Design and Implementation of a 4-bit 4x4 Systolic Matrix Multiplier

1st Colin Barnum
dept. of Electrical and Computer Engineering
Virginia Tech
Blacksburg, United States
colinmbarnum@vt.edu

2nd Sean McTernan
dept. of Electrical and Computer Engineering
Virginia Tech
Blacksburg, United States
seanmct@vt.edu

3rd Marwan Abbas
dept. of Electrical and Computer Engineering
Virginia Tech
Blacksburg, United States
marwaneltoukhy@vt.edu

***Abstract*—In this paper, we design and implement a 4-bit 4x4 systolic array multiplier. We further describe the schematics and layouts of the individual components to create the full matrix multiplier. Our DPU design is compact, at only 274.48µm^2. The full systolic array layout takes only 4.531mm^2.**

***Index Terms*—systolic array, VLSI, matrix multiplier**

I. Introduction

A systolic array is a method of matrix multiplication using an array of individual data processing units (DPU). The top-left node in the DPU array receives two 4-bit operands. On the first clock cycle, the first DPU multiplies these two operands and adds the result to whatever was previously in its 12-bit register, which should be cleared at initialization. On the next clock cycle, the original operands pass on to adjacent DPUs while new data is fed into the first DPU. The same operations occur in each DPU for seven total clock cycles.

Data going into the systolic array has to be zero-padded in certain places. In particular, the data entering the lowest rows and rightmost columns must be padded with three zeros. In general, data in one next row or column is padded with an extra zero. Data in upper rows and and left columns must be followed by zeros after a certain number of clock cycles.

The DPU operates with the following steps where P is the product from the multiplier, C is the result, and A and B are operands:

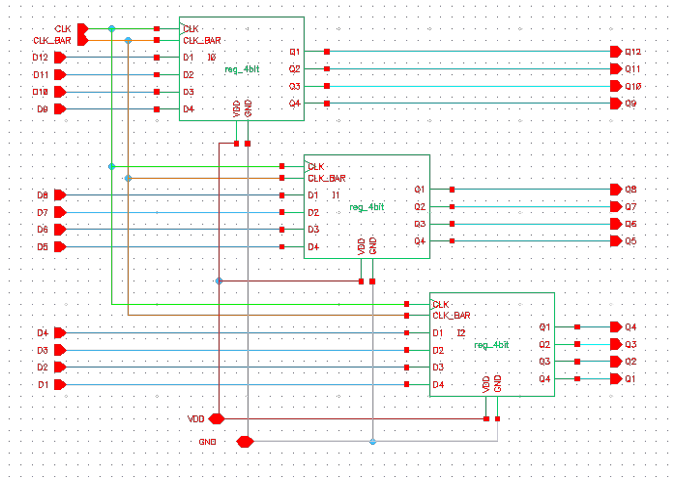
1) Calculate $P = A \times B$
2) Calculate $C = C + P$
3) Pass A and B to adjacent DPUs

II. Data Processing Unit (DPU)

A. Part Designs

The DPU design requires 4 components:

1) 4-bit register
2) 12-bit register
3) 12-bit buffer
4) 12-bit adder



(a) 12-bit Register Schematic

Fig. 1: 12-bit Register Layout

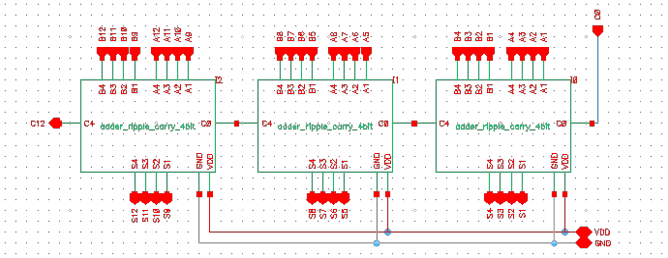
5) 4-bit multiplier

We tested each part thoroughly before moving on to the final design. For the sake of brevity, those simulations are not included in the report.

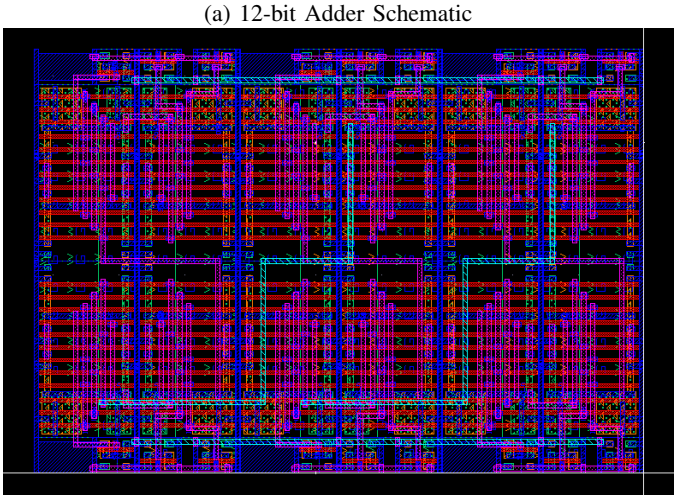
The parts are described in detail in the following sections.

a) Registers: The 4-bit and 12-bit registers are arrays of D-Flip Flops (DFF). The DFFs were borrowed from a previous project [2]. These were strung together in an alternating flipped pattern to connect GNDs and VDDs and save space. Schematic and layout for just the 12-bit register are shown in *Figure 1* since the 12-bit register includes 4-bit registers.

b) 12-bit adder: The 12-bit adder uses 3 4-bit adders from a previous project [3] connected together by their carry-outs and carry-ins. The 4-bit adder uses a series of mirror

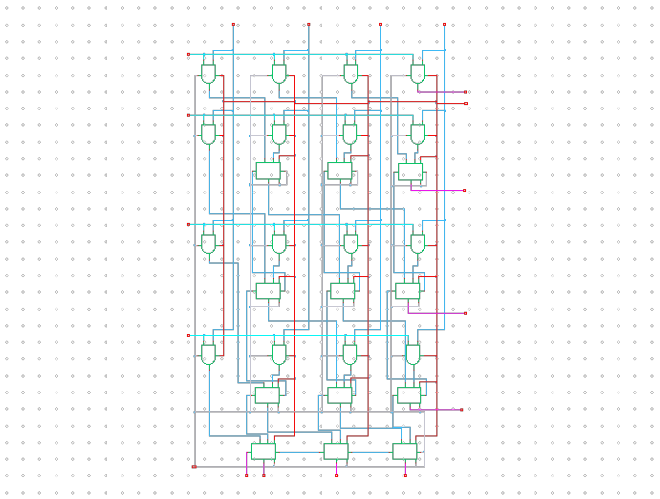


(a) 12-bit Adder Schematic

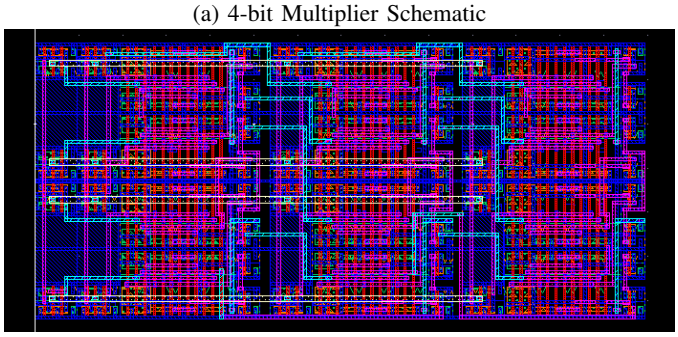


(b) 12-bit Adder Layout

Fig. 2: Adder Schematics & Layouts



(a) 4-bit Multiplier Schematic



(b) 4-bit Multiplier Layout

Fig. 3: Multiplier Schematics & Layouts

adders to increase speed. The adder is shown in *Figure 2*.

c) 4-bit multiplier: The 4-bit multiplier is an implementation of a Braun multiplier [1]. It re-uses the 4-bit adder described earlier. One change to the individual 4-bit adder component is the addition of two inverters to correct the inverted carry and sum outputs inherent to the mirror adder. Schematic and layout are shown in *Figure 3*.

d) 12-bit buffer: Part of the way through the project, we realized that the multiplier did not properly drive the 12-bit register in simulation. To solve this problem, we added a 12-bit buffer, composed of series double inverters, to boost the multiplier's signal. The 12-bit buffer schematic and layout are shown in *Figure 4*.

B. Test Bench & Simulation

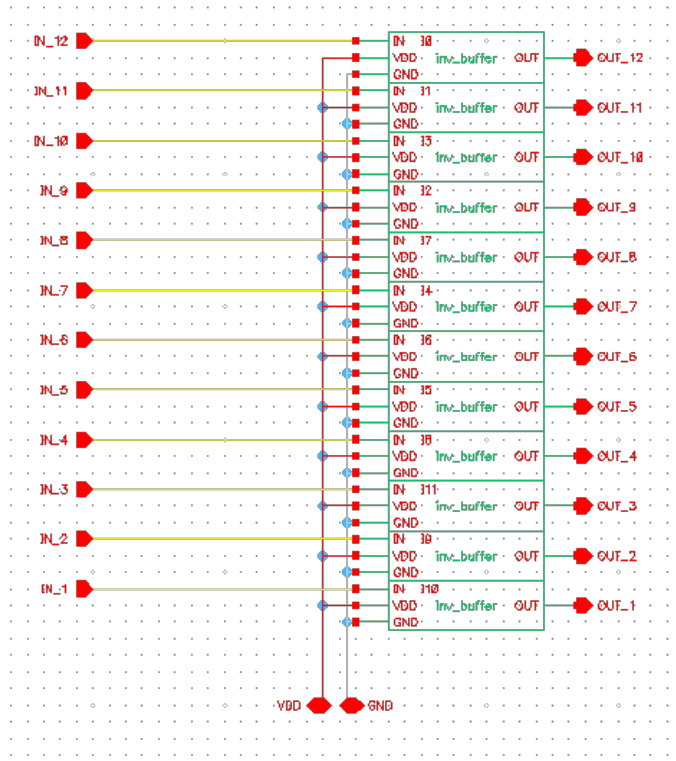
C. Layout & Simulation

The layout for the DPU with its bounding box measurements is shown in *Figure 5*. The layout area is 274.48µm^2. *Figure 4*.

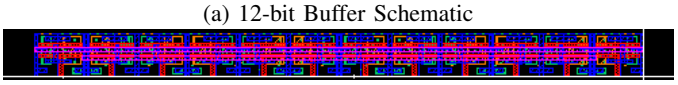
The layout passes DRC and LVS as shown in *Figure 6*.

We tested the DPU with the test bench shown in *Figure 7*.

The results are shown in *Figure 8*.



(a) 12-bit Buffer Schematic



(b) 12-bit Buffer Layout

Fig. 4: Buffer Schematics & Layouts

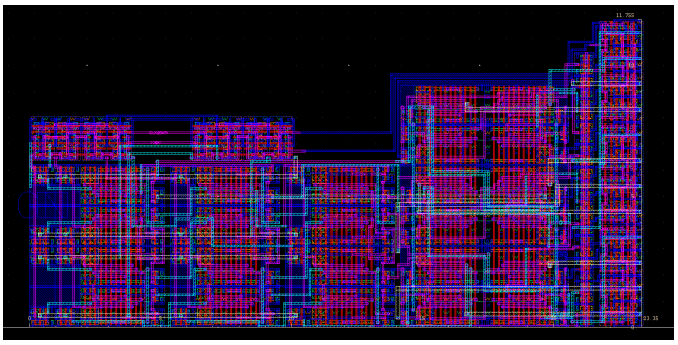
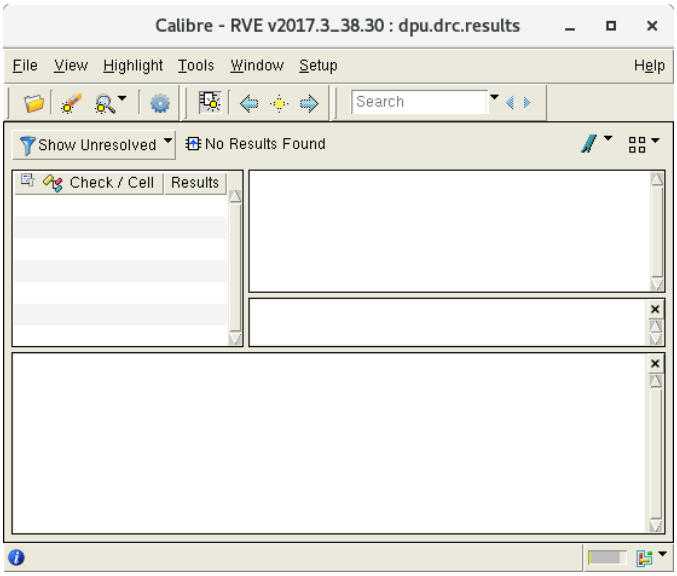
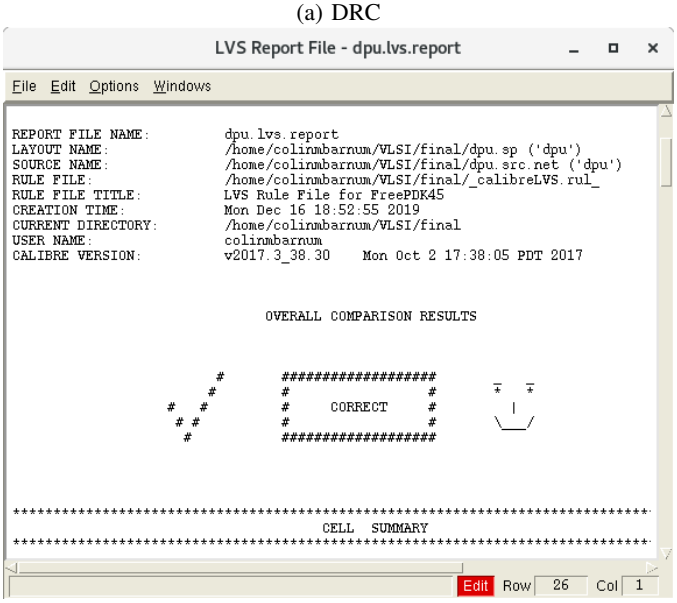


Fig. 5: DPU Layout



(a) DRC



(b) LVS

Fig. 6: DRC & LVS

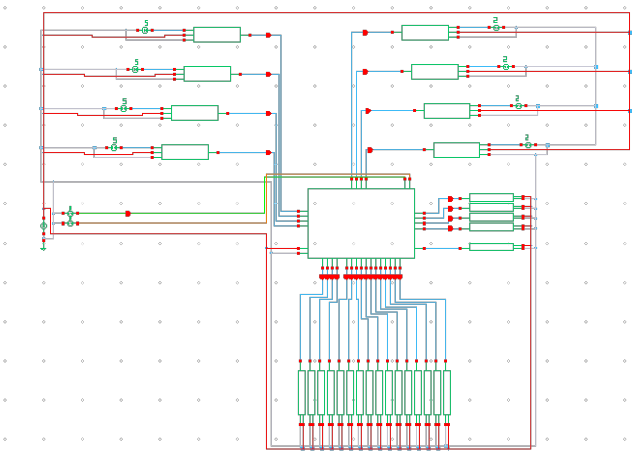
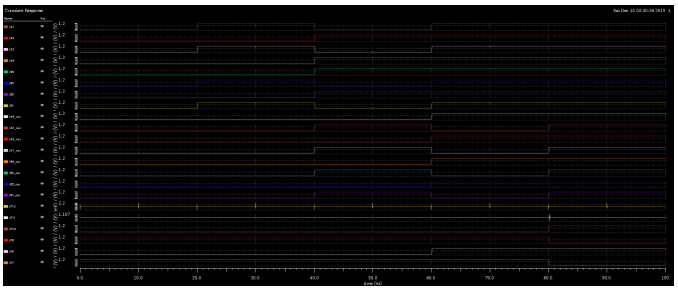
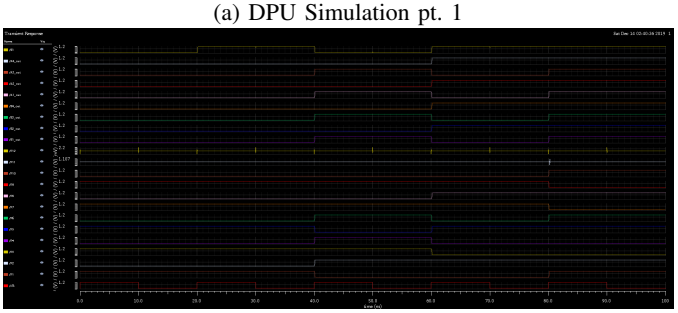


Fig. 7: DPU Test Bench



(a) DPU Simulation pt. 1



(b) DPU Simulation pt. 2

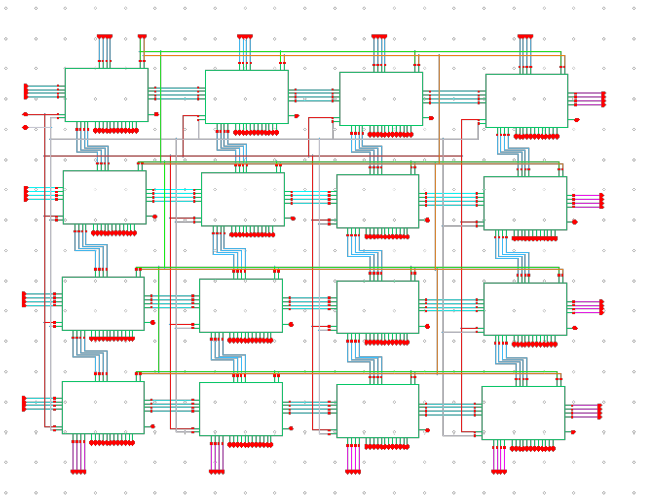
Fig. 8: DPU Simulations

III. Systolic Matrix Multiplier

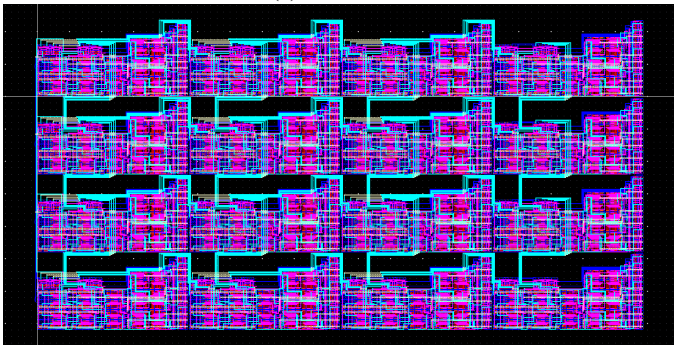
The systolic matrix multiplier is simply an array of 16 DPUs designed earlier. The schematic and layout are shown in *Figure 9*.

Because designing and refining the DPU took so long, we ran out of time to complete the systolic array. While the layout is mostly complete, it encounters some LVS warnings (*Figure 10*). However, we believe these are minor bugs in the layout that could be fixed with a few changes. For example, we did not add carry pins in the layout though they are present in the DPUs. Others could be caused by the few DRC errors we encountered.

As a result, we did not simulate the systolic array layout. However, we did create a test bench and simulate the pre-layout systolic array schematic. *Figure 11* shows this test bench. Simulations can be seen in *Figure 12*.



(a) Schematic



(b) Layout

Fig. 9: Systolic Array Schematic & Layout

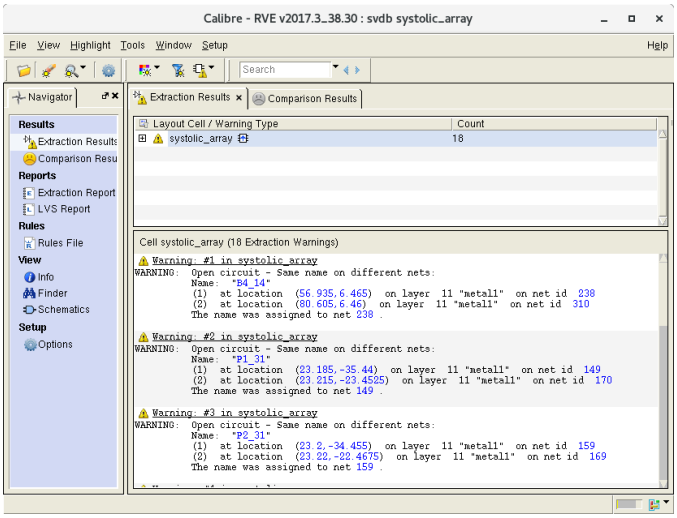


Fig. 10: Systolic Array Layout LVS

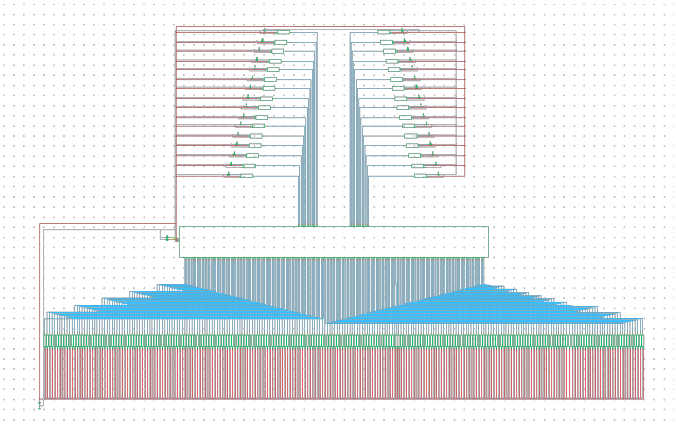
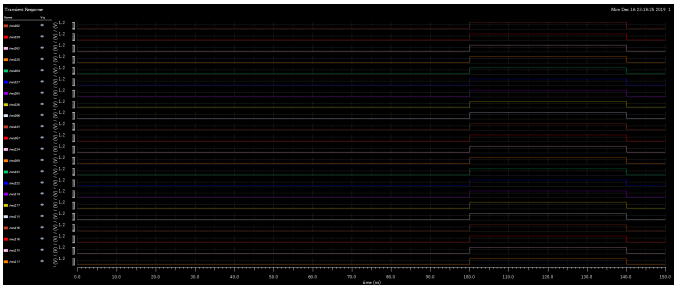
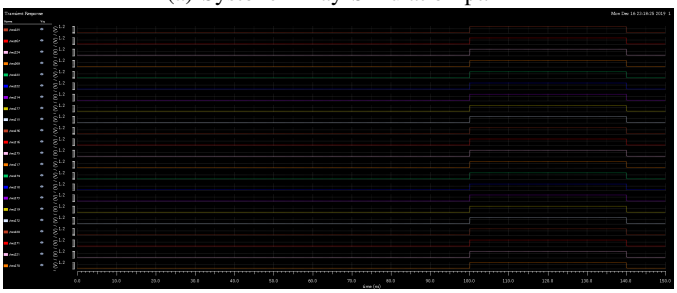


Fig. 11: Systolic Array Test Bench



(a) Systolic Array Simulation pt. 1



(b) Systolic Array Simulation pt. 2

Fig. 12: Systolic Array Simulation

ACKNOWLEDGMENT

Special thanks to Victor Gan for his selfless hard work in helping us with this project.

REFERENCES

[1] M. Thakur, J. Ashraf, "Design of Braun Multiplier with Kogge Stone Adder & Its Implementation on FPGA," International Journal of Scientific & Engineering Research, Volume 3, Issue 10, pp. 529–551, October 2012.

[2] C. Barnum, "ECE4540 Lab 4: 4-bit Ripple Carry Adder With Mirror Adder Cells," December 2019.

[3] C. Barnum, "ECE4540 Lab 3: Design and Layout of a Reduced Clock Load Positive Edge Triggered D-FlipFlop" December 2019.