Alan Li

EE 371

February 12th, 2022

Lab 3 Report

Task1

## Section 1: Procedure

Chart, waterfall chart

Description automatically generated

My approach to draw a line for task 1 is to display all pixels on the line one by one at each clock edge. There are 3 modules for task1, DE1\_SoC is top-level module, VGA\_framebuffer has already been designed. I implement a line\_drawer module to finish the task. The algorithms for developing line\_drawer can be found here: <http://members.chello.at/easyfilter/bresenham.html>. I did some modifications since the algorithm is originally written in C which is a programming language. There is some differences between programing languages and VHDL. I put the lines inside for loop of c code in a always\_ff block, in that way the registers will get updated at each rising clock edge. I put everything else inside an always\_comb block.

## Section 2: Results

Line\_drawer testbench

Graphical user interface

Description automatically generated

For line\_drawer testbench, I set x0, y0, x1, y1 as 100,100,150,300. After reset, both x and y are increasing at different rates because the slope is not 45 degrees. After hundreds of clock cycles, xPointer and yPointer has reached the destination coordinates and will stop there like displayed below.

Graphical user interface, application

Description automatically generated

## Section 3: Appendix

Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated Text

Description automatically generated

Task2

## Section 1: Procedure

I’m using the same lilne\_drawer module as task1, for task2 I added a line\_animation module that make the line move around the screen. The module periodically updates the start and end point. I implemented a module of a three-state FSM. The state diagram is given below.

A picture containing diagram

Description automatically generated

After the counter reaches the maximum, the drawing would be finished, the system goes to the state of refresh and redraws the same path but with color black. Before moving to the next drawing state, the system updates the new coordinate and makes sure that the coordinate stays within the bound. If the reset switch is enabled, the module goes immediately from drawing to refreshing. After clearing the screen, the system stays at reset until the reset switch is off.

A picture containing table

Description automatically generated

## Section 2: Results

Graphical user interface

Description automatically generated

From the simulation for line animation, you can see that the x0, y0 is getting updated. At the same time, the color is toggling between 1 and 0.

Graphical user interface

Description automatically generated

The simulation for line\_drawer is the same as task1

## Section 3: Appendix

Text

Description automatically generatedText

Description automatically generated

Graphical user interface, text, application

Description automatically generated

A picture containing text

Description automatically generated

Text, letter

Description automatically generated

Text

Description automatically generated