COE 405
Design & Modeling of Digital Systems

Course Project#2 – Term 131
COE PONG GAME[i]
Due on: Tuesday Dec. 24, 2013

In this project, you are required to design and implement a COE pong game. The game is based on a single player. The user will be able to control a paddle on the left side of the screen, and use the paddle to hit and direct a ball to the wall on the right side of the screen. The ball should be able to bounce freely off of the top and bottom borders, and should bounce off the wall back toward the left side, so the user may hit the ball again. Should the user miss the ball with the paddle, the ball will continue and freeze once it reaches the very edge of the left side, signaling the game is over.

In the beginning of the game, display the title "COE PONG GAME" on the LCD screen in the first line. In the second line, display the score of the player initialized with 0. When the game is over, display in the first line "COE PONG GAME OVER" and in the second line, display the score.

If the user plays well and does not miss the ball, his score should be incremented by 1 for every 5 seconds passing without missing the ball. The score is updated continuously on the LCD screen.

Your overall system should have several user inputs:
1) A **reset** button, which the user presses to restart the game, either in the middle of a game or when the game is over.
2) **Rotary knob** which the user uses to control the paddle. (Left rotation moves the paddle up while right rotation moves it down)
3) A 4-bit **speed** switch (2-bits for x and 2-bits for y) which controls the initial speed of the ball during game play. The speed sets the initial velocity of the ball in units of pixels per frame (both in the x and y directions). You are free to choose the values. You may use the following values: 1, 3, 6, and 9.

The playing field is 640x480 pixels. Display a border around the top, bottom and right screen sides. You need also to display the text "COE" in the middle of the screen. You can make your own assumptions about the sizes and colors of the border, paddle, ball and "COE" text. For example, the border could be 8 lines. The ball could be an 8x8 square. The paddle could be an 8x64 rectangle. The position of the paddle could be 28 pixels from the left side. The next figure shows sample pixel specifications.

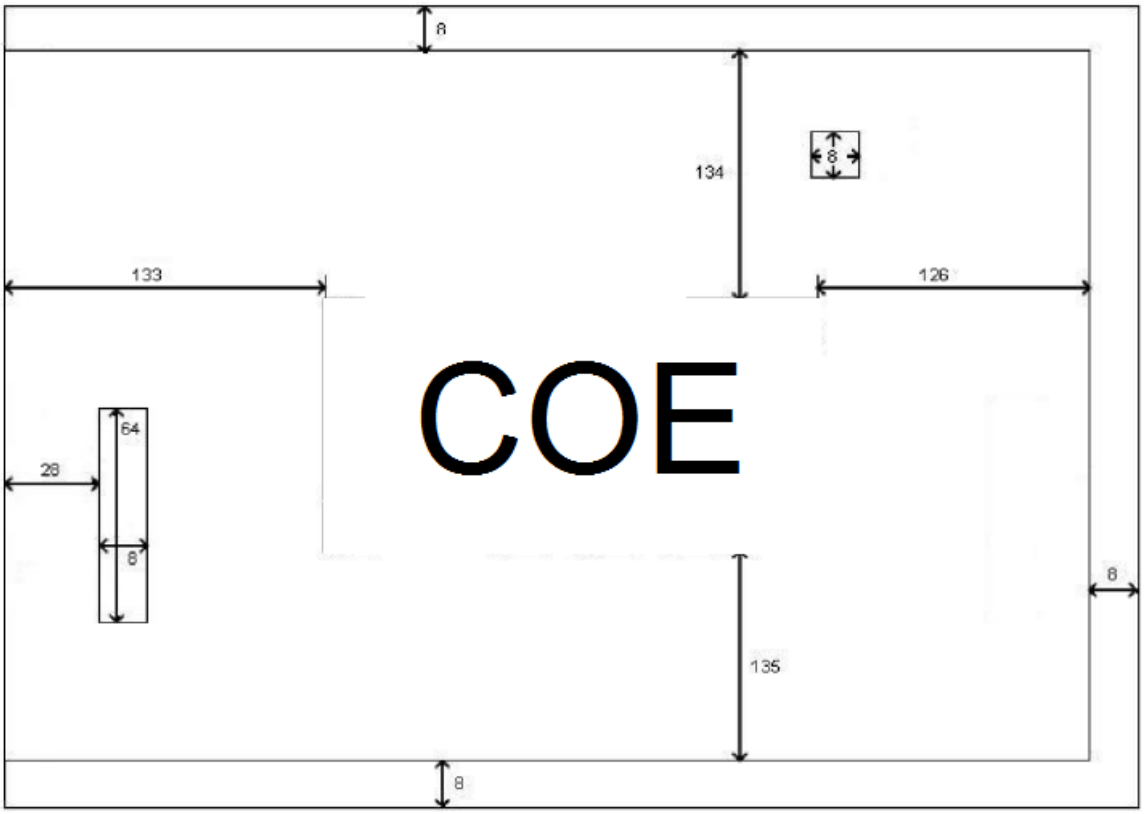

Figure 1 Sample Pixel Specifications (Not to Scale),

Game Mechanics

The goal of this game is to keep the ball from hitting the left edge of the screen by deflecting it with the paddle. Because there is no second player on the right side, the ball will inevitably return when it bounces off of the stationary wall, and the user must once again defend the goal. The game is over if the paddle misses the ball and the ball reaches the left edge of the screen.

When the ball hits the walls along the top, bottom, and right edges of the screen, the ball should rebound as expected. The magnitudes of the horizontal and vertical velocity components remain constant, and only the sign of one component is changed, as shown in Figure 2(a). (Note that, in accordance with our suggested coordinate system, a positive y-velocity corresponds to the ball moving downward.) The physics of collisions between the ball and the paddle, however, are tweaked a bit in order to give the player some control over balls' trajectory. The paddle is divided into four regions. Depending on which region the ball collides with, the ball rebounds with some bias added to its velocity, as shown in Figure 2(b).

The top half of the paddle deflects the ball with some upward bias, and the bottom half deflects it with some downward bias. As side effect of the added bias, the speed of the ball will increase slightly with every collision with the paddle, making the game gradually harder. Feel free to tweak the bias velocities, and even the number or configuration of the paddle regions as you see fit. (Specify what you did in the report.)

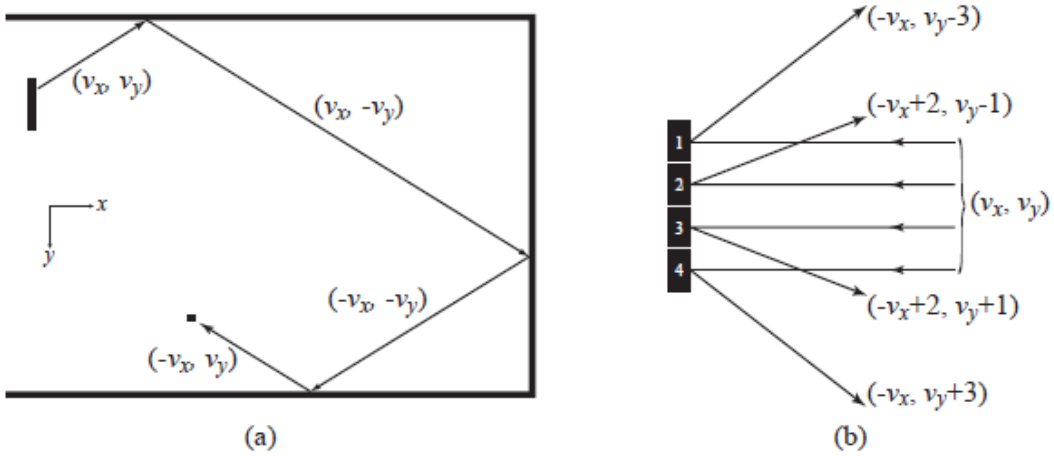

Figure 2 The physics of Pong: (a) collisions with the walls, (b) collisions with the paddle.

Generate three different sounds to distinguish between when the ball hits the paddle, when it hits any of the top, bottom and right sides and when it is lost indicating a game over.

You can add any additional features to the game to make it more fun. Interesting features added will grant you a 1% bonus.

VGA Display

The Spartan®-3A/3AN FPGA Starter Kit board includes a VGA display port via a standard high-density HD-DB15 female connector. As shown in Figure 3, the VGA connector is the left-most connector along the top of the board.

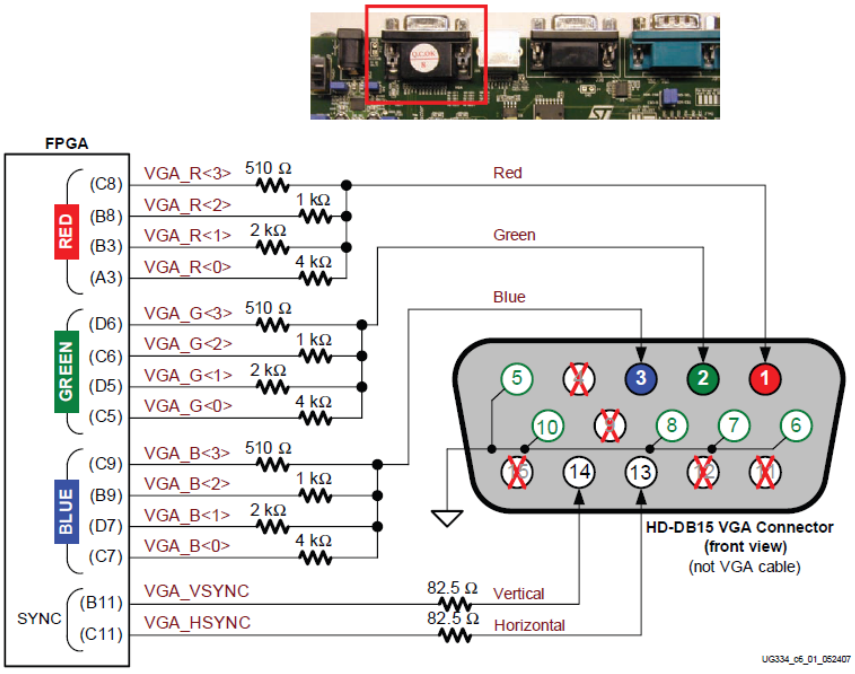

Figure 3 VGA Connections from the Starter Kit Board.

The FPGA directly drives the five VGA signals via resistors. Each red, green, and blue signal has four outputs from the FPGA that feed a resistor-divider tree. This approach provides 4-bit resolution per color, generating 12-bit color, or 4,096 possible colors. Drive the VGA_R[3:0], VGA_G[3:0], and VGA_B[3:0] signals High or Low to generate the desired colors. For simplicity, the FPGA application can also treat the VGA port as a three-bit interface by driving all four color outputs with the same digital value. The corresponding eight basic color values are shown in Table 6-1.

Table 1Example Display Color Codes.

VGA_R[3:0]	VGA_G[3:0]	VGA_B[4:0]	Resulting Color
0000	0000	0000	Black
0000	0000	1111	Blue
0000	1111	0000	Green
0000	1111	1111	Cyan
1111	0000	0000	Red
1111	0000	1111	Magenta
1111	1111	0000	Yellow
1111	1111	1111	White

VGA signal timing is specified, published, copyrighted, and sold by the Video Electronics Standards Association (VESA). The following VGA system and timing information is provided as an example of how the FPGA might drive the VGA monitor in 640 by 480 mode.

Within a CRT display, current waveforms pass through the coils to produce magnetic fields that deflect electron beams to transverse the display surface in a raster pattern, horizontally from left to right and vertically from top to bottom. As shown in Figure 4, information is only displayed when the beam is moving in the forward direction—left to right and top to bottom—and not during the time the beam returns back to the left or top edge of the display. Much of the potential display time is therefore lost in blanking periods when the beam is reset and stabilized to begin a new horizontal or vertical display pass.

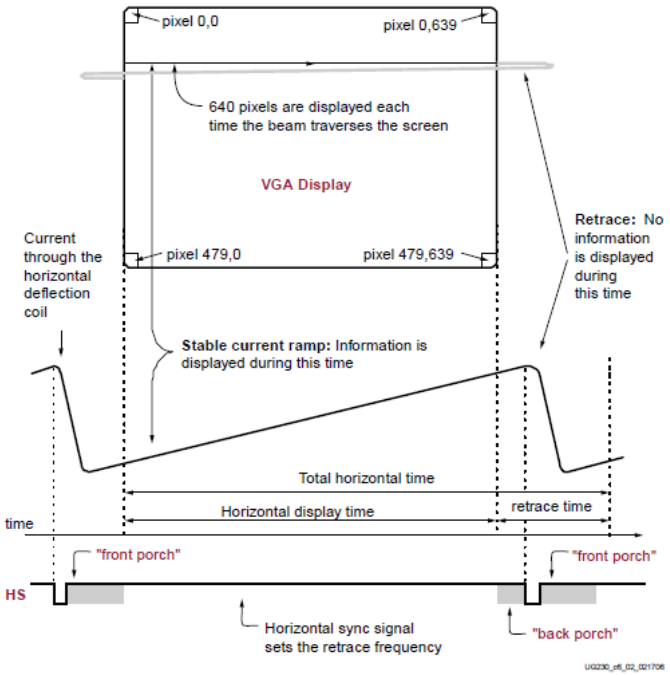

Figure 4 CRT Display Timing Example.

The signal timings in Table 6-2 are derived for a 640-pixel by 480-row display using a 25 MHz pixel clock and 60 Hz ± 1 refresh. Figure 5 shows the relation between each of the timing symbols. The timing for the sync pulse width (TPW) and front and back porch intervals (TFP and TBP) is based on observations from various VGA displays. The front and back porch intervals are the pre- and post-sync pulse times. Information cannot be displayed during these times.

Table 2 640x480 Mode VGA Timing.

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

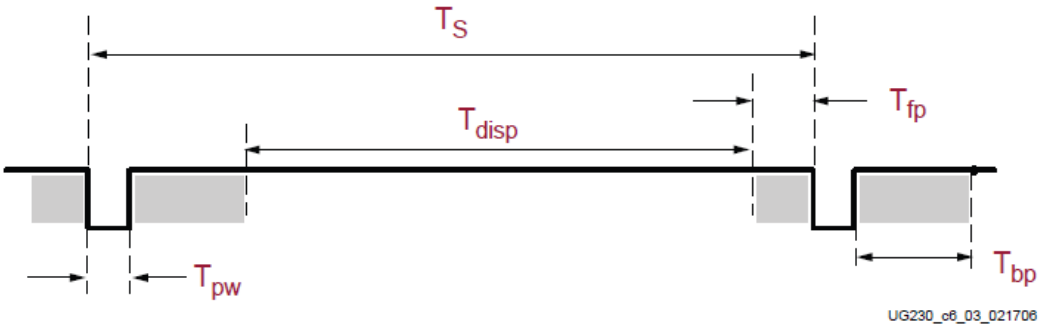

Figure 5 VGA Control Timing.

Project Report

The report document must contain sections highlighting the following:

Cover Page/Abstract

Include a cover page with a title, your names, the name of your instructor, the course name, and the date.

Introduction

Give a brief description of the problem and a block diagram of your system.

Module Description/Implementation

(a) Describe your datapath and control FSMs

• Include ASMD diagrams for your FSMs.
• Describe (in words) the operation of your FSMs. Include your commented Verilog code.

(b) Detailed block diagrams of logic implemented in your FPGA

Try to make a reasonable compromise between legibility and detail. You do not have to draw detailed equivalent circuits to describe the contents of the FPGA. You should include a detailed block diagram showing major functional units with annotated widths. Each block should be accompanied by a paragraph describing the function of the FPGA circuitry.

Testing/Debugging

Describe how you tested your digital system. Provide a description of the design methodology you used in the creation of your digital system, and how you planned on testing each block in the design stage and how you actually ended up testing it. Be sure to include specific details.

Teamwork

This project is a team work project with two to three students per team. Make sure to write the names of all the group members on the project report title page. Each group should assign a group leader that leads the conduction of the project, divide the project tasks among the team members. Project tasks should be divided among the group members so that each group member contributes equally in the project and everyone is involved in all activities. Clearly show the work done by each group member. Students who **help** other team members should mention that to earn credit for that.

Conclusion

Summarize your learning experience from this project and what you think are the important concepts to take away from the design of this digital system.

Submission Guidelines

All submissions will be done through blackboard.

Attach one zip file containing all Verilog and constraint files used in your design, a video demo of your project, as well as the report document. Upload your video demo in youtube or other sites and provide a link in the report.

Submit also a hard copy of the report during the class lecture.

Grading policy:

Grading Criteria	Mark
Demonstration of correct functionality of project on FPGA	70
Project Documentation and Report Organization	30
Total	100

[i] Courtesy of MIT PONG GAME