Compteur modulo

Solution

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity compteur_modulo is
   generic (TAILLE : integer := 8;
       MODULO : integer := 10 );
   port(
          CLK     : in  std_logic;
          RESET   : in  std_logic;
          ENABLE  : in  std_logic;
          DOUT : out std_logic_vector(TAILLE-1 downto 0);
          TC      : out std_logic);
end;
architecture behavior of compteur_modulo is begin
   clk_proc:process(CLK)
      variable COUNT:unsigned(TAILLE-1 downto 0) := (others => '0');
   begin
      if rising_edge(CLK) then
```

Compteur modulo

Solution

```
        if RESET = '1' then
           COUNT := (others=>'0');
        else
           TC    <= '1';
           if ENABLE = '0' then
              COUNT := COUNT + 1;
              if COUNT = MODULO then
                 COUNT := (others => '0');
                 TC    <= '0';
              end if;
           end if;
        end if;
     end if;
     DOUT <= std_logic_vector(COUNT);
  end process clk_proc;
end behavior;
```

Compteur modulo

Solution

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;


entity compteurs is
   generic (TAILLE : integer := 8;
       MODULO : integer := 10 );
   port(
          CLK     : in  std_logic;
          RESET   : in  std_logic;
          ENABLE  : in  std_logic;
          DOUT1   : buffer std_logic_vector(TAILLE-1 downto 0);
          DOUT2   : buffer std_logic_vector(TAILLE-1 downto 0);
          SEG1 : out std_logic_vector(6 downto 0);
          SEG2 : out std_logic_vector(6 downto 0);
          TC      : out std_logic);
end;
```

Compteur modulo

Solution

```
architecture arch_compteurs of compteurs is
   component compteur_modulo
   generic (TAILLE : integer := 8;
       MODULO : integer := 10 );
   port(
          CLK     : in  std_logic;
          RESET   : in  std_logic;
          ENABLE  : in  std_logic;
          DOUT : out std_logic_vector(TAILLE-1 downto 0);
          TC      : out std_logic);
   end component;
   component dec_7seg is
   port(hex_in : in  std_logic_vector(3 downto 0);
        segs       : out   std_logic_vector(6 downto 0));
   end component;
   signal TC1, TC2   :  std_logic;
begin
   unit_1: compteur_modulo generic map (4,10) port
    map(CLK,RESET,ENABLE,DOUT1,TC1);
```

Compteur modulo

Solution

```
   unit_2: compteur_modulo generic map (4,10) port
     map(CLK,RESET,TC1,DOUT2,TC2);
   seg_1: dec_7seg port map(DOUT1,SEG1);
   seg_2: dec_7seg port map(DOUT2,SEG2);
   TC <= TC2;
end;
```

```
library ieee;
use ieee.std_logic_1164.all;
entity top_compteurs is
   port (KEY   : in  std_logic_vector(3 downto 0);
         SW    : in  std_logic_vector(9 downto 0);
         LEDR  : out std_logic_vector(9 downto 0);
         HEX0  : out std_logic_vector(0 to 6);
         HEX1  : out std_logic_vector(0 to 6);
         CLOCK_50 : in std_logic);
end;

architecture arch_top_compteurs of top_compteurs is
```

Compteur modulo

Solution

```
    component compteurs
    generic (TAILLE : integer := 8;
        MODULO : integer := 10 );
    port(
           CLK      : in  std_logic;
           RESET    : in  std_logic;
           ENABLE   : in  std_logic;
           DOUT1    : out std_logic_vector(TAILLE-1 downto 0);
           DOUT2    : out std_logic_vector(TAILLE-1 downto 0);
           SEG1 : out std_logic_vector(6 downto 0);
           SEG2 : out std_logic_vector(6 downto 0);
           TC       : out std_logic);
    end component;
begin
    compteurs_1: compteurs generic map (4,10) port
      map(CLOCK_50,SW(0),KEY(0),
                    LEDR(3 downto 0), LEDR(7 downto 4),
                    HEX0,HEX1,LEDR(8));
    LEDR(9) <= '0';
```

Compteur modulo

Solution

```
end;
```

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;


entity compteurs6 is
   generic (TAILLE : integer := 8;
       MODULO : integer := 10 );
   port(
          CLK     : in  std_logic;
          RESET   : in  std_logic;
          ENABLE  : in  std_logic;
          SEG1 : out std_logic_vector(6 downto 0);
          SEG2 : out std_logic_vector(6 downto 0);
          SEG3 : out std_logic_vector(6 downto 0);
          SEG4 : out std_logic_vector(6 downto 0);
          SEG5 : out std_logic_vector(6 downto 0);
```

Compteur modulo

Solution

```
          SEG6 : out std_logic_vector(6 downto 0);
          TC   : out std_logic);
end;

architecture arch_compteurs6 of compteurs6 is
   component compteur_modulo
   generic (TAILLE : integer := 8;
       MODULO : integer := 10 );
   port(
          CLK    : in  std_logic;
          RESET  : in  std_logic;
          ENABLE : in  std_logic;
          DOUT : out std_logic_vector(TAILLE-1 downto 0);
          TC   : out std_logic);
   end component;
   component dec_7seg is
   port(hex_in : in  std_logic_vector(3 downto 0);
        segs   : out   std_logic_vector(6 downto 0));
   end component;
```

Compteur modulo

Solution

```
   signal TC1, TC2, TC3, TC4, TC5, TC6 :  std_logic;
   signal DOUT1,DOUT2,DOUT3,DOUT4,DOUT5,DOUT6 :
     std_logic_vector(TAILLE-1 downto 0);
begin
   unit_1: compteur_modulo generic map (4,10) port
     map(CLK,RESET,ENABLE,DOUT1,TC1);
   unit_2: compteur_modulo generic map (4,10) port
     map(CLK,RESET,TC1,DOUT2,TC2);
   unit_3: compteur_modulo generic map (4,10) port
     map(CLK,RESET,TC2,DOUT3,TC3);
   unit_4: compteur_modulo generic map (4,10) port
     map(CLK,RESET,TC3,DOUT4,TC4);
   unit_5: compteur_modulo generic map (4,10) port
     map(CLK,RESET,TC4,DOUT5,TC5);
   unit_6: compteur_modulo generic map (4,10) port
     map(CLK,RESET,TC5,DOUT6,TC6);
   seg_1: dec_7seg port map(DOUT1,SEG1);
   seg_2: dec_7seg port map(DOUT2,SEG2);
   seg_3: dec_7seg port map(DOUT3,SEG3);
```

Compteur modulo

Solution

```
   seg_4: dec_7seg port map(DOUT4,SEG4);
   seg_5: dec_7seg port map(DOUT5,SEG5);
   seg_6: dec_7seg port map(DOUT6,SEG6);
   TC <= TC6;
end;
```

```
library ieee;
use ieee.std_logic_1164.all;
entity top_compteurs6 is
   port (KEY  : in  std_logic_vector(3 downto 0);
         SW   : in  std_logic_vector(9 downto 0);
         HEX0 : out std_logic_vector(0 to 6);
         HEX1 : out std_logic_vector(0 to 6);
         HEX2 : out std_logic_vector(0 to 6);
         HEX3 : out std_logic_vector(0 to 6);
         HEX4 : out std_logic_vector(0 to 6);
         HEX5 : out std_logic_vector(0 to 6);
         CLOCK_50 : in std_logic);
end;
```

Compteur modulo

Solution

```
architecture arch_top_compteurs6 of top_compteurs6 is
   component compteurs6
   generic (TAILLE : integer := 8;
       MODULO : integer := 10 );
   port(
          CLK     : in  std_logic;
          RESET   : in  std_logic;
          ENABLE  : in  std_logic;
          SEG1 : out std_logic_vector(6 downto 0);
          SEG2 : out std_logic_vector(6 downto 0);
          SEG3 : out std_logic_vector(6 downto 0);
          SEG4 : out std_logic_vector(6 downto 0);
          SEG5 : out std_logic_vector(6 downto 0);
          SEG6 : out std_logic_vector(6 downto 0);
          TC      : out std_logic);
   end component;
begin
```

Compteur modulo

Solution

```
   compteurs_1: compteurs6 generic map (4,10) port
     map(CLOCK_50,SW(0),KEY(0),
                   HEX0,HEX1,HEX2,HEX3,HEX4,HEX5);
end;
```

Compteur modulo « visible »

Solution

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;

entity compteur_modulo_mod is
   generic (TAILLE : integer := 8;
       MODULO : integer := 10 );
   port(
          CLK     : in  std_logic;
          RESET   : in  std_logic;
          ENABLE  : in  std_logic;
          DOUT : out std_logic_vector(TAILLE-1 downto 0);
          TC      : out std_logic);
end;
architecture behavior of compteur_modulo_mod is
signal enable_del, enable_edge : std_logic;
begin

   clk_proc:process(CLK)
```

Compteur modulo « visible »

Solution

```
    variable COUNT:unsigned(TAILLE-1 downto 0) := (others => '0');
begin
    if rising_edge(CLK) then
        if RESET = '1' then
            COUNT := (others=>'0');
            enable_del <= '0';
        else
            enable_del <= ENABLE;
            TC    <= '1';
            if enable_edge = '0' then
                COUNT := COUNT + 1;
                if COUNT = MODULO then
                    COUNT := (others => '0');
                    TC    <= '0';
                end if;
            end if;
        end if;
    end if;
    DOUT <= std_logic_vector(COUNT);
```

Compteur modulo « visible »

Solution

```
   end process clk_proc;
   enable_edge <= not(not ENABLE and enable_del);
end behavior;
```

```
library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.numeric_std.all;


entity compteurs6_mod is
   generic (TAILLE : integer := 8;
       MODULO : integer := 10 );
   port(
          CLK     : in  std_logic;
          RESET   : in  std_logic;
          ENABLE  : in  std_logic;
          SEG1 : out std_logic_vector(6 downto 0);
          SEG2 : out std_logic_vector(6 downto 0);
          SEG3 : out std_logic_vector(6 downto 0);
```

Compteur modulo « visible »

Solution

```
          SEG4 : out std_logic_vector(6 downto 0);
          SEG5 : out std_logic_vector(6 downto 0);
          SEG6 : out std_logic_vector(6 downto 0);
          TC       : out std_logic);
end;

architecture arch_compteurs6_mod of compteurs6_mod is
   component compteur_modulo_mod
   generic (TAILLE : integer := 8;
       MODULO : integer := 10 );
   port(
          CLK      : in  std_logic;
          RESET    : in  std_logic;
          ENABLE   : in  std_logic;
          DOUT : out std_logic_vector(TAILLE-1 downto 0);
          TC       : out std_logic);
   end component;
   component dec_7seg is
   port(hex_in : in  std_logic_vector(3 downto 0);
```

Compteur modulo « visible »

Solution

```
        segs        : out   std_logic_vector(6 downto 0));
    end component;
    signal TC1, TC2, TC3, TC4, TC5, TC6 :  std_logic;
    signal DOUT1,DOUT2,DOUT3,DOUT4,DOUT5,DOUT6 :
      std_logic_vector(TAILLE-1 downto 0);
begin
    unit_1: compteur_modulo_mod generic map (4,10) port
      map(CLK,RESET,ENABLE,DOUT1,TC1);
    unit_2: compteur_modulo_mod generic map (4,10) port
      map(CLK,RESET,TC1,DOUT2,TC2);
    unit_3: compteur_modulo_mod generic map (4,10) port
      map(CLK,RESET,TC2,DOUT3,TC3);
    unit_4: compteur_modulo_mod generic map (4,10) port
      map(CLK,RESET,TC3,DOUT4,TC4);
    unit_5: compteur_modulo_mod generic map (4,10) port
      map(CLK,RESET,TC4,DOUT5,TC5);
    unit_6: compteur_modulo_mod generic map (4,10) port
      map(CLK,RESET,TC5,DOUT6,TC6);
    seg_1: dec_7seg port map(DOUT1,SEG1);
```

Compteur modulo « visible »

Solution

```
   seg_2: dec_7seg port map(DOUT2,SEG2);
   seg_3: dec_7seg port map(DOUT3,SEG3);
   seg_4: dec_7seg port map(DOUT4,SEG4);
   seg_5: dec_7seg port map(DOUT5,SEG5);
   seg_6: dec_7seg port map(DOUT6,SEG6);
   TC <= TC6;
end;
```

```
library ieee;
use ieee.std_logic_1164.all;
entity top_compteurs6_mod is
   port (KEY  : in  std_logic_vector(3 downto 0);
         SW   : in  std_logic_vector(9 downto 0);
         HEX0 : out std_logic_vector(0 to 6);
         HEX1 : out std_logic_vector(0 to 6);
         HEX2 : out std_logic_vector(0 to 6);
         HEX3 : out std_logic_vector(0 to 6);
         HEX4 : out std_logic_vector(0 to 6);
         HEX5 : out std_logic_vector(0 to 6);
```

Compteur modulo « visible »

Solution

```
          CLOCK_50 : in std_logic);
end;

architecture arch_top_compteurs6_mod of top_compteurs6_mod is
   component compteurs6_mod
   generic (TAILLE : integer := 8;
       MODULO : integer := 10 );
   port(
          CLK      : in  std_logic;
          RESET    : in  std_logic;
          ENABLE   : in  std_logic;
          SEG1 : out std_logic_vector(6 downto 0);
          SEG2 : out std_logic_vector(6 downto 0);
          SEG3 : out std_logic_vector(6 downto 0);
          SEG4 : out std_logic_vector(6 downto 0);
          SEG5 : out std_logic_vector(6 downto 0);
          SEG6 : out std_logic_vector(6 downto 0);
          TC       : out std_logic);
   end component;
```

Compteur modulo « visible »

Solution

```
begin
   compteurs_1: compteurs6_mod generic map (4,10) port
     map(CLOCK_50,SW(0),KEY(0),
                   HEX0,HEX1,HEX2,HEX3,HEX4,HEX5);
end;
```