- The K-mapping for J4

q1q2q3 / q4q5q6	**000**	**001**	**011**	**010**	**100**	**101**	**111**	**110**
000	0	0	X	X	X	X	X	X
001	1	1	1	1	X	X	X	X
011	X	X	X	X	X	X	X	X
010	0	0	1	0	X	X	X	X
100	X	X	X	X	X	X	X	X
101	X	X	X	X	X	X	X	X
111	X	X	X	X	X	X	X	X
110	X	X	X	X	X	X	X	X

The K-map for the input J4 is done using the SOP and the Boolean expression obtained is

J4 = (Q3) + (Q5*Q6)

Qais Abu El Haija , abuelhaq
400294443

- The same thing is also done for the other flip-flop input for the first flip flop !K4.

q1	**q2**	**q3**	**q4**	**q5 (counter)**	**q6 (counter)**	**!K4**
0	1	0	0	0	0	X
0	0	0	0	0	0	X
0	0	0	0	0	1	X
0	0	1	0	X	X	X
1	0	0	1	X	X	0
0	1	0	0	0	1	X
0	1	0	0	1	0	X
0	1	0	0	1	1	X
0	0	1	1	X	X	0

As all the inputs are either don’t care or 0 the value for the input !K4 can be made equal to 0.

Therefore, !K4 = 0

Qais Abu El Haija , abuelhaq
400294443

- The before and after outputs Q5 of the counter flip flop are compared to attain the inputs for the flip flop.

q5 (current)	**Q5 (next)**	**J5**	**!K5**
0	0	0	X
0	0	0	X
0	X	X	X
X	X	X	X
X	0	0	0
0	1	1	X
1	1	X	1
1	X	X	X
X	0	0	0

- The values of the input J5 are compared with the outputs of the 6 flip-flops q1, q2, q3, q4, q5, and q6.

q1	**q2**	**q3**	**q4**	**q5 (counter)**	**q6 (counter)**	**J5**
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	1	X
0	0	1	0	X	X	X
1	0	0	1	X	X	0
0	1	0	0	0	1	1
0	1	0	0	1	0	X
0	1	0	0	1	1	X
0	0	1	1	X	X	0

Qais Abu El Haija , abuelhaq
400294443

- The K-mapping for J5

q1q2q3 / q4q5q6	000	001	011	010		100	101	111	110
000	0	X	X	X		X	X	X	X
001	X	X	X	X		0	0	0	0
011	X	X	X	X		X	X	X	X
010	0	1	X	X		X	X	X	X
100	X	X	X	X		0	0	0	0
101	X	X	X	X		X	X	X	X
111	X	X	X	X		X	X	X	X
110	X	X	X	X		X	X	X	X

The K-map for the input J5 is done using the SOP and the Boolean expression obtained is

J5 = (!Q4 * Q6)

- The same thing is also done for the other flip-flop input for the first flip flop !K5.

q1	q2	q3	q4	q5 (counter)	q6 (counter)	!K5
0	1	0	0	0	0	X
0	0	0	0	0	0	X
0	0	0	0	0	1	X
0	0	1	0	X	X	X
1	0	0	1	X	X	0
0	1	0	0	0	1	X
0	1	0	0	1	0	1
0	1	0	0	1	1	X
0	0	1	1	X	X	0

Qais Abu El Haija , abuelhaq
400294443

- The K-mapping for !K5

q1q2q3 / q4q5q6	**000**	**001**	**011**	**010**	**100**	**101**	**111**	**110**
000	X	X	X	X	X	X	X	X
001	X	X	X	X	0	0	0	0
011	X	X	X	X	X	X	X	X
010	X	X	X	1	X	X	X	X
100	X	X	X	X	0	0	0	0
101	X	X	X	X	X	X	X	X
111	X	X	X	X	X	X	X	X
110	X	X	X	X	X	X	X	X

The K-map for the input !K5 is done using the SOP and the Boolean expression obtained is

!K5 = !Q4

- The before and after outputs Q6 of the counter flip flop are compared to attain the inputs for the flip flop.

q6 (current)	**Q6 (next)**	**J6**	**!K6**
0	0	0	X
0	1	1	X
1	X	X	X
X	X	X	X
X	1	1	1
1	0	X	0
0	1	1	X
1	X	X	X
X	0	0	0

Qais Abu El Haija , abuelhaq
400294443

- The values of the input J6 are compared with the outputs of the 6 flip-flops q1, q2, q3, q4, q5, and q6.

q1	q2	q3	q4	q5 (counter)	q6 (counter)	J6
0	1	0	0	0	0	0
0	0	0	0	0	0	1
0	0	0	0	0	1	X
0	0	1	0	X	X	X
1	0	0	1	X	X	1
0	1	0	0	0	1	X
0	1	0	0	1	0	1
0	1	0	0	1	1	X
0	0	1	1	X	X	0

- The K-mapping for J6

q1q2q3 / q4q5q6	**000**	**001**	**011**	**010**	**100**	**101**	**111**	**110**
000	1	X	X	X	X	X	X	X
001	X	X	X	X	0	0	0	0
011	X	X	X	X	X	X	X	X
010	0	X	X	1	X	X	X	X
100	X	X	X	X	1	1	1	1
101	X	X	X	X	X	X	X	X
111	X	X	X	X	X	X	X	X
110	X	X	X	X	X	X	X	X

The K-map for the input J6 is done using the SOP and the Boolean expression obtained is

J6 = (!Q2 * !Q3) + (!Q3 * Q5)

Qais Abu El Haija , abuelhaq
400294443

- The same thing is also done for the other flip-flop input for the first flip flop !K6.

q1	q2	q3	q4	q5 (counter)	q6 (counter)	!K6
0	1	0	0	0	0	X
0	0	0	0	0	0	X
0	0	0	0	0	1	X
0	0	1	0	X	X	X
1	0	0	1	X	X	1
0	1	0	0	0	1	0
0	1	0	0	1	0	X
0	1	0	0	1	1	X
0	0	1	1	X	X	0

- The K-mapping for !K6

q1q2q3 / q4q5q6	000	001	011	010		100	101	111	110
000	X	X	X	X		X	X	X	X
001	X	X	X	X		0	0	0	0
011	X	X	X	X		X	X	X	X
010	X	0	X	X		X	X	X	X
100	X	X	X	X		1	1	1	1
101	X	X	X	X		X	X	X	X
111	X	X	X	X		X	X	X	X
110	X	X	X	X		X	X	X	X

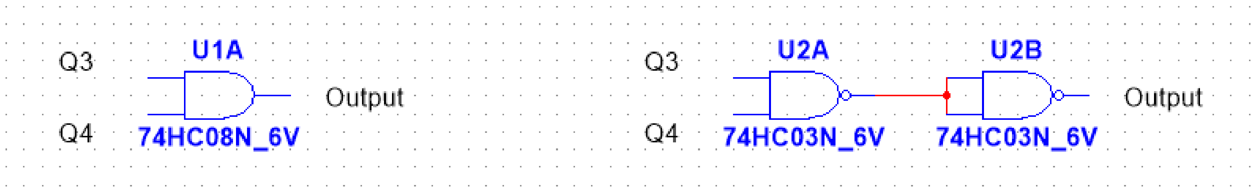
The K-map for the input !K6 is done using the SOP and the Boolean expression obtained is

!K6 = Q1

Qais Abu El Haija , abuelhaq
400294443

- An overall table for all the logic can be found down below. We can deduce the number of possible NAND gates that can be used using De Morgan's theorem.

For instance, J1:



Input	Logic	Number of Gates	Number of NAND
J1	(Q3 * !Q4)	1	**2**
!K1	0	0	0
J2	Q4	0	0
!K2	(!Q5 * Q6) + (Q5 * !Q6)	3	7
J3	(Q6 * !Q4) * (!Q2 + Q5)	3	7
!K3	0	0	0
J4	Q3 + (Q5 * Q6)	2	5
!K4	0	0	0
J5	(!Q4 * Q6)	1	2
!K5	!Q4	0	0
J6	(!Q2 * !Q3) + (!Q3 * Q5)	3	7
!K6	Q1	0	0
TOTAL		**13**	**30**

Qais Abu El Haija , abuelhaq
400294443

- Some of the optimizations done for ensuring using a smaller number of gates and flip-flops are:

1) Doing both POS and SOP for all K-maps ensures that the most efficient Boolean expression is acquired which is also simplified using Boolean algebra laws.

   This can be seen in J3, where if SOP was used to obtain the Boolean expression a total of 5 gates would be used, whereas if POS was used only 3 gates would be needed. Therefore, you can see why using POS benefited and simplified the circuit.

   SOP: J3 = (!Q1 * !Q2 * Q6) + (!Q1 * Q5 + Q6), (Total = 5 gates. 3 AND, and 2 OR)
   POS: J3 = (Q6 * !Q4) * (!Q2 + Q5), (Total = 3 gates. 2 AND, and 1 OR)

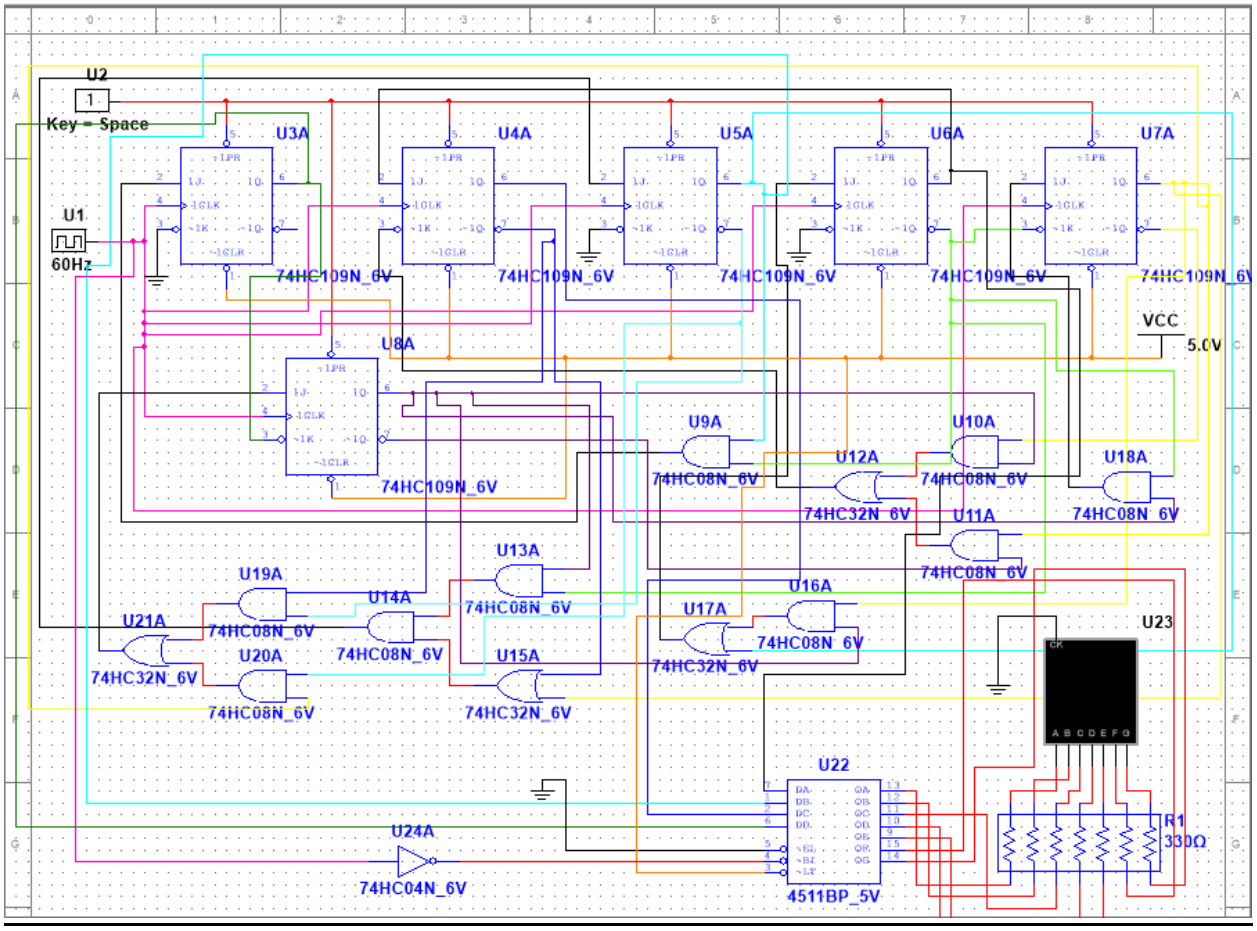
2) Not using NAND gates since the resulting number of gates would be almost twice the number of gates used. (Shown in the table above)

- Reasons to follow my build strategy:

1) Since I have the number 4 repeated 4 times in the student number, I had to have 2 counters to input, to make every 4 unique and different. Therefore, that leads to having 6 flip-flops.
2) Also, since most people don’t have a number exceeding 8 in their student number, they were prone to ground Q1, however, that was not the case for me, since I had the ‘9’ in the student number.

Qais Abu El Haija , abuelhaq
400294443

Multisim Method



The circuit is set up in Multisim as shown above, the circuit consists of 6 JK Flip-flop which has a code of 74HC109N. Each Flip-flop has two individual inputs and a third for the clock; the clock input is provided by a digital clock which is common for all six flip-flops. The inputs for each flip flop are varied and are directly or indirectly dependent on the output of the previous JK flip-flop. The output for the circuit is a 7-segment display that is connected to a decoder which is responsible for converting the binary signal into a signal that was accepted by the 7-segment display. The circuit also consists of 9 AND, 4 OR, and 1 NOT gates which are connected in the logic shown above.

Qais Abu El Haija , abuelhaq
400294443

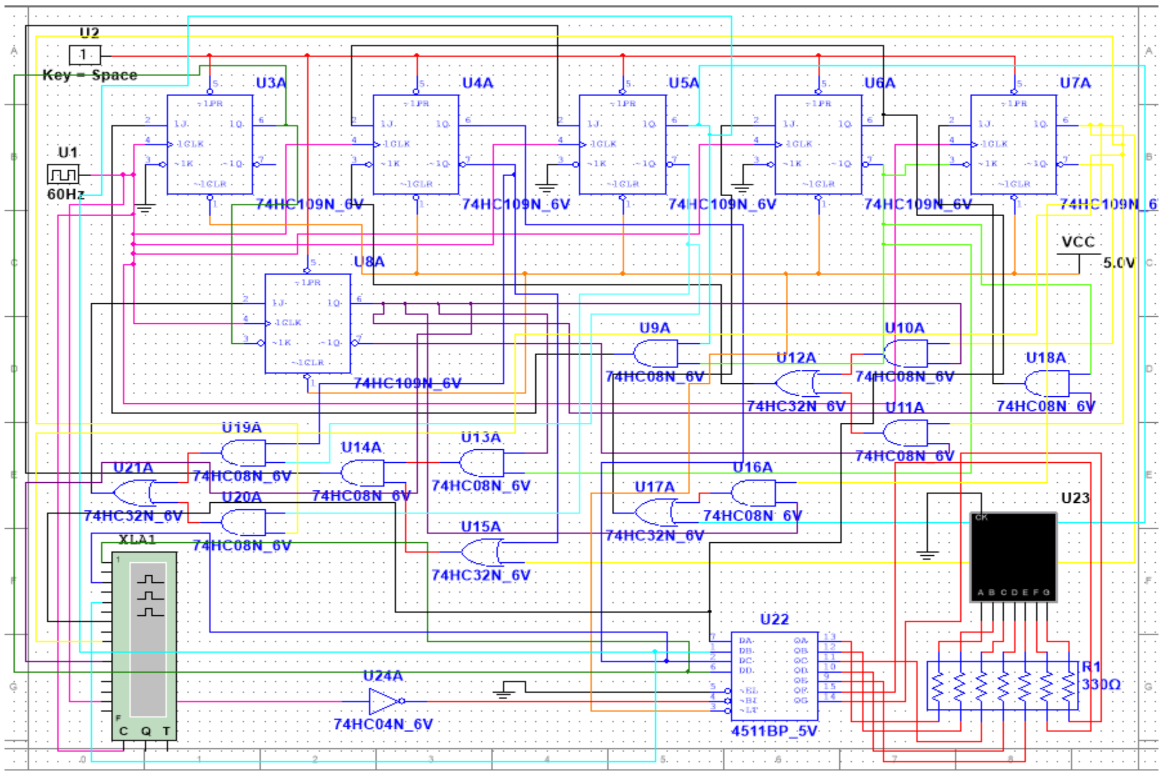
- The color code is following

Input	Color
Q1	🟩
Q2	🟦
Q3	🟦
Q4	🟩
Q5	🟨
Q6	🟪
VCC	🟧
J/K	⬛
Clock	🟪

- The demonstration is shown in the video shown below:
https://drive.google.com/file/d/1i8IsilBkD7ausCmwII8dITbeoyU6ZLmg/view?usp=sharing

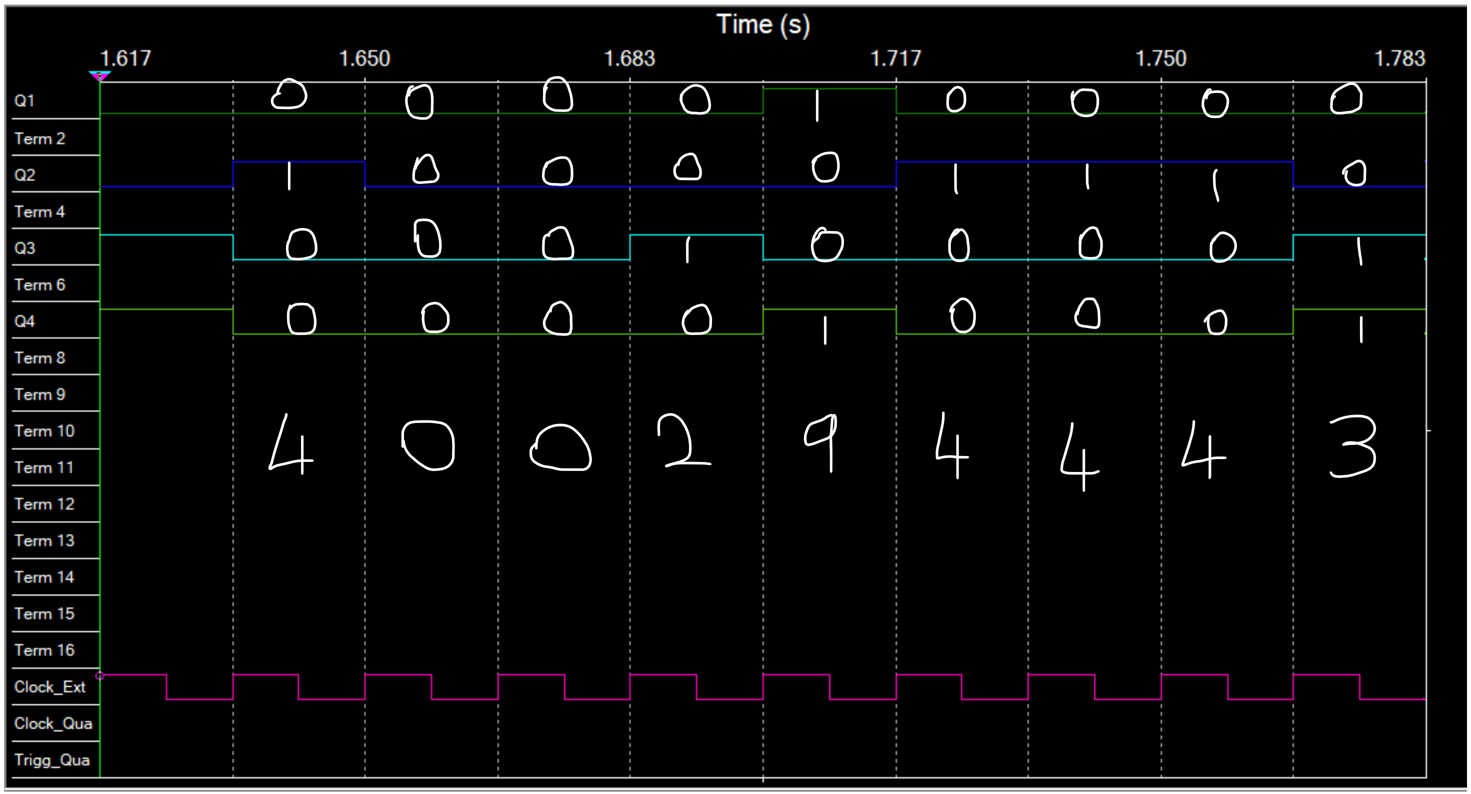
Qais Abu El Haija , abuelhaq
400294443

- The timing diagrams for the outputs can be obtained using the logic analyzer function in Multisim as shown in the picture below.



Qais Abu El Haija , abuelhaq
400294443

- The below figure is the **timing diagram** obtained after setting up the Logic Analyzer in the circuit as shown previously.



As we can see in the figure the diagram follows this 4-bit pattern,

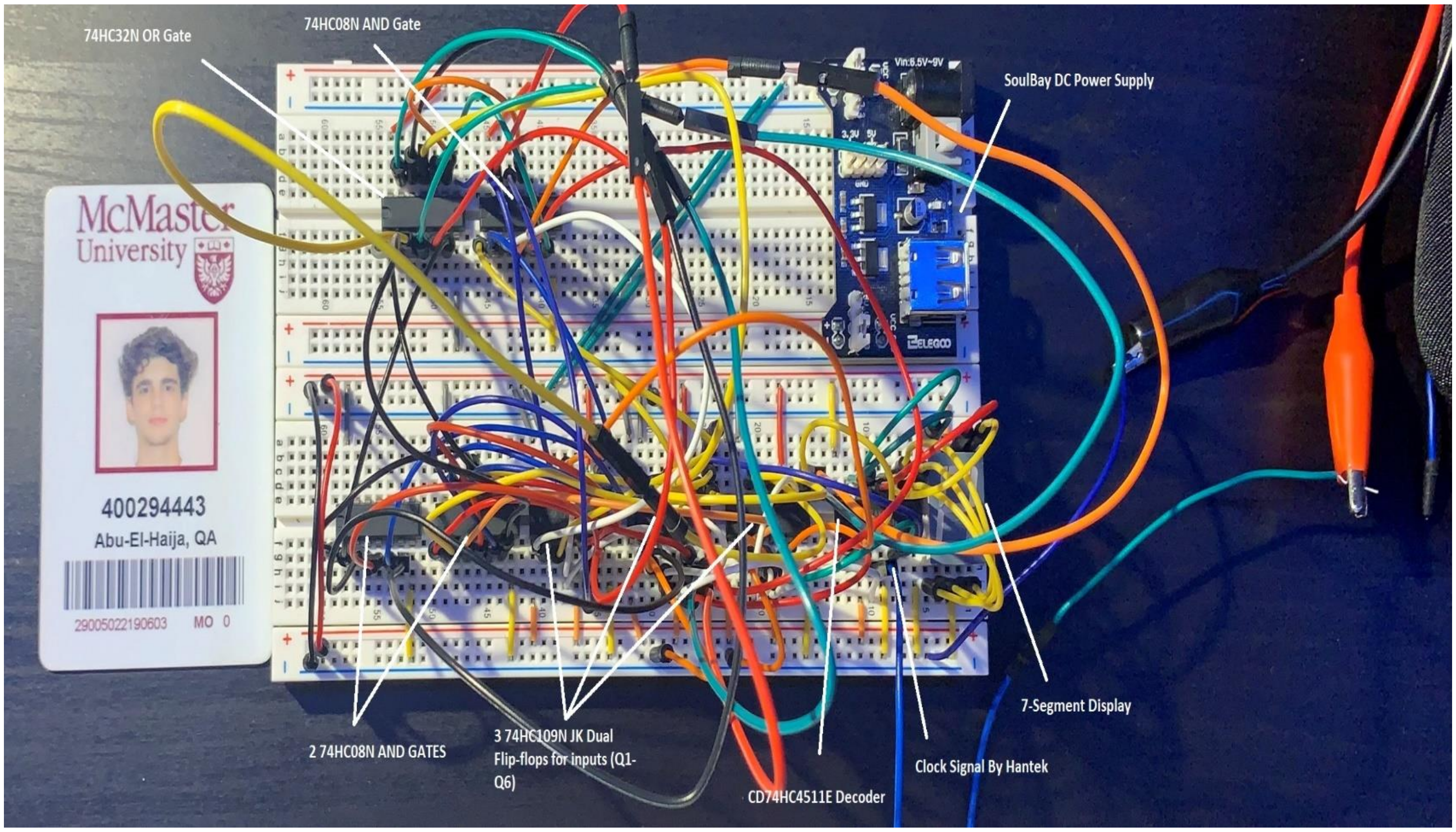
0100 -> 0000 -> 0000 -> 0010 -> 1001 -> 0100 -> 0100 -> 0100 -> 0011.

This 4-bit corresponds to the following student number, 40029443. This means that the timing diagram produced is correct, and both the analytical and Multism methods are done correctly.

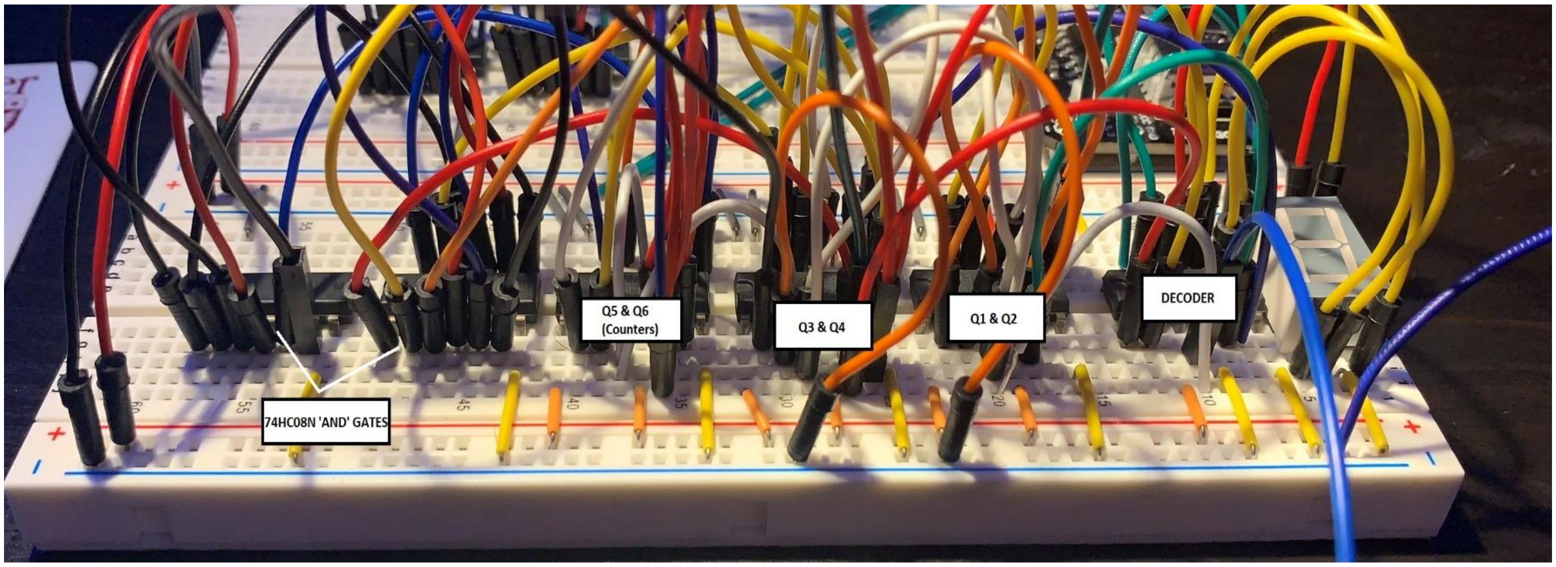
Qais Abu El Haija , abuelhaq
400294443

Experimental Method

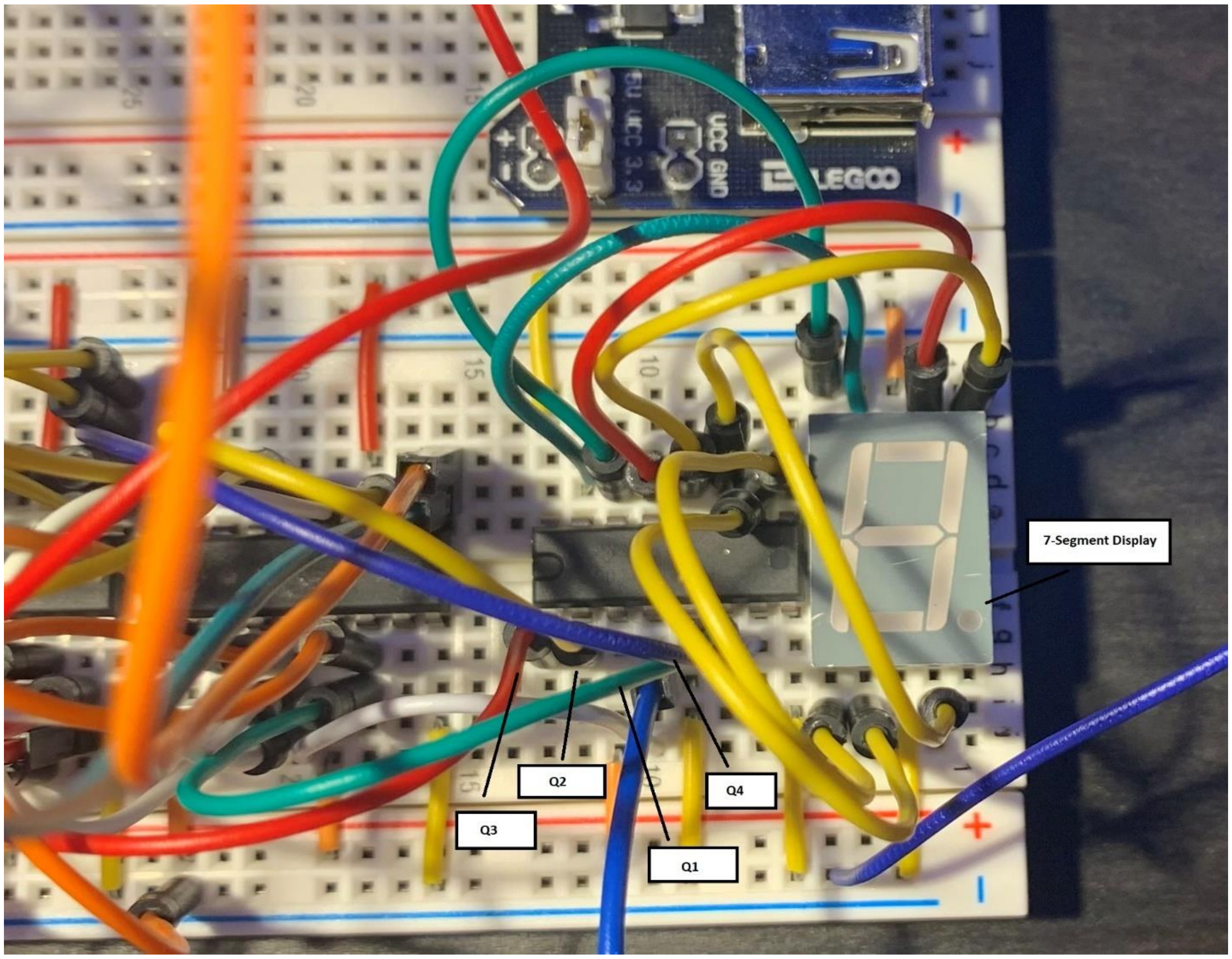
The circuit was set up using the Home-Kit as shown below

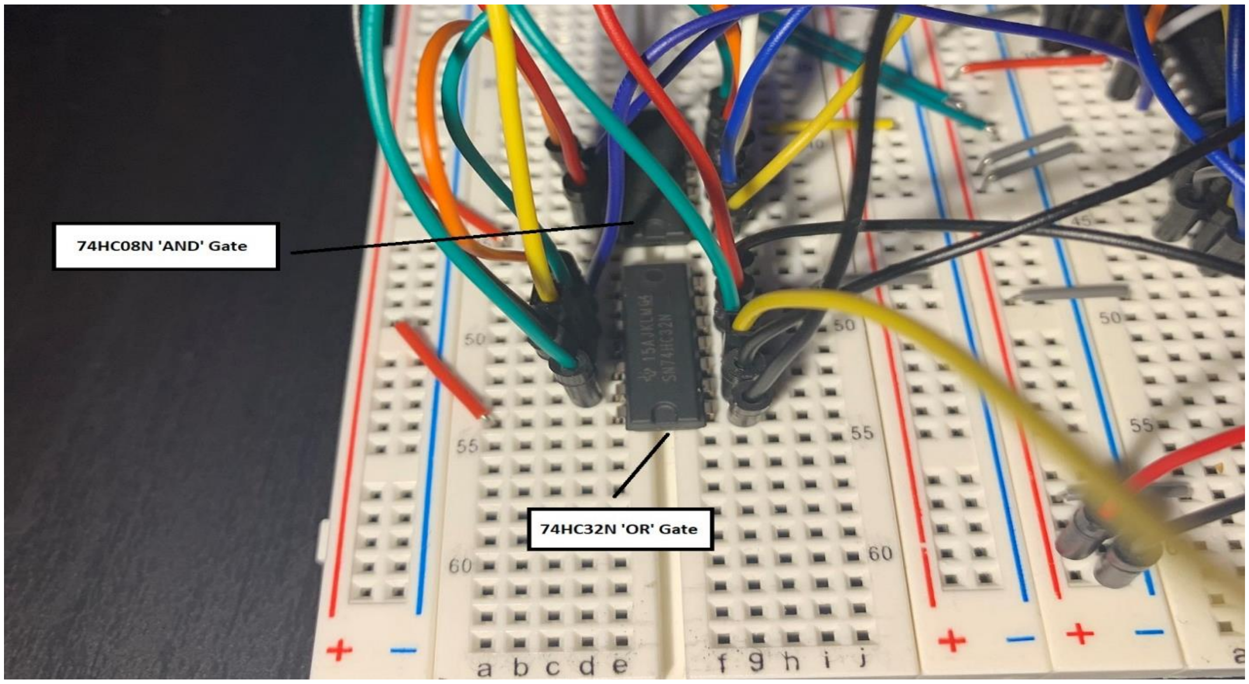


- Looking at the circuit from different angles to help understand the setup:



Qais Abu El Haija , abuelhaq
400294443





Qais Abu El Haija , abuelhaq
400294443

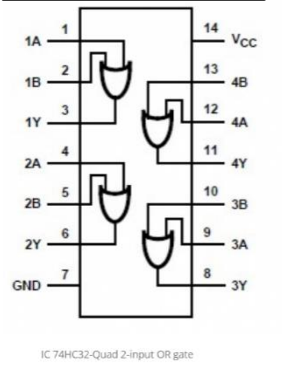
As it's known, the main aim of the project is to represent or output the following student number, 400294443, using the 7-segment display.

Trying to break down the modified circuit into its components. The circuit uses the Soul-bay power supply and the Hantek as the input. The Soul-bay power supply is used to provide the input voltage to the breadboard which is set at 5V on each side. The Hantek is used to provide the Clock signal to the decoder. It sets it as a square wave with a frequency of 1 Hz, an amplitude of 1V, and an offset of 1.5 V. The circuit consists of 7 IC, the first three is the 74HC109N JK Dual Flip-Flop, which consists of J and !K inputs and outputs of Q1-Q6. Also, the circuit had 3 74HC08N 'AND' quad IC's as well as 1 74HC32N OR quad IC. The colors code was corresponding with the outputs. The green wires were for Q1. The yellow wires were for Q2. The red wires were for Q3, and finally, the blue wires were for Q4. Due to a shortage of jumper wires, some other colors were used to complete the circuit. Those wires were set up according to the Boolean expressions that were found from the SOP's and POS's done during the K-mapping in the analytical method.

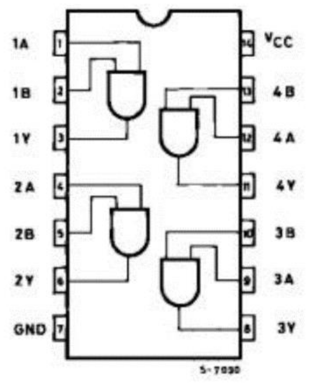
Input	Color
Q1	🟩
Q2	🟨
Q3	🟥
Q4	🟦

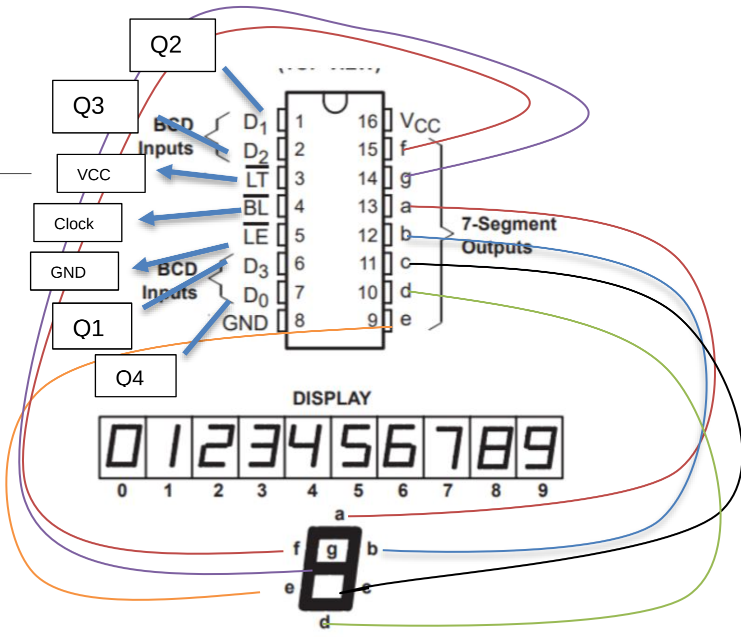
To assure full understanding of the build, diagrams of the inputs/outputs of some ICs are shown below,

74HC32N
'OR' quad IC

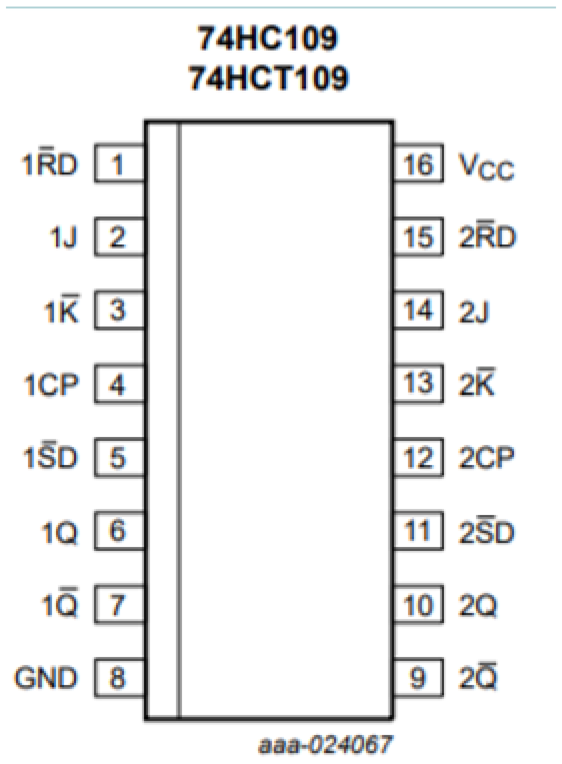


74HC08N
'AND' quad IC





Qais Abu El Haija , abuelhaq
400294443



Usually VCC, (!S)D, CP, (!R)D are set to high
GND is set to Low (Ground)

- More about the circuit is explained in the video shown below:

https://drive.google.com/file/d/1beHkz1r9HBvvDeiv86J213JuG3hoDFkN/view?usp=sharing

- The following link would direct you to the simulation of the experimental method, and the output via the 7-segment display:

https://drive.google.com/file/d/1sptHRuv9756CxvI6b73i_bZvhsS-qPSm/view?usp=sharing

- LAB VIDEO:

https://drive.google.com/file/d/1tk1p35LkGZMzZq64ji03UWySXjwbPSv6/view?usp=sharing

for some reason the video was cut in the ending, it was basically showing the output of the experimental method, to see that clearly, the link above this link shows the full output.

Thanks.

Qais Abu El Haija , abuelhaq
400294443