Game Implementation Using FPGA

Mini Project Report

Submitted in partial fulfillment of the requirements

for

Semester V in Bachelor of Technology (ECE)

Supervisor

Dr. Manish Goswami

Submitted by

Patlola Praveen(IEC2011025)
M Santhosh Mohan(IEC2011029)

Indian Institute of Information Technology - Allahabad

Allahabad – 211012

Certificate

This is to certify that the project report entitled "**Game Implementation Using FPGA**", is being submitted in fulfillment of Mini Project for the academic session July 2013- December 2013. The declaration made by the candidates is true to the best of my knowledge.

Date:

_____

Dr. Manish Goswami
Faculty
IIIT Allahabad
Allahabad

Declaration

We declare that this written submission represents our ideas and reference has been quoted where others' ideas or words have been included. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Date:

_____ _____

M Santhosh Mohan
IEC2011029

Patlola Praveen Reddy
IEC2011025

Acknowledgements

We would like to express our gratitude to our project mentor Dr. Manish Goswami. We are privileged to experience a sustained enthusiasm and involved interest from his side. This fueled our interest even further and encouraged us to boldly step into what was a totally dark and unexplored expanse before us.

We would like to thank our friends and elders for motivating us at every piece of work we picked up. Last but not least, we would like to thank the IIIT-A faculty members and the Institute, in general, for extending a helping hand at every juncture of need.

Table of Contents

Abstract

Field Programmable Gate-Array (FPGA) technology is gaining popularity among Application-Specific Integrated Circuit (ASIC) designers. Ease of development and maintenance makes FPGAs an attractive solution to many speed and efficiency-critical applications. The purpose of this project is to explore the world of FPGAs by implementing an arcade game on top of a VGA driver. The project was implemented on Xilinx Spartan-3E development board using Verilog hardware description language.

The project was started by learning Verilog as well as familiarising with the Spartan-3E development board and Xilinx ISE WebPACK design software. A number of simple applications were developed in order to comfortably proceed on to investigation of working principles of a VGA driver. It turned out that the synchronisation logic was rather trivial and that the custom implementation would not add much value to the project.

Verilog implementation of the classic Pong game was the first major task of the project. In this Game player tries to hit the moving ball with the paddle which will be under his control. A number of basic features, such as acceleration of the ball, textual information display, have been implemented. All the graphic elements will be designed from scratch.

Chapter 1
Introduction

This chapter introduces the most important concepts of relevance to the project. It clears the existence of a project of this type including the reasons for picking this topic.

1.1 Objective

The aim of this project was to explore the capabilities of modern programmable logic devices while getting hands-on experience of FPGA development.

The above definition is rather abstract and can be hard to assess. For this purpose, a number of objectives were set. Below, is the amended list of objectives.

• Learn about FPGAs, development tools and Verilog, start programming.

• Produce a simple application , that uses a VGA driver to display graphics on a computer monitor.

• Create a game .

Milestones were created to give a rough estimate of duration of the project.

1.2 Motivation

Good motivation for learning is a powerful part of any successful curriculum. Especially in project courses where we design and build an artifact, the time spent working on the project, and the depth of knowledge that comes out of the project, are dramatically increased if there is some strong internal motivation to go beyond the basics and delve into the project details. In particular we focus on a course where we learn about computer design by designing and building a computer from scratch on a Field Programmable Gate Array (FPGA).

Best way to learn about computer design is to design a computer. For us the opportunity to develop interactive games on a computer they designed is a strong motivator.

Easy way to get familiarize with the FPGA kit, Verilog coding , Xilinx WEBPack design software and finally with the I/O devices.

Chapter 2
Literature Survey

This chapter aims to bring forth some of the existing work in the area of chess programming, studying them in more detail and in proper context and then drawing conclusions about existing and required features.

2.1 Field-Programmable Gate-array(FPGA) Architecture – An Overview

FPGA - is an acronym for Field Programmable Gate Array. It belongs to a class of user programmable digital devices called Programmable Logic Devices (PLD's) . A programmable logic device is an integrated circuit that enables the user to configure it in many ways,enabling the implementation of various digital logic functions, of varying sizes and complexities. PLD's can be classified into various categories :

1. Simple programmable logic devices (SPLD)
   (a) Programmable logic array (PLA) : A programmable logic array is an integrated circuit that consists two levels of programmable logic ; an AND plane and an OR plane.
   (b) Programmable array logic (PAL): A PAL is an integrated circuit that contains a fixed OR plane followed by a programmable AND plane.
2. Complex Programmable Logic Device (CPLD)
3. Field Programmable Gate Array (FPGA)

2.2 Field-Programmable Gate-array (FPGA)

FPGAs are modern programmable logic devices that can be configured to perform any logic operation. An FPGA typically contains a matrix of programmable elements, also known as, Configurable Logic Blocks (CLBs). CLBs contain Look-Up Tables (LUTs), that can be used as logic or storage elements. The configuration data is stored in the memory.

Spartan-3E series FPGAs, used in this project contain the following structures :

- **Configurable Logic Block** – logic and basic storage elements are implemented using the Look-UP Tables (LUTs).
- **Input/Output Block (IOB)** – control the data flow between the I/O pins and internal logic of the device. LVTTL and LVCMOS logic standards are supported among others.
- **Block RAM** – memory used for data storage. Organized as 216kb dual - port blocks.
- **Digital clock manager block** – provides management for clock signals.

2.3 Hardware Description Language :

Verilog, standardized as IEEE 1364, is a hardware description language (HDL) used to model electronic systems. It is most commonly used in the design and verification of digital circuits at the register-transfer level of abstraction. It is also used in the verification of analog circuits and mixed-signal circuits.

Verilog is case-sensitive and has a basic preprocessor (though less sophisticated than that of ANSI C/C++). Its control flow keywords (if/else, for, while, case, etc.) are equivalent, and its operator precedence is compatible. Syntactic differences include variable declaration (Verilog requires bit-widths on net/reg types [clarification needed]), demarcation of procedural blocks (begin/end instead of curly braces {}), and many other minor differences.

Definition of syntax:
The definition of constants in Verilog supports the addition of a width parameter. The basic syntax is:
<Width in bits>'<base letter><number>

Digital systems theory

Verilog is a good example of practical application of digital circuit theory. Many of the basic concepts, such as gate and register-transfer level design, combinational logic and finite state machines have been used throughout the project.

Other HDLs :

As with the FPGAs, there are two predominant hardware description languages – VHDL and Verilog **.**

VHDL

"VHDL is a language for describing digital electronic systems. It arose out of the United States government's Very High Speed Integrated Circuit (VHSIC) program. In the course of this program, it became clear that there was a need for a standard language for describing the structure and function of Integrated Circuits (ICs).

VHDL supports many built-in and user-defined data types. Some of most often used are

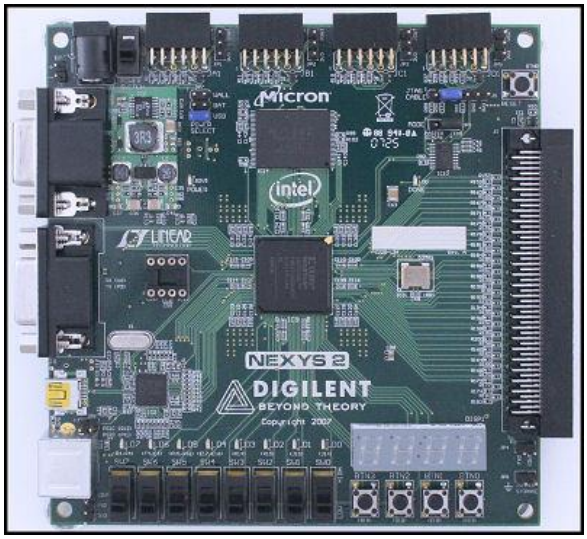
- std_logic (single bit)
- std_logic_vector (bit vector)
- numerical types such as integer
- arrays, enumerated lists, etc.

2.3 Development board

The Nexys-2 is a powerful digital system design platform built around a Xilinx Spartan 3E FPGA. With 16Mbytes of fast SDRAM and 16Mbytes of Flash ROM, the Nexys-2 is ideally suited to embedded processors like Xilinx's 32-bit RISC Microblaze. The on-board high-speed USB2 port, together with a collection of I/O devices, data ports, and expansion connectors, allow a wide range of designs to be completed without the need for any additional components.

Specifications :-

- Xilinx Spartan-3E FPGA 1200K gate
- USB2 port providing board power, device configuration, and high-speed data transfers
- Works with ISE/Webpack and EDK
- 16MB fast Micron PSDRAM
- 16MB Intel StrataFlash Flash R
- Xilinx Platform Flash ROM
- High-efficiency switching power supplies (good for battery-powered applications
- 50MHz oscillator, plus a socket for a second oscillator
- 75 FPGA I/O's routed to expansion connectors (one high-speed Hirose FX2 connector with 43 signals and four 2x6 Pmod connectors)
- All I/O signals are ESD and short-circuit protected, ensuring a long operating life in any environment.
- On-board I/O includes eight LEDs, four-digit seven-segment display, four pushbuttons, eight slide switches
- Ships in a DVD case with a high-speed USB2 cable



Spartan 3E FPGA kit

2.4 VGA

Hardware

VGA is an analogue video standard, that is mostly used in personal computers. VGA can also refer to a piece of display hardware developed by IBM (Video Graphics Array) or a display mode, that uses 640 x 480 pixels resolution. VGA connector uses a total of 15 pins, but only 5 signals are needed for operation:

- HSYNC – horizontal synchronization signal. This signal controls the horizontal position of the active pixel

- VSYNC – vertical synchronization signal. This signal controls vertical position of the active pixel. VSYNC rate can also be referred to as a refresh rate (i.e. number of times per second the screen is redrawn)

- RED – red colour channel

- GREEN – green colour channel

- BLUE – blue colour channel

Other pins include ground, return paths and I2C clock/data or are reserved [5].

HSYNC and VSYNC are TTL signals, so logic one is represented by 5V and logic zero is represented by 0V [5].

RED, GREEN and BLUE signals are analogue. The maximum voltage that can be used is 0.7V and will result in full intensity of that colour .
Xilinx Spartan-3E board is only capable of producing eight colours (3-bits) as a digital-to analog converter (DAC) is not used. The board uses 270Ω resistors, which form a potential divider with internal 75Ω termination. This divider scales the 3.3V signal from FPGA to required 0.7V Back Porch) in Figure 2.2..

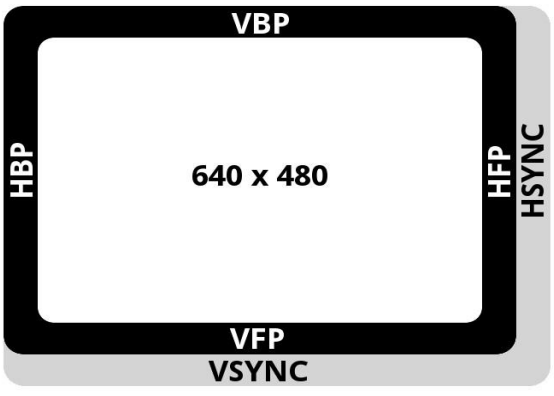


FIGURE : PARTS OF A SCREEN

Timing

Pixels on the screen are drawn in sequence, one by one. Rows are arranged top to bottom, and columns go from left to right. The row and column addresses are constantly incremented thus changing the position of currently drawn pixel. Synchronisation signals are used to tell the monitor to return the pixel back to the first row (VSYNC) or the first column (HSYNC).
Sequence and duration of these signals are discussed below.

Visible part of the screen is shown as the white area. It has a resolution of 640 by 480 pixels.
The black and grey borders denote parts of the screen that are not visible, but required for synchronization. With these parts, the total width of the screen is 800 pixels, and the total height is 524 pixels. Below is the short description of each part of the screen, followed by a table of dimensions.

Active video : This is the visible part of the screen, video output is enabled.

Front porch : When the trace reaches the end of the visible part of the screen, the video output is disabled. These areas are denoted as VPF (Vertical Front Porch) and HPF (Horizontal Front Porch) in Figure.

Sync pulse : In case of HSYNC, the trace goes back to column zero. If pulse is VSYNC, the trace goes back to row zero. This part is also known as the retrace period.

Back porch : This is the part that goes before the active video starts. These areas are denoted as VBF (Vertical Back Porch) and HBF (Horizontal Back Porch) in Figure.

Table-a lists the dimensions for each part of the screen. These numbers are only valid for resolution of 640 x 480 and refresh rate of 60Hz. Other modes are described in the source [6].

25MHz pixel clock	Active video	Front porch	Sync pulse	Back porch
Horizontal	640	16	96	48
Vertical	480	11	2	31

Table – a VGA Timings

Chapter 3

Proposed Approach

This chapter specifies the basic idea about how the project is implemented and how we proceed further.

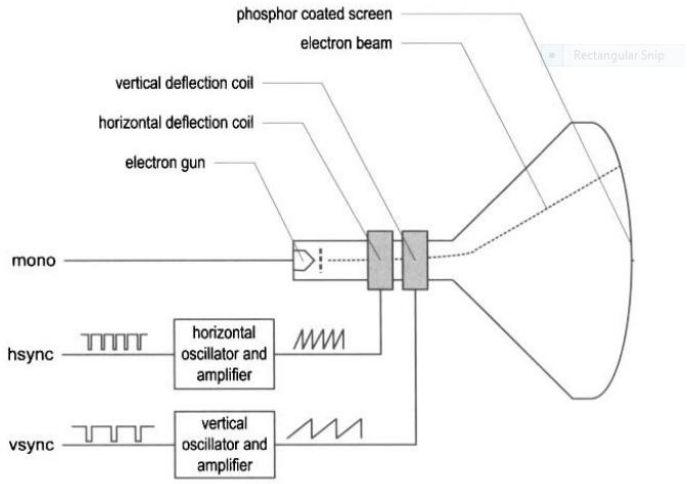
3.1 Introduction

VGA (video graphics array) is a video display standard introduced in the late 1980s in IBM PCs and is widely supported by PC graphics hardware and monitors. We discuss the design of a basic eight-color 640-by-480 resolution interface for CRT (cathode ray tube) monitors in this book. CRT synchronization and basic graphic processing are examined.

3.2 Basic operation of a CRT

The conceptual sketch of a monochrome CRT monitor is shown in Figure 13.1. The electron gun (cathode) generates a focused electron beam, which traverses a vacuum tube and eventually hits the phosphorescent screen. Light is emitted at the instant that electrons hit a phosphor dot on the screen. The intensity of the electron beam and the brightness of the dot are determined by the voltage level of the external video input signal, labeled mono in Figure 13.1. The mono signal is an analog signal whose voltage level is between 0 and 0.7 V.

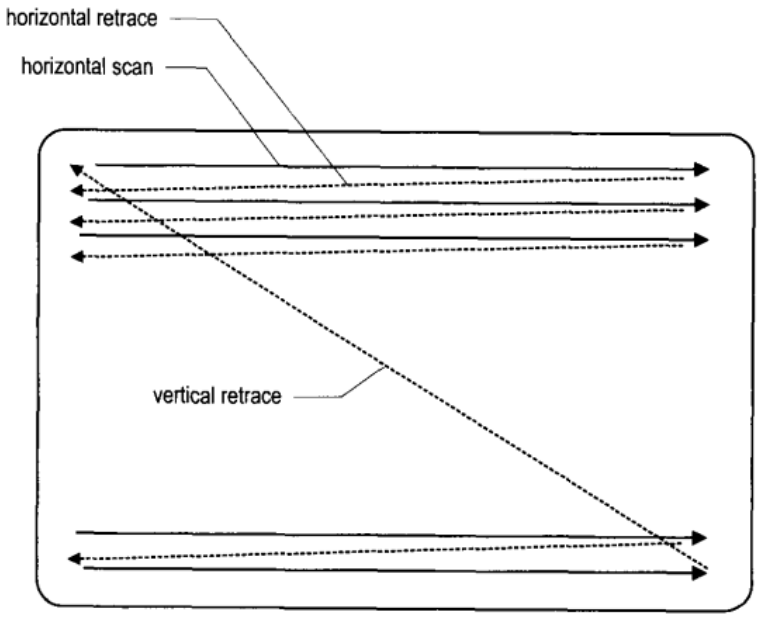
A vertical deflection coil and a horizontal deflection coil outside the tube produce mag- netic fields to control how the electron beam travels and to determine where on the screen the electrons hit.



Conceptual Diagram

Three-bit VGA color combinations

RED(R)	GREEN(G)	BLUE(B)	RESULTING COLOUR
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



CRT scanning pattern.

The monitor's internal oscillators and amplifiers generate sawtooth waveforms to control the two deflection coils. For example, the electron beam moves from the left edge to the right edge as the voltage applied to the horizontal deflection coil gradually increases. After reaching the right edge, the beam returns rapidly to the left edge (i.e., retraces) when the voltage changes to 0. The relationship between the sawtooth waveform and the scan is shown in Figure 13.4.

Two external synchronization signals, hsync and vsync, control generation of the sawtooth waveforms. These signals are digital signals. The relationship between the hsync signal and the horizontal sawtooth is also shown in Figure.

Note that the " 1 " and "0" periods of the hsync signal correspond to the rising and falling ramps of the sawtooth waveform. The basic operation of a color CRT is similar except that it has three electron beams, which are projected to the red, green, and blue phosphor dots on the screen. The three dots are combined to form a pixel. We can adjust the voltage levels of the three video input signals to obtain the desired pixel color.

3.3 Timing calculation of VGA synchronization signals

As mentioned earlier, we assume that the pixel rate is 25 MHz. It is determined by three parameters:

- P : the number of pixels in a horizontal scan line. For 640-by-480 resolution, it is

  P = 800 (pixels/line)

- L : the number of lines in a screen (i.e., a vertical scan). For 640-by-480 resolution, it is

  l = 525(lines/screens)

- S : the number of screens per second. For flickering-free operation, we can set it to

  S = 60 (screens/second)

The s parameter specifies how fast the screen should be refreshed. For a human eye, the refresh rate must be at least 30 screens per second to make the motion appear to be continuous. To reduce flickering, the monitor usually has a much higher rate, such as the 60 screens per second specification above. The pixel rate can be calculated by the three parameters :

pixel rate = p * 1 * s = 25M (pixels/second)

The pixel rate for other resolutions and refresh rates can be calculated in a similar fashion. Clearly, the rate increases as the resolution and refresh rate grow.

Chapter 4

Hardware and Software Requirements

4.1 Hardware Required

- FPGA Kit (SPARTAN 3E).
- Monitor
- Power Supply Cable
- VGA Connector

4.2 Software Required

- ISE WebPACK Design Suite Xilinx
    - ISE Design Tools
    - Plan Ahead
- Digilent Adept 2.3

Chapter – 5

Activity Time Chart

BEFORE MID-SEM :

03/08/2013 – Got familiar with the FPGA kit.

10/08/2013 - Implemented LOGIC gates.

24/08/2013 - Implemented ADDER.

07/09/2013 - Implemented 4 BIT COUNTER.

14/09/2013 - Familarised with the VGA monitor such as timing, horizontal , vertical Synchronization signal and RGB colours.

05/10/2013 - Implemented the synchronization between the Monitor and the kit and displayed different colours.

AFTER MID-SEM :

26/10/2013 - Implementation of frame on the monitor.

2/11/2013 - Implementation of paddle and ball.

09/11/2013 - Implementation of Game Logic.

16/11/2013 - Rechecking the total code and tries to improve better.

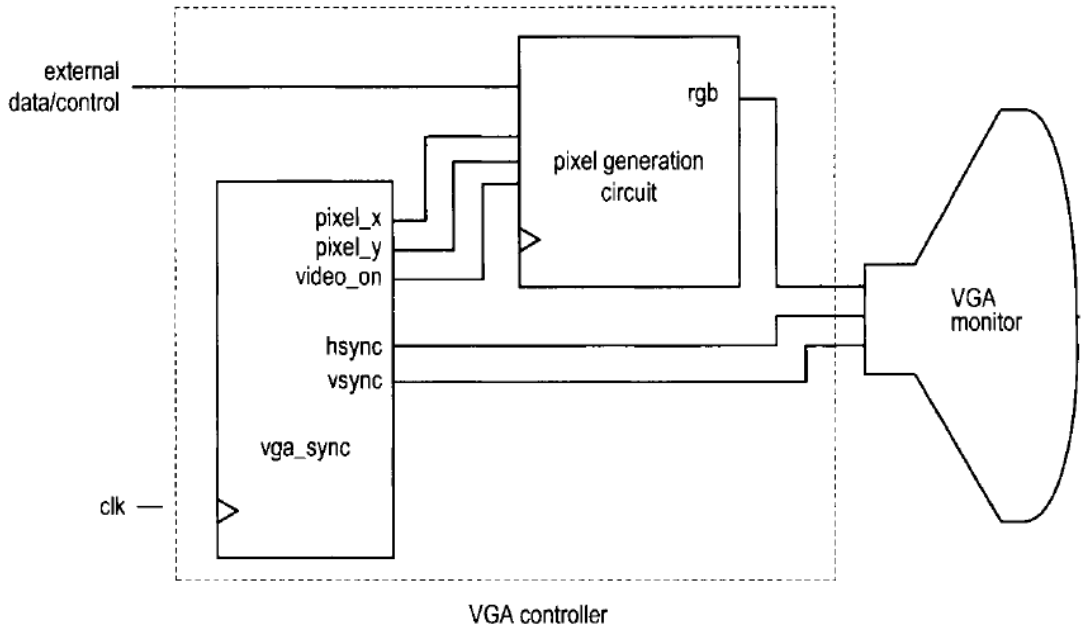
Chapter 6
Work Done Till MID SEM

This chapter specifies about how much amount of work has completed on the project till mid sem.

6.1 Generating VGA Signals :

The VGA port has five active signals, including the horizontal and vertical synchronization signals, hsync and vsync , and three video signals for the red, green, and blue beams. It is physically connected to a 15-pin D-subminiature connector. A video signal is an analog signal and the video controller uses a digital-to-analog converter to convert the digital output to the desired analog level. If a video signal is represented by an N-bit word, it can be converted to 2N analog levels. The three video signals can generate 8N different colors. This is also known as 3N-bit color since a color is defined by 3 N bits. In the S3 board, a 1 -bit word is used for each video signal and this leads to only eight possible colors. If we use the same 1 -bit signal to drive the video signals, they become either "000" or " 1 1 1" and the monitor functions as a black-and-white monochrome monitor.

A video controller generates the synchronization signals and outputs data pixels serially. A simplified block diagram of a VGA controller is shown in Figure. It contains a synchronization circuit, labeled vga-sync, and a pixel generation circuit.



Simplified Block diagram of a VGA controller.

The vga-sync circuit generates timing and synchronization signals. The hsync and vsync signals are connected to the VGA port to control the horizontal and vertical scans of the monitor. The two signals are decoded from the internal counters, whose outputs are the pixel-x and pixel-y signals. The pixel-x and pixel-y signals indicate the relative positions of the scans and essentially specify the location of the current pixel. The vga-sync circuit also generates the video-on signal to indicate whether to enable or disable the display.

4.2 VGA SYNCHRONIZATION :

The video synchronization circuit generates the hsync signal, which specifies the required time to traverse (scan) a row, and the vsync signal, which specifies the required time to traverse (scan) the entire screen. Subsequent discussions are based on a 640-by-480 VGA screen with a 25-MHzpixel rate, which means that 25M pixels are processed in a second. Note that this resolution is also know as the VGA mode.

4.3 HDL IMPLEMENTATION :

The function of the vga-sync circuit is discussed in Section 13.1.3. If the frequency of the system clock is 25 MHz, the circuit can be implemented by two special counters: a mod-800 counter to keep track of the horizontal scan and a mod-525 counter to keep track of the vertical scan. Since our designs generally use the 50-MHz oscillator of the prototyping board, the system clock rate is twice the pixel rate. Instead of creating a separate 25-MHz clock domain and violating the synchronous design methodology, we can generate a 25-MHz enable tick to enable or pause the counting. The tick is also routed to the p-tick port as an output signal to coordinate operation of the pixel generation circuit.

Implemented code :-

In order to display simple colors by switching the colors with the switches on the board.

Code :

←----------→

```
`timescale 1ns / 1ps
	module vgacolour(
	   input wire clk,
	   input wire reset,
	   input wire [2:0] sw,
	   output wire hsync,
	   output wire vsync,
	   output wire [2:0] rgb_red,
	         output wire [2:0] rgb_green,
	         output wire [1:0] rgb_blue
	   );
	         reg [2:0] rgb_reg; // signal declaration
	         wire video_on;
	// instantiate v sync circuit
	         vga_sync vsync_unit( .clk(clk) , .reset (reset), .hsync (hsync), .vsync(vsync),
	         .video_on(video_on), .p_tick(), .pixel_x(), .pixel_y( ));
```

```
        always @( posedge clk , posedge reset)
                if (reset)
                        rgb_reg <= 0;
                else
                        rgb_reg <= sw;
    //output
        assign rgb_red = (video_on) ? rgb_reg[0] : 3'b0;
        assign rgb_green = (video_on) ? rgb_reg[1]: 3'b0;
        assign rgb_blue = (video_on) ? rgb_reg[2] : 2'b0;
endmodule
```

←----------→

Here the vga_sync unit code is written separately and is not reported here.

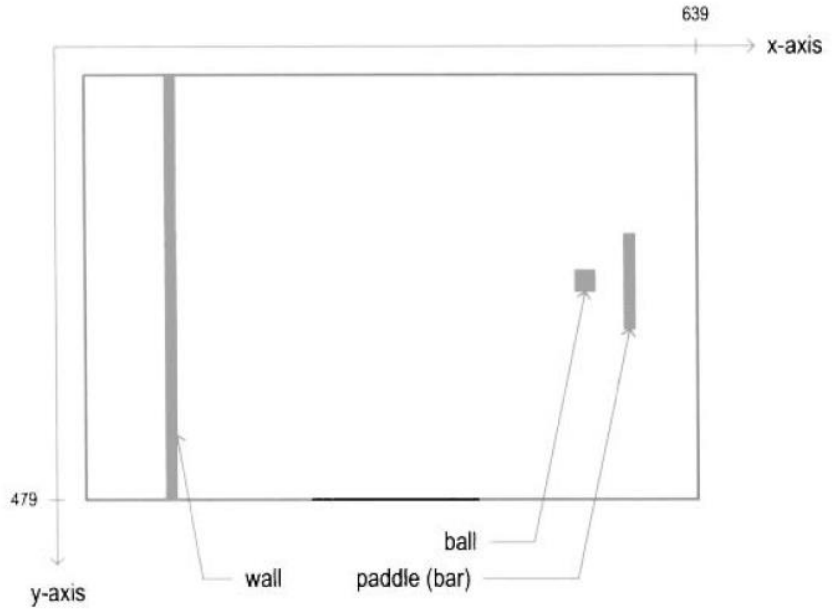
Results :- Snap shots

Chapter 7

Work To Be Done After MID SEM

- Creating a Frame on the monitor.
- Creating Paddle and Ball on the monitor along with the frame.
- Familarisation of Designing Game Logic.
- Implementing Game Logic.

Expected OUTPUT :



Expected output on the VGA monitor

Chapter 8

References

- Verilog HDL- A Guide to Digital Design and Synthesis IEEE 1364-2001 Compliant By Samir Palnitkar.
- FPGA Prototyping –A Book by PONG P.Chu.
- CAST. High Resolution Display Controller - Datasheet. Technical report, October 2010.
- Xilinx. Spartan-3E FPGA Family: Data Sheet. Technical report, August 2009.
- Xilinx WebPACK Design Tools 14.2 :User Manual.
- Richard Herveille. VGA/LCD Controller v2.0. Technical report, 2009.
- Guohui Wang, Yong Guan, and Yan Zhang. Designing of VGA Character String Display Module Based on FPGA. In International Symposium on Intelligent Ubiquitous Computing and Education(IUCE), pages 499-502, 2009.

Chapter 9

Suggestions and Comments