Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

Комп’ютерна схемотехніка – 1

Комп’ютерна схемотехніка

Лабораторна робота №6

# «Створення таймера в САПР Quartus ii.»

Виконала:

студентка групи ІО-64

Бровченко А. В.

Залікова книжка № 6403

Номер у списку групи 3

Перевірив [доц. Верба О. А.](http://rozklad.kpi.ua/Schedules/ViewSchedule.aspx?v=3616fe25-c15f-4d3e-986b-deb3928e21b8)

Київ

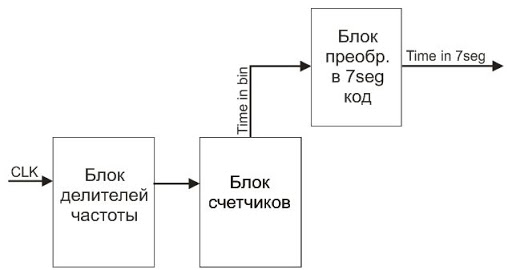
2018 р.

**Мета:**

* *Навчитись працювати з генераторами та перетворювати частоти*
* *Навчитись працювати з самисегментними індикаторами*
* *Робота зі стендом DE2 Board Altera*

**Хід роботи**

Нам знадобиться чотири семисегментних індикатора «ГГ: ХХ». Для відліку часу нам потрібен більш-менш точний сигнал 1Hz. Його ми отримаємо шляхом ділення частоти 27Mhz, потім ми будемо відраховувати його на 60 (секунди), потім ще раз на 60 (хвилини), а потім на 24 (години). З останніх двох блоків двійкове число хвилин і годин буде надходити на декодер Bin -> BCD (binary-coded decimal) -> 7seg. Схема асинхронна. Усі блоки написані на VHDL.



Блок ділення частоти:

library ieee;  
use ieee.std\_logic\_1164.all;  
use ieee.std\_logic\_unsigned.all;  
  
entity Div\_27Mhz\_to\_1Hz is  
port( clk:in std\_logic; clk\_out:out std\_logic);  
end Div\_27Mhz\_to\_1Hz;  
  
architecture div\_behavior of Div\_27Mhz\_to\_1Hz is  
begin  
process(clk)  
variable cnt : integer range 0 to 27000000;  
begin  
if(clk'event and clk = '1')   
then  
  
if(cnt >= 13500000)  
then  
clk\_out <= '1';  
else   
clk\_out <= '0';  
end if;  
  
if(cnt = 27000000)  
then  
cnt := 0;  
else  
cnt := cnt + 1;  
end if;   
  
end if;  
end process;  
end div\_behavior;

Лічильник від 0 до 59, який ми використовуємо для відліку хвилин і секунд:

library ieee;  
use ieee.std\_logic\_1164.all;  
use ieee.std\_logic\_unsigned.all;  
  
-- For CONV\_STD\_LOGIC\_VECTOR:  
use ieee.std\_logic\_arith.all;  
  
entity cnt\_0\_to\_59 is  
port( clk:in std\_logic; c59:out std\_logic; vector:out std\_logic\_vector(5 downto 0));  
end cnt\_0\_to\_59;  
  
architecture cnt\_behavior of cnt\_0\_to\_59 is  
begin  
process(clk)  
variable cnt : integer range 0 to 59;  
begin  
if(clk'event and clk = '1')   
then  
if(cnt = 59)  
then  
cnt := 0;  
c59 <= '1';  
vector <= CONV\_STD\_LOGIC\_VECTOR(cnt, 6);  
el  
cnt := cnt + 1;  
c59 <= '0';  
vector <= CONV\_STD\_LOGIC\_VECTOR(cnt, 6);  
end if;  
end if;  
end process;   
end cnt\_behavior;

Лічильник від 0 до 23 для рахування годин:

library ieee;  
use ieee.std\_logic\_1164.all;  
use ieee.std\_logic\_unsigned.all;  
  
-- For CONV\_STD\_LOGIC\_VECTOR:  
use ieee.std\_logic\_arith.all;  
  
entity cnt\_0\_to\_23 is  
port( clk:in std\_logic; vector:out std\_logic\_vector(4 downto 0));  
end cnt\_0\_to\_23;  
  
architecture cnt\_behavior of cnt\_0\_to\_23 is  
begin  
process(clk)  
variable cnt : integer range 0 to 23;  
begin  
if(clk'event and clk = '1')   
then  
if(cnt = 23)  
then  
cnt := 0;  
vector <= CONV\_STD\_LOGIC\_VECTOR(cnt, 5);  
else  
cnt := cnt + 1;  
vector <= CONV\_STD\_LOGIC\_VECTOR(cnt, 5);  
end if;  
end if;  
end process;  
end cnt\_behavior;

Перетворювач Binary в BCD для 5 біт:

library ieee;  
use ieee.std\_logic\_1164.all;  
use ieee.std\_logic\_unsigned.all;  
  
-- For CONV\_STD\_LOGIC\_VECTOR:  
use ieee.std\_logic\_arith.all;  
  
entity bin2bcd\_5bit is  
port( bin:in std\_logic\_vector(4 downto 0);   
bcd1:out std\_logic\_vector(3 downto 0);  
bcd10:out std\_logic\_vector(3 downto 0)   
);  
  
end bin2bcd\_5bit;  
  
architecture converter\_behavior of bin2bcd\_5bit is   
begin  
process(bin)  
variable i : integer range 0 to 23;  
variable i1 : integer range 0 to 9;  
begin  
i := conv\_integer(bin);  
i1 := i / 10;  
bcd10 <= CONV\_STD\_LOGIC\_VECTOR(i1, 4);   
i1 := i rem 10;  
bcd1 <= CONV\_STD\_LOGIC\_VECTOR(i1, 4);   
end process;  
end converter\_behavior;

Перетворювач Binary в BCD для 6 біт:

library ieee;

use ieee.std\_logic\_1164.all;

use ieee.std\_logic\_unsigned.all;

-- For CONV\_STD\_LOGIC\_VECTOR:

use ieee.std\_logic\_arith.all;

entity bin2bcd\_6bit is

port( bin:in std\_logic\_vector(5 downto 0);

bcd1:out std\_logic\_vector(3 downto 0);

bcd10:out std\_logic\_vector(3 downto 0)

);

end bin2bcd\_6bit;

architecture converter\_behavior of bin2bcd\_6bit is

begin

process(bin)

variable i : integer range 0 to 59;

variable i1 : integer range 0 to 9;

begin

i := conv\_integer(bin);

i1 := i / 10;

bcd10 <= CONV\_STD\_LOGIC\_VECTOR(i1, 4);

i1 := i rem 10;

bcd1 <= CONV\_STD\_LOGIC\_VECTOR(i1, 4);

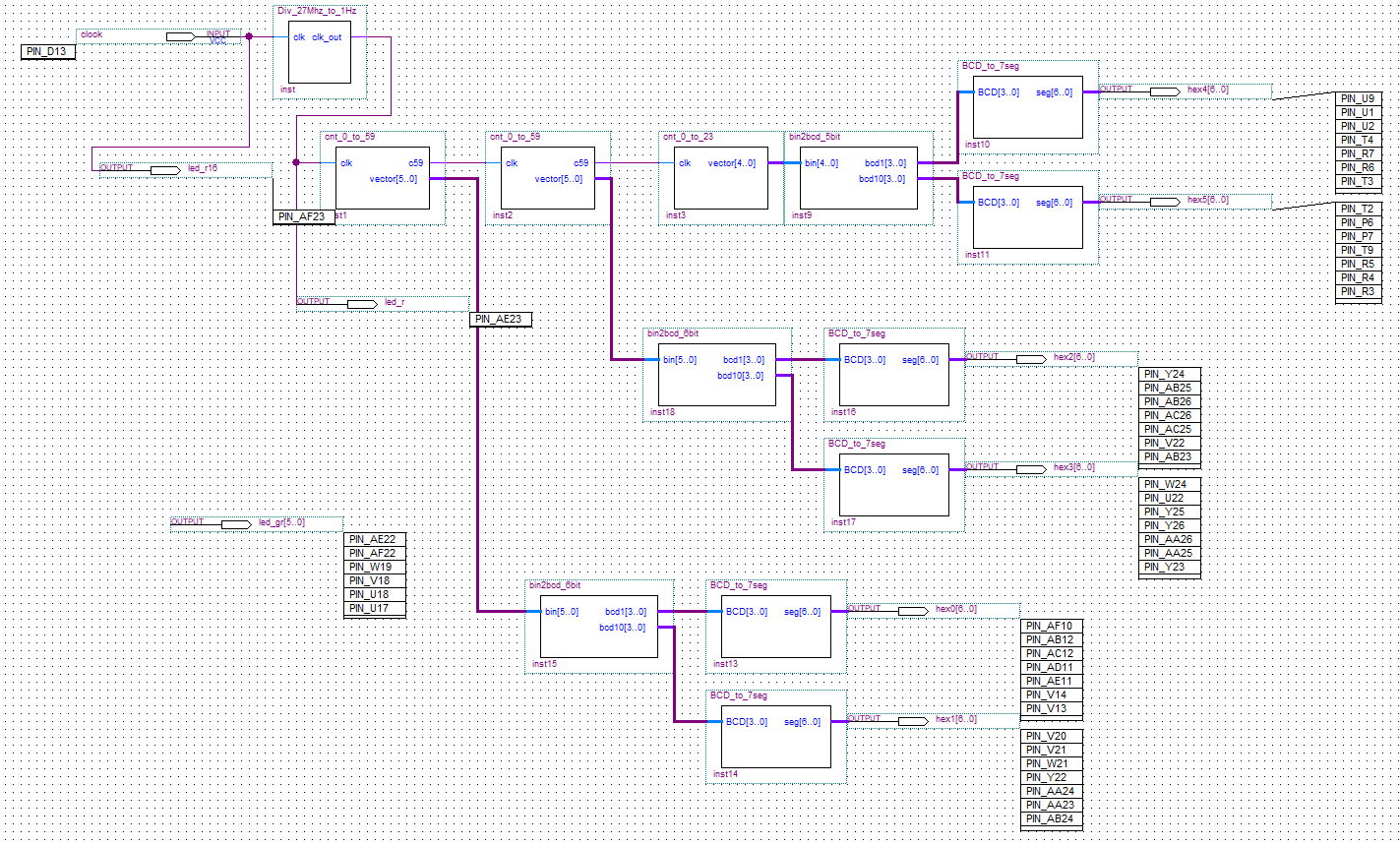
end process;

end converter\_behavior;

Перетворювач одного розряду в 7-сегментний код (інверсна таблиця істиності):

library ieee;  
use ieee.std\_logic\_1164.all;  
use ieee.std\_logic\_unsigned.all;  
  
entity BCD\_to\_7seg is  
port(   
BCD:in std\_logic\_vector(3 downto 0);   
seg:out std\_logic\_vector(6 downto 0)  
);  
  
end BCD\_to\_7seg;  
  
architecture conv\_behavior of BCD\_to\_7seg is  
begin  
process(BCD)  
begin  
if BCD = "0000" then seg <= "0000001";--0  
elsif BCD = "0001" then seg <= "1001111";--1  
elsif BCD = "0010" then seg <= "0010010";--2   
elsif BCD = "0011" then seg <= "0000110";--3  
elsif BCD = "0100" then seg <= "1001100";--4  
elsif BCD = "0101" then seg <= "0100100";--5  
elsif BCD = "0110" then seg <= "0100000";--6  
elsif BCD = "0111" then seg <= "0001111";--7  
elsif BCD = "1000" then seg <= "0000000";--8   
elsif BCD = "1001" then seg <= "0000100";--9  
else seg <= "1001001";--err  
end if;  
end process;  
end conv\_behavior;

Зібрана схема:



Підключення пінів:

