

An FSM Based VGA Controller with 640×480 Resolution

Ila.Nagarjuna, Pillem. Ramesh

Abstract: picture caught more attention than verbal voice. A video graphic adapter provides interface between the computer and monitor. The main purpose of this project is to design and implement VGA controller on FPGA. The proposed VGA controller is written based on the block diagram using verilog HDL. Also functions required for the VGA controller are included in the verilog code and test bench is created to test the functions written to ensure the FPGA VGA works correctly and accurately without an errors. This design has display capability supported by a virtually every video adapter on the market, and can be extendable up to 1368x768. we have generated the shapes of coordinate geometry to get in motion, any moving geometric objects can be implemented. The motion can be generated using finite state machines (FSM), with raster pattern from left to right and top to bottom, many shapes can be generated and motion can be in zigzag. We used FILE OUT operations, so that the generated object can be verified before it gets implemented, no need of storage devices like FIFO (leads to complexity with FIFO depth calculations), More economic, we can even check the implementation results without FPGA in static image format. Cadence tool is used for observation and verification in terms of performance. And the code coverage of the design is achieved.

Index Terms—Xilinx 12.3, modelsim SE 6.3f, spartan3E,

I. INTRODUCTION

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. 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. Verilog Hardware Description Language (Verilog HDL) is 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. Thus, a hardware model can be defined in terms of switches, gates, RTL, or behavioral code.

Manuscript published on 30 April 2013.

* Correspondence Author (s)

Nagarjuna. Ila, Pursuing M.Tech in VLSI at K L University, Vijayawada, AP, India.

Pillem. Ramesh, Asst. Professor in K. L. University, Vijayawada, AP, India.

© The Authors. Published by Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP). This is an [open access](http://creativecommons.org/licenses/by-nc-nd/4.0/) article under the CC-BY-NC-ND license <http://creativecommons.org/licenses/by-nc-nd/4.0/>

Also, most popular logic synthesis tools support Verilog HDL. This makes it the language of choice for designers. A few basic knowledge need to be understand before starting the design process.

II. PREREQUISITES

A. Field-programmable Gate Arrays

FPGA's are a semiconductor device level containing programmable logic components Called "logic blocks" and programmable interconnects. Logic blocks can be programmed to perform the function of basic logic gates such as AND, and XOR, or more complex combinational functions such as decoders or simple mathematical functions. FPGAs are also known as reconfigurable devices. These reconfigurable FPGAs are generally favored in prototype building because the device does not need to be thrown away every time a change is made. This allows one piece of hardware to perform several different functions. Of course, those functions cannot be performed at the same time. Besides that, FPGAs are standard parts, they are not designed for any particular function but are programmed by the customer for a particular purpose.

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.

B. Verilog Hardware Description Language

Verilog HDL is a description language that can be used to model a digital system at many levels of abstraction ranging from the algorithmic-level to the switch-level. The complexity of the digital system being modeled could vary from that of a simple gate to a complete electronic digital system, or anything in between. The digital system can be described hierarchically and timing can be explicitly modeled within the same description.

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.

C. FILE OUTPUT FUNCTIONS

FORMAL DEFINITION

File I/O functions performs operations on files.

SIMPLIFIED SYNTAX

```
$fopen (file_name);
$fclose (file_name);
$fwrite (arguments);
$fdisplay (arguments);
$fmonitor (arguments);
$strobe (arguments);
$readmemb
("file",memory_identifer[,begin_address[,end_address]]);
$readmemh
("file",memory_identifer[,begin_address[,end_address]]);
```

DESCRIPTION

The \$fopen function opens a file and returns a multi-channel descriptor in the format of an unsigned integer. This is unique for each file. All communications between the simulator and the file take place through the file descriptor. Users can specify only the name of file as n argument; this will create a file in the default folder or a folder given in the full path description. To close an opened file use the \$fclose function. To read the data from a file and store in a memory, use the functions: \$readmemb and \$readmemh. The \$readmemb task reads binary data and \$readmemh reads hexadecimal data. Data has to exist in a text file. White space is not allowed to improve readability, as well as comments in both single line blocks. The numbers have to be stored as binary or hexadecimal values. The basic form of memory file contains numbers.

II. DESIGN AND IMPLEMENTATION

A. Block Diagram

The frequency of clock and refresh rate of VGA controller block are 25 MHz to 60 Hz. Meanwhile, “v_sync,h_sync” signals are used to generate timing and synchronized 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. Besides that, “address generator” block is used to generate address for the image data means for row address and column address” block by using the “h_sync” and “v_sync” signal. The proposed VGA architecture can be viewed.

B. Synchronization of horizontal and vertical counter pulse

First and foremost, the h_count keep on increasing, when reset button is sensed means 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 0 or not and makes the h_sync equals to 0. If this condition fails it will check whether h_count reaches 95, at that instance h_sync equals to 1. If this condition fails again as the h_count increasing reaches to 799, then h_count and h_sync becomes 0 from that instance v_count gets incremented. When v_count reaches to 2 it makes v_sync equals to 1, else if v_count equals to 0, then v_sync also equals to 0. When v_count reaches to 521 all variables v_count, h_count, v_sync, h_sync are equals to 0.

The h_count and v_count indicates the horizontal and vertical counters, which we again used as row address and column address for storing them in address generator module. According to timing control diagram it checks whether the value of h_count is less than 785 and greater than 143. This indicates the display area for horizontal scan. During this condition only the horizontal portion can be visualized in the form of pixels. Turning on and off of particular pixels will create an impression. Similarly it checks whether the value is less than 512 and greater than 31. By the use of and operation between the horizontal and vertical regions the image impression can be visualized.

The synchronization of pulses can be better viewed. The timing for synchronization is based on the timing table.

Symbol	Parameter	Vertical Sync			Horizontal Sync	
		Time	Clocks	Lines	Time	Clocks
T _S	Sync pulse time	16.7 ms	416,800	521	32 μs	800
T _{DISP}	Display time	15.36 ms	384,000	480	25.6 μs	640
T _{PW}	Pulse width	64 μs	1,600	2	3.84 μs	96
T _{FP}	Front porch	320 μs	8,000	10	640 ns	16
T _{BP}	Back porch	928 μs	23,200	29	1.92 μs	48

C. Design Flow of VGA Controller

First and foremost, a 25 MHz “vga_clk” is generated from a 60 MHz input clock. Then, v_sync, h_sync is generated for counter module. The timing diagram for horizontal and vertical scan is generated also. After that, address is generated for storing row and column address each are of 10 bit size. These signals are further given as inputs to the shape generator. The interface of Spartan 3E can be visualized.

D. Implementation

In any design specifications are considered first, according to the specifications the architecture is build, which is like a blue print. As it consists it requires five blocks namely signal generator, address generator, shape generator, motion controller, and generator. The stored row address and column address are given as input to the shape generator along with shape enable, which is of 2 bit size. In this module all possible geometric shapes can be coded. But the output can be taken as 1 bit shape out. The row address, column address, shape enable and motion enable of size 1 bit are the inputs. According to screen resolution some intermediate x, y values have taken as horizontal and vertical representation of image respectively.



The motion controller output is taken as input to the shape generator. The motion controller uses the FSM (finite state machine), where we generate motion for the objects implemented. Finally shape out, select of size 3 bit, which can generate all 8 possible combinations, by using decoder logic with R, G, B signals.

Further the design is observed and verified in Cadence tool . The synthesis result of VGA controller is mentioned in the diagrams.

Speed Grade: -5:

Minimum period: 15.260ns (Maximum Frequency: 65.531MHz)

Minimum input arrival time before clock: 5.924ns

Maximum output required time after clock: 4.063ns

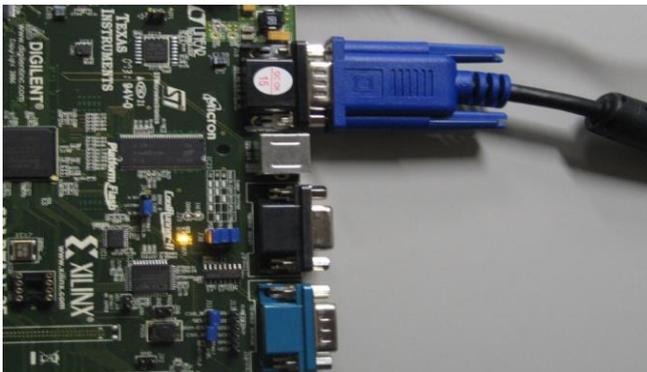
Maximum combinational path delay: No path found

vga_top Project Status (04/24/2013 - 11:21:50)			
Project File:	pc_show.xise	Parser Errors:	No Errors
Module Name:	vga_top	Implementation State:	Synthesized
Target Device:	xc3s500e-3fg320	• Errors:	No Errors
Product Version:	ISE 12.3	• Warnings:	1 Warning (1 new)
Design Goal:	Balanced	• Routing Results:	
Design Strategy:	Xilinx Default (unlocked)	• Timing Constraints:	
Environment:	System Settings	• Final Timing Score:	

Device Utilization Summary (estimated values)			
Logic Utilization	Used	Available	Utilization
Number of Slices	241	4635	5%
Number of Slice Flip Flops	38	3312	0%
Number of 4 input LUTs	445	3312	4%
Number of bonded IOBs	13	232	5%
Number of MULT18K1:8SIOs	2	20	10%
Number of GCLKs	1	24	4%

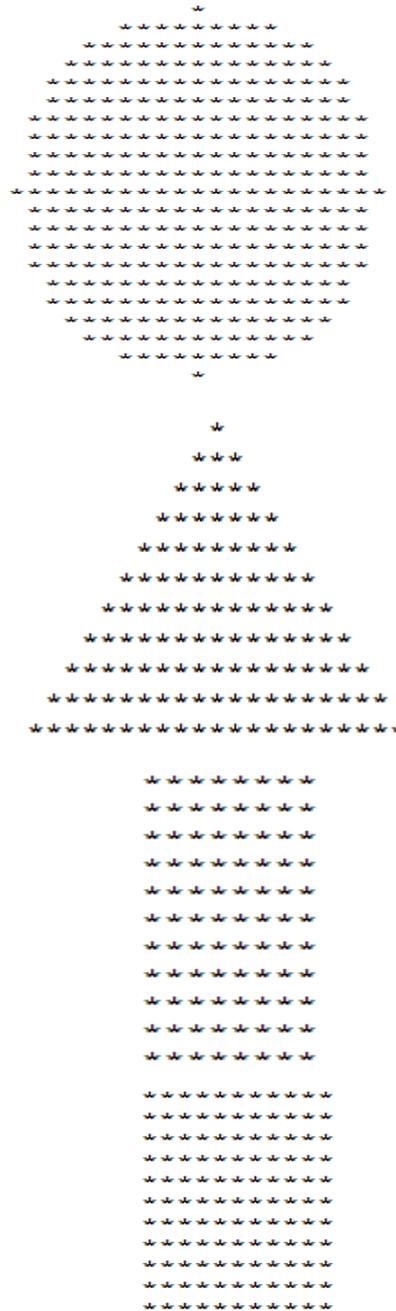
The possible color combinations depends on the number of color select pins we take.

VGA_RED	VGA_GREEN	VGA_BLUE	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

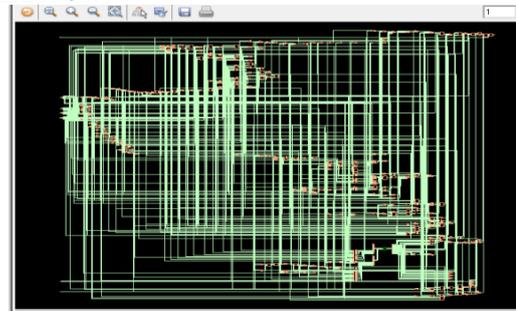


VGA interface with PCB(printed circuit board).

III. FILE OUT FUNCTION RESULTS



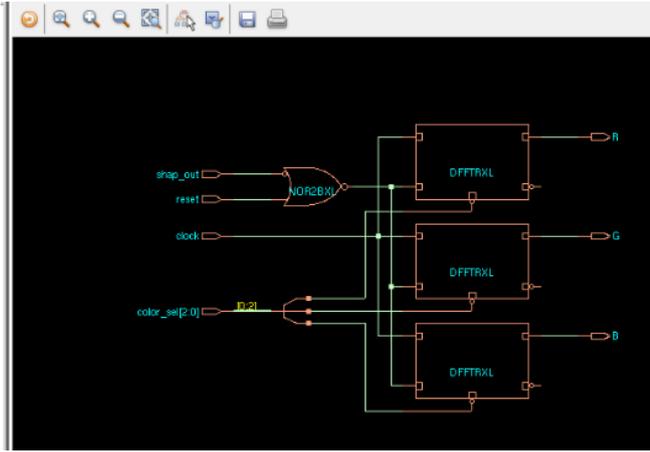
CADENCE RESULTS OF MODULES SHAPE GENERATOR



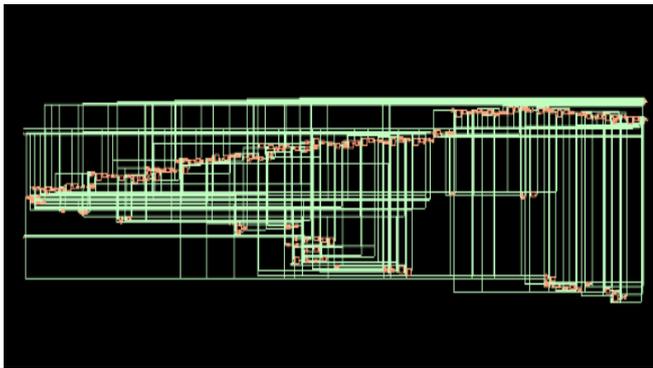
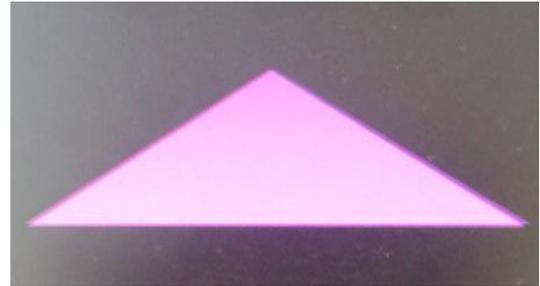
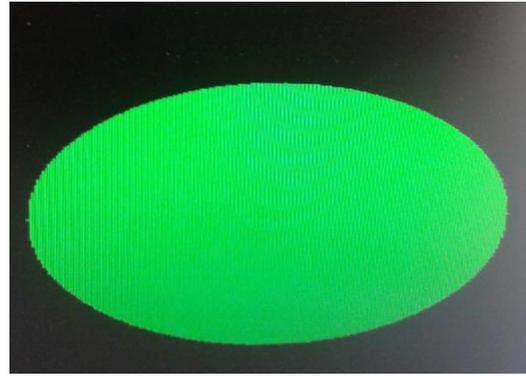
COLOUR GENERATOR



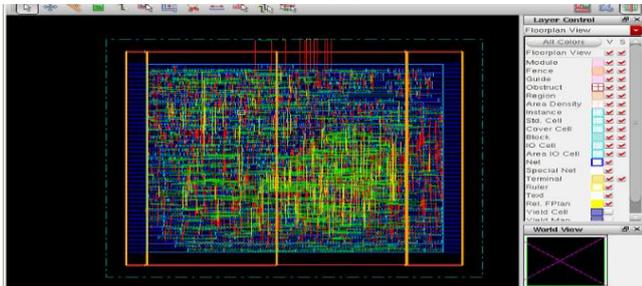
An FSM Based VGA Controller with 640x480 Resolution



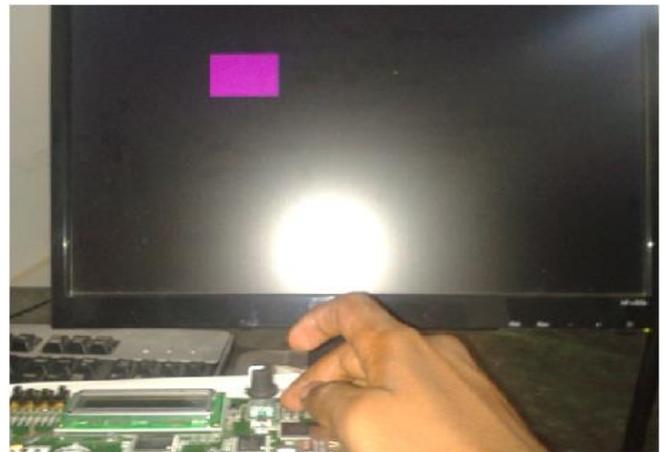
MOTION CONTROLLER



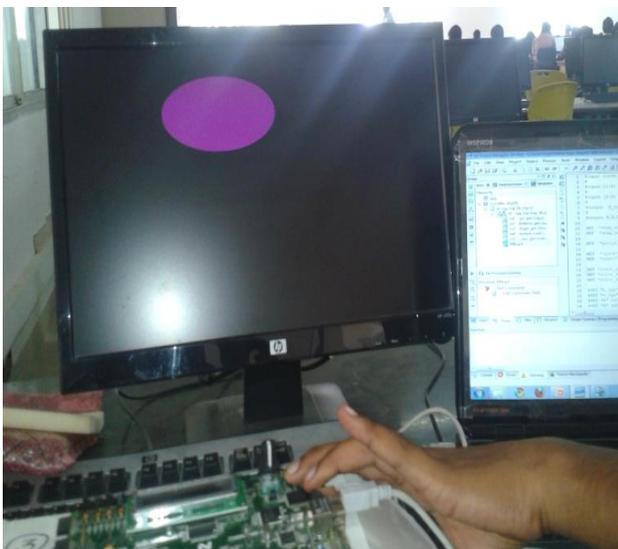
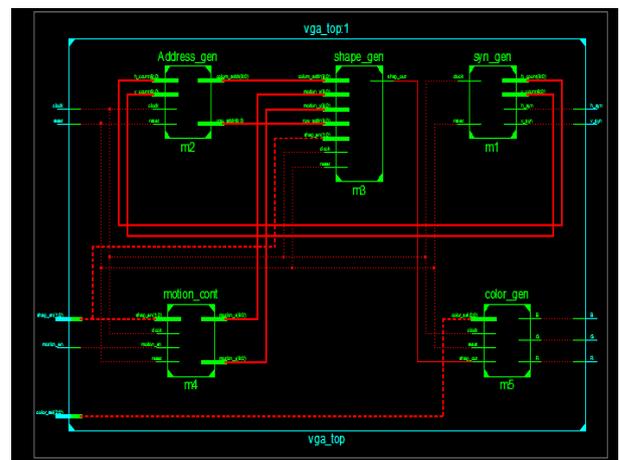
LAYOUT



FPGA IMPLEMENTATION RESULTS IN COLOUR COMBINATIONS



XILINX TOP MODULE SYNTHESIS



Total power consumption in cadence

Units	:	1mw	
Total internal power	:	0.2591	49.73%
Total switching power	:	0.2129	40.86%
Total leakage power	:	0.04906	9.416%
Total power:		0.521	

IV. CONCLUSION

We have different types of VGA controllers. In this we have presented an efficient VGA controller as an alternate of FIFO based, where it requires customizable internal FIFO memory, leads to complex with the FIFO depth calculations and FPGA devices with limited SRAM memory. Through this we are not storing any memory, but just capturing image for motion. Generally we use file input and output functions for representing text information. But here we are using those functions to specify them in a given folder with static representation of image, which sometimes avoid availability of hardware. And the design can be extendable up to 1368x 768. The design is observed and verified with the CADENCE tool in terms of performance. And code coverage of the design is achieved.

REFERENCES

- [1] C. Maxfield, "FPGAs: Instant Access", Elsevier Ltd., USA, 2008.
- [2] Altera Corporation, "What is an FPGA?" (26 October 2011), [Online]. Available: <http://www.altera.com/products/fpga.html>
- [3] D. Bing, Z. Qi, and W. Rui, "The design and implementation of VGA image controller based on FPGA", College of Automation, Harbin Engineering University, China, 2006.
- [4] CAST. High Resolution Display Controller -Datasheet. .technical report, 2010.
- [5] Think Silicon Ltd. Think LCD Display Controller - product specification. Technical report, 2010.
- [6] Javior Valcarce Garcia. Monochrome Test-Mode VGA Video Display Adapter. Technical report, 2009.
- [7] E. Hwang, "Build A VGA Monitor Controller", Circuit Cellar, USA, 2004.
- [8] Altera Corporation, "Introduction to Quartus II" (28 February 2012), [Online]. Available: http://users.ece.gatech.edu/~hamblen/UP3/intro_to_quartus2.pdf
- [9] M. Khalil-Hani, "Starter's Guide to Digital Systems VHDL & Verilog Design", Prentice Hall, Malaysia, 2007.



Nagarjuna. ILA was born at Chinarikatla village, Konankana mitla mandal, Prakasam district, Andhrapradesh state, India, on 10th August 1990. He received his B.Tech in Electronics & Communication Engineering from Noor college of Engineering & Technology (NCET), affiliated to JNTU. And pursuing M.Tech in VLSI at K L University, Vijayawada, AP, India. His research areas of interests are Digital VLSI Design, Low Power VLSI Design.



Pillem. Ramesh was born at mittagudem village, vissannapeta mandal, Krishna district, Andhra Pradesh state, India. He completed Post Graduation in VLSI System Design from SITAMS, Chittoor, JNTU, Hyderabad. Presently he is working as Asst.Professor in K. L. University. His interested areas are Analog VLSI circuits NANO CMOS and Semiconductor Devices