Instructions for Lab – 1

Wed. 11 March and Fri. 13 March (2 –4 pm)

[2 Marks]

Using ModelSim for Simulation of Digital Systems

1. The purpose of the first part of this lab is to learn how to use **ModelSim** simulation tool to perform functional simulation of the following digital circuit described in VHDL.

   a) There are syntax errors in the following code. Find and correct them.
   b) The given code should model an up counter, which counts even numbers from 0 to 14 periodically. Perform the functional simulation using the given testbench code (*test_counter.vhd*).
   c) Change the code to make it a new counter, which counts numbers which are multiple of 3 from 0 to 30 (such as 0, 3, 6, …, 30) as an up counter periodically. Do the simulation for the new code and explain if you need to change the given testbench code (*test_counter.vhd*) for this part.

```
entity counter is
      port (Clk, Reset : in bit;
            Q : out integer);
end entity counter;

architecture behaviour of counter is
begin
    process (Clk)
         variable v_Q: integer;
    begin
        if (Clk'event and Clk = 1) then
           if (Reset := ‘1’) then
              v_Q := 0;
           elsif (v_Q < 14) then
              v_Q := v_Q + 2;
           else
              v_Q := ‘0’;
           end if;
        end if;
        Q := v_Q;
     end process;
 end architecture behaviour;
```

➢ **Summary of Simulation Steps (in ModelSim)**:
   I. Change to the directory where your VHDL files are located.
   II. Create a working library (**work**)
   III. Compile your VHDL files.
   IV. Start Simulation (load the top level entity, which is ***test_counter***)
   V. Add signals to the waveform viewer.
   VI. Perform the simulation for at least **1 ms**.

File: **test_counter.vhd**

```
        -- Testbench for the counter.
        entity test_counter is
        end entity test_counter;

        architecture my_test of test_counter is
            signal t_clk, t_reset : bit;
            signal t_Q : integer;

            component counter is
                port (Clk, Reset : in bit;
                      Q : out integer);
            end component;
        begin
             DUT: counter port map (t_clk, t_reset, t_Q);

             -- Initialization process (code that executes only
        once).
             init: process
             begin
               -- reset pulse
               t_reset <= '0', '1' after 2 ns, '0' after 7 ns;
                 wait;
             end process init;

             -- clock generation
             clk_gen: process
             begin
                 wait for 5 ns;
                 t_clk <= '1';
                 wait for 5 ns;
                 t_clk <= '0';
             end process clk_gen;
        end architecture my_test;
```

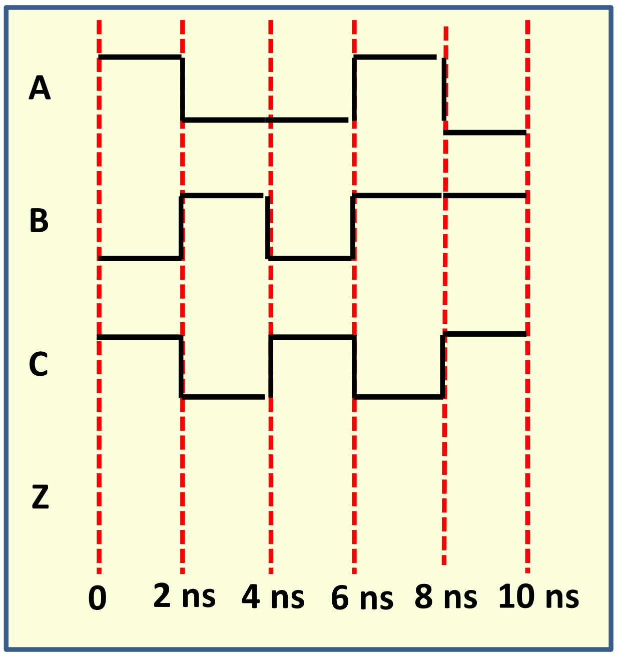
2. Write the VHDL code for a digital circuit, which generates the following output waveforms. Use the provided testbench (*test_wgen1.vhd* for part (a) and *test_wgen2.vhd* for part (b)) to test your VHDL code. Perform the simulation for at least 300 ns using **ModelSim** simulation tool.

(a) Output signals A, B, and C are periodic waves as shown below with a period of 10 ns and Z is determined based on the value of a 2-bit input signal ***D*** as follows:

Z :

A or B or C	**if D is "00"**
B xor C	**if D is "01"**
A and B	**if D is "10"**
'0'	**if D is "11''**

(b) Add an additional input signal called ***Enable*** to this system. Change your code from part (a) so when ***Enable*** is '0' all outputs are '0'. Otherwise, the outputs are similar to part (a).



File: **test_wgen1.vhd**

```
 -- A Testbech to test wgen1
 entity test_wgen1 is
 end entity test_wgen1;

 architecture test of test_wgen1 is
    signal t_A, t_B, t_C, t_Z : bit;
    signal t_D : bit_vector(1 downto 0);

    component wgen1 is
               port (D : in bit_vector(1 downto 0);
                     A, B, C, Z : out bit);
    end component;

    begin
      my_design: wgen1 port map (t_D, t_A, t_B, t_C, t_Z);

      -- Initialization process (code that executes only once).
      init: process
      begin
        t_D <= "00", "01" after 30 ns, "10" after 45 ns, "11" after 80 ns;
        wait;
      end process init;
end architecture test;
```

File: **test_wgen2.vhd**

```
 -- A Testbech to test wgen2
 entity test_wgen2 is
 end entity test_wgen2;

 architecture test of test_wgen2 is
    signal t_Enable, t_A, t_B, t_C, t_Z : bit;
    signal t_D : bit_vector(1 downto 0);

    component wgen2 is
               port (Enable : in bit;
                     D : in bit_vector(1 downto 0);
                     A, B, C, Z : out bit);
    end component;

    begin
      my_design: wgen2 port map (t_Enable, t_D, t_A, t_B, t_C, t_Z);

      -- Initialization process (code that executes only once).
      init: process
      begin
        -- enable signal
        t_Enable <= '1', '0' after 90 ns, '1' after 180 ns;
        t_D <= "00", "01" after 30 ns, "10" after 45 ns, "11" after 80 ns;
        wait;
      end process init;
end architecture test;
```