Eray Gündoğdu 25.12.2022

EEE102 Term Project:

Traffic Intersection

Brief Explanation of the Project:

In this Project an traffic intersection is designed.Intersection consists of 5 lights.4 of the lights are time based.A left turn light is operates with a sensor.In order to obtain this design a 8 state FSM is used .FSM goes to next state ın 20 seconds.This was done by clock division(100Mhz/2^n =1/20,n determines the vector dimension for the clock division).In this design ıf a movement is detected at light5,left turn light,light3 and light5 turn into green and other lights turn into red.In additoion there is a reset button assigned to R2 switch on BASYS3 which assignes current state to s0.

In s0: light1 is green, light2 is red, light3 is red, light4 is red,light5 is red.

In s1: light1 is green, light2 is yellow, light3 is red, light4 is red,light5 is red.

In s2: light1 is red, light2 is green, light3 is red, light4 is red,light5 is red.

In s3: light1 is red, light2 is green, light3 is yellow, light4 is red,light5 is red.

In s4: light1 is red, light2 is red, light3 is green, light4 is red,light5 is red.

In s5: light1 is red, light2 is red, light3 is green, light4 is yellow,light5 is red.

In s6: light1 is red, light2 is red, light3 is red, light4 is green,light5 is red.

In s7: light1 is yellow, light2 is red, light3 is red, light4 is green,light5 is red.

And if the output of the infrared sensor is 1 then,

light1 is red, light2 is red, light3 is green, light4 is green,light5 is green.

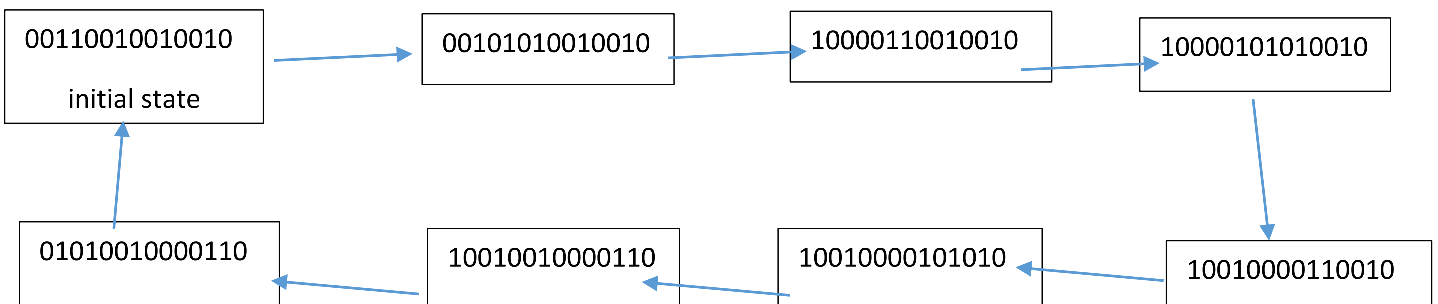
The states and sensor's way of working is demonstrated in the youtube video clearly.

The youtube video is sped up to save time,the state transition time might be decieving it is originally set to be 20 seconds.

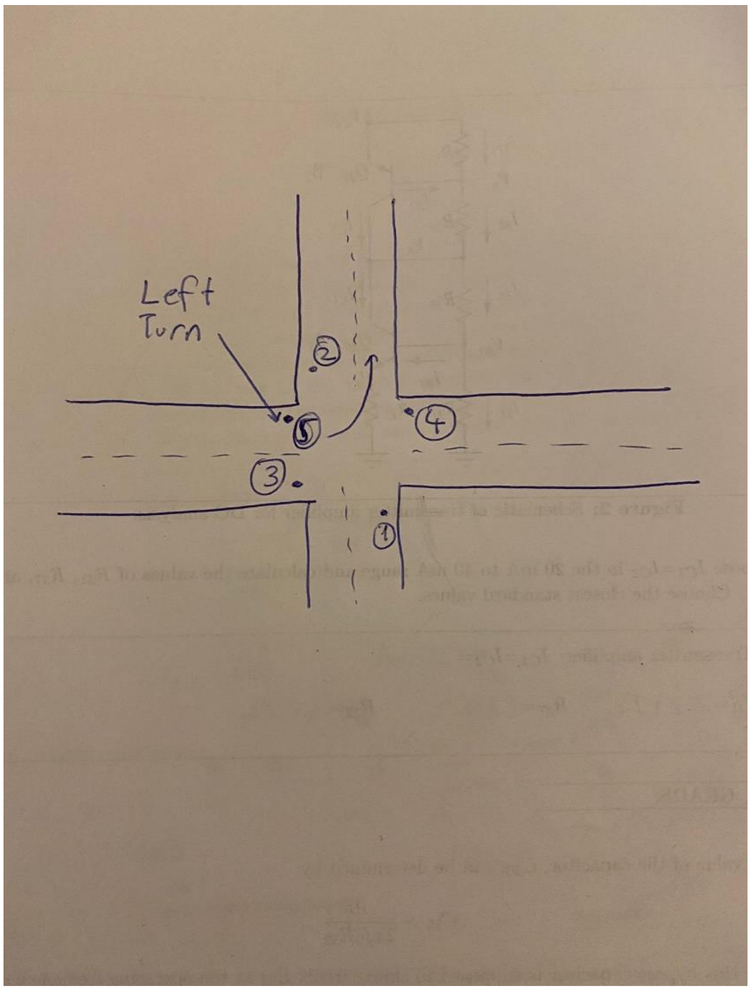
https://youtu.be/tHNObx1FZIc

State Diagram(With sensor output 0):

The lights can be considered as 13 downto 0 vector(4 times 3 +green and yellow of light5).



A sample drawing for this intersection design:



Conclusion:

The term project was successful and desired designed was achieved.This Project made me confident about VHDL,clock usage,counters,FSM and sensor usage in a digital design.

Appendices:

Code of Main Project:

```
library IEEE;

use IEEE.STD_LOGIC_1164.ALL;

    use ieee.std_logic_unsigned.all;




entity traffic is

port( clk: IN STD_LOGIC;

                        rst: IN STD_LOGIC;

                         red1:out std_logic;

       yellow1:out std_logic;

       green1:out std_logic;

       red2:out std_logic;

       yellow2:out std_logic;

       green2:out std_logic;

       red3:out std_logic;

       yellow3:out std_logic;

       green3:out std_logic;

       red4:out std_logic;

       yellow4:out std_logic;

       green4:out std_logic;

          red5:out std_logic;

           green5:out std_logic;

       pir :in std_logic;

       outputpir: inout std_logic
```

```
                    );

end traffic;

architecture Behavioral of traffic is



TYPE stateX IS (dummy1,s0,s1, dummy2, s2,s3,dummy3,s4,s5, dummy4,s6,s7);
signal current_state :stateX;
signal next_state: stateX:=s0 ;
signal counter: std_logic_vector(35 downto 0):= x"000000000";

begin
outputpir <= not(pir);

Process(clk)
begin
if (rising_edge(clk))
                    then counter <= counter + 1;
                              if ( counter = x"FFFFFFFFF")
                              then counter <= x"000000000";
                              end if;
          end if;
end process;





process(counter(30), rst)
begin
if(rst='1')
```

```
then current_state<=s0;
elsif(rising_edge(counter(30)))
then current_state<=next_state;
end if;
end process;

process(current_state,outputpir)
begin
case current_state is

when s0=>
if outputpir = '1' then
next_state<=s4;
green1<='0';
      green2<='0';
                  green3<='1';
                  green4<='0';

                  yellow1<='0';
                  yellow2<='0';
                  yellow3<='0';
                  yellow4<='0';

                  red1<='1';
                  red2<='1';
                  red3<='0';
                  red4<='1';

                  green5<= '1';
                  red5 <= '0';
else
```

```
                        next_state<=s1;
                        green1<='1';
        green2<='0';
                        green3<='0';
                        green4<='0';

                        yellow1<='0';
                        yellow2<='0';
                        yellow3<='0';
                        yellow4<='0';

                        red1<='0';
                        red2<='1';
                        red3<='1';
                        red4<='1';

                        green5<= '0';
                        red5 <= '1';
                        end if;
when s1=>
if outputpir = '1' then
next_state<=s4;
green1<='0';
        green2<='0';
                        green3<='1';
                        green4<='0';

                        yellow1<='0';
                        yellow2<='0';
                        yellow3<='0';
                        yellow4<='0';
                        red1<='0';
                        red2<='1';
                        red3<='0';
                        red4<='1';

                        green5<= '1';
                        red5 <= '0';
                        else
                        next_state<=dummy2;
                        green1<='1';
        green2<='0';
                        green3<='0';
                        green4<='0';

                        yellow1<='0';
                        yellow2<='0';
                        yellow3<='0';
                        yellow4<='0';

                        red1<='0';
                        red2<='1';
                        red3<='1';
                        red4<='1';

                        green5<= '0';
                        red5 <= '1';
                        end if;

    when dummy2=>
    if outputpir = '1' then
```

```
next_state<=s4;
green1<='0';
	green2<='0';
			green3<='1';
			green4<='0';

			yellow1<='0';
			yellow2<='0';
			yellow3<='0';
			yellow4<='0';

			red1<='0';
			red2<='1';
			red3<='0';
			red4<='1';

			green5<= '1';
			red5 <= '0';
			else
			next_state<=s2;
			green1<='1';
	green2<='0';
			green3<='0';
			green4<='0';

			yellow1<='0';
			yellow2<='1';
			yellow3<='0';
			yellow4<='0';

			red1<='0';
```

```
                    red2<='0';
                    red3<='1';
                    red4<='1';

                    green5<= '0';
                    red5 <= '1';
                    end if;

when s2=>
if outputpir = '1' then
next_state<=s4;
green1<='0';
      green2<='0';
                    green3<='1';
                    green4<='0';

                    yellow1<='0';
                    yellow2<='0';
                    yellow3<='0';
                    yellow4<='0';

                    red1<='0';
                    red2<='1';
                    red3<='0';
                    red4<='1';

                    green5<= '1';
                    red5 <= '0';
                    else
                    next_state<=s3;
                    green1<='0';
```

```
green2<='1';
                        green3<='0';
                        green4<='0';

                        yellow1<='0';
                        yellow2<='0';
                        yellow3<='0';
                        yellow4<='0';

                        red1<='1';
                        red2<='0';
                        red3<='1';
                        red4<='1';
                        green5<= '0';
                        red5 <= '1';
                        end if;

when s3=>
if outputpir = '1' then
next_state<=s4;
green1<='0';
       green2<='0';
                        green3<='1';
                        green4<='0';

                        yellow1<='0';
                        yellow2<='0';
                        yellow3<='0';
                        yellow4<='0';

                        red1<='0';
```

```
                    red2<='1';
                    red3<='0';
                    red4<='1';

                    green5<= '1';
                    red5 <= '0';
                    else
                    next_state<=dummy3;
                    green1<='0';
green2<='1';
                    green3<='0';
                    green4<='0';

                    yellow1<='0';
                    yellow2<='0';
                    yellow3<='0';
                    yellow4<='0';

                    red1<='1';
                    red2<='0';
                    red3<='1';
                    red4<='1';

                    green5<= '0';
                    red5 <= '1';
                    end if;

when dummy3=>


                    next_state<=s4;
```

```
                green1<='0';
green2<='1';
                green3<='0';
                green4<='0';

                yellow1<='0';
                yellow2<='0';
                yellow3<='1';
                yellow4<='0';

                red1<='1';
                red2<='0';
                red3<='0';
                red4<='1';

                green5<= '1';
                red5 <= '0';
when s4=>

                next_state<=s5;
                green1<='0';
green2<='0';
                green3<='1';
                green4<='0';

                yellow1<='0';
                yellow2<='0';
                yellow3<='0';
                yellow4<='0';

                red1<='1';
```

```
                        red2<='1';

                        red3<='0';

                        red4<='1';


                        green5<= '1';

                        red5 <= '0';


when s5=>

if outputpir = '1' then

next_state<=s4;

green1<='0';

       green2<='0';

                        green3<='1';

                        green4<='0';


                        yellow1<='0';

                        yellow2<='0';

                        yellow3<='0';

                        yellow4<='0';


                        red1<='0';

                        red2<='1';

                        red3<='0';

                        red4<='1';


                        green5<= '1';

                        red5 <= '0';

                        else

                        next_state<=dummy4;

                        green1<='0';

green2<='0';
```

```
                              green3<='1';
                              green4<='0';

                              yellow1<='0';
                              yellow2<='0';
                              yellow3<='0';
                              yellow4<='0';

                              red1<='1';
                              red2<='1';
                              red3<='0';
                              red4<='1';

                              green5<= '0';
                              red5 <= '1';
                              end if;



when dummy4=>
if outputpir = '1' then
next_state<=s4;
green1<='0';
        green2<='0';
                              green3<='1';
                              green4<='0';

                              yellow1<='0';
                              yellow2<='0';
                              yellow3<='0';
                              yellow4<='0';
```

```
                    red1<='0';
                    red2<='1';
                    red3<='0';
                    red4<='1';

                    green5<= '1';
                    red5 <= '0';
                    else
                    next_state<=s6;
                    green1<='0';
green2<='0';
                    green3<='1';
                    green4<='0';

                    yellow1<='0';
                    yellow2<='0';
                    yellow3<='0';
                    yellow4<='1';

                    red1<='1';
                    red2<='1';
                    red3<='0';
                    red4<='0';

                    green5<= '0';
                    red5 <= '1';
                    end if;


when s6=>
if outputpir = '1' then
```

```
next_state<=s4;
green1<='0';
      green2<='0';
                     green3<='1';
                     green4<='0';

                     yellow1<='0';
                     yellow2<='0';
                     yellow3<='0';
                     yellow4<='0';

                     red1<='0';
                     red2<='1';
                     red3<='0';
                     red4<='1';
                     green5<= '1';
                     red5 <= '0';
                     else
                     next_state<=s7;
                     green1<='0';
green2<='0';
                     green3<='0';
                     green4<='1';

                     yellow1<='0';
                     yellow2<='0';
                     yellow3<='0';
                     yellow4<='0';

                     red1<='1';
                     red2<='1';
```

```
                         red3<='1';

                         red4<='0';


                         green5<= '0';

                         red5 <= '1';


                         end if;


when s7=>

if outputpir = '1' then

next_state<=s4;

green1<='0';

       green2<='0';

                         green3<='1';

                         green4<='0';


                         yellow1<='0';

                         yellow2<='0';

                         yellow3<='0';

                         yellow4<='0';


                         red1<='0';

                         red2<='1';

                         red3<='0';

                         red4<='1';

                         green5<= '1';

                         red5 <= '0';

                         else

                         next_state<=dummy1;

                         green1<='0';

green2<='0';
```

```
                    green3<='0';

                    green4<='1';


                    yellow1<='0';

                    yellow2<='0';

                    yellow3<='0';

                    yellow4<='0';


                    red1<='1';

                    red2<='1';

                    red3<='1';

                    red4<='0';


                    green5<= '0';

                    red5 <= '1';

                    end if;

when dummy1=>

if outputpir = '1' then

next_state<=s4;

green1<='0';

      green2<='0';

                    green3<='1';

                    green4<='0';


                    yellow1<='0';

                    yellow2<='0';

                    yellow3<='0';

                    yellow4<='0';


                    red1<='0';

                    red2<='1';
                    red3<='0';
                    red4<='1';

                    green5<= '1';
                    red5 <= '0';
                    else
                    next_state<=s0;
                    green1<='0';
        green2<='0';
                    green3<='0';
                    green4<='1';

                    yellow1<='1';
                    yellow2<='0';
                    yellow3<='0';
                    yellow4<='0';

                    red1<='0';
                    red2<='1';
                    red3<='1';
                    red4<='0';

                    green5<= '0';
                    red5 <= '1';
                    end if;



end case;
end process;
end Behavioral;
```

Code for constraints:

```
set_property PACKAGE_PIN W5 [get_ports clk]
set_property IOSTANDARD LVCMOS33 [get_ports clk]
set_property PACKAGE_PIN R2 [get_ports rst]
set_property IOSTANDARD LVCMOS33 [get_ports rst]

set_property PACKAGE_PIN A14 [get_ports {red1}]
set_property IOSTANDARD LVCMOS33 [get_ports {red1}]
set_property PACKAGE_PIN A16 [get_ports {yellow1}]
set_property IOSTANDARD LVCMOS33 [get_ports {yellow1}]
set_property PACKAGE_PIN B15 [get_ports {green1}]
set_property IOSTANDARD LVCMOS33 [get_ports {green1}]

set_property PACKAGE_PIN A15 [get_ports {red2}]
set_property IOSTANDARD LVCMOS33 [get_ports {red2}]
set_property PACKAGE_PIN A17 [get_ports {yellow2}]
set_property IOSTANDARD LVCMOS33 [get_ports {yellow2}]
set_property PACKAGE_PIN C15 [get_ports {green2}]
set_property IOSTANDARD LVCMOS33 [get_ports {green2}]

set_property PACKAGE_PIN K17 [get_ports {red3}]
set_property IOSTANDARD LVCMOS33 [get_ports {red3}]
set_property PACKAGE_PIN M18 [get_ports {yellow3}]
set_property IOSTANDARD LVCMOS33 [get_ports {yellow3}]
set_property PACKAGE_PIN N17 [get_ports {green3}]
set_property IOSTANDARD LVCMOS33 [get_ports {green3}]
set_property PACKAGE_PIN L17 [get_ports {red4}]

set_property IOSTANDARD LVCMOS33 [get_ports {red4}]

set_property PACKAGE_PIN M19 [get_ports {yellow4}]

set_property IOSTANDARD LVCMOS33 [get_ports {yellow4}]

set_property PACKAGE_PIN P17 [get_ports {green4}]

set_property IOSTANDARD LVCMOS33 [get_ports {green4}]

set_property PACKAGE_PIN P18 [get_ports red5]

set_property IOSTANDARD LVCMOS33 [get_ports red5]

set_property PACKAGE_PIN R18 [get_ports green5]

set_property IOSTANDARD LVCMOS33 [get_ports green5]




 set_property PACKAGE_PIN C16 [get_ports {pir}]

set_property IOSTANDARD LVCMOS33 [get_ports {pir}]


set_property PACKAGE_PIN  U16 [get_ports {outputpir}]

set_property IOSTANDARD LVCMOS33 [get_ports {outputpir}]
```