Quetion: TA

Ryan Shill

Question: brief summary of the functionality

We made the game Breakout. Basically, like everyone knows, there is one paddle that moves back and forth to send up a ball to break blocks at the top of the screen. We used the VGA display to show the ongoing game round and buttons 3 and 2 on the FPGA board to control the paddle. The buttons 0 and 1 are start and reset buttons respectively. The seven segment display keeps the score.

Question: Provide a detailed description…

Our project replicates the common game breakout. When the program starts up, it enters into an idle state, where the game is ready to begin when the player is ready. Once ready, the player can hit button 0 to begin. Button 3 will move the paddle to the left, and button 2 will move the paddle to the right. The object of the game is to eliminate all of the bricks at the top of the screen to win. Button 1 will reset the game if the user wants to start over for any reason.

The top level design has the following inputs: clk and btn (3 bits). It also has the following outputs: red\_out (3 bits), green\_out (3 bits), blue\_out (2 bits), hs\_out, vs\_out, seg (7 bits), dp, and  an (4 bits). Unless otherwise specified, assume the the input or output signal is 1 bit in length.

The direction of the ball is determined by where the ball hits the paddle. We divided our paddle into six parts of equal size. If the right most side is hit, the ball will project in an upward direction at a 30 degree angle. If it hits the next section of the paddle, it will project at a 45 degree angle. It will continue this paddle of projections at 60 degrees, 120 degrees, 135 degrees, and 150 degrees. The direction of the current paddle movement does not change the direction of the ball, only where the ball collides with the paddle.

The bricks form a 10x7 layout at the top of the screen. Each brick is 58 x 16 pixels in area, and each total vertical and horizontal distance between bricks is 4 pixels apart. The bricks are divided by color, where the seven rows of bricks are determined by the colors red, orange, yellow, green, blue, indigo, and violet. The higher the bricks are, the more point value that the brick is worth. Violet bricks are worth 1 point, indigo bricks are worth 2, blue bricks are worth 4, green bricks are worth 8, yellow bricks are worth 16, orange bricks are worth 32, and finally the red bricks are worth 64 points.

Score is kept on the seven segment display during the game. Once the game ends, the game enters into a freeze or idle state, where the game resets and can restart when the user is ready. We used the VGA display as a layout from previous labs to get the display to function properly.

Question: Describe how you are going to divide…

Because we will be working in a group of 2, we will be dividing the project into two different parts, and then integrating the two parts together. For the first part, one of us will create the paddle, the paddle movements, a state machine to determine the current directions and next direction of ball movements, and the reset and start signals. For the other part, the other individual will create the bricks, the brick layout, the color of the bricks, make the bricks disappear when the ball makes contact, and determine the new direction of the ball upon impact.

Question: level of effort

50% : 50%. We only worked on this project when we met together and spent equal time on it. So we could say that we contributed on this project equally. We divided up to two different parts of the project and both had good amount of work to do. Jae took care of paddle bouncing back the ball to 6 different angles and Bryant took care of blocks. We enjoyed.

Question: Summarize some of the positive and negative…

I enjoyed working a partner on this lab. Because it was open to work with one another, we were able to assign individual parts to a specific individual, and then work together on combining their efforts, where each individual would be responsible for debugging his particular errors when things would not work as planned. I found this very beneficial because we didn’t follow the construction worker phenomena, where one is working and others are supervising. We were available to ask each other for various ideas, input, suggestions, and even assistance when there was something that we did not understand. Normally, team projects are a drag as we often times feel that assignments are not split apart equally, but in these circumstances, working with a partner was a great experience.

Question: changes to the original specification

Our initial design was to make blocks with several different colors indicating that how many times they have to be hit to disappear. Also, we didn’t think about making the ball move to 6 different angles depending on where on paddle it hits at first. However, as we move along, we decided to not make the blocks to change colors because we thought it would take more effort and time than we could afford. Instead, we added the feature of 6 different angles to give more control of the game to the user.

READ ME:

----------------------

-------BREAKOUT-------

----------------------

Authors: Bryant Baird and Jae Lee

Course: ECEn 320

Semester: Winter 2014

Instructor: Dr. Wirthlin

University: BYU

Summary: This project is a replication of the game BREAKOUT. One must move the bottom paddle to continually keep the ball bouncing until all of the upper bricks disappear.

Instructions: To play this game, you will only need to use 4 buttons on the FPGA controller.

Button 3: This button will move the paddle to the left

Button 2: This button will move the paddle to the right

Button 1: This button will reset the game

Button 0: This button will start the game when the player is ready

Score will be kept on the seven segment display controller, displayed in hexadecimal. Each brick color is assigned the following point values:

RED: 1 point

ORANGE: 2 points

YELLOW: 4 points

GREEN: 8 points

BLUE: 16 points

INDIGO: 32 points

VIOLET: 64 points

Future Work: Obviously, this project is not up to par with the original game of breakout. Future additions that we would make would be to:

Include lives: Have anywhere from 3-5 lives

Bonus bricks: Change speed of ball, adjust width of paddle

Brick color: The higher bricks can change color rather than disappearing

Project Build Options: There are no special options or build procedures needed to synthesize the circuit, as no external equipment was used

READ ME:

----------------------

-------BREAKOUT-------

----------------------

Authors: Bryant Baird and Jae Lee

Course: ECEn 320

Semester: Winter 2014

Instructor: Dr. Wirthlin

University: BYU

Summary: This project is a replication of the game BREAKOUT. One must move the bottom paddle to continually keep the ball bouncing until all of the upper bricks disappear.

Instructions: To play this game, you will only need to use 4 buttons on the FPGA controller.

Button 3: This button will move the paddle to the left

Button 2: This button will move the paddle to the right

Button 1: This button will reset the game

Button 0: This button will start the game when the player is ready

Score will be kept on the seven segment display controller, displayed in hexadecimal. Each brick color is assigned the following point values:

RED: 1 point

ORANGE: 2 points

YELLOW: 4 points

GREEN: 8 points

BLUE: 16 points

INDIGO: 32 points

VIOLET: 64 points

Future Work: Obviously, this project is not up to par with the original game of breakout. Future additions that we would make would be to:

Include lives: Have anywhere from 3-5 lives

Bonus bricks: Change speed of ball, adjust width of paddle

Brick color: The higher bricks can change color rather than disappearing

Project Build Options: There are no special options or build procedures needed to synthesize the circuit, as no external equipment was used

Project Size:  1155 lines of VHDL code

Number of Slices:                      919  out of   4656    19%

Number of Slice Flip Flops:            248  out of   9312     2%

Number of 4 input LUTs:               1763  out of   9312    18%

Number of IOs:                          28