FH University of Applied Sciences TECHNIKUM WIEN

Chip Design 2

Project - VGA Controller

Version: 1.0
Author: R. Höller

Copyright Notice

The VGA Controller Project

In this project, a simple VGA (Video Graphics Array) controller shall be implemented using an FPGA. The VGA controller should be able to display images with a resolution of 640x480 pixels. Furthermore, it should be possible to select between three different images, depending on the position of the two switches SW0 and SW1. One of the images is read from the internal block memory. Finally, depending on the position of another switch (SW2), a so called “movable object” should be displayed, whose position can be controlled with the four push buttons BTNU, BTND, BTNL and BTNR. This document specifies the project requirements and gives useful information and hints in order to successfully implement the VGA controller design.

Block Diagram

Figure 1 shows a block diagram of the VGA Controller project. The design will be implemented on a Xilinx Artix-7 FPGA contained on the Basys3 development board from Digilent, see [1]. The board is connected to a monitor over a VGA interface, utilizing the two digital signals h_sync and v_sync and the three analog signals red, green and blue.

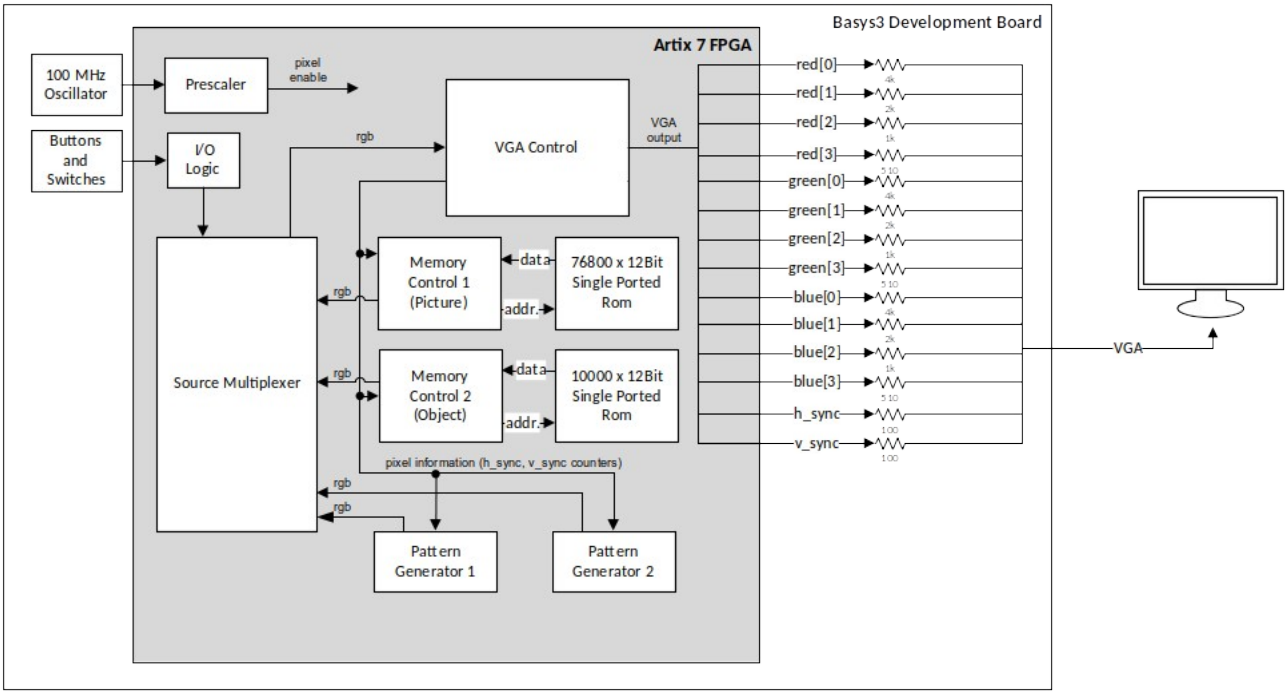

Figure 1 – Block diagram of the VGA controller project

The VGA controller consists of the following units:

- A “Prescaler” which generates a “pixel enable signal” that is logic high every 4th clock cycle (the clock signal coming from the board’s clock oscillator has a frequency of 100 MHz). The pixel enable signal is used by all sub-units of the VGA controller in order to process all actions in the VGA controller with an update rate of 25 MHz (this fits to the VGA pixel clock of 25 MHz). Note, that you have to use this signal as an enable signal, not as a clock signal!
- The “VGA control” unit drives the VGA signals. It generates the vertical and horizontal sync signals v_sync and h_sync, according to the timing defined in Table 3, Table 4 and Table 5.
- The “source multiplexer” routes the RGB signals to the “VGA control” unit according to the position of the switches SW0 and SW1.
- The main purpose of the “I/O logic” is debouncing of the push buttons. Furthermore, this unit handles the three switches SW0, SW1 and SW2.
- The “memory control” units 1 and 2 generate addresses for the two ROMs according to the sync counters.
- The “pattern generator” units 1 and 2 generate two different image patterns. Their timing is also controlled by the sync counters.

Functionality of the VGA Controller

The VGA controller shall adhere to the following requirements:

- Memory Control Unit 1 reads a 320 x 240 x 12 bit image from the 76800x12 bit single-port ROM. This image is displayed four times per frame to accommodate the desired resolution of 640x480 pixels (Figure 2).

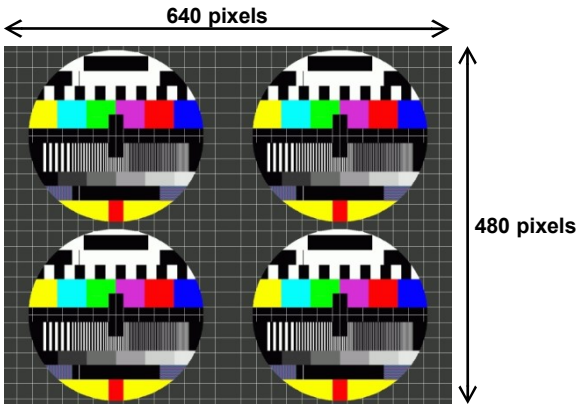

Figure 2 – Image read from 76800x12 bit ROM displayed 4 times

- Memory Control Unit 2 reads a 100 x 100 x 12 bit image from the 10000x12 bit single-port ROM. This image represents a movable object, which can be controlled by the buttons BTNU, BTND, BTNL and BTNR. According to the position of the object and the position of SW2, this image is displayed once per frame (Figure 3). Note, that the movable object should hover over the image generated by Pattern Generator 1 or 2 or from the 76800x12 bit ROM.

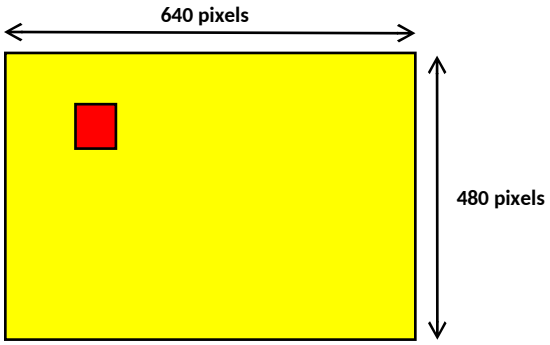

Figure 3 – Movable Object

- Pattern Generator 1 generates a pattern of equally distributed horizontal color stripes that are repeated four times per frame. Each stripe consists of a red, green, blue and black segment. The resulting pattern is depicted in Figure 4.

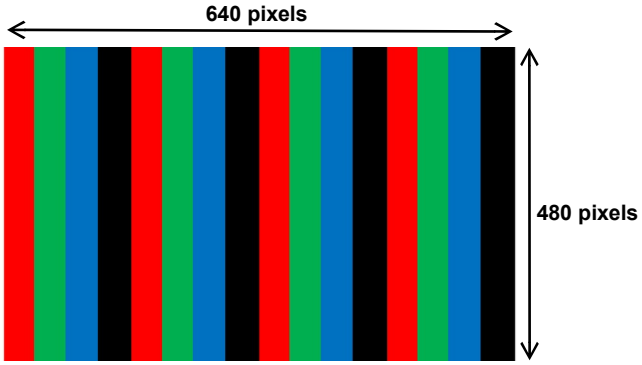

Figure 4 – Pattern One

- Pattern Generator 2 generates a tile pattern with 10 x 10 tiles per frame. The color pattern of the tiles (red, green and blue) is shown in Figure 5.

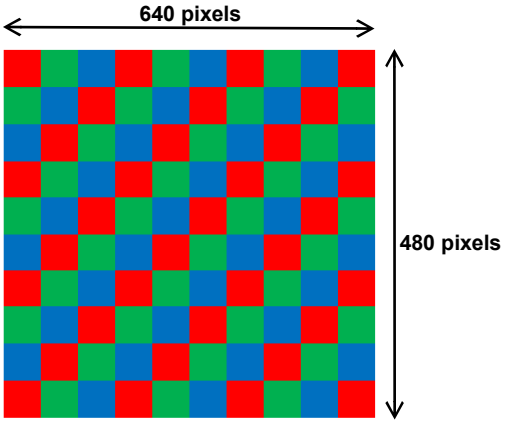

Figure 5 – Pattern Two

- The switches SW0 and SW1 of the Basys3 development board determine which image is currently displayed, see Table 1.

SW0 – Position	SW1 – Position	Selected Image
1	X	Image from Memory Control 1
0	1	Image from Pattern Generator 2
0	0	Image from Pattern Generator 1

Table 1 – Selection of Image

- SW2 of the BASYS3 development board shall determine if a movable object is displayed. If SW2 is logic 1, the object shall be displayed (Figure 3). If SW2 is logic 0, the object shall not be displayed.
- BTNC is used as a global asynchronous reset signal for the FPGA design.
- The remaining four buttons BTNU, BTND, BTNL and BTNR are used to control the position of the movable object (Figure 3). Once a button is pressed, the object shall move by 30 pixels according to Table 2.

Button Pressed	Result
BTNU	Move Object 30 pixels up
BTND	Move Object 30 pixels down
BTNL	Move Object 30 pixels left
BTNR	Move Object 30 pixels right

Table 2 – Controlling the Position of the Moveable Object

VGA Basics

Historically, the VGA (Video Graphics Array) interface was developed when so called CRT (Cathode Ray Tube) monitors were state-of-the-art which today have been replaced completely by LCD technology. However, the original VGA standard still works for LCD monitors.

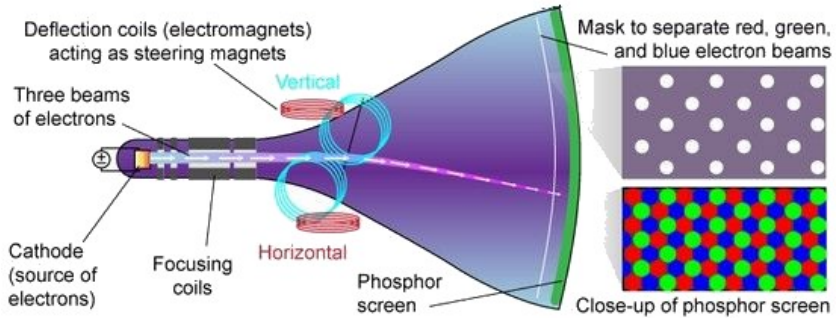

Figure 6 – CRT-based Color Monitor

CRT-based VGA monitors use three moving electron beams (one for red, one for blue, and one for green), generated by a cathode, to energize the phosphor that coats the inner side of the screen (see Figure 6). The phosphor surface glows brightly at the impact point of the electron beams, and it continues to glow for several hundred microseconds after the beam is removed. Between the cathode and the display surface, the beam passes through the neck of the CRT where two coils of wire produce orthogonal electromagnetic fields. Because cathode rays are composed of charged particles (electrons), they can be deflected by these magnetic fields. Current waveforms are passed through the coils to produce magnetic fields that interact with the cathode rays and cause them to transverse the display surface in a “raster” pattern, horizontally from left to right (left hand side of Figure 7) and vertically from top to bottom (right hand side of Figure 7). As the cathode ray moves over the surface of the display, the current sent to the electron guns can be increased or decreased to change the brightness and color of the display at the cathode ray impact point. Information is only displayed when the beam is moving in the "forward" direction (left to right and top to bottom), and not during the time the beam is reset back to the left or top edge of the display (red dotted arrows shown in Figure 7). Thus, the electron beams are switched off during the time “H Frontporch”, “H Sync” and “H Backproch” as well as during “V Frontporch”, “V Sync” and “V Backporch”.

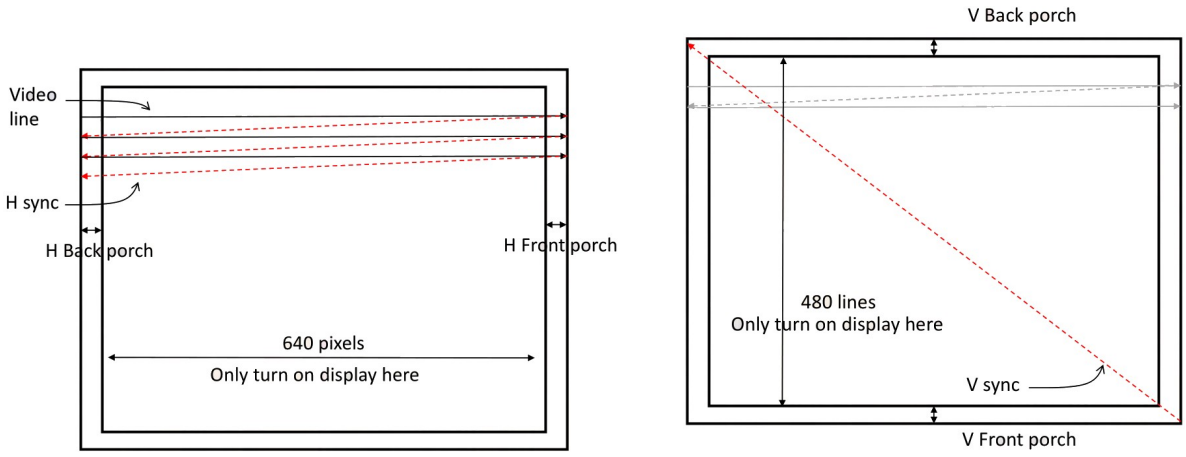

Figure 7 – Electron Beams Movement

VGA Interface Timing

The VGA interface consists of the following signals:

- **RED/GREEN/BLUE** Color information of a specific pixel (analog)
- **H_SYNC** Horizontal sync signal (digital)
- **V_SYNC** Vertical sync signal (digital)

Figure 8 shows the horizontal timing of the VGA signals. The analog RGB signals represent the color composition of each pixel. Each of the three color components (red, green and blue) is represented by four bits in the FPGA. Each of the three digital 4-bit signals is converted to an analog signal by a simple voltage divider contained on the Basys3 board, see [2].

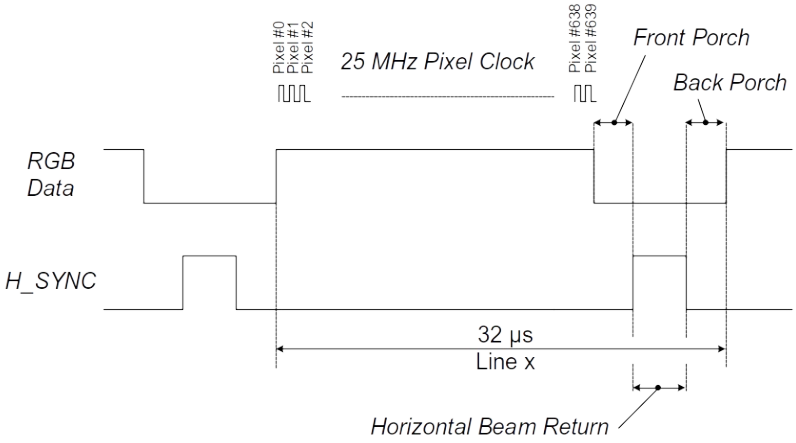

Figure 8 – Horizontal VGA Timing

The pixels are serially transmitted with the pixel clock (25 MHz) on the RGB signals, starting with pixel 0 in line 0, followed by pixel 1 in line 0, pixel 2 in line 0, … pixel 638 in line 0 and finally

pixel 639 in line 0. The sync pulse (logic 1) on the H_SYNC line signal indicates the beginning of a new line. The RGB signals have to stay low during the time “front porch”, “sync pulse” and “back porch” (“blanking period”)! Afterwards, the pixels of the next line are transmitted, starting with pixel 0 in line 1, pixel 1 in line 1, pixel 2 in line 1, … pixel 638 in line 1 and finally pixel 639 in line 1. Another sync pulse on H_SYNC line signal indicates the beginning of the next line (line 2) … and so on.

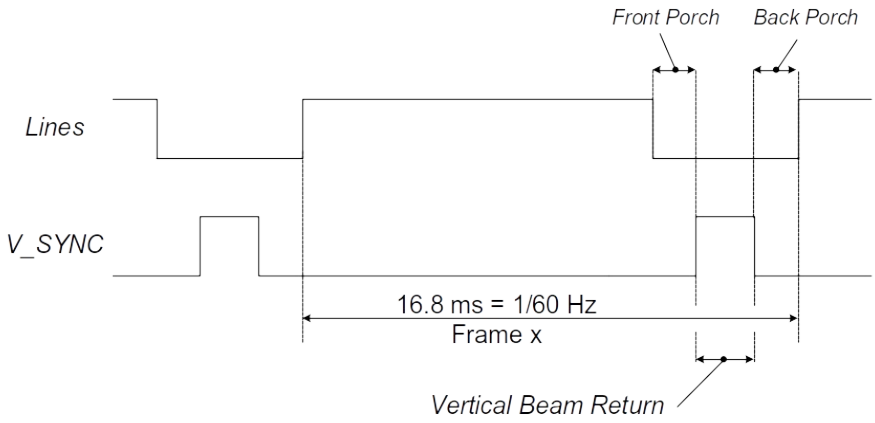

Figure 9 – Vertical VGA Timing

The sync pulse (logic 1) on the V_SYNC line indicates the start of a new frame (Figure 9). Again, the RGB signals must stay low during the time “front porch”, “sync pulse” and “back porch” (“blanking period”)! However, the horizontal sync pulses must be generated even during the vertical front porch, back porch and the duration of the vertical sync pulse, according to Figure 8. After the vertical “back porch”, the pixels of the next frame are transmitted on the RGB lines starting with pixel 0 in line 0.

The following tables specify the timing for a 640x480 VGA signal with a refresh rate of 60 Hz. Note, that the exact frequency of the pixel clock is 25.175 MHz but we want to use 25.000 MHz since this simplifies some things related to simulation.

Screen refresh rate	60 Hz
Pixel clock frequency	25 MHz

Table 3 – General VGA Timing

The polarity of the horizontal sync pulse is **positive**.

Scanline Part	Pixels	Time [us]
Visible area	640	25.6
Horizontal front porch	16	0.64
h-Sync pulse	96	3.84
Horizontal back porch	48	1.92
Whole line	800	32

Table 4 – Horizontal Timing

The polarity of the vertical sync pulse is **positive**.

Frame Part	Lines	Time [ms]
Visible area	480	15.36
Vertical front porch	10	0.32
v-Sync pulse	2	0.064
Vertical back porch	33	1.056
Whole frame	525	16.8

Table 5 – Vertical Timing

Preparation of the ROM Content

Data for the image that is displayed when SW0 = 1 as well as for the movable object is stored in two on-chip ROMs, see Figure 1. The distance learning letter “Getting Started with Xilinx IP Catalog” describes how to generate these ROMs, see [3]. There will be also a lab exercise on this topic during the attendance phase of the course.

Generally, the content of the ROMs is defined via COE (Coefficient) files. You can download the ZIP file “images.zip” from the CIS (Campus Information System) webpage of this lecture which includes six COE files of already prepared images (the images are also included in JPG format if you want to have a look at them). The distance learning letter [3] describes how to initialize the ROMs using the COE files. **Important Note:** Everytime you change the COE file for one of the ROMs you have to close Vivado and reopen the Vivado project of the VGA controller after the COE file was modified! Otherwise, Vivado will not recognize that the ROM has to be re-synthesized, see also Section “Re-Customizing Generated Blocks and Memory Content” in [3].

Table 6 and Table 7 show the memory map of the ROMs.

ROM Address	Line	Pixel	Data[11:8]	Data[7:4]	Data[3:0]
0	0	0	Red	Green	Blue
1	0	1	Red	Green	Blue
2	0	2	Red	Green	Blue
:	:	:	:	:	:
318	0	318	Red	Green	Blue
319	0	319	Red	Green	Blue
320	1	0	Red	Green	Blue
321	1	1	Red	Green	Blue
:	:	:	:	:	:
76798	239	318	Red	Green	Blue
76799	239	319	Red	Green	Blue

Table 6 – Memory Map of 76800x12 bit ROM

ROM Address	Line	Pixel	Data[11:8]	Data[7:4]	Data[3:0]
0	0	0	Red	Green	Blue
1	0	1	Red	Green	Blue
2	0	2	Red	Green	Blue
:	:	:	:	:	:
98	0	98	Red	Green	Blue
99	0	99	Red	Green	Blue
100	1	0	Red	Green	Blue
101	1	1	Red	Green	Blue
:	:	:	:	:	:
9998	99	98	Red	Green	Blue
9999	99	99	Red	Green	Blue

Table 7 – Memory Map of 10000x12 bit ROM

Useful Hints

Please do not underestimate the complexity of this project! It is highly recommended that you work on the project on a regular basis. Make use of the attendance phase of the course to ask questions concerning the project and its realization.

Try to partition the entire project in smaller sub-blocks that can be implemented and tested more easily. For example, you could start with the implementation of “Pattern Generator 1”. Once the pattern generator was tested successfully, continue to add additional functionality, e.g., “Pattern Generator 2” and so on.

Check the correct behavior of your design in a simulation! For example, simulate the prescaler if it provides the correct pixel enable signal! Also, check the VGA timing for the values shown in Table 3, Table 4 and Table 5 since the timing of the h_sync and v_sync signals is very sensitive! Make use of the “VGA Monitor Simulation Model” that can be downloaded from the CIS website of this lecture!

Project Submission

Please prepare the following things for the presentation of your project in the lab:

- Demonstration of the functionality of your VGA controller on the FPGA board
  - The design must implement 100% of the specified functionality. Otherwise, you will obtain zero points!
- Commented and compilable VHDL testbenches for your VGA controller’s FPGA design:
  - If you cannot present a simulation at least at the top-level of your design, you will obtain zero points since this (i) reflects a poor design style and (ii) indicates that you have not really developed the design by yourself!
- Commented and synthesizable VHDL code of your VGA controller’s FPGA design:
  - If your design is not compilable/synthesizable, you will obtain zero points.
  - Note that the number of points that can be obtained depends heavily on the quality of your code and project structure, e.g., the application of a clean and structured coding style, naming conventions for signals and files, quality of testbenches and simulation setup, number of warnings reported by Xilinx Vivado as well as the quality of comments in the source code (you will lose points for both undercommented and overcommented code!).
  - It is also important that your source code adheres to synchronous design guidelines.
  - Finally, the grading of your project depends on how well you can answer the questions of the instructor during the presentation. It is assumed that every student understands 100% of the source code and the design flow, even If the project was developed by a group of students (this changes from year to year).

Please upload a ZIP file of your project (which should contain VHDL files only!) after the presentation via the CIS website of this course. Moreover, return the Basys3 board to the course supervisors. Note, that you cannot finish the course before you did not upload the design and return the board!

Abbreviations

COE	Coefficient (a data format used by Xilinx)
CRT	Cathode Ray Tube
FPGA	Field Programmable Gate Array
RGB	Red Green Blue
VGA	Video Graphics Array
VHDL	Very High Speed Integrated Circuit Hardware Description Language

References

[1] Basys 3 FPGA Board Reference Manual, Document No. 502-183, April 8, 2016, Digilent Inc.

[2] Basys 3 FPGA Board Schematics, Document No. 500-183, 5/21/2014, Revision C.0, Digilent Inc.

[3] P. Rössler, R. Höller: Getting Started with Xilinx IP Catalog, Lecture Note, University of Applied Sciences Technikum Wien, 2/2017

Version

Version 1.0	Update to entire document for CHIP2

If you find errors or inconsistencies, please report them to the supervisors of this course. Thank you!