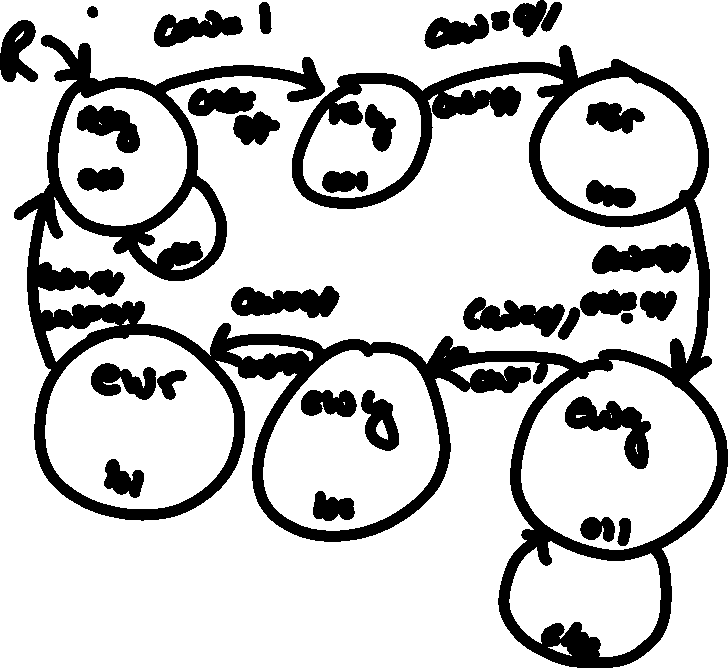
HOMEWORK 8

Traffic Light 14.6 State Diagram



Traffic Light 14.6 VHDL

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

--Traffic Light in problem 14.6

entity TrafficLight is

port(r, clk, cew, cns : in std\_logic;--(clock, reset, carnorthsouth. careastwest)

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

end;

architecture beh of TrafficLight is

type state is (nsg, nsy, nsr, ewg, ewy, ewr);--north south green, north south yellow etc..

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

begin

process(r,clk)

begin

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

cs<=nsg;--to the original state

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

cs<= ns;

end if;

end process;

process(cs,cns,cew)

begin

case cs is

when nsg =>

if(cew = '1' and (cns = '0' or cns = '1')) then

ns <= nsy;--next light

else

ns <= nsg;-- stay light

end if;

light<= "000";

when nsy =>--auto transition so input dosn't matter

ns<=nsr;

light<= "001";

when nsr =>--auto transition so input dosn't matter

ns<= ewg;

light<= "010";

when ewg =>

if(cns = '1' and (cew = '0' or cew = '1')) then

ns <= ewy;--next light

else

ns <= ewg;-- stay light

end if;

light<= "011";

when ewy =>--auto transition so input dosn't matter

ns<= ewr;

light<= "100";

when ewr =>--auto transition so input dosn't matter

ns<= nsg;

light<= "101";

end case;

end process;

end beh;

Test Bench 14.6

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity tbhw8 is

end;

architecture beh of tbhw8 is

--component dec

component TrafficLight is

port(r, clk, cew, cns : in std\_logic;--(clock, reset, carnorthsouth. careastwest)

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

end component;

--input signals

signal r, clk, cew, cns : std\_logic;

--output signals

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

begin

--instantiation

TL: TrafficLight port map(r, clk, cew, cns,light);

sim : process

begin

r<='0';clk<='1';cew<='1';cns<='0'; wait for 100ns;

r<='0';clk<='0';cew<='1';cns<='0'; wait for 100ns;

r<='0';clk<='1';cew<='1';cns<='0'; wait for 100ns;

r<='0';clk<='0';cew<='1';cns<='0'; wait for 100ns;

r<='0';clk<='1';cew<='1';cns<='0'; wait for 100ns;

r<='0';clk<='0';cew<='0';cns<='1'; wait for 100ns;

r<='0';clk<='1';cew<='0';cns<='1'; wait for 100ns;

r<='0';clk<='0';cew<='0';cns<='1'; wait for 100ns;

r<='0';clk<='1';cew<='0';cns<='1'; wait for 100ns;

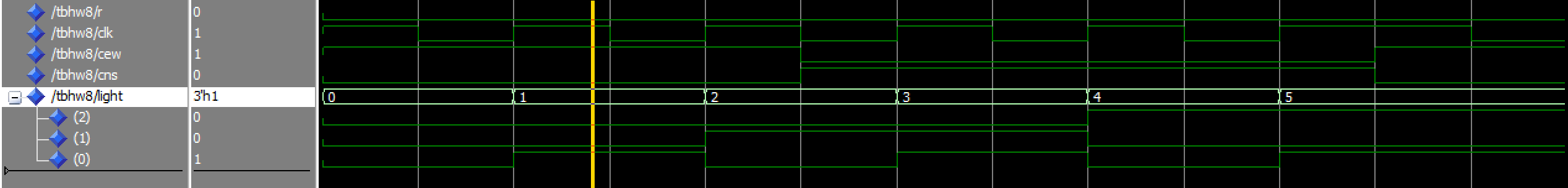
r<='0';clk<='0';cew<='0';cns<='1'; wait for 100ns;

r<='0';clk<='1';cew<='0';cns<='1'; wait for 100ns;

end process;

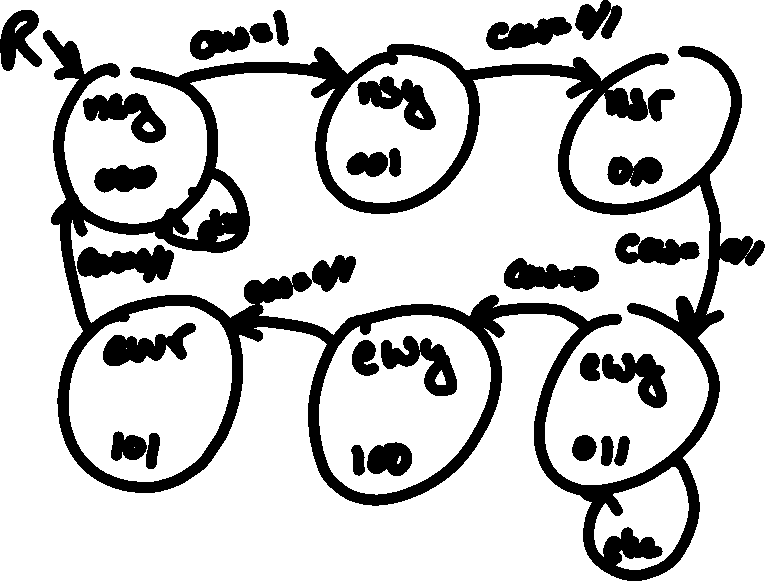
end beh;

Simulation 14.6



NOTE: The numbers 0-5 are the 6 different states of the traffic lights

State Diagram 14.9



Traffic Light 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 TrafficLight2 is

port(r, clk, cew: in std\_logic;--(clock, reset, carnorthsouth. careastwest)

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

end;

architecture beh of TrafficLight2 is

type state is (nsg, nsy, nsr, ewg, ewy, ewr);--north south green, north south yellow etc..

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

begin

process(r,clk)

begin

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

cs<=nsg;--to the original state

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

cs<= ns;

end if;

end process;

process(cs,cew)

begin

case cs is

when nsg =>

if(cew = '1' ) then

ns <= nsy;--next light

else

ns <= nsg;-- stay light

end if;

light<= "000";

when nsy =>--auto transition so input dosn't matter

ns<=nsr;

light<= "001";

when nsr =>--auto transition so input dosn't matter

ns<= ewg;

light<= "010";

when ewg =>

if(cew = '0') then

ns <= ewy;--next light

else

ns <= ewg;-- stay light

end if;

light<= "011";

when ewy =>--auto transition so input dosn't matter

ns<= ewr;

light<= "100";

when ewr =>--auto transition so input dosn't matter

ns<= nsg;

light<= "101";

end case;

end process;

end beh;

Test Bench 14.9

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity tbhw82 is

end;

architecture beh of tbhw82 is

--component dec

component TrafficLight2 is

port(r, clk, cew : in std\_logic;--(clock, reset, carnorthsouth. careastwest)

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

end component;

--input signals

signal r, clk, cew: std\_logic;

--output signals

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

begin

--instantiation

TL: TrafficLight2 port map(r, clk, cew, light);

sim : process

begin

r<='0';clk<='1';cew<='1'; wait for 100ns;

r<='0';clk<='0';cew<='1'; wait for 100ns;

r<='0';clk<='1';cew<='1'; wait for 100ns;

r<='0';clk<='0';cew<='1'; wait for 100ns;

r<='0';clk<='1';cew<='1'; wait for 100ns;

r<='0';clk<='0';cew<='0'; wait for 100ns;

r<='0';clk<='1';cew<='0'; wait for 100ns;

r<='0';clk<='0';cew<='0'; wait for 100ns;

r<='0';clk<='1';cew<='0'; wait for 100ns;

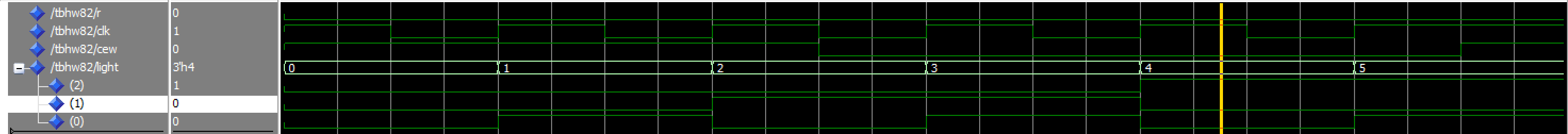
r<='0';clk<='0';cew<='0'; wait for 100ns;

r<='0';clk<='1';cew<='0'; wait for 100ns;

end process;

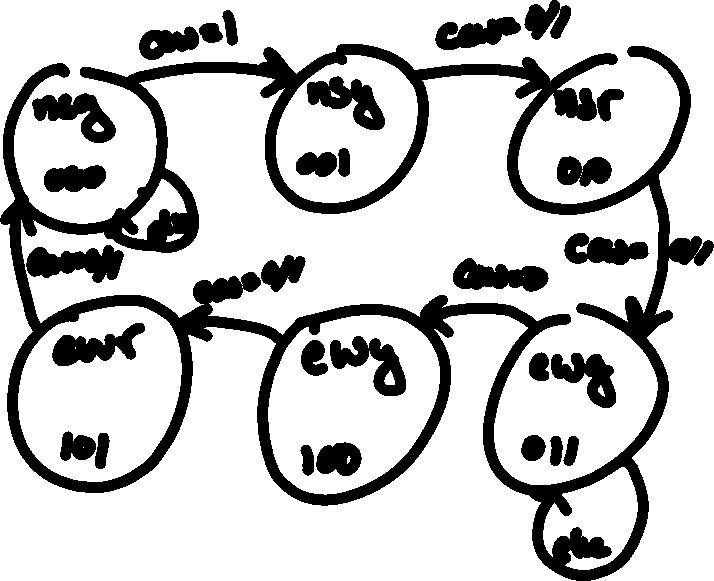
end beh;

Simulation 14.9



NOTE: The numbers 0-5 are the 6 different states of the traffic lights

State Diagram 14.9 with crosswalk



Traffic Light 14.9 with crosswalk

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 with a cross walk

entity TrafficLight3 is

port(r, clk, cew, c : in std\_logic;--(clock, reset, careastwest, crosswalk )

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

end;

architecture beh of TrafficLight3 is

type state is (nsg, nsy, nsr, ewg, ewy, ewr);--north south green, north south yellow etc..

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

begin

process(r,clk)

begin

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

cs<=nsg;--to the original state

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

cs<= ns;

end if;

end process;

process(cs,c,cew)

begin

case cs is

when nsg =>

if(cew = '1' and c = '0') then

ns <= nsy;--next light

else

ns <= nsg;-- stay light

end if;

light<= "000";

when nsy =>--auto transition so input dosn't matter

ns<=nsr;

light<= "001";

when nsr =>--auto transition so input dosn't matter

ns<= ewg;

light<= "010";

when ewg =>

if(cew = '0' and (c = '0' or c = '1')) then

ns <= ewy;--next light

else

ns <= ewg;-- stay light

end if;

light<= "011";

when ewy =>--auto transition so input dosn't matter

ns<= ewr;

light<= "100";

when ewr =>--auto transition so input dosn't matter

ns<= nsg;

light<= "101";

end case;

end process;

end beh;

Test Bench 14.9 with crosswalk

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_arith.all;

use ieee.std\_logic\_unsigned.all;

entity tbhw83 is

end;

architecture beh of tbhw83 is

--component dec

component TrafficLight3 is

port(r, clk, cew, c : in std\_logic;--(clock, reset, careastwest, crosswalk)

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

end component;

--input signals

signal r, clk, cew, c: std\_logic;

--output signals

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

begin

--instantiation

TL: TrafficLight3 port map(r, clk, cew, c,light);

sim : process

begin

r<='0';clk<='1';cew<='1';c<='0'; wait for 100ns;

r<='0';clk<='0';cew<='1';c<='0'; wait for 100ns;

r<='0';clk<='1';cew<='1';c<='0'; wait for 100ns;

r<='0';clk<='0';cew<='1';c<='0'; wait for 100ns;

r<='0';clk<='1';cew<='1';c<='0'; wait for 100ns;

r<='0';clk<='0';cew<='0';c<='1'; wait for 100ns;

r<='0';clk<='1';cew<='0';c<='1'; wait for 100ns;

r<='0';clk<='0';cew<='0';c<='1'; wait for 100ns;

r<='0';clk<='1';cew<='0';c<='1'; wait for 100ns;

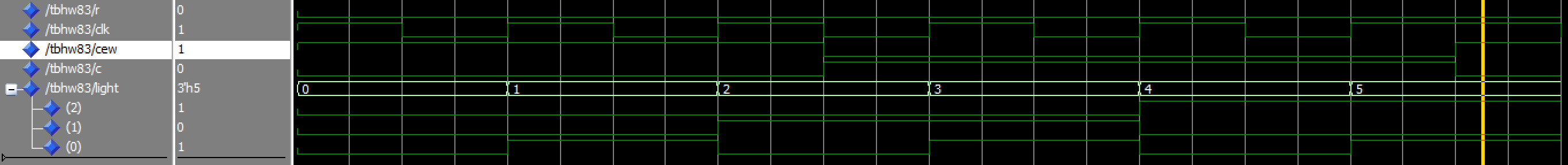
r<='0';clk<='0';cew<='0';c<='1'; wait for 100ns;

r<='0';clk<='1';cew<='0';c<='1'; wait for 100ns;

end process;

end beh;

Simulation 14.9 with crosswalk



NOTE: The numbers 0-5 are the 6 different states of the traffic lights

Videos:

*I tried to upload to YouTube and it wouldn’t work*

*So here is a link to my google drive folder with the videos in it*

https://drive.google.com/open?id=1oQ6EOLb70DrzD1gFBAHaumcKWLaIUaY2