Note to Instructor:

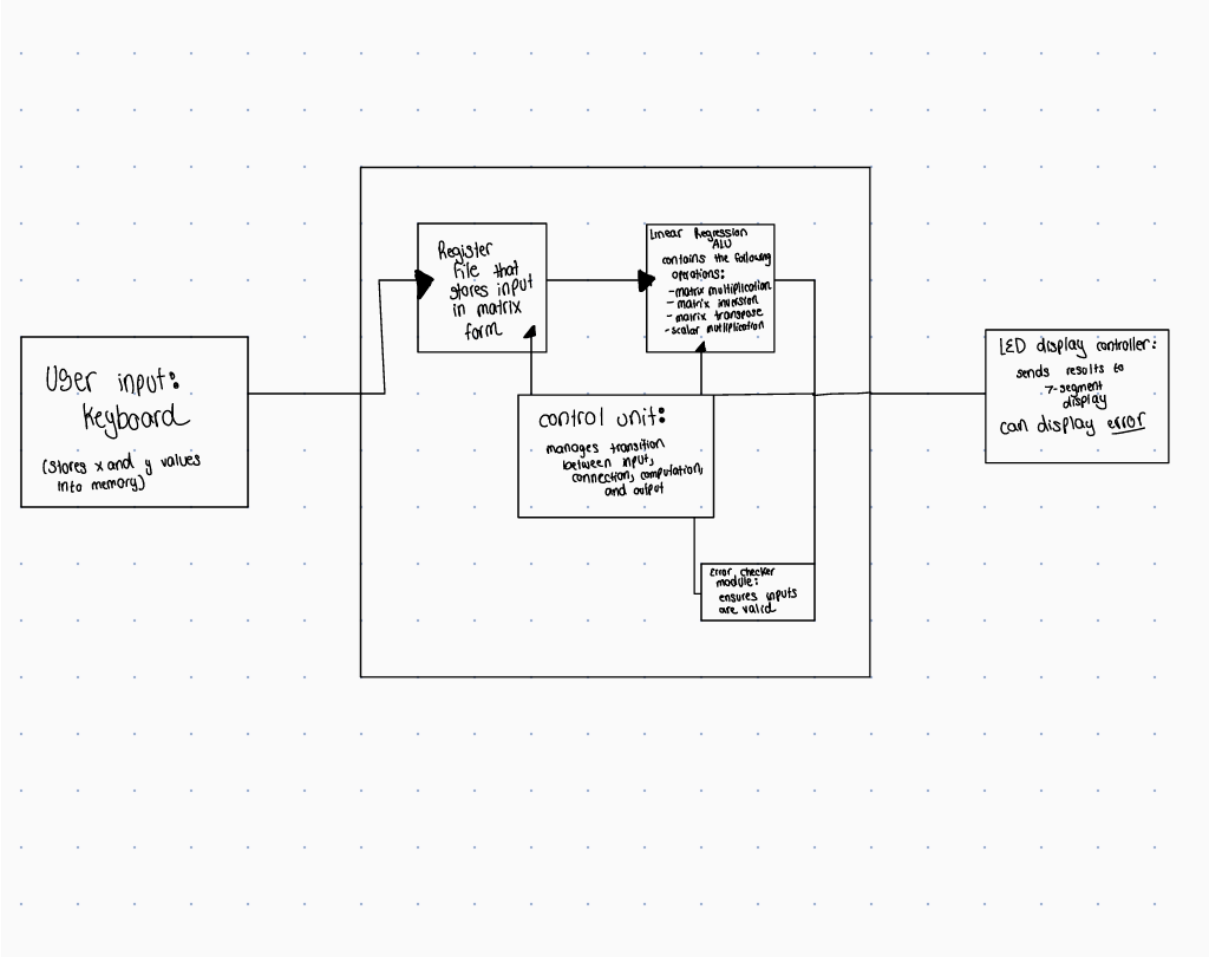
The FPGA fully synthesizes on the FPGA. It's too big to synthesize on the Github Actions.

Here is a video of the DEMO to prove it synthesizes:
https://youtube.com/shorts/59fyLhhhODE?si=ylCrKA2hUuUrgr8e

Project Milestone: Linear Regression Calculator

Overview: This project aims to design a linear regression calculator that will represent the best-fit line based on the given data points from the user.

High Level Datapath:



Core Modules

The primary modules in the design are:

1. **Register File Unit** - This is responsible for collecting user input for the X and Y coordinates.

2. **Linear Regression ALU** - This preforms the required mathematical operations, including matrix multiplication, matrix inversion, and scalar multiplication to compute the best-fit line.
    a. It follows the closed form equation for linear regression.

$$\tilde{\mathbf{w}}^{LMS} = \left(\tilde{\mathbf{X}}^{\top}\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^{\top}\mathbf{y}$$

3. **Control Unit** - This coordinates the system's actions and sequences the operations.
4. **Error Checker -** This checks validity of the input/output values, displaying error if needed
5. **Output**: LED display
6. **Input**: Keyboard

1.1 Input Register Values

The input register captures the X and Y coordinates entered by the user. A keyboard is used to collect these values, with the "E" key indicating an input and the "D" key marking the end of the input sequence. It stores these values in vector form.

Modules:

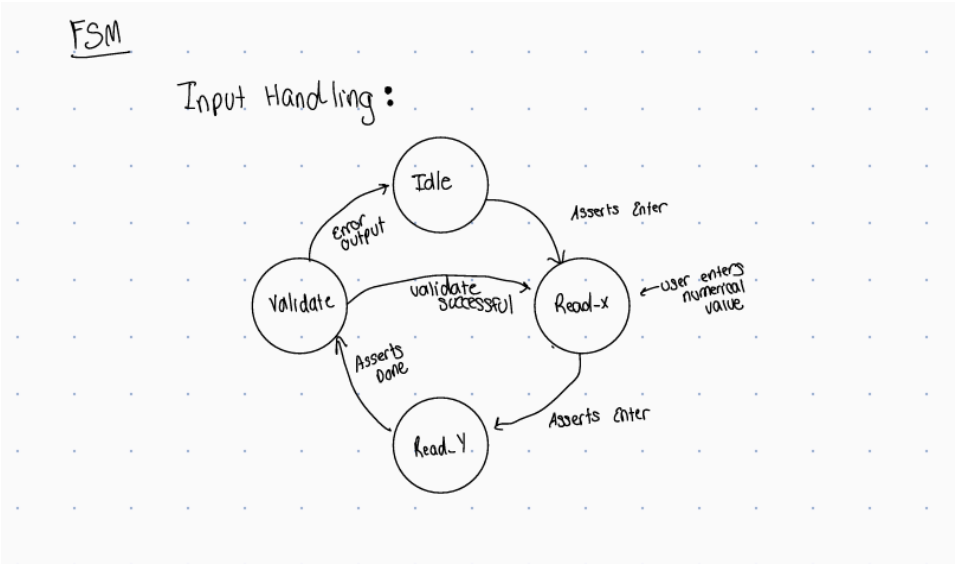
- Keyboard Input (X, Y values)

- Validity Check for Input

FSM:

- Input values are stored in a register file for later processing.

- The system ensures only valid inputs are processed, with error handling in case of invalid data. Invalid data could be wrong coordinates or the same coordinate.



1.2 Linear Regression ALU

This alu performs the computational part of the linear regression calculation. It processes the matrices involved in the closed-form solution for linear regression:

$$\tilde{\mathbf{w}}^{LMS} = \left(\tilde{\mathbf{X}}^{\top}\tilde{\mathbf{X}}\right)^{-1}\tilde{\mathbf{X}}^{\top}\mathbf{y}$$

recover the uni-variate solution that we had computed earlier:

$$w_1 = \frac{\sum(x_n - \bar{x})(y_n - \bar{y})}{\sum(x_i - \bar{x})^2} \qquad \text{and} \qquad w_0 = \bar{y} - w_1\bar{x}$$

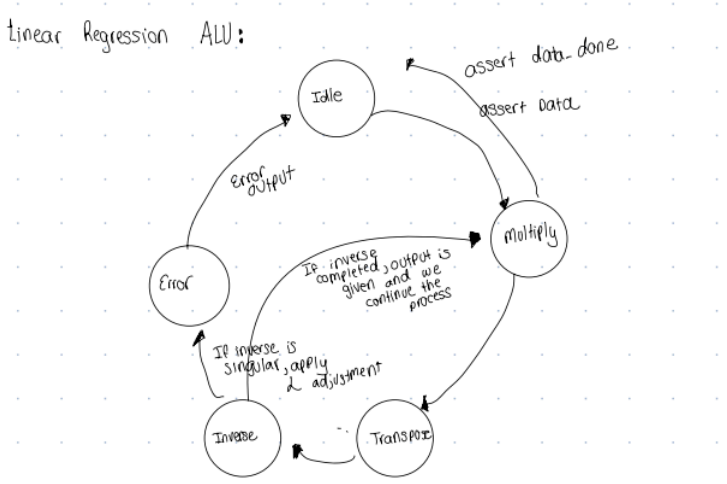
where $\bar{x} = \frac{1}{N}\sum_n x_n$ and $\bar{y} = \frac{1}{N}\sum_n y_n$.

1. Matrix multiplication
2. Matrix inversion (with a check for singularity)
3. Scalar multiplication for scaling the result.
4. If need be, it will also add lambda to ensure the matrix can be inverted

FSM:

- The ALU communicates with the register file that holds the matrix data (X and Y).

- The ALU takes the data in vector form and computes the matrix multiplication in a pipelined manner, doing each computation every clock cycle



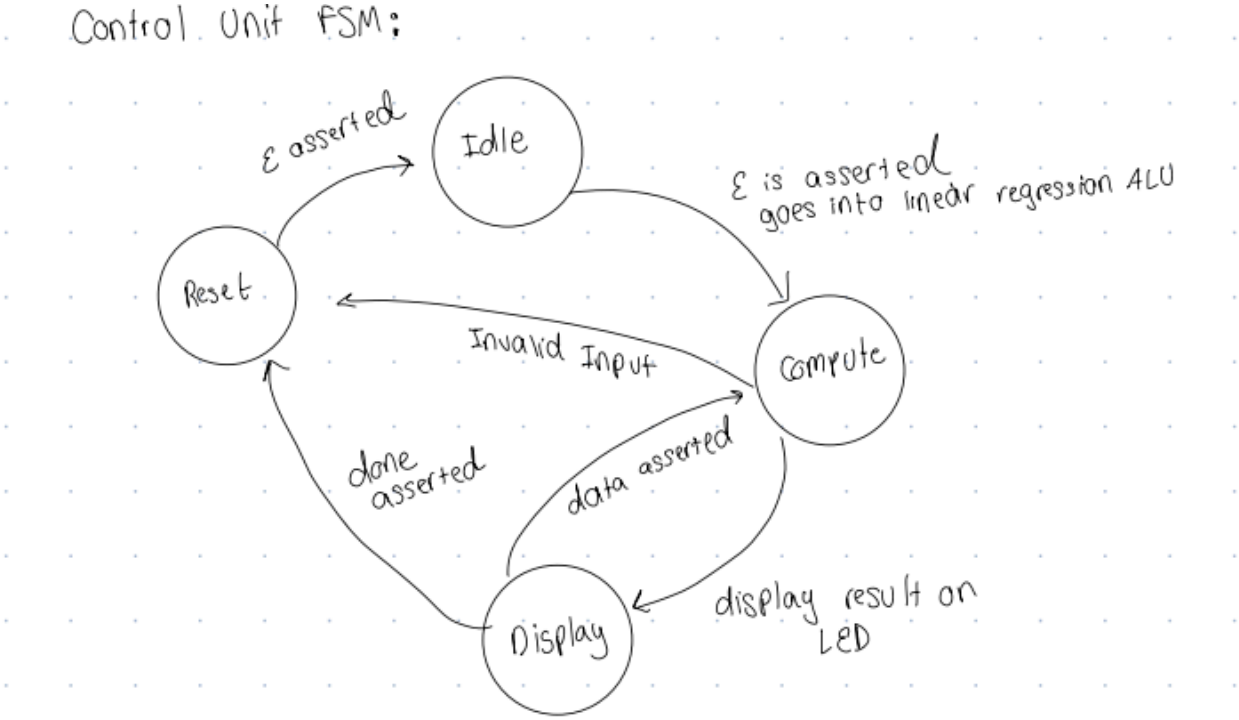
-

1.3 Control Unit

The control unit acts between the entire process, ensuring each module performs its respective function in sequence. It also manages the reset function to clear the system and communicates with the error module.

FSM:

The Control Unit signals the Input Handling Unit, Matrix ALU, and Display Unit based on the current state.



2.1 Testbench Verification:

To verify the functionality of this design, there are various test sets that need to be passed for each module.

Input Register File:

- Verify that the system processes valid X and Y coordinates.
- Verify that the coordinates are valid

- Able to put the X and Y coordinates in vector form for processing
- Able to identity matching coordinates
- Test for non-numeric coordinates

Linear Regression ALU:

- Test the functionality of the matrix operations
- Ensure multiplication,transpose, inverse are working properly
- Ensure that the best-fit line matches in input data
- Able to handle errors (matrix not invertible, needs to add lambda)
- Avoiding division by zero

I would test these by checking functionality with edge cases (non-numeric values, same coordinates) and do unit testing with cocoTB to see my design under various stresses.