Design of Embedded System Advanced Course

EDA385

PingPong Report

Submitted by:
Mittapalli Radhika(aso10rmi)

Mittapalli Radhika

Abstract

The main objective of the project is to understand the concept of Hardware Software co-design in embedded system design. As part of this project course, main idea was to implement a pingpong game with score displayed on seven segment display and with sound effects. The hardware platform used for this project was Digilent NEXYS-2 board with a Spartan 3E FPGA. Unfortunately, the complete game as mentioned in initial project proposal could not be implemented due to drop of other project members and due to lack of time. Many more improvements can be done to the basic game implemented.

Table of Contents

2. Introduction

The game has two pads on left and right and a round ball displayed on the VGA screen. Two pads are for two players. The game works same as table-tennis, two players hitting the ball and if a player misses the ball the other player will get a point.

The game is developed on a Digilent Nexys-2 board with a Spartan3E FPGA. The players can play the game by moving the pads up and down with the help of push buttons on FPGA.

The complete game was implemented on hardware though initiallly the game logic was planned to implement on software. Therefore Microblaze processor contains the game logic implemented on hardware.

3. System Architecture

The architecture is shown below in figure 1.

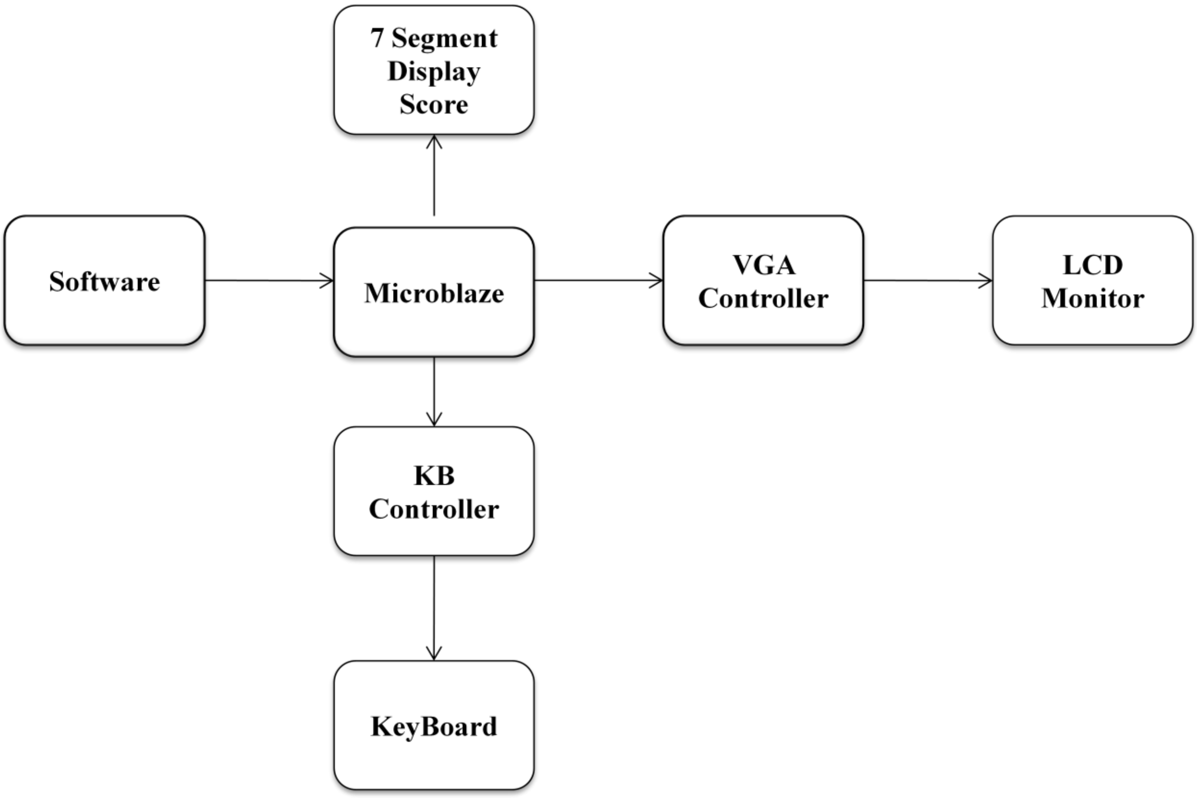


Fig 1:System Architecture

The Microblaze processor on Digilent Nexys-2 board is used for running the game. VGA controller to display on LCD screen, Keyboard controller for interface of keyboard, Seven segment diaplay for displaying the score, Software contains the game logic. Microblaze communicates with different hardware through PLB bus.

With the above architecture proposed initially, VGA Controller was successfully implemented in hardware and the Game logic was also implemented in hardware.

3.1 VGA Controller

VGA controller is used to generate the pixels and to synchronize the signals to display with resolution 640*480. H-sync and V-sync are horizontal and vertical synchronization signals that controls horizontal and vertical scans. Pixel x and y coordinatesl specify the position of the pixel. With *video_on* enable we can have display on the screen for the corresponding pixels and *rgb* indicates the colour of display. The board clock is at 50 MHz, VGA monitor works at 25 MHz. Hence the on board clock is divided by two. A mod-2 counter is used for this purpose.

Initially the objects the left and right wall, the left and right pads and a round ball are created and displayed on the screen by enabling the corresponding video signals and colour can be specified using rgb signal.

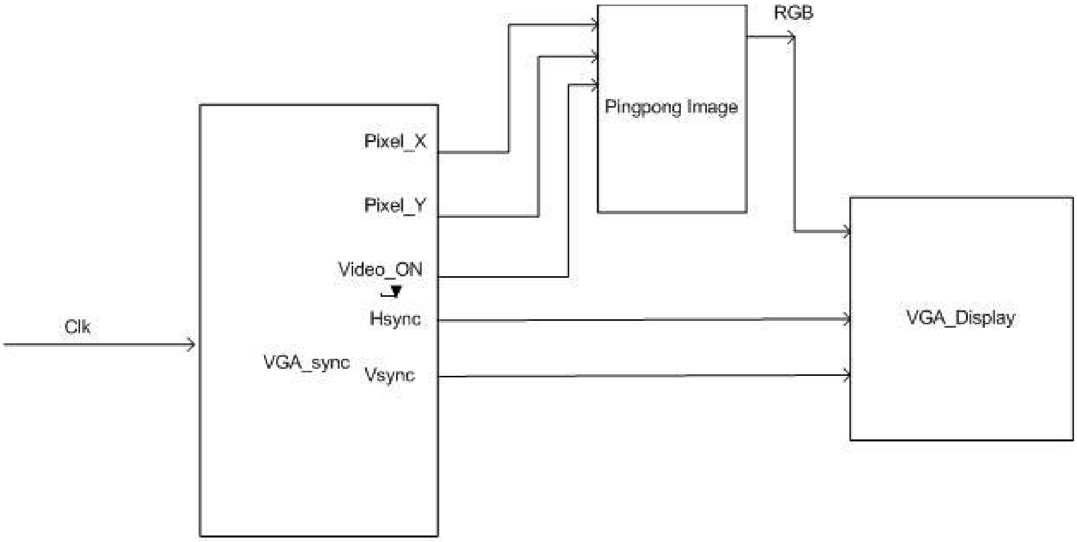
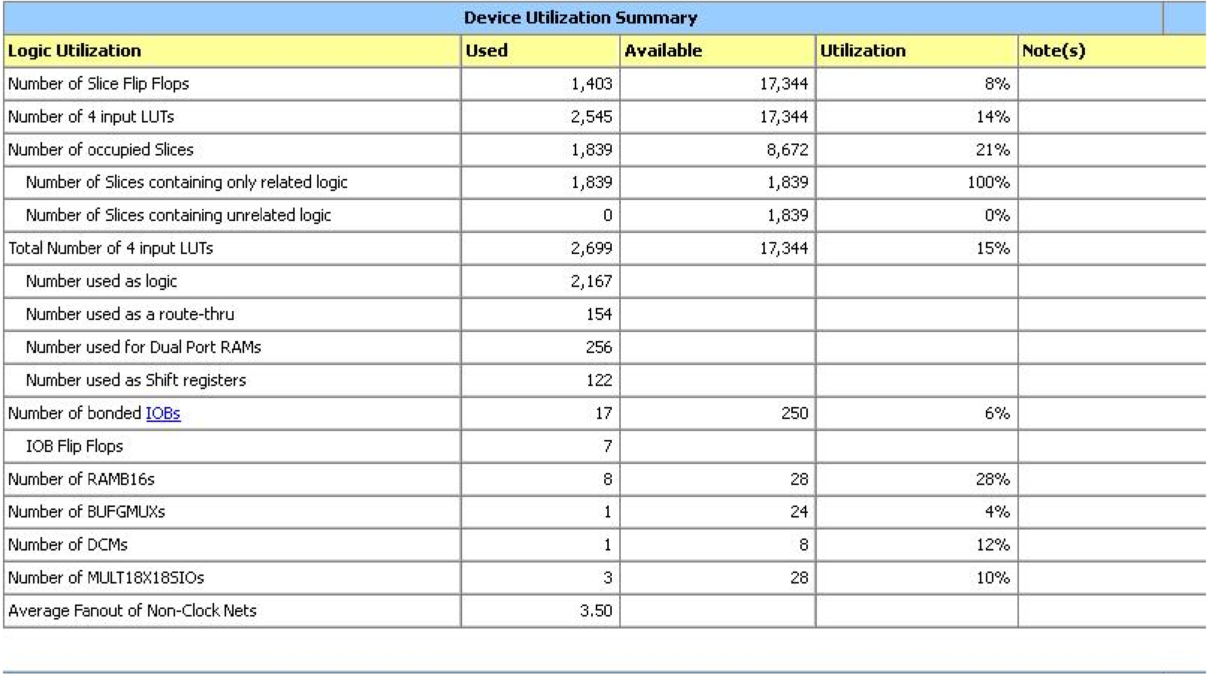


Fig 2: VGA Controller

3.2 Game Logic

The VGA screen is refreshed 60 times per second. A control signal is enabled after completing the entire horizontal and vertical scan (pix_y =481 and pix_x=0). The pads are moved up and down with constant velocity. The pads x coordinates are fixed and only we change y-coordinates so that the pads move up and down until it reach top and bottom.

Animation or movements can be provided to the ball by storing the current x and y coordinates and providing the positive and negative velocity to change direction of the ball based on either hitting the wall, pads or top and bottom of the screen.



Device Utilization Summary				
Logic Utilization	Used	Available	Utilization	Note(s)
Number of Slice Flip Flops	1,403	17,344	8%	
Number of 4 input LUTs	2,545	17,344	14%	
Number of occupied Slices	1,839	8,672	21%	
Number of Slices containing only related logic	1,839	1,839	100%	
Number of Slices containing unrelated logic	0	1,839	0%	
Total Number of 4 input LUTs	2,699	17,344	15%	
Number used as logic	2,167			
Number used as a route-thru	154			
Number used for Dual Port RAMs	256			
Number used as Shift registers	122			
Number of bonded IOBs	17	250	6%	
IOB Flip Flops	7			
Number of RAMB16s	8	28	28%	
Number of BUFGMUXs	1	24	4%	
Number of DCMs	1	8	12%	
Number of MULT18X18SIOs	3	28	10%	
Average Fanout of Non-Clock Nets	3.50			

Fig 3: Device Utilization

4. Problems

Initially faced few difficulties in setting up *.mpd* and *.pao* files. Also there were few problems in configuring the *.mhs* and *.mss* file in XPS. But could solve it with help of professor and also by looking into previous projects.

There were few non technical problems within the team as none of them turned up after initial project proposal. But given the time could implement basic working pingpong game which could be improved further.

5. Contributions

VGA controller and the Game logic parts were implemented in hardware as part of this project and was implemented by Radhika. There were no other contributions for this project.

6. Instructions to use

The game can be downloaded using *download.bit* on to FPGA board and can be played using buttons on FPGA board. Left player can use button4 and button3 to move up and down. Button2 and button1 for the right player to move the pad up and down.

7. Conclusions

This project course helps us to understand and get familiar with Xilinx XPS and also interfacing the different hardware components. Also we get real time experience for understanding hardware-software co design. Pingpong game was implemented and could download the game on to the board and start playing using the buttons. As further improvements scores can be displayed on seven segment which I have tried but due to lack of time it was not implemented completely. Also a keyboard interface can be implemented instead of using buttons on the board to play the game. Graphics can be added and acceleration can be added to the ball movement based on position of hitting the pad.

8. References

1.”FPGA Prototyping VHDL emamples-xilinxSpartan-3version” by Pong
2. Nexys2 Reference Manual