ECSE 323: LAB REPORT 5

BREAKOUT GAME

Group # 38

Intesarul Khan # 260579991
Tamim Sujat # 260551583

Group # 38

Intesarul Khan # 260579991

Tamim Sujat # 260551583

TABLE OF CONTENTS

Group # 38

Intesarul Khan # 260579991

Tamim Sujat # 260551583

FEATURES OF OUR DESIGN

- Successfully able to bounce the ball of the reflecting surfaces in multiples of 45 degrees.
- Successfully increments score counter every time the ball breaks a block.
- Successfully decrements life bar every time the user fails to stop the ball hitting the bottom surface
- Displays a line of text giving score, level number and life total
- Reflective walls are placed on the left, top and right sides
- Successful implementation of a moving paddle that reflects the ball
- An array 5x12 rectangular blocks are used – A block disappears every time the ball is in contact
- Allows the user to control the following:
    - Moving paddle: to direct the paddle left or right
    - Reset button: resets the entire game, score, ball & paddle position
    - Power Button – to switch the VGA output on or off

This report provides detailed descriptions of all the modules used for designing the Breakout game, as well as the tests performed to test the functionality of the circuits. Furthermore, separate images for gate level schematic diagram are also attached throughout the course of the report. References to previous lab reports have also been made for details on working procedures of modules which were not designed in this lab.

Circuit name: BREAKOUT GAME

INPUT: Clock1 (1bit), rst1 (1 bit), rst_Center (1 bit), Paddle1_left (1 bit), Paddle1_right (1 bit), life (3 bits)

OUTPUT: R (4 bits), G (4 bits), B (4 bits), HSYNC1 (1 bit), VSYNC1 (1bit)

CIRCUIT DESCRIPTION:

Our aim is to implement a game consisting of reflective bouncing ball, wall boundaries, an array of blocks and a movable paddle(controlled by the user) onto a VGA display. Figure 1.2c illustrates this desired functionality indicating the intial position of the ball and the paddle. When implementing the game on the altera board, the counter will be using the 50MHz clock of the board as its 'clock' input.

Intially we designed the ball as a 8x8 square. The row and the column coordinates of the ball is represented using two lpm counters: ballrow and ballcol. The motion of the ball was given by incrementing the counter periodically. The ball was made to change direction in multiples of 45 degrees by taking 8 cases into consideration. In each case the counter was either said to increment or decrement based on the direction of motion.

We then designed the array of 60 blocks for the ball to break. In the process 'alpha' of our VHDL where all the required elements were printed, we used a 60 element std_logic_vector to represent the blocks. We set the colour of the block to yellow and it only displayed if its corresponding bit is 1 (block(i) = '1'). Everytime the ball hit block the corresponding bit of the block was set to 0 along with its RGB value. Further testing were done to see if the blocks broken corresponding to a change in score and if the level rose everytime the all the blocks were crushed.

Finally we made the moving paddle and set its RGB value to Red. The user was given control over the paddle at the bottom of the display to reflect ball back in the opposite direction. This is further discussed in the testing section of the report.

Reflection of the ball was achieved by reversing on of the components of the balls motion when the ball's location is the same as the corresponding surface coordinates (i.e- block,wall,paddle).

The completed circuit can be seen in figure 1.1(Pinout diagram) amd 1.3(RTL view).

*Please refer to the previously submitted PDFs titled 'g38_lab report 3'and 'g38_lab report 4' for details on the working procedures of the used modules (VGA display, life, level and score)

Group # 38

Intesarul Khan # 260579991

Tamim Sujat # 260551583

PINOUT DIAGRAM:

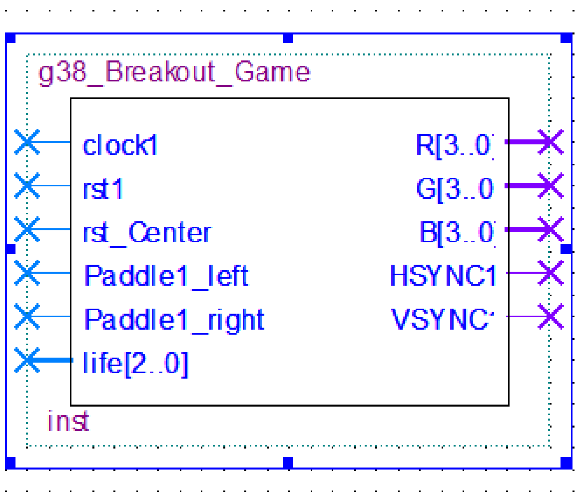

Figure 1.1: Pinout diagram for Breakout Game circuit

TESTING:

A step by step approach was taken to test out the Breakout Game circuit on the Altera DE1 board. The process was repeated several times to achieve the desired results. The process we followed is below:

Step 1: Loading the 'sof' file generated after compilation of the breakout game circuit onto the DE1board (Figure 1.2a below)

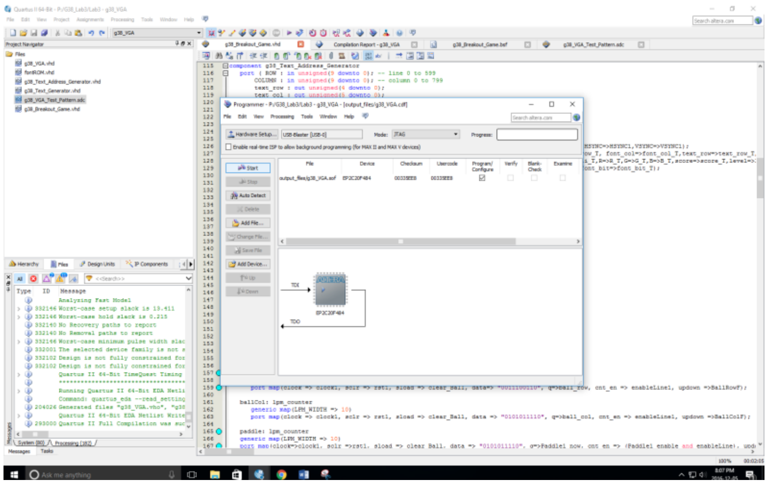

Figure 1.2a: The code is being loaded to DE1 board

Step 2: The switch SW0 in the DE1 board is toggled up to reset the ball and paddle to the center. (Figure 1.2b & 1.2c)

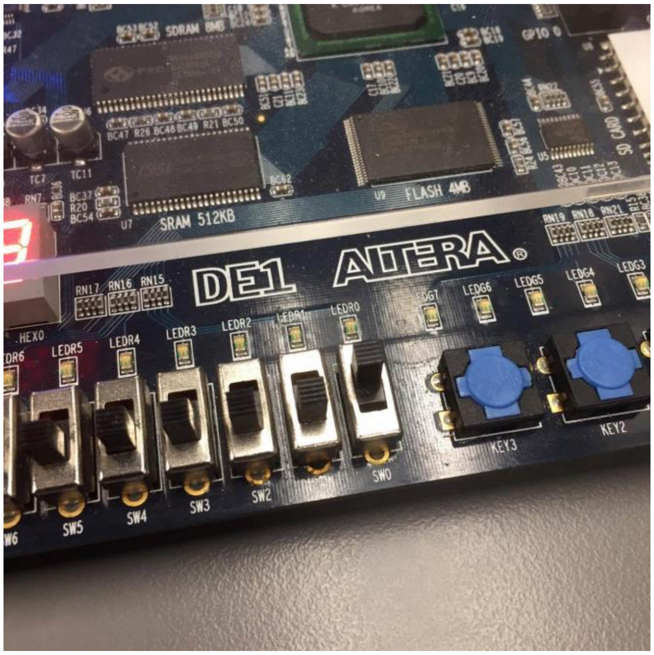

Figure 1.2b: Step 2 of testing in action

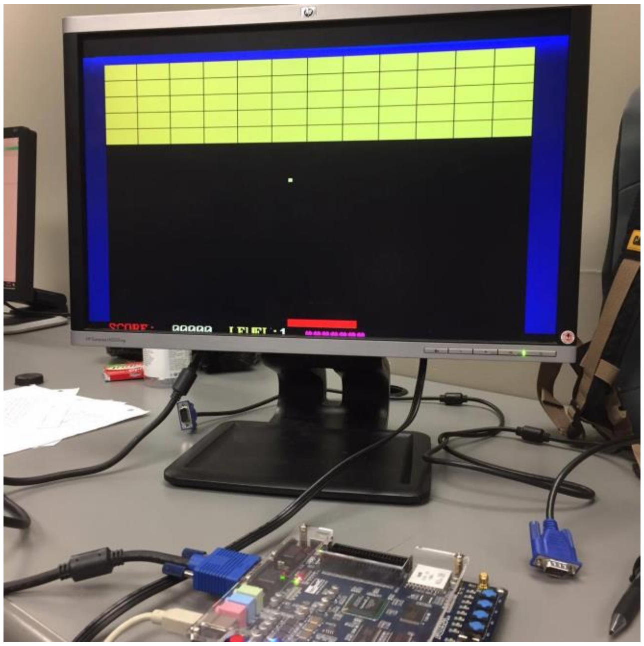

Figure 1.2c: Step 2 of testing in action

Step 3: The game is played using the push buttons KEY1 and KEY0 on the DE1 board to move the paddles left and right respectively. During this step it is checked if the ball is reflecting properly in the walls and after hitting blocks. The block breaking mechanism is also checked. The score and life features were kept in track and performed as desired. (Figure 1.2d +Figure 1.2e)

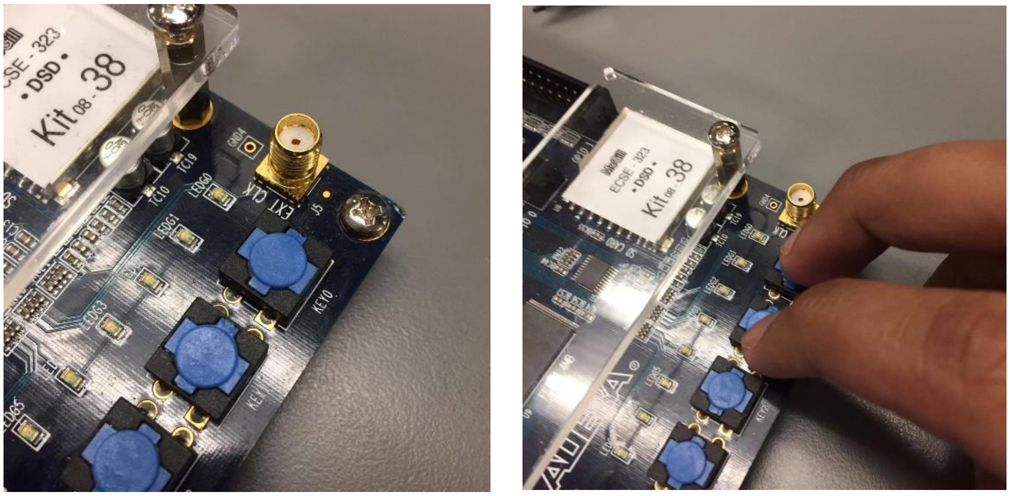

Figure 1.2d: Step 3 in action

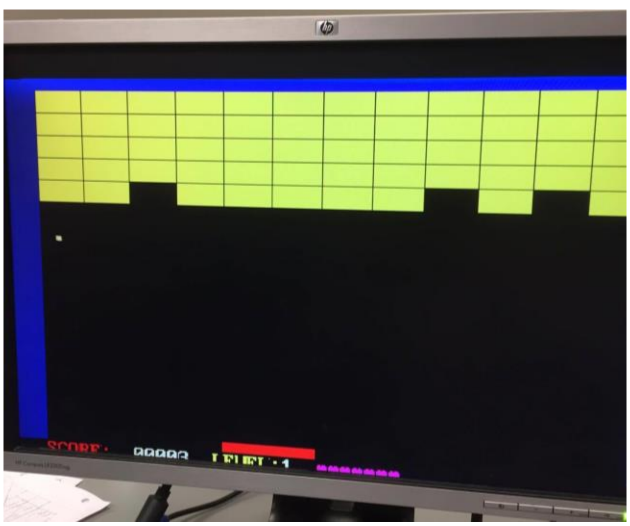

Figure 1.2e: Step 3 in action

Step 4: After the end ends, the ball resets back to the original position. The game can be played again by toggling the SW0 switch up and down again. (Figure 1.2f)

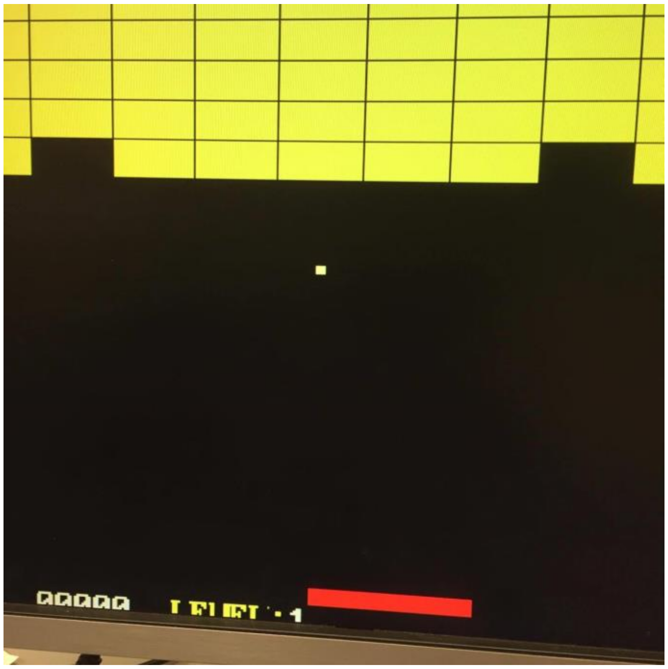

Figure 1.2f: Step 4 in action

Figure 1.3: RTL view for the entire Breakout Game Circuit

SUMMARY OF TIME QUEST TIMING ANALYSIS

The time quest timing analysis was carried out by creating a sdc file with the instructions for the lab 4 slides. Upon execution we got positive values fast model hold slack and slow model setup slack. The slow model Fmax was also greater than 50 Mhz. All these combined has concluded that the timing requirements of the circuit was met. The results are shown below:

Fast Model Hold Summary

	Clock	Slack	End Point TNS
1	clock1	0.215	0.000

Figure 1.4a: Fast model hold slack value

Slow Model Fmax Summary

	Fmax	Restricted Fmax	Clock Name	Note
1	56.02 MHz	56.02 MHz	clock1	

Figure 1.4b: Slow model Fmax

Slow Model Setup Summary

	Clock	Slack	End Point TNS
1	clock1	2.149	0.000

Figure 1.4c: Slow model setup slack value

SUMMARY OF FPGA RESOURCE UTILIZATION

Flow Summary	
Flow Status	Successful - Mon Dec 05 17:29:46 2016
Quartus II 64-Bit Version	13.0.1 Build 232 06/12/2013 SP 1 SJ Full Version
Revision Name	g38_VGA
Top-level Entity Name	g38_Breakout_Game
Family	Cyclone II
Device	EP2C20F484C7
Timing Models	Final
Total logic elements	2,216 / 18,752 (12 %)
Total combinational functions	1,935 / 18,752 (10 %)
Dedicated logic registers	744 / 18,752 (4 %)
Total registers	744
Total pins	22 / 315 (7 %)
Total virtual pins	0
Total memory bits	45,056 / 239,616 (19 %)
Embedded Multiplier 9-bit elements	0 / 52 (0 %)
Total PLLs	0 / 4 (0 %)

CONCLUSION

The first problem faced was that the walls seemed to be of the incorrected size, with enough testing we figured out that this was a problem due to incorrect monitor settings. The next problem we faced was to set an appropritate speed of the ball and paddle. To fix the problem, we initiated multiple counters and had a specific counter for each element movement. We also had difficulty in implementing the breaking of the blocks to be working. Finally, with detailed consultation with our TA we figured out an algorithm which is in our process ‘BlockBreaking’ to be functional.

Presently the game functions at a slow pace and a future improvement can be improving the speed to be much faster. Another problem we faced was that breaking the blocks took a long time since the ball was reflected at the same angle regardless of which side of the paddle it hits. A future solution to this can be implementing different refections angles for each portion of the paddle.

Future enhancements to the system can be implementing levels that changes the block design. There can be blocks which requires multiple hits to break and solid blocks which act as walls. A size decrementing/increating paddle is another future addition that can be implemented.