Exercise 5: VGA Controller

G. Kemnitz*, TU Clausthal, Institute of Computer Science

May 29, 2012

Abstract

A VGA controller circuit, that displays examples of graphic elements, is given and should be tested. This controller should be afterward extended by other examples of graphic elements, mainly a pixel graphic that can be moved on the screen by buttons.

1 Function

The exercise board has also a VGA connector to plug in a monitor cable. The monitor interface has been designed for a classic monitor with a picture tube. In a picture tube an electron beam draws the image line-by-line. From the three cathodes (one for red, one for green and one for blue) controllable quantities of electrons are emitted and accelerated to the projection screen by an anode voltage of approximately 20,000 V. On its way to the anode electromagnets deflect the electron beam in x and y direction. At the point of contact the electrons produce light spots (figure 1). The intensity of the three color components are controlled by the cathode currents.

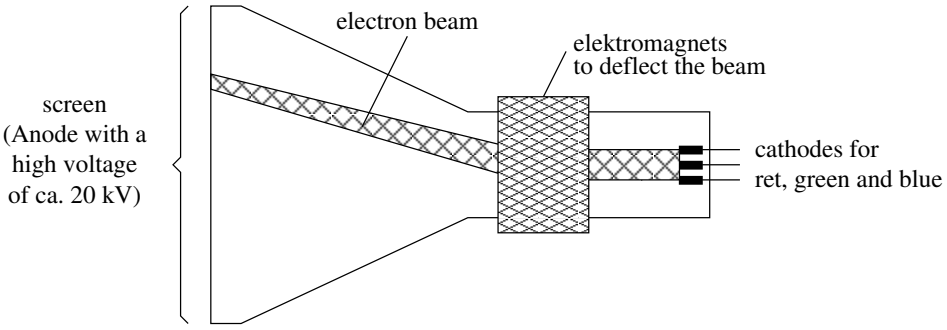

Figure 1: Picture tube

In VGA mode the beam writes 480 lines of 640 pixels. During return time between two lines the beam has to be blanked, i.e. the cathode currents have to be set to zero (figure 2). Between two picture during return of the beam from bottom to top of the image the beam is also blanked.

The VGA controller has to send the following signals to the monitor (see figure 3):

- the three RGB signals with voltages between 0 and 1 V to control the cathode currents
- a line or horizontal and an image or vertical synchronization signal.

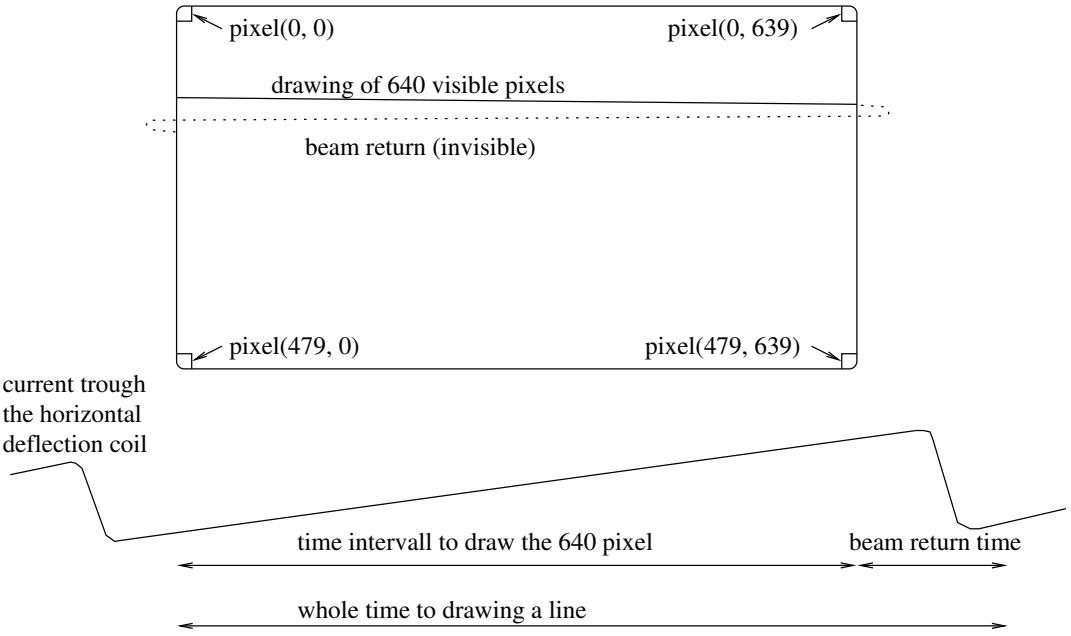

Figure 2: Image drawing by an electron beam

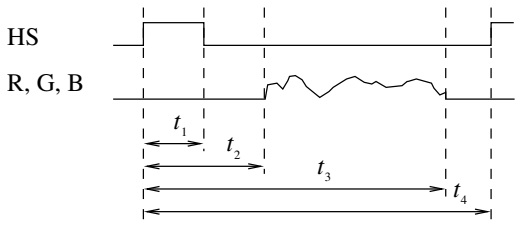

Figure 3: Time frame to draw one line

Figure 2 shows the time frame to draw one line. It starts with the horizontal synchronization pulse with the length t_1 followed by an extended blank time, the time to draw the visible line and some blank time. The following tabular shows the duration values t_1 to t_4 for an image resolution of 640×480 pixel and 60 images per second in micro second an clock periods of a 25 MHz clock:

	duration	number of clocks ($f_{Pixel} = 25\,\mathrm{MHz}$)
t_1	3,84µs	96
t_2	5,76µs	144
t_3	31,36µs	784
t_4	32µs	800

The vertical control works similar. At the begin of each image a vertical synchronization impulse has to be generated followed by a few dark lines. Then follow 480 visible and a few invisible lines (figure 4).

*Tel. 05323/727116

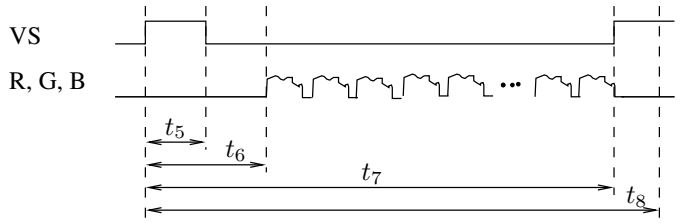

Figure 4: VGA control signals for one picture

	duration	number of lines ($f_{Zeile} = \frac{1}{32\mu s} = 31,25\text{kHz}$)
t_5	64µs	2
t_6	992µs	31
t_7	16,352ms	511
t_8	16,7ms	521

A VGA controller circuit needs at least a clock input, two synchronization outputs, and outputs to control the intensity and color of the pixels to be written.

```
entity a5_vga is
  port(
    clk : in std_logic;
    hSync : out std_logic; -- horizontal syncronization impuls (line
    vSync : out std_logic; -- vertical ysncronization impuls (picture)
    rgb : out std_logic_vector(2 downto 0) -- RGB signal
  );
end entity;
```

Our experimental board produces only one bit per color channel. That allows only to display 8 different colors that should be declared as constants:

```
constant schwarz: std_logic_vector(2 downto 0) := "000";
constant rot:     std_logic_vector(2 downto 0) := "100";
constant gruen:   std_logic_vector(2 downto 0) := "010";
constant blau:    std_logic_vector(2 downto 0) := "001";
constant gelb:    std_logic_vector(2 downto 0) := "110";
constant cyan:    std_logic_vector(2 downto 0) := "011";
constant violett: std_logic_vector(2 downto 0) := "101";
constant weis:    std_logic_vector(2 downto 0) := "111";
```

The control circuit itself needs a pixel counter and a line counter. The pixel counter counts the pixels with a 25 MHz clock. The horizontal synchronization impulse should be turned to one when resetting the pixel counter and set back to zero when the counter reaches the value 95. The blank signal for line return is set to zero at counter value 143 and returns to one at counter value 783. Counter value 799 resets the counter. The line counter increases its value by one during each line return and controls in a similar way the corresponding synchronization and blank signal.

Image elements may only be produced in the visible pixel range, i.e. only if both blank signals are zero. To create lines, pixel and line counter are compared with given values. A stored image is displayed by addressing an image memory with addresses calculated from the pixel and line counter values and controlling the RGB signals by the memory content. In the circuit description a5_vga.vhd are some examples.

Exercises for preparation

1. What kind of drawing elements are displayed by the VGA controller in the file a5_vga.vhd?

2. What causes the statement "vsl_vCt := to_unsigned(vCounter, 10);"?

3. Why the image in the small memory array is displayed mirrored? Is it mirrored horizontally, vertically or at a screen diagonal?

4. What is the size of the stored image and what the size of the displayed image (number of rows and columns)?

2 Test of the given VGA control circuit

Download the files a5_vga.ucf, a5_vga.vhd and a5_vga.xise from the web in the working directory of the project. Open it in ISE. Translate and program it in the FPGA. To Test it switch the monitor to input signal 1 (VGA).

3 Design exercises

1. Change the given circuit in that way that it displays in addition a black grid on white background. Distance between grid lines 10 pixel.

2. Add to the grid a cursor image in the middle of the screen. This image, e.g. a circle, should be as the bit map picture in the give description declared a bit array constant, e.g. the initialization of a read only memory.

3. Next the circuit should be extended by functions to control the cursor position via buttons:

   - BTN0: move to right
   - BTN1: move to left
   - BTN2: move down
   - BTN3: move up

   During pressing a button the cursor should move approximately by a speed of 4 pixel per second. If a button for horizontal and a button for vertical movement are pressed simultaneously, the cursor should move on a diagonal. It should stop when touching the visible border.

4 Check list for the compliance test

- working VGA controller circuit