การทดลองที่ 10 วงจรนับเลขฐานสิบขนาด 4 หลัก

แสดงผลลอก LED

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity dec2seg2 is
  port (bin : in INTEGER range 15 downto 0;
          segm : out std_logic_vector(7 downto 0));
end dec2seg2;

architecture Behavioral of dec2seg2 is
begin
      with bin select
            segm <= "00000011" when 0,
                    "10011111" when 1,
                    "00100101" when 2,
                    "00001101" when 3,
                    "10011001" when 4,
                    "01001001" when 5,
                    "01000001" when 6,
                    "00011111" when 7,
                    "00000001" when 8,
                    "00001001" when others;
end Behavioral;
```

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity codisp is
    Port (clk1, clk2, ce, reset : in std_logic;
              seg : out std_logic_vector(7 downto 0);
              dig : out std_logic_vector(3 downto 0));
end codisp;

architecture Behavioral of codisp is

component dec2seg2
  port (bin : in INTEGER range 15 downto 0;
          segm : out std_logic_vector(7 downto 0));
end component;
component binco
    port (clk : in std_logic;
            y : out std_logic_vector(1 downto 0));
end component;
signal s :  std_logic;
signal ya, yb,yc, yd, bus4 : INTEGER range 15 downto 0;
signal bus5 : std_logic_vector(1 downto 0);
begin

process (CLK1, CE, Reset)
begin
   if Reset='1' then
      ya <= 0;
      yb <= 0;
      yc <= 0;
```

```
      yd <= 0;
   else
      if CLK1='1' and CLK1'event then
         if CE = '1' then

            if ya >= 9 then
               ya <= 0;
               if yb >= 9 then
                  yb <= 0;
                  if yc >= 9 then
                     yc <= 0;
                     if yd >= 9 then
                        yd <= 0;
                     else
                        yd <= yd + 1;
                     end if;
                  else
                    yc <= yc + 1;
                  end if;
               else
                  yb <= yb + 1;
               end if;
            else
               ya <= ya + 1;
            end if;


         else
            ya <= ya;
            yb <= yb;
            yc <= yc;
            yd <= yd;
         end if;
      end if;
   end if;
end process;

process (bus5, ya, yb, yc, yd)
begin
    case bus5 is
                when "00"  =>
                   bus4 <= ya;
                   dig <= "1110";
                when "01"  =>
                   bus4 <= yb;
                   dig <= "1101";
                when "10"  =>
                   bus4 <= yc;
                   dig <= "1011";
                when others  =>
                   bus4 <= yd;
                   dig <= "0111";
    end case;
end process;
    c1: dec2seg2 port map(bus4, seg);
    c2: binco port map(clk2, bus5);
end Behavioral;
```

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_unsigned.ALL;
entity binco is
    port (clk : in std_logic;
            y : out std_logic_vector(1 downto 0));
end binco;
architecture binco_beh of binco is
type state_type is (a, b, c, d);
signal state: state_type;
begin
  process (clk)
  begin
    if clk = '1' and clk'event then
       case state is
               when a =>  state <= b;
               when b =>  state <= c;
               when c =>  state <= d;
               when d =>  state <= a;
       end case;
    end if;
end process;
process (state)
   begin
      case state is
      when a => y <= "00";
      when b => y <= "01";
      when c => y <= "10";
      when d => y <= "11";
      end case;
   end process;
end binco_beh;
```