TRAFFIC LIGHT CONTROLLER

By: James Todd, Thierno Barry, Andrew Tamer, Gurashish Grewal

Date: 12/07/2017

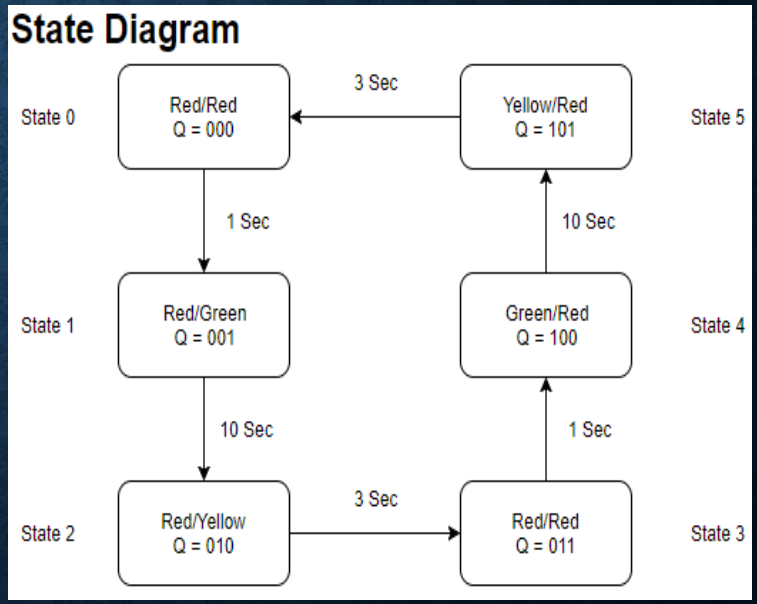
FALL-2017

INTRODUCTION

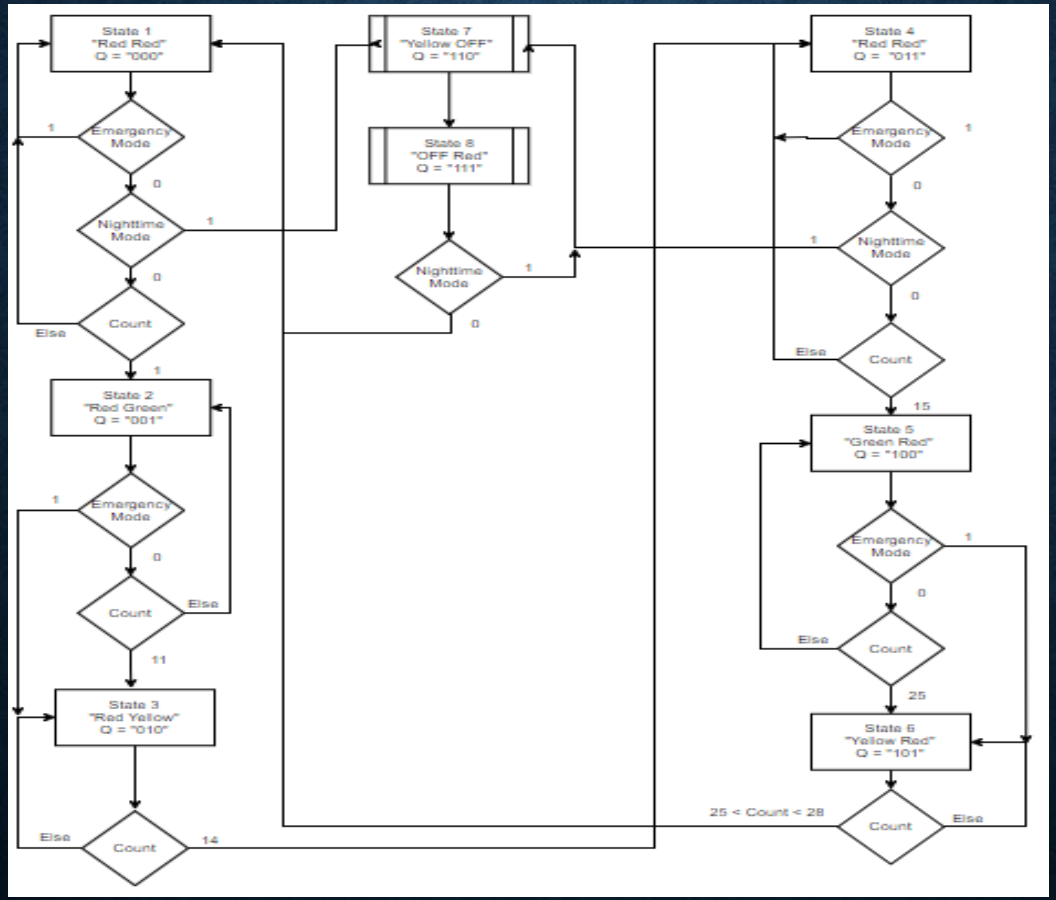
- 4 way intersection light system.
- Night time system- Lights start blinking Red after defined amount of time.
- Emergency Light System- If an emergency switch is on, all lights go to red.
- Counter, Clock divider, LEDs, FSM, Seven Segment Display files were used.

METHODOLOGY

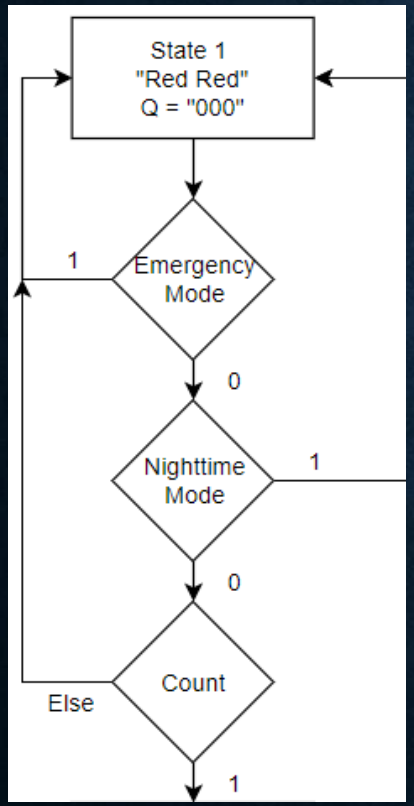
- The states would move on as per inputs, with delays accounted at each state change in the diagram.
- As the states change, the LED displays change simultaneously as per the states showing both NS and EW signals.
- Seven Segment display constantly displays in letters the states of the signals. For ex- Red/Red or Red/Green and so on.

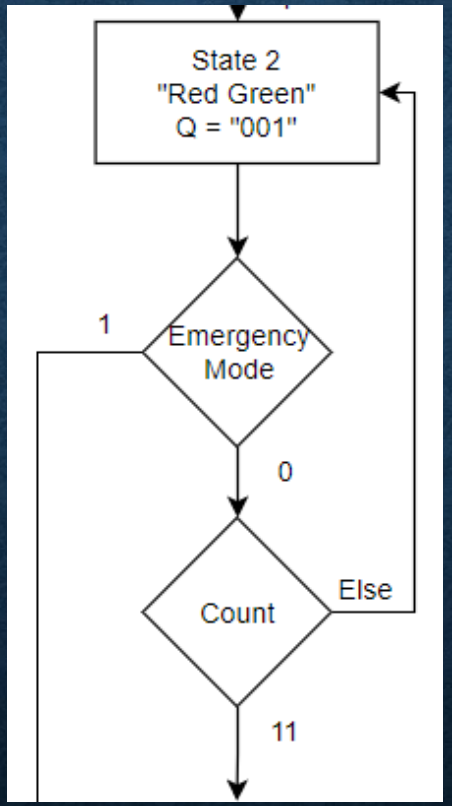


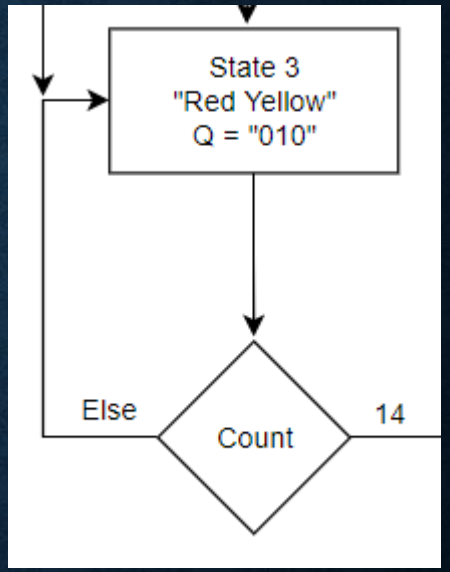
FSM


State 1
"Red Red"
Q = "000"
Emergency Mode
1
0
Nighttime Mode
1
0
Count
Else
1
State 2
"Red Green"
Q = "001"
Emergency Mode
1
0
Count
Else
11
State 3
"Red Yellow"
Q = "010"
Else
Count
14
State 7
"Yellow OFF"
Q = "110"
State 8
"OFF Red"
Q = "111"
Nighttime Mode
1
0
State 4
"Red Red"
Q = "011"
Emergency Mode
1
0
Nighttime Mode
1
0
Else
Count
15
State 5
"Green Red"
Q = "100"
Emergency Mode
1
0
Else
Count
25
State 6
"Yellow Red"
Q = "101"
25 < Count < 28
Count
Else

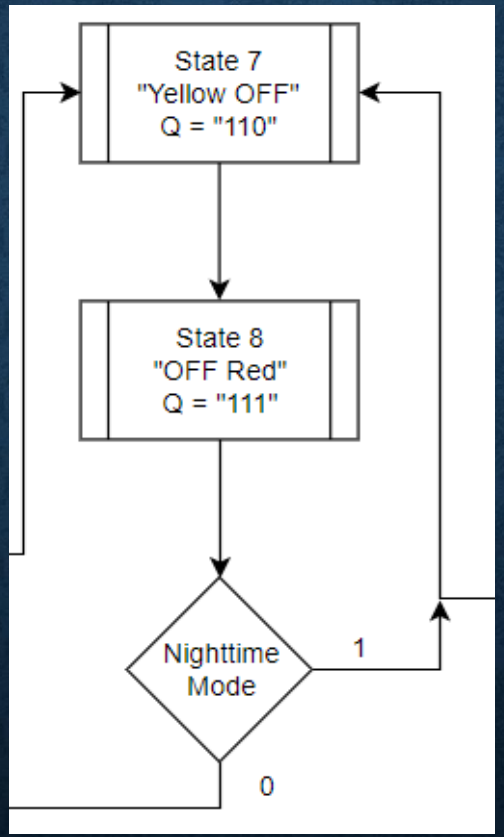
FSM


State 1
"Red Red"
Q = "000"
Emergency Mode
1
0
Nighttime Mode
1
0
Count
Else
1


State 2
"Red Green"
Q = "001"
Emergency Mode
1
0
Count
Else
11


State 3
"Red Yellow"
Q = "010"
Count
Else
14

FSM


State 7
"Yellow OFF"
Q = "110"
State 8
"OFF Red"
Q = "111"
Nighttime
Mode
1
0

NEXYS DISPLAY

- Two modes (Day time mode, and the night mode)
- Used four colors for the LEDs (RED, YELLOW, GREEN, and BLUE for demonstration purposes)
- Clock and States run simultaneously in Seven Segment file in order to be able to make all the letters visible at the same time.

The LED and Seven Segment Display files receive a three bit input unique to the present state controlled by the top file. Based upon the present state, the LEDs will display the appropriate color and the Seven Segment will display the state the signal is in. Both, the LEDs and Seven Segment will represent the north/south signal, and the east west signal.

LED file

```
begin
    if ColorChange = '1' then
        case sel is
            when "000" => NorthSouthLED <= "100"; EastWestLED <= "100"; -- red red
            when "001" => NorthSouthLED <= "010"; EastWestLED <= "100"; -- green red
            when "010" => NorthSouthLED <= "110"; EastWestLED <= "100"; -- yellow red
            when "011" => NorthSouthLED <= "100"; EastWestLED <= "100"; -- red red
            when "100" => NorthSouthLED <= "100"; EastWestLED <= "010"; -- red green
            when "101" => NorthSouthLED <= "100"; EastWestLED <= "110"; -- red yellow
            when "110" => NorthSouthLED <= "100"; EastWestLED <= "000"; -- red OFF
            when "111" => NorthSouthLED <= "000"; EastWestLED <= "110"; -- OFF Yellow
            when others => NorthSouthLED <= "100"; EastWestLED <= "100"; -- red red
        end case;

    elsif ColorChange = '0' then
        case sel is
            when "000" => NorthSouthLED <= "100"; EastWestLED <= "100"; -- red red
            when "001" => NorthSouthLED <= "010"; EastWestLED <= "100"; -- green red
            when "010" => NorthSouthLED <= "001"; EastWestLED <= "100"; -- yellow red
            when "011" => NorthSouthLED <= "100"; EastWestLED <= "100"; -- red red
            when "100" => NorthSouthLED <= "100"; EastWestLED <= "010"; -- red green
            when "101" => NorthSouthLED <= "100"; EastWestLED <= "001"; -- red yellow
            when "110" => NorthSouthLED <= "100"; EastWestLED <= "000"; -- red OFF
            when "111" => NorthSouthLED <= "000"; EastWestLED <= "001"; -- OFF Yellow
            when others => NorthSouthLED <= "100"; EastWestLED <= "100"; -- red red
        end case;
    end if;
```

yellow

yellow

yellow

blue

blue

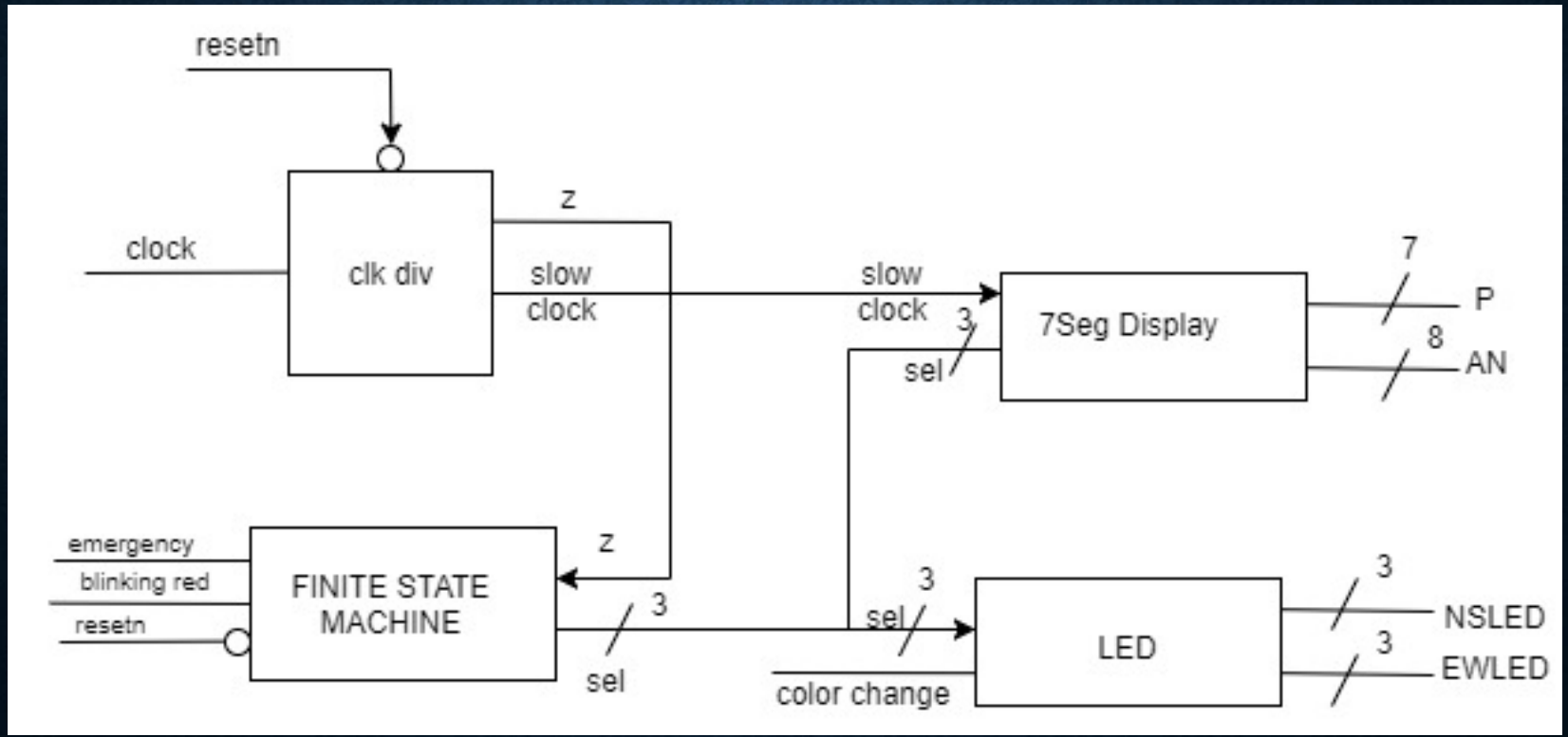
blue

Seven Segment Display

```vhdl
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity SevSeg is
    Port ( SlowerClock : in STD_LOGIC;
           P : out STD_LOGIC_VECTOR (6 downto 0);
           AN : out STD_LOGIC_VECTOR (7 downto 0);
           sel : in STD_LOGIC_VECTOR (2 downto 0));
end SevSeg;

architecture Behavioral of SevSeg is

type state is (D0, D1, D2, D3, D4, D5, D6, D7);
signal y: state;
signal Letter: STD_LOGIC_VECTOR (6 downto 0);
signal NotAN: std_logic_vector (7 downto 0);

begin

P <= not Letter;
AN <= not NotAN;

    CycleThroughDisplays: process ( SlowerClock )
    begin
        if (SlowerClock'event and SlowerClock = '1') then
            case y is
                when D0 => y <= D1;
                when D1 => y <= D2;
                when D2 => y <= D3;
                when D3 => y <= D4;
                when D4 => y <= D5;
                when D5 => y <= D6;
                when D6 => y <= D7;
                when D7 => y <= D0;
            end case;
        end if;
    end process;
```

```vhdl
    DisplayOutput: process (sel, y)
    begin
        case sel is
            when "000" =>
                case y is
                        when D0 => NotAN <= "10000000"; Letter<= "1010000"; -- r
                        when D1 => NotAN <= "01000000"; Letter<= "1111001"; -- E
                        when D2 => NotAN <= "00100000"; Letter<= "1011110"; -- d
                        when D3 => NotAN <= "00010000"; Letter<= "0000000";
                        when D4 => NotAN <= "00001000"; Letter<= "1010000"; -- r
                        when D5 => NotAN <= "00000100"; Letter<= "1111001"; -- E
                        when D6 => NotAN <= "00000010"; Letter<= "1011110"; -- d
                        when D7 => NotAN <= "00000001"; Letter<= "0000000";
                    end case;
            when "001" =>
                case y is
                        when D0 => NotAN <= "10000000"; Letter<= "1010000"; -- r
                        when D1 => NotAN <= "01000000"; Letter<= "1111001"; -- E
                        when D2 => NotAN <= "00100000"; Letter<= "1011110"; -- d
                        when D3 => NotAN <= "00010000"; Letter<= "0000000";
                        when D4 => NotAN <= "00001000"; Letter<= "1101111"; -- G
                        when D5 => NotAN <= "00000100"; Letter<= "1010000"; -- r
                        when D6 => NotAN <= "00000010"; Letter<= "1111001"; -- E
                        when D7 => NotAN <= "00000001"; Letter<= "1111001"; -- E
                    end case;
            when "010" =>
                case y is
                        when D0 => NotAN <= "10000000"; Letter<= "1010000"; -- r
                        when D1 => NotAN <= "01000000"; Letter<= "1111001"; -- E
                        when D2 => NotAN <= "00100000"; Letter<= "1011110"; -- d
                        when D3 => NotAN <= "00010000"; Letter<= "0000000";
                        when D4 => NotAN <= "00001000"; Letter<= "1101110"; -- Y
                        when D5 => NotAN <= "00000100"; Letter<= "1111001"; -- E
                        when D6 => NotAN <= "00000010"; Letter<= "0111000"; -- L
                        when D7 => NotAN <= "00000001"; Letter<= "0111000"; -- L
                    end case;
```

TRAFFIC LIGHT SYSTEM - DATAPATH



DIFFICULTIES/ ANY IMPROVEMENTS

R1 = A + BCD + B'C'D'E'

	A	B	C	D	E	North/South			East/West		
time (s)	q4	q3	q2	q1	q0	R	Y	G	R	Y	G
0	0	0	0	0	0	1	0	0	1	0	0
1	0	0	0	0	1	0	0	1	1	0	0
2	0	0	0	1	0	0	0	1	1	0	0
3	0	0	0	1	1	0	0	1	1	0	0
4	0	0	1	0	0	0	0	1	1	0	0
5	0	0	1	0	1	0	0	1	1	0	0
6	0	0	1	1	0	0	0	1	1	0	0
7	0	0	1	1	1	0	0	1	1	0	0
8	0	1	0	0	0	0	0	1	1	0	0
9	0	1	0	0	1	0	0	1	1	0	0
10	0	1	0	1	0	0	0	1	1	0	0
11	0	1	0	1	1	0	1	0	1	0	0
12	0	1	1	0	0	0	1	0	1	0	0
13	0	1	1	0	1	0	1	0	1	0	0
14	0	1	1	1	0	1	0	0	1	0	0
15	0	1	1	1	1	1	0	0	0	0	1
16	1	0	0	0	0	1	0	0	0	0	1
17	1	0	0	0	1	1	0	0	0	0	1
18	1	0	0	1	0	1	0	0	0	0	1
19	1	0	0	1	1	1	0	0	0	0	1
20	1	0	1	0	0	1	0	0	0	0	1
21	1	0	1	0	1	1	0	0	0	0	1
22	1	0	1	1	0	1	0	0	0	0	1
23	1	0	1	1	1	1	0	0	0	0	1
24	1	1	0	0	0	1	0	0	0	0	1
25	1	1	0	0	1	1	0	0	0	1	0
26	1	1	0	1	0	1	0	0	0	1	0
27	1	1	0	1	1	1	0	0	0	1	0

DE / ABC	00	01	11	10
000	1	0	0	0
001	0	0	0	0
011	0	0	1	1
010	0	0	0	0
100	1	1	1	1
101	1	1	1	1
111	x	x	x	x
110	1	1	1	1

DIFFICULTIES/ ANY IMPROVEMENTS

```
--clkDiv: process (Divider)
--begin
--    case Divider is
--        when 0 => Y <= "000";
--        when 1 => Y <= "001";
--        when 2 => Y <= "010";
--        when 3 => Y <= "011";
--        when 4 => Y <= "100";
--        when 5 => Y <= "101";
--        when 6 => Y <= "110";
--        when 7 => Y <= "111";

--    end case;
--end process;
```

```
if qt = 1000 then --101_1111_0101_1110_0001_0000_0000 1 sec 10111110101111000010000000
```

```
--        Q: out std_logic_vector (26 downto 0);
```

```
01DCD6500
```

```
signal count: integer := 0;
signal DividerCount: integer := 0;
```

Demonstration

Left LED

- East/West Signal

Right LED

- North/South Signal

QUESTIONS?