

# Sprawozdanie

## Laboratorium nr 13

**Temat:** Płyta Spartan-3E Starter: obsługa portu VGA, wyświetlanie obrazu na monitorze

### 1. Cele i założenia laboratorium:

Celem zajęć laboratoryjnych było zaprojektowanie, zasymulowanie i wykorzystanie w sprzęcie układu stanowiącego sterownik portu VGA obsługującego wyświetlacz o rozdzielczości 800x520 zdolnego wyświetlać obraz o rozdzielczości 640x480. Był to jedyny układ realizowany w czasie tych zajęć z powodu jego złożoności.

### 2. Synteza układu

#### 2.1. Opis wejść i wyjść modułu VGA

Wejściami modułu VGA są 3 linie – **colors\_in** - odpowiedzialne za kolor wyświetlanego piksela, linia zegara taktowanego z częstotliwością 50 MHz oraz linia resetująca stan liczników modułu. Wyjściami modułu są linie bezpośrednio sterujące wyświetlaczem podłączonym poprzez port VGA oraz wyprowadzenie stanów liczników poziomej i pionowej pozycji na wyświetlaczu z tym że wyjście **row** z przedziału wartości [0, 479] oznacza pozycję możliwą do wyświetlenia, a wartość 480 oznacza pozostałe wartości – niemożliwe do wyświetlenia. Analogicznie w przypadku wyjścia **column** – wartości [0,639] wyświetlane, 640 nie wyświetlane.

```
entity VGA is
    port(clk, reset : in std_logic;
          colors_in : in std_logic_vector ( 2 downto 0);
          r,g,b : out std_logic;
          hsync, vsync : out std_logic;
          row : out std_logic_vector(8 downto 0);
          column : out std_logic_vector(9 downto 0));
end VGA;
```

## 2.2. Opis procesów modułu VGA

### 2.2.1. Proces: dzielnik\_częstotliwosci

#### 2.2.1.1. Opis słowny

Proces jest niezbędny ze względu na dwukrotnie większą częstotliwość zegara urządzenia względem potrzebnej częstotliwości do obsługi portu VGA która wynosi 25 MHz. Proces ten zmienia odpowiednio wartość sygnału wewnętrznego **nasz\_clock**, który dalej jest wykorzystywany innych procesach modułu jako wejście zegarowe.

#### 2.2.1.2. Kod VHDL

```
dzielnik_czestotliwosci: process(clk)
begin
    if rising_edge(clk) then
        nasz_clock <= not nasz_clock;
    end if;
end process;
```

### 2.2.2. Proces: liczniki

#### 2.2.2.1. Opis słowny

Zadaniem tego procesu jest kontrola liczników wskazujących na pozycję na wyświetlaczu w pionie i poziomie. Oprócz inkrementacji licznika **hcounter** odpowiadającego za pozycję w poziomie co takt zegara wewnętrznego, oraz inkrementacji licznika **vcounter** odpowiadającego za pozycję w pionie co przepełnienie licznika **hcounter**, proces odpowiada za ustawienie wartości wyjść **row** i **column**.

#### 2.2.2.2. Kod VHDL

```
liczniki : process (nasz_clock)
begin
    if rising_edge(nasz_clock) then
        if hcounter = 799 then
            hcounter <= "0000000000";
            if vcounter = 520 then
                vcounter <= "0000000000";
            else
                vcounter <= vcounter + 1;
            end if;
        else
            hcounter <= hcounter + 1;
        end if;

        if hcounter >= 48 and hcounter <= 687 then
            column <= hcounter - 48;
        else
            column <= "1010000000";
        end if;
    end if;
end process;
```

```

        if vcounter >= 29 and vcounter <= 509 then
            row <= vcounter(8 downto 0) - 29;
        else
            row <= "111100000";
        end if;
    end if;
end process;

```

### 2.2.3. Proces: sterowanie\_synchronizacją

#### 2.2.3.1. Opis słowny

Proces odpowiedzialny jest za sterowanie wyjściami synchronizacji **vsync**, **hsync**. Zgodnie z dokumentacją sygnał **hsync** ma mieć wartość 0 przez  $3,84 \mu s$ , a sygnał **vsync** ma mieć wartość 0 przez  $64 \mu s$ . Po przeprowadzeniu symulacji stwierdzono poprawność działania układu, czego dowodem są następujące rysunki:



#### 2.2.3.2. Kod VHDL

```

sterowanie_synchronizacja : process (hcounter, vcounter)
begin
    if hcounter >= 704 then
        hsync <= '0';
    end if;
end process;

```

```

else
    hsync <= '1';
end if;

if vcounter >= 519 then
    vsync <= '0';
else
    vsync <= '1';
end if;
end process;

```

#### 2.2.4. Proces: kolory

##### 2.2.4.1. Opis słowny

Proces ten jest tylko odpowiedzialny za to by w momencie pozycji liczników wyświetlacza poza obszarem wyświetlanym linie były ustawione w stan 0. W przeciwnym razie widoczne były by ślady powrotów plamki na początek wiersza. W przypadku gdy liczniki wskazują na pozycję w obszarze wyświetlania proces przypisuje wejściowe wartości barw do wyjść sterujących kolorami

##### 2.2.4.2. Kod VHDL

```

kolory : process ( hcounter, vcounter, colors_in)
begin
    if hcounter >= 48 and hcounter <= 687 and vcounter >= 29 and vcounter <=
                                                                    509 then

        r <= colors_in( 0 );
        g <= colors_in( 1 );
        b <= colors_in( 2 );

    else
        r <= '0';
        g <= '0';
        b <= '0';
    end if;
end process;

```

#### 2.3. Opis modułu color\_giver

##### 2.3.1. Opis słowny

Moduł ten odpowiada za zadawanie odpowiednich wartości kolorów sterownikowi VGA w zależności od wskazywanej przez niego pozycji. W zrealizowanym na zajęciach układzie zaimplementowano wyświetlanie trzech poziomych pasów o kolorach odpowiednio czerwony, zielonym i niebieskim.

### 2.3.2. Kod VHDL

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

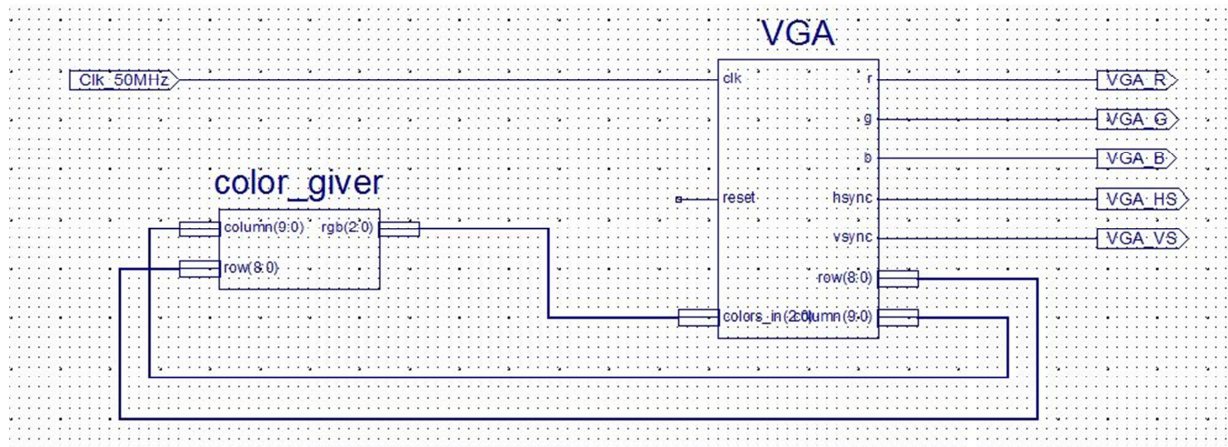
use IEEE.NUMERIC_STD.ALL;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;

entity color_giver is
    Port ( column : in STD_LOGIC_VECTOR (9 downto 0);
          row : in STD_LOGIC_VECTOR (8 downto 0);
          rgb : out STD_LOGIC_VECTOR (2 downto 0));
end color_giver;

architecture Behavioral of color_giver is

begin
    process(column)
    begin
        if column /= 640 and row /= 480 then
            if row < 160 then
                rgb <= "001";
            elsif row < 320 then
                rgb <= "010";
            else
                rgb <= "100";
            end if;
        else
            rgb <= "000";
        end if;
    end process;
end Behavioral;
```

## 2.4. Schemat układu



## 3. Wnioski

W czasie realizowania zadania laboratoryjnego nie napotkano większych trudności. Ważne jest zapoznanie się z charakterystyką obsługi portu VGA. Niewłaściwa interpretacja pozycji plamki i sygnałów synchronizacji może powodować niewłaściwe wyświetlanie obrazu lub w skrajnych przypadkach ryzyko uszkodzenia sprzętu.