CSE 371 Lab 5 Report

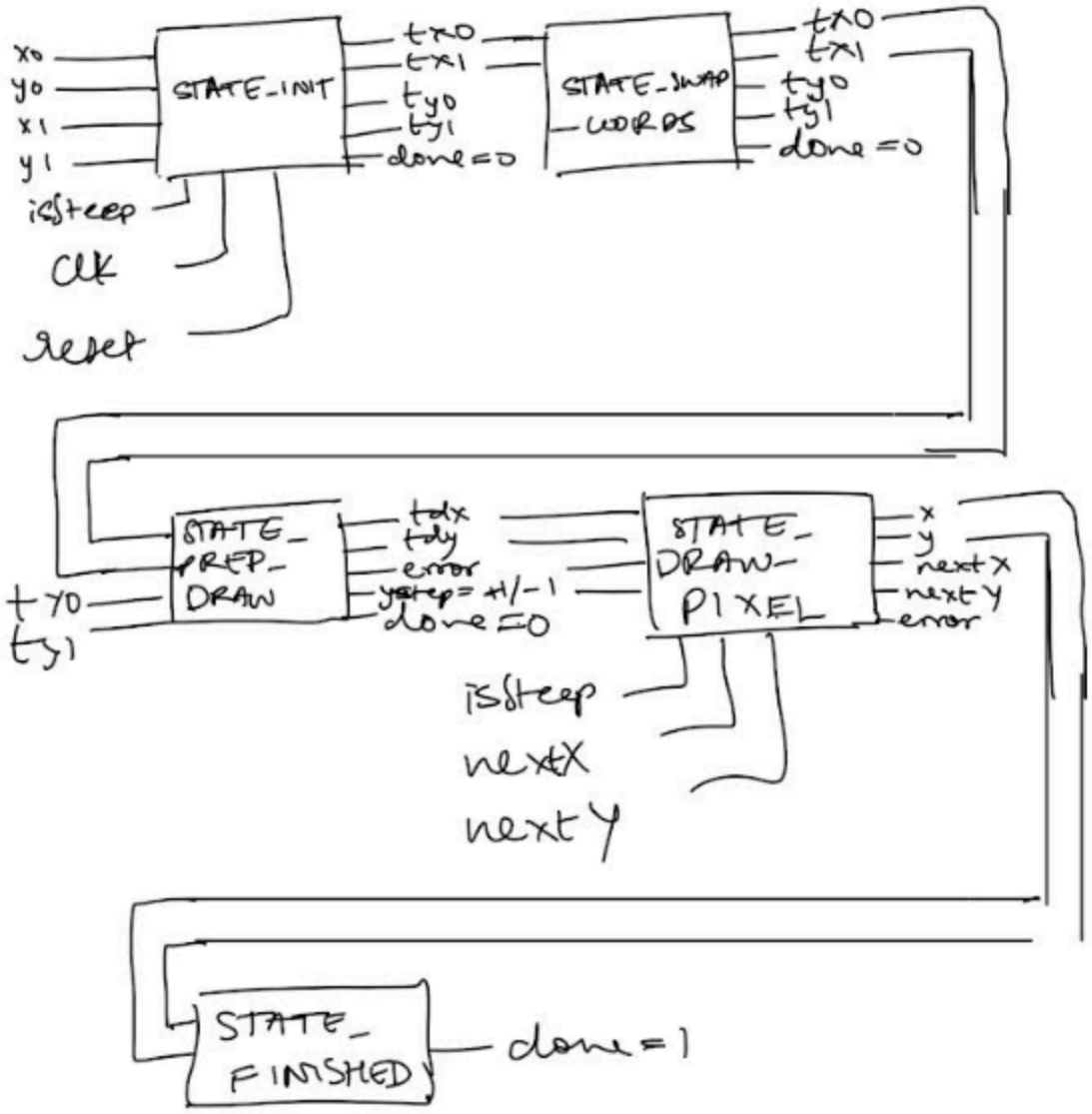
Elijah Melton, 2164822
Ankith Tunuguntla, 2234509

Design Procedure

line_drawer.sv (Task 2)

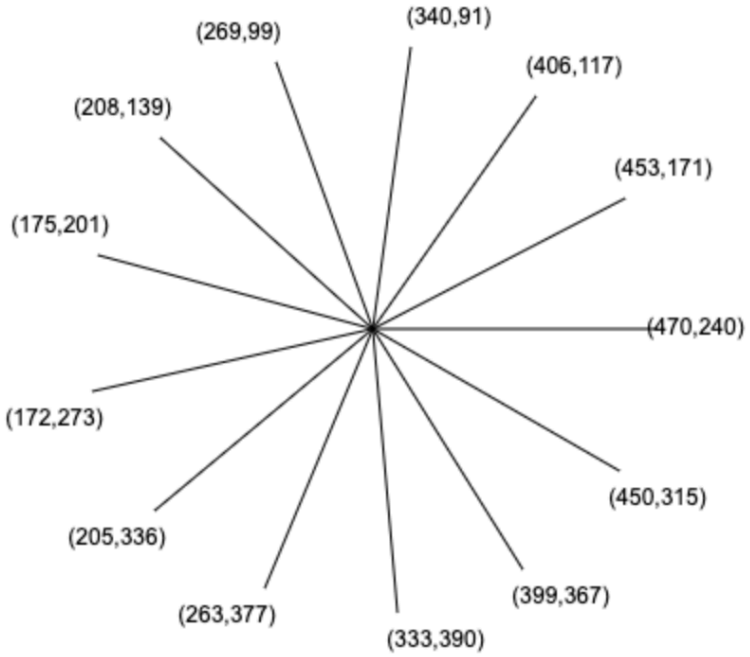
The line_drawer module implements Bresenham's line drawing algorithm using an FSM with five states: STATE_INIT, STATE_SWAP_COORDS, STATE_PREP_DRAW, STATE_DRAW_PIXEL, STATE_FINISHED. It outputs a sequence of x,y using the algorithm when given x0, y0, x1, y1. Using the coordinates, features like steepness and traversal in either direction is determined and a line is drawn pixel by pixel. The module begins by checking whether the line is steep and conditionally swaps the x and y coordinates to simplify processing. It then ensures the traversal proceeds from left to right by swapping endpoints if necessary. Once initialized, it calculates the required deltas, step direction, and error term used for vertical correction. The drawing process advances one pixel per clock cycle, outputting each coordinate while adjusting the y-position based on the accumulated error. The `done` signal is asserted when the entire line has been drawn, allowing external modules to detect completion.

Here is a diagram that supports the explanation.



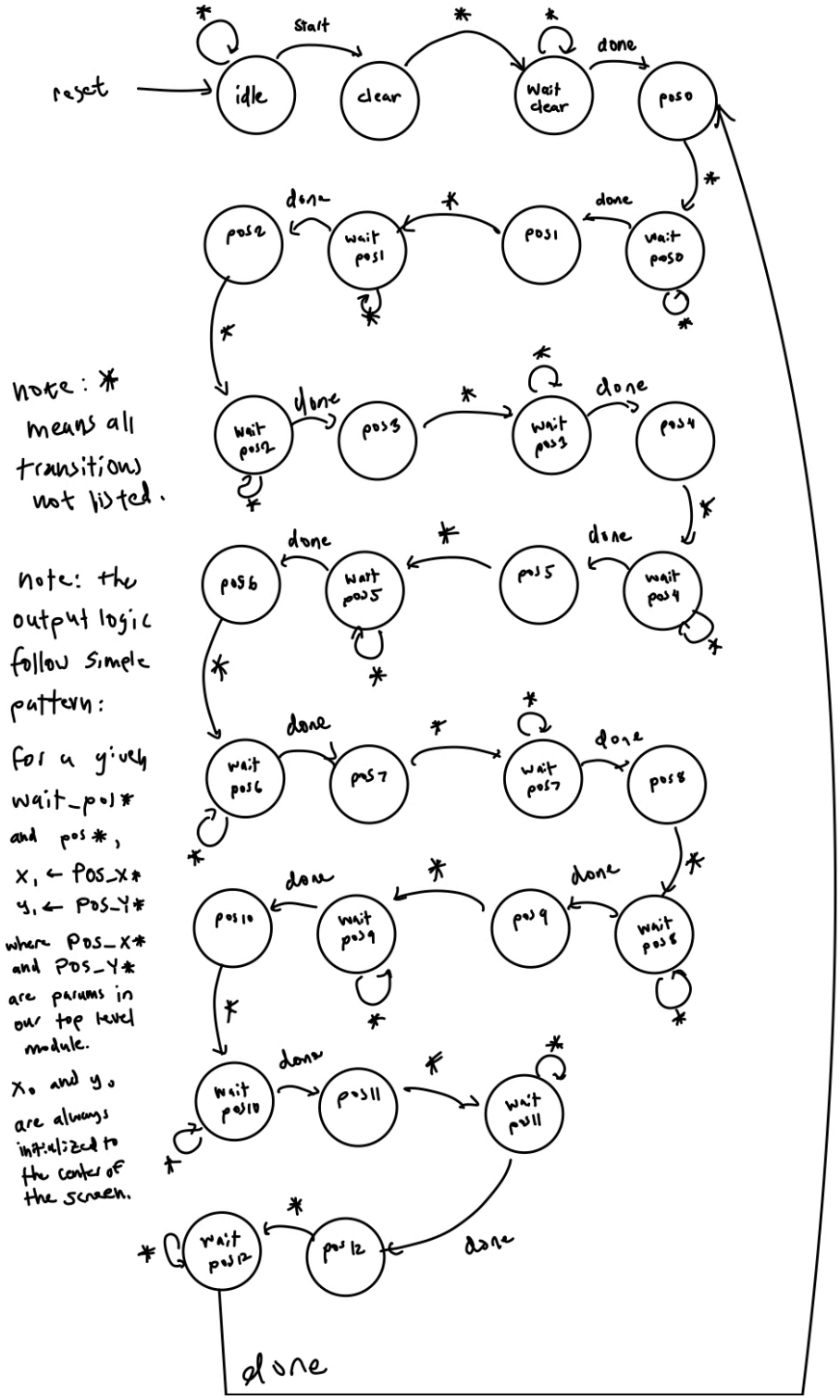
Top Level Module

Our animation outputs a “snowflake”, e.g. evenly space lines starting from the center and going outwards in roughly a uniform circle using the following approximations:



Each line has two states, `POS<i>`, and `POS<i>_WAIT` in an FSM. The module starts writing the pixels during the `POS<i>` state, and then immediately moves to `POS<i>` wait until `line_drawer` asserts `done`, at which point it moves to `POS<i+1>` and continues. Once it reaches `POS12_WAIT`, it loops back to `POS1`

Animation FSM

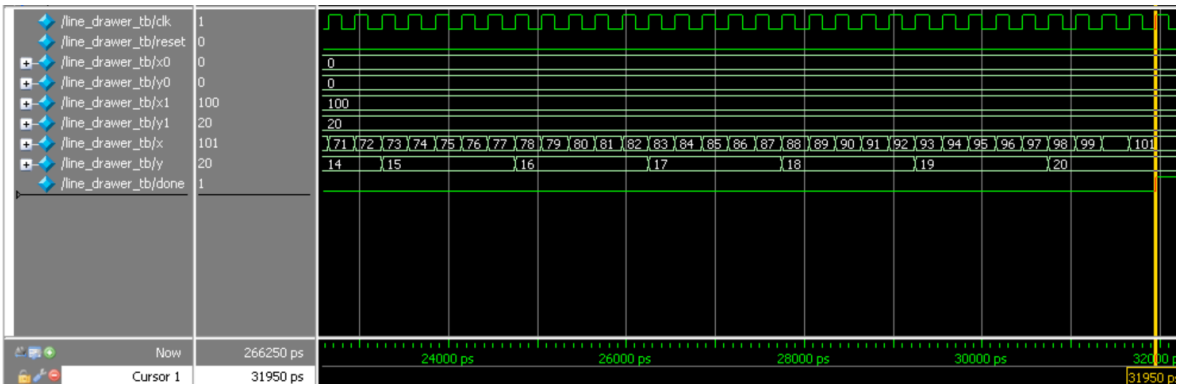


Results

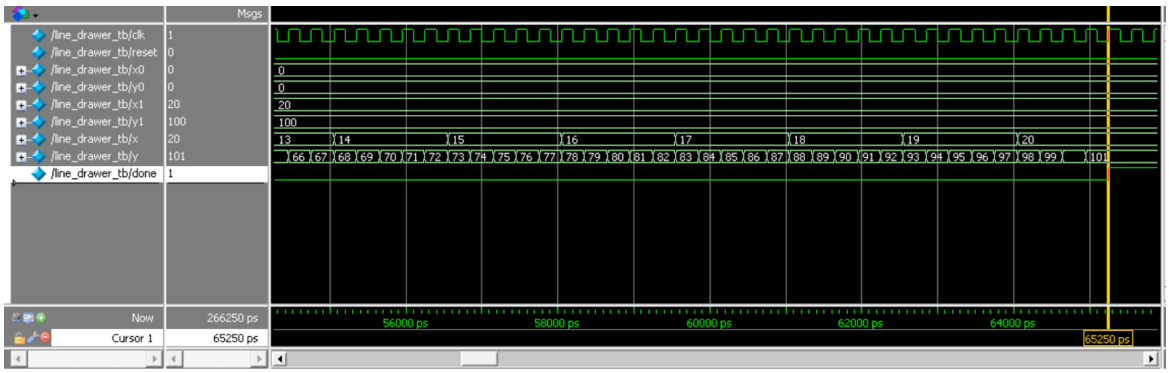
Simulations (Task 2)

According to the test cases in line_drawer_tb.sv, the simulations are below according to the formula $y=y_1+slope*(x-x_1)$.

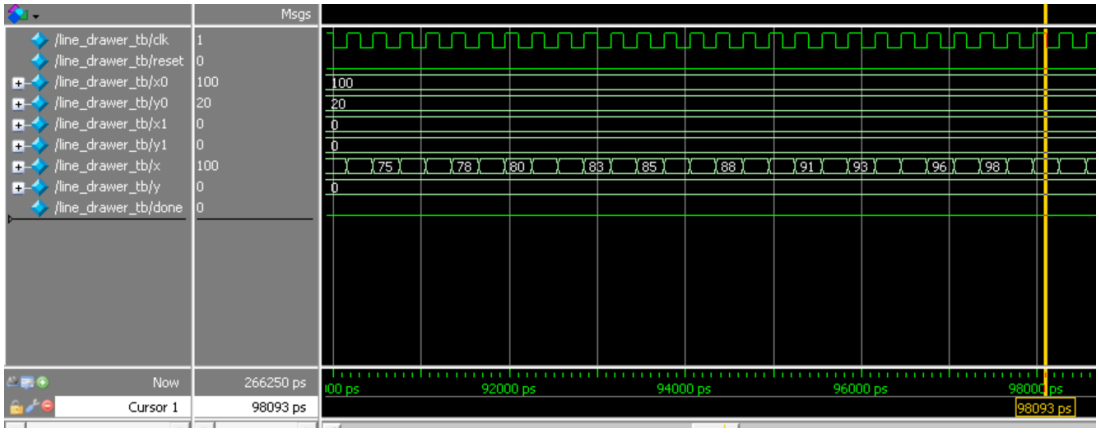
1. Gradual, Left-Up



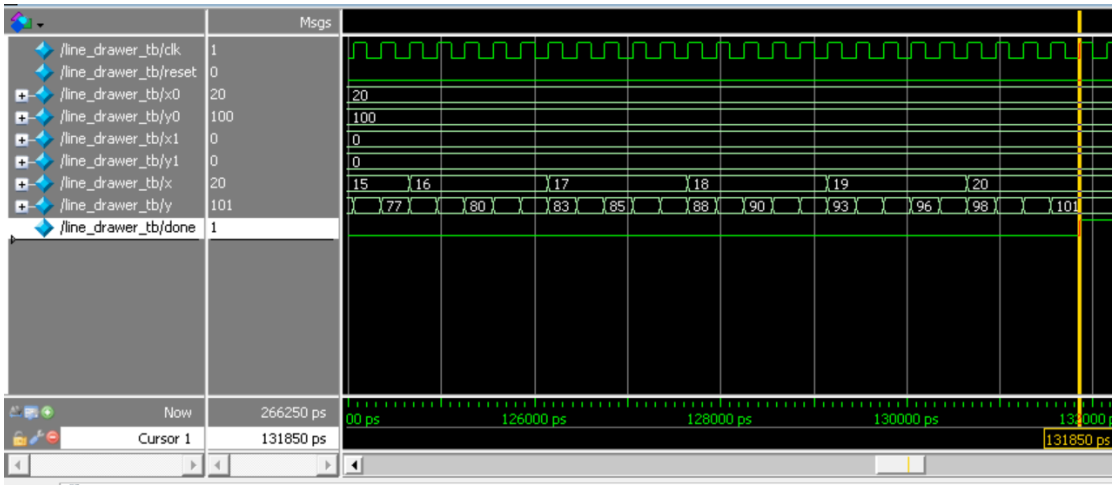
2. Steep, Left-Up



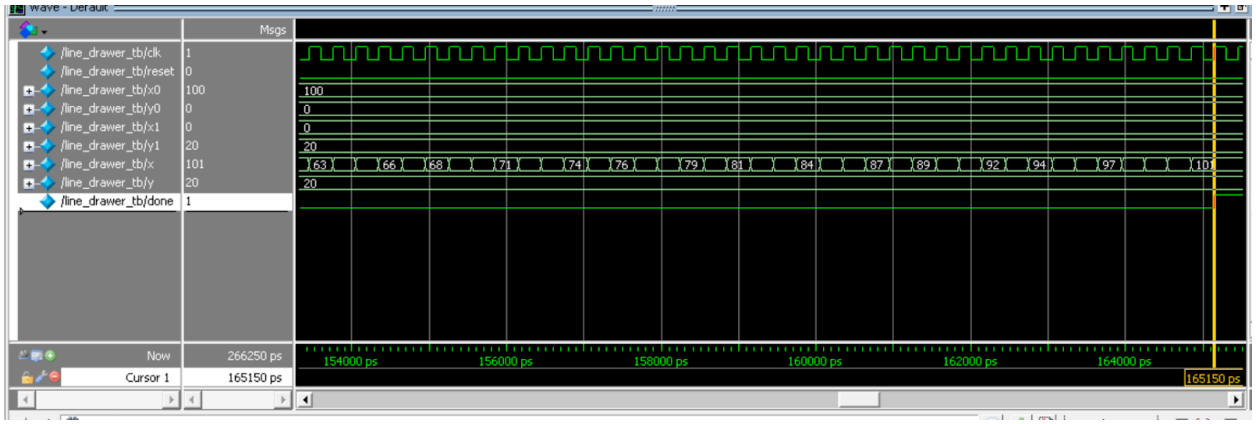
3. Gradual, Right-Down



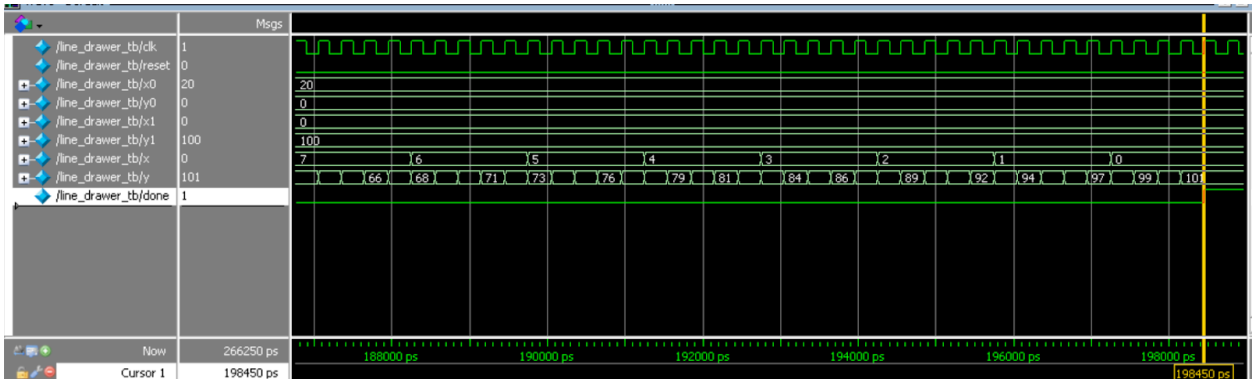
4. Steep, Right-Down



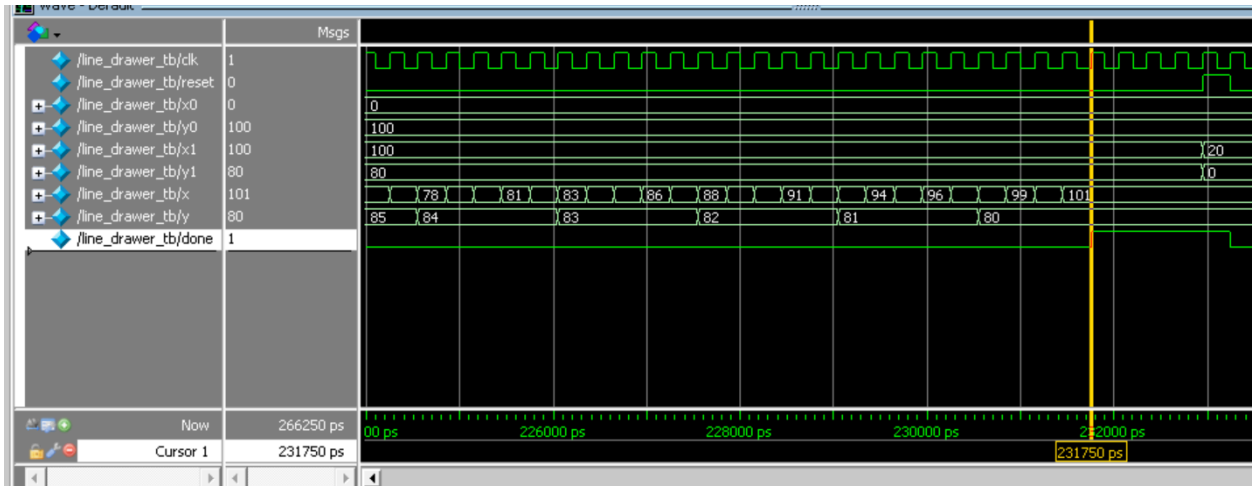
5. Gradual, Right-Left, Up



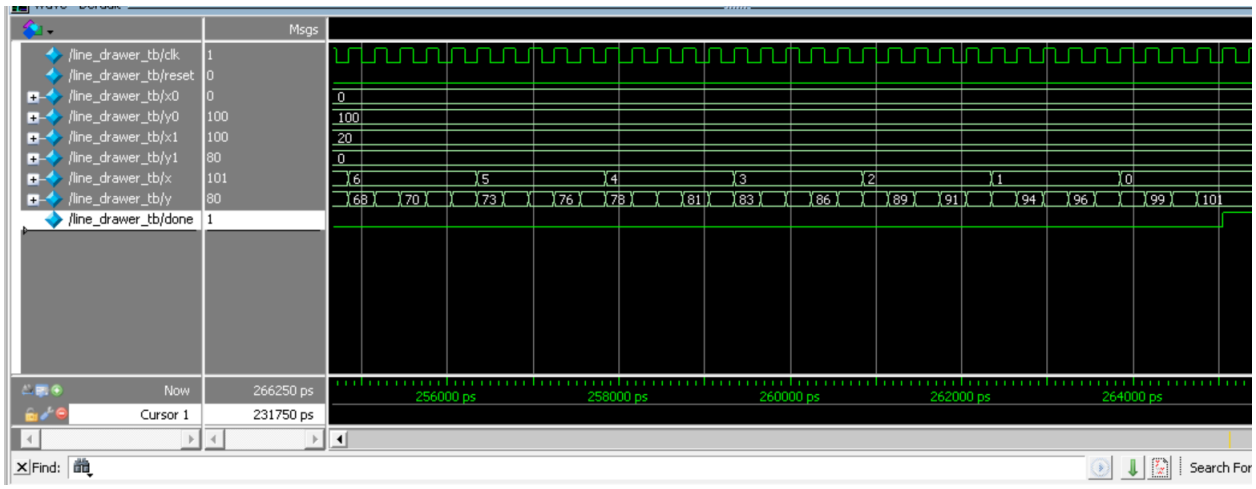
6. Steep, Right-Left, Up



7.Gradual, Left-Right, Down



8. Steep, Left-Right, Down



Experience Report

Feedback

This lab was the best one so far. It was cool to understand how such a prolific video interface works. It’s a little sad we didn’t get to use a physical board for this, since I would have loved to see it actually appear on my display.

Significant Issues

Nothing major. The only frustrating thing is that one of the boards on labsland doesn’t actually have a VGA display show up even when you select VGA.

Tips/Tricks

- show state progression on the LEDs
- use Labsland as an IDE for quick debugging if you don’t have a windows machine
- sometimes all you need is a state machine

Time Spent

9 hours