2E04: Design Project

sur21

December 13th, 2024

Contents

1 Analytical Solution

1.1 Student Number Binary Conversion

My student number is 400507973. This will give binary conversion of:

Digit	Current State ($Q_1Q_2Q_3Q_4$)	Counters (C_1C_2)	Don't Care
4	0100	00	*
0	0000	01	-
0	0000	10	-
5	0101	11	*
0	0000	00	-
7	0111	01	-
9	1001	10	*
7	0111	11	-
3	0011	00	*

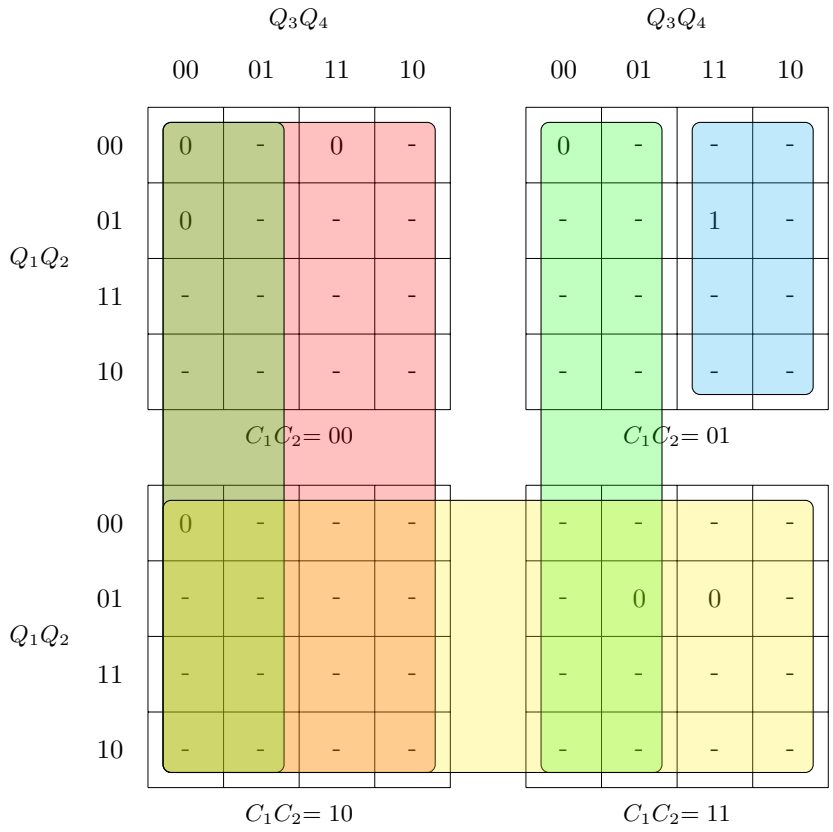
For the don't care unique digits, its counter states can be chosen to be anything. For the repeating digits, 0 and 7, its counter states can be arranged in a way such that the counters form a loop a simple loop, making the logic easier to implement and K-Map.

1.2 Excitation Table

#	Q_1	Q_2	Q_3	Q_4	C_1	C_2	$J1$	$!K1$	$J2$	$!K2$	$J3$	$!K3$	$J4$	$!K4$	J_{C_1}	$!K_{C_1}$	J_{C_2}	$!K_{C_2}$
4	0	1	0	0	0	0	0	X	X	0	0	X	0	X	0	X	1	X
0	0	0	0	0	0	1	0	X	0	X	0	X	0	X	1	X	X	0
0	0	0	0	0	1	0	0	X	1	X	0	X	1	X	X	1	1	X
5	0	1	0	1	1	1	0	X	X	0	0	X	X	0	X	0	X	0
0	0	0	0	0	0	0	0	X	1	X	1	X	1	X	0	X	1	X
7	0	1	1	1	0	1	1	X	X	0	X	0	X	1	1	X	X	0
9	1	0	0	1	1	0	X	0	1	X	1	X	X	1	X	1	1	X
7	0	1	1	1	1	1	0	X	X	0	X	1	X	1	X	0	X	0
3	0	0	1	1	0	0	0	X	1	X	X	0	X	0	0	X	0	X

The above table shows the binary conversion for each Q and the next inputs J and $!K$ required to create the proceeding number.

1.3 K-Map for J1



1.3.1 Explanation of K-Map Coloring for J1

The POS terms are coloured in red, green, and yellow. The SOP terms are covered in the blue implicant.

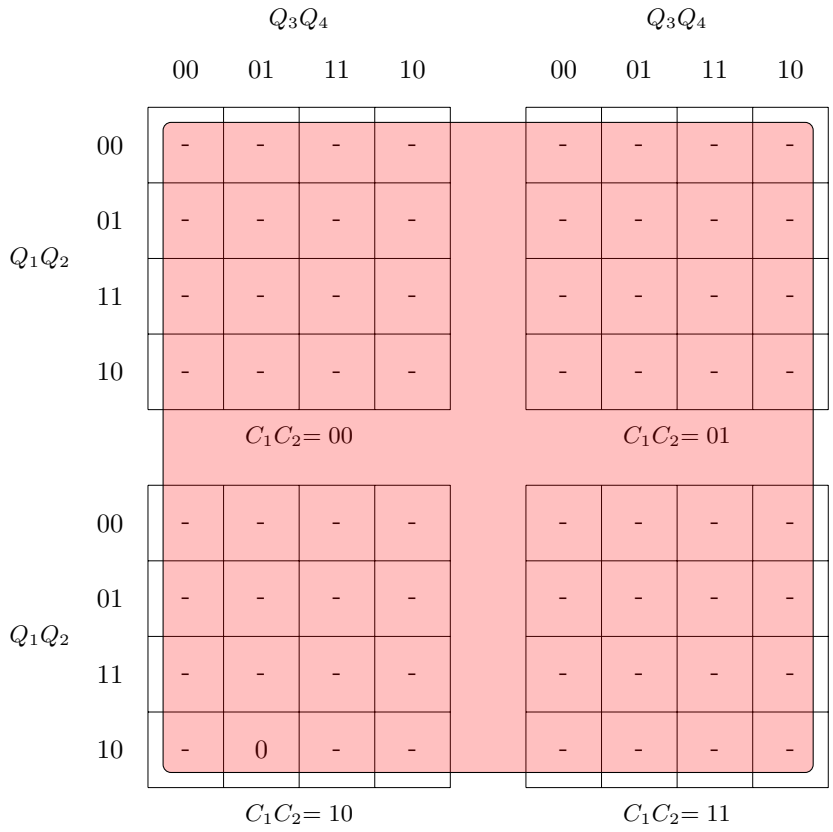
1.3.2 SOP Implementation for J1

The SOP implementation for this K-Map will be $Q_3 \cdot \overline{C_1} \cdot C_2$. This expression covers the blue implicant.

1.3.3 POS Implementation for J1

The POS implementation will be $\overline{Q_4 + \overline{C_1} + Q_3}$ which can also be expressed as $Q_3 \cdot \overline{C_1} \cdot C_2$. The expression Q_4 is represented by the red box, the expression $\overline{C_1}$ is represented by the green implicant, and the expression Q_3 is represented by the yellow box.

1.4 K-Map for !K1



1.4.1 Explanation of K-Map Coloring for !K1

The entire K-Map is covered in red. As there are no minterms, and all other terms are maxterms or indeterminants, the expression is simply 0.

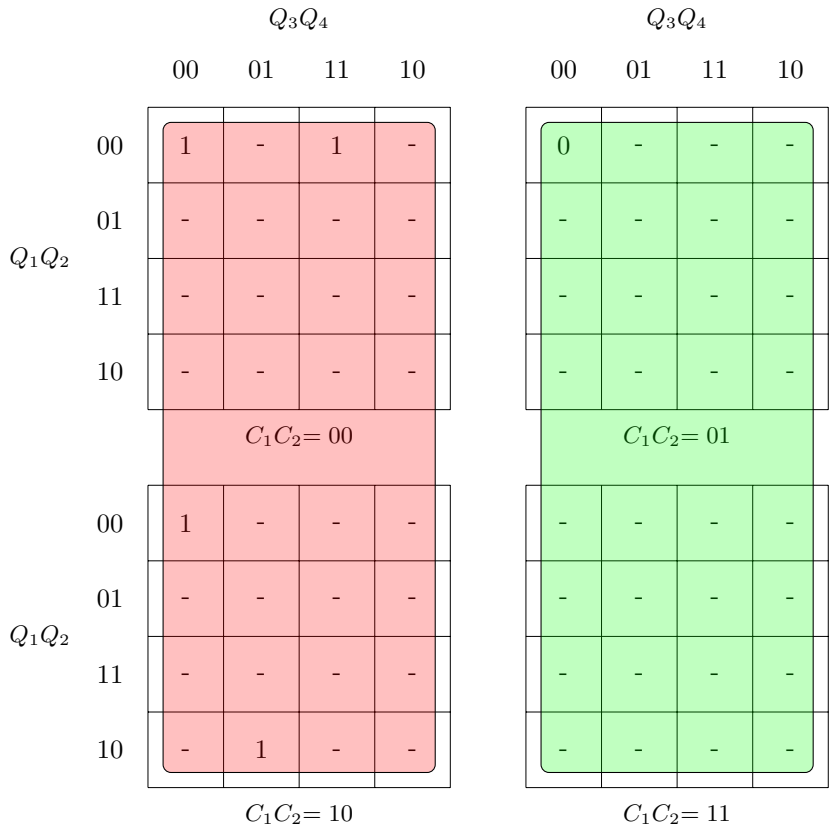
1.4.2 SOP Implementation for !K1

The SOP implementation for this K-Map is 0

1.4.3 POS Implementation for !K1

The POS implementation for this is 0

1.5 K-Map for J2



1.5.1 Explanation of K-Map Coloring for J2

The red implicant represents SOP and the green implicant represents POS.

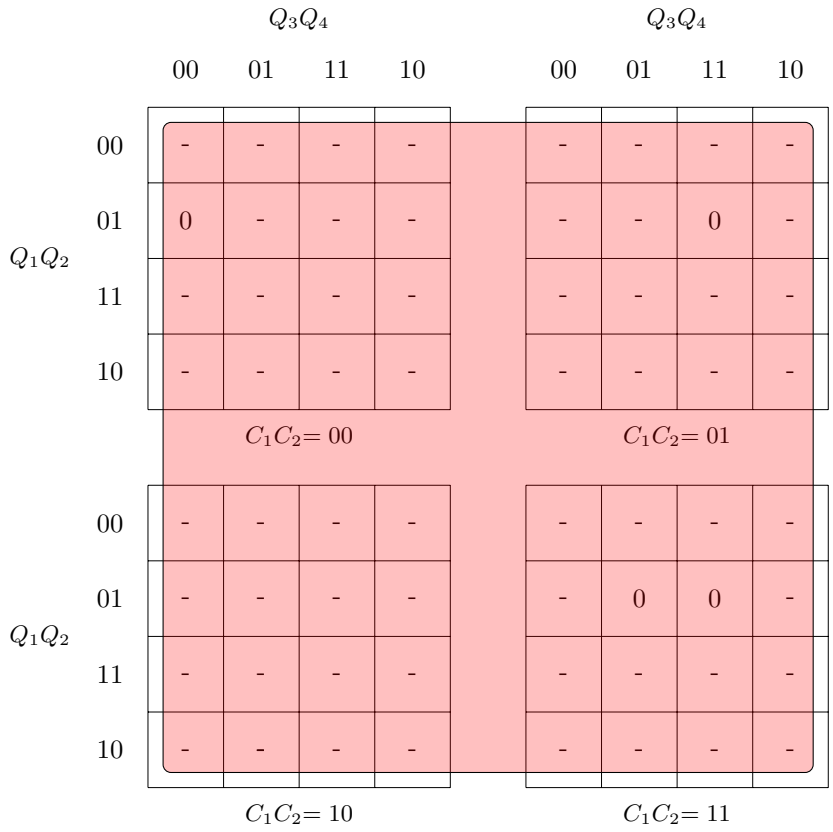
1.5.2 SOP Implementation for J2

The SOP implementation for this K-Map is $\overline{C_2}$

1.5.3 POS Implementation for J2

The POS implementation is $\overline{C_2}$

1.6 K-Map for !K2



1.6.1 Explanation of K-Map Coloring for !K2

The entire K-Map is covered in red, as there are only minterms and indeterminants, the expression for is simply 0.

1.6.2 SOP Implementation for !K2

The SOP expression is 0.

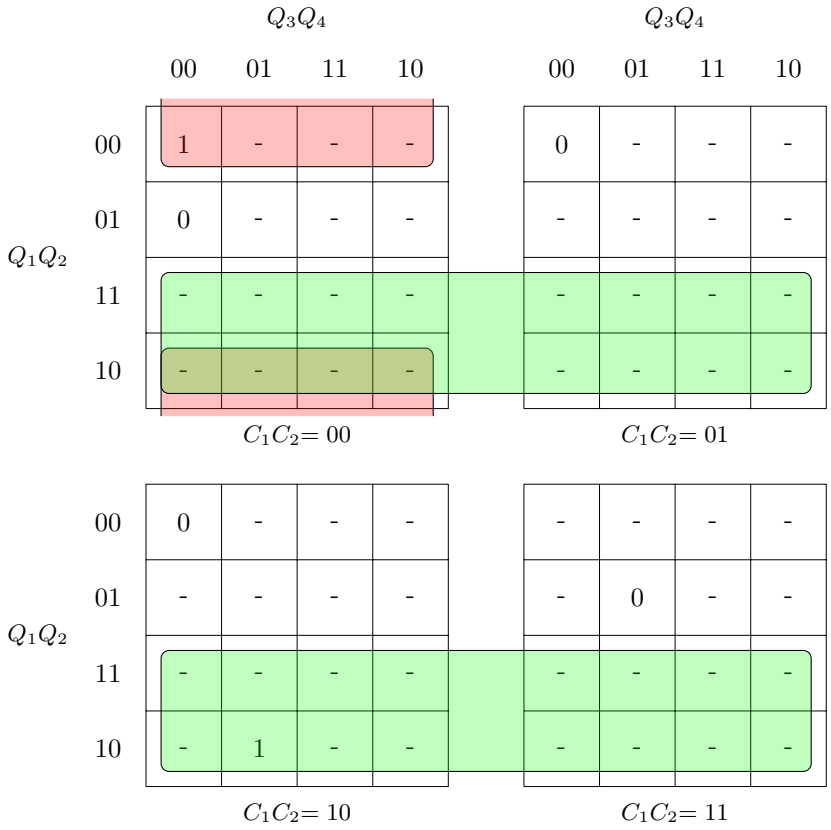
1.6.3 POS Implementation for !K2

The POS expression is 0.

1.7 K-Map for J3

Due to the many overlaps in the J3 K-Map, the SOP and POS implicants will be drawn on seperate K-Maps to reduce visual clutter.

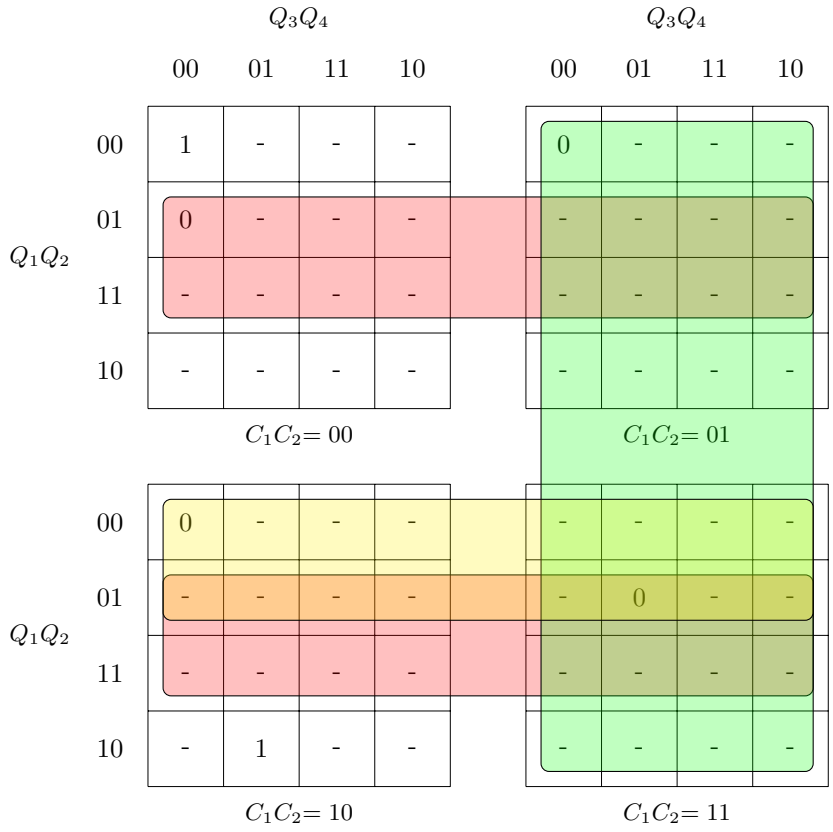
1.7.1 K-Map for J3 SOP



1.7.2 SOP Implementation for J3

The SOP implementation for this K-Map is $Q_1 + \overline{C_1} \cdot \overline{C_2} \cdot \overline{Q_2}$. The expression $\overline{C_1} \cdot \overline{C_2} \cdot \overline{Q_2}$ is represented by the red implicant, and the expression Q_1 is represented by the green implicant.

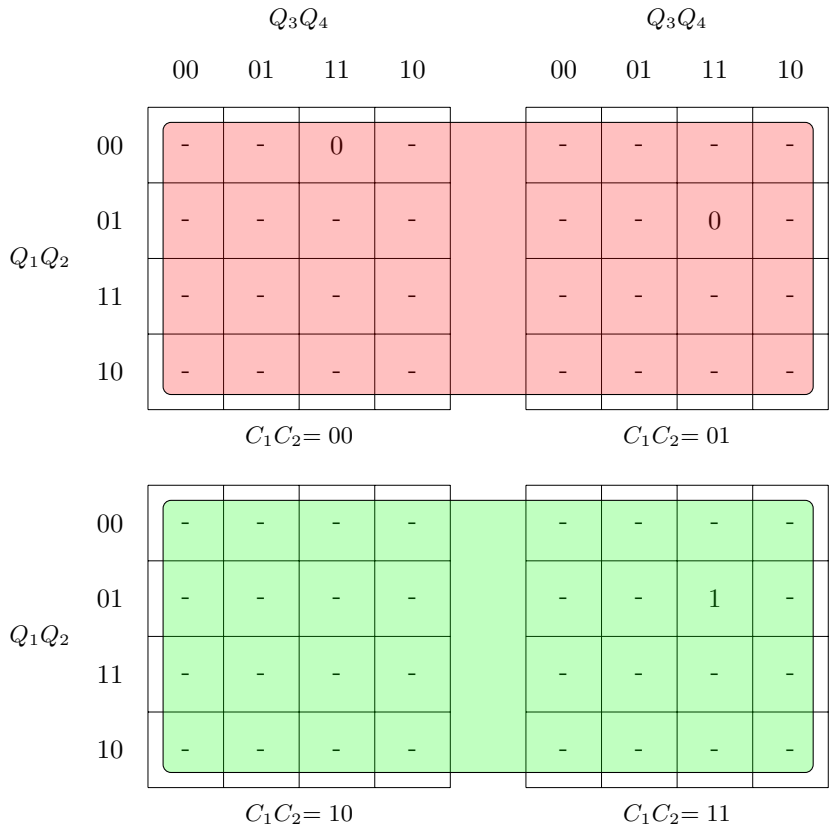
1.7.3 K-Map for J3 POS



1.7.4 POS Implementation for J3

The POS implementation is $\overline{Q_2} \cdot \overline{C_2} \cdot (\overline{C_1} + \overline{Q_1})$. The expression $\overline{Q_2}$ is represented by the red implicant, the expression $\overline{C_2}$ is represented by the green implicant, and the expression $\overline{A} + \overline{C}$ is represented by the yellow implicant.

1.8 K-Map for !K3



1.8.1 Explanation of K-Map Coloring for !K3

The POS is covered by the red implicant, and the SOP is covered by the green implicant.

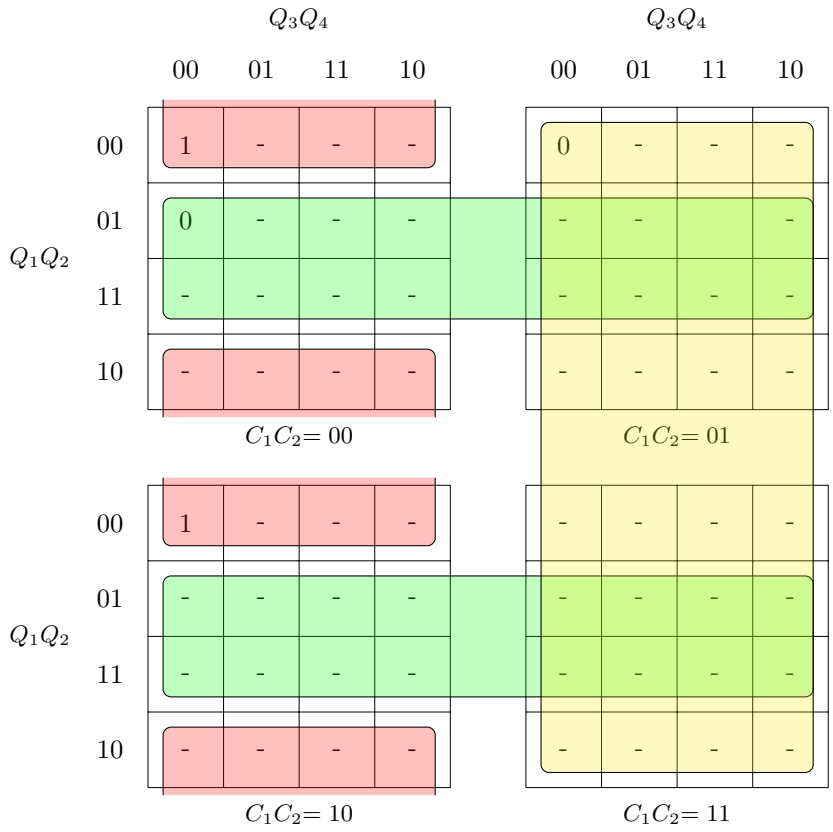
1.8.2 SOP Implementation for !K3

The SOP expression is C_1.

1.8.3 POS Implementation for !K3

The POS expression is C_1.

1.9 K-Map for J4



1.9.1 Explanation of K-Map Coloring for J4

The red implicant represent SOP and the yellow and green implicants represents POS.

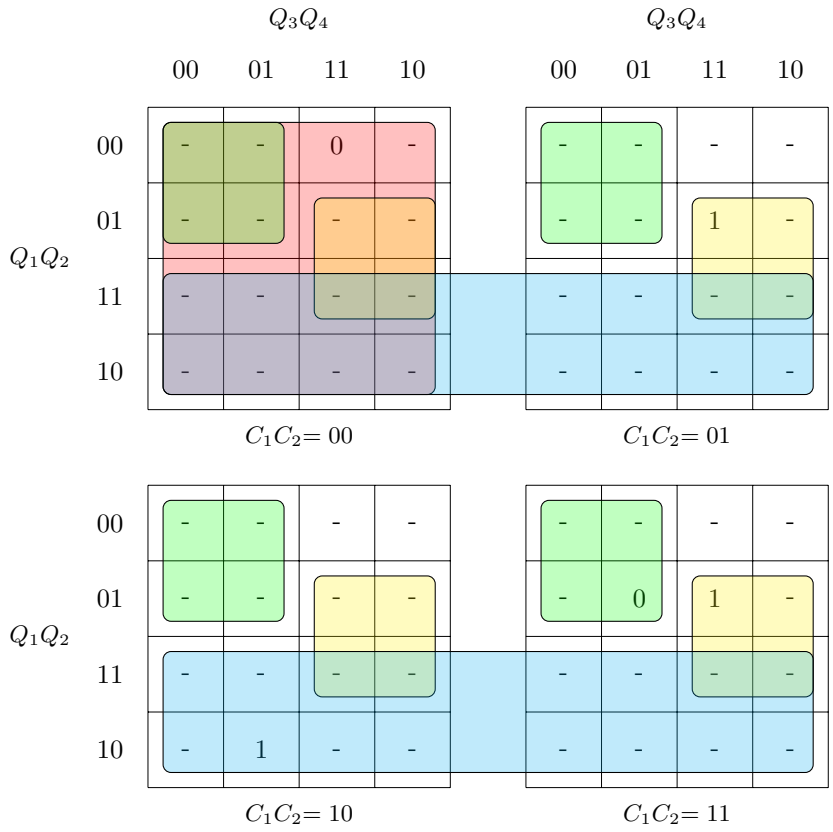
1.9.2 SOP Implementation for J4

The SOP expression is $\overline{Q_2} \cdot \overline{C_2}$. This expression is represented by the red and green sections which are a part of the same implicant.

1.9.3 POS Implementation for J4

The POS expression is $\overline{Q_2 + C_2}$ which can be simplified to $\overline{Q_2} \cdot \overline{C_2}$. This expression $\overline{Q_2}$ is represented by the yellow and purple implicant, and the expression $\overline{C_2}$ is represented by the blue implicant.

1.10 K-Map for !K4



1.10.1 Explanation of K-Map Coloring for !K4

The blue and yellow implicants represent SOP, and the red and green implicants represent POS.

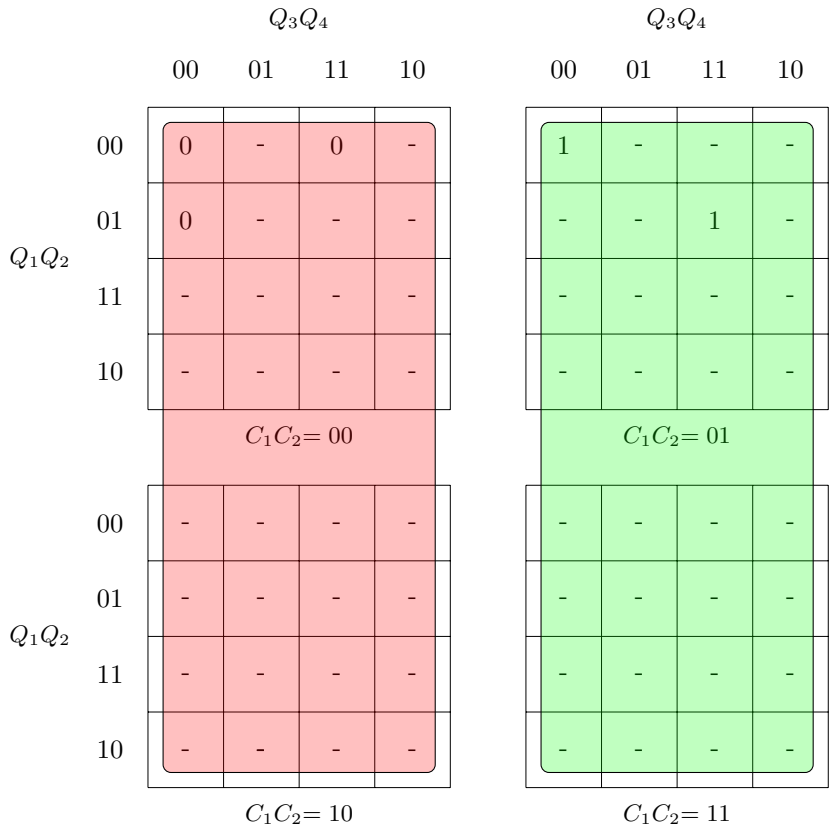
1.10.2 SOP Implementation for !K4

The SOP expression is $Q_1 + Q_2 \cdot Q_3$. The expression Q_1 is covered by the blue implicant, and the expression $Q_2 \cdot Q_3$ is covered by the yellow implicant.

1.10.3 POS Implementation for !K4

The POS expression is $(C_1 + C_2) \cdot (Q_1 + Q_3)$ which can also be written as $\overline{C_1 \cdot C_2} + \overline{Q_1 \cdot Q_3}$. This expression $(C_1 + C_2)$ is represented by the red implicant, and the expression $Q_1 + Q_3$ is represented by the green implicant.

1.11 K-Map for JC1



1.11.1 Explanation of K-Map Coloring for JC1

The SOP is covered by the green implicant and the POS is covered by the red implicant.

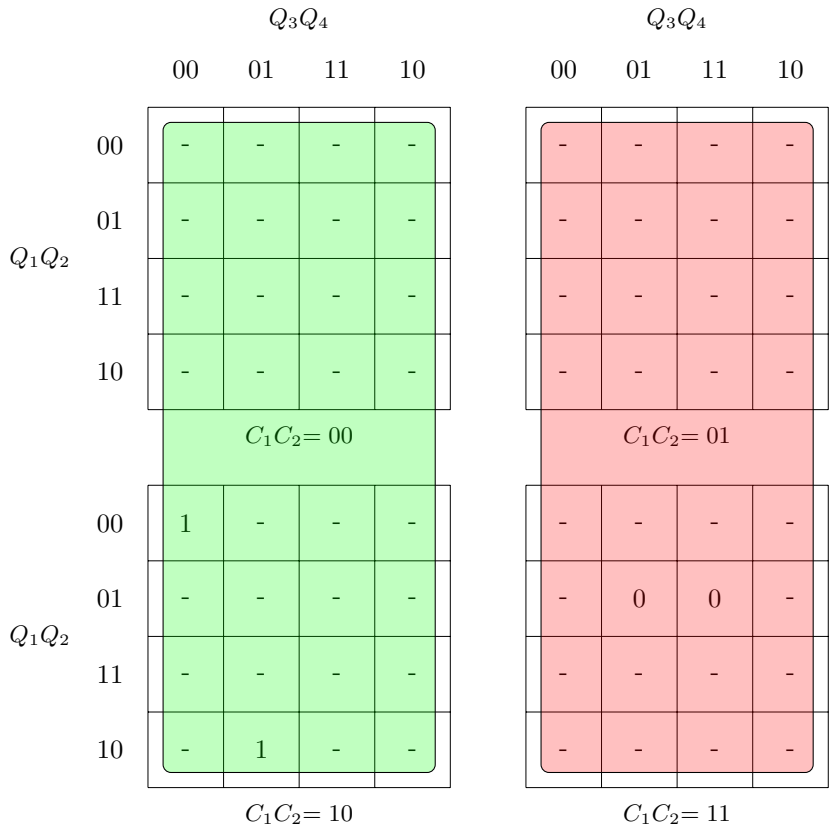
1.11.2 SOP Implementation for JC1

The expression for SOP is C_2

1.11.3 POS Implementation for JC1

The expression for POS is $\overline{\overline{C_2}}$ or C_2.

1.12 K-Map for !KC1



1.12.1 Explanation of K-Map Coloring for !KC1

The SOP is covered by the green implicant and the POS is covered by the red implicant.

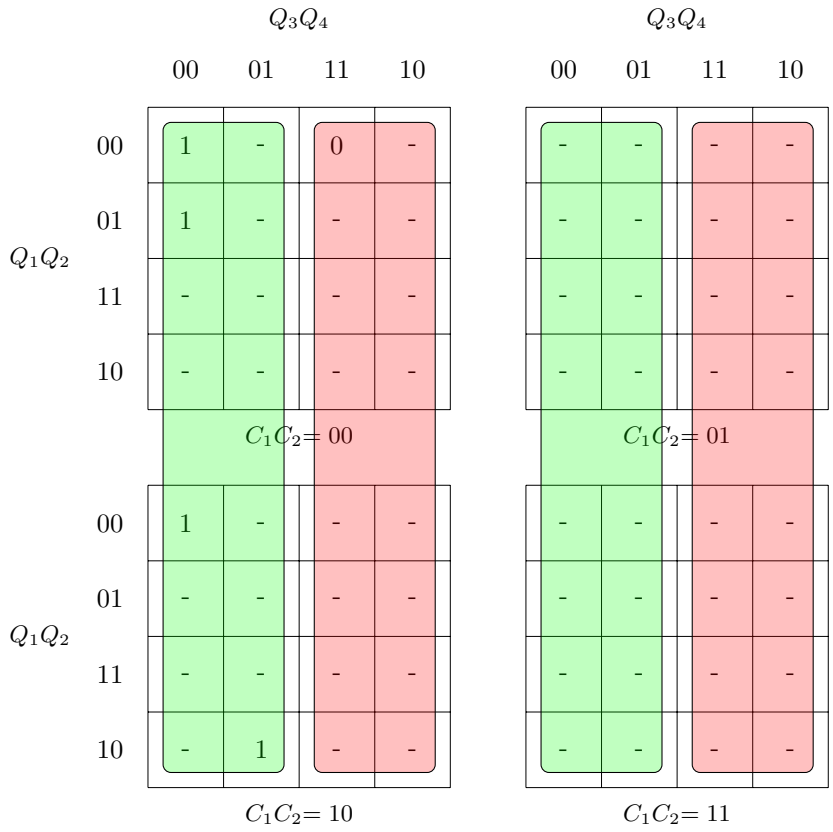
1.12.2 SOP Implementation for !KC1

The expression for SOP is $\overline{C_2}$

1.12.3 POS Implementation for !KC1

The expression for POS is $\overline{C_2}$

1.13 K-Map for JC2



1.13.1 Explanation of K-Map Coloring for JC2

The SOP is covered by the green implicant and the POS is covered by the red implicant.

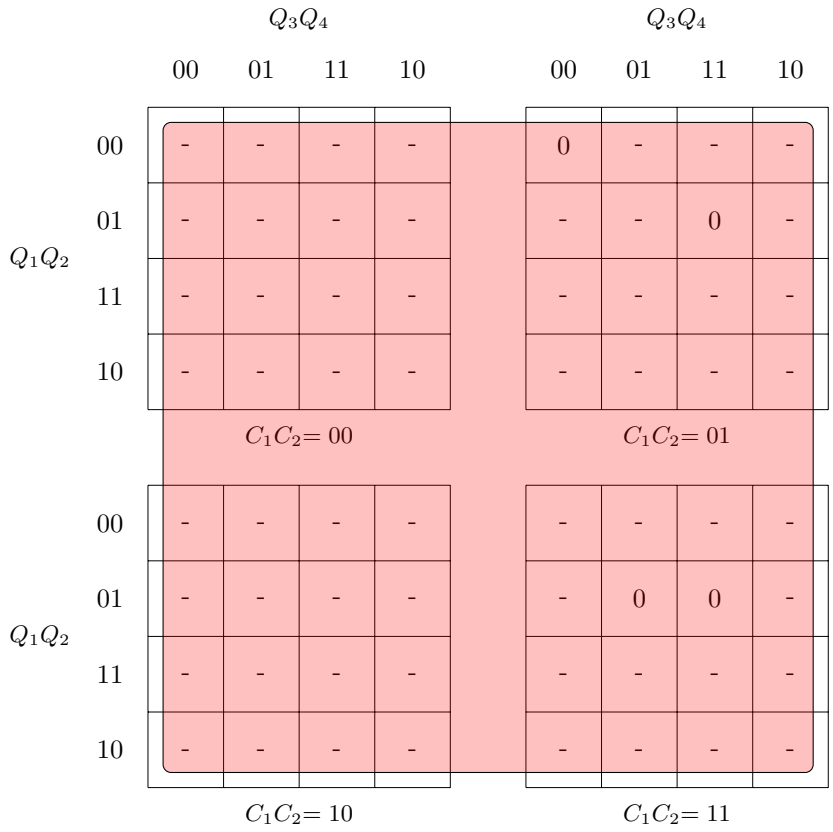
1.13.2 SOP Implementation for JC2

The expression for SOP is $\overline{Q_3}$.

1.13.3 POS Implementation for JC2

The expression for POS is $\overline{Q_3}$

1.14 K-Map for !KC2



1.14.1 Explanation of K-Map Coloring for !KC2

The entire K-Map is covered by a red implicant with maxterms and indeterminants. Thus, is just 0.

1.14.2 SOP Implementation for !KC2

The SOP expression is 0.

1.14.3 POS Implementation for !KC2

The POS expression is 0.

1.15 Implementations Table

Input	SOP Implementation	POS Implementation	Optimal Implementation
J_1	$Q_3 \cdot \overline{C_1} \cdot C_2$	$Q_3 \cdot \overline{C_1} \cdot C_2$	$Q_3 \cdot \overline{C_1} \cdot C_2$
$!K_1$	0	0	0
J_2	$\overline{C_2}$	$\overline{C_2}$	$\overline{C_2}$
$!K_2$	0	0	0
J_3	$Q_1 + \overline{C_1} \cdot \overline{C_2} \cdot \overline{Q_2}$	$\overline{Q_2} \cdot \overline{C_2} \cdot (\overline{C_1} + \overline{Q_1})$	$Q_1 + \overline{C_1} \cdot \overline{C_2} \cdot \overline{Q_2}$
$!K_3$	C_1	C_1	C_1
J_4	$\overline{Q_2} \cdot \overline{C_2}$	$\overline{Q_2} \cdot \overline{C_2}$	$\overline{Q_2} \cdot \overline{C_2}$
$!K_4$	$Q_1 + Q_2 \cdot Q_3$	$(C_1 + C_2) \cdot (Q_1 + Q_3)$	$Q_1 + Q_2 \cdot Q_3$
J_{C_1}	C_2	C_2	C_2
$!K_{C_1}$	$\overline{C_2}$	$\overline{C_2}$	$\overline{C_2}$
J_{C_2}	$\overline{Q_3}$	$\overline{Q_3}$	$\overline{Q_3}$
$!K_{C_2}$	0	0	0

2 Simulated Results

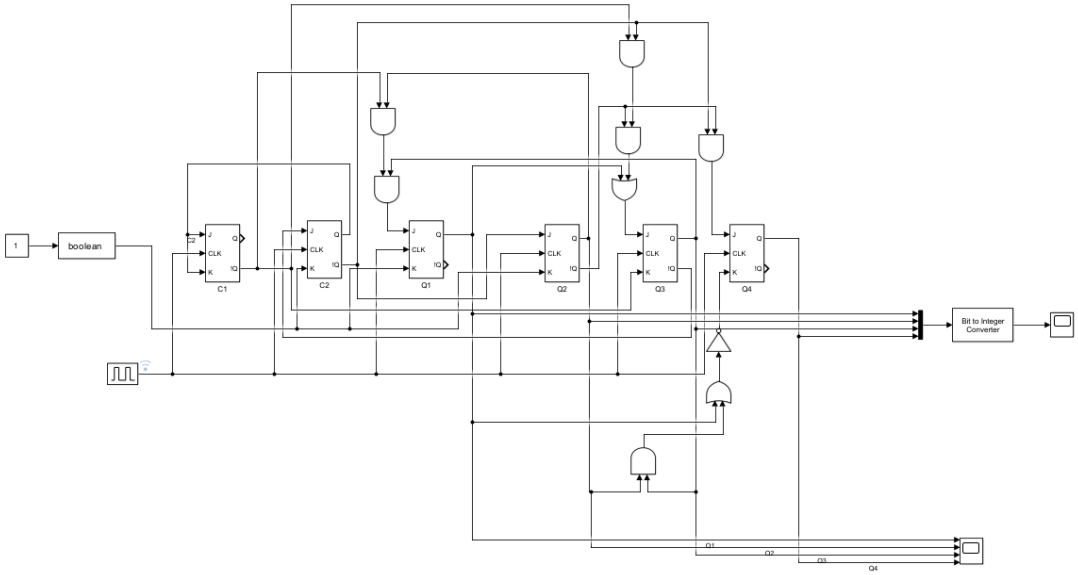


Figure 1: Simulated Circuit

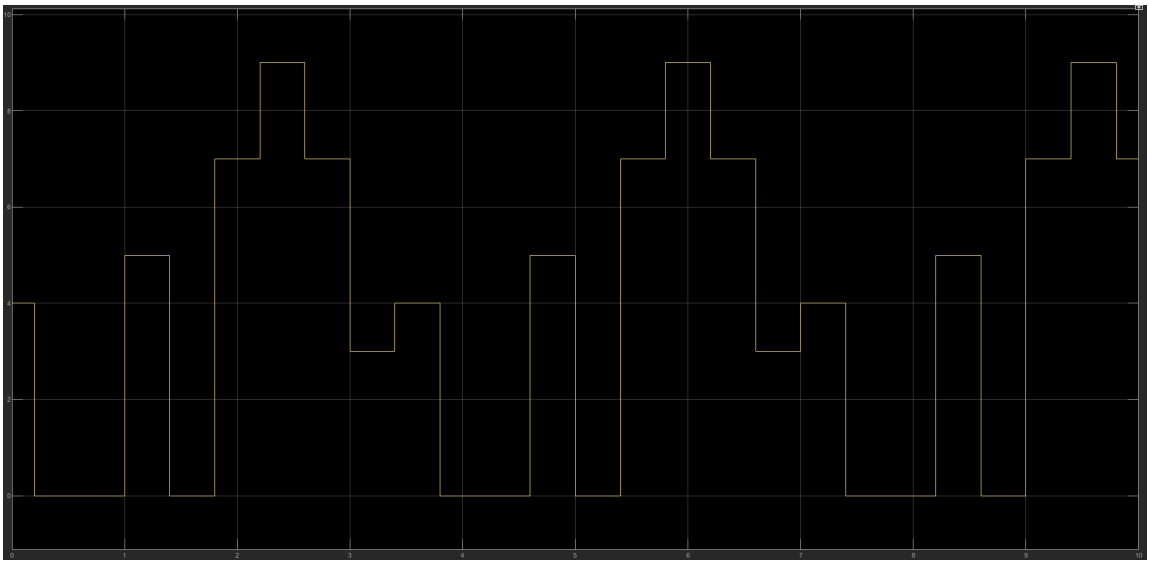


Figure 2: Binary To Integer Conversion of Qs

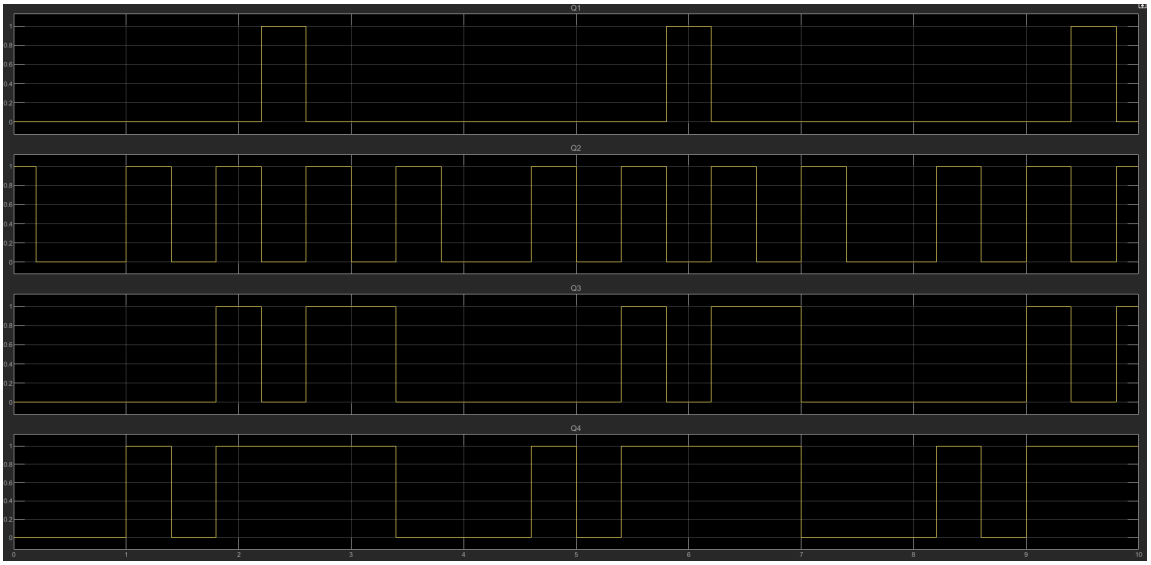


Figure 3: Timing Diagrams of Q

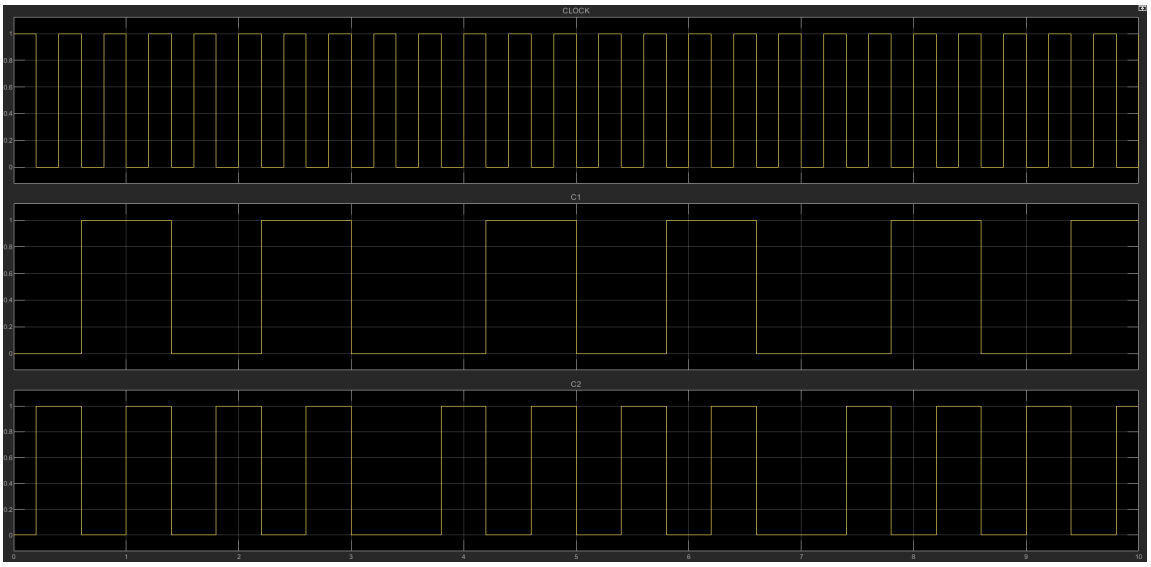


Figure 4: Timing Diagrams of C and Clock

2.1 Discussion of Simulated Results

We see that the all timings and output match expected results. The JK-Flip-Flops in Simulink however, are negative-edge triggered thus will be out of phase by half a period, but otherwise will be unaffected. The output, shown in figure 2, shows the number student number 400507973,

being displayed sequentially at a clock frequency of 2.5Hz. As Simulink does not have a 7SD or 7SD encoder, a scope and bit/binary to integer converter was used in place of it, the physical build will include the ICs instead. While the Simulation's initial state is 000100, in the order of $C_1C_2Q_1Q_2Q_3Q_4$, for simplicity it can be started at 000000, as it is also a part of the stable loop, the third 0 in the student number.

2.2 Debugging

The circuit was initially:

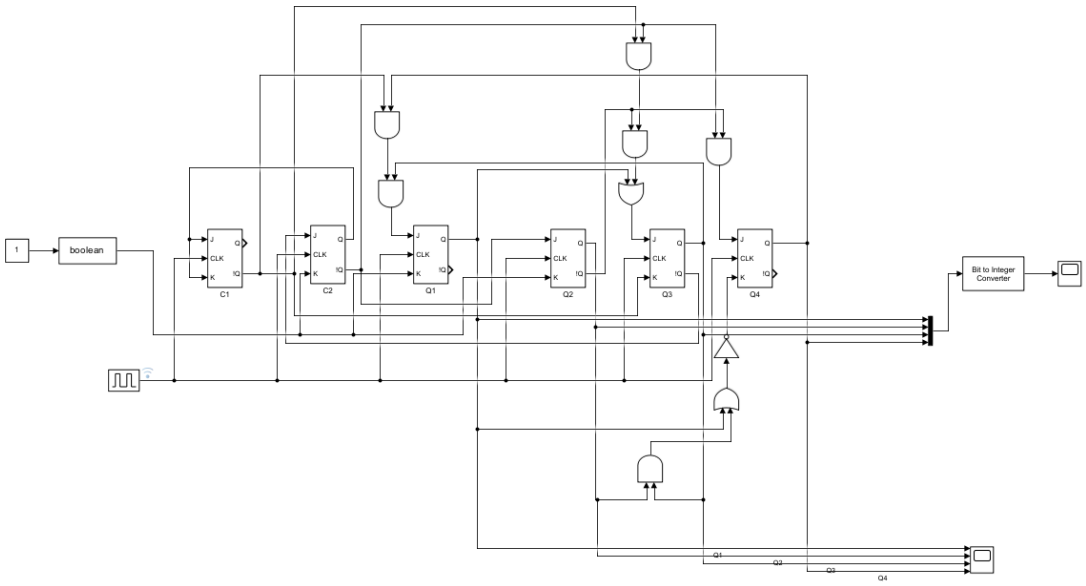


Figure 5: Initial Implementation of Circuit in Simulation

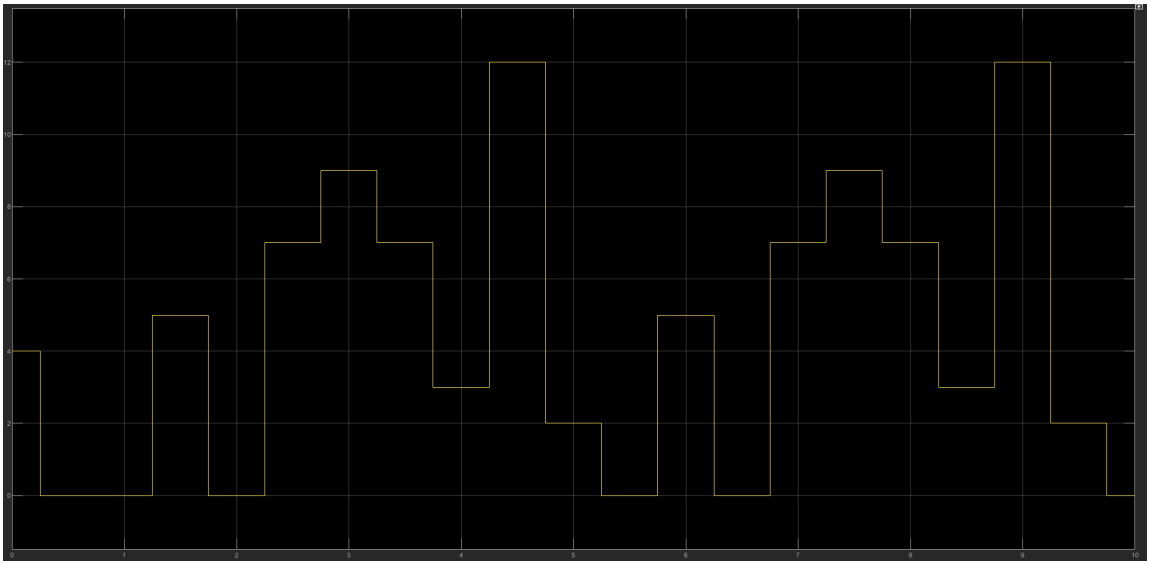


Figure 6: Initial Output of Circuit

From the above output, it can be observed that the initial loop correctly displays the student number. However, something goes wrong in following loops. The issue can be narrowed down by looking through the timing diagrams for each Q, and comparing these to the excitation table.

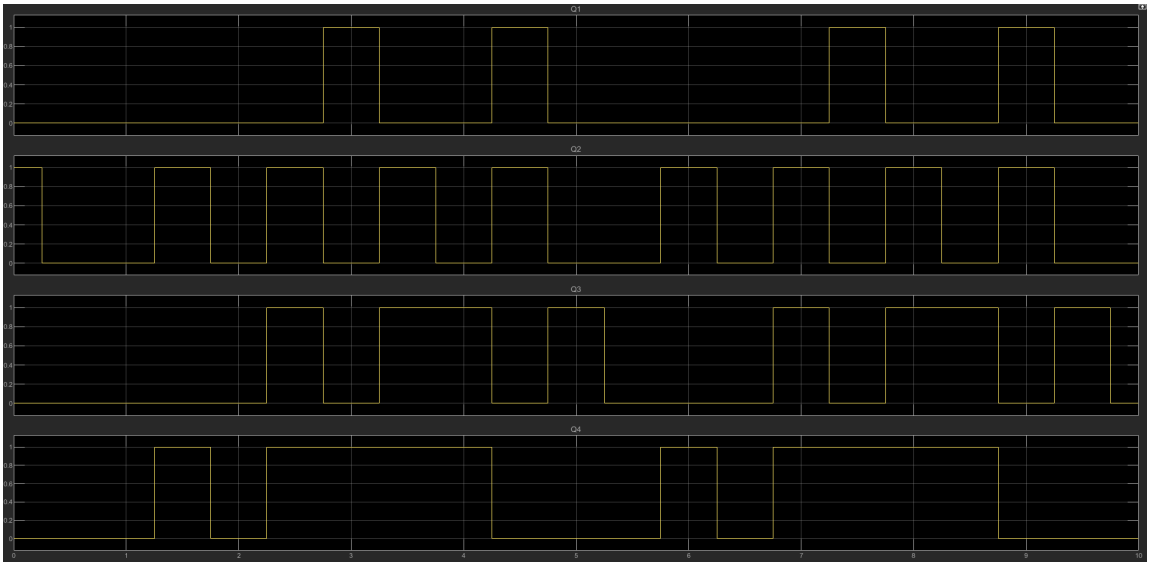


Figure 7: Scope for Output Qs

From this graph we find that Q_1 becomes high an additional time. Reviewing the connections for Q_1, it was determined that Q_4 was connected rather than Q_2. Changing this connection, we find

the circuit to be working as intended, repeating all numbers in a cycle. Furthermore, as having a NOT gate would introduce a new IC

3 Physical Circuit