International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Review on Design of VGA Controller Using FPGA

Renuka Wasu[1], Vijay R. Wadhankar[2]

[1] Research Scholar, Agnihotri College of Engineering, Nagthana Road, Wardha (M.S), India

[2] H.O.D of E & C Dept., Agnihotri College of Engineering, Nagthana Road, Wardha (M.S), India

Abstract: ***The work presents a design of VGA Controller using FPGA, as a standard display interface. VGA (Video Graphics Array) has been widely used. In this paper we have given its top layer module design and the timing function. This controller is developed using Verilog HDL based in IEEE standards, to ensure the portability with any manufacture. In this paper we have also given design flow of VGA synchronization signal.***

Keywords: Field-programmable gate arrays (FPGA's), Verilog Hardware Description Language (Verilog HDL), Altera Quartus II Compiler Software, VGA controller**.**

1. Introduction

At present, the research and development of applied digital systems for specific tasks are increasing, such as video conference systems, face recognition systems, surveillance and remote vehicle guidance systems, etc. Among them, VGA (Video Graphic Array) is the most popular display interface. A VGA Controller is a hardware exclusive system that works with high frequency signals and which it was initially implemented on PCB (Printed Circuit Board). But such solutions are often quite oversize and high power dissipation. Digital Signal Processors (DSP), or Graphics Processing Units (GPU) are often used too. DSP is very simple to program, and often result in systems that consume less power than FPGA, but have lesser capabilities in terms of parallelism. On the other hand, GPU provides a very flexible environment for parallelism, but consume a lot of power. With the development of electronic and semiconductor technology, FPGA (Field Programmable Gates Array) provides an impacted size and low power consumption solution. FPGA also has other advantages over these platforms, these advantages are: high clock frequency, high operations per second, code portability, code libraries reusability, low cost, parallel processing, capability of interacting with high or low interfaces, security and Intellectual Property (IP) retention Only the emerging of FPGA technology made VGA controller design accessible and suitable for study, experimentation and research[4].

In this paper, we take the programming methodology and adopt the integration tools (Quartus version 13.0 of Altera) [2]. After giving every module compile, function simulation, layout and timing simulation, each module can be downloaded into FPGA. This method can greatly reduce the size of the circuit board and enhance the reliability of system and design flexibility. As a result, it can greatly reduce the system cost.

2. Literature Review

A few basic information need to be understand before starts the design process.

2.1 Field-programmable Gate Arrays

A field-programmable gate array (FPGA) is an integrated circuit designed to be configured by a customer or a designer after manufacturing – hence "field-programmable". The FPGA configuration is generally specified using a hardware description language (HDL), similar to that used for an application-specific integrated circuit (ASIC).

FPGAs contain programmable logic components called "logic blocks", and a hierarchy of reconfigurable interconnects that allow the blocks to be "wired together" – somewhat like many logic gates that can be inter-wired in different configurations. Logic blocks 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.

Field-Programmable Gate Arrays (FPGAs) are digital integrated circuits (ICs) that contain configurable blocks of logic along with configurable interconnects between these blocks [8]. 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. In most FPGAs, the logic blocks also include memory elements, which may be simple flip-flops or more complete blocks of memory [1].

FPGAs have compensating advantages, largely due to the fact that they are standard parts. There is no wait from completing the design to obtaining a working chip. The design can be programmed into the FPGA and tested immediately. Apart from that, FPGAs are excellent prototyping vehicles. When the FPGA is used in the final design, the jump from prototype to product is much smaller and easier to negotiate. Also, the same FPGA can be used in several different designs, reducing inventory costs [6].

2.2 Verilog Hardware Description Language

Verilog HDL is a Hardware Description Language developed in 1984-1985 by Philip Moorby who needed a simple, intuitive and effective way of describing digital circuits for modeling, simulation and analysis purposes. The language

Licensed Under Creative Commons Attribution CC BY

became the property of Gateway Design Automation, which are later acquired by Cadence Design Systems. From 1990 Cadence Opened the language to the public, which led to the standardization of the language by the IEEE in 1995[9].

Verilog Hardware Description Language (Verilog HDL) is a popular and standard hardware description language which is now extensively used by engineers and scientists on digital hardware designs. Verilog HDL offers many useful features for digital hardware design, that is, Verilog HDL is a general-purpose hardware description language that is easy to learn and easy to use. It is similar in syntax to the C programming language. Verilog HDL allows different levels of abstraction to mix in the same model [7]. Thus, a hardware model can be defined in terms of switches, gates, RTL, or behavioral code. Also, most popular logic synthesis tools support Verilog HDL. This makes it the language of choice for designers [7].

The Verilog HDL language includes capabilities to describe the behavioral nature of a design, the dataflow nature of a design, a design's structural composition, delays and a waveform generation mechanism including aspects of response monitoring and verification, all modeled using one single language. In addition, the language provides a programming language interface through which the internals of a design can be accessed during simulation including the control of a simulation run.

The language not only defines the syntax but also defines very clear simulation semantics for each language construct. Therefore, models written in this language can be verified using a Verilog simulation. The language inherits many of its operator symbols and constructs from the C programming language. Verilog HDL provides an extensive range of modeling capabilities, some of which are quite difficult to comprehend initially. However, a core subset of the language is quite easy to learn and use. This is sufficient to model most application. The complete language, however, has sufficient capabilities to capture the descriptions from the most complex chips to a complete electronic system.

2.3 Altera Quartus II Compiler Software

Quartus II design software is Altera's primary development system. It provides a comprehensive environment for digital design and is an ideal platform for learning both basic and advanced design techniques.

Altera provides the Quartus II Web Edition software, which can be downloaded and used free of charge. This software includes the Nios II soft processor and the SOPC Builder tool.

SOPC Builder, a tool in Quartus II software that eliminates manual system integration tasks by automatically generating interconnects logic and creating a test bench to verify functionality.

Altera Quartus is programmable logic device design software from Altera. Its features include an implementation of VHDL and Verilog for hardware description, visual edition of logic circuits, and vector waveform simulation [2].

Quartus II is a software tool produced by Altera for analysis and synthesis of HDL designs, which enables the developer to compile their designs, perform timing analysis, examine RTL diagrams, simulate a design's reaction to different stimuli, and configure the target device with the programmer [2].

2.4 VGA Controller

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, especially the development of high speed image processing [5]. Apart from that, display will be replacing paper for future. Words of wisdom; seeing is believing and picture telling thousand words, display can give correct information about something. Display is used when people present something. Pictures or texts at display catch more attention than verbal voice when people are doing presentation. When people do that kind of presentation, there must be some device involved in control the display.

2.4.1 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 Fig. 2, 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].

Licensed Under Creative Commons Attribution CC BY

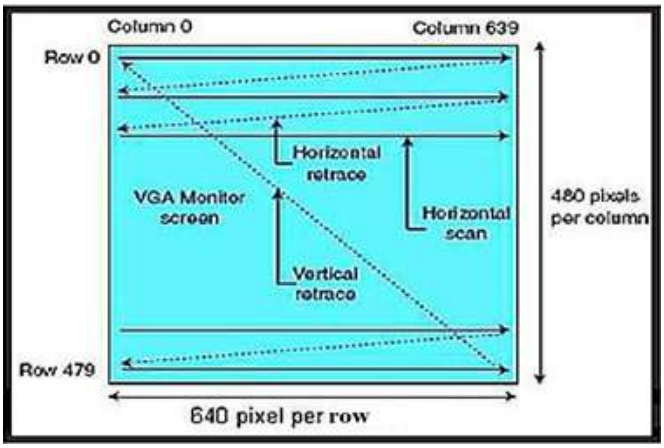


Figure 1: Scanning pattern of VGA Controller

2.4.2 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].

The horizontal and vertical synchronization signals are used to control the timing of the scan rates. Unlike the three analog RGB signals, these two sync signals are digital signals. In other words, they take on either logic 0 or logic 1 value. The horizontal synchronization signal is used to control the horizontal deflection circuit in the VGA monitor, so that the start and end of a line of pixels is correctly displayed across the visible display area of the screen. Meanwhile, the vertical synchronization signal is used to control the vertical deflection circuit in the VGA monitor, so that the start and end of a frame (of lines) is correctly displayed between the top and bottom edges of the visible display area of the screen. In other words, the horizontal synchronization signal determines the time it takes to scan a row, while the vertical synchronization signal determines the time it takes to scan the entire screen. By manipulating these two sync signals and the three RGB signals, images are formed on the monitor screen [3].

2.4.3 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:

$$\frac{1}{25.175MHz} \approx 0.0397\ \mu s \text{ per clock cycle.}$$

Referred to Figure 2, for section B of the horizontal synchronization signal, 3.81 µs is needed, which is approximately 96 clock cycles (3.81/0.0397). For section C, 1.90 µs is needed, which is approximately 46 clock cycles.

Similarly, 640 clock cycles (section D) for the 640 columns of pixels and 16 clock cycles for section E [3].

The total number of clock cycles needed for each row scan is 800 clock cycles (96 + 48 + 640 + 16). Notice that with a 25.175 MHz clock, section D requires exactly 640 cycles, generating the 640 columns per row. If a different clock speed is used, a different screen resolution will be obtained. Because the vertical sync signal is analogous to the horizontal sync signal, the same calculations can be performed as with the horizontal sync regions to obtain the number of cycles needed for each vertical region. However, instead of using the number of periods of a 25.175 MHz clock, the times for each vertical region are multiples of the horizontal cycles. For example, the time for a horizontal cycle is 31.77 µs, and section P requires 64 µs, which is approximately two horizontal cycles (2 × 31.77). Section Q requires 1,048 µs, which equals to 33 horizontal cycles (1,048/31.77). The calculation for section R is 480 horizontal cycles (15,250/31.77).

The number of clock cycles required by the four regions in the horizontal and vertical sync signal is summarized in Table 2 (a) and 2 (b).

Table 1(a): The number of clock cycles required by the four regions in the horizontal sync signal

	B	C	D	E	Total
Time	3.81µs	1.90µs	25.4µs	0.64µs	31.77µs
No. of a 25.175MHz clock cycles	96 cycles	48 cycles	640 cycles	16 cycles	800 cycles

Table 1(b): The number of clock cycles required by the four regions in the vertical sync signal

	P	Q	R	S	Total
Time	64µs	1048µs	15250µs	318µs	16784 µs
No. of Horizontal cycles	2 cycles	33 cycles	480 cycles	10 cycles	525 cycles

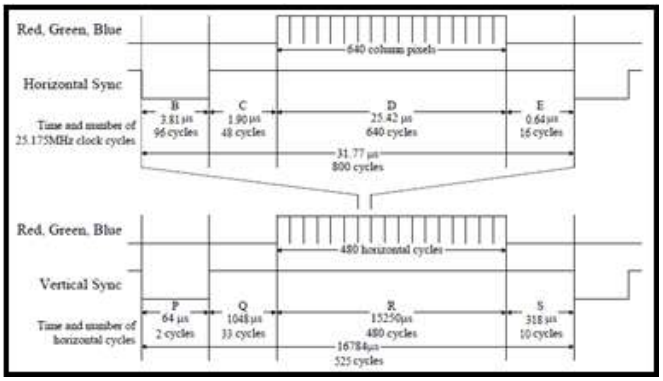


Figure 2: The horizontal and vertical synchronization signal-timing diagram

3. 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

Licensed Under Creative Commons Attribution CC BY

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 horizontal scan 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.

While "h_count" is increased by 1, it will check whether the value of "h_count" is less than 96 or not. The value of "h_count" less than 96 indicates horizontal retrace. If the value of "h_count" is less than 96, "h_sync" will be set to 0. If the value of "h_count" is not less than 96, "h_sync" will be set to 1.

Also, while "h_count" is increased by 1, it will check whether the value of "h_count" is less than 799 and greater or equal to 96 or not. The value of "h_count" less than 784 and greater or equal to 144 indicates the display area for horizontal scan. If the value of "h_count" is less than or equal to 799 and greater or equal to 96, "h_sync" will be set to 1. If the value of "h_count" is greater than 799 and less than or equal to 96, "h_sync" will be set to 0.

While "v_count" is increased by 1, it will check whether the value of "v_count" is less than 2 or not. The value of "v_count" less than 2 indicates vertical retrace. If the value of "v_count" is less than 2, "v_sync" will be set to 0. If the value of "v_count" is not less than 2, "v_sync" will be set to 1.

Also, while "v_count" is increased by 1, it will check whether the value of "v_count" is less than or equal to 524 and greater or equal to 2 or not. The value of "v_count" less than or equal to 515 and greater or equal to 2 indicates the display area for vertical scan. If the value of "v_count" is less than or equal 524 and greater or equal to 2, "v_sync" will be set to 1. If the value of "v_count" is greater than 524 and less than or equal to 2, "v_sync" will be set to 0.

4. Conclusion

This paper shows the efficient used of Field-programmable Gate Array (FPGA) in developing a VGA Controller. By using Verilog Hardware Description Language (Verilog HDL) on FPGA, VGA Controller could be constructed easily without construction the circuit manually; just to write a behavioral model based on its logic flows, then simulate and synthesize it. Thus, VGA Controller using FPGA might be good choice as it is easy to be designed and used.

References

[1] Altera Corporation, "What is an FPGA?" Oct. 26- 2011, [Online]. Available: http://www.altera.com/products/fpga.html.
[2] Altera Corporation, "Introduction to Quartus II", Feb. 2013, [Online]. Available: http://www.altera.com/education/univ/software/quartus2/unvquartus-2.html.
[3] E. Hwang, "Build a VGA Monitor Controller", Nov. 2004, [Online]. Available: www.circuitcellar.com.
[4] Fangqin Ying, Xiaoqing Feng, "Design and Implementation of VGA Controller Using FPGA", International Journal of Advancements in Computing Technology (IJACT), Vol. 4, No. 17, pp. 458-465, Sep 2012.
[5] Radi H.R, Caleb W.W.K, M.N.Shah Zainudin, M.Muzafar Ismail, "The Design and Implementation of VGA Controller on FPGA", International Journal of Electrical & Computer Sciences, Vol. 12, No. 5, pp. 56-60, Oct. 2012.
[6] Wayne Wolf, "FPGA-Based System Design", Pearson Education Inc., India, 2004.
[7] Samir Palnitkar, Verilog HDL: A Guide to Digital Design and Synthesis", Sun Microsystems, Inc., USA, 2003.
[8] Clive "Max" Maxfield, "FPGAs: Instant Access", Elsevier Ltd., USA, 2008.
[9] ALDEC, Inc., "EVITA Enhanced Verilog Tutorial with Applications" 1998, Available: www.aldec.com.

Author Profile

Renuka Wasu received her Diploma in Electronics & Telecommunication from Acharya Shrimannarayan Polytechnic, Pipri (Wardha), India in 2010 and B.E. in Electronics & Telecommunication from S.K.N.C.O.E., Pune, India in 2013.Currently she is research scholar and pursuing M.Tech in Electronics from Agnihotri College of Engg. Nagthana Wardha, India.

Prof. Vijay R. Wadhankar received his B.E. in Electronics from B.D.C.O.E., Wardha, India and M.Tech in VLSI from G.H.Raisoni College of Engg., Nagpur, India. He is working as Head of Dept. in Agnihotri College of Engg. Nagthana Wardha, India.

Licensed Under Creative Commons Attribution CC BY