

**14-6**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity TL is

port(r,clk,censorew, censorns: in std\_logic; --This instruction will have the sensor ffor the east west and north south

light: out std\_logic\_vector(2 downto 0)); --Green, yellow and red for all the directions

end;

architecture beh of TL is

type state is (nsgreen,nsyellow,nsred,ewgreen,ewyellow,ewred);-- the state for each light color in each direction

signal cs,ns: state; -- current state, and next state

begin

process(r,clk)

begin

if(r = '1') then --reset

cs<=nsgreen;--to the original state

elsif(clk'event AND clk = '1') then

cs<= ns;

end if;

end process;

process(cs,censorns,censorew)

begin

case cs is

when nsgreen =>

if(censorew = '1' and (censorns = '0' or censorns = '1')) then

ns <= nsyellow;--doesnt change the light

else

ns <= nsgreen;-- stays at the same light

end if;

light<= "000";--the light will be green

when nsyellow =>--auto transition so input doesnt matter

ns<=nsred;

light<= "001";--light will be yellow

when nsred =>--input doesnt matter

ns<= ewgreen;

light<= "010";--light will be yellow

when ewgreen =>

if(censorns = '1' and (censorew = '0' or censorew = '1')) then

ns <= ewyellow;--doesnt change the light

else

ns <= ewgreen;-- stays at the same light

end if;

light<= "011";--light will be red

when ewyellow =>--input doesnt matter

ns<= ewred;

light<= "100";

when ewred =>---input doesnt matter

ns<= nsgreen;

light<= "101";

end case;

end process;

end beh;

**14-6 test bench**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity tb2 is

end;

architecture beh of tb2 is

-

component TL is

port(r, clk, censorew, csensorns : in std\_logic;--same port as 14.6

light : out std\_logic\_vector(2 downto 0)); --codes for all the lights and their direcitons of green yelllow and red

end component;

signal r, clk, censorew, censorns : std\_logic;

signal light : std\_logic\_vector(2 downto 0);

begin

TL0: TL port map(r, clk, censorew, censorns,light);

sim : process

begin

r<='0';clk<='1';censorew<='1';censorns<='0';

wait for 100ns;

r<='0';clk<='0';censorew<='1';censorns<='0';

wait for 50ns;

r<='0';clk<='1';censorew<='1';censorns<='0';

wait for 100ns;

r<='0';clk<='0';censorew<='1';censorns<='0';

wait for 50ns;

r<='0';clk<='1';censorew<='1';censorns<='0';

wait for 100ns;

r<='0';clk<='0';censorew<='0';censorns<='1';

wait for 50ns;

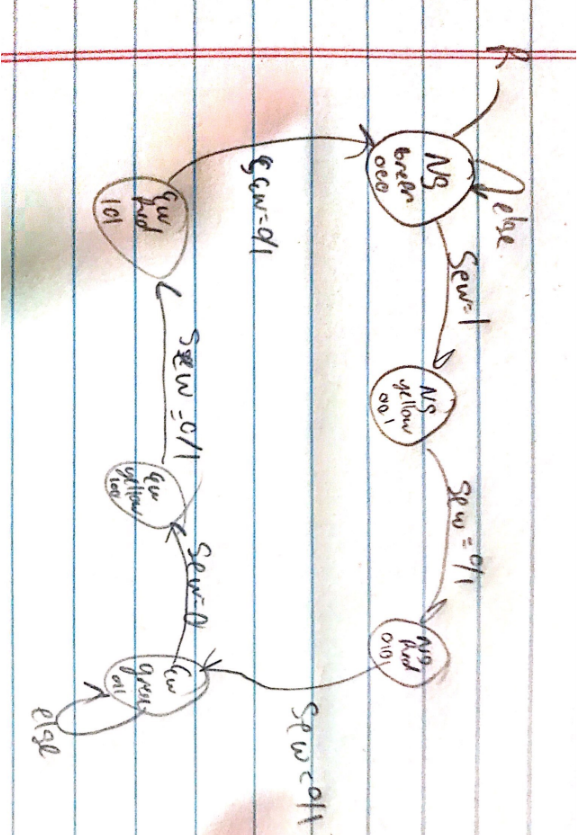
r<='0';clk<='1';censorew<='0';censorns<='1';

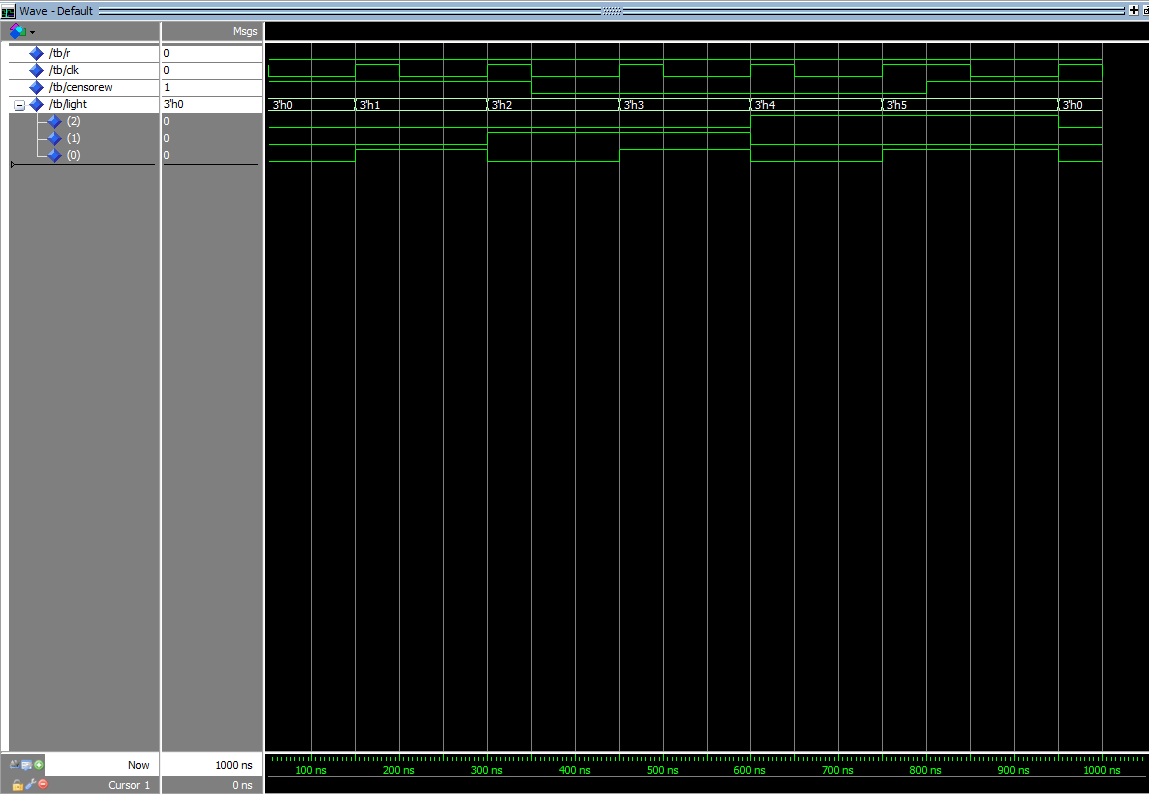
wait for 100ns;

r<='0';clk<='0';censorew<='0';censorns<='1';

wait for 50ns;

end process;

end beh; 



**14-9**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

--Traffic Light in problem 14.9

entity TL2 is

port(r, clk, censorew: in std\_logic;--(clock, reset, carsensornorthsouth/carsensoreastwest)

light : out std\_logic\_vector(2 downto 0)); --codes for lights (nsg, nsy, nsr, ewg, ewy, ewr)

end;

architecture beh of TL2 is

type state is (nsgreen, nsyellow, nsred, ewgreen, ewyellow, ewred);--directions

signal cs, ns :state;--current state, next state

begin

process(r,clk)

begin

if(r = '1') then --reset

cs<=nsgreen;--takes the sequence back to its original

elsif(clk'event AND clk = '1') then

cs<= ns;

end if;

end process;

process(cs,censorew)

begin

case cs is

when nsgreen =>

if(censorew = '1' ) then--if a car is detected

ns <= nsyellow;--next light in the sequence

else

ns <= nsgreen;-- stay light at the current light

end if;

light<= "000";--light changes here

when nsyellow =>--input doesnt matter

ns<=nsred;

light<= "001";

when nsred =>--inputter doesnt matter

ns<= ewgreen;

light<= "010";--light changes here

when ewgreen =>

if(censorew = '0') then-- if a car is detected

ns <= ewyellow;--next light needed

else

ns <= ewgreen;-- stay at the same light

end if;

light<= "011";--light changes here

when ewyellow =>--input doesnt matter

ns<= ewred;

light<= "100";

when ewred =>--input doesnt matter

ns<= nsgreen;

light<= "101";

end case;

end process;

end beh;

**14-9 test bench**

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity tb is

end;

architecture beh of tb is

--component of the traffic light

component TL2 is

port(r, clk, censorew : in std\_logic;--same port as 14.9

light : out std\_logic\_vector(2 downto 0)); --codes for the lights and direction from green,yellow, and red

end component;

--input signals

signal r, clk, censorew: std\_logic;

--output signals

signal light : std\_logic\_vector(2 downto 0);

begin

TL1: TL2 port map(r, clk, censorew, light);

sim : process

begin

r<='0';clk<='1';censorew<='1';

wait for 50ns;

r<='0';clk<='0';censorew<='1';

wait for 100ns;

r<='0';clk<='1';censorew<='1';

wait for 50ns;

r<='0';clk<='0';censorew<='1';

wait for 100ns;

r<='0';clk<='1';censorew<='1';

wait for 50ns;

r<='0';clk<='0';censorew<='0';

wait for 100ns;

r<='0';clk<='1';censorew<='0';

wait for 50ns;

r<='0';clk<='0';censorew<='0';

wait for 100ns;

r<='0';clk<='1';censorew<='0';

wait for 50ns;

r<='0';clk<='0';censorew<='0';

wait for 100ns;

r<='0';clk<='1';censorew<='0';

wait for 50ns;

end process;

end beh;

youtube-<https://www.youtube.com/watch?v=tYa4tBkGzac>

<https://www.youtube.com/watch?v=e-jZ-YJtr9g>