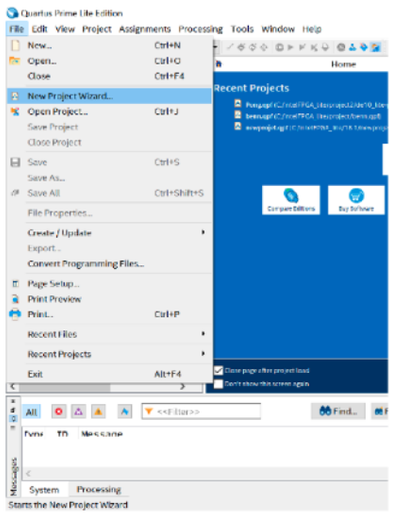
TP5

Exercise 1: Simulation of the AND gate with 03 inputs: F = A and B and C

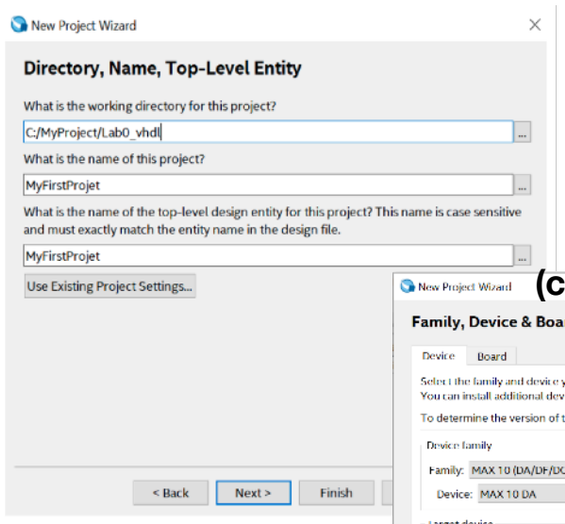
Follow these steps to simulate this function:

1. On Quartus, create a new project (Project Name: TP5) in your work directory, and select the Device Family as in the previous labs. In the **EDA Tools setting** page, select **ModelSim Altera** on **simulation line**

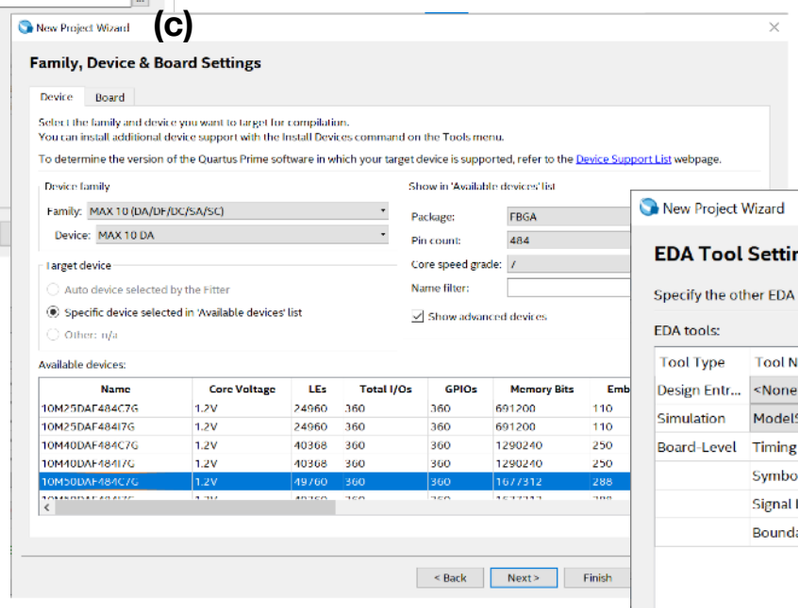
(a)



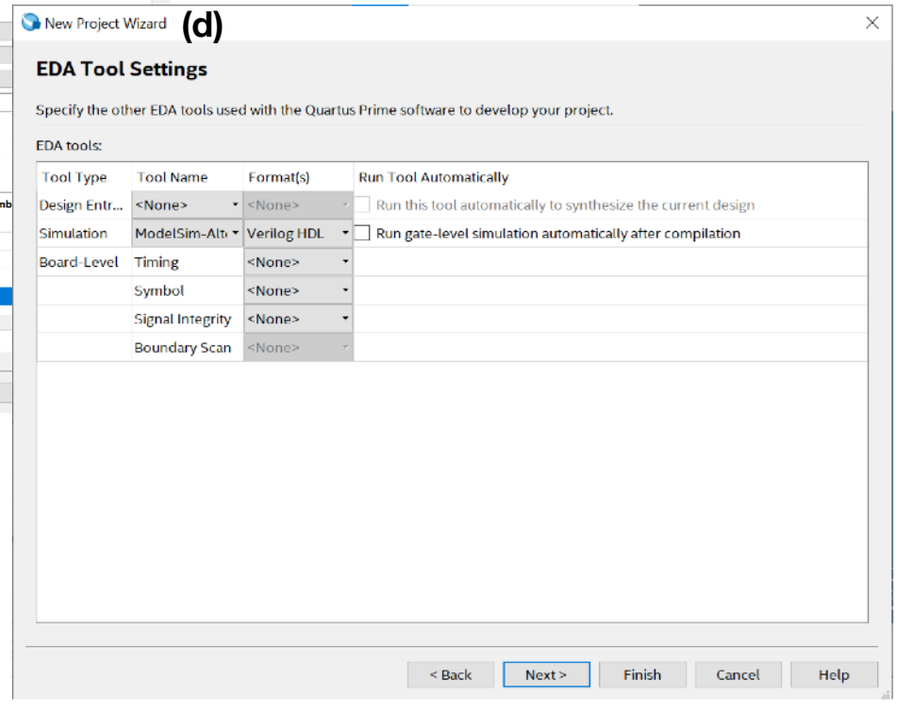
(b)



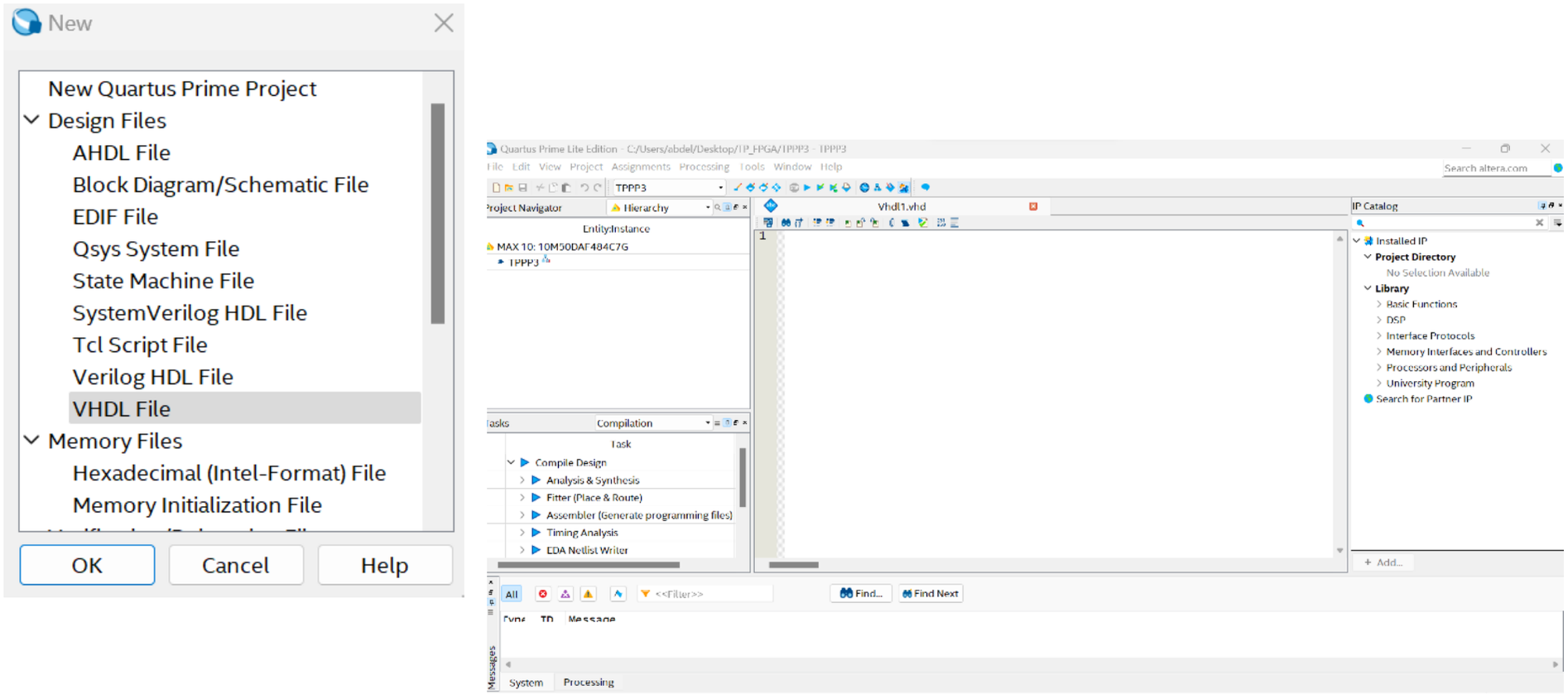
(c)



(d)



2. Now, create a description of the function F (And gate with 03 inputs):
File -> New ->VHDL File (TP5 name of the VHDL file)



3. Complete the VHDL file as the given code:

```
1  library IEEE;
2  use ieee.std_logic_1164.all;
3
4  Entity TP5 is
5
6  Port ( A,B,C    : in STD_LOGIC;
7         F: out STD_LOGIC);
8  end TP5;
9
10 architecture Behavioral of TP5 is
11
12 Begin
13
14 F <= A and B and C;
15
16  end Behavioral;
17
```

4. Compile your code to check that there aren't errors

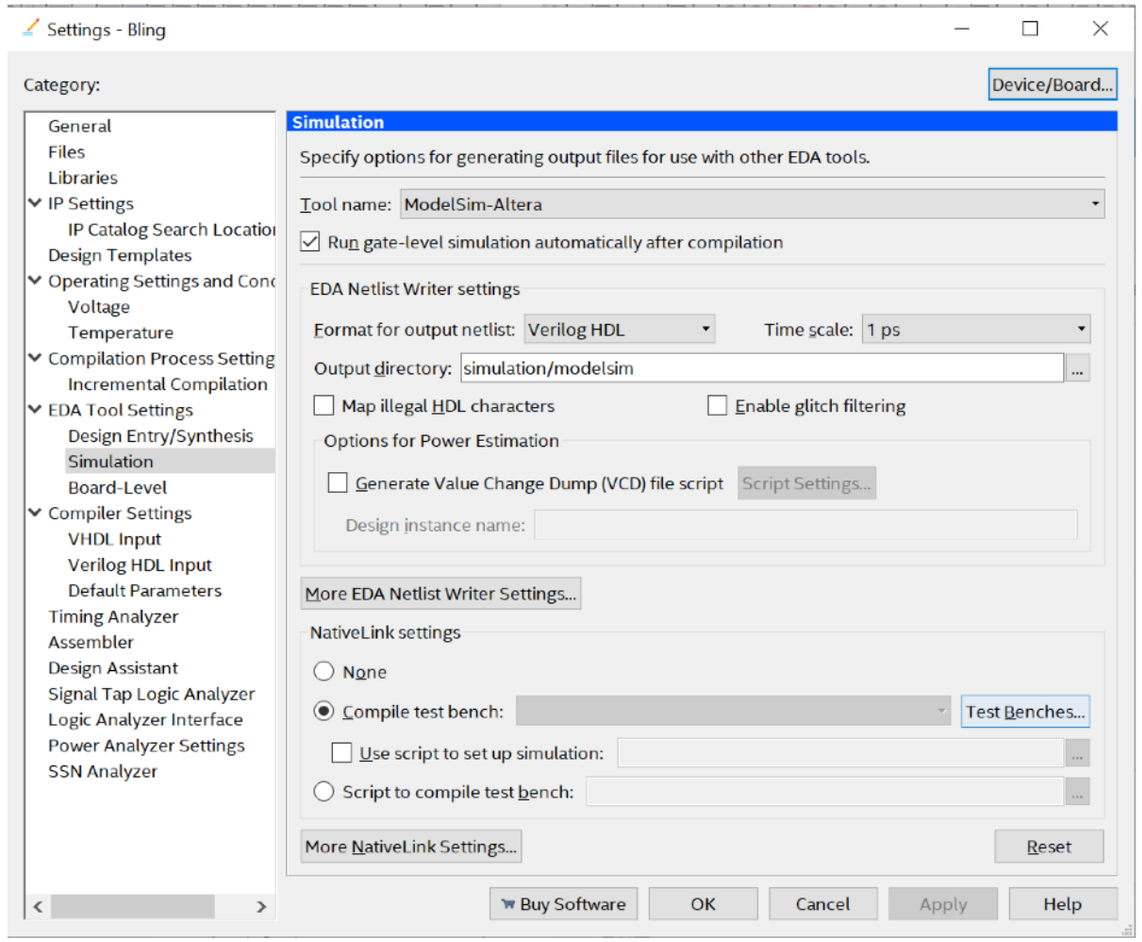
5. To simulate you code, create a testbench file. It's another VHDL file:
   File -> New -> VHDL file -> complete as a code given below.
   save as TPAnd3TP, don't forget to check the box "Add the file to the current project"

```
 1  library IEEE;
 2  use ieee.std_logic_1164.all;
 3
 4  entity TBAnd3TP is
 5  end TBAnd3TP;
 6
 7  architecture DESCRIPTION of TBAnd3TP is
 8
 9  Component TP5
10  Port ( A, B, C    : in STD_LOGIC;
11         F : OUT std_logic);
12  end component;
13
14  -- Inputs
15  signal A : std_logic := '0';
16  signal B : std_logic := '0';
17  signal C : std_logic := '0';
18  -- Outputs
19  signal F : std_logic;
20
21  Begin
22
23  Uut : TP5 port map
24  (  A => A,
25     B => B,
26     C => C,
27     F => F);
28
29   stim_proc: process
30   begin
31
32   wait for 100 ns;
33
34   A<='1'; B<='0'; C<='1'; wait for 20 ns;
35   A<='0'; B<='1'; C<='0'; wait for 30 ns;
36   A<='1'; B<='1'; C<='1'; wait for 40 ns;
37
38   wait;
39
40  end process;
41
42  end;
43
```

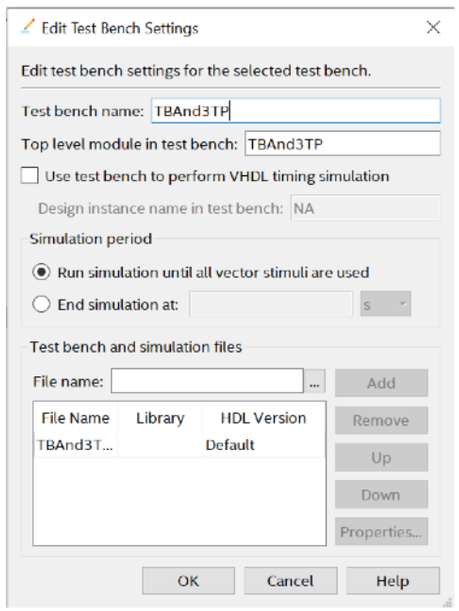
6. Compile this code to check that there aren't errors

7. To run simulation, follow these steps:

- Assignment -> Settings ..
- In the page Setting-Bling -> Simulation
- In Tool name -> ModelSim-Altera
- In Format for netlist -> Verilog HDL

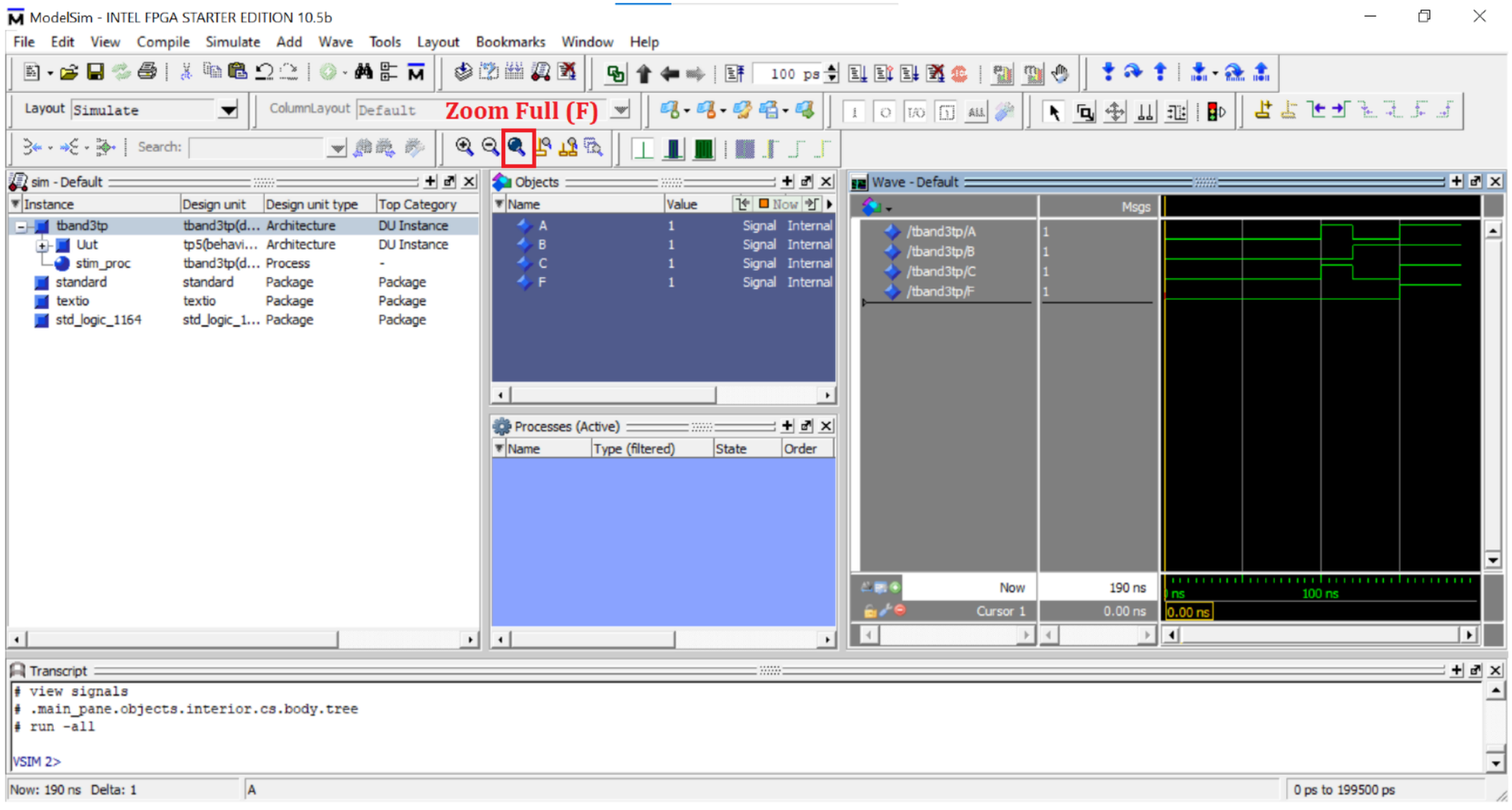


- Click on Test Benches and a window of New Test Bench Setting appear. Write the name of the test bench file
- In Test Benches and simulation files click on TPAnd3TP.vhd -> ADD -> ok -> ok -> Apply ->ok



- Select Tools ->Tools → Run Simulation (to launch ModelSim)

- A ModelSim window appears. Click on “Zoom full“ to visualize better the simulation



- Check the output F according the inputs A,B and C

- In your testbench file add this line in Stimulus process: A=0, B=1, C=1 for 50ns

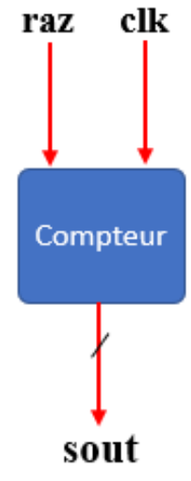
Exercise 2: Simulation of the binary counter from 0 to 9 with clock period = 20ns

1- Create a New VHDL code and complete the code given below

```
entity compteurRAZ is
Port ( clk:     in STD_LOGIC ;
       raz :    in STD_LOGIC ;
       sout :   out STD_LOGIC_VECTOR (3 downto 0));
end compteurRAZ;

architecture Behavioral of compteurRAZ is

SIGNAL s_sout : STD_LOGIC_VECTOR (3 downto 0) :="0000";
```



- **Complete this VHDL code**

2- To simulate the counter, create a testbench file in which, we should:

- Set the clk-period to 20ns
- Create a process to generate a clk signal
- In stimulus process, add these instructions:
  raz=1 and wait for 100ns
  raz=0 and wait for 50ns
  raz=1 and wait for 20ns
  raz=0 and wait forever

Exercise 3: Simulation and implementation of the counter from 0 to 9 with clock period =1s and display the result on 7-segment displayer

Exercise 4: Simulation and implementation of the counter from 0000 to 9999 with clock period 1s and display the result on 7 segment displayer

```
1   library IEEE;
2   use ieee.std_logic_1164.all;
3   use ieee.std_logic_unsigned.all;
4   use ieee.numeric_std.all;
5
6   Entity CompteurRAZSig is
7
8   Port ( clk, raz    : in STD_LOGIC;
9          SEG1: out STD_LOGIC_VECTOR(6 downto 0)
10         SEG2: out STD_LOGIC_VECTOR(6 downto 0)
11         SEG3: out STD_LOGIC_VECTOR(6 downto 0)
12         SEG4: out STD_LOGIC_VECTOR(6 downto 0));
13  end CompteurRAZSig;
14
15
16  architecture Behavioral of CompteurRAZSig is
17
18  Signal INPUT1 : STD_LOGIC_VECTOR(3 downto 0):="0000";
19  Signal INPUT2 : STD_LOGIC_VECTOR(3 downto 0):="0000";
20  Signal INPUT3 : STD_LOGIC_VECTOR(3 downto 0):="0000";
21  Signal INPUT4 : STD_LOGIC_VECTOR(3 downto 0):="0000";
22  Signal cnt : integer :=0;
23
```