Lecture-12

VHDL Design of VGA Video Interfaces

VGA (video graphics array) is a standard interface introduced by IBM in 1987 for connecting computers to analog video monitors.

The circuits responsible for generating, processing, and storing the video signals are called graphics controller (computer side) and display controller (monitor side).

![Diagram of computer connected to an analog VGA monitor and circuits responsible for video signals](image)

(a) Computer connected to an analog VGA monitor; (b) Circuits responsible for the video signals (graphics controller on the host side, display controller on the device side).

**Figure 12-1**

VGA monitors are CRTs (cathode ray tubes). Their current most basic resolution is 640 x 480 x 60Hz VGA; it consists of 640 columns by 480 lines of pixels (picture elements), refreshed 60 times per second.

![Diagram of CRT components](image)

**Figure 12-1A**
VGA Display

In standard VGA format, the screen contains 640x480 pixels
- 640 pixels in a row
- 480 rows

The standard refresh rate for a screen is ≈ 60 Hz
- The entire screen is refreshed 60 times per second

Many Still Images

- Video (and movies) are a series of stills
- If it goes fast enough
  - 50-60 Hz or more to not see flicker
- Your brain interprets as moving imagery
- Electron beam scans across
- Turned off when
  - Scanning back to the left (horizontal retrace)
  - Scanning to the top (vertical retrace)
Simple Scanning TV

VGA Timing

You supply two pulses, hsync and vsync, that let the monitor lock onto timing
One hsync per scan line
One vsync per frame
Figure 12-1B
Horizontal Synchronization

Display: 0..639, width = 640

Right border (front porch): 640..655, width = 16

Retrace (horizontal flyback): 656..751, width=96

Left border (back porch): 752..799, width=48
**Horizontal Timing**

- **BP**: 144
- **FP**: 784
- **SP**:

Pixel clock = 25 MHz  
Pixel time = 0.04 \( \mu s \)

Horizontal video = 640 pixels \( \times 0.04 \ \mu s = 25.60 \ \mu s \)

Back porch, BP = 16 pixels \( \times 0.04 \ \mu s = 0.64 \ \mu s \)

Front porch, FP = 16 pixels \( \times 0.04 \ \mu s = 0.64 \ \mu s \)

Sync pulse, SP = 128 pixels \( \times 0.04 \ \mu s = 5.12 \ \mu s \)

Horizontal Scan Lines = SP + BP + HV + FP

= 128 + 16 + 640 + 16
= 800 pixels \( \times 0.04 \ \mu s = 32 \ \mu s \)

1/60 Hz = 16.67 ms / 32 \( \mu s = 521 \) horizontal scan lines per frame
### Vertical Synchronization

**Vertical synchronization**

<table>
<thead>
<tr>
<th>Horizontal</th>
<th>Vertical</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td><strong>Duration</strong></td>
</tr>
<tr>
<td>Video Active</td>
<td>640</td>
</tr>
<tr>
<td>Front Porch</td>
<td>16</td>
</tr>
<tr>
<td>Sync Pulse</td>
<td>96</td>
</tr>
<tr>
<td>Back Porch</td>
<td>48</td>
</tr>
<tr>
<td><strong>Total:</strong></td>
<td><strong>800</strong></td>
</tr>
</tbody>
</table>

**Table 1: VGA Resolution Timing Parameters**

Vertical synchronization:
Display: 0..479, width = 480 lines
Bottom border (front porch): 480..489, width = 10
Retrace (vertical flyback): 490..491, width=2
Top border (back porch): 491..524, width=33

**Vertical Timing**

![Diagram of vertical timing](image)

Pixel clock = 25 MHz  
Horizontal scan time = 32 μs  
Vertical video = 480 lines x 32 μs = 15.360 ms  
Back porch, BP = 29 lines x 32 μs = 0.928 ms  
Front porch, FP = 10 lines x 32 μs = 0.320 ms  
Sync pulse, SP = 2 lines x 32 μs = 0.064 ms  
Vertical Scan Lines = SP + BP + VV + FP  
= 2 + 29 + 480 + 10  
= 521 lines x 32 μs = 16.672 ms  

1/50 Hz = 16.67 ms

![Diagram of horizontal scanning](image)
Pixel Rate:

p: the number of pixels in a horizontal scan line
p = 800 pixels/line
l: the number of horizontal lines in a screen
l = 525 lines/screen
s: the number of screens per second (refresh rate)
s = 60 screens/second

Pixel Rate = p x l x s = 25 Mpixels/second
It employs three colors (R = red, G = green, B = blue) per pixel, with the intensity of each color determined by an analog voltage in the 0V-to-0.7V range (indeed, for green, it can be either the regular 0V-to-0.7V range or 0.3 V higher; that is, 0.3V-to-1V when sync-on-green is used—explained later).

To generate the colors, six bits per color were initially employed, thus allowing a total of \( (2^6)^3 = 266,144 \) distinct colors or shades (though only 16 or 256 of them were made available in the original VGA palette).

Other, higher-resolution standards succeeded VGA, such as SVGA (super VGA), XGA (extended graphics array), SXGA (super extended graphics array), and more.

Such analog interfaces are collectively referred to as VGA modes, with the original 640 x 480 x 60Hz version still being the default mode for most analog computer monitors (that is the mode in which PCs operate at the beginning of the start-up sequence).

VGA monitors are now being replaced with LCDs (liquid crystal displays), which operate differently from CRTs. They are fully digital, and their standard interface is called DVI (digital visual interface).

Despite its digital behavior, in many cases the graphics controller also contains a VGA section, so VGA monitors can still be used. This option (analog + digital) is called DVI-I (integrated DVI), while the digital-only version is called DVI-D.
VGA Connector

VGA connector is a 15-pin connector called DB15 whose pinout is defined by the VESA (Video Electronics Standards Association) through a document called DDC (display data channel).

The functions of the pins of DB15 shown in Figure 12-3 are described in Table 12-1.
1) The table is separated into two parts, with the upper part containing the standard signals and the lower part containing the signals used for monitor identification.

2) In the upper part, three wires are employed for the colors (R, G, B), which are analog voltages between 0 V and 0.7 V on two parallel 75 ohm resistors (= 37.5 ohm). As already mentioned, G can be 0.3 V higher.

3) Still in the upper part, two wires are employed for the horizontal and vertical synchronization signals (Hsync, Vsync), which consist of 0V/5V digital waveforms.

4) The remaining six wires in the upper part of the table are either for ground or for an optional +5V supply voltage.

5) Finally, the lower part of the table shows the signals used for monitor identification. In modern monitors, the DDC standard is used (described later). Note that the pins’ functionalities change with the DDC version.

**Table 12-1**

<table>
<thead>
<tr>
<th>Pin</th>
<th>Signal</th>
<th>Direction</th>
<th>Simplest setup</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R (analog red, 0V-0.7V on 37.5Ω)</td>
<td>To monitor</td>
<td>Connected (analog)</td>
</tr>
<tr>
<td>2</td>
<td>G (analog green, 0V-0.7V or 0.3V-1V on 37.5Ω)</td>
<td>To monitor</td>
<td>Connected (analog)</td>
</tr>
<tr>
<td>3</td>
<td>B (analog blue, 0V-0.7V on 37.5Ω)</td>
<td>To monitor</td>
<td>Connected (analog)</td>
</tr>
<tr>
<td>5</td>
<td>GND (general and for +5V)</td>
<td>To monitor</td>
<td>GND</td>
</tr>
<tr>
<td>6</td>
<td>GND for R</td>
<td>To monitor</td>
<td>GND</td>
</tr>
<tr>
<td>7</td>
<td>GND for G</td>
<td>To monitor</td>
<td>GND</td>
</tr>
<tr>
<td>8</td>
<td>GND for B</td>
<td>To monitor</td>
<td>GND</td>
</tr>
<tr>
<td>9</td>
<td>No pin or +5V (optional)</td>
<td>To monitor</td>
<td>N/C</td>
</tr>
<tr>
<td>10</td>
<td>GND for HSync and VSync</td>
<td>To monitor</td>
<td>GND</td>
</tr>
<tr>
<td>13</td>
<td>Vsync (horizontal sync, 0V/5V waveform)</td>
<td>To monitor</td>
<td>Connected (digital)</td>
</tr>
<tr>
<td>14</td>
<td>Vsync (vertical sync, 0V/5V waveform)</td>
<td>To monitor</td>
<td>Connected (digital)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Old (non-DDC):</th>
<th>DDC1:</th>
<th>DDC2B (and E-DDC):</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 ID2 From monitor</td>
<td>(ID2 From monitor)</td>
<td>(ID2 From monitor) N/C</td>
</tr>
<tr>
<td>10 ID0 From monitor</td>
<td>(ID0 From monitor)</td>
<td>(ID0 From monitor) N/C</td>
</tr>
<tr>
<td>12 ID1 From monitor</td>
<td>Data From monitor</td>
<td>SDA Bidirectional N/C</td>
</tr>
<tr>
<td>15 ----- -----</td>
<td>(ID3 From monitor)</td>
<td>SCL To monitor N/C</td>
</tr>
</tbody>
</table>

**DDC and EDID**

The lower part of the Table 12-1 shows three alternatives for monitor identification (parameter passing), identified as non-DDC, DDC1, and DDC2B.

In the case of non-DDC (obsolete), the monitor is identified by means of three wires, called ID0, ID1, and ID2 (ID3 was added later).

Each of these wires has a pull-up resistor connected to VCC on the computer side, so the default logic value on these pins is '1'.
For a ‘0’ to occur, the display controller must provide a short circuit to GND. A typical monitor identification using ID2-ID1-ID0 (in this order) is the following:

"111" ! no monitor,
"101" ! monochrome monitor with resolution under 1024 x 768,
"110" ! color monitor with resolution under 1024 x 768, and
"010" ! color monitor with the 1024 x 768 resolution included.

DDC (display data channel) is a standard procedure for computer-monitor communication developed by VESA.

With the introduction of DDC, such communication was greatly improved because it allowed much more room for the display to tell its parameters to the graphics controller.

Such information is stored in a ROM on the monitor side, and obeys a standard format called EDID (extended display identification data), also defined by VESA. Up to 128 bytes can be stored in the ROM containing the manufacturer’s name, supported resolutions, and other information.

Such information is made available every time a pulse occurs in Vsync (in other words, Vsync acts as the memory-read clock). ID3 and the +5V supply were also included in DDC1, though monitor identification through ID pins became rapidly obsolete, with the EDID structure (accessed through pin 12) preferred instead.

DDC1 was rapidly superseded by DDC2B, which introduced an important change in the DDC channel—it determined that the EDID data should be transmitted using the I2C bus.

EDID is a standard data format for storing display-related information in a ROM at the monitor. Like DDC, EDID was also specified by VESA. Currently, the EDID ROM is accessed using the I2C bus.

**Circuit Diagram**

Figure 12-4 shows a block diagram for a VGA system, with the graphics controller on the left and the display controller on the right.

The VGA interface is the circuit used in the physical layer that interconnects these two controllers. The computer-side circuit can be divided into four sections: image generator, control generator, EDID interface, and finally the DACs (digital to analog converters).
The pixel signals are generated digitally (normally with 10 bits), then converted to analog by the three DACs. These analog voltages constitute the image that will be displayed by the VGA monitor.

Contrary to the pixel signals, the control signals have a fixed constitution (for a given VGA mode, of course). The whole sequence is controlled by pixel_clk, whose frequency, for the original (default) VGA, is 25.175 MHz.

Five control signals are generated (though in many cases not all are needed): Hsync (horizontal synchronism), Vsync (vertical synchronism), Hactive (portion of Hsync during which pixels are displayed), Vactive (portion of Vsync during which lines of pixels are displayed), and dena (display enable).

Hsync and Vsync are responsible for determining when a new line or new frame should start, with their timings also determining the VGA mode.

Hactive and Vactive represent the time intervals during which an image is actually being drawn on the screen.

Finally, dena is responsible for turning the pixel signals OFF during retrace (that is, while the electron beam returns to the beginning of a new line or of a new frame), so it can be obtained by simply ANDing Hactive and Vactive.

Note that only two of the five control signals are transmitted to the monitor.

**Control Signals**
As we have seen, the control signals are Hsync, Vsync, Hactive, Vactive, and dena, and their timings define the VGA mode. Several such modes are listed in Figure 12-5, where the first line corresponds to the original (default) VGA. Note that in this case the frequency of pixel_clk is 25.175 MHz.

Figure 12-5 also shows the waveforms for Hsync and Hactive (based on pixel_clk), which consist of four parts (all measured in number of pixels; i.e., number of clock cycles), called Hpulse (width of the horizontal synchronization pulse), HBP (horizontal back porch), Hactive (active line display interval), and HFP (horizontal front porch).

The vertical timing diagram is depicted in Figure 12-6, also consisting of four parts (all measured in number of lines or number of Hsync cycles), called Vpulse (width of the vertical synchronization pulse), VBP (vertical back porch), Vactive (active column display interval), and VFP (vertical front porch).

For example, in the 640 x 480 x 60Hz VGA option, the drawing of one line takes 800 clock cycles (Figure 12-5), while one frame requires a time equivalent to 525 lines (Figure 12-6).

Consequently, to be able to generate 60 frames per second, the clock frequency must be 800 x 525 x 60 = 25.175 MHz, which is the value listed in the figures (indeed, this clock frequency is for a refresh rate of 59.94 Hz, a value inherited from the NTSC television system).

The final control signal is dena, which must be low during the blanking intervals (retrace), so it can be easily obtained by ANDing Hactive and Vactive.

As already mentioned, some monitors support also a form of composite sync called sync-on-green, which consists of combining both horizontal and vertical synchronization pulses with the green signal, thus eliminating the need for the Hsync and Vsync wires.

In this case, the voltage range of G is made 0.3 V higher (0.3V-to-1V), so the downward pulses (toward 0 V) of Hsync and Vsync can be easily detected by the display controller, which can also distinguish Hsync from Vsync because the latter is much longer.
Pixel Signals

Figure 12.7 shows additional details regarding the DACs (pixel signals). R, G, and B are turned OFF when dena = '0'.

Note that pixel_clk is needed to control the data sequence that is applied to the triple DAC. Observe the presence of two new control signals, called nblank and nsync, which are specifically for the DACs.
The purpose of nsync is to cause G to operate in the regular (when low) or composite sync (when high) range.

The purpose of nblank is to blank the screen (when low) by bringing R, G, and B to their lowest levels: 0 mA (thus 0 V) for R and B and 0 mA (if operating in the regular range), or 8 mA (hence 0.3 V, if using composite sync) for G.

(a) Details regarding the DACs;

Figure 12-7

VHDL Design for VGA Interface

For the VHDL design of VGA interface, the structure in Figure 12-8 will be employed.
Based on figure 12.8a, such circuits will be divided into two parts: control generator, which is always the same (for a given VGA mode), hence application-independent, and image generator, responsible for generating the images used in the examples, hence application-dependent.

The generation/processing of images can be done in a number of ways.

For example, images can be built by the VHDL code itself (local hardware, proper only for very simple, geometric images), or can be retrieved from some kind of memory (SRAM, file, etc.), or can be real-time images produced by a video camera.
The main purpose of this first design is to illustrate how the control signals can be constructed.

The image consists of just four horizontal stripes of solid colors, with widths 1, 2, 3, and 474 pixels. The first three are red, green, and blue, respectively, while the last one (wide) is determined by three toggle switches, hence allowing eight colors.

red\_switch = '0' \rightarrow no red (R = "0000000000" \rightarrow 0V).
red\_switch = '1' \rightarrow maximum red intensity (R = "1111111111" \rightarrow 0.7V).
green\_switch = '0' \rightarrow no green (G = "0000000000" \rightarrow 0V).
green\_switch = '1' \rightarrow maximum green intensity (G = "1111111111" \rightarrow 0.7V).
blue\_switch = '0' \rightarrow no blue (B = "0000000000" \rightarrow 0V).
blue_switch = '1' → maximum blue intensity (B = "1111111111" → 0.7V).

A 50 MHz clock and a triple 10-bit DAC will be used. The DAC’s control signals, nblank and nsync, must be kept at '1' and '0', respectively.

Part 1 (lines 30–77) implements the control generator (figure 12.8a). Note that the code for each signal obeys the timing diagrams of figure 15.2b.

PS 12-1

A VHDL code for this VGA interface is presented above. The control signal parameters of figure 12.8b were entered using GENERIC declarations (lines 7–14), so the code can be easily adjusted to other VGA modes. The signal names (lines 16–21) are from figure 12.8a. Note that because only two (Hsync, Vsync) of the five control signals are transmitted to the monitor (figure 12.8a), the other three were declared internally (line 25).
BEGIN
--Part 1: CONTROL GENERATOR

--Static signals for DACs:
  nblank <= '1'; --no direct blanking
  nsync <= '0'; --no sync on green

--Create pixel clock (50MHz->25MHz):
  PROCESS (clk)
  BEGIN
    IF (clk'EVENT AND clk='1') THEN
      pixel_clk <= NOT pixel_clk;
    END IF;
  END PROCESS;

--Horizontal signals generation:
  PROCESS (pixel_clk)
  BEGIN
    VARIABLE Hcount: INTEGER RANGE 0 TO Hd;
    IF (pixel_clk'EVENT AND pixel_clk='1') THEN
      Hcount := Hcount + 1;
      IF (Hcount=Ha) THEN
        Hsync <= '1';
      ELSIF (Hcount=Hb) THEN
        Hactive <= '1';
      ELSIF (Hcount=Hc) THEN
        Hactive <= '0';
      ELSIF (Hcount=Hd) THEN
        Hsync <= '0';
      Hcount := 0;
    END IF;
  END IF;
  END PROCESS;

--Vertical signals generation:
  PROCESS (Hsync)
  BEGIN
    VARIABLE Vcount: INTEGER RANGE 0 TO Vd;
    IF (Hsync'EVENT AND Hsync='0') THEN
      Vcount := Vcount + 1;
      IF (Vcount=Va) THEN
        Vsync <= '1';
      ELSIF (Vcount=Vb) THEN
        Vactive <= '1';
      ELSIF (Vcount=Vc) THEN
        Vactive <= '0';
      ELSIF (Vcount=Vd) THEN
        Vsync <= '0';
      Vcount := 0;
    END IF;
  END IF;
  END PROCESS;

--Display enable generation:
  dena <= Hactive AND Vactive;

PS 12-2
Part 2 of the code (lines 81–115) implements the image generator. In lines 85–91, a counter is used to construct a pointer (called line_counter) to the image rows. If it points to row 1 (lines 93–96), the color is red. If it points to rows 2–3 (lines 97–100), the color is green. When pointing to rows 4–6 (lines 101–104), it is blue. Finally, when pointing to rows 7–480 (lines 105–108), the color is determined by the three toggle switches. In lines 110–113, dena = ‘0’ is used to turn the image OFF during retrace.

This design could obviously have been done using a structural approach, with each of these parts implemented using the COMPONENT construct.

```
78  -------------------------------
79  --Part 2: IMAGE GENERATOR
80  -------------------------------
81  PROCESS (Hsync, Vsync, Vactive, dena, red_switch,
82      green_switch, blue_switch)
83    VARIABLE line_counter: INTEGER RANGE 0 TO Vc;
84  BEGIN
85  IF (Vsync='0') THEN
86    line_counter := 0;
87  ELSIF (Hsync'EVENT AND Hsync='1') THEN
88    IF (Vactive='1') THEN
89      line_counter := line_counter + 1;
90    END IF;
91  END IF;
92  IF (dena='1') THEN
93    IF (line_counter=1) THEN
94      R <= (OTHERS => '1');
95      G <= (OTHERS => '0');
96      B <= (OTHERS => '0');
97    ELSIF (line_counter>1 AND line_counter<=3) THEN
98      R <= (OTHERS => '0');
99      G <= (OTHERS => '1');
100     B <= (OTHERS => '0');
101    ELSIF (line_counter>3 AND line_counter<=6) THEN
102      R <= (OTHERS => '0');
103     G <= (OTHERS => '0');
104     B <= (OTHERS => '1');
105    ELSE
106      R <= (OTHERS => red_switch);
107     G <= (OTHERS => green_switch);
108     B <= (OTHERS => blue_switch);
109    END IF;
110  ELSE
111    R <= (OTHERS => '0');
112    G <= (OTHERS => '0');
113    B <= (OTHERS => '0');
114  END IF;
115  END PROCESS;
116  END vga;
```