VGA Driver and Image Generation Documentation

08-04-2023

Tags: #SEM4_PROJECT #VHDl #Version_1
Related files/topics:

- Zynq-7000, VHDL and Vivado
- VGA - Video Graphic Array

Driver and Entity description

Inputs:

- **100MHz clock** - the Zedboard has an onboard 100MHz and 33MHz clock. This clock signal needs to get converted to a specific frequency *(depends on the resolution to be driven)* to drive the VGA output., for 640x480, 60Hz it is 25MHz.

Outputs:

- **Hsync and Vsync** - responsible for determining when a new line starts or a new frame starts.
- **R, G, B** - analog signals for pixels.

The driver code consists of horizontal and vertical position counters, to keep track of the current pixel being displayed as well as to insure synchronization.

The code includes processes for **Hsync** and **Vsync** to ensure and synchronize properly between the monitor and video generator.

A **video on** process is implemented that enables or disables 'drawing' on the screen.

A reset button is also implemented for all processes.

Needed Timing Constants

For every supported resolution the front-porch, back-porch and sync constants differ. If the constants are wrong, the display driver wont work.

Horizontal Constants

- **Display interval/Horizontal display** *(HD or H_HIGH)*- 639, including 0
- **Right border**, horizontal front porch *(HFP)* - 16
- **Left border**, horizontal back porch *(HBP)* - 48
- **Sync Pulse** *(H_LOW or HSP)* - 96

Note: The constants are measured in pixels

Vertical Constants

- **Display interval/Vertical display** *(VD or V_HIGH)*- 480, including 0
- **Right border**, vertical front porch *(VFP)* - 10
- **Left border**, vertical back porch *(VBP)* - 33
- **Sync Pulse** *(V_LOW or VSP)* - 2

Note: The constants are measured in pixels

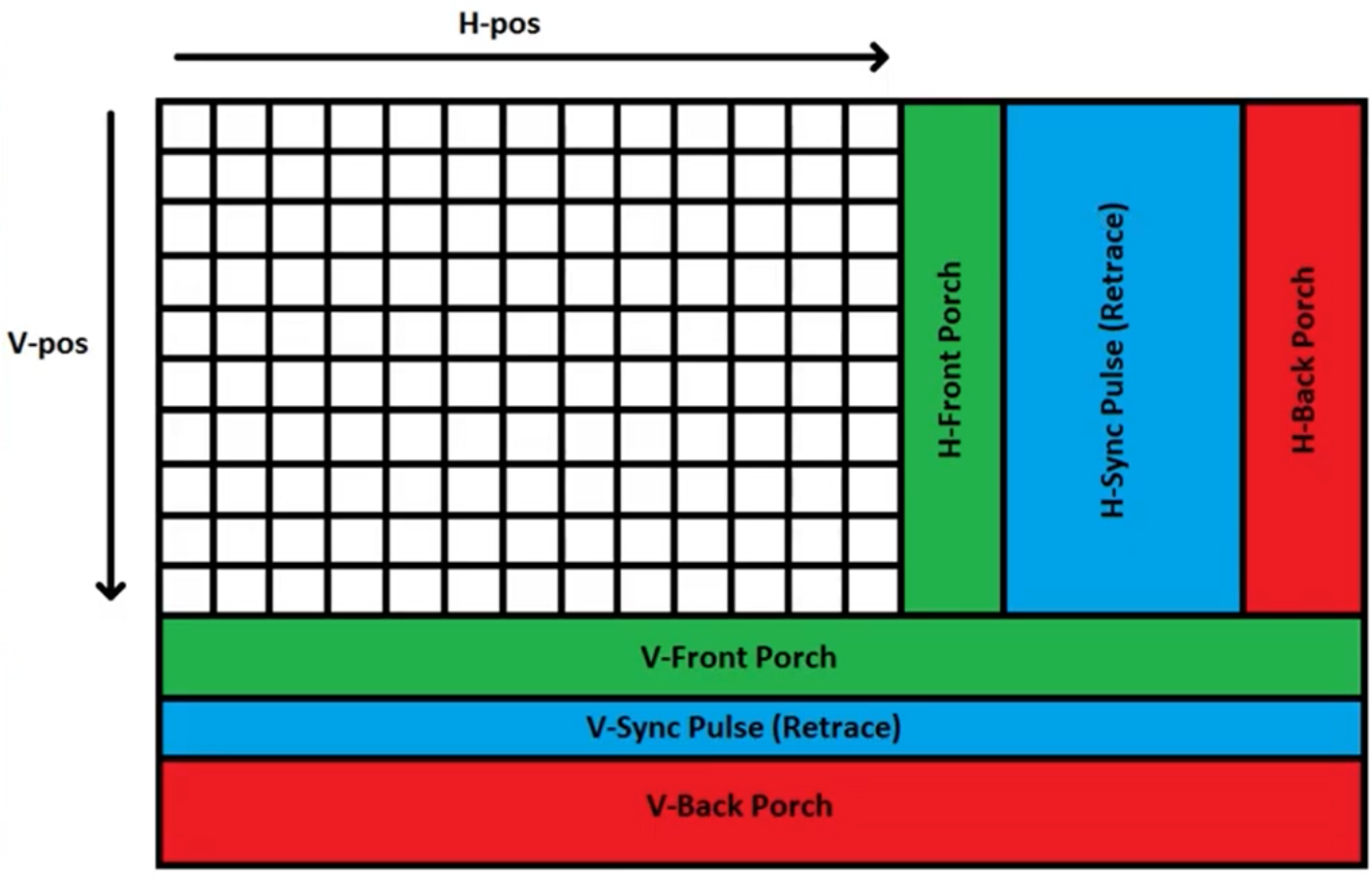
Position Counters

Image generation on the screen starts from the top-left corner. The **horizontal position counter** increments at each pixel clock cycle, and keeps track of the current horizontal position of the displayed pixel. When it reaches the end of a line, it resets to zero and starts counting again.

```
hor_pos_c: process(clk_25, RST)
    begin
        if (RST = '1')then
            hPos <= 0;
        elsif (clk_25'event and clk_25 = '1') then
            if (hPos = (HD + HFP + HSP + HBP)) then
                hPos <= 0;
            else
                hPos <= hPos + 1;
            end if;
        end if;
    end process;
```

The **vertical position counter** increments at the end of each line, and keeps track of the current vertical position of the displayed pixel. When it reaches the end of a frame, it resets to zero and starts counting again.

```
ver_pos_c: process(clk_25, RST)
    begin
        if (RST = '1')then
            vPos <= 0;
        elsif (clk_25'event and clk_25 = '1') then
            if (hPos = (HD + HFP + HSP + HBP)) then
                if (vPos = (VD + VFP + VSP + VBP)) then
                    VPos <= 0;
                else
                    vPos <= vPos + 1;
                end if;
            end if;
        end if;
    end process;
```



Vertical and Horizontal sync

The horizontal sync *(hsync)* signal indicates the start and end of a new line of pixels, and the vertical sync *(vsync)* signal indicates the start and end of a new frame. By using these signals, the monitor knows when to start a new line and when to start a new frame, which ensures that the image is displayed correctly.

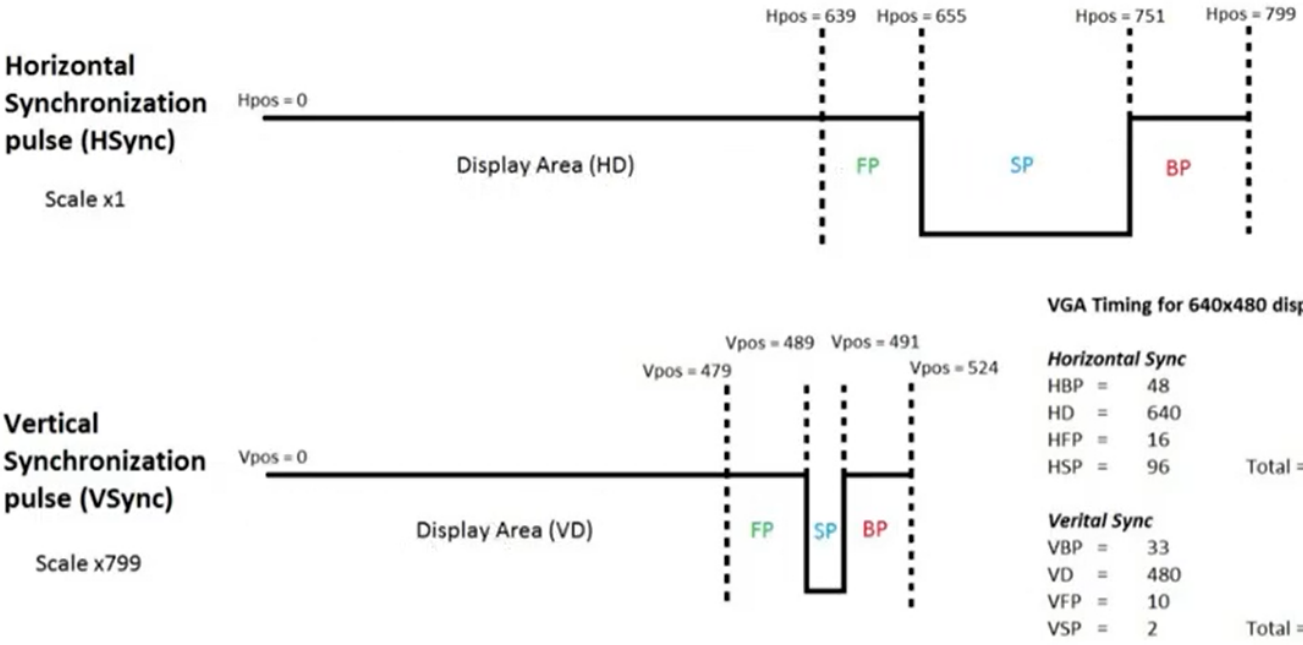
The hsync signal is typically a pulse that goes low at the start of each new line and high again when the line is finished.

```
Horizontal_sync: process(clk_25, RST, hPos)
    begin
        if (RST = '1')then
            Hsync <= '0';
        elsif (clk_25'event and clk_25 = '1') then
            if((hPos <= (HD + HFP)) OR (hPos > HD + HFP + HSP)) then
                Hsync <= '1';
            else
                Hsync <= '0';
            end if;
        end if;
    end process;
```

The vsync signal is typically a pulse that goes low at the start of each new frame and high again when the frame is finished.

```
Vertical_sync: process(clk_25, RST, vPos)
    begin
        if (RST = '1')then
            Vsync <= '0';
        elsif (clk_25'event and clk_25 = '1') then
            if((vPos <= (VD + VFP)) OR (vPos > VD + VFP + VSP)) then
                Vsync <= '1';
            else
```

```
                Vsync <= '0';
            end if;
        end if;
    end process;
```



VGA Timing for 640x480 display

Horizontal Sync

HBP	=	48	
HD	=	640	
HFP	=	16	
HSP	=	96	Total = 800

Verital Sync

VBP	=	33	
VD	=	480	
VFP	=	10	
VSP	=	2	Total = 525