## ECE 4250/ 7250: VHDL and Programmable Logic Devices Laboratory

**Lab # 2**

**Lab Title: Sequential Process Design: BCD Counter**

**Group# 8**

**Group Names: Phanindra Sai, Karanam**

**Zhengyu Li**

**Teaching Assistant Use Only:**

**Points Earned Reasons for Deduction**

**Pre-lab:**

**Post Lab report**:

**Demonstration:**

**Final Lab Grade:**

**Comments to students:**

# Objective

To familiarize with the sequential process design by modeling a counter VHDL Process and verify the functionality of the model by using ModelSim.

# Lab Work

### **VHDL code for 0-9 BCD counter**

library IEEE; --including library

use IEEE.std\_logic\_1164.all;

entity bcdc is

port(D: in std\_logic\_vector(3 downto 0);

en,l,up,clr,clk: in std\_logic;

Q: out std\_logic\_vector(3 downto 0);

Co:out std\_logic);

end bcdc;

architecture behav of bcdc is

signal Qi:signed (3 downto 0);

signal W: std\_logic\_vector(3 downto 0);

begin

Co <= '1' when ((Qi = 0000 and up = '0' and clr = '1' ) or (Qi = 1001 and up = '1' and clr ='1'))

else '0';

process(clk,clr)

begin

w <= std\_logic\_vector(Qi);

if clr = '0' then Qi<= (others => '0');

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

if(en = '1'and l ='1') then Qi <= signed(D);

elsif(w = "1001" and en ='1' and up ='1') then Qi <= "0000";

elsif(w = "0000" and en= '1' and up ='0') then Qi <="1001";

elsif(l='0' and en ='1' and up ='1') then Qi <= Qi + 1;

elsif(l='0' and en= '1' and up ='0') then Qi <= Qi - 1;

end if;

end if;

end process;

Q <= std\_logic\_vector(Qi);

end behav;

### **VHDL code for 00-99 BCD counter**

library IEEE; --including library

use IEEE.std\_logic\_1164.all;

use IEEE.std\_logic\_arith.all;

--use IEEE.numeric\_bit.all;

entity bcdc2 is

port(D1: in std\_logic\_vector(3 downto 0);

D2: in std\_logic\_vector(3 downto 0);

en,l,up,clr,clk: in std\_logic;

Q1: out std\_logic\_vector(3 downto 0);

Q2: out std\_logic\_vector(3 downto 0);

Co:out std\_logic );

end bcdc2;

architecture behav of bcdc2 is

component bcdc is

port(D: in std\_logic\_vector(3 downto 0);

en,l,up,clr,clk: in std\_logic;

Q: out std\_logic\_vector(3 downto 0);

Co:out std\_logic);

end component;

signal Ci,t:std\_logic;

begin

t<= Ci or l;

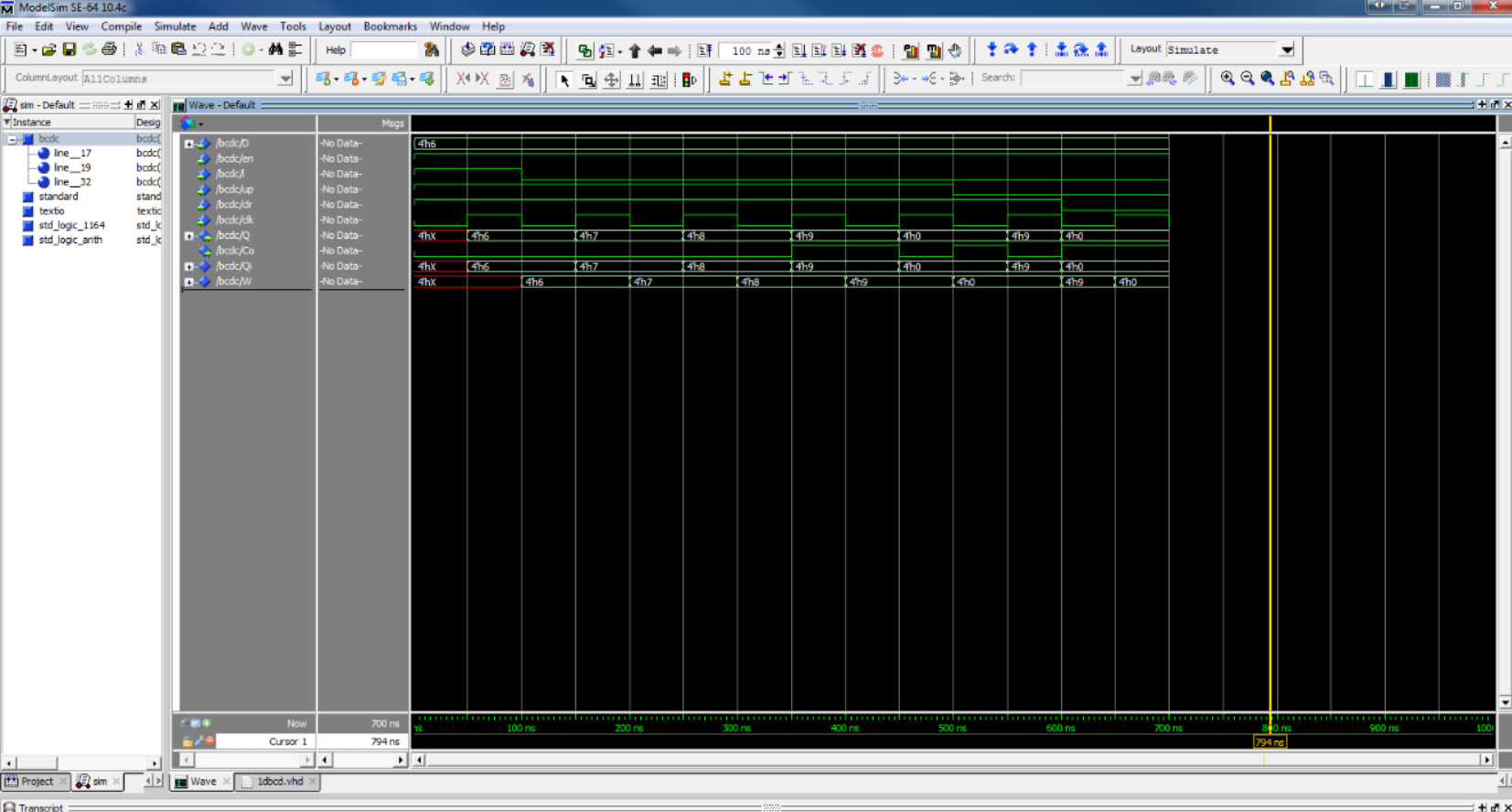
c1: bcdc port map (D1,en,l,up,clr,clk,Q1,Ci);

c2: bcdc port map (D2,t,l,up,clr,clk,Q2,Co);

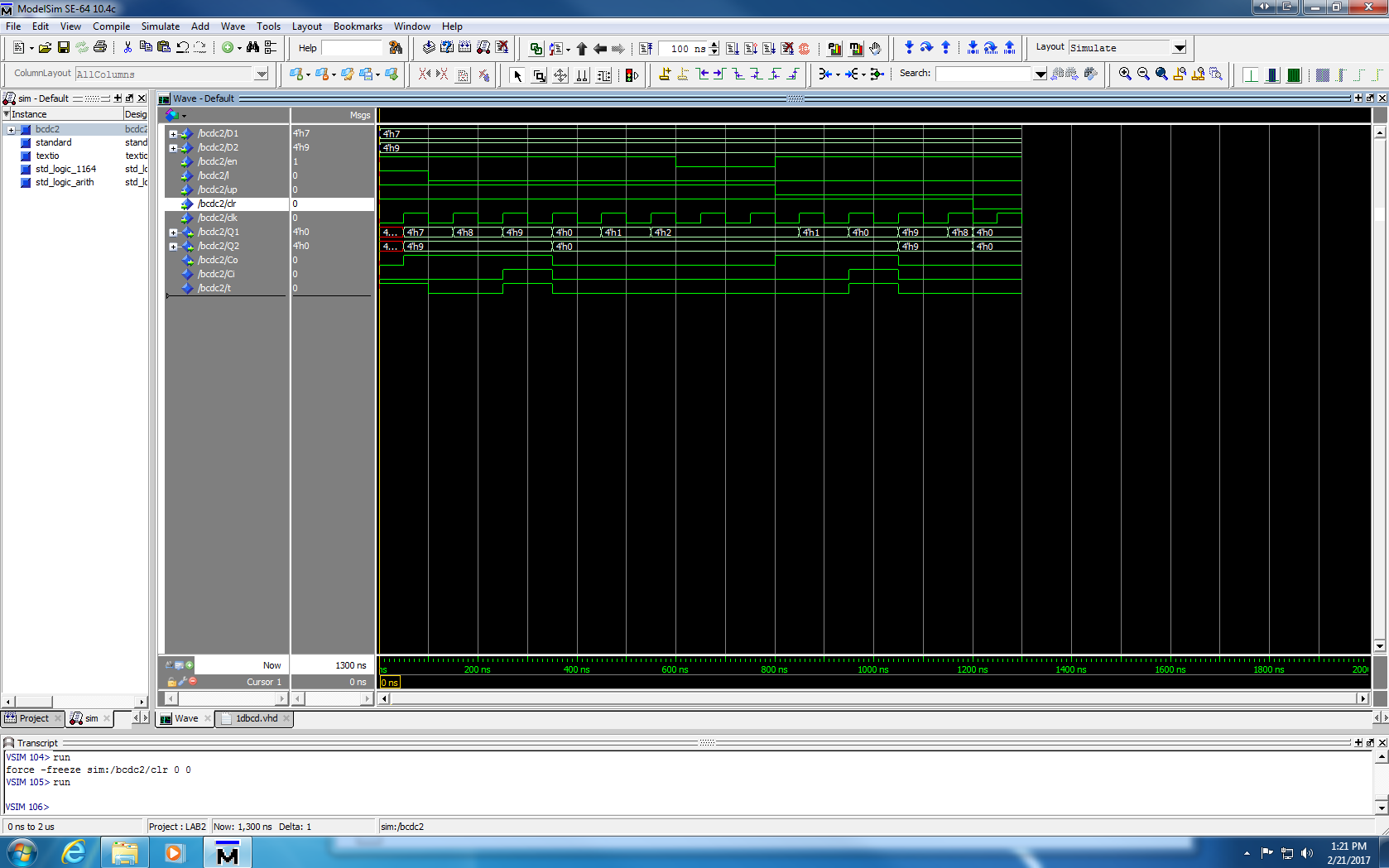
end behav;

**Screenshots**

**4 bit Full adder**

****

**6 bit Subtractor**

****

**Answers**

1. **In part a, step 2, what is the value of Co? Justify your answer?**

The value of Co should be 0 in the first three increments, the last increment should be 1. Below is Co segment of the 0-9 BCD counter.

Co <= '1' when ((Qi = 0000 and up = '0' and clr = '1' ) or (Qi = 1001 and up = '1' and clr ='1'))

else '0';

From 6 to 9, the code will be executed else because Qi isn’t equal 0000 or 1001; when do the last increment, Qi=1001 and up=’1’ and clr=’1’ which is satisfy “when” statement, therefore, the value of Co should be 1 at this time.

1. **If the** **counter current state is 0, and you have UP=0 (decrement)? How do you handle this case, explain?**

When counter current state is 0 and do decrement, we need a statement to judge the Co like this: Co <= '1' when ((Qi = 0000 and up = '0' and clr = '1' ); Co is 1 in this case and the next state will be 1001 in this problem.

**3.In part b, when the counter current state is 00, and you have UP=0(decrement)? What is the expected output?**

Counter current state is 00 and UP=0 after 9 clock periods since loaded data. Q1 and Q2 is 0; Co is 1.

# Conclusion

# 

We had learnt how to use Model Sim to design a counter VHDL Process and verify the functionality of the model.