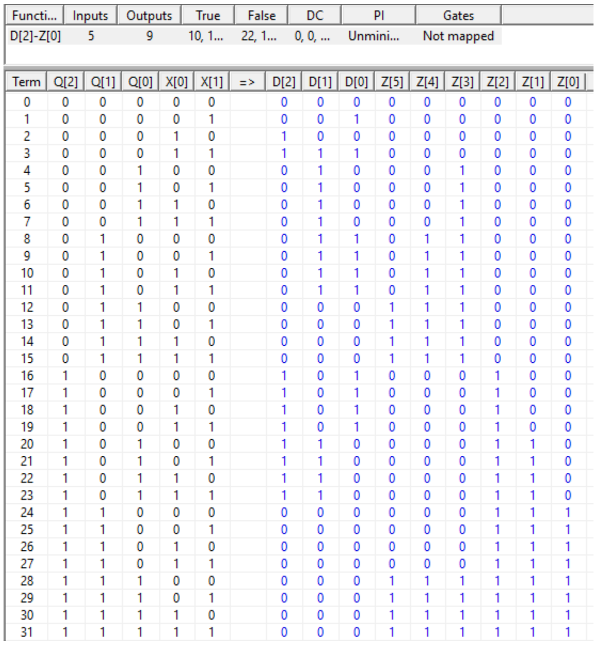
Statement of the Problem

The purpose of this lab is to create a sequential taillight controller. These types of taillight controllers can be found in cars such as the Ford Mustangs. Students will develop a state machine in order to differentiate idle, left, and right signals. They will implement the synchronous state machine into Quartus Lite. It will then b tested using an FPGA board and waveforms.

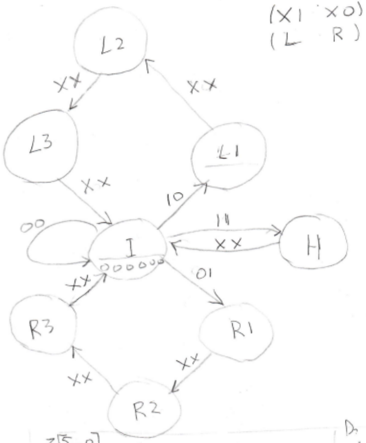
Description of Solution/Design



Functi...	Inputs	Outputs	True	False	DC	PI	Gates
D[2]-Z[0]	5	9	10, 1...	22, 1...	0, 0, ...	Unmini...	Not mapped

Term	Q[2]	Q[1]	Q[0]	X[0]	X[1]	=>	D[2]	D[1]	D[0]	Z[5]	Z[4]	Z[3]	Z[2]	Z[1]	Z[0]
0	0	0	0	0	0		0	0	0	0	0	0	0	0	0
1	0	0	0	0	1		0	0	1	0	0	0	0	0	0
2	0	0	0	1	0		1	0	0	0	0	0	0	0	0
3	0	0	0	1	1		1	1	1	0	0	0	0	0	0
4	0	0	1	0	0		0	1	0	0	0	1	0	0	0
5	0	0	1	0	1		0	1	0	0	0	1	0	0	0
6	0	0	1	1	0		0	1	0	0	0	1	0	0	0
7	0	0	1	1	1		0	1	0	0	0	1	0	0	0
8	0	1	0	0	0		0	1	1	0	1	1	0	0	0
9	0	1	0	0	1		0	1	1	0	1	1	0	0	0
10	0	1	0	1	0		0	1	1	0	1	1	0	0	0
11	0	1	0	1	1		0	1	1	0	1	1	0	0	0
12	0	1	1	0	0		0	0	0	1	1	1	0	0	0
13	0	1	1	0	1		0	0	0	1	1	1	0	0	0
14	0	1	1	1	0		0	0	0	1	1	1	0	0	0
15	0	1	1	1	1		0	0	0	1	1	1	0	0	0
16	1	0	0	0	0		1	0	1	0	0	0	1	0	0
17	1	0	0	0	1		1	0	1	0	0	0	1	0	0
18	1	0	0	1	0		1	0	1	0	0	0	1	0	0
19	1	0	0	1	1		1	0	1	0	0	0	1	0	0
20	1	0	1	0	0		1	1	0	0	0	0	1	1	0
21	1	0	1	0	1		1	1	0	0	0	0	1	1	0
22	1	0	1	1	0		1	1	0	0	0	0	1	1	0
23	1	0	1	1	1		1	1	0	0	0	0	1	1	0
24	1	1	0	0	0		0	0	0	0	0	0	1	1	1
25	1	1	0	0	1		0	0	0	0	0	0	1	1	1
26	1	1	0	1	0		0	0	0	0	0	0	1	1	1
27	1	1	0	1	1		0	0	0	0	0	0	1	1	1
28	1	1	1	0	0		0	0	0	1	1	1	1	1	1
29	1	1	1	0	1		0	0	0	1	1	1	1	1	1
30	1	1	1	1	0		0	0	0	1	1	1	1	1	1
31	1	1	1	1	1		0	0	0	1	1	1	1	1	1

(X1 X0) > 00
(L   R)   01
10
11

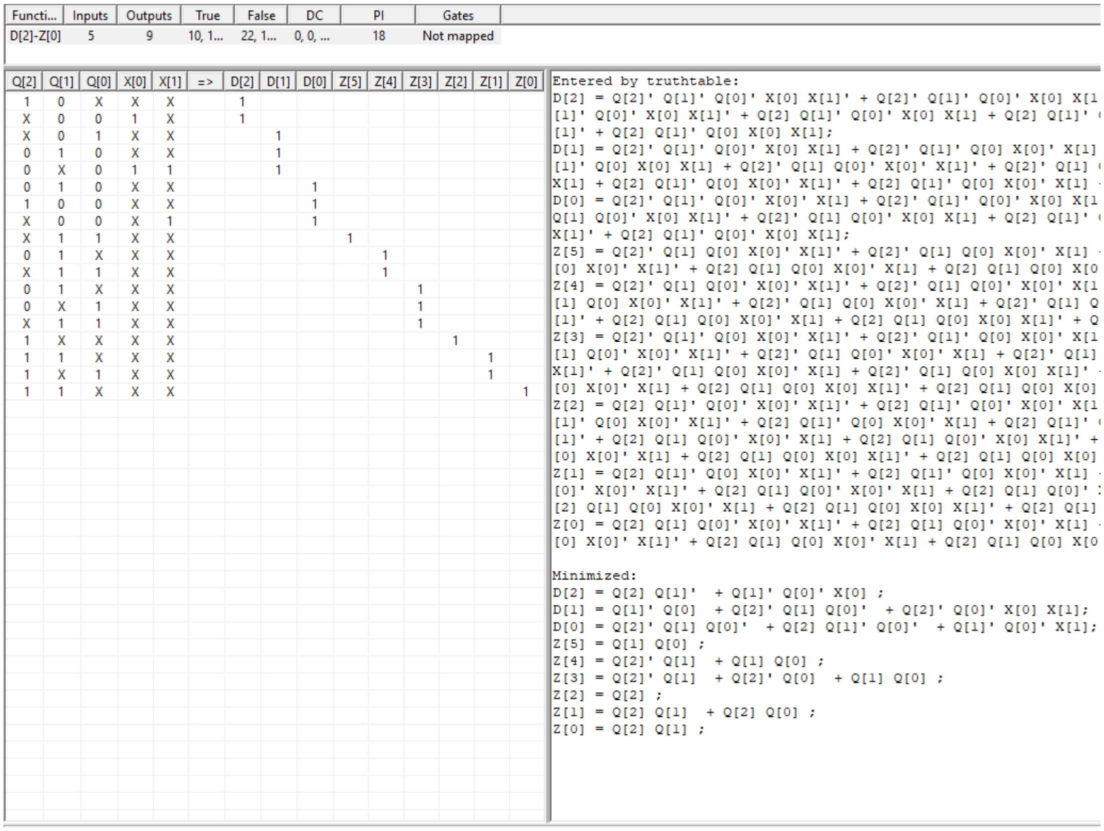


P.S.	N.S				Z[5..0]
	X1X0				
Q2Q1Q0	00	01	10	11	
000 I	000 I	100 R1	001 L1	111 H	000 000
001 L1	010 L2	010 L2	010 L2	010 L2	001 000
010 L2	011 L3	011 L3	011 L3	011 L3	011 000
011 L3	000 I	000 I	000 I	000 I	111 000
100 R1	101 R2	101 R2	101 R2	101 R2	000 100
101 R2	110 R3	110 R3	110 R3	110 R3	000 110
110 R3	000 I	000 I	000 I	000 I	000 111
111 H	000 I	000 I	000 I	000 I	111 111

	D_2 D_1 D_0
$Q_2Q_1Q_0X_1X_0$	$Q_2^+Q_1^+Q_0^+$
0 0 0 00	0 0 0 000000
0 0 0 01	1 0 0 000000
0 0 0 10	
0 0 0 11	

Z output table in the slides

I 000000.
L1 001000
etc.

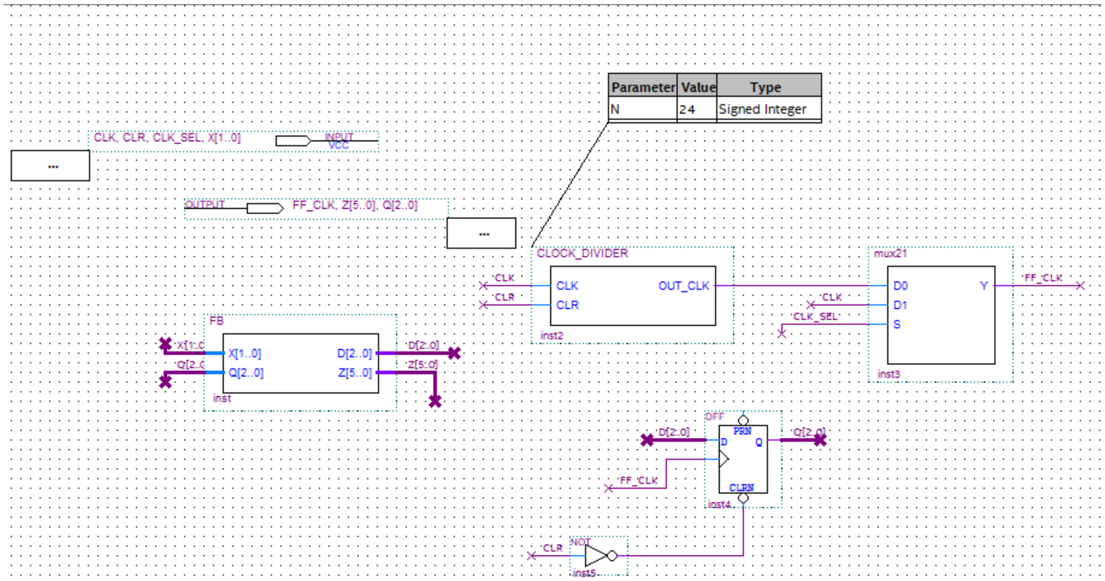


Functi...	Inputs	Outputs	True	False	DC	PI	Gates
D[2]-Z[0]	5	9	10, 1...	22, 1...	0, 0, ...	18	Not mapped

Q[2]	Q[1]	Q[0]	X[0]	X[1]	=>	D[2]	D[1]	D[0]	Z[5]	Z[4]	Z[3]	Z[2]	Z[1]	Z[0]
1	0	X	X	X		1								
X	0	0	1	X		1								
X	0	1	X	X			1							
0	1	0	X	X			1							
0	X	0	1	1			1							
0	1	0	X	X				1						
1	0	0	X	X				1						
X	0	0	X	1				1						
X	1	1	X	X					1					
0	1	X	X	X						1				
X	1	1	X	X						1				
0	1	X	X	X							1			
0	X	1	X	X							1			
X	1	1	X	X							1			
1	X	X	X	X								1		
1	1	X	X	X									1	
1	X	1	X	X									1	
1	1	X	X	X										1

```
Entered by truthtable:
D[2] = Q[2]' Q[1]' Q[0]' X[0] X[1]' + Q[2]' Q[1]' Q[0]' X[0] X[1
[1]' Q[0]' X[0] X[1]' + Q[2] Q[1]' Q[0]' X[0] X[1] + Q[2] Q[1]'
[1]' + Q[2] Q[1]' Q[0] X[0] X[1];
D[1] = Q[2]' Q[1]' Q[0]' X[0] X[1] + Q[2]' Q[1]' Q[0] X[0]' X[1]
[1]' Q[0] X[0] X[1] + Q[2]' Q[1] Q[0]' X[0]' X[1]' + Q[2]' Q[1]
X[1] + Q[2] Q[1]' Q[0] X[0]' X[1]' + Q[2] Q[1]' Q[0] X[0]' X[1]
D[0] = Q[2]' Q[1]' Q[0]' X[0]' X[1] + Q[2]' Q[1]' Q[0]' X[0] X[1
Q[1] Q[0]' X[0] X[1]' + Q[2]' Q[1] Q[0]' X[0] X[1] + Q[2] Q[1]'
X[1]' + Q[2] Q[1]' Q[0]' X[0] X[1];
Z[5] = Q[2]' Q[1] Q[0] X[0]' X[1]' + Q[2]' Q[1] Q[0] X[0]' X[1]
[0] X[0]' X[1]' + Q[2] Q[1] Q[0] X[0]' X[1] + Q[2] Q[1] Q[0] X[0
Z[4] = Q[2]' Q[1] Q[0]' X[0]' X[1]' + Q[2]' Q[1] Q[0]' X[0]' X[1
[1] Q[0] X[0]' X[1]' + Q[2]' Q[1] Q[0] X[0]' X[1] + Q[2]' Q[1] Q
[1]' + Q[2] Q[1] Q[0] X[0]' X[1] + Q[2] Q[1] Q[0] X[0] X[1]' + Q
Z[3] = Q[2]' Q[1]' Q[0] X[0]' X[1]' + Q[2]' Q[1]' Q[0] X[0]' X[1
[1] Q[0]' X[0]' X[1]' + Q[2]' Q[1] Q[0]' X[0]' X[1] + Q[2]' Q[1]
X[1]' + Q[2]' Q[1] Q[0] X[0]' X[1] + Q[2]' Q[1] Q[0] X[0] X[1]'
[0] X[0]' X[1] + Q[2] Q[1] Q[0] X[0] X[1]' + Q[2] Q[1] Q[0] X[0]
Z[2] = Q[2] Q[1]' Q[0]' X[0]' X[1]' + Q[2] Q[1]' Q[0]' X[0]' X[1
[1]' Q[0] X[0]' X[1]' + Q[2] Q[1]' Q[0] X[0]' X[1] + Q[2] Q[1]'
[1]' + Q[2] Q[1] Q[0]' X[0]' X[1] + Q[2] Q[1] Q[0]' X[0] X[1]' +
[0] X[0]' X[1] + Q[2] Q[1] Q[0] X[0] X[1]' + Q[2] Q[1] Q[0] X[0]
Z[1] = Q[2] Q[1]' Q[0] X[0]' X[1]' + Q[2] Q[1]' Q[0] X[0]' X[1]
[0]' X[0]' X[1]' + Q[2] Q[1] Q[0]' X[0]' X[1] + Q[2] Q[1] Q[0]'
[2] Q[1] Q[0] X[0]' X[1] + Q[2] Q[1] Q[0] X[0] X[1]' + Q[2] Q[1]
Z[0] = Q[2] Q[1] Q[0]' X[0]' X[1]' + Q[2] Q[1] Q[0]' X[0]' X[1]
[0] X[0]' X[1]' + Q[2] Q[1] Q[0] X[0]' X[1] + Q[2] Q[1] Q[0] X[0

Minimized:
D[2] = Q[2] Q[1]'  + Q[1]' Q[0]' X[0] ;
D[1] = Q[1]' Q[0]  + Q[2]' Q[1] Q[0]'  + Q[2]' Q[0]' X[0] X[1];
D[0] = Q[2]' Q[1] Q[0]'  + Q[2] Q[1]' Q[0]'  + Q[1]' Q[0]' X[1];
Z[5] = Q[1] Q[0] ;
Z[4] = Q[2]' Q[1]  + Q[1] Q[0] ;
Z[3] = Q[2]' Q[1]  + Q[2]' Q[0]  + Q[1] Q[0] ;
Z[2] = Q[2] ;
Z[1] = Q[2] Q[1]  + Q[2] Q[0] ;
Z[0] = Q[2] Q[1] ;
```

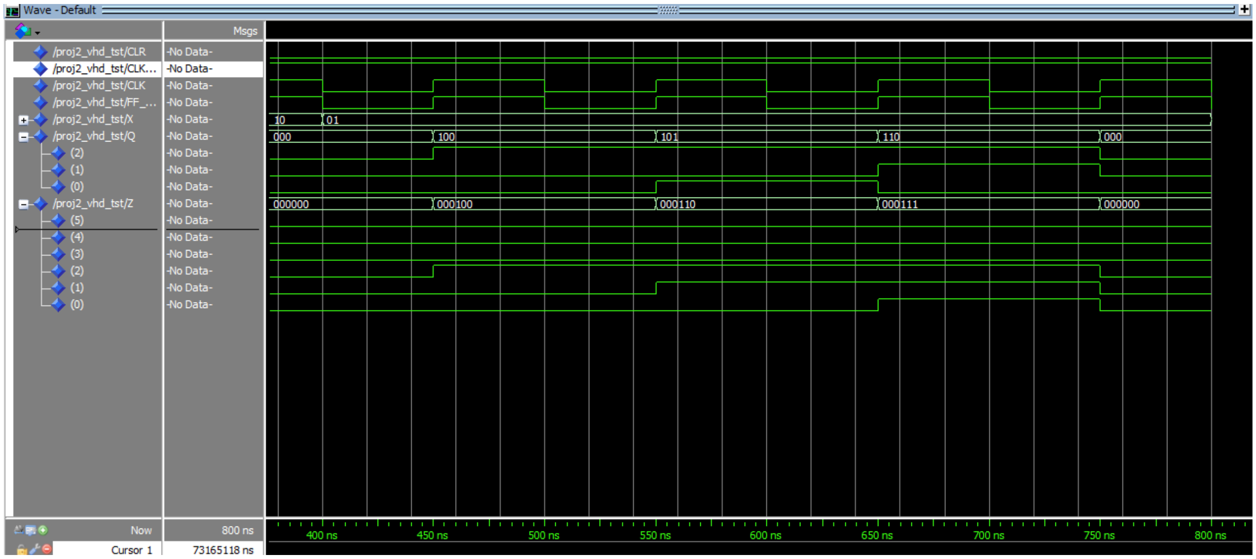


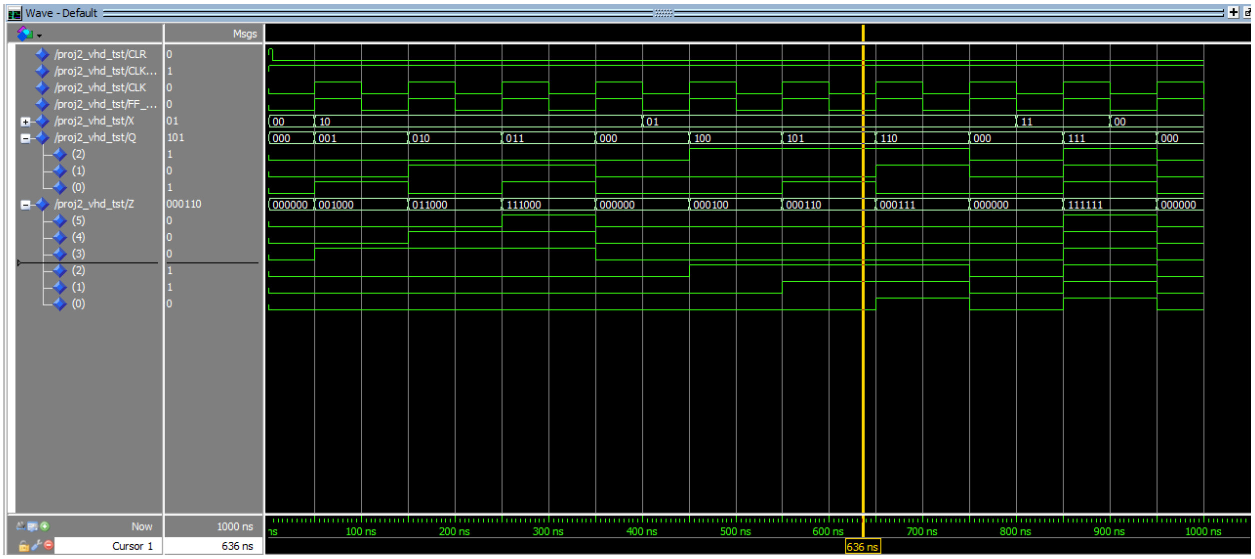
Parameter	Value	Type
N	24	Signed Integer

The schematic makes use of feedback, clock divider, 2x1 multiplex, and a D-flipflop. The Input variables include the CLK, CLR, CLK_SEL, and X[1..0]. The

output is the FF_CLK, Z[5..0], and Q[2..0]. The state table is developed in order of idle, left, and right, for implementation of the feedback input. At the bottom right of the Logic Friday table is the minimized equation for each specific light. The minimized equation is used to create the feedback input in Quartus Lite.

Results and Testing





The waveform shows which digits will “light” up to synchronous with each other. You can see during certain parts of the waveform when the light will indicate right, left, or idle.

Conclusion

In conclusion, we can use Quartus Lite to implement a synchronous state machine. We can develop a synchronous state machine to create a sequential taillight controller. We can test our schematic to see the results on a waveform. We can finally implement our code on an FPGA board.