# Power, Performance, and Area Analysis of Asynchronous Stochastic Neural Accelerator PASSOv1

*Steven Lu*

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-94
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-94.html

May 13, 2022

# Power, Performance, and Area Analysis of Asynchronous Stochastic Neural Accelerator PASSOv1

by Steven Lu

## Research Project

Submitted to the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, in partial satisfaction of the requirements for the degree of **Master of Science, Plan II**.

Approval for the Report and Comprehensive Examination:

**Committee:**

Sayeef Salahuddin

Professor Sayeef Salahuddin
Research Advisor

5.13.2022

(Date)

* * * * * * *

Professor Sophia Shao
Second Reader

5/11/2022

(Date)

# Power, Performance, and Area Analysis of Asynchronous Stochastic Neural Accelerator PASSOv1

Steven Lu

May 2022

## Abstract

Many NP-hard combinatorial optimization problems have practical applications in modern society, from vehicle routing to ASIC place-and-route for digital flows. However, traditional synchronous processor-based solutions struggle to keep up with the demands imposed by these problems, scaling exponentially with the input problem size. The Salahuddin group recently taped out the PASSOv1 processor, an asynchronous stochastic processor that demonstrates a significant power and performance improvement in solving a 100-node max-cut problem over other state-of-the-art systems. This report provides an analysis of the power, performance, and area of the processor's analog core to facilitate the redesign process for future iterations of the chip. In addition, the report explores a potential lower-power topology to be used as a replacement for one of the chip's subcircuits to reduce the power consumption.

## Acknowledgements

I would like to thank the entire PASSOv1 project team, including Saavan Patel, Philip Canoza, and Adhiraj Datar for their hard work and dedication to making the tapeout of the chip and this project a reality. Furthermore, I would like to thank Professor Sayeef Salahuddin for his amazing sponsorship and guidance for this project and my undergraduate research. Finally, I would also like to thank my family for supporting me throughout my education.

# Contents

**1 Introduction** ....................................................................................**4**

1.1 Background ........................................................................................4

1.2 Outline ...............................................................................................4

**2 PASSOv1 Processor** ..........................................................................**5**

2.1 Full Processor .....................................................................................5

2.2 Analog Core ........................................................................................5

2.3 Current Bias Generation .....................................................................6

2.4 Mixed Signal Neuron ..........................................................................6

2.4.1 Noise and Amplification Block .....................................................7

2.4.2 Sigmoid Generation and Output Digitization ...............................10

2.5 Test Cluster .......................................................................................11

**3 Area Analysis of PASSOv1** ...............................................................**11**

3.1 Area Breakdown of the Processor ....................................................11

3.2 Area Analysis Conclusions ...............................................................15

**4 Power Analysis of Integrated Neuron** ..............................................**16**

4.1 Power Analysis Setup .......................................................................16

4.2 Power Breakdown of an Isolated Integrated Neuron ........................16

4.3 Power Breakdown of an Integrated Neuron with Maximal Switching ........18

4.4 Power Analysis Conclusions .............................................................20

**5 Performance Analysis of Integrated Neuron** ....................................**21**

5.1 Autocorrelation Performance Analysis .............................................21

5.2 Delay Analysis ..................................................................................24

**6 Design Exploration** ...........................................................................**26**

6.1 Modified Flipped Voltage Follower (Prelayout) ...............................27

6.2 Prelayout Modified FVF Conclusion ................................................35

6.3 Modified Flipped Voltage Follower (Postlayout) .............................36

6.4 Postlayout Modified FVF Conclusion ...............................................44

**7 Conclusion and Future Work** ............................................................**44**

# 1 Introduction

## 1.1 Background

As the complexity of difficult computing problems continues to increase, many conventional solutions are struggling to keep up with the data processing demands imposed by such problems. Many problems that have useful applications to modern society, such as vehicle routing, are classified as NP-hard, meaning that they grow exponentially in computational difficulty with the size of its input. Modern computing heuristics commonly apply centralized clock-driven approaches that evaluate instructions sourced from memory sequentially to solve these NP-hard problems, but these approaches don't effectively take advantage of problem level parallelism, scale poorly with input size, and require algorithms designed to be split across multiple von-Neumann cores [1]. Recently, the Salahuddin group has experimented with decentralized computing systems that eliminate this bottleneck, the latest of which was the "Parallel Asynchronous Stochastic Sampling Optimizer" (PASSO), taped out in Global Foundries 14nm in April 2021.

The PASSO processor is a stochastic, asynchronous processor based on a system called the Ising model. The Ising model uses a fabric of binary spins with weights for interactions between neighboring spins and optimizes problems by finding the minimum energy state for the system. Its fine-grained parallelism leads to it being a good candidate for next generation large scale systems, including traveling salesman, max-cut, and integer factorization [2]. There are many past realizations of Ising machines, such as the oscillator-based machine presented by Wang, et al. that was demonstrated with max-cut and half-adder solutions [3]. The PASSO chip is fundamentally different from these previous implementations, utilizing mixed signal neurons with intrinsic noise integration and stochastic parallel update capacity to realize massively parallel asynchronous neuron state updates [1].

To scale up the problem size that the PASSO processor can solve, the number of neurons needs to be increased, which increases area and power consumption, and their performance needs to be improved. To facilitate the redesign of the subcircuits in PASSOv1, the first iteration of the processor, to optimize for these variables in future iterations, this report details the power, performance, and area analysis of PASSOv1's analog core, and explores a new topology for one of its subcircuits to reduce its power consumption.

## 1.2 Outline

First, this report will describe the design and functionality of the various components making up the PASSOv1 processor and break down the area taken by each component to determine the primary target for area optimizations in the future. Because the analog core's power consumption and performance depend on the individual neurons, this report will focus on the power consumption of an individual neuron and its subcomponents, and analyze an individual neuron's performance, consisting of the propagation delay of the neuron and the autocorrelation function. Lastly, the report explores using a new topology, the "flipped voltage follower" [4], as a replacement for one of the neuron's subcircuits to reduce its power consumption.

# 2 PASSOv1 Processor

## 2.1 Full Processor

The PASSOv1 processor consists of a few main blocks: the main analog core, consisting of the main 16x16 fabric of 256 neurons, a small cluster of 4 neurons for testing, SRAM, circuitry for streaming out outputs serially, and I/O circuitry.

Before using the processor to solve a particular trained problem of interest, the processor first needs to be properly configured. The configuration bits must be shifted into the configuration shift registers, which are connected to form a long chain for the entire processor. There are 74 configuration bits for each neuron, 7 bits for each of the 32 trimmable current biases used for biasing the neurons, and 3 bits that encode the sampling frequency and number of neurons sampled for the streamout of the processor, totaling to 19171 configuration bits [1].

To move the outputs of the neurons in the analog core off-chip for data processing and analysis, the asynchronous neurons must be sampled with a sampling clock. The nominal sampling frequency is 300 MHz, but only 16 neurons can be sampled at once at this frequency due to I/O limitations. The 3 bits used to configure the sampling of the neurons defines presets of sampling frequency and the number of neurons sampled, ranging from 16 neurons at 300MHz to all 256 neurons at 18.75 MHz, maintaining a constant throughput of 4.8 Gsamples/sec [1].

Since the I/O circuitry limits the speed of the off-chip data transfer, the sampled neuron outputs are first written in burst mode into an SRAM buffer at the sampling clock frequency (maintaining the fixed 4.8 Gsamples/sec throughput) [1]. The SRAM buffer is then read out at a slower frequency to meet the I/O speed specifications (20MHz I/O clock) and serialized. In the case of a 300MHz sampling frequency producing 16-bit samples, the SRAM is read at a frequency of 20/16 = 1.25 MHz [1].

## 2.2 Analog Core

The analog core of PASSOv1 is made up of a 16x16 fabric of 256 analog neurons, their "synapses" (circuitry used to connect the neuron to other neurons), the current bias circuitry for the neurons, and the shift registers used to configure the neurons and the biasing circuits, all of which share a VDD of 0.8V. The analog neuron, its synapse, and its configuration shift registers are bundled into what this report will call an "integrated neuron". Each integrated neuron is connected to its immediate neighbors in a "king's move" pattern via its synapse.

### 2.3 Current Bias Generation

The schematic of the constant-$g_m$ current bias circuitry is shown in Figure 1. A 7-bit digitally trimmed resistor is used to configure the current bias with 8 settings in a one-hot fashion, with the highest current setting of roughly 10 uA corresponding to setting only b6 to 1 and the lowest current setting of roughly 2 uA corresponding to all bits being set to 0. The bias is mirrored to 16 outputs, one for each neuron the bias generator is biasing. As each neuron requires two of these biases, there are 32 total current bias generators in the analog core.

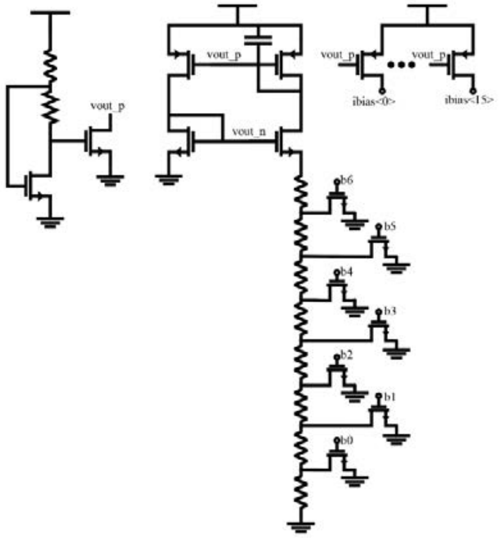

Figure 1: Schematic of current bias circuitry

### 2.4 Mixed Signal Neuron

The integrated neuron consists of 2 main blocks (excluding the configuration shift registers): the analog neuron and its synapse, or connection circuit. As depicted in Figure 2, it implements the function: $P(v_i = 1) = \sigma(W^T h + b_v)$; that is, the probability of the integrated neuron outputting a 1 is equal to the sigmoid function evaluated at the voltage calculated from the configured weights, configured bias, and inputs from the neighboring integrated neurons.

$$P(v_i = 1) = \sigma(W^T h + b_v)$$

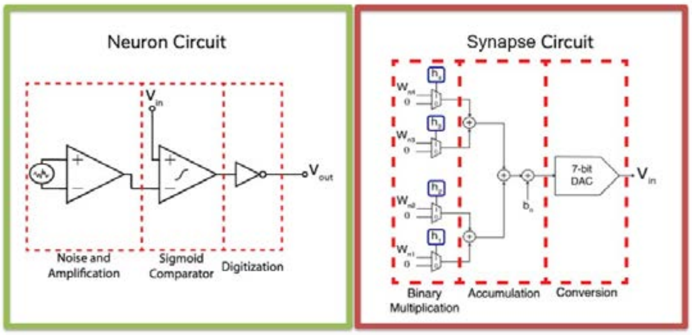

Figure 2: Block diagram of integrated neuron showing division of operation between synapse and neuron

The synapse is made of two subcircuits: the digitally synthesized “vecmul” (a multiply-accumulate block implementing the binary multiplication of the weights with the neighboring integrated neuron inputs and the accumulation with the bias), and the 7-bit DAC. To minimize power consumption, the DAC was designed to be a capacitive DAC (capdac), and to reduce the area, the C-2C topology was selected. The capdac schematic is shown in Figure 3.

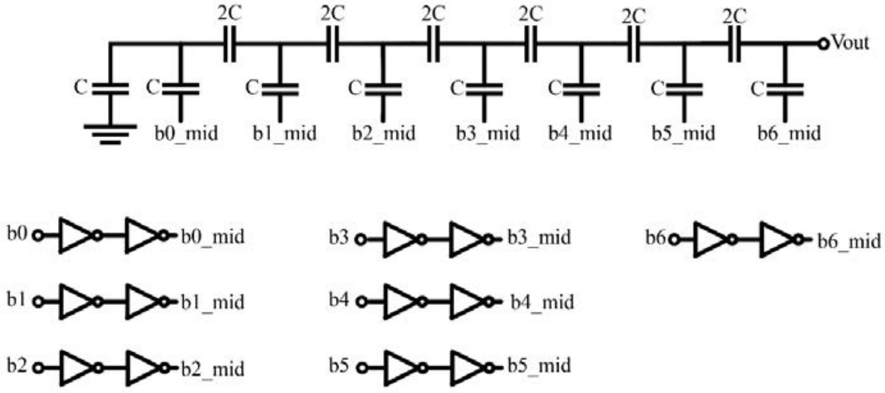

Figure 3: Schematic of the C-2C Capdac

The analog neuron consists of 3 main blocks: noise and amplification, sigmoid generation, and output digitization. It takes in an analog input voltage from its synapse and outputs a binary voltage signal. The neuron was designed to use a voltage input to enable the use of a capdac in the synapse and designed to output a voltage for easy integration with the vecmul.

### 2.4.1 Noise and amplification block

The noise and amplification block provides the stochastic nature of the neuron and is made of 2 subcircuits: a noise source, and a noise amplifier. The first subcircuit, the noise source, is depicted in Figure 4. The noise source is single-ended due to observations that when using a differential noise source, the variation was so high that it saturated the noise amplifier. Grounding the n-well of the PMOS transistor as shown in the schematic was done to increase the noise significantly. So long as the substrate voltage is stable, there should not be latchup issues. The resulting design is characterized using the noise amplifier as its load and tabulated in Table 1.

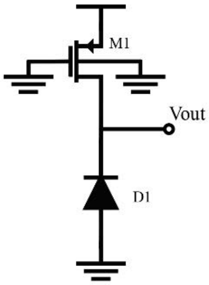

Figure 4: Schematic of noise source for analog neuron

| Parameter | Value |
|---|---|
| **Common Mode Vout** | 643 mV |
| **Peak to Peak Vout** | 8.9 mV |
| **RMS Vout** | 1.32 mV |

Table 1: Characteristics of noise source for analog neuron

The second subcircuit, the noise amplifier, is a three-stage single ended amplifier with internal resistive feedback. The design goal was to have a relatively high speed amplifier with an intended closed loop gain of roughly 200. The first stage is a noise buffer, which takes the output of the noise source as its input. Due to the output resistance of the buffer playing a role in the closed loop resistive feedback of the last two stages of the amplifier, minimizing the $R_{out}$ was prioritized in the design. The design also was required to maintain an output common mode voltage of around 400mV (half rail). The "super source follower" topology [5] was chosen for this reason, shown in Figure 5. This stage of the amplifier shares the same current bias circuit as the next two stages, though the mirrored currents are not the same between the noise buffer and next two stages. The design is characterized at the lowest current bias setting and tabulated in Table 2.

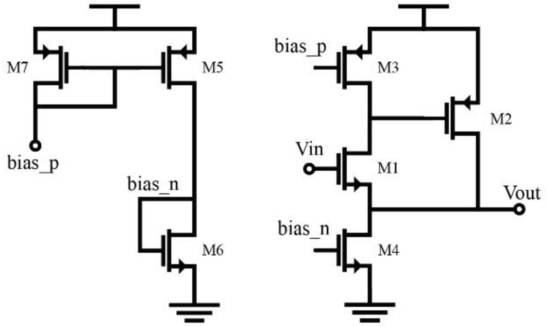

Figure 5: Schematic of noise buffer for noise amplifier

| Parameter | Value |
|---|---|
| **Input Capacitance** | 2.8578 fF |
| **RMS Vout** | 1.996 mV |
| **Common Mode Vout** | 378 mV |
| **Low Frequency $R_{out}$** | 154.5 Ω |
| **Low Frequency Gain** | -0.1639 dB |

Table 2: Characteristics of noise buffer for analog neuron

The remainder of the amplifier is what this report will call the noise OTA, a two-stage p-input OTA in resistive feedback, with the feedback resistor R2 and the combination of the output resistance of the buffer and input butterfly switch forming the feedback. The schematic of the noise OTA is shown in Figure 6a, and the schematic of the p-input OTA is shown in Figure 6b. The noise amplifier also has a reset-triggered auto-zeroing circuit used to eliminate the effects of the input offset voltage resulting from PVT variation. The overall noise amplifier is characterized at the lowest current bias setting and tabulated in Table 3.

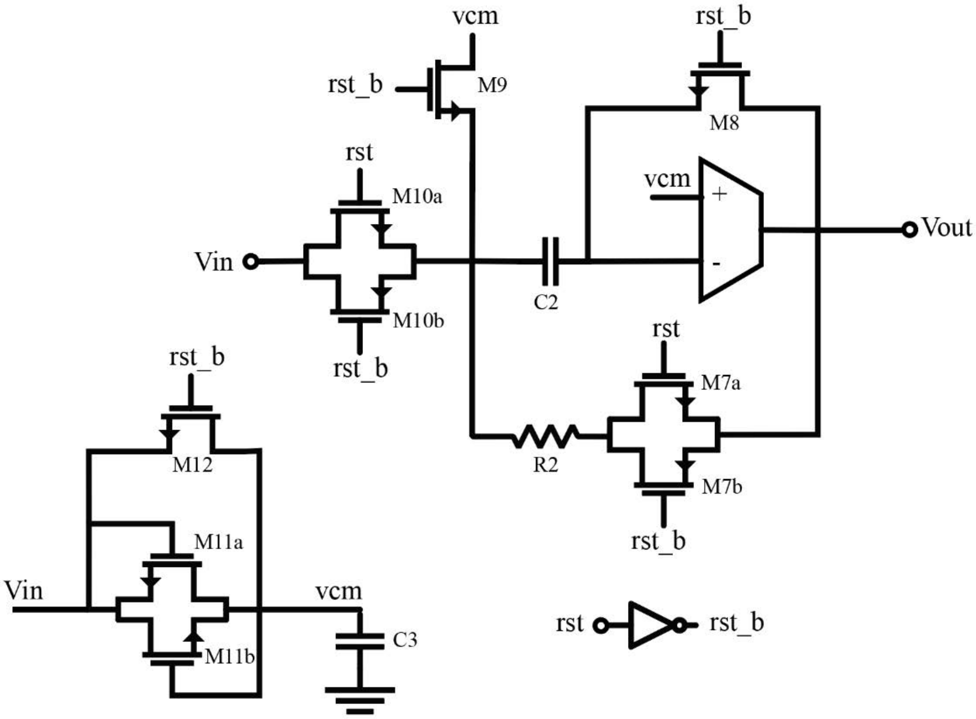

Figure 6a: Schematic of noise OTA for analog neuron

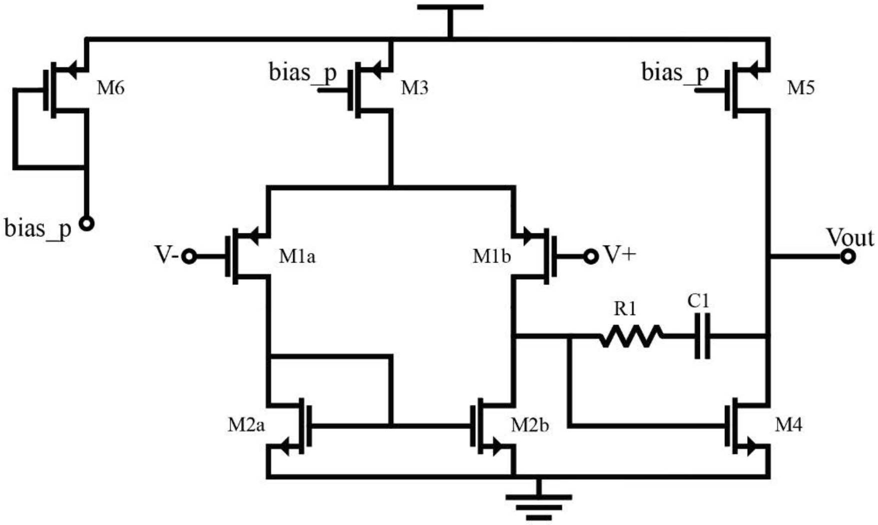

Figure 6b: Schematic of p-input OTA used for noise OTA

| Parameter | Value |
|---|---|
| **Low Frequency Gain** | 251.48 (48.01 dB) |
| **Corner Frequency** | 13.95 MHz |
| **Output Common Mode** | 416.443 mV |

Table 3: Characteristics of noise OTA

### 2.4.2 Sigmoid Generation and Output Digitization

The second block of the analog neuron is the sigmoid generation circuit, whose schematic is shown in Figure 7. It consists of 3 stages: a modified CMOS Gilbert cell, a circuit to convert from differential to single ended current, and a current comparator. It takes in the output of the noise amplifier and the output voltage of the synapse's capdac as input to generate an output whose probability of equaling a 1 (with VDD = 0.8V) is specified by the sigmoid function evaluated at the capdac output voltage. The bias for this block comes from the second bias generation circuit connected to the neuron, so it is separated from the bias of the noise amplifier block. The sigmoid function generated by this block is shown in Figure 8.

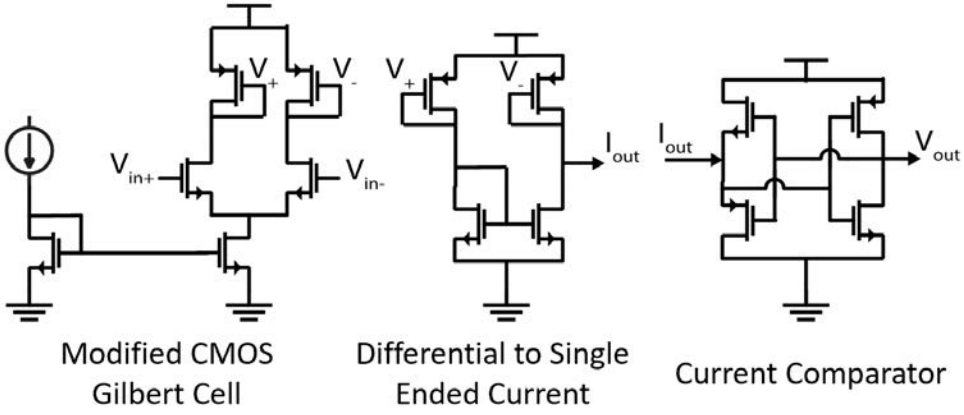

Figure 7: Schematic of sigmoid generator for analog neuron

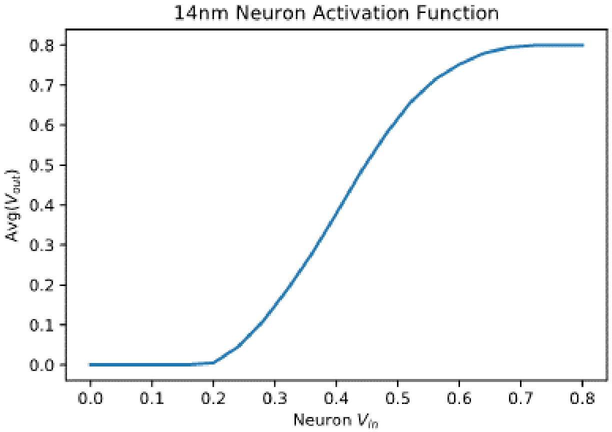

Figure 8: Function generated by sigmoid generator

The output digitization is handled by a chain of three CMOS inverters. Its purpose is to ensure the output is a clean binary signal.

### 2.5 Test Cluster

The test cluster consists of 4 integrated neurons separate from the analog core whose outputs can be directly pulled out via the chip's GPIO pins. Each integrated neuron is connected to the other 3 via its synapse. The test cluster has its own current bias generators used to bias the integrated neurons and configuration shift register chain used to program the weights, biases, and clamps of these integrated neurons, separate from those in the analog core. Its purpose is to enable probing of the integrated neurons' outputs directly, without them having passed through the sampling and streamout circuitry, for verifying the functionality of the integrated neurons.

## 3 Area Analysis of PASSOv1

### 3.1 Area Breakdown of the Processor

The PASSOv1 processor has a total die area of 2mm x 2mm. Table 4 shows the breakdown of the total area among the main blocks of the processor, and Figure 9 shows the breakdown of the layout of the processor.

| Block | Area (mm$^2$) | % of Total Area |
|---|---|---|
| **Analog Core** | 1.136 | 28.4% |
| **Test Cluster** | 0.0108 | 0.27% |
| **SRAM** | 0.0613 | 1.53% |
| **Streamout** | 0.0192 | 0.48% |
| **I/O** | 0.479 | 11.98% |
| **Routing/Fill/Empty Space** | 2.2937 | 57.34% |
| **Total** | 4.0 | 100% |

Table 4: Breakdown of die area by component

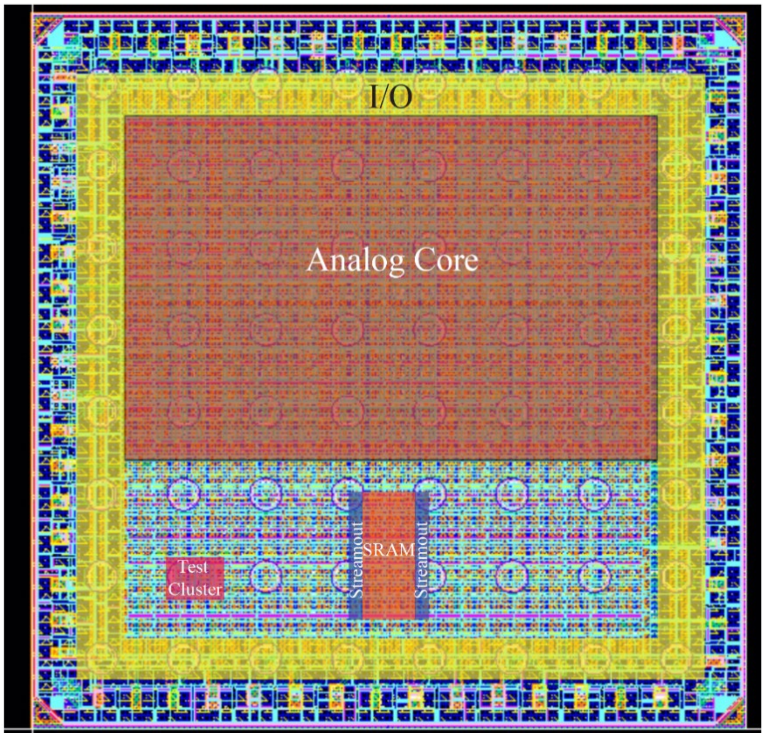

Figure 9: Breakdown of processor layout

Table 5 breaks down the area of the analog core by component.

| Block | Area (um$^2$) | % of Analog Core Area |
|---|---|---|
| **Integrated Neurons (x256)** | 540561.92 | 47.58% |
| **Current Bias (x32)** | 11621.12 | 1.02% |
| **Routing/Fill/Empty Space** | 583816.96 | 51.39% |
| **Total** | 1136000 | 100% |

Table 5: Breakdown of analog core area by component

Table 6 breaks down the area taken by the components of the current bias circuitry, and Figure 10 shows the breakdown of the layout by component.

| Block | Area ($um^2$) | % of Current Bias Circuitry Area |
|---|---|---|
| **Constant- $g_m$ + outputs** | 114.70 | 31.58% |
| **Starter Circuit** | 47.286 | 13.02% |
| **Digital Trim** | 103.33 | 28.45% |
| **Routing/Fill/Empty Space** | 97.87 | 26.95% |
| **Total** | 363.16 | 100% |

Table 6: Breakdown of current bias circuitry area by component

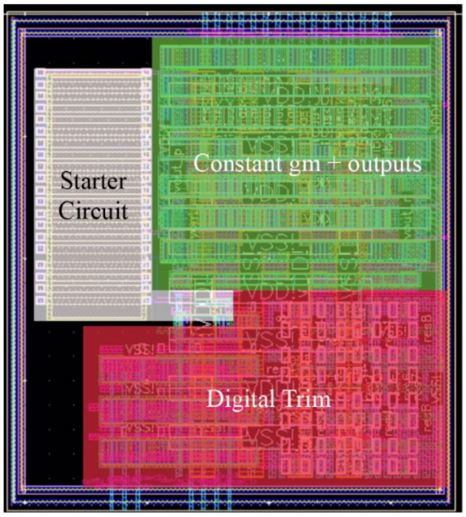

Figure 10: Breakdown of current bias generator layout

Table 7 breaks down the total integrated neuron area by component. Figure 11 breaks down the layout of the integrated neuron by component. Note that the breakdown of the area in the table and the layout shown below is for the integrated neurons used in the test cluster, not the analog core. The configuration circuitry for the analog core is implemented in a distributed manner, rather than being integrated with the integrated neuron like it is in the test cluster, making it difficult to obtain an estimate of the area it consumes per integrated neuron. So, for the purposes of this area analysis, the breakdown of the test cluster integrated neuron is reported instead.

| Component | Area (um$^2$) | % of Integrated Neuron Area |
|---|---|---|
| **Noise Generation** | 4.956 | 0.23% |
| **Noise Buffer** | 38.44 | 1.82% |
| **Noise OTA** | 159.375 | 7.55% |
| **Sigmoid Generation** | 31.096 | 1.47% |
| **Vecmul** | 345.50 | 16.36% |
| **Capdac** | 976.262 | 46.23% |
| **Configuration Circuitry** | 188.55 | 8.93% |
| **Output inverters** | 7.282 | 0.34% |
| **Routing/Fill/Empty Space** | 360.109 | 17.05% |
| **Total** | 2111.57 | 100% |

Table 7: Breakdown of test cluster integrated neuron area by component

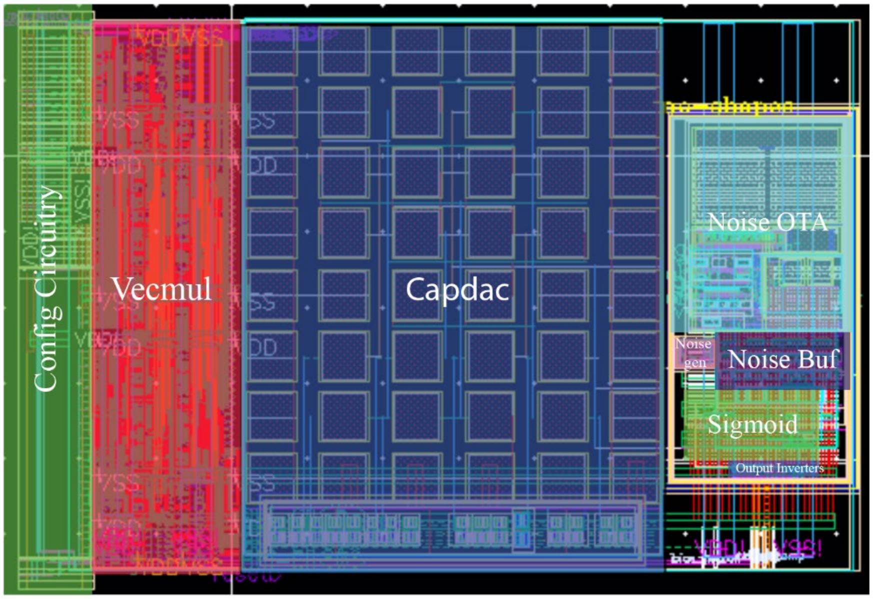

Figure 11: Breakdown of test cluster integrated neuron layout

Table 8 breaks down the area of the capdac, and the breakdown of the layout by component is shown in Figure 12. The dummy capacitors were used to form a ring around functional capacitors to reduce the layout effects.

| Block | Area (um$^2$) | % of Capdac Area |
|---|---|---|
| **Capacitors (functional) x24** | 261.76 | 26.8% |
| **Capacitors (dummy) x24** | 261.76 | 26.8% |
| **Driver Circuitry** | 103.33 | 9.72% |
| **Routing/Fill/Empty Space** | 94.903 | 36.68% |
| **Total** | 976.262 | 100% |

Table 8: Breakdown of capdac area by component

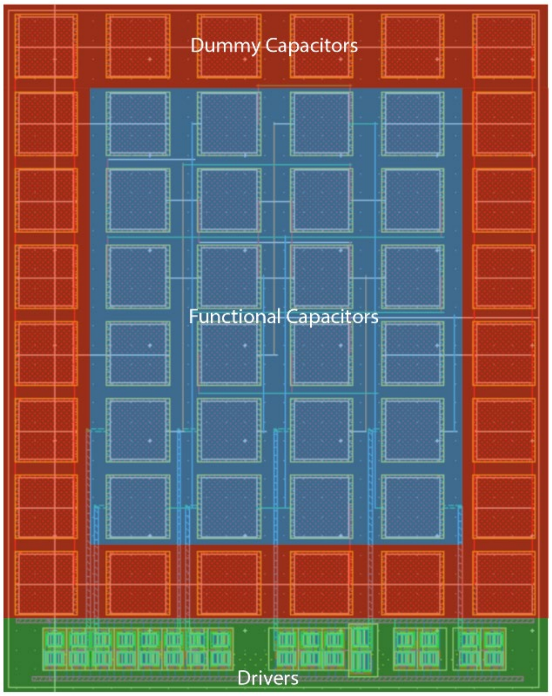

Figure 12: Breakdown of capdac layout

### 3.2 Area Analysis Conclusions

With 256 integrated neurons, roughly half of the processor die area is not utilized, meaning in terms of area, there is still room to increase the number of integrated neurons on the chip by roughly a factor of 2. If an increase beyond that is desired, however, the area taken by each integrated neuron must be reduced.

The capdac consumes the most area in the integrated neuron; almost half of the area taken by the integrated neuron is for the capdac. A quarter of that capdac area is non-functional and used for dummies. Thus, the capdac should be the first block to target for optimizing and redesigning with area in mind. Part of the reason the capdac is so large is due to the size of a single unit capacitor in the DAC; each one is roughly 11 um$^2$, sized to reduce the effect of layout parasitics on the DAC as the C-2C capdac topology is quite vulnerable to layout effects.

One potential improvement that could reduce the area is to switch to a topology that is more resilient to layout parasitics than the C-2C topology, like a mixed topology of a C-2C and binary capdac. Such a topology, despite using capacitors larger than 2C (e.g., 4C), may yield a net decrease in the area if the new unit base capacitance C is smaller than the base capacitance used for the original C-2C topology. Apart from topology optimizations, changing the dimensions of the capdac and some of the other components, like the config circuitry and the vecmul, to have the same vertical dimension as the analog neuron would net some area savings as well, as some empty space that was only populated with fill cells could be eliminated (e.g., the space in the top right and bottom right corners of the integrated neuron layout).

# 4 Power Analysis of Integrated Neuron

## 4.1 Power Analysis Setup

Two power analysis simulations were run to generate the results. First, the power was characterized for a single integrated neuron in isolation; there were no other integrated neurons connected to it, so the analog voltage input to the sigmoid generator was held at half rail. Any observed switching behavior would just be due to the noise generation and amplification within the neuron.

The second simulation used a 50% duty cycle, 330 MHz square wave input to model one of the neighboring integrated neurons. The weight for that “neuron” input was set to the maximum value, meaning that toggling that square wave causes the vecmul and capdac output to also toggle between 0 and their maximum values, resulting in maximal switching.

For both simulations, since the entire integrated neuron uses the same VDD, it is sufficient to compare and measure the currents in place of power. The simulations are run for the prelayout integrated neuron in the TT 25℃ corner, as the currents are not expected to change by a large margin from prelayout to postlayout, and the purpose of this analysis is to observe the breakdown of the power. The patterns observed with these results are expected to be present in the postlayout results as well, assuming the layout was done properly.

## 4.2 Power Breakdown of an Isolated Integrated Neuron

The breakdown of the current consumed by the integrated neuron at the lowest current bias setting is tabulated in Table 9, and the current consumed at the highest current bias setting is tabulated in Table 10.

| Block | Average Current (uA) | % of Total Average Current | Peak Current (uA) |
|---|---|---|---|
| **Noise Generation** | 6.652 | 5.45% | 6.8 |
| **Noise Buffer** | 19.04 | 15.59% | 22.232 |
| **Noise OTA** | 25.59 | 20.96% | 28.916 |
| **Sigmoid Generator** | 26.37 | 21.60% | 35.821 |
| **Output Stage** | 1.557 | 1.28% | 1303.03 |
| **Vecmul** | .3467 | 0.28% | 51.909 |
| **Capdac** | .2143 | 0.18% | 4.061 |
| **Noise Amplifier Current Bias** | 17.799 | 14.58% | 18.355 |
| **Sigmoid Generator Current Bias** | 17.819 | 14.59% | 18.17 |
| **Config** | 6.712 | 5.50% | 6.945 |
| **Total** | 122.1 | 100% | 8067.86 |

Table 9: Breakdown of isolated integrated neuron current consumption in the lowest current bias setting

| Block | Average Current (uA) | % of Total Average Current | Peak Current (uA) |
|---|---|---|---|
| **Noise Generation** | 6.652 | 3.041% | 6.8 |
| **Noise Buffer** | 72.59 | 33.181% | 77.948 |
| **Noise OTA** | 94.44 | 43.168% | 102.66 |
| **Sigmoid Generation** | 42.82 | 19.573% | 83.471 |
| **Output Stage** | 2.269 | 1.037% | 1422.2 |
| **Vecmul** | 4.515 | 0.28% | 73.739 |
| **Capdac** | .2143 | 0.18% | 4.061 |
| **Amplifier Current Bias** | 62.74 | 14.58% | 63.866 |
| **Sigmoid Generator Current Bias** | 62.84 | 14.59% | 63.773 |
| **Config** | 12.52 | 5.70% | 12.74 |
| **Total** | 361.6 | 100% | 8331.6 |

Table 10: Breakdown of isolated integrated neuron current consumption in the highest current bias setting (b6)

### 4.3 Power Breakdown of an Integrated Neuron with Maximal Switching

The breakdown of the power consumed by the integrated neuron at the lowest current bias setting is tabulated in Table 11, and the power consumed at the highest current bias setting is tabulated in Table 12.

| Block | Average Current (uA) | % of Total Average Current | Peak Current (uA) |
|---|---|---|---|
| **Noise Generation** | 6.313 | 3.68% | 6.314 |
| **Noise Buffer** | 18.61 | 9.73% | 19.404 |
| **Noise OTA** | 23.39 | 12.23% | 24.496 |
| **Sigmoid Generator** | 27.63 | 14.44% | 53.18 |
| **Output Stage** | 8.271 | 4.32% | 1392.7 |
| **Vecmul** | 28.65 | 14.98% | 1495.6 |
| **Capdac** | 35.7 | 18.66% | 3624.5 |
| **Noise Amplifier Current Bias** | 17.92 | 9.37% | 18.43 |
| **Sigmoid Generator Current Bias** | 18.121 | 9.47% | 18.832 |
| **Config** | 6.712 | 3.51% | 6.945 |
| **Total** | 191.317 | 100% | 5817.3 |

Table 11: Breakdown of integrated neuron current consumption in the lowest current bias setting with maximal switching

| Block | Average Current (uA) | % of Total Average Current | Peak Current (uA) |
|---|---|---|---|
| **Noise Generation** | 6.313 | 1.47% | 6.314 |
| **Noise Buffer** | 74.25 | 17.25% | 77.035 |
| **Noise OTA** | 94.39 | 21.93% | 102.021 |
| **Sigmoid Generation** | 70.06 | 16.28% | 168.80 |
| **Output Stage** | 8.828 | 2.05% | 1416.7 |
| **Vecmul** | 28.66 | 6.66% | 1497.2 |
| **Capdac** | 35.0 | 8.13% | 3813.6 |
| **Amplifier Current Bias** | 63.98 | 14.86% | 65.31 |
| **Sigmoid Generator Current Bias** | 65.12 | 15.13% | 67.38 |
| **Config** | 12.52 | 2.91% | 12.74 |
| **Total** | 430.461 | 100% | 6118.1 |

Table 12: Breakdown of integrated neuron current consumption in the highest current bias setting (b6) with maximal switching

### 4.4 Power Analysis Conclusions

As expected in the isolated simulation, the capdac and vecmul contribute very little to the total current, as they do not switch at all. Since only the noise buffer, noise OTA, and sigmoid generator are affected by the changes in the current bias setting, it follows that their contributions toward the total current increase from the lowest to the highest bias settings, as the other components consumed roughly the same current for both settings.

In the maximal switching simulation, the capdac and vecmul contribute significantly more to the total current compared to the isolated integrated neuron simulation, as expected. The frequent switching resulted in rapid charging and discharging the capacitors in the capdac and the effective switching capacitance of the vecmul, increasing their power consumption dramatically compared to when they did not switch at all. Due to its input voltage from the capdac constantly switching, the sigmoid generation block's current consumption also increased. This simulation is an upper bound on the power consumption, as for most normal operations, the vecmul and capdac will not be stepping from 0 to their maximum values at a 330 MHz frequency, but instead take smaller, incremental steps up and down. The 330

MHz frequency was derived from the autocorrelation time of the integrated neurons, a rough measure of how quickly an integrated neuron switches.

The source of the largest power consumption for the integrated neuron is the noise OTA, contributing almost half of the power consumption at the highest current bias setting in the isolated neuron simulation. The noise OTA thus presents itself as a high priority target for power optimization. Another target for power optimizations would be the noise buffer. It consumes roughly a third of the power at the highest current bias setting and because it is tied to the same bias generator as the OTA, its power consumption increases at higher current settings. For a noise buffer, this current consumption may be unnecessary, as the primary purpose of having multiple current bias settings was to modify the speed of the neuron. Using a fixed bias current for the noise buffer across the bias settings, or at least decoupling the buffer from the OTA's bias generator and tying the noise buffer its own biasing circuit, could decrease its power consumption significantly. The super source follower could also be replaced with a topology that consumes less power, so long as the performance of the integrated neuron is not affected significantly.

# 5 Performance Analysis of Integrated Neuron

## 5.1 Autocorrelation Performance Analysis

The autocorrelation function derived from the output of the integrated neuron is one of the two metrics used to evaluate the performance of the integrated neuron. The goal of this analysis is to see if there is a trend in the autocorrelation function of an isolated integrated neuron across the different current bias settings (which correspond to the integrated neuron's speed settings, with b6 being the fastest) and across prelayout to postlayout. For each speed setting, the autocorrelation function and OTA dominant pole was extracted from simulations at the 25℃ TT corner that involved setting one of the integrated neuron's subcomponents to its postlayout view while the others were held at prelayout. The vecmul, capdac, and the biasing circuits were excluded from this "one-hot" postlayout analysis; they were only simulated using their prelayout views. The autocorrelation was then also extracted for the full integrated neuron set to prelayout and the full integrated neuron set to postlayout, again excluding the synapse and biasing circuits. The synapse and biasing circuits were held at prelayout because changing those components to their postlayout views changed the bias points and capdac output voltage, which dramatically affected the autocorrelation function. To ensure a fair comparison between the one-hot postlayout, full prelayout, and full postlayout simulations, the biases and capdac output had to be unchanging across the simulations, so the synapse and biasing circuits were simulated using only their prelayout views.

After extracting the autocorrelation functions, an exponential of the form $y = Ae^{-K\Delta t} + (1 - A)$ was fit to the function to extract the coefficients, as seen from the examples in Figures 13 and 14. The constant offset term of the exponential can be set to 1-A because the autocorrelation function is always equal to 1 at $\Delta t = 0$ by definition. The resulting coefficients A and K were then tabulated in Tables

13a and 13b. The coefficient of interest for this analysis is the K coefficient, as it governs how quickly the autocorrelation function decays (and therefore the speed of the integrated neuron) and can be used as a measure of how quickly the integrated neuron switches. The K coefficient can then be compared to the OTA dominant pole to determine if there is a relationship between the OTA's dominant pole and the integrated neuron's speed.

| Speed | Set | K ($s^{-1}$) | A | OTA Dominant Pole |
|---|---|---|---|---|
| **b6** | Full Neuron Prelayout | 3.641e08 | 0.950 | 50.208 MHz |
| | Postlayout Sigmoid Gen | 3.430e08 | 0.938 | 50.006 MHz |
| | Postlayout Noise OTA | 2.817e08 | 0.929 | 76.161 MHz |
| | Postlayout Noise Buffer | 3.490e08 | 0.929 | 59.250 MHz |
| | Postlayout Noise Gen | 3.293e08 | 0.914 | 55.453 MHz |
| | Full Neuron Postlayout | 2.652e08 | 0.933 | 77.137 MHz |
| **b5** | Full Neuron Prelayout | 2.451e08 | 0.829 | 29.085 MHz |
| | Postlayout Sigmoid Gen | 2.282e08 | 0.909 | 28.972 MHz |
| | Postlayout Noise OTA | 1.970e08 | 0.925 | 28.748 MHz |
| | Postlayout Noise Buffer | 2.178e08 | 0.905 | 29.362 MHz |
| | Postlayout Noise Gen | 2.169e08 | 0.910 | 28.357 MHz |
| | Full Neuron Postlayout | 1.864e08 | 0.912 | 28.647 MHz |
| **b4** | Full Neuron Prelayout | 2.031e08 | 0.928 | 23.741 MHz |
| | Postlayout Sigmoid Gen | 1.868e08 | 0.881 | 23.642 MHz |
| | Postlayout Noise OTA | 1.529e08 | 0.874 | 22.420 MHz |
| | Postlayout Noise Buffer | 1.698e08 | 0.879 | 24.238 MHz |
| | Postlayout Noise Gen | 1.762e08 | 0.903 | 22.937 MHz |
| | Full Neuron Postlayout | 1.479e08 | 0.900 | 22.143 MHz |

Table 13a: Performance of integrated neuron at speeds b6-b4

| Speed | Set | K ($s^{-1}$) | A | Amplifier Dominant Pole |
|---|---|---|---|---|
| **b0** | Full Neuron Prelayout | 1.656e08 | 0.882 | 15.810 MHz |
| | Postlayout Sigmoid Gen | 1.065e08 | 0.836 | 15.756 MHz |
| | Postlayout Noise OTA | 1.002e08 | 0.919 | 14.794 MHz |
| | Postlayout Noise Buffer | 1.063e08 | 0.789 | 16.085 MHz |
| | Postlayout Noise Gen | 1.265e08 | 0.902 | 15.303 MHz |
| | Full Postlayout | 9.860e07 | 0.868 | 14.584 MHz |
| **slowest** | Full Neuron Prelayout | 1.418e08 | 0.893 | 14.864 MHz |
| | Postlayout Sigmoid Gen | 1.068e08 | 0.846 | 14.906 MHz |
| | Postlayout Noise OTA | 1.080e08 | 0.895 | 14.001 MHz |
| | Postlayout Noise Buffer | 1.012e08 | 0.783 | 15.208 MHz |
| | Postlayout Noise Gen | 1.151e08 | 0.888 | 14.489 MHz |
| | Full Postlayout | 9.374e07 | 0.884 | 13.805 MHz |

Table 13b: Performance of integrated neuron at speeds b0-slowest

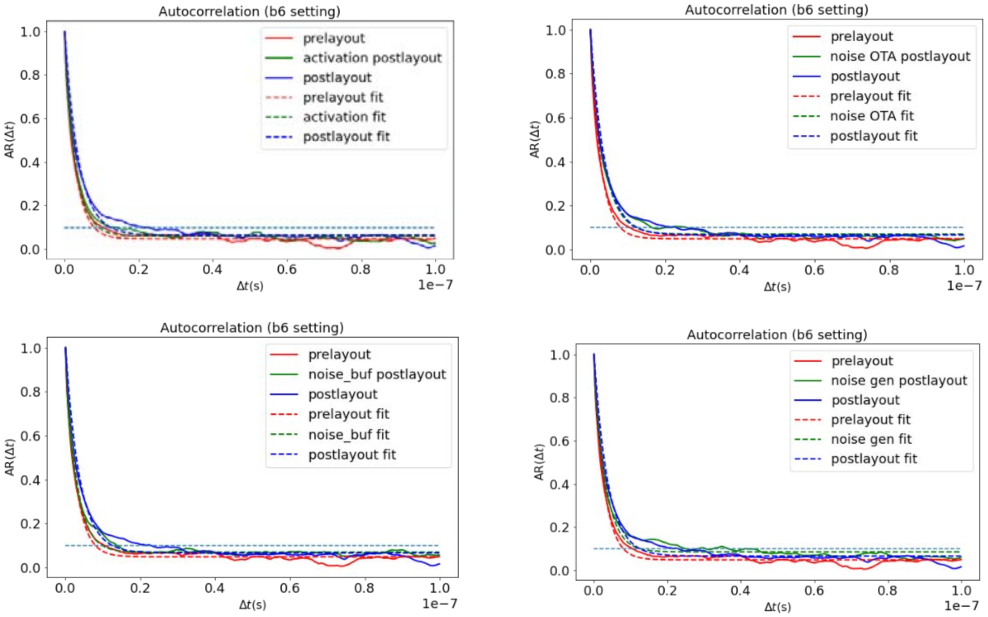

Figure 13: Autocorrelation functions for b6 speed setting

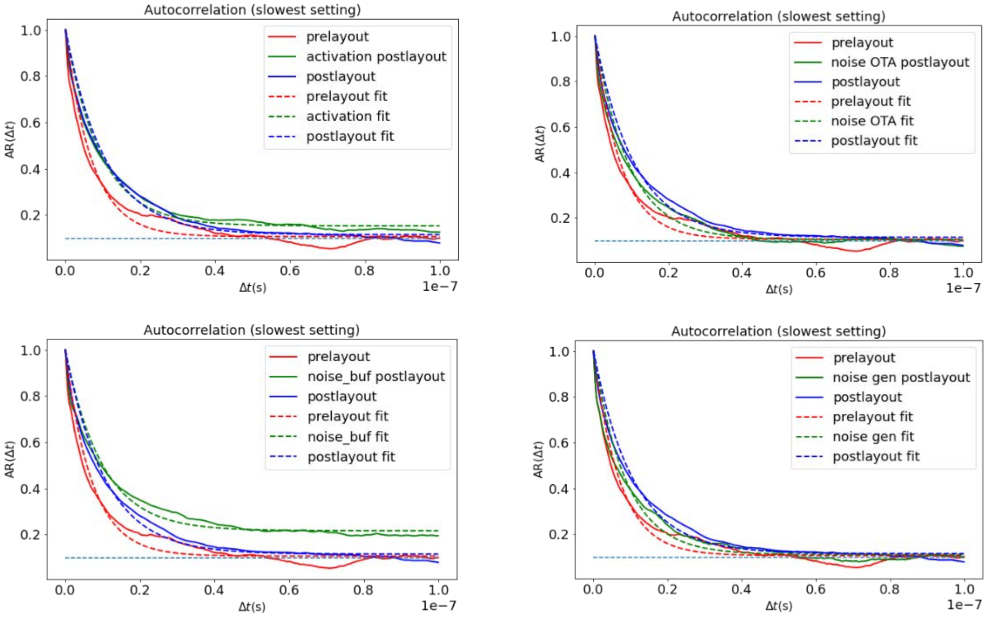

Figure 14: Autocorrelation functions for slowest speed setting

There are a few clear trends that can be drawn from the data. First, there is a decrease in the speed of the integrated neuron when moving from prelayout to postlayout simulations, which is expected. Second, the one-hot postlayout simulations also show a decrease in the speed of the integrated neuron for all components, indicating that they all contribute to the autocorrelation function in some way.

Out of all the components, the noise OTA seems to contribute the most towards the autocorrelation function, as changing it from prelayout to postlayout yielded the biggest difference in the value of K for most speeds. Intuitively, this makes sense; a slower amplifier should result in a slower integrated neuron. However, comparing the K values to the amplifier dominant pole does not yield a robust trend. While for some speed settings and some components, the K value matches $2\pi$ times the amplifier dominant pole with 10% error, there are enough points that do not follow any pattern such that no conclusive trend can be drawn. Because of this, even the trend of the noise OTA contributing the most towards the autocorrelation function may not be statistically significant and could just be due to random chance. A larger sample size and better modelling of the system would be necessary to draw any definitive conclusions about what affects the autocorrelation and speed of the integrated neurons.

## 5.2 Delay Analysis

The second metric used to evaluate the performance of the integrated neuron is the component-to-component propagation delay. To get the delay, a square wave input was used to model a neighboring integrated neuron, and the weight corresponding to that input was set to its maximum value to generate maximal switching for the vecmul and capdac. The prelayout delays from the inputs of each component reaching 50% of their final value to the outputs of each component reaching

50% of its final value was measured and averaged across 30 samples. These delays were computed across the SS, TT, FF, SF, and FS process corners at 25°C and 85°C at the slowest and fastest settings for the biasing circuits.

The results are tabulated in Tables 14-18. Since the noise generation and amplifier do not take any external inputs, they are excluded from the delay computations. The "total" delay refers to the propagation delay from the input to the output of the integrated neuron itself (from input of vecmul to output of inverters). The critical path of the integrated neuron goes through the vecmul to bit 5 of the vecmul output, through the capdac, through the sigmoid generator, and finally through the output inverters to reach the output of the integrated neuron.

| **TT** | **25°C** | | **85°C** | |
|---|---|---|---|---|
| | Slowest | Fastest | Slowest | Fastest |
| **Total** | 254.4 ps | 209.3 ps | 282.1 ps | 229.8 ps |
| **Vecmul** | 126 ps | 125.8 ps | 132 ps | 132.1 ps |
| **Capdac** | 9.006 ps | 8.863 ps | 8.992 ps | 8.967 ps |
| **Sigmoid** | 107.3 ps | 46.44 ps | 94.64 ps | 50.03 ps |
| **Output inverters** | 11.85 ps | 28.03 ps | 48.49 ps | 39.39 ps |

Table 14: Delay values for each component in the TT process corner

| **FF** | **25°C** | | **85°C** | |
|---|---|---|---|---|
| | Slowest | Fastest | Slowest | Fastest |
| **Total** | 242 ps | 199.3 ps | 253.2 ps | 205.2 ps |
| **Vecmul** | 108.3 ps | 108.3 ps | 111.9 ps | 111.9 ps |
| **Capdac** | 12.84 ps | 12.89 ps | 9.843 ps | 9.996 ps |
| **Sigmoid** | 99.48 ps | 45.8 ps | 85.13 ps | 47.45 ps |
| **Output inverters** | 21.13 ps | 32.32 ps | 46.41 ps | 35.88 ps |

Table 15: Propagation delay of each component in the FF process corner

| **SS** | **25°C** | | **85°C** | |
|---|---|---|---|---|
| | Slowest | Fastest | Slowest | Fastest |
| **Total** | 302.3 ps | 248.1 ps | 296.5 ps | 247.9 ps |
| **Vecmul** | 151.9 ps | 152.4 ps | 153.7 ps | 153.1 ps |
| **Capdac** | 10.1 ps | 10.52 ps | 8.819 ps | 10.89 ps |
| **Sigmoid** | 112 ps | 48.19 ps | 98.58 ps | 45.43 ps |
| **Output inverters** | 27.22 ps | 36.65 ps | 36.69 ps | 38.03 ps |

Table 16: Propagation delay of each component in the SS process corner

| SF | 25°C | | 85°C | |
|---|---|---|---|---|
| | Slowest | Fastest | Slowest | Fastest |
| **Total** | 250.8 ps | 206.3 ps | 279.6 ps | 221.6 ps |
| **Vecmul** | 116.2 ps | 116.3 ps | 121.3 ps | 121.3 ps |
| **Capdac** | 14.41 ps | 14.57 ps | 12.71 ps | 12.76 ps |
| **Sigmoid** | 115.8 ps | 47.43 ps | 105.8 ps | 47.25 ps |
| **Output inverters** | 1.722 ps | 28.24 ps | 40.07 ps | 41.15 ps |

Table 17: Propagation delay of each component in the SF process corner

| FS | 25°C | | 85°C | |
|---|---|---|---|---|
| | Slowest | Fastest | Slowest | Fastest |
| **Total** | 265.5 ps | 234.8 ps | 278.5 ps | 238.8 ps |
| **Vecmul** | 140.1 ps | 140.1 ps | 143.1 ps | 143.5 ps |
| **Capdac** | 8.316 ps | 8.324 ps | 9.416 ps | 8.411 ps |
| **Sigmoid** | 101 ps | 47.99 ps | 88.62 ps | 47.42 ps |
| **Output inverters** | 16.81 ps | 38.42 ps | 38.03 ps | 39.46 ps |

Table 18: Propagation delay of each component in the FS process corner

As expected, the corner with the shortest delays was the FF process corner, and the corner with the longest delays was SS, while the TT, FS, and SF corners were all somewhere in the middle. For each corner, the delay at 25°C was shorter than it was at 85°C as expected, since the mobility decreases at higher temperatures.

The longest delay is the vecmul propagation from its input to its output bit 5. The capdac delay is the shortest delay across the different corners, speeds, and temperatures because it overlaps with the vecmul delay; while bit 5 of the vecmul is the longest delay, the other bits have already propagated, allowing the capdac voltage to change before bit 5 finishes propagating. As the sigmoid generator is tied to the current bias, changing the current from the slowest to fastest setting decreases the delay of the sigmoid generation block significantly. Based on these figures, the primary target for optimizing to reduce the delay is the vecmul.

# 6 Design Exploration

Based on the previous analyses, the best block to optimize for area would be the capdac, the best block to optimize for power would be the noise OTA or buffer, and the best block to optimize for delay performance would be the vecmul.

Out of these analyses, the power consumption is the most pressing issue. Since there is still unused die area on the processor, and since the propagation delays are significantly shorter than the neuron sampling period, the priority is to minimize the power consumption of the integrated neurons to enable the scaling of the number of integrated neurons for future processors. With that, this report will discuss a topology that was explored as a lower-power replacement for the super source follower that makes up the noise buffer: the “flipped voltage follower” [4].

### 6.1 Modified Flipped Voltage Follower (Prelayout)

The flipped voltage follower, shown in Figure 15, is a topology similar to the super source follower (SSF), but the feedback transistor is the same type as the input transistor (NFET for n-input, PFET for p-input) [4]. The flipped voltage follower has a low output resistance of roughly $2/(g_{m1}g_{m2}r_{o1})$ [4] and could potentially be a lower power design than the super source follower because it only needs a single bias for its single current source. As the super source follower uses two current sources, it requires two biases (one PFET, one NFET). To produce NFET bias in the current noise buffer design, the input PFET current needs to be mirrored onto an extra branch connected to an NFET mirror (see Figure 5), resulting in extra current being drawn for that path. The FVF does not need this extra path, and so would save that current.

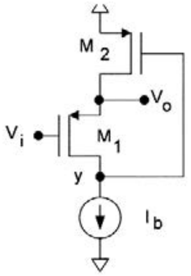

Figure 15: Schematic of basic flipped voltage follower

A modified n-input version of this flipped voltage follower (FVF), shown in Figure 16, was explored as an alternative to the current super source follower design. Because of the input and output common mode restrictions (input of 643mV, output 400mV), to ensure M1 stayed in saturation, M1 needed to have a relatively large size, and Vfb needed to be biased at roughly 600mV. M2 needed to be sized much smaller than M1 to increase its Vgs (and therefore Vfb) as much as possible, but Vfb was not high enough through sizing M2 alone. A diode-connected transistor M3 needed to be added between the source of M2 and VSS to forcibly push the Vfb node up in voltage.

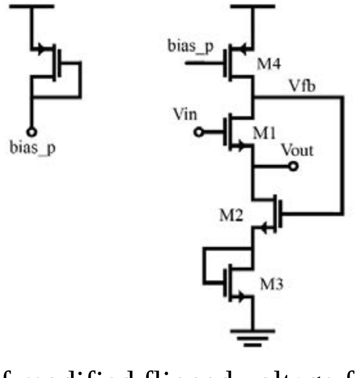

Figure 16: Schematic of modified flipped voltage follower for noise buffer

The modified FVF was designed with a length of 160nm, as 160nm length devices produced the best analog performance in terms of $g_m$, $r_o$, etc. It takes in a

2uA current input and mirrors it to a nominal 10 uA bias current flowing through the main transistor stack.

The plot of the output resistance $R_{out}$ versus frequency of the modified FVF is shown in Figure 17, and for comparison, the plot of $R_{out}$ versus frequency of the SSF is shown in Figure 18. The values of $R_{out}$ at 1 MHz, 10MHz, 100 MHz, and 1 GHz for both designs are tabulated in Table 19.

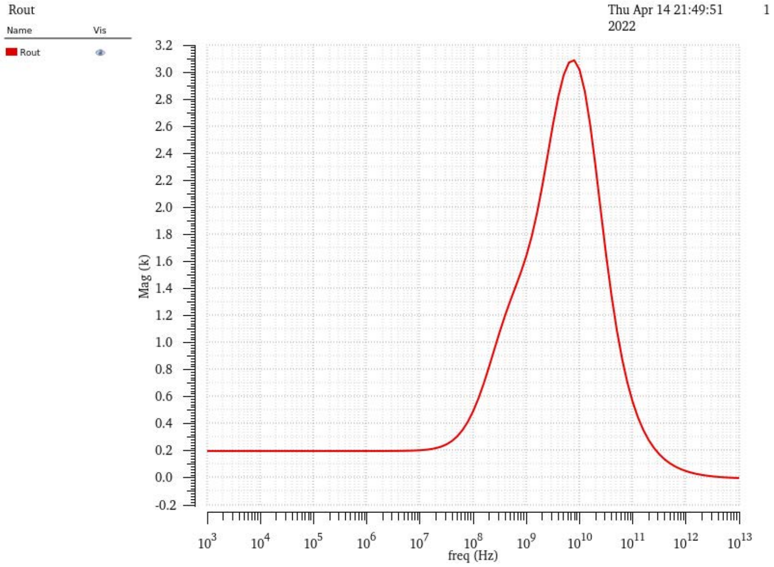

Figure 17: $R_{out}$ of modified FVF

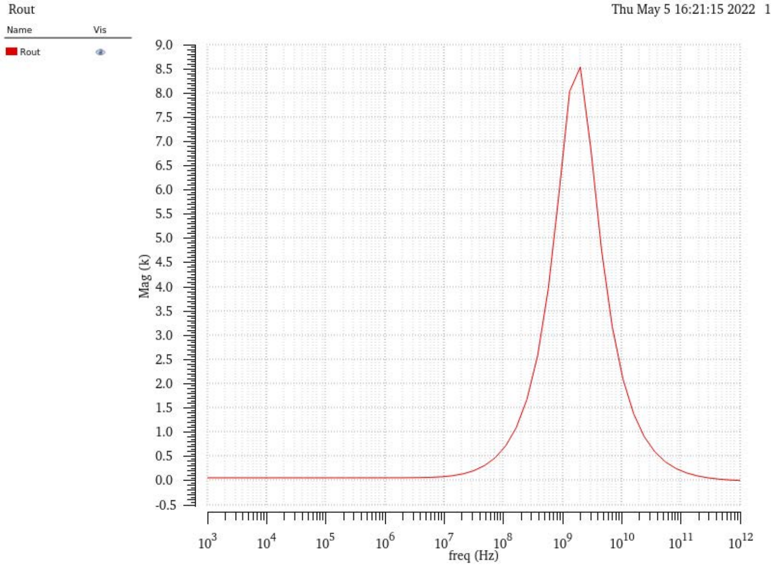

Figure 18: $R_{out}$ of SSF

| Design | $R_{out}$ @ 1 MHz | $R_{out}$ @ 10 MHz | $R_{out}$ @ 100 MHz | $R_{out}$ @ 1 GHz |
|---|---|---|---|---|
| **Modified FVF** | 205.98 Ω | 211.74 Ω | 498.75 Ω | 1645.23 Ω |
| **SSF** | 78.19 Ω | 102.83 Ω | 672.43 Ω | 6538.7 Ω |

Table 19: Comparison of $R_{out}$ for modified FVF and SSF

As seen from the figures and table, the $R_{out}$ of the SSF is less than half of the $R_{out}$ of the modified FVF at frequencies below 10 MHz; however, the $R_{out}$ of the SSF increases rapidly after 10 MHz at a faster rate than the $R_{out}$ for the modified FVF does. At 100 MHz and above, the modified FVF has less $R_{out}$ than the SSF. This suggests that the low frequency/DC gain from the noise buffer input to noise OTA output will be higher for the SSF, but at higher frequencies it will be higher for the modified FVF. Using only the $R_{out}$ as a metric for comparison does not yield an immediate answer for which design yields a higher performance, so other metrics must be considered.

The gain and phase of the modified FVF transfer function is shown in Figure 19, while the gain and phase of the SSF transfer function is shown for comparison in Figure 20. Table 20 tabulates the low frequency gain and corner frequency of the two designs. These simulations were run with a purely capacitive load.

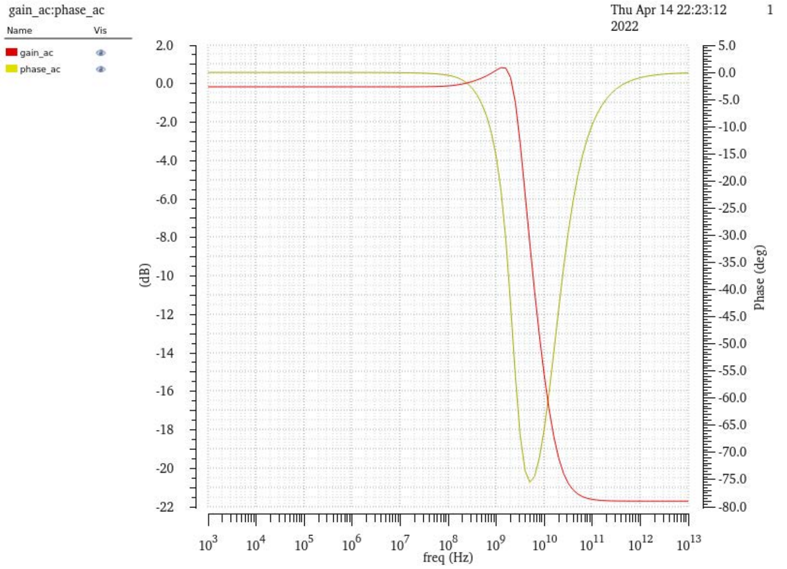

Figure 19: Gain (red) and phase (yellow) of the modified FVF transfer function

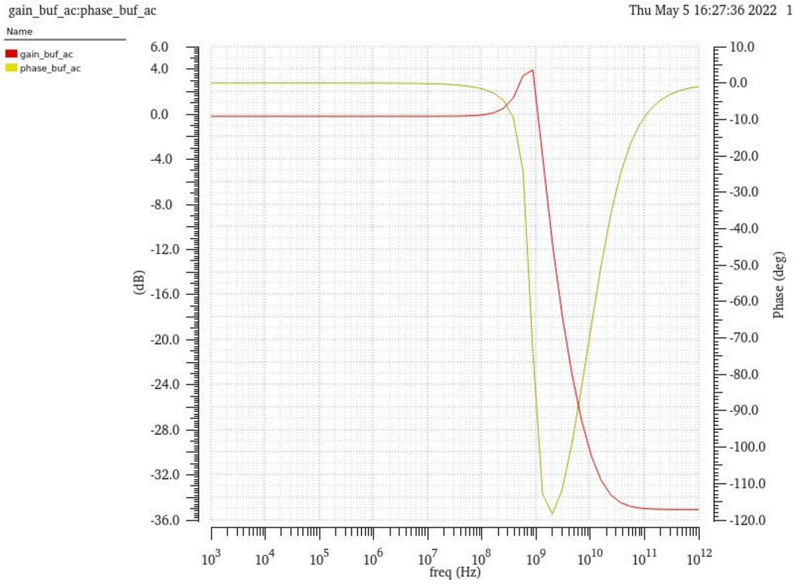

Figure 20: Gain (red) and phase (yellow) of the SSF transfer function

| Design | Low Frequency Gain | Corner Frequency |
|---|---|---|
| **Modified FVF** | -0.157 dB | 3.015 GHz |
| **SSF** | -0.167 dB | 1.287 GHz |

Table 20: Comparison of gain for modified FVF and SSF

The low frequency gains of the two designs are almost identical, while the corner frequency of the modified FVF is higher than that of the SSF. The plots show that the gain of the modified FVF and SSF both go above unity gain for a small frequency range; however, there is still sufficient phase margin for both designs such that stability is not an issue and compensation is not needed.

The outputs of the two designs were then connected to the noise OTA, forming the complete noise amplifier, and simulated. Figures 21 (modified FVF) and 22 (SSF) show the gain of the noise buffer driving the noise OTA as its load, and Table 21 tabulates that gain evaluated at different frequencies.

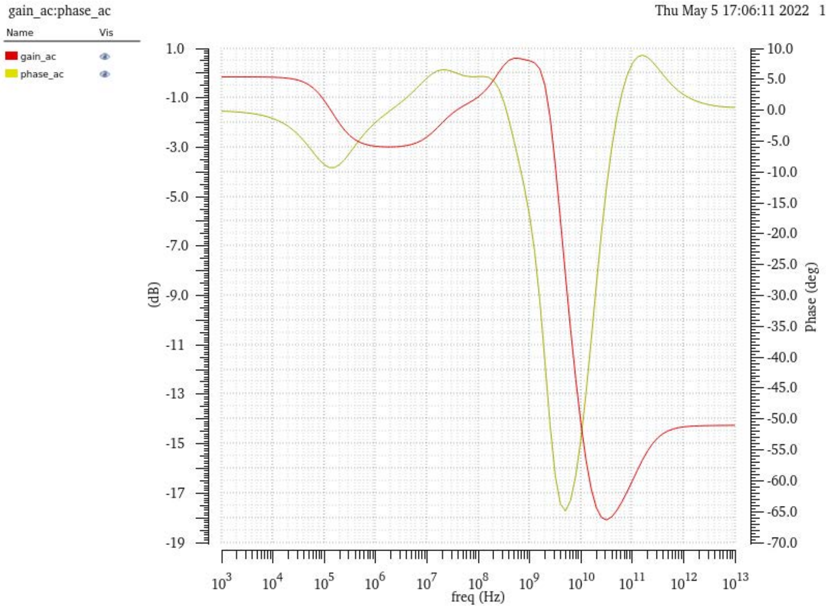

Figure 21: Gain (red) and phase (yellow) of loaded modified FVF

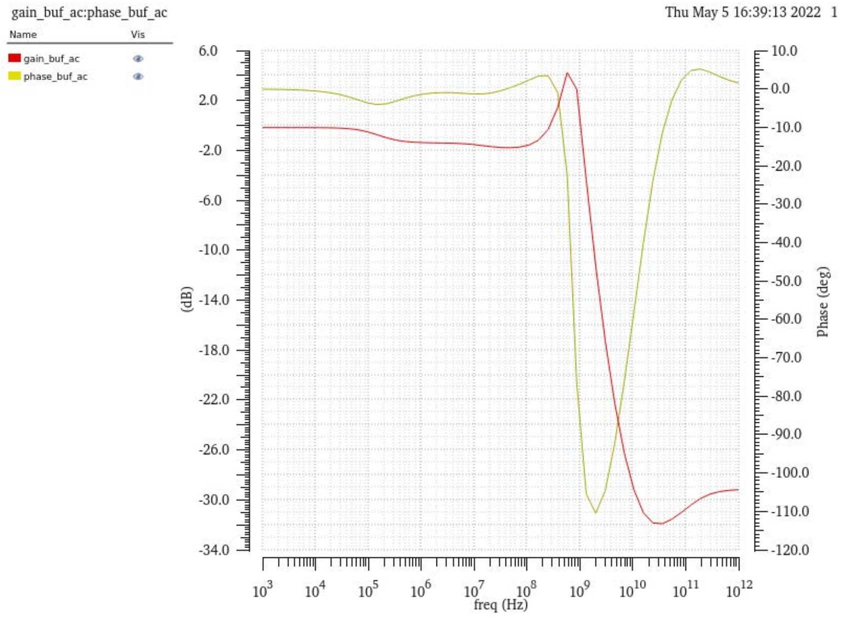

Figure 22: Gain (red) and phase (yellow) of loaded SSF

| Design | Loaded Gain @ 1 MHz | Loaded Gain @ 10 MHz | Loaded Gain @ 100 MHz | Loaded Gain @ 1 GHz |
|---|---|---|---|---|
| **Modified FVF** | -2.953 dB | -2.578 dB | -0.959 dB | 0.501 dB |
| **SSF** | -1.357 dB | -1.522 dB | -1.598 dB | 0.817 dB |

Table 21: Comparison of gain of loaded modified FVF and loaded SSF

With the noise OTA as its load, the modified FVF gain degrades significantly compared to the SSF gain. This is likely due to the difference in the $R_{out}$ between the modified FVF and SSF, as the noise OTA is not a purely capacitive load due to the resistive feedback.

Figures 23 (modified FVF) and 24 (SSF) show the gain and phase of the noise amplifier transfer function (from the input of the noise buffer to the output of the noise OTA), and Table 22 tabulates that gain evaluated at different frequencies as well as the corner frequency. The gain of the noise amplifier transfer function will be referred to as amp_gain.

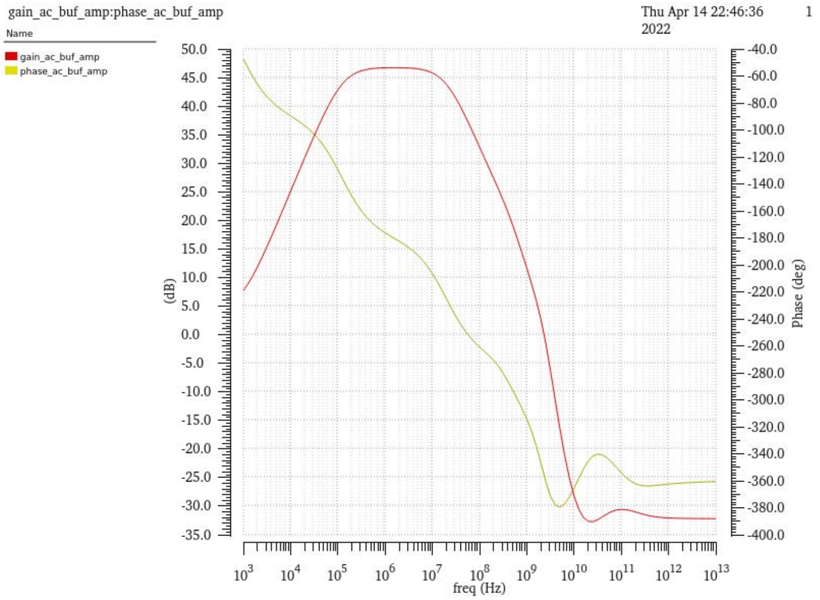

Figure 23: Gain (red) and phase (yellow) plot of the transfer function from input of modified FVF to output of noise OTA.

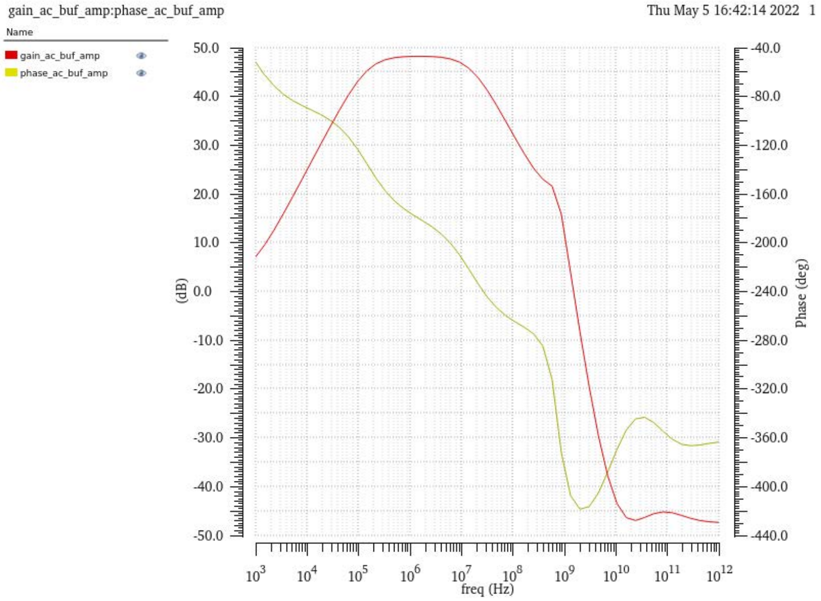

Figure 24: Gain (red) and phase (yellow) plot of the transfer function from input of SSF to output of noise OTA.

| Design | Amp_gain @ 1 MHz | Amp_gain @ 10 MHz | Amp_gain @ 100 MHz | Corner Frequency |
|---|---|---|---|---|
| **Modified FVF** | 46.719 dB | 45.824 dB | 32.855 dB | 20.328 MHz |
| **SSF** | 48.2 dB | 46.827 dB | 32.437 dB | 16.128 MHz |

Table 22: Comparison of amp_gain for the modified FVF and SSF

As expected, the modified FVF amp_gain is lower than that of the SSF design due to the higher $R_{out}$, though it still meets the gain requirements of 200 (46.02 dB) at 1 MHz. Beyond that frequency, however, the gain drops past the desired value, which could harm the integrated neuron's performance. As the SSF can meet the target gain of 200 for a wider range of frequencies, the performance of the integrated neuron using the SSF is expected to be higher than a integrated neuron using the modified FVF.

The variation of the $R_{out,}$ gain, amp_gain, and common mode output voltage was characterized and tabulated in Table 23 using a 1000-sample Monte Carlo simulation, in which only the noise buffer's transistors were subject to variation.

| | **Modified FVF** | | **SSF** | |
|---|---|---|---|---|
| **Parameter** | Mean | Std Dev | Mean | Std Dev |
| **Output common mode** | 394.9 mV | 9.728 mV | 385.3 mV | 9.127 mV |
| **$R_{out}$ @ 10 MHz** | 199.8 Ω | 44.18 Ω | 103.7 Ω | 7.822 Ω |
| **$R_{out}$ @ 100 MHz** | 440.2 Ω | 29.6 Ω | 676.4 Ω | 68.66 Ω |
| **$R_{out}$ @ 1 GHz** | 1525 Ω | 40.25 Ω | 6416 Ω | 553.1 Ω |
| **Gain @ 1 MHz** | -0.1664 dB | 0.02273 dB | -0.1701 dB | 0.01459 dB |
| **Gain @ 10 MHz** | -0.1507 dB | 0.02273 dB | -0.1677 dB | 0.01467 dB |
| **Gain @ 100 MHz** | -0.1183 dB | 0.02289 dB | 0.07309 dB | 0.03769 dB |
| **Gain @ 1 Ghz** | 0.7396 dB | 0.07655 dB | -6.828 dB | 0.9361 dB |
| **Loaded Gain @ 1 MHz** | -3.027 dB | 0.1973 dB | -1.301 dB | 0.102 dB |
| **Loaded Gain @ 10 MHz** | -2.654 dB | 0.2223 dB | -1.768 dB | 0.1398 dB |
| **Loaded Gain @ 100 MHz** | -0.9705 dB | 0.1052 dB | -2.093 dB | 0.2151 dB |
| **Loaded Gain @ 1 Ghz** | 0.4695 dB | 0.08297 dB | -9.248 dB | 0.9489 dB |
| **Amp_gain @ 1 MHz** | 46.59 dB | 0.429 dB | 47.68 dB | 0.4195 dB |
| **Amp_gain @ 10 MHz** | 45.69 dB | 0.3794 dB | 45.89 dB | 0.4214 dB |
| **Amp_gain @ 100 MHz** | 32.71 dB | 0.5667 dB | 30.88 dB | 0.5499 dB |
| **Amp_gain @ 1 GHz** | 11.58 dB | 0.6726 dB | 0.086 dB | 1.4 dB |

Table 23: Variation results from 1000-sample Monte Carlo simulation

The standard deviation in the output common mode is almost the same between the two designs; at a roughly 10 mV standard deviation, the common mode is sufficiently controlled for the integrated neuron. The standard deviation in $R_{out}$ for the SSF remains at roughly 10% of the mean value, whereas the standard deviation in $R_{out}$ for the modified FVF remained relatively at a constant $R_{out}$ value. The large variation in the low frequency $R_{out}$ of the modified FVF likely comes from the fact that the feedback transistor M2 is extremely small, almost minimally sized, so variations in that transistor have a large effect on the characteristics of the design. All three gain measurements for the SSF design also showed a trend of the 1 GHz values being significantly different from the 1, 10, and 100 MHz values in terms of both mean and standard deviation. Apart from these patterns, there aren't many trends that can be drawn from the rest of the data, so in terms of variation, it is difficult to determine if one design is better than the other.

The input capacitance $C_{in}$ of each noise buffer was characterized using a transient simulation. Then, the noise source subcircuit was connected to the input of the buffer and a transient noise simulation was run to characterize the RMS of the noise signals at the output of each block. The results are compiled in Table 24.

| Parameter | Modified FVF | SSF |
|---|---|---|
| **Noise buffer $C_{in}$** | 7.347 fF | 1.994 fF |
| **Noise source out RMS** | 0.713 mV | 1.935 mV |
| **Noise buffer out RMS** | 0.890 mV | 3.051 mV |
| **Noise OTA out RMS** | 97.24 mV | 150.559 mV |

Table 24: RMS results from transient noise simulation

The RMS voltage of the modified FVF design is lower than that of the SSF design at every stage. The RMS of the source output has degraded because of the large increase in input capacitance compared to the SSF design, which ends up loading the noise source. The modified FVF design also has a lower amp_gain than the SSF, which contributes to the large discrepancy in the RMS voltage at the noise OTA output.

In terms of power, the results are tabulated in Table 25 using the isolated integrated neuron simulation from the power analysis at the slowest setting. The modified FVF design offers a 37.5% reduction in the power consumption.

| Design | Average Current | Peak Current |
|---|---|---|
| **Modified FVF** | 11.92 uA | 12.7463 uA |
| **SSF** | 19.04 uA | 22.232 uA |

Table 25: Power comparisons between modified FVF and SSF designs.

As a final metric for comparison between the two designs, the autocorrelation functions were extracted, fitted, and tabulated in Table 26. Since the modified FVF was decoupled from the bias circuit, its bias current did not change with the speed setting. To have a fair point of comparison, a separate autocorrelation function was extracted from the simulations decoupling the SSF from the bias circuit as well.

| Speed | Design | K ($s^{-1}$) | A |
|---|---|---|---|
| **b6** | Modified FVF | 1.057e08 | 0.847 |
| | Decoupled SSF | 2.178e08 | 0.917 |
| | SSF | 3.641e08 | 0.950 |
| **b5** | Modified FVF | 9.930e07 | 0.867 |
| | Decoupled SSF | 1.792e08 | 0.891 |
| | SSF | 2.451e08 | 0.829 |
| **b4** | Modified FVF | 9.964e07 | 0.850 |
| | Decoupled SSF | 1.711e08 | 0.887 |
| | SSF | 2.031e08 | 0.928 |
| **b0** | Modified FVF | 1.113e08 | 0.924 |
| | Decoupled SSF | 1.268e08 | 0.830 |
| | SSF | 1.656e08 | 0.882 |
| **slowest** | Modified FVF | 1.038e08 | 0.939 |
| | Decoupled SSF | 1.277e08 | 0.822 |
| | SSF | 1.418e08 | 0.893 |

Table 26: Results of autocorrelation function extraction of prelayout designs

The K coefficient for the modified FVF design was 50% smaller than that of the SSF, even after decoupling the SSF from the current bias, indicating that the speed of the integrated neuron was cut in half. Much of this can be attributed to the reduction in the RMS of the noise OTA output, which in turn is attributed to the large input capacitance, higher low-frequency $R_{out}$, and lower low-frequency gain of the modified FVF design. Using the autocorrelation function, the SSF clearly outperforms the modified FVF design.

The decoupled SSF also showed worse performance than the coupled SSF across all speed settings. Having a trimmable current bias to tune the speed of the noise buffer can improve the performance of the integrated neuron at the cost of burning some more power. If the power consumption is not a limiting factor, then setting the noise buffer to a higher speed setting makes for a simple method to improve the performance of the integrated neuron.

### 6.2 Prelayout Modified FVF Conclusion

While the modified FVF decreases the current consumed by the noise buffer by 35%, it also cut the speed of the integrated neuron by 50%, and thus is more suitable for a lower power implementation, where the priority is power efficiency over rather than speed. The decrease in the speed can be attributed mainly to the large input capacitance, which increased the load on the noise source and thus decreased the RMS voltage of its output, as well as higher output resistance, which lowered the gain of the noise amplifier. In terms of $R_{out}$, the low frequency resistance matters more than the resistance at higher frequencies, as the SSF had a significantly higher resistance at high frequencies beyond 100 MHz but had better performance than the modified FVF. The input and output common modes and input noise source were a poor match for this topology; because of those operating conditions, keeping the input transistor in saturation required making it very large, while the input noise source required a small load capacitance.

Although the modified FVF topology is not currently feasible as a replacement, it may be a viable option if the speed of the integrated neuron is not a limiting factor for the chip performance. It may turn out that the integrated neuron doesn't need to be as fast and can tolerate a degradation in speed in favor of power savings, in which case this topology may be a suitable design. If the requirements for the input or output common mode are changed via the redesign of the noise source or OTA, this topology become more viable. For that reason, the design was pushed through layout and postlayout simulations.

## 6.3 Modified Flipped Voltage Follower (Postlayout)

Figure 25 shows the layout of the modified FVF. Its dimensions are 3.36 um x 10.71 um, resulting in a total area of 35.99 um$^2$.

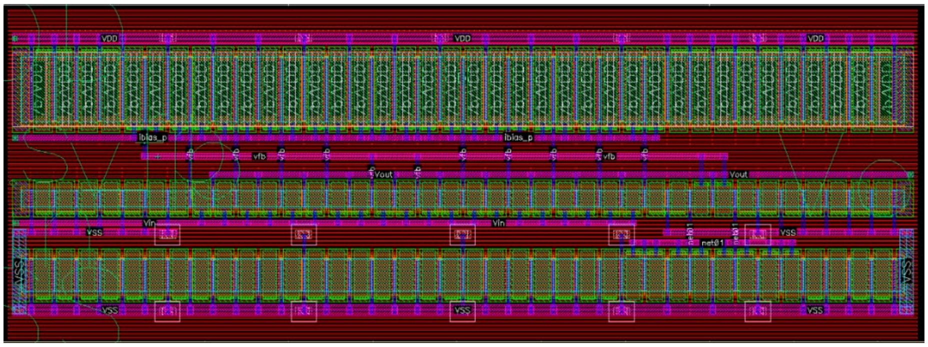

Figure 25: Layout of Modified FVF

The plot of the output resistance $R_{out}$ versus frequency of the postlayout modified FVF is shown in Figure 26, and for comparison, the plot of $R_{out}$ versus frequency of the postlayout SSF is shown in Figure 27. The values of $R_{out}$ at 1 MHz, 10MHz, 100 MHz, and 1 GHz for both designs are tabulated in Table 27.

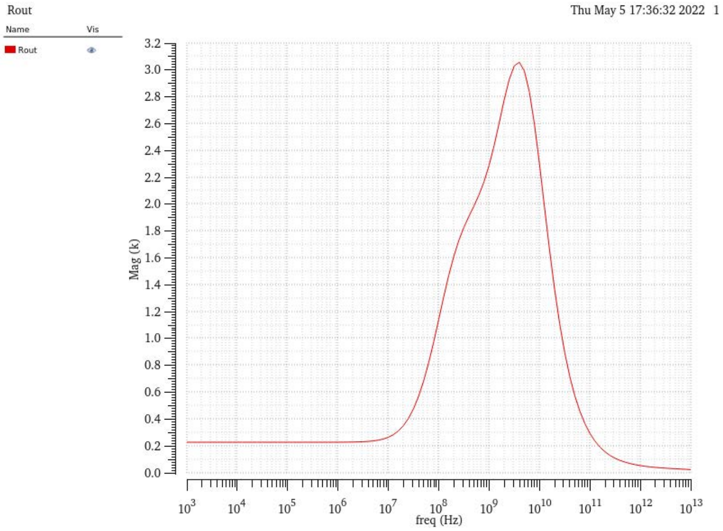

Figure 26: $R_{out}$ of postlayout modified FVF

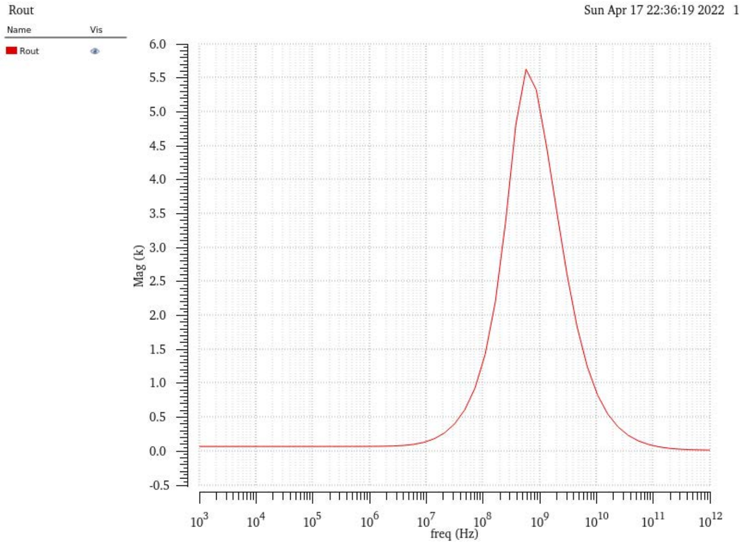

Figure 27: $R_{out}$ of postlayout SSF

| Design (postlayout) | $R_{out}$ @ 1 MHz | $R_{out}$ @ 10 MHz | $R_{out}$ @ 100 MHz | $R_{out}$ @ 1 GHz |
|---|---|---|---|---|
| **Modified FVF** | 234.404 Ω | 271.035 Ω | 1147.46 Ω | 2292.76 Ω |
| **SSF** | 83.998 Ω | 153.06 Ω | 1309.7 Ω | 5087.8 Ω |

Table 27: Comparison of $R_{out}$ for postlayout modified FVF and postlayout SSF

Like in the prelayout trends, the $R_{out}$ of the postlayout SSF is less than half of the $R_{out}$ of the postlayout modified FVF at frequencies below 10 MHz, but increases rapidly after 10 MHz at a faster rate than the $R_{out}$ for the modified FVF does. At 100 MHz and above, the modified FVF has less $R_{out}$ than the SSF. Compared to the prelayout versions, the postlayout designs have an increased $R_{out}$ as expected.

The gain and phase of the postlayout modified FVF transfer function is shown in Figure 28, while the gain and phase of the postlayout SSF transfer function is shown for comparison in Figure 29. Table 28 tabulates the low frequency gain and corner frequency of the two postlayout designs. These simulations were run with a purely capacitive load.

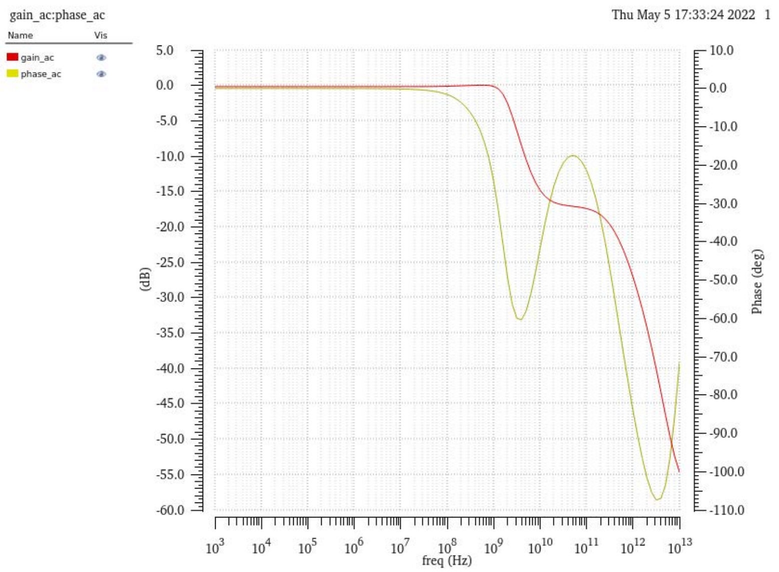

Figure 28: Gain (red) and phase (yellow) of the postlayout modified FVF transfer function

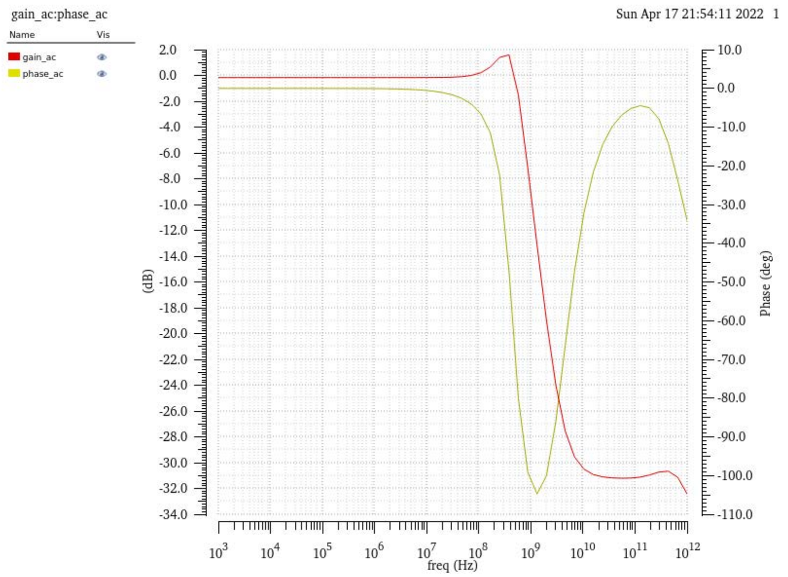

Figure 29: Gain (red) and phase (yellow) of the postlayout SSF transfer function

| Design (postlayout) | Low Frequency Gain | Corner Frequency |
|---|---|---|
| **Modified FVF** | -0.157 dB | 2.184 GHz |
| **SSF** | -0.162 dB | 662.704 MHz |

Table 28: Comparison of gain for postlayout modified FVF and postlayout SSF

The low frequency gains of the two postlayout designs are almost identical, just like in the prelayout simulations. The corner frequencies were also degraded compared to prelayout, as expected. The layout parasitics of the modified FVF seem to have pushed the zero previously found at 100 GHz in the prelayout simulation to the left, to 10 GHz in the postlayout, which could prove to be a problem for the performance of the integrated neuron later.

The outputs of the two designs were then connected to the noise OTA and simulated. Figures 30 (postlayout modified FVF) and 31 (postlayout SSF) show the gain of the noise buffer driving the noise OTA as its load, and Table 29 tabulates that gain evaluated at different frequencies.

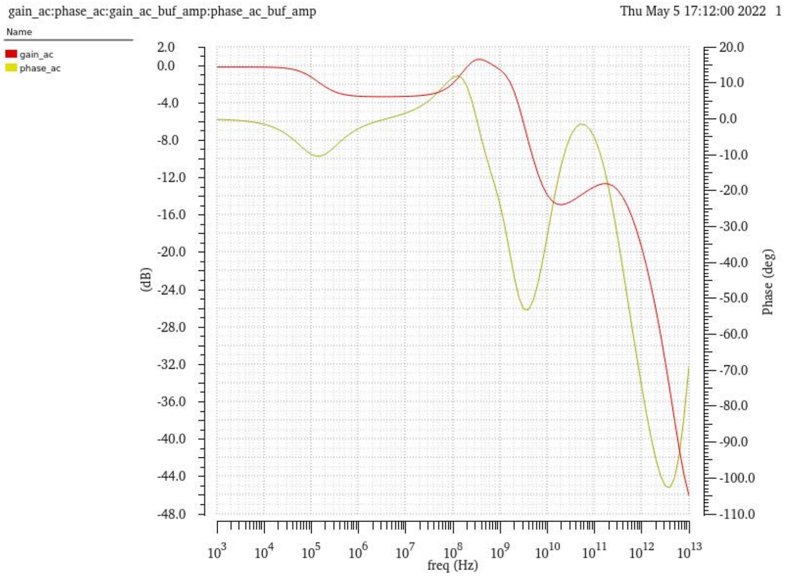

Figure 30: Gain (red) and phase (yellow) of loaded postlayout modified FVF

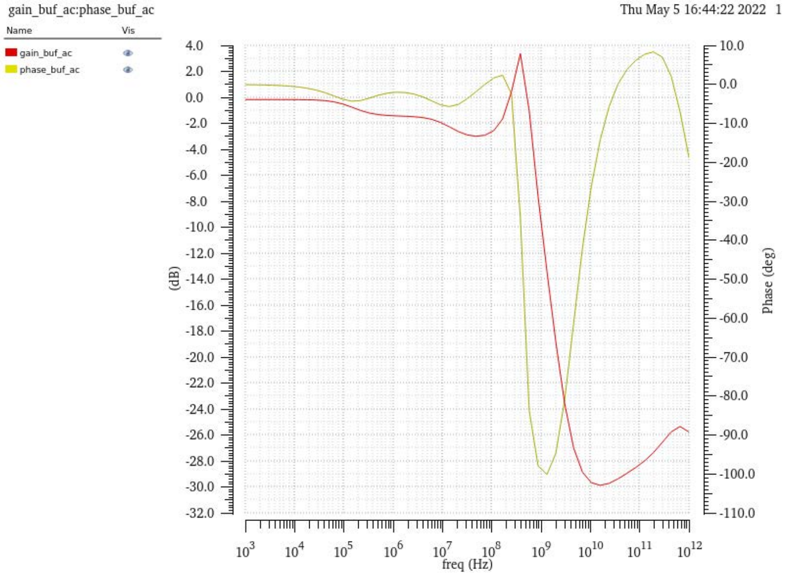

Figure 31: Gain (red) and phase (yellow) of loaded postlayout SSF

| Design (postlayout) | Loaded Gain @ 1 MHz | Loaded Gain @ 10 MHz | Loaded Gain @ 100 MHz | Loaded Gain @ 1 GHz |
|---|---|---|---|---|
| **Modified FVF** | -3.271 dB | -3.290 dB | -1.897 dB | -0.461 dB |
| **SSF** | -1.403 dB | -1.981 dB | -2.645 dB | -9.274 dB |

Table 29: Comparison of gain of loaded postlayout modified FVF and loaded postlayout SSF

With the noise OTA as its load, the postlayout modified FVF gain degrades significantly compared to the postlayout SSF gain, just like the prelayout simulations. The postlayout gain is also lower than the prelayout gain for both designs, as expected.

Figures 32 (postlayout modified FVF) and 33 (postlayout SSF) show the gain and phase of the noise amplifier transfer function, and Table 30 tabulates that gain evaluated at different frequencies as well as the corner frequency.

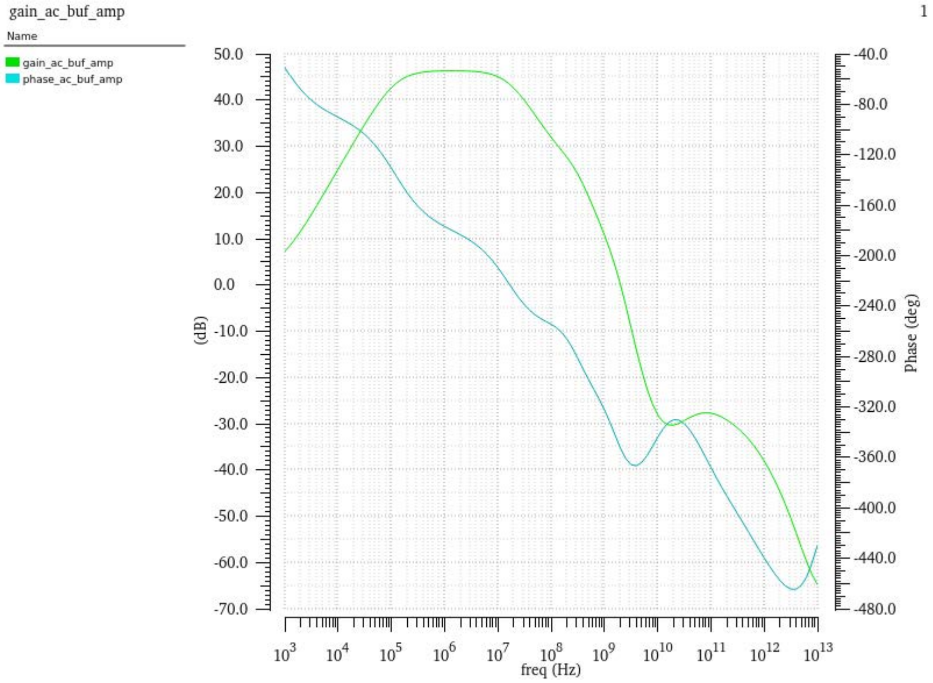

Figure 32: Gain (green) and phase (blue) plot of the transfer function from input of postlayout modified FVF to output of noise OTA.

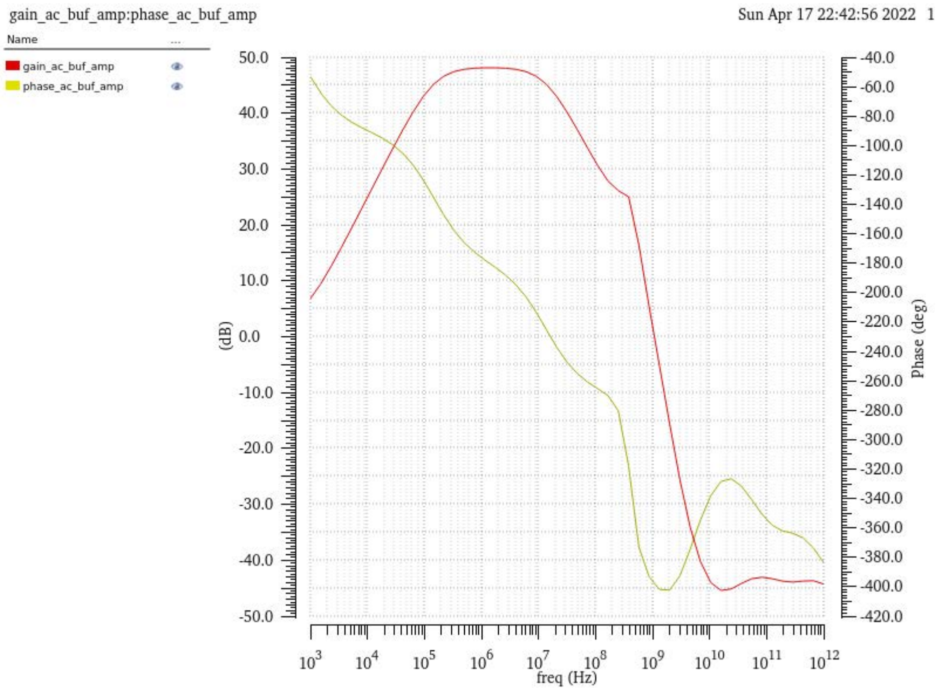

Figure 33: Gain (red) and phase (yellow) plot of the transfer function from input of postlayout SSF to output of noise OTA.

| Design (postlayout) | Amp_gain @ 1 MHz | Amp_gain @ 10 MHz | Amp_gain @ 100 MHz | Corner Frequency |
|---|---|---|---|---|
| **Modified FVF** | 46.324 dB | 45.080 dB | 31.982 dB | 17.110 MHz |
| **SSF** | 48.089 dB | 46.145 dB | 31.463 dB | 13.952 MHz |

Table 30: Comparison of amp_gain and corner frequency for the postlayout modified FVF and postlayout SSF

As expected, the postlayout modified FVF amp_gain is lower than that of the postlayout SSF design due to the higher $R_{out}$, like what was observed in prelayout. Both designs show a slight degradation in the corner frequency when compared to the prelayout simulations, as expected.

The variation of the $R_{out}$, gain, loaded gain, amp_gain, and common mode output voltage was characterized and tabulated in Table 31 using a 1000-sample Monte Carlo simulation, in which only the noise buffer's transistors were subject to variation.

| | Postlayout Modified FVF | | Postlayout SSF | |
|---|---|---|---|---|
| **Parameter** | Mean | Std Dev | Mean | Std Dev |
| **Output common mode** | 388.2 mV | 9.895 mV | 378 mV | 9.208 mV |
| **$R_{out}$ @ 10 MHz** | 283 Ω | 31.09 Ω | 154.5 Ω | 17.96 Ω |
| **$R_{out}$ @ 100 MHz** | 1164 Ω | 59.64 Ω | 1324 Ω | 180.6 Ω |
| **$R_{out}$ @ 1 GHz** | 2304 Ω | 101.2 Ω | 5035 Ω | 247.8 Ω |
| **Gain @ 1 MHz** | -0.1653 dB | 0.01741 dB | -0.1639 dB | 0.01128 dB |
| **Gain @ 10 MHz** | -0.1646 dB | 0.01789 dB | -0.1606 dB | 0.0114 dB |
| **Gain @ 100 MHz** | -0.1203 dB | 0.05412 dB | 0.158 dB | 0.0451 dB |
| **Gain @ 1 Ghz** | -0.1737 dB | 0.1976 dB | -8.862 dB | 0.8714 dB |
| **Loaded Gain @ 1 MHz** | -3.379 dB | 0.3192 dB | -1.405 dB | 0.0919 dB |
| **Loaded Gain @ 10 MHz** | -3.392 dB | 0.2718 dB | -1.993 dB | 0.1491 dB |
| **Loaded Gain @ 100 MHz** | -1.924 dB | 0.2081 dB | -2.658 dB | 0.2736 dB |
| **Loaded Gain @ 1 Ghz** | -0.523 dB | 0.2004 dB | -9.28 dB | 0.824 dB |
| **Amp_gain @ 1 MHz** | 46.15 dB | 0.6104 dB | 48.01 dB | 0.4403 dB |
| **Amp_gain @ 10 MHz** | 44.94 dB | 0.5467 dB | 46.29 dB | 0.4525 dB |
| **Amp_gain @ 100 MHz** | 31.93 dB | 0.4892 dB | 31.43 dB | 0.5891 dB |
| **Amp_gain @ 1 GHz** | 10.81 dB | 0.6675 dB | 2.439 dB | 1.259 dB |

Table 31: Variation results from 1000-sample postlayout Monte Carlo simulation

There are not many trends that can be drawn from this postlayout data, so just in terms of variation, it is difficult to determine if one design is better than the other.

The input capacitance $C_{in}$ of each postlayout noise buffer was characterized using a transient simulation. Then, the noise source subcircuit was connected to the input of the buffer and a transient noise simulation was run to characterize the rms of the noise signals at the output of each block. The results are compiled in Table 32.

| Parameter | Postlayout Modified FVF | Postlayout SSF |
|---|---|---|
| **Noise buffer $C_{in}$** | 10.76 fF | 2.8578 fF |
| **Noise source out RMS** | 0.5034 mV | 1.428 mV |
| **Noise buffer out RMS** | 0.5901 mV | 1.996 mV |
| **Noise OTA out RMS** | 71.190 mV | 122.78 mV |

Table 32: RMS results from postlayout transient noise simulation

Like in the prelayout simulations, the rms voltage of the modified FVF design is worse than that of the SSF design at every stage. The RMS of the modified FVF design has degraded even further with the increased input cap from the layout.

The autocorrelation functions were extracted, fitted, and tabulated in Table 33. Like the prelayout simulations, a separate autocorrelation function was extracted from the simulations decoupling the SSF from the bias circuit as well.

| Speed | Design (Postlayout) | K ($s^{-1}$) | A |
|---|---|---|---|
| **b6** | Modified FVF | 6.503e07 | 0.913 |
| | Decoupled SSF | 2.002e08 | 0.960 |
| | SSF | 2.652e08 | 0.933 |
| **b5** | Modified FVF | 6.105e07 | 0.908 |
| | Decoupled SSF | 1.711e08 | 0.947 |
| | SSF | 1.864e08 | 0.912 |
| **b4** | Modified FVF | 6.049e07 | 0.948 |
| | Decoupled SSF | 1.602e08 | 0.960 |
| | SSF | 1.479e08 | 0.900 |
| **b0** | Modified FVF | 4.665e07 | 1.026 |
| | Decoupled SSF | 1.659e08 | 0.966 |
| | SSF | 9.860e07 | 0.868 |
| **slowest** | Modified FVF | 5.294e07 | 1.025 |
| | Decoupled SSF | 1.635e08 | 0.981 |
| | SSF | 9.374e07 | 0.884 |

Table 33: Results of autocorrelation function extraction of postlayout designs

The K coefficient for the postlayout modified FVF design was significantly smaller than that of the SSF. Even compared to the prelayout modified FVF, the speed of the postlayout modified FVF was only 65% of the speed in prelayout. The difference is most likely due to the increase in the input capacitance, causing another degradation of the RMS when moving from the prelayout to postlayout modified FVF, as well as coupling capacitors between the input and the Vfb node of the circuit that provide a feedforward path. The layout was made with those coupling capacitors in mind, trying to minimize them by increasing the separation of the metal lines and moving the Vin line away from the Vfb line; however, there was still enough coupling capacitance from the parasitics of the input transistor itself to cause the performance to degrade.

### 6.4 Postlayout Modified FVF Conclusion

As the postlayout modified FVF was only able to achieve 65% of the speed of the prelayout design, the layout of the modified FVF needs to be reworked with a better understanding of what degrades the performance of the buffer and the integrated neuron. Currently, it is known that the coupling capacitor between the input and Vfb node degrades the performance significantly, but there may be other factors involved too, such as the coupling capacitor between Vout and Vfb, or Vin and Vout, etc. It is also difficult to tell if the results are statistically significant without more data and knowledge about the autocorrelation function and its relationship with the integrated neuron components. Without thoroughly modelling and understanding what affects the autocorrelation the most, it will be difficult to redo the layout and/or redesign the modified FVF to make the topology more viable.

## 7 Conclusion and Future Work

The PASSOv1 processor utilizes a fabric of 256 stochastic integrated neurons to solve NP hard problems such as travelling salesman, integer factorization, and max-cut more quickly and efficiently than current state of the art solutions using compute heuristics with clock-driven approaches. To increase the size of the problem the processor can solve and the speed at which it can solve them, the number of integrated neurons needs to be scaled up, consuming more power and area on the chip, and the performance of the chip improved. This report presents an area, power, and performance analysis of the PASSOv1 processor's analog core, detailing the blocks that should be targeted first when trying to reduce power (noise OTA or noise buffer), reduce area (capdac), or improve performance (vecmul for delay), so that future iterations of the chip can redesign those blocks and be able to scale the size of the problem they can compute.

This report also explored one potential design topology, the FVF, for reducing the power consumption of the noise buffer subcircuit used in the integrated neurons. As with many design tradeoffs, the modified FVF's speed is lowered for a drop in power consumption. There is still potential to use it in the future if that cost can be accepted in exchange for power savings. Some of the performance analysis was inconclusive due to the lack of understanding of the autocorrelation function generated by the integrated neurons. To determine how to improve integrated neuron performance, some further investigation and modelling is warranted in the future to identify what in the integrated neuron affects the autocorrelation.

At the time of the end of this project, the PASSOv1 processor has been fabricated and received by the Salahuddin group and is now awaiting packaging so that the team can test, validate, and benchmark the processor. This work will continue to occur beyond the termination of this MS project, and the team has plans for next generation chips aiming for more complex systems with even better performance. This project offers a promising guide for the redesign process to improve upon the power consumption, area consumption, and performance of the PASSOv1 so that future iterations of the chip can continue to further accelerate the solving of NP-hard problems.

# References

[1] Datar, A. (2021). Digital System Design and Fullchip Integration for Asynchronous Stochastic Neural Accelerator [EECS Department, University of California, Berkeley]. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2021/EECS-2021-161.html

[2] M. Hayashi, M. Yamaoka, C. Yoshimura, T. Okuyama, H. Aoki and H. Mizuno, "An Accelerator Chip for Ground-State Searches of the Ising Model with Asynchronous Random Pulse Distribution," 2015 Third International Symposium on Computing and Networking (CANDAR), 2015, pp. 542-546, doi: 10.1109/CANDAR.2015.64.

[3] T. Wang and J. Roychowdhury, "Oscillator-based Ising Machine," arXiv preprint, arXiv:1709.08102

[4] R. G. Carvajal et al., "The flipped voltage follower: a useful cell for low-voltage low-power circuit design," in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 52, no. 7, pp. 1276-1291, July 2005, doi: 10.1109/TCSI.2005.851387.

[5] P. R. Gray, P. J. Hurst, S. H. Lewis, and R. G. Meyer, Analysis and Design of Analog Integrated Circuits, 5th ed. Wiley, 2009.