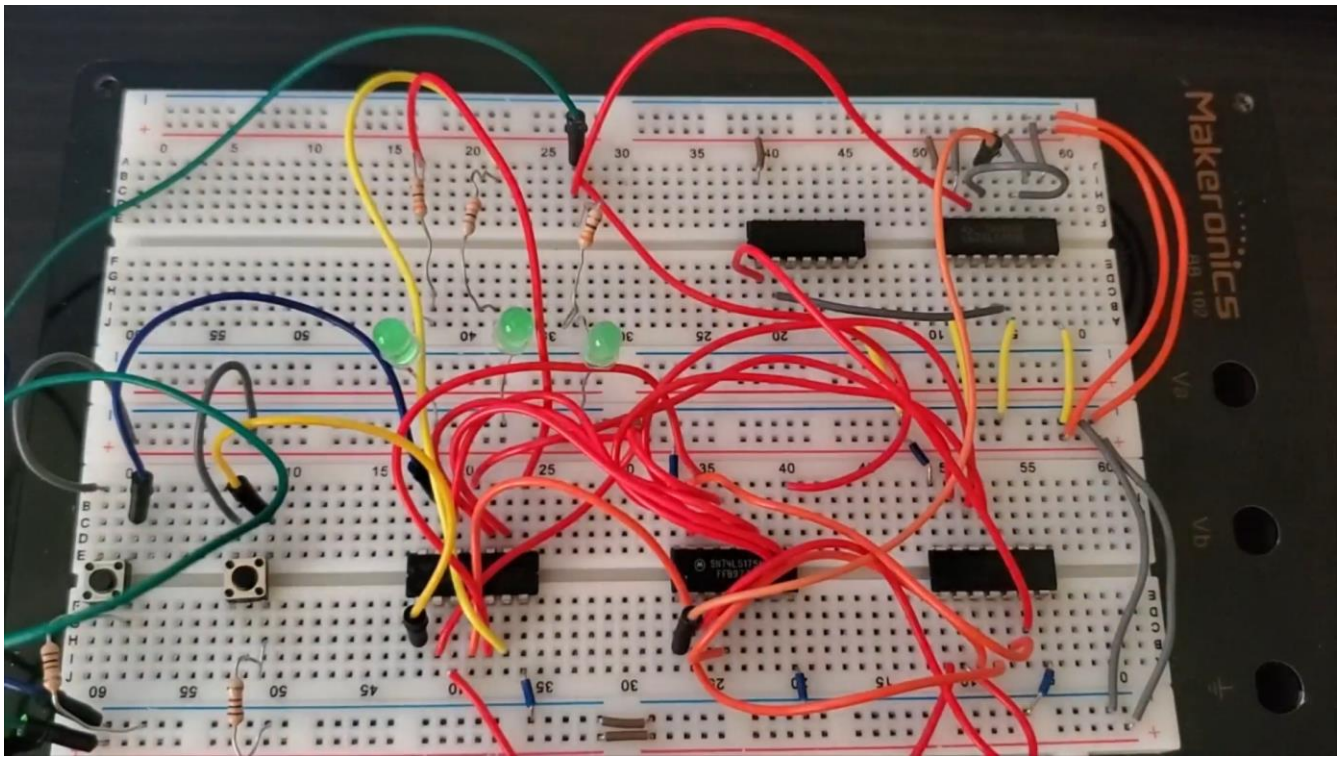
Finite State Machines

ECE2300L Module 6 Report



Kyler Martinez

December 6th, 2020

Introduction

The objectives of the lab are to be able to analyze a sequential circuit diagram and then create the state table and state diagram to describe the behavior of circuit. Students will then be able to synthesize a circuit by using a state diagram to derive the state table and finally the circuit schematic after determining the type of flip flops used. Finally, these skills would be applied to FSM applications such as pattern recognition and sensors as well as finding ways to turn complex problems into simpler ones that can be accomplished with a finite number of states.

Activity 6.1 — Analysis of a Finite State Machine

a.

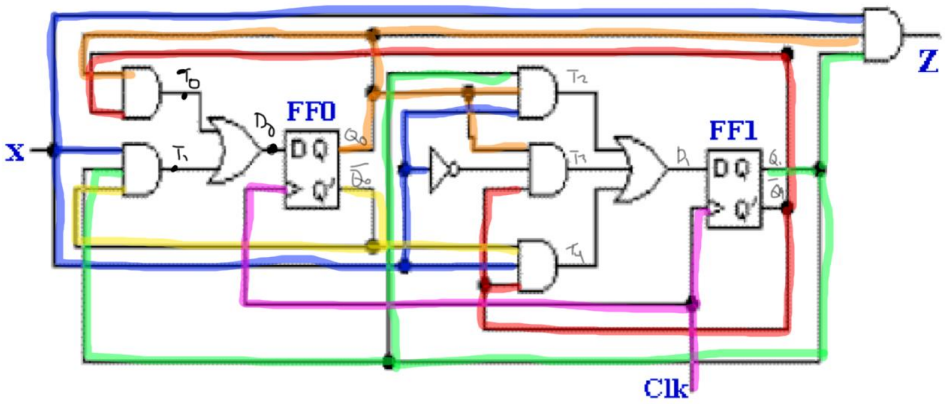


Figure 1: Circuit With Wires Highlighted For Visibility Along With Intermediate Labels.

$T_0 = Q_0Q_1'$ $T_1 = XQ_1Q_0'$ $D_0 = T_0+T_1 = Q_0Q_1' + XQ_1Q_0'$

$T_2 = XQ_0Q_1$ $T_3 = X'Q_0Q_1'$ $T_4 = XQ_0'Q_1'$ $D_1 = T_2+T_3+T_4 = XQ_0Q_1 + X'Q_0Q_1' + XQ_0'Q_1'$

$Z = XQ_1Q_0$

b.

X	Q_1Q_0	$Q_1^+Q_0^+$	Z
0	00	00	0
0	01	11	0
0	11	00	0
0	10	00	0
1	00	10	0
1	01	01	0
1	11	10	1
1	10	01	0

c.

X	Present State	Next State	Z
0	S_0	S_0	0
0	S_1	S_3	0
0	S_2	S_0	0
0	S_3	S_0	0
1	S_0	S_2	0
1	S_1	S_1	0
1	S_2	S_1	0
1	S_3	S_2	1

d.

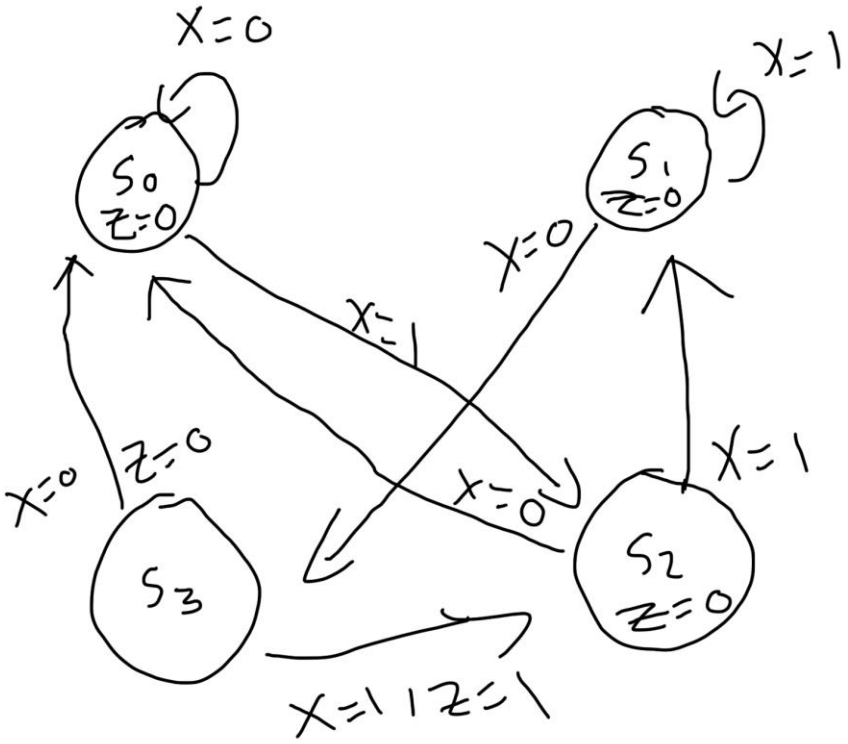


Figure 2: State Diagram

Assuming 000 state zero is the reset point then the circuit detects the input sequence 1101 or the user could input 1 for n amount of times and eventually put a 0 for the circuit to detect the sequence since an x input of 1 at S_1 will cause it to loop. Regardless, the circuit detects an input of 1101 even if the user loops at state 1.

Activity 6.2 — Synthesis of a Finite State Machine

a.

A shift register would be a faulty desing for the level to pulse converter since the output would stay at logic level 1 if the input is at logic one for more than two clock cycles, hoiwever our design states that we want the output to only go to logic one if the input is at logic one for two clock cycles and is then at logic level 0 until the 011 combination is reentered.

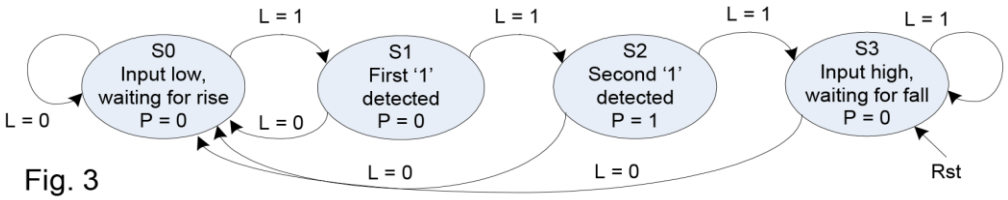


Figure 3: State Diagram

b.

Input L	Present State	Next State	Output P
0	S_0	S_0	0
0	S_1	S_0	0
0	S_2	S_0	1
0	S_3	S_0	0
1	S_0	S_1	0
1	S_1	S_2	0
1	S_2	S_3	1
1	S_3	S_3	0

c.

Input L	Present State Q_1Q_0	Next State $Q_1^+Q_0^+$	Output P
0	00	00	0
0	01	00	0
0	11	00	1
0	10	00	0
1	00	01	0
1	01	11	0
1	11	10	1
1	10	10	0

d.

L \ Q_1Q_0	00	01	11	10
0	0	0	0	0
1	0	1	1	1

$Q_1^+ = LQ_0 + LQ_1$

L \ Q_1Q_0	00	01	11	10
0	0	0	0	0
1	1	1	0	0

$Q_0^+ = LQ_1'$

L \ Q_1Q_0	00	01	11	10
0	0	0	1	0
1	0	0	1	0

$P = Q_1Q_0$

e.

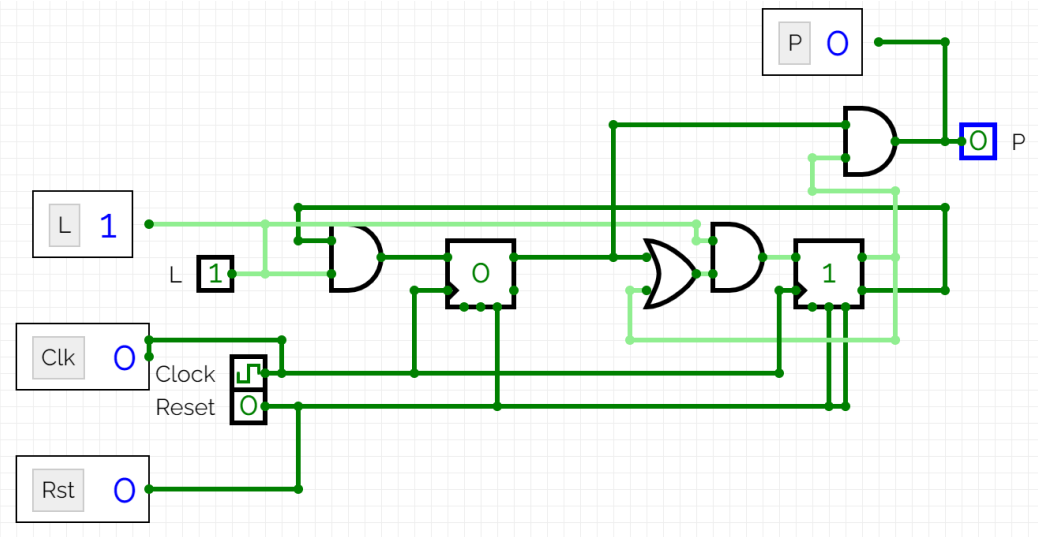


Figure 4: Circuit Diagram

f.

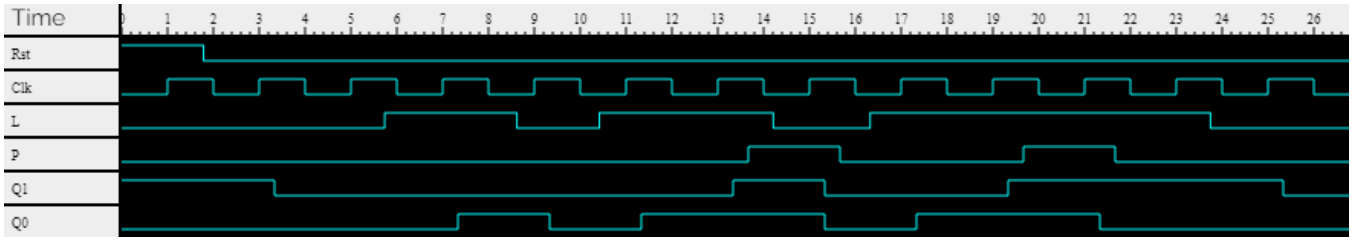


Figure 5: Timing Diagram

Activity 6.3 — Game Show FSM

Present State $Q_1\ Q_0$	Next State $Q_1^+\ Q_0^+$ $B_1\ B_0$ = 00	01	10	11	Outputs $Z_1\ Z_0$
0 0 (Ready)	00	01	10	11	0 0
0 1 (Win0)	01	01	01	01	0 1
1 0 (Win1)	10	10	10	10	1 0
1 1 (Tie)	11	11	11	11	0 0

Figure 6: State/ Transition Table of Game Show FSM

a.

Figure 7: State Diagram

b.

B_1B_0 \ Q_1Q_0	00	01	11	10
00	0	0	1	1
01	0	0	1	1
11	1	0	1	1
10	1	0	1	1

$Q_1^+ = Q_1 + B_1Q_0'$

B_1B_0 \ Q_1Q_0	00	01	11	10
00	0	1	1	0
01	1	1	1	0
11	1	1	1	0
10	0	1	1	0

$Q_0^+ = Q_0 + B_0Q_1'$

B_1B_0 \ Q_1Q_0	00	01	11	10
00	0	0	0	1
01	0	0	0	1
11	0	0	0	1
10	0	0	0	1

$Z_1 = Q_1Q_0'$

B_1B_0 \ Q_1Q_0	00	01	11	10
00	0	1	0	0
01	0	1	0	0
11	0	1	0	0
10	0	1	0	0

$Z_0 = Q_0Q_1'$

e.

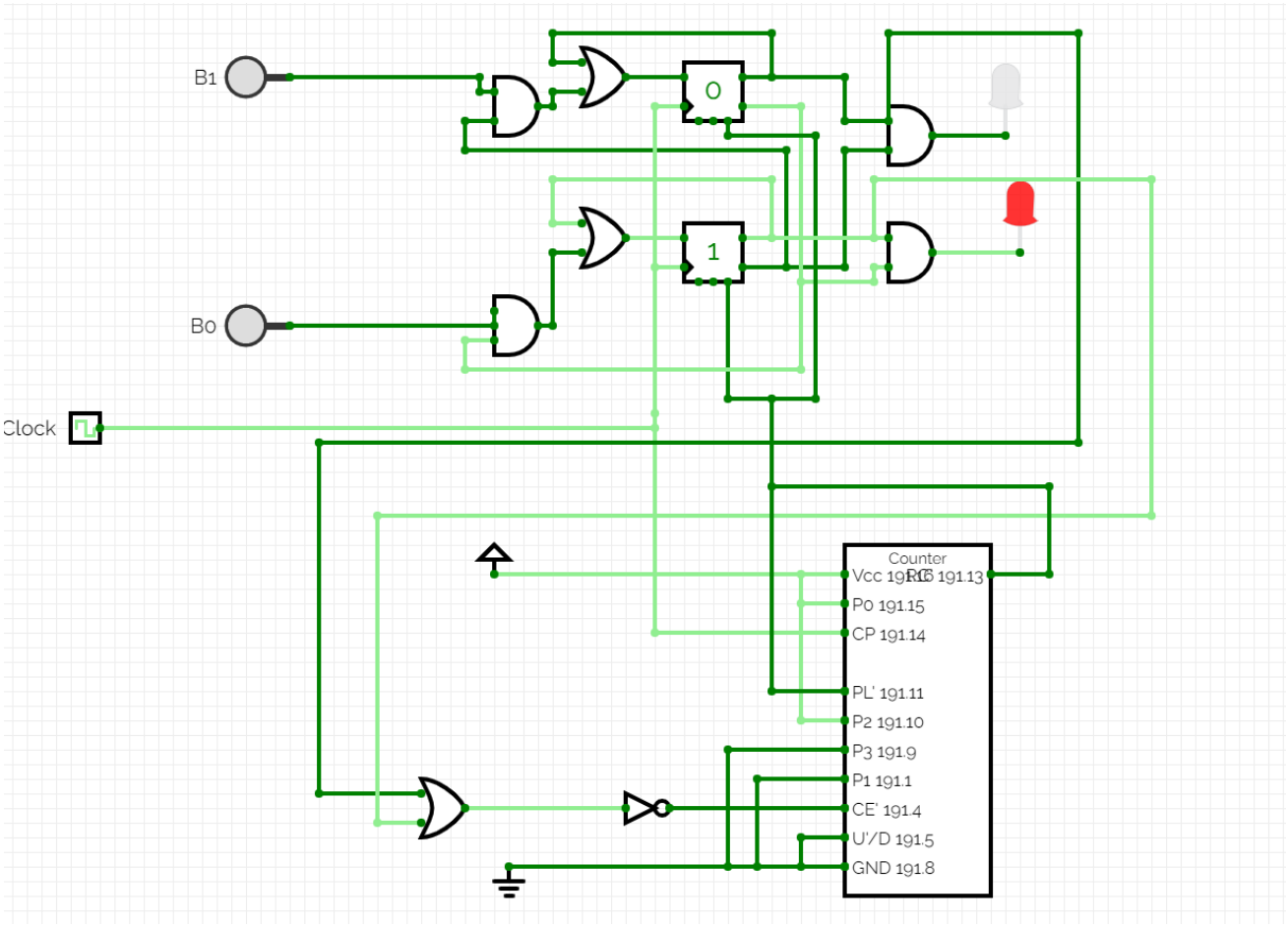


Figure 8: CircuitVerse Diagram

Note: The Counter block has no logic in it and has the pins of the physical boards, and CircuitVerse allowed the names to crunch but the top nodes are VCC and RC'.

f. Note: I do explain the circuit more in the video done for part C, see bonus at the end, whereas in this video I spend a little more time on the NOT gate and the counter I added.

Video: https://youtu.be/NDcmNd1l1ZA

Exercise 6.1 — Activity 6.2 Revisited

a.

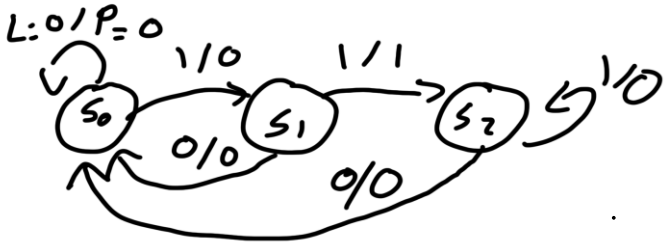


Figure 9: State Diagram

b.

Input L	Present State	Next State	Output P
0	S_0 (00)	S_0	0
0	S_1 (01)	S_0	0
0	S_2 (10)	S_0	0
1	S_0	S_1	0
1	S_1	S_2	1
1	S_2	S_2	0

c.

Input L	Present State Q_1Q_0	Next State $Q_1^+Q_0^+$	Output P
0	00	00	0
0	01	00	0
0	10	00	0
1	00	01	0
1	01	10	1
1	10	10	0

d.

L \ Q_1Q_0	00	01	11	10
0	0	0	X	0
1	0	1	X	1

$Q_1^+ = LQ_0 + LQ_1$

L \ Q_1Q_0	00	01	11	10
0	0	0	X	0
1	1	0	X	0

$Q_0^+ = LQ_1' Q_0'$

L \ Q_1Q_0	00	01	11	10
0	0	0	X	0
1	0	1	X	0

$P = LQ_0$

e.

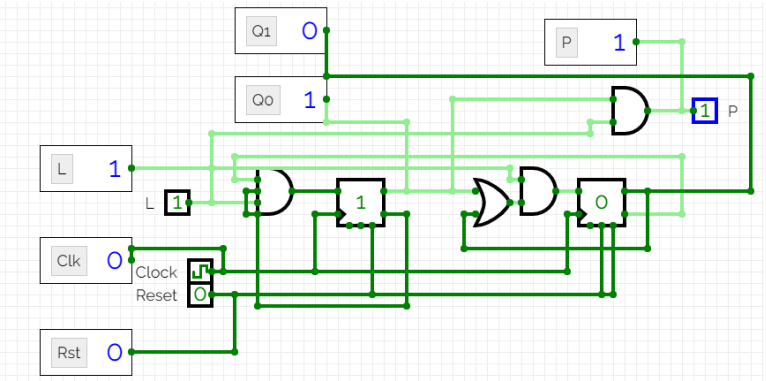


Figure 10: Diagram In CircuitVerse

Exercise 6.2 — An Unconventional FSM

a.

Z is the output, with 1 meaning the number is divisible by 3 and I is the input.

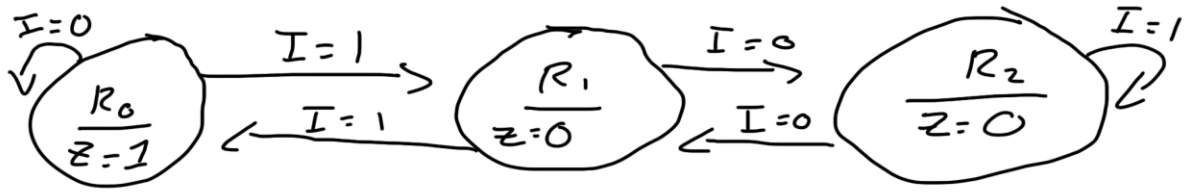


Figure 11: State Diagram

b.

Input L	Present State	Next State	Output Z
0	R_0	R_0	1
0	R_1	R_2	0
0	R_2	R_1	0
1	R_0	R_1	1
1	R_1	R_0	0
1	R_2	R_2	0

c.

State Assignments: $R_0 = 00$ $R_1 = 01$ $R_2 = 10$

Input I	Present State Q_1Q_0	Next State $Q_1^+Q_0^+$	Output Z
0	00	00	1
0	01	10	0
0	10	01	0
1	00	01	1
1	01	00	0
1	10	10	0

d.

I \ Q_1Q_0	00	01	11	10
0	0	1	X	0
1	0	0	X	1

$Q_1^+ = I'Q_0 + IQ_1$

I \ Q_1Q_0	00	01	11	10
0	0	0	X	1
1	1	0	X	0

$Q_0^+ = IQ_1'Q_0' + I'Q_1$

I \ Q_1Q_0	00	01	11	10
0	1	0	X	0
1	1	0	X	0

$Z = Q_1'Q_0'$

e.

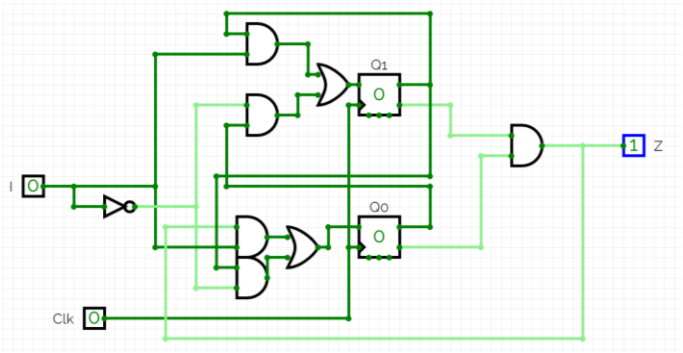


Figure 12: Circuit Diagram In CircuitVerse

f.

Label	I	Z
BitWidth	1	1
1	0	X
2	1	1
3	1	0
4	0	1
5	1	1
6	0	0
7	1	0
8	0	0
9	1	0
10	0	1
11	1	1

Figure 13: Testbench

g.

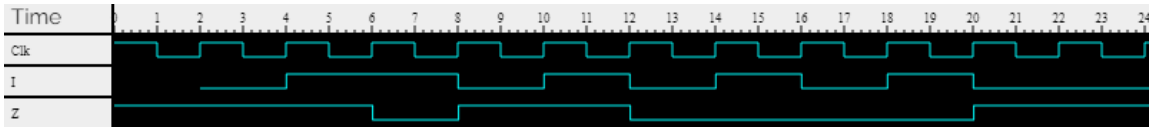


Figure 14: Waveform For Output Of TestBench

Note: I reset the flip flops to 0 and the testbench to prevent any previous tests from affecting the next circuit testing.

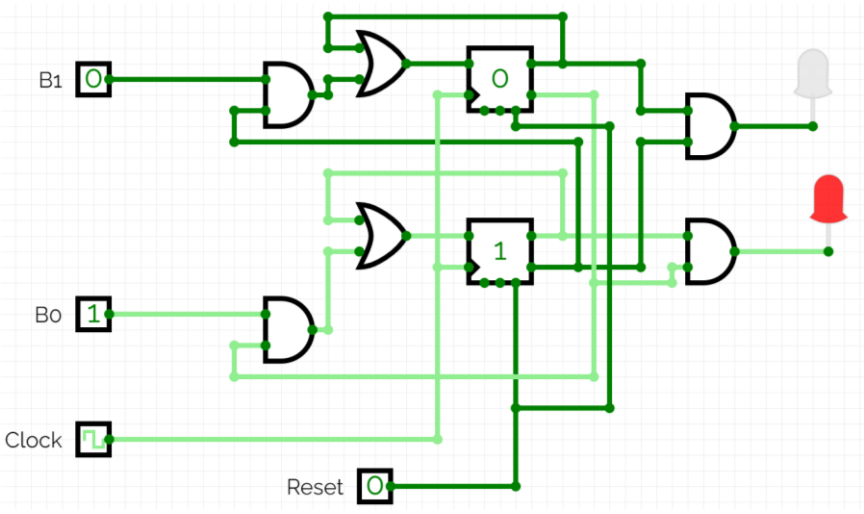
h.

Any integer can be written as 3n + R where n is an integer and R is the integer 0, 1, or 2. From this we know that if we our number is divisible by 3, R = 0, and if not will either equal 1 or 2. When a 0 is shifted into the bitstream, the number is multiplied by two which will result in the number still falling in the form of 3n + R since if R is equal to 2, it will produce a number greater than two which will be absorbed into the integer n representing the number along with the 2 multiplied to the number. This will result in a remainder that is still between 0 and 2 and a number that still falls within the three states defined at the beginning. Similarly, if a one is shifted in then the number is multiplied by 2 and then a one is added. At the end of the multiplication process, we will have a number in the form of 3n+R, and when we add the 1 the number will still be in this form since if R=2, the resulting sum will be 3 which would then be absorbed into the 3n form and leave a R = 0. Due to this regardless of what bit stream we put into the FSM we can write the number in the form 3n + R and this will allow us to use the states when R = 0, R =1, and R = 2, since n does not change whether the input stream is divisible by 3 or not, only R does.

Activity 6.3 — Bonus

I had made this before finsihing the circuit witht eh counter and is at the end in case you needed to look at it for any reason.

c.



d.

Video: https://youtu. be/yQRnBJOgyWk