Piotr Chmiel 200608 Układy Cyfrowe i Systemy Wbudowane 1

Piotr Sotor 200882 Data laboratorium: 17:05 15.01.2015

Prowadzący: mgr inż. Antoni Sterna

Sprawozdanie

Laboratorium nr 13

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

# 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.

# Synteza układu

* 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 przedziale 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; |

## Opis procesów modułu VGA

* + 1. **Proces: dzielnik\_częstotliwosci**
       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.

* + - 1. **Kod VHDL**

|  |
| --- |
| **dzielnik\_czestotliwosci: process(clk)**  **begin**  **if rising\_edge(clk) then**  **nasz\_clock <= not nasz\_clock;**  **end if;**  **end process;** |

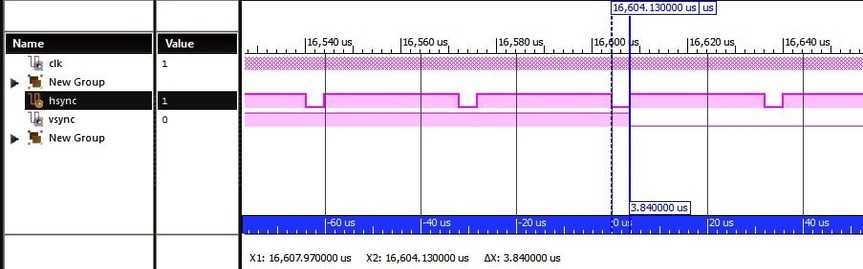
### Proces: liczniki

* + - 1. **Opis słowny**

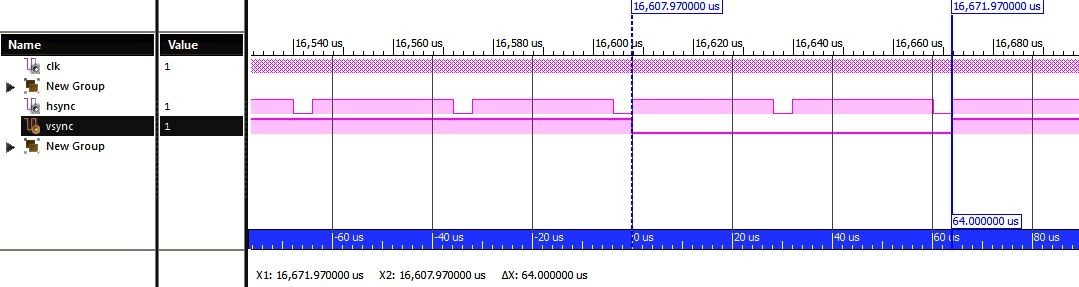
Zadaniem tego procesu jest kontrola liczników wskazujących na pozycję na wyświetlaczu w pionie i poziomie. Oprócz inkrementacji licznika **hcounter** odpowiadającego za pozycję w poziomie co takt zegara wewnętrznego, oraz inkrementacji licznika **vcounter** odpowiadającego za pozycję w pionie co przepełnienie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

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

* + 1. **Proces: sterowanie\_synchronizacją**
       1. **Opis słowny**

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



* + - 1. **Kod VHDL**

|  |
| --- |
| sterowanie\_synchronizacja : process (hcounter, vcounter)  begin  if hcounter >= 704 then  hsync <= '0';  else  hsync <= '1';  end if;    if vcounter >= 519 then  vsync <= '0';  else  vsync <= '1';  end if;  end process; |

* + 1. **Proces: kolory**
       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 ma początek wiersza.

W przypadku gdy liczniki wskazują na pozycję w obszarze wyświetlania proces przypisuje wejściowe wartości barw do wyjść sterujących kolorami

* + - 1. **Kod VHDL**

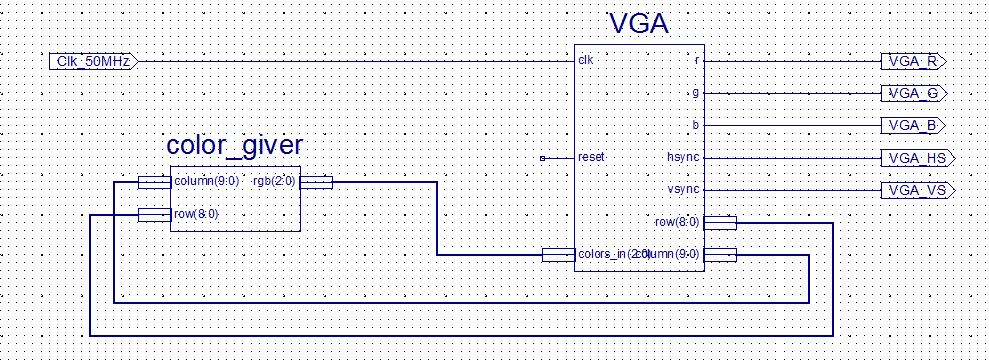
|  |
| --- |
| kolory : process ( hcounter, vcounter, colors\_in)  begin  if hcounter >= 48 and hcounter <= 687 and vcounter >= 29 and vcounter <= 509 then  r <= colors\_in( 0 );  g <= colors\_in( 1 );  b <= colors\_in( 2 );  else  r <= '0';  g <= '0';  b <= '0';  end if;  end process; |

* 1. **Opis modułu color\_giver**
     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.

* + 1. **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; |

* 1. **Schemat układu**

# 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.