Assignment 1: Navigation Sign Light Controller

CPSC 359 - Winter 2017

BenKun Chen(30005337)

Jiongkai Jiang(101387431)

01/24/2017

Introduction:

In this assignment, we will design and implement a sequential circuit that simulates a flashing traffic navigation sign for drivers.

Counter:

Truth Table:

Current State				**Next State**			
S_0	S_1	S_2	S_3	$\mathbf{S_0^*}$	$\mathbf{S_1^*}$	$\mathbf{S_2^*}$	$\mathbf{S_3^*}$
0	0	0	0	**0**	**0**	**0**	**1**
0	0	0	1	**0**	**0**	**1**	**0**
0	0	1	0	**0**	**0**	**1**	**1**
0	0	1	1	**0**	**1**	**0**	**0**
0	1	0	0	**0**	**1**	**0**	**1**
0	1	0	1	**0**	**1**	**1**	**0**
0	1	1	0	**0**	**1**	**1**	**1**
0	1	1	1	**1**	**0**	**0**	**0**
1	0	0	0	**1**	**0**	**0**	**1**
1	0	0	1	**1**	**0**	**1**	**0**
1	0	1	0	**1**	**0**	**1**	**1**
1	0	1	1	**1**	**1**	**0**	**0**
1	1	0	0	**1**	**1**	**0**	**1**
1	1	0	1	**0**	**0**	**0**	**0**

Logical expression:

$C0^* = x'y'z'u' + x'y'zu' + x'yz'u' + x'yzu' + xy'z'u' + xy'zu' + xyz'u'$

$C1^* = x'y'z'u + x'y'zu' + x'yz'u + x'yzu' + xy'z'u + xy'zu'$

$C2^* = x'y'zu + x'yz'u' + x'yz'u + x'yzu' + xy'zu + xyz'u'$

$C3^* = x'yzu + xy'z'u' + xy'z'u + xy'zu' + xy'zu + xyz'u'$

Note: x, y, z, u stand for S0, S1, S2, S3

Controller:

Truth Table for Controller:

I0	I1	I2	I3	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14	S15
0	0	0	0	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	0	0	1	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	0	1	0	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	0	1	1	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	1	0	0	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	1	0	1	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	1	1	0	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
0	1	1	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	0	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
1	0	0	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**	**0**
1	0	1	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**	**0**
1	0	1	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**	**0**
1	1	0	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**0**
1	1	0	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**	**0**
1	1	1	0	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**	**0**
1	1	1	1	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**1**

Logical expression:

S0 = x'y'z'u'

S1 = x'y'z'u

S2 = x'y'zu'

S3 = x'y'zu

S4 = x'yz'u'

S5 = x'yz'u

S6 = x'yzu'

S7 = x'yzu

S8 = xy'z'u'

S9 = xy'z'u

S10 = xy'zu'

S11 = xy'zu

S12 = xyz'u'

S13 = xyz'u

S14 = xyzu'

S15 = xyzu

Note: x, y, z, u stand for I0, I1, I2, I3

After decoding the address, the outputs will be distributed into single or multiple "OR" logical gates in order to achieve the sequence in which the arrow is animated. Since the state one will keep litting before tick 13, the "S1" output has been connected to the same "OR" gate as tick 2 to tick 13. Also, all the states would be lit at tick 11 and 13, so the "S11" and "S13" outputs are connected to all "OR" gates. In order to trigger state 2, "S3" and "S4" should connect with the same "OR" gate. Similarly, state 3 have to link with "S5" and "S6" by using one "OR" gate. Also, "S7" and "S8 " must be connected with the same "OR" gate. Finally, the sate 5 have to link with "S9" and "S10" by an "OR" gate. Therefore, we can accomplish the multi-function.