

# Układy cyfrowe i systemy wbudowane

## Sprawozdanie z laboratorium

Data	Tytuł zajęć	Uczestnicy
08.11.2017 13:00	Implementacja prostych układów w języku VHDL	Iwo Bujkiewicz (226203) Adrian Wąż (226042)

### Zadania

Na zajęciach należało opisać za pomocą języka VHDL, a następnie zaprogramować na mikroukładzie CPLD, następujące układy logiczne:

1. Licznik synchroniczny modulo 9 negatywny, pracujący w kodzie +3
2. Komparator 4-bitowy pracujący w kodzie Aikena, z wyjściem 3-bitowym (mniejsze, większe, równe)

### Kolejne etapy realizacji

#### Licznik synchroniczny

Licznik miał za zadanie realizować dokładnie tę samą funkcjonalność, co licznik z poprzednich zajęć, tj. odliczać od 8 do 0, zgodnie z taktem zegara, w kodzie z obciążeniem +3. Kod VHDL licznika prezentuje poniższy listing.

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use ieee.numeric_std.all;

entity counter_vhdl is
    Port (C : in  STD_LOGIC;
          CLR : in  STD_LOGIC;
          Q : out STD_LOGIC_VECTOR(3 downto 0));
end counter_vhdl;

architecture Behavioral of counter_vhdl is
    signal tmp: UNSIGNED(3 downto 0);
begin
    process (C, CLR)
    begin
        if (CLR='1') then
            tmp <= "1011";
        elsif rising_edge( C ) then
            if tmp="0011" then
                tmp<="1011";
            else
                tmp <= tmp - 1;
            end if;
        end if;
    end process;
    Q <= STD_LOGIC_VECTOR( tmp );
end Behavioral;
```

Mikroukład CPLD zaprogramowany na podstawie powyższego kodu działał zgodnie z oczekiwaniami. Wejścia i wyjścia wyprowadzono na porty mikroukładu zgodnie z poniższą tabelą.

Nazwa portu	Węzeł
Clk_LF	C
K0	CLR
LED0	Q(0)
LED1	Q(1)
LED2	Q(2)
LED3	Q(3)

## Komparator

Komparator miał za zadanie stwierdzić, czy pierwsza ze znajdujących się na wejściu układu liczb binarnych (A) jest mniejsza, większa czy równa drugiej liczbie (B). Jako, że komparator miał pracować w kodzie Aikena, zaimplementowano dodatkowo czwarty bit wyjścia, którego stan wysoki oznaczał wystąpienie wyjątku, gdy przynajmniej jedna z liczb była poza zakresem zdefiniowanym w kodzie Aikena. Stan wysoki bitu wyjścia, kolejno od najstarszego do najmłodszego, oznaczał: wyjątek,  $A < B$ ,  $A > B$ ,  $A == B$ . Poniżej zamieszczono kod implementacji komparatora.

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

entity comparator_vhdl is
    Port ( IN_A : in  STD_LOGIC_VECTOR(3 downto 0);
          IN_B : in  STD_LOGIC_VECTOR(3 downto 0);
          CMP : out  STD_LOGIC_VECTOR(3 downto 0));
end comparator_vhdl;

architecture Behavioral of comparator_vhdl is
begin
    process(IN_A, IN_B)
    begin
        if ((IN_A > "0100") and (IN_A < "1011"))
            or ((IN_B > "0100") and (IN_B < "1011")) then
            CMP <= "1000";
        elsif (IN_A < IN_B) then
            CMP <= "0100";
        elsif (IN_A > IN_B) then
            CMP <= "0010";
        else
            CMP <= "0001";
        end if;
    end process;
end Behavioral;
```

Programowanie na mikroukładzie CPLD przebiegło bez zarzutu. Wejścia i wyjścia wyprowadzono na porty mikroukładu zgodnie z poniższą tabelą.

Nazwa portu	Węzeł
K0	IN_B(0)
K1	IN_B(1)
K2	IN_B(2)
K3	IN_B(3)
K4	IN_A(0)
K5	IN_A(1)
K6	IN_A(2)
K7	IN_A(3)
LED0	CMP(0)
LED1	CMP(1)
LED2	CMP(2)
LED3	CMP(3)