The George Washington University
School of Engineering and Applied Science
Department of Electrical and Computer Engineering
ECE 4140
Fall 2021

Final Project Report

5-BIT ALU

Jaret Williams – Electrical Engineering
Nathan Pen – Electrical Engineering

Professor Jerry Wu
December 19, 2021

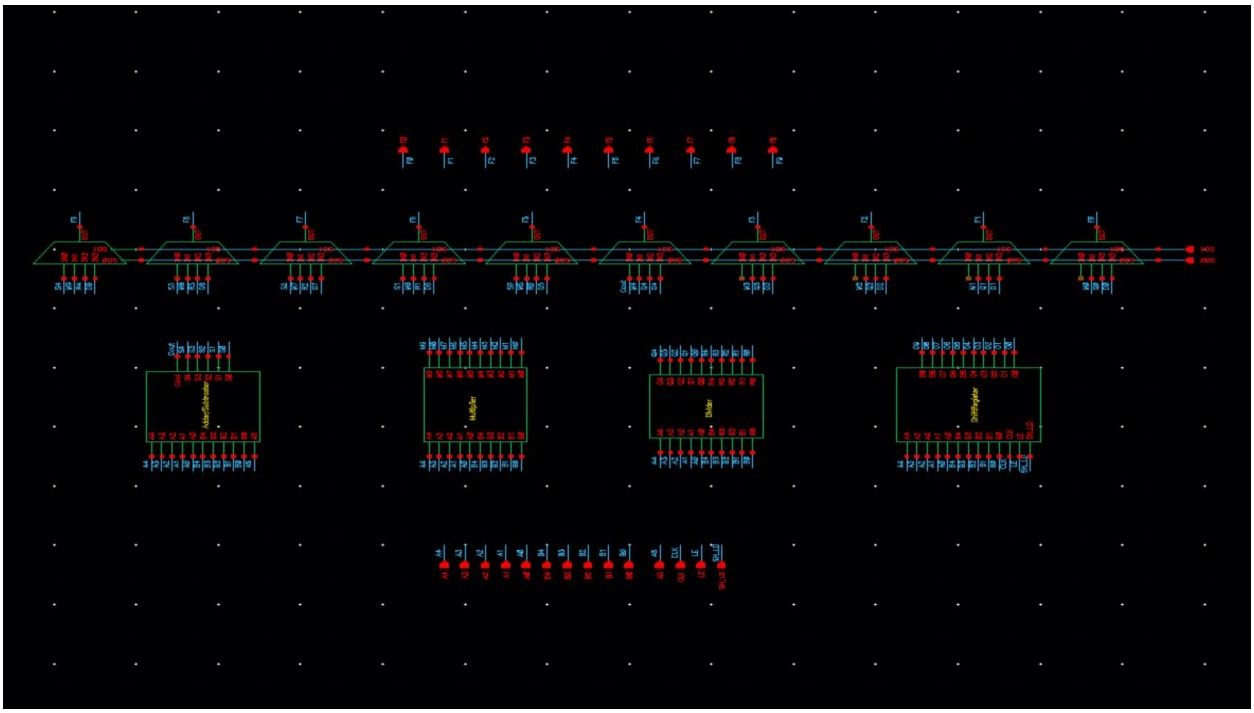
Contents

I. Abstract

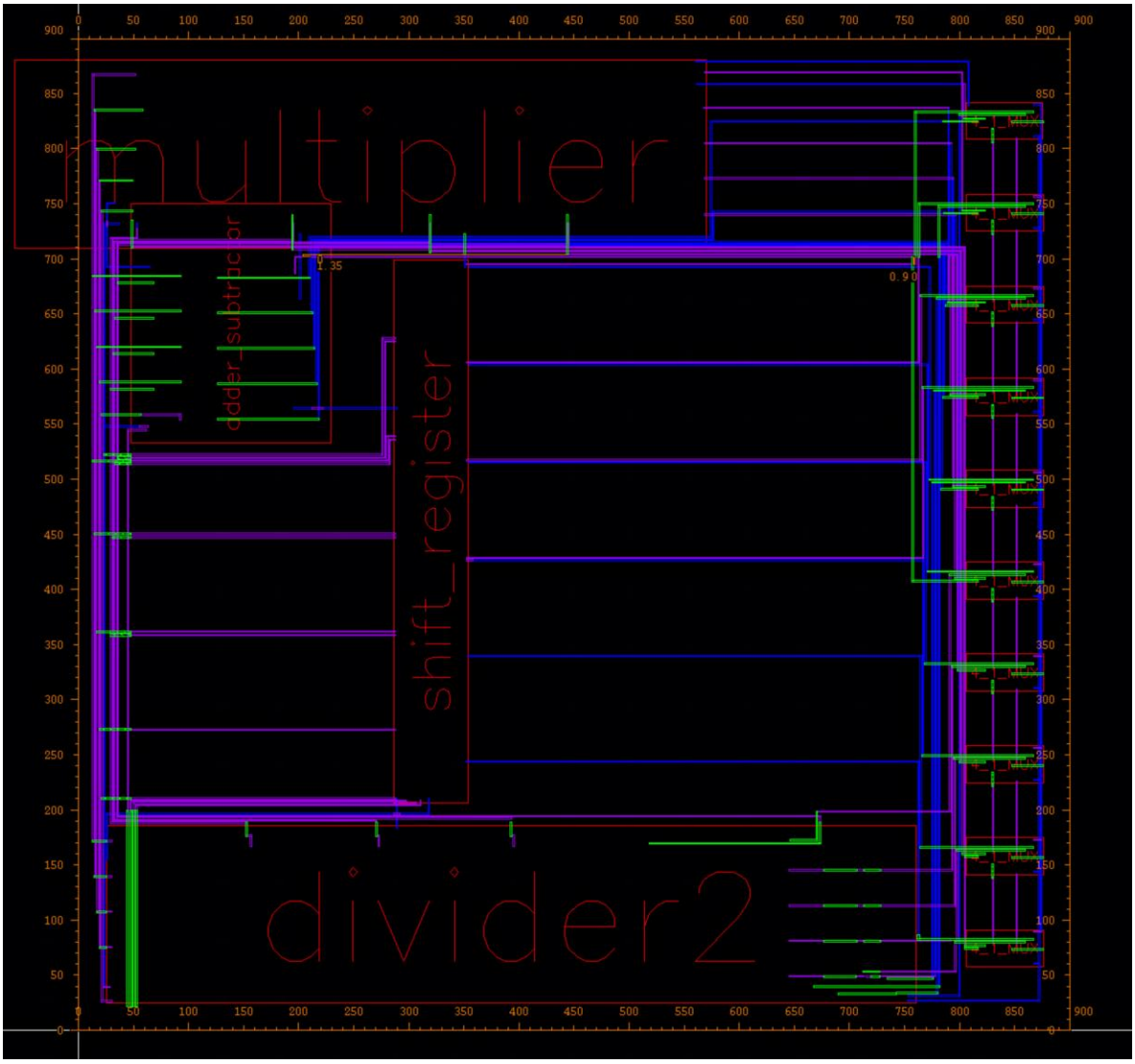
In this project, the students designed, tested, and created a layout for a 5-bit ALU. This chip performs four operations: addition/subtraction, multiplication, division, and shifting. These operations are selected by the user through the control inputs and the corresponding values are sent to the output. ALU's are a central component in central processing units (CPUs) and although this design is smaller than chips for full applications, the proof of concept is there, and functionality was proven by the designers. Please note that this report is being written before chip fabrication, so the team was unable to do physical testing of the chip and limited to simulations in cadence of the designed layout. However, a plan for if the chip is fabricated is described and outlined.

II. Introduction

This project followed each step of the chip design process. This started with a clear understanding of the ALU the team will design and its desired inputs/outputs. Then, the team moved into designing the project on the schematic level (screenshot shown below). The team built and simulated each module individually and then combined all the blocks together and simulated the full system.



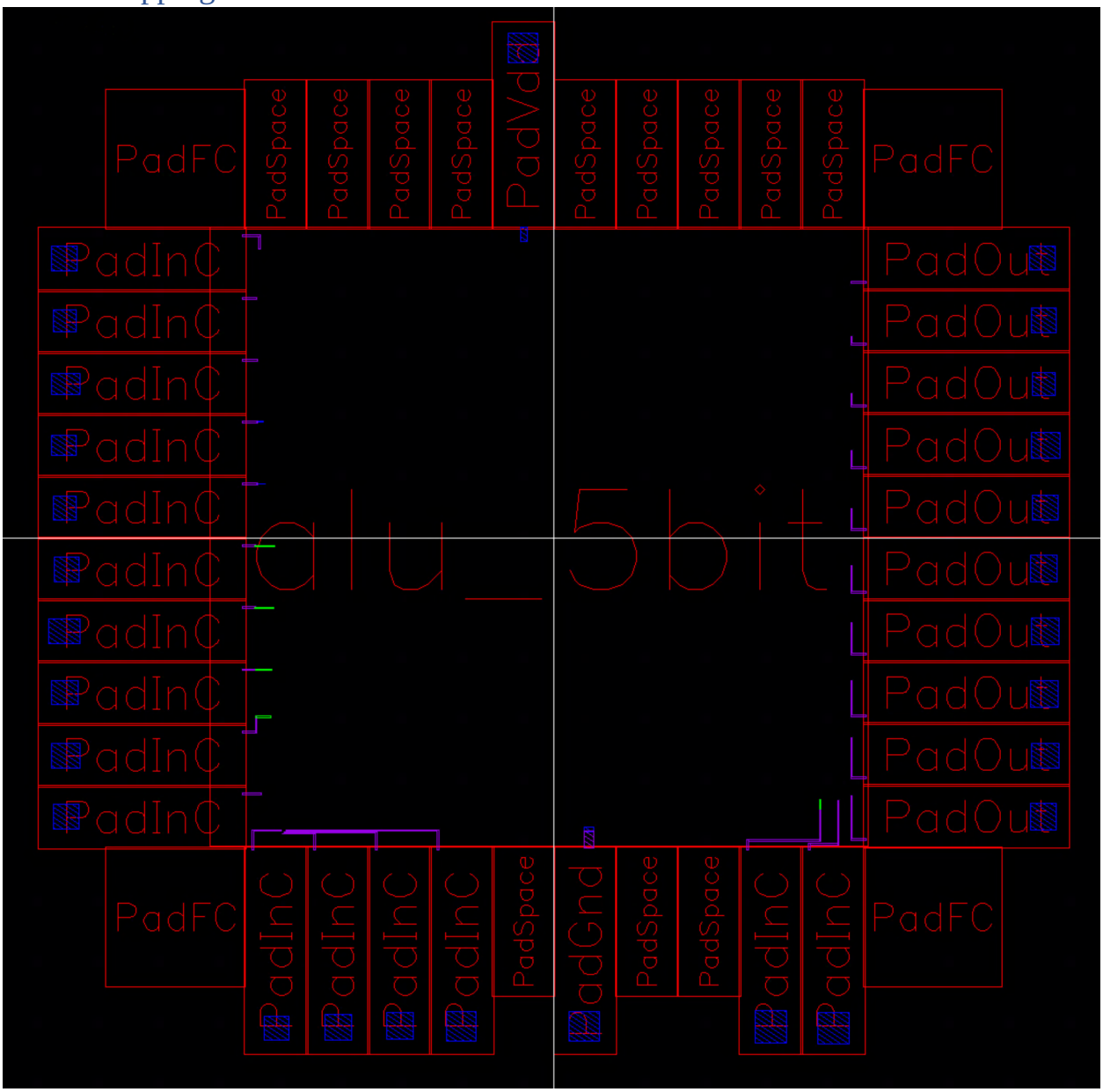
Next, the team designed the layouts for each operation, making sure to keep in mind the 900mm x 900mm pad frame limitations. With LVS matches for each module confirmed, the blocks were then connected together which is shown below.

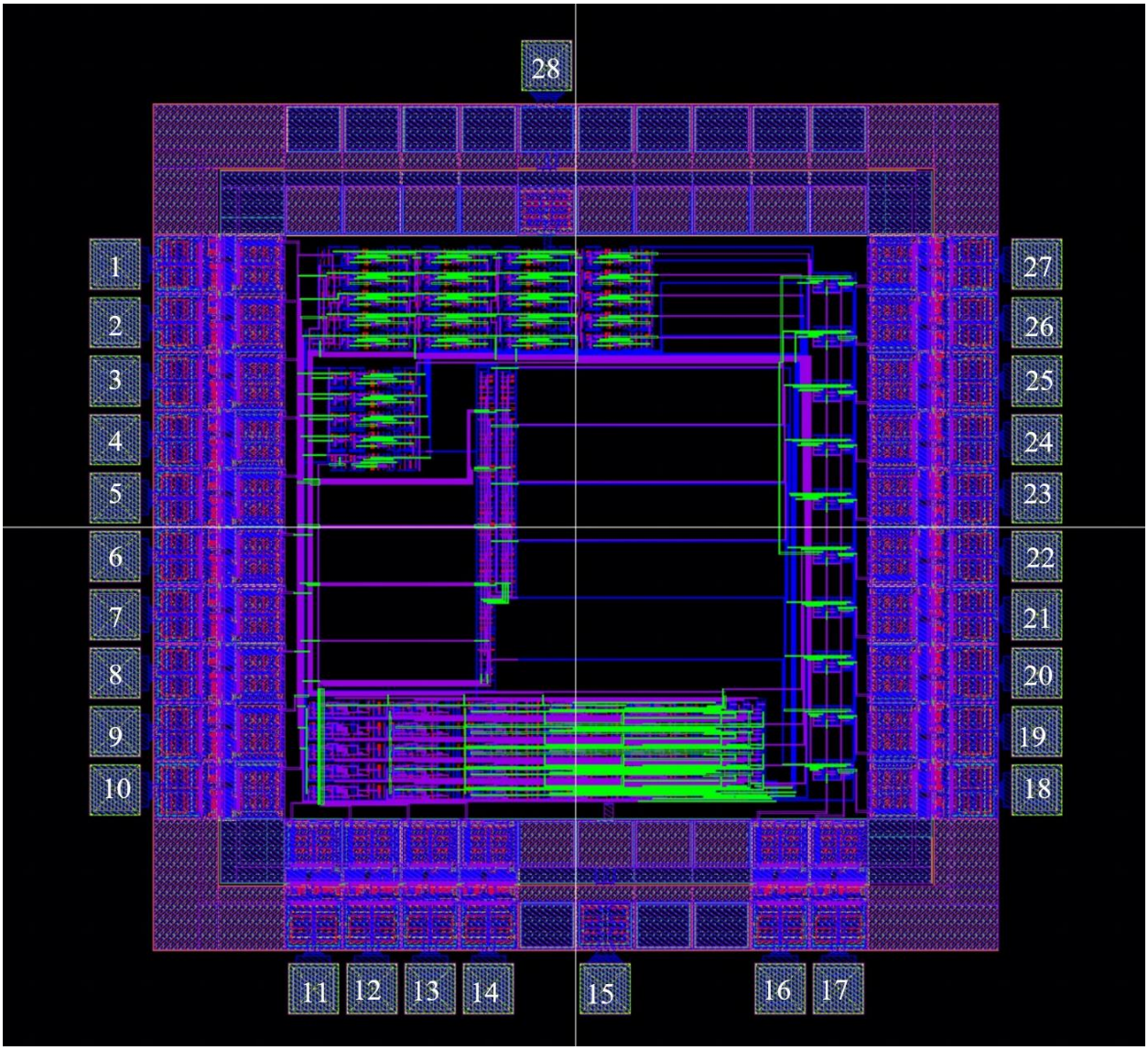


This design was then placed into the pad frame and all the inputs/outputs connected. This step is detailed further in future sections. This project allowed the team to see first hand the of the design flow of a chip and it was a very valuable experience.

III. Spec Sheet Data

Pin Mapping





Pin Number	Pin Name	Pin Function
1	A_4	Input A bit 4
2	A_3	Input A bit 3
3	A_2	Input A bit 2
4	A_1	Input A bit 1
5	A_0	Input A bit 0
6	B_4	Input B bit 4
7	B_3	Input B bit 3
8	B_2	Input B bit 2
9	B_1	Input B bit 1
10	B_0	Input B bit 0
11	AS	Add/Subtract toggle
12	CLK	Clock
13	LE	Load Enable

14	SH_LD	Shift
15	GND	Ground
16	S_1	Control bit 1
17	S_0	Control bit 2
18	F_9	Output bit 9
19	F_8	Output bit 8
20	F_7	Output bit 7
21	F_6	Output bit 6
22	F_5	Output bit 5
23	F_4	Output bit 4
24	F_3	Output bit 3
25	F_2	Output bit 2
26	F_1	Output bit 1
27	F_0	Output bit 0
28	VDD	VDD

(Input/Output)

Remaining Characteristics

VDD = 5V

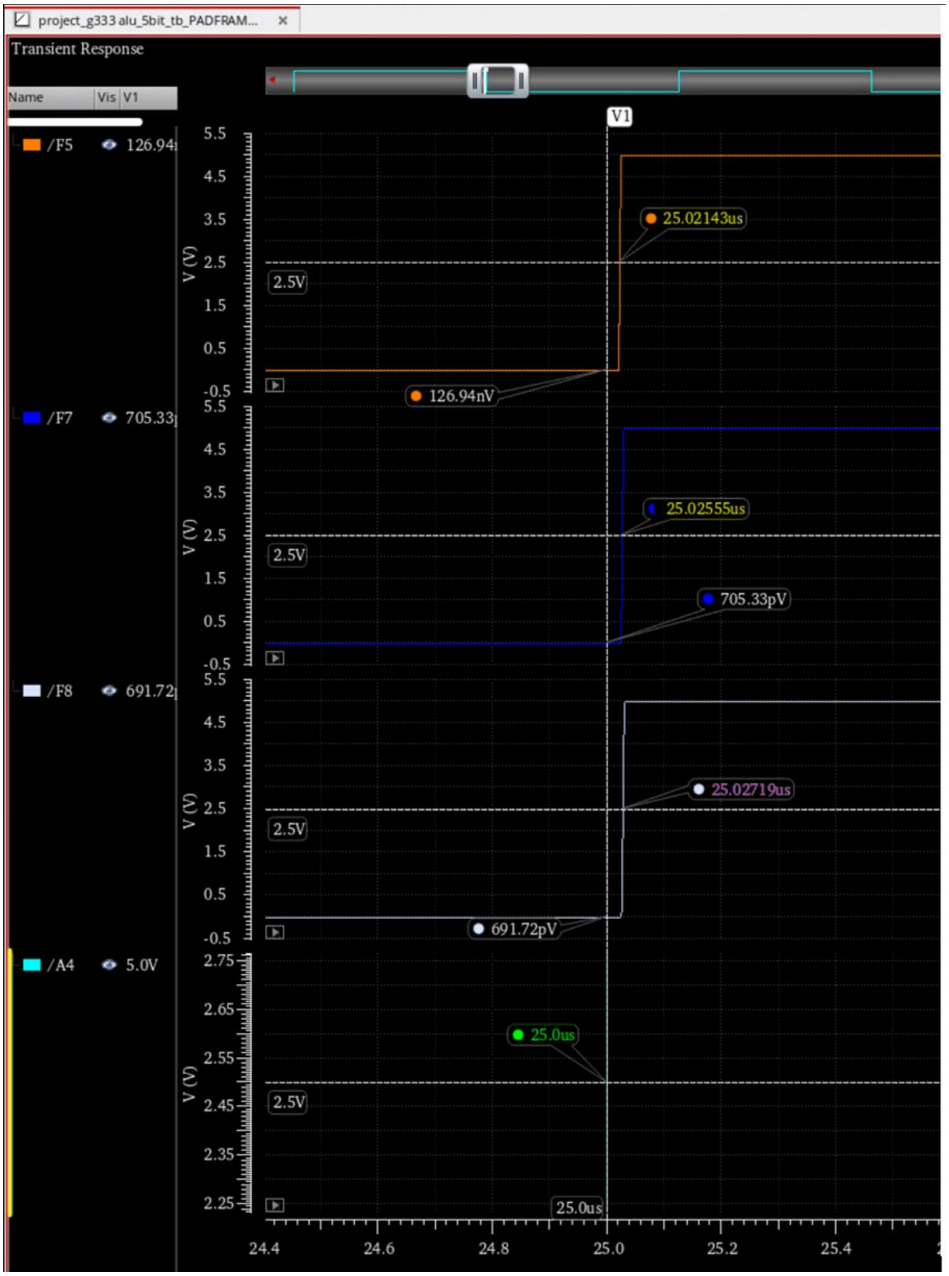
GND = 0V

Max Clock Speed

Longest path is in the divider and goes from A4 to the output. Testing this input specifically guarantees the signal passing through the maximum number of gates before reaching the output.

High to low:

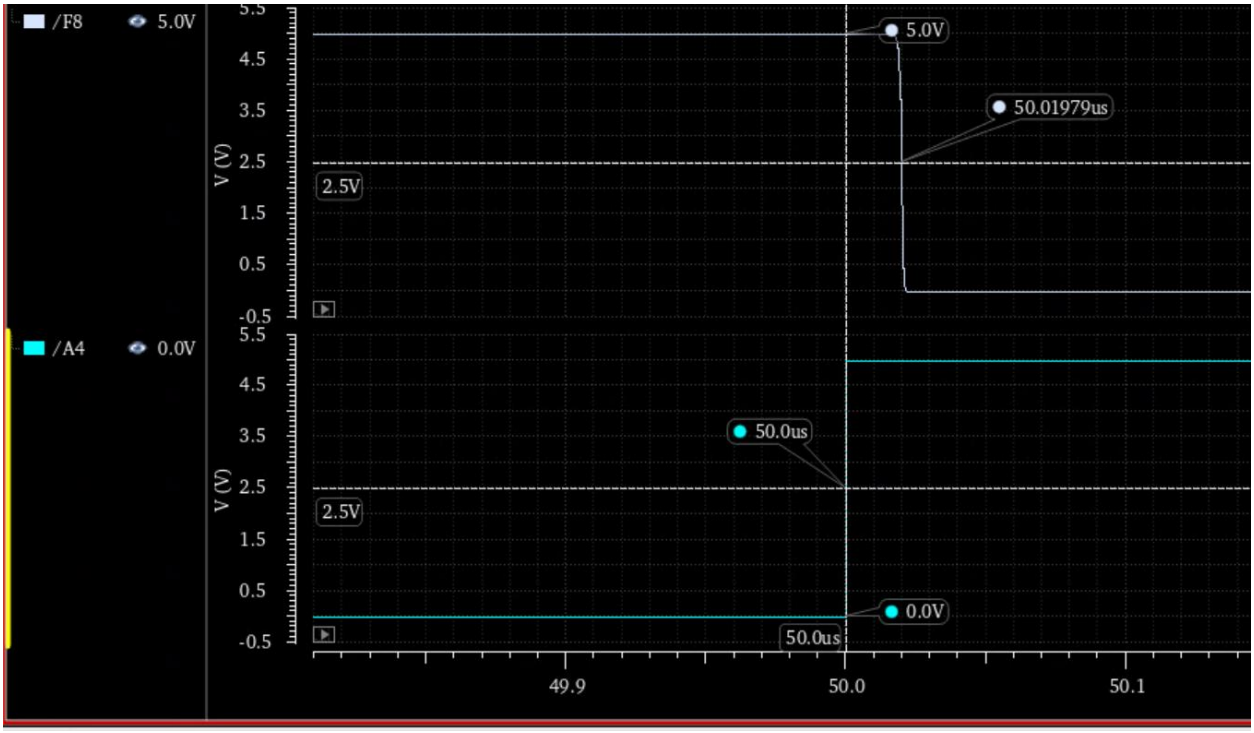
A_4 goes from 5V → 0V



Out of the outputs the changes during the high to low transition of A_4, the worst case is F_8:

t_{PHL} = 25.02719us – 25us = **0.02719us**

Low to High: A_4 goes from 0V → 5V



t_{PLH} = 50.01979us – 50us = **0.01979us**

Using t_{PHL} and t_{PLH}, the overall progation delay t_P is shown below:

$$t_P = \frac{0.02719\mu s + 0.01979\mu s}{2} = \mathbf{0.02349\mu s}$$

Maximum Clock Speed:

$$f_{max} = \frac{1}{t_P} = \frac{1}{0.02349\mu s} = \mathbf{42.571MHz}$$

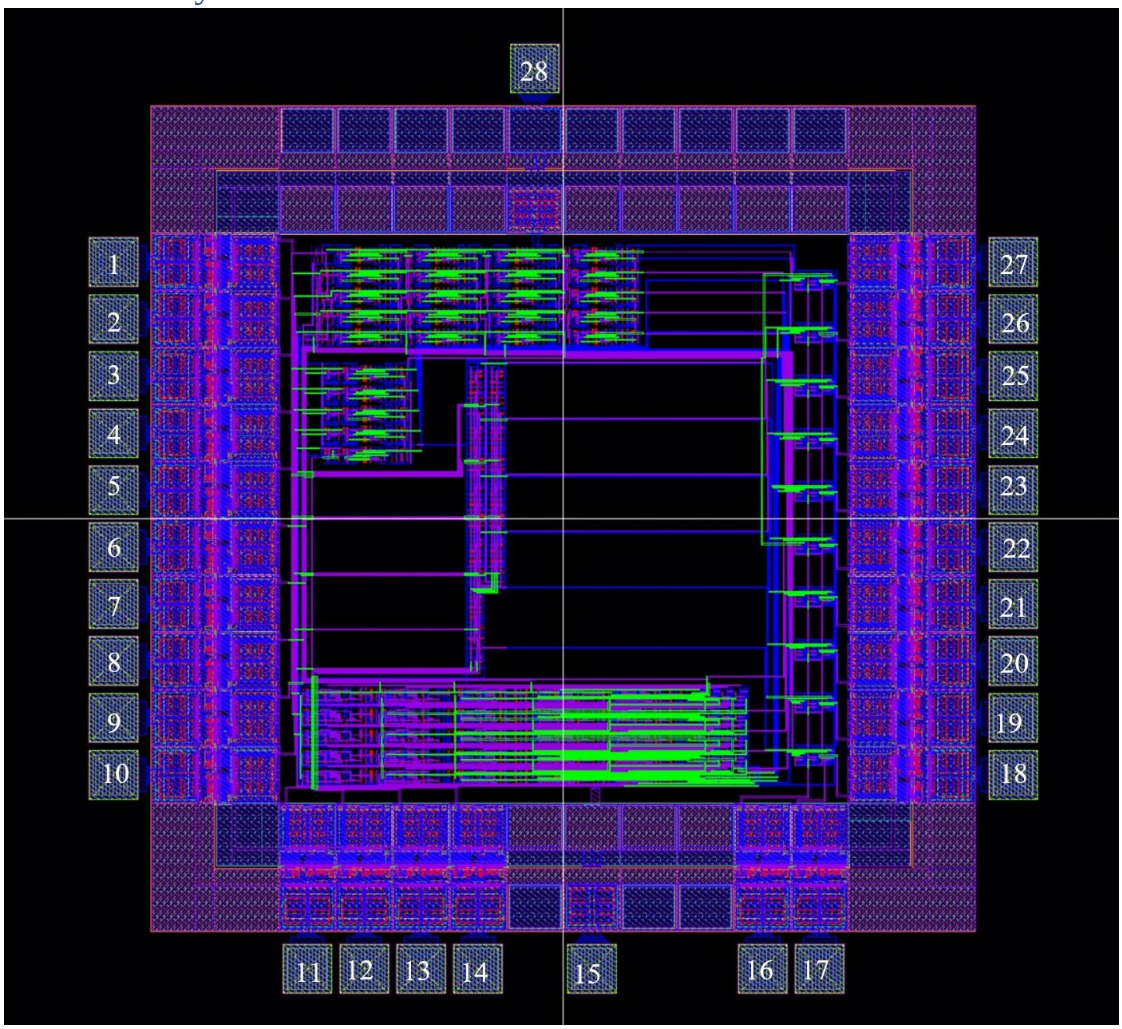
IV. Test Plan

How we intend to test chip is connecting the chip created from the PAD frame to the MOSIS chip housing and then connecting the inputs of the chip to the output of an FPGA test board like the BASYS 3 that was utilized in ECE 3135. The inputs will be defined in the Verilog code that controls the FPGA. The inputs will be the same as the input functions used in the simulation schematics. Then the outputs of the chip can be checked with the oscilloscope to see if the outputs match the desired outputs that were displayed for the simulation. A percent error will then be calculated, and the team will be able to see if anything went wrong during fabrication.

V. Design vs Fabricated Results

No fabricated design currently available so the layout design is shown in this section.

Overall Layout



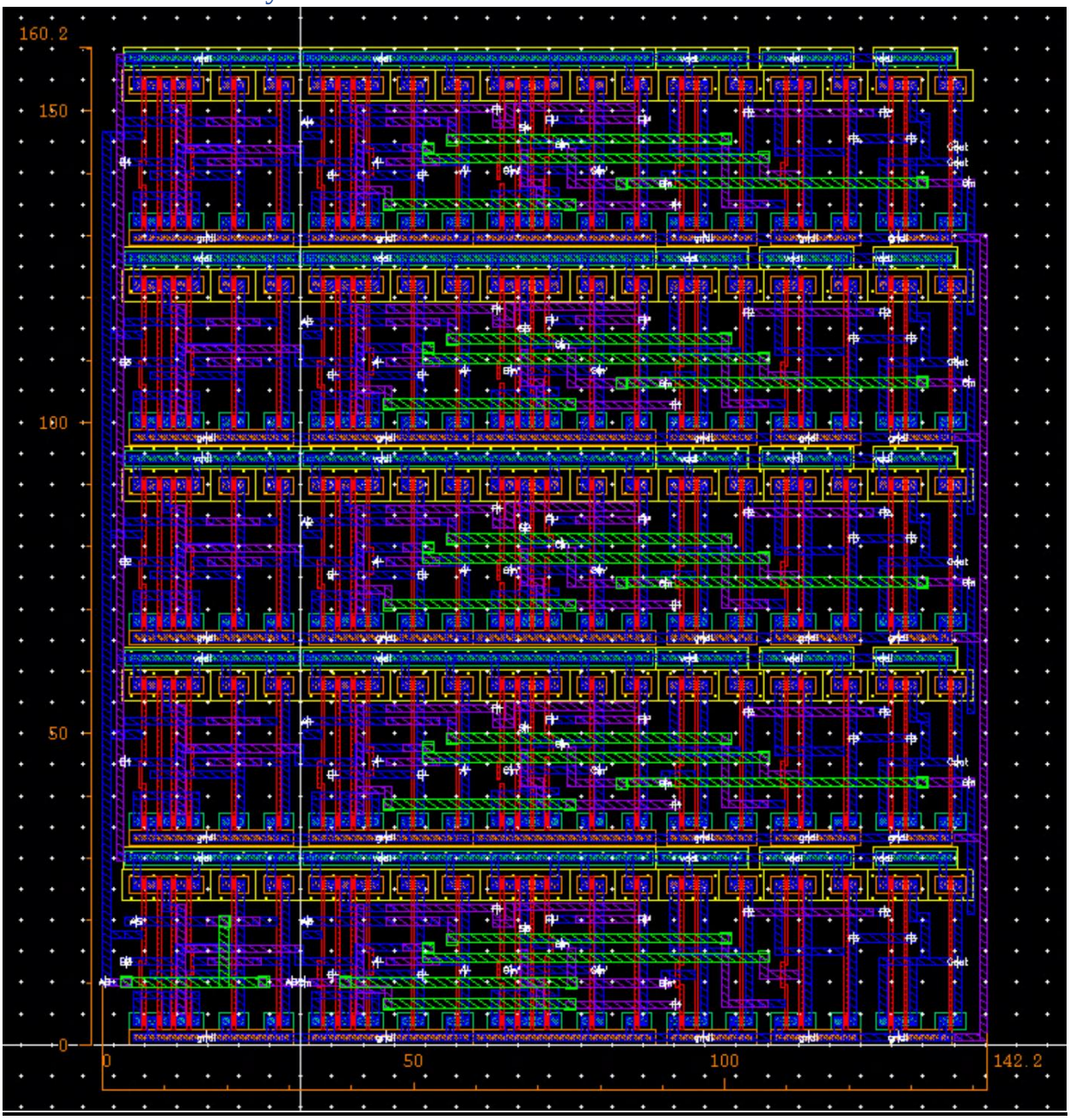
No microscope views yet.

```
@(#)$CDS: LVS version 6.1.7-64b 08/21/2018 19:40 (sjfhw315) $

Command line: /apps/vlsi_2018/cadence/IC617/tools.lnx86/dfII/bin/64bit/LVS -dir /home/ead/napen/cadence/LVS -l -s -t /home/ead/napen/cadence/LVS/layout /home/ead/napen/cadence/LVS/schematic
Like matching is enabled.
Net swapping is enabled.
Using terminal names as correspondence points.
Compiling Diva LVS rules...

    Net-list summary for /home/ead/napen/cadence/LVS/layout/netlist
       count
        1486            nets
        26              terminals
        1505            pmos
        1505            nmos

    Net-list summary for /home/ead/napen/cadence/LVS/schematic/netlist
       count
        1486            nets
        28              terminals
        1505            pmos
        1505            nmos


    Terminal correspondence points
    N1464     N48       A0
    N1463     N49       A1
    N1462     N52       A2
    N1461     N53       A3
    N1460     N55       A4
    N1480     N9        A5
    N1485     N42       B0
    N1484     N43       B1
    N1483     N44       B2
    N1482     N45       B3
    N1481     N47       B4
    N1468     N36       CLK
    N1478     N30       F0
    N1477     N28       F1
    N1476     N26       F2
    N1475     N23       F3
    N1474     N22       F4
    N1473     N18       F5
    N1472     N16       F6
    N1471     N15       F7
    N1470     N14       F8
    N1469     N13       F9
    N1479     N35       LE
    N1465     N34       SH_LD
    N1467     N24       SO0
    N1466     N31       SO1

Devices in the netlist but not in the rules:
        pcapacitor
Devices in the rules but not in the netlist:
        cap nfet pfet nmos4 pmos4
```

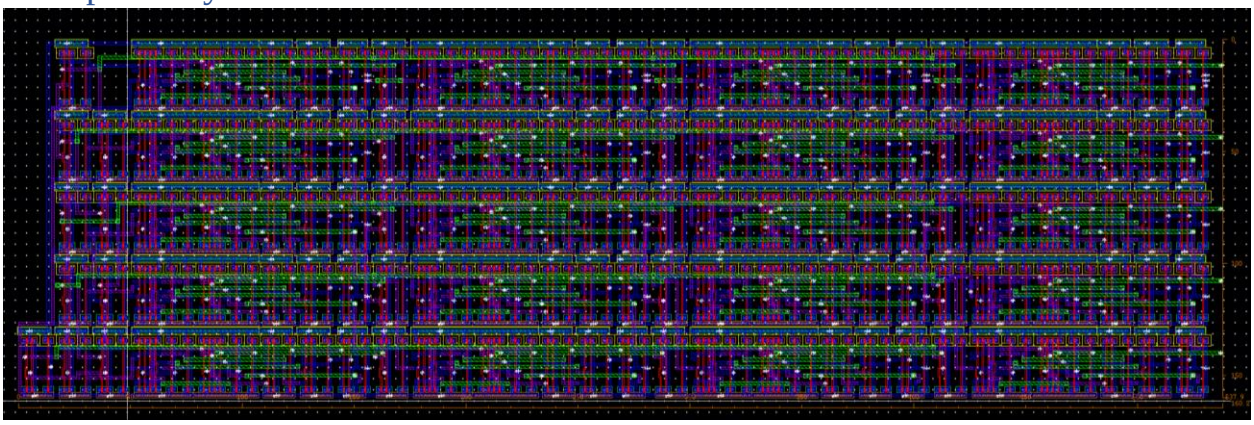
```
The net-lists match.

                            layout  schematic
                               instances
        un-matched             0       0
        rewired                0       0
        size errors            0       0
        pruned                 0       0
        active                 3010    3010
        total                  3010    3010

                                 nets
        un-matched             0       0
        merged                 0       0
        pruned                 0       0
        active                 1486    1486
        total                  1486    1486

                               terminals
        un-matched             0       0
        matched but
        different type         0       0
        total                  26      28


Probe files from /home/ead/napen/cadence/LVS/schematic

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:


Probe files from /home/ead/napen/cadence/LVS/layout

devbad.out:

netbad.out:

mergenet.out:

termbad.out:

prunenet.out:

prunedev.out:

audit.out:
```

The team was able to obtain a LVS match for the entire system as shown above. For each individual module layout shown below, LVS matches were also obtained before combining them together into the overall system.

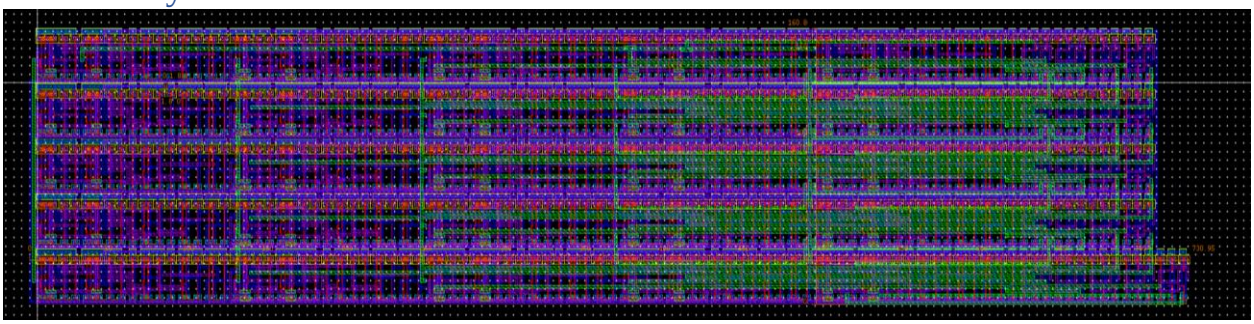
Adder/Subtractor Layout



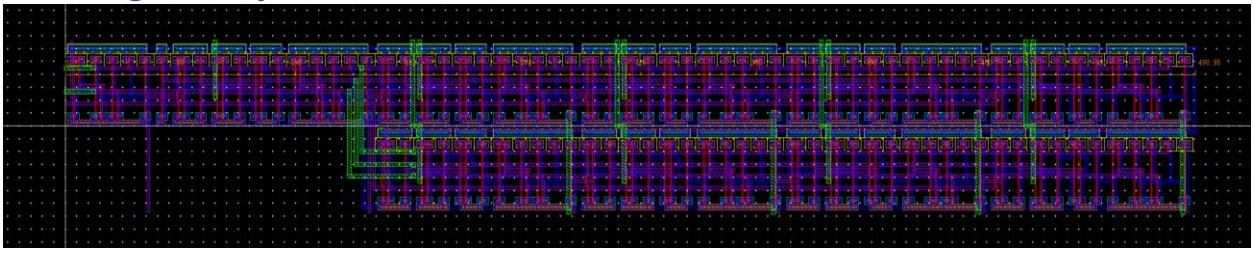
Multiplier Layout



Divider Layout



Shift Register Layout

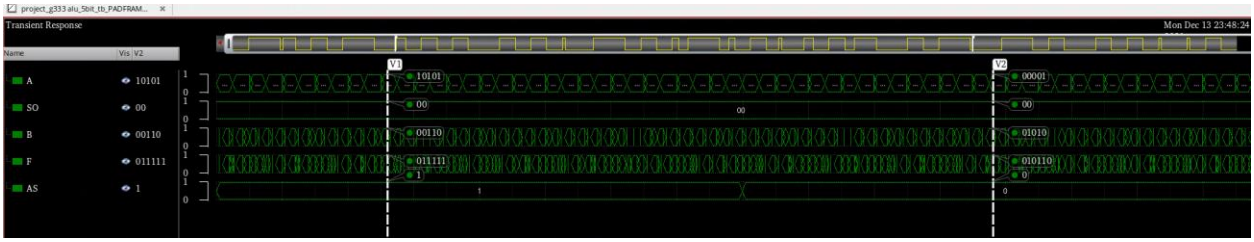


VI. Simulation vs Measured Results

Pad Frame Simulations

A test bench is set up using the symbol from the pad frame and the control pins are set up (00, 01, 10, 11) to simulate each operation. The simulation results are shown below with markers to show the values at certain points to prove the functionality of that operation. Although only a couple cases are shown in each screenshot, the team ran the simulation for all input combinations to confirm the functionality of each operation.

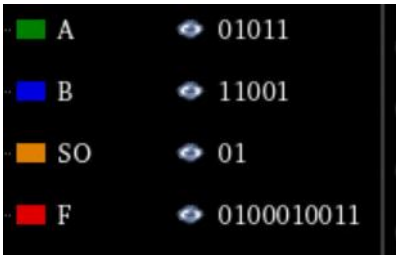
Add/Subtract: SO=00



10101 - 00110 = 01111

00001 + 01010 = 01011

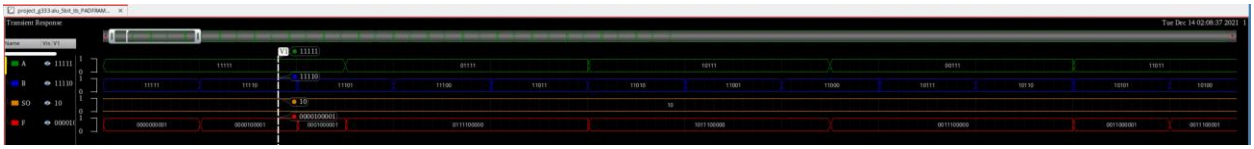
Multiplication: SO=01



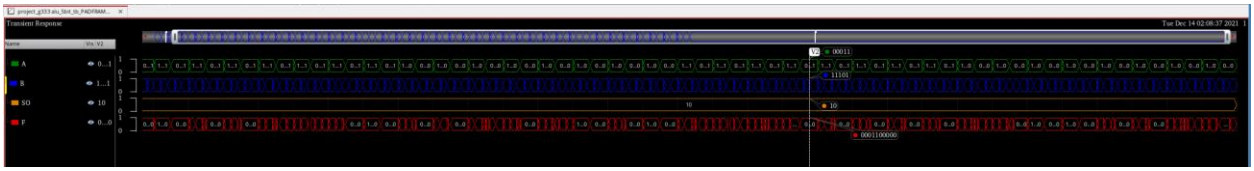
01011 x 11001 = 0100010011

Division: SO=10

$A_4A_3A_2A_1A_0 / B_4B_3B_2B_1B_0 = R_4R_3R_2R_1R_0Q_4Q_3Q_2Q_1Q_0$



11111 / 11110 = 0000100001



00011 / 11101 = 0001100000

Shift: SO=11

Shifts from 1001100111 -> 0011100000 (5 clock cycles)

VII. Failure Cases

One of the failure case the team ran into was in the shift register. When having the SH_LD input switching after one clock cycle the shift register adds a 1 to the end of the output instead of a 0. This issue was remedied by allowing the SH-LD to switch after two clock periods so that the correct output of a 0 being added to the end was achieved. This issue may also be solved with a different clock cycle and may no longer have to be an issue.

VIII. Problems/Debugging

Throughout this project the team encountered numerous problems whether it was getting the layouts to function or the schematics to work. The first major problem the team ran into was the first divider schematic that was created. The first divider schematic was chosen because of its low gate and transistor count. It helped the team reach the minimum transistor count of 3000 without going over the 3500-transistor limit. Once the schematic was created and simulated using the gate symbols the team realized that it did not function properly so we had to look for a new one. The team did some research and found a new divider design that worked but at the cost of using more transistors. Implementing the new divider put the transistor count to over 3500 so the team had to find somewhere to cut down. This occurred with the D-flip flop. Instead of using a traditional gate D-flip flop that is about 40 transistors, the team opted to use one that implemented dynamic logic and is only 12 transistors each.

Since the ALU used 10 flip flops in the shift register design, this brought down the transistor count significantly so that it is in the acceptable range.

Once the issues with the schematic levels were sorted out the team had to figure out the layouts. The layouts went smoothly except for some issues that occurred when trying to change instantiated gates into a design. This issue was resolved quickly by just making copies of the gate layouts and instantiating the copies. The most common problem with layouts is getting a LVS check. Thanks to the helpful tools in Cadence, the team was able to see where the net lists did not match and zoom in on the nets in the extracted view and figure out the issue. After numerous hours of debugging each of the layouts for the adder/subtractor, multiplier, divider, shift register, and the overall ALU, the team was able to have a complete layout. This layout was then put into the PAD frame and had all the inputs and outputs connected to the pin; the only error occurred when the outputs were not attached to the proper pin. This issue was quickly noticed and resolved, and the team were able to get a fully working chip that can now be fabricated.

IX. Conclusion

In conclusion, the team was very successful in designing, creating, and simulating a 5-bit ALU. Throughout the fall semester the students learned to use one of the most useful tools in the field of VLSI, Cadence. At the beginning of the class both Jaret and Nathan had never touched Cadence before and after completing the homework assignments, labs, and the final project we can confidently say that we know the basic functionality of the software, especially for digital design. The team chose the ALU with an adder/subtractor, multiplier, divider, and shift register because it combined the course material, we have learned in our previous classes such as ECE 2140, ECE 3130, and ECE 3135. ECE 4140/6240 allows the students to learn the final step of chip fabrication, which is the layout design, and now the students have all the basic tools knowledge to begin work in the field of chip design/VLSI. The design of the ALU was meticulously planned so that we could have the most efficient fitment in the PAD frame. Details such as the cell pitch of all the cells were equal to keep consistency, the multiplier and divider were designed to be wide and short because they are the biggest segments and then the adder/subtractor and shift register were fit around the multiplier and divider because they are smaller components. Throughout the project, both team members put in late nights and early mornings drawing and debugging the schematics and layouts of each major block until we got a final working ALU in the PAD frame. The team achieved the goal at the end of the project with not just a completed design that can be fabricated but a feeling of fulfillment and accomplishment moving forward into future classes and our electrical engineering career.