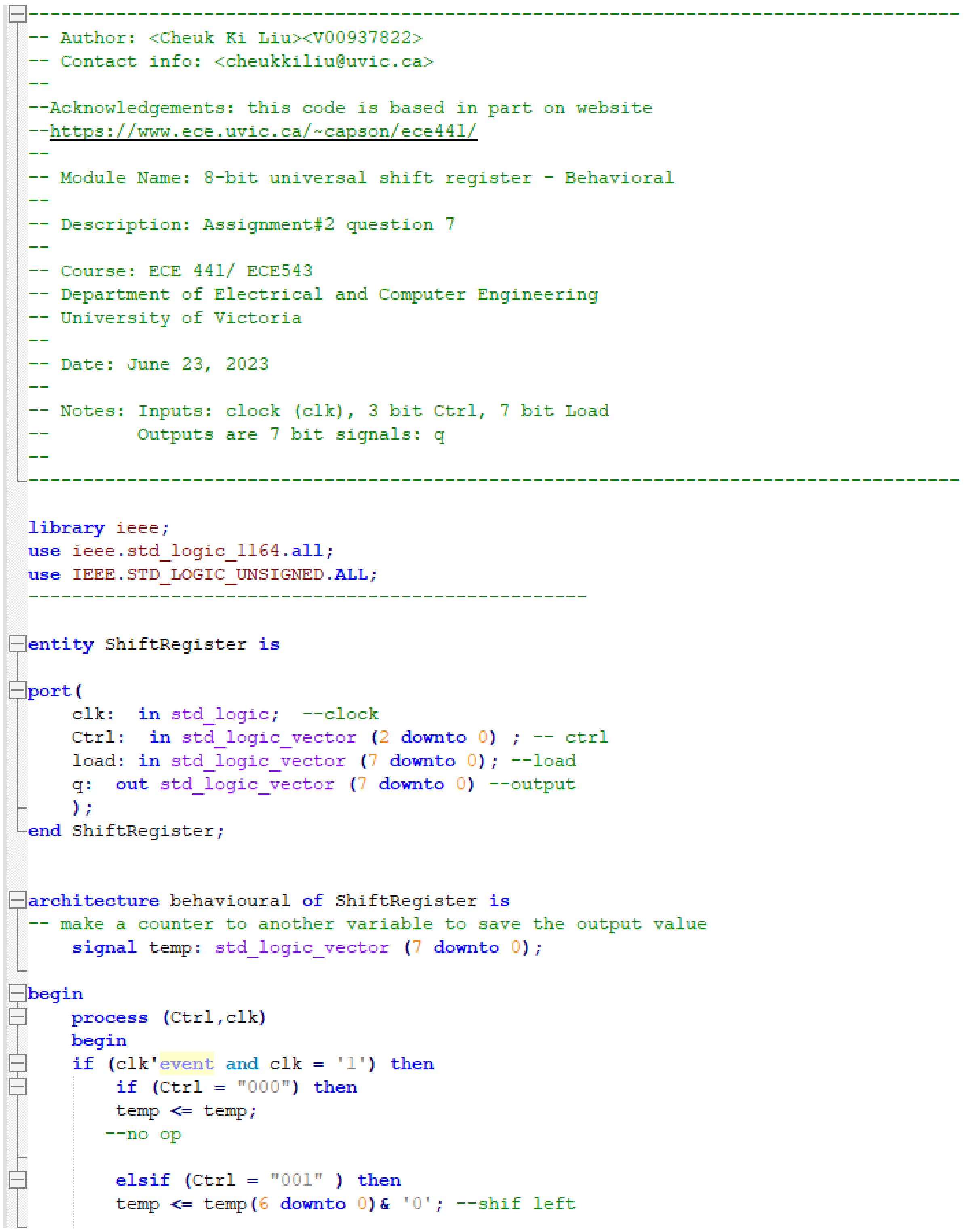
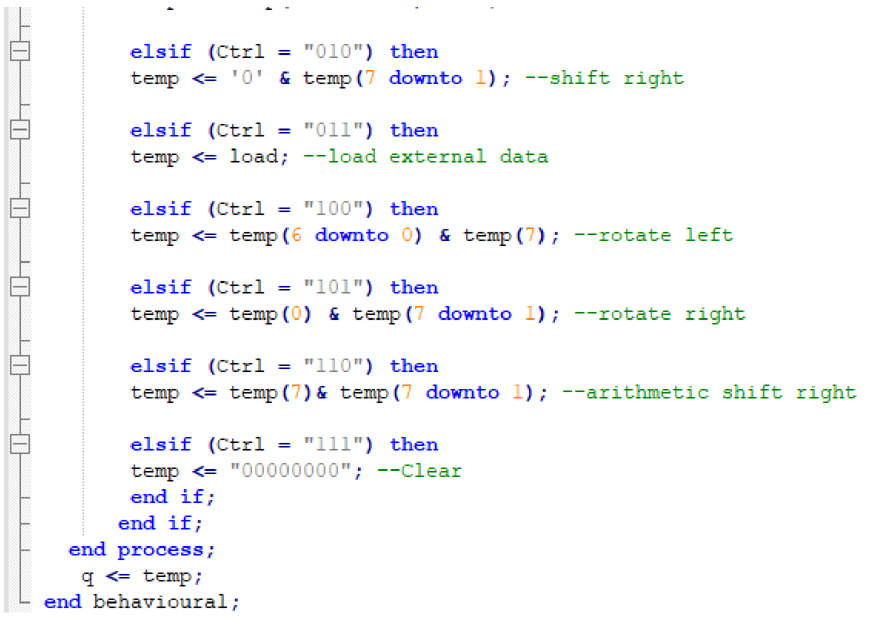
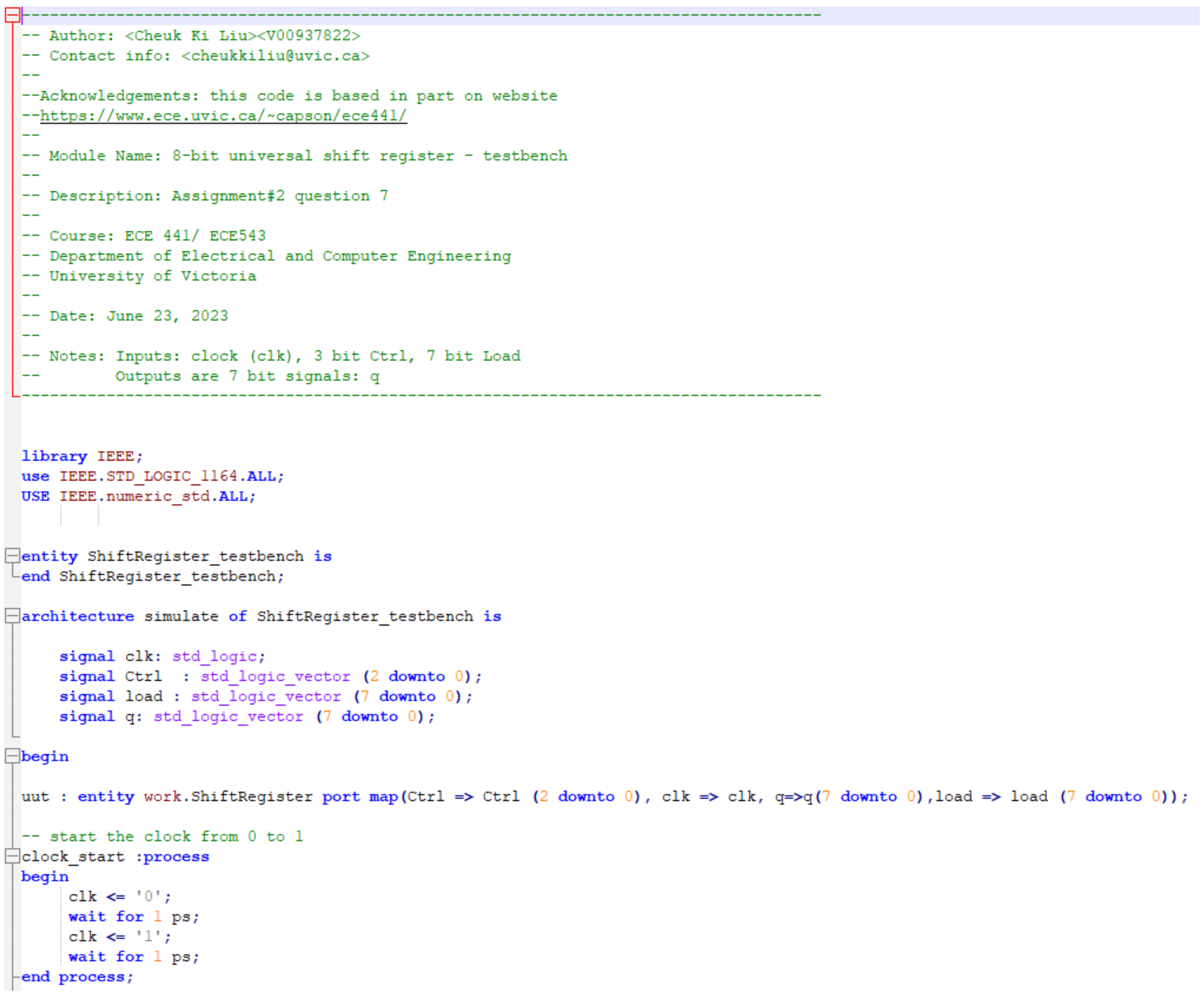
Question 7



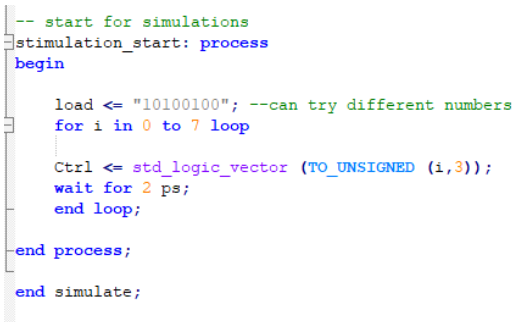
```
--------------------------------------------------------------------------------
-- Author: <Cheuk Ki Liu><V00937822>
-- Contact info: <cheukkiliu@uvic.ca>
--
--Acknowledgements: this code is based in part on website
--https://www.ece.uvic.ca/~capson/ece441/
--
-- Module Name: 8-bit universal shift register - Behavioral
--
-- Description: Assignment#2 question 7
--
-- Course: ECE 441/ ECE543
-- Department of Electrical and Computer Engineering
-- University of Victoria
--
-- Date: June 23, 2023
--
-- Notes: Inputs: clock (clk), 3 bit Ctrl, 7 bit Load
--        Outputs are 7 bit signals: q
--
--------------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
---------------------------------------------------

entity ShiftRegister is

port(
    clk:  in std_logic;  --clock
    Ctrl:  in std_logic_vector (2 downto 0)  ; -- ctrl
    load: in std_logic_vector (7 downto 0); --load
    q:  out std_logic_vector (7 downto 0) --output
    );
end ShiftRegister;


architecture behavioural of ShiftRegister is
-- make a counter to another variable to save the output value
    signal temp: std_logic_vector (7 downto 0);

begin
    process (Ctrl,clk)
    begin
    if (clk'event and clk = '1') then
        if (Ctrl = "000") then
        temp <= temp;
       --no op

        elsif (Ctrl = "001" ) then
        temp <= temp(6 downto 0)& '0'; --shif left
```



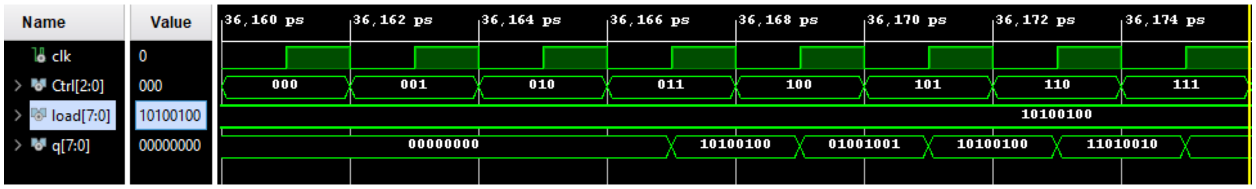
```vhdl
        elsif (Ctrl = "010") then
        temp <= '0' & temp(7 downto 1); --shift right

        elsif (Ctrl = "011") then
        temp <= load; --load external data

        elsif (Ctrl = "100") then
        temp <= temp(6 downto 0) & temp(7); --rotate left

        elsif (Ctrl = "101") then
        temp <= temp(0) & temp(7 downto 1); --rotate right

        elsif (Ctrl = "110") then
        temp <= temp(7)& temp(7 downto 1); --arithmetic shift right

        elsif (Ctrl = "111") then
        temp <= "00000000"; --Clear
        end if;
       end if;
   end process;
    q <= temp;
 end behavioural;
```



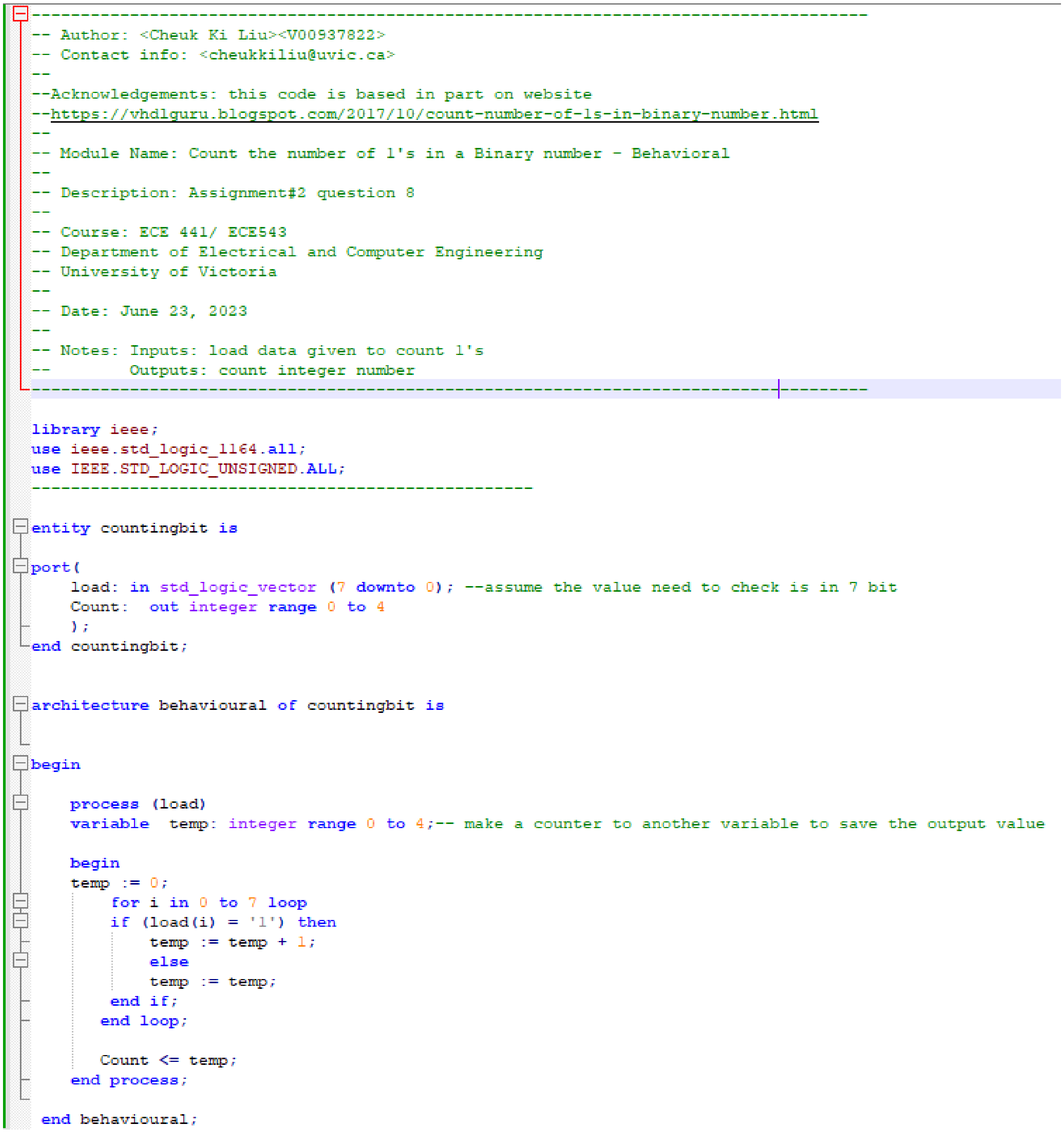
```vhdl
--------------------------------------------------------------------------------------
-- Author: <Cheuk Ki Liu><V00937822>
-- Contact info: <cheukkiliu@uvic.ca>
--
--Acknowledgements: this code is based in part on website
--https://www.ece.uvic.ca/~capson/ece441/
--
-- Module Name: 8-bit universal shift register - testbench
--
-- Description: Assignment#2 question 7
--
-- Course: ECE 441/ ECE543
-- Department of Electrical and Computer Engineering
-- University of Victoria
--
-- Date: June 23, 2023
--
-- Notes: Inputs: clock (clk), 3 bit Ctrl, 7 bit Load
--        Outputs are 7 bit signals: q
--------------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.numeric_std.ALL;


entity ShiftRegister_testbench is
end ShiftRegister_testbench;

architecture simulate of ShiftRegister_testbench is

    signal clk: std_logic;
    signal Ctrl  : std_logic_vector (2 downto 0);
    signal load : std_logic_vector (7 downto 0);
    signal q: std_logic_vector (7 downto 0);

begin

uut : entity work.ShiftRegister port map(Ctrl => Ctrl (2 downto 0), clk => clk, q=>q(7 downto 0),load => load (7 downto 0));

-- start the clock from 0 to 1
clock_start :process
begin
     clk <= '0';
     wait for 1 ps;
     clk <= '1';
     wait for 1 ps;
end process;
```



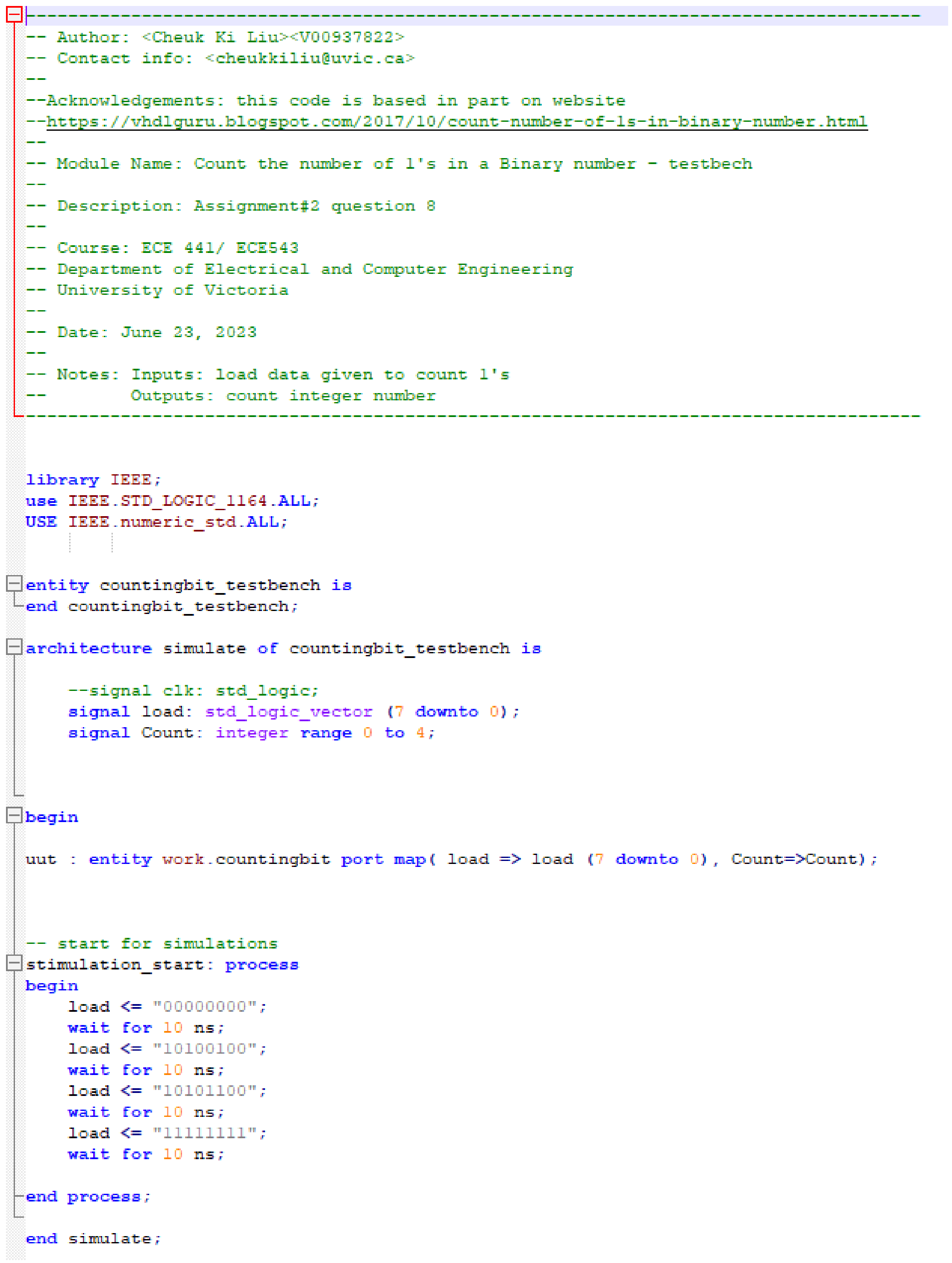
```
-- start for simulations
stimulation_start: process
begin

    load <= "10100100"; --can try different numbers
    for i in 0 to 7 loop

    Ctrl <= std_logic_vector (TO_UNSIGNED (i,3));
    wait for 2 ps;
    end loop;

end process;

end simulate;
```



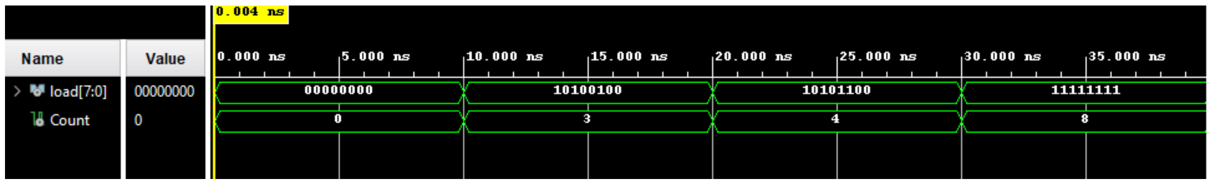
Question 8



```
-------------------------------------------------------------------------------------
-- Author: <Cheuk Ki Liu><V00937822>
-- Contact info: <cheukkiliu@uvic.ca>
--
--Acknowledgements: this code is based in part on website
--https://vhdlguru.blogspot.com/2017/10/count-number-of-1s-in-binary-number.html
--
-- Module Name: Count the number of 1's in a Binary number - Behavioral
--
-- Description: Assignment#2 question 8
--
-- Course: ECE 441/ ECE543
-- Department of Electrical and Computer Engineering
-- University of Victoria
--
-- Date: June 23, 2023
--
-- Notes: Inputs: load data given to count 1's
--        Outputs: count integer number
-------------------------------------------------------------------------------------

library ieee;
use ieee.std_logic_1164.all;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
---------------------------------------------------

entity countingbit is

port(
    load: in std_logic_vector (7 downto 0); --assume the value need to check is in 7 bit
    Count:  out integer range 0 to 4
    );
end countingbit;


architecture behavioural of countingbit is


begin

    process (load)
    variable  temp: integer range 0 to 4;-- make a counter to another variable to save the output value

    begin
    temp := 0;
        for i in 0 to 7 loop
        if (load(i) = '1') then
            temp := temp + 1;
            else
            temp := temp;
        end if;
       end loop;

       Count <= temp;
    end process;

 end behavioural;
```



```vhdl
----------------------------------------------------------------------------------
-- Author: <Cheuk Ki Liu><V00937822>
-- Contact info: <cheukkiliu@uvic.ca>
--
--Acknowledgements: this code is based in part on website
--https://vhdlguru.blogspot.com/2017/10/count-number-of-1s-in-binary-number.html
--
-- Module Name: Count the number of 1's in a Binary number - testbech
--
-- Description: Assignment#2 question 8
--
-- Course: ECE 441/ ECE543
-- Department of Electrical and Computer Engineering
-- University of Victoria
--
-- Date: June 23, 2023
--
-- Notes: Inputs: load data given to count 1's
--         Outputs: count integer number
----------------------------------------------------------------------------------


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
USE IEEE.numeric_std.ALL;


entity countingbit_testbench is
end countingbit_testbench;

architecture simulate of countingbit_testbench is

    --signal clk: std_logic;
    signal load: std_logic_vector (7 downto 0);
    signal Count: integer range 0 to 4;



begin

uut : entity work.countingbit port map( load => load (7 downto 0), Count=>Count);



-- start for simulations
stimulation_start: process
begin
    load <= "00000000";
    wait for 10 ns;
    load <= "10100100";
    wait for 10 ns;
    load <= "10101100";
    wait for 10 ns;
    load <= "11111111";
    wait for 10 ns;

end process;

end simulate;
```


0.004 ns
Name
Value
load[7:0]
00000000
Count
0
0.000 ns
5.000 ns
10.000 ns
15.000 ns
20.000 ns
25.000 ns
30.000 ns
35.000 ns
00000000
10100100
10101100
11111111
0
3
4
8