**Generación de Señal  VGA  implementado con FPGA**

El término **Video Graphics Array** (**VGA**) (Adaptador Gráfico de Video) se utiliza tanto para denominar a una pantalla de computadora analógica estándar, al conector VGA de 15 contactos. Se aplicó en las tarjeta gráfica que comercializó  IBM por primera vez en 1988 con la resolución  **640 × 480**.

Una **tarjeta gráfica**, **tarjeta de vídeo**, **placa de vídeo**, **tarjeta aceleradora de gráficos** o **adaptador de pantalla**, es una tarjeta de expansión para una computadora, encargada de procesar los datos provenientes de la CPU y transformarlos en información comprensible y representable en un dispositivo de salida, como un monitor

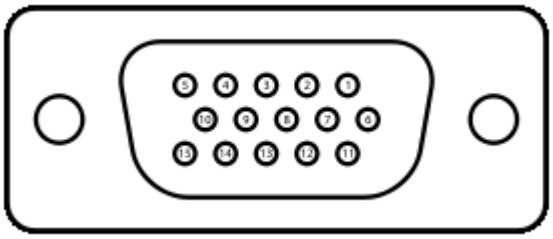
La norma VGA fue oficialmente reemplazada por Extended Graphics Array de IBM pero en realidad ha sido sustituida por numerosas extensiones clónicas ligeramente distintas a VGA realizadas por los fabricantes y que llegaron a ser conocidas en conjunto como "Super VGA".

| Conector VGA (Conector VGA (DE-15/HD-15) | |
|---|---|
|  |  |
| Un conector VGA | |

| | |
|---|---|
| **Tipo** | Conector analógico de video en alta definición |
| **Historia de producción** | |
| **Diseñador** | IBM |
| **Diseñado en** | 1987 |
| **Producido** | 1987 - Presente |
| **Especificaciones** | |
| **Señal de Video** | RGB más sincronismo H y V |

| | |
|---|---|
| **Señal de Datos** | I²C canal de datos para información DDC |
| **Pines** | 15 |
| **Conector** | DE-15 |

**Patillaje**

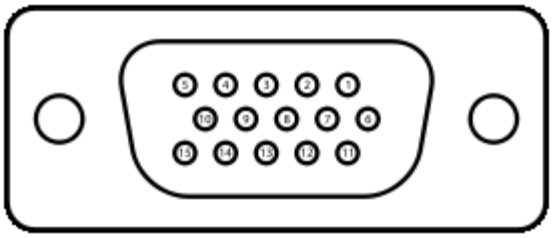


Un conector DE15 hembra.

| | | |
|---|---|---|
| **Pin 1** | RED | Canal Rojo |
| **Pin 2** | GREEN | Canal Verde |
| **Pin 3** | BLUE | Canal Azul |
| **Pin 4** | N/C | Sin contacto |
| **Pin 5** | GND | Tierra (HSync) |
| **Pin 6** | RED_RTN | Vuelta Rojo |
| **Pin 7** | GREEN_RTN | Vuelta Verde |
| **Pin 8** | BLUE_RTN | Vuelta Azul |
| **Pin 9** | +5 V | +5 V (Corriente continua) |
| **Pin 10** | GND | tierra (Sincr. Vert, Corriente continua) |
| **Pin 11** | N/C | Sin contacto |
| **Pin 12** | SDA | I²C datos |
| **Pin 13** | HSync | Sincronización horizontal |
| **Pin 14** | VSync | Sincronización vertical |
| **Pin 15** | SCLAdfgg | $I^2$Velocidad Reloj |

Un conector VGA como se le conoce comúnmente (otros nombres incluyen conector RGBHV, D-sub 15, sub mini mini D15 y D15), de tres hileras de 15 pines DE-15. Hay cuatro versiones: original, DDC2, el más antiguo y menos flexible DE-9, y un Mini-VGA utilizados para computadoras portátiles. El conector común de 15 pines se encuentra en la mayoría de las tarjetas gráficas, monitores de computadoras, y otros dispositivos, es casi universalmente llamado "HD-15". HD es de "alta densidad", que la distingue de los conectores que tienen el mismo factor de forma, pero sólo en 2 filas de pines. Sin embargo,

este conector es a menudo erróneamente denominado DB-15 o HDB-15. Los conectores VGA y su correspondiente cableado casi siempre son utilizados exclusivamente para transportar componentes analógicos RGBHV (rojo - verde - azul - sincronización horizontal - sincronización vertical), junto con señales de vídeo DDC2 reloj digital y datos. En caso de que el tamaño sea una limitación (como portátiles) un puerto mini-VGA puede figurar en ocasiones en lugar de las de tamaño completo conector VGA. Con la revolución digital, a partir de 2009 se comienza a reemplazar estos conectores VGA por conectores HDMI que debido a sus características avanzadas en tarjetas gráficas, pantallas y monitores actuales.

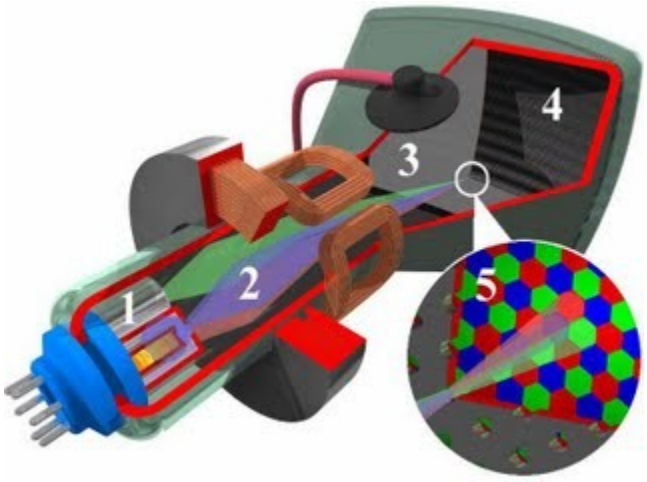


| | | |
|---|---|---|
| **Pin 1** | RED | Canal Rojo |
| **Pin 2** | GREEN | Canal Verde |
| **Pin 3** | BLUE | Canal Azul |
| **Pin 4** | N/C | Sin contacto |
| **Pin 5** | GND | Tierra (HSync) |
| **Pin 6** | RED_RTN | Vuelta Rojo |
| **Pin 7** | GREEN_RTN | Vuelta Verde |
| **Pin 8** | BLUE_RTN | Vuelta Azul |
| **Pin 9** | +5 V | +5 V (Corriente continua) |
| **Pin 10** | GND | tierra (Sincr. Vert, Corriente continua) |
| **Pin 11** | N/C | Sin contacto |
| **Pin 12** | SDA | I²C datos |
| **Pin 13** | HSync | Sincronización horizontal |
| **Pin 14** | VSync | Sincronización vertical |
| **Pin 15** | SCLAdfgg | $I^2$Velocidad Reloj |

## Monitor VGA en modo 640x480

En este apartado se explicará cómo deben generarse las señales HS, VS, R, G y B para representar una imagen en un monitor en modo 60Hz, 640x480 y 8 colores. Es decir, explicaremos brevemente el protocolo VGA.

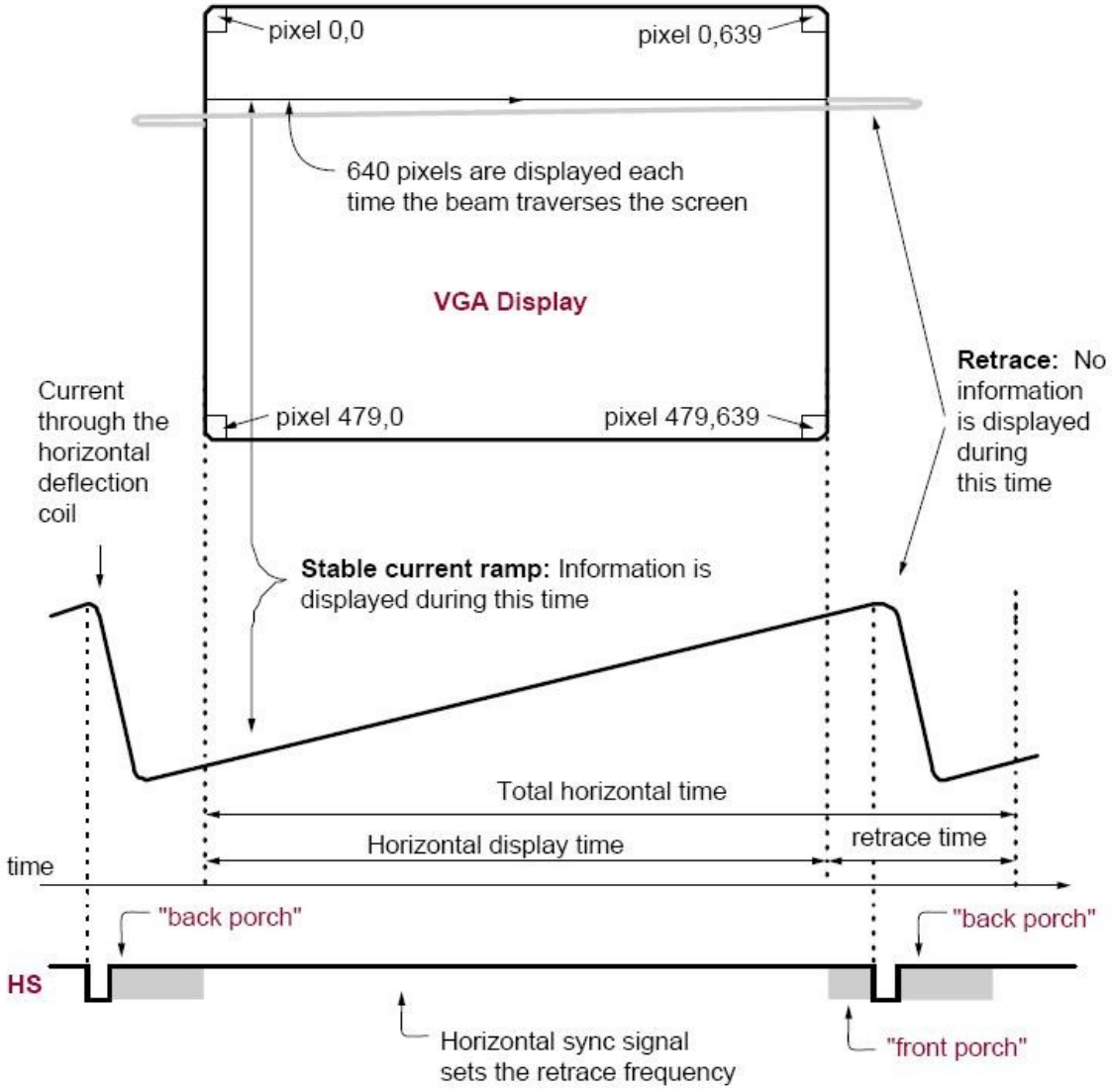
Para explicar el protocolo VGA es necesario conocer el funcionamiento básico de un Tubo de Rayos Catódicos. El funcionamiento es:




Tres cañones de electrones (1) producen tres haces de electrones, uno por cada color básico (rojo, azul y verde). Estos haces de electrones se focalizan (2), utilizando una bobina, para conseguir que converjan en un punto de la pantalla. Posteriormente, mediante un campo magnético generado por un par de bobinas (horizontal y vertical) se direccionan al punto deseado de la pantalla. En la pantalla de visualización (5), los rayos son separados (3) por una máscara e inciden en una capa fosforescente con zonas receptivas para cada color (4).

Por otro lado, la intensidad luminosa dependerá de la potencia a la que se emite el haz de electrones, controlable mediante las señales analógicas R, G y B.

Se representarán imágenes en pantalla haciendo que el haz de electrones la recorra en una sucesión de líneas (de izquierda a derecha) comenzando por la esquina superior izquierda (figura 4).



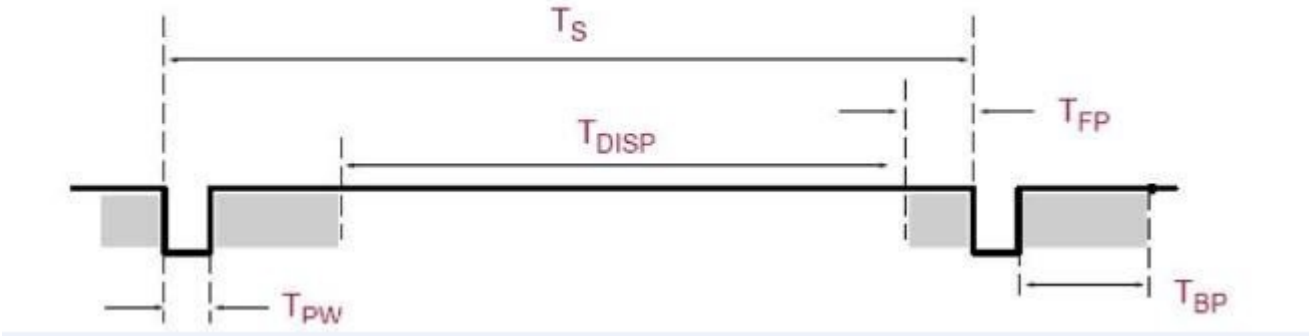

La onda en forma de diente de sierra representa la corriente que pasa por la bobina que crea que campo magnético para la deflexión horizontal. Se puede comprobar, que se representa la imagen (Horizontal display time) cuando el haz de electrones va de izquierda a derecha y está apuntando correctamente a la pantalla, existiendo un tiempo (retrace time) para que el cañón vuelva al comienzo de la siguiente línea. El pulso de **sincronismo vertical (HS)** marca el instante en que el cañón debe **comenzar una nueva línea**, pudiendo distinguir un tiempo anterior ("front porch") en que el cañón está apuntando fuera de la pantalla y un tiempo posterior ("pack porch") necesario para que el haz se posicione en el inicio de la siguiente línea.
Es importante que durante todo el "retrace time" los cañones de electrones estén apagados.

La frecuencia a la que se proporcionan los pulsos de sincronismos vertical y horizontal determina la velocidad y el número de veces que el haz de electrones recorre la pantalla. De esta forma seleccionamos la resolución de visualización de la imagen en el monitor. El estándar VGA admite diferentes resoluciones, para las que tendríamos que utilizar

diferentes frecuencias en HS y VS.

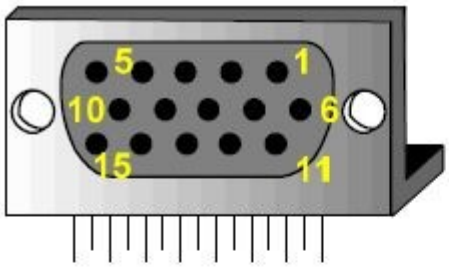
En esta práctica vamos diseñar un controlador VGA para la resolución 640x480 píxeles y 60 Hz de refresco de pantalla. Consideraremos una frecuencia de píxel de 25MHz, para lo que dividiremos por dos la frecuencia de reloj disponible de 50MHz utilizando un biestable.

| Symbol | Parameter | Vertical Sync | | | Horizontal Sync | |
|---|---|---|---|---|---|---|
| | | Time | Clocks | Lines | Time | Clocks |
| $T_S$ | Sync pulse time | 16.7 ms | 416,800 | 521 | 32 µs | 800 |
| $T_{DISP}$ | Display time | 15.36 ms | 384,000 | 480 | 25.6 µs | 640 |
| $T_{PW}$ | Pulse width | 64 µs | 1,600 | 2 | 3.84 µs | 96 |
| $T_{FP}$ | Front porch | 320 µs | 8,000 | 10 | 640 ns | 16 |
| $T_{BP}$ | Back porch | 928 µs | 23,200 | 29 | 1.92 µs | 48 |

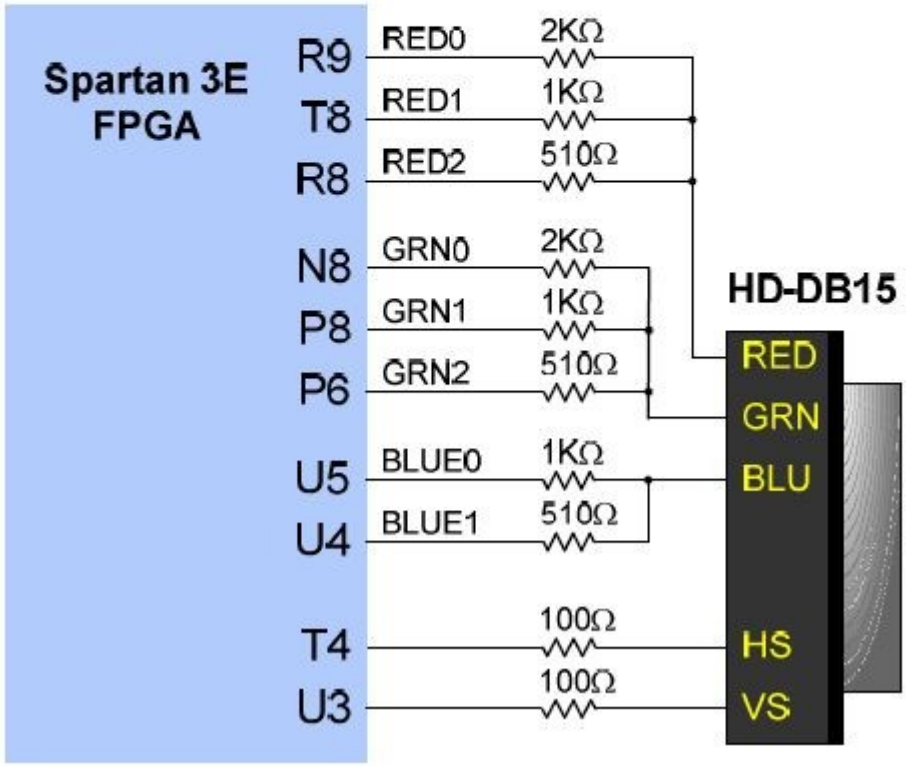



Todos los datos para generar la señal VGA se muestra en la imagen anterior. En la tabla la señal Clk(Clock) se refiere a la señal de reloj de frecuencia de pixel a 25 Mhz y no a la frecuencia de reloj de la FPGA.

**VGA Port (colores)**
La Nexys2 usa señales de la FPGA para crear 8-bit de color y dos HS VS
Se utilize resistencias para generar la señal analogical.
0V (fully off) and 0.7V (fully on).
Se obtienen256 colores.




Pin 1: Red
Pin 2: Grn
Pin 3: Blue
Pin 13: HS
Pin 14: VS
Pin 5: GND
Pin 6: Red GND
Pin 7: Grn GND
Pin 8: Blu GND
Pin 10: Sync GND



**Como describir las señales VS- HS en VHDL**

El problema a resolver es como generar los pulsos VS y HS con la frecuencia y la duración adecuados para que sean entendidos por un monitor compatible con el estandar VGA.

```
----------------------------------------------------------------------
-- vga_controller_640_60.vhd
----------------------------------------------------------------------
-- Author : Ulrich Zoltán
--         Copyright 2006 Digilent, Inc.
----------------------------------------------------------------------
-- Software version : Xilinx ISE 7.1.04i
--                    WebPack
-- Device                    : 3s200ft256-4
----------------------------------------------------------------------
-- This file contains the logic to generate the synchronization signals,
-- horizontal and vertical pixel counter and video disable signal
-- for the 640x480@60Hz resolution.
----------------------------------------------------------------------
--  Behavioral description
----------------------------------------------------------------------
-- Please read the following article on the web regarding the
-- vga video timings:
-- http://www.epanorama.net/documents/pc/vga_timing.html

-- This module generates the video synch pulses for the monitor to
-- enter 640x480@60Hz resolution state. It also provides horizontal
-- and vertical counters for the currently displayed pixel and a blank
-- signal that is active when the pixel is not inside the visible screen
-- and the color outputs should be reset to 0.

-- timing diagram for the horizontal synch signal (HS)
-- 0                    648    744          800 (pixels)
-- -------------------------|_______|-----------------
-- timing diagram for the vertical synch signal (VS)
-- 0                             482    484  525 (lines)
-- -----------------------------------|_______|-------

-- The blank signal is delayed one pixel clock period (40ns) from where
-- the pixel leaves the visible screen, according to the counters, to
-- account for the pixel pipeline delay. This delay happens because
-- it takes time from when the counters indicate current pixel should
-- be displayed to when the color data actually arrives at the monitor
-- pins (memory read delays, synchronization delays).
```

```
----------------------------------------------------------------------
--  Port definitions
----------------------------------------------------------------------
-- rst             - global reset signal
-- pixel_clk        - input pin, from dcm_25MHz
--                 - the clock signal generated by a DCM that has
--                 - a frequency of 25MHz.
-- HS               - output pin, to monitor
--                 - horizontal synch pulse
-- VS               - output pin, to monitor
--                 - vertical synch pulse
-- hcount           - output pin, 11 bits, to clients
--                 - horizontal count of the currently displayed
--                 - pixel (even if not in visible area)
-- vcount           - output pin, 11 bits, to clients
--                 - vertical count of the currently active video
--                 - line (even if not in visible area)
-- blank            - output pin, to clients
--                 - active when pixel is not in visible area.
----------------------------------------------------------------------
-- Revision History:
-- 09/18/2006(UlrichZ): created
----------------------------------------------------------------------

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_ARITH.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

-- simulation library
library UNISIM;
use UNISIM.VComponents.all;

-- the vga_controller_640_60 entity declaration
-- read above for behavioral description and port definitions.
entity vga_controller_640_60 is
port(
  rst       : in std_logic;
  pixel_clk   : in std_logic;

  HS        : out std_logic;
  VS        : out std_logic;
```

```
    hcount      : out std_logic_vector(10 downto 0);
    vcount      : out std_logic_vector(10 downto 0);
    blank       : out std_logic
);
end vga_controller_640_60;

architecture Behavioral of vga_controller_640_60 is

----------------------------------------------------------------------
-- CONSTANTS
----------------------------------------------------------------------

-- maximum value for the horizontal pixel counter
constant HMAX  : std_logic_vector(10 downto 0) := "01100100000"; -- 800
-- maximum value for the vertical pixel counter
constant VMAX  : std_logic_vector(10 downto 0) := "01000001101"; -- 525
-- total number of visible columns
constant HLINES: std_logic_vector(10 downto 0) := "01010000000"; -- 640
-- value for the horizontal counter where front porch ends
constant HFP   : std_logic_vector(10 downto 0) := "01010001000"; -- 648
-- value for the horizontal counter where the synch pulse ends
constant HSP   : std_logic_vector(10 downto 0) := "01011101000"; -- 744
-- total number of visible lines
constant VLINES: std_logic_vector(10 downto 0) := "00111100000"; -- 480
-- value for the vertical counter where the front porch ends
constant VFP   : std_logic_vector(10 downto 0) := "00111100010"; -- 482
-- value for the vertical counter where the synch pulse ends
constant VSP   : std_logic_vector(10 downto 0) := "00111100100"; -- 484
-- polarity of the horizontal and vertical synch pulse
-- only one polarity used, because for this resolution they coincide.
constant SPP   : std_logic := '0';

----------------------------------------------------------------------
-- SIGNALS
----------------------------------------------------------------------

-- horizontal and vertical counters
signal hcounter : std_logic_vector(10 downto 0) := (others => '0');
signal vcounter : std_logic_vector(10 downto 0) := (others => '0');

-- active when inside visible screen area.
signal video_enable: std_logic;
```

```
begin

  -- output horizontal and vertical counters
  hcount <= hcounter;
  vcount <= vcounter;

  -- blank is active when outside screen visible area
  -- color output should be blacked (put on 0) when blank in active
  -- blank is delayed one pixel clock period from the video_enable
  -- signal to account for the pixel pipeline delay.
  blank <= not video_enable when rising_edge(pixel_clk);

  -- increment horizontal counter at pixel_clk rate
  -- until HMAX is reached, then reset and keep counting
  h_count: process(pixel_clk)
  begin
    if(rising_edge(pixel_clk)) then
      if(rst = '1') then
        hcounter <= (others => '0');
      elsif(hcounter = HMAX) then
        hcounter <= (others => '0');
      else
        hcounter <= hcounter + 1;
      end if;
    end if;
  end process h_count;

  -- increment vertical counter when one line is finished
  -- (horizontal counter reached HMAX)
  -- until VMAX is reached, then reset and keep counting
  v_count: process(pixel_clk)
  begin
    if(rising_edge(pixel_clk)) then
      if(rst = '1') then
        vcounter <= (others => '0');
      elsif(hcounter = HMAX) then
        if(vcounter = VMAX) then
          vcounter <= (others => '0');
        else
          vcounter <= vcounter + 1;
        end if;
```

```
      end if;
    end if;
  end process v_count;

  -- generate horizontal synch pulse
  -- when horizontal counter is between where the
  -- front porch ends and the synch pulse ends.
  -- The HS is active (with polarity SPP) for a total of 96 pixels.
  do_hs: process(pixel_clk)
  begin
    if(rising_edge(pixel_clk)) then
      if(hcounter >= HFP and hcounter < HSP) then
        HS <= SPP;
      else
        HS <= not SPP;
      end if;
    end if;
  end process do_hs;

  -- generate vertical synch pulse
  -- when vertical counter is between where the
  -- front porch ends and the synch pulse ends.
  -- The VS is active (with polarity SPP) for a total of 2 video lines
  -- = 2*HMAX = 1600 pixels.
  do_vs: process(pixel_clk)
  begin
    if(rising_edge(pixel_clk)) then
      if(vcounter >= VFP and vcounter < VSP) then
        VS <= SPP;
      else
        VS <= not SPP;
      end if;
    end if;
  end process do_vs;

  -- enable video output when pixel is in visible area
  video_enable <= '1' when (hcounter < HLINES and vcounter < VLINES) else '0';

end Behavioral;
```