

Actividad 3 Adaptador de monitor VGA

Objetivo

Implementación y síntesis de un circuito digital que genere las señales necesarias para controlar un monitor de vídeo (VGA).

Desarrollo

Para entender como es posible generar una imagen de video con la placa spartan 3 de Xilinx hay que entender como opera básicamente una tarjeta de video VGA. Un adaptador de VGA contiene 5 señales activas. Dos señales (compatibles TTL) para sincronización del video. Tres señales analógicas, cuyo rango se encuentra entre [0.7, 1Vpp], para generar o controlar el color. Podemos etiquetar estas señales según su funcionalidad: sincronismo horizontal, sincronismo vertical, y los colores R (red), G (green) y B(blue).

Un formato VGA estándar contiene una pantalla de 640 x 480 pixeles. La señal de video debe redibujar una pantalla completa 60 veces por segundo para suministrar movimiento y reducir el parpadeo. Este tiempo es llamado periodo de refresco. El ojo humano puede detectar parpadeo a refrescos menores de 30Hz.

Sobre la placa de desarrollo existe un reloj (50Mhz) que hay que dividirlo para producir uno de 25Mhz y producir un refresco fijo de aproximadamente 60Hz. El color de cada píxel es determinado mediante el valor de la señal RGB cuando ésta pasa por cada píxel. En un modo de 640x480 con 60Hz de refresco, el paso por cada píxel dura aproximadamente 40ns.

Documentación:

De la placa la pueden encontrar en xilinx

<http://www.digilentinc.com/Products/Detail.cfm?Prod=S3BOARD>

Códigos vhdl en libro de texto *FPGA Prototyping by VHDL Examples: Xilinx Spartan-3*,

http://academic.csuohio.edu/chu_p/rtl/fpga_vhdl.html

Códigos:

-- generar reloj de 25Mhz a partir de uno de 50

```
process (clk50_in)
begin
    if clk50_in'event and clk50_in='1' then
        clk25 <= not clk25;
    end if;
end process;
```

Cambio de color cada segundo de la pantalla,

```
-- change color every one second
p1: process (clk25)
    variable cnt: integer;
begin
    if clk25'event and clk25='1' then
        cnt := cnt + 1;
        if cnt = 25000000 then
            rgb <= rgb + "001";
            cnt := 0;
        end if;
    end if;
end process;
```

Para el proceso principal podrían usar esta como referencia el siguiente código, al que habrá que añadirle Reset sustituyendo el wait por una construcción if-elseif.

```
library IEEE;
use IEEE.STD_LOGIC_1164.all;
use IEEE.STD_LOGIC_ARITH.all;
use IEEE.STD_LOGIC_UNSIGNED.all;

ENTITY VGA_SYNC IS
    PORT( clock_25Mhz, red, green, blue          : IN STD_LOGIC;
          red_out, green_out, blue_out, horiz_sync_out,
vert_sync_out          : OUT STD_LOGIC;
          pixel_row, pixel_column: OUT STD_LOGIC_VECTOR(9
DOWNT0 0));
```

```

END VGA_SYNC;
ARCHITECTURE a OF VGA_SYNC IS
    SIGNAL horiz_sync, vert_sync : STD_LOGIC;
    SIGNAL video_on, video_on_v, video_on_h : STD_LOGIC;
    SIGNAL h_count, v_count :STD_LOGIC_VECTOR(9 DOWNT0 0);

BEGIN

-- video_on esta alta en ventana activa
video_on <= video_on_H AND video_on_V;

PROCESS
BEGIN
    WAIT UNTIL(clock_25Mhz'EVENT) AND (clock_25Mhz='1');

--Generate Horizontal and Vertical Timing Signals for Video Signal
-- H_count counts pixels (640 + extra time for sync signals)
--
-- Horiz_sync -----
-- H_count      0          640          659          755
799
--
    IF (h_count = 799) THEN
        h_count <= "0000000000";
    ELSE
        h_count <= h_count + 1;
    END IF;

--Generate Horizontal Sync Signal using H_count
    IF (h_count <= 755) AND (h_count >= 659) THEN
        horiz_sync <= '0';
    ELSE
        horiz_sync <= '1';
    END IF;

--V_count counts rows of pixels (480 + extra time for sync signals)
--
-- Vert_sync -----
-- V_count      0          480          493-
494          524
--
    IF (v_count >= 524) AND (h_count >= 799) THEN
        v_count <= "0000000000";
    ELSIF (h_count = 699) THEN
        v_count <= v_count + 1;
    END IF;

-- Generate Vertical Sync Signal using V_count
    IF (v_count <= 494) AND (v_count >= 493) THEN
        vert_sync <= '0';
    ELSE
        vert_sync <= '1';
    END IF;

-- Generate Video on Screen Signals for Pixel Data
    IF (h_count <= 639) THEN
        video_on_h <= '1';
        pixel_column <= h_count;

```

```

ELSE
    video_on_h <= '0';
END IF;

IF (v_count <= 479) THEN
    video_on_v <= '1';
    pixel_row <= v_count;
ELSE
    video_on_v <= '0';
END IF;

-- FF
red_out <= red AND video_on;
green_out <= green AND video_on;
blue_out <= blue AND video_on;
horiz_sync_out <= horiz_sync;
vert_sync_out <= vert_sync;

END PROCESS;
END a;

```

Estos códigos fueron hechos para una placa con una Spartan 3 , hay que adaptarlos. A la nueva placa

Se pide:

Diseñar, simular y sintetizar en VHDL el código que permita generar las señales de sincronismo y color RGB para un adaptador de VGA.

Como propuesta el módulo VHDL podría tener las entradas y salidas que se indican en la figura siguiente:

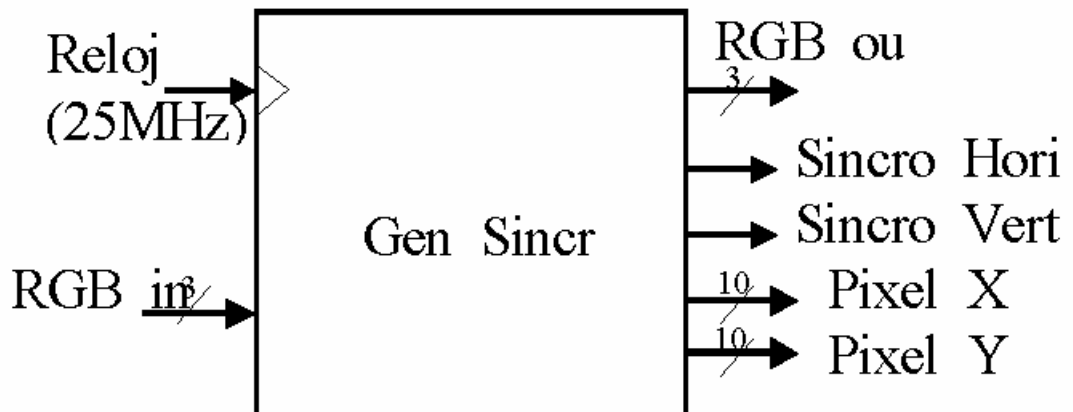


Diagrama de bloque del generador de sincronismo de la VGA. Representación de sus entradas y salidas.