ENGPHYS 2E04 Design Project

Marko Buha
buham
400411764

Analytical

Note the designs and alternatives have been presented in the order that I explored them, so the alternatives precede the design.

Alternative 1

The goal is to create a finite state machine that cycles through my student number on a clock signal. I will begin by defining what the four q bits and relevant counter bits must be to create this machine. The four q bits represent the binary representation of my number, from q1 being the most significant bit to q4 being the least significant bit. The counter bits, of which I need two since the number 4 appears three times in my student number, have been allocated in a method that allows the bits to toggle as much as possible since that is easy to implement. Additionally, using don't cares as much as possible to allow the machine to be more robust and work for more starting states, and means my partner and I would be less likely to have to pre-set or clear pins, saving us a few minutes of the limited time we have in the lab room.

Number	q1	q2	q3	q4	c5	c6
4	0	1	0	0	0	0
0	0	0	0	0	X	1
0	0	0	0	0	X	0
4	0	1	0	0	0	1
1	0	0	0	1	X	0
1	0	0	0	1	X	1
7	0	1	1	1	X	X
6	0	1	1	0	X	X
4	0	1	0	0	1	X

Table 1. Design 1 Desired Binary Values for All Bits

The obvious simplification here is that the gods running McMaster's random student number generator have blessed me with a number in which no digits are greater than 7. Thus the most significant bit is always 0 and can simply be grounded instead of using a flipflop. This reduces the flipflop count to 5 as q1 does not require a flipflop. Of the remaining 3 q bits and 2 counters, none are identical to each other, and none appear to be determined uniquely by the state of another bit. Thus, I conclude that no major simplifications remain.

State Transition Table

Firstly, I will list in a truth table what each J and !K combo does on the flipflop for later reference.

J	!K	Qnext
0	0	0 (reset)
0	1	q (hold)
1	0	!q (toggle)
1	1	1 (set)

Table 2. Flipflop Truth Table

Secondly, I must create an excitation table to determine which inputs cause which outputs on the flipflop. In the excitation table, there are always 2 ways to get from one number to another, so there will be 1 don't care condition.

Current State	Next State	J	!K
0	0	0	X
0	1	1	X
1	0	X	0
1	1	X	1

Table 3. Excitation Table

The third thing that must be done is the creation of a truth table or state transition diagram. This table will determine what the J and !K inputs must be to perform the desired flip-flop operation. Note that lowercase q refers to the current state, while capital Q refers to the next state. q1 and Q1 have been excluded as they are always grounded.

Note that may have to satisfy 2 rows of the excitation table at once when I cannot guarantee what the current state is because it is a don't care. However, I can set J and !K to don't care whenever I don't care about what the flipflop does, that is, when its next state is a don't care.

q2	q3	q4	c5	c6	Q2	Q3	Q4	C5	C6	J2	!K2	J3	!K3	J4	!K4	J5	!K5	J6	!K6
1	0	0	0	0	0	0	0	X	1	X	0	0	X	0	X	X	X	1	X
0	0	0	X	1	0	0	0	X	0	0	X	0	X	0	X	X	X	X	0
0	0	0	X	0	1	0	0	0	1	1	X	0	X	0	X	0	0	1	X
1	0	0	0	1	0	0	1	X	0	X	0	0	X	1	X	X	X	X	0
0	0	1	X	0	0	0	1	X	1	0	X	0	X	X	1	X	X	1	X
0	0	1	X	1	1	1	1	X	X	1	X	1	X	X	1	X	X	X	X
1	1	1	X	X	1	1	0	X	X	X	1	X	1	X	0	X	X	X	X
1	1	0	X	X	1	0	0	1	X	X	1	X	0	0	X	1	1	X	X
1	0	0	1	X	1	0	0	0	0	X	1	0	X	0	X	X	0	0	0
q2	q3	q4	c5	c6	Q2	Q3	Q4	C5	C6	J2	!K2	J3	!K3	J4	!K4	J5	!K5	J6	!K6

Table 4. Alternative 1 Truth Table/State Transition Table

Recolouring this table by row to make future K-mapping easier.

q2	q3	q4	c5	c6	Q2	Q3	Q4	C5	C6	J2	!K2	J3	!K3	J4	!K4	J5	!K5	J6	!K6
1	0	0	0	0	0	0	0	X	1	X	0	0	X	0	X	X	X	1	X
0	0	0	X	1	0	0	0	X	0	0	X	0	X	0	X	X	X	X	0
0	0	0	X	0	1	0	0	0	1	1	X	0	X	0	X	0	0	1	X
1	0	0	0	1	0	0	1	X	0	X	0	0	X	1	X	X	X	X	0
0	0	1	X	0	0	0	1	X	1	0	X	0	X	X	1	X	X	1	X
0	0	1	X	1	1	1	1	X	X	1	X	1	X	X	1	X	X	X	X
1	1	1	X	X	1	1	0	X	X	X	1	X	1	X	0	X	X	X	X
1	1	0	X	X	1	0	0	1	X	X	1	X	0	0	X	1	1	X	X
1	0	0	1	X	1	0	0	0	0	X	1	0	X	0	X	X	0	0	0
q2	q3	q4	c5	c6	Q2	Q3	Q4	C5	C6	J2	!K2	J3	!K3	J4	!K4	J5	!K5	J6	!K6

Table 5. Alternative 1 Useful Truth Table/State Transition Table

K-Mapping

I will need to create 10 K-Maps, one for each J and !K. I will then use Product of Sums K-mapping, as there seem to be generally more 0s than 1s in my Qs and Cs, allowing me to create larger boxes and hopefully simpler expressions. I will need to use 5 input K-maps as I have 5 different flip-flops that need to change together. Note that states which are not in my student number are represented with don’t care conditions.

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	1	1	0	0	00	0	0	1	1
01	X	X	X	X	01	X	X	X	X
11	X	X	X	X	11	X	X	X	X
10	X	X	X	X	10	X	X	X	X

Table 6. Alternative 1 J2 K-map

$J2 = (\overline{q4} + c6)(q4 + \overline{c6})$
$J2 = q4\overline{q4} + c6q4 + \overline{q4c6} + c6\overline{c6}$
$J2 = c6q4 + \overline{q4c6}$
$J2 = \overline{q4 \oplus c6}$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	Q2Q3/Q4C5	00	01	11	10
00	X	X	X	X	00	X	X	X	X
01	X	X	X	X	01	X	X	X	X
11	1	1	1	1	11	1	1	1	1
10	0	1	X	X	10	0	1	X	X

Table 7. Alternative 1 !K2 K-map

$\overline{K2} = q3 + c5$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	0	0	0	0	00	0	0	1	1
01	X	X	X	X	01	X	X	X	X
11	X	X	X	X	11	X	X	X	X
10	0	0	X	X	10	0	0	X	X

Table 8. Alternative 1 J3 K-map

$J3 = c6q4$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	Q2Q3/Q4C5	00	01	11	10
00	X	X	X	X	00	X	X	X	X
01	X	X	X	X	01	X	X	X	X
11	0	0	1	1	11	0	0	1	1
10	X	X	X	X	10	X	X	X	X

Table 9. Alternative 1 !K3 K-map

$$\overline{K3} = \overline{q4}$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	0	0	X	X	00	0	0	X	X
01	X	X	X	X	01	X	X	X	X
11	0	0	X	X	11	0	0	X	X
10	0	0	X	X	10	1	0	X	X

Table 10 Alternative 1. J4 Design 1 K-map (This K-map is hard to draw out, but there are 4 boxes that cover 16 input possibilities each. That happens when the 1 is almost surrounded by 0s)

$$J4 = c6\ q2\ \overline{q3}\ \overline{c5}$$

q2q3/q4c5	00	01	11	10	Q2Q3/Q4C5	00	01	11	10
00	X	X	1	1	00	X	X	1	1
01	X	X	X	X	01	X	X	X	X
11	X	X	0	0	11	X	X	0	0
10	X	X	X	X	10	X	X	X	X

Table 11. Alternative 1 !K4 K-map

$$\overline{K4} = \overline{q2}$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	0	0	X	X	00	X	X	X	X
01	X	X	X	X	01	X	X	X	X
11	1	1	X	X	11	1	1	X	X
10	X	X	X	X	10	X	X	X	X

Table 12. Alternative 1 J5 Design 1 K-map

$$J5 = q2$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	0	0	X	X	00	X	X	X	X
01	X	X	X	X	01	X	X	X	X
11	1	1	X	X	11	1	1	X	X
10	X	0	X	X	10	X	0	X	X

Table 13. Alternative 1 !K5 Design 1 K-map

$$\overline{K5} = q2q3$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	1	1	1	1	00	X	X	X	X
01	X	X	X	X	01	X	X	X	X
11	X	X	X	X	11	X	X	X	X
10	1	0	X	X	10	X	0	X	X

Table 14. Alternative 1 J6 Design 1 K-map

$$J6 = \overline{q2} + \overline{c5}$$

There is no k-map for !K6 as it never needs to be 1, thus I can simply ground it.

$$\overline{K6} = 0$$

Alternative 1 requires 8 two-input gates and 5 flip-flops. This includes 5 AND gates, 2 OR gates, and 1 XNOR gate. Alternative 1 requires 7 chips. This includes 2 AND chips, 1 OR chip, 1 XNOR chip, 1 NOT chip, and 3 flip-flop chips. 1 chip could have been saved by using ANDs and ORs rather than the XNOR but that complicates the logic further, making the physical harder to test. It also adds 2 gates.

Alternative 2

Well, that was a lot of gates. However, numbering all the counters so that they follow a toggling like sequence as much as possible should save me some gates. 2 consecutive 0s in the c5 and c6 values were required as there are a non-even number of states. Excitation and truth tables for the flip-flop are unchanged and will always remain unchanged until the lab buys a different model of flip-flops.

State Transition Table

Number	q1	q2	q3	q4	c5	c6
4	0	1	0	0	0	1
0	0	0	0	0	1	0
0	0	0	0	0	0	1
4	0	1	0	0	1	0
1	0	0	0	1	0	1
1	0	0	0	1	1	0
7	0	1	1	1	0	1
6	0	1	1	0	1	0
4	0	1	0	0	0	0

Table 15. Alternative 2 Desired Binary Values for All Bits

Updating the columns for flip-flops 5 and 6 in the truth table.

q2	q3	q4	c5	c6	Q2	Q3	Q4	C5	C6	J2	!K2	J3	!K3	J4	!K4	J5	!K5	J6	!K6
1	0	0	0	1	0	0	0	1	0	X	0	0	X	0	X	1	X	X	0
0	0	0	1	0	0	0	0	0	1	0	X	0	X	0	X	X	0	1	X
0	0	0	0	1	1	0	0	1	0	1	X	0	X	0	X	1	X	X	0
1	0	0	1	0	0	0	1	0	1	X	0	0	X	1	X	X	0	1	X
0	0	1	0	1	0	0	1	1	0	0	X	0	X	X	1	1	X	X	0
0	0	1	1	0	1	1	1	0	1	1	X	1	X	X	1	X	0	1	X
1	1	1	0	1	1	1	0	1	0	X	1	X	1	X	0	1	X	X	0
1	1	0	1	0	1	0	0	0	0	X	1	X	0	0	X	X	0	0	X
1	0	0	0	0	1	0	0	0	1	X	1	0	X	0	X	0	X	1	X
q2	q3	q4	c5	c6	Q2	Q3	Q4	C5	C6	J2	!K2	J3	!K3	J4	!K4	J5	!K5	J6	!K6

Table 16. Alternative 2 Truth Table/State Transition Table

Recolouring this table by row to make future K-mapping easier.

q2	q3	q4	c5	c6	Q2	Q3	Q4	C5	C6	J2	!K2	J3	!K3	J4	!K4	J5	!K5	J6	!K6
1	0	0	0	1	0	0	0	1	0	X	0	0	X	0	X	1	X	X	0
0	0	0	1	0	0	0	0	0	1	0	X	0	X	0	X	X	0	1	X
0	0	0	0	1	1	0	0	1	0	1	X	0	X	0	X	1	X	X	0
1	0	0	1	0	0	0	1	0	1	X	0	0	X	1	X	X	0	1	X
0	0	1	0	1	0	0	1	1	0	0	X	0	X	X	1	1	X	X	0
0	0	1	1	0	1	1	1	0	1	1	X	1	X	X	1	X	0	1	X
1	1	1	0	1	1	1	0	1	0	X	1	X	1	X	0	1	X	X	0
1	1	0	1	0	1	0	0	0	0	X	1	X	0	0	X	X	0	0	X
1	0	0	0	0	1	0	0	0	1	X	1	0	X	0	X	0	X	1	X
q2	q3	q4	c5	c6	Q2	Q3	Q4	C5	C6	J2	!K2	J3	!K3	J4	!K4	J5	!K5	J6	!K6

Table 17. Alternative 2 Useful Truth Table/State Transition Table

K-Mapping

It is clear to see that more strategic numbering of the counters has provided major advantages at the K-Mapping stage. Namely, !K5 and !K6 can be tied to low, eliminating 2 K-maps and making them not dependent on the previous parts of the circuit functioning properly. Remaking the K-maps for Alternative 2 requires that the boxes are colour-coded again. It also expands the number of don't care conditions in each map as each state in the state transition table now only corresponds to one state in the K-map, not 2 or 4 like some previously did.

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	0	1	X	00	1	X	X	0
01	X	X	X	X	01	X	X	X	X
11	X	X	X	X	11	X	X	X	X
10	X	X	X	X	10	X	X	X	X

Table 18. Alternative 2 J2 K-map

$$J2 = (q4 + c6)(\overline{q4} + \overline{c6})$$
$$J2 = q4\overline{q4} + c6\overline{q4} + q4\overline{c6} + c6\overline{c6}$$
$$J2 = q4 \oplus c6$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	X	X	X	00	X	X	X	X
01	X	X	X	X	01	X	X	X	X
11	X	1	X	X	11	X	X	X	1
10	1	0	X	X	10	0	X	X	X

Table 19. Alternative 2 !K2 K-map

$$\overline{K2} = \left(q3+\overline{c5}\right)\left(q3+\overline{c6}\right)$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	0	1	X	00	0	X	X	0
01	X	X	X	X	01	X	X	X	X
11	X	X	X	X	11	X	X	X	X
10	0	0	X	X	10	0	X	X	X

Table 20. Alternative 2 J3 K-map

$$J3 = \; q4\overline{c6}$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	X	X	X	00	X	X	X	X
01	X	X	X	X	01	X	X	X	X
11	X	0	X	X	11	X	X	X	1
10	X	X	X	X	10	X	X	X	X

Table 21. Alternative 2 !K3 K-map (few different ways to make this one but already going to have to wire up C6 for J2 so might as well use it again)

$$\overline{K3} = c6$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	0	X	X	00	0	X	X	X
01	X	X	X	X	01	X	X	X	X
11	X	0	X	X	11	X	X	X	X
10	0	1	X	X	10	0	X	X	X

Table 22. Alternative 2 J4 K-map

$$J4 = \; q2\overline{q3}c5$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	X	1	X	00	X	X	X	1
01	X	X	X	X	01	X	X	X	X
11	X	X	X	X	11	X	X	X	0
10	X	X	X	X	10	X	X	X	X

Table 23. Alternative 2 !K4 K-map

$$\overline{K4} = \overline{q2}$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	X	X	X	00	1	X	X	1
01	X	X	X	X	01	X	X	X	X
11	X	X	X	X	11	X	X	X	1
10	0	X	X	X	10	1	X	X	X

Table 24. Alternative 2 J5 K-map

$$J5 = c6$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	1	1	X	00	X	X	X	X
01	X	X	X	X	01	X	X	X	X
11	X	0	X	X	11	X	X	X	X
10	1	1	X	X	10	X	X	X	X

Table 25. Alternative 2 J6 K-map

$$J6 = \overline{q3}$$

$$\overline{K5} = 0$$

$$\overline{K6} = 0$$

Alternative 2 requires 7 two-input gates and 5 flip-flops. This includes 4 AND gates, 2 OR gates, and 1 XOR gate. Alternative 2 requires 6 chips. This includes 1 AND chip, 1 OR chip, 1 XOR chip, and 3 flip-flop chips. Note that replacing the XOR gate with AND and OR chips does not save a chip as a 5th AND gate would require a 2nd AND chip since the lab has quad-gate chips.

Design

What if I use Sum of Products K-mapping and see if that does anything about the villainous !K2 and J4 setting out on their evil plans to overcomplicate the project by requiring multiple gates? Thankfully, I do not need to re-do the state transition table and can reuse the one for Alternative 2 since the counter bits do not change. Thus, !K5 and !K6 can still be grounded. The relevant excitation and truth tables for the flip-flops can be found in Alternative 1.

K-Mapping

Same as before but with Sum of Products.

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	0	1	X	00	1	X	X	0
01	X	X	X	X	01	X	X	X	X
11	X	X	X	X	11	X	X	X	X
10	X	X	X	X	10	X	X	X	X

Table 26. Design J2 K-map

$$J2 = q4\,\overline{c6} + \overline{q4}\,c6$$
$$J2 = q4 \oplus c6$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	X	X	X	00	X	X	X	X
01	X	X	X	X	01	X	X	X	X
11	X	1	X	X	11	X	X	X	1
10	1	0	X	X	10	0	X	X	X

Table 27. Design !K2 K-map

$$\overline{K2} = q3 + \overline{c5}\,\overline{c6}$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	0	1	X	00	0	X	X	0
01	X	X	X	X	01	X	X	X	X
11	X	X	X	X	11	X	X	X	X
10	0	0	X	X	10	0	X	X	X

Table 28. Design J3 K-map

$$J3 = q4\overline{c6}$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	X	X	X	00	X	X	X	X
01	X	X	X	X	01	X	X	X	X
11	X	0	X	X	11	X	X	X	1
10	X	X	X	X	10	X	X	X	X

Table 29. Design !K2 K-map (few different ways to make this one but already going to have to wire up C6 for J2 so might as well use it again)

$$\overline{K3} = c6$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	0	X	X	00	0	X	X	X
01	X	X	X	X	01	X	X	X	X
11	X	0	X	X	11	X	X	X	X
10	0	1	X	X	10	0	X	X	X

Table 30. Design J4 K-map

$$J4 = q2\overline{q3}c5$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	X	1	X	00	X	X	X	1
01	X	X	X	X	01	X	X	X	X
11	X	X	X	X	11	X	X	X	0
10	X	X	X	X	10	X	X	X	X

Table 31. Design !K4 K-map

$$\overline{K4} = \overline{q2}$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	X	X	X	00	1	X	X	1
01	X	X	X	X	01	X	X	X	X
11	X	X	X	X	11	X	X	X	1
10	0	X	X	X	10	1	X	X	X

Table 32. Design J5 K-map

$$J5 = c6$$

	c6=0	c6=0	c6=0	c6=0		c6=1	c6=1	c6=1	c6=1
q2q3/q4c5	00	01	11	10	q2q3/q4c5	00	01	11	10
00	X	1	1	X	00	X	X	X	X
01	X	X	X	X	01	X	X	X	X
11	X	0	X	X	11	X	X	X	X
10	1	1	X	X	10	X	X	X	X

Table 33. Design J6 K-map

$$J6 = \overline{q3}$$

$$\overline{K5} = 0$$

$$\overline{K6} = 0$$

The design requires 6 two-input gates and 5 flip-flops. This includes 4 AND gates, 1 OR gate, and 1 XOR gate. The design requires 6 chips. This includes 1 AND chip, 1 OR chip, 1 XOR chip, and 3 flip-flop chips. Note that replacing the XOR gate with AND and OR chips does not save a chip as a 5th AND gate would require a 2nd AND chip since the lab has quad-gate chips.

Saving 2 gates compared to the more robust design saves at least 6 wires (2 inputs and 1 output per gate, times 3 gates), along with a chip and the 2 wires that needs to get powered and grounded. All implementing a proper reset function would require is linking up a combination of the 5 preset and clear pins corresponding to any state in my student number to a wire acting like a switch between low and high. Therefore, I have now determined to go with the design rather than the more robust Alternative 1, as it would save at least a few wires and a chip. !K2 of Alternative 1 is simpler than the design but is not usable as it was developed with a different counter system. I am taking the design over Alternative 2 as Alternative 2 is never simpler than the design.

Input	Alternative 1 (Don't Care PoS)	Alternative 2 (PoS)	Design (SoP)
J2	$J2 = \overline{q4 \oplus c6}$	$J2 = q4 \oplus c6$	$J2 = q4 \oplus c6$
!K2	$\overline{K2} = q3 + c5$	$\overline{K2} = \left(q3 + \overline{c5}\right)(q3 + \overline{c6})$	$\overline{K2} = q3 + \overline{c5}\,\overline{c6}$
J3	$J3 = c6q4$	$J3 = q4\overline{c6}$	$J3 = q4\overline{c6}$
!K3	$\overline{K3} = \overline{q4}$	$\overline{K3} = c6$	$\overline{K3} = c6$
J4	$J4 = c6\ q2\ \overline{q3}\ \overline{c5}$	$J4 = q2\overline{q3}c5$	$J4 = q2\overline{q3}c5$
!K4	$\overline{K4} = \overline{q2}$	$\overline{K4} = \overline{q2}$	$\overline{K4} = \overline{q2}$
J5	$J5 = q2$	$J5 = c6$	$J5 = c6$
!K5	$\overline{K5} = q2q3$	$\overline{K5} = 0$	$\overline{K5} = 0$
J6	$J6 = \overline{q2} + \overline{c5}$	$J6 = \overline{q3}$	$J6 = \overline{q3}$
!K6	$\overline{K6} = 0$	$\overline{K6} = 0$	$\overline{K6} = 0$

Table 34. Design Summary

Multisim

The below is a figure of my circuit constructed in Multisim along with 2 zoomed-in figures of it constructed in Multisim and a figure of the timing diagram. The timing diagram is of the states in the same order as my student number, which was easy to get as I wired up the not preset pins for the states that needed to be 1 and the not clear pins for the states that needed to be 0 to the same switch. The clock was set externally on the logic analyzer as an external clock is being used.

The flip-flops in the figures are, in order from left to right, q2, q3, q4, c5, c6.

To turn the analytical outputs into binary outputs corresponding to each segment of the 7-segment display, a decoder (U10 in the figures) was used. q1 (ground), q2, q3, and q4 were wired up to DD, DC, DB, and DA, in that order, to represent the binary number that should be displayed. Meanwhile, the not light test pin was wired to high to prevent all the lights from turning on at once, the not blank pin was set up using a not gate to blank the display on the rising clock edge, so it does not display random numbers while the flip-flops change state, and the not latch enable pin was grounded.

I have used blue for the outputs of Q2, green for the outputs of Q3, orange for the outputs of Q4, light grey for the outputs of C5, and purple for the outputs of C6. I have used red for constants that are high, and black for constants that are low. I have used a yellow-ish beige for the clock signal. I have used magenta for the pre-setter and clearer for the first state. I have used dark grey for the outputs of all AND gates and brown for the outputs of all OR and XOR gates.

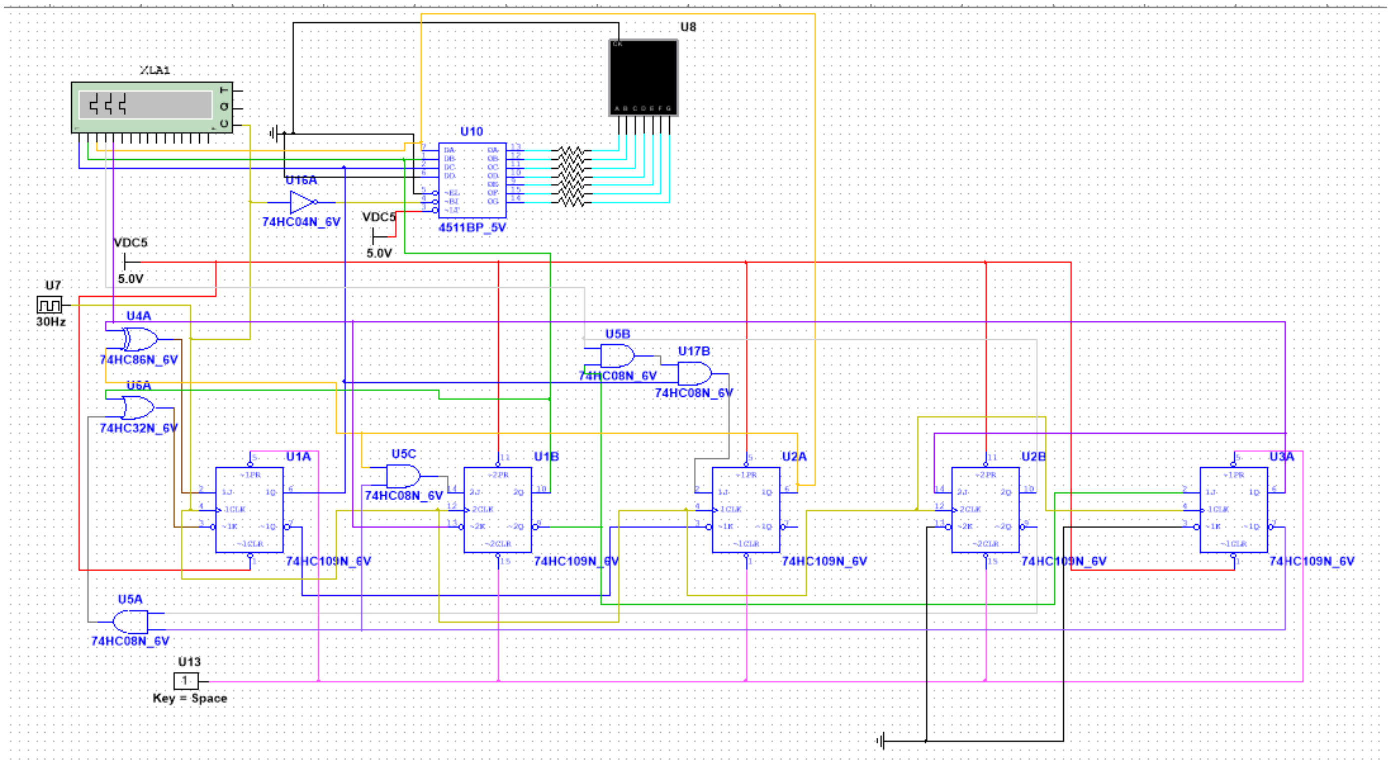


Figure 1. Multisim circuit

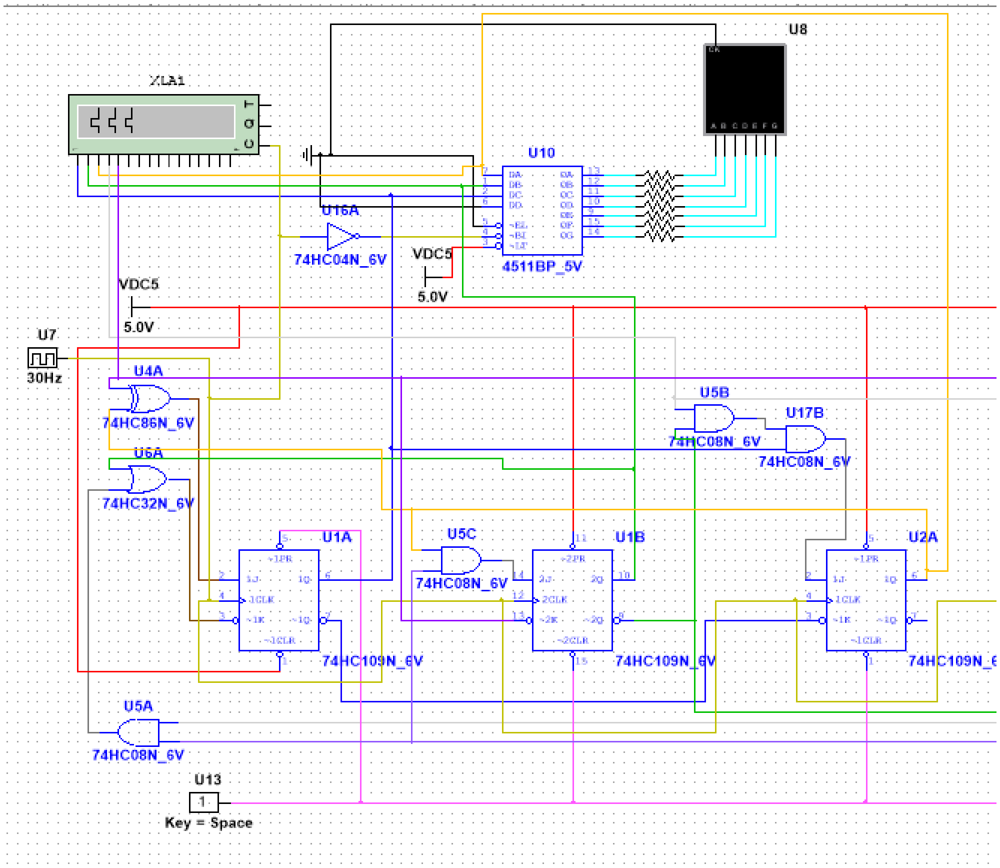


Figure 2. Zoomed-in Multisim circuit (Left side)

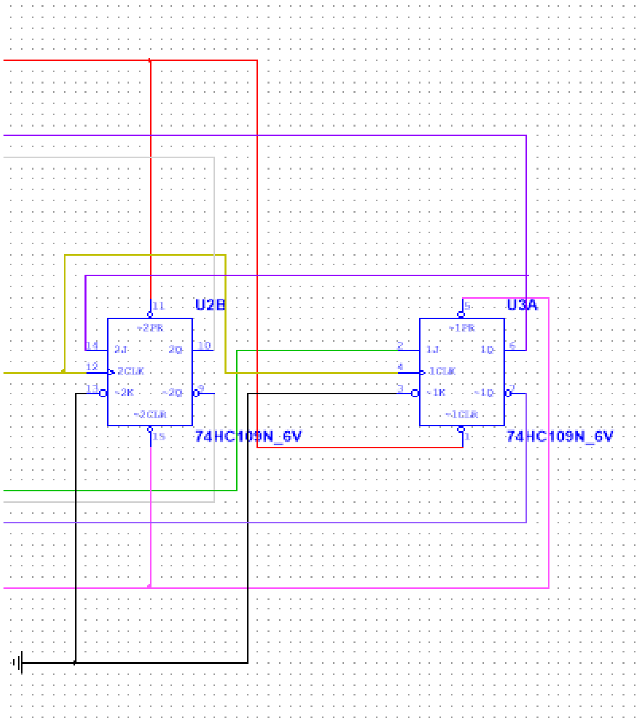
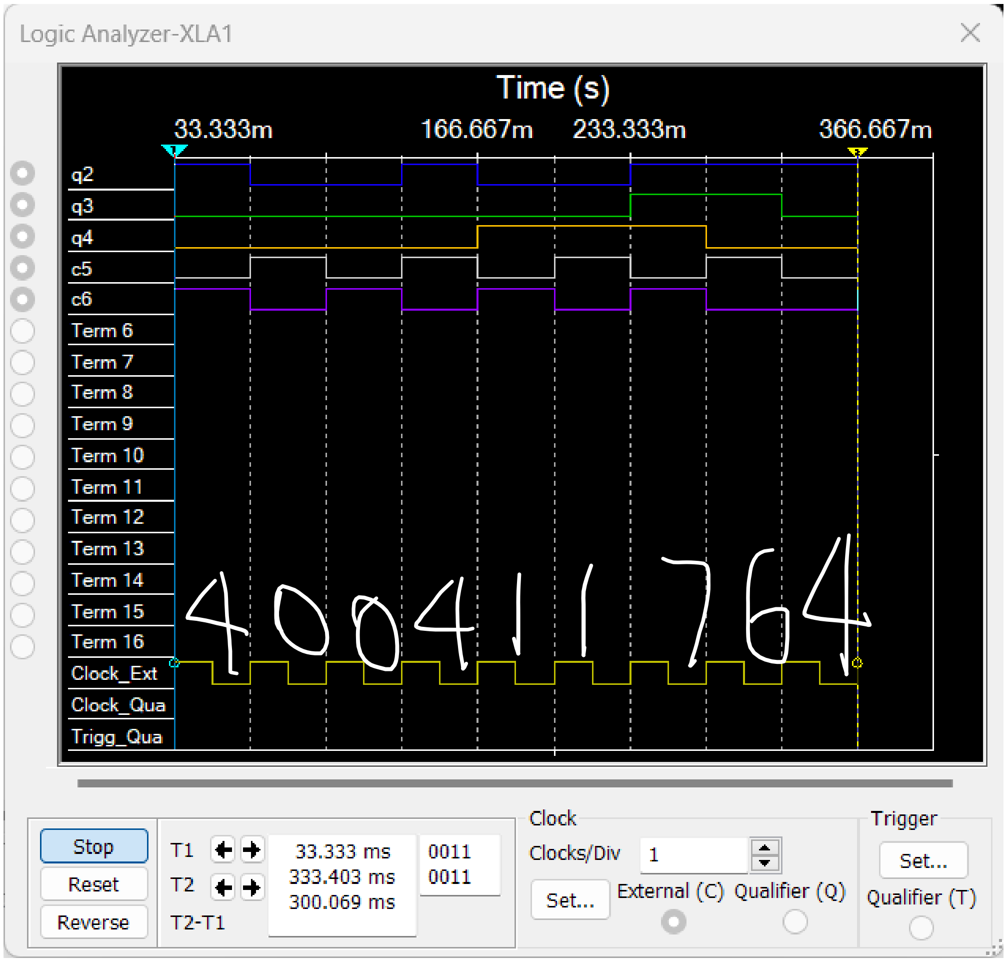


Figure 3. Zoomed-in Multisim circuit (Right side)



T1 33.333 ms 0011

T2 333.403 ms 0011

T2-T1 300.069 ms

Clocks/Div 1

Figure 4. Timing Diagram for the 5 flip-flops I have

The above timing diagram indicates the implementation of the analytical design produces the desired student number. Thus, the analytical must be correct.