EE314: DIGITAL CIRCUITS LABORATORY

TERM PROJECT MILESTONE REPORT

"YanaYana"

Group 66: Caner Süren, Barbaros Şair, Abdussamed Turabi

1.Introduction

In this term project, we will try to design and implement a game that is similar to "Footsies" game. This game is played by two players and run on FPGA using VGA output on screen. In this game, the players can attack, move, and block the other player. Since this report is only the milestone report, we will focus on the objectives and operations of our milestone code game.

Our main objectives in this project's milestone report and demonstration are to take inputs from clock and buttons, process the inputs in state machine, and render resulting images on screen utilizing VGA. The characters are able to move, block, and attack on the other player using keys of the FPGA. For different types of attack and movements, different blocks (hitbox, hurtbox, et cetera) should be shown on the screen. For this purpose, we implemented structures like clock divider, finite state machine, VGA controller, et cetera.

This milestone report summarizes our study and implementation that we have done for the milestone demonstration so far.

2.Theoretical Background

Implementing the game requires a technical knowledge spanning different areas like: State machines, clock dividers, VGA controllers.

To start and put emphasis on the theoretical background of the VGA controller, it has a pixel clock of 25 MHz. To operate VGA in 25MHz, FPGA's clock signal, which is 50MHz should be divided in two. Since we have 800x525 signals for 640x480 pixels, it refreshes the screen on 60Hz frequency computed by the Equation 1.

$$Refreshing\ Frequency = (\frac{25MHz}{800 * 525})$$

Equation 1

VGA driver module has 2 signals of VSYNC and HSYNC that is in active mode initially, and HSYNC signal remains in active mode for 640 pixel clock cycles which is also width resolution.

Starting from the upper left corner, HSYNC sends 8-bit color information for each 640 clock cycles, then it is converted to the analog voltage signal which varies between 0 and 0.7 Volts. After 640 pixel clock signal cycle ends, color voltage values are set to 0 and HSYNC remains active high for the frontporch. Frontporch is not included in the screen pixel resolution, though it is present in the 800x525 signals. Then HSYNC is set low for Sync pulse and active

high again for the backporch. These blanking regions are left to stabilize the driver though they doesn't have any analog RGB values. VSYNC signal does exactly the same operation for 480 pixel clock cycles and its corresponding blanking region.

Secondly, we have a state machine by which we determine the next states and the outputs from the current state and inputs, also, we track the positions. Inputs of the state machine are the keys and switches of FPGA and also the clock signal. Different modules should work together to determine which pixels belong to the character and which pixels are used for the background.

To eloborate on technical background of finite state machines, they work with registers and counters and they are sequential circuits, so they have a feedback systems. When an input is given to the finite state machine, it determines the output and the next state from the current state.

Thirdly, we utilized a clock divider to produce required clock signal frequencies. Clock divider divides the clock signal to less frequencies by utilizing a counter. So, the resulting output clock signal toggles less frequently than the original clock.

3.Proposed Solution

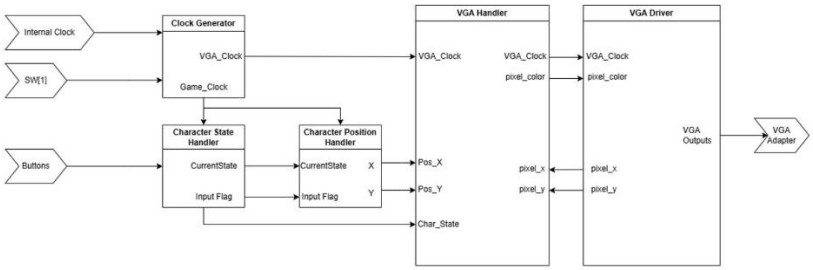

Figure II: Proposed Solution Block Diagram

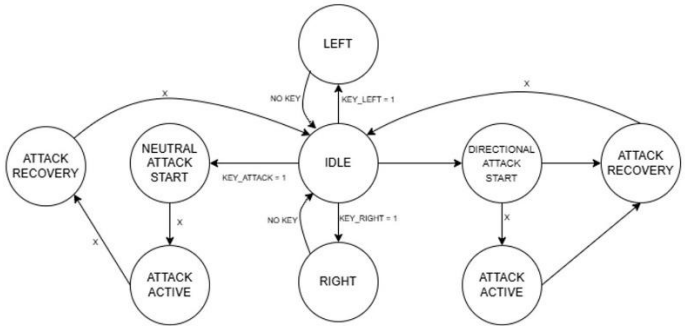

Figure III: State Machine Diagram

4.Results

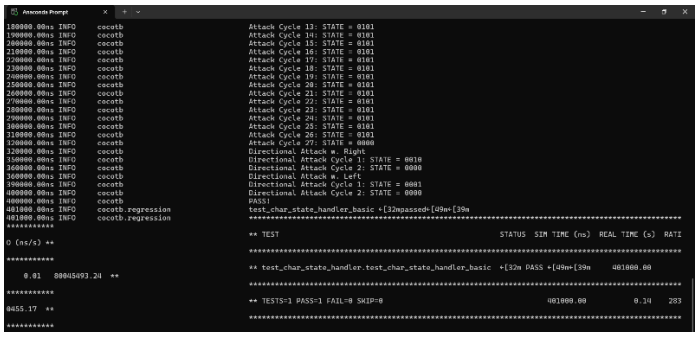

Figure IV: Succesful Testbench Results

IDLE STATE

ATTACK STARTUP STATE

ATTACK ACTIVE STATE

ATTACK RECOVERY STATE

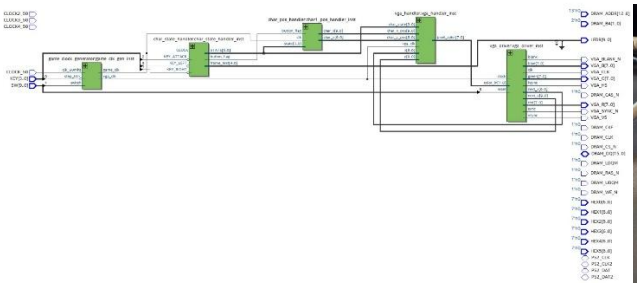

Figure V: RTL Schematic for Proposed Solution

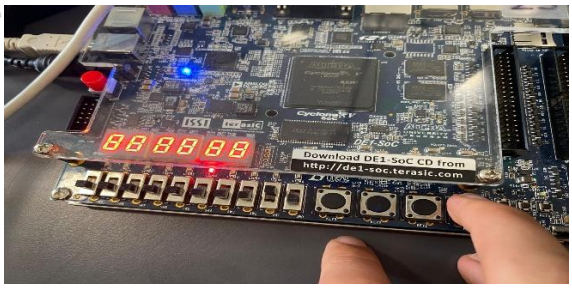

Figure VI: Testing States on Hardware

5.Challenges and Discussion

The challanges that we have encountered in this project were mainly clock divider based. Because, we implemented our clock divider falsely at the beginning, VGA was giving no outputs. We had a hard time debugging it fixing the problem to get visuals on screen using VGA.

6.Conclusion and Future Work

To summarize, we have obtained visuals on screen utilizing VGA, written state machine code that will be the backbone of the project, and implemented moves and attack stages. In the future, we will add the second character, implement blocking dynamic, paint our characters using pixel art, and health-block boxes. In the end, our characters will be able to attack and block each other, and we will count health points and block point resulting in win.

7.References

1-https://hifight.github.io/footsies/

2-https://vanhunteradams.com/DE1/VGA_Driver/Driver.html

3-https://www.chipverify.com/verilog

4-http://www.javatpoint.com/verilog