VLSI Design Project
ECEN 4303: Digital Integrated Circuit Design

Micah McCombs

December 8, 2023

1 Introduction

A bit slice is a clever technique layout engineers use to simplify VLSI layout, and improve device performance. It involves trying to modularize a design, so that there is an easy path for data to travel through. This is almost essential for innovation and to ensure performance in total data throughput and increasing bit lengths.

In this project we will use this bit slice technique to create a simple custom data path for an adder and a register. The project success criteria includes creating a transistor level schematic for logic simulation, a xschem schematic for Spice and LVS, and the layout of the custom data path in magic.

2 Strategy for Bit-Slice Design

2.1 Information Overview

As provided with the project, are a few resources that suggest how the custom device might be developed. Of these resources, a Full carry adder is provided. The transistor diagram, as well as the fabrication layout is given. The only concern is that the layout may need to be tested with LVS to verify the device is indeed what it is.

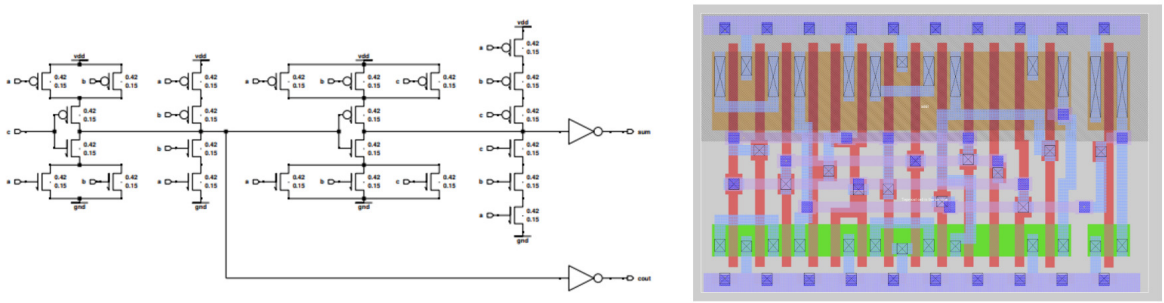

Figure 1: Mirror Adder, Transistor Sch. left, Layout right

Also, a transistor schematic for a D-flip flop, is provided; it utilizes transmission gates in its design, however it lacks any enable based control, which is a design constraint. The schematic, if valid, will need to be designed in layout, and tested further. There is a sample image of the layout for this design, however it still needs to be built from scratch, then modified for the enable functionality.

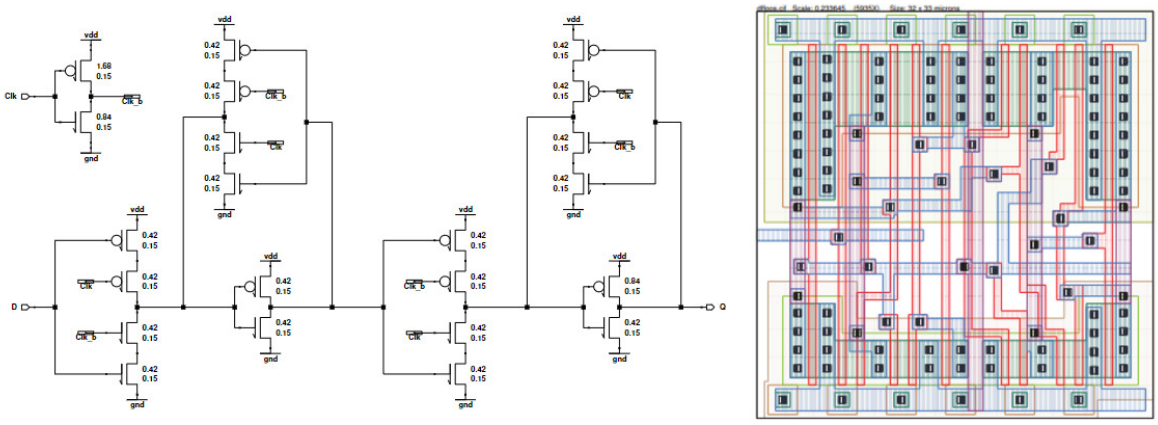

Figure 2: DFF, Transistor Sch. left, Layout right

As for the final required component for the design, the Multiplexer (Mux), has a sample, gate based idea of how it might be built. Therefore, it will be necessary to build the entire Mux, Gate level, Transistor level, stick diagram, and built in layout; as well as testing along the way.

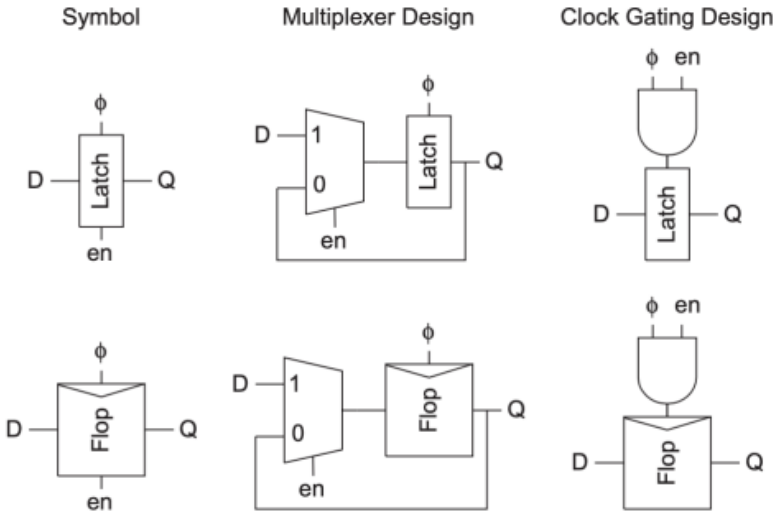

Figure 3: Multiplexer design options

2.2 Objectives

After considering what is given, the strategy will be to understand the larger circuit, and break it done into individual modular components that can be built and tested individually, and then re-integrated back into the larger design, piece by piece.

2.2.1 Mirror Adder

In this top down approach, it is necessary to build and simulate the circuit before even attempting to do layout or LVS. Since the Adder was given, it will be the first part modeled in Sue, and logically tested with IRSIM to verify it indeed, could do 1-bit binary addition.
Once the Adder has been verified for logical functionality, a separate schematic is created in Xschem

for tested at a later time.

2.2.2 D-Flip Flop

The D Flip-flop also had a low transistor count schematic provided, and it is the next device to be built in tested. A secondary design would be necessary to achieve the clock signals, and a third design would also be necessary to achieve the enable based functionality as ordered in the project outline. As before, the D Flip-Flop will also be modeled in Xschem for additional testing.

2.2.3 Multiplexer

When designing the Multiplexer, there are two valid designs: a dual NAND structured device, or a transmission gate approach that will use additional inverters for stability, and output logic. Overall, both perform the exact same functionality, and will be compatible with the rest of the design. However, the Transmission gate approach may offer a lower transistor count, and therefore a simpler design. Unfortunately, the device was first constructed with the NAND approach, and the designed worked without issue, thus it was not replaced.

3 Circuit Design

Below is the overall circuit design

3.1 Top Level

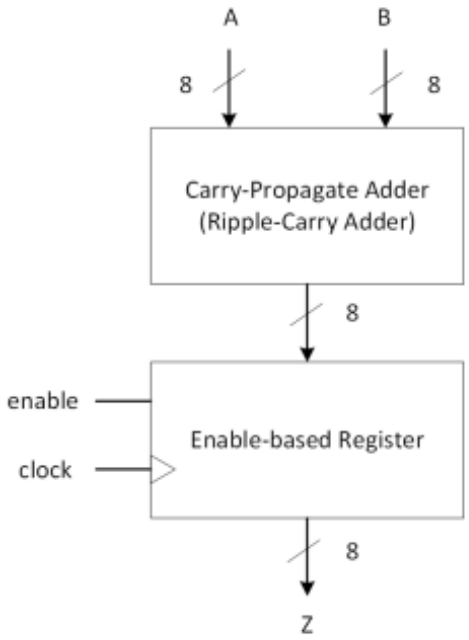

Figure 4: Block Diagram of Top level

3.2 Transistor Level

3.2.1 Full Adder

Here is the Full Adder Layout in Sue.

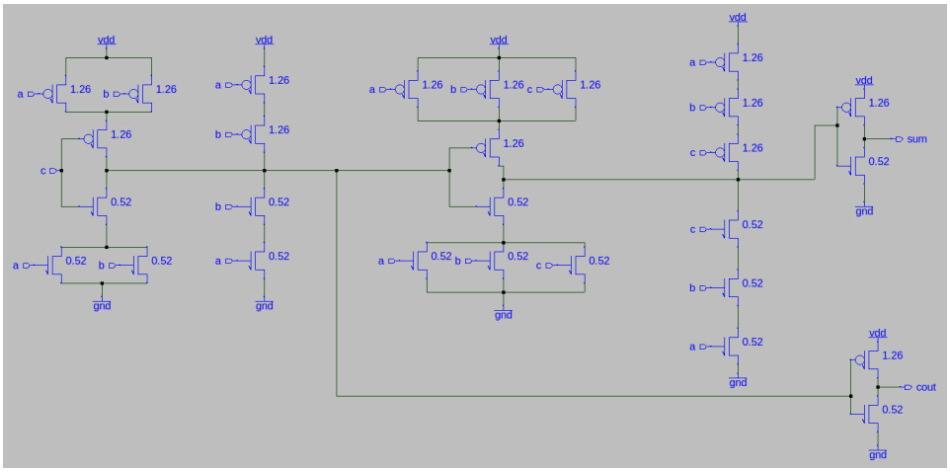

Figure 5: Full Adder in Sue

Full Adder Testing with IRSIM

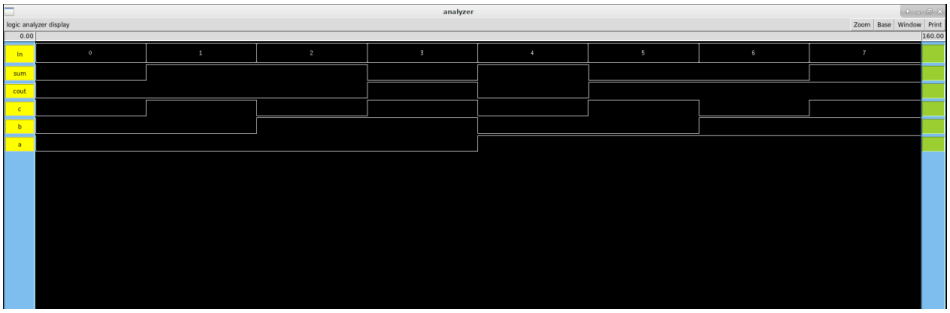

Figure 6: Full Adder IRSIM

3.2.2 D Flip-Flop

Here is the D flip flop layout in Sue.

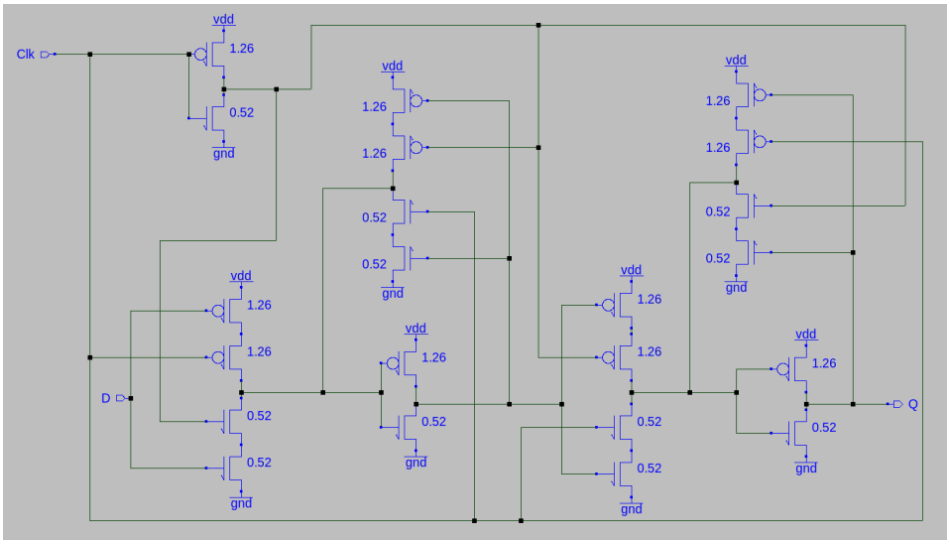

Figure 7: D Flip-Flop w/o EN in Sue

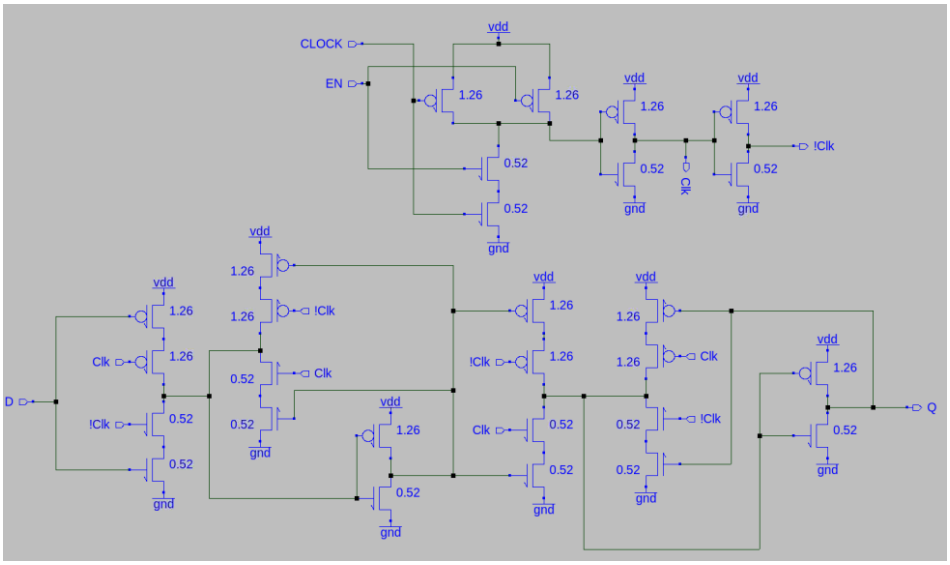

Figure 8: D Flip-Flop with EN, in Sue

D flip-flop with EN, in IRSIM

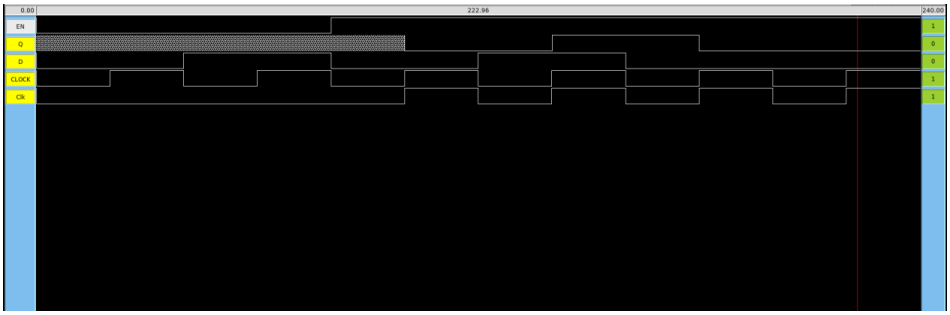

Figure 9: D Flip-Flop simulation

3.2.3 Multiplexer

Here is the Multiplexer layout in Sue.

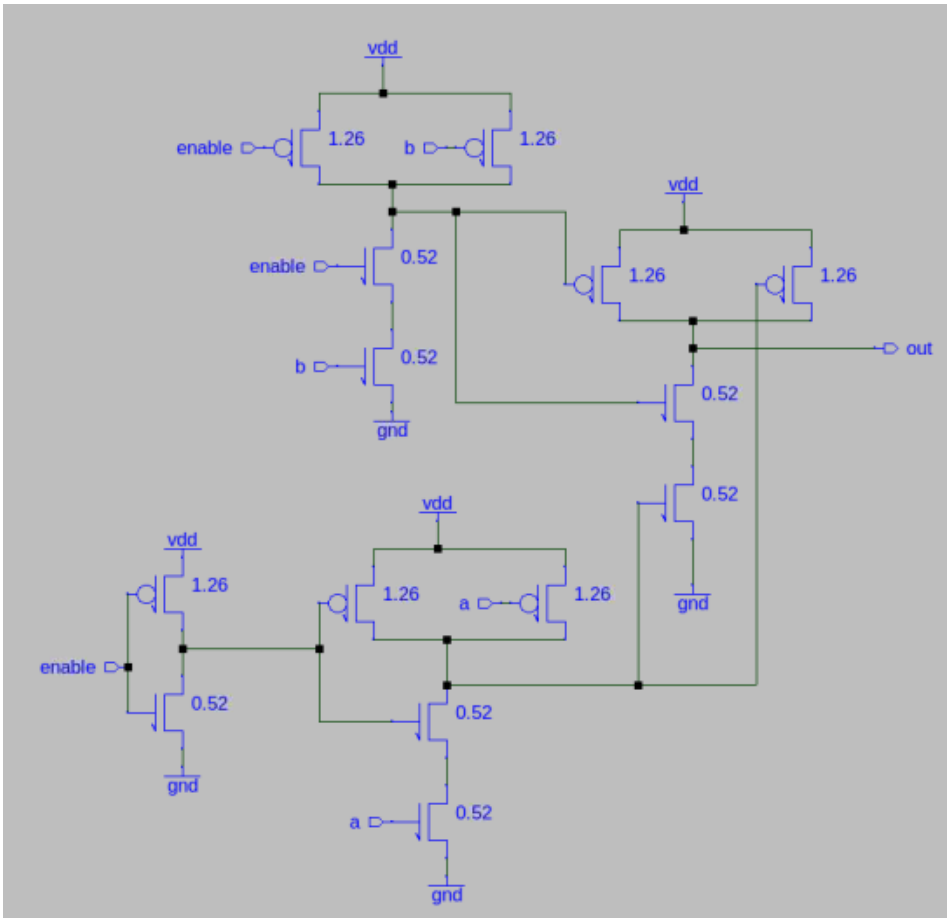

Figure 10: Multiplexer in Sue

Multiplexer simulation in IRSIM

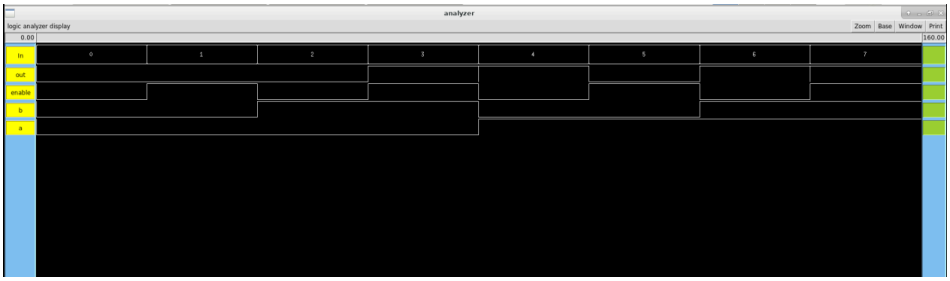

Figure 11: Multiplexer simulation in IRSIM

3.3 1-Bit Path Assembly

1-Bit Path Assembly:

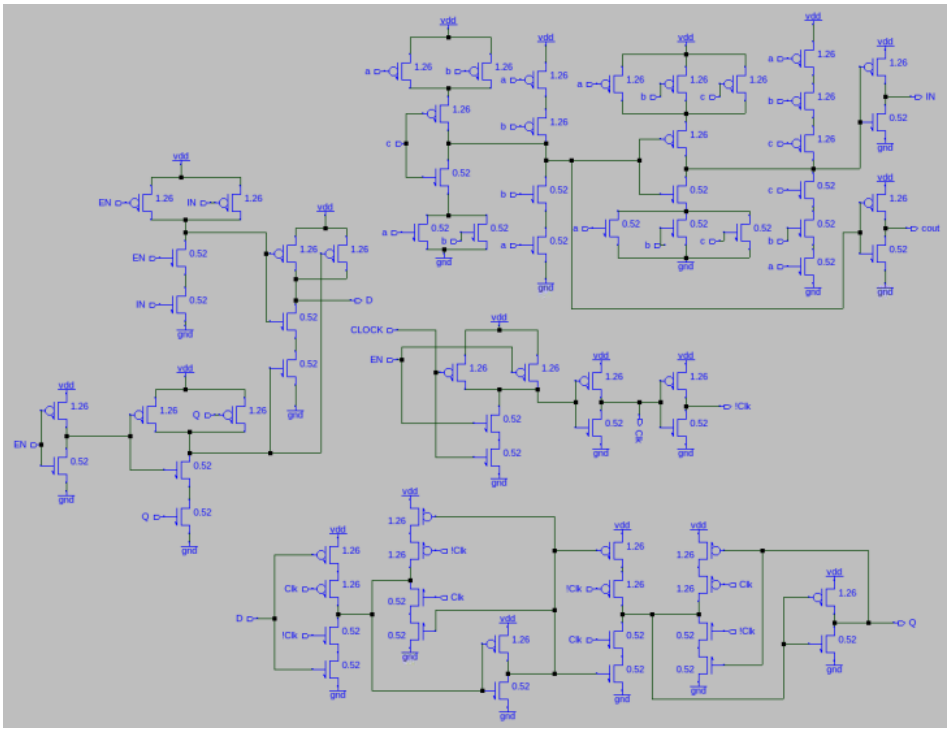

Figure 12: 1-bit path

1-bit path assembly testing:

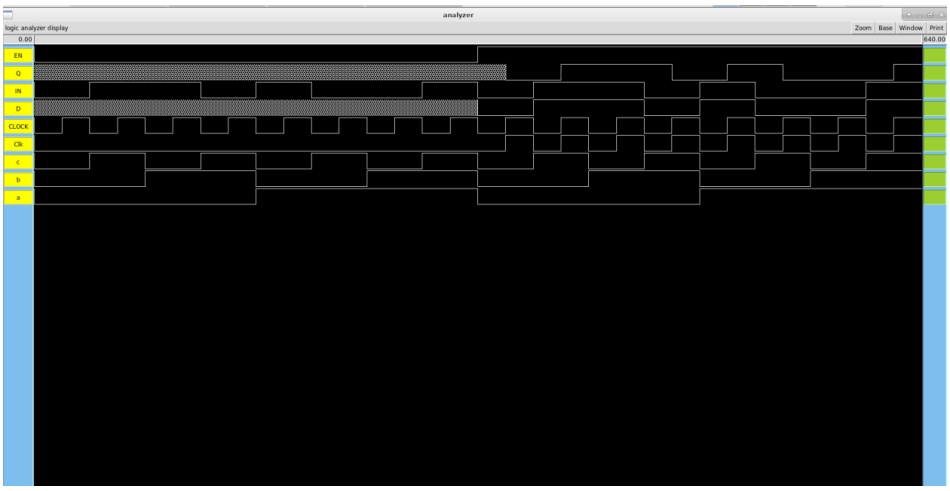

Figure 13: 1-bit path

4 Xschem

4.1 Adder

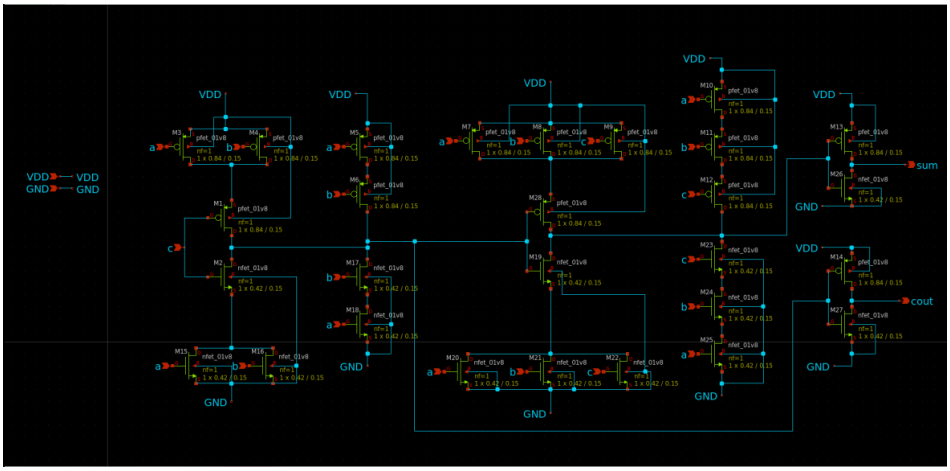

Figure 14: Adder in Xschem

4.2 D Flip-Flop with Enable

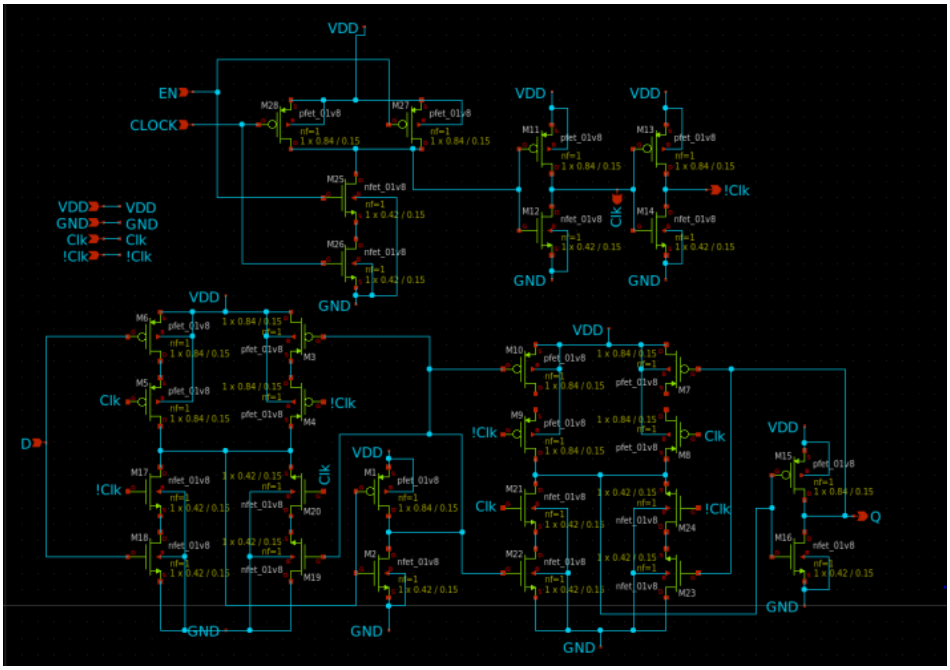

Figure 15: FF in Xschem

4.3 Multiplexer

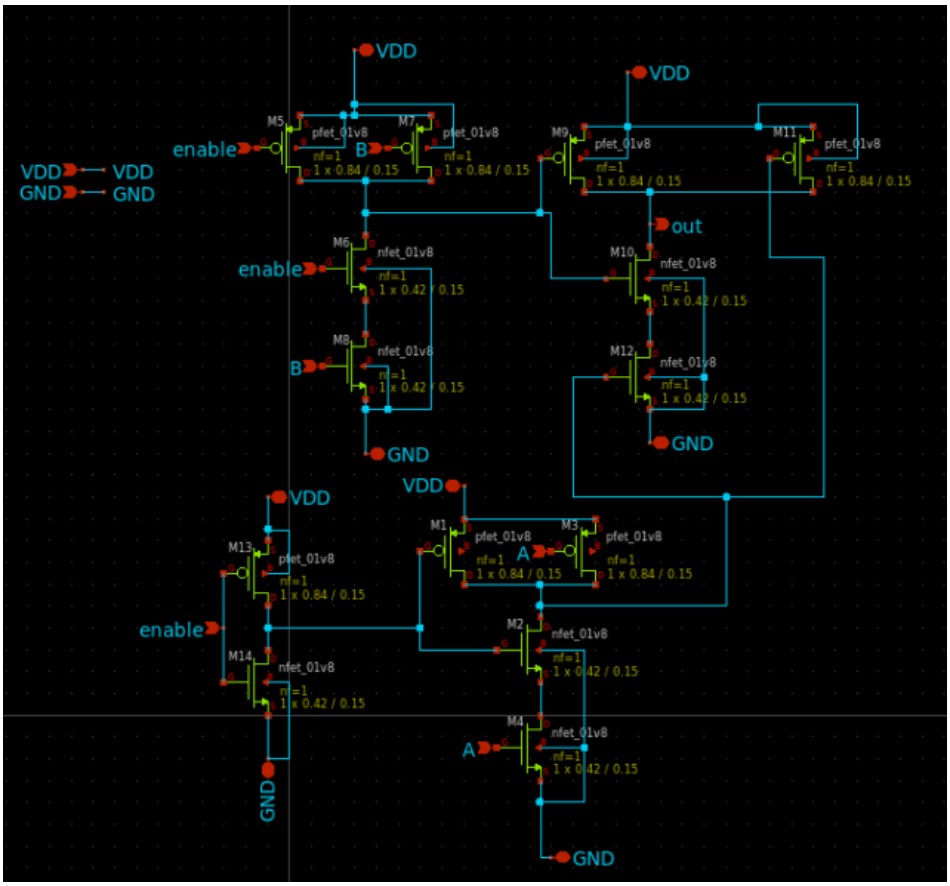

Figure 16: Multiplexer in Xschem

5 Layout

In order to perform well and minimize layout errors, we can utilize a process of stick diagramming. We will use this to help visualize how a transistor level design can be recreated in layout. Before starting layout, we will walk through the simple stick diagrams for each of the project's components.

5.1 Stick Diagrams

5.1.1 Adder

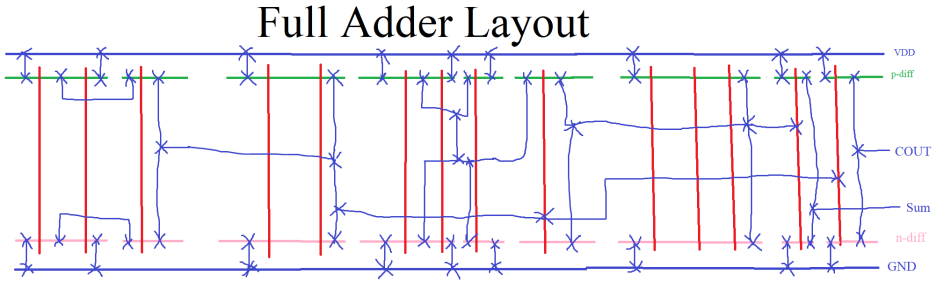

Figure 17: Stick Diagram of Adder

5.1.2 D-ff

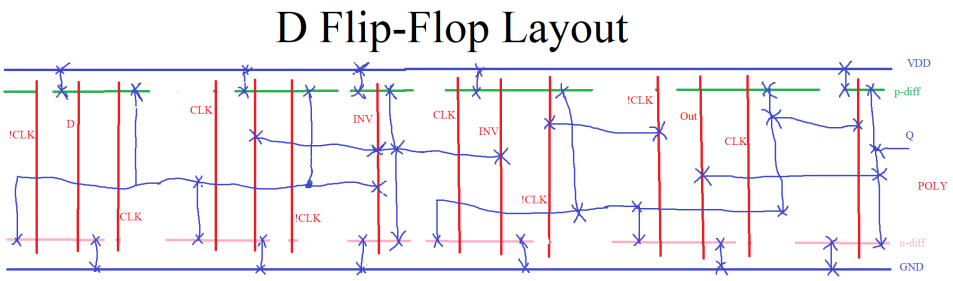

Figure 18: Stick Diagram of DFF

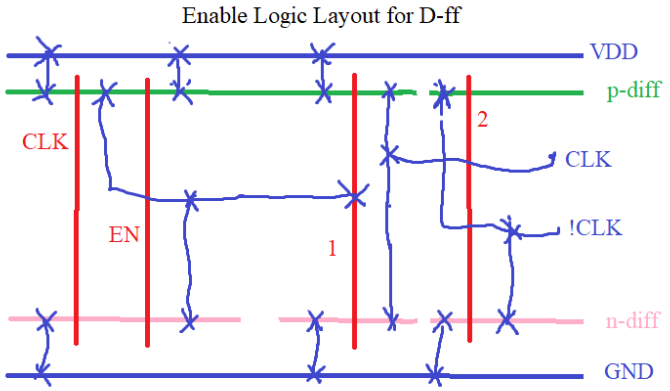

Figure 19: Stick Diagram of Enable for DFF

5.1.3 Multiplexer

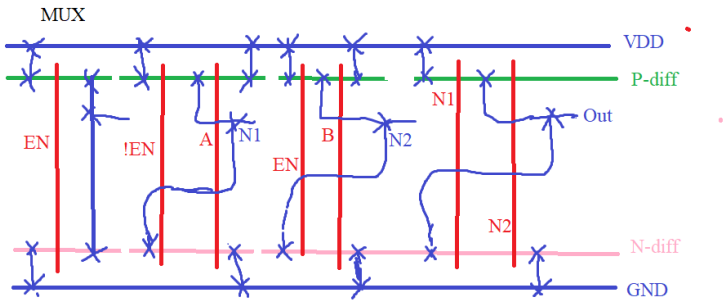

Figure 20: Stick Diagram of Multiplexer

5.2 Magic Layouts

5.2.1 Adder

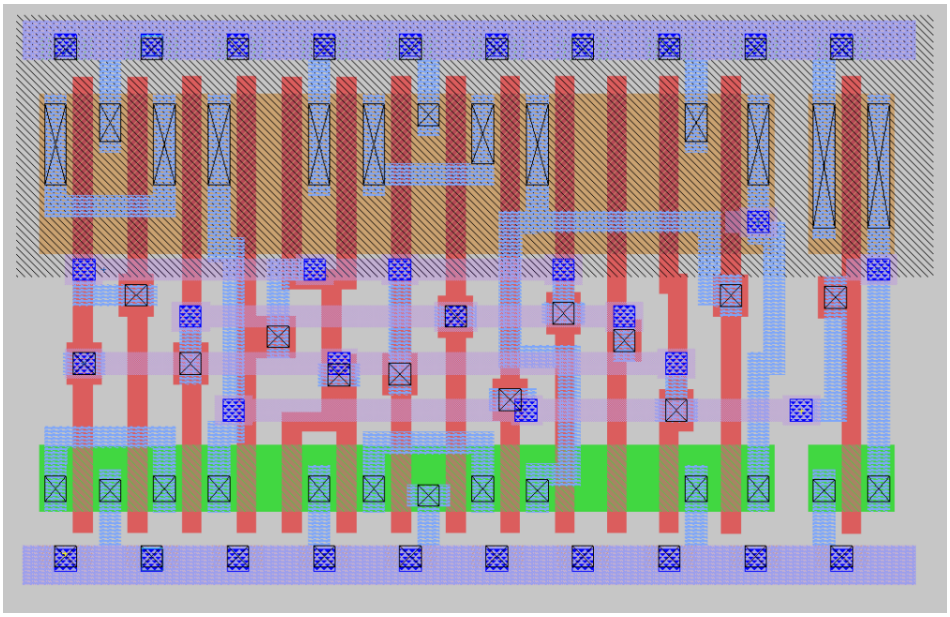

Figure 21: Layout of Adder

5.2.2 D-flip flop

The D-flip flop is expected to be modeled after the stick diagram.

5.2.3 Multiplexer

The Multiplexer is expected to be modeled after the stick diagram.

6 Files and Documents

6.1 Repository

The project files will be submitted with this document, however more can be found on the project repository:

https://github.com/mimccom/4303-VLSI-Final-Project

The project can be cloned with:

git clone https://github.com/mimccom/4303-VLSI-Final-Project.git

7 Conclusions

Utilizing the bit slice technique, this project aimed to forge a streamlined custom data path for an adder and a register. Success was contingent upon generating intricate schematics for logic simulation, ensuring compliance through Spice and LVS with xschem, and culminating in the layout of a bespoke data path in VLSI Magic. This endeavor aimed not only for functionality but also for efficiency, harnessing the power of modular design to optimize performance.