International Journal on Recent and Innovation Trends in Computing and Communication ISSN: 2321-8169
Volume: 3 Issue: 5 159 – 162

Design of VGA Controller Using FPGA

Renuka A. Wasu
Research Scholar, Electronics and Communication Dept.
Agnihotri College of Engineering
Nagthana Road, Wardha (M.S), India.
wasu.renuka@gmail.com

Vijay R. Wadhankar
HOD, Electronics and Communication Dept.
Agnihotri College of Engineering
Nagthana Road, Wardha (M.S), India.
vijayhingan@yahoo.com

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 design of VGA synchronization signal with timing function. This is developed using Verilog HDL based in IEEE standards, to provide the portability with any manufacture.

Keywords- *Integrated Circuit, Field Programmable Gate Array (FPGA), VGA Controller, Altera Quartus II.*

*****

I. INTRODUCTION

For specific tasks, at present, the research and development of applied digital systems are increasing, such as video conference systems, face recognition systems, surveillance and remote vehicle guidance systems, etc. Among them, the most popular display interface is VGA (Video Graphic Array). VGA (video graphics array) is a video display standard. It offers a simple method to connect a system with a monitor for showing images or information and it was initially implemented on PCB (Printed Circuit Board). But such solutions are big and high power dissipation. Digital Signal Processors (DSP) or Graphics Processing Units (GPU) is 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 for parallelism, but consume a lot of power.

With the development of technology, FPGA (Field Programmable Gates Array) provides an effected size and low power consumption solution. FPGA also has other advantage, these 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[1].

II. INTEGRATED CIRCUIT

Integrated circuit (IC), an assembly of electronic components, fabricated as a single unit, in which miniaturized active devices and passive device and their interconnections are built up on a thin substrate of semiconductor material, also called microelectronic circuit or chip,.

The resulting circuit is thus a small monolithic “chip”, which may be as small as a few squ.cm or only a few squ.mm.

The individual circuit components are generally microscopic in size.

ICs have two main advantages, cost is low because the chips, rather than being constructed one transistor at a time, it is printed as a unit by photolithography with all their components. Furthermore, packaged ICs use very less material than any discrete circuits.

Following are types of Integrated Circuit:

1) Standard ICs: ICs that are used for particular digital logic. They are not reconfigurable. E.g.7400 series.

2) Programmable ICs: ICs that enable designers to program customized digital logic in the field. They are reconfigurable. E.g. FPGA.

3) Application Specific Standard Product (ASSP): ICs that are dedicated to a specific application market and sold to one user. They are hard-wired. E.g. these ICs perform video and/or audio encoding and/or decoding.

4) Application Specific Integrated Circuit (ASIC): ICs that are dedicated to a specific application market and sold to a single customer. They are not reconfigurable. E.g. Voice recorder and speed controller for remote control cars.

The VGA controller is an Application Specific Standard Product; these circuits are permanently fabricated on silicon wafers. We simply can’t change the circuitry and replace it with something else. If we need to change something in the design, we have to scrap the whole chip and start with new one.

But, FPGA have advantages over the ASSP, they are

1) FPGA, enable us to build exactly the hardware we need.
2) An FPGA, we can do operations in a simpler, faster, more energy-efficient way than they could be done in the microprocessor cores of an ASSP.
3) Rapid prototyping.
4) Shorter time to market.

So, it’s better to design VGA Controller using FPGA.

III. FIELD PROGRAMMABLE GATE ARRAY (FPGA)

A Field programmable gate array (FPGA) is an integrated circuit designed to be configured by a designer after manufacturing or a customer. Hence “field programmable”.

By modern standards, logic with 10,000 gates is not large. To implement larger circuits, it is convenient to use a different type of chip with larger logic capacity. A FPGA is a programmable logic device that supports implementation of large logic circuits. FPGAs do not contain AND or OR planes, instead FPGAs provide logic blocks for implementation of the required functions.

The general structure of an FPGA is shown in Fig. 1. It contains three main types of resources: logic blocks, I/O blocks for connecting to the pins of the package, and interconnection wires and switches. The logic blocks are arranged in a two dimensional array, and the interconnection

wires are organized as horizontal and vertical routing channels between rows and columns of logic blocks. The routing channels contain wires and programmable switches that allow the logic blocks to be interconnected in many ways.

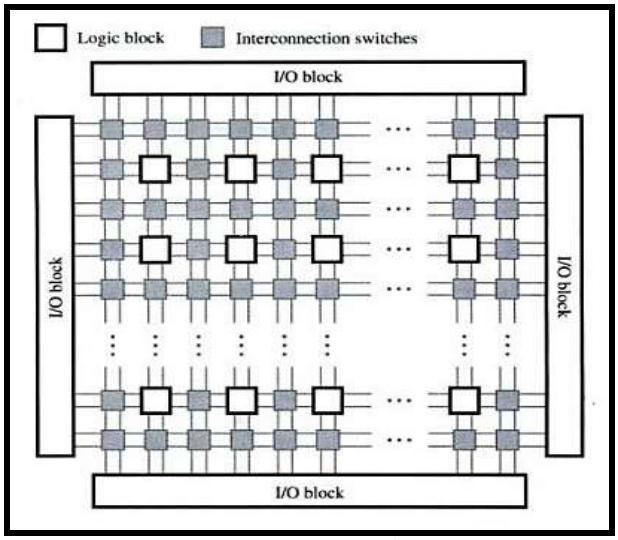

Fig. 1 The general structure of an FPGA

FPGAs can be used to implement logic circuits of more than a million equivalent gates in size. Some of commercial FPGA products are Altera and Xilinx[2].

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 [3].

IV. VGA Controller

VGA (video graphics array) is a video display standard. It offers a simple method to connect a system with a monitor for showing images or information.

For a standard VGA format monitor screen have 640 columns by 480 rows of picture elements called pixel. By turning on and off individually pixels, an image is displayed on the screen. An image generates by turning on combine numerous pixels, not by turning on one pixel. The entire screen of monitor is continuously scans, by 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 in the top left corner of the screen from row 0, column 0 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. 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. All the pixels are turned off during the horizontal and the vertical retraces [4].

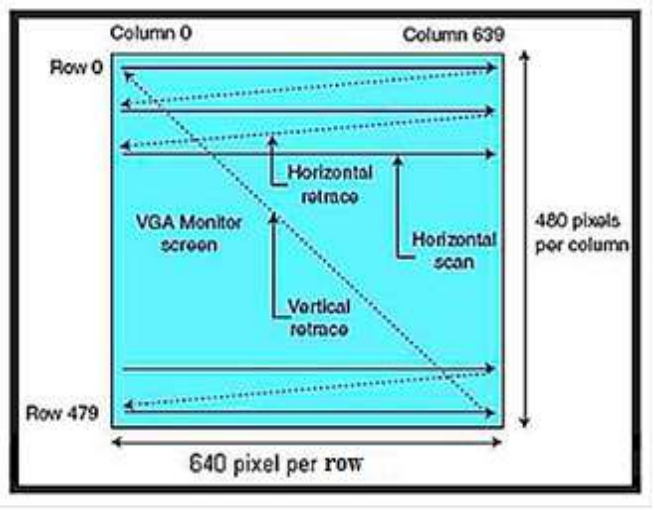

Fig. 2 Scanning pattern of VGA Controller

The VGA monitor has 5 control 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 [4].

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 the timing information for each line of the picture, 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 the timing information for whole picture i.e. number of picture per second and when each one should start. By handling these two sync signals and the three RGB signals, images are formed on the monitor screen [4].

RGB Color model is an additive color model. It generates various colors by adding three basic colors– red, green, and blue. A color can be represented by how much each of the three basic colors is included. Each of the three basic colors is one color component of the represented color.

TABLE I
3-Bit Display Color Code

RED (R)	GREEN (G)	BLUE (B)	Resulting Color
0	0	0	Black
0	0	1	Blue
0	1	0	Green
0	1	1	Cyan
1	0	0	Red
1	0	1	Magenta
1	1	0	Yellow
1	1	1	White

A clock with a 25.175 MHz frequency is used, to obtain the 640 × 480 screen resolution. For a higher screen

resolution, a higher clock frequency is required. 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 Fig. 3, for section B of the horizontal synchronization signal, 3.81 µs is required, which is around 96 clock cycles (3.81/0.0397). For section C, 1.90 µs is required, which is around 46 clock cycles. Similarly, for section D, 640 clock cycles, for the 640 columns of pixels and 16 clock cycles for section E.

For each row scan, the total number of clock cycles required is 800 clock cycles (96 + 48 + 640 + 16). It is notice that with a clock 25.175 MHz, 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. The same calculations can be performed as with the horizontal sync regions to obtain the number of cycles needed for each vertical region, because the vertical sync signal is homologous to the horizontal sync signal. 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 needs 64 µs, which is approximately two horizontal cycles (2 × 31.77). For section Q, 1048 µs is needed which equals to 33 horizontal cycles (1,048/31.77). For section R, 480 horizontal cycles (15,250/31.77) are required [5].

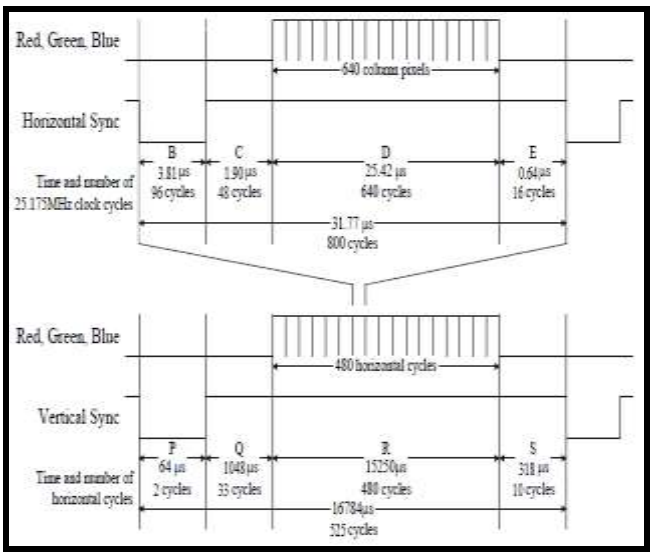

Fig. 3 The horizontal and vertical synchronization signal-timing diagram

By the four regions, in the horizontal and vertical sync signal the number of clock cycles required is summarized (see table II and III) [4].

Table II
THE NUMBER OF CLOCK CYCLES REQUIRED BY THE FOUR REGIONS IN THE HORIZONTAL SYNC SIGNAL

Time	B	C	D	E	Total
	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 III
THE NUMBER OF CLOCK CYCLES REQUIRED BY THE FOUR REGIONS IN THE VERTICAL SYNC SIGNAL

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

V. SOFTWARE DESIGN

The VGA controller is developed using Verilog (hardware description language) based in the IEEE standards, to make sure the flexibility with any user. The main module of VGA controller is Timing Control module. It is designed using Verilog language in Quartus II software.

A. *Altera Quartus II Software*

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 [6].

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 [6].

The Altera Quartus II design software is a multiplatform design environment that easily adapts to your specific needs in all phases of FPGA and CPLD design. Quartus II software delivers the highest productivity and performance for Altera FPGAs, CPLDs, and HardCopy ASICs [6].

This design software makes it easy to implement a desired logic using a programmable logic design such as FPGA chip. A FPGA Computer Aided Design (CAD) flow is shown in Fig. 4.

1) *Design Entry:* the desired circuit is design either by means of a schematic diagram, or by using a hardware description language, such as Verilog or VHDL.
2) *Synthesis:* the entered design is synthesized into a circuit that consists of the logic elements (LEs) provided in the FPGA chip.
3) *Functional Simulation:* the synthesized circuit is tested to verify its functional correctness; this simulation does not take into account any timing issues.
4) *Fitting:* the CAD Fitter tool determines the placement of the LEs defined in the netlist into the LEs in an actual FPGA chip; it also chooses routing wires in the chip to make the required connections between specific LEs.
5) *Timing Analysis:* propagation delays along the various paths in the fitted circuit are analyzed to provide an indication of the expected performance of the circuit.
6) *Timing Simulation:* the fitted circuit is tested to verify both its functional correctness and timing.
7) *Programming and Configuration:* the designed circuit is implemented in a physical FPGA chip by programming the configuration switches that configure the LEs and establish the required wiring connections [6].

 ISSN: 2321-8169
159 – 162

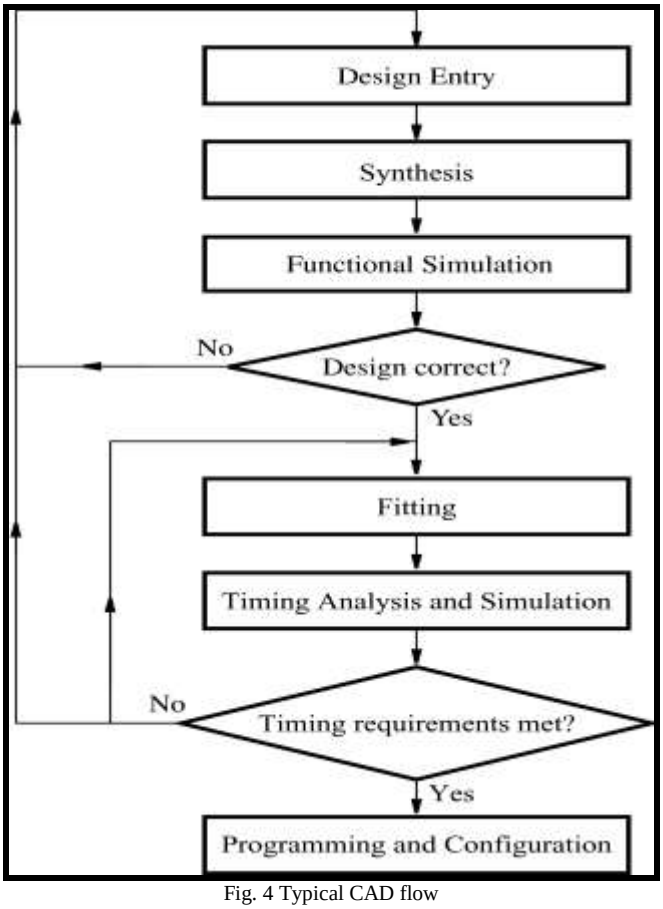

Fig. 4 Typical CAD flow

B. Designing of Timing Control Module

The Timing Control Module is a main module of VGA Controller. This module consists of horizontal synchronization signal (H_SYNC) and vertical synchronization signal (V_SYNC). These signals must be design in accordance with the VGA timing standard. By using, a clock of 25.175MHz frequency, screen resolution of 640*480 is obtained. VGA_clk defines the needed time to display the information. The controller decodes the output of the H_count to generate the H_SYNC signal time. This counter may be used to locate the pixel of a given row. The output for the module V_count that increases the H_SYNC pulse may be used to generate the V_SYNC output time.

The module called clock divider generates a frequency of 25MHz to work with a 640*480 resolution. Fig. 5 shows timing control module [1].

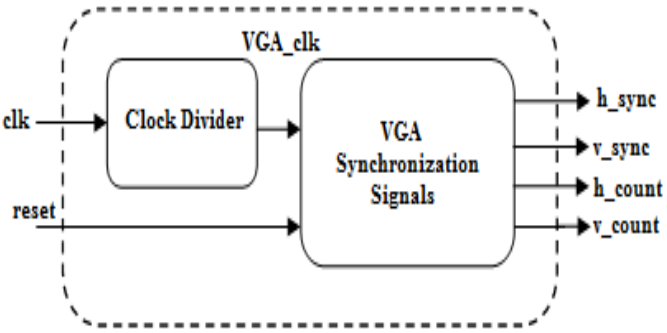

Fig. 5 Timing control module

C. Results of Timing Simulation

In order to observe that whether the design meet the requirements, simulation test for timing module is carried out by using Quartus II software.

When reset signal is "low", horizontal and vertical scanning takes place. The horizontal and vertical synchronization signals are able to meet the requirements of the VGA display. Fig 6 shows results of timing control module.

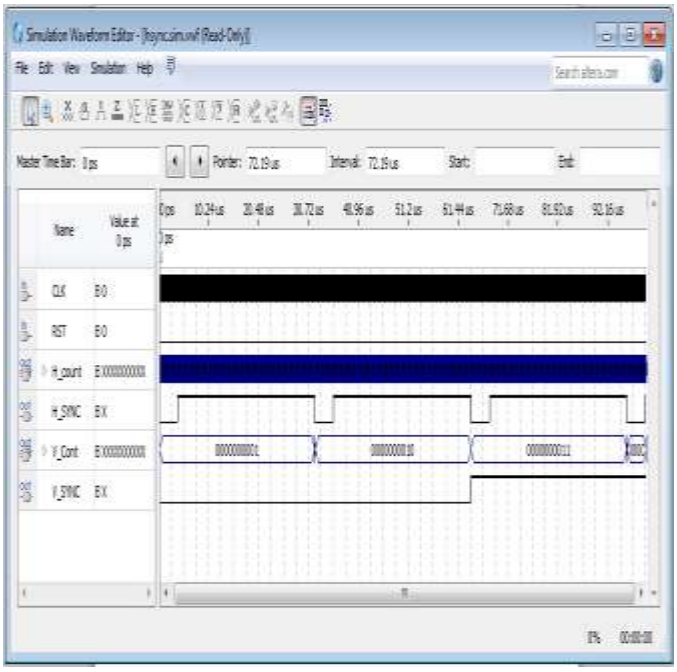
Fig. 6 Result of Timing simulation

VI. CONCLUSION

This paper shows the systematic used of FPGA in developing a VGA Controller by using Verilog hardware description language (Verilog HDL). Timing control module of VGA controller is constructed, just by writing a program based on its logic flows, then simulates and synthesize it.

REFERENCES

[1] 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.

[2] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with Verilog Design*, 2rd ed., McGraw Hill Education (India) Private Limited, 2008, pp.109-114.

[3] Wayne Wolf, *FPGA-Based System Design*, Pearson Education Inc., India, 2004.

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

[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] Altera Corporation, "Introduction to Quartus II", Feb. 2013, [Online]. Available: http://www.altera.com/education/univ/software/quartus2/unvquartus-2.html.