The purpose of this exercise is to become familiar with the sequential architecture of VHDL by implementing and simulating a simple scoreboard, which can display scores from 0 to 99 (decimal). The decimal count gets incremented by 1 if increment signal is high and is decremented by 1 if decrement signal is high. If increment and decrement are high/low simultaneously, no action occurs.

Write your code in VHDL by taking the following into consideration:

1. The library files that must be used before the start of the VHDL code is
   
   ```vhdl
   LIBRARY ieee;
   USE ieee.std_logic_1164.ALL;
   USE ieee.numeric_std.ALL;
   ```

2. The 4-bit counter has the following input and output signals:
   a) `clk, inc, dec, reset : in std_logic;`
   b) `seg7disp1, seg7disp2 : out unsigned(6 downto 0);`

3. The counter should behave as follows
   a) If the reset is high, the counter must be set to 0.
   b) If the reset is low, inc is 1 and dec is 0, the counter will increment by 1 (i.e. each counter will count from 0-9 and repeat the cycle as long as the inputs stay the same).
   c) If the reset is low, inc is 0 and dec is 1, the counter will decrement by 1 (i.e. each counter will count backwards from 9-0 and repeat the cycle as long as the inputs stay the same).
   d) If the reset is low, inc is 1 and dec is 1, the counter should stop working.
   e) If the reset is low, inc is 0 and dec is 0, the counter should stop working.

4. **No report is needed. Include comments for the VHDL Code.**
   a) The VHDL Code.
   b) Four waveforms (show the output of two BCD counter’s also) (clock period = 10ns, endtime=320ns )
      1. inc = 1, dec = 0 and reset = 0
      2. inc = 0, dec = 1 and reset = 0
      3. inc = 1 (0-150ns), dec = 1 (100-200ns) and reset = 0
      4. reset = 1 in the middle

5. Hints:
   I. BCD0 and BCD1 are temporary unsigned 4-bit signals initialized to “0000”.
   II. Define an array (`sevsegarray`) of ten 4-bit signed numbers.
   III. Define a constant (`seg7rom`) of type `sevsegarray`. Initialize the constant to the values corresponding to each BCD number for seven segment display.
   IV. To access the array use `to_integer(...)"