Pingpong Game On Zedboard

EEM465 System on Chip – Project Status Report

Lab Exercise Suggestion

Open-Source Project

ESKISEHIR TECHNICAL UNIVERSITY

Electrical and Electronics Engineering Department

Spring _ 2019

Meysera GENCO

LEARN AND FUN

20/05/2019

Student's Name : Meysera GENCO
Student's ID : ***********

Abstract

Final Report

Ping-Pong Game on Zedboard

Meysera GENCO

Eskisehir Technical University
Engineering Faculty
Electrical and Electronics Engineering

Supervisor: Assist. Prof. Dr. İsmail SAN

My project's idea is to create the Pong game running on Zedboard with a monitor connected to it. The game is implemented from scratch, it has the two pads and a ball drawn on the screen and the background is painted in different color. Everything is implemented in hardware on the FPGA, so this way at the end one can convert the implementation to create an ASIC with the Pong game. The goal is to create the Pong game on a single chip.

Keywords: *Zedboard, chip, display, VGA, FPGA, Pong game, Push button.*

Student's Name : Meysera GENCO
Student's ID : *********

Table of Contents

Error! Bookmark not defined.

Table of Figures

}

Student's Name : Meysera GENCO
Student's ID : ***********

I. Introduction

The creation of games was always in the mind of the engineers. From the start of engineering, scientists always thought about how to create something to entertain themselves, not only the required research work. I wanted to make an open source project and a lab exercise suggestion that would attract student's attention by adding some fun and recreate the Pong game from scratch, by drawing the paddles and the ball, panting the background and displaying it on a monitor. I chose this game, because it is one of the big classic games and it needs not so much drawing, but it's much fun to play or program.

II. Problem Statement

The task is clear and simple; I need to refresh the history by recreating one of the pioneers of the computer gaming history, the Pong game. To make it a little more difficult and interesting, the task is updated to be creating the game on FPGA, where in the future it may be implemented as a standalone ASIC, the chip with the Pong game. The hardware that the system will be implemented on is the Zedboard development board with Zynq®-7000 All Programmable SoC. With this we need also other hardware which is the display to be able to have the whole computer game system. A block diagram of the experimental setup can be seen on Fig. 1, we can see that all the Pong game is on the FPGA board which is connected via VGA port to a PC monitor. The game could be controlled with mouse, keyboard or with the 4 push buttons from the FPGA board.

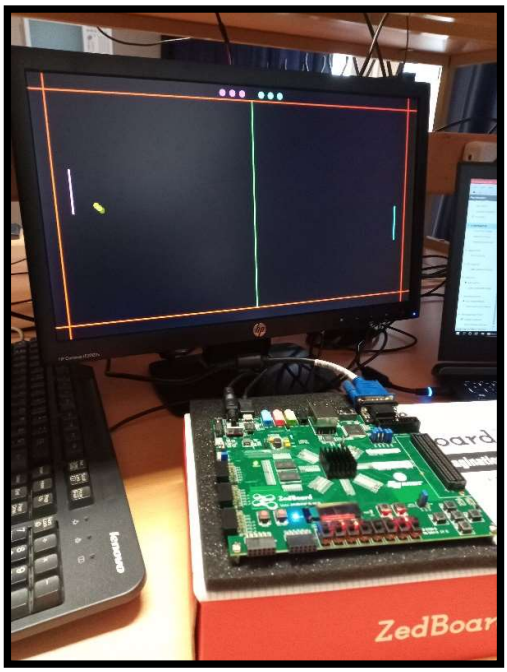

Figure 1: View of the experimental setup and connections.

III. Possible Contributions

One of the possible contributions is a similar design for a team from Politehnica University in Romania. They implemented the idea on NEXYS 2 development board using VHDL programing language in Xilinx ISE environment. Also, other designs and ideas to implement Pong game on FPGA's can be reach from the links in the references section. **[1]**

IV. Specifications & Design

While the game rules are simple, the design also consists of Zedboard itself and an external monitor.

ZedBoard (**Z**ynq **E**valuation & **D**evelopment **B**oard): ZedBoard is a low-cost development board for the Xilinx Zynq™-7000 SoC. This board contains everything necessary to create a Linux, Android, Windows® or other OS/RTOS-based design. Additionally, several expansion connectors expose the processing system and programmable logic I/O's for easy user access. Take advantage of the Zynq™-7000 SoC tightly coupled ARM® processing system and 7 series programmable logic to create unique and powerful designs with ZedBoard. ZedBoard is supported by the zedboard.org community website where users can collaborate with other engineers also working on Zynq designs. **[2]**

Computer Display Monitor: In my case a Monitor produced by DELL with a maximum resolution of 1600x900 and a Refresh Rate of 60Hz is used to display the game objects.

According to the Industry Standards and Guidelines for Computer Display Monitor Timing (DMT), for a 60Hz refresh rate we should offer it a Horizontal Frequency of 60.0 kHz, a Pixel Frequency of 108.000 MHz using the VGA port. **[3]**

V. Implementation

Making the work easier, the task may be divided into two parts. The first part is the Hardware Design part that will be responsible of adjusting the connection between Zedboard and the Monitor "RGB-Display" through the VGA port and getting pure display of the game parts on the screen. Then get control over that displayed figures in order to run the game and enjoy.

The second part will be the Software Design part where it will include a timer interrupt to modify the game mode by rising the ball's velocity making the game harder by the time.

a. Hardware Design

Hardware design forms the main part of my project. Where, a VHDL code is written in Vivado 2018.3 environment to perform the design.

The VHDL code is responsible of generating the fife signals required for the VGA output port of Zedboard (shown in Figure 2.). It's also responsible for controlling the Pong game objects and maintaining the main and actual flow of the game like checking the buttons statements, controlling the ball the ball movement directions and adjusting the scores of players.

Student's Name : Meysera GENCO
Student's ID : ***********

A section of the VHDL code that is responsible of initiating some game objects specifications and registers can be seen in figure 3. To reach the full source code you may visit my GitHub profile from the link in the references section. **[4]**

```
-- Left paddle's left, right, top, bottom and height.
-- left & right are constants. top & bottom are signals to allow movement.
-- L_bar_y_t driven by reg below.
    constant L_PAD_X_L:      integer := 100;
    constant L_PAD_X_R:      integer := 105;
    constant L_PAD_Y_SIZE:   integer := 75;
-- reg to track top boundary (x position is fixed)
    signal   L_PAD_Y_P:      unsigned(10 downto 0) := "00111000010";
    signal   L_PAD_Y_P_next: unsigned(10 downto 0) := "00111000010";
-- Left paddle's OFFSET FROM IT'S POSITION
    signal   L_PAD_Y_T:      unsigned(10 downto 0);
    signal   L_PAD_Y_B:      unsigned(10 downto 0);

-- Right paddle's left, right, top, bottom and height.
-- left & right are constants. top & bottom are signals to allow movement.
-- R_bar_y_t driven by reg below.
```

Figure 2: A part of the VHDL code of the custom IP.

2.4.2 VGA Connector

The ZedBoard also allows 12-bit color video output through a through-hole VGA connector, TE 4-1734682-2. Each color is created from resistor-ladder from four PL pins.

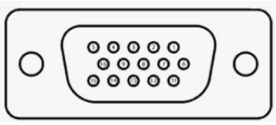

Figure 10 - DB15

Table 8 - VGA Connections

VGA Pin	Signal	Description	Zynq Pin
1	RED	Red video	V20, U20, V19, V18
2	GREEN	Green video	AB22, AA22, AB21, AA21
3	BLUE	Blue video	Y21, Y20, AB20, AB19
4	ID2/RES	formerly Monitor ID bit 2	NC
5	GND	Ground (HSync)	NC
6	RED_RTN	Red return	NC
7	GREEN_RTN	Green return	NC
8	BLUE_RTN	Blue return	NC
9	KEY/PWR	formerly key	NC
10	GND	Ground (VSync)	NC
11	ID0/RES	formerly Monitor ID bit 0	NC
12	ID1/SDA	formerly Monitor ID bit 1	NC
13	HSync	Horizontal sync	AA19
14	VSync	Vertical sync	Y19
15	ID3/SCL	formerly Monitor ID bit 3	NC

Figure 3: VGA Connections for Zedboard

An IP component is generated from the VHDL code to be used in the main Block design as shown in figure 3.

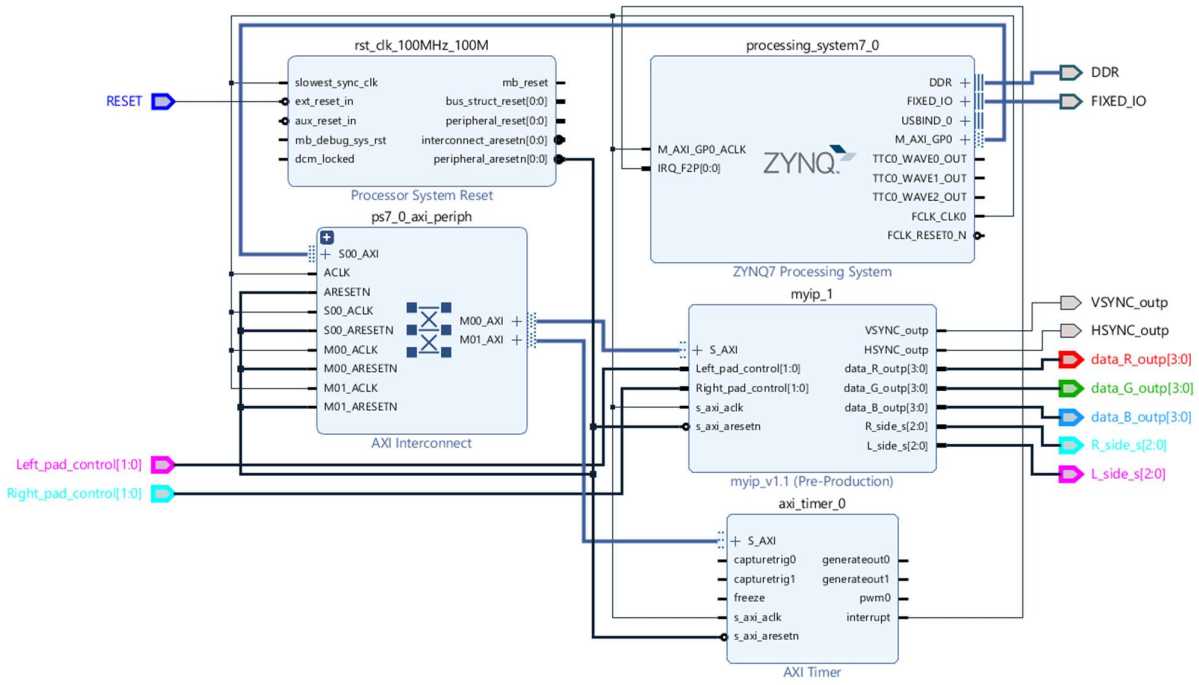

Figure 4: Block Design

The block design consists mainly of three blocks, the Zynq processing system, the custom IP generated from our VHDL code and an AXI Timer IP used to generate Timer interrupts to be handled in the Software part of this project.

b. Software Design

In this part as declared before the timer interrupt is handled and a small modification on the ball velocity register and rewriting the new value to the register at the memory of the custom IP.

Note: A small mysterious problem is occurring presently while handling the Timer interrupt, In other words, when the new value of ball velocity is written to the register the stars to go in strange directions. So, the current system works with a constant velocity.

VI. Results

As a result of this project I got a Ping Pong Game Hardware Design (the software part is still under testing) that can be implemented on one chip so one can order an ASIC "Application Specific Integrated Circuit for commercial purposes for example.

VII. Conclusion

Working with Zedboard, Vivado and SDK environments gives the user an easy way to realize his new ideas and imaginations going through an interesting trip through the hardware and software parts of the design and designing special and custom components and IPs or simply use or modify the IPs provided by Vivado IP Categories.

A demo video for the work and a brief explanation about this project may be reached from the link in the reference section below. **[5]**

VIII. Future Work

A joystick gamepad may be interfaced with this design using the UART protocol and the USB port of Zedboard.

IX. References

[1] HTTPS://GRADEBUDDY.COM/DOC/84547/ECE-443-DESIGN-EXAMPLE-VGA
[2] HTTP://WWW.ZEDBOARD.ORG/SITES/DEFAULT/FILES/AVNET%20ZEDBOARD%20BROCHURE%20ENGLISH%20VERSION.PDF
[3] HTTP://GFILES.CHINAAET.COM/CRAZYBINGO/GROUP/20170606/334-6363235373199427985140055.PDF
[4] HTTPS://GITHUB.COM/MEYSERA
[5] HTTPS://WWW.YOUTUBE.COM/WATCH?V=HBCQTMSMXFS&T=20S