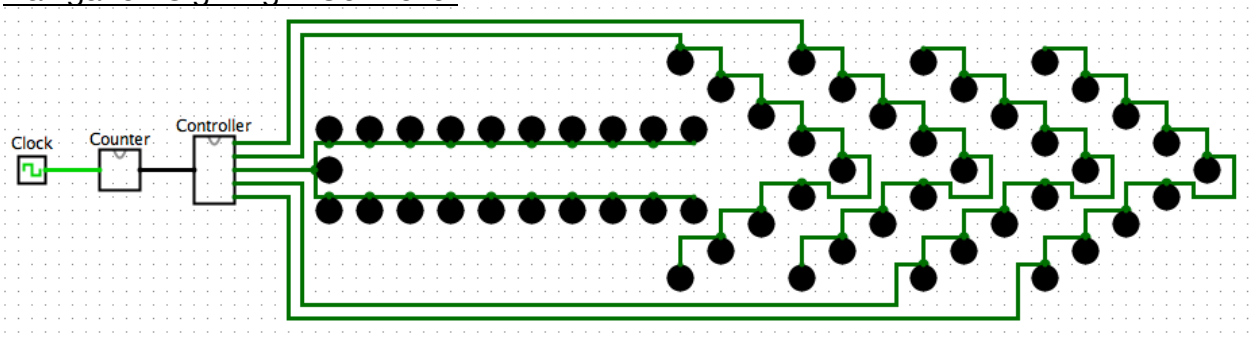
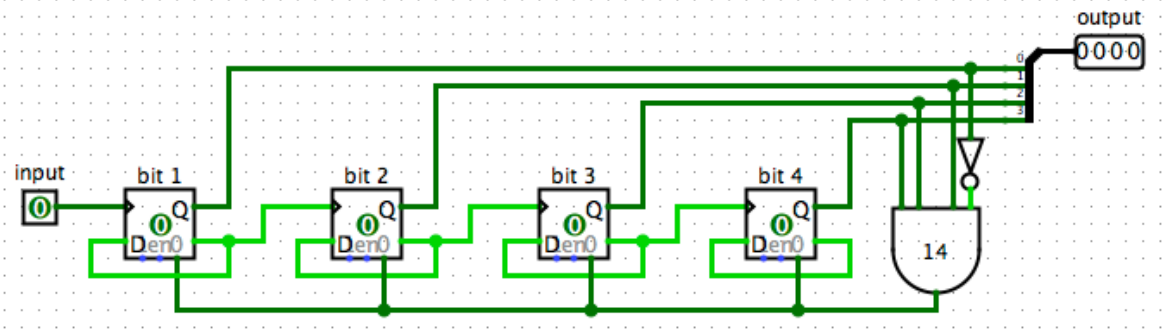
CPSC 359 T06 W17 Assignment 1
Richard Truong 10058625

Navigation Sign Light Controller



4-bit Counter



1. Steps

The counter consists of an input pin and an output pin. The input is a single bit from the clock and the output is four bits (condensed through a splitter). To build the counter, four D-Flipflops were used (each representing a single bit in its output "Q"). The arrangement of bits is from left to right rather than right to left due to certain design choices.

The D-Flipflop transitions on each rising edge of the clock (input of 1) and the change is dependent on the value of the input "D". This is always the opposite of the current state Q, so that the Flipflop is alternating between 0 and 1 consistently. Therefore, ¬ Q is rerouted back into itself as shown above.

Additionally, ¬ Q is also wired to the following D-Flipflop to indicate when that D-Flipflop should change. This means that the value of the following D-Flipflop should change to 1 on the first instance when the previous D-Flipflop changes back to 0 and thereafter. Consequently, the following D-Flipflop will also change to 0 when the previous changes to 1. Here it is important to note that not all the D-Flipflops will respond to the rising edge of the clock every time.

Finally, a regular four-bit counter would count up to 1111 (15) but we only need the counter to count up to 1101 (13). This is done by tapping into the all the Q values from the D-Flipflops before it reaches the counter's output pin and detecting 1101 (14) with an AND gate. The AND gate is wired directly to all of the D-Flipflops' reset pin and sets them to 1 when 14 is reached.

CPSC 359 T06 W17 Assignment 1
Richard Truong 10058625

4-bit Counter

2. Truth Table

CLOCK INPUT					BIT 4			BIT 3			BIT 2			BIT 1			COUNTER OUTPUT				
Clock	W	X	Y	Z	Q	¬Q	D	Q	¬Q	D	Q	¬Q	D	Q	¬Q	D	Count	W	X	Y	Z
1	0	0	0	1	0	1	1	0	1	1	0	1	1	1	0	0	1	0	0	0	1
2	0	0	1	0	0	1	1	0	1	1	1	0	0	0	1	1	2	0	0	1	0
3	0	0	1	1	0	1	1	0	1	1	1	0	0	1	0	0	3	0	0	1	1
4	0	1	0	0	0	1	1	1	0	0	0	1	1	0	1	1	4	0	1	0	0
5	0	1	0	1	0	1	1	1	0	0	0	1	1	1	0	0	5	0	1	0	1
6	0	1	1	0	0	1	1	1	0	0	1	0	0	0	1	1	6	0	1	1	0
7	0	1	1	1	0	1	1	1	0	0	1	0	0	1	0	0	7	0	1	1	1
8	1	0	0	0	1	0	0	0	1	1	0	1	1	0	1	1	8	1	0	0	0
9	1	0	0	1	1	0	0	0	1	1	0	1	1	1	0	0	9	1	0	0	1
10	1	0	1	0	1	0	0	0	1	1	1	0	0	0	1	1	10	1	0	1	0
11	1	0	1	1	1	0	0	0	1	1	1	0	0	1	0	0	11	1	0	1	1
12	1	1	0	0	1	0	0	1	0	0	0	1	1	0	1	1	12	1	1	0	0
13	1	1	0	1	1	0	0	1	0	0	0	1	1	1	0	0	13	1	1	0	1
14	1	1	1	0	0	1	1	0	1	1	0	1	1	0	1	1	0	0	0	0	0

As described above, each bit value Q is changed to the value of D (¬ Q). This occurs due to the rising edge of the clock for the first bit, and the negation of Q of the previous bit for following bits. The following bits begin to change only after the previous bit has changed from 1 to 0 for the first time.

1. Boolean Functions

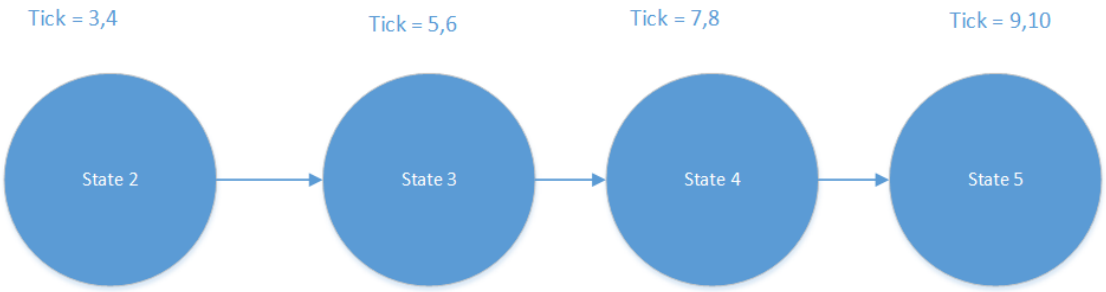
Fc (w, x, y, z) = $\sum(0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13)$ =
w'x'y'z' + w'x'y'z + w'x'yz' + w'x'yz + w'xy'z' + w'xy'z + w'xyz' +
w'xyz + wx'y'z' + wx'y'z + wx'yz' + wx'yz + wxy'z' + wxy'z

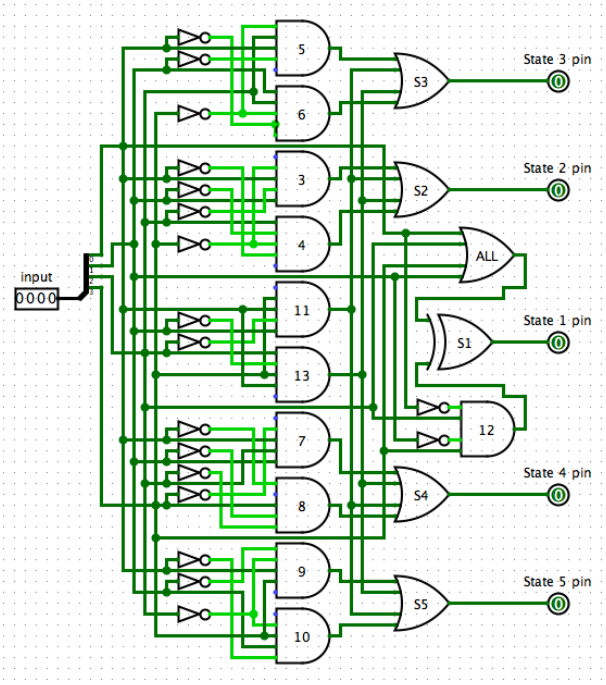
2. Simplification

Fc (w, x, y, z) = $\underline{m_1 + m_3} + \underline{m_3 + m_2} + \underline{m_4 + m_5} + \underline{m_5 + m_7} + \underline{m_7 + m_6} + \underline{m_6 + m_4} +$
$\underline{m_5 + m_1} + \underline{m_3 + m_7} + \underline{m_2 + m_6} + \underline{m_5 + m_{13}} + \underline{m_8 + m_9} + \underline{m_9 + m_{11}} +$
$\underline{m_{10} + m_{11}} + \underline{m_{13} + m_9} + \underline{m_9 + m_1} + \underline{m_{11} + m_3} + \underline{m_{10} + m_2} + \underline{m_{12} + m_4}$
$+ \underline{m_{12} + m_3} + \underline{m_{12} + m_8} + \underline{m_0 + m_1} + \underline{m_0 + m_2} + \underline{m_0 + m_4} + \underline{m_0 + m_8}$
= w' + w'z' + w'x + w'xz' + w'z' + wx'z + wy + wxy + x'y + x'z' +
xy'z

CPSC 359 T06 W17 Assignment 1
Richard Truong 10058625

Controller





3. Steps

The controller consists of a single pin for a 4-bit input and five 1-bit output pins. The input is split into four 1-bit values; each is spliced several times to provide control over each of the five LED states.

The detection of a 0 in a bit is done by negating the wiring for the specified bit before inputting it to a gate. Detecting is 1 simply requires directly wiring the bit to the input of the gate.

As shown above, each LED state consists of two conditions that dictate when the state is ON. State 1 also has two conditions: it is OFF for tick 0 and 12.

Additionally, all States are ON for tick 11 and 13. This is implemented into all the controllers of each State.

The detection of ticks is done using AND gates for every tick count from 3 to 13. The States are activated using OR gates which assess necessary tick count criteria for each State. The OR gates are then as the output pins of the controller to connect to the corresponding sets of LEDs.

CPSC 359 T06 W17 Assignment 1
Richard Truong 10058625

4. Truth Table

COUNTER INPUT					STATE 1					STATE 2					STATE 3					STATE 4					STATE 5				
Clock	W	X	Y	Z	W	X	Y	Z	Status	W	X	Y	Z	Status	W	X	Y	Z	Status	W	X	Y	Z	Status	W	X	Y	Z	Status
0	0	0	0	0	0	0	0	0	OFF	0	0	0	0	OFF	0	0	0	0	OFF	0	0	0	0	OFF	0	0	0	0	OFF
1	0	0	0	1	0	0	0	1	ON	0	0	0	1	OFF	0	0	0	1	OFF	0	0	0	1	OFF	0	0	0	1	OFF
2	0	0	1	0	0	0	1	0	ON	0	0	1	0	OFF	0	0	1	0	OFF	0	0	1	0	OFF	0	0	1	0	OFF
3	0	0	1	1	0	0	1	1	ON	0	0	1	1	ON	0	0	1	1	OFF	0	0	1	1	OFF	0	0	1	1	OFF
4	0	1	0	0	0	1	0	0	ON	0	1	0	0	ON	0	1	0	0	OFF	0	1	0	0	OFF	0	1	0	0	OFF
5	0	1	0	1	0	1	0	1	ON	0	1	0	1	OFF	0	1	0	1	ON	0	1	0	1	OFF	0	1	0	1	OFF
6	0	1	1	0	0	1	1	0	ON	0	1	1	0	OFF	0	1	1	0	ON	0	1	1	0	OFF	0	1	1	0	OFF
7	0	1	1	1	0	1	1	1	ON	0	1	1	1	OFF	0	1	1	1	OFF	0	1	1	1	ON	0	1	1	1	OFF
8	1	0	0	0	1	0	0	0	ON	1	0	0	0	OFF	1	0	0	0	OFF	1	0	0	0	ON	1	0	0	0	OFF
9	1	0	0	1	1	0	0	1	ON	1	0	0	1	OFF	1	0	0	1	OFF	1	0	0	1	OFF	1	0	0	1	ON
10	1	0	1	0	1	0	1	0	ON	1	0	1	0	OFF	1	0	1	0	OFF	1	0	1	0	OFF	1	0	1	0	ON
11	1	0	1	1	1	0	1	1	ON	1	0	1	1	ON	1	0	1	1	ON	1	0	1	1	ON	1	0	1	1	ON
12	1	1	0	0	1	1	0	0	OFF	1	1	0	0	OFF	1	1	0	0	OFF	1	1	0	0	OFF	1	1	0	0	OFF
13	1	1	0	1	1	1	0	1	ON	1	1	0	1	ON	1	1	0	1	ON	1	1	0	1	ON	1	1	0	1	ON

To summarize, State 2 is ON when the tick count is either 3, 4, 11, or 13. State 3 is ON when the tick count is either 5, 6, 11, or 13. State 4 is ON when the tick count is either 7,8, 11, or 13. State 5 is ON when the tick count is either 9, 10, 11, or 13. The states above use a single OR gate with 4 inputs each. Each of these inputs come from the previous AND gates which detect each specific tick counts.

Additionally, State 1 is only OFF when the tick count is 0 or 12. An OR gate for all four bits would set State 1 to OFF when it is 0000 (0). Then an AND gate is used to detect 1100 (12). Now, these two gates are piped through an XOR gate to achieve the truth table for State 1. This XOR gate sets State 1 LEDs to off when the tick count is 0 and 12.

CPSC 359 T06 W17 Assignment 1
Richard Truong 10058625

5. Boolean Functions

F1 (w, x, y, z) = $\sum(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 13)$
= w'x'y'z + w'x'yz' + w'x'yz + w'xy'z' + w'xy'z + w'xyz' + w'xyz +
wx'y'z' + wx'y'z + wx'yz' + wx'yz + wxy'z

F2 (w, x, y, z) = $\sum(3, 4, 11, 13)$ = w'x'yz + w'xy'z' + wx'yz + wxy'z

F3 (w, x, y, z) = $\sum(5, 6, 11, 13)$ = w'xy'z + w'xyz' + wx'yz + wxy'z

F4 (w, x, y, z) = $\sum(7, 8, 11, 13)$ = w'xyz + wx'y'z' + wx'yz + wxy'z

F5 (w, x, y, z) = $\sum(9, 10, 11, 13)$ = wx'y'z + wx'yz' + wx'yz + wxy'z

6. Simplification

F1 (w, x, y, z) = $\underline{m_1 + m_3} + \underline{m_3 + m_2} + \underline{m_4 + m_5} + \underline{m_5 + m_7} + \underline{m_7 + m_6} + \underline{m_6 + m_4} +$
$\underline{m_5 + m_1} + \underline{m_3 + m_7} + \underline{m_2 + m_6} + \underline{m_5 + m_{13}} + \underline{m_8 + m_9} + \underline{m_9 + m_{11}} +$
$\underline{m_{10} + m_{11}} + \underline{m_{13} + m_9} + \underline{m_9 + m_1} + \underline{m_{11} + m_3} + \underline{m_{10} + m_2}$
= x'y + x'yz' +xy'z + w'z + w'x + w'yz' + wx'z + wy'z

F2 (w, x, y, z) = w'x'yz + w'xy'z' + wx'yz + wxy'z

F3 (w, x, y, z) = w'x'yz + w'xy'z' + wx'yz + wxy'z
= $\underline{m_5 + m_{13}} + m_6 + m_{11}$
= xy'z(w' + w) + w'xyz' + wx'yz
= xy'z(w' + w) + w'xyz' + wx'yz
= xy'z + w'xyz' + wx'yz

F4 (w, x, y, z) = w'xyz + wx'y'z' + wx'yz + wxy'z

F5 (w, x, y, z) = wx'y'z + wx'yz' + wx'yz + wxy'z
= $\underline{m_{13} + m_9} + \underline{m_9 + m_{11}} + \underline{m_{11} + m_{10}}$
= wy'z + wx'z + wx'y