EEL 3701 — Digital Logic & Computer Systems
Revision 1
Lab 5 Report: **Example State Machine: Traffic-Light Controller**

Dupuis, Connor
Class #: 12478
Kevin Lovell
Month Day, Year

PRE-LAB QUESTIONS OR EXERCISES

N/A

PROBLEMS ENCOUNTERED

N/A

REQUIREMENTS NOT MET

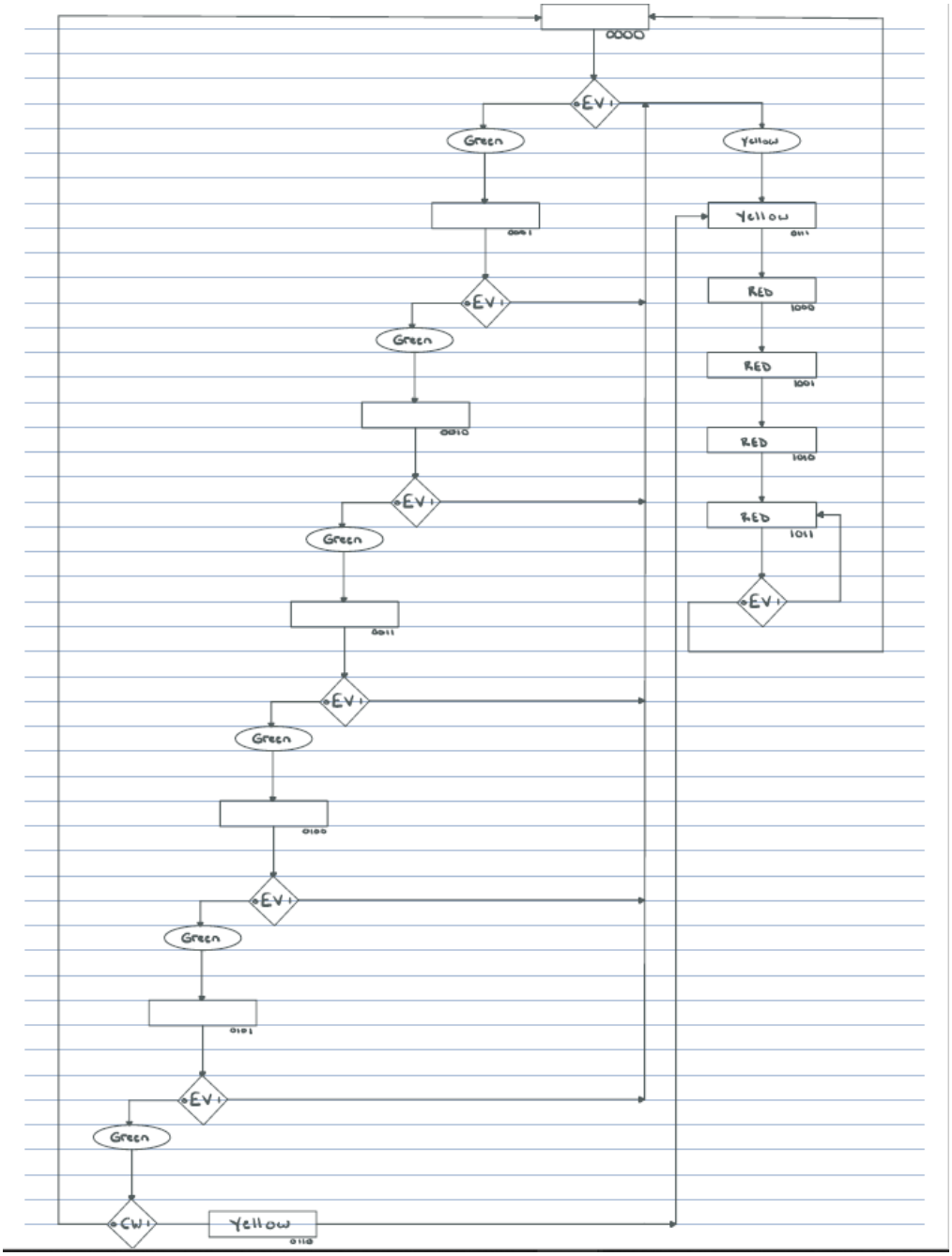
N/A

FUTURE WORK/APPLICATIONS

This lab had us create a traffic light controller, which can already be applied to future applications because it is something that is already used daily. This lab also introduced us to more in-depth state machines and how we can create diagrams to help aid us in the creation of these state machines. The use of VHDL code instead of creating large amounts of combinatorial logic is also something that may help us in the future.

PRE-LAB REQUIREMENTS (Design, Schematic, ASM Chart, VHDL, etc.)

ASM Chart:

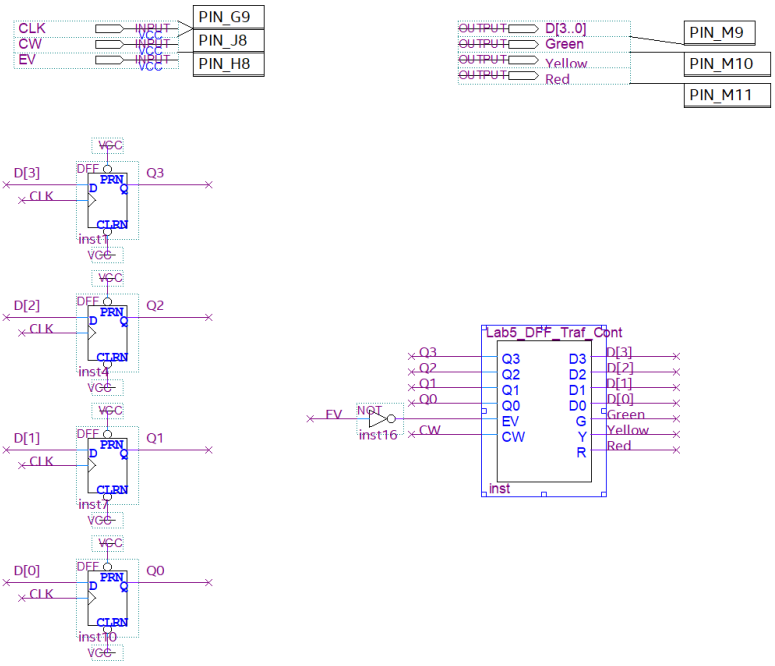


Next State Truth Table:

Q_3	Q_2	Q_1	Q_0	EV	CW	D_3 Q_3^+	D_2 Q_2^+	D_1 Q_1^+	D_0 Q_0^+	G	Y	R
0	0	0	0	0	X	0	0	0	1	1	0	0
0	0	0	0	1	X	0	1	1	1	0	1	0
0	0	0	1	0	X	0	0	1	0	1	0	0
0	0	0	1	1	X	0	1	1	1	0	1	0
0	0	1	0	0	X	0	0	1	1	1	0	0
0	0	1	0	1	X	0	1	1	1	0	1	0
0	0	1	1	0	X	0	1	0	0	1	0	0
0	0	1	1	1	X	0	1	1	1	0	1	0
0	1	0	0	0	X	0	1	0	1	1	0	0
0	1	0	0	1	X	0	1	1	1	0	1	0
0	1	0	1	0	0	0	0	0	0	1	0	0
0	1	0	1	0	1	0	1	1	0	1	0	0
0	1	0	1	1	X	0	1	1	1	0	1	0
0	1	1	0	X	X	0	1	1	1	0	1	0
0	1	1	1	X	X	1	0	0	0	0	1	0
1	0	0	0	X	X	1	0	0	1	0	0	1
1	0	0	1	X	X	1	0	1	0	0	0	1
1	0	1	0	X	X	1	0	1	1	0	0	1
1	0	1	1	0	X	0	0	0	0	0	0	1
1	0	1	1	1	X	1	0	1	1	0	0	1

Design/Simulation:

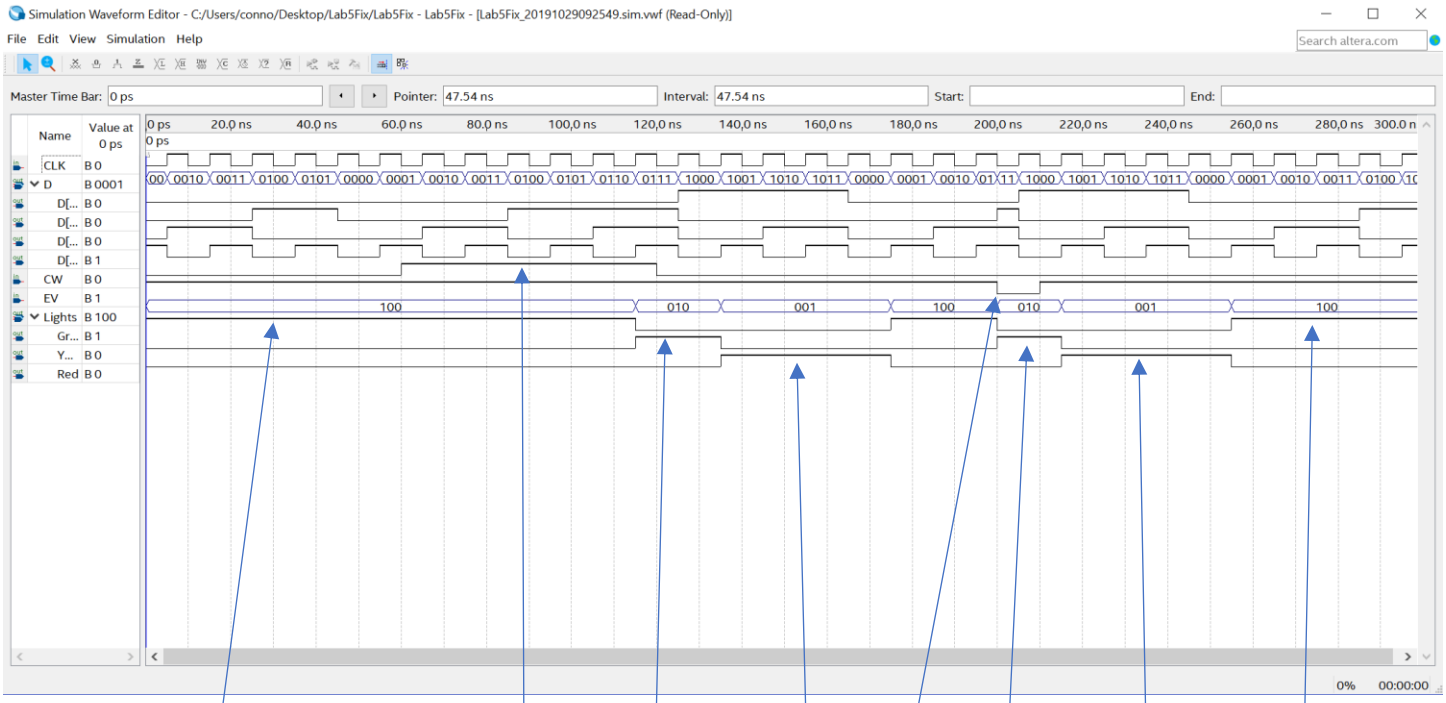
Lab 5 Part 1
Name: Connor Dupuis
Class: 12478
PI Name: Kevin Lovell
Description: Implementaion of VHDL code



EEL 3701 — Digital Logic & Computer Systems
Revision 1

Lab 5 Report: Example State Machine: Traffic-Light Controller

Dupuis, Connor
Class #: 12478
Kevin Lovell
Month Day, Year



The first section has no emergency vehicle and no car waiting so the light always remains green.

Car waiting turns true, so when the end of the current cycle occurs the light turns yellow for two clocks and red for four and finally goes back to green.

An emergency vehicle approaches so the light immediately turns yellow for at least one clock, but no more than two, then turns red for four clocks. Because there is no more emergency vehicle it goes back to green.

VHDL Code:

```
library ieee; use ieee.std_logic_1164.all;
entity Lab5_DFF_Traf_Cont is port (
        Q3, Q2, Q1, Q0: in std_logic;
        EV, CW: in std_logic;
        D3, D2, D1, D0: out std_logic;
        G, Y, R: out std_logic
);
end Lab5_DFF_Traf_Cont;
architecture logic OF Lab5_DFF_Traf_Cont IS
begin

--D3 = (/Q3*Q2*Q1*Q0)+(Q3*/Q2*/Q1*/Q0)+(Q3*/Q2*/Q1*Q0)+(Q3*/Q2*Q1*/Q0)+(Q3*/Q2*Q1*Q0*EV)

--D2 =
(/Q3*/Q2*/Q1*/Q0*EV)+(/Q3*/Q2*/Q1*Q0*EV)+(/Q3*/Q2*Q1*/Q0*EV)+(/Q3*/Q2*Q1*Q0*/EV)+(/Q3*/Q2*Q1*Q0*EV)+(/Q3*Q2*/Q1*/Q0*/
EV)+(/Q3*Q2*/Q1*/Q0*EV)+(/Q3*Q2*/Q1*Q0*/EV*CW)+(/Q3*Q2*/Q1*Q0*EV)+(/Q3*Q2*Q1*/Q0)

--D1 =
(Q3+Q2+Q1+Q0+EV)(Q3+Q2+/Q1+/Q0+EV)(Q3+/Q2+Q1+Q0+EV)(Q3+/Q2+Q1/Q0+EV+CW)(Q3+/Q2+/Q1+/Q0)(/Q3+Q2+Q1+Q0)(/Q3+Q2+/
Q1+/Q0+EV)

--D0 =
(Q3+Q2+Q1+/Q0+EV)(Q3+Q2+/Q1+/Q0+EV)(Q3+/Q2+Q1+/Q0+EV+CW)(Q3+/Q2+Q1+/Q0+EV+/CW)(Q3+/Q2+/Q1+/Q0)(/Q3+Q2+Q1+/Q0)(/
Q3+Q2+/Q1+/Q0+EV)

--G =
(/Q3*/Q2*/Q1*/Q0*/EV)+(/Q2*/Q2*/Q1*Q0*/EV)+(/Q3*/Q2*Q1*/Q0*/EV)+(/Q3*/Q2*Q1*Q0*/EV)+(/Q3*Q2*/Q1*/Q0*/EV)+(/Q3*Q2*/Q1*Q
0*/EV*/CW) (/Q3*Q2*/Q1*Q0*/EV*CW)+(Q3*/Q2*Q1*Q0*/EV)

--Y =
(/Q3*/Q2*/Q1*/Q0*EV)+(/Q2*/Q2*/Q1*Q0*EV)+(/Q3*/Q2*Q1*/Q0*EV)+(/Q3*/Q2*Q1*Q0*EV)+(/Q3*Q2*/Q1*/Q0*EV)+(/Q3*Q2*/Q1*Q0*E
V)+(/Q3*Q2*Q1*/Q0)(/Q3*Q2*Q1*Q0)

--R = (Q3*/Q2*/Q1*/Q0)+(Q3*/Q2*/Q1*Q0)+(Q3*/Q2*Q1*/Q0)+(Q3*/Q2*Q1*Q0*/EV)+(Q3*/Q2*Q1*Q0*EV)

D3<=(((NOT Q3)AND(Q2)AND(Q1)AND(Q0))
  OR((Q3)AND(NOT Q2)AND(NOT Q1)AND(NOT Q0))
  OR((Q3)AND(NOT Q2)AND(NOT Q1)AND(Q0))
  OR((Q3)AND(NOT Q2)AND(Q1)AND(NOT Q0))
  OR((Q3)AND(NOT Q2)AND(Q1)AND(Q0)AND(EV)));

D2<=(((NOT Q3)AND(NOT Q2)AND(NOT Q1)AND(NOT Q0)AND(EV))
  OR((NOT Q3)AND(NOT Q2)AND(NOT Q1)AND(Q0)AND(EV))
  OR((NOT Q3)AND(NOT Q2)AND(Q1)AND(NOT Q0)AND(EV))
  OR((NOT Q3)AND(NOT Q2)AND(Q1)AND(Q0)AND(NOT EV))
  OR((NOT Q3)AND(NOT Q2)AND(Q1)AND(Q0)AND(EV))
  OR((NOT Q3)AND(Q2)AND(NOT Q1)AND(NOT Q0)AND(NOT EV))
  OR((NOT Q3)AND(Q2)AND(NOT Q1)AND(NOT Q0)AND(EV))
  OR((NOT Q3)AND(Q2)AND(NOT Q1)AND(Q0)AND(NOT EV)AND(CW))
  OR((NOT Q3)AND(Q2)AND(NOT Q1)AND(Q0)AND(EV))
  OR((NOT Q3)AND(Q2)AND(Q1)AND(NOT Q0)));

D1<=(((Q3)OR(Q2)OR(Q1)OR(Q0)OR(EV))
  AND((Q3)OR(Q2)OR(NOT Q1)OR(NOT Q0)OR(EV))
  AND((Q3)OR(NOT Q2)OR(Q1)OR(Q0)OR(EV))
  AND((Q3)OR(NOT Q2)OR(Q1)OR(NOT Q0)OR(EV)OR(CW))
  AND((Q3)OR(NOT Q2)OR(NOT Q1)OR(NOT Q0))
  AND((NOT Q3)OR(Q2)OR(Q1)OR(Q0))
  AND((NOT Q3)OR(Q2)OR(NOT Q1)OR(NOT Q0)OR(EV)));

D0<=(((Q3)OR(Q2)OR(Q1)OR(NOT Q0)OR(EV))
  AND((Q3)OR(Q2)OR(NOT Q1)OR(NOT Q0)OR(EV))
  AND((Q3)OR(NOT Q2)OR(Q1)OR(NOT Q0)OR(EV)OR(CW))
```

```
  AND((Q3)OR(NOT Q2)OR(Q1)OR(NOT Q0)OR(EV)OR(NOT CW))
  AND((Q3)OR(NOT Q2)OR(NOT Q1)OR(NOT Q0))
  AND((NOT Q3)OR(Q2)OR(Q1)OR(NOT Q0))
  AND((NOT Q3)OR(Q2)OR(NOT Q1)OR(NOT Q0)OR(EV)));

G<=(((NOT Q3)AND(NOT Q2)AND(NOT Q1)AND(NOT Q0)AND(NOT EV))
  OR((NOT Q3)AND(NOT Q2)AND(NOT Q1)AND(Q0)AND(NOT EV))
  OR((NOT Q3)AND(NOT Q2)AND(Q1)AND(NOT Q0)AND(NOT EV))
  OR((NOT Q3)AND(NOT Q2)AND(Q1)AND(Q0)AND(NOT EV))
  OR((NOT Q3)AND(Q2)AND(NOT Q1)AND(NOT Q0)AND(NOT EV))
  OR((NOT Q3)AND(Q2)AND(NOT Q1)AND(Q0)AND(NOT EV)AND(NOT CW))
  OR((NOT Q3)AND(Q2)AND(NOT Q1)AND(Q0)AND(NOT EV)AND(CW)));

Y<=(((NOT Q3)AND(NOT Q2)AND(NOT Q1)AND(NOT Q0)AND(EV))
  OR((NOT Q3)AND(NOT Q2)AND(NOT Q1)AND(Q0)AND(EV))
  OR((NOT Q3)AND(NOT Q2)AND(Q1)AND(NOT Q0)AND(EV))
  OR((NOT Q3)AND(NOT Q2)AND(Q1)AND(Q0)AND(EV))
  OR((NOT Q3)AND(Q2)AND(NOT Q1)AND(NOT Q0)AND(EV))
  OR((NOT Q3)AND(Q2)AND(NOT Q1)AND(Q0)AND(EV))
  OR((NOT Q3)AND(Q2)AND(Q1)AND(NOT Q0))
  OR((NOT Q3)AND(Q2)AND(Q1)AND(Q0)));

R<=(((Q3)AND(NOT Q2)AND(NOT Q1)AND(NOT Q0))
  OR((Q3)AND(NOT Q2)AND(NOT Q1)AND(Q0))
  OR((Q3)AND(NOT Q2)AND(Q1)AND(NOT Q0))
  OR((Q3)AND(NOT Q2)AND(Q1)AND(Q0)AND(NOT EV))
  OR((Q3)AND(NOT Q2)AND(Q1)AND(Q0)AND(EV)));

end logic;
```