The Double Threat Game

Zack Che

Cal Poly Pomona

I. INTRODUCTION

The Xilinx's Spartan-3E field-programmable gate array (FPGA) board is mainly intended for students in order to learn and practice Verilog source code or VHDL. This paper will analyze and summarize the process of creating a custom game called Double Threat by using the Verilog source code to implement this board and a PS/2 keyboard to control the movements of the object(s) in the game.

II. GAMEPLAY

The game is played by using a bar to touch a moving ball, and the user will be rewarded one point when this objective is met. The game involves a ball and two bars, and each bar will be controlled by a user via the PS/2 keyboard. The bar could be moved vertically or horizontally. It is important to note that once a key is pressed, the bar will be kept moving in that direction until another key of another direction is pressed. For instance, the A key is set to "left" direction; therefore, during the game, if the user presses the A key, his or her bar will move to the left continuously until another key of another direction is activated. The reason for this is to make the game more challenging, thus more fun. When the bar is set to move continuously, it would require the user to pay full attention to the game or the movements of the bar and ball in specific, or else this user will not score a point. Once the user scores a point, this point will be displayed in binary in the form of LEDs.

III. SOFTWARE

For this project, a main module, which also includes other sub-modules, must be created. The main module shall have all of the necessary inputs and outputs provided by the FPGA board. However, the numbers or names of these inputs and outputs are not known yet because the sub-modules must be created and tested first before one could be really sure what inputs and outputs to use. For this reason, the first sub-module that shall be created is the process of video graphics array (VGA) synchronization. The main function of this module shall be too constantly declare or set the VGA synchronization to be within the 640-by-480 pixels parameter. Next, a sub-module that creates the custom game shall be developed. In this sub-module, the background, objects, and other lay-out(s) of the game should be coded properly so that no error(s) or complication(s) would happen while the game is running. After that sub-module, another sub-module shall also be created, and its function is to instantiate, implement, and connect the PS/2 keyboard to the FPGA board in order for that keyboard to control the object(s) in the game. Last but not least, a sub-module which includes the functions of the universal asynchronous receiver/transmitter (UART) shall be created. It is important to note that the UART is an important sub-module for this project because it is used for serial communications such as between a computer and a FPGA board. The transmitter function of the UART "loads data in parallel and then shifts it out bit by bit," and "...the receiver shifts in data bit by bit and then reassembles the data" (Chu, 2008, p. 215).

IV. HARDWARE

Once Verilog source codes of the project have been written, modified, and cleared of errors, the next step is connecting the FPGA board to a monitor to test to see if the game works. For this process, the Starter Kit Board User Guide shows that the "board includes a VGA display port via a DB15 connector," and "the VGA connector is the left most connector along the top board" ("Spartan-3E...", 2011, p. 55). Since the PS/2 keyboard is used in this game to control the movements of the bar(s), the UART sub-module is created to receive inputs from the keyboard via the PS/2 mouse/keyboard port, and then the transmitting function on the DCE-style port will transmit the data from the keyboard to the computer via a RS-232 cable, and the data will be displayed in ASCII codes on the PuTTY software. Moreover, it is critical to note that "the keyboard sends data to the host in 11-bit words that contain a '0' start bit, followed by eight bits of scan code (LSB) first, followed by an odd parity bit and terminated with a '1' stop

bit" ("Spartan-3E...", 2011, p. 65). Additionally, the eight bit scan code will be divided in two, and each part will be displayed in the corresponding ASCII code.

V. CONCLUSION

In conclusion, the game can be created via the Spartan-3E FPGA board by using the proper Verilog source code, cables, and knowledge. Moreover, a good understanding of the lay-out of the FPGA board, pin assignments on the board and the cable, and coding modules is required in order for the game to run smoothly with any complication(s). The first step is to analyze what hardware is needed for this project, and the next step is to write Verilog source code on connecting each piece of hardware together. For this reason, the PS/2 keyboard, which is used to control the movements of the bar(s) in the game, the VGA cable, which is used to connect to a monitor in order to physically display the game, and a RS-232 cable, which is used to transmit data from the FPGA board, are all connected physically to the FPGA board and also connected "internally" via the Verilog source code, which is written by the author. By large, when the game is tested, the bar(s) will move as the button(s) on the keyboard is pressed, and the score(s) will be displayed in the form of LEDs as expected.

REFERENCES

[1] P. P. Chu. *FPGA Prototyping by Verilog Examples*. Hoboken, New Jersey: John Wiley & Sons, Inc, 2008.

[2] *Spartan-3E FPGA Starter Kit Board User Guide*. San Jose, California: XILINX, 2011, V 1.2.