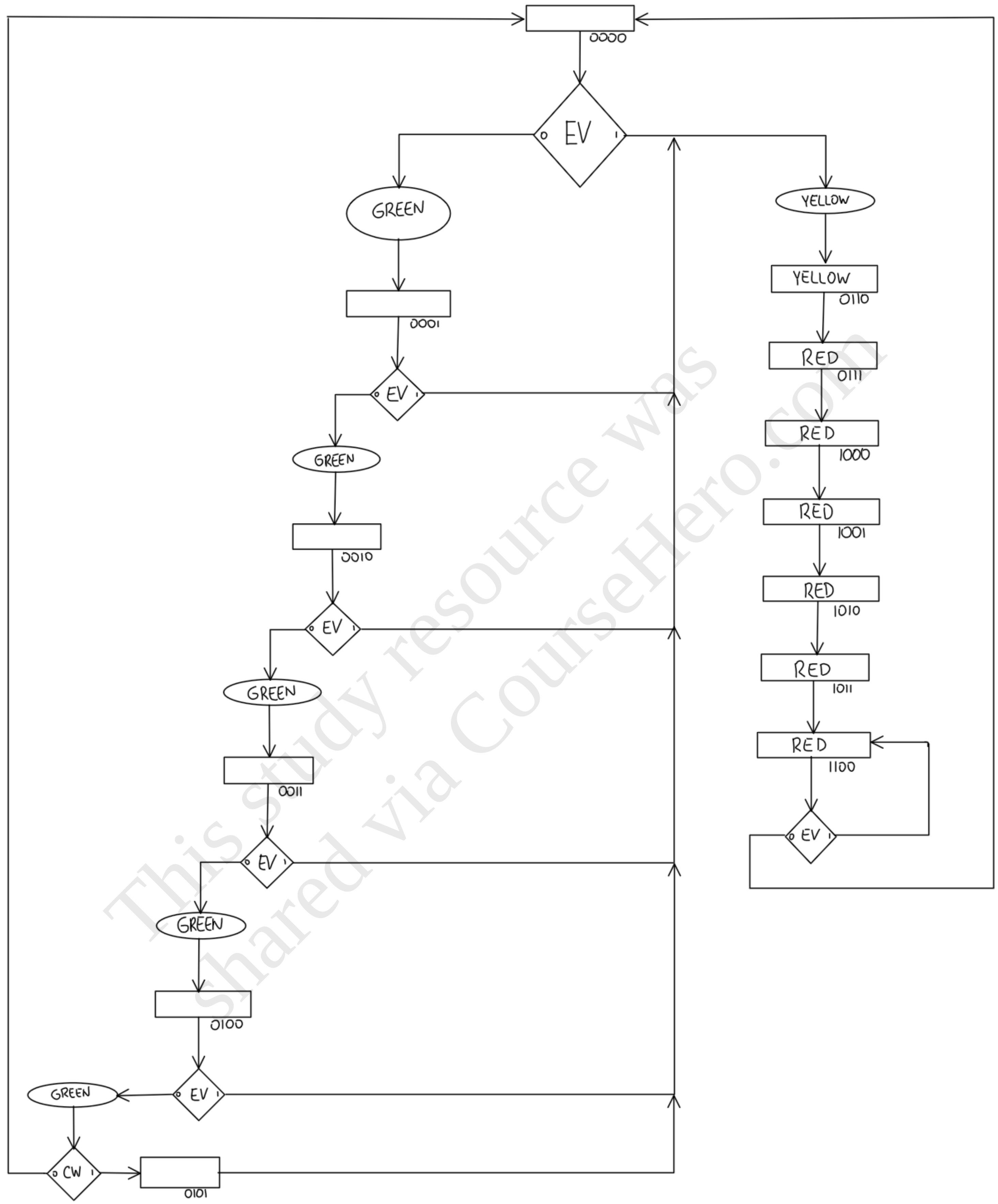
Lab 5: Traffic Light Controller

Name: Juan Valbuena

Lab Section: 1491

ASM DESIGN


0000
0
EV
1
GREEN
0001
0
EV
1
GREEN
0010
0
EV
1
GREEN
0011
0
EV
1
GREEN
0100
0
EV
1
GREEN
0
CW
1
0101
YELLOW
YELLOW
0110
RED
0111
RED
1000
RED
1001
RED
1010
RED
1011
RED
1100
0
EV
1

Part 1: Lab5_DFF_Traf_Cont

Next State Truth Table

						D_3	D_2	D_1	D_0			
Q_3	Q_2	Q_1	Q_0	EV	CW	Q_3^+	Q_2^+	Q_1^+	Q_0^+	GREEN	YELLOW	RED
0	0	0	0	0	X	0	0	0	1	1	0	0
0	0	0	0	1	X	0	1	1	0	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	0	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	0	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	0	0	1	0
0	1	0	0	0	0	0	0	0	0	1	0	0
0	1	0	0	0	1	0	1	0	1	1	0	0
0	1	0	0	1	X	0	1	1	0	0	1	0
0	1	0	1	X	X	0	1	1	0	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	0	1
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	X	X	1	1	0	0	0	0	1
1	1	0	0	0	X	0	0	0	0	0	0	1
1	1	0	0	1	X	1	1	0	0	0	0	1

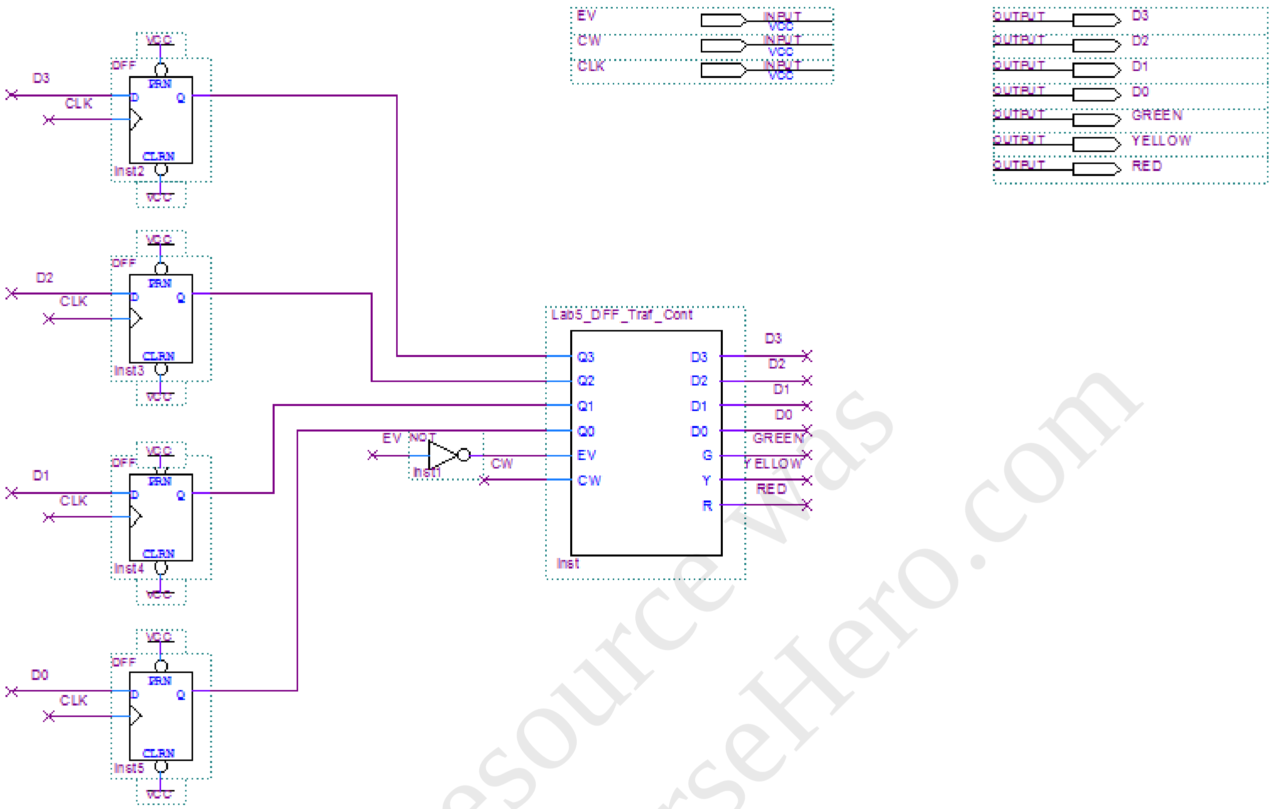
VHDL CODE

```
library ieee; use ieee.std_logic_1164.all;
entity Lab5_DFF_Traf_Cont is port (
                                        Q3, Q2, Q1, Q0: in bit;
                                        EV, CW: in bit;
                                        D3, D2, D1, D0: out bit;
                                        G, Y, R: out bit
);
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) + (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)
                                        OR (Q3 AND Q2 AND (NOT Q1) AND (NOT Q0) AND EV);

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

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

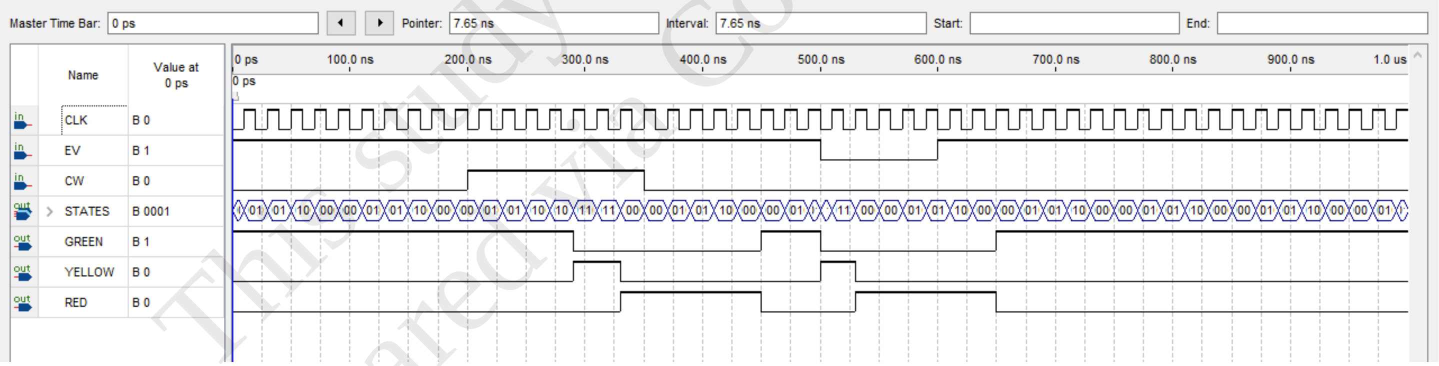
```
-- D1 = (Q3+Q2+Q1+Q0+EV) * (Q3+Q2+/Q1+/Q0+EV) * (Q3+/Q2+Q1+Q0+EV+CW) * (Q3+/Q2+Q1+Q0+EV+/CW) * (Q3+/Q2+/Q1+/Q0) * (/Q3+Q2+Q1+Q0)
--              * (/Q3+Q2+/Q1+/Q0) * (/Q3+/Q2+Q1+Q0+EV) * (/Q3+/Q2+Q1+Q0+/EV)

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

-- D0 = (/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)

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

-- G = (/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*CW)

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

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

                    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)
                              OR ((NOT Q3) AND Q2 AND Q1 AND (NOT Q0));

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

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

end logic;
```

Schematic

Name: Juan Valbuena
Lab 5 Part 1
Section #: 1491
TA Name: Veronica Frie
Des cription: the s chematic for the traffic controller us ing only D-FF



Simulation



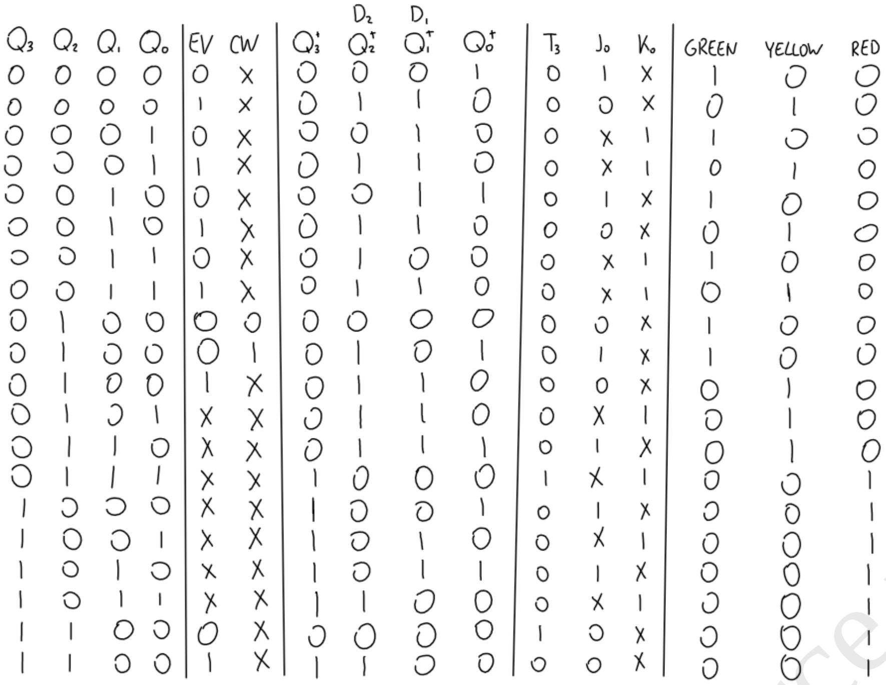
Master Time Bar: 0 ps
Pointer: 7.65 ns
Interval: 7.65 ns
Start:
End:

Name	Value at 0 ps
CLK	B 0
EV	B 1
CW	B 0
STATES	B 0001
GREEN	B 1
YELLOW	B 0
RED	B 0

The simulation is working perfectly. I adjusted the clock cycles to occur every 10 ns to have various cycles occur and see all the possible combinations for the controller in one simulation. The first section has both EV and CW set to 0 and the light stays green at all times. Afterwards, I set CW to 1 and after 5 cycles of green it goes to yellow for two cycles and then three for 6 cycles as instructed. Finally, I set EV to 1 a little after the beginning of the new cycle. It goes green for two cycles and recognizes EV is 1 and then changes to yellow for one cycle and then red for 6. The rest of the simulation stays green since both EV and CW are set to 0.

Part 2: Lab5_not_DFF_Traf_Cont

Next State Truth Table



							D2	D1							
Q_3	Q_2	Q_1	Q_0	EV	CW	Q_3^+	Q_2^+	Q_1^+	Q_0^+	T_3	J_0	K_0	GREEN	YELLOW	RED
0	0	0	0	0	X	0	0	0	1	0	1	X	1	0	0
0	0	0	0	1	X	0	1	1	0	0	0	X	0	1	0
0	0	0	1	0	X	0	0	1	0	0	X	1	1	0	0
0	0	0	1	1	X	0	1	1	0	0	X	1	0	1	0
0	0	1	0	0	X	0	0	1	1	0	1	X	1	0	0
0	0	1	0	1	X	0	1	1	0	0	0	X	0	1	0
0	0	1	1	0	X	0	1	0	0	0	X	1	1	0	0
0	0	1	1	1	X	0	1	1	0	0	X	1	0	1	0
0	1	0	0	0	0	0	0	0	0	0	0	X	1	0	0
0	1	0	0	0	1	0	1	0	1	0	1	X	1	0	0
0	1	0	0	1	X	0	1	1	0	0	0	X	0	1	0
0	1	0	1	X	X	0	1	1	0	0	X	1	0	1	0
0	1	1	0	X	X	0	1	1	1	0	1	X	0	1	0
0	1	1	1	X	X	1	0	0	0	1	X	1	0	0	1
1	0	0	0	X	X	1	0	0	1	0	1	X	0	0	1
1	0	0	1	X	X	1	0	1	0	0	X	1	0	0	1
1	0	1	0	X	X	1	0	1	1	0	1	X	0	0	1
1	0	1	1	X	X	1	1	0	0	0	X	1	0	0	1
1	1	0	0	0	X	0	0	0	0	1	0	X	0	0	1
1	1	0	0	1	X	1	1	0	0	0	0	X	0	0	1

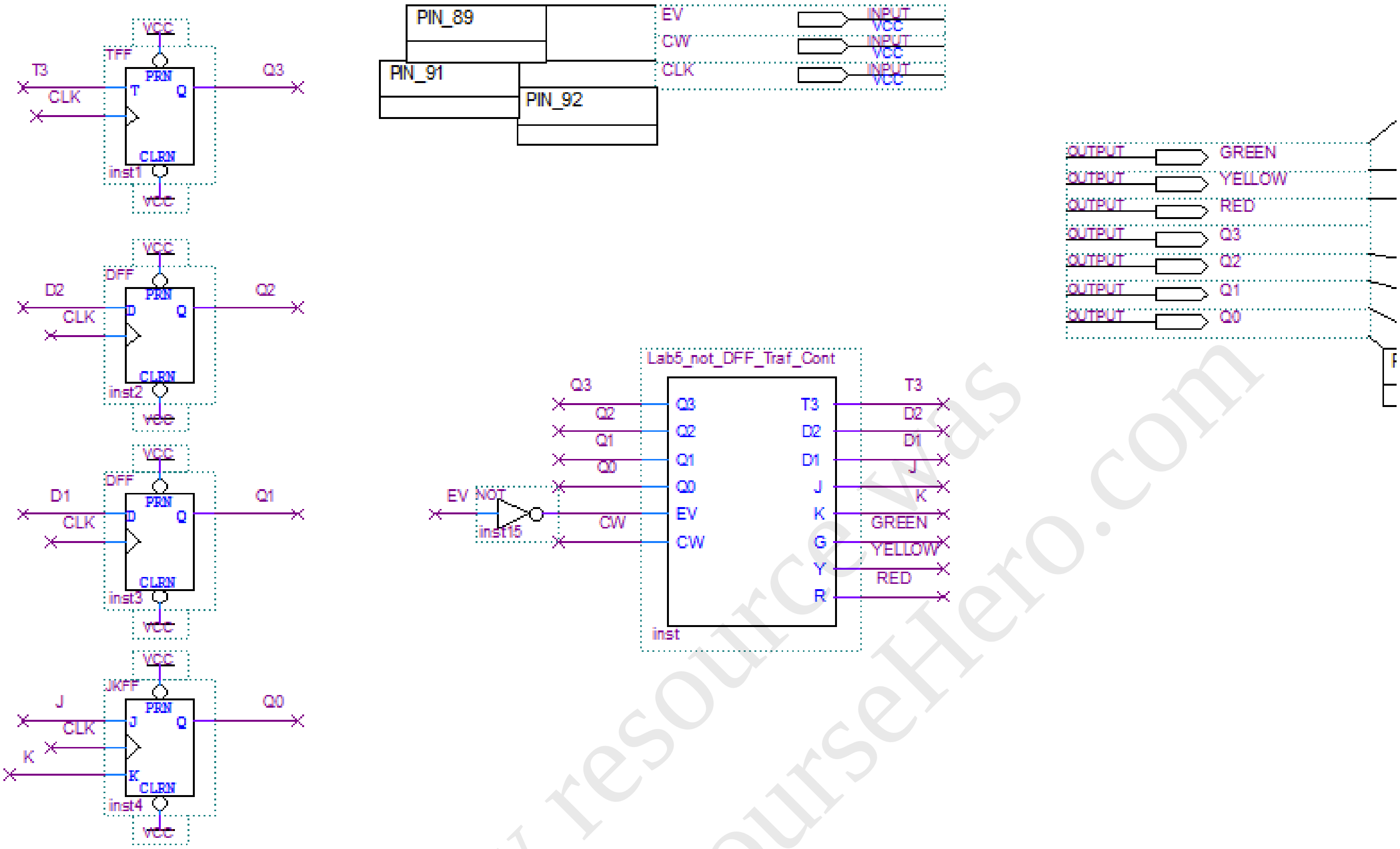
VHDL Code

```
library ieee; use ieee.std_logic_1164.all;
entity Lab5_not_DFF_Traf_Cont is port (
                                        Q3, Q2, Q1, Q0: in bit;
                                        EV, CW: in bit;
                                        T3, D2, D1, J, K: out bit;
                                        G, Y, R: out bit
);
end Lab5_not_DFF_Traf_Cont;
architecture logic OF Lab5_not_DFF_Traf_Cont IS
begin

-- T3 = (/Q3*Q2*Q1*Q0) + (Q3*Q2*/Q1*/Q0*/EV)

                    T3 <= ((NOT Q3) AND Q2 AND Q1 AND Q0)
                              OR (Q3 AND Q2 AND (NOT Q1) AND (NOT Q0) AND (NOT EV));

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

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

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

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

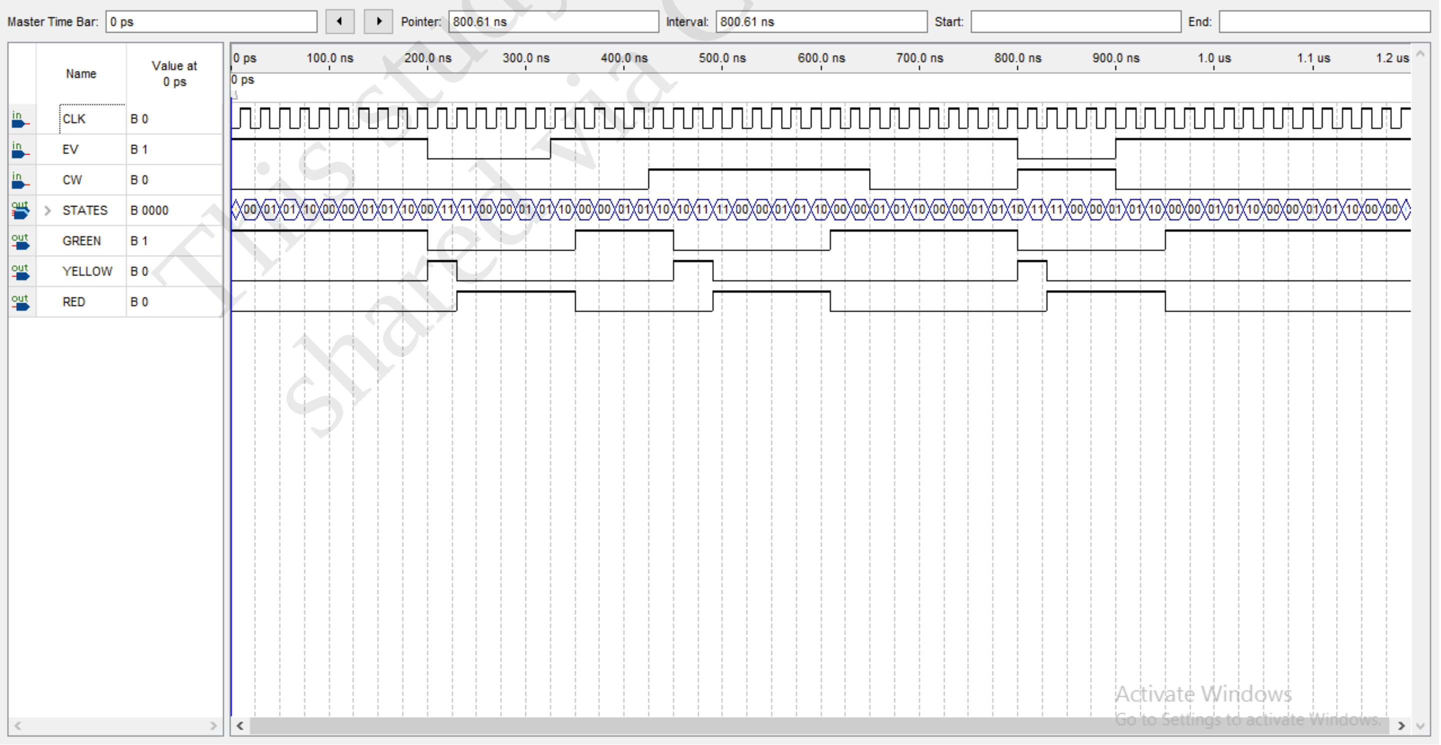
```
            AND (Q3 OR (NOT Q2) OR Q1 OR Q0 OR EV OR CW)
            AND (Q3 OR (NOT Q2) OR Q1 OR Q0 OR EV OR (NOT CW))
            AND (Q3 OR (NOT Q2) OR (NOT Q1) OR (NOT Q0))
            AND ((NOT Q3) OR Q2 OR Q1 OR Q0)
            AND ((NOT Q3) OR Q2 OR (NOT Q1) OR (NOT Q0))
            AND ((NOT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

-- J = (/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)

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

-- K = 1

        K <= '1';

--   G   =   (/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*CW)

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

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

        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)
            OR ((NOT Q3) AND Q2 AND Q1 AND (NOT Q0));

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

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

end logic;
```

Schematic

Name: Juan Valbuena
Lab 5 Part 2
Section #: 1491
TA Name: Veronica Pirie
Description: Schematic of the traffic controller using T-FF for most sig bit, JK-FF for the least sig big, and D-FF for everthing else.



Simulation



Master Time Bar: 0 ps
Pointer: 800.61 ns
Interval: 800.61 ns
Start:
End:

Name	Value at 0 ps
CLK	B 0
EV	B 1
CW	B 0
STATES	B 0000
GREEN	B 1
YELLOW	B 0
RED	B 0

The simulation is working perfectly with the customization of the T-FF, D-FF, and JK-FF. I adjusted the clock cycles to occur every 10 ns to have various cycles occur and see all the possible combinations for the controller in one simulation. The first section has both EV and CW set to 0 and the light stays green always. Afterwards, I set CW to 1 and after 5 cycles of green it goes to yellow for two cycles and then three for 6 cycles as instructed. Finally, I set EV to 1 a little after the beginning of the new cycle. It goes green for two cycles and recognizes EV is 1 and then changes to yellow for one cycle and then red for 6. The rest of the simulation stays green since both EV and CW are set to 0.

Problems Encountered

The hardware was giving me problems and even thought the simulation was working the LED's weren't lighting up how they should. I also used the static I/O and it was giving me similar problems.

Future Work/Application

Gave me better insight on ASM designs and get a better grasp on VHDL which will help with future labs and projects.