Practical 4 LCD interface

Code

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;
---- Uncomment the following library declaration if instantiating
---- any Xilinx primitives in this code.
--library UNISIM;
--use UNISIM.VComponents.all;
entity lcd12 is
   Port ( clk : in  STD_LOGIC;
          lcd_rw : out  STD_LOGIC;
          lcd_e : out STD_LOGIC;
          lcd_rs : out  STD_LOGIC;
          data : out  STD_LOGIC_VECTOR (7 downto 0));
end lcd12;
architecture Behavioral of lcd12 is
constant N:integer:=17;
type arr is array (1 to N) of std_logic_vector(7 downto 0);
constant
datas:arr:=(X"38",X"0c",X"06",X"01",X"C0",X"57",x"45",x"4c",x"43",x"4f",x"4d"
,x"45",x"20",x"45",x"4e",
 x"54",x"43"); --command and data to display
begin
lcd_rw <= '0';  --lcd write
process(clk)
```

```
variable i : integer := 0;
variable j : integer := 1;
begin
if clk'event and clk = '1' then
if i <= 1000000 then
i := i + 1;
 lcd_e <= '1';
data <= datas(j)(7 downto 0);
elsif i > 1000000 and i < 2000000 then
 i := i + 1;
 lcd_e <= '0';
 elsif i = 2000000 then
 j := j + 1;
 i := 0;
 end if;
 if j <= 5  then
 lcd_rs <= '0';    --command signal
 elsif j > 5   then
 lcd_rs <= '1';   --data signal
 end if;
 if j = 17 then  --repeated display of data
 j := 5;
 end if;
end if;
end process;
end Behavioral;
```

TestBench Practical 4 LCD

```
LIBRARY ieee;
USE ieee.std_logic_1164.ALL;
USE ieee.std_logic_unsigned.all;
USE ieee.numeric_std.ALL;
ENTITY lcd123_vhd IS
END lcd123_vhd;
ARCHITECTURE behavior OF lcd123_vhd IS
    -- Component Declaration for the Unit Under Test (UUT)
    COMPONENT lcd12
    PORT(
        clk : IN std_logic;
        lcd_rw : OUT std_logic;
        lcd_e : OUT std_logic;
        lcd_rs : OUT std_logic;
        data : OUT std_logic_vector(7 downto 0)
        );
    END COMPONENT;
    --Inputs
    SIGNAL clk :  std_logic := '0';
    --Outputs
    SIGNAL lcd_rw :  std_logic;
    SIGNAL lcd_e :  std_logic;
    SIGNAL lcd_rs :  std_logic;
    SIGNAL data :  std_logic_vector(7 downto 0);
BEGIN
    -- Instantiate the Unit Under Test (UUT)
    uut: lcd12 PORT MAP(
        clk => clk,
        lcd_rw => lcd_rw,
			lcd_e => lcd_e,
			lcd_rs => lcd_rs,
			data => data
		);
		tb : PROCESS
		BEGIN
			-- Wait 100 ns for global reset to finish
			wait for 100 ns;
			clk <= '0';
  wait for 10 ns;
  clk <= '1';
  wait for 10 ns;
    -- wait for clk_period*10;
			wait; -- will wait forever
		END PROCESS;
	END;
```