

# FPGA Implementation of Pong Game

Roland Szabó, Aurel Gontean  
Politehnica University Timișoara,  
Faculty of Electronics and  
Telecommunications, Timișoara,  
Romania  
Email: roland.szabo@etc.upt.ro,  
aurel.gontean@upt.ro

**Abstract-**This paper presents the creation of the pong game on a FPGA using a computer display. The game is optimized to work on both LCD and CRT display too. The game is implemented from scratch, it has two pads drawn, the ball is also drawn and the background is painted in white. Everything is implemented in hardware on FPGA, so this way at the end we are able to create ASIC with game. The goal is to create the pong game on a single chip.

## I. INTRODUCTION

THE creation of games was always in the bottom of the engineers. From the start of engineering, people always thought about how to create something to entertain themselves, not only the required research work.

The beginning of the electronics gave big opportunities to the engineers to create games. The first games were also blinking lights and LEDs, but the real games could be made after the introduction of the displays. even the Spartan(3E pins.

One of the first games created was the predecessor of the Pong game, which original name is Tennis For Two, developed on oscilloscope and created by William Higinbotham in 1958.

On oscilloscope there were created two paddles and a ball, which could be hit by the paddles, if the ball was missed, the other player got a point.

We wanted to reproduce the history, but on FPGA and after on a standalone chip.

We wanted to recreate the Pong game from scratch, by drawing the paddles and the ball, painting the background and displaying it on a computer display.

We chose this game, because it one of the big classic games and it needs not so much drawing, but it's really much fun to play.

## II. PROBLEM FORMULATION

Our task was clear and simple; we needed 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, we set the task to create the game on FPGA, because in the future we plan to create the stand alone ASIC, the chip with the Pong game.

The hardware what we had was the NEXYS 2 development board (Fig. 1) with Spartan(3E FPGA (Fig. 3). With this we had also other hardware that we could use like a CRT or LCD computer display to be able to have the whole computer game system. A block diagram of the experimental setup can be seen on Fig. 2, 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 as in our implementation with the 4

We needed to study the whole structure of the development board in order to be able to create the game and to be able to build the correct driver for the VGA monitor,

On Fig. 4 we can see the Spartan(3E other pins which can be connected to the switches, to the push buttons, to the LEDs or to the 7 segment display from the development board. In our Pong game we used the 4 push buttons, 2 for one paddle and 2 for the other paddle for moving them up

The VGA port on the NEXYS 2 development board is interesting. On Fig. 5 we can see the structure of the VGA port which can be found on the NEXYS 2 development

Nexys2 board uses 10 FPGA lines to create a port with 8( bit color VGA and two standard lines of synchronization (HS ( horizontal sync, VS ( vertical sync). Color signals use a resistive divider circuit which together with the 75 A termination VGA display, create eight levels of signals, red and green lines, and four signal levels on the blue line.

The VGA timing needs to be set correctly. A VGA controller circuit must generate vertical sync signals – VS and horizontal sync signals – HS and a coordinate delivery of video data on a pixel clock. The pixel clock defines the time available to display one pixel of information. VS signal defines the frequency of the refresh of the screen and is common to all the information on the screen when is redrawn. The minimum refresh frequency is a function of

the intensity of the phosphor screen and electronic spot. Basically refresh frequency is in the range of 50(120 Hz. For a screen of 480 lines with 640 pixels per line, using a 25 MHz pixel clock and a refresh of 60 + /( 1Hz, signal timings are shown in Fig. 6. Synchronization time pulse width for intervals of “front” and “back porch” (these intervals are times pre(and post(synchronization, during when infor( mation cannot be displayed) are based on observations taken from actual VGA monitors.

A VGA controller circuit (Fig. 7) decodes the output horizontal sync counter, which is driven by the pixel clock, to generate horizontal sync HS times. This counter can be used to locate any pixel on a given line. Similarly, the output vertical sync counter that increments with each HS pulse can be used to generate VS vertical syn(chronization times and this number can be used to locate any given line. These two counters (continuously operating) can be used to address a memory.

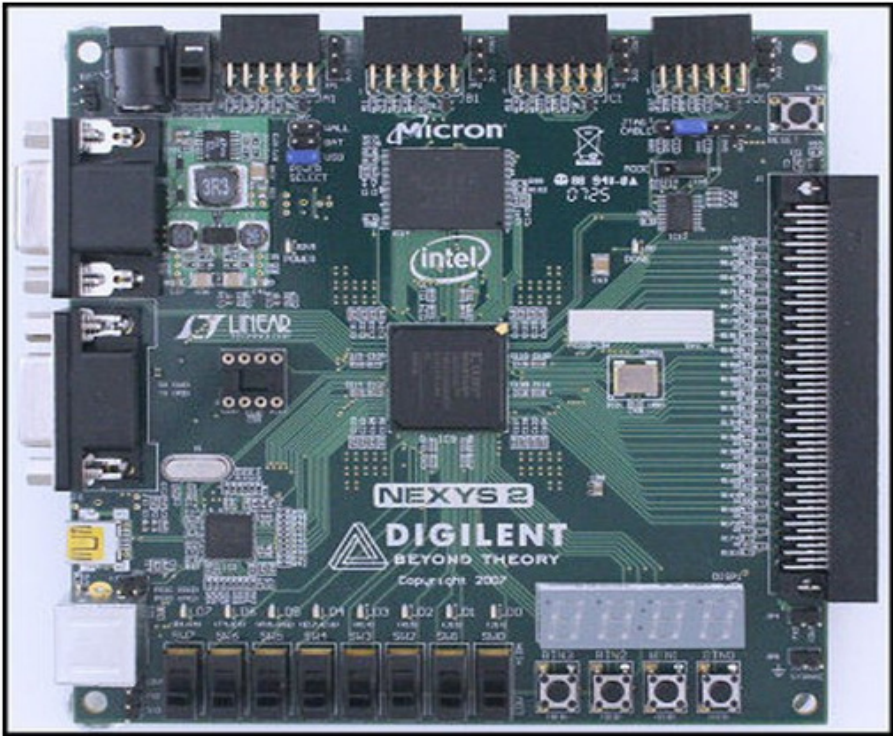


Fig. 1 The NEXYS 2 development board from Digilent with Spartan(3E FPGA

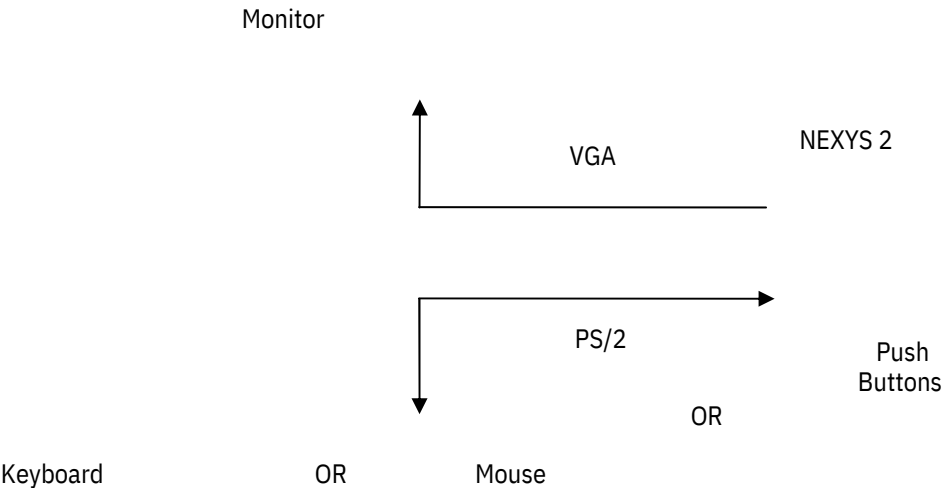


Fig. 2 The block diagram of the experimental setup

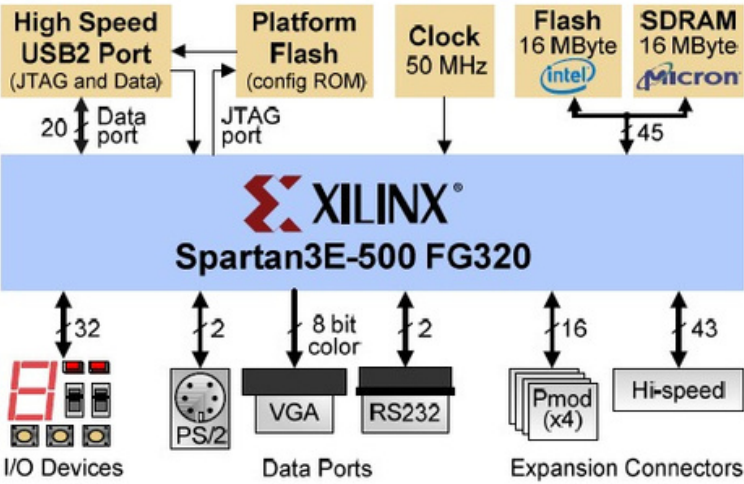


Fig. 3 The Spartan(3E FPGA from the NEXYS 2 development board

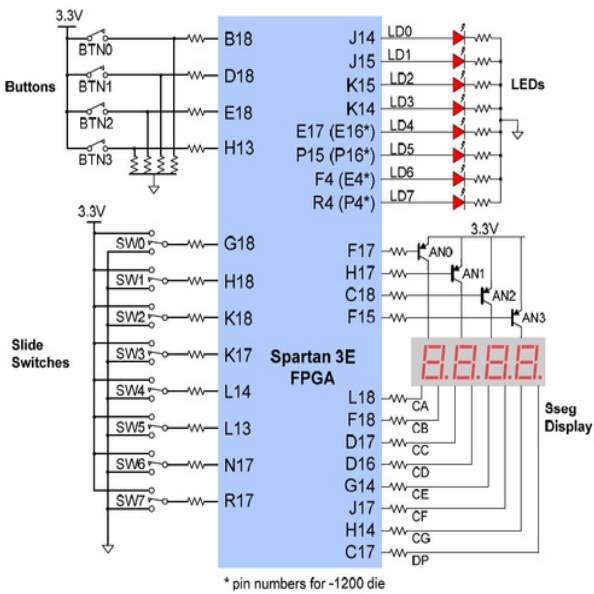


Fig. 4 Spartan(3E pins

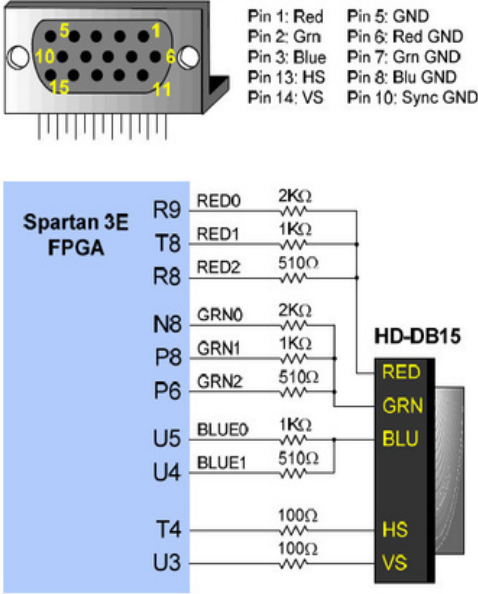
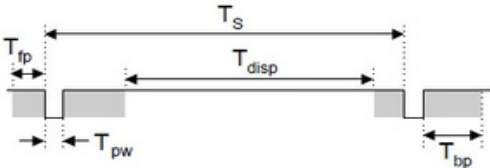


Fig. 5 The VGA port on the NEXYS 2 development board



Symbol	Parameter	Vertical Sync			Horiz. Sync	
		Time	Clocks	Lines	Time	Clks
$T_S$	Sync pulse	16.7ms	416,800	521	32 us	800
$T_{disp}$	Display time	15.36ms	384,000	480	25.6 us	640
$T_{pw}$	Pulse width	64 us	1,600	2	3.84 us	96
$T_{fp}$	Front porch	320 us	8,000	10	640 ns	16
$T_{bp}$	Back porch	928 us	23,200	29	1.92 us	48

Fig. 6 The timing for a resolution of 640x480

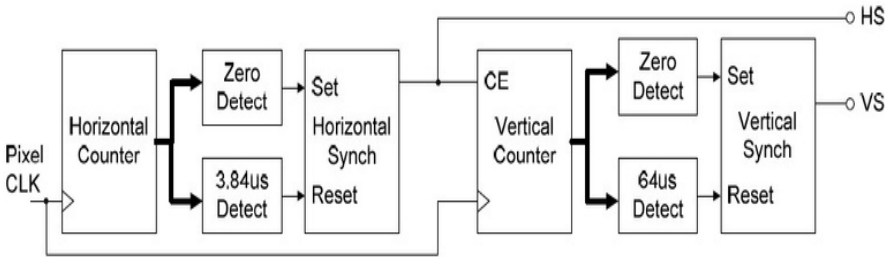


Fig. 7 The circuit for a VGA controller

The Circuit of the Game is not very simple. The chip of the Pong game can be seen on Fig. 8.  
The inside structure of the Pong game can be seen on Fig. 9. As we can see that the inside structure is quite simple for

our circuit. We have only 3 chips inside, one for the VGA sync and one for the graph part for drawing the balls and paddles. We made even the ball to be round by loading the binary values from ROM memory. The last chip is a D(latch for creating a delay for timing.

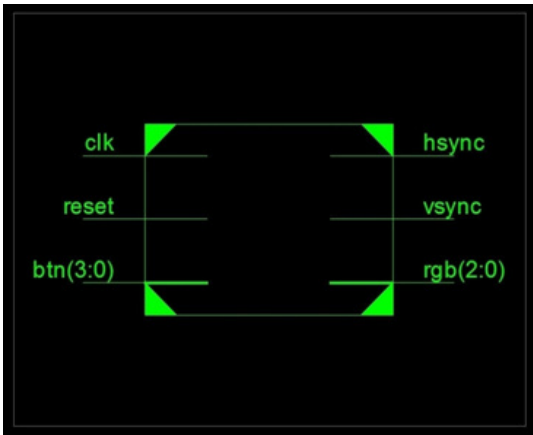


Fig. 8 The Pong game chip

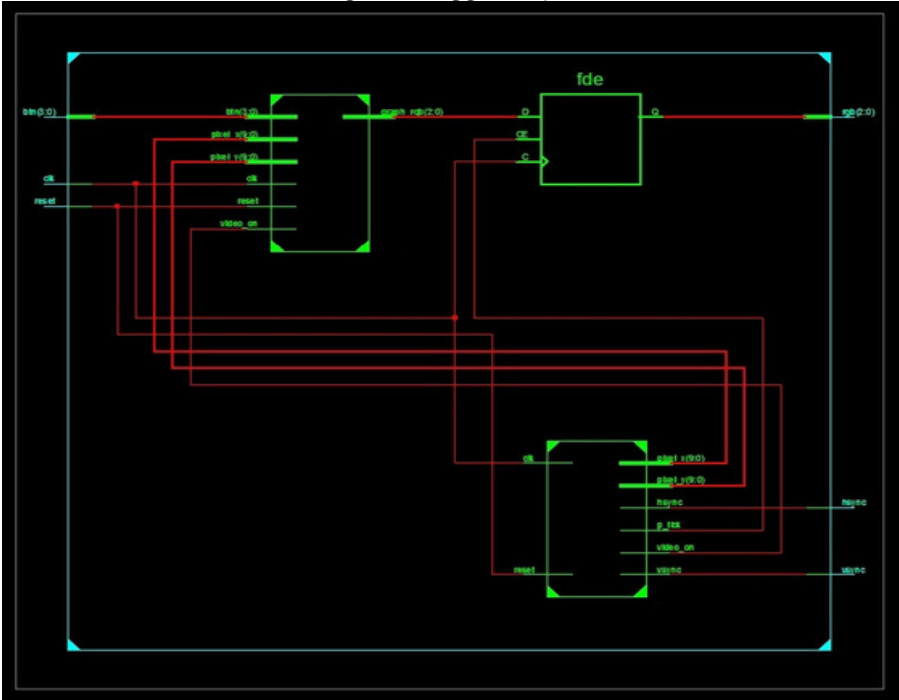


Fig. 9 The inside structure of the Pong game chip



Explanation of the game rules will be presented next. On Fig. 10 we can see the result of the Pong game on a CRT monitor with VGA input. The paddles are controlled with the push buttons from the NEXYS 2 development board.

The game consists of two blades and a ball. Background color of the paddles and the ball is configurable in software. The background is white, the paddles are red and blue, and the ball is green.

Two players can play the game with the 4 push buttons on the board. If the board is oriented in normal position, with the 4 push buttons right below, the first player, with blue paddle, will have the first 2 push buttons (BTN0 movement up, BTN1 movement down) and the second player, with red

paddle, will have the last 2 push buttons (BTN2 movement up, BTN3 movement down).

The ball moves automatically hitting the top and bottom edge of the screen and the paddles when the ball it's hit by a player. If a player misses the ball, then he gets a goal, so the ball goes off the screen to the right or left side. For example, if a player misses the ball on the right, then left player gave a goal to the right player. After this it will be a new ball on the left, so the left player serves. The ball will move straight from left to right as long as the right player manages to hit the ball. If the right player scores a goal, then the ball starts its automatic movement from the right. The game is repeated until it's stopped. The player, who manages to give the most goals, wins.



Fig. 10 The result of the Pong game on a CRT monitor

#### IV. CONCLUSION

As we could see we created a Pong game on FPGA. Our goal was to reproduce one of the most entertaining games of the computer game history. The Pong game was one of the first electronic games and maybe made the bases of the computer games history.

It was a good choice to create it on FPGA, because we had a suitable development board which has VGA port for interfacing a computer monitor and push buttons for controls. Making it on FPGA was a good idea also, because this way we had an embedded system, this way we don't have other dependencies not even operating system dependency. We also chose this platform, because this way we could create an ASIC, a standalone chip, by converting the VHDL code to Verilog with Mentor Graphic tools and we could create the layout of the chip for the final product. This way we had a game implemented on a chip, on hardware; this means that we have a standalone device with no speed issues, like if it would be made in software on a microcontroller.

In future we plan to extend the controls from push buttons to mouse and keyboard on PS/2 interface. We plan also to port the project to other platform too, like the ATLYS board, and create the video drivers on DVI or HDMI interfaces, for newer monitor types and create the keyboard and mouse drivers on USB interface or even add joystick drivers on USB interface.

#### REFERENCES

- [1] R. Szabó, A. Gontean, "Creating a Serial Driver Chip for Commanding Robotic Arms," , 2013, pp. 671–674.
- [2] R. Szabó, A. Gontean, "Creating a Programming Language for the AL5 Type Robotic Arms," , 2013, pp. 62–65, <http://dx.doi.org/10.1109/TSP.2013.6613892>.
- [3] R. Szabó, A. Gontean, I. Lie, "Smart Commanding of a Robotic Arm with Mouse and a Hexapod with Microcontroller", # \$ % & ' ( & ) , 2012, pp. 476–481.
- [4] Ye Xien, Ye Zhiquan, "Implementation of PS/2 Mouse and Keyboard Based on Windows CE," , vol. 3, 2009, pp. 318–321, <http://dx.doi.org/10.1109/IFCSTA.2009.318>.
- [5] Punj Pokharel, Binod Bhatta, Anand D. Darji, "Optimized drivers for PS/2 and VGA using HDL," & & & , vol. 3, 2011 pp. 262–266, <http://dx.doi.org/10.1109/CSAE.2011.5952677>.
- [6] Guoping Zhang, Man(de Xie, "Design of visual based(FPGA Ping( Pang game with multi(models," + , vol 2, 2010, pp. 31–34, <http://dx.doi.org/10.1109/PACCS.2010.5626899>.
- [7] Deep Vardhan Bhatt, D. Du Toit, Gerhard P. Hancke, "Design of a controller for a universal input/output port," & & % , 2012, pp. 647–652, <http://dx.doi.org/10.1109/I2MTC.2012.6229492>.
- [8] Kyu(Sam Sam Lim, Kon(Woo Kwon, Heeju Ju Park, Jeong(Hun Hun Kim, Suki S. Kim, Jun(Jea Sung, Kwang(Hyun Hyun Baek, "Design an infrared wireless optical mouse system and a dual(band infrared receiver," #- & & & + & , 2008, pp. 810–813, <http://dx.doi.org/10.1109/ICECS.2008.4674977>.
- [9] Noriyuki Aibe, Moritoshi Yasunaga, "Reconfigurable I/O interface for mobile equipments," & & , 2004, pp. 359–362, <http://dx.doi.org/10.1109/FPT.2004.1393299>.
- [10] Liu Kai, Yang Yuliang, Zhu Yanlin, "Tetris game design based on the FPGA," , \* & + ' / & ' , 2012, pp. 2925–2928, <http://dx.doi.org/10.1109/CECNet.2012.6201435>.