Proc. Annual Symposium on Research & Industrial Training, 02(2015) 171-177
Department of Electronics – Wayamba University of Sri Lanka

HARDWARE IMPLEMENTATION OF VGA CONTROLLER ON FPGA

E.M.S.L.B. Ekanayake*, W.A.S. Wijesinghe

Department of Electronics, Wayamba University of Sri Lanka, Kuliyapitiya, Sri Lanka

*Sachinda88@gmail.com**

ABSTRACT

As a standard display interface Video Graphics Array has been widely used. This paper presents the design and implementation of VGA controlleron Field Programmable Gate Array (FPGA). Most VGA controllers are used software drivers. However, FPGA are increasingly used in hardware acceleration and video processing, it is required to directly display video output on VGA. Therefore, it is required to implement VGS controller in FPGA hardware. We used Verilog hardware description language to design the VGA controller and tested it with random walk algorithm.

Keywords: VGA Controller, FPGA, Verilog, Xilinx ISE

1. INTRODUCTION

Field-Programmable Gate Arrays (FPGAs) are digital integrated circuits (ICs) that contain configurable blocks of logic along with configurable interconnects between these blocks.[1] Specifically, an FPGA contains programmable logic components called logic elements (LEs) and a hierarchy of reconfigurable interconnects that allow the LEs to be physically connected. LEs can be configured to perform complex combinational functions, or merely simple logic gates like AND and XOR. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory.[2]

VGA (video graphics array) is a video display standard. It provides a simple method to connect a system with a monitor for showing information or images. As a standard display interface, VGA has been widely used. There is more and more need in displaying the result of the process in real time as the fast development of embedded system.[2]

Diffusion-limited aggregation (DLA) is the process of cluster growth by particles undergoing a random walk due to Brownian motion. The theory of DLA, proposed by Witten and Sander in 1981, is useful in explaining the aggregation of particles in any system where diffusion is the primary means of transport. Several methods are available to accomplish this. One approach is to place particles in a lattice of any desired geometry and to simulate their aggregation due to sticking together while they perform a random walk. The project mainly focused on developing random walker for implementation of DLA. VGA interfaced is also developed for the showing of position of the random walker.

VGA Controller module and random walker position generator module are written in Verilog HDL using Xilinx ISE. Once the simulation is succeeded, the program will be burnt into Xilinx Spartan 3E, which will process the VGA Controller module and random walker module and display the image on LCD screen.

1.1 VGA principal

The monitor screen for a standard VGA format contains 640 columns by 480 rows of picture elements called pixel. An image is displayed on the screen by turning on and off individually pixels. Turning on one pixel does not represent much, but combining numerous pixels generates an image. The monitor continuously scans through the entire screen, rapidly turning individual pixels on and off. Although pixels are turned on one at a time, we get the impression that all the pixels are on because the monitor scans so quickly. This is why old monitors with slow scan rates flicker.

Referred to Figure 1, the scanning process starts from row 0, column 0 in the top left corner of the screen and moves to the right until it reaches the last column. When the scan reaches the end of a row, it retraces to the beginning of the next row. When it reaches the last pixel in the bottom right corner of the screen, it retraces back to the top-left corner and repeats the scanning process. In order to reduce flicker on the screen, the entire screen must be scanned 60 times per second. This period is called the refresh rate. The human eye can detect flicker at refresh rates less than 30 Hz. To reduce flicker from interference from fluorescent lighting sources, refresh rates higher than 60 Hz are sometimes used in PC monitors. During the horizontal and the vertical retraces, all the pixels are turned off [3].

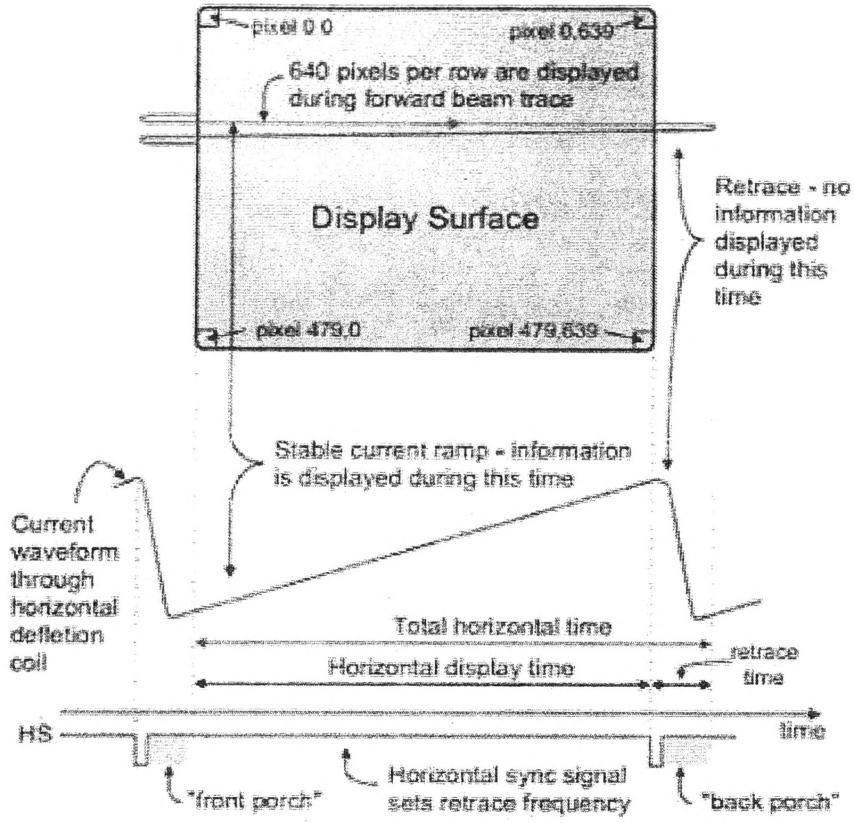

Figure 1: Scanning pattern of VGA controller

1.2 VGA interface signal

The VGA monitor is controlled by 5 signals: red, green, blue, horizontal synchronization, and vertical synchronization. The three color signals, collectively referred to as the RGB signal, control the color of a pixel at a given location on the screen. They are analog signals with voltages ranging from 0.7 to 1.0 volt. Different color intensities are obtained by varying the voltage. For simplicity, these three-color signals are treated as digital signals, so we can just turn each one on or off.[3]

1.3 Timing Control

To obtain the 640 × 480 screen resolution, a clock with a 25.175 MHz frequency is used. A higher clock frequency is needed for a higher screen resolution. For the 25.175 MHz clock, the period is as below.

$$\frac{1}{25.175MHz} = 0.0397us \text{ per clock cycle}$$

1.4 Random Walker

The random walker is generated using module called Linear Feedback Shift Register Module LFSRM that uses XOR gate logic operation and data shifting both for generate random walker position. And the random walker address is 15bit value.

2. EXPERIMENTAL METHODOLOGY

In figure 2 the function of "clock generator" block is to reduce the frequency of input clock from 50 MHz to 25 MHz. Meanwhile, "vga_sync" block is used to generate timing and synchronization signals. The "h_count" and "v_count" indicate the relative positions of the scans and essentially specify the location of the current pixel while the "h_sync" signal specifies the required time to scan a row, and the "v_sync" signal specifies the required time to scan the entire screen. "vga_sync" block also generates the "video_on" signal which indicates whether to enable or disable the display. Besides that, "address generator" block is used to generate address for the "img_data" block by using the "h_sync" and "v_sync" signal. "img_data" block will get the index data (q) from the UCF file according to the address generated. Note that the index data are connected to the "img_index" block to use as the address. The "img_index" block will get the RGB data (q) from UCF file according to the address generated (index data). The RGB data consist of 24-bits, whereas "q [23:16] ", "q [15:8]" and "q [7:0]" indicate the "R_data", "G_data" and "B_data" respectively. [2]

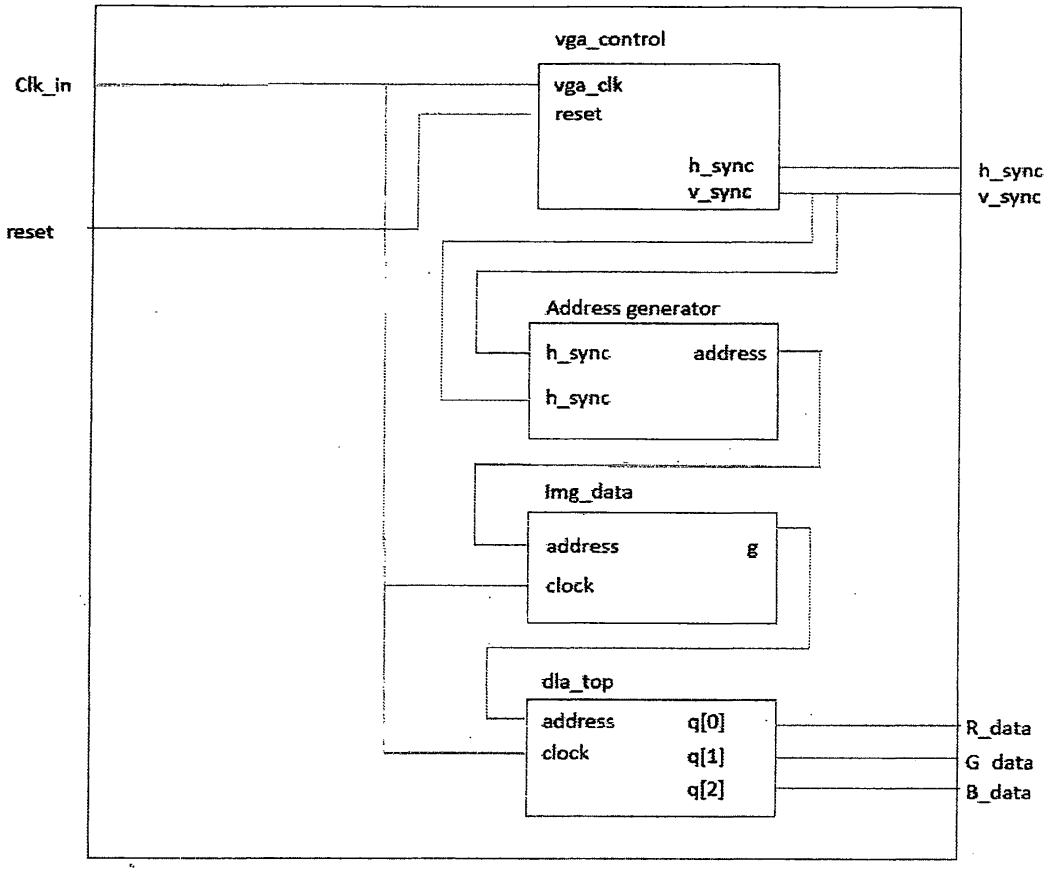

Figure 2: Block diagram of VGA controller

2.1 Design flow of VGA Synchronization signal

First and foremost, reset is sensed. If reset is equal to 1, "h_count" and "v_count" will be reset to 0. If reset is equal to 0, it will check whether the value of "h_count" is equal to 799 or not. If the value of "h_count" is not equal to 799, it will be increased by 1.Meanwhile, if the value of "h_count" is equal to 799, it will be reset to 0. This is due to one complete horizontalscan is start from 0 to 799. Then, it will check whether the value of "v_count" is equal to 524 or not. If the value of "v_count" is not equal to 524, it will be increased by 1. If the value of "h_count" is equal to 799, it will be reset to 0. This is due to one complete vertical scan is start from 0 to 524.

2.2 Random walker

Every time the top module accesses the Linear Feedback Shift Register Module(LFRM) module to get random walker position. The LFRM module generates the 15bits of random walker address and fed to the top module. After processing the data in top module, the top module fed that data into VGA interface. The process is done by real time therefore no memory involved.

3. RESULTS AND DISCUSSION

In this paper we implemented a VGA controller directly on FPGA hardware. To test the system a random walker is generated displayed on the VGA monitor. To display the random motion of a particle on VGA display, we used 128x128 pixel area due to lack of Block RAM of the FPGA. Figure 3 shows the screen shot of the random walker.

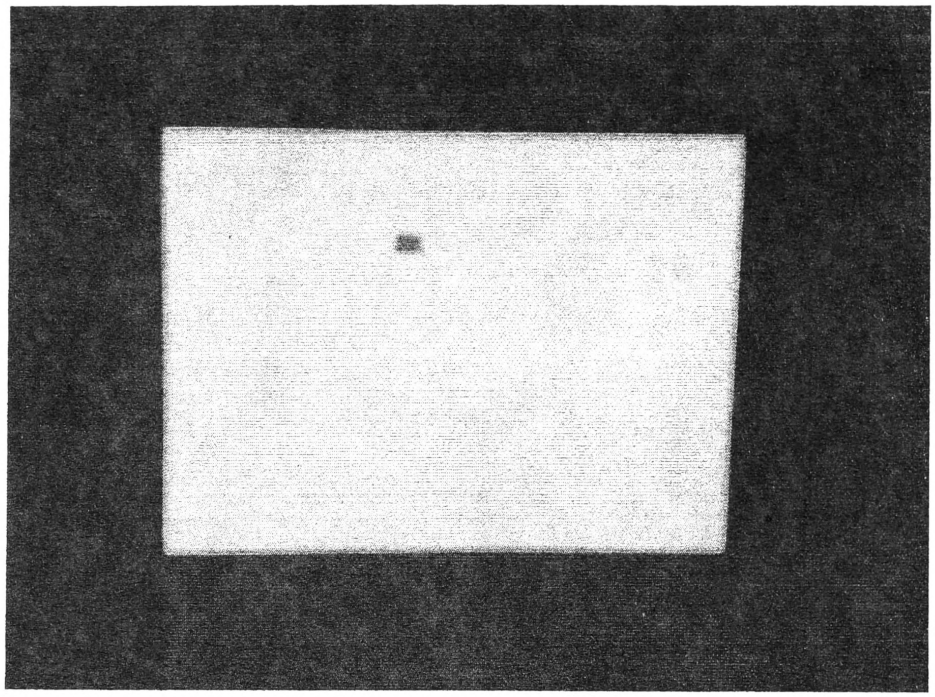

Figure 3: VGA output

Following simulation figure represent the linear feedback shift register to display the random numbers for the given address.

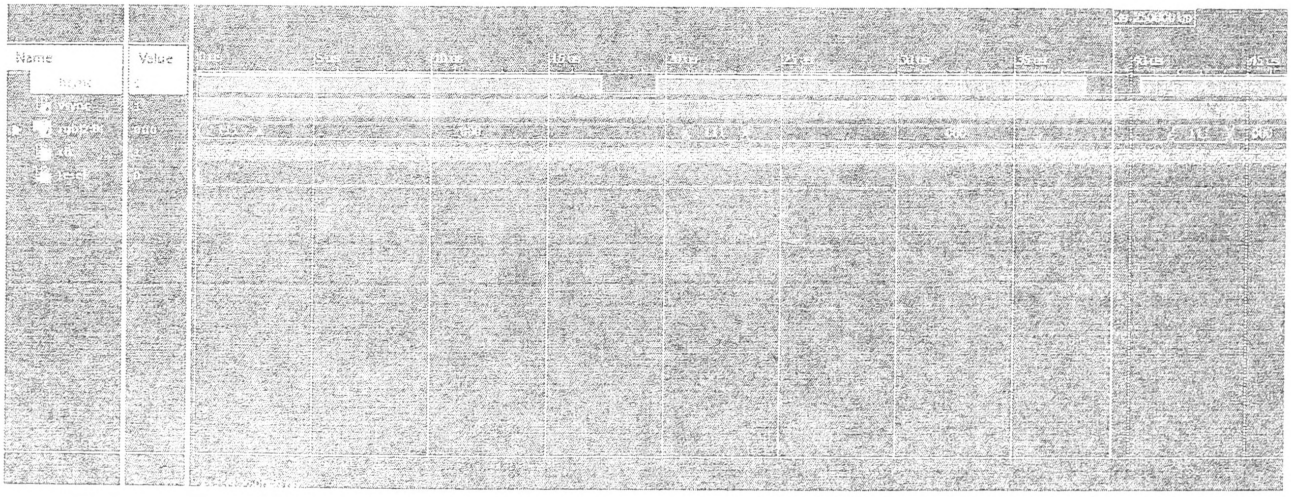

Figure 4: Simulation results for VGA controller

4. CONCLUSION

We implemented VGA controller on FPGA hardware using Verilog Hardware Description Language. To test the system a random walker algorithm was generated and position of the walker displayed on the screen real-time. This system can be used to display outputs of video processing and machine vision application where FPGA hardware is used.

ACKNOWLEDGEMENTS

The author would like to thank many individuals who helped and guided to complete the research project.

REFERENCES

[1]. C. Maxfield FPGAs: Instant, "Access", Elsevier Ltd., USA, 2008

[2]. E. Hwang, "Build a VGA Monitor Controller", Nov. 2004, [Online]. Available: www.circuitcellar.com.

[3]. http://www.researchgate.net/publication/233743657

[4]. [ONLINE] Available at: http://ece.wpi.edu/~rjduck/vga_serial_project.pdf. [Accessed 19 March 2015].

[5]. Zainudin, R. and Caleb, W. (2012). The Design and Implementation of VGA Controller on FPGA. 1st ed. Melaka, Malaysia.

[6]. Spartan-3E Starter Kit Board User Guide, 2006)

[7]. RahmanRokon, I. (2011). Hardware Implementation of PCI Interface Using Verilog & FPGA.1st ed. Pattaya.

[8]. LogiCORE IP Block Memory Generator v7.3. (2012). 1st ed. Xilinx.

[9]. Wasu, R. and R Wadhankar, V. (2015).Review of Design of VGA Controller on FPGA.1st ed. International Journal of Science and research.

[10]. Caleb, R. and Ismail, M. (2012). The Design and Implementation of VGA Controller on FPGA. 1st ed. Melaka: International Journal of Electrical & Computer Sciences.

Proc. Annual Symposium on Research & Industrial Training, 02(2015) 171-177
Department of Electronics – Wayamba University of Sri Lanka